

About the role and importance of the Service Domain

Created by [Iwan Gramatikoff](#)

Following the long thread "Farewell to the service domain", it appeared valuable to provide some form of summary on the role and importance of the Service domain in the SID and in the Framework in general.

Some members have specific contexts where the differentiation between Services and Resources is not obvious. This has led to the questioning of the necessity of a differentiation between the Service and Resource domains. A typical situation would be the one of a Service Provider whose services are very technology / infrastructure oriented. If the Service Provider offering is for example Carrier Ethernet Services, there is not a lot of differentiation between the features / characteristics qualifying the service and the underlying infrastructure.

Here is a more specific example:

How is the configuration of an Optical Ring for a customer (CFS/RFS) different than the configuration within a provider's / operator's infrastructure (Resource)? Perhaps on the spec side the Service Configuration reuses the Resource Configuration, but adds more, such as protection from use by / assignment to other customers. On the entity side the ring is configured within the Resource domain, with RFSs that represent the protection "service".

This is how it can be explained when asked, but without the Configuration ABEs in the SID it is often not that easy.

But even in the case where offered services are very technology / infrastructure oriented, it is valuable to abstract a vendor/ technology neutral service layer from the infrastructure layer. It is about applying the loose coupling concept (just as in SOA) to enable an architectural service layer encompassing the applications capabilities, the processes and information which is decoupled from the vendor / technology specific resource layer. It is basically about providing a platform enabling business agility, meaning which can evolve with changing business requirements.

Indeed, we all understand that the Service Provider landscape is transforming and that the transformation is ongoing. And it is a primary goal of the Framework to provide help and guidance in this business and systems transformations. The importance of this role has been enforced by the new positioning of the Framework as an integrated business architecture framework for the ICE service industry.

Service Providers face paradoxes, such as:

1. Differentiating from each other by providing highly individual offerings through the combination and integration of various services. The services to be combined and integrated can range over various technologies / platforms and some might be provided by third party providers. On the other hand, these same providers need to improve their operational effectiveness through standardization and automation of their production model.
2. Integrating a wider range of services / technologies leading to an increasing complexity that cannot be relayed to the customer. Underlying technology has to be invisible to the users, the focus has to be placed on the customer experience, on the convenience.

The domain concept promoted by the Framework and more specifically by the Information Framework is essential to address these paradoxes. We will focus here on how the Product / Service / Resource domains are essential to address these challenges.

We will start by 2 first simplified statements that position the core business object (product, service, resource) of these 3 domains in a generic business context. Before we proceed, we have to explain that the 2 following statements align respectively with the two accepted viewpoints on differentiating Product vs. Service (namely viewpoint 1 & viewpoint 2. see Product/Service nugget) We will highlight that these viewpoints do not differ as far as (1) the differentiation between Services and Resources and (2) the importance of the service domain (core focus of this post) are concerned

View point 1 would promote the following statement:

"A customer buys a product offering, pays a product, uses services that are realized by resources"

View point 2 would promote the following statement:

"A customer buys a product offering, pays and uses a product functionally made available by services that are realized by resources"

- The product offering holds what the customer buys: The product offering is the service provider` "promises" to the customer (see TR153). It is the point of convergence of the 4Ps (Price, Product, Place and Promotion).
- The product holds what the customer pays for (viewpoint 1) completed, in the view point 2, by what the user/customer uses. It is the object of the contractual agreement between customer and provider.
- The service holds the functionality made available to the user/customer in the frame of a purchased product. It holds the functionalities fulfilling the customer relevant usage scenario ("use cases" Viewpoint 1 in TR153). It also holds the necessary technical functionality supported by the underlying technical resources (viewpoint 2 in the TR153).
- Resources are the necessary components (physical or non-physical) that realize the services according to the customer expectation set by service Provider` promise.

These statements highlight the pivotal role the service plays between:

- (1) The market/customer perspective encompassing what the customer sees, buys, uses and pays for.
- (2) The technical perspective encompassing the tangible or intangible things "allocated", "installed", "configured", "activated" and "consumed" in the underlying SP infrastructure.

The Information Framework further sharpens the definition of these business objects by introducing: ProductOffering, Product(Specifications), CustomerFacingService(Specifications), ResourceFacingService(Specifications), LogicalResource(Specifications), PhysicalResource(Specifications).

Revisiting these 2 simplified statements with the Information Framework entities leads to the following:

View Point 1

“A Customer buys a ProductOffering, pays a Product, uses CustomerFacingServices that are enabled by ResourceFacingServices, which are realized by Resources (PhysicalResources and/or LogicalResources).”

View point 2

“A Customer buys a ProductOffering, pays and uses Product made functionally available by CustomerFacingServices that are enabled by ResourceFacingServices, which are realized by Resources (PhysicalResources and/or LogicalResources).”

- The CustomerFacingService (CFS) may give the means to present customer “friendly” functionality to the customer. The meaning of friendly may range from “Technology agnostic”, to “protocol agnostic” up to just “convenient”. This depends on how the CFSspec are instantiated. It is all about modeling! J
- The ResourcesFacingService (RFS) holds the technical functionality supported by the underlying resources.

However, what we want to show here is the adequacy of this statement to the mentioned paradoxes. In the background of this statement, there is the business domain structure with the Service domain allowing a clear decoupling between the Product and the Resource domains. The service domain enables the service provider’ artifact bridging the customer perspective and the engineering perspective by focusing on functionality while filtering the technology. It also gives the provider the option to offer and manage services without owning the technological infrastructure.

In an architectural perspective, the service domain ranging over processes and applications, this decoupling of layers is not only defined in terms of semantic definitions of Product vs. Service vs. Resource, but even more in terms of processes governing the lifecycle of these business objects and application capabilities supporting the execution of these processes. It is the capturing of the interactions between process, information and application into product independent business services, aligned with the business roles involved in the execution of these business services which provides the agile and flexible execution platform we were mentioning in the beginning of this article.

In this perspective, one key consideration in differentiating Services from Resources, remains the differentiation in terms of processes or lifecycle of these business objects. Such a process oriented perspective is extremely helpful in defining what a given business object is. Indeed, the question will rather be what processes, what lifecycle steps a given business object will go through. Thus referring back to the process definitions along the Service / Resource domains will guide in answering if a given business object is a Service or a Resource (What do I intend to do with it?)

The domain structure Product / Service / Resource completed with this process / lifecycle reading is also essential to set the architectural footprint aligning business and technical stakeholders (NW & IT) concerns. This is not only critical from the process perspective but also from a portfolio management and business roles perspective. The distinction between Product (Spec / Offering), Service (Spec) and Resource (Spec) helps to structure the concerns of the various stakeholders / roles in the organization (portfolio manager, product manager, service manager, service engineers, production architect, resource engineer...).

Finally this domain structure allows members to use it, as an Integrated Business Architecture framework in their business and systems transformation process. Built on the concept of separations of concerns, this structure enables the Service Providers to partition their business and systems transformation program by enabling a consistent scoping of project along the process, information, application and integration dimensions. This partitioning is essential to target quick wins in such programs.

As such the full visibility of these 3 domains in the Frameworkx structure appears as fundamental.

As a result when pursuing Frameworkx optimization initiative it is important to bear in mind that any structural enhancement has to be relevant from a business perspective. Considering this, moving toward an optimized representation may not be relevant for a better business perspective.

Now let’s remember that the Frameworkx are static frameworks. They are just like dictionaries. Each ICE player will pick the elements which are relevant to the dynamics of its own business/operation model and leave the other ones. “It’s all about choice” as long as you do not collapse the industry business consensus.

Iwan Gramatikoff & Serge Garcia, Edelweiss Service Consulting (www.edelweiss-sc.net)

With the contribution of John Reilly, TM Forum