



Cisco Configuration Engine 3.5



Agenda



Product Introduction

Day 0 Deployment

Day 2 Configuration and Image Updates

Interfaces: Web GUI and Web Services API

Configuration Templates

Summary and Q&A

Cisco Configuration Engine

Scalable, Secure Solution to Automate Distribution of Cisco® IOS® Software Configuration and Images to Cisco Routers and Switches

Cisco Configuration Engine Product Features

- Day 0 Operations
 - Highly efficient and scalable initial deployment
- Day 2+ Operations
 - Bulk Cisco IOS Software image upgrades
 - Bulk configuration changes
- Template-based configurations
- Web-based GUI
- Web services API
- Secure access over SSL
- Supports RedHat Linux, Solaris and VMware

Configuration Engine - cfigs-sunf1 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://cfigs-sunf1/cns/LoadPage?htmlFilename=home.html

Configuration Engine 3.0(0.0)

Home Devices Users Jobs Tools Image Service UserID: admin Logout

Important Instructions:

- Do NOT use the browser Back and Forward buttons.
- Please navigate using the links in the pages.

Configuration Engine Service Overview

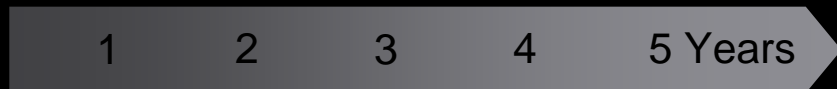
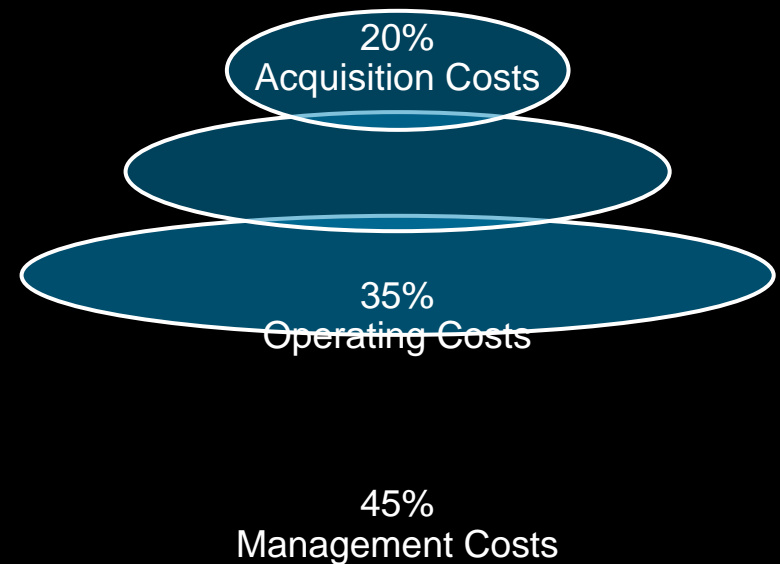
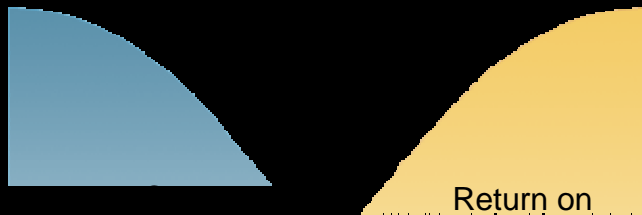
- Devices
 - Device Management and Sub device management.
- Users
 - User Management: Add/Edit/Delete user or Change password.
- Jobs
 - Query/Cancel/Stop/Restart Jobs
- Tools
 - Group Management/Namespace Management/Query Management/Data Management/Directory Management/Template Management/Security Management/Log Management/Service Management/Bulk Data Management/Email Management
- Image Service
 - Images/Search Parameters.

What Is New for Version 3.5?

- Support for RedHat Linux 5.0 and VMware
- External Authentication with Microsoft Active Directory
- Automated port assignment
 - Eliminate the need for changing port number every 500 devices in previous versions
- Dual-zone support
 - Allows separation of device access and application for more flexible security policy per network zone
- Other Enhancements
 - Support Cisco Unique Device Identifier (UDI) as device identifier
 - Secure FTP for importing and exporting configuration templates

Lower Costs through Simplified Management and Operations

- Lower costs of deployment; Cisco IT 12 to 14% savings for Zero Touch Deployment (ZTD) for customer premises equipment (CPE)
- Accelerate CPE deployment
- Reduce manual errors by automation of routine operations



Product Feature Benefits

Secure and Scalable	<ul style="list-style-type: none">▪ Scalable up to 30,000 CPE devices on single server▪ Secure deployment and management over SSL
Initial Deployment	<ul style="list-style-type: none">▪ Significant reduction in time to roll out new service through elimination of staging and manual processes
Configuration Services	<ul style="list-style-type: none">▪ Configuration update to one or a group of devices▪ Email and epage notification of outcome
Image Services	<ul style="list-style-type: none">▪ Policy-based validation of device resources▪ Support for devices behind firewall and dynamic IP addressing
Web-Based GUI	<ul style="list-style-type: none">▪ Feature-rich Web GUI that enables customers to use the product without additional configuration
Configuration Templates	<ul style="list-style-type: none">▪ Customizable to meet customers' business and operation requirements▪ Work flow control and support for scripts
Web Services	<ul style="list-style-type: none">▪ XML and Simple Object Access Protocol (SOAP) Web Service Description Language (WSDL) available for all features supported from the Web GUI

Cisco Configuration Engine: Supported Platforms

Access Routers



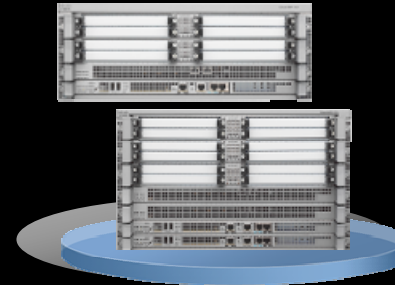
- Cisco 800 Series
- Cisco 1800 Series
- Cisco 1900 Series
- Cisco 2800 Series
- Cisco 2900 Series
- Cisco 3800 Series
- Cisco 3900 Series
- Cisco 1700 Series
- Cisco 2600 Series
- Cisco 3600 Series
- Cisco 3700 Series
- Cisco 3200 Series
- Cisco UC500 Series
- Cisco 500 Series Secure Routers

Access and Metro Switches



- Cisco Catalyst 2950
- Cisco Catalyst 2960
- Cisco Catalyst 3550
- Cisco Catalyst 3560-E
- Cisco Catalyst 3750-E
- Cisco Catalyst 4500
- Cisco Catalyst 3750
- Cisco ME 3400E Series
- Cisco ME 4900 Series
- Cisco Catalyst 6500

Aggregation Routers



- Cisco 7200 Series
- Cisco 7300 Series
- Cisco 7500 Series
- Cisco ASR 1000 Series
- Cisco 7600 Series
- Cisco 10000 Series
- Cisco 10700 Series
- Cisco 12000 Series

Voice and Wireless



- Cisco AS5300 Series
- Cisco AS5400 Series
- Cisco AS5800 Series
- Cisco IAD2400 Series
- Cisco IAD880 Series
- Cisco MWR 1900
- Cisco MWR 2900

Configuration Engine Recommended Hardware

Linux Platform: Red Hat v4.0 and v5.0

Recommended Specifications (20,000 Devices)

- CPU: Intel Xeon Processors
4 @ 2.33 GHz
- RAM: 4 GB
- Hard drive: 72 GB

Minimum Specifications (5,000 Devices)

- CPU: Intel Pentium III
- RAM: 1 GB
- Hard drive: 40 GB

Solaris Platform: Solaris 10

Recommended Specifications (30,000 Devices)

- CPU: Sun T1000 with 8 core,
1.0 GHz UltraSPARC T1 Processor
- RAM: 16 GB
- Hard drive: 146 GB 10K RPM
SAS drive

Minimum Specifications (10,000 Devices)

- CPU: Sun Sparc
- RAM: 1 GB
- Hard drive: 40 GB

VMware

Recommended Specifications

Guest OS specification same
as Linux Platform

Day 0 Initial Deployment



Step 1: Add CPE in Cisco Configuration Engine



Set up device in Cisco Configuration Engine before device deployment

- Add Cisco Network Services ID: Unique device identifier
- Cisco Network Services ID can be hostname, IP address, MAC address, hardware serial number, unique device identifier (UDI), or any string
- Associate configuration template with device

Step 2: Load Bootstrap Configuration on CPE

- Bootstrap configuration is initial set of dynamic startup configuration commands
 - Specific to the customer
 - Service, location, etc., agnostic
- Few lines of Cisco IOS Software Network Services agent configuration commands to execute the one-time initial Dial-Home configuration download
- Several mechanisms to get bootstrap on CPE

```
Router # ip host cns-ce1 10.1.3.7
Router # cns config initial 10.1.3.7 80
Router # cns id hardware-serial
Router # cns id hardware-serial event
Router # cns id hardware-serial image
Router # cns event cns-ce1 11011 keepalive 60 3
Router # cns exec 80
```

Bootstrap Options

Staging

Location-Agnostic
Bootstrap Configuration
for All Devices



Aladdin eToken

Device Removable
Flash Storage
Installed in Devices
at Deployment



IOS Feature to Allow
the Device to Obtain
Bootstrap Configuration
Automatically When
It First Boots Up



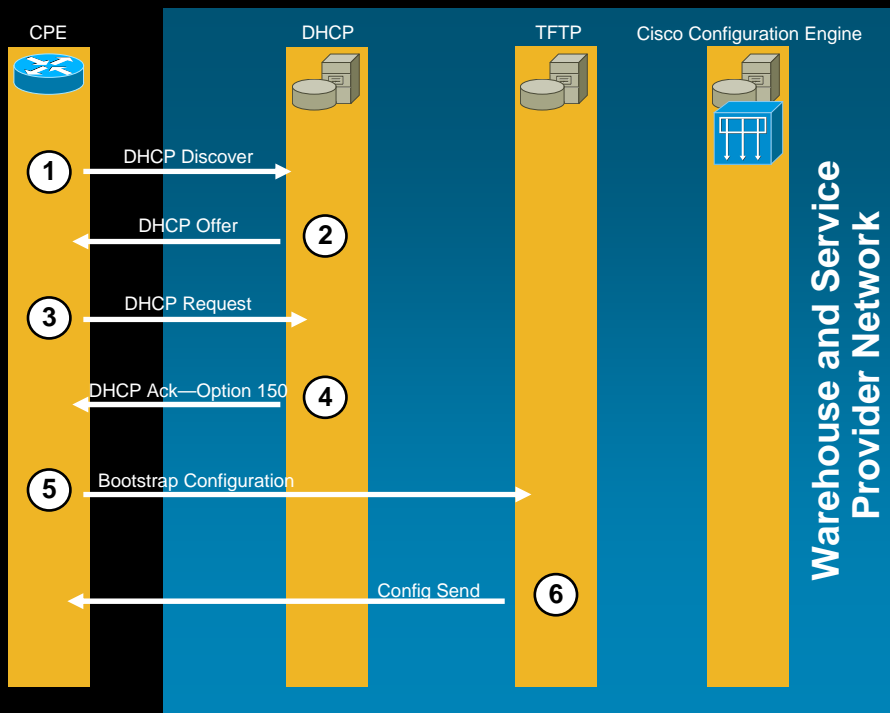
Remote-Site Users Can
Securely Bootstrap CPE
through Web GUI



AutoInstall

Secure Device Provisioning

Bootstrap Options: AutoInstall

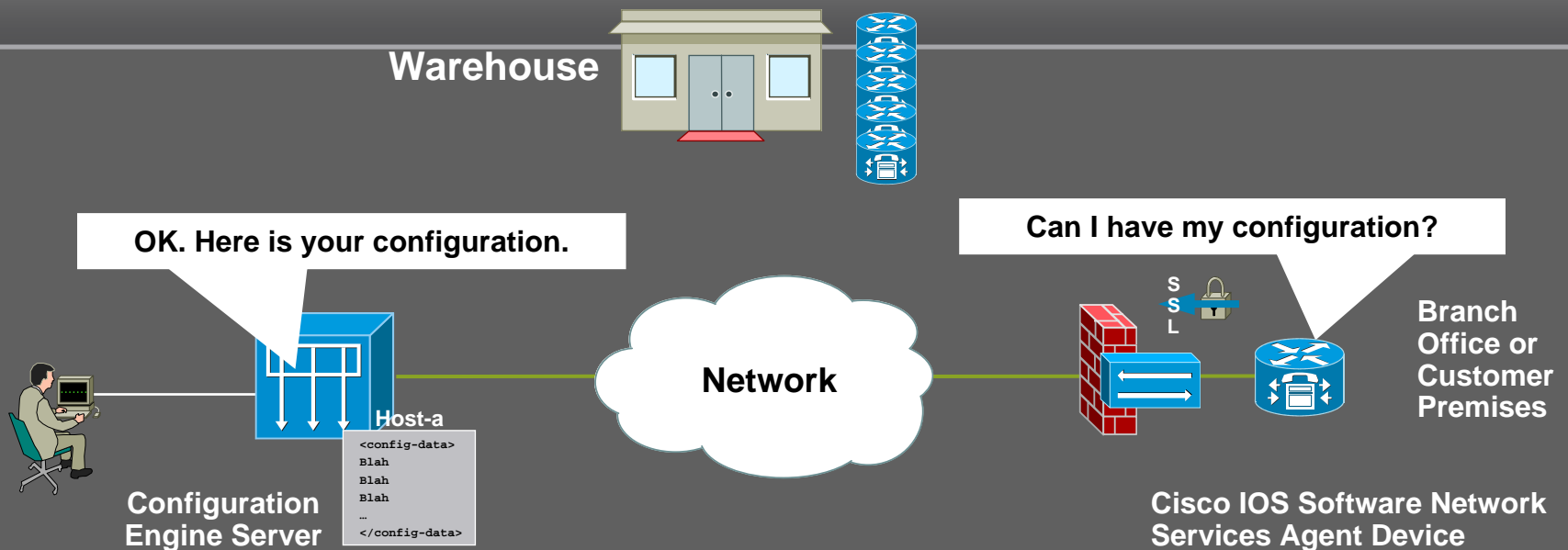


1. CPE sends Dynamic Host Configuration Protocol (DHCP) Discover
2. DHCP server replies with Offer
3. CPE sends DHCP Request
4. DHCP server replies with Option 150
5. CPE requests bootstrap file over TFTP
6. TFTP server sends CPE bootstrap file

Note: To use this option for ISR and ISR G2, use:

- **ISR-CCP-EXP-NOCONF or ISR-CCP-CD-NOCONF** when ordering ISR G2 1900, 2900 and 3900 routers
- **CCP-EXPRESS-NOCF or CCP-CD-NOCF** when ordering ISR 800, 1800, 2800 and 3800 routers

Day 0 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 or 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

Day 2: Post-Deployment Management



Configuration and Image Services

- Configuration Changes

 - Secure configuration updates to thousands of devices in minutes

 - Secure distribution of service configuration (voice, VPN, and security)

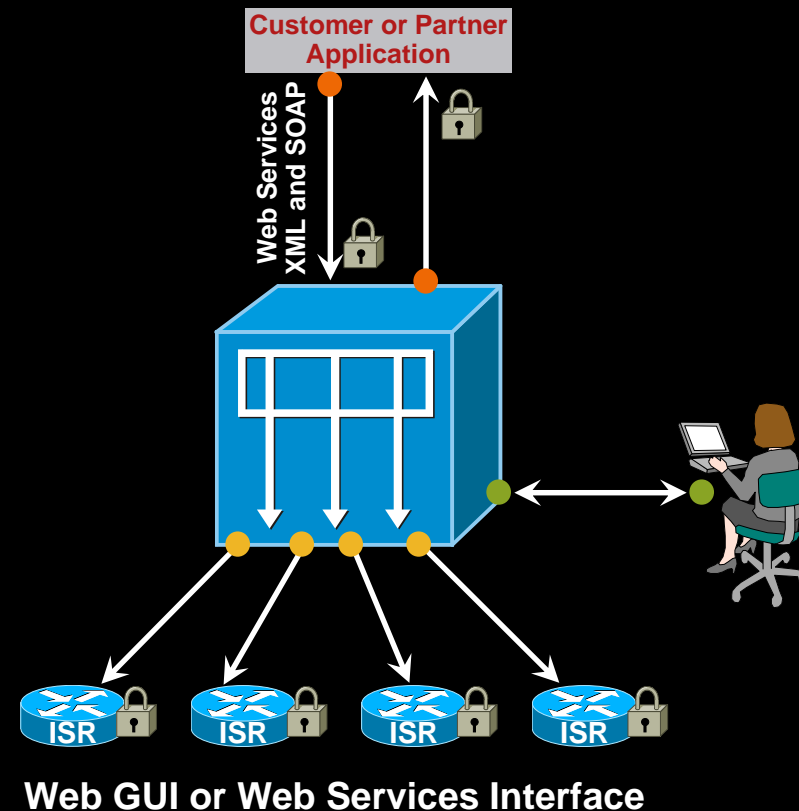
- Image Distribution

 - Cisco IOS Software images, Cisco Catalyst software images, intrusion protection system (IPS) files, Cisco Security Manager files, Cisco IP Phone images, music-on-hold (MOH) files, interactive voice response (TCL IVR) files, and more

- Image Activation: Any File, Anywhere

 - Cisco IOS Software and Cisco Catalyst software images can be activated and the device reloaded and verified

 - Configuration commands can be applied immediately prior to image activation



Hierarchical Group for Any Taxonomy

- Each device can be a member of any number of groups
- Groups within groups are allowed
- Updates of parent group inherit all child groups
- Device data can be queried to create views
- Groups can be based on any number of customer criteria, such as location, services, model number, features, interfaces, and time zones
- Entire branches can be moved from one location to another



Web GUI and Web Services



Hierarchical View

The screenshot displays the Configuration Engine 3.0(0.0) web interface. The top navigation bar includes links for Home, Devices, Users, Jobs, Tools, and Image Service. The user is logged in as 'admin'. The main content area is titled 'View Device' and features a left-hand sidebar with a tree view of configuration groups. The selected group is '/config/PrithiGroup1', which contains several subgroups and individual devices. A list of devices is shown on the right, including PrithiOtherDevice0 through PrithiOtherDevice10.

Configuration Engine 3.0(0.0) Cisco

Home | Devices | Users | Jobs | Tools | Image Service | UserID: admin | Logout

View Device [Advanced Search>>](#)

Groups

- config
 - PrithiGroup1
 - PrithiSubGroup1
 - PrithiSubGroup2
 - PrithiGroup2
 - PrithiSubGroup1
 - PrithiSubGroup2
 - default

Group: /config/PrithiGroup1

- PrithiOtherDevice0
- PrithiOtherDevice1
- PrithiOtherDevice10
- PrithiOtherDevice2
- PrithiOtherDevice3
- PrithiOtherDevice4
- PrithiOtherDevice5
- PrithiOtherDevice6
- PrithiOtherDevice7
- PrithiOtherDevice8
- PrithiOtherDevice9


Device Search and Dynamic Creation of Groups

The screenshot displays the Cisco Directory Administration Tool interface. At the top, the title 'Directory Administration Tool' is on the left, and the Cisco Systems logo is on the right. Below the title is a navigation bar with tabs for 'Home', 'Devices', 'Groups', 'Applications', 'Setup', and 'Bulk Data', along with a 'Logout' link. A dark green sidebar on the left contains a menu with options: 'View Groups', 'Create Group', 'Edit Groups', 'Clone Group', 'Move Group', 'Delete Groups', and 'Create Group using search'. The main content area is titled 'Create Group Using Search' and features a 'Step 1: Search for Devices:' section. Below this, a sample filter string is shown: '[Sample Filter String: ((cn=D*)&(IOSEventID=D*))]'. The search configuration includes three fields: 'Attribute' (set to 'community'), 'Operator' (set to '='), and 'Value' (set to 'Test1'). An 'Add to Query String' button is positioned to the right of these fields. Below the search fields, a 'Query:' label is followed by a text box containing the generated query: 'community=Test1'. At the bottom of the search area are three buttons: 'Reset', 'Query', and '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
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 0 under Directory[]], name=[startup-config],
--

Configuration Services

Configuration Engine 3.0(0.0) 

Home | Devices | Users | Jobs | Tools | Image Service | UserID: admin | Logout

Update Config
Update Image
Customize
<< Up

Step 1:

- Update device with pre-configured template and parameters
- Send Config:
- Select static configuration file: DemoRouter.cfgtpl

Step 2:

Config Action

- Apply to running config
- Apply and save to NVRAM
- Overwrite NVRAM

Step 3:

Syntax Check

Step 4:

If devices are not connected yet, send out triggers again after device connected for 5 minutes.

Step 5:

- Immediate
- At a future time: 00 : 15 (hh:mm) on January 1 2008

Step 6:

Device Batch Size: 20

Step 7:

Text Description for Job:

Image Services

Configuration Engine 3.0(0.0) CISCO

Home | Devices | Users | Jobs | Tools | Image Service | UserID: admin | Logout

Update Config
Update Image
Customize
<< Up

Update Image

Please complete the steps below to perform an Image Update:

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

Step 2: Immediate
 At a future time: 00 : 15 (hh:mm) on January 1 2008

Step 3: Device Batch Size: 2

Step 4: Setup Search Parameters to delete files:

Available Search Parameters:		Selected Search Parameters:
pri ----- End of list -----	<input type="button" value=">>"/> <input type="button" value="<<"/>	----- End of list -----

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

Web Services Overview

- Provides standards-based programmatic access to configuration engine
- Combines WSDL and client library approaches
- Packaged as a separate configuration engine SDK sister product
- Available web services:

ConfigService: Send and acquire configurations to and from devices

ImageService: Distribute and activate images, obtain hardware inventory and file systems, and delete files

ExecService: Execute show commands and reload devices

AdminService: Create and manage system objects used by configuration engine to manage devices (devices, users, groups, templates, etc.)

NSMService: Manage namespace and subjects and subject mappings in namespace

The screenshot shows a web browser window displaying the Cisco CE Web Services API documentation. The page is titled "Cisco CE Web Servi" and lists various methods for the service. The methods are organized into sections: "All Classes" and "All Classes". The methods listed include:

- `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.
- `updateConfigWithTemplate(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.

Configuration Templates



Configuration Engine Templates

- Enables user customization based on device configuration and service activation requirements
- Supports Java, Perl, Expect, and other scripting tools
- Enables customers to generate configuration dynamically through interaction with the device
- Enables customers to develop and plug in scripts to validate device attributes entered by network operations center staff
- Supports scripts to autopopulate attribute values retrieved from customer's data repository
- Provides native support for
 - Variables
 - Nested conditions
 - Compound conditions
 - Loops
 - Range operators
- Velocity user guide:
<http://jakarta.apache.org/velocity/user-guide.html>



Configuration Engine Velocity Templates: Examples

Variables

```
#set( $vpi = 101 ) - Digits
#set( $encap = "aal5snap" ) - Strings
#set( $subnetmask =
"$!{dsobj.getValue('IOSipaddress')}")
- Substitute template attribute
```

Nested Conditions

```
#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
```

```
#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
```

```
#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
```

Compound Conditions

Loops and Range Operators

Case Studies



1. Case Study: EU Incumbent SP

Customer— Project	<ul style="list-style-type: none">▪ Provide managed security and voice services on ISRs to Enterprise and SMB customers
Objective	<ul style="list-style-type: none">▪ Provide IP VPN connectivity, managed FW, managed voice services to customers
Challenges	<ul style="list-style-type: none">▪ Massive configuration updates to several thousands of CPEs simultaneously▪ Massive image upgrades to several thousands of CPEs at a time
Solution	<ul style="list-style-type: none">▪ Seven (7) Config Engines deployed integrated with in-house OSS▪ Uses IMGW to provision all types of CPEs
Achievements	<ul style="list-style-type: none">▪ 50,000 sites deployed to date, planning to expand to 100,000▪ Reduces Day-2 operational expenses via Config Engine▪ Improves operational cash flow from 5-weeks to 5-days▪ Tight integration of provisioning, monitoring and management of CPEs between Configuration Engine and in-house OSS via SDK/CE APIs

2. Case Study: US Service Provider

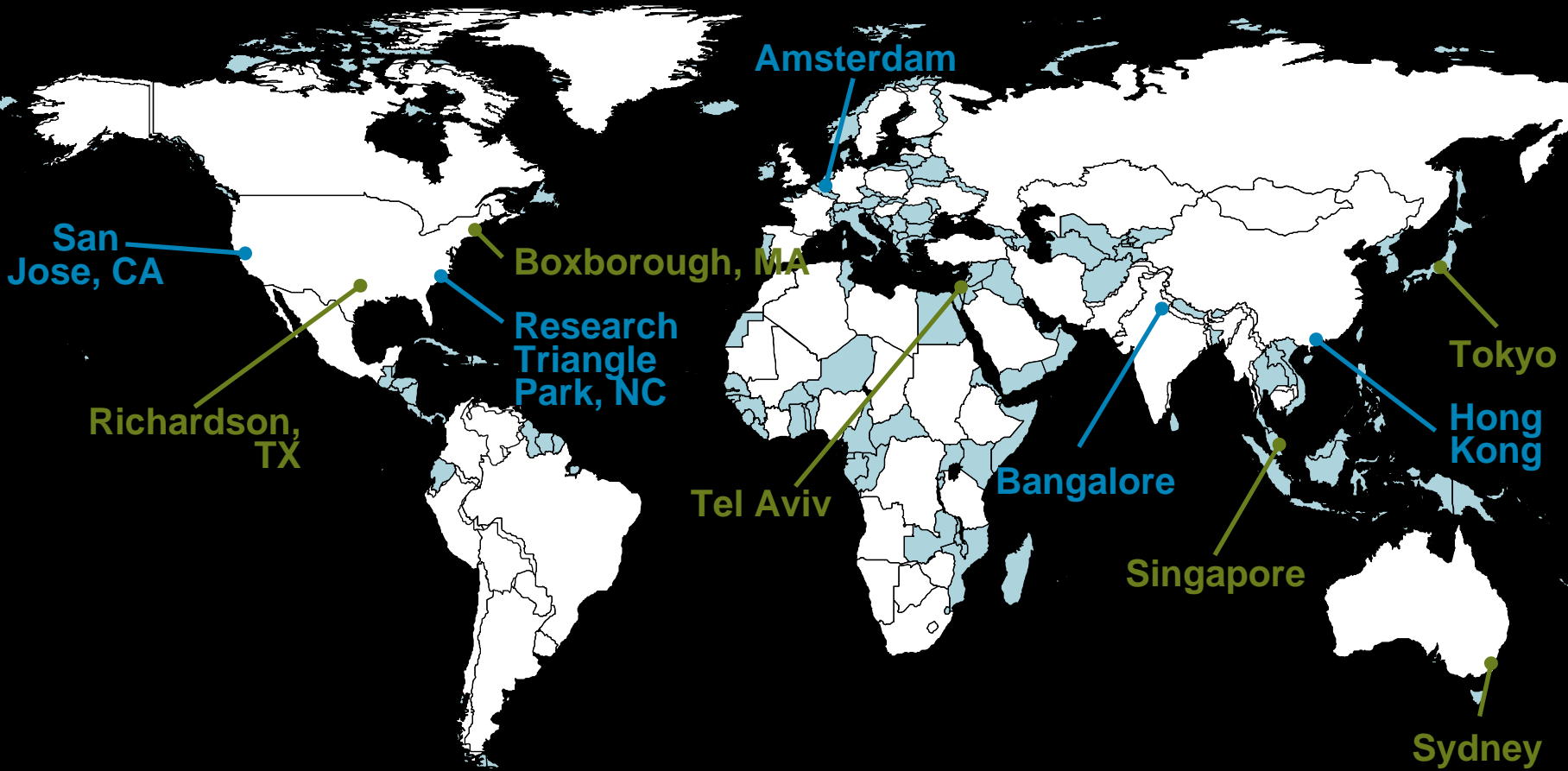
Customer— Project	<ul style="list-style-type: none">▪ Provide host static ip services to SMB customers
Objective	<ul style="list-style-type: none">▪ Provide IP connectivity to connected customers
Challenges	<ul style="list-style-type: none">▪ Configuration updates to several thousands of CPEs at a time▪ Image upgrades to several thousands of CPEs concurrently
Solution	<ul style="list-style-type: none">▪ Two (2) Config Engines integrated with in-house OSS to manage 25K devices▪ Uses CNS Agent to provision all types of CPEs
Achievements	<ul style="list-style-type: none">▪ 25,000 sites deployed to date, planning to expand to 40,000, adding ~350 ISRs/week▪ Reduces Day-2 operational expenses via Config Engine▪ Tight integration of provisioning, monitoring and management of CPEs between Configuration Engine and in-house OSS

3. Case Study: Cisco IT

Customer and Project	<ul style="list-style-type: none">▪ Cisco IT; Cisco Enterprise Class Teleworker Solution at employees' home offices
Objective	<ul style="list-style-type: none">▪ Provide IP Security (IPSec) VPN CPE, managed firewall, managed wireless, and managed unified communications to Cisco employee teleworkers connected over local service providers' broadband Internet service
Challenges	<ul style="list-style-type: none">▪ Unsecure and unreliable management connectivity to CPE▪ Dynamic IP address assignment (by local service provider) to CPE
Solution	<ul style="list-style-type: none">▪ Cisco Configuration Engine integrated into Cisco Virtual Office solution▪ CPE identification over Cisco Network Services
Achievements	<ul style="list-style-type: none">▪ 18,000 sites deployed to date▪ Reliable and secure deployment using Cisco Configuration Engine▪ Fully managed CPE model (configuration change is considered security breach)

3. Case Study: Cisco IT (Cont.)

More than 18,000 Users Expanding to More than 30,000



Management and Data Hub

Data Hub

Resources: Cisco Configuration Engine

- Cisco Configuration Engine Website

<http://www.cisco.com/en/US/products/sw/netmgtsw/ps4617/index.html>

- Cisco Configuration Engine Overview Presentation

http://www.cisco.com/en/US/products/sw/netmgtsw/ps4617/prod_presentation_list.html

- Cisco Configuration Engine Data Sheet

http://www.cisco.com/en/US/products/sw/netmgtsw/ps4617/products_data_sheets_list.html

- Cisco Configuration Engine FAQ

http://www.cisco.com/en/US/products/sw/netmgtsw/ps4617/prod_qandas_list.html

- Cisco Virtual Office Solution

<http://www.cisco.com/go/cvo>

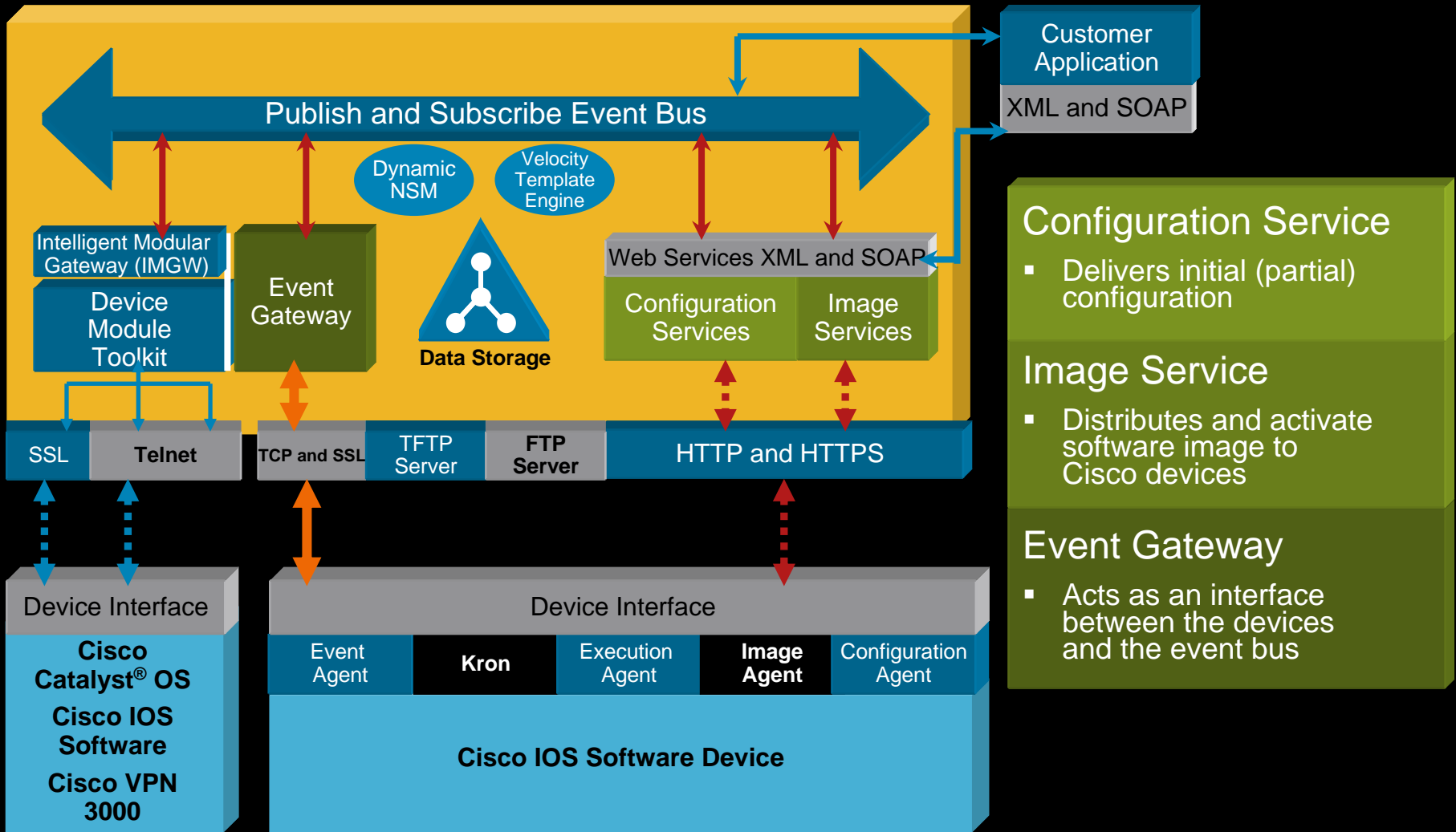
Q and A





Copyright. 2010 Cisco Systems, Inc. All rights reserved. Cisco and the Cisco logo are trademarks or registered trademarks of Cisco Systems, Inc. or its affiliated entities in the United States and other countries. All other trademarks are the property of their respective owners.

Cisco Configuration Engine Architecture



Support for Platforms Other than Cisco IOS Software Network Services or Other Cisco Platforms

Device Development Module (IMGW)

- Device Communication Driver

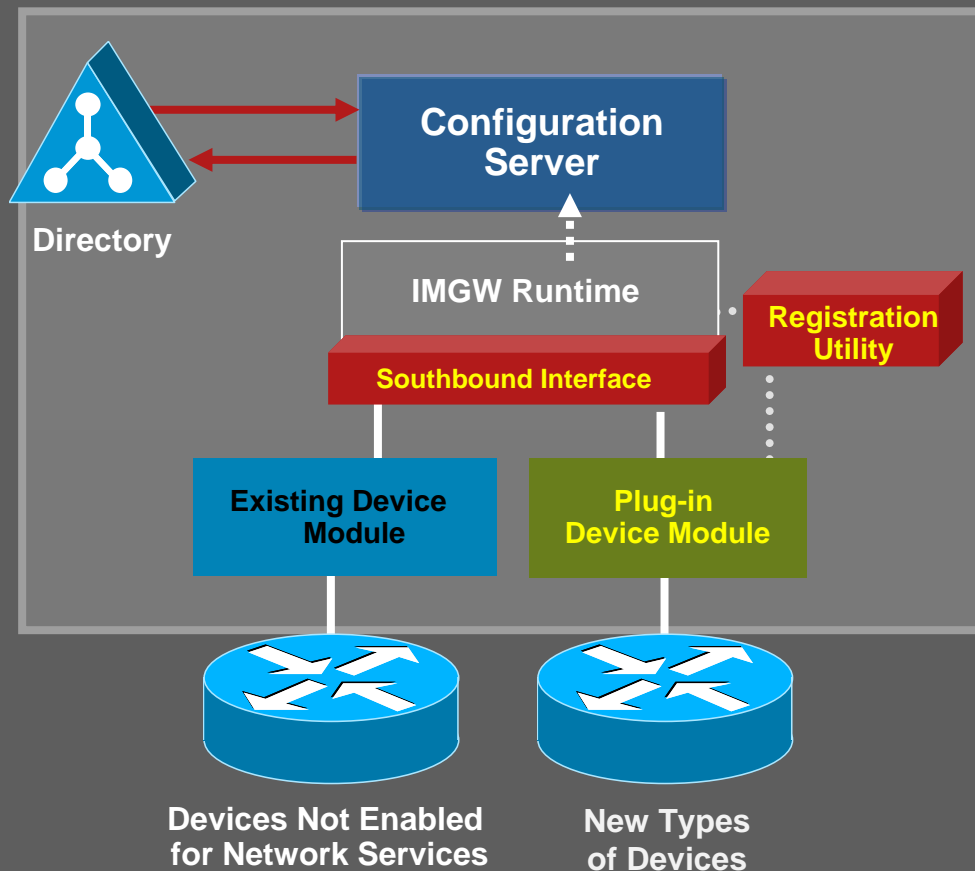
Extend configuration engine to support protocol native to the device (Telnet, SSH, HTTP, TL1, etc.)

Enable customers to develop their own scripts and communicate with devices not supported by configuration engine

- Sample Applications

Non-IP: Develop module to communicate to the devices through modem or console server and use all product functions available through GUI or API

Third-party devices: Extend configuration engine functions to support third-party devices and platforms



Deploying Redundancy

- Recommend deploying CCE cluster in a n+1 configuration
 - At most up to one failure can be handled in this configuration
- **Completely redundant configuration requires twice the number of CCEs**
- Recommend the use of an external LDAP server
 - Recommend commercially available LDAP servers be used instead of open source LDAP server (for support issues)
- Redundancy setup is completely transparent to the day-to-day operation of the CE
- Failover to the standby (from Active) is automatic

Deploying Load Sharing

- Employ external LDAP for ease of sharing devices database
- Requires the use of an external network device (like ACE, IOS load balancer) to proxy the CE server farm
- CPEs can established initial Day-1 connection with one CE and receiving Day-2 updates with another CE
- Load may or may not be equally distributed amongst the CE cluster—it depends on load sharing algorithm deployed (RR, least connections, hash)
- Recommend to use least-connections algorithm to take account CE server load