BS ISO/IEC 18384-2:2016



BSI Standards Publication

Information technology — Reference Architecture for Service Oriented Architecture (SOA RA)

Part 2: Reference Architecture for SOA Solutions

bsi.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 18384-2:2016

National foreword

This British Standard is the UK implementation of ISO/IEC 18384-2:2016.

The UK participation in its preparation was entrusted to Technical Committee IST/38, Cloud Computing and Distributed Platforms.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2016.
Published by BSI Standards Limited 2016

ISBN 978 0 580 81294 1

ICS 35.100.05

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 July 2016.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 18384-2:2016

BS ISO/IEC 18384-2:2016

INTERNATIONAL STANDARD

ISO/IEC 18384-2

First edition
2016-07-01

Information technology — Reference Architecture for Service Oriented Architecture (SOA RA) —

Part 2: Reference Architecture for SOA Solutions

Technologie de l'information — Architecture de référence pour l'architecture orientée service (SOA RA) —

Partie 2: Architecture de référence pour les solutions de l'architecture orientée service

Reference number
ISO/IEC 18384-2:2016(E)

Documents
Document Sharing Hub
© ISO/IEC 2016

Full document details: BS ISO/IEC 18384-2:2016



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2016, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Documents
Document Sharing Hub

© ISO/IEC 2016 – All rights reserved

Full document details: BS ISO/IEC 18384-2:2016

Contents

Page

Foreword ... **viii**
Introduction ... **ix**
1 Scope ... **1**
2 Normative references ... **1**
3 Terms, definitions and abbreviated terms ... **1**
3.1 Terms and definitions ... 1
3.2 Abbreviated terms ... 1
4 Notations ... **2**
5 Conventions ... **3**
6 Conformance ... **5**
7 Overview ... **5**
7.1 Introduction to SOA ... 5
7.2 Introduction to the SOA Reference Architecture ... 6
7.3 Metamodel ... 7
7.4 Capabilities ... 11
7.5 Reference Architecture for SOA Solutions ... 12
7.5.1 Overview of Reference Architecture ... 12
7.5.2 Operational and IT Systems Layer ... 14
7.5.3 Service Component Layer ... 14
7.5.4 Services Layer ... 15
7.5.5 Process Layer ... 16
7.5.6 Consumer Layer ... 16
7.5.7 Integration Aspect ... 17
7.5.8 Management and Security Aspect ... 18
7.5.9 Information Aspect ... 19
7.5.10 Governance Aspect ... 20
7.5.11 Development Aspect ... 20
7.6 Common Services Categories ... 21
7.7 Assumptions and Key Concepts ... 23
7.7.1 General ... 23
7.7.2 Functional and Non-functional ... 23
7.7.3 Requirements ... 23
7.7.4 Services ... 23
7.7.5 Documenting the Layers ... 24
7.7.6 Logical and Physical Elements ... 24
7.7.7 Interactions between Layers ... 25
7.7.8 Understanding ABBs ... 26
7.7.9 Provisioning Services ... 27
7.7.10 Invoking Services ... 27
7.7.11 Registries and Repositories ... 27
7.7.12 Policies and Business Rules ... 27
7.7.13 Events ... 27
7.7.14 Auditing and Logging ... 28
7.7.15 Understanding different logical elements ... 28
8 Operational and IT Systems Layer ... **30**
8.1 Overview ... 30
8.1.1 Summary ... 30
8.1.2 Context and Typical Flow ... 31
8.1.3 Capabilities ... 32
8.1.4 Structural Overview of the Layer ... 33
8.2 Details of ABBs and Supported Capabilities ... 34

Documents
Document Sharing Hub

© ISO/IEC 2016 – All rights reserved

Full document details: BS ISO/IEC 18384-2:2016

8.2.1 Service Delivery ... 34
8.2.2 Runtime Environment ... 35
8.2.3 Virtualization and Infrastructure Services ... 36
8.3 Inter-Relationships between the ABBs ... 36
8.4 Significant Intersection Points with other Layers ... 37
8.4.1 General ... 37
8.4.2 Intersection with the rest of the SOA RA ... 37
8.4.3 Interaction with Cross-Cutting Aspects ... 38
8.4.4 Interaction with Horizontal Layers ... 40
8.5 Usage Implications and Guidance ... 40
8.5.1 Options and Design Decisions ... 40
8.5.2 Implementation Considerations ... 41
8.5.3 Runtime and Deployment View of the SOA RA ... 42

9 Service Component Layer ... 43
9.1 Overview ... 43
9.1.1 Summary ... 43
9.1.2 Context and Typical Flow ... 44
9.1.3 Capabilities ... 44
9.1.4 Structural Overview of the Layer ... 45
9.2 Details of ABBs and Supported Capabilities ... 46
9.2.1 Service Realization and Implementation ... 46
9.2.2 Service Publication and Exposure ... 47
9.2.3 Service Deployment ... 47
9.2.4 Service Invocation ... 47
9.2.5 Service Binding ... 47
9.3 Inter-Relationships between the ABBs ... 48
9.4 Significant Intersection Points with other Layers ... 50
9.4.1 General ... 50
9.4.2 Interaction with Cross-Cutting Aspects ... 50
9.4.3 Interaction with Horizontal Layers ... 52
9.4.4 Interaction with the Services Layer ... 53
9.4.5 Interactions with the Operational and IT Systems Layer ... 55
9.5 Usage Implications and Guidance ... 55
9.5.1 Options and Design Decisions ... 55
9.5.2 Implementation Considerations ... 56

10 Service Layer ... 58
10.1 Overview ... 58
10.1.1 Summary ... 58
10.1.2 Context and Typical Flow ... 59
10.1.3 Capabilities ... 59
10.1.4 Structural Overview of the Layer ... 60
10.2 Details of ABBs and Supported Capabilities ... 62
10.2.1 Service Definition ... 62
10.2.2 Service Runtime Enablement ... 62
10.2.3 Policy Management ... 63
10.3 Inter-Relationships between the ABBs ... 63
10.4 Significant Intersection Points with other Layers ... 66
10.4.1 Interaction with Cross-Cutting Aspects ... 66
10.4.2 Interaction with Horizontal Layers ... 67
10.5 Usage Implications and Guidance ... 68

11 Process Layer ... 69
11.1 Overview ... 69
11.1.1 Summary ... 69
11.1.2 Context and Typical Flow ... 69
11.1.3 Capabilities ... 72
11.1.4 Structural Overview of the Layer ... 73
11.2 Details of ABBs and Supported Capabilities ... 75

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

11.2.1 Process Definition ... 75
11.2.2 Event Handling ... 75
11.2.3 Process Runtime Enablement ... 75
11.2.4 Process Information Management ... 76
11.2.5 Process Integration ... 76
11.2.6 Decision Management ... 76
11.2.7 Process Monitoring and Management ... 77
11.3 Inter-Relationships between the ABBs ... 77
11.4 Significant Intersection Points with other Layers ... 77
11.4.1 Interaction with Cross-Cutting Aspects ... 77
11.4.2 Interaction with Horizontal Layers ... 79
11.5 Usage Implications and Guidance ... 79

12 Consumer Layer ... 80
12.1 Overview ... 80
12.1.1 Summary ... 80
12.1.2 Context and Typical Flow ... 80
12.1.3 Capabilities ... 81
12.1.4 Structural Overview of the Layer ... 82
12.2 Details of ABBs and Supported Capabilities ... 83
12.2.1 Consumer Services ... 83
12.2.2 Presentation Services ... 84
12.2.3 Backend Integration ... 84
12.2.4 Caching and Streaming Content ... 84
12.2.5 Security and Privacy ... 85
12.2.6 Information Access ... 85
12.3 Inter-Relationships between the ABBs ... 85
12.4 Significant Intersection Points with other Layers ... 87
12.4.1 Interaction with Cross-Cutting Aspects ... 87
12.4.2 Interaction with Horizontal Layers ... 88
12.5 Usage Implications and Guidance ... 89

13 Integration Aspect ... 90
13.1 Overview ... 90
13.1.1 Summary ... 90
13.1.2 Context and Typical Flow ... 90
13.1.3 Capabilities ... 91
13.1.4 Structural Overview of the Layer ... 92
13.2 Details of ABBs and Supported Capabilities ... 94
13.2.1 Communication, Service Interaction and Integration ... 94
13.2.2 Message Processing ... 95
13.2.3 Security ... 96
13.3 Inter-Relationships between the ABBs ... 96
13.4 Significant Intersection Points with other Layers ... 98
13.4.1 Interaction with Cross-Cutting Aspects ... 98
13.4.2 Interaction with Horizontal Layers ... 99
13.5 Usage Implications and Guidance ... 101

14 Management and Security (MaS) Aspect ... 101
14.1 Overview ... 101
14.1.1 Summary ... 101
14.1.2 Context and Typical Flow ... 103
14.1.3 Capabilities ... 104
14.1.4 Structural Overview of the Layer ... 108
14.2 Details of ABBs and Supported Capabilities ... 109
14.2.1 Facilities Security Management ... 109
14.2.2 Security Management ... 110
14.2.3 IT Systems Monitoring and Management ... 111
14.2.4 SOA Solution Monitoring and Management ... 112
14.2.5 Business Activity Monitoring and Management ... 113

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

14.2.6 Event Management ... 114
14.2.7 Policy Monitoring and Enforcement ... 114
14.2.8 Configuration and Change Management ... 115
14.2.9 Registry and Repository ... 115
14.3 Inter-Relationships between the ABBs ... 116
14.4 Significant Intersection Points with other Layers ... 118
14.4.1 Interaction with Cross-Cutting Aspects ... 118
14.4.2 Interaction with Horizontal Layers ... 119
14.5 Usage Implications and Guidance ... 120

15 Information Aspect ... 121
15.1 Overview ... 121
15.1.1 Summary ... 121
15.1.2 Context and Typical Flow ... 122
15.1.3 Capabilities ... 122
15.1.4 Structural Overview of the Layer ... 124
15.2 Details of ABBs and Supported Capabilities ... 125
15.2.1 Information Service ... 125
15.2.2 Information Integration ... 126
15.2.3 Information Security and Protection ... 128
15.2.4 Business Information Management ... 128
15.2.5 Business Analytics ... 129
15.2.6 Information Definition and Modeling ... 130
15.2.7 Information Registry/Repository ... 130
15.3 Inter-Relationships between the ABBs ... 130
15.4 Significant Intersection Points with other Layers ... 133
15.4.1 Interaction with Cross-Cutting Aspects ... 134
15.4.2 Interaction with Horizontal Layers ... 135
15.5 Usage Implications and Guidance ... 135

16 Governance Aspect ... 136
16.1 Overview ... 136
16.1.1 Summary ... 136
16.1.2 Context and Typical Flow ... 137
16.1.3 Capabilities ... 138
16.1.4 Structural Overview of the Layer ... 140
16.2 Supported Capabilities ... 143
16.2.1 Governance Lifecycle ... 143
16.2.2 SOA Metadata Storage and Management ... 143
16.2.3 Rule Definition and Management ... 144
16.2.4 Policy Definition and Management ... 144
16.2.5 Monitoring ... 145
16.2.6 Management ... 145
16.2.7 Workflow ... 145
16.3 Inter-Relationships between the ABBs ... 145
16.4 Significant Intersection Points with other Layers ... 147
16.4.1 Interaction with Cross-Cutting Aspects ... 148
16.4.2 Interaction with Horizontal Layers ... 149
16.5 Usage Implications and Guidance ... 150
16.5.1 Options and Design Decisions ... 150

17 Development Aspect ... 151
17.1 Overview ... 151
17.1.1 Summary ... 151
17.1.2 Context and Typical Flow ... 152
17.1.3 Capabilities ... 157
17.1.4 Structural Overview of the Layer ... 159
17.2 Details of ABBs and Supported Capabilities ... 160
17.2.1 Description Development ... 160
17.2.2 Operations Enablement ... 162

© ISO/IEC 2016 – All rights reserved
AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

17.2.3 Testing ... 163
17.2.4 Maintenance ... 164
17.2.5 Publication ... 164
17.2.6 Process Development ... 165
17.2.7 Deployment ... 165
17.2.8 Subscription ... 165
17.3 Inter-Relationships between the ABBs ... 166
17.4 Significant Intersection Points with other Layers ... 171
17.4.1 Intersection with the Rest of the SOA RA ... 171
17.4.2 Interaction with Cross-Cutting Aspects ... 171
17.4.3 Interaction with Horizontal layers ... 173
17.5 Usage Implications and Guidance ... 174
17.5.1 Options and Design Decisions ... 174

18 Common Service Categories ... 179
18.1 General ... 179
18.2 Mediation Services ... 180
18.3 Interaction Services ... 181
18.4 Process Services ... 181
18.5 Information Services ... 182
18.6 Access Services ... 182
18.7 Security Services ... 182
18.8 Partner Services ... 183
18.9 Lifecycle Services ... 183
18.10 Asset and Registry/Repository Services ... 184
18.11 Infrastructure Services ... 184
18.12 Management Services ... 184
18.13 Development Services ... 185
18.14 Strategy and Planning Services ... 185
18.15 Business Application Services ... 185
18.16 Business Services ... 186
18.17 Considering Implementations of Common Service Categories using Reference Architecture ... 186
18.18 Summary ... 188

19 Related Work and Usages of the SOA RA ... 188

Bibliography ... 190

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 38, *Cloud Computing and Distributed Platforms.*

ISO/IEC 18384 consists of the following parts, under the general title *Information Technology — Reference Architecture for Service Oriented Architecture (SOA RA)*:

— *Part 1: Terminology and concepts for SOA*

— *Part 2: Reference Architecture for SOA Solutions*

— *Part 3: Service Oriented Architecture Ontology*

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

Introduction

Service oriented architecture (SOA) is an architectural style in which business and IT systems are designed in terms of services available at an interface and the outcomes of these services. A service (see ISO/IEC 18384-1:2016, 3.20) is a logical representation of a set of activities that has specified outcomes, is self-contained and may be composed of other services but consumers of the service need not be aware of any internal structure.

SOA uses services to create and integrate information systems so that they are suitable for a variety of business and application requirements. SOA enables interactions between businesses without needing to specify specifics of any particular business domain. Using the SOA architectural style can improve the efficiency of developing information systems and integrating and reusing IT resources. In addition, using the SOA architectural style can help enable rapid response of information systems to ever-changing business needs.

ISO/IEC 18384 is intended to be a single set of SOA technical principles, specific norms, and standards for the world-wide market to help remove confusion about SOA and improve the standardization and quality of solutions.

ISO/IEC 18384 defines the terminology, technical principles, reference architecture, standard service categories and ontology for SOA. This part of ISO/IEC 18384 can be used to introduce SOA concepts, as a guide to the development and management of SOA solutions, as well as be referenced by business and industry standards.

ISO/IEC 18384 contains three parts:

a) ISO/IEC 18384-1, which defines the terminology, basic technical principles and concepts for SOA;

b) ISO/IEC 18384-2, which defines the detailed SOA reference architecture layers, including a metamodel, capabilities, architectural building blocks, as well as a set of categories or types of services in SOA solutions;

c) ISO/IEC 18384-3, which defines the core concepts of SOA and their relationships in an ontology.

The targeted audience of ISO/IEC 18384 includes, but is not limited to, standards organizations, architects, architecture methodologists, system and software designers, business people, SOA service providers, SOA solution and service developers, and SOA service consumers who are interested in adopting and developing SOA.

Users of this part of ISO/IEC 18384 will find it useful to read ISO/IEC 18384-1 for an understanding of SOA basics. ISO/IEC 18384-1 should be read before reading or applying this part of ISO/IEC 18384. For those new to the SOA reference architecture, Clause 4 provides a high-level understanding of the Reference Architecture for SOA Solutions. The remaining clauses provide comprehensive details of the architectural building blocks and trade-offs needed for an SOA solution and a set of common categories (or types) of SOA services to help populate that architecture. ISO/IEC 18384-3 contains the SOA ontology, which is a formalism of the core concepts and terminology of SOA, with mappings to both UML (see Reference [16]) and OWL (see Reference [17]). ISO/IEC 18384-3 can be used independent of or in conjunction with ISO/IEC 18384-1 and this part of ISO/IEC 18384.

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 18384-2:2016

INTERNATIONAL STANDARD ISO/IEC 18384-2:2016(E)

Information technology — Reference Architecture for Service Oriented Architecture (SOA RA) —

Part 2: Reference Architecture for SOA Solutions

1 Scope

This part of ISO/IEC 18384 describes a Reference Architecture for SOA Solutions which applies to functional design, performance, development, deployment and management of SOA Solutions. This part of ISO/IEC 18384 includes a domain-independent framework, addressing functional requirements and non-functional requirements, as well as capabilities and best practices to support those requirements.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 18384-1, *Information technology — Reference Architecture for Service Oriented Architecture (SOA RA) — Part 1: Terminology and concepts for SOA*

ISO/IEC 18384-3, *Information technology — Reference Architecture for Service Oriented Architecture (SOA) – Part 3: Service Oriented Architecture Ontology*

ISO/IEC 15474-1, *Information technology — CDIF framework — Part 1: Overview*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 18384-1 apply.

3.2 Abbreviated terms

ABB	Architectural Building Block
B2B	Business To Business
BAM	Business Activity Monitoring
BPEL	Business Process Execution Language
BPMN	Business Process Model and Notation
CEP	Complex Event Processing
CICS	Customer Information Control System
CRM	Customer Relationship Management
EA	Enterprise Architecture
EAI	Enterprise Application Integration
EJB	Enterprise Java Beans
ERP	Enterprise Resource Planning

© ISO/IEC 2016 – All rights reserved

AVA Documents Document Sharing Hub

Full document details: BS ISO/IEC 18384-2:2016

FCAPS	Fault, Configuration, Accounting, Performance, Security
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IT	Information Technology
ITIL	Information Technology Infrastructure Library
JAX-WS	Java Api For Xml Web Services
KPI	Key Performance Indicator
MDM	Master Data Management
NFR	Non Functional Requirement
POCO	Plain Old Clr Object
POJO	Plain Old Java Object
QOS	Quality Of Service
RA	Reference Architecture
RAS	Reliability Availability Scalability
SBB	Solution Building Block
SCA	Service Component Architecture
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Database Language SQL
WSDL	Web Services Description Language
WSRP	Web Services For Remote Portlet

4 Notations

Interpretation of diagrams should be done as follows.

4.1 UML

Most diagrams are not UML. Those that are have text to that effect before the diagram identifying the type of UML diagram so that the reader knows how to interpret it.

4.2 Entity-Relationship

Entity Relationship diagrams with boxes, lines arrows and circled numbers should be interpreted according to the following rules.

— Boxes are the metamodel concepts, layers, architectural building blocks, capabilities, or components.

— Arrows are relationships between metamodel concepts, where single arrow heads show direction of relationship; double-headed arrows indicate the relationship is bidirectional.

— Relationships are named, represented as labelled lines or arrows, and no cardinality is implied.

— Cardinality indications are participation in the relationship and well known mathematical conventions are used to express them (*== 0..*; 0..1==optional and only 1; 1==required as defined in ISO/IEC 15474-1).

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

4.3 Flows

Flows should be interpreted with the following rules:

— boxes that are layers, architectural building blocks or components;

— directional arrows showing the direction of the flow between the boxes;

— circled numbers on the flow arrows show the sequence of the flow and are used as a point of reference in any explanatory text.

4.4 Layer Diagrams

Layer diagrams, layered boxes and arrows, layer diagrams are usually some part of the SOA RA Layer and Aspects diagram in Figure 3 and should be interpreted using the following rules.

— Boxes that are layers or capabilities with architectural building blocks as smaller boxes positioned with in them. Horizontal boxes are functional layers. Vertical boxes or underlying boxes are cross-cutting aspects.

— Arrows between layers indicate interactions between the layers; capabilities of one layer are used by another.

— Arrows between ABB boxes within layers indicate interactions between ABBs within or across layers where single arrow heads show direction of interaction and double-headed arrows indicate the interaction is bidirectional.

4.5 Capability diagrams

Capability diagrams showing larger boxes as capabilities containing smaller boxes being architectural building blocks needed to fulfil that capability should be interpreted with the following rules.

— Relative positioning of the capability boxes is not relative.

— White indicates ABBs that are defined in the current layer. Some essential ABBs owned by other layers that are used to support the capabilities of this layer are shown in darker shades of grey. Additional ABBs defined in other layers may be used as needed.

— Grey ABBs are named using the name of the layer or aspect owning the ABB as a prefix followed by a colon and then the ABB name. For example: Governance: Registry/Repository indicates that the Registry/Repository ABB is owned by the Governance Aspect and the reader can go there to find out more information about it.

5 Conventions

The introduction is followed by a high-level summary of the 10 layers and the service types defined in this part of ISO/IEC 18384 for the convenience of the reader and to allow some readers who are only looking for a high-level 'executive level' understanding to easily read 7.1, 7.2, and 7.6. Each summary in 7.5 is repeated in the clause documenting the respective layers in the first clause labelled 'summary'.

This is followed by the definition and explanation of the metamodel used in this part of ISO/IEC 18384. The metamodel defines Layer, Capabilities and ABB concepts along with other core logical concepts. ABBs and capabilities are each defined uniquely in each layer. Capabilities and ABBs may require capabilities and ABBs defined in other layers in order to do fulfil their architectural requirements. The layers, capabilities and ABBs in this part of ISO/IEC 18384 are all logical elements and any reference to these logical element 'performing', 'supporting', 'interacting', or 'responsible for' means that when a SOA solution is developed, the physical realization of the capabilities and ABBs are actually 'performing', 'supporting', 'interacting', or 'responsible for'.

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

Each layer of the SOA RA is documented in a separate clause, Clause 5 through Clause 14. Each Layer is documented using the same organization:

1. Name of the Layer	
1.1 Overview	
1.1.1 Summary	
1.1.2 Context and Typical Flow	
1.1.3 Capabilities	- explains capabilities supported by layer
1.2 Details of ABBs and Supported Capabilities	
1.2.1 Details of ABBs	- detailed definitions of the ABBs
1.2.2 Structural Overview of the Layer	- overview of capabilities and ABBs they support
1.3 Inter-Relationships between the ABBs	- interactions between ABBs with in the layer
1.4 Significant Intersection Points with other Layers	
1.4.1 Interaction with Cross-Cutting Aspects	- interactions with Aspects supporting layer
1.4.2 Interaction with Horizontal Layers	- interactions with functional layers
1.5 Usage implications and Guidance	- best practices and advice
1.5.1 Options and Design Decisions	- considerations

At the beginning of each layer, there is small unlabelled figure of the SOA Solution stack for this SOA RA in the upper right hand corner that indicates with dark grey which layer in the reference architecture that this clause is defining.

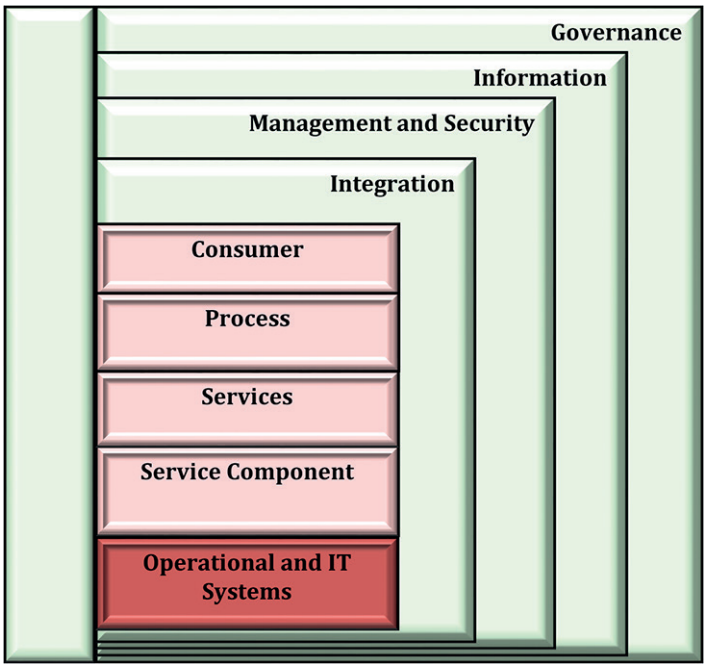


Figure 1 — SOA RA Layer/Aspect Indicator Diagram

For example, Clause 8 has Figure 1, at the beginning that indicates that the Clause is documenting the Operational and IT Systems Layer.

Clause 18 defines in detail the types of services commonly found in SOA solutions.

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

6 Conformance

ISO/IEC 18384 contains three parts which have different conformance requirements, as follows:

a) *Part 1: Terminology and concepts for SOA* – conformance only to terms and adherence to the semantics in the definitions;

b) *Part 2: Terminology and concepts for SOA* – conformance only to semantics of the metamodel and any Layers, ABBs, or capabilities that are used;

c) *Part 3: Service Oriented Architecture Ontology*, – conformance for OWL or non-OWL applications.

Conformance to this part of ISO/IEC 18384 is defined as follows.

This part of ISO/IEC 18384 is intended to be a set of best practices and guidance on creating successful architectures using the SOA paradigm. It is not intended to be mandatory or normative or to use for claiming conformance.

It is a qualitative standard in that users of the SOA RA may choose to deviate from the standard in certain areas. An organization may choose among the various ABBs provided by the SOA RA for conducting assessments, designing, or implementing architectures.

If a document, product or standard claims conformance with this part of ISO/IEC 18384 then it shall use the same semantics for the Metamodel, and any specific Layers, ABBs, or Capabilities used.

Some of the ABBs in the SOA RA are foundational and may be needed in most SOA solutions and others are only needed for some SOA solutions.

A service-oriented architecture or SOA solution does not conform to the partially layered architecture if there are certain things that are logically missing, then the missing building blocks that are key for the specific instance of the architecture should be identified. The corresponding Solution Building Blocks should be present.

Service-oriented architectures and SOA solutions can be different but conformant.

7 Overview

7.1 Introduction to SOA

Service Oriented Architecture (SOA) is an architectural style that supports service orientation and is a paradigm for business and IT (see ISO/IEC 18384-1:2016, 3.48). This architectural style is for designing systems in terms of services available at an interface and the outcomes of services. A service is a logical representation of a set of activities that has specified outcomes, is self-contained, may be composed of other services and is a "black box" to consumers of the service. (see ISO/IEC 18384-1:2016 ,3.20).

In common with other architectural styles, SOA

— places unique requirements on system infrastructure,

— has environment-specific implementations, constrained or enabled by context and described within that context,

— requires appropriate governance of IT and systems and EA,

— has business solutions that are designed to mirror real-world business activities, and

— provides criteria to allow consumers to determine whether the business solution offered has been properly and completely executed in accordance with their expectations.

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

In addition, SOA has distinguishing characteristics that set it apart from other architectural styles, notably

— it promotes the use of open standards and interfaces in order to achieve interoperability and location opacity,

— services and processes are designed explicitly to operate both within or between organizations,

— it requires clear descriptions of the service offered,

— services and processes are designed to mirror real-world business activities,

— service representation uses business descriptions to provide context (i.e. business process, goal, rule, policy, service interface, and service component),

— it requires appropriate governance of service representation and implementation,

— service composition is used as a means to implement business processes, and

— it provides criteria to allow service consumers to determine whether the service has been properly and completely executed in accordance with the service description.

Service orientation is utilized for enabling efficient co-operation between autonomous (business) entities (e.g. clients, service providers, and third parties) that wish to collaborate to achieve common goals. Collaboration between the business entities can take the form of simple client-provider interaction, supply chains or virtual organizations that may take the form of bilateral or multi-lateral choreographies. In fact, organizations are a general concept and can represent multiple organizations or ecosystems of organizations.

Business-oriented SOA takes 'service' as its basic element to constitute and integrate information systems so that they are suitable for a wider variety of application requirements. Some of the benefits of using SOA are improvement in the efficiency of development of information systems, efficiency of integration and efficiency of re-use of IT resources. It also enables agile and rapid response of information systems to constantly changing business needs.

While an increasing number of solutions are being implemented using SOA in many different industries, a single set of SOA technical principles, specific norms, and standards have not been established for the world-wide market. Existing products and solutions have used various standards, methods and technologies. As a result, there is confusion about the effectiveness of SOA. To improve standardization and quality of solutions, it is necessary to establish a unified set of general technical principles for SOA, a standard reference architecture and a common set of service types based on best practices and experience.

It should be noted that the SOA principles defined in ISO/IEC 18384-1 are applicable to software engineering and can also be applicable to system engineering in order to formalize service-based systems (i.e. complex systems, federation of systems, systems of systems, enterprise architecture).

Service oriented computing is a software engineering paradigm for developing, delivering and governing services whose functionality is implemented as software components and where co-operation between business entities is enabled by information and communication technology. These activities can be private to an organization (e.g. deploying a service), collaborative between a set of business entities (e.g. service invocations and choreographies), or joint activities for maintaining the viability of the service ecosystem (e.g. publishing new services).

7.2 Introduction to the SOA Reference Architecture

The SOA Reference Architecture consists of two complementary parts: first, a reference architecture for SOA solutions and second, a set of common service types or categories. Both of these will support architects to develop architectures for SOA solutions and service-oriented business applications. This

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

includes all of the logical and physical design and runtime components, needed to develop, deploy, and operate service-oriented solutions for businesses. The complementary parts are enumerated as follows.

a) The SOA Reference Architecture enumerates the fundamental elements of an SOA solution or enterprise architecture standard for solutions and provides the architectural foundation for the solution by specifying the capabilities and architectural building blocks that support the realization of those capabilities.

   The SOA Reference Architecture (SOA RA) has 10 layers representing 10 key clusters of capabilities, considerations and responsibilities that typically emerge in the process of designing a SOA solution.

   Each layer supports, and is defined in terms of: requirements, logical, and physical concerns. The requirements concerns reflect what the layer enables and includes all of its capabilities. The logical concern includes all the architectural building blocks (ABBs), design decisions, options, and metrics. The physical concern of each layer includes the realization of each logical concerns using a particular chosen technology platform, standards and products that are determined by taking into consideration the different options and selecting one along with the documentation of architectural decisions that led to the selection. The actual realization of the architecture will be through a set of products or platforms that will be left open to the implementer of the standard.

   This Clause documents a high-level description of each of the layers. The detailed descriptions of the requirements and capabilities, logical architectural building blocks, and physical mapping are in Clauses 8 to 17.

b) Clause 18 documents a range of common domain specific (functional) and non-domain specific (supporting) categories of services that can be used as a checklist to help architects understand and decide which of these types of services are appropriate for their solution.

   Services are naturally a key concept in any service-oriented architecture and it is important to realize that there can be many different kinds. Services are categorized according to what they do, i.e. their function or purpose, in order to aid in ensuring both coverage and shared understanding. Of course, other categorization schemes are also possible and helpful.

   Partitioning services into groups and common categories is a common activity in the development of the services and service portfolio. These groups and common services type can help how both business and IT stakeholders have a common view and understanding of the architecture and the development and deployment of the portfolio of services that are part of that architecture.

   The layers and the service categories are intended to support the entire lifecycle of SOA solutions, from planning, design and development, though to deployment, execution, updating and eventual decommissioning.

   This SOA RA is derived from The Open Group SOA Reference Architecture technical standard (see Reference [8] for more information), China's Technical Reference Architecture for SOA Solutions (see ISO/IEC TR 30102:2012, Annex D; see Reference [30]), and SOA Related Function — Japanese Technical Reference Model (TRM) for the Government Procurement of Information Systems (see ISO/IEC TR 30102:2012, Annex F; see Reference [30]).

7.3 Metamodel

The metamodel defines the terms and relationships needed to understand this part of ISO/IEC 18384.

A textual description of the model with an illustrative diagram (UML) is provided. This includes a discussion on the capabilities concept as used in this part of ISO/IEC 18384. This subclause finishes with the architectural assumptions that the reader should understand before trying to understand the SOA RA.

Taking a pragmatic approach, for each layer, there are three concerns that should be supported by the SOA RA: requirements (exemplified by the capabilities for each layer), logical (exemplified by the architectural building blocks) and physical (this concern will be left to the implementation of the standard by an adaptor of the standard).

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

The requirements concern reflects what the layer enables and includes all of its capabilities; the logical concern includes all the architectural building blocks, design decisions, options, KPIs, etc.; while the physical concern of each layer includes the realization of each logical concern using technology, standards and products that are determined by taking into consideration the different architectural decisions that are necessary to be made to realize and construct the architecture. The actual realization by a set of products or platform will be left open to the implementer of the standard.

This specification provides specific focus on the logical concerns of the SOA Reference Architecture and provides a model for including key architectural considerations and making architectural decisions through the elements of the metamodel. Examples of and best practices for these considerations and decisions are documented at the end of each layer.

Layers provide an abstraction and consolidation mechanism that consistently groups a set of related and cohesive capabilities and associates them with logical components called Architectural Building Blocks (ABBs). Thus, each layer provides a logical container for a unique set of ABBs and the capabilities they help realize. However, realizing some capabilities might require the use of ABBs assigned to multiple layers and cross-cutting aspects in order to realize those capabilities in their SOA solutions.

The separation of concerns[1)] and design considerations that entail from these layers are integral to an SOA. The SOA Reference Architecture has 10 layers, out of which three primarily support the provider side responsibilities of service interface and implementation and, three support service consumption and five are cross-cutting in that they provide support for all layers. The 10 layers comprise a comprehensive set of identified capabilities from which a SOA solution may be constructed. Most SOA architectures and solutions will use a subset of the ABBs in the layers. This may be due to the business context and constraints imposed on the solution. In addition, as an organization matures in their use of SOA and their SOA solutions become more sophisticated, they may choose to use a different set of ABBs.

Indeed, the need to combine the capabilities represented by ABBs in different layers demonstrates that a strict separation of concerns may not be possible for more complex (composed) capabilities.

It is important to realize that a Reference Architecture is not a Solution Architecture. Capabilities satisfy the requirements for functionality that these layers collect. Architectural Building Blocks (ABBs) are the logical components whose implementations carry out that functionality specified by the capabilities. An ABB can be realized by a Solution Building Blocks (SBB). An SBB is a physical or socio-technical construct whose structure and properties conform to the requirements associated with an ABB.

To summarize, a layer is an abstraction that is useful in grouping cohesive capabilities. Similarly, an ABB is an abstraction that is associated with a capability and the outcome the capability will produce. To be precise, it is the implementation of ABBs that realize the capability and its outcomes in SOA solutions. Layers, as a collection of ABBs, consolidate capabilities. The implementations constitute Solution Building Blocks (SBBs) that are "concrete", identifiable, and subject to governance and management.

When this part of ISO/IEC 18384 refers to a layer as "providing" or being "responsible" for a capability, it should be interpreted as actually the implementation of the necessary ABBs collected in the layer that is realizing the capability and its outcomes. Similarly, the ABBs, as abstract entities, represent elements that provide capabilities. When this part of ISO/IEC 18384 refers to ABBs providing, interacting, or being responsible for, in fact, it is the implementation of the ABBs that is providing, interacting, or behaving in a solution.

SOA solutions use ABBs in their abstract sense to communicate or document the design or architecture of the solution and use implementations of the ABBs to realize the solutions. Interactions between ABBs associated with a single layer are referred to as "interactions within a layer". However, solutions often make use of ABBs in multiple layers to represent more complex patterns; the use of such ABBs may be

---

1) If we are attempting to separate concern A from concern B, then we are seeking a design that provides that variations in A; do not induce or require a corresponding change in B (and usually, the converse). If we manage to successfully separate those concerns, then we say that they are "decoupled", for obvious reasons. As a simple example, a car's side view mirror controls are decoupled from its accelerator, permitting us to adjust our mirrors without fear that the car will speed up or slow down. [http://www.infoq.com/articles/separation-of-concerns]

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

referred to as “interactions across layers”. The simpler wording is often used for convenience; however, in practice, the interactions occur between ABB implementations.

To be explicit, the layers, as defined, are available to be used in any SOA solution. However, for any given problem/solution context, only a proper subset of all the capabilities of a layer may be required to be realized. For example, a simple SOA solution may not have any processes and therefore the capabilities and ABBs in associated with the Process Layer are logically not required or need to be realized.

Often, there is a need to use similar capabilities in multiple layers, like checking security access or accessing assets in a registry/repository. The ABBs that represent such widely used functionality are factored into special layers called cross-cutting aspects. For example, many layers might have local security permission checking and access to special registry/repositories but the capabilities and the associated ABBs are collected in the cross-cutting Management and Security Aspect.

Three of the layers address the implementation and interface with a service (the Operational and IT Systems Layer, the Service Component Layer and the Services Layer). Two Layers and an Aspect support the consumption of services (the Process Layer, the Consumer Layer and the Integration Aspect). Five of them support cross-cutting aspects of a more supporting (sometimes called non-functional or supplemental) nature (the Information Architecture Aspect, the Management and Security Aspect, the Integration Aspect and the Governance Aspect). The SOA RA, as a whole, provides the framework for the support of all the elements of an SOA solution, including all the components that support services and their interactions.

This logical view of the SOA Reference Architecture addresses the question, “If I build an SOA solution, what would it look like and what abstractions should be present?” The question answered by this part of ISO/IEC 18384 is, informally, “If I assess an architecture proposing to be built upon SOA principles, what considerations and building blocks should be present and what should be assessed with?”

The SOA Reference Architecture has 10 layers representing 10 key clusters of considerations and responsibilities that typically emerge in the process of designing an SOA solution or defining enterprise architecture standard. Also, each layer is designed to correspond to reinforce and facilitate the realization of each of the various perspectives of SOA business value.

The SOA reference architecture enumerates the fundamental elements of an SOA solution or enterprise architecture standard for solutions and provides the architectural foundation for the solution.

Figure 2 uses UML (ISO/IEC 19505-2) to illustrate the relationship between the metamodel concepts.

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

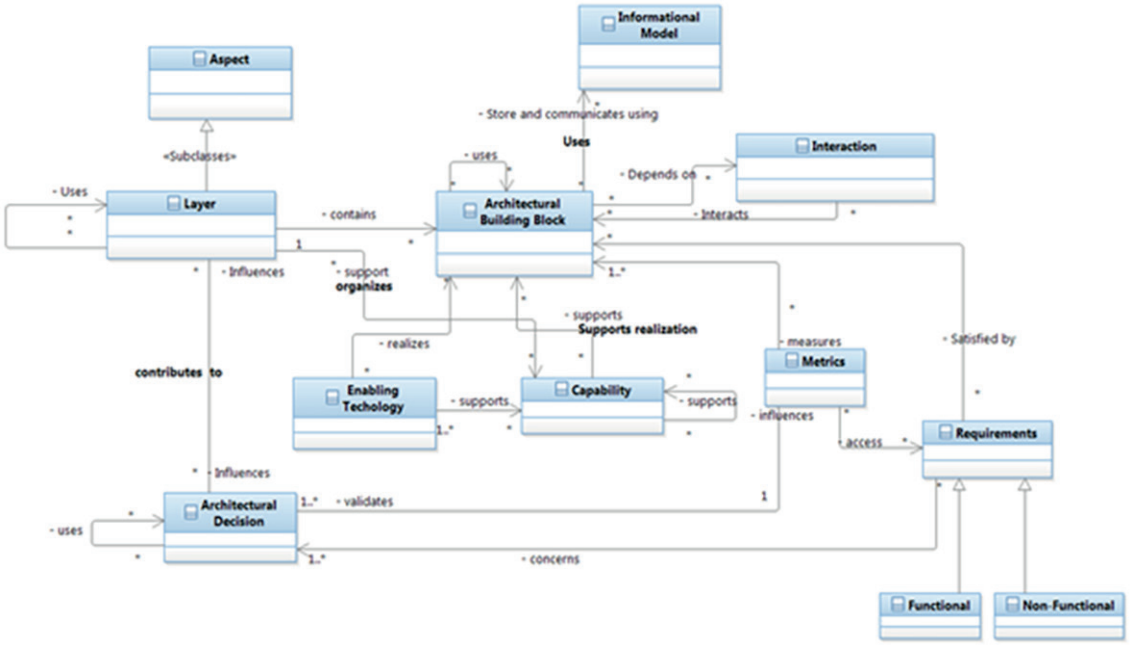


Figure 2 — Metamodel for instantiating the SOA Reference Architecture for a Given Solution

As shown in Figure 2, the metamodel of the SOA Reference Architecture includes the following elements.

— Layer: an abstraction of a grouping of a cohesive set of related capabilities and their identified ABBs, interactions among ABBs, interactions among layers, and the influences on and by architectural decisions.

— Aspect: is a layer that contains capabilities and functionality that are widely useful across functional layers and may need to be coordinated across multiple roles (see ISO/IEC 18384-1:2016 ,8.3).

— Capability: an ability that an organization, person, or system possesses to deliver a product or service. A capability satisfies a requirement or category of requirements that fulfil a strongly cohesive set of needs.

— Architectural building block (ABB): a logical element that supports realization of one or more capabilities. Implementations of ABBs provide the basic building blocks for composing SOA solutions. Each layer can be thought to contain a set of ABBs that define the key responsibilities of that layer. In addition, ABBs are connected to one another across layers and thus provide a natural definition of the association between layers. The particular connection between architectural building blocks that recur consistently in order to solve certain classes of problems can be thought of as patterns of architectural building blocks. These patterns can be described not only in terms of static relationships between ABBs but also in terms of the interaction sequences between the ABBs that address specific scenarios. In this reference architecture, each ABB resides in a layer, supports capabilities, and has responsibilities. It contains attributes, dependencies, constraints and relationships with other ABBs in the same layer or different layer.

NOTE ABBs from other layers are named with an owning layer name prefix, e.g. Governance Aspect: Registry/Repository.

— Solution Building Block (SBB): Solution Building Block realizes one or more architectural building blocks by providing a technical, organizational, or social implementation, a system, that conforms with the requirements associated with the ABB through the capabilities it supports. A solution building block can be implemented with enabling technology, such as a service directory, for example. (See ISO/IEC/IEEE 42010.)

— Architectural decision: a decision derived from the identification and analysis of the available options. The architectural decision is driven by architectural requirements, and involves [illegible]

© ISO/IEC 2016 – All rights reserved
AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

rules and standards, ABBs, Key Performance Indicators (KPI), and Non Functional Requirements (NFRs) to decide on standards and protocols to define a particular instance of an Architectural Building Block. This can be extended, based on the instantiation of the reference architecture to the configuration and usage of ABBs. Existing architectural decisions can also be reused by other layers or ABBs.

— Interaction: an abstraction of the various relationships between ABBs. This may be represented by the use of diagrams, other patterns, pattern languages and interaction protocols.

— Metrics: analytical measurements intended to quantify the state of a system and are often called Key Performance Indicators (KPI). Architectural decision will help decide what key performance indicators should be defined and over time the KPIs will validate that the right decisions were made and if they should be adjusted.

— Requirements: documented needs that a particular layer, ABB, or SOA solution satisfies. Requirements may act as input to an architectural decision.

— Functional requirements: requirements that specify functions that systems or system components perform (see ISO/IEC 25000).

— Non-functional requirements (NFR): specify the overall quality requirements of the SOA solution or system. NFRs help address architectural cross-cutting concerns such as security and management.

— Enabling technology: technology used to realize ABBs.

— Information Model: a structural model of the information associated with ABBs including data exchange between layers and external services. The information model includes the metadata about the data being exchanged.

7.4 Capabilities

A capability is an ability that an organization, person, or system possesses. Capabilities are typically expressed in general and high-level terms and may require a combination of organization, people, processes, and technology to achieve. There are pure business capabilities such as Process Claim or Provision Service Request and there are technical capabilities such as Service Mediation or policy based Content Based Routing. Both business and technical capabilities are represented in SOA and enabled and supported by SOA.

Using a capability modeling as part of the approach has some major advantages.

— Allows architects to focus on the “what” rather than the “how”. This supports an abstract approach that is focused on the requirements of the solution.

— Enables business capabilities to be aligned to the technical capabilities required to service them.

— Enables architects to derive and re-balance the plan for SOA adoption in an agile fashion. For example, if an organization foresees the need for integrating services across different business units, it might require a certain set of SOA layers and architectural building blocks to be enabled.

This SOA Reference Architecture enables deriving the solution architecture using capabilities; however, the capability mapping process itself is out of scope for this part of ISO/IEC 18384 and is normally a part of the organizational service modeling methodologies. For example, the business capability to cross-sell requires a technical capability to have a common shareable set of data, where the data is from different systems in an enterprise. This, in turn, requires shareable metadata about data, supporting “information services” in some form and the ability to transport, mediate and share data from the disparate systems in a common form. Thus, a business capability (cross-selling) is dependent on technical capabilities [the need to be able to have a common view of data (information services), the need to mediate, integrate and transport the data, etc.]. Each of these capabilities maps to architectural building blocks supported by the layers of the SOA RA.

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

A capability-based approach enables architects to determine when they need a particular SOA RA layer and functions and to help facilitate making decisions when organizational priorities change. In this part of ISO/IEC 18384, the SOA RA enables architects to determine if there are interdependencies and technical requirements for a layer and its constituent building blocks, beyond those defined by business capabilities, to create a coherent set of capabilities which the SOA RA needs to satisfy.

The layers in the SOA reference architecture provide a convenient means of consolidating and categorizing the various capabilities and building blocks that are required to implement a given service oriented architecture. 7.5 explores the details of these layers and their constituent elements.

7.5 Reference Architecture for SOA Solutions

7.5.1 Overview of Reference Architecture

This subclause presents an overview of the reference architecture for SOA solutions. The reference architecture is a high level abstraction of an SOA, partitioned and factored into layers. Each layer provides a set of required capabilities that addresses a specific subset of characteristics and responsibilities.

The following subclauses refer to Figure 3 and provide detailed descriptions for each of the 10 layers and their interactions.

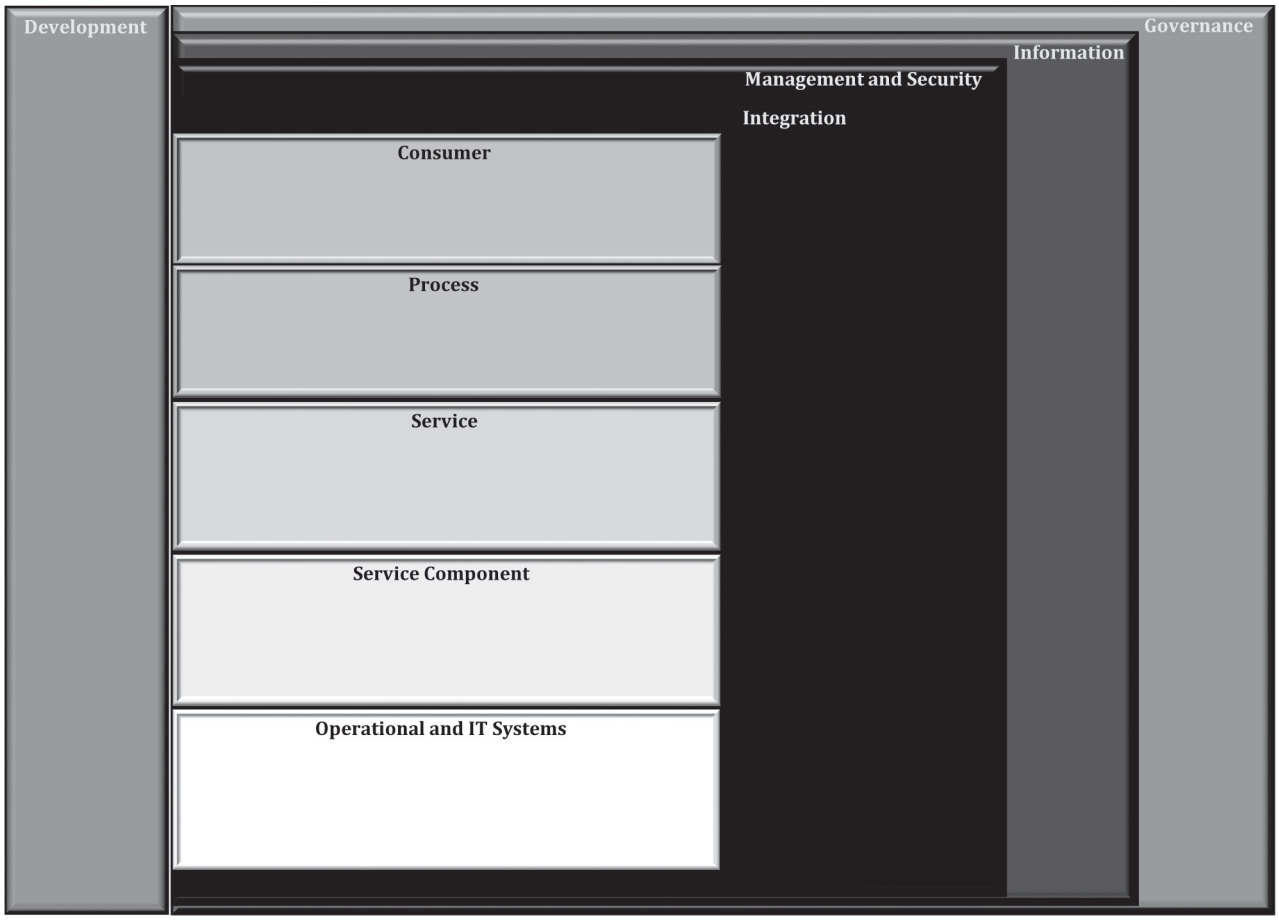


Figure 3 — Reference Architecture for SOA Solutions

Figure 3 depicts an SOA Reference Architecture for SOA solutions as a set of logical layers as a partially-layered architecture. This means that one layer does not solely depend upon the layer below it. Therefore, a consumer can access both the Process Layer and the Service Layer directly. This is important because a given SOA solution may not need the capabilities and ABBs in the Process Layer and would not benefit

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

from the business value proposition associated with the Process Layer. The Consumer Layer may then interact directly with the Service Layer for efficiency and architectural simplicity. Capabilities and ABBs of the Process Layer can be added to the architecture at any time in the future when there is a need for the business value they provide. The degree to which a given organization realizes the full SOA Reference Architecture will differ according to the level of SOA maturity they exhibit and the underlying requirements of the organization.

Three of the horizontal layers (dark grey) address the implementation and interface with a service (the Operational and IT Systems Layer, the Service Component Layer and the Services Layer). Three of the horizontal layers support the consumption of services (the Service Layer, Process Layer, and the Consumer Layer). Service layer is used by both. Five of them (light grey) support cross-cutting Aspects of a more supporting (sometimes called non-functional or supplemental) nature (the Information Aspect, the Management and Security Aspect, the Integration Aspect and the Governance Aspect). One vertical layer supports development for both functional layers and Aspects. The SOA RA as a whole provides the framework for the support of all the elements of a SOA, including all the components that support services and their interactions.

The SOA Reference Architecture does not assume that the provider and the consumer are in one organization (or many organizations) and supports both SOA within the enterprise, as well as across multiple enterprises in the industry ecosystem. The need for both intra-and inter-enterprise SOA is important, since SOA a foundation of Cloud Computing as being defined in ISO/IEC 17789 (see Reference [29]). Decoupling providers from consumers using services is essential when the providers and consumers are in different organization. Decoupling is just as important for providers and consumers in the same organization when they need separation along the lines of business relationships. These organizations use this architectural style (where one is the consumer and the other is the provider) and customize it for their own needs for integrating and interacting among themselves. The lower layers (Services Layer, Service Component Layer and Operational and IT Systems Layer) are concerns for the provider and the upper ones (Service Layer, Process Layer and Consumer Layer) are Aspects for the consumer. Note that the Service Layer is a shared layer of concern between both Providers and Consumers. Each layer and the relationships between the layers is described in the subsequent clauses.

Note that there are five horizontal layers that are functional in nature and relate to the functionality and the overall capabilities that are provided and implemented by the SOA solution. The underlying layers support the horizontal layers by providing a set of cross-cutting Aspects. These Aspects span and apply to all of the horizontal functional layers but are clustered around independent notions such as information, integration and governance in their own right and serve as cross-cutting Aspects of the SOA architectural style. The layering of these cross-cutting aspects indicates that each aspect on the bottom applies to all of the aspects above it. For example, governance applies to and should be considered for all elements in all of the layers above it. Information is important for Management and Security, Integration and the functional layers. Management and Security apply to integration and the functional layers. However, each layer can use elements in the layers above and beneath them, so Governance uses metrics monitoring in the management layer and the management layer uses policies defined in the Governance aspect.

In addition to being an important template for defining an SOA solution at a logical level, the SOA RA is also a useful tool in the design of vendor-neutral SOA solutions. This is because it enables the objective identification of SOA infrastructure requirements. The SOA RA provides a well-factored decomposition of the SOA problem space, which allows architects to focus on those parts of an SOA solution that are important in the context of the problem they are solving and to map the required capabilities onto vendor product capability rather than try and reverse engineer an SOA solution architecture from the capability of a particular vendor's products. This set of requirements can be used to better leverage the various capabilities provided by a mix of different vendors who may offer the same ABB. Using the same SOA RA, SOA business services can be delivered based on the same deployment framework. The following clauses provide a short high-level description of each of the layers.

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

7.5.2 Operational and IT Systems Layer

The Operational and IT Systems Layer captures the new and existing organization infrastructure needed to support the SOA solution at design, deploy and run time. This includes the following:

— all infrastructures to run the SOA and its components;

— all operational and runtime hosting of components, both physical and infrastructural;

— all deployment time components;

— all resources to support services, data, and application systems;

— all assets required to support the functionality of the service in the SOA, where the assets may include custom or packaged application assets, new services, services created through composition or orchestration, infrastructure services, etc.

This layer provides the building blocks supporting the operational systems which implement the functional capabilities of the other horizontal layers and the supporting/cross-cutting Aspects. In particular, the capabilities supported by this layer include providing operational and runtime hosting, infrastructure services and infrastructure virtualization, functional delivery support including support for service implementations and realizations.

A number of existing software systems are part of this layer. Those systems include but are not limited to the following:

— existing monolithic custom applications;

— existing transaction processing systems;

— existing databases;

— existing package applications and solutions including ERP and CRM packages;

— legacy applications and systems;

— access to existing web services;

— systems built from web services;

— IT Infrastructure;

— Enterprise Application Integration (EAI) systems;

— Service Resources providing access to existing business elements or third-party elements;

— Data Resources providing physical storage of data in the business solution;

— Application System Resources providing specific business functions;

— access to existing business services.

This layer represents the intersection point between the actual runtime infrastructure and the rest of the SOA which runs on that infrastructure. In addition, it is the integration point for an underlying Infrastructure as a service construct and the rest of the SOA in the wider context of cloud computing. Key requirements for this layer are outlined in the capability clause describing the capabilities provided to fulfil those requirements.

7.5.3 Service Component Layer

The Service Component Layer contains capabilities that support components which represents the implementation or “realization” for services or operation on services; hence, the name Service Component. The layer contains the functional and technical components that facilitate a Service

© ISO/IEC 2016 – All rights reserved
AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

Component to realize one or more services. Service components reflect the definition of the service they represent, both in its functionality and its management and quality of service interactions. They "bind" the service interface to the implementation of the service in the Operational and IT Systems Layer. Service components may be hosted in containers which support a service description (see 10.2.2.1 for more information on containers).

The Service Component Layer manifests the IT conformance with each service interface defined in the services layer; it helps guarantee the alignment of IT implementation with service description.

Each Service Component

— realizes one or more services,

— provides an enforcement point for "faithful" service realization (ensures quality of service and service level agreements),

— enables IT flexibility by strengthening the decoupling in the system by hiding volatile implementation details from service consumers,

— offers a façade behind which technologies can be deployed as required to enable service functionality, and

— generally contains business-specific logic with no reference to integration logic.

The Service Component Layer enables flexibility through encapsulation and by enabling loose coupling. A separation of concerns is achieved such that the service consumer can assume that the realization of the service is faithful to its published description (service compliance) and the service provider will ensure that such compliance is achieved. The details of the service realization are of no consequence to the consumer. The service provider is consequently able to replace one component with another having the same interface, producing the same outcomes (real world effects), and having identical conditions of use without any impact on service consumers.

7.5.4 Services Layer

The Services Layer consists of the logical representation for all the services. The Service Layer can be thought of as containing the service descriptions for business capabilities, services and IT manifestation used and created during design time, as well as runtime service contracts and descriptions that are used at runtime.

The Services Layer is one of the horizontal layers which provide the business functionality supported in the SOA and describes functional capabilities of the services in the SOA.

The description provides consumers with information needed to invoke the business functions exposed by a provider of the service; ideally, this may be done in a platform independent manner. Service descriptions may include or have links to the following:

— descriptions of the abstract functionality offered by the service similar to the abstract stage of a WSDL description (see Reference [14]). Note that the use of WSDL is illustrative and the description can be done in any language supporting description of the functionality;

— policy documents;

— SOA management descriptions;

— attachments that categorize or show service dependencies.

Some of the services in the Service Layer may be versions of other services in the set implying that a significant successor/predecessor relationship exists between them.

This layer contains the contracts which include service descriptions that bind the provider and consumer. Services are offered by service providers and are consumed by service consumers (service requestors). Service Components or existing enterprise applications (such as legacy systems and packaged

© ISO/IEC 2016 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

applications) are responsible for the actual implementation or realization of a service. The Operational and IT Systems Layer supports the runtime environment; therefore, implementation of the service components may reside in or use a container and other ABBs in the Operational and IT Systems Layer.

The Services Layer supports the following:

— functional capabilities or services that enable business capabilities those businesses perform in order to achieve a business outcome;

— supporting capabilities to define and specify the “services” in terms of service description;

— supporting capabilities to enable the runtime execution of service and the support of service virtualization.

7.5.5 Process Layer

The Process Layer covers the process representation, composition methods, and building blocks for aggregating loosely coupled services as a sequence of steps aligned with business goals. Data flow and control flow are used to enable interactions between any combination of services and business processes. The interaction may exist within an enterprise or across multiple enterprises.

Business capabilities are realized through the execution of one or more business processes. These business processes may be realized through service compositions (e.g. orchestrations, choreographies, or collaborations) and include support for manual human interactions and long-lived transactions. The evolution of service composition into flows (e.g. choreographies of services bundled into a flow) can act together to establish an SOA solution. These SOA solutions support specific use cases and business processes.

This layer includes information exchange flows between participants (individual users and business entities), resources, and processes where the exchanged information may include unstructured and non-transactional messages. The business logic is used to form service flows as parallel tasks or sequential tasks based on business rules, policies, and other business requirements.

The Process Layer performs three-dimensional process-level handling: top-down, bottom-up, and horizontal. From the top-down direction, this layer provides capabilities and ABBs to help decompose business requirements into tasks comprising activity flows, each being realized by existing business processes, services, and service components. From the bottom-up direction, the layer provides facilities to compose existing business processes, services, and service components into new business processes. From the horizontal direction, the layer provides service-oriented collaboration control between business processes, services, and service components.

In summary, the Process Layer in the SOA Reference Architecture plays a central coordinating role in connecting business level requirements and IT-level solution components through collaboration with the Integration Aspect, Management and Security Aspect, Information Aspect, Governance Aspect, and Services Layer.

7.5.6 Consumer Layer

The Consumer Layer is the point where consumers, either human actors or SOA solutions, interact with the SOA solution ecosystem. It enables SOA solutions to support a client-independent, channel agnostic set of functionality, which is separately consumed and rendered through one or more channels (client platforms and devices). In this discussion, channels can be thought of as the platforms by which SOA consumers access services through the SOA. Examples of channels include front-ends and interactive voice response (IVR) systems, which could both leverage the same core functionality within the SOA. Thus, the Consumer Layer is the point of entry for all internal and external interactive consumers, including services acting as consumer (for example, in B2B scenarios).

For human consumers, the Consumer Layer is often realized through a user interface that accepts requests and returns responses. This interface may enable specific users to customize preferences, integrate with consumer channels, including rich clients such as mashups and Ajax (see Reference [18])

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

and act as a mechanism for the underlying SOA to expose its functionality. Standards such as Web Services Remote Portlets (WSRP) (see Reference [15]) may leverage services at the application interface or presentation level.

The user interface provides the visible portion of the Consumer Layer capabilities but the Consumer Layer may also incorporate other business processes dictated by policy or desired business outcomes. For example, consumer layer capabilities may include the point where in-bound service consumer requests have security and other quality of service policies asserted to ensure that the request is secure and brought into the context of the SOA via collaboration with other Aspects.

For consumers that are other services or SOA solutions, the Consumer Layer points to defined service interfaces, where in parallel with the human use of the Consumer Layer, the service interface may point to a composition which includes the application of other business processes, such as security and quality of service from the Management and Security Aspect. The Consumer Layer provides the capability to quickly create a front end for business processes and other service compositions to respond to changes in business requirements. This "front end" can be a new service interface, a new user interface, or an appropriate combination. It enables channel independent access to those business processes supported by a variety of applications and platforms.

This decoupling between the consumer and the rest of the underlying SOA provides organizations the ability to support agility, enhance reuse and improve quality and consistency.

7.5.7 Integration Aspect

That Integration Aspect enables the loose coupling between the request and the concrete provider by matching the Service Request and Service Implementation. This loose coupling provided by the Integration Aspect is not only a technical loose coupling addressing protocols, binding, locations or platforms, but can also be a business semantic loose coupling performing required adaptations between service requester and provider.

There are numerous sets of capabilities the Integration Layer supports to overcome structural and semantic mismatches at service interfaces. For example, the Integration Aspect supports the integration with solution platforms by the other layers in the SOA RA using mediators, transformers, or adapters to enable the access of services by other ABBs, layers and the capabilities associated with service transport. Mediation includes transformation, routing, and protocol conversion. Integration includes adapters and service enablement. Routing includes service interaction and service virtualization. Transport includes service messaging and message processing. It can be thought of as the plumbing that interconnects the SOA solution. Both of these can be driven by policy.

Integration support includes capabilities that enables and provides the capability to mediate between the service requester and the service provider. It provides capabilities to transform, route, and convert protocols, enabling support for heterogeneous environment, adapters, service interaction, service enablement, service virtualization, service messaging, message processing and transformation.

Routing support includes supporting capabilities through which a consumer/requestor can connect to the correct service provider. These can start with point-to-point capabilities for tightly coupled endpoint integration and cover the spectrum to a set of intelligent routing, mediation, and other transformation mechanisms often associated with, but not limited to, mediation services provided by an Enterprise Service Bus (ESB). The service description specifies a location where a service is provided and an associated binding and for part of a service contract. A mediation service, on the other hand, provides a location independent mechanism for integration, service substitution, and virtualization.

Transportation support that occurs here is primarily the integration of service component, service, and Process Layers (the "functional" layers). For example, this is where binding (late or otherwise) of services occurs for process execution. This allows a service to be exposed consistently across multiple customer facing channels such as web, IVR, CRM client (used by Customer Service Rep), etc. The transformation of response to HTML (for web), Voice XML (for IVR), or XML string can be done via XSLT functionality supported through mediation service transformation capability in the Integration Aspect.

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

7.5.8 Management and Security Aspect

The Management and Security Aspect supports non-functional requirement (NFR) related issues as a primary concern of SOA and provides a focal point for dealing with them in any given solution. It contains the capabilities for ensuring that a SOA meets its requirements with respect to monitoring, reliability, availability, manageability, transactionality, maintainability, scalability, security, safety, life cycle, auditing and logging, etc. It includes the same scope as the traditional Fault, Configuration, Accounting, Performance, and Security (FCAPS) from ITIL (see Reference [27]) or Reliability, Availability, Serviceability (RAS) (see Reference [28]).

Management and Security is especially important for SOA solutions to enable the loosely coupled solutions to scale and efficiently fulfil non-functional requirements. This layer provides capabilities that maintain and ensure the "quality of service (QOS)". In order to enable both management and security, this layer provides capabilities that

— provides solution management of various concerns, such as availability, reliability, security, and safety, as well as the mechanisms to support, track and monitor, and manage solution quality controls,

— provides the ability to monitor and enforce a multitude of policies and corresponding business rules including business level policies, security policies, access privileges and data access policies,

— serves as an observer of the other layers and can create events when a non-compliance condition is detected or (preferably) when a non-compliance condition is anticipated,

— provides the service and SOA solution lifecycle processes with the capabilities required to ensure that the defined policies, non-functional requirements (NFRs), and governance regimens are adhered to for service and SOA solution lifecycle processes,

— supports the ability to monitor and manage at both the business level [in terms of key performance indicators (KPIs), events and activities in business processes] and the IT systems level (for the security, health and well-being of IT systems, services, applications, networks, storage and processors), and

— supports monitoring and capturing of service and solution metrics in an operational sense and signalling of non-compliance with non-functional requirements relating to service qualities and policies associated with each SOA layer. Service metrics are captured and connected with individual services to allow service consumers to evaluate service performance, creating increased service trust levels.

Finally, the same kinds of management and monitoring that apply to businesses are important for managing services and SOA solutions and may need extensions to handle the service-oriented nature and the cross domain boundaries of many SOA solutions. These traditional capabilities supported by SOA solutions include

— IT Systems Monitoring and Management,

— Service and SOA Solution Monitoring and Management,

— Business Activity Monitoring and Management,

— Event Management,

— Configuration and Change Management,

— Policy Monitoring and Enforcement,

— Lifecycle management, and

— Auditing and Logging.

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

SOA security addresses the protection of the SOA solution against threats across the vulnerability dimensions of a service-oriented architecture. This includes protecting the interactions between service consumers and service providers, as well as protecting all of the elements that contribute to the architecture. Examples of threats to be protected from are destruction, corruption, removal, disclosure, and interruption. Some of the security dimensions to help protect against these threats include access control, authentication, non-repudiation, data confidentiality, communication security, data integrity, availability, and privacy.

The capabilities that explicitly address security are as follows.

— Security Management: Manages and monitors security and secure solutions. This provides ability to manage roles and identities, access rights and entitlements, protect unstructured and structured data from unauthorized access and data loss, enable the IT organization to manage IT-related risks and compliance, and provide the automation and audit basis for security management.

— Facilities Security Management**:** This category of capabilities provides the command centre for security management, as well as the operational security capabilities for non-IT assets and services to ensure protection, response, continuity and recovery. It also supports for security of physical assets such as locations, facilities, services, inventory, physical access control, human identity, etc.

Important areas for policy enforcement are security, auditing, messaging transportation, infrastructure availability, service availability and reliability. Responses (dispensations and appeals) to non-compliance and exceptions are also defined by the governance aspect.

7.5.9 Information Aspect

The Information Aspect provides capabilities for enabling development of a unified representation of the information assets of an organization as represented by its IT services, systems and SOA solutions. The unified representation for an organization may require rationalization and ongoing coordination of assets from across many organizations. Representing information enables business needs and processes to be in alignment with one or more business vocabularies.

The Information Aspect includes information architecture, business analytics and intelligence, metadata considerations. It focuses on the inclusion of key considerations pertaining to information architectures that can be used as the basis for the creation of business analytics and business intelligence through the analysis of data held in repositories. These repositories include metadata content that is stored using capabilities provided in this layer. The Information Aspect supports the establishment of information services capability architectures that can be used as the basis for the creation of business analytics and business intelligence through data marts and data warehouses. It also supports the ability for an information services capability, enabling a virtualized information data layer capability. This enables the SOA to support data consistency and a systematic improvement in data quality.

The Information Aspect supports the following capabilities:

— ability to support information services capability that will support a shared, common and consistent expression of data;

— ability to integrate information across diverse actors and organizations in order to effectively communicate across the different organizational domains;

— ability to define metadata that is used across the SOA RA and, in particular, the metadata that is shared across the layers;

— ability to enable secured and protected information via interaction with the Management and Security Aspect;

— ability to support business activity monitoring and critical to the usage of the RA and its realization.

An information virtualization and information service capability may involve the ability to retrieve data from different sources, transform it into a common format and expose it to consumers using different protocols and formats.

Documents
Document Sharing Hub

© ISO/IEC 2016 – All rights reserved

Full document details: BS ISO/IEC 18384-2:2016

7.5.10 Governance Aspect

SOA Governance defines policies, guidelines, standards and processes that reflect the objectives, strategies and regulations to which services and SOA solutions align with business needs over time. The success of a SOA solution is often in terms of meeting business value objectives. SOA governance activities should conform to Corporate, IT and Enterprise Architecture governance principles and standards relevant to the SOA ecosystem in which the services and SOA solutions are intended to interact. The SOA Governance should also be adapted to match and support an appropriate SOA maturity level.

The Governance Aspect includes both SOA solution governance (governance of processes for policy definition and enforcement), as well as service governance (service life-cycle). This covers the entire lifecycle and portfolio management of the service and SOA solutions (e.g. SLAs, capacity, performance, security and monitoring). This Aspect also supports governance that needs to be coordinated across organizations, where service consumers and service providers are using services from other organizations.

The goal of the SOA Governance Aspect is to ensure consistency of the service and solution portfolio and supporting lifecycle processes. A given service or SOA solution may be part of more than one portfolio but it needs to conform to the defined governance for each Corporate or Enterprise domain to which it applies. Thus, some services or SOA solutions may not be appropriate for every governance domain.

The SOA Governance Aspect provides an extensible and flexible SOA governance framework based on ISO/IEC 17998 that supports the alignment of business and IT, including the following:

— Service Level Agreements based on requirements for quality of service and key performance indicators (KPIs);

— capacity and performance management policies;

— design time concerns such as business rules.

As a part of the governance framework, a governance regimen (i.e. the customized compliance, dispensation and communication processes to govern the SOA lifecycle and portfolio management) will need to make use of capabilities to store and access governance artefacts, as well as capabilities for managing and enforcing policy, monitoring metrics and managing the configuration and governance of the solution. Organizations may also need a strong change control capability to support changes to governance and the ensuing management of those changes.

To ensure ongoing business and IT alignment, the governance processes and business condition should be continuously evaluated and updated. These separate processes may use the same capabilities as the governance regimen and Management and Security Aspect.

The Governance Aspect supports the following capabilities:

— definition of policies, compliance and exceptions characteristics;

— monitoring the health of SOA services, solution and governance via the Management and Security Aspect;

— identification of metrics reporting on compliance, exceptions, service health, and versions;

— an incorporation of for business rules into the governance structure.

7.5.11 Development Aspect

The Development Aspect contains all of the components and products needed to develop and change implementations of SOA services and solutions. The service implementations should include the development of or use of implementations in the Operational IT and Systems layer, Service Component Layer, Service Layer, Process Layer and cross cutting aspects. Service implementations should

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

encapsulate existing systems and resources such that late binding of services can be supported to promote loose coupling.

Development includes solution and service design, modeling, implementation and deployment. Operational and management capabilities are the responsibility of the Management and Security Aspect. Maintenance uses capabilities from Development Aspect and Management and Security Aspect.

Tools that support the Development Aspect include the entire suite of architecture tools, modeling tools, development tools, visual composition tools, assembly tools, methodologies, debugging aids, instrumentation tools, asset repositories, source code control, discovery agents, and publishing mechanisms that may be used to construct a SOA solution.

The Development Aspect supports the following capabilities:

— development, configuration, debugging and testing environments for the construction of services;

— testing of services and SOA solutions, ranging from isolated testing to testing within the operational environment or ecosystem;

— coordination with monitoring to effectively provide for continuous testing throughout the operational lifetime;

— service encapsulation of existing application systems or data resources;

— reusing existing assets to develop services.

7.6 Common Services Categories

Service is naturally a key concept in any service-oriented architecture and it is important to realize that services can be categorized in many different ways. Clause 18 defines a standard categorization scheme for services where they are categorized according to what they do, i.e. their function or purpose, in order to aid in ensuring both coverage and shared understanding. This Clause provides a brief listing of the functional categories; for details, see Clause 18.

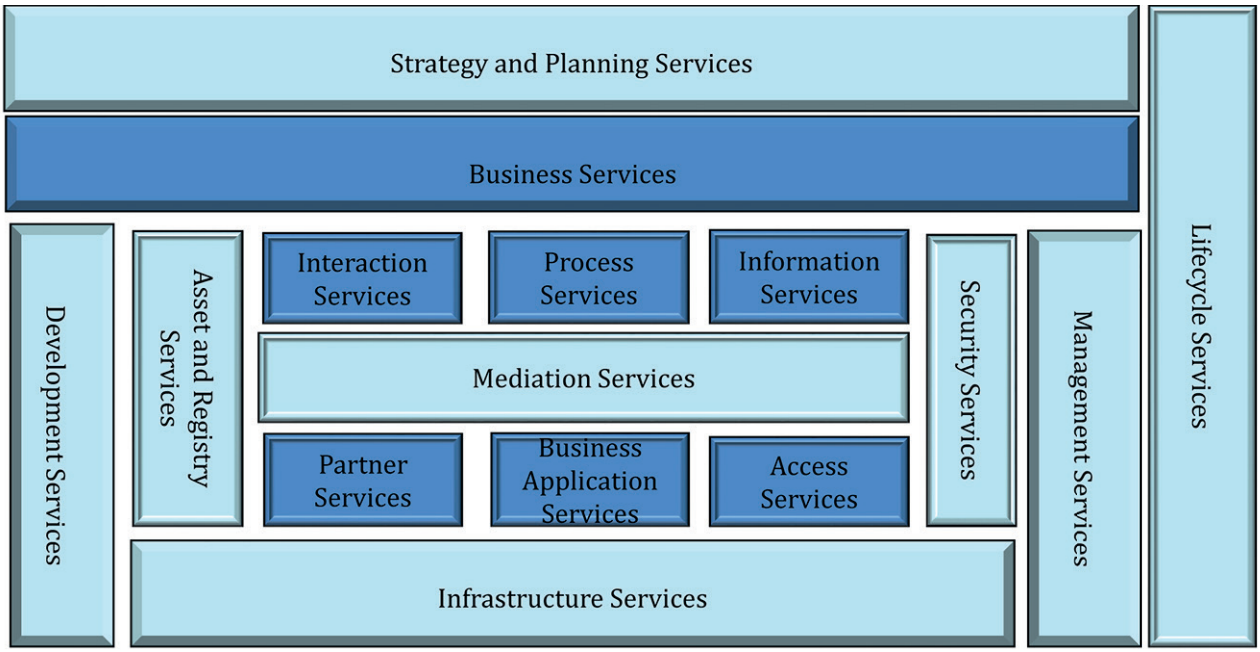


Figure 4 — Common Service Categories

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

The categories of services are broken down in Figure 4. Dark coloured services are considered to be domain specific where implementations of the services are required to be unique to the domain or solution being developed.

The remaining services categories are considered to be domain independent where implementations of services of this category can be used directly in many different domains or solutions and may be used without extensive customization.

Services categories are as follows.

— Mediation Services - category of services that provide the functions related to connecting service consumers with service providers where connecting can support effective and optimized mediation, transformation and routing of requests across the network and meet the goals of the business.

— Interaction Services - category of services that provide the presentation logic of the business design, supporting the interaction between other solutions and end-users.

— Process Services - category of services that execute various forms of compositional logic, such as business process flows.

— Information services - category of services that contain the data logic of business solution including the provision of access to the persistent data of the business, the support of data composition and to provide an architecture for managing the flow of data across an organization.

— Access services - category of services that encapsulate adapters to integrate legacy and new functionality into the SOA solution, including by wrapping or augmenting the logic of the existing functions to better meet the needs of the business design.

— Security services - category of services that address the protection against threats across the vulnerable dimensions of an SOA, including interactions between service consumers and service providers, as well as all of the elements that contribute to the architecture.

— Partner services - category of services that enable custom interaction between business partners, such as the semantics of partner interoperability that have a direct representation in the business design.

— Lifecycle services - category of services that support managing the lifecycle of SOA solutions and all of the elements that comprise them across development and management ranging from strategy to infrastructure.

— Asset and registry/repository services - category of services that manage and provide access to the information assets stored in configuration management databases, registries and repositories.

— Infrastructure services - category of services that form the core of the information technology environment for hosting SOA solutions.

— Management services - category of services that represent the set of management tools used to monitor metrics, service flows, the health of the underlying system, the attainment of service goals, the enforcement of administrative policies and recovery from failures.

— Development services - category of services that encompass the entire suite of architecture tools, modeling tools, development tools, visual composition tools, assembly tools, methodologies, testing, debugging aids, instrumentation tools, and discovery agents needed to construct services and SOA solutions.

— Strategy and planning services - category of services that support creating vision, blueprint and transition plan for improving business outcomes.

— Business application services - category of services that implement core business logic where these service implementations are created specifically within a business model.

Documents
Document Sharing Hub

© ISO/IEC 2016 – All rights reserved

Full document details: BS ISO/IEC 18384-2:2016

— Business services - category of services that capture the business function and are offered to external consumers, often referred to as higher level or coarse-grained services.

Note that all common services should provide well-defined functionality that can be customized for consumer needs rather than a new service for every variation. For example, an Interaction Service can access data in XML and apply custom style sheets to give the desired presentation; it does not require a different service for each presentation.

See Clause 18 for more information on these service categories and how they relate to the underlying implementations and architectural building blocks of SOA.

7.7 Assumptions and Key Concepts

7.7.1 General

This Clause discusses the consequences of using the major metamodel concepts of "Capability" and "Requirements" on this part of ISO/IEC 18384 and the approach taken to describe the layers of the SOA Reference Architecture and how those layers may interact.

7.7.2 Functional and Non-functional

A Service Oriented Architecture (SOA) solution is defined by the set of Functional and Non-functional Requirements (NFRs) that constrain it. Functional requirements are business capabilities imperative for business operations including business processes, the business and IT services, the components, and underlying systems that implement those services. NFR categories for SOA include security, availability, reliability, manageability, scalability, latency, governance, etc.

Security refers to the ability to ensure that a service can be accessed in a secure way (including authorization and authentication) and that the information provided by a service is only available by those that should be able to access the data. Availability refers to the percentage of how a specific service can be available during a specific time frame. Reliability refers to the ability of a specific service that stays no fail during a specific time frame. Scalability refers to the ability of a specific service that supports various scales of consumer groups. Latency refers to the delay of accessing a specific service due to internal implementation details and processes. Governance is defined in 7.5.10 and Clause 16.

7.7.3 Requirements

The underlying requirements which determine the capabilities that the SOA solution architecture supports are determined by the following:

— a set of service requirements which includes business (a.k.a functional) and NFRs on a service;

— service requirements result in the documented capability that a service needs to deliver or is expected to deliver;

— the provider view of a service requirement is the business and technical capability that a given service needs to deliver given the context of all of its consumers;

— the consumer view of a service requirement is the business and technical capability that the service is expected to deliver in the context of that consumer alone.

The fulfilment of any service requirement may be achieved through the combination of capabilities from one or more layers in the SOA Reference Architecture (SOA RA).

7.7.4 Services

Services themselves have a description element and a functional element. The service description states what the service does for consumers while the functional element implements what a service is obligated to provide based on the service description. The service description identifies a service

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

interface and associated binding through which the underlying functional element realizes the capability. This model addresses services exposing capabilities implemented through legacy assets, new assets, services composed from other services, or infrastructure services.

7.7.5 Documenting the Layers

The five vertical or cross-cutting layers, namely, Development Aspect, Integration Aspect, Management and Security Aspect, Information Aspect, and Governance Aspect, are basically supporting capabilities realized through implementation and vendor products. The five functional or horizontal layers, namely, Operational and IT Systems Layer, Service Component Layer, Services Layer, Process Layer, and Consumer Layer, will support the functional capabilities of the architecture. The Operational and IT Systems Layer will provide the actual runtime for all layers, vertical or horizontal.

Thus, in the clauses of this part of ISO/IEC 18384 that describe the layers in detail, we

— provide an overview and description of the layer and motivation behind the layer,

— describe the key capabilities supported by the layer,

— provide a structural overview of the layer which includes detailed descriptions of ABBs that enable the capabilities of the layer, and

— describe the interactions within the layer and across layers of the SOA RA.

In general, a theme is followed where each layer has a part which supports a set of capabilities/ABBs which support the interaction of the layer with other elements in the SOA RA, a part which supports the actual capabilities that the layer satisfies, and a part which supports the orchestration and management of the other ABBs to support the layer's dynamic, runtime existence. Thus, in the clauses of this part of ISO/IEC 18384 that describe the layers in greater detail, we

— provide an overview and description of the layer and motivation behind the layer,

— provide the key capabilities supported by the layer,

— provide a structural overview of the layer which includes detailed description of ABBs enabling the responsibilities of the layer, and

— describe the interactions within the layer and across other layers in the SOA RA.

7.7.6 Logical and Physical Elements

The distinction between logical/design-time and physical/runtime elements of the SOA are described below.

— All runtime elements that are part of the physical or operational or deployment architecture are actually represented in the Operational and IT Systems Layer of the SOA RA.

— The SOA RA will provide a logical abstraction of the runtime at a moment in time and expand that view in terms of a set of layers that are depicted by horizontal/functional layers and a set of supporting or cross-cutting layers (backdrop) of the SOA RA. The five horizontal layers deal with enabling the business capabilities required by the application running on the architecture. The five supporting layers support the functionality that is provided by the horizontal layers. The horizontal or functional layers include the Service Component Layer, the Services Layer, the Process Layer, and the Consumer Layer.

— Each part of the runtime is abstracted into a layer whose capabilities differ significantly from that of the other layers. For example, the Consumer Layer provides capabilities that support interaction with consumers of the service, whereas the Service Component Layer provides capabilities that support the implementation of the service in a Service Component. That Service Component, in turn, will be running in a container within the Operational and IT Systems Layer, at runtime.

Documents
Document Sharing Hub

© ISO/IEC 2016 – All rights reserved

Full document details: BS ISO/IEC 18384-2:2016

7.7.7 Interactions between Layers

From 7.3, layers are an abstraction useful in grouping cohesive capabilities. Layers, as a collection of ABBs, consolidate capabilities. The implementations of ABBs are Solution Building Blocks (SBBs) that are "concrete", and identifiable, and subject to governance and management. When this part of ISO/IEC 18384 refers to a layer as "providing" or being "responsible" for a capability, it should be interpreted as "the implementation of the necessary ABBs collected in the layer that is realizing the capability and its outcomes".

SOA solutions use ABBs in their abstract sense to communicate or document the design or architecture of the solution and use implementations of the ABBs to realize the solutions. Interactions between ABBs associated with a single layer are referred to as "interactions within a layer". However, solutions often make use of ABBs in multiple layers to represent more complex patterns; the use of such ABBs may be referred to as "interactions across layers". The simpler wording is often used for convenience; however, in practice, the interactions occur between ABB implementations.

For example, Figure 5 describes an interaction between the Consumer Layer and the Process Layer using the Integration Aspect.

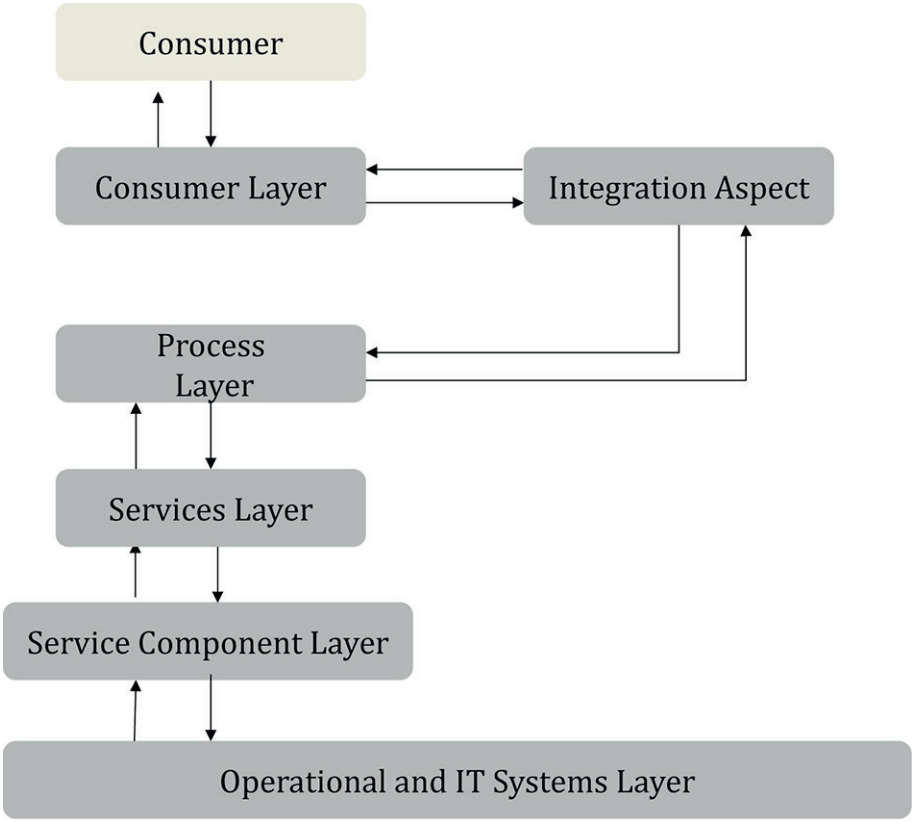


Figure 5 — Typical Interactions among the Layers of the SOA RA

A typical interaction flow among the layers of the SOA RA is described below.

— Service consumers request services using the Integration Aspect.

— The Integration Aspect invokes the business process in the Process Layer which is using one or more services.

— Business Process Layer invokes the Services Layer.

— The Services Layer binds and invokes Service Components in the Service Component Layer.

© ISO/IEC 2016 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

— Service Components in the Service Component Layer invoke Solution Components from the Operational and IT Systems Layer to carry out the service request.

— The response is sent back up to the service consumer.

7.7.8 Understanding ABBs

ABBs are the key building blocks of a particular layer.

An ABB is an abstraction that is associated with a capability and the outcome the capability will produce. To be precise, it is the implementation of ABBs that realize the capability and its outcomes in SOA solutions. The implementations constitute Solution Building Blocks (SBBs) that are "concrete", identifiable, and subject to governance and management.

When this part of ISO/IEC 18384 refers to ABBs 'providing', 'interacting', or being 'responsible' for, in fact, it is the implementation of the ABBs that is providing, interacting, or behaving in a solution. The simpler wording is often used for convenience; however, in practice, the interactions occur between ABB implementations.

ABBs are the components providing key capabilities that are expected from that layer. For example, the Integration Aspect is expected to provide meditation, routing, and protocol transformation capabilities. Therefore, these capabilities are realized by a set of solution building blocks, each of which provides exactly that atomic unit of architectural capability needed.

ABBs in the SOA RA and artefacts that are created as part of creating a solution are not always the same, For example, a process definition or a process model artefact is used by the Business Process ABB in the Process Layer to describe the underlying business process.

Some ABBs that are related to the functionality within an application such as

— Consumer Layer: Portal to loan processing application,

— Process Layer: Initiation of a loan processing application,

— Services Layer: Services that are required to support the loan processing application,

— Component Layer: Software components that need to be built that support the realization and implementation of the services, and

— Operational and IT Systems Layer: Actual runtime environment in which the components, legacy systems, all back-end applications, and packaged applications reside and run.

The cross-cutting layers, which include the Integration Aspect, serve as the central point where integration occurs across the enterprise and consolidates the design decisions into a set of software components that facilitate and enable the actual integration of applications.

The Information Aspect includes all data and information needed to support and instantiate each of the layers. Note that it is a cross-cutting layer indicating that each of the horizontal layers may have and will have data associated with their functioning and they will draw upon this data from the Information Aspect via metadata, actual data, or analytics.

The Management and Security Aspect ensures Quality of Service (QoS) by serving, as a collection point for the administration and control, or monitoring and management of most if not all of the Non-Functional Requirements (NFRs). This includes security, availability, configuration, monitoring, and management capabilities, to name a few.

Lastly, the Governance Aspect provides a central point in which policies are set into registries and repositories. In general, governance capabilities and processes are administered and run centrally via this layer. Note again that this layer is the underlying layer for all of the other layers within the architecture and it relates to and touches upon all other functional and cross-cutting layers of the SOA RA.

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

Not all ABBs are required for every single implementation of an SOA. Instead, every project will choose from the list of building blocks within each layer of the SOA RA and select those that are appropriate for the particular project. In the case where SOA RA is applied for an enterprise architecture standard, there will be patterns of ABBs within the layers that will be selected. Some will be mandatory and some will be optional and projects will be chosen to conform to a fixed set of patterns and configurations of the ABBs along with product selections and implementations.

7.7.9 Provisioning Services

Provisioning means doing all of the tasks necessary to make the service available for consumers to be able to invoke. Part of the responsibility of provisioning is updating the registry/repository (in the Governance Aspect) which contains the service metadata that consumers need to find, bind, and invoke services.

7.7.10 Invoking Services

The Integration Aspect is a layer of choice through which services are invoked but it is not the only means to invoke services. Services can, in a less mature SOA, be invoked directly by the consumer without the level of indirection associated with an integration aspect.

The Consumer Layer provides access to services. It allows consumers to consume services. The means of consumption of services can be a GUI or programmatic means of access to the services.

The Consumer Layer provides capabilities that support invoking the service endpoint. It accesses the services via the Integration Aspect or it can access the service directly if the given architecture does not wish to implement an Integration Aspect. Invocation is done by opening up access to the Consumer Layer.

7.7.11 Registries and Repositories

Registries and repositories may need to exist or be used in multiple layers during a physical implementation in many projects; therefore, the registry/repository is organized in the Governance Aspect of the SOA RA. In this way, it can be managed, monitored, and administered from a single logical location even though physically it may be federated or distributed. Logically, the functional layers and aspects may use the Registry/Repository ABB in the Governance Layer when needed.

7.7.12 Policies and Business Rules

Policy definition is the responsibility of the Governance Aspect. The definition of Policy Enforcement Points (PEPs) and policy enforcers is the responsibility of the individual horizontal/functional layers, such as Services Layer, Process Layer, etc. The enforcement of those policies is then the responsibility of a cross-cutting Management and Security Aspect where these capabilities have been consolidated. Therefore, monitoring and enforcement of the policies are the responsibility of the Management and Security Aspect while the administration of the policies remains the responsibility of the Governance Aspect.

Business rules are also a cross-cutting architectural concern. They need to be consistently applied across the multiple layers in the SOA or, over time, there is a tendency to develop rule divergence and lose the consistency that SOA brings. Therefore, business rule definition and management is the responsibility of the Governance Aspect and its validation and enforcement is the responsibility of the Management and Security Aspect.

7.7.13 Events

No single layer is responsible for events. Many layers and corresponding ABBs collaborate to produce a composite capability related to events such as Complex Event Processing (CEP), Business Activity Monitoring (BAM), etc. The cross-cutting nature of events is handled within the Integration Aspect, Management and Security Aspect, and Information Aspect. The building blocks for producing and listening for events reside within the Integration Aspect. The Management and Security Aspect is responsible for monitoring and managing the events. The Information Aspect is used to define the events.

Documents
Document Sharing Hub

© ISO/IEC 2016 – All rights reserved

Full document details: BS ISO/IEC 18384-2:2016

Business events occur during the course of business process execution. An example of this is Fraud management which occurs during the execution of a business process certain data items' sequences and patterns are spotted that trigger a set of rules relating to possible fraud. This will trigger a notification and subsequent action. A set of data access or update patterns can also be a reason to trigger events in the Information Aspect.

Thus, CEP is addressed across multiple layers of the SOA RA; it may start from the Process Layer or Information Aspect.

7.7.14 Auditing and Logging

Enabling, auditing, and logging appropriately is important to the long term health of the SOA Solution. Auditing and logging enable compliance with regulations, resolution of problems, and provide a base of information about how the SOA Solution is running long term that can be analysed and mined using business analytics tools. In addition, these analytics can feed into governance of the SOA solution, providing key information on how well the SOA solution is continuing to meet business needs and enabling the business to choose to update the SOA solution.

Auditing and logging is done using capabilities from the Integration Aspect (logger and auditor ABBs), the management and security layer (Traceability Enabler/Auditor and Audit and Log ABB and logging manager).

Care should be taken when defining auditing and logging policies because there is a trading off between what will be audited and logged and resource consequences, e.g. effects on performance, network bandwidth, and storage availability.

7.7.15 Understanding different logical elements

What is the difference between an ABB, Service Component, Deployment Unit, Solution Component, and a Solution Building Block?

An ABB is a logical entity. Each layer is composed of ABBs with which it implements its capabilities.

ABBs can be separated into those that provide the infrastructure to support the SOA and the services themselves. An example of an ABB which provides infrastructure support could be a Service Container ABB in the Services Layer or a Mediator ABB in the Integration Aspect. An example of an ABB related to implementing or providing a service is the Service Component ABB.

Let us understand the ABBs which form a service first. Each service has a Service Component, which is an ABB in the Service Component Layer. A Service Component is composed of a Functional and a Technical Component. The Functional Component provides the functional capability that the service implements. This could be a legacy system invocation, a Java POJO (Plain Old Java Object), a COBOL subroutine, etc. or one or more other Functional Components. The Technical Component encapsulates the technical capabilities to support standards compliance and technical support that a service provides. However, the Service Component and its associated Technical and Functional Components are logical, design-time assets.

Now let us look at the creation of a service and its conversion into a runtime entity. The service is created using an Integrated Development Environment (IDE) of some sort, with resulting Service, Technical, and Functional Components. Next, the Service, Functional, and Technical Components are bundled up into a Deployment Unit. Finally, the Deployment Unit is deployed into a runtime environment, becoming a Solution Component. Figure 6 shows the structural relationship between the different service-related ABBs.

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

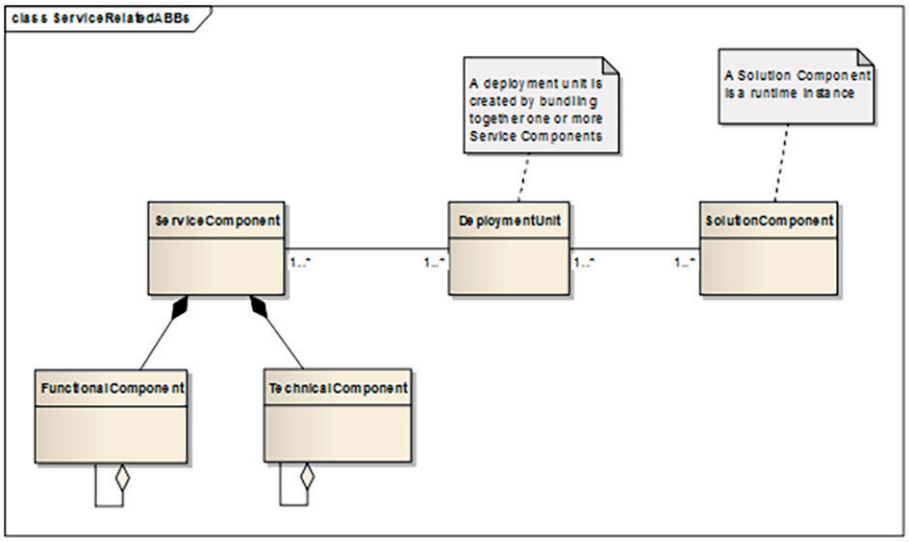


Figure 6 — Structural Relationship between Service-Related ABBs

Figure 7 shows the "lifecycle" or the dynamic relationship between the design and development time Service, Functional, and Technical Components, the bundled Deployment Unit, and the runtime Solution Component.

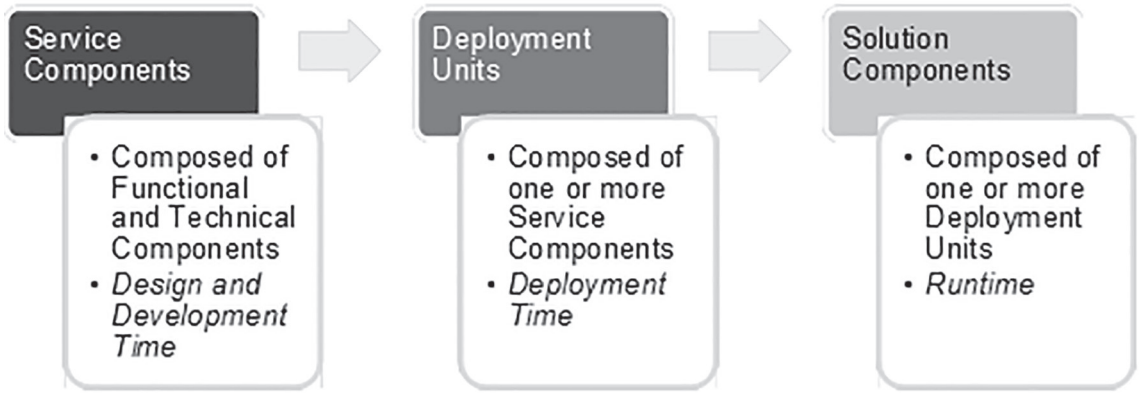


Figure 7 — Dynamic/Temporal Relationship between Service Components, Deployment Units and Solution Components

To understand this in more detail, let us consider an example. In this example, a CheckCredit Service, which is a SOAP doc-style web service, which checks credit by invoking other services – CheckInternalCredit, CheckCreditAcmeRatingAgency1, and CheckCreditAcmeRatingAgency2. CheckInternalCredit is a service which invokes a JEE component which encapsulates a mainframe-based relational database.

Let us consider CheckCredit first. A Service Component is developed which deploys on a .NET environment, whose Technical Component encapsulates the SOAP binding with the .NET stack and integration with the QoS implementation interfaces. The Functional Component for CheckCredit is a Plain Old CLR Object. (POCO), which contains the logic to invoke the services CheckInternalCredit, CheckCreditAcmeRatingAgency1, and CheckCreditAcmeRatingAgency2, as well as to compose a response (the data) which is going to be sent back to the service consumer of CheckCredit. The build process compiles and bundles all the resources and parts of the Service Component together to form the .zip file and .dll component which is then placed in the runtime environment. When placed

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

in the runtime environment (in the appropriate directory), the .dll component becomes the Solution Component for the CheckCredit service.

In practice, SOA typically involves multiple platforms and environments, so CheckInternalCredit will be considered next. This service is a SOAP doc-style web service using JAX-WS as the binding, a Plain Old Java Object (a POJO) to wrap invocations to the database and build (compose) the response. In this case, the Service Component includes the Technical Component which manages the SOAP stack binding and support, and the POJO, database components, and other elements which form the Functional Component. When the build script compiles and puts all the parts of the Service Component together, it creates a Deployment Unit (let us call it a servlet for this discussion), which is then deployed into the appropriate directory to be used at runtime, becoming a Solution Component.

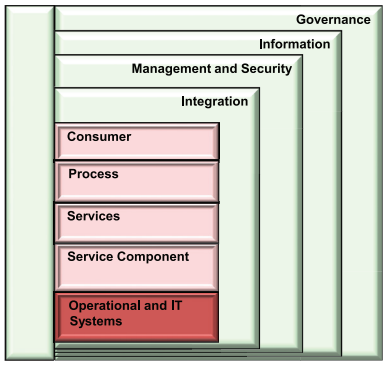
So what is a Solution Building Block? In both services considered in the example above, the service gets deployed into a Solution Building Block which provides the runtime implementation of a Service Container. In the case of the .NET service (CheckCredit), the Solution Building Block is (for the sake of our example) an Internet Information Services (IIS)/Windows Activation Service (WAS) or IIS/WAS and in the case of the JEE service (CheckInternalCredit), the Solution Building Block is (for the sake of our example) an Acme JEE Server. In both cases, the corresponding ABB for the Solution Building Block is the Service Container ABB in the Services Layer.

As can be seen, the Solution Building Block is the runtime instance of an ABB that provides the infrastructure to run services. Solution components, on the other hand, are ABBs which provide the runtime instances of the services themselves.

8 Operational and IT Systems Layer

8.1 Overview

8.1.1 Summary



(From 7.5.2) The Operational and IT Systems layer captures the new and existing organization infrastructure needed to support the SOA solution at design, deploy and run time. This includes the following:

— all infrastructures to run the SOA and its components;

— all operational and runtime hosting of components, both physical and infrastructural;

— all deployment time components;

— all resources to support services, data, and application systems;

— all assets required to support the functionality of the service in the SOA, where the assets may include custom or packaged application assets, new services, services created through composition or orchestration, infrastructure services, etc.

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

This layer provides the building blocks supporting the operational systems which implement the functional capabilities of the other horizontal layers and the supporting/cross-cutting Aspects. In particular, the capabilities supported by this layer include providing operational and runtime hosting, infrastructure services and infrastructure virtualization, functional delivery support including support for service implementations and realizations.

A number of existing software systems are part of this layer. Those systems include, but are not limited to, the following:

— existing monolithic custom applications;

— existing transaction processing systems;

— existing databases;

— existing package applications and solutions including ERP and CRM packages;

— legacy applications and systems;

— access to existing web services;

— systems built from web services;

— IT Infrastructure;

— Enterprise Application Integration (EAI) systems;

— Service Resources providing access to existing business elements or third-party elements;

— Data Resources providing physical storage of data in the business solution;

— Application System Resources providing specific business functions;

— access to existing business services.

This layer represents the intersection point between the actual runtime infrastructure and the rest of the SOA which runs on that infrastructure. In addition, it is the integration point for an underlying infrastructure as a service construct and the rest of the SOA in the wider context of Cloud computing. Key requirements for this layer are outlined in the capability clause describing the capabilities provided to fulfil those requirements.

8.1.2 Context and Typical Flow

All runtime elements of architecture reside in this layer. Effectively, this layer can conceptually be thought of as the runtime or deployment time of the solution. If a thought experiment is conducted which "freezes" the operations within this layer in a frame of time and expands it out, the separation of concerns that tend to cluster building blocks within the architecture will be discovered: the parts most closely connected with the consumption of the services, the processes that are choreographed into a flow, the services whose interfaces are exposed for consumption, the Service Components that will ultimately be used to realize the implementation of the services, along with the other five major cross-cutting and supporting Aspects (development, integration, information, Management and Security (MaS) factors, and governance).

Since this layer represents a point in time and logical categorization and generalization of the runtime environment, then this layer supports all the capabilities of the infrastructure that are needed for running/executing all software. Therefore, this layer supports the execution of the capabilities and responsibilities of the other layers of the SOA RA, including the components implementing the services, i.e. those components that a service relies on for providing it with its functional capabilities.

For example, if a capability for a SOA solution involves systems using mainframe and Java EE (see Reference [24]) platforms, necessary Architecture Building Blocks (ABBs) from the Integration Aspect

Documents
Document Sharing Hub

© ISO/IEC 2016 – All rights reserved

Full document details: BS ISO/IEC 18384-2:2016

and Service Component Layer need to be instantiated using the underlying mainframe and Java EE components which provide the functional capability.

This formula can be expressed as:

Operational and IT Systems Layer = (infrastructural elements of all other layers) + [underlying infrastructure to run the infrastructural elements (i.e. operating systems, etc.)] + (elements that realize the Functional Components of services)

8.1.3 Capabilities

There are multiple categories of capabilities that the Operational and IT Systems Layer needs to support. These categories of capabilities are as follows.

— **Service Delivery**: This category of capabilities is required for delivery of the functional elements of services. This includes the finding of the components implementing the services, the wrapping and the composition/decomposition of the underlying services, and the implementation of the services.

— **Runtime Environment**: This category of capabilities is required for providing a runtime environment representing runtime infrastructure for SOA. This includes capabilities to support both the components required to support service functionality and those required to actually run the components and building blocks of the SOA RA itself. This includes capabilities for the following:

  — the hardware, operating system components;

  — the Solution Building Blocks, which are the runtime instances or realizations of the ABBs of all layers in the SOA RA that have been selected for inclusion in a particular operating environment.

— **Virtualization and Infrastructure Services**: This category of capabilities provides underlying infrastructure such as computing power, network, storage, etc. in native or a virtualized manner.

This layer features the following capabilities in these categories.

— **Service Delivery**

  1) Ability to locate components implementing services
  2) Ability to host applications and functionality to deliver service features
  3) Ability to host databases needed for service implementation
  4) Ability to host legacy systems needed for service implementation
  5) Ability to act as a broker between services requests and invoking implementations
  6) Ability to map service functional requirements to underlying or legacy solution
  7) Ability to compose service function from underlying services and implementation of services
  8) Ability to wrap custom and legacy platforms for service implementation
  9) Ability to find service component associated with Solution Building Blocks
  10) Ability to delegate request or invoke Solution Component for service

— **Runtime Environment**

  11) Ability to support operating systems platforms
  12) Ability to support runtime hosting platforms
  13) Ability to support runtimes of software needed to run service implementation
  14) Ability to support runtimes and software needed to deploy service implementations

© ISO/IEC 2016 – All rights reserved
AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

15) Ability to run supporting ABBs and Solution Building Blocks from other layers of the SOA RA

16) Ability to support the software environment in which the Solution Component runs

— **Virtualization and Infrastructure Services**

17) Ability to provide infrastructure needed by runtime infrastructure

18) Ability to provide infrastructure in a virtualized manner to platforms

19) Ability to provide infrastructure in a virtualized manner to service implementation

20) Ability to manage infrastructure and virtualized infrastructure

21) Ability to provide a single point of control for security of the Operational and IT Systems Layer

These capabilities can be used to group the ABBs in 8.1.4.

8.1.4 Structural Overview of the Layer

The ABBs in the Operational and IT Systems Layer are logically partitioned into the categories that support the following:

— Solution Components which provide the functional capability of services and the solution;

— the runtime environment required by the SOA solution and that runs the actual Solution Components and their supporting infrastructure;

— service delivery which provides interfaces with underlying infrastructure components, such that they can be virtualized and effectively leveraged by the architecture.

In the diagrams that are used throughout this part of ISO/IEC 18384 to provide a structural overview of the layers of the SOA RA, the ABBs have been colour-coded to match the architecture layers in the to which they belong and a prefix has been added to the name of the ABB for additional clarity. White indicates ABBs that are defined in this layer. ABBs owned by other layers that are used to support the capabilities of the current layer are shown in darker shades of grey which match the colours of the layers in the SOA RA layers diagram as shown in Figure 3. Each ABB includes one or more numbers in the box which indicate which capabilities in the list in 8.1.3 that the ABB supports. For example, in Figure 8, the ABBs from the Management and Security Aspect are a very dark grey (with a prefix of 'MaS:') while the ABB from the Integration Aspect is shown as black (with a prefix of "Integration"). Therefore, 'MaS: Policy Enforcer 'and it supports capability number '2: Ability to host applications and functionality to deliver service features' and 'Integration: Integration Controller' supports '5: Ability to act as a broker between service requests and invoking implementations.

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

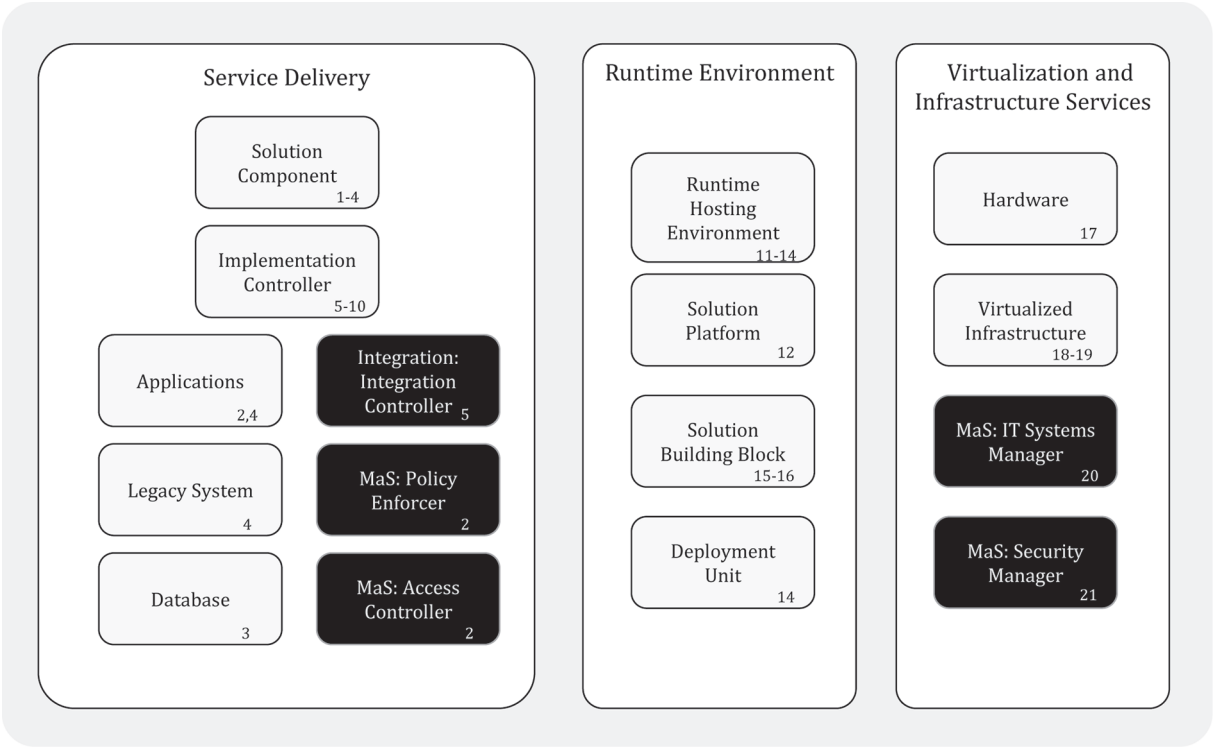


Figure 8 — ABBs in the Operational and IT Systems Layer

The details of the ABBs in 8.2 are grouped by the capabilities they support with ABBs from other layers listed last.

8.2 Details of ABBs and Supported Capabilities

8.2.1 Service Delivery

8.2.1.1 Solution Component

This ABB represents realization of systems that represent logical groupings of functionality and associated functionally cohesive services. Its bill of material contains the Service Component, Functional Components, and Technical Components from the Service Component Layer realizing services that provide well-defined interfaces for the systems. For example, it could be an invocation of a legacy component, or an existing or new database request, application, or a component wrapped within a Commercial Off-The-Shelf (COTS) package. Thus, it is a runtime instantiation of a group of cohesive components residing within a system that collectively provide an implementation for a set of related services.

8.2.1.2 Implementation Controller (IC)

This ABB represents elements that receive a request to invoke an underlying Solution Component and delegates it to the appropriate Solution Component. It also incorporates logic for composition and decomposition of legacy applications into Solution Component. This is required because historically, most legacy applications have not been written with the intent of being elements in an SOA and the service Solution Components within them need to be exposed through service composition and decomposition.

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

8.2.1.3 Applications (Packaged and Custom)

This ABB represents the applications and operations performed on them that are running as units of execution within the runtime environment of an SOA solution, for example, existing human resources systems, SAS systems, and other domain specific systems.

8.2.1.4 Legacy system

This ABB represents the legacy systems and operations performed on them in an SOA solution.

8.2.1.5 Database

This ABB represents the databases and operations performed on them in an SOA solution.

8.2.1.6 Integration Aspect: Integration Controller

This ABB represents elements responsible for coordinating and brokering or mediating the interactions between the applications, database, security, etc. that need to work in concert to effectively provide a runtime experience.

8.2.1.7 Management and Security Aspect: Policy Enforcer

See 14.2.7.1.

8.2.1.8 Management and Security Aspect: Access Controller

See 14.2.2.8.

8.2.2 Runtime Environment

8.2.2.1 Runtime Hosting Environment (RHE)

This ABB represents elements that provide support for operational and runtime services. These include software services such as the operating system instance on which the Solution Platforms run, as well as underlying infrastructural services such as hardware support, memory, storage, networks, etc.

8.2.2.2 Solution Platform

This ABB represents elements that support the software environment in which the solution components and solution building blocks deploy and run. Examples would be Java virtual machines (JVMs) hosting a service container solution building block or a CICS environment.

8.2.2.3 Solution building block

This ABB represents the runtime component of ABBs from other layers in the SOA RA. Thus, for example, a Mediation ABB from the Integration Aspect runs as a Solution Building Block.

8.2.2.4 Deployment Unit

This ABB represents an executable application that can be deployed as a single unit (e.g. exe, war, ear, etc.) in the target hosting environment. The instantiation of this ABB is deployed on Solution Platforms.

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

8.2.3 Virtualization and Infrastructure Services

8.2.3.1 Hardware

This ABB represents an abstraction of the physical hardware that is the platform on which Deployment Units are actually deployed and are executing (running).

8.2.3.2 Virtualized Infrastructure

This ABB represents elements that support the utilization of infrastructure in a virtualized manner by the operational and hosting runtime environment. Thus, the use of shared disk space in a Cloud environment would be an example.

8.2.3.3 Management and Security Aspect: IT Systems Manager

See 14.2.3.1.

8.2.3.4 Management and Security Aspect: Security Manager

See 14.2.2.1.

8.3 Inter-Relationships between the ABBs

The Solution Component ABB represents a key ABB interacting with other ABBs. Since it contains the Service Component, Functional Components, and Technical Components from the Service Component Layer for realizing services, it interacts with the implementation controller and other ABBs as needed. Requests are first validated as secure by the Access Controller ABB and Security Manager ABB in the Management and Security Aspect, and then translated into Solution Component ABB requests by the Implementation Controller ABB and executed by the Solution Components in the Solution Platform.

The runtime hosting and operational environment capability is supported by the Solution Platform ABB and Runtime Hosting Environment ABB. Thus, both ABBs from all layers of the SOA RA run as Solution Building Blocks on the Solution Platform hosted by the Runtime Hosting Environments.

The infrastructure services and infrastructure virtualization capability, basically, is responsible for exposing the underlying infrastructure in an on-demand manner, encapsulating the invoking Runtime Hosting Environment and enabling rapid scaling.

In Figure 9, the arrows between the ABBs indicate an interaction from one ABB to another.

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

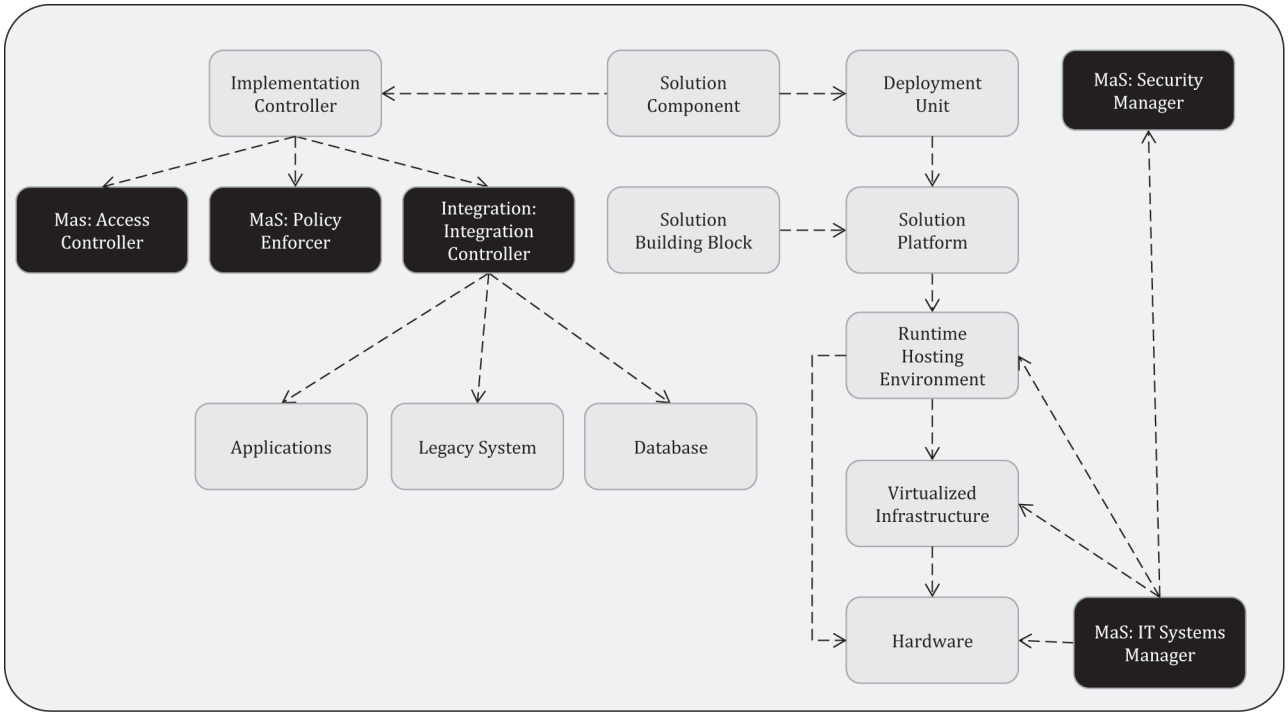


Figure 9 — Relationships among ABBs in the Operational and IT Systems Layer

8.4 Significant Intersection Points with other Layers

8.4.1 General

There is a significant intersection between the Operational and IT Systems Layer and all other layers of the SOA RA as this layer provides the actual runtime environment to the other layers to execute at runtime. It intersects with the rest of the SOA RA including, of course, the cross-cutting aspects, such as the Management and Security Aspect which acts as an aggregation point for monitoring, managing, and securing the runtime environment.

This connection is summarized in 8.3 in terms of intersection with the rest of the SOA RA including cross-cutting aspects.

8.4.2 Intersection with the rest of the SOA RA

The following are the two points of intersection of the Operational and IT Systems Layer with the rest of the SOA RA:

a) The first perspective is that the Operational and IT Systems Layer supports the functionality of the SOA RA rendered by the more abstract layers above. There are two parts to this perspective: providing a wrapper which exposes the service from the Service Component Layer, and providing a mapping to the underlying Solution Component ABB which actually supports the capability. The Solution Platform ABB provides a platform to deploy and run the Solution Components realizing the functionally cohesive services in a system;

b) The second perspective is that the Operational and IT Systems Layer provides the runtime environment for the ABB from other layers. These ABBs from the other layers are instantiated as Solution Building Blocks for the runtime environment. The Solution Platform ABB provides a platform to deploy and run these ABBs from other layers.

© ISO/IEC 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016

8.4.3 Interaction with Cross-Cutting Aspects

The Operational and IT Systems Layer relies on cross-cutting Aspects of the architecture to fulfil its responsibilities. These interactions are based on common scenarios and best practices.

It relies on the Development Aspect for the following capabilities:

— ability to implement and test resources and connectors to Service Components with tools;

— ability to process deployment descriptions and hosting environment descriptions in order to deploy supporting resources into the Runtime Hosting Environment;

— ability to perform performance load testing and simulation to optimize resource usage.

It relies on the Management and Security Aspect for the following capabilities:

— ability to authenticate/authorize for service invocation;

— ability to enforce operational policies;

— ability to monitor health and well-being of underlying infrastructure and solution and applications deployed on the infrastructure.

It relies on the Information Aspect for the following capabilities:

— ability to store and retrieve metadata and data.

It relies on the Integration Aspect for the following capabilities:

— ability to invoke business processes and/or services;

— ability to transform data from one format to another.

It relies on the Governance Aspect for the following capabilities:

— ability to set and store business rules;

— ability to manage policies for IT management;

— ability to manage security policies.

© ISO/IEC 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18384-2:2016