Subject Areas Get Involved

About Us

The SOA **Source Book**

SOA Source Book

Architecture

Service-Oriented

The Open Group Service **Integration Maturity** Model (OSIMM) Version 2

- Introduction The Model **Business Dimension:**
- Base Model Organization & Governance Dimension: Base Model
- Method Dimension: Base Model **Application Dimension:** Base Model
- Architecture Dimension: Base Model Information Dimension:
- Infrastructure & Management Dimension: Base Model

Base Model

- The OSIMM Assessment Method
- **Example Assessment** Benefits of Moving to
- Higher Maturity Levels Relationship to Other SOA Standards
- Relationship to Other International Standards Referenced Documents

Acknowledgements

Legal **SOA Reference**

Architecture

Service-Oriented Cloud Computing Infrastructure (SOCCI) Framework

Using TOGAF to Define and Govern Service-**Oriented Architectures**

Framework Service-Oriented

SOA Governance

Architecture Ontology Version 2.0

Legacy Evolution to SOA Microservices

Architecture

SOA for Business Technology

Navigating the SOA **Open Standards** Landscape Around **Architecture**

SOA in the Digital Age

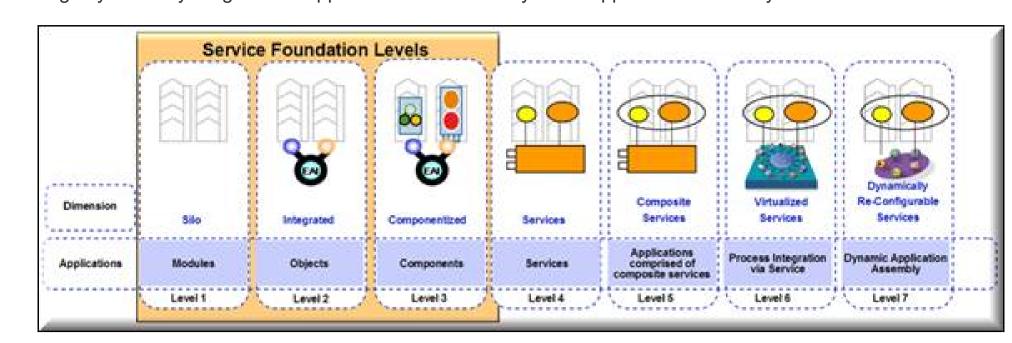
SOA Work Group SOA Standards Discussion Forum SOA Work Group Members Page

The Open Group Service Integration Maturity Model (OSIMM) Version 2 – Application Dimension: Base Model

Events

This chapter defines the base model for the OSIMM Application dimension base model. The base model defines a set of generic maturity indicators and attributes that can be used to assess an organization's SOA maturity level against the OSIMM maturity matrix. Additional maturity indicators, assessment questions, and attribute mappings can be added by vendors or user organizations to extend the base OSIMM model.

The assessment questions that follow help elicit the level of formality to which an organization has successfully applied SOA application and system design, development, and deployment principles, and adopted SOA-enabling technologies such as an ESB and service registry. Maturity ranges from application modules to dynamic application assembly.



OSIMM Application Dimension

Application Dimension: Base Model Maturity Indicator

The base OSIMM model provides one of many possible maturity indicators per dimension. Organizations, vendors, and consultants can provide additional maturity indicators, assessment questions, and attribute mappings to provide additional guidance necessary for the maturation of an organization's SOA.

- The following Application dimension maturity indicator is provided as part of the base OSIMM specification:
 - An SOA maturity assessment of the OSIMM Application dimension can be conducted by identifying the application architectures that are designed and implemented using SOA principles and development practices and utilize constructs such as loose-coupling, separation of concerns, and employ the use of service-enabled technologies such as XML, web services, service bus, service registries, and virtualization.

Application Dimension: Assessment Questions

By gathering information using these assessment questions, an assessor can map a maturity indicator to the associated maturity attributes, thereby determining the Application dimension maturity level.

- 1. What is your current application development style?
- 2. How common is re-use in your organization?
- 3. What types of re-use do you engage in and how is re-usability measured?
- 4. How are your organization's applications/systems integrated?
- 5. What types of languages does your organization use?
- 6. What types of integration technologies has your organization employed?
- 7. How is business logic represented within your organization's applications?
- 8. How reliable are your organization's business-critical applications?
- 9. How widely is XML used in your organization? How sophisticated is its use?
- 10. What is the rate of change and required time-to-market of your current applications?
- 11. Are SOA-enabling technologies, such as ESB, shared data environment, or registry, being used?

Application Dimension: Maturity Indicator-to-Attribute Mapping

The following are the base set of maturity indicators for the OSIMM Application dimension. Each maturity indicator is associated with a set of maturity attributes. Maturity attributes are those observed characteristics of a maturity indicator for each maturity level. The assessment questions are used to survey an organization's application or system architectures. Survey data obtained through the Application dimension assessment questions is used to determine the maturity level by assessing the data and matching to the maturity attributes that best fit the information obtained. The maturity weighting is used to determine an average maturity score across multiple maturity indicators. The model can be extended by adding additional maturity indicators and assigning weighting to the indicator by maturity level according to the value placed on the maturity indicator by the assessing organization.

Maturity Indicators for the Application Dimension

Maturity Level Cell Name	Maturity Indicator	Maturity Attributes	Maturity Weighting	Assessment Question Mapping
Silo (Level 1) <i>Modules</i>	Application architectures are designed and implemented using SOA principles and development practices that utilize constructs such as loose-coupling, separation of concerns, and employ the use of service-enabled technologies such as XML, web services, service bus, service registries, and virtualization.	Low or nonexistent Application architectures and topologies are monolithic and lack integration between other systems across the enterprise. The use of web services or other SOA constructs are not present.	10	1, 4, 7, 10
		, ,		6, 9
ntegrated (Level 2) Objects	Application architectures are designed and implemented using SOA principles and development practices that utilize constructs such as loose-coupling, separation of concerns, and employ the use of service-enabled technologies such as XML, web services, service bus, service registries, and virtualization.	Limited Application architectures and topologies are monolithic with minimal separation of concerns between architectural layers or application tiers. Applications use minimal integration between other systems. Integration is usually implemented using point-topoint techniques.	20	1, 2, 3, 7
				4, 6
Componentized (Level 3) Components	Application architectures are designed and implemented using SOA principles and development practices that utilize constructs such as loose-coupling, separation of concerns, and employ the use of service-enabled technologies such as XML, web services, service bus, service registries, and virtualization.	Cross-organizational SOA development practices are applied inconsistently across the organization.	30	1, 2, 3, 4
		Most application architecture topologies have a separation of concerns both physically and logically in presentation, business logic, and data tiers. The use of SOA-enabling technologies – such as an ESB – is inconsistent across the enterprise.		5, 7, 10
				6, 9, 11
Services (Level 4) Services	Application architectures are designed and implemented using SOA principles and development practices that utilize constructs such as loose-coupling, separation of concerns, and employ the use of service-enabled technologies such as XML, web services, service bus, service registries, and virtualization.	Enterprise-wide Service components of application architectures employ SOA patterns such as separation of concerns between logical and physical layers of the presentation and business logic. Service integration is achieved using an ESB in some but not all business	40	1, 2, 3, 4
		units.		5, 6
Composite	Application architectures are designed and implemented using SOA	Integrated Enterprise-wide	50	
Services (Level 5) Applications Composed of Composite Services	principles and development practices that utilize constructs such as loose-coupling, separation of concerns, and employ the use of service-enabled technologies such as XML, web services, service bus, service registries, and virtualization.	Application architectures are designed with a separation of concerns at the logical and physical layers. ESB integration patterns are used to support application and process integration to achieve sharing of		1, 2, 3, 7
		services.		4, 6, 11
Virtualized Services (Level 6) Virtualized Services	Application architectures are designed and implemented using SOA principles and development practices that utilize constructs such as loose-coupling, separation of concerns, and employ the use of service-enabled technologies such as XML, web services, service bus, service registries, and virtualization.	Integrated across the enterprise and externally between business partners. Application architecture is decoupled from infrastructure components.	60	
		Extensive use of ESB architecture patterns to support Business Process Management (BPM).		1, 2, 3, 10
				6, 7, 8, 11
Oynamically Re- Configurable Services Level 7) Oynamic Application Assembly, Context-aware	Application architectures are designed and implemented using SOA principles and development practices that utilize constructs such as loose-coupling, separation of concerns, and employ the use of service-enabled technologies such as XML, web services, service bus, service registries, and virtualization.	Adaptive Enterprise Application architecture supports dynamically reconfigurable business and infrastructure services and SOA solution for internal or external partner consumption.	70	All

⇒ Architecture Dimension: Base Model















© 1995-2016 The Open Group