

“The Federated Data Model”

This article appeared in Volume One of
“*VIEWPOINT, Focus On: CMDB,*” published by BMC Software.



To receive a copy of *VIEWPOINT*, please go to www.bmc.com/cmdbviewpoint.



The **FEDERATED** Data Model

By **DOUG MUELLER**

A federated data model enables a configuration management database to provide a single source of record for configuration items.

Many IT organizations are looking to implement a configuration management database, known as a CMDB, so they can view information from many sources in a uniform way. The federated data model has been recognized as an effective means to structure a CMDB to ensure data consistency and simplify integration among the different service management processes, while still offering IT organizations a single source of truth. Companies can realize significant business value by having a single source of record that provides a logical model of the IT infrastructure to identify, manage, and verify all configuration items in the environment.

One effective source of guidance is the IT Infrastructure Library, known as ITIL[®], the industry framework for IT Service Management. To begin developing a Configuration Management strategy, set your objectives according to ITIL goals, which state that Configuration Management accounts for all the IT assets and configurations within the organization and its services. According to ITIL, the ideal CMDB should also provide accurate information on configurations and their documentation to support all the other Service Management processes. In addition, it must provide a sound basis for Incident Management, Problem Management, Change Management, and Release Management. It must have the ability to verify the configuration records against the infrastructure and correct any exceptions.



HOW THE ELEMENTS OF A FEDERATED CMDB FIT TOGETHER

A CMDB and its infrastructure consist of three layers. The first two, the CMDB layer and the CMDB extended data layer (which includes related data linked to or from the CMDB), make up what ITIL refers to as a CMDB. The third, the CMDB environment layer, includes applications that interact with the first two. Separating the ITIL-based CMDB into two layers is what differentiates the federated CMDB approach from other approaches.

The CMDB layer holds information about configuration items and their relationships. It stores only key attributes, and it links to CMDB extended data for additional details. This link ensures that even though the CMDB layer doesn't physically hold all attribute data or related data, it still serves as the source of record for configuration data. All requests for information can be made to the CMDB. If the data you need is not stored there, you will find reference links to where that data is stored and information regarding how to access it.

With integration to the CMDB extended data layer available, you don't have to modify the CMDB itself to hold related data.

The CMDB extended data layer holds related data that may not be considered a key attribute, such as change requests, a service impact model, and any configuration attributes that are not stored in the CMDB itself. Information in the CMDB extended data layer is linked to configuration data stored in the CMDB layer.

WHAT THIS STRUCTURE MEANS

A change request record could have a link through which you can access the instances of the configuration items it will change. Conversely, each configuration item instance could have a link through which you can access the change requests that affect it. Without a way to access this information, you run a risk that when someone makes a change to your IT environment it could impact other applications, causing outages and inconvenience.

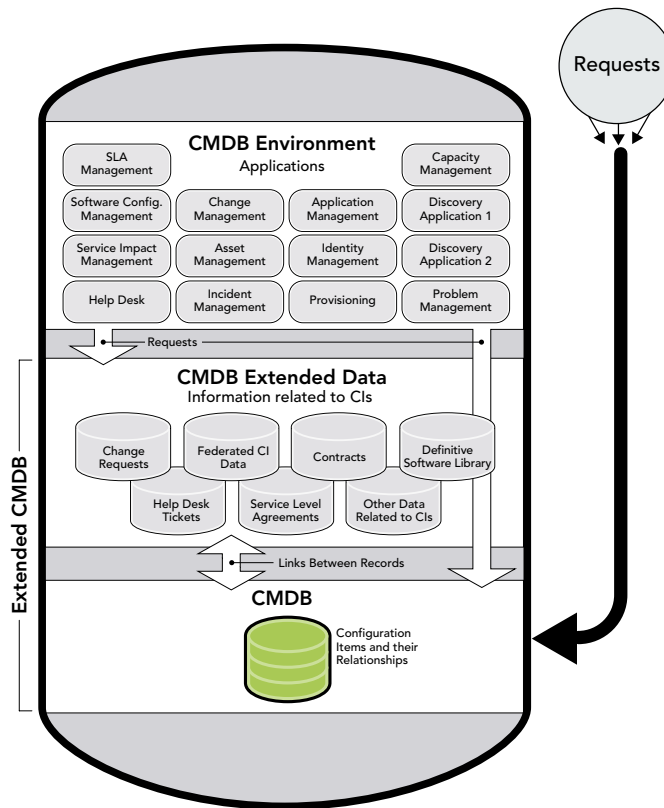


Figure 1. Elements of a Federated CMDB

With integration to the CMDB extended data layer available, you don't have to modify the CMDB itself to hold related data. And, with the boundary drawn at configuration items and their relationships, the question of whether to store some new type of data in the CMDB itself is already answered. If the data does not fit into your CMDB parameters, you can store it as part of the CMDB extended data layer, and save the trouble of changing the data model to accommodate the new information.

The CMDB extended data layer holds related data that may not be considered a key attribute.

Similarly, transactional data can be stored in databases better able to handle a high volume of requests, instead of in the CMDB. As a result, data is provided more efficiently. For example, data consumers (applications that view or modify data already in the CMDB) can get information from individual data stores that are optimized to provide the specific type of data being requested, knowing that the data is being pulled from the CMDB and is therefore up-to-date.

A federated data model eliminates the need for several data migrations and application integrations to move change requests, help desk tickets, and other configuration item data into the CMDB. Rather, applications that use this data can continue to access it where

5 KEY BENEFITS OF A FEDERATED CMDB

1 Provides significant business value by having a single source of record.

2 Provides a logical model of the IT infrastructure to identify, manage, and verify all configuration items in the environment.

3 Ensures data consistency and simplifies integration among the different service management processes.

4 Eliminates the need to modify the CMDB itself to hold related data.

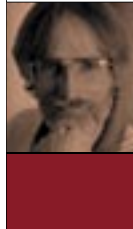
5 Provides data more efficiently because transactional data can be stored in databases better able to handle a high volume of requests.

it is currently stored. As a result, the CMDB becomes an enabler of information, and is not a bottleneck. The requests for related data are handled by other databases, so the CMDB does not have to accommodate all of the traffic related to processing that information. The load can be spread across multiple systems, reducing the potential for outages caused by changes. Although you could store your CMDB extended data in one place, you don't have to. The different types of data in this layer are not necessarily linked to or related to each other. The only thing they need to have in common is a link to or from the CMDB.

Together, the CMDB layer and the CMDB extended data layer form the CMDB environment. The CMDB environment is devoted to the applications that provide and consume data. These applications can access the CMDB, the CMDB extended data, or both. For example, an asset management application that views and modifies configuration item instances in the CMDB is part of this environment as a consumer of information. A discovery application that creates configuration item instances in the CMDB is part of this environment as a provider of information. An application is part of the CMDB environment, while its configuration-related data is part of the extended CMDB. The value of this layer is the ability to provide open access to data. This federated approach for a CMDB provides a single, common set of information on the configuration and

relationships between items in the IT environment in a manner that can be leveraged by ITIL processes. By using this federated data model, organizations can fully integrate critical service and infrastructure management applications and break down traditional functional silos that often exist within an IT organization and improve the management of their IT environment. ●

ABOUT THE AUTHOR



Doug Mueller is the chief technology officer of the BMC Service Management business unit at BMC Software and a co-founder of Remedy, now a part of BMC.

