Cisco live!

What You Make Possible

TOMORROW starts here.

CISCO

Automating UCS for System Administrators

BRKCOM-1004

Session Agenda

- System Overview
- Management Information Model
- Documentation and Tools
- API Methods, Responses and Filters
- Working Example

System Overview

Unified Computing System

UCS Physical Components

- Servers – blade and rack mount
- Server BIOS and firmware
- Disk and disk controllers
- Interface cards (Adapters)
- Integrated Management Controllers (CIMC)
- IO Modules and Fabric Extenders
- Fabric Interconnects
- Ports
- PSU
- Fans
- Blade Chassis
- SEEProm
- Software Images
- Expansion Modules
- UCS Manager

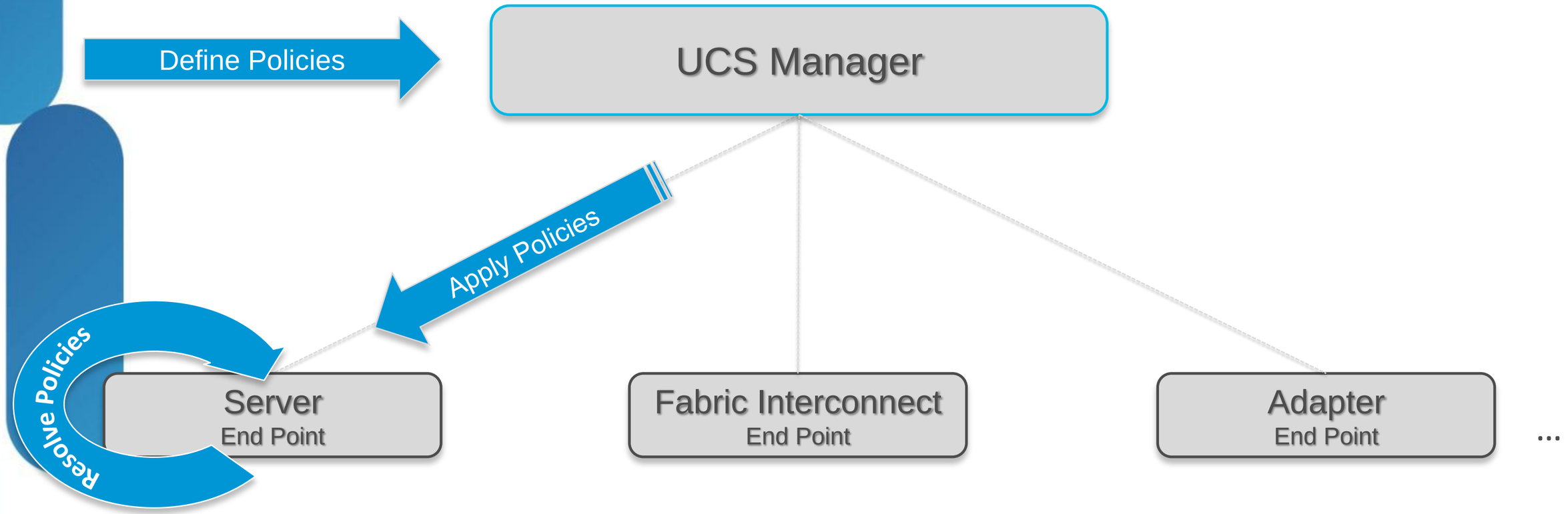
Unified Computing System

UCS Logical Components

- Service Profiles
- Policies
- VMs
- VLANs
- VSANs
- Templates
- Pools
- Port profiles
- Licenses
- Events
- Orgs
- Admin Configuration
- Files
- Port Channels

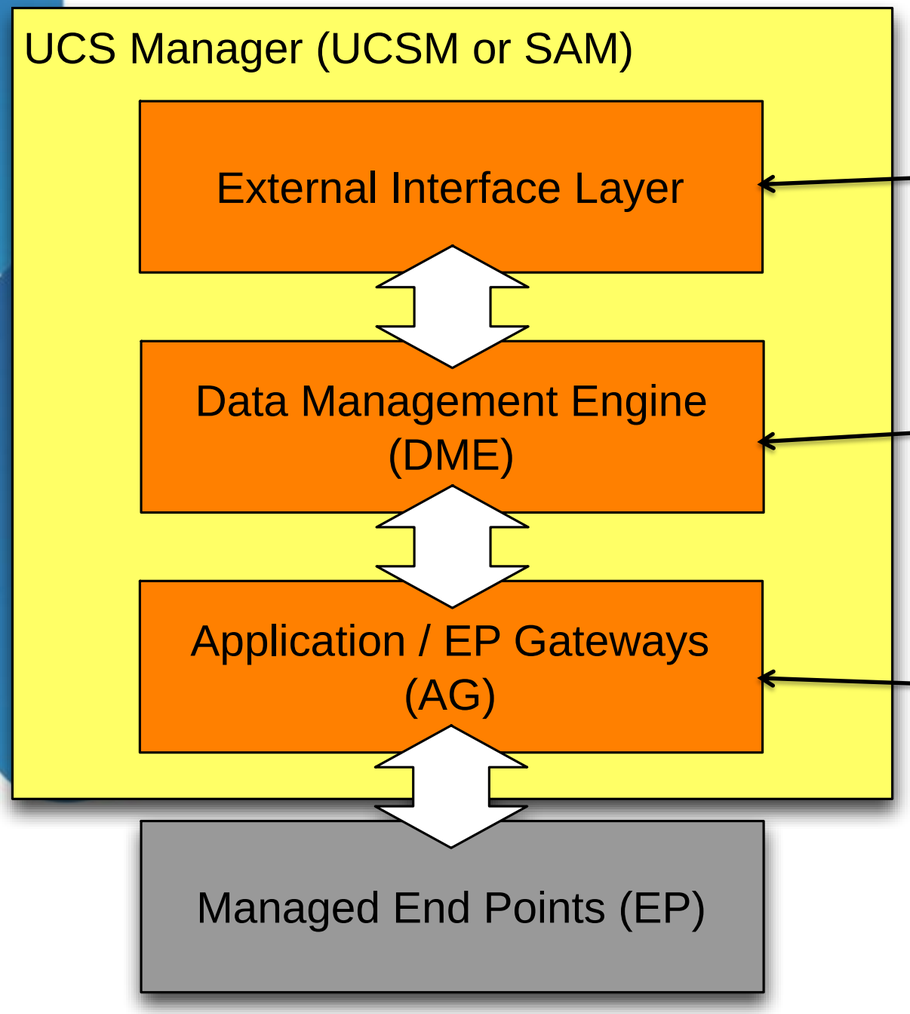
UCS Manager

Policy Based Management



- UCS Manager is the policy manager
- End points resolve the policies defined in UCS Manager

UCSM Overall Architecture



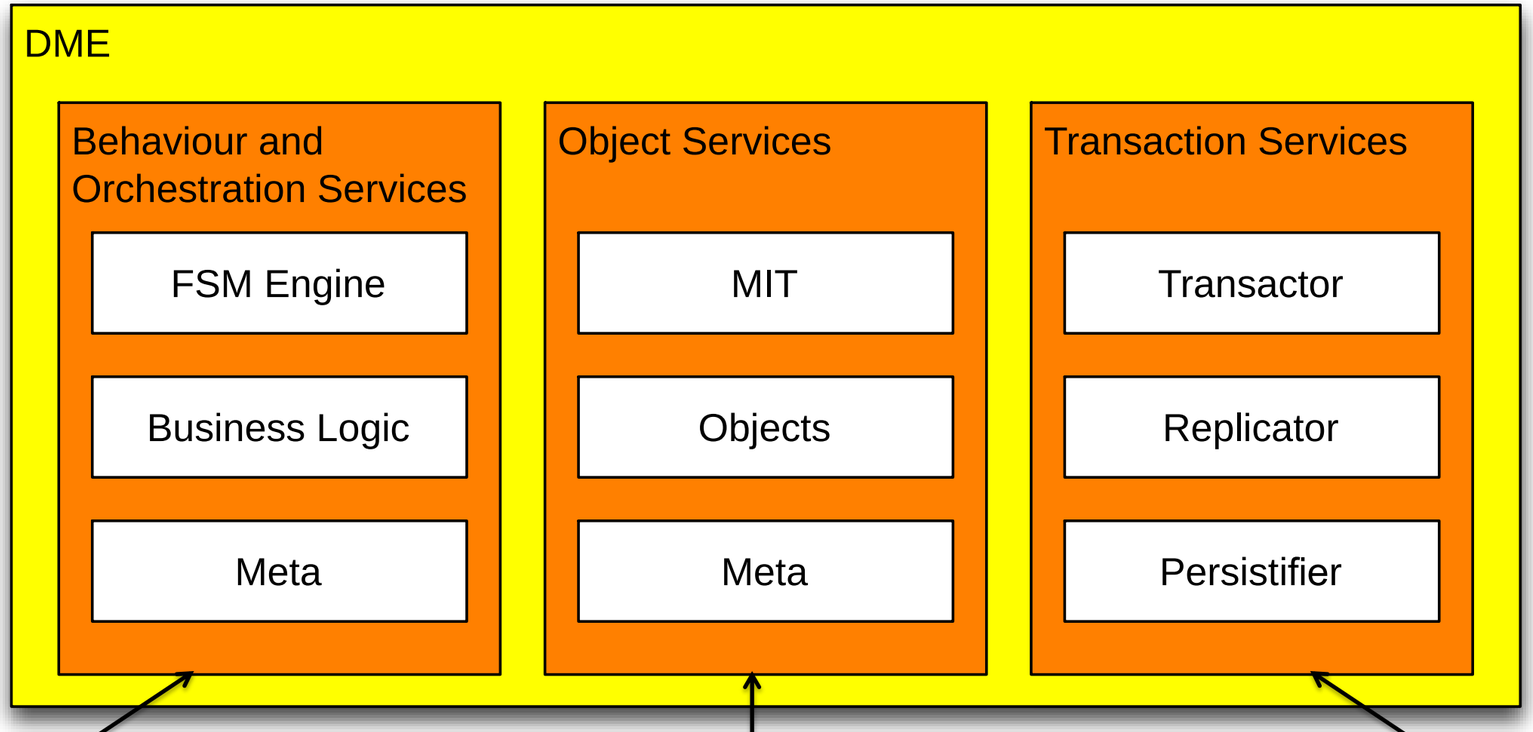
- External communication
- Specific protocol components
- Stateless
- UCSM-CLI, SMASH CLP, CIM-XML, ...

- Central component
- Multiple internal services (next slide)
- Stores desired configuration and state in MOs (Managed Objects)

- Stateless agents
- Hardware abstraction for MOs
- Propagate changes to between DME and EPs
- Host AG, NIC AG, Blade/Chassis AG, Port/NXOS AG

Note: you will often see documentation and objects refer to “SAM”
SAM (Server Array Manager) = UCSM

DME Layer

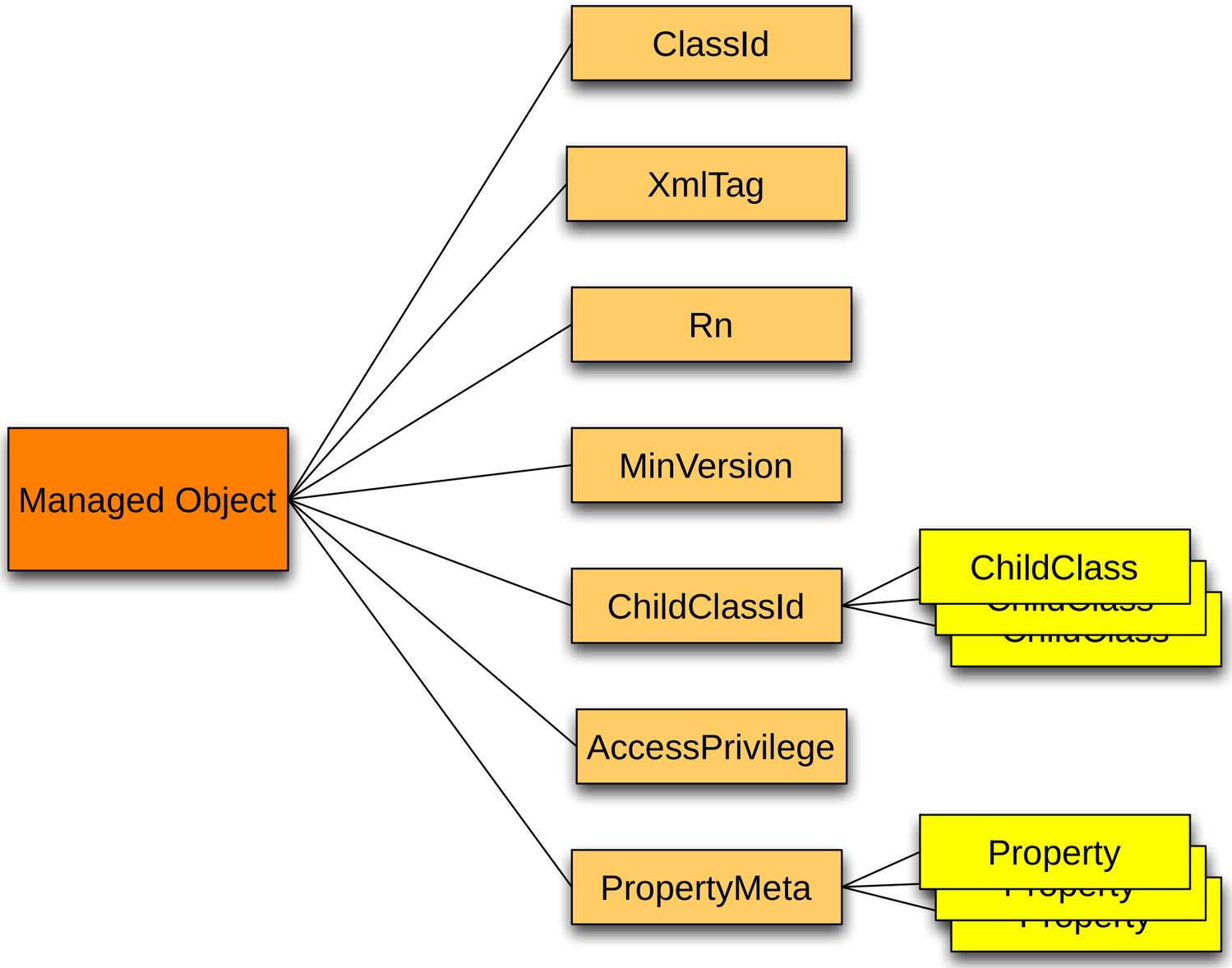


- Meta-model driven
- Object rules and behaviour
- MOs have a FSM child object
- Task schedule and execution

- Meta-model driven
- HW & SW components
- Stats, faults, events
- Model / Management Information Tree

- Object data mutations
- Replication to HA secondary
- Persistent storage
- Asynchronous and transactional

MO Meta Model



AG Layer

- Stateless agents
- Convert between native and MO representations

Application Gateways

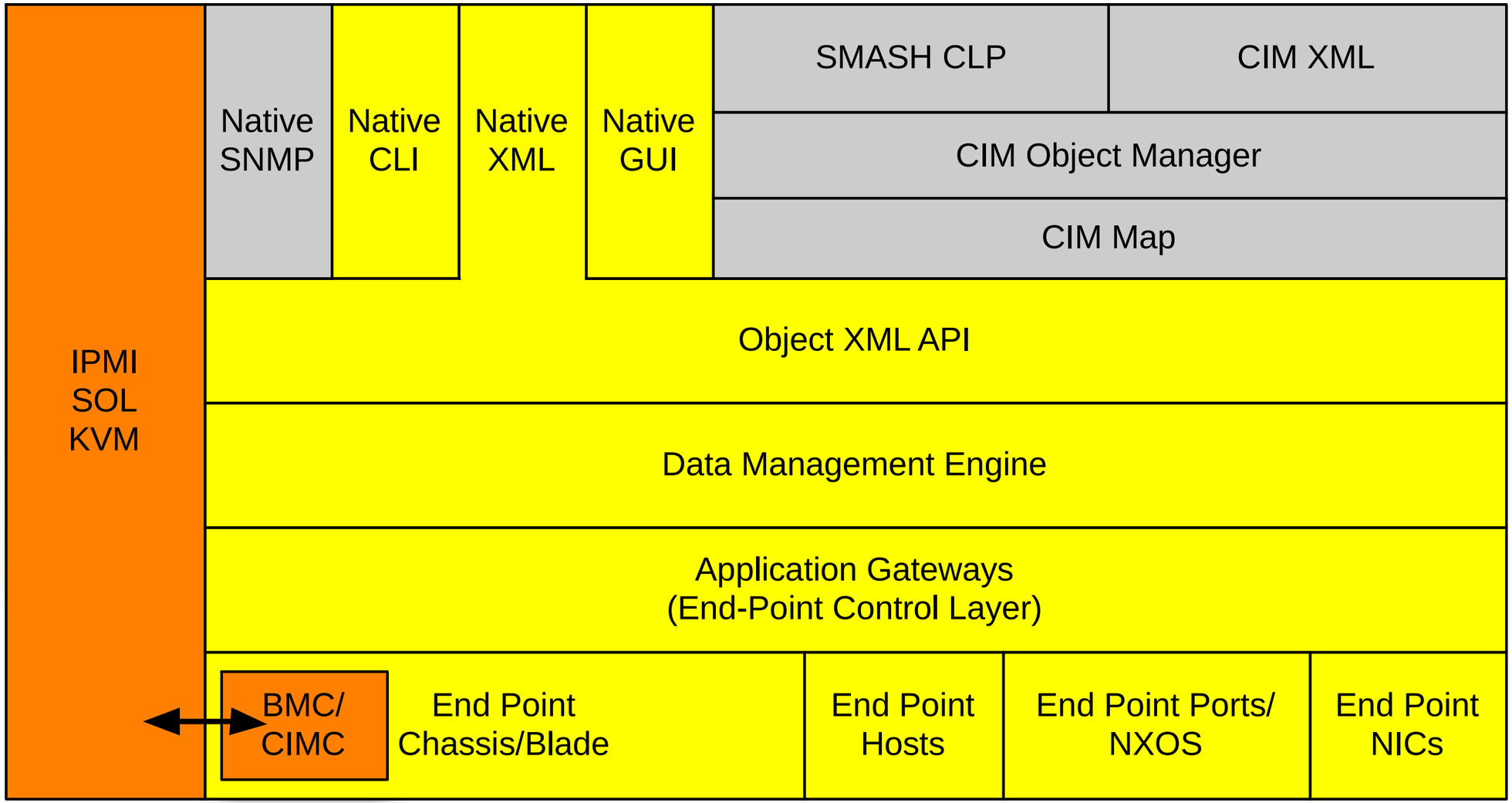
Host AG
- Server inventory
- Interacts with EP via PNuOS
- RAID, BIOS firmware, 3rd Party Option ROMs
- Local disk scrub

NIC AG
- Adapters
- MAC, WWN

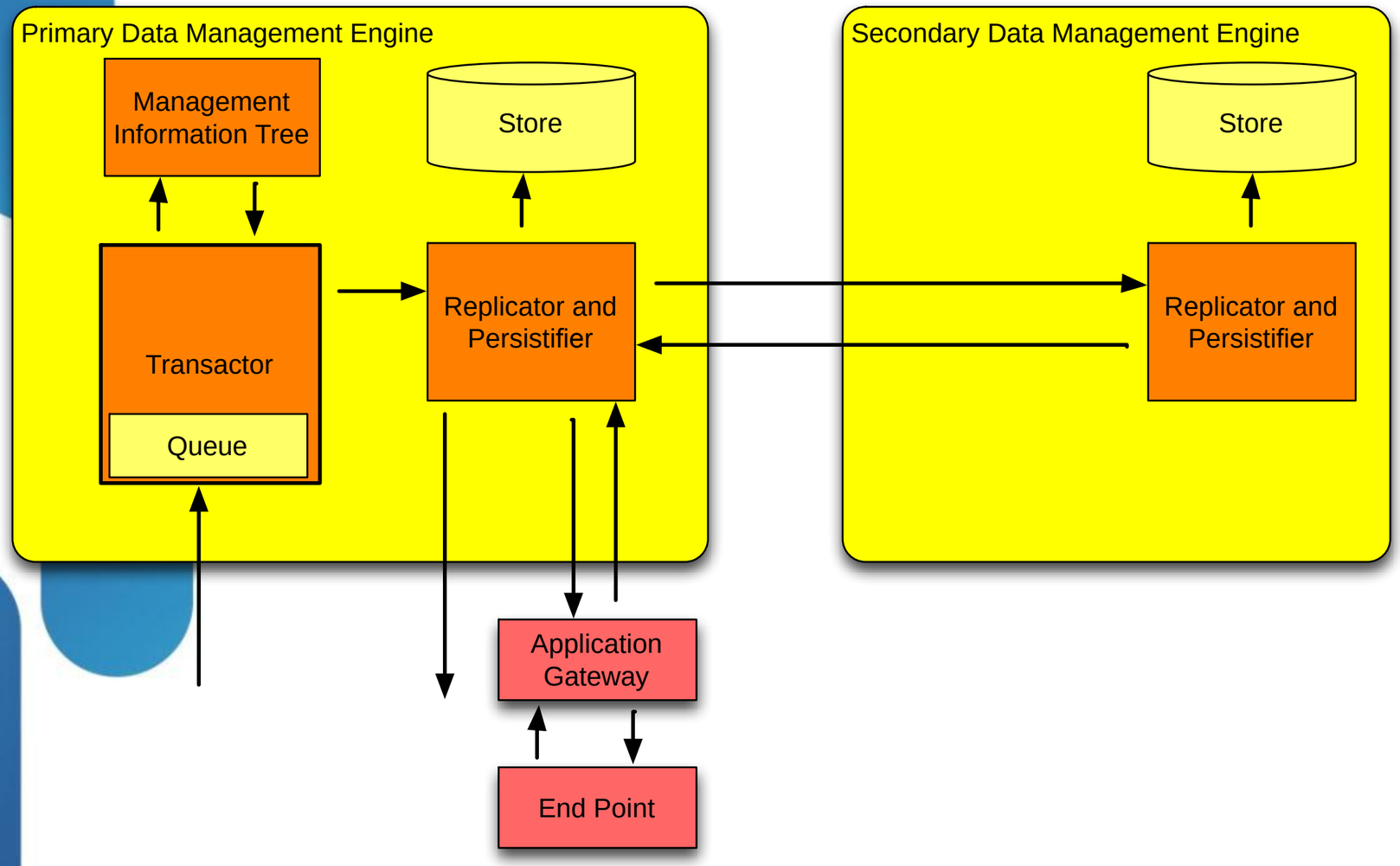
Blade/Chassis AG
- CMC
- BMC/CIMC
- Power on/off
- BIOS settings

Port/NXOS AG
- Fabric interconnects
- VLANs
- port channels, trunks
- vNICs, vHBAs

System Interface Stack



Model Driven Framework



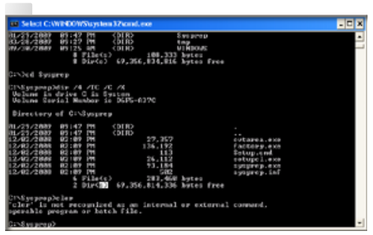
1. Config request
2. DME validates request
3. Apply config to MIT
4. Persist to DB
5. Persist to peer DB
6. Respond to client (change events sent asynchronously)
7. Request to AG
8. Change to EP
9. AG polls new state from EP and updates MIT

XML API Overview

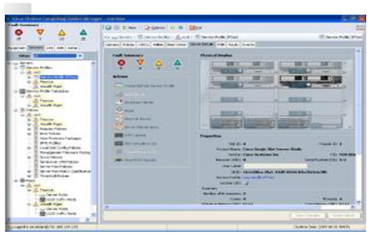
Programmatic Infrastructure

- Comprehensive XML API, standards-based interfaces
- Bi-Directional access to physical & logical internals

Direct



UCS CLI



UCS GUI

3rd Party

Self Serve portals

Management Tools

Auditing Tools

Customer

XML API

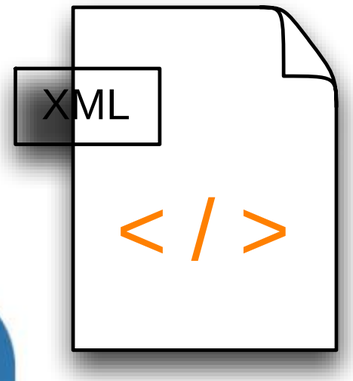
System Status
Physical Inventory
Logical Inventory

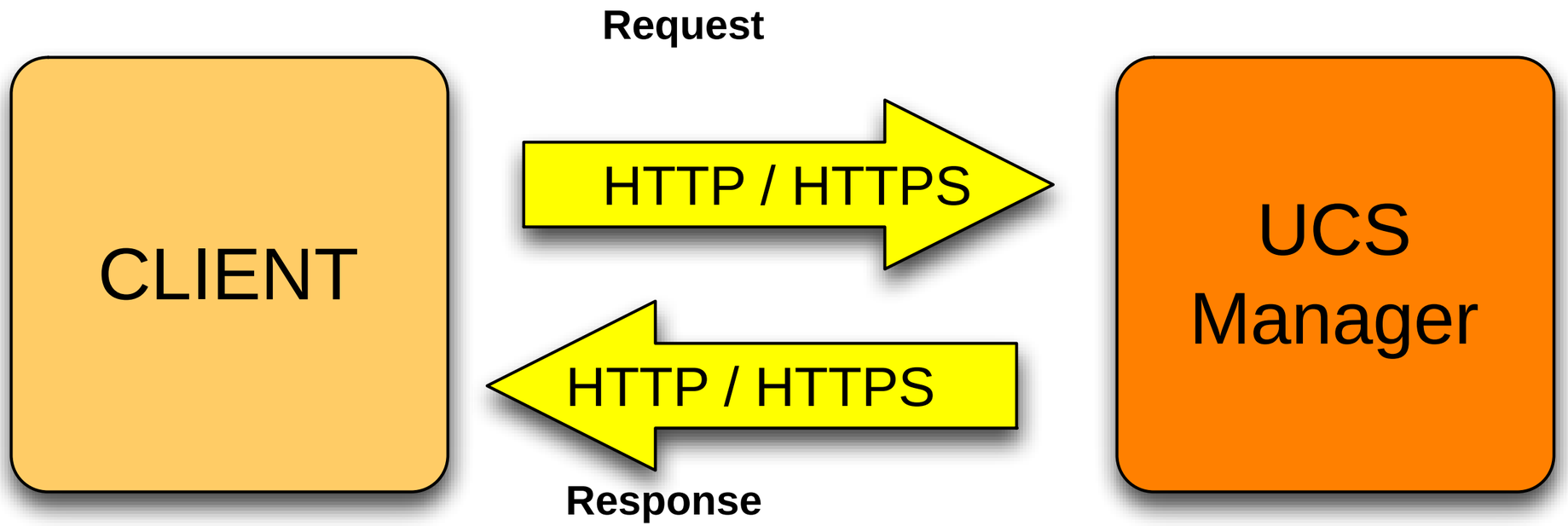
- Broad 3rd party integration support
- Faster custom integration for customer use cases
- Consistent data and views across ALL interfaces

UCS XML API Features Summary

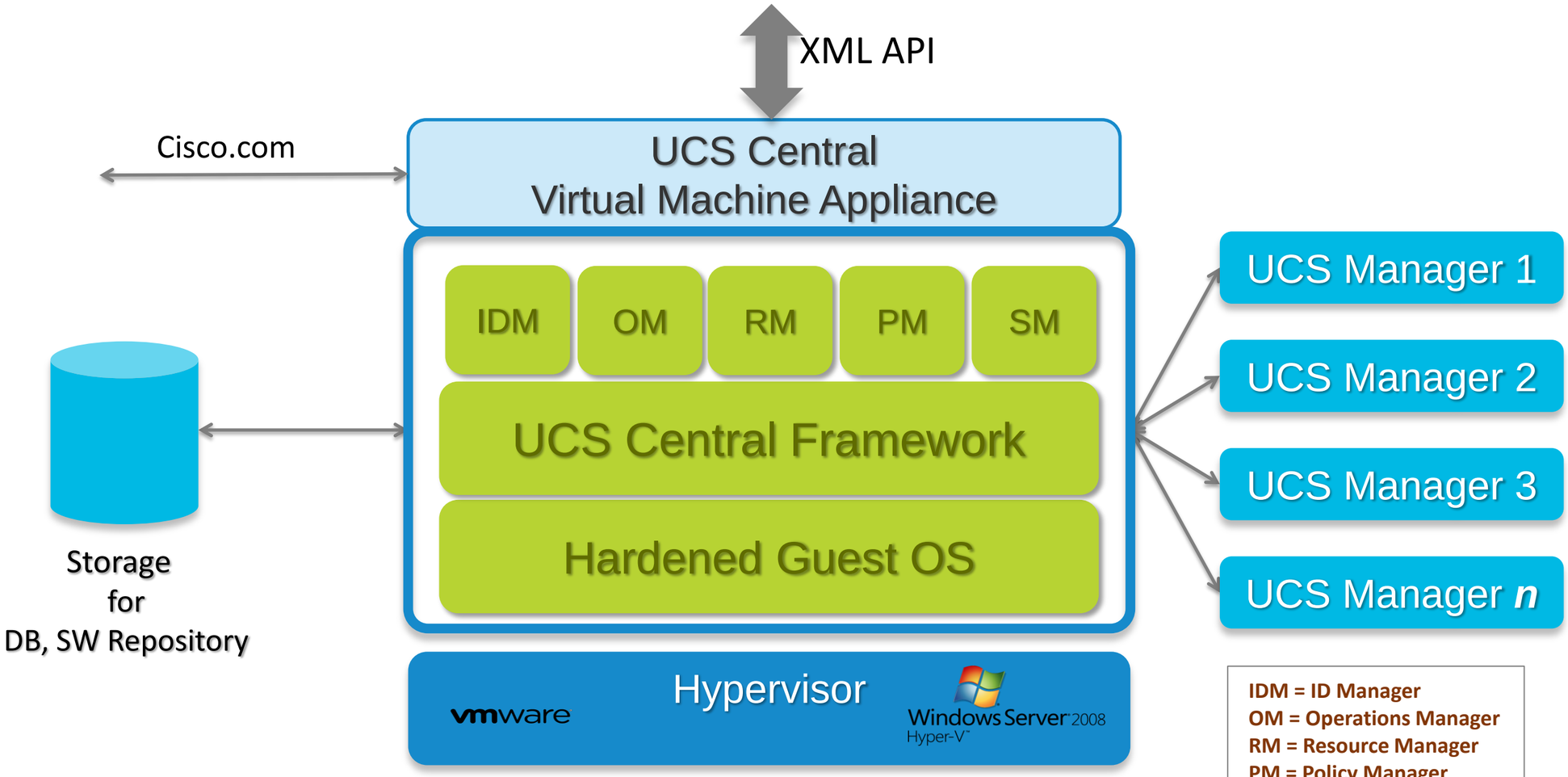
- Communicates over HTTP/HTTPS
- XML Based, Transactional
- XML Transactions are Order Agnostic
- Standard Request/Response cycle
- Role Based Authentication
- Object Model Hierarchy
- Built-in Object Browser
- Published Schema
- Java Doc Style documentation
- High Availability
- EventStream

Communications

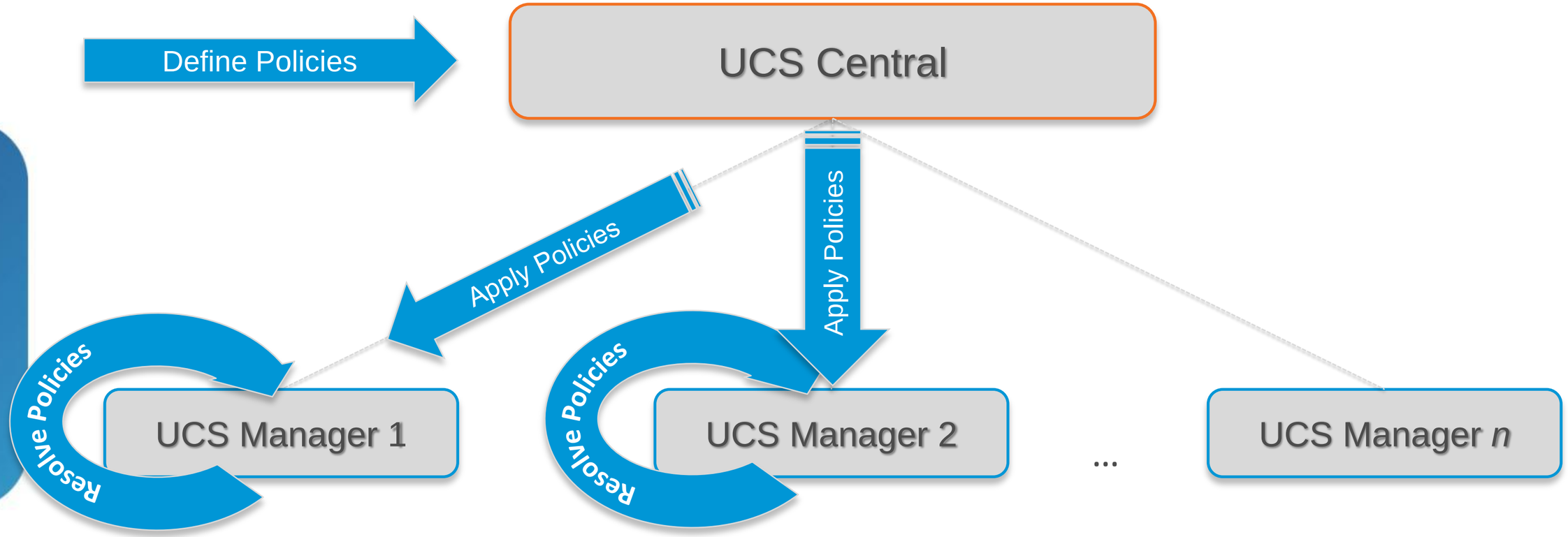




UCS Central Architecture



UCS Manager
Policy Based Management



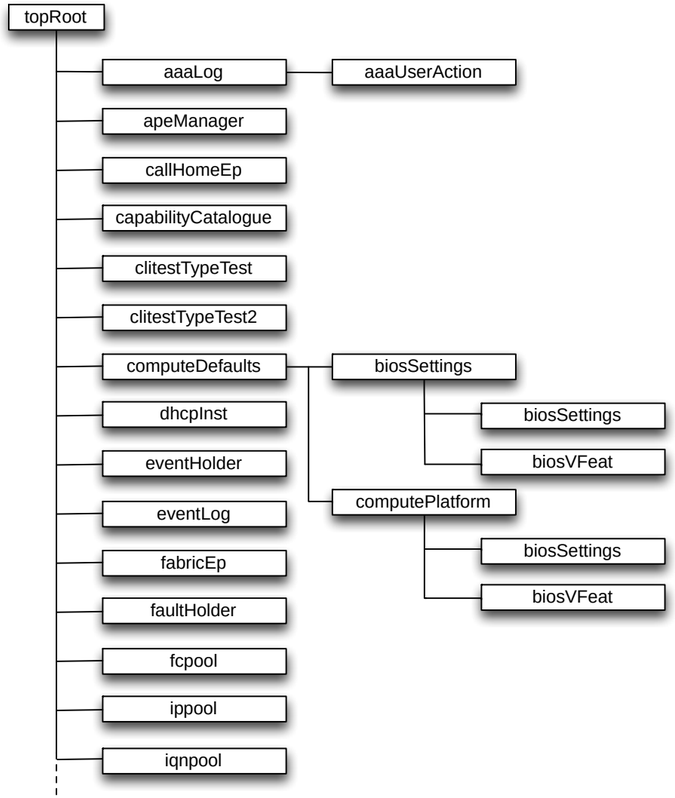
- Modeled identical to UCS Manager on how policies work
- UCS Central becomes the policy manager
- UCS Manager becomes the policy recipient and resolver

Management Information Model

Management Information Model

- A BIG tree structure of Managed Objects (MO)
- MOs represent physical and logical components
- Map to Object Classes
  - Containment (has a)
    - adaptor:Unit has a adaptor:ExtEthIf
  - Inheritance (is a)
    - Adaptor:ExtEthIf is a network:IfEp

    and has all of it's properties
  - Relationships



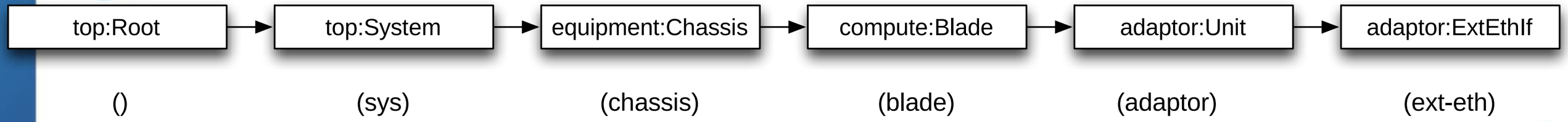
Object Naming

- Object instances addressed by
  - Distinguished Name (DN)
  - Relative Name (RN)
- Naming convention
  - prefix-[namingproperty]

  e.g. ls-testServer
  - dn = rn / rn / rn

  e.g. dn = sys/chassis-1/blade-1/adaptor-1/ext-eth-1

DN maps to object class hierarchy



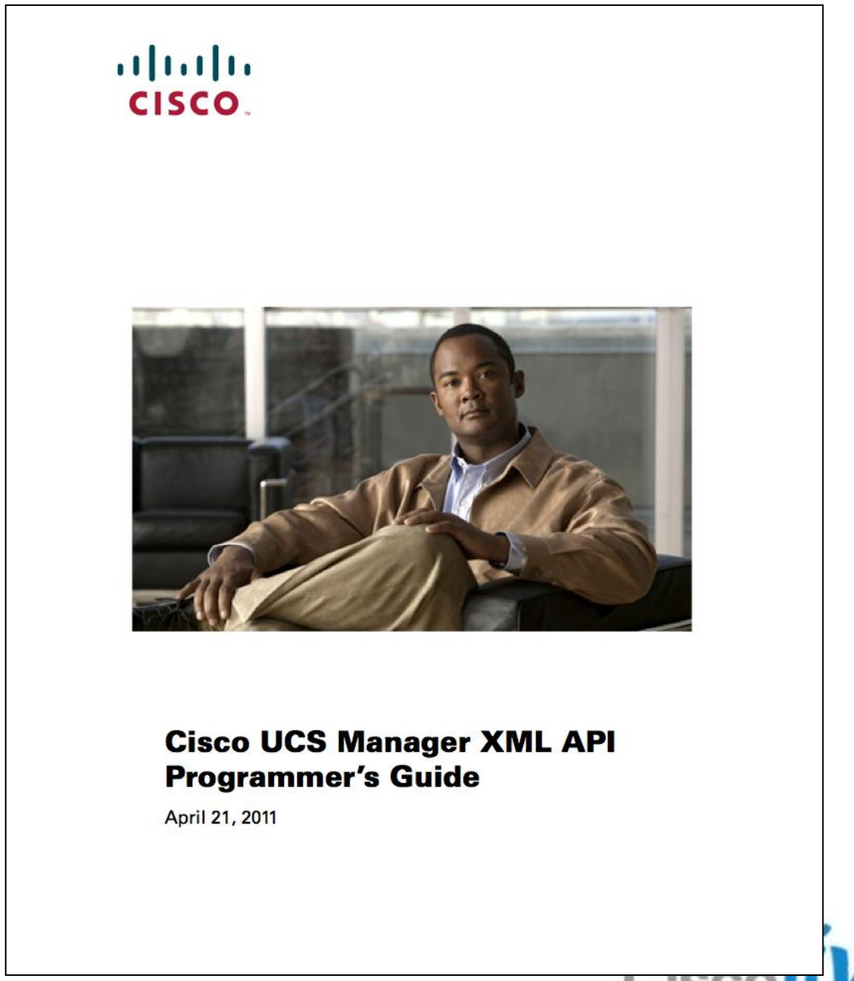
Documentation and Tools

XML API Programmer's Guide

http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/api/ucs_api_book.html

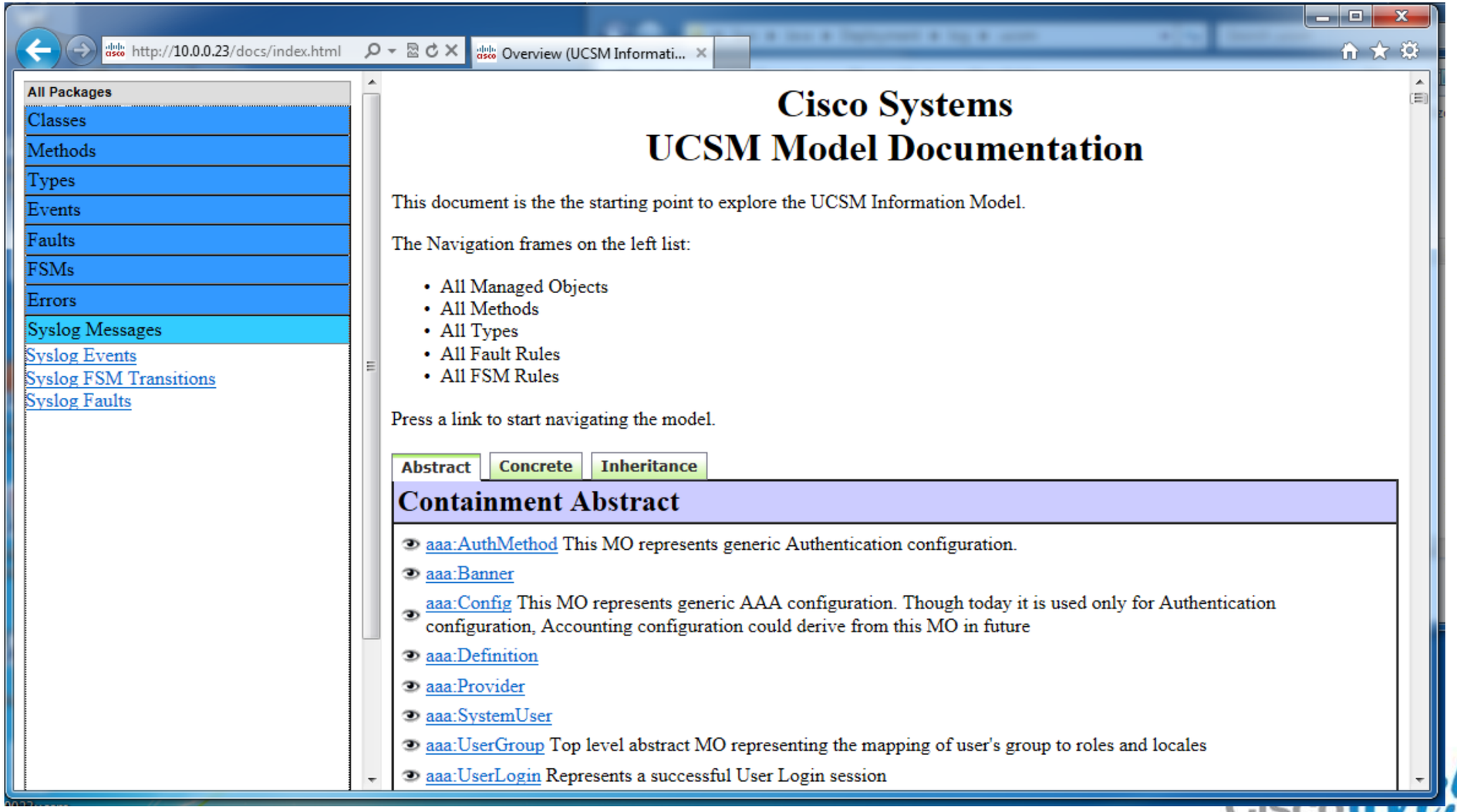
- Introduction to the API
- Description of the Methods and Filters
- Example Request and Response strings
- Summary of role based access privileges

- Browse HTML or download PDF



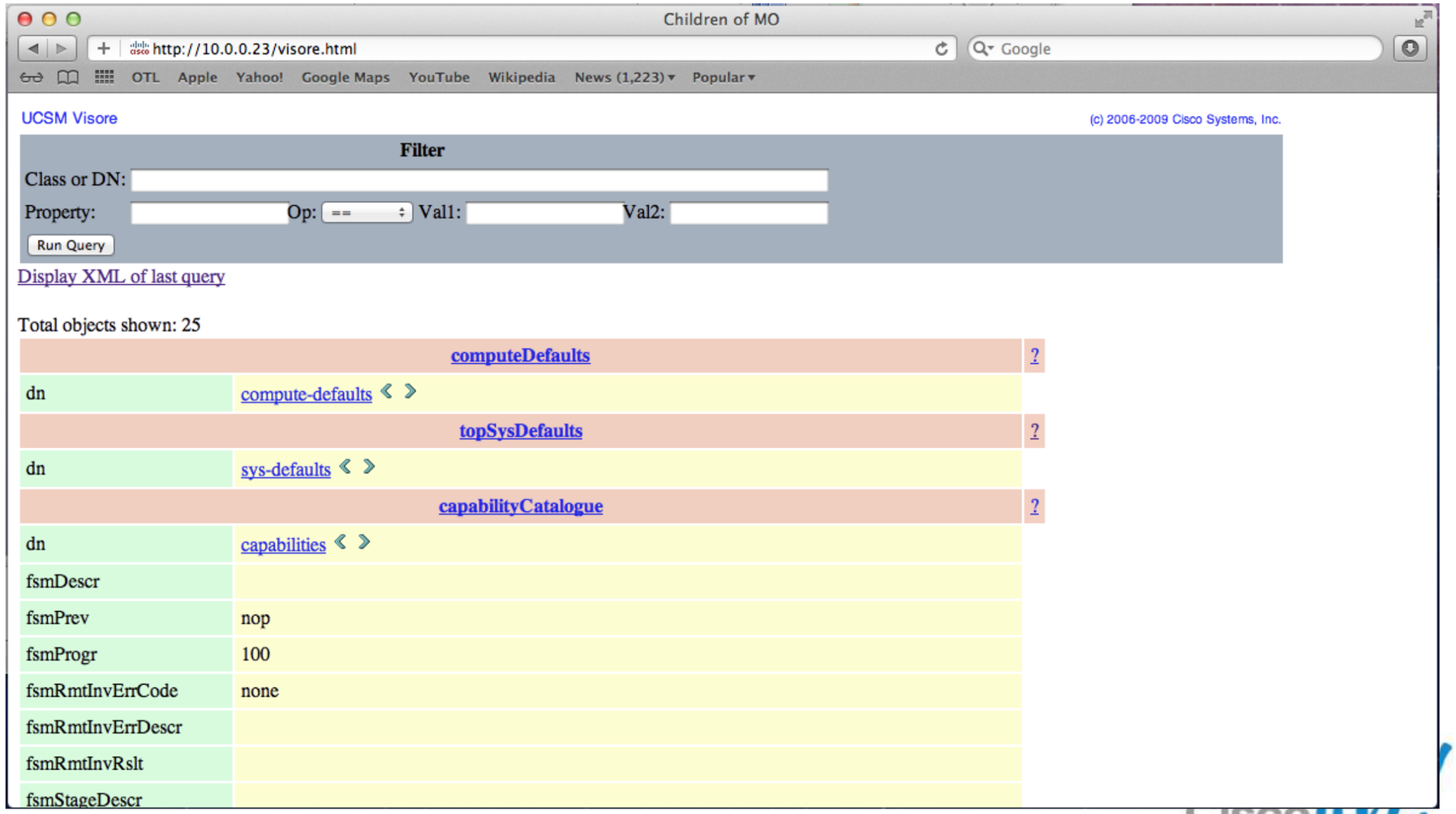
Model Documentation

http://<yourUcsm>/docs/index.html



Object Browser

http://<yourUcsm>/visore.html



Related Documentation

- UCS Fault and Error Message Reference

http://www.cisco.com/en/US/docs/unified_computing/ucs/ts/faults/reference/2.0/UCSFaultsErrorsRef_20.html

- Error Message Decoder (cisco.com login required)

http://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

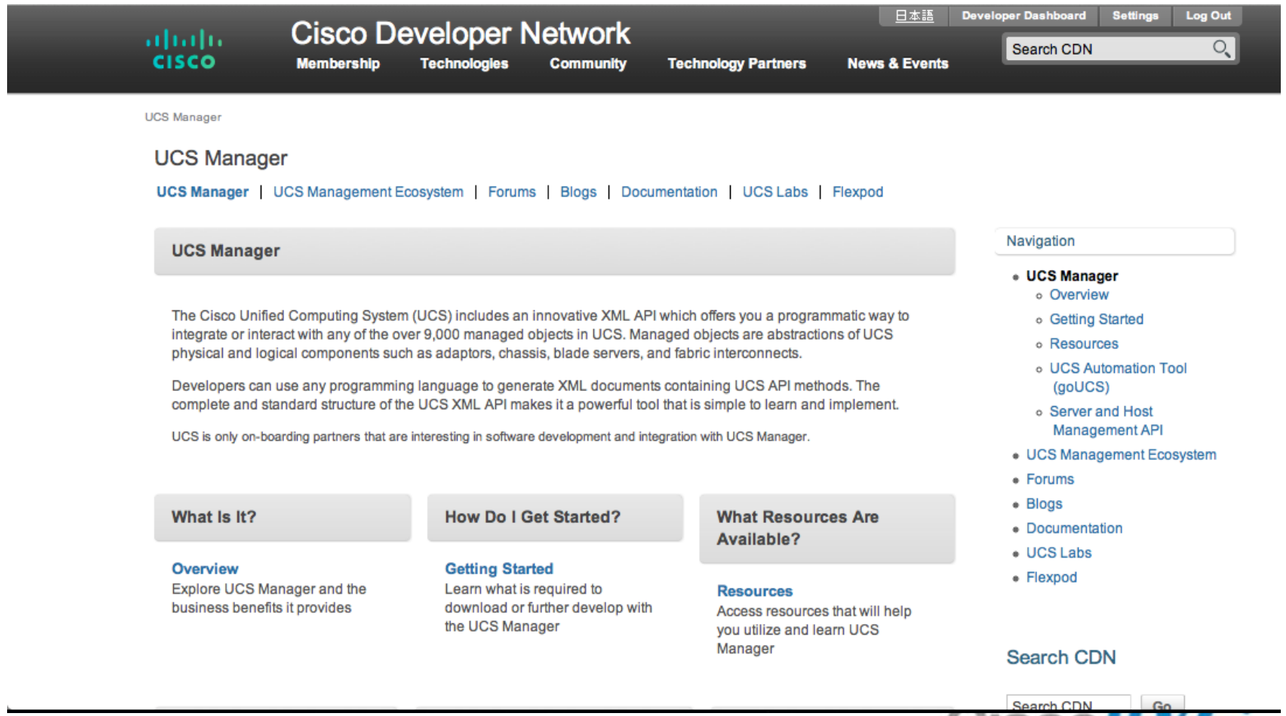
- UCS Rack-mount Server XML API Programmer's Guide

http://www.cisco.com/en/US/docs/unified_computing/ucs/c/sw/api/b_cimc_api_book.html

Cisco Developer Connection

http://developer.cisco.com/web/unifiedcomputing/home

- How to become part of the developer community
- Access to documentation
- Access to Cisco and 3rd Party tools and plugins
- Sample scripts
- Blogs and discussion forums

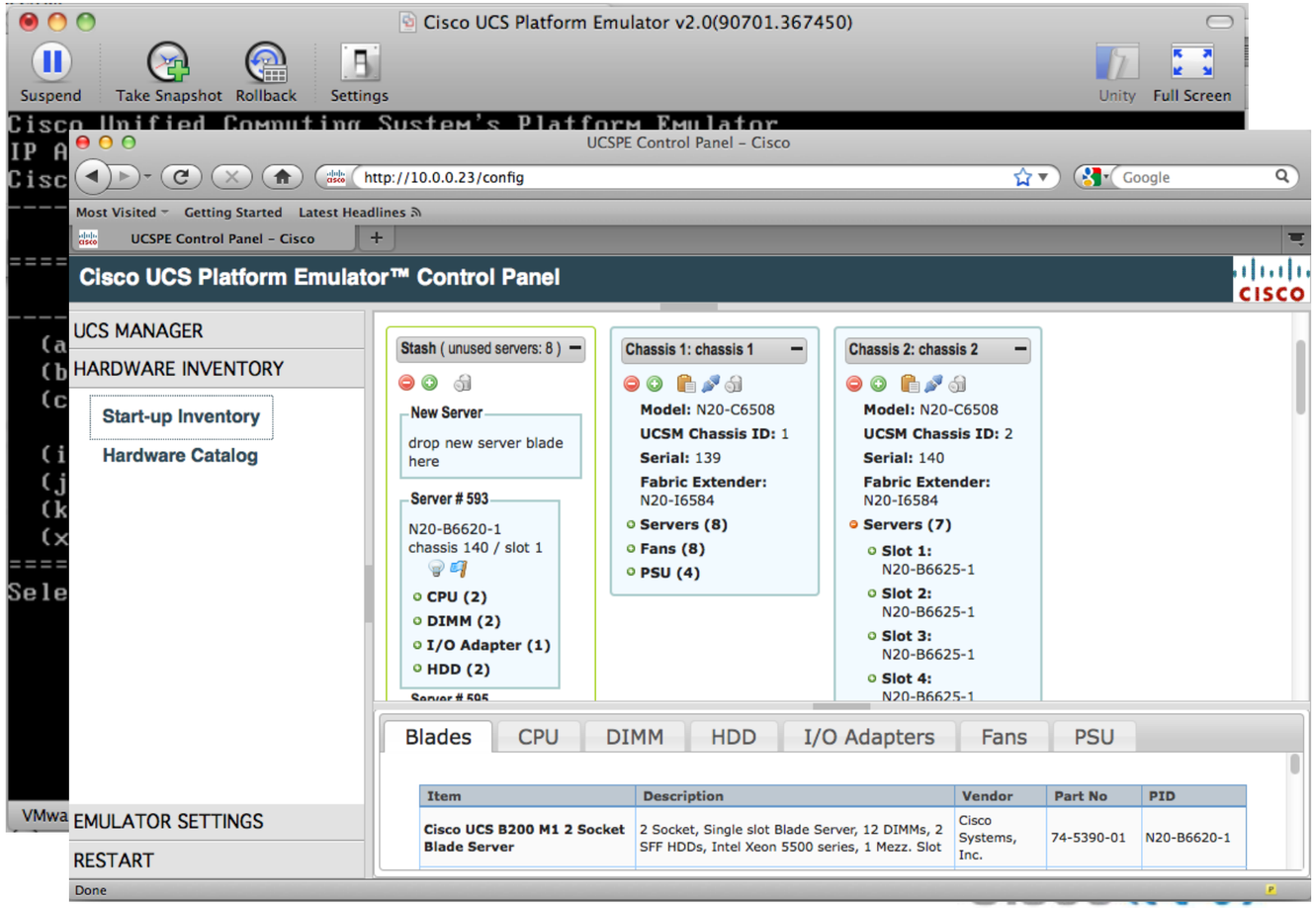


UCS Platform Emulator

- http://developer.cisco.com/web/unifiedcomputing/ucsemulatordownload
- Linux VM
- Provides full API and CLI emulation for UCSM
- Drag-Drop GUI build tool

System Requirements

- VMware
  - Player
  - Workstation
  - Fusion
  - ESX
- 1GB free RAM
- 8GB disk
- 1.8Ghz single CPU
- Mozilla compatible browser
- JRE 1.6



goUCS

- Java based tool developed by UCS Technical Marketing team
- Fully documented
- Uses a wrapper paradigm to abstract programmatic details from the developer and deliver a flexible command line tool (http only)
- XML wrapper

  goucs class realtime table:*:dn-numOfCpus-totalMemory computeBlade false

- CMD wrapper

  goucs listblades_cpu_mem

- Filterlog
- Transaction
- Connect

System Requirements

- Windows XP, 7, Server 2008
- JRE 1.6

goUCS XML Wrapper Syntax

goucs <session> <xmlWrapper> <processType> <outputType> <userArgs>

Where:

<session> = session identifier, default or all
<xmlWrapper> = the wrapper to execute
<processType> = realtime, raw or cached
<outputType> = indent, xml, quiet, xpath, csv or table + format and filters
<userArgs> = user input such as a class name or variables

Examples:

goucs all class realtime table:*:dn-numOfCpus-mem computeBlade

goucs default createvlan realtime quiet XXX 3000

goUCS Command Wrapper Syntax

goucs <session> <cmdWrapper> <userArgs>

Where:

<session> = session identifier, default or all
<cmdWrapper> = the wrapper to execute
<userArgs> = user input such as or variables

Example:

goucs addVlan XXX 3000

goUCS Filterlog Syntax

goucs filterlog [logfile] <replace> <search> <trig1>...<trigN>

goucs filterlog logtail [eof]

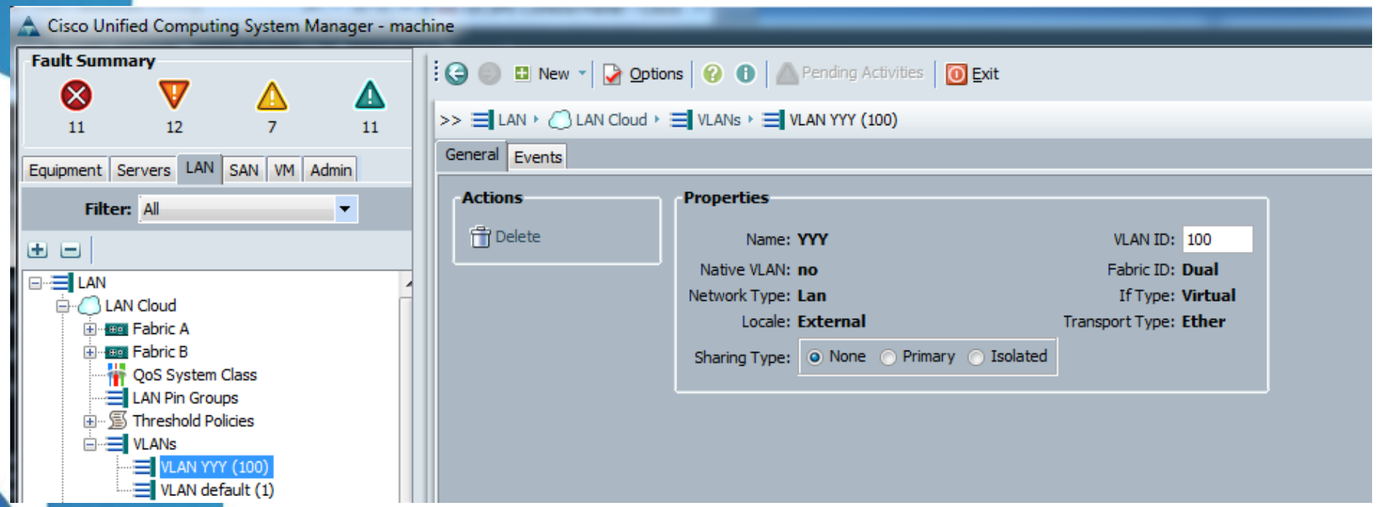
Where:

<logfile> = name of the UCSM GUI log to parse
<replace> = replace cookie and argument strings in the output
<search> = ignore keyword or a text filter string
<trigX> = text string to replace with an argument variable
or #<text>:<tag> to replace a tag
or #<text>:* to replace everything in double quotes

Example:

goucs filterlog trueco ignore YYY #id:100

goUCS Filterlog Example




```
c:\goucs>goucs filterlog trueco ignore YYY #id:100
<configConfMos
 cookie="$COOKIE$" inHierarchical="true">
<inConfigs>
<pair key="fabric/lan/net-$ARG1$">
<fabricVlan
defaultNet="no"
dn="fabric/lan/net-$ARG1$"
id="$ARG2$"
name="$ARG1$"
pubNwName=""
sharing="none"
status="created">
</fabricVlan>
</pair>
</inConfigs>
</configConfMos>

c:\goucs>
```

API Methods and Filters

API Method Types

- Authentication – login and logout
- Query – View MOs and attribute values
- Configuration – Create and modify MOs and attributes
- Event Subscription – Monitor system events

Authentication Methods

aaaLogin – Initial login
aaaKeepalive – keep the session open
aaaRefresh – Refresh current authentication cookie (600second expiry)
aaaLogout – Exit the session and deactivate the cookie

```
<aaaLogin
   inName="cliuser"
   inPassword="cliuser">
</aaaLogin>
```

Query Methods

configResolveDn – Retrieve MOs by DN
configResolveDns – Retrieve MOs by a set of DNs
configResolveClass – Retrieve MOs of a specified class
configResolveClasses – Retrieve MOs of multiple classes
configFindDnsByClassId – Retrieves DNs of a specified class
configResolveChildren – Retrieves the child MOs of a MO
configResolveParent – Retrieves the parent MO of a MO
configScope – Performs class queries on a DN

Query Samples

```
<configResolveDn
  dn="sys"
  inHierarchical="false">
</configResolveDn>
```

```
<configResolveClasses
  inHierarchical="false">
    <inIds>
      <classId value="computeBlade"/>
      <classId value="equipmentIOCard"/>
    </inIds>
</configResolveClasses>
```

Query filters

Simple filters

- True filter – boolean true
- False filter – boolean false

Property filters

- Equality filter – eq
- Not equal filter – neq
- Greater than filter – gt
- Greater than or equal filter – ge
- Less than filter - lt
- Less than or equal filter – le
- Wildcard filter – wcard. “%” or “*” (any sequence of characters), “?” or “-” (any single character).

Filter Sample

```
<configResolveClass inHierarchical="false" cookie="1329083125/3b45fea1-77fb-4f53-b67a-dfc2edb29845" classId="computeBlade">
   <inFilter>
      <and>
         <eq class="computeBlade" property="numOfCores" value="8"/>
         <gt class="computeBlade" property="availableMemory" value="8192"/>
      </and>
   </inFilter>
</configResolveClass>
```

Filter Sample 2

```
<configResolveClass inHierarchical="false" cookie="1329083125/3b45fea1-77fb-
4f53-b67a-dfc2edb29845" classId="computeBlade">
   <inFilter>
         <bw class="computeBlade"
             property="availableMemory"
             firstValue="4000"
             secondValue="66000"/>
   </inFilter>
</configResolveClass>
```

Configuration Methods

configConfMo – configures a single subtree (DN)
configConfMos – configures multiple subtrees (several DNs)
configConfMoGroup – makes the same changes to multiple subtree structures or MOs

Configuration Sample

```
<configConfMos
   cookie="1329083125/3b45fea1-77fb-4f53-b67a-dfc2edb29845"
   inHierarchical="true">
   <inConfigs>
      <pair
         key="fabric/lan/net-SIN">
         <fabricVlan
            defaultNet="no"
            dn="fabric/lan/net-SIN"
            id="104"
            name="SIN"
            pubNwName=""
            sharing="none"
            status="created">
         </fabricVlan>
      </pair>
   </inConfigs>
</configConfMos>
```

Event Subscription Method

eventSubscribe – register to receive event notifications

```
<eventSubscribe
   cookie="1329094394/eda7c20e-3cd4-4265-8332-110cfef56f17">
</eventSubscribe>
```

Working Examples

Hello World

$ **telnet 10.0.0.23 80**
Trying 10.0.0.23...
Connected to 10.0.0.23.
Escape character is '^]'.
POST /nuova HTTP/1.1
Host: 10.0.0.23
Content-Length: 60

<aaaLogin inName="cliuser" inPassword="cliuser"/>

HTTP/1.1 200 OK
Date: Thu, 09 Feb 2012 01:59:05 GMT
Server: Apache/2.2.17 (Unix) mod_ssl/2.2.17 OpenSSL/FIPS
Content-Length: 426
Content-Type: application/soap+xml

<aaaLogin cookie="" response="yes" **outCookie="1328752745/22003f61-fe00-455d-9494-22bbe857e512"** outRefreshPeriod="600" outPriv="aaa,admin,ext-lan-config,ext-lan-policy,ext-lan-qos,ext-lan-security,ext-san-config,ext-san-policy,ext-san-security,fault,operations,pod-config,pod-policy,pod-qos,pod-security,read-only" outDomains="org-root" outChannel="noencssl" outEvtChannel="noencssl" outSessionId="" outVersion=""> </aaaLogin>
Connection closed by foreign host.
$

Game Cheat

Where is the UCSM Client Log file?

%userprofile%\AppData\LocalLow\Sun\Java\Deployment\log\.ucsm

Note: AppData is a hidden folder

[------------ Sending Request to Server ------------
<aaaLogin
inName="cliuser"
inPassword="cliuser">
</aaaLogin>
--]

[----------debugBuffer----------]
<aaaLogin cookie="[hidden]" response="yes" outCookie="[hidden]" outRefreshPeriod="600" outPriv="aaa,admin,ext-lan-config,ext-lan-policy,ext-lan-qos,ext-lan-security,ext-san-config,ext-san-policy,ext-san-security,fault,operations,pod-config,pod-policy,pod-qos,pod-security,read-only" outDomains="org-root" outChannel="noencssl" outEvtChannel="noencssl" outSessionId="" outVersion=""> </aaaLogin>
--]

Example in PERL

The following script

- Authenticates with UCSM with user provided credentials
- Can send a single-quoted multiline string to UCSM
- Can subscribe to the UCSM event stream
- Returns the response from UCSM
- Does NO syntax checking of the user provided string
- Output can be directed to a file but defaults to STDOUT
- Is for demonstration purposes
- Carries no warranty and no responsibility is held by the author or Cisco for what YOU do with it
- Might be useful as a starting point when you are learning the UCSM XML API

PERL Example

```
#!/usr/bin/perl -w

# use strict;
use Socket;
use LWP::UserAgent;
use Getopt::Std;

my $ua = LWP::UserAgent->new;
my $cookie;
my %option = ();
my $FH;
my $username;
my $password;
my $targetUcs;
my $outfile;
my $request;
my $line;
```

```
getopts("o:u:p:t:r:eh", \%option);

# usage() if ($option{h});

if($option{u}){
	$username = $option{u};
print "Username: $username\n";
}else{
print "No username entered\n";
	usage();
	exit(0);
}
if($option{p}){
	$password = $option{p};
print "Password: $password\n";
}else{
	usage();
	exit(0);
}
if($option{t}){
	$targetUcs = $option{t};
print "UCS: $targetUcs\n";
}else{
	usage();
	exit(0);
}
```

```
if($option{o}){
	$outfile = $option{o};
	open(OUT,">>$outfile") || die "Unable to open output file $outfile:$!\n";
	$FH = *OUT;
}else{
		print "Using stdout\n\n";
	$FH = *STDOUT;
}
if($option{r}){
		$request = $option{r};
}
```

```
$ua->agent("grosenb/0.1 ");
# Create an authentication request
my $authRequest = HTTP::Request->new(POST => "http:\/\/$targetUcs\/nuova");
$authRequest->content_type('application/x-www-form-urlencoded');
$authRequest->content('<aaaLogin inName="'.$username.'" inPassword="'.$password.'"/>');

# Pass request to the user agent and get a response back
my $authResult = $ua->request($authRequest);

# Check the outcome of the response
if ($authResult->is_success) {
   print "Got auth result:\n", $authResult->content, "\n";
   $authResult->content =~ /.+outCookie=\"(.+)\" outRefreshPeriod.+/;
   $cookie = $1;
   defined($cookie) || die "Unable to obtain cookie!";
}else {
   die "Authentication failed!\n\n", $authResult->status_line, "\n";
}
print "Got cookie: $cookie\n";

if($option{e}){
                eventSubscribe($FH, $cookie, $targetUcs);
}
close $FH;
```

```
# Create a user request
if ($request){
                $request =~ s/(\S+?\s+?cookie=\")(\S*)(\"\s+?)/$1$cookie$3/;
print $FH "$request\n";
                my $userRequest = HTTP::Request->new(POST => "http:\/\/$targetUcs\/nuova");
                $userRequest->content($request);

                my $userResult = $ua->request($userRequest);

                if ($userResult->is_success) {
                                print "Got user result:\n", $userResult->content, "\n";
                }else {
                   die "User request failed!\n\n", $userResult->status_line, "\n";
                }
}
```

```
sub eventSubscribe
{
        my ($EFH, $authCookie, $ucs) = @_;

        my $line = "";
        openTcp(SOCKFH, $ucs, 80) || die "Error connecting to server at $targetUcs: $!\n";

                print SOCKFH 'POST /nuova HTTP/1.1'."\n";
                print SOCKFH "Host: $targetUcs"."\n";
                print SOCKFH 'Accept: text/html, *;'."\n";
                print SOCKFH 'Connection: keep-alive'."\n";
                print SOCKFH 'Content-type: application/x-www-form-urlencoded'."\n";
                print SOCKFH 'Content-length: 91'."\n\n";
                print SOCKFH '<eventSubscribe cookie="'.$authCookie.'"></eventSubscribe>'."\n";


        while($line=<SOCKFH>){
                print $EFH $line;
        }
        1;
}
```

```
sub openTcp
{
	my ($FS, $dest, $port) = @_;

	my $proto = getprotobyname('tcp');
	socket($FS, PF_INET, SOCK_STREAM, $proto);
	my $sin = sockaddr_in($port, inet_aton($dest));
	connect($FS, $sin) || return undef;

	my $old_fh = select($FS);
	$| = 1;
	select($old_fh);
	1;
}
1;
```

```
sub usage{

  print qq{Usage: $0
        -u <username>: UCSM login username
        -p <password>: UCSM login password
        -t <ipAddress>: ip address of the UCSM
        -o <outputFilename>: filename for output (defaults to stdout)
        -r '<string>': user request string enclosed in single quotes (') - may be multiline
        -e: subscribe to event stream
        -h: this help listing
        sends a request string to UCSM API - no syntax checking is done on the string!
                The [realcookie] can be any string without whitespace but MUST be in double quotes!
};
  exit(0);
}
```

UCS Powertool

UCS Powertool

- Windows Powershell library
- Download available from Cisco Developer Network
- Batteries not supplied
- 1463 Cmdlets
- 26 Aliases
- 17 Functions
- Manage multiple UCS instances simultaneously

- Download and getting started guide
  http://developer.cisco.com/web/unifiedcomputing/pshell-download

Powertool Example

```
Connect-Ucs 10.0.0.27

$("Name,Id";foreach ($vlan in 500..550) { ”VLAN${vlan},${vlan}" }) > C:\Demo\vlans.csv

$lc = (Get-UcsLanCloud)

$lc | Get-UcsVlan | select ucs,name,id

import-csv C:\Demo\vlans.csv | % { $lc | add-ucsvlan -Name $_.Name -Id $_.Id }

$lc | Get-UcsVlan | select ucs,name,id

# Delete the added VLANs

$lc | Get-UcsVlan |  ? { $_.Id -ge 500 -and $_.Id -le 550 }  | Remove-UcsVlan -Force
```

Q & A

Complete Your Online Session Evaluation

Give us your feedback and receive a Cisco Live 2013 Polo Shirt!

Complete your Overall Event Survey and 5 Session Evaluations.

- Directly from your mobile device on the Cisco Live Mobile App
- By visiting the Cisco Live Mobile Site www.ciscoliveaustralia.com/mobile
- Visit any Cisco Live Internet Station located throughout the venue

Polo Shirts can be collected in the World of Solutions on Friday 8 March 12:00pm-2:00pm

Cisco live! 365

Don’t forget to activate your Cisco Live 365 account for access to all session material, communities, and on-demand and live activities throughout the year. Log into your Cisco Live portal and click the "Enter Cisco Live 365" button.
www.ciscoliveaustralia.com/portal/login.ww

Cisco live!