



Root Cause Analysis for  
End-to-End Service Delivery

---

Whitepaper

Copyright © Fidelia Technology, Inc.  
All Rights Reserved  
July 2004

## Root Cause Analysis for End-to-End Service Delivery

Today's businesses depend on their IT infrastructure to provide business services. The applications providing business services are distributed in nature, spanning multiple departments as well as geographical locations. Rather than focus on just "network downtime", root cause analysis in today's enterprises must concentrate on minimizing "business downtime". If a router fails, the immediate question that needs to be answered is "what business service did it impact" and what is the cost to my business.

Traditional systems limit root cause analysis to data networks. However, with the enhanced role of IT as an enabler of business, it is becoming increasingly important to extend this correlation to the "service" layer- i.e. to answer questions such as "why is my payroll service or banking service running slow". Getting an answer to such questions in real-time is critical to avoid impact to the business.

Fidelia NetVigil provides advanced root cause analysis (RCA) features that extend beyond traditional network level analysis. The root cause analysis engine is based on a Service Object Model designed for analyzing end-to-end *business* impact instead of just stopping at the network layer.

The tight integration of the root cause engine as part of the Fidelia NetVigil suite allows providing a seamless solution, which begins with discovering network elements as well as servers, databases and applications. Real-time alarms are triggered

based on approaching maximum capacity, traps, log messages, user defined maintenance, etc., taking into account the complex relationships between IT elements for delivering distributed applications. Fidelia's Business Visibility Engine uses tens of thousands of data samples ranging from the network all the way up to the application stack in order to identify the root cause of service degradation.

Fidelia NetVigil offers a distributed database and distributed processing architecture unlike any other legacy product in the industry. This distributed processing handled seamlessly by the Business Visibility Engine allows it to process root-cause data in real-time, as opposed to doing it after the fact. The multi-department security model allows doing analysis on services that span multiple departments as opposed to just the network or just the database.

This whitepaper describes the features of the Fidelia root cause engine and how it translates directly into lower downtime for your overall business.

### **Categorization of Root Cause Systems**

---

Root-cause systems are typically based on the following types identified below. Greater details on these methods can be found in other publications. There are probably no business advantages to using any one approach for event correlation,

since each type is best suited to a particular situation- in fact it could even be detrimental trying to shoe-horn all IT problems to a specific type of root cause methodology.

### **Rules Based**

These systems consist of a collection of 'rules' which get triggered based on the current state of various elements. The simplest example of a rule is "if X is true and Y is false, then Z is true". The advantage of a rule based system is that one can easily add arbitrary logic into the rules database since they are intuitive and easy to describe. However, it is almost impossible to create rules for every possible scenario in a large environment, and processing and scaling such a system is very difficult.

### **Model Based**

These systems break a complex system into smaller elements (models) and then describe the attributes and behavior of each model. The behavior is further described for the model itself, as well as related to other models. Using object oriented concepts such as inheritance, creating a model library even inside very large networks becomes an easily manageable task.

As mentioned earlier, practical implementations of root cause engines typically use more than one type of analysis. Ultimately for the user, trying to base a decision on which root cause "method" is better is meaningless. The underlying requirement for today's business environment still remains very basic: fast and accurate identification of the cause of business service failure.

## **Fidelia NetVigil - Root Cause for Business Services**

---

Most legacy root cause systems were designed for complex networks but hopelessly inadequate for correlating business service failures. Today's distributed applications are even more complex-delivery of service to the customer consists of applications layered across a complex IT infrastructure managed by different departments in different geographies.

NetVigil's root cause engine is primarily model-based, and was designed to be adaptable for today's distributed business applications. The model behavior is not isolated to the network layer, but actually defines the behavior between the different protocol stacks all the way to the application layer. To allow for analysis of complex business services which do not have well defined models associated with them, NetVigil allows creating rules to define the behavior of these business containers. This hybrid, best-of-breed approach delivers root cause analysis effectively to reduce MTTR where it matters most- the delivery of service to the customers.

### **DISCOVERY**

A rich and complete information database is essential for any root cause engine. The Fidelia root cause engine starts building its database by a complete and intelligent discovery on the network.

NetVigil's discovery process does a complete L2/L3 network discovery to detect the relationship between the various devices on the network. It discovers ATM connectivity, disks, controllers, VLANs, file systems, fiber channel switches, printers,

SAN, NAS devices as well as multiple, redundant paths in the network to prevent false suppressions.

The discovery process then discovers the capabilities, size, capacity, etc. of each element. It goes on to discover applications running on each of these devices such as databases, active directory, radius, DNS, mail, application servers, etc. Once it has discovered all these applications, it starts monitoring the behavior of these devices and applications based on their known model.

**RICH DATA INPUT SOURCES**

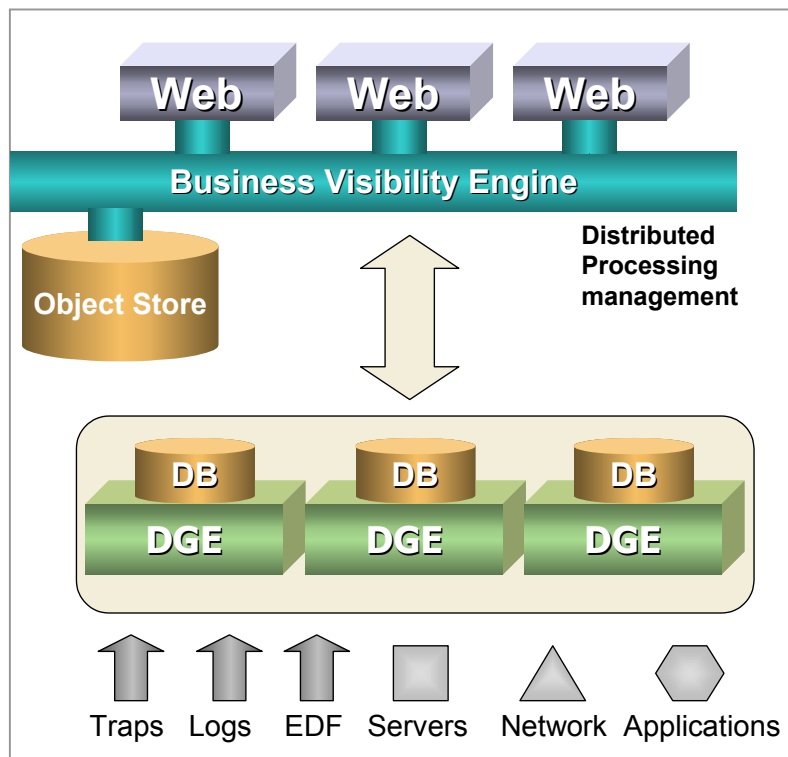
In order to determine accurate root cause of service failure, it is important to have a complete and rich set of data to get a holistic picture. NetVigil periodically polls each element in the network, each application and server, and stores the data for historical trend analysis. It also accepts SNMP traps, application logs, system events and external data sources via its API so as to get a complete picture of the IT network for its analysis.

**DISTRIBUTED, SCALABLE ARCHITECTURE**

Business Managers need root cause analysis of business services in real-time. Identifying the cause even hours after the fact is lost business and possibly revenue. A good RCA system for Business Services must be

scalable to handle a very large number of input data sources (network, servers, applications) and be able to analyze all data in real-time. NetVigil uses a loosely coupled distributed database and processing engine which allows it to scale to level not previously available in legacy architectures.

Because of the size of enterprise networks today, the number of managed elements is typically very large. A seemingly “simple” service such as tax and payroll processing

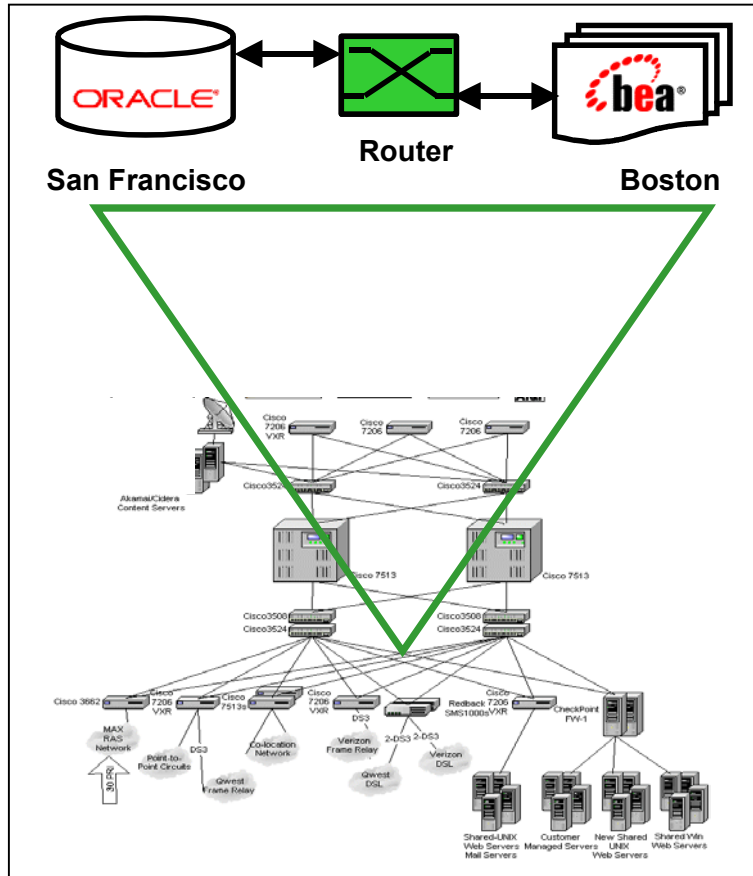


can involve a database, printer, payroll application and a network connecting all these remote elements. The database and payroll application run on servers, and the performance of these servers impacts the performance of the database and application. There are about 200 potential

measurement points that could impact the above service (ranging from memory on the servers, to bandwidth on the router, to Oracle transaction rates).

**INTEGRATED FAULT AND PERFORMANCE**

Trying to determine the cause of service degradation requires that the system be scalable for measuring the performance of tens of thousands of metrics. It must be extensible so that it can look holistically across networks, servers and applications.



**MULTI-DEPARTMENTAL SECURITY MODEL**

Business Services in today's enterprise span multiple departments- a simple payroll service might have the network managed by one group, the database by another group and the servers by a third group. Traditional systems require that each department get their own copy of the software product-however, you now have three independent copies of the software with no correlation possible between them.

In order to analyze performance degradation and not just failure, it is important to be looking at historical performance data in addition to the fault data.

NetVigil has a multi-departmental security model which allows each department to "share" the same system while maintaining their independence. However, since it is a shared system, it can analyze services that span multiple departments easily.

Fidelia incorporates a unified fault and performance engine that stores data for several years using very compact databases. This data is distributed on multiple databases and available instantaneously to the Business Visibility Engine for analyzing

performance degradation in addition to fault analysis.

### **Example – A Distributed eCommerce Application**

Consider the case of a ecommerce service distributed geographically across several different departments in an enterprise. There is a backup network link of lower capacity that is automatically used when the primary link goes down.

Fidelia's business containers automatically take into account the underlying topology, the behavior of each element in the network and the impact of degraded performance on the service. The containers are formed based on the individual elements, so they inherit the underlying model-based analysis. In addition, custom rules can be created to alter the behavior patterns based on the requirements of the business service. Due to the property of inheritance, rules need to be created only to alter the behavior on an as-needed basis (which is easily maintainable).

In the eCommerce example above, it is easy to create a business container in NetVigil which includes all the various elements that are part of the business service, and then add rules to account for the backup link. The root cause engine models the underlying infrastructure, correlates data from the network and analyzes the relationship between the protocol stacks and then applies any custom rules specified in the containers. If the server running the database is slow, it will be identified as the cause of degraded service performance and not the database because of the underlying behavior models.

Now consider the case of the primary link going down, and the backup link (of lower capacity) automatically being used. For all practical purposes, the service is still running and not impacted. However, because of congestion on the backup link, the service would run slow. Unlike other products which would be waiting for an alarm from the network in order to diagnose this problem, Fidelia's integrated performance features would quickly highlight that the congested link is the cause of the degraded service performance.

This is just one simple example of Fidelia NetVigil's Service Level root cause engine. This system is being used in production in business critical airline reservation systems, Sony Online gaming networks, retail store chains, healthcare and service providers environments. It is addressing the critical aspect of reducing business service downtime, and not just network level root cause.

*NOTE: Fidelia NetVigil successfully incorporates the design ideas described above to achieve a real-time root cause analysis engine that applies this advanced model behavior to Business Services.*