

# Scalability Requirements for Managing the World's Most Complex IT Systems

## *EMC Smarts Distributed Architecture*

Abstract: This paper provides an overview of the architecture, design principles, and implementation choices that have made Smarts the industry's most scalable service assurance platform.

Date 1/12/2006

Copyright © 2005, 2006 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC<sup>2</sup>, EMC, ApplicationXtender, Celerra, CentraStar, CLARAlert, CLARiiON, Connectrix, Dantz, Direct Matrix Architecture, DiskXtender, Documentum, EmailXtender, EmailXtract, HighRoad, Legato, Navisphere, PowerPath, RepliStor, ResourcePak, Retrospect, Smarts, SnapView/IP, SRDF, Symmetrix, TimeFinder, VisualSAN, and where information lives are registered trademarks and EMC ControlCenter, EMC Developers Program, EMC OnCourse, EMC Proven, EMC Snap, EMC Storage Administrator, Access Logix, ArchiveXtender, Automated Resource Manager, AutoSwap, AVALONidm, C-Clip, Celerra Replicator, Centera, CLARevent, CopyCross, CopyPoint, DatabaseXtender, Direct Matrix, DiskXtender 2000, EDM, E-Lab, EmailXaminer, Enginuity, eRoom, FarPoint, FLARE, FullTime, InfoMover, MirrorView, NetWin, NetWorker, OnAlert, OpenScale, Powerlink, RepliCare, SafeLine, SAN Advisor, SAN Copy, SAN Manager, SDMS, SnapSure, SnapView, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix DMX, and VisualSRM are trademarks of EMC Corporation.

All other brand names are trademarks or registered trademarks of their respective owners.

S0058

**Table of Contents**

- Introduction .....4**
  - The Need for Scalability ..... 4
- EMC Smarts Approach to Scalability .....4**
  - Organization of This Paper..... 5
- The Smarts Server's Efficient Internal Architecture.....6**
  - Design Principles for Scalability ..... 6
    - Codebook Correlation Technology™..... 6
    - Extensive Use of Asynchronous Processing and Multithreading ..... 7
    - Selective Monitoring..... 7
    - In-memory Smarts Repository ..... 7
    - Buffering Events and Data for High/Bursty Event Rates ..... 7
  - Flexible and Efficient Distribution Among Smarts Servers..... 8
    - Distributing Both Data and Analysis ..... 8
    - Seamless Integration of Distributed Processes into a Single System Image..... 9
  - Flexible Distribution Schemes ..... 10
  - Operational Flexibility ..... 10
- Examples of Distributed Smarts Implementations.....11**
- EMC Smarts in the Marketplace.....13**
- Conclusion.....14**

# Introduction

## *The Need for Scalability*

Today's IT infrastructures are larger, more complex, and more heterogeneous than envisioned even three to five years ago. They can contain millions of network, system, application, database, storage, security, and other elements, from hundreds of vendors, configured in meshed, multi-layer network topologies with complex application and service interdependencies. Each element can have thousands of parameters that can be monitored, thousands of alarms it can send, and a host of agents providing additional realtime status information. Managing such complex infrastructures requires sophisticated systems with the ability to scale in an unlimited fashion.

While increasing infrastructure complexity makes effective management ever harder to achieve, IT managers are faced with increasing demands to support profitability goals by meeting an aggressive set of cost and performance metrics. These new metrics—increased system availability and performance, faster mean-time-to-repair problems, rapid launch of new services, and consistent service-level agreement (SLA) compliance—create difficult scalability challenges for management solutions that must accurately diagnose and resolve service-affecting problems in real time. This paper provides an overview of the architecture, design principles, and implementation choices that make EMC Smarts the industry's most scalable service assurance platform.

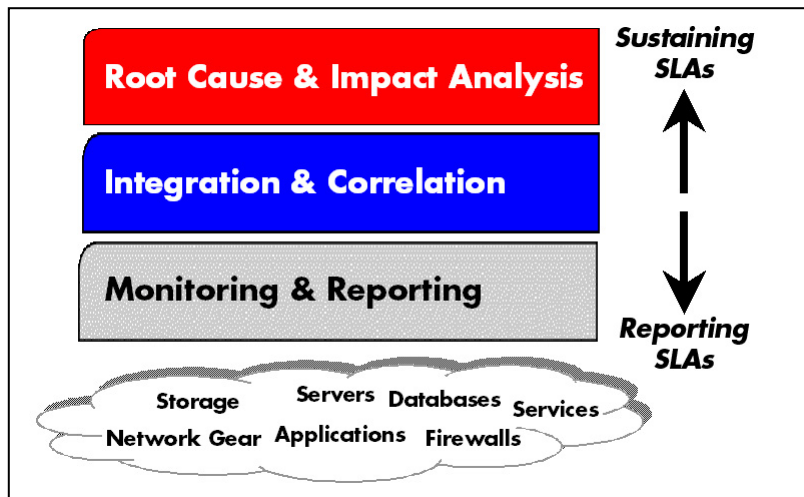
## **EMC Smarts Approach to Scalability**

Smarts' vision when designing its system in the early 1990s was to manage dynamic distributed systems of unlimited size and complexity, end-to-end across network, systems, and application domains. While some companies are only now beginning to address lingering and increasingly critical scalability issues, Smarts was designed from the start to manage the world's largest and most complex and dynamic infrastructures.

Smarts set out to achieve its vision by designing a management architecture that could be flexibly distributed, enabling management functions to run as close as possible to the systems they managed. This vision required the flexibility of a broad range of implementations, from element managers embedded in managed elements; through any number of layers of mid-level managers—each mid-level manager responsible for one or more domains distributed technology, or geographical or organizational, or any other boundaries; to a central manager across an organization's global IT infrastructure, all working cooperatively, and in synchrony, with one another.

Smarts' vision of "self-healing networked systems" was endorsed by the National Science Foundation and the Department of Defense, which awarded Smarts Innovative Research grants for the development of the Smarts Common Information Model, the Smarts distributed architecture, and Codebook Correlation Technology™ for Self-Healing Networked Systems. The grants played a key role in enabling Smarts to leverage innovative technology in practical products to solve the hardest problems faced by network and IT managers.

To date, only Smarts has created a highly distributed and scalable architecture not just for monitoring and event processing, but also for the higher value analysis functions of integration, correlation, and root-cause and impact analysis. As Giga Information Group observed in its Market Overview: Infrastructure Performance Management, March 9, 2001, customer requirements for service assurance solutions have shifted from simple monitoring and reporting to integration and correlation (of not just events but also topology) and root-cause and business impact analysis (see Figure 1). As a result, any discussion of scalability must address not just monitoring but analysis as well. EMC Smarts is the only company that has developed and implemented a highly scalable, distributed architecture across all layers of a highly intelligent service assurance solution.



**Figure 1: Layers of Intelligence in Infrastructure Management**

Giga Group uses the diagram in Figure 1 to illustrate the layers of intelligence required to sustain service. Beyond the simple monitoring at the bottom layer, Giga points out that management applications need to add integration and correlation of the monitored data within and across management domains, as well as root-cause and impact analysis to identify what needs to be fixed, and the priority of fixing it relative to business objectives.

## ***Organization of This Paper***

In designing the Smarts architecture, the approach to scalability centered on two principles:

- Make each individual Smarts server as efficient and scalable as possible<sup>1</sup>
- Support the most flexible distributed architecture, making possible any distributed configuration of Smarts servers that makes operational sense

This paper is organized along these principles, with the first section focusing on what makes a single Smarts server scale better than any other solution. In the second section, we describe Smarts' recursive distributed architecture. The unique features of this architecture enable Smarts deployments to scale to larger environments than any vendor, while including powerful analysis that no other vendor offers: integration, correlation, and root-cause and impact analysis. Finally, we conclude with some scalability facts from Smarts deployments around the world.

---

<sup>1</sup> The term "Smarts Server" refers to a software process running Smarts. Multiple Smarts servers can run on a single hardware server.

## The Smarts Server's Efficient Internal Architecture

Smarts servers scale to a larger number of managed elements than any other product in the marketplace. A single Smarts server can typically replace 5-10 servers of competitor products. This scalability is particularly impressive considering that Smarts solutions are the only ones to perform deep fault and performance analysis, in addition to the simple monitoring, filtering, and standard event processing provided by other systems.

Many single-server Smarts deployments today manage more than 500,000 objects, and in most cases Smarts users never reach the capacity limits of a single server. However, other considerations—such as customer boundaries, geographical locations, organizational boundaries, NOC responsibilities, different types of analysis for different domains—may cause organizations to distribute the implementation among multiple servers. The Smarts distributed architecture makes any such partitioning straightforward.

### *Design Principles for Scalability*

EMC Smarts uses several techniques and design principles to make individual Smarts servers as efficient as possible.

### **Codebook Correlation Technology™**

Smarts patented Codebook Correlation Technology (CCT) is a key contributor to the exceptional scalability of Smarts servers. CCT contributes to scalability in several ways:

- **Removes computation-intensive operations from the realtime critical path.** CCT pre-computes the topology-dependent correlation logic (in CCT, topology information is factored into unique problem signatures). This results in eliminating topology traversal—the most computation-intensive part of any correlation system—from the realtime critical path. Topology is re-traversed to update signatures only following changes—an occurrence that is many orders of magnitude less frequent than event receipt. Other systems that use topology in their processing—in particular those relying on downstream suppression—are forced to traverse the topology every time an event is received. This makes their processing too slow for all but very simple, small topologies.
- **Leverages the fastest, most robust computational algorithms.** CCT algorithms are orders of magnitude faster than any rules-based system. In addition to pre-computation of signatures, many other factors contribute to this speed:
  - Correlation is a fast vector-distance operation that matches events to problem signatures. This operation is orders of magnitude faster than searching for, retrieving, and executing all the rules that apply to each event, each time an event arrives.
  - The complexity of Codebook is linear in the number of managed elements—the number of signatures that must be matched is proportional to the number of managed elements—while the number of rules is exponential in the number of elements.
  - As symptoms arrive, Codebook uses incremental processing to narrow down which authentic problems™ might be the root cause.

## Extensive Use of Asynchronous Processing and Multithreading

Multithreading makes software more efficient by executing different parts of a program simultaneously. Asynchronous processing is essential for taking advantage of multiprocessor hardware architectures. We mentioned earlier that Codebook signatures are computed asynchronously with event processing. In addition, Smarts solutions leverage multithreading extensively across all processing functions, including monitoring, inter-process communications, signature computation, and Codebook Correlation. Some examples of asynchronous activities within a Smarts server include:

- **Multiple concurrent monitors.** Smarts extends monitoring efficiency by using multiple concurrent monitoring threads in each Smarts process. For further parallelism, monitoring functions can be distributed among multiple cooperating Smarts processes.
- **Multithreaded asynchronous polling.** Smarts leverages multithreading within each monitoring function. For example, it uses a high-performance, asynchronous ICMP poller with two asynchronous threads—one for sending polls and one for receiving polls. Because the send and receive threads operate separately, polling rates remain consistent regardless of poll response times.
- **Concurrent processing of publish and subscribe actions.** Smarts performs concurrent processing of requests issued by clients subscribing to be notified of events, and of notifications of events to subscribers.
- **Concurrent processing of queries.** Smarts processes queries against the Smarts Repository in parallel with other activities within a server. (The Smarts Repository is the representation of a specific customer environment based on the Smarts Common Information Model [SCIM]).

## Selective Monitoring

Many management tools collect all possible raw events and alarms and leave to skilled operators the task of finding the source of the problem. A tremendous amount of network and computing resources is wasted collecting these alarms, only to discard much of this data as soon as it arrives at the management system.

Smarts takes the opposite approach, focusing its monitoring and analysis only on those symptoms (data, events, alarms, traps) that are known to indicate service-affecting authentic problems. This approach reduces network traffic and event processing cycles consumed by the management system by orders of magnitude.

Smarts' policy-based management further contributes to scalability. Policy-based management allows users, or subscribing client applications (which may themselves be other Smarts servers), to further narrow down monitoring by specifying object groups, the specific problems to monitor for each group, and how often to monitor for them. This focus on collecting only relevant data reduces both bandwidth and CPU requirements, and allows Smarts solutions to scale to much larger environments than any other product.

## In-memory Smarts Repository

The Smarts Common Information Model (SCIM) Repository is a representation of a specific customer environment, providing a rich context for end-to-end integration and correlation of topology, events, data, and analysis from Smarts and third-party sources. Repository information is structured to support fast navigation. The SCIM Repository resides in memory for fast, realtime access, and asynchronous checkpointing to stable storage ensures reliability and fault-tolerance.

In contrast, many management systems store topology information on disk all the time, making access too cumbersome for fast, realtime processing. In addition, most systems maintain their topology in a collection of simple tables, making search and navigation both slow and inefficient.

## Buffering Events and Data for High/Bursty Event Rates

Smarts uses buffering to prevent loss of realtime data and events. In situations where thousands of events occur simultaneously, events are buffered as they arrive. Buffered events are asynchronously retrieved and processed as quickly as possible, smoothing out traffic bursts in the most optimal fashion.

# Flexible and Efficient Distribution Among Smarts Servers

## Distributing Both Data and Analysis

A fundamental principle of distributed processing is that scalability requires distributing not just the data, but also its processing. Therefore, to effectively scale to large and complex systems, solution architectures must be able to distribute management responsibilities among multiple cooperating servers, and at the same time assure that the results of different servers can be easily integrated and correlated at higher levels.

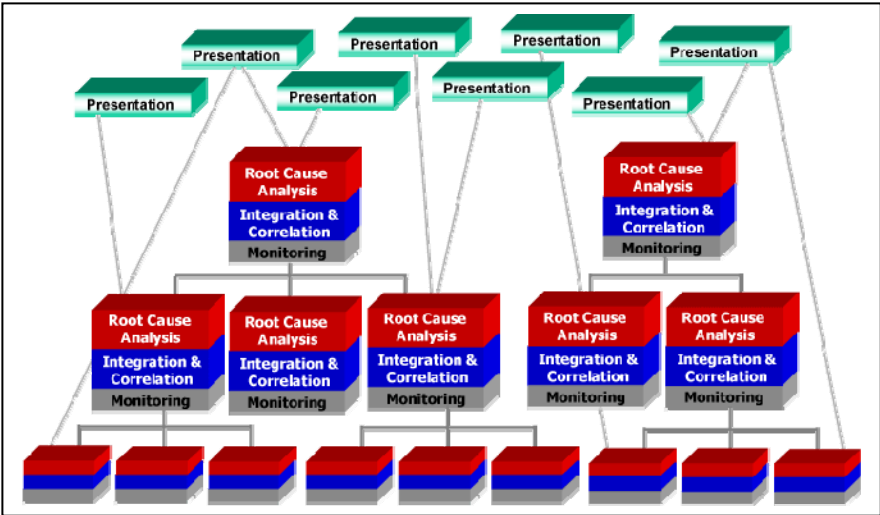


Figure 2a: Smarts solutions achieve scalability by distributing both the data and its processing among multiple cooperating servers.

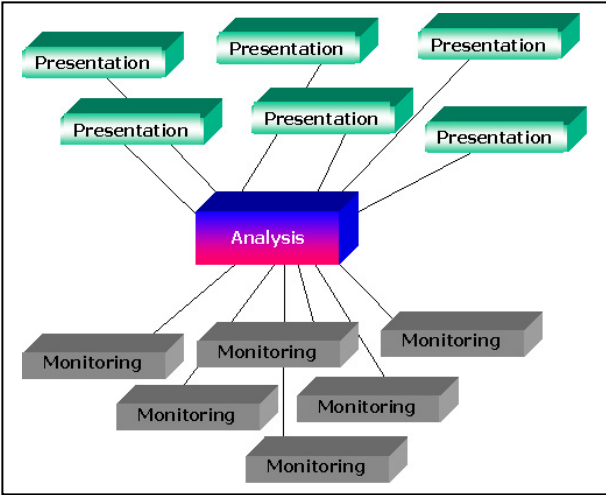


Figure 2b: This figure depicts the approach most vendors take, which focuses distribution on the data only. Collected data then needs to be funneled into a single server, creating a scalability bottleneck.

Smarts embeds this principle in its recursive distributed architecture (see Figure 2a). In a recursive distributed architecture, any process can be partitioned into sub-processes that cooperate to deliver the functionality of the original process. Similarly, the Smarts architecture allows partitioning of a management domain into sub-domains managed by separate cooperating Smarts servers. Each sub-domain has its own objects on which it performs both monitoring and analysis.

Smarts APIs allow servers to communicate with one another via a standards-based publish/subscribe mechanism. A Smarts server can act as a client, a server, or both of another Smarts server. As a client, a Smarts server can subscribe to be notified of certain events and authentic problems detected by the other server. As a server, it notifies its clients when the events they subscribed to are detected. As a result, solutions can deploy any number of Smarts servers working together in hierarchical, matrixed, or any other configuration of communicating processes. Figure 4 shows an example of a hierarchical, multi-tier distributed Smarts implementation. The results of these servers are integrated and correlated at each tier.

Many vendors do not understand the need for a distributed architecture to distribute both the data and its analysis, and instead focus their distribution on the data only. For example, some vendors support multiple collection stations, but the collected data then needs to be funneled into a single server, creating a scalability bottleneck. Other vendors implement a distributed object store to distribute objects among multiple servers, but their analysis requires knowledge of the global topology, again creating a scalability bottleneck that cannot be overcome (see Figure 2b).

## **Seamless Integration of Distributed Processes into a Single System Image**

A second important requirement for distributed processing is the ability to integrate the distributed results, and present users and other clients with a single system image. Presenting a single system image hides the fact of distribution altogether, so that the collection of distributed processes appears as if it were a single larger process. There are many advantages to this approach, including the flexibility to change the partitioning within a domain as needed without affecting any process that communicates with the manager for the domain.

Providing a single system image is not a trivial undertaking, since it requires supporting multiple levels of abstraction within a common information model. This allows the same object to be represented more or less abstractly, i.e., with less or more detail, at successive tiers in a hierarchical partitioning.

Smarts supports a single system image of distributed cooperating Smarts managers. It leverages the Smarts Common Information Model to support seamless roll-up across any number of tiers. In moving to a successively higher tier, information is represented at a higher level of abstraction, and details are dropped. The details remain available from subtending Smarts domain managers for seamless retrieval, and can be retrieved on demand by other applications and the GUI console or business dashboard. Multi-tier hierarchies that automatically roll up to a higher level of abstraction at successive levels and provide a single system image are key to making distribution practical.

Many management vendors do not support a common information model. Instead, they use a simple topology database that cannot support multiple levels of abstraction for the same class of object. As a result, they cannot synchronize distributed processes into a single system image without confronting the very scalability bottleneck they were trying to avoid via distribution.

Some vendors limit their distributed architectures to two or three-tier hierarchies in recognition of the inherent scalability limits of their approach. In contrast, Smarts offers unlimited multi-tier hierarchical configurations, as well as non-hierarchical configurations.

## ***Flexible Distribution Schemes***

When designing a distributed Smarts implementation, architects have tremendous flexibility to define the boundaries of a domain managed by an Smarts server. One option is to divide responsibilities by topology. In this case, domain boundaries are aligned with topology boundaries such as geographic regions, operations centers, organizational responsibilities, etc. This distribution is extremely straightforward. For example, to distribute the load of a single Smarts server among two or more Smarts servers, the administrator simply configures the discovery filters of the distributed servers to divide the topology among them. An EMC Smarts Service Assurance Manager configured as a client of these distributed servers automatically synchronizes both their topology and analysis results, presenting a single system image of the combined domains.

In practice this means that clients of this Service Assurance Manager are not even aware that several Smarts servers are cooperating to provide this single system image. When an object in Service Assurance Manager is accessed, its details are retrieved automatically from the subtending Smarts servers. The maps of this Service Assurance Manager show a single seamless domain. No other product offers such seamless distribution.

A second alternative unique to Smarts is the ability to align domain boundaries with the type of analysis performed. For example, the responsibility of managing a single network domain could be divided between two Smarts servers: one focusing on device performance problems and the other focusing on network-level connectivity problems of the same devices.

Aligning server boundaries with type of analysis provides a future-proof foundation for accommodating constant technology evolution—Smarts users can add new technologies and new analysis servers to their existing architectures without disrupting pre-existing solutions.

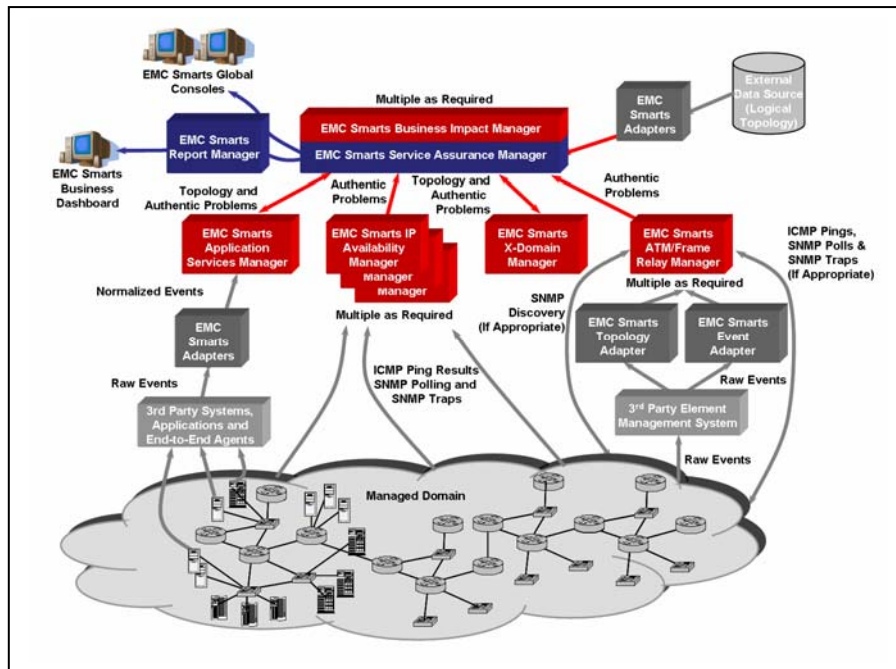
For example, service providers already using Smarts to manage their ATM/Frame Relay core, IP network, and VPN services can integrate EMC Smarts MPLS Manager into their deployments. MPLS Manager interacts with the other Smarts servers via Smarts publish/subscribe interfaces.

The flexibility to add new Smarts solutions to manage new aspects of existing infrastructure elements is analogous to the ability to add new blades supporting new technologies to an existing chassis. Users benefit from an easy transition to a more powerful solution for their enhanced infrastructure.

## ***Operational Flexibility***

A key benefit of Smarts' innovative distributed architecture is operational flexibility. An organization can architect the solution to fit its operational, geographic, or organizational structure, and can incrementally add new solutions to enhance management functionality without disrupting existing Smarts solutions. This makes Smarts much more operations-friendly than systems that require organizations to change their processes to fit into the limitations of their product architecture.

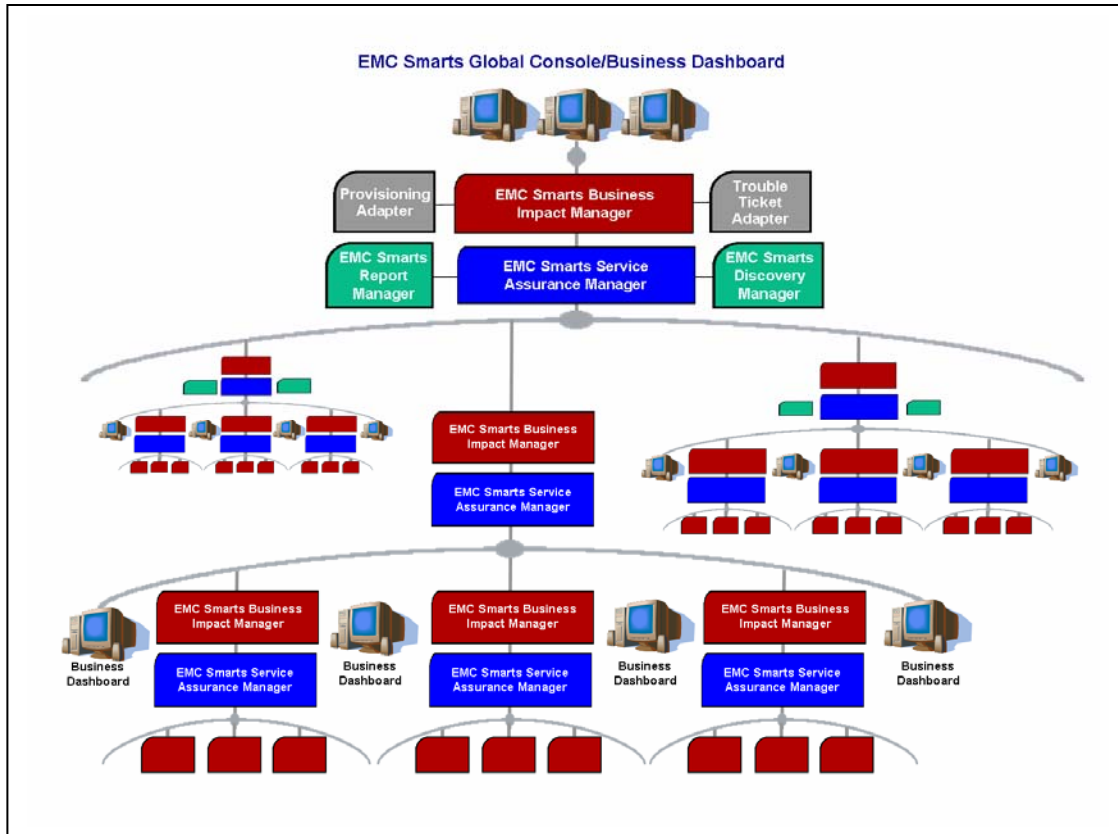
## Examples of Distributed Smarts Implementations



**Figure 3: A Typical Smarts Deployment Architecture**

Figure 3 depicts a typical Smarts deployment architecture that uses a number of Smarts solutions to achieve end-to-end management of the infrastructure.

- EMC Smarts IP Availability Manager manages the availability of SNMP-manageable WAN and LAN devices ("customer" network elements).
- EMC Smarts ATM/Frame Relay Manager manages the availability of generally non-SNMP-manageable Frame Relay and ATM switches, using a combination of vendor-proprietary element management systems (EMSs) and SNMP polling as a source of topology and events.
- EMC Smarts Service Assurance Manager integrates and correlates information from other Smarts analysis servers as well as the Smarts Adapters that receive and process events from third-party monitoring solutions. It provides a single end-to-end view of the managed infrastructure; provides a single connection point for all users and northbound integration; manages user interaction and presentation; and supports hierarchical implementations for multi-tiered management and reporting.
- EMC Smarts Business Impact Manager calculates problem impacts for easy problem prioritization.
- EMC Smarts Cross-Domain Correlation correlates events from different subtending Smarts domains. It subscribes to edge topology and events from Service Assurance Manager from both IP Availability Manager and ATM/FR Manager and performs cross-domain correlation to diagnose failures and identify their impacts.
- EMC Smarts Application Services Manager subscribes to discovered topology and authentic problems from AM and events from third-party application monitoring agents to perform correlation at the application services layer, pinpointing problems that affect application services and these problems' impacts.
- EMC Smarts Global Console provides a single point for users to interact with the managed infrastructure.
- EMC Smarts Adapters mediate information provided by or to third-party products.
- EMC Smarts Business Dashboard presents customized summary and detail views via the web.
- EMC Smarts Report Manager provides standard and custom reports based on the event data archived to a database server. Users access reports via a browser-based client.



**Figure 4: A Hierarchical, Multi-Tiered Distributed Smarts Architecture**

Figure 4 shows an example of a Smarts architecture for a multinational corporation with network operations centers (NOCs) in EMEA, North America, and Asia. Each NOC has visibility to three or more regional NOCs. The North America NOC, for example, has visibility to the Eastern, Central, and Western Regional NOCs. Each Regional NOC implements one or more Smarts applications, such as IP Availability Manager, ATM/Frame Relay Manager, and Applications Services Manager. Each regional NOC uses Service Assurance Manager to provide an end-to-end view of the region, and Business Impact Manager to provide impact analysis on the business the region supports.

## EMC Smarts in the Marketplace

Smarts scalability is proven in the marketplace—more than 120 of the Fortune 1000, including eleven of the top 12 North American and UK service providers, are EMC Smarts customers. These organizations know first-hand the exceptional scalability offered by Smarts.

- One of the world's largest retailers chose Smarts to manage its massive 60,000+ device network spread over 4,200 locations. The retailer deployed Service Assurance Manager, IP Availability Manager, IP Performance Manager, and Business Impact Manager in every store and distribution center, gaining visibility into the root cause and impact analysis of every authentic problem that occurs in every location, down to the cash register level. All events feed into regional mid-tier Service Assurance Managers, and from there to a top-level Service Assurance Manager at the company's command center.
- A market-leading software company chose Smarts to manage more than 370,000 ports and interfaces in its internal global corporate network supporting more than 50,000 employees. At this company, Smarts runs on Pentium 750 Xenon quad processors running WIN2K. The company replaced 18 competitor servers with 4 Smarts servers: one for the lab/technical facilities at the corporate headquarters, one for other headquarters employees, one for the rest of the US, and one for the rest of the world.
- A global managed network, system, and application service provider manages more than 1 million customer ports and interfaces with Smarts. This company deployed 40 Smarts primary servers and 40 backup servers in a fault tolerant architecture. The partitioning among servers was driven by business considerations such as customer (e.g., contractual requirements), NOC, and organizational boundaries.
- A large financial services company had been managing its network—including 790 routers and 510 switches supporting more than 25,000 internal and external users—using 14 of a competitor's servers. When the company deployed Smarts, it determined it could manage the same network with only one Smarts server, while gaining better auto-discovery and powerful analysis the previous product could not deliver. The company chose to implement two Smarts servers in a fail-over architecture—one server for primary processing and one for backup.
- A global managed service provider manages networks for 200 of the world's largest companies. Scalability features that sold this company on Smarts include: highly efficient data acquisition and analysis, distributed architecture, ability to scale to manage carrier-class networks with millions of ports and interfaces, and the ability to support multi-layered Smarts server hierarchies. This MSP runs multiple instances of Smarts on a single host system so that several large customer networks can be managed, with individual configurations, from a single server.
- A large U.S. service provider that averaged 300,000 alarms per day using the market-leading network management system was able to reduce the volume of events to 200 per day on average. All 200 resulting alarms represented authentic problems. The ability to dispatch technicians to address only authentic problems resulted in significant cost savings, leading to unprecedented ROI.

## Conclusion

To achieve the vision of managing the world's largest, most complex dynamic distributed systems, the Smarts engineering team drew on the most advanced distributed systems software technology principles. The team leveraged object-oriented design and development, programming languages, compilers, operating systems, middleware, networking protocols, and other software technology disciplines to provide the most scalable system in the industry. This paper summarizes the key principles applied to reach these goals.

Smarts scalability advantages fall into two main categories:

1. Assuring the most efficient implementation of each Smarts server.
2. Supporting the most flexible and robust distributed system architecture.

Smarts' recursive distributed architecture meets two important requirements:

1. Distributing both data collection and data analysis to eliminate the scalability bottlenecks of systems that distribute only the data.
2. Seamlessly synchronizing multiple cooperating distributed Smarts processes to provide a single system image.

Smarts scalability advantages have been proven repeatedly in the marketplace, as Smarts currently successfully manages the world's largest and most complex enterprise, service provider, and government infrastructures.

For more information, see the following whitepapers:

High Speed and Robust Event Correlation

Automating Root Cause Analysis