



Asset Management, ITIL®, and the CMDB: Connecting the Dots between IT Operations and the Bottom Line

Table of Contents

- INTRODUCTION** 1

- SECTION 1**
 - Wading Through Terminology.....2

- SECTION 2**
 - Contrasts: Same Data, Different Goals.....2

- SECTION 3**
 - Comparisons: Sharing a CMDB.....3

- SECTION 4**
 - Scalability and Process Integration through Federation.....4

- SECTION 5**
 - Operationally Integrated ITAM6

- SECTION 6**
 - Which IT Organization Should Maintain the CMDB?.....6

- SECTION 7**
 - What To Look for in a CMDB6

- CONCLUSION**
 - The Bottom-line Advantage.....7

Introduction

Since the 1990s, IT organizations that embarked on enterprise IT Asset Management programs gave asset managers a unique mandate: to reach across technical and organizational silos in order to account for and manage all IT assets in the enterprise. Some data about IT assets would always stay inside specialized tools for application development, UNIX or WinTel administration, network management, and the like. But as IT became a larger and more mature part of the business, the need for a cross-functional, cross-platform approach to fundamental business issues, such as IT procurement, cost accounting, and compliance, became increasingly apparent. And thus, so did the need for a repository of enterprise assets.

Now, the growing popularity and maturity of IT Infrastructure Library (ITIL)-driven initiatives introduce a similar mandate: to account for all IT components (Configuration Items or “CIs”) and their inter-relationships in a Configuration Management Database (CMDB). In ITIL terminology, this discipline of establishing and maintaining a CMDB is called Configuration Management.

To date, ITIL has not been particularly clear on the role of IT Asset Management (ITAM) and its fit with ITIL processes like Configuration Management. Do ITAM and Configuration Management have redundant mandates to account for IT components, resulting in an asset repository in one organization and a totally separate CMDB in another? How does an asset repository differ from a CMDB, or an asset from a CI? Is much of ITAM’s role becoming subsumed by ITIL?

IT executives will be encouraged to learn that, far from creating a redundancy, IT Asset Management and ITIL Configuration Management strategically complement one another, each providing different but synergistic benefits while leveraging many of the same investments in software tools, implementation, and staff. And IT asset managers should be relieved to discover that, far from encroaching on their turf, a CMDB will empower them to play an even more strategic and integrated role in IT.

Through a CMDB, the best practices of ITAM for costs, contracts, usage, and compliance can be efficiently woven into both daily and strategic IT operations, connecting the dots between IT operations and the bottom line.

Wading Through Terminology

Before we look at how a CMDB, Configuration Management, and ITAM can work together to deliver new levels of efficiency, it is helpful to first clarify what we mean by these terms.

The term “IT Asset Management” is subject to broad interpretation, creating a semantic trap where almost anything in IT can be called an “asset,” and anything done to it is “management.” For example, a server is certainly an IT asset. Proper configuration of a server can be said to be part of managing that server. But if a server needs to be reconfigured with an application or OS patch to stop a memory leak, is this an “Asset Management” function? Most (but perhaps not all) organizations would agree that patching servers falls well outside the responsibility and expected skill set of ITAM.

What if you use a discovery tool to see what you have deployed in your IT environment? Is this “Asset Management?” The answer depends on what you will do with this discovered data. Asset Management relies on discovery data as a starting point, but requires additional strategies, processes, and data to manage discovered entities as assets. When you use the discovered data to manage software licenses, track leases, or reconcile invoices, then you are performing Asset Management. Since many discovery tools are marketed by vendors as “Asset Management solutions,” it is no wonder that a single tool (discovery) is often confused with the broader discipline of IT Asset Management.

Many disciplines other than IT Asset Management rely on discovered data — from change and incident management to network and datacenter operations. Indeed, many companies use “asset repositories” either partly or solely to support service desk operations, such as incident, problem, and change management. For example, a database containing information about which assets are used by which employees is helpful to resolve incidents more quickly, eliminating the game of “20 questions” between the technician and user. In this way, troubleshooting can begin with knowledge of the exact make, model, version, and configuration of desktop, printer, or application that is creating the incident. But this is not Asset Management.

Admittedly, all definitions of IT processes (or any business discipline for that matter) are imperfect, but for the purposes of this discussion, let’s stipulate a working definition:

IT Asset Management is the discipline of managing finances, contracts, and usage of IT assets throughout their lifecycles for the purpose of maintaining an optimal balance between business service requirements, total costs, budget predictability, and contractual and regulatory compliance. Traditional ITAM activities include the management of inventory, software licenses, vendors, procurement, leases, warranties, cost accounting, retirement, and disposal.

Thankfully, ITIL’s Configuration Management is easier to describe since ITIL is responsible for popularizing the concept and because its mission includes forging agreement on terminology. To summarize ITIL:

The goal of Configuration Management is to provide a logical model of the IT infrastructure that is accessed by all ITIL processes to drive consistency among them. Activities include identifying, controlling, maintaining, and verifying the versions of configuration items (CIs). This CI information is to be stored in a single repository: the Configuration Management Database (CMDB).

With these definitions and descriptions in place, it’s easier to draw both contrasts and comparisons between ITAM and ITIL’s Configuration Management.

Contrasts: Same Data, Different Goals

While ITAM tends to be focused on financial and administrative aspects of IT components (assets), Configuration Management is more concerned with supporting operational concerns, such as stability, availability, and quality of the business services IT delivers and supports. The understanding of these two different sets of goals is crucial to answering the frequently asked question, “What is the difference between a CI and an asset?”

Whether a given IT component is considered an “asset” or “CI” – or both – is determined by what you do (or plan to do) with the data about that component. A component should be considered an “asset” if you decide it is worth managing its contract, cost, and usage attributes throughout its lifecycle. A component is deemed a “CI” if you decide it is worth managing operationally for incidents, problems, changes, releases, capacity, etc. The same component can be both an asset and CI if you decide to manage it for both administrative and operational purposes.

Following is a further look at how the disciplines of IT Asset Management and ITIL Configuration Management compare:

IT Asset Management	ITIL Configuration Management
Goals: Manage asset costs, contracts, and usage/ownership throughout lifecycles.	Goals: Provide logical model of IT environment in a CMDB as basis for ITIL processes.
Value: Lower asset TCO/acquisition costs, reduced purchasing, more efficient allocation, more accurate budgeting/planning.	Value: Greater business service stability, availability, quality (via ITIL processes that rely on configuration data).
Organization: Can be part of IT operations, typically growing out of service desk organization, but often traditionally aligned with IT Finance and/or Procurement.	Organization: Is almost always in Operations, typically growing out of service desk or data center organizations, or both.
Asset: Physical IT component tracked based on value, contractual compliance. A CI can be an asset if it is worth tracking for cost, contract, and usage. Assets not likely to be managed as CIs because they have little or no operational impact include bulk items like toner cartridges, assets on order (not yet received).	Configuration Item (CI): Physical or logical IT component managed for its operational impact. An asset can be a CI if it is worth managing for operational stability. CIs not likely to be managed as assets because they have little or no administrative impact include a custom Java component, a business process document, a business service model. Etc.
Relationship tracking: Basic relationships (peer, parent, child) between assets are maintained for retirement process, ownership, software license matching.	Relationship tracking: Sophisticated relationships between CIs are maintained to assess change risk, analyze root cause of problems, and assess service impact.
Versioning: An asset should be tracked and managed according to its lifecycle state (e.g. on order, awaiting deployment, deployed, under maintenance, and retired). For proper lease, retirement, and cost accounting, it should also offer comparisons between original and modified states (e.g. added applications or storage components to a server).	Versioning: ITIL processes, such as Change Management, also require CI status, including past, future, and desired states of IT infrastructure configuration, for proper planning, risk assessment, remediation, and rollback. In ITIL terms, this is part of CI Status Accounting, which also includes ITAM-type status, such as ordered, received, under repair, and for disposal.

Now that we've untangled ITAM from ITIL Configuration Management to make their distinctions more clear, let's see how they can fit back together to complement one another.

Comparisons: Sharing a CMDB

Both ITAM and ITIL Configuration Management share the need for reliable data about components in the IT environment. Thus, discovery tools (a scalable means of keeping accurate data on deployed IT components) and a CMDB (a repository for reconciling and accessing the discovered data) can serve both ITAM and Configuration Management. With ITAM, discovery and a repository are just enabling means to other ends; they are the beginning of a process. With Configuration Management, accurate recording of the IT environment's configuration is the end goal itself, with the important understanding that other ITIL processes (Incident or Change Management, for example) will then use that data and derive value from it. Put another way: IT Asset Management – just like Change, Incident and Capacity Management – can be just another process consuming and contributing data to the CMDB, creating value by taking actions based in part on CMDB data.

While it is possible to use the same discovery tools to populate separate repositories (an asset repository for ITAM, a CMDB for Configuration Management), this would not only create redundant efforts, it would also deny a golden opportunity to connect ITAM to other ITIL processes. Because there is a significant overlap between what you will deem worth tracking as CIs and as assets,

you can benefit from a unified approach to IT component tracking, whereby objects in the CMDB can be managed as both CIs and assets — without duplication of investments and integrations between separate repositories.

But as we have already seen, there is not 100 percent overlap between assets and CIs. And more importantly, different kinds of data must be related to the IT components depending on how they are being managed (e.g. costs and contracts for ITAM, relationships and configuration detail for ITIL). So the question arises, what data belongs in the CMDB?

First, free yourself from the daunting task and paralyzing thought of starting a CMDB implementation where every discoverable attribute of every component and relationship in your enterprise must be catalogued in the CMDB. Instead, use a more pragmatic approach by intelligently applying the following principle: Populate the CMDB with the minimum amount of data needed for maximum value. This requires that you start with a clear understanding of your business objectives, followed by an analysis of what processes need to interact with which data to achieve those objectives. So long as your CMDB is architected for growth over time in both breadth (number of CIs and relationships) and depth (attributes of CIs and relationships), you can make decisions incrementally on what CIs and assets to manage and at what level of detail, gradually adding more as your processes mature and expand.

For example, you may decide to start CMDB population based on an IT Asset Management project with desktops and servers, or your change management effort with only servers, or your service impact initiative with CI relationships within a single critical business application. As your real-world experience makes you more comfortable with the level of CI attribute and relationship detail needed in the CMDB to support your core asset, change, or service impact requirements, you can expand to more manage more assets, CIs, and relationships at the appropriate level of detail.

To maintain CMDB scalability and focus on the IT production environment, assets not in the production environment (e.g. a network switch on order from a vendor, or toner cartridges in a stock room) need not be entered into the core CMDB. These asset records can be stored in a logically (if not physically) separate asset repository, which should ideally share the CMDB's data model to facilitate easy migration of asset inventory records to the production CMDB.

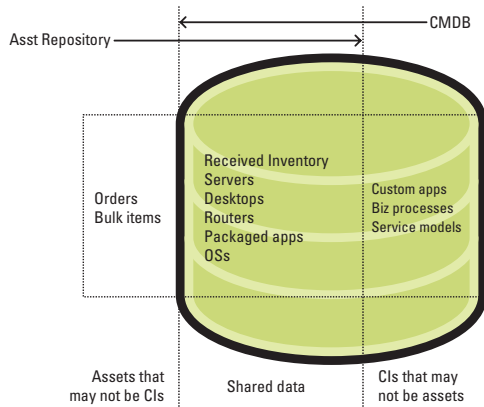


Figure 1. A CMDB can provide almost all of the same IT component data as an asset repository.

So why not just use the CMDB for storing all assets and management data, including bulk items, software licenses, purchase and lease records, and the like? If ITAM was the only discipline that ever needed to use the CMDB, this would be feasible. But the whole point of a CMDB is that it be shared across IT disciplines and their tools. How can every IT discipline associate its own unique management data to CIs in the CMDB without bloating the core CMDB itself?

Scalability and Process Integration through Federation

A large repository of assets or CIs is, by itself, of relatively limited value. Without automated processes or tools connected to the CMDB, anything other than reporting on CIs only would require manual processes. As more processes and tools are connected to the CMDB, greater value is derived from the CMDB.

Not only do a variety of tools need to connect to a CMDB in order to realize the value of ITIL Configuration Management, they must also associate their unique management data to CIs in the CMDB. ITAM needs automated workflows that can relate software licenses, leases, and purchase records to CIs. Change management needs automated processes that can relate a request for change, change task, or schedule to CIs. Software configuration management needs to relate detailed settings data to understand the readiness of a CI to accept a software release. Service impact management needs to relate real-time event data to a CI's relationships with other CIs to interpret the events' impact and priority to the business.

So where does all this varied "management" data belong? Can we put all software licenses, leases, purchase records, incidents, changes, registry settings, real-time events, and more into a single CMDB?

Even if database software and hardware were infinitely scalable, the likelihood that you will get all IT functions to agree on a single data model for storing everything from leases to real-time events in a single database – and getting all tool vendors and homegrown applications to write to this single intergalactic data model – is about as likely as the "paperless office." Indeed, if you've heard of companies who have tried and failed at a CMDB project in previous years, it is likely because the scope of data included in the proposed CMDB was so large that the project collapsed under its own weight.

The path of least resistance is to leave unique management data within the specialized tools and repositories that serve each discipline, and then to let each tool "freeload" from the CMDB — reading and perhaps writing CI data, but never contributing any of its own external but related management data to other tools and processes. While this model of multiple tools all freeloading from a centralized CI repository is an improvement over today's fractured views of the IT environment, it misses the bigger promise of a CMDB: to use a single source of truth on IT environment configuration as a means of connecting otherwise siloed IT processes. In short, leaving the management data in place is a good answer, but freeloading is not.

To strike the proper balance between the scalability of a CMDB and the need for different tools to share their specialized management data with other processes, we can use a federated approach. Through federation, a limited amount of information identifying the key attributes of CIs and their relationships is stored in the CMDB, while more detailed, specialized and/or highly dynamic management data pertaining to those CIs are stored separately. By sharing the same CIs as reference points, different applications and tools are better enabled to share management data with one another.

In the ideal case, the various management tools (e.g. asset, change, incident, and problem tools) will share a common platform that provides automated cross-functional workflows together with seamless sharing of data within a single user interface. This allows an asset manager, for example, to see incidents against a particular CI or entire class of CIs without leaving the IT Asset Management application, or a service desk technician to see the warranty status of a CI for which an incident or problem has been reported without leaving the help desk application.

When management tools are on different platforms, other means of federated data-sharing can also be effective. Links to various applications connected to the CMDB can enable a user of one application to open up a view in an entirely different application within the context of the CI(s) in question.

In a more robust implementation, you can use federation as the basis for application integration. For example, to provision an application to a new user, a change request issued from one application may automatically grant new user access privileges by invoking an Identity Management application on an entirely different platform. Although sharing CI data doesn't take care of the entire process integration between Change Management and Identity Management applications, this automated interaction is possible only if both applications first share a common understanding of which applications and underlying infrastructure are involved.

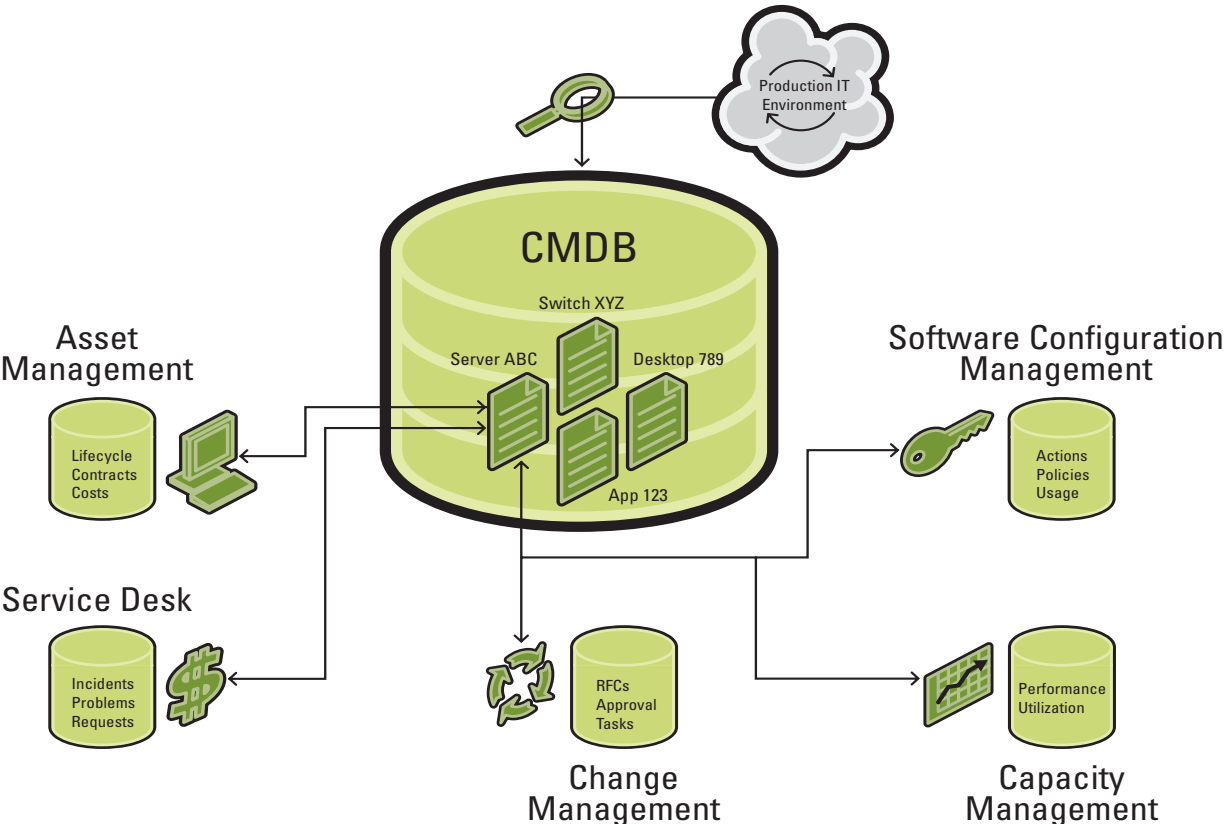


Figure 2. A CI, "Server ABC" in the CMDB, acts as a common reference point for multiple tools to share their federated management data.

Operationally Integrated ITAM

By using a CMDB as its primary asset repository and federation to relate to other processes, IT Asset Management can benefit from the same process consistency and synergy that the CMDB has always promised to the core ITIL disciplines, such as Change, Incident, Problem, Release, and Capacity Management. Where ITAM and ITIL processes share the same view of IT components and relationships, they now have a set of common reference points to connect automated processes and to share management data within those processes.

Consider:

- > A record of poor performance and/or high total cost of ownership for a particular server type and configuration should be visible to asset managers for negotiations with server vendors. Incidents, problems, and capacity data can be synchronized to the vendors and procurement contracts that supplied those servers if they are all related to the same CI records in the CMDB.
- > A server asset's application usage, retirement status, and depreciated value can be factored into a capacity manager's server consolidation plans.
- > A change request requiring a significant quantity of new software deployments should include a task of assessing the impact on software licenses. This can be largely automated if both the change request and software license records are related to the same IT components (i.e. CIs in the CMDB).
- > A service desk technician shouldn't waste time troubleshooting a user's unlicensed software or repairing a server that's under warranty. An incident opened against a CI in the CMDB will correspond to the warranty records that IT Asset Management maintains against the same CI record.

Which IT Organization Should Maintain the CMDB?

It should be clear by now that a well-maintained CMDB functions as a strategic part of many IT disciplines. So who should own the process for maintaining the CMDB? There is no single answer, as there are many different ways in which IT is organized. The answer may depend on which ITIL process will be the first focus for the use of the CMDB or which technology or vendor platform the CMDB will be built upon. Or, it may be as practical as which IT organization has the budget or staff resources at a particular point in time.

Rather than go into a lengthy discussion on the pros and cons of network operations vs. service desk ownership, let

us consider an obvious candidate for who should at least be intimately involved in CMDB maintenance: IT Asset Management. If an IT Asset Management program is already in place, chances are this staff already possesses the skills to manage a large repository of IT components, including how to work best with discovery tools, how to reconcile data, and how to work across multiple technology and organizational silos. Whether or not ITAM is the answer to CMDB maintenance in your enterprise, it certainly should be part of the answer.

What To Look for in a CMDB

It is often said that ITIL is descriptive, not prescriptive. Perhaps nowhere is this more evident than with the CMDB. While ITIL describes what goes into it and what processes should interact with it, ITIL does not specify what automated functionality a CMDB must provide. A pad of paper (very thick) and a pencil (preferably mechanical with a case of lead refills) could qualify as a CMDB if CIs and their relationships were duly noted, and all ITIL process practitioners dutifully consulted the pad of paper to report incidents or plan a change.

So it should come as no surprise that many software vendors claim to have a CMDB. Some even claim thousands of deployments dating back years before ITIL had any significant traction. Most are asset repositories with a new name. Although the CMDB can function as an asset repository, it must accomplish much more:

- > Relationships are a key difference between a simple database of assets and a CMDB. The CMDB is a logical model of the IT production environment, documenting how infrastructure is configured to provide business services. A CMDB should model CI relationships to the degree needed to assess change risk, service impact, root cause analysis, etc. in a business context. The simple parent/child/peer model found in most asset repositories is insufficient for these functions.
- > A CMDB stores "configuration items" (CIs), which include not just physical assets, but also logical entities like services and business processes. A CMDB must be able to store a business service model (e.g. all the infrastructure and users and documentation) involved in delivery of a service, such as "supply chain order management", as a CI.
- > Because of the breadth of data a CMDB must ultimately handle, it will likely be fed by multiple discovery tools and external repositories. It therefore should have an efficient, repeatable mechanism for reconciling data from multiple discovery sources, including rules for when to merge certain CI attributes from one data source with different attributes for the same CIs from another data source.

- > To realize its intended value, a CMDB must connect to a variety of operational tools and processes for incident, problem, change, and other important functions. Open interfaces are a must, but ready integrations within the workflows of automated tools will provide more immediate and deeper value.
- > To provide ready access to related management data, a CMDB should provide a mechanism for efficient federation to other tools and data stores that can share management data with one another when related to a given CI.

The Bottom-line Advantage

Enterprises evaluating different IT Asset Management approaches should strongly consider how they can use a CMDB to better align themselves with IT operations. By sharing federated management data with ITIL-compatible tools and processes, the ITAM function gains greater asset data accuracy, better enforce software license compliance, uncover hidden asset costs, and gain valuable insights into asset performance.

Enterprises approaching a CMDB initiative from an ITIL or operations orientation should take a close look at the advantages of adding (and even beginning with) automated IT Asset Management controls to those CIs worth managing as assets. IT Asset Management can leverage the investment made in operational CI data to get relatively fast, measurable business value such as consolidating or redeploying underused servers rather than procuring new ones, reducing lease penalties, or renegotiating software licenses with vendors whose software is being underused.

Over time, operationally integrated IT Asset Management helps IT organizations continuously optimize the asset portfolio against business requirements, thus connecting the dots between IT operations and the bottom line.



ACTIVATE BUSINESS WITH THE POWER OF IT.™

About BMC Software

BMC Software helps IT organizations drive greater business value through better management of technology. Our industry-leading Business Service Management solutions ensure that everything IT does is prioritized according to business impact, so IT can proactively address business requirements to lower costs, drive revenue and mitigate risk. BMC solutions share BMC Atrium™ technologies to enable IT to manage across the complexity of diverse systems and processes—from mainframe to distributed, databases to applications, service to security. Founded in 1980, BMC has offices worldwide and fiscal 2005 revenues of more than \$1.46 billion. BMC Software. Activate your business with the power of IT. For more information, visit www.bmc.com.

