



Design Pattern Technical Details

The detailed technical objectives for the Seros solution strategy are presented below. These objectives provide an alternative to “SOA Monoliths,” and they embrace open standards, open source products, vendors that perform specific (but not monolithic) functionality, metadata strategies based on business domain vocabularies, and building partnerships.

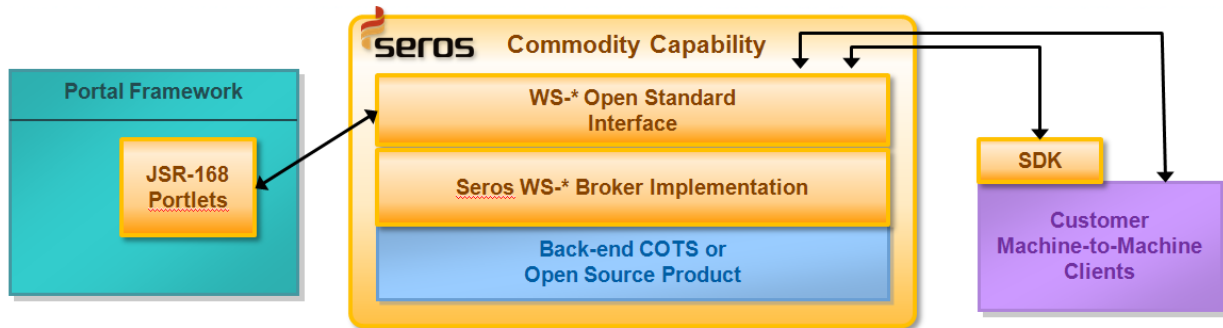
SEROS Technical Solution Objectives

1. **Provide an open standards, web-services based implementation for SOA Core Enterprise Services**
 1. Provide SOA Core Enterprise Services implemented as commodities
 - Treat a heterogeneous SOA environment as a commodity pattern to be achieved versus a specific vendor product solution
 - Focus on interface standardization using the WS-* open standards instead of the elusive homogeneous solution, i.e., a single vendor stack
 - Utilize the WS-* open standards based interfaces to create a reference architecture foundation that also achieves interoperability and federation
 - Create a set of SOA Core Enterprise Services that is transparent to the consumer for a specific vendor or open source product perspective
 - Use governance and policy enforcement to achieve homogenous life cycle management
 2. Provide a high degree of flexibility for organizations to define, implement, and govern their SOA life cycle management
 3. Provide a loosely coupled set of core services with a high degree of interoperability, composability, and flexibility in Vendor/Open Source product choices
 - Based on a repeatable design pattern
 - Prevent vendor and/or specific technology “lock-in”
 - Provide flexibility in making architectural choices and decisions
 4. Provide both machine-to-machine (via fully described web services) and human user interfaces (via JSR-168 portlets)
 - Provide detailed definitions of the web-service interfaces to include the WSDL files, XML schemas, developer documentation, software development kits (SDK), and reference implementations available on the network
 - Provide a reference implementation of a SOA Dashboard for operations management, service administration, and end-user access to services
2. **Provide transparent precision search capabilities tailored to existing business processes and tools**
 1. Automated publishing based on the business vocabulary
 2. Normalization of common search terms within vocabularies
 3. Desktop search metaphors that work within existing tools to exploit the business vocabulary based metadata
 4. Recurring search patterns with stylized search results
 5. Search result relevancy algorithms that are vocabulary based
 6. Transparent implementation of information governance/management policies

3. Provide support for recurring patterns of service composition/orchestration that can be parameterized and utilize the power of precision search for services and data sources
4. Provide highly scalable solutions that can be federated across large distributed environment

Seros has developed a design pattern based approach to realize our technical objectives and implement a set of Product Suites. Our design pattern is illustrated in the figure below.

Design Approach and Guiding Principles



The Seros Commodity Capability Design Pattern composes 4 components

1. A WS-* interface based on SOAP web services
 - Multiple WS-* open standards can be composed together to create a commodity capability
 - For example, WS-Notification can be composed with Ws-Security and WS-Reliable Messaging to create a secure, reliable messaging interface
2. A WS-* broker implementation that connects the WS-* web service interface to one (or more) back-end Commercial-off-the-Shelf (COTS) or Open Source products
3. A set of JSR-168 portlets that create a human user interface for the WS-* web services. For example, a SOA Dashboard
4. A Software Developer's Kit (SDK) that allows software engineers to easily create client/application software
 - The SDK provides an abstraction of the WS-* interface for ease of programming (indicated by the dashed line)
 - Programmers can also directly program to the appropriate WS-* web services and operations

Packaging options include: Virtual Machine Instance, Software Installation, and Network Appliance

The standards-based, public web service definitions provide the power to the Seros products. These fully described and documented web services provide the commoditized, web service based interface definitions required to implement a robust set of enterprise core (or business domain) services while simultaneously providing vendor neutrality/independence. The web services provide functionality for capabilities such as: service registration, service discovery, service governance, service monitoring, service reporting, messaging, enterprise service bus, data publishing and transformation, orchestration, and many others.

The power of the Seros web services is realized via the resulting composability and interoperability of the commoditized web services. For example, it is now an easy matter to compose service registration with service monitoring provisioning; to compose messaging with enterprise service reporting; to compose messaging with transformation, adaptation, and automated publishing; to compose precision searching with authoritative data sources; to compose security services with both SOA infrastructure and business services; to compose SOA infrastructure services with existing applications and products; etc. Lastly, the Seros public web services provide the vendor neutrality/independence layer from the underlying vendor or open source products being used to provide specific functionality.

The Seros design pattern can integrate a wide variety of vendor/open source products. These products can be individually selected based on factors such as: an organization has already purchased these products, an organization desires to utilize what is considered the best-of-breed products, and/or an organization desires to utilize what is considered to be the best value product for satisfying their needs. Additionally, the Seros design

pattern allows for the incorporation of additional value-added features and functions that can significantly enhance each product's out-of-the-box capabilities. Simultaneously, Seros' commitment to open standards and WSDL-based service specifications affords Seros the unique opportunity to adopt new or substitute products as needed in order to meet evolving requirements.

The Seros public web services also provide a standard way of implementing human user access. Using open standards like JSR-168, a rich set of portlets are provided that present an intuitive human user interface. Additionally, the portlets access the same public web services. Therefore, both machine-to-machine and human user access is standardized and guaranteed to remain consistent. Because the portlets are implemented in compliance with the JSR-168 open standard, these portlets will easily compose into many of the major portal framework products being used today.

The net effect of the Seros Design Pattern is that it allows a SOA reference architecture to be specified via a fully described set of web services. Thus, a high degree of interoperability and composability is achieved between the web services. Additionally, the commoditized nature of the design pattern based web services provides a high degree of vendor neutrality/independence.

To learn more about Seros, please visit us on the Web: www.seros.com.

For more information, please contact:

Rick Rosenberg, President and CEO
950 N. Glebe Rd.
Suite 1100
Arlington, VA 22203
703.841.5977
FAX 703.908.9353
info@seros.com