SERVICE TECHNOLOGY MAGAZINE

Issue LXXIII
June 2013
www.servicetechmag.com

FUNDAMENTAL CLOUD ARCHITECTURES

BY THOMAS ERL AND AMIN NASERPOUR

SOA Maturity & Enterprise Service Bus
by Jürgen Kress, Berthold Maier, Hajo Normann, Danilo Schmiedel, Guido Schmutz, Bernd Trops, Clemens Utschig-Utschig, Torsten Winterberg

Promoting Organizational Visibility for SOA and SOA Governance Initiatives - Part I
by Manuel Rosa, André Sampaio

Service Security and Compliance in the Cloud - Part I
by Raghu Yeluri, Sudhir S. Bangalore

$9.99USD $9.69CAD €6.99
25500 75150

Contents

PUBLISHER
Arcitura Education Inc.

EDITOR
Thomas Erl

COPY EDITOR
Maria Lee

SUPERVISING PRODUCTION MANAGER
Pamela Janice Yau

COVER DESIGN
Briana Lee

WEB DESIGN
Jasper Paladino

CONTRIBUTORS
Sudhir S. Bangalore
Thomas Erl
Jürgen Kress
Berthold Maier
Amin Naserpour
Hajo Normann
Manuel Rosa
André Sampaio
Danilo Schmiedel
Guido Schmutz
Bernd Trops
Clemens Utschig-Utschig
Torsten Winterberg
Raghu Yeluri

From the Editor

When considering the adoption of cloud platforms, it is vital that organizations understand how the inclusion of cloud computing technologies and mechanisms impacts the technology architecture that will host business data and automation solutions. Different cloud architectural models emphasize different cloud characteristics and implement these characteristics to varying extents. These models can also be combined to establish architectural layers with inter-dependencies that can increase processing potential ,while inhibiting application functionality. An understanding of fundamental cloud architectures provides the clarity necessary to blueprint a cloud-based solution prior to development and deployment. Especially in environments where multiple organizations share the same underlying infrastructure, this foreknowledge is essential to mitigating risk, optimizing application performance and assessing cloud technology vendor offerings.

Thomas Erl, Series Editor and Site Editor

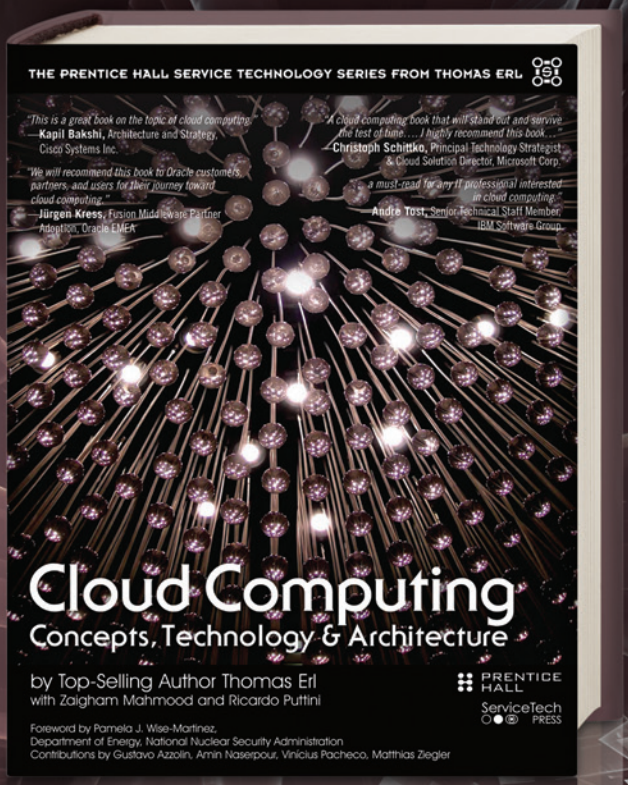
Cloud Computing
Concepts, Technology & Architecture

"This is a great book on the topic of cloud computing."
Kapil Bakshi, Architecture and Strategy,
Cisco Systems Inc.

"We will recommend this book to Oracle customers, partners, and users for their journey toward cloud computing."
Jürgen Kress, Fusion Middleware Partner Adoption,
Oracle EMEA

"A cloud computing book that will stand out and survive the test of time…. I highly recommend this book…"
Christoph Schittko, Principal Technology Strategist,
Microsoft Corp.

"… a must-read for any IT professional interested in cloud computing."
Andre Tost, Senior Technical Staff Member,
IBM Software Group



The Cloud Computing: Concepts, Technology & Architecture text book will be released on May 16, 2013. This is the ninth title in the Prentice Hall Service Technology Series from Thomas Erl. The book is authored by Thomas Erl, Zaigham Mahmood and Ricardo Puttini.

To learn more about this book, visit:
www.servicetechbooks.com/cloud

TABLE OF CONTENTS

PRENTICE HALL
ServiceTech PRESS
Arcitura
the IT education company

Fundamental Cloud Architectures

by Thomas Erl, Arcitura Education Inc. and Amin Naserpour, Technical Consultant, Hewlett-Packard

The following is an excerpt from the new book "Cloud Computing: Concepts, Technology & Architecture". For more information about this book, visit www.servicetechbooks.com/cloud.

This chapter introduces and describes several of the more common foundational cloud architectural models, each exemplifying a common usage and characteristic of contemporary cloud-based environments. The involvement and importance of different combinations of cloud computing mechanisms in relation to these architectures are explored.

11.1 Workload Distribution Architecture

IT resources can be horizontally scaled via the addition of one or more identical IT resources, and a load balancer that pro-vides runtime logic capable of evenly distributing the workload among the available IT resources (Figure 11.1). The resulting workload distribution architecture reduces both IT resource over-utilization and under-utilization to an extent dependent upon the sophistication of the load balancing algorithms and runtime logic.

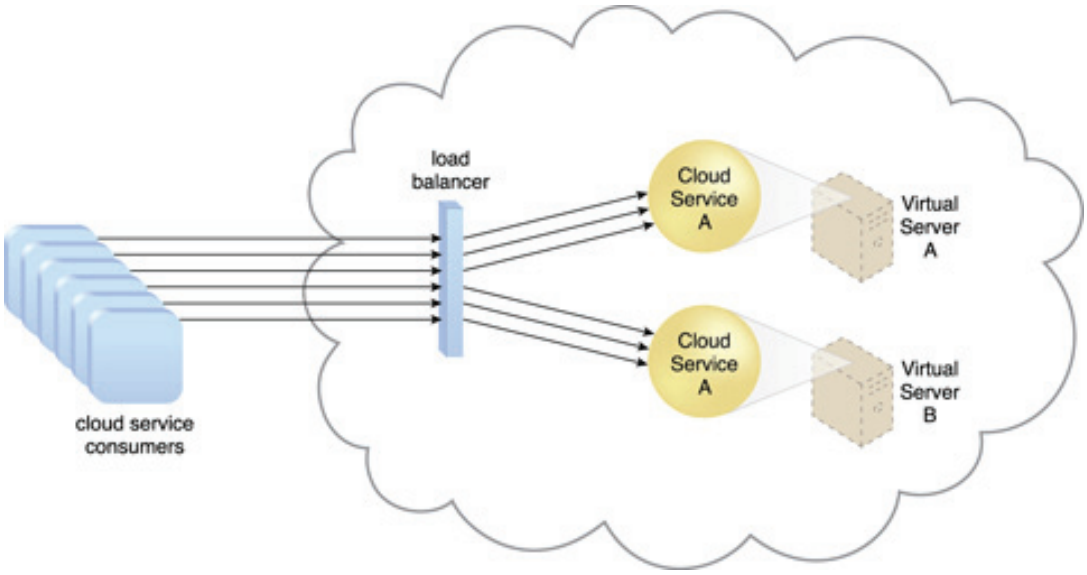


Figure 11.1 - A redundant copy of Cloud Service A is implemented on Virtual Server B. The load balancer intercepts cloud service consumer requests and directs them to both Virtual Servers A and B to ensure even workload distribution.

This fundamental architectural model can be applied to any IT resource, with workload distribution commonly carried out in support of distributed virtual servers, cloud storage devices, and cloud services. Load balancing

systems applied to specific IT resources usually produce specialized variations of this ar-chitecture that incorporate aspects of load balancing, such as:

- the service load balancing architecture explained later in this chapter
- the load balanced virtual server architecture covered in Chapter 12
- the load balanced virtual switches architecture described in Chapter 13

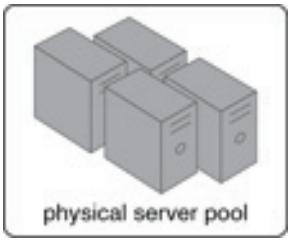
In addition to the base load balancer mechanism, and the virtual server and cloud storage device mechanisms to which load bal-ancing can be applied, the following mechanisms can also be part of this cloud architecture:

- Audit Monitor - When distributing runtime workloads, the type and geographical location of the IT resources that process the data can determine whether monitoring is necessary to fulfill legal and regulatory requirements.
- Cloud Usage Monitor - Various monitors can be involved to carry out runtime workload tracking and data processing.
- Hypervisor - Workloads between hypervisors and the virtual servers that they host may require distribution.
- Logical Network Perimeter - The logical network perimeter isolates cloud consumer network boundaries in relation to how and where workloads are distributed.
- Resource Cluster - Clustered IT resources in active/active mode are commonly used to support workload balancing between different cluster nodes.
- Resource Replication - This mechanism can generate new in-stances of virtualized IT resources in response to runtime workload distribution demands.

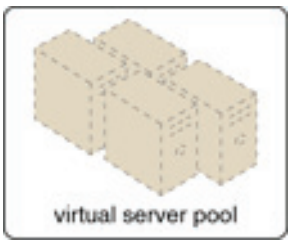
11.2 Resource Pooling Architecture

A resource pooling architecture is based on the use of one or more resource pools, in which identical IT resources are grouped and maintained by a system that automatically ensures that they remain synchronized.

Provided here are common examples of resource pools:

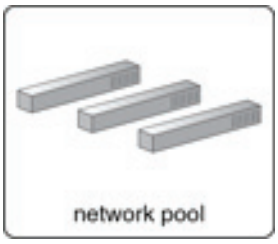


Physical server pools are composed of networked servers that have been installed with operating systems and other necessary programs and/or applications and are ready for immediate use.

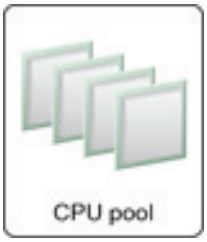


Virtual server pools are usually configured using one of several available templates chosen by the cloud consumer during pro-visioning. For example, a cloud consumer can set up a pool of mid-tier Windows servers with 4 GB of RAM or a pool of low-tier Ubuntu servers with 2 GB of RAM.

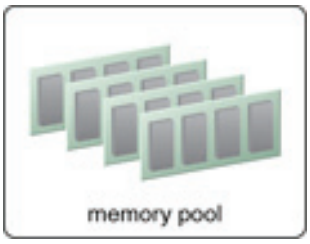
Storage pools, or cloud storage device pools, consist of file-based or block-based storage structures that contain empty and/or filled cloud storage devices.



Network pools (or interconnect pools) are composed of different preconfigured network connectivity devices. For example, a pool of virtual firewall devices or physical network switches can be created for redundant connectivity, load balancing, or link ag-gregation.



CPU pools are ready to be allocated to virtual servers, and are typically broken down into individual processing cores.



Pools of physical RAM can be used in newly provisioned physical servers or to vertically scale physical servers.

Dedicated pools can be created for each type of IT resource and individual pools can be grouped into a larger pool, in which case each individual pool becomes a sub-pool (Figure 11.2).

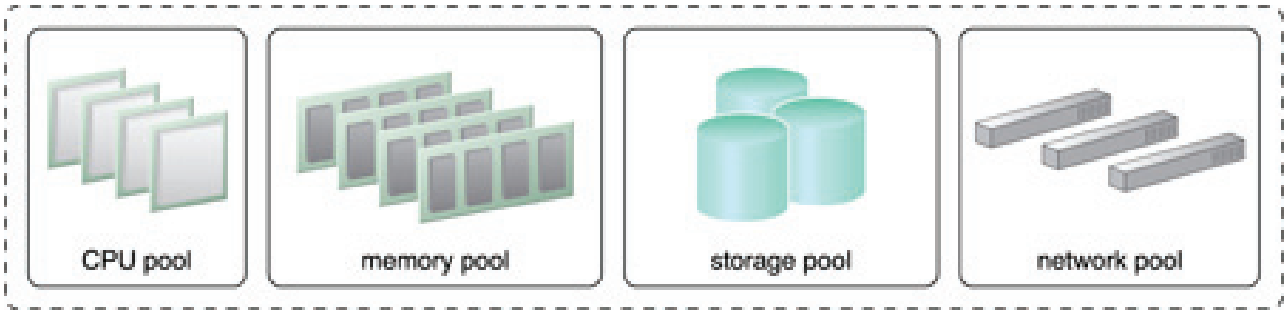


Figure 11.2 - A sample resource pool that is comprised of four sub-pools of CPUs, memory, cloud storage devices, and virtual network devices.

Resource pools can become highly complex, with multiple pools created for specific cloud consumers or

applications. A hierar-chical structure can be established to form parent, sibling, and nested pools in order to facilitate the organization of diverse resource pooling requirements (Figure 11.3).

Sibling resource pools are usually drawn from physically grouped IT resources, as opposed to IT resources that are spread out over different data centers. Sibling pools are isolated from one another so that each cloud consumer is only provided access to its respective pool.

In the nested pool model, larger pools are divided into smaller pools that individually group the same type of IT resources to-gether (Figure 11.4). Nested pools can be used to assign re-source pools to different departments or groups in the same cloud consumer organization.

After resources pools have been defined, multiple instances of IT resources from each pool can be created to provide an in-memory pool of “live” IT resources.

In addition to cloud storage devices and virtual servers, which are commonly pooled mechanisms, the following mechanisms can al-so be part of this cloud architecture:

- Audit Monitor - This mechanism monitors resource pool usage to ensure compliance with privacy and regulation require-ments, especially when pools contain cloud storage devices or data loaded into memory.

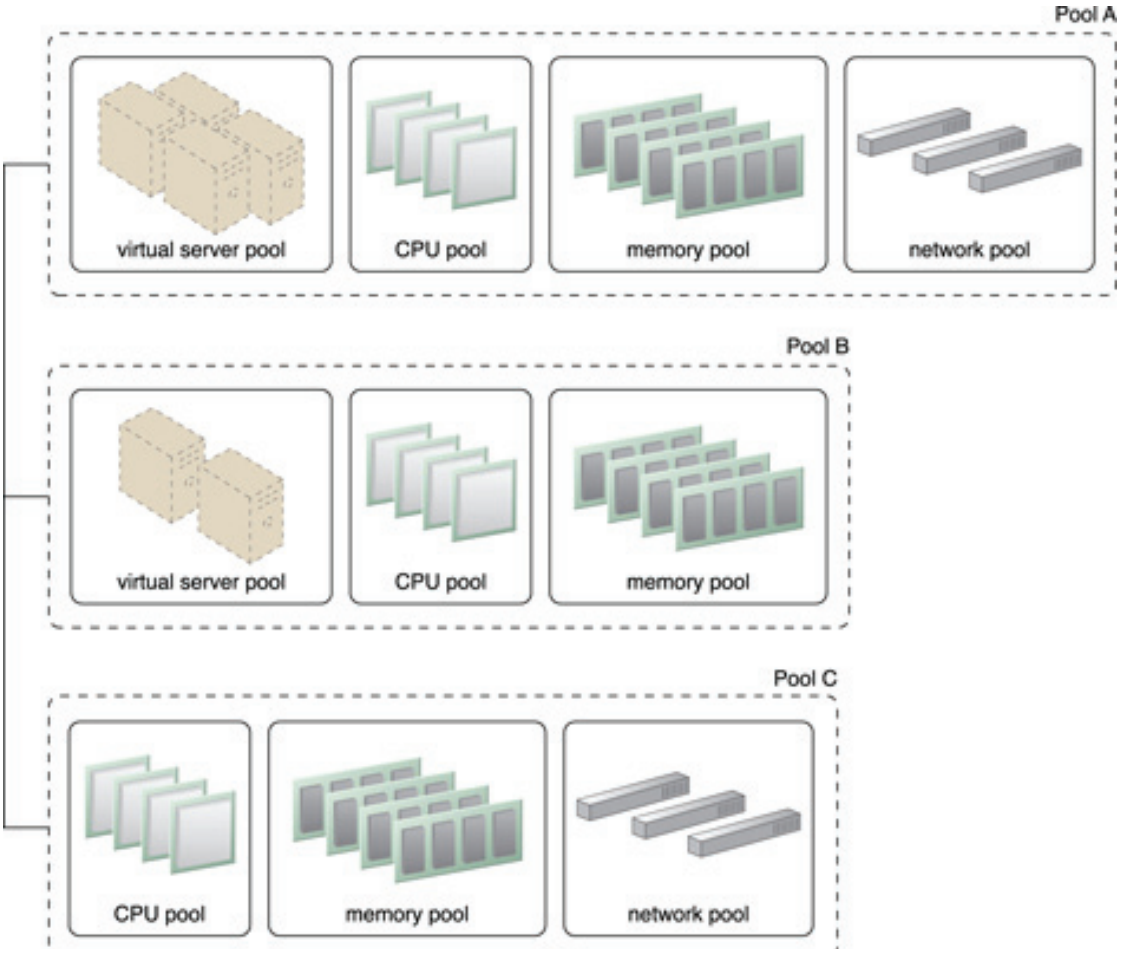


Figure 11.3 - Pools B and C are sibling pools that are taken from the larger Pool A, which has been allocated to a cloud consumer. This is an alternative to taking the IT resources for Pool B and Pool C from a general reserve of IT resources that is shared throughout the cloud.

- Cloud Usage Monitor - Various cloud usage monitors are in-volved in the runtime tracking and synchronization that are required by the pooled IT resources and any underlying man-agement systems.
- Hypervisor - The hypervisor mechanism is responsible for providing virtual servers with access to resource pools, in addition to hosting the virtual servers and sometimes the resource pools themselves.

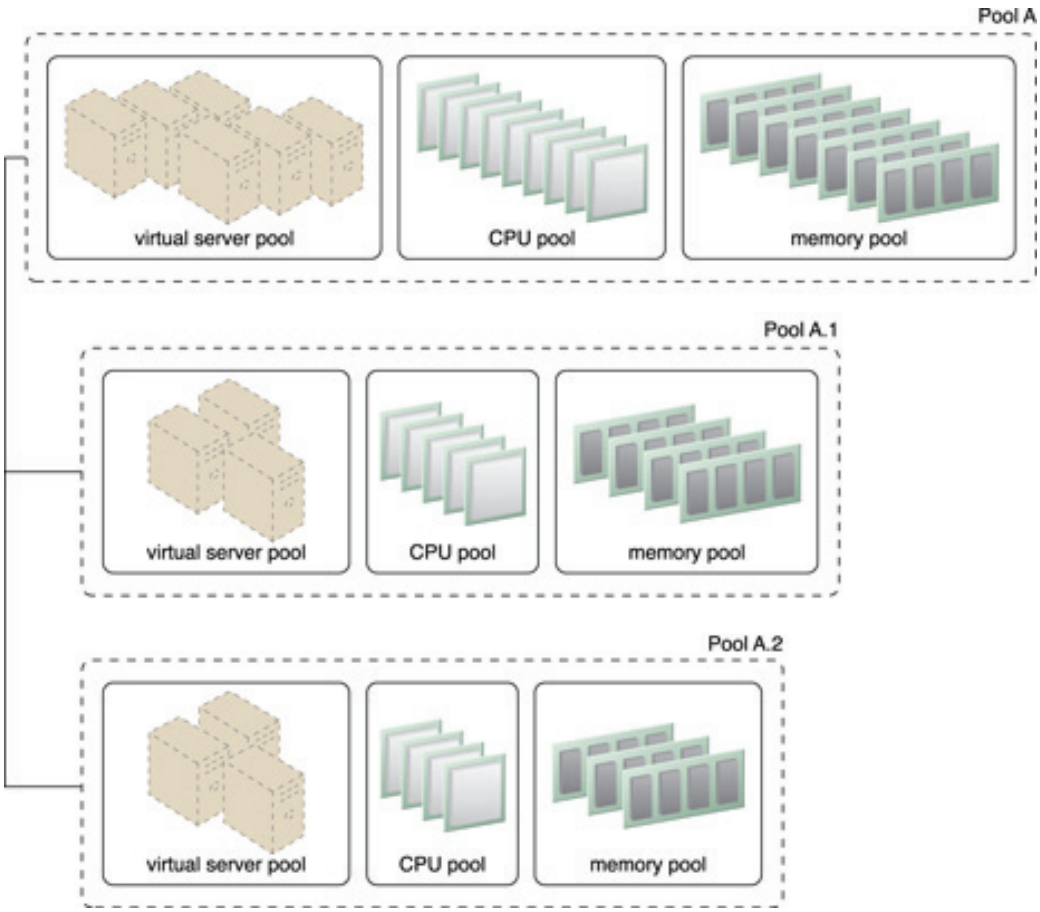


Figure 11.4 - Nested Pools A.1 and Pool A.2 are comprised of the same IT resources as Pool A, but in different quantities. Nested pools are typically used to provision cloud services that need to be rapidly instantiated using the same type of IT resources with the same configuration settings.

- Logical Network Perimeter - The logical network perimeter is used to logically organize and isolate resource pools.
- Pay-Per-Use Monitor - The pay-per-use monitor collects usage and billing information on how individual cloud consumers are allocated and use IT resources from various pools.
- Remote Administration System - This mechanism is commonly used to interface with backend systems and programs in order to provide resource pool administration features via a front-end portal.
- Resource Management System - The resource management system mechanism supplies cloud consumers with the tools and per-mission management options for administering resource pools.

- Resource Replication - This mechanism is used to generate new instances of IT resources for resource pools.

11.3 Dynamic Scalability Architecture

The dynamic scalability architecture is an architectural model based on a system of predefined scaling conditions that trigger the dynamic allocation of IT resources from resource pools. Dy-namic allocation enables variable utilization as dictated by usage demand fluctuations, since unnecessary IT resources are efficiently reclaimed without requiring manual interaction.

The automated scaling listener is configured with workload thresholds that dictate when new IT resources need to be added to the workload processing. This mechanism can be provided with logic that determines how many additional IT resources can be dynamically provided, based on the terms of a given cloud con-sumer's provisioning contract.

The following types of dynamic scaling are commonly used:

- Dynamic Horizontal Scaling - IT resource instances are scaled out and in to handle fluctuating workloads. The automatic scaling listener monitors requests and signals resource replication to initiate IT resource duplication, as per requirements and permissions.
- Dynamic Vertical Scaling - IT resource instances are scaled up and down when there is a need to adjust the processing capacity of a single IT resource. For example, a virtual server that is being overloaded can have its memory dynami-cally increased or it may have a processing core added.
- Dynamic Relocation - The IT resource is relocated to a host with more capacity. For example, a database may need to be moved from a tape-based SAN storage device with 4 GB per second I/O capacity to another disk-based SAN storage device with 8 GB per second I/O capacity.

Figures 11.5 to 11.7 illustrate the process of dynamic horizon-tal scaling.

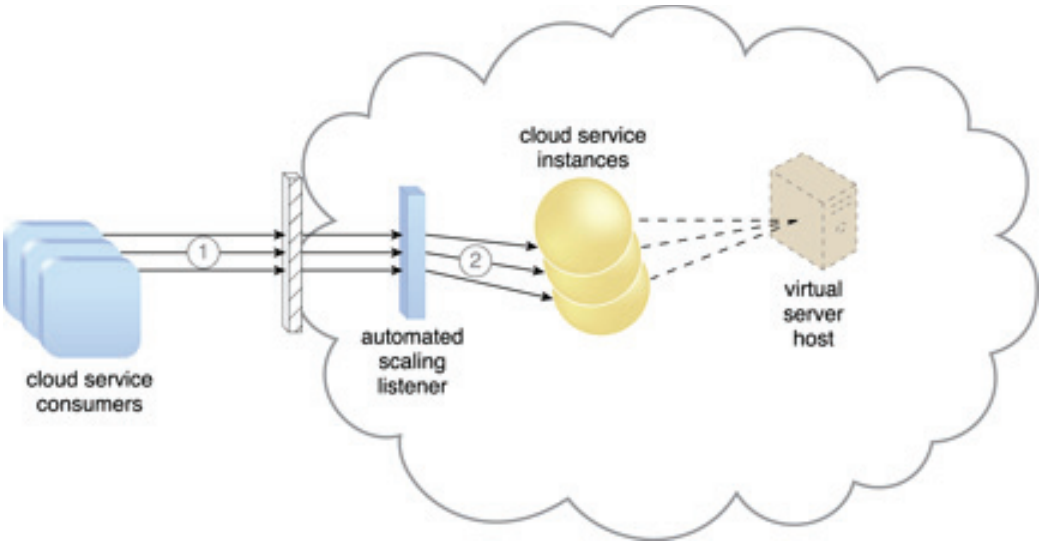


Figure 11.5 - Cloud service consumers are sending requests to a cloud service (1). The automated scaling listener monitors the cloud service to determine if predefined capacity thresholds are being exceeded (2).

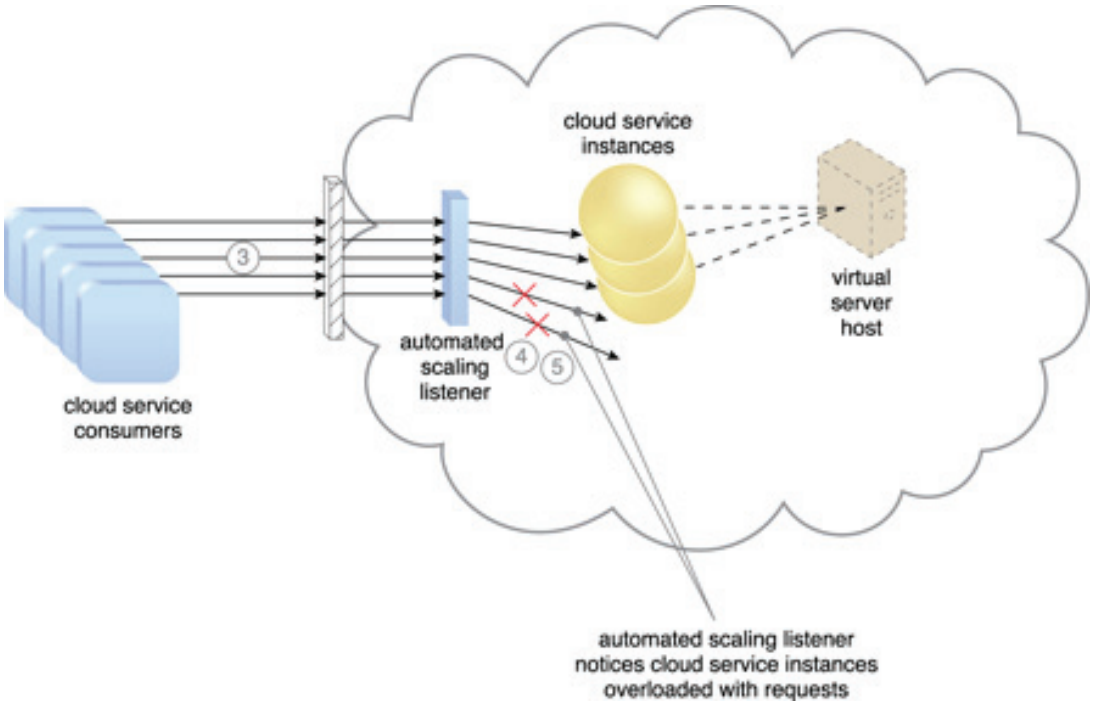


Figure 11.6 - The number of requests coming from cloud service consumers increases (3). The workload exceeds the performance thresholds. The automated scaling listener determines the next course of action based on a predefined scaling policy (4). If the cloud service implementation is deemed eligible for additional scaling, the automated scaling listener initiates the scaling process (5).

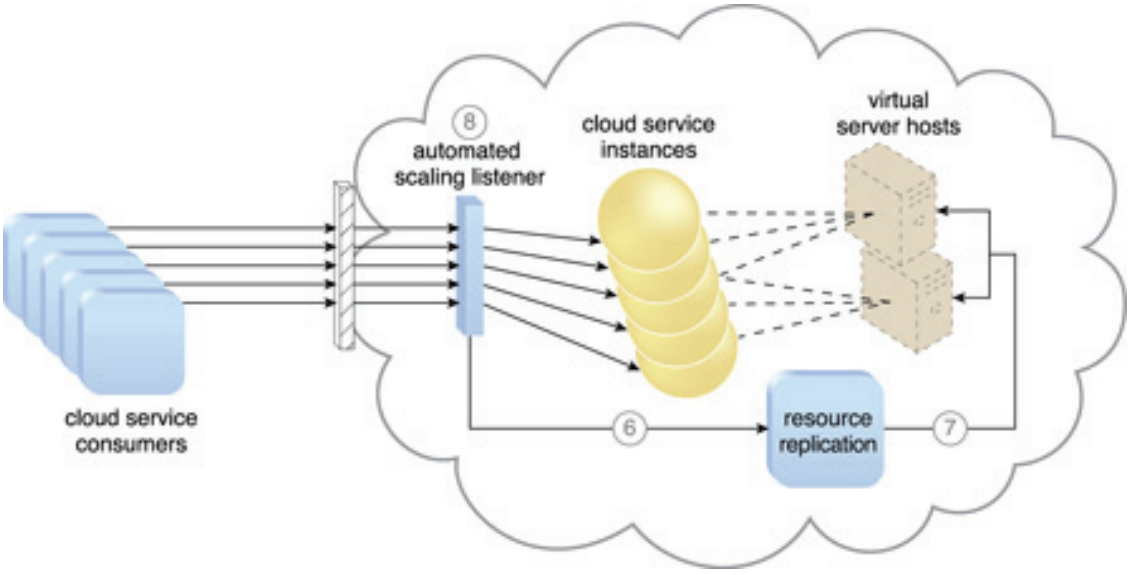


Figure 11.7 - The automated scaling listener sends a signal to the resource replication mechanism (6), which creates more instances of the cloud service (7). Now that the increased workload has been accommodated, the automated scaling listener resumes monitoring and detracting and adding IT resources, as required (8).

The dynamic scalability architecture can be applied to a range of IT resources, including virtual servers and cloud storage de-vices. Besides the core automated scaling listener and resource replication mechanisms, the following mechanisms can also be used in this form of cloud architecture:

- Cloud Usage Monitor - Specialized cloud usage monitors can track runtime usage in response to dynamic fluctuations caused by this architecture.
- Hypervisor - The hypervisor is invoked by a dynamic scala-bility system to create or remove virtual server instances, or to be scaled itself.
- Pay-Per-Use Monitor - The pay-per-use monitor is engaged to collect usage cost information in response to the scaling of IT resources.

11.4 Elastic Resource Capacity Architecture

The elastic resource capacity architecture is primarily related to the dynamic provisioning of virtual servers, using a system that allocates and reclaims CPUs and RAM in immediate response to the fluctuating processing requirements of hosted IT re-sources (Figures 11.8 and 11.9).

Resource pools are used by scaling technology that interacts with the hypervisor and/or VIM to retrieve and return CPU and RAM resources at runtime. The runtime processing of the virtual server is monitored so that additional processing power can be leveraged from the resource pool via dynamic allocation, before capacity thresholds are met. The virtual server and its hosted applications and IT resources are vertically scaled in response.

This type of cloud architecture can be designed so that the in-telligent automation engine script sends its scaling request via the VIM instead of to the hypervisor directly. Virtual servers that participate in elastic resource allocation systems may require rebooting in order for the dynamic resource allocation to take effect.

Some additional mechanisms that can be included in this cloud architecture are the following:

- Cloud Usage Monitor - Specialized cloud usage monitors col-lect resource usage information on IT resources before, dur-ing, and after scaling, to help define the future processing capacity thresholds of the virtual servers.
- Pay-Per-Use Monitor - The pay-per-use monitor is responsible for collecting resource usage cost information as it fluctu-ates with the elastic provisioning.
- Resource Replication - Resource replication is used by this architectural model to generate new instances of the scaled IT resources.

Intelligent Automation Engine

The intelligent automation engine automates administration tasks by executing scripts that contain workflow logic.

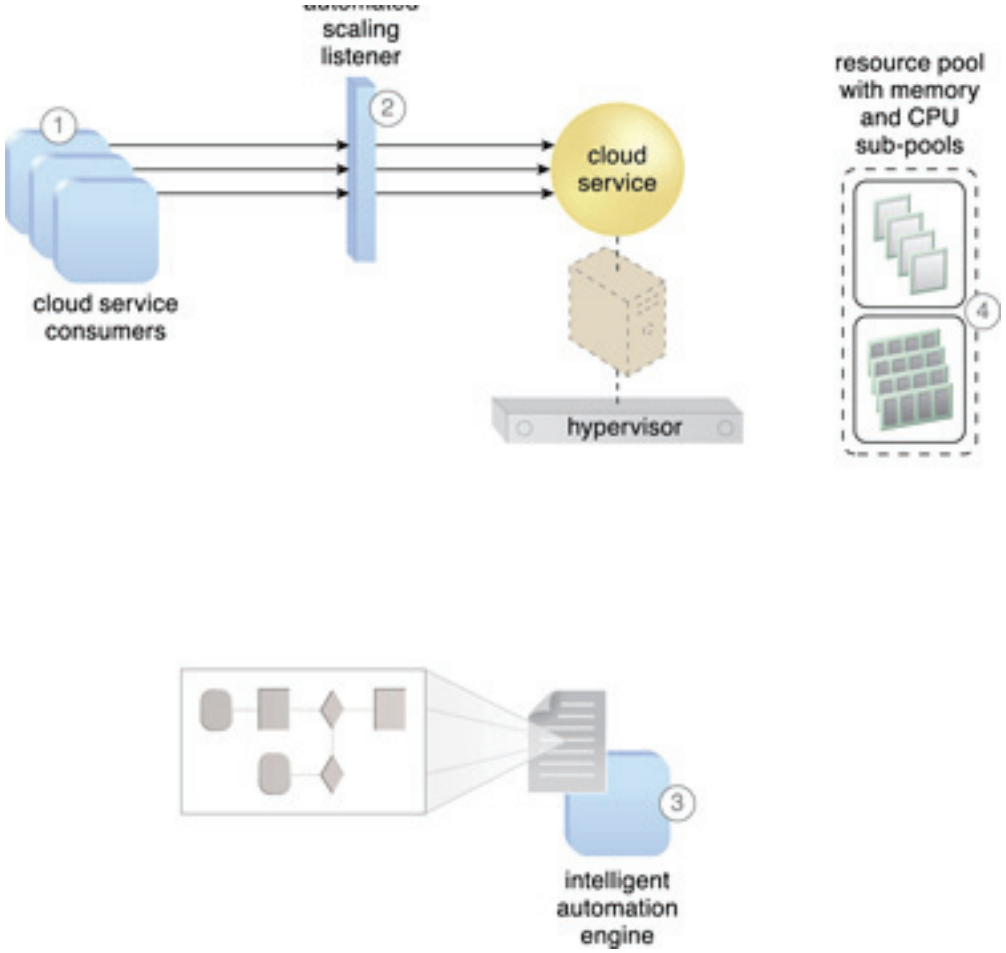


Figure 11.8 - Cloud service consumers are actively sending requests to a cloud service (1), which are monitored by an automated scaling listener (2). An intelligent automation engine script is deployed with workflow logic (3) that is capable of notifying the resource pool using allocation requests (4).

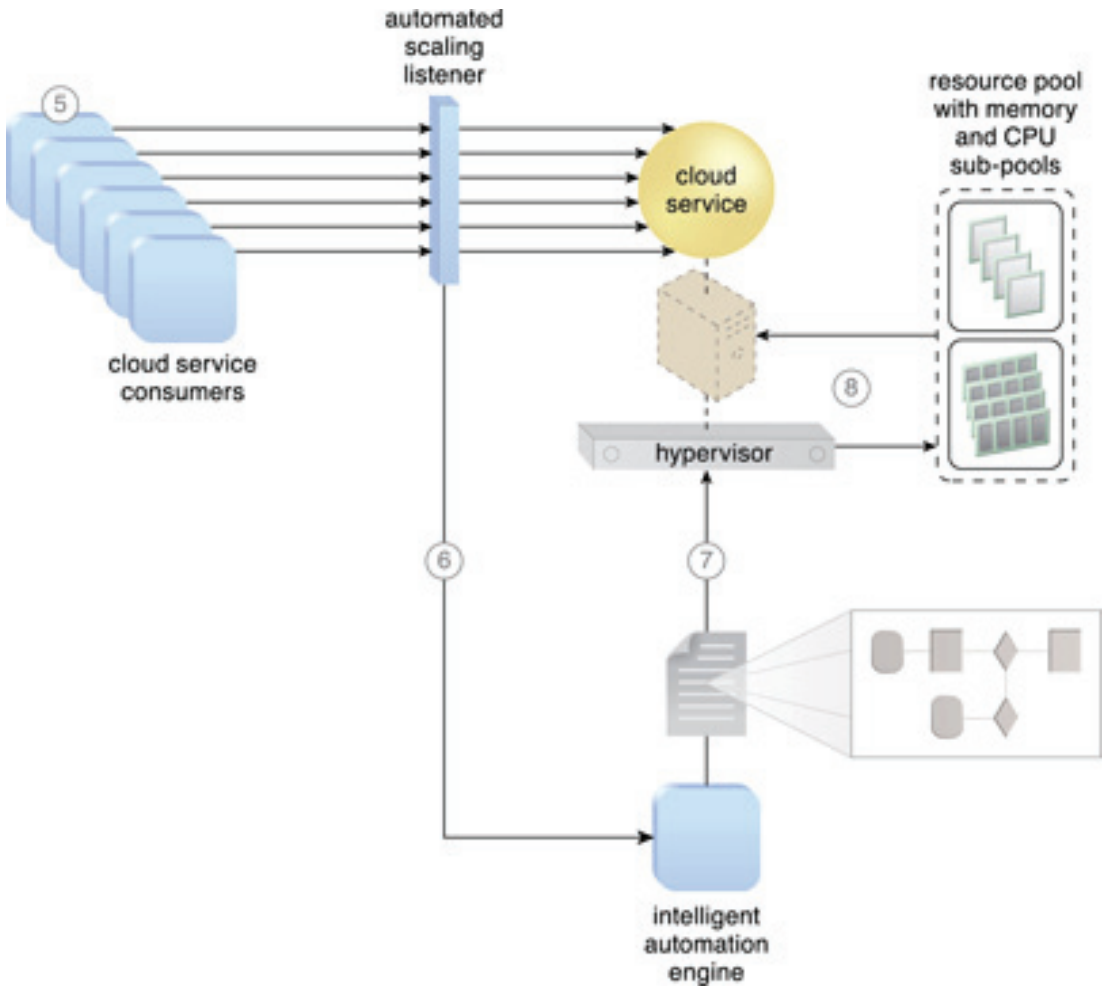


Figure 11.9 - Cloud service consumer requests increase (5), causing the automated scaling listener to signal the intelligent automation engine to execute the script (6). The script runs the workflow logic that signals the hypervisor to allocate more IT resources from the resource pools (7). The hypervisor allocates additional CPU and RAM to the virtual server, enabling the increased workload to be handled (8).

11.5 Service Load Balancing Architecture

The service load balancing architecture can be considered a specialized variation of the workload distribution architecture that is geared specifically for scaling cloud service implemen-tations. Redundant deployments of cloud services are created, with a load balancing system added to dynamically distribute workloads.

The duplicate cloud service implementations are organized into a resource pool, while the load balancer is positioned as either an external or built-in component to allow the host servers to balance the workloads themselves.

Depending on the anticipated workload and processing capacity of host server environments, multiple instances of each cloud service implementation can be generated as part of a resource pool that responds to fluctuating request volumes more effi-ciently.

The load balancer can be positioned either independent of the cloud services and their host servers (Figure 11.10), or built-in as part of the application or server's environment. In the latter case, a primary server with the load balancing logic can communicate with neighboring servers to balance the workload (Figure 11.11).

The service load balancing architecture can involve the follow-ing mechanisms in addition to the load balancer:

- Cloud Usage Monitor - Cloud usage monitors may be involved with monitoring cloud service instances and their respective IT resource consumption levels, as well as various runtime monitoring and usage data collection tasks.
- Resource Cluster - Active-active cluster groups are incorpo-rated in this architecture to help balance workloads across different members of the cluster.
- Resource Replication - The resource replication mechanism is utilized to generate cloud service implementations in support of load balancing requirements.

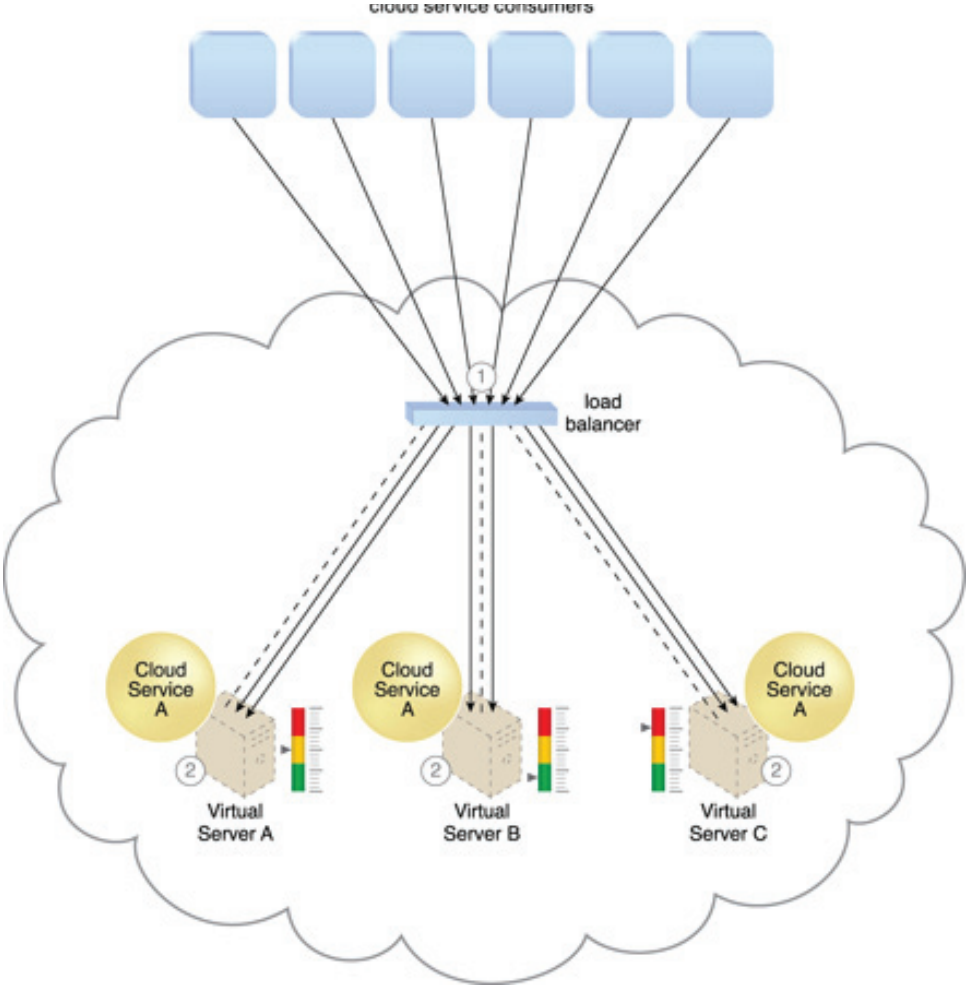


Figure 11.10 - The load balancer intercepts messages sent by cloud service consumers (1) and forwards them to the virtual servers so that the workload processing is horizontally scaled (2).

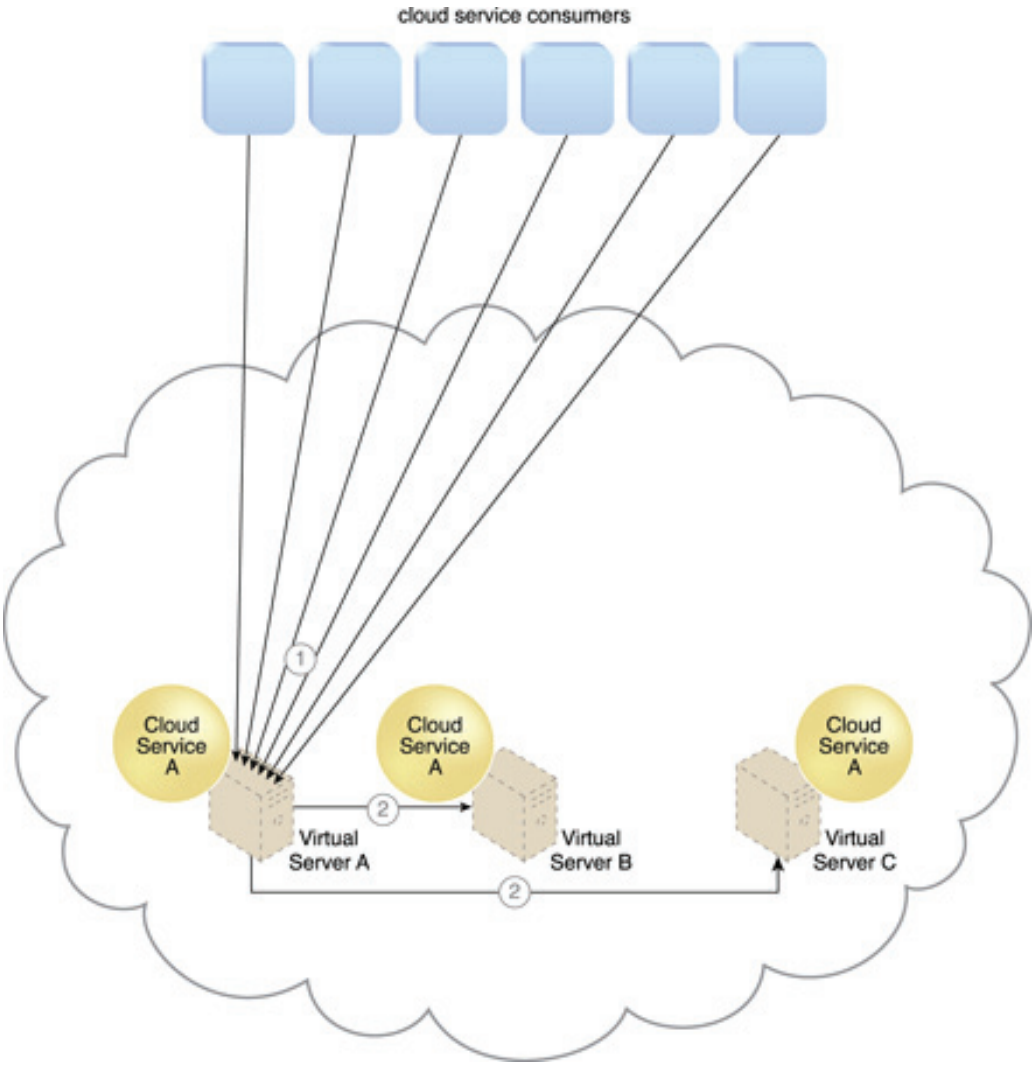


Figure 11.11 - Cloud service consumer requests are sent to Cloud Service A on Virtual Server A (1). The cloud service implementation includes built-in load balancing logic that is capable of distributing requests to the neighboring Cloud Service A implementations on Virtual Servers B and C (2).

11.6 Cloud Bursting Architecture

The cloud bursting architecture establishes a form of dynamic scaling that scales or “bursts out” on-premise IT resources into a cloud whenever predefined capacity thresholds have been reached. The corresponding cloud-based IT resources are redun-dantly pre-deployed but remain inactive until cloud bursting occurs. After they are no longer required, the cloud-based IT resources are released and the architecture “bursts in” back to the on-premise environment.

Cloud bursting is a flexible scaling architecture that provides cloud consumers with the option of using cloud-based IT re-sources only to meet higher usage demands. The foundation of this architectural model is based on the automated scaling lis-tener and resource replication mechanisms.

The automated scaling listener determines when to redirect re-quests to cloud-based IT resources, and resource replication is used to maintain synchronicity between on-premise and cloud-based IT resources in relation to state information (Figure 11.12).

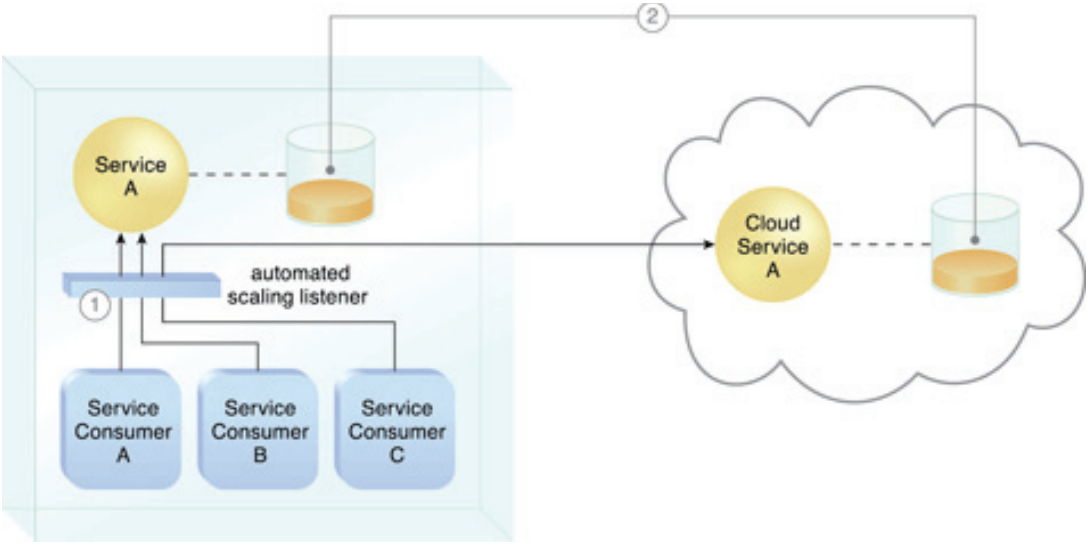


Figure 11.12 - An automated scaling listener monitors the usage of on-premise Service A, and redirects Service Consumer C's request to Service A's redundant implementation in the cloud (Cloud Service A) once Service A's usage threshold has been exceeded (1). A resource replication system is used to keep state management databases synchronized (2).

In addition to the automated scaling listener and resource rep-lication, numerous other mechanisms can be used to automate the burst in and out dynamics for this architecture, depending pri-marily on the type of IT resource being scaled.

11.7 Elastic Disk Provisioning Architecture

Cloud consumers are commonly charged for cloud-based storage space based on fixed-disk storage allocation, meaning the charg-es are predetermined by disk capacity and not aligned with ac-tual data storage consumption. Figure 11.13 demonstrates this by illustrating a scenario in which a cloud consumer provisions a virtual server with the Windows Server operating system and three 150 GB hard drives. The cloud consumer is billed for using 450 GB of storage space after installing the operating system, even though the operating system only requires 15 GB of storage space.

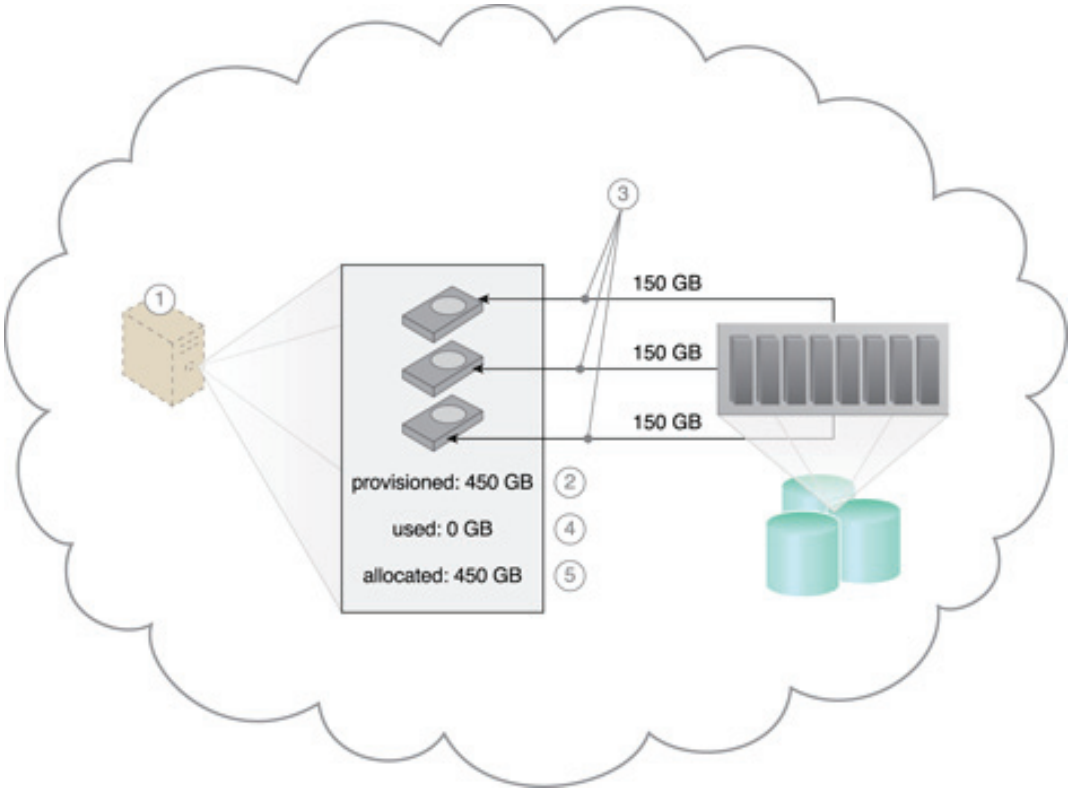


Figure 11.13 - The cloud consumer requests a virtual server with three hard disks, each with a capacity of 150 GB (1). The virtual server is provisioned according to the elastic disk provisioning architecture, with a total of 450 GB of disk space (2). The 450 GB is allocated to the virtual server by the cloud provider (3). The cloud consumer has not installed any software yet, meaning the actual used space is currently 0 GB (4). Because the 450 GB are already allocated and reserved for the cloud consumer, it will be charged for 450 GB of disk usage as of the point of allocation (5).

The elastic disk provisioning architecture establishes a dynamic storage provisioning system that ensures that the cloud consumer is granularly billed for the exact amount of storage that it actually uses. This system uses thin-provisioning technology for the dynamic allocation of storage space, and is further sup-ported by runtime usage monitoring to collect accurate usage data for billing purposes (Figure 11.14).

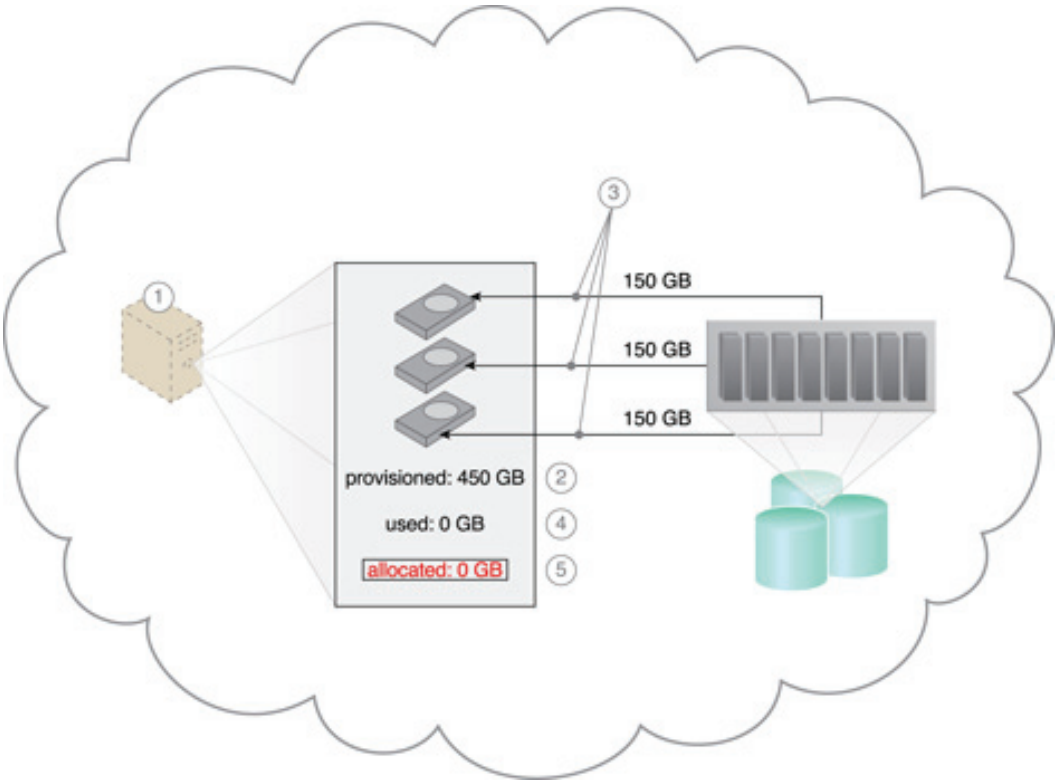


Figure 11.14 - The cloud consumer requests a virtual server with three hard disks, each with a capacity of 150 GB (1). The virtual server is provisioned by this architecture with a total of 450 GB of disk space (2). The 450 GB are set as the maximum disk usage that is allowed for this virtual server, although no physical disk space has been reserved or allocated yet (3). The cloud consumer has not installed any software, meaning the actual used space is currently at 0 GB (4). Because the allocated disk space is equal to the actual used space (which is currently at zero), the cloud consumer is not charged for any disk space usage (5).

Thin-provisioning software is installed on virtual servers that process dynamic storage allocation via the hypervisor, while the pay-per-use monitor tracks and reports granular billing-related disk usage data (Figure 11.15).

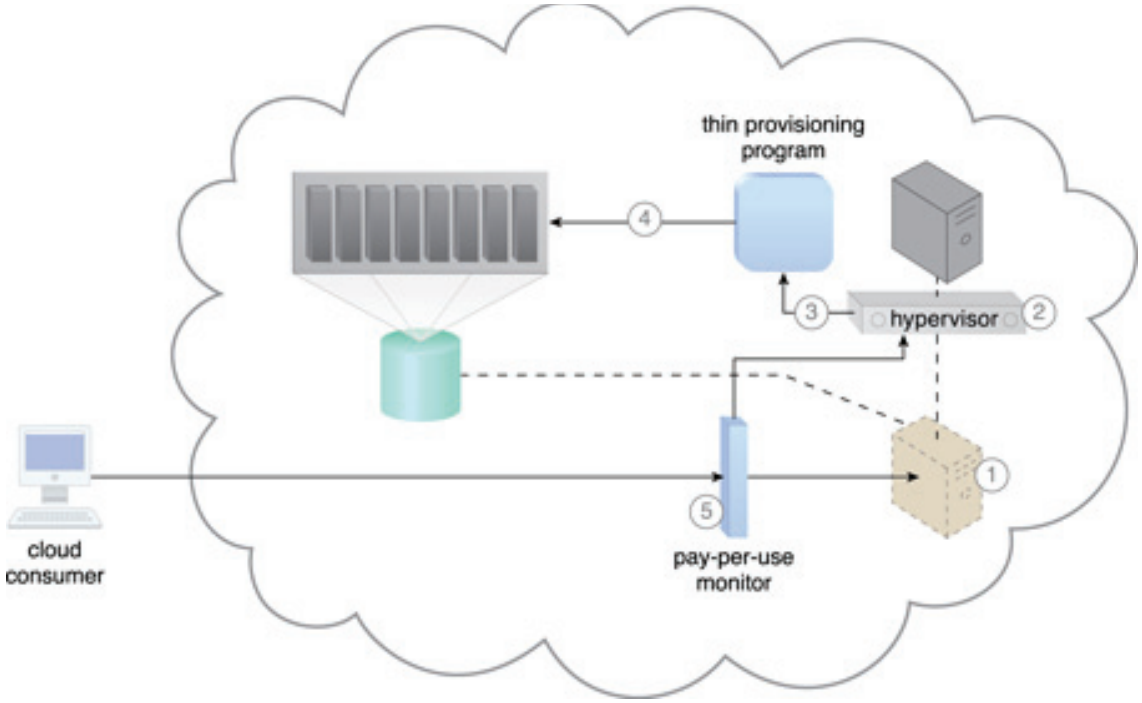


Figure 11.15 - A request is received from a cloud consumer, and the provisioning of a new virtual server instance begins (1). As part of the provisioning process, the hard disks are chosen as dynamic or thin-provisioned disks (2). The hypervisor calls a dynamic disk allocation component to create thin disks for the virtual server (3). Virtual server disks are created via the thin-provisioning program and saved in a folder of near-zero size. The size of this folder and its files grow as operating applications are installed and additional files are copied onto the virtual server (4). The pay-per-use monitor tracks the actual dynamically allocated storage for billing purposes (5).

The following mechanisms can be included in this architecture in addition to the cloud storage device, virtual server, hypervisor, and pay-per-use monitor:

- Cloud Usage Monitor - Specialized cloud usage monitors can be used to track and log storage usage fluctuations.
- Resource Replication - Resource replication is part of an elastic disk provisioning system when conversion of dynamic thin-disk storage into static thick-disk storage is required.

11.8 Redundant Storage Architecture

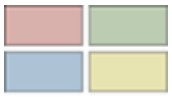
Cloud storage devices are occasionally subject to failure and disruptions that are caused by network connectivity issues, con-troller or general hardware failure, or security breaches. A compromised cloud storage device's reliability can have a ripple effect and cause impact failure across all of the services, ap-plications, and infrastructure components in the cloud that are reliant on its availability.

The storage service gateway is a component that acts as the external interface to cloud storage services, and is capable of automatically redirecting cloud consumer requests whenever the location of the requested data has changed.

The redundant storage architecture introduces a secondary du-plicate cloud storage device as part of a failover system that synchronizes its data with the data in the primary cloud storage device. A storage service gateway diverts cloud consumer requests to the secondary device whenever the primary device fails (Figures 11.16 and 11.17).

LUN

A logical unit number (LUN) is a logical drive that represents a partition of a physical drive.



LUNs

Storage Service Gateway

The storage service gateway is a component that acts as the external interface to cloud storage services, and is capable of automatically redirecting cloud consumer requests whenever the location of the requested data has changed.

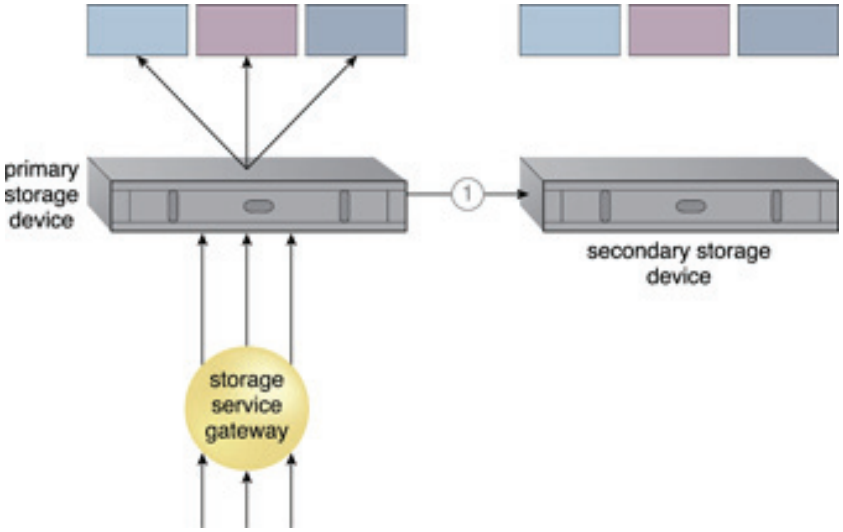


Figure 11.16 - The primary cloud storage device is routinely replicated to the secondary cloud storage device (1).

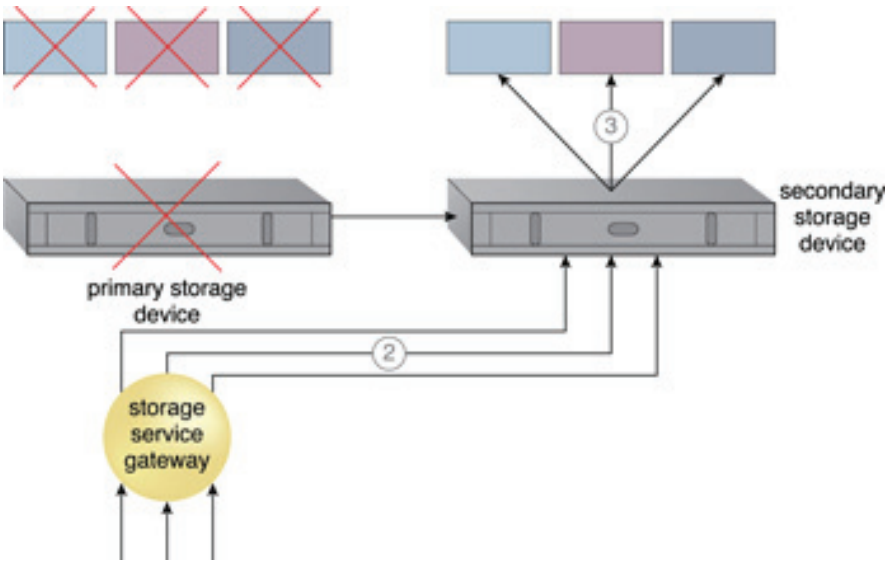


Figure 11.17 - The primary storage becomes unavailable and the storage service gateway forwards the cloud consumer requests to the secondary storage device (2). The secondary storage device forwards the requests to the LUNs, allowing cloud consumers to continue to access their data (3).

This cloud architecture primarily relies on a storage replica-tion system that keeps the primary cloud storage device syn-chronized with its duplicate secondary cloud storage devices (-Figure 11.18).

> Storage replication is a variation of the resource replication mechanisms used to synchronously or asynchronously replicate data from a primary storage device to a secondary storage device. It can be used to replicate partial and entire LUNs.

Cloud providers may locate secondary cloud storage devices in a different geographical region than the primary cloud storage device, usually for economic reasons. However, this can introduce legal concerns for some types of data. The location of the sec-ondary cloud storage devices can dictate the protocol and method used for synchronization, as some replication transport pro-tocols have distance restrictions.

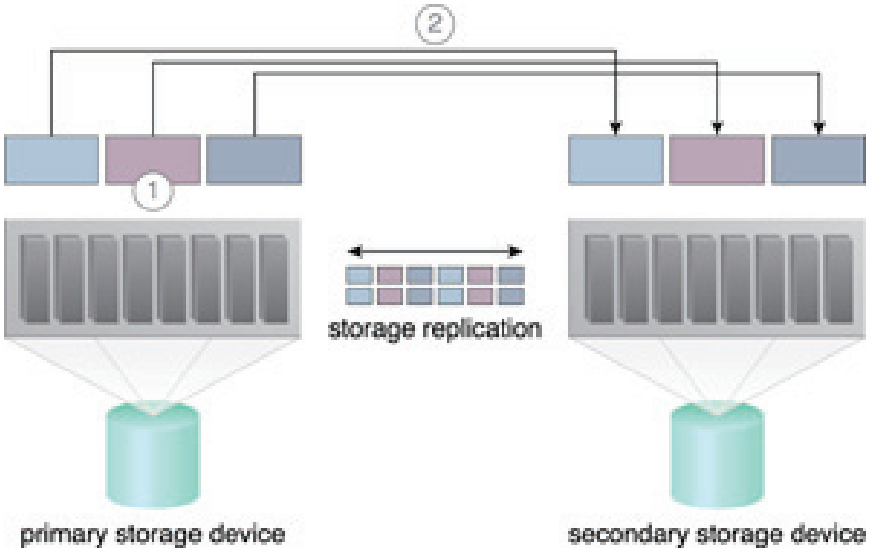


Figure 11.18 - Storage replication is used to keep the redundant storage device synchronized with the primary storage device.

Some cloud providers use storage devices with dual array and storage controllers to improve device redundancy, and place sec-ondary storage devices in a different physical location for cloud balancing and disaster recovery purposes. In this case, cloud providers may need to lease a network connection via a third-party cloud provider in order to establish the replication between the two devices.

11.9 Case Study Example

An in-house solution that ATN did not migrate to the cloud is the Remote Upload Module, a program that is used by their cli-ents to upload accounting and legal documents to a central archive on a daily basis. Usage peaks occur without warning, since the quantity of documents received on a day-by-day basis is unpredictable.

The Remote Upload Module currently rejects upload attempts when it is operating at capacity, which is problematic for us-ers that need to archive certain documents before the end of a business day or prior to a deadline.

ATN decides to take advantage of its cloud-based environment by creating a cloud-bursting architecture around the on-premise Remote Upload Module service implementation. This en-ables it to burst out into the cloud whenever on-premise pro-cessing thresholds are exceeded (Figures 11.19 and 11.20).

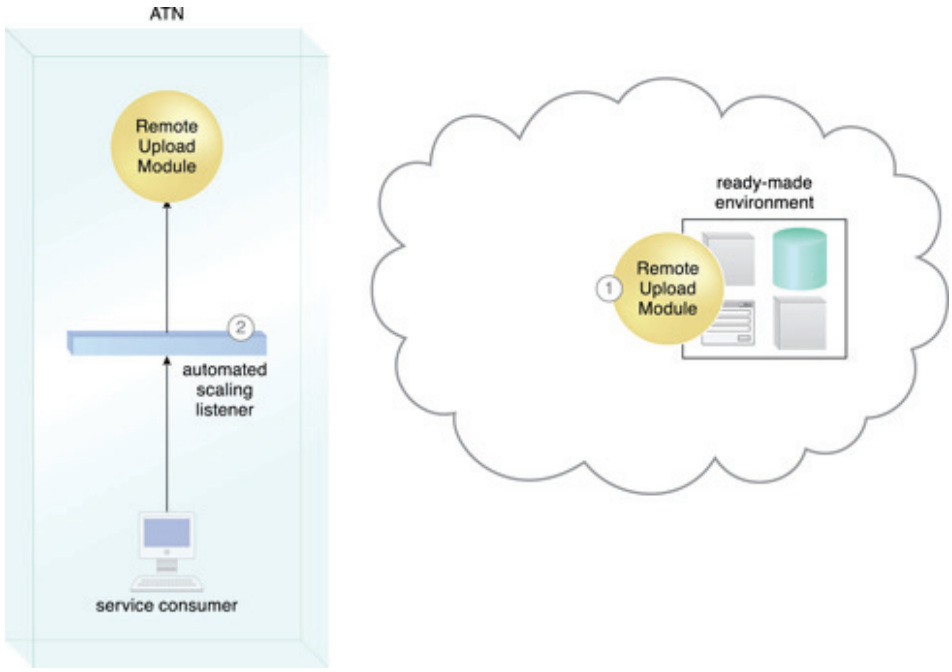


Figure 11.19 - A cloud-based version of the on-premise Remote Upload Module service is deployed on ATN's leased ready-made environment (1). The automated scaling listener monitors service consumer requests (2).

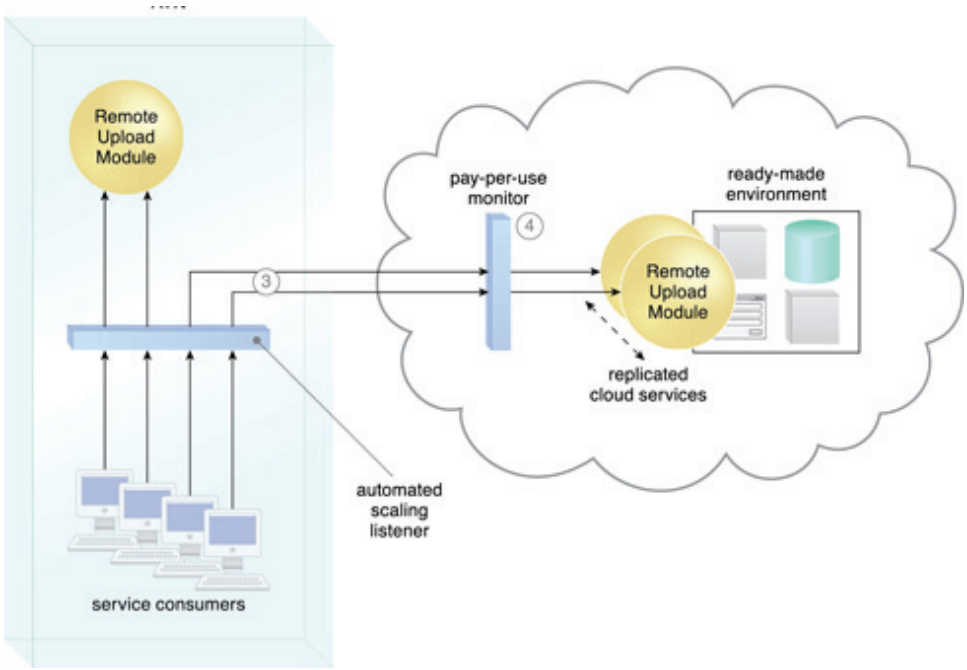


Figure 11.20 - The automated scaling listener detects that service consumer usage has exceeded the local Remote Upload Module service's usage threshold, and begins diverting excess requests to the cloud-based Remote Upload Module implementation (3). The cloud provider's pay-per-use monitor tracks the requests received from the on-premise automated scaling listener to collect billing data, and Remote Upload Module cloud service instances are created on-demand via resource replication (4).

A “burst in” system is invoked after the service usage has de-creased enough so that service consumer requests can be pro-cessed by the on-premise Remote Upload Module implementation again. Instances of the cloud services are released, and no additional cloud-related usage fees are incurred.

Cloud Architect Certification

Self-Study Kit Bundles
Now Available and Shipping Worldwide

www.cloudschool.com/certifications/architect

Arcitura
the IT education company

PROMETRIC

SOA Maturity Alongside Contract Standardization

by Jürgen Kress, Oracle, Speaker, Author, Hajo Normann, Oracle ACE Director, Danilo Schmiedel, Senior Consultant, Optiz Consulting, Guido Schmutz, Technology Manager, Trivadis, Bernd Trops, Senior Principal Consultant, Talend Inc., Clemens Utschig-Utschig, Chief Architect, Shared Service Centre, Global Business Services, Boehringer Ingelheim, Torsten Winterberg, Business Developement and Innovation, Optiz Consulting, and Berthold Maier, Enterprise Architect, T-Systems International department of Telekom Germany

Introduction: In Search of the Holy Grail of SOA

In this article, we present and explore the fundamentals of applying the factory approach to modern service-oriented software development in an attempt to marry SOA industrialization with service contracts. As service developers and designers, how can we successfully fulfill factory requirements and achieve the essential characteristic of industrialized SOA while remaining compliant with standards on the service contract level?

Thinking in terms of contracts has been found to be requisite for granular sourcing strategies that virtualize underlying implementations. Contracts also function as a common language between business units and IT teams, across cloud computing technologies, and for future-proof and agile enterprises in general.

Let's imagine that today's "pre-industrialized" world has become one in which contracts are been replaced by organizational and technical silos and the best solutions available. In today's SOA landscape, functional components are created for specific applications, often redundantly and lacking organization-wide standardization at the interface level. These components work well in a "silo" landscape in which the "application SOA" architecture is particularly suitable within the context of single applications.

Figure 1 illustrates the simplicity of combining services within applications that results from standardized design and structures being used as the framework for interfaces and exchanged data:

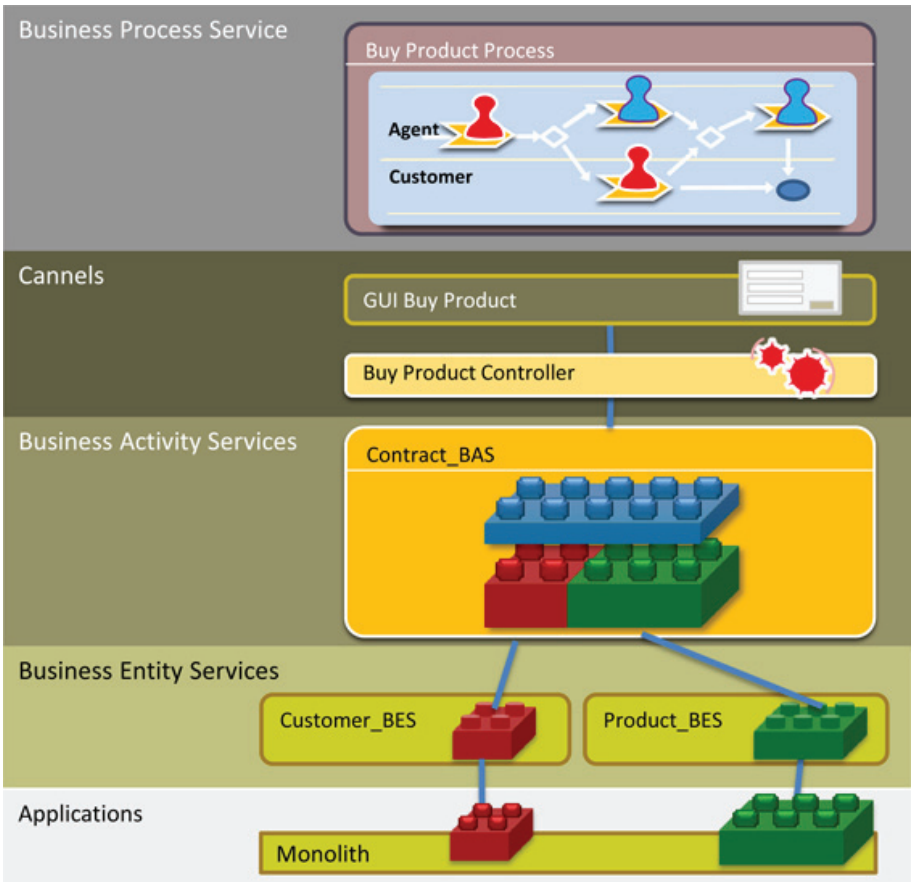


Figure 1 - SOA standards within applications result in compatibility.

If a business activity service (BAS) comprises business entity services (BESs) of different designs from multiple “application SOAs,” the data that is exchanged will vary greatly in structure. A single “contract” business object can be structured very differently for each application service (Fig. 2).

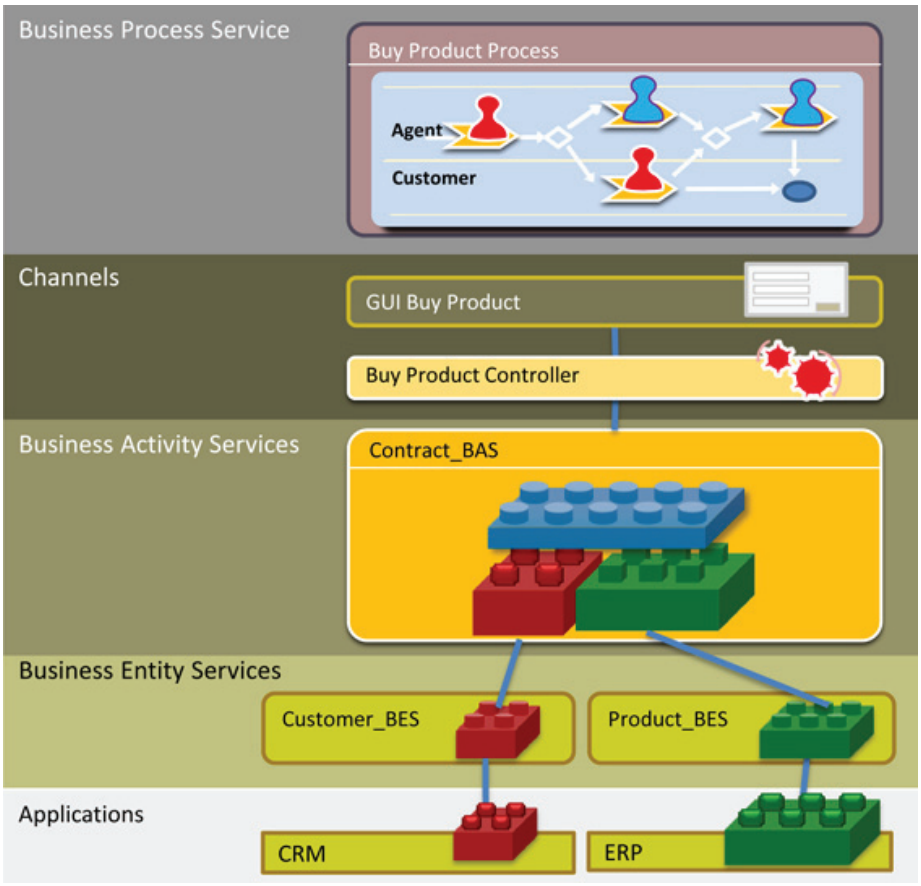


Figure 2 - Various standards in different applications.

Figure 3 illustrates how a high level of integration effort is required for each compiled service even though all of the services are based on underlying standards like SOAP and WSDL. Great integration effort is caused by the different structural characteristics of the data types.

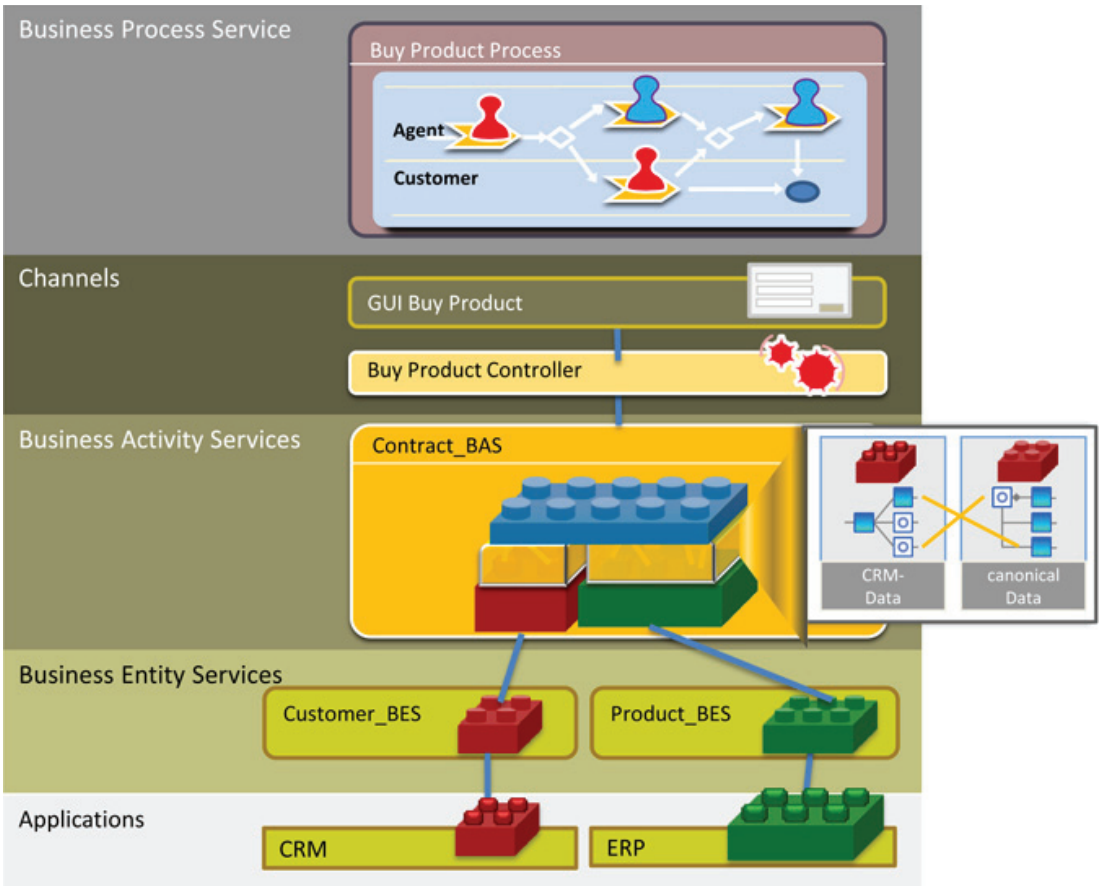


Figure 3 - A high level of integration effort is required in all service layers.

A high level of integration effort is required for service usage due to an absence of industry standards, which forces service reuse efforts to remain difficult and highly cost-intensive. These drawbacks cause developers to prefer building functionality themselves rather than using services. Integration effort is therefore a major obstacle on the path to successful SOA adoption.

The solution lies in domain-wide standards that are placed at the level of functional data exchanges and technical cross-sectional data exchanges. LEGO is chosen as a metaphor to demonstrate the necessity of standardization in service contracts, since the uniformity of LEGO brick structure produces the perfect fit. A LEGO brick that has just a minor imperfection or a small protrusion cannot be properly attached to other bricks. Similarly, simplified service reuse can be achieved by merely standardizing the exchanged data.

Figure 4 illustrates how combining services that are based on the same standards can be as basic as building structures out of LEGO bricks. This is the holy grail of SOA. The artificial connector, the integration layer, becomes very small or ceases to exist.

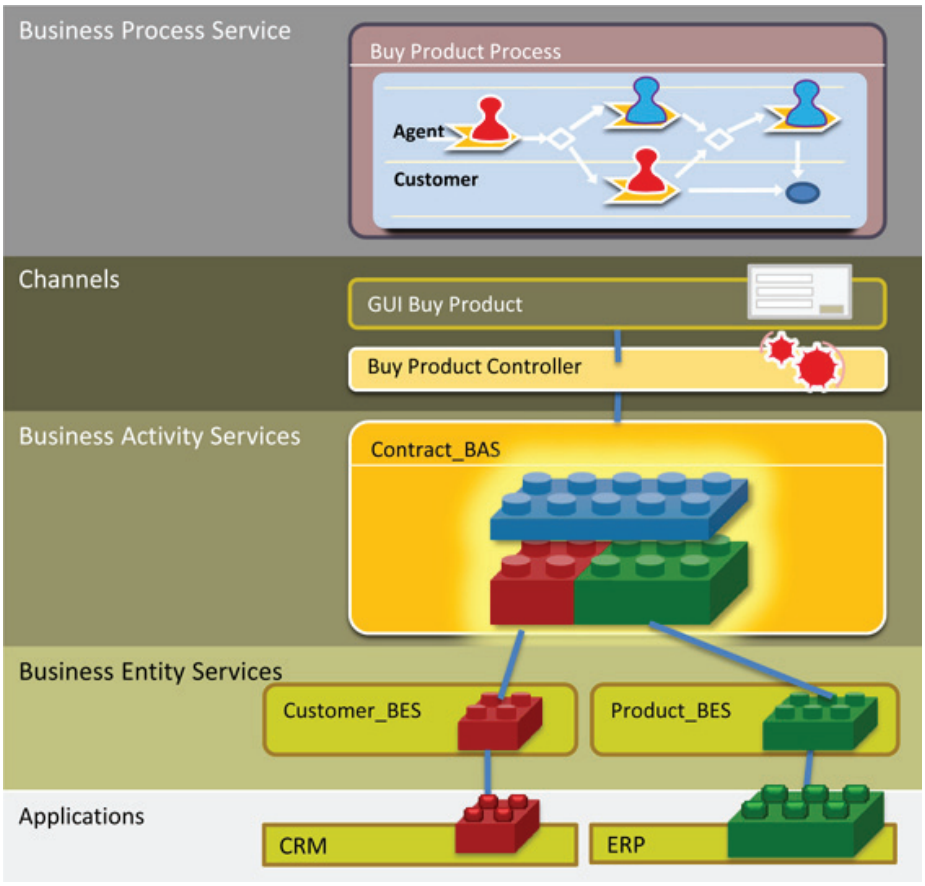


Figure 4 - The "holy grail" of SOA: easy use through standardized interfaces.

Achieving Standardization

One of the most important characteristics of the industrialization approach is a domain-wide standardization of interfaces or "service-level agreements." This agreement is a contract into which a service enters with its users, and is the prerequisite for BASs that are incorporated as function components for processes and portals. Considerable integration effort isn't required, and SLA compliance can in evaluated through enterprise asset management (EAM) or business process management (BPM) dashboards.

It is conceivable for specifications to be centrally determined on an SOA dashboard as an option for establishing the standards throughout an enterprise or organization. SOA governance tasks check and verify their use and compliance with the applicable specifications. These governance tasks require substantial effort to remain compliant with the specifications of standardized interfaces, which must be fulfilled for each service. The model-driven generation of service contracts can be implemented to reduce this effort and allow service developers to focus their attention on service implementation instead.

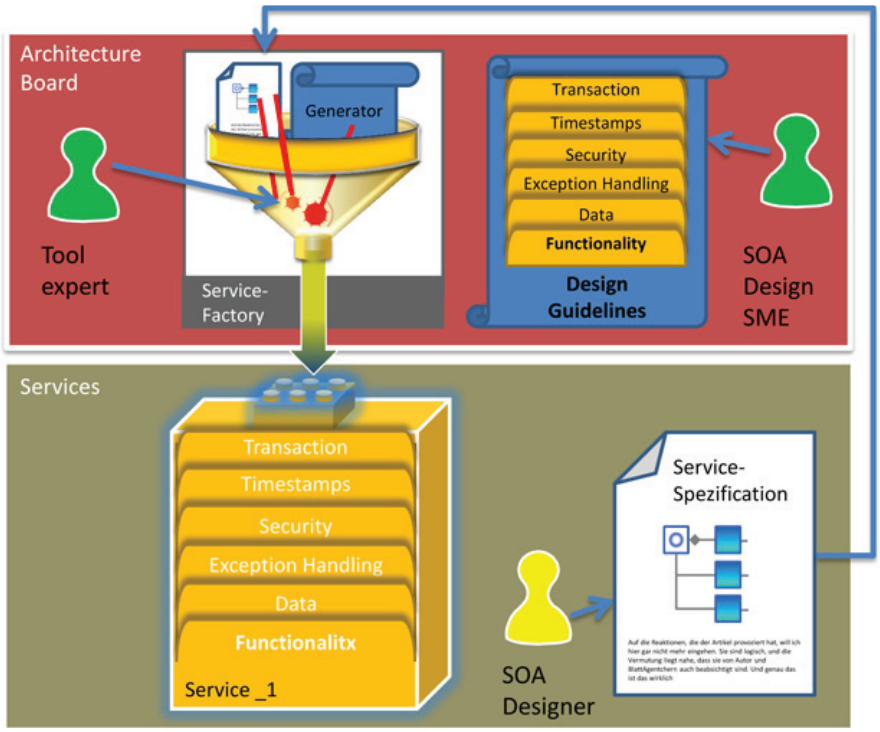


Figure 5 - Model-driven generators assist standardization.

SOA architects define business object data types for reference data types and cross-sectional data types to prepare for a range of errors. The underlying business objects and their attributes are saved as an object model, such as the unified modeling language (UML), in a central repository. Service designers use this centralized object repository when defining contracts and “feed” the repository into the generator, which creates the service interface in accordance with previously specified rules. This process demonstrates how a tool expert has defined the compilation of a standardized WSDL from selected data, cross-sectional data, and reference data.

SOAP messages are exchanged during runtime. The standardized filling in of SOAP headers, which also contain essential information for dashboard evaluations, is one of the tasks of the enterprise service bus (ESB). The ESB is configured for this task only once and doesn’t typically require any further involvement from the developer who programs services.

The interface specification, WSDL, and SOAP messages are now created according to the standards of the central architecture. Generating service contracts beyond or outside of this standard structure is not possible. The shift away from developer specifications and governance tasks that monitor specifications compliance, towards a generator-driven contract manufacturing process, is a key aspect of the implementation of the factory approach to industrialized SOA.

NEW

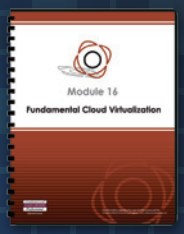
Cloud Certified Virtualization Specialist

CloudSchool.com™

CLOUD CERTIFIED

Virtualization Specialist

Arcitura Education Inc. has announced a new certification for the Cloud Certified Professional (CCP) program from CloudSchool.com, dedicated to cloud computing-based virtualization technology and practices. The new Certified Cloud Virtualization Specialist designation requires the completion of Prometric exams C90.01, C90.02, C90.16, C90.17 and C90.18. This certification track correspondingly introduces three new CCP courses and self-study kits.



CCP Module 16: Fundamental Cloud Virtualization

Core topic areas pertaining to the fundamental virtualization mechanisms and types used within contemporary cloud computing platforms.

CCP Module 17: Advanced Cloud Virtualization

A range of specialized and advanced topics that build upon module 16 to explore virtualization-related reliability, performance and integration, as well as combinations of mechanisms.

CCP Module 18: Cloud Virtualization Lab

A hands-on lab during which participants apply the models, concepts, and techniques covered in previous courses, in order to complete a series of complex exercises.

Workshops & Self-Study

Attend an instructor-led workshop or purchase the official Cloud Virtualization Specialist Certification Self-Study Kit Bundle.

PROMETRIC

Arcitura
the IT education company

Enterprise Service Bus

by Jürgen Kress, Oracle, Speaker, Author, Hajo Normann, Oracle ACE Director, Danilo Schmiedel, Senior Consultant, Optiz Consulting, Guido Schmutz, Technology Manager, Trivadis, Bernd Trops, Senior Principal Consultant, Talend Inc., Clemens Utschig-Utschig, Chief Architect, Shared Service Centre, Global Business Services, Boehringer Ingelheim, Torsten Winterberg, Business Developement and Innovation, Optiz Consulting, and Berthold Maier, Enterprise Architect, T-Systems International department of Telekom Germany

Introduction

Everyone seems to need to use an enterprise service bus (ESB) nowadays, but there is much confusion about its actual benefit and the various concepts this term entails. This uncertainty is revealed in statements like, “Help! My boss says we need an ESB,” or “Why do I need an ESB at all? Can’t I achieve the same thing with BPEL or BPMN?” or even “We can do everything ourselves in language X.” This article is an attempt to answer some of the most important questions surrounding this term using concrete examples, so that the areas of application that can be deemed “correct” for ESBs can be clarified:

- What exactly is the definition of an ESB? Is it a product or an architecture pattern?
- What are some practical uses for an ESB?
- Do I need an ESB to build an SOA platform?
- Which requirements do I need to satisfy?
- Which criteria can I use to select the ESB that is most suitable for my needs?

Defining the ESB

An accepted definition for this term has yet to be firmly established that is most likely caused by a lack of industry standards, whereas standards like BPEL and BPMN 2.0 exist for process engines and other components. The term “Enterprise Service Bus” was coined by Gartner in 2002, and further introduced by the analyst Roy Schulte to describe a category of software products that he observed were available on the market at that time. Ten years later, there is still very little agreement on what exactly an ESB is or what it should deliver. There are different definitions depending on the manufacturer or source. Among other things, an ESB is defined as:

“A style of integration architecture that allows communication via a common communication bus that consists of a variety of point-to-point connections between providers and users of services.”

“An infrastructure that a company uses for integrating services in the application landscape.”

“An architecture pattern that enables interoperability between heterogeneous environments, using service orientation.” (Figure 1)

Figure 1 - The ESB architecture pattern is divided into these main system architectures.

The ESBs that are available in today's market essentially differ in terms of the architecture of their systems. As shown in the preceding figure, they are mostly based on the following architectures:

Extended Message-Oriented Middleware (MOM)

These systems correspond to the original definition of ESB and typically distribute multiple nodes across the network, using a MOM infrastructure to support reliable messaging and event-driven processing among the nodes. Although the ESB nodes communicate using a proprietary protocol, service endpoints don't need to be aware of the MOM. Services can be exposed using a WSDL or other protocols.

Extended Integration Brokers

Over the last five years, traditional integration broker vendors have been adding support for Web services and repositioning their products as ESBs. These systems are more standards-compliant than they once were but still tend to be more proprietary than most ESBs. They also tend to provide a very centralized solution in which all messages pass through a centralized broker.

Extended Application Servers

A number of ESB vendors use a Java EE application server as the basis for their ESB products. These products are typically stronger in terms of service creation and composition than they are in legacy integration. They tend to be rather centralized, although they do support distributed nodes.

Endpoint-Based Plug-In Channels

A few ESB vendors support an extremely distributed model that implements service mediation at the service endpoint, and supports heterogeneous communications using a channel plug-in architecture

Mediation Agents

Although these products don't technically qualify as ESBs because a service platform is provided, more than one vendor has been known to label this type of product as such. Mediation agents can be centralized or distributed and support service mediation. There are also related product categories that implement parts of ESBs but are not officially marketed as ESBs by manufacturers [REF-1]:

XML Gateway

XML gateways are hardware appliances that primarily support service mediation, which is one of the key features of ESBs. In fact, XML gateways often support service mediation capabilities that ESBs do not or cannot support, such as transformation acceleration and the decryption and encryption of XML documents. However, XML gateways do not provide a service platform, a feature that is typically associated with ESBs.

Message-Oriented Middleware (MOM)

Message-oriented middleware is middleware that is based on the asynchronous sending and receiving of messages to offer looser coupling between applications. The format of the messages is not defined, although, in practice, XML has become established as the preferred format. A MOM product usually supports communication via message queuing and publish/subscribe. There is usually no provision for the transformation of messages in the MOM, and an application's interface to the message-oriented middleware is relatively complex and not well standardized. A MOM can be used as the basis for reliable message forwarding by an ESB.

Integration Brokers (EAI)

Some developers of traditional EAI tools are now positioning their products as ESBs. These products are often complex and proprietary and typically based on a hub-and-spoke architecture. The broker acts as the central message exchange (hub) around which the senders and recipients (spokes) are arranged radially. Connections to the broker are made via adapter ports that support the required message format.

Business Process Management Systems (BPMS)

Some ESB developers view service orchestration and automated business process management (BPEL, BPMN) as ESB characteristics, while others consider service buses to be a separate product that belongs to the BPMS category. This is especially the case when business aspects are expressed in process models or the category of an "Orchestration Engine," when technical integration processes can be expressed as a long process chain that calls services in a distinct order.

Application Servers

Many ESB platforms offer a service platform for building and hosting services. In this case, the ESB is also an application server. Many application servers provide containers for operating services and also offer a restricted capability for message processing and policy enforcement. Application server adapters support the integration of legacy systems through technologies such as Java EE Connector Architecture (JCA). In most cases, an application server only supports a few protocols, and a precise separation of ESB and application servers is difficult. Many developers require an application server as the basis for their ESB.

API Gateways

Companies and organizations aim to expose access to a subset of their key services and data to business partners and customers in an easy-to-use, standardized manner in the form of an API. This implies distinct security, performance, integration concerns that can then be addressed by API gateways. They encompass features for threat protection to ensure Quality of Service. They are not an ESB but do provide a certain overlap in terms of features. Examples for such shared features are transformations

and routings.

A general rule of thumb to note is that when services are exposed to the outside world, an API gateway is a tool to be considered. It is positioned outside of the Intranet, typically in the DMZ. It manages only the subset of services that is communicating with external parties.

An ESB is used for service virtualization, typically manage a much larger set of services, and is positioned inside the Intranet.

When Should an ESB be Used?

Are there best practices to determine when the implementation of an ESB is worthwhile? The use of an ESB is worth considering when three or more applications or services need to be integrated. A simple point-to-point integration is significantly easier and much more cost-effective when connecting two applications. An ESB can also be worthwhile if services are going to be incorporated from external service providers over which the company has no control. The ESB can then be used to monitor the service level agreements that the external provider guarantees. The impact of the adjustments to service contracts can further be kept to a minimum, as the ESB continues to provide a stable interface while making the necessary changes to the messages.

If many different protocols, such as HTTP, SOAP, and FTP, are to be used and standardized to one protocol like SOAP, the ESB can perform the necessary protocol transformation. If services are to be consistently incorporated into an architecture to be able to receive, process, and produce messages, then the use of ESB is also suitable. This is also applicable if a collection of pre-defined components and adapters need to be accessed, which allows various protocols and legacy applications to be integrated in a standardized fashion. If messages need to be reliably and securely processed, the ESB can simplify the implementation of transactional message flows between two heterogeneous transactional data sources.

Using an ESB can become problematic if large volumes of data need to be sent via the bus as a large number of individual messages. ESBs should never replace traditional data integration like ETL tools. Data replication from one database to another can be resolved more efficiently using data integration, as it would only burden the ESB unnecessarily.

An ESB should support stateless message flows if long-term business processes are to be implemented. Long-running business processes are stateful and can best be implemented using BPEL and/or BPMN. These are not usually available via the ESB, but rather via a business process management

system (BPMS).

ESB Blueprint

Due to the lack of standardization, the ESB market is rather confusing. There are many products that claim to be ESBs but offer quite different solutions and are based on different architectures. To allow for more effective evaluation of ESB products, the various functions that are assigned to an ESB have been arranged into a blueprint (Figure 2).

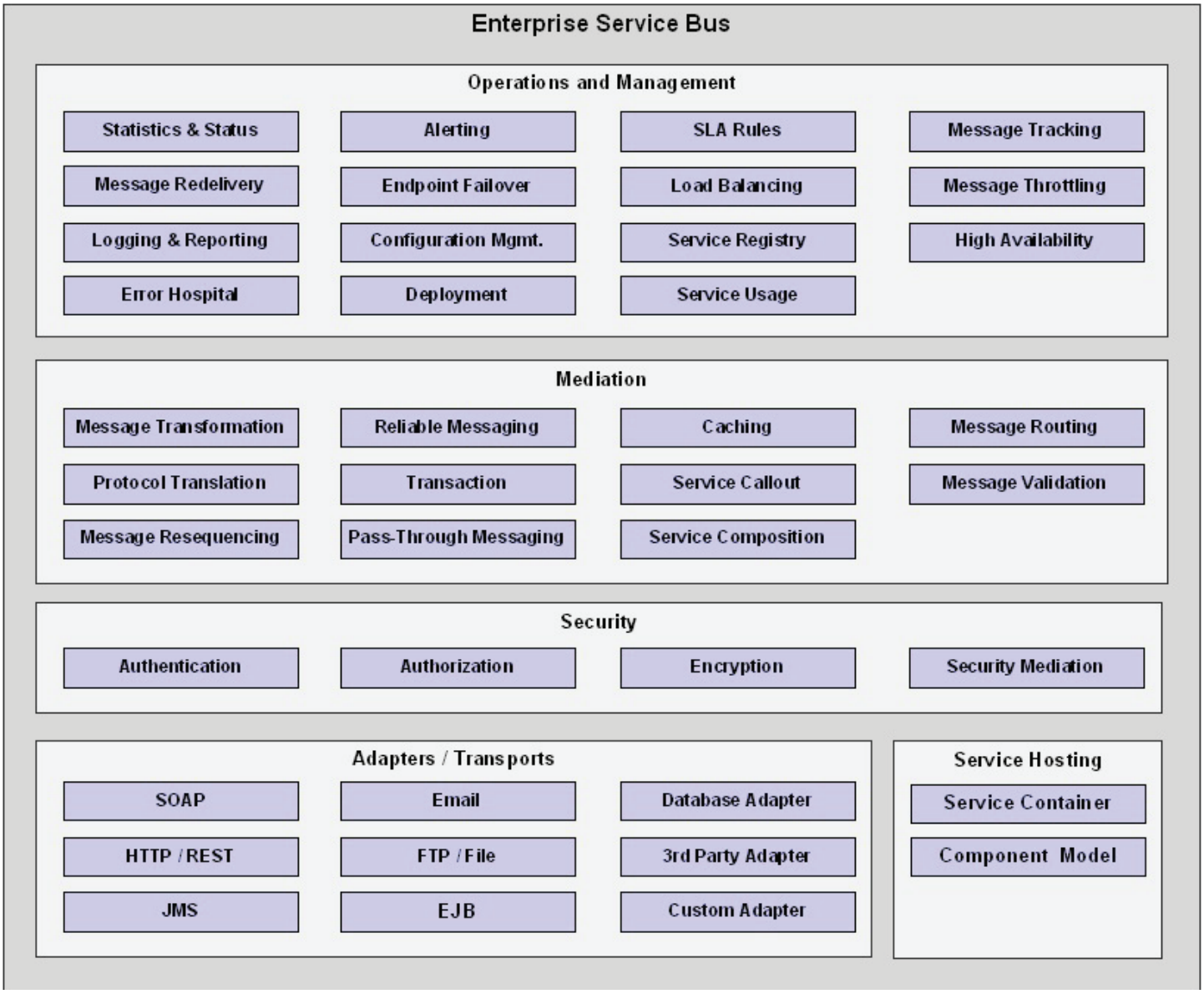


Figure 2 - The blueprint for an enterprise service bus.

The ESB blueprint diagram doesn't include an "orchestration" or "process choreography" component, as it is considered to be part of the BPMS category. These offer dedicated runtime environments for long-running, stateful business processes that are optimized for this and support languages such as BPMN or BPEL. ESBs should be stateless and configured to process messages as efficiently as possible.

Operations and Management Module

The following functional components in this module enable reliable operation and management of the enterprise service bus. Statistics & Status provides the services' ESB statistics, such as their number of errors, minimum and maximum response times, and number of processed messages. Alerting offers a mechanism for

sending alert messages that can be sent via various channels so that existing monitoring environments can also be incorporated. SLA Rules are rules that can be defined on the basis of information from the Statistics & Status functional component. This allows SLAs to be measured and monitored. Any SLA infringements are notified using the Alerting component.

Message Tracking provides the option of easily tracking messages within the ESB, and should be activated whenever required so as to minimize any associated overhead. Message Redelivery ensures that messages that aren't processed immediately are automatically resent after a pre-defined period of time. The number of attempts and the interval between them can be configured. This component is primarily suitable for technical errors that only last a certain length of time, such as temporary network outages. Endpoint Failover enables the option of specifying an alternate service provider that is automatically called whenever the primary service provider is not available.

Load Balancing allows several service endpoints to be listed for one logical service provider endpoint. It uses redundant service implementations that are called alternately for each request according to a defined strategy, which can be round-robin or according to message priority or load dependency.

Message Throttling makes it possible to define a maximum number of messages per unit of time for a service endpoint that should be sent to the service provider. It prevents the service provider from being overloaded at peak times by buffering messages that lie over the threshold in a queue in the ESB. Message Throttling can also support message priorities so that messages of higher priority are always processed first. Logging & Reporting allows messages to be logged and then easily displayed at a later time. It can also provide functional auditing.

Configuration Management enables secure configuration adjustments to the ESB on an operational system, while constantly upholding the integrity of the configuration. Artifacts and attributes can be adapted and replaced during operation. A history of the changes can also be kept so that an ESB service can be rolled back to an earlier status at any time. Service Registry offers the option of registering and managing services on the ESB. High Availability ensures that the services provided by the ESB are failsafe, regardless of the status of the server on which it is operated.

The Error Hospital is the destination for the messages that can't be processed after multiple redelivery attempts, where they can be viewed, corrected if necessary, and reprocessed. Deployment offers the option of installing services automatically on the ESB. Environment-specific parameters such as endpoint URIs are typically overridable by this component. Service Usage allows the use of services to be logged and charged to the user.

Mediation Module

The mediation module contains the functional components that are used to implement the message flow of an ESB service. Message Routing allows messages to be forwarded to a particular service endpoint depending on their content. The criteria for forwarding may originate from the message body or the message header, if the protocol used or the message format supports a header area that is independent of the message body.

Routing based on headers can be an attractive option to improve service performance and scalability, as direct access to the header is more efficient than parsing the routing information from the body. This is of particular consequence for larger messages.

Message Transformation enables conversion from one message format to another that is applicable to text and binary messages as well as XML formats. In addition, there is also the option of converting from text,

such as the CSV format, to XML and vice versa. XML transformations use the well-known standard XSLT, which enables declarative descriptions of transformations and has graphical resources with drag-and-drop functionality for creation purposes.

A major drawback of XSLT transformations is a high memory usage if large documents are being processed, which may restrict the scalability of a solution. It is preferable if the ESB offers transformation options that support XML streaming, such as via XQuery.

Service Callout offers the option of accessing other services within a message flow in the ESB, such as to enhance a message. A service may be a Web service but the ESB can conceivably enable program code that's installed locally on the ESB to be called directly, such as a Java class method. Reliable Messaging is the support of reliable message transfer using queuing or WS* standards, such as WS-ReliableMessaging.

Protocol Translation means the possibility of switching from a certain communication protocol to a different one without any programming effort, such as from TCP/IP to HTTP. Message Validation ensure that messages are valid. In the case of XML, this means that the message contains well-defined XML and corresponds to a certain XML schema or WSDL. However, the ESB can also offer other validation means, such as Schematron or a rules engine.

Message Exchange Pattern (MEP) is the support of message exchange patterns, such as synchronous and asynchronous request/reply, one-way call, and publish/subscribe. Result Cache provides the option of saving results from a service call in a cache, so that each subsequent call returning the same result can be answered from the cache without calling the service again. This particularly applicable if the data is static or requires sporadic or infrequent changes. Potentially expensive operations, such as accessing a legacy system, can be reduced significantly.

Transaction allows ESBs offer transactional integrity through message processing. The persistent queues that the ESB uses to support Reliable Messaging generally work as transactional data sources, and can therefore participate in heterogeneous transactions. In addition, the ESB can offer a distributed transaction manager that can coordinate distributed transactions via heterogeneous data sources using the two-phase commit protocol.

Message Resequencing allows a flow of messages that belong together but aren't in the correct order to be resequenced. In a message-oriented solution with parallel processing of messages, the sequence in which the messages enter the ESB can be lost. Message Resequencing can be incorporated into the message flow If the sequence is of import for the service provider. A resequencer contains an internal buffer that processes the messages until the complete sequence is available and can be sent.

Pass-Through Messaging provides efficient forwarding of messages by the ESB. This is useful if the ESB is to be used for service virtualization and the messages are forwarded from the service consumer to the service provider unchanged. In this case, it is applicable if the ESB doesn't touch the message and simply passes it on as is.

Security Module

This module supports both the transport-level and message-level security using a number of components. Authentication authenticates service consumers when they access the service in the ESB and verifies ESB authentication for the service provider. Authorization provides an authorization system for services that can often be configured via XACML to be assigned to users or roles.

Security Mediation provides support for interactions that communicate outside of security domains by converting credentials from one domain to the corresponding credentials of the other domain. Encryption/ Decryption supports the encryption and decryption of the content of a message.

Adapters/Transport Module

This module includes adapters for connecting services that are provided by the ESB via the Service Hosting module. The ESB can be assumed to provide a set of adapters from the ground up, and also has the option for customers or third-party developers to develop additional adapters for

customer-specific requirements.

Service Hosting Module

This module allows services to be installed and operated directly on the ESB and is usually required if the ESB is based on an application server. Service Container provides one or more containers in which the services are installed and service lifecycles managed. It offers the service access to technical cross-sectional functions, such as transactions and security.

The Component Model provides an abstract component model, such as Java EJB, Java Spring Framework, or Microsoft COM+, on the basis on which the services are created.

Scenarios for the Practical Use of an ESB

The symbols shown in Figure 3 are used to graphically describe the various scenarios without needing to refer to products and tools. The symbols have been taken from [1] and further added to, in accordance with ESB functionalities.

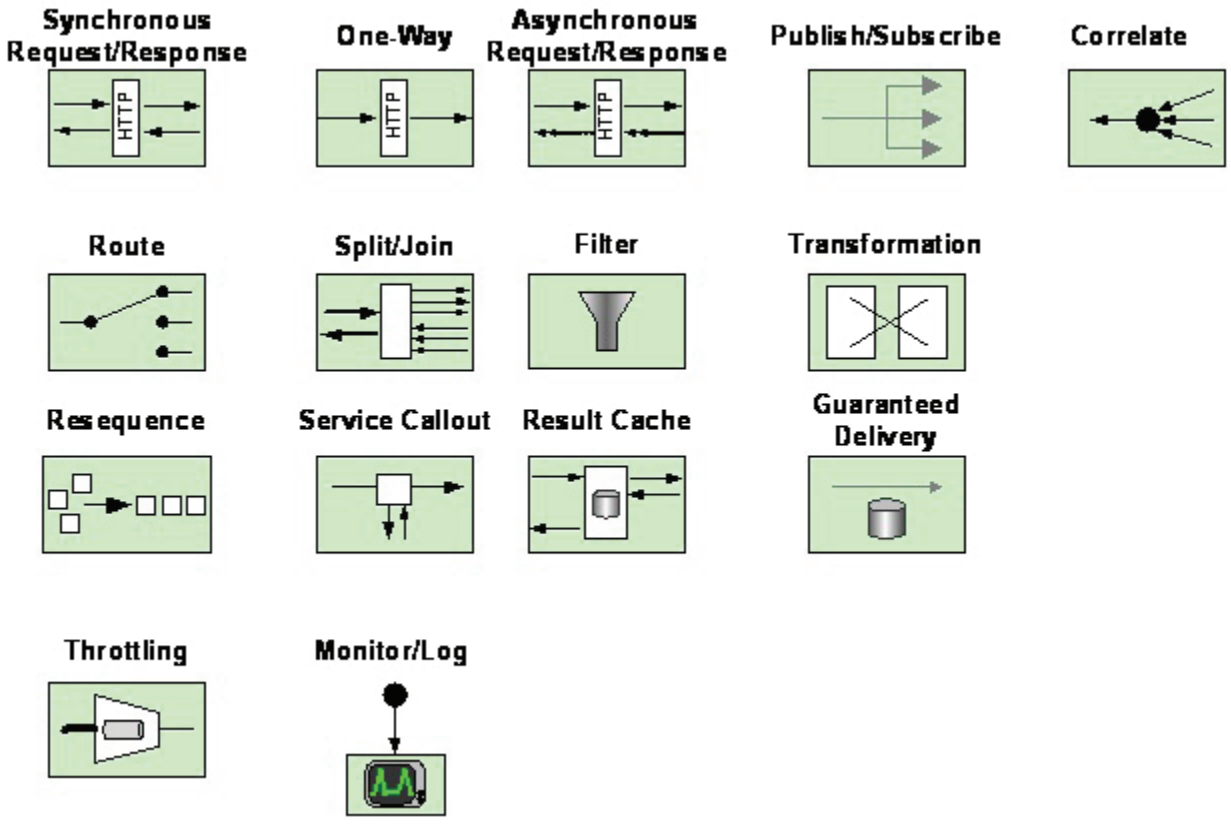


Figure 3 - Symbols for an ESB

Scenario 1 - Service Virtualization

Service consumers tend to prefer hard-wiring the actual endpoints of services, particularly in BPEL processes, because it's easy to perform with the tools that are provided. However, changes to the address of a server during runtime must not be able to produce changes that require redeployment at the service consumer side. An elegant solution around this problem is provided by the use of an ESB to virtualize endpoints. Figure 4 illustrates this scenario with a service provider that is providing a Web service interface that is no longer being directly used by the service consumer but by the ESB instead. The ESB can deliver the interface exactly as is to potential service consumers.

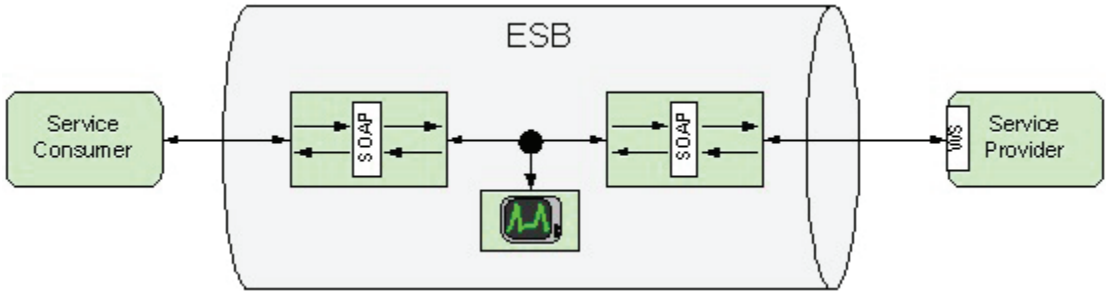


Figure 4 - Service virtualization with an additional monitoring interceptor

Any changes that need to be made to endpoint addresses can be easily implemented in the configuration of the ESB so that service consumers can continue to run as before. However, the ESB needs to be able to be incorporated into an existing message flow. The service virtualization also allows the use of the ESB's monitoring functions that extend to service statistics, so that SLA compliance can be checked and the appropriate actions configured if noncompliant. Service virtualization can be performed if the service provider makes a change to the service contract but doesn't want to impact the service consumer. In this case, a simple transformation of the exchanged messages can resolve the issue.

Scenario 2 - Service Enablement

When services with functional interfaces are incorporated, a situation often arises in which service consumers and service providers do not speak the same language at the protocol level. Figure 5 depicts two service providers that are technically offering the same service, since one provides a SQL interface and the other a FTP interface. The enterprise service bus can be used as a protocol converter to leave the interfaces untouched, since it provides the means to ensure that the defined interfaces are used.

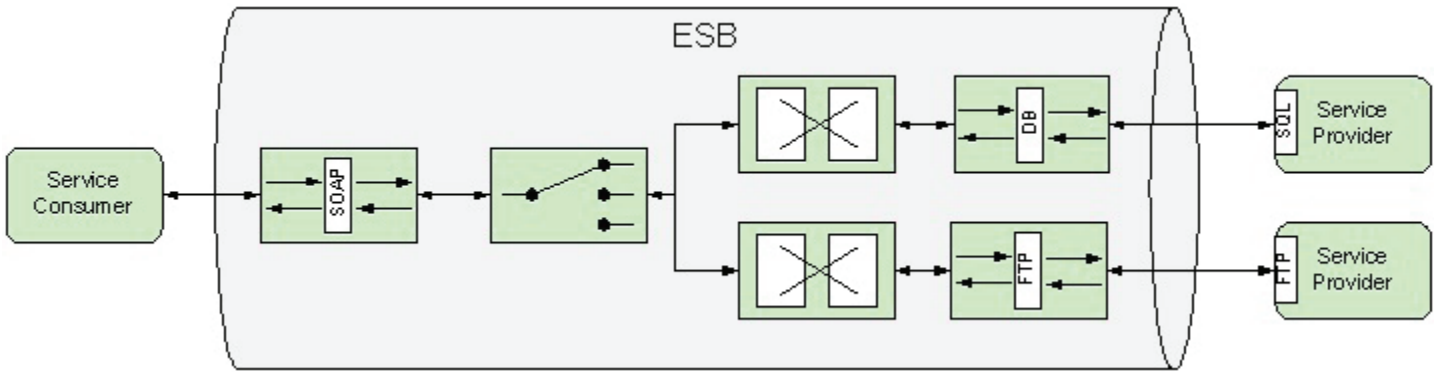


Figure 5 - Service enablement

Similar to Scenario 1, the ESB can ensure that no subsequent changes need to be made on either the service consumer or service provider side if the communication protocol were to change in the future.

Scenario 3 - Secure Message Processing

An ESB is also capable of supporting traditional integration scenarios in which the primary purpose is to forward messages from one system to another. Figure 6 illustrates a scenario in which the ESB consumes messages from an external queue, enriches them with a service callout to a Web service, and sends them to the destination system via a DB adapter.

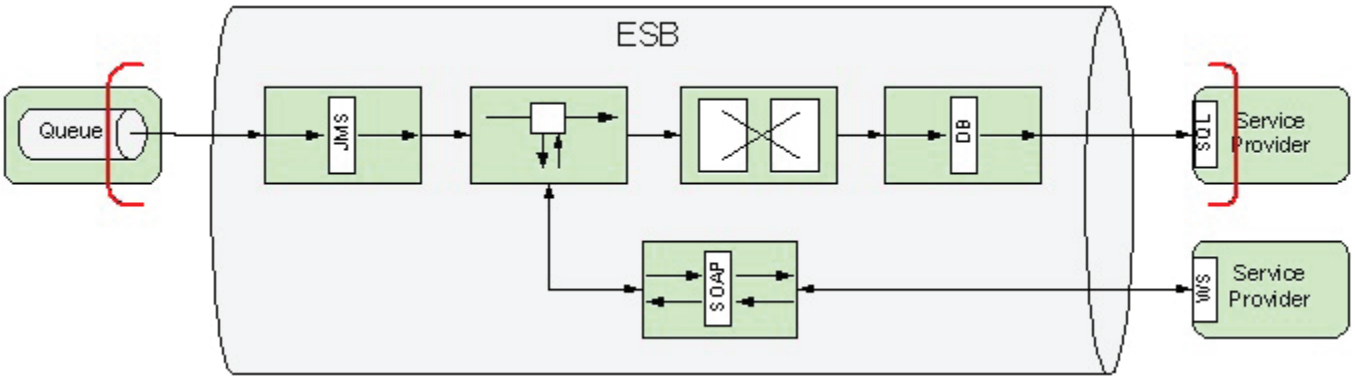


Figure 6 - Secure message processing

Processing in the ESB is transactional, meaning message flows are configured to become involved in a distributed XA transaction as additional participants. The transaction starts when the message is consumed from the queue, and also comprises the database operations that are discharged by the DB adapter. If the message flow is completed successfully, then committing of the distributed transaction follows.

Scenario 4 - Service Versioning

Services tend to change over time, and requirements that necessitate the introduction of a new and incompatible version are usually added. In such cases, the ESB can be used to perform the transformation from the "old" interface to the "new" interface. Figure 7 depicts a scenario in which the service provider introduces version 2.0 of a service that Service Consumer B immediately installs. Service Consumer A has been using Interface 1.0 and doesn't want to switch to version 2.0, since it wouldn't bring any added value to their business. However, the service provider doesn't want to keep running the two versions in parallel. Running the two interfaces simultaneously may be difficult or not even technically possible, as well as unnecessarily tie up resources.

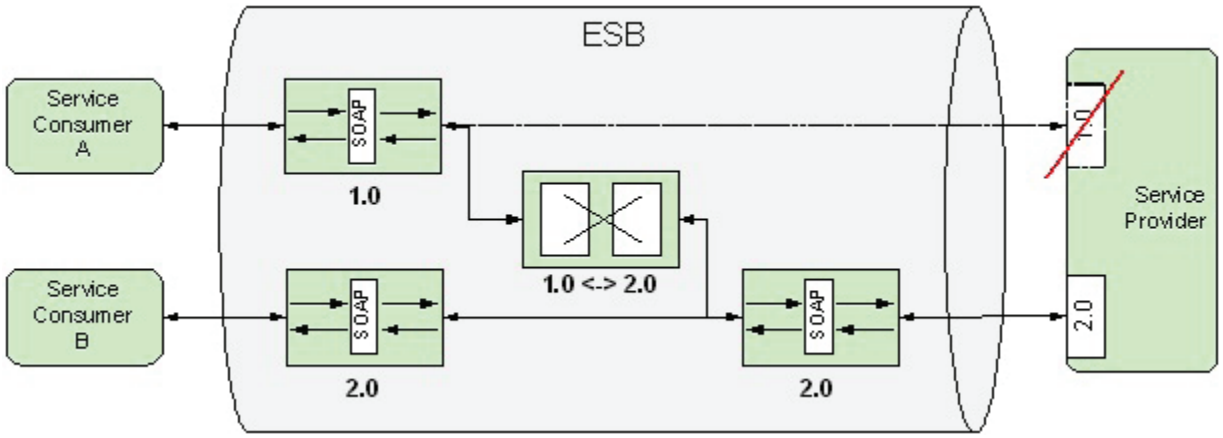


Figure 7 - Service versioning

The situation can be simplified if the ESB delivers version 2.0 directly via a pass-through that is similar to Scenario 1. At the same time, it has to keep providing version 1.0 of the interface while adapting the existing message flow so that version 1.0 is no longer called from the service provider. The message is instead transformed from versions 1.0 to 2.0 and sent to the new service. This transformation can be relatively complex, depending on how great the differences are between the two versions. Additional enrichment of the version 1.0 message may be necessary in order to deliver all of the information that is required to call version 2.0.

The transformation and Interface 1.0 in the ESB needs to be maintained until none of the service consumers are using the old interface. The reasons behind the decision to perform this transformation in the ESB instead of mapping from versions 1.0 to 2.0 within the service provider are as follows:

- Mapping is technical and unrelated with business-related issues.
- External service providers cannot be influenced.
- ESB makes the use of the old interface transparent.
- No changes are required for the service provider when all of the service consumers have switched to using Interface 2.0.

Conclusion: Summary

An ESB is a middleware solution that uses the service-oriented model to promote and enable interoperability between heterogeneous environments. There is no specification that defines exactly what an ESB is, or which functions it should provide. Even though an ESB is mostly associated with concepts like “mediation” and “integration,” it is also suitable as a platform for providing services in a way that is similar to an application server. The ESB represents the consolidation of the product categories that are called “integration” and “application server.”

One of the key features of an ESB is service virtualization. The ESB blueprint proposed in this article offers an orderly arrangement of its various functionalities and forms the basis for evaluating

ESB products.

Takeaways

- An enterprise service bus should be considered as an architecture style or pattern and not as a product.
- There is no definition or specification for the ESB and therefore no standard.
- An ESB can help achieve looser coupling between systems.
- A service on an ESB is stateless. Long-term processes do not belong to an ESB, but are supported in BPM systems in languages like BPEL and BPMN.
- Care should be taken when an ESB is “misused” for batch processing, as performance can be negatively affected.

Refernces

[REF- 1] Anne Thomas Manes: “Enterprise Service Bus: A Definition,” Burton Group”

SOA with REST

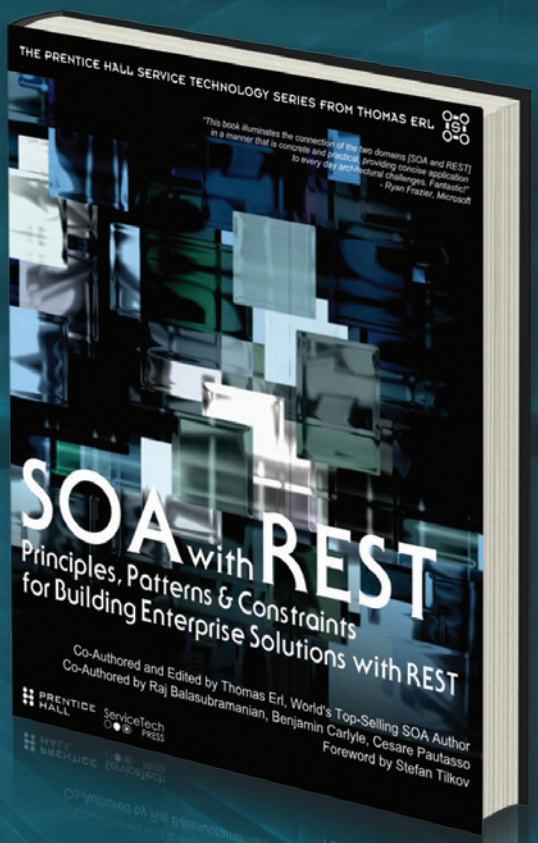
Principles, Patterns & Constraints for Building Enterprise Solutions with REST

"An inspirational book that provides deep insight into the design and development of next-generation service-oriented systems based on the use of REST. This book clarifies the convergence of SOA and REST with no-nonsense content that addresses common questions and issues head-on. An essential 'instrument of modern service implementation' and a powerful body of knowledge for software designers, architects and consultants."

Pethuru Raj PhD,
Enterprise Architecture Consultant,
Wipro Consulting Services

"This book illuminates the connection of the two domains - SOA and REST - in a manner that is concrete and practical, providing concise application to every day architectural challenges. Fantastic!"

Ryan Frazier,
Technology Strategist,
Microsoft Corp.



The "SOA with REST: Principles, Patterns & Constraints for Building Enterprise Solutions with REST" book has just been released. This is the eighth published book, as part of the recently rebranded, Prentice Hall Service Technology Series. The book authored by Thomas Erl, Benjamin Carlyle, Cesare Pautasso and Raj Balasubramanian was officially launched at the 5th International SOA, Cloud + Service Technology Symposium.

To learn more about this book, visit:
www.servicetechbooks.com/rest

TABLE OF CONTENTS

PRENTICE HALL
ServiceTech PRESS
Arcitura the IT education company

Promoting Organizational Visibility for SOA and SOA Governance Initiatives

by Manuel Rosa, SOA Governance Practice Leader, Link Consulting and André Sampaio, Enterprise Architecture Senior Consultant, Link Consulting

The costs of technology assets can become significant and the need to centralize, monitor and control the contribution of each technology asset becomes a paramount responsibility for many organizations. Through the implementation of various mechanisms, it is possible to obtain a holistic vision and develop synergies between different assets, empowering their re-utilization and analyzing the impact on the organization caused by IT changes. When the SOA domain is considered, the issue of governance should therefore always come into play.

Although SOA governance is mandatory to achieve any measure of SOA success, its value still passes incognito in most organizations, mostly due to the lack of visibility and the detached view of the SOA initiatives. There are a number of problems that jeopardize the visibility of these initiatives: Understanding and measuring the value of SOA governance and its contribution – SOA governance tools are too technical and isolated from other systems. They are inadequate for anyone outside of the domain (Business Analyst, Project Managers, or even some Enterprise Architects), and are especially harsh at the CxO level.

Lack of information exchange with the business, other operational areas and project management – It is not only a matter of lack of dialog but also the question of using a common vocabulary (textual or graphic) that is adequate for all the stakeholders. We need to generate information that can be useful for a wider scope of stakeholders like Business and enterprise architectures. In this article we describe how an organization can leverage from the existing best practices, and with the help of adequate exploration and communication tools, achieve and maintain the level of quality and visibility that is required for SOA and SOA governance initiatives.

Introduction

Understanding and implementing effective SOA governance has become a corporate imperative in order to ensure coherence and the attainment of the basic objectives of SOA initiatives:

- develop the correct services
- control costs and risks bound to the development process
- reduce time-to-market

The criticality and difficulty of achieving these objectives steeply increases with both the increase of the number of services and services' complexity. It is therefore crucial that a strategy is implemented to support a long-term vision of SOA endeavors, based on rules, policies, procedures, standards and best-practices, to best support the decision process and to control the effort involved in the design, implementation and maintenance of the services. A correct SOA governance model answers key questions:

- How can an existing service be leveraged to add value to other SOA solutions?
- Which decisions need to be made in your organization to have effective SOA, and who should make them?
- How can SOA decisions be monitored?
- Which structures, processes, and tools should be defined and deployed?

- Which metrics are required to ensure that an organization's SOA implementation meets their strategic goals?
- How can your Project and Application Portfolios be leveraged through SOA?

These questions can be partially answered by the information collected from different sources into SOA governance repositories, like Oracle Enterprise Repository (that we will use as an example throughout this article). Nevertheless, this may not be enough to answer all of them, at least not directly.

Although a solid governance strategy is fundamental for SOA, it is not complete without a global enterprise architecture Vision that provides mechanisms and tools to enrich the information needed for an application and project portfolio management.

By creating a formal common representational model, a language (graphic and textual), and standard viewpoints, and extending the basic capabilities of a SOA governance tool, one can leverage the information for a greater scope and number of analysis possibilities (time-based, dependency).

Addressing a Wider Audience

SOA governance is based on rather simple and intuitive concepts, but is very difficult to implement in its own full range. This is often due to the fact that not every role in the organization understands its real value or objective. The problem resides in the way that information is transmitted from the SOA Governance Center of Excellence to the rest of the company's audience.

In a typical SOA governance approach, a SOA Center of Excellence [REF-1] is comprised by stakeholders from different backgrounds and with varying competencies, therefore guaranteeing that each perspective of a SOA initiative is covered. Those stakeholders can be SOA Governance Specialists, Enterprise Architects, Operations, Security, Quality Assurance Specialists or Business Analysts. It becomes clear right from the start that every role has different objectives and needs regarding SOA, and that its significance varies from one to another.

Even though this group often concentrates the core SOA knowledge in an organization, the information still needs to be passed on to their peers, and handing over unstructured information that doesn't fully meet the target-audience needs can easily become an obstacle for SOA and SOA governance adoption. This can be simply because the information is not present in the repository, or, if it is, is too technical for some audiences. It may sometimes even just be from the need of having to produce some of the information manually, therefore becoming time-consuming, from unstructured models over and over again.

A SOA repository can only provide some views over the information out-of-the box, and some roles' needs and concerns are therefore not fully addressed. Through the usual harvesting mechanisms, technical information is collected from service buses, UDDI Registries, directly from service contracts, monitoring/operational systems or other sources, to a SOA repository. But there's still a gap to fill for Enterprise or Business Architects / Analysts or Project Managers who, without some kind of treatment over that information, can't make use of the full advantages of a SOA governance practice.

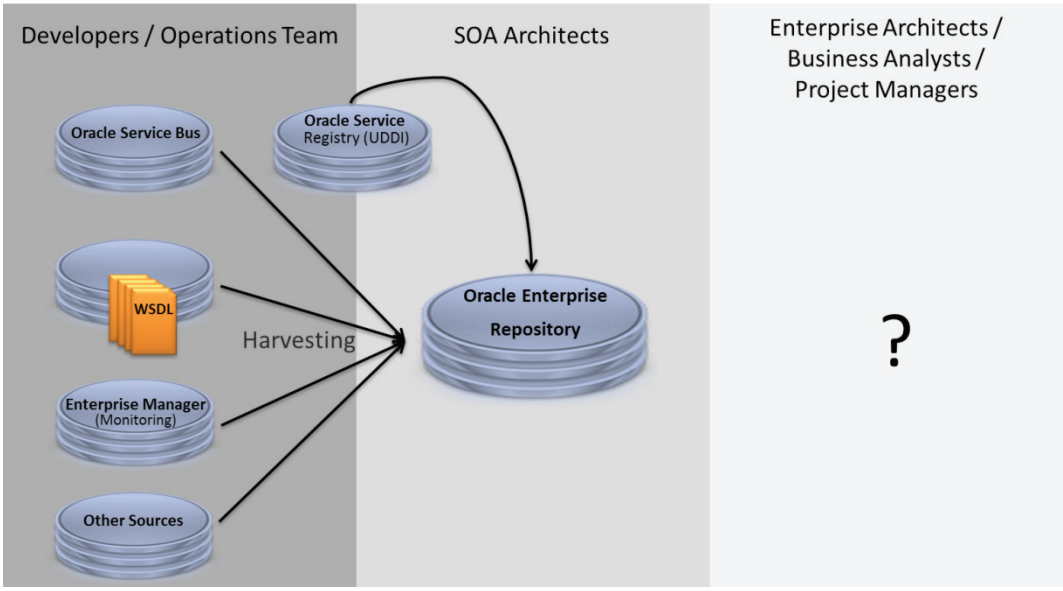


Figure 1 – The SOA Information Cycle, role-oriented.

A Catalog is Not Enough

Even though SOA governance tools are complete and offer a set of functionalities out-of-the-box for the common integration architect, developer and even for operation teams, they nevertheless still feel rather complex when the project manager or the business analyst try to use them. The asset type list is usually long and has a lot of details that these types of roles don't need to be confronted with. The dependency network between assets is often insufficient and complex, and the most important part of it, the detection of patterns, is somewhat neglected.

SOA Governance as Part of IT Governance

By filtering out the information that's not needed and correlating assets through connection, patterns can be fundamental for higher level impact analysis. For instance, a "simple" SOA/API Repository can be used for some Application Architecture analysis without even mentioning the services that some applications can provide or subscribe.

The same happens while managing Project Portfolios. If you can extend service lifecycle information with estimate dates for each stage, you can use that and predict dependencies and impacts between different projects. Another fundamental aspect is the relationship between SOA and the Information Architecture (often through the definition of a Common Data/Information Model), or Business/Process Architecture (through Business Processes) that can very useful for Business Analysts. They tend to search in SOA Repositories for business concepts and not for technical aspects, so it's fundamental to have this alignment between the services being developed and the organization's business concepts to formalize such relationships within the SOA Repository.

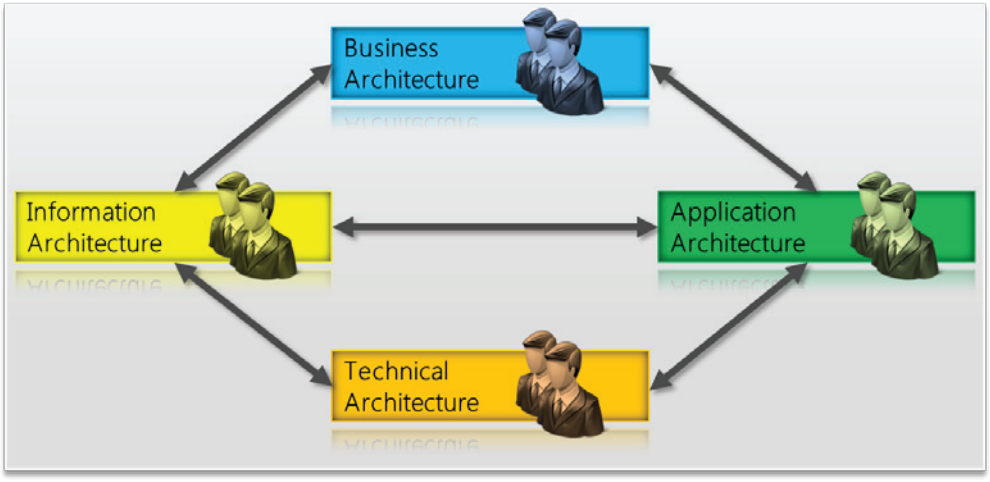


Figure 2 – SOA governance as part of enterprise architecture

Establishing a Common Model

Communication is critical. Human communication can be analyzed as a matter of encoding information (formulating a sentence), transmitting that message (speaking), and decoding the message (listening and understanding). Successful communication requires clear channels of transmission, and shared codes. Misunderstandings result from mistranslated messages, or from gaps or extraneous noise in the message [REF-5]. The information encoding is achieved by combining both syntax and semantics.

In the current context, it is straightforward to assume that, when we need to ensure an effective communication process surrounding the subject, the least one can do is to define a common vernacular. By establishing the concepts allowed, describing them in an unambiguous fashion, and expressing the relationships between the concepts, we are defining the language, the model.

This metamodel is the foundation for communication. SOA has a surrounding environment, the people, the processes, the ongoing projects, and by combining part of the holistic view of enterprise architecture, we can add value and broaden the reach of the base technologic assets that compose typical SOA governance tools, thus creating more robust foundations for the SOA initiative and establishing a clearer, symbiotic embrace with the IT transformation programs of the organization.

Communication vehicles include our applications, reports, mails and presentations that encompass SOA-related information. Any current SOA governance tool will be able to present lists and tables containing the services, the WSDLs, and so forth. The need for a broader audience encompasses different stakeholder concerns. Current tools are designed for technical roles and have no focus on other roles inside the organization. As pointed out by (Moody, 2009) [REF-2], a quarter of our brain is dedicated to vision, more than the sum of all the other senses. Our brain likes to receive information in a visual way, and it can process it in quite an efficient way. We defend that in many cases, drawings and diagrams can transmit the information in a more concise and precise way than through textual language.

To communicate with diagrams, we must unequivocally assign a representation element to each concept that is part of the metamodel. There are a number of notations, and other efforts to normalize the representation and the structure of SOA related information. Let's begin with the notation established by books in the Prentice

Hall Service Technology Series from Thomas Erl [REF-3] where each concept matches a specific symbol and the aggregation and organization of the later aid the organization in the endeavor of transmitting the correct vision of each principle or pattern.

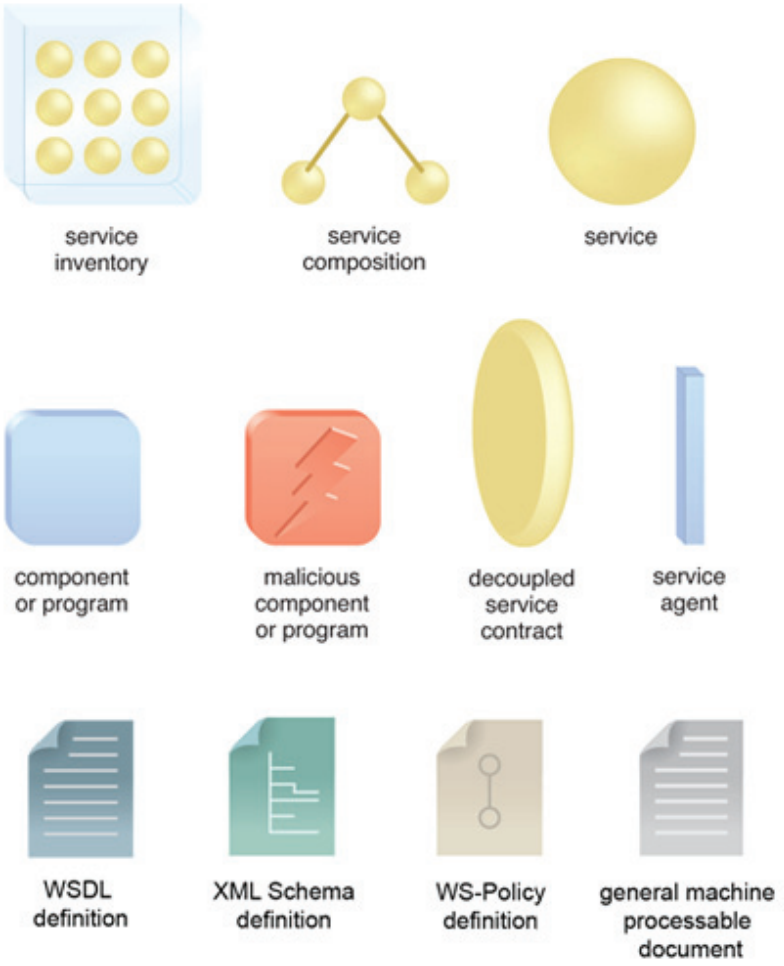


Figure 3 – Notation examples

Technical diagrams are more than just a set of symbols; they have rules, they have viewpoints, which help establish how a given view should be constructed. With a defined metamodel, the next step is to identify and specify the viewpoints. With all three defined we get our model and communication vehicles, as explored further in Part II of this article.

Q3/2013 Workshop Calendar

Certified SOA Architect

July 8-12, 2013
Las Vegas, Nevada, USA

SOASchool.com™ SOA CERTIFIED *Architect*

Certified Cloud Architect

July 22-26, 2013
Dubai, UAE

CloudSchool.com™ CLOUD CERTIFIED *Architect*

Certified SOA Consultant

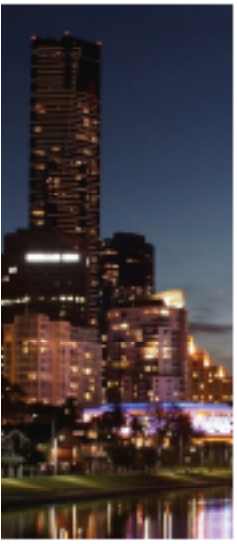
July 29 - August 2, 2013
Las Vegas, NV, USA

SOASchool.com™ SOA CERTIFIED *Consultant*

Certified Cloud Architect

July 29 - August 2, 2013
London, UK

CloudSchool.com™ CLOUD CERTIFIED *Architect*



Certified SOA Architect

August 12-16, 2013
Singapore, Singapore

SOASchool.com™ SOA CERTIFIED *Architect*

Certified Cloud Architect

August 12-16, 2013
Amsterdam, Netherlands

CloudSchool.com™ CLOUD CERTIFIED *Architect*

Certified Cloud Architect

August 19-23, 2013
Sydney, Australia

CloudSchool.com™ CLOUD CERTIFIED *Architect*

Certified Cloud Architect

August 19-23, 2013
Bangalore, India

CloudSchool.com™ CLOUD CERTIFIED *Architect*

Cloud Technology Professional

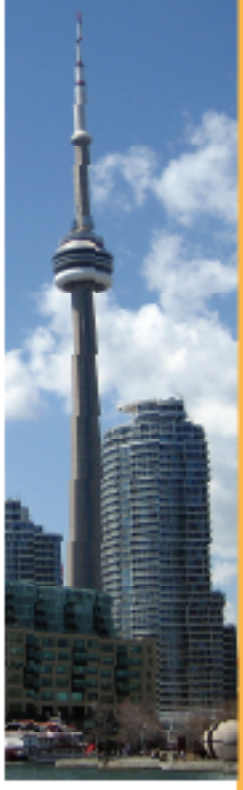
August 24-26, 2013
Cairo, Egypt

CloudSchool.com™ CLOUD CERTIFIED *Technology Professional*

SOA Governance Specialist

September 23-27, 2013
Vancouver, BC, Canada

SOASchool.com™ SOA CERTIFIED *Governance Specialist*



Arcitura
the IT education company

www.arcitura.com/workshops

SOA Architect Certification
June 24-28, 2013 • Quito, Ecuador

SOA Architect Certification
June 24-28, 2013 • Fairfax, VA, USA

Cloud Computing Architect Training Certification
June 26-28, 2013 • Singapore, Singapore

Cloud Architect Certification
July 1-5, 2013 • IST

Cloud Technology Professional Certification
July 8-10, 2013 • Guadalajara, Mexico

SOA Architect Certification
July 8-12, 2013 • Las Vegas, NV, USA

SOA Governance Specialist Certification
July 15-17, 2013 • Vancouver, BC, Canada

Cloud Architect Certification
July 15-19, 2013 • Las Vegas, NV, USA

Cloud Technology Professional Certification
July 22-24, 2013 • Toronto, ON, Canada

Cloud Architect Certification
July 22-26, 2013 • Dubai, UAE

SOA Consultant Certification
July 29 - August 2, 2013 • Vancouver, BC, Canada

SOA Architect Certification
July 29 - August 2, 2013 • Utrecht, Netherlands

Cloud Architect Certification
July 29 - August 2, 2013 • London, UK

SOA Architect Certification
July 29 - August 2, 2013 • London, UK

SOA Security Specialist Certification
August 5-7, 2013 • London, UK

SOA Architect Certification
August 5-9, 2013 • Melbourne, Australia

Cloud Architect Certification
August 5-9, 2013 • Las Vegas, NV, USA

SOA Professional Certification
August 9-10, 2013 • Panama City, Panama

SOA Architect Certification
August 12-16, 2013 • Singapore, Singapore

Cloud Architect Certification
August 12-16, 2013 • Melbourne, Australia

SOA Architect Certification
August 12-16, 2013 • Las Vegas, NV, USA

SOA Analyst Certification
August 12-16, 2013 • Amsterdam, Netherlands

Cloud Architect Certification
August 12-16, 2013 • Amsterdam, Netherlands

SOA Security Specialist Certification
August 19-23, 2013 • Vancouver, BC, Canada

Cloud Architect Certification
August 19-23, 2013 • Sydney, Australia

Cloud Architect Certification
August 19-23, 2013 • Singapore, Singapore

Cloud Architect Certification
August 19-23, 2013 • Bangalore, India

Cloud Technology Professional Certification
August 24-29, 2013 • Cairo, Egypt

Cloud Technology Professional Certification
August 26-28, 2013 • Lagos, Nigeria

SOA Analyst Certification
August 26-30, 2013 • Vancouver, BC, Canada

Cloud Architect Certification
August 26-30, 2013 • Stockholm, Sweden

Cloud Architect Certification
August 26-30, 2013 • São Paulo, Brazil

Cloud Architect Certification
August 26-30, 2013 • Lagos, Nigeria

SOA Architect Certification
September 2-6, 2013 • Sydney, Australia

SOA Architect Certification
September 2-6, 2013 • Singapore, Singapore

SOA Architect Certification
September 2-6, 2013 • Quito, Ecuador

Cloud Architect Certification
September 2-6, 2013 • Munich, Germany

SOA Consultant Certification
September 2-4, 9-10, 2013 • Quito, Ecuador

SOA Analyst Certification
September 2, 4, 9, 11, 12, 2013 • Quito, Ecuador

SOA Governance Specialist Certification
September 2, 4, 9, 13, 16-17, 2013 • Quito, Ecuador

SOA Security Specialist Certification
September 7-11, 2013 • Riyadh, Saudi Arabia

Cloud Architect Certification
September 9-13, 2013 • Singapore, Singapore

Cloud Architect Certification
September 9-13, 2013 • Melbourne, Australia

Cloud Architect Certification
September 9-13, 2013 • Fairfax, VA, USA

Cloud Architect Certification
September 9-13, 2013 • Dublin, Ireland

Cloud Architect Certification
September 9-13, 2013 • CET

Cloud Technology Professional Certification
September 16-18, 2013 • Naarden, Netherlands

SOA Architect Certification
September 16-20, 2013 • Rotterdam, Netherlands

SOA Architect Certification
September 16-20, 2013 • Fairfax, VA, USA

Cloud Architect Certification
September 16-20, 2013 • Dubai, UAE

SOA Architect Certification
September 16-20, 2013 • Dubai, UAE

Cloud Technology Professional Certification
September 16-20, 23, 2013 • Athens, Greece

Cloud Architect Certification
September 16-20, 23-27, 2013 • Athens, Greece

Updated: June 28, 2013

Service Security and Compliance in the Cloud

by Raghu Yeluri, Principal Engineer, Intel Architecture Group, Sudhir S. Bangalore, Senior Systems Engineer, Intel Architecture and Systems Integration (IASI)

Abstract: One of the biggest barriers impeding broader adoption of cloud computing is security—the real and perceived risks of providing, accessing and control services in multitenant cloud environments. IT managers need higher levels of assurance that their cloud-based services and data are adequately protected as these architectures bypass or reduce the efficacy and efficiency of traditional security tools and frameworks. The ease with which services are migrated and deployed in a cloud environment brings significant benefits, but they are a bane from a compliance and security perspective. IT managers are looking for greater assurances of end-to-end service level integrity for these cloud-based services. This article explores challenges in deploying and managing services in a cloud infrastructure from a security perspective, and as an example, discusses work that Intel is doing with partners and the software vendor ecosystem to enable a security enhanced platform and solutions with security anchored and rooted in hardware and firmware to increase visibility and control in the cloud.

Introduction

The cloud computing approach applies the pooling of an on-demand, self- managed virtual infrastructure, consumed as a service. This approach abstracts applications from the complexity of the underlying infrastructure, which allows IT to focus on the enabling of business value and innovation. In terms of cost savings and business flexibility, this presents a boon to organizations. But IT practitioners unanimously cite security, control, and IT compliance as primary issues that slow the adoption of cloud computing. These results often denote general concerns about privacy, trust, change management, configuration management, access controls, auditing, and logging. Many customers also have specific security requirements that mandate control over data location, isolation, and integrity that typically use legacy solutions that rely on fixed hardware infrastructures.[27]

Under the current state of cloud computing, the means to verify a service's compliance with most of the aforementioned security challenges and requirements are labor-intensive, inconsistent, nonscalable, or just not possible. For this reason, many corporations only deploy less critical applications in the public cloud and restrict sensitive applications to dedicated hardware and traditional IT architectures.[28] For business- critical applications and processes and sensitive data, however, third-party attestations of security controls usually aren't enough. In such cases, it is absolutely critical for organizations to be able to verify for themselves that the underlying cloud infrastructure is secure enough for the intended use.[29]

This requirement drives the next frontier of cloud security and compliance: building a level of transparency at the bottom-most layers of the cloud by developing the standards, instrumentation, tools, and linkages to monitor and prove that the IaaS clouds' physical and virtual servers are actually performing as they should and meet defined security criteria. Today, security mechanisms in the lower stack layers (for example, hardware, firmware, and hypervisors) are almost absent.

Cloud providers and the IT community are working earnestly to address these requirements, enabling cloud services to be deployed and managed with confidence, with controls and policies in place to monitor trust and compliance of these services in cloud infrastructures. Specifically, Intel Corporation and other technology companies have come together to enable a cloud infrastructure that is highly secure and based on a hardware

root of trust, which provides tamper proof measurements of key physical and virtual components in the computing stack, including the hypervisors. These organizations are collaborating to develop a framework to integrate the secure hardware measurements provided by the hardware root of trust into adjoining virtualization and cloud management software. The intent is to improve visibility, control, and compliance for cloud services. For example, having visibility into the trust and integrity of cloud servers allows cloud orchestrators to provide improved controls on onboarding services for their more sensitive workloads—offering more secure hardware and subsequently better controlling the migration of workloads and meeting security policies.

We will discuss how cloud providers and organizations can use the hardware root of trust as the basis for deploying secure and trusted services. In particular, we'll cover Intel® Trusted Execution Technology (Intel TXT) and the Trusted Compute Pool usage models, and envision the necessary ecosystem for implementing them.

Security in the Cloud

Security is a key barrier to the broader adoption of cloud computing. The real and perceived risks of providing, accessing and controlling services in multitenant cloud environments can slow or preclude the migration to services by IT organizations. In a non-virtualized environment, the separation provided by physical infrastructure is assumed to provide a level of protection for applications and data. In the cloud, this traditional physical isolation between applications no longer exists. Cloud infrastructure is multi-tenant, with multiple applications utilizing a shared common physical infrastructure. This provides the benefit of much more efficient resource utilization. However, because the physical barriers between applications have been eliminated, it is important to establish compensating security controls to minimize the potential for malware to spread through the cloud. This section covers the security challenges in the cloud, and provides a set of requirements that have to be addressed for cloud security.

Cloud Concepts

Cloud computing moves us away from the traditional model where organizations dedicate computing power (and devices) to a particular business application, to a flexible model for computing where users access applications and data in shared environments.[3] Cloud computing is a model for enabling ubiquitous, on-demand network access to a shared pool of convenient and configurable computing resources (such as networks servers, storage, applications, and services). Considered a disruptive technology, cloud computing has the potential to enhance collaboration, agility, efficiency, scaling, and availability; it provides the opportunity for cost reduction through optimized and efficient computing.

Many definitions attempt to address cloud computing from the perspective of different roles—academicians, architects, engineers, developers, managers, and consumers. For this article we'll focus on the perspective of IT network and security professionals; more specifically, for the security architects at service providers and enterprises in their quest to provide a more transparent and secure platform for cloud services.

The National Institute of Standards and Technology (NIST) defines cloud computing through five essential characteristics, three cloud service models, and four cloud deployment models.[14][30] They are summarized in visual form in Figure 1.

Cloud service delivery is divided among three archetypal models and various derivative combinations. The three fundamental classifications are often referred to as the SPI Model, where SPI refers to Software, Platform, or Infrastructure (as a Service), respectively defined thus[5]:

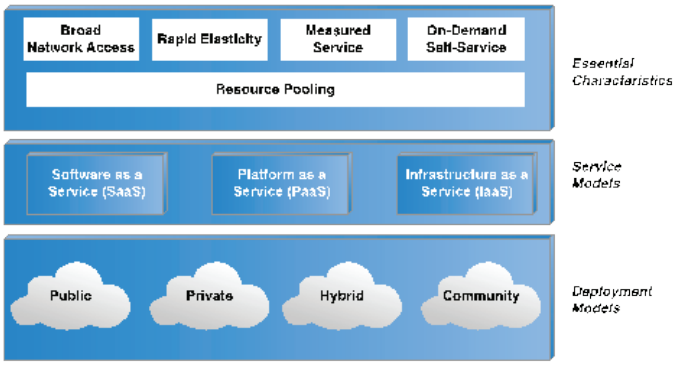


Figure 1 - nIST cloud computing dimensions[14] (Source: nIST Special publication 800-53, "recommended Security Controls for Federal Information Systems and organizations," revision3, 2010)

- *Software as a Service (SaaS)* - The capability where applications are hosted and delivered online via a web browser offering traditional desktop functionality, such as Google Docs, Gmail, and MySAP.
- *Platform as a Service (PaaS)* - The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications developed using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.
- *Infrastructure as a Service (IaaS)* - The capability where a set of virtualized computing resource, such as compute and storage and network are hosted in the cloud; customers deploy and run their own software stacks to obtain services. The consumer does not manage or control the base, underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (such as host firewalls).

In support of these service models and the NISTs deployment models (public, private, and hybrid), many efforts are centered around the development of both open and proprietary APIs that seek to enable things such as management, security, and interoperability for cloud computing. Some of these efforts include the Open Cloud Computing Interface Working Group, Amazon EC2 API, VMware's DMTF-submitted vCloud API, Sun's Open Cloud API, Rackspace API, and GoGrid's API, to name just a few. Open, standard APIs will play a key role in cloud portability, federation, and interoperability as well as common container formats such as the DMTF's Open Virtualization Format (OVF).[5]

The architectural mindset used when designing solutions has clear implications on the future flexibility, security, and mobility of the resultant solution, as well as its collaborative capabilities. As a rule of thumb, perimeterized solutions are less effective than de-perimeterized solutions in each of the four areas. Careful consideration should also be given to the choice between proprietary and open solutions for similar reasons.

The NIST definition emphasizes the flexibility and convenience of the cloud, which allows customers to take advantage of computing resources and applications that they do not own for advancing their strategic objectives. It also emphasizes the supporting technological infrastructure, considered an element of the IT supply chain that can be managed to respond to new capacity and technological service demands without the need to acquire or expand in-house complex infrastructures.

Understanding the dependencies and relationships between the cloud computing deployment and service models is critical to understanding cloud security risks and controls. With PaaS and SaaS built on top of IaaS, as described in the NIST model (above)[14] inherited capabilities introduce security issues and risks. In all cloud models the risk profile for data and security changes, and is an essential factor in deciding which models are appropriate for an organization. The speed of adoption depends on how security and trust in the new cloud models can be established.

Cloud Security, Trust, and Assurance

There is a significant amount of focus and activity across various standards organizations and forums to define the challenges, issues, and a solution framework to address these drivers. The Cloud Security Alliance, NIST, and the Open Cloud Computing Interface (OCCI), are examples of organizations promoting cloud security standards. Following are the key drivers for cloud security[17][12]:

- Visibility, compliance, and monitoring: providing seamless access to the security controls, conditions, and operating states within the cloud's virtualization and hardware layers for auditability and at the bottom-most infrastructure layers of the cloud security providers. The measured evidence would enable organizations to comply with security policies and with regulated data standards and controls such as FISMA and DPA.[13]
- Data discovery and protection: cloud computing places data in new and different places—not just user data, but also the application and VM data (source). Key issues include data location and segregation, data footprints, backup, and recovery.
- Architecture: standardized infrastructure and applications lead to more opportunity to exploit a single vulnerability many times—the BORE (Break Once, Run Everywhere) principle. Considerations for the architecture include: 1) Protection; how do you protect against attacks with standardized infrastructure when the same vulnerability can exist at many places, due to the standardization? 2) Multitenant environment; how do you ensure that systems and applications are appropriately and sufficiently isolated? 3) Security policies; how do you ensure that security policies are accurately and fully implemented across cloud architectures?
- Identity management: identity management (IdM) is described as the management of individual identities, their authentication, authorization, roles, and privileges/permissions within or across system and enterprise boundaries with the goal of increasing security and productivity while decreasing cost, downtime, and repetitive tasks.[25] From a cloud security perspective, questions like, "how do you control passwords and access tokens in the cloud?" and "how do you federate identity in the cloud?" are very real and thorny questions for cloud providers and subscribers to address.
- Automation and policy orchestration: the efficiency, scale, flexibility, and cost-effectiveness that cloud computing brings is because of the automation; the ability to rapidly deploy resources, scale up and scale down with processes, applications, and services provisioned securely "on-demand." A high degree of automation and policy evaluation and orchestration are required so that security controls and protections are handled correctly with very minimal scope of errors and with minimal intervention.

This article focuses on the first set of security drivers–visibility, compliance, and monitoring. Before we delve into these, we should start with a baseline definition of some key security concepts. These terms will form the foundation of what visibility, compliance and monitoring would entail. Let us start with baseline definitions for the terms security, trust, and assurance[23]:

- Security. Concerns the confidentiality, availability and integrity of data or information. Security also includes authentication and nonrepudiation.

- Trust. Revolves around the assurance and confidence that people, data, entities, information or processes will function or behave in expected ways. Trust may be human to human, machine to machine (for example, handshake protocols negotiated with in certain protocols), human to machine (for example, when a consumer reviews a digital signature advisory notice on a Web site), or machine to human. At a deeper level, trust might be regarded as a consequence of progress towards security or privacy objectives.
- Assurance. Provides the evidence or grounds for confidence that the security controls implemented within an information system are effective in their application. Assurance can be obtained by: 1) actions taken by developers, implementers, and operators in the specification, design, development, implementation, operation, and maintenance of security controls; 2) actions taken by security control assessors to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the system.

While the cloud provides organizations with a more efficient, flexible, convenient, and cost-effective alternative to owning and operating their own servers, storage, networks, and software, it also erases many of the traditional, physical boundaries and controls that help define and protect an organization's data assets. Physical servers are replaced by virtual ones. Perimeters are established not by firewalls alone but also by highly mobile virtual machines. As virtualization proliferates throughout the data center, the IT manager can no longer point to a specific physical node as being the home to any one critical process or data, because virtual machines (VMs) move around to satisfy policies for high availability or resource allocation. Public cloud resources usually host multiple tenants concurrently, increasing the need for isolated and trusted compute infrastructure as a compensating control. However, mitigating risk becomes more complex as the cloud introduces ever expanding, transient chains of custody for sensitive data and applications. Regulatory compliance for certain types of data would similarly become increasingly difficult to enforce in such models.

For this reason, the vast majority of data and applications handled by clouds today isn't business critical and has lower security requirements and expectations, tacitly imposing a limit on value delivered. Most organizations are already leasing computing capacity from an outside data center to host noncritical workloads such as Web sites or corporate e-mail. Some gone a small step further and have outsourced business functions such as sales force management to providers in the cloud. If their workloads were compromised or the business processes became unavailable for a short period of time, the organization might be highly inconvenienced, but the consequences would probably not be disastrous.

Higher-value business data and processes, however, have been slower to move into the cloud. These business-critical functions—for example, the cash management system for a bank or patient records management within a hospital—are usually run instead on in-house IT systems to ensure maximum control over the confidentiality, integrity, and availability of those processes and data. Although some organizations are using Cloud for higher value information and business processes, they're still reluctant to outsource the underlying IT systems, because of concerns about their ability to enforce security strategies and to use familiar security controls in proving compliance.

Security and Compliance Challenges

The four basic security and compliance challenges to organizations are:

- Governance. Cloud computing typically increases an organization's reliance on the cloud providers' logs, reports, and attestations in proving compliance. When companies outsource parts of their IT infrastructure to cloud providers, they effectively give up some control over their information infrastructure and processes, even as they are required to bear greater responsibility for data confidentiality and compliance. While enterprises still get to define how information is handled, who gets access to that information, and under

- what conditions in their private or hybrid clouds, they must largely take cloud providers at their word or their SLA that security policies and conditions are indeed being met and may be forced to compromise to a capability that the provider can deliver. The organization's ability to monitor actual activities and verify security conditions within the cloud is usually very limited and there are no standards or commercial tools to validate conformance to policies and SLAs.[4][12]
- Co-Tenancy and Noisy or Adversarial Neighbors. Cloud computing introduces new risk resulting from co-residency, which is when different users within a cloud share the same physical requirement to run their virtual machines. Creating secure partitions between co-resident VMs has proven challenging for many cloud providers, ranging from the unintentional, "noisy-neighbor" (a workload that consumes more than its fair share of compute, storage or I/O resources, therefore "starving" other virtual tenants on that host), to the deliberately malicious; such as when malware is injected into the virtualization layer, enabling hostile parties to monitor and control any of the VMs residing on a system. Researchers at UCSD and MIT were able to pinpoint the physical server used by programs running on the EC2 cloud and then extract small amounts of data from these programs, by placing their own software there and launching a side-channel attack.[4][20]
- Architecture and Applications. Cloud services are typically virtualized, which adds a hypervisor layer to the traditional IT services stack. This new layer in the service stack introduces opportunities for improving security and compliance, but also creates new attack surfaces and potential exposure to risks. Organizations must evaluate the new monitoring opportunities and the risks presented by the hypervisor layer and account for them in policy definition and compliance reporting.[4][20]
- Data. Cloud services raise access and protection issues for user data and applications, including source code. Who has access, and what is left behind when you scale down a service? How do you protect data from the virtual infrastructure administrators and cloud co-tenants? Encryption of data—at rest, in transit, and eventually in use—would become a basic requirement. But encryption comes with a performance cost. If we truly want to encrypt everywhere, how do we do it cost effectively and efficiently? Finally, one area that is least discussed is "data destruction." There are clear regulations on how long data has to be saved (after which it has to be destroyed) and how to handle data disposal. Examples of these regulations include the Sarbanes-Oxley Act (SOX), Section 802 (7 years)[22], HIPAA, 45 C.F.R. § 164.530(j) (6 years)[17], and FACTA Disposal Rule.[8]

Given that most organizations are using cloud services today for applications that are not mission critical or are of low value, security and compliances challenges seem manageable—but this is a policy of avoidance. These services don't deal with data and applications that are governed by strict information security policies such as health regulations, FISMA regulations, and the Data Protection Act in Europe. The security and compliance challenges mentioned above would become central to cloud providers and subscribers once these higher-value business functions and data begin migrating to private cloud and hybrid clouds, creating very strong requirements for cloud security to provide and prove compliance. Industry pundits believe that cloud value proposition will increasingly drive the migration of these higher-value applications and information and business processes to cloud infrastructures. And as more and more sensitive data and business-critical processes move to cloud environments, the implications for security officers in organizations would be very wide- ranging to provide a transparent and deep compliance and monitoring framework for information security.

Cloud Service Security Requirements

This section addresses the key Security requirements as they pertain to launching Services in a Cloud Infrastructure.

Trust in the Cloud

One of the pillars of security in the cloud is trust.[12] A trusted computing system will consistently behave in expected ways, and hardware and software will enforce these behaviors. Trusted computing uses cryptographic and measurement techniques to help enforce a selected behavior because it authenticates the launch and authorized processes. This authentication allows someone to verify that only authorized code runs on a system. This typically covers initial booting and may also cover applications and scripts. Usually, the establishment of trust of a particular component implies the ability to establish the trust for that component with respect to other trusted components. This trust path is known as the chain of trust, with the first component known as the root of trust.[4][9] It is implied that the root of trust be a trusted set of functions that are immune from physical and other attacks. Since an important requirement for trust is to be tamper-proof, cryptography or some immutable unique signature that identifies a component is used. For example: the hardware platform is usually a good proxy for a root of trust, since for most attackers the risk and difficulty of tampering directly with hardware exceeds the potential benefits. With the use of hardware as the initial root of trust, one can then measure software (such as hypervisor or operating system) to determine whether unauthorized modifications have been made to it. In this way, a chain of trust relative to the hardware can be established.

Trust techniques include hardware encryption, signing, machine authentication, secure key storage, and attestation.[9][6] Encryption and signing are well-known techniques, but these are hardened by the placement of keys in protected hardware storage. Machine authentication provides a user a higher level of assurance, as the machine is indicated as known and authenticated. Attestation provides a means for a third party (also called trusted third party) to affirm that firmware and software that are loaded are correct, true, or genuine. This is particularly important to cloud architectures based on virtualization.

Cloud Governance, Risk, and Compliance

Centralized controls and the specialized expertise of cloud services providers will enable security technologies for the computing infrastructure to be deployed far faster and more efficiently in cloud environments than if those same technologies were deployed in traditional enterprise IT environments. In fact, we believe by the end of this year, cloud providers will be able to introduce the first IaaS clouds built on measured trust environments. These new, more secure clouds will give organizations more flexible, affordable and efficient alternatives for shifting high-value business processes and data into private clouds.

Although not every organization will need the high security afforded by a trusted computing environment, every organization using cloud services could benefit from the vastly improved control and transparency that a measured chain of trust enables. Simply being able to verify conditions in the cloud services stack down through the hypervisor is a huge step forward in providing visibility into actual states and activities within the cloud and in better regulating how cloud resources are managed. Internal and private clouds built on a measured chain of trust will:

- Strengthen an organization’s ability to enforce differentiated policies in private clouds.
- Enhance monitoring for compliance at all layers within the cloud.
- Streamline the auditing process.
- Allow for more flexible usage and billing for secure computing resources.

Hardware Root of Trust: Building Security from the Ground Up

Organizations that are using (or desire to use) cloud services are starting to require cloud service providers to

better secure the hardware layer and provider greater transparency into the system activities within and below the hypervisor.[4][6] This means that cloud providers should be able to:

- Give organizations greater visibility into the security states of the hardware platforms running the IaaS for their private clouds.
- Produce automated, standardized reports on the configuration of the physical and virtual infrastructure hosting customer virtual machines and data.
- Provide policy-based control based on the physical location of the server where the virtual machines are and control the migration of these virtual machines onto acceptable locations based on policy specifications (such as some FISMA and DPA requirements dictate).
- Provide measured evidence that their services infrastructure complies with security policies and with regulated data standards.

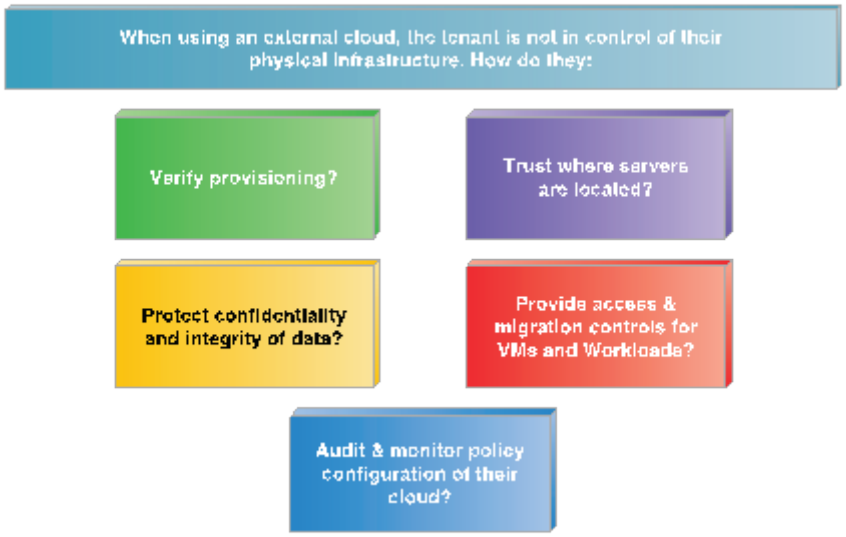


Figure 2 - Summary of the top five trust issues from a cloud subscriber's perspective [17] (Source: orrin, S. Information Security and risk management Conference, 2011 ISaCa, Session 241: Building Trust and Compliance in the Cloud)

What is needed are a set of building blocks for the development of "Trustworthy Clouds." These building blocks can be summarized as [17]:

- Creating a chain of trust rooted in hardware that extends to include the hypervisor.
- Hardening the virtualization environment using known best methods.
- Providing visibility for compliance and audit.

- Using trust as part of the policy management for cloud activity.
- Leveraging infrastructure and services to address data protection requirements.
- Using automation to bring it all together and achieve scale and management efficiency.

Cloud providers and other members of the IT community are preparing to address this need. A growing ecosystem of technology companies is collaborating to develop a new interoperable trusted computing infrastructure. The goal of this emerging infrastructure is to eliminate attacks such as virtual rootkits and to provide the foundation for a hardware root of trust, which establishes a bottom-up security structure based on hardware.

The Advantages of Cloud Services on a Trusted Computing Chain

The advantages of building cloud services on a trusted chain of computing resources include:

- Improving co-residency security by ensuring the launch of only trusted code. Protecting against untrusted software isn't just about malware; it also applies to more benign conditions, such as the improper migration or deployment of virtual machines. To illustrate, if load-balancing software or a cloud administrator attempts to move virtual machines from an unsecured computing platform to a secure, trusted one, the management software would prevent the incoming VMs, since it originated from an unsecured platform.
- Preventing the unsafe transit of secure virtual machines. In the same way that VMs coming from an unsecured platform would not be allowed to move to secured platforms, VMs originating on secured platforms would not be allowed to move to unsecured ones. If, for instance, an administrator attempted to transfer a secured VM onto a new server, the virtualization management console would first perform a policy check on the outgoing VM and then measure the security configurations of the new server against accepted standards. If the new server couldn't meet the secure standards required to host the VM, the virtualization management console or security policy engine would block the VM's move and log the attempt.
- Maximizing operational efficiency by creating trusted pools of systems. Once platform trustworthiness can be measured, cloud providers can put such measurements to use in building trusted pools of systems, all with identical security profiles. Hypervisors can then make more efficient use of secure clouds, moving VMs with similar security profiles within zones of identically secured systems for load balancing and other administrative purposes—all while protecting data in conformance with regulated standards and policies.
- Building secure clouds customized to comply with the most rigorous requirements. The secure cloud's ability to map high-trust zones of systems will enable organizations and cloud providers to customize their clouds to comply specifically with PCI DSS, HIPAA, or other highly controlled information standards. Then, trusted pools of cloud-based resources— all compliant with the same set of information standards—could be dynamically allocated to optimize workloads. Such a scenario would extend the cloud's efficiency and scalability benefits to even the most strictly controlled business processes and heavily regulated industries. Furthermore, cloud services could be fine-tuned to provide different levels of data security. For instance, two clouds could be proven HIPAA-compliant, with one cloud tuned to provide lower-level security at a lower cost for data such as patients' insurance information. The other HIPAA-compliant cloud, handling sensitive health information such as patient medical histories, could be tuned for maximum security. By tailoring cloud service levels, security and pricing to the value of information handled within each cloud, organizations provisioning private clouds can buy only what they need, making the cost benefits and business case for moving into the cloud even more compelling.[REF-1]

Conclusion

The use models we've discussed in this article are early-stage implementations to address requirements that customers and industry bodies are defining now. However, these models do provide a foundation for enhanced security that can evolve with new technologies from Intel and others in the hardware and software ecosystem.

There are no "silver bullets" for security, where a single technology solves all problems—security is too multifaceted for such a simplistic approach. But it is very clear that a new set of security capabilities are needed, and it is best to start at the most foundational elements. Trusted platforms provide such a foundation. Such platforms provide:

- Increased visibility into the operational state of the critical controlling software of the cloud environment through attestation capabilities; and
- A new control point, capable of identifying and enforcing local "known good" configuration of the host operating environment and reporting the resultant launch trust status to cloud and security management software for subsequent use.

Each of these capabilities complements the other as they address the joint needs for visibility and control in the cloud. Of equal importance, these attributes can be available to both consumers of cloud services and the cloud service providers, thanks to common standards for key functions such as attestation, but also due to the work for the ecosystem to enable solutions are many layers. It is only through the integration of trust-based technologies into the virtualization and security management tools in traditional IT environments (tools such as security event information management (SEIM) or governance, risk, and compliance (GRC) console) that will deliver the required scale and seamless management that will help customers realize the benefits of cloud computing.

References

[1] Amazon Web Services, "Overview of Security Processes," August 2010.

[2] Barros, A. and Kylau, U., "Service Delivery Framework—An Architectural Strategy for Next-Generation Service Delivery in Business Network, Proceedings of the 2011 Annual SRII Global Conference, pp. 47–37, 2011.

[3] Demirkan, H., Harmon, R.R., Goul, M., "A Service Oriented Web Application Framework," IT Professional, Vol. 13, no. 5, 15–21, 2011.

[4] Curry S., Darbyshire J, Fisher Douglas, et al., "RSA Security Brief", March 2010.

[5] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing v2.1," 2009.

[6] Cloud Security Alliance Group, "CSA-GRC Stack," , Accessed January 2012

[7] E. Castro-Leon, E., Golden, B., Yeluri, R., Gomez, M., Sheridan, C., Creating the Infrastructure for Cloud Computing: An Essential Handbook for IT Professionals, Chapter 4, Intel Press, 2011.

[8] "FACTA Disposal Rule goes into Effect Jun1 2005," Report on Federal Trade Commission Website. Retrieved February 2012 from http://www. ftc.gov/opa/2005/06/disposal.shtm

[9] Intel Corporation. "Intel TXT," white paper, 2012, retrieved from http://www.intel.com/technolog y/security/downloads/arch-overview.pdf

[10] ISACA (auditor's perspective journal) http://www.isaca.org/Journal/ Past-Issues/2009/Volume-6/Pages/ Cloud-Computing-An-Auditor-s- Perspective1.aspx.

[11] Kang, K., Strong, R., Fang, H., Chen, T., Rhodes, J., Zhou, R., "Complex service management in a hybrid cloud," Proceedings of the 2011 Annual SRII Global Conference, pp. 34–46, 2011.

[12] Khan, K.M, Malluhi, Q., Establishing Trust in Cloud Computing, IEEE IT Professional, Vol. 12-5, pp. 20–27, September/October 2010.

[13] NIST, "Recommended Security Controls for Information Systems," Special Publication 800-31-1, National Institute of Standards and Technology, 2005.

[14] NIST Special Publication 800-53, "Recommended Security Controls for Federal Information Systems and Organizations," revision3, 2010.

[15] NIST Special Publication 800-37 Guide for Applying the Risk Management Framework to Federal Information Systems, 2010.

[16] NIST Special Publication 800-88 Guidelines for Media Sanitization, Computer Security, 2008.

[17] Orrin, S. Information Security And Risk Management Conference, 2011 ISACA, Session 241: Building Trust and Compliance in the Cloud.

[18] Research paper—"Cloud Computing and Security - A Natural Match," http://www.trustedcomputinggroup.org/resources/cloud_computing_ and_security__a_natural_match

[19] Research paper—"Security Issues and Solutions in Cloud Computing" http://wolfhalton.info/2010/06/25/ security-issues-and-solutions-incloud- computing/"

[20] Ristenpart, T., Tromer, E., et al., "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds," CCS'09, Chicago, Illinois, USA.

[21] SOX 802 Final Rule: Retention of Records Relevant to Audits and Reviews. Retrieved February 2012 from http://www.sec.gov/rules/ final/33-8180.htm

[22] Technical Report from RAND Corporation, Europe, "Handbook of Legal Procedures of Computer and Network Misuse in EU Countries, RAND" 2005 edition of the CSIRT Legal Handbook. Retrieved February 2012 from http://www.rand.org/pubs/technical_reports/2006/ RAND_TR337.pdf

[23] Wesam Dawoud, Ibrahim Takouna and Christoph Meinel, "Infrastructure as a service security: Challenges and solutions," in 2010 The 7th International Conference on Informatics and Systems, 2010, pp. 1–8.

[24] Wikipedia—Cloud computing security; Retrieved February 2012 from http://en.wikipedia.org/wiki/Cloud_ computing_security

[25] Yeluri.R, and Moreau. D., "Building Trust and Compliance in the Cloud," VMworld 2011 Technical Session.

[26] Kaufman, L.M., Potter, B., Can Public-Cloud Security Meet Its Unique Challenges? IEEE Security & Privacy, July/August 2010.

[27] Weinberg, N, Gartner: "Private Clouds Are a Last Resort," Network World, October 19, 2011 quoting Garner's Daryl Plummer; retrieved February 25, 2012 from http://www.networkworld.com/ news/2011/101911-gartner-private-cloud-252151.html

[28] Wenbo Mao, Attestation as a Service—Local Attestation for Cloud Security vs. Remote Attestation for Grid Security, EMC Community Network, retrieved February 25, 2012 from https://community.emc.com/ blogs/

WenboMao/2010/04/25/attestation-as-a-service-local-attestation- for-cloud-security-vs-remote-attestation-for-grid-security

[29] Mell, P, Grance, Timothy, The NIST Definition of Cloud Computing, NIST Special Publication 800-145, September 2011.

Copyright

Q4/2013 Workshop Calendar

Certified Cloud Architect

September 30 - October 12, 2013
Johannesburg, South Africa

CloudSchool.com™ CLOUD CERTIFIED Architect

Certified Cloud Architect

October 7-11, 2013
Hong Kong, Hong Kong

CloudSchool.com™ CLOUD CERTIFIED Architect

Cloud Technology Professional

November 4-6, 2013
Paris, France

CloudSchool.com™ CLOUD CERTIFIED Technology Professional

Certified SOA Architect

November 4-8, 2013
Virtual, Pacific Standard Time

SOASchool.com® SOA CERTIFIED Architect

Certified Cloud Architect

November 11-15, 2013
Bangkok, Thailand

CloudSchool.com™ CLOUD CERTIFIED Architect

Cloud Technology Professional

November 8-10, 2013
Wellington, New Zealand

CloudSchool.com™ CLOUD CERTIFIED Technology Professional

Certified SOA Architect

December 2-6, 2013
Toronto, ON, Canada

SOASchool.com® SOA CERTIFIED Architect

Cloud Technology Professional

December 2-4, 2013
Kuala Lumpur, Malaysia

CloudSchool.com™ CLOUD CERTIFIED Technology Professional

SOA Quality Assurance Specialist

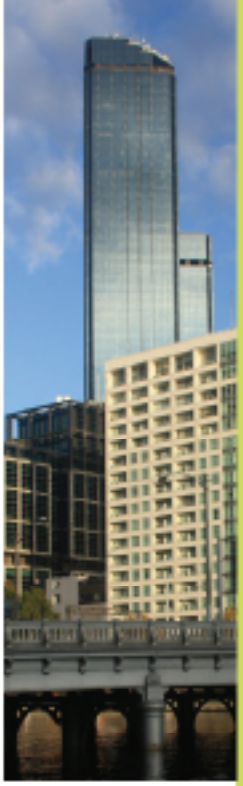
December 9-13, 2013
Vancouver, BC, Canada

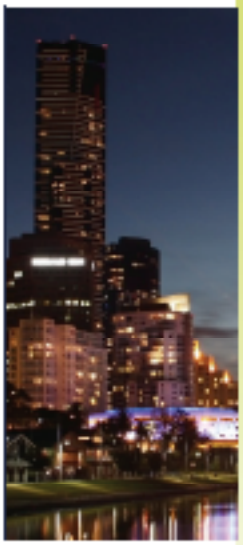
SOASchool.com® SOA CERTIFIED Quality Assurance Specialist

Certified Cloud Architect

December 9-13, 2013
Tokyo, Japan

CloudSchool.com™ CLOUD CERTIFIED Architect





Arcitura
the IT education company

www.arcitura.com/workshops

Cloud Technology Professional Certification
September 17-19, 2013 • Guadalajara, Mexico

SOA Governance Specialist Certification
September 23-27, 2013 • Vancouver, BC, Canada

SOA Architect Certification
September 23-27, 2013 • Munich, Germany

Cloud Architect Certification
September 23-27, 2013 • Milan, Italy

Cloud Architect Certification
September 23-27, 2013 • Dubai, UAE

SOA Architect Certification
September 30 - October 4, 2013 • Toronto, ON, Canada

Cloud Architect Certification
September 30 - October 4, 2013 • São Paulo, Brazil

Cloud Architect Certification
September 30 - October 4, 2013 • Johannesburg, South Africa

Cloud Architect Certification
October 7-11, 2013 • Toronto, ON, Canada

Cloud Architect Certification
October 7-11, 2013 • Singapore, Singapore

Cloud Architect Certification
October 7-11, 2013 • Hong Kong, Hong Kong

Cloud Architect Certification
October 7-11, 2013 • Brasília, Brazil

SOA Architect Certification
October 7-11, 14-18, 2013 • Athens, Greece

SOA Governance Specialist Certification
October 7, 9-10, 16, 22-24, 2013 • Bogotá, Colombia

SOA Consultant Certification
October 7-11, 14, 22-25, 2013 • Athens, Greece

SOA Architect Certification
October 8-13, 2013 • Bogotá, Colombia

SOA Consultant Certification
October 8-10, 15-17, 2013 • Bogotá, Colombia

SOA Analyst Certification
October 8, 10, 15, 17-19, 2013 • Bogotá, Colombia

Cloud Architect Certification
October 14-18, 2013 • Naarden, Netherlands

Cloud Professional Certification
October 19-20, 2013 • Las Vegas, NV, USA

Cloud Architect Certification
October 21-25, 2013 • Sydney, Australia

SOA Architect Certification
October 21-25, 2013 • Singapore, Singapore

Cloud Architect Certification
October 21-25, 2013 • Bangalore, India

SOA Consultant Certification
October 28 - November 1, 2013 • Vancouver, BC, Canada

Cloud Technology Professional Certification
November 4-6, 2013 • Paris, France

SOA Architect Certification
November 4-8, 2013 • PST

Cloud Architect Certification
November 4-8, 2013 • London, UK

SOA Architect Certification
November 4-8, 2013 • London, UK

Cloud Architect Certification
November 11-13, 2013 • Bangkok, Thailand

SOA Architect Certification
November 11-15, 2013 • Singapore, Singapore

Cloud Architect Certification
November 11-15, 2013 • São Paulo, Brazil

SOA Architect Certification
November 11-15, 2013 • Amsterdam, Netherlands

Cloud Technology Professional Certification
November 18-20, 2013 • Wellington, New Zealand

Cloud Architect Certification
November 18-22, 2013 • Singapore, Singapore

Cloud Architect Certification
November 18-22, 2013 • Las Vegas, NV, USA

Cloud Technology Professional Certification
November 19-21, 2013 • Guadalajara, Mexico

Cloud Professional Certification
November 23-24, 2013 • Toronto, ON, Canada

Cloud Technology Professional Certification
November 25-27, 2013 • Naarden, Netherlands

SOA Architect Certification
November 25-29, 2013 • Las Vegas, NV, USA

SOA Architect Certification
November 25-29, 2013 • Berlin, Germany

Cloud Architect Certification
November 25-29, 2013 • Dubai, UAE

Cloud Technology Professional Certification
December 2-4, 2013 • Kuala Lumpur, Malaysia

SOA Architect Certification
December 2-6, 2013 • Toronto, ON, Canada

Cloud Architect Certification
December 2-6, 2013 • Singapore, Singapore

Cloud Architect Certification
December 2-6, 2013 • São Paulo, Brazil

SOA Quality Assurance Specialist Certification
December 9-13, 2013 • Vancouver, BC, Canada

Cloud Architect Certification
December 9-13, 2013 • Tokyo, Japan

SOA Architect Certification
December 9-13, 2013 • Singapore, Singapore

SOA Architect Certification
December 9-13, 2013 • CET

Cloud Architect Certification
December 16-20, 2013 • PST

Updated: June 28, 2013

Contributors

Sudhir S. Bangalore

Sudhir S Bangalore is a Senior Systems Engineer in Intel Architecture and Systems Integration (IASI) group, which is part of Intel Architecture Group, and is focused on developing solutions to enable virtualization and cloud security, with focus on Intel Architecture and associated ingredients. He is responsible for understanding enterprise and data center needs, developing reference implementations and innovative solutions to meet these needs with Intel technologies. Prior to this role, he has worked as a architect and a key engineer on Intel's Enterprise Access Management framework and implementation. Sudhir has a Master's degree in Computer Science, and has been with Intel for more than 10 years.

This article and more on similar subjects may be found in the Intel Technology Journal, Volume 16 issue 4 "End to End Cloud Computing". More information can be found at http://noggin.intel.com/technology-journal/2012/164/end-to-end-cloud-computing

Contributions

- Service Security and Compliance in the Cloud - Part I

Thomas Erl

Thomas Erl is a best-selling IT author and founder of CloudSchool.com™ and SOASchool.com®. Thomas has been the world's top-selling service technology author for over five years and is the series editor of the Prentice Hall Service Technology Series from Thomas Erl (www.servicetechbooks.com), as well as the editor of the Service Technology Magazine (www.servicetechmag.com). With over 175,000 copies in print world-wide, his eight published books have become international bestsellers and have been formally endorsed by senior members of major IT organizations, such as IBM, Microsoft, Oracle, Intel, Accenture, IEEE, HL7, MITRE, SAP, CISCO, HP, and others.

Four of his books, Cloud Computing: Concepts, Technology & Architecture, SOA Design Patterns, SOA Principles of Service Design, and SOA Governance, were authored in collaboration with the IT community and have contributed to the definition of cloud computing technology mechanisms, the service-oriented architectural model and service-orientation as a distinct paradigm. Thomas is currently working with over 20 authors on several new books dedicated to specialized topic areas such as cloud computing, Big Data, modern service technologies, and service-orientation.

As CEO of Arcitura Education Inc. and in cooperation with CloudSchool.com™ and SOASchool.com®, Thomas has led the development of curricula for the internationally recognized SOA Certified Professional (SOACP) and Cloud Certified Professional (CCP) accreditation programs, which have established a series of formal, vendor-neutral industry certifications.

Thomas is the founding member of the SOA Manifesto Working Group and author of the Annotated SOA Manifesto (www.soa-manifesto.com). He is a member of the Cloud Education & Credential Committee, SOA Education Committee, and he further oversees the SOAPatterns.org and CloudPatterns.org initiatives, which are dedicated to the on-going development of master pattern catalogs for service-oriented computing and cloud computing.

Thomas has toured over 20 countries as a speaker and instructor for public and private events, and regularly participates in international conferences, including SOA, Cloud + Service Technology Symposium and Gartner events. Over 100 articles and interviews by Thomas have been published in numerous publications, including the Wall Street Journal and CIO Magazine

Jürgen Kress

An eleven year Oracle veteran, Jürgen works at EMEA Alliances and Channels. As the founder of the Oracle WebLogic and SOA Partner Community and the global Partner Advisory Councils. With more than 4,000 members from all over the world the SOA and WebLogic Partner Communities are the most successful and active communities at Oracle. Jürgen hosts the communities with monthly newsletters, webcasts and conferences. He hosts his Fusion Middleware Partner Community Forums, where more than 200 partners get the latest product updates, roadmap insights and hands-on trainings. Supplemented by many web 2.0 tools like twitter, discussion forums, online communities, blogs and wikis. For the SOA & Cloud Symposium by Thomas Erl Jürgen was a member of the steering board. He is also a frequent speaker at conferences like the SOA & BPM Integration Days, Oracle Open World or the JAX.

Contributions

- Enterprise Service Bus
- Canonizing a Language for Architecture: An SOA Service Category Matrix
- Industrial SOA
- SOA Blueprint: A Toolbox for Architects

Berthold Maier

Berthold Maier works in the T-Systems International department of Telekom Germany as Enterprise Architect. He has more than 19 years experience as developer, coach and architect in the area of building complex mission critical applications and integrations scenarios. Within eleven years as Oracle employee he has held several leading positions including chief architect in the consulting organization. Hi is the founder of many frameworks and take over the responsible for reference architectures around BPM/SOA and Enterprise Architecture Management. Berthold is also well-known as a conference speaker, book author and magazine writer.

Contributions

- Enterprise Service Bus
- Canonizing a Language for Architecture: An SOA Service Category Matrix
- Industrial SOA
- SOA Blueprint: A Toolbox for Architects

Hajo Normann

Hajo Normann works for Accenture in the role of SOA & BPM Community of Practice Lead in ASG. Hajo is responsible for the architecture and solution design of SOA/BPM projects, mostly acting as the interface between business and the IT sides. He enjoys tackling organizational and technical challenges and motivates solutions in customer workshops, conferences, and publications. Hajo leads together with Torsten Winterberg the DOAG SIG Middleware and is an Oracle ACE Director and an active member of a global network within Accenture, as well as in regular contact with SOA/BPM architects from around the world.

Contributions

- Enterprise Service Bus
- Canonizing a Language for Architecture: An SOA Service Category Matrix
- Industrial SOA
- SOA Blueprint: A Toolbox for Architects

Manuel Rosa

Manuel Rosa began his career participating in Enterprise Architecture projects in Portugal, Brazil and Luxembourg, mainly focusing on Application and Business Architectures. In the past few years, Manuel has coordinated the development of SOA Governance programs and has also been responsible for various SOA Governance projects in major Portuguese enterprises in the telecom and utilities industries, as well as SOA projects in Spain. He is currently SOA Governance Practice Leader at Link Consulting.

Manuel was a speaker at the 5th International SOA, Cloud + Service Technology Symposium in London. He is currently a teaching assistant on the subjects of Enterprise Architecture I and II for the POSI postgraduate degree provided by Intituto Superior Técnico, INESC and INOV, in a one-year program.

Contributions

- Promoting Organizational Visibility for SOA and SOA Governance Initiatives

André Sampaio

André de Oliveira Sampaio (MSc), is a Senior Consultant of Enterprise Architecture for Link Consulting in Portugal, and has participated in and coordinated various Enterprise Architecture projects in Portugal, Brazil and Luxembourg. As a professional, André has been involved in several major transformation projects, most of which in the financial services and public administration sectors.

André is responsible for the development of the Enterprise Architecture Management System (www.link.pt/eams), and has published work centered around the themes of Enterprise Architecture, Enterprise Transformation, System Theory, Service-Oriented Architecture and Formal Viewpoints.

Contributions

- Promoting Organizational Visibility for SOA and SOA Governance Initiatives

Danilo Schmiedel

Danilo Schmiedel is one of the leading BPM and SOA System Architects at OPITZ CONSULTING. He has been involved in large integration-, business processes automation and BPM / SOA development projects where he implemented solutions for various customers. His main field of interest is focused on the practical use of BPM and SOA on a large scale. Additionally he works as BPM and SOA project coach. Danilo is a frequent speaker in the German Java and Oracle communities and has written numerous articles about the above topics. Before joining OPITZ CONSULTING Danilo worked as Software Engineer in several international projects. The Leipzig University of Applied Science has awarded his outstanding reputation in 2009.

Contributions

- Enterprise Service Bus
- Canonizing a Language for Architecture: An SOA Service Category Matrix
- Industrial SOA
- SOA Blueprint: A Toolbox for Architects

Guido Schmutz

Guido Schmutz works as Technology Manager for the IT services company Trivadis. He has over 25 years as a software developer, consultant, architect, trainer, and coach. In Trivadis he is responsible for SOA, BPM and application integration, and is head of the Trivadis Architecture Board. His interests lie in the architecture, design, and implementation of advanced software solutions. He specializes in Java EE, Spring, Oracle SOA Suite and Oracle Service Bus. He is a regular speaker at international conferences and is the author of articles and several books. Guido is an Oracle ACE Director for Fusion Middleware & SOA.

Contributions

- Enterprise Service Bus
- Canonizing a Language for Architecture: An SOA Service Category Matrix
- Industrial SOA
- SOA Blueprint: A Toolbox for Architects

Bernd Trops

Bernd Trops is a Senior Principal Consultant at Talend Inc. In this role he is responsible for client project management and training.

Bernd is responsible for all Talend projects within the Deutsche Post and the introductions of new versions and components.

Before Talend, Bernd was a Systems Engineer working on various projects for GemStone, Brocade and WebGain and therefore has extensive experience in J2EE and SOA. From 2003 to 2007 Bernd Trops worked as a SOA Architect at Oracle.

Contributions

- Enterprise Service Bus
- Canonizing a Language for Architecture: An SOA Service Category Matrix
- Industrial SOA
- SOA Blueprint: A Toolbox for Architects

Clemens Utschig-Utschig

Clemens worked as Chief Architect for the Shared Service Centre, Global Business Services, Boehringer Ingelheim in architecture, master data, service management and innovation.

At the moment he works with holistic enterprise architecture that provides the methodological platform for the new master data management.

He previously worked as a Platform Architect at Oracle Inc. in the United States, where he helped to develop next product strategy as well as the SOA BPM Suite.

Contributions

- Enterprise Service Bus
- Canonizing a Language for Architecture: An SOA Service Category Matrix
- Industrial SOA
- SOA Blueprint: A Toolbox for Architects

Torsten Winterberg

Torsten Winterberg works for Oracle Platinum Partner OPITZ CONSULTING. As a director of the competence center for integration and business process solutions he follows his passion to build the best delivery unit for customer solutions in the area of SOA and BPM. He has long-time experience as developer, coach and architect in the area of building complex mission critical Java EE applications. He is a known speaker in the German Java and Oracle communities and has written numerous articles on SOA/BPM related topics. Torsten is part of the Oracle ACE director team (ACE=Acknowledged Community Expert) and leads the DOAG middleware community.

Contributions

- Enterprise Service Bus
- Canonizing a Language for Architecture: An SOA Service Category Matrix
- Industrial SOA
- SOA Blueprint: A Toolbox for Architects

Raghu Yeluri

Raghu Yeluri is a Principal Engineer in the Intel Architecture Group at Intel with focus on virtualization, security and cloud architectures. He is responsible for understanding enterprise and data center needs, developing reference architectures and implementations aligned with Intel virtualization, security and cloud related platforms and technologies. Prior to this role, he has worked in various architecture and engineering management positions in systems development, focusing on service-oriented architectures in engineering analytics and information technology. He has multiple patents and publications, and has co-authored an Intel Press book on Cloud Computing – "Building the Infrastructure for Cloud Computing, an essential handbook for IT Professionals".

Contributions

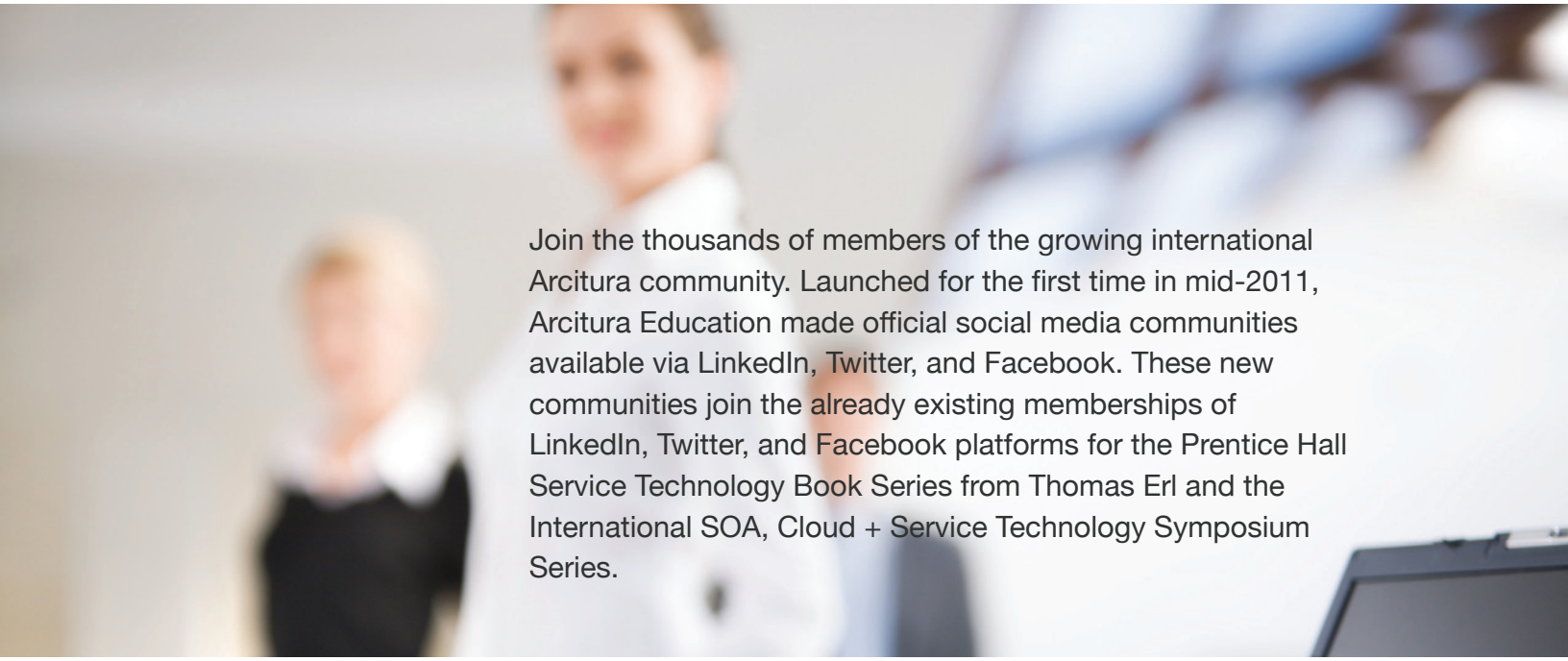
- Service Security and Compliance in the Cloud - Part I

Arcitura IT Certified Professionals (AITCP) Community

Arcitura

the IT education company

www.arcitura.com/community



Join the thousands of members of the growing international Arcitura community. Launched for the first time in mid-2011, Arcitura Education made official social media communities available via LinkedIn, Twitter, and Facebook. These new communities join the already existing memberships of LinkedIn, Twitter, and Facebook platforms for the Prentice Hall Service Technology Book Series from Thomas Erl and the International SOA, Cloud + Service Technology Symposium Series.

LinkedIn twitter facebook YouTube

SERVICE TECHNOLOGY MAGAZINE

The Service Technology Magazine is a monthly online publication provided by Arcitura Education Inc. and officially associated with the "Prentice Hall Service Technology Book Series from Thomas Erl." The Service Technology Magazine is dedicated to publishing specialized articles, case studies, and papers by industry experts and professionals in the fields of service-oriented architecture (SOA), cloud computing, semantic Web technologies, and other areas of services-based technology, innovation, and practice.

www.servicetechnologymagazine.com

www.servicetechmag.com