CLAIMS

1. A device comprising:

a storage system including one or more computer-readable storage mediums, wherein the storage system stores:

a computer program comprised of a set of encoded computer-executable instructions; and

a genomic data set that includes a genomic differentiation object that was used to encode the set of encoded computer-executable instructions;

genomic regulation instructions (GRI) corresponding to the computer program, wherein the genomic regulation instructions were used to encode the set of encoded computer-executable instructions;

a processing system comprising:

a VDAX that is configured to:

receive encoded instructions to be executed from the set of encoded computer-executable instructions; and

decode the encoded instructions into decoded executable instruction based on a modified genomic differentiation object and sequences extracted from metadata associated with the encoded instructions, wherein the modified genomic differentiation object is modified from the genomic differentiation object using the GRI; and

a set of processing cores that are configured to receive the decoded executable instructions from the VDAX and to execute the decoded executable instructions.

2. The device of claim 1, wherein the VDAX is configured to:

modify the genomic differentiation object based on the GRI using a set of computational functions to obtain the modified genomic differentiation object; and

for each respective encoded instruction that is to be executed by the processing device:

obtain a genomic engagement factor (GEF) corresponding to the respective encoded computer-executable instruction, wherein the GEF is generated based on a sequence that is extracted from metadata corresponding to the respective encoded computer-executable instruction, the GRI, and the modified genomic data object;

decode the respective encoded instruction using the GEF corresponding to the respective encoded instruction to be executed to obtain a respective decoded executable instruction; and

output the respective decoded instruction to a processing core of the set of processing cores.

3. The device of claim 2, wherein decoding the respective encoded instruction using the GEF includes decrypting the respective encoded instruction using the GEF as a key to obtain the respective decoded instruction.

4. The device of claim 2, wherein decoding the respective encoded instruction using the GEF includes disambiguating the respective decoded instruction from the respective encoded instruction using the respective GEF, and wherein disambiguating the respective decoded instruction from the respective encoded instruction includes executing an XOR operation using the encoded instruction and the GEF as input to the XOR operation.

~~5. The device of claim 4, wherein disambiguating the respective decoded instruction from the respective encoded instruction includes executing an XOR operation using the encoded instruction and the GEF as input to the XOR operation.~~

~~6~~5. The device of claim 2, wherein each respective encoded instruction was encoded with a different respective GEF, such that each respective GEF is determined based on a respective sequence extracted from respective metadata that uniquely corresponds to the respective encoded instruction.

~~7~~6. The device of claim ~~6~~ 5, wherein the respective metadata that uniquely corresponds to the respective encoded instruction includes one or more of random-access memory (RAM) metadata, application metadata, operating system-application metadata, and filesystem metadata.

~~8~~7. The device of claim ~~7~~ 6, wherein the respective sequence that is used to generate the respective GEF that is used to decode the respective encoded instruction is a set of bits extracted from a bit representation of the respective metadata.

~~9~~8. The device of claim 2, wherein the set of encoded computer-executable instructions includes a plurality of sub-sets of instructions, such that each subset of plurality of subsets is decoded with a respective GEF that is generated based on a respective sequence extracted from metadata that respectively corresponds to the respective subset of computer-executable instructions.

~~10~~9. The device of claim ~~9~~ 8, wherein the respective metadata from which the respective sequence was extracted includes one or more of random-access memory (RAM) metadata corresponding to the subset, application metadata corresponding to the subset, operating system-application metadata corresponding to the subset, and filesystem metadata corresponding to the subset.

~~11~~10. The device of claim 2, wherein obtaining the GEF corresponding to the respective encoded computer-executable instruction includes sequence mapping the sequence into the modified genomic data object to obtain the GEF, and wherein sequence mapping the sequence into the modified genomic data object includes:

generating a sequence conversion vector (SCV) based on the sequence and the GRI;

mapping the modified genomic differentiation object based on the SCV to obtain a mapped modified genomic differentiation object; and

generating the GEF based on the mapped modified genomic differentiation object.

~~12. The device of claim 11, wherein sequence mapping the sequence into the modified genomic data object includes:~~

~~generating a sequence conversion vector (SCV) based on the sequence and the GRI;~~

~~mapping the modified genomic differentiation object based on the SCV to obtain a mapped modified genomic differentiation object; and~~

~~generating the GEF based on the mapped modified genomic differentiation object.~~

~~13. The device of claim 1, wherein the storage system stores a second computer program comprising a second set of encoded computer-executable instructions and second GRI corresponding to the second computer program that are used to decode the second set of encoded computer-executable instructions.~~

~~14. The device of claim 11, wherein the second GRI are different from the GRI used to decode the set of encoded computer-executable instructions.~~

~~15. The device of claim 14, wherein the storage device further stores a second genomic differentiation object that is used to encode and decode the second set of encoded computer-executable instructions and is different from the genomic differentiation object used to encode and decode the set of encoded computer-executable instructions.~~

16. The device of claim 14, wherein the genomic differentiation object is modified using the second GRI to obtain a second modified genomic differentiation object, such that the second modified genomic differentiation object is used to decode the second set of encoded computer-executable instructions.

17. The device of claim 1, wherein the processing system is a processing device and the VDAX is a dedicated core of the processing device.

18. The device of claim 1, wherein the VDAX is a field programmable gate array that is in electrical communication with the set of processing cores.

19. The device of claim 1, wherein the VDAX is a microprocessor that is in electrical communication with the set of processing cores.

20. The device of claim 1, wherein the VDAX is included in an operating system that is executed by the set of processing cores.

21. The device of claim 1, wherein the genomic differentiation object is a ZNA object.

22. The device of claim 1, wherein the device is a server computing device.

23. The device of claim 1, wherein the device is a mobile computing device.

24. The device of claim 1, wherein the device is a personal computing device.

25. The device of claim 1, wherein the device is a vehicle.

26. The device of claim 1, wherein the device is an internet of things device.

27. The device of claim 1, wherein the device is a sensor device.

28. The device of claim 1, wherein the device is a video recording device.

29. The device of claim 1, wherein the device is a wearable computing device.

30. The device of claim 1, wherein the device is a smart appliance.

31. The device of claim 1, wherein the device is a network router.

32. The device of claim 1, wherein the device is a gaming device.

33. The device of claim 1, wherein the computer program is a software application.

34. The device of claim 1, wherein the computer program is a middleware application.

35. The device of claim 1, wherein the computer program is a firmware application.

36. The device of claim 1, wherein the computer program is an operating system.

37. A method for executing computer programs in a trusted execution environment of a device comprising:

retrieving a genomic differentiation object corresponding to a set of one or more computer programs including a first computer program, wherein the first computer program comprises a first set of encoded computer-executable instructions and the genomic differentiation object comprises a binary vector of specific length;

modifying the genomic differentiation object based on first genomic regulation instructions (GRI) to obtain a first modified genomic differentiation object, wherein the first GRI were used to encode the first set of encoded executable instructions of the first computer program;

obtaining a first encoded instruction that is to be executed from the first set of encoded computer-executable instructions of the first computer program;

obtaining a first sequence from first metadata associated with the first encoded instruction;

generating a first genomic engagement factor (GEF) based on the first sequence and the first modified genomic differentiation object;

decoding the first encoded instruction using the first GEF to obtain a first decoded instruction; and

executing the first decoded instruction.

38. The method of claim 37, further comprising:

obtaining a second encoded instruction that is to be executed from the first set of encoded computer-executable instructions of the first computer program;

obtaining a second sequence from second metadata associated with the second encoded instruction;

~~generating a second GEF based on the second sequence and the modified genomic differentiation object;~~

~~decoding the second encoded instruction using the second GEF to obtain a second decoded instruction; and~~

~~executing the second decoded instruction.~~

~~39. The method of claim 37, further comprising:~~

~~modifying the genomic differentiation object with second GRI corresponding to a second computer program of the set of computer programs to obtain a second modified genomic differentiation object, wherein the second computer program comprises a second set of encoded computer-executable instructions;~~

~~obtaining a third encoded instruction that is to be executed from the second set of encoded computer-executable instructions of the second computer program;~~

~~obtaining a third sequence from third metadata associated with the third encoded instruction;~~

~~generating a third GEF based on the third sequence and the second modified genomic differentiation object;~~

~~decoding the third encoded instruction using the third GEF to obtain a third decoded instruction; and~~

~~executing the third decoded instruction.~~

~~40. The method of claim 39, wherein the genomic differentiation object is assigned to both the first computer program and the second computer program.~~

~~41. The method of claim 40, wherein the first GRI corresponds only to the first computer program and the second GRI corresponds only to the second computer program.~~

~~42. The method of claim 40, wherein the first computer program and the second computer program and any other computer programs of the set of computer programs are distributed by the same entity.~~

~~43. The method of claim 42, wherein the entity assigns the genomic differentiation object to the first and second computer programs.~~

~~44. The method of claim 39, wherein a VDAX of the device assigns the genomic differentiation object to the first and second computer programs.~~

45. The method of claim 44, wherein the VDAX generates the first GRI and the second GRI.

46. The method of claim 37, further comprising:

installing the first computer program prior to decoding the first encoded instruction and the second encoded instruction.

47. The method of claim 46, wherein installing the first computer program includes:

obtaining a set of unencoded computer-executable instructions that comprise the first computer program;

allocating memory on a computer-readable storage medium of the device to the first computer program;

generating the first GRI and associating the first GRI with the first computer program;

modifying the genomic differentiation object based on the first GRI to obtain the modified genomic differentiation object;

generating the first set of encoded computer-executable instructions by, for each respective unencoded instruction in the set of unencoded computer-executable instructions:

determining a respective sequence from respective metadata associated with the respective unencoded instruction;

generating a respective GEF based on the respective sequence and the modified genomic differentiation object;

encoding the respective unencoded using the respective GEF to obtain a respective encoded instruction of the first set of encoded computer-executable instructions; and

storing the respective encoded instruction at a memory location of the allocated memory.

48. The method of claim 47, wherein the respective metadata from which the respective sequence is extracted includes one or more of random-access memory (RAM) metadata, application metadata, operating system-application metadata, and filesystem metadata.

49. The method of claim 37, wherein the execution environment is a processing device that includes an executable isolation components VDAX that is a dedicated core of the processing device that decodes encoded instructions and a set of processing cores that execute the decoded instructions.

50. The method of claim 37, wherein the genomic differentiation object is a ZNA object.

~~51. The method of claim 37, wherein the device is a server computing device.~~

~~52. The method of claim 37, wherein the device is a mobile computing device.~~

~~53. The method of claim 37, wherein the device is a personal computing device.~~

~~54. The method of claim 37, wherein the device is a vehicle.~~

~~55. The method of claim 37, wherein the device is a gaming device.~~

~~56. The method of claim 37, wherein the device is an internet of things device.~~

~~57. The method of claim 37, wherein the device is a sensor device.~~

~~58. The method of claim 37, wherein the device is a video recording device.~~

~~59. The method of claim 37, wherein the device is a wearable computing device.~~

~~60. The method of claim 37, wherein the device is a smart appliance.~~

~~61. The method of claim 37, wherein the device is a network router.~~

~~62. The method of claim 37, wherein the first computer program is a software application.~~

~~63. The method of claim 37, wherein the first computer program is a middleware application.~~

~~64. The method of claim 37, wherein the first computer program is a firmware application.~~

~~65. The method of claim 37, wherein the first computer program is an operating system.~~

~~66~~11. A method for executing computer programs in a trusted execution environment of a device comprising:

retrieving a genomic differentiation object corresponding to a set of one or more computer programs including a computer program;

modifying the genomic differentiation object based on genomic regulation instructions (GRI) to obtain a modified genomic differentiation object;

executing a first executable instruction of the computer program, wherein executing the first executable instruction includes:

retrieving first encoded data that is input to the first executable instruction;

extracting a first sequence from first metadata associated with the first encoded data;

generating a first genomic engagement factor (GEF) based on the first sequence, the GRI and, and the modified genomic differentiation object;

decoding the first encoded data based on the first GEF to obtain first decoded data;

executing the first executable instruction using the first decoded data; and

executing a second executable instruction of the computer program, wherein executing the second executable instruction includes:

retrieving second encoded data that is input to the second executable instruction;

extracting a second sequence from second metadata associated with the second encoded data;

generating a second GEF based on the second sequence, the GRI and, and the modified genomic differentiation object;

decoding the second encoded data based on the second GEF to obtain second decoded data; and

executing the second executable instruction using the second decoded data.

~~67~~12. The method of claim ~~66~~ 11, wherein executing the first executable instruction further includes:

obtaining output data resulting from execution of the first executable instruction;

generating a third GEF based on a third sequence corresponding to the output data, the GRI, and the modified genomic differentiation object; ~~and~~

transforming the output data into third encoded data based on the third GEF, and

writing the third encoded data to memory

~~68. The method of claim 67, further comprising writing the third encoded data to memory.~~

~~69~~13. The method of claim ~~68~~ 12, wherein the third sequence is extracted from metadata corresponding to the output data, wherein the metadata includes one or more of random-access memory (RAM) metadata, application metadata, operating system-application metadata, and filesystem metadata, the method further comprising executing a third executable instruction of the computer program by:

retrieving the third encoded data that is input to the third executable instruction;

extracting the third sequence from a value representing the metadata corresponding to the output data;

generating the third GEF based on the third sequence, the GRI and, and the modified genomic differentiation object;

decoding the third encoded data based on the third GEF to obtain the output data; and

executing the first executable instruction using the output data resulting from execution of the first executable instruction

~~70. The method of claim 69, further comprising executing a third executable instruction of the computer program by:~~

~~retrieving the third encoded data that is input to the third executable instruction;~~

~~extracting the third sequence from a value representing the metadata corresponding to the output data;~~

~~generating the third GEF based on the third sequence, the GRI and, and the modified genomic differentiation object;~~

~~decoding the third encoded data based on the third GEF to obtain the output data; and~~

~~executing the first executable instruction using the output data resulting from execution of the first executable instruction.~~

~~71~~14. The method of claim ~~66~~ 11, wherein the first metadata from which the first sequence is extracted and the second metadata from which the second sequence is extracted includes one or more of random-access memory (RAM) metadata, application metadata, operating system-application metadata, and filesystem metadata, and wherein generating the first GEF includes:

generating a sequence conversion vector (SCV) based on the first sequence and the first GRI;

mapping the modified genomic differentiation object based on the SCV to obtain a mapped modified genomic differentiation object; and

generating the first GEF based on the mapped modified genomic differentiation object.

~~72. The method of claim 66, wherein generating the first GEF includes:~~

~~generating a sequence conversion vector (SCV) based on the first sequence and the first GRI;~~

~~mapping the modified genomic differentiation object based on the SCV to obtain a mapped modified genomic differentiation object; and~~

~~generating the first GEF based on the mapped modified genomic differentiation object.~~

~~73. The method of claim 72, wherein generating the SCV includes combining the first sequence and the GRI.~~

~~74. The method of claim 73, wherein generating the SCV further includes applying one or more computational functions to the combination of the first sequence and the GRI to obtain the SCV.~~

~~75~~15. The method of claim ~~66~~ 11, wherein decoding the first encoded data includes decrypting the first encoded data using the first GEF and decoding the second encoded data includes decrypting the second encoded data using the second GEF.

~~76~~16. The method of claim ~~66~~ 11, wherein decoding the first encoded data includes disambiguating the first encoded data using the first GEF and disambiguating the second encoded data includes decrypting the second encoded data using the second GEF.

~~77. The method of claim 66, wherein the genomic differentiation object is an ZNA object.~~

~~78. The method of claim 77, wherein the ZNA object is an N x M binary matrix.~~

~~79. The method of claim 77, wherein a VDAX of the device generates and assigns the ZNA object to the computer program.~~

~~80. The method of claim 66, wherein the device is a server computing device.~~

~~81. The method of claim 66, wherein the device is a mobile computing device.~~

~~82. The method of claim 66, wherein the device is a personal computing device.~~

~~83. The method of claim 66, wherein the device is a vehicle.~~

~~84. The method of claim 66, wherein the device is a gaming device.~~

~~85. The method of claim 66, wherein the device is an internet of things device.~~

~~86. The method of claim 66, wherein the device is a sensor device.~~

87. The method of claim 66, wherein the device is a video recording device.

88. The method of claim 66, wherein the device is a wearable computing device.

89. The method of claim 66, wherein the device is a smart appliance.

90. The method of claim 66, wherein the device is a network router.

91. The method of claim 66, wherein the computer program is a software application.

92. The method of claim 66, wherein the computer program is a middleware application.

93. The method of claim 66, wherein the computer program is a firmware application.

94. The method of claim 66, wherein the computer program is an operating system.

95. A method for verifying an unverified portion of a material data chain (MDC) that is maintained by a creator cohort in a ledger-based digital ecosystem, the method comprising:

receiving, by a processing system of a verification cohort, the unverified portion of the MDC from the creator cohort, the unverified portion of the MDC including a set of consecutive material data blocks (MDBs) generated by the creator cohort, wherein each respective MDB includes respective material data that is stored in the respective MDB, respective metadata relating to the MDB, and a creator-generated verification value generated by the creator cohort with respect to the verification cohort;

retrieving, by the processing system of the verification cohort, a genomic differentiation object assigned to the verification cohort with respect to the ledger-based ecosystem;

retrieving, by the processing system of the verification cohort, first genomic regulation instructions (GRI) used by the creator cohort to generate the creator-generated verification value;

modifying, by the processing system of the verification cohort, the genomic differentiation object based on the first GRI using a set of computational functions to obtain a modified differentiation object;

for each MDB in the unverified portion of the MDC:

generating, by the processing system, a genomic engagement factor (GEF) based on a sequence extracted with respect to the MDB, the modified genomic differentiation object, and the first GRI;

~~determining a verifier-generated verification value based on the MDB, an MDB preceding the MDB in the MDC, and the GEF;~~

~~determining, by the processing system, whether the verifier-generated verification value matches the creator-generated verification value in the MDB; and~~

~~in response to determining that the verifier-generated verification value matches the creator-generated verification value, generating a verification record corresponding to the MDB indicating that the MDB has been verified by the verification cohort,~~

~~wherein the unverified portion of the MDC is verified by the verification cohort when each of the MDBs in the unverified portion of the MDC have been verified by the verification cohort.~~

~~96. The method of claim 95, wherein the creator-generated verification value of each MDB is generated by the creator cohort based on the MDB, an MDB preceding the MDB in the MDC, and a second modified genomic differentiation object that is derived by the creator cohort by modifying a second genomic differentiation object assigned to the creator cohort using second GRI, wherein the second GRI matches the first GRI.~~

~~97. The method of claim 96, wherein the creator-generated verification value of a respective MDB matches the verifier-generated verification value generated by the verification cohort only if the genomic differentiation object assigned to the verification cohort is sufficiently correlated with the second genomic differentiation object assigned to the creator cohort.~~

~~98. The method of claim 96, wherein the creator-generated verification value of a respective MDB matches the verifier-generated verification value generated by the verification cohort only if the genomic differentiation object assigned to the verification cohort exactly matches the second genomic differentiation object assigned to the creator cohort.~~

~~99. The method of claim 96, wherein the genomic differentiation object is assigned to the verification cohort by an ecosystem VDAX that controls a genomic topology of the ledger-based ecosystem and the second genomic differentiation object is assigned to the creator cohort by the ecosystem VDAX.~~

~~100. The method of claim 95, wherein the first GRI was generated and provided by the creation cohort during a one-time link exchange process.~~

~~101. The method of claim 95, wherein the first GRI was generated by the verification cohort and provided to the creation cohort during a one-time link exchange process.~~

~~102. The method of claim 95, wherein the first GRI was generated by an enclave VDAX and provided to the creation cohort, the verification cohort, and one or more other verification cohorts, such that the first GRI is used by the verification cohort and the one or more other verification cohorts to verify the creator-generated verification value.~~

~~103. The method of claim 95, wherein generating the GEF includes:~~

~~generating a sequence conversion vector based on the sequence and the first GRI; and~~

~~mapping the modified genomic differentiation object based on the sequence conversion vector to obtain a mapped genomic differentiation object; and~~

~~generating the GEF based on the mapped genomic differentiation object.~~

~~104. The method of claim 95, wherein the generating the verifier-generated verification value includes:~~

~~generating a linking value based on the MDB and the preceding MDB in the MDC, the linking value being indicative of a combination of the MDB and the preceding MDB; and~~

~~applying one or more computational functions to the linking value based on the GEF to obtain the verifier-generated verification value.~~

~~105. The method of claim 104, wherein generating the linking value includes:~~

~~generating a first hash value of at least a portion of the MDB;~~

~~generating a second hash value of at least a portion of the preceding MDB; and~~

~~combining the first hash value and the second hash value to obtain the linking value.~~

~~106. The method of claim 104, wherein the one or more computational functions include one or more cipher-based functions that transform the linking value into the verifier-generated verification value using the GEF as a key.~~

~~107. The method of claim 106, wherein the one or more cipher-based functions include an encryption function.~~

~~108. The method of claim 106, wherein the one or more cipher-based functions include a disambiguation function.~~

~~109. The method of claim 105, wherein the one or more computational functions include one or more cipherless functions that transform the linking value into the verifier-generated verification value using the GEF as an input parameter.~~

~~110. The method of claim 109, wherein the one or more cipherless functions include a hash function.~~

~~111. The method of claim 104, wherein the one or more computational functions include one or more cipherless functions and one or more cipher-based functions that in combination transform the linking value into the verifier-generated verification value using the GEF as an input parameter.~~

~~112. The method of claim 95, further comprising transmitting the verification record to the creator cohort.~~

~~113. The method of claim 112, further comprising storing, by the processing system of the verification cohort, the verification record in a verification chain that is maintained by the verification cohort.~~

~~114. The method of claim 113, wherein storing the verification record includes:~~

~~generating a verification block containing one or more verification records corresponding to one or more respective MDBs in the MDC; and~~

~~adding the verification block to the verification chain.~~

~~115. The method of claim 114, wherein the verification chain is a sidechain of a second MDC maintained by the verification cohort.~~

~~116. The method of claim 112, wherein the creator cohort stores the verification record in a creator-cohort verification chain that maintains verification records provided by verification cohorts.~~

~~117. The method of claim 116, wherein the creator-cohort verification chain is a sidechain of the MDC.~~

~~118. A method for maintaining a material data blockchain (MDC), comprising:~~

~~receiving, by a first set of processors that executes a ledger-based application and a creator VDAX that corresponds to a creator cohort, an unnotarized material data block (MDB) from the ledger-based application, wherein the unnotarized MDB includes a metadata portion and a payload portion that contains substantive data being stored in the MDB;~~

~~extracting, by the creator VDAX, a first sequence of bits from the metadata portion of the MDB;~~

~~generating, by the creator VDAX, a first genomic engagement factor (GEF) based on the first sequence, a first genomic differentiation object that is assigned to the creator VDAX, and first genomic regulation instructions (GRI) that are maintained by the creator VDAX;~~

~~applying, by the creator VDAX, one or more computational functions to the unnotarized MDB using the first GEF as input to obtain a creator-generated value corresponding to the unnotarized MBD;~~

~~updating, by the first set of processors, the unnotarized MBD by digitally signing the unnotarized MBD with the creator-generated value;~~

~~providing, by the first set of processors, the unnotarized MDB to one or more notary cohorts;~~

~~receiving, by the first set of processors, a respective notary value from each of the one or more notary cohorts, wherein each respective notary value is generated by a respective notary cohort of the one or more notary cohorts using respective GRI that is held by the respective notary cohort and a respective genomic differentiation object that is assigned to the respective notary cohort;~~

~~updating, by the first set of processors, the unnotarized MDB with the respective notary values received from the one or more notary cohorts to obtain a notarized MDB; and~~

~~adding, by the first set of processors, the notarized MDB to the MDC.~~

~~119. The method of claim 118, further comprising:~~

~~receiving, by a second set of processors that executes a notary VDAX corresponding to a notary cohort of the one or more notary cohorts, the unnotarized MDB;~~

~~extracting, by the notary VDAX, a second sequence of bits from the metadata portion of the MDB;~~

~~generating, by the notary VDAX, a second genomic engagement factor (GEF) based on the second sequence, a second genomic differentiation object that is assigned to the notary VDAX, and second GRI that are maintained by the notary VDAX;~~

~~applying, by the notary VDAX, one or more computational functions to the unnotarized MDB using the second GEF as input to obtain a notary value corresponding to the unnotarized MBD; and~~

~~providing, by the second set of processors, the notary value to the creator cohort.~~

~~120. The method of claim 119, further comprising:~~

~~digitally signing, by the second set of processors, the unnotarized MDB with the notary value to obtain an at least partially notarized MDB; and~~

~~updating, by the second set of processors, a second MDC that is maintained by the notary cohort with the at least partially notarized MDB.~~

~~121. The method of claim 120, wherein providing the notary value to the creator cohort includes providing the at least partially notarized MDB to the creator cohort.~~

~~122. The method of claim 119, wherein generating the second GEF includes determining a sequence conversion vector (SCV) based on the second sequence and the second GRI.~~

~~123. The method of claim 122, wherein generating the second GEF includes:~~

~~mapping the second genomic differentiation object based on the SCV to obtain a mapped genomic differentiation object;~~

~~modifying the mapped genomic differentiation object based on at least one of the SCV and the second GRI to obtain a modified mapped genomic differentiation object; and~~

~~determining the second GEF based on the modified mapped genomic differentiation object.~~

~~124. The method of claim 121, wherein generating the second GEF includes:~~

~~modifying the second genomic differentiation object based on the second GRI to obtain a modified genomic differentiation object;~~

~~mapping the modified genomic differentiation object based on the SCV to obtain a mapped modified genomic differentiation object; and~~

~~determining the second GEF based on the mapped modified genomic differentiation object.~~

~~125. The method of claim 124, wherein the method further includes storing the modified genomic differentiation object, such that the modified genomic differentiation object is used to determine subsequent GEFs for other MDBs received from the creator cohort.~~

~~126. The method of claim 119, wherein the second sequence of bits is equal to the first sequence of bits.~~

~~127. The method of claim 119, wherein the second sequence is different than the first sequence of bits.~~

128. The method of claim 119, wherein the one or more computational functions include one or more cipher-based functions that transform at least a portion of the unnotarized MDB into the notary value using the second GEF as a key.

129. The method of claim 128, wherein the one or more cipher-based functions include an encryption function.

130. The method of claim 128, wherein the one or more cipher-based functions include a disambiguation function.

131. The method of claim 119, wherein the one or more computational functions include one or more cipherless functions that transform at least a portion of the unnotarized MDB into the notary value using the second GEF as an input parameter.

132. The method of claim 131, wherein the one or more cipherless functions include a hash function.

133. The method of claim 119, wherein the one or more computational functions include one or more cipherless functions and one or more cipher-based functions that in combination transform that transform at least a portion of the unnotarized MDB into the notary value using the second GEF as an input parameter.

134. The method of claim 119, wherein the second genomic differentiation object and the first genomic differentiation object are correlated genomic differentiation objects.

135. The method of claim 119, wherein the second genomic differentiation object and the first genomic differentiation object are uncorrelated genomic differentiation objects.

136. The method of claim 119, further comprising:

receiving, by the second set of processors, a confirmation request from a verifying cohort, the confirmation request including the notarized MDB; and

generating, by the notary VDAX, a recreated notary value based on the notarized MDB based on a third sequence of bits that is extracted from the notarized MDB, the second GRI, and the second genomic data object.

137. The method of claim 136, further comprising:

determining, by the second set of processors, whether the recreated notary value matches the notary value used to notarize the notarized MDB; and

in response to determining the recreated notary value matches the notary value used to notarize the notarized MDB, providing, by the second set of processors, a response to the verifying cohort confirming an integrity of the notarized MDB.

138. The method of claim 136, further comprising, providing, by the second set of processors, a response to the verifying cohort indicating the recreated notary value, wherein the verifying cohort confirms an integrity of the notarized MDB in response to determining the recreated notary value matches the notary value used to notarize the notarized MDB.

139. A device comprising:

a VDAX having a genomic data set assigned thereto, the genomic data set corresponding to a digital ecosystem to which the device is a member and including a genomic eligibility object, a genomic correlation object, and a genomic differentiation object, the VDAX comprising:

link spawning means for spawning a first link for a second device in the digital ecosystem, wherein the link includes first genomic regulation instructions (GRI) that are used by the second device to generate virtual binary language script (VBLS) that is decodable by the VDAX;

VBLS decoding means for decoding VBLS generated by the second device for the device based on the first GRI and the genomic differentiation object to obtain decoded digital objects encoded in the VBLS.

140. The device of claim 139, wherein the first link includes encoded GRI that contains the first GRI, encoded link information that includes information that is used by the second device to decode the first GRI from the encoded GRI, and a public sequence that is used by the second device to decode the link information from the encoded link information.

141. The device of claim 140, wherein the link spawning means generates the first link based on the genomic eligibility object, the genomic correlation object, first credentials corresponding to the device, and second credentials corresponding to the second device.

142. The device of claim 141, wherein the link spawning means generates the first link by generating a correlation vector based on the genomic eligibility object assigned to the VDAX, the first credentials, and the second credentials, wherein the correlation vector is used to encode the link information.

~~143. The device of claim 142, wherein the genomic eligibility object is a first CNA object that is uniquely derived from a master CNA object using the first credentials.~~

~~144. The device of claim 143, wherein the link spawning means includes means for generating the correlation vector based on the first CNA object, the first credentials, and the second credentials.~~

~~145. The device of claim 144, wherein the means for generating the correlation vector is configured to:~~

~~apply a code mapping function to the first credentials to obtain a first code word vector that uniquely corresponds to the first device and indicates respective portions of the master CNA object that were allocated to the first CNA object;~~

~~apply the code mapping function to the second credentials to obtain a second code vector corresponding to that uniquely corresponds to the second device and indicates respective portions of the master CNA object that were allocated to a second CNA object that is assigned to the second device;~~

~~determine an intersection vector that indicates an intersection of the first code vector and the second code vector; and~~

~~determine the correlation vector from the first CNA object based on the intersection vector.~~

~~146. The device of claim 144, wherein the correlation vector represents a unique correlation between the first CNA object of the device and the second CNA object of the second device.~~

~~147. The device of claim 142, wherein the genomic eligibility object is a PNA object that is uniquely derived from a master PNA object using the first credentials corresponding to the device, wherein the PNA object includes a secret ratio that is maintained in secret by the device, a public primitive polynomial assigned to the device, and a public component of the master PNA object.~~

~~148. The device of claim 147, wherein the public component of the master PNA object includes a first master primitive polynomial of degree M, a second master primitive polynomial of degree M, and a public component that includes (N-1) respective pairs of M-bit values, wherein the first credentials and the second credentials are represented in N-bit vectors.~~

~~149. The device of claim 147, wherein the link spawning means includes means for generating the correlation vector based on the PNA object of the device and the engagement information of the second device.~~

150. The device of claim 149, wherein the engagement information of the second device includes credentials corresponding to the second device and a second public polynomial assigned to the second device.

151. The device of claim 140, wherein the link spawning means includes:

means for generating a correlation vector based on the genomic eligibility object, first credentials of the device, and second credentials of the second device;

means for generating the first GRI;

means for generating a set of zone references, wherein each zone reference indicates a respective portion of the genomic correlation object;

means for generating a window vector based on the genomic correlation object and the set of zone references;

means for generating the public sequence based on the window and the first GRI;

means for generating transformation value based on the public sequence, the window, and the genomic correlation object of the device;

means for encoding the first GRI using the transformation value to obtain the encoded GRI; and

means for encoding the set of zone references using the correlation vector to obtain the encoded link information.

152. The device of claim 151, wherein the device provides the encoded link information, the public sequence, and the encoded GRI to the second device and the second device decodes the encoded GRI based on the encoded link information, the public sequence, a second genomic data set assigned to the second device, the first credentials, and the second credentials.

153. The device of claim 139, wherein the means for decoding VBLS comprises:

means for generating a genomic engagement factor (GEF) based on the first GRI, a sequence extracted from an unencoded portion of a VBLS object, and the genomic differentiation object;

means for decoding the VBLS object based on the GEF to obtain a decoded portion of a digital object.

154. The device of claim 153, wherein the means for decoding the VBLS object decrypts an encoded portion of the VBLS object using the GEF as a key to obtain a decoded portion of the digital object.

~~155. The device of claim 153, wherein the means for decoding the VBLS object disambiguates an encoded portion of the VBLS object by XORing an unencoded portion of the VBLS object and the GEF to obtain the decoded digital object.~~

~~156. The device of claim 153, wherein the digital object is a media frame.~~

~~157. The device of claim 153, wherein the digital object is a network packet.~~

~~158. The device of claim 153, wherein the digital object is a file.~~

~~159. The device of claim 139, further comprising:~~

~~link hosting means for decoding a second link received from the second device to obtain second GRI generated by the second device, the second link including second encoded GRI, second encoded link information for decoding the encoded GRI, and a second public sequence that is used to decode the second encoded link information;~~

~~VBLS encoding means for encoding digital objects for the second device based on the second GRI and the genomic differentiation object.~~

~~160. The device of claim 159, wherein the link hosting means comprises:~~

~~means for generating a correlation vector based on the genomic eligibility object, first credentials of the device, and second credentials of the second device;~~

~~means for decoding the second encoded link information based on the correlation vector to obtain a set of zone references;~~

~~means for generating a window vector based on the set of zone references and the genomic correlation object;~~

~~means for generating a transformation value based on the public sequence, the window, and the genomic correlation object of the device; and~~

~~means for decoding the second GRI using the transformation value to obtain the second GRI.~~

~~161. The device of claim 160, wherein the VBLS encoding means comprises:~~

~~means for generating a genomic engagement factor (GEF) based on the first GRI, a sequence extracted from an unencoded portion of a digital object, and the genomic differentiation object;~~

~~means for encoding a digital object based on the GEF to obtain a VBLS object that is decodable by the second device.~~

~~162. The device of claim 161, wherein the means for encoding the digital object encrypts a payload of the digital object using the GEF as a key to obtain a VBLS object.~~

~~163. The device of claim 161, wherein the means for encoding the digital object XORs payload of the digital object and the GEF to obtain the VBLS object.~~