Discovery, Control, and Monitoring

David Merbach

Agenda

- Issues with Storage Discovery and Control
  - Queries
  - Locations
  - Schema
  - Detectability
- Discovery Engine
  - Flow
  - Engine
- Control Engine
- Monitor
  - Change Detection

Discovery Issues

- Inconsistencies between devices
  - Performance, Errors, etc. requiring some unique processing
  - Handle in ‘Process Block’ and ‘DB Mapper’
- Scaling
  - Large amounts of data can be returned by SMI-S calls or other functions
  - Handle in SAX parsing, chunking
- System limitations
  - Memory, CPU cycles, etc. will become bottlenecks as larger demands are made
  - Handle in queues, ‘job info’ summary (for analysis), Job Filtering, Distribution

Discovery – location of ‘collectors’

- SMI-S

  Used to collect Subsystem and Fabric information

- Outband Fabric Discovery

  SNMP and Brocade API

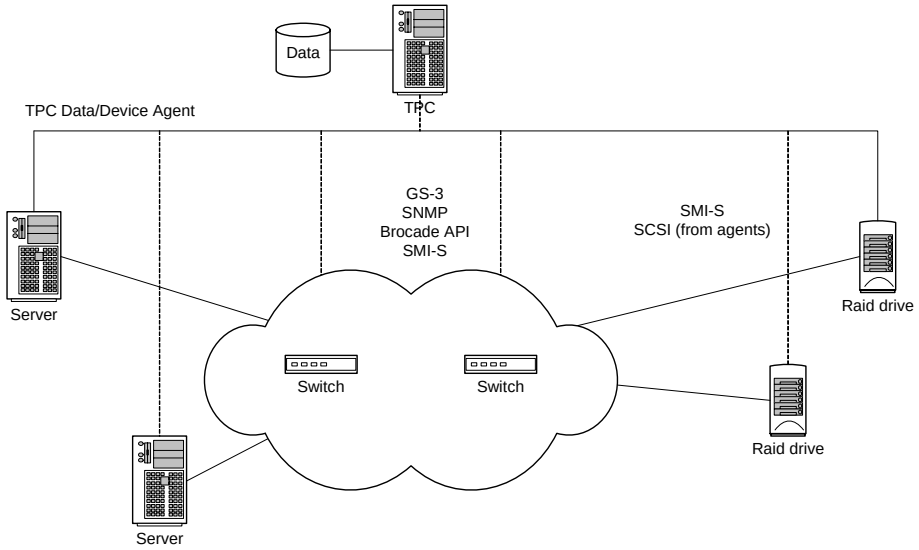
  Used to collect Fabric, Switch, Connection information

- Inband Data Discovery

  Collect Host and File level information for SRM reports

- Inband Fabric Discovery

  Collect Host, Port and Device information



One off Processing

- SMI-S defines standard algorithms
- Reasons for customizations

Discovery	Control
Some switches optimize with enumerate instance traversals Others faster with with association traversal	Some subsystems allow groups of hosts to be masked or mapped to a volume, others do not Association Traversal is different in the above case
Different traversal for SMI-S 1.02 vs. 1.1	Different traversal for SMI-S 1.02 vs 1.1 for LUN masking
Overcome specific issues in a vendor CIM/OM	Vendor-specific extensions are present in control too e.g getSupportedSize()
Vendor specific extensions have relationship information in fields, which can be used to get association info in 1 query instead of 2	

Data Source Modeling Complexities

- SMI-S model is very extensive

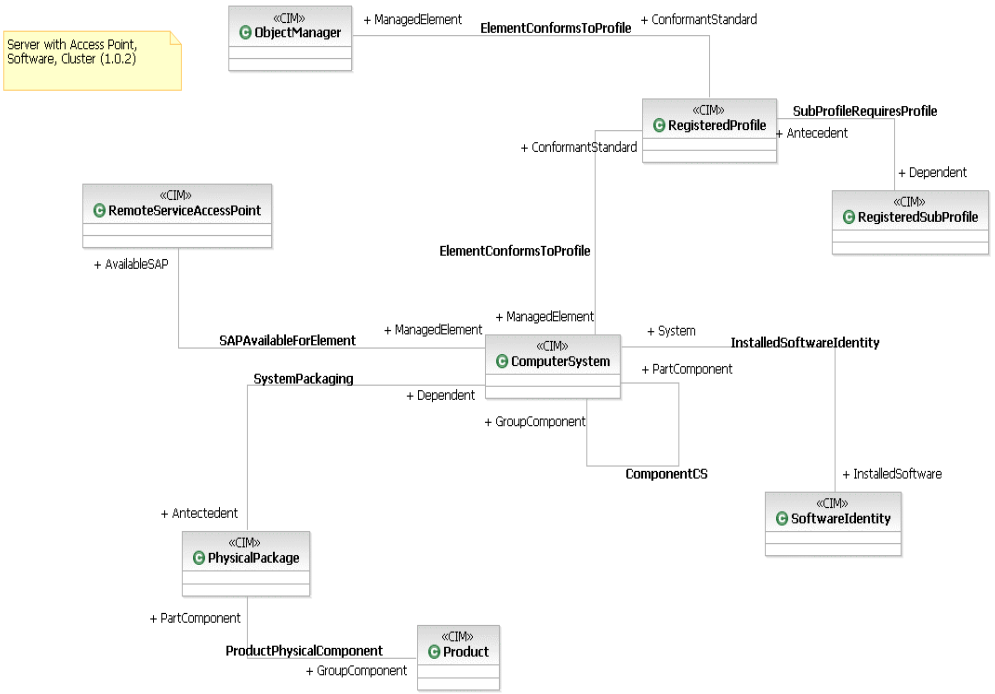
  Many profiles and subprofiles for subsystems, hosts, mapping assignments, fabric, clusters, etc.

  Model is very flexible to handle vendor differences

  Many auxiliary classes

  Vendors can provide extensions to the mode

  - Extensions for specific aspects
  - Extensions for specific fields
  - Extensions for controls



Detectability- Overlapping Collection

- Multiple collectors may be able to detect a ‘thing’
- Inability of a single collector does not mean that the thing is not detectable by other collectors
- Example:

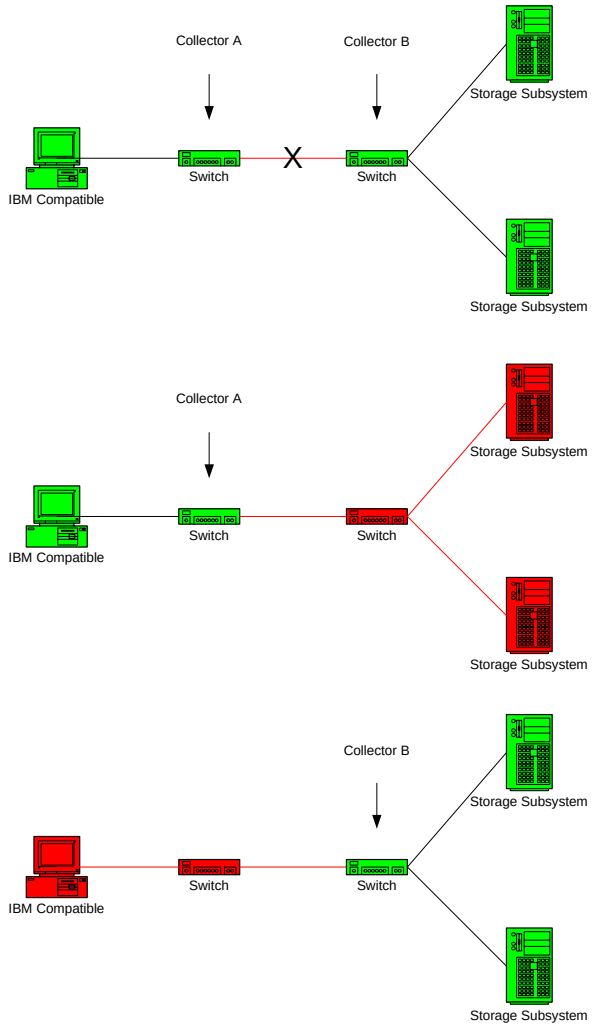
  Fabric collectors can get switch, connection and some host/device info (e.g. ports/nodes)

  A failing connection between the switches will segment the fabric

  Each switch will only report the information it can detect

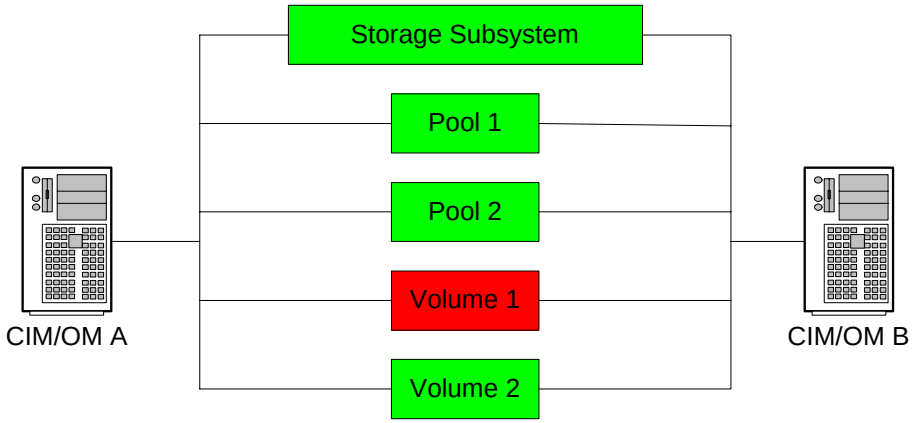
  If ANY switch reports then device present

  If NO switch reports then device is missing

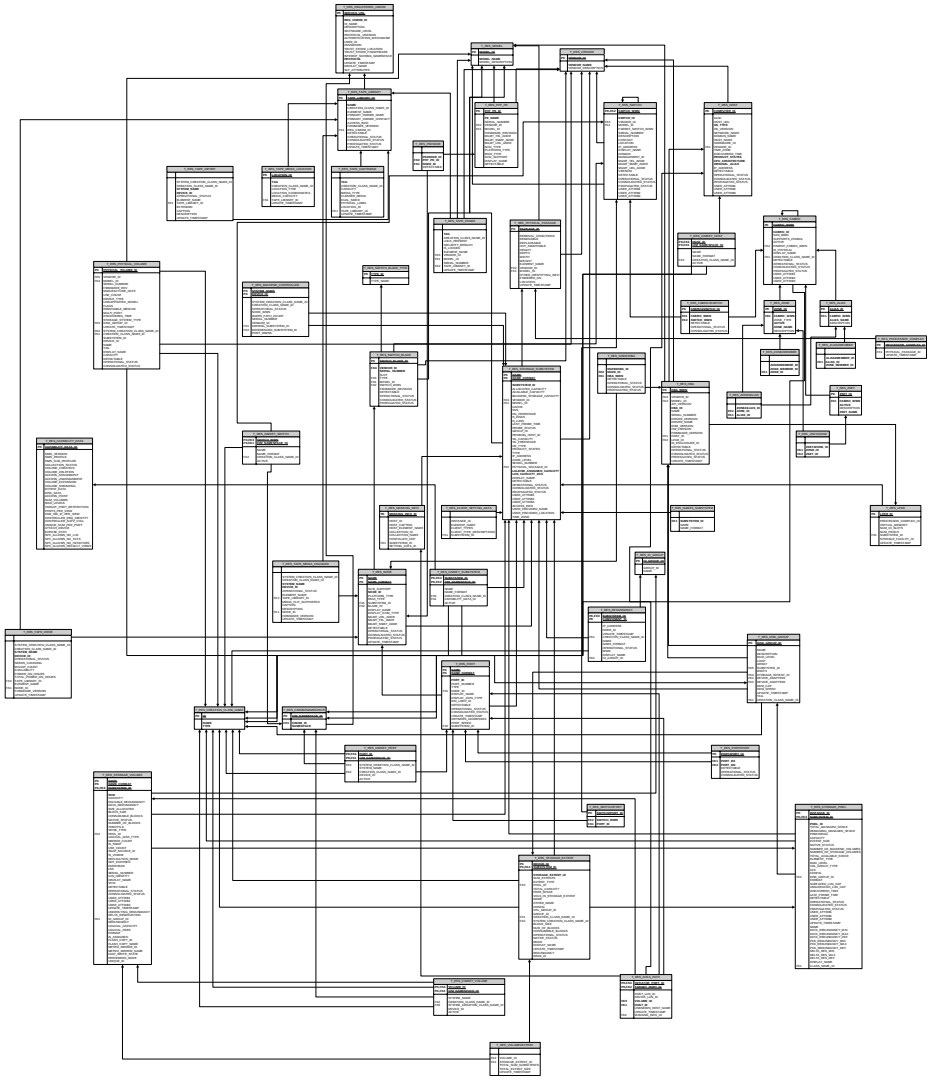


Detectability – Authoritative Source of Information

- Multiple collectors may be able to detect a ‘thing’
- Each collector has same capability
  - If it can detect the subsystem can detect the pools and volumes in the subsystem
- Example:
  - CIM/OM A and B both managing subsystem
  - Discovery and control operations can be performed with either
  - Removal of a volume is detectable by either



Schema

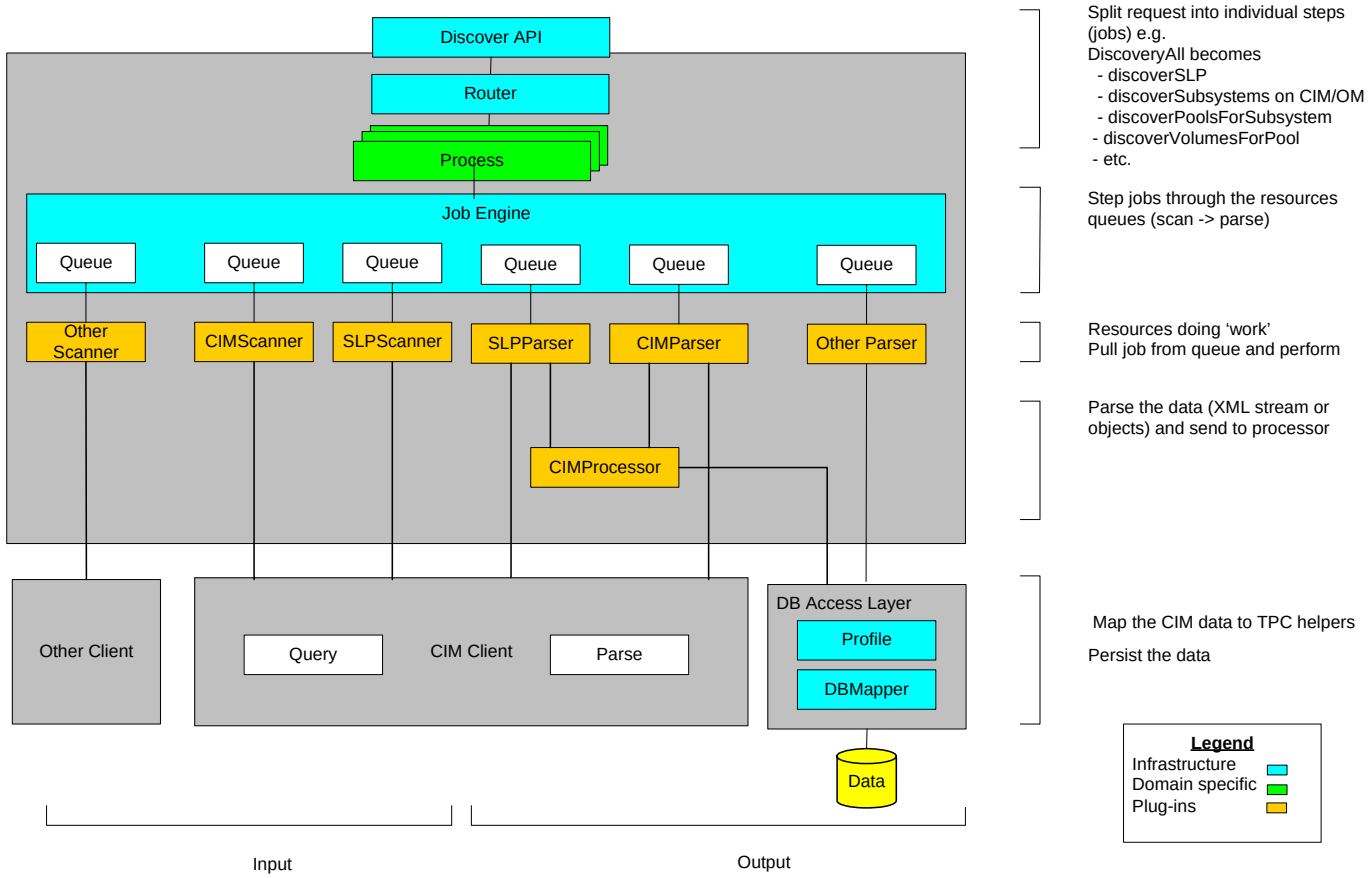


Discovery

Discovery Design Points

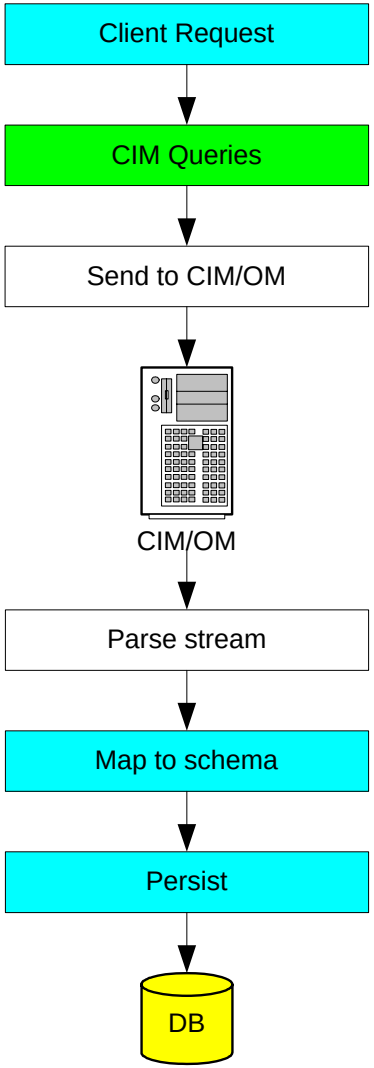
- Discovery may be segregated, but coordinated
  - Run on differing management servers
  - Distributed to agents
- Output may be in differing forms
  - Direct database insert
  - XML or other format to a file (e.g. for later upload)
  - Streamed over a communications session
- Input may have differing formats
  - ESSN parsing
  - Brocade API
  - CIM XML
    - Differences in elements/ordering between vendors
  - JDBC
    - Used to migrate from legacy databases or to federate/roll-up database information

Architecture



Discovery Flow

- CIM Queries
  - Order that queries are constructed and put together
  - Handle differences in way queries put together
- Communicate with CIM/OM
  - CIM Client streams commands to CIM/OM
  - Parses XML response stream
- Map data to Output
  - Normalize to DB schema
  - Handle different types of responses
  - Can do other types of output



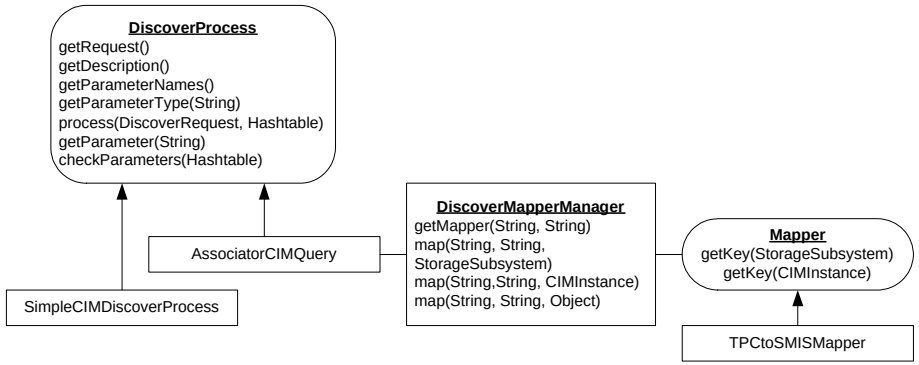
Process

- Generalized algorithms for most devices
- Need to account for specific functions
  - Handle errors/differences in vendor implementation
  - Handle performance issues in implementations
  - Handle differences in SMI-S versions
- Mapping requests to processes
  - Simple name mapping
    - ‘myProcess’ → Process Implementation
  - One off Mapping by parameters
    - CIM/OM info
    - SMI-S version
    - Profile

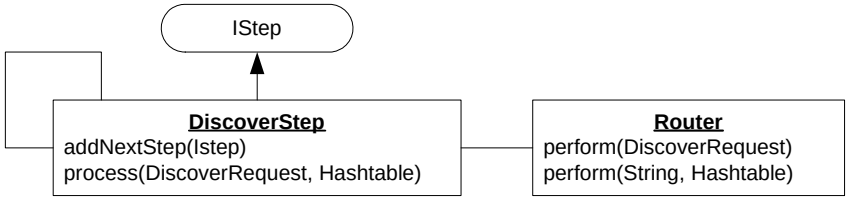
Cisco CIM/OM functions	Brocade CIM/OM functions	Default Functions
		Default ‘Get Fabric Info’ function • call ‘get SANs’ • call ‘get Switches’ • call ‘get Nodes’
	Modified ‘Get SANs’ function	Default ‘Get SANs’ function
		Default ‘Get Switches’ function
Modified ‘Get Nodes’ function		Default ‘Get Nodes’ function
		Default ‘Get Ports’ function
		Default ‘Get Port 2 Port’ Function

SMI-S Processes

- Need to be able to specify/restructure SMI-S commands easily
  - Need to traverse through model to get info
  - Order matters (1 CIM/OM may be faster with one traversal, another faster in a different order)



- Provide ‘building block’ processes
  - SimpleCIMDiscoveryProcess – issues a set of SMI-S commands (mainly for enumerate instance calls)
  - AssociatorCIMQuery – maps an input parameter to the ‘base’ of the traversal
  - DiscoverStep – specifies a process to perform, enumerates through responses and calls any ‘next steps’



Queue

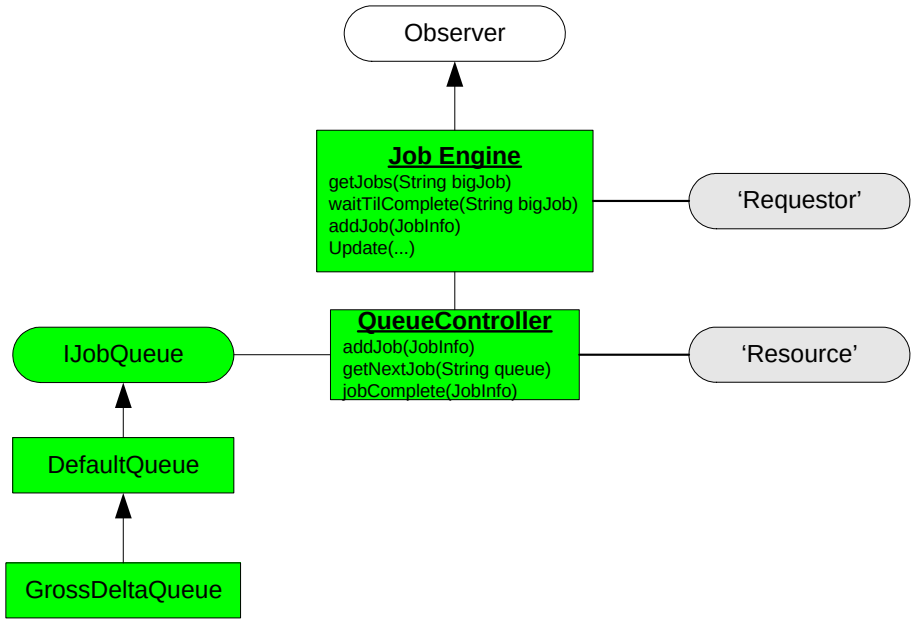
- Requestor (e.g. process or client) adds jobs to the engine

- Resources (e.g. CIM Scanner) wait for jobs
- Queue is generic

- Plugable Queues allows custom operations

  For ‘CIM Scanner’ want to discard new requests if one is pending in the queue

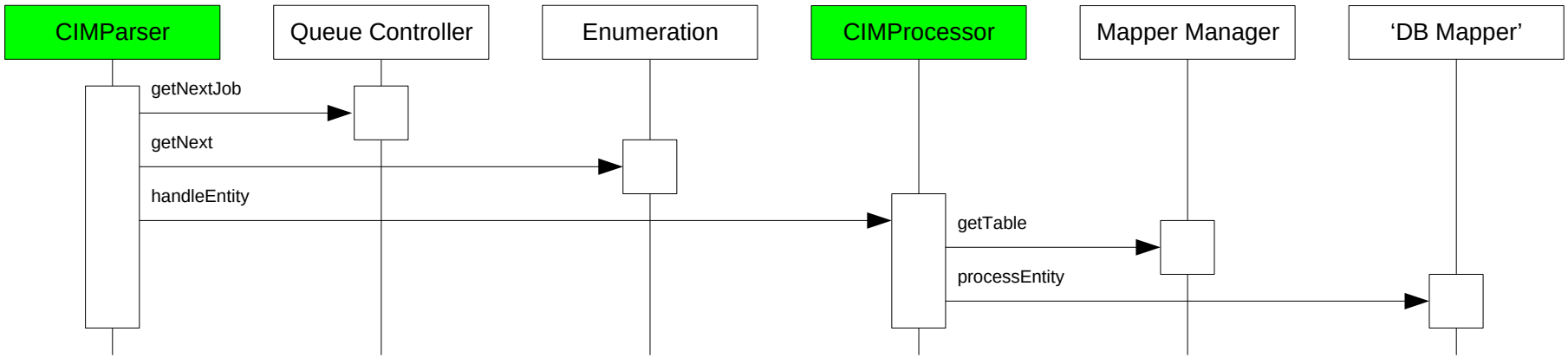
  For ‘CIM Parsing’ want to replace older requests with newer requests (which has newer data)



Parser

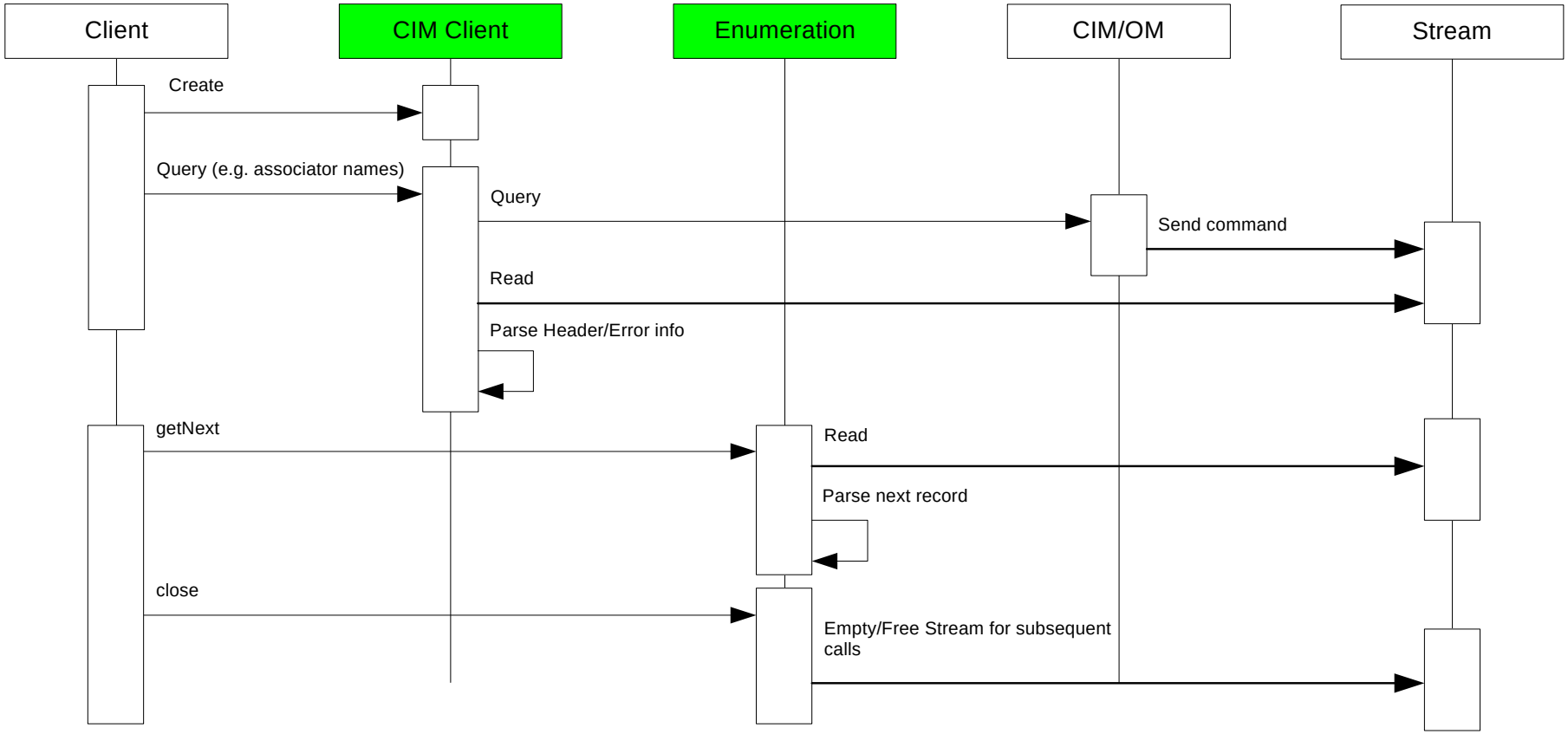
- General Process
  - Parse
    - Parse data from source
    - SMI-S Parser – uses CIM Client to parse
    - Fabric Legacy Parser – SAX parser
  - Send to DB Mapper
    - ‘Processor’ is basic block that wrappers access to DB Mapper
    - Extension for SMI-S and Legacy data

- CIM Processor incorporates request information as applicable
  - Associator response only contains decedent info
  - CIM Processor adds calls/info to process Antecedent and Associations



CIM Client

- SBLIM (IBM donated) CIM Client adds:
  - SAX parsing
  - Object re-use



Detectability

- TPC uses many information sources (scanners) to collect SAN information. E.g.:

  CIMOMs: storage subsystem, switch

  Fabric agents: host-based inband, out-of band dusing SNMP or proprietary APIs

  Data (TSRM) agents: host-based, inband.
- Authoritative scanner: if it does not report element it previously reported, its gone
- Non-authoritative: No such conclusion can be drawn from report of single scanner

  Consensus: element declared as “missing” only if all scanners that reported it in earlier scans cannot see it any more
- Detectability Service: works with Discovery Engine to track what’s “missing”