

# WHAT IS A CMDB AND HOW TO OPTIMIZE IT FOR SERVICE DELIVERY



As a consultant I have seen how, in many organizations, the configuration management database (CMDB) is always one of two things – an ideal fantasy, or a real nightmare. From my experience, if you start early and clearly establish the foundation of what to include and a strategy for how to manage it, your CMDB can be a valuable tool for the business for years to come.

## What is a CMDB?

The CMDB is a database used to store configuration records, or configuration items (CIs), throughout their lifecycle. At a functional level, the CMDB provides a means of understanding the organization's critical assets and their relationships, such as information systems, parent sources or dependencies of assets, and the child relationships of assets.

To remain agile and proactive, organizations need their CMDB to take on a bigger role—it should allow them to understand different environments and make real-time decisions around infrastructure problems and changes. This matured CMDB can move from a repository used for manually tracking assets to a full configuration management system (CMS) capable of maintaining one or more CMDBs that in turn store attributes of CIs—relating to incidents, problems, changes, and releases, as well as a myriad of other types of data and tools—and their relationships to other CIs.

# The importance of a configuration management strategy

One challenge faced by every IT department is ensuring that the core services the company depends on stay up and running—and can recover quickly when they do go down. Outages and their time to resolution are directly impacted by the organization's configuration management strategy. By knowing what's present in your environment with your CMDB, you can control it, maintain it, and improve it. In fact, with the technology available today, you can not only account for all IT assets but also provide a sound basis for incident management, problem management, change management, and release management. This drives operating costs lower, reduces the number of reported issues, and facilitates faster issue identification.

A solid strategy understands that a CMDB isn't limited to storing CIs and their relationships. When you choose a CMDB strategy, you're often not limited in what you can store in CMDB extended data. Also, the types of CIs and relationships in your CMDB must be capable of changing, so you need a data model that is extensible. You should be able to add and remove attributes from your classes, and even add and remove classes or types of configuration items that can be stored in the CMDB. An object-oriented data model has a hierarchical set of classes where each class inherits attributes from the class above it in the hierarchy, called a superclass—and then adds its own attributes to create a more specific type of object, a subclass. Subclasses can have their own subclasses, extending the hierarchy to whatever level of detail you want to track. Though this feature is important, I would suggest that you should take care not to overuse it. The CMDB should hold only CIs and their relationships for critical configuration items. Adding classes and attributes for non-required CIs will needlessly tax your CMDB, which can negatively impact performance. Also, subclassing too far can leave you with classes so narrowly defined that they have very few members. Balance the need for categorization with the need to store similar CIs together.

## CMDB Best Practice in Action

To give you a better sense of how strategy and CMDB best practice can have positive impacts on your service management, I've summarized two customer examples below. Hopefully, from them you can see a path forward for your organization

### Customer example #1: Adjusting and Cleansing an Existing CMDB

Recently, we revamped and cleansed a financial service company's CMDB by adhering to CMDB best practices. These adjustments were necessary because the CMDB had data that was repetitive—appearing in multiple classes and subclasses. It also created a maintenance nightmare when these attributes needed to be modified. Initially, when they started using their CMDB, instead of grouping their CIs and organizing them in similar classes, they decided to create individual classes and subclasses for each type of asset. To repair this, we removed a few subclasses that were miscategorized because they existed as a subclass while the class already had most of the attributes for the subclass. We handled this by adding the required attributes in the parent class and re-architecting the CMDB strategy to utilize attributes that stored similar data. We could now remove the subclasses that were no longer necessary. This limited the number of custom attributes required to store similar information.

While we were making these adjustments, we also established procedures to prepare the CMDB for future growth in case new types of assets are added to it and to avoid creating multiple sub-classes

if it is not required. As part of this future planning, we helped the customer create and enable a team of dedicated CMDB Administrators who enforce the processes we defined for mapping new types of assets in the CMDB. These procedures included guidelines around the type of asset and the expected class mapping that can be used to store the information. This made it possible to plan for decommissioning one of the in-house tools, maintain asset ownership, improve cost center billing, and add efficiency all within the CMDB because this data was migrated and stored in the CMDB with existing integrations that updated it nightly. By making the CMDB the single source of truth for all assets, the customer is able to run Operations more efficiently. The customer not only reduced their spend by decommissioning the in-house tool but also created a robust CMDB. With it, they can control change lifecycles to mitigate risks to their IT services. With the ability to perform impact assessments for any change against the infrastructure, they can determine the true impact to the assets.

With the CMDB in place, the potential risk of downtime and asset unavailability is easily visible, enabling the Change Advisory Board to make more informed decisions around actions like delaying change requests that lack backout plans and outage windows. The result is fewer outages and incidents.

## **Customer example #2: Utilizing the CMDB to Support Asset and Application Licensing**

An oil & gas customer had a home-grown tracking tool they were using for Asset and Application Licensing. The tool was designed to be a repository of assets that were tracked manually. Application licenses were manually tracked based on installation requests from end users. This is how the customer described the challenging situation they found themselves in:

*Visibility into our IT infrastructure was limited because asset data was scattered across the organization. Some of the data was stored in spreadsheets but often it existed only in the heads of veteran staffers. We didn't have a clear picture of what resources we had, where they were located, who owned them, and how they related to each other. Consequently, we couldn't follow best practices for managing incidents, identifying and resolving recurring problems, or rolling out changes quickly and reliably. Nor could we readily assess the risks that a change might have on the rest of the environment.*

In this situation, we first transitioned the customer to a CMDB (specifically [BMC Helix CMDB](#)) and utilized [CMDB best practices](#). Once we completed populating the CMDB, the oil & gas company quickly saw the value that provided around ownership tracking, cost center billing, asset tracking (including applications installed on assets, like servers, desktops, laptops, and smartphones), asset support, and asset lifecycle history.

In addition to an improved CMDB, the oil & gas company needed to expand their use of the CMDB to reduce risks presented by software licenses that aren't centrally tracked by automating notifications of expirations. To achieve a system that could also give them visibility into handling their software license agreements and license models, we recommended connecting their CMDB to Flexera FlexNet Manager Suite, a BMC strategic partner's tool. We made this recommendation because it would add the following critical capabilities leveraging the CMDB and thus ensure the oil & gas company knew what software license agreements they had, who owned them, and when they were up for renewal in order to remain compliant and reduce the risk of incurring fines and other issues:

- Automated discovery and analysis of the application across all assets
- License repository tracking and updates based on usage
- Increased license compliance to reduce software audit risk
- End-to-end software lifecycle management
- Reporting for consolidation and vendor management

They gained software license optimization maturity by pairing Flexera with [BMC Helix Discovery](#) and [BMC Helix CMDB](#) for a framework to store the information and create relationships with existing CIs. This reduced the number of instances where they might be out of compliance or over-subscribed to an application licensing agreement.

As you can see, a well-planned and implemented CMDB enables your IT team to manage critical IT business processes with ease. Even further, with a strategy in place to mature your CMDB, it will make it possible to run and reinvent your IT. Even further, by combining your CMDB strategy with complementary software license management and a comprehensive asset management solution, you can optimize all aspects of your hardware, software, and network resources to streamline IT and business operations, allowing you to deliver solutions for key IT processes. The dream of providing the best quality data to your business via a KPI-driven user experience is possible with CMDB best practices. With a mature CMDB in place, just think about the benefits you and your organization can achieve with all the information now at your disposal whenever you need it.

If you can use assistance with your CMDB strategy, [contact BMC Customer Success](#) and speak to one of our experts.