Process and Service Design

Professor: Pierluigi Plebani

Author: Simone Staffa

Politecnico di Milano
Computer Science & Engineering — Fall 2019-2020

Contents

1 Introduction

The meaning of term **service** is intuitive but is definition can be different in different contexts like:

- Economic
- ICT

We firstly need to understand similarities and differences among these different terms:

- Service
- e-Service
- Web-Service

Definition of service (economic perspective):
"A service is a change in the condition of a person, or a good belonging to some economic unit, which is brought about as the result of the activity of some other economic unit, with the prior agreement of the former person or economic unit," (P. Hill, On goods and services)

Figure 1: Looking for a dress, in different eras

The above picture represents the evolution of the services along the years. Specially, it refers to the dress service. In the Pre Industrial era was more like a personal service, where a tailors offered totally customized dresses to people directly in their houses. Then with the Industrial Era, dresses were sold by retailer, so the final product was not really customized by the consumers, who just had to select the already made dress they liked. Finally, nowadays in the Information Era, service started to be offered through IT Platforms, with full customization without even

the need to go to the shop, dresses can be just bought and received at home.
This can be also applied to the IT domain, for what we call nowadays Web Services.
The Pre Industrial Era corresponds to when people used to build their own computers, buying different parts and installing them by their selves. The Industrial Era corresponds to when companies started building personal computers to be sold to consumers, who just had to pay and bring them home. Finally, the Information Era corresponds to nowadays Cloud Platform and Web Services that allow people to rent virtual computers online, to perform their own computation.

Just to recap:

- **Personal service (Pre Industrial era)**: highly and man-made customization but limited production
- **Industrial service (Industrial era)**: mass production but no customization
- **Electronic service (Information era)**: highly and computer-based customization combined with mass production

This to say that we are **evolving to a service-oriented society**.
"We don't need a drill, we need a hole in the wall."

From an economic perspective, we are switching from a Good-dominant logic to a Service-dominant logic.

- **Good-dominant logic** focuses on goods exchange, that's why goods are considered to be tangible and each of them has an embedded value that is taken into account to operate exchange transactions.
- **Service-dominant logic** focuses on service provision, where resources are intangible and require a co-creation of value.

S-D logic is useful because it opens to more interactions with customers and allows to focus on why a product fits to the customer needs instead of technical specification.

When technology comes into play, there are two different perspectives of ICT in S-D logic, which go under the name of e-Service.

- As the improvement of a programming paradigm
  - **ICT is a goal** and it is offered itself as a service
  - SOA (Service Oriented Architecture) becomes the new way of developing software to be service-ready
- As the automation of economics activities and self-service
  - **ICT is a mean** and can improve the effectiveness and the efficacy
  - The goal is to make the business able to provide services

As a mean, ICT has a fundamental role in the evolution towards the S-D logic. Who deliver products must pay a lot of attention on Customization, Scalability, Reliability and Security/Privacy. As a goal, ICT is also a service to be delivered and a possible way to deliver these service could be using a Web Service.

2 Service Definition

Definition of service (IT perspective):
"A service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. A service is provided by an entity – the service provider – for use by others, but the eventual consumers of the service may not be known to the service provider and may demonstrate uses of the service beyond the scope originally conceived by the provider." (P. Hill, On goods and services)

2.1 Software Oriented Architecture

Service-Oriented Architecture (SOA) is a style of software design where services are provided to the other components by application components, through a communication protocol over a network. In service oriented architecture, a number of services communicate with each other, in one of two ways: through passing data or through two or more services coordinating an activity.

2.1.1 Reference Model

SOA in the end is a reference model. A reference model is an abstract **framework** for understanding significant relationships among the entities of some environment. It enables the development of specific reference or concrete architectures using consistent standards or specifications supporting that environment. A reference model consists of a minimal set of unifying concepts, axioms and relationships withing a particular problem domain, and is independent of specific standards, technologies, implementations, or the concrete details.

2.1.2 SOA Reference Model

The goal of this reference model is to define the essence of service oriented architecture, and emerge with a vocabulary and a common understanding of SOA.

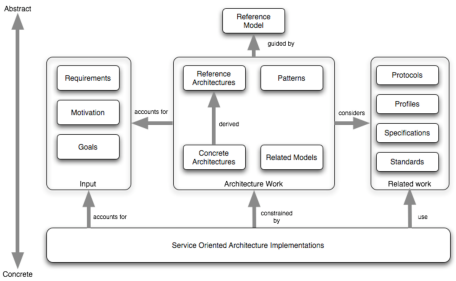

Figure 2: How the Referenced Model relates to other work

Figure 2 shows how a reference model for SOA relates to other distributed systems architectural inputs. The concepts and relationships defined by the reference model are intended to be the basis for describing references architectures and patterns that will define more specific categories of SOA designs. Concrete architectures arise from a combination of reference architectures, architectural patterns and additional requirements, including those imposed by technology environments.

2.1.3 What is Service Oriented Architecture

Service Oriented Architecture is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.
In general, entities (people and organizations) create capabilities to solve or support a solution for the problems they face in the course of their business. It is natural to think of one person's needs being met by capabilities offered by someone else; or, in the world of distributed computing, one computer agent's requirements being met by a computer agent belonging to a different owner.
The perceived value of SOA is that it provides a powerful framework for matching needs and capabilities and for combining capabilities to address those needs.

Visibility, interaction, and effect are key concepts for describing the SOA paradigm. **Visibility** refers to the capacity for those with needs and those with capabilities to be able to see each other. This is typically done by providing descriptions.
Interaction is the activity of using a capability. Typically mediated by the exchange of messages, an interaction proceeds through a series of information exchanges and invoked actions. This permits service providers and consumers to interact and provides a decision point for any policies and contracts that may be in force.

The purpose of using a capability is to realize one or more **real world effects**. At its core, an interaction is "an act" as opposed to "an object" and the result of an interaction is an effect. This effect may be the return of information or the change in the state of entities (known or unknown) that are involved in the interaction.

We are careful to distinguish between *public* actions and *private* actions; private actions are inherently unknowable by other parties. On the other hand, public actions result in changes to the *state* that is shared between at least those involved in the current execution context and possibly shared by others. Real world effects are, then, couched in terms of changes to this **shared state**.

This description of SOA has yet to mention what is usually considered the central concept: the **service**. The noun "service" is defined in dictionaries as "The performance of work (a function) by one for another".
While both needs and capabilities exist independently of SOA, **in SOA services are the mechanism by which needs and capabilities are brought together.**

SOA is a means of organizing solutions that promotes reuse, growth and interoperability. It is an organizing and delivery paradigm that enables one to get more value from use both of capabilities which are locally "owned" and those under the control of others.

The concepts of visibility, interaction, and effect apply directly to services in the same manner as these were described for the general SOA paradigm. Visibility is promoted through the service description which contains the information necessary to interact with the service and describes this in such terms as the service inputs, outputs, and associated semantics. The service description also conveys what is accomplished when the service is invoked and the conditions for using the service.

In general, entities (people and organizations) offer capabilities and act as service providers. Those with needs who make use of services are referred to as service consumers.

2.1.4 Benefits of SOA

The value of SOA is that it provides a simple scalable paradigm for organizing large networks of systems that require interoperability to realize the value inherent in the individual components. Indeed, SOA is scalable because it makes the fewest possible assumptions about the network and also minimizes any trust assumptions that are often implicitly made in smaller scale systems.

Through this inherent ability to scale and evolve, SOA enables an IT portfolio which is also adaptable to the varied needs of a specific problem domain or process architecture. The infrastructure SOA encourages is also more agile and responsive than one built on an exponential number of pair-wise interfaces. Therefore, SOA can also provide a solid foundation for business agility and adaptability.

2.2 The Reference Model

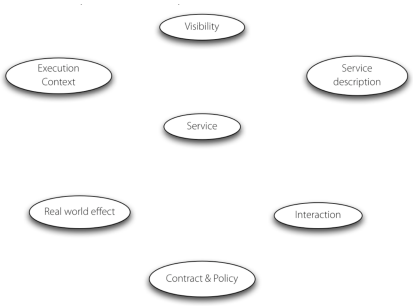

Figure 3: Principal concepts in the Reference Model

2.2.1 Service

A service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. A service is provided by an entity – the **service provider** – for use by others, but the eventual consumers of the service may not be known to the service provider and may demonstrate uses of the service beyond the scope originally conceived by the provider. A service is accessed by means of a service interface, where the interface comprises the specifics of how to access the underlying capabilities.

A service is opaque in that its implementation is typically hidden from the service consumer except for (1) the information and behavior models exposed through the service interface and (2) the information required by service consumers to determine whether a given service is appropriate for their needs.

The consequence of invoking a service is a realization of one or more real world effects. These effects may include:

1. Information returned in response to a request for that information
2. A change to the shared state of defined entities
3. A combination of (1) and (2)

2.2.2 Dynamics of Services

From a dynamic perspective, there are three fundamental concepts that are important in understanding what is involved in interacting with services: the visibility between service providers and consumers, the interaction between them, and the real world effect of interacting with a service.

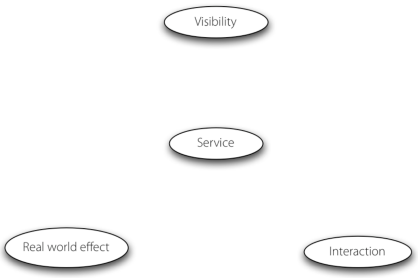

Figure 4: Concepts around the dynamics of a service

Visibility

For a service provider and consumer to interact with each other they have to be able to "see" each other. This is true for any consumer/provider relationship. In the case of SOA, visibility needs to be emphasized because it is not necessarily obvious how service participants can see each other.
Preconditions to visibility are awareness, willingness and reachability. The initiator MUST be aware of the other parties, the participants MUST be predisposed to interaction, and the participants must be able to interact.

- **Awareness**: both the service provider and the service consumer MSUT have information that would lead them to know of the other's existence. Awareness of service offerings is often effected by various *discovery* mechanisms. For a service consumer to discover a a service, the service provider must be capable of making details (service description and policies) of the service available to potential consumers.
- **Willingness**: t is an intentional act to initiate and to participate in a service interaction. For example, if a service consumer discovers a service via its description in a registry, and the consumer initiates an interaction, if the service provider does not cooperate then there can be no interaction. The extent of a service participant's willingness to engage in service interactions may be the subject of policies.
- **Reachability**: the relationship between service participants where they are able to interact; possibly by exchanging information. This is an essential pre-requisite for service interaction - participants MUST be able to communicate with each other. A service consumer may have the intention of interacting with a service, however, if the service is not reachable, the service would not be visible by the consumer.

Interaction

Interacting with a service involves performing actions against the service. In many cases, this is accomplished by sending and receiving messages, but there are other modes possible that do not involve explicit message transmission. For example, a service interaction may be effected by modifying the state of a shared resource. However, for simplicity, we often refer to message exchange as the primary mode of interaction with a service.

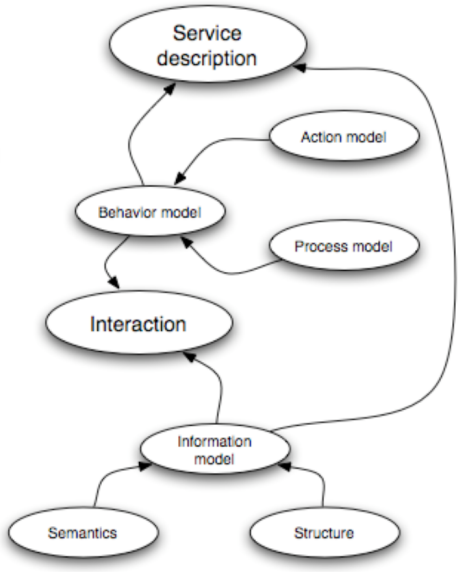

Figure 5: Service Interaction concepts

Figure 5 illustrates the key concepts that are important in understanding what it is involved in interacting with services; these revolve around the service description – which references a information model and a behavior model.

- **Information model**: a characterization of the information that may be exchanged with the service. Only information and data that are potentially exchanged with a service are generally included within that service's information model. The scope of the information model includes the format of information that is exchanged, the structural relationships within the exchanged information and also the definition of terms used. Knowing the representation, structure, and form of information required is a key initial step in ensuring effective interactions with a service. There are several levels of such structural information; including the encoding of character data, the format of the data and the structural data types associated with elements of the information.

- **Behavior model**: knowledge of the actions invoked against the service and the process or temporal aspects of interacting with the service. This is characterized as knowledge of the actions on, responses to, and temporal dependencies between actions on the service. For example, in a security-controlled access to a database, the actions available to a service consumer include presenting credentials, requesting database updates and reading results of queries. The action model of a service is the characterization of the actions that may be invoked against the service. The process model characterizes the temporal relationships

and temporal properties of actions and events associated with interacting with the service.

Real World Effect

There is always a particular purpose associated with interacting with a service. Conversely, a service provider (and consumer) often has a priori conditions that apply to its interactions. The service consumer is trying to achieve some result by using the service, as is the service provider. At first sight, such a goal can often be expressed as "trying to get the service to do something". This is sometimes known as the "real world effect" of using a service. For example, an airline reservation service can be used to learn about available flights, seating and ultimately to book travel – the desired real world effect being information and a seat on the right flight.
As previously stated, a real world effect can be the response to a request for information or the change in the state of some defined entities shared by the service participants. In this context, the shared state does not necessarily refer to specific state variables being saved in physical storage but rather represents shared information about the affected entities.

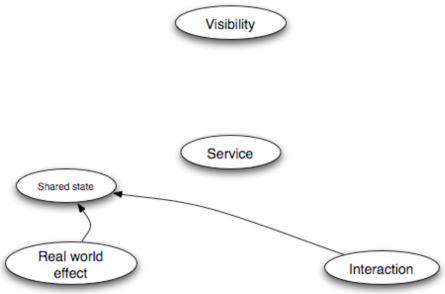

Figure 6: Real World Effect and shared state

In addition, the internal actions that service providers and consumers perform as a result of participation in service interactions are, by definition, private and fundamentally unknowable. By unknowable we mean both that external parties cannot see others' private actions and, furthermore, SHOULD NOT have explicit knowledge of them. Instead we focus on the set of facts shared by the parties – the shared state. Actions by service providers and consumers lead to modifications of this shared state; and the real world effect of a service interaction is the accumulation of the changes in the shared state.

For example, when an airline has confirmed a seat for a passenger on a flight this represents a fact that both the airline and the passenger share – it is part of their shared state. Thus the real world effect of booking the flight is the modification of this shared state – the creation of the fact of the booking. Flowing from the shared facts, the passenger, the airline, and interested third parties may make inferences – for example, when the passenger arrives at the airport the airline confirms the booking and permits the passenger onto the airplane (subject of course to the passenger meeting the other requirements for traveling).
For the airline to know that the seat is confirmed it will likely require some private action to record the reservation. However, a passenger should not have to know the details of the airline internal procedures. Likewise, the airline does not know if the reservation was made by the passenger or someone acting on the passenger's behalf. The passenger's and the airline's understanding of the reservation is independent of how the airline maintains its records or who initiated the action.

2.2.3 About services

In support of the dynamics of interacting with services are a set of concepts that are about services themselves. These are the service description, the execution context of the service and the contracts and policies that relate to services and service participants.

Figure 7: About services

Service description

The service description represents the information needed in order to use a service. In most cases, there is no one "right" description but rather the elements of description required depend on the context and the needs of the parties using the associated entity. While there are certain elements that are likely to be part of any service description, most notably the information model, many elements such as function and policy may vary. The purpose of description is to facilitate

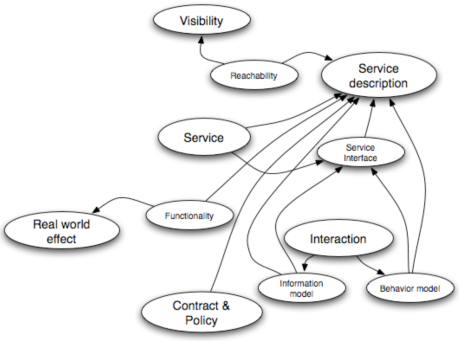

Figure 8: Service description

interaction and visibility, particularly when the participants are in different ownership domains, between participants in service interactions. By providing descriptions, it makes it possible for potential participants to construct systems that use services and even offer compatible services. Best practice suggests that the service description SHOULD be represented using a standard, referenceable format.

The service description makes available critical information that a consumer needs in order to decide whether or not to use a service. In particular, a service consumer needs to possess the following items of information:

1. That the service exists and is **reachable**

2. That the service performs a certain function or set of functions
3. That the service operates under a specified set of constraints and policies
4. That the service will (to some implicit or explicit extent) comply with policies as prescribed by the service consumer
5. How to interact with the service in order to achieve the required objectives, including the format and content of information exchanged between the service and the consumer and the sequences of information exchange that may be expected

Service Reachability Reachability is an inherently pairwise relationship between service providers and service consumers. However, a service description SHOULD include sufficient data to enable a service consumer and service provider to interact with each other. This MAY include metadata such as the location of the service and what information protocols it supports and requires. It MAY also include dynamic information about the service, such as whether it is currently available.

Service Functionality A service description SHOULD unambiguously express the function(s) of the service and the real world effects that result from it being invoked. The description of functionality may include, among other possibilities, a textual description intended for human consumption or identifiers or keywords referenced to specific machine-processable definitions. Part of the description of functionality may include underlying technical assumptions that determine the limits of functionality exposed by the service or of the underlying capability.

Policies Related to a Service A service description MAY include support for associating policies with a service and providing necessary information for prospective consumers to evaluate if a service will act in a manner consistent with the consumer's constraints.

Service Interface The service interface is the means for interacting with a service. It includes the specific protocols, commands, and information exchange by which actions are initiated that result in the real world effects as specified through the service functionality portion of the service description. The specifics of the interface SHOULD be syntactically represented in a standard referenceable format. These prescribe what information needs to be provided to the service in order to access its capabilities and interpret responses. This is often referred to as the service's information model. it is assumed that for a service to be usable, its interface MUST be represented in a format that allows interpretation of the interface information by its consumers.

Policies and Contracts

A policy represents some constraint or condition on the use, deployment or description of an owned entity as defined by any participant. A contract, on the other hand, represents an agreement by two or more parties. Like policies, agreements are also about the conditions of use of a service.

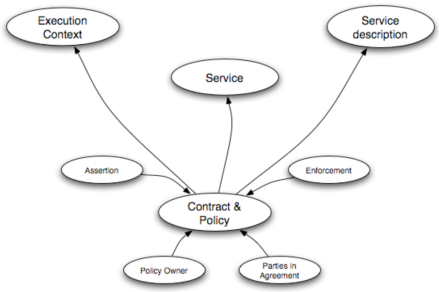

Figure 9: Policies and Contracts

Service Policy Conceptually, there are three aspects of policies: the policy assertion, the policy owner (sometimes referred to as the policy subject) and policy enforcement.

- **Policy Assertion**: for example, the assertion: "All messages are encrypted" is an assertion regarding the forms of messages. As an assertion, it is measurable: it may be true or false depending on whether the traffic is encrypted or not. Policy assertions are often about the way the service is realized; i.e., they are about the relationship between the service and its execution context.
- **Policy Enforcement**: techniques for the enforcement of policies depend on the nature of the policy. Conceptually, service policy enforcement amounts to ensuring that the policy assertion is consistent with the real world. This might mean preventing unauthorized actions to be performed or states to be entered into; it can also mean initiating compensatory actions when a policy violation has been detected.

Policies potentially apply to many aspects of SOA: security, privacy, manageability, Quality of Service and so on. Beyond such infrastructure-oriented policies, participants MAY also express business-oriented policies – such as hours of business, return policies and so on.

Service Contract Whereas a policy is associated with the point of view of individual participants, a contract represents an agreement between two or more participants. Like policies, contracts can cover a wide range of aspects of services: quality of service agreements, interface and choreography agreements and commercial agreements. Note that we are not necessarily referring to legal contracts here.

Thus, following the discussion above, a service contract is a measurable assertion that governs the requirements and expectations of two or more parties. Unlike policy enforcement, which is usually the responsibility of the policy owner, contract enforcement may involve resolving disputes between the parties to the contract. The resolution of such disputes may involve appeals to higher authorities.

Execution Context

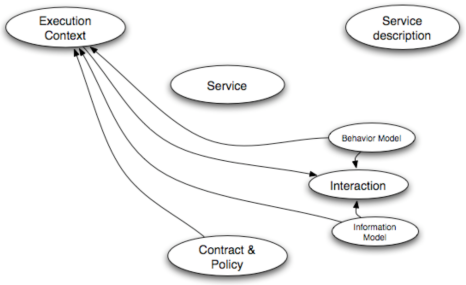

Figure 10: Execution Context

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.

As discussed in previous sections of this document, the service description (and a corresponding description associated with the service consumer and its needs) contains information that can include preferred protocols, semantics, policies and other conditions and assumptions that describe how a service can and may be used. The participants (providers, consumers, and any third parties as noted below) must agree and acknowledge a consistent set of agreements in order to have a successful service interaction, i.e. realizing the described real world effects. The execution context is the collection of this consistent set of agreements.

The consumer and provider can be envisioned as separate places on a map and, for a service to actually be invoked, a path must be established between those two places. This path is the execution context.

The execution context also allows us to distinguish services from one another. Different instances of the same service – denoting interactions between a given service provider and different service consumers for example – are distinguished by virtue of the fact that their execution contexts are different.

Finally, the execution context is also the context in which the interpretation of data that is exchanged takes place. A particular string has a particular meaning in a service interaction in a particular context – the execution context.

2.2.4 IBM SOA Reference Architecture

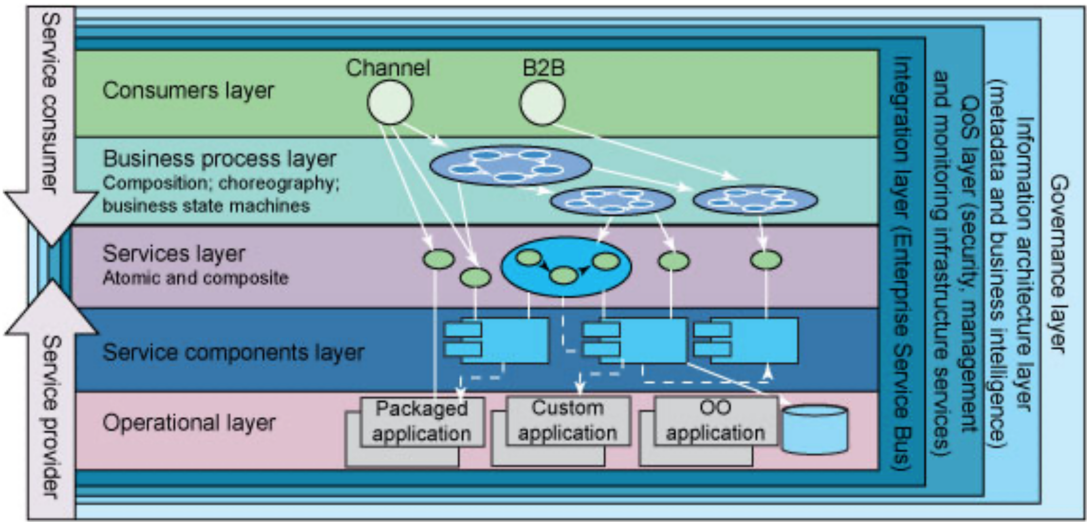

Figure 11: IBM SOA Reference Architecture

-

The SOA RA allows architects to use its contents such as the building blocks as a checklist of elements: architectural building blocks and their relations in each layer, the options available, decisions that need to be made at each layer. The layers provide a starting point for the separation of concerns needed to build an SOA. Each group of the separated concerns are represented in their own "layer."

Within each of the layers depicted in Figure 11 there are Architectural Building Blocks (ABBs) that represents a basic element of reusable functionality and fulfill the key responsibilities of that layer. Each ABB resides in a layer, supports capabilities, and has responsibilities. ABBs are also connected to one another across the layers and provide a natural association between layers. If a particular connection between ABBs occurs consistently across layers to solve a certain problem this then defines a pattern of ABBs as well as the valid interaction sequences between the architectural building blocks.

Along with the ABBs there are also the capabilities that they support. A capability is defined by the Open Group in TOGAF 9, as "an ability that an organization, person, or system possesses". So, expanding on that definition, ABBs provide the technical resource to allow an organization, person or system to be able to provide their defined capability. An ABB, providing support for one or more capabilities, that can be realized by one or more components or products; examples of the responsibilities of an ABB include: service definition, mediation, routing, etc.

Summary of the layers

First, as we can see from Figure 11, there are some layers that are visible only to service providers and some visible only to service consumers which are Operational Layer + Service Components Layer and Consumers Layer + Business Process Layer respectively. Then in the middle we have the Service Layer which is the only one visible to both actors.
The layers that are defined in an SOA RA are as follows:

- **Operational Layer**: captures the organization's infrastructure, both new and existing, needed to support the SOA solution at design, deploy and run time. 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 (IaaS) construct and the rest of the SOA in the wider context of cloud computing.

- **Service Component Layer**: contains software components, each of which provide the implementation or "realization" for a service, or operation on a service. The layer also contains the functional and technical components that facilitate a Service Component to realize one or more services. Software components hide the complexity of the underlying system and exposes with standard interface what defined in the Service Layer Its aim is to satisfy the Service Level Agreements.

- **Service Layer**: consists of all the logical services defined within the SOA. This layer contains the descriptions for services, business capabilities, and IT manifestation that are used/created during design time as well as service contracts and descriptions that are used at runtime. Here service consumer is able to read the service description, while the service provider can decide to offer service components 'as-ease' or can combine services to create something new to offer.

- **Business Process Layer**: covers the process representation, composition methods, and building blocks for aggregating loosely coupled services as a sequenced process aligned with business goals. Data flow and control flow are used to enable interactions between services and business processes. The role of this layer is central in pursuing the Business-IT alignment. Service consumer here can compose services using two strategies: orchestration and choreography.

- **Consumer Layer**: is the point where consumers, whether if be a person, program, browser or automation, interact with the SOA. It enables an SOA solution 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). This layer provides the capability to quickly create the front end of the business processes and composite applications in order to respond to changes in the marketplace.

- **Integration Layer**: is a cross-cutting concern that enables and provides the capability to mediate, which includes transformation, routing and protocol conversion to transport service requests from the service requester to the correct service provider.

- **Quality of Service Layer**: is a cross-cutting concern that supports non functional requirement (NFR) related concerns of an SOA and provides a focal point for dealing with them in any given solution. It provides the means of ensuring that an SOA meets its requirements with respect to: monitoring, reliability, availability, manageability, transactionality, maintainability, scalability, security, safety, life cycle, etc.

- **Information Architecture Layer**: is a cross-cutting concern that is responsible for manifesting a unified representation of the information aspect of an organization as provided by its IT services, applications, and systems enabling business needs and processes and aligned with the business vocabulary. This layer includes information architecture, business analytics and intelligence.
- **Governance Layer**: is a cross-cutting concern that ensures that the services and SOA solutions within an organization are adhering to the defined policies, guidelines and standards that are defined as a function of the objectives, strategies and regulations applied in the organization and that an SOA solutions are providing the desired business value.