

System and Method for Orchestration of Multi-Agent Operations Using Language Models

Technical Field

The present application relates generally to artificial intelligence and machine learning systems, and in particular to systems and methods for orchestrating multi-agent operations using language models.

Background

In the current data-driven landscape, organizations across various industries are encountering increased challenges in managing, securing, and deriving actionable insights from their extensive and rapidly expanding analytical data assets. Traditional methods of data governance, access control, and exploration are insufficient to handle the volume, diversity, and complexity of today's big data environments. These systems struggle with manual and non-scalable data feature discovery and cataloguing, as well as a lack of granularity in data access control mechanisms required for effective management in complex, distributed ecosystems. Consequently, organizations face a higher risk of data breaches, unauthorized access, and underutilization of data assets.

Furthermore, traditional keyword-based search methods fail to capture the intricate semantic relationships and nuanced meanings within large datasets, impeding users from formulating queries that accurately reflect their intentions. This leads to subpar data exploration and analysis outcomes. The challenges also extend to maintaining data integrity and provenance as data undergoes numerous transformations and moves across various platforms. Establishing a robust, tamper-evident audit trail is increasingly challenging, eroding trust in analytical insights and raising the risk of non-compliance with regulatory standards. Additionally, the vast scale, distributed nature, and complexity of modern data ecosystems often overwhelm conventional systems, which struggle to deliver the necessary performance, scalability, and operational resilience to support real-time, high-concurrency analytical workloads across hybrid and multi-cloud environments.

These systemic shortcomings necessitate a fundamental overhaul of existing approaches. Outdated conventional manual metadata management techniques, rule-based access control models, and keyword-based search methods often yield incomplete or irrelevant results and fail to ensure the security and integrity of data throughout its lifecycle. Adapting systems to the scale, distribution, and heterogeneity of today's data landscapes is increasingly crucial.

Therefore, it is desirable to provide systems and methods that leverage the capabilities of large language models (LLMs), generative AI (GenAI), quantum computing, and advanced

machine learning techniques to address the disadvantages or limitations of existing technologies or, at the very least, offer the public a useful alternative.

Summary

Embodiments herein provide new and useful systems and methods for orchestrating multi-agent operations using language models in artificial intelligence environments.

In broad terms, the present disclosure proposes a multi-agent system for processing information, including a data processing agent configured to ingest and normalize raw data inputs to produce standardized data and a standards integration agent configured to apply reporting standards into the standardized data thereby generating integrated reporting standards. The system further includes a performance alignment agent configured to align performance indicators based on the standardized data and the integrated reporting standards. The system further includes an information synthesis agent configured to process narrative information from the standardized data and the integrated reporting standards. The system further includes an orchestration framework configured to manage operations of the data processing agent, the standards integration agent, the performance alignment agent, and the information synthesis agent to produce a regulatory report compliant with regulatory requirements, wherein the orchestration framework is executable by a large language model.

In embodiments, the data processing agent further includes a data cleansing module capable of removing errors and inconsistencies from the raw data inputs to improve the accuracy of the standardized data.

In embodiments, the system further includes a materiality alignment agent configured to evaluate and align materiality and boundaries based on the standardized data and the integrated reporting standards.

In embodiments, the system further includes a compliance alignment agent configured to align assurance processes based on the standardized data and the integrated reporting standards.

In embodiments, the performance alignment agent uses machine learning to dynamically adapt the performance indicators based on updates to the reporting standards and corresponding real-time data.

In embodiments, the system further includes a report consolidation agent configured to employ a data integration platform for merging and alignment of output data from the data

processing agent, the standards integration agent, the performance alignment agent and the information synthesis agent.

5 In embodiments, the orchestration framework further includes a scheduling module that adjusts a sequence and priority of tasks based on real-time assessments of data processing needs and agent capacity.

In embodiments, the information synthesis agent further includes a contextual analysis module configured to integrate contextual cues from the standardized data and the integrated reporting standards.

10 The present disclosure further proposes a system for managing metadata, including an ingestion module configured to preprocess data inputs in compliance with a metadata standard, an extraction module configured to employ language processing techniques to extract and normalize metadata from the preprocessed data inputs, an abstraction module configured to transform the extracted and normalized metadata into structured schemas, a mapping module configured to use artificial intelligence techniques to translate the
15 transformed metadata based on taxonomies, and a storage module configured to index and enable searches on the translated metadata.

In embodiments, the mapping module is further configured to utilize ontology-based reasoning and semantic similarity measures to establish mappings between metadata elements from different taxonomies.

20 In embodiments, the mapping module is further configured to employ neural networks and transfer learning to refine and enhance the translation of metadata between taxonomies.

In embodiments, the storage module includes a vector database configured to store the translated metadata, an indexing module configured to generate vector embeddings corresponding to the translated metadata, and a search engine module configured to use
25 the vector embeddings for performing searches based on contextual relevance and semantic similarity.

In embodiments, the system for managing metadata further includes a data lineage and provenance tracking module configured to capture an audit trail of metadata management processes across the system for compliance with regulatory requirements.

30 In embodiments, the ingestion module is further configured to directly interface with external data sources to automatically retrieve the data inputs.

The present disclosure further proposes a system for extracting and normalizing metadata, including an input interface configured to receive preprocessed data inputs that comply with

a metadata standard, a processing module configured to apply natural language processing techniques to extract metadata from the received preprocessed data inputs, a normalization engine configured to normalize the extracted metadata according to predetermined metadata standards, a refinement module configured to enhance the normalized metadata with additional information derived from external sources for generating a refined metadata, and an output interface configured to output the refined metadata for additional processing within a metadata management system.

The present disclosure further proposes a system for integrating data patterns into a data representation, including a data input interface configured to receive data from a plurality of sources, a feature extraction module configured to extract features within the received data, a security module configured to encrypt the extracted features to generate a secured data representation, a data integration module configured to integrate the encrypted features corresponding to the secured data representation into a unified data representation, and a validation module configured to validate the unified data representation against a predefined standard or regulation.

In embodiments, the security module employs cryptographic hash functions to verify authenticity and integrity of the secured data representation.

In embodiments, the feature extraction module is configured to use natural language processing and machine learning algorithms to extract features from the received data.

In embodiments, the system for integrating data patterns includes an access management module configured to translate at least one of user roles, user requirements, or data access patterns into a unified mathematical representation.

In embodiments, the access management module is further configured to use natural language processing and machine learning to encode the user roles and access privileges into unique numerical codes.

In embodiments, the access management module includes a control engine, which serves as an authority for enforcing access controls based on a unified security access model.

In embodiments, the access management module is configured to automatically synchronize its access control rules with external compliance monitoring systems to maintain adherence to changes in regulations.

The present disclosure further proposes a method for processing information in a multi-agent system, including ingesting raw data inputs via a data processing agent and normalizing the ingested raw data to produce standardized data using said data processing agent. The

method further includes applying reporting standards to the standardized data using a standards integration agent to generate integrated reporting standards, aligning performance indicators based on the standardized data and the integrated reporting standards using a performance alignment agent and processing narrative information from the standardized data and the integrated reporting standards using an information synthesis agent. The method further includes orchestrating operations of the data processing agent, the standards integration agent, the performance alignment agent, and the information synthesis agent through an orchestration framework to produce a regulatory report compliant with regulatory requirements, wherein the orchestration framework is executable by a large language model.

The above description is provided as an overview of some implementations of the present disclosure. Further description of those implementations, and other implementations, are described in more detail below.

Brief Description of the Drawings:

Embodiments of the invention will now be explained for the sake of example only, with reference to the following figures in which:

FIG. 1 is a functional block diagram illustrating example services and functionalities within a multi-agent artificial intelligence framework, according to an embodiment herein.

FIG. 2 is a functional block diagram illustrating an example of a reporting integration system within the multi-agent artificial intelligence framework according to an embodiment herein.

FIG. 3 is a functional block diagram illustrating an example of a metadata management system within the multi-agent artificial intelligence framework, according to an embodiment herein.

FIG. 4 is a functional block diagram illustrating an example of a data management and access control system within the multi-agent artificial intelligence framework, according to an embodiment herein

FIG. 5 is a block diagram illustrating an example computer system which may be configured to implement the systems and methods as disclosed herein.

Detailed Description

Embodiments will now be discussed with reference to the accompanying FIGs, which depict one or more exemplary embodiments. These embodiments are described in sufficient detail to enable those skilled in the art to practice the embodiments and it is to be understood that mechanical, logical, and other changes may be made without departing from the scope of the embodiments. Therefore, embodiments may be implemented in many different forms

and should not be construed as limited to the embodiments set forth herein, shown in the FIGs, and/or described below.

As used in this disclosure, the terms “component,” “module,” “system,” “apparatus,” “interface,” “agent,” or the like are generally intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component or a module may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a controller and the controller can be a component or a module. One or more components/modules may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers.

Unless otherwise defined, all terms (including technical and scientific terms) used herein are to be interpreted as is customary in the art. It will be further understood that terms in common usage should also be interpreted as is customary in the relevant art.

A system and method for implementing neuro-symbolic knowledge representation called GenFoundry (Generative Foundation for Metadata Management, Reporting Alignment, and Data Access Control) is disclosed herein. In embodiments, the GenFoundry system and method combines the strengths of neural networks and symbolic reasoning to represent knowledge. It mirrors the brain's ability to encode and retrieve distributed conceptual representations. GenFoundry forms the foundation for COGNIGEN-AX, a neuromorphic reasoning module inspired by the brain's distributed conceptual representations. GenFoundry can encode and retrieve contextualized knowledge fragments as high-dimensional vector embeddings, using the distributional semantics and compositional abilities of Large Language Models (LLMs). This integration of neural and symbolic approaches captures the nuanced relationships and affordances in the data, just like the brain's ability to construct rich situational models.

In embodiments, the GenFoundry system may be part of a neuromorphic architecture that aims to replicate the complex cognitive abilities of the human brain and nervous system. This architecture, known as brain-inspired computing, combines large language models (LLMs), quantum computing, and advanced neuro-symbolic knowledge representation frameworks. Together, these elements overcome the limitations of classical computing architectures when dealing with complex, unstructured data.

In embodiments, GenFoundry may combine the computational strengths of neural networks with the precision of symbolic reasoning, similar to how the brain encodes and retrieves distributed conceptual representations. As the foundational neuro-symbolic knowledge base, GenFoundry may encode contextual knowledge as high-dimensional vector embeddings. It may utilize the distributional semantics and compositional abilities of LLMs to capture nuanced relationships and construct situational models, resembling the brain's conceptual processing. GenFoundry has the ability to capture and encode complex information in a structured framework, including rich meanings, relationships, and affordances present in the data. GenFoundry may support other components, such as COGNIGEN-AX, which utilizes the neuro-symbolic outputs provided by GenFoundry for dynamic, metacognitive processing, and episodic narrative construction.

Integration with Other Neuromorphic Components:

Neuromorphic Reasoning Core (COGNIGEN-AX): This component complements GenFoundry by utilizing its neuro-symbolic outputs to model the brain's metacognitive processes. It weaves multimodal sensory data into coherent narratives, mirroring the neural processes of the prefrontal cortex.

Quantum Neural Networks (Quantum Computing Optimization Framework): This framework enhances the capabilities of GenFoundry's outputs by processing narratives in a quantum-enhanced format. It explores multiple interpretive pathways simultaneously, uncovering deeper semantic associations that classical methods cannot achieve.

In embodiments, GenFoundry is a multi-agent neural architecture designed for automated metadata management, dynamic data access control, intelligent semantic search, and comprehensive data governance across analytical datasets. GenFoundry seamlessly integrates and coordinates advanced AI agents such as MEGAN (Metadata Extraction, Generation, and Alignment Network), SEPHYR (Self-Evolving Pattern Harmonization for Unified Reporting), and SEPHYR (Self-Evolving Pattern Harmonization for Unified Reporting) and supporting modules.

GenFoundry's containerized, cloud-native architecture seamlessly coordinates various components, such as LLM agents, generative AI agents, metadata management, quantum encryption, controlled semantic query processing, and cryptographic auditing. A Petri-net-based neural architecture orchestrates the interactions among these agents. The system dynamically adapts based on data changes, user feedback, regulations, and new information sources.

This unified application of AI, quantum computing, and distributed systems creates a flexible framework for governing analytical datasets. It provides powerful query and data exploration capabilities aligned with enterprise data policies, user privileges, and regulatory mandates. The coordinated neural choreography combines metadata automation, quantum encryption, query interpretation, and cryptographic integrity auditing into a trusted solution for scalable data-driven analytics.

LLM-Driven Automated Metadata Discovery and Enrichment: GenFoundry may use large language models and deep learning to automate the discovery, extraction, normalization, and semantic enrichment of metadata from various data sources. It generates ISO 11179-compliant metadata repositories without manual intervention or predefined taxonomies, overcoming the limitations of traditional metadata management approaches.

Mathematical Encoding of Metadata within a Unified Feature Space: GenFoundry introduces a unique approach to encoding metadata features and their relationships using dynamic number sequence representations. This encoding enables efficient indexing, mapping, and semantic similarity search, and serves as the foundation for integrating access control policies and query processing within the metadata feature space.

Fusion of Semantically-Aware Query Processing and Policy Enforcement: GenFoundry introduces a novel approach to intelligent query processing. It leverages LLMs and generative AI to interpret natural language queries based on the semantic context and domain knowledge encoded in the metadata feature space. It then combines this semantic understanding with mathematically encoded security policies and user privileges to filter and retrieve authorized data insights that align with the user's true intent.

Coordinated Multi-Agent Neural Architecture: GenFoundry establishes a pioneering multi-agent neural architecture that coordinates and orchestrates various components such as LLM agents, generative AI models, quantum computing components, encryption engines, semantic query processors, and audit logging modules. This integrated approach brings together AI, quantum information, data security, and distributed systems principles in a cohesive and extensible architecture for trusted data-driven analytics.

Continuous AI Model Refinement and System Adaptation: The Petri-net-based neural architecture in GenFoundry incorporates neural architecture search, automated model tuning, and reinforcement learning. This allows the system to continuously extend, refine, and optimize AI agents, query interpretation models, security policy enforcement, and overall system capabilities based on evolving data landscapes, user feedback, expanded training data, and changes to governance or regulatory mandates. This self-improving and self-regulating AI fabric sets the Genfoundry system and method apart from static approaches.

GenFoundry is a multi-agent neural architecture that integrates and coordinates advanced AI modules to enable automated metadata management, dynamic data access control, intelligent semantic search, and comprehensive data governance across diverse analytical datasets. At its core, GenFoundry orchestrates the following key components:

- 5 ALFINI (Alignment, Language Models, Financial and Non-Financial Reporting Integration): ALFINI employs a synergistic ensemble of AI agents to align financial and non-financial reporting practices based on the ISO 5116-3:2021 standard. These agents leverage natural language processing and machine learning to analyze, assess, and iteratively enhance the harmonization of reporting information.
- 10 SEPHYR (Self-Evolving Pattern Harmonization for Unified Reporting): SEPHYR harmonizes diverse data patterns and features into a unified, cryptographically verifiable representation. Its agents utilize mathematical algorithms to encode features for efficient cataloguing and search. SEPHYR ensures data lineage, provenance, and trustworthiness through digital native mathematical representations, consensus mechanisms, and cryptographic verifiability.
- 15 GAMA (Governance-Aware Multilevel Access Management Architecture): GAMA orchestrates the mathematical encoding of user roles, permissions, and regulatory policies into integrated access control models. It employs agents like RoleIdentifier, RoleEncoder, AccessClassifier, and RegulationConsensus to enforce fine-grained, context-aware access policies that dynamically adapt to evolving organizational structures, data sensitivity levels,
20 and compliance requirements.
- 25 MEGAN (Metadata Extraction, Generation, and Alignment Network): MEGAN automates the generation of ISO 11179 and UN/CEFACT-compliant metadata registries. It leverages advanced natural language processing agents to extract, normalize, enrich, and align metadata from diverse data sources. MEGAN also abstracts DataVault2.0 schemas and
30 enables cross-taxonomy mapping and translation to facilitate data interoperability.
- 30 QSEM (Quantum Security and Encryption Module): QSEM provides robust data security using quantum encryption algorithms, quantum key distribution, and post-quantum cryptography. It employs quantum random number generation and quantum state mapping techniques to enhance encryption strength. Additionally, QSEM ensures data integrity
30 through quantum-resilient digital signatures.
- 30 CATM (Cryptographic Audit Trail Module): CATM generates immutable audit trails that cryptographically capture the provenance of data elements throughout their lifecycle, from originating sources to analytical consumption. It leverages quantum computing techniques

for efficient provenance verification at scale and integrates with distributed ledger technologies for transparent, multi-party governance of audit trails.

NODM (Neural Orchestration and DevOps Module): NODM provides a cloud-native architecture for deploying and orchestrating GenFoundry modules across hybrid, multi-cloud environments. It utilizes Petri-net-based choreography workflows to coordinate the modules' intricate interactions and data flows. NODM also automates continuous delivery pipelines, enabling seamless updates and extensions to the modules.

In embodiments, the LLM provides a central intelligence in GENFOUNDARY (ALFINI, MEGAN, SEPHYR, etc.), dynamically invoking, implementing, and coordinating various agents to process prompts, perform specific tasks, and interact with COGNIGEN-AX. Its role involves interpreting input prompts, understanding their context and requirements, and activates the relevant agents to generate the appropriate output. In embodiments, the LLM constructs and adapts Petri net models at runtime based on the specific requirements of each prompt, its responses, and the LLM's neutral state. It utilizes its understanding of the prompt and the available agents to determine the necessary processing steps, the order of agent invocation, and data flows between them. The LLM designs the Petri net by defining places, transitions, and arcs that represent data flow or control between places and transitions. It assigns appropriate agents to the transitions, specifying their required inputs and produced outputs. As the prompt is processed, the LLM dynamically executes the Petri net by firing transitions (invoking agents) based on the availability of required input data and fulfillment of necessary conditions. Tokens representing data or control flow move through the Petri net, triggering agent execution and enabling information flow between them. The LLM may continuously monitor Petri net execution, tracking agent progress, handling exceptions or errors, and adapting the Petri net structure as needed based on intermediate results and changing prompt requirements. This dynamic adaptation optimizes processing flow, efficiently allocates resources, and ensures timely and accurate output generation. Throughout processing, the LLM may leverage formal properties of Petri nets such as reachability, boundedness, and liveness to verify the correctness and efficiency of agent interactions and detect/resolve potential deadlocks or performance bottlenecks. Ongoing research involves modeling neuromodulatory effects and integrating them into orchestration models and network data, including the development of neuromorphic data structures to capture the thought patterns of the LLM.

In embodiments, there may be a single master LLM agent that serves as the central orchestrator and coordinator for various specialized agents (ALFINI, MEGAN, SEPHYR, etc.). This master LLM agent typically operates on the most neutral and powerful model,

such as Opus, to handle the overall orchestration and management of the neural state. It interprets the input prompt, understands its context and requirements, and dynamically creates and executes Petri-net models to coordinate the interactions and workflows of the specialized agents. The master LLM agent assigns the appropriate agents to the transitions in the Petri network and monitors its execution, adapting as needed based on the intermediate results and changing requirements of the prompt. However, for specific structured tasks where creativity needs to be limited or constrained, the master LLM agent can invoke "student" agents through platforms like AWS Bedrock. These student agents operate on lower models, such as Sonnet, which are more suitable for focused, specific tasks requiring less creativity and more structured outputs. The master LLM agent determines when to invoke these student agents based on the nature of the task and the desired level of creativity or constraint. It communicates with the student agents, providing the necessary input data and instructions, and integrates their outputs into the overall workflow coordinated by the Petri-net model. While the student agents perform their specific tasks using the lower models, the overall orchestration and management of the neural state remain under the control of the master LLM agent operating on the most powerful model. The master LLM agent ensures the coherence and consistency of the neural state across the different agents and models, maintaining a unified and contextualized representation of the ongoing tasks and their progress. By leveraging this hierarchical architecture, with a master LLM agent coordinating student agents on lower models for specific structured tasks, the system can balance creativity and constraint, optimize the allocation of computational resources, and ensure the generation of accurate and contextually relevant outputs.

FIG. 1 is a functional block diagram illustrating example services and functionalities within a multi-agent artificial intelligence framework, according to an embodiment herein. The multi-agent artificial intelligence framework corresponds to an embodiment of the GenFoundry neural architecture.

As shown in FIG. 1, Regulation may set the standards and guidelines that influence the overall system, impacting various services directly and indirectly to ensure compliance and governance within the framework. ALFINI may enable the alignment and harmonization of financial and non-financial reporting practices according to ISO 5116-3:2021 standards, utilizing natural language processing and machine learning agents. ALFINI may interact directly with both MEGAN and SEPHYR to integrate harmonized reporting practices into broader metadata management systems. SEPHYR may harmonize diverse data patterns and features into a unified, cryptographically verifiable representation through mathematical algorithms and consensus mechanisms. SEPHYR facilitates efficient cataloging, search, and metadata management and feeds into the GAMA system for

further processing of access control models. MEGAN 108 may automate the generation of metadata registries compliant with ISO 11179 and UN/CEFACT. MEGAN 108 processes extraction, normalization, enrichment, and alignment of metadata from various sources, aiding in the creation of structured and interoperable data sets. GAMA 114 orchestrates the mathematical encoding of user roles, permissions, and regulatory policies into integrated access control models. It employs fine-grained, context-aware policy enforcement agents to manage access and security protocols within the system. QSEM 116 may provide quantum encryption, key distribution, and post-quantum cryptography mechanisms. QSEM 116 secures data throughout the system, complemented by quantum random number generation and state mapping techniques to ensure the integrity and security of data transmissions and storage. CATM 112 generates cryptographic immutable audit trails that capture data provenance throughout the data lifecycle. CATM 112 utilizes quantum computing for verification at scale and integrates with distributed ledger technologies to enhance the reliability and traceability of data transactions. NODM 110 handles the Petri-net choreography for the deployment and orchestration of modules across LLM agents. NODM 110 integrates automated DevOps pipelines to streamline operations and maintain system efficiency and scalability.

Relationship between ALFINI 104 and SEPHYR 106:

ALFINI 104 leverages SEPHYR's 106 cryptographically verifiable data representations. ALFINI employs a synergistic ensemble of AI agents to analyse, assess, and iteratively enhance the harmonization of reporting information based on the ISO 5116-3:2021 standard. These agents leverage natural language processing and machine learning techniques to align financial and non-financial reporting practices. SEPHYR 106, on the other hand, harmonizes diverse data patterns and features into a unified, cryptographically verifiable representation. Using mathematical algorithms, it utilizes agents to encode features. This enables efficient cataloguing, search, and metadata management while ensuring data lineage, provenance, and trustworthiness through consensus mechanisms and cryptographic verifiability. The association relationship between ALFINI 104 and SEPHYR 106 signifies that ALFINI 104 leverages or utilizes the cryptographically verifiable data representations and mathematical algorithms provided by SEPHYR 106 to enhance the alignment and harmonization of reporting information. SEPHYR's 106 capabilities in encoding data patterns, establishing consensus, and ensuring cryptographic verifiability can complement and support ALFINI's 104 efforts to align financial and non-financial reporting practices. While ALFINI 104 focuses on the reporting alignment aspect using AI agents, it can benefit from associating with SEPHYR's 106 harmonized data representations,

consensus mechanisms, and cryptographic techniques to enhance the overall reporting integration process.

Relationship between ALFINI 104 and MEGAN 108:

5 MEGAN 108 and ALFINI 104 are associated as they both deal with extracting, aligning, and harmonizing metadata and reporting information in different contexts and using various techniques. MEGAN 108 automates extracting, normalizing, enriching, and aligning metadata from diverse sources to generate ISO 11179 / UN/CEFACT-compliant metadata registries and facilitate data interoperability. ALFINI 104 employs AI agents leveraging natural language processing and machine learning to analyse and align financial and non-
10 financial reporting information based on the ISO 5116-3:2021 standard. While MEGAN 108 focuses on metadata extraction and alignment, and ALFINI 104 on reporting alignment, both involve harmonizing data from disparate sources using advanced techniques. Their outputs of aligned metadata registries and harmonized reporting could be complementary.

Relationship between MEGAN 108 and SEPHYR's GAMA 114:

15 MEGAN 108 provides metadata inputs and data models that influence GAMA's access control models and policies. MEGAN may generate 108 ISO 11179 and UN/CEFACT compliant metadata registries, DataVault2.0 schemas, and cross-taxonomy mappings through agents. These outputs capture the structure, relationships, and semantics of the data elements managed within the GenFoundry system. GAMA 114 utilizes this detailed
20 metadata to translate user roles, permissions, and regulatory policies into robust, integrated access control models. The access control models and security policies enforced by GAMA 114 are directly shaped by the metadata, schemas, and data models provided by MEGAN 108, ensuring that GAMA's 114 access controls are well-informed and securely applied across various data landscapes.

25 Relationship between GAMA 114 and QSEM 116:

GAMA's 114 access control models depend on QSEM 116 for encryption and key management, vital for ensuring data security and compliance. GAMA 114, or the Governance-Aware Multilevel Access Management Architecture, orchestrates the mathematical encoding of user roles, permissions, and regulatory policies into sophisticated
30 access control models. It leverages fine-grained, context-aware policy enforcement agents such as RoleIdentifier, RoleEncoder, AccessClassifier, and RegulationConsensus. These capabilities are secured by QSEM 116, which provides quantum encryption algorithms, key distribution, and post-quantum cryptography mechanisms. QSEM's 116 quantum security solutions are crucial for GAMA 114 to protect its access control models, maintain the

confidentiality of user roles, and ensure the integrity of sensitive data elements. By relying on QSEM's 116 advanced security mechanisms, GAMA 114 achieves a high level of security and resilience, essential for protecting against potential quantum computing threats and ensuring long-term data security.

5 Relationship between MEGAN 108 and CATM 112:

MEGAN (Metadata Extraction, Generation, and Alignment Network) provides essential metadata inputs that CATM (Cryptographic Audit Trail Module) uses to generate audit trails. MEGAN automates the generation of ISO 11179 and UN/CEFACT-compliant metadata registries, abstracts DataVault2.0 schemas, and facilitates cross-taxonomy mapping and
10 translation. This process involves extracting, normalizing, enriching, and aligning metadata from diverse sources through agents. A key output from MEGAN is the metadata and data lineage information, particularly captured by the DLPTA (Data Lineage and Provenance Tracking Agent), which tracks the origin, transformations, and evolution of data elements. CATM relies on this comprehensive metadata and data lineage information to create
15 immutable audit trails that cryptographically document the entire lifecycle of data elements, effectively fulfilling its role in audit trail generation and data provenance tracking.

Relationship between CATM 112 and NODM 110:

The auditing needs dictated by CATM, such as data provenance tracking, cryptographic verification, and multi-party governance, significantly influence the deployment and
20 orchestration strategies employed by NODM. These requirements guide NODM in its management of deployment, configuration, and maintenance of systems across the hybrid cloud infrastructure. CATM's demands for audit trail generation, verification, and governance act as both constraints and drivers, shaping how NODM provisions resources, manages configurations, and automates delivery pipelines for the GenFoundry modules, ensuring that
25 CATM's auditing capabilities are effectively integrated and maintained.

NODM orchestration:

NODM orchestrates the deployment and integration of various modules across the infrastructure, ensuring seamless interaction and operational coherence among ALFINI, SEPHYR, GAMA, MEGAN, and QSEM. This role involves carefully managing the setup and
30 ongoing operations of these modules to support the system's overall functionality and performance, aligning each module's specific capabilities and requirements with the strategic objectives of the GenFoundry system.

Use Case/Petri-Net Example:

The Financial Stability Board (FSB), in collaboration with the International Organization for Standardization (ISO), has introduced a new regulation to enhance the transparency, consistency, and reliability of financial and non-financial reporting by organizations. The regulation mandates the use of advanced technologies, such as artificial intelligence (AI) and quantum computing, to automate and streamline the reporting process while ensuring compliance with internationally recognized standards, such as ISO/IEC 15909-1:2019 for Petri net modeling.

The regulation emphasizes the integration of financial and non-financial data, aligning key performance indicators (KPIs), narrative information, materiality assessments, and assurance processes to provide a comprehensive and coherent view of an organization's performance and compliance. The ultimate goal is to improve the quality and trustworthiness of regulatory reporting, enabling better decision-making by stakeholders and more effective oversight by regulatory bodies.

Using GENFOUNDRY, COGNIGEN-AX, and Quantum in the Design: To meet the requirements of this new regulation, organizations can leverage the powerful capabilities of GENFOUNDRY, a multi-agent AI framework, in combination with COGNIGEN-AX, an advanced reasoning core, and quantum computing technologies. These technologies can be orchestrated using a Large Language Model (LLM) to create an intelligent, automated, and compliant regulatory reporting system.

The LLM, trained on vast amounts of financial and regulatory data, can understand the complex requirements of the regulation and design a Petri net architecture that optimally utilizes the GENFOUNDRY agents and COGNIGEN-AX to achieve the desired outcomes. The Petri net, adhering to the ISO/IEC 15909-1:2019 standard, provides a formal, mathematical foundation for modeling the reporting process, ensuring consistency and reliability.

Within the Petri net, GENFOUNDRY's ALFINI agents, such as the Data Harmonization Agent (DHA), Reporting Standards Knowledge Base Agent (RSKBA), and Narrative Information Alignment Agent (NIAA), work together to ingest, harmonize, and align the various types of data required for regulatory reporting. These agents leverage advanced AI techniques, such as natural language processing (NLP), machine learning, and knowledge representation, to extract insights and ensure data quality and consistency.

COGNIGEN-AX, the metacognitive reasoning core, is crucial in evaluating the integrated regulatory report generated by the ALFINI agents. It uses its advanced reasoning capabilities to check the report against the encoded regulatory requirements, identify areas for improvement, and optimize the report iteratively until it meets all the necessary

compliance criteria. COGNIGEN-AX's ability to "think about thinking" allows it to adapt and learn from each iteration, continually enhancing the quality and efficiency of the reporting process.

Quantum computing technologies can be integrated into the design to accelerate complex computations, such as risk assessments, scenario analyses, and optimization tasks. Quantum algorithms can help identify patterns and anomalies in the data more efficiently, enabling faster and more accurate detection of potential compliance issues. Integrating quantum computing with the GENFOUNDRY framework and COGNIGEN-AX creates a powerful, future-proof solution for regulatory reporting.

The LLM orchestrates the entire process, from designing the initial Petri net to monitoring and optimizing the system's performance over time. It can interpret the requirements of the regulation, understand the organization's specific needs and constraints, and adapt the design accordingly. As new regulations emerge or existing ones change, the LLM can quickly modify the Petri net and redeploy the updated system, ensuring continuous compliance.

Petri-Net Design/Agent Orchestration

The Petri net automates and streamlines the process of generating compliant regulatory reports using GENFOUNDRY's AI agents ALFINI and COGNIGEN-AX. It begins by ingesting raw financial and non-financial data (P1) from various sources within the organization.

The first transition, T1, activates the Data Harmonization Agent (DHA) to clean, normalize, and integrate the raw data into a consistent format for further processing. The harmonized data is then stored in P2.

From P2, the data flows to several parallel transitions:

T2: The Reporting Standards Knowledge Base Agent (RSKBA) integrates relevant reporting standards and regulatory requirements from its knowledge base (P3) with the harmonized data.

T3: The Key Performance Indicator Alignment Agent (KPIAA) aligns the relevant KPIs from the harmonized data (P2) and the reporting standards (P3), storing the aligned KPIs in P4.

T4: The Narrative Information Alignment Agent (NIAA) processes and aligns the narrative information from the harmonized data (P2) and the reporting standards (P3), storing the aligned narrative information in P5.

T5: The Materiality and Boundary Alignment Agent (MBAA) assesses and aligns the materiality and reporting boundaries based on the harmonized data (P2) and the reporting

standards (P3), storing the aligned materiality assessments and boundaries in P6.

T6: The Assurance Alignment Agent (AAA) aligns the assurance processes based on the harmonized data (P2) and the reporting standards (P3), storing the aligned assurance processes in P7.

- 5 The outputs from P4, P5, P6, and P7 flow into transition T7, where the Reporting Integration Agent (RIA) consolidates all the aligned information into an integrated regulatory report stored in P8.

- 10 The COGNIGEN-AX reasoning core, represented by transition T8, evaluates the integrated report against the encoded regulatory requirements. This is an iterative process, represented by the self-loop arc from P8 to T8 and back to P8. COGNIGEN-AX utilizes its metacognitive capabilities to critique and optimize the report, ensuring compliance with all necessary criteria.

- 15 Once COGNIGEN-AX determines that the report is fully compliant and coherent, it approves the final version. The Petri net then moves to the final transition, T9, which outputs the approved regulatory report.

Throughout the process, the GENFOUNDRY framework maintains end-to-end traceability and machine-readable semantic encoding, ensuring data lineage and efficient data management and analysis.

- 20 The Petri net architecture follows the standard elements and graphical notations specified in ISO/IEC 15909-1:2019, ensuring compatibility and interoperability with other systems and tools that support this standard. This architecture automates and integrates the different stages of the regulatory reporting process using advanced AI agents and a formal Petri net model. It showcases how GENFOUNDRY can greatly improve the efficiency, accuracy, and compliance of regulatory reporting while reducing manual work and the risk of errors.

- 25 Petri Net: ALFINI Regulatory Reporting System

Places:

P1: Raw financial and non-financial data

P2: Harmonized financial and non-financial data (output of DHA agent)

P3: Integrated reporting standards knowledge base (output of RSKBA agent)

- 30 P4: Aligned key performance indicators (output of KPIAA agent)

P5: Aligned narrative information (output of NIAA agent)

P6: Aligned materiality assessments and boundaries (output of MBAA agent)

P7: Aligned assurance processes (production of AAA agent)

P8: Integrated regulatory report (output of RIA agent)

Transitions:

5 T1: Ingest and harmonize raw financial and non-financial data (DHA agent)

T2: Integrate reporting standards knowledge base (RSKBA agent)

T3: Align key performance indicators (KPIAA agent)

T4: Align narrative information (NIAA agent)

T5: Align materiality assessments and reporting boundaries (MBAA agent)

10 T6: Align assurance processes (AAA agent)

T7: Integrate aligned information into regulatory report (RIA agent)

T8: COGNIGEN-AX reasoning core evaluates report against regulatory requirements

T9: COGNIGEN-AX optimizes and approves the final regulatory report

Arcs:

15 P1 to T1, T1 to P2

P2 to T2, T2 to P3

P2 to T3, P3 to T3, T3 to P4

P2 to T4, P3 to T4, T4 to P5

P2 to T5, P3 to T5, T5 to P6

20 P2 to T6, P3 to T6, T6 to P7

P4 to T7, P5 to T7, P6 to T7, P7 to T7, T7 to P8

P8 to T8, T8 to P8 (self-loop)

P8 to T9

25 The Petri net utilizes the ALFINI agents in a coordinated workflow to ingest raw financial and non-financial data, align it with reporting standards, evaluate materiality, integrate narrative information, align KPIs and assurance processes, and generate an integrated regulatory report.

Then, the COGNIGEN-AX reasoning core assesses this preliminary report against the encoded regulatory requirements in a iterative loop, using its metacognitive capabilities to evaluate and optimize the report until it is fully compliant and coherent.

Finally, the approved report is produced, with the GENFOUNDRY framework maintaining end-to-end traceability and machine-readable semantic encoding throughout the process. The Petri net is defined using the standard elements and graphical notations specified in ISO/IEC 15909-1:2019

Financial Reporting Use Cases

GenFoundry has the potential to revolutionize how banks implement and comply with financial regulations by leveraging advanced AI, quantum computing, and distributed systems technologies. The methods described in this patent could help banks reduce costs, accelerate time-to-market, improve quality, and mitigate operational risks associated with regulatory compliance. Here's how:

Automated Metadata Management and Data Governance: MEGAN can automatically generate ISO 11179 and UN/CEFACT-compliant metadata registries, abstract DataVault2.0 schemas, and enable cross-taxonomy mapping. This streamlines regulatory reporting processes across various jurisdictions and data standards, reducing manual efforts and ensuring consistency. As a result, banks can significantly reduce costs and operational risks associated with data governance and regulatory reporting.

Dynamic Access Control and Data Security: GAMA translates user roles, permissions, and regulatory policies into integrated access control models, ensuring granular, context-aware data access control. This mitigates risks of data breaches and unauthorized access. Additionally, the integration of quantum encryption algorithms and post-quantum cryptography by QSEM provides robust data security, ensuring long-term protection against emerging threats from quantum computing.

Intelligent Semantic Search and Query Processing: SEPHYR harmonizes diverse data patterns and features into a unified, cryptographically verifiable representation. This enables efficient cataloging, search, and metadata management. Furthermore, it enables intelligent, semantically aware query processing, allowing banks to quickly retrieve relevant data insights. This reduces the time and effort required for regulatory reporting and compliance tasks.

Comprehensive Data Lineage and Provenance Tracking: CATM generates immutable audit trails and cryptographically captures data provenance. QUASAR's quantum computing techniques optimize provenance verification, ensuring end-to-end traceability and

transparency. This comprehensive data lineage and provenance tracking significantly enhance regulatory compliance, reduce operational risks, and facilitate audits and investigations.

Scalable and Resilient Deployment: HYDRA automates cloud provisioning, configuration management, and container orchestration. This ensures scalable and resilient deployment of GenFoundry modules across hybrid cloud environments. Moreover, NEMO's Petri-net-based workflow generation and optimization, combined with OASIS's automated continuous delivery, enable efficient and adaptable orchestration of regulatory compliance processes. GenFoundry offers robust capabilities that can streamline compliance with various financial reporting standards and regulations. With its advanced metadata management, data governance, access control, and intelligent query processing, GenFoundry becomes a valuable tool in achieving compliance. Here are some examples of how GenFoundry can facilitate compliance with critical financial reporting standards and regulations:

Basel III Capital and Liquidity Standards: Leveraging MEGAN's ability to generate ISO 11179-compliant metadata registries and abstract DataVault2.0 schemas, banks can establish a consistent and unified data model for risk data aggregation and reporting. This is crucial to meeting the requirements set by Basel III's Principles for Effective Risk Data Aggregation and Risk Reporting (BCBS 239). SEPHYR's harmonization of diverse data patterns and GAMA's dynamic access control capabilities ensure that risk data is consistently cataloged, searchable, and accessible only to authorized personnel, reducing operational risks.

International Financial Reporting Standards (IFRS): ALFINI's alignment of financial and non-financial reporting practices, combined with MEGAN's cross-taxonomy mapping capabilities, helps banks reconcile and integrate accounting data across multiple IFRS jurisdictions and reporting taxonomies (e.g., IFRS Taxonomy, US GAAP Taxonomy). CATM's immutable audit trails and QUASAR's quantum-optimized provenance verification provide a tamper-evident record of data transformations and calculations, thus enhancing the transparency and audibility of IFRS-compliant financial statements.

Solvency II (Insurance Regulation): MEGAN's ability to extract and normalize metadata from diverse sources is instrumental in helping insurance companies establish a consistent data foundation for Solvency II's Pillar 3 reporting requirements. These requirements involve complex calculations and disclosures related to capital adequacy, risk management, and governance. SEPHYR's unified data representation and GAMA's role-based access control

ensure the secure and controlled retrieval of data required for Solvency II reporting, reducing operational risks and ensuring compliance with data privacy regulations.

Markets in Financial Instruments Directive (MiFID II): ALFINI's alignment of key performance indicators and narrative information aids banks and investment firms in complying with MiFID

- 5 II's requirements for comprehensive reporting on best execution, transaction cost analysis, and product governance. CATM's cryptographic audit trails and QUASAR's post-quantum security mechanisms provide a secure and tamper-evident record of trade executions, thus ensuring compliance with MiFID II's audit trail and record-keeping requirements.

Foreign Account Tax Compliance Act (FATCA): MEGAN's ability to extract and normalize
10 customer data from various sources can help financial institutions establish a consistent and up-to-date repository of customer tax information. This, in turn, facilitates compliance with FATCA's reporting requirements. Additionally, GAMA's dynamic access control and QSEM's quantum encryption provide secure handling and protection of sensitive customer tax data. This helps to mitigate the risks of data breaches and non-compliance.

- 15 By leveraging GenFoundry's advanced capabilities in metadata management, data governance, access control, intelligent query processing, and quantum-resilient security, financial institutions can streamline their compliance efforts across a wide range of reporting standards and regulations. Consequently, operational costs are reduced, risks are mitigated, and the quality and suitability of regulatory reporting processes are enhanced.

20 Financial Crime Use Cases

GenFoundry can provide a comprehensive horizontal capability for combating financial crime. Its core component, COGNIGEN-AX, covers multiple domains, including anti-money laundering (AML), fraud detection, and regulatory compliance. The different components of GenFoundry contribute to this horizontal capability in various ways.

- 25 COGNIGEN-AX is an AI-powered agent that generates detailed narratives about potential financial crime scenarios. It analyzes various types of data, such as transaction records, customer profiles, watchlist entries, and open-source intelligence. The tool's metacognitive capabilities enable it to continuously adapt its knowledge models and reasoning strategies. This ensures the accuracy and relevance of the generated narratives as new information
30 emerges or patterns evolve.

MEGAN: Automated Metadata Management and DataVault2.0 Modeling: MEGAN's ability to generate ISO 11179-compliant metadata registries and abstract DataVault2.0 schemas establishes a consistent and unified data foundation for financial crime data. It spans various sources, including transaction monitoring systems, customer due diligence databases, and

external watchlists. The DataVault2.0 modeling approach promotes data historicization and audit trails, facilitating the reconstruction of financial crime narratives by preserving the chronological sequence of events and data transformations.

SEPHYR: Harmonized Data Representation and Cryptographic Verifiability: SEPHYR

5 harmonizes diverse data patterns and features into a unified, cryptographically verifiable representation. This enables efficient cataloging, search, and analysis of financial crime data, facilitating cross-channel monitoring and holistic risk assessments. The cryptographic verifiability provided by SEPHYR ensures the integrity and immutability of financial crime data, supporting robust investigations, audits, and regulatory compliance.

10 GAMA: Dynamic Access Control and Regulatory Compliance: GAMA translates user roles, permissions, and regulatory policies into integrated access control models. This ensures that financial crime data is accessible only to authorized personnel, mitigating the risks of data breaches and unauthorized access. By enforcing fine-grained, context-aware access policies, GAMA supports the segregation of duties and maintains the confidentiality of
15 sensitive financial crime investigations while facilitating cross-functional collaboration.

QSEM and CATM: Quantum-Resilient Security and Auditable Data Provenance: QSEM integrates quantum encryption algorithms and post-quantum cryptography to protect the confidentiality and integrity of financial crime data. This ensures long-term security against emerging threats from quantum computing. CATM generates immutable audit trails and
20 cryptographically captures data provenance. This provides a tamper-evident record of financial crime data, supporting robust investigations, regulatory audits, and forensic analysis.

By integrating these components, GenFoundry creates horizontal capabilities for combating financial crime. These capabilities span data governance, narrative intelligence, access
25 control, security, and auditing. This holistic approach empowers financial institutions to detect, investigate, and report financial crimes more effectively. It also ensures robust data integrity, regulatory compliance, and operational resilience.

ALFINI

ALFINI (Alignment, Language Models, Financial and Non-Financial Reporting, Integration,
30 NLP and AI, ISO 5116-3:2021) is an advanced system and method that efficiently aligns financial and non-financial reporting by ISO 5116-3:2021. At its core is a synergistic ensemble of specialized agents and methods collaborating to analyze, assess, and enhance the harmonization of reporting practices through continuous optimization cycles.

Key ALFINI advantages:

Diverse Implementation Modalities: Accommodating large language models (LLMs), quantum computing, data mesh architecture, and graph and vector data stores for adaptability across computational contexts.

5 Synergistic Agent System and Methods: ALFINI integrates agents like DHA, RSKBA, KPIAA, NIAA, MBAA, AAA, and RIA with unique capabilities to align reporting practices.

Coordinated Alignment Protocol: Rigorously evaluate and iteratively enhance financial and non-financial reporting alignment by integrating multi-agent functionalities to improve transparency, reliability, and decision-usefulness.

10 Adaptive Optimization: ALFINI facilitates system adaptation using advanced technologies and architectures to derive improvement strategies and fine-tune the alignment process.

Agents:

DHA: Analyzes financial and non-financial data for quality and consistency to ensure accuracy, completeness, and timeliness.

15 RSKBA: Integrates principles and guidelines from various reporting frameworks and standards to provide a comprehensive knowledge base for alignment.

KPIAA: Employs machine learning techniques to identify and align key performance indicators (KPIs), ensuring consistency and comparability across reporting periods and entities.

20 NIAA: Utilizes natural language processing (NLP) to align narrative information, such as management commentary and sustainability reports, ensuring consistency and coherence.

MBAA: Applies machine learning algorithms to align materiality assessments and reporting boundaries, enhancing the relevance and comparability of reported information.

AAA Coordinates with assurance providers to align assurance processes, promote consistent and reliable assurance opinions, and enhance credibility and trust.

25 RIA: Integrates aligned information into comprehensive reports, following principles of connectivity, consistency, and accessibility to cater to diverse stakeholder needs.

Evaluation & Validation:

The selection of ground truth data and evaluation metrics is carefully tailored to align with the specific functionalities and objectives of each ALFINI component:

30 DHA (Data Harmonization Agent): DHA's performance is evaluated using ground truth data from established financial and non-financial databases, reporting templates, and expert

annotations. Data accuracy, completeness, and timeliness are employed to assess DHA's effectiveness in ensuring the quality and consistency of the data used for alignment.

RSKBA (Reporting Standards Knowledge Base Agent): RSKBA's evaluation relies on ground-truth data from internationally recognized reporting frameworks, standards, and best practices. Metrics like knowledge base coverage, update frequency, and query response accuracy measure RSKBA's ability to provide comprehensive and up-to-date guidance for aligning reporting practices.

KPIAA (Key Performance Indicator Alignment Agent): KPIAA's performance is assessed using ground truth data from industry benchmarks, historical KPI data, and expert-validated alignments. Metrics such as KPI alignment accuracy, consistency score, and comparability index are employed to evaluate KPIAA's effectiveness in identifying and aligning financial and non-financial KPIs across reporting periods and entities.

NIAA (Narrative Information Alignment Agent): NIAA is evaluated using ground truth data from manually aligned narrative reports, linguistic resources, and domain-specific ontologies. Metrics like thematic consistency, risk and opportunity identification accuracy, and coherence score are used to assess NIAA's ability to align narrative information effectively, ensuring consistency and coherence across reports.

MBAA (Materiality and Boundary Alignment Agent): MBAA's performance is evaluated using ground truth data from stakeholder surveys, industry materiality maps, and expert-defined boundary scenarios. Metrics such as materiality alignment accuracy, boundary consistency, and topic prioritization relevance are employed to measure MBAA's effectiveness in aligning materiality assessments and reporting boundaries.

AAA (Assurance Alignment Agent): AAA's evaluation relies on ground-truth data from assurance standards, historical assurance opinions, and expert-reviewed alignment cases. Metrics like assurance consistency, opinion reliability, and alignment adherence assess AAA's ability to coordinate with assurance providers effectively and promote consistent and reliable assurance processes.

RIA (Reporting Integration Agent): RIA's performance is evaluated using ground truth data from integrated reporting frameworks, stakeholder feedback, and expert-validated report samples. Metrics such as information connectivity, consistency, accessibility, and stakeholder satisfaction measure RIA's effectiveness in integrating aligned information into comprehensive reports catering to diverse stakeholder needs.

Agents:

The ALFINI agent architecture comprises several sub-agents, each with specific roles and responsibilities, working together to ensure compliance with the clauses and guidelines of ISO 5116-3:2021:

5 Data Harmonization Agent (DHA) aligns with Clause 6, ensuring data quality and consistency.

Reporting Standards Knowledge Base Agent (RSKBA) aligns with Clause 5, providing principles and guidelines for alignment.

Key Performance Indicator Alignment Agent (KPIAA) aligns with Clause 7, identifying and aligning KPIs.

10 Narrative Information Alignment Agent (NIAA) aligns with Clause 8, aligning narrative information using NLP techniques.

Materiality and Boundary Alignment Agent (MBAA) aligns with Clause 9, aligning materiality assessments and reporting boundaries using machine learning.

15 Assurance Alignment Agent (AAA) aligns with Clause 10, coordinating with assurance providers to align assurance processes.

Reporting Integration Agent (RIA) aligns with Clause 11, integrating aligned information into comprehensive reports.

20 The sub-agents are orchestrated using a Petri net model, which ensures a coordinated and efficient execution of the alignment and integration tasks by the principles outlined in Clause 4 of ISO 5116-3:2021. The Petri net models the dependencies and interactions between the sub-agents, ensuring that each clause is addressed in the appropriate sequence and with the necessary inputs and outputs.

25 For example, the Petri net ensures that the DHA (Clause 6) provides quality-assured data to the KPIAA (Clause 7) and NIAA (Clause 8) for their respective alignment tasks. The Petri net then triggers the MBAA (Clause 9) to align the materiality assessments and reporting boundaries based on the outputs of the KPIAA and NIAA. Once all alignment tasks are complete, Petri Net initiates the RIA (Clause 11) to integrate the aligned information into the final report. Finally, the Petri net triggers the AAA (Clause 10) to independently ensure the alignment and integration processes.

30 FIG. 2 is a functional block diagram 200 illustrating an example of a reporting integration system within the multi-agent artificial intelligence framework according to an embodiment

herein. The reporting integration system corresponds to an embodiment of the ALFINI agent framework.

DHA (Data Harmonization Agent) 206:

The DHA 206 is a data processing agent that is configured to ingest and normalize raw data inputs to produce standardized data. Additionally, the data processing agent includes a data cleansing module, enhancing its capability to remove errors and inconsistencies, thus improving the accuracy of the standardized data. This agent also acts as the core for the report consolidation agent, employing data integration platforms to merge data outputs.

DHA 206 further analyses and harmonizes financial and non-financial data to ensure quality, consistency, and accuracy. DHA 206 further serves KPIAA 214, NIAA 212, and MBAA 202 by providing harmonized data as inputs to these agents, facilitating their respective functions.

RSKBA Reporting Standards Knowledge Base Agent 210:

RSKBA 210 is a standards integration agent which maintains reporting standards and guidelines from various frameworks. The standards integration agent applies reporting standards into the standardized data, thereby generating integrated reporting standards. This integration ensures that the data aligns with regulatory and compliance requirements, reflecting the functions of the compliance alignment agent. RSKBA 210 forms associations with KPIAA 214, NIAA 212, MBAA 202, and AAA 204 to inform alignment activities based on standards, ensuring compliance across all reporting.

KPIAA Key Performance Indicator Alignment Agent 214:

KPIAA 214 is a performance alignment agent which employs NLP and machine learning to identify and align KPIs across reports. The performance alignment agent may use machine learning to dynamically adapt performance indicators based on the standardized data and integrated reporting standards. KPIAA 214 provides aligned KPIs as inputs to RIA 208, supporting comprehensive report integration and leverages harmonized data from DHA 206 and standards from RSKBA 210, enhancing the accuracy of performance metrics.

NIAA (Narrative Information Alignment Agent) 212:

NIAA 212 is an information synthesis agent that utilizes NLP to align narrative information such as risks and opportunities. This agent processes narrative information from the standardized data and integrated reporting standards. It includes a contextual analysis module that integrates contextual cues from the data, enhancing narrative accuracy and relevance.

NIAA 212 may receive aligned narratives from RIA 208 for integration into comprehensive reports and associates with data from DHA 206 and standards from RSKBA 210, ensuring narratives are contextually accurate and compliant.

MBAA (Materiality and Boundary Alignment Agent) 202:

- 5 MBAA 202 is a materiality alignment agent and applies machine learning to align materiality and reporting boundaries. This agent evaluates and aligns materiality and boundaries based on the standardized data and integrated reporting standards, ensuring that reports meet specific materiality thresholds and boundary definitions as per global standards.

- 10 MBAA may receive aligned materiality and boundary inputs to RIA 208, crucial for report structuring and associates with data and standards from DHA 206, RSKBA 210, KPIAA 214, and NIAA 212, ensuring a comprehensive approach to boundary definition.

AAA (Assurance Alignment Agent) 204:

- 15 AAA is a compliance alignment agent and coordinates with assurance providers to align assurance processes based on the standardized data and integrated reporting standards, ensuring that the reports are compliant and verified.

AAA associates with RSKBA's 210 standards and RIA's 208 processes to ensure reports are not only compliant but also verified, and further ensures that the assurance processes align with the integrated reports crafted by RIA 208.

RIA (Reporting Integration Agent) 208:

- 20 RIA is an orchestration framework that integrates aligned information from various agents into comprehensive reports. The orchestration framework acts as the central hub for receiving inputs from agents like KPIAA 214, NIAA 212, and MBAA 202, consolidating them into final reports. RIA forms associations with other ALFINI agents, utilizing and synthesizing their outputs to produce coherent and comprehensive reporting outputs for ALFINI.

- 25 Summary of relationships shown in FIG. 2:

- DHA 206 provides a harmonized data foundation, crucial for all data-dependent processes. RSKBA 210 offers guidance on standards, crucial for maintaining compliance across functions. KPIAA 214, NIAA 212, and MBAA 202 perform specific alignments using data and standards provided by DHA 206 and RSKBA 210. AAA 204 coordinates assurance processes, ensuring reports meet external and internal assurance criteria. RIA 208 consolidates inputs from all agents, producing the final integrated reports that reflect the comprehensive and collaborative efforts of the ALFINI system.
- 30

ALFINI achieves the following goals and capabilities:

Integrate multi-modal data for comprehensive alignment:

ALFINI has mechanisms to collect, process, and incorporate data from various sources, such as financial statements, sustainability reports, performance metrics, and external frameworks. Considering diverse data modalities, the system and methods give AI agents a holistic view of the reporting landscape and enable them to identify patterns, correlations, and insights that may not be apparent from a single data source. Furthermore, ALFINI has techniques for weighting and prioritizing different data types based on their relevance, reliability, and significance, ensuring that

Facilitate collaboration among AI agents and human experts:

ALFINI enables smooth collaboration between AI agents and human experts, such as financial professionals, sustainability specialists, and auditors. The system provides interfaces and protocols for experts to input their knowledge, guidelines, and feedback into the alignment process. AI agents can incorporate this human expertise into their decision-making and output generation. Moreover, ALFINI includes mechanisms for AI agents to explain their alignment decisions and recommendations to human experts, promoting transparency and trust in collaboration.

Capabilities:

Automated alignment of financial and non-financial information:

ALFINI enables AI agents to automatically align financial and non-financial information from disparate sources, ensuring consistency and comparability across reporting periods and entities. The system leverages advanced NLP techniques, such as named entity recognition, coreference resolution, and semantic similarity, to identify and link related concepts across different reports. Additionally, ALFINI employs ontology alignment techniques to map and harmonize the terminology and taxonomies used in various reporting frameworks and standards.

Continuous learning and adaptation to evolving reporting requirements:

ALFINI enables AI agents to continuously learn and adapt to reporting requirements, frameworks, and best practices changes. The system includes mechanisms for monitoring and incorporating updates to relevant standards, such as ISO 5116-3:2021, and adjusting the alignment process accordingly. Moreover, ALFINI employs machine learning techniques, such as reinforcement learning and transfer learning, to allow AI agents to improve their alignment performance over time based on feedback and experience.

Explainable and auditable alignment decisions:

ALFINI ensures that the alignment decisions made by AI agents are explainable and auditable. The system includes techniques for generating human-readable explanations of the reasoning behind specific alignments, including the data sources considered, the rules and guidelines applied, and the confidence levels of the outputs. Additionally, ALFINI maintains detailed logs of all alignment activities, including data inputs, intermediate results, and final outputs, enabling thorough auditing and verification of the alignment process.

Data Harmonization Agent (DHA):

Methods

- 10 The DHA employs advanced data validation, transformation, and harmonization techniques to ensure financial and non-financial data quality, consistency, and integrity across systems and processes.

Data Validation:

- 15 The DHA performs comprehensive data validation to assess accuracy, completeness, and timeliness. It employs rule-based validation, constraint checking, and cross-field validations to identify and resolve data inconsistencies, missing values, or anomalies. The validation process follows a multi-step approach:

Data profiling: The DHA analyzes the data's structure, content, and statistical properties to identify potential issues, such as outliers, data type mismatches, or business rules violations.

- 20 Rule-based validation: The data is subject to a set of predefined validation rules, which cover aspects like data formats, value ranges, and logical constraints.

Cross-field validation: The DHA checks for consistency across related data fields, ensuring that interdependent values align with business rules and data integrity constraints.

- 25 Exception handling: Identified data issues are flagged, and appropriate remediation actions, such as data cleansing, imputation, or manual intervention, are taken.

The time complexity of the data validation process is $O(n)$, where n is the number of data records, as each record needs to be processed and validated against the defined rules and constraints.

Data Transformation:

- 30 The DHA harmonizes data by applying transformation techniques to ensure consistent data representations across systems and processes. It employs data mapping, standardization,

and normalization methods to align data formats, units, and semantics. The transformation process involves the following steps:

Data mapping: The DHA maps data elements from source systems to a standardized data model, resolving structural and semantic differences.

- 5 Data standardization: Consistent data formats, units, and coding schemes are applied to ensure uniformity across systems and processes.

Data normalization: The DHA normalizes data values to a standard scale or range, enabling meaningful comparisons and analysis across different data sources.

- 10 Data enrichment: Additional data attributes or derived values may be added to the harmonized data to enhance its usability and analytical capabilities.

The time complexity of the data transformation process is $O(n \log n)$, where n is the number of data records, as sorting and merging operations may be required during the transformation steps.

- 15 The DHA's data validation and transformation techniques are designed to ensure compliance with Clause 6.2.2 of the standard, which outlines data accuracy, completeness, and timeliness requirements. By applying these methods, the DHA contributes to the quality and consistency of financial and non-financial data, enabling reliable decision-making and regulatory compliance.

Business Layer

- 20 Processes:

Data Ingestion and Preprocessing: This process involves collecting and preprocessing data from various sources, ensuring data quality and consistency through validation and transformation techniques.

- 25 Data Harmonization and Integration: This process focuses on harmonizing and integrating data from multiple sources, resolving structural and semantic differences to create a consistent and unified data representation.

Data Quality Monitoring and Reporting: This process involves continuously monitoring the quality and consistency of harmonized data, identifying, and addressing issues, and generating reports for stakeholders.

- 30 Functions:

Enhanced Data Quality: By validating and transforming data, the DHA ensures the accuracy, completeness, and timeliness of financial and non-financial data, enabling reliable decision-making and regulatory compliance.

- 5 Consistent Data Representation: The DHA harmonizes data from multiple sources, aligning formats, units, and semantics to create a consistent and unified data representation across systems and processes.

Data Integrity and Traceability: The DHA maintains data integrity by resolving inconsistencies, addressing missing values, and providing traceability through data lineage and audit trails.

- 10 Improved Data Governance: The DHA supports data governance initiatives by enforcing data quality standards, promoting consistency, and enabling effective data management practices.

Application Layer

The DHA comprises the following key components:

- 15 DataValidator: This component performs data validation tasks, including rule-based validation, constraint checking, and cross-field validations.

DataTransformer: This component handles data transformation tasks, such as mapping, standardization, normalization, and enrichment, to ensure consistent data representations.

- 20 DataIntegrator: This component integrates and harmonizes data from multiple sources, resolving structural and semantic differences to create a unified data model.

DataQualityMonitor: This component continuously monitors the quality and consistency of harmonized data, identifying and reporting issues or deviations from established standards.

Services:

- 25 DataValidationService: Provides methods for validating data against predefined rules, constraints, and business requirements.

DataTransformationService: Offers services for transforming data, including data mapping, standardization, normalization, and enrichment.

DataIntegrationService: This service facilitates integrating and harmonizing data from multiple sources, creating a consistent and unified data representation.

- 30 DataQualityMonitoringService: Enables continuous monitoring and reporting of data quality and consistency, ensuring adherence to established standards.

Interfaces:

DataValidatorInterface: Defines the methods and parameters for performing data validation tasks.

5 DataTransformerInterface: Specifies the methods and input/output formats for data transformation operations.

DataIntegratorInterface: Describes the methods and data models for integrating and harmonizing data from multiple sources.

DataQualityMonitorInterface: Provides methods for monitoring data quality, setting thresholds, and generating reports.

10 Implementation

Data Ingestion and Preprocessing: This process involves collecting and preprocessing data from various sources, ensuring data quality and consistency through validation and transformation techniques.

15 Data Harmonization and Integration: This process focuses on harmonizing and integrating data from multiple sources, resolving structural and semantic differences to create a consistent and unified data representation.

Data Quality Monitoring and Reporting: This process involves continuously monitoring the quality and consistency of harmonized data, identifying, and addressing issues, and generating reports for stakeholders.

20 Reporting Standards Knowledge Base Agent (RSKBA):

Methods

The RSKBA maintains a comprehensive knowledge base of internationally recognized reporting frameworks and standards, enabling the alignment of financial and non-financial reporting practices by Clause 5 of the ISO 5116-3:2021 standard.

25 Knowledge Base Management:

The RSKBA employs advanced knowledge management techniques to store, update, and retrieve relevant reporting standards and guidance. The knowledge base uses ontologies and semantic web technologies, facilitating efficient querying and reasoning. The knowledge base management process involves the following steps:

30 Knowledge acquisition: The RSKBA ingests and processes various reporting standards, frameworks, and guidelines from authoritative sources, such as the International Financial

Reporting Standards (IFRS), the Global Reporting Initiative (GRI), and the Task Force on Climate-related Financial Disclosures (TCFD).

Knowledge representation: The acquired knowledge is represented using ontologies and semantic web languages like the Web Ontology Language (OWL) and the Resource Description Framework (RDF). This enables explicitly representing concepts, relationships, and rules within the reporting domain.

Knowledge enrichment: The RSKBA applies natural language processing (NLP) techniques and machine learning algorithms to enrich the knowledge base with additional contextual information, such as synonyms, related concepts, and inferred relationships.

Knowledge updating: The RSKBA continuously monitors authoritative sources for updates or changes to reporting standards and frameworks and automatically updates the knowledge base accordingly.

The time complexity of the knowledge base management process depends on the size of the knowledge base and the complexity of the ingested standards and frameworks. The size of the ontologies and the associated metadata determines the space complexity.

Reporting Guidance Retrieval:

The RSKBA provides a query interface that enables stakeholders to access and interpret relevant reporting guidance, aligning with Clause 5.3 of the standard. The interface supports natural language queries and employs semantic search techniques to retrieve the most pertinent information from the knowledge base. The retrieval process involves the following steps:

Query processing: The RSKBA preprocesses and parses the natural language query, identifying key concepts, entities, and relationships using NLP techniques.

Semantic matching: The RSKBA maps the extracted query components to the ontological concepts and relationships within the knowledge base, enabling semantic matching and reasoning.

Result ranking: The RSKBA ranks the retrieved results based on their relevance to the query, considering concept similarity, relationship strength, and contextual information.

Result presentation: The RSKBA presents the relevant reporting guidance to the user, providing explanations, examples, and interpretations to facilitate understanding and application.

The time complexity of the reporting guidance retrieval process depends on the knowledge base's size and the query's complexity. The space complexity is determined by the size of the ontologies and any intermediate data structures used during query processing and result ranking. By maintaining a comprehensive knowledge base of reporting standards and providing an intuitive query interface, the RSKBA supports aligning financial and non-financial reporting practices with internationally recognized frameworks, as outlined in Clause 5 of the ISO 5116-3:2021 standard.

Business Layer

Processes:

- 10 Reporting Standards Acquisition: This process involves identifying and acquiring relevant reporting standards, frameworks, and guidelines from authoritative sources, ensuring the comprehensiveness and currency of the knowledge base.

- 15 Knowledge Base Management: This process focuses on structuring, enriching, and maintaining the knowledge base, ensuring efficient storage, retrieval, and updating reporting standards and guidance.

Reporting Guidance Dissemination: This process involves providing stakeholders with easy access to relevant reporting guidance, enabling the alignment of reporting practices with recognized standards and frameworks.

Functions:

- 20 Reporting Alignment: The RSKBA supports aligning financial and non-financial reporting practices with internationally recognized standards and frameworks, promoting organization consistency and comparability.

- 25 Regulatory Compliance: By maintaining a comprehensive knowledge base of reporting standards and providing guidance, the RSKBA facilitates compliance with regulatory requirements and industry best practices.

Decision Support: The RSKBA's query interface enables stakeholders to access relevant reporting guidance, supporting informed decision-making and the effective implementation of reporting practices.

- 30 Knowledge Sharing: The RSKBA promotes knowledge sharing and dissemination within the organization, fostering a culture of continuous learning and improvement in reporting practices.

Application Layer

KnowledgeBaseManager: This component is responsible for acquiring, structuring, enriching, and maintaining the knowledge base of reporting standards and frameworks.

QueryProcessor: This component handles natural language queries from stakeholders, employing NLP techniques and semantic matching to retrieve relevant reporting guidance from the knowledge base.

ResultRanker: This component ranks and prioritizes the retrieved reporting guidance based on relevance, considering concept similarity, relationship strength, and contextual information.

GuidancePresenter: This component presents the relevant reporting guidance to stakeholders in a transparent and interpretable manner, providing explanations, examples, and interpretations to facilitate understanding and application.

Services:

KnowledgeBaseManagementService: Provides methods for acquiring, updating, and maintaining the knowledge base of reporting standards and frameworks.

ReportingGuidanceQueryService: Offers an interface for stakeholders to submit natural language queries and retrieve relevant reporting guidance.

ResultRankingService: Enables the ranking and prioritization of retrieved reporting guidance based on defined relevance criteria.

GuidancePresentationService: Facilitates the transparent and interpretable presentation of reporting guidance, including explanations, examples, and interpretations.

Interfaces:

KnowledgeBaseManagerInterface: Defines the methods and parameters for managing the knowledge base, including knowledge acquisition, enrichment, and updating.

QueryProcessorInterface: Specifies the input and output formats for natural language queries and the methods for query processing and semantic matching.

ResultRankerInterface: Describes the methods and criteria for ranking and prioritizing retrieved reporting guidance based on relevance.

GuidancePresenterInterface: Provides methods for presenting reporting guidance in various formats, including text, visualizations, and interactive explanations.

Key Performance Indicator Alignment Agent (KPIAA):

Methods

The KPIAA ensures the alignment of financial and non-financial key performance indicators (KPIs) by Clause 7 of the ISO 5116-3:2021 standard, promoting consistency and comparability across reporting periods and entities.

5 KPI Identification and Alignment:

The KPIAA employs natural language processing (NLP) and machine learning techniques to identify and align relevant KPIs from various data sources, including financial statements, operational reports, and industry-specific benchmarks. The KPI alignment process involves the following steps:

- 10 Data ingestion: The KPIAA ingests structured and unstructured data sources containing KPI-related information, such as financial reports, management commentaries, and industry guidelines.

KPI extraction: The KPIAA identifies and extracts potential KPIs from the ingested data sources using NLP techniques like named entity recognition and semantic parsing.

- 15 KPI normalization: The extracted KPIs are normalized to a standardized representation, resolving inconsistencies in naming conventions, units of measurement, and calculation methodologies.

- KPI alignment: The KPIAA aligns the normalized KPIs with industry-specific taxonomies and reporting frameworks, ensuring consistency and comparability across entities and reporting periods.

KPI validation: The aligned KPIs are validated against predefined rules, constraints, and business logic to ensure accuracy and relevance.

- The time complexity of the KPI alignment process depends on the size and complexity of the ingested data sources and the number of KPIs to be processed. The size of the intermediate data structures and the KPI alignment models determine the space complexity.

KPI Discrepancy Detection and Resolution:

The KPIAA employs machine learning techniques to detect and resolve discrepancies in KPI definitions, calculations, and reporting practices, as encouraged by Clause 7.3 of the standard. The discrepancy detection and resolution process involve the following steps:

- 30 Baseline establishment: The KPIAA establishes a baseline of expected KPI values and trends based on historical data, industry benchmarks, and domain-specific rules.

Discrepancy detection: The KPIAA uses anomaly detection algorithms and predictive modeling techniques to identify deviations or discrepancies between the reported KPIs and the expected baseline.

5 Root cause analysis: The KPIAA analyzes the identified discrepancies to determine their root causes, which may include calculation errors, data quality issues, or changes in reporting practices.

Discrepancy resolution: Based on the root cause analysis, the KPIAA recommends corrective actions or adjustments to resolve the discrepancies, such as recalculating KPIs, updating formulas, or aligning reporting practices.

10 Continuous monitoring: The KPIAA monitors KPI reporting practices and updates the baseline and models as new data becomes available, ensuring ongoing alignment and consistency.

The time complexity of the discrepancy detection and resolution process depends on the number of KPIs, the size of the historical data, and the complexity of the anomaly detection
15 and predictive modeling algorithms. The size of the KPI data and the discrepancy detection models determine the space complexity.

By aligning KPIs and detecting and resolving discrepancies, the KPIAA promotes consistency, comparability, and accuracy in KPI reporting, aligning with the principles outlined in Clause 7 of the ISO 5116-3:2021 standard.

20 Business Layer

Processes:

KPI Data Ingestion: This process involves collecting and ingesting structured and unstructured data sources containing KPI-related information from various internal and external sources.

25 KPI Alignment: This process focuses on identifying, extracting, normalizing, and aligning KPIs with industry-specific taxonomies and reporting frameworks, ensuring consistency and comparability across reporting periods and entities.

KPI Discrepancy Management: This process involves detecting and resolving discrepancies in KPI definitions, calculations, and reporting practices, using machine learning techniques
30 and root cause analysis.

Functions:

Consistent KPI Reporting: The KPIAA ensures consistent and comparable KPI reporting across the organization, enabling stakeholders to make informed decisions based on reliable and aligned performance metrics.

5 Regulatory Compliance: The KPIAA facilitates compliance with regulatory requirements and industry best practices by aligning KPIs with industry-specific reporting frameworks and standards.

Performance Monitoring: The KPIAA's aligned and validated KPIs enable effective performance monitoring and benchmarking, supporting data-driven decision-making and continuous improvement initiatives.

10 Automated Discrepancy Detection: The KPIAA's machine learning capabilities enable automated detection of KPI discrepancies, reducing manual efforts and increasing the efficiency of the reporting process.

Application Layer

The KPIAA comprises the following key components:

15 KPIExtractor: This component is responsible for ingesting data sources and employing NLP techniques to identify and extract potential KPIs.

KPINormalizer: This component normalizes the extracted KPIs to a standardized representation, resolving inconsistencies in naming conventions, units of measurement, and calculation methodologies.

20 KPIAligner: This component aligns the normalized KPIs with industry-specific taxonomies and reporting frameworks, ensuring consistency and comparability across entities and reporting periods.

25 DiscrepancyDetector: This component employs machine learning techniques to detect discrepancies in KPI definitions, calculations, and reporting practices, using anomaly detection algorithms and predictive modeling.

DiscrepancyResolver: This component analyzes the identified discrepancies, determines their root causes, and recommends corrective actions or adjustments to resolve them.

Services:

30 KPIExtractionService: Provides methods for ingesting data sources and extracting potential KPIs using NLP techniques.

KPINormalizationService: Offers services for normalizing extracted KPIs to a standardized representation, resolving inconsistencies in naming conventions, units, and calculation methods.

5 KPIAlignmentService: Enables the alignment of normalized KPIs with industry-specific taxonomies and reporting frameworks, ensuring consistency and comparability.

DiscrepancyDetectionService: This service facilitates the detection of KPI discrepancies using machine learning techniques, anomaly detection algorithms, and predictive modeling.

DiscrepancyResolutionService: Provides services for analyzing identified discrepancies, determining root causes, and recommending corrective actions or adjustments.

10 Interfaces:

KPIExtractorInterface: Defines the methods and parameters for ingesting data sources and extracting potential KPIs using NLP techniques.

15 KPINormalizerInterface: This interface specifies the input and output formats for KPI normalization and the methods for resolving inconsistencies in naming conventions, units, and calculation methods.

KPIAlignerInterface: Describes the methods and criteria for aligning normalized KPIs with industry-specific taxonomies and reporting frameworks.

DiscrepancyDetectorInterface: Provides methods for detecting KPI discrepancies using machine learning techniques, anomaly detection algorithms, and predictive modeling.

20 DiscrepancyResolverInterface: This interface defines the methods and output formats for analyzing identified discrepancies, determining root causes, and recommending corrective actions or adjustments.

Narrative Information Alignment Agent (NIAA):

Methods

25 The NIAA ensures the alignment of narrative information in financial and non-financial reports by Clause 8 of the ISO 5116-3:2021 standard, promoting consistency, coherence, and decision-usefulness.

Narrative Information Extraction and Harmonization:

30 The NIAA employs advanced natural language processing (NLP) techniques to extract and harmonize key themes, risks, and opportunities from narrative information across various reports and sources. The narrative alignment process involves the following steps:

Data ingestion: The NIAA ingests unstructured narrative information from management commentaries, sustainability reports, and regulatory filings.

Text preprocessing: The ingested text data is preprocessed using tokenization, stemming, and stop-word removal to prepare it for further analysis.

- 5 Topic modeling: The NIAA applies topic modeling algorithms, such as Latent Dirichlet Allocation (LDA) or Non-Negative Matrix Factorization (NMF), to identify and extract critical themes, risks, and opportunities from the narrative information.

- 10 Sentiment analysis: The NIAA performs sentiment analysis on the extracted themes, risks, and opportunities to determine their polarity (positive, negative, or neutral) and potential impact.

Cross-report harmonization: The extracted and analyzed themes, risks, and opportunities are harmonized across different reports and sources, resolving inconsistencies and aligning their presentation and interpretation.

- 15 The time complexity of the narrative alignment process depends on the volume of narrative information and the complexity of the NLP and topic modeling algorithms employed. The space complexity is determined by the size of the text data and the intermediate data structures required for processing.

Narrative Clarity and Conciseness:

- 20 The NIAA follows principles of balance, clarity, and conciseness in narrative alignment, as Clause 8.3 of the standard requires. It employs text summarization, language generation, and readability analysis to enhance the understandability and decision-usefulness of the aligned narrative information. The narrative clarity and conciseness process involves the following steps:

- 25 Key information extraction: The NIAA identifies and extracts the most relevant and significant information from the harmonized narrative themes, risks, and opportunities.

Text summarization: The extracted critical information is summarized using abstractive or extractive text summarization algorithms, providing concise yet comprehensive representations of the narrative content.

- 30 Language generation: The NIAA generates clear and coherent narrative statements using natural language generation techniques, ensuring a consistent tone, style, and level of detail across reports.

Readability assessment: The generated narrative statements are assessed for readability using metrics such as the Flesch-Kincaid Grade Level or the Gunning Fog Index, ensuring that the intended audience quickly understands the narrative information.

Refinement and iteration: Based on the readability assessment, the NIAA iteratively refines the generated narrative statements to improve clarity and conciseness while maintaining the accuracy and completeness of the information.

The time complexity of the narrative clarity and conciseness process depends on the volume of narrative information and the complexity of the text summarization, language generation, and readability assessment algorithms employed. The space complexity is determined by the size of the narrative data and the intermediate data structures required for processing.

Business Layer

Processes:

Narrative Information Ingestion: This process involves collecting and ingesting unstructured narrative information from various sources, such as management commentaries, sustainability reports, and regulatory filings.

Narrative Extraction and Harmonization: This process focuses on extracting key themes, risks, and opportunities from the narrative information and harmonizing their presentation and interpretation across different reports and sources.

Narrative Clarity and Conciseness: This process involves enhancing the clarity and conciseness of the aligned narrative information using text summarization, language generation, and readability assessment techniques.

Functions:

Consistent Narrative Reporting: The NIAA ensures consistent and coherent narrative reporting across the organization, enabling stakeholders to gain a comprehensive understanding of the organization's performance, risks, and opportunities.

Regulatory Compliance: The NIAA facilitates compliance with regulatory requirements and industry best practices by aligning narrative information with industry-specific reporting frameworks and standards.

Enhanced Decision-Usefulness: The NIAA's ability to extract and harmonize key themes, risks, and opportunities from narrative information while ensuring clarity and conciseness enhances the decision-usefulness of reported information for stakeholders.

Automated Narrative Processing: The NIAA's NLP and text processing capabilities enable automated extraction, harmonization, and enhancement of narrative information, reducing manual efforts and increasing the efficiency of the reporting process.

Application Layer

- 5 The NIAA comprises the following key components:

TextPreprocessor: This component is responsible for ingesting and preprocessing unstructured narrative information, performing tasks such as tokenization, stemming, and stop-word removal.

- 10 ThemeExtractor: This component employs topic modeling algorithms to identify and extract critical themes, risks, and opportunities from the preprocessed narrative information.

SentimentAnalyzer: This component performs sentiment analysis on the extracted themes, risks, and opportunities to determine their polarity and potential impact.

- 15 NarrativeHarmonizer: This component harmonizes the extracted and analyzed themes, risks, and opportunities across different reports and sources, resolving inconsistencies and aligning their presentation and interpretation.

NarrativeEnhancer: This component enhances the clarity and conciseness of the aligned narrative information using text summarization, language generation, and readability assessment techniques.

Services:

- 20 TextPreprocessingService: Provides methods for ingesting and preprocessing unstructured narrative information, performing tasks such as tokenization, stemming, and stop-word removal.

- 25 ThemeExtractionService: This service offers services for identifying and extracting key themes, risks, and opportunities from preprocessed narrative information using topic modeling algorithms.

SentimentAnalysisService: Enables the analysis of sentiment polarity and potential impact for the extracted themes, risks, and opportunities.

- 30 NarrativeHarmonizationService: Facilitates harmonizing extracted and analyzed themes, risks, and opportunities across different reports and sources, resolving inconsistencies and aligning their presentation and interpretation.

NarrativeEnhancementService: Provides services to enhance the clarity and conciseness of aligned narrative information using text summarization, language generation, and readability assessment techniques.

Interfaces:

- 5 TextPreprocessorInterface: This interface defines the methods and parameters for ingesting and preprocessing unstructured narrative information, including tokenization, stemming, and stop-word removal.

- 10 ThemeExtractorInterface: Using topic modeling algorithms, this interface specifies the input and output formats for extracting key themes, risks, and opportunities from preprocessed narrative information.

SentimentAnalyzerInterface: Describes the methods and output formats for analyzing the sentiment polarity and potential impact of extracted themes, risks, and opportunities.

- 15 NarrativeHarmonizerInterface: Provides methods and parameters for harmonizing extracted and analyzed themes, risks, and opportunities across different reports and sources, resolving inconsistencies, and aligning their presentation and interpretation.

NarrativeEnhancerInterface: Defines the methods and output formats for enhancing the clarity and conciseness of aligned narrative information using text summarization, language generation, and readability assessment techniques.

Materiality and Boundary Alignment Agent (MBAA):

- 20 Methods

The MBAA ensures the alignment of materiality assessments and reporting boundaries across financial and non-financial reports in accordance with Clause 9 of the ISO 5116-3:2021 standard, enhancing the relevance and comparability of reported information.

Materiality Assessment and Prioritization:

- 25 The MBAA applies machine learning algorithms to identify and prioritize material topics based on stakeholder concerns and business impacts, as outlined in Clause 9.2. The materiality assessment and prioritization process involve the following steps:

Data ingestion: The MBAA ingests data from various sources, including stakeholder surveys, social media sentiment analysis, industry reports, and internal risk assessments.

Topic modeling: The MBAA employs topic modeling algorithms, such as Latent Dirichlet Allocation (LDA) or Non-Negative Matrix Factorization (NMF), to identify potential material topics from the ingested data.

5 Feature engineering: The MBAA extracts relevant features from the data, such as stakeholder sentiment scores, risk impact assessments, and industry trends, to be used as input for the machine learning models.

Materiality scoring: The MBAA applies supervised or unsupervised machine learning algorithms, such as logistic regression, decision trees, or clustering algorithms, to score and prioritize the identified topics based on their materiality.

10 Topic validation: The prioritized material topics are validated and refined through expert review, stakeholder consultations, and iterative model training.

The time complexity of the materiality assessment and prioritization process depends on the volume of input data, the complexity of the topic modeling and machine learning algorithms employed, and the number of iterations required for model training and validation. The space
15 complexity is determined by the size of the input data and the intermediate data structures needed for processing.

Boundary Alignment:

The MBAA ensures consistent application of materiality and boundary principles across financial and non-financial reports, as required by Clause 9.3. The boundary alignment
20 process involves the following steps:

Report analysis: The MBAA analyzes existing financial and non-financial reports to identify the current reporting boundaries and their underlying assumptions.

25 Materiality mapping: The prioritized material topics are mapped to the identified reporting boundaries, identifying potential gaps, overlaps, or inconsistencies in applying materiality principles.

Boundary adjustment: Based on the materiality mapping, the MBAA recommends adjustments to the reporting boundaries to ensure consistent coverage of material topics across financial and non-financial reports.

30 Stakeholder consultation: The proposed boundary adjustments are reviewed and validated through stakeholder consultations, ensuring alignment with their expectations and concerns.

Boundary finalization: The finalized reporting boundaries are documented and communicated to relevant stakeholders, providing a consistent and transparent basis for reporting on material topics.

The time complexity of the boundary alignment process depends on the number of reports analyzed, the complexity of the materiality mapping, and the extent of stakeholder consultations required. The space complexity is determined by the size of the report data and the intermediate data structures needed for processing.

By applying machine learning algorithms for materiality assessment and prioritization, and ensuring consistent application of materiality and boundary principles, the MBAA promotes the relevance and comparability of reported information across financial and non-financial reports, aligning with the principles outlined in Clause 9 of the ISO 5116-3:2021 standard.

Business Layer

Processes:

Data Ingestion and Preprocessing: This process involves collecting and preprocessing data from various sources, such as stakeholder surveys, social media, industry reports, and internal risk assessments, to be used as input for the materiality assessment and prioritization.

Materiality Assessment and Prioritization: This process identifies and prioritizes material topics using machine learning algorithms, considering stakeholder concerns and business impacts.

Boundary Alignment: This process ensures consistent application of materiality and boundary principles across financial and non-financial reports, adjusting reporting boundaries to provide comprehensive and comparable coverage of material topics.

Functions:

Stakeholder-focused Reporting: The MBAA enables the identification and prioritization of material topics based on stakeholder concerns, ensuring that reported information addresses the most relevant and impactful issues for stakeholders.

Risk-aligned Reporting: The MBAA promotes aligning reported information with the organization's risk management strategies and priorities by considering business impacts and internal risk assessments.

Consistent Reporting: The MBAA ensures consistent application of materiality and boundary principles across financial and non-financial reports, enhancing the comparability and decision-usefulness of reported information for stakeholders.

5 Automated Materiality Assessment: The MBAA's machine learning capabilities enable automated identification and prioritization of material topics, reducing manual efforts and increasing the efficiency of the materiality assessment process.

Application Layer

The MBAA comprises the following key components:

10 DataIngestor: This component is responsible for ingesting and preprocessing data from various sources, such as stakeholder surveys, social media, industry reports, and internal risk assessments.

TopicModeler: This component employs modeling algorithms to identify potential material topics from the ingested data.

15 FeatureExtractor: This component extracts relevant features from the data, such as stakeholder sentiment scores, risk impact assessments, and industry trends, to be used as input for the machine learning models.

MaterialityScorer: This component applies supervised or unsupervised machine learning algorithms to score and prioritize the identified topics based on their materiality.

20 BoundaryAligner: This component maps the prioritized material topics to existing reporting boundaries, identifies gaps or inconsistencies, and recommends boundary adjustments to ensure consistent coverage of material topics across reports.

The MBAA exposes the following application services and interfaces:

Services:

25 DataIngestionService: Provides methods for ingesting and preprocessing data from various sources for input for the materiality assessment and prioritization process.

TopicModelingService: Offers services for identifying potential material topics from the ingested data using topic modeling algorithms.

30 FeatureExtractionService: This service enables the extraction of relevant features from the data, such as stakeholder sentiment scores, risk impact assessments, and industry trends, to be used as input for the machine learning models.

MaterialityScoringService: Using supervised or unsupervised machine learning algorithms, this service facilitates the scoring and prioritization of identified topics based on their materiality.

BoundaryAlignmentService: This service maps prioritized material topics to existing reporting boundaries, identifies gaps or inconsistencies, and recommends boundary adjustments to ensure consistent coverage across reports.

Interfaces:

DataIngestionInterface: This interface defines the methods and parameters for ingesting and preprocessing data from various sources, such as stakeholder surveys, social media, industry reports, and internal risk assessments.

TopicModelerInterface: Specifies the input and output formats for identifying potential material topics from the ingested data using topic modeling algorithms.

FeatureExtractorInterface: Describes the methods and output formats for extracting relevant features from the data for input for the machine learning models.

MaterialityScorerInterface provides methods and parameters for scoring and prioritizing identified topics based on their materiality using supervised or unsupervised machine learning algorithms.

BoundaryAlignerInterface: This interface defines the methods and output formats for mapping prioritized material topics to existing reporting boundaries, identifying gaps or inconsistencies, and recommending boundary adjustments to ensure consistent coverage of material topics across reports.

Assurance Alignment Agent (AAA):

Methods

The AAA coordinates with internal and external assurance providers to ensure consistent and reliable assurance opinions for financial and non-financial reporting by Clause 10 of the ISO 5116-3:2021 standard, enhancing the credibility and trust in reported information.

Assurance Provider Coordination:

The AAA facilitates coordination and alignment among internal and external assurance providers, ensuring consistent application of assurance standards and methodologies across financial and non-financial reporting domains. The assurance provider coordination process involves the following steps:

Assurance provider identification: The AAA identifies and maintains a registry of internal and external assurance providers, including their areas of expertise, methodologies, and accreditations.

5 Assurance scope alignment: The AAA collaborates with assurance providers to align the scope and objectives of assurance engagements, ensuring comprehensive coverage of material topics and reporting boundaries.

Methodology harmonization: The AAA works with assurance providers to harmonize assurance methodologies, sampling techniques, and evidence-gathering procedures, promoting consistency in assurance practices.

10 Assurance resource allocation: The AAA optimizes the allocation of assurance resources based on risk assessments, materiality considerations, and stakeholder expectations, ensuring efficient and effective assurance processes.

15 Assurance timeline synchronization: The AAA coordinates and synchronizes assurance timelines across different reporting domains, enabling timely and integrated assurance opinions.

The time complexity of the assurance provider coordination process depends on the number of assurance providers involved, the complexity of the assurance engagements, and the extent of coordination required. The space complexity is determined by the size of the assurance provider registry and the associated metadata.

20 Assurance Findings Communication:

The AAA communicates assurance findings and recommendations to the Reporting Integration Agent (RIA) for transparent disclosure, as required by Clause 10.3. The assurance findings communication process involves the following steps:

25 Assurance report consolidation: The AAA consolidates assurance reports and opinions from various assurance providers, ensuring consistency in reporting formats and taxonomies.

Findings analysis: The AAA analyzes the assurance findings, identifying areas of concern, opportunities for improvement, and potential discrepancies across reporting domains.

30 Recommendation formulation: Based on the findings analysis, the AAA formulates recommendations for enhancing the reliability, transparency, and credibility of reported information.

Communication with the RIA: The AAA communicates the consolidated assurance findings and recommendations to the RIA, ensuring transparent disclosure and integration into the final reporting outputs.

Stakeholder engagement: The AAA facilitates stakeholder engagement and communication regarding the assurance findings, addresses concerns, and provides additional context or clarifications.

The time complexity of the assurance findings communication process depends on the volume of assurance reports, the complexity of the findings analysis, and the extent of stakeholder engagement required. The size of the assurance reports and the associated metadata determines the space complexity. By coordinating with assurance providers and communicating assurance findings and recommendations, the AAA enhances the credibility and trust in reported information, aligning with the principles outlined in Clause 10 of the ISO 5116-3:2021 standard.

Business Layer:

Processes:

Assurance Provider Management: This process involves identifying, registering, and coordinating with internal and external assurance providers, ensuring alignment in their assurance objectives, methodologies, and timelines.

Assurance Engagement Planning: This process focuses on defining the scope and objectives of assurance engagements, allocating assurance resources based on risk assessments and materiality considerations, and synchronizing assurance timelines across reporting domains.

Assurance Findings Consolidation: This process involves consolidating assurance reports and opinions from various assurance providers, analyzing the findings, formulating recommendations, and communicating them to the Reporting Integration Agent (RIA) for transparent disclosure.

Functions:

Consistent Assurance Practices: The AAA ensures consistent application of assurance standards and methodologies across financial and non-financial reporting domains, promoting comparability and reliability of assurance opinions.

Efficient Assurance Resource Allocation: By optimizing the allocation of assurance resources based on risk assessments and materiality considerations, the AAA contributes to the efficiency and cost-effectiveness of assurance processes.

5 Transparent Assurance Disclosure: The AAA facilitates transparent disclosure of assurance findings and recommendations, enhancing stakeholder trust and confidence in the reported information.

Collaborative Assurance: The AAA promotes collaboration and coordination among internal and external assurance providers, fostering a culture of integrated and aligned assurance practices within the organization.

10 Application Layer

The AAA comprises the following key components:

AssuranceProviderRegistry: This component maintains a registry of internal and external assurance providers, including their areas of expertise, methodologies, and accreditations.

15 AssuranceEngagementPlanner: This component defines the scope and objectives of assurance engagements, allocates assurance resources based on risk assessments and materiality considerations, and synchronizes assurance timelines across reporting domains.

AssuranceReportConsolidator: This component consolidates assurance reports and opinions from various assurance providers, ensuring consistency in reporting formats and taxonomies.

20 FindingsAnalyzer: This component analyzes the consolidated assurance findings, identifying areas of concern, opportunities for improvement, and potential discrepancies across reporting domains.

RecommendationGenerator: This component formulates recommendations based on the findings analysis to enhance reported information's reliability, transparency, and credibility.

25 Services:

AssuranceProviderRegistryService: Provides methods for managing the registry of internal and external assurance providers, including registration, update, and retrieval functionalities.

30 AssuranceEngagementPlanningService: Offers services for defining the scope and objectives of assurance engagements, allocating assurance resources based on risk assessments and materiality considerations, and synchronizing assurance timelines across reporting domains.

AssuranceReportConsolidationService: Enables the consolidation of assurance reports and opinions from various assurance providers, ensuring consistency in reporting formats and taxonomies.

FindingsAnalysisService: Facilitates the analysis of consolidated assurance findings, identifying areas of concern, opportunities for improvement, and potential discrepancies across reporting domains.

RecommendationGenerationService: Provides services for formulating recommendations for enhancing the reliability, transparency, and credibility of reported information based on the findings analysis.

10 Interfaces:

AssuranceProviderRegistryInterface: This interface defines the methods and parameters for managing the registry of internal and external assurance providers, including registration, update, and retrieval functionalities.

AssuranceEngagementPlannerInterface: This interface specifies the input and output formats for defining the scope and objectives of assurance engagements, allocating assurance resources based on risk assessments and materiality considerations, and synchronizing assurance timelines across reporting domains.

AssuranceReportConsolidatorInterface: Describes the methods and input/output formats for consolidating assurance reports and opinions from various assurance providers, ensuring consistency in reporting formats and taxonomies.

FindingsAnalyzerInterface: Provides methods and parameters for analyzing the consolidated assurance findings, identifying areas of concern, opportunities for improvement, and potential discrepancies across reporting domains.

RecommendationGeneratorInterface: Defines the methods and output formats for formulating recommendations for enhancing the reliability, transparency, and credibility of reported information based on the findings analysis.

Reporting Integration Agent (RIA):

Methods

The RIA integrates aligned financial and non-financial information into comprehensive reports, following the principles of connectivity, consistency, and accessibility outlined in Clause 11.2 of the ISO 5116-3:2021 standard.

Information Integration and Connectivity:

The RIA employs advanced data integration and visualization techniques to combine aligned information from various sources, ensuring connectivity and traceability across financial and non-financial domains. The information integration and connectivity process involves the following steps:

- 5 Data ingestion: The RIA ingests aligned data from the Data Harmonization Agent (DHA), Key Performance Indicator Alignment Agent (KPIAA), Narrative Information Alignment Agent (NIAA), and Materiality and Boundary Alignment Agent (MBAA).

Data mapping and transformation: The RIA maps the aligned data to a standard data model, ensuring consistent representation and enabling cross-domain connectivity and traceability.

- 10 Data linking and relationship modeling: The RIA establishes relationships and links between related data elements across financial and non-financial domains, facilitating integrated reporting and analysis.

Data visualization and storytelling: The RIA employs data visualization techniques and narrative storytelling approaches to present integrated information coherently and

- 15 meaningfully, highlighting connections and interdependencies across domains.

Report generation: The RIA generates comprehensive reports that seamlessly integrate financial and non-financial information, enabling stakeholders to gain a holistic understanding of the organization's performance, risks, and opportunities.

- 20 The time complexity of the information integration and connectivity process depends on the volume and complexity of the aligned data and the complexity of the data mapping, transformation, and visualization techniques employed. The space complexity is determined by the size of the aligned data and the intermediate data structures required for processing.

Report Formatting and Accessibility:

- 25 The RIA generates integrated reports in various formats, as specified in Clause 11.3, catering to the diverse needs and preferences of stakeholders. The report formatting and accessibility process involves the following steps:

Stakeholder requirements analysis: The RIA analyzes the specific reporting requirements and preferences of different stakeholder groups, including preferred formats, accessibility needs, and distribution channels.

- 30 Report template design: The RIA designs report templates that comply with industry standards and best practices, ensuring consistent and accessible presentation of integrated information.

Content adaptation: The RIA adapts the integrated information to the specific report formats, optimizing content layout, typography, and visual elements for each format.

Accessibility enhancements: The RIA incorporates accessibility features, such as alternative text descriptions, screen reader compatibility, and color contrast adjustments, to ensure equal access to information for all stakeholders.

Report distribution: The RIA distributes the generated reports through various channels, such as print, digital platforms, or interactive web-based interfaces, catering to stakeholder preferences and ensuring wide dissemination of integrated information.

The time complexity of the report formatting and accessibility process depends on the number of report formats required, the complexity of the content adaptation, and the extent of accessibility enhancements needed. The space complexity is determined by the size of the report templates and the generated reports. By integrating aligned information and generating comprehensive reports in various accessible formats, the RIA promotes connectivity, consistency, and accessibility in the presentation and communication of financial and non-financial information, aligning with the principles outlined in Clause 11 of the ISO 5116-3:2021 standard.

Business Layer:

Processes:

Aligned Information Ingestion: This process involves ingesting aligned financial and non-financial information from various sources, including the DHA, KPIAA, NIAA, and MBAA.

Information Integration and Connectivity: This process focuses on integrating the aligned information, establishing cross-domain connections and traceability, and generating comprehensive reports seamlessly combining financial and non-financial information.

Report Formatting and Distribution: This process involves formatting the integrated reports in various accessible formats, catering to diverse stakeholder needs and preferences, and ensuring wide dissemination through appropriate distribution channels.

Functions:

Holistic Reporting: The RIA enables the generation of comprehensive reports that integrate financial and non-financial information, providing stakeholders with a holistic understanding of the organization's performance, risks, and opportunities.

Cross-domain Connectivity: The RIA facilitates integrated analysis and decision-making by establishing connections and traceability across financial and non-financial domains, promoting a more comprehensive understanding of the organization's overall performance.

Stakeholder-centric Reporting: The RIA caters to diverse stakeholder needs and preferences by generating reports in various accessible formats, ensuring equal access to information, and promoting greater transparency and accountability.

Efficient Report Generation: The RIA streamlines the report generation process by automating data integration, formatting, and distribution tasks, reducing manual efforts, and increasing the efficiency of the reporting process.

10 Application Layer:

The RIA comprises the following key components:

DataIntegrator: This component ingests aligned data from various sources and performs data mapping, transformation, and linking to establish cross-domain connections and traceability.

- 15 ReportGenerator: This component generates comprehensive reports by combining integrated financial and non-financial information, employing data visualization and narrative storytelling techniques.

FormatAdapter: This component adapts the integrated information to specific report formats, optimizing content layout, typography, and visual elements for each format.

- 20 AccessibilityEnhancer: This component incorporates accessibility features, such as alternative text descriptions, screen reader compatibility, and color contrast adjustments, to ensure equal access to information for all stakeholders.

ReportDistributor: This component manages the distribution of generated reports through various channels, such as print, digital platforms, or interactive web-based interfaces, catering to stakeholder preferences.

- 25

Services:

DataIntegrationService: Provides methods for ingesting aligned data from various sources, performing data mapping, transformation, and linking to establish cross-domain connections and traceability.

- 30 ReportGenerationService: Offers services for generating comprehensive reports by combining integrated financial and non-financial information, employing data visualization and narrative storytelling techniques.

FormatAdaptationService: This service enables the adaptation of integrated information to specific report formats, optimizing content layout, typography, and visual elements for each format.

5 AccessibilityEnhancementService: Facilitates incorporating accessibility features, such as alternative text descriptions, screen reader compatibility, and color contrast adjustments, to ensure equal access to information for all stakeholders.

ReportDistributionService: Provides services for managing the distribution of generated reports through various channels, such as print, digital platforms, or interactive web-based interfaces, catering to stakeholder preferences.

10 Interfaces:

DataIntegratorInterface: This interface defines the methods and parameters for ingesting aligned data from various sources, performing data mapping, transformation, and linking to establish cross-domain connections and traceability.

15 ReportGeneratorInterface: Specifies the input and output formats for generating comprehensive reports by combining integrated financial and non-financial information, employing data visualization and narrative storytelling techniques.

FormatAdapterInterface: Describes the methods and input/output formats for adapting integrated information to specific report formats, optimizing content layout, typography, and visual elements for each format.

20 AccessibilityEnhancerInterface: Provides methods and parameters for incorporating accessibility features, such as alternative text descriptions, screen reader compatibility, and color contrast adjustments, to ensure equal access to information for all stakeholders.

ReportDistributorInterface: This interface defines the methods and parameters for managing the distribution of generated reports through various channels, such as print, digital
25 platforms, or interactive web-based interfaces, catering to stakeholder preferences.

ALFINI Petri-net Agent

Methods

The ALFINI Petri-net Agent orchestrates the coordination and execution of the various ALFINI sub-agents, ensuring a seamless and efficient alignment of financial and non-
30 financial reporting processes by the principles outlined in Clause 4 of the ISO 5116-3:2021 standard.

Petri-net Modeling and Execution:

The ALFINI Petri-net Agent employs Petri-net modeling techniques to define the interactions, dependencies, and execution sequences of the ALFINI sub-agents. The Petri-net modeling and execution process involves the following steps:

Agent Interaction Modeling: The ALFINI Petri-net Agent models the interactions and

- 5 dependencies between the ALFINI sub-agents, such as the DHA, RSKBA, KPIAA, NIAA, MBAA, AAA, and RIA, using Petri-net constructs like places, transitions, and arcs.

Execution Sequence Definition: The ALFINI Petri-net Agent defines the execution sequences and control flows of the ALFINI sub-agents, ensuring that each agent is triggered at the appropriate time and with the necessary inputs and outputs, by the principles outlined

10 in Clause 4 of the ISO 5116-3:2021 standard.

Petri-net Validation: The ALFINI Petri-net Agent validates the constructed Petri-net model to ensure its correctness, completeness, and adherence to relevant Petri-net standards, such as ISO/IEC 15909-1:2004 (Petri-net markup language) and ISO/IEC 15909-2:2011 (Transfer format for Petri-nets).

- 15 Petri-net Execution and Monitoring: The ALFINI Petri-net Agent executes the Petri-net model, monitoring the progress of the ALFINI sub-agents and ensuring that the alignment and integration tasks are carried out in the appropriate sequence and with the necessary coordination.

- Dynamic Adaptation: The ALFINI Petri-net Agent supports dynamic adaptation of the Petri-net model, allowing for adjustments and modifications to the execution sequences and agent interactions based on feedback, monitoring data, or changes in the reporting requirements.
- 20

The time complexity of the Petri-net modeling and execution process depends on the complexity of the Petri-net model, the number of ALFINI sub-agents involved, and the volume of data and interactions between the sub-agents. The space complexity is

- 25 determined by the size of the Petri-net model and the associated data structures required for execution and monitoring.

Petri-net Visualization and Analysis:

The ALFINI Petri-net Agent provides visualization and analysis capabilities to monitor, troubleshoot, and optimize the ALFINI sub-agent coordination and execution. The Petri-net visualization and analysis process involves the following steps:

30

Petri-net Rendering: The ALFINI Petri-net Agent renders visual representations of the Petri-net model, including the places, transitions, arcs, and token markings, adhering to Petri-net

visualization standards such as ISO/IEC 15909-4:2017 (Graphical representation of Petri-nets).

Execution Tracing: The ALFINI Petri-net Agent traces the execution of the Petri-net model, highlighting the active transitions, token movements, and sub-agent interactions. This

5 enables stakeholders to monitor the progress of the alignment and integration tasks.

Performance Analysis: The ALFINI Petri-net Agent analyzes the performance of the Petri-net execution, collecting metrics such as execution times, resource utilization, and bottlenecks and providing insights for optimization and improvement.

Deadlock and Liveness Analysis: The ALFINI Petri-net Agent performs deadlock and

10 liveness analysis on the Petri-net model, identifying potential deadlocks, livelocks, or other execution issues that may impede the completion of the alignment and integration tasks.

Reporting and Visualization: The ALFINI Petri-net Agent generates reports and visualizations summarizing the execution status, performance metrics, and analysis results, enabling stakeholders to monitor and optimize the ALFINI sub-agent coordination and

15 execution effectively.

The time complexity of the Petri-net visualization and analysis process depends on the size of the Petri-net model, the complexity of the rendering and analysis algorithms employed, and the volume of execution data to be processed. The size of the Petri-net model determines the space complexity, the associated execution data, and the intermediate data

20 structures required for rendering and analysis. By employing Petri-net modeling and execution techniques, the ALFINI Petri-net Agent ensures the coordinated and efficient execution of the ALFINI sub-agents, aligning with the principles outlined in Clause 4 of the ISO 5116-3:2021 standard and adhering to relevant Petri-net ISO standards.

Business Layer:

25 Processes:

ALFINI Sub-Agent Coordination: This process involves modeling the interactions and dependencies between the ALFINI sub-agents, defining their execution sequences, and orchestrating their coordinated execution to align financial and non-financial reporting processes.

30 Execution Monitoring and Analysis: This process focuses on monitoring the execution of the Petri-net model, tracing the progress of the ALFINI sub-agents, and analyzing performance metrics, deadlocks, and liveness to identify optimization opportunities and potential issues.

Reporting and Visualization: This process involves generating reports and visualizations summarizing the execution status, performance metrics, and analysis results, enabling stakeholders to monitor and optimize the ALFINI sub-agent coordination and execution effectively.

5 Functions:

Efficient Alignment and Integration: The ALFINI Petri-net Agent ensures the efficient and coordinated execution of the ALFINI sub-agents, enabling the effective alignment and integration of financial and non-financial reporting processes.

10 Compliance with ISO Standards: By adhering to relevant Petri-net ISO standards, the ALFINI Petri-net Agent promotes compliance with industry best practices and ensures interoperability with other Petri-net-based systems and tools.

Transparency and Monitoring: The ALFINI Petri-net Agent provides transparency into the execution of the ALFINI sub-agents, enabling stakeholders to monitor progress, identify bottlenecks, and optimize the alignment and integration processes.

15 Adaptability and Extensibility: The ALFINI Petri-net Agent supports dynamic adaptation of the Petri-net model, allowing for adjustments and modifications to accommodate changes in reporting requirements or the introduction of new ALFINI sub-agents.

Application Layer:

The ALFINI Petri-net Agent comprises the following key components:

20 PetriNetModeler: This component is responsible for modeling the interactions and dependencies between the ALFINI sub-agents using Petri-net constructs, adhering to relevant Petri-net ISO standards.

PetriNetExecutor: This component executes the Petri-net model, orchestrating the coordinated execution of the ALFINI sub-agents and monitoring their progress and
25 interactions.

PetriNetAnalyzer: This component performs various analyses on the Petri-net model and its execution, including performance analysis, deadlock and liveness analysis, and identification of optimization opportunities.

30 PetriNetVisualizer: This component renders visual representations of the Petri-net model and its execution, adhering to Petri-net visualization standards and enabling stakeholders to monitor the progress of the alignment and integration tasks.

ReportingEngine: This component generates reports and visualizations summarizing the execution status, performance metrics, and analysis results, enabling stakeholders to monitor and optimize the ALFINI sub-agent coordination and execution effectively.

Services:

- 5 PetriNetModelingService: Provides methods for modeling the interactions and dependencies between the ALFINI sub-agents using Petri-net constructs, adhering to relevant Petri-net ISO standards.

- 10 PetriNetExecutionService: Offers services for executing the Petri-net model, orchestrating the coordinated execution of the ALFINI sub-agents, and monitoring their progress and interactions.

PetriNetAnalysisService: Enables the performance of various analyses on the Petri-net model and its execution, including performance analysis, deadlock and liveness analysis, and identification of optimization opportunities.

- 15 PetriNetVisualizationService: Facilitates the rendering of visual representations of the Petri-net model and its execution, adhering to Petri-net visualization standards and enabling stakeholders to monitor the progress of the alignment and integration tasks.

ReportingService: Provides services for generating reports and visualizations summarizing the execution status, performance metrics, and analysis results, enabling stakeholders to monitor and optimize the ALFINI sub-agent coordination and execution effectively.

- 20 Interfaces:

PetriNetModelerInterface: This interface defines the methods and parameters for modeling the interactions and dependencies between the ALFINI sub-agents using Petri-net constructs, adhering to relevant Petri-net ISO standards.

- 25 PetriNetExecutorInterface: Specifies the methods and parameters for executing the Petri-net model, orchestrating the coordinated execution of the ALFINI sub-agents, and monitoring their progress and interactions.

PetriNetAnalyzerInterface: Describes the methods and parameters for performing various analyses on the Petri-net model and its execution, including performance analysis, deadlock and liveness analysis, and identification of optimization opportunities.

- 30 PetriNetVisualizerInterface: Provides methods and parameters for rendering visual representations of the Petri-net model and its execution, adhering to Petri-net visualization

standards, and enabling stakeholders to monitor the progress of the alignment and integration tasks.

Reporting interface: This interface defines the methods and parameters for generating reports and visualizations summarizing the execution status, performance metrics, and analysis results. It enables stakeholders to monitor and optimize the ALFINI sub-agent coordination and execution effectively.

MEGAN

MEGAN (Metadata Extraction, Generation, and Alignment Network) is an advanced, agent-based module of GenFoundary that automates the generation of ISO 11179 and UN/CEFACT compliant metadata registries, abstracts DataVault2.0 schemas using the CoLLEGe framework, and enables cross-taxonomy mapping and translation from diverse data requirements. At its core, MEGAN employs a synergistic ensemble of specialized agents collaborating through continuous optimization cycles, overseen by a Petri-net orchestration model.

Key MEGAN advantages:

Automated Metadata Generation: Leveraging generative AI models and prompt engineering techniques to automatically generate rich, context-aware, and standards-compliant metadata from diverse data requirements.

DataVault2.0 Schema Abstraction: Employing the CoLLEGe framework to abstract key business concepts and relationships from ISO 11179 metadata, enabling the automated generation of DataVault2.0 schemas.

Cross-Taxonomy Mapping and Translation: This involves using AI-driven techniques to automatically map and translate metadata elements across different reporting frameworks, jurisdictions, and data standards.

Continuous Learning and Adaptation: Incorporating user feedback, new data sources, and evolving requirements to improve the accuracy and relevance of generated metadata and DataVault2.0 schemas over time.

Agents

The MEGAN agent architecture comprises several sub-agents, each with specific roles and responsibilities, working together to ensure the automated generation and alignment of metadata registries and DataVault2.0 schemas:

Data Requirements Ingestion Agent (DRIA): This agent ingests data requirements from various sources and formats, preprocessing and normalizing the data to ensure conformance with ISO 11179 and ISO 20022 principles.

5 Metadata Extraction and Normalization Agent (MENA): This agent employs advanced NLP techniques to extract relevant metadata elements from the ingested data and normalizes them into a consistent format compliant with ISO 11179 and ISO 20022.

DataVault2.0 Schema Abstraction Agent (DSAA): This agent applies the CoLLEGe framework to abstract key business concepts and relationships from the generated metadata, producing DataVault2.0 schemas aligned with ISO 11179 and ISO 20022.

10 Cross-Taxonomy Mapping and Alignment Agent (CTMAA): Utilizes AI and NLP techniques to automatically map and translate metadata elements across different reporting frameworks, jurisdictions, and data standards.

Vector Database Agent (VDA): Stores and indexes the generated metadata, DataVault2.0 schemas, and cross-taxonomy mappings in a high-performance vector database, enabling
15 efficient similarity-based search and knowledge discovery.

Data Lineage and Provenance Tracking Agent (DLPTA): Captures and maintains a complete audit trail of the metadata management, schema generation, and mapping processes, ensuring end-to-end traceability and compliance.

Ontology-based Reasoning Agent (OBRA): This agent leverages domain knowledge
20 represented in ontologies to infer relationships and align concepts across different taxonomies, enabling more accurate and context-aware metadata mapping and integration.

The sub-agents are orchestrated using a Petri-net model, ensuring a coordinated and efficient execution of the metadata generation, schema abstraction, and cross-taxonomy mapping tasks. The Petri-net models the dependencies and interactions between the sub-
25 agents, guaranteeing that each task is performed in the appropriate sequence and with the necessary inputs and outputs.

Evaluation & Validation

The selection of ground truth data and evaluation metrics is carefully tailored to align with the specific functionalities and objectives of each MEGAN component:

30 Data Requirements Ingestion Agent (DRIA): DRIA's performance is evaluated using ground truth data from established regulatory schemas, data dictionaries, and expert-validated data requirements. Metrics such as ingestion accuracy, normalization consistency, and ISO

11179/ISO 20022 conformance are employed to assess DRIA's effectiveness in preprocessing and normalizing diverse data requirements.

Metadata Extraction and Normalization Agent (MENA): MENA's evaluation relies on ground truth data from manually annotated metadata, ISO 11179/ISO 20022 metadata registries, and expert-verified extractions. Metrics like extraction accuracy, normalization consistency, and ISO compliance are used to measure MENA's ability to extract and normalize relevant metadata elements.

DataVault2.0 Schema Abstraction Agent (DSAA): DSAA's performance is assessed using ground truth data from manually designed DataVault2.0 schemas, CoLLEGe framework guidelines, and expert-validated abstractions. Metrics such as schema correctness, business concept alignment, and ISO compatibility are employed to evaluate DSAA's effectiveness in abstracting DataVault2.0 schemas from ISO 11179 metadata.

Cross-Taxonomy Mapping and Alignment Agent (CTMAA): CTMAA is evaluated using ground truth data from manually mapped metadata elements, cross-jurisdiction reporting standards, and expert-verified alignments. Metrics like mapping accuracy, translation consistency, and interoperability are used to assess CTMAA's ability to automatically map and translate metadata elements across different taxonomies and frameworks.

Vector Database Agent (VDA): VDA's performance is evaluated using ground truth data from established metadata repositories, schema catalogs, and expert-curated knowledge bases. Metrics like indexing efficiency, search relevance, and knowledge discovery accuracy are employed to measure VDA's effectiveness in storing, indexing, and retrieving metadata, schemas, and mappings.

Ontology-based Reasoning Agent (OBRA): OBRA's performance is evaluated using ground truth data from established domain ontologies, expert-curated concept alignments, and manually verified inferences. Metrics such as reasoning accuracy, concept alignment precision, and inference completeness are employed to assess OBRA's effectiveness in leveraging ontologies to infer relationships and align concepts across taxonomies.

Data Lineage and Provenance Tracking Agent (DLPTA): DLPTA's evaluation relies on ground truth data from manually maintained audit trails, regulatory compliance requirements, and expert-verified lineage records. Metrics such as lineage completeness, provenance accuracy, and compliance adherence assess DLPTA's ability to capture and maintain a comprehensive and reliable audit trail of metadata management processes.

FIG. 3 is a functional block diagram 300 illustrating an example of a metadata management system within a multi-agent artificial intelligence framework, according to an embodiment

herein. The metadata management system corresponds to an embodiment of the MEGAN agent architecture.

DRIA (Data Requirements Ingestion Agent) 304:

DRIA is an ingestion module which ingests and preprocesses diverse data requirements into
5 ISO 11179/20022 format. DRIA may interface directly with external data sources to
automatically retrieve the data inputs, ensuring that metadata extraction begins from a
standardized baseline. DRIA may serve MENA 306 by providing preprocessed ISO-
compliant data, ensuring that metadata extraction starts from a standardized baseline. DRIA
associates with external data sources, receiving raw data inputs that are crucial for the initial
10 data processing steps.

MENA (Metadata Extraction and Normalization Agent) 306:

MENA is a metadata extraction and normalization agent that employs natural language
processing (NLP) to extract, normalize, and enrich metadata from the outputs provided by
DRIA 304.

15 MENA may serve DSAA 308 by providing extracted and normalized metadata, which is
essential for the subsequent abstraction into DataVault2.0 schemas.

MENA integrates with DRIA's outputs and external knowledge sources, ensuring a
comprehensive approach to metadata handling and enhancement.

DSAA (DataVault2.0 Schema Abstraction Agent) 308:

20 DSAA is an abstraction module which abstracts ISO 11179/20022 metadata from MENA into
DataVault2.0 schemas, creating structured data models suitable for diverse applications.

Serves CTMAA 312 by providing abstracted DataVault schemas, which are used for further
cross-taxonomy mapping.

Maintains a close association with MENA 306, receiving metadata inputs that are critical for
25 schema abstraction.

CTMAA (Cross-Taxonomy Mapping and Alignment Agent) 312:

CTMAA is a mapping module which utilizes advanced AI techniques to map and translate
metadata across different taxonomies and standards.

CTMAA serves VDA 314 by providing cross-taxonomy mappings, which are essential for
30 enhancing the semantic search capabilities of the database.

CTMAA associates with DSAA's (308) schemas and external ontologies, integrating these resources to perform accurate and efficient taxonomy mappings.

VDA (Vector Database Agent) 314:

5 VDA is a storage module which stores, indexes, and enables semantic search over the metadata received from CTMAA 312, facilitating efficient data retrieval and utilization.

VDA maintains an association with CTMAA 312, receiving and managing mapped metadata for enhanced search functionalities.

DLPTA (Data Lineage and Provenance Tracking Agent) 310:

10 Tracks audit trails capturing data lineage and provenance across all processes within MEGAN.

Integrates and associates with other MEGAN agents, ensuring comprehensive tracking and documentation of data flows and transformations, which is critical for audit and compliance purposes.

Summary of relationships shown in FIG. 3:

15 DRIA 304 acts as the entry point for raw data, processing it into a form suitable for MENA 306, which then extracts and normalizes the metadata. MENA 306 passes this refined metadata to DSAA 308, which abstracts it into DataVault2.0 schemas that are further mapped by CTMAA 312. CTMAA's 312 outputs are utilized by VDA 314 for storage and indexing, ensuring that the metadata is readily accessible and searchable. DLPTA 310
20 overlays this process by tracking data lineage and associations across all agents, reinforcing the system's integrity and traceability.

DRIA (Data Requirements Ingestion Agent):

25 The Data Requirements Ingestion Agent (DRIA) is a highly advanced component of MEGAN that leverages the neural capabilities of large language models (LLMs) like Claude-3 to automate the ingestion, preprocessing, normalization, and validation of data requirements from diverse sources and formats. By harnessing the power of LLMs, the DRIA can understand and process complex data structures, such as ISO 11179-compliant metadata registries, data dictionaries, conceptual data models, and ISO 20022 financial message schemas and business process definitions.

30 The DRIA employs LLMs' natural language understanding and generation capabilities to preprocess and normalize the ingested data, ensuring conformance with ISO 11179 and ISO 20022 principles and structures. This involves extracting relevant metadata elements, such

as data element concepts, data elements, value domains, classification schemes, message components, data types, and business rules, as defined by these international standards.

Moreover, the DRIA utilizes the reasoning abilities of LLMs to perform syntactic and semantic validation of the ingested data against ISO 11179 and ISO 20022 specifications.

5 This ensures the quality and consistency of the data requirements, identifying and flagging any discrepancies or non-compliant elements.

Once the data requirements have been ingested, preprocessed, normalized, and validated, the DRIA sends the ISO 11179 and ISO 20022-aligned data to the Metadata Extraction and Normalization Agent for further processing. Using LLMs enables the DRIA to handle the
10 complexity and variability of data requirements from different sources and domains, adapting to new formats and structures as needed.

By leveraging the neural capabilities of LLMs, the DRIA achieves a high level of automation, accuracy, and efficiency in managing data requirements. It reduces manual effort and ensures the quality and interoperability of metadata across the MEGAN system. This
15 ultimately contributes to the system's overall effectiveness in generating ISO 11179 and UN/CEFACT compliant metadata registries, abstracting DataVault2.0 schemas, and enabling cross-taxonomy mapping and translation.

DRIA Methods:

The DRIA employs advanced data ingestion, preprocessing, normalization, and validation
20 techniques to ensure data requirements' quality, consistency, and compliance with ISO 11179 and ISO 20022 standards.

Data Ingestion and Preprocessing:

The DRIA ingests data requirements from various sources and formats, including ISO 11179-compliant metadata registries, data dictionaries, conceptual data models, and ISO
25 20022 financial message schemas and business process definitions. The ingestion process involves the following steps:

Source Identification: The DRIA identifies and connects to the relevant data sources, such as databases, file systems, or APIs, containing the data requirements.

Data Retrieval: The DRIA retrieves the data requirements from the identified sources using
30 appropriate protocols and methods, such as SQL queries, file parsing, or API calls.

Format Detection: The DRIA automatically detects the format of the ingested data, such as XML, JSON, CSV, or proprietary formats, using intelligent format detection algorithms.

Data Cleaning: The DRIA applies data cleaning techniques to remove any irrelevant, inconsistent, or corrupted data from the ingested data requirements, ensuring a clean and reliable dataset for further processing.

The time complexity of the data ingestion and preprocessing process depends on the volume and variety of the data sources and the complexity of the data cleaning and format detection algorithms employed. The size of the ingested data and any intermediate data structures used during preprocessing determine the space complexity.

Data Normalization and Alignment:

The DRIA pre-processes and normalizes the ingested data to ensure conformance with ISO 11179 and ISO 20022 principles and structures. The normalization and alignment process involves the following steps:

Metadata Element Extraction: The DRIA extracts relevant metadata elements, such as data element concepts, data elements, value domains, classification schemes, message components, data types, and business rules, as defined by ISO 11179 and ISO 20022, from the ingested data requirements.

Syntactic Normalization: The DRIA normalizes the extracted metadata elements to conform to the syntactic rules and conventions specified by ISO 11179 and ISO 20022, such as naming conventions, data type representations, and cardinality constraints.

Semantic Alignment: The DRIA aligns the normalized metadata elements with the semantic concepts and relationships defined in ISO 11179 and ISO 20022 metamodels, ensuring consistent and interoperable metadata representations across different data sources and domains.

Cross-Referencing: The DRIA establishes cross-references and mappings between related metadata elements, such as data elements and value domains, or message components and business processes, to facilitate data integration and traceability.

The time complexity of the data normalization and alignment process depends on the number and complexity of the metadata elements and the efficiency of the extraction, normalization, and alignment algorithms used. The space complexity is determined by the size of the extracted metadata and any auxiliary data structures employed during the process.

Data Validation and Quality Assurance:

To ensure data quality and consistency, the DRIA performs syntactic and semantic validation of the ingested data against ISO 11179 and ISO 20022 specifications. The validation process involves the following steps:

5 Syntactic Validation: The DRIA validates the ingested data against the syntactic rules and constraints defined in ISO 11179 and ISO 20022, such as data type compatibility, mandatory field presence, and format compliance.

Semantic Validation: The DRIA validates the ingested data against the semantic rules and relationships specified in ISO 11179 and ISO 20022 metamodels, such as data element concept bindings, value domain consistency, and business rule adherence.

10 Quality Checks: The DRIA applies additional quality checks to the ingested data, such as completeness, uniqueness, and referential integrity, to identify and flag any data quality issues or anomalies.

15 Error Handling and Reporting: The DRIA handles validation and quality issues by generating detailed error reports, logging the problems, and triggering appropriate error resolution workflows or notifications to data stewards and stakeholders.

The time complexity of the data validation and quality assurance process depends on the volume and complexity of the ingested data and the number and intricacy of the validation rules and quality checks applied. The space complexity is determined by the size of the ingested data and any error logs or reports generated during the process.

20 Business Layer:

Processes:

Data Source Identification and Connection: This process involves identifying the relevant data sources containing data requirements and establishing secure connections to retrieve the data.

25 Data Ingestion and Preprocessing: This process focuses on ingesting data requirements from various sources and formats, detecting the data format, and applying data cleaning techniques to ensure a clean and reliable dataset for further processing.

30 Metadata Extraction and Normalization: This process involves extracting relevant metadata elements from the ingested data requirements and normalizing them to conform to ISO 11179 and ISO 20022 syntactic rules and conventions.

Data Alignment and Cross-Referencing: This process involves aligning the normalized metadata elements with ISO 11179 and ISO 20022 semantic concepts and relationships and establishing cross-references and mappings between related metadata elements.

5 Data Validation and Quality Assurance: This process focuses on performing syntactic and semantic validation of the ingested data against ISO 11179 and ISO 20022 specifications, applying quality checks, and handling validation errors and quality issues.

Functions:

Automated Data Ingestion: The DRIA enables automated ingestion of data requirements from diverse sources and formats, reducing manual effort and improving efficiency.

10 Data Format Detection and Cleaning: The DRIA automatically detects the format of the ingested data and applies data cleaning techniques to ensure data reliability and consistency.

ISO 11179 and ISO 20022 Compliance: The DRIA ensures that the ingested data requirements are preprocessed, normalized, and aligned with ISO 11179 and ISO 20022 principles and structures, promoting data interoperability and standardization.

15 Metadata Element Extraction and Mapping: The DRIA extracts relevant metadata elements from the ingested data and establishes cross-references and mappings between related elements, facilitating data integration and traceability.

Data Quality and Validation: The DRIA performs rigorous syntactic and semantic validation of the ingested data, applies quality checks, and handles validation errors and quality issues, ensuring data accuracy and consistency.

Application Layer

The DRIA comprises the following key components:

DataSourceConnector: This component identifies and establishes secure connections to the relevant data sources containing data requirements.

DataIngestor: This component focuses on ingesting data requirements from various sources and formats, detecting the data format, and applying data cleaning techniques to ensure data reliability.

30 MetadataExtractor: This component extracts relevant metadata elements from the ingested data requirements and normalizes them to conform to ISO 11179 and ISO 20022 syntactic rules and conventions.

DataAligner: This component aligns the normalized metadata elements with ISO 11179 and ISO 20022 semantic concepts and relationships and establishes cross-references and mappings between related metadata elements.

5 DataValidator: This component performs syntactic and semantic validation of the ingested data against ISO 11179 and ISO 20022 specifications, applies quality checks, and handles validation errors and quality issues.

Services:

DataSourceConnectionService: Provides methods for identifying and establishing secure connections to the relevant data sources containing data requirements.

10 DataIngestionService: Offers services for ingesting data requirements from various sources and formats, detecting the data format, and applying data cleaning techniques.

MetadataExtractionService: This service enables the extraction of relevant metadata elements from the ingested data requirements and their normalization to conform to ISO 11179 and ISO 20022 syntactic rules and conventions.

15 DataAlignmentService: Facilitates the alignment of normalized metadata elements with ISO 11179 and ISO 20022 semantic concepts and relationships and the establishment of cross-references and mappings between related metadata elements.

DataValidationService: Provides services for performing syntactic and semantic validation of the ingested data against ISO 11179 and ISO 20022 specifications, applying quality checks,
20 and handling validation errors and quality issues.

Interfaces:

DataSourceConnectorInterface: Defines the methods and parameters for identifying and establishing secure connections to the relevant data sources containing data requirements.

DataIngesterInterface: Specifies the methods and input/output formats for ingesting data
25 requirements from various sources and formats, detecting the data format, and applying data cleaning techniques.

MetadataExtractorInterface: Describes the methods and input/output formats for extracting relevant metadata elements from the ingested data requirements and normalizing them to conform to ISO 11179 and ISO 20022 syntactic rules and conventions.

30 DataAlignerInterface: This interface defines the methods and parameters for aligning normalized metadata elements with ISO 11179 and ISO 20022 semantic concepts and

relationships and establishing cross-references and mappings between related metadata elements.

DataValidatorInterface: Specifies the methods and input/output formats for performing syntactic and semantic validation of the ingested data against ISO 11179 and ISO 20022 specifications, applying quality checks, and handling validation errors and quality issues.

Metadata Extraction and Normalization Agent (MENA):

The Metadata Extraction and Normalization Agent (MENA) is a highly advanced component of MEGAN that leverages the neural capabilities of large language models (LLMs) like Claude-3 to automate the extraction, normalization, and semantic enrichment of metadata from the data requirements ingested by the Data Requirements Ingestion Agent (DRIA). By harnessing the power of LLMs, MENA can understand and process complex metadata structures, such as those defined by ISO 11179 and ISO 20022 standards.

MENA employs LLMs' natural language understanding and generation capabilities to extract relevant metadata elements, such as data element concepts, data elements, value domains, classification schemes, message components, data types, and business rules, from the preprocessed and normalized data requirements. It then applies semantic enrichment techniques, such as entity linking, synonym identification, and contextual analysis, to enhance the extracted metadata with additional semantic information and relationships.

Moreover, MENA utilizes LLMs' reasoning abilities to perform semantic validation and consistency checks on the extracted and enriched metadata, ensuring compliance with ISO 11179 and ISO 20022 metamodels and ontologies. This involves identifying and resolving semantic conflicts, ambiguities, and inconsistencies in the metadata, such as duplicate data element concepts, inconsistent value domain definitions, or conflicting business rules.

Once the metadata has been extracted, normalized, and semantically enriched, MENA sends the ISO 11179 and ISO 20022-compliant metadata to the Metadata Registry Generation Agent for further processing and integration into the target metadata registries and data models. Using LLMs enables MENA to handle the complexity and variability of metadata across different domains and use cases, adapting to new semantic requirements and evolving standards as needed.

By leveraging the neural capabilities of LLMs, MENA achieves a high level of automation, accuracy, and semantic richness in metadata extraction and normalization. It reduces manual effort and ensures metadata quality, consistency, and interoperability across the MEGAN system. This ultimately contributes to the system's overall effectiveness in

generating ISO 11179 and UN/CEFACT compliant metadata registries, abstracting DataVault2.0 schemas, and enabling cross-taxonomy mapping and translation.

Methods

- 5 MENA employs advanced metadata extraction, normalization, semantic enrichment, and validation techniques to ensure metadata quality, consistency, and compliance with ISO 11179 and ISO 20022 standards.

Metadata Extraction: MENA extracts relevant metadata elements from the preprocessed and normalized data requirements ingested by the DRIA. The extraction process involves the following steps:

- 10 Metadata Element Identification: MENA identifies the critical metadata elements, such as data element concepts, data elements, value domains, classification schemes, message components, data types, and business rules, as defined by ISO 11179 and ISO 20022, within the ingested data requirements.

- 15 Syntactic Parsing: MENA applies syntactic parsing techniques, such as regular expressions, grammar-based parsing, or machine learning-based named entity recognition, to extract the identified metadata elements from the source data.

Metadata Element Structuring: MENA organizes the extracted metadata elements into structured representations, such as JSON, XML, or object-oriented models, based on the ISO 11179 and ISO 20022 metamodels.

- 20 Metadata Cleaning and Normalization: MENA applies data cleaning and normalization techniques to the extracted metadata elements, such as removing duplicates, standardizing formats, and resolving inconsistencies, to ensure data quality and consistency.

- 25 The time complexity of the metadata extraction process depends on the volume and complexity of the ingested data requirements and the efficiency of the identification, parsing, structuring, and cleaning algorithms employed. The space complexity is determined by the size of the extracted metadata and any intermediate data structures used during the extraction process.

- 30 Semantic Enrichment: MENA applies semantic enrichment techniques to enhance the extracted metadata elements with additional semantic information and relationships. The semantic enrichment process involves the following steps:

Entity Linking: MENA links the extracted metadata elements to relevant concepts, entities, or resources in external knowledge bases, taxonomies, or ontologies, such as industry-specific vocabularies, ISO 11179 or ISO 20022 reference models, or Linked Open Data sources.

Synonym and Variant Identification: MENA identifies synonyms, abbreviations, and variant terms for the extracted metadata elements using LLM-based techniques like word embeddings, semantic similarity, or contextual analysis to enhance semantic interoperability.

Semantic Relationship Extraction: MENA extracts semantic relationships, such as hierarchical, associative, or mappings, between the extracted metadata elements using LLM-based techniques like dependency parsing, co-reference resolution, or knowledge graph embedding.

Contextual Metadata Enrichment: MENA enriches the extracted metadata elements with contextual information, such as data lineage, provenance, usage, or quality metrics, by analyzing the surrounding text, data flows, or system logs using LLM-based techniques like sentiment analysis, topic modeling, or sequence labeling.

The time complexity of the semantic enrichment process depends on the number and complexity of the extracted metadata elements, the size and diversity of the external knowledge sources, and the efficiency of the entity linking, synonym identification, relationship extraction, and contextual enrichment algorithms employed. The size of the enriched metadata determines the space complexity, the external knowledge sources, and any intermediate data structures used during the enrichment process.

Semantic Validation and Consistency Checking: MENA performs semantic validation and consistency checking on the extracted and enriched metadata to ensure compliance with ISO 11179 and ISO 20022 metamodels and ontologies. The validation process involves the following steps:

Semantic Constraint Validation: MENA validates the extracted and enriched metadata against the semantic constraints, such as domain and range restrictions, cardinality constraints, or logical axioms, defined in the ISO 11179 and ISO 20022 metamodels and ontologies using LLM-based reasoning techniques like description logic, rule-based inference, or graph-based constraint checking.

Consistency and Completeness Checking: MENA checks the consistency and completeness of the extracted and enriched metadata by identifying and resolving semantic conflicts, ambiguities, or gaps, such as duplicate or missing metadata elements, inconsistent value domain definitions, or conflicting business rules, using LLM-based techniques like anomaly detection, clustering, or knowledge graph completion.

Semantic Mapping and Alignment: MENA maps and aligns the extracted and enriched metadata to the ISO 11179 and ISO 20022 metamodels and ontologies using LLM-based techniques, such as ontology matching, schema matching, or semantic similarity, to ensure semantic interoperability and compliance.

- 5 Semantic Error Handling and Reporting: MENA handles semantic validation and consistency issues by generating detailed error reports, logging the problems, and triggering appropriate error resolution workflows or notifications to data stewards and stakeholders.

The time complexity of the semantic validation and consistency checking process depends on the volume and complexity of the extracted and enriched metadata, the size and
10 expressiveness of the ISO 11179 and ISO 20022 metamodels and ontologies, and the efficiency of the constraint validation, consistency checking, mapping, and error handling algorithms employed. The space complexity is determined by the size of the validated metadata, the metamodels and ontologies, and any error logs or reports generated during the process.

15 Business Layer

Processes:

Metadata Extraction and Structuring: This process involves identifying and extracting relevant metadata elements from the preprocessed and normalized data requirements and organizing them into structured representations based on ISO 11179 and ISO 20022
20 metamodels.

Semantic Enrichment and Linking: This process focuses on enhancing the extracted metadata elements with additional semantic information, such as synonyms, related concepts, or contextual metadata, and linking them to relevant external knowledge sources and reference models.

- 25 Semantic Validation and Consistency Checking: This process involves validating the extracted and enriched metadata against the semantic constraints and consistency rules defined in ISO 11179 and ISO 20022 metamodels and ontologies and handling any semantic errors or inconsistencies.

Metadata Mapping and Alignment: This process maps and aligns the validated and
30 consistent metadata to the target ISO 11179 and ISO 20022 metamodels and ontologies to ensure semantic interoperability and compliance.

Metadata Delivery and Integration: This process involves delivering the extracted, enriched, validated, and aligned metadata to the Metadata Registry Generation Agent for further processing and integration into the target metadata registries and data models.

Functions:

- 5 Automated Metadata Extraction: MENA enables automated extraction of relevant metadata elements from diverse data requirements, reducing manual effort and improving efficiency.

Semantic Enrichment and Linking: MENA enhances the extracted metadata with additional semantic information and relationships, leveraging external knowledge sources and reference models to improve semantic interoperability and richness.

- 10 ISO 11179 and ISO 20022 Compliance: MENA ensures that the extracted and enriched metadata is validated against and aligned with ISO 11179 and ISO 20022 metamodels and ontologies, promoting standardization and consistency.

- Semantic Validation and Error Handling: MENA performs rigorous semantic validation and consistency checking on the extracted and enriched metadata, identifies and resolves
15 semantic errors and inconsistencies, and generates detailed error reports and notifications.

Metadata Integration and Delivery: MENA seamlessly integrates the extracted, enriched, validated, and aligned metadata with the Metadata Registry Generation Agent and other components of MEGAN, enabling efficient metadata delivery and consumption.

Application Layer

- 20 The MENA comprises the following key components:

MetadataExtractor: This component identifies and extracts relevant metadata elements from the preprocessed and normalized data requirements and organizes them into structured representations based on ISO 11179 and ISO 20022 metamodels.

- SemanticEnricher: This component enhances the extracted metadata elements with
25 additional semantic information, such as synonyms, related concepts, or contextual metadata, and links them to relevant external knowledge sources and reference models.

SemanticValidator: This component validates the extracted and enriched metadata against the semantic constraints and consistency rules defined in ISO 11179 and ISO 20022 metamodels and ontologies and handles any semantic errors or inconsistencies.

- 30 MetadataAligner: This component maps and aligns the validated and consistent metadata to the target ISO 11179 and ISO 20022 metamodels and ontologies to ensure semantic interoperability and compliance.

MetadataDeliverer: This component delivers the extracted, enriched, validated, and aligned metadata to the Metadata Registry Generation Agent for further processing and integration into the target metadata registries and data models.

Services:

- 5 MetadataExtractionService: Provides methods for identifying and extracting relevant metadata elements from the preprocessed and normalized data requirements and organizing them into structured representations.

- 10 SemanticEnrichmentService: Offers services for enhancing the extracted metadata elements with additional semantic information and linking them to relevant external knowledge sources and reference models.

SemanticValidationService: Enables the validation of the extracted and enriched metadata against the semantic constraints and consistency rules defined in ISO 11179 and ISO 20022 metamodels and ontologies and the handling of semantic errors and inconsistencies.

- 15 MetadataAlignmentService: Facilitates the mapping and alignment of the validated and consistent metadata to the target ISO 11179 and ISO 20022 metamodels and ontologies.

MetadataDeliveryService: Provides services for delivering the extracted, enriched, validated, and aligned metadata to the Metadata Registry Generation Agent and other components of MEGAN.

Interfaces:

- 20 MetadataExtractorInterface: This interface defines the methods and parameters for identifying and extracting relevant metadata elements from the preprocessed and normalized data requirements and organizing them into structured representations.

- 25 SemanticEnricherInterface: Specifies the methods and input/output formats for enhancing the extracted metadata elements with additional semantic information and linking them to relevant external knowledge sources and reference models.

SemanticValidatorInterface: Describes the methods and input/output formats for validating the extracted and enriched metadata against the semantic constraints and consistency rules defined in ISO 11179 and ISO 20022 metamodels and ontologies and handling semantic errors and inconsistencies.

- 30 MetadataAlignerInterface: Defines the methods and parameters for mapping and aligning the validated and consistent metadata to the target ISO 11179 and ISO 20022 metamodels and ontologies.

MetadataDelivererInterface: Specifies the methods and input/output formats for delivering the extracted, enriched, validated, and aligned metadata to the Metadata Registry Generation Agent and other components of MEGAN.

In embodiments, the Metadata Extraction and Normalization Agent (MENA), as part of its integration within the larger MEGAN system, may process complex transformations of metadata to meet specific standards like ISO 11179. For example, MENA may be used to convert an existing JSON schema into one that is fully compliant with this standard, ensuring that each metadata element is correctly extracted, defined, and structured. In embodiments, the transformation process may include the following steps:

10 Metadata Extraction and Structuring: This step involves extracting all necessary data from the original JSON schema. For each element identified, the script assigns a unique identifier, maps descriptions to standardized definitions, and categorizes each element according to specified attributes and mappings related to FIBO (Financial Industry Business Ontology) and FIRE standards.

15 Data Elements Formation: Each element from the data is assigned a unique ID and structured into a new array. This includes mapping names, defining elements based on previous descriptions, and aligning data types to the ISO standard.

Classification and Membership Assignment: Elements are categorized into classification schemes to organize the metadata logically. Membership arrays are then populated to associate elements with their respective classifications.

Relationship and Usage Guidelines: Relationships between data elements are defined, and comprehensive guidelines for using, customizing, and integrating the new schema are provided.

25 Documentation: User documentation is generated to assist in understanding, implementing, and utilizing the new schema effectively.

Neuromorphic Data Structure

The data structure presented in JSON schemas incorporates neurocomputing principles and attributes within the schema elements, making it unique and neuromorphic. This design enables artificial neural networks and graph-based machine learning algorithms to process and utilize the schema more efficiently.

The key neuromorphic features include:

Activation Functions: Each node in the schema's graph structure (e.g., concepts, data elements, columns, rows) is assigned an activation function, such as ReLU (Rectified Linear Unit), Sigmoid, or Tanh. These activation functions introduce non-linearity, allowing the nodes to learn and represent complex patterns and relationships in the data.

- 5 Embedding Sizes: The schema elements are assigned embedding sizes, which determine the dimensionality of their vector representations. These embeddings allow the nodes to be represented in a continuous vector space, capturing their semantic and structural properties. The embedding sizes can be adjusted based on the complexity and granularity of the data.

- 10 Weights: The relationships (edges) between the schema elements are assigned weights, indicating the strength or importance of the connections. These weights can be learned and optimized during the training process of machine learning algorithms, enabling the discovery of meaningful patterns and associations within the data.

- 15 These neuromorphic attributes make the schema more compatible with neural network architectures and graph-based learning algorithms. The activation functions, embedding sizes, and weights allow the data to be processed and analyzed to mimic the behavior of biological neural networks, enabling the extraction of complex patterns, relationships, and insights from the data.

- 20 Furthermore, the neuromorphic design facilitates the integration of the schema with other neuro-symbolic systems, such as knowledge graphs, ontologies, and reasoning engines. The semantic annotations and ontological mappings provided in the schema (e.g., using SKOS properties) can be leveraged to establish connections and perform inference across different data sources and domains.

- 25 The neuromorphic features also enhance the schema's adaptability and scalability. As new data is added or the schema evolves, the activation functions, embedding sizes, and weights can be fine-tuned and optimized to accommodate the changes and maintain the schema's effectiveness in representing and processing the data.

DataVault2.0 Schema Abstraction Agent (DSAA):

- 30 The DataVault2.0 Schema Abstraction Agent (DSAA) is a highly advanced component of the MEGAN that leverages the neural capabilities of large language models (LLMs) like Claude-3 to automate the generation of DataVault2.0 schemas from ISO 11179/ISO 20022 compliant metadata. By harnessing the power of LLMs, DSAA can understand and process the complex semantic structures and relationships defined in the metadata and apply domain-specific knowledge to abstract key business concepts, processes, and relationships.

DSAA employs LLMs' natural language understanding and generation capabilities to analyze the ISO 11179/ISO 20022 compliant metadata received from the Generative AI Model Agent (GAMA) and apply the Concept Embedding Generation for Large Language Models (CoLLEGe) framework to abstract the metadata into a format suitable for DataVault2.0 modeling. The CoLLEGe framework enables DSAA to identify and extract key business concepts, processes, and relationships from the metadata while preserving the semantic integrity, traceability, and adherence to ISO 11179 and ISO 20022 standards.

Moreover, DSAA utilizes LLMs' reasoning abilities to generate semantically rich and well-organized DataVault2.0 schemas, including HUBS, LINKS, and SATELLITES, ensuring alignment with ISO 11179 and ISO 20022 principles, the CoLLEGe framework, and data vault modeling best practices. The LLM's ability to understand and generate complex data structures allows DSAA to create schemas that accurately represent the underlying business domain and are tailored to financial data management and analysis requirements.

DSAA also maintains bidirectional traceability and lineage between the ISO 11179/ISO 20022 metadata elements and the corresponding DataVault2.0 schema components by leveraging LLMs' reasoning and memory capabilities. This enables DSAA to keep track of the relationships between the source metadata and the generated schemas, facilitating impact analysis, change management, and regulatory compliance.

To ensure the quality and consistency of the generated DataVault2.0 schemas, DSAA validates them against ISO 11179 and ISO 20022 conformance rules and industry-specific data quality checks using the LLM's ability to understand and apply complex validation rules. This allows DSAA to identify and flag any issues or inconsistencies in the schemas, ensuring their accuracy and reliability.

Finally, DSAA sends the generated ISO 11179/ISO 20022 aligned DataVault2.0 schemas to the Cross-Taxonomy Mapping and Alignment Agent for further processing and integration into the overall MEGAN. Using LLMs enables DSAA to handle the complexity and variability of schema abstraction tasks, adapting to new standards and domain-specific requirements as needed.

By leveraging the neural capabilities of LLMs, DSAA achieves a high level of automation, accuracy, and efficiency in generating DataVault2.0 schemas from ISO 11179/ISO 20022 compliant metadata. This reduces manual effort, ensures the quality and semantic integrity of the schemas, and facilitates the overall effectiveness of MEGAN in generating ISO 11179 and UN/CEFACT compliant metadata registries, abstracting DataVault2.0 schemas, and enabling cross-taxonomy mapping and translation.

Methods

DSAA employs advanced metadata abstraction, schema generation, traceability management, and validation techniques to ensure the quality, consistency, and compliance of the generated DataVault2.0 schemas with ISO 11179 and ISO 20022 standards.

- 5 Metadata Abstraction using CoLLEGe Framework: DSAA applies the Concept Embedding Generation for Large Language Models (CoLLEGe) framework to abstract the ISO 11179/ISO 20022 compliant metadata into a format suitable for DataVault2.0 modeling. The abstraction process involves the following steps:

10 Semantic Analysis: DSAA leverages the natural language understanding capabilities of LLMs to analyze the semantic structures and relationships defined in the ISO 11179/ISO 20022 metadata, identifying key business concepts, processes, and relationships.

15 Concept Embedding Generation: DSAA generates concept embeddings for the identified business concepts, processes, and relationships using LLM-based techniques like word embeddings, sentence embeddings, or graph embeddings to capture their semantic meaning and context.

Metadata Abstraction: DSAA abstracts the ISO 11179/ISO 20022 metadata into a format suitable for DataVault2.0 modeling by mapping the generated concept embeddings to the corresponding DataVault2.0 schema components, such as HUBS, LINKS, and SATELLITES, while preserving the semantic integrity and traceability.

- 20 Domain-Specific Enrichment: DSAA enriches the abstracted metadata with additional attributes and relationships specific to the financial industry, leveraging the domain knowledge embedded in LLMs to tailor the resulting schemas to financial data management and analysis requirements.

25 The time complexity of the metadata abstraction process depends on the volume and complexity of the ISO 11179/ISO 20022 metadata, the size and diversity of the LLM's knowledge base, and the efficiency of the semantic analysis, concept embedding generation, and abstraction algorithms employed. The space complexity is determined by the size of the abstracted metadata, the concept embeddings, and any intermediate data structures used during the abstraction process.

- 30 DataVault2.0 Schema Generation: DSAA generates semantically rich and well-organized DataVault2.0 schemas from the abstracted metadata, ensuring alignment with ISO 11179 and ISO 20022 principles, the CoLLEGe framework, and data vault modeling best practices. The schema generation process involves the following steps:

HUB Generation: DSAA generates HUBS for the critical business concepts identified during the metadata abstraction process, using the LLM's ability to understand and develop complex data structures to define the HUB attributes, keys, and constraints.

LINK Generation: DSAA generates LINKS to represent the relationships between the HUBS, leveraging the LLM's understanding of the semantic relationships defined in the abstracted metadata to create the LINK attributes, keys, and cardinality constraints.

SATELLITE Generation: DSAA generates SATELLITES to capture the time-varying attributes and historical changes associated with the HUBS and LINKS, using the LLM's ability to understand temporal aspects and generate appropriate SATELLITE structures and attributes.

Schema Optimization: DSAA optimizes the generated DataVault2.0 schemas by applying data vault modeling best practices, such as denormalization, historicization, and conformance to naming conventions. It leverages the LLM's knowledge of these practices to ensure the schemas are efficient, maintainable, and scalable.

The time complexity of the schema generation process depends on the size and complexity of the abstracted metadata, the number and diversity of the business concepts, relationships, and attributes, and the efficiency of the HUB, LINK, SATELLITE generation, and optimization algorithms employed. The space complexity is determined by the size of the generated DataVault2.0 schemas and any intermediate data structures used during the generation process.

Traceability and Lineage Management: DSAA maintains bidirectional traceability and lineage between the ISO 11179/ISO 20022 metadata elements and the corresponding DataVault2.0 schema components using LLMs' reasoning and memory capabilities. The traceability management process involves the following steps:

Metadata-Schema Mapping: DSAA establishes mappings between the ISO 11179/ISO 20022 metadata elements and the corresponding DataVault2.0 schema components during the schema generation process, leveraging the LLM's ability to keep track of these relationships.

Lineage Capture: DSAA captures the lineage information, such as the source metadata elements, transformation rules, and target schema components, for each mapping using LLM-based techniques like knowledge graphs, provenance modeling, or version control.

Impact Analysis: DSAA leverages the LLM's reasoning capabilities to analyze the impact of changes in the ISO 11179/ISO 20022 metadata or the DataVault2.0 schemas on the

mappings and lineage, identifying the affected components and propagating the changes accordingly.

Traceability Reporting: DSAA generates traceability reports and visualizations, such as lineage diagrams, impact analysis matrices, or audit trails, using the LLM's natural language generation capabilities to communicate the traceability and lineage information to users and stakeholders.

The time complexity of the traceability management process depends on the number and complexity of the metadata schema mappings, the frequency and scope of the changes, and the efficiency of the lineage capture, impact analysis, and reporting algorithms employed.

The space complexity is determined by the traceability and lineage metadata size, the knowledge graphs or version control systems, and any intermediate data structures used during the management process.

DataVault2.0 Schema Validation: DSAA validates the generated DataVault2.0 schemas against ISO 11179 and ISO 20022 conformance rules and industry-specific data quality checks to ensure their accuracy, consistency, and reliability. The validation process involves the following steps:

Conformance Rule Validation: DSAA validates the generated DataVault2.0 schemas against the conformance rules and constraints defined in the ISO 11179 and ISO 20022 standards, leveraging the LLM's ability to understand and apply complex validation rules to identify any violations or inconsistencies.

Data Quality Checking: DSAA applies industry-specific data quality checks, such as data type consistency, referential integrity, or business rule compliance, to the generated DataVault2.0 schemas, using the LLM's domain knowledge and reasoning capabilities to identify any data quality issues or anomalies.

Schema Consistency Verification: DSAA verifies the internal consistency and coherence of the generated DataVault2.0 schemas, checking for issues like naming conflicts, circular references, or structural inconsistencies. It leverages the LLM's ability to analyze and reason about complex data structures.

Validation Reporting and Resolution: DSAA generates validation reports and notifications, highlighting any conformance, data quality, or consistency issues in the generated DataVault2.0 schemas and suggests possible resolutions or remediation steps using the LLM's natural language generation and problem-solving capabilities.

The time complexity of the schema validation process depends on the size and complexity of the generated DataVault2.0 schemas, the number and diversity of the conformance rules, data quality checks, consistency constraints, and the efficiency of the validation algorithms employed. The size of the validation metadata determines the space complexity, the rule bases or knowledge graphs used, and any intermediate data structures used during the validation process.

Business Layer

Processes:

ISO 11179/ISO 20022 Metadata Acquisition: This process involves acquiring the ISO 11179/ISO 20022 compliant metadata from the Generative AI Model Agent (GAMA) and preparing it for the metadata abstraction process.

Metadata Abstraction and Concept Embedding Generation: This process focuses on applying the CoLLEGe framework to abstract the ISO 11179/ISO 20022 metadata into a format suitable for DataVault2.0 modeling, generating concept embeddings for crucial business concepts, processes, and relationships.

DataVault2.0 Schema Generation and Optimization: This process involves generating semantically rich and well-organized DataVault2.0 schemas, including HUBS, LINKS, and SATELLITES, from the abstracted metadata and optimizing them based on data vault modeling best practices.

Traceability and Lineage Management: This process maintains bidirectional traceability and lineage between the ISO 11179/ISO 20022 metadata elements and the corresponding DataVault2.0 schema components, capturing lineage information and performing impact analysis.

DataVault2.0 Schema Validation and Issue Resolution: This process involves validating the generated DataVault2.0 schemas against ISO 11179 and ISO 20022 conformance rules and industry-specific data quality checks, identifying issues or inconsistencies, and suggesting resolutions or remediation steps.

Functions:

Automated Metadata Abstraction: DSAA enables automated abstraction of ISO 11179/ISO 20022 metadata into a format suitable for DataVault2.0 modeling, reducing manual effort and improving efficiency.

Semantic Enrichment and Domain-Specific Tailoring: DSAA enriches the abstracted metadata with additional attributes and relationships specific to the financial industry, leveraging LLMs' domain knowledge to tailor the resulting schemas to financial data management and analysis requirements.

- 5 Intelligent Schema Generation: DSAA generates semantically rich and well-organized DataVault2.0 schemas, ensuring alignment with ISO 11179 and ISO 20022 principles, the CoLLEGe framework, and data vault modeling best practices.

- 10 Comprehensive Traceability and Lineage: DSAA maintains bidirectional traceability and lineage between the metadata elements and schema components, facilitating impact analysis, change management, and regulatory compliance.

Rigorous Validation and Quality Assurance: DSAA validates the generated schemas against conformance rules and data quality checks, identifies issues, and suggests resolutions, ensuring accuracy, consistency, and reliability.

Application Layer

- 15 The DSAA comprises the following key components:

MetadataAcquisitionManager: This component acquires the ISO 11179/ISO 20022 compliant metadata from the Generative AI Model Agent (GAMA) and prepares it for the metadata abstraction process.

- 20 MetadataAbstractionEngine: This component applies the CoLLEGe framework to abstract the ISO 11179/ISO 20022 metadata into a format suitable for DataVault2.0 modeling, generating concept embeddings for crucial business concepts, processes, and relationships.

SchemaGenerationOptimizer: This component generates semantically rich and well-organized DataVault2.0 schemas, including HUBS, LINKS, and SATELLITES, from the abstracted metadata and optimizes them based on data vault modeling best practices.

- 25 TraceabilityLineageManager: This component maintains bidirectional traceability and lineage between the ISO 11179/ISO 20022 metadata elements and the corresponding DataVault2.0 schema components, capturing lineage information and performing impact analysis.

- 30 SchemaValidationResolverAgent: This component validates the generated DataVault2.0 schemas against ISO 11179 and ISO 20022 conformance rules and industry-specific data quality checks, identifies issues or inconsistencies, and suggests resolutions or remediation steps.

Services:

MetadataAcquisitionService: Provides methods for acquiring ISO 11179/ISO 20022 compliant metadata from the Generative AI Model Agent (GAMA) and preparing it for the metadata abstraction process.

5 MetadataAbstractionService: Offers services for applying the CoLLEGe framework to abstract ISO 11179/ISO 20022 metadata into a format suitable for DataVault2.0 modeling, generating concept embeddings for crucial business concepts, processes, and relationships.

SchemaGenerationService: This service enables the generation of semantically rich and well-organized DataVault2.0 schemas, including HUBS, LINKS, and SATELLITES, from the abstracted metadata and optimizes them based on data vault modeling best practices.

10 TraceabilityLineageService: Facilitates the maintenance of bidirectional traceability and lineage between ISO 11179/ISO 20022 metadata elements and corresponding DataVault2.0 schema components, capturing lineage information and performing impact analysis.

SchemaValidationService: Provides services for validating generated DataVault2.0 schemas against ISO 11179 and ISO 20022 conformance rules and industry-specific data quality
15 checks, identifying issues or inconsistencies, and suggesting resolutions or remediation steps.

Interfaces:

MetadataAcquisitionInterface: Defines the methods and parameters for acquiring ISO
11179/ISO 20022 compliant metadata from the Generative AI Model Agent (GAMA) and
20 preparing it for the metadata abstraction process.

MetadataAbstractionInterface: Specifies the methods and input/output formats for applying the CoLLEGe framework to abstract ISO 11179/ISO 20022 metadata into a format suitable for DataVault2.0 modeling, generating concept embeddings for crucial business concepts, processes, and relationships.

25 SchemaGenerationInterface: Describes the methods and input/output formats for generating semantically rich and well-organized DataVault2.0 schemas, including HUBS, LINKS, and SATELLITES, from the abstracted metadata and optimizing them based on data vault modeling best practices.

TraceabilityLineageInterface: Defines the methods and parameters for maintaining
30 bidirectional traceability and lineage between ISO 11179/ISO 20022 metadata elements and corresponding DataVault2.0 schema components, capturing lineage information and performing impact analysis.

SchemaValidationInterface: Specifies the methods and input/output formats for validating generated DataVault2.0 schemas against ISO 11179 and ISO 20022 conformance rules and industry-specific data quality checks, identifying issues or inconsistencies, and suggesting resolutions or remediation steps.

5 In embodiments, the DataVault2.0 Schema Abstraction Agent (DSAA) may leverage the capabilities of LLMs to automate the generation and management of DataVault2.0 schemas from metadata that complies with ISO 11179 and ISO 20022 standards. For example, this operation begins with DSAA employing the neural capabilities of LLMs, like Claude-3, to perform a deep semantic analysis of the metadata. This analysis focuses on understanding
10 the complex semantic structures and relationships embedded within the metadata. Utilizing the Concept Embedding Generation for Large Language Models (CoLLEGe) framework, DSAA identifies key business concepts, processes, and relationships. This framework helps in generating concept embeddings that effectively capture the semantic essence and contextual relevance of these elements. Once the key concepts are identified and their
15 embeddings generated, DSAA abstracts this metadata into a structured format suitable for DataVault2.0 modeling. This abstraction involves mapping the generated concept embeddings to corresponding schema components—HUBS, LINKS, and SATELLITES—while meticulously preserving semantic integrity and ensuring traceability. This step is crucial as it transitions raw metadata into a structured schema that reflects the business's
20 underlying processes and relationships accurately.

Following metadata abstraction, DSAA proceeds to generate the DataVault2.0 schemas. This phase leverages the LLM's reasoning abilities to create semantically rich and well-organized schema components. DSAA constructs HUBS to represent core business entities, LINKS to depict the relationships among these entities, and SATELLITES to capture
25 detailed, time-varying attributes associated with both HUBS and LINKS. Each component is crafted to ensure it aligns with established data vault modeling best practices and the semantic framework provided by ISO standards. DSAA applies data vault modeling best practices such as denormalization and historicization, tailoring the schema to enhance data retrieval efficiency and scalability. This optimization ensures that the schemas are not only
30 technically sound but also aligned with the operational and analytical needs of the business.

A feature of DSAA's operation is maintaining bidirectional traceability and lineage. This process involves tracking the relationships between the original ISO metadata elements and the newly generated DataVault2.0 schema components. Such traceability is vital for effective impact analysis and change management, allowing businesses to understand how
35 alterations in metadata affect the schema and vice versa. Furthermore, DSAA conducts

thorough validations of the generated schemas against ISO 11179 and ISO 20022 conformance rules and performs industry-specific data quality checks. This validation ensures that the schemas are not only compliant with international standards but also meet high data quality benchmarks. Finally, once validated, the DataVault2.0 schemas are forwarded to the Cross-Taxonomy Mapping and Alignment Agent for further processing. This step integrates the schemas into the broader MEGAN system, enabling comprehensive data management and facilitating advanced data analytics capabilities.

Cross-Taxonomy Mapping and Alignment Agent (CTMAA):

The Cross-Taxonomy Mapping and Alignment Agent (CTMAA) is a highly advanced component of MEGAN that leverages the neural capabilities of large language models (LLMs) like Claude-3 to automate the mapping and translation of metadata across different taxonomies, jurisdictions, and financial messaging standards. By harnessing the power of LLMs, CTMAA can understand and process the complex semantic relationships, conceptual data modeling principles defined in ISO 11179 and ISO 20022 standards, and industry-specific ontologies like the Financial Industry Business Ontology (FIBO).

CTMAA employs LLMs' natural language understanding and reasoning capabilities to analyze the ISO 11179/ISO 20022 compliant metadata and DataVault2.0 schemas received from the DataVault2.0 Schema Abstraction Agent (DSAA) and apply state-of-the-art AI and NLP techniques, such as ontology-based reasoning, semantic similarity measures, graph neural networks, and transfer learning, to establish mappings and translations between metadata elements from different taxonomies and standards. These techniques enable CTMAA to identify and extract meaningful relationships and equivalences between metadata elements, ensuring consistent and context-aware mappings.

Moreover, CTMAA utilizes the reasoning abilities of LLMs to process ISO 11179 classification schemes, value domains, ISO 20022 business process catalogs, message component dictionaries, and data dictionaries to establish interoperability and semantic alignment across regulatory and enterprise contexts and different financial messaging domains. The LLM's ability to understand and process these complex semantic structures allows CTMAA to create accurate and comprehensive mappings that facilitate seamless data exchange and integration between agents and domains.

CTMAA also generates ISO 11179 and ISO 20022-compliant cross-references, mapping specifications, and transformation rules by leveraging LLMs' natural language generation capabilities. These artifacts provide a transparent and traceable record of the mappings and alignments, enabling users to understand and validate the relationships between metadata elements across different contexts.

To ensure the quality and consistency of the generated mappings and alignments, CTMAA validates them against ISO 11179 and ISO 20022 semantic constraints, data quality rules, and industry-specific reconciliation and validation frameworks using the LLM's ability to understand and apply complex validation rules. This allows CTMAA to identify and flag any issues or inconsistencies in the mappings, ensuring their accuracy and reliability.

Finally, CTMAA sends the ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations to the Vector Database Agent and Large Language Model Agent for further processing and integration into the MEGAN. Using LLMs enables CTMAA to handle the complexity and variability of cross-taxonomy mapping and alignment tasks, adapting to new standards and domain-specific requirements as needed.

By leveraging the neural capabilities of LLMs, CTMAA achieves a high level of automation, accuracy, and efficiency in mapping and translating metadata across different taxonomies, jurisdictions, and financial messaging standards. This reduces manual effort, ensures the quality and semantic integrity of the mappings, and facilitates the overall effectiveness of MEGAN in generating ISO 11179 and UN/CEFACT compliant metadata registries, abstracting DataVault2.0 schemas, and enabling cross-taxonomy mapping and translation.

Methods

CTMAA employs advanced AI and NLP techniques, semantic reasoning, and validation methods to ensure the quality, consistency, and compliance of the cross-taxonomy mappings and alignments with ISO 11179 and ISO 20022 standards.

Ontology-based Reasoning and Semantic Similarity:

CTMAA leverages ontology-based reasoning and semantic similarity techniques to establish mappings and translations between metadata elements from different taxonomies and standards. The mapping process involves the following steps:

Ontology Alignment: CTMAA aligns the ISO 11179 and ISO 20022 ontologies with industry-specific ontologies like FIBO, using LLM-based techniques like ontology matching, concept embedding, or graph alignment, to establish a common semantic framework for mapping metadata elements across different taxonomies and standards.

Semantic Similarity Computation: CTMAA computes semantic similarity scores between metadata elements from different taxonomies and standards using LLM-based techniques like word embeddings, sentence embeddings, or graph embeddings, considering their semantic context, relationships, and conceptual equivalence.

Mapping Candidate Generation: CTMAA generates mapping candidates between metadata elements based on their semantic similarity scores, ontology alignments, and domain-specific mapping rules, leveraging the LLM's reasoning capabilities to identify the most likely and meaningful mappings.

- 5 Mapping Validation and Refinement: CTMAA validates the generated mapping candidates against ISO 11179 and ISO 20022 semantic constraints, data quality rules, and industry-specific reconciliation frameworks, using the LLM's ability to understand and apply complex validation rules, and refines the mappings based on the validation results.

10 The time complexity of the ontology-based reasoning and semantic similarity process depends on the size and complexity of the ISO 11179 and ISO 20022 ontologies, the number and diversity of the metadata elements, and the efficiency of the ontology alignment, semantic similarity computation, and mapping generation algorithms employed. The size of the ontologies determines the space complexity, the semantic similarity matrices, and any intermediate data structures used during the mapping process.

- 15 Graph Neural Networks and Transfer Learning: CTMAA applies graph neural networks (GNNs) and transfer learning techniques to model the complex relationships and dependencies between metadata elements across different taxonomies and standards and to adapt the mapping and alignment models to new domains and contexts. The GNN-based mapping process involves the following steps:

- 20 Graph Representation Learning: CTMAA constructs graph representations of the ISO 11179 and ISO 20022 metadata elements, their relationships, and their alignments with industry-specific ontologies, using techniques like knowledge graphs, property graphs, or hypergraphs, to capture the rich semantic structure of the metadata.

25 Graph Neural Network Training: CTMAA trains GNN models, such as Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), or Relational Graph Convolutional Networks (RGCNs), on the constructed graph representations, using supervised or unsupervised learning techniques, to learn the complex patterns and dependencies between metadata elements across different taxonomies and standards.

30 Transfer Learning and Domain Adaptation: CTMAA applies transfer learning techniques, such as fine-tuning, domain adaptation, or meta-learning, to adapt the trained GNN models to new taxonomies, jurisdictions, or financial messaging standards, leveraging the LLM's ability to generalize and transfer knowledge across different domains and contexts.

Mapping Inference and Refinement: CTMAA infers the cross-taxonomy mappings and alignments using the trained and adapted GNN models, considering the learned patterns

and dependencies between metadata elements. It then refines the mappings based on ontology-based reasoning and semantic similarity techniques.

The time complexity of the GNN-based mapping process depends on the size and complexity of the graph representations, the number and diversity of the metadata elements and their relationships, and the efficiency of the GNN training, transfer learning, and inference algorithms employed. The size of the graph representations determines the space complexity, the GNN model parameters, and any intermediate data structures used during the learning and inference process.

Mapping Artifact Generation and Validation: CTMAA generates ISO 11179 and ISO 20022-compliant cross-references, mapping specifications, and transformation rules to document the cross-taxonomy mappings and alignments and validates them against semantic constraints and data quality rules. The artifact generation and validation process involves the following steps:

Cross-reference Generation: CTMAA generates ISO 11179-compliant cross-references between metadata elements from different taxonomies and standards, using the LLM's natural language generation capabilities to create human-readable and machine-processable descriptions of the mappings and their semantics.

Mapping Specification Generation: CTMAA generates ISO 20022-compliant mapping specifications, detailing the relationships, equivalences, and derivations between metadata elements from different message types, business processes, and data dictionaries, using the LLM's ability to understand and generate complex data structures and specifications.

Transformation Rule Generation: CTMAA generates executable transformation rules, such as XSLT stylesheets, SQL scripts, or API specifications, to enable the automated translation and conversion of data between different taxonomies and standards. This leverages the LLM's ability to generate code and data transformation pipelines.

Artifact Validation and Consistency Checking: CTMAA validates the generated cross-references, mapping specifications, and transformation rules against the ISO 11179 and ISO 20022 semantic constraints, data quality rules, and industry-specific reconciliation frameworks, using the LLM's ability to understand and apply complex validation rules, and checks their consistency and completeness.

The time complexity of the artifact generation and validation process depends on the number and complexity of the cross-taxonomy mappings, the size and diversity of the ISO 11179 and ISO 20022 standards and specifications, and the efficiency of the natural language generation, transformation rule generation, and validation algorithms employed. The size of

the generated artifacts determines the space complexity, the validation rule sets, and any intermediate data structures used during the generation and validation process.

Business Layer

Processes:

- 5 ISO 11179/ISO 20022 Metadata and Schema Acquisition: This process involves acquiring the ISO 11179/ISO 20022 compliant metadata and DataVault2.0 schemas from the DataVault2.0 Schema Abstraction Agent (DSAA) and preparing them for the cross-taxonomy mapping and alignment process.

- 10 Ontology Alignment and Semantic Similarity Computation: This process aligns the ISO 11179 and ISO 20022 ontologies with industry-specific ontologies like FIBO and computing semantic similarity scores between metadata elements from different taxonomies and standards.

- 15 Graph-based Mapping and Transfer Learning: This process involves constructing graph representations of the metadata elements and their relationships, training GNN models to learn the complex patterns and dependencies, and adapting the models to new domains and contexts using transfer learning techniques.

- 20 Mapping Artifact Generation and Validation: This process generates ISO 11179 and ISO 20022-compliant cross-references, mapping specifications, and transformation rules and validates them against semantic constraints, data quality rules, and industry-specific reconciliation frameworks.

- Mapped Metadata and Schema Delivery: This process involves delivering the ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations to the Vector Database Agent and Large Language Model Agent for further processing and integration.

- 25 Functions:

Automated Cross-Taxonomy Mapping: CTMAA enables automated mapping and translation of metadata elements across different taxonomies, jurisdictions, and financial messaging standards, reducing manual effort and improving efficiency.

- 30 Semantic Reasoning and Similarity Analysis: CTMAA leverages ontology-based reasoning and semantic similarity techniques to establish meaningful and context-aware mappings between metadata elements from different taxonomies and standards.

Graph-based Relationship Modeling: CTMAA applies GNNs and transfer learning techniques to model the complex relationships and dependencies between metadata elements across different taxonomies and standards and to adapt the mapping models to new domains and contexts.

- 5 ISO-Compliant Artifact Generation: CTMAA generates ISO 11179 and ISO 20022-compliant cross-references, mapping specifications, and transformation rules, providing a transparent and traceable record of the mappings and alignments.

Rigorous Validation and Quality Assurance: CTMAA validates the generated mappings and artifacts against semantic constraints, data quality rules, and industry-specific reconciliation
10 frameworks, ensuring accuracy, consistency, and reliability.

Application Layer

The CTMAA comprises the following key components:

MetadataSchemaAcquisitionManager: This component acquires the ISO 11179/ISO 20022 compliant metadata and DataVault2.0 schemas from the DataVault2.0 Schema Abstraction
15 Agent (DSAA) and prepares them for the cross-taxonomy mapping and alignment process.

OntologyAlignmentEngine: This component aligns the ISO 11179 and ISO 20022 ontologies with industry-specific ontologies like FIBO and computes semantic similarity scores between metadata elements from different taxonomies and standards.

GraphMappingLearner: This component constructs graph representations of the metadata
20 elements and their relationships, trains GNN models to learn the complex patterns and dependencies, and adapts the models to new domains and contexts using transfer learning techniques.

MappingArtifactGenerator: This component generates ISO 11179 and ISO 20022-compliant cross-references, mapping specifications, and transformation rules and validates them
25 against semantic constraints, data quality rules, and industry-specific reconciliation frameworks.

MappedMetadataDeliveryManager: This component delivers the ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations to the Vector Database Agent and Large Language Model Agent for further
30 processing and integration.

Services:

MetadataSchemaAcquisitionService: Provides methods for acquiring ISO 11179/ISO 20022 compliant metadata and DataVault2.0 schemas from the DataVault2.0 Schema Abstraction Agent (DSAA) and preparing them for the cross-taxonomy mapping and alignment process.

OntologyAlignmentService: Offers services for aligning ISO 11179 and ISO 20022

- 5 ontologies with industry-specific ontologies like FIBO and computing semantic similarity scores between metadata elements from different taxonomies and standards.

GraphMappingLearningService: This service enables the construction of graph representations of metadata elements and their relationships, training GNN models to learn complex patterns and dependencies, and adapting models to new domains and contexts

10 using transfer learning techniques.

MappingArtifactGenerationService: Facilitates the generation of ISO 11179 and ISO 20022-compliant cross-references, mapping specifications, transformation rules, and their validation against semantic constraints, data quality rules, and industry-specific reconciliation frameworks.

- 15 MappedMetadataDeliveryService: Provides services for delivering ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations to the Vector Database Agent and Large Language Model Agent for further processing and integration.

Interfaces:

- 20 MetadataSchemaAcquisitionInterface: Defines the methods and parameters for acquiring ISO 11179/ISO 20022 compliant metadata and DataVault2.0 schemas from the DataVault2.0 Schema Abstraction Agent (DSAA) and preparing them for the cross-taxonomy mapping and alignment process.

OntologyAlignmentInterface: Specifies the methods and input/output formats for aligning ISO

25 11179 and ISO 20022 ontologies with industry-specific ontologies like FIBO and computing semantic similarity scores between metadata elements from different taxonomies and standards.

GraphMappingLearningInterface: Describes the methods and input/output formats for constructing graph representations of metadata elements and their relationships, training

30 GNN models to learn complex patterns and dependencies, and adapting models to new domains and contexts using transfer learning techniques.

MappingArtifactGenerationInterface: This interface defines the methods and parameters for generating ISO 11179 and ISO 20022-compliant cross-references, mapping specifications,

and transformation rules and validating them against semantic constraints, data quality rules, and industry-specific reconciliation frameworks.

MappedMetadataDeliveryInterface: Specifies the methods and input/output formats for delivering ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations to the Vector Database Agent and Large Language Model Agent for further processing and integration.

Vector Database Agent (VDA):

The Vector Database Agent (VDA) is a highly advanced component of MEGAN that leverages state-of-the-art vector database technologies and semantic indexing techniques to store, retrieve, and analyze ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations. VDA enables efficient and contextual discovery, exploration, and utilization of financial industry metadata and knowledge by harnessing the power of vector databases and semantic similarity search.

VDA employs advanced indexing and retrieval techniques, such as hierarchical navigable small world graphs, approximate nearest neighbor search, and locality-sensitive hashing, to organize and index the metadata elements, relationships, mappings, and semantic representations received from the Cross-Taxonomy Mapping and Alignment Agent (CTMAA). These techniques enable fast, accurate, and context-aware retrieval of relevant metadata based on ISO 11179 and ISO 20022 semantic similarity and relevance, supporting various financial industry use cases and regulatory compliance needs.

Moreover, VDA implements domain-specific optimizations tailored to financial industry requirements, such as taxonomic navigation, faceted search, and semantic query expansion, to enhance the usability and effectiveness of metadata discovery and retrieval. These optimizations leverage the rich semantic information and contextual knowledge encoded in the ISO 11179/ISO 20022 metadata and schemas to provide intuitive and user-friendly querying experiences.

VDA provides a scalable, high-performance, and API-driven querying interface that allows users and applications to search, retrieve, and explore ISO 11179/ISO 20022 compliant metadata, schemas, mappings, and related artifacts based on semantic similarity, contextual relevance, and user-defined criteria. This interface supports complex querying scenarios, such as cross-taxonomy and cross-jurisdiction metadata discovery, impact analysis, and semantic traceability, enabling users to navigate and analyze metadata across different financial industry domains and regulatory contexts.

Furthermore, VDA offers advanced analytics and visualization capabilities, such as semantic clustering, topic modeling, and network analysis, to derive insights, patterns, and relationships from the ISO 11179/ISO 20022 metadata and schemas. These capabilities leverage the rich semantic information and latent structures encoded in the vector representations to uncover hidden connections, trends, and anomalies within the financial industry metadata, supporting data-driven decision-making and knowledge discovery.

VDA achieves high performance, scalability, and semantic richness in storing, retrieving, and analyzing ISO 11179/ISO 20022 aligned metadata and schemas by leveraging vector database technologies and semantic indexing techniques. This enables efficient and contextual discovery, exploration, and utilization of financial industry metadata and knowledge, supporting various use cases and regulatory compliance needs. VDA's advanced querying, analytics, and visualization capabilities empower users and applications to derive valuable insights and make informed decisions based on the rich semantic information and contextual knowledge encoded in the metadata.

Methods

VDA employs advanced vector database technologies, semantic indexing techniques, and domain-specific optimizations to ensure efficient storage, retrieval, and analysis of ISO 11179/ISO 20022 aligned metadata and schemas.

Vector Database Storage and Indexing: VDA stores and indexes the ISO 11179/ISO 20022 compliant metadata, schemas, mappings, and semantic representations received from CTMAA in a high-performance, scalable, and distributed vector database. The storage and indexing process involves the following steps:

Metadata Ingestion: VDA ingests the ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations received from CTMAA, parsing and transforming them into a suitable format for vector database storage and indexing.

Vector Embedding Generation: VDA generates vector embeddings for the metadata elements, relationships, mappings, and semantic representations using techniques like word embeddings, graph embeddings, or semantic encoders, capturing their semantic meaning and contextual information in a dense, continuous vector space.

Indexing and Partitioning: VDA indexes the generated vector embeddings using advanced techniques, such as hierarchical navigable small-world graphs, approximate nearest neighbor search, or locality-sensitive hashing, to enable fast and accurate similarity-based retrieval. It also partitions the vector database based on domain-specific criteria, such as

financial industry taxonomies, jurisdictions, or data governance standards, to optimize storage and retrieval performance.

Metadata Persistence: VDA persists the indexed vector embeddings, along with their associated metadata, schemas, mappings, and semantic representations, in a distributed and fault-tolerant manner, ensuring high availability, scalability, and data durability.

The time complexity of the vector database storage and indexing process depends on the volume and dimensionality of the metadata elements, the complexity of the vector embedding generation and indexing algorithms, and the efficiency of the database partitioning and persistence mechanisms. The space complexity is determined by the size of the vector embeddings, the associated metadata, and any auxiliary data structures used for indexing and partitioning.

Semantic Similarity Search and Retrieval: VDA enables efficient and contextual retrieval of ISO 11179/ISO 20022 compliant metadata, schemas, mappings, and related artifacts based on semantic similarity and relevance. The search and retrieval process involves the following steps:

Query Parsing and Expansion: VDA parses and analyzes the user queries, extracting relevant keywords, entities, or semantic concepts. It then expands the queries using techniques like semantic query expansion, synonym resolution, or ontology-based reasoning to enhance the recall and relevance of the search results.

Vector Embedding Retrieval: VDA retrieves the vector embeddings of the metadata elements, relationships, mappings, and semantic representations that are semantically similar or relevant to the expanded user queries, using the indexed vector database and efficient similarity search algorithms like cosine similarity, Euclidean distance, or maximum inner product search.

Ranking and Filtering: VDA ranks the retrieved vector embeddings based on semantic similarity scores, contextual relevance, and user-defined criteria, using techniques like TF-IDF weighting, BM25 scoring, or learning-to-rank models. It also applies domain-specific filters, such as taxonomic constraints, faceted navigation, or data governance rules, to refine the search results.

Result Aggregation and Presentation: VDA aggregates the ranked and filtered search results, retrieving the associated metadata, schemas, mappings, and semantic representations from the vector database. It then presents the results to the users or applications in a structured and intuitive format, along with relevant contextual information and navigational cues.

The time complexity of the semantic similarity search and retrieval process depends on the size of the vector database, the dimensionality of the vector embeddings, the complexity of the query expansion and ranking algorithms, and the efficiency of the similarity search and result aggregation mechanisms. The space complexity is determined by the size of the retrieved vector embeddings, the associated metadata, and any intermediate data structures used for ranking and filtering.

Domain-Specific Optimizations and Analytics: VDA implements domain-specific optimizations and analytics capabilities to enhance the usability, performance, and insights derived from the ISO 11179/ISO 20022 metadata and schemas. The optimization and analytics process involves the following steps:

Taxonomic Navigation and Faceted Search: VDA optimizes the search and retrieval process for financial industry taxonomies and hierarchies by implementing taxonomic navigation and faceted search capabilities. It leverages the hierarchical relationships and semantic attributes encoded in the ISO 11179/ISO 20022 metadata to enable users to browse and filter search results based on taxonomic categories, facets, or properties.

Cross-Taxonomy and Cross-Jurisdiction Querying: VDA supports complex querying scenarios, such as cross-taxonomy and cross-jurisdiction metadata discovery, by leveraging the cross-taxonomy mappings and semantic representations stored in the vector database. Semantic similarity and alignment techniques enable users to search and navigate metadata across different financial industry domains, standards, and jurisdictions.

Semantic Clustering and Topic Modeling: VDA applies semantic clustering and topic modeling techniques, such as k-means clustering, hierarchical clustering, or latent Dirichlet allocation, to group and categorize the ISO 11179/ISO 20022 metadata and schemas based on their semantic similarity and latent themes. This helps users discover and explore related metadata, identify patterns and trends, and gain insights into the semantic structure of the financial industry knowledge.

Impact Analysis and Semantic Traceability: VDA enables impact analysis and semantic traceability by leveraging the cross-taxonomy mappings, semantic representations, and contextual information stored in the vector database. It allows users to assess the potential impact of changes in metadata, schemas, or regulations on related artifacts and to trace the semantic lineage and dependencies across different financial industry domains and standards.

The time complexity of the domain-specific optimizations and analytics process depends on the volume and complexity of the ISO 11179/ISO 20022 metadata and schemas, the

efficiency of the taxonomic navigation and faceted search algorithms, the complexity of the semantic clustering and topic modeling techniques, and the performance of the impact analysis and traceability mechanisms. The space complexity is determined by the size of the vector embeddings, the associated metadata, and any intermediate data structures used for clustering, modeling, and analysis.

Business Layer

Processes:

Metadata Ingestion and Vector Embedding Generation: This process involves ingesting the ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations received from CTMAA, and generating vector embeddings that capture their semantic meaning and contextual information.

Vector Database Storage and Indexing: This process focuses on storing and indexing the generated vector embeddings, along with their associated metadata, schemas, mappings, and semantic representations, in a high-performance, scalable, and distributed vector database, using advanced techniques like hierarchical navigable small world graphs, approximate nearest neighbor search, or locality-sensitive hashing.

Semantic Similarity Search and Retrieval: This process involves parsing and expanding user queries, retrieving semantically similar or relevant vector embeddings from the indexed vector database, ranking and filtering the search results based on semantic similarity scores, contextual relevance, and user-defined criteria, and aggregating and presenting the results to users or applications.

Domain-Specific Optimizations and Analytics: This process focuses on implementing domain-specific optimizations and analytics capabilities, such as taxonomic navigation, faceted search, cross-taxonomy and cross-jurisdiction querying, semantic clustering, topic modeling, impact analysis, and semantic traceability, to enhance the usability, performance, and insights derived from the ISO 11179/ISO 20022 metadata and schemas.

API-Driven Querying and Result Delivery: This process involves providing a scalable, high-performance API-driven querying interface for users and applications to search, retrieve, and explore ISO 11179/ISO 20022 compliant metadata, schemas, mappings, and related artifacts based on semantic similarity, contextual relevance, and user-defined criteria and delivering the results in a structured and intuitive format.

Functions:

Efficient and Scalable Metadata Storage: VDA enables efficient and scalable storage of ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations in a high-performance, distributed vector database, ensuring high availability, fault tolerance, and data durability.

- 5 Semantic Similarity-Based Retrieval: VDA facilitates fast, accurate, and context-aware retrieval of relevant metadata, schemas, mappings, and artifacts based on semantic similarity and relevance, using advanced indexing techniques and efficient similarity search algorithms.

- 10 Domain-Specific Optimizations: VDA implements domain-specific optimizations tailored to financial industry requirements, such as taxonomic navigation, faceted search, and semantic query expansion, to enhance the usability and effectiveness of metadata discovery and retrieval.

- 15 Advanced Analytics and Visualization: VDA offers advanced analytics and visualization capabilities, such as semantic clustering, topic modeling, and network analysis, to derive insights, patterns, and relationships from the ISO 11179/ISO 20022 metadata and schemas, supporting data-driven decision-making and knowledge discovery.

Application Layer

The VDA comprises the following key components:

- 20 MetadataIngestionProcessor: This component ingests the ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations received from CTMAA, and preprocesses them for vector embedding generation and database storage.

- 25 VectorEmbeddingGenerator: This component generates vector embeddings for the metadata elements, relationships, mappings, and semantic representations using techniques like word embeddings, graph embeddings, or semantic encoders, capturing their semantic meaning and contextual information.

- 30 VectorDatabaseIndexer: This component indexes the generated vector embeddings, along with their associated metadata, schemas, mappings, and semantic representations, in a high-performance, scalable, and distributed vector database, using advanced techniques like hierarchical navigable small world graphs, approximate nearest neighbor search, or locality-sensitive hashing.

SemanticSimilaritySearchEngine: This component enables fast, accurate, and context-aware retrieval of relevant metadata, schemas, mappings, and artifacts based on semantic similarity and relevance, using efficient similarity search algorithms and ranking techniques.

DomainOptimizationAnalytics: This component implements domain-specific optimizations and analytics capabilities, such as taxonomic navigation, faceted search, cross-taxonomy querying, semantic clustering, topic modeling, impact analysis, and semantic traceability, to enhance the usability, performance, and insights derived from the ISO 11179/ISO 20022 metadata and schemas.

Services:

MetadataIngestionService: Provides methods for ingesting and preprocessing ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations received from CTMAA.

VectorEmbeddingService: Offers services for generating vector embeddings for metadata elements, relationships, mappings, and semantic representations, capturing their semantic meaning and contextual information.

VectorDatabaseIndexingService: This service enables the indexing and storage of vector embeddings, along with their associated metadata, schemas, mappings, and semantic representations, in a high-performance, scalable, and distributed vector database.

SemanticSimilaritySearchService: This service facilitates the fast, accurate, and context-aware retrieval of relevant metadata, schemas, mappings, and artifacts based on semantic similarity and relevance.

DomainOptimizationAnalyticsService: Provides services for implementing domain-specific optimizations and analytics capabilities, such as taxonomic navigation, faceted search, cross-taxonomy querying, semantic clustering, topic modeling, impact analysis, and semantic traceability.

QueryingAPIService: Offers a scalable, high-performance, and API-driven interface for users and applications to search, retrieve, and explore metadata, schemas, mappings, and artifacts based on semantic similarity, contextual relevance, and user-defined criteria.

Interfaces:

MetadataIngestionInterface: This interface defines the methods and parameters for ingesting and preprocessing ISO 11179/ISO 20022 aligned metadata, DataVault2.0 schemas, cross-taxonomy mappings, and semantic representations received from CTMAA.

VectorEmbeddingInterface: This interface specifies the methods and input/output formats for generating vector embeddings for metadata elements, relationships, mappings, and semantic representations.

5 VectorDatabaseIndexingInterface: Describes the methods and parameters for indexing and storing vector embeddings, along with their associated metadata, schemas, mappings, and semantic representations, in a high-performance, scalable, and distributed vector database.

SemanticSimilaritySearchInterface: Defines the methods and input/output formats for fast, accurate, and context-aware retrieval of relevant metadata, schemas, mappings, and artifacts based on semantic similarity and relevance.

10 DomainOptimizationAnalyticsInterface: Specifies the methods and parameters for implementing domain-specific optimizations and analytics capabilities, such as taxonomic navigation, faceted search, cross-taxonomy querying, semantic clustering, topic modeling, impact analysis, and semantic traceability.

15 QueryingAPIInterface: Describes the methods and input/output formats for providing a scalable, high-performance, and API-driven querying interface for users and applications to search, retrieve, and explore metadata, schemas, mappings, and artifacts based on semantic similarity, contextual relevance, and user-defined criteria.

Data Lineage and Provenance Tracking Agent (DLPTA):

20 The Data Lineage and Provenance Tracking Agent (DLPTA) is a critical component of MEGAN that implements comprehensive data lineage and provenance tracking capabilities to capture and maintain a complete audit trail of the system's metadata management, schema generation, and mapping processes. By leveraging advanced graph-based and temporal modeling techniques, DLPTA enables end-to-end traceability, reproducibility, and governance of the generated artifacts.

25 DLPTA employs a combination of provenance graphs and temporal databases to represent and store the complex relationships and dependencies between data elements, schemas, mappings, and processing activities. This allows the agent to capture and persist detailed metadata about each step in the data lifecycle, including data sources, transformations, quality checks, validations, and usage, providing a rich and contextualized view of the data's
30 origin, evolution, and impact.

To ensure consistency and completeness of the audit trail, DLPTA integrates with the other agents in the MEGAN architecture, automatically capturing and propagating lineage and

provenance metadata as data flows through the system. This minimizes manual effort and reduces the risk of errors or omissions in the lineage and provenance records.

DLPTA provides intuitive querying and visualization interfaces that allow users to easily explore and analyze data lineage and provenance information. These interfaces enable users to trace the origin and evolution of specific data elements, schemas, and mappings, understand their dependencies and impact, and gain insights into the data's quality, consistency, and compliance with governance policies and regulatory requirements.

Moreover, DLPTA supports complex traceability and impact analysis scenarios, such as identifying the downstream consequences of metadata changes, troubleshooting data quality issues, and assessing compliance with data governance policies and regulatory requirements. By leveraging the rich lineage and provenance information captured by the agent, users can quickly identify and resolve issues, minimize the risk of data inconsistencies, and ensure the overall integrity and reliability of the generated artifacts.

To facilitate communication, collaboration, and knowledge sharing among stakeholders, DLPTA generates comprehensive lineage reports, data dictionaries, and data catalogs that document the end-to-end flow and transformation of metadata. These artifacts provide a clear and concise view of the data's lineage and provenance, enabling users to quickly understand the data's context, purpose, and quality.

Finally, DLPTA continuously monitors and validates the integrity and consistency of the captured lineage and provenance metadata, detecting and alerting any anomalies, gaps, or inconsistencies that may indicate data quality or compliance issues. This proactive monitoring ensures that the lineage and provenance information remains accurate, up-to-date, and trustworthy, supporting effective data governance and decision-making.

By implementing comprehensive data lineage and provenance tracking capabilities, DLPTA plays a crucial role in ensuring the traceability, reproducibility, and governance of MEGAN's metadata management, schema generation, and mapping processes. This enables organizations to maintain a complete and reliable audit trail of their data assets, comply with regulatory requirements, and make informed decisions based on a deep understanding of the data's origin, evolution, and impact.

Methods

DLPTA employs advanced graph-based and temporal modeling techniques, integration approaches, and monitoring and validation mechanisms to ensure comprehensive and reliable data lineage and provenance tracking.

Provenance Graph Modeling: DLPTA represents and stores the complex relationships and dependencies between data elements, schemas, mappings, and processing activities using provenance graphs. The provenance graph modeling process involves the following steps:

Provenance Data Capture: DLPTA captures detailed metadata about each step in the data lifecycle, including data sources, transformations, quality checks, validations, and usage, by integrating with the other agents in the MEGAN architecture and automatically extracting relevant provenance information.

Graph Construction: DLPTA constructs a directed acyclic graph (DAG) representation of the captured provenance data, where nodes represent data entities, processing activities, or agents, and edges represent the relationships or dependencies between them. The graph is enriched with temporal and contextual attributes to provide a comprehensive view of the data's lineage and provenance.

Graph Storage and Indexing: DLPTA stores the provenance graph in a graph database or a specialized provenance store that supports efficient querying, traversal, and analysis of the graph structure. The graph is indexed based on various attributes, such as data entity identifiers, timestamps, or provenance types, to enable fast and targeted retrieval of provenance information.

Graph Querying and Traversal: DLPTA provides graph querying and traversal capabilities, enabling users to explore and analyze the provenance graph quickly. This includes support for graph pattern matching, shortest path queries, reachability analysis, and subgraph extraction, allowing users to trace the lineage and impact of specific data elements, schemas, or mappings.

The time complexity of the provenance graph modeling process depends on the size and complexity of the data lifecycle, the number of entities and relationships captured, and the efficiency of the graph construction, storage, and querying algorithms employed. The space complexity is determined by the size of the provenance graph, the number of attributes and temporal dimensions stored, and any indexing structures used for efficient querying and traversal.

Temporal Lineage Modeling: DLPTA captures and represents the temporal aspects of data lineage and provenance using temporal database techniques. The temporal lineage modeling process involves the following steps:

Temporal Data Capture: DLPTA captures the temporal metadata associated with each provenance event, such as the start and end timestamps of processing activities, the valid

time and transaction time of data entities, and the temporal validity of relationships or dependencies.

Temporal Schema Design: DLPTA designs a temporal schema that extends the provenance graph model with temporal dimensions, such as valid time, transaction time, or bi-temporal attributes. The temporal schema allows for representing and querying the historical and current state of the data lineage and provenance.

Temporal Data Storage: DLPTA stores the temporal provenance data in a database system that supports temporal querying and reasoning, such as a temporal-relational database, a temporal graph database, or a specialized temporal provenance store. The temporal data is organized and indexed based on the temporal dimensions to enable efficient retrieval and analysis of provenance information across time.

Temporal Querying and Analysis: DLPTA provides temporal querying and analysis capabilities to enable users to explore and reason about the temporal aspects of data lineage and provenance. This includes support for time-slice queries, temporal aggregation, temporal joins, and temporal pattern matching, allowing users to understand the evolution and validity of data entities, schemas, and mappings over time.

The temporal lineage modeling process's time complexity depends on the temporal provenance data's size and complexity, the temporal dimensions' granularity and range, and the efficiency of the temporal schema design, storage, and querying techniques employed.

The space complexity is determined by the size of the temporal provenance data, the number of temporal attributes and dimensions stored, and any indexing structures used for efficient temporal querying and analysis.

Integration and Propagation: DLPTA integrates with the other agents in the MEGAN architecture to automatically capture and propagate lineage and provenance metadata as data flows through the system. The integration and propagation process involves the following steps:

Provenance Metadata Extraction: DLPTA defines standard interfaces and protocols for extracting provenance metadata from the other agents in the MEGAN architecture, such as the data ingestion, schema generation, and mapping agents. This includes specifying the format, structure, and semantics of the provenance metadata to be captured.

Provenance Metadata Propagation: DLPTA establishes communication channels and data flow mechanisms to propagate the captured provenance metadata across MEGAN's different agents and components. This ensures that the lineage and provenance information is consistently and continuously updated as data moves through the system.

Provenance Metadata Integration: DLPTA integrates the propagated provenance metadata into the centralized provenance graph and temporal lineage model, merging and reconciling any overlapping or conflicting information. This involves applying data integration techniques, such as entity resolution, schema matching, and data fusion, to ensure the consistency and accuracy of the integrated provenance metadata.

Provenance Metadata Synchronization: DLPTA implements mechanisms for synchronizing the provenance metadata across the agents and components of MEGAN, ensuring that all parties have access to the most up-to-date and consistent view of the data lineage and provenance. This may involve techniques such as distributed versioning, conflict resolution, and real-time updates.

The time complexity of the integration and propagation process depends on the number of agents and components involved, the volume and frequency of provenance metadata updates, and the efficiency of the extraction, propagation, integration, and synchronization mechanisms employed. The space complexity is determined by the size of the provenance metadata exchanged and stored across the different agents and components and any intermediate data structures used for integration and synchronization.

Monitoring and Validation: DLPTA continuously monitors and validates the integrity and consistency of the captured lineage and provenance metadata, detecting and alerting any anomalies, gaps, or inconsistencies. The monitoring and validation process involves the following steps:

Provenance Data Quality Checks: DLPTA defines and executes a set of data quality checks and validation rules to assess the completeness, accuracy, consistency, and timeliness of the captured provenance metadata. This includes checking for missing or invalid values, detecting inconsistencies or contradictions in the provenance graph or temporal lineage model, and verifying the adherence to predefined data quality standards or constraints.

Anomaly Detection: DLPTA applies anomaly detection techniques, such as statistical analysis, pattern matching, or machine learning algorithms, to identify unusual or suspicious patterns in the provenance metadata that may indicate data quality issues, data drift, or potential compliance violations. This involves establishing baseline profiles and thresholds for standard provenance patterns and detecting deviations or outliers from these profiles.

Alerting and Notification: DLPTA generates alerts and notifications when data quality issues, anomalies, or inconsistencies are detected in the provenance metadata. The alerts are triggered based on predefined rules and thresholds. They are sent to the relevant

stakeholders, such as data stewards, data owners, or compliance officers, for further investigation and resolution.

Provenance Data Cleansing and Reconciliation: DLPTA provides mechanisms for cleansing and reconciling the provenance metadata to address any detected data quality issues or inconsistencies. This includes applying data cleansing techniques, such as data standardization, deduplication, or data imputation, and reconciling conflicting or inconsistent provenance information across different sources or agents.

The time complexity of the monitoring and validation process depends on the volume and complexity of the provenance metadata, the number and sophistication of the data quality checks and validation rules, and the efficiency of the anomaly detection and alerting algorithms employed. The size of the provenance metadata determines the space complexity, the number of data quality metrics and thresholds maintained, and any intermediate data structures used for anomaly detection and cleansing.

Business Layer

Processes:

Provenance Metadata Capture and Extraction: This process involves capturing and extracting detailed metadata about each step in the data lifecycle, including data sources, transformations, quality checks, validations, and usage, by integrating with the other agents in the MEGAN architecture.

Provenance Graph Construction and Storage: This process focuses on constructing a directed acyclic graph (DAG) representation of the captured provenance data and storing it in a graph database or specialized provenance store that supports efficient querying, traversal, and analysis.

Temporal Lineage Modeling and Storage: This process involves designing a temporal schema that extends the provenance graph model with temporal dimensions and storing the temporal provenance data in a database system that supports temporal querying and reasoning.

Provenance Metadata Integration and Propagation: This process integrates the provenance metadata captured from different agents and components of MEGAN, propagates the metadata across the system, and synchronizes it to ensure consistency and accuracy.

Provenance Data Quality Monitoring and Validation: This process involves continuously monitoring and validating the integrity and consistency of the captured lineage and

provenance metadata, detecting anomalies, triggering alerts, and applying data cleansing and reconciliation techniques to address data quality issues.

Functions:

5 End-to-end Traceability and Reproducibility: DLPTA enables end-to-end traceability and reproducibility of the metadata management, schema generation, and mapping processes performed by MEGAN, providing a complete audit trail of the data's origin, evolution, and impact.

10 Comprehensive Provenance Capture and Storage: DLPTA captures and stores detailed metadata about each step in the data lifecycle using advanced graph-based and temporal modeling techniques, ensuring a rich and contextualized view of the data's lineage and provenance.

15 Seamless Integration and Propagation: DLPTA integrates with the other agents in the MEGAN architecture to automatically capture and propagate lineage and provenance metadata as data flows through the system, minimizing manual effort and ensuring consistency and completeness of the audit trail.

Intuitive Querying and Visualization: DLPTA provides intuitive querying and visualization interfaces that allow users to explore and analyze data lineage and provenance information quickly, enabling them to trace the origin and evolution of specific data elements, schemas, and mappings.

20 Proactive Data Quality Monitoring and Alerting: DLPTA continuously monitors and validates the integrity and consistency of the captured lineage and provenance metadata, detecting anomalies, triggering alerts, and applying data cleansing and reconciliation techniques to ensure the accuracy and reliability of the provenance information.

Application Layer

25 The DLPTA comprises the following key components:

ProvenanceMetadataExtractor: This component captures and extracts detailed metadata about each step in the data lifecycle by integrating with the other agents in the MEGAN architecture and automatically extracting relevant provenance information.

30 ProvenanceGraphConstructor: This component constructs a directed acyclic graph (DAG) representation of the captured provenance data, enriching it with temporal and contextual attributes, and stores it in a graph database or specialized provenance store.

TemporalLineageModeler: This component designs a temporal schema that extends the provenance graph model with temporal dimensions and stores the temporal provenance data in a database system that supports temporal querying and reasoning.

ProvenanceMetadataIntegrator: This component integrates the provenance metadata captured from different agents and components of MEGAN, propagates the metadata across the system, and synchronizes it to ensure consistency and accuracy.

ProvenanceDataQualityMonitor: This component continuously monitors and validates the integrity and consistency of the captured lineage and provenance metadata, detects anomalies, triggers alerts, and applies data cleansing and reconciliation techniques to address data quality issues.

LineageQueryVisualizer: This component provides intuitive querying and visualization interfaces that allow users to explore and analyze data lineage and provenance information quickly, enabling them to trace the origin and evolution of specific data elements, schemas, and mappings.

Services:

ProvenanceMetadataExtractionService: This service provides methods for capturing and extracting detailed metadata about each step in the data lifecycle by integrating with the other agents in the MEGAN architecture.

ProvenanceGraphConstructionService: Offers services for constructing a directed acyclic graph (DAG) representation of the captured provenance data, enriching it with temporal and contextual attributes, and storing it in a graph database or specialized provenance store.

TemporalLineageModelingService: Enables the design of a temporal schema that extends the provenance graph model with temporal dimensions and the storage of temporal provenance data in a database system that supports temporal querying and reasoning.

ProvenanceMetadataIntegrationService: Facilitates the integration of provenance metadata captured from different agents and components of MEGAN, propagating the metadata across the system and synchronizing it to ensure consistency and accuracy.

ProvenanceDataQualityMonitoringService: Provides services for continuously monitoring and validating the integrity and consistency of the captured lineage and provenance metadata, detecting anomalies, triggering alerts, and applying data cleansing and reconciliation techniques to address data quality issues.

LineageQueryVisualizationService: Offers intuitive querying and visualization interfaces that allow users to explore and analyze data lineage and provenance information quickly, enabling them to trace the origin and evolution of specific data elements, schemas, and mappings.

5 Interfaces:

ProvenanceMetadataExtractionInterface: This interface defines the methods and parameters for capturing and extracting detailed metadata about each step in the data lifecycle by integrating with the other agents in the MEGAN architecture.

10 ProvenanceGraphConstructionInterface: Specifies the methods and input/output formats for constructing a directed acyclic graph (DAG) representation of the captured provenance data, enriching it with temporal and contextual attributes, and storing it in a graph database or specialized provenance store.

15 TemporalLineageModelingInterface: Describes the methods and parameters for designing a temporal schema that extends the provenance graph model with temporal dimensions and storing the temporal provenance data in a database system that supports temporal querying and reasoning.

20 ProvenanceMetadataIntegrationInterface: Defines the methods and input/output formats for integrating provenance metadata captured from different agents and components of MEGAN, propagating the metadata across the system, and synchronizing it to ensure consistency and accuracy.

ProvenanceDataQualityMonitoringInterface: Specifies the methods and parameters for continuously monitoring and validating the integrity and consistency of the captured lineage and provenance metadata, detecting anomalies, triggering alerts, and applying data cleansing and reconciliation techniques to address data quality issues.

25 SEPHYR

SEPHYR (Self-Evolving Pattern Harmonization for Unified Reporting) is an agent-based module that harmonizes diverse data patterns and features into a unified and verifiable representation for reliable and efficient reporting across analytical and decision-making processes. At its core, SEPHYR employs a synergistic ensemble of three specialized agents
30 collaborating through continuous optimization cycles, overseen by a Petri-net orchestration model.

Key SEPHYR advantages:

Digital Native Mathematical Representation: This method represents data features and relationships using a unique and dynamic numbering system based on natural number sequences, ensuring non-overlapping and unified encoding.

5 Self-Cataloging Capabilities: Employ supervised learning models to enable self-discovery and cataloguing data columns, rows, or cells into a standard object/feature model.

Proof of Ownership Consensus: Utilizing a consensus method to define primary parent-child relationships and classify encoding ensures accurate data lineage and provenance.

Univariant and Multivariant Support: Incorporating mathematical algorithms to handle univariant (single feature) and multivariant (feature combinations) data at varying depths.

10 Cryptographic Verifiability: This involves employing SHA-256-based digital signatures to verify the authenticity and integrity of data records, enabling immutability and trustworthiness.

Agents of SEPHYR:

15 The SEPHYR agent architecture comprises three key agents, each with specific roles and responsibilities, working together to ensure the harmonization of data patterns and features:

SAFFRON (Self-Adaptive Feature Fusion and Representation Orchestration Network):
SAFFRON is responsible for identifying, extracting, encoding, and cataloging data features into a standard object/feature model. It leverages natural language processing, machine learning, mathematical algorithms, and consensus mechanisms to ensure accurate feature
20 representation, data lineage, and provenance.

Critical components of SAFFRON include:

FeatureIdentifier: Identifies and extracts relevant features from input data.

FeatureEncoder: Assigns unique numerical codes to features based on natural number sequences.

25 FeatureClassifier: Employs supervised learning models to catalog data into a common object/feature model.

OwnershipConsensus: Establishes consensus on primary parent-child relationships and classification encoding.

DataProcessor: Handles univariant and multivariant data processing.

30 FeatureStore: The central authority feature store for the unified data model, encoding, and classification schema.

GAMA (Governance-Aware Multilevel Access Management Architecture): GAMA translates user roles, requirements, and data access patterns into a unified mathematical representation for efficient role management, user access control, and security configuration. It employs natural language processing, machine learning, mathematical algorithms, consensus mechanisms, and cryptographic techniques.

Critical components of GAMA include:

RoleIdentifier: Identifies and extracts relevant roles and secure features.

RoleEncoder: Assigns unique numerical codes to roles and features.

AccessClassifier: Catalogs user access to roles and user associations to roles.

10 ManagementConsensus: Establishes consensus on regulatory requirements, toxic combinations, and data masking rules.

AccessControlEngine: The central access control engine for the unified security access model.

15 SENTINEL (Secure, Efficient, and Intelligent Access Control System): SENTINEL employs intelligent search algorithms to manage and apply security access rules efficiently, enabling granular control over user access to data based on roles, hierarchies, and regulatory or organizational policies.

Critical components of SENTINEL include:

UserProfileManager: Manages user profiles, roles, and positions.

20 SecurityRuleManager: Defines, maintains, and updates security rules.

AccessControlEngine: Enforces defined security access rules during user access attempts.

RuleAdaptationMonitor: Monitors changes in requirements and policies to ensure adaptation and compliance.

25 The SAFFRON, GAMA, and SENTINEL agents are orchestrated using a Petri-net model, which models their dependencies and interactions. This ensures the coordinated and efficient execution of data harmonization, access management, and security control tasks.

SAFFRON

SAFFRON (Self-Adaptive Feature Fusion and Representation Orchestration Network) is an advanced agent that harmonizes diverse data features into a unified and verifiable representation for efficient cataloging, search, matching, and metadata management. At its

core, SAFFRON employs a synergistic ensemble of specialized agents collaborating through continuous optimization cycles, overseen by a Petri-net orchestration model.

The SAFFRON agents, including FeatureIdentifier, FeatureEncoder, FeatureClassifier, OwnershipConsensus, DataProcessor, and FeatureStore, work in concert to identify, extract,
5 encode, and catalog data features into a standard object/feature model. This process involves leveraging natural language processing, machine learning, mathematical algorithms, and consensus mechanisms to ensure accurate feature representation, data lineage, and provenance.

SAFFRON's digital native mathematical representation, based on unique dynamic
10 numbering systems, ensures non-overlapping and unified encoding of data features and their relationships. Its self-cataloguing capabilities, powered by supervised learning models, enable automated discovery and cataloguing of data columns, rows, and cells, reducing manual effort and enabling scalability.

Furthermore, SAFFRON incorporates a proof of ownership consensus method to define
15 primary parent-child relationships and classification encoding, ensuring accurate data governance and regulatory compliance. It also supports univariant and multivariant data payloads, handling complex feature combinations at varying depths through versatile mathematical algorithms.

Underpinning SAFFRON's trustworthiness is its use of cryptographic verifiability through
20 SHA-256 digital signatures, verifying the authenticity, integrity, and immutability of data records stored in the central authority feature store. This robust feature store is the unified repository for efficient data search, matching, and metadata management across the consumption network. With its modular design and well-defined interfaces, SAFFRON integrates with enterprise data sources, identity systems, regulatory compliance platforms,
25 business intelligence tools, and model-serving environments, enabling flexible deployments in cloud, on-premises, or hybrid architectures.

Methods

The Self-Building Feature Store employs advanced techniques to convert analytical and reporting feature columns and values into a unified mathematical representation, enabling
30 efficient data cataloguing, search, matching, and metadata management, even for large-scale datasets.

Digital Native Mathematical Representation:

The Self-Building Feature Store represents data features and their relationships (parent, child, combination permutations) using a unique and dynamic numbering system based on the natural number sequence. This mathematical representation ensures non-overlapping and unified encoding of data values, enabling efficient processing and analysis. The process involves the following steps:

Feature Identification: The Self-Building Feature Store identifies and extracts relevant features from the input data, such as columns, rows, or cells, using natural language processing (NLP) and machine learning techniques.

Feature Encoding: Each identified feature is assigned a unique numerical code based on its position in the natural number sequence, ensuring non-overlapping representations.

Relationship Modeling: Using mathematical operations and data structures, the Self-Building Feature Store models the relationships between features, such as parent-child hierarchies and combination permutations.

Mathematical Representation: The encoded features and their relationships are combined to create a unified mathematical representation of the data, enabling efficient storage, retrieval, and analysis.

The time complexity of the digital native mathematical representation process depends on the number of features and their relationships and the complexity of the feature identification and encoding algorithms employed. The space complexity is determined by the size of the input data and the mathematical representation of data structures.

Self-Cataloging Feature Dictionary:

The Self-Building Feature Store employs supervised learning classification models to enable the system to self-discover and catalog columns, rows, or cells of data into a standard object/feature model. The self-cataloging process involves the following steps:

Training Data Preparation: The Self-Building Feature Store prepares a training dataset by manually labeling a subset of the input data with feature classifications and annotations.

Model Training: The system trains supervised learning classification models using the labeled training data, such as deep learning neural networks or ensemble methods.

Feature Classification: The trained models are used to classify and catalog the remaining input data into the shared object/feature model based on the learned patterns and relationships.

Model Retraining: The Self-Building Feature Store continuously monitors the classification performance and retrains the models with additional labeled data to improve accuracy and adapt to changes in the data distribution.

The time complexity of the self-cataloguing process depends on the size of the input data, the complexity of the classification models, and the number of training iterations required. The size of the training data and the trained classification models determine the space complexity.

Proof of Ownership Consensus:

The Self-Building Feature Store employs a consensus method using proof of ownership to define primary parent-child relationships and classification encoding. The consensus process involves the following steps:

Ownership Claim Submission: Data owners or stakeholders submit claims of ownership over specific features or feature combinations, along with supporting evidence or documentation.

Claim Validation: The Self-Building Feature Store validates the submitted claims by verifying the provided evidence and cross-checking against existing ownership records.

Consensus Calculation: The system calculates a consensus score for each claim based on the strength of the evidence, the number of supporting claims, and the reputation or trust score of the claimants.

Consensus Resolution: The Self-Building Feature Store resolves conflicting claims and establishes the primary parent-child relationships and classification encoding based on the consensus scores and a predefined resolution mechanism.

The time complexity of the proof of ownership consensus process depends on the number of ownership claims, the complexity of the validation and consensus calculation algorithms, and the number of stakeholders involved. The space complexity is determined by the size of the ownership claim data and the associated metadata. By employing these methods, the Self-Building Feature Store enables the creation of a unified data model, encoding, and classification schema within a central authority feature store, resulting in a common language across the consumption network and facilitating efficient data cataloguing, search, matching, and metadata management.

Business Layer

Processes:

Data Ingestion and Preprocessing: This process involves collecting and preprocessing data from various sources, ensuring data quality and consistency through validation and transformation techniques.

- 5 Feature Identification and Encoding: This process focuses on identifying and extracting relevant features from the input data, assigning unique numerical codes, and modeling their relationships using mathematical operations and data structures.

- 10 Feature Classification and Cataloging: This process employs supervised learning classification models to enable the system to self-discover and catalog columns, rows, or data cells into a common object/feature model, continuously improving classification accuracy through model retraining.

Ownership Consensus and Encoding: This process establishes a consensus on primary parent-child relationships and classification encoding using a proof of ownership method, resolving conflicting claims and ensuring accurate data lineage and provenance.

Functions:

- 15 Unified Data Representation: The Self-Building Feature Store enables the creation of a unified mathematical representation of data features and their relationships, facilitating efficient data processing, analysis, and storage.

- 20 Automated Feature Discovery and Cataloging: The Self-Building Feature Store automates the feature discovery and cataloging process by employing supervised learning classification models. This reduces manual effort and enables scalability to large-scale datasets.

Data Governance and Provenance: The proof of ownership consensus method ensures accurate data lineage and provenance, enabling effective data governance and supporting compliance with regulatory requirements.

- 25 Efficient Data Search and Matching: The unified data model, encoding, and classification schema within the central authority feature store enable efficient data search, matching, and metadata management across the consumption network.

Application Layer

The Self-Building Feature Store comprises the following key components:

- 30 FeatureIdentifier: This component identifies and extracts relevant features from the input data using NLP and machine learning techniques.

FeatureEncoder: This component assigns unique numerical codes to the identified features based on their position in the natural number sequence and models their relationships using mathematical operations and data structures.

5 FeatureClassifier: This component employs supervised learning classification models to catalog columns, rows, or data cells into a common object/feature model, continuously improving classification accuracy through model retraining.

OwnershipConsensus: This component establishes a consensus on primary parent-child relationships and classification encoding using a proof of ownership method, resolving conflicting claims and ensuring accurate data lineage and provenance.

10 FeatureStore: This component is the central authority feature store, storing the unified data model, encoding, and classification schema, enabling efficient data search, matching, and metadata management across the consumption network.

Services:

15 FeatureIdentificationService: Provides methods for identifying and extracting relevant features from the input data, using NLP and machine learning techniques.

FeatureEncodingService: Offers services for assigning unique numerical codes to the identified features and modeling their relationships using mathematical operations and data structures.

20 FeatureClassificationService: Enables the cataloguing of columns, rows, or cells of data into a standard object/feature model using supervised learning classification models, including model training and retraining capabilities.

OwnershipConsensusService: Facilitates the establishment of a consensus on primary parent-child relationships and classification encoding using a proof of ownership method, resolving conflicting claims and ensuring accurate data lineage and provenance.

25 FeatureStoreService: Provides services for storing, retrieving, and managing the unified data model, encoding, and classification schema within the central authority feature store, enabling efficient data search, matching, and metadata management across the consumption network.

Interfaces:

30 FeatureIdentifierInterface defines the methods and parameters for identifying and extracting relevant features from the input data using NLP and machine learning techniques.

FeatureEncoderInterface: Specifies the methods and parameters for assigning unique numerical codes to the identified features and modeling their relationships using mathematical operations and data structures.

FeatureClassifierInterface: Describes the methods and parameters for cataloguing columns, rows, or cells of data into a standard object/feature model using supervised learning classification models, including model training and retraining capabilities.

OwnershipConsensusInterface: Provides methods and parameters for establishing a consensus on primary parent-child relationships and classification encoding using a proof of ownership method, resolving conflicting claims and ensuring accurate data lineage and provenance.

FeatureStoreInterface defines the methods and parameters for storing, retrieving, and managing the unified data model, encoding, and classification schema within the central authority feature store. This interface enables efficient data search, matching, and metadata management across the consumption network.

FIG. 4 is a functional block diagram 400 illustrating an example of a data management and access control system within the multi-agent artificial intelligence framework, according to an embodiment herein. The data management and access control system correspond to an embodiment of the SEPHYR agent-based framework.

SAFFRON (Self-Adaptive Feature Fusion and Representation Orchestration Network) 402:

SAFFRON 402 is a data input interface and feature extraction module which identifies, extracts, encodes, and catalogs data features into a standardized model, serving as the foundational unit for data representation within the SEPHYR system.

SAFFRON 402 provides unified data representations as inputs to both GAMA 404 and SENTINEL 406, ensuring that these subsequent systems receive consistently processed and standardized data.

SAFFRON 402 maintains an association relationship with upstream data sources, from which it gathers and processes input data, ensuring a comprehensive and up-to-date data catalog.

SAFFRON 402 incorporates a security module that employs cryptographic hash functions to encrypt the extracted features, generating a secured data representation. A validation module in SAFFRON 402 validates this unified data representation against predefined standards, ensuring integrity and compliance.

GAMA (Governance-Aware Multilevel Access Management Architecture) 404:

GAMA 404 is an access management module which translates user roles, permissions, and policy information into integrated access models, serving as the system's core for access management and policy integration.

- 5 GAMA 404 provides these encoded access models as inputs to SENTINEL 406, which uses them to enforce access controls.

GAMA 404 also associates with SAFFRON's 402 data representations to align access models with the latest data features and with regulatory sources to ensure compliance and relevance in its policy applications.

- 10 GAMA 404 may encode user roles and access privileges into unique numerical codes and includes a control engine that serves as an authority for enforcing access controls based on a unified security access model. GAMA 404 is further configured to automatically synchronize its access control rules with external compliance monitoring systems to maintain adherence to regulatory changes.

15 SENTINEL (Secure, Efficient, and Intelligent Access Control System) 406:

SENTINEL 406 employs algorithms to manage and apply security access rules effectively across the SEPHYR system.

- SENTINEL 406 associates with both SAFFRON 402 and GAMA 404, receiving inputs that include unified data representations from SAFFRON and integrated access models from
20 GAMA.

SENTINEL 406 enforces access rules over the data model provided by SAFFRON 402 and applies the access control policies provided by GAMA 404, ensuring secure and efficient control over data access.

Summary of relationship shown in FIG. 4:

- 25 SAFFRON 402 acts as the primary provider of unified, encoded data representations, extracting, encoding, and cataloging features into a standard model. This foundational data is then served to both GAMA 404 and SENTINEL 406. GAMA 404 processes this data further by translating roles, permissions, and policies into coherent access control models, based on both the data representation from SAFFRON 402 and inputs from regulatory
30 sources. These models are then provided to SENTINEL 406. SENTINEL 406 utilizes these models to manage and enforce detailed access controls, applying sophisticated rules over

the data infrastructure shaped by SAFFRON's 402 representations and GAMA's 404 policy frameworks.

GAMA

GAMA (Governance-Aware Multilevel Access Management Architecture) is an advanced agent that translates user roles, requirements, and data access patterns into a unified mathematical representation for efficient role management, user access control, and security configuration across large-scale enterprise systems. At its core, GAMA employs a synergistic ensemble of specialized agents collaborating through continuous optimization cycles, overseen by a Petri-net orchestration model.

The GAMA agents, including RoleIdentifier, RoleEncoder, AccessClassifier, RegulationConsensus, and AccessControlEngine, work in concert to identify, extract, encode, classify, and enforce user roles and access privileges into a centralized access control model. This process involves leveraging natural language processing, machine learning, mathematical algorithms, consensus mechanisms, and cryptographic techniques to ensure accurate role representation, regulatory compliance, and auditable user access administration.

GAMA's digital native role representation is based on a unique dynamic numbering system that ensures non-overlapping and unified encoding of user roles, secure features, and intricate relationships, such as hierarchies and permission inheritance. Its self-cataloguing capabilities, driven by supervised learning models like deep neural networks, enable automated discovery and classification of roles, user-role associations, and access privileges, reducing manual effort and facilitating scalability.

Furthermore, GAMA incorporates a regulatory consensus method that defines toxic access combinations, data masking rules, and toxic permission constraints by establishing an evidence-based consensus among stakeholder claims. This consensus integration mechanism ensures adherence to compliance requirements and mitigates security risks.

Underpinning GAMA's trustworthiness is its use of cryptographic identity assurance through digital signatures, verifying user identities and enabling non-repudiation and secure auditing of access control activities. This robust auditing capability is facilitated by the central AccessControlEngine, which serves as the authoritative repository for the unified, encoded access control model, enabling efficient role management and user access administration.

With its modular design and well-defined interfaces, GAMA integrates with enterprise user directories, identity providers, identity and access management (IAM) systems, cloud access

brokers, and API gateways. Its containerized, microservices architecture enables flexible deployments across cloud, on-premises, or hybrid environments.

GAMA's architecture aligns with access governance frameworks like the Policy-Based Access Control (POLP) and Privacy by Design (PbD) principles, promoting ethical considerations such as privacy and the principle of least privilege access. GAMA's robust validation mechanisms, which include benchmarking against regulatory frameworks, enterprise policies, and simulated scenarios, as well as its comprehensive auditing capabilities, facilitate compliance with security and governance standards within enterprises.

Methods

- 10 The Dynamic Security Access system employs advanced techniques to translate user roles, requirements, and data access patterns into a unified mathematical representation, enabling efficient role management, user access control, and security configuration, even for large-scale systems with billions or trillions of records.

Digital Native Mathematical Representation of User Roles:

- 15 The Dynamic Security Access system represents user roles, secure features, and their relationships (parent, child, combination permutations), including child permission inheritance, using a unique and dynamic numbering system based on the natural number sequence. This mathematical representation ensures non-overlapping and unified encoding of roles, users, and access privileges, enabling efficient processing and analysis. The process involves the following steps:
- 20

Role and Feature Identification: The system identifies and extracts relevant roles and secure features from the input data, such as user management systems, organizational structures, and security policies, using natural language processing (NLP) and machine learning techniques.

- 25 Role and Feature Encoding: Each identified role and secure feature is assigned a unique numerical code based on its position in the natural number sequence, ensuring non-overlapping representations.

- Relationship Modeling: The Dynamic Security Access system models the relationships between roles and features, such as parent-child hierarchies, permission inheritance, and combination permutations, using mathematical operations and data structures.
- 30

Mathematical Representation: The encoded roles, features, and their relationships are combined to create a unified mathematical representation of the security access model, enabling efficient storage, retrieval, and analysis.

The time complexity of the digital native mathematical representation process depends on the number of roles, features, and relationships and the complexity of the identification and encoding algorithms employed. The space complexity is determined by the size of the input data and the mathematical representation of data structures.

5 Self-Cataloging Role and Access Dictionary:

The Dynamic Security Access system employs supervised learning classification models to enable the system to self-discover and catalog roles, user access to roles, and user associations to roles. The self-cataloging process involves the following steps:

10 Training Data Preparation: The system prepares a training dataset by manually labeling a subset of the input data with role classifications, access patterns, and user associations.

Model Training: The Dynamic Security Access system trains supervised learning classification models using the labeled training data, such as deep learning neural networks or ensemble methods.

15 Role and Access Classification: The trained models classify and catalog the remaining input data, including roles, user access to roles, and user associations, based on the learned patterns and relationships.

Model Retraining: The system continuously monitors the classification performance and retrains the models with additional labeled data to improve accuracy and adapt to changes in the security access model or user behavior patterns.

20 The time complexity of the self-cataloging process depends on the size of the input data, the complexity of the classification models, and the number of training iterations required. The size of the training data and the trained classification models determine the space complexity.

Proof of Management Consensus, Regulations, and Toxic Combinations:

25 The Dynamic Security Access system employs a consensus method using evidence of ownership to define all toxic combinations and regulatory requirements on data access, dynamic data masking permissions, and tokenization. The consensus process involves the following steps:

30 Requirement Claim Submission: Management, compliance officers, or stakeholders submit claims of regulatory requirements, toxic combinations, or data masking rules, along with supporting evidence or documentation.

Claim Validation: The system validates the submitted claims by verifying the provided evidence and cross-checking against existing regulatory frameworks and security policies.

Consensus Calculation: The Dynamic Security Access system calculates a consensus score for each claim based on the strength of the evidence, the number of supporting claims, and the reputation or trust score of the claimants.

Consensus Resolution: The system resolves conflicting claims and establishes the regulatory requirements, toxic combinations, and data masking rules based on the consensus scores and a predefined resolution mechanism.

The time complexity of the proof of management consensus process depends on the number of requirement claims, the complexity of the validation and consensus calculation algorithms, and the number of stakeholders involved. The space complexity is determined by the size of the requirement claim data and the associated metadata. By employing these methods, the Dynamic Security Access system creates a unified security access model, role encoding, and user association schema, facilitating efficient role management, user access control, and security configuration across the organization.

Business Layer

Processes:

User and Role Data Ingestion: This process involves collecting and ingesting user, role, and security access data from various sources, such as user management systems, organizational structures, and security policies.

Role and Feature Identification and Encoding: This process focuses on identifying and extracting relevant roles and secure features from the input data, assigning unique numerical codes, and modeling their relationships using mathematical operations and data structures.

Role and Access Classification and Cataloging: This process employs supervised learning classification models to enable the system to self-discover and catalog roles, user access to roles, and user associations to roles, continuously improving classification accuracy through model retraining.

Management Consensus and Security Configuration: This process establishes a consensus on regulatory requirements, toxic combinations, and data masking rules using a proof of ownership method, resolving conflicting claims and configuring the security access model accordingly.

Functions:

Unified Security Access Model: The Dynamic Security Access system creates a unified mathematical representation of roles, secure features, and their relationships, facilitating efficient role management, user access control, and security configuration.

- 5 Automated Role and Access Discovery: The system automates role and access discovery by employing supervised learning classification models, reducing manual effort and enabling scalability to large-scale systems.

Regulatory Compliance and Risk Mitigation: The proof of management consensus method ensures adherence to regulatory requirements, identification of toxic combinations, and appropriate data masking rules, mitigating security risks and supporting compliance efforts.

- 10 Efficient Access Control and Monitoring: The unified security access model and mathematical representation enable efficient user access control, monitoring, and integration with cyber security systems for enhanced protection and threat detection.

Application Layer

The Dynamic Security Access system comprises the following key components:

- 15 RoleIdentifier: This component identifies and extracts relevant roles and secure features from the input data, using NLP and machine learning techniques.

RoleEncoder: This component assigns unique numerical codes to the identified roles and features based on their position in the natural number sequence. It models their relationships using mathematical operations and data structures.

- 20 AccessClassifier: This component employs supervised learning classification models to catalog user access to roles and user associations to roles, continuously improving classification accuracy through model retraining.

- 25 ManagementConsensus: This component establishes a consensus on regulatory requirements, toxic combinations, and data masking rules using a proof of ownership method, resolving conflicting claims and configuring the security access model accordingly.

AccessControlEngine: This component is the central access control engine, storing the unified security access model, role encoding, and user association schema. It enables efficient role management, user access control, and security configuration.

Services:

- 30 RoleIdentificationService: Provides methods for identifying and extracting relevant roles and secure features from the input data, using NLP and machine learning techniques.

RoleEncodingService: This service offers services for assigning unique numerical codes to the identified roles and features and modeling their relationships using mathematical operations and data structures.

5 AccessClassificationService: Enables the cataloguing of user access to roles and user associations to roles using supervised learning classification models, including model training and retraining capabilities.

ManagementConsensusService: Facilitates the establishment of a consensus on regulatory requirements, toxic combinations, and data masking rules using a proof of ownership method, resolving conflicting claims and configuring the security access model accordingly.

10 AccessControlService: Provides services for managing user access control, role assignments, and security configurations based on the unified security access model, role encoding, and user association schema.

Interfaces:

15 RoleIdentifierInterface: Defines the methods and parameters for identifying and extracting relevant roles and secure features from the input data, using NLP and machine learning techniques.

RoleEncoderInterface: Specifies the methods and parameters for assigning unique numerical codes to the identified roles and features and modeling their relationships using mathematical operations and data structures.

20 AccessClassifierInterface: Describes the methods and parameters for cataloguing user access to roles and user associations to roles using supervised learning classification models, including model training and retraining capabilities.

25 ManagementConsensusInterface: Provides methods and parameters for establishing a consensus on regulatory requirements, toxic combinations, and data masking rules using a proof of ownership method, resolving conflicting claims and configuring the security access model accordingly.

AccessControllInterface: This interface defines the methods and parameters for managing user access control, role assignments, and security configurations based on the unified security access model, role encoding, and user association schema.

30 SENTINEL

SENTINEL (Secure, Efficient, and Intelligent Access Control System) employs advanced techniques to manage and apply security access rules efficiently. It enables granular control

over user access to data based on roles, hierarchies, and regulatory or organizational policies.

Position-Linked Role-Based Data Access:

The Intelligent Search Algorithms maintain a database of security access rules, where each user's access privileges are linked to their respective roles and positions within the organization. The process involves the following steps:

Role and Position Mapping: The system maps each user's role and position within the organizational hierarchy to corresponding security access rules.

Rule Extraction: When a user attempts to access data, the Intelligent Search Algorithms extract the relevant security access rules based on the user's role and position.

Rule Application: The extracted rules are applied to determine the user's access privileges, granting or denying access to specific data elements or hierarchies based on the defined rules.

Rule Updates: As roles, positions, or security policies change, the corresponding security access rules are automatically updated and propagated throughout the system, ensuring real-time access control.

The time complexity of the position-linked role-based data access process depends on the number of users, roles, and security access rules and the complexity of the rule extraction and application algorithms. The space complexity is determined by the size of the security access rule database and the associated user and role metadata.

Hierarchy Tree Security and Inheritance:

The Intelligent Search Algorithms support granular security access control based on hierarchical data structures, such as geographic hierarchies (e.g., region, country, state, city). The hierarchy tree security and inheritance process involve the following steps:

Hierarchy Traversal: The system traverses the hierarchical data structure, identifying the relevant elements and their parent-child relationships.

Access Rule Application: Security access rules are applied at each level of the hierarchy, granting or denying access to specific elements based on the defined rules.

Inheritance Propagation: If access to an element is granted, the Intelligent Search Algorithms automatically grant access to all descendant elements within the hierarchy. Conversely, if access to a component is denied, all descendants are denied.

Level Restriction: The system supports restricting access to specific levels of the hierarchy, preventing users from accessing granular data (e.g., city-level) or highly summarized data (e.g., regional aggregations) based on their access privileges.

5 The time complexity of the hierarchy tree security and inheritance process depends on the depth and complexity of the hierarchical data structure, as well as the number of security access rules to be applied. Space complexity is determined by the size of the hierarchical data and the metadata associated with the security access rule.

Regulatory and Toxic Combination Rules:

10 The Intelligent Search Algorithms incorporate advanced rules to handle complex regulatory requirements and organizational policies, including toxic combinations of data access that may pose risks or violate compliance standards. The regulatory and toxic combination rules process involves the following steps:

15 Rule Definition: Regulatory requirements, toxic combinations, and other complex access rules are defined and stored within the system, often through a consensus-based approach involving stakeholders and domain experts.

Rule Parsing: The system parses and interprets the defined rules, translating them into executable logic or constraints that can be applied during data access attempts.

20 Rule Evaluation: When a user attempts to access data, the Intelligent Search Algorithms evaluate the relevant regulatory and toxic combination rules, considering the user's role, the requested data elements, and other contextual factors.

Access Decision: Based on the rule evaluation, the system grants or denies access to the requested data elements, ensuring compliance with regulatory requirements and organizational policies.

25 Rule Updates: As regulatory requirements or organizational policies evolve, the corresponding rules are updated and propagated throughout the system, enabling real-time adaptation of the access control mechanisms.

30 The time complexity of the regulatory and toxic combination rules process depends on the number and complexity of the defined rules and the complexity of the rule parsing and evaluation algorithms. The space complexity is determined by the size of the rule database and the associated metadata. By employing these intelligent search algorithms, the security access control model ensures granular control over user access to data, supporting position-linked role-based access, hierarchy tree security and inheritance, and enforcing regulatory

and toxic combination rules while maintaining efficient real-time access control and seamless adaptation to policy changes.

Business Layer

Processes:

- 5 User and Role Management: This process involves managing user profiles, roles, and positions within the organization, ensuring accurate mapping of access privileges based on each user's role and position.

- 10 Security Rule Management: This process focuses on defining, maintaining, and updating security rules, including role-based access rules, hierarchy tree security rules, and regulatory or toxic combination rules.

Data Access Control: This process involves enforcing the defined security access rules during user attempts to access data, granting or denying access based on the user's role, position, and the applicable laws.

- 15 Rule Adaptation and Compliance: This process ensures the continuous adaptation of security access rules to align with evolving regulatory requirements, organizational policies, and changes in user roles or positions, maintaining compliance and mitigating risks.

Functions:

- 20 Granular Access Control: The Intelligent Search Algorithms enable granular control over user access to data, supporting position-linked role-based access, hierarchy tree security and inheritance, and enforcing regulatory and toxic combination rules.

Efficient Real-Time Access: The system provides efficient real-time access control by maintaining a database of security access rules and employing intelligent search algorithms. This ensures seamless user experiences and minimizes performance impacts.

- 25 Regulatory Compliance and Risk Mitigation: Incorporating regulatory and toxic combination rules ensures compliance with applicable regulations and organizational policies, mitigating the risks associated with unauthorized data access or violations.

Adaptability and Scalability: The Intelligent Search Algorithms support continuous adaptation to changes in user roles, positions, regulatory requirements, and organizational policies, enabling scalability and future-proofing of the security access control model.

- 30 Application Layer

The Intelligent Search Algorithms for Security Access Control comprise the following key components:

UserProfileManager: This component manages user profiles, roles, and positions within the organization, ensuring accurate mapping of access privileges based on each user's role and position.

SecurityRuleManager: This component is responsible for defining, maintaining, and updating security rules, including role-based access rules, hierarchy tree security rules, and regulatory or toxic combination rules.

AccessControlEngine: This component enforces the defined security access rules during user attempts to access data, granting or denying access based on the user's role, position, and the applicable laws.

RuleAdaptationMonitor: This component monitors changes in regulatory requirements, organizational policies, and user roles or positions, ensuring the continuous adaptation and compliance of security access rules.

Services:

UserProfileManagementService: Provides methods for managing user profiles, roles, and positions within the organization, ensuring accurate mapping of access privileges based on each user's role and position.

SecurityRuleManagementService: Offers services for defining, maintaining, and updating security access rules, including role-based access rules, hierarchy tree security rules, and regulatory or toxic combination rules.

AccessControlService: Enables the enforcement of defined security access rules during user attempts to access data, granting or denying access based on the user's role, position, and the applicable rules.

RuleAdaptationMonitoringService: Facilitates the monitoring of changes in regulatory requirements, organizational policies, and user roles or positions, ensuring the continuous adaptation and compliance of security access rules.

Interfaces:

UserProfileManagerInterface: This interface defines the methods and parameters for managing user profiles, roles, and positions within the organization. It ensures accurate mapping of access privileges based on each user's role and position.

SecurityRuleManagerInterface: Specifies the methods and parameters for defining, maintaining, and updating security rules, including role-based access rules, hierarchy tree security rules, and regulatory or toxic combination rules.

AccessControlEngineInterface: Describes the methods and parameters for enforcing defined security access rules during user attempts to access data, granting or denying access based on the user's role, position, and the applicable laws.

RuleAdaptationMonitorInterface provides methods and parameters for monitoring changes in regulatory requirements, organizational policies, and user roles or positions. It ensures continuous adaptation and compliance and enables reliable and efficient reporting across analytical and decision-making processes.

Cryptographic Audit Trail (CATM)

The Cryptographic Audit Trail Module (CATM) is an advanced, agent-based system that ensures data integrity, provenance tracking, and transparency within the GenFoundry system. At its core, CATM employs a synergistic ensemble of three specialized agents collaborating through continuous optimization cycles, overseen by a distributed ledger orchestration model.

Agents

The CATM agent architecture comprises three key agents, each with specific roles and responsibilities, working together to ensure data integrity, provenance tracking, and transparency:

ATLAS (Audit Trail Lifecycle and Security): ATLAS generates immutable audit trails that cryptographically capture the complete lifecycle of data elements from their originating sources through various transformations, encryption states, access events, and analytical consumption. It employs advanced cryptographic techniques to track and record the provenance of each data element, ensuring integrity and non-repudiation.

QUASAR (Quantum Verification and Security Assurance): QUASAR leverages quantum computing techniques, such as quantum parallelism and quantum algorithms, to optimize provenance verification at scale. It also employs quantum-resilient cryptographic algorithms and post-quantum digital signatures to ensure audit trails' long-term security and integrity.

NEXUS (Network Integration and Multi-Party Governance): NEXUS integrates with distributed ledger technologies to enhance transparency and enable multi-party governance of audit trails. It facilitates the seamless integration of the CATM system with various

distributed ledger networks, ensuring efficient consensus and resistance to known distributed ledger vulnerabilities.

The ATLAS, QUASAR, and NEXUS agents are orchestrated using a distributed ledger orchestration model, which ensures a coordinated and efficient execution of the audit trail management tasks. The model captures the dependencies and interactions between the agents, ensuring that each task is performed in the appropriate sequence and with the necessary inputs and outputs.

Evaluation & Validation

The selection of ground truth data and evaluation metrics is carefully tailored to align with the specific functionalities and objectives of each CATM component:

ATLAS: ATLAS's performance is evaluated using simulated data lifecycles, known data tampering techniques, and industry-standard audit trail benchmarks. Metrics such as audit trail integrity, data lifecycle completeness, and resistance to tampering are employed to assess ATLAS's effectiveness.

QUASAR: QUASAR's performance is assessed using simulated quantum computing environments, known quantum optimization techniques, and quantum algorithm benchmarks. Metrics such as verification speedup, scaling efficiency, and resistance to known quantum computing vulnerabilities are employed to evaluate QUASAR's effectiveness in optimizing provenance verification at scale.

NEXUS: NEXUS is evaluated using simulated distributed ledger networks, known consensus protocols, and distributed ledger benchmarks. Metrics like network transparency, consensus efficiency, and resistance to known distributed ledger vulnerabilities are used to assess NEXUS's ability to integrate with distributed ledger technologies for multi-party governance.

Using these three key agents and evaluation methods, CATM ensures secure, transparent, and trustworthy audit trail management solutions for the GenFoundry system, enabling data integrity, provenance tracking, and multi-party governance.

ATLAS

ATLAS (Audit Trail Lifecycle and Security) is an advanced agent that generates immutable audit trails. It captures the complete lifecycle of data elements from their originating sources through various transformations, encryption states, access events, and analytical consumption. It employs advanced cryptographic techniques to track and record the provenance of each data element, ensuring integrity and non-repudiation.

At its core, ATLAS employs a synergistic ensemble of specialized agents collaborating through continuous optimization cycles, overseen by a cryptographic orchestration model. The ATLAS agents, including SourceTracker, TransformationRecorder, EncryptionMonitor, AccessLogger, and ConsumptionAuditor, work in concert to capture, record, and secure the audit trails throughout the data lifecycle.

ATLAS's source tracking capabilities leverage data lineage and provenance capture techniques to identify and record the originating sources of data elements, including the timestamps, source systems, and data owners. This information is securely stored in the audit trails, providing a verifiable record of the data's origin and ownership.

Furthermore, ATLAS incorporates transformation recording mechanisms that capture the details of any transformations, modifications, or derivations applied to the data elements as they move through the data pipeline. Each transformation event is cryptographically signed and recorded in the audit trails, maintaining a tamper-evident history of the data's evolution.

Underpinning ATLAS's encryption monitoring capabilities is its integration with the Quantum Security and Encryption Module (QSEM) of the GenFoundry system. ATLAS tracks the encryption states of data elements, recording the encryption algorithms, keys, and timestamps associated with each encryption or decryption event. This ensures a complete and auditable record of the data's encryption lifecycle.

With its access logging and consumption auditing features, ATLAS captures and records all access events and analytical consumption of data elements. Each access event, including the user identity, timestamp, purpose, and access type, is cryptographically signed and stored in the audit trails. This provides a comprehensive and non-repudiable record of who accessed the data, when, and for what purpose.

ATLAS's architecture aligns with immutability, non-repudiation, and cryptographic security principles. Its robust validation mechanisms, which include cryptographic integrity checks, consistency verification, and reconciliation with other audit trail sources, ensure the reliability and trustworthiness of the generated audit trails.

Methods

ATLAS employs advanced cryptographic techniques, data lineage and provenance capture, and secure logging mechanisms to generate immutable and tamper-evident audit trails throughout the data lifecycle.

Source Tracking and Provenance Capture: ATLAS captures the originating sources and provenance of data elements, recording the necessary details in the audit trails. The source tracking and provenance capture process involves the following steps:

Source Identification: ATLAS identifies the originating sources of data elements, such as source systems, databases, or data providers, using data lineage and provenance capture techniques.

Metadata Extraction: The SourceTracker agent extracts relevant metadata associated with the data elements, such as timestamps, data owners, and source system details.

Cryptographic Signing: ATLAS cryptographically signs the captured source and provenance metadata using digital signatures or hash functions to ensure integrity and non-repudiation.

Audit Trail Recording: The signed source and provenance metadata are securely recorded in the audit trails, providing a verifiable record of the data's origin and ownership.

Provenance Linking: ATLAS establishes cryptographic links between the data elements and their provenance records in the audit trails, enabling efficient traceability and verification.

The time complexity of the source tracking and provenance capture process depends on the volume of data elements, the complexity of the data lineage and provenance capture techniques, and the efficiency of the cryptographic signing and recording mechanisms. The space complexity is determined by the size of the provenance metadata and the cryptographic signatures stored in the audit trails.

Transformation Recording: ATLAS captures and records the details of any transformations, modifications, or derivations applied to the data elements as they move through the data pipeline. The transformation recording process involves the following steps:

Transformation Detection: ATLAS monitors the data pipeline to detect any transformation events applied to the data elements, such as data cleansing, normalization, or aggregation.

Transformation Metadata Capture: The TransformationRecorder agent captures the metadata associated with each transformation event, including the transformation type, parameters, and timestamps.

Cryptographic Signing: ATLAS cryptographically signs the captured transformation metadata using digital signatures or hash functions to ensure integrity and non-repudiation.

Audit Trail Recording: The signed transformation metadata is securely recorded in the audit trails, maintaining a tamper-evident history of the data's evolution.

Transformation Linking: ATLAS establishes cryptographic links between the transformed data elements and their corresponding transformation records in the audit trails, enabling efficient traceability and verification.

5 The time complexity of the transformation recording process depends on the volume of transformation events, the complexity of the transformation detection and metadata capture techniques, and the efficiency of the cryptographic signing and recording mechanisms. The space complexity is determined by the size of the transformation metadata and the cryptographic signatures stored in the audit trails.

10 Encryption Monitoring: ATLAS tracks the encryption states of data elements, recording the encryption algorithms, keys, and timestamps associated with each encryption or decryption event. The encryption monitoring process involves the following steps:

Encryption Event Detection: ATLAS integrates with the Quantum Security and Encryption Module (QSEM) to detect encryption and decryption events applied to the data elements.

15 Encryption Metadata Capture: The EncryptionMonitor agent captures the metadata associated with each encryption event, including the encryption algorithm, key identifiers, and timestamps.

Cryptographic Signing: ATLAS cryptographically signs the captured encryption metadata using digital signatures or hash functions to ensure integrity and non-repudiation.

20 Audit Trail Recording: The signed encryption metadata is securely recorded in the audit trails, maintaining a complete and auditable record of the data's encryption lifecycle.

Encryption State Linking: ATLAS establishes cryptographic links between the encrypted data elements and their corresponding encryption records in the audit trails, enabling efficient traceability and verification of the data's encryption state.

25 The time complexity of the encryption monitoring process depends on the volume of encryption events, the efficiency of the encryption event detection and metadata capture mechanisms, and the performance of the cryptographic signing and recording operations. The size of the encryption metadata and the cryptographic signatures stored in the audit trails determine the space complexity.

30 Access Logging and Consumption Auditing: ATLAS captures and records all access events and analytical consumption of data elements, providing a comprehensive and non-repudiable data access and usage record. The access logging and consumption auditing process involves the following steps:

Access Event Detection: ATLAS monitors the data access points and analytical interfaces to detect any access or consumption events related to the data elements.

Access Metadata Capture: The AccessLogger agent captures the relevant metadata associated with each access event, including the user identity, timestamp, purpose, and access type (e.g., read, write, or execute).

Cryptographic Signing: ATLAS cryptographically signs the captured access metadata using digital signatures or hash functions to ensure integrity and non-repudiation.

Audit Trail Recording: The signed access metadata is securely recorded in the audit trails, maintaining a comprehensive and tamper-evident data access and consumption record.

Access Linking: ATLAS establishes cryptographic links between the accessed data elements and their corresponding access records in the audit trails, enabling efficient traceability and verification of data access and usage patterns.

The time complexity of the access logging and consumption auditing process depends on the volume of access events, the efficiency of the access event detection and metadata capture mechanisms, and the performance of the cryptographic signing and recording operations. The space complexity is determined by the size of the access metadata and the cryptographic signatures stored in the audit trails.

Business Layer

Processes:

Data Lineage and Provenance Tracking: This process involves identifying the originating sources of data elements, extracting relevant metadata, and capturing the lineage and provenance information in the audit trails.

Data Transformation Auditing: This process focuses on detecting and recording any transformations, modifications, or derivations applied to the data elements as they move through the data pipeline, maintaining a tamper-evident history of the data's evolution.

Encryption Lifecycle Monitoring: This process involves tracking the encryption states of data elements, recording the encryption algorithms, keys, and timestamps associated with each encryption or decryption event, and maintaining a complete and auditable record of the data's encryption lifecycle.

Access and Consumption Logging: This process focuses on capturing and recording all access events and analytical consumption of data elements, providing a comprehensive and non-repudiable record of data access and usage patterns.

Functions:

Immutable Audit Trails: ATLAS generates immutable and tamper-evident audit trails that capture the complete lifecycle of data elements, ensuring the integrity and non-repudiation of the recorded events and metadata.

- 5 Cryptographic Security: ATLAS employs advanced cryptographic techniques, such as digital signatures and hash functions, to sign and verify the integrity of the recorded audit trail entries, protecting against tampering and ensuring the trustworthiness of the audit records.

- 10 Data Provenance and Lineage Tracking: ATLAS enables the tracking and verification of data provenance and lineage by capturing the originating sources, transformations, and derivations of data elements, providing a complete and auditable history of the data's lifecycle.

- 15 Comprehensive Access and Consumption Auditing: ATLAS provides comprehensive auditing of data access and consumption patterns by logging all access events, user identities, timestamps, purposes, and access types, enabling efficient traceability and verification of data usage.

Application Layer

The ATLAS system comprises the following key components:

- SourceTracker: This component identifies the originating sources of data elements, extracts relevant metadata, and records the source and provenance information in the audit trails.
- 20 TransformationRecorder: This component detects and records any transformations, modifications, or derivations applied to the data elements, capturing the relevant metadata and maintaining a tamper-evident history of the data's evolution.
- EncryptionMonitor: This component tracks the encryption states of data elements, recording the encryption algorithms, keys, and timestamps associated with each encryption or decryption event and maintaining a complete and auditable record of the data's encryption lifecycle.
- 25 AccessLogger: This component captures and records all access events and analytical consumption of data elements, logging the user identities, timestamps, purposes, and access types and providing a comprehensive and non-repudiable record of data access and usage.
- 30

ConsumptionAuditor: This component analyzes the recorded access and consumption logs to identify patterns, anomalies, or potential security breaches and generates audit reports and alerts for further investigation and remediation.

Services:

- 5 SourceTrackingService: Provides methods for identifying the originating sources of data elements, extracting relevant metadata, and recording the source and provenance information in the audit trails.

- TransformationRecordingService: Offers services for detecting and recording transformations, modifications, or derivations applied to the data elements, capturing the
10 relevant metadata, and maintaining a tamper-evident history of the data's evolution.

EncryptionMonitoringService: This service tracks the encryption states of data elements, records the encryption algorithms, keys, and timestamps associated with each encryption or decryption event, and maintains a complete and auditable record of the data's encryption lifecycle.

- 15 AccessLoggingService: Facilitates the capturing and recording all access events and analytical consumption of data elements, logging the user identities, timestamps, purposes, and access types and providing a comprehensive and non-repudiable record of data access and usage.

- ConsumptionAuditingService: Provides services for analyzing the recorded access and
20 consumption logs to identify patterns, anomalies, or potential security breaches and generating audit reports and alerts for further investigation and remediation.

Interfaces:

- SourceTrackerInterface: This interface defines the methods and parameters for identifying the originating sources of data elements, extracting relevant metadata, and recording the
25 source and provenance information in the audit trails.

TransformationRecorderInterface: Specifies the methods and input/output formats for detecting and recording transformations, modifications, or derivations applied to the data elements, capturing the relevant metadata and maintaining a tamper-evident history of the data's evolution.

- 30 EncryptionMonitorInterface: Describes the methods and parameters for tracking the encryption states of data elements, recording the encryption algorithms, keys, and

timestamps associated with each encryption or decryption event, and maintaining a complete and auditable record of the data's encryption lifecycle.

AccessLoggerInterface: Defines the methods and output formats for capturing and recording all access events and analytical consumption of data elements, logging the user identities, timestamps, purposes, and access types, and providing a comprehensive and non-repudiable record of data access and usage.

ConsumptionAuditorInterface: Specifies the methods and input/output formats for analyzing the recorded access and consumption logs to identify patterns, anomalies, or potential security breaches and generating audit reports and alerts for further investigation and remediation.

NODM

The Neural Orchestration DevOps Module (NODM) is an advanced, agent-based method that provides a cloud-native architecture for deploying and orchestrating the various GenFoundry modules across hybrid, multi-cloud environments. At its core, NODM employs a synergistic ensemble of three specialized agents collaborating through continuous optimization cycles, overseen by a neural network-driven DevOps orchestration model.

Agents

The NODM agent architecture comprises three key agents, each with specific roles and responsibilities, working together to ensure efficient deployment, orchestration, and continuous delivery of the GenFoundry modules:

HYDRA (Hybrid Deployment and Resilience Agent): HYDRA manages the deployment of GenFoundry modules across multiple cloud platforms and on-premises environments, leveraging cloud-native technologies and infrastructure-as-code principles. It ensures resilient and efficient resource utilization, scalability, and fault isolation across the hybrid cloud infrastructure.

NEMO (Neural Workflow and Microservices Orchestrator): NEMO employs large language models (LLMs) and neural networks to generate and optimize Petri-net-based workflows for orchestrating the interactions and data flows between GenFoundry modules. It also leverages containerization technologies like Docker and Kubernetes to package and deploy GenFoundry modules as lightweight, scalable, and portable microservices.

OASIS (Operations Automation and System Insights Supervisor): OASIS incorporates DevOps principles and automation toolchains to streamline the continuous delivery of updates and extensions to the GenFoundry modules, automating the build, testing, and

deployment processes. It also monitors the performance, health, and resource utilization of deployed GenFoundry modules, providing feedback to NEMO for continuous optimization and adaptation of the Petri-net choreography workflows.

The HYDRA, NEMO, and OASIS agents are orchestrated using a neural network-driven DevOps orchestration model. This model ensures coordinated and efficient execution of deployment, orchestration, and continuous delivery tasks. It captures the dependencies and interactions between the agents, ensuring that each task is performed in the appropriate sequence and with the necessary inputs and outputs.

Evaluation & Validation

The selection of ground truth data and evaluation metrics is carefully tailored to align with the specific functionalities and objectives of each NODM component:

HYDRA: HYDRA's performance is evaluated using simulated cloud deployment scenarios, known deployment challenges, and industry-standard cloud benchmarks. Metrics such as deployment success rate, resource utilization efficiency, and compliance with cloud provider best practices are employed to assess HYDRA's effectiveness.

NEMO: NEMO's evaluation relies on simulated workflow scenarios, known orchestration challenges, and Petri-net benchmarks. Metrics like workflow optimization efficiency, data flow coordination accuracy, and scalability are used to measure NEMO's ability to generate and optimize Petri-net-based workflows for efficient module orchestration.

OASIS: OASIS's performance is assessed using simulated continuous delivery pipelines, known DevOps challenges, and industry-standard DevOps benchmarks. Metrics such as pipeline automation efficiency, deployment success rate, and compliance with DevOps best practices are employed to evaluate OASIS's effectiveness in streamlining the continuous delivery of updates and extensions.

By employing these three key agents and evaluation methods, NODM ensures efficient deployment, orchestration, and continuous delivery solutions for the GenFoundry system, enabling seamless integration, coordination, and scalability across hybrid, multi-cloud environments.

HYDRA

HYDRA (Hybrid Deployment and Resilience Agent) is an advanced agent that manages the deployment of GenFoundry modules across multiple cloud platforms and on-premises environments, leveraging cloud-native technologies and infrastructure-as-code principles. It

ensures resilient and efficient resource utilization, scalability, and fault isolation across the hybrid cloud infrastructure.

At its core, HYDRA employs a synergistic ensemble of specialized agents collaborating through continuous optimization cycles, overseen by a hybrid deployment orchestration model. The HYDRA agents, including EnvironmentProvisioner, ConfigurationManager, DeploymentExecutor, ResilienceMonitor, and OptimizationPlanner, work in concert to provision, configure, deploy, and manage GenFoundry modules across diverse computing environments.

HYDRA's environment provisioning capabilities leverage infrastructure-as-code tools like Terraform or AWS CloudFormation to automatically provision and configure the necessary cloud resources, such as virtual machines, networks, and storage, based on the requirements of GenFoundry modules. This ensures consistent and reproducible deployment environments across different cloud platforms.

Furthermore, HYDRA incorporates a configuration management system that uses tools like Ansible or Puppet to automatically install, configure, and maintain the software dependencies and settings required by GenFoundry modules. This enables consistent and auditable configuration management across the hybrid cloud infrastructure.

Underpinning HYDRA's deployment execution capabilities is its use of containerization and orchestration technologies like Docker and Kubernetes. HYDRA integrates with NEMO to deploy the containerized GenFoundry modules across the provisioned cloud environments, ensuring efficient resource utilization, scalability, and fault isolation.

With its modular architecture and well-defined interfaces, HYDRA integrates with the other GenFoundry components, such as NEMO, OASIS, and the underlying cloud platforms. It consumes the deployment configurations, resource requirements, and performance insights generated by these components to inform the provisioning, configuration, and deployment processes.

HYDRA's architecture aligns with cloud-native design principles, promoting infrastructure automation, immutable infrastructure, and declarative configuration management. Its robust validation mechanisms, which include infrastructure testing, compliance checks, and chaos engineering techniques, ensure the reliability, security, and resilience of the deployed GenFoundry modules across the hybrid cloud infrastructure.

Methods

HYDRA employs advanced provisioning, configuration management, deployment execution, and resilience monitoring techniques to ensure efficient, scalable, and resilient deployment of GenFoundry modules across hybrid cloud environments.

Environment Provisioning: HYDRA provides the necessary cloud resources and environments to deploy GenFoundry modules using infrastructure-as-code (IaC) principles. The environment provisioning process involves the following steps:

Infrastructure Definition: HYDRA uses IaC tools like Terraform or AWS CloudFormation to define the desired state of the cloud infrastructure, including virtual machines, networks, storage, and security configurations.

Resource Provisioning: The EnvironmentProvisioner agent executes the IaC definitions to automatically provision the required cloud resources across different cloud platforms, such as AWS, Azure, or Google Cloud.

Environment Validation: HYDRA validates the provisioned environments using infrastructure testing tools to ensure they meet the specified requirements and are ready to deploy GenFoundry modules.

Infrastructure Monitoring: HYDRA continuously monitors the provisioned cloud resources to ensure their availability, performance, and compliance with defined policies and standards.

Infrastructure Optimization: Based on the monitoring data and feedback from other components like OASIS, HYDRA optimizes the provisioned infrastructure by adjusting resource allocations, scaling configurations, and applying cost optimization strategies.

The time complexity of the environment provisioning process depends on the number and complexity of the cloud resources being provisioned and the efficiency of the IaC tools and provisioning algorithms used. The size of the IaC definitions and the infrastructure metadata stored by HYDRA determines the space complexity.

Configuration Management: HYDRA configures the provisioned environments and GenFoundry modules using configuration management tools and practices. The configuration management process involves the following steps:

Configuration Definition: HYDRA uses configuration management tools like Ansible or Puppet to define the desired state of the software configurations, including dependencies, settings, and security policies.

Configuration Provisioning: The ConfigurationManager agent applies the defined configurations to the provisioned environments, installing and configuring the software components and libraries required by GenFoundry modules.

Configuration Testing: HYDRA tests the applied configurations using automated testing tools to ensure they meet the specified requirements and do not introduce any security vulnerabilities or compatibility issues.

Configuration Monitoring: HYDRA continuously monitors the configurations of the deployed GenFoundry modules to detect any drift or deviations from the desired state.

Configuration Remediation: If any configuration issues or drifts are detected, HYDRA automatically remediates them by applying the necessary configuration changes or rolling back to a previous stable state.

The time complexity of the configuration management process depends on the number and complexity of the software configurations being managed and the efficiency of the configuration management tools and algorithms used. The space complexity is determined by the size of the configuration definitions and the metadata stored by HYDRA.

Deployment Execution: HYDRA integrates with NEMO to deploy containerized GenFoundry modules across the provisioned cloud environments. The deployment execution process involves the following steps:

Deployment Planning: HYDRA collaborates with NEMO to plan the deployment of GenFoundry modules, considering factors such as resource requirements, dependencies, and placement constraints.

Container Orchestration: The DeploymentExecutor agent integrates with container orchestration platforms like Kubernetes to deploy and manage the containerized GenFoundry modules across the provisioned cloud environments.

Service Discovery and Networking: HYDRA configures the necessary service discovery and networking mechanisms to enable communication and collaboration between the deployed GenFoundry modules.

Deployment Monitoring: HYDRA monitors the health, performance, and resource utilization of the deployed GenFoundry modules, collecting metrics and logs for analysis and optimization.

Deployment Optimization: Based on the monitoring data and feedback from other components like OASIS, HYDRA optimizes the deployments by adjusting resource allocations, scaling configurations, and applying performance optimization techniques.

5 The time complexity of the deployment execution process depends on the number and size of the containerized GenFoundry modules being deployed and the efficiency of the container orchestration and deployment algorithms used. The space complexity is determined by the size of the container images and the deployment metadata stored by HYDRA.

Resilience Monitoring: HYDRA ensures the resilience and fault tolerance of the deployed GenFoundry modules by continuously monitoring their health, performance, and availability.

10 The resilience monitoring process involves the following steps:

Health Monitoring: The ResilienceMonitor agent continuously monitors the health and availability of the deployed GenFoundry modules, using techniques like heartbeat checks, liveness probes, and synthetic transactions.

15 Performance Monitoring: HYDRA collects performance metrics and logs from the deployed GenFoundry modules, analyzing them to detect performance bottlenecks, resource contentions, or anomalies.

Fault Detection: HYDRA uses anomaly detection and machine learning algorithms to identify potential faults, failures, or security breaches in the deployed GenFoundry modules.

20 Self-Healing and Recovery: If faults or failures are detected, HYDRA automatically triggers self-healing and recovery mechanisms, such as restarting failed containers, migrating workloads to healthy nodes, or scaling up resources to handle increased load.

Chaos Engineering: HYDRA incorporates chaos engineering practices to proactively test the resilience of the deployed GenFoundry modules by introducing controlled failures and disruptions and observing how the system responds and recovers.

25 The time complexity of the resilience monitoring process depends on the number of deployed GenFoundry modules, the frequency of monitoring checks, and the efficiency of the monitoring and anomaly detection algorithms used. The space complexity is determined by the size of the monitoring data and the metadata stored by HYDRA.

Business Layer

30 Processes:

Cloud Infrastructure Provisioning: This process involves defining and provisioning the necessary cloud resources and environments for deploying GenFoundry modules using infrastructure-as-code (IaC) principles and tools.

5 Configuration Management and Compliance: This process focuses on configuring the provisioned environments and GenFoundry modules using configuration management tools and practices, ensuring compliance with defined policies and standards.

Deployment Planning and Execution: This process involves collaborating with NEMO to plan and execute the deployment of containerized GenFoundry modules across the provisioned cloud environments, considering resource requirements, dependencies, and placement
10 constraints.

Resilience Monitoring and Optimization: This process focuses on continuously monitoring the health, performance, and availability of the deployed GenFoundry modules, detecting faults, and triggering self-healing and optimization mechanisms to ensure resilience and fault tolerance.

15 Functions:

Automated Cloud Provisioning: HYDRA enables the automated provisioning of cloud resources and environments using IaC principles, reducing manual effort, ensuring consistency, and allowing scalability across different cloud platforms.

Consistent Configuration Management: HYDRA provides consistent and auditable
20 configuration management across the hybrid cloud infrastructure, using tools like Ansible or Puppet to install, configure, and maintain the software dependencies and settings required by GenFoundry modules.

Seamless Deployment Orchestration: HYDRA integrates with NEMO and container orchestration platforms like Kubernetes to seamlessly deploy and manage containerized
25 GenFoundry modules across the provisioned cloud environments, ensuring efficient resource utilization and fault isolation.

Proactive Resilience Assurance: HYDRA ensures the resilience and fault tolerance of the deployed GenFoundry modules by continuously monitoring their health, performance, and availability, detecting faults, and triggering self-healing and recovery mechanisms to
30 minimize downtime and service disruptions.

Application Layer

The HYDRA system comprises the following key components:

EnvironmentProvisioner: This component uses IaC tools like Terraform or AWS CloudFormation to automatically provision the required cloud resources and environments based on the defined infrastructure specifications.

5 ConfigurationManager: This component manages the configuration of the provisioned environments and GenFoundry modules using configuration management tools like Ansible or Puppet, ensuring consistent and compliant software configurations.

DeploymentExecutor: This component integrates with NEMO and container orchestration platforms like Kubernetes to execute the deployment of containerized GenFoundry modules across the provisioned cloud environments.

10 ResilienceMonitor: This component continuously monitors the health, performance, and availability of the deployed GenFoundry modules, detecting faults and triggering self-healing and recovery mechanisms to ensure resilience and fault tolerance.

15 OptimizationPlanner: This component analyzes the monitoring data and feedback from other components like OASIS to optimize the provisioned infrastructure and deployments, applying resource optimization, scaling, and cost-efficiency strategies.

Services:

EnvironmentProvisioningService: Provides methods for automatically provisioning cloud resources and environments based on the defined infrastructure specifications using IaC tools.

20 ConfigurationManagementService: Offers services for configuring provisioned environments and GenFoundry modules using configuration management tools, ensuring consistent and compliant software configurations.

25 DeploymentExecutionService: Enables the integration with NEMO and container orchestration platforms to execute the deployment of containerized GenFoundry modules across the provisioned cloud environments.

ResilienceMonitoringService: Facilitates the continuous monitoring of the health, performance, and availability of deployed GenFoundry modules, detecting faults, and triggering self-healing and recovery mechanisms.

30 OptimizationPlanningService: Provides services for analyzing monitoring data and feedback to optimize the provisioned infrastructure and deployments, applying resource optimization, scaling, and cost-efficiency strategies.

Interfaces:

EnvironmentProvisionerInterface: Defines the methods and parameters for automatically provisioning cloud resources and environments based on the defined infrastructure specifications using IaC tools.

5 ConfigurationManagerInterface: Specifies the methods and input/output formats for managing the configuration of provisioned environments and GenFoundry modules using configuration management tools.

DeploymentExecutorInterface: Describes the methods and parameters for integrating with NEMO and container orchestration platforms to execute the deployment of containerized GenFoundry modules across the provisioned cloud environments.

10 ResilienceMonitorInterface: Defines the methods and output formats for continuously monitoring the health, performance, and availability of deployed GenFoundry modules, detecting faults, and triggering self-healing and recovery mechanisms.

OptimizationPlannerInterface: Specifies the methods and input/output formats for analyzing monitoring data and feedback to optimize the provisioned infrastructure and deployments,
15 applying resource optimization, scaling, and cost-efficiency strategies.

NEMO

NEMO (Neural Workflow and Microservices Orchestrator) is an advanced agent that employs large language models (LLMs) and neural networks to generate and optimize Petri-net-based workflows for orchestrating the interactions and data flows between GenFoundry
20 modules. It also leverages containerization technologies like Docker and Kubernetes to package and deploy GenFoundry modules as lightweight, scalable, and portable microservices.

At its core, NEMO employs a synergistic ensemble of specialized agents collaborating through continuous optimization cycles, overseen by a Petri-net orchestration model. The
25 NEMO agents, including WorkflowGenerator, PetriNetOptimizer, MicroserviceComposer, DeploymentOrchestrator, and AdaptationManager, work in concert to generate, optimize, and deploy Petri-net workflows and microservices for GenFoundry modules.

NEMO's workflow generation capabilities leverage LLMs and neural networks to automatically create Petri-net models based on the requirements and dependencies of
30 GenFoundry modules. These models capture the complex interactions, data flows, and control logic between the modules, enabling efficient coordination and execution.

Furthermore, NEMO incorporates a continuous optimization mechanism that analyzes the runtime performance, resource utilization, and feedback from OASIS to identify improvement

opportunities in the Petri-net workflows. Using advanced optimization algorithms and reinforcement learning techniques, NEMO refines the workflows to enhance efficiency, scalability, and resilience.

Underpinning NEMO's microservice orchestration capabilities is its use of containerization technologies like Docker and Kubernetes. NEMO automatically packages GenFoundry modules into lightweight, self-contained containers, enabling flexible deployment, scaling, and management across diverse computing environments.

With its modular architecture and well-defined interfaces, NEMO integrates with the other GenFoundry components, such as MEGAN, SEPHYR, GAMA, and OASIS. It consumes the metadata, data patterns, access control models, and performance insights generated by these components to inform the workflow generation and optimization processes.

NEMO's architecture aligns with modern distributed systems and microservices design principles, promoting loose coupling, high cohesion, and fault isolation. Its robust validation mechanisms, which include model checking, simulation, and chaos engineering techniques, ensure the correctness, reliability, and resilience of the generated Petri-net workflows and microservice deployments.

Methods

NEMO employs advanced workflow generation, optimization, and microservice orchestration techniques to create efficient, scalable, and adaptable Petri-net models and containerized deployments for GenFoundry modules.

Petri-Net Workflow Generation: NEMO generates Petri-net-based workflows to orchestrate the interactions and data flows between GenFoundry modules. The workflow generation process involves the following steps:

Requirements Gathering: NEMO collects requirements and specifications from the GenFoundry modules, including their functionalities, dependencies, and performance expectations.

Dependency Analysis: NEMO analyzes the modules' dependencies to identify the required interactions, data flows, and control logic.

Petri-net Modeling: The WorkflowGenerator agent employs LLMs and neural networks to automatically create Petri-net models based on the gathered requirements and dependencies. These models capture the states, transitions, and conditions governing the module interactions.

Model Validation: NEMO validates the generated Petri-net models using formal verification techniques, such as model checking and reachability analysis, to ensure their correctness and deadlock freedom.

Workflow Deployment: The validated Petri-net workflows are deployed to the GenFoundry system, where they orchestrate the execution and coordination of the modules.

The time complexity of the Petri-net workflow generation process depends on the number of modules, their interactions, and the efficiency of the LLMs and neural networks used for modeling. The space complexity is determined by the size of the Petri-net models and the infrastructure required for their storage and execution.

Workflow Optimization: NEMO continuously optimizes the Petri-net workflows based on runtime performance, resource utilization, and feedback from OASIS. The workflow optimization process involves the following steps:

Performance Monitoring: NEMO collects performance metrics and resource utilization data from the deployed Petri-net workflows in collaboration with OASIS.

Bottleneck Identification: The PetriNetOptimizer agent analyzes the collected data to identify performance bottlenecks, resource contentions, and workflow inefficiencies.

Optimization Strategies: NEMO applies various optimization strategies, such as parallelization, task reordering, resource allocation, and caching, to improve the efficiency and scalability of the workflows.

Reinforcement Learning: NEMO employs reinforcement learning techniques to continuously learn and adapt the optimization strategies based on the observed performance improvements and feedback from OASIS.

Workflow Redeployment: The optimized Petri-net workflows are redeployed to the GenFoundry system, replacing the previous versions and enabling continuous performance enhancement.

The workflow optimization process's time complexity depends on the Petri-net models' size, the performance data volume, and the efficiency of the optimization algorithms and reinforcement learning techniques. The space complexity is determined by the size of the performance data and the infrastructure required for analysis and optimization.

Microservice Orchestration: NEMO leverages containerization technologies to package and deploy GenFoundry modules as microservices, enabling flexible orchestration and scalability. The microservice orchestration process involves the following steps:

Containerization: The MicroserviceComposer agent automatically packages each GenFoundry module into a lightweight, self-contained container using technologies like Docker. These containers encapsulate the module's code, dependencies, and runtime environment.

- 5 Container Repository: The containerized modules are stored in a secure container repository, such as Docker Registry or Artifactory, for versioning and distribution.

Orchestration Configuration: NEMO defines the orchestration configuration for the containerized modules using declarative formats like Kubernetes YAML or Docker Compose. These configurations specify the microservices' deployment, scaling, networking, and
10 resource requirements.

Deployment and Scaling: The DeploymentOrchestrator agent deploys the containerized modules to a container orchestration platform, such as Kubernetes or Docker Swarm, based on the defined configurations. The platform manages the microservices' scheduling, scaling, and load balancing.

- 15 Continuous Monitoring: NEMO monitors the deployed microservices' health, performance, and resource utilization in collaboration with OASIS to ensure optimal operation and identify any issues or anomalies.

The time complexity of the microservice orchestration process depends on the number of modules, the size of the container images, and the efficiency of the containerization and
20 orchestration technologies used. The space complexity is determined by the size of the container images and the infrastructure required for their storage, deployment, and scaling.

Business Layer

Processes:

- Workflow Generation and Modeling: This process involves gathering requirements,
25 analyzing dependencies, and generating Petri-net models to orchestrate the interactions and data flows between GenFoundry modules.

Workflow Optimization and Adaptation: This process focuses on continuously optimizing the Petri-net workflows based on runtime performance, resource utilization, and feedback from OASIS, employing optimization strategies and reinforcement learning techniques.

- 30 Microservice Containerization and Packaging: This process involves automatically packaging GenFoundry modules into lightweight, self-contained containers using technologies like Docker, enabling flexible deployment and scalability.

Container Orchestration and Deployment: This process focuses on defining orchestration configurations, deploying containerized modules to container orchestration platforms, and managing their scheduling, scaling, and load balancing.

Functions:

- 5 Automated Workflow Generation: NEMO enables the computerized generation of Petri-net workflows based on the requirements and dependencies of GenFoundry modules, using LLMs and neural networks, reducing manual effort and ensuring consistency.

- Continuous Workflow Optimization: NEMO optimizes Petri-net workflows based on runtime performance and feedback, employing advanced optimization algorithms and reinforcement
10 learning techniques to improve efficiency and scalability.

Flexible Microservice Packaging: NEMO facilitates the flexible packaging of GenFoundry modules into lightweight, self-contained containers, enabling portability, scalability, and ease of deployment across different computing environments.

- Scalable Container Orchestration: NEMO offers scalable container orchestration capabilities,
15 leveraging platforms like Kubernetes or Docker Swarm to manage the deployment, scaling, and load balancing of containerized GenFoundry modules.

Application Layer

The NEMO system comprises the following key components:

- WorkflowGenerator: This component employs LLMs and neural networks to automatically
20 generate Petri-net models based on the requirements and dependencies of GenFoundry modules, capturing the interactions, data flows, and control logic.

PetriNetOptimizer: This component continuously analyzes the runtime performance, resource utilization, and feedback from OASIS to identify optimization opportunities in the Petri-net workflows, applying optimization strategies and reinforcement learning techniques.

- 25 MicroserviceComposer: This component automatically packages GenFoundry modules into lightweight, self-contained containers using technologies like Docker, ensuring portability and scalability.

- DeploymentOrchestrator: This component manages the deployment and orchestration of containerized GenFoundry modules to container orchestration platforms, such as
30 Kubernetes or Docker Swarm, based on defined configurations.

AdaptationManager: This component collaborates with OASIS to continuously monitor deployed workflows and microservices' performance and resource utilization, triggering adaptations and optimizations based on the collected insights.

Services:

- 5 WorkflowGenerationService: Provides methods for automatically generating Petri-net models based on the requirements and dependencies of GenFoundry modules, using LLMs and neural networks.

- 10 PetriNetOptimizationService: Offers services for continuously optimizing Petri-net workflows based on runtime performance, resource utilization, and feedback from OASIS, applying optimization strategies and reinforcement learning techniques.

MicroserviceCompositionService: Enables the automatic packaging of GenFoundry modules into lightweight, self-contained containers using technologies like Docker.

- 15 DeploymentOrchestrationService: Facilitates the deployment and orchestration of containerized GenFoundry modules to container orchestration platforms, such as Kubernetes or Docker Swarm, based on defined configurations.

AdaptationManagementService: Provides services for collaborating with OASIS to continuously monitor deployed workflows and microservices' performance and resource utilization, triggering adaptations and optimizations based on the collected insights.

Interfaces:

- 20 WorkflowGeneratorInterface: Defines the methods and parameters for automatically generating Petri-net models based on the requirements and dependencies of GenFoundry modules, using LLMs and neural networks.

- 25 PetriNetOptimizerInterface: Specifies the methods and input/output formats for continuously optimizing Petri-net workflows based on runtime performance, resource utilization, and feedback from OASIS, applying optimization strategies and reinforcement learning techniques.

MicroserviceComposerInterface: Describes the methods and parameters for automatically packaging GenFoundry modules into lightweight, self-contained containers using technologies like Docker.

- 30 DeploymentOrchestratorInterface: This interface defines the methods and input/output formats for deploying and orchestrating containerized GenFoundry modules to container

orchestration platforms, such as Kubernetes or Docker Swarm, based on defined configurations.

AdaptationManagerInterface: Specifies the methods and parameters for collaborating with OASIS to continuously monitor deployed workflows and microservices' performance and resource utilization, triggering adaptations and optimizations based on the collected insights.

OASIS

OASIS (Operations Automation and System Insights Supervisor) is an advanced agent-based process incorporating DevOps principles and automation toolchains to streamline the continuous delivery of updates and extensions to the GenFoundry modules. It also monitors the performance, health, and resource utilization of deployed GenFoundry modules, providing feedback to NEMO for continuous optimization and adaptation of the Petri-net choreography workflows.

At its core, OASIS employs a synergistic ensemble of specialized services collaborating through continuous optimization cycles, overseen by a DevOps automation model. The OASIS services, including BuildAutomator, TestOrchestrator, DeploymentManager, PerformanceMonitor, and ResourceOptimizer, work in concert to automate the build, testing, and deployment processes and monitor the performance and resource utilization of GenFoundry modules.

OASIS's automated continuous delivery capabilities leverage DevOps best practices and toolchains to ensure efficient and reliable delivery of updates and extensions to GenFoundry modules. Its self-adaptive monitoring and optimization capabilities, driven by advanced analytics and machine learning models, enable real-time insights into system performance and resource utilization, facilitating continuous improvement and adaptation.

Furthermore, OASIS incorporates a feedback-driven adaptation mechanism that collaborates with NEMO to optimize and adapt the Petri-net choreography workflows based on the monitored performance and resource utilization data. This closed-loop feedback method ensures that the GenFoundry remains optimized and aligned with evolving requirements and workload patterns.

OASIS's reliability and scalability are underpinned by its adherence to DevOps best practices and industry standards, such as Continuous Integration/Continuous Deployment (CI/CD), Infrastructure as Code (IaC), and containerization. OASIS's modular architecture and well-defined interfaces enable seamless integration with various DevOps tools, cloud platforms, and monitoring solutions.

With its automated delivery, monitoring, and optimization capabilities, OASIS ensures that the GenFoundry system remains up-to-date, performant, and resource-efficient. Its collaborative approach with NEMO enables continuous adaptation and improvement of the overall architecture, enhancing the reliability, scalability, and maintainability of the GenFoundry platform.

OASIS's architecture aligns with DevOps and Site Reliability Engineering (SRE) principles, promoting a culture of automation, collaboration, and continuous improvement. OASIS's robust validation mechanisms, which include automated testing, performance benchmarking, and chaos engineering techniques, ensure the stability and resilience of the GenFoundry system in production environments.

Methods

OASIS employs advanced DevOps automation, performance monitoring, and optimization techniques to streamline the continuous delivery and management of GenFoundry modules, ensuring efficient operations and resource utilization.

Automated Continuous Delivery: OASIS automates the build, testing, and deployment processes for GenFoundry modules, ensuring efficient and reliable delivery of updates and extensions. The automated continuous delivery process involves the following steps:

Code Integration: OASIS integrates the latest code changes from the development team into a centralized repository, triggering the automated build process.

Automated Build: The BuildAutomator agent compiles the code, resolves dependencies, and generates deployment artifacts, such as container images or executables.

Automated Testing: The TestOrchestrator agent manages the execution of automated tests, including unit tests, integration tests, and performance tests, ensuring the quality and reliability of the code changes.

Automated Deployment: The DeploymentManager agent handles deploying the tested and validated artifacts to the target environments, such as staging or production, using Infrastructure as Code (IaC) and containerization technologies.

Continuous Monitoring: OASIS continuously monitors the deployed modules, collecting performance metrics and logs and providing real-time visibility into the system's behavior.

The time complexity of the automated continuous delivery process depends on the size of the codebase, the number of tests, and the complexity of the deployment process. The

space complexity is determined by the size of the deployment artifacts and the infrastructure required for testing and deployment.

Performance Monitoring and Optimization: OASIS continuously monitors the performance, health, and resource utilization of deployed GenFoundry modules, providing real-time insights and enabling proactive optimization. The performance monitoring and optimization process involves the following steps:

Metric Collection: The PerformanceMonitor agent collects relevant performance metrics, such as response times, throughput, error rates, and resource utilization, from the deployed GenFoundry modules.

Anomaly Detection: OASIS analyzes the collected metrics using statistical methods and machine learning algorithms to detect anomalies, such as performance degradation or unusual resource consumption patterns.

Root Cause Analysis: When anomalies are detected, OASIS performs root cause analysis by correlating metrics, logs, and traces to identify the underlying issues, such as application bottlenecks, infrastructure failures, or configuration problems.

Optimization Recommendations: Based on the analysis, the ResourceOptimizer agent provides recommendations for optimizing the GenFoundry modules, such as scaling resources, tuning configurations, or refactoring code.

Continuous Improvement: OASIS collaborates with NEMO to implement the optimization recommendations and continuously monitor the impact of the changes, enabling a closed-loop feedback system for ongoing performance improvement.

The time complexity of the performance monitoring and optimization process depends on the volume of metrics collected, the complexity of anomaly detection algorithms, and the analysis frequency. The space complexity is determined by the size of the metrics data and the infrastructure required for storage and processing.

Feedback-Driven Adaptation: OASIS collaborates with NEMO to enable continuous optimization and adaptation of the Petri-net choreography workflows based on the monitored performance and resource utilization data. The feedback-driven adaptation process involves the following steps:

Data Collection: OASIS collects performance and resource utilization data from the deployed GenFoundry modules and shares it with NEMO.

Workflow Analysis: NEMO analyzes the collected data to identify inefficiencies, bottlenecks, or improvement opportunities in the Petri-net choreography workflows.

Adaptation Planning: Based on the analysis, NEMO generates adaptation plans, such as modifying the workflow structure, adjusting resource allocations, or optimizing task scheduling.

Workflow Adaptation: NEMO implements the adaptation plans by updating the Petri-net models and deploying the modified workflows to the GenFoundry system.

Continuous Monitoring: OASIS monitors the impact of the adaptations on the system's performance and resource utilization, providing feedback to NEMO for further refinement and optimization.

The time complexity of the feedback-driven adaptation process depends on the size and complexity of the Petri-net workflows, the frequency of the adaptations, and the efficiency of the optimization algorithms employed by NEMO. The space complexity is determined by the size of the performance and resource utilization data, as well as the infrastructure required for workflow analysis and adaptation.

Business Layer

Processes:

Continuous Integration and Delivery: This process involves automating the build, testing, and deployment of GenFoundry modules, ensuring efficient and reliable delivery of updates and extensions.

Performance Monitoring and Analysis: This process focuses on continuously monitoring the performance, health, and resource utilization of deployed GenFoundry modules, detecting anomalies, and providing real-time insights for optimization.

Resource Optimization and Scaling: This process involves analyzing resource utilization patterns, identifying inefficiencies, and recommending optimizing and scaling the GenFoundry modules to ensure optimal performance and cost-efficiency.

Feedback-Driven Adaptation and Improvement: This process collaborates with NEMO to enable continuous optimization and adaptation of the Petri-net choreography workflows based on the monitored performance and resource utilization data, ensuring ongoing system improvement.

Functions:

Automated Software Delivery: OASIS enables automated build, testing, and deployment of GenFoundry modules, reducing manual effort, increasing reliability, and accelerating the delivery of updates and extensions.

Proactive Performance Management: OASIS provides real-time monitoring and analysis of system performance and resource utilization, enabling proactive identification and resolution of issues before they impact end-users.

Continuous Optimization and Efficiency: OASIS facilitates continuous optimization of the GenFoundry system by providing actionable insights and recommendations for resource scaling, configuration tuning, and code refactoring, ensuring optimal performance and cost-efficiency.

Collaborative Workflow Adaptation: OASIS collaborates with NEMO to continuously adapt and improve the Petri-net choreography workflows based on real-time performance data, enabling a self-optimizing and resilient system architecture.

Application Layer

The OASIS system comprises the following key components:

BuildAutomator: This component automates the build process for GenFoundry modules, including code compilation, dependency management, and artifact generation.

TestOrchestrator: This component manages the automated testing of GenFoundry modules, including unit tests, integration tests, and performance tests, ensuring the quality and reliability of the delivered updates and extensions.

DeploymentManager: This component handles the automated deployment of GenFoundry modules across various environments, such as development, staging, and production, leveraging containerization and infrastructure-as-code technologies.

PerformanceMonitor: This component continuously monitors the performance of deployed GenFoundry modules, collecting metrics such as response times, throughput, and error rates, and provides real-time insights and alerts.

ResourceOptimizer: This component analyzes the resource utilization of deployed GenFoundry modules, identifying bottlenecks and inefficiencies, and provides recommendations for optimization and scaling.

Services:

BuildAutomationService: Provides methods for automating the build process of GenFoundry modules, including code compilation, dependency management, and artifact generation.

TestOrchestrationService: Offers services for managing the automated testing of GenFoundry modules, ensuring the quality and reliability of the delivered updates and extensions.

5 DeploymentManagementService: Enables the automated deployment of GenFoundry modules across various environments, leveraging containerization and infrastructure-as-code technologies.

PerformanceMonitoringService: Facilitates the continuous monitoring of deployed GenFoundry modules, collecting performance metrics and providing real-time insights and alerts.

10 ResourceOptimizationService: Provides services for analyzing resource utilization, identifying inefficiencies, and generating optimization and scaling recommendations for GenFoundry modules.

Interfaces:

15 BuildAutomatorInterface: This interface defines the methods and parameters for automating the build process of GenFoundry modules, including code compilation, dependency management, and artifact generation.

TestOrchestratorInterface: Specifies the methods and input/output formats for managing the automated testing of GenFoundry modules, ensuring the quality and reliability of the delivered updates and extensions.

20 DeploymentManagerInterface: This interface describes the methods and parameters for automating the deployment of GenFoundry modules across various environments, leveraging containerization and infrastructure-as-code technologies.

25 PerformanceMonitorInterface: Defines the methods and output formats for continuously monitoring the performance of deployed GenFoundry modules, collecting metrics, and providing real-time insights and alerts.

ResourceOptimizerInterface: Specifies the methods and input/output formats for analyzing resource utilization, identifying inefficiencies, and generating optimization and scaling recommendations for GenFoundry modules.

30 Genfoundry has a wide range of applications in industries that face challenges with data management, reporting, and compliance. Some key industries that can benefit from Genfoundry include:

1. Financial Services: Banks, insurance companies, asset managers, and capital market firms within the financial services industry have to comply with various regulatory reporting requirements such as Basel III, Solvency II, MiFID II, and FATCA. Genfoundry can help these organizations streamline their regulatory reporting processes, reduce compliance costs and risks, and improve data quality and consistency across different jurisdictions and business lines. With its AI-driven metadata management, DataVault2.0 schema generation, and cross-taxonomy mapping capabilities, financial institutions can adapt quickly to changing regulatory mandates and extract valuable insights from their data assets.
2. Healthcare and Life Sciences: The healthcare and life sciences industry generate vast amounts of complex and sensitive data, including electronic health records, clinical trial data, and research publications. Genfoundry can assist healthcare organizations and pharmaceutical companies in managing and integrating their data assets more effectively while ensuring compliance with data privacy and security regulations such as HIPAA and GDPR. Its AI-driven knowledge discovery and reasoning capabilities, combined with the CoLLEGe framework for abstracting key concepts and relationships, enable researchers and clinicians to extract valuable insights from unstructured medical data and accelerate drug discovery and personalized medicine.
3. Energy and Utilities: The energy and utilities sector is heavily regulated, with requirements for environmental impact, safety, and operational efficiency reporting. Genfoundry can streamline data management and reporting for energy companies, ensuring compliance with standards such as NERC CIP and ISO 50001. Through AI-driven metadata generation, DataVault2.0 schema abstraction, and mapping capabilities, energy firms can integrate and analyze data from various sources, including smart meters, sensor networks, and geospatial systems, to optimize operations and reduce costs.
4. Government and Public Sector: Government agencies and organizations handle a large amount of data related to public services, infrastructure, and citizen engagement. Genfoundry can greatly benefit these organizations by improving the efficiency, transparency, and accountability of their data management and reporting practices. It ensures compliance with open data standards and freedom of information regulations. With the use of AI-driven knowledge discovery and natural language query capabilities supported by the CoLLEGe framework, citizens and policymakers can access and analyze public data more efficiently. This promotes data-driven decision-making and innovation.
5. Manufacturing and Supply Chain: In the manufacturing and supply chain sector, data plays a crucial role in optimizing production processes, managing inventory, and ensuring product quality and safety. Genfoundry can assist manufacturers and logistics providers by

integrating and harmonizing data from different systems and formats, including ERP, MES, and SCM platforms. The AI-driven metadata generation, DataVault2.0 schema abstraction, and cross-taxonomy mapping capabilities enable firms to create a unified view of their operations, identifying opportunities for process improvement and cost reduction.

- 5 6. Telecommunications and Media: The telecommunications and media industry generates vast amounts of structured and unstructured data, such as customer records, network logs, and digital content. Genfoundry can significantly aid telecom and media companies in managing and monetizing their data assets effectively. It ensures data privacy and compliance with copyright regulations. By utilizing AI-driven knowledge discovery and
- 10 recommendation capabilities, along with the CoLLEGe framework for abstracting key concepts and relationships, firms can personalize their services, optimize content delivery, and identify new sources of revenue.
- Retail and Consumer Goods: The retail and consumer goods industry is currently facing numerous challenges when it comes to managing and utilizing customer data, supply chain information, and market trends. This is crucial in order
- 15 to stay competitive and meet the changing preferences of consumers. Fortunately, Genfoundry offers great support to retailers and consumer goods companies by helping them integrate and analyze data from various sources such as point-of-sale systems, customer relationship management (CRM) platforms, and social media channels. With the help of AI-driven metadata management, DataVault2.0 schema generation, and cross-
- 20 taxonomy mapping capabilities, firms are able to gain a comprehensive view of their customers. As a result, they can optimize their product offerings and enhance the efficiency of their supply chain.

The present disclosure, primarily discussed in the context of a multi-agent neural architecture operated by large language models (LLMs), generative AI (GenAI), and

25 quantum computing components, has the potential to be adapted and extended to include various alternative embodiments and variations. These modifications highlight Genfoundry's flexibility and versatility, making it capable of addressing multiple challenges related to data governance, security, and exploration across different domains and use cases.

One noteworthy variation involves the incorporation of federated learning techniques to

30 facilitate collaborative training and refinement of the LLM agents and generative AI models across distributed data silos. In situations where data cannot be centralized due to privacy, regulatory, or data residency limitations, federated learning enables multiple parties to collectively train the AI components using their respective local datasets without sharing raw data. This variation enhances Genfoundry's capacity to leverage diverse, globally distributed

data sources while ensuring data confidentiality and complying with stringent data sovereignty requirements.

Another alternative embodiment focuses on expanding the dynamic security access control system to support attribute-based encryption (ABE) and ciphertext-policy attribute-based encryption (CP-ABE) schemes. These advanced encryption methods allow for fine-grained, attribute-based access control over encrypted data, thereby enhancing the precision and flexibility of the policy-driven security model. By incorporating ABE and CP-ABE, the Genfoundry system can enforce complex access policies that take into account various attributes such as user roles, data sensitivity levels, environmental conditions, and purpose-based access restrictions.

Genfoundry can also be modified to integrate with homomorphic encryption techniques, allowing computations and query processing to be directly executed on encrypted data without decryption. This alternative embodiment further strengthens Genfoundry's capabilities in terms of data security and privacy since sensitive data remains encrypted throughout the processing and analysis pipeline, reducing the risk of unauthorized access or exposure.

In situations where data lineage and provenance tracking are crucial, particularly in regulated industries such as pharmaceuticals or financial services, Genfoundry can be extended to include blockchain-based provenance tracking. By utilizing distributed ledger technologies, the immutable audit trails documenting data transformations, access events, and analytical usage can be decentralized and cryptographically secured among multiple stakeholders. This enhances transparency, trust, and collaboration in data governance and regulatory compliance efforts. Genfoundry's modular, containerized architecture also allows for the integration of alternative quantum computing technologies, such as quantum annealing or ion trap quantum computers, as they mature and become commercially available. This flexibility ensures that Genfoundry can leverage the latest advancements in quantum hardware and algorithms, facilitating the smooth transition to more powerful quantum computing capabilities for encryption, provenance verification, and optimization of complex analytical workloads.

Another variation of the multi-agent architecture involves incorporating explainable AI (XAI) techniques to provide transparency and interpretability into the decision-making processes of the LLM agents, generative AI models, and query processors. By incorporating techniques such as attention visualization, concept activation vectors, and rule-based explanations, the multi-agent neural architecture can offer human-understandable insights into the reasoning

behind metadata extraction, security policy enforcement, and query result ranking, enhancing trust and enabling accountability in the overall system.

The multi-agent neural architecture can also support multi-modal data sources, including images, videos, audio recordings, sensor data, and structured and unstructured text data. By integrating computer vision, speech recognition, and signal processing capabilities, the LLM agents and generative AI models can automatically extract and enrich metadata from these diverse data modalities, enabling comprehensive data governance and exploration capabilities across various analytical datasets.

In summary, the alternative embodiments and variations discussed above demonstrate the flexibility and extensibility of the present disclosure. By incorporating federated learning, attribute-based encryption, homomorphic encryption, blockchain-based provenance tracking, alternative quantum computing technologies, explainable AI techniques, and multi-modal data support, the multi-agent architecture can be adapted to address a wide range of data governance, security, and exploration challenges across different industries and domains.

The modular, multi-agent architecture ensures that the system can seamlessly integrate with emerging technologies and evolving data landscapes, future-proofing the solution and enabling organizations to remain at the forefront of trusted, AI-powered data analytics.

FIG. 5 depicts a schematic representation of computing device 500, which is configured to support the operations and functionalities of the multi-agent neural architecture, as described in the embodiments herein. The device comprises a central processing unit (CPU) 522, which interfaces with various data storage and memory components: secondary storage 524, read-only memory (ROM) 526, and random-access memory (RAM) 528.

Within this configuration, a Quantum Computing Unit (QCU) 536 is integrated. The QCU 536 utilizes quantum mechanical phenomena to enhance computation. It is engineered to execute algorithms that are particularly suitable for quantum computation, such as those involving large-scale number factorization, quantum simulations, and specific optimization problems, enabling computational performance that exceeds the capabilities of a standalone CPU.

The secondary storage 524 may include a dedicated sector 524a containing instructions executable by both the CPU 522 and the QCU 536. These instructions enable the device 500 to perform operations that may be optimized through quantum computing.

The ROM 526 stores immutable code essential for an initial booting process and routine operations of the computing device 500. The RAM 528 provides volatile memory for immediate access to data by the CPU 522 and the QCU 536 during active tasks.

Peripheral devices are managed through input/output (I/O) interfaces 530, while network connectivity is enabled via a network interface 532. A graphics processing unit (GPU) 534 is present to handle parallel processing tasks, which may be separate from or integrated with quantum computing processes.

- 5 The CPU 522, as the primary processor for general computing tasks, is responsible for executing sequential operations and handling a variety of computational processes in the classical domain. It manages routine tasks with high efficiency and interfaces with the system's memory, including RAM 528 and ROM 526, for data storage and retrieval.

10 Within this dual-capability system, the CPU 522 often acts as a coordinator, determining when to engage the QCU 536 based on the computational requirements. For quantum-suitable tasks, the CPU 522 prepares and relays data to the QCU 536, which then processes this information utilizing its quantum computing power. After the QCU 536 completes the quantum processing, it can transmit the results back to the CPU 522, which may perform additional classical processing or output the results.

- 15 The collaboration between the CPU 522 and the QCU 536 effectively expands the device's 500 computational range, enabling it to switch between classical and quantum operations. This ensures that the device 600 utilizes the most efficient processing method available, whether that be the classical computation provided by the CPU 522 or the quantum processing offered by the QCU 536.

20 Although the computing device 500 is described with reference to a single computer, it should be appreciated that the computing device may be formed by two or more computers in communication with each other that collaborate to perform a task. For example, but not by way of limitation, an application may be partitioned in such a way as to permit concurrent and/or parallel processing of the instructions of the application. Alternatively, the data
25 processed by the application may be partitioned in such a way as to permit concurrent and/or parallel processing of different portions of a data set by the two or more computers. In an embodiment, virtualization software may be employed by the computing device 500 to provide the functionality of a number of servers that is not directly bound to the number of computers in the computing device 500. In an embodiment, the functionality disclosed above may be
30 provided by executing the application and/or applications in a cloud computing environment. Cloud computing may comprise providing computing services via a network connection using dynamically scalable computing resources. A cloud computing environment may be established by an enterprise and/or may be hired on an as-needed basis from a third-party provider.

Additional components, such as one or more application specific integrated circuits, neuromorphic computing units, field programmable gate arrays, or other electronic or photonic processing components can also be included and used in conjunction with or in place of the processor 522 to perform processing operations. The processing operations can include machine learning operations, other operations supporting the machine learning operations, or a combination thereof.

The technical solution detailed in present disclosure may be embodied in the form of a computer program product. The computer program product may be stored in a non-volatile or non-transitory storage medium, which can be a compact disk read-only memory, USB flash disk, or a removable hard disk. The computer program product includes a number of instructions that enable a computer device (personal computer, server, or network device) to execute the methods provided in the embodiments described herein. For example, such an execution may correspond to a simulation of the logical operations, including the training and aggregation of model updates in the federated learning process, as described herein. The software product may additionally or alternatively include number of instructions that enable the computing device 600 to execute operations for configuring or programming a digital logic apparatus in accordance with embodiments of the present disclosure.

By programming and/or loading executable instructions onto the computing device, at least one of the CPU 522, the RAM 528, and the ROM 526 are changed, transforming the computing device in part into a specific purpose machine or apparatus having the novel functionality taught by the present disclosure. It is fundamental to the electrical engineering and software engineering arts that functionality that can be implemented by loading executable software into a computer can be converted to a hardware implementation by well-known design rules.

Some embodiments (referred to as TinyLLMs) implement the GenFoundry methodology and agent architecture through its various components, such as MEGAN, SEPHYR, GAMA, ALFINI, QSEM, and CATM while focusing on domain-specific information. The domain specific information constitutes a knowledge base relevant to a specific problem context. For example, the domain specific information may constitute unstructured text. The unstructured text may include a technical manual for a software, information relevant to building applications, services, or solutions across various domains.

TinyLLM embodiments comprises a Knowledge Extraction Module for extracting data from the domain specific information and transform the knowledge into a structured representation of the knowledge. The structured knowledge representation is designed to be modular, highly extensible, and infinitely scalable, allowing for the seamless integration of new data sources, knowledge representation formats, and reasoning engines as the domain knowledge evolves.

In some embodiments, the structure knowledge representation may comply with relevant ISO standards, including ISO 25964 (Thesauri and Interoperability with Other Vocabularies), ISO 11179 (Metadata Registries), ISO 20944 (Ontology Development Methodologies), ISO 9001 (Quality Management Systems), and ISO 30401 (Knowledge Management Systems).

5 TinyLLM embodiments comprises a Knowledge Integration Module for consolidating the extracted knowledge into a structured representation (for example, a unified JSON representation). This module implements mechanisms for resolving conflicts that may arise during the integration process by incorporating voting algorithms or trust scoring techniques. It may also handle the deduplication and reconciliation of redundant knowledge entries,
10 improving accuracy and consistency. Furthermore, this module facilitates the mapping of extracted knowledge to well-defined formal ontologies or conceptual models. This improves semantic alignment and interoperability.

TinyLLM embodiments comprises an Ontology Representation component that enables the integration of well-defined formal ontologies and conceptual models within a Structured
15 knowledge representation (for example, a JSON structure). This component supports the representation of domain concepts, their properties, and relationships, allowing for modelling complex domains such as CDS (Concept Description Systems) models. It also includes metadata sections to capture essential information like document titles, descriptions, relevant ISO standards, and ontology development methodologies for improving compliance and
20 precision. The structured knowledge representation (example, JSON structure) follows a modular and hierarchical organization, mirroring the structure of the source documents from which the knowledge was extracted. TinyLLM embodiments may records changes made to the integrated knowledge over time and capture metadata about the sources from which knowledge was extracted. This enables transparent updates to the knowledge representation,
25 rollbacks, and traceability of knowledge sources and improves integrity and audibility of the knowledge representation.

TinyLLM embodiments may comprise a Knowledge Validation and Evaluation component for assessing the integrated knowledge's quality, completeness, and compliance with domain-specific constraints. This component implements mechanisms for automated validation and
30 workflows for incorporating human expertise through manual validation processes with absolute accuracy. Compliance checks against established guidelines and standards are performed by this component to improve the integrity of the knowledge representation.

TinyLLM embodiments enable integration of knowledge extracted from multiple structured knowledge representation sources (for example JSON representations) into a
35 unified/integrated structured knowledge representation. The integrated representation may be

compliant with relevant ISO standards, including ISO 25964 (Thesauri and Interoperability with Other Vocabularies), ISO 24564-2 (Interoperability with Other Vocabularies), ISO 11179 (Metadata Registries), ISO 20944 (Ontology Development Methodologies), ISO 9001 (Quality Management Systems), and ISO 30401 (Knowledge Management Systems), with utmost
5 precision.

When TinyLLM embodiments encounter conflicting elements (i.e., elements with the same key but different values) across the different structured knowledge representations, the embodiments resolve these conflicts by creating new keys to represent the conflicting elements from different sources or by merging the values of the contradictory aspects. Merging
10 techniques may include concatenating strings, combining arrays, or recursively merging objects with absolute accuracy.

TinyLLM embodiments may comprise a Metadata Integration Component which merges and harmonizes metadata information from individual structured knowledge representations.

TinyLLM embodiments may comprise an Ontology-based Representation Component that
15 leverages ontologies and formal knowledge representation methods to accurately capture domain concepts, their properties, and relationships, enabling semantic interoperability and reasoning capabilities.

It will be appreciated to those skilled in the art that the preceding examples and embodiments are exemplary and not limiting to the scope of the present disclosure. Whilst the foregoing
20 description has described exemplary embodiments, it will be understood by those skilled in the art that many variations of the embodiment can be made within the scope and spirit of the present disclosure. It shall be noted that elements of any claims may be arranged differently including have multiple dependencies, configurations and combinations.

25

30

Claims:

1. A multi-agent system for processing information, comprising:

a data processing agent configured to ingest and normalize raw data inputs to produce standardized data;

5 a standards integration agent configured to apply reporting standards into the standardized data thereby generating integrated reporting standards;

a performance alignment agent configured to align performance indicators based on the standardized data and the integrated reporting standards;

an information synthesis agent configured to process narrative information from the
10 standardized data and the integrated reporting standards; and

an orchestration framework configured to manage operations of the data processing agent, the standards integration agent, the performance alignment agent, and the information synthesis agent to produce a regulatory report compliant with regulatory requirements, wherein the orchestration framework is executable by a large language model.
- 15 2. The system of claim 1, wherein the data processing agent further comprises a data cleansing module capable of removing errors and inconsistencies from the raw data inputs to improve the accuracy of the standardized data.
3. The system of claim 1 or 2, further comprising a materiality alignment agent configured to evaluate and align materiality and boundaries based on the standardized data and the
20 integrated reporting standards.
4. The system of any one of the preceding claims, further comprising a compliance alignment agent configured to align assurance processes based on the standardized data and the integrated reporting standards.
5. The system of any one of the preceding claims, wherein the performance alignment agent
25 uses machine learning to dynamically adapt the performance indicators based on updates to the reporting standards and corresponding real-time data.
6. The system of any one of the preceding claims, further comprising a report consolidation agent configured to employ a data integration platform for merging and alignment of output data from the data processing agent, the standards integration agent, the performance
30 alignment agent and the information synthesis agent.

7. The system of any one of the preceding claims, wherein the orchestration framework further comprises a scheduling module that adjusts a sequence and priority of tasks based on real-time assessments of data processing needs and agent capacity.
8. The system of any one of the preceding claims, wherein the information synthesis agent further comprises a contextual analysis module configured to integrate contextual cues from the standardized data and the integrated reporting standards.
9. A system for managing metadata, comprising:
- an ingestion module configured to preprocess data inputs in compliance with a metadata standard;
- an extraction module configured to employ language processing techniques to extract and normalize metadata from the preprocessed data inputs;
- an abstraction module configured to transform the extracted and normalized metadata into structured schemas;
- a mapping module configured to use artificial intelligence techniques to translate the transformed metadata based on taxonomies; and
- a storage module configured to index and enable searches on the translated metadata.
10. The system of claim 9, wherein the mapping module is further configured to utilize ontology-based reasoning and semantic similarity measures to establish mappings between metadata elements from different taxonomies.
11. The system of claim 9 or 10, wherein the mapping module is further configured to employ neural networks and transfer learning to refine and enhance the translation of metadata between taxonomies.
12. The system of any one of claims 9 to 11, wherein the storage module includes:
- a vector database configured to store the translated metadata, an indexing module configured to generate vector embeddings corresponding to the translated metadata, and a search engine module configured to use the vector embeddings for performing searches based on contextual relevance and semantic similarity.
13. The system of any one of claims 9 to 12, further comprising a data lineage and provenance tracking module configured to capture an audit trail of metadata management processes across the system for compliance with regulatory requirements.

14. The system of any of claims 9 to 13, wherein the ingestion module is further configured to directly interface with external data sources to automatically retrieve the data inputs

15. A system for extracting and normalizing metadata, comprising:

an input interface configured to receive preprocessed data inputs that comply with a
5 metadata standard;

a processing module configured to apply natural language processing techniques to extract metadata from the received preprocessed data inputs;

a normalization engine configured to normalize the extracted metadata according to predetermined metadata standards;

10 a refinement module configured to enhance the normalized metadata with additional information derived from external sources for generating a refined metadata; and

an output interface configured to output the refined metadata for additional processing within a metadata management system.

16. A system for integrating data patterns into a data representation, comprising:

15 a data input interface configured to receive data from a plurality of sources;

a feature extraction module configured to extract features within the received data;

a security module configured to encrypt the extracted features to generate a secured data representation;

20 a data integration module configured to integrate the encrypted features corresponding to the secured data representation into a unified data representation; and

a validation module configured to validate the unified data representation against a predefined standard or regulation.

17. The system of claim 16, wherein the security module employs cryptographic hash functions to verify authenticity and integrity of the secured data representation.

25 18. The system of claim 16 or 17, wherein the feature extraction module is configured to use natural language processing and machine learning algorithms to extract features from the received data.

19. The system of any one of claims 16 to 18, further comprising an access management module configured to translate at least one of user roles, user requirements, or data access
30 patterns into a unified mathematical representation.

20. The system of claim 19, wherein the access management module is further configured to use natural language processing and machine learning to encode the user roles and access privileges into unique numerical codes.

21. The system of claim 19 or 20, wherein the access management module includes a control engine, which serves as an authority for enforcing access controls based on a unified security access model.

22. The system of any one of claims 19 to 21, wherein the access management module is configured to automatically synchronize its access control rules with external compliance monitoring systems to maintain adherence to changes in regulations.

23. A method for processing information in a multi-agent system, comprising:

ingesting raw data inputs via a data processing agent;

normalizing the ingested raw data to produce standardized data using said data processing agent;

applying reporting standards to the standardized data using a standards integration agent to generate integrated reporting standards;

aligning performance indicators based on the standardized data and the integrated reporting standards using a performance alignment agent;

processing narrative information from the standardized data and the integrated reporting standards using an information synthesis agent; and

orchestrating operations of the data processing agent, the standards integration agent, the performance alignment agent, and the information synthesis agent through an orchestration framework to produce a regulatory report compliant with encoded regulatory requirements, wherein the orchestration framework is executable by a large language model;