1.6 Introducción a las capas de servicios, modelos de servicios y composición de servicios.

Raúl Estrada

Noviembre 2020

Service Models and Service Layers

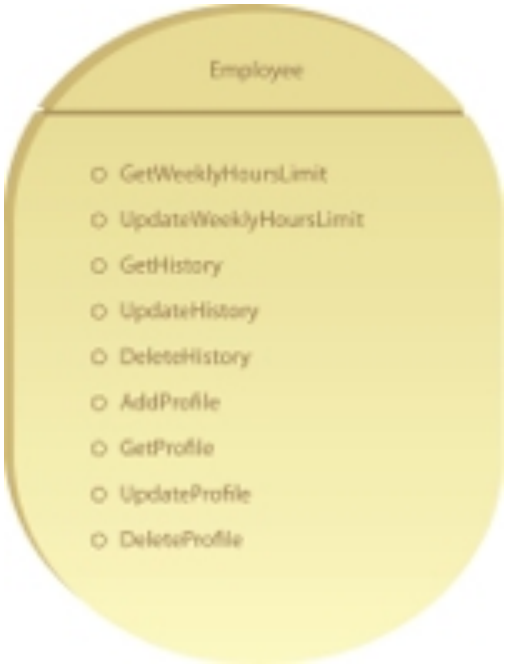
- Service models are used to classify units of logic according to the nature of the logic itself.
- A service should only contain units of logic that belong to the same service model. Furthermore, service models are used to provide organization and structure to a service inventory through the introduction of service layers.
- A service layer groups services belonging to the same service model to facilitate the governance and management of a service inventory.

Service Models and Service Layers (2)

- A given service inventory will usually contain multiple services that are grouped based on each service model.
- A service layer is therefore a collection of services related by the type of logic they encapsulate.
- Service layers help to establish logical domains that support meaningful governance.

Entity Services

- In just about every enterprise, there will be business model documents that define the organization's relevant business entities.
- Examples of business entities include customer, employee, invoice, and claim.
- The *entity service* model represents a business-centric service that bases its functional boundary and context on one or more related business entities.
- It is considered a highly reusable service because it is agnostic to most parent business processes.

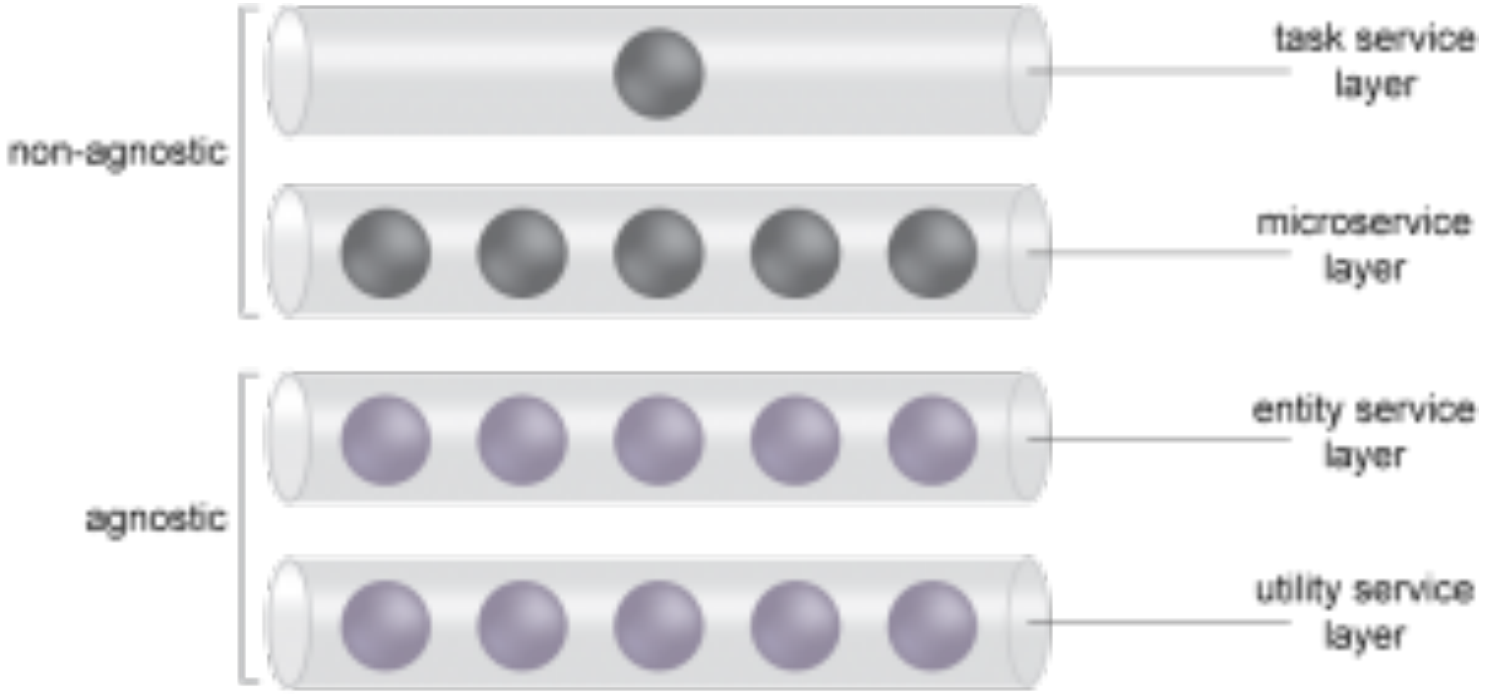


Task Services

- A business service with a functional boundary directly associated with a specific parent business task or process is based on the *task service* model.
- This type of service tends to have less reuse potential and is generally positioned as the controller of a composition responsible for composing other, more process-agnostic services

Utility Services

- It is dedicated to providing reusable, cross-cutting utility functionality, such as event logging, notification, and exception handling.
- It is ideally application agnostic in that it can consist of a series of capabilities that draw from multiple enterprise systems and resources, while making this functionality available within a very specific processing context.

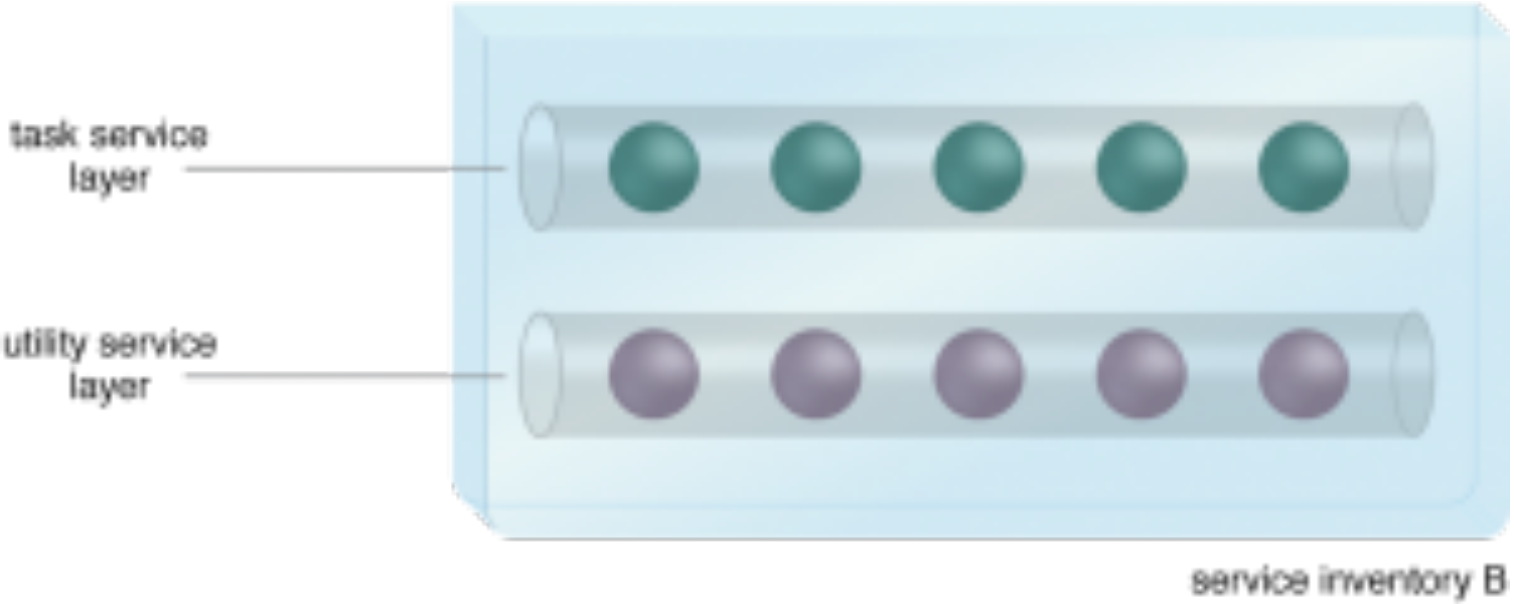


The common service layers, each of which is based on a service model.

The four layers

- *Although it is common to model a service inventory with all four layers, it is not absolutely required that they be used at the same time.*
- *The rule of thumb is that at least two layers exist.*
- *Given that it is generally desirable to have one layer abstract non-agnostic logic, for the purposes of driving a composition, the task service layer can be applied together with either the utility service layer or the entity service layer.*

Task + Utility

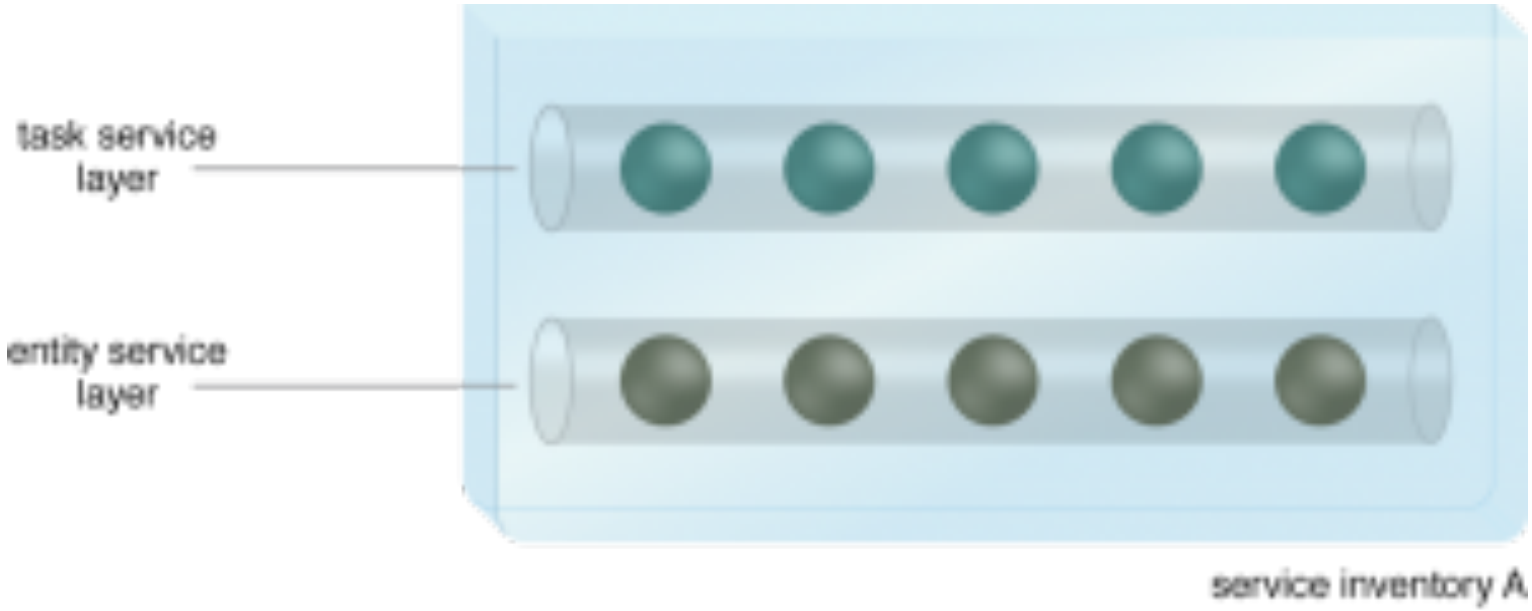


For example, the task + utility layer combination may be suitable when an organization does not have mature business entity documentation in place and cannot afford to invest in creating these models.

Task + Entity

- *The task + entity layer combination may be suitable when it is determined that the enterprise infrastructure cannot handle large service compositions.*
- *In this case, the organization may choose to invest in building entity services that also include the required utility logic to carry out all processing, and then perhaps add a utility layer later on by extracting utility logic into a separate set of services.*

Task + Entity



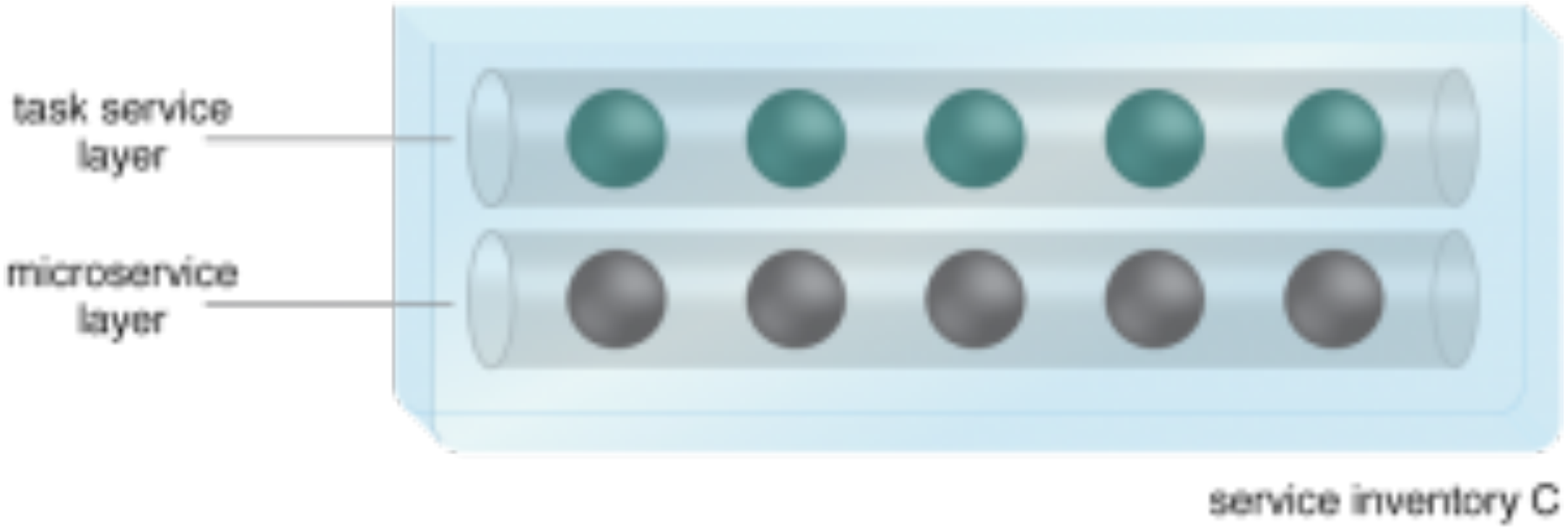
An example of the task + entity layer combination

Utility + Entity

- Even though they are technically possible, there are some combinations that are not recommended.
- For example, an inventory modeled only with the utility + entity layer combination may be insufficient because there is no abstraction of non-agnostic logic within services.
- In these cases, the non-agnostic logic must exist in traditional applications and the creation and use of agnostic services is likely due to an application modernization effort.

Task + Microservice

- Another approach that is not recommended is a task + microservice layer combination or a single task service layer.
- A service inventory based on just only non-agnostic services makes little sense because it would essentially be comprised of a series of independent, silo-based applications.
- Although it is common to find a set of single-purpose programs that represent some or all of an IT enterprise, from a service-orientation perspective, there is not much gain to this approach.
- One or more agnostic service layers are required to establish services as reusable enterprise resources.



An example of a task service + microservice layer combination, which is reminiscent of a silo-based application.