iTKO LISA™

SAP CO-INNOVATION LAB

Validation and Virtualization of SAP Enterprise SOA Applications with iTKO LISA

iTKO:

Rajeev Gupta

Jason English

SAP Labs:

Joerg Nalik

An SAP Co-Innovation Lab Project

April 2009	1.0	

http://coil.sap.com

Table of Contents

1 Introduction

CIOs and IT experts are facing ever increasing pressure from the business to extend their services and make them available everywhere around the world. Cost pressures are forcing consolidation of data centers and integration of business applications. They need to pay attention to new compliance rules and security regulations. IT now has to support the *extended enterprise* (Fig 1), which no longer ends at the organizational boundaries of an enterprise.

Company employees, business partners and customers are all end-users of IT and demand instant, secure access to business applications from wherever they are. In addition, web service style message traffic in between application components is very much on the rise due to the need to facilitate inter- and intra company business application integration.

SAP addresses modern business needs through enterprise service-oriented architecture (SOA) built solutions. The business requirement for global use of SAP business applications adds new requirements to global IT deployments and operations compared to traditional 1990's ERP solutions, which were deployed and used mostly within the same local location and consisted out of only one central component.

The challenge is how to ensure the integrity and quality of these global enterprise SOA deployments. Functional and stress testing of application components is still required in order to prepare new application deployments for productive use. However, maintaining redundant copies of an application component productive landscape for testing purposes is difficult and costly.

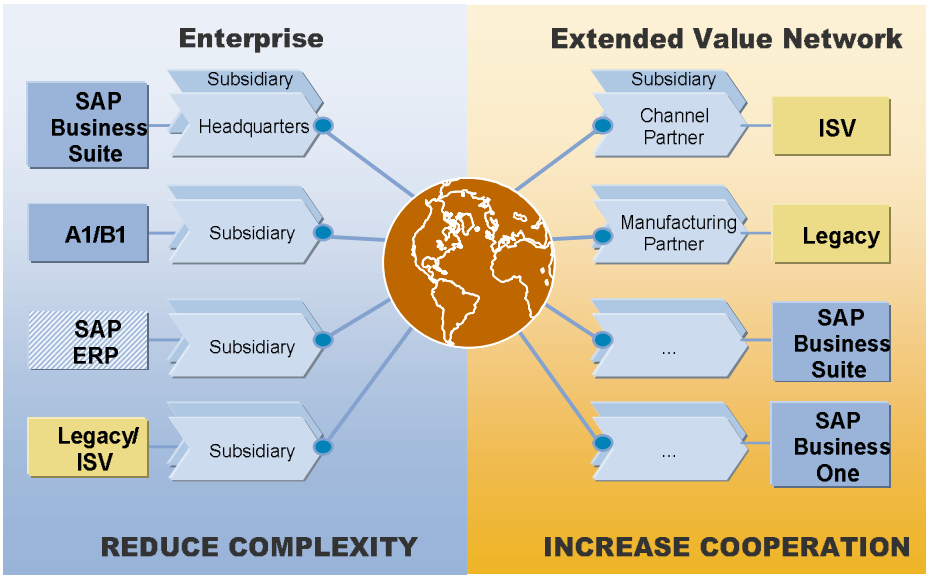

Figure 1: The Extended Enterprise

An alternative could be to simulate most components of an existing landscape and test new software components within such simulated landscapes before deploying it productively. This idea is followed by iTKO LISA. This paper describes results of a proof of concept project, which iTKO and SAP conducted jointly in SAP's Co-Innovation Lab.

2 Elements of a Productive Enterprise SOA Landscape

2.1 Typical Enterprise SOA architecture

In order to understand the testing requirements of SAP business solutions based on enterprise SOA, it helps to see in some detail how an enterprise SOA IT landscape develops out of business requirements. The typical enterprise SOA elements might be characterized the following way:

- Using integration components as part of the application platform (Fig. 2a). Examples of integration components are the SAP NetWeaver Portal component, for giving end users unified access to all business applications they have to use for their work, the SAP NetWeaver Business Intelligence (SAP NetWeaver BI) component, for consolidated reports on all business transactions, and data from the SAP NetWeaver Master Data Management (SAP NetWeaver MDM) component, as one source of truth for master data records, which are needed by multiple application components.
- Using composite application components to quickly build, deploy and use new business scenarios, which leverage access to multiple backend application systems. (Fig. 2b)
- Use of standard web services for communication of different application components with each other (Fig. 2b). Available Web Services for building composite applications are listed in Enterprise Services Repository.

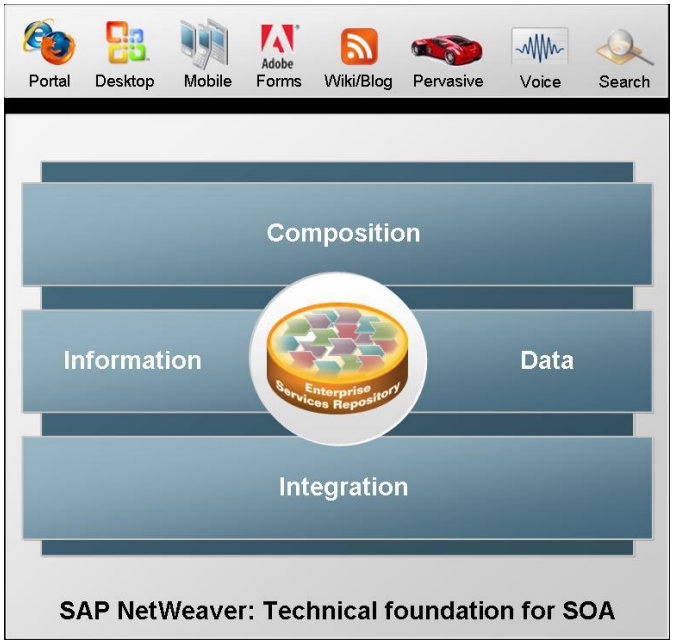

Figure 2: The SAP NetWeaver platform technology stack contains individual integration, information and composition components

A good example of a combination of these enterprise SOA elements is provided through the SAP Discovery System appliance. Conveniently this demonstration system comes as a one server

deployment offering. However, a typical productive enterprise SOA landscape would have components deployed on multiple hardware servers for added high availability and throughput scalability. Even more, in order to support business scenarios of the "extended Enterprise", which includes suppliers, customers, 3rd party service providers like credit checking and more, and to leverage hosting and outsourcing services, the application components themselves, as service provider endpoints, can reside completely outside an enterprise's internal network. This poses another particular testing challenge because outside SOA interfaces might just not have testing interfaces and can't be duplicated for QA and stress testing.

3 iTKO's LISA SOA Testing, Validation & Virtualization Platform

Services-based approaches to enterprise IT such as SAP NetWeaver have significantly evolved and extended yesterday's client-server apps. SOA allows distributed teams to assemble a variety of components and systems into enterprise applications and business processes with a higher degree of flexibility. However, with this increase in agility and business value of IT comes greater complexity and additional points of potential failure introduced across the software lifecycle. It is important to address this increased complexity with proper tools and methodologies in order to keep SOA costs low and implementation projects on track and on budget.

Today's distributed IT environment requires an approach that accounts for many moving parts. SOA architectures call for best practices that answer the "4 C's" of SOA Quality - Complete, Collaborative, Continuous and Constraint-Free:

- **Complete, Collaborative Test Coverage:** Traditional testing techniques are focused on the user interface, or at the code level alone. Testing should be automated across multi-tier dynamic enterprise applications, and shared across teams to support the lifecycle of component and middleware-based workflows.

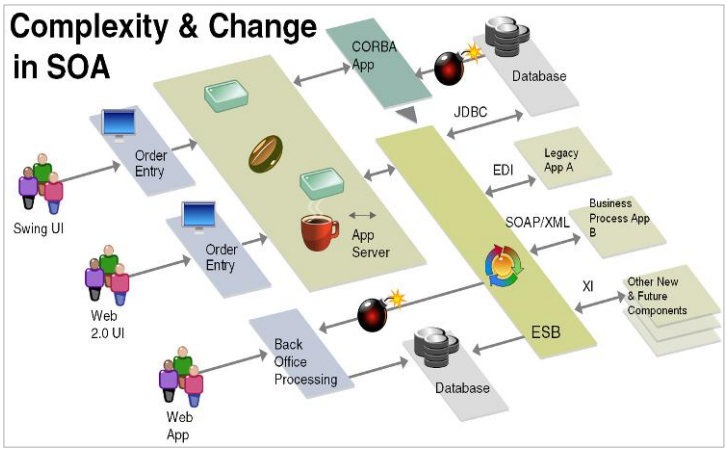

Figure 3. Complexity of typical multi-tier, heterogeneous SOA environment

- **Continuous Validation to Eliminate Unintended Consequences:** Individual components and technologies are each being changed by different stakeholders at their own pace, and the interactions between dependent systems can create unexpected problems that must be proactively traced and resolved.

- **Reducing Constraints on Key Systems:** Access limitations, transaction costs, security and capacity constraints with key enterprise systems can greatly limit the agility of multiple interdependent development and test teams and business partners throughout the software lifecycle. Needed assets must be accessible and available for use, as critical live systems are often unavailable for testing.

It follows that as complexity and interdependency increases in the SOA environment, and delivery dates and resource costs are held constant, quality can suffer. Because of the above challenges, companies usually seek to shield themselves from change by slowing down release cycles in order to maintain quality. This limits the company's agility and market competitiveness.

3.1 The iTKO LISA Suite

iTKO's LISA solution tests, validates and virtualizes heterogeneous, service-based applications, as they are supported by leading platforms such as SAP NetWeaver.

3.1.1 Key Components of the LISA Suite

The LISA Suite provides an automated testing, validation and virtualization solution for SAP NetWeaver environments and underlying integration layers.

- **Heterogeneous SOA Testing.** Test business context and workflows across the application lifecycle and technology stack. Trigger business processes in by directly invoking exposed interfaces using any of the built-in adapters: SOAP, HTTP, File, JDBC, JMS or PI. Place and intercept messages, and implement real business workflows.

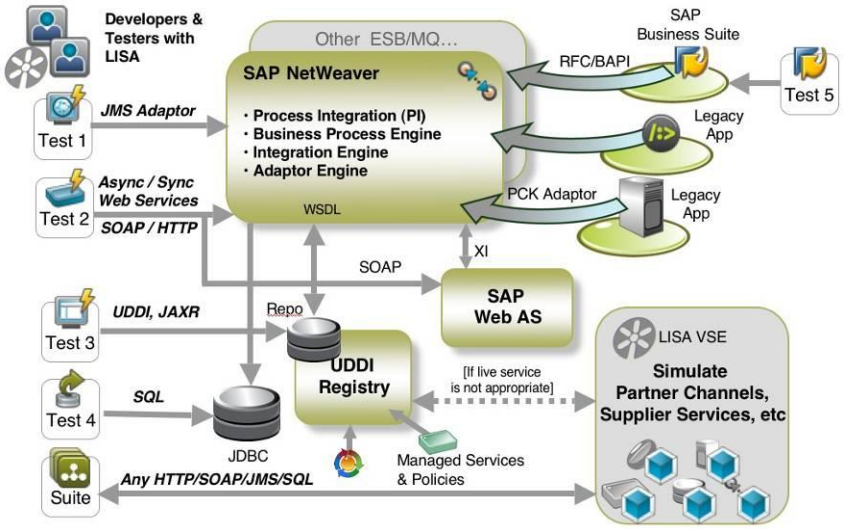

Figure 43. Example of iTKO LISA for SAP Solution

- **Cross-component Business Process Validation.** Validate business process workflows and allow quality efforts to span across business systems. LISA test cases can follow a complex workflow and validate multiple web sites, web services, Java objects and application servers. LISA supports active sessions, SSL, authentication and user status, so teams can test systems in context of the transaction workflow.
- **Synchronous and Asynchronous processes.** LISA drives transactions to act as both producer and consumer for synchronous and asynchronous services. *LISA Virtualize* provides simulation and emulation capabilities which eliminate the need to have a complete partner environment in any B2B communication.
- **Continuous Validation.** LISA orchestrates test suites against all layers of SOA applications, so as heterogeneous Services and their implementation layers evolve, continuous validation ensures that the application will meet business requirements and service level agreements over time.
- **LISA Virtualize.** The Virtual Service Environment in LISA allows teams to capture, emulate, and simulate the behavior of Services and their underlying mainframes, data and systems of record, eliminating the expense and effort of attempting to maintain a realistic environment to work against. This feature of LISA can reduce costs and save time in the lifecycle by virtualizing away constraints, and allowing multiple teams to work in parallel using any leading test tool.

- **Extensibility API.** Since most enterprises have a mix of proprietary and custom software components, iTKO's LISA Extension SDK lets developers "test enable" software and make their code testable with the LISA API, getting metrics and debugging information from within custom components.

4 SOA Testing with iTKO LISA in the SAP Co-Innovation Lab

4.1 SAP COIL Original Landscape

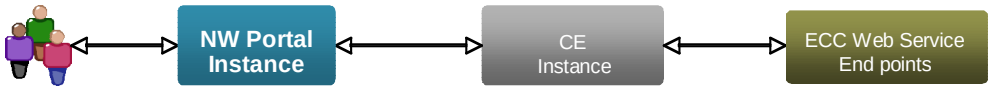

Figure 5: SAP COIL Landscape

The diagram above shows the high-level system landscape that was used for different business scenarios inside the SAP Co-Innovation Lab. User load was generated using SAP LoadRunner by HP into the NW Portal, which in turn made calls into the Composite Environment (CE) component. The web services on the backend were standard SAP ECC 6.0 examples filled with random data for simplicity of testing and respond to the incoming requests. Following two business scenarios were used:

- Customer Fact Sheet (CFS)
- Technical Document Management (TDM)

These test scenarios are focused on improving application to application (A2A) communication via web service calls. In Customer Fact Sheet (CFS) scenario, a sales person looks up his/her purchase order (PO) history with a particular customer. A list of all past purchase orders is returned. This list could have different numbers of PO listed, which is reflected in different data volume sizes of web service call responses. The XML in such responses could be as large as 3MB in tests for 1000 POs in the list. In Technical Document Management (TDM) scenario, the web service returns with a 1MB embedded PDF document.

For manual testing a browser could be opened from the performance test tool server's desktop. From there requests are directed to the NetWeaver Portal (NP), which sends a subsequent web service request to the composite application, which, in turn, triggers the composite application to send further web service calls to the emulated SAP ERP system. Responses flow respectively through all these components back to the browser.

4.2 Test, Validation and Virtualization using LISA

4.2.1 LISA acting as Customer Fact Sheet (CFS) WS provider

Figure 6. LISA Simulating as CFS WS provider

In this test scenario, the backend logic was designed such that identical web service requests delivered different amounts of data back from call to call. This reflects a typical production scenario where business object sizes, example a list of past POs of a certain customer, change over time. Therefore, LISA Virtualize sampled all possible 50 responses in this scenario and played them back in sequence exactly like SAP test setup does. For that LISA had to maintain the state (or the context of the "conversation" between components for a particular customer) in the virtual services model (VSM) in form of a counter value for enumerating the different result playback. The scenario also dealt with the less common WSIL standard, and LISA was extended to directly invoke WSIL actions as a native step. This example demonstrated that state full behavior can be easily implemented with LISA Virtualize.

In real life, a WS call with one set of parameter values can produce very different result sets (e.g., the list of customers of a sales rep can change over time). LISA was able to serve dynamic responses out of a response pool set based on a predefined logic, rather then sending just the same hard-wired response time and again. LISA's capabilities to send varying responses based on identical requests helps testers reflect state changes over time in a simulated component.

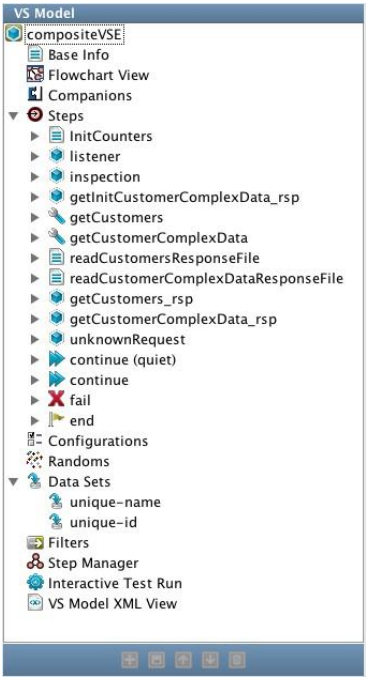

Figure 7. CFS Virtual Service Model

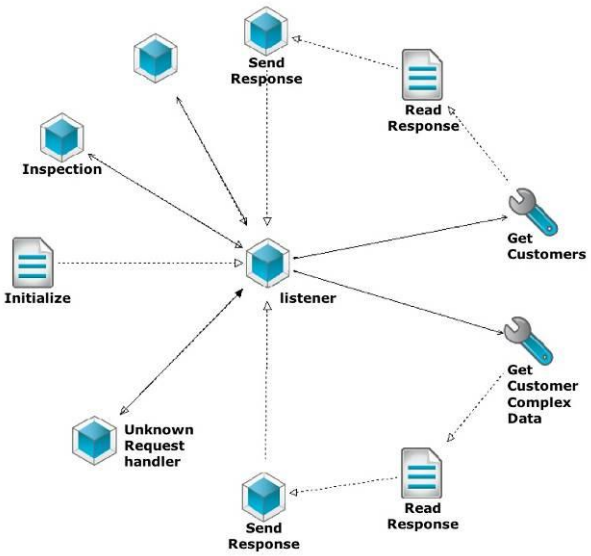

Figure 8. CFS Service Execution Steps

This scenario was recorded in original SAP landscape and the steps to create the virtual services model include:

- Recording the traffic between the consumer and the CFS service
- Creating a virtual service model and service image from the captured traffic
- Deploying the virtual model inside virtual service environment (VSE)
- Shutting down the CE instance (optional) to make sure that the data is served from the virtualized service
- Executing the CFS test scenario against the virtual service

4.2.2 LISA acting as TDM (Technical Document Management) WS provider

In this user scenario, LISA Virtualize implemented the TDM web service instead of capturing the traffic and conversations over the wire. LISA comes bundled with a container, which is used to host server side implementations based on the WDSL and WSIL files. LISA provides the ability to implement web methods defined in the WSDL, rather than responding with mere mock

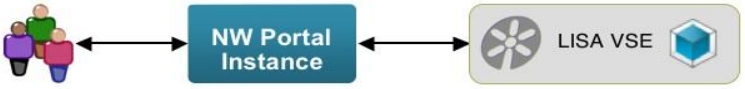

Figure 9. LISA Simulating as CFS WS provider

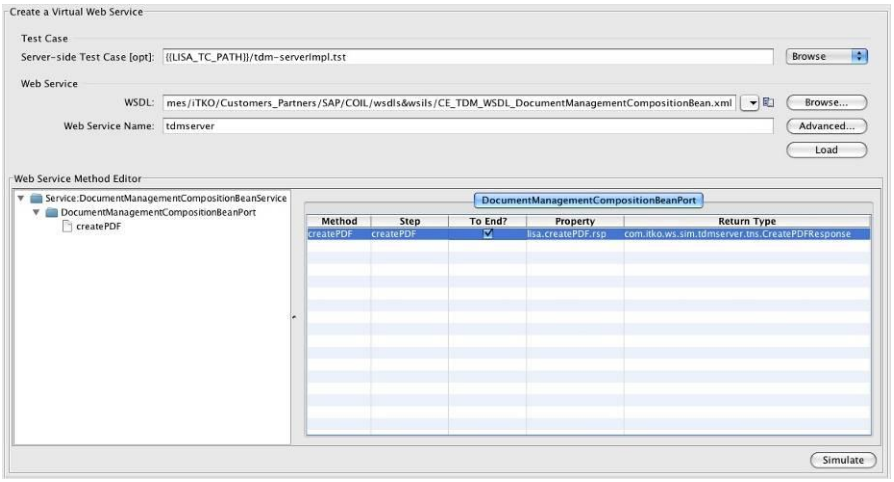

Figure 10. Simulating TDM Web Service

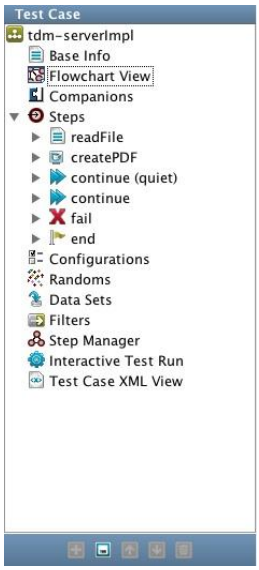

Figure 11. TDM Server Side Test

responses. LISA does that by using a server-side test case concept and mapping the methods defined in the web service with steps inside the server side test case.

This mapping allowed LISA to take control when the web service call is invoked from the consumer side. Inside the server side test case, LISA allowed the use of all different technology step types to help implement the business logic. LISA's capability to handle binary objects (like PDF files) made it even easier to implement and provide true implementation of "createPDF" method defined by the TDM service.

4.2.3 Landscape with iTKO service consumer

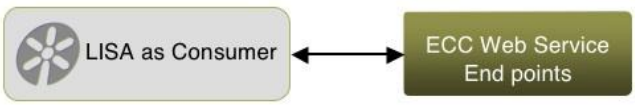

Figure 12. LISA as CFS WS Consumer

In this scenario, iTKO LISA acted as CFS web service consumer. This scenario was also recorded in original SAP landscape. The goal of this test scenario was to bypass NW portal and generate user load directly onto the backend.

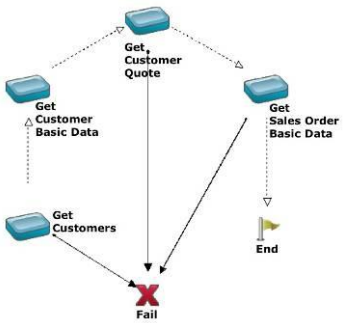

Figure 13. Test Workflow

In this case, LISA tested CFS web service exposed by ECC backend for its functional behavior and performance characteristics.

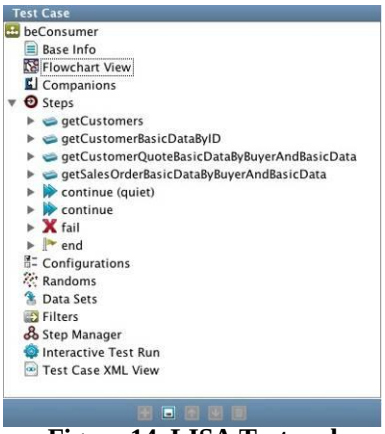

Figure 14. LISA Test and Validate

One of the main challenges of test automation is to keep the test cases in sync with the requirements. Modular testing concepts makes it possible to reuse parts of the test assets when things change. For example, concepts like Data driven testing allows test logic to be separated from test data, which enables reuse of test logic and therefore lowers maintenance. LISA has the ability to read complex objects directly from spreadsheets using a feature Data Transfer Objects (DTO). Use of DTOs simplified test cases and takes away the pain of understanding XML structures and creating complex objects with arrays. This feature allowed us to separate the test data from the test logic and hence made it possible to run the same test logic through different data scenarios.

LISA's modular approach also made it possible to leverage the same functional test asset for load and performance testing of the web service.

4.2.4 Landscape with iTKO replacing CE end-to-end

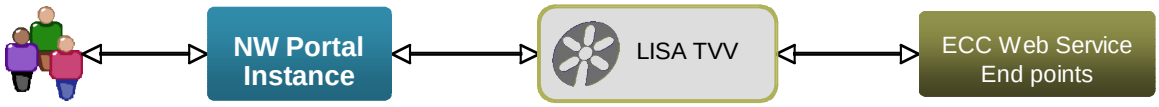

Figure 15. LISA Test, Validate and Virtualize (TVV)

In this scenario, iTKO replaced just the composite environment (CE) and simultaneously acted as both the consumer and the provider (or acts simultaneously as server and as client to use other familiar terms). LISA is fully capable of emulating a middle tier component a both ends of the transaction. Such emulation requires processing results received as consumer (from the ECC backed) for sending them out (to NW portal) as a provider.

Figure 16. LISA CE end-to-end model

As a consumer, LISA invoked CFS web services into the ECC backend and validated the response for correctness. As a producer, LISA intercepted the calls from NW portal and responded them with the information coming from the ECC backend. Not only was the XML format of the web service calls into the composite environment different from that of ECC web services, but also a different set of business logic was executed with three calls to the backend system. LISA Virtualize emulated this response logic from the Virtual Service and automatically varied the data in the correct

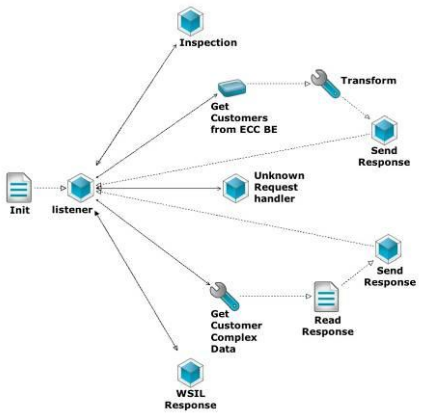

Figure 17. LISA TVV Model

order, based on different expected web service responses. LISA's support for XSL allowed data to be transformed on-the-fly, thereby making it possible to replace the CE middle tier from the SAP COIL landscape.

4.3 Results

4.3.1 iTKO LISA Virtualize acting as the service provider

The graph below compares response times of a LISA Virtual Service "standing in" as the SAP-CE Service Provider in the Customer Fact Sheet test case. LISA Virtualize is rated to run and respond at thousands of Transactions Per Second (TPS)* so it can support the tolerance levels needed for validating business workflows. However, for a specific testing scenarios, it can be configured at runtime to respond faster, slower, or exactly like the real service, so you can run "what if" scenarios for varying traffic and load levels that may affect the system under test. Without "applying the brakes" to transaction volume, iTKO LISA VS was able to respond at about half response time of the original SAP-CE component (figure 18).

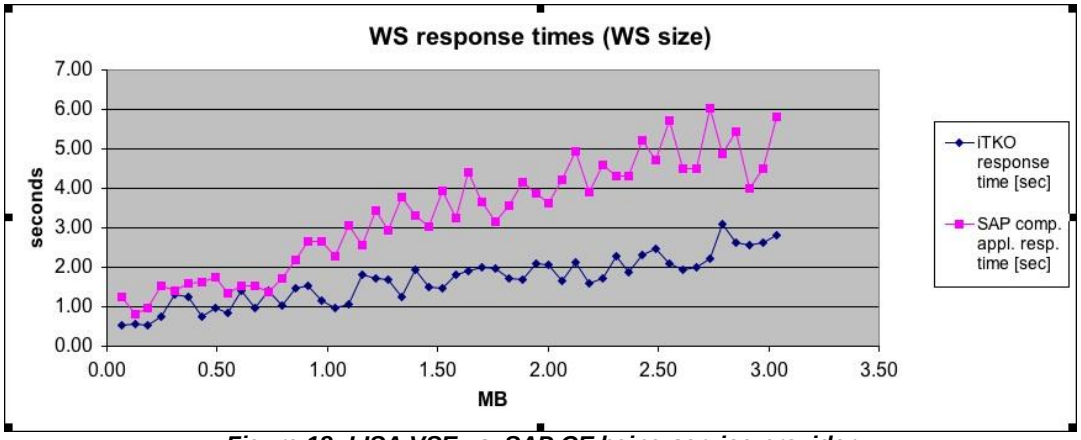

Figure 18: LISA VSE vs. SAP CE being service provider

** Usage statistics of 5000+ TPS measured from a standard 1.8GHz, dual-core PC or server, reported from iTKO's field statistics. Maximum TPS may decrease as the complexity of the simulated web service or instruction set is increased.*

4.3.2 iTKO LISA as the service consumer

Shown here is the workflow test executing as a load and performance test. Metrics are collected during a running load test, which LISA server can stage with thousands of users. In COIL, the consumer workflow test case was run with only 20 virtual users. Metric data can be reported on a system-wide or "process step" basis, that comprise many transactions, or at a high degree of detail for specific machine-to-machine transaction and response times. These re-

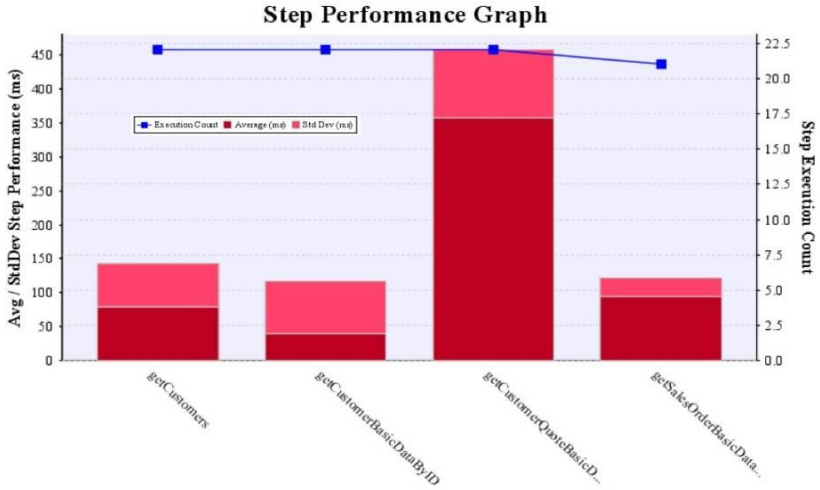

Figure 19. Step performance graph of CFS WS

ports can be viewed within the LISA interface, output in the format of choice (for viewing in other metrics and Business Intelligence tools), or fed into the customer's Test Management or software development process dashboards.

5 Summary

Supporting SAP NetWeaver customers with LISA Testing, Validation & Virtualization

Today's services-based architectures call for a deeper testing and virtualization discipline and toolsets.

When integration and SOA enablement projects are conducted without appropriate quality practices in place, every new element added to the workflow becomes a potential point of failure. Ensuring responsible shared testing practices within development and integration lifecycles as early as possible enables you to realize the ROI you should expect from adopting SOA. iTKO LISA software helps companies reduce the risk of migrating to SOA environments, while increasing trust and reuse benefits.

iTKO has deep experience helping SAP customers deliver SAP SOA based business solutions with confidence. LISA test cases are built to survive the changes and data variability inherent in today's dynamic applications, letting you continually reuse tests to validate business functionality, even during migrations to SAP landscapes from other platforms, or upgrades between software editions. The more service-oriented and agile your SAP Business Suite is, the more compelling testing and virtualizing them for testing with LISA becomes.

In this paper, we described successful testing of iTKO's LISA tools for a SOA based application test landscape in SAP's Co-Innovation Lab. We found that LISA can emulate web-service provider as well as consumer components for perfectly functionally correct end-to-end scenarios, with the ability to emulate application state correctly and with excellent tool performance for stress testing.

6 References

SAP References:

Ref 1: SDN http://sdn.sap.com/

Ref 2: ESC http://esc.sap.com/

Ref 3: SAP Discovery System

https://www.sdn.sap.com/irj/sdn/soa?rid=/webcontent/uuid/7072240e-f336-2a10-a5bb-da8854b5ad47

Ref 4: SAP Co-Innovation Lab https://www.sdn.sap.com/irj/sdn/coil

iTKO References:

Ref 1: White papers http://www.itko.com/site/resources/whitepapers.jsp

Ref 2: iTKO blog: http://blog.itko.com/

About iTKO
iTKO helps customers transform the software development and testing lifecycle for greater efficiency and quality in an environment of constant change. iTKO's award-winning LISA™ product suite can dramatically lower quality assurance costs, shorten release cycles, and eliminate the critical development and testing constraints of accessibility, capacity and security that hinder agility and competitive advantage. LISA provides test, validation, and virtualization capabilities optimized for distributed, multi-tier applications that leverage SOA, BPM, integration suites, and ESBs. iTKO customers include eBay, American Airlines, Time Warner, Swiss Re, Bank of America and government agencies. For more information, visit http://www.itko.com.

Contact iTKO
iTKO LISA (Interactive TKO, Inc.)
1505 LBJ Freeway, Suite 250
Dallas, TX 75234
877-BUY-ITKO (289-4856)
Sales: sales@itko.com