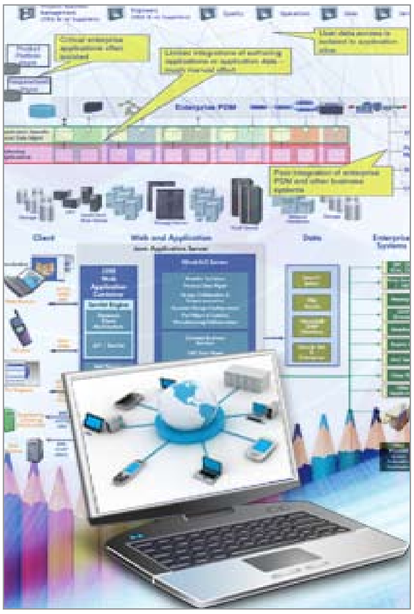
WHITE PAPER



Services Oriented Architecture in PLM

LakshmiNadh Javvadi

Delivery Manager, MphasiS PLM Practice

January, 2011

Table of Contents

Abstract

This paper describes Service Oriented Architecture (SOA), discusses the use of SOA in Product Lifecycle Management (PLM) implementations, the benefits that SOA can provide and how different PLM vendors like Siemens PLM, PTC and IBM are adopting SOA throughout their PLM product suites.

Introduction to SOA

In the field of software development, service-oriented architecture provides methods for systems development and integration. This enables to package the system functionalities and business processes into interoperable services. SOA also describes IT infrastructure which allows different applications to exchange data with one another as they participate in business processes. Service-orientation aims at a loose coupling of services with operating systems, programming languages and other technologies which underlie applications. SOA separates functions into distinct functional units or services, which developers make accessible over a network in order that users can combine and reuse them in the production of business applications. These services communicate with each other by passing data from one service to another, or by coordinating an activity between two or more services. These SOA concepts are built upon and evolving from older concepts of distributed computing and modular programming.

A service oriented architecture (SOA) is defined as a group of services that communicate with each other. An example is shown in figure1 describing a service oriented model. The process of communication involves either simple data-passing or two or more services coordinating some activity.

Intercommunication implies the need for some means of connecting services to each other. Services comprise intrinsically unassociated units of functionality that have no calls to each other embedded in them. They typically implement functionality most humans would recognize as a service, common examples such as filling out an online application for an account, viewing an online bank-statement, or placing an online booking or airline ticket order. Instead of services embedding calls to each other in their source code, they use defined protocols which describe how one or more services can talk to each other. This architecture then relies on a business process expert to link and sequence services, in a process known as orchestration, to meet a new or existing business system requirement. An application designer or engineer associates SOA objects by using orchestration. In the process of orchestration, a software engineer or process engineer associates relatively large chunks of software functionality (services) in a non-hierarchical arrangement by using a tool which contains an exhaustive list of all of the services (typically the tool is XML), their characteristics, and a means to record the designer's choices which the designer can manage and the software system can consume and use at run-time. Underlying and enabling all this, one requires metadata in sufficient detail to describe not only the characteristics of these services, but also the data that drives them. Programmers extensively use XML in SOA to create data which is wrapped in a nearly exhaustive description container. Also, the services themselves are usually described by WSDL, and communications protocols by SOAP.

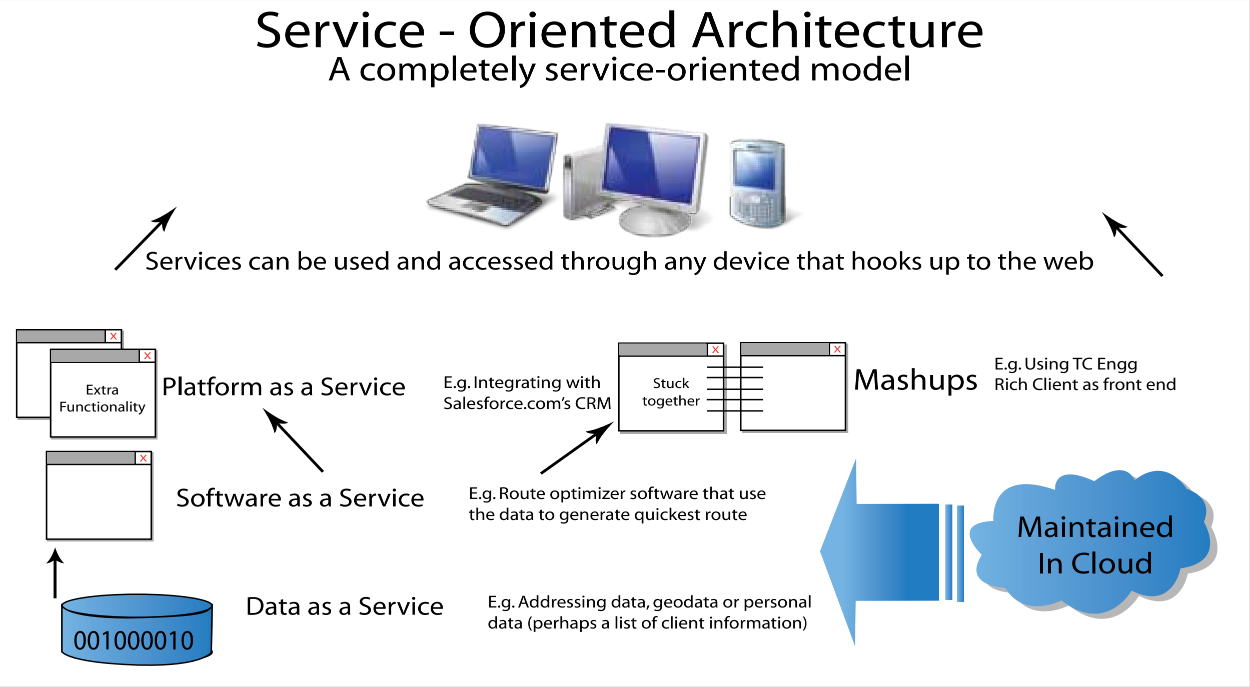

Service - Oriented Architecture
A completely service-oriented model
Services can be used and accessed through any device that hooks up to the web
Extra Functionality
Platform as a Service
E.g. Integrating with Salesforce.com's CRM
Stuck together
Mashups
E.g. Using TC Engg Rich Client as front end
Software as a Service
E.g. Route optimizer software that use the data to generate quickest route
Maintained In Cloud
001000010
Data as a Service
E.g. Addressing data, geodata or personal data (perhaps a list of client information)

Figure 1: SOA: Service Oriented Model Diagram

SOA Necessity in PLM Environment

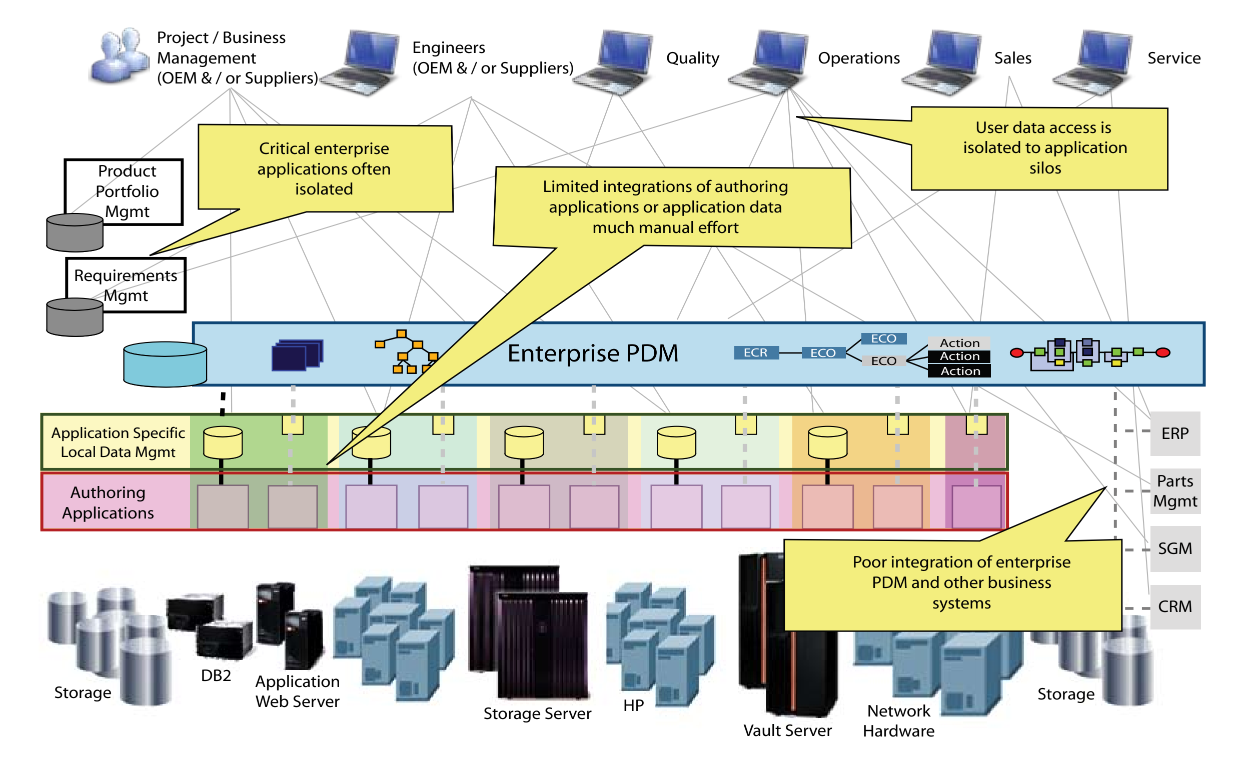

Project / Business Management (OEM & / or Suppliers)
Engineers (OEM & / or Suppliers)
Quality
Operations
Sales
Service
Product Portfolio Mgmt
Requirements Mgmt
Critical enterprise applications often isolated
Limited integrations of authoring applications or application data much manual effort
User data access is isolated to application silos
Enterprise PDM
ECR
ECO
ECO
ECO
Action
Action
Action
Application Specific Local Data Mgmt
Authoring Applications
ERP
Parts Mgmt
SGM
CRM
Poor integration of enterprise PDM and other business systems
Storage
DB2
Application Web Server
Storage Server
HP
Vault Server
Network Hardware
Storage

Figure 2: PLM Environment in the Industry (courtesy IBM; www.ibm.com/plm)

The common known problem in any manufacturing industry is that an organization consists of many departments like business management, engineering department, quality management, operations, sales etcetera using their own custom applications. Along with that a company also uses ERP, CRM, SCM and BOM Management systems etc.

The systems lack interoperability. Because of this, any department needs more time to access another department's data e.g. a manufacturing engineer may need CAD data from the Engineering Design department. Hence the delay for the companies in bringing their products into the market. Thus there is a need for an architecture which can bridge these applications and provide a single platform of application for all the departments. One such architecture is SOA.

Apart from these, SOA is also needed

- To enable enterprise-wide applications on a standardized way,
- To be able to interface different kinds of applications without hitting boundaries which are introduced when defining/ developing custom applications,
- SOA is still evolving and new technologies and tools are being proposed and released to even deliver these kind of enterprise-wide applications in a declarative environment,
- SOA can be regarded as a style of Information Systems architecture that enables the creation of applications that are built by combining loosely-coupled and interoperable services,
- Combining the power of SOA and the advantages of PLM, an industry can adopt SOA which help a product industry to reduce time to the market and reduce maintenance cost of its software,
- SOA is needed to provide new features to users, engineering managers, and IT managers as well as to remove much of the complexity in integrations of PLM environment with other business applications and processes and changing business needs.

Services Oriented Architecture Framework

In the initial phases of SOA evolution, service consumers invoke service providers directly in much the same way that desktop rich-client applications used to interact directly with back-end databases in the client-server area. Figure 3 below shows one application (service requestor) invoking a web-service offered via a second application (service provider) directly. This is a basic evolution model in SOA. In this scenario, the URL/function of the service that is being invoked is known by the service requestor. It is embedded directly in the source code of the service requestor but is often stored in some sort of configuration file or database so that it can be easily changed without re-compiling the application. Typically in software development, this kind of hard-coding should be avoided because such changes are needed to be coordinated in advance with every single service requestor that utilized the service. This problem is addressed by the registry concept which evaluated as UDDI. The UDDI approach offers a slightly improved approach. Instead of requiring all service requestors to know the URL of the services they use the UDDI registry concept. This is known as the find/bind/invoke model because the requestor first (1) finds the URL of the service it needed in the registry, (2) binds to it and (3) invokes the service.

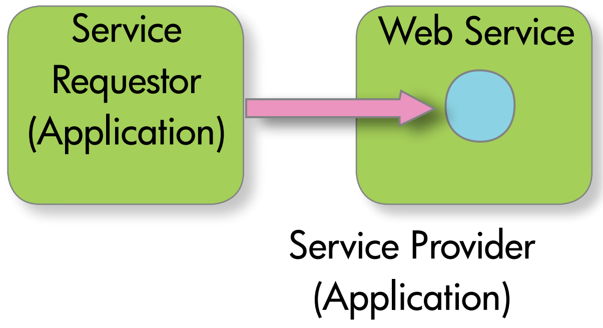

Service Requestor (Application)
Web Service
Service Provider (Application)

Figure 3: Client-Server SOA in the Initial SOA evolution phases

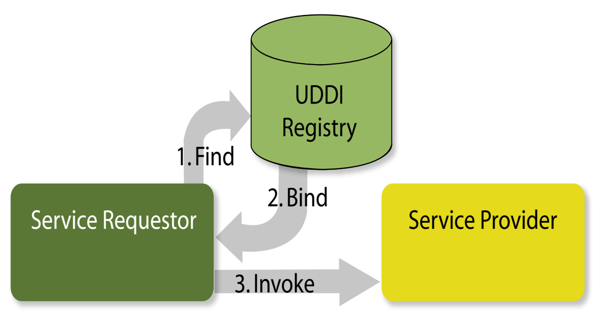


Figure 4: Client Server + UDDI

This approach is evolved gradually with the matured model of SOA, which is shown in the figure 5.

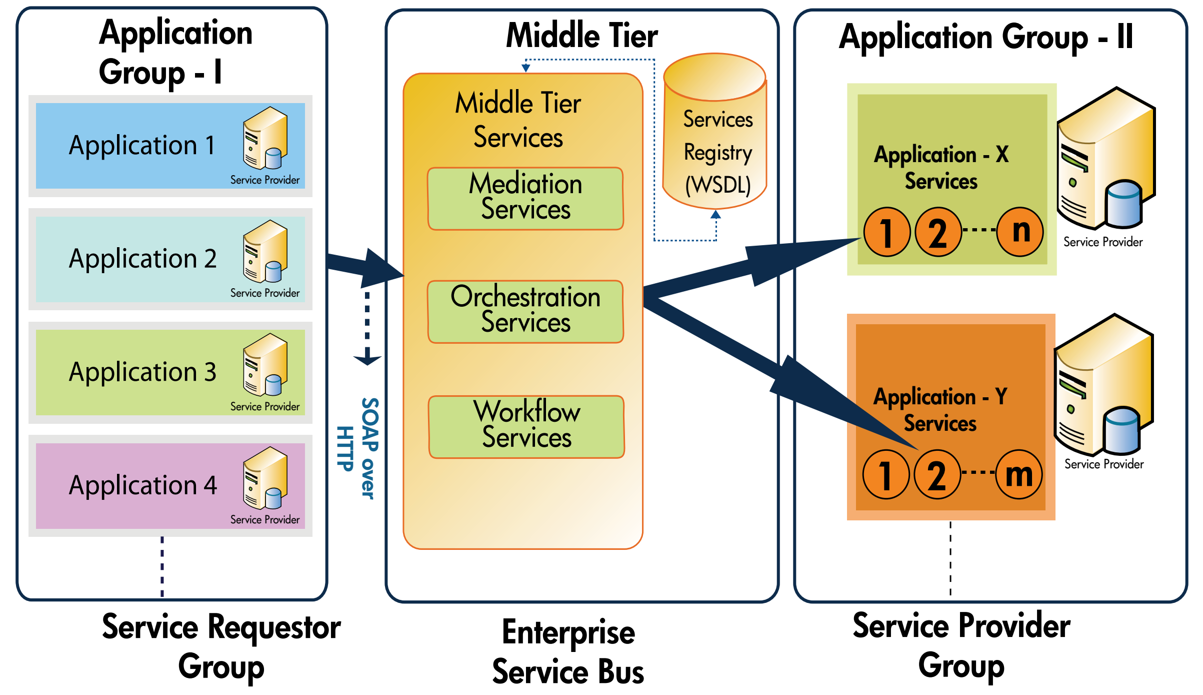


Figure 5: SOA Architecture Framework

- **Orchestration service** – A service provided by the ESB (Enterprise Service Bus) when more than one application service from one or more service providers are channeled into a single course grained web service, this is called as an orchestration service.
- **Workflow service** – Workflow services are those which are long running and they involve waiting for humans to take actions and cannot be executed as a single transaction.

Basic blocks of SOA

SOA is frequently, but not always, based on Web Service standards such as SOAP that have gained broad industry acceptance. These standards, referred to as Web Service specifications, also provide greater interoperability and some protection from lock-in to proprietary vendor software. Web Services standards relevant to SOA include:

- Extensible Markup Language (XML)—a markup language for describing data in message packages or documents in a document format,
- Hypertext Transfer Protocol (HTTP and HTTPS) —a request/ response protocol between clients and servers used to transfer or convey information,
- SOAP—a protocol for exchanging XML-based messages over a computer network, normally using HTTP.

SOAP is generally a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts:

- An envelope that defines a framework for describing what is in a message and how to process it.
- A set of encoding rules for expressing instances of application-defined data types.
- A convention for representing remote procedure calls and responses,

- WSDL—an XML-based service description that describes the public interface, protocol bindings, and message formats required to interact with a web service.
  WSDL is a structured way of communicating between web services. This addresses need, by defining an XML grammar describing network services as collections of communication endpoints capable of exchanging messages. WSDL service definitions provide documentation for distributed systems and serve as a recipe for automating the details involved in applications communication. A typical web services communication is as shown in figure 6 that describes in detail about the step by step process involved in communication using web services.
- UDDI - Universal Description, Discovery, and Integration (UDDI)—an XML-based registry to publish service descriptions (e.g., WSDL) and allow their discovery.

Design Principles for SOA Based PLM Solutions

Service-Oriented Architecture of Applications is an approach that builds applications using software services. SOA is compatible with various programming methodologies, especially object-oriented (OO) programming. Using OO programming, you can design finely-grained objects that encapsulate data and functions. These objects can then be architected into a service-oriented system, where they can be easily used by

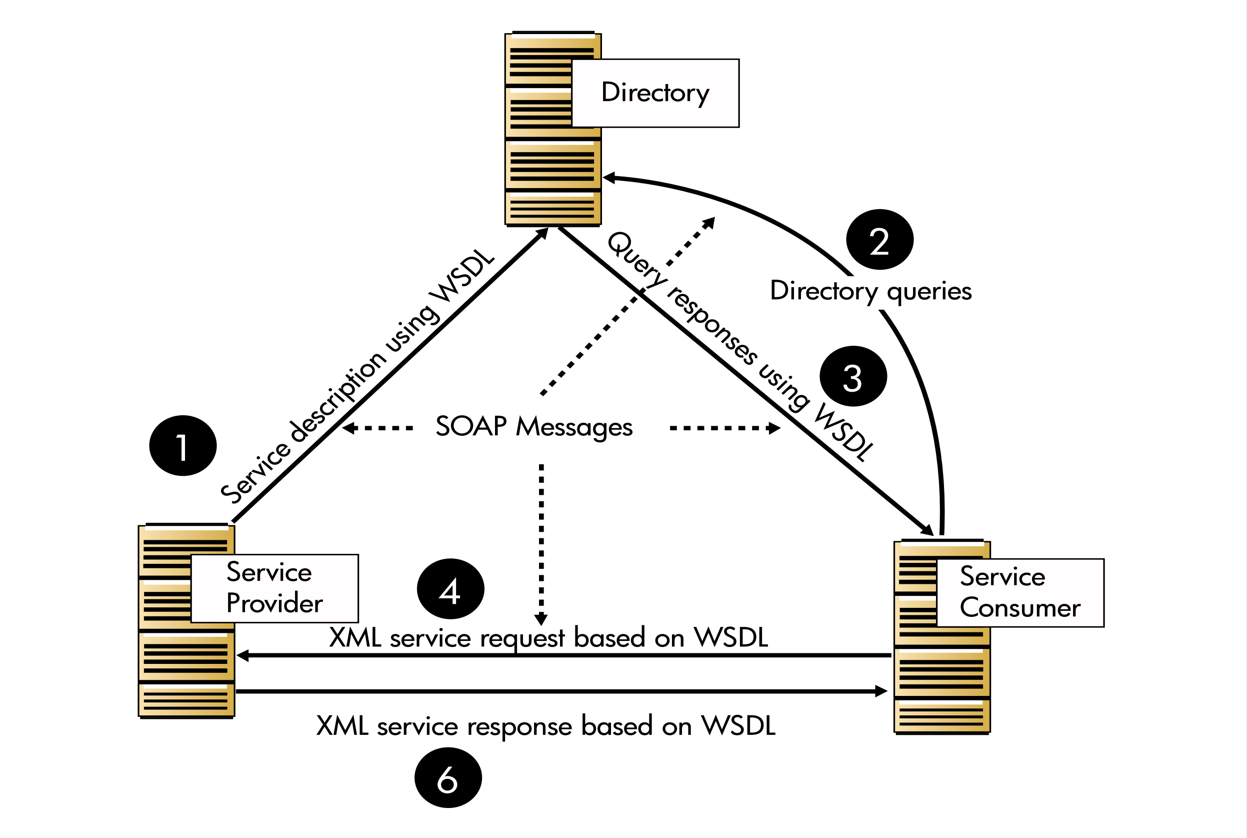

Directory
Service description using WSDL
1
Query responses using WSDL
2
Directory queries
3
SOAP Messages
Service Provider
Service Consumer
4
XML service request based on WSDL
XML service response based on WSDL
6

Figure 6: Communication using Web services

individual services, or composite applications, as needed. The successful creation of service-oriented applications requires

- Deciding what functionality makes sense to expose as a service,
- Separating and modularizing the business logic to facilitate reuse and flexibility,
- Loosely coupling services to support rapid development when requirements change,
- Designing an appropriate granularity of services,
- Planning and implementing all the SOA steps.

This can be shown in the following picture

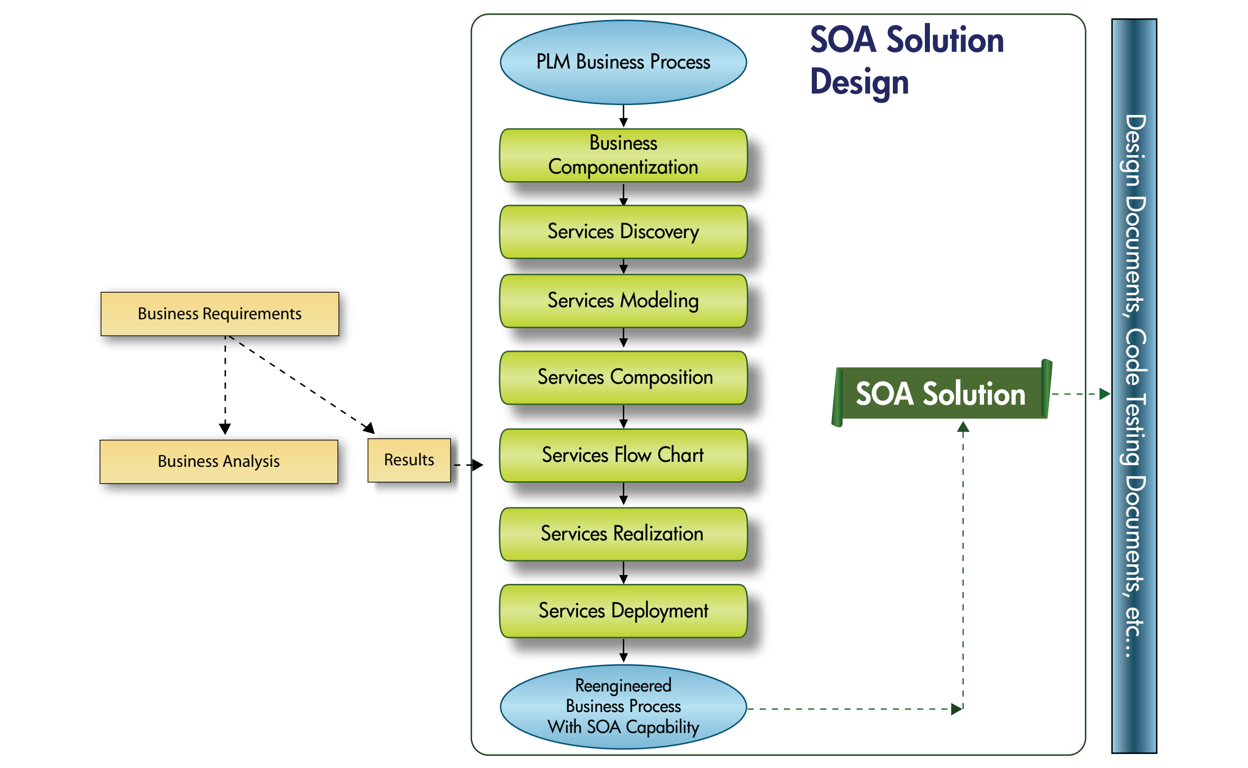


Figure 7: SOA Solution Design for PLM Environment courtesy: SOA Forum

SOA Comparison with other Architectures

SOA is different from previous client/server architectures by providing a service-based, message-oriented integration that is normally built on industry-standard technologies. The services are delivered using common components versus early client/server architectures in which each client and process interaction with a server used application-specific interfaces and protocols. Key characteristics of SOA include that it is loosely-coupled, highly reliable, business process- focused and supporting heterogeneous IT environments. The following table compares traditional client/server architecture vs. SOA.

Traditional Client/Server Architecture	Component/Module Based Architecture	Service Oriented Architecture
Object-oriented	Object Based	Services based
Method or function driven	Messages need to be Stateful	Message oriented
Tightly coupled	Components cannot be reused as they are tightly coupled with the application.	Loosely coupled
Application silos		Interoperating solutions
Application-specific interfaces and protocols for client and server interaction		Common interface definitions and protocols

Table 1: Comparison with Legacy Architectures

SOA in PLM Products

The use of SOA by developers of PDM, ERP and other business applications is enabling the implementation of PLM environments, allowing them to present new features to users, engineering managers, and IT managers as well as remove much of the complexity in integrating the PLM environment with other business applications and processes. Because of the previously described services. Teamcenter's SOA services are accessible through a set of language-specific libraries that make it easy for programmer's with C, C++, C# (.NET), and Java experience to easily incorporate Teamcenter functionality into new and existing applications. A Web Services Interoperability (WS-I) compliant WSDL helps integrate with almost any legacy technology. The TC UA is as shown below in figure 8 which uses SOA.

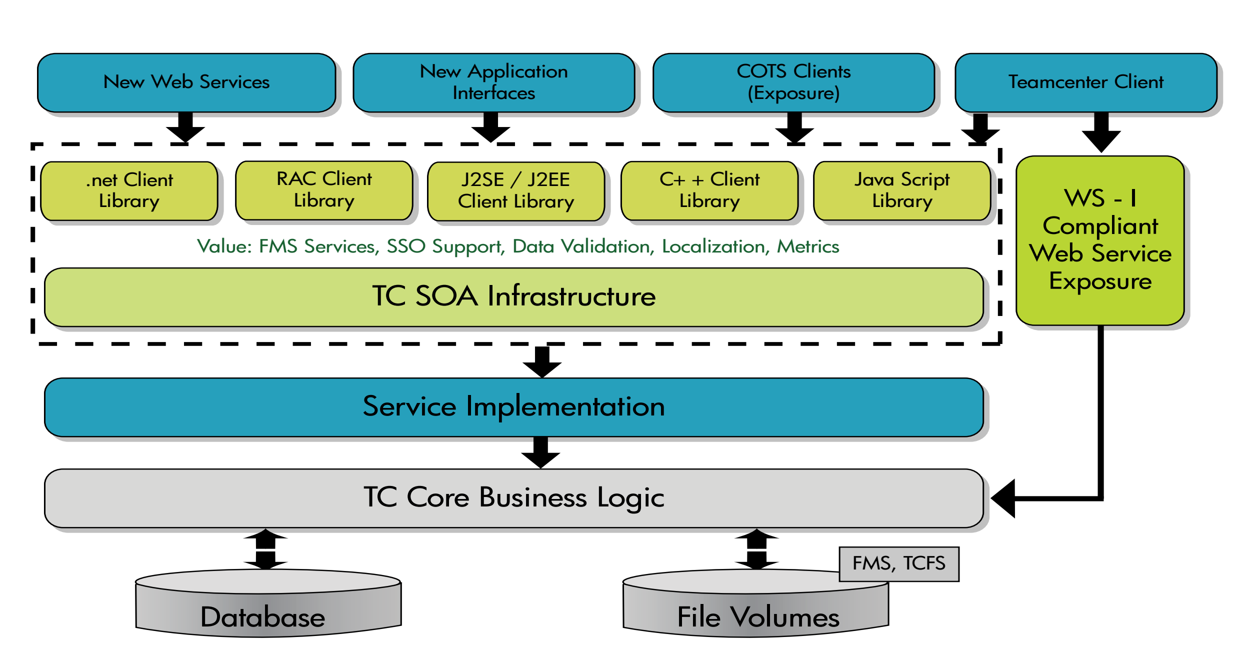


Figure 8: Teamcenter Unified Architecture

strengths of SOA, many of the major PLM solution providers are converting to an SOA for development and delivery of the product suites. A high-level status of the use of SOA by a number of the leading PLM solution suppliers is presented below.

- Agile (now part of Oracle) has been incorporating services and components of a service-oriented architecture within its product suite for several years. Agile's Integration Server is built on a proprietary services foundation but has since been converted to use industry-standard services protocols with the advantage of SOA, this approach enables to use services to create a more flexible architecture upon which to develop and deliver its products and as a platform to deliver business-oriented services. Agile supports J2EE, XML, and Web services standards and their Supply Chain Security facility is built on a services paradigm.
- Siemens PLM implemented SOA in Teamcenter unified. SOA has replaced CORBA and TC Services in the application. Apart from that Teamcenter has provided SOA extensions and interfaces for third-party software integration. This allows a pluggable software development approach. SOA brings technology and capabilities to Teamcenter that enable deployment of high performance, scaleable, WAN friendly reliable
- PTC created the Windchill PLM integration solution based on Service Oriented Architecture (SOA) standards which has a flexible integration strategy that allows them to adopt a more flexible "design anywhere, build anywhere" development strategy. As a production-proven PLM system, Windchill is essential for companies who need to collaborate with suppliers, or for suppliers who must react to demands from OEMs to take on responsibility for the design of key components in a complex product. Windchill manages all product content and business processes throughout the product lifecycle. Controlling everything from MCAD and ECAD data to software, calculations, illustrations and technical publications, Windchill enables all users – across the global enterprise and supply chain – to work with a single representation of the entire product. As the single source of product related information, Windchill has the control and configuration capabilities needed for users to collaborate across the extended enterprise. Windchill also enables users to optimize key product development processes and help them achieve specific business initiatives. A typical Windchill system is as shown in figure 9 below.

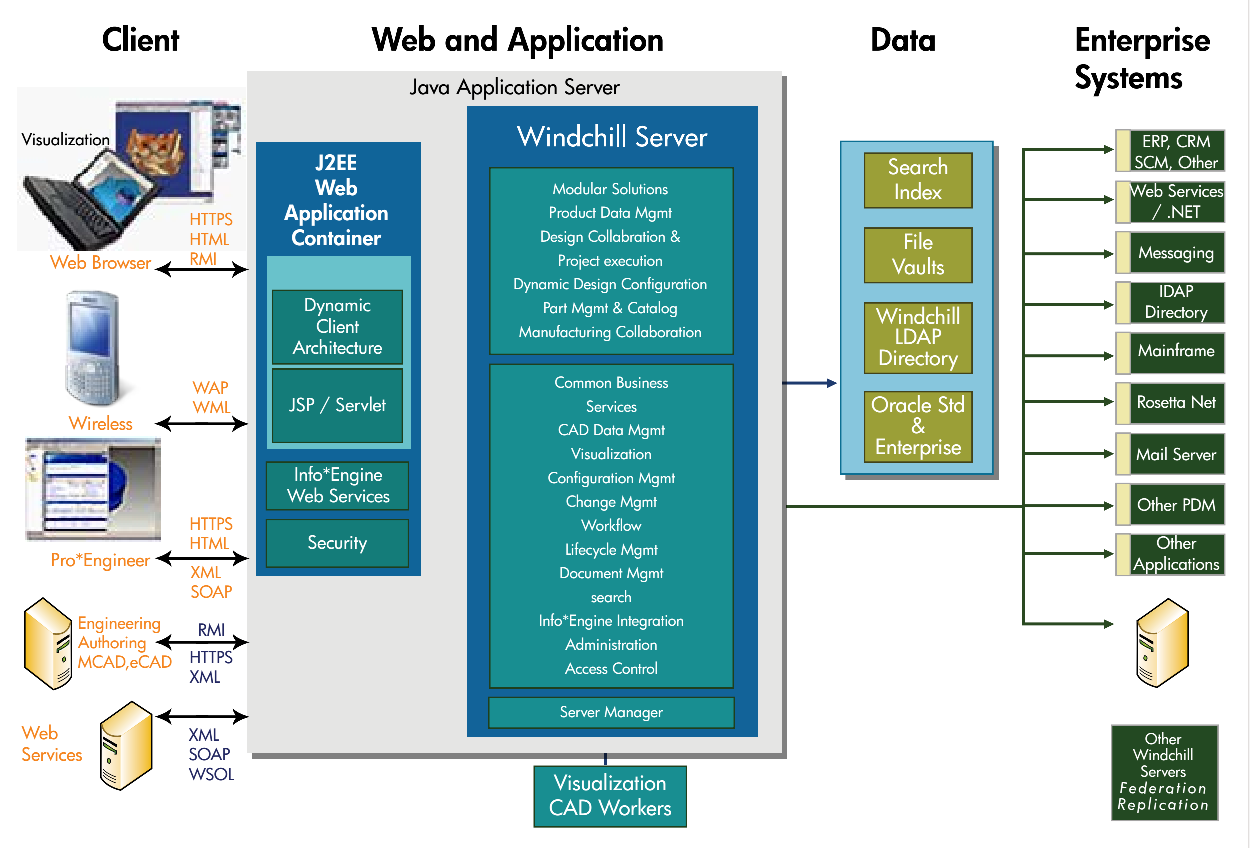


Figure 9: Windchill Architecture

A Case Study

A shipyard company that is existing since six generations; the company specializes in cruise liners, luxury cars and passenger ferries, roll-on, roll-off and passenger ships, gas tankers and livestock carriers. It employs approximately 2,500 people. The company has a production capacity of 2.5 cruise liners per year. Sharp competition is driving time and cost pressures to achieve order to delivery in less than 30 months. A closely interlocked process between the company and its suppliers is necessary, as 75 percent of the company's material and components are sourced externally. The company's existing product life cycle management (PLM) process and system landscape was historically grown and lacked flexibility and scalability to meet current and future business demands. The applications were tightly coupled, which resulted in an inflexible infrastructure. The company's applications were data-focused, less process oriented, so that the end users often needed to access various applications to complete one business process. Data is duplicated without a mechanism for consistency. This situation did not provide a solid base for either engineering or business decisions. Further, a lot of the company's application interfaces were based on an error-prone file transfer mechanism.Change management between manufacturing and engineering departments is the most critical task. The automated lines are very sensitive to changes, which makes engineering change management a specific challenge. To make proper decisions, the engineering and production departments needed consistent and integrated design information and production data.

The possible solution would be a SOA-based infrastructure which has loosely coupled Web Services to access the client's legacy applications and PLM system. In addition to that a common logical data model to enable a flexible data exchange between clusters in terms of mapping, routing and process adoption, can lead to lower maintenance costs. Figure 10 shows the SOA Based architecture solution proposed.

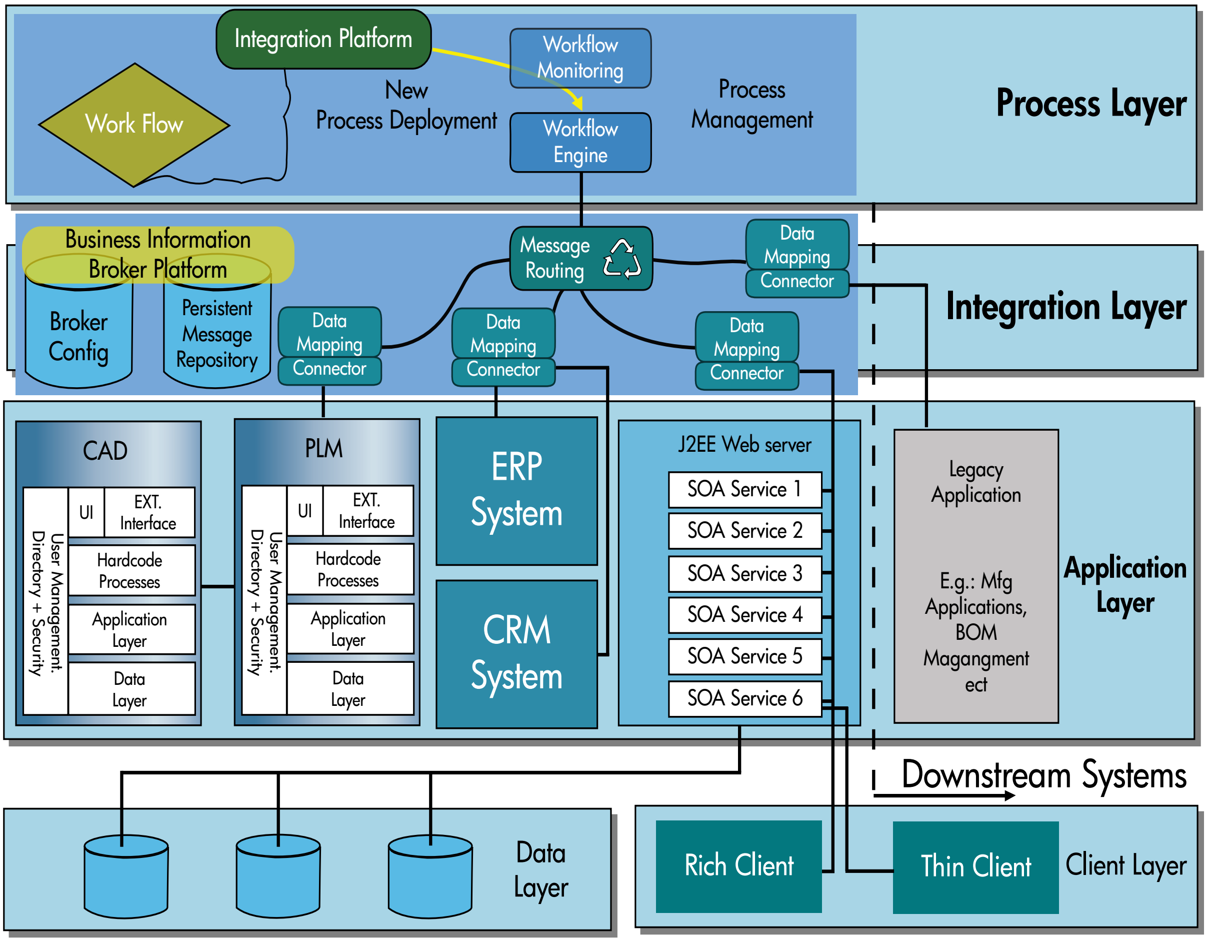


Figure 10: SOA Based Solution

SOA Benefits to PLM

PLM benefits extend beyond engineering to the entire value chain delivering more than software applications and technology. The infrastructure is able to adapt quickly to new business models and processes. PLM needs to be integrated with the rest of the enterprise to become a source of product information for all stakeholders involved in product development decisions. PLM systems enabled by a Service Oriented Architecture (SOA) infrastructure can address these needs, treating information and processes as services. SOA industry extensions for PLM provides an IT foundation on which virtually all enterprise applications can operate synchronously—sharing and reusing processes and information, and exposing results in a consistent manner across the enterprise.

SOA provides the glue that integrates product development with other business operations. It helps break down the silos that separate engineering from other organizations by providing a framework for heterogeneous systems and processes to operate in a homogeneous manner. The PLM information and processes are thus exposed and available at the executive level, where the impact of product development decisions on the entire business can be evaluated.

SOA Disadvantages

While the SOA has many advantages in the PLM Industry, this kind of architecture is not suitable in stand alone, non distributed applications that do not necessitate application or component integration e.g.: a word processing application that does not require request and response based calls. Also SOA is not suitable in homogenous application environments such as an environment wherein all applications were built utilizing J2EE components. In these instances, it is not a good idea to introduce XML over HTTP for inter-component communications rather than utilizing Java remote method invocation.

Conclusion

Most of the manufacturing industries can adopt SOA based PLM solutions that provide the ability to expand the level of functionality available, improve the user's experience even as more and more diverse users work within the PLM environments, and reduce the cost and complexity of deploying and maintaining a distributed PLM environment.

Service oriented architectures offer long-term benefits to both solution developers and to those companies that adopt SOA within their IT infrastructure and environments – faster development of applications and industry/user/company tailored solutions, faster and less costly deployment, and less dependency on a given set of technology. Many PLM Product vendors have been incorporating SOA capabilities within their product suite for the past several years, expanding the use of SOA with each release. For example with the release of TeamCenter 2007, Siemens PLM has started the journey. This updated architecture will enable UGS to rapidly expand their product suite to meet the needs of their users and help those users more quickly upgrade their PLM environment to take full advantage of UGS offerings.

Contact us

USA

MphasiS
460 Park Avenue South
Suite # 1101, New York
NY 10016, U.S.A.
Tel: +1 212 686 6655
Fax: +1 212 686 2422

UK

MphasiS
88 Wood Street
London EC2V 7RS, UK
Tel: +44 208 528 1000
Fax: +44 208 528 1001

AUSTRALIA

MphasiS
410 Concord Road
Rhodes, NSW 2138, Australia
Tel: +61 290 221 146
Fax: +61 290 221 134

INDIA

MphasiS
Bagmane Technology Park
Byrasandra
C.V. Raman Nagar
Bangalore 560 093, India
Tel: +91 80 4042 6000
Fax: +91 80 2534 6760

About MphasiS

MphasiS is a global service provider with $1B in revenues, delivering technology based solutions to clients across the world. With currently over 39,000 people, MphasiS services over 200+ clients in Banking and Capital Markets, Insurance, Manufacturing, Communications, Media & Entertainment, Healthcare & Life Sciences, Transportation & Logistics, Retail & Consumer Packaged Goods, Energy & Utilities, and Governments around the world. Our competency lies in our ability to offer integrated service offerings in Applications, Infrastructure Services & Business Process Outsourcing capabilities. We are uniquely positioned to service our clients with best cost-performance. To know more about MphasiS, log on to www.mphasis.com

MPHASIS
an HP company