Shared IT Services in the Enterprise

A general discussion of service

The dictionary definition of *service[[1]](#footnote-1)* is an act of helpful activity and, in the context of business[[2]](#footnote-2), the help is providing some business function that generates business value. In general, services are used to realize desired effects that we either lack the ability or inclination to fulfill ourselves.

The general idea of service may not have an IT component but it does imply the existence of a capability to produce the service effects. Without the underlying capability, there is no service. For example, the service of filling a prescription requires a pharmacist and an adequately stocked pharmacy; a random person in a white jacket standing in front of a cash register with a sign saying Pharmacy is not sufficient.

Figure 1 - Perspectives of Service

Service in the SOA context 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 the terms specified in the service description.[[3]](#footnote-3)  Here, we focus on service as the use of software to enable business.

Note that the SOA service[[4]](#footnote-4) explicitly depends on the existence of the underlying capability and is the means of using this capability without explicit knowledge of how the capability realizes its effects. We have no idea how the pharmacy maintains its inventory, but the person at the cash register of a functioning pharmacy provides the access through a readily understandable interface and consistent with policies and practices for operating a pharmacy.

In the following, the unmodified term *service* refers to the combination of the underlying capability and the means to access the capability. Thus, a consumer walks up to the cash register at the pharmacy, hands over a prescription, and upon payment gets what the prescription ordered. The term *underlying capability* refers to the unseen processes and activities that go into providing the desired effects, such as the full range of activities performed by the pharmacy staff. The term *SOA service* refers to the software ~~interface~~ through which a consumer interacts in order to make use of the underlying capability and realize the desired effects, and is analogous in our pharmacy example to the front counter person with whom the consumer interacts.

As shown in Figure 1, the external consumer, here identified as an application, interacts with the SOA service through a well-defined interface accessible over a network. The capability being accessed is an integral part of the Service (shown as the blue, enclosing rectangle) and the interface of the SOA service acts also as the interface of the Service.

Note that an underlying capability may be accessed by any number of access mechanisms. In the pharmacy example, the pharmacy capability may be accessed through a phone call or through an online request. In the case of email, any number of software clients (e.g. Outlook, Apple Mail, gmail) that use a recognized and supported email protocol (e.g. POP, IMAP, Exchange) may be used to access the underlying email capability.

Services in the enterprise

An **Enterprise Shared Service** (ESS) is a SOA service that is visible and can be accessed by users across the enterprise, subject to appropriate security and privacy restrictions. An important aspect of ESS as a SOA service is the ability to effectively cross ownership boundaries, i.e. the service provider can be in a different organization than the service consumer.  The focus is on ESS that are implemented according to the SOA paradigm.  This does not preclude hybrid environments where non-SOA services may concurrently be available.

Whereas having numerous pharmacies may provide value through competition, redundant services provided within an organization often represent additional costs to develop and maintain, including additional costs to provide support to the consumers. This is particularly the case when the software is involved in delivering the service. While some degree of duplication may be warranted to provide a platform for innovation or as a means of risk mitigation, a major driver for ESS is to effectively reduce unwarranted duplication[[5]](#footnote-5).

ESS is intended to provide other benefits. With standalone software systems, the development team often needs to include expertise to develop areas that are not of priority interest to those who want the systems. For example, an organization may specialize in data analysis, where visualization of results may be important for the final product. However, visualization may not be part of the organization’s core expertise and using visualization from experts in the visualization field relieves the organization of extraneous work. Indeed, the use of someone else’s visualization services may free sufficient resources to allow the organization to offer their analysis specialty as a service for others.

Identifying and defining SOA services

As stated above, a service provides a business function that generates business value; in other words, the service satisfies a business need. The more ubiquitous is the need, the larger is the prospective consumer community available to make use of the service.

As with any system development, the first step in identifying the service is to identify the need and the outcomes that will satisfy the need. This sets the stage for identifying the underlying capabilities that must be available, the target consumer community to be served, and the access mechanism that enables the consumer community to access the capabilities. In addition, the conditions of use must be described, both in terms of what a provider requires of a consumer and what the consumer can expect in terms of using the service.

These attributes of a service can be summarized in terms of the following:

* Preconditions: what conditions need to exist for service to realize intended results
* Inputs: what information needs to be communicated in an interaction with a service to initiate and control its actions
* Behavior: given input, what does the accessed capability do to produce outputs and post-conditions
* Outputs: what is returned by service as an effect of an interaction, including possible fault responses
* Post-conditions: what conditions result after the successful completion of an interaction

To restate, the preconditions describe the world as it must be for the service (both the underlying capability and the SOA service access) to be effective. In the OASIS SOA Reference Model (SOA-RM), the collection of these preconditions is discussed in terms of the *execution context*. Given necessary preconditions, the service will exhibit the described behavior, taking inputs to be acted upon and which may act as control parameters of the service and generating output and post-conditions. In a request-response interaction pattern, the response is the output, but the service may have effects that go beyond its output. For example, the output of an ATM withdrawal is cash and a receipt but a post-condition is the account is debited by the amount of cash received.

The five attributes of a service are likely to be captured in detail in the service description of an operating service, but it is important to be able to succinctly capture these attributes early in order to understand clearly what the service is expected to accomplish. This also describes a common pattern and common interface template through which a consumer can interact with the service without needing to specify all the details of the interaction. The SOA-RAF[[6]](#footnote-6) discusses details of the structure and use of the service description.

Characterizing SOA services

A SOA service of interest may provide access to a single function or set of functions

* focused on a single need that is associated with an extremely large user base, such as enterprise email; or
* accessing a capability implementation that may be part of many diverse solutions, such as Google translator; or
* realized through numerous underlying capability implementations but accessed through a common interface and delivering common outcomes, such as a search box in a browser for accepting search terms and returning a list of search results.

As discussed above, the first step in identifying ESS should be to decide what benefit the service is to provide, then who will use the service and how they will use it. This should lead to understanding which of the service patterns best describes how the service needs to be built and how it will provide access to consumers.

Understanding the service pattern to be employed is important because this indicates the benefits to be gained and the constraints that may apply. For example, reducing the number of email implementations may save in enterprise costs but its benefits do not go beyond optimizing email and likely only requires installation and testing of an existing email solution.

Conversely, a service accessing a translation capability can be used as part of any solution needing that capability. Savings here come from reuse of the capability; the SOA service provides the consistent access for any of those consuming solutions.

Finally, early SOA advocates concentrated on the idea of a uniform, coarse-grained interface that could act as access to any suitable underlying capability that implemented the same interface. Search engines are a compelling example where an area of active research continues to provide new tools that fall under the heading of “search” but certainly do not represent unwarranted duplication. The OpenSearch initiative[[7]](#footnote-7), as an example, enables browsers to use the same interface for any number of conformant search engines. The Content Discovery and Retrieval (CDR)[[8]](#footnote-8) work demonstrates how a single search interface can support an unlimited number of query formulations.

The types of SOA services discussed point to different ownership[[9]](#footnote-9) models and require different trust relationships between the consumer and the provider. In the email pattern, an IT organization will likely be designated to own the service over the long term, and trust that the service will continue to be available amounts to a combination of trust in the designated IT organization and possibly trust in the designating organization that a replacement provider will, as necessary, be identified. In the Google translator pattern, someone (Google) offers a service and trust amounts to the consumer trusting that Google will continue to provide that single service. Here, trust is likely based on past experience with the provider, the experiences of other consumers, and criticality of the use. The common interface pattern is similar to the Google translator pattern but the common interface facilitates engaging a replacement service if the chosen service is temporarily or permanently unavailable. A more complete discussion of trust and risk in the SOA ecosystem can be found in the OASIS SOA-RAF.

Related issues on which we may want to expand:

* ownership
* control (governance) of marketplace and participants
* funding to develop / payment for use

-------------

Other things to possibly add / draw upon

- steps in Aligning with the SOA Paradigm

- more examples and discussion of CDR

-------------

MP:

This (waranted/unwarranted duplication) is more complex issue: on one hand, an enterprise must provide means for the business continuity, i.e. has more than a single means (single point of failure) that provide the same business functionality and RWE; from another hand, a notion of ownership and independence lead to duplication of the means.

IMO, the major purpose of shared services is not de-duplication, but unification and centralization of specialized competency.

Moreover, an introduction of shared services never succeeds without centralized governance that prohibits and controls any attempts to construct services for local need that may be used to overcome shared services.

KL:

If I thought I had a better way of providing an enterprise/shared service, how do I “get permission” to demonstrate my ideas? I realize there is a conflict in goals here, but how to clearly state that conflict and what to provide as guidance around it.

1. http://dictionary.reference.com/browse/service [↑](#footnote-ref-1)
2. By business we mean any shared activity whose objective is to satisfy particular needs of each participant. [↑](#footnote-ref-2)
3. OASIS Reference Model for Service Oriented Architecture 1.0, OASIS Standard, October 12 2006. <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf> [↑](#footnote-ref-3)
4. [For discussion] The term *SOA service* is not in the RM or RAF. I use it to indicate the access mechanism piece, which has a large IT component but is likely not exclusively IT. The major challenge is to appreciate the distinction between the IT and the business while realizing that both aspects must be brought into the discussion without conflating the two. I have no problem substituting a different term. [↑](#footnote-ref-4)
5. This is often discussed in terms of “eliminating” duplication, but it is unlikely duplication will be completely eliminated because achieving such an absolute is often a poor return on investment. As noted, some duplication may be warranted and have value for risk mitigation and as an incubator for innovation. The goal should be to create a mindset where unwarranted duplication, i.e. duplication without obvious value, is not tolerated. [↑](#footnote-ref-5)
6. Reference Architecture Foundation for Service Oriented Architecture Version 1.0, OASIS Committee Specification, 4 December 2012. <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/soa-ra-v1.0-cs01.pdf> [↑](#footnote-ref-6)
7. http://www.opensearch.org [↑](#footnote-ref-7)
8. CDR reference [↑](#footnote-ref-8)
9. In the current context, to “own” a service means to take responsibility for creating and/or maintaining and for provisioning a service (both the underlying capability and the SOA service to access the capability). [↑](#footnote-ref-9)