

# ADVANCED CONFIGURATION AND ASSET MANAGEMENT



## DERIVING REAL BUSINESS VALUE FROM A CENTRALISED DATABASE

# White Paper

Written by Keith Inight

## Executive Summary

Configuration and Asset Management is not a subject to attract immediate enthusiasm from executives. Why is this? Probably because they have other business critical issues on their agenda. Or is it because – plainly stated – Configuration and Asset Management is all about collecting and storing data in a database?

This subject is often perceived as boring – an inevitable cost without any very obvious benefits. But that perception is quite simply wrong. There is tremendous business value to be realised from this data (and severe consequences if it is not kept). The key to achieving that value is focus – to ensure that every piece of data is required for a clearly understood business reason. Data is then not a passive burden – it is an active business value driver. And failure to keep accurate and up-to-date data is easily associated with business consequences. This data must be accurately maintained in an increasingly dynamic environment and to succeed it must be backed by a new generation of sophisticated tools.

This paper reviews the business drivers behind Configuration and Asset Management, how they are evolving in importance and sophistication and the issues they generate. It then goes on to define Best Practice and look at the emerging solutions and the practical steps that can be taken to achieve it.

## Contents

Database uses	2
Why are configuration and asset management important?	3
The job just got tougher	6
Data Management can be very costly	7
The future is federated and automated	8
Automated tools emerge	11
Data lifecycle management	17
Implementation	18

# 1 DATABASE USES

The uses of the database can be grouped as follows:

## 1.1 Compliance

There are few areas of modern business untouched by compliance. Crucially, it is not enough to have processes that are compliant – it is necessary to have an audit trail to show that they were followed. This necessity leads to areas where a properly configured and managed database overcomes a variety of potential problems and failures. For example:-

### 1.1.1 Software licensing

Failure to abide by licensing terms can be expensive. Unfortunately there are no standards for how software will be licensed<sup>1</sup>. So the task of documenting which software is loaded on which machines and then matching these documents with license records is very onerous. The move to virtualised infrastructure makes this even more complex.

### 1.1.2 Security

Corporate IT needs to be able to demonstrate that it has taken all reasonable steps to secure the IT infrastructure against threats. In practice this means being able to show that specific versions of software (patches, virus signatures, application code) are present on specific machines. It is not enough to show that the software was distributed at some time in the past – somebody may have deleted it - it is necessary to show the current status (which, by definition, constantly changes in an active business environment.)

### 1.1.3 Business control

Here's a common business control scenario. The IT department is meeting all of its service levels, yet users are unhappy. When this is the case (as it often is) what's going on? It's usually because IT departments measure things that have relevance to them – such as servers and networks. But business managers have a different agenda; they need to be able to demonstrate that the infrastructure which supports their particular business process is working properly – so their staff are able to do their jobs. Tools to draw a dashboard are readily available – but they are powerless unless there is good quality data showing which pieces of infrastructure affect which business processes.

### 1.1.4 Reporting

Businesses are required to provide a full report of their assets. One example is a financial assets register. But it is common for these reports to be treated and commissioned as a tactical activity. As a result, they are often performed manually, accruing high labour costs, and deliver only poor quality data. A lose, lose, lose situation.

## 1.2 Delivering ITIL

The IT Infrastructure Library (ITIL) has become the de facto standard for how infrastructure should be managed. It has been adopted by the standards bodies as BS15000 and ISO 20000. The maintenance of a Configuration Management Database (CMDB) is pivotal to the delivery of ITIL. Key benefits include:

### 1.2.1 Change management

It has been estimated that 90% of incidents are unintended consequences of planned changes. Put another way, the impact analysis didn't work. Organisations typically respond to this failure by setting up complex Change Advisory Boards (CABs) to sign off changes. Cynics comment that this is more about spreading the blame than better impact analysis. So cynics take note – a sophisticated centrally managed database offers the potential to provide automated impact analysis and significantly reduce the number of incidents. It will also reduce the workload of the CABs and thereby implement changes much faster.

### 1.2.2 Incident and problem management

The first two questions that support staff will ask when faced with an incident are, "Where are all the details about the infrastructure affected?", and, "Who changed what, when?" A good CMDB, with an audit trail, is an invaluable aid to answering these questions and solving related issues.

## 1.3 Business alignment of IT

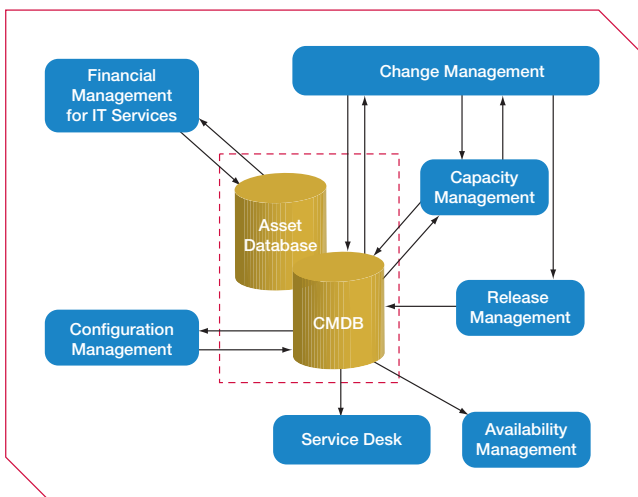
This is both an old dream and a cliché – and it is still as important as ever. CIOs want to be able to continuously demonstrate the link between specific pieces of infrastructure and the costs incurred by their business for the services provided. Credible recharging improves asset utilisation by giving business leaders what they need to minimise costs.

<sup>1</sup> Part 2 of ISO 19770 currently in draft seeks to address this

## 2 WHY ARE CONFIGURATION AND ASSET MANAGEMENT IMPORTANT?

An analogy is helpful. Think of a factory full of machine tools. To manage the factory efficiently it is clearly necessary to have an up-to-date list of machine tools and their capabilities. It is also necessary to know which operators are qualified to use which machines and when they are working on those machines. This information can then be used to schedule the flow of work through the factory. If any of this information is inaccurate or missing there will inevitably be disruption and potentially chaos in the factory. This may seem to be stating the obvious. But the collection and maintenance of the equivalent IT information in a business is often seen as a chore rather than an integral part of the production process.

Figure 1: ITIL processes



ITIL recognises the importance of this information and places Configuration Management at the heart of its methodology. As the high-level diagram above shows, most of the ITIL processes depend upon this data for their effectiveness.

Before proceeding further, it is worth defining what a CMDB is, and how it differs from an Asset Inventory and an Asset Management Database. Whilst very different in their objectives and scope, they overlap in many places and it helps to consider them in parallel. A word or two about terminology. Standard database terminology is used here. In particular a configurable item (such as “server”) would have a number of **attributes** (name, amount of memory, processor type, etc). Each record of a configurable item (CI) is called an instance. So in the example here, there would be one **instance** per server.

### 2.1 Asset Inventory

Essentially, an Asset Inventory focuses on the cost of individual assets. It's driven by the need to know the location and ownership of assets in order to then determine their financial value. Most organisations will have an Asset Inventory – typically held as spreadsheets – that is updated when items are purchased or written off. The motivation for keeping this information is largely to provide lists or partial lists for accounting purposes.

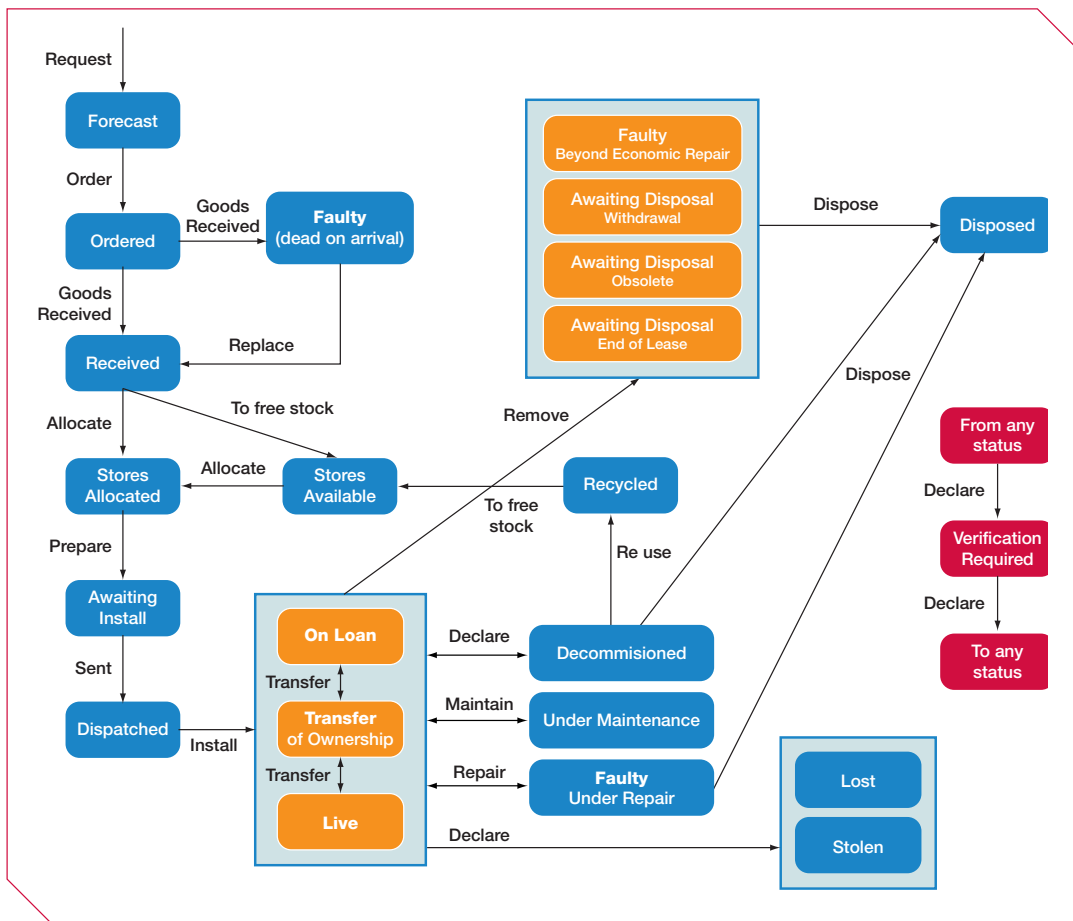
### 2.2 Asset Management Database

The focus of an Asset Management Database is also essentially financial. However, it addresses more ambitious needs. For example, where an organisation wants to actively investigate its IT resources and discover unused or underused assets. These assets might be physical (PCs for instance) or logical (such as software licences).

These modern requirements – and particularly the need to demonstrate that all software in use is properly licensed – require a more sophisticated approach. This is illustrated by the diagram which shows a basic process for managing the lifecycle of one CI (a hardware asset) – surprisingly complex for an essentially simple requirement.

It's worth restating that, in general terms, assets are Configurable Items (CI's) that have a financial impact; the company owns them or they have a financial component attached. Any CI that is 'in support' of the company's infrastructure, but has no financial impact will not in practice be administered in an Asset Management environment.

Figure 2: A typical Asset Management lifecycle



### 2.3 The CMDB

A CMDB is the structured collection of the Configurable Items (CIs) in a particular domain together with the relationships between them.

The primary focus of a CMDB should be operational – providing information to enable the IT environment to function. A common mistake that organisations make is to focus on collecting data because “it is a good thing” without identifying a clear operational need.

The CMDB is driven by the need to know which CI's are part of the infrastructure, including their status, relations, dependencies and (physical) location.

Configurable items are not explicitly defined in ITIL – in principle an organisation can choose their own. In practice many CIs – such as servers, networks devices, storage and applications – are self-evident. Other intangible ones – such as business services, support teams and virtual machines – less so.

### 2.4 The importance of relationships!

The collection and maintenance of the data on the CIs is a daunting task in itself. And the maintenance of relationship data is even more complex. So why does it matter?

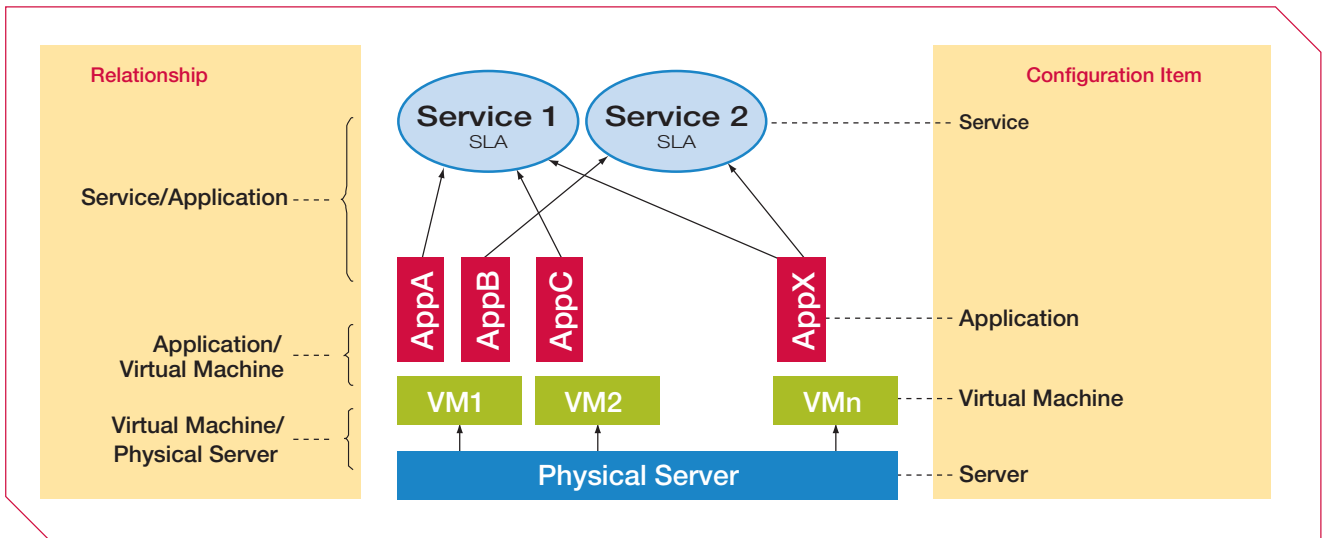
Relationship data is driven by the more sophisticated and business-focused Service Level Agreements (SLAs) that have arisen from the evolution of Business Service Management.

Consider Figure 3. A physical server in a data centre supports a number of virtual machines (VM1, VM2 etc) each of which will have one or more applications running on it.

A service (such as email) will require a number of applications to be running and there will be an SLA (or several) governing service delivery. What happens when the physical server fails? In order to know which SLAs are being broken it is necessary to know three relationships. These are illustrated in the yellow panel on the left. From the top down, they are:

- > Service to application
- > Application to virtual machine
- > Virtual machine to physical server

Figure 3: Example of the relationships needed in a modern CMDB



These relationships can change rapidly as virtual machines and applications are moved. Any attempt to manage the relationships manually (without the benefit of automation) is very labour intensive, requiring strictly imposed Change Management by all of the IT stakeholders. In reality that is very hard to achieve.

## 2.5 Key stakeholders

There are a surprisingly large number of them. They include:

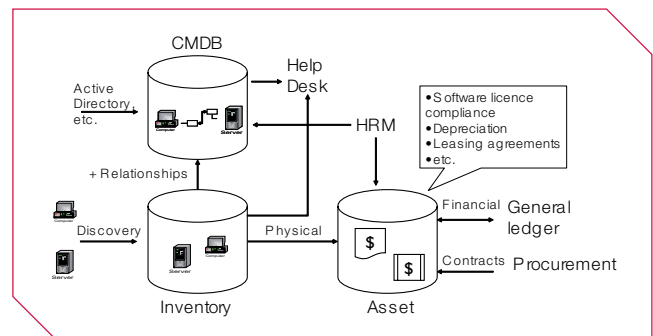
- > IT service management
- > Business managers
- > The finance department
- > The IT service desk
- > Project managers implementing changes
- > IT support teams (server, network, storage etc.)
- > Capacity planners
- > Data centre management and operations
- > Application development teams
- > Software compliance managers

Even a partial list like this one provides an indication of the biggest problem organisations face - making sure that the data held by every single stakeholder is accurate and up-to-date. No one wants to do, or take responsibility, for that task. Such is human nature that stakeholders want all the information to be available whilst someone else does the work needed to provide it.

## 2.6 Relationship between Asset Inventories, Asset Management and CMDB

From the earlier discussion it will be apparent that there are distinct but connected relationships between these databases.

Figure 4: The difference between Inventory Management, Asset Management and CMDB



In order to clarify the relationships between them, let's use a Server as an example:

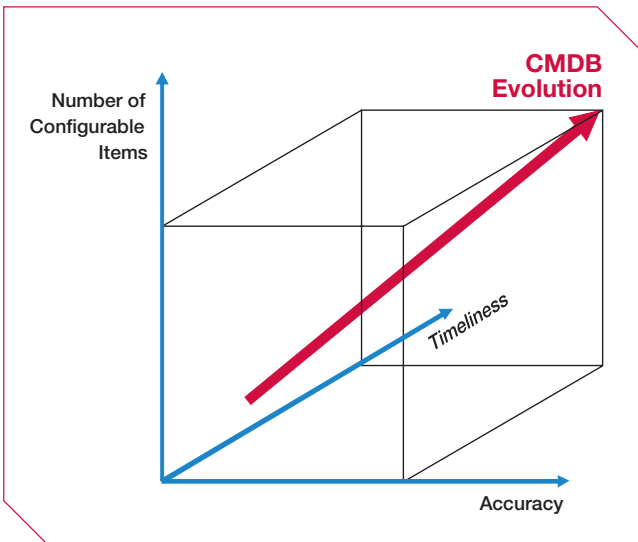
- > On an Asset Inventory there should be a simple record of the Server's location and the department that owns it (and other minor information).
- > On an Asset Management database the same Server would be linked to other data such as the software licences that are allocated to it, the usage statistics (and who was using it) and the amount of support it required.
- > On a CMDB the same Server would be linked to the software executables (i.e. the programs not the licences) deployed on it, user access rights, technical information such as IP addresses, worldwide names, memory and I/O adapters etc.

# 3 THE JOB JUST GOT TOUGHER

## 3.1 New demands

The demands upon CMDBs and Asset Management are growing all the time – and there are sound business reasons for this. Until recently most servers were bought to run a specific application and continued to run that application until they were decommissioned. The task of configuration management was therefore relatively simple – a flat file of servers showing the applications running on each could suffice. An audit once or twice a year would be enough to maintain adequate accuracy.

Figure 5: The dimensions of CMDB complexity



However, driven by the need to improve hardware utilisation, data centre managers are now partitioning servers into virtual machines and loading multiple applications onto each server. New technology enables them to move applications from one server to another in a few minutes – in order to balance load or to prevent outage during planned maintenance.

In this new world there are new CIs (virtual machines for example) and relationships frequently change.

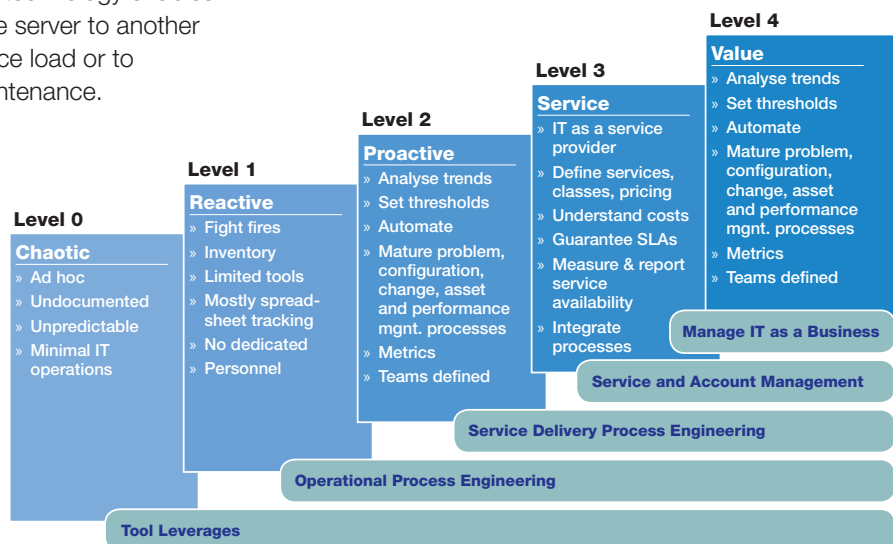
There is a trend towards Service Oriented Architectures (SOA). An SOA will consist of loosely coupled, and reusable, application services. These application services will be assembled into business services or workflows. And these all form new CIs and relationships.

There is also a further trend towards automated distribution of software and patches to desktops and servers. Accurate and up-to-date information about the existing software on the machine is crucial to achieving reliable distribution. Moreover, personal devices are proliferating, resulting in more CIs and less standardisation. Gaining access to corporate services is increasingly through non-corporate assets (home PCs, PDAs, hotel PCs, internet cafes etc.) which is yielding new complexities to be addressed. All of this means there is a continuous upward curve of complexity and expectations regarding the number of CIs, together with the timeliness and accuracy of their attributes and relationships (Figure 5).

## 3.2 Better processes

Configuration and Asset Management needs to be seen as part of the broader IT picture within an organisation, and its relationship to Business Management. When considered in this light, it becomes part of “Service Management Architecture”, which includes processes, tools, organisational structure and governance. The degree of refinement of this architecture is a measure of a company’s IT maturity, as shown by the Gartner Maturity model (Figure 6).

Figure 6: “Developing an IT Asset Management Tool Strategy That’s Ready for Prime Time” by Patricia Adams, September 2006



# 4 DATA MANAGEMENT CAN BE VERY COSTLY

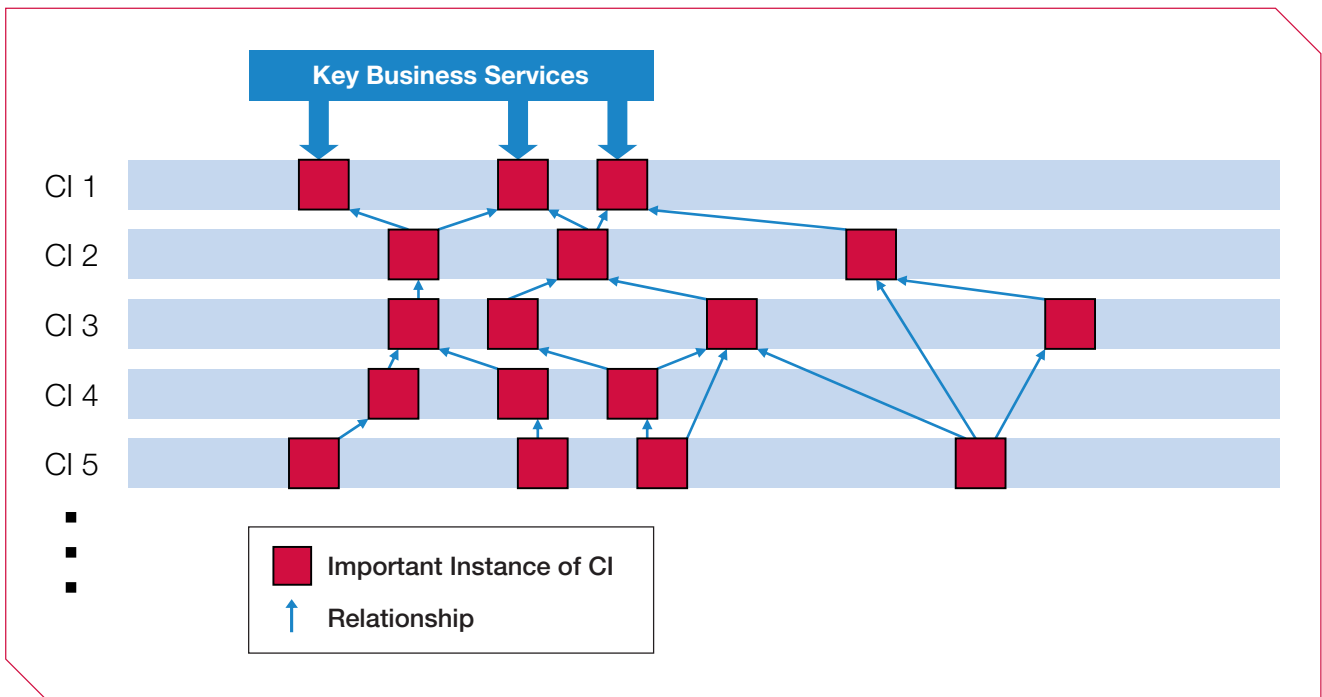
What is the impact upon costs in the new world of even more demand for CIs, delivered with greater accuracy and timeliness? The impact is easiest to see in CIs where there is a degree of manual data collection and maintenance. Atos Origin's experience is that the amount of manual activity, and hence cost, rises markedly as more information, accuracy and timeliness are specified. These costs result from more frequent auditing, more proactive data verification and dual keying.

But there is a key point to note and acknowledge. Just because a CI is identified in the CMDB does not mean that all instances of a CI have the same importance or that they carry the same expectations for accuracy and timeliness (a mainframe and a cheap server are extreme examples).

Given this difference in importance from one CI to another, it is common practice, when initially populating a CMDB to only partially populate a CI – concentrating on the most important instances. Focusing on the desired output in the initial stages of a project will limit what is initially captured and avoid unmanageably large quantities of data.

Best Practice is to focus on the most important instances of each CI which will usually be related to key business services.

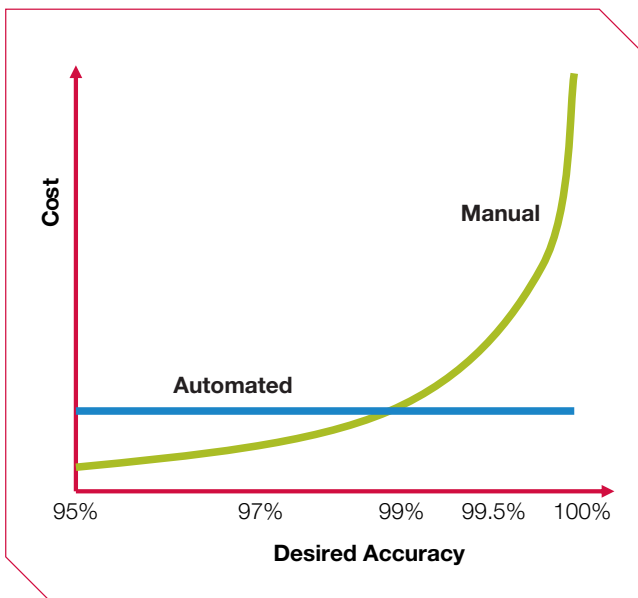
Figure 7: Focusing data collection on instances related to key services



# 5 THE FUTURE IS FEDERATED AND AUTOMATED

It is axiomatic that the demands for accuracy and timeliness - particularly in the data centre – cannot be achieved manually. Automation is therefore a highly desirable objective, because costs arise from the initial development and are essentially fixed.

Figure 8: Accuracy requires automation



However, automated tools have not been consistently successful. Most focus on a specific objective, such as the management of desktops, and fail to take a holistic view of wider and interdependent objectives. The challenge to be faced is the federation of different tools that complement each other and assist in generating relationships for the CMDB.

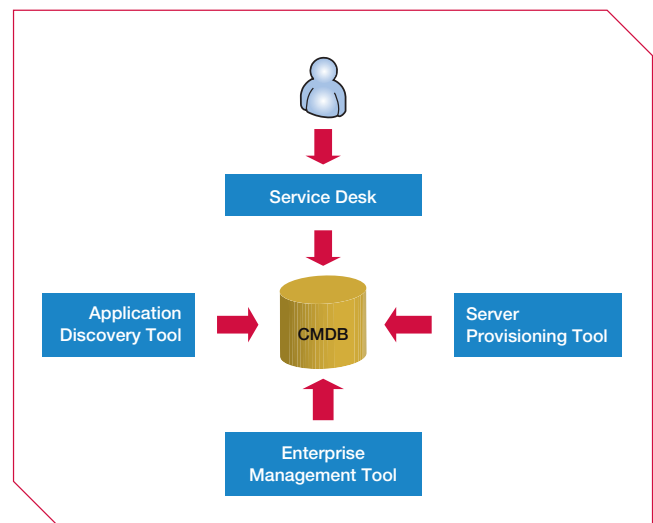
One problem has been that automated tools tend to focus on what is easy to collect rather than what is useful – for example collecting the filenames of the executables on a disk rather than identifying the versions of (licensable) applications installed.

It is clearly also necessary to assess the value of investment in automation – the additional cost of trying to automate the most difficult piece of the jigsaw may not be worthwhile. Often, for example, the most important piece of information about a laptop is who actually has it at any given point in time – a very difficult thing to automate.

In automating data collection the following concepts need to be considered:

## 5.1 Reconciliation

Figure 9: Multiple data sources discover information



Today, most organisations have multiple discovery tools collecting overlapping CI information. Where there are overlaps there will sometimes be differences. Here's an example where the amount of RAM on a server is in doubt:

- > The service desk shows 8GB – an increase from 4GB due to a change implemented on May 1st
- > The Enterprise Management tool shows 4GB from its last scan
- > The Server Provisioning tool shows 4GB from its last scan
- > The Application Discovery Tool couldn't find the server on its last scan.

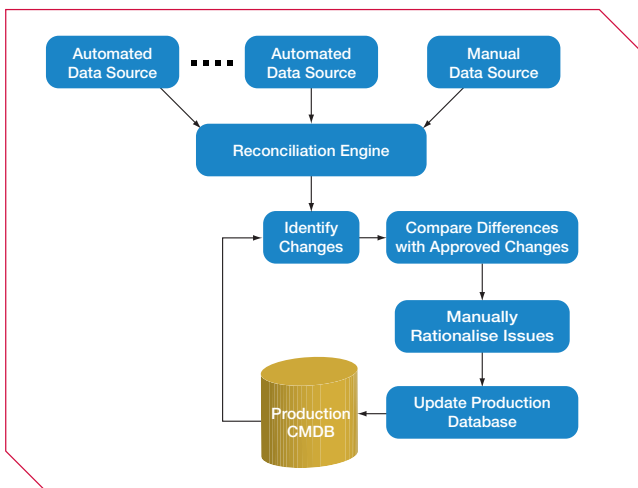
Which of the four possibilities is correct? In order to determine that, numerous scenarios can be conceived:

- > The wrong server name was entered by the Service Desk when the instance was created
- > An attempt was made to upgrade the memory which didn't work (but the Change record was updated)
- > The additional memory was actually taken out again after May 1st but no Change record was made
- > The most recent scan by the discovery tools predated the Change
- > The server was offline when the most recent scans happened and so the earlier data was carried forward.

Reconciliation is the automated process of merging multiple data sources and determining which values to use when conflicts arise. It uses business-driven rules which are applied when there are data overlaps to determine which takes precedence and to avoid duplicate entries (with slightly different key values – “MS word” and “Microsoft Word” for example). The data that takes precedence is henceforth referred to as the Master record.

A control regime is imposed to ensure data quality is maintained. A typical approach is to combine all of the automated sources of data (discovery tools etc.) with manually added records. The reconciliation engine then processes these to produce a candidate production CMDB. This is compared with the current production database to identify any differences. Such differences may be due to authorised changes, unauthorised changes or incorrect data. In principle these issues need to be resolved before the candidate CMDB becomes the new production CMDB.

Figure 10: The reconciliation and rationalisation process



But in practice, particularly in the early days, there can be so many issues to address that a complete solution is impractical – the task is too big and it would take far too long. A better solution is to prioritise issues according to the business requirements for accuracy.

## 5.2 Federation

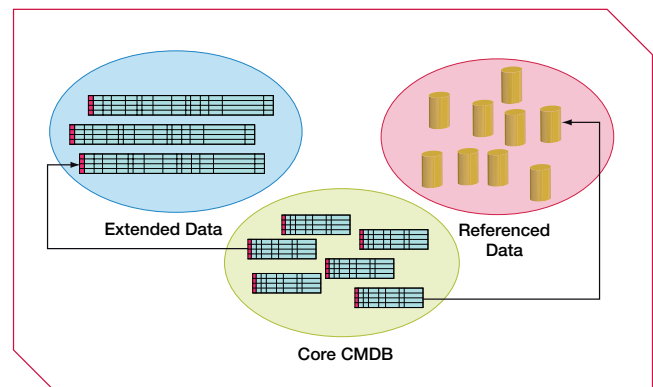
The conceptual diagram (Figure 9) shows data flowing towards the CMDB – but this is rather simplistic. When considering all the information that could conceivably be held in a CMDB it doesn't take long to arrive at a huge list – all the registry settings in a server or PC, for example.

Practical considerations mean that a single central database would be too expensive, too cumbersome and cause a bottleneck. So the next step is to recognise that there will usually be key tools (such as the software provisioning tool) which are the authoritative source of some CIs, but whose performance would be degraded if forced to read and update a remote database.

This recognition leads to a three-level hierarchy:

- > Core Data that belongs in a central database – such as server name and number of processors
- > Extended data that will be held in a management tool or even the device itself – such as registry information. Each record will relate to one or more pieces of Core Data
- > Referenced information – often free format text such as contracts, help desk tickets, pertaining to that CI.

Figure 11: The federated CMDB



In this way a federated CMDB provides a consistent and easy method in which to access these separate data stores so that a composite view of the information is provided.

Two important criteria for deciding whether data is Core or Extended are:

- > If data is transient or changes rapidly, then it should not be part of the Core. An example is the relationships between virtual machines and physical servers – these would not be part of the Core
- > All data necessary to perform searches should be part of the Core. In practice this means that database keys are Core, so that queries can join tables without accessing extended data.

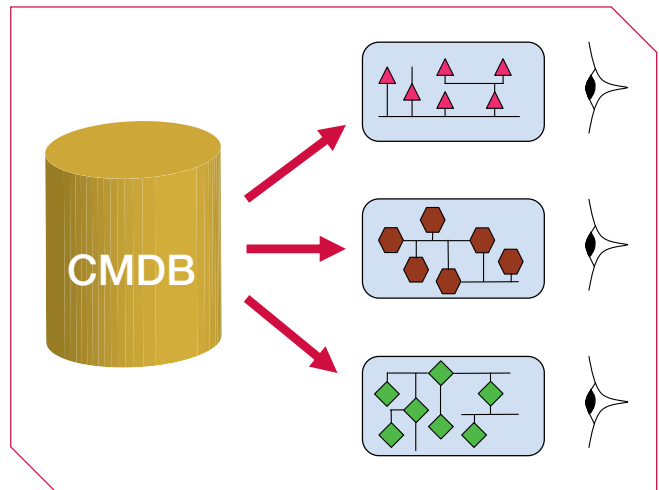
### 5.3 Mapping and visualisation

It has already been noted that the relationships between CIs are at least as important as the CIs themselves; these relationships differentiate the CMDB from a traditional Asset Inventory list. Presenting relationships in the form of lists or spreadsheets or text screens doesn't work – we do not digest the information easily; errors are missed and significant facts overlooked.

The real power of visualisation is therefore the ability to present dynamically drawn diagrams from the CMDB specific to users' roles. Here's an example where a change is proposed to a particular server:

- > The server management team will want to see a diagram showing the server plus all of the other servers with which it communicates. But they will not want to see any of the network infrastructure that connects them – or any servers with which they do not communicate
- > The network management team will want to see any network devices connected to that server – but not other servers or other infrastructure
- > The Service Manager will want to see all of the business services that in some way use that server.

Figure 12: Multiple views of the same infrastructure



Accurate relationship details are essential to produce these diagrams.

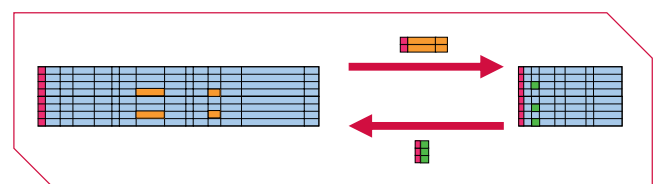
### 5.3 Synchronisation

Most enterprises will have a variety of data sources from more than one vendor. In order to maintain the core CMDB it will be necessary to replicate data from these sources. Some of these links may be two-way – a source might be the master for some attribute data but not for other attributes.

Whilst it is technically feasible to set up closely coupled links so that replicated data is updated at the same time as the master, most enterprises do not do this. There are issues regarding processing power and network bandwidth, together with complexity, that result in scheduled updates – at frequencies from an hour to a week. Normally, only the changes are transferred. To achieve this all of the federated data has to have:

- > A copy of the state of the data when the last update was performed
- > The current state of the data
- > The changes calculated from the differences between the above.

Figure 13: Synchronisation can be a 2-way process



# 6 AUTOMATED TOOLS EMERGE

Automated tools have made an impact upon both the Data Centre and the distributed environment – albeit in different ways.

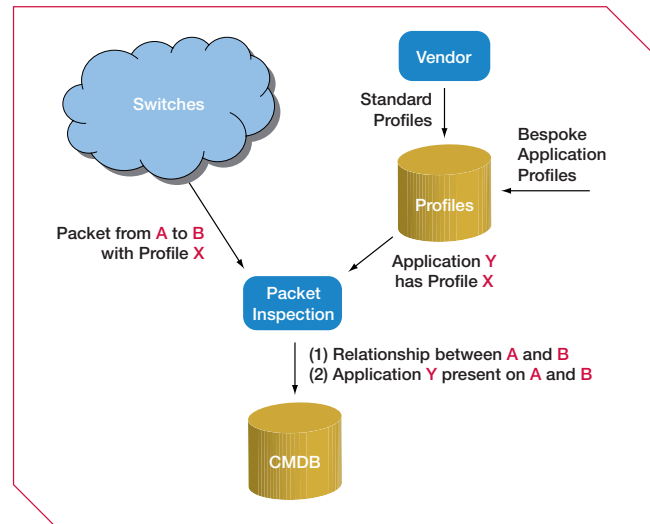
## 6.1 Application Discovery Tools

As previously stated the major gap has been discovery tools that automatically populate the relationships between services and infrastructure (and servers in particular.) Over the last two years a large number of innovative software companies have developed tools focused upon solving this problem. Most have subsequently been acquired by the major tool vendors. The methods used by these tools deploy one of the following techniques:-

### 6.1.1 Network sniffing

These tools are attached to network switches and monitor the traffic passing across the network<sup>2</sup>. Every packet is examined to determine the IP addresses of the sending and receiving servers in order to establish the relationship. Deep packet inspection tries to determine the protocols (e.g. JDBC, HTTP, TDS etc.) and the services at each end (Apache, IIS, Oracle, Internet Explorer etc). The volume of traffic between two servers can also be captured and is indicative of the type of relationship. A reference database of profiles of common applications (ERP, CRM, databases etc) is then used to determine the type of application. The relationships are populated using the IP addresses to find the server CIs. The profiles of the enterprise's bespoke applications are added to the database to complete the picture.

Figure 14: The network sniffing method



This approach has the advantage of being non-invasive – it generates no additional load on servers or the network and does not require administrator privileges for the servers. On the other hand, physical network connections are required to the core switches.

These devices are typically dedicated appliances that are permanently installed. They provide daily updates of changed relationships – both new ones and those which are no longer active.

One drawback to this approach is that it cannot identify the passive servers in a cluster – the only traffic to these servers is the heartbeat<sup>3</sup> and this frequently uses a dedicated connection that will not be monitored.

Another issue is that discovery does not work well if traffic is encrypted.

<sup>2</sup> Port mirroring is used to replicate traffic from other switches – note that this generates extra traffic on the inter-switch connections

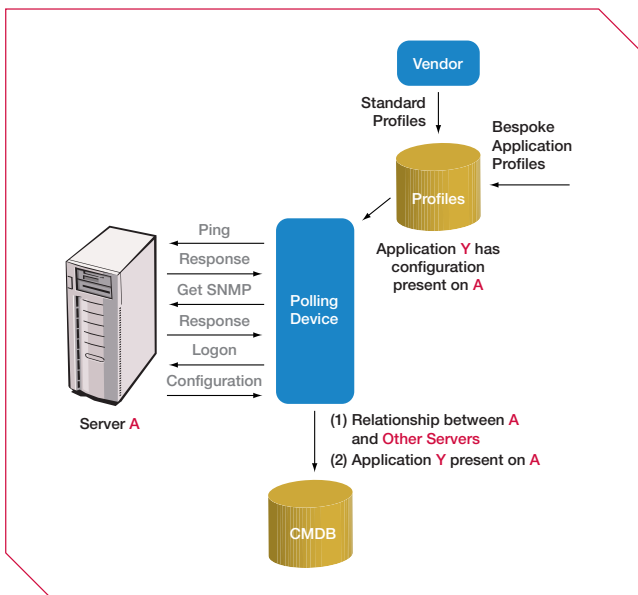
<sup>3</sup> A common way of providing resilience for devices (such as servers) resilience is to deploy an "active/passive pair". In this case one device (the primary) provides all of the service under normal conditions while the other (the secondary) is dormant until the primary fails. A regular message sent from the primary to the secondary (called a heartbeat) is often used to control the transfer – when the secondary fails to receive a heartbeat it takes over the delivery of service.

### 6.1.2 Active polling

These tools will actively poll every device within specified IP address ranges. Typically a “ping” will be used to establish if a device is present. An adaptive approach is then taken for each device discovered. Basic device information will be found using the SNMP MIB<sup>4</sup> for example. Servers can be interrogated to establish the operating system in use. Specific information for that operating system can then be obtained (e.g. through WMI<sup>5</sup> for Windows and SSH<sup>6</sup> for UNIX) together with the software applications installed on the server. Once logged on, utilities such as Netstat can be used to determine which other servers and ports a server is communicating with (the basis of the relationship).

Profiles (sometimes called fingerprints) are then used to determine the applications in use. These profiles are the real intellectual property of the tool vendors and are specific to the application. For example Oracle databases have a System Identifier (SID) that identifies a specific instance of a database and is used by other servers when accessing it. Discovering the same SID on an application server and a database server will confirm a relationship.

Figure 15: The active polling method

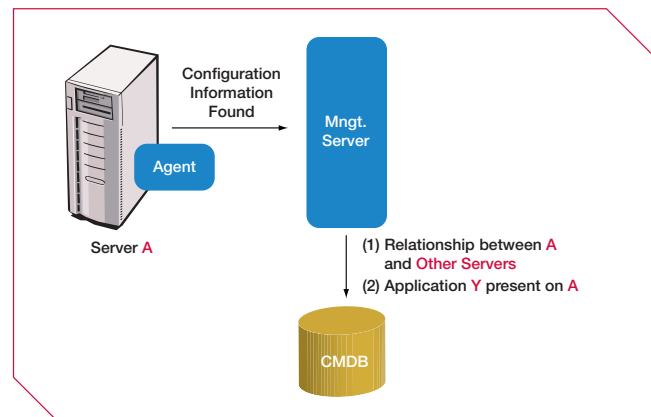


The invasive nature of these tools is their chief drawback – their behaviour is identical to that exhibited by a hacker – and hence network security officers will need to be reassured. Furthermore, the network traffic generated by these devices can cause problems – particularly on large sites or where remote sites are being polled over a WAN.

### 6.1.3 Agent based

With these tools an agent is installed on each server to be discovered. The agents perform detailed scans of installed software. When specific products are identified their configuration tables are accessed to determine which other servers they communicate with. But the agents’ real power is the ability to immediately detect changes to relationships due to changes made in the configuration files. This approach can also detect traffic with other servers, offering similar benefits to the “sniffing” approach.

Figure 16: The agent-based method



The main drawback to this approach is the need to deploy the agents in the first place. Although inherently “read only” agents must be checked to make sure they do not interfere with production applications – a significant testing exercise. But once installed they provide excellent and granular information. Although security concerns have to be satisfied for the initial deployment agents are safer to maintain than polling approaches as there is no need to lodge credentials<sup>7</sup> with the management console. The network traffic arising from repeated polling is also avoided.

<sup>4</sup> A management information base (MIB) is a formal description of a set of network objects that can be managed using the Simple Network Management Protocol (SNMP). The format of the MIB is defined as part of the SNMP. Hence any device that supports SNMP will expose a MIB with fields containing relevant data about the device. Internet documentation RFCs discuss MIBs notably RFC 1155 (Structure and Identification of Management Information for TCP/IP based internets), and its two companions, RFC 1213 (Management Information Base for Network Management of TCP/IP-based internets) and RFC 1157 (A Simple Network Management Protocol).

<sup>5</sup> Windows Management Instrumentation. WMI is a management technology allowing scripts to monitor and control managed resources throughout the network. Resources include hard drives, file systems, operating system settings, processes, services, shares, registry settings, networking components, event logs, users, and groups. WMI is built into clients with Windows 2000 or above, and can be installed on any other 32-bit Windows client.

<sup>6</sup> Secure Shell. Secure Shell is a program designed to perform a number of functions, such as file transfer between computers, execution of commands on a remote computer, or logging on to a computer over a network. It is intended to be able to do these tasks with greater security than previous programs such as telnet or ftp.

<sup>7</sup> Credentials are typically logon and password combinators with top level privileges.

Figure 17: Comparing the discovery methods

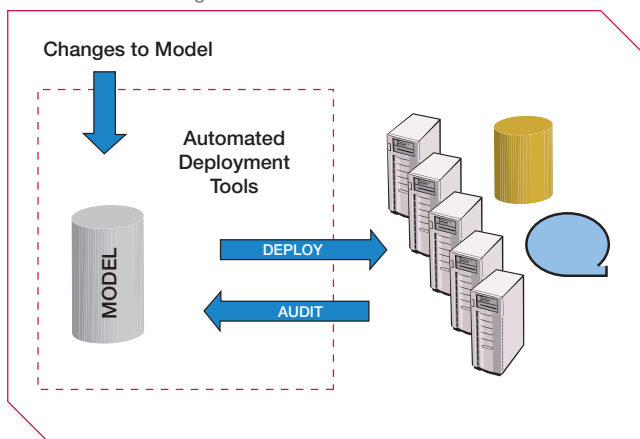
	Sniffing	Polling	Agent-based
<b>Intrusive</b>	Not at all	Highly	Highly – can conflict with applications
<b>Network Load</b>	None <sup>8</sup>	Heavy	Modest
<b>Identifies Secondary devices in Active/Passive pairs</b>	No	Yes	Yes
<b>Immediately identifies changes</b>	Yes	Only when next polled	Only if agent installed
<b>Requires Security Credentials</b>	No	Yes	Only for installation of agent
<b>Deployment effort</b>	Modest	Significant	Very high – can take a long time

## 6.2 Model based provisioning

This trend is for automation to use a model-based approach to Data Centre management. The key points of this approach are:

- > Administrators never directly modify infrastructure. They change the model and then automated tools modify the infrastructure to reflect the model
- > The effort goes into developing reliable tools to instantiate the model onto physical infrastructure
- > An automated audit process ensures that the infrastructure is always synchronised with the model
- > The approach addresses Governance issues with a robust audit trail for changes.

Figure 18: Model-based configuration management of servers and storage



<sup>8</sup> Unless multiple switches are used and port replication in which case there is extra traffic on the inter-switch links.

The model-based approach is highly scalable because the majority of costs are fixed. It's also an approach that fits perfectly with the proposition of a CMDB – because the model forms the reference for a number of CIs and the relationships between them.

The area that has benefited most from automation is that of software provisioning – essentially implementing ITIL Release Management. Software provisioning is the process of installing and initiating software onto a server. This is a more systematic approach than the traditional method of a system administrator installing a software release on a server. Software provisioning involves setting up a core server to hold the set of software releases (the Definitive Software Library in ITIL terms) and then the provisioning software automatically deploys the right software to the right target server. This ensures both consistency and the ability to easily rebuild a server following a failure. Examples of model-based software provisioning tools include Opsware and HP Radia.

The use of a software provisioning tool significantly simplifies the task of software licence compliance by ensuring the existence of accurate deployment information. Controls can also be built into the provisioning process to make sure that an unlicensed installation is prevented.

## 6.3 Standards

As a CMDB is simply a database it will need a schema. In practice the schema will usually be defined within a software product – albeit customisable. However, most enterprises will have tools deployed from more than one vendor. Implementing a federated CMDB therefore requires data to be interchanged; and in order to address interfacing issues the enterprise will be forced to understand the schema details. This has been recognised by the major software vendors for many years and there have been a number of initiatives to develop standards for both the format of data and the way it is exchanged.

The key ones are:



### 6.3.1 The Distributed Management Task Force (DMTF)

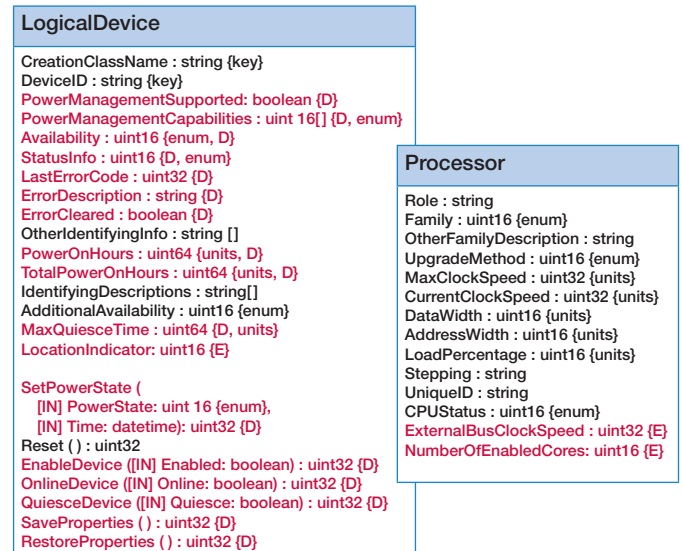
The relevant initiative supported by DMTF is the Common Information Model (CIM). This is a comprehensive object-oriented model that includes:

- > Physical devices – down to interface cards, ports and discs
- > Logical devices
- > Software
- > Batch jobs
- > Services
- > Storage
- > Resilience (redundancy)
- > Settings and configuration information
- > Networks
- > Events
- > Databases
- > Users

A detailed description of the model is beyond the scope of this paper – please refer to [www.dtmf.org](http://www.dtmf.org) for further information. But an indication of the model's completeness can be gained from the two examples of detailed attributes shown in Figure 19 – logical device and processor. The object model also defines the different types of relationships between objects. Although the model continues to be enhanced (version 2.13 was released on 7 September 2006) hardware vendors in particular have adopted the schema, providing a degree of information standardisation embedded in products.

It is incumbent upon users to ensure that they employ similarly consistent standards – for example in server names.

Figure 19: An extract of the CIM



### 6.3.2 The CMDB Federation Consortium

A consortium of HP, IBM, CA, BMC and Fujitsu has been formed to define an interface standard which permits the exchange of data in a heterogeneous federated CMDB. This is an important step as it provides a realistic goal (for interchange) whilst allowing vendors to retain the flexibility to implement CMDBs as they see fit. Products supporting this standard are likely to emerge in 2008.

### 6.3.3 The future?

Attempts to develop and implement standards are naturally welcome. However, the CMDB market is still immature and leading analysts predict that it will be several years before mature products emerge. In the interim there is a danger that standards efforts will fail through becoming over-complicated or through vendors withdrawing support.

## 6.4 Distributed Computing

Maintaining accurate CMDB information outside the controlled environment of the Data Centre is especially challenging - but equally important. There are many aspects to this challenge.

### 6.4.1 Ownership

At any given point in time a PC (and particularly a laptop) will be in the hands of a particular member of staff. Knowing their identity is important for recharging and asset ownership reasons, let alone the practical matter of contacting them when maintenance is required.

Microsoft Active Directory (AD) is a rich source of CMDB information, but it needs to be synchronised with the Human Resources (HR) database so that up-to-date information about an individual (and particularly their cost centre) is maintained. Similarly the HR system would provide the contact data that is often associated with assets.

### 6.4.2 Location

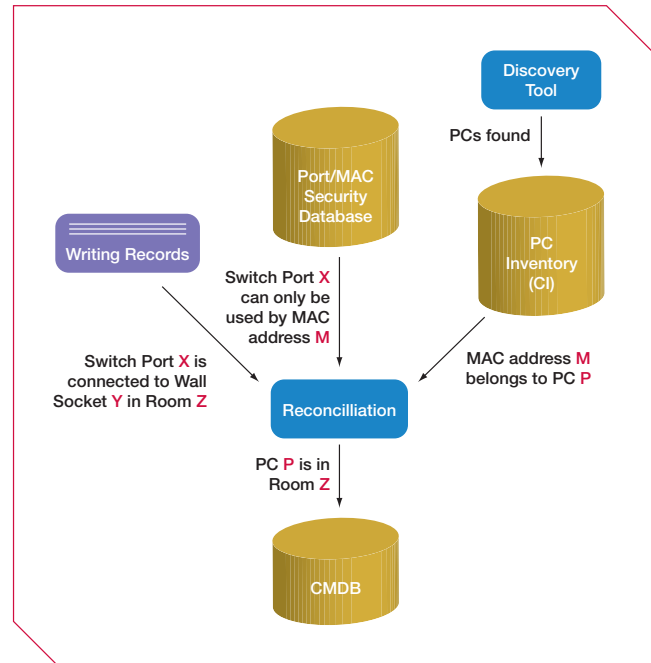
Not only is the physical location of an asset needed, but, for portable devices, it is also necessary to know which network a device is connected to. Three particular problems arise from these requirements:

- > Devices that are in storage (and so never appear on a network scan)
- > Devices that are deliberately not connected to the internal network - local printers, external web servers etc
- > Mobile devices – particularly laptops – that connect intermittently over a variety of networks with different transmission speeds.

Even fixed devices such as desktop PCs can pose a problem if they are moved to a different office without the CMDB being updated.

A common approach for desktops is to configure the network switches so that they will only recognise traffic on a particular port from a device with a specific MAC address. The location of the device is derived from knowing which cable the port is patched to and which office it is terminated in. It should be noted that this approach assumes accurate records of the structured wiring are maintained and form part of the CMDB.

Figure 20: Deriving the location of a PC from its network connection



If the desktop is moved then it will only work if the port security is updated – which in itself guarantees that accurate records are maintained. This even takes account of the situation when an Ethernet card fails (and is replaced by a card with a different MAC address) – the replacement will have to be entered into the security system and associated with the desktop machine. Knowing the physical location of a mobile device is rarely useful – mobile devices change location by virtue of being mobile. But it is valuable to capture the user’s logon-id and connection speed each time a connection is made. This validates that the PC has not been lost or stolen and that ownership data is current. This is also useful information for the software distribution system – trying to download large patches over a dial-up connection is not desirable!

### **6.4.3 Naming conventions for devices**

Every device will have a name by which it is known to the system – the Computer Name in the case of PCs running Windows. Most organisations also give a PC a physical asset tag – often a bar-coded label – which is used during audits. It is frequently necessary to correlate the two and the simplest approach is to give them the same value. Users will typically report problems by quoting the asset tag whilst the support teams will perform remote management using the system name. A strict manual process to enforce data quality is clearly required – otherwise the support group may try to fix the wrong PC, resulting in two unhappy users!

### **6.4.4 Roaming and role-based computing**

Some organisations implement roaming so that a user can go to any PC and, when they logon, be given access to all the software to which they are entitled (as defined in Active Directory). Any further software the user requires, if not already on that PC, will have to be downloaded to it.

This has the potential to create a software compliance issue, as most licenses are based upon software being installed on a specific device.

The normal way of addressing this is to remove software from a PC after a number of days of inactivity. But this devalues information derived from discovery tools as they will over-report the number of licences required.

Any physical changes to a machine also run the risk of confusing the software deletion process and again resulting in excessive licence demand.

# 7 DATA LIFECYCLE MANAGEMENT

Achieving the objectives of accuracy and timeliness cannot happen without careful planning. For each CI it is necessary to devise and implement processes for:

- > **Creation** – how an instance of the CI first appears on the CMDB. The issues here are how to ensure that a new instance is captured promptly and reliably and how all of the attributes are validated for accuracy
- > **Change** – the attributes of a CI instance may be updated for any number of reasons. This naturally gives rise to multiple processes. An exhaustive list of the reasons needs to be created along with a robust update process. In extreme cases the format of the CI itself could change
- > **Decommission** – when an instance of the CI is removed from the environment it must be captured. If an historical record is required for instances that no longer exist, the situation can become complex. It is usually preferable to define a new CI for these rather than retain ambiguous records.

A scan by automated discovery tools will provide validation of this process. An example helps to understand this – here the CI is a PC.

- > **Creation** – this process could be automatically linked to:
  - The purchasing system - in order to generate a record when the delivery is accepted
  - The change management system – when the purchase is requested or when a request to commission the new machine is raised
  - The network security system when the MAC address is enabled.
- > **Change** – this process could be initiated when:
  - A new piece of software is requested or installed
  - The virus signatures are updated
  - A new piece of hardware is installed (e.g. a memory upgrade)
  - The PC ownership changes or it goes into storage
  - The PC is moved to a different location
  - The PC fails and is swapped out.
- > **Decommission** – this process would be:
  - Initiated when the PC was released by the last user
  - Involve the secure removal of the data and software
  - Remove the hardware inventory record and move to the archive
  - Update the software licence records (for potential re-use).

# 8 IMPLEMENTATION

Once you've decided that a project is needed, where do you start? Before answering that question, there are a few places where not to start:

- > Planning the schema
- > Talking to vendors about software tools
- > Looking at the existing information sources.

Remember – most projects are stopped because senior management started looking at the implementation costs and asking the question – why are we doing this?

## 8.1 Analysis and design

The place to start is here - establish what the CMDB will be used for and why/how that benefits your organisation. This is the analysis and design phase. It is important to recognise that there will be business requirements (such as compliance, audit etc.) that complement IT requirements (relationships to support SLA reporting, etc) and that these involve different stakeholders.

The next step is to see how few CIs are needed in order to deliver the business requirements.

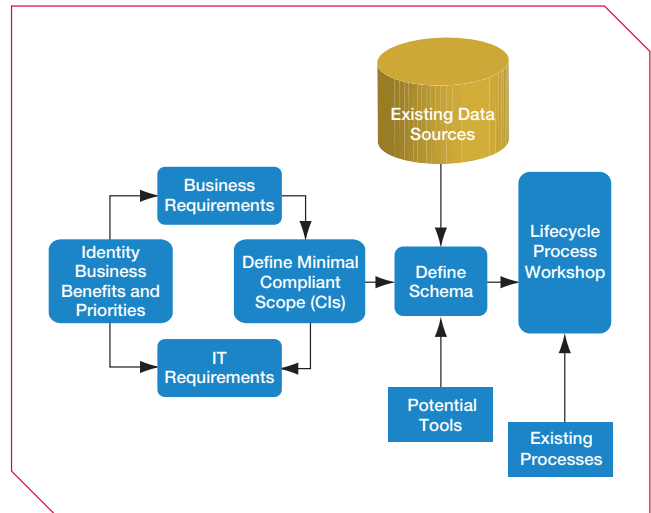
At this point it is necessary to consider the use of “group” CIs. When a large number of instances share a lot of common attributes (for example a thousand PCs with identical software builds) it makes sense to define a CI to hold a single instance of the common attributes. This saves storage and more importantly simplifies the maintenance task.

The schema (possibly based on CIM) is then produced to encompass these CIs and nothing else – the minimal compliant scope.

Once the schema is understood it is necessary to work out the most effective way of delivering the information into the schema. This has to take account of the constraints of the proposed tools – most of which will limit the entities and attributes that can be changed.

The lifecycle of each CI needs to be documented and all the associated processes re-worked.

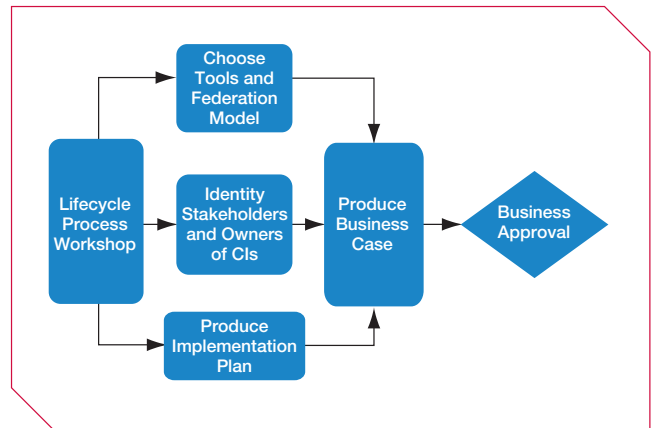
Figure 21:Phase 1 - Analysis and Design



## 8.2 Planning and business case

The analysis of lifecycles will usually identify the need for one or more tools. If multiple tools are used you will need to federate the CMDB – which will require further design work to define the synchronisation approach.

Figure 22: Phase 2 - Planning and Business Case



For data to be accurately maintained each CI must have a clearly defined Owner who accepts responsibility for all of the associated processes. Their commitment must be formally documented along with the other planning artifacts.

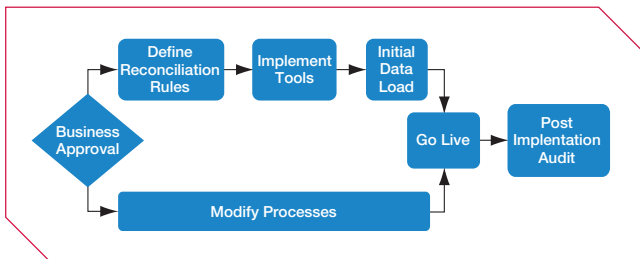
All of the components are then assembled into a Business Case for management approval. This step is absolutely key – management must unambiguously back the project or it will be a costly failure.

### 8.3 Implementation

Having gained approval, you can now undertake IT-related tasks, such as resolving the details of reconciliation and implementing the tools. Try not to underestimate the time it will take to modify business processes. Modifying the process documentation is often simpler than communicating with, and training, staff to follow them consistently.

Once the Big Day is over and the CMDB is implemented, you will need to schedule an audit to ensure that everything is working as it should be. Typically, a sample is used to verify data accuracy and completeness.

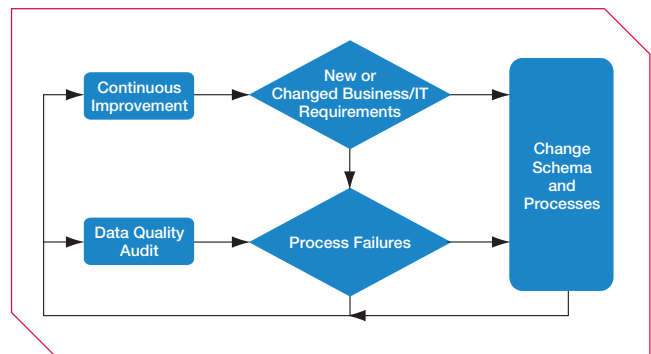
Figure 23: Phase 3 - Implementation



### 8.4 Maintenance

Many projects are completed successfully, but the results are wasted because the CMDB is not maintained adequately and falls into disrepute. If this is not to happen, it needs to be recognised that business requirements change and evolve, and the CMDB and supporting processes must be modified to accommodate those (continual) changes. Furthermore, data quality needs to be regularly checked so that faulty processes can be corrected quickly.

Figure 24: Phase 4 - Maintenance



## Conclusion

Implementing and maintaining a CMDB is no longer optional. The pressures to maintain more data to higher standards will grow – and automation is the only practical way to address this demand.

Organisations are already recognising that this is neither easy nor cheap. Furthermore, the necessary tools are still immature and will be for several years. Consequently, pragmatic approaches will be necessary – because waiting for the tools to mature is also not an option.

Pragmatism should be combined with a clear strategy driven by current and projected business needs. As tools mature they can be progressively deployed and benefits progressively realised. Satisfying regulatory demands as cheaply and efficiently as possible, whilst gaining all the business benefits a CMDB can offer, is a genuine win-win situation.

Grasping the challenge now rather than waiting, will deliver both immediate and long-term benefits. The CMDB will then take its place at the epicentre of the modern IT operation.

**For further information, please call +44 (0)20 7830 4444  
or email [MO.Marketing@atosorigin.com](mailto:MO.Marketing@atosorigin.com)**

## About Atos Origin

Atos Origin is an international information technology services company. Its business is turning client vision into results through the application of consulting, systems integration and managed operations. The company's annual revenues are EUR 5.5 billion and it employs over 47,000 people in 40 countries. Atos Origin is the Worldwide Information Technology Partner for the Olympic Games and has a client base of international blue-chip companies across all sectors. Atos Origin is quoted on the Paris Eurolist Market and trades as Atos Origin, Atos Euronext Market Solutions, Atos Worldline and Atos Consulting™.

## About Atos Consulting

Atos Consulting is a leading provider of business, process and technology consulting services. With more than 2,500 staff globally, it focuses on delivering proven, pragmatic solutions to the telecom, manufacturing, financial services and public sectors.