ITIL® Asset and Configuration Management

New for 2019: Access the ITIL 4 Guide on BMC Blogs ›

ITIL asset and configuration management

IT services are typically made up of a bunch of individual components – things like servers, software and middleware, and unique configuration information. In ITIL, Service Asset and Configuration Management, or SACM, is about properly planning and managing (and even being able to report and audit) the relationships and attributes of all of these components, across every service in your infrastructure.

In other words, SACM is a combination of two important processes:

1. Asset management, which addresses the assets you use to deliver IT services
and

2. Configuration management, which tracks the configurations of and relationships between the various components (or configuration items) of your various IT services

BMC Helix: Next Generation ITSM

According to ITIL, SACM is:

The process responsible for ensuring that the assets required to deliver services are properly controlled, and that accurate and reliable information about those assets is available when and where it is needed. This information includes details of how the assets have been configured and the relationships between assets.

It’s a critical practice, and one that spans the entire service lifecycle. Nailing SACM is important to the health of both your individual services and your entire IT organization – and as such, it’s often one of the first ITIL processes implemented in top IT organizations.

Top objectives of service asset and configuration management

At its core, SACM is about ensuring that you are able to identify and control all assets across your infrastructure, and can manage their integrity through effective recording, reporting, and auditing.

More specifically, ITIL dictates that the objectives of SACM are to:

- Ensure that assets under the control of the IT organization are identified, controlled and properly
cared for throughout their lifecycle.

- Identify, control, record, report, audit and verify services and other configuration items (CIs), including versions, baselines, constituent components, their attributes and relationships.
- Account for, manage and protect the integrity of CIs through the service lifecycle by working with change management to ensure that only authorized components are used and only authorized changes are made.
- Ensure the integrity of CIs and configurations required to control the services by establishing and maintaining an accurate and complete configuration management system (CMS).
- Maintain accurate configuration information on the historical, planned and current state of services and other CIs.
- Support efficient and effective service management processes by providing accurate configuration information to enable people to make decisions at the right time — for example, to authorize changes and releases, or to resolve incidents and problems.

A few key definitions

Before we dive even deeper, it’s helpful to get a few definitions out of the way that you’ll see regularly throughout this process and beyond. There are a ton of definitions, but we’ll limit it to the most important ones for a basic functional understanding of the ITIL Asset and Configuration Management.

- We’ll start with Configuration Management System (or CMS). A Configuration Management System is basically just a set of tools (like databases, files, etc.) that are used to gather, update, and analyze data about all of your configuration items and their relationships. A CMS may also include information about incidents,
problems, known errors, changes, and releases — and even link to corporate data about employees, suppliers, locations and business units, customers and users.

- That’s different from a **Configuration Management Database (or CMDB)**, which is a database that stores configuration records. One or more CMDB’s may be part of the overarching CMS.

- **Configuration records**, then, are just records about attributes (like name, location, version number, etc.) and relationships of your CIs. Basically, they are records that describe your CIs.

- A **Configuration Item (or CI)** is simply any component that needs to be managed in order to deliver an IT service. A server, a virtual server, or even the configuration of an application could be considered a CI, for example. There are quite a few different types of CIs, including:

  - Service lifecycle CIs such as business cases, service management plans, service lifecycle plans, service design package, release and change plans and test plans, etc.
  - Organizational CIs (like the organization’s business strategy, regulatory requirements that need to be managed, etc.)
  - External CIs such as external customer requirements and agreements, releases from suppliers or sub-contractors and external services.
  - Interface CIs that are required to deliver the end-to-end service across a service provider interface (SPI) — like an escalation document that specifies how incidents will be transferred between two service providers.

- **Service assets** can be any resources or capabilities that contribute to the delivery of a service. Unlike CI’s
above, which can be specifically managed by IT, service assets can’t always. The knowledge of an IT worker, for example, is a service asset. The overall ability for a team to respond to an incident or problem is a service asset.

The scope and benefits of service asset management and configuration

If it’s an asset that you use during the Service Lifecycle, it typically falls under the scope of SACM. But there are a few exceptions.

Service assets that are not CIs, like the knowledge used by an experienced service desk agent to manage incidents, for example, are excluded from the scope. So are assets that are not under the control of change management (like information stored on a server), and non-IT assets in general. Everything else is fair game, including virtual assets.

In general, the scope of SACM includes management of the complete lifecycle of every CI — including interfaces to internal and external service providers where there are assets and configuration items that need to be controlled.

It’s incredibly valuable to the business, too. Benefits include:

1. Better cost management of services
2. Improved planning and delivery of changes and releases
3. More efficient resolution of incidents and problems, meeting SLAs more frequently
4. Less risk of non-compliance to important legal, regulatory, and procedural standards

Along the way, service asset and configuration management will interface with nearly all of your other IT process, including change management, financial management, incident and problem
Key activities of SACM

1. Planning
   For each of the services you offer, ITIL suggests that you create a Service Management Plan — which is essentially just a document that addresses the critical components of a service before you implement it. It typically covers the scope and objective of a service, the activities and procedures (and even roles and people) required, the relationship with other processes, and the tools and CIs also involved. Service Management Plans typically go into quite a bit more detail, though — outlining explicitly how the change management and configuration management processes will interface, how and when CI data will be audited, and much more.

2. Identification
   This is essentially creating a complete “inventory” of all the CI’s in your infrastructure. In this activity, you are essentially recording every bit of data about your CIs that is necessary for effective operations — from the version of a piece of hardware or software to all the documentation, configurations, ownership details, etc. To do so, ITIL recommends that you give all CIs unique identifiers (for example, numbers), as well as record all the relevant attributes of the CI (including the owner). Potential attributes you may want to capture include:

   - The unique CI identifier and CI type. Every CI should have a unique identification number that makes it easily identifiable.
   - The name and description of the CI
   - Version numbers, since multiple versions of the same CI often exist
   - Location and owner information, so you know where
to find it

- Current status (ordered, in production, etc.)
- When necessary, supplier information, related documentation, etc.

3. **Control**

Here, ITIL recommends that all CIs follow a strict process for being added, changed, and removed from your CMS or CMDB. The goal is to ensure that changes of any kind don’t occur without following your approved procedures for a wide variety of processes like license management, change management, version management, and even deployment. Along the way, you will need to create your own policies and procedures for things like controlling software licenses (to avoid noncompliance and prevent financial waste), controlling access to facilities and systems, and how you will properly capture the baseline of your assets and CIs before releases, to give you an accurate way to verify the success of actual deployments.

4. **Status accounting and reporting**

Throughout the lifecycle of every CI, ITIL encourages you to keep track of the complete status – including what changes have been proposed, the status of approved changes, etc. Being able to view and provide status reports gives you important insight into both the current and historical state of your CI’s, and can even help you detect unauthorized CI’s along the way. Reports typically include things like:

- An inventory of CIs and their baseline configurations
- An itemization of any unauthorized CIs
- Updates on recent changes or exceptions
- An itemization of hardware and software assets

5. **Verification and audit**
At any given time, you need assurance that your data is accurate on all of your CIs. Regular reviews and audits are essential, and ITIL recommends that you perform them to prevent discrepancies between your actual environment and how it is documented. In ITIL terminology, verification is an ongoing activity, consistently ensuring that the CMDB accurately reflects all of your CIs. An audit, on the other hand, is a more formal, occasional deep dive to confirm not only that records are accurate, but that processes are being followed and standards (including SLAs, etc.) are being met. Verifications and audits can be performed anytime, both randomly and according to a planned schedule. Tools are available to automate the process, too — comparing the configuration of designated servers with the master configuration you’ve recorded, for example. Audits are also often performed before major changes or releases are deployed, to avoid potential incidents or service disruptions.

6. Managing information
Ensuring the integrity of your configuration and asset data and systems is equally important. As part of the asset and configuration management process, you need to regularly back up the CMS, keep detailed records about archived and historical CI versions, and take appropriate measures to ensure data integrity across the entire lifecycle.

Key recommendations
First, always ensure that changes to every configuration item are authorized by change management, and that all changes and updates also modify the relevant configuration records accordingly.

Also, ensure that you have the right checks and balances in place to prevent unauthorized personnel from moving hardware
assets, or making changes to any assets or CIs. When this happens, the CMS becomes out of date — and it’s important that your records stay accurate.

And finally, be as vigilant as possible about performing regular verifications and occasional audits, too. The goal is to prevent the accuracy of your configuration records, etc. from diminishing over time — and that can happen when you stop paying attention to the process and the results.

**BMC Helix: Next Generation ITSM**

*BMC Helix ITSM* combines the latest in digital and cognitive automation technologies to enable best-practice ITSM principles, helping you to provide intelligent and predictive service management across any environment. Learn more about **BMC Helix ITSM**

- Optimized for **ITIL³ 4**
- Predictive service management through auto-classification, assignment, and routing of incidents
- Integrations with leading agile DevOps tools such as Jira
- Delivered in containers to enable operational and cloud deployment efficiencies