



Cisco Configuration Engine 2.0 Overview



Vikram Rao
Product Manager, ARTG
vikram@cisco.com

Agenda

- Product Introduction
- Features
 - Zero Touch Deployment
 - Mass Configuration Updates and Image Upgrades
 - Web GUI
 - Northbound API: Web Services
 - Velocity Templates
- Summary

Cisco Configuration Engine (CCE)?

- A scalable, secure solution designed to automate distribution of CPE configuration and software images to **large** number of devices **efficiently and quickly**.
- Configuration Engine features:
 - Day 0 Deployment: Initial CPE rollout via Zero Touch device deployment
 - Day 2 Management: Efficient mass configuration changes and IOS image upgrades
- Customer Value Proposition:
 - Significantly reduce CPE initial rollout cost (No more truck rolls!)
 - OPEX Reduction - Reduce ongoing OPEX by providing efficient mass configuration and image upgrades
 - Reduce operations failures caused by manual operator errors
 - Accelerate CPE deployment (and consequently customer revenue ramp-up)

Source of Network Failures

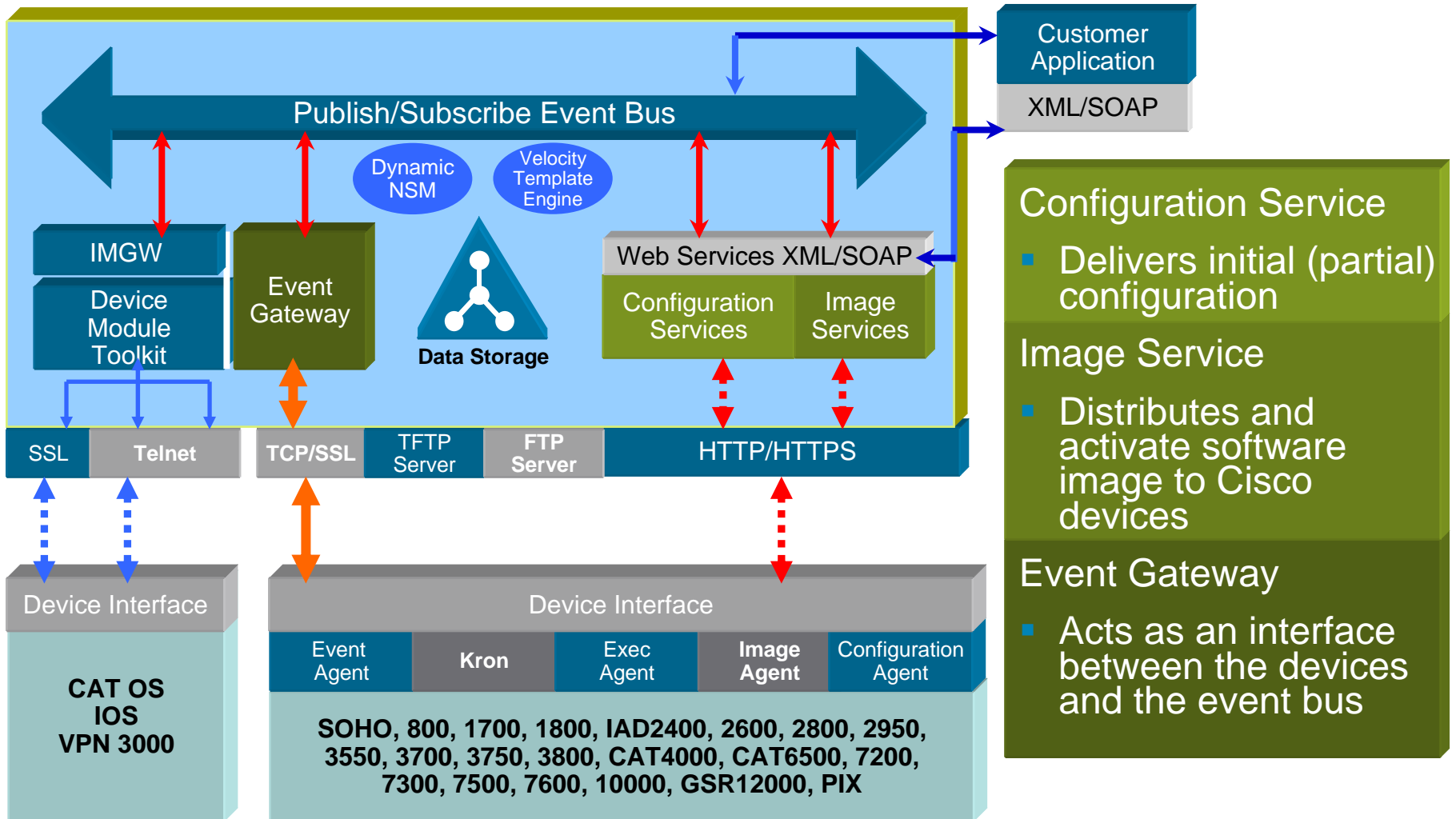
Network Operations Failures by Cause Source: Sage Research, Inc.

Network Operations Failure Types	Percent of respondents that ranked type as most frequent source of network operations failure	
Configuration errors	39%	75%
Upgrade errors	27%	
Data entry errors	10%	
Maintenance errors	10%	
Errors in monitoring the network	7%	
Version control errors	7%	

Product Feature-by-Feature Benefits

Features	Benefits
Cisco Configuration Engine can support CPE devices using SSL transport	<ul style="list-style-type: none"> •This scalable solution enables large-scale secure deployment and management of Cisco CPE over SSL and allows users to reduce deployment costs and service turn up time.
Zero-touch service deployment	<ul style="list-style-type: none"> •Time to roll out new service is significantly reduced through eliminating staging and manual processes
Configuration Services	<ul style="list-style-type: none"> •Configuration update to one or group of devices •Email/Epage notification of outcome
Image Services	<ul style="list-style-type: none"> •Policy based validation of devices resources •Support for devices behind firewall and dynamic IP address
Web based Graphical User Interface	<ul style="list-style-type: none"> •Feature rich Web GUI enables customer use the product out of the box
Velocity Template Engine	<ul style="list-style-type: none"> • Customizable to meet customers business and operation requirement. • Work flow control and support for scripts
Web Services	<ul style="list-style-type: none"> •XML/SOAP WSDL available for all feature supported from the Web GUI

CNS-CE Architecture



Zero-Touch Initial Deployment



Pre-deployment

1. Add device in Configuration Engine before device bootup
 - CNS ID – device unique identifier
 - Configuration template
2. Load bootstrap configuration in device
 - Use Configuration Express or Autoinstall or eToken or SDM

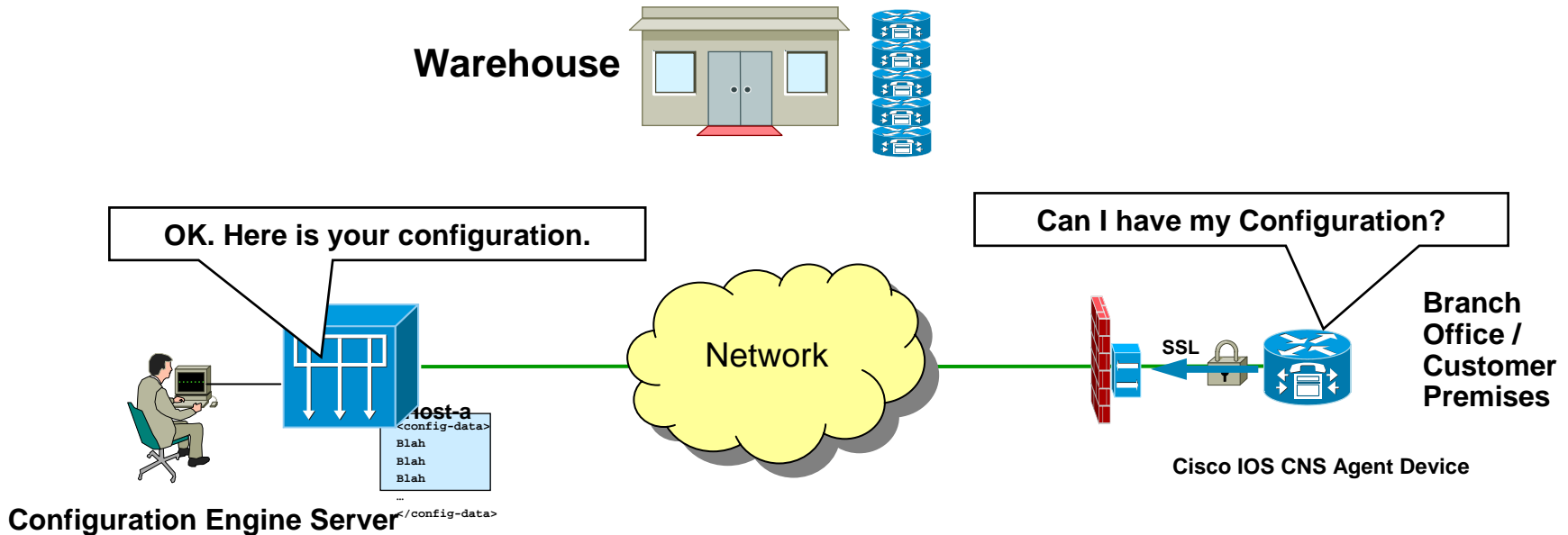
Notes:

- Each device (CNS ID) is associated with a template
- One of the following can be used as an CNS ID:
Hostname, IP address, MAC address, Hardware Serial Number, UDI
(Unique Device Identifier) or any string

Sample Bootstrap – FastEthernet

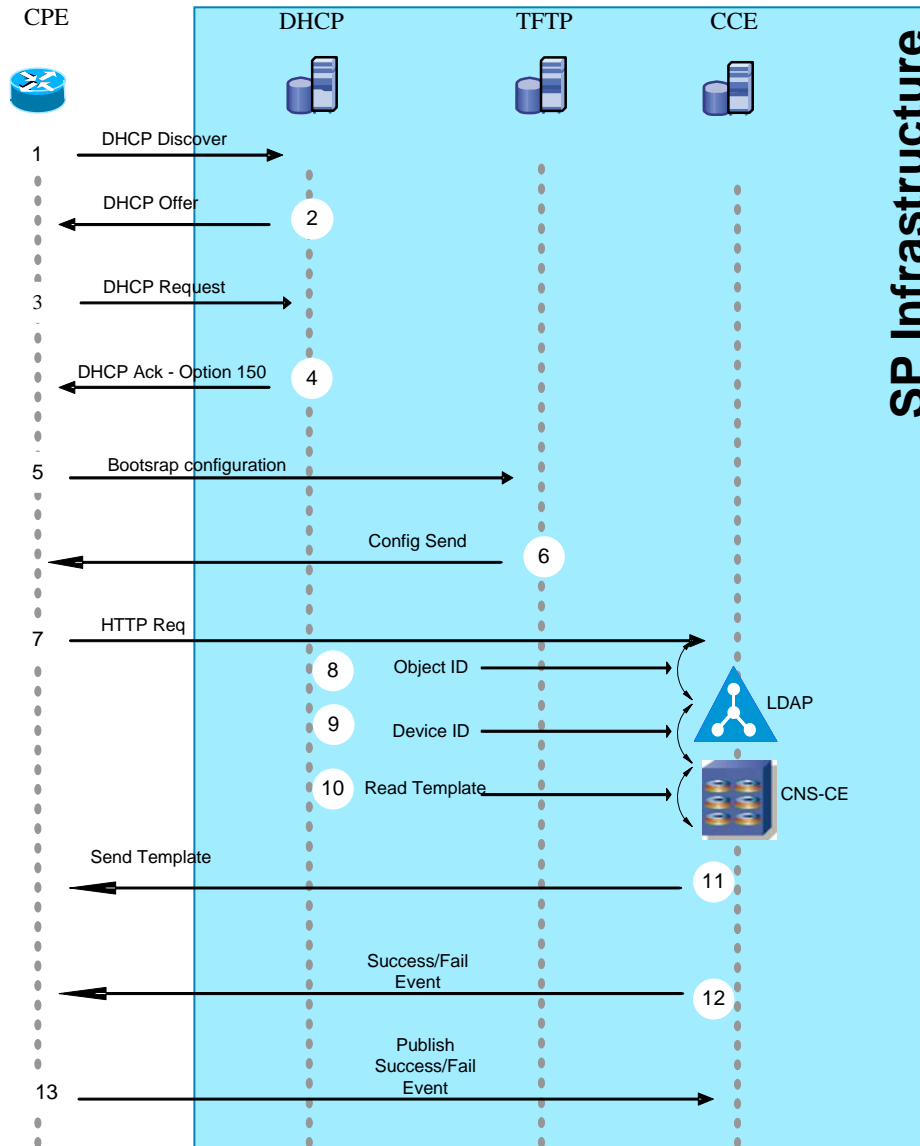
```
!  
Version 12.2  
!  
cns config connect-intf FastEthernet ping-interval 30 retries 3  
  config-cli description Customer #BA26718 #4659ZK7  
  config-cli ip address dhcp  
  config-cli no shutdown  
  config-cli ip route 0.0.0.0 0.0.0.0 &  
!  
cns id hardware-serial  
cns config initial MyConfigEngine.com encrypt event no-persist  
!
```

Zero Touch Deployment Illustration



1. Device (CPE-A) is selected from warehouse and loaded with bootstrap
2. CNS ID and template for CPE-A is entered in the Configuration Engine
3. CPE-A is shipped to the customer premises/branch office
4. CPE-A powers up and calls home to the Configuration Engine
5. Upon authentication, Configuration Engine sends configuration to CPE-A
6. CPE-A applies configuration and becomes operational

Zero Touch Flow with Autoinstall to load bootstrap

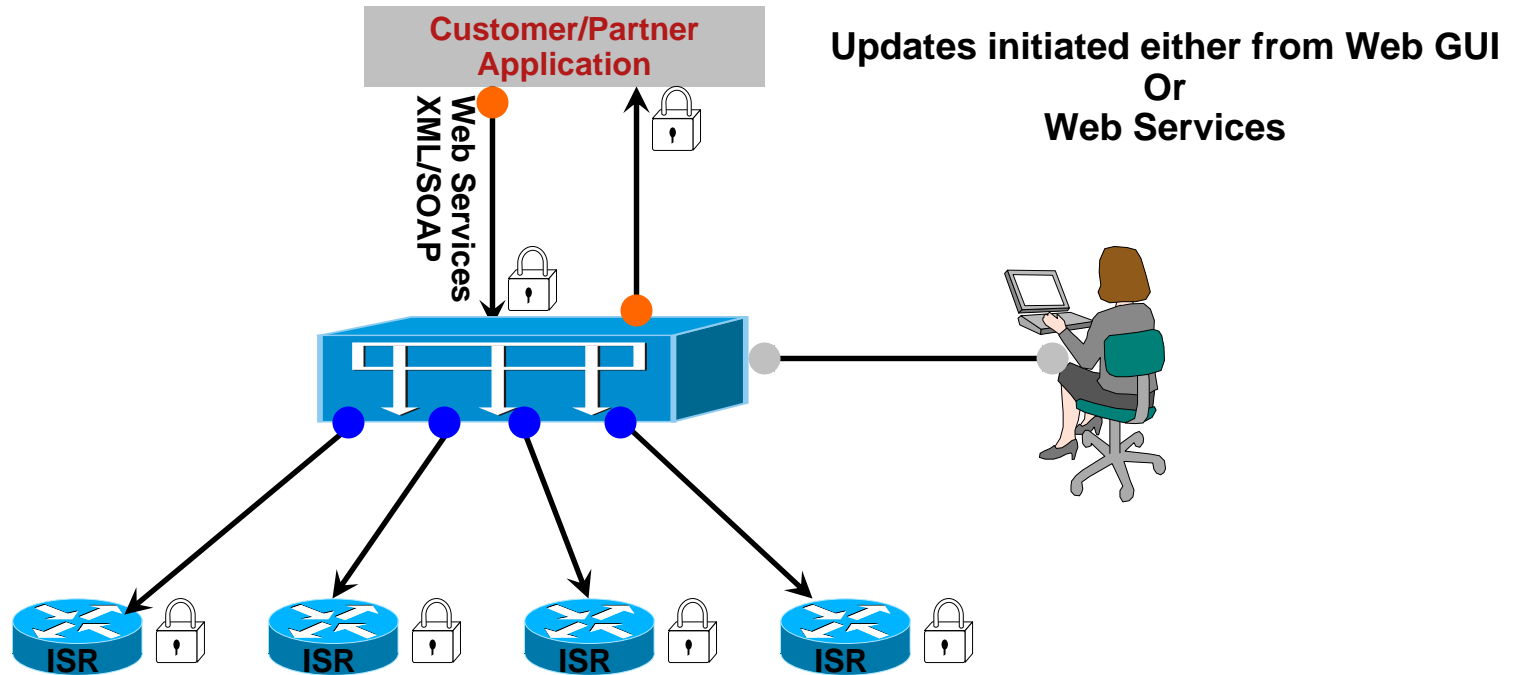


1. CPE sends DHCP Discover
2. DHCP Server replies with Offer
3. CPE sends DHCP Request
4. DHCP Server replies with option 150
5. CPE requests **bootstrap** file via TFTP
6. TFTP server sends CPE **bootstrap** file
7. CPE sends HTTP request to CNS-CE
8. CNS-CE verifies object ID
9. CNS-CE verifies Device ID
10. CNS-CE reads template from File System
11. CNS-CE sends template
12. Successful event
13. Publish success event

Post Deployment Management



On Going Changes And Network Updates



Either using Web GUI or Web Services, Cisco Configuration Engine supports

- Secure configuration updates to 1000's of devices in minutes
- Secure distribution of service configuration (Voice, VPN, Security)
- Policy based secure software image upgrades to all Cisco devices
- Secure distribution of Signature files (SDF)

✓ Increased ROI
✓ Decreased OPEX & TOC

Image Services

Image Distribution + Activation: Any File - Anywhere

- **Image Distribution**

Cisco IOS Images, Cisco CatIOS Images, Intrusion Protection Signature (IPS) Files, Security Device Manager (SDM) Files, IP Phone Images, Music On Hold (MOH) files, Interactive Voice Response (TCL IVR) Files and more

- **Image Activation**

Cisco IOS and CatIOS Images can be activated and the device reloaded and verified.

Configuration commands can be applied immediately prior to activating an image.

Hierarchical Group for any Taxonomy

- Devices can be a member of any number of groups
- Groups within Groups
- Group Updates of Parent inherit all Child Groups
- Device data can be queried to create Views
- Groupings can be based on any number of customer criteria such as Location, Services, Model No., Features, Interfaces, Time-zones, etc.
- Entire branches can be moved from one location to another.

Grouping Illustration

Groups:

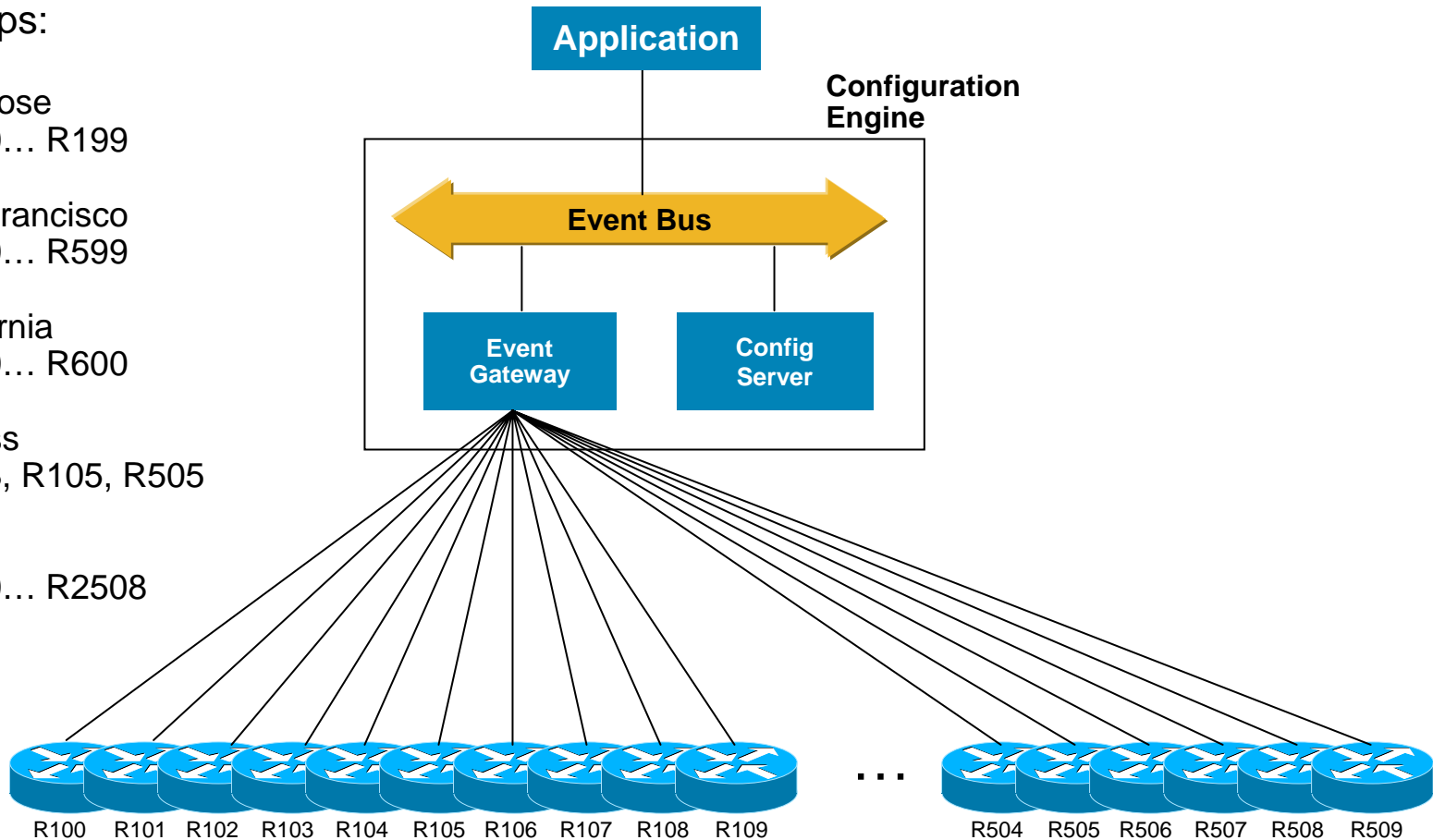
San Jose
-R100... R199

San Francisco
-R500... R599

California
-R100... R600

Access
-R103, R105, R505

All
-R100... R2508



Web GUI



Hierarchical View

The screenshot displays the Configuration Engine web interface. At the top, the title "Configuration Engine" is shown with version "1.5(0.3)" and the Cisco Systems logo. A navigation bar includes "Home", "Devices", "Users", "Jobs", "Tools", "Image Service", and "Service". The user is logged in as "cisco".

The main content area is titled "View Device" and includes a link for "Advanced Search>>". On the left, a dark sidebar lists various actions: View Device, Add Device, Discover Device, Edit Device, Resync Device, Clone Device, Delete Device, Update Device, Subdevices, Query Device Inventory, Delete Files on Devices, Execute Query, and a back button.

The central "Groups" section shows a tree structure under "config":

- config
 - CE-Team
 - TargetStores
 - California
 - Minnesota
 - Wisconsin
 - default

Device Search/Dynamic Creation Of Groups

Directory Administration Tool CISCO SYSTEMS

Home | **Devices** | Groups | Applications | Setup | Bulk Data | Logout

View Groups
Create Group
Edit Groups
Clone Group
Move Group
Delete Groups
Create Group using search

Create Group Using Search

Step 1: Search for Devices:
[Sample Filter String: ((cn=D*)&(IOSEventID=D*))]

Attribute: community Operator: = Value: Test1 Add to Query String

Query: community=Test1

Reset Query Cancel

Directory Administration Tool CISCO SYSTEMS

Home | **Devices** | Groups | Applications | Setup | Bulk Data | Logout

View Groups
Create Group
Edit Groups
Clone Group
Move Group
Delete Groups
Create Group using search

Create Group Using Search

Step 2: Select Devices to be added to the Group

Select All

Sao1 Rio1 Recife1

Back Next Cancel

Device Inventory

Configuration Engine

NSM mode: provider

Home
Devices
Users
Tools
Jobs
Image Service
UserID: admin [Logo](#)

- [View Device ▾](#)
- [Add Device](#)
- [Discover Device](#)
- [Edit Device ▾](#)
- [Resync Device](#)
- [Clone Device](#)
- [Delete Device](#)
- [Update Device ▾](#)
- [Subdevices ▾](#)
- [Query Device Inventory](#)
- [Delete Files on Devices](#)
- [<< Up](#)

ImageID:c7200-ha2
SUCCESS

Running Image Information

Description (Version String)	Cisco IOS Software, 7200 Software (C7200-I-M), Experimental Version 12.3(20040825:142830) [jbalestr-geotpi2 109] Copyright (c) 1986-2004 by Cisco Systems, Inc. Compiled Wed 01-Sep-04 14:12 by jbalestr		
Image File	disk0:c7200-i-mz.james.08-31-04	Image MD5	bc4f9da206bb7c268c641820504715b9
Config Variable		Config Reg	0x2102
		Config Reg Next Boot	
Boot Variable	disk0:c7200-i-mz.image6,1;	Bootldr Variable	
		Return To ROM Reason	reload
Return To ROM Time		Started At	

Hardware Information

Vendor	Cisco	Platform Name	7204	Hardware Revision	A
Processor Type	NPE225	Main Mem Size	117440512	IO Mem Size	16777216
Hardware Serial #	16068814	MidPlane Version	4 slot midplane, Version 1.0		
Processor Rev	R527x CPU at 262MHz, Implementation 40, Rev 10.0, 2048KB L2 Cache				
Hardware Rework					

File System List

	[FileSys name=[nvram], type=[nvram], size=[129016], freespace=[123370], readable=[true],
	file under directory], name=[startup-config],


Presentation_ID

© 2006 Cisco Systems, Inc. All rights reserved. Cisco Confidential

20

Configuration Services

Configuration Engine 1.5(0.3)



HomeDevicesUsersJobsToolsImage ServiceUserID: **cisco** [Logout](#)

[Update Config](#)
[Update Image](#)
[<< Up](#)


Update Config

Please complete the steps below to perform an Config Update:

Step 1:	<input checked="" type="radio"/> Update device with pre-configured template and parameters <input type="radio"/> Send Config: <input style="width: 100%; height: 40px;" type="text"/> <input type="radio"/> Select static configuration file: <input type="text" value="CE-ACLS-111.vm"/>
Step 2:	Config Action <input checked="" type="radio"/> Apply to running config <input type="radio"/> Apply and save to NVRAM <input type="radio"/> Overwrite NVRAM
Step 3:	<input type="checkbox"/> Syntax Check
Step 4:	<input checked="" type="radio"/> Immediate <input type="radio"/> At a future time: <input type="text" value="00"/> : <input type="text" value="15"/> (hh:mm) on <input type="text" value="January"/> <input type="text" value="1"/> <input type="text" value="2005"/>
Step 5:	Device Batch Size: <input type="text" value="20"/>
Step 6:	Text Description for Job: <input style="width: 100%;" type="text"/>

Image Services

Configuration Engine 1.5(0.3)



Home | Devices | Users | Jobs | Tools | **Image Service** | UserID: cisco | Log

Update Config
Update Image
<< Up

Step 1: Option 1: Distribute Image
Option 2: Activate Image

Step 2: Immediate
 At a future time: : (hh:mm) on

Step 3: Device Batch Size:

Step 4: Setup rule(s) to delete files:

Available Rules:		Selected Rules:
End of list	<input type="button" value=">>"/>	End of list
	<input type="button" value="<<"/>	

Step 5: Always perform delete file operation.
 Perform delete file operation if free space is needed.

Step 6: Text Description for Job:

Step 7: Apply activation template to nvram.
 Overwrite startup-config with activation template.

Please check here if you want to perform an [Evaluation](#) and not an actual [Image Update](#).

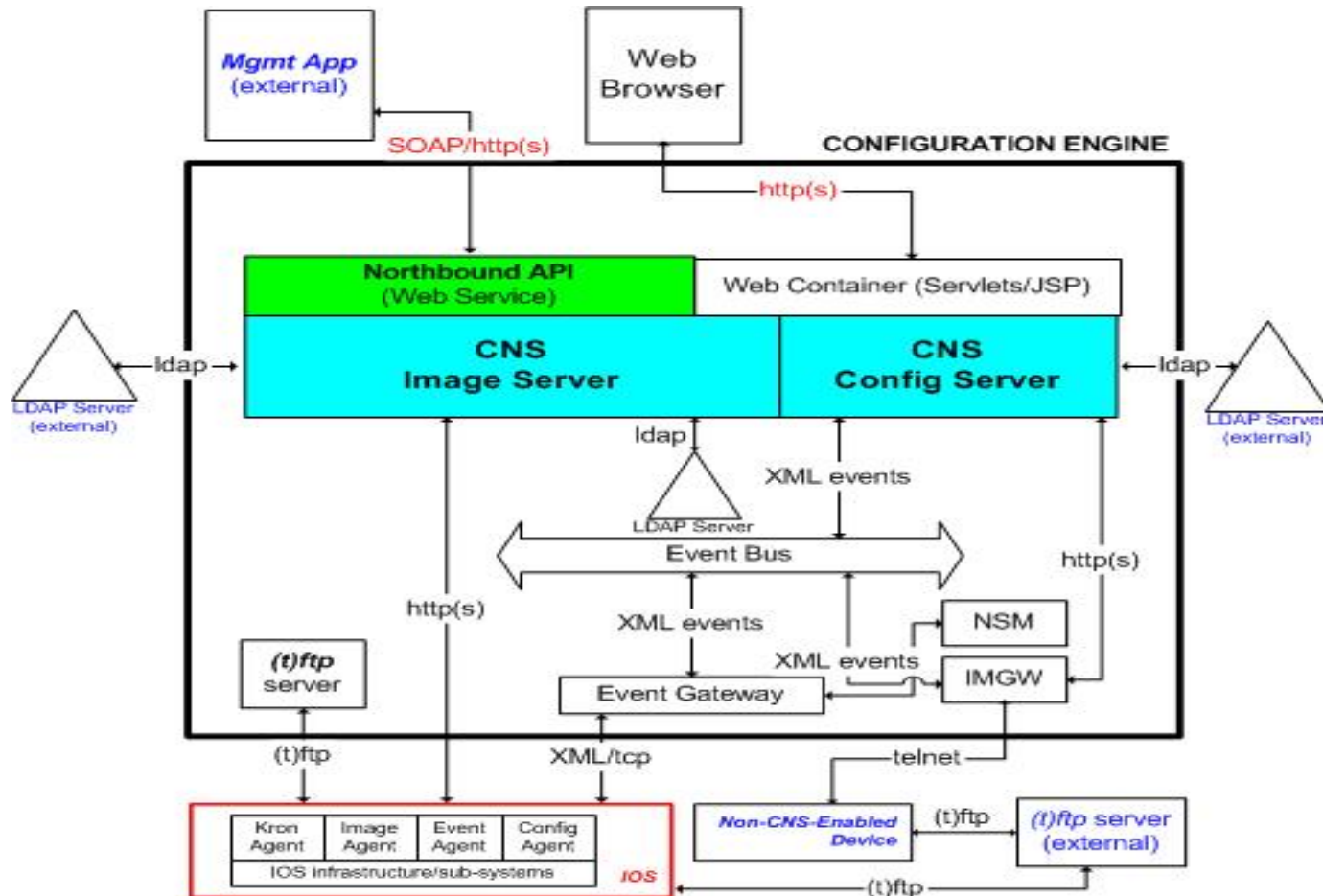
Web Services



Web Services Overview

- Provide standards-based programmatic access to Cisco Configuration Engine
- Combination of WSDL and Client Library approach
- Packaged as a separate Configuration Engine SDK sister product
- Web services available:
 - ConfigService:** Send/acquire configurations to/from devices
 - ImageService:** Distribute/activate images, obtain hardware inventory and file systems, delete files
 - ExecService:** Execute show commands or reload devices
 - AdminService:** Create and manage system objects used by CE to manage devices (e.g., devices, users, groups, templates)
 - NSMService:** Manage namespace, subjects in namespace and subject mappings in namespace

Web Service Programmatic API - Architecture



Web Service Programmatic API - Example

The screenshot displays the Cisco CE Web Service API documentation in Internet Explorer. The browser window title is "Cisco CE Web Servi". The address bar shows a URL starting with "com.cisco.netmgmt.ce.w". The page content is organized into sections: "All Classes" (listing classes like AcquiredConfig, AdminService, etc.), "Packages" (listing packages like com.cisco.netmgmt.ce.w), and "All Classes" (listing classes like AcquiredConfig, AdminService, etc.). The main content area lists the following methods:

- `getVersion ()`: Get Server's version.
- `listJob (Token token)`: Lists all the config job status existing on the server.
- `listJobByDevice (java.lang.String deviceName, java.lang.String status, Token token)`: Lists all the config job status existing on the server by the given device name and job status.
- `listJobWithStatus (java.lang.String status, Token token)`: Lists all the config job status existing on the server by the given status.
- `registerNotification (java.lang.String jobId, NotificationProperties prop, boolean reregister)`: Register the job with the notification service on the given subject names.
- `restartJob (java.lang.String[] jobIds, Token token)`: Restarts config jobs.
- `stopJob (java.lang.String[] jobIds, Token token)`: Stops config jobs.
- `unregisterNotification (java.lang.String jobId)`: Unregisters the job with the notification service.
- `updateConfig (Config[] configs, ConfigJobProperty jobProperty, Token token)`: Submits a config job.
- `updateConfigWithQuery (java.lang.String queryName, ConfigJobProperty jobProperty, Token token)`: Submits a config job with pre-defined devices, templates and parameters.
- `updateConfigWithTempl (ConfigSource[] templates, ConfigJobProperty jobProperty, Token token)`: Submits a config job.
- `updateDeviceConfig (java.lang.String[] deviceNames, ConfigJobProperty jobProperty, Token token)`: Submits a config job with pre-defined devices, templates and parameters.

The browser's status bar at the bottom shows "Discussions not available for this document" and "My Computer".

Device Module Development Toolkit



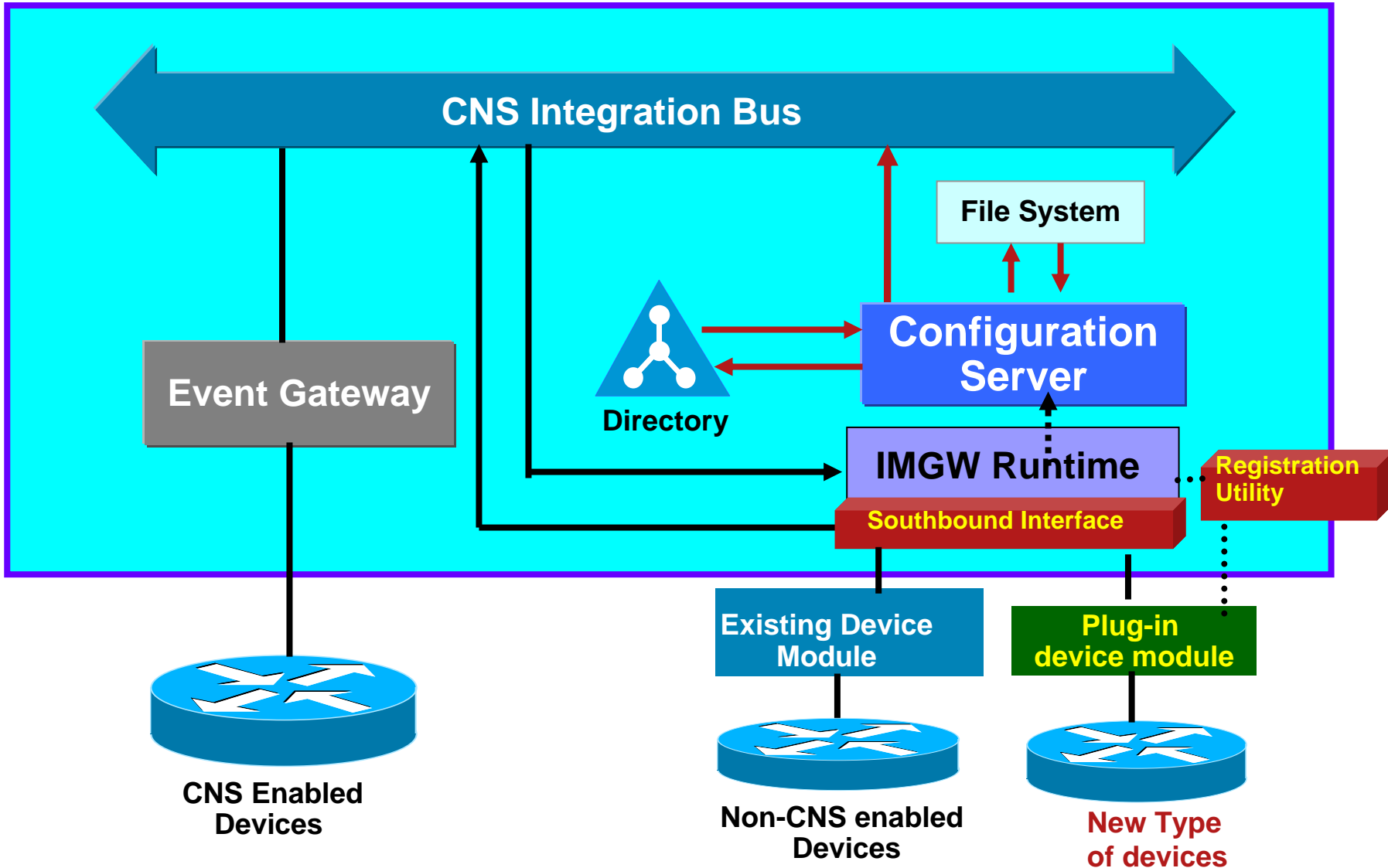
Benefit:

Enable OSS applications to support devices not covered by Configuration Engine

Allow OSS integration developers to develop its own device communication program and plug in to CNS Configuration Engine infrastructure

Support any scripting language and programming language

What is Device Module Development Toolkit?



How to use Device Module Development Toolkit?

Step One

1. Understand how to manage your devices.
2. Develop plug-in device module to conform IMGW southbound interface

Step Two

1. Install the plug-in device module onto CNS CE.
2. Run registration utility to register the plug-in device module into IMGW.

Step Three

Use the plug-in device module to configure devices.

Velocity Templates



Velocity Templates

- User customization based on their device configuration and service activation requirements
- Support for Java, Perl, Expect and other scripting tools
- Enables customers generate configuration dynamically through interaction with the device
- Enable customers develop and plug in scripts to validate device attributes entered by NOC personnel
- Support for scripts to auto-populate attribute values retrieved from customer's data repository
- Native support for
 - Variable
 - Nested Condition
 - Compound Condition
 - Loops
 - Range Operator
- Velocity User Guide: <http://jakarta.apache.org/velocity/user-guide.html>

Velocity Templates - Example

- **New parameter Format**

`#{dsobj.getValue('IOSsubnetmask')}` – Template attribute

- **Variable**

`#set($vpi = 101)` - Digits

`#set($encap = "aal5snap")` - Strings

`#set($subnetmask = "${dsobj.getValue('IOSipaddress')}")` – Substitute template attribute

- **Nested Condition**

```
#if ( $ip_address_needed == "YES" )
  ip address 10.10.1.1 255.255.255.0
  #if ( $no_atm_keepalive == "YES" )
    no atm ilmi-keepalive
  #end
#else
  no ip address
#end
```


Velocity Templates - Example

- **Compound Condition**

```
#if ( $ip_address_needed == "YES" && $no_atm_keepalive == "YES")
  ip address 10.10.1.1 255.255.255.0
  no atm ilmi-keepalive
#else
  no ip address
#end
```

- **Loops and Range Operator**

```
#set( $ip_digits = [1..10] )
#foreach( $ip_d in $ip_digits )
  ip route 30.0.0.$ip_d 255.255.255.255 10.0.0.2
#end
```

Configuration Engine Hardware

Linux Platform

Recommended Specs

- Red Hat Enterprise Server 3.0
- Intel Xeon Processors @ 2.33GHz
- 2 GB RAM - Support for 10,000
- 72 GB HD

Minimum Hardware Specs

- CPU: Intel Pentium III
- OS version: RH release 2.4.X
- RAM: 1GB
- Disk Space: 40GB

Solaris Platform

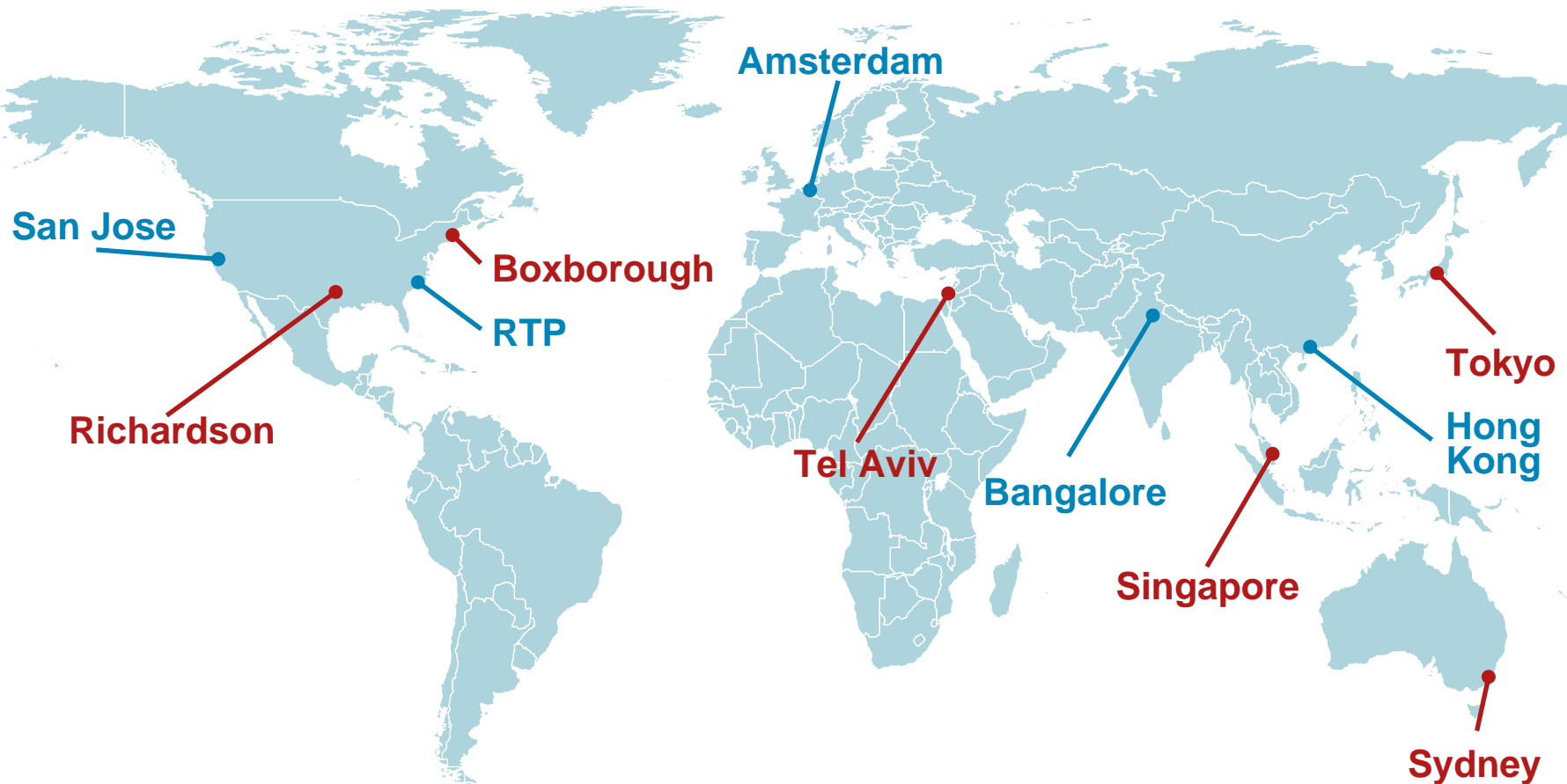
Recommended Specs

- Solaris 2.8
- Sunfire V240 with 2 CPUs @ 1.5GHz
- 4 GB RAM - Support for 20,000
- 72 GB HD

Minimum Hardware Specs

- CPU: Sun Sparc
- OS version: Solaris 2.8
- RAM: 1GB
- Disk Space: 40GB

Reference Customer - Cisco IT-ECT Deployment: 13,000+ users expanding to 30,000+



Management and Data Hub

Data Hub

More Details

- [Cisco Configuration Engine WebSite](#)
- [Data Sheet](#)
- [FAQ](#)
- [Pricing Guide](#)

