CLAIMS:

1. A computer-implemented method of network discovery, comprising:

collecting configuration files from at least one network, each configuration file comprising configuration data that characterizes different assets in the at least one network and their respective connections and properties;

applying on the configuration files a large language model (LLM) trained to provide as output responsive to at least one received instruction, a respective response comprising specific configuration data extracted from the configuration files; wherein the LLM is capable of retrieving configuration data pertaining to multiple assets and their respective connections in the at least one network.

2. The computer-implemented method of claim 1 comprising generating, based on the retrieved configuration data, an ontology of the at least one network including, at least part of: information on the assets, information on network topology, and information on rules and policies.

3. The computer-implemented method of claim 1 or 2 comprising:

using the LLM for generating, for each asset and/or connection in the at least one network, a respective data object, wherein all data-objects are characterized by a uniform data-object format.

4. The computer-implemented method of any one of the preceding claims comprising: generating a graph representing the retrieved configuration data, wherein each data-object of a respective asset retrieved from the retrieved configuration data is designated as a node in the graph, and each connection retrieved from the retrieved configuration data is designated as a vertex in the graph connecting between two nodes, and where each node or vertex is stored with its respective properties.

5. The computer-implemented method of any one of the preceding claims comprising:

applying the LLM on the configuration files in the at least one network for determining a type of configuration file, and, responsive to determining a configuration file of a certain type, automatically generating a regular expression

adapted for parsing the configuration file and retrieving from the configuration file the respective configuration data.

6. The computer-implemented method of any one of the preceding claims comprising:

applying the LLM on the configuration files in the at least one network for determining a type of configuration file, and, responsive to determining a configuration file of a certain type, automatically generating a computer program code adapted for parsing the configuration file and retrieving from the configuration file the respective configuration data.

7. The computer-implemented method of any one of the preceding claims comprising: comparing configuration data in a respective response of one or more instructions to configuration data in the configuration files, and, in case of a discrepancy, discarding the respective response and re-processing the one or more instructions by the LLM.

8. The computer-implemented method of any one of the preceding claims wherein a training dataset for training the LLM comprises synthetic configuration files.

9. The computer-implemented method of claim 8, wherein the synthetic configuration files are generated by a process that utilizes network design software, comprising:

manipulating parameters of the network design software to thereby obtain a diverse collection of network designs; and

executing the network designs to thereby obtain a respective diverse collection of configuration files.

10. The computer-implemented method of claim 9, wherein manipulation of parameters of network design software includes intentionally inserting errors into network designs in the diverse collection of network designs, thereby obtaining configuration files that contain errors.

11. The computer-implemented method of any one of claims 8 to 10, wherein the training dataset includes an assembly of the synthetic configuration files,

proprietary configuration files, and human annotated configuration files.

12. The computer-implemented method of any one of claims 1 to 11, wherein the training dataset comprises providing, as input, multiple instruction-response pairs, and training the LLM to provide, in response to a given instruction, a respective response comprising specific configuration data extracted from the configuration files.

13. The computer-implemented method of claim 12, wherein the multiple instruction-response pairs include at least one of:

i. instruction-Regex pairs used for training the LLM to automatically generate, responsive to determining that a configuration file is of a certain type, a regular expression adapted for parsing the configuration file and retrieving from the configuration file the respective configuration data; and

ii. instruction-computer code pairs used for training the LLM to automatically generate, responsive to determining that a configuration file is of a certain type, a computer program code adapted for parsing the configuration file and retrieving from the configuration file the respective configuration data.

14. The computer-implemented method of any one of the preceding claims, wherein the at least one network is an operation technology network and/or an information technology network.

15. The computer-implemented method of any one of claims 1 to 14 wherein the LLM includes 'n' embedded tokens, each generated from a respective token extracted from the configuration files, the computer-implemented method comprising:

applying in the LLM a selective attention mechanism, when providing a response to a received instruction, comprising:

selecting a subset of 'm' from the 'n' embedded tokens and using only the embedded tokens in the subset when applying the attention mechanism.

16. The computer-implemented method of claim 15, further comprising:
for at least one embedded token generated from the received instruction:
calculating a dot product using the subset of 'm' embedded tokens, giving rise

to $\frac{n}{m}$ dot products;

selecting from the subset, a group of k embedded tokens with the highest respective dot products;

expanding each of 'k' embedded tokens with a plurality of additional 'p' adjacent tokens, giving rise to a final collection of $\frac{n}{m} + k * p$ embedded tokens;

applying a normalization function on respective dot product of the final collection of embedded tokens, thereby obtaining a respective set of weights, and applying the weights for obtaining a contextualized vector for the at least one embedded token.

17. A computer program product comprising a computer readable storage medium retaining a program of instructions, which, when read by a computer processor, causes the computer processor to perform a method according to any one of claims 1 to 16.

18. A computer system comprising at least one processing circuitry configured to execute a method of network discovery using an LLM according to any one of claims 1 to 16.