



# InCommon Certificate Manager

# InCommon CM Extra Agent

InCommon c/o Internet2 1000 Oakbrook Drive, Suite 300 Ann Arbor MI, 48104



### **Table of Contents**

| 1 Introduction  | 3  |
|---|----|
| 2 InCommon CM Extra Agent Infrastructure                | 5  |
| 3 InCommon CM Extra Agent Workflow                      | 7  |
| 3.1 Configuration Synchronization and Discovery         | 7  |
| 3.2 Certificate Enrollment, Renewal and Installation    | 9  |
| 4 InCommon CM Extra Agent Communication Interfaces      | 10 |
| 4.1 InCommon CM Extra Agent - InCommon CM               | 10 |
| 4.2 InCommon CM Extra Agent - Web Server Control System | 11 |
| 5 InCommon CM Extra Agent States                        | 14 |
| 6 Inject InCommon CM with Extra Agent                   | 17 |
| 7 Prepare Apache runtime                                | 17 |
| 8 Schedule  | 17 |



# **1** Introduction

The InCommon Certificate Manager (CM) Extra Agent is a single agent installed on customer premises that performs two core services for the product - automatic SSL installation and certificate discovery:

- Certificate installation The agent will communicate with your remote web-hosts, inform you when a certificate is due to expire and can automatically apply for and install certificates on your hosts. The agent is installed on a local machine, configured via InCommon CM and can also be set to communicate directly with InCommon CA infrastructure through a proxy server.
- Certificate discovery The agent allows administrators to scan local hosts for SSL certificates, certificates issued by third party vendors and self-signed certificates. Discovery scan ranges are configured in InCommon CM. The agent will connect to hosts in the range over SSL/TLS connections. All discovered certificates can be imported into InCommon CM for management.

The key principles governing the agent are privacy and security. The agent is installed on a machine on customer premises and uses a pull mechanism to communicate with InCommon CM. This ensures the agent doesn't require open ports on the agent side. Communication uses the SOAP messaging protocol with TLS/SSL encryption.

| Configuration<br>Web UI  | Web Service Client<br>(SOAP)               |          |  |  |  |  |  |
|--------------------------|--|----------|--|--|--|--|--|
| Certificate<br>Discovery | Certificate Installer<br>(Local Shell/SSH) | KeyStore |  |  |  |  |  |
|                          | Java                                       |          |  |  |  |  |  |

### InCommon CM Extra Agent Architecture

#### Configuration

InCommon Certificate Manager | InCommon CM Extra Agent |

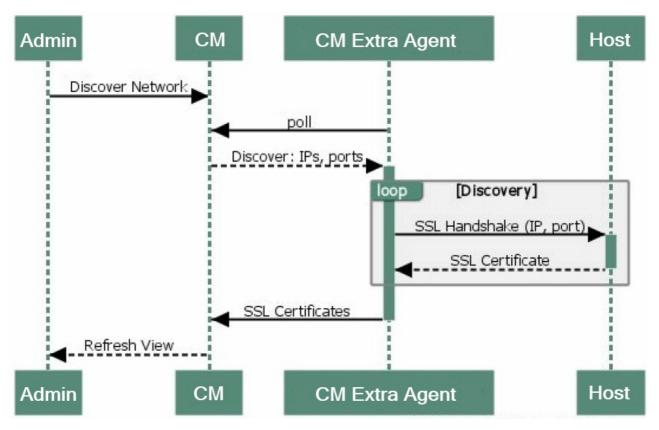


The agent can be configured via a configuration file or by using the agent's local configuration Web UI.

#### Auto-Update

The agent periodically checks whether a new version is available on InCommon CM. If available, the agent can download and install the update automatically.

#### **Discovery Data Flow**



- An admin configures the discovery scan in InCommon Certificate Manager. Scan parameters include the IP range (CIDR) and port range.
- The agent pulls these parameters from InCommon CM and starts the discovery process.
- Certificates are discovered by the agent simply making a HTTPS connection to each host-port pair. Each host in the range will send their certificate as part of the regular SSL 'handshake' process. Each certificate received contains publicly available information such as the public key, certificate subject, issuer, key size, validity, etc.
- The agent sends this information to InCommon CM so that the admin may import the discovered certificates into the InCommon CM workflows and manage them going forward.

#### **Hardware Requirements**

The agent can be used with either the default or a custom thread count setting. The maximum thread count of 4000 speeds up the discovery process for big IP ranges.

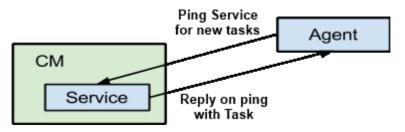
RAM

- 1 GB for default threads count value
- 4 GB for maximum threads count value.



# 2 InCommon CM Extra Agent Infrastructure

InCommon Certificate Manager generates services and presents them to the agent whenever new tasks are generated. The agent polls InCommon CM periodically to see if any new tasks are generated. If any pending task is present, InCommon CM sends Task java object which contains all needed data to the agent. The agent does the job and calls corresponding InCommon CM service to post the results.



Following is the list of tasks:

- Sync/Refresh configuration
- Discover certificates in the net by CIDR
- Discover certificates on local FS
- Enroll certificate for installation on Web Server
- Renew certificate in case of expiration
- Install certificate on Web Servers (IIS, Apache, Tomcat)

InCommon CM generates a queue for tasks to send to the agent. The task may have different states:

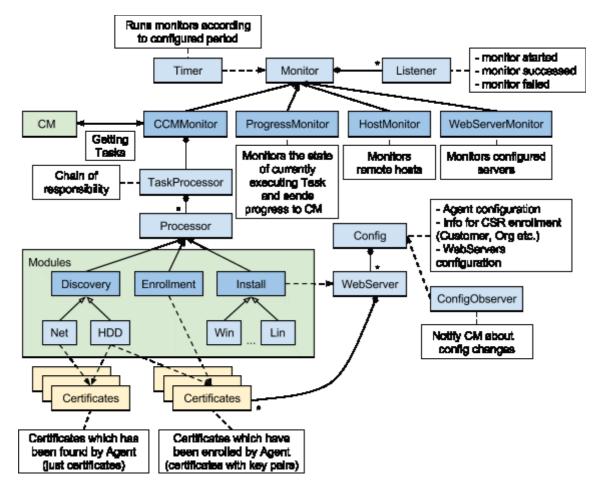
- Requested
- Done
- Failed

Also task should have some text definition like error description etc.

The agent consists of three Monitors which are run periodically by a timer according to set intervals. Each Monitor has a number of Listeners which are notified about monitored changes.

- 1. ProgressMonitor Watches currently processing task and sends its progress to InCommon CM
- 2. WebServerMonitor Follows changes in the configuration of web servers
- 3. HostMonitor Checks the availability of configured hosts in InCommon CM Extra Agent's network
- 4. The Incommon CM Monitor Contains a Task Processor which delegates the received Tasks to respective processing modules:
  - Discovery Module Responsible for discovering certificates in agent's local network and on local file system. The discovery process on local File Server (FS) should be able to detect and parse Key Stores. In case of valid Key Pair and Certificate, this Key Store should become the target for Installation module.
  - ii. Enrollment Module Responsible for enrolling new certificates for Extra agent to install on Web Servers.
  - iii. Installation Module Responsible for installation of certificates. Different implementations are available for different operating systems (OSs) and Web Servers.





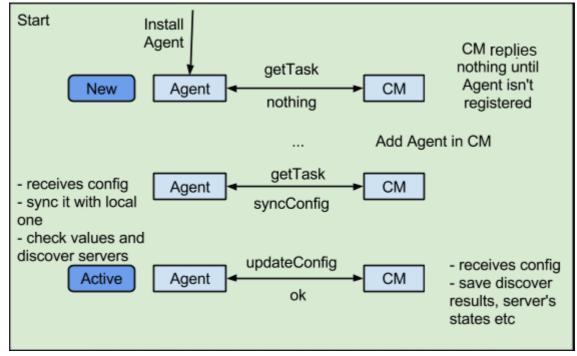
The agent maintains its configuration in a xml file and loads it on start. In case of any changes the configuration file is automatically updated. The agent also stores the (installable) certificates data, containing the bunch CertId and the KeyStore's location.

The agent should periodically check configuration changes in the server on which it is installed and update the same on the InCommon CM server.



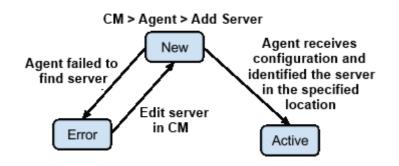
### 3 InCommon CM Extra Agent Workflow

#### 3.1 Configuration Synchronization and Discovery

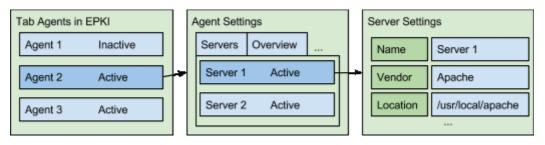


#### Sync Configuration

The agent receives the configuration changes from the InCommon CM through Tasks and syncs it with the local configuration file. Also, if a new server has been added to the network, the agent determines its details like vendor, version etc and updates its local configuration and sends them to the CM.



The diagram below explains the UI steps for the agent's configuration.



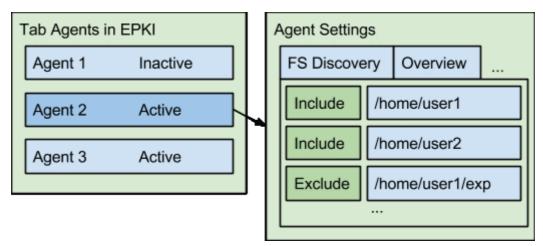


#### Discovery

The InCommon CM Extra Agent covers the discovery through Net. The local FS discovery should be configurable with the 'INCLUDE' and 'EXCLUDE' rules. The agent scans the servers and discovers Certificates in two types:

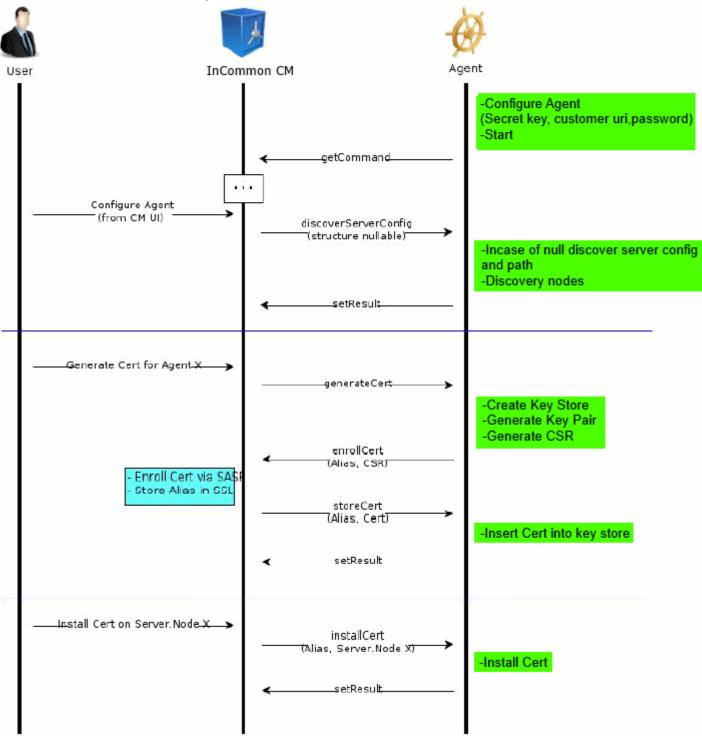
- Simple Certificate
- Certificate with the Key Pair

The certificate with key pair is installable and is the target for installation on Web Server.





#### 3.2 Certificate Enrollment, Renewal and Installation



#### Enrollment

InCommon CM sends a task to enroll new certificate for installation on Web Server to the agent. Upon receiving the task the agent should:

- i. Check that there is no pending enrollment
- ii. Generate and store Key Pair for new certificate
- iii. Generate CSR using this Key Pair and send CSR to InCommon CM



- iv. InCommon CM in its turn should process CSR, enroll new certificate and pass it to the Agent
- v. Agent will receive the certificate and store it into Key Store together with Key Pair. This Key Store will be target for Installation module. Agent also stores the bunch CertId and KeyStore location on its side, to be able to install the certificate in future.

#### Renewal

In case of expiration of a certificate the agent will:

- i. Request InCommon CM for renewal with renew\_id
- ii. InCommon CM will renew the certificate and pass it back to the agent.
- iii. The agent will receive the certificate and replace the old one. This could require the Web Server restart in case if the certificate has been installed somewhere.

#### Installation

Installation is organized like wizard. For Tomcat and Apache there should be the following common steps:

- i. Select certificate to install (this should be installable certificate with KeyPair stored on the Agent's side), it will automatically select the Agent.
- ii. Select the Web Server.
- iii. Select Connector or VirtualHost to use (this info is getting from server's config).
- iv. Start installation.
- v. Backup old configuration.
- vi. Change configuration to use SSL connection with new Certificate.
- vii. Restart Web Server.

# 4 InCommon CM Extra Agent Communication Interfaces

#### 4.1 InCommon CM Extra Agent - InCommon CM

- getCommand(AgentAuthData authData)
- sendResult(AgentAuthData authData, String commandId, CcmExtraAgentResult result)
- sendProgress(...)
- enrollCert(AgentAuthData authData, String alias, String csr) ???keyStoreId

#### InCommonCM Extra Agent receives the following commands:

- RunCD
- NoCommand
- SetPoolInterval
- DiscoverServerConfig(CcmExtraAgentDTO agent)
- GenerateCert(String commonName) Generate CSR
- StoreCert(String cert, String alias) Receive and store certificate in local Key Store
- InstallCert(TargetServerNodeId serverNodeId) Install certificate on target server/node
- Restart



#### 4.2 InCommon CM Extra Agent - Web Server Control System

#### 1. Getting Web Server Config Info.

| A. Command line:  |
|---|
| i. Unix (Apache): ./getwebserverconfiginfo.sh %s                |
| ii. Windows (IIS): ccmapi32.exe /command getwebserverconfiginfo |
| B. Unix input params:   |
| i. String apachectrlPath  |
| C. Windows input params: None                                   |
| D. Result output example:                                       |
| <webserverconfiginforesult></webserverconfiginforesult>         |
| <webserver></webserver>   |
| <name>Apache IIS</name>   |
| <version>2.2.6 6.0</version>                                    |
| <nodes></nodes>   |
| <node></node>   |
| <protocol>HTTP HTTPS</protocol>                                 |
| <address>195.5.5.1 *</address>                                  |
| <servername>foo.com</servername>                                |
| <port>443</port>  |
| <cert>/usr/certs/host.cer</cert>                                |
| <status>????</status> ??????                                    |
|   |
|   |
|   |
| <status>Running Stopped </status> <math ????? →                 |
| <listenports></listenports>                                     |
|   |



| <listenport>80</listenport>                     |  |
|---|--|
| <listenport>443</listenport>                    |  |
| <listenport>8443</listenport>                   |  |
|   |  |
|   |  |
| <os></os>                                       |  |
| <name>Windows Linux</name>                      |  |
| <version>7 XP Debian</version>                  |  |
| <build>6.1.7600 5.1.2600 2.6.32-5-amd64</build> |  |
| <arch>x86_32 x86_64</arch>                      |  |
|   |  |
| <resulterrorcode>0</resulterrorcode>            |  |
| <resulterrormessage></resulterrormessage>       |  |
|   |  |
|   |  |

#### 2. Install Certificate.

| A. Apache command line:   |
|---|
| ./getinfotoinstallcertificate.shnode-ip=%snode-port=%snode-servername=%sapache-ctl=%s |
| B. Apache WebServer input params:   |
| i. String nodeAddress   |
| ii. String nodeServerName   |
| iii. int nodePort   |
| C. Apache WebServer result output example:  |
| <apacheinfotoinstallcertificateresult></apacheinfotoinstallcertificateresult>         |
| <nodeconfigpath>/opt/apache/cfg/1.cfg</nodeconfigpath>                                |
| <nodeconfiglinenumber>332</nodeconfiglinenumber>                                      |
| <resulterrorcode>0</resulterrorcode>  |



| <resulterrormessage></resulterrormessage>   |
|---|
|   |
| D. IIS command line:  |
| ccmapi32.exe /command installcertificate /node-ip %s /node-port %s /node-servername %s /certificate-aliasname %s<br>/keystorefile %s /keystorepass %s |
| E. IIS WebServer input params:  |
| i. String certificateFriendlyName   |
| ii. String nodeAddress  |
| iii. String nodeServerName  |
| iv. int nodePort  |
| F. IIS WebServer result output example:   |
| <li>lisInstallCertificateResult&gt;</li>  |
| <webserverstatus>Running Stopped</webserverstatus> < ????   |
| <resulterrorcode>0</resulterrorcode>  |
| <resulterrormessage></resulterrormessage>   |
|   |

#### 3. Restart Web Server.

#### A. Command line:

- i. Unix (Apache): ./restart.sh %s
- ii. Windows (IIS): ccmapi32.exe /command restart

#### B. Unix input params:

i. String keyStorePath

#### C. Windows input params: None

D. Result output example:

<RestartWebServerResult>

<WebServerStatus>Running|Storred|...</WebServerStatus> <!-- <--- ??????  $\rightarrow$ 

<ResultErrorCode>0</ResultErrorCode>

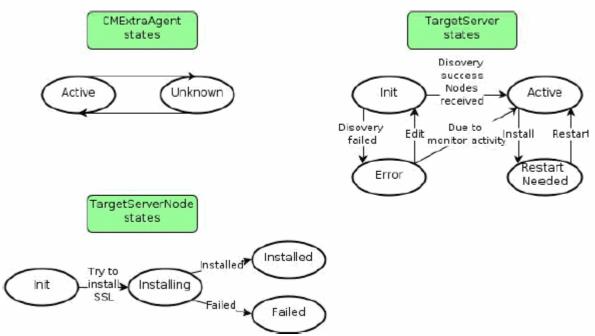


<ResultErrorMessage></ResultErrorMessage>

</RestartWebServerResult>

# 5 InCommon CM Extra Agent States

The state change diagram for main CM Extra entities



#### **State-Command Relationships:**

| Comman<br>d             | n Comma<br>nd State |                   |                   |               |      |            | Server Sta | ite                 | Node State  |            |               |        |
|-------------------------|---------------------|-------------------|-------------------|---------------|------|------------|------------|---------------------|-------------|------------|---------------|--------|
|                         |                     | NOT_AVAILAB<br>LE | NOT_CONNEC<br>TED | CONNECTE<br>D | INIT | ACTIV<br>E | ERROR      | RESTART<br>REQUIRED | UNKNOW<br>N | INSTALLING | INSTALLE<br>D | FAILED |
| DiscoverS<br>erverConfi |                     |                   |                   | *             |      | *          |            |                     | *           |            | *             |        |
| g                       | Part<br>Success     |                   |                   | *             |      | *          | *          |                     | *           |            | *             |        |
|                         | Failed              |                   |                   | *             |      |            |            |                     |             |            |               |        |
| Generate<br>Certificate | Success             |                   |                   | *             |      |            |            |                     |             |            |               |        |
| *                       | Failed              |                   |                   | *             |      |            |            |                     |             |            |               |        |
| StoreCerti<br>ficate*   | Success             |                   |                   | *             |      |            |            |                     |             |            |               |        |
|                         | Failed              |                   |                   | *             |      |            |            |                     |             |            |               |        |
| InstallCert<br>ificate* | Created             |                   |                   | *             |      |            |            |                     |             | *          |               |        |
|                         | Success             |                   |                   | *             |      |            |            | *                   |             |            | *             |        |
|                         | Failed              |                   |                   | *             |      |            |            |                     |             |            |               | *      |
| Restart                 | Success             |                   |                   | *             |      | *          |            |                     |             |            |               |        |
|                         | Failed              |                   |                   | *             |      |            | *          |                     |             |            |               |        |

| Command    | This state only | If last request | Create |  |  |  |  |
|------------|-----------------|-----------------|--------|--|--|--|--|
| s with     | until Agent     | from Agent was  | d Only |  |  |  |  |
| asterisk   | request first   | long ago        |        |  |  |  |  |
| should be  | command         |                 |        |  |  |  |  |
| restartabl |                 |                 |        |  |  |  |  |
| e in case  |                 |                 |        |  |  |  |  |
| of failure |                 |                 |        |  |  |  |  |
|            |                 |                 |        |  |  |  |  |



# 6 Inject InCommon CM with Extra Agent

The latest build for the agent is available from the FTP location:

ftp://91.196.95.17/release/< latest InCommon CM version>

For example, the agent build for the InCommon CM version 5.8, is available from ftp://91.196.95.17/release/5.8

• Download the file 'cd-agent.tar.gz' and extract it to [tomcat]/cd\_agent/ folder

Note: Make sure that all .sh files have executable permissions.

### 7 Prepare Apache runtime

- Install Apache using std tools of your distributive
- Create config file (anyname) in /etc/apache2/conf.d

Listen 8010

NameVirtualHost \*:8010

<VirtualHost \*:8010>

DocumentRoot /var/www/html/test.vhost.com

ServerName test.vhost.com

</VirtualHost>

Get apache bin by `whereis -b apachectl`, by default: '/usr/sbin/apachectl'

# 8 Schedule

