IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

	§	
BMC SOFTWARE, INC.,	§	
	§	
Plaintiff,	§	
	§	
v.	§	Civil Action No. 2:14-CV-903-JRG
	§	
SERVICENOW, INC.,	§	
	§	
Defendant.	§	
	§	

MEMORANDUM OPINION AND ORDER

On July 10, 2015, the Court held a hearing to determine the proper construction of the disputed claim terms in United States Patent Nos. 5,978,594 ("the '594 Patent"), 6,816,898 ("the '898 Patent"), 6,895,586 ("the '586 Patent"), 7,062,683 ("the '683 Patent"), 7,617,073 ("the '073 Patent"), 8,646,093 ("the '093 Patent"), and 8,674,992 ("the '992 Patent") (collectively, the "Asserted Patents"). After considering the arguments made by the parties at the hearing and in the parties' claim construction briefing (Dkt. Nos. 99, 106, and 108), the Court issues this Claim Construction Memorandum and Order.

TABLE OF CONTENTS

I. BACKGROUND 4

A. The ’586 Patent.. 4

B. The ’898 Patent.. 6

C. The ’594 Patent.. 8

D. The ’683 Patent.. 11

E. The ’093 Patent.. 14

F. The ’073 Patent.. 16

G. The ’992 Patent.. 18

II. APPLICABLE LAW 21

III. CONSTRUCTION OF AGREED TERMS 23

IV. CONSTRUCTION OF DISPUTED TERMS 25

A. The ’586 Patent.. 25

1. “sharing the plurality of objects with a plurality of the one or more computer system[s]” 25

2. “hierarchical namespace” 29

3. “dynamically inherits traits from the prototype” and “wherein the values of the traits inherited from the prototype change dynamically”............................... 33

4. “traits”.. 37

B. The ’898 Patent.. 42

1. “periodically”... 42

2. “script-based program”... 46

3. “service monitor”.. 49

C. The ’594 Patent.. 55

1. “interpreting the instructions”... 55

2. “interpretable high-level computer programming language” 58

3. “uninterpreted form” and “stored on the storage device in their uninterpreted form” 61

D. The ’683 Patent 65

1. “node” and “nodes” 65

2. “fault model” and “fault model having a plurality of nodes” 70

3. “enterprise” 73

4. “up-stream,” “most up-stream,” and “down-stream” 76

5. “a root cause” 79

6. “impact value” 81

E. The ’093 Patent 84

1. “license certificate” 84

2. “exception indication” 88

F. The ’073 Patent 91

1. “wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit” 91

2. “subcomponent” and “IT subcomponent” 96

3. “IT component processor,” “IT subcomponent processor,” and “processor” 99

G. The ’992 Patent 101

1. “importance” and “importance of the corresponding service” 101

2. “service level agreement (SLA)” and “SLA violation” 105

3. “graph” and “node” 108

4. “variable graphical image,” “a variable graphical image positioned with the node,” “spotlight,” and “displaying a spotlight with each of the nodes of the plurality of nodes” 113

V. CONCLUSION 118

I. BACKGROUND

A. The ’586 Patent

The ’586 Patent is titled “Enterprise Management System and Method which Includes a Common Enterprise-wide Namespace and Prototype-based Hierarchical Inheritance.” It was filed on August 30, 2000, and issued on May 17, 2005. The ’586 Patent generally relates to an improved namespace and object description system for enterprise management. *See* ’586 Patent at Abstract.[1]

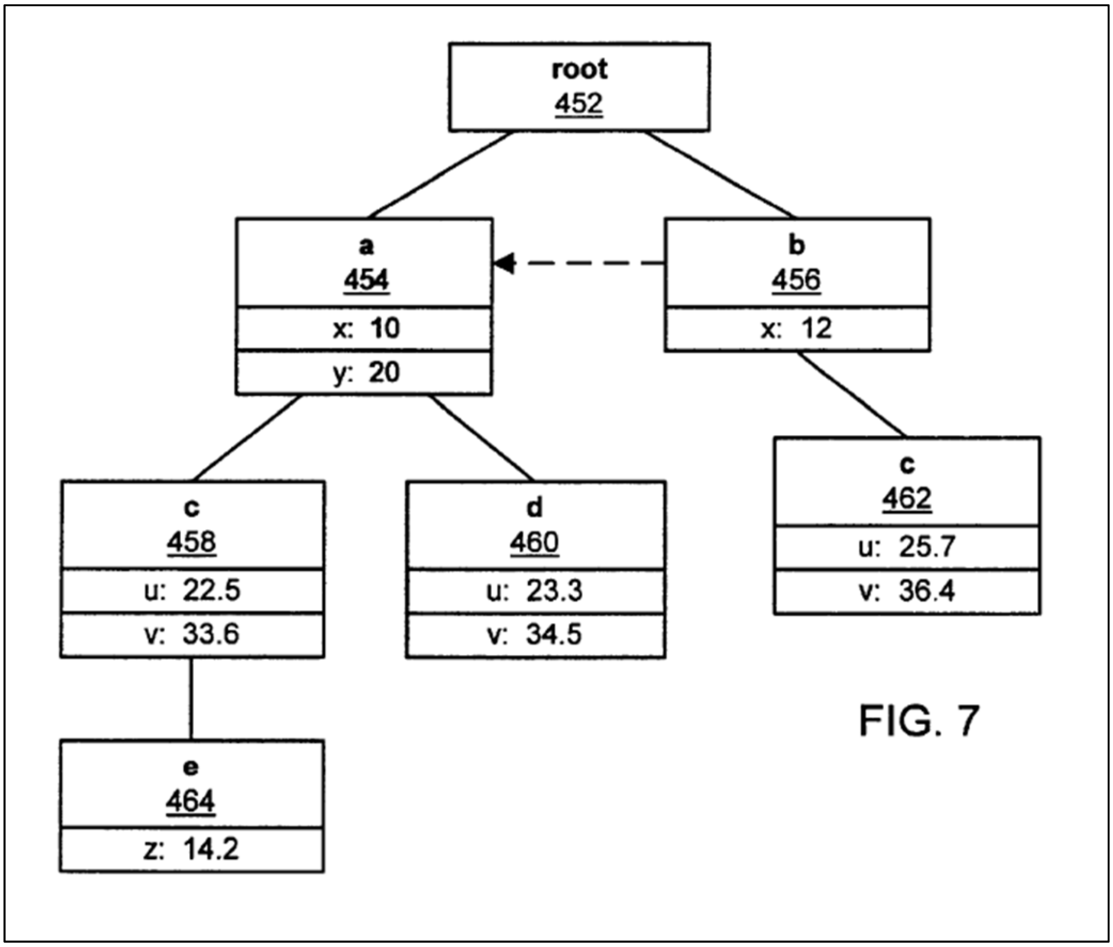
The specification states that “the term ‘namespace’ generally refers to a set of names in which all names are unique,” and that “[a] namespace is typically a logical organization and not a physical one.” *Id.* at 1:52–54; 2:12–13. The specification describes an embodiment where “[t]he namespace comprises a logical arrangement of the objects, stored hierarchically.” *Id.* at 3:62–63. The specification states that “a plurality of objects may be added to the namespace, wherein the objects relate to software and hardware of the one or more computer systems.” *Id.* at 3:63–65. The specification adds that “at least one of the objects is a prototype and at least one of the objects is an instance.” *Id.* at 4:8–10.

The specification defines “prototype” as “an object in a namespace from which attributes, values, and/or children are dynamically inherited by another object.” *Id.* at 14:44–46. The specification further defines “instance” as “an object in a namespace which dynamically inherits attributes, values, and/or children from another object in the namespace.” *Id.* at 14:47–49. The specification states that “[t]he instance inherits from the prototype traits such as attribute values

[1] The Abstract of the ’586 Patent follows:

> A system and method for providing an improved namespace and object description system for enterprise management are disclosed. The system and method employ a hierarchical namespace with objects including prototypes and instances where an instance inherits traits from a prototype, such as attribute values and/or child objects.

and/or child objects." *Id.* at 4:8–10. The specification states that Figure 7 illustrates an example of a namespace which includes a prototype-instance relationship. *Id.* at 14:49–51.



Id. at Figure 7. The specification states that Figure 7 illustrates a "dynamic inheritance link from object 'b' 456 to object 'a' 454; the link is shown as a dashed arrow." *Id.* at 14:53–54. The specification further states that "[o]bject 'a' 454 functions as the prototype and object 'b' 456 functions as the instance." *Id.* at 14:54-55. The specification concludes that "object 'b' 456 dynamically inherits the attributes, values, and children of object 'a' 454." *Id.* at 4:55–57. For example, "object 'b' 456 has an attribute called 'x' of its own and also inherits the attribute 'y' from object 'a' 454." *Id.* at 14:60–62.

Claim 1 of the '586 Patent is representative of the asserted claims and recites the following elements (disputed terms in italics):

> 1. A method for managing an enterprise, wherein the enterprise comprises one or more networked computer systems, the method comprising:
>
> providing a *hierarchical namespace*;
>
> adding a plurality of objects to the namespace, wherein the objects relate to software and hardware of the one or more computer systems;
>
> *sharing the plurality of objects with a plurality of the one or more computer system*, wherein at least one of the objects is a prototype and at least one of the objects is an instance, wherein the instance *dynamically inherits traits from the prototype*; and *wherein the values of the traits inherited from the prototype change dynamically*.

B. The ’898 Patent

The ’898 Patent is titled “Interfacing External Metrics into a Performance Management System.” It was filed on August 16, 2000, and issued on November 9, 2004. The ’898 Patent generally relates to performing operations on performance management data, and generating output data for display using collected performance management data. *See* ’898 Patent at Abstract.[2]

The specification describes Figure 3 as illustrating “a data flow diagram of the claimed invention.” *Id.* at 7:13–14.

[2] The Abstract of the ’898 Patent follows:

> A method and apparatus for network management is described. In one embodiment, a method comprises collecting performance data having accompanying meta data including information defining the performance management data and information indicating operations to be performed on the performance management data, and generating output data for display using collected performance management data according to the information indicating the operations to be performed on the performance management data.

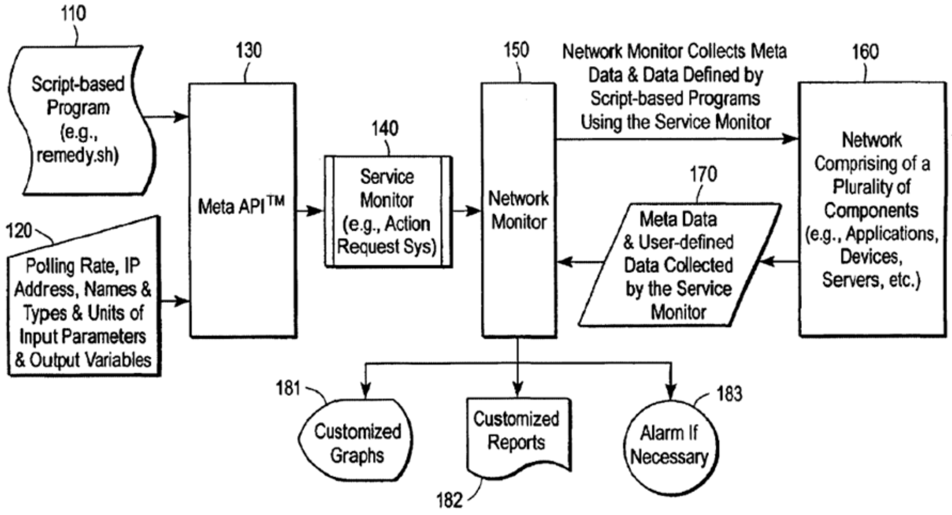

110
Script-based Program (e.g., remedy.sh)
120
Polling Rate, IP Address, Names & Types & Units of Input Parameters & Output Variables
130
Meta API™
140
Service Monitor (e.g., Action Request Sys)
150
Network Monitor
Network Monitor Collects Meta Data & Data Defined by Script-based Programs Using the Service Monitor
160
Network Comprising of a Plurality of Components (e.g., Applications, Devices, Servers, etc.)
170
Meta Data & User-defined Data Collected by the Service Monitor
181
Customized Graphs
182
Customized Reports
183
Alarm If Necessary

FIG. 3

Id. at Figure 3. Referring to Figure 3, the specification states that "a user provides at least one script-based program [110] to the meta API 130." *Id.* at 7:14–15. The specification further states that "[i]n one embodiment, the user provides the script-based program by copying the script into a directory server on a server used by the network managing system (e.g., network monitor 150)." *Id.* at 7:15–18. The specification adds that "[t]he user may also provide information 120 to the meta API." *Id.* at 7:22–23. The specification further discloses that "[i]nformation 120 may comprise poling [sic] rate, IP address, names and types, and units of input and output variables." *Id.* at 7:23–24. "In other words, information 120 comprise user defined customized data types." *Id.* at 7:24–26.

The specification further states that "[n]etwork monitor 150 collects meta data and data defined by the script-based programs from the network 160 using service monitor 140." *Id.* at 7:28–30. The specification states that "[t]he returned data 170 is then processed by network monitor 150." *Id.* at 7:30–31. The specification adds that "[t]he processing by network monitor 150 may include generated customized graphs 181, customized records 182, and/or setting an alarm 183." *Id.* at 7:31–34.

Claim 6 of the '898 Patent is representative of the asserted claims and recites the following elements (disputed terms in italics):

> 6. A method for providing an interface between a user and a performance management system, the performance management system being connected with a network, the network including a plurality of components coupled by a plurality of connections, the performance management system collecting data of the components, the method comprising:
>
> receiving at least one *script-based program* from the user, the *script-based programs* defining data types not provided by the performance management system;
>
> integrating the program to the performance management system as a *service monitor*, the performance management system using the *service monitor* to *periodically* collect data of the defined data types from the components.

C. The '594 Patent

The '594 Patent is titled "System for Managing Computer Resources Across a Distributed Computing Environment by First Reading Discovery Information about How to Determine System Resources Presence." It was filed on March 6, 1997, and issued on November 2, 1999. The '594 Patent generally relates to method and apparatus for managing a computer network. *See* '594 Patent at Abstract.[3]

[3] The Abstract of the '594 Patent follows:

> A method and apparatus are disclosed for managing a computer network. A manager software system is installed on a network management computer system within the network, and one agent software system is installed on each of the server computer systems in the network. A knowledge module in the form of a text fie [sic] is stored on the network manager computer system so that the manager software system can transmit knowledge to the various agent software systems throughout the network, for use by the agents in monitoring and managing the server on which they are installed. Interpretable script language programs are present on all computers in the network, expanding and customizing the functionality of the agent software systems. A method is disclosed for using the high level interpretable script language programs in connection with the agent

The Background section of the specification states that a need existed "for a network management system that [would] provide an increase in automation and efficiency for network management and a decrease in the complexity of such management." *Id.* at 1:56–58. The specification states that "FIG. 8 shows a preferred procedure, implemented according to the method of the invention, for discovering resources on a server computer system 14 in the network using a high-level interpretable language." *Id.* at 7:45–48.

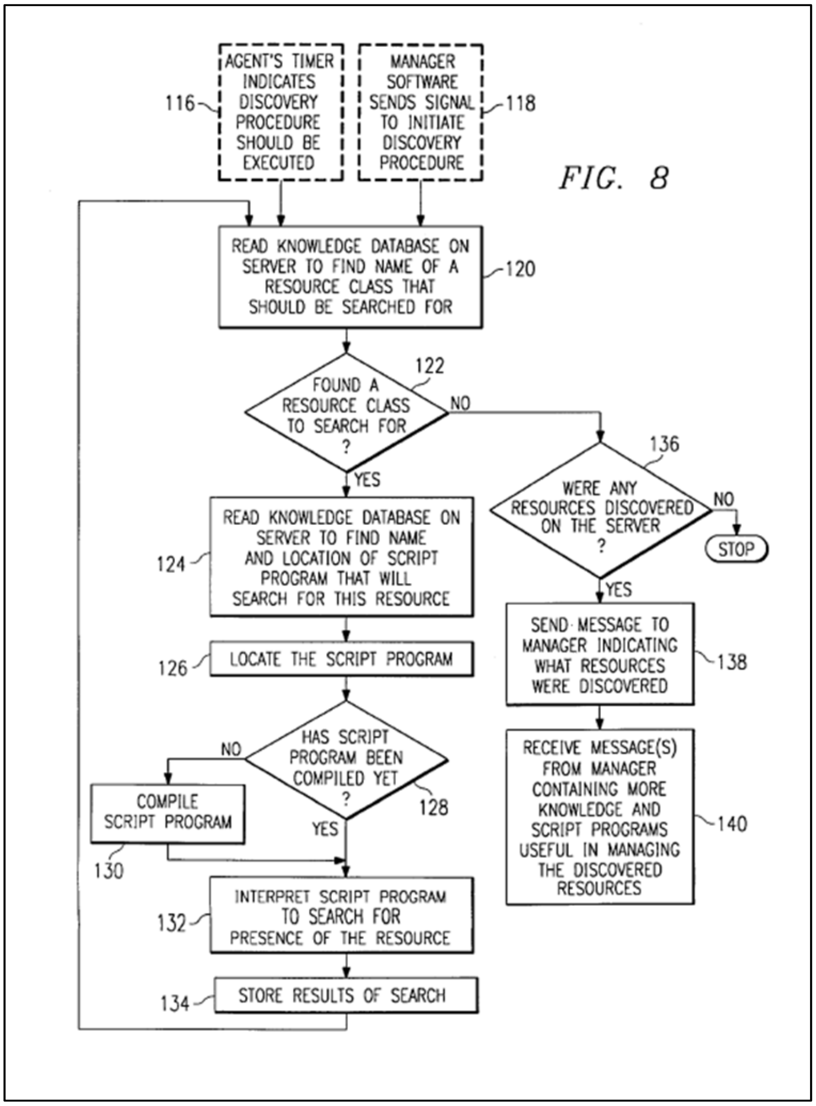


FIG. 8
AGENT'S TIMER INDICATES DISCOVERY PROCEDURE SHOULD BE EXECUTED
116
MANAGER SOFTWARE SENDS SIGNAL TO INITIATE DISCOVERY PROCEDURE
118
READ KNOWLEDGE DATABASE ON SERVER TO FIND NAME OF A RESOURCE CLASS THAT SHOULD BE SEARCHED FOR
120
FOUND A RESOURCE CLASS TO SEARCH FOR ?
122
NO
YES
WERE ANY RESOURCES DISCOVERED ON THE SERVER ?
136
NO
STOP
YES
READ KNOWLEDGE DATABASE ON SERVER TO FIND NAME AND LOCATION OF SCRIPT PROGRAM THAT WILL SEARCH FOR THIS RESOURCE
124
LOCATE THE SCRIPT PROGRAM
126
SEND MESSAGE TO MANAGER INDICATING WHAT RESOURCES WERE DISCOVERED
138
HAS SCRIPT PROGRAM BEEN COMPILED YET ?
128
NO
YES
COMPILE SCRIPT PROGRAM
130
RECEIVE MESSAGE(S) FROM MANAGER CONTAINING MORE KNOWLEDGE AND SCRIPT PROGRAMS USEFUL IN MANAGING THE DISCOVERED RESOURCES
140
INTERPRET SCRIPT PROGRAM TO SEARCH FOR PRESENCE OF THE RESOURCE
132
STORE RESULTS OF SEARCH
134

software systems for discovering resources on the network, monitoring aspects of resources, and taking recovery actions automatically in the event of an alarm condition.

Id. at Figure 8. The specification discloses that "[t]he discovery procedure is initiated either in step 116 when the timer within agent software system 36 indicates that a discovery procedure stored in run queue 71 is ready to be executed, or in step 118 when manager software system 34 sends a message to agent software system 36 indicating that a discovery procedure should be executed." *Id.* at 7:48–53. The specification adds that "[w]hen the discovery procedure begins, in step 120, the agent software system 36 reads knowledge database 75 to determine the name of a resource class that should be searched for." *Id.* at 7:56–58. The specification states that "[i]n step 122, if a resource class is found that should be searched for, execution continues with step 124." *Id.* at 7:58–59.

The specification continues that "[i]n step 124, the knowledge database on the server is read to find the name and location of the script program that will search for the particular resource in question." *Id.* at 7:60–62. The specification states that "[i]n step 126, the script program indicated is found," and "[i]n step 128, agent software system 36 determines whether or not the script program has yet been compiled." *Id.* at 7:62–65. The specification further discloses that if the script program has not been compiled, "script program compiler 64 compiles the script program in step 130 and execution continues with step 132, in which the script program is interpreted, thereby searching for the presence of the resource in question." *Id.* at 7:65–8:2. The specification further states that "[t]he results of the search are stored in step 134, and the process continues at step 120 once again until in step 122 no further resources are found to be searched for, in which case execution continues with step 136." *Id.* at 8:2–6.

Claim 1 of the '594 Patent is representative of the asserted claims and recites the following elements (disputed terms in italics):

> 1. A method of determining whether a resource is present on a computer system, comprising the steps of:

> (a) reading, from a storage device coupled to the computer system, discovery information about how to determine whether the resource is present on the computer system;
> (b) finding, on the storage device, instructions that are referred to in the discovery information, that are written in an *interpretable high-level computer programming language*, and that are *stored on the storage device in their uninterpreted form*;
> (c) *interpreting the instructions* for the purpose of collecting data for use in determining whether the resource is present on the computer system; and
> (d) determining, responsive to the collected data, whether the resource is present on the computer system.

D. The ’683 Patent

The ’683 Patent is titled “Two-phase Root Cause Analysis.” It was filed on April 22, 2003, and issued on June 13, 2006. The ’683 Patent generally relates to a two-phase method to perform root-cause analysis over an enterprise-specific fault model. *See* ’683 Patent at Abstract.[4]

The specification states that Figure 1 is a flowchart that illustrates an enterprise monitoring and analysis method in accordance with one embodiment of the invention.

[4] The Abstract of the ’683 Patent follows:

> A two-phase method to perform root-cause analysis over an enterprise-specific fault model is described. In the first phase, an up-stream analysis is performed (beginning at a node generating an alarm event) to identify one or more nodes that may be in failure. In the second phase, a down-stream analysis is performed to identify those nodes in the enterprise whose operational condition are impacted by the prior determined failed nodes. Nodes identified as failed as a result of the up-stream analysis may be reported to a user as failed. Nodes identifies [sic] as impacted as a result of the down-stream analysis may be reported to a user as impacted and, beneficially, any failure alarms associated with those impacted nodes may be masked. Up-stream (phase 1) analysis is driven by inference policies associated with various nodes in the enterprise’s fault model. An inference policy is a rule, or set of rules, for inferring the status or condition of a fault model node based on the status or condition of the node’s immediately down-stream neighboring nodes. Similarly, down-stream (phase 2) analysis is driven by impact policies associated with various nodes in the enterprise’s fault model. An impact policy is a rule, or set of rules, for assessing the impact on a fault model node based on the status or condition of the node’s immediately up-stream neighboring nodes.

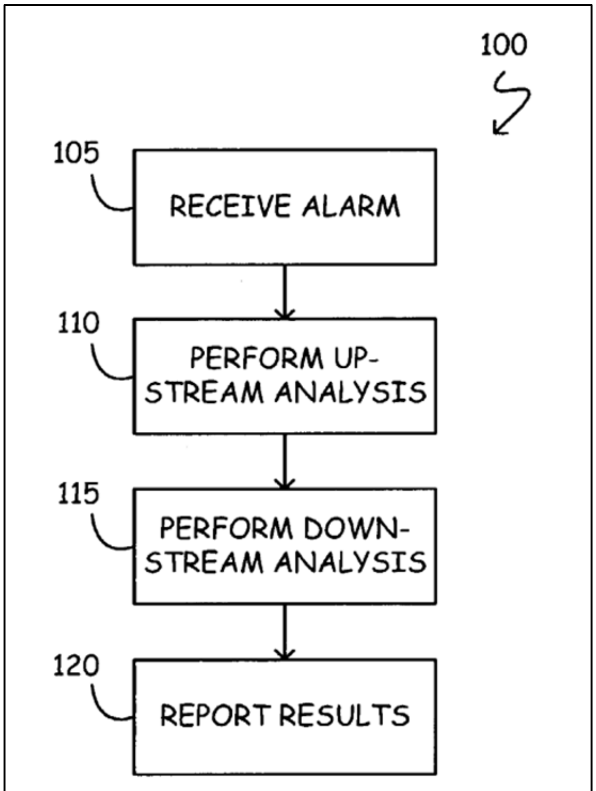


FIG. 1

’683 Patent at Figure 1. Referring to FIG. 1, the specification discloses “a model based reasoning (MBR) approach 100 to enterprise monitoring and fault analysis in accordance with the invention uses a combination of up-stream analysis (based on the evaluation of inference policies) and down-stream analysis (based on the evaluation of impact policies) on an Impact Graph to efficiently and effectively identify and isolate root cause faults from the myriad of event notifications or alarms, many or most of which may be ‘sympathetic,’ that one or more underlying fault conditions may trigger.” *Id.* at 4:31–40. The specification adds that “[o]n event notification (block 105), an up-stream analysis of the Impact Graph beginning with the node receiving the event notification is performed (block 110).” *Id.* at 4:40–43. The specification states that an “[u]p-stream analysis in accordance with block 110 may modify the status value of zero or more nodes in the enterprise’s Impact Graph up-stream from the node receiving the event notification." *Id.* at 4:43–46. The specification describes the next step as "the furthest up-stream node (relative to the node receiving the initial event notification) whose status value was modified in accordance with block 110 is selected as a starting point from which a down-stream analysis is performed (block 115)." *Id.* at 4:46–50. The specification further states that "[d]own-stream analysis in accordance with block 115 may modify the impact value of zero or more nodes in the enterprise's Impact Graph down-stream from the down-stream analysis' starting node." *Id.* at 4:50–53.

The specification adds that "[o]ne of ordinary skill in the art will recognize that if there is more than one node equally distant from the node receiving the event notification and which has had its status value modified in accordance with block 110, an arbitrary one of these nodes may be selected to begin the down-stream analysis in accordance with block 115." *Id.* at 4:53–59. The specification further states that "[w]ith up-stream and down-stream analysis completed enterprise status, including identification of one or more root-cause failures and identification of sympathetic event notifications, may be reported (block 120)." *Id.* at 4:60–63. The specification concludes that "those furthest up-stream nodes in the Impact Graph having a status value indicative of failure are identified as 'root causes.'" *Id.* at 4:63–65.

Claim 1 of the '683 Patent is representative of the asserted claims and recites the following elements (disputed terms in italics):

> 1. An enterprise fault analysis method, wherein at least a portion of the *enterprise* is represented by a *enterprise*-specific *fault model having a plurality of nodes*, comprising:
>
> receiving an event notification for a first *node* in the *fault model*;
>
> performing an *up-stream* analysis of the *fault model* beginning at the first *node*;
>
> identifying a second *node*, the second *node* having a *status value* modified during the *up-stream* analysis to indicate a failed status;

> performing a *down-stream* analysis of the *fault model* beginning at the second *node*;
> identifying those *nodes* in a contiguous path between the second *node* and the first *node* in the *fault model* whose *impact values* indicate an impacted performance condition in accordance with the *down-stream* analysis;
> reporting the second *node* as *a root cause* of the received event notification; and
> reporting at least one of the identified *nodes* as impacted by the root cause of the received event notification and not as root causes of the received event notification.

E. The ’093 Patent

The ’093 Patent is titled “Method and System for Configuration Management Database Software License Compliance.” It was filed on December 9, 2009, and issued on February 4, 2014. The ’093 Patent generally relates to a software license engine that allows an enterprise to model software license contracts and evaluate deployment of software for compliance with the software license contracts. *See* ’093 Patent at Abstract.[5]

The specification states that Figure 2 is “a block diagram illustrating a system 200 according to one embodiment with a Configuration Management Database (“CMDB”) server 110 and a pair of clients 210 and 220.” *Id.* at 3:50–52.

[5] The Abstract of the ’093 Patent follows:

> A software license engine allows an enterprise to model software license contracts and evaluate deployment of software for compliance with the software license contracts. Deployment of software products in the enterprise is modeled in a configuration management database. The software license engine maintains a license database for connecting software license contracts with software deployment modeled by the configuration management database. Users of the software license engine may use license types that are predefined in the software license engine or may define custom license types. The software license engine may indicate compliance or non-compliance with the software license contracts.

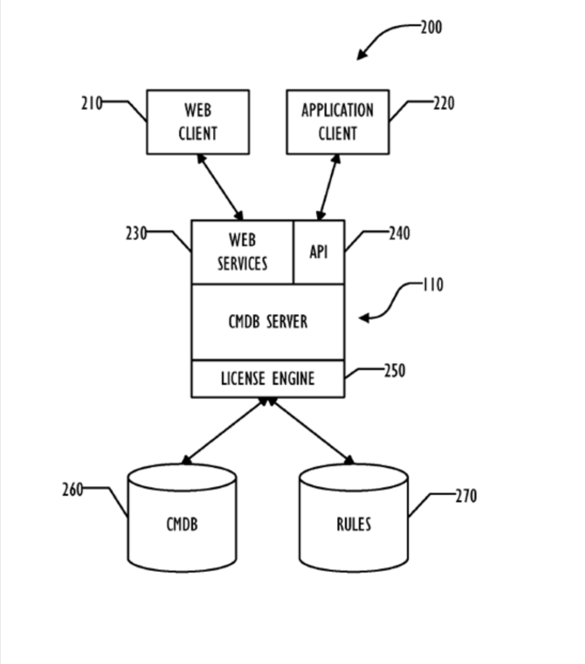


FIG. 2

Id. at Figure 2. The specification states that "[t]he CMDB server 110 may comprises [sic] a number of software components, including a web services component 230 for interacting with a web client computer 210, and an Application Programming Interface (API) 240 for interacting with an application client computer 220." *Id.* at 3:53–57. The specification further states that "[t]he application client computer 220 may be a computer running any application designed to interact with the CMDB server 110 through the API, including, for example, a desktop computer with a CMDB client application . . . that provides a graphical user interface (GUI) to the user of the client computer 220." *Id.* at 3:57–65. The specification adds that "[t]he CMDB server 110 also comprises a license engine 250" and "other software components for providing CMDB functionality as desired." *Id.* at 3:66–4:3.

The specification continues that "[d]ata for the CMDB server 110 is illustrated as stored in a CMDB datastore 260 and a license datastore 270." *Id.* at 4:4–5. The specification states that "[t]he CMDB datastore 260 comprises the storage for the conventional CMDB data, including

CIs [Configuration Item].” *Id.* at 4:5–7. The specification further states that “[t]he license datastore 270 provides storage for to model software contracts, including rules against which the CIs are evaluated for software license compliance and other information necessary for processing those rules.” *Id.* at 4:14–17.

Claim 1 of the ‘093 Patent is representative of the asserted claims and recites the following elements (disputed terms in italics):

> 1. A computer-implemented method, comprising:
> modeling deployment of a software product and a software license contract for the software product;
> storing a first model of the modeled deployment of the software product in a configuration management database (CMDB) by storing information related to the software product as a first configuration item in the CMDB and by storing information related to the software license contract as a second configuration item in the CMDB;
> storing a second model of the modeled software license contract for the software product in a license database by generating a *license certificate* corresponding to the software license contract and storing the *license certificate* in the license database; and
> evaluating the deployment of the software product for compliance with the software license contract, comprising: connecting and comparing the first model and the second model by comparing the first configuration item with the *license certificate* and connecting the *license certificate* with the second configuration item responsive to comparing the first configuration item with the *license certificate*; and generating *an exception indication* if the act of comparing the first model and the second model indicates non-compliance with the software license contract.

F. The ’073 Patent

The ’073 Patent is titled “System and Method for Assessing and Indicating the Health of Components.” It was filed on February 28, 2003, and issued on November 10, 2009. The ’073 Patent generally relates to a system and method for visualization of the components of an enterprise system and the rendering of information about the health or status of the enterprise system. *See* ’073 Patent at Abstract.[6]

In the Summary of the Invention section, the specification states that “[t]he invention comprises using a combination of color codes or other indicators and a combination of algorithms and/or rules-based systems to control the computation of status/severities to associate to components and setup the color codes and indicators.” *Id.* at 2:57–61. The specification adds that “[t]he invention remedies the disadvantages of using a single color code or indicator for providing feedback on the health/status or components in a complex Enterprise System.” *Id.* at 2:64–67. The specification describes Figure 1 as a preferred embodiment. *Id.* at 3:65–67.

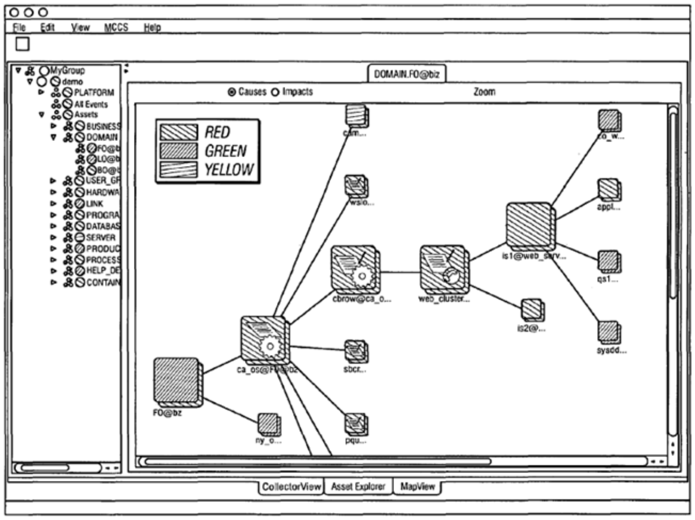


FIG. 1

Id. at Figure 1. The specification states that Figure 1 “illustrates the use of two color indicators in a tree presentation.” *Id.* at 3:66–67. The specification adds that Figure 1 “shows a

[6] The Abstract of the ’073 Patent follows:

> A system and method for visualization of the components of an enterprise system and the rendering of information about the health or status of the enterprise system, its components, and/or its subcomponents. The invention uses a combination of color codes or other indicators and a combination of algorithms and/or rules-based systems to control the computation of status/severities to associate to components and setup the color codes and indicators.

representative interface showing multiple indicators in use." *Id.* at 3:67–4:1. The specification states that "a green indicator is used when a component is healthy and a red one when it is not." *Id.* at 4:12–13. The specification states that "[a]s shown, component FO@biz is composed of components ca_os@FO@biz and ny_os@FO@biz." *Id.* at 4:13–15. The specification further states that "[c]omponent FO@biz contains a plurality of indicators, namely a green indicator in the foreground and a red indicator in the background." *Id.* at 4:15–17. The specification concludes that "a user can assess that component FO@biz is healthy while an underlying component is not healthy." *Id.* at 4:17–19.

Claim 1 of the '073 Patent is representative of the asserted claims and recites the following elements (disputed terms in italics):

> 1. A system for indicating the health status of an IT component and at least one *IT subcomponent* comprising:
> an *IT component processor* adapted to compute a component health status of the IT component;
> an *IT subcomponent processor* adapted to compute a *subcomponent* health status for the at least one *IT subcomponent*; and
> a renderer adapted to *display the health status of the IT component by showing a first indicator for the IT component and a second indicator for the at least one IT subcomponent*, wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit.

G. The '992 Patent

The '992 Patent is titled "Spotlight Graphs." It was filed on July 14, 2010, and issued on March 18, 2014. The '992 Patent generally relates to using a spotlight to indicate multiple attributes or states of an object represented by a node in a graph. *See* '992 Patent at Abstract.[7]

[7] The Abstract of the '992 Patent follows:

> In a computer-displayed graph, indications of multiple attributes or states of an object represented by a node of the graph are displayed using a spotlight, in which

The specification states that "FIG. 2 is a block diagram of a service model graph 200 using spotlights according to one embodiment." *Id.* at 3:39–40.

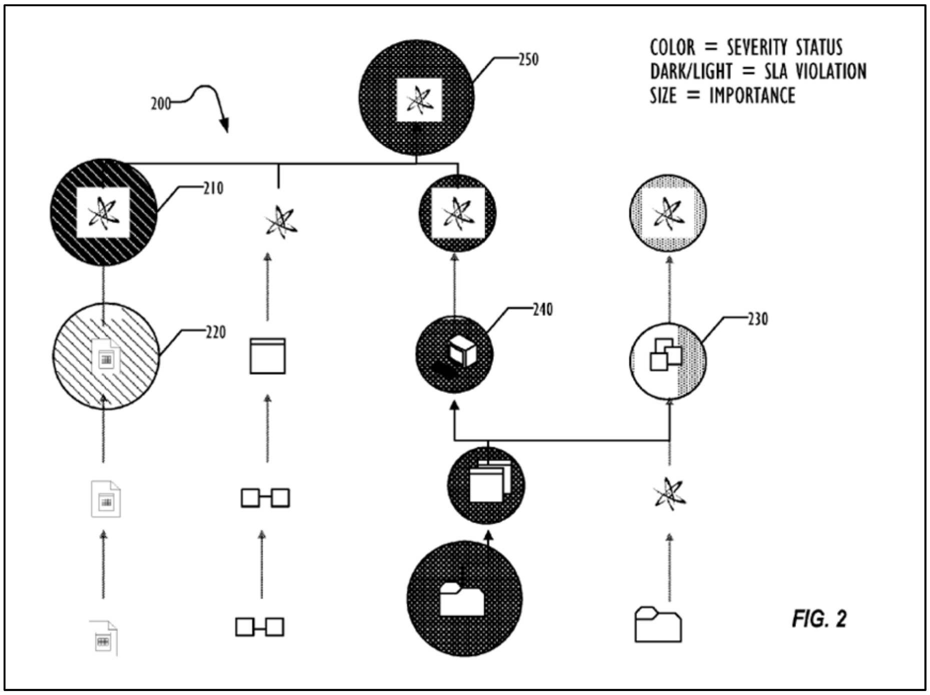


FIG. 2

Id. at Figure 2. The specification states that "[i]n this embodiment, this spotlight [*e.g.*, 220, 210, 250, 230] may vary in three dimensions: color, size, and brightness." *Id.* at 3:40–42. The specification adds that "[t]hese three dimensions may encode three metrics on an object and remove the need for three indicator icons." *Id.* at 3:42–44. The specification further states that "[t]he spotlights also have the advantage of being easily visible from a much longer distance," and that "[t]heir meaning is much easier to understand . . . because of the simple graphical encoding of the data." *Id.* at 3:44–48. The specification also states that "[t]he encoding may eliminate or reduce the need to memorize and compare the individual metrics icons, although in some embodiments, one or more additional icons may still be associated with the graph nodes."

attributes of the spotlight correspond to attributes of the object represented by the node. The attributes of the spotlight each correspond to an attribute of the object and may include the color, brightness, and size of the spotlight. The spotlight may be positioned with the node, including overlaying the spotlight on the node and positioning the spotlight relative to the node.

Id. at 3:48–51. The specification concludes that "[t]he spotlights make establishing a mental ranking of the importance of each object very easy to do, simply by looking at the relative size, color, and brightness of the spotlights behind each monitored object." *Id.* at 3:53–56.

Referring to Figure 2, the specification states that "the color of the spotlight indicates a severity status associated with the corresponding node, as illustrated with–shaded spotlights indicating a higher severity status than diagonal line shaded spotlights." *Id.* at 3:59–62. The specification discloses that "[t]he dot-shaded spotlights 230, 240, and 250 may be implemented using red, while the diagonal line shaded spotlights 210 and 220 may be implemented using yellow." *Id.* at 3:62–65. The specification adds that "[t]he spotlight may be dark or light depending upon whether an SLA violation has occurred at the corresponding node." *Id.* at 3:65–67. The specification concludes that "[t]he size of the spotlight may correspond to an importance of the node, as illustrated, with more important nodes having a larger spotlight than less important nodes." *Id.* at 3:67–4:3.

Claim 1 of the '992 Patent is representative of the asserted claims and recites the following elements (disputed terms in italics):

> 1. A method, comprising:
> displaying a *graph* on a display screen, the *graph* including a plurality of *nodes*, each of the plurality of *nodes* representing a service of a plurality of services;
> determining a metric for each of a plurality of attributes associated with a *service level agreement (SLA)* for each of the plurality of services, the plurality of attributes including at least one *SLA violation*, a severity of the incident causing the *SLA violation* and an *importance of the corresponding service*; and
> *displaying a spotlight with each of the nodes of the plurality of nodes*, the *spotlight* including a plurality of characteristics, each of the plurality of characteristics corresponding to one of the attributes of the service of the plurality of services represented by the *node*, the displayed *spotlight* being graphically varied based on the determined metric

> such that, a size of the *spotlight* varies based on the *importance of the corresponding service*, and a color of the *spotlight* varies based on the severity of the incident causing the *SLA violation*.

II. APPLICABLE LAW

A. Claim Construction

"It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention to which the patentee is entitled the right to exclude.'" *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (quoting *Innova/Pure Water Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)). To determine the meaning of the claims, courts start by considering the intrinsic evidence. *See id.* at 1313. *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 861 (Fed. Cir. 2004); *Bell Atl. Network Servs., Inc. v. Covad Commc'ns Group, Inc.*, 262 F.3d 1258, 1267 (Fed. Cir. 2001). The intrinsic evidence includes the claims themselves, the specification, and the prosecution history. *See Phillips*, 415 F.3d at 1314; *C.R. Bard, Inc.*, 388 F.3d at 861. Courts give claim terms their ordinary and accustomed meaning as understood by one of ordinary skill in the art at the time of the invention in the context of the entire patent. *Phillips*, 415 F.3d at 1312–13; *Alloc, Inc. v. Int'l Trade Comm'n*, 342 F.3d 1361, 1368 (Fed. Cir. 2003).

The claims themselves provide substantial guidance in determining the meaning of particular claim terms. *Phillips*, 415 F.3d at 1314. First, a term's context in the asserted claim can be very instructive. *Id*. Other asserted or unasserted claims can also aid in determining the claim's meaning because claim terms are typically used consistently throughout the patent. *Id*. Differences among the claim terms can also assist in understanding a term's meaning. *Id*. For example, when a dependent claim adds a limitation to an independent claim, it is presumed that the independent claim does not include the limitation. *Id*. at 1314–15.

"[C]laims 'must be read in view of the specification, of which they are a part.'" *Id.* (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc)). "[T]he specification 'is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.'" *Id.* (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)); *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed. Cir. 2002). This is true because a patentee may define his own terms, give a claim term a different meaning than the term would otherwise possess, or disclaim or disavow the claim scope. *Phillips*, 415 F.3d at 1316. In these situations, the inventor's lexicography governs. *Id*. The specification may also resolve ambiguous claim terms "where the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone." *Teleflex, Inc.*, 299 F.3d at 1325. But, "'[a]lthough the specification may aid the court in interpreting the meaning of disputed claim language, particular embodiments and examples appearing in the specification will not generally be read into the claims.'" *Comark Commc'ns, Inc. v. Harris Corp.*, 156 F.3d 1182, 1187 (Fed. Cir. 1998) (quoting *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571 (Fed. Cir. 1988)); *see also Phillips*, 415 F.3d at 1323. The prosecution history is another tool to supply the proper context for claim construction because a patent applicant may also define a term in prosecuting the patent. *Home Diagnostics, Inc., v. Lifescan, Inc.*, 381 F.3d 1352, 1356 (Fed. Cir. 2004) ("As in the case of the specification, a patent applicant may define a term in prosecuting a patent.").

Although extrinsic evidence can be useful, it is "'less significant than the intrinsic record in determining the legally operative meaning of claim language.'" *Phillips*, 415 F.3d at 1317 (quoting *C.R. Bard, Inc.*, 388 F.3d at 862). Technical dictionaries and treatises may help a court understand the underlying technology and the manner in which one skilled in the art might use claim terms, but technical dictionaries and treatises may provide definitions that are too broad or may not be indicative of how the term is used in the patent. *Id*. at 1318. Similarly, expert testimony may aid a court in understanding the underlying technology and determining the particular meaning of a term in the pertinent field, but an expert's conclusory, unsupported assertions as to a term's definition are entirely unhelpful to a court. *Id*. Generally, extrinsic evidence is "less reliable than the patent and its prosecution history in determining how to read claim terms." *Id*.

B. Construction Indefiniteness

Patent claims must particularly point out and distinctly claim the subject matter regarded as the invention. 35 U.S.C. § 112(b). Whether a claim meets this definiteness requirement is a matter of law. *Young v. Lumenis, Inc.*, 492 F.3d 1336, 1344 (Fed. Cir. 2007). A party challenging the definiteness of a claim must show it is invalid by clear and convincing evidence. *Takeda Pharm. Co. v. Zydus Pharms. USA, Inc.*, 743 F.3d 1359, 1368 (Fed. Cir.2014). The ultimate issue is whether someone working in the relevant technical field could understand the bounds of a claim. *Haemonetics Corp. v. Baxter Healthcare Corp.*, 607 F.3d 776, 783 (Fed. Cir. 2010). Specifically, "[a] patent is invalid for indefiniteness if its claims, read in light of the specification delineating the patent, and the prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention." *Nautilus Inc. v. Biosig Instruments, Inc.*, 134 S. Ct. 2120, 2124 (2014).

III. CONSTRUCTION OF AGREED TERMS

The parties have agreed to the construction of the following terms:

Claim Term/Phrase	**Agreed Construction**
"computer system"	Plain meaning.

(‘586 Patent, Asserted Claims 1, 7)	(Dkt. No. 109-3 at 1)
“prototype” (‘586 Patent, Asserted Claim 1)	“an object in a namespace from which attributes, values, and/or children are dynamically inherited by another object” (Dkt. No. 109-3 at 2)
“instance” (‘586 Patent, Asserted Claim 1)	“an object in a namespace which dynamically inherits attributes, values, and/or children from another object in the namespace” (Dkt. No. 109-3 at 2)
“object(s)” (‘586 Patent, Asserted Claims 1, 4)	“self-contained entity that contains data and/or procedures to manipulate the data” (Dkt. No. 109-3 at 1)
“meta data” (‘898 Patent, Asserted Claim 1)	“data about other data” (Dkt. No. 109-2 at 1)
“resource” (‘594 Patent, Asserted Claim 1)	The word “resource” is “intended in its broad sense to include, without limitation, hardware such as computers, printers, memory or other network devices, applications such as database management systems, and logical devices such as logical disk drives or filing systems.” (Dkt. No. 109-1 at 1)
“computer system” (‘594 Patent, Asserted Claim 1)	Plain meaning. (Dkt. No. 109-1 at 1)
“discovery information” (‘594 Patent, Asserted Claim 1)	“information about how to determine whether a resource is present on a computer system” (Dkt. No. 109-1 at 1)
“status value” (‘683 Patent, Asserted Claims 1, 3, 24, 26, 56, 58, 79, 88)	“value indicating the status or condition of a node” (Dkt. No. 109-4 at 2)
“**generating a graphical display**” (‘683 Patent, Asserted Claims 80, 85)	“generating graphical information for display on a display device” (Dkt. No. 109-4 at 15)
“enterprise fault analysis” (‘683 Patent, Asserted Claims 1, 24)	“fault analysis in an enterprise” (Dkt. No. 109-4 at 1)
“Impact Graph” (‘683 Patent, Asserted Claims 2, 25,	“topology or architecture of a specific fault model”

57)	(Dkt. No. 109-4 at 3)
“inference policy” (‘683 Patent, Asserted Claims 3, 26, 58, 88)	“rule, or set of rules, for inferring the status or condition of a fault model node based on the status or condition of the node’s immediately down-stream neighboring nodes” (Dkt. No. 109-4 at 3)
“impact policy” (‘683 Patent, Asserted Claims 14, 37, 69, 89)	“rule, or set of rules, for assessing the impact on a fault model node based on the status or condition of the node’s immediately up-stream neighboring nodes” (Dkt. No. 109-4 at 4)
“east” (‘683 Patent, Asserted Claim 22)	“least” (Dkt. No. 109-4 at 5)
“mat” (‘683 Patent, Asserted Claim 24)	“root” (Dkt. No. 109-4 at 5)
“connecting” (‘093 Patent, Asserted Claim 1)	“joining or linking together” (Dkt. No. 109-6 at 2)

During the claim construction hearing the parties also agreed that the term **“script”** should be construed to mean **“set of instructions, procedures, and/or functions and related data written in an interpretable programming language.”** In view of the parties’ agreement on the proper construction of each of the identified terms, the Court hereby **ADOPTS** the parties’ agreed constructions.

IV. CONSTRUCTION OF DISPUTED TERMS

A. The ’586 Patent

The parties’ dispute focuses on the meaning and scope of four terms/phrases in the ’586 Patent.

1. “sharing the plurality of objects with a plurality of the one or more computer system[s]”

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"sharing the plurality of objects with a plurality of the one or more computer system[s]"	plain meaning. Alternatively: "making the plurality of objects available to a plurality of the one or more computer systems"[8]	Indefinite. Alternatively: "making objects accessible to one or more applications and/or computer systems and/or sending objects to one or more applications and/or computer systems"

a) The Parties' Position

The parties dispute whether the phrase "sharing the plurality of objects with a plurality of the one or more computer system[s]" is indefinite.[9] The parties also dispute whether "sharing" includes "making the plurality of objects available to a plurality of the one or more computer systems," as Plaintiff proposes. Plaintiff contends that the phrase should be given its plain meaning. In the alternative, Plaintiff argues its construction is consistent with the specifications discussion of what sharing "may include." (Dkt. No. 99 at 10)[10] (citing '586 Patent at 9:28–32, 12:32–35, 12:56–60). Plaintiff contends that Defendant's construction improperly requires every embodiment to include what the specification describes as permissive. (Dkt. No. 99 at 11.)

Defendant responds that for "sharing the plurality of objects," the specification provides definitional guidance. (Dkt. No. 106 at 9) (citing '586 Patent at 9:28–32). Defendant contends that the term "sharing the plurality of objects" therefore includes making the plurality of objects accessible or sending them to one or more applications and/or systems. (*Id.* at 9.) However, Defendant contends that the portion of the "sharing" step that recites that the plurality of objects

[8] Plaintiff provided a new proposed construction during the claim construction hearing. Plaintiff's original proposed construction was "making the plurality of objects available to a plurality of the one or more computer system, including making objects accessible to one or more applications and/or computer systems and/or sending objects to one or more applications and/or computer systems."

[9] The parties agree that the term "system" in the phrase "sharing the plurality of objects with a plurality of the one or more computer system[s]" should be "systems."

[10] All references to page numbers refer to the pagination system assigned by ECF, not the original internal pagination of the document.

be shared "with a plurality of the one or more computer system [sic]," renders the claim indefinite because it is internally contradictory. (*Id.*) Defendant argues that the word "plurality" requires more than one computer system, but the remaining claim language is plainly satisfied by only one such system. (*Id.*) According to Defendant, a person of ordinary skill in the art could not determine with reasonable certainty if the claim requires two computer systems, or is satisfied by just one. (*Id.*)

Defendant argues that the preamble recites an enterprise that "comprises one or more networked computer systems," which is consistent with the express definition of "enterprise" in the specification. (*Id.*) (citing '586 Patent at 1:22–23). Defendant further argues that the preamble language provides the antecedent basis for "the one or more computer system[s]" recited later in the claim. (*Id.*) Defendant contends that this phrase is then contradicted in the sharing step's recital of "a plurality of the one or more computer system [sic]." (*Id.* at 10.) Defendant argues that the specification does not reconcile the term "plurality" with "one or more" in specifying the number of computer systems with which objects must be shared. (*Id.*) According to Defendant, a person of ordinary skill in the art could not determine with any confidence whether the sharing step requires one, or more than one, computer system. (*Id.*)

Plaintiff replies that the claims specify that an "enterprise comprises one or more networked computer systems" and that sharing occurs with "a plurality" of the "one or more computer system." (Dkt. No. 108 at 5.) Plaintiff contends that the claim simply requires that sharing occurs among a plurality of the enterprise computers, and Defendant's misunderstanding of the claims cannot be a basis for finding the claim indefinite. (*Id.*)

For the following reasons, the Court finds that the phrase **"sharing the plurality of objects with a plurality of the one or more computer system[s]"** is not indefinite, and should be construed to mean **“making accessible and/or sending the plurality of objects to a plurality of applications and/or a plurality of computer systems.”**

b) Analysis

The phrase “sharing the plurality of objects with a plurality of the one or more computer system” appears in asserted claim 1 of the ’586 Patent. The claim language indicates that the recited “a plurality of the one or more computer system[s]” refers to a plurality of computer systems. The parties do not dispute that the preamble language of “wherein the enterprise comprises one or more networked computer systems” provides antecedent basis for “the one or more computer system[s].” The plain meaning of a “plurality” means more than one, whether it be a plurality of “one computer system” or a plurality of “computer systems.” Thus, in the context of claim 1, “a plurality of the one or more computer system[s]” is simply another way of stating “a plurality of computer systems.” Accordingly, the Court finds that the claim language, “viewed in light of the specification . . ., inform[s] those skilled in the art about the scope of the invention with reasonable certainty.” *Nautilus*, 134 S. Ct. at 2129.

The specification further provides a definition for the phrase “sharing objects.” Specifically, the specification states that “[a]s used herein, ‘sharing objects’ may include making objects accessible to one or more applications and/or computer systems and/or sending objects to one or more applications and/or computer systems.” ’568 Patent at 9:28–32. The claim language recites “sharing the plurality of objects,” thus a person of ordinary skill would understand that the phrase “sharing the plurality of objects with a plurality of the one or more computer system[s]” means “making accessible and/or sending the plurality of objects to a plurality of applications and/or a plurality of computer systems.”

Plaintiff contends that in addition to making objects accessible, they must also be made available. Given the explicit definition provided in the specification, the Court is not persuaded that including "making the plurality of objects available" is required in the construction. The Court agrees that the specification states that sharing "may" include making objects accessible, but the claim language recites actually "sharing the plurality of objects," and is not permissively worded like the specification. Indeed, in reply to Defendant's indefinite contention, Plaintiff argues "[t]he claims specify that an 'enterprise comprises one or more networked computer systems' and that sharing occurs with 'a plurality' of the 'one or more computer systems'. . . ." (Dkt. No. 108 at 5.) Accordingly, the Court does not adopt this aspect of Plaintiff's construction.

c) Court's Construction

In light of the evidence presented by the parties, the Court finds that the phrase **"sharing the plurality of objects with a plurality of the one or more computer system"** is not indefinite. The Court further finds that the phrase should be construed to mean **"making accessible and/or sending the plurality of objects to a plurality of applications and/or a plurality of computer systems."**

2. "hierarchical namespace"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"hierarchical namespace"	"a memory or plurality of memories which are coupled to one another, whose contents are uniquely addressable and are arranged in a hierarchical way"	"hierarchical set of unique names"

a) The Parties' Position

The parties dispute whether the term "hierarchical namespace" should be construed to include "a memory or plurality of memories which are coupled to one another," as Plaintiff proposes, or if it should be construed as "hierarchical set of unique names," as Defendant proposes. Plaintiff contends that its construction is taken directly from the specification. (Dkt.

No. 99 at 11) (citing ’586 Patent at 1:54–56). According to Plaintiff, a “hierarchical” namespace is merely a namespace whose contents are arranged in a hierarchical way. (Dkt. No. 99 at 11) (citing ’586 Patent at 13:52–53). Plaintiff argues that Defendant’s construction requires that “names” themselves be hierarchical. (Dkt. No. 99 at 11.) Plaintiff contends that the specification gives an example where it is the contents of the namespace that may be arranged hierarchically. (Dkt. No. 99 at 12) (citing ’586 Patent at 3:61–66).

Defendant responds that one of ordinary skill in the art would have understood “namespace” to refer to a set of unique names. (Dkt. No. 106 at 11.) Defendant contends that this definition is consistent with the ’586 Patent. (*Id.* at 11) (citing ’586 Patent at 1:52–60). Defendant further argues that some of the statements in the specification provide definitional language for “namespace” while others merely identify properties of a namespace, such as its ability to refer to a memory or a plurality of memories. (Dkt. No. 106 at 11.)

Defendant also argues that Plaintiff’s proposed construction confuses a “namespace” with the underlying items to which the unique names refer. (*Id.*) Defendant argues that Plaintiff’s construction would entirely eliminate the concept of “names” from a “namespace.” (*Id.*) Defendant contends that Plaintiff ignores the sentence in the specification that expressly defines “uniquely addressable” as “the property that items in a namespace have unique names such that any item in the namespace has a name different from the names of all other items in the namespace.” (*Id.*) (citing ’586 Patent at 1:57–59). Defendant also contends that its construction is consistent with contemporaneous dictionary definitions. (Dkt. Bo. 106 at 11) (citing Dkt. No. 106-7 at 6, Dkt. No. 106-8 at 4).

Plaintiff replies that Defendant’s construction ignores the specification’s recitation that “[a]s used herein, a ‘namespace’ may refer to a memory, or a plurality of memories which are coupled to one another, whose contents are uniquely addressable." (Dkt. No. 108 at 5-6) (citing '586 Patent at 1:54–56). Plaintiff contends that Defendant argues that a namespace is a set of names in and of itself, which disregards the aforementioned description of "a memory" in the specification. (Dkt. No. 108 at 6.) Plaintiff further contends that this is contrary to Defendant's argument that "uniquely addressable" requires that "items in a namespace have unique names . . . ."—i.e. the namespace is not a set of names in and of itself. (*Id.*)

For the following reasons, the Court finds that the term **"hierarchical namespace"** should be construed to mean **"a memory or a plurality of memories coupled to one another containing a hierarchical set of objects such that any object in the set has a different name from all other objects in the set."**

b) Analysis

The term "hierarchical namespace" appears in asserted claim 1 of the '586 Patent. As an initial matter, the Court notes that neither party seeks to construe the term "hierarchical." Thus, the dispute focuses on the term "namespace." The Court finds that the specification provides an explicit definition of "namespace." Specifically, the specification states the following:

> An enterprise-wide namespace is one way to make data available throughout an enterprise. A namespace provides efficient referencing and retrieval of information. The term "namespace" generally refers to a set of names in which all names are unique. As used herein, a "namespace" may refer to a memory, or a plurality of memories which are coupled to one another, whose contents are uniquely addressable. "Uniquely addressable" refers to the property that items in a namespace have unique names such that any item in the namespace has a name different from the names of all other items in the namespace.

'586 Patent at 1:50–60, *see also id.* at 11:21–30. As indicated above, the specification states that the term "namespace" generally refers to a set of names such that any item in the namespace has a name different from the names of all other items in the namespace. Plaintiff's construction fails to clearly capture this aspect of the definition. Although Plaintiff's construction includes the phrase "uniquely addressable," it does not clearly state that this "refers to the property that items in a namespace have unique names such that any item in the namespace has a name different from the names of all other items in the namespace." '586 Patent at 1:57–60.

As indicated above, the specification also states that a "namespace" may refer to a memory or a plurality of memories which are coupled to one another, whose contents are uniquely addressable. Defendant's construction fails to capture this aspect of the definition. As stated in the specification, "uniquely addressable" requires "items in a namespace have unique names such that any item in the namespace has a name different from the names of all other items in the namespace." This indicates that the namespace is not a set of names in and of itself. Indeed, the specification states that "[a] namespace is typically a logical organization and not a physical one," and that "a namespace may be thought of as a plurality of distinct physical memories which are organized as a single, and possibly distributed, logical memory." '586 Patent at 2:12–17. To the extent that Defendant argues that a namespace is a set of names in and of itself, the Court rejects this argument.

Finally, the specification states that the objects in the namespace are arranged or stored hierarchically. For example, the specification states that "[t]he namespace comprises a logical arrangement of the objects, stored hierarchically." '586 Patent at 3:62–63. Likewise, the specification states that "[t]he enterprise-wide namespace may employ a simple hierarchical information model in which the objects are arranged hierarchically." *Id.* at 11:44–47. Accordingly, the Court finds that the term "hierarchical namespace" should be construed to mean "a memory or a plurality of memories coupled to one another containing a hierarchical set of objects such that any object in the set has a different name from all other objects in the set."

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the term **"hierarchical namespace"** to mean **"a memory or a plurality of memories coupled to one another containing a hierarchical set of objects such that any object in the set has a different name from all other objects in the set."**

3. "dynamically inherits traits from the prototype" and "wherein the values of the traits inherited from the prototype change dynamically"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"dynamically inherits traits from the prototype"	construe "traits" and "prototype." Otherwise, plain meaning.	"derives attributes, values, and/or children from another object, where the attributes, values, and/or children may change over time"
"wherein the values of the traits inherited from the prototype change dynamically"	"wherein the values of the traits inherited from the prototype are inherited as they are changed"	"wherein the values of the attributes in the prototype object change over time"

a) The Parties' Position

The parties dispute whether the phrase "dynamically inherits traits from the prototype" requires construction. The parties also dispute whether the phrase "wherein the values of the traits inherited from the prototype change dynamically" should be construed as "wherein the values of the traits inherited from the prototype are inherited as they are changed," as Plaintiff proposes, or as "wherein the values of the attributes in the prototype object change over time," as Defendant proposes. For the phrase "dynamically inherits traits from the prototype," Plaintiff argues that the Court should construe "traits" and "prototype," and then give the phrase its plain meaning.

For the phrase "wherein the values of the traits inherited from the prototype change dynamically," Plaintiff argues that its construction is supported by the intrinsic evidence, which explains that “[t]he attributes in a prototype may change over time, and such changes are ‘dynamically’ inherited by an instance. That is changes in one part of the enterprise system may be transmitted to one or more related parts of the system.” (Dkt. No. 99 at 13) (citing Dkt. No. 99-6 at 12 (’586 FH, April 8, 2004 Amendment)). Plaintiff also argues that the specification further states, “[a]s used herein, ‘dynamic inheritance’ includes the ability to derive attributes, values, and/or children from another object, where the attributes, values, and/or children may change over time.” (Dkt. No. 99 at 13) (citing ’586 Patent at 14:57–60). According to Plaintiff, where prototype attribute values or children change over time, dynamic inheritance involves the ability to derive those as they change. (Dkt. No. 99 at 13.) Plaintiff argues that Defendant’s construction is contrary to this intrinsic evidence and would eviscerate the notion of “dynamic” inheritance because any type of inheritance could be argued to be “dynamic” if the prototype values simply change over time. (*Id.*)

For the phrase “dynamically inherits traits from the prototype,” Defendant argues that it should be construed using the specification language verbatim. (Dkt. No. 106 at 12.) According to Defendant, the phrase should be construed as requiring that the instance “derive[s] attributes, values, and/or children from another object, where the attributes, values, and/or children may change over time.” (*Id.*) (citing ’586 Patent at 14:57–60).

For the phrase “wherein the values of the traits inherited from the prototype change dynamically,” Defendant argues that the phrase “change dynamically” in the claim corresponds to “change over time.” (Dkt. No. 106 at 12.) Defendant contends that Plaintiff argues that “change dynamically” requires that the values of the traits be “inherited as they are changed.” (*Id.*) Defendant argues that Plaintiff’s construction misapplies the word “dynamically” in this part of the claim. (*Id.*) According to Defendant, the adverb “dynamically” modifies “change,” and specifies that the values "change dynamically." (*Id.*) Defendant argues that the plain language of "change dynamically" does not refer to inheritance but to the ability of the values to change over time. (*Id.*)

Plaintiff replies that Defendant insists on a tautological interpretation that "change" occurs "over time" but does not address the discussion in the intrinsic evidence of what is meant by "dynamic" inheritance. (Dkt. No. 108 at 6) (citing Dkt. No. 99-6 at 12 ('586 FH, April 8, 2004 Amendment)). Plaintiff argues that its construction defines "dynamic inheritance" consistent with the intrinsic evidence and in a manner helpful to the jury. (Dkt. No. 108 at 7.)

For the following reasons, the Court finds that the phrase **"dynamically inherits traits from the prototype"** should be construed to mean **"derives traits from the prototype that may change over time,"** and the phrase **"wherein the values of the traits inherited from the prototype change dynamically"** should be construed to mean **"wherein the values of the traits inherited from the prototype change as or after the traits of the prototype change."**

b) Analysis

The phrase "dynamically inherits traits from the prototype" appears in claims 1, 8, 14, and 21 of the '586 Patent. The Court finds that the phrase is used consistently in the claims and is intended to have the same meaning in each claim. The phrase "wherein the values of the traits inherited from the prototype change dynamically" appears in claims 1, 8, 14, 21, 28 of the '586 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The Court further notes that it will construe the term "traits," and that the parties have agreed to constructions for the terms "instance" and "prototype." Thus, the Court finds no reason to provide a different construction for these terms as they are used in the disputed phrases. The Court also finds that these terms are used consistently in the claims and are intended to have the same meaning in each claim. Accordingly, the Court focuses on the phrases “dynamically inherits” and “change dynamically.”

Starting with the claim language, the Court finds that it indicates that the phrases complement one another and should be considered together. Specifically, all of the independent claims recite “wherein the instance dynamically inherits traits from the prototype; and wherein the values of the traits inherited from the prototype change dynamically.”[11] Thus, the claim language indicates that the “instance” inherits “traits” from the “prototype,” and as a result, the “values of the traits” inherited from the “prototype” change as or after the traits of the prototype change.

This is further confirmed by the specification that states that “[a]s used herein, ‘dynamic inheritance’ includes the ability to derive attributes, values, and/or children from another object, where the attributes, values, and/or children may change over time.” ’586 Patent at 14:57–60. The specification provides an example where “object ‘b’ 456 [the instance] has an attribute called ‘x’ of its own and also inherits the attribute ‘y’ from object ‘a’ 454 [the prototype].” *Id.* at 14:60–62. Thus, in the context of the disputed phrase, and as illustrated by this example, the Court finds that “dynamically inherits” means that the instance derives traits from the prototype.

The claim language further requires that “value of the traits … change dynamically.” Again, as indicated in the specification, this means that the “values of the traits” inherited from the prototype change as or after the traits of the prototype change. This is confirmed by the arguments made by the patentees during prosecution. Specifically, the patentees argued that “[t]he attributes in a prototype may change over time, and such changes are ‘dynamically’ inherited by an instance. That is changes in one part of the enterprise system may be transmitted

[11] Claim 28 omits “dynamically” from the first phrase and recites “wherein the instance inherits traits from the prototype.”

to one or more related parts of the system." (Dkt. No. 99-6 at 12 ('586 FH, April 8, 2004 Amendment)). Indeed, the specification discusses "publishing" and "subscribing" as one way to receive the data and/or changes in the data over time. '586 Patent at 17:37–18:10. Accordingly, the Court finds that the phrase "change dynamically" means that the values of the traits inherited from the prototype change as or after the traits of the prototype change.

Turning to the parties' constructions, the Court does not adopt Defendant's construction because it includes Defendant's construction of "traits." In addition, Defendant's construction does not capture the "dynamic" relationship between the prototype and instance. Instead, Defendant's construction simply states that the values in the prototype object change over time. The Court does not adopt Plaintiff's construction for the phrase "wherein the values of the traits inherited from the prototype change dynamically" because it is confusing and not as clear as the Court's construction.

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the phrase **"dynamically inherits traits from the prototype"** to mean **"derives traits from the prototype that may change over time;"** and the phrase **"wherein the values of the traits inherited from the prototype change dynamically"** to mean **"wherein the values of the traits inherited from the prototype change as or after the traits of the prototype change."**

4. "traits"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"traits"	plain meaning	"As used herein, an object is a self-contained entity that contains data and/or procedures to manipulate the data. Objects may be stored in a volatile memory and/or a nonvolatile memory."

a) The Parties' Position

The parties dispute whether the term "traits" requires construction. Plaintiff contends that the term should be given its plain meaning because a "trait" is readily understood by those of ordinary skill in the art. (Dkt. No. 99 at 12.) Plaintiff argues that the specification discusses "traits" in the context of "object traits" and "attribute traits." (*Id.*) (citing '586 Patent at 17:9–11, 17:21–23). Plaintiff contends that the word "trait" already conveys the understanding of one of ordinary skill in the art that an object has traits that hold information. (Dkt. No. 99 at 12.)

According to Plaintiff, Defendant's construction that a "trait" itself is an attribute value and/or child object is incorrect. (*Id.*) Plaintiff argues that the specification dictates that a trait is not itself an "attribute value and/or child object," but rather may comprise (i.e., may include or may hold) attribute values or child objects. (*Id.*) (citing '586 Patent at 17:9–11, Claims 5, 6). Plaintiff further argues that Defendant's construction seeks to incorporate claims 5 and 6 into independent claim 1 and is presumptively wrong according to claim differentiation. (Dkt. No. 99 at 12-13.)

Defendant responds that the term "traits" in claim 1 refers to the information inherited from a prototype object. (Dkt. No. 106 at 7-8.) Defendant argues that the specification makes clear that inheritance of "traits" is synonymous with inheritance of "attribute values and/or child objects." (Dkt. No. 106 at 8) (citing '586 Patent at 4:9−11, 11:58−62). Defendant contends that the definition is consistent with the specification's definitions of "prototype" and "instance." (Dkt. No. 106 at 8.) According to Defendant, those definitions do not describe inheritance of "traits" but of "attributes, values and/or children." (*Id.*) Defendant argues this is consistent with "traits" being synonymous with "attribute values and/or child objects." (*Id.*)

Regarding Plaintiff's claim differentiation argument, Defendant argues that the doctrine of claim differentiation only ensures that independent and dependent claims have different scopes. (*Id.*) Defendant argues that its construction requires inheritance of "attribute values and/or child objects" for claim 1, thus being satisfied by inheritance of either or both of them. (*Id.*) Defendant further argues that dependent claims 5 and 6 narrow claim 1 to require that the inherited traits include "child objects" (claim 5), or both attribute values and child objects (claim 6). (*Id.*) According to Defendant, dependent claims 5 and 6 are narrower in scope than claim 1 under its construction and are therefore not rendered superfluous. (*Id.*)

Defendant also argues that the analysis of Plaintiff's expert confirms that there is no such thing as a "plain meaning" for the term "traits." (*Id.*) Defendant argues that the "object traits" referred to by Plaintiff's expert are not the same "traits" that are dynamically inherited from the prototype recited in claim 1. (*Id.*) Defendant contends that the specification consistently describes the inherited "traits" as attribute values and/or child objects. (*Id.* at 9.)

Plaintiff replies that contrary to Defendant's construction, "traits" may comprise attribute values or child objects. (Dkt. No. 108 at 6) (citing '586 Patent at Claims 5, 6). Plaintiff contends that the specification explains that traits may be "pieces of data that hold additional information about" object types or attributes and gives an example of a trait. (Dkt. No. 108 at 6) (citing '586 Patent at 17:9–11, 17:21–23). Plaintiff argues that defining a trait as an attribute value or child object ignores that a trait may also have a name ("display") and is used to hold values or objects. (Dkt. No. 108 at 6.) Plaintiff further argues that this distinction also appears in claim 1, which requires that "values of the traits" – not the "traits" themselves – change dynamically. (*Id.*) According to Plaintiff, the values or objects that a trait may hold may be inherited along with the trait, but the specification's discussion of such inheritance does not mean that a "trait" is an "attribute value and/or child object" in and of itself. (*Id.*)

For the following reasons, the Court finds that the term **"traits"** should be construed to mean **"information about or representation of an object or an attribute, such as attribute values and/or child objects."**

b) Analysis

The term "traits" appears in asserted claim 1 of the '586 Patent. The Court finds that the claim indicates that the recited "traits" can include values and may comprise attribute values and child objects. For example, claim 1 recites "wherein the values of the traits inherited from the prototype change dynamically." Likewise, claim 6 recites "wherein the inherited traits comprise attribute values and child objects." Thus, the claim language indicates that the recited "traits" can include attribute values or child objects. However, contrary to Defendant's construction, the specification does not indicate that the recited "traits" can only be attribute values or child objects.

Specifically, the specification states that a schema object may maintain "object traits" for a particular object type and "attribute traits" for a particular attribute. In describing the "object traits" and "attribute traits," the specification states the following:

> In one embodiment, a schema object may maintain object traits for a particular object type. As used herein, "object traits" are pieces of data that hold additional *information about an object type*. For example, to support generic object editors, it is advantageous to know the valid child types for a particular object when creating a child of that object. An object trait called "child-types" may be defined to provide this information. The "child-types" object trait may be an array of strings *representing object types*. The generic object editor may refer to the "child-types" object trait to present a list of valid child types to the user of the generic object editor.
>
> In one embodiment, a schema object may maintain attribute traits for a particular attribute. As used herein, "attribute traits" are pieces of data that hold additional *information about one or more attributes*. For example, an attribute trait called "values" may hold the set of valid values for a given attribute. An attribute trait called "display" may hold the set of valid values as they would be displayed in a user interface. An attribute trait called "read-only" may be true if the attribute is read-only. An attribute trait called "description" may hold a description of the attribute as a string. An attribute trait called "units" may hold the units of measurement in which the attribute is expressed. An attribute trait called "visible"

> may be true if user interfaces should make the attribute visible. An attribute trait called “path” may be used to indicate that the attribute contains a path to another object (i.e., the attribute is part of an association)

’586 Patent at 17:8–35 (emphasis added). As indicated above, the specification states that “traits” are information about or representation of an object type or an attribute. For example, they may be valid child types for a particular object, or valid values for an attribute. Thus, the Court finds that a person of ordinary skill in the art would understand that the recited “traits” means “information about or representation of an object or an attribute.”

Defendant is correct that the specification also indicates that the recited “traits” may include “attribute values and/or child objects.” *See, e.g.*, ’586 Patent at 4:9–11 (“The instance inherits from the prototype traits such as attribute values and/or child objects.”). However, as discussed above, the intrinsic evidence indicates that the term “traits” is not limited or strictly synonymous with “attribute values and/or child objects.” Defendant contends that “object traits” are not the same “traits” that are dynamically inherited from the prototype recited in claim 1. (Dkt. No. 106 at 8) (citing 106-1 at 25 (Lavian Decl. ¶ 61)). According to Defendant’s expert, “the claims are not limited to ‘schema objects’ and nothing in the specification indicates that these schema object traits are the same ‘traits’ that are dynamically inherited from the prototype as recited in the claim.” (Dkt. No. 106-1 at 25 (Lavian Decl. ¶ 61)). The Court is not persuaded by Defendant’s conclusory analysis. The specification states that “one or more schemas may be designated, wherein each schema is a template which specifies one or more valid traits for a type of object.” ’586 Patent at 4:32–34. As discussed above, it is the “object traits” that enable valid objects to be created, and it is after the objects are created that they and their respective attributes can be dynamically inherited.

c) Court’s Construction

In light of the evidence presented by the parties, the Court construes the term **"traits"** to mean **"information about or representation of an object or an attribute, such as attribute values and/or child objects."**

B. The '898 Patent

The parties' dispute focuses on the meaning and scope of three terms/phrases in the '898 Patent.

1. "periodically"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"periodically"	plain meaning. Alternatively: at an established interval of time	Indefinite

a) The Parties' Position

The parties dispute whether the term "periodically" is indefinite. Plaintiff contends that this term would be understandable by the jury and, thus, should be left for plain meaning. (Dkt. No. 99 at 9.) In the alternative, Plaintiff argues that its construction is consistent with the intrinsic record that the periodic collection of data may occur at a "5 minutes interval" in one example, or at a "15 minute interval" in another example. (*Id*) (citing '898 Patent at 8:1–2, Fig. 6C). Plaintiff argues that its construction is also consistent with other intrinsic evidence, including references cited during prosecution and appearing on the cover of the '898 Patent. (Dkt. No. 99 at 9.)

Plaintiff further argues that a person of ordinary skill in the art reading the claims, in light of the specification and prosecution history, would be informed with reasonable certainty about the meaning of "periodically" within the scope of the invention. (*Id* at 10) (citing Dkt. No. 99-19 at 5-6 (Smith Decl. ¶¶17-20)). Plaintiff contends that Defendant had no trouble construing "periodically" for purposes of alleging prior art. (Dkt. No. 99 at 10.)

Defendant responds that the term “periodically” has no specific technical meaning and has two inconsistent definitions. (Dkt. No. 106 at 13) (citing Dkt. No. 106-1 at 41 (Lavian Decl. ¶ 100)). Defendant argues that Random House defines “periodic” as “occurring at regular intervals of time,” or as “recurring irregularly; intermittent.” (Dkt. No. 106 at 13.) Defendant further argues that American Heritage acknowledges this confusion by stating “[p]eriodic has long been used loosely to mean ‘occasional, intermittent,’ but this usage may be confusing for readers who are accustomed to that word only in its narrower sense of ‘at regular or predictable intervals.” (*Id.*)

Defendant argues that systems that collect data from network components were well-known to skilled artisans, and in such systems, it is important to know when that data must be collected. (Dkt. No. 106 at 13) (citing Dkt. No. 106-1 at 42 (Lavian Decl. ¶ 102)). According to Defendant, the term “periodically” in claims 2 and 6 either allows data to be collected irregularly, or requires collection of data at specific or regular intervals. (Dkt. No. 106 at 13.) Defendant contends that the specification’s reference to 5 minute intervals is simply an embodiment, and there is no disclaimer or other basis in the intrinsic record to warrant importing that embodiment into the claim. (*Id.*) (citing ’898 Patent at 12:51–53). Defendant also argues that the intrinsic record does not suggest that Plaintiff intended to adopt or incorporate anything from the two patents cited by the examiner. (Dkt. No. 106 at 14.) Defendant further argues that the fact that those patents felt the need to provide express guidance on “periodically” supports a finding of indefiniteness here. (*Id.*)

Plaintiff replies that Defendant does not identify anything in the intrinsic record inconsistent with its proposed construction, nor provide any reasoning why the Court should not adopt its construction. (Dkt. No. 108 at 4.) Plaintiff contends that “periodically” means “at an established interval of time." (*Id.* at 5.) Plaintiff argues that Defendant does not dispute that the '898 Patent teaches collecting data "at an established interval of time," such as every 5 minutes. (*Id.*) Plaintiff also argues that two intrinsic references teach collecting data "at an established time," consistent with its construction. (*Id.*)

For the following reasons, the Court finds that the term **"periodically"** should be given its **plain and ordinary meaning.**

b) Analysis

The term "periodically" appears in claims 2, 3, and 6 of the '898 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The Court further finds that the term is not ambiguous, is easily understandable by a jury, and requires no construction. For example, claim 3 recites that the service monitor is used "to periodically collect the performance data." Consistent with the claims, the specification states that "[t]he network performance monitoring and management system schedules the script for periodic execution and the data collected is stored in its database." '898 Patent at 9:37–39; *see also, id.* at 10:46–48 ("The system provides the information from the user to the system such that the system can automatically call the service monitor and run it periodically.").

As another example, the specification states that "[b]y default, the configuration poll is scheduled once a day." *Id.* at 7:64. In yet another example, the specification states that "[b]y default, the Stat poll is scheduled at 5 minutes interval." *Id.* at 8:1–2. The specification further states that "[t]he poll rates for both Config and Stat polls may be modified via an Admin graphical user interface (GUI) by editing the service monitor's instance Control tab." *Id.* at 8:6–8. Figure 6C also states that "[s]tatistics data variables are polled for many times a day typically (i.e., every 15 minutes)." *Id.* at Fig. 6C. In the context of the claims, all of these examples indicate that "periodically" should be given its plain and ordinary meaning. Accordingly, the Court finds that the claim language, "viewed in light of the specification . . ., inform[s] those skilled in the art about the scope of the invention with reasonable certainty." *Nautilus*, 134 S. Ct. at 2129.

Defendant argues that the term "periodically" is indefinite because an extrinsic dictionary provides contradictory definitions for the term "periodic." (Dkt. No. 106 at 13) Specifically, Defendant argues that Random House defines "periodic" as "occurring at regular intervals of time," or as "recurring irregularly; intermittent." (*Id.*) Defendant contends that there is no way to know when that data must be collected because this extrinsic evidence defines "periodically" as both "regular intervals" and "recurring irregularly." The Court disagrees with Defendant's argument and finds that it is contrary to well established law.

Most importantly, Defendant does not contend that the intrinsic evidence indicates that the term "periodically" is used to mean both "regular" and "irregularly." Instead, Defendant argues that the consistent use of "periodically" in the intrinsic record is simply describing embodiments that should not be read into the claim. To be clear, the Court is not reading the 5 minute interval, 15 minute interval, or even the daily interval into the claims. Instead, the Court finds that this intrinsic evidence informs a person or ordinary skill in the art of the scope of the claims with reasonable certainty. That is, a person of ordinary skill in the art would understand that the data will be collected "periodically." There is no mention of "irregular" intervals in the intrinsic evidence. Accordingly, the Court rejects Defendant's attempt to use extrinsic evidence to contradict the intrinsic evidence. *Gart v. Logitech, Inc.*, 254 F.3d 1334, 1340 (Fed. Cir. 2001) ("[E]xtrinsic evidence cannot be used to contradict the established meaning of the claim language.").

c) Court’s Construction

In light of the evidence presented by the parties, the Court finds that the term **“periodically”** is not indefinite and should be given its **plain and ordinary meaning**.

2. “script-based program”

Disputed Term	Plaintiff’s Proposal	Defendant’s Proposal
“script-based program”	program based on a script	a set of instructions written in a plain text, interpretable language

a) The Parties’ Position

The parties dispute whether the term “script-based program” should be construed as “program based on a script,” as Plaintiff proposes, or as “a set of instructions written in a plain text, interpretable language,” as Defendant proposes. Plaintiff contends that its constructions have express definitional support in the intrinsic record. (Dkt. No. 99 at 8.) Plaintiff argues that Defendant’s construction limits the type of language in which a script can be written to only a “plain text, interpretable language,” which it contends is contrary to the specification’s teaching. (Dkt. No. 99 at 8) (citing ’898 Patent at 3:63–65, 9:60–62). Plaintiff also argues that Defendant’s construction of “script-based program” erroneously excludes the word “program.” (Dkt. No. 99 at 8.)

Defendant responds that skilled artisans understood “script” to refer to a particular type of computer program. (Dkt. No. 106 at 14) (citing Dkt. No. 106-1 at 37-38 (Lavian Decl. ¶ 93)). According to Defendant, the key characteristic distinguishing a “script” from other types of computer programs is that a “script” is written in a plain text, interpretable programming language, and its instructions are carried out by an interpreter. (*Id.*) Defendant argues that this is consistent with the ’898 specification, which provides an exemplary script program having a set of instructions written in plain text. (Dkt. No. 106 at 15) (citing ’898 Patent at Figure 4, 9:52–

56).

Defendant further argues Plaintiff's construction is wrong because it essentially describes any possible computer program and therefore seeks to erase the word "script" or "script-based" from the claim. (Dkt. No. 106 at 15.) According to Defendant, the '898 specification confirms that a "script" or "script-based program" does not refer to all possible computer programs as Plaintiff's construction would suggest. (Dkt. No. 106 at 15) ('898 Patent at 4:54–55, 7:15–21). Defendant argues that Plaintiff's construction could even cover programs that are directly executable without an interpreter. (*Id.*)

Plaintiff replies that Defendant's construction of "script" is based on extrinsic evidence and identifies nothing in the intrinsic record requiring the "plain text" and "interpretable" language. (Dkt. No. 108 at 3.) Plaintiff argues that the '898 Patent teaches that "the present invention is not described with reference to any particular programming language." (*Id.*) (citing '898 Patent 3:63–65). Plaintiff also argues the Defendant's extrinsic dictionaries do not impose both a "plain text" and an "interpretable" language requirement in one definition. (Dkt. No. 108 at 4.) Plaintiff contends that its construction has both intrinsic definitional support and extrinsic support. (*Id.*) (citing Dkt. 99-19 at 3-4 (Smith Decl. ¶¶8-12)). Plaintiff also argues that Defendant does not explain how any of the passages cited by Plaintiff are inconsistent with its construction. (Dkt. No. 108 at 4.) Plaintiff also disagrees with Defendant's argument that the definition of "script" in the intrinsic Bromberg reference is "not relevant." (*Id.*)

For the following reasons, the Court finds that the term **"script-based program"** should be construed to mean **"program based on a script."**

b) Analysis

The term "script-based program" appears in asserted claims 2, 3, 6, and 8 of the '898

Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The Court further finds that the intrinsic and extrinsic evidence indicates that the recited "script-based program" is program based on a script. Claim 6 recites that the performance management system receives "at least one script-based program from the user." Consistent with the claims, the specification further states the following:

> In one embodiment, the user provides the script-based program by copying the script into a directory server on a sever used by the network managing system (e.g., network monitor 150). In one embodiment, the script can be in any shell script language, PERL, or any similar program language. Alternatively, an executable of the script-based program could be used.

'898 Patent at 4:54–60. In further describing the script-based programs, the specification states:

> Such script-based programs are either existing programs used before installation of the performance management system or newly written executables to collect additional data not covered by the existing performance management system. The script in FIG. 4 is solely for [i]llustration, not to restrict the type of scripts that can be accepted nor the type of data that can be defined within the scripts. In one embodiment, the user copies the file containing the script to a directory in the performance management system server. For example, "remedy.sh" is copied into a designated directory in network router 150.

Id. at 9:56–66. Thus, the claims and the specification indicate that the user provides the script-based program that is incorporated or integrated into the system. In other words, a person of ordinary skill in the art would understand that the script-based program is a program based on script.

This is consistent with the extrinsic evidence that defines "script" as a "[a] program written in an interpreted programming language typically smaller in scope and function than full-blown compiled languages such as C and C++." (Dkt. No. 106-9 at 4 (Computer Desktop Encyclopedia (9th ed. 2001)). Similarly, the Newton's Telecom Dictionary defines "script" as "[a] type of computer code than can be directly executed by a program that understands the language in which the script is written. Scripts do not need to be compiled into object code to be executed." (Dkt. No. 106-10 at 4 (Newton's Telecom Dictionary (16th ed. 2000)).

Turning to the parties' construction, the Court does not adopt Defendant's construction because it limits the term to "plain text." As Plaintiff argues, the extrinsic dictionaries submitted by Defendant do not consistently refer to "plain text" as a requirement for a "script-based" program. In fact, the definition provided in the Collins Computing Dictionary appears to be directed to the very specific application of making a connection with a remote computer over telephone lines using a modem. (Dkt. No. 106-12 at 5 (Collins Computing Dictionary (3d ed. 2000))). The claims are not so limited, and the specification explicitly states that "[t]he script in FIG. 4 is solely for [i]llustration, not to restrict the type of scripts that can be accepted nor the type of data that can be defined within the scripts." '898 Patent at 9:60–63. Accordingly, the Court does not adopt this aspect of Defendant's construction.

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the term **"script-based program"** to mean **"program based on a script."**

3. "service monitor"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"service monitor"	program for monitoring a device, application or server in a network	a program running on the performance management system that automatically collects user-defined data from the components of the network

a) The Parties' Position

The parties dispute whether the term "service monitor" is a program running on the performance management system, as Defendant proposes. The parties also dispute whether the construction of "service monitor" should require the program to "automatically collects user-defined data," as Defendant proposes. Plaintiff contends that its construction is drawn directly from the specification. (Dkt. No. 99 at 9.) Plaintiff argues that Defendant's construction is incorrect because the specification includes an embodiment in which the "service monitor" appears to the user in a list of monitors and must be activated before the collection of data. (*Id.*) (citing '898 Patent at 10:56–11:6). Plaintiff also argues that in one embodiment, the system "call[s] the service monitor" and "run[s] it periodically," in contrast to Defendant's construction where the "service monitor" is constantly "running." (Dkt. No. 99 at 9) (citing '898 Patent at 10:46–55).

Defendant responds that "service monitor" does not have a commonly understood meaning to persons of ordinary skill in the art. (Dkt. No. 106 at 15.) Defendant contends that the patent explains that a service monitor is a program integrated into the performance management system. (*Id.* at 16) (citing '898 Patent at 8:44–46). Defendant further argues that a service monitor runs on the performance management system automatically to collect data from components on the network. (Dkt. No. 106 at 16) (citing '898 Patent at 11:13–17). Defendant also argues that the collected data includes user-defined data. (Dkt. No. 106 at 16) (citing '898 Patent at 7:22–27, 7:28–30). According to Defendant, a "service monitor" is therefore "a program running on the performance management system that automatically collects user-defined data from the components of the network." (Dkt. No. 106 at 16.) Regarding Plaintiff's activation argument, Defendant contends that the '898 Patent makes clear that the service monitor runs automatically after being activated. (Dkt. No. 106 at 16) (citing '898 Patent at 11:13–16.) Defendant contends that activation does not prevent subsequent automatic data collection. (Dkt. No. 106 at 16.)

Plaintiff replies that Defendant seeks to add limitations that are unnecessary and contradict the intrinsic record by including "automatically collects user-defined data" and "running on the performance management system." (Dkt. No. 108 at 4.) Plaintiff argues that the patent teaches an embodiment where the service monitor appears to the user in a list of monitors and must be activated before data collection. (Dkt. No. 108 at 4) (citing '898 Patent at 10:56–11:6). According to Plaintiff, the "service monitor" is not by definition "running" and "automatically" collecting data. (Dkt. No. 108 at 4.) Plaintiff further argues that Defendant's argument that the service monitor runs automatically after being activated does not track its construction, and ignores the specification teaching that the service monitor may be called by a network component, after activation but before collecting data. (Dkt. No. 108 at 4) (citing '898 Patent at 10:48–55).

For the following reasons, the Court finds that the term **"service monitor"** should be construed to mean **"script-based program used by the performance management system to monitor a device, application, or server."**

b) Analysis

The term "service monitor" appears in asserted claims 3 and 6-12 of the '898 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The Court further finds that the intrinsic evidence indicates that recited "service monitor" is a script-based program used by the performance management system. The specification states that "[o]ne advantage of the network performance monitoring and management system is to make it easier to integrate individual script-based programs with the performance management system." '898 Patent at 11:55–58. The specification further states that "[n]etwork monitor 150 collects meta data and data defined by the script-based programs from the network 160 using service monitor 140." *Id.* at 7:28–30. Likewise, the specification states that "[a]s soon as the new service monitor is added and activated, the network performance management system automatically uses it to collect the customized data from the components of the network . . . .” *Id.* at 11:13–16. Thus, the Court finds that the recited “service monitor” is a script-based program used by the performance management system. This is further indicated by claim 6, which recites “the performance management system using the service monitor.” *See also*, Claim 3 (“integrating the at least one script-based program into the performance management system as a service monitor; and using the service monitor to periodically collect the performance data”).

The Court further finds that the intrinsic evidence indicates that the recited “service monitor” is used by the performance management system to monitor a device, application, or server. For example, the specification states that “[t]he new service monitor can then be activated to monitor any applicable devices, applications or servers in the network.” ’898 Patent at 8:52–54. Claim 6 further recites that the “service monitor” is used “to periodically collect data of the defined data types from the components.” Similarly, claim 3 recites “using the service monitor to periodically collect the performance data.” Accordingly, the Court finds that the recited “service monitor” is a script-based program used by the performance management system to monitor a device, application, or server.

Turning to the parties’ constructions, the Court does not adopt Defendant’s construction because it requires a program “running on the performance management system.” As discussed above, the intrinsic evidence indicates that the “service monitor” is a script-based program used by the performance management system. However, to the extent that a party contends that “running on” is different than “used by,” the Court rejects this argument. Furthermore, claim 3 recites “integrating the at least one script-based program into the performance management system as a service monitor.” Likewise, claim 6 recites “integrating the program to the performance management system as a service monitor." Thus, the Court is not persuaded that it should further specify that the "service monitor" is "running on the performance management system."

The Court also does not adopt Defendant's construction because it requires "automatically collects user-defined data from the components of the network." The claim language recites what data the recited "service monitor" collects. For example, claim 3 recites "using the service monitor to periodically collect the performance data." Likewise, claim 6 recites "using the service monitor to periodically collect data of the defined data types from the components." Thus, it would be redundant to state that the service monitor "collects user-defined data from the components of the network." More importantly, the claim language recites that the data is collected "periodically." Thus, the Court is not convinced that the construction needs to state that the data is collected "automatically."

However, to the extent that Plaintiff contends that once the recited "service monitor" is activated, the periodic data collection requires further user input (*i.e.*, the periodic data collection is not done "automatically"), the Court rejects this argument. The specification clearly states that "[t]he system provides the information from the user to the system such that the system can automatically call the service monitor and run it periodically." '898 Patent at 10:46–48. Likewise the specification states that "[a]s soon as the new service monitor is added and activated, the network performance management system automatically uses it to collect the customized data from the components of the network, and apply all its core functions to the customized data." *Id.* at 11:13–17. Thus, the specification's discussion of including the service monitor in a list before it is activated does not indicate that the periodic collection is not done automatically. Instead, it only indicates that the performance management system is not yet using the recited "service monitor," as required by the claims. Indeed, the specification states that "[o]nce the monitor is activated, the network performs [sic] monitoring and management system runs it periodically to collect the type of data defined in the service monitor." *Id.* at 10:67–11:3.

Additionally, the specification states that "the meta data may indicate that an alarm is to be associated with the data" and that "an alarm signal may be communicated to alarm generator 40." *Id.* at 4:17–19, 7:4–5. The specification further states that "[i]n response to the alarm signal, alarm generator 40 may generate an e-mail message to a network administrator or other personnel, initiate a page to a network administrator's pager, or communicate the alarm information to another system or application." *Id.* at 7:5–10. This further indicates that after the service monitor is activated, further user input is not required because it runs automatically and generates an alarm when an issue arises (i.e., without human input).

Turning to Plaintiff's construction, the Court finds that it fails to indicate that the recited "service monitor" is a program used by the performance management system. As discussed above, the specification states "[o]ne advantage of the network performance monitoring and management system is to make it easier to integrate individual script-based programs with the performance management system." '898 Patent at 11:55–58. "Thus, the user can have network performance monitoring and management to handle his business-oriented data in the same way as the network infrastructure data, meta data." *Id.* at 11:46-49. As this intrinsic evidence indicates, the claims require that the performance management system use the program to monitor a device, application, or server. Plaintiff's construction fails to capture this aspect of the recited "service monitor."

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the term **"service monitor"** to mean **"script-based program used by the performance management system to monitor a device, application, or server."**

C. The '594 Patent

The parties' dispute focuses on the meaning and scope of three terms/phrases in the '594 Patent.

1. "interpreting the instructions"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"interpreting the instructions"	plain meaning. Alternatively: "using the instructions to execute commands"	"translating and executing the instructions one at a time"

a) The Parties' Position

The parties dispute whether phrase "interpreting the instructions" should be construed as "using the instructions to execute commands," as Plaintiff proposes, or as "translating and executing the instructions one at a time," as Defendant proposes. Plaintiff contends that its construction is supported by the specification. (Dkt. No. 99 at 5) (citing '594 Patent at 9:12–14). Plaintiff further contends that its construction is further supported by dependent claim 7, which recites that the software can be "directly executable by the server computer system without interpretation or compilation." (Dkt. No. 99 at 5) (citing '594 Patent at 10:11–14). Plaintiff argues that Defendant's construction is based entirely on extrinsic evidence, and introduces an undefined concept, "executing [] instructions one at a time," which is not discussed anywhere in the intrinsic record and would serve only to confuse the jury as to when one instruction starts and another begins. (Dkt. No. 99 at 6.)

Defendant responds that the patent explains that the script is "interpreted" by a "script program interpreter 66," but does not describe how interpretation takes place. (Dkt. No. 106 at

17) (citing ’594 Patent at 6:43–50). Defendant argues that a software program known as an “interpreter” executes (runs) a program written in a high-level programming language by reading in an instruction, translating it and then executing it before moving on to the next instruction. (Dkt. No. 106 at 17.) Defendant argues that this plain understanding is consistent with at least four contemporaneous technical dictionaries. (Dkt. No. 106 at 17–18.)

Defendant argues that Plaintiff’s construction could describe any process of executing computer program instructions, including direct execution of native instructions that involves no interpretation whatsoever. (Dkt. No. 106 at 18.) Defendant contends that Plaintiff seeks to equate “interpretation” with any form of computer instruction execution. (*Id.*)

Plaintiff replies that Defendant improperly relies on extrinsic evidence for its “one at a time” limitation, and ignores that the specification demonstrates that interpreters take actions simply by using instructions. (Dkt. No. 108 at 3) (citing ’594 Patent at 9:12–14). Plaintiff argues that Defendant’s “one at a time” construction is found nowhere in the intrinsic record, and introduces ambiguity for a skilled artisan. (Dkt. No. 108 at 3.) Plaintiff further argues that “using” instructions to execute commands per its construction is different than executing the instructions directly. (*Id.*)

For the following reasons, the Court finds that the phrase **“interpreting the instructions”** should be construed to mean **“translating and executing the instructions with an interpreter.”**

b) Analysis

The phrase “interpreting the instructions” appears in claim 1 of the ’594 Patent. The Court finds that the phrase is used consistently in the claims and is intended to have the same meaning in each claim. During the claim construction hearing, the parties agreed that "interpreting" means translating and executing. This is consistent with the intrinsic evidence that describes software systems that "make use of script programs written in a high-level interpretable language in order to execute certain procedures." '594 Patent at 2:26–29. The specification further states that "a script program written in an interpretable language can be used to define a command or routine, such as (in this example) a routine for collecting information and determining the number of users logged into a particular server computer system as well as the number of processes per user." *Id.* at 6:51–56. Thus, the Court finds that "interpreting" means "translating and executing." This is also consistent with the extrinsic evidence submitted by Defendant, which includes numerous references to "translating and executing." (Dkt. No. 106 at 17–18.)

The Court further finds that the intrinsic evidence indicates that "interpreting the instructions" means "translating and executing the instructions with an interpreter." For example, the specification states that "[e]xecution continues with step 184 in which script program interpreter 66 interprets the script program, thereby taking the desired recovery action." '594 Patent at 9:12–14. Likewise, the specification states that "[s]cript program compiler 64 is responsible for compiling script programs. Such compilation is only partial, however, resulting in an intermediate code that is not directly executable, but that is interpretable by script program interpreter 66." '594 Patent at 4:46–51. Accordingly, the specification indicates that "interpreting the instructions" requires an interpreter. This is also consistent with the extrinsic evidence submitted by Defendant. (Dkt. No. 106 at 17–18.)

Regarding Defendant's "one at a time" language, the Court finds that it is not discussed anywhere in the intrinsic record and would only confuse the jury. For example, the specification describes that "[i]n a preferred embodiment of the invention, a script language was defined such that it could be partially compiled according to conventional methods into an intermediate form." '594 Patent at 6:38–41. The specification adds that "[i]nterpretable languages that are capable of being compiled into such an intermediate form may be interpreted more quickly than languages that must be interpreted from ASCII text." *Id.* at 6:41–44. Thus, the specification provides examples where "interpreting the instructions" includes starting with an intermediate form that is partially compiled. This intrinsic evidence indicates that the dictionary definitions distinction between "interpreting" and "compiling" becomes blurred, and determining "what constitutes an 'instruction' and where one 'instruction' begins and ends" would introduce unnecessary ambiguity into the claims. Accordingly, the Court does not adopt this portion of Defendant's construction.

Likewise, the Court does not adopt Plaintiff's construction because it could potentially describe any process of executing computer program instructions, including ones without an interpreter. Plaintiff argues that "'interpreting the instructions' means using instructions to execute commands rather than executing the commands directly." (Dkt. No. 108 at 3.) The Court finds that the actual wording of Plaintiff's construction is not so limiting. Therefore, the Court's construction specifies that what "uses" the instruction is the interpreter. Accordingly, the Court does not adopt Plaintiff's construction.

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the phrase **"interpreting the instructions"** to mean **"translating and executing the instructions with an interpreter."**

2. "interpretable high-level computer programming language"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"interpretable high-level computer programming language"	plain meaning. Alternatively: "a computer language that provides a level of abstraction from the underlying machine language, and that can be translated and executed or compiled into an intermediate form and translated and executed"	No construction required if "interpreting the instructions" is construed. Alternatively: "a computer language that provides a level of abstraction from the underlying machine language, and that can be translated and executed one instruction at a time"

a) The Parties' Position

The parties agree that the phrase "interpretable high-level computer programming language" should be construed as "a computer language that provides a level of abstraction from the underlying machine language, and that can be translated and executed." As with the previous term, the parties dispute whether it should further be construed as "translated and executed one instruction at a time," as Defendant proposes. Plaintiff contends that it should be construed as "translated and executed or compiled into an intermediate form and translated and executed." Plaintiff argues that its construction is entirely consistent with the '594 specification. (Dkt. No. 99 at 6) (citing '594 Patent at 6:38–41). Plaintiff also provides a dictionary definition for the term "high-level language." (Dkt. No. 99 at 6.)

Plaintiff further argues that Defendant's "executing [] instructions one at a time" language is based entirely on extrinsic evidence and improper for the same reasons it argued for above. (*Id.*) Plaintiff contends that its construction contemplates interpretable high-level computer programming languages that can be compiled into an intermediate form before translation and execution, as depicted in Figure 10, step 182. (*Id.* at 6) (citing '594 Patent at 9:9–14).

Defendant responds that the only dispute pertains to the term "interpretable." (Dkt. No. 106 at 19.) Defendant argues that it is not necessary for the Court to construe "interpretable," if the Court construes "interpreting the instructions." (*Id.*) Defendant contends that if the Court does provide a construction, its construction incorporates the correct view of translating and executing the instructions, as explained above. (*Id.*)

For the following reasons, the Court finds that the phrase **"interpretable high-level computer programming language"** should be construed to mean **"a computer language that provides a level of abstraction from the underlying machine language, and can be translated and executed by an interpreter."**

b) Analysis

The phrase "interpretable high-level computer programming language" appears in asserted claim 1 of the '586 Patent. The parties agree that phrase should be construed as "a computer language that provides a level of abstraction from the underlying machine language, and that can be translated and executed." The Court finds that this is consistent with the intrinsic and extrinsic evidence. For example, the Microsoft Computer Dictionary defines "high-level language" as "[a] computer language that provides a level of abstraction from the underlying machine language." (Dkt. No. 99-3 at 5 (Microsoft Computer Dictionary, (5th ed. 2002)). Thus, the Court finds that "interpretable high-level computer programming language" means "a computer language that provides a level of abstraction from the underlying machine language, and can be translated and executed."

The Court further finds that the intrinsic evidence indicates that the phrase "interpretable high-level computer programming language" means the language "is translated and executed by an interpreter." For example, the specification states that "[e]xecution continues with step 184 in which script program interpreter 66 interprets the script program, thereby taking the desired recovery action." '594 Patent at 9:12–14. Likewise, the specification states that "[s]cript program compiler 64 is responsible for compiling script programs. Such compilation is only partial, however, resulting in an intermediate code that is not directly executable, but that is interpretable by script program interpreter 66." *Id.* at 4:46–51. Accordingly, the specification indicates "interpretable high-level computer programming language" means the language "is translated and executed by an interpreter."

For the reasons discussed above, the Court does not adopt Defendant's language of "translated and executed one instruction at a time." The Court also does not adopt Plaintiff's construction because the additional language is unnecessary. As discussed above, the specification states that "[i]n a preferred embodiment of the invention, a script language was defined such that it could be partially compiled according to conventional methods into an intermediate form." '594 Patent at 6:38–41. Thus, the Court agrees that the intrinsic evidence indicates that the interpretable high-level computer programming languages can be compiled into an intermediate form before translation and execution. However, the Court finds this embodiment is included within the scope of the Court's construction, and that this additional language is not necessary and could be confusing to a jury.

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the phrase **"interpretable high-level computer programming language"** to mean **"a computer language that provides a level of abstraction from the underlying machine language, and can be translated and executed by an interpreter."**

3. "uninterpreted form" and "stored on the storage device in their uninterpreted form"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"stored on the storage device in their uninterpreted form"	construe "uninterpreted form." Otherwise, plain meaning.	stored as a text file containing high level programming language instructions
"uninterpreted form"	plain meaning. Alternatively: a form not interpreted	See construction of "stored . . . in . . . uninterpreted form"

a) The Parties' Position

The parties dispute whether the term "uninterpreted form" should be construed as "a form not interpreted," as Plaintiff proposes, or as "a text file containing high level programming language instructions," as Defendant proposes. Plaintiff contends that a person of ordinary skill in the art would understand "uninterpreted form" to mean "a form not interpreted." (Dkt. No. 99 at 7.) Plaintiff argues that Defendant's construction is directly contrary to the intrinsic evidence. (*Id.*) Plaintiff contends that the patent and file history discuss storing intermediately-compiled versions of the computer programming language, and the advantages of storing compiled versions as well, while indicating that storing as text files is merely "preferabl[e]." (*Id.*) (citing '594 Patent at 6:47–49, 6:34-38, (Dkt. No. 99-4 at 7 ('594 FH at 9/20/96 Amendment))).

Defendant responds that a person of ordinary skill in the art would understand "stored . . . in . . . uninterpreted form" to mean "stored as a text file containing high level programming language instructions." (Dkt. No. 106 at 19.) Defendant contends that the '594 specification description explains that high-level language instructions can be stored in two different formats: (1) "uninterpreted form" and (2) "intermediate form." (*Id.* at 20-21) (citing '594 Patent at 6:34–50). According to Defendant, the patent therefore draws a distinction between "uninterpreted form" which is "preferably in the form of an ASCII text file," and "intermediate form" which is partially compiled. (Dkt. No. 106 at 20.)

Regarding Plaintiff's construction, Defendant argues that the portion of the specification that Plaintiff relies on makes clear that “intermediate form” is distinct from and is an alternative to “uninterpreted form.” (*Id.* at 20) (citing ’594 Patent at 6:41–44). According to Defendant, one of ordinary skill in the art would not understand “uninterpreted form” to subsume “intermediate form” in light of the clear distinction between those two forms of storage as described in the specification. (Dkt. No. 106 at 20.) Defendant contends that if the two forms were intended to be coextensive, a person of ordinary skill in the art would have expected to see a statement to that effect in the specification. (*Id.*)

Plaintiff replies that Defendant’s construction is wrong because it would exclude a preferred embodiment where an “intermediate form” is a type of uninterpreted form that is not ASCII text. (Dkt. No. 108 at 3) (citing ’594 Patent at 6:34–50).

For the following reasons, the Court finds that the term **“uninterpreted form”** should be construed to mean **“form requiring an interpreter to translate and execute.”** The Court further finds that the phrase **“stored on the storage device in their uninterpreted form”** should be given its plain and ordinary meaning.

b) Analysis

The term “uninterpreted form” appears in asserted claim 1 of the ’594 Patent. The Court further finds that the specification indicates that “uninterpreted form” means a form requiring an interpreter to translate and execute. Specifically, the specification describes two embodiments of the recited “instructions” stored in an “uninterpreted form.” The first is a preferred embodiment where the instructions are stored in the form of an ASCII text file. ’594 Patent at 6:34–38. The specification also describes a second preferred embodiment where the instructions are “defined such that it could be partially compiled according to conventional methods into an intermediate form.” *Id.* at 6:39–41. The specification states that “[i]nterpretable languages that are capable of being compiled into such an intermediate form may be interpreted more quickly than languages that must be interpreted from ASCII text." '*Id.* at 6:41–44, *see also*, (Dkt. 99-4 at 9 ('594 FH at 9/20/96 Amendment)). For either of these "uninterpreted forms," the specification states that "script program interpreter 66" is required to interpret the script program. '594 Patent at 9:11–14. Accordingly, the Court finds that the term "uninterpreted form" means "form requiring an interpreter to translate and execute."

Turning to the parties' constructions, the Court does not adopt Defendant's construction because it improperly limits the scope of the claims to "text files." As discussed above, the specification disclose two embodiments for storing the recited instructions in "uninterpreted form." Defendant characterizes the first embodiment as the "uninterpreted form" and the second embodiment as an "intermediate form." According to Defendant, a person of ordinary skill in the art would not understand "uninterpreted form" to subsume "intermediate form." (Dkt. No. 106 at 20.) The Court disagrees with Defendant's characterization. As indicated in the specification, both of these forms are "uninterpreted" because they require an interpreter to translate and execute the instructions. '596 Patent at 4:46–51 ("Such compilation is only partial, however, resulting in an intermediate code that is not directly executable, but that is interpretable by script program interpreter 66."). Accordingly, the Court rejects Defendant's construction because it would exclude a preferred embodiment. *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1583 (Fed. Cir. 1996) (stating that it is "rarely, if ever, correct" to interpret a claim to exclude a preferred embodiment). The Court does not adopt Plaintiff's construction because it would not be helpful to a jury.

Regarding the phrase "stored on the storage device in their uninterpreted form", the Court finds that the phrase should be given its plain and ordinary meaning in light of the Court's construction for “uninterpreted form.” Given this context, the phrase “stored on the storage device in their uninterpreted form” is unambiguous, is easily understandable by a jury, and requires no construction. Therefore, the phrase will be given its plain and ordinary meaning.

c) Court’s Construction

In light of the evidence presented by the parties, the Court construes the term **“uninterpreted form”** to mean **“form requiring an interpreter to translate and execute.”** The phrase “stored on the storage device in their uninterpreted form” will be given its **plain and ordinary meaning.**

D. The ’683 Patent

The parties’ dispute focuses on the meaning and scope of six terms/phrases in the ’683 Patent.

1. “node” and “nodes”

Disputed Term	Plaintiff’s Proposal	Defendant’s Proposal
“node”	“operatively coupled monitored component”	“representation of a condition in a fault model”
“nodes”	“operatively coupled monitored components”	

a) The Parties’ Position

The parties dispute whether the term “node” should be construed as a “operatively coupled monitored component,” as Plaintiff proposes, or as a “representation of a condition in a fault model,” as Defendant proposes. Plaintiff argues that its construction has express definitional support in the specification. (Dkt. No. 99 at 14) (citing ’683 Patent at claim 47, 2:53–59). Plaintiff contends that Defendant’s construction is inconsistent with the specification. (Dkt. No. 99 at 14.) Plaintiff argues that the claims recite limitations such as “receive an event notification from a first node,” and Defendant has not explained how a notification could be received from a “condition” per Defendant’s construction of “node.” (Dkt. No. 99 at 14) (citing ’683 Patent at 13:16–18). Plaintiff also argues that the specification teaches that nodes may “generat[e] alarm events” and “receiv[e] event notifications,” and Defendant has not explained how a “condition” generates alarm events or receives event notifications. (Dkt. No. 99 at 14) (citing ’683 Patent at Abstract, 4:53–59).

Plaintiff further argues that although that patent teaches one embodiment where a node may “represent a ‘condition’ of a modeled component,” nodes still must be “operatively coupled monitored components.” (Dkt. No. 99 at 14) (citing ’683 Patent at 3:59–62, 2:53–59, 9:18–22). Plaintiff also contends that importing “in the fault model” into the definition of “node” would create redundancy in the claim language and confusion because the claim language recites “node in a fault model” or “fault model having a plurality of nodes” (Dkt. No. 99 at 14) (citing ’683 Patent at 17:31–32, 11:51; 15:64–65). Plaintiff also argues that “nodes” are “operatively coupled monitored components” not limited to existing “in the fault model.” (Dkt. No. 99 at 14–15) (citing ’683 Patent at Abstract).

Defendant responds that one of ordinary skill in the art would have understood “node” to mean a “representation of a condition in a fault model.” (Dkt. No. 106 at 22) (citing ’683 Patent at 2:25–29, 3:60–62). Defendant argues that “node” has an accepted meaning in the prior art, and that the inventors did not intend to depart from that meaning of this term. (Dkt. No. 106 at 22.) Defendant contends that Plaintiff’s proposal confuses the condition of the component, the “node,” with the component itself. (*Id.* at 23.) Defendant argues that the specification describes a fault model for a single Automatic Teller Machine (ATM) that includes a number of “nodes” indicating, for example, that the ATM has no money or has a paper jam. (*Id.*) (citing ’683 Patent at Figure 6A). According to Defendant, these nodes do not represent separate ATMs, but conditions applicable to a particular ATM. (Dkt. No. 106 at 23.)

Plaintiff replies that Defendant's attempt to distance the asserted claims from claim 47 is unavailing, in view of the comparable claim language. (Dkt. No. 108 at 7) (citing '683 Patent at claim 47 and claim 24). Plaintiff also argues that Defendant's construction is inconsistent with other intrinsic evidence, including "receive an event notification from a first node" and "node generating an alarm event." (Dkt. No. 108 at 7) ('683 Patent at 13:16–18). Plaintiff further argues that Defendant does not explain how a "condition" (Defendant's "node") communicates notifications or generates alarms. (Dkt. No. 108 at 7.) Plaintiff contends that nodes also are "operatively coupled." (*Id.*) (citing '683 Patent at 2:53–59, 9:18–22, 13:16–18). According to Plaintiff, Defendant's argument that the nodes in the ATM embodiment are not separate ATMs misses the point, because these nodes are still "operatively coupled monitored components" that receive alarms and communicate information. (Dkt. No. 108 at 7.) Finally, Plaintiff argues that Defendant's construction for "node" is not conducive to the plural form "nodes" that appears in the claims. (*Id.*)

For the following reasons, the Court finds that the term **"node"** should be construed to mean **"representation of a status or condition of a monitored component;"** and the term **"nodes"** should be construed to mean **"a plurality of representations of status or condition of one or more monitored components."**

b) Analysis

The terms "node" and "nodes" appear in asserted claims 1, 24, 56, and 79 of the '683 Patent. The Court finds that the terms are used consistently in the claims and are intended to have the same meaning in each claim. All of the independent claims recite either a "fault model having a plurality of nodes" or "a plurality of nodes in an enterprise-specific fault model." Thus, the claims indicate that the recited "nodes" are nodes in a fault model. This is confirmed by the specification when it states that "the enterprise (or that portion of the enterprise being monitored) is represented by a fault model having a plurality of communicatively coupled nodes." '683 Patent at 2:57–59. The specification further indicates that nodes represent the status or condition of a monitored component by stating the following:

> The following embodiments of the invention, described in terms of model based reasoning approaches using object-oriented characterizations of monitored components, are illustrative only and are not to be considered limiting in any respect. Specifically, the following embodiments of the invention utilize an object-oriented modeling approach for characterizing: (1) monitored components; (2) their physical and/or logical connectivity and (3) the propagation of detected anomalies or faults. In this approach, each component (hardware, software and logical) that is to be monitored is defined by a software object that characterizes that object in terms of its function and possible relationships with other modeled objects. The collection of all such object definitions is referred to as the Management Schema.
>
> . . .
>
> Based on the Management Schema, a Fault Model (a directed graph or digraph) may be determined, wherein each node in the Fault Model represents a "condition" of a modeled component. Thus, if a single managed object (i.e., an object in the Management Schema) is characterized by a plurality of conditions, it may be represented by a plurality of nodes in a Fault Model. The topology or architecture of a specific Fault Model is referred to as an Impact Graph.

Id. at 3:40–67. The specification further describes an "inference policy" as "a rule, or set of rules, for inferring the status or condition of a fault model node based on the status or condition of the node's immediately down-stream neighboring nodes, wherein a down-stream node is a node that is coupled to another (digraph) node in the direction of information flow." *Id.* at 4:11–17. Similarly, the specification describes an "impact policy" as a "rule, or set of rules, for assessing the impact on a fault model node based on the status or condition of the node's immediately up-stream neighboring nodes, wherein an up-stream node is a node that is coupled to another (digraph) node in the direction against information flow." *Id.* at 4:17–22. In each of these portions of the specification the nodes are described as representing the status or condition of a monitored component.

Indeed, the specification describes that "[t]ypically, nodes in a fault model in accordance with the invention utilize a status value (e.g., a Boolean value to indicate whether a node is failed or not-failed, or a real number such as 0.0 to 1.0) to record a node's status or condition and an impact value (e.g., a Boolean value to indicate whether a node is impacted or not-impacted by its up-stream neighbors, or a real number such as 0.0 to 1.0) to record the node's impact value." *Id.* at 4:22–30. The specification further states that Figure 6 illustrates a "Fault Model 600" for a single Automatic Teller Machine (ATM) that includes a number of condition "nodes." *Id.* at 8:1–18. These condition nodes may indicate that the ATM has no money or has a paper jam. *Id.* Thus, a person of ordinary skill in the art would understand that the recited "nodes" are representations of status or condition of one or more monitored components.

Plaintiff contends that claim 47 defines "nodes" as "operatively coupled monitored components." (Dkt. No. 99 at 14.) The Court notes that this "definition" appears in the preamble of claim 47, and that the body of claim 47 recites that the "first node" is "one of a plurality of nodes in an enterprise-specific fault model," and that "nodes" are identified "in a contiguous path between the second node and the first node in the fault model." Thus, consistent with the other independent claims and the specification, claim 47 indicates that the recited "nodes" are nodes in a fault model. As described in the specification, the recited "nodes" are representations of status or condition of one or more monitored components. Accordingly, the Court does not adopt Plaintiff's construction because it too broad and construes "nodes" as simply a "monitored component," thereby potentially removing it from the context of the recited "fault model."

The Court does not adopt Defendant's construction because it does not connect the representation to a monitored component. Plaintiff argues that Defendant has not explained how a notification could be received from a “condition,” or how a “condition” generates alarm events. (Dkt. No. 99 at 14.) Unlike Defendant’s construction, the Court’s construction requires the “node” to be a representation of a status or condition of a monitored component. As discussed above, the specification states that “the following embodiments of the invention utilize an object-oriented modeling approach for characterizing: (1) monitored components; (2) their physical and/or logical connectivity and (3) the propagation of detected anomalies or faults.” ’683 Patent at 3:44–48. It is through the modeling approach of using “nodes” that notifications are received and/or alarms are generated. The Court also does not adopt Defendant’s construction because it includes the redundant “fault model” language. As Plaintiff argues, the claims recite “fault model,” and the Court finds including it in the construction would confuse the claim language.

c) Court’s Construction

In light of the evidence presented by the parties, the Court construes the term **“node”** to mean **“representation of a status or condition of a monitored component;”** and the term **“nodes”** to mean **“a plurality of representations of status or condition of one or more monitored components.”**

2. “fault model” and “fault model having a plurality of nodes”

Disputed Term	Plaintiff’s Proposal	Defendant’s Proposal
“fault model”	“directed graph (aka digraph) for monitoring faults”	“a directed graph (aka digraph) with nodes representing conditions of modeled components”
“fault model having a plurality of nodes”	“directed graph (aka digraph) for monitoring faults, having a plurality of operatively coupled monitored components”	construe in accordance with the included terms “fault model” and “nodes”

a) The Parties' Position

The parties agree that "fault model" is a "directed graph (aka digraph)." The parties dispute whether the term "fault model" should be construed to include "nodes representing conditions of modeled components," as Defendant proposes. Plaintiff argues that Defendant's construction incorrectly limits "nodes" to "representing conditions of modeled components," which is inconsistent with the patentee's express definition of "nodes." (Dkt. No. 99 at 15.) Plaintiff also argues that Defendant's construction is erroneous because incorporating "nodes" into the definition of "fault model" would be redundant of the claim language and would create confusion because Defendant defines "node" and "fault model" with respect to each other. (Dkt. No. 99 at 15.)

Defendant responds that the specification makes clear that a fault model is a directed graph with nodes representing conditions of modeled components. (Dkt. No. 106 at 23) (citing '683 Patent at 3:59–62). Defendant contends that the "nodes" are an integral part of the "fault model" and how it monitors faults. (Dkt. No. 106 at 23.) Defendant argues that its construction does not exclude a computer-implemented fault model, it simply does not require one. (*Id.*)

Plaintiff replies that Defendant's construction limits "nodes" to "representing conditions of modeled components," which is inconsistent with the patentee's definition of "nodes" and other intrinsic evidence. (Dkt. No. 108 at 8.) Plaintiff further contends that incorporating "nodes" into the definition of "fault model" is redundant of the claim language and creates confusion. (*Id.*)

For the following reasons, the Court finds that the term **"fault model"** should be construed to mean **"a directed graph (aka digraph) used for monitoring, diagnosis, and recovery of error conditions."** The Court further finds that the phrase **"fault model having a plurality of nodes"** should be given its **plain and ordinary meaning.**

b) Analysis

The term “fault model” appears in asserted claims 1, 24, 56, and 79 of the ’683 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The phrase “fault model having a plurality of nodes” appears in asserted claims 1, 24, 56, and 79 of the ’683 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The Court further finds that the specification indicates that a “fault model” is “a directed graph or digraph.” ’683 Patent at 3:60–61, *see also, id.* at 2:25–29. This is consistent with the parties’ agreement for this term. The Court also finds that the specification indicates that a “fault model” is used for monitoring, diagnosis, and recovery of error conditions. *Id.* at 1:8–10. This is illustrated in Figure 6, which includes “Fault Model 600” for an ATM. *Id.* at 8:1–18. Specifically, Figure 6 illustrates a directed graph or digraph used for monitoring, diagnosis, and recovery of error conditions for an ATM.

Turning to the parties’ constructions, the Court does not adopt Defendant’s construction because it reads the term “nodes” into the construction of “fault model.” This is redundant of the claim language and would be confusing to a jury. As discussed above, the claims indicate that the recited “nodes” are nodes in a fault model, but “nodes” is not necessary for this construction because it is a separate term construed by the Court. The Court further clarifies Plaintiff’s construction by stating that the recited “fault model” is used for monitoring, diagnosis, and recovery of error conditions, as described in the specification. ’683 Patent at 1:8–10.

Regarding the phrase “fault model having a plurality of nodes,” the Court finds that it should be given its plain and ordinary meaning in light of the Court’s construction for “fault mode” and “nodes.” Given this context, the phrase “fault model having a plurality of nodes” is unambiguous, is easily understandable by a jury, and requires no construction. Therefore, the phrase will be given its plain and ordinary meaning.

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the term **"fault model"** to mean **"a directed graph (aka digraph) used for monitoring, diagnosis, and recovery of error conditions."** The phrase **"fault model having a plurality of nodes"** will be given its **plain and ordinary meaning.**

3. "enterprise"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"enterprise"	"collection of hardware and software entities in a computer network"	"collection of components that can be monitored"

a) The Parties' Position

The parties dispute whether the recited "enterprise" should be construed as a "collection of hardware and software entities in a computer network," as Plaintiff proposes, or as a "collection of components that can be monitored," as Defendant proposes. Plaintiff argues its construction has express definitional support in the specification. (Dkt. No. 99 at 17) (citing '683 Patent at 1:45–50). Plaintiff argues that Defendant's construction ignores the patentee's definition and is generic with little meaning. (Dkt. No. 99 at 17.)

Defendant argues that Plaintiff's construction limits the enterprise "to hardware and software entities in a computer network," which improperly imports a limitation into the claims. (Dkt. No. 106 at 24.) Defendant further argues that the statement relied on by Plaintiff does not provide an absolute definition applicable in all instances, and other portions of the specification make clear that an enterprise can encompass entities or components beyond computer hardware and computer software. (*Id.*) Defendant argues that the specification describes using a fault model for Automatic Teller Machines (ATMs), and a component of those ATMs that can be monitored is a mechanical "binder" that dispenses bills to customers. (*Id.*) (citing '683 Patent at 7:31–34, 7:40–42, 8:12–14). According to Defendant, these mechanical devices are not computer hardware or software, but they are clearly intended to be covered. (Dkt. No. 106 at 24.) Defendant further argues that the specification also confirms that the claimed fault analysis method applies outside the computer context. (*Id.*) (citing '683 Patent at 11:32–40).

Plaintiff replies that its construction has definitional support in the specification. (Dkt. No. 108 at 8) ('683 Patent at 1:45–50). Plaintiff further argues that Defendant defines "enterprise" to not require hardware/software based on a passage that does not even recite "enterprise." (Dkt. No. 108 at 9) (citing '683 Patent at 11:32–40). Plaintiff contends that even if the passage is relevant, the system would still require computer hardware/software. (Dkt. No. 108 at 9.)

For the following reasons, the Court finds that the term **"enterprise"** should be construed to mean **"a collection of monitored components some of which may be hardware and some of which may be software."**

b) Analysis

The term "enterprise" appears in asserted claims 1 and 24 of the '683 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The Court further finds that the specification indicates that the recited "enterprise" is a collection of monitored components. Specifically, the specification states that "[i]n one embodiment the invention provides an enterprise fault analysis method, where an enterprise includes a plurality of monitored components some of which may be hardware and some of which may be software." '683 Patent at 2:53–56. Similarly, the specification states that "[c]ontemporary corporate computer networks comprise a plurality of different computer platforms and software applications interconnected through a number of different paths and various hardware devices . . . . The collection of such entities–hardware and software–is often referred to as an 'enterprise.'" *Id.* at 1:11–30. Accordingly, the Court finds that the recited "enterprise" is "a collection of monitored components some of which may be hardware and some of which may be software."

Defendant argues that the specification make clear that an enterprise can encompass entities or components beyond computer hardware and computer software. (Dkt. No. 106 at 24.) The Court agrees. For example, the specification includes an example of an enterprise consisting of a plurality of Automatic Teller Machines (ATMs). '683 Patent at 7:31–33. In this example, one of the monitored components is a mechanical "binder" that dispenses bills to customers. *Id.* at 7:40–44 ("Binder object 415 represents a money dispensing 40 mechanism (each ATM machine typically includes as many binders as types of bills that it dispenses) . . . ." To monitor the binder component, the specification states that "[b]inder NO_MONEY condition node 630 is coupled to ATM NO_MONEY condition node 635 through a HAS_MONEY IN relation." *Id.* at 8:12–14. To the extent that Plaintiff argues that this would not be considered a monitored component within the recited "enterprise," the Court rejects this argument. Indeed, the specification explicitly states "[i]n still other embodiments, the system being monitored is not a computer network. For example, a mechanical system comprising pumps, valves and motors may benefit from the claimed fault analysis method." *Id.* at 11:32–35.

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the term **"enterprise"** to mean **"a collection of monitored components some of which may be**

hardware and some of which may be software.”

4. “up-stream,” “most up-stream,” and “down-stream”

Disputed Term	Plaintiff’s Proposal	Defendant’s Proposal
“up-stream”	“in the direction of cause”	“in the direction of one or more nodes that can impact a given node in the fault model”
“most up-stream”	“most in the direction of cause”	“having no nodes in the direction of nodes that can impact a given node in the fault model”
“down-stream”	“in the direction of impact”	“in the direction of one or more nodes that can be impacted by a given node in the fault model”

a) The Parties’ Position

The parties dispute whether the terms “up-stream” and “down-stream” should be construed as “in the direction of cause/impact,” as Plaintiff proposes, or as “in the direction of one or more nodes that can impact or be impacted by a given node,” as Defendant proposes. Plaintiff argues that its constructions are consistent with the intrinsic evidence, including the claim language. (Dkt. No. 99 at 17) (citing ’683 Patent at 11:54–67, 11:59–12:3). Plaintiff argues that Defendant’s constructions erroneously include “in the fault model,” which would be redundant of claim language and would add confusion. (Dkt. No. 99 at 18) (citing ’683 Patent at 11:54–55, 11:60–61).

Defendant responds that the patent uses the term “up-stream” with reference to the fault model. (Dkt. No. 106 at 25) (citing ’683 Patent at 4:20–22, 5:16–18). According to Defendant, an “up-stream” node is a node in the direction of one or more nodes that can impact a given node in the fault model. (Dkt. No. 106 at 25.) Defendant argues that the problem with Plaintiff’s construction is that the “cause” of a particular fault is unknown at the time the “up-stream analysis” commences. (*Id.*) Defendant contends that Plaintiff argues that the specification envisions scenarios in which independent “up-stream” nodes connect to a given node, but some of those nodes may not be a cause. (*Id.*)

Defendant further argues that the ’683 Patent states that “a down-stream node is a node that is coupled to another (digraph) node in the direction of information flow,” and that “if a node’s impact value is true, then it is impacted by one or more up-stream nodes.” (Dkt. No. 106 at 25) (citing ’683 Patent at 4:11–17, 5:16–18). Defendant argues that Plaintiff’s construction for “down-steam” is incorrect because there may be situations where at least some of the down-stream nodes may not be impacted. (Dkt. No. 106 at 25.) According to Defendant, it makes no sense to construe “down-stream” as “in the direction of impact” when there may be no impact. (*Id.*)

Plaintiff replies that Defendant’s construction includes limitations redundant of claim language. (Dkt. No. 108 at 9.) Plaintiff further argues that the claims recite “performing an up-steam analysis” and reporting a “root cause” node. (*Id.*) (citing ’683 Patent at 11:54–55). According to Plaintiff, this indicate that “up-stream” in the claim is “in the direction of cause.” (Dkt. No. 108 at 9.) Plaintiff further argues that Defendant’s construction of “up-stream” is also unworkable for “most up-stream.” (*Id.*) Finally, Plaintiff argues that the “down-stream” disputes mirror those for “up-stream.” (*Id.*)

For the following reasons, the Court finds that the term **“up-stream”** should be construed to mean **“in the direction against information flow;”** and the term **“most up-stream”** should be construed to mean **“furthest in the direction against information flow.”** The Court finds that the term **“down-stream”** should be construed to mean **“in the direction of information flow.”**

b) Analysis

The term “up-stream” appears in asserted claims 1, 3, 12, 24, 26, 35, 56, 58, 67, 79, and 88 of the ’683 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The term “most up-stream” appear in asserted claims 12, 35, and 67 of the ’683 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The term “down-stream” appear in asserted claims 1, 14, 24, 37, 56, 69, 79, and 89 of the ’683 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim.

The Court further finds that the intrinsic evidence indicates that “up-stream” means “in the direction against information flow.” For example, the specification states that “an up-stream node is a node that is coupled to another (digraph) node in the direction against information flow.” ’683 Patent at 4:20–22. This indicates that “up-stream” is “in the direction against information flow.” Accordingly, “most up-stream” means “furthest in the direction against flow.” The Court also finds that the intrinsic evidence indicates that “down-stream” means “in the direction of information flow.” For example, the specification states that “a down-stream node is a node that is coupled to another (digraph) node in the direction of information flow.” *Id.* at 4:15–17. This indicates that “down-stream” is “in the direction of information flow.”

Regarding the parties’ construction, the Court does not adopt Plaintiff’s construction for “up-stream” because the “cause” is less clear than the Court’s construction. Likewise, the Court does not adopt Plaintiff’s construction for “down-stream” because “impact” is less clear than the Court’s construction. The Court does not adopt Defendant’s construction because it is confusing and redundant of the claim language.

c) Court’s Construction

In light of the evidence presented by the parties, the Court construes the term **“up-**

stream" to mean **"in the direction against information flow;"** and the term **"most up-stream"** to mean **"furthest in the direction against information flow."** The Court construes the term **"down-stream"** to mean **"in the direction of information flow."**

5. "a root cause"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"a root cause"	"a fundamental source of a problem"	"a node with no up-stream nodes in the fault model indicating a failed status"

a) The Parties' Position

The parties dispute whether the term "root cause" should be construed as "a fundamental source of a problem," as Plaintiff proposes, or as "a node with no up-stream nodes in the fault model indicating a failed status," as Defendant proposes. Plaintiff contends that its construction has express definitional support in the specification. (Dkt. No. 99 at 16) (citing '683 Patent at 1:42–47). Plaintiff further contends that its construction is consistent with U.S. Patent No. 6,072,777 cited on the cover of the '683 Patent. (*Id.*) Plaintiff argues that Defendant's construction ignores the specification definition and attempts to limit the term based on the specification's observation that "the most up-stream nodes having a status value indicative of failure are identified as the 'root causes' of the alarm event…" (Dkt. No. 99 at 16–17) (quoting '683 Patent at 10:34–38). According to Plaintiff, this passage does not define "root cause," but rather, observes where in one embodiment "root causes" may be identified in an impact graph. (Dkt. No. 99 at 17.)

Defendant responds that its construction is consistent with the specification and the claim language. (Dkt. No. 106 at 26) (citing '683 Patent at 10:32–38, 4:63–65). Defendant argues that Plaintiff's construction is incorrect because it takes the "root cause" out of the context of a fault model. (Dkt. No. 106 at 26.) Defendant contends that the "root cause," in both the claims and the specification, is a specific node in the fault model identified through up-stream analysis. (*Id.*) According to Defendant, whether that identified node is a "fundamental source of a problem" is unknown, and a question outside the scope of the fault model itself. (*Id.*) Defendant further argues that Plaintiff's construction is also objectionable since it is based on statements in an unrelated patent from a different field. (*Id.*)

Plaintiff replies that Defendant's argument that "root cause" must be defined by a fault model ignores the patentees' chosen definition and other teachings that the patent aims to identify the "root cause" or fundamental source of a problem. (Dkt. No. 108 at 8) (citing '683 Patent at 1:42–47, 1:21–52). Plaintiff argues that Defendant's construction is unnecessarily limiting and based on a passage that does not define "root cause," but rather provides an observation about one embodiment where the "root causes" may be identified in an impact graph. (Dkt. No. 108 at 8.) (citing '683 Patent at 10:34–38). Plaintiff further argues that Defendant's argument that intrinsic evidence (the '777 Patent) cited by the examiner during prosecution should be ignored is unavailing. (Dkt. No. 108 at 8).

For the following reasons, the Court finds that the term **"root cause"** should be construed to mean **"most up-stream node or nodes having a status value indicative of failure."**

b) Analysis

The term "root cause" appears in asserted claims 1, 21, 24, 44, 56, 76, and 79 of the '683 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The intrinsic evidence indicates that the term "root cause" means "most up-stream node or nodes having a status value indicative of failure." For example, claim 1 recites "performing an up-stream analysis of the fault model beginning at the first node; identifying a second node, the second node having a status value modified during the up-stream analysis to indicate a failed status." Claim 1 further recites "reporting the second node as a root cause of the received event notification." Consistent with the claims, the specification states "the most up-stream nodes having a status value indicative of failure are identified as the 'root cause' of the alarm event . . ." '683 Patent at 10:34–36. Similarly, the specification states "those furthest up-stream nodes in the Impact Graph having a status value indicative of failure are identified as 'root causes.'" *Id.* at 4:63–65. Accordingly, the Court finds that a person of ordinary skill in the art would understand "root cause" to mean "most up-stream node or nodes having a status value indicative of failure."

Turning to the parties' construction, the Court does not adopt Plaintiff's construction because it takes the "root cause" out of the context of a fault model. As discussed above, all of independent claims indicate that the recited "nodes" are nodes in a fault model. Thus, the "root cause" in the claims is one or more nodes in the fault model identified through up-stream analysis. The portion of the specification cited by Plaintiff is in the Background section and describes a general scenario that is not in the context of a fault model. Accordingly, the Court finds that Plaintiff's construction is too broad because it reads the term "root cause" outside the context of a fault model. The Court does not adopt Defendant's construction because it does not track the language used in the intrinsic evidence as closely as the Court's construction.

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the term the term **"root cause"** to mean **"most up-stream node or nodes having a status value indicative of failure."**

6. "impact value"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"impact value"	"value representing the extent of impact"	"value indicating whether a node is impacted or not impacted by its up-stream neighboring nodes"

a) The Parties' Position

The parties dispute whether the term "impact value" is a "value representing the extent of impact," as Plaintiff proposes, or if it is a "value indicating whether a node is impacted or not impacted by its up-stream neighboring nodes," as Defendant proposes. Plaintiff contends that its construction is consistent with the intrinsic record and captures the fact that an "impact value" may be Boolean ("impacted" or "not impacted") or "any real number between 0.00 and 1.00," such as 0.51, representing a more precise extent (or degree) of impact. (Dkt. No. 99 at 18) (citing '683 Patent at 10:53–59). Plaintiff argues that Defendant's construction is erroneous to the extent it limits "impact value" to the Boolean embodiment ("impacted" or "not impacted") and excludes the real number embodiment. (Dkt. No. 99 at 18.)

Defendant responds that its construction for "impact value" comes directly from the specification. (Dkt. No. 106 at 26) (citing '683 Patent at 4:22–30). Defendant contends that Plaintiff's construction is overly limiting because the specification confirms that an "impact value" could be a simple Boolean value. (Dkt. No. 106 at 26–27) (citing '683 Patent at 5:13–16). According to Defendant, a Boolean value does not indicate the extent of impact, but simply whether a node is impacted. (Dkt. No. 106 at 27.) Defendant contends that its construction covers Boolean values, real values, or any other type of value indicating whether a node is impacted. (*Id.*)

Plaintiff responds that Defendant allows only one embodiment of "impact value," the Boolean embodiment, and improperly excludes the embodiment where an "impact value" may represent "any real number between 0.00 and 1.00," such as 0.51. (Dkt. No. 108 at 9) (citing

’683 Patent at 10:53–59). Plaintiff contends that its construction is consistent with all embodiments in the specification, including both Boolean and real number. (Dkt. No. 108 at 9.)

For the following reasons, the Court finds that the term **“impact value”** should be construed to mean **“value indicating whether a node is impacted or not impacted.”**

b) Analysis

The term “impact value” appears in asserted claims 1, 24, 56, and 79 of the ’683 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The Court further finds that the intrinsic evidence indicates that the term “impact value” means “value indicating whether a node is impacted or not impacted.” Specifically, the specification states the following:

> Typically, nodes in a fault model in accordance with the invention utilize a status value (e.g., a Boolean value to indicate whether a node is failed or not-failed, or a real number such as 0.0 to 1.0) to record a node’s status or condition and *an impact value* (e.g., a Boolean value *to indicate whether a node is impacted or not-impacted* by its up-stream neighbors, or a real number such as 0.0 to 1.0) to record the node’s impact value.

’683 Patent at 4:22–30 (emphasis added). As indicated above, an “impact value” is a “value indicating whether a node is impacted or not impacted.”

Plaintiff argues that not including “the extent of impact” in the construction limits the “impact value” to the Boolean embodiment (“impacted” or “not impacted”) and excludes the real number embodiment. (Dkt. No. 99 at 18.) The Court disagrees and finds that including “the extent of impact” could have the opposite effect of excluding the Boolean embodiment. This is because it could be argued that a Boolean value does not indicate “the extent of impact,” but instead only indicates whether a node is impacted. To the extent that a party argues that the Court’s construction excludes either the Boolean value or real number embodiment, the Court rejects this argument. Turning to Defendant’s construction, the Court does not adopt

Defendant's construction because it is not a concise as the Court's construction.

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the term **"impact value"** to mean **"value indicating whether a node is impacted or not impacted."**

E. The '093 Patent

The parties' dispute focuses on the meaning and scope of two terms/phrases in the '093 Patent.

1. "license certificate"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"license certificate"	"an indication of the right to deploy software in the environment managed by the CMDB"	"an indication of the right to deploy software"

a) The Parties' Position

The parties agree that the recited "license certificate" is "an indication of the right to deploy software." The parties dispute whether the construction for the term "license certificate" should also include "in the environment managed by the CMDB," as Plaintiff proposes. Plaintiff contends that its construction faithfully tracks the actual language used in the written description. (Dkt. No. 99 at 22) (citing '093 Patent at 8:61–63). Plaintiff argues that Defendant adopts only part of that language and truncates the rest. (Dkt. No. 99 at 22.)

Defendant responds that Plaintiff's construction improperly incorporates the phrase "in the environment managed by the CMDB." (Dkt. No. 106 at 27.) Defendant contends that this language was based on truncating the specification statement that "[a] license certificate indicates the right to deploy software in the environment managed by the CMDB server 110." (Dkt. No. 106 at 27) (citing '093 Patent at 8:61–63). Defendant argues that Plaintiff's construction replaces "CMBD server 110" with just "CMDB," rendering the construction nonsensical. (Dkt. No. 106 at 27.)

Defendant further argues that the "CMDB server 110" referenced in the actual sentence from the specification is not the same as a "CMDB" alone. (*Id.*) Defendant contends that the CMDB server is a component that actually "manages a collection of computer systems 120 for an enterprise 100." (Dkt. No. 106 at 27) (citing '093 Patent at 3:29–31). Defendant argues that a CMDB, is simply a database that stores data for the CMDB server. (Dkt. No. 106 at 27–28) (citing '093 Patent at 4:4–7). According to Defendant, this database has no ability to manage a computing environment as Plaintiff's construction requires. (Dkt. No. 106 at 28.) Defendant further argues that the claims do not recite a CMDB server or an "environment." (*Id.*)

Plaintiff replies that the "CMDB server 110" in a preferred embodiment is referred to generally as part of a "CMDB." (Dkt. No. 108 at 10–11.) Plaintiff argues that the paragraph cited by Defendant describes a "CMDB datastore 260," not the "CMDB" as a whole. (*Id.* at 11) (citing '093 Patent at 4:4–7).

For the following reasons, the Court finds that the term **"license certificate"** should be construed to mean **"an indication of the right to deploy software."**

b) Analysis

The term "license certificate" appears in asserted claim 1 of the '093 Patent. In describing "license certificate," the specification states that "[a]fter any new license types are created to handle the terms of the new software contracts terms, license certificates may be created in block 430, to link software contracts to CIs. A license certificate indicates the right to deploy software in the environment managed by the CMDB server 110." '093 Patent at 8:58–63. Based on this portion of the specification, the parties agree that the recited "license certificate" is "an indication of the right to deploy software." The Court agrees that this is the proper construction for the term "license certificate."

Plaintiff argues that the construction should further state that the recited "license certificate" is "an indication of the right to deploy software in the environment managed by the CMDB." Plaintiff contends that its construction faithfully tracks the actual language used in the written description. (Dkt. No. 99 at 22.) The Court disagrees and finds that Plaintiff's construction changes the actual language of "CMDB server" to "CMDB." Thus, if Plaintiff's construction faithfully tracked the written description it would include "CMDB server." However, the claims do not recite a "CMDB server." To avoid this issue, Plaintiff argues that the "CMDB server 110" in a preferred embodiment is referred to generally as part of a "CMDB." (Dkt. No. 108 at 11.) In other words, Plaintiff argues that a part should be read as the whole. The Court disagrees and finds that including the additional phrase proposed by Plaintiff confuses the claim language by reading in a limitation that is not required by the claims. Furthermore, the claims do not recite an "environment," and the Court finds that it would unnecessarily confuse the claim language.

The real issue appears to be whether the recited "CMDB" is simply a database that stores data for the CMDB server, as Defendant contends. Defendant argues that this database has no ability to manage a computing environment as Plaintiff's construction requires. The Court finds that the claims require storing a first model in "in a configuration management database (CMDB)," and storing a second model "in a license database." Thus, at a minimum, the Court agrees that the claim language indicates that the "CMDB" includes a database for storing a first model. However, the Court disagrees that the specification indicates that the term "CMDB" is limited to a simple database. For example, the specification states that "FIG. 1 shows, in block diagram form, an example of a collection of computer systems of an enterprise that are *managed by a CMDB* according to one embodiment." '093 Patent at 2:45–48 (emphasis added). Furthermore, the specification states that "*a CMDB* may comprise a plurality of computer systems that together provide the services of the CMDB server 110." *Id.* at 3:35–37 (emphasis added).

The specification also states that "[o]ne kind of CI [Configuration Item] that may be managed in a CMDB is a software asset." *Id.* at 1:43–44. To provide context for this statement, the specification states that "[c]onventional CMDBs, however, do not provide adequate capability for that an enterprise is in compliance with the terms of its software license contracts." *Id.* at 2:6–8. Thus, the Court finds that the claim language indicates that the recited "CMDB" includes a database, but is not limited to only a database. Indeed, the specification states that "[d]ata for the CMDB server 110 is illustrated as stored in a CMDB datastore 260 and a license datastore 270." *Id.* at 4:4–5. Thus, the specification explicitly refers to the database aspect of the CMDB as a "CMDB datastore."

The specification further states that "[b]ecause the license engine 250 is integrated with the CMDB server 110 and CMDB datastore 260, the various embodiments may allow for immediate and automatic feedback on the effect on software compliance of changes to the infrastructure modeled by the CMDB, in addition to on-demand runs of the license engine 250." *Id.* at 13:25–30. This indicates that a "CMDB" may include a "CMDB" datastore and a "CMDB" server. Indeed, the specification states that "FIG. 2 shows, in block diagram form, a *CMDB system* according to one embodiment." *Id.* at 2:48–49 (emphasis added). Thus, to the extent that Defendant argues that the term "CMDB" is limited to strictly a database, the Court rejects this argument. However, as indicated above, a person of ordinary skill in the art would understand that the claims require the recited "CMDB" to include at least a database for "storing a first model."

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the term **"license certificate"** to mean **"an indication of the right to deploy software."**

2. "exception indication"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"exception indication"	"plain meaning. Alternatively: an indication of an exception"	"information that indicates a condition or warning"

a) The Parties' Position

The parties dispute whether the term "an exception indication" requires construction. Plaintiff's contends the term should be given its plain meaning. (Dkt. No. 99 at 23.) Alternatively, Plaintiff argues that the term should be construed to mean "an indication of an exception." (*Id.*) Plaintiff further argues that Defendant's construction may be overly broad and overly narrow in different respects. (*Id.*) Plaintiff contends that Defendant's construction appears overly narrow because it paraphrases different portions of the specification that begin with the description "in one embodiment," where the specification is not limited to these embodiments. (*Id.*) (citing '093 Patent at 10:54–57, 10:56–64, 10:64–11:7, 11:9–20, 11:21–25). Plaintiff further argues that Defendant's construction appears overly broad because it expands the scope of the term to encompass any "condition" or "warning," when the patent teaches that exceptions are used to indicate "non-compliance conditions." (Dkt. No. 99 at 24.)

Defendant responds that the key dispute is the meaning of the word "exception." (Dkt. No. 106 at 28.) Defendant argues that this term should be construed because its ordinary meaning to persons of ordinary skill in the art differs from the way the term is used in the specification. (*Id.*) According to Defendant, skilled artisans use “exception” to refer to a problem or error in the operation of a program or system. (*Id.*) Defendant contends that the patent gives “exception” a broader meaning that includes a “warning” before non-compliance exists. (*Id.*) (citing ’093 Patent at 10:52–63). Defendant argues that this term should therefore be construed as “information that indicates a condition or warning.” Defendant further argues that Plaintiff’s alternative construction simply rearranges the words of “exception indication.” (Dkt. No. 106 at 28.)

For the following reasons, the Court finds that the term **“exception indication”** should be construed to mean **“indication of a non-compliance condition or an unresolved connection.”**

b) Analysis

The term “an exception indication” appears in asserted claim 1 of the ’093 Patent. The Court finds that the claim language indicates that the recited “exception indication” is an “indication of a non-compliance condition or an unresolved connection.” Specifically, claim 1 recites that an “exception indication” is generated “if the act of comparing the first model and the second model indicates non-compliance with the software license contract.” Similarly, dependent claim 7 recites that an “exception indication” is generated “if the first model cannot be connected to the second model.” Dependent claim 8 recites that an “exception indication” is generated “if the first model can be connected to a plurality of models in the license database.” Thus, the claims indicate that the recited “exception indication” is an “indication of a non-compliance condition or unresolved connection.”

Consistent with the claims, the specification states that “[i]n block 550, compliance rules may be evaluated to determine whether each of the software CIs complies with the terms of the software contract. In block 560, if any CI is not in compliance, then any desired exception processing may be performed." '093 Patent at 10:49–53. The specification further states that "[t]he exception may indicate a non-compliance condition." *Id.* at 10:56–57. Consistent with dependent claims 7 and 8, the specification describes a scenario where the CI "cannot be connected to a software contract through a license certificate. The license engine 250 may flag this as a connection exception (in block 540 of FIG. 5), requesting intervention by a contract or asset manager to resolve the connection exception." *Id.* at 12:41–45. Accordingly, the Court finds that the intrinsic evidence indicates that the recited "exception indication" is an "indication of a non-compliance condition or an unresolved connection."

Turning to the parties' construction, the Court finds that Plaintiff's construction simply rearranges the words of "exception indication," and thus, would be unhelpful to a jury. Accordingly, the Court does not adopt Plaintiff's construction. Defendant argues that the patent gives "exception" a broader meaning that includes a "warning" before non-compliance exists. (Dkt. No. 106 at 28.) The specification states that "[e]xception processing as performed in blocks 540 and 570 in one embodiment may be simply producing an error message or report indicating the exception." *Id.* at 10:54–56. The Court finds that providing "indication of a non-compliance condition" includes providing a warning of a non-compliance condition whether active, or before it exist. Thus, the Court finds that Defendant's proposal of "warning" is captured by the Court's construction.

The Court does not adopt the remaining portion of Defendant's construction because it introduces ambiguity into the claims. The claims indicate that the "exception indication" is an "indication of a non-compliance condition or an unresolved connection." Defendant's proposed "condition" is broader than a "non-compliance condition" and could expand the scope of the term to encompass any "condition," as Plaintiff argues. (Dkt. No. 99 at 24.) As discussed above, the intrinsic evidence indicates that an "exception indication" provides an "indication of a non-compliance condition or an unresolved connection." Accordingly, the Court does not adopt Defendant's construction.

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the term **"an exception indication"** to mean **"indication of a non-compliance condition or an unresolved connection."**

F. The '073 Patent

The parties' dispute focuses on the meaning and scope of three terms/phrases in the '073 Patent.

1. "wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"display the health status of the IT component by showing a first indicator for the IT component and a second indicator for the at least one IT subcomponent"	This phrase should be construed according to the meaning of the terms therein (see "IT subcomponent") and otherwise according to plain meaning	"show a first indicator for the IT component, and a second indicator for the at least one IT subcomponent, to display the health status of the IT component. The indicators are displayed via a single act of rendering, displaying both component and subcomponent health status indicators without requiring the user to perform any affirmative action, so that the first and second indicator are each separately visible at the same time on a single display window of a display unit"

a) The Parties' Position

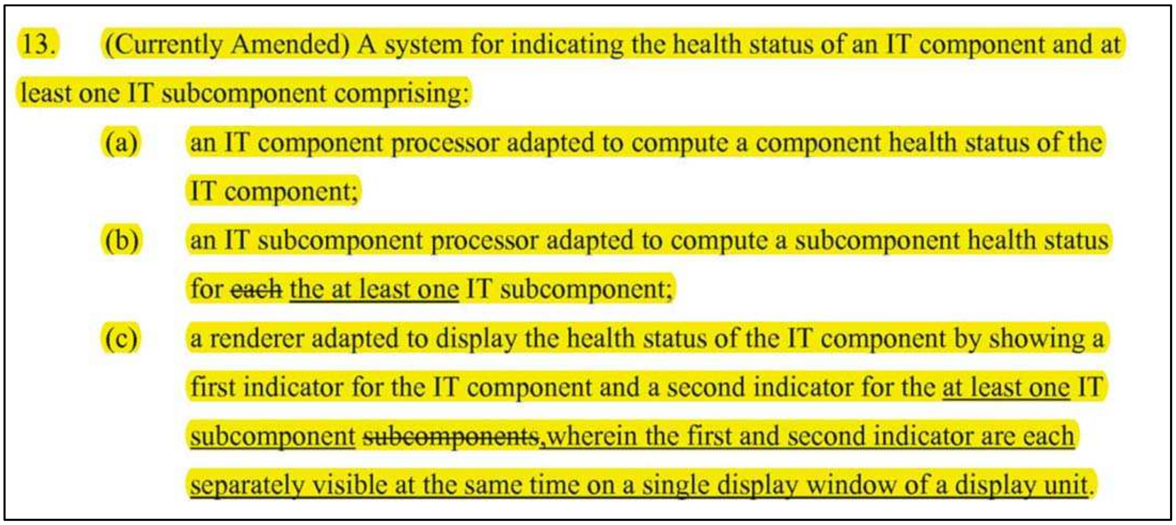
The parties dispute whether the phrase "display the health status of the IT component by showing a first indicator for the IT component and a second indicator for the at least one IT subcomponent” requires construction. Plaintiff contends that Defendant’s construction must be rejected as unintelligible. (Dkt. 99 at 20–21.) Defendant responds that during prosecution of the ’073 Patent, the patentees made clear statements in order to overcome the prior art. (Dkt. No. 106 at 28) (citing Dkt. No. 106-17 at 12). Defendant argues that the patentee must be held to their representations. (Dkt. No. 106 at 29.) Defendant further argues that a skilled artisan would have relied on these statements in ascertaining the claim scope. (*Id.*)

Plaintiff replies that there is no clear and unambiguous statement of narrowing scope. (Dkt. No. 108 at 10.) Plaintiff contends that Defendant misreads the prosecution history. (*Id.*) Plaintiff argues that the applicant explained that a difference with the prior art reference was that “[the prior art] explicitly teaches that the user must navigate through plural windows.” (*Id.*) According to Plaintiff, there is no clear and unambiguous disavowal. (*Id.*)

For the following reasons, the Court finds that the phrase **“wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit”** should be construed to mean **“wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit without requiring the user to perform any affirmative action (i.e., ‘navigate’).”**

b) Analysis

As an initial matter, the parties presented the phrase “display the health status of the IT component by showing a first indicator for the IT component and a second indicator for the at least one IT subcomponent” for construction. The Court finds that the parties’ dispute is properly resolved by construing the phrase “wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit,” which appears in claim 1 of the ’073 Patent. The Court further finds that the claims language is unambiguous and easily understandable by a jury. Thus, the only issue before the Court is whether the intrinsic evidence indicates that the patentees further limited claim 1. The Court finds that the patentees did further limit claim 1 by amending claim 13 as follows:



13. (Currently Amended) A system for indicating the health status of an IT component and at least one IT subcomponent comprising:

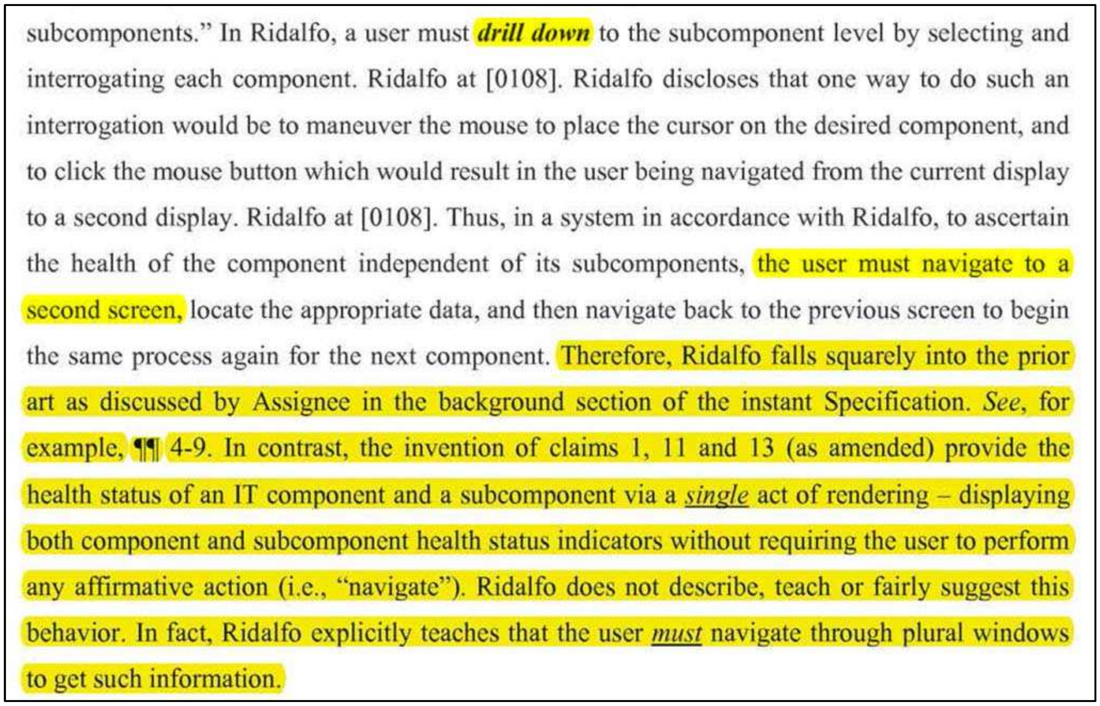
(a) an IT component processor adapted to compute a component health status of the IT component;

(b) an IT subcomponent processor adapted to compute a subcomponent health status for ~~each~~ the at least one IT subcomponent;

(c) a renderer adapted to display the health status of the IT component by showing a first indicator for the IT component and a second indicator for the at least one IT subcomponent ~~subcomponents,~~wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit.

(Dkt. 106-16 at 7 (1/12/09 Office Action Response)) (highlighting added).[12] As indicated, the patentees amended claim 13 to include "wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit." Regarding this amendment and the prior art, the patentees argued the following:

[12] Pending claim 13 referenced in the passage above ultimately issued as claim 1.



subcomponents." In Ridalfo, a user must ***drill down*** to the subcomponent level by selecting and interrogating each component. Ridalfo at [0108]. Ridalfo discloses that one way to do such an interrogation would be to maneuver the mouse to place the cursor on the desired component, and to click the mouse button which would result in the user being navigated from the current display to a second display. Ridalfo at [0108]. Thus, in a system in accordance with Ridalfo, to ascertain the health of the component independent of its subcomponents, the user must navigate to a second screen, locate the appropriate data, and then navigate back to the previous screen to begin the same process again for the next component. Therefore, Ridalfo falls squarely into the prior art as discussed by Assignee in the background section of the instant Specification. *See*, for example, ¶¶ 4-9. In contrast, the invention of claims 1, 11 and 13 (as amended) provide the health status of an IT component and a subcomponent via a *single* act of rendering – displaying both component and subcomponent health status indicators without requiring the user to perform any affirmative action (i.e., "navigate"). Ridalfo does not describe, teach or fairly suggest this behavior. In fact, Ridalfo explicitly teaches that the user *must* navigate through plural windows to get such information.

(Dkt. 106-16 at 12 (1/12/09 Office Action Response)) (highlighting added). As indicated, the patentees distinguished the claims form the prior art by arguing that the prior art required a user to "drill down to the subcomponent level" and "navigate to a second screen." *Id.* The patentees contrasted this with the amended claims that "provide the health status of an IT component and a subcomponent via a *single* act of rendering – displaying both component and subcomponent health status indicators without requiring the user to perform an affirmative action i.e. 'navigate'." *Id.* (emphasis in original). Accordingly, a person of ordinary skill in the art would understand that the phrase "wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit" means that the indicators are visible "without requiring the user to perform any affirmative action (i.e., 'navigate')."

This is further confirmed by the specification that states that in the prior art "[t]he users can navigate through the representation of the systems by expanding parts of the tree or by selecting the icons representing the component they want to explore further." '073 Patent at

1:46–49. The specification further states that "[t]he invention remedies the disadvantages of using a single color code or indicator for providing feedback on the health/status or components in a complex Enterprise System." *Id.* at 2:64–67. Thus, given the intrinsic evidence, the Court finds that the patentee clearly and unambiguously limited the phrase "wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit" to "without requiring the user to perform any affirmative action (i.e., 'navigate')."

Turning to the parties' arguments, Plaintiff contends that there is no clear and unambiguous disavowal. Plaintiff argues that the patentees explained that a difference with the Ridalfo reference was that "Ridalfo explicitly teaches that the user must navigate through plural windows." (Dkt. No. 108 at 10). Plaintiff's argument fails to address the patentees' statement of what amended claim 13 actually included, which was "displaying both component and subcomponent health status indicators without requiring the user to perform an affirmative action i.e. 'navigate'." Accordingly, the Court disagrees with Plaintiff's conclusory analysis.

Turning to Defendant's construction, the Court finds that it unnecessarily rewrites the clear and unambiguous claim language. As discussed, the Court finds that the patentees did further limit claim 1, and thus, includes the limitation at the end of the unambiguous phrase. Accordingly, the Court does not adopt either party's construction.

c) Court's Construction

In light of the evidence presented by the parties, the Court construes **"wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit"** should be construed to mean **"wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit without requiring the user to perform any affirmative action (i.e., 'navigate')."**

2. “subcomponent” and “IT subcomponent”

Disputed Term	Plaintiff’s Proposal	Defendant’s Proposal
“subcomponent”	plain meaning	Indefinite. Alternatively, “a part or portion of a component”
“IT subcomponent”	plain meaning	Indefinite. Alternatively, “a part or portion of an IT component”

a) The Parties’ Position

The parties dispute whether the terms “subcomponent” and “IT subcomponent” require construction. Plaintiff contends that the terms are not indefinite and should be given their plain meaning. (Dkt. No. 99 at 19.) Plaintiff argues that the “subcomponent” terms merely introduces a “sub” prefix, such that, consistent with plain meaning, a “subcomponent” is part of or has a dependency relationship with a “component.” (*Id.*) (citing ’073 Patent at 7:9–10, 7:16–21, 2:24–30). Plaintiff argues that the prosecution history is in full accord, describing “components” and “subcomponents” to have their “conventional” meaning, including dictionary definition. (Dkt. No. 99 at 20) (citing Dkt. No. 99-11 at 13 (6/5/08 Office Action Response)). According to Plaintiff, the intrinsic evidence indicates that the term “subcomponent” is not indefinite. Regarding Defendant’s alternate construction, Plaintiff contends that it appears to apply one restrictive definition and ignores the contemplation of a dependency relationship. (Dkt. No. 99 at 20.)

Defendant responds that the scope of “subcomponent” was rendered unclear when the patentee responded to an objection regarding the antecedent basis for the term “IT subcomponent.” (Dkt. No. 106 at 29) (citing Dkt. 106-18 at 14 (6/5/08 Office Action Response)). Defendant argues that the patentees use of three distinct definitions with several exemplary usages clouded the term “subcomponent” with uncertainty. (Dkt. No. 106 at 30.) According to Defendant, it is unclear which definition should apply to the claimed “subcomponent.” (*Id.*) Defendant argues that these multiple definitions render the claim indefinite because something that qualifies as a “subcomponent” under one definition may not qualify under either of the other two. (*Id.*) Defendant further contends that the patentees provided no guidance on which ones are applicable. (*Id.*)

Plaintiff replies that the dictionary definitions provided to the examiner show a consistent meaning for the prefix “sub.” (Dkt. No. 108 at 10.) Plaintiff argues that rather than causing confusion, the definition inform a skilled artisan of the scope of asserted claims. (*Id.*)

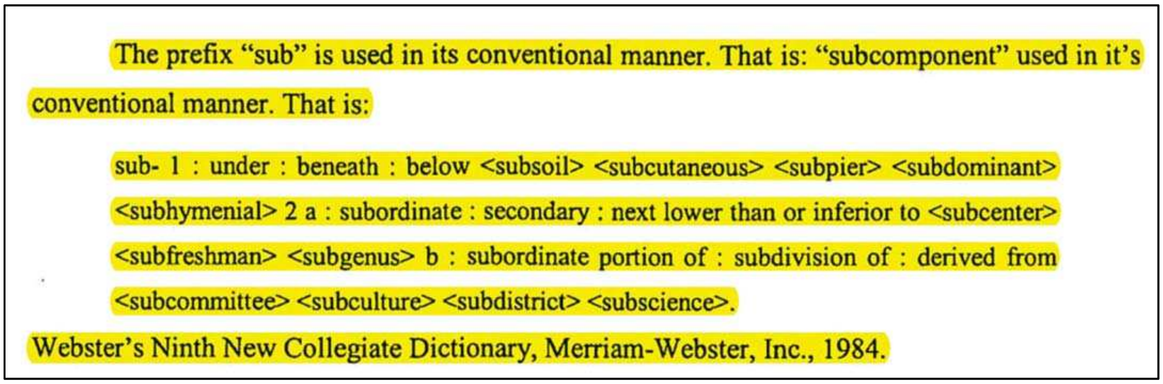
For the following reasons, the Court finds that the terms **“subcomponent”** and **“IT subcomponent”** are not indefinite, and should be given their **plain and ordinary meaning.**

b) Analysis

The term “subcomponent” and “IT subcomponent” appear in asserted claims 1 and 4 of the ’073 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The Court further finds that the claim language “viewed in light of the specification . . ., inform[s] those skilled in the art about the scope of the invention with reasonable certainty.” *Nautilus*, 134 S. Ct. at 2129. For example, the specification states that “components can be computers, computer peripherals, computer programs, networking equipment, and manufacturing equipment. Components can also be virtual components like business processes that can be combined into a business system.” ’073 Patent at 1:27–31. The specification further states that “[t]he different components relate to each other in different ways. Some components are being parts of others, while some components are using the service provided by other components in some way.” *Id.* at 1:32–36. Thus, the Court finds that the intrinsic evidence provides a number of examples of components and possible relationships between these components, one of which is a component-subcomponent relationship.

In further describing the relationship between components, the specification states “[u]sing this technique, the health/status of a component can be completely independent from the health/status of its subcomponents or components depending from it.” *Id.* at 3:39–41. The specification further describes computing a “‘sub-severity’ based on the health/status of the subcomponents or the components that the component depends upon.” *Id.* at 3:52–54. The specification further provides one example where “both the indicator in the foreground and the indicator in the background are red, and thus the user may assess that the component ca_os@FO@biz is not healthy and at least one component depending from the component ca_os@FO@biz is unhealthy.” *Id.* at 4:27–32. Thus, a person of ordinary skill in the art would understand, with reasonable certainty, the scope of the terms “subcomponent” and “IT subcomponent.”

The prosecution history further informs, with reasonable certainty, the scope of the terms “subcomponent” and “IT subcomponent.” Specifically, the patentees provides the following dictionary definition to the examiner:



The prefix “sub” is used in its conventional manner. That is: “subcomponent” used in it’s conventional manner. That is:

sub- 1 : under : beneath : below <subsoil> <subcutaneous> <subpier> <subdominant> <subhymenial> 2 a : subordinate : secondary : next lower than or inferior to <subcenter> <subfreshman> <subgenus> b : subordinate portion of : subdivision of : derived from <subcommittee> <subculture> <subdistrict> <subscience>.

Webster’s Ninth New Collegiate Dictionary, Merriam-Webster, Inc., 1984.

(Dkt. 106-18 at 14 (6/5/08 Office Action Response)). Contrary to Defendant’s contention, this dictionary definition is consistent with the specification’s description of subcomponents as being a subpart of a component or depending from another component. Nothing in the intrinsic records limits the recited “subcomponent” to just one subpart of the cited dictionary definition.

Thus, the Court disagrees with Defendant's argument that the patentee provided no guidance on which one of the three definitions are applicable. (Dkt. No. 106 at 30.) Instead, the Court finds that the dictionary definition provided to the examiner show a consistent meaning for the prefix "sub" and informs a person of ordinary skill in the art about the scope of the asserted claims.

Regarding Defendant's alternate construction, the Court finds that it is inconsistent with the intrinsic evidence. Specifically, Defendant's construction appears to limit the term "subcomponents" to only those components that are a part of a component and would exclude "subcomponents" that depend on another component. As discussed above, the intrinsic evidence indicates that the recited "subcomponent" includes components that are parts of other components, as well as components depending on other components. '073 Patent at 1:32–36.

c) Court's Construction

In light of the evidence presented by the parties, the Court finds that the terms **"subcomponent"** and **"IT subcomponent"** are not indefinite, and will be given their **plain and ordinary meaning.**

3. "IT component processor," "IT subcomponent processor," and "processor"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"IT component processor"	"a processor that computes the health status of an IT component"	plain meaning
"IT subcomponent processor"	a processor that computes the health status of an IT subcomponent	plain meaning
"processor"	Plain meaning.	plain meaning, which is "hardware and/or software for processing"

a) The Parties' Position

The parties dispute whether the terms "IT component processor," "IT subcomponent processor," and "processor" require construction. Plaintiff contends that its construction comes from the intrinsic record. (Dkt. No. 99 at 21.) Plaintiff argues that the claim language expressly provides that "an IT component processor" is "adapted to compute a component health status of the IT component," and that "an IT subcomponent processor" is "adapted to compute a subcomponent health status for at least one IT subcomponent." (*Id.*) (citing '073 Patent at 7:11–15). Plaintiff argues that Defendant's construction seeks to give an unnecessary construction to the plain meaning term "processor," and then fails to provide a construction for the specialized term "IT component processor." (Dkt. No. 99 at 21.) Plaintiff argues that Defendant's approach is exactly backwards to a correct claim construction approach based on the intrinsic evidence. (*Id.*)

Defendant responds that Plaintiff does not dispute that "processor" can be left to plain meaning, which is hardware and/or software for processing. (Dkt. No. 106 at 31.) Defendant further argues that "IT component processor" and "IT subcomponent processor" can also be left to plain meaning because Plaintiff does not show any need to construe them. (*Id.*)

For the following reasons, the Court finds that the terms **"IT component processor"** and **"IT subcomponent processor"** should be given their **plain and ordinary meaning.**

b) Analysis

As an initial matter, the Court finds that the term "processor" is not recited by itself in the claims. Instead, the term "processor" is preceded by either "IT component" or "IT subcomponent." Thus, the Court finds that the term "processor" should be considered with the terms "IT component processor" and "IT subcomponent processor." The term "IT component processor" appears in claims 1, 3, 5, and 7-9 of the '073 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The term “IT subcomponent processor” appears in claims 1, 4, 5, and 7-9 of the ’073 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim.

The Court further finds that the claim language itself defines the terms “IT component processor” and “IT subcomponent processor.” Specifically, claim 1 recites that the “IT component processor” is “adapted to compute a component health status of the IT component.” Likewise, claim 1 recites that the “IT subcomponent processor” is “adapted to compute a subcomponent health status for the at least one IT subcomponent.” Accordingly, the Court finds that in light of the claim language, the terms “IT component processor” and “IT subcomponent processor” should be given their plan and ordinary meaning. Indeed, Plaintiff’s construction is redundant of the claim language and would not be helpful to a jury. Likewise, the recited “processor” is “adapted to compute a health status of a component/subcomponent. Accordingly, the Court does not adopt either party’s proposal.

c) Court’s Construction

In light of the evidence presented by the parties, the terms **“IT component processor”** and **“IT subcomponent processor”** will be given their **plain and ordinary meaning**.

G. The ’992 Patent

The parties’ dispute focuses on the meaning and scope of four terms/phrases in the ’992 Patent.

1. “importance” and “importance of the corresponding service”

Disputed Term	Plaintiff’s Proposal	Defendant’s Proposal
“importance”	plain meaning. Alternatively: the fact of being important, or the degree to which something is important	Indefinite

“importance of the corresponding service”	Construe “importance.” Otherwise, plain meaning.	Indefinite.

a) The Parties’ Position

The parties dispute whether the term “importance” and the phrase “importance of the corresponding service” are indefinite. Plaintiff contends that the specification refers to importance as a term of degree, which can vary from component to component. (Dkt. No. 99 at 27) (citing ’992 Patent at 2:6–8, 7:25–26). Plaintiff contends that the patentee made this point clear by referring to dictionary definitions during prosecution of the underlying patent application. (*Id.*) (citing Dkt. No. 99-12 at 6 (7/29/13 Office Action Response)). Plaintiff also argues that the patentee also indicated during prosecution that “importance” may be based on a ranking of the underlying components. (Dkt. No. 99 at 27) (citing Dkt. No. 99-12 at 7 (7/29/13 Office Action Response)).

Defendant responds that the patentee provided highly subjective definitions of “importance” to overcome prior art rejections during prosecution. (Dkt. No. 106 at 31) (citing Dkt. No. 106-20 at 7 (7/29/13 Office Action Response)). Defendant argues that Plaintiff is bound by these definitions. (Dkt. No. 106 at 31.) Defendant further argues that the term “importance” is inherently subjective. (*Id.* at 32.) According to Defendant, whether a particular service is “important,” i.e. “has a major effect on someone or something,” will vary from one person to another. (*Id.*) Defendant contends that the claim is indefinite because “importance” lacks any objective definition to a person of ordinary skill in the art. (*Id.*) Defendant further argues that Plaintiff’s alternative construction must be rejected because it adopts only a portion of the definition of “importance” provided during prosecution, and omits the incorporated definition of “important” the patentee provided to the Examiner. (*Id.*)

Plaintiff responds that Defendant focuses only on the notion that “importance” is a “term of degree," and ignores the surrounding claim language. (Dkt. No. 108 at 12.) Plaintiff argues that the claim language requires a "metric" for each "attribute," which includes "importance of the corresponding service" and can vary from component to component. (*Id.*) (citing '992 Patent at 2:6–8). According to Plaintiff, a skilled artisan would appreciate that the intrinsic evidence dictionary definition supports that a node may have higher "importance" than another node. (Dkt. No. 108 at 12.)

For the following reasons, the Court finds that the term **"importance"** is not indefinite, and should be given its **plain and ordinary meaning.** The Court further finds that the phrase **"importance of the corresponding service"** does not require construction.

b) Analysis

The term "importance" appears in asserted claims 1 and 8 of the '992 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The claim language itself recites that the "size of the spotlight varies based on the importance of the corresponding service." '992 Patent at Claim 1. Claim 1 further recites that this is based on a "determined metric." Thus, the "importance" of the corresponding service is indicated by size. In other words, one node may have a higher "importance" than another node. *See, e.g.*, '992 Patent at 4:4–19.

Indeed, the specification states that "the mental workload of the viewer may be greatly reduced" because "[t]he spotlights make establishing a mental ranking of the importance of each object very easy to do, simply by looking at the relative size, color, and brightness of the spotlights behind each monitored object." '992 Patent at 3:52–56. Consistent with this statement, the patentee argued during prosecution that "importance" is based on a ranking. (Dkt. No. 106-20 at 9 (7/29/13 Office Action Response)) ("By contrast importance, as disclosed by [prior art], relates to the importance of an SLA violation based on a ranking of SLA violations."). The patentee also quoted a dictionary definition during prosecution that defined "importance" as "the fact of being important, or the degree to which something is important." *Id.* at 7. Thus, a person of ordinary skill would understand that the term "importance" relates to relative ranking or the degree to which something is important. '992 Patent at 2:6–8 ("Users frequently need to try to determine the *relative importance* of each 'in trouble' object, so that they can prioritize their work.") (emphasis added). Accordingly, the Court finds that the claim language, "viewed in light of the specification . . ., inform[s] those skilled in the art about the scope of the invention with reasonable certainty." *Nautilus*, 134 S. Ct. at 2129.

Defendant contends that the definition of "importance" provided in the prosecution history incorporated a definition of "important." Specifically, the dictionary defined "important" as "something that is important has a major effect on someone or something, for example because it affects someone's life or the way a situation develops." (Dkt. No. 106-20 at 7 (7/29/13 Office Action Response)). According to Defendant, this makes the term "importance" inherently subjective because whether a particular service is "important," i.e. "has a major effect on someone or something," will vary from one person to another. (Dkt. No. 106 at 32.) The Court is not persuaded by Defendant's analysis.

As an initial matter, the Court finds that the definition of "important" provided in the prosecution history is not entirely applicable to the recited claim language. The claims recite the "importance of the corresponding service," which is a different context than the definition of "important: as affecting "someone's life." Moreover, Defendant's analysis ignores the surrounding claim language. As discussed above, the claim language requires a "metric" for each "attribute," which includes "importance of the corresponding service," which can vary from component to component. ’992 Patent at 2:6–8 (“Users frequently need to try to determine the *relative importance* of each ‘in trouble’ object, so that they can prioritize their work.”) (emphasis added). Likewise, in clarifying the definition of “importance,” the patentee argued that “importance” is based on ranking. (Dkt. No. 106-20 at 9 (7/29/13 Office Action Response)) (“By contrast importance, as disclosed by [prior art], relates to the importance of an SLA violation based on a ranking of SLA violations.”). In other words, “importance” can be the relative ranking based on an actor’s values. Accordingly, the Court finds that Defendant failed to show by clear and convincing evidence that the term “importance” is indefinite. Finally, having construed the term “importance,” the Court finds that the phrase “importance of the corresponding service” does not require construction.

c) Court’s Construction

In light of the evidence presented by the parties, the Court finds that the term **“importance”** is not indefinite, and will be given its **plain and ordinary meaning.** The Court further finds that the phrase **“importance of the corresponding service”** should be given its **plain and ordinary meaning**.

2. “service level agreement (SLA)” and “SLA violation”

Disputed Term	Plaintiff’s Proposal	Defendant’s Proposal
“service level agreement (SLA)”	one or more established IT service level commitments by an IT service provider, which can be violated resulting in an SLA violation	“an agreement between two parties defining one or more service level objectives
“SLA violation”	plain meaning. Alternatively: “violation of an SLA”	“failure to meet an objective defined in an SLA”

a) The Parties’ Position

The parties dispute whether a “service level agreement (SLA)” is an agreement between two parties. Plaintiff contends that its construction adheres to how “SLA” is used in the claims and throughout the specification. Plaintiff argues that nothing in the intrinsic record requires legal contractual interpretation to determine an SLA. (Dkt. No. 99 at 28.) Plaintiff contends that ITIL provides a definition for SLA that follows its construction and does not require a contract. (*Id.*)

Defendant responds that the plain meaning of "service level agreement (SLA)" is an agreement between two parties defining one or more service level objectives. (Dkt. No. 106 at 32.) Defendant argues that multiple references in the field consistently describe a service level agreement (or "SLA") in this manner. (*Id.*) Defendant further argues that even Plaintiff's extrinsic evidence supports Defendant's construction. (*Id.*) Defendant also argues that because an SLA defines one or more service level objectives, an "SLA violation" is simply the failure to meet one of those objectives. (*Id.* at 33.) Defendant contends that Plaintiff improperly tries to strike "agreement" from the claim by redefining "service level agreement" term to cover unilateral "commitments." (*Id.*)

Plaintiff replies that a skilled artisan would not understand an SLA to include details of a *quid pro quo* between two parties. (Dkt. 108 at 12.) According to Plaintiff, an SLA refers to the IT service provider's IT service level commitments. (*Id.*) Plaintiff also argues that Defendant's own evidence suggests an "internal" SLA. (*Id.*)

For the following reasons, the Court finds that the term **"service level agreement (SLA)"** should be construed to mean **"an agreement between a customer and a service provider, which may be an internal IT department or external organization."** The Court further finds that the term **"SLA violation"** should be given its **plain and ordinary meaning**.

b) Analysis

The term "service level agreement (SLA)" appears in asserted claims 1 and 8 of the '992

Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The term "SLA violation" appears in asserted claims 1 and 8 of the '992 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The Court further finds that the '992 Patent only refers to a "service level agreement" once, but refers to an "SLA violation" multiple times. Specifically, the specification states "FIG. 1 is a graph 100 illustrating a service model displayed according to the prior art. In this example, for [sic] icon groups are defined: severity status 110, Service Level Agreement (SLA) violation 120, importance 130, and events present 140." '992 Patent at 3:26–29. The specification further states that "[p]erformance monitoring . . . may involve monitoring a very large number of metrics, with the need to monitor over a million metrics in many enterprises." *Id.* at 1:66–2:1. Thus, the intrinsic evidence indicates that a SLA relates to service metrics. Indeed, claim 1 recites "determining a metric for each of a plurality of attributes associated with a service level agreement (SLA) for each of the plurality of services."

Turning to the extrinsic evidence submitted by the parties, the Court finds that a "SLA" is consistently defined as an "agreement" between a customer and a service provider. For example, the extrinsic definition submitted by Plaintiff states that a "service level agreement" is "an agreement between an IT service provider and a customer." (Dkt. No. 99-3 at 3 (ITIL glossary and abbreviations)). Thus, the Court disagrees with Plaintiff that a person of ordinary skill would understand an "SLA" to be a unilateral "commitment" by an IT service provider. Indeed, the plain language of "service level agreement" is that it is an "agreement."

However, the Court does agree that the extrinsic evidence indicates that the agreement can be between a customer and a service provider that may be either an internal IT department or external organization. For example, one extrinsic definition submitted by Defendant states that "[s]ervice level agreements (SLAs) are an agreement between the customer and the 'service provider,' which may be an internal IT department or an external organization offering IT services for a fee." (Dkt. 106-23 at 6 (SLM Solutions – A Buyer's Guide (2004)). Accordingly, the Court finds that a person of ordinary skill in the art would understand that a SLA is an agreement between a customer and a service provider, which may be an internal IT department or external organization.

Finally, the extrinsic evidence indicates that an SLA generally describes the service, documents service level metrics, and specifies the responsibilities of the service provider and the customer. *See, e.g.*, (Dkt. No. 99-3 at 3 (ITIL glossary and abbreviations)), (Dkt. 106-23 at 6 (SLM Solutions – A Buyer's Guide (2004))), (Dkt. No. 106-26 at 6 (Service Level Management Using IBM Tivoli Service Level Advisor and Tivoli Business Systems Manager (2004))). However, the Court finds that this aspect does need to be included in the construction of "SLA" because it is indicated by the words "Service Level." To the extent that a party contends that an SLA would not describe the service, document service level metrics, and/or specify the responsibilities of the service provider and the customer, the Court rejects this argument. Furthermore, a person of ordinary skill in the art would understand that a "SLA violation" means a violation of an SLA, which is its plain and ordinary meaning.

c) Court's Construction

In light of the evidence presented by the parties, the Court construes the term **"service level agreement (SLA)"** to mean **"an agreement between a customer and a service provider, which may be an internal IT department or external organization."** The term **"SLA violation"** will be given its **plain and ordinary meaning**.

3. "graph" and "node"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"graph"	plain meaning	"a visual representation of nodes and the connections between the nodes"
"node"	"a representation of a component or service in an IT service mode"	"a visual object connected to and having a relationship with at least one other node in the graph"

a) The Parties' Position

The parties dispute whether the term "graph" requires construction. The parties also dispute whether the term "node" should be construed as "a representation of a component or service in an IT service mode," as Plaintiff proposes, or as "a visual object connected to and having a relationship with at least one other node in the graph," as Defendant proposes. Plaintiff contends that "graph" carries its plain meaning. (Dkt. No. 99 at 24.) Plaintiff argues that '992 Patent refers to a general representation of nodes, not as "graphs," but as "directed acyclic graphs." (*Id.*) ('992 Patent at 1:36–41). Plaintiff argues that other references to "graph" carry its plain and ordinary meaning. (Dkt. No. 99 at 24.) Regarding the term "node," Plaintiff argues that its construction comes from the intrinsic evidence. (*Id.* at 25) (citing '992 Patent at 7:10–11). Plaintiff argues that Defendant's construction ignores the definitional aspect that a node represents components of an IT service model. (*Id.*)

Defendant responds that the key dispute revolves around the visual aspects of a "graph" and "node." (Dkt. No. 106 at 35.) Defendant argues that its construction is consistent with the understanding of persons of ordinary skill in the art. (*Id.*) Defendant contends that claim 1 recites that the graphs and nodes are displayed and that a graph includes "a plurality of nodes," which are visual objects. (*Id.*) (citing '992 Patent at Claim 1, 7:19–20). Defendant further argues that the specification also repeatedly describes a "graph" as being a visual representation that shows the interconnections between nodes. (Dkt. No. 106 at 35) (citing '992 Patent at Abstract, 1:26–60, 2:48–53, 3:26–59, Figures 1–3). Defendant further argues that the patent distinguishes these "graph" visualizations from the "block diagram" forms shown in Figures 4–6. (Dkt. No. 106 at 35) (citing '992 Patent 2:54–62, Figures 4–6). Defendant contends that Plaintiff's construction for "graph" is too broad as it would include "a mathematical representation." (Dkt. No. 106 at 35.)

Plaintiff replies that "graph" is well understood to be a mathematical representation, as supported by the intrinsic evidence and by its expert's declaration. (Dkt. No. 108 at 11.) Plaintiff argues that the claims require "graph" to be "generate[d]" first, and it is only visual upon "display[ing] the graph." (*Id.*) Plaintiff further argues that the claims require a "node" to "represent[ ] a service of a plurality of services" rather than generic "objects." (*Id.*)

For the following reasons, the Court finds that the term **"graph"** should be construed to mean **"a visual representation;"** and the term **"node"** should be construed to mean **"a representation of a component or service in a graph."**

b) Analysis

The term "graph" appears in asserted claims 1 and 8 of the '992 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The term "node" appears in asserted claims 1 and 8 of the '992 Patent. The Court finds that the term is used consistently in the claims and is intended to have the same meaning in each claim. The Court further finds that the intrinsic and extrinsic evidence indicates that a person of ordinary skill in the art would understand that the recited "graph" means "a visual representation." The claim language itself states that the recited graph is displayed. '992 Patent at Claim 1 ("displaying a graph on a display screen"), Claim 8 ("a second computer system . . . configured to display the graph").

Moreover, the specification states that the advantage of the disclosure is that it "greatly reduce[s]" the "mental workload of the *viewer*" by "establishing a mental ranking of the importance of each object . . . *simply by looking* at the relative size, color, and brightness of the spotlights behind each monitored object." '992 Patent at 3:51–56 (emphasis added). As the specification states, one issue with the prior art was that "as the amount of data shown by the graph increases . . . Users must scan many different icons in the graph and memorize what all the different icons mean." *Id.* at 2:2–7. Likewise, the specification states that "[f]or big service models, however, the *visualization aspect* may become challenging for the user" because "it[s] difficult to present the most important data given the *limited screen display area* that is available." *Id.* at 1:41–45 (emphasis added). Thus, the intrinsic evidence indicates that the recited "graph" is "a visual representation." The extrinsic evidence submitted by Defendant is consistent with the specification and states that "[g]raphs are those visual representations . . . ." (Dkt. 106-21 at 7 (Introduction to Information Visualization)). Accordingly, the Court finds that a person of ordinary skill in the art would understand "graph" to mean "a visual representation."

The Court does not adopt Plaintiff's construction because it is too broad and does not capture the visual aspect that the specification emphasizes as being important. The Court agrees that divorced from context of the intrinsic evidence, the plain meaning of graph could be a "mathematical representation." But claims are not construed in a vacuum and must be analyzed in the context of the intrinsic evidence. *Adams Respiratory Therapeutics, Inc. v. Perrigo Co.*, 616 F.3d 1283, 1290 (Fed. Cir. 2010) ("Claim terms are not construed in a vacuum divorced from the specification."). In other words, "a basic internet search" is not the proper starting point for determining the meaning of a disputed term. (Dkt. No. 108 at 11 n.3.) Accordingly, to the extent that Plaintiff argues that the recited "graph" is purely a "mathematical representation," the Court rejects this argument.

Turning to the term “node,” the Court finds that the intrinsic evidence indicates that it means “a representation of a component or service in a graph.” The claims recite that the graph includes nodes that represent services. ’992 Patent at Claim 1 (“the graph including a plurality of nodes, each of the plurality of nodes representing a service of a plurality of services”), Claim 8 (“a graph including a plurality of nodes, each node of the plurality of nodes modeling a service of a plurality of services”). In describing the relationship between nodes and components, the specification states “where each component of the service model, whether a business user, a service component, or an IT infrastructure component, is represented as a node in the graph.” *Id.* at 1:38–41. The extrinsic evidence submitted by Defendant indicates that graphs include nodes, which “represent instances of the data.” (Dkt. 106-21 at 7 (Introduction to Information Visualization)). Accordingly, the Court finds that a person of ordinary skill in the art would understand “node” to mean “a representation of a component or service in a graph.”

The Court does not adopt Defendant’s construction for “node” because it fails to indicate that a node represents a component or service. Instead, Defendant’s construction defines a node as a visual object. The Court’s construction for “graph” captures the visual aspect that the specification emphasizes as being important. Thus, defining a “node” as a “visual object” is arguably broader than the claims require. As discussed above, the intrinsic evidence indicates that the recited “node” is “a representation of a component or service in a graph.” Accordingly, the Court does not adopt Defendant’s construction for “nodes.”

c) Court’s Construction

In light of the evidence presented by the parties, the Court construes the term **“graph”** to mean **“a visual representation;”** and the term **“node”** to mean **“a representation of a component or service in a graph.”**

4. "variable graphical image," "a variable graphical image positioned with the node," "spotlight," and "displaying a spotlight with each of the nodes of the plurality of nodes"

Disputed Term	Plaintiff's Proposal	Defendant's Proposal
"variable graphical image	"a graphical image having multiple attributes that can be varied to indicate relative status of a node in an IT service model"	plain meaning. Alternatively: "a graphical image that can be changed"
"a variable graphical image positioned with the node"	construe "variable graphical image" and "node." Otherwise, plain meaning.	Indefinite. Alternatively: "an image that can be changed and is displayed in addition to the displayed node"
"spotlight"	"a graphical image having multiple characteristics that can be varied to indicate relative status of a node in an IT service model"	"an image or graphic displayed in addition to a node"
"displaying a spotlight with each of the nodes of the plurality of nodes"	construe "spotlight" and "node." Otherwise, plain meaning	Indefinite. Alternatively: "displaying an image or graphic in addition to each displayed node"

a) The Parties' Position

The parties essentially dispute whether the terms "variable graphical image" and "spotlight" indicate relative status of a node in an IT service model. The parties also dispute whether the phrase "a variable graphical image positioned with the node" and the phrase "displaying a spotlight with each of the nodes of the plurality of nodes" are indefinite because the patent fails to indicate the required spatial or visual relationship. Regarding the term "variable graphical image," Plaintiff contends that its construction draws directly from the intrinsic evidence for this term chosen by the patentee. (Dkt. No. 99 at 24) (citing '992 Patent at 8:14–18, 2:37–39). Plaintiff argues that Defendant's construction does not adhere to the claim language and fails to account for the significance of attributes indicating the status of nodes. (Dkt. No. 99 at 25.)

Regarding the term “spotlight,” Plaintiff argues that the term is not a term in the art, and is the patentee’s own term for indicating the relative status of a node in an IT service model. (Dkt. No. 99 at 26) (citing ’992 Patent at 7:19–22, 5:49–51, 2:25–27). Plaintiff contends that Defendant’s construction ignores the claim language and embodiments that link a “spotlight” with a “node” in an IT service model, where the “spotlight” conveys “characteristics” about the corresponding “node.” (Dkt. No. 99 at 26.)

Regarding the phrases that Defendant contends are indefinite, Plaintiff argues that various figures illustrate “variable graphical image[s] positioned with the node” or exemplar spotlights displayed with nodes. (Dkt. No. 99 at 25–26) (citing ’992 Patent at Figures 1-6). Plaintiff contends that Defendant’s argument that the term is indefinite apparently results from Defendant being unable to piece its own constructions together. (Dkt. No. 99 at 25.)

Defendant responds that “variable graphical image” should be given its plain meaning, and in the alternative, “a graphical image that can be changed.” (Dkt. No. 106 at 33.) Defendant argues that a plain meaning construction is appropriate because the patent does not suggest any intention to depart from that plain meaning. (*Id.*) (citing ’992 Patent at 5:49–52). Defendant contends that Plaintiff’s construction rearranges surrounding language of the claim and improperly imports an “IT service model.” (Dkt. No. 106 at 33.) According to Defendant, nothing in the intrinsic record limits the meaning of “variable graphical image” to that narrow context. (*Id.* at 33) (citing ’992 Patent at 5:62–67).

Regarding the term “spotlight,” Defendant argues that the term is analogous to the “variable graphical image” in claim 8. (Dkt. No. 106 at 34.) Defendant contends that the “spotlight” and “variable graphical image” terms share substantially the same claim construction disputes. (*Id.*) Defendant argues that the proper construction of “spotlight” to a skilled artisan is "an image or graphic displayed in addition to a node." (*Id.*) Defendant further argues that Plaintiff's construction of "spotlight" should be rejected for the same reasons as Plaintiff's construction of "variable graphical image." (*Id.*)

Regarding the phrase that Defendant contends are indefinite, Defendant argues that the limitations require contemporaneous display of the variable graphical image/spotlight and the node. (Dkt. No. 106 at 33.) According to Defendant, the patent does not explain the required spatial or visual relationship between them. (*Id.* at 33–34) (citing '992 Patent at 5:56–61). Defendant argues that a person of ordinary skill in the art could not determine with reasonable certainty or confidence whether a particular display showing the variable graphical image and the node falls within the "positioned with" limitation. (Dkt. No. 106 at 33-–34.)

Plaintiff replies that Defendant argues plain meaning for the term "variable graphical image" and "spotlight," but ignores that the claims speak in terms of a "variable graphical image" having "attributes" and of a "spotlight" having "characteristics." (Dkt. No. 108 at 11–12.) Regarding the phrases that Defendant contends are indefinite, Plaintiff argues that the patent figures inform skilled artisans of "variable graphical image[s] positioned with the node," and that the phrases are not indefinite. (*Id.*)

For the following reasons, the Court finds that the term **"variable graphical image"** should be given its **plain and ordinary meaning**; and the term **"spotlight"** should be construed to mean **"a graphical image."** The Court further finds that the phrases **"a variable graphical image positioned with the node"** and **"displaying a spotlight with each of the nodes of the plurality of nodes"** are not indefinite, and should be given their **plain and ordinary meaning.**

b) Analysis

The term "variable graphical image" appears in asserted claim 8 of the '992 Patent. The term “spotlight” appears in asserted claim 1 of the ’992 Patent. The Court finds that the term “variable graphical image” is unambiguous, is easily understandable by a jury, and should be given its plain and ordinary meaning. The word “variable” and the claim language indicates that it is “a graphical image that can vary in one or more dimensions (for example color, size, and brightness). Specifically, claim 8 recites “the graphical image having a plurality of attributes, . . . each of the attributes being varied based on the determined metric for each associated state such that, a size of the variable graphical image varies based on the importance of the corresponding service, and a color of the variable graphical image varies based on the severity of the incident causing the SLA violation.” ’992 Patent at Claim 8. Accordingly, a person of ordinary skill in the art would understand the plain and ordinary meaning of the term “variable graphical image” when considered in the context of the claims.

Regarding the term “spotlight,” the intrinsic evidence indicates that it is also “a graphical image.” Claim 1 recites that the “displayed spotlight” is “graphically varied based on the determined metric such that, a size of the spotlight varies based on the importance of the corresponding service.” ’992 Patent at Claim 1. The specification states that in one embodiment the “spotlight may vary in three dimensions: color, size, and brightness.” *Id.* at 3:41–42. The specification also states that “[v]arious embodiments of the present invention replace some or all of the metric indicator icons with a single colored spotlight that appears behind the object, reducing the mental workload of determining the relative importance of multiple objects in the graph view.” *Id.* at 3:21–25. The specification further states that “[t]he spotlights may be implemented using any graphical technique known to the art for placing a graphical image over or below another image on a screen.” *Id.* at 5:49–51. Thus, the specification indicates that the recited “spotlight” is “a graphical image.”

Turning to the parties' construction for the terms "variable graphical image" and "spotlight," the Court does not adopt Plaintiff's construction because it is redundant of the claim language and imports an "IT service model." The specification explicitly states that the claims are not limited to service models. Specifically, the specification states that "[a]lthough described herein in terms of service model graphs, the disclosed techniques are not so limited, and may be used in other types of graphs, and in any situation where the desire may arise to replace multiple icons or symbols may in a display with a simpler, more easily usable representation of multiple characteristics." *Id.* at 5:62–67. Thus, the Court rejects this portion of Plaintiff's construction.

Plaintiff further argues that the terms require construction because a "spotlight" has "characteristics" and "variable graphical images" has "attributes." The Court finds that this is explicitly recited in the claims. For example, claim 1 recites "the spotlight including a plurality of characteristics", and claim 8 recites "the graphical image having a plurality of attributes." Accordingly, the Court is not persuaded that it should confuse the claim language by including this recited aspect in the dispute terms.

Regarding the phrases that Defendant contends are indefinite, the Court finds that the claim language, "viewed in light of the specification . . ., inform[s] those skilled in the art about the scope of the invention with reasonable certainty." *Nautilus*, 134 S. Ct. at 2129. Claim 1 recites "displaying a spotlight with each of the nodes of the plurality of nodes." The claim language is straightforward and indicates that a spotlight is displayed with each node. The specification provides examples of a spotlight displayed with a node and states that "[t]he spotlights may be implemented using any graphical technique known to the art for placing a graphical image over or below another image on a screen." '929 Patent at 5:49–51, Figures 2 and 3. Moreover, neither the claims nor the specification require the spotlight to be within a "subjective" distance. Instead, the claims and specification make it clear that "the spotlights may be positioned anywhere relative to the node, including positions where the spotlight intersects, but does not surround the node, as well as positions where the spotlight does not intersect, but is separately positioned relative to the node." *Id.* at 5:57–61. In other words, the critical aspect is not the exact location of the spotlight, but instead is that the characteristic corresponds to one of the attributes of the service, as recited in claim 1.

For these same reasons, the Court finds that the phrase "a variable graphical image positioned with the node" is not indefinite. Claim 8 recites "represent the plurality of states with a variable graphical image positioned with the node." The claim language is straightforward and indicates that variable graphical image is positioned with the node. The specification provides examples of a variable graphical image positioned with a node. '992 Patent at Figures 2 and 3. Accordingly, the Court finds that the claim language, "viewed in light of the specification . . ., inform[s] those skilled in the art about the scope of the invention with reasonable certainty." *Nautilus*, 134 S. Ct. at 2129. Moreover, the Court finds that the phrases are unambiguous and are easily understandable by a jury. Therefore, the phrases will be given their plain and ordinary meaning.

c) Court's Construction

In light of the evidence presented by the parties, the term **"variable graphical image"** will be given its **plain and ordinary meaning**; the term **"spotlight"** is construed to mean **"a graphical image."** The Court further finds that the phrases **"a variable graphical image positioned with the node"** and **"displaying a spotlight with each of the nodes of the plurality of nodes"** are not indefinite, and will be given their **plain and ordinary meaning.**

V. CONCLUSION

The Court adopts the above constructions. For ease of reference, the Court's claim interpretations are set forth in a table in Appendix A. The parties are ORDERED that they may not refer, directly or indirectly, to each other's claim construction positions in the presence of the jury. Likewise, the parties are ORDERED to refrain from mentioning any portion of this opinion, other than the actual definitions adopted by the Court, in the presence of the jury. Any reference to claim construction proceedings is limited to informing the jury of the definitions adopted by the Court.

Within thirty (30) days of the issuance of this Memorandum Opinion and Order, the parties are hereby ORDERED, in good faith, to mediate this case with the mediator agreed upon by the parties. As a part of such mediation, each party shall appear by counsel and by at least one corporate officer possessing sufficient authority and control to unilaterally make binding decisions for the corporation adequate to address any good faith offer or counteroffer of settlement that might arise during such mediation. Failure to do so shall be deemed by the Court as a failure to mediate in good faith and may subject that party to such sanctions as the Court deems appropriate.

So ORDERED and SIGNED this 13th day of August, 2015.

Rodney Gilstrap

RODNEY GILSTRAP
UNITED STATES DISTRICT JUDGE

APPENDIX A

Term/Phrase	Court's Construction
"computer system" ('586 Patent)	Plain and ordinary meaning.
"prototype" ('586 Patent)	"an object in a namespace from which attributes, values, and/or children are dynamically inherited by another object"
"instance" ('586 Patent)	"an object in a namespace which dynamically inherits attributes, values, and/or children from another object in the namespace"
"object(s)" ('586 Patent)	"self-contained entity that contains data and/or procedures to manipulate the data"
"sharing the plurality of objects with a plurality of the one or more computer system" ('586 Patent)	"making accessible and/or sending the plurality of objects to a plurality of applications and/or a plurality of computer systems"
"hierarchical namespace" ('586 Patent)	"a memory or a plurality of memories coupled to one another containing a hierarchical set of objects such that any object in the set has a different name from all other objects in the set"
"dynamically inherits traits from the prototype" ('586 Patent)	"derives traits from the prototype that may change over time"
"wherein the values of the traits inherited from the prototype change dynamically" ('586 Patent)	"wherein the values of the traits inherited from the prototype change as or after the traits of the prototype change"
"traits" ('586 Patent)	"information about or representation of an object or an attribute, such as attribute values and/or child objects"
"meta data" ('898 Patent)	"data about other data"
"periodically" ('898 Patent)	Plain and ordinary meaning.
"script" ('898 Patent)	"set of instructions, procedures, and/or functions and related data written in an interpretable programming language"
"script-based program" ('898 Patent)	"program based on a script"
"service monitor" ('898 Patent)	"script-based program used by the performance management system to monitor a device, application, or server"

“resource” (’594 Patent)	The word “resource” is “intended in its broad sense to include, without limitation, hardware such as computers, printers, memory or other network devices, applications such as database management systems, and logical devices such as logical disk drives or filing systems”
“computer system” (’594 Patent)	Plain and ordinary meaning.
“discovery information” (’594 Patent)	“information about how to determine whether a resource is present on a computer system”
“interpreting the instructions” (’594 Patent)	“translating and executing the instructions with an interpreter”
“interpretable high-level computer programming language” (’594 Patent)	“a computer language that provides a level of abstraction from the underlying machine language, and can be translated and executed by an interpreter”
“uninterpreted form” (’594 Patent)	“form requiring an interpreter to translate and execute”
“stored on the storage device in their uninterpreted form” (’594 Patent)	Plain and ordinary meaning.
“status value” (’683 Patent)	“value indicating the status or condition of a node”
“**generating a graphical display**” (’683 Patent)	“generating graphical information for display on a display device”
“enterprise fault analysis” (’683 Patent)	“fault analysis in an enterprise”
“Impact Graph” (’683 Patent)	“topology or architecture of a specific fault model”
“inference policy” (’683 Patent)	“rule, or set of rules, for inferring the status or condition of a fault model node based on the status or condition of the node’s immediately down-stream neighboring nodes”
“impact policy” (’683 Patent)	“rule, or set of rules, for assessing the impact on a fault model node based on the status or condition of the node’s immediately up-stream neighboring nodes”
“east” (’683 Patent)	“least”
“mat” (’683 Patent)	“root”
“node” (’683 Patent)	“representation of a status or condition of a monitored component”
“nodes” (’683 Patent)	“a plurality of representations of status or condition of one or more monitored components”

“fault model” (’683 Patent)	“a directed graph (aka digraph) used for monitoring, diagnosis, and recovery of error conditions”
“fault model having a plurality of nodes” (’683 Patent)	Plain and ordinary meaning.
“enterprise” (’683 Patent)	“a collection of monitored components some of which may be hardware and some of which may be software”
“up-stream” (’683 Patent)	“in the direction against information flow”
“most up-stream” (’683 Patent)	“furthest in the direction against information flow”
“down-stream” (’683 Patent)	“in the direction of information flow”
“root cause” (’683 Patent)	“most up-stream node or nodes having a status value indicative of failure”
“impact value” (’683 Patent)	“value indicating whether a node is impacted or not impacted”
“connecting” (’093 Patent)	“joining or linking together”
“license certificate” (’093 Patent)	“an indication of the right to deploy software”
“exception indication” (’093 Patent)	“indication of a non-compliance condition or an unresolved connection”
“wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit” (’073 Patent)	“wherein the first and second indicator are each separately visible at the same time on a single display window of a display unit without requiring the user to perform any affirmative action (i.e., ‘navigate’)”
“subcomponent” / “IT subcomponent” (’073 Patent)	Plain and ordinary meaning.
“IT component processor” / “IT subcomponent processor” (’073 Patent)	Plain and ordinary meaning.
“importance” (’992 Patent)	Plain and ordinary meaning.
“service level agreement (SLA)” (’992 Patent)	“an agreement between a customer and a service provider, which may be an internal IT department or external organization”
“SLA violation” (’992 Patent)	Plain and ordinary meaning.
“graph” (’992 Patent)	“a visual representation”
“node” (’992 Patent)	“a representation of a component or service in a graph”
“variable graphical image”	Plain and ordinary meaning.

(’992 Patent)	
“spotlight” (’992 Patent)	“a graphical image”
“a variable graphical image positioned with the node” (’992 Patent)	Plain and ordinary meaning.
“displaying a spotlight with each of the nodes of the plurality of nodes” (’992 Patent)	Plain and ordinary meaning.