Cracking the Enigmatic CMDB

Glenn O’Donnell

Cracking the Enigmatic CMDB

Why the CMDB is suddenly the hot topic

Defining the CMDB

Automating CMDB management

Using the CMDB

Configuration Is the Heart of Process

All processes benefit from accurate configuration

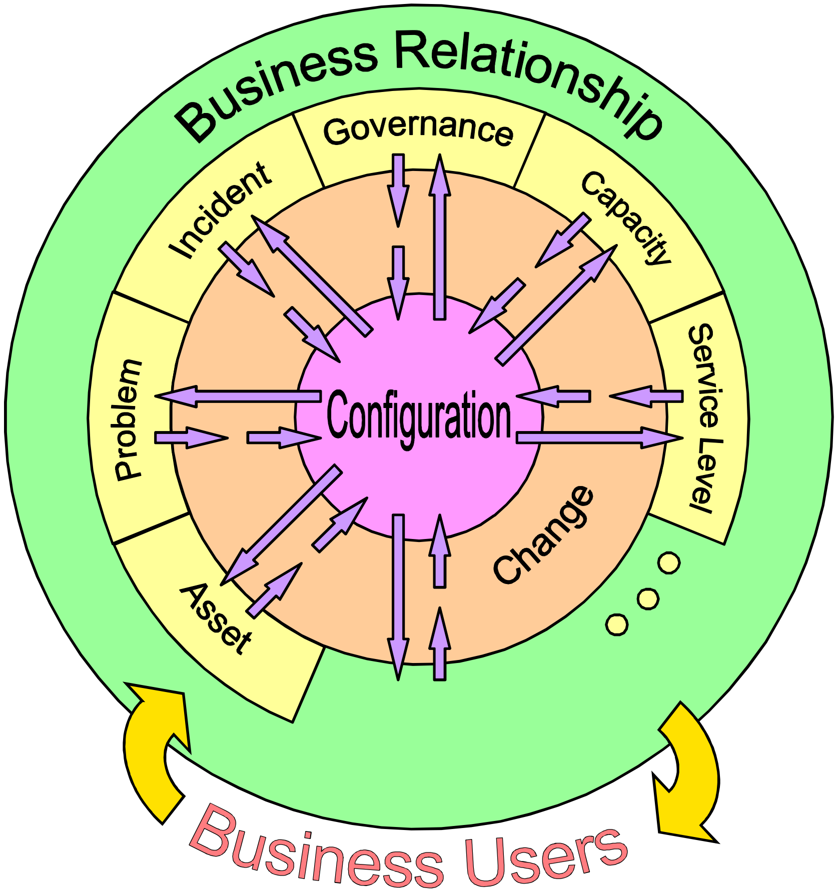
Configuration management answers the basic IT question

- "What is the authentic description of my world?"

Configuration is now the hottest topic in Operations

Regulatory compliance is the main driving force

- A "perfect storm" of forces is fueling configuration



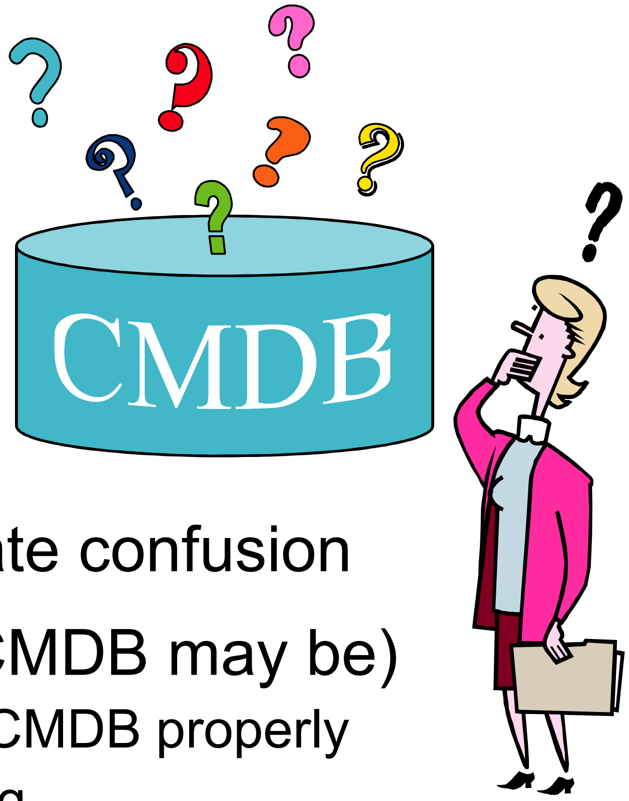
Hyperbolic Growth = Mass Confusion

ITIL references CMDB extensively

- But its definition must be clarified

The "Perfect Storm" fuels the hysteria

- Users want to jump on the bandwagon
- Mass media reacts to satisfy the voracious appetite for information
- Vendors see revenue and fabricate CMDB messaging to compete

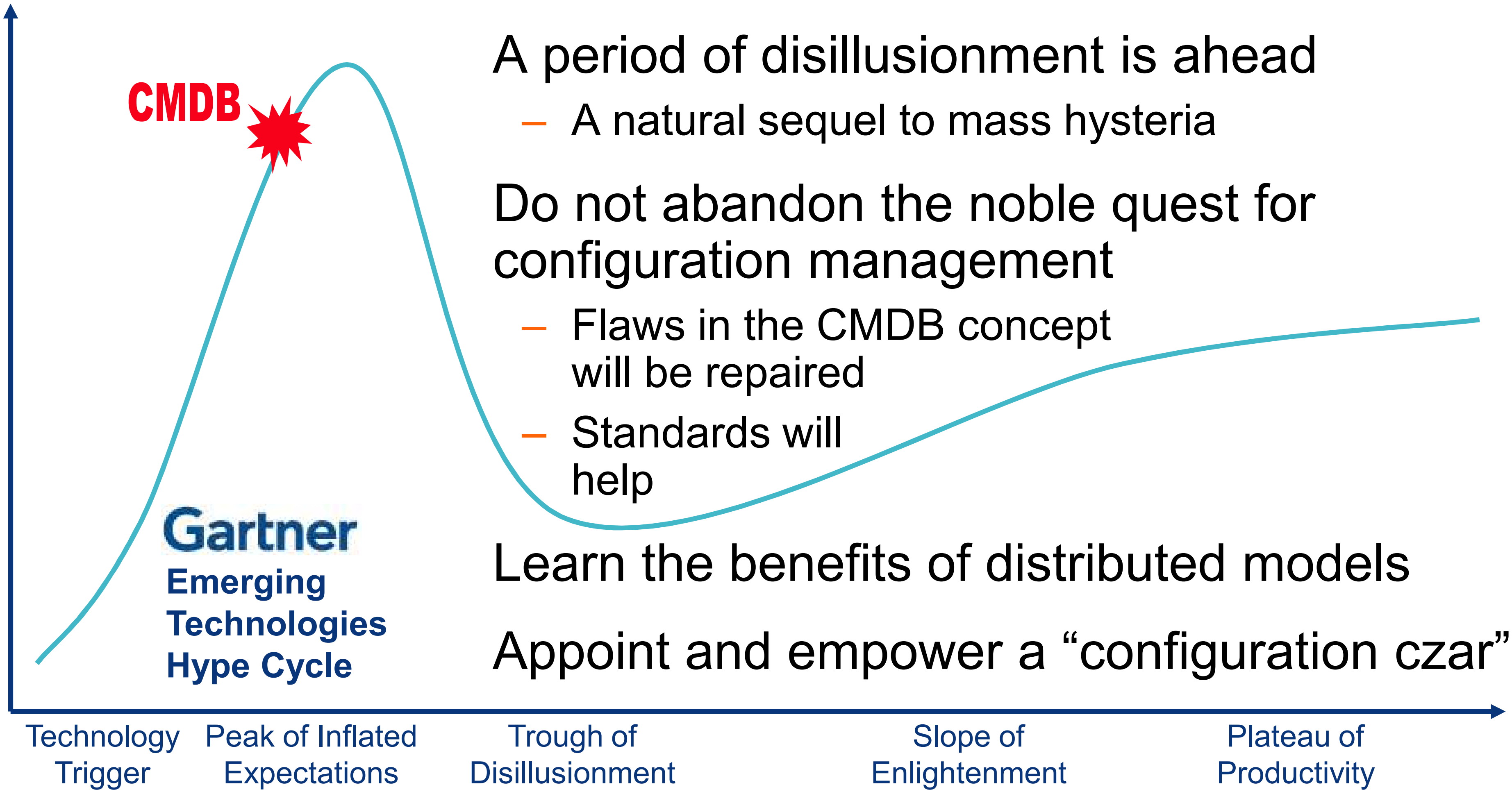


Inevitable conflicts in messages generate confusion

Configuration management is no fad (CMDB may be)

- Still, there is enormous value in building a CMDB properly
- Technology solutions are only now maturing

Settling into Pragmatism



A period of disillusionment is ahead

- A natural sequel to mass hysteria

Do not abandon the noble quest for configuration management

- Flaws in the CMDB concept will be repaired
- Standards will help

Learn the benefits of distributed models

Appoint and empower a “configuration czar”

Cracking the Enigmatic CMDB

Why the CMDB is suddenly the hot topic

Defining the CMDB

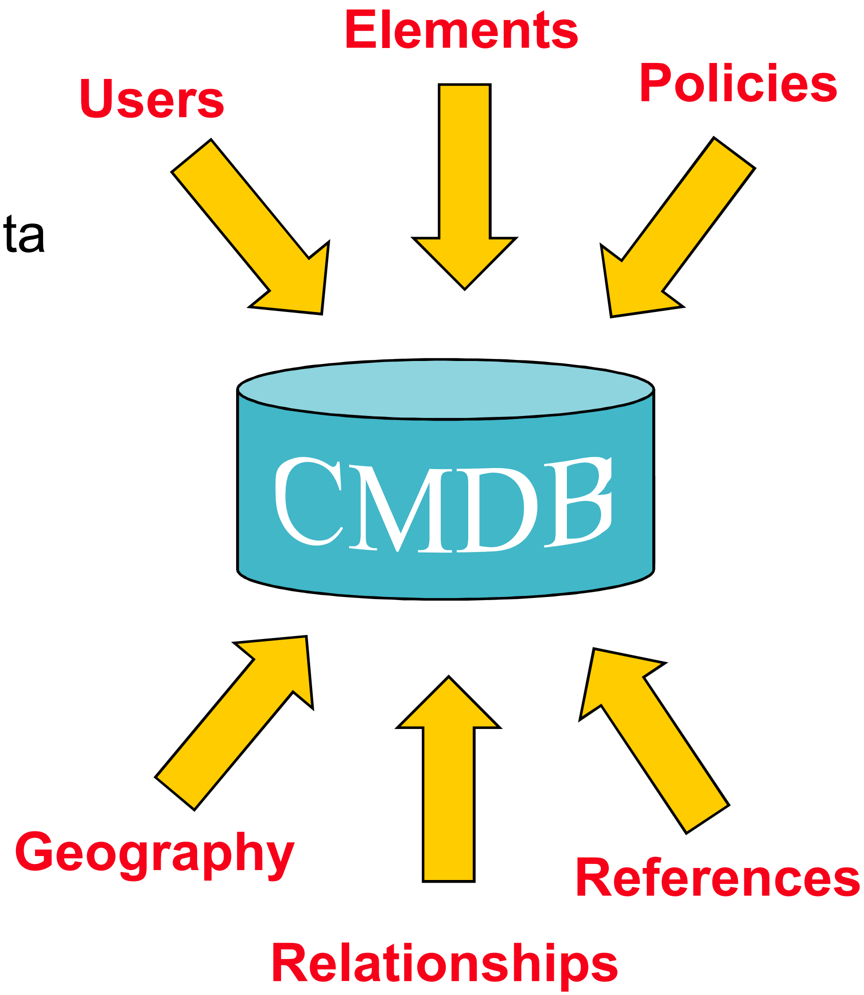
Automating CMDB management

Using the CMDB

EMC² where information lives

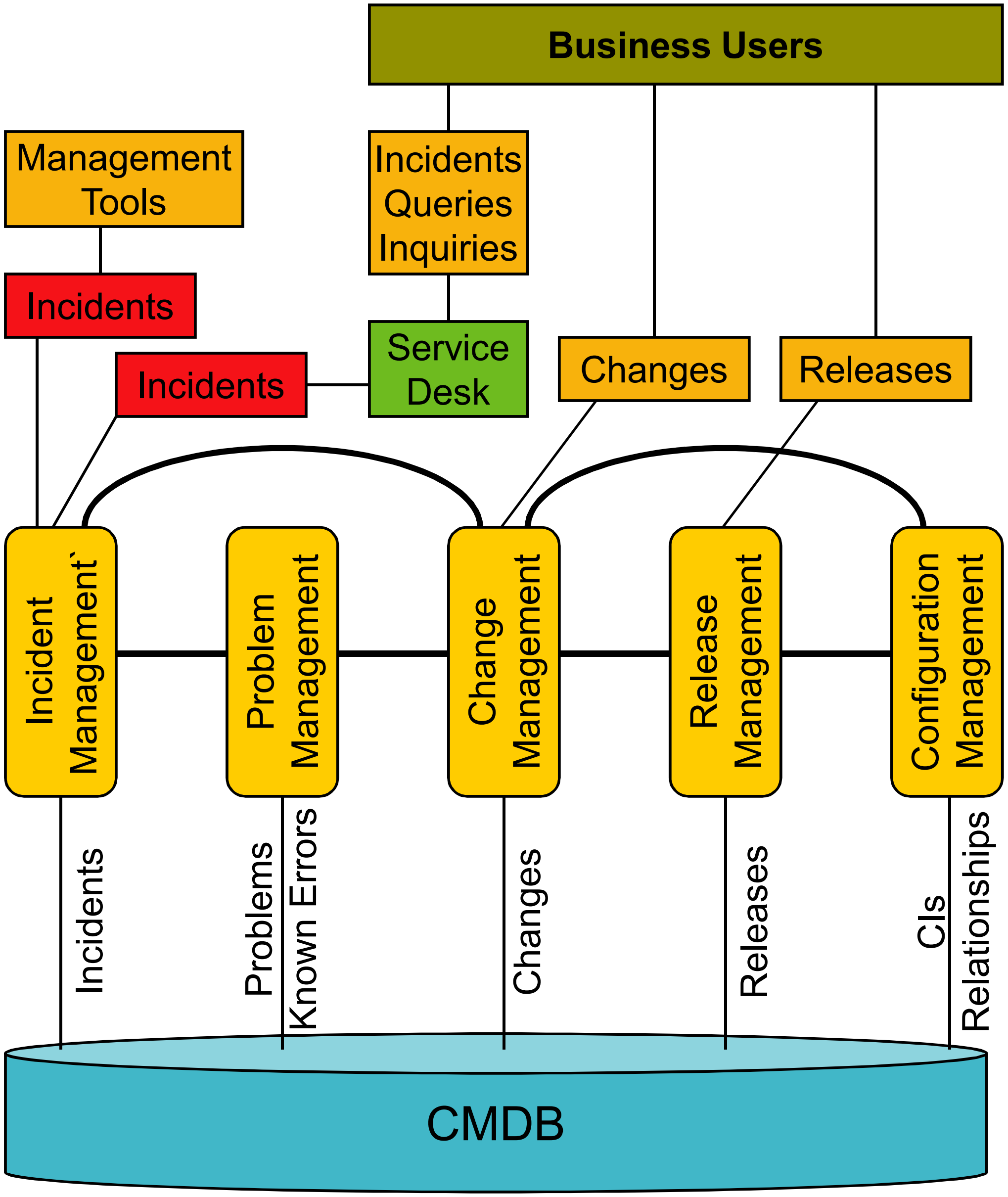
The Configuration Management Database (CMDB)

- ITIL defines the CMDB as the configuration repository
  - The central trusted store for all data
- CMDB is a misnomer
  - The correct approach is not a DB
- A federated objected model should link all trusted sources
- Reconciliation policies are needed to resolve conflicts
  - e.g., which source is THE trusted source?



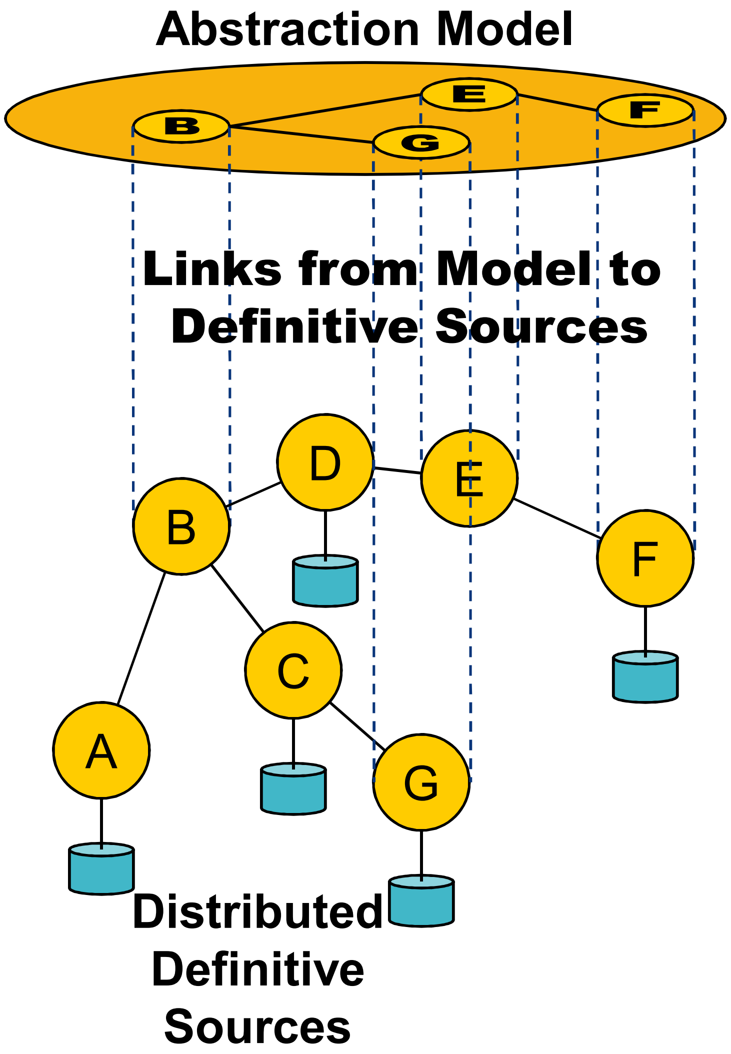
The ITIL View of CMDB

- CMDB is referenced frequently in ITIL
- Little actual definition
- No schema or model
- Use cases defined for each process
- Weak mention of discovery or data reconciliation



Federating Data Repositories

- Data is everywhere
- Consolidation into a monolithic data warehouse will not work
- Use distributed object models
- Models define high-level structure
  - Link to definitive sources
- Reconcile to real-world snapshot
- Historical data is useful, but little detail is needed


Abstraction Model
B
E
G
F
Links from Model to Definitive Sources
D
E
B
F
C
A
G
Distributed Definitive Sources

What Goes in the “CMDB” Structure?

Enough detail for the purpose!

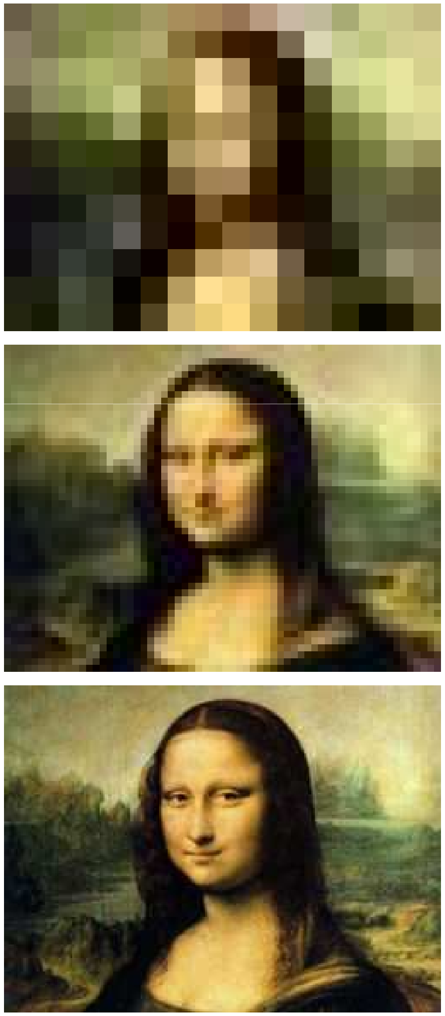
Most attempt to collect too much data

In the federated model, enough is:

- High-level Abstraction:
  Coarse relationships and links to the definitive sources
- Definitive Sources:
  Details needed to uniquely describe the source
- The Objects Themselves:
  Every detail necessary for its operation

The true definitive source is the object itself

- CMDB components help act on the data and provide a historical record



Multidimensional Context of the Relationship Map

A CMDB without relationships is incomplete

- Like an unassembled jigsaw puzzle

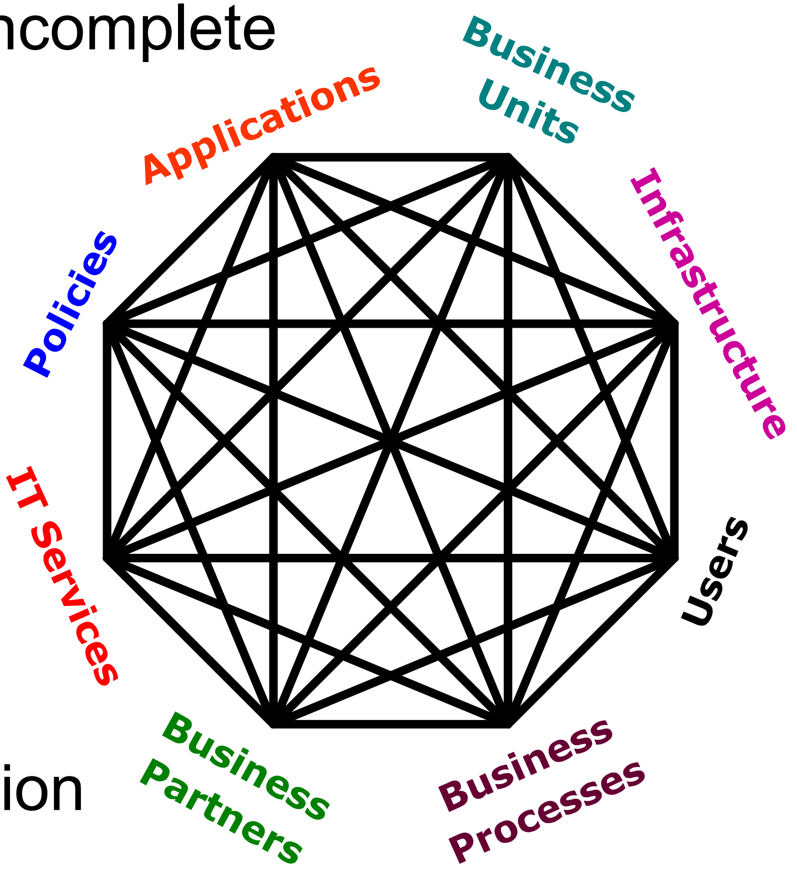
More than physical connections

Any relationship that offers context should be reflected

- Difficult to implement, however
- IT and business must define relationships at a general level

The roadmap to navigate automation

Discovery enriches the relationship map



Cracking the Enigmatic CMDB

Why the CMDB is suddenly the hot topic

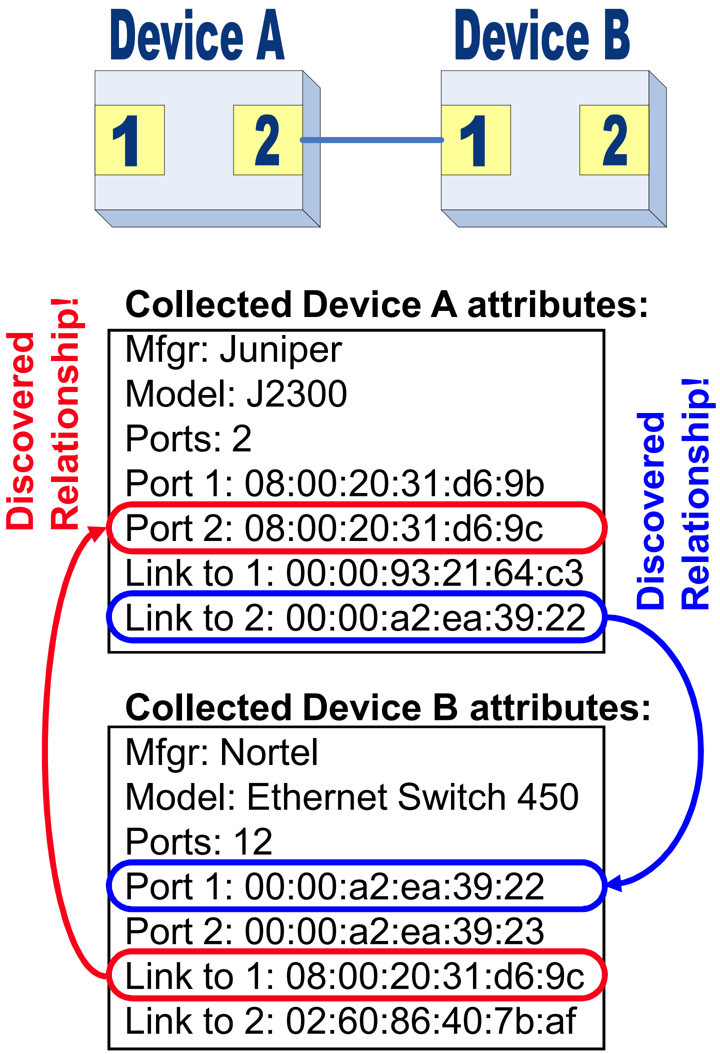
Defining the CMDB

Automating CMDB management

Using the CMDB

Automated Discovery

- The CMDB concept is not new
- Most efforts failed because of stale data
  - Manual entry and maintenance is never suitable
- The only cure is to automate CMDB population and maintenance
  - Autodiscovery is growing rapidly
- Discovery requires instrumentation
  - Some mechanism to present details
  - Networks have SNMP instrumentation
  - Others are weak or missing



Collected Device A attributes:

Mfgr: Juniper
Model: J2300
Ports: 2
Port 1: 08:00:20:31:d6:9b
Port 2: 08:00:20:31:d6:9c
Link to 1: 00:00:93:21:64:c3
Link to 2: 00:00:a2:ea:39:22

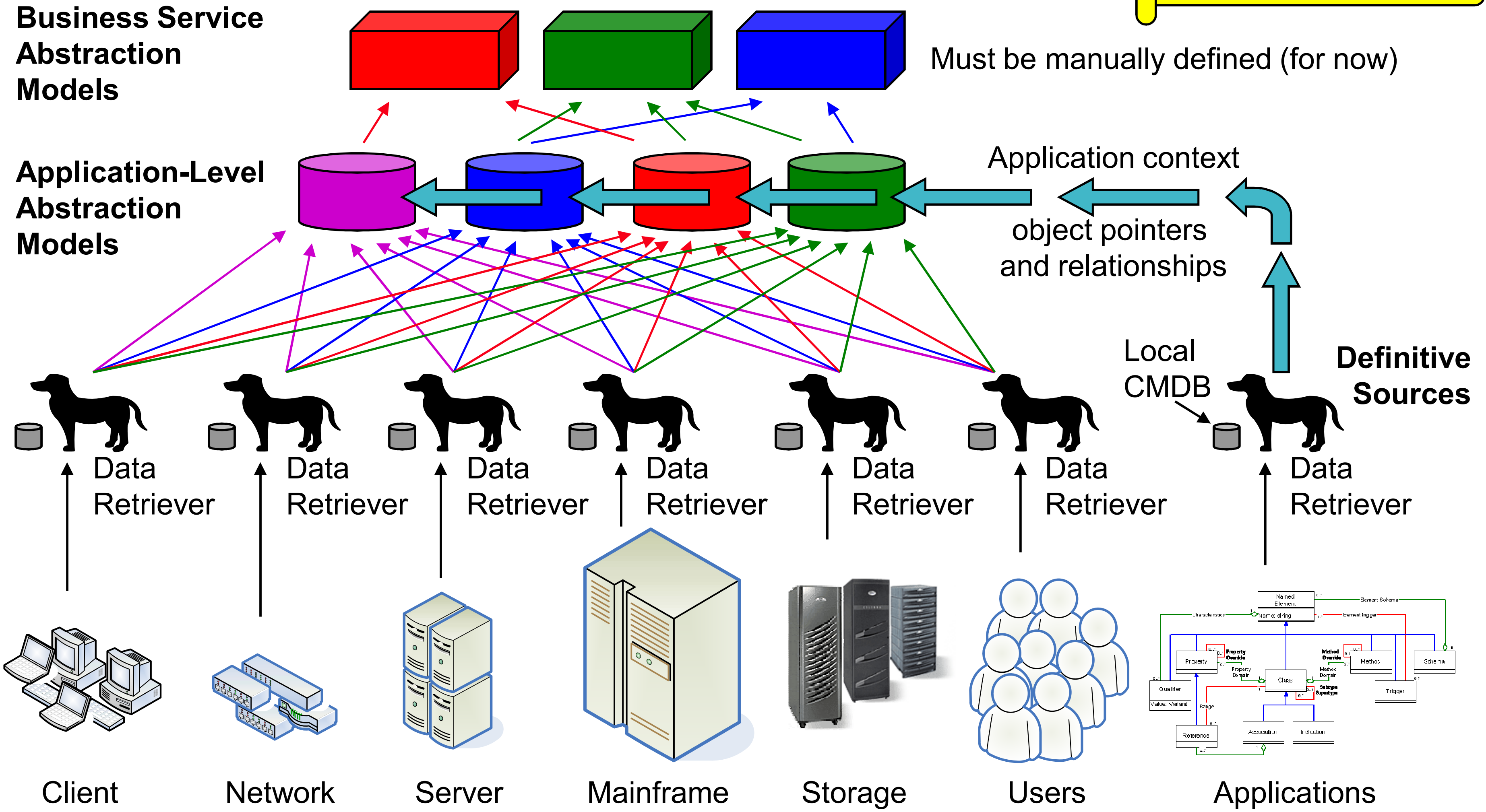
Collected Device B attributes:

Mfgr: Nortel
Model: Ethernet Switch 450
Ports: 12
Port 1: 00:00:a2:ea:39:22
Port 2: 00:00:a2:ea:39:23
Link to 1: 08:00:20:31:d6:9c
Link to 2: 02:60:86:40:7b:af

Simple Discovery Example

Technology Solutions for a Unified “CMDB”

Note: No single central repository!



Business Service Abstraction Models

Must be manually defined (for now)

Application-Level Abstraction Models

Application context

object pointers and relationships

Local CMDB

Definitive Sources

Data Retriever

Data Retriever

Data Retriever

Data Retriever

Data Retriever

Data Retriever

Data Retriever

Client

Network

Server

Mainframe

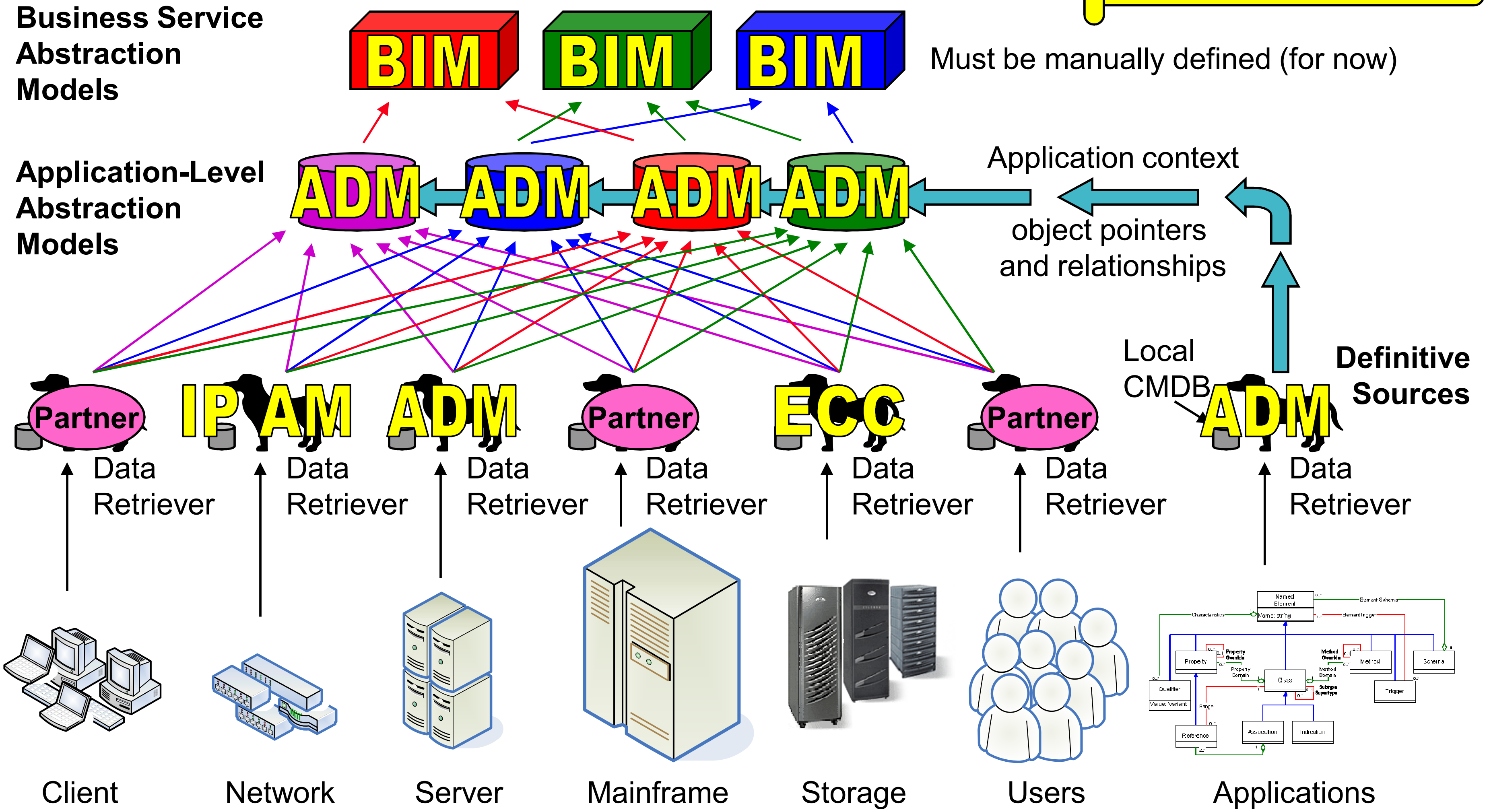
Storage

Users

Applications

Using EMC to Build an Effective "CMDB"

EMC Products (more than shown here)



Business Service Abstraction Models

Application-Level Abstraction Models

Must be manually defined (for now)

Application context

object pointers and relationships

Local CMDB

Definitive Sources

Cracking the Enigmatic CMDB

Why the CMDB is suddenly the hot topic

Defining the CMDB

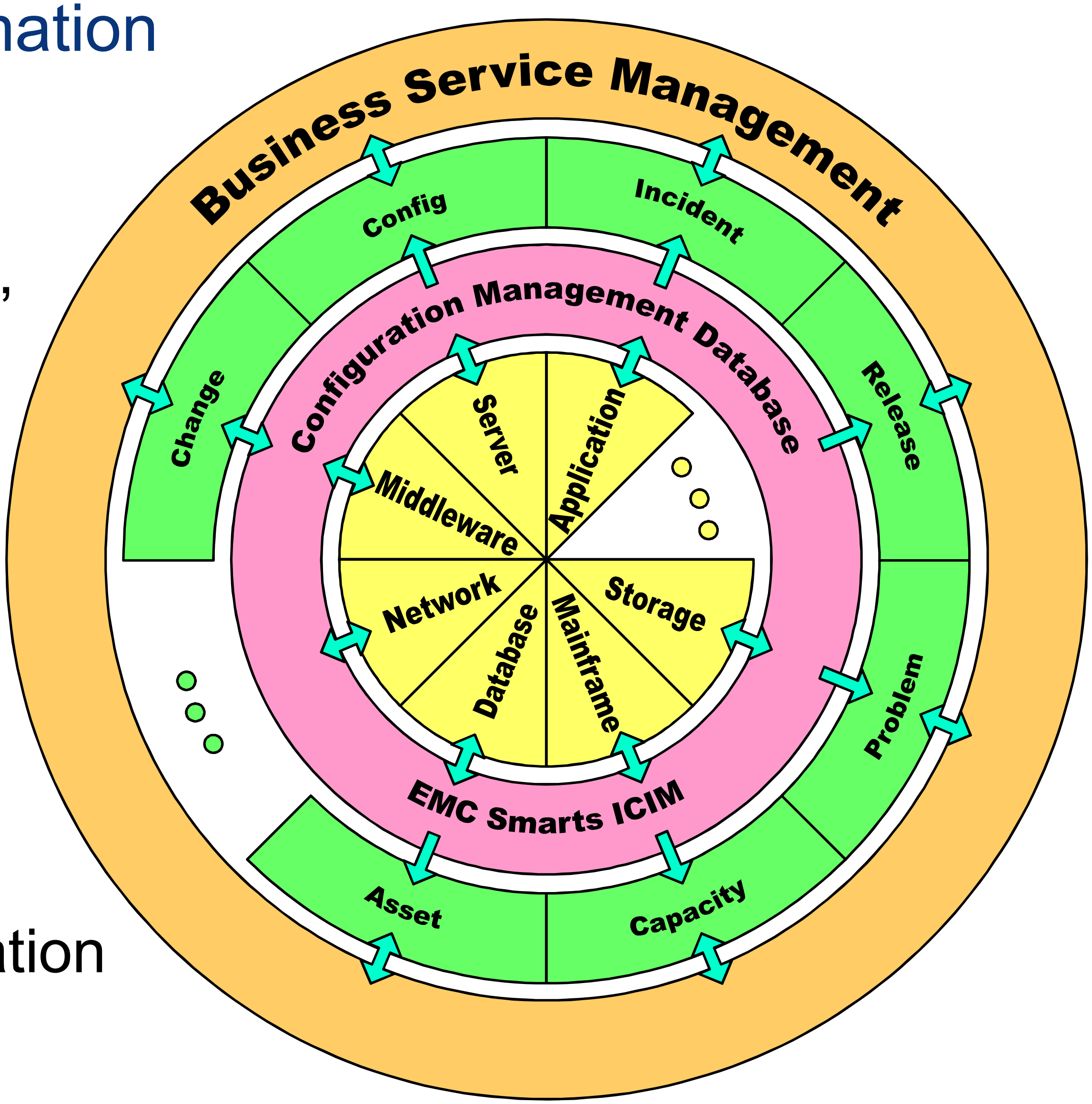
Automating CMDB management

Using the CMDB

EMC² where information lives®

Driving Service Automation

- Without effective use cases and integration, CMDB is just an academic exercise
- Use the CMDB to automate analysis
  - Analytic models are derived from configuration
- Results of analysis determine actions for other process automation



Linking to Change Management

- Operational discipline mandates regulation of changes
  - Whether human approval gates or automated checks
- Without change management, “CMDB” becomes useless
- Without configuration, change has nothing to operate upon
- Define process to account for emergency changes
- Avoid overbearing “bureaucracy” for routine changes
  - But log EVERY change for historical analysis
- Analysis models are used as sanity checks for changes
  - Test before implementing change

Action Plan for “CMDB” Development

Appoint and empower a “Configuration Czar”

- Somebody must be accountable, but also able to dictate action

Align with ITIL, but clarify the ambiguities

Pursue a federated model, but understand federation

Use object-oriented modeling technologies as the “glue” to bind the disparate sources into a unified (and useful) view

Leverage application discovery as context for building application-level abstractions

Couple CMDB with change management (MANDATORY!!)

Be pragmatic about the state of technology evolution

- We are still in the early days of configuration automation

EMC^2®

where information lives®