



SOA REFERENCE ARCHITECTURE: SOA FRAMEWORKS

SOA Blueprint

A structured blog by Yogish Pai

SOA Frameworks


SOA frameworks are reusable services that are the backbone of the SOA. These re-usable services must be enterprise-class, designed well enough to scale under load and meet the demands of a diverse set of consuming applications and stakeholders.

SOA frameworks support the move to an SOA by helping development teams rapidly design, develop and deploy well-designed, modular, flexible, scalable, and supportable web services, web applications, and portlets. As companies start adopting SOA principles to transform their IT architectures, these underlying services must be created to perform as enterprise-class assets.

A framework can be defined as a reusable skeleton application that teams extend in order to build specific services or applications. Frameworks improve consistency in the delivered software. Frameworks control themselves and the application or services that are created on top of them.

Frameworks typically provide a set of high-level programming abstractions and a strong starting point for creating enterprise-class services. They often specify a layered architecture for services that incorporates several design patterns and software engineering best practices. The architecture also specifies the responsibilities of the components in each of the layers and the collaboration between them.

Services inherit the good architecture and best practices built into the frameworks. Using frameworks, a team of average developers can develop well architected services that take advantage of design patterns and best practices. The typical layers that a services creation framework would offer include:

- **Transformation layer:** supports protocol and data-type conversions to support multiple access protocols, while at the same time keeping most, if not all, of the service implementation protocol and access mechanism agnostic.
 - **Business logic layer:** holds all the business logic in the system. This includes such abstractions as request, result, UseCaseController, and BusinessPolicy objects.
 - **Business data layer:** the layer for domain objects, which are the objects that have a consistent definition across many applications in the enterprise. The business data layer should provide location transparency so that users of the domain objects don't need to know the exact physical location of the underlying persistent data. This layer should be able to manage persistence to and retrieval from various persistence repositories in the enterprise.
 - **Integration layer:** a placeholder for connection technology implementations ranging from JDBC to JNI to Java connectors. All the infrastructure code that is needed to access extended enterprise systems—such as ERP systems and content repositories—fits into this layer.
- 



SOA frameworks benefit both developers and the corporation. When developers use frameworks, they:

- Gain a solid foundation to create services, web applications and portlets
- Improve productivity by incorporating design patterns and best practices
- Utilize off-the-shelf features of the frameworks and write less code
- Don't need to understand the nuts and bolts of J2EE standards and specifications
- Don't need to be an expert at object-oriented design and design patterns to benefit from using them.

For IT organizations and the company as a whole, the SOA frameworks deliver:

- A catalyst for getting to a SOA quickly and at a lower cost
- Consistency of design and development across projects
- Repeatability and the ability to guarantee a minimal level of architecture and design rigor.
- Improved business agility as a result of having modular solutions that can be changed easily, often via configuration changes
- Leverage of software engineering best practices among developers with varying skill levels
- More consistent, predictable, and better tested solutions
- Improved mobility of developers among projects.

IT organizations are using SOA principles to aggressively create re-usable services that encapsulate and expose key business processes. By combining a layered architecture, ease of use, and a deep emphasis on good architecture and re-use, SOA frameworks companies to create enterprise-class mission-critical services that are vendor-neutral and portable.

Copyright

Copyright © 2007.

The author grants a non-exclusive licence to everyone to publish this document in full or part by acknowledging the source. Any other usage is prohibited without the express permission of the authors.

