THE Open GROUP

Approaches to Boundaryless Information Flow Architecture

Report from the Washington Conference

Version 1.0

A Working Paper of the Boundaryless Information Flow Reference Architecture Initiative by:

Eliot Solomon
Principal, Eliot M. Solomon Consulting, Inc.

January 2004

Approaches to Boundaryless Information Flow Architecture

Published by The Open Group, January 2004.

Any comments relating to the material contained in this document may be submitted to:
The Open Group
44 Montgomery St. #960
San Francisco, CA 94104

or by email to:

ogpubs@opengroup.org

Contents

Table of Figures

Boundaryless
Information Flow
™

Boundaryless Information Flow™
achieved through global interoperability
in a secure, reliable, and timely manner

Executive Summary

In 2002, The Open Group updated its vision to be "Boundaryless Information Flow achieved through global interoperability in a secure, reliable, and timely manner". In January 2003, a White Paper entitled "Boundaryless Information Flow Reference Architecture" was produced to help the membership of The Open Group engage in efforts to achieve this vision. That document presented a framework in which to elicit specific contributions.

In order to focus work on specific and actionable business objectives, a second White Paper was written entitled "Boundaryless Information Flow Reference Architecture: Six Example Boundaryless Business Models". It described six business models of Boundaryless Information Flow including its business objectives, the constraints under which it must be accomplished, and other relevant considerations. A "Call For Boundaryless Information Flow Architecture Papers" was issued to solicit IT technology and solutions providers to offer their solutions (specific technologies, standards, products, and architectural elements) to the problems described in the six business models.

This Report

This Report reviews the papers and presentations that were submitted in response to the Call for Papers. The Open Group received eight responses – from IBM, Fujitsu, Hewlett-Packard, The Open Group Directory Interoperability Forum (DIF), RSA Security, CyberRave, Red Hat, and Architecting-The-Enterprise – of which six were selected for presentation to the Washington Conference. In this Report the author looks at commonalities and differences, and begins to draw conclusions about the future of IT Architecture in general, and the Reference Architecture for Boundaryless Information Flow in particular. The views expressed in this Report are solely the views of the author.

Introduction

Background

The Open Group solicited papers and presentations describing architectural approaches to Boundaryless Information Flow

The Open Group solicited papers and presentations from vendors and other technology providers by issuing its "Call For Boundaryless Information Flow Architecture Papers". In response, the vendors and providers were asked to describe architectural approaches addressing one or more of the six business models of Boundaryless Information Flow presented in "Boundaryless Information Flow Reference Architecture: Six Example Boundaryless Business Models" (available from The Open Group web site). Responses were particularly encouraged which described approaches to achieving Boundaryless Information Flow based on available technology, or technologies that will be available in the near term.

In response to the Call For Papers, eight submissions were received. Six of those were selected for presentation at the Washington Conference. The submissions all addressed issues associated with Boundaryless Information Flow. Some introduced specific architectures, and some addressed architectural methodology. Most addressed, to a greater or lesser degree, the characteristics an IT Architecture should have in order to achieve the objectives of Boundaryless Information Flow.

Material Presented in this Report

The six submissions that were accepted included three that described integration architectures, one that discussed a common system architecture, and two that described solution architectures. (These terms are explained in more detail below.) Each submission included a set of slides; most of them included Proposal Papers or White Papers. The slides and papers are available on The Open Group web site.

In this Report we provide an overview of the six submissions, with brief descriptions of two other submissions that were not accepted for presentation at the Washington Conference. For each of the six submissions (and one other, a scenario) there is a section providing a more detailed analysis.

Overview of Washington Presentations

This section gives an overview of each presentation made at the Washington Conference.

The presentations can be grouped into three categories: Integration Architectures, Common System (Subsystem) Architectures, and Solution Architectures. The presentations are described here using those categories.

Some of the presentations covered more than a single type of architecture and, where applicable, a cross-reference is provided. Each presentation is described in more detail in a later part of this document.

Integration Architectures

Integration Architectures are architectures or architectural approaches for constructing complete system architectures

Three of the presentations described integration architectures. These are architectures or architectural approaches for constructing complete system architectures and, more often, enterprise or larger systems. It is fair to say that any well-conceived, flexible enterprise-scale approach to IT Architecture will contribute to Boundaryless Information Flow by including the entire enterprise within a single architectural model. But truly enabling boundarylessness within an enterprise requires more than an architecture. As we all know, there are many barriers to the successful accomplishment of boundarylessness, even with a comprehensive architectural model. Each of the presented integration architectures included a “twist” or particular characteristic or capability that may be used to help break through the obstacles to boundarylessness.

IBM's e-business on-Demand Architecture

The goal of IBM’s on-Demand Architecture is to eliminate the boundaries created by the physical artifacts of IT. It starts by “virtualizing” the platform by making the physical extent of a computer transparent (or invisible) to all software residing above it. Virtualization is accomplished using a model that posits three contributing services: virtualization, integration, and automation.

The ability to use available computer resources on-demand, wherever they may be, removes one of the main organizational rationales for maintaining boundaries between IT “stacks”: the notion of ownership of computers as fixed capital resources. In a fully implemented on-demand architecture, computers are not fixed objects. Perhaps more important, computing resource need not be a capital item for corporate lines of business. Computing resources can be provided centrally in any of a number of financial models that allow them to be treated as an expense to the line of business.

Fujitsu's Triole Architecture

Fujitsu presented their Triole Architecture, which is based on a model of service-oriented integration. It features a specific approach to constructing subsystems (based on a modeling technique called Platform Integration (PI) Templates) and integrating them into the larger architecture. Subsystems are constructed by combining hardware and software offerings, possibly from diverse vendors, into tightly integrated packages that offer well-defined service interfaces to the external environment, but hide the interior construction of the subsystem.

Benefits claimed for this service-oriented integration approach include: enabling existing systems to adapt quickly to changes in the customer's business; helping develop an optimum IT system by allowing utilization of existing IT assets; and allowing a rapid restructuring and integration of business operations.

Hewlett-Packard's Darwin Reference Architecture

HP's Darwin approach to system architecture emphasizes the incorporation into the system of features or characteristics that will allow the system or architecture to evolve in response to changes in the business (or other external) environment. It uses as one of its central abstractions a model that shows "People", "IT", and "Other Resources" composing the platform on which "Business" and its strategy and processes are constructed.

While Darwin includes a low-level platform of virtualized services, and includes service-oriented abstractions, these elements are subservient to the primary abstraction of business and the objective of building an adaptive enterprise.

"*The HP Darwin Reference Architecture provides a framework to continuously balance business demand for IT services against resources supplied by the infrastructure. Adaptiveness is built-in through consistent simplification, standardization, integration, and modularity of the elements and aspects of the feedback loop*."

Common System Architectures

Common systems include:
- *Messaging*
- *Information (Data)*
- *Management*
- *Directory and Location*
- *Network and Communications*
- *Transaction Management*
- *User Interface and Ontology*
- *Security*
- *System and Network Mgmt.*

Common system architectures describe the elements, rationale, construction, and similar aspects of common systems. (Common systems are described in TOGAF.) Examples of common system architectures are system management, information management, and security. An architecture for one of these systems might include descriptions of systems that specifically implement aspects of the common system (alert management system, database, or authentication server, respectively, for the example common systems). They might describe how systems of the common system are linked to each other (for example, a management framework, a distributed or replicated data management system, a public key infrastructure). And they may describe how a system that is not part of the common system takes advantage of the services of the common system (how to report manageable events, how to construct queries or updates, how to check that a transaction is fully authorized).

Common system architectures are intimately related to certain "views" of the entire system or enterprise architecture. For instance, the management view, information management view, or security view of an enterprise architecture will feature or relate to the relevant common system architecture.

- Compare Common System Architectures to Solution Architectures, below.

DIF Directories for Boundaryless Information Flow

The Directory Interoperability Forum (DIF) provided a thoughtful analysis of several ways in which directory services can contribute to achieving boundaryless business objectives. The analysis begins with the assumption that every enterprise has its own identity management practice, relating to who can access which systems, services, and information, and under what circumstances they can do so. The differences range from the role of explicit corporate policy in guiding identity management, through the sorts of relationships a firm has with trading partners, to detailed assignment of specific identity management tasks to organizations or individuals within the firm.

The presentation from the DIF described a variety of such driving factors, and discussed their significance in creating an identity management architecture for Boundaryless Information Flow. The theory is illustrated by examples addressing several of the specific boundaryless business models that were formulated for use at the Washington Conference.

Other Subsystem Architectures

Some of the other presentations included or suggested common system architectures.

- IBM's on-Demand Architecture includes important parts of an information management architecture and a system management architecture.
- RSA's discussion of identity management describes an access management architecture, which is part of the security common system architecture.

Solution Architectures

Similar in scale or scope to common system architectures, solution architectures are created to reflect coherent chunks of business functionality.

Solution architectures solve specific problems that can contribute to a system, organization, enterprise, or larger architecture. Solution architectures are typically built on multiple common system architectures. They are similar in scale or scope to common system architectures, but rather than being created to reflect coherent "chunks" of technology, solution architectures are created to reflect coherent chunks of business functionality.

RSA Identity Management

RSA discussed the key role that identity plays in business. In their presentation they identified the challenge of sharing information about identity across enterprise boundaries as one of the major barriers to achieving truly Boundaryless Information Flow.

RSA's approach strongly encourages migration from centralized models of identity management that may have been developed as part of so-called "single sign-on" initiatives toward federated identity management models. RSA particularly advocates the Liberty Alliance's emerging models and technologies.

CyberRave's Remote Access Virtual Environment

CyberRave described a virtual environment that can be created within or for a community of interest to allow remote access to information shared among members of the community. The Remove Access Virtual Environment (RAVE) is modeled after Private Virtual Networks (PVNs), at least with respect to the operational, trust, or user experiential aspects.

CyberRave's model includes a business model (actually a social or societal model) that supports and enables the effective operation of the technical implementation. This is reflected in CyberRave's mission: "To provide Communities-of-Interest (COI) with ever-increasing levels of information security, actionable intelligence, and simplified access to remote resources by establishing Vertical Communities governed by democratic online Advisor Groups (VCAG)."

Other Solution Architectures

- IBM's on-Demand Architecture describes a high-performance computation architecture.

Additional Submissions

Red Hat's Open Source as a Boundaryless Business Model

Red Hat's submission addressed the challenge of business transformation. In some cases, businesses seek to transform themselves because their goal is to become boundaryless. Other companies value boundarylessness as a means to a business end, which is the company's goal. In some cases, the drive to eliminate the cost of supporting IT systems and infrastructures with vestigial boundaries results in a transformation of the business. Red Hat holds up the Open Source software development model as an exemplary boundaryless organization model. They explained that Open Source development is not only a boundaryless business, but that engaging in Open Source can actually act as the catalyst for a company's boundaryless transformation.

Architecting-The-Enterprise Methodology

Architecting-The-Enterprise submitted a paper on how an organization's IT architects and designers can approach or address the challenge of designing systems with Boundaryless Information Flow as an objective. It included discussion of:

- Skills required in the IT community to understand the responsibilities of architecture and design in this new context
- Skills required within the business community to use and manage the flow of information
- The notion of cross-enterprise architecture or mega-architecture as a distinct model (different from federated or multiple enterprise architectures) as a necessary approach or skill set

Architecting-The-Enterprise will offer their complete presentation at a future date.

Integration Architectures

This section takes a more detailed look at integration architectures.

Three of the presentations covered comprehensive or integration architectures:

- IBM's on-Demand Architecture
- Fujitsu's Triole Architecture
- Hewlett-Packard’s Darwin Reference Architecture

While each of these architectures has many dimensions, each has a primary emphasis that is different from the others:

- on-Demand focuses on virtualizing the infrastructure.
- Triole focuses on quality integration of services to platforms.
- Darwin emphasizes the ability to align and evolve IT configurations to meet the needs of an adaptive enterprise.

There are a number of common themes in all of these, including staples of The Open Group such as standards and easy interoperability. But perhaps the most significant common theme is “virtualization”. This may represent the influence of grid computing as an emerging paradigm; or it might be the convergence of distributed computing and the Internet into web services. Or both. Whatever the genesis of this theme, its ability to facilitate Boundaryless Information Flow is obvious and compelling.

e-business on-Demand: The Next Phase of IT

Introduction

The goal of IBM’s on-Demand Architecture is to eliminate the boundaries created by the physical artifacts of IT. It starts by virtualizing the platform by making the physical extent of a computer transparent (or invisible) to all software residing above it. Virtualization is accomplished using a model that posits three contributing services: virtualization, integration, and automation.

The ability to use available computer resources on-demand, wherever they may be, removes one of the main organizational rationales for maintaining boundaries between IT “stacks”: the notion of ownership of computers as fixed capital resources. In a fully-implemented on-demand architecture, computers are not fixed objects. Perhaps more important, computing resource need not be a capital item for corporate lines of business. Computing resources can be provided centrally in any of a number of financial models that allow them to be treated as an expense to the line of business.

Further Analysis

IBM’s on-Demand Architecture is based on a horizontal platform architecture that aims to create a comprehensive, ubiquitous, seamless environment of compute capability. Its intent is to allow any application or service to be run anywhere at any time, with the appropriate computing resource. (Contrast this with Fujitsu’s Triole Architecture that integrates underlying compute capability with specific overlying services.) In addition to the platform architecture, the on-Demand model describes overlying and related architectures. It encompasses (at least by implication) a comprehensive integration architecture based on virtualization and delocalization.

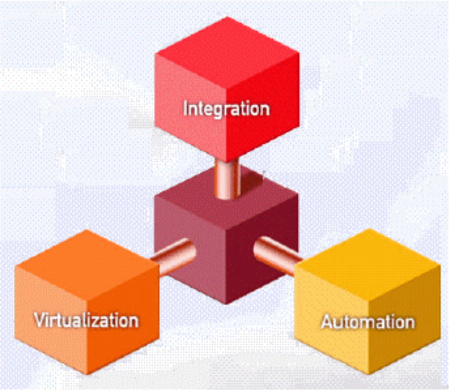


Figure 1: IBM's on-Demand Architecture

Elements of the on-Demand Architecture

As already described, the on-Demand platform architecture is based on three specific subsystems or services: integration, automation, and virtualization. These are linked by a set of shared components that are used by the three services. The shared components are also used to manage or unify the services.

Integration: IBM defines integration in this context as: “The efficient and flexible combination of resources to optimize operations across and beyond the enterprise; it is about people, processes, and information.” In terms of boundarylessness, integration represents the elimination of boundaries between physical resources; IBM (appropriately) identifies this process as applying, both as a prerequisite and as a consequence, to the business organizations and processes that use those physical resources.

Automation: Automation is defined as: “The capability to reduce the complexity of management to enable better use of assets, improve availability and resiliency, and reduce costs based on business policy and objectives.” One of the major difficulties in integrating physical resources that are to be used by different organizations is that of ensuring that resources are managed effectively and fairly for the various organizations' use. In part, the impediment is each organization's fear that it will not get the resources it needs (and may have paid for) in an appropriately timely manner. In part, the concern is that too-broad integration will not contain system faults or failures, and weaken efforts to achieve business continuity and reduce the risk of IT failures. To address both of these concerns, the management of virtualized resources must address not only their provisioning to the appropriate users, but also fault containment, contingency, and recovery.

Virtualization: “Provides a single, consolidated view of and easy access to all available resources in a network – no matter where the data resides.” This is what makes it possible for users of the IT resources to take advantage of the benefits of boundarylessness. Virtualization hides the residual boundaries or discontinuities between IT resources. This includes not only the physical boundaries between instances of like resources, but should also include the hiding of “type” boundaries, as the boundaries between different operating systems or different databases. This hiding of both physical and type boundaries makes possible the elimination of “time” boundaries: the boundaries between the IT that was in place yesterday and the IT that will be in place tomorrow. From a business perspective, elimination of these boundaries can make the notion of “legacy” systems obsolete, allowing businesses to identify IT systems only as those that serve those businesses well, those that are no longer needed, and those that need to be improved. “Length of service” won't be a disability for a successful IT system.

Included, derived, and implied architectures

The on-Demand Architecture may be viewed as a comprehensive integration architecture based on a ubiquitous platform architecture. It also includes a number of sub-architectures or derived architectures, some of which correspond to specific common system architectures and solution architectures of The Open Group Boundaryless Information Flow Reference Architecture Initiative.

Fujitsu's Triole Architecture

Introduction

Fujitsu presented their Triole Architecture, which is based on a model of service-oriented integration. It features a specific approach to constructing subsystems (based on a modeling technique called Platform Integration (PI) Templates) and integrating them into the larger architecture. Subsystems are constructed by combining hardware and software offerings, possibly from diverse vendors, into tightly integrated packages that offer well-defined service interfaces to the external environment, but hide the interior construction of the subsystem.

Benefits claimed for this service-oriented integration approach include: enabling existing systems to adapt quickly to changes in the customer's business; helping develop an optimum IT system by allowing utilization of existing IT assets; and allowing a rapid restructuring and integration of business operations.

Further Analysis

Applicability to Boundaryless Information Flow

The Triole Architecture is an integration architecture that is applicable in any context. While it is no more applicable in a boundaryless model than in any other, it does make important contributions to the ability to construct a boundaryless system.

Triole is a platform-oriented integration architecture that explicitly considers service boundaries at the platform level. (Contrast this to IBM's on-Demand Architecture, which addresses a ubiquitous platform independent of service boundaries, but see also the discussion of the abstract Triole model below.) The defining artifact of Triole's architectural methodology is the PI Template. Illustrated at the left, the PI Template is used to describe with a substantial degree of specificity the construction of a service-oriented application as a vertically integrated platform.

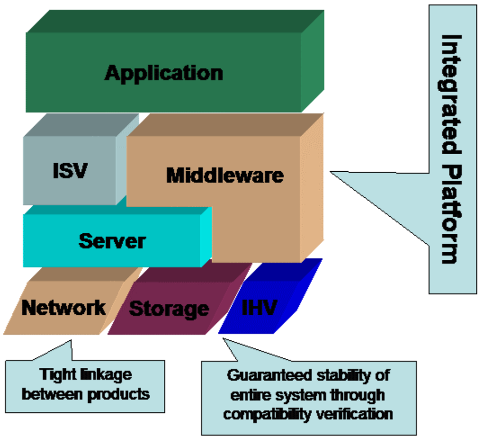
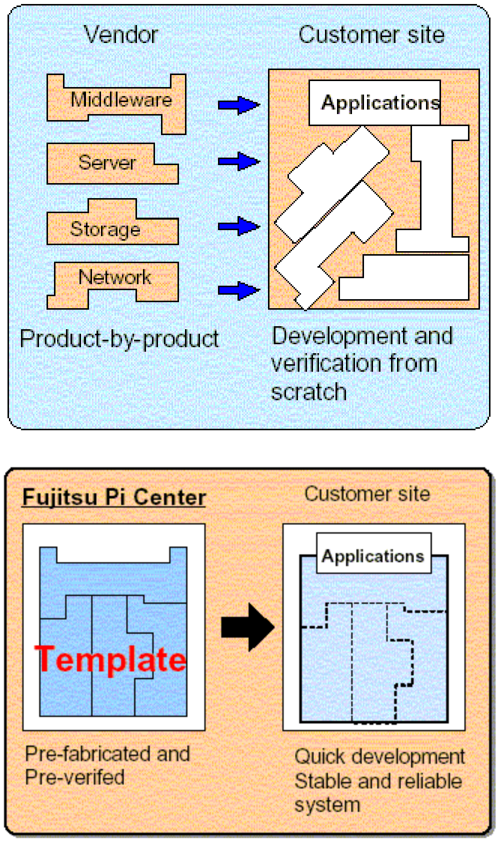

Application
ISV
Middleware
Server
Network
Storage
IHV
Integrated Platform
Tight linkage between products
Guaranteed stability of entire system through compatibility verification

Figure 2: Fujitsu's PI Template



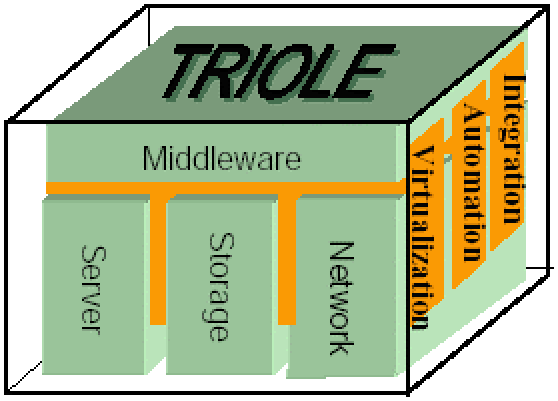
Using the PI Template

The top-left illustration shows Fujitsu's view of the current state of enterprise systems integration. The integration of various IT elements – including hardware, software, networking, and other types of element – from a variety of vendors is left to the customer. Fujitsu's contention is that this leads to inefficient or suboptimal integration.

In the Triole model (represented by the lower-left illustration), integration is based on a PI Template and accomplished in (or at least with the support of) the Fujitsu Platform Integration Center.

This seems at first to run counter to some of the objectives of The Open Group and the goals of the Boundaryless Information Flow Reference Architecture Initiative. (These include simplifying integration for the end user by standards and guaranteed interoperability, and increasing the purchaser's flexibility and choice, avoiding vendor lock-in.) However, this business model and its approach, if used as a Boundaryless Information Flow architectural model, would presumably be replicable and extensible. The processes and procedures for creating and working with PI Templates would be available to end users who choose to create the competence necessary to undertake that work.

Figure 3: Enterprise Systems Integration with and without PI Templates



An abstract view of Triole

While the largest emphasis of the Fujitsu presentation focused on the PI Template model of service-oriented integration, several other themes were also presented. The illustration on the left shows an abstract view of the Triole Architecture.

Notice the inclusion of the concepts of integration, automation, and virtualization. This is a key element of IBM's on-Demand Architecture. (See the analysis of the IBM model for a more complete discussion of this.)

Figure 4: Fujitsu's Triole Architecture

Hewlett-Packard's Darwin Reference Architecture

> *"The Darwin Reference Architecture is based on the assumption that an enterprise is a set of business processes that link customers to the company, employees to one another, and often customers and suppliers to all of them."*
>
> *Nora Denzel, Senior Vice President, Adaptive Enterprise in "The Reality of the Adaptive Enterprise"*

Introduction

HP's Darwin approach to system architecture emphasizes the incorporation into the system of features or characteristics that will allow the system or architecture to evolve in response to changes in the business (or other external) environment. It uses as one of its central abstractions a model that shows "People", "IT", and "Other Resources" composing the platform on which "Business" and its strategy and processes are constructed.

While Darwin includes a low-level platform of virtualized services and includes service-oriented abstractions, these elements are subservient to the primary abstraction of business and the objective of building an adaptive enterprise.

"*The HP Darwin Reference Architecture provides a framework to continuously balance business demand for IT services against resources supplied by the infrastructure. Adaptiveness is built in through consistent simplification, standardization, integration, and modularity of the elements and aspects of the feedback loop.*"

Further Analysis

The Darwin Reference Architecture is a standards-based framework that leverages technology and components from HP and industry partners to create a new level of business and IT integration and lower IT acquisition and operating costs. The Darwin Reference Architecture is based on the premise that all components of the enterprise architecture should adhere to the following design principles:

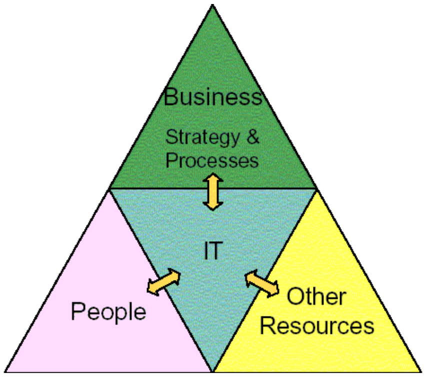


Figure 5: HP's Darwin Reference Architecture

Simplification: Simplify existing IT environments through consolidation of underutilized assets and by ensuring that management and control levers exist at all the layers of the environment – infrastructure, applications, and business processes. An infrastructure that contains fewer elements is easier to manage and therefore can deliver results faster and more easily, especially when executing changes.

Standardization: Standards – applied across processes, procedures, technologies, and applications – extend the benefits of simplification. According to the Darwin Reference Architecture, standardization can be achieved in many ways including: adopting industry standard interfaces, which reduce communications overhead and speed change; establishing common processes and policies for managing change; and defining common requirements for manageability, security, collaboration, configuration management, capacity, and performance management.

Modularity: Modularity in the context of the Darwin Reference Architecture applies both to physical networks of storage and servers and to the virtual resources they support. Modularity allows one aspect of a system to be changed without impacting any on the other components, leading to improved manageability and responsiveness. With modularity, storage and computing power can be dynamically scaled and redeployed to meet upward or downward processing requirements for individual applications or for entire business processes. Modularity helps to substantially reduce the time required to integrate, or separate, business systems.

Integration: Eliminating artificial barriers between elements of the IT environment frees capacity of underutilized resources and promotes interoperability across the IT environment. By designing IT resources and systems for integration, the infrastructure can be managed holistically, linking the resources and elements of the IT environment back to the services that it provides to the business.

Views of Darwin

Balancing Supply and Demand: The basic representation of the Darwin Reference Architecture is shown in the figure to the left. It places "Business Processes" at the top, and "Virtualized Resources" at the bottom. (Note the symmetrically placed control abstractions: "Business Strategy" above to direct business processes, and "Integrate and Orchestrate" below to direct the use of virtualized resources.)

Prominently featured in the model are links between Business Processes and Virtualized Resources, labeled "Supply" and "Demand". The prominence of this aspect of the architecture reflects the emphasis on HP's Adaptive Enterprise strategy and the Darwin framework's goal to "continuously balance business demand ... against resources ."

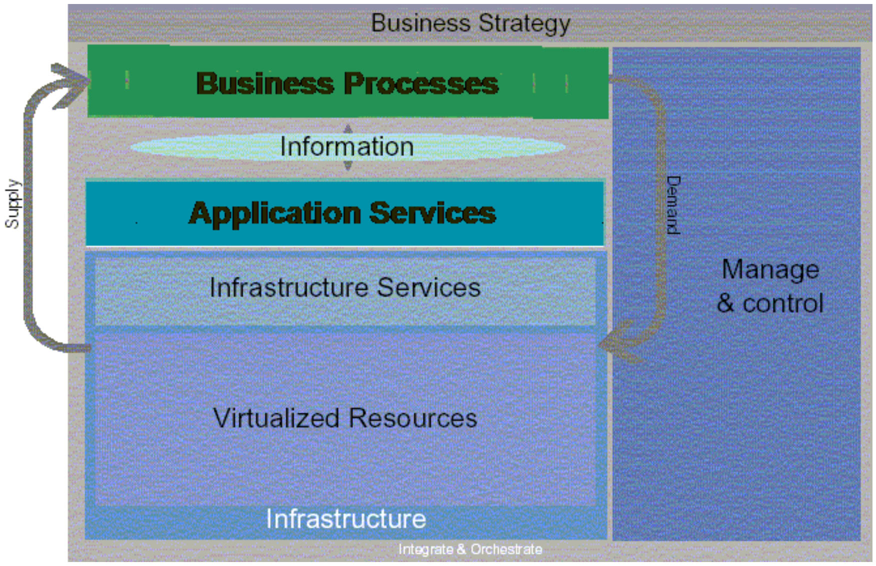


Figure 6: Darwin's Black Box Stack

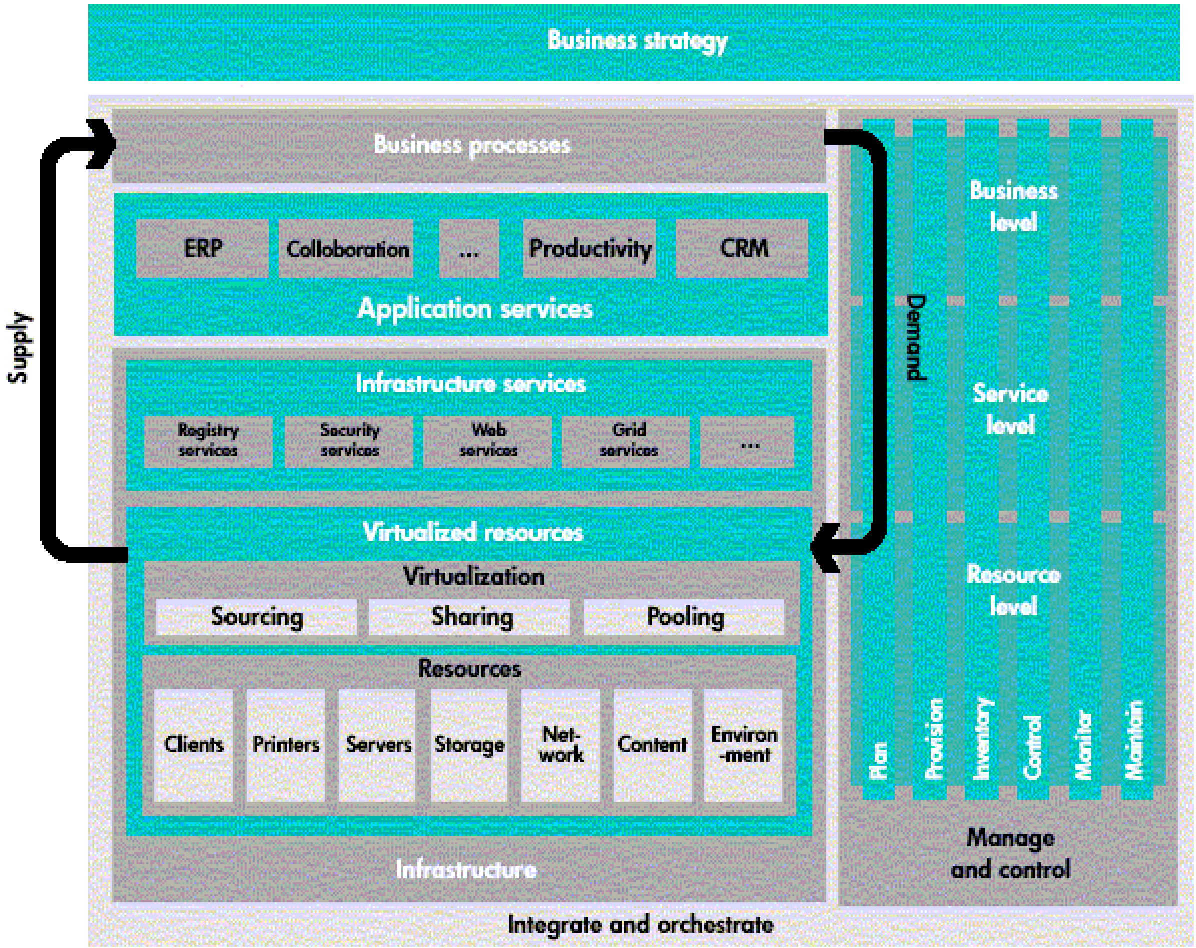


Figure 7: Inside the Boxes of Darwin

Looking inside the Boxes: The figure above shows the large abstractions of the previous diagram broken down (or “opened up”). In the IT stack above, we see applications as (predominantly) enterprise-scale business functions, such as ERP and CRM. Also included are “Collaboration” and “Productivity”, not applications themselves, but classes or types of application.

Below that are “Infrastructure Services”. (Compare this to the middleware layer in Fujitsu's Triole Architecture.) These include common systems such as security and registry (presumably equivalent to directory). It also includes integrative frameworks such as web services and grid services.

Below these are the “Virtualized Resources”. These present underlying resources – such as printers and storage (device resources), clients and servers (presumably compute or platform resources), and content and environment – to the higher levels of the architecture through a virtualization layer. (Compare this model of virtualization to that of IBM's on-Demand architecture.)

Darwin's model of system management

To the side of the IT stack of the Darwin References Architecture is an element labeled “manage and control”. This includes elements called “control”, “monitor”, and “maintain”, roughly equivalent to the traditional OA&M model. It also includes elements identified as “plan”, “provision”, and “inventory”. Each of these six manage and control functions spans the entire stack, from the lowest IT architecture up to business processes. This suggests (and other HP literature about the Darwin Reference Architecture confirms) an architectural model that relies on sophisticated system management capabilities that relate technical metrics to business performance.

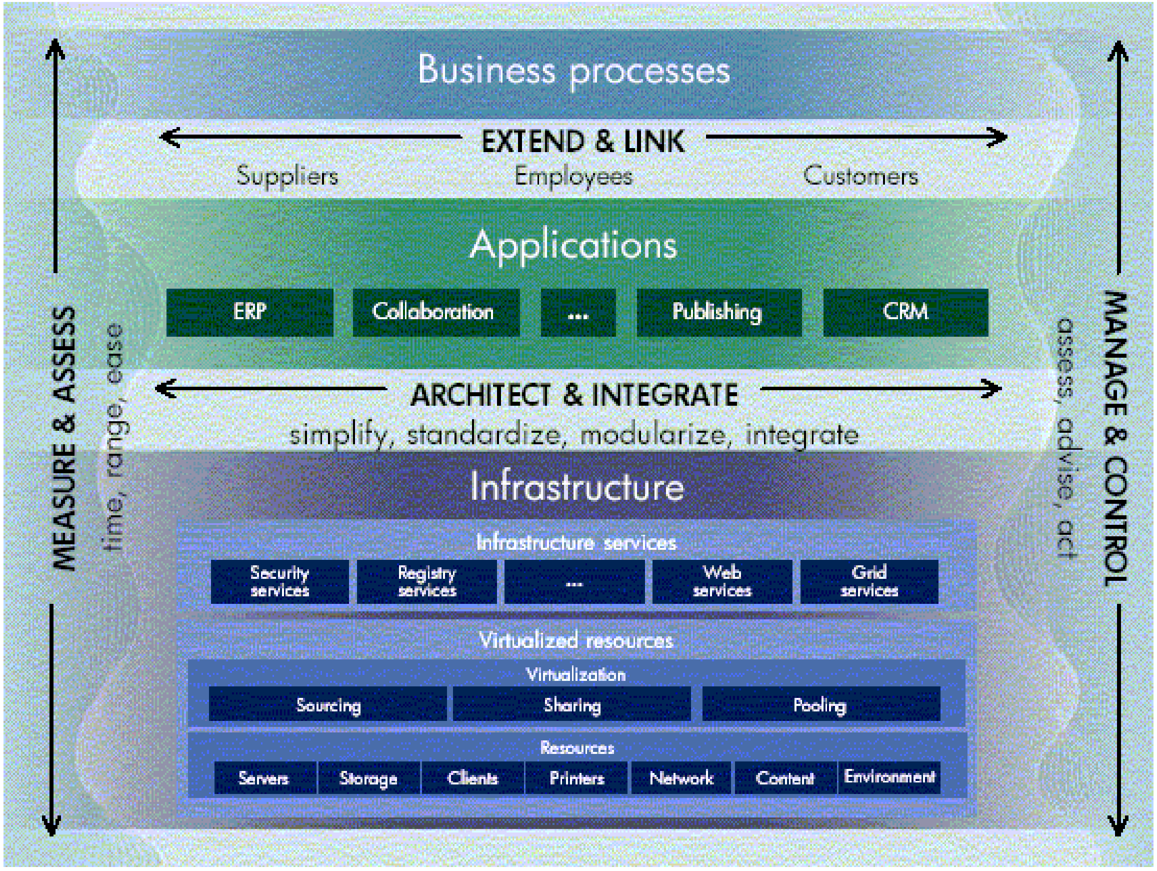


Figure 8: Measure and Manage in Darwin

This figure, from “Building an Adaptive Anterprise”, shows the significance of measurement and management in the Darwin Reference Architecture. Refer to: www.hp.com/large/globalsolutions/ae/pdfs/HP_whitePaper_v19.pdf.

Common System (Subsystem) Architectures

This section takes a more detailed look at common system architectures.

Common system architectures describe the elements, rationale, construction, and similar aspects of common systems. (Common systems are described in TOGAF.) Examples of common system architectures are system management, information management, and security. An architecture for one of these systems might include descriptions of systems that specifically implement aspects of the common system (alert management system, database, or authentication server, respectively, for the example common systems.) They might describe how systems of the common system are linked to each other (for example, a management framework; a distributed or replicated data management system; a public key infrastructure). And they may describe how a system that is not part of the common system take advantage of the services of the common system (how to report manageable events, how to construct queries or updates, how to check that a transaction is fully authorized).

Common system architectures are intimately related to certain "views" of the entire system or enterprise architecture. For instance, the management view, information management view, or security view of an enterprise architecture will feature or relate to the relevant common system architecture.

DIF Directories and Identity Management for Boundaryless Information Flow

Using the directory subsystem to achieve an identity management solution

Introduction

The Directory Interoperability Forum (DIF) provided a thoughtful analysis of several ways in which directory services can contribute to achieving boundaryless business objectives. The analysis begins with the assumption that every enterprise has its own identity management practice, relating to who can access which systems, services, and information, and under what circumstances they can do so. The differences range from the role of explicit corporate policy in guiding identity management, through the sorts of relationships a firm has with trading partners, to detailed assignment of specific identity management tasks to organizations or individuals within the firm.

The presentation from the DIF described a variety of such driving factors, and discussed their significance in creating an identity management architecture for Boundaryless Information Flow. The theory is illustrated by examples addressing several of the specific boundaryless business models that were formulated for use at the Washington Conference.

Further Analysis

Architecting identity management

In the first part of the paper, the DIF discussed the architectural context and challenges of identity management. Their paper:

- Introduced the concepts of identity management and Boundaryless Information Flow
- Described the role of identity management in the boundaryless enterprise
- Discussed the process of designing identity management into enterprise information systems
- Introduced a basic identity management implementation model
- Looked at the components that are available to build identity management solutions
- Considered the basic strategies for identity management implementation

What is identity and what does it mean to manage it?

The DIF paper takes the position that "identity" is meaningful within a specific context, which is called here a "community".

- Identity has meaning within the context of a community. It is what distinguishes an individual from the other members.
- From the individual's viewpoint, identity management includes managing the different identities that he or she has in different communities.
- From the community's viewpoint, identity management means the management, for community purposes, of the identities of its members.

The natural balance between the individual's view and the community's view has, in recent years, tipped toward the view of commercial communities:

"As the movement to define standards for Identity Management gathered pace through 2002, the community's point of view seemed to gain in importance, and the individual's point of view was heard less often. This was perhaps natural. The companies developing identity management products saw the communities, rather than individuals, as their customers. They were selling single sign-on and related systems to web service providers and enterprises. The individual would benefit, but only indirectly."

The boundaryless enterprise has been poorly supported by IT

This difference [between a boundaryless enterprise and a traditional one] has been likened to the difference between a jazz band and a conventional orchestra. In a jazz band, the musicians improvise to the limit of their skill within a harmonic and rhythmic framework. In an orchestra, they play a predefined series of notes under the direction of a conductor.

Boundarylessness has become common organizational practice, but the IT products and systems available today usually do not support it well. The boundaries between departments and people have gone, but the boundaries between information systems remain. Too often, information is held in "stovepipe" systems that were put in place for particular purposes or to serve particular departments, and that cannot share the information with other systems installed for other purposes or in other parts of the organization.

Architecting identity management for the boundaryless enterprise

Every organization has its own needs for identity management. Each enterprise has:

- Identity management for different purposes than other enterprises
- Different business drivers from other enterprises
- Different specific requirements from other enterprises
- Different policy and practice from other enterprises

Architecting identity management for a boundaryless enterprise is a significant challenge. Identity is perspective or context-based. The boundaryless enterprise's agility allows it to change its business perspective relatively quickly, and shape itself to be meaningfully different from its competitors. On top of the challenge presented by this malleability of context, we find that, in the boundaryless enterprise, the individual's independence is emphasized (like a jazz musician improvising), making the individual's view of identity relatively more important than in a conventional enterprise.

Because of the differences in uses, business drivers, requirements, policy, and practice, each enterprise has an individual identity management solution. Identity management for the enterprise cannot be bought "off-the-shelf".

Top-level view of identity management

This analysis of identity management emphasizes the role of the "identity store". While there are many ways to implement an identity store, the common assumption in current practice is that the identity store will be implemented as or using a directory. (The DIF paper expressly compares directories to relational database management systems, and finds directories better suited for this use.)

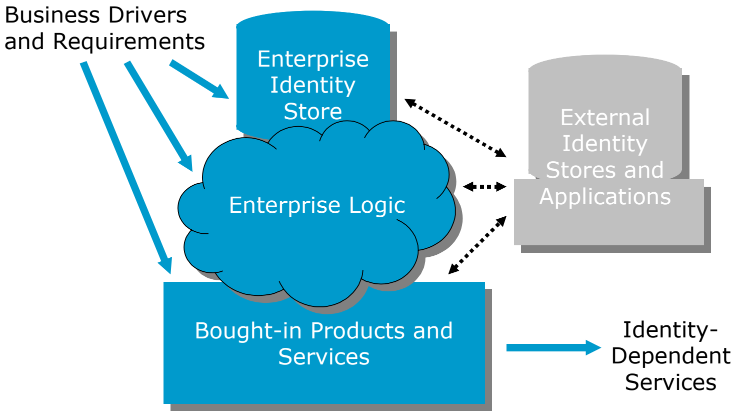

Business Drivers and Requirements
Enterprise Identity Store
External Identity Stores and Applications
Enterprise Logic
Bought-in Products and Services
Identity-Dependent Services

Figure 9: Top-Level View of Identity Management Architecture

In most current enterprise IT environments, individual applications or systems often provide their own identity stores. This is, in part, a consequence of the lack of standard models for identity stores and their usage by applications and other software systems. *"Nevertheless, it is good long-term policy to introduce a single – but probably distributed – identity store, and extend its use into the* *enterprise's information systems as they are modified and extended to meet business objectives."*

An important element of this architectural view is the inclusion of external identity stores and external applications. This recognizes that a boundaryless enterprise (or any enterprise that seeks to participate in the modern market) must interact with the systems of other organizations. The "sharing" will include information about individuals and access to applications or services. The relationship between these features is dependent on the specific practices and competitive strategy of the enterprise. For example, should applications in one firm use the identity store of another? Or should applications always use their own firm's identity store and leave it to that system to coordinate with the identity stores of other firms?

The DIF paper describes several strategies for composing identity management solutions: synchronization, distributed directory, meta-directory, federation, and composite strategies. The paper goes on to analyze the following three of the boundaryless business models from the Call for Papers, and to identify the significant architectural challenges, objectives, and options for each:

- Strategic Decision Support
- Supply Chain Automation
- Interpersonal Interaction

Solution Architectures

Bigger than a breadbox

This section takes a more detailed look at solution architectures.

Solution architectures solve specific problems that can contribute to a system, organization, enterprise, or larger architecture. Solution architectures are typically built on multiple common system architectures. They are similar in scale or scope to common system architectures, but rather than being created to reflect coherent "chunks" of technology, solution architectures are created to reflect coherent chunks of business functionality.

Two of the presentations made at the Washington Conference addressed solution architectures:

- RSA addressed identity management and a larger solution – identity and access management.
- CyberRave addressed information sharing, a combination of information management, security, and publishing.

Several presentations addressed solution architectures that were related to the primary theme of the presentation:

- The DIF presentation on directories addressed their use in identity management solution architectures.
- IBM's on-Demand Architecture described an architecture for supporting high-performance, specialized computational functions.

Identity Management: The Next Critical Step on the Internet

Introduction

RSA discussed the key role played by identity in business. In their presentation they identified the challenge of sharing information about identity across enterprise boundaries as one of the major barriers to achieving truly Boundaryless Information Flow.

RSA's approach strongly encourages migration from centralized models of identity management that may have been developed as part of so-called "single sign-on" initiatives toward federated identity management models. RSA particularly advocates the Liberty Alliance's emerging models and technologies.

Further Analysis

Collaborative Commerce

To achieve your e-business objectives, you must be able to selectively expose your information assets – such as mission-critical applications, sensitive data and intellectual property – to an ever-changing mix of users, including employees, customers, partners, suppliers, and channels.

Identity federation is the best identity management strategy for unifying disparate user accounts as well as facilitating single sign-on.

A solid **access management** strategy is a core element of an identity management solution, and necessary to seize e-business opportunities associated with identity management.

Two Basic Models

RSA Security's presentation described two models of identity management that can be used at an enterprise scale. One – the centralized model – features a single identity operator; that is, a single authority (with an appropriate IT capability) that is responsible for all aspects of managing identity, usually including enrollment of users, management of user authentication mechanisms (such as password maintenance, issuing tokens, smart cards, and so on), provisioning of services to users, and often management of authorization systems and some access control lists. This model can work well within an enterprise, but tends to form rigid boundaries, both technical and business, at the perimeter of the organization that "owns" the identity operator.

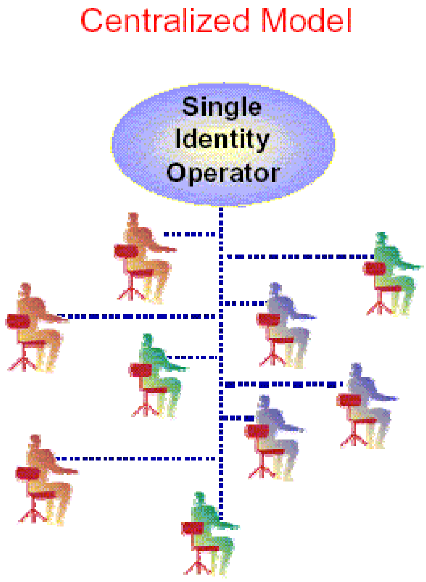


Figure 10: Centralized Model of Identity Management

The boundaries tend to "harden" around issues of administrative responsibility and potential liability for misadministration. Delegated administrative authority can address some aspects of this problem, but the network of delegated administrative relationships that might be required in even a relatively small group of cooperating firms quickly becomes unmanageable. Even within a single firm, a single, centralized administration for identity and, within that, authority can become extremely difficult to administer, potentially requiring substantial coordination of processes and policies of different lines of business or operating units for which there is no genuine business need for coordination.

The alternative model, recommended by RSA as the one toward which to migrate, is the federated model of identity management. RSA particularly references the model of identity federation created by the Liberty Alliance. In this model, firms can take responsibility for enrolling and authenticating users they "know", then use federation techniques to assert to other affiliated firms that the logged-on user is known to be who he claims to be. (See the Liberty Alliance web site at www.projectliberty.org/ for more on their federation model.)

Architectural elements and subsystem architectures

RSA represents federated identity management as being built on a "stack" of standards and services. In this figure the Liberty Alliance model of federated identity management is shown built upon the SOAP transport service and security services, such as WS-security, SAML, XACML, and XKMS. It also relies on SPML, which is a standard that contributes both to security and system management common system architectures.

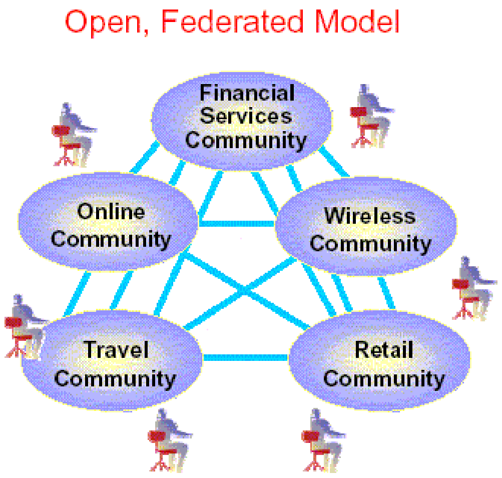
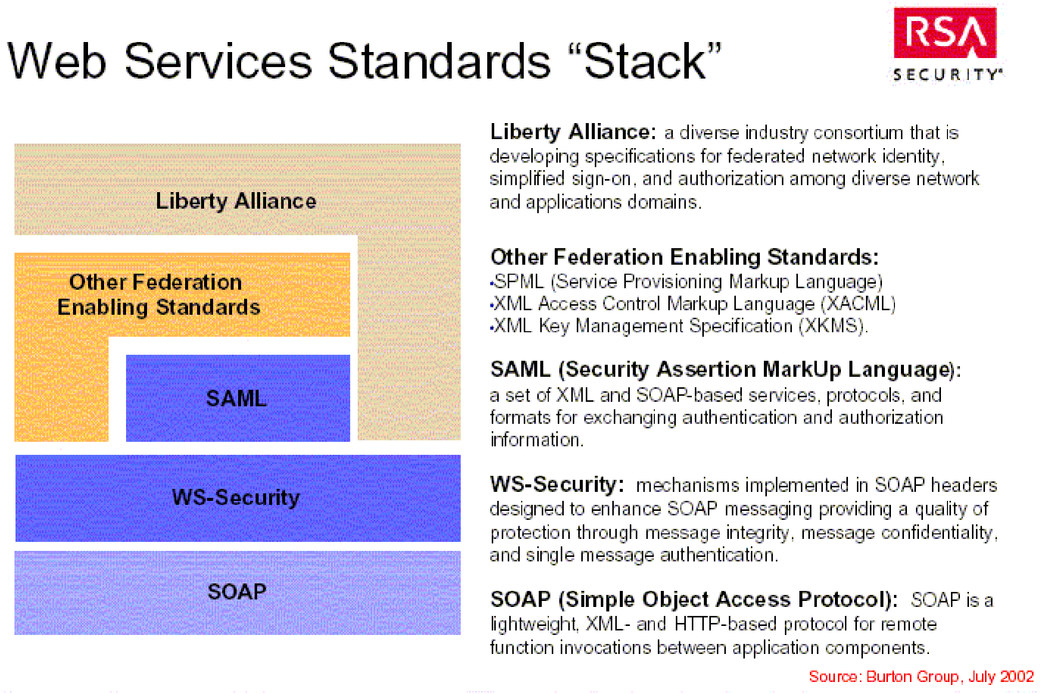


Figure 11: A Federated Model for Identity Management



Web Services Standards "Stack"

Liberty Alliance: a diverse industry consortium that is developing specifications for federated network identity, simplified sign-on, and authorization among diverse network and applications domains.

Other Federation Enabling Standards:
- SPML (Service Provisioning Markup Language)
- XML Access Control Markup Language (XACML)
- XML Key Management Specification (XKMS).

SAML (Security Assertion MarkUp Language): a set of XML and SOAP-based services, protocols, and formats for exchanging authentication and authorization information.

WS-Security: mechanisms implemented in SOAP headers designed to enhance SOAP messaging providing a quality of protection through message integrity, message confidentiality, and single message authentication.

SOAP (Simple Object Access Protocol): SOAP is a lightweight, XML- and HTTP-based protocol for remote function invocations between application components.

Source: Burton Group, July 2002

Figure 12: Identity Federation Protocols in the Interoperability Stack

Not shown explicitly are such related and required services as directory services.

Other related services

RSA's presentation made a case for integrating identity management and access management. While the exact scope of access management as compared to access control is unclear, it is reasonable to interpret it as including elements of privilege management, authorization, policy evaluation, policy enforcement, and access control, which we see as elements of the Security Common System Architecture. It also explicitly includes some portions of provisioning, which is a service that is appropriately (but not universally) associated with system management.

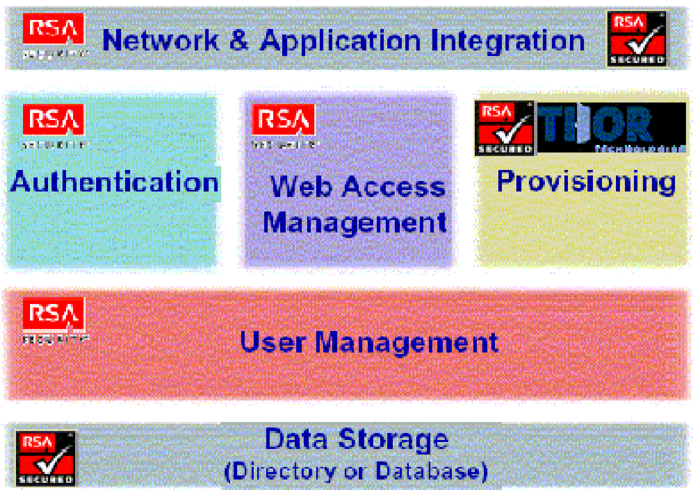


Figure 13: Architectural Elements of Identity and Access Management

Figure 13 shows RSA's schematic representation of the architectural elements that comprise an integrated approach to identity and access management.

Remote Access Virtual Environment (RAVE): A VPN Knowledge Grid

Introduction

CyberRave describes a virtual environment that can be created within or for a community of interest to allow remote access to information shared among members of the community. This Remote Access Virtual Environment (RAVE) is modeled after Private Virtual Networks (PVNs), at least with respect to the operational, trust, or user experiential aspects.

CyberRave's model includes a business model (actually a social or societal model) that supports and enables the effective operation of the technical implementation. This is reflected in CyberRave's mission: “To provide Communities-of-Interest (COI) with ever-increasing levels of information security, actionable intelligence, and simplified access to remote resources by establishing Vertical Communities governed by democratic online Advisor Groups (VCAG).”

Further Analysis

The objectives of the RAVE architecture include:

- Strengthen relationships among participating RAVE network users
- Increase the level of data security for RAVE users
- Recognize and support individual privacy and personal freedom to the greatest extent
- Reduce burdens associated with regulatory control
- Provide timely access to community intelligence, and deliver decision support through real-time situation awareness
- Establish a democratic framework for online communities
- Foster data transmission rights and responsibilities

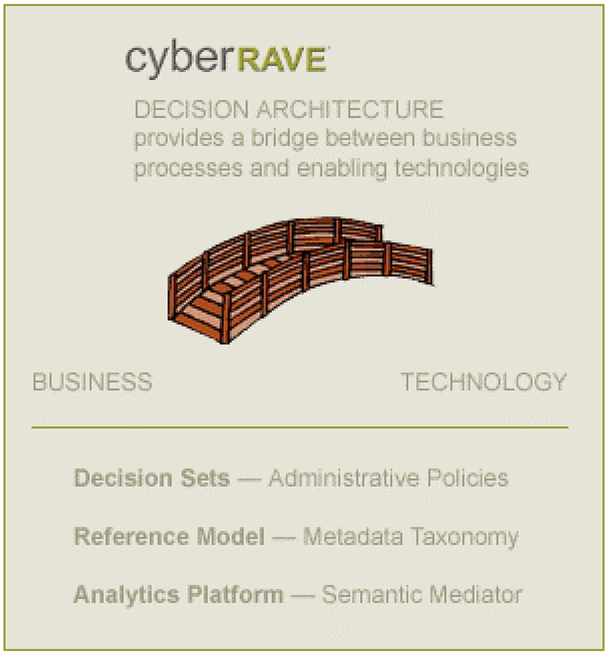
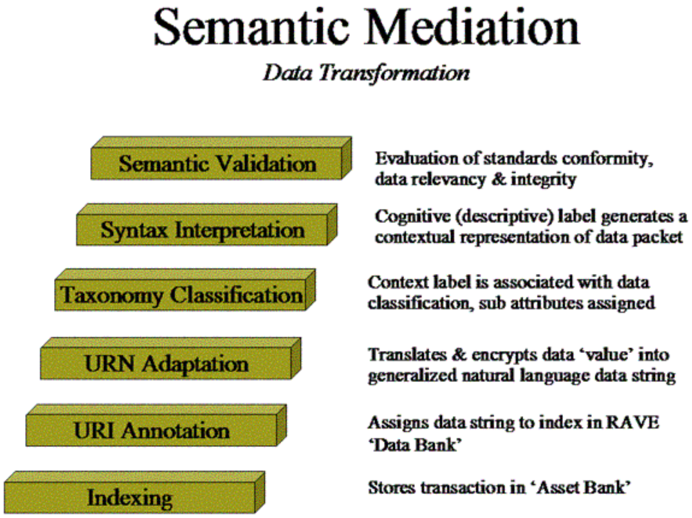

cyberRAVE
DECISION ARCHITECTURE
provides a bridge between business
processes and enabling technologies
BUSINESS
TECHNOLOGY
Decision Sets — Administrative Policies
Reference Model — Metadata Taxonomy
Analytics Platform — Semantic Mediator

Figure 14: Elements of a RAVE

Elements of a RAVE

While a RAVE is defined largely in terms of the social or business objectives it addresses, CyberRave does give some insight into the technical elements that are required or included. As shown in Figure 14, decision support is an important aspect of a RAVE. (In this regard, it resembles a response to the Strategic Decision Support business model from the Call for Papers.)

To that end, key elements of a RAVE are an approach to creating "Standardized data resources [that] establish an information commodity that Virtual Community users can exchange," and a set of semantic mediation services, as shown in Figure 14. A RAVE also includes an administrative function acting on behalf of the community, as shown in Figure 15.


Semantic Mediation
Data Transformation
Semantic Validation
Evaluation of standards conformity, data relevancy & integrity
Syntax Interpretation
Cognitive (descriptive) label generates a contextual representation of data packet
Taxonomy Classification
Context label is associated with data classification, sub attributes assigned
URN Adaptation
Translates & encrypts data 'value' into generalized natural language data string
URI Annotation
Assigns data string to index in RAVE 'Data Bank'
Indexing
Stores transaction in 'Asset Bank'

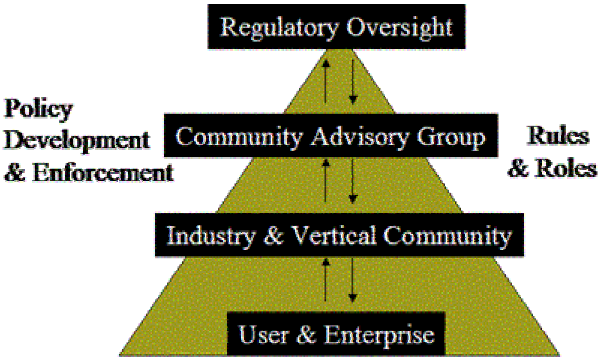


Figure 15: RAVE Governance

Figure 16: Semantic Mediation in a RAVE

Additional Models

Open Source: A Boundaryless Information Architecture

Introduction

Red Hat's submission addressed the challenge of business transformation. In some cases, businesses seek to transform themselves because their goal is to become boundaryless. Other companies value boundarylessness as a means to a business end, which is the company's goal. In some cases, the drive to eliminate the cost of supporting IT systems and infrastructures with vestigial boundaries results in a transformation of the business. Red Hat holds up the Open Source software development model as an exemplary boundaryless organization model. They explain that Open Source development is not only a boundaryless business, but that engaging in Open Source can actually act as the catalyst for a company's boundaryless transformation.

Further Analysis

See the full Red Hat submission for their complete description of "Customer-as-Innovator".

An important boundary in the creation of IT solutions for innovative businesses is the boundary between those who understand the problem, and those who understand how to solve the problem. Red Hat quotes Stefan Thomke and Eric von Hippel:

> *"In a nutshell, product development is often difficult because the "need" information (what the customer wants) resides with the customer, and the "solution" information (how to satisfy those needs) lies with the manufacturer. Traditionally, the onus has been on manufacturers to collect the need information through various means, including market research and information gathered from the field. The process can be costly and time-consuming because customer needs are often complex, subtle, and fast-changing. Frequently, customers don't fully understand their needs until they try out prototypes to explore exactly what does and doesn't work (referred to as "learning by doing").*
>
> "Customers and Innovators: A New Way to Create Value"
> Stefan Thomke and Eric von Hippel
> Harvard Business Review, April 2002.

Red Hat suggests that a way to eliminate this boundary is to empower the customer to be the innovator in IT solutions to his own business problems. They cite the example of GE Plastics' formation of the Polymerland division in 1998. GE exposed much of their accumulated company knowledge of plastic resins, including company data sheets, engineering expertise, and simulation software. on a web site, for use by customers and potential customers. *"GE's leap of insight was the realization that the proprietary knowledge they had kept from their customers was keeping their business from performing to its potential."* Again quoting Thomke and von Hippel:

"With the Customer-as-Innovator approach, a supplier provides customers with tools so that they can design and develop the application-specific part of a product on their own. This shifts the location of the supplier-customer interface, and the trial-and-error iterations necessary for product development are now carried out by the customer only.

Implications and Directions

While the number of responses to the Call for Papers was small (compared to the universe of work being done on boundarylessness in its many forms) a number of important trends or themes begin to emerge. Here we present a few that have been confirmed or reinforced in other Open Group meetings and forums.

Technical and Business Architectures Converge

Relating IT to the business needs it is intended to address is clearly one of the primary objectives of software engineering. Requirements analysis techniques of varying degrees of formality and rigor have been introduced (and, in many cases, eventually discarded) for years. Often these techniques have been tied to specific software engineering techniques; sometimes to specific programming languages. In recent years, more and more of the promotional material published by software and software services companies emphasizes that the advertised products and services are designed for "business".

Among the recurring themes in the presentations given at the Washington Conference were several suggestions that mapping IT to business through the traditional techniques of requirement analysis will be insufficient in the future. Among these themes:

- Boundaryless IT must be able to evolve as businesses adapt to changing environments.
- Boundaryless IT will cause the business to change as it drives strategic analysis and decisions.
- IT systems should measure their own performance in terms of business metrics, and manage, control, and reconfigure themselves in ways traditionally considered business decisions.
- Boundaryless IT systems can be used to "reach out" to business partners – accessing their services or federating workforces – to create trading and partnering opportunities that might not otherwise have been recognized or possible.

None of these themes is new. Any of them could be addressed with traditional software engineering or IT architectural techniques or methodologies. But taken in combination, they point to a need for a more fundamental change in approach. Viewed in the light of some of the other emerging trends, they suggest a need to develop a new paradigm of IT Architecture, more intimately connected not to business requirements, but to business objectives or even aspirations.

How will this be addressed in architecture? Answers in architectural methodology may include more sophisticated forms of requirements analysis; new ways to create "views" that hypothesize alternate future realities; and ways to specify designs that better represent interchangeable alternate configurations. It might also entail a different view of software systems as being plastic and malleable rather than rigid and well-defined.

In terms of the technical reference models used, a Reference Architecture for Boundaryless Information Flow will conceive the underlying elements of IT systems in new ways. Some of these new perspectives are described in the next few sections.

Virtualization will be the Foundation

The distinction between "platform" and "application" will still be important, but the definition of "platform" is radically changing. In the last several years we have already witnessed a transformation of conventional notions of platform. It is now assumed that a computer of any size will be networked, and that it will have applications that are entirely dependent on resources elsewhere on the network. Several of our presentations suggest that, just as "applications" and "information" have been distributed over the network, so in the future will the platform. This is called "virtualization".

All three of the integration architectures (IBM, Fujitsu, HP) included virtualization services. In their abstract architectures (if not yet in their products) IBM and HP unambiguously identify the virtualized platform as the platform of the future. While Fujitsu includes virtualization in Triole, their emphasis on the PI Template abstraction seems to make virtualization less central.

Virtualization is often identified with the cluster of technologies known as "grid computing". Certainly grid computing will be an important business and technical model of virtualization. We should also anticipate other models of virtualization, that will operate at a variety of scales meeting a variety of objectives. Storage systems have already been substantially virtualized (for largescale storage applications) by NAS and SAN architectures, and infrastructure technologies such as fibrechannel. We might also expect virtualization on a personal scale to be accomplished by the merging of plug-and-play architectures and local connectivity infrastructures such as firewire and Bluetooth.

How should virtualization be conceived, and how is it included in IT Architecture? Our presentations represent virtualization as a service abstraction at a relatively low level. Below that abstraction are the fundamental services that support virtualization. In IBM's model these are virtualization, integration, and automation; in HP's model they are sourcing, sharing, and pooling. However virtualization is composed, it will have significant impact on our approach to achieving boundarylessness.

Will it be possible to rely on a virtualized platform infrastructure to achieve boundarylessness, so that the architect of an application only has to avoid disabling the capability? Or will the application architect be required to affirmatively create the ability for his or her application to act boundarylessly? The most likely answer will be “it depends”. In particular, it will depend on whether the services infrastructure – the infrastructure above the platform boundary – is structured to support boundarylessness.

IT Architecture will be Driven by Functionality, not Technology

The architectural elements of the boundaryless architecture that lie above the platform interface are the common systems and solution architectures. These are the elements that may be called upon to mediate the boundarylessness of the virtualized platform, or to create that boundarylessness where virtualization doesn't supply it.

The presentations offered for the Washington Conference included only one that addressed a conventional common system defined in terms of technical capabilities or implementation.[1]

Common systems are likely to be recast, as mature IT tends to reorganize itself around business functions rather than technical taxonomies. For example, identity management is replacing parts of directory and security as the relevant architectural guide. Similarly, digital rights management is replacing elements of security and publish and subscribe messaging. Things that we had in the past considered “architectural” may need to be demoted to being considered simply “building materials”.

The Transformation is Underway

One thing that is apparent from the presentations is that the transformation of IT, and the way we think about it and use it, is already underway. The need for boundaryless business and the IT systems that support it are already keenly felt. The Internet and all it has enabled has shown that IT can help to achieve the goal of the boundaryless enterprise and more. The challenge now, as it has always been, is to shape the technologies that are brought forward by the combination of necessity and opportunity so that they merge and interoperate to the benefit of all users of IT.

[1] Common systems identified at the start of the Boundaryless Information Flow Refernce Architecture initiative were: Workflow Management, Messaging, Security, Directory, System Management, Information, User Interface and Ontology, and Transaction Management.

About the Author

Eliot M. Solomon has worked on the leading edge of IT for more than thirty years. He gained experience in such diverse fields as electronic warfare, military C^3, international telecommunications, medical electronics, and office automation equipment. Common to this work is real-time operation, mission or life-critical significance, distributed and networked computing, and a need for security, privacy, and assured integrity for the information being processed.

For the last seventeen years Solomon has brought his expertise and creativity to the Securities Industry. In fifteen years at the Securities Industry Automation Corporation (SIAC) and its subsidiary SECTOR, Solomon has made significant contributions to the architectures of systems and networks on which the entire market relies. He was appointed SIAC's first Distinguished Technologist and Vice President, in recognition not only of his contributions to SIAC, but also to the entire industry.

Solomon is founder and chair of the Securities Industry Middleware Council, Inc. (SIMC), an industry organization that works to improve the infrastructure of the Securities Industry. Solomon has guided SIMC since its founding in 1996, and helped it become a significant influence on what software vendors deliver to the industry, and the way the Securities Industry uses infrastructure technology. At The Open Group, Solomon chaired the DCE Program and is now a member of the Security Forum Steering Committee. He is frequently invited to speak at major conferences on the subject of linking information security and business policy, and the management of risk and trust. He holds an A.B. from Columbia University, and an M.S. from Polytechnic University.

About The Open Group

The Open Group is a vendor-neutral and technology-neutral consortium, whose vision of Boundaryless Information Flow will enable access to integrated information within and between enterprises based on open standards and global interoperability. The Open Group works with customers, suppliers, consortia, and other standards bodies. Its role is to capture, understand, and address current and emerging requirements, establish policies, and share best practices; to facilitate interoperability, develop consensus, and evolve and integrate specifications and Open Source technologies; to offer a comprehensive set of services to enhance the operational efficiency of consortia; and to operate the industry's premier certification service, including UNIX certification. Further information on The Open Group can be found at http://www.opengroup.org.