

SMI-S Client and Server Development, from Check Box to Industrial Strength

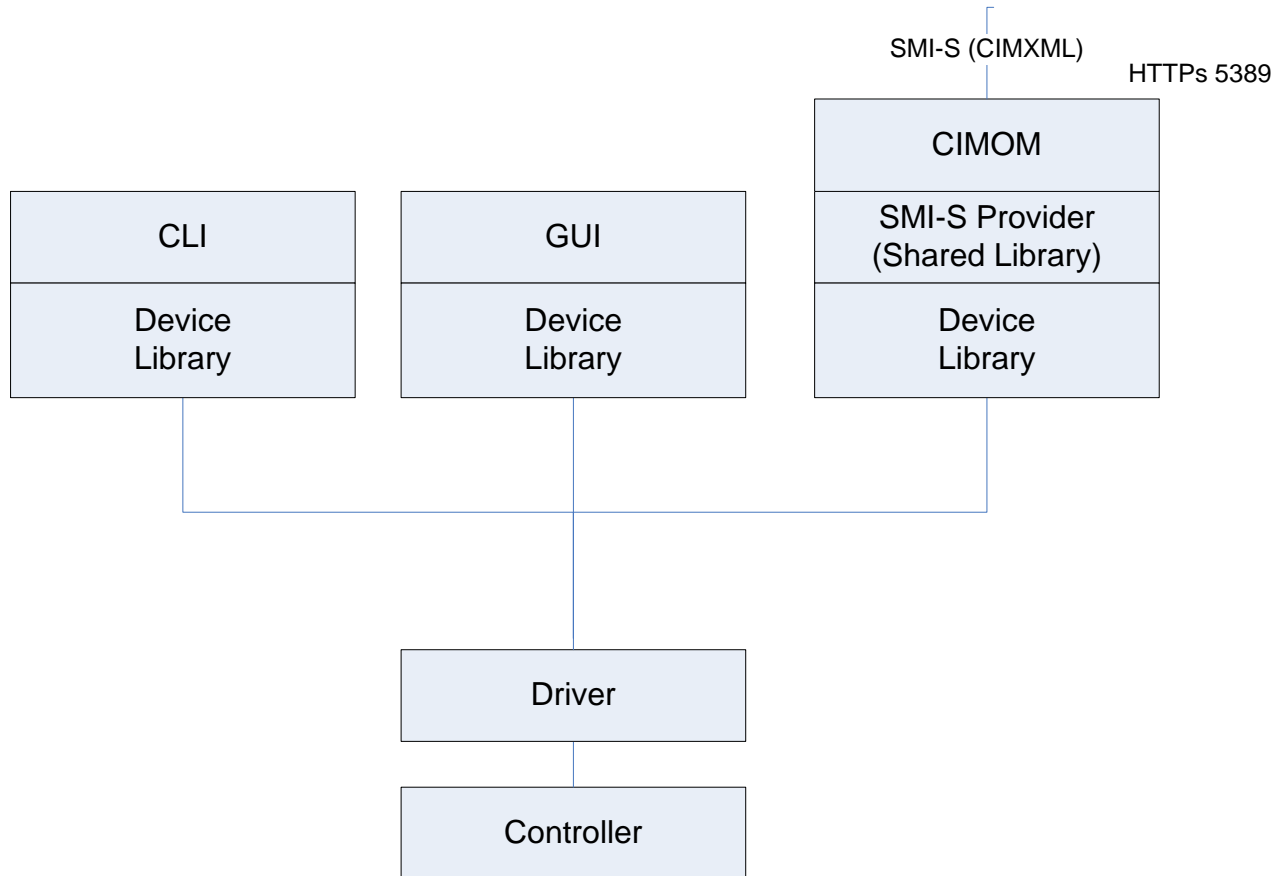
Steve Peters

PMC-Sierra

What a checkbox implementation looks like

- ❑ Polling
 - ❑ Everybody polls
 - ❑ Updates take forever
- ❑ No caching
 - ❑ Direct access for all data
- ❑ No View Classes
 - ❑ Just more classes to implement
 - ❑ Can get certified without it
- ❑ No integrated whole
 - ❑ SMI-S is just for “third party access”
 - ❑ We use native interface for our own

Check Box



How to get to Industrial strength

- ❑ Events and Indications
 - ❑ Nobody polls
 - ❑ Push the updates
- ❑ Caching
 - ❑ Off load device
 - ❑ Improve performance
- ❑ View Classes
 - ❑ Reduces CIM operations
 - ❑ Simplifies the model
- ❑ Extra features
- ❑ The Stack
 - ❑ Putting it all together as an integrated whole

❑ Checkbox

❑ GUI, Provider, and CLI all poll for information

- ❑ Polling places Load on driver

- ❑ Change in one takes time to reflect on other

❑ Industrial

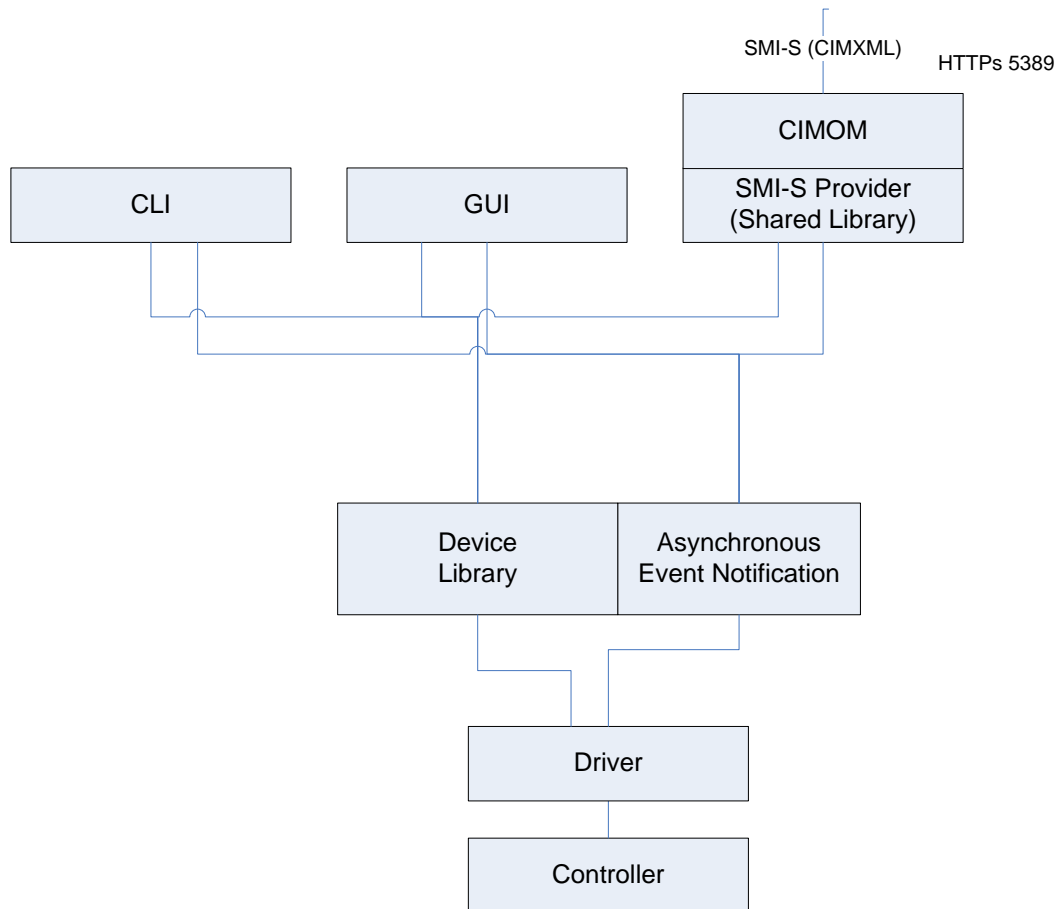
- ❑ Driver interface sends update notices

- ❑ Alternate is for one process to poll and send events to everyone Subscribed for events (indications)

- ❑ Central place to log or distribute events

- ❑ Two types of SMI-S indications
 - ❑ Lifecycle – management data state change
 - ❑ Used to update management data
 - ❑ Alerts – messages to the user
 - ❑ Logged locally
 - ❑ Displayed by GUI or CLI
- ❑ A single real world event can create both

Indications added

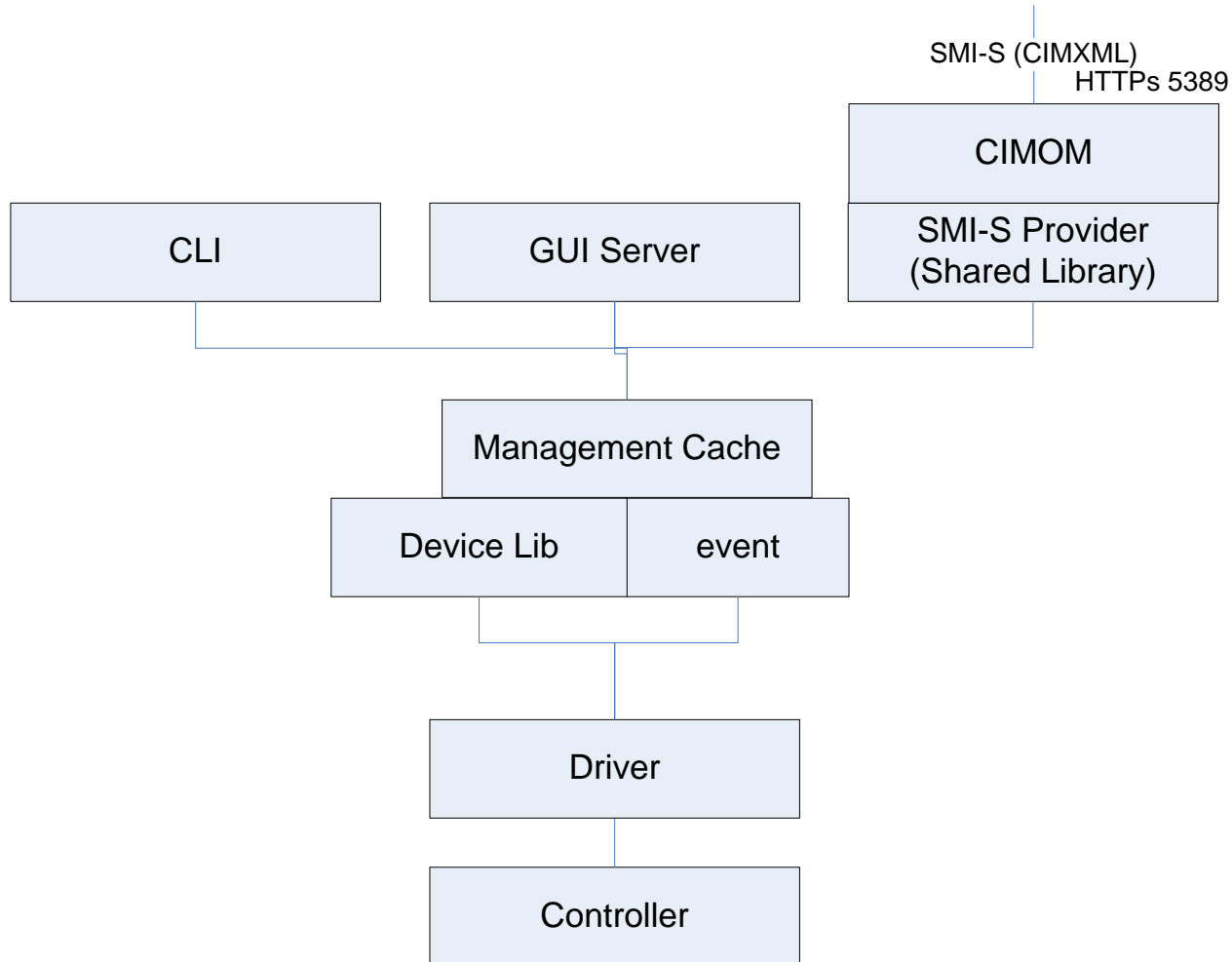


- ❑ Checkbox - directly access the device
 - ❑ CIM commands places Load on driver and device
 - ❑ Devices may be slow to respond
 - ❑ It has I/O to do
- ❑ Industrial
 - ❑ Central cache for management data
 - ❑ Quick response to requests

Provider Management Data Cache

- ❑ Keep data in device format
 - ❑ Convert to CIM when accessed
 - ❑ Most efficient format – no need for CIM static data
- ❑ Synchronized with the device by events
 - ❑ Changes trigger lifecycle indications
- ❑ Write through cache methods and property settings

Caches

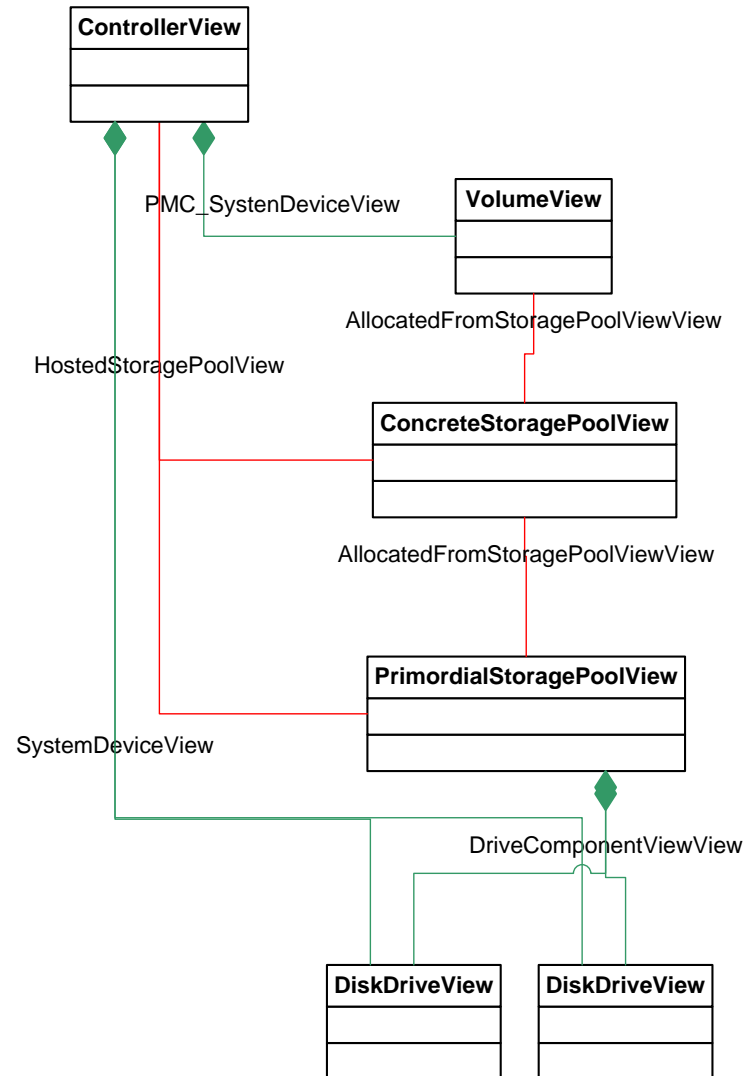


- ❑ Checkbox
 - ❑ Implement the basic commands
 - ❑ Not required to get certified
- ❑ Industrial
 - ❑ Operations are expensive
 - ❑ Simple model

- ❑ View Classes makes Clients job fast and easy
 - ❑ I0 to I reduction in operations
 - ❑ Performance
 - ❑ Less code
 - ❑ Lifecycle indications on views reduces indications
 - ❑ One get instance call updates cache
 - ❑ Simplified Model

- ❑ Sample - Get Disk drive
 - ❑ Checkbox - 6 objects and 5 associations
 - ❑ Get all disks – 6 operations per disk
 - ❑ View – 1 Object
 - ❑ Get all disks – one CIM operation

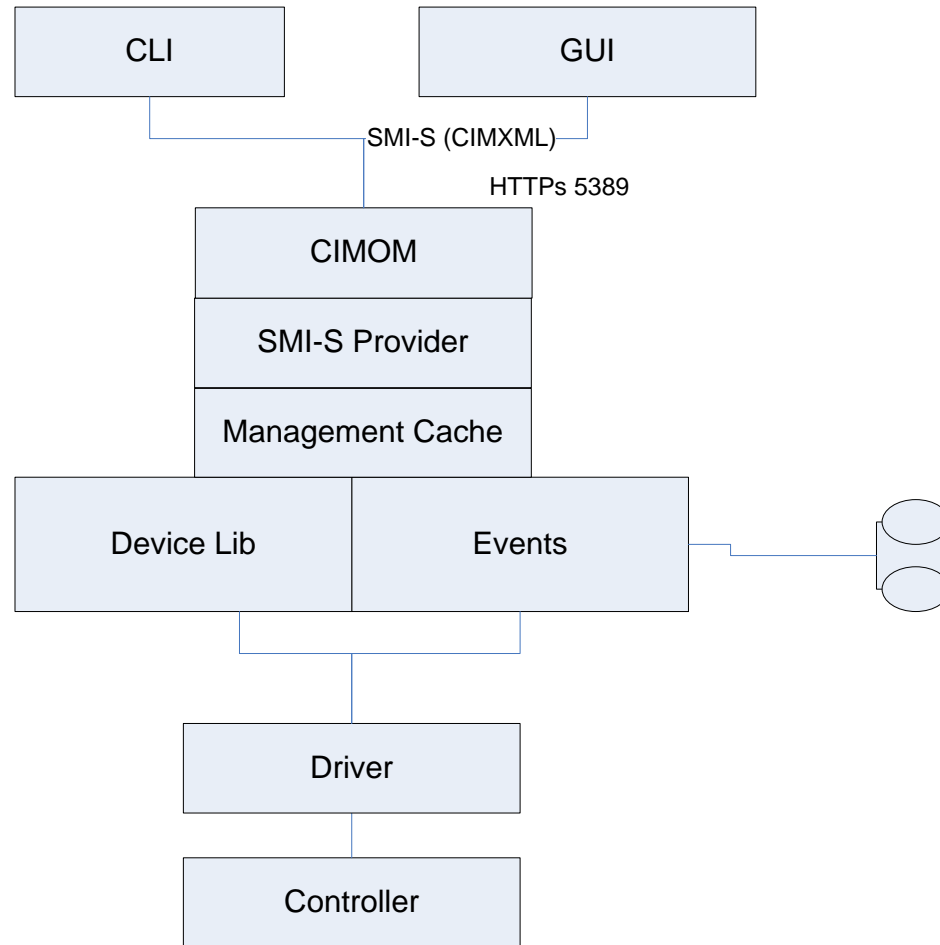
HHRC in Views



Added features

- ❑ CMPI interface allows use of different CIMOMs
 - ❑ Support as many environments as possible
- ❑ WS-man / CIMxml
 - ❑ Most CIMOMs support both protocols
- ❑ Authenticate with OS
 - ❑ Default is CIMOM user database
 - ❑ OS based authentication makes it easy to manage users.

□ Now it is industrial strength **USE IT!**



- ❑ Driver interface for properties, signals for events
- ❑ Provider and CIMOM
 - ❑ Event Notification, logging
 - ❑ Device library
 - ❑ Cache
 - ❑ Provider (instance, indication, method, ViewClass)
 - ❑ CMPI interface
- ❑ GUI and CLI uses SMI-S