SOA Management with Integrated solution from SAP and Sonoa Systems
A report from SAP Co-Innovation Lab

Sonoa Systems:
Ravi Chandra, Co-Founder & VP of Engineering
Kishore Sannidhanam, Business Development

SAP Co-Innovation Lab:
Canyang Kevin Liu, Joerg Nalik, Siva Gopal Modadugula

November, 2008
# Table of Contents

1 INTRODUCTION .................................................................................................................................1

2 INTEGRATION OF SONOA SYSTEMS’ COMPONENTS INTO SAP COIL LANDSCAPE ....1

  2.1 THE BASELINE COIL LANDSCAPE ..................................................................................................2
  2.2 SOA MANAGEMENT SOLUTION FROM SONOA ........................................................................3
    2.2.1 Sonoa Systems Concepts and Architectural Overview .........................................................3
    2.2.2 Sonoa ServiceNet’s Key Components ..................................................................................5
  2.3 NETWORK INTEGRATION ..............................................................................................................5
    2.3.1 Recommended Deployment Pattern for SOA Management Proxies .................................6
    2.3.2 The Integrated Landscape ..................................................................................................6

3 PERFORMANCE AND STRESS TEST RESULTS ..............................................................................7

  3.1 IMPACT TO RESPONSE TIME ......................................................................................................8
  3.2 STRESS TEST RESULTS ..............................................................................................................10

4 CONCLUSION ......................................................................................................................................13

5 REFERENCES .........................................................................................................................................13
1 Introduction

In today’s Web-driven environment, traditional enterprise boundaries are being fundamentally reshaped. Enterprises can no longer survive just as largely self-contained silos. The success of an enterprise depends more and more on how well they are integrated with their business networks where hundreds of customers and partners each may require a unique relationship. Enterprise SOA adoption is a critical part of business network transformation and provides a more flexible and agile approach for enterprises to offer specialized business services to interweaved business networks. A successful SOA adoption brings enterprises new levels of competitive differentiations and new revenue opportunities.

From IT perspective, Web services technologies form the underpinning of SOA adoption, and as the usage of Web services increases, more and more sensitive business data may flow in and out the web and critical operational and performance problems can arise. SOA management addresses the management challenges introduced by SOA. It provides the capabilities to safeguard, operate, and evolve an enterprise’s SOA runtime production environment.

SOA Management is best viewed as a solution, not a product by itself, and the overall solution may utilize the capabilities of several infrastructure products from multiple vendors to synergistically address the SOA management challenges. SAP seeks to offer its customers the very best SOA management solutions. With thousands of packaged high quality enterprise services delivered from SAP, customers can immediately reap the benefits of SAP’s proven design time governance that focuses on building alignment of SOA infrastructure with the key business processes. Once customers decided to put these enterprise services into production use in their own SOA landscape, they can leverage SAP Solution Manager and SAP NetWeaver Administrator as the cornerstones for runtime management of these enterprise services. SAP Solution Manager is an industry standard ITIL based solution that provides first class application management capabilities, including growing features for managing the challenges and opportunities of SOA landscapes. With a consistent management approach for SOA and non-SOA components in a customer’s landscape, SAP Solution Manager ensures low TCO based on proven IT management best practices.

In addition to the SOA management capabilities provided in SAP Solution Manager and SAP NetWeaver Administrator, SAP works closely via its Co-Innovation Lab (COIL) with its customers and leading SOA management vendors in defining integrated innovative solutions to address management challenges of customers’ highly heterogeneous SOA landscape. COIL is one of the newest communities of innovation that make up the SAP ecosystem. It serves as a hands-on working environment for SAP, its customers, and partners to execute joint projects, and work on proof of concepts, enabling them to discover and promote new business applications and technology solutions. Via the best practices and integrated solutions for SOA management developed in COIL, customers benefit from reduced integration costs and faster innovation cycles while gaining a new generation of highly agile and adaptable solutions. The cooperation between SAP and leading SOA management vendors gives customers the freedom to choose which solution fits best with their infrastructure.

This paper provides a summary of the integrated SOA management solution developed in COIL by SAP and Sonoa Systems. Based in Santa Clara California, Sonoa Systems is a privately held company focusing on solutions for SOA management and Cloud gateways. Its ServiceNet network appliance enables enterprises to manage Web services with secure data protection, flexible access control, and enhanced performance.

2 Integration of Sonoa Systems’ Components into SAP COIL Landscape

© 2008
SAP LABs
An SAP Enterprise Service Community Paper
April 2008
Page 1
Every customer may have a different landscape, and it would not be possible to replicate each and every one of them in the lab. At COIL, we started with a simplified SAP landscape which still includes the key elements of a modern data center and should be able to provide a common pattern for SOA management solution integration. Complexity of the landscape may be increased in future phases of the project.

2.1 The baseline COIL Landscape

The COIL landscape is described in the following figure, and has the following characteristics:

- It includes a SAP Portal 7.0 server, a Composition Environment 7.10 SP3 Server, and an emulated ECC 6.0 backend which provides simulated SAP enterprise services with CE 7.10
- All application instances are deployed inside VMware virtual machines
- All applications are deployed into two instances to demonstrate high availability and scalability
- A commercial Application Delivery (AD) appliance is added which provides the following services:
  - Providing a virtual server proxy endpoint for the two instance of an application component and load balanced routing of network traffic to the component instances
  - Termination of incoming SSL connection and decryption of such traffic
  - Optionally applying routing rules and transformation operations on traffic content
  - Connection multiplexing of many incoming TCP/IP connections to fewer connections towards the server side
  - Optional re-encryption of traffic towards the application components

![Figure 1 - COIL baseline landscape](image)

On top of the above production-like landscape, COIL also provides a simple test application which leverages all the key components above. As shown in the screenshots below, the application addresses a Customer Fact Sheet (CFS) scenario. It allows a sales person to log in to the Portal and look up customers. A clicking on a customer record displays a quote and an order history list for that customer. Behind the scene, web service calls from the Portal to a composite application running on CE are triggered, and in turn the composite triggers further web service calls to a backend system where the customer data is persisted.

In the first step, the user enters a customer name to search for a customer, and a list of matching customers are returned. Clicking on a line item in the returned customer list gets more details about the customer and the history of quotes and orders submitted by the selected customer.
The whole scenario can be simulated using load testing tools. In COIL, we use HP Load Runner for performance testing.

2.2 SOA management solution from Sonoa

Sonoa ServiceNet is a network appliance that mediates traffic between service providers and consumers. The appliance supports service virtualization, policy development and enforcement, performance monitoring and reporting, and a full range of visibility and control required for service operations.

2.2.1 Sonoa Systems Concepts and Architectural Overview

Key modules of Sonoa Systems’ ServiceNet runtime are illustrated in Figure 2.1 and described in the table followed.
### Sonoa ServiceNet Module

<table>
<thead>
<tr>
<th>Sonoa ServiceNet Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Policy Enforcement Module</strong></td>
<td>As the heart of the runtime, this module implements the enforcement model, which includes the request and response cycle, and the policy hierarchy used to enforce policies. The request and response cycle has five major phases: a client request phase, a routing phase, a target request phase, a target response phase, and a client response phase. Policy hierarchy support includes: operational domains, sets of services grouped into applications, individual services, and operations within services.</td>
</tr>
<tr>
<td><strong>Acceleration Module</strong></td>
<td>This module includes the software and hardware required for low latency, high throughput, and high concurrent load policy enforcement. Acceleration support includes IP acceleration, Crypto acceleration, and XML acceleration.</td>
</tr>
<tr>
<td><strong>Security Module</strong></td>
<td>This module implements functions that enforce security related policies. Security is a major functional component of policy enforcement and requires a lot of specialized processing support provided by ServiceNet.</td>
</tr>
<tr>
<td><strong>Protocols Module</strong></td>
<td>This module implements the transport and payload protocols required for supporting a broad range of services. As discussed previously, the diversity in definition of services requires flexible protocol and payload support. This module facilitates flexibility by implementing payload mediation methods that allow efficient insertion, deletion, and replacement of payload elements.</td>
</tr>
<tr>
<td><strong>Caching Module</strong></td>
<td>This module implements caching support that allows for caching various types of data, including policy decisions obtained from external systems, data obtained from external systems, as well as responses received from services. Caching can provide significant reduction in latencies and offload processing from other servers in the deployment environment.</td>
</tr>
<tr>
<td><strong>Management Module</strong></td>
<td>This module maintains the policy configuration to be used for policy enforcement at different points in the message flow.</td>
</tr>
<tr>
<td><strong>Traffic Control Module</strong></td>
<td>This module implements the traffic control policies. Traffic control policies typically involve maintaining various statistics and using these statistics to take specific actions, such as rejecting requests, raising alerts, throttling or queuing requests based on customer profiles.</td>
</tr>
<tr>
<td><strong>Context Integration Module</strong></td>
<td>This module is responsible for integration with external systems required for policy enforcement. As previously mentioned, external systems may provide policy decisions and additional data to be used for</td>
</tr>
<tr>
<td>Monitoring Module</td>
<td>This module is responsible for enforcing all logging and auditing policies. Logging and auditing support includes collecting and transforming data for better usability and protection, among other requirements.</td>
</tr>
</tbody>
</table>

Aforementioned modules enable ServiceNet to address the following challenges in Web Services delivery:

- **Security and Compliance**: Enforcing security models and ensuring regulatory and contractual compliance with authentication, authorization, encryption, signatures, credential mediation, auditing, logging, client and target specific SLA monitoring, schema validation, and so on.
- **Traffic Management**: Meeting business priorities for service usage with client- and target-specific rate limiting, target concurrency limiting, quota management, traffic shaping, traffic prioritization, and so on.
- **Performance Acceleration**: Reducing the load on servers in the deployment environment with response caching, credential caching, profile caching, SSL termination, data compression, schema validation, and so on.
- **Fault Management**: Mitigating the effects of outages, exceptions, and maintenance with logging, debugging, fault isolation, fault generation, exception routing, and so on.
- **Personalization**: Easy enforcement of different policies for different categories of clients and targets with profile lookup, policy selection, target selection, and so on.
- **Mediation**: Mediation between the client’s view of a service and the actual implementation of the service offered by the target with transport mediation, payload mediation, content-based routing, version mediation, and so on.

### 2.2.2 Sonoa ServiceNet’s Key Components

Sonoa ServiceNet is offered in a few modules including the 3000 series and the 5000 series. For this exercise at COIL, ServiceNet 5000 was used. ServiceNet 5000 Series comes in a 2U rack-mountable form factor and offers enhanced performance and expendability over the 3000 Series. The 5000 Series appliances offer faster processing, more storage, and more memory capacity than the 3000 Series. In addition, the 5000 Series has the capability to support Sonoa’s advanced XML acceleration technology. ServiceNet 5000 has 8 core processor, with 4 GB Ethernet interfaces, redundant power supplies and RAID disk system. The software running on the appliance is a combination of C and Java runtime running on a highly optimized Linux Kernel.

Along with ServiceNet Runtime, the following key components were also used in:

- **Control Center** – Browser based tool intended for the application / security operators to proxy and configure services
- **Design Studio** – Eclipse based developer tool geared towards the Architect / developer to create custom policies to control the behavior of services
- **CLI** – Command line tool for the data center administers to monitor and provision the appliance in the data center

### 2.3 Network Integration

The most relevant Sonoa Systems run-time component to be added to the COIL landscape is the ServiceNet appliance which can manage multiple runtime enforcement points for Web services policies.
2.3.1 Recommended Deployment Pattern for SOA Management Proxies

ServiceNet proxies provide virtual Web service endpoints, and route web service requests, after applying policies and load balancing to multiple physical web service endpoints.

As described in Figure 2.3, for our landscape, we determined that the best ServiceNet deployment pattern was to put one in front of each SAP application instance. When it’s necessary to have an Application Delivery (AD) solution in the landscape, ServiceNet fits better sitting between the AD and the application instances.

2.3.2 The Integrated Landscape

ServiceNet’s Control Center design-time and/or Design Studio can be used to provide policy management and design capability. The overall landscape with Sonoa Systems’ components looked like the following.
COIL landscape also includes a NetApp file system storage which is made available on the local network as so called Storage Attached Network (SAN). During our tests with Sonoa Systems, the SAN was mounted as a NFS mount in ServiceNet to facilitate logging. ServiceNet automatically mounts the disk based on the configuration. Mounting the disk manually is not required.

**Figure 3 - The physical view of the integrated COIL and Sonoa Systems landscape at COIL**

### 3 Performance and Stress Test Results

A number of performance and stress tests were performed to measure the additional response time and resource usage impact of the Sonoa Systems solution.

Summary of key findings is listed in the subsections further below:

- In our message-size ramp-up testing, we didn't detect Response time increase from Sonoa ServiceNet within our 5% error margin range.
CPU consumption was very low which means that the Sonoa appliance can handle higher volumes of Web services traffic than required by our landscape.

### 3.1 Impact to Response Time

As described in section 2.1, the COIL landscape supports various test cases. To measure response time impact of Sonoa components, our test case focus on the Web services traffic between CE and the ECC Backend. HP LoadRunner emulates a Servlet running on CE that calls a set of ECC services to get customer basic data, quote history, and order history. This test was designed to only focus on web services traffic.

In order to measure the response time impact for different size of web-service calls and for different policy configurations in the ServiceNet proxies, the SAP backend component was configured to increase the response data set from call to call linearly. With the first servlet call, the response message contains 10 sale order line items (message size around 30KB); second call with a response message containing 30 line items, so on so forth - with each of the following call, the response message contains 20 more items. After 50 iterations, the response message is reset back to 10 line items, and the increasing cycle starts again. To measure the consistency of the impact, the test contains three cycles of increasing.

The baseline results represented by the think purple line in the graphs below did not include any SOA management.

The following diagram shows the result of the message size ramp-up test with the the following landscape configuration:
- Load Runner configuration: 1 user, 1 second think time, 3 increasing cycles
- Components in place: CE→Sonoa ServiceNet ←→ ECC
- Transport: HTTP all the way
- Policies enforced by ServiceNet: a “badGuy” content-checking policy is enabled.
Fig. 5.1: Response times from size ramp-up test 1: Sonoa vs SAP Baseline.

In figure 5.1, the saw-tooth like thick green line is the measurement of the landscape with Sonoa proxy sitting between CE and ECC backend. The purple thin line is the measurement of the SAP landscape without any SOA Management solution in between. As can be seen in picture 5.1, three findings are worth noting:

1. The response time follows a clear linear increase as the load increases.

2. The response time doesn’t show large fluctuations with Sonoa. Fluctuation in the diagram is mainly due to the garbage collection activities in the SAP java stacks in the landscape. It is not unusual for Java applications to occasionally have longer response times due to Garbage Collection short time holds of the Java VMs. In this measurement there have been synchronous calls from the CE-Composite component to the SERVICENET and then to the ECC-ERP application and back. Hops through Java based components increase the likelihood of more response time from Java Garbage Collection, in particular for calls with larger web-service content sizes. The low response times in Figure 6.1 are occasional calls where no Garbage collection occurred and therefore they follow the systematic linear increase as it is expected from a sequence of linearly in size growing web service calls.

3. Compared to the SAP baseline data, the overhead added by Sonoa is not significant. When the message size is small, the response times are pretty much same with or without Sonoa. When it reaches the end of each iteration (approaching 3mb message size), Sonoa results appear to be higher. We had doubts about why it’s such case, so we continued with another test described below.

Though Sonoa was able to show a very good result in the above test, we realized that the above results are slightly tainted by gzip impacts on the SAP application server, and the comparison of the Sonoa result to the baseline result is not a strict apples-to-apples comparison.

To get a better comparison, we repeated the Sonoa measurements with some change of the Sonoa landscape configuration: we explicitly disabled compression processing in the SAP application server, thus Sonoa got uncompressed traffic from the SAP server. With such change, the SAP application server does the same as with our baseline measurements. We also turned on SSL termination in Sonoa to assess the impact of HTTPS. Now this is an apples-to-apples comparision:

- In SAP baseline test, F5 provides SSL termination and load balancing, F5 handles gzip compression processing
- In Sonoa test, Service Net provides SSL termination and load balancing, Service Net handles gzip compression processing
- Policies enforced by ServiceNet: a “badGuy” content-checking policy is enabled,

The following diagram shows the new result.
In Figure 5.2, the thicker line is the Sonoa result, and the thin line is the SAP+F5 baseline result. We are glad to notice that the response time measurement with Sonoa in place appears to be even faster by a small amount of about 3% than the SAP+F5 baseline. If the 3% acceleration would be real, it might mean that the SAP server is somewhat becoming faster with the compression processing off-loaded to ServiceNet and at the same time the additional processing for compression inside Sonoa practically took no time. However, we don't have a solid understanding for such acceleration effects, and tend to attribute the slight difference rather to the significant level of random statistic fluctuations which we see in the measurements too.

Given the results above, we can conclude that no performance difference with and without the Sonoa appliance could be detected in our tests above the maybe 5% accuracy of our measurements. In other words, the Sonoa SOA management policy enforcement service comes with no detectable response time degradation in our test landscape.

### 3.2 Stress Test Results

A few stress tests were performed with variations of COIL landscape. In all these tests, the number of users working in the NW Portal was linearly increased over a period of 1+ hours, followed by a 5+ hour period of constant high load. While the ramp-up phase demonstrate linear resource consumption proportional to the load, the high load phase checks for stability and reliability over time.

The following diagram shows the result of the message size ramp-up test with the following landscape configuration:

- Load Runner configuration: ramp up from 1 to 70 users, 10 second think time, constant message size of 30KB
- Components in place: CE → Sonoa ServiceNet ↔ ECC
- Transport: HTTP all the way
- Policies enforced by ServiceNet: a “badGuy” content-checking policy is enabled,

![Average Transaction Response Time](chart1.png)

**Fig 5.3** Response time measurement from user ramp-up test 1 (http) with Sonoa

As shown in Figure 5.3, during the long running test, the response times increased during the ramp-up phase only a little. This is quite normal behavior for this type of testing in landscapes which have a lot of inbound and outbound queues in the various components. Under high load, the response time is pretty consistent.

![UNIX Resources](chart2.png)

**Fig 5.4** Resource consumption measurement from user ramp-up test 1 (http) with Sonoa
Figure 5.4 indicates that in multi-user load testing the Sonoa appliance showed only 0.35% CPU usage which is a very good low number indicating great scalability of the Sonoa solution. In other words, the Sonoa appliance can handle much bigger load than demanded by the COIL landscape.

To measure the impact of HTTPS, we repeated the same test above with one change to the landscape - ServiceNet is now used for SSL termination.

Compared to Figure 5.3, the response time slightly increased under high load, but still consistent with good numbers.

![Graph showing Average Transaction Response Time](image)

**Fig 5.5** Response time measurement from user ramp-up test 2 (https) with Sonoa

![Graph showing UNIX Resources](image)

**Fig 5.6** Resource consumption measurement from user ramp-up test 2 (https) with Sonoa
Compared to Figure 5.4, Sonoa appliance CPU usage with HTTPs is a little more than doubled than the http case, but with 0.75% usage of CPU, it's still a very good result.

4 Conclusion

Based on the testing performed at COIL with our specific landscape and the above described Sonoa ServiceNet components, we conclude that no performance difference with or without the Sonoa ServiceNet appliance could be detected in our tests above the 5% accuracy of our measurements. In other words, the Sonoa SOA management policy enforcement service came with no detectable response time degradation in our test landscape.

A 6+ hour stress test conducted in the SAP Co-Innovation Lab concluded that Sonoa added very little to zero latency while performing intelligent policy enforcement for a group of SAP enterprise services that conduct over 20,000 sales order and quote history lookups per hour with each response message containing 100 line items. In the test, over 40,000 sale order and quote history lookup Web services were called per hour. During the test, ServiceNet CPU utilization was under 1%.

5 References

SAP Co-Innovation Lab: https://www.sdn.sap.com/irj/sdn/coil
Sonoa Systems: http://www.sonoasystems.com