

# How to Create a Service Catalog for Delivering DBaaS

Michael Timpanaro-Perrotta  
Burt Clouse  
Oracle Corporation  
Redwood City, CA USA

**Keywords:** standardization database-as-a-service database cloud service catalog

## Introduction

A good service catalog is essential for IT service delivery. We will give an overview of service catalogs, and explain what makes a good catalog in general and specifically for DBaaS.

We will then present a standardized set of service offerings for Oracle DBaaS which DBaaS providers can use as-is, or tailor to their needs. Our goal is for providers to leverage these definitions to jumpstart their DBaaS initiatives on the right path.

## Evolution to Database as a Service

The promise of cloud computing—greater agility, less risk, and lower costs—is real, but realizing that promise depends on the approach you adopt. We have seen many enterprises evolve from a traditional IT estate towards a modern cloud environment by organizing their efforts into a phased journey:

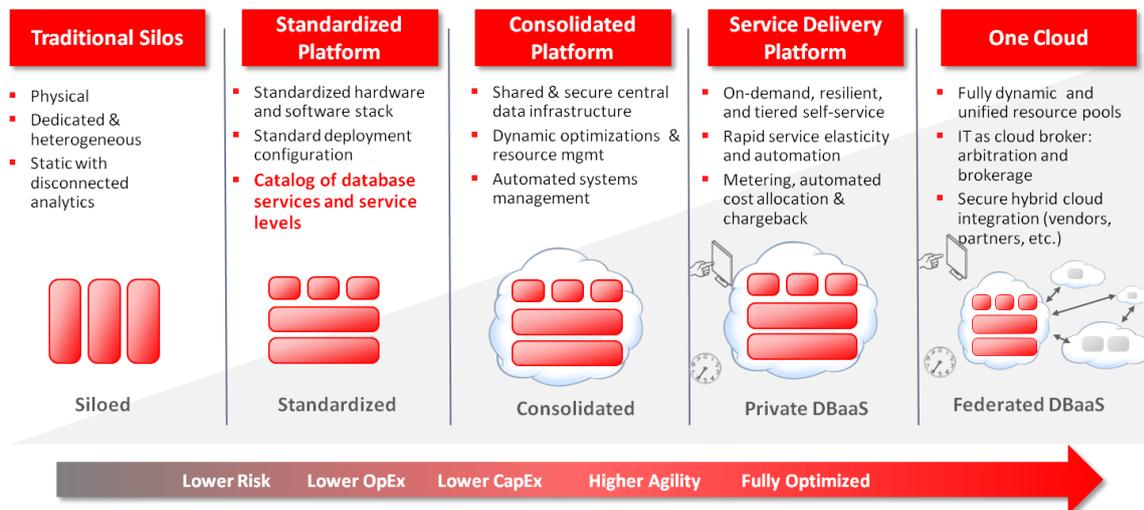


Figure 1: The Evolution to Database as a Service

The first step of this transformation is standardization, and one of the key deliverables that supports standardization is a service catalog. A service catalog is a collection of documents and artifacts which describe the services an IT organization provides, and specifies how those services are delivered and managed.

**Service Catalogs enable the evolution to enterprise cloud**

Standardized services can be deployed more quickly and repeatably than custom services. This benefits consumers directly since they have faster access to more reliable services, while the provider spends less time with the nuts and bolts of provisioning, and can focus on higher-value initiatives.

During its lifecycle, a standardized service will behave predictably during maintenance, unplanned outages, and under system load. Consumers and providers will have common, documented expectations for these scenarios.

Moving to a standardized environment is a significant change with important benefits. If done properly, standardization also paves the way for further steps. For example, if most deployments use the same operating system and database version, it is easier to consolidate those deployments together into a shared operating environment.

The service delivery phase of the evolution focuses on dynamic, policy-driven resource management. If the standardized components support those capabilities, then enabling a service delivery model is a simple matter of activating the supporting features and options of the environment – no upgrades or rearchitecting are needed.

### **Effective Standardization**

The effectiveness of standardization depends on several factors. One might assume that the more rigidly standardized an enterprise's services are, the better. But it is rarely possible to meet all of a large enterprise's IT requirements with a single deployment option. At the other end of the spectrum, each department or functional team cannot create individual "standards" that simply describe what each group happens to be doing.

For standardization to be effective, it must

- Apply across the entire enterprise
- Satisfy the majority of current and future use cases
- Allow for but minimize exceptions

Finding the right balance between per-consumer flexibility (low standardization) and business agility (high standardization) is possible only when consumers and providers collaborate to develop and evolve a framework that is consistent for all stakeholders and can be adapted over time. The service catalog and its proper management meet those goals.

### **Service Catalog Participants and Structure**

The catalog is divided into different components for different participants. Collectively, the service catalog components provide an end-to-end view of the entire IT estate and its management.

The two participants are consumers and providers. Consumers use the catalog to see what services they can order, and what are the consumer-relevant characteristics of those services, such as cost, support terms and conditions, and availability service level.

Providers own the entire catalog. They create the consumer-visible portion to present the service offerings. These offerings should be tailored to the company's consumers. The provider also

maintains a provider-only section of the catalog. This section documents how the provider will deliver those services to the consumers.

## Service Catalog Structure

Essential to separate business and technical aspects

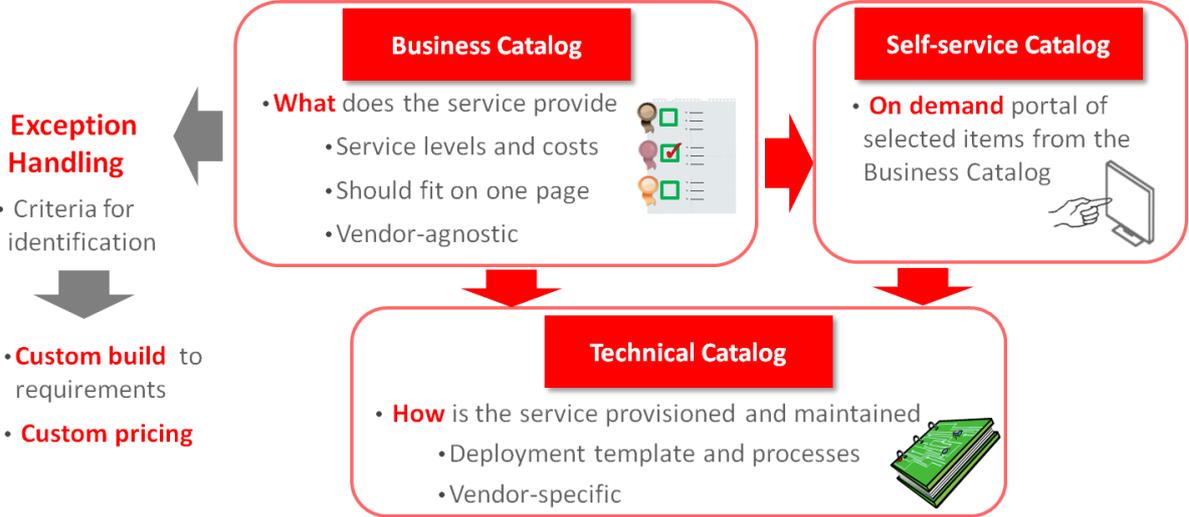


Figure 2: Service Catalog Structure

### Business catalog

The business catalog is the consumers’ view of the services available. Typically, three or more service offerings will be defined. Often they are labeled with a scheme such as Bronze/Silver/Gold to provide a high-level differentiation. For each service, the capabilities, policies and procedures of the service are documented in formal terms that are relevant to the consumer.

### Self-service catalog

An enterprise may wish to offer some of the services from the business catalog in an on-demand, self-service model. This will usually be a subset of the business catalog, i.e., those services well-suited for full automation, and services that are provisioned and deprovisioned frequently. Database services for test and development are a common example. More complex configurations, such as those with unique compliance or performance requirements, are not usually offered in a self-service catalog.

The self-service catalog will be an interactive portal through which consumers can deploy, monitor and manage services on-demand.

### Exceptions

It is unrealistic to expect that a service catalog can provide standardized offerings that address every possible service that a business may need. In fact, trying to enumerate every possible service variant will lead to a complex and confusing catalog. Instead, service attributes that trigger an exception should be spelled out, along with the response to non-standard requests. Unique sizing or isolation requirements are typical examples of such triggers.

To encourage customers to adopt the standard services, the exception process should be less agile and more expensive than for standard services. By designing services which address most current and future needs, and encouraging their adoption, exceptions can be minimized to perhaps fewer than ten percent over time.

### **Technical service catalog**

The technical service catalog is the provider's detailed guide for how to deploy and manage each service offering. For each service, there will be a standardized deployment template. The template includes every detail needed for provisioning the service: database edition, version, patch level; number of database instances; configuration parameters; options to be enabled (such as encryption to support a security requirement), and so on.

Ideally, the template should be fully portable so that the service can be deployed in a private or a public cloud. The template would specify any variations required to address the differences between the two provisioning models. There will also be variants to describe instantiations of the service for the different consolidation models in which services can be deployed.

### **Service Definition in the Business Catalog**

One of the challenges of creating a good business catalog is presenting the right amount and level of information to consumers. The goal is to enable consumers to easily choose the correct service for their needs, while ensuring that they know exactly what the service does and does not provide.

From studying industry artifacts and actual catalogs implemented by several of our customers, we have identified these essential service attributes which address the consumer-relevant aspects of service definition for Database as a Service offerings:

# Defining a DBaaS Service Offering

What is the business value delivered, and usage information:

## Service Attributes with Service Levels

<b>Availability</b>	• Resilience, Protection from Loss and Corruption
<b>Security</b>	• Secure Data & Config, Control and Monitor Access
<b>Performance</b>	• Capacity, Monitoring, Tuning, Management to Objectives
<b>Agility</b>	• Elasticity, Mobility, Scalability

## Generic Service Attributes

<b>Workload</b>	• Type, Application information, Lifecycle
<b>Support</b>	• Terms and Conditions of Support
<b>Capabilities</b>	• Included and Optional Capabilities and Services
<b>Cost</b>	• Charging Model and Metrics

Figure 3: Database Service Definition

To facilitate the bundling of service attributes in to a small set of service offerings, we have defined the following standardized services for Oracle Database as a Service in a private database cloud:

## Oracle DBaaS Service Offerings

<b>PLATINUM</b>	Mission-critical Trading
<b>GOLD</b>	Business-critical Customer Facing
<b>SILVER</b>	Production Departmental
<b>BRONZE</b>	Development Test

Figure 4: Oracle DBaaS Standardized Offerings

Within each offering, the service level for the various attributes are as shown below:

# Standardized Service Offerings

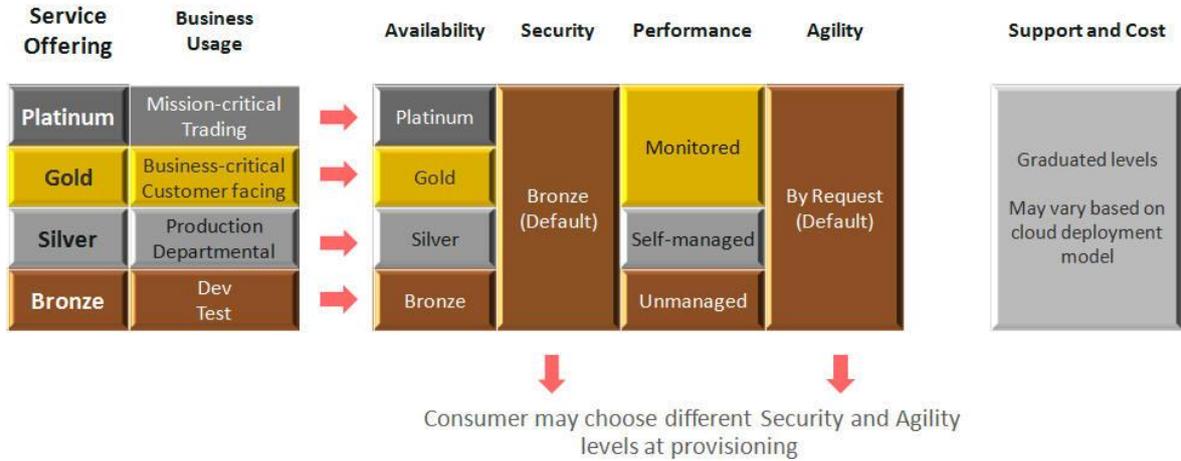


Figure 5: Service Levels within each Offering

For Availability, the four standard levels can be described to consumers in global terms, or in more detail (per class of outage):

# Standardized Availability Levels

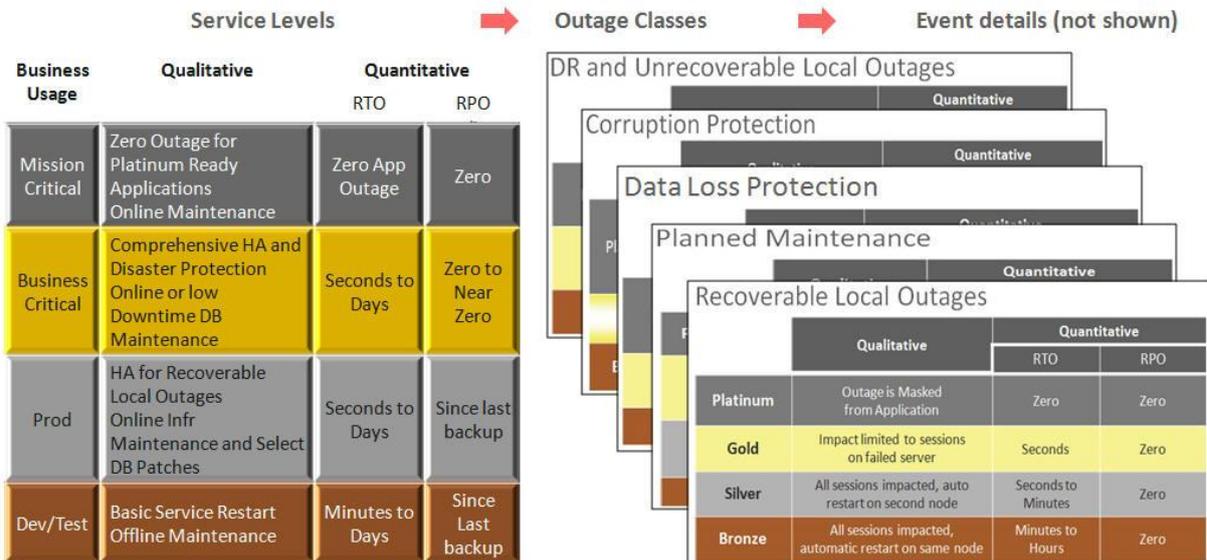


Figure 6: Standard Availability Levels

For Security, there are also four standard levels:

# Standardized Security Levels

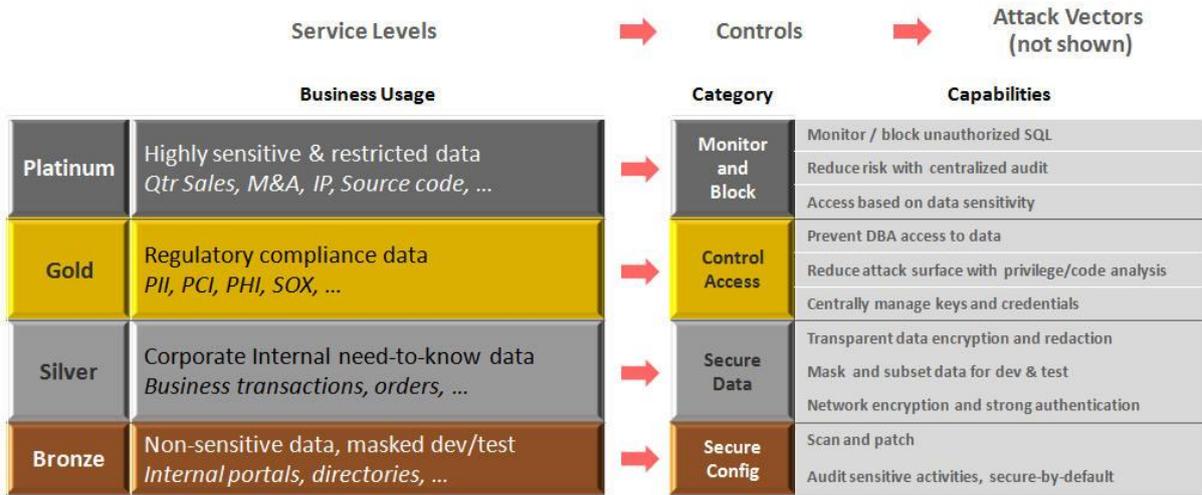


Figure 7: Security Levels

We are currently developing detailed standards for Performance and Agility service levels.

## Requesting Services

There are three approaches for allowing consumers to request services:

## Service Request Models Three Approaches



Figure 8: Service Request Models

We anticipate that most providers will adopt our standard offerings and will use one or both of the first two options. For providers who choose to offer service of their own design, the third option applies.

**Technical Catalog**

For the standard availability and security levels described in the business catalog section above, we have defined the architectural implementations. These are documented in the technical catalog and inform the provider how to deliver the respective service levels.

For example, Gold Availability is implemented as follows:

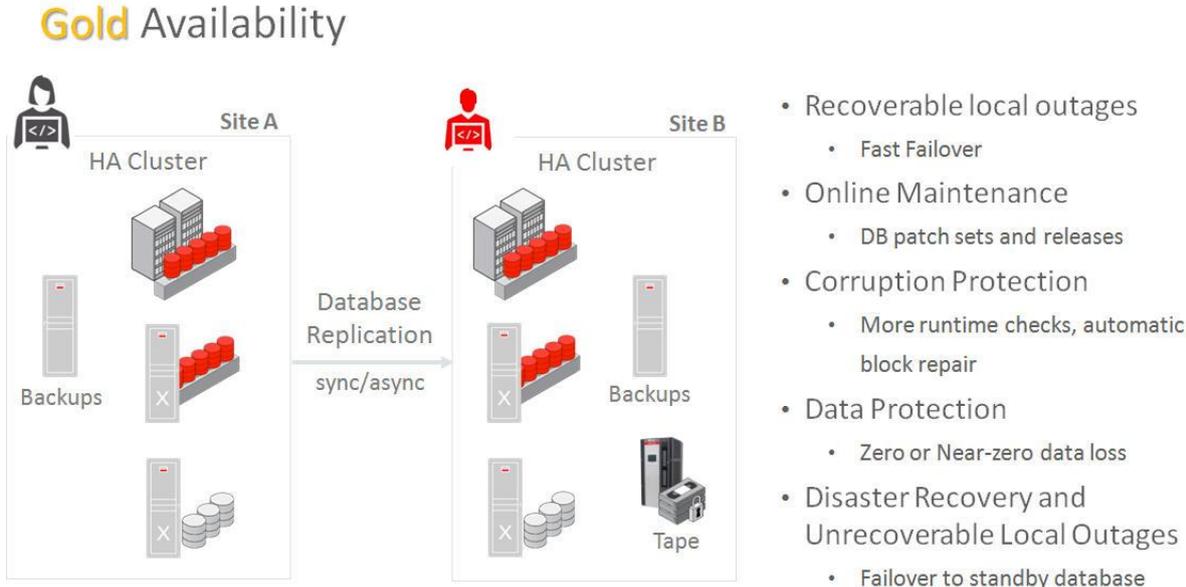
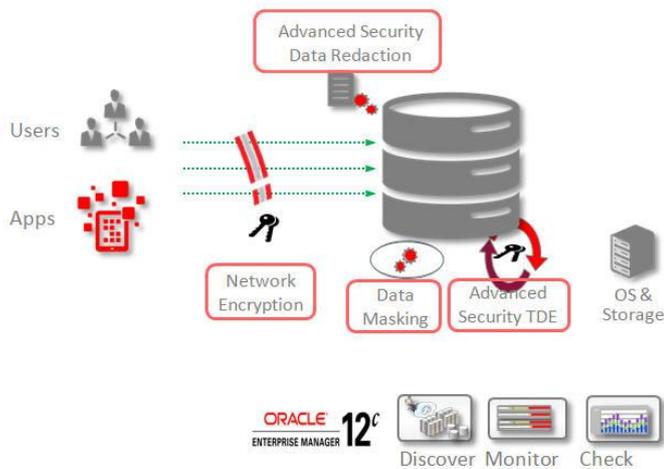


Figure 9: Gold Availability Architecture

As another example, Silver Security is implemented as follows:

## Silver Security Corporate Internal need-to-know



## Secure Data controls for Bronze **plus:**

- Preventive
  - Network encryption
  - Strong authentication
  - Encrypt data at rest
  - Data masking for dev/test
  - Data redaction for applications
- Detective
  - Same as Bronze
- Administrative
  - Discover sensitive data

Figure 10: Silver Security Architecture

Implementation guides for all availability and security levels are spelled out in our service catalog white paper on [OTN](#), and availability is described in greater detail in [MAA](#) collateral.

### Summary and Key Lessons Learned

- Keep it simple – clear, concise, consistent
- Clean separation of business (what it is) vs. technical (how it's done)
- Have plans for consumer buy-in with adoption incentives
- Define a process to identify and handle exceptions
- Minimize the number of distinct environments you will support
- Start small, add services incrementally as needed
- Stay with the plan but be willing to adjust
- Become your enterprise's optimal service provider

### Contact address:

**Michael Timpanaro-Perrotta**  
**Oracle Corporation**  
 400 Oracle Parkway  
 Redwood City CA 94065

Phone: +011 650-506-6177  
 email: michael.tp@oracle.com

**Burt Clouse**  
**Oracle Corporation**  
400 Oracle Parkway  
Redwood City CA 94065

Phone: +011 650-506-1458  
email: [bernard.clouse@oracle.com](mailto:bernard.clouse@oracle.com)  
Internet: <https://blogs.oracle.com/dbcloudcoverage/>