

**A Field Guide to
Services On Demand
and Sun™ ONE**

**By Diana Reichardt
Sun ONE Interactive Marketing**

September 2001

Table of Contents

1.0 Services On Demand

2.0 What Are Web Services?

- 2.1 This Really Is Different
- 2.2 Components vs Web Services

3.0 Market Drivers

- 3.1 Device Market Exploding
- 3.2 PC Market Stagnant
- 3.3 Desktop vs Beltloop (Mobile/Wireless)
- 3.4 Intel Is Not Inside
- 3.5 The Non-Windows Metaphor
- 3.6 Transactions Replace Office Applications

4.0 The Value Proposition

- 4.1 Richer User Experience
- 4.2 Reduced Costs And Increased Flexibility
- 4.3 Agile Business Relationships

5.0 The Business Model For Web Services

- 5.1 Business-to-Consumer (B2C)
- 5.2 Business-to-Business (B2B)
 - 5.2.1 A B2B Example

6.0 The Language Of Web Services

- 6.1 XML: eXtensible Markup Language
- 6.2 ebXML: electronic business XML
- 6.3 UDDI: Universal Description Discovery Interoperability
- 6.4 Registries And Repositories
- 6.5 SOAP: Simple Object Access Protocol
- 6.6 WSDL: Web Services Description Language
- 6.7 XML + Java Enable Web Services

7.0 The Competitive Landscape

- 7.1 What Is Hailstorm?
 - 7.1.1 Instant User Base
 - 7.1.2 The Hailstorm Business Model

8.0 What Is Sun ONE?

- 8.1 The Sun ONE DART Diagram
- 8.2 Sun ONE Road Map
- 8.3 The Role Of Partnerships

9.0 Conclusions

1.0 Services On Demand

IT departments equipped to manage and control tightly coupled, monolithic applications running behind a firewall are faced with a new reality in the software world: The need to integrate business processes across corporate and geographic boundaries that can not be fully prescribed or anticipated.

Services on Demand is derived from the recognition that many different types of services will be delivered over electronic networks, including a new breed of applications that extend well beyond the tightly coupled services available today. Services on Demand will facilitate the delivery of anytime computing, anywhere, to anyone, using any device, and they will quickly evolve to include support for emerging Web services and new peer-to-peer applications.

Sun provides the end-to-end architecture that will enable customers to support their existing application needs while building a solid foundation for Web services of the future. The Sun™ Open Net Environment (Sun ONE) platform is an open, integratable stack designed to create and deploy Services on Demand and emerging Web services.

Much of the foundation software required to enable Services on Demand is already available. Still, the ability to fully deploy Web services will require an approach to computing that is quite different from the client-server model most applications use today.

2.0 What Are Web Services?

Web services are self-describing components that can discover and engage other Web services or applications to complete complex tasks over the Internet. Unlike the hard-wired applications of client-server computing days, these loosely coupled software components can dynamically locate and interact with other components on the Internet to provide a service, without intervention.

These flexible software components can interact in a new distributed computing model that extends across company boundaries. This raises the prospects for collaboration, as companies contemplate pairing their Web service offerings with partners to provide best-of-breed solutions to customers on demand. It also highlights a new set of challenges as predictability and control are rapidly replaced by a complex Web of dynamic Internet transactions.

2.1 This Really Is Different

Application protocols, the rules that describe the data that flows across the Internet, are fundamentally different for Web services. Client-server computing was designed around the intranet model of computing, where a server sitting on the third floor of a building could service 500 workstations located on the second floor. Software development in this environment was based on components operating inside the firewall, using technologies like DCOM, CORBA, J2EE, and EJB. This model of computing wasn't designed to let a Palm VII communicate with an Internet service like MapQuest.

First generation Internet services typically use transport protocols like TCP/IP in combination with inflexible application protocols. But Web services rely on a different kind of distributed computing program model that is both flexible and extensible. Web services primarily use HTTP as a transport mechanism and XML as the application protocol. As a result, one of the main differences between Web services and client-server applications in the distributed computing realm relates to the protocols used. Web services will build upon, and coexist with traditional distributed computing systems.

2.2 Components vs Web Services

Web services provide an abstract model that describes a mechanism for a producer of a service to advertise something a consumer can find. Still, code needs to be written to implement a service. And even though object-oriented programmers may view Web services as just another component model, components and Web services differ in two very important aspects: granularity, and coupling.

Granularity refers to the complexity of the description of a service. A simple UNIX call can be very specific and thus have a *fine* level of granularity. But a purchase order from a manufacturer to a supplier may have global implications that introduce levels of complexity that are less well defined, resulting in a degree of granularity that is more *coarse*.

Coupling refers to the nature of the interface between the producer of the service and the consumer of the service. It is concerned with the impact on the consumer of the service, if the implementation changes. For example, if Sun decided to change the Solaris "GetTimeOf Day" call, every application that used that call would have to change. This is an example of a tightly coupled application relationship.

As it turns out, components are very good at doing fine-grained, tightly coupled kinds of things, while Web services are very good at doing loosely coupled, coarse-grained kinds of things. Another way of thinking about it is that components (tightly coupled) are capable of interoperating with other components designed using the same APIs or object model. Whereas Web services (loosely coupled) are designed to interact with any other component, regardless of its origin, as long as it is encapsulated in a self-describing XML wrapper.

For programmers, this creates something of a boundary in the design space. The main reason that this partition occurs is tied to the characteristics of the transport mechanism, TCP/IP or HTTP. So for programmers, the question of whether a component, or a Web service is required really depends on the nature of the service request. Web services won't completely replace software components—they will expand on, and complement them.

Component-Based Model	Web Services Model
Tightly coupled software applications	Loosely coupled software applications
Intranet-based software	Internet-based software
Designed for processes within the enterprise	Designed for processes across enterprises
Transport mechanism TCP/IP	Transport mechanism HTTP
Fine granularity	Coarse granularity

3.0 Market Drivers

3.1 Device Market Exploding

Services on Demand are being driven by a number of trends. The first important trend is tied to the widespread adoption of mobile devices, like cell phones, PDAs, and pagers, which is driving demand for *new* software services that can be delivered over the Internet. The demand for so-called "smart" handheld devices is increasing at a rapid pace. According to Forrester Research, the world only has about 450 million PCs, but the extended Internet, including embedded chips, will increase the Internet population by billions of nodes.¹ It's clear that the number of devices coming online will outpace anything we have seen before. These devices, or the next wave of "users," will drive exponential growth in Web services. Gartner Research estimates that the software component of Web services (excluding services) will represent a \$1.7 billion market by 2002.²

Metcalfe's Law states that the value of a network increases exponentially with the number of users. Also referred to as the Net effect, the premise here is that the bigger the network, the greater the potential value provided to those participating in the network. A familiar example is our telephone system, which only has value to users because of widespread participation in the network. The same will be true for a range of newly networked devices.

3.2 PC Market Stagnant

Secondly, the decline of PCs and shrink-wrapped software is forcing companies, like Microsoft, to abandon their focus on the delivery of perpetual program licenses, and make the shift to the delivery of subscription services.

¹(Forrester Research: The X Internet, By Carl D. Howe, May 2001)

² (Gartner Research: The Future of Web Services: Dynamic Business Webs, April 2001)

Office applications will continue to facilitate our word processing and spreadsheet application needs, it's just that soon, these services will be provided over the Internet as modular services.

3.3 Desktop vs Beltloop (Mobile/Wireless)

Even as we are increasingly mobile, we are communicating more than ever. This is especially true for workers who are frequently away from their desks, communicating via text and voice messages. Unlike desktop software which makes assumptions about the user (i.e., that it knows the user's fixed location and system profile), beltloop devices, like cell phones and pagers, have no predictable configuration. That means that the services that work with these devices are inherently different. They will require the ability to dynamically modify their user interfaces, deliver modular functionality, and autonomously identify themselves to other components and devices.

3.4 Intel Is Not Inside

Consumers have long been the beneficiaries of embedded systems from a variety of manufacturers, mostly unaware. Chips are routinely installed in consumer devices, like cars and cell phones, without much fanfare. Yet, when it comes to the PC, the assumption, mostly, has been that it's Intel inside. Now that companies like Nokia, Motorola, and Ericsson are supplying much of the underlying technology for a growing number of networked devices, the Intel architecture is no longer a given.

3.5 The Non-Windows Metaphor

A Web service is a programmatic interface between a producer of a service and a consumer of a service. As such, Web-based services will bypass browser-based PC applications completely, using new user interfaces like speech-to-text that have no relationship at all to the Windows metaphor. In fact, the operating system for Web devices is not only likely to change, it's bound to be irrelevant as new communication protocols dictate standards for the exchange of information from virtually any device.

Furthermore, communications and Web-based interactions will no longer be limited to a Windows frame on a computer screen. Text messages are already delivered to cell phones and pagers. Also, onboard communications and navigation systems such as the OnStar service, offered by General Motors Corp., are using wireless technology and the Global Positioning System Satellite network to link drivers to service advisors 24 hours a day, 365 days per year.

3.6 Transactions Replace Office Applications

Microsoft's biggest developer group is comprised of Visual Basic (VB) programmers; a community estimated at over 3.3 million³. VB programmers write sophisticated desktop, or fat-client applications. For example, brokers and traders that require adaptations to their desktop applications might

³ eWeek, Visual Basic .NET: Is It Too Complex?, Roberta Holland, June 15, 2001

engage VB programmers to integrate their office applications with some other back-office system. VB is designed to be a client-side programming system. But in a world of Web services, applications will be replaced by transactions, which will require different programming skills. The transition to Microsoft's Visual Basic .NET is acknowledged as a difficult and complex task for many developers.

4.0 The Value Proposition

Services on Demand hold promise for corporations interested in boosting ROI (Return On Investment) and increasing profits through streamlining operations, automating the supply chain, reducing time-to-market, and implementing direct selling models to mass markets. Services on Demand can enable companies to market their core competencies over the Internet to provide new Web-enabled revenue streams. This will result in profit models tied to subscriptions, term licenses, royalties, and transaction-based fees.

4.1 Richer User Experience Through Personalization

Users will benefit from a richer experience through the personalization of requested services. Services on Demand have the potential to put users in context, which is to say that they will have knowledge of the user's profile and preferences, and place in time. This knowledge about the user's preferences can be used to determine which services are delivered, to whom, in what form, and at what time. Sending an HTML newsletter to a cell phone doesn't make sense, but a short notification message that a newsletter is now available might be fine, assuming the user requested it. Personalization will also provide an added level of intelligence that will make it possible for applications to combine services and information automatically to deliver exactly what the user is interested in.

4.2 Reduced Costs and Increased Flexibility

The business trend towards buying software rather than building it is gaining in popularity because of the reduced costs and increased flexibility this allows. The Internet makes it easy for companies to outsource all or part of their enterprise requirements, which in turn makes it easier to change services or switch vendors as required. As companies interact with partners to provide Services on Demand in real-time, a flexible infrastructure that can adapt rapidly, favors faster time to market, more efficient processes, and less costly solutions for continued long-term business growth.

4.3 Agile Business Relationships

As Web services are implemented, Gartner Research predicts that the role of partnerships will increase the desire for agile business relationships and force "operational dynamism" to become a primary requirement for system implementations. They see the concept of the private exchange as a mechanism for community interaction that offers a degree of control and limits the community to a select group of participants. Gartner predicts that

through 2004, the adoption of private service registries will outpace public service registries by more than a 2-to-1 ratio.⁴

5.0 The Business Model For Web Services

Services on Demand will address two broad markets: Business-to-Consumer (B2C), and Business-to-Business (B2B). While the underlying technology and software delivery mechanisms remain the same, the end user, and the resulting benefits will vary. A significant business opportunity will exist for companies that can broker the exchange of software services over the Internet in both of these models.

B2C Web services, which are user-centric, are concerned with the delivery of personal productivity tools and products delivered to mass markets. B2B Web services will focus on streamlining operations, automating the supply chain, improving ROI for both company-facing, and customer-facing applications, and delivering revenue generating services on the Internet.

5.1 B2C Web Services

As Web services take hold, consumers will have the opportunity to tap into much more than the simple convenience of Web-based retail shopping. Soon, a finely tuned set of personal preferences will facilitate the exchange of a series of cascading transactions, capable of operating automatically on behalf of the consumer. Based on an individual's preferences profile, the purchase of an airline ticket could automatically trigger a hotel and car reservation, which in turn could fire off an email notification that the reservations have been booked. Additional opportunities exist in the entertainment area, where game enthusiasts will have the opportunity to interact with other gamers online in a variety of computer-activated scenarios.

5.2 B2B Web Services

Still, most research firms agree that viable business applications for Web services will arrive much sooner than consumer applications. In its Dynamic Hosting Update⁵, Chase H&Q says that in the Internet-driven transformation of the business landscape, companies will find ways to encapsulate their core market position (loan syndication, retail distribution, and package delivery) into accessible Web services. These services will then have the ability to combine and create a larger network of Web-based services.

5.2.1 The Online Loan Request: A B2B Example

An online loan request is a good candidate for a B2B Web service since it fits the definition of a service that can be requested and transacted entirely over the Internet. It also provides an example of a request that could be made just as easily using a cell phone, or a PDA. Even a fairly light treatment of the

⁴ (Gartner Research: Private Exchange and Web Services: A Match That Matters, March 2001)

⁵ (Chase H&Q, Equity Research: Dynamic Hosting Update, March 2000)

business processes involved demonstrates the efficiencies of a completely electronic series of programmatic events from a business perspective.

Consider this scenario:

1. An applicant running a small business gets online to apply for a \$300,000 loan and requests five bids from online banking institutions offering business loans at rates below 6.5 percent.
2. Five bids are returned to the applicant that can be completed and submitted online.
3. The applicant requests a loan application from the banks with the lowest rate, then completes and submits the application online.
4. The bank receives the application and a program automatically deconstructs the loan application into component parts that can independently connect with other Web services to perform the necessary credit checks and business verifications. A loan identification code is included with each online service request.
5. Each of the independent Web services returns its results to the bank, along with a loan identification code, and an invoice for the service just completed. Another program kicks in to recombine these results and then sends that information to a decision-based program that makes the final determination to accept or reject the loan.
6. Once the loan is approved, the bank can automatically trigger a wire transfer that deposits the \$300,000 loan amount directly into the company's bank account, along with a notification to the applicant (wherever they may be) that the money is now available.

This is just one example, but it illustrates the speed and efficiency that Web services can offer. Giga Information Group expands on the B2B opportunities for companies by describing four main categories of Web services for businesses:⁶

1. Packaging an existing capability as a Web service.
2. Acting as an integrator of Web services.
3. Providing a granular Web service that other companies can use.

⁶ (Giga IdeaByte: Web Services Will Open Opportunities For New Business Models, April 27, 2001)

4. Enhancing, supplementing, or replacing its own internal systems with Web services.

Giga also predicts that the availability of off-the-shelf business service components will rise steadily during 2001, and that core business components will be delivered as collections, or “frameworks” of components. Giga analyst Randy Heffner states that the growth of e-business is forcing many enterprises to adopt multi-tier, Web-enabled architectures that encourage the use of server-based business components. He adds that e-commerce vendors are the largest contingent adopting component strategies.⁷

6.0 The Language Of Web Services

Web services require new levels of intelligence that go beyond HTML and TCP/IP to support the communication of information between unrelated systems and devices. Rules governing the discovery, description, and exchange of information are being addressed by emerging standards like XML(eXtensible Markup Language), ebXML(electronic business XML), UDDI (Universal Discovery Description Interface), SOAP (Simple Object Access Protocol), and WSDL (Web Services Description Language). The goal is to eliminate communication barriers between a myriad of devices and systems, and facilitate the delivery of Web services in a vendor neutral way.

Gartner Research has identified five initial requirements for a Web service⁸ platform:

1. **Discovery:** The mechanism by which services make themselves known and are discovered.
2. **Description:** The way in which specifications are made for what information is passed into and out of the service.
3. **Transport:** The communication method between the user of the service and the service itself.
4. **Environment:** The runtime in which the service executes — the e-business platform describes the environment.
5. **Event Notification:** A mechanism by which a service can be invoked as part of a series of event notification queues.

6.1 XML (eXtensible Markup Language)

XML adds intelligence to Web components because it *describes* the data content through the use of metadata, or tags. This will allow Web services to

⁷ (Giga Planning Assumption: Key Trends for 2001: Off-the-Shelf Components and Component Marketplaces, December 2000)

⁸ (Gartner Research Note, Web Service Platforms: A Brief Look at Pioneer Vendors, October 2000)

communicate with each other by exchanging messages expressed as XML documents. XML uses plain text for data representation, which makes it easy for both humans and search engines to read.

Any service that can process an XML file and output messages in an XML format can communicate with any other service or application that can do the same. This type of architecture enables a new level of interoperability and virtually limitless options for the creation of value-added services.

6.2 ebXML: (electronic business XML)

This is a complete B2B framework that enables business collaboration through the sharing of Web-based business services. The framework supports the definition and execution of B2B business processes expressed as choreographed sequences of business service exchanges. The ebXML Initiative is a joint project of UN/CEFACT (the United Nations body for Trade Facilitation and Electronic Business) and OASIS (Organization for the Advancement of Structured Information Standards). The ebXML membership includes representatives from more than 2000 businesses, governments, institutions, standards bodies, and individuals from around the world.

6.3 UDDI (Universal Description Discovery and Interoperability)

This is an industry consortium lead by Accenture, Ariba, Commerce One, Compaq, Edifecs, Fujitsu, HP, I2, IBM, Intel, Microsoft, Oracle, SAP, Sun Microsystems, and VeriSign. More than 130 companies have joined the UDDI initiative. The group is developing specifications for a universal, Web-based business directory called the UDDI Business Registry.

6.4 Registries and Repositories

Nearly all distributed computing systems have some naming service, which is essentially the component that lets the producer of service publish the existence of that service in a well known place. A potential consumer of a service can go to this well-known place, look up a service, and find out how to connect to it. UDDI and the ebXML Registry/Repository, are two different global naming services that can act as a registry and repository for Web services. (Initially Sun will focus on supporting UDDI, but support for the ebXML registry/repository is expected in the future, because it provides a much richer schema.)

6.5 SOAP (Simple Object Access Protocol)

SOAP is technology developed by DevelopMentor, IBM, Lotus, Microsoft, and Userland. SOAP provides an extensible XML messaging protocol and also supports an RPC programming model. A number of SOAP implementations are available. The two most popular implementations are an open source Java technology implementation from the Apache Software Foundation and a Microsoft implementation within the .NET SDK.

6.6 WSDL (Web Services Description Language)

This is technology developed by Ariba, IBM, and Microsoft. It specifies a common XML framework for describing a Web service. At this time, IBM has released a WSDL for Java technology developer toolkit through IBM alphaWorks.

6.7 XML + Java Enables Web Services

XML and Java are two key enabling technologies for the loosely coupled architecture of Web services. Together, they provide a standardized, platform independent, development and deployment environment for Web services and electronic business transactions. Java provides a portable, platform-independent software execution environment, while XML provides portable, platform-independent data.

The J2ME, J2SE and J2EE platforms integrate XML technology at a core level that ensures vendors and customers freedom of choice for products, services and components, eliminating the risk of vendor lock. Java 2 and XML together enable customers to assemble standards based, best-of-breed solutions for services and products including integrated software stacks from multiple vendor's products.

7.0 The Competitive Landscape

According to Gartner Group⁹, the notion of Web services really started when Hewlett-Packard first introduced its “e-services” campaign. Since then, IBM and Microsoft have teamed as a so-called “gang of two” to define new Web services standards in an effort to revitalize their respective companies’ software strategies.

With the decline in PC sales and Windows software, Microsoft needed a strategy that would ensure new revenue streams for the company. While IBM seemed determined to devise a software strategy that could guarantee a future for its mainframe and AS/400 applications. As a result, Microsoft and IBM collaborated on standards such as SOAP, UDDI, and WSDL in an effort to take an early lead in the definition of Web services.

In June 2000, Microsoft introduced its .NET platform strategy with support for Web services and the development of service-oriented applications. Soon after, in November 2000, IBM announced its road map for the support of Web services, which included plans to embed the SOAP, UDDI and WSDL protocols into its middleware products (e.g., WebSphere Application Server, and MQSeries) and in DB2, Lotus Domino and Tivoli).

Sun followed suit when it announced its Sun ONE software strategy at an analyst conference in February 2001. As with Microsoft and IBM, Sun ONE

⁹ (Gartner Group, Will Web Services Standards Ever Happen?, April 3, 2001, M. Pezzini)

products support the new Web services standards such as XML, SOAP, and UDDI. In addition, Sun ONE also supports ebXML and key distributed computing standards upon which Web services are often built, like LDAP and Java.

7.1 What is Hailstorm?

Hailstorm is part of Microsoft's .NET initiative. It represents the company's vision for a user-centric Web services architecture and a set of XML-based Web services. Microsoft describes Hailstorm as a set of services that helps manage and protect personal information and interactions across a range of applications, devices, and services. Hailstorm services will be accessed through SOAP and XML, which Microsoft claims makes these services truly open.

The initial set of Hailstorm services will include consumer-focused applications like: myAddress, myProfile, myContacts, myLocation, myNotifications, etc. According to Microsoft, these Hailstorm services for .NET will deliver personally relevant information, "through the Internet to a user, to software running on the user's behalf, or to a device." The company uses an airline reservation scenario to illustrate the planned personalization features.

Since Microsoft is expected to deliver the first Hailstorm services itself, on an exclusive basis, using its own data repository, serious questions related to trust have arisen. Given Microsoft's track record for monopolistic business practices, placing a user's identity, and wallet, under Microsoft's guard naturally raises concerns for many.

7.1.1 Instant User Base

Hailstorm will use Microsoft's Passport user authentication system as the basic user credential. The company plans to take the 160 million Passport accounts it currently has, and as Hailstorm service become available, "provision" these accounts with these new services. Microsoft also says that sites like MSN will "take advantage of Passport very deeply."

Microsoft says that Hailstorm services are not tied to the Windows platform at all. Still, the new release of Microsoft Windows XP is described as a great "end-point" for Hailstorm services, because things like "Messenger extensions," and unified login capabilities are built in.

7.1.2 The Hailstorm Business Model

In its white paper, *Introduction to Microsoft Hailstorm*, Microsoft claims that end users will be the primary source of revenue through paid subscriptions. But some analysts argue that the real money to be made will come from the "service operators," or companies that host one or more Hailstorm compatible services. Microsoft says specific pricing for users, developers, and service operators will be announced in the future. Still, it's easy to interpret

this strategy as one more attempt by Microsoft to establish itself as the gatekeeper for the next generation of software on the Internet: Web services.

In a Summit Strategies Executive Briefing, analyst Dwight Davis says that the Web services .NET definition gets sticky because much of what Microsoft will use to deliver on its vision has roots firmly in the Windows world, giving rise to the criticism that .NET is little more than a Windows wolf in Internet clothing.¹⁰ According to Microsoft, Hailstorm beta for developers is expected Q4 this year, with services going live sometime in 2002.

8.0 What is Sun ONE?

Sun ONE is Sun's recommended software platform for delivering Services on Demand.

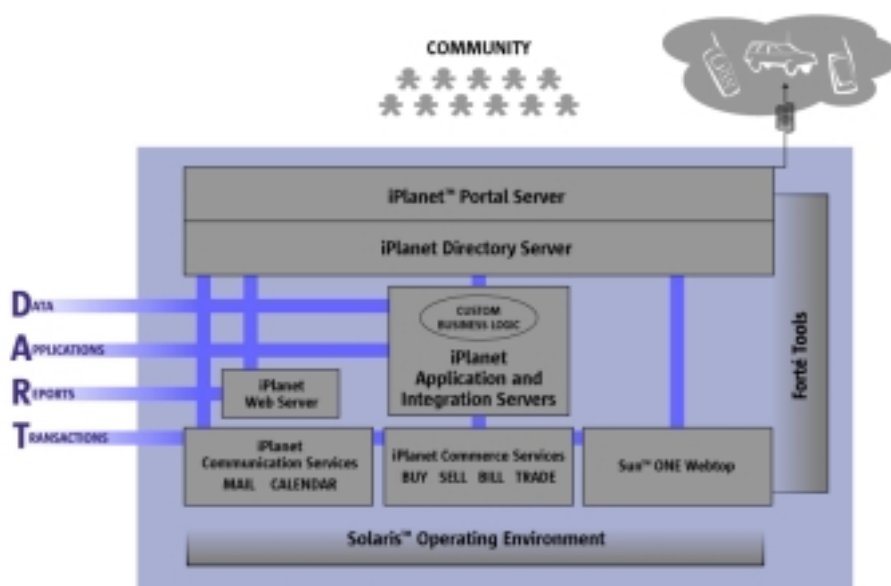
The Sun Open Net Environment (Sun™ ONE) is Sun's vision, platform, products, and expertise for building and deploying Services on Demand. Sun ONE represents a commercial software solution for building a Services on Demand infrastructure.

Services on Demand are based on open standards that can easily traverse partners, customers, and suppliers with no change in the quality of each service's capabilities. The Sun ONE product offering is an open, integratable stack designed to create and deploy Services on Demand - from the Solaris™ Operating Environment and Forte tools, through to Java™ technology as the development platform, and the entire stack of iPlanet™ products - iPlanet Application Server, iPlanet Portal Server, iPlanet Integration Server, iPlanet Directory Server, and iPlanet commerce applications.

Sun ONE expertise includes Sun Professional Services, architecting to facilitate service delivery, iForce, Sun's customer solution-driven strategy, and SunTone, providing an optimal foundation for developing Services on Demand.

¹⁰ (Summit Strategies: Assessing Microsoft .NET: Just a Windows Wolf in Internet Clothing?, March 2001)

8.1 The Sun ONE DART diagram



Where do these services come from? They are built out of information assets. Sun uses the acronym DART (Data, Applications, Reports, Transactions) to describe information assets and an architecture that can bring those assets to life in the form of services. In short, DART is a checklist to start thinking about what assets are available and could be leveraged through a software platform to meet new and existing business needs.

This architecture begins at the top by focusing on the community. Who is it that a business needs to talk to? Is it employees around the world? Partners in a local area? Business is driving something towards those communities first and foremost. So to reach them, the architecture has to include something that can cohesively and attractively present and aggregate data. That's the portal portion of the architecture. It is one of the ways to leverage data through the Internet to communities. Another way to manage that data is through a directory that can house all of a user's data, like who they are, what privileges they have, and what part of the business they need to interact with.

8.2 The Sun ONE Road Map

8.2.1 First Wave - Today

This wave helps early adopter companies start experimenting with Web services technologies while solving their immediate and ongoing platform needs. Developers can manually create and assemble basic Web services through SOAP support across various products. This includes the ability to leverage and assemble existing technologies as basic Web services through standards-based integration.

8.2.2 Second Wave - Year-end 2001

This wave helps companies start deploying their services to explore code sharing and new business models based on exchanged services. Developers

can begin deploying Web services on the J2EE platforms of the Web and Application Servers. Developers can take their assembled basic Web services and register them in a central directory for broader code availability.

8.2.3 Third Wave - 2002

This wave helps companies get a head start providing cutting-edge services based on their unique market value, while helping early majority companies understand the possibilities for capitalizing on Web services. Developers can create Web services that incorporate the ability to act on a user's context, traverse different IP-connected devices, and leverage any other Web services built on open standards.

8.3 The Role of Partnerships

Core to Sun's Services on Demand strategy is an extensive partner program that offers customers flexibility and choice in implementation. As always, Sun supports collaboration on standards and competition through implementation. Sun offers partnership programs for Application Software Providers, Independent Software Vendors, Systems Integrators and e-Integrators, Distributors, and OEMs.

9.0 Conclusions

Technically speaking, any software application has the potential to be converted to a Web service using agreed upon XML schemas and standard protocols. In fact, analyst firms have already expanded their Web services research to include deeper discussions of Dynamic Business Webs (Gartner), and Net-Native applications (Forrester). Further evidence that interest in this emerging software market is not only intense, but full of potential.

Schemas, protocols, and directories will play a critical role in cracking the code of Web services. As will having the right products to address the creation, assembly, and deployment of Web services. But before Web services can really take off, industry agreement will be required when it comes to open standards. Sun ONE is leading the way.

References:

- 1) CBDi Forum, Interact 2001, Spotlight on Hailstorm, David Sprott
- 2) CBDiForum, The Platforms are Great, But Will We "Trust" the Services?, David Sprott, April 2001
- 3) Chase H&Q, Dynamic Hosting Update: The Revolution Continues, February 2000
- 4) EAI Journal, e-Services: A Key Component for Success, Larry Wong, March 2001
- 5) EAI Journal, Disciplined XML, Don Estes, January 2001
- 6) Gartner Group, Hailstorm Clarifies .NET Vision, Will Force MS/AOL Battle, March 2001
- 7) Gartner Research, The Future of Web Services: Dynamic Business Webs, April 2001
- 8) Gartner Group, Private Exchange and Web Services: A Match That Matters, March 2001
- 9) Gartner Research, Will Web Services Standards Ever Happen?, April 2001
- 10) Giga Information Group, Web Services Will Open Possibilities for New Business Models, Randy Heffner, April 2001
- 11) Giga Information Group, Key Trends for 2001: Off-the-Shelf Components and Component Marketplaces, Randy Heffner, December 2000
- 12) Giga Information Group, Microsoft Hailstorm: Interesting Web Services Application, Business Success Uncertain, March 2001
- 13) Information Week, Plug-And-Play Redefined, Jason Levitt, April 2001
- 14) Microsoft Business, Microsoft .NET: Realizing the Next Generation Internet, www.microsoft.com/business/vision/netwhitepaper.asp
- 15) Microsoft, Hailstorm Launch Transcript, www.microsoft.com/billgates/speeches/2001/03-19hailstorm.asp
- 16) Summit Strategies, Microsoft bCentral Shifts Gears, April 2001
- 17) Summit Strategies, Assessing Microsoft .NET: Just a Windows Wolf in Internet Clothing?, March 2001
- 18) XMLFund, XMLFund Roadmap, February 2001
- 19) Zdnet, MS Hailstorm Demands Pay to Play, Joe Wilcox, March 22, 2001
- 20) Forrester Research, The X Internet, By Carl D. Howe, May 2001