TECHNICAL REPORT

ISO/IEC TR 30102

First edition
2012-12-01

Information technology — Distributed Application Platforms and Services (DAPS) — General technical principles of Service Oriented Architecture

Technologie de l'information — Plate-formes et services d'applications distribuées (DAPS) — Principes techniques généraux de l'architecture orientée services

Reference number
ISO/IEC TR 30102:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide to publish a Technical Report. A Technical Report is entirely informative in nature and shall be subject to review every five years in the same manner as an International Standard.

Attention is drawn to the possibility that some of the elements of this Technical Report may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 30102 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Subcommittee SC 38, *Distributed application platforms and services (DAPS)*.

Introduction

Service Oriented Architecture (abbreviated SOA) is an architectural style that supports service orientation and is a paradigm for business and IT (see 3.1.40). 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 repeatable business activity that has specified outcomes, is self contained, may be composed of other services and is a “black box” to consumers of the service (see 3.1.14).

To enable this co-operation and collaboration 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 ever-changing business needs. Many companies across many industries world-wide have developed SOA enterprise architectures, solutions and products.

This report 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, improve the standardization and quality of solutions, as well as promote effective large-scale adoption of SOA. The benefits of this technical report contribute to improving the standardization, interoperability, and quality of solutions supporting SOA.

This technical report defines the basic technical principles and reference architecture for SOA rather than being focused on the business aspects. It also discusses the functional, performance, development, deployment, and governance aspects of SOA. This technical report 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.

This technical report includes the following clauses:

Clause 3 – terminology – defines terms used when discussing or designing service oriented solutions. Terms defined here are used in some unique fashion for SOA. It does not define terms that are used in general English manner.

Clause 4 – Concepts and Principles – articulates basic SOA concepts and expands on the key terms in clause 3.

Clause 5 – SOA Technical Framework – documents an overview of a reference architecture for building SOA based solutions.

The targeted audience of this technical report includes, but is not limited to, standards organizations, architects, SOA service providers, SOA solution and service developers, and SOA service consumers who are interested in adopting and developing SOA.

Information technology — Distributed Application Platforms and Services (DAPS) — General technical principles of Service Oriented Architecture

1 Scope

This Technical Report describes the general technical principles underlying Service Oriented Architecture (SOA), including principles relating to functional design, performance, development, deployment and management. It provides a vocabulary containing definitions of terms relevant to SOA.

It includes a domain-independent technical framework, addressing functional requirements and non-functional requirements.

2 Terms and definitions

For the purposes of this document, the following terms and definitions apply

2.1 Definitions

2.1.1
actor
person or system component who interacts with the system as a whole and who provides stimulus which invoke actions

NOTE See ISO/IEC 16500-8:1999, 3.1.

2.1.2
architecture
fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution ISO/IEC/IEEE 42010:2011, 3.2). ISO/IEC 40210:2011

2.1.3
choreography
omposition whose elements interact in a non-directed fashion with each autonomous member knowing and following an observable predefined pattern of behavior for the entire (global) composition

NOTE See Bibliography Reference [21].

2.1.4
collaboration
omposition whose elements interact in a non-directed fashion, each according to their own plans and purposes without a predefined pattern of behavior

NOTE See Bibliography Reference [21].

2.1.5
composition
result of assembling a collection of things for a particular purpose

NOTE See Bibliography Reference [21].

2.1.6
effect
outcome of an interaction with a service

NOTE 1 If service contracts exist, they usually define effects. The effect is how a service, through the element that performs it, delivers value to its consumer.

NOTE 2 See Bibliography Reference [21].

2.1.7
element
unit that is indivisible at a given level of abstraction,and has a clearly defined boundary

NOTE 1 An Element can be any type of entity.

NOTE 2 See Bibliography Reference [21].

2.1.8
entity
individual in a service system with an identity which can act as a service provider or consumer

NOTE Examples of entities are organizations, enterprises and individuals, software and hardware.

2.1.9
event
something that occurs,to which an element may choose to respond

NOTE Events can be responded to by any element and events may be generated (emitted) by any element.

2.1.10
execution context

set of technical and business elements that form a path between those with needs and those with capabilities and that permit service providers and consumers to interact

NOTE 1 The execution context of a service interaction is the set of infrastructure elements, process entities, policy assertions and agreements that are identified as part of an instantiated service interaction, and thus forms a path between those with needs and those with capabilities.

NOTE 2 See Bibliography Reference [19].

2.1.11
human actor
person or an organizational entity

NOTE 1 In principle, this classification is not exhaustive.

NOTE 2 See Bibliography Reference [21].

2.1.12
human tasks
tasks which are done by people or organizations, specifically instances of Human Actor

2.1.13
information Type
type of information given or received in a service interface

2.1.14
orchestration
composition for which there is one particular element used by the composition that oversees and directs the other elements

NOTE 1 The element that directs an orchestration by definition is different than the orchestration (Composition instance) itself.

NOTE 2 See Bibliography Reference [21].

2.1.15
process
composition whose elements are composed into a sequence or flow of activities and interactions with the objective of carrying out certain work

NOTE 1 A process may also be a collaboration, choreography, or orchestration.

NOTE 2 See Bibliography Reference [21].

2.1.16
REST
architectural style for distributed hypermedia systems. REST provides a set of architectural constraints that, when applied as a whole, emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems

NOTE See REST "Fielding, Roy Thomas (2000), Architectural Styles and the Design of Network-based Software Architectures, Doctoral dissertation, University of California, Irvine).

2.1.17
service
logical representation of a set of repeatable activities that has specified outcomes, is self-contained, may be composed of other services, and is a “black box” to consumers of the service

NOTE 1 See Bibliography Reference [21].

NOTE 2 The word “activity” in the ‘Service’ definition above is used in the general English language sense of the word, not in the process-specific sense of that same word (i.e., activities are not necessarily process activities).

2.1.18
service broker
implements service intermediaries that provide unified service registration and publishing

NOTE They can also provide other important supports for SOA, such as service discovery, routing, location-transparent service access, for service providers and service consumers.

2.1.19
service bus
intermediary IT infrastructure that supports service access and consumption, event-driven message routing among services

NOTE The core functionalities of Service Bus might include: service routing, message transformation, event handling, providing service call, and related intermediary services, connecting a variety of applications, services, information, and platform resources. Service bus is widely used in enterprise contexts and usually equates to the Enterprise Service Bus (ESB).

2.1.20
service catalogue
service registry
service repository
component that supports publication, registration, search, and retrieval of metadata and artifacts for services

NOTE A service registry is typically a limited set of metadata to facilitate interaction with services and accessing content from a service repository containing the full artifacts.

2.1.21
service choreography
composition whose elements are services that interact in a non-directed fashion with each autonomous member knowing and following an observable predefined pattern of behavior for the entire (global) composition

NOTE See Bibliography Reference [21].

2.1.22
service collaboration
composition whose elements are services that interact in a non-directed fashion, each according to their own plans and purposes without a predefined pattern of behavior

NOTE See Bibliography Reference [21].

2.1.23
service component
element that implements services

2.1.24
service composition
service assembly
result of assembling a collection of services to achieve a particular purpose

NOTE A composition can support different composition patterns: such as. collaboration, choreography, orchestration.

NOTE See Bibliography Reference [21].

2.1.25
service consumer
entity that uses services

NOTE Consumers may interact with services operationally or with contractually (legal responsibility).

2.1.26
service contract
terms, conditions, and interaction rules that interacting consumers and providers must agree to (directly or indirectly)

NOTE 1 A service contract is binding on all participants in the interaction, including the service itself and the element that provides it for the particular interaction in question.

NOTE 2 See Bibliography Reference [21].

2.1.27
service description
information needed in order to use, or consider using, a service

NOTE 1 The service description usually includes the service interfaces, contracts, and policies

NOTE 2 See Bibliography Reference [19].

2.1.28
service deployment
process that makes implementations of services deployed and able to actually run in a specific hardware and software environment

NOTE 1 Service deployment includes static deployment and dynamic deployment. Static deployment means that the call relations between the services is defined before runtime. Dynamic deployment means that when the application system is running, it needs to determine the call relations by dynamic routing.

NOTE 2 In terms of a single service, a service after deployment can be actually called by end users, other IT systems or services. In terms of multiple service-based processes, after deployment, they can form a complete application system to provide appropriate IT support for users.

2.1.29
service development
service implementation
technical development and physical implementation of the service that is part of a service lifecycle

2.1.30
service discovery
process that service consumers use to search and retrieve desired services according to their specific functional or non-functional requirements

2.1.31
service governance
strategy and control mechanism definition on service lifecycle, which includes establishment of chains of responsibility, ensures its compliance with policies by providing appropriate processes and measurements, addressing service modifications, version updates, notice of termination, decomposition subdivision, agency capacity, decomposition capacity, ability to meet individual demands

2.1.32
service interaction
activity involved in making using of a capability offered, usually across an ownership boundary, in order to achieve a particular desired real-world effect

NOTE See Bibliography Reference [19].

2.1.33
service interface
way in which other elements can interact and exchange information with a service as the outcome of a request in the definition of a service

NOTE See Bibliography Reference [21].

2.1.34
service interoperability
ability of providers and consumers to communicate, invoke services and exchange information at both the syntactic and semantic level

2.1.35
Service Level Agreement (SLA)
service contract that defines the interaction and measureable conditions of interaction between a service provider and a service consumer

NOTE A Service Level Agreement usually contains: the set of services the provider will deliver, a complete, specific definition of each service, the responsibilities of the provider and the consumer, the set of metrics to determine whether the provider is delivering the service as promised, an auditing mechanism to monitor the service, the remedies available to the consumer and provider if the terms of the SLA are not met, and how the SLA will change over time.

2.1.36
service lifecycle
set of phases for a service throughout its life, from identification to instantiation and retirement

2.1.37
service management
monitoring, controlling, maintaining, and operating services

2.1.38
service modeling
service oriented analysis process of finding and modelling a series of service candidates for functions or actions which can be defined independently or by decomposing business processes

2.1.39
service monitor
monitoring and controlling operational state and performance of service

2.1.40
service orchestration
composition of services for which there is one particular element used by the composition that oversees and directs the other elements

NOTE 1 the element that directs an orchestration by definition is different than the orchestration (Composition instance) itself.

NOTE 2 See Bibliography Reference [21].

2.1.41
service orientation
approach to designing systems in terms of services and service-based development

2.1.42
service oriented analysis
preparatory information gathering steps that are completed in support of a service modeling sub-process that results in the creation of a set of service candidates

NOTE Service Oriented Analysis is the first phase in the cycle, though the service-oriented analysis process might be carried out iteratively, once for each business process. It provides guidance to the following process.

2.1.43
service oriented architecture
architectural style that supports service orientation and is a paradigm for building business solutions using IT

NOTE 1 Services realized in this style utilize activities that comprise business processes, have descriptions to provide context may be implemented via service composition, have environment-specific implementations which are described in the context that constrains or enables them, require strong governance, and place requirements on the infrastructure to achieve interoperability and location transparency using standards to the greatest extent possible.

NOTE 2 See Bibliography Reference [21].

2.1.44
SOA governance
extension of IT governance specifically focused on management strategies and mechanisms for the end users' specific SOA solution

NOTE 1 It manages the entire SOA lifecycle by setting out personnel, roles, management procedures and decision-making. SOA governance needs to adopt the appropriate methodology and best practices. SOA governance usually requires tools for assistance to customize and manage the governance strategy according to the needs.

NOTE 2 While management means the specific process for governance and control to execute the policies, governance looks at assigning the rights to make decisions, and deciding what measures to use and what policies to follow to make those decisions.

2.1.45
SOA implementation
process methods and techniques used to develop SOA based solutions

2.1.46
SOA lifecycle
process for engineering SOA-based solutions, including analysis, design, implementation, deployment, test and management

2.1.47
SOA management
measurement, monitoring, and configuration of the entire lifecycle of a SOA solution

NOTE At runtime it is the process for the specific measurement and operation of the implementation of the SOA solution according to the strategies and mechanisms identified by the SOA governance process.

2.1.48
SOA maturity
quantitative description of the level of SOA application adoption within an IT architecture in an organization

2.1.49
SOA maturity model
framework and method to evaluate SOA maturity

2.1.50
service policy
statement that an entity may intend to follow or may intend that another entity should follow

NOTE See Bibliography Reference [21].

2.1.51
service provider
entity providing services

NOTE Providers may be responsible for the operation of the services or the contract for the service (legal responsibility).

2.1.52
service publishing
publishing information for registered services

2.1.53
SOA resource
elements that provide the IT resources used by services

2.1.54
SOA solution
solutions implemented by applying SOA concepts, methods, and techniques

2.1.55
SOAP
stateless, one-way message exchange paradigm, but applications can create more complex interaction patterns (e.g., request/response, request/multiple responses, etc.) by combining such one-way exchanges with features provided by an underlying protocol and/or application-specific information

NOTe See SOAP - SOAP Version 1.2 Part 0: Primer http://www.w3.org/TR/soap12-part0/

2.1.56
task
atomic action which accomplishes a defined result

NOTE See Bibliography Reference [21].

2.1.57
Web Services
software system designed to support interoperable machine-to-machine interaction over a network

NOTE It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. (See Web Services Architecture http://www.w3.org/TR/ws-arch/).

2.2 Acronyms

ABB - Architectural Building Block

BMM – Business Motivation Model (see OMG)

BPMN – Business Process Management Notation

IT – Information Technology

EA – Enterprise Architecture

RA – Reference Architecture

SLA – Service Level Agreement

SOA - Service Oriented Architecture

SOSE – Service Oriented Software Engineering

SQL – Structured Query Language

WSDL – Web Services Description Language

WSRP – Web Services Remote Portlet

KPI – Key Performance Indicator

3 SOA Principles and Concepts

3.1 Introduction to SOA

Service Oriented Architecture (abbreviated SOA) is an architectural style that supports service orientation and is a paradigm for business and IT (see 3.1.40). 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 repeatable business activity that has specified outcomes, is self-contained, may be composed of other services and is a “black box” to consumers of the service. (see 3.1.14).

As a foundation for understanding, SOA is an architectural style that has the following distinguishing characteristics:

1. It is based on the design of the services and processes – which mirror real-world business activities – comprising the enterprise (or inter-enterprise) business processes.

2. Service representation utilizes business descriptions to provide context (i.e., business process, goal, rule, policy, service interface, and service component) and implementations of services are provided use processes and service composition.

3. It places unique requirements on the infrastructure – it is recommended that implementations use open standards to realize interoperability and location transparency.

4. Implementations are environment-specific – they are constrained or enabled by context and must be described within that context.

5. It requires strong governance of service representation and implementation.

6. It requires a criteria to determine what a “good service”. (see [20])

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.

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 ever-changing business needs.

In recent years, SOA has become a business organization and technology hot spot that is recognized and respected in industry. Many companies have developed SOA enterprise architecture, solutions and products world-wide. At the same time, an increasing number of solutions are being implemented using SOA in many different industries.

However, 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, as well as promote effective large-scale adoption of SOA, it is necessary to establish a unified set of general technical principles for SOA.

It should be noted that these SOA principles defined here 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).

The engineering of SOA based systems and solutions, 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).

3.2 Concepts

3.2.1 Roles

Providers

A service provider is an entity providing services. Providers can be responsible for providing services in two different ways:

- Operationally - the provider is responsible for responding to the exchange of messages with the consumer as well as producing the promised effect of invoking the service. Assuming the operational responsibility for providing a service implies the following across the lifecycle of said service:

    - Service Creation: Creating a service implementation that can provide the service in question
    - Providing Services: Providing the implemented service for use by others