Implementing Services on Demand With the Sun™ Open Net Environment — Sun ONE

A Sun Professional Services White Paper

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
1 (800) 786.7638
1 (512) 434.1511

Contents

Introduction 1
The Sun ONE Platform 5
- The Core of the Platform: The Directory 6
- Talking to the Community: The Portal Server 7
- The Home of Business Logic: Application and Integration Servers ...8
- Reporting: The iPlanet Web Server 9
- Personal Transactions: iPlanet Messaging and Calendar Servers 9
- Business Transactions: The iPlanet Xpert Suite 10
- Foundations: The Solaris™ 8 Operating Environment 11
- Inventing the Future: The Forte™ Tools 11
- Conclusion: Sun ONE 12

Sun ONE Roadmap 13
- Precursors to a Successful Sun ONE Architecture 14
- Sun ONE Implementation Roadmap 14

The 3-Dimensional Framework 20
Web Applications 22
Web Services: Building a Foundation 24
- Infrastructure Services 25
- Example 26
- Iterations 26
- Summary 27

Future Web Services 28
Extending Web Services 28
Qualities of Service 29
Example 30
Summary 31
The SunTone Architecture Methodology 32
Project Phases 33
Workflows 34
Milestones 36
Artifacts 36
Patterns 37
Iterative Refinement 37
Sun Professional Services Capabilities 38
Conclusions 39

CHAPTER 1

Introduction

At Sun, we view Services on Demand as the most realistic and viable approach for companies moving towards the development and deployment of Web-based services. The Services on Demand vision recognizes that businesses will need to use and support a variety of service delivery styles, such as **traditional dedicated applications**, **Web applications**, **Web services**, and **future Web services**. And the Services on Demand model includes all of these styles.

Services on Demand are how enterprises use their IT environment to transact and report business operations and to communicate with others — anywhere, anytime, on any device. The Services on Demand concept is the foundation for a modular, flexible, and automated access to digital assets, including computing resources, from virtually anywhere. The Services on Demand vision is of a comprehensive framework, encompassing traditional Net-based services, such as security, authentication, and directory, along with more advanced capabilities, such as virtualized storage and composite services (those created by combining separate services.)

Services on Demand offer businesses a way to improve efficiencies in business software usage. Using Services on Demand not only delivers value and productivity today, but also prepares your IT infrastructure for tomorrow. New protocols are emerging that remove even more barriers for doing business. Global initiatives that will enable components to combine and recombine are nearly complete, providing a new generation of IT services that will evolve as part of today's Services on Demand.

As the next generation of Services on Demand technologies quickly emerge, and standards bodies composed of Sun and industry partners move to finalize the underlying protocols, organizations faced with sorting through the marketing hyperbole and XML alphabet soup are looking for a real answers as to tough questions, such as:

- What are Services on Demand about?
- Why are Services on Demand meaningful to my business model?
- How do I implement Services on Demand in an enabling, yet non-disruptive manner?

This white paper answers these questions in the context of the Sun™ Open Net Environment (Sun ONE) Architecture Framework and SunTone℠ Architecture Methodology. It provides examples and lays out a roadmap for organizations interested in applying Services on Demand technology as a means to achieving lower operating costs, easier access to business assets, and increased responsiveness to customer service requests.

Sun Professional Services has many years of experience implementing Services on Demand, taking legacy application functionality and exposing it to the Internet in a reliable, highly-available, scalable, flexible, manageable, and secure manner. Already, Sun Professional Services is implementing next-generation services to provide enterprises with sustainable competitive advantages via streamlined workflow and more efficient business processes. This execution on the Sun ONE vision is possible by meshing deep platform and technology expertise with serious, business-specific domain knowledge and architecture experience.

Sun fully understands and fully supports the concept of Services on Demand. What Sun does not understand (or support) is the notion that Services on Demand — specifically, Web services — are anything revolutionary, or even new. Web services are self-describing software components that can automatically discover and engage other Web components to complete complex tasks over the Internet. In other words, Web services are just software components that are networked — instead of applications that are networked.

For several years, Sun has been helping enterprise customers expose existing, legacy business data and business applications to the Internet and develop new Internet-based business applications for anywhere, anytime, any device access. And Sun has been doing it with the same approach being trumpeted as revolutionary by newcomers to the Web services arena — open standards, robust technologies, and an integrated stack of software, hardware, support, and architectural services.

So, what's all the fuss about? There are, in fact, a new wave of Web services initiatives that are emerging. Standards such as SOAP (Simple Object Access Protocol), ebXML (e-business eXtended Markup Language), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery, and Identification), SAML (Security Assertion Markup Language), as well as a host of Java™ API (Application Programmer Interface) specifications for handling these standards, are in the process of being formulated.

Sun has a long history of putting technological expertise at the forefront — ahead of marketing and hype. With Web services, this has not changed. Sun's architectural and implementation expertise has been field-tested and proven to provide robust, scalable, secure, manageable service infrastructures for governments, financial institutions, service providers, telecommunication companies, and a world of other global industries.

What is new is the evolution of Services on Demand — from anytime, anywhere, any device access to business functionality over the Internet into services that are dynamically discoverable, reusable, and recombinable into flexible, interoperable, innovative business applications. Like any evolutionary change, this development is happening in stages, from Web applications to current XML-based Web services, to future federated and dynamic Web services, each described below:

- Web Applications: Business data and functionality exposed on the Internet for access anywhere, anytime, from a browser. Legacy applications exposed include unified messaging, monolithic CRM and ERP systems, and packaged groupware. Internet protocols include HTTP(S), HTML, IIOP, JDBC™, SMTP, SMS, and IMAP technologies.
- Web Services: Web services deployed in a structured container or environment that allows the separation of business logic from quality-of-service issues such as client device independence, availability, security, and manageability. In addition to those above, Internet protocols include LDAP, JavaServer Pages™ (JSP™), Java 2 Platform, Enterprise Edition (J2EE™), SSL tunneling, and XML technologies.
- Future Web Services: Web services that are deployed using a combination of registries and interface descriptions to create discoverable, self-describing units of functionality that can be invoked and combined dynamically. Protocols include UDDI, WSDL, ebXML, SOAP, SAML, Java API XML extensions (collectively referred to as JAX*), Jini™ and Liberty[1] technologies.

Sun Professional Services has extensive experience from the past and present in architecting and deploying Services on Demand, and is using that experience to develop best practices for deploying future Services on Demand as the protocols emerge.

The heart of successful implementations of enterprise-ready functionality and qualities of service is the Sun ONE platform. This integrated and integratable suite provides the flexibility to transform the legacy services of the past into the Web services of the present, and to continue to evolve them into Web services of the future.

The remainder of this paper discusses the architectural and implementation strategies that Sun Professional Services has developed to create flexible, available, scalable, secure, manageable business functionality and to make it available anywhere, anytime, and on any device.

Section 2 describes the Sun ONE platform, the constituents it serves, and the resources it combines to serve those constituents.

Section 3 describes the roadmaps an organization or enterprise might take in deploying Sun ONE, emphasizing the non-disruptive nature of the process.

1. For more information on the Liberty Alliance Project, go to www.projectliberty.org.

Section 4 provides a brief introduction to the Sun Professional Services 3-Dimensional Framework (the “Cube”), a key component of the SunTone Architecture Methodology.

Using the Cube as an organizing framework, sections 5-7 present some of the architectural principles and technologies used to design Web Applications (Section 5), Web Services Building a Foundation (Section 6), and Future Web Services (Section 7). Each section contains a generic discussion followed by a case study.

Section 8 outlines the SunTone Architecture Methodology used in creating Sun ONE architectures with the Sun Professional Services methodology.

Section 9 describes some of the services Sun Professional Services uses in delivering Sun ONE architectures.

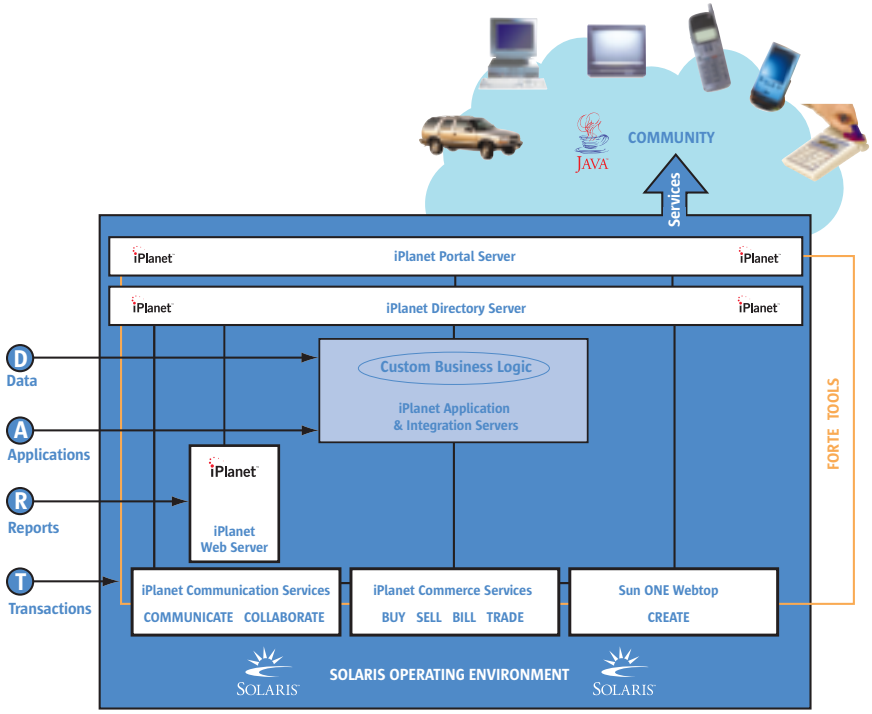
CHAPTER 2

The Sun ONE Platform

The Sun ONE platform is specifically engineered to support the information lifeblood of the modern enterprise: data, applications, reports, and transactions (summarized by the acronym, “DART”). Whether mainframe/terminal-based or deployed on the Internet, collectively, these four resources subsume the IT needs of a modern enterprise.

The goal of the Sun ONE platform is to provide a more flexible, cost-effective environment where DARTs can be developed, deployed, discovered, and utilized. This goal is realized through the use of open standards and robust software and hardware components that create an integratable stack for implementing enterprise-class, Internet-accessible services.

The diagram below summarizes the DART organizational model of the Sun ONE platform.



At the top of the diagram is the community being served by the framework. This community could be customers, partners, suppliers, or even (perhaps most important) employees. The goal of modern business is to identify a target community and to monetize that community by providing value — in the form of services — to that community.

The raw materials for building the services that are offered to this community are the data, applications, reports, and transactions that are the information assets of an enterprise. Understanding what assets an enterprise has and what their costs and values are allows an enterprise to focus on its higher-value, lower-cost assets and to outsource or eliminate its lower-value, higher-cost assets. Being able to create valuable services from these assets more quickly and more economically provides an enterprise with a sustainable, competitive advantage in its industry.

The Core of the Platform: The Directory

The first step in turning the raw information assets into value for the community is creating a directory of both the assets and the community. This is the role of the iPlanet™ Directory Server, and it sits in the middle of every connection from the assets to the community.

The iPlanet Directory Server is the core of an LDAP-based, multipurpose directory framework that acts as both an information repository and a policy management infrastructure. As a result, it not only provides enterprise-wide and extranet-wide access to information about the supported communities and information assets, it also provides authentication, authorization, non-repudiation, metering, rating (pricing), and billing support.

As a result of its central role in managing community information and deploying enterprise assets, the directory contains an enterprise's most valuable assets. The iPlanet Directory Server suite of software includes a number of auxiliary tools to fulfill this central role:

- iPlanet Meta-Directory: A software framework and a set of adaptors and connectors to provide bidirectional integration between the master LDAP directory and all of the special purpose directories throughout an enterprise (such as PeopleSoft, SAP, Microsoft Active Directory, etc.)
- iPlanet Certificate Management System: A digital certificate management system for producing, signing, and distributing the digital tokens that implement a typical chain-of-trust authentication and authorization mechanism.
- iPlanet Policy Server: A framework for defining access and authorization rights for the enterprise, including both constituent-based rights, resources-based rights, and delegated administration.
- iPlanet Directory Access Router: A load-balancing mechanism for examining and re-directing LDAP requests to appropriate servers and an invaluable aid to scaling and distributing directory resources.

Together, these components provide the software platform necessary to collect, distribute, manage, and protect the most valuable core information assets of an enterprise.

Talking to the Community: The Portal Server

Since the Sun ONE platform is about providing value to a community, the next important component is the iPlanet Portal Server. This is where the constituent meets the enterprise. There are two main parts to the iPlanet Portal Server and a number of auxiliary parts. The main parts are content aggregation and customization and secure remote access. The auxiliary parts include the Instant Collaboration Pack, Personalized Knowledge Access Pack, Access Pack (includes a Wireless Access Protocol (WAP) server), and Secure Remote Access Pack.

The content aggregation component allows for the aggregation of any HTML or XML encoded content as well as virtually any application that can be run on any major server OS, including X Windows-based applications running on the UNIX® operating system. In addition, the content aggregation component includes Java clients for standard Internet operations, such as mail and file transfer. Content and applications services are defined as channels in the iPlanet Portal Server, and these channels can be combined into a flexible, seamless, personalized, browser-based user interface.

The secure remote access component uses a Java technology-based virtual private network (VPN) client that can run on any Java technology-enabled Web browser — Sun's patented Netlet technology. A constituent at his or her Web browser anywhere on the Internet can download the Netlet and all further communications between the constituent and the enterprise — mail, internal enterprise Web sites, custom applications — regardless of what native protocol they use, will be tunneled through an encrypted Secure Sockets Layer connection (port 443 usually). This provides protection from eavesdropping.

The portal server is integrated into the LDAP framework for directory information and can call on enterprise-wide, shared directory resources for user authentication and group membership information. A constituent logs in through one of a variety of authentication mechanisms including native UNIX user information, a digital certificate, LDAP directory, challenge and response, and Kerberos. Once a constituent is logged in, the portal server provides personalized information and authorized service access.

The auxiliary packages provide additional functionality to the iPlanet Portal Server, including a flexibly configurable search engine, a WAP (Wireless Access Protocol) server, and transcoding services for converting content for presentation on a variety of devices, including HTML/XML to Wireless Markup Language (WML).

Together, these features provide flexible, secure access and personalized, customized interfaces to the enterprise's constituents.

The Home of Business Logic: Application and Integration Servers

Because they are crucial resources that are shared across the entire enterprise, the directory and portal infrastructures are fundamental to the Sun ONE architecture. However, by themselves, the provide none of the business logic that make up Services on Demand.

For the most part, the business logic exists as Enterprise JavaBeans™ (EJB™) and other J2EE components residing in the iPlanet Application Server. The application server provides a scalable, available, robust, J2EE standard execution environment for subroutines, such as validate an address and store it as a constituent's preferred shipping address, that make up the services offered by an enterprise to support its communities.

In addition to providing a J2EE technology standard implementation of the runtime environment and libraries that programmers can rely on, the application server also provides for redundant replication of state information through software clustering. This provides high availability to stateful subroutines — even in the event of hardware failure. The application server also manages pools of reusable resources, such as execution threads and database connections, making business logic easier to write and more robust.

The application server is enhanced by two additional servers, the iPlanet Integration Server, EAI Edition (formerly the Forte Fusion™ product) and the iPlanet Integration Server, B2B Edition (formerly iPlanet ECXpert). The iPlanet Integration Server is a framework for integrating a variety of legacy applications. such as SAP and PeopleSoft (using available connectors), and defining workflows that create single services from a variety of functions that cross services. For example, the integration server could retrieve information from both a human resources system and a separate legacy enterprise resource planning system to optimize and automate shop floor employee scheduling. This allows an enterprise to integrate all of its internal services and make them available for use as Services on Demand.

The iPlanet Integration Server, B2B Edition is a facility for secure document exchange. Just as the iPlanet Integration Server, EAI Edition allows for integration of an enterprise's internal services, the iPlanet Integration Server, B2B Edition allows for integration of one enterprise's information with another's. It provides a framework for mapping document data from one enterprise to another and a mechanism for securely exchanging that data using EDI or XML protocols.

The application server therefore provides a scalable, available, open-standard environment for the execution of business logic, while the integration server provides integration of internal legacy application services and external data services.

Reporting: The iPlanet Web Server

The humble Web server has made a transition over the last five years, from a simple HTTP server to a Java technology-enabled, automatic report generation package. Where static pages, augmented by simple routines written to the Common Gateway Interface (CGI) protocol, were all that was served in the beginning, modern Web servers integrate JavaServer Pages technology to dynamically create complex, customized documents with up-to-the second information.

By providing a stable environment for running Java routines, the iPlanet Web Server effectively functions as a lightweight application server, providing an execution environment and resource pooling to subroutines implementing reporting logic. Using these facilities, content can be shared seamlessly across multiple Web server machines, allowing unparalleled scalability.

In addition, the extensibility of the Web server allows it to adopt new protocols as they are finalized and approved. SOAP and XML are being accommodated as new releases of the protocol specifications emerge. This not only protects the Web server investment from obsolescence, it also provides a facility for transcoding content from a base XML form to a device-specific form.

By providing a robust, scalable, adaptable J2EE execution environment, the Web server has become more than just a simple protocol engine. It has become an auxiliary execution environment for assembling and displaying information.

Personal Transactions: iPlanet Messaging and Calendar Servers

Effectively, transactions on the Internet can be divided into two types, person-to-person and machine-to-machine. The former involves technologies such as e-mail, calendar, and instant messaging, where either the producer or consumer of the information is a person. The iPlanet communications portfolio provides the most scalable, flexible e-mail server available, as well as a comprehensive calendar server for scheduling people and resources.

The iPlanet Messaging Server combines Simple Mail Transport Protocol (SMTP) for sending and receiving e-mail to and from other machines with Internet Mail Access Protocol (IMAP) and Post Office Protocol (POP) servers for client mail access (that is, allowing people to retrieve and read their mail). In addition, it provides an easy-to-use Web mail interface for remote access to any Web browser, as well as seamless integration with the iPlanet Portal Server mail client for secured anywhere, anytime access.

The iPlanet Calendar Server has provisions for personal and business calendars, as well as facilities for scheduling enterprise-wide events (such as product releases), group events (such as team meetings), and resources (such as conference rooms). These calendars can be combined in flexible ways to provide personalized, context-relevant scheduling.

The iPlanet communications portfolio is extensible enough to accommodate additional communications channels and protocols, such as Short Messaging Service (SMS) and instant messaging. Open standards allow seamless transaction integration with whatever services support open standard communications protocols.

Business Transactions: The iPlanet Xpert Suite

While much of the business logic that an enterprise deploys may be based on custom code running in the application server environment, the most common business functions — buying, selling, billing, and auctioning — have been collected into a suite of software.

The iPlanet BuyerXpert implements electronic procurement functions. This includes a catalogue of items that an enterprise wants suppliers to deliver, as well as mechanisms for managing approvals and payments from within the enterprise.

iPlanet SellerXpert implements electronic sales. Like iPlanet BuyerXpert, this provides a catalogue of items and prices, and then iPlanet SellerXpert adds a shopping cart and substitutes a settlement system that allows for collections in place of mechanisms for managing approvals and payments.

iPlanet BillerXpert simplifies bill generation and presentment. Automating invoicing and payment tracking provides cost savings to one of the most value-generating aspects of business. In addition, financial service providers are beginning to offer bill presentment systems that can cut down on the high costs of letters of credit and other financial arrangements for international business exchanges.

Finally, iPlanet Market Maker allows for auctions and reverse auctions. This brings electronic business exchanges as close to perfect pricing as possible by allowing for the market to dynamically decide the value of a service.

Combined, these off-the-shelf e-business packages automate most of the basic functions required for automating business-to-business transactions. They can, therefore, dramatically reduce time-to-market, while increasing the flexibility of Services on Demand and freeing up resources to focus on providing enterprise-specific value in the form of custom business logic.

Foundations: The Solaris™ 8 Operating Environment

The components of the Sun ONE platform described thus far provide the functionality required to deploy Services on Demand:

- Data, information that's meaningful to an organization's constituents
- Applications, both off-the-shelf integration functionality as well as environments for customization
- Reports that are dynamically generated and transcoded for anywhere access
- Transactions, both person-to-person and machine-to-machine

Underneath this functionality, the Solaris™ 8 Operating Environment provides a scalable, manageable, secure, robust foundation to provide enterprise-class qualities of service. The Solaris Operating Environment not only implements a world-class, POSIX standards-compliant UNIX platform, it also includes clustering and multipathing technologies for hardware failover, RAID technologies for disk space and volume management, a journaling file system for transactional consistency in case of hardware failure, and resource and bandwidth management facilities for guaranteed service levels.

The operating environment also includes world-class implementations of the basic facilities needed for a capable networked platform, such as Domain Name Service (DNS), Network File Service (NFS), and the File Transfer Program/Protocol (FTP). Augmenting the value of the operating environment is its close integration with Sun's UltraSPARC™ workstations and servers. The operating environment includes facilities for monitoring, metering, and managing both hardware and systems software.

Inventing the Future: The Forte™ Tools

All of the Sun ONE components described so far comprise the overall operating environment for Sun ONE. In addition, Forte™ Tools for Java, C, C++, and FORTRAN languages provide a rich set of tools for developing, testing, packaging, and deploying new functionality that is demanded by the specific needs of an enterprise.

In addition to compilers and linkers, the Forte for Java development environment provides code browsing and editing tools, debuggers, class-hierarchy management tools, and support for the protocols, such as XML and WSDL, that are needed to expose application logic as Services on Demand now. Plus, it offers for emerging technologies, such as ebXML, Liberty and JAX*, needed to support dynamic, complex, context-sensitive business trading relationships in the future.

Support for current and emerging Services on Demand technologies is delivered by the Forte tools and the Forte for Java Community and Enterprise Editions via an innovative modular architecture based upon the NetBeans™ framework[2]. Developers can choose the best implementation from a growing number of third-party vendors eager to deliver technology support and advanced capabilities unimaginable from a single vendor.

Conclusion: Sun ONE

While this description of the Sun ONE architecture has treated the various products point-by-point, the true value of the platform is best seen when it is taken as a whole. The Sun ONE platform simply provides the premier complete, integrated, robust, scalable, secure Services on Demand platform available. Because the Sun ONE platform rigorously adheres to open standards, it is also integratable with other open standards based products, Sun ONE is the platform for providing end-to-end and application-to-hardware support.

Furthermore, the Sun ONE platform is not just new technology for its own sake. As discussed in the next section, the Sun ONE platform provides an incremental, evolutionary roadmap from legacy systems and Web applications, to fully automated and exposed Web services. The next section describes this roadmap in detail; the following sections provide examples of the technologies used in legacy applications and Web applications and how they can be migrated to Web services without disrupting an enterprise.

2. For more information on this open source initiative, go to www.netbeans.org.

CHAPTER 3

Sun ONE Roadmap

Seasoned IT executives and architects can cite many historical examples of the pitfalls associated with employing technology for the sake of technology. Many still carry the scars of heavy investment in well-designed, but fatally-flawed technologies such as CORBA, which did not deliver the benefits envisioned due to a lack of direct correlation to specific business goals. Blindly employing Web services technology as a means to an ill-defined business goal would prove a similar recipe for disaster, thus, the need for a calculated, phased and measured strategy.

The Sun ONE vision addresses the concerns of customers by defining an evolutionary roadmap that can, and should, be customized to meet the competitive business needs of each enterprise. Customers are urged to go cautiously in their integration of Web services to ensure that the investment adds value to their organization. Critical success factors against which any new technology deployment should be measured include:

- The ability to measure return on investment ROI
- Protection of brand and customer satisfaction
- The ability to define, deliver, and measure quality of service (QoS)
- Protection of existing technology investments

The Sun ONE Implementation Roadmap addresses each of these success criteria by leveraging a proven architectural methodology, the SunTone Architecture Methodology, to custom fit an architecture as a means to a business goal, whether it be enhanced employee productivity, quicker time to market of new business services, more efficient supplier and distributor relationships, or increased end-customer value through more personalized services.

How and where do you start? If there were a one-size-fits-all roadmap to meet all organizations' requirements, we would publish it. While there is no such roadmap, a few common denominators do exist that transcend industry and organizational boundaries, laying a foundation that can support a variety of business-specific strategies that leverage the power of a service-driven architecture.

Precursors to a Successful Sun ONE Architecture

The precursors to any successful Sun ONE architecture, regardless of organization or industry differences, are three key elements: resources, constituents, and business strategy. IT executives and architects must ask themselves “what do I want to expose and to whom should I expose it, and why should I even go to the trouble?” Thus, the first step an organization takes is to identify their current resources and constituents. Armed with an understanding of these critical elements, a business strategy can be developed that lays out a phased, customized approach to prioritized implementation of Services on Demand while delivering the greatest ROI possible.

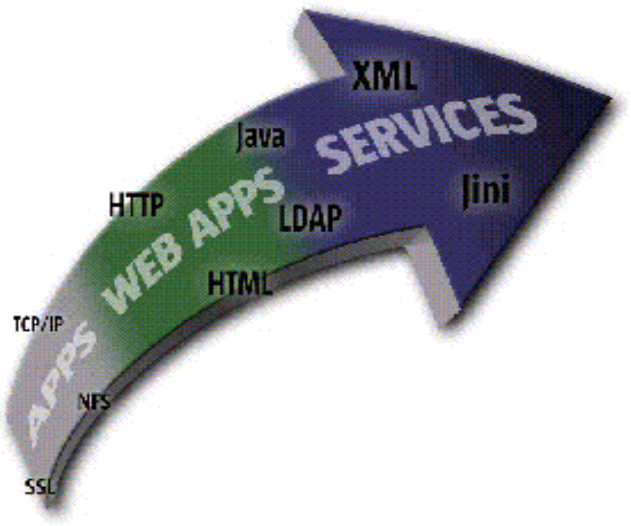
The critical precursors to a successful implementation of the Sun ONE architecture are:

- DART Assessment – Identify key resources (DARTS) and constituents (employees, customers, suppliers, etc.) of the organization or enterprise, mapping constituents to relevant resources.
- Business E-Strategy Definition – Develop a long-term business strategy that identifies and prioritizes e-initiatives capable of delivering significant, measurable ROI, such as employee productivity (Company.Net), value chain/SCM (eCompany) or one-to-one marketing/CRM (MyCompany.Com).

Once these critical precursors have been analyzed, an organization is ready to employ the Sun ONE architecture and platform in support of the enterprise e-strategy.

Sun ONE Implementation Roadmap

A Sun ONE Services on Demand roadmap should be based upon the Services on Demand Continuum, as shown in the following diagram.



This continuum identifies the technology evolution enabling Sun ONE Services on Demand, from early TCP/IP-based local applications, to current XML-based, self-defined-DTD Web services, to emerging UDDI, ebXML, and Jini technology-based dynamic and collaborative Web services.

The Sun ONE Implementation Roadmap for the majority of organizations involves three phases:

- Phase I: Web Applications
  - Develop enabling point-to-point access between humans and legacy applications and/or data
  - Characterized by applications-specific authentication mechanisms and tightly-integrated, often proprietary, interfaces
- Phase II: Web Services
  - Evolve Web applications by introducing open standards-based, loosely-coupled interfaces that leverage common enterprise resources, such as a single user directory for authentication and authorization
  - Develop new Web services which leverage open standards-based, loosely coupled interfaces for statically-defined application-to-application business process automation
  - Employ internal registry of services to enable manual discovery of service availability and interface requirements
  - Leverage proven Sun ONE platform to deliver massive scalability, continuous availability, real-time flexibility, and end-to-end security inherent in the integrated product suite

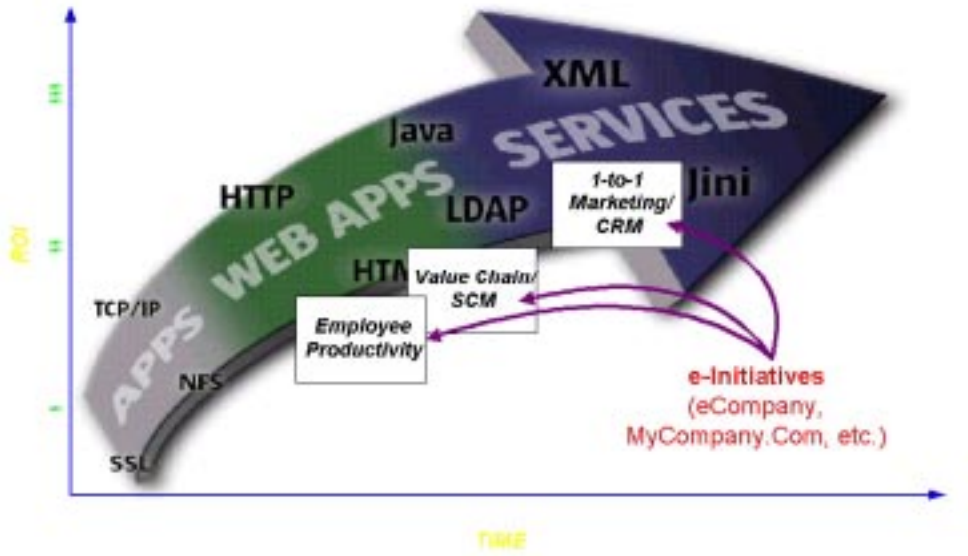
- Phase III: Future Web Services
  - Adopt as-yet-to-be-defined business schemas and supporting open standards-based technologies that will enable dynamic, federated Web service collaborations, delivering the right mix of Web services to meet specific business requirements with guaranteed QoS levels
  - Deploy Web services that dramatically improve existing business processes, gaining competitive advantages inherent in being able to dynamically source the right part, at the lowest price, from the most reliable bidder, in real-time as an order is received

Most organizations do not fall comprehensively into any one phase, but instead have different applications and services spanning all phases. Thus, an effective e-strategy is actually a calculated prioritization of all e-initiatives, the prioritization based upon a balance of ROI opportunities and budget realities.

Most organizations have already developed and deployed some number of Web applications that have made critical applications and data available in a point-to-point manner over the Internet. Authentication procedures and mechanisms may vary from application to application, but the benefits of exposing customers and/or employees to these critical business resources are well accepted. These organizations are in Phase I of the Sun ONE Implementation Roadmap, and possess the technical infrastructure and business experience upon which to base their move to Web services, where the enabling Web protocols, such as LDAP and HTTP, are extended to support B2E and enterprise-managed B2B interactions via emerging protocols such as UDDI, SOAP and XML. Many Sun customers have already begun leveraging these enabling technologies for inter-application processing in their production environments.

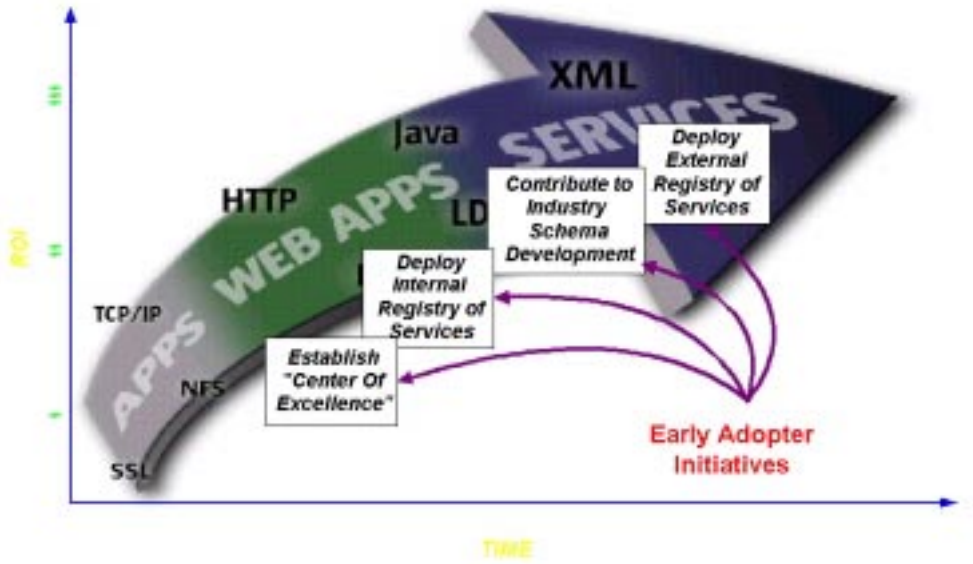
Note that passing through Phase I of the Sun ONE Implementation Roadmap is not a prerequisite for entering into Phase II; organizations just beginning to execute on their e-initiatives need not repeat the learning curve that early adopters have passed through. The Sun ONE Implementation Roadmap and Sun ONE platform easily support the exposure of existing legacy data, applications, reports, and transactions (DARTs) as Web services, even if never previously exposed as Web applications.

Following is one example of how an organization might implement a Sun ONE architecture based on the Sun ONE Implementation Roadmap.



Each iteration would focus on deploying business services that provide business value measured as return on investment. An organization's goals might align with a specific business value proposition, managed as an e-initiative, such as Company.Net as the e-initiative to deliver on the employee productivity value proposition, or multiple value propositions delivered over time in a prioritized manner. The key is to utilize the technology (Sun ONE platform and Services on Demand protocols) to deliver significant, measurable business benefits (ROI) on a timeline that makes sense for your organization.

For Early Adopter organizations, those interested in gaining competitive advantage by pushing the envelope toward a future Web services architecture earlier than their industry as a whole, a Sun ONE Early Adopter Roadmap exists that varies from the standard Sun ONE Implementation Roadmap by introducing key components of the Services on Demand architecture into the customer's advanced technology labs to enable early experimentation and development work. Early adopters can leverage the Sun ONE platform to support prototype and pilot development based on emerging technical specifications, utilizing implementations and early access code from Sun, Sun partner, and open source technology organizations, such as Apache.Org or JUDDI.Org.



Examples of ways that Sun's early adopter customers are currently leveraging the Sun ONE platform include:

- Development of UDDI and/or ebXML-based service registries
- Deployment of production Web services based on self-defined business schemas
- Collaborative development and testing environment for industry-specific, standards-based business schemas
- Web services reference testing platform to ensure solutions adhere to open standards

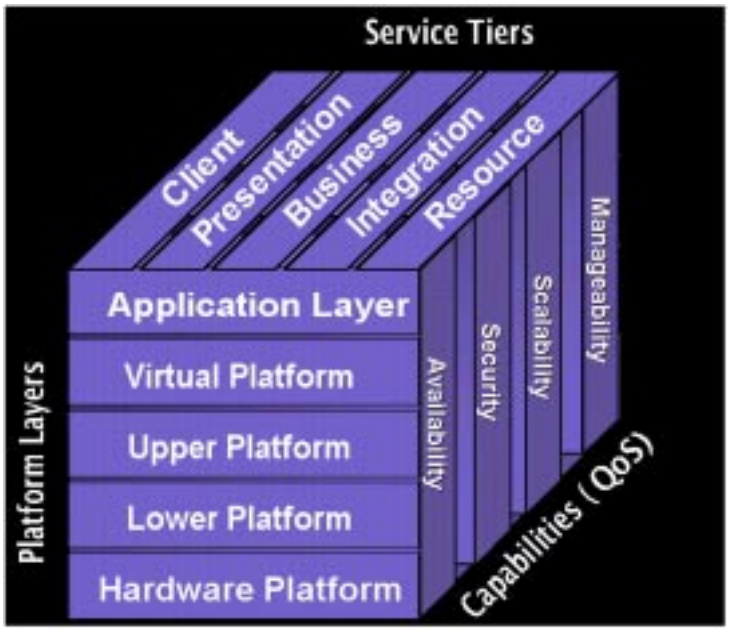
The Sun ONE architecture is flexible enough to accommodate any organization's current situation. Whether a conservative organization that is just now developing an e-strategy, or an Early Adopter with existing Web services in production, an appropriate Sun ONE implementation roadmap exists, per the Sun ONE Implementation Matrix shown in the following diagram.

	Phases		
Scenarios	**Phase I**	**Phase II**	**Phase III**
No Existing Web Applications	• Develop e-strategy • Assess DARTs and operational environment • Architect Web services • Architect directory	• Implement Web services • Implement directory • Architect future Web services	• Implement future Web services
Existing Web Applications	• Revise e-strategy • Assess DARTs and operational environment • Architect Web services • Architect consolidated directory	• Implement Web services • Implement consolidated directory • Architect future Web services	• Implement future Web services
Existing Web Services/ Early Adopter	• Revise e-strategy • Assess existing Web services • Architect future Web services • Implement Sun ONE R&D platform, including service registry	• Implement first-generation future Web services • Architect second-generation future Web services	• Implement second-generation future Web services

TABLE 1 Sun ONE Implementation Matrix

CHAPTER 4

The 3-Dimensional Framework



More details of the Sun Professional Services methodology will be provided later; the organization of the next few sections follows a 3-dimensional framework for describing Internet-facing data center architectures, developed by Sun Professional Services℠ .Com Consulting organization. The framework, sometimes referred to as the "Cube," is shown in illustration 1.

The Cube 3-dimensional framework addresses the complexities of architectural design and analysis. The top face represents the traditional segregation of architectures into "tiers" (client tier, presentation tier, business logic tier, resources (legacy) tier) and drives the construction of an "n-tier" architecture. This segregation is largely motivated by security and manageability requirements.

The front face represents the architectural levels that must be addressed in the bought and built product space, from custom application code development down through protocols and standards and operating systems and hardware.

The side face represents the measurable qualities that need to be managed to make the ongoing operations of a deployment successful.

- User-level qualities include usability and accessibility
- Service qualities include availability and performance
- Strategic qualities include scalability and flexibility
- Systemic qualities include security and manageability

A more in-depth explanation of the various tiers, layers, and qualities is available in "Dot-Com and Beyond: Breakthrough Internet Based Architectures and Methodologies," Sun Microsystems Press, June, 2001.

The next three sections of this paper outline various stages in the evolution of a Services on Demand architecture. Most enterprises possess a wide range of applications and services among their various organizations in differing architectural states depending upon a variety of business priorities and constraints. By understanding the differences between these Services on Demand states, organizations are able to assess the current state of their business processing environment and define an evolutionary roadmap that leverages the Sun ONE vision to deliver more competitive, more responsive, lower cost business services.

CHAPTER 5

Web Applications

To give a better description of how Services on Demand are deployed using specific technologies, and how the roadmap can guide an organization from one phase of development to the next without disrupting ongoing operations, the next sections present an example of the transition from current systems to a Sun ONE architecture. This section describes what a typical legacy existing environment might look like before beginning that transition.

The "traditional" legacy environment is a glass-house dinosaur pen with large mainframes running enterprise applications designed for client/server computing. While many of these applications are attempting to evolve to an Internet client/service architecture, the underlying mechanisms (such as network capabilities) may hinder Internet exposure.

A legacy system is hardly ever this isolated (although there are cases), so Web applications are typically distinguished by ad hoc, one-off solutions that are narrowly applied to one information source or legacy application. Issues such as reliability, software scalability, and availability often have to be handled explicitly in the development of business logic. Although multiple applications can be hosted on the same Internet server, hardware, and system infrastructure, the isolated nature of the legacy silo deployments is reflected in the lack of integration of application functionality.

As an example, consider how SAP[3] functionality, such as employee time sheets, payroll calculation, and accounts payable and receivable, might be exposed to the Internet. Working from the data center toward the client, the resources tier would contain the legacy SAP systems and large SQL-compliant database systems running on clustered, mainframe-class machines, such as a Sun Enterprise™ 10000 server, using fibre-channel, server-attached storage.

The SAP system would not only provide business application functionality for the exposed Web service, it would also provide security, many of the usability features, and manageability. The environment itself does little to provide the "ilities." Even where clustering is used, application-specific scripts are usually involved.

3. Sap was chosen as a common, enterprise-scale legacy application that demonstrates many of the issues involving Web services.

The integration tier would be a combination of the SAP Business APIs (BAPI) and listeners installed as part of the SAP package. This would interact with the services tier via the Internet Transaction Server (ITS), which is an adaptor to and from BAPI and BHTML (SAP's Business HTML extensions). The systems in the services tier would run on medium-sized machines (4-14 CPU). Storage requirements for this tier are relatively low and are served by SCSI server-attached storage.

From the ITS, transactions would make their way to a Web server in the presentation tier by way of BHTML and a plug-in to the API hooks of the iPlanet Web Server. The plug-in would translate BHTML to and from HTML for transmission and reception by the HTTP server component.

From a network perspective, the tiers are generally rigidly structured with intervening firewalls. Such network topologies create security problems when communications need to pass through two tiers (for example, some Java servlets that might run under the Web server in the presentation tier can send BAPI calls directly to the SAP application in the resources tier). In addition, the firewalls often create bottlenecks.

Web applications reflect the silo problem inherent in simply exposing legacy applications to the Internet. The example implementation successfully exposes SAP functionality to the network for anytime, anywhere access. It is fairly robust, but not very flexible. The proprietary protocols require duplication of effort (especially for authentication) across however many other services are being exposed. The network architecture is somewhat constrained with potential bottlenecks at the firewalls.

In short, Web applications successfully accomplish their goal — exposing some functionality to the Internet for browser access. However, they lack the interoperability and concomitant economies that can be realized when systems are capable of seamless interaction.

The following sections use the Cube framework to examine the steps needed to move from this state to a Sun ONE, past-proof, future-proof architecture. The next section is about having the right infrastructure in place, a goal that will provide returns on investment on its own. The following section describes what can be done with the environment once it is in place.

CHAPTER 6

Web Services: Building a Foundation

The first steps on the road to a Sun ONE Services on Demand architecture are putting a solid foundation in place. Before a Sun ONE program can address architecture, however, two other pieces have to be in place. One is a concrete set of business requirements for the system. The other is an adequate data center operations and management infrastructure. The importance of these two items to a successful program should not be underestimated.

Business requirements are what define what will and will not be done in the Sun ONE architecture. There are pieces that some businesses will not need, and other pieces might be "nice-to-haves." In any event, no enterprise will want to attempt to transition everything at once.

As a result, all of the things that can be done with a Sun ONE architecture must be ordered by priority (generally, expected return vs. estimated cost) and addressed in priority order. Even the choices made in installing the common Services on Demand infrastructure outlined in this section will require an understanding of the business requirements for the program.

Data center operations are equally important. Without proper management and maintenance procedures in place, even the best architecture will fail for a lack of real-world execution. Change control; rigidly controlled testing, staging, and deployment processes; and routine maintenance and disaster recovery drills are all necessary elements of successful data center operations.

In short, while the architectural principles of the Sun ONE platform increase flexibility, availability, security, and scalability, the also increase the demands on manageability. While these demands can be addressed, they can only be addressed through adequate data center operational processes.

Infrastructure Services

The cornerstones of a Web services infrastructure are resources that can be shared across a variety of services: the iPlanet Directory Server, the iPlanet Application Server, and the iPlanet Portal Server. In addition to providing crucial services more efficiently, creating a shared infrastructure from these components can also unify an enterprise's knowledge resources and provide more consistent service to its constituents.

The first step in an iteration of a Sun ONE architecture is to bring the needed directory resources into a compatible, consistent framework with the other services that will be supported. The tools for doing this are the iPlanet Directory Server LDAP Proxy Module for LDAP directory services, and the iPlanet Directory Server Meta-Directory Module for integrating legacy resources into directory infrastructure.

A shared directory service gives an enterprise a single point of registration for all of its constituents and resources. This facilitates the availability of all of the resources with all of the constituents in a secure, authorized manner, as well as creating a more usable single sign-on facility for constituents. As more and more iterations of the Sun ONE framework are executed, more resources will become available in this single directory structure.

The second step in an iteration of the Sun ONE architecture is to bring the application functionality into a shared-execution environment. The tools for accomplishing this are the iPlanet Application Server, iPlanet Integration Server, and Forte Tools.

In conjunction with a shared directory infrastructure, the iPlanet Application Server provides an enterprise with a consistent J2EE development and deployment platform for its business logic. In addition, the iPlanet Application Server also handles qualities of service, such as scalability and availability, through built-in clustering and state replication.

Finally, a shared portal server acts as both a single point for personalization and customization of information and services and provides a single point for transcoding information to provide any device access. The tools for handling this task are the iPlanet Portal Server and the various Access Packs.

What makes these iPlanet cornerstones so effective in Web services is that they are built on open standards that can be used by any Web services. Directory services are all based on LDAP; application servers on J2EE technology; and portal servers on XML. Where legacy applications are involved, iPlanet Directory Server Meta-Directory Module and iPlanet Integration Server are used to translate closed protocols into the open ones that modern Web servers are based on.

Example

Taking our SAP example and applying the Sun ONE roadmap in order to effect a seamless transition from Web applications to Web services, the first step is to create a centralized LDAP directory service. This is done by using iPlanet Directory Server for all Web services directory information requirements and connecting the iPlanet Directory Server to the SAP directory using iPlanet Directory Server Meta-Directory Module. In addition to the information stored in the native LDAP directory, all information from the SAP directory will also be available to any Web service that needs it, using standard LDAP queries.

The second step is to deploy iPlanet Application Server as a standard J2EE runtime environment. The application server provides scalability and availability automatically through clustering, and also provides a consistent runtime environment that allows all custom business logic to interact. As a result, new services can be created from reused business services that are deployed as EJB components, simply by recombining or minimally augmenting the existing services. For the SAP example, finer-grained services from SAP can be exposed to the J2EE application server environment by using iPlanet Integration Server to wrap calls and responses in XML. These calls and responses can in turn be handled by the Java API XML extensions to perform parsing, manipulation, etc.

The third step is to install iPlanet Portal Server and configure the necessary channels. Just as the directory server creates a common environment for authentication and authorization, and the application server creates a common environment for code execution, the portal server creates a common environment for interface definition, customization, and personalization. For the SAP example, channels in the iPlanet Portal Server can be defined to pull information from a variety of sources and present it in a clear, coherent manner.

Alone, these three changes to the architecture of Web services create an open, standard backbone based on LDAP, XML, and J2EE technology. As further iterations of the Sun ONE architecture are undertaken, additional business functionality and enterprise resources are converted into Services on Demand. Together with other Web services, these changes create a richer, more flexible Web services environment where any service can utilize the directory, display, and execution functionality made available by these shared services.

Iterations

An important point to note is that this process of putting a Sun ONE architectural infrastructure in place is iterative. Not every enterprise resource will have its directory information integrated into the LDAP framework at the same time. The choices of which will be done when are determined by business requirements.

Integrating the SAP system, for example, would probably be carried out in a number of phases with various groups of users and various pieces of functionality brought into the shared directory, deployment, and interface environments over time.

Once a second iteration is completed, however, the power of the Sun ONE architecture becomes apparent. Application functionality that was previously only available in and ad hoc fashion is now available to any service — past, present, or future — that is also integrated into the Sun ONE architecture.

The more iterations are performed integrating existing and new services into the Sun ONE platform, the more value will be realized from the program. New initiatives will be able to combine resources from anywhere in the enterprise with rapid time-to-market and the desired qualities of service.

SUMMARY

Beginning with shared directory, application service, and portal infrastructures, Services on Demand provide an open-standard environment where all Web services, both legacy-based and newly implemented, can share information and resources seamlessly and flexibly. In addition, these shared services remove the burden of security, scalability, and availability issues from the business logic developers, enabling them to focus on implementations of business logic that provide unique value for an enterprise.

Besides being more flexible due to integratability, the Sun ONE architectural approach to Services on Demand creates additional robustness through redundancy and availability features at the network, hardware, and lower platform layers. The advantage of the Sun ONE suite of products is the tight integration that already exists among them. Although the products are integratable with any other product supporting open standards, the tested-and-proven integration within the Sun ONE product suite reduces deployment and time to market, as well as support costs.

For the SAP example, the use of integration servers to translate both the directory information and application services to open standards allows for much more seamless integration of fine-grained SAP services with any other resources available in the enterprise.

CHAPTER 7

Future Web Services

As more iterations of the Sun ONE architecture (like the ones described in the previous section) are executed, more and more services become interoperable and available to be combined into purpose-built, workflow guiding, business applications.

In addition to iterating to integrate more resources into the Sun ONE architecture, the roadmap for moving toward Sun ONE future Web services also includes iterations for adding advanced features. The two most important sets of these features are the infrastructure for advertising and describing Web services to constituent audiences (and particularly to other Web services) and the addressing of quality of service issues by the Sun ONE infrastructure itself.

This is the future, where not only will services, like those derived from the SAP functionality, be exposed for anywhere, anytime, any device access as they are in Web services, but they will also be self-describing and advertised in an appropriate registry. The difficulty lies in addressing issues of security, availability, and billing in the context of self-describing services. Some of these issues have not been addressed with accepted standards yet, but they will soon, and when they are, the Sun ONE platform plans support them.

Extending Web Services

The evolutionary advance that Web services are leading toward is an arena where services are advertised and automatically accessed. The mechanism for doing this involves a UDDI registry that holds the locations of service descriptions. The UDDI registry is simply a directory, and initial versions of UDDI registries have already been built into iPlanet Directory Server.

Services wanting to invoke other services automatically will look in selected UDDI registries for services matching their needs. When one is found, the UDDI registry gives the location of the service's description, including rules for invoking the service, a description of what calls and returns to expect, and the security requirements for using the service. This information is given using the WSDL, ebXML, SAML, and to some extent, SOAP standards.

Once the description information has been gathered, the calling service can then interact with the desired service through SOAP and ebXML. Initial support for the current versions of these protocols has been implemented using the iPlanet Application Server, iPlanet Web Server, and the Forte Tools suite.

The ability to automatically discover and invoke services extends the reach of an enterprise's Web services to the external world. Suppliers and distributors can use published services to write their own applications to better support an enterprise, thus, a competitive advantage for a supplier becomes a competitive advantage for the enterprise.

For the SAP example, Forte tools could be used to write services calling on SAP services through standard J2EE technology. These services can then be wrapped in UDDI/WSDL/ebXML descriptors (as these protocols are finalized), and made available to the desired constituents. More to the point, once SAP functionality is advertised to the desired constituents, new services can find and access the SAP functionality and combine it with other services from PeopleSoft functionality or custom services in a seamless, advertised Web service. The value of the SAP system and all of the other integrated systems then goes beyond the sum of their individual values.

Qualities of Service

So far, the emphasis of the Sun ONE architecture iterations described has been toward increasing the functionality of the various services offered. This has been done by creating a shared framework, integrating all resources into that framework, and then extending that framework to support emerging protocols for automatic discovery and consumption of Web services.

But a significant part of the power of the Sun ONE architecture is that the iPlanet servers provide qualities of service for the business logic, independent of business logic implementation. The initial infrastructure described in the previous section illustrates how some of these qualities of service are put in place.

The iPlanet Directory Server, including the policy manager and delegated administration, forms the basis for a single authorization and authentication mechanism that can then provide single sign-on services to any service in the Sun ONE framework.

Similarly, the iPlanet Application Server provides scalability and availability through software clustering that includes load distribution as well as replication and backup of transaction state. Thus, a transaction can continue even when hardware is overloaded or fails.

The iPlanet Portal Server takes care of the interface issues, including searching and transcoding for any device access. Together with the iPlanet Web Server, they eliminate the need to integrate interface code into business logic.

At the lower platform and network layers, bandwidth and resource (e.g., CPU and memory) allocation can be provided to guarantee minimal levels of performance. This is an important step in the profitable use of service level agreements. Availability and security can also be enhanced through system level configurations such as TCP service wrappers and IP multipathing.

Beyond these qualities of service, monitoring and metering will be provided as a callable service within the Sun ONE environment. This will allow for better management of services, better measurement of service use, and billing for service use. Again, these services will be available without forcing changes to business logic that is already integrated into the Sun ONE environment.

EXAMPLE

Changes to the SAP configuration will be very small for iterations through this part of the Sun ONE architecture. Instead, the capabilities and the qualities of service of the SAP functionality will be extended and enhanced externally to the business logic.

By advertising parts of the SAP functionality in UDDI registries, partners can plug into an enterprise's ERP system to optimize supply chain management. Suppliers and distributors can write services based on the enterprise's SAP functionality that can create value for both the vendor and the enterprise itself. All of this can be done without modifying the SAP implementation or business logic.

Further, the security, availability, and scalability can be set to any level desired, again without changing the SAP business functionality. If higher levels of security are desired, the single sign-on system can be configured to require certificates or VPN connections for access to SAP functionality.

If higher levels of scalability are required, the J2EE containers can be expanded and partitioned, again without affecting the business logic. Similarly, availability can be enhanced by adding more redundancy to the server clusters supporting the Sun ONE environment.

SUMMARY

Many of the protocols providing the capabilities described in this section are not fully defined yet, but we have enough proofs of concept to know these scenarios work. The crucial elements are support for open standards and the separation of quality-of-service issues from business logic.

For the SAP application, not much has changed on the inside. On the outside, though, thanks to services registries and description, the SAP functionality can be advertised, described, and therefore, made available to be flexibly combined with other Web services to create a wide variety of new services. Thanks to the separation of qualities of service and business logic, additional service levels can be implemented without affecting the business logic.

CHAPTER 8

The SunTone Architecture Methodology

Having described the Services on Demand evolutionary steps, from early-generation legacy and Web applications to Web services, this section describes the methodology Sun Professional Services uses when engaging Sun ONE customers. This methodology combines with project management methodologies and operational methodologies to provide requirements to deployment support for Sun ONE projects.

The SunTone Architecture Methodology defines essential practices common across any project, allowing for an increasing level of complexity as required.

SunTone Architecture Methodology is built on three foundational themes:

- Use-case Driven. Use cases provide a means for capturing requirements, organizing activities, and keeping the entire team focused on the end result.
- Architecture Centric. The central technical activity is architecture, which is developed and validated early; the rest of the system is built around it.
- Iterative and Incremental. The bigger system is evolved from a series of smaller systems, each of which extends the other.

The SunTone Architecture Methodology also draws heavily on the foundational ideas of the Unified Process[4]. Our diversion from the UP is primarily in the area of architecture. The UP discusses useful, but generic ideas on the subject. In the SunTone Architecture Methodology, we enhance the meaning of architecture -centric by incorporating pattern-based reasoning based on systemic qualities. In addition, the SunTone Architecture Methodology attempts to keep the overall process description smaller and more manageable.

Large organizations typically create programs as staffed, budgeted and goaled initiatives that cross group boundaries. Programs are typically long lived (years, not months) and spawn multiple projects as a means to achieving the overall program goals. An example of a program would be eSun, Sun's corporate-wide initiative to e-enable all business processes.

4. The Unified Software Development Process, I. Jacobsen, G. Booch, J. Rumbaugh, Addison-Wesley 1999

The SunTone Architecture Methodology is definitely not a program management methodology, but can be utilized to help deliver on the goals set forth in a program. It does this by providing a flexible framework for the definition and realization of one or more projects needed to develop a dynamic technical architecture that can be modified and enhanced, iteratively and incrementally.

Key concepts embodied in the SunTone Architecture Methodology, and described further in the following sections of this document, are:

- Phases
- Workflows
- Milestones
- Artifacts
- Iterations

Project Phases

The SunTone Architecture Methodology builds on the concepts of phases and workflows from the Unified Process (UP). A phase is all about focus and priority. At any given time, any kind of activity could be going on within a project, but at different times, the maximum payoff comes from being focused on key issues. The partitioning of the project timeline into phases serves to clarify and emphasize these priorities both internally and externally to the project. Each phase is defined by the artifacts that constitute its deliverables, which in turn drive the activities that must occur within that phase.

Four phases are defined for each product release, which proceed in order.

Inception is the first phase, during which the scope of a project is defined and its risks and major milestones are estimated. Understanding scope involves a certain amount of exploration and documentation of the systems requirements, but only enough at this time to understand scope and prepare for the next phase. Inception formalizes the vision, estimates what it will take to get there, and outlines how to know when and if success has been achieved.

Elaboration follows inception. Elaboration has two primary threads, one that focuses on architecture and the other on fleshing out requirements that were outlined mostly breadth-first in inception. Of these, architecture usually requires the most expenditure of effort and lies on the critical path. This is due to not only its definition, but also its validation based on constructing an architectural prototype. Elaboration is truly over only when there are no significant risks or unknowns remaining. This makes the milestone at the end of elaboration the most important of all, since the greatest allocation of expense is yet to come.

Construction follows elaboration. This is where the bulk of functionality will be built on the stable foundation established in elaboration. Senior team members can be added or removed for this purpose, since the predictability and foundation established during elaboration ensures that economies of scale can be achieved.

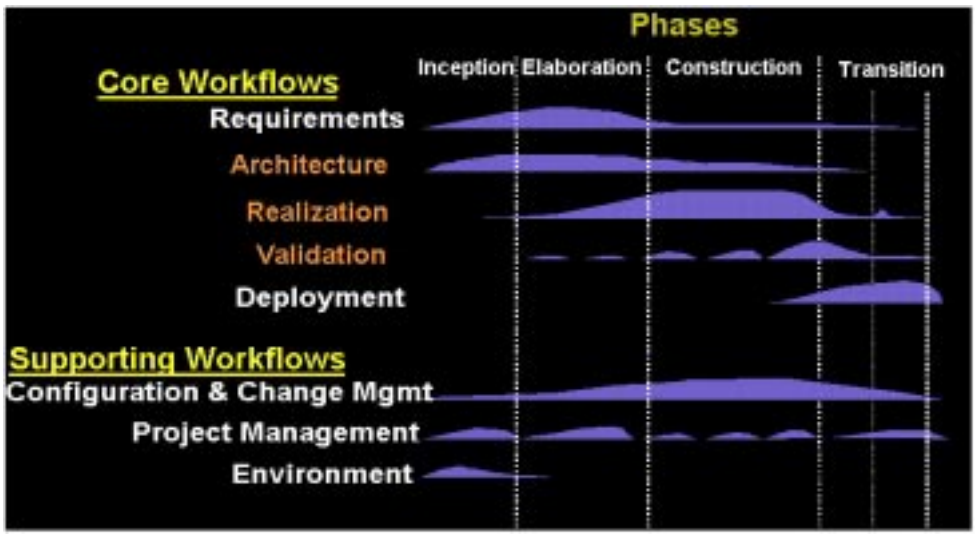
Transition initiates the final preparations for making the system production ready, and simultaneously prepares the user community for its use. This includes activities such as the following:

- Finalizing the entire suite of required artifacts, such as documentation and training materials
- Executing a possibly incremental rollout, including ETL from replaced systems
- Overall quality assurance drawing from field experience

Transition usually begins with a beta test period to a restricted number of users, and ends with an official system release. Following transition, the project enters sustainment, maintenance, and optimization. Optionally and simultaneously, a new release cycle may be initiated beginning with inception, possibly even before the previous cycle (ending in transition) has completed.

WORKFLOWS

The activities involved in building a system tend to be cohesive in terms of their interactions with other activities as well as the artifacts that are produced as a result. We call these groupings workflows. The entirety of a given iterations work can be partitioned across well-defined workflows. With some exceptions at the ends of a project, each workflow is more or less active within each iteration. As the project progresses, the relative amount of expended effort in each workflow varies as illustrated in the figure below.

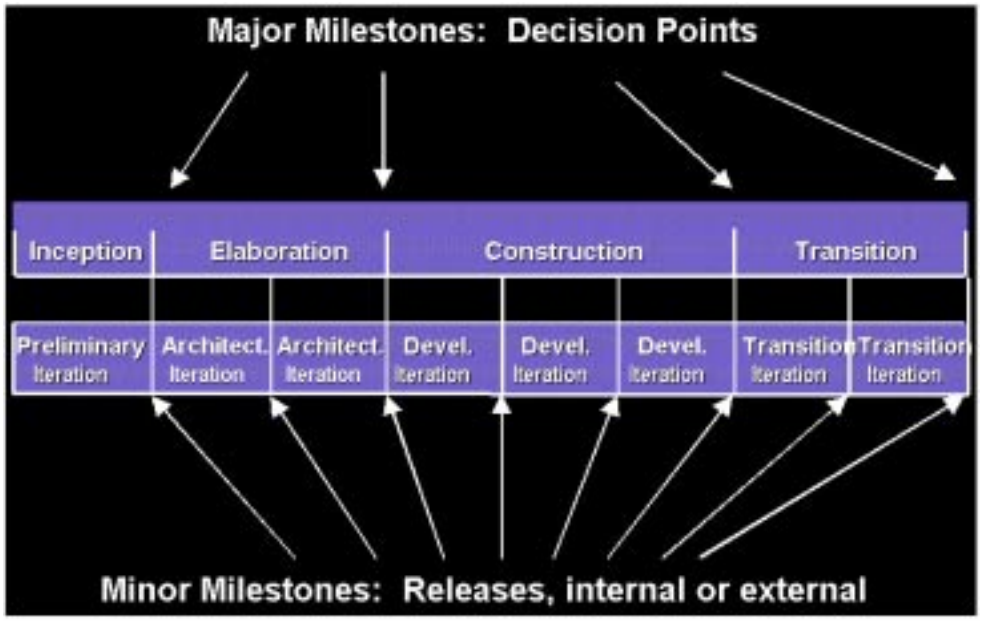


Important workflows that are part of the SunTone Architectural Methodology include:

- Requirements: The requirements workflow involves the collection and/or definition of necessary business, functional and service-level attributes of the system to be developed.
- Architecture: The architecture workflow involves analysis of requirements and application of architectural patterns to identify a solution framework that is also capable of accommodating new requirements over time.
- Realization: The realization workflow involves the transformation of units with well-defined responsibilities into designed, working, and tested system components.
- Validation: The validation workflow involves testing to determine whether realizations conform to requirements.
- Project Management: The project management workflow involves project resource and timeline estimation, deliverable and milestone planning, progress tracking, and risk management activities.

MILESTONES

The transitions between phases are considered major milestones. The end of each phase is accompanied by a go/no-go decision about whether to proceed to the next phase. No-go might mean that the preceding phase must be extended, or it might mean that the project should be terminated in its current form. Extending a phase does not mean that all work that would otherwise have been done in the next phase must be delayed. It simply means we are really not ready yet to change our primary focus in this project. Making all stakeholders aware of the current phase helps communicate what the primary focus is. Conversely, prematurely forcing a phase transition can delay the resolution of the preceding phases goals.



ARTIFACTS

Artifacts are things that are produced. This could refer to a single class or type, a package, a model, the whole design model, etc. A document is a kind of aggregate artifact suitable for printing. Most commonly, we use the term artifact to reflect the larger or aggregate variants that might be specifically identified as project deliverables.

Examples of artifacts utilized in typical Sun Professional Services architecture engagements include:

- Project Plan
- Vision Statement
- Requirements Document
- Architecture Framework
- Prototype
- Test Plan
- Construction Plan

Patterns

A pattern is a recurring solution to a problem in a context.

- A context is the environment, surroundings, situation, or interrelated conditions within which something exists.
- A problem is an unsettled question, something that needs to be investigated and solved. It can be specified by a set of causes and effects. Typically, the problem is constrained by the context in which it occurs.
- The solution refers to the answer to the problem in a context that helps resolve the issues.

Patterns are significant within the SunTone Architectural Methodology in that they are utilized heavily, but not exclusively, in the architecture workflow of the methodology. The SunTone Architectural Methodology utilizes well-defined architectural patterns, based on Sun Professional Services work with many clients across all industries over many years, providing a mechanism for collective experience to be leveraged by all Sun clients.

In the Services on Demand space, architectural patterns are just now beginning to emerge and will, along with J2EE patterns catalogued by the Sun Java Center℠ program, be captured and disseminated to clients and industry professionals. Additional information can be found the book “Core J2EE Patterns Best Practices and Design Strategies.” Prentice Hall, June 2001.

Iterative Refinement

Within phases, and particularly the middle phases, work is organized in terms of iterations. Iterations provide a way of treating system development as many small releases (internal or external) in place of one big release. The product of each iteration is an executable mini-release, built upon the release of the previous iteration, such that a system is grown toward its target.

CHAPTER 9

Sun Professional Services Capabilities

Sun Professional Services addresses client needs by offering a set of Sun ONE Core Services, based upon the SunTone Architectural Methodology, that provide a quick, cost-effective, and efficient means for delivering solutions to challenging business problems. These services offerings include:

- Sun ONE Architecture Assessment Service: A 4-6 week project, led by Sun Professional Services architects and/or consultants, involving an assessment of an organization's technical architecture, with the focus on identifying best practices as well as opportunities for enhancement to support evolution toward a Sun ONE Services on Demand architecture.
- Sun ONE Workshop: A 3-day onsite workshop, facilitated by Sun Professional Services architects and consultants, focused on assisting the client in the definition of their e-strategy, a strategic plan for moving to a Sun ONE Services on Demand architecture.
- Sun ONE Inception Service: A 6-8 week project, led by a Sun Professional Services Project Manager and a senior architect, focused on definition of an Architecture Framework that addresses the business, functional, and service-level requirements of an organization's Sun ONE strategic plan. Additionally, an Elaboration Plan, including a Prototype Plan, is developed as a part of this service.
- Sun ONE Elaboration Service: An 8-12 week project, composed of multiple iterations and led by Sun Professional Services project manager and senior architect(s), focused on refinement and prototyping of the Architecture Framework developed in the Inception Phase, as well as creation of a comprehensive Construction Plan.
- Sun ONE Construction Service: A 4-8 month project, depending upon scope defined in the Construction Plan, involving Sun Professional Services architects acting as mentors and in quality assurance roles to ensure the integrity of the Sun ONE architecture as it is constructed by Sun architects, engineers, and SI and ISV partners.
- Sun ONE Transition Service: A 4-8 week project, led by Sun Professional Services architects and engineers, focused on knowledge and skills transfer to enable client staff or consultants on the ongoing maintenance and support of the delivered system(s).

Sun Professional Services engagement managers are adept at understanding industry and client-specific challenges and crafting a service offering to meet each client's unique needs.

CHAPTER 10

Conclusions

Web services have been around as long as there have been Web servers whence to deploy them (over 8 years). Sun has developed an excellent track record in the deployment of reliable, secure, manageable, scalable Web services and Services on Demand. Sun have been at the forefront of the evolution away from simplistic Web servers of the early 90s to the multitiered, multilayered, quality-of-service focused, services-driven architectures involving application servers, directory servers, messaging servers, etc. that one finds in any serious Web server environment in operation today.

Other vendors are making a lot of noise about the new emerging standards. To hear them talk, one might well think that these emerging standards are the whole story and that to support them, one must dump one's entire enterprise software stack and even programming language. Sun offers a more thorough, less disruptive, evolutionary approach. Sun ONE products enable the easy deployment of Services on Demand based on the emerging open, intervendor standards. Sun has been active in the development of these standards and pledges to adhere rigorously to them as they are adopted.

But the emerging Web services standards are just a small part of the big picture. Whether Web services are based on new or existing standards, the scalability, reliability, and manageability issues that have made the real difference to the success of Services on Demand in the past will continue to decide the game in the future. The Sun track record in this area, as well as our integrated hardware and software expertise, continue to make Sun the implementation partner of choice for cost-effective Services on Demand that work, whatever the underlying standards may be.

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303

1 (800) 786.7638
1 (512) 434.1511

Sun – http://www.sun.com
Sun ONE – http://www.sun.com/sunone
Sun Professional Services – http://www.sun.com/service

October 2001
FE1682-0