

- e From the Value list, select Get Questions > Number of products deployed as shown in Figure B-9.

Figure B-9: GetQuestions>Number of products deployed

- f Click Add.

- 16 Create a second update action in which you set the compliance flag to return a true or false to indicate whether the licenses are compliant. Leave the Update dialog box open and complete the rest of the steps.

TIP

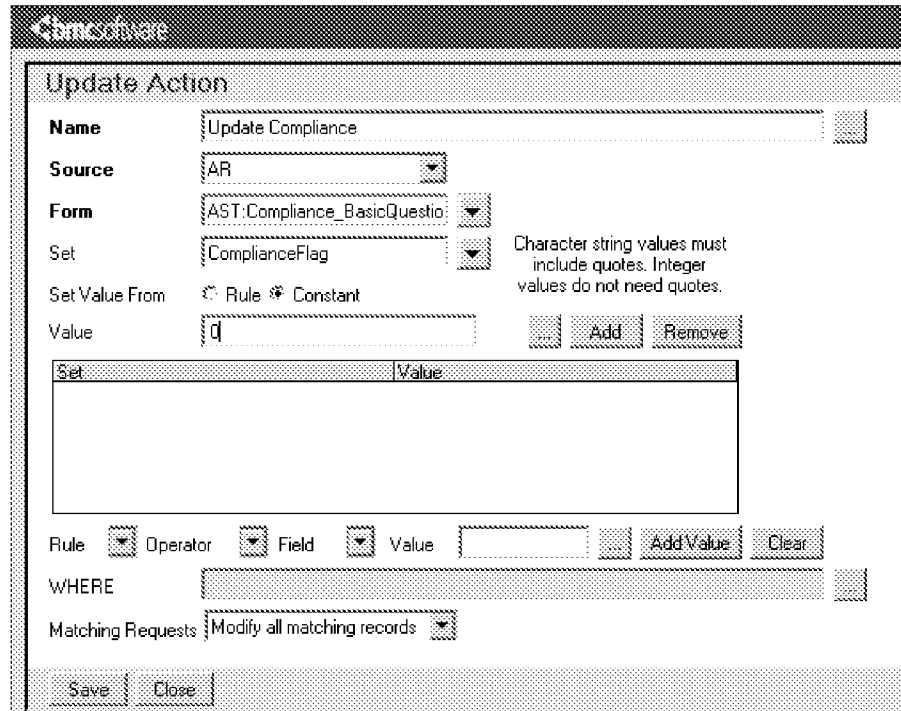
For the Site default license type, if a site licenses exists, it is compliant because all of the license certificates fall under the site license. That means the compliance flag can be set to 1, indicating true for compliance.

- a For the Source select AR.
 b From the Set list, select ComplianceFlag. This value indicates that the question field should be updated based on the count retrieved in the Get.
 c In the Set Value From field, select Constant.
 d In the Value field, enter 1.

The compliance flag is set to 1 because this is the true setting and the site license is compliant.

The Update Action dialog box should appear as shown in Figure B-10.

Figure B-10: Selections to set compliance flag value



e Click Add.

Both update actions appear in the list.

17 Now build the WHERE qualifier for the update action.

a In the Matching Requests list, select Modify all matching records.

b From the Rule list, select GetCertificates > instanceId.

This is added to the WHERE field as the clause starts to be built.

c From the Operator drop-down, select =.

d From the Value list, select Certificate_InstanceId.

The WHERE qualifier should look like Figure B-11.

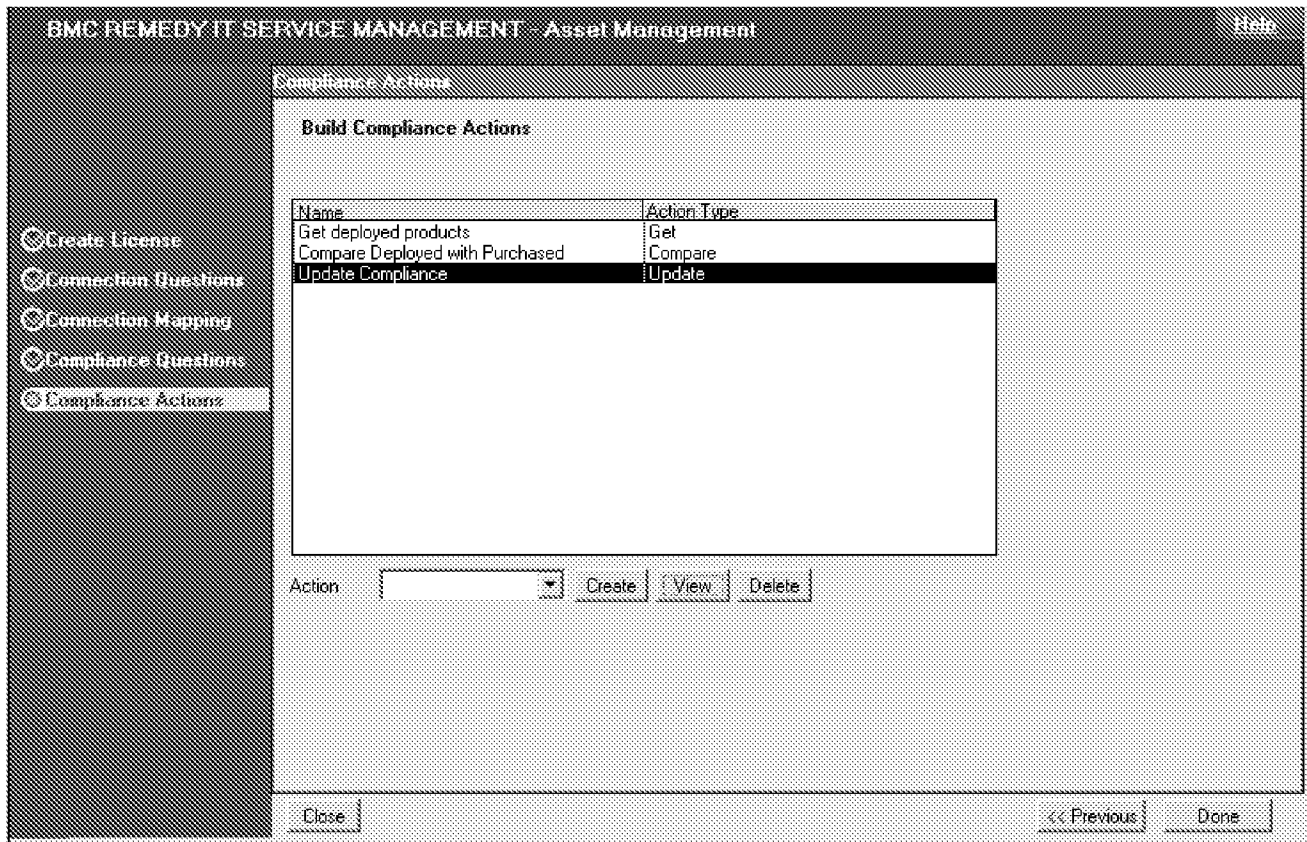
Figure B-11: WHERE qualifier built



18 Click Save.

The Compliance Actions page of the license wizard is redisplayed and all of the compliance actions appear in the list as shown in Figure B-12.

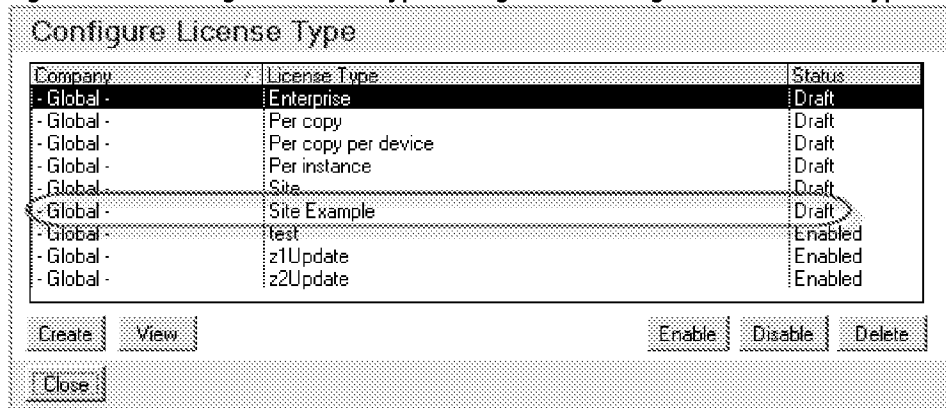
Figure B-12: Compliance Actions page showing all of the actions created



- 19 Click Done.
- 20 At the prompt, click Yes to create the license type.

The license type is created and displayed in the list in the Configure License Type dialog box, as shown in Figure B-13.

Figure B-13: Configure License Type dialog box showing the new license type



By default the license type is created in draft mode.

- 21 To enable the license type, select it and click Enable.
- 22 Click Close.

Per instance license type example—basic mode

This section provides an example to show you how to create a custom license type in basic mode based on the default Per instance license type. For more information about default license types, see “Default system license types” on page 418.

► **To create the site example license type in basic mode**

- 1 From the Application Administration Console, click the Custom Configuration tab.
- 2 From the Application Settings list, choose Asset Management > Advanced Options > Configure License Type, and then click Open.

The Configure License Type dialog box appears as shown in “Configure License Type dialog box” on page 498.

- 3 Click Create.

The license type wizard starts.

- 4 Update the Create License Type page as follows.
 - a Select Global as the company. This makes the license type available to all companies.
 - b Enter a name for the license type. For example, you would enter Per instance example.

..... **NOTE**

Ensure that you do not name this license type Per instance, since that license type already exists in the system.

.....

- c Enter a description for the license type. For example, you would enter Example per instance license that includes all instances of a software application.

By default, the status is set to Draft.

- d For Groupable, select Yes.
- e For License Type Mode, select Basic.

..... **NOTE**

After a license type is created, it has to be enabled. This example shows you how in a later step.

.....

Figure B-14 shows what the Create License Type page should look like.

Figure B-14: Create License Type page filled in for the example

The screenshot displays the 'Create License Type' page in BMC Remedy IT Service Management. The page title is 'BMC REMEDY IT SERVICE MANAGEMENT - Asset Management'. The main content area is titled 'Create License Type' and contains the following information:

- License Type Information:**
 - Company:** Global
 - License Type:** Per instance example (with a 'Manage Locales' button)
 - Description:** Example per instance license that includes all instances of a software application
 - Status:** Draft
 - Groupable:** Yes
 - License Type Mode:** Basic

The left sidebar shows a navigation menu with the following items: Create License (selected), Connection Questions, Connection Mapping, Compliance Questions, and Compliance Actions. At the bottom of the page, there are three buttons: Close, Previous, and Next >>.

5 Click Next.

The Build Connection Questions page of the license type wizard appears. For this license type, the entire set of CIs are used so no connection questions are necessary to filter the set.

6 Click Next to bypass the Build Connection Questions page.

The Connection Mapping page is also bypassed and the Build Compliance Questions page appears.

7 On the Compliance Questions page that appears, build the first compliance question as follows:

- a Enter the question. For this example, you would enter Number of licenses purchased.
- b For Field Type, select Int.

Since this is also a groupable license type, the Integer Field Type field appears.

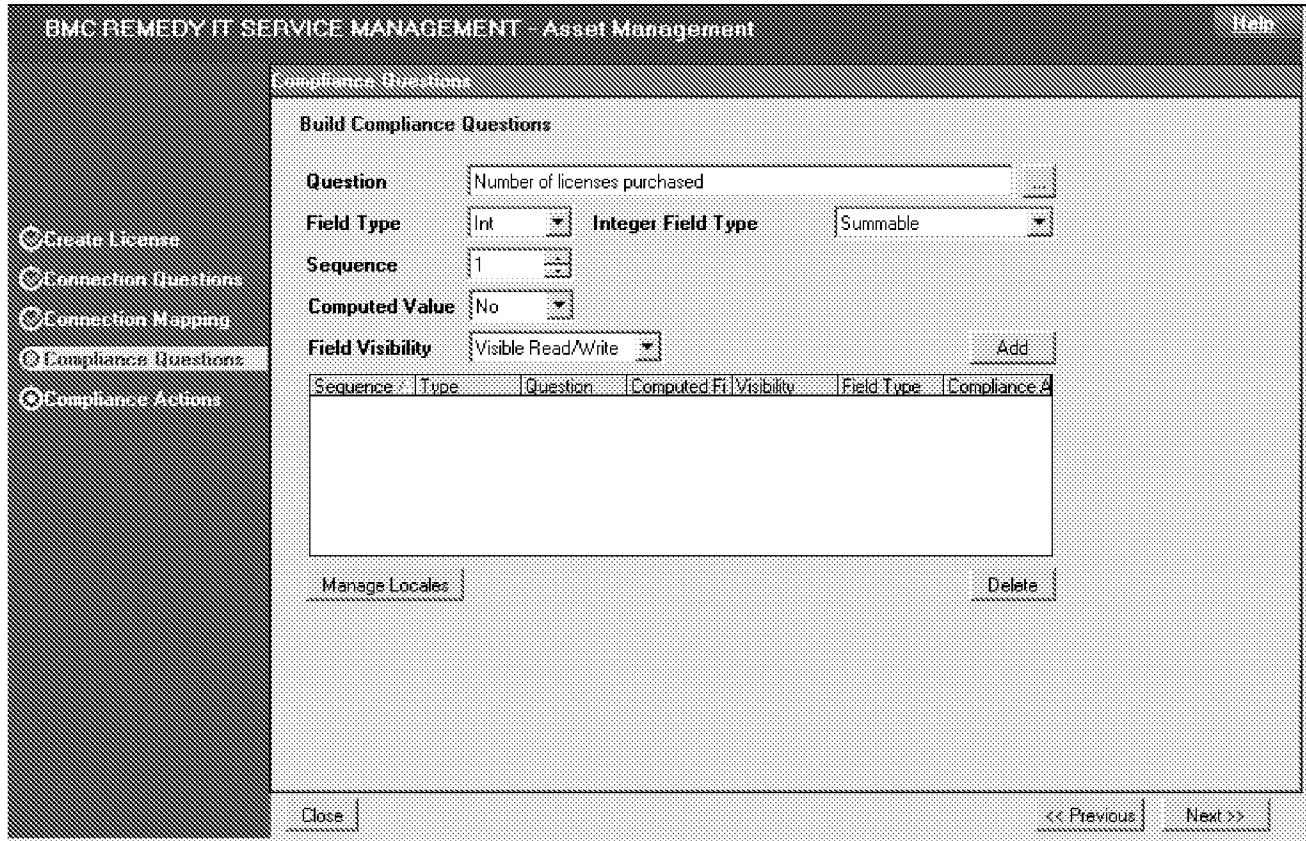
- c For Sequence, select 1.

This indicates the sequence in which the question is asked when the license certificate is created.

- d Since this is a value that cannot be computed, select No for Computed Value.
 - e For the Field Visibility, select Visible Read/Write.
- Since this field is being update by a value entered when the license certificate is created, the field visibility needs to be set to read / write access.
- f For the Integer Field Type field, select Summable.

Figure B-15 shows how the Compliance Questions page should look.

Figure B-15: Build Compliance Questions page filled out



- g Click Add.
- The first question is added to the list.
- 8 On the Compliance Questions page that appears, build the second compliance question as follows:
 - a Enter the question. For this example, you would enter Number of licenses deployed.
 - b For Field Type, select Int.

Since this is also a groupable license type, the Integer Field Type field appears.

 - c For Sequence, select 2.

This indicates the sequence in which the question is asked when the license certificate is created.

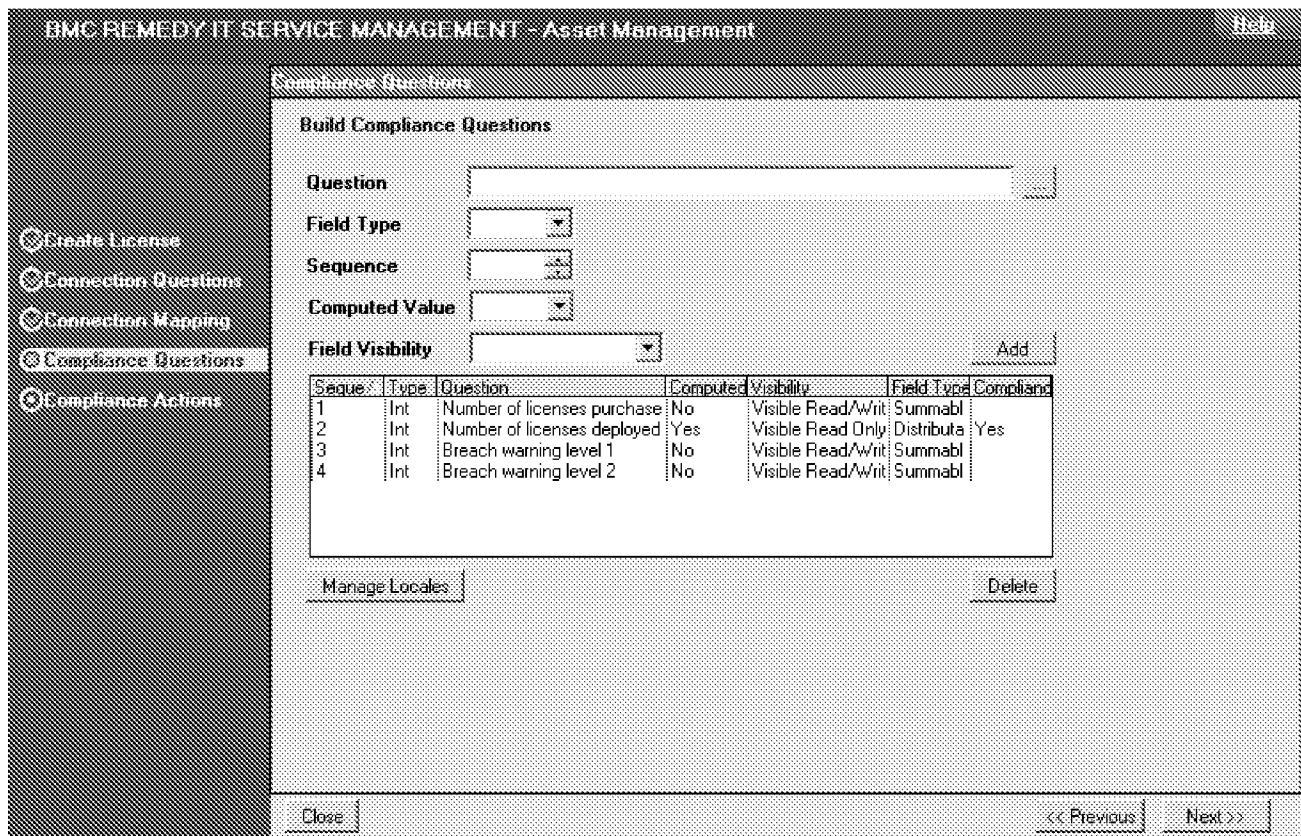
- d Since this is a value that must be derived, select **Yes** for **Computed Value**.
- e For the **Field Visibility**, select **Visible Read Only**.
Since this field is being computed, the field visibility needs to be set to read only access.
- f For the **Integer Field Type** field, select **Distributable**.
The **Summable Field List** and **Set Compliance Alarm** fields appear.
- g From the **Summable Field List** list, select **Number of licenses purchased** as shown in Figure B-16.

Figure B-16: Summable Field List selection

- h From the **Set Compliance Alarm** list, select **Yes**.
If the calculation for this field is not compliant, the alarm has to be set to go off.
 - i Click **Add**.
The second question is added to the list.
- 9 On the **Compliance Questions** page that appears, build the third compliance question as follows:
- a Enter the question. For this example, you would enter **Breach warning level 1**.
 - b For **Field Type**, select **Int**.
Since this is also a groupable license type, the **Integer Field Type** field appears.
 - c For **Sequence**, select **3**.
This indicates the sequence in which the question is asked when the license certificate is created.
 - d Since this is a value that cannot be computed, select **No** for **Computed Value**.
 - e For the **Field Visibility**, select **Visible Read/Write**.
Since this field is being update by a value entered when the license certificate is created, the field visibility needs to be set to read / write access.
 - f For the **Integer Field Type** field, select **Summable**.
 - g Click **Add**.
The third question is added to the list.
- 10 On the **Compliance Questions** page that appears, build the fourth compliance question as follows:
- a Enter the question. For this example, you would enter **Breach warning level 2**.

- b For Field Type, select Int.
Since this is also a groupable license type, the Integer Field Type field appears.
- c For Sequence, select 4.
This indicates the sequence in which the question is asked when the license certificate is created.
- d Since this is a value that cannot be computed, select No for Computed Value.
- e For the Field Visibility, select Visible Read/Write.
Since this field is being update by a value entered when the license certificate is created, the field visibility needs to be set to read / write access.
- f For the Integer Field Type field, select Summable.
- g Click Add.
The fourth question is added to the list. The list of questions should appear as shown in Figure B-17.

Figure B-17: Four compliance questions added for per instance example



11 Click Next.

In the next section of the example, you will be shown how to create the actions that work with the compliance questions to determine whether or not the list of certificates retrieved by the default connection behavior are compliant. For more about default license type behavior, see “Default behavior for customized license types” on page 418.

12 On the Compliance Actions page of the create license wizard, select Get in the Action field and then click Create.

The Get Action dialog box is displayed.

13 Build the Get action:

- a Enter the name of the Get action. For this example, you would use `Get number of licenses deployed`, because this action is retrieving the number of deployed licenses from the list of certificates returned by the default connection behavior (since no connection questions and mappings were configured for this example).
- b Select AR as the source from which to retrieve the information.
- c In the Target field, enter the form name from which to retrieve the information. In this example we are need to retrieve the information from `CTR:Contract_Relationship`, which holds all connection information stored for processed connection questions.

TIP

You configure a license type with the conditions that will be present at the time the license certificate is created. The form `CTR:Contract_Relationship` will be populated with the information that this Get action needs at that time.

- d For this license type we want to count the number of deployed licenses so select Count as the type.
- e From the Rule list, select `GetCertificates > instanceId`, as shown in Figure B-6, to start the rule count to retrieve the number of license certificates.
- f From the Operator list, select `=`.
- g From the Value list, select `Child Instance ID*`.
- h The WHERE field now contains `GetCertificates.instanceId = 'Child_Instance_ID'` as shown in Figure B-18.

Figure B-18: WHERE field updated with built WHERE qualifier

Rule Operator Field Value

WHERE `GetCertificates.instanceId = 'Child_Instance_ID'`

i Click Save.

The Compliance Actions page of the license wizard is redisplayed and the Get action appears in the list.

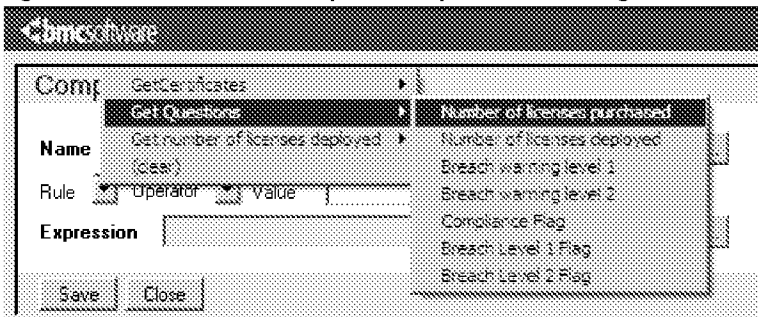
The next steps create the compare action to determine if the number of deployed licenses is compliant based on the number purchased.

14 From the Action list, select Compare and click Create.

15 Build the action that will compare the number of licenses deployed with the number purchased, as follows:

- a Enter the name. For this example, you would enter Check if compliant.
- b From the Rule list, select Get Questions > Number of licenses purchased as shown in Figure B-19.

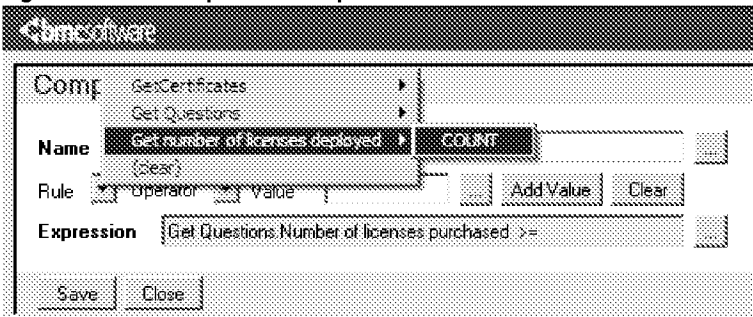
Figure B-19: Get number of products purchased being selected to build compare



c From the Operator list, select the >= sign.

d From the Rule drop-down, select Get number of licenses deployed > COUNT as shown in Figure B-20.

Figure B-20: Completed compare



The completed expression should look like:

Get Questions.Number of licenses purchased >= Get number of licenses deployed.COUNT

e Click Save.

The next set of steps calculate how many licenses are left based on the number purchased and the number deployed.

16 From the Action list, select Calculate and click Create.

- 17 Build the calculate action as follows:
 - a Enter the name. For this example, you would enter Check licenses left.
 - b From the Rule list, select Get Questions > Number of licenses purchased as shown in Figure B-19.
 - c From the Operator list, select the - (minus) sign.
 - d From the Rule drop-down, select Get number of licenses deployed > COUNT.

The completed expression should look like:

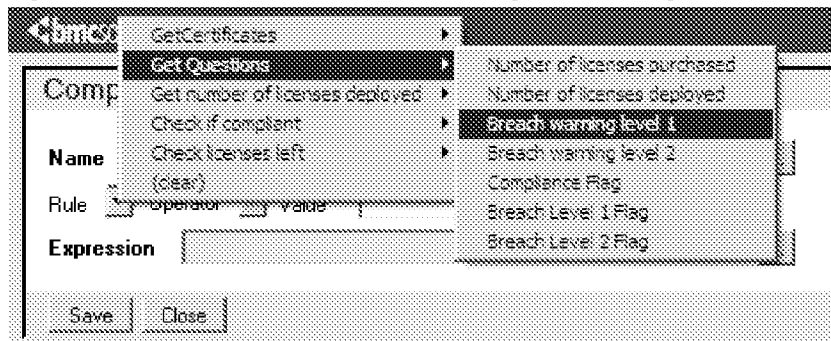
```
Get Questions.Number of licenses purchased - Get number of
licenses deployed.COUNT
```

- e Click Save.

Next, two breach warning flags are configured as compare actions to warn an operator when licenses are approaching non-compliance.

- 18 From the Action list, select Compare and click Create.
- 19 Build the compare action to set breach warning 1, as follows:
 - a Enter the name. For this example, you would enter Check if approaching breach warning level 1.
 - b From the Rule list, select Get Questions > Breach warning level 1 as shown in Figure B-21.

Figure B-21: Get Questions>Breach warning level 1 being selected



- c From the Operator list, select >=.
- d From the Rule list, select Check licenses left > RESULT.

The completed expression should look like:

```
Get Questions.Breach warning level 1 >= Check licenses
left.RESULT
```

- 20 Build the compare action to set breach warning 2, as follows:
 - a Enter the name. For this example, you would enter Check if approaching breach warning level 2.

- b From the Rule list, select Get Questions > Breach warning level 2.
- c From the Operator list, select >=.
- d From the Rule list, select Check licenses left > RESULT.

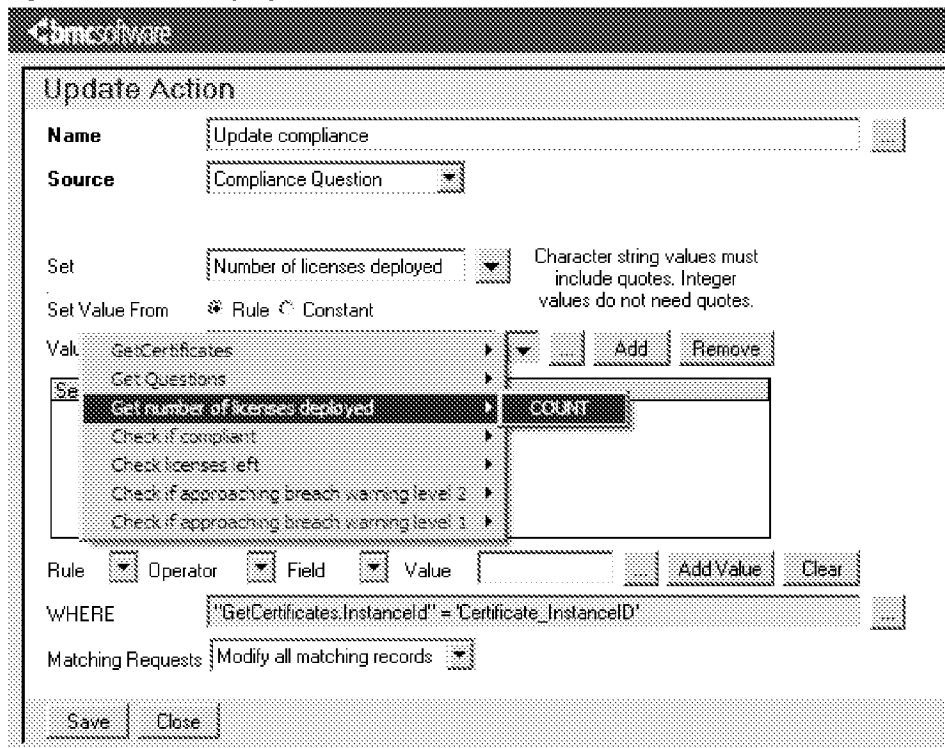
The completed expression should look like:

```
Get Questions.Breach warning level 2 >= Check licenses left.RESULT
```

In order to display the results of the Get action and set the breach level and compliance flags, you must build an update action.

- 21 From the Action list, select Update and click Create.
- 22 Build the update action as follows:
 - a Enter the name. For this example, you would enter Update compliance.
 - b For the Source select Compliance Question.
 - c From the Set list, select Number of products deployed. This value indicates that the question field should be updated based on the count retrieved in the Get.
 - d In the Set Value From field, select Rule.
 - e From the Value list, select Get number of licenses deployed > COUNT as shown in Figure B-22.

Figure B-22: GetDeployed>COUNT selected for Value



- f Click Add.

- 23 Create a second update action in which you set the compliance flag to return a true or false to indicate whether the licenses are compliant. Leave the Update dialog box open and complete the rest of the steps.
 - a For the Source select Compliance Question.
 - b From the Set list, select ComplianceFlag.
 - c In the Set Value From field, select Rule.
 - d In the Value field, select Check if compliant > RESULT.

This value indicates that the compliance flag should be set based on the result from the Check if compliant compare configured earlier.
 - e Click Add.

Both update actions appear in the list.
- 24 Create a third update action in which you set the breach level 1 flag. Leave the Update dialog box open and complete the rest of the steps.
 - a For the Source select Compliance Question.
 - b From the Set list, select Breach Level 1 Flag.
 - c In the Set Value From field, select Rule.
 - d In the Value field, select Check licenses left > RESULT.

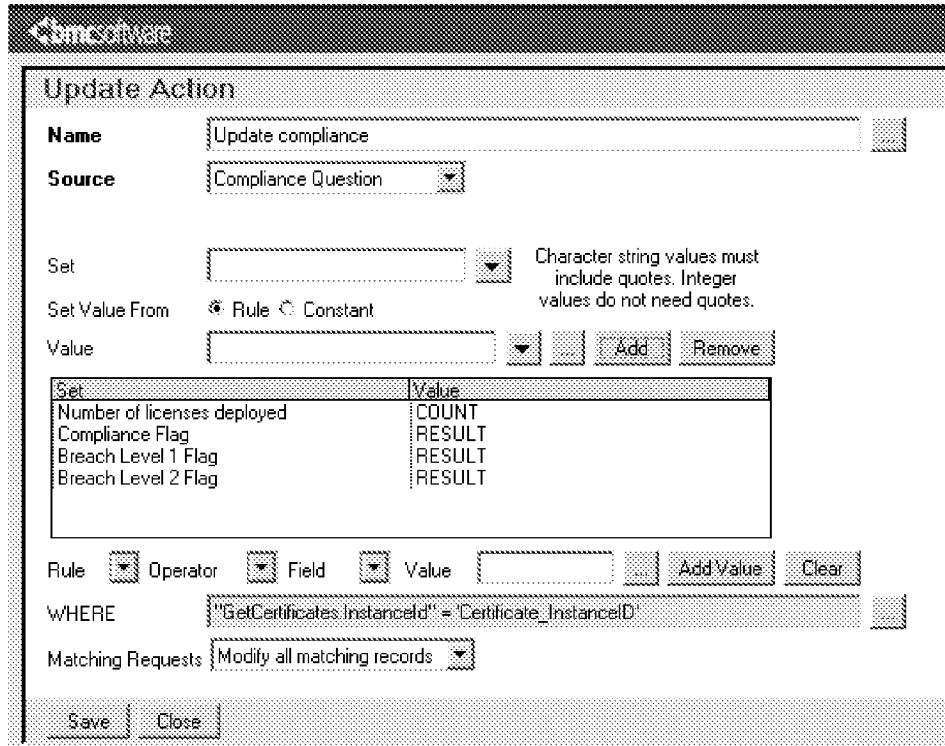
This value indicates that the Breach level 1 flag should be set based on the result of the Check licenses left calculate configured earlier.
 - e Click Add.

Three update actions appear in the list.
- 25 Create a fourth update action in which you set the breach level 2 flag. Leave the Update dialog box open and complete the rest of the steps.
 - a For the Source select Compliance Question.
 - b From the Set list, select Breach Level 2 Flag.
 - c In the Set Value From field, select Rule.
 - d In the Value field, select Check licenses left > RESULT.

This value indicates that the Breach level 2 flag should be set based on the result of the Check licenses left calculate configured earlier.
 - e Click Add.

Four update actions should appear in the list as shown in Figure B-23.

Figure B-23: Update actions created for per instance example



Since Compliance Question was selected for Source, the following default WHERE qualifier is supplied:

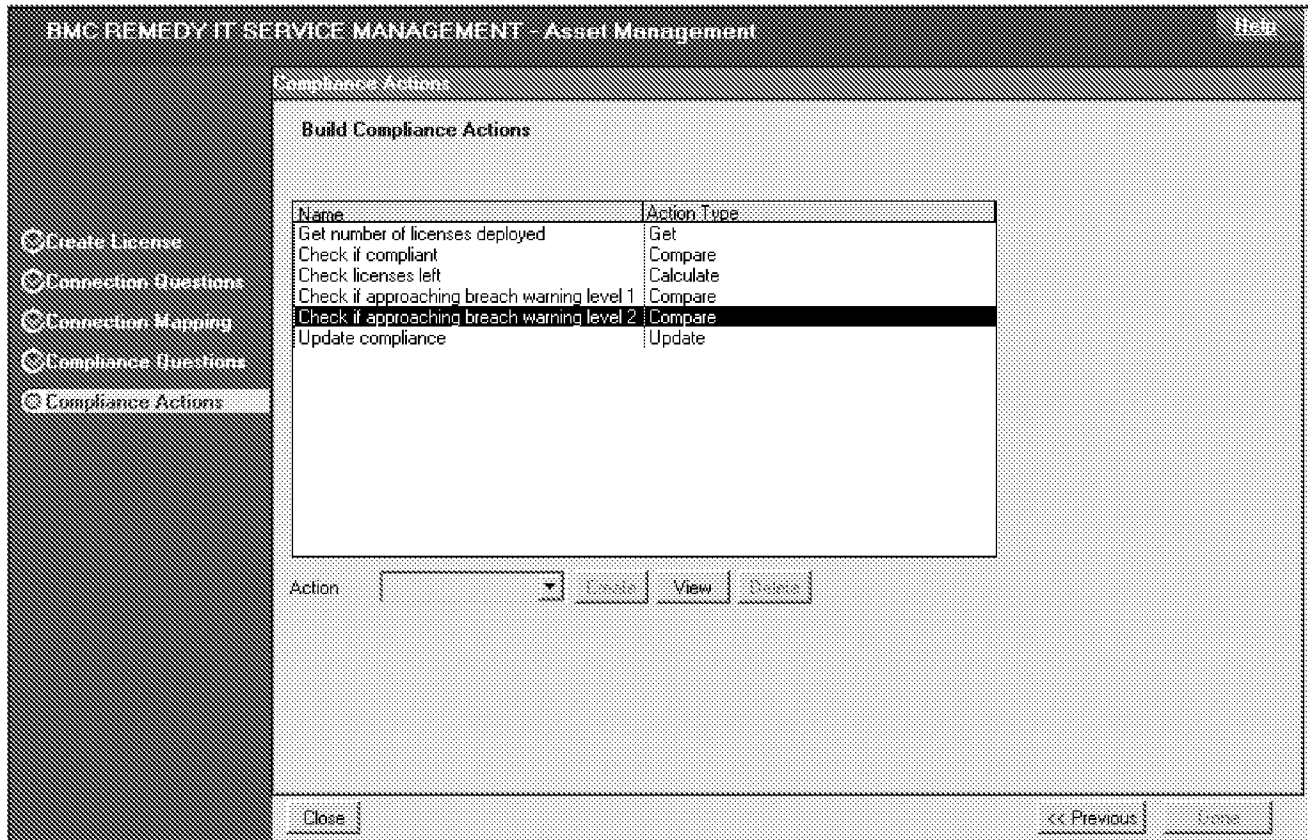
"GetCertificates.InstanceId" = 'Certificate_InstanceID'

For this example, this WHERE qualifier is sufficient and does not need to be modified.

26 Click Save.

The Compliance Actions page of the license wizard is redisplayed and all of the compliance actions appear in the list as shown in Figure B-24.

Figure B-24: Compliance Actions page showing all of the actions created

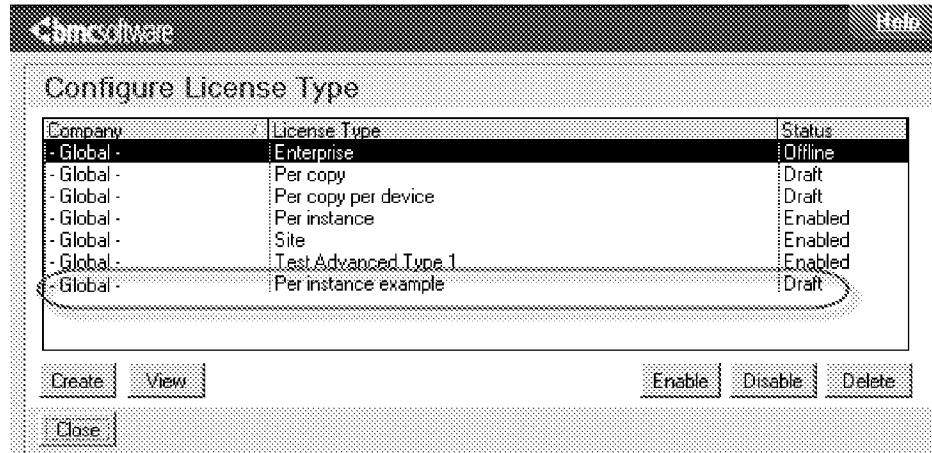


27 Click Done.

28 At the prompt, click Yes to create the license type.

The license type is created and displayed in the list in the Configure License Type dialog box, as shown in Figure B-25.

Figure B-25: Configure License Type dialog box showing the new license type



By default the license type is created in draft mode.

- 29 To enable the license type, select it and click Enable.
- 30 Click Close.

BMC Remedy AR System fixed and floating license type example—advanced mode

The BMC Remedy AR System fixed and floating default license type is an example of a license type created in advanced mode. To understand more about how advanced license types are created, you can view the different parts of this license type.

- ▶ **To view the BMC Remedy AR System fixed and floating license type**
 - 1 From the Application Administration Console, click the Custom Configuration tab.
 - 2 From the Application Settings list, choose Asset Management > Advanced Options > Configure License Type, and then click Open.
The Configure License Type dialog box appears.
 - 3 Select the BMC Remedy AR System fixed and floating default license type and click View.
The license type is displayed in the license type wizard.
 - 4 Click Next to view the different parts of the license type.

7 Software license management

You can use the software license management feature in BMC Remedy Asset Management to facilitate, focus, and follow-through on compliance. This section describes how to use BMC Remedy Asset Management to manage software licenses and their compliance within your organization.

The following topics are provided:

- About software asset management and software license management (page 150)
- About the Software Asset Management console (page 157)
- Creating a software contract (page 161)
- Reviewing a software contract (page 163)
- Adding a license certificate to a software contract (page 163)
- About certificate groups (page 166)
- Manually managing certificate groups (page 166)
- Managing jobs that automatically attach CIs to license certificates (page 167)
- Reviewing a software license certificate (page 171)
- Relating one license certificate to another license certificate (page 172)
- Determining which CIs use a license certificate (page 173)
- Manually managing CIs attached to a license certificate (page 174)
- Manually specifying a license certificate for a software CI (page 176)
- Recording the purchase cost for a license (page 177)
- Troubleshooting software license management (page 177)

About software asset management and software license management

Organizations can acquire software in different manners. Software can be built for a specific purpose within the company. Software can be purchased from a software vendor or outsourcer. Software can be acquired through an acquisition or merger between companies, or between departments within a single company.

Software asset management is a core component of an overall asset management policy. IT Infrastructure Library (ITIL) in the *Software Asset Management Book* defines software asset management as “all of the infrastructure and processes necessary for the effective management, control and protection of the software assets within an organization, throughout all stages of their lifecycle.”

ITIL indicates that the following processes make up the holistic approach to software asset management:

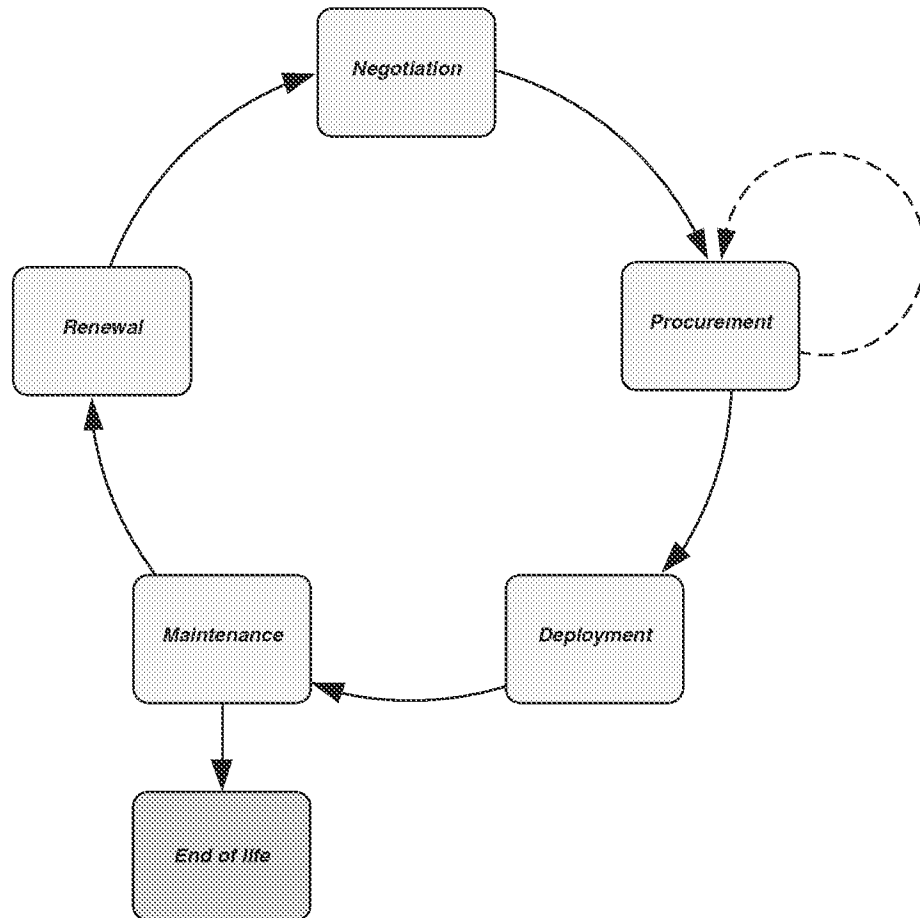
- **Overall management processes**—The management processes surrounding the other software assets management processes. The overall management processes are related to change management.
- **Core asset management processes**—Identification of software assets, including maintaining this information in the configuration management database (CMDB).
- **Logistic processes**—Control of the software asset lifecycle. These processes include procurement, deployment, and end of life.
- **Verification and compliance processes**—Verification and compliance of software asset management policies and procedures, including license compliance.
- **Relationship management processes**—Software contract management.

BMC provides solutions in each of these processes. This section focuses on the key processes for verification and compliance (license compliance), relationship processes (contract management), and logistic processes (deployment) in the context of the solution that BMC provides for software license management.

About the software lifecycle

The software lifecycle comprises stages for negotiation, procurement, deployment, maintenance, renewal, and end of life, as illustrated in Figure 7-1.

Figure 7-1: Software lifecycle



Negotiation

When you plan to procure software from another company, one of the first steps is to negotiate a software license contract with the vendor. If you have BMC IT Business Management suite, you can use the Vendor Relationships Management module during the negotiation stage.

In BMC Remedy Asset Management, you can use the Contract Management console to track the different types of contracts for each vendor, including software contracts. You can use the Software Contract form to track the terms of the contract, the cost of the contract, and the individual certificates that represent the license purchased. The Software Contract form provides links to extended information about the contract, such as the purchase order, the Digital Media Library (DML), and the deployed software configuration items (CIs) in BMC Atrium Configuration Management Database (BMC Atrium CMDB).

Procurement

You can use BMC Remedy Asset Management to generate a purchase order for the software, or you can use other procurement software. You can relate the purchase order for the software to the software license certificates, so that you can track the financial information regarding software procurement.

Deployment

When the software arrives, several procedures must be performed to deploy the software, as indicated in Table 7-1.

Table 7-1: Procedures for deploying software

Procedure	Description
Managing the deployment process	BMC provides several mechanisms to manage the deployment of the software. You can use BMC Remedy Change Management to manage the deployment of software into the IT infrastructure as described in the <i>BMC Remedy Change Management User's Guide</i> . BMC Remedy Change Management tracks the tasks involved in deploying the software, and can help you understand the risks or dependencies on the software. BMC also provides solutions to manage the actual deployment.
Deploying the software to existing systems in the IT infrastructure	BMC Configuration Automation for Clients can manage the deployment of software to existing desktops. It uses policies to enforce that the software is deployed only to the systems entitled to have the software.
Managing the "bare metal" provisioning of new systems	You can use BMC BladeLogic and BMC Atrium Orchestrator to manage the bare metal provisioning of systems. BMC Atrium Orchestrator integrates with BMC Remedy Change Management to take information about the systems that are being deployed. BMC Atrium Orchestrator works with BMC BladeLogic and with BMC Configuration Automation for Clients to deploy the appropriate software packages to the system.

Table 7-1: Procedures for deploying software (Continued)

Procedure	Description
<p>Accurately populating data into BMC Atrium CMDB</p>	<p>After the software is deployed, BMC discovery products, such as BMC Configuration Automation for Clients and BMC Atrium Discovery and Dependency Mapping (BMC Atrium Discovery), can find the software deployed on the IT infrastructure and populate the data into BMC Atrium CMDB. For information, see the <i>BMC Configuration Automation for Clients Configuration Discovery Integration for CMDB Implementation Guide</i> and the <i>BMC Atrium Discovery and Dependency Mapping: Populating BMC Atrium CMDB</i>.</p> <p>As part of this process, data is normalized and reconciled, as described in the <i>BMC Atrium CMDB Normalization and Reconciliation Guide</i>.</p>
<p>Accurately connecting the software represented in BMC Atrium CMDB to the software license certificate that represents the license agreement for that software</p>	<p>BMC Remedy Asset Management provides dynamic mechanisms to connect the software that has been deployed and represented in BMC Atrium CMDB. License certificates provide a definition of how to find the software in BMC Atrium CMDB and leverage information from the DML, so that the searches are performed using a consistent, normalized set of data. The license engine uses this information to query BMC Atrium CMDB to find the new or updated software that has been deployed, and to connect that software to the appropriate license certificates. If deployed software cannot be related to a certificate, it is treated as an exception for the software asset manager to evaluate and determine the appropriate action.</p>

Maintenance

Maintenance is an ongoing activity. BMC Remedy Asset Management provides a way for you to track the ongoing license compliance for the software. Tracking compliance is rule-based and can vary based on the license agreement for the specific software and specific vendor.

To manage the health of the software, you can use the following BMC products:

- ❖ **BMC Remedy Service Desk**—Manage incident requests, problem investigations, and known errors related to the software.
- ❖ **BMC monitoring software**—Monitor application and server performance using programs such as BMC ProactiveNet Analytics, BMC Performance Manager, and BMC Transaction Management.
- ❖ **BMC Service Level Management**—Manage service level agreements related to the software.
- ❖ **BMC Service Impact Manager**—Track the impact to the company if the software has issues.

You can use BMC Configuration Automation for Clients to track and understand the usage of software. By understanding usage, you can proactively maintain the deployment of software to allow for the most effective use of the purchased software licenses.

Renewal

When software contracts are nearing expiration, BMC Remedy Asset Management can send notification. BMC Remedy Asset Management provides processes for renewing contracts and for tracking the additional purchase of licenses. The renewal process feeds back into the negotiation process, providing a closed loop vision of the software lifecycle.

End of life

If software is being put through an end-of-life process, you can use BMC Remedy Asset Management to help determine where the software is deployed, which can help you decide whether to upgrade the software to newer or other versions.

Implementing software license management

The contract manager and software asset manager create software contracts, add license certificates, and relate software CIs to license certificates.

The power of software license management comes when you use the License Engine to automate the process. When you add a license certificate, you specify the license type (such as a per instance or site license) and specify details required for that license type. Each license type provides a set of connection rules, which the License Engine uses to query BMC Atrium Configuration Management Database (BMC Atrium CMDB) and select the appropriate CIs to connect to the license. Each license type also provides a set of compliance rules, which the License Engine uses to calculate whether the license is in compliance. The software asset manager schedules license jobs, so that the License Engine regularly connects software CIs to license certificates and checks for compliance.

You are not restricted to the license types that come with BMC Remedy Asset Management. An application administrator can create new license types, as described in the *BMC Remedy IT Service Management Configuration Guide*. The application administrator can create sophisticated queries and calculations based on the data in BMC Atrium CMDB. For example, the Per Copy license type, which comes with BMC Remedy Asset Management, calculates the number of users for a software product by looking at the number of users (stored as BMC_Person) with a dependency on the computer system of which the software CI is a component. For compliance calculations, you can get data from an BMC Remedy AR System form, in addition to data stored in BMC Atrium CMDB.

The software license management lifecycle, as described in “About the software lifecycle” on page 151, is implemented as described in Table 7-2.

Table 7-2: Software license management lifecycle

Stage	What happens	Described in
Negotiation	The contract manager creates the software contract with a draft status.	“Creating a software contract” on page 161
	If the contract requires a new license type, the application administrator creates the new license type.	<i>BMC Remedy IT Service Management Configuration Guide</i>
Procurement	The contract manager changes the status of the software contract to executed.	“Reviewing a software contract” on page 163
	The configuration administrator requisitions and purchases software.	“Creating purchase requisitions” on page 105 and “Working with purchase orders” on page 118
	The software asset manager adds license certificates for purchased software.	“Adding a license certificate to a software contract” on page 163
Deployment	The software asset manager creates and schedules a license job to connect CIs to the license certificate and to check compliance of the license.	“Managing jobs that automatically attach CIs to license certificates” on page 167
	The configuration administrator receives the software.	“Receiving and returning purchase items” on page 124

Table 7-2: Software license management lifecycle (Continued)

Stage	What happens	Described in
Maintenance	Discovery products, such as BMC Configuration Automation for Clients or BMC Atrium Discovery, discover the software. The discovery products populate BMC Atrium CMDB with the CI for the software and with the relationship between software and the computer system on which it is installed.	<i>BMC Configuration Automation for Clients Configuration Discovery Integration for CMDB Implementation Guide</i> and <i>BMC Atrium Discovery and Dependency Mapping: Populating BMC Atrium CMDB</i>
	The BMC Atrium Reconciliation Engine runs, populating the production dataset (BMC Asset) with the discovered data.	<i>BMC Atrium CMDB Normalization and Reconciliation Guide</i>
	The License Engine runs the license job to connect CIs to the license certificate and to check compliance of the license. The software asset manager can check the history from the Manage License Jobs console.	"About the Manage License Jobs console" on page 168
	The software asset manager monitors the status of software license certificates.	"About the Software Asset Management console" on page 157 and "Reviewing a software license certificate" on page 171
	To prepare for an audit, the software asset manager runs a license job immediately, to check for compliance, and then prints a report.	"Running a job immediately" on page 170 and "Table 11-1 describes the predefined reports included with BMC Remedy Asset Management, organized by the type of report." on page 241
Renewal and End of Life	When the software license expires, the next time that the License Engine runs a connection and compliance job, it removes the software CIs from the expired license. If there is another non-expired license, the License Engine attaches the software CIs to the license. Otherwise, the software CIs are unlicensed.	Not applicable
	If the license is renewed, the software asset manager opens the license certificate and renews the license. The next time that the License Engine runs a connection and compliance job, it attaches the software CIs to the renewed license certificate.	"Reviewing a software license certificate" on page 171

Software license management and multi-tenancy

Each software contract and software license applies to the company that you specify on the software contract. The License Engine connects CIs only for that company to the software license. This means that you must specify the Company field in the CI, or configure your discovery product to specify the Company field.

About the Software Asset Management console



The Software Asset Management console is the primary console for software asset managers. Anyone who manages software licensing and has any of the following permissions can use this console:

- Asset Admin
- Asset User
- Contract User
- Contract Admin

Functional areas of the Software Asset Management console

Figure 7-2 on page 158 illustrates the functional areas of the Software Asset Management console.

Figure 7-2: Functional areas of the Software Asset Management console

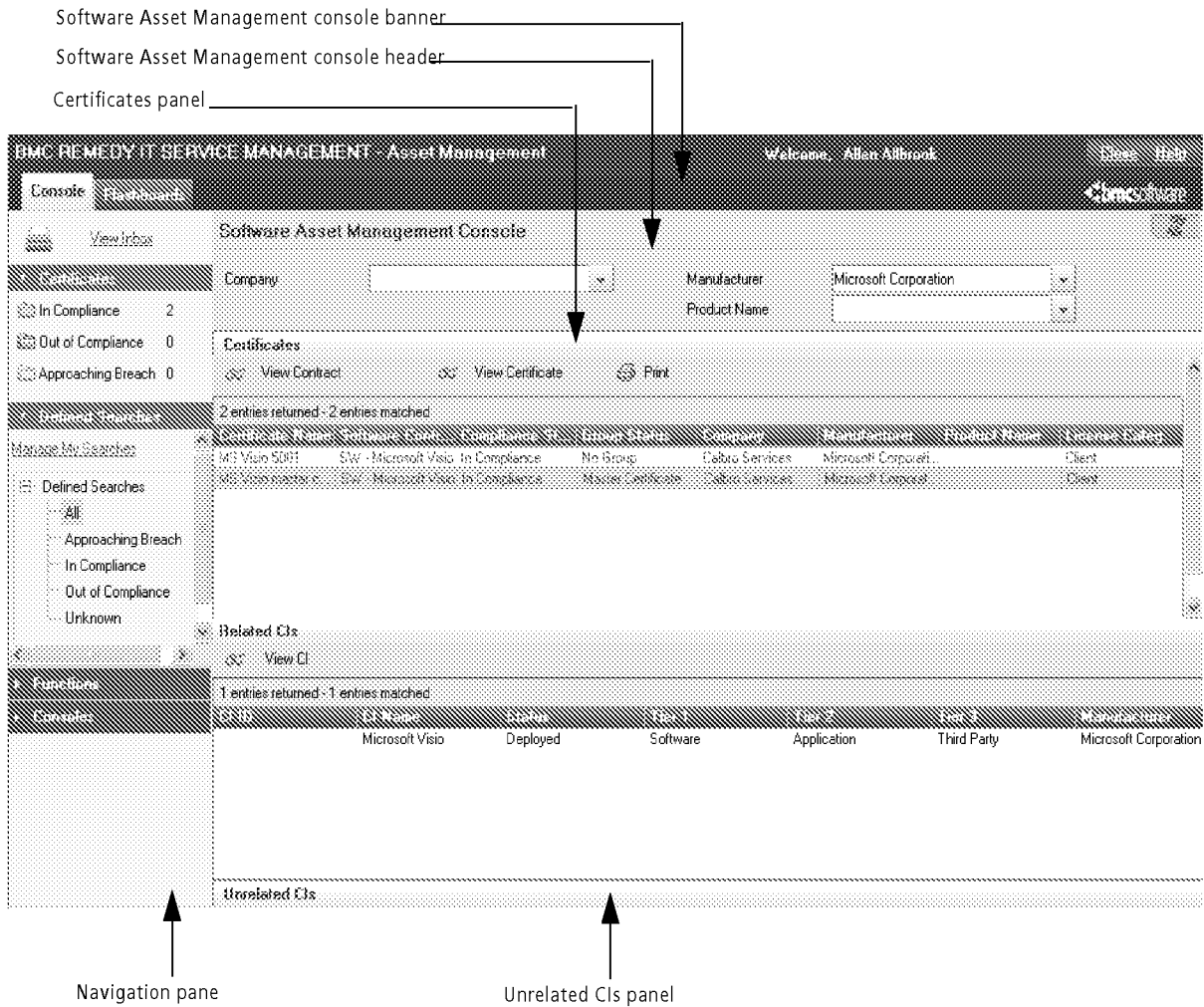


Table 7-3 describes what you can do in each of the functional areas.

Table 7-3: Software Asset Management console functional areas

Functional area	Description
Software Asset Management console banner	
Console tab	If the Flashboards tab is open, click the Console tab to return to the console. Use this tab to perform tasks from the Software Asset Management console.
Flashboards tab	Click the Flashboards tab to select a dashboard. The dashboards that appear in the Software Asset Management console represent, in graphical format: <ul style="list-style-type: none"> ⌘ Certificates by compliance status ⌘ Certificates by manufacturer ⌘ Certificates by supplier
Software Asset Management console header	
Company, Manufacturer, and Product Name	You can filter the software license certificates listed in the certificates table by any company, by manufacturer, by product name, or by a combination. For example, you can list all Microsoft Visio licenses for your company.

Table 7-3: Software Asset Management console functional areas (Continued)

Functional area	Description
Refresh	Refreshes the data in the tables.
Navigation pane	
Inbox	<p>The inbox displays events generated when the License Engine runs, and other messages about contract and software compliance information.</p> <p>The inbox displays messages for the following areas:</p> <ul style="list-style-type: none"> ※ Contracts that have expired or are about to expire. ※ Software license certificates that are not compliant or are approaching non-compliance. ※ Software license certificates that have expired or are about to expire. ※ Summary results of the engine run. ※ List of new CIs that are linked to software license certificates. <p>Note: You might see different types of messages displayed, depending on how the application administrator has configured inbox preferences.</p> <p>From the inbox, you can indicate whether you have dealt with a message. To view the applicable contract or certificate, select the message and click View.</p>
Certificates summary	<p>A summary of certificate software license compliance with counts of how many certificates are in compliance, out of compliance, or approaching breach.</p> <p>CI counts reflect the filters—your selections for Company, Manufacturer, and Product Name. Using search definitions does <i>not</i> change the certificate counts.</p>
Search Definitions	<p>Use the defined searches to search for certificates that are approaching breach, in compliance, out of compliance, or of unknown compliance. Use the Manage My Searches link to define additional searches.</p>
Functions	<p>The Software Asset Management console provides access to the following functionality:</p> <ul style="list-style-type: none"> ※ Managing software license jobs. For more information, see “Manually managing CIs attached to a license certificate” on page 174. ※ Managing CIs. For more information, see “Working with configuration items” on page 51. ※ Additional functions, as described in “Secondary functions on the Software Asset Management console” on page 160.
Consoles	<p>Depending on your permissions and what other applications are installed, use these links to open:</p> <ul style="list-style-type: none"> ※ Asset Management console ※ Change Management console ※ Contract Management console ※ Incident Management console ※ Overview console ※ Problem Management console ※ Release Management console ※ BMC IT Business Management

Table 7-3: Software Asset Management console functional areas (Continued)

Functional area	Description
Certificates panel	
Note: If you cannot see the Certificates panel, click the Certificates title.	
View Contracts	To view details of a contract, select the certificate and click View Contract. For more information about viewing contracts, see “Reviewing a software contract” on page 163.
View Certificate	View details of the selected certificate. For more information, see “Adding a license certificate to a software contract” on page 163.
Print	Print the selected contract.
Certificates table	Lists software license certificates. The Compliance Status column indicates whether the license is in compliance.
Select	Select whether to view related contracts or related CIs.
Related CIs	To view which CIs are related to a certificates, select the certificate. The Related CIs table lists the related CIs. To view details of one of the CIs, select the CI and click View CI. For information about manually managing CIs attached to a certificate, see “Manually managing CIs attached to a license certificate” on page 174.
Unrelated CIs panel	
Note: If you cannot see the Unrelated CIs panel, click the Unrelated CIs title.	
View CI	Select a CI and click View CI to view details about the CI. For more information, see “Working with configuration items” on page 51.
Job Name	When running connection rules for a job, if the License Engine cannot link a CI or a list of CIs to contracts, the engine creates an exception record for each CI that it cannot relate. Select a job to see the unrelated CIs for that job.
CIs table	The CIs table lists the exception records for the most recent run of the selected job. These records are software CIs that should be related to a certificate, but are not. Note: A CI that is unrelated on one job might be related as the result of a different subsequent job.

Secondary functions on the Software Asset Management console

This console includes links to perform the following functions that are not related to either contract or software license management:

- Open other consoles, such as the Purchasing console and the Reporting console.
- Use the BMC Atrium CMDB query dialog box to build complex searches, as described in “Using the BMC Atrium CMDB query dialog box to search for CIs” on page 35.
- Create reminders, as described in “Creating reminders” on page 43.
- View your profile.

Creating a software contract

You can track compliance and usage of software license contracts. Unlike most other contracts, however, you do not relate the software CI directly to the contract. For software contracts, you add software license certificates to the contract, and relate the software CIs to the certificates.



For software contracts, you maintain the cost information in the attached certificates. You can view the total cost on the contract.

► **To create a software license contract**

- 1 On the Contract Management console, click Create, and select Software License.

Figure 7-3: Software Contract form displaying License Details tab

The screenshot shows the 'Software Contract' form in the 'License Details' tab. The contract ID is 'SW - Microsoft Visio'. The summary is 'Software Contract for M...'. The term is 'Never ending'. The status is 'Executed'. The license certificates table lists three entries for 'MS Visio' licenses, all with a status of 'Executed' and 'In Compliance'.

- 2 On the Software Contract form, complete all the required fields.

Field name	Description
ID	A unique alphanumeric value.
Summary	Brief description of the contract.
Term	Select the applicable term: <ul style="list-style-type: none"> ⌘ Fixed—Contract expires at the expiration date. ⌘ Never Ending—Contract never expires. ⌘ Rolling Contract—Contract automatically renews at the expiration date.





Field name	Description
Status	<p>Select the applicable status of the contract:</p> <ul style="list-style-type: none"> ※ Draft—The contract has never been executed but is going through the process of being executed. You can use draft status for contracts that are in negotiation or pending signature. ※ Executed—The contract that is executed and valid. Executed contracts include active contracts that have started. An executed contract might have the following status reasons: active, change pending, on hold, requires attention, under renegotiation. ※ Historical—The contract is not valid. Historical contracts include contracts that have expired, been terminated, or been canceled. ※ Delete—The contract is scheduled for deletion.
Company	The company associated with this contract.
View Access	<p>Select who can view or modify this contract:</p> <ul style="list-style-type: none"> ※ Public—Anyone who can access contracts can view or modify the contract. ※ Internal—Only people in the support group managing this contract can view or modify the contract.
Supplier Name	The supplier associated with this contract.
Cost Center	The cost center that owns this contract. This field and the following fields are located on the General tab.
Support Company	The support company associated with this contract.
Support Organization	The support organization associated with this contract.
Notification Group	The notification group associated with this contract.
Notification Contact	Optionally, you can assign an individual to receive notifications for this contract.
Owner Group	The group responsible for this contract.
Owner Contact	Optionally, you can indicate an individual responsible for this contract.
Expiration Date	Date contract expires. When a contract expires, individuals are notified first, then groups.
Notification Date	<p>When the contract expires, the notification contact and the owner contact are notified on this date.</p> <p><i>Note:</i> If individual contacts are not specified on the Contacts tab, the notification group and owner group are notified.</p>



- 3 Save the contract.
- 4 Add license certificates, as described in “Adding a license certificate to a software contract” on page 163.

Reviewing a software contract

The Software Contract form displays the following information:



- Contract details, including the status, and the terms and conditions
- A list of all license certificates for the contract
- The cost rollup from the license certificates

From the Software Contract form, you can perform the following actions:



- Update the status of the contract.
- Record payment for the contract.
- Add license certificates, as described in “Adding a license certificate to a software contract.”.

You can perform high-level reviews of all software contracts from the Software Asset Management console and the Contract Management console. For information about the consoles, see “About the Software Asset Management console” on page 157 and “About the Contract Management console” on page 131.

Adding a license certificate to a software contract



A license certificate indicates the right to deploy software in your environment. Because one contract might have multiple certificates, software compliance is tracked at the software level.

To maintain a history of the purchase, you can link the certificate to a purchase order line item.

When the License Engine runs, it attaches CIs to the certificate, based on the following information about the license certificate:

- Company
- Product manufacturer and name
- Categorization
- Answers to connection questions

► To add a license certificate to a software contract

- 1 On the Contract Management console, select the software license contract, and click View.
- 2 On the Software License Contract form, click the License Details tab, and click Add.

3 On the License Certificate form, complete all the required fields.

Field name	Description
<ul style="list-style-type: none"> ※ Company ※ Software Contract ID 	This information comes from the software contract.
Certificate ID	The certificate ID identifies the license certificate in listings and reports. It does not have to be unique.
Summary	This field provides additional space to describe the certificate.
Status	When you create a license certificate, the status is set to Draft.
License Category Type	Select from Client, Server, or Mainframe.
License Type	<p>Select the appropriate license type. The license type determines the connection questions and the compliance questions. BMC Asset Management comes with the following license types:</p> <ul style="list-style-type: none"> ※ Enterprise—A company-wide site license for the product. ※ Per copy—This license is based on the number of unique people using the license. Unique people are the people related to the computer system on which the product is installed. These people include both people related to the computer system through BMC Remedy ITSM with the “used by” relationship and people who are discovered with dependency on the computer system. ※ Per copy per device—If two copies per device are permitted, each device with either one or two copies of the product consumes one license. In this example, a device with three copies of the product consumes two licenses. ※ Per instance—Each instance of the software requires a separate license. ※ Site—The product is licensed for an entire site within the company. It can be restricted to a region, site group, or site. The site is determined by the site of the computer system on which the product is installed. <p>If your administrator created other license types, select the appropriate license type.</p>
Cost Center	This information comes from the software contract, but can be changed.
Effective Date	Specify the date that the license becomes effective.
Expiration Date	Specify the date that the license expires. If the license does not expire, leave this field blank.

- 4 To link the certificate to a purchase, perform the following steps:
 - a In the Purchase Line Item area, click Search Line Item.
 - b In the Search Purchase Line Items dialog box, search for the purchase line item.
 - c Select the appropriate line item, and click Relate.
- 5 Click Next.

- 6 Select the software from the product dictionary.

You must select the manufacturer. Optionally, you can select the product name, the version, or the categorization.

If the same type of certificate exists, you are asked whether to group the certificates. You are prompted to group only certificates that have the same product categorization.

TIP



If you do not need to track license certificates separately, BMC recommends that you group them. If you must track license certificates separately, however, do not group them. For example, if each department pays separately for their own Microsoft Visio licenses, do not group them. For information about certificate groups, see “About certificate groups” on page 166.

- 7 If you are prompted to group the certificate, perform the following steps to add it to a group:
- a Click Manage Grouping.
 - b Search for and select the master certificate.

If there is no appropriate master certificate, you can create a master certificate.
 - c Click Add to Group.
 - d After you finish grouping certificates, click Next.
- 8 Provide connection details and compliance details.

Detail type	Description
Connection details	To determine which CIs use the license certificate, the License Engine uses the information that you provide about connection details. These details provide information about how to connect the license certificate to the appropriate CI.
Compliance details	To determine compliance, the License Engine uses the information that you provide about compliance details. For example, a Per Instance license certificate asks you how many licenses were purchased and displays how many licenses were deployed.

NOTE

For some license types, a dialog box prompts you for connection details and compliance details. Use the Save button to save the details and continue.

- 9 On the certificate, click Save.

About certificate groups



Certificate groups consolidate the tracking of license certificates. A master certificate is grouped with individual child license certificates. The CIs are attached to the master certificates. License allocation numbers are attached to the child license certificates.

For example, under the same software contract, you might buy 200 licenses for Microsoft Word. Later, you might buy 100 more licenses. In this example, it does not matter which CI is attached to a specific license certificate. For compliance, it only matters that you do not exceed 300 Microsoft Word instances for the contract.



By grouping license certificates, you gain flexibility in how the license certificates are applied. On the master certificate, you modify the sequence to which license certificates are allocated to matching CIs. When the first license in the sequence is fully used, the License Engine starts applying CIs to the next license. As a result, only the last certificate can be out of compliance.

TIP

If you have multiple contracts with different costs for being out of compliance, make sure that the most expensive certificate is allocated first, because only the last certificate can become out of compliance.

Certificate groups help you avoid unnecessary warnings. Consider the preceding Microsoft Word license example. If you do not group the license certificates, you might receive a warning when 190 CIs are attached to the first license certificate. Although you have another license certificate that is valid for 100 instances, the first certificate would be approaching the maximum usage. If, however, you group the certificates, for compliance checks, it is equivalent to having one certificate for 300 instances. You receive a warning only when the last certificate in the sequence approaches being completely allocated.

When a certificate expires, the License Engine checks for compliance. If you have enough licenses remaining in the group, you do not receive a warning. If a license certificate is not part of a group, when it expires, all the related CIs are out of compliance.

Manually managing certificate groups



You can manually add or remove a certificate from a certificate group.

When you remove a certificate from a group, the CIs remain attached to the master certificate; when you run a licence job, CIs might be attached to the ungrouped certificate. If the group contained two certificates only, when you remove a certificate from the group, the master certificate is removed, because you cannot have a group of only one certificate.

► **To remove a certificate from a group**

- 1 From the Software Asset Management console, open the certificate.
- 2 In the navigation pane, choose Functions > Unrelate From Group.
A message prompts you to confirm that you want to unrelate the certificate from the group.
- 3 To unrelate the certificate from the group, click Yes.

► **To manually add a certificate to a certificate group**

- 1 From the Software Asset Management console, open the certificate.
- 2 In the navigation pane, choose Functions > Manage Grouping.
The Group Certificates dialog box displays certificate groups and ungrouped certificates that can be grouped with the open certificate.
- 3 Select the appropriate certificate or certificate group, and click Select Certificate, and then click Close.

Managing jobs that automatically attach CIs to license certificates



The License Engine automatically connects CIs to license certificates, based on company, product information, and answers to connection questions. It also calculates compliance based on answers to compliance questions.

You can schedule the License Engine to run immediately, at a specific time, or after reconciliation.

NOTE

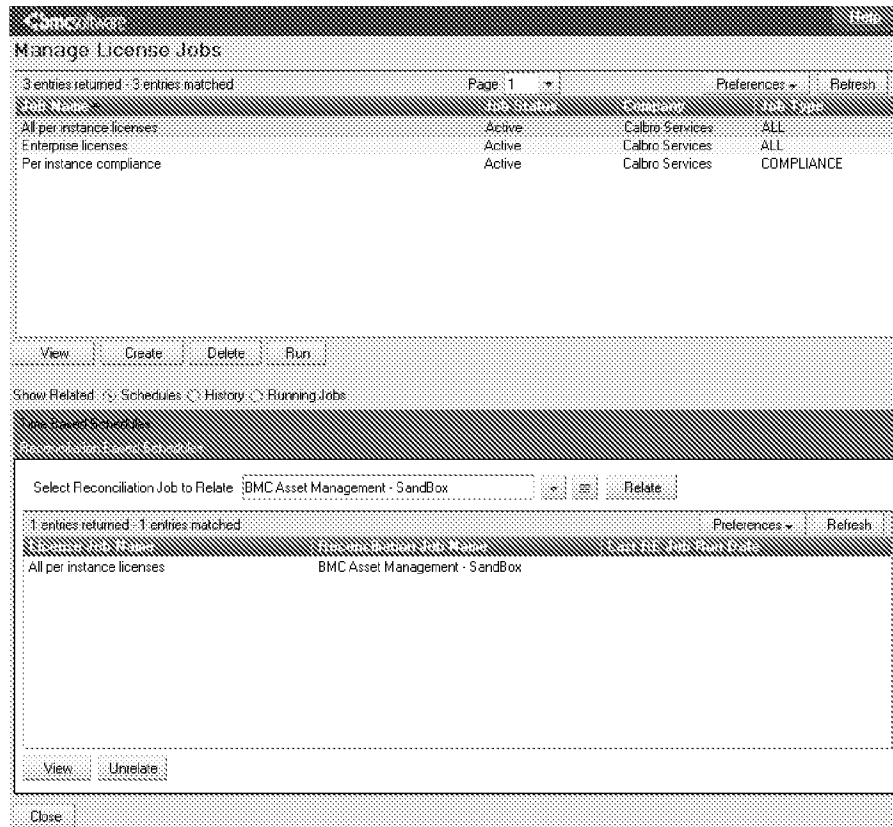
The License Engine runs only jobs that you create and schedule.

You can manage these jobs from the Manage License Jobs console. From this console, you can also see the results of license job executions.

About the Manage License Jobs console

From the Manage License Jobs console, which is illustrated in Figure 7-4, you can manage license jobs and view the results of license job executions. The top half of the console lists license jobs.

Figure 7-4: Manage License Jobs console



Depending on your selection for the Show Related option, the bottom half of the console shows one of the following sections:

- **Schedule**—Lists the schedules for the selected license job. You can create new schedules, view (and modify) details of schedules, and delete schedules. Schedules can be time-based or reconciliation-based. To manage time-based schedules, click the Time Based Schedules title. To manage the reconciliation-based schedules, click the Reconciliation Based Schedules table title.
- **History**—Displays the history of the selected license job. The history displays the following information about the license job:
 - The number of certificates connected to CIs.
 - The number of certificates with multiple certificates.
 - The number of software CIs that are not related to certificates.
 - The number of certificates out of compliance.

To view details for the license job run, select the job and click View Details.

- **Running Jobs**—If the select job is running, displays the status.

Creating a job

You can create a job to check license compliance, or to connect CIs to licenses and then check compliance. After you create the job, you can either run it immediately or schedule it to run at a later time.

► To create a job

- 1 In the navigation pane of the Software Asset Management console, choose Functions > Manage License Jobs.
- 2 In the Manage Licence Jobs console below the list of license jobs, click Create.
- 3 In the Create License Job dialog box, specify the following information.

Field	Description
Company	This job applies to licenses for the company that you select.
Job Name	Enter a descriptive name to identify this job.
Job Status	Select whether the job is active or inactive.
Job Type	To check for CIs to connect to the license and then check compliance, select CONNECTION AND COMPLIANCE. To check compliance without connecting CIs, select COMPLIANCE.

- 4 To restrict the certificates being checked, specify the job criteria.

Field	Description
License Type	You can run jobs for a specific license type, such as Per Instance.
Product catalog fields	You can specify the product manufacturer, product name, product model/version, or product categorization. <i>Note:</i> If you specify model/version in the job, but the certificate does not indicate model/version, no license certificates are checked.
Advanced Certificate Criteria	You can build any certificate criteria with the advanced certificate criteria.

- 5 To specify the CIs being checked, specify the following information:

Field	Description
DataSet Name	To check the production dataset, leave the default value BMC Asset. To check another dataset, such as a test dataset, select the dataset.
Advanced CI Criteria	The license certificate contains connection questions that determine applicable CIs for the license. However, you can specify Advanced CI Criteria to restrict the scope of CIs that are checked. <i>Note:</i> Limiting the scope of CIs that are checked improves performance when running the license job.

- 6 Click Save.

Running a job immediately

After you create a job, you can run it immediately. For example, to check compliance in preparation for an audit, create a compliance job, as described in “Creating a job” on page 169, and then run it.

► Running a job immediately

- 1 In the navigation pane of the Software Asset Management console, choose Functions > Manage License Jobs.
- 2 Select the job and click Run.

The job runs immediately. You can view the results in the History, as described in “Viewing the results of a license job” on page 170.

Scheduling a time-based license job

After you create a job, you can schedule it to run on a recurring basis.

► To schedule a time-based license job

- 1 In the navigation pane of the Software Asset Management console, choose Functions > Manage License Jobs.
- 2 Select the job and, in the bottom half of the console, click Create.
- 3 In the Job Schedule Information dialog box, select each day that the job should run, such as Sunday and Wednesday.
- 4 Select the schedule time.
- 5 Click Save.

Viewing the results of a license job

History displays the results of a license job.

► To view the history of a license job

- 1 In the navigation pane of the Software Asset Management console, choose Functions > Manage License Jobs.
- 2 Select the license job.
- 3 For Show Related, select History.

Scheduling a reconciliation-based license job



After you create a job, you can schedule it to run after reconciliation. Reconciliation-based jobs only check CIs that were modified after the last time the job was run.

► To schedule a reconciliation-based license job

- 1 In the navigation pane of the Software Asset Management console, choose Functions > Manage License Jobs.
- 2 Select the job and, in the bottom half of the console, click Reconciliation Based Schedules.
- 3 Select the applicable reconciliation job, and click Relate.

Reviewing a software license certificate



The License Certificate form displays the following information:

- License certificate details, including the license type, status, company, and manufacturer
- Accounting and purchasing information about the license
- A list of all software CIs attached to the license certificate
- A list of certificates grouped with the current license certificate

From the License Certificate form, you can perform the actions listed in Table 7-4.

Table 7-4: Actions that you can perform on a license certificate

Action	Description
View information not displayed on the certificate form.	From the Functions menu in the navigation pane, you can view the following information: <ul style="list-style-type: none"> ※ Audit trail ※ Connection details—Information that you provided when adding the license certificate ※ Compliance Details—Information that you provided when adding the license certificate, and information resulting from a license job run, such as the number of licenses deployed.
Manage grouping.	Manage certificate groups, as described in “Manually managing certificate groups” on page 166.

Table 7-4: Actions that you can perform on a license certificate (Continued)

Action	Description
Relate the license certificate to another license certificate.	When you renew or upgrade a software license, you can relate the new and old license certificates to each other, to help maintain traceability, as described in “Relating one license certificate to another license certificate.”
Modify the certificate.	<p>For master certificates and ungrouped certificates, you can modify any information on the license certificate, except for the software contract ID, license type, and company. You cannot modify grouped certificates, because the details on the master certificate apply to all the grouped certificates.</p> <p>Note: If you modify the connection details or the compliance details, you have the option to remove all software CIs from the certificate. To connect the appropriate CIs, you must run a license job, as described in “Manually managing CIs attached to a license certificate” on page 174.</p>

You can perform high-level reviews of all software license certificates from the Software Asset Management console. For information about the console, see “About the Software Asset Management console” on page 157.

Relating one license certificate to another license certificate

When you renew or upgrade a software license, you can relate the new and old license certificates to each other, to help maintain traceability.

For upgrades, both certificates might be executed. Software CIs for the old version are connected to the old certificate, and software CIs for the new version are connected to the new certificate.

For renewals, the old certificate is typically historical. Only the new certificate is executed.

You can relate certificates for other reasons, if appropriate.

NOTE

Relating licence certificates is not the same as grouping license certificates. For information about grouping license certificates, see “Manually managing certificate groups” on page 166.

► **To relate one license certificate to another**

- 1 Open the Software Asset Management console.
- 2 Select the license certificate, and click View.

- 3 On the License Certificate form, click Related Certificates.
- 4 In the Create Type list, select from Renew, Upgrade, or Relate.
- 5 To create a new certificate to relate to the current license certificate, click Create.
Information from the current license certificate is copied to the Create License Certificate wizard. For information about creating a license certificate, see “Adding a license certificate to a software contract” on page 163.
- 6 To relate the license certificate to an existing license certificate, complete the following steps.
 - a Click Search.
 - b In the Searching for Certificates dialog box, specify the search criteria and click Search.
 - c Select the appropriate license certificate.
 - d Select whether the relationship indicates an upgrade, a renewal, or that the two certificates are otherwise related, and click Relate.

Determining which CIs use a license certificate



You can determine which CIs use a license certificate, which can help you handle the following situations:

- A license certificate is out of compliance. You want to find software instances that can be removed.
- A software contract is up for negotiation. You want to determine if all the licenses are required.
- You have a license certificate that is not applied automatically, at least for some CIs. You want to confirm that the appropriate CIs are related to the license certificate. If necessary, you can manually relate CIs to the license certificate.

► To determine which CIs use a license certificate

- 1 Open the Software Asset Management console.
- 2 Select the license certificate.

The Related CIs table lists CIs that use the selected license certificate.

Manually managing CIs attached to a license certificate



The License Engine automatically attaches CIs to license certificates, based on company, product information, and answers to connection questions. However, if necessary, you can manually add or remove software CIs from the license certificate. Software CIs include the following CI types: Operating System, Package, Patch, Product, Software Server, and System Software.

TIP

If you manually remove a CI from a license certificate, manually add it to the correct license certificate. Otherwise, when the License Engine runs, it might attach the CI to the same certificate.

► **To add or remove a software CI from a license certificate**

- 1 Open the license certificate from either the Software Asset Management console or the Contract Management console, as described in the following table.

Console	Steps to open the license certificate
Software Asset Management console	<ol style="list-style-type: none"> 1 Select the license certificate. 2 Click View certificate.
Contract Management console	<ol style="list-style-type: none"> 1 Select the applicable software license contract. 2 Click View. 3 On the License Details tab, select the license certificate and click View.

Figure 7-5: License Certificate form

- 2 Click the Software Assets tab.

The Software Assets tab lists all the CIs that use the license certificate.

- 3 To add a CI, perform the following steps.

- a From the CI Type lists, select the type of configuration item that you want to relate to the license certificate.
- b Specify the CI to which you are relating the current contract.
 - To create the CI, click Create and complete the new CI form.
 - To select a CI, search for the CI, select the relationship type, and then click Relate.

- 4 To remove a CI, select it and click Remove.

- 5 In the message window that appears, select whether to run a license job.
The license job checks for connections and compliance.
- 6 If you run a license job, complete the Create License Job dialog box and click Submit.
The Company, License Type, and product catalog fields are filled in with values from the license certificate. In the Job Name field, you must enter a descriptive name to identify this job.

Manually specifying a license certificate for a software CI



The License Engine automatically attaches software CIs to license certificates, based on company, product information, and answers to connection questions. However, if necessary, you can manually specify the license certificate for a software CI. Software CIs include the following CI types: Operating System, Package, Patch, Product, Software Server, and System Software.

► **To manually specify a license certificate for a software CI**

- 1 Open the software CI from either the Software Asset Management console or the Asset Management console, as described in the following table.

Console	Steps to open the CI
Software Asset Management console	<ol style="list-style-type: none"> 1 Expand the Unrelated CIs panel. 2 From the Job Name list, select the job that found the unrelated CI. 3 Select the CI from the list of Unrelated CIs, and click View.
Asset Management console	Search for the CI, as described in "Searching for CIs" on page 32.

- 2 On the CI form, click the Contracts tab.
- 3 In the Search Existing Contracts Area, select the License Certificate contract type, and click Search.
In the Searching for Certificates dialog box, you can refine your search.
- 4 Select the applicable software contract, select a relationship type of Attached to, and click Relate.

- 5 In the message window that appears, select whether to run a license job.
The license job checks for connections and compliance.
- 6 If you run a license job, complete the Create License Job form and click Submit.
The Company, License Type, and product catalog fields are filled in with values from the license certificate. In the Job Name field, you must enter a descriptive name to identify this job.

Recording the purchase cost for a license



On a certificate, you can record the purchase cost for a license. Each software contract displays the costs for the licenses attached to the contract.

► To record the purchase cost for a license

- 1 From the Software Asset Management console, open the license certificate.
- 2 In the Purchase Cost field, type the cost to purchase the license, and select the currency.
The cost center comes from the software contract.
- 3 Click Save.

Troubleshooting software license management



This section helps you troubleshoot the following problems that you might encounter in software license management:

- No CIs are related to a software license certificate.
- One or more CIs are not related to any software license certificates.
- Incorrect CIs are related to a software license certificate.
- A “per copy” license is approaching breach faster than expected.

For your records, and to assist with troubleshooting, each certificate includes an audit trail of modifications to the certificate and of notifications sent about the certificate.

No CIs are related to a software license certificate

Table 7-5 lists possible causes why no CIs are related to a software license certificate, along with the diagnostic method to determine the cause.

Table 7-5: Tips for troubleshooting why no CIs are related to a software license certificate

Possible cause	Diagnostic method
Categorization specified on the license certificate does not match categorization of discovered CIs.	View the license certificate, as described in "Reviewing a software license certificate" on page 171. If categorization fields are completed, open a CI to see whether the categorization matches. To fix this issue, change the categorization on the certificate either to match the CI, or to be blank.
Connection questions are not answered correctly. Connection questions must be answered correctly to connect the CI to the certificate.	View the license certificate and check the answers to connection questions.
No job has been run to connect the CIs to the certificate.	Open the Manage License Jobs console, as described in "Manually managing CIs attached to a license certificate" on page 174, and check the history. View the jobs to determine whether any apply to the license type.
The license type is not correctly configured. When creating a license type, an administrator specifies how the connection question answers map to CIs in BMC Atrium CMDB. An error can result in no CIs being related to the certificate.	An administrator can examine the license type, as described in the <i>BMC Remedy IT Service Management Configuration Guide</i> . To check for issues with rules, open the AST:ConfigRuleSet form. This is an intermediate form between the Configure License Type wizard and the License Engine.
The License Engine did not complete its run.	Check the following forms: <ul style="list-style-type: none"> ※ RLE:RunHistory—Provides information about each License Engine run, including status (Pending, Running, Aborted, Completed, Completed with Warning, Completed with Errors). ※ RLE:EngineExceptions - Lists Java™ exceptions, RLE exceptions, and the ruleset ID. ※ arjavaplugin.log—This log file is typically located in the ARserver\Db directory of the server. If you turn on logging, this log file provides details of the License Engine job run. Logging is configured from the Application Administration console, as described in the <i>BMC Remedy IT Service Management Configuration Guide</i>.
The Company attribute (or another attribute used to connect the CI to the license certificate) is not set on the CIs.	Search for and open the appropriate CI, as described in "Searching for CIs" on page 32. The Company and Manufacturer attributes are always used to connect a CI to a license certificate. Whether other attributes are required for a connection depends on the license type. For example, for a site license, the Site attribute must be specified on the computer system on which the product is installed.

One or more CIs are not related to any software license certificates

Table 7-6 lists possible causes why one or more CIs would not be related to any software license certificates, along with the diagnostic method to determine the cause.

Table 7-6: Tips for troubleshooting one or more CIs are not related to a software license certificate

Possible cause	Diagnostic method
No software license certificates are applicable for the CI.	On the Software Asset Management console, check the Unrelated CIs panel. This panel lists CIs that could not be matched to license certificates by a job. View the license certificate, as described in “Reviewing a software license certificate” on page 171. If no other CIs appear on the Software Assets tab, continue to diagnose the cause, as described in “No CIs are related to a software license certificate” on page 178. If the Software Assets tab lists incorrectly related CIs, continue to diagnose the cause, as described in “Incorrect CIs are related to a software license certificate” on page 179.
A CI could be related to multiple certificates. Because the License Engine cannot determine the appropriate certificate, the CI is not related to any software license certificate.	On the Software Asset Management console, check the Unrelated CIs panel. This panel lists CIs that could not be matched to license certificates by a job. Check the inbox for a message with additional details. From the inbox message, you can relate the CI to the appropriate certificate.

Incorrect CIs are related to a software license certificate

Table 7-7 lists possible causes why incorrect CIs are related to a software license certificate, along with the diagnostic method to determine the cause.

Table 7-7: Tips for troubleshooting why incorrect CIs are related to a software license certificate

Possible cause	Diagnostic method
Categorization is too broad. Many CIs match the specified configuration.	View the license certificate, as described in “Reviewing a software license certificate” on page 171. If a product is not specified, all CIs that match the manufacturer and categorization are related to the certificate.
Connection questions are not answered correctly. Connection questions must be answered correctly to connect the correct CI to the certificate.	View the license certificate and check the answers to connection questions.
The license type is not correctly configured. When creating a license type, an administrator specifies how the connection question answers map to CIs in BMC Atrium CMDB. An error can result in incorrect CIs being related to the certificate.	An administrator can examine the license type, as described in the <i>BMC Remedy IT Service Management Configuration Guide</i> .

A “per copy” license is approaching breach faster than expected

The “per copy” license type is based on the number of unique people using the license. Unique people are the people related to the computer system on which the software is installed. These people include both people related to the computer system through BMC Remedy ITSM with the “used by” relationship and people who are discovered with dependency on the computer system.



People created in BMC Remedy ITSM are stored in the People form, but are also reconciled into BMC Atrium CMDB in the BMC_Person class. People who are discovered by BMC discovery products are stored in BMC Atrium CMDB in the BMC_Person class. If more people are discovered on a computer system than expected, you might be using more licenses than expected.

BMC_Person records, when related to computer systems, are displayed on the Relationships tab of the Computer System form. The People tab of the Computer System form displays the people related through BMC Remedy ITSM, but does not include the people related through BMC discovery products. The Relationships tab displays records for all related people.

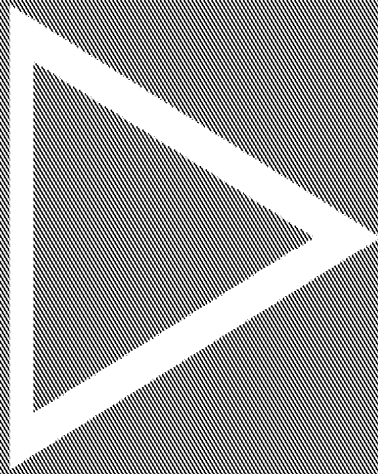
You can build a query to search for people as described in “Using the BMC Atrium CMDB query dialog box to search for CIs” on page 35.

► To determine the people using a per copy license

- 1 Open the Software Asset Management console and select the license.
- 2 In the Related CIs table, select a related CI and click View CI.
- 3 In the Product form, click the Relationships tab.
- 4 Select the computer system CI, and click View.
- 5 In the Computer System form, click the Relationships tab and count the number of BMC_Person records.
- 6 Repeat step 2 through step 5 for each product CI that is related to the selected license, and total the number of BMC_Person records.



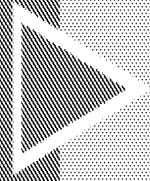
ACTIVATE BUSINESS WITH THE POWER OF I.T.™



7.5 AM License Implementation Connection & Compliance

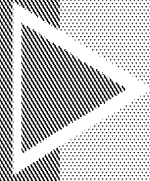
Tony Myers
April 25th, 2008

Agenda



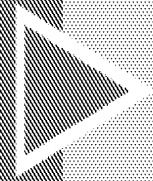
- › **Review Client License Types**
 - Types supported
 - Out of compliance scenarios
- › **Review Server License Types**
 - Types supported
 - Out of compliance scenarios
- › **Review Mainframe License Types**
 - Types supported
- › **Ensuring data is accurate in the CMDB**

Agenda



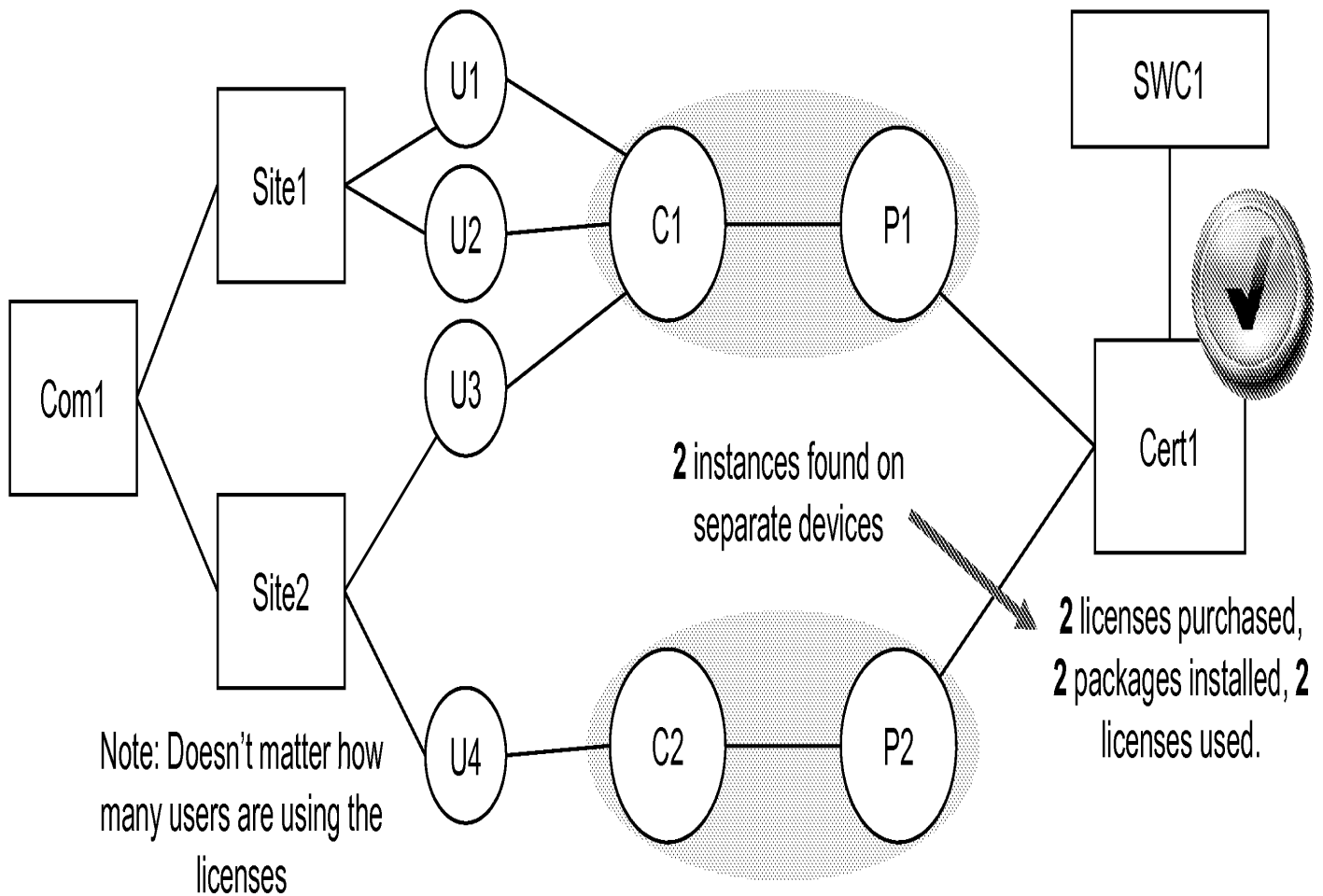
- › **Review Client License Types**
 - Types supported
 - Out of compliance scenarios
- › **Review Server License Types**
 - Types supported
 - Out of compliance scenarios
- › **Review Mainframe License Types**
 - Types supported
- › **Ensuring data is accurate in the CMDB**

*Client License – Per Copy Per Device

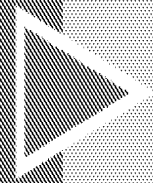


- > # licenses purchased supplied
- > Compliance = comparison between
 - # licenses purchased vs.
 - # of PRODUCT CI's with appropriate categorization

Rally ID: S5347

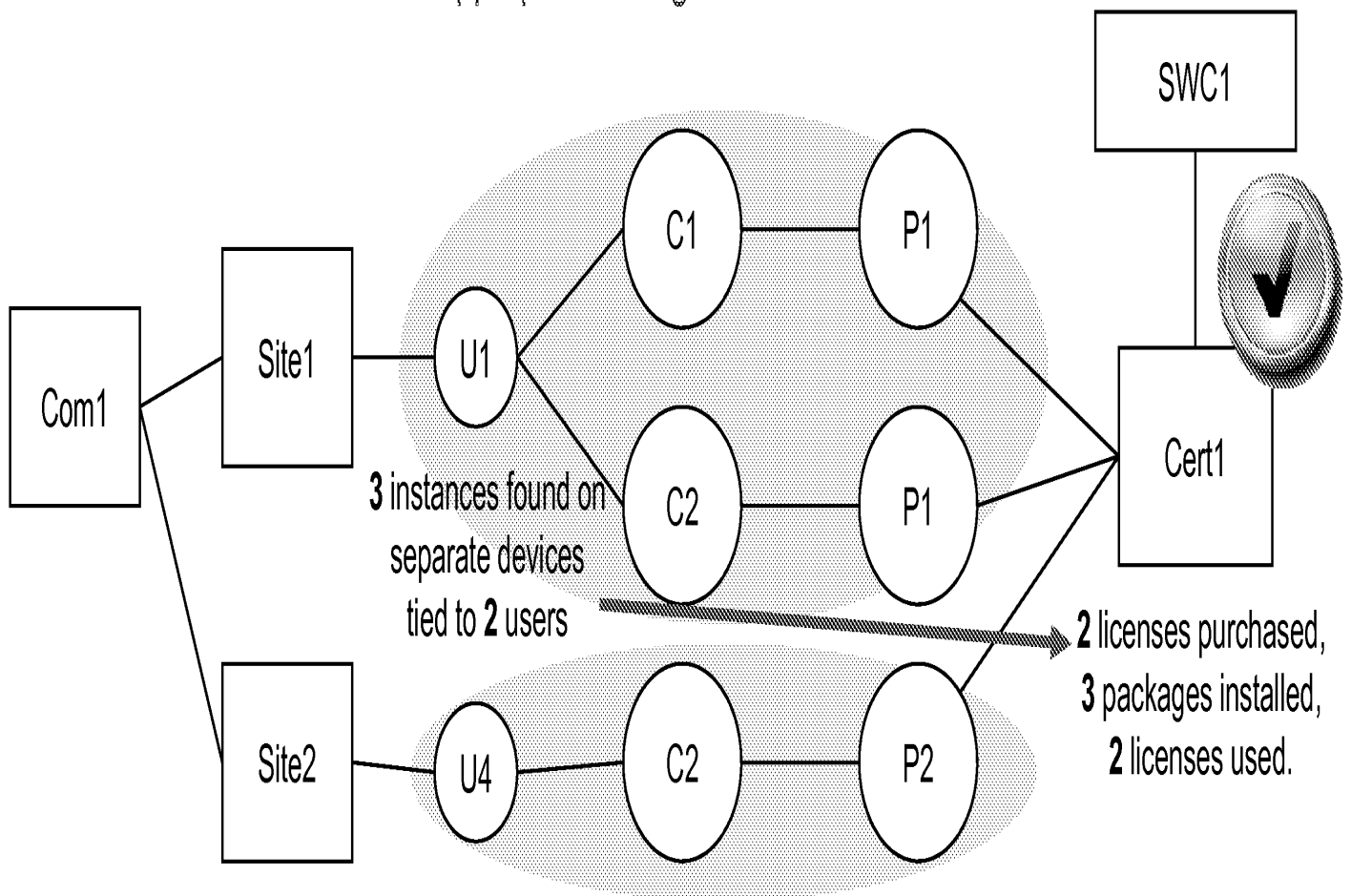


*Client License – Per Copy

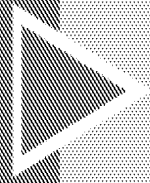


- > # licenses purchased supplied
- > # installations supplied
- > Compliance = comparison between
 - # licenses purchased vs.
 - # of PRODUCT CI's with appropriate categorization for different users

Rally ID: S5349

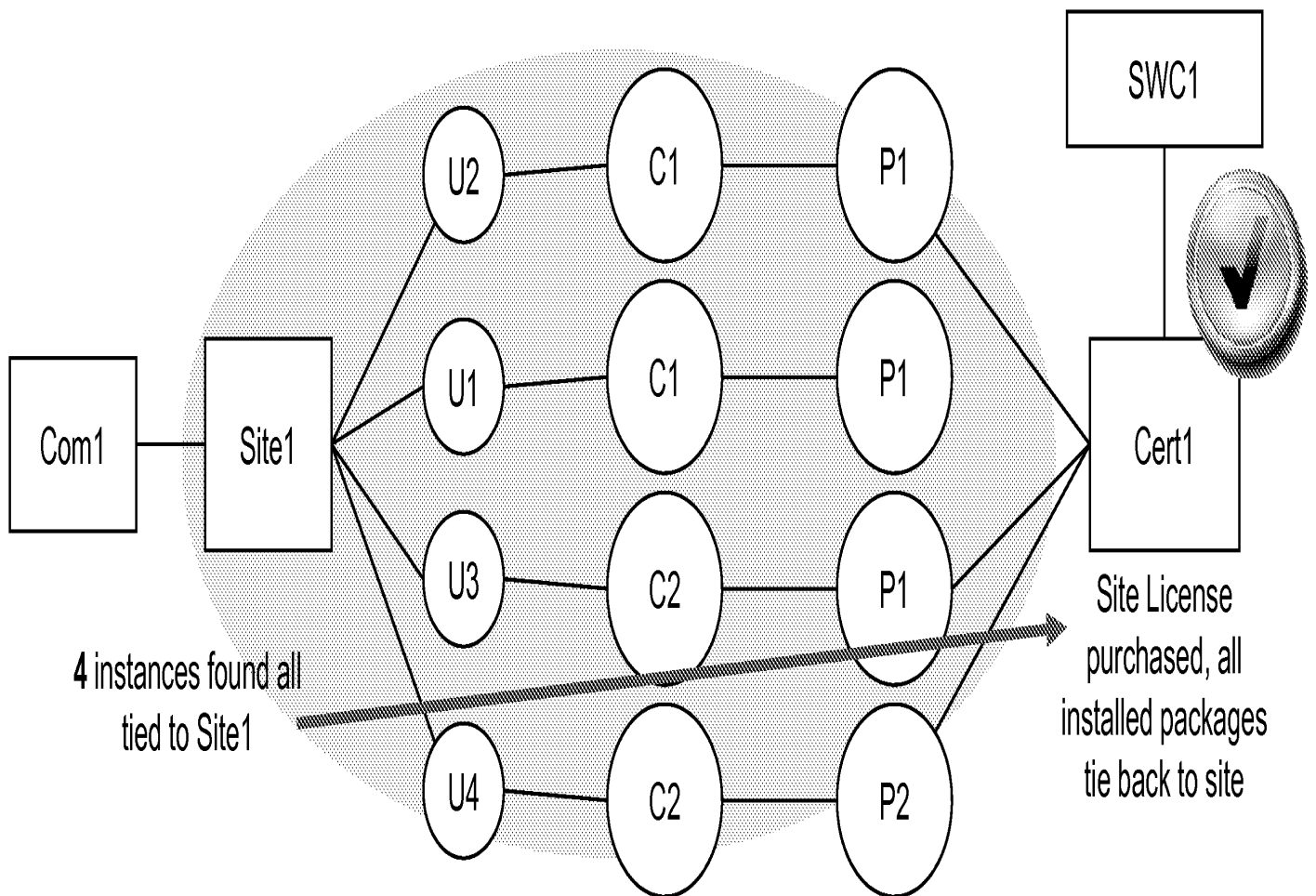


*Client License – Site (Location)

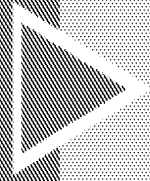


- › Location that restricts license is supplied
- › Compliance = comparison between
 - PRODUCT CI's tied to user that belongs to the site

Rally ID: S5343

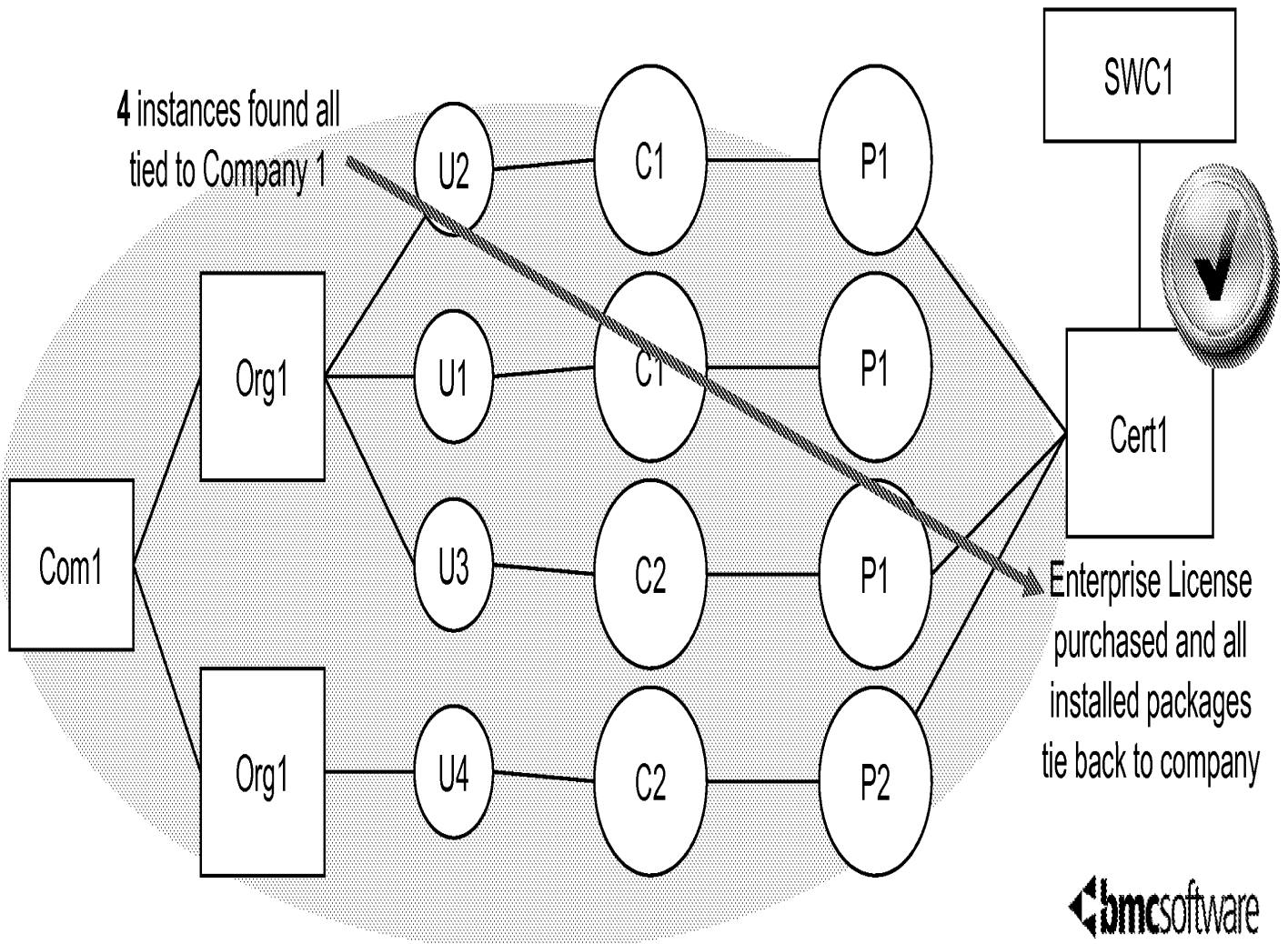


*Client License – Enterprise (Organization)

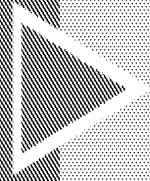


- › Organization structure that restricts license is supplied
- › Compliance = comparison between
 - PRODUCT CI's tied to user that belongs to that Organization

Rally ID: S5343



Agenda



› Review Client License Types

- Types supported
- **Out of compliance scenarios**

› Review Server License Types

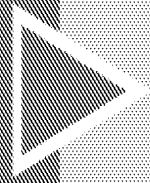
- Types supported
- Out of compliance scenarios

› Review Mainframe License Types

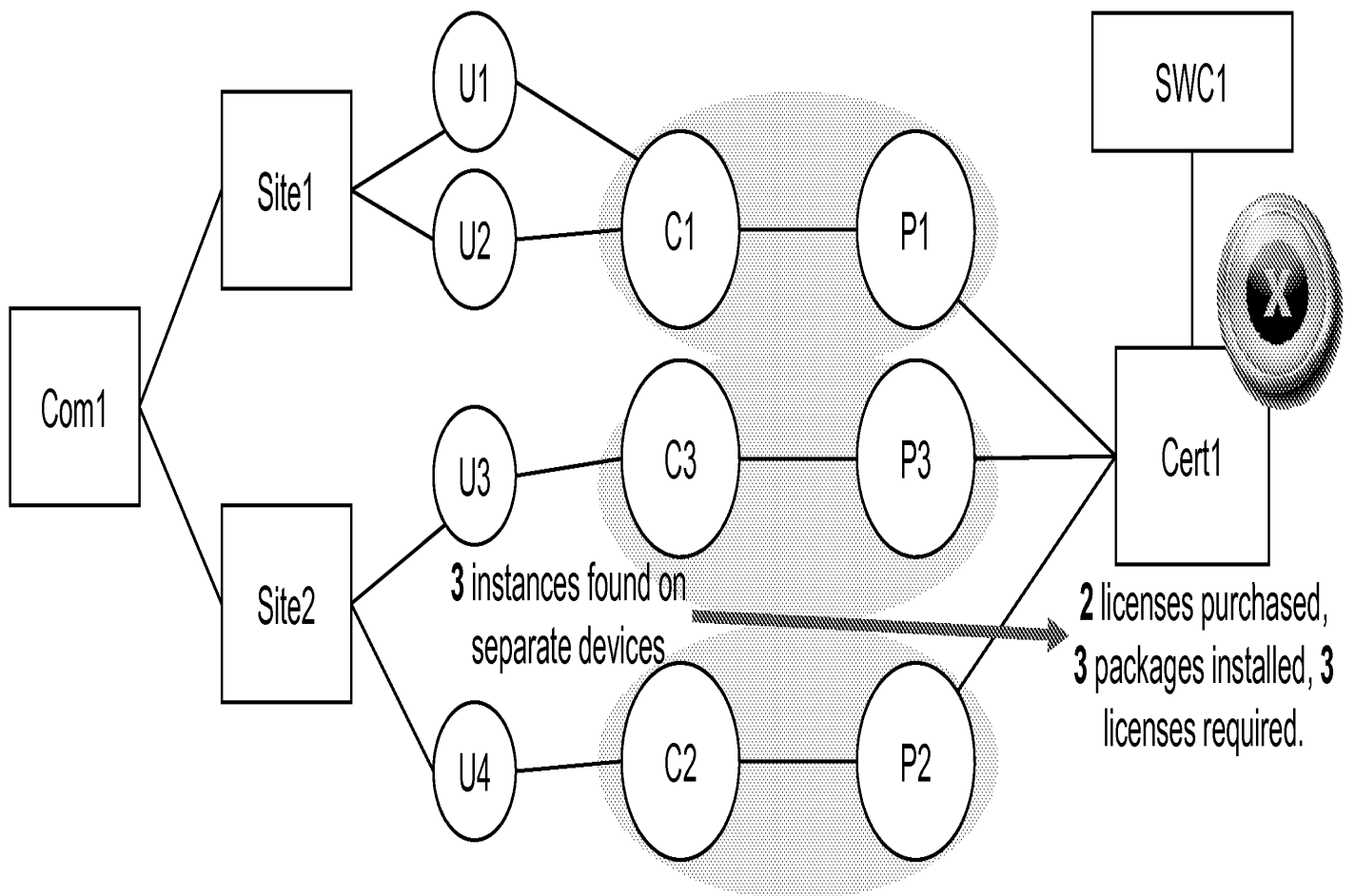
- Types supported

› Ensuring data is accurate in the CMDB

Client License – Non-Compliance Example 1



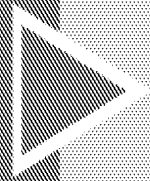
> Per Copy Per Device Non Compliance Example



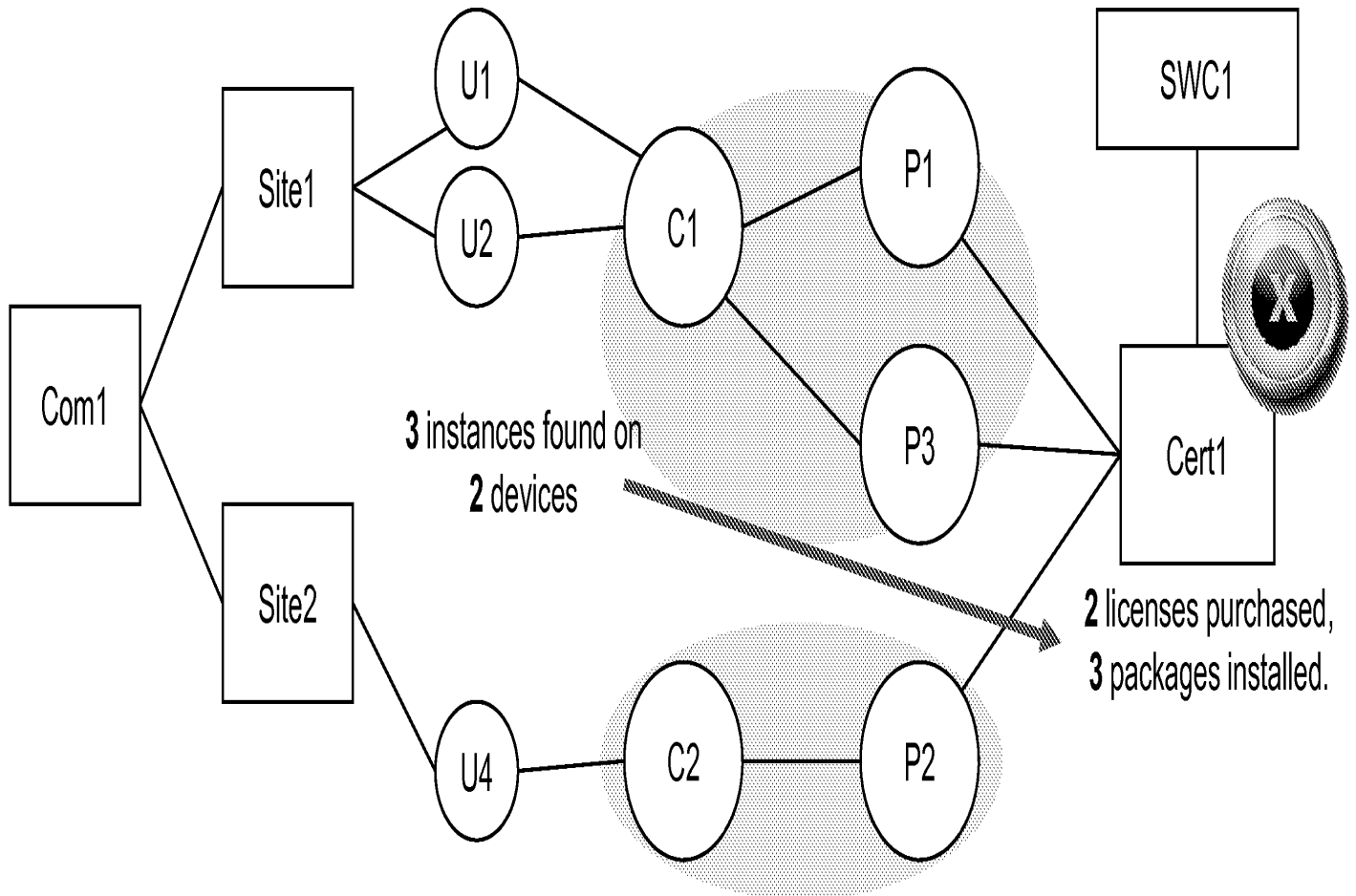
Requires new license to be purchased or the removal of 1 License



Client License – Non-Compliance Example 2



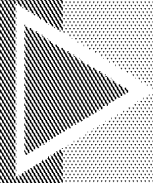
> Per Copy Per Device Non Compliance Example



Requires P1 or P3 to be removed

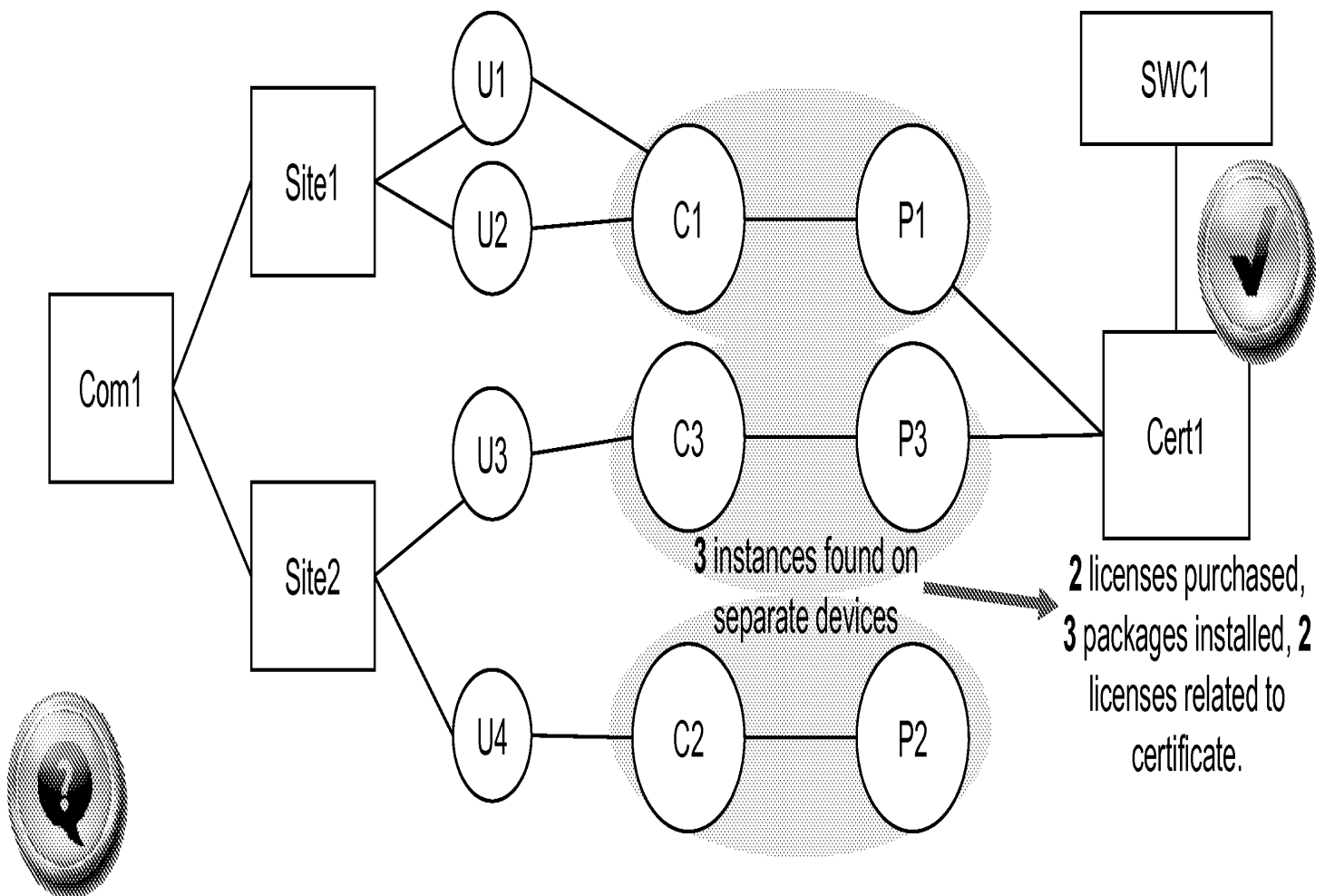


Client License – Non-Compliance Example 3



> Per Copy Per Device Non Compliance Example

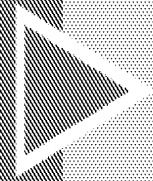
– Engine cannot connect Instance to a certificate



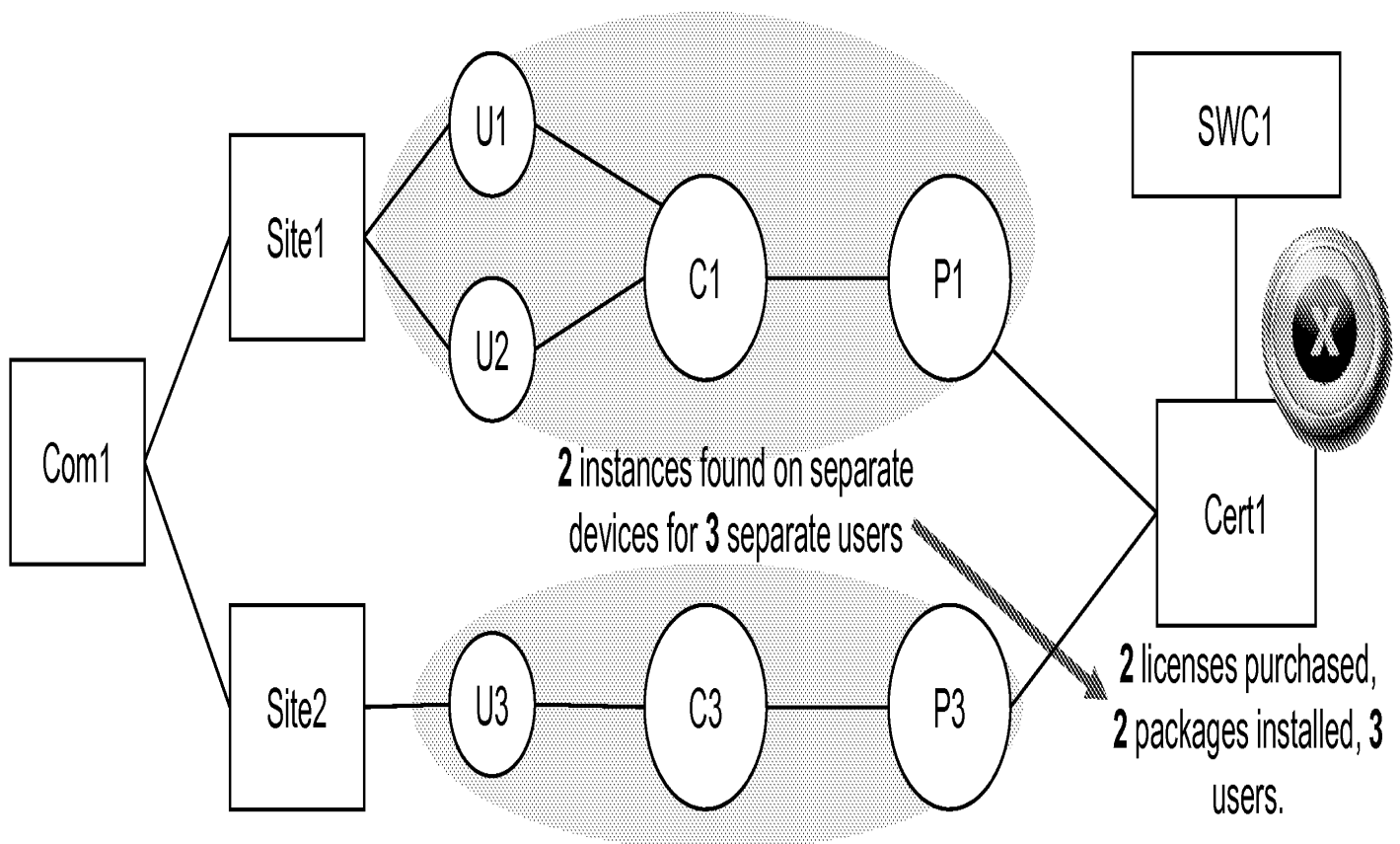
Certificate is in compliance but overall compliance with vendor is unknown.
Requires Manual intervention to associate to correct certificate.



Client License – Non-Compliance Example 4



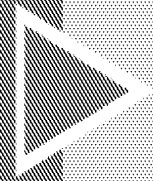
> Per Copy Non Compliance Example



Requires U1 or U2 to be removed from C1

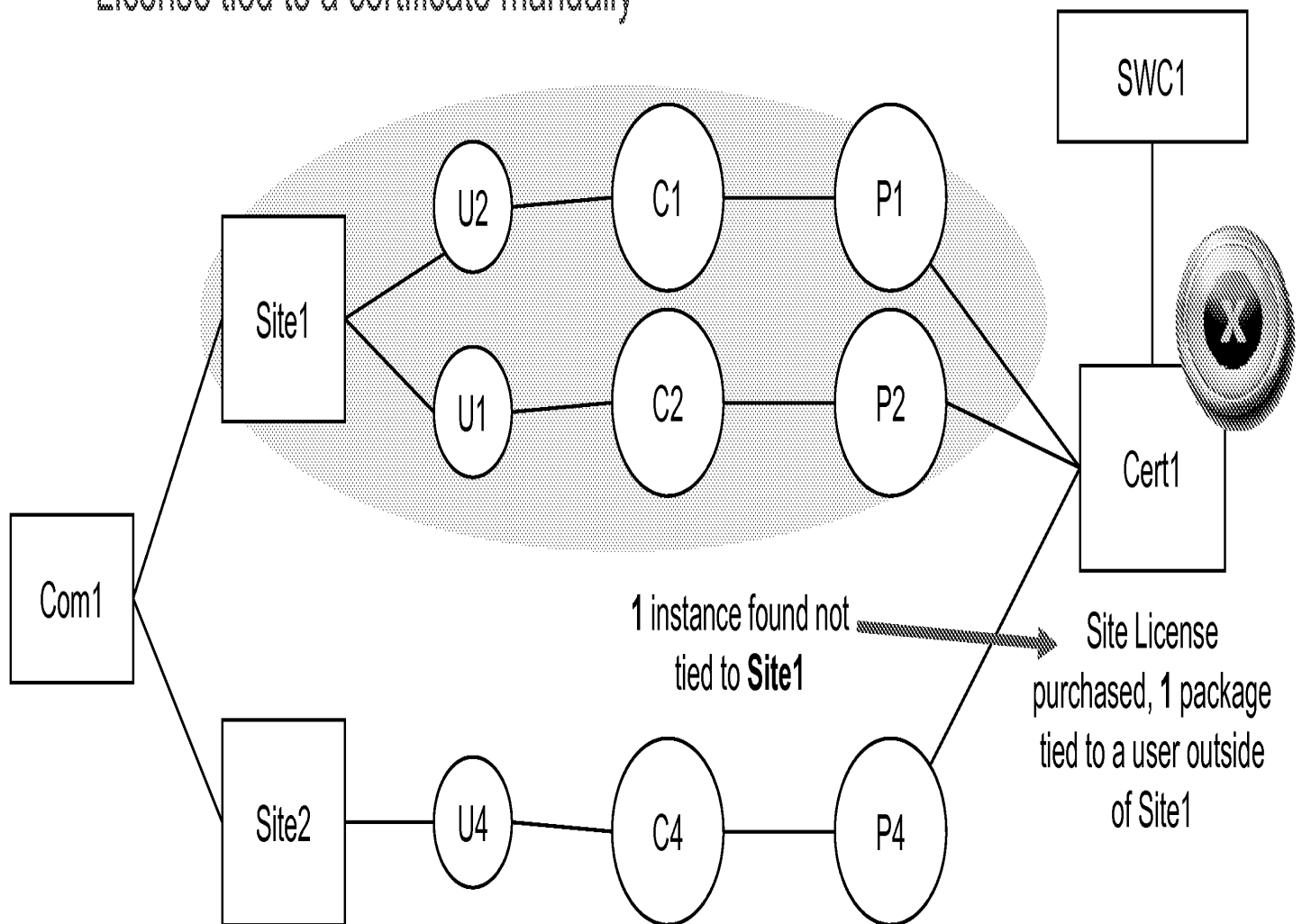


Client License – Non-Compliance Example 5



> Site non-compliance Example (Site1)

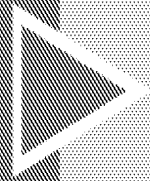
– License tied to a certificate manually



Requires P4 to be removed from C4 or associate with a valid certificate

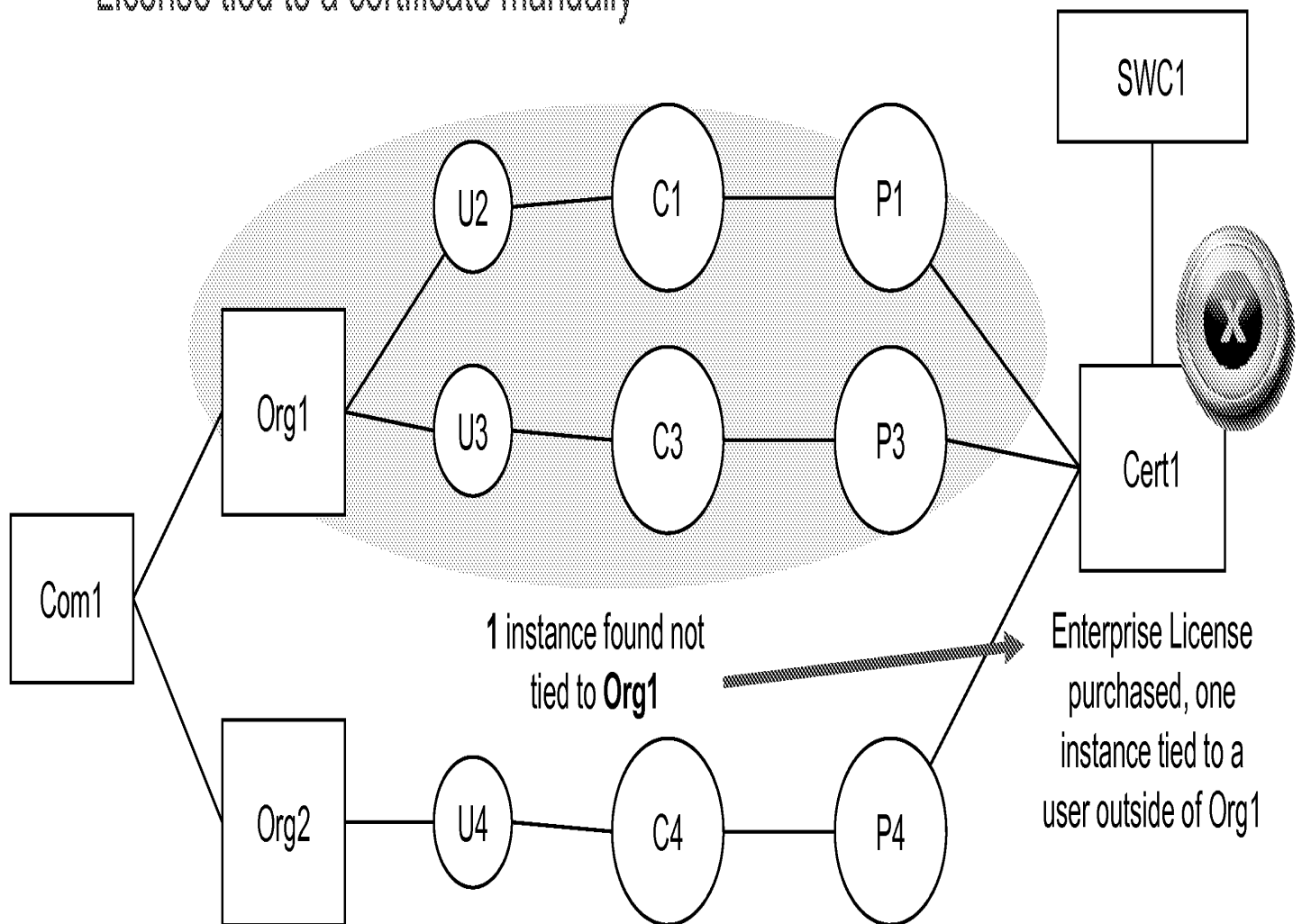


Client License – Non-Compliance Example 6



> Enterprise non-compliance Example (Org1)

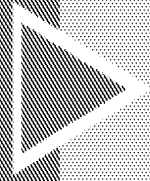
– License tied to a certificate manually



Requires P4 to be removed from C4 or associate with a valid certificate



Agenda



- › **Review Client License Types**
 - Types supported
 - Out of compliance scenarios
- › **Review Server License Types**
 - **Types supported**
 - Out of compliance scenarios
- › **Review Mainframe License Types**
 - Types supported
- › **Ensuring data is accurate in the CMDB**

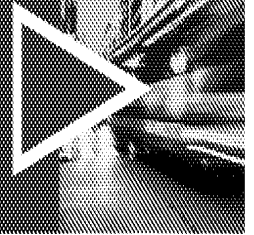


ACTIVATE BUSINESS WITH THE POWER OF IT.™



SWLM High Level Architecture

SWLM Goals



› Scalable/Extensible Model to support SWLM

- Should not add additional load to the insert of data into the CDMB.
- Needs to be able to manage new contract types easily.

› Be able to track multiple types of Software Contracts

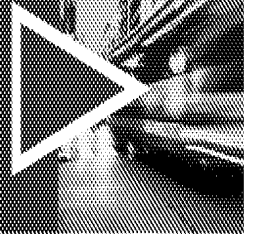
- Desktop Related
 - Count type rules based on number of copies deployed.
 - Site based rules.
 - Shrink/Click Wrap type rules
- Enterprise Related
 - CPU or Core based calculations.
 - Mainframe based models.
 - Various other calculation type models negotiated by the customer.

› Support ability to do what if scenarios

› Ability to run reports, track exceptions etc.

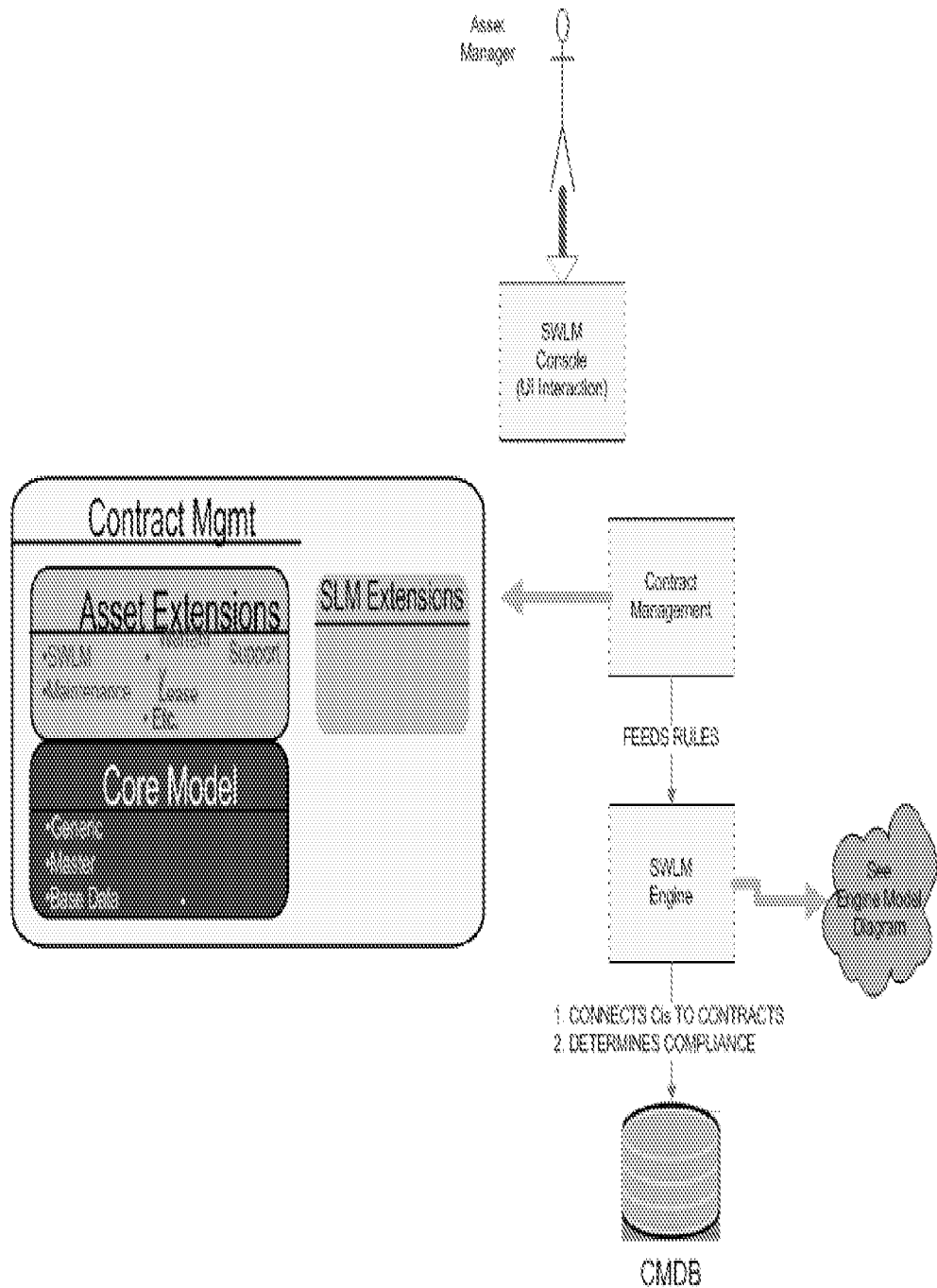
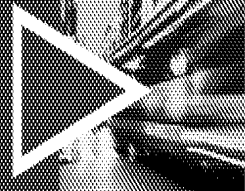


SWLM Components

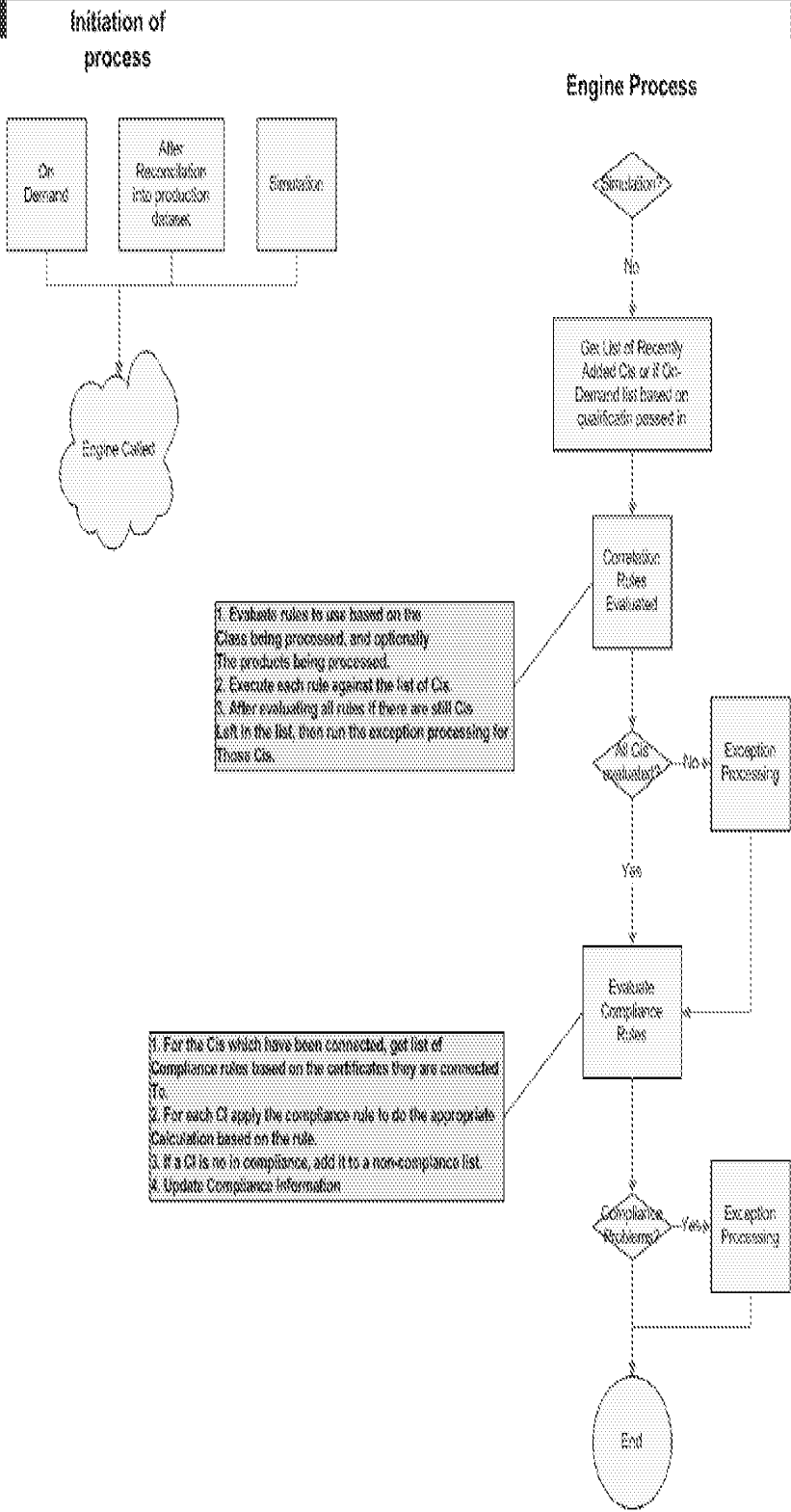
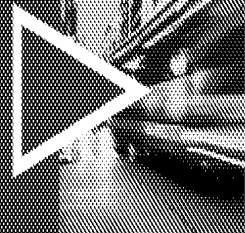


- › Console to provide interaction model for Asset Manager.
- › Extensions to Contract Model.
- › Rules based SWLM Engine.

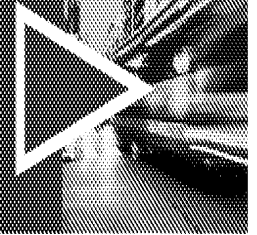
SWLM High Level View



SWLM High Level Engine Flow



Contract Model Extensions



› Extensions to Base Model

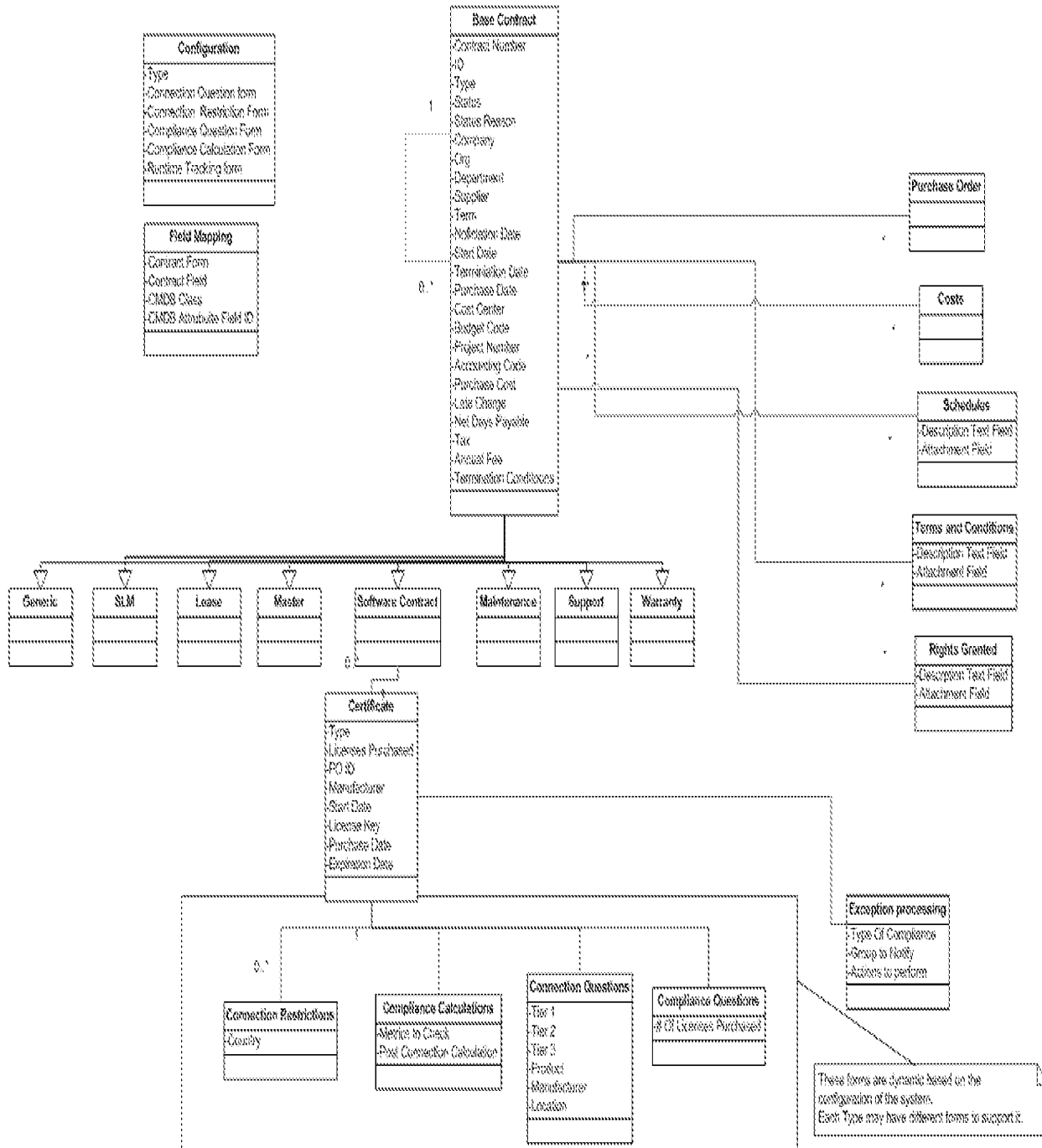
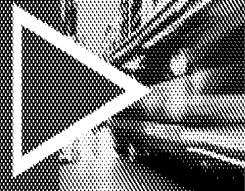
- New Master Contract Concept
- Ability to group together disparate contract types under one master or just related to each other.

› Extensions to Asset SWLM Model

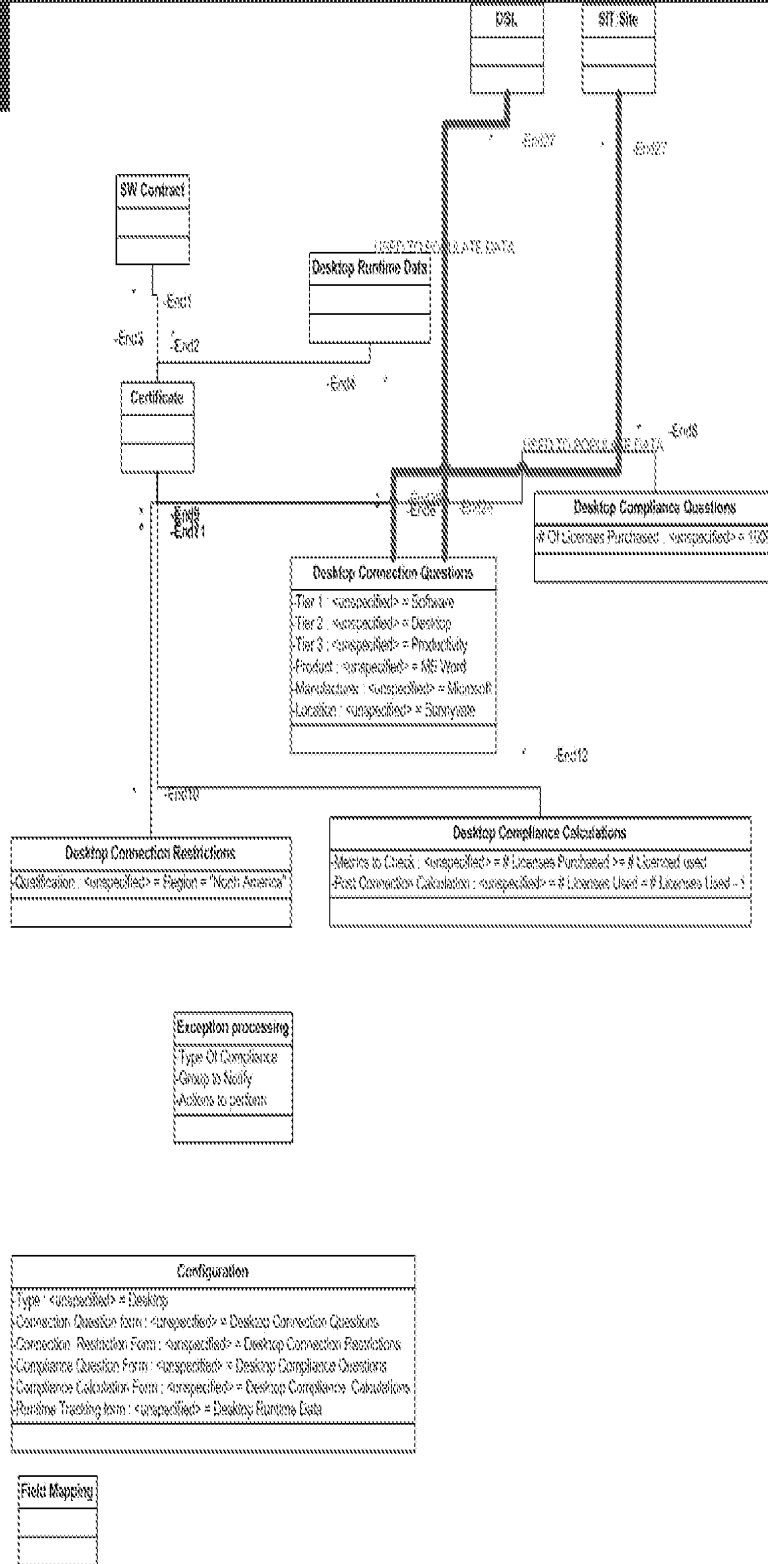
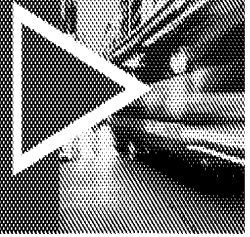
- New Certificate Concept
 - Represents individual purchases of licenses underneath a Software License contract.
- New Constructs to Enable Rules Capture
 - Tracks Information about how to connect a CI to a contract.
 - Tracks Information about how to calculate Compliance.
 - Plug-in Model based on License Type
 - Different forms can be exposed to capture different data based on the license type.
 - Forms are registered as part of configuration.
 - First Release of SWLM will come with forms for Desktop License Support



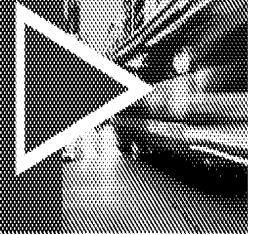
Contracts Data Model



Contract Instantiation Model



Engine Requirements



› Needs to run in different modes

– Batch Mode

- Async process run after all data has been reconciled.

- Called from Recon Engine

– On-Demand Mode

– Simulation Mode

- Ability to make use of simulation rules and runtime data to do evaluations.

› Set of Cis to process is done via seed qualifications

› Two Sets of Rule Types

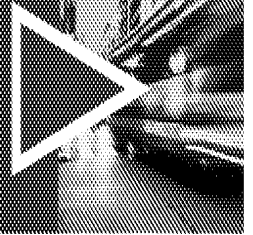
– Compliance

– Calculation

› Rules Data Model Designed to be extensible

- Can add new rules repositories easily.

Engine Requirements (Cont.)



- › Rules should be cached and as much processing done in memory as possible to improve performance.

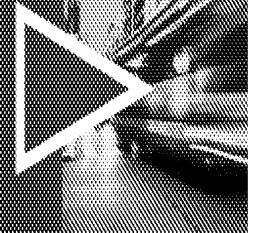


ACTIVATE BUSINESS WITH THE POWER OF IT.™



Use Case 1 – Per instance licencing and Grouping

Use Case – Example – Visio licenses




FLOW	EVENTS	ACTIONS	RESULTS
<p>Licenses Purchased</p>	<ul style="list-style-type: none"> • New purchase of 100 Visio licenses • Executed under existing Software License Contract with Vendor 	<ul style="list-style-type: none"> • Wizard for new certificate is executed • Client license model is selected • # of licenses is supplied (100) 	<ul style="list-style-type: none"> • New certificate is created that has 100 copies licensed for Visio (CERT1)
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> • 20 copies of software is installed on desktops • Discovery runs, recognizes the new s/w on the desktop • Reconciliation runs and populates the golden dataset 	<ul style="list-style-type: none"> • License engine runs and recognizes 20 new copies of software installed • Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> • Software is linked to the certificate CERT1 • Licenses used is incremented by 20 – new total is 20 • Licenses available is decremented by 20 – new total is 80
<p>Compliance understood</p>	<ul style="list-style-type: none"> • License engine completes job • Need to understand compliance of Microsoft 	<ul style="list-style-type: none"> • Summary of engine run is documented as messages • Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name 	<ul style="list-style-type: none"> • Message of “20 Visio licenses connected to CERT1” is created • SAM console shown Visio Product as “Green” – in compliance



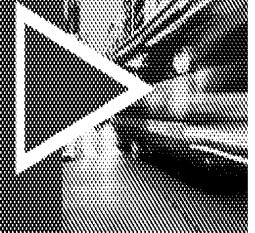
Use Case – Example – More Visio licenses



FLOW	EVENTS	ACTIONS	RESULTS
<p>Licenses Purchased</p>	<ul style="list-style-type: none"> • No new licenses are purchased 		
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> • 83 more copies of software is installed on desktops • Discovery runs, recognizes the new s/w on the desktop • Reconciliation runs and populates the golden dataset 	<ul style="list-style-type: none"> • License engine runs and recognizes 83 new copies of software installed • Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> • 83 instances of software is linked to the certificate CERT1 • Licenses used is incremented by 83 – new total is 103 • Licenses available is decremented by 83 – new total is -3
<p>Compliance understood</p> 	<ul style="list-style-type: none"> • License engine completes job • Need to understand compliance of Microsoft 	<ul style="list-style-type: none"> • Summary of engine run is documented as messages • Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name 	<ul style="list-style-type: none"> • Message of “83 Visio licenses connected to CERT1” is created • Message of “ERROR – CERT1 is out of compliance by 3 licenses. Please investigate and either purchase more licenses or uninstall to ensure compliance” • SAM console shown Visio Product as “Red” – out of compliance



Use Case – Example – Grouping certificates



FLOW	EVENTS	ACTIONS	RESULTS
<p>Licenses Purchased</p>	<ul style="list-style-type: none"> • New purchase of 50 Visio licenses • Executed under existing Software License with Vendor 	<ul style="list-style-type: none"> • Wizard for new certificate is executed • Client license model is 	<ul style="list-style-type: none"> • New certificate is created that has 50 copies licensed for Visio (CERT2) • System recognizes that there is another certificate for the same software and prompts user if they want to group them • User confirms grouping. • System now rolls up numbers from certificate
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> • 12 copies of installed on desktop • Discovery runs, recognizes the new s/w on the desktop • Reconciliation runs and populates the golden dataset 	<ul style="list-style-type: none"> • software installed • Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> • New software is linked to the certificate CERT2 • Total licenses available is 150 – 100 on CERT1 & 50 on CERT2 • Licenses available is decremented by 12 – new total is 0 on CERT1 and 35 on CERT2 • Licenses used is incremented by 12 – new total is 100 on CERT1 and 15 on CERT2
<p>Compliance understood</p>	<ul style="list-style-type: none"> • License engine completes job • Need to understand compliance of Microsoft 	<ul style="list-style-type: none"> • Summary of engine run is documented as messages • Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name 	<ul style="list-style-type: none"> • Message of “12 Visio licenses connected to CERT2” is created • Message of “3 Visio licenses re-connected to CERT2” is created • SAM console shown Visio Product as “Green” – in compliance

CERT1 was out of compliance by 2 - 102 licenses are used for only 100 licenses purchased. Grouping the certificate will cause the re-allocation of licenses across the certificates - # used on CERT1 is 100, # used on CERT2 is 15 even though only 12 were discovered





ACTIVATE BUSINESS WITH THE POWER OF IT.™



Use Case 2 – Per copy licencing

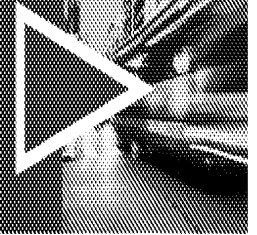
Use Case – Example – Adobe Licenses



FLOW	EVENTS	ACTIONS	RESULTS
<p>Licenses Purchased</p>	<ul style="list-style-type: none"> • New purchase of 100 Adobe Acrobat licenses • Executed under existing Software License Contract with Vendor 	<ul style="list-style-type: none"> • Wizard for new certificate is executed • Per Copy license model is selected • # of licenses is supplied (100) 	<ul style="list-style-type: none"> • New certificate is created that has 100 copies licensed for Adobe Acrobat (CERT1)
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> • 20 copies of software is installed. One copy is installed by the same user on 3 different machines • Discovery runs, recognizes the new s/w on the desktop • Reconciliation runs and populates the golden dataset 	<ul style="list-style-type: none"> • License engine runs and recognizes 20 new copies of software installed • Of the 20 new pieces of software, 18 users are unique • Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> • 20 instances of Software is linked to the certificate CERT1 • Licenses used is incremented by 18 – new total is 18 • Licenses available is decremented by 18 – new total is 82
<p>Compliance understood</p>	<ul style="list-style-type: none"> • License engine completes job • Need to understand compliance of Adobe 	<ul style="list-style-type: none"> • Summary of engine run is documented as messages • Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name 	<ul style="list-style-type: none"> • Message of “20 Adobe licenses connected to CERT1” is created • “18 licenses consumed” message is also displayed • SAM console shown Adobe Product as “Green” – in compliance



Use Case – Example – More Adobe software installed



FLOW	EVENTS	ACTIONS	RESULTS
<p>Licenses Purchased</p>	<ul style="list-style-type: none"> • No new licenses are purchased 		
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> • 100 more copies of software is installed by 80 different users. Of those 80, 2 already have other adobe licenses installed. • Discovery runs, recognizes the new s/w on the desktop • Reconciliation runs and populates the golden dataset 	<ul style="list-style-type: none"> • License engine runs and recognizes 100 new copies of software installed • Of the 80 different users, only 78 are new users of the software • Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> • 100 instances of software is linked to the certificate CERT1. Total # of software linked to the certificate is 120 • Licenses used is incremented by 78 – new total is 96 • Licenses available is decremented by 78 – new total is 4
<p>Compliance understood</p>	<ul style="list-style-type: none"> • License engine completes job • Need to understand compliance of Adobe 	<ul style="list-style-type: none"> • Summary of engine run is documented as messages • Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name 	<ul style="list-style-type: none"> • Message of “100 Adobe licenses connected to CERT1” is created. • “78 licenses consumed” message is also displayed • SAM console shown Adobe Product as “Green” – in compliance





ACTIVATE BUSINESS WITH THE POWER OF IT.™



**Use Case 3 – Upgrade purchased for Visio Licenses
(See use case 1 for background of scenario)**

Use Case – Upgrading Visio licenses



FLOW	EVENTS	ACTIONS	RESULTS
<p>Licenses Purchased</p>	<ul style="list-style-type: none"> • Vendor buys an upgrade license for 150 Visio licenses (2003>2007) • Executed under existing Software License Contract with Vendor 	<ul style="list-style-type: none"> • Wizard for new certificate is executed • Client license model is selected. Upgrade license is specified and the software being upgraded is specified. In this case, it will be a group of licenses (2 certificates > 1) • # of licenses is supplied (150) 	<ul style="list-style-type: none"> • New certificate is created that has 150 copies licensed for Visio 2007 upgrade (CERT3) • Total available on new cert takes existing installation into consideration • Existing certificates indicates they have been upgraded.
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> • 20 copies of software is upgraded on desktops • Discovery runs, recognizes the 2003 software has been removed and the new s/w is installed • Reconciliation runs and populates the golden dataset 	<ul style="list-style-type: none"> • License engine runs and recognizes 20 copies of 2003 is deleted – 15 from CERT1 and 5 from CERT2. 20 new copies of 2007 is recognized • Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> • 2003 software link to CERT1/2 is removed • Licenses used for 2003 Certs is decremented by 20 – new group total is 95 • Licenses available for 2003 DOES NOT DECREASE – total is still 35 • 20 copies of 2007 software linked to CERT 3. • Licenses used for 2007 is incremented by 20 – new total is 20 • Licenses available DOES NOT CHANGE – total still remains at 35
<p>Compliance understood</p>	<ul style="list-style-type: none"> • License engine completes job • Need to understand compliance of Microsoft 	<ul style="list-style-type: none"> • Summary of engine run is documented as messages • Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name 	<ul style="list-style-type: none"> • The Link between the 2003 software and the upgrade software is understood. Overall, only 150 copies of the software (either 2003 or the 2007 upgrade can exist to be in compliance). • SAM console shows Visio 2003 and 2007 upgrade licensed Product as “Green” – in compliance (20 copies of 2007 upgrade and 95 of 2003)

Use Case – New Visio 2003 install



FLOW	EVENTS	ACTIONS	RESULTS
<p>Licenses Purchased</p>	<ul style="list-style-type: none"> • No new licenses purchased 		
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> • A new copy of Visio 2003 is installed on a desktop (not upgraded) • Discovery runs, recognizes the new s/w on the desktop • Reconciliation runs and populates the golden dataset 	<ul style="list-style-type: none"> • License engine runs and recognizes the new copy of Visio 2003 is installed • Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> • 1 new instances of software is linked to the certificate CERT2 • Licenses used is incremented by 1 – new group total is 96 • Licenses available is decremented by 1 – new total is 34 • Total available for upgraded certificate is changed to 34
<p>Compliance understood</p>	<ul style="list-style-type: none"> • License engine completes job • Need to understand compliance of Microsoft 	<ul style="list-style-type: none"> • Summary of engine run is documented as messages • Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name 	<ul style="list-style-type: none"> • Message of “1 Visio 2007 license connected to CERT2” is created • SAM console shows Visio 2003 and 2007 upgrade licensed Product as “Green” – in compliance (20 copies of 2007 upgrade and 96 of 2003 out of a total of 150)



Use Case – New Visio 2007 licenses install



FLOW	EVENTS	ACTIONS	RESULTS
<p>Licenses Requested</p>	<ul style="list-style-type: none"> • 40 new users requests a copy of Visio 2007 through SRM 	<ul style="list-style-type: none"> • New request submitted and appropriate approvals are received 	<ul style="list-style-type: none"> • The users get access to Visio 2003 and the upgrade to 2007 • The users installs Visio 2003 and then runs the upgrade for 2007. • ALTERNATE FLOW – Configuration discovery pushes install and upgrade via change that is launched from the SRM process
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> • Discovery runs, recognizes the 40 new copies of s/w on the desktop • Reconciliation runs and populates the golden dataset 	<ul style="list-style-type: none"> • License engine runs and recognizes the new copy of Visio 2007 upgrade is installed • Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> • 40 new instances of software is linked to the certificate CERT3 • Licenses used is incremented by 40 – new total is 60 • Licenses available is decremented by 40 – new total is -6 • Visio 2003 licenses show licenses available as -6 as well
<p>Compliance understood</p>	<ul style="list-style-type: none"> • License engine completes job • Need to understand compliance of Microsoft 	<ul style="list-style-type: none"> • Summary of engine run is documented as messages • Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name 	<ul style="list-style-type: none"> • Message of “40 Visio 2007 Upgrade licenses connected to CERT3” is created • SAM console shows Visio 2003 and 2007 upgrade licensed Product as “Red” – out-of-compliance (60 copies of 2007 upgrade and 96 of 2003 out of a total of 150)



Use Case – All Visio 2003 licenses upgraded



FLOW	EVENTS	ACTIONS	RESULTS
<p>Licenses upgraded</p>	<ul style="list-style-type: none"> IT mandates that all licenses will be upgraded to Visio 2007 	<ul style="list-style-type: none"> Visio 2003 instances are identified and upgraded ASSUME there are a total of 90 Visio 2003 software instances out there as 6 were already removed to make sure of compliance 	<ul style="list-style-type: none"> Configuration discovery pushes upgrade for identified machines via change that is launched from the SRM process
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> Discovery runs, recognizes the 90 removed 2003 software and the 90 new copies of Visio 2007 Upgrade Reconciliation runs and populates the golden dataset 	<ul style="list-style-type: none"> License engine runs and recognizes the removed copies of 2003 and the new copies of Visio 2007 upgrade Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> 2003 software link to CERT1/2 is removed Licenses used for 2003 Certs is decremented by 90 – new group total is 0 Licenses available for 2003 DOES NOT DECREASE – total is still 0 90 copies of 2007 software linked to CERT 3. Licenses used for 2007 DOES NOT CHANGE –total is still 150 Licenses available DOES NOT CHANGE – total still remains at 0
<p>Compliance understood</p>	<ul style="list-style-type: none"> License engine completes job Need to understand compliance of Microsoft 	<ul style="list-style-type: none"> Summary of engine run is documented as messages Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name CERT 1 and 2 are set to historical as they are not needed to be active anymore. 	<ul style="list-style-type: none"> Message of “90 Visio 2003 licenses Removed from CERT1/2” is created Message of “90 Visio 2007 Upgrade licenses connected to CERT3” is created SAM console shows Visio 2003 and 2007 upgrade licensed Product as “Green” – in compliance (150 copies of 2007 upgrade and 0 of 2003 out of a total of 150)





ACTIVATE BUSINESS WITH THE POWER OF IT.™



Use Case 4 – Microsoft Virtualization in a server farm

Background



Microsoft Virtualization Licensing Changes Benefit Customers

http://www.gartner.com/DisplayDocument?doc_cd=160859

21 August 2008

Stewart Buchanan Frances O'Brien

Microsoft will allow application licenses to be moved between physical servers using virtual server technologies. Harmonization of licensing and support policies between product groups can help to reduce customer licensing costs.

News Analysis

Event

On 19 August 2008, Microsoft announced changes to its licensing and support policy to enable application mobility under virtual servers. Specifically, 41 Microsoft application server products no longer need a server license for every physical server in the server farm, or a processor license for every processor in the farm, to dynamically move within the server farm. Customers now only need enough licenses to cover applications running simultaneously in a server farm. (For further details, see Microsoft's Web site.)

Analysis

Effective 1 September 2008, Microsoft will provide volume licensing customers with an exception to the 90-day rule (which only allows licenses to be transferred between devices once every 90 days). This exception allows selected application server licenses to be moved freely between servers within a server farm, effectively creating a concurrent licensing scheme. It reduces the number of server or processor licenses customers need to buy to the number they use. This is positive news, but this scheme requires asset management processes or tools to meter license use.

Microsoft defines a server farm as up to two physical data centers that are no more than four hours apart by universal time zone or within the European trade zone. A data center can only belong to a single server farm, but licenses can be reassigned between farms under the 90-day rule. There are no changes to Windows Server licenses. The 90-day rule still applies.

Harmonization of server application licenses and support policy marks a significant advance for Microsoft and its customers. Virtualization vendors already offering live migration capabilities may find these changes overdue and timed to favor Microsoft's own product releases. However, all server virtualization vendors stand to benefit from licensing that is more virtualization-friendly.

Microsoft also announced significant technical support policy changes. Thirty-one Microsoft application technologies are now supported under Microsoft's Server Virtualization Validation Program (see <http://support.microsoft.com/kb/957006> for a list of supported applications). This program coordinates multiple virtualization vendor support efforts, including VMware, under the Technical Support Alliance Network (TSA.net) and its code of conduct.

Recommendations

Customers:

- Take advantage of these support and licensing changes. They could significantly reduce the cost and risk of running these applications under the supported server virtualization technology of your choice.
- Recognize that counting licenses is not enough to ensure compliance. Your software asset management program should also start recording license assignments by server farm.



Use Case – SQL Server licensees in a virtual environment



FLOW	EVENTS	ACTIONS	RESULTS
<p>Licenses Purchased</p>	<ul style="list-style-type: none"> • Vendor buys 10 instances of SQL Server licenses for their server farm to be used for virtual applications • Executed under existing Software License Contract with Vendor 	<ul style="list-style-type: none"> • Wizard for new certificate is executed • Virtual Server license model is selected. Server farm is supplied • Restriction of 90 days between servers is supplied • # of licenses is supplied (10) 	<ul style="list-style-type: none"> • New certificate is created that has 10 copies licensed for SQL Server (CERT 9) • Server farm 1 has 5 physical servers • There is also a Server Farm 2 that has other SQL Server licenses
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> • 4 Virtual servers are instantiated that have SQL Server running. • The virtual images are running on 2 of the 5 servers in the server farm • Discovery runs, recognizes the software exists • Reconciliation runs and populates the golden dataset 	<ul style="list-style-type: none"> • License engine runs and recognizes 4 copies of SQL server exists • Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> • 4 Instances of SQL Server software is connected to CERT9 • Total used is incremented by 4 – new total is 4 • Total available is decremented by 4 – new total is 6
<p>Compliance understood</p>	<ul style="list-style-type: none"> • License engine completes job • Need to understand compliance of Microsoft SQL Server 	<ul style="list-style-type: none"> • Summary of engine run is documented as messages • Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name 	<ul style="list-style-type: none"> • Compliance determines the # of copies installed is below the total available, there was no movement between servers before 90 days, and all images stayed in server farm 1. • SAM console shows SQL Server as “Green” – in compliance



Use Case – SQL Server License Server Farm Move




FLOW	EVENTS	ACTIONS	RESULTS
<p>Image moved</p>	<ul style="list-style-type: none"> • An existing VM image moves to server farm 2. • Server farm 2 is not licensed for SQL Server licenses. 		
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> • The virtual image in question is connected to another server farm • Discovery runs, recognizes the image has changed locations • Reconciliation runs and updates the golden dataset 	<ul style="list-style-type: none"> • License engine runs and recognizes copy of SQL server has moved to another server farm • Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> • Software is removed from the certificate • Total available and used is adjusted • Compliance runs and recognizes that the physical server has changed for the software and the new server is not in the server farm
<p>Compliance understood</p>	<ul style="list-style-type: none"> • License engine completes job • Need to understand compliance of Microsoft SQL Server 	<ul style="list-style-type: none"> • Summary of engine run is documented as messages • Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name 	<ul style="list-style-type: none"> • Message of “Server moved to non-licensed server farm – please investigate and correct”. Compliance determines the moved image went to another server farm that is not licensed. • SAM console shows SQL Server as “Red” – out-of-compliance



Use Case – SQL Server License Restricted Move



FLOW	EVENTS	ACTIONS	RESULTS
<p>Image moved</p>	<ul style="list-style-type: none"> • An existing VM image running on server 1 moves to server 2. • The VM image was on server 1 for only 32 days 		
<p>Software Connected to Contract</p>	<ul style="list-style-type: none"> • The virtual image in question is connected to the new server • Discovery runs, recognizes the image has changed locations • Reconciliation runs and updates the golden dataset 	<ul style="list-style-type: none"> • License engine runs and recognizes copy of SQL server has moved to server 2 • Uses the certificate information to determine if and how connections are to be made 	<ul style="list-style-type: none"> • No adjustment to certificate connection • Compliance runs and recognizes that the physical server has changed for the software and the restriction period has not been observed
<p>Compliance understood</p> 	<ul style="list-style-type: none"> • License engine completes job • Need to understand compliance of Microsoft SQL Server 	<ul style="list-style-type: none"> • Summary of engine run is documented as messages • Software Asset Manager looks at Software Asset Management (SAM) Console to understand compliance for Visio Product Name 	<ul style="list-style-type: none"> • Message of “VM Image move to another server violated restriction – please investigate and correct”. Compliance determines the moved image went to another server under the restriction period. • SAM console shows SQL Server as “Red” – out-of-compliance



Definition.

Complex rule is a collection of qualifications. Each qualification defines a class (CI or relationship) it's attributes and relations to other qualification if any as nested query for example. Qualifications sorted by their priority and the order is important as it specifies for the Engine the order of how to build complex query.

The reason for not store all information in one rule entry is that it can be nested and depth can't be predictable.

Data module.

In the database Complex rule will be spread on 3 forms – RLE:BaseRule, RLE:GetRuleSpecificData, RLE:ComplexRuleSpecificData. Foreign key that tight all together is a Rule ID.

For each complex rule will be:

1 entry in RLE:BaseRule, 1 entry on RLE:GetRuleSpecificData, number of entries on RLE:ComplexRuleSpecificData as a number of classes that combine the complex query

Complex Rule

RuleID – (String) Rule ID. There are will be more than one record with the same RuleID as now primary key is RuleID and Serial Number

Sequence – (Int) Sequence in the rule. Sequence sets the order engine adds CMDBQueryObjects to the list that later on passed to the QBP API. Therefore RelationshipDirection – IN or OUT should be considered according to this parameter.

Class – (String) CMDB class

NameSpace – (String) Name Space of the class

CMDBCOMPLEX_DataSetID - (String) Data Set ID of the data

Type – (Selection) Type of the object Relationship or CI. Engine will create class accordingly

RelationshipDirection – (Selection) NULL / CMDB_RELATIONSHIP_DIRECTION_OUT, CMDB_RELATIONSHIP_DIRECTION_IN.IN: Destination on the LEFT, Source on the RIGHT
OUT: Destination on the RIGHT, Source on the LEFT

Select – (String) Fields to select. Comma separated list of field names. Might be null.

Where – (String) Query statement in AR advanced query format as GetRule. Might be null.

HasNestedQuery – (Selection) True/False

NestedQueryOperator – (Selection) IN/NOT_IN/NULL

SelectorForNestedAttribute – (String) Attribute of the current class specifies join key with the nested query.

FirstElement - (Selection) True/False. Specifies first element of nested query.

Process.

Engine reads all the rules and sort them by sequence. It also reads and sorts qualifications entries of the complex rule. It will process them from the bottom to the top. Each time it finds a record with FirstElement true it knows that entries bellow (those that it didn't process yet) are of nested query. It builds then a nested query and continue. This way it goes up until it reaches the first entry then it has all the information required to build CMDBQuery and submit CMDBQueryByPath.

RLE:BaseRule

Name – (String) Rule name
RuleID – (String) Rule ID.
Sequence – (Int) Sequence in the rule
Status – (Selection)
RuleSetID – (String)

RLE:GetRuleSpecificData

RuleID (String) Rule ID
Source (Selection) = "CMDB_COMPLEX" This is a new selection option
From = NULL
Select = NULL
Where = NULL
CMDB_COMPLEX – (Selection) YES/NO
CMDBCOMPLEX_NameSpace - (String) Name Space

RLE:ComplexRuleSpecificData
Unique Index is RuleID+Sequence

RuleID – (String) Rule ID.
Sequence – (Int) Sequence in the rule

Class – (String) CMDB class
Type – (Selection) Type of the object Relationship or CI. Engine will create class accordingly
RelationshipDirection – (Selection) NULL / CMDB_RELATIONSHIP_DIRECTION_OUT, CMDB_RELATIONSHIP_DIRECTION_IN
Select – (String) Fields to select. Comma separated list of field names. Might be null.
Where – (String) Query statement in AR advanced query format as GetRule. Might be null.
HasNestedQuery – (Selection) True/False
NestedQueryOperator – (Selection) IN/NOT_IN/NULL
SelectorForNestedAttribute – (String) Attribute of the current class specifies join key with the nested query.
FirstElement - (Selection) True/False. Specifies first element of nested query.

Call following interfaces in order to create complex rule

On the RLE:ComplexRuleInterface

Input:

Action = "CREATE"

RuleName = "Rule name"

RuleSetID = Rule Set ID the rule will belong to

Sequence = Sequence of the rule within the rule set

DataSetID – (String) Dataset from which you want to retrieve instances

Optional input

RuleID = Rule ID

Output:

RuleID = Rule ID

For each qualification call **RLE:ComplexRuleQueryInterface**

Input:

RuleID – (String) Rule ID. There will be more than one record with the same RuleID as now primary key is RuleID and Serial Number

Sequence – (Int) Sequence in the rule

Class – (String) CMDB class

NameSpace – Class name space

Type – (Selection) Type of the object RELATIONSHIP or CI. Engine will create class accordingly

RelationshipDirection – (Selection) NULL / CMDB_RELATIONSHIP_DIRECTION_OUT, CMDB_RELATIONSHIP_DIRECTION_IN

Select – (String) Fields to select. Comma separated list of field names. Might be null.

Where – (String) Query statement in AR advanced query format as GetRule. Might be null.

HasNestedQuery – (Selection) True/False

NestedQueryOperator – (Selection) IN/NOT_IN/NULL

SelectorForNestedAttribute – (String) Attribute of the current class specifies join key with the nested query.

FirstElement - (Selection) True/False. Specifies first element of nested query.

Make Rule Enabled

Complex Rule is in disabled status by default as when it just created not all its qualification are created. Submit a record to RLE:RuleActionInterface with RuleId as a key and Action = ENABLE

All Computers Located in Sunnyvale

- * "All Computers located in Sunnyvale."
- * QBP:
- * BMC_ComputerSystem.
- * Source:BMC_ElementLocation:Destination.
- * BMC_PhysicalLocation(City="Sunnyvale")[0,Name,Instanceld]

First call RLE:ComplexRuleCreateInterface to create a container for qualifications of the new complex rule

On the RLE:ComplexRuleCreateInterface

Input:

Action = "CREATE"
RuleName = "MyFirstComplexRule"
RuleSet = 123
Sequence = 5
DataSetID = BMC.SAMPLE

Output:

RuleID = 789 – Use this id to create qualifications within the rule

Now call RLE:ComplexQualificationInterface to create sub qualifications that combine complex query

On the RLE:ComplexQualificationInterface

Action = CREATE

RuleID = 789

Sequence = 0 (This is not a sequence of the rule, but a sequence of the qualification within the rule)

Class = BMC_ComputerSystem

NameSpace = BMC.CORE

Type = Regular

RelationshipDirection = NULL

Select = Name,Instanceld

Query = NULL

HasNestedQuery = False

SelectorForNested Attribute = NULL

NestedQueryOperator = NULL

FirstElement = False

On the RLE:ComplexQualificationInterface

ACTION = CREATE

RuleID = 789

Sequence = 1

Class = BMC_ElementLocation

NameSpace = BMC.CORE

Type = Relationship

RelationshipDirection = CMDB_RELATIONSHIP_DIRECTION_OUT

Select = NULL

Query = NULL

HasNestedQuery = False

SelectorForNested Attribute = NULL

NestedQueryOperator = NULL

FirstElement = FALSE

Continued from previous page
All Computers Located in Sunnyvale

- * "All Computers located in Sunnyvale."
- * QBP:
- * BMC_ComputerSystem.
- * Source:BMC_ElementLocation:Destination.
- * BMC_PhysicalLocation(City="Sunnyvale")[0,Name,InstanceId]

On RLE:ComplexQualificationInterface

ACTION = CREATE
Rule = Complex Rule
Rule ID = MyFirstComplexRule
Sequence = 2
Class = BMC_PhysicalLocation
NameSpace = BMC.CORE
Type = Destination
Relationship Direction = NULL
Select = NULL
Query = City="Sunnyvale"
Nested Query = False
Selector For Nested Attribute = NULL
Nested Query Operator = NULL
First Element = FALSE

Now call RLE:RuleActionInterface to enable the rule

On the RLE:RuleActionInterface
Input:

Action = ENABLE
RuleID = 789

All Computers Located in Sunnyvale

- * "All Computers located in Sunnyvale."
- * QBP:
- * BMC_ComputerSystem.
- * Source:BMC_ElementLocation:Destination.
- * BMC_PhysicalLocation(City="Sunnyvale")[0,Name,Instanceld]

Rule = Complex Rule
RuleID = MyFirstComplexRule (Note – all the sub rules have the same ID)
Sequence = 100
Serial Number = 0
Class = BMC_ComputerSystem
Type = Regular
RelationshipDirection = NULL
Select = Name,Instanceld
Query = NULL
NestedQuery = False
SelectorForNested Attribute = NULL

```
queryObject = new CMDBQueryObjectRegular( cnameKey, alias, null );  
queryObjectList.add( queryObject );
```

Rule = Complex Rule
RuleID = MyFirstComplexRule
Sequence = 100
Serial Number = 1
Class = BMC_ElementLocation
Type = RelationShip
RelationshipDirection = CMDB_RELATIONSHIP_DIRECTION_OUT
Select = NULL
Query = NULL
NestedQuery = False
SelectorForNested Attribute = NULL

```
recursiveList.add(new CMDBQueryObjectRelationship( cnameKey, alias,  
null,CMDBQueryObjectRelationship.CMDB_RELATIONSHIP_DIRECTION_OUT ));
```

Rule = Complex Rule
RuleID = MyFirstComplexRule
Sequence = 100
Serial Number = 2
Class = BMC_PhysicalLocation
Type = Destination
RelationshipDirection = NULL
Select = NULL
Query = City="Sunnyvale"
NestedQuery = False
SelectorForNested Attribute = NULL

```
CMDBQueryQualifierValue opRight =  
    new CMDBQueryQualifierValueTypeValue(new Value( "Sunnyvale"));  
CMDBQueryQualifier qualifier = new CMDBQueryQualifierSimpleOp(  
CMDBQueryQualifierSimpleOp.CMDB_QUERY_QUAL_OP_EQUAL,  
"Name",opRight),
```


All Computers with MACOS operating system located Sunnyvale

```
* QBP:  
* BMC_PhysicalLocation(City="Sunnyvale").  
* Destination:BMC_ElementLocation:Source.BMC_ComputerSystem  
* (  
*   Instanceld=  
*     {  
*       BMC_ComputerSystem.  
*       Source:BMC_HostedSystemComponents:Destination.  
*       BMC_OperatingSystem(OSType="MACOS")[0,Instanceld]  
*     }  
*   )[2,Name,Instanceld]
```

On RLE:ComplexRuleInterface

Input:

Action = "CREATE"
RuleName = "MySecondComplexRule"
RuleSet = 123
Sequence = 5
RuleID = MySecondComplexRule
DataSetID = BMC.SAMPLE

Output:

RuleID = MySecondComplexRule

On RLE:ComplexQualificationInterface

RuleID = MySecondComplexRule
Sequence = 0
Class = BMC_PhysicalLocation
NameSpace = BMC.CORE
Type = Regular
RelationshipDirection = NULL
Select = Name, Instanceld
Query = City="Sunnyvale"
NestedQuery = False
SelectorForNested Attribute = NULL
FirstElement = False

On RLE:ComplexQualificationInterface

Rule = Complex Rule
RuleID = MySecondComplexRule
Sequence = 1
Class = BMC_ElementLocation
NameSpace = BMC.CORE
Type = Relationship
RelationshipDirection = CMDB_RELATIONSHIP_DIRECTION_IN
Select = NULL
Query = NULL
NestedQuery = False
SelectorForNested Attribute = NULL
FirstElement = False

Continued from the previous page
All Computers with MACOS operating system located Sunnyvale

```
* QBP:  
* BMC_PhysicalLocation(City="Sunnyvale").  
* Destination:BMC_ElementLocation:Source.BMC_ComputerSystem  
* (  
*   Instanceld=  
*     {  
*       BMC_ComputerSystem.  
*       Source:BMC_HostedSystemComponents:Destination.  
*       BMC_OperatingSystem(OSType="MACOS")[0,Instanceld]  
*     }  
*   )][2,Name,Instanceld]
```

On RLE:ComplexQualificationInterface

Rule = Complex Rule
RuleID = MySecondComplexRule
Sequence = 2
Class = BMC_Computer_System
NameSpace = BMC.CORE
Type = Regular
RelationshipDirection = NULL
Select = NULL
Query = NULL
HasNestedQuery = True
SelectorForNested Attribute = Instanceld
FirstElement = False

On RLE:ComplexQualificationInterface

Rule = Complex Rule
RuleID = MySecondComplexRule
Sequence = 3
Class = BMC_ComputerSystem
NameSpace = BMC.CORE
Type = Regular
RelationshipDirection = NULL
Select = NULL
Query = City="Sunnyvale"
HasNestedQuery – False
SelectorForNested Attribute
FirstElement = **True**

On RLE:ComplexQualificationInterface

Rule = Complex Rule
RuleID = MySecondComplexRule
Sequence = 4
Class = BMC_HostedSystemComponents
NameSpace = BMC.CORE
Type = Relationship
RelationshipDirection = CMDB_RELATIONSHIP_DIRECTION_OUT
Select = NULL
Query = NULL
HasNestedQuery = False
SelectorForNested Attribute = NULL
FirstElement = False

Continued from the previous page
All Computers with MACOS operating system located Sunnyvale

```
* QBP:  
* BMC_PhysicalLocation(City="Sunnyvale").  
* Destination:BMC_ElementLocation:Source.BMC_ComputerSystem  
* (  
*   Instanceld=  
*     {  
*       BMC_ComputerSystem.  
*       Source:BMC_HostedSystemComponents:Destination.  
*       BMC_OperatingSystem(OSType="MACOS")[0,Instanceld]  
*     }  
*   ) [2,Name,Instanceld]
```

On RLE:ComplexQualificationInterface

Rule = Complex Rule
RuleID = MySecondComplexRule
Sequence = 5
Class = BMC_OperatingSystem
NameSpace = BMC.CORE
Type = Regular
RelationshipDirection = NULL
Select = NULL
Query = OSType = 2 (Actually it's OSType="MACOS" but since it's enumeration use this value, SWLM Wizard has to know about it)
NestedQuery = False
SelectorForNested Attribute = NULL

On RLE:RuleActionInterface

Input:
Action = ENABLE
RuleID = MySecondComplexRule

We need not to allow parallel runs with the same Company and Run Tag to avoid conflicts. We can resolve it by managing one main thread in the Engine which will recognize whether to proceed or push to a waiting queue some Runs. This approach will required thread managing.

We prefer to take advantage of the thread managing by AR. The resolution will be a queue that will be managed by the Rules Engine AR application. It will decide if execute Run or push it to a waiting queue. Any new Run will be pushed to **RLE:EngineRuns** with Status = "Pending". Another workflow triggered by new entry on that form or change status of any entry will pass run to the Plug-in Engine. Engine will check whether another run for the same RunTag is active and if not will start processing and will set Status = "Running".

Plug-in will have to signal upon completion of each run by changing the Status of completed Run to "Completed" or "Completed with errors".

In case of restart of the Engine Plug-in it will signal the interface to reset all Running Runs and change status to Aborted. To avoid situation where newly coming Runs will be aborted by the Engine Plug-in's restart a global field will be added to the **RLE:EngineRuns** form. This field will flag the workflow not to pass anything to the Plug-in (will not change any Status to Started). Reset process will set that flag to False. Changing of the flag to False will cause another workflow to fire. This workflow will group all the Waiting Runs by Company and Run Tag and pass to the Engine Plug-in the oldest one from each group.

Run status optional values: Pending, Running, Completed, Completed with Errors, Aborted

Interface will allow passing parameters to Rules throw ParametersToRules parameter. It would be a semicolon separated list of <Attribute Name>=<Attribute Value>;.....

Rules will use those parameters in the following way:

@AttributeName

Note, Use of dot sign in the attribute name is not allowed!

```
RLE:RuleEngineFireInterface
  RunTag
  RuleSetType
  ParametersToRules
```

An entry created on
RLE:RuleEngineFireInterface
 RunTag
 RuleSetType
 ParametersToRules

ParametersToRules list will be passed to the Engine's filter API and then will be accessible from Rules.

As a result a workflow will be fired and entry with run parameters and additional data will be created on
RLE:EngineRuns
 RunID = DATE+GUID
 Date Called = Current Date
 Status = Pending
 RuleSetType
 Company
 ParametersToRules

InReset = TRUE

NO

RLE:Pass2Engine workflow is triggered whether by creating **entry** or change of **status** field on **RLE:EngineRuns**.
 It will find entries with the same Company and RunTag and (Status = Pending or Status = Started).

YES

Found with Status = "Started" or Status = "Running"

YES

Another run for the same Company and RunTag in progress or Reset in progress – do nothing and return.

For entries found with Status="Pending" call Engine Filter API with parameters of that entry.
RLE:FireEngine
 Parameter 1 = RunID
 Parameter 2 = RunTagID
 Parameter 3 = RuleSetType
 Parameter 4 = ParametersToRules

For each completed run the java engine will update the status on **RLE:EngineRuns**
 RunID = Completed Run ID;
 Status = Completed/Completed with errors
 EndDate = Current date

Which will fire the RLE:Pass2Engine workflow and pending run will be passed to the Engine.

Completed with errors?

YES

NO

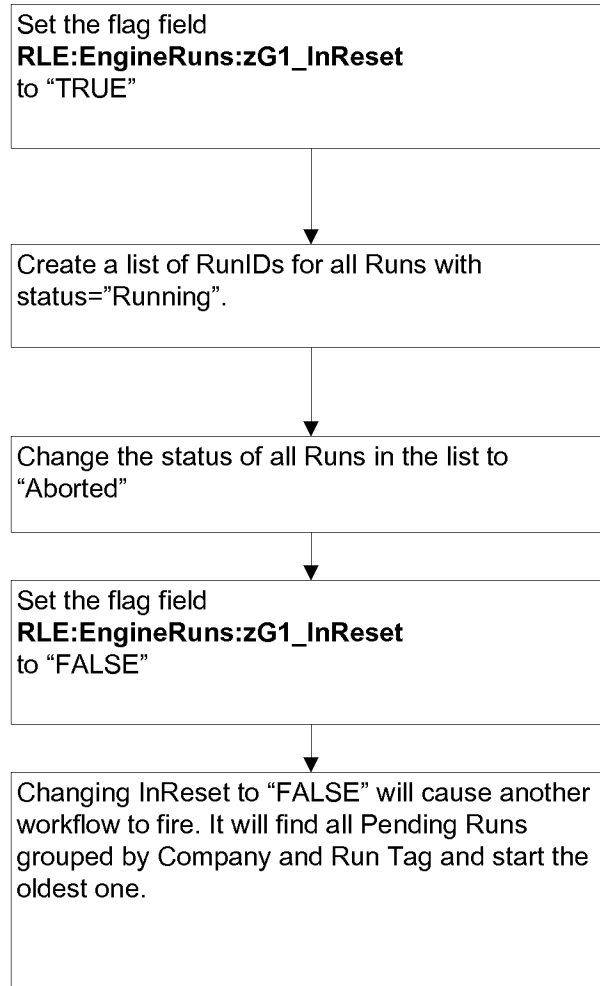
It also should add error messages grouped by Run ID to the **RLE:RunsErrors** form.
 Tom, what do you think about that form structure?

End

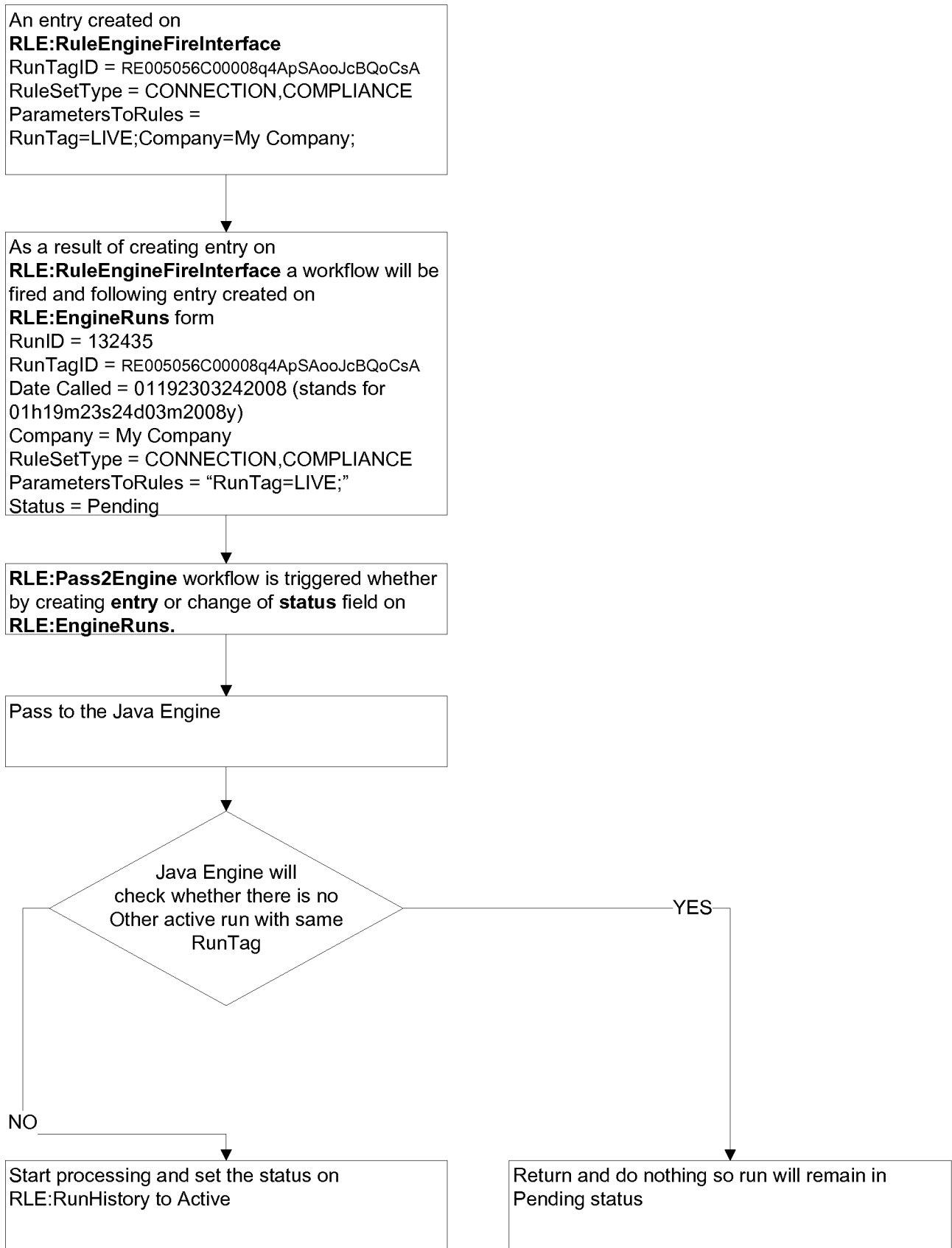
Rule Engine should take care of possible errors and dead locks. Following mechanism resolve a situation where Plug-in Engine been restarted during processing of Run/s or before java has updated the status to running. In that case status will remain Running or Started and will never turned to Completed.

Open issues:

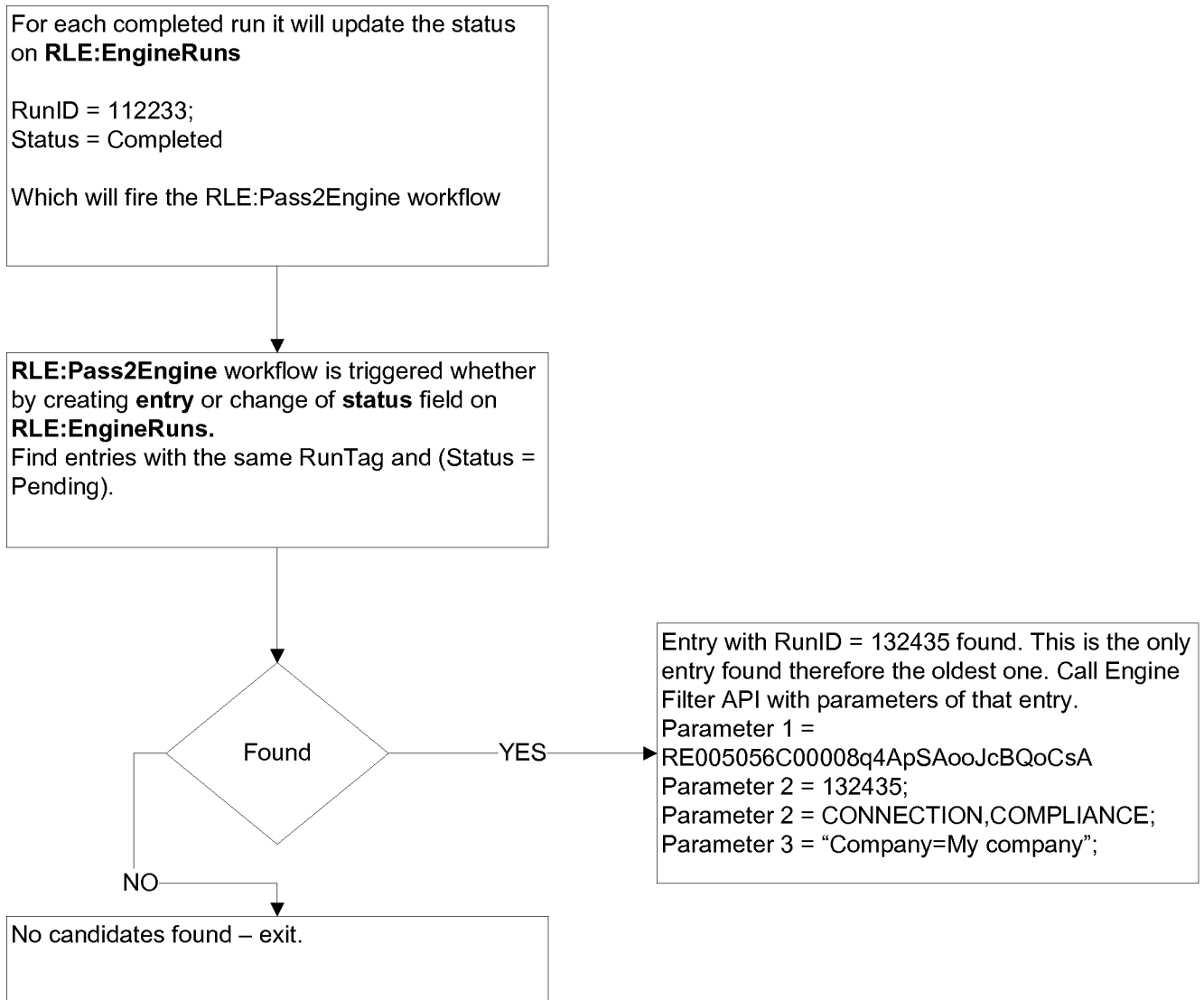
1. Engine will reset all Running Runs to aborted. The question however whether we want it to be canceled or want to put it to the queue again – make it Pending. Tom?



This is the example of how to fire rules engine for company = Target with ran tag = LIVE and rule set types = CONNECTION,COMPLIANCE. In this example another run for same company and same run tag with RunID=112233 is in progress. So this one will be pushed to a queue.

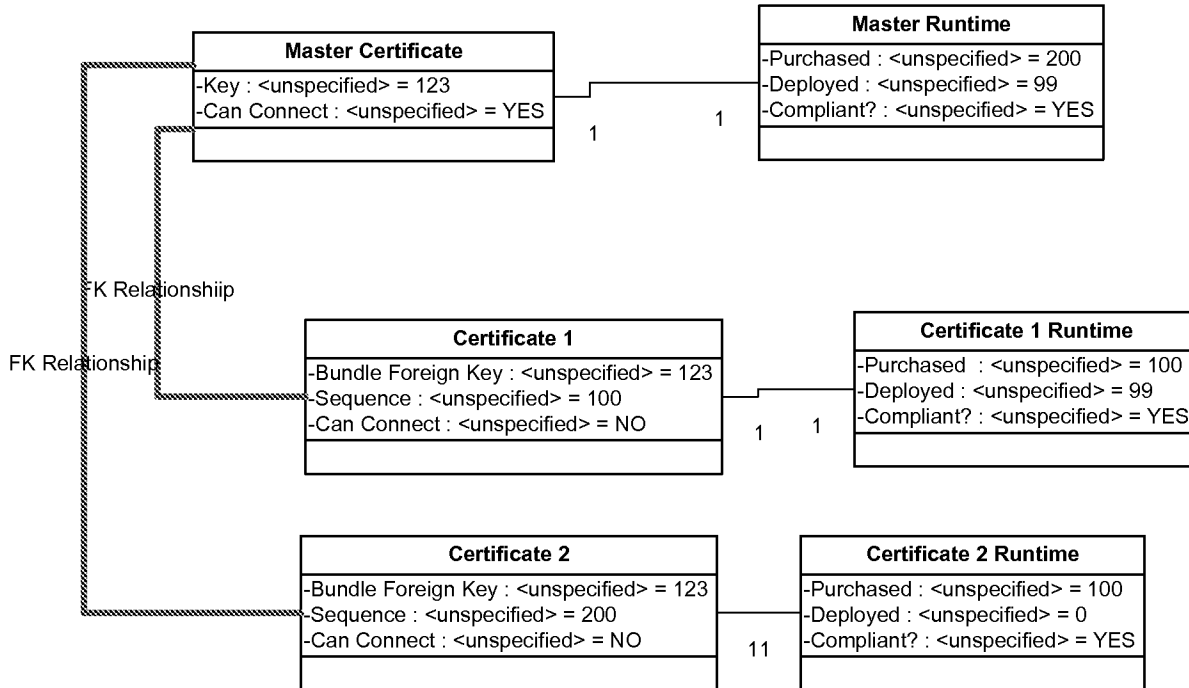


In this example run with RunID=112233 and RunTagID = RE005056C00008q4ApSAooJcBQoCsA has been completed. Engine will change the status to Completed which will cause workflow **RLE:Pass2Engine** to fire. It will find another pending run with the same runtag RunID = 132435 and RunTagID = RE005056C00008q4ApSAooJcBQoCsA. It will pass that run from to the Engine.

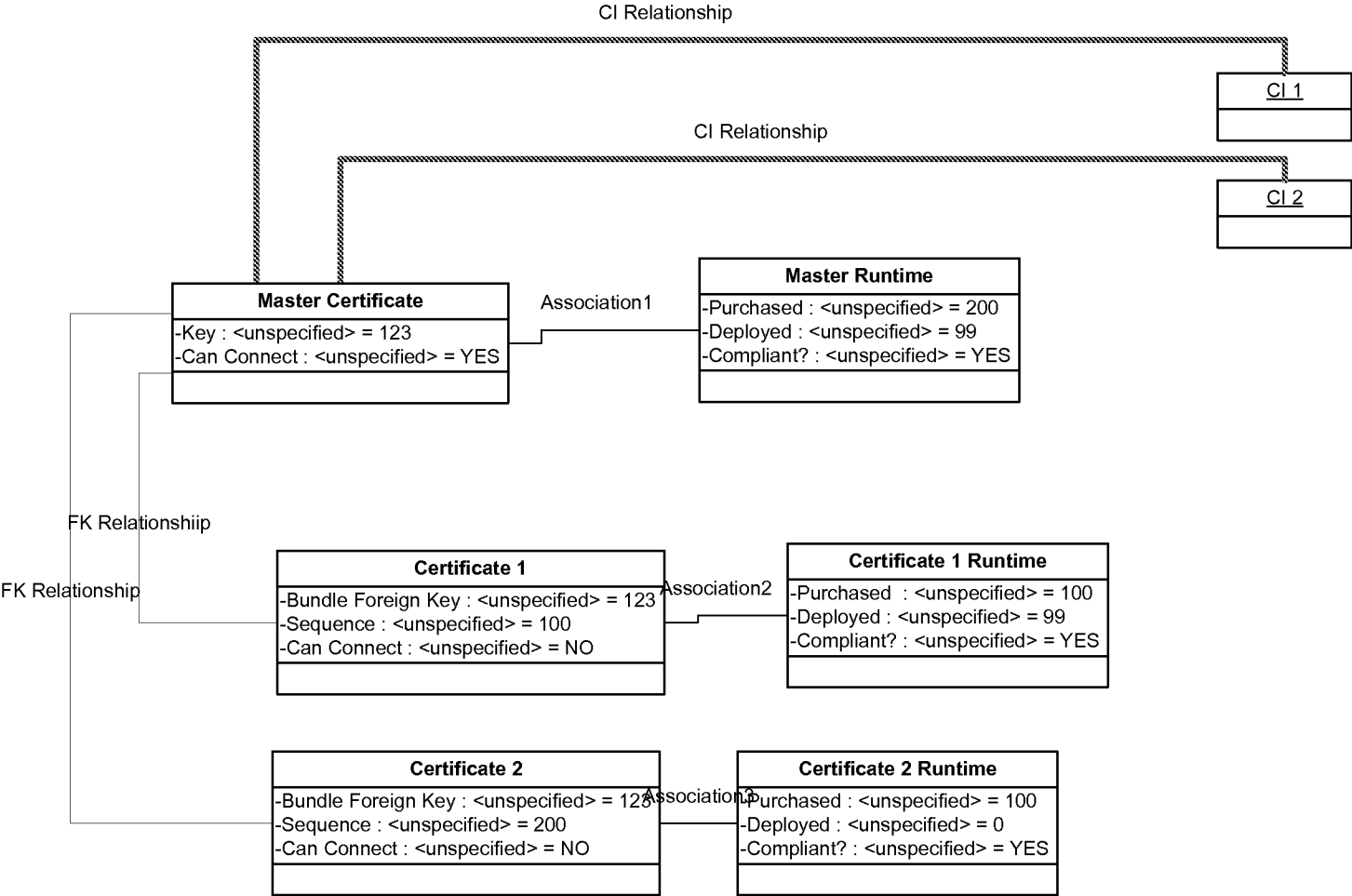


Step 1 – Create Bundle by relating a certificate defined to be a master, and the certificates that are part of the bundle. Each certificate in the bundle will have a Foreign Key to the Master Certificate.

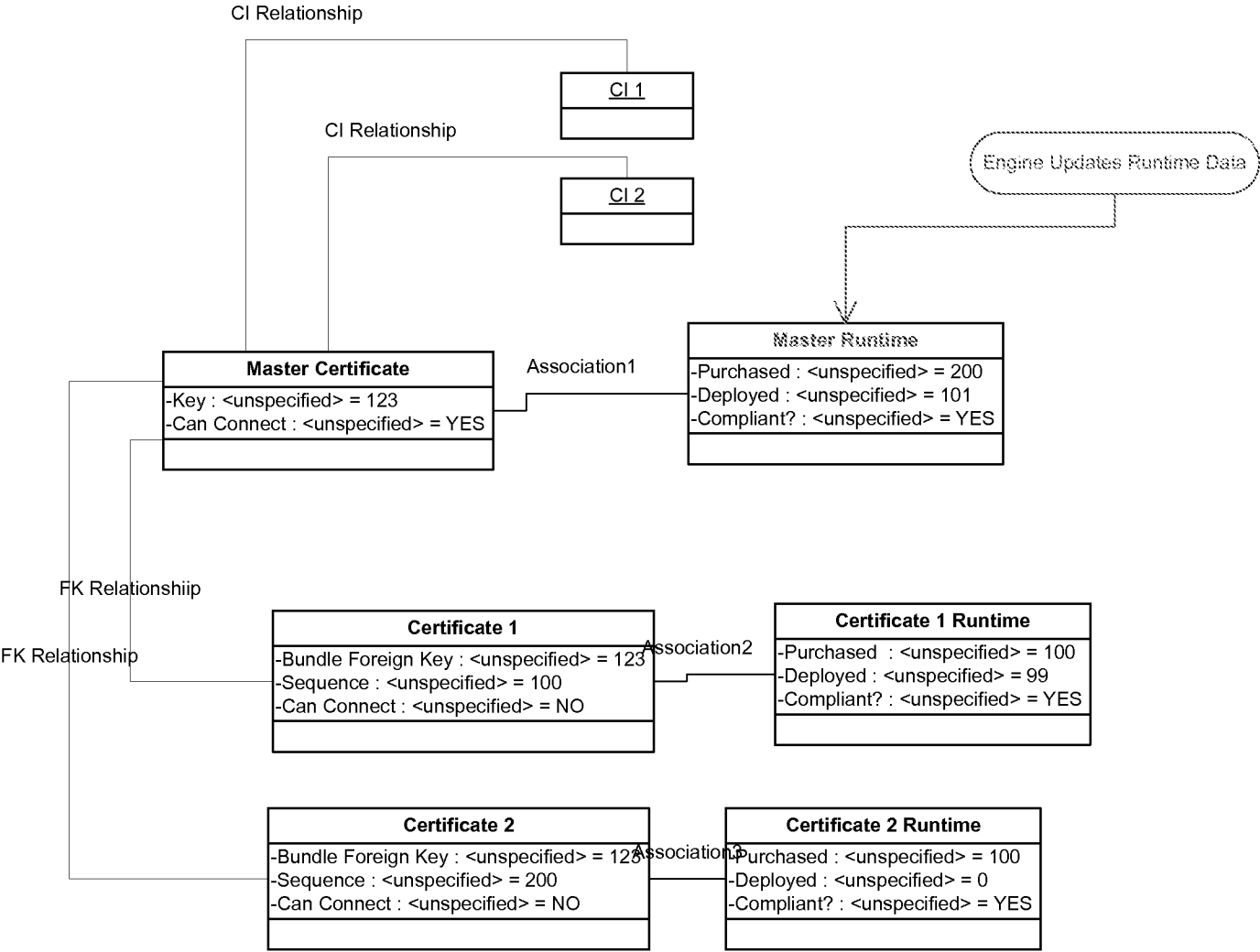
CI 1
CI 2



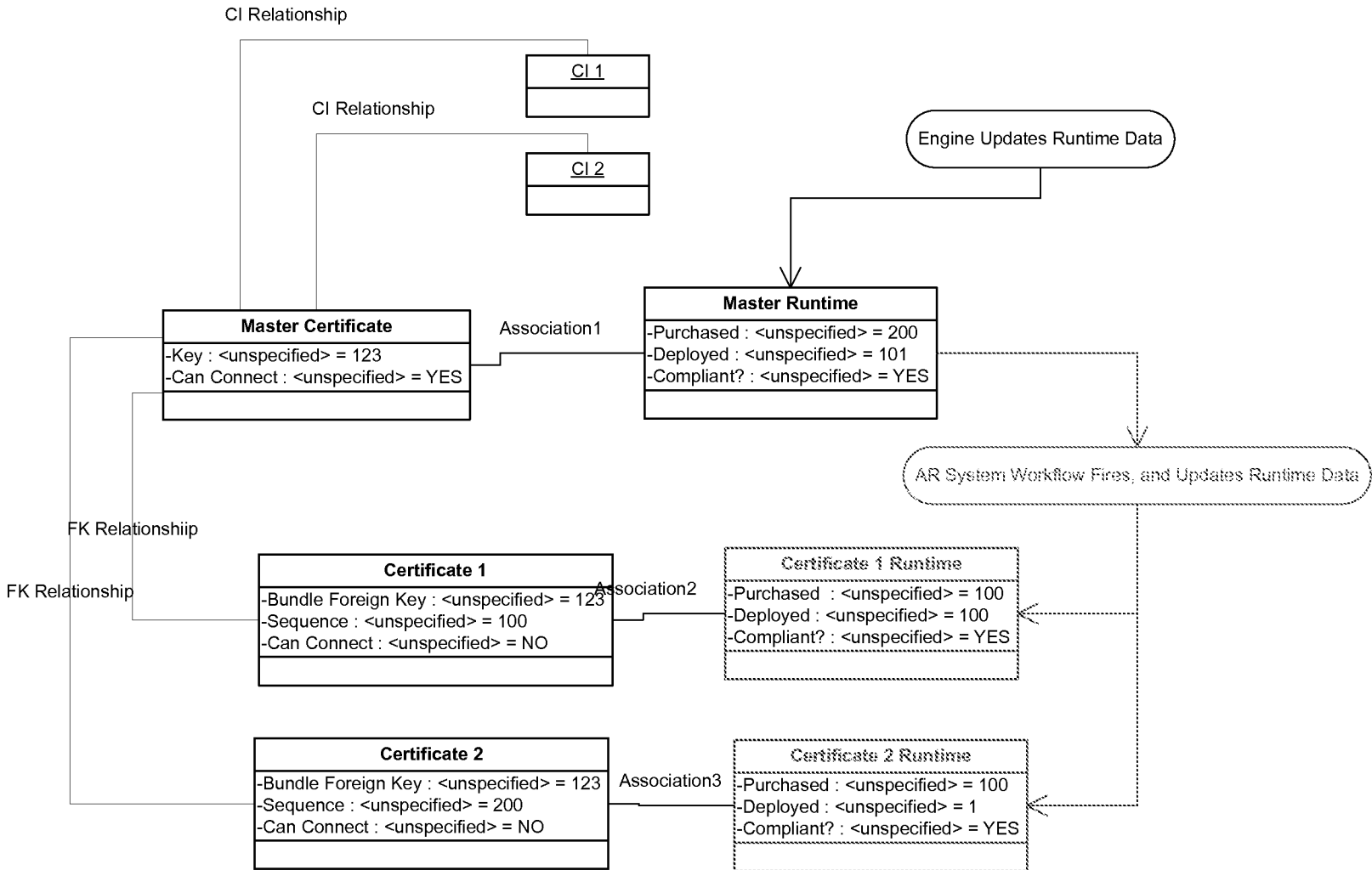
Step 2 – Engine runs through connection rulesets. The query will only pull certificates that have a flag of can connect. Certificate 1 and 2 would be set to N because they are children of the master certificate. Any certificate which is not tied to a parent would be set to Y. So, the engine will only find the master certificate and relate CI 1 and CI 2 to the master certificate as part of the connection processing.



Step 3 – Engine will update the compliance numbers. This will happen at the level of the Master Runtime form. Lets Certificate 1 had 100 licenses purchased on it, and has 99 deployed, and Certificate 2 has 100 licenses on it but 0 are currently deployed. The master runtime will be 200 purchased and 99 deployed before this process. After this process the engine will have updated the Master runtime to be 101 deployed.

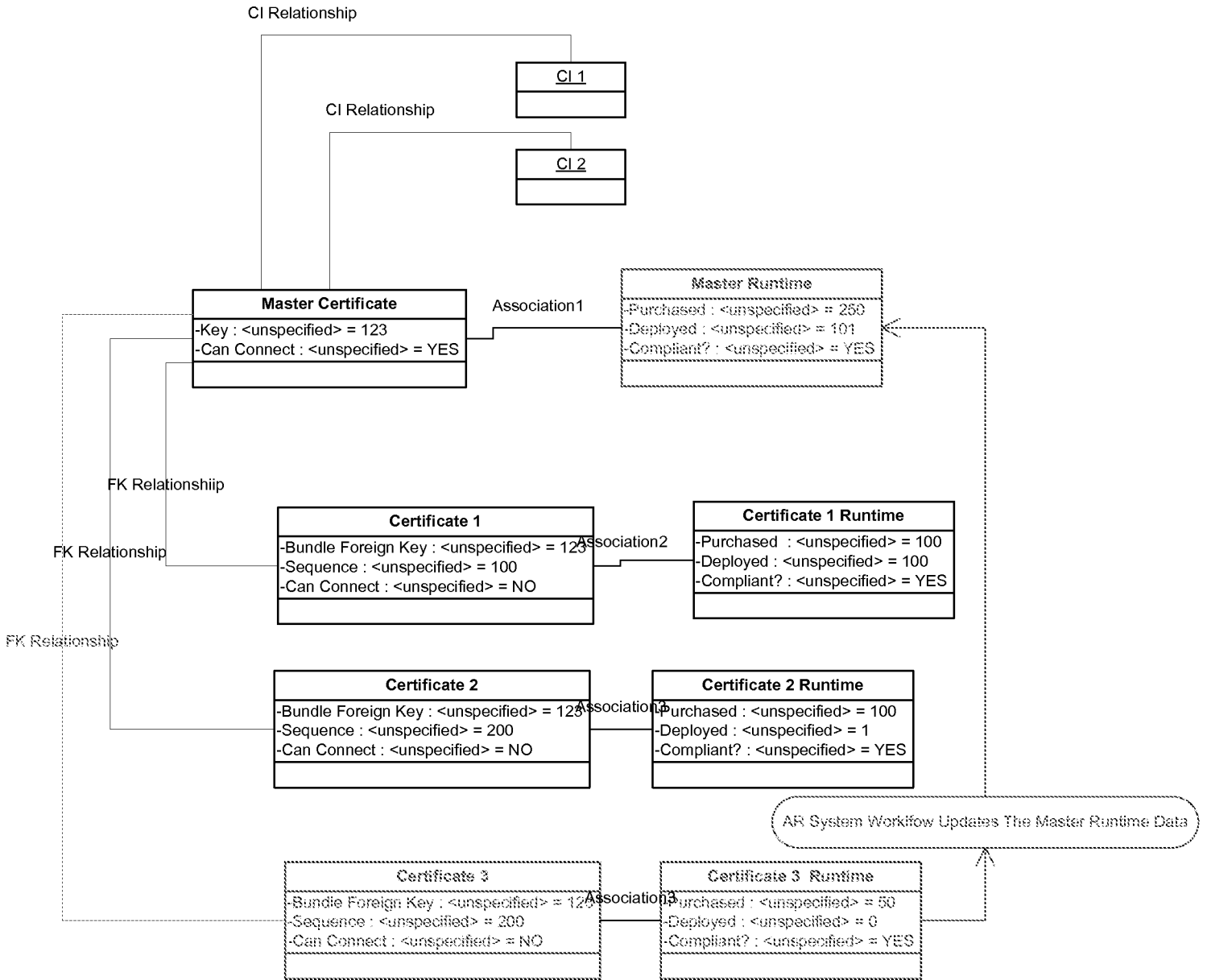


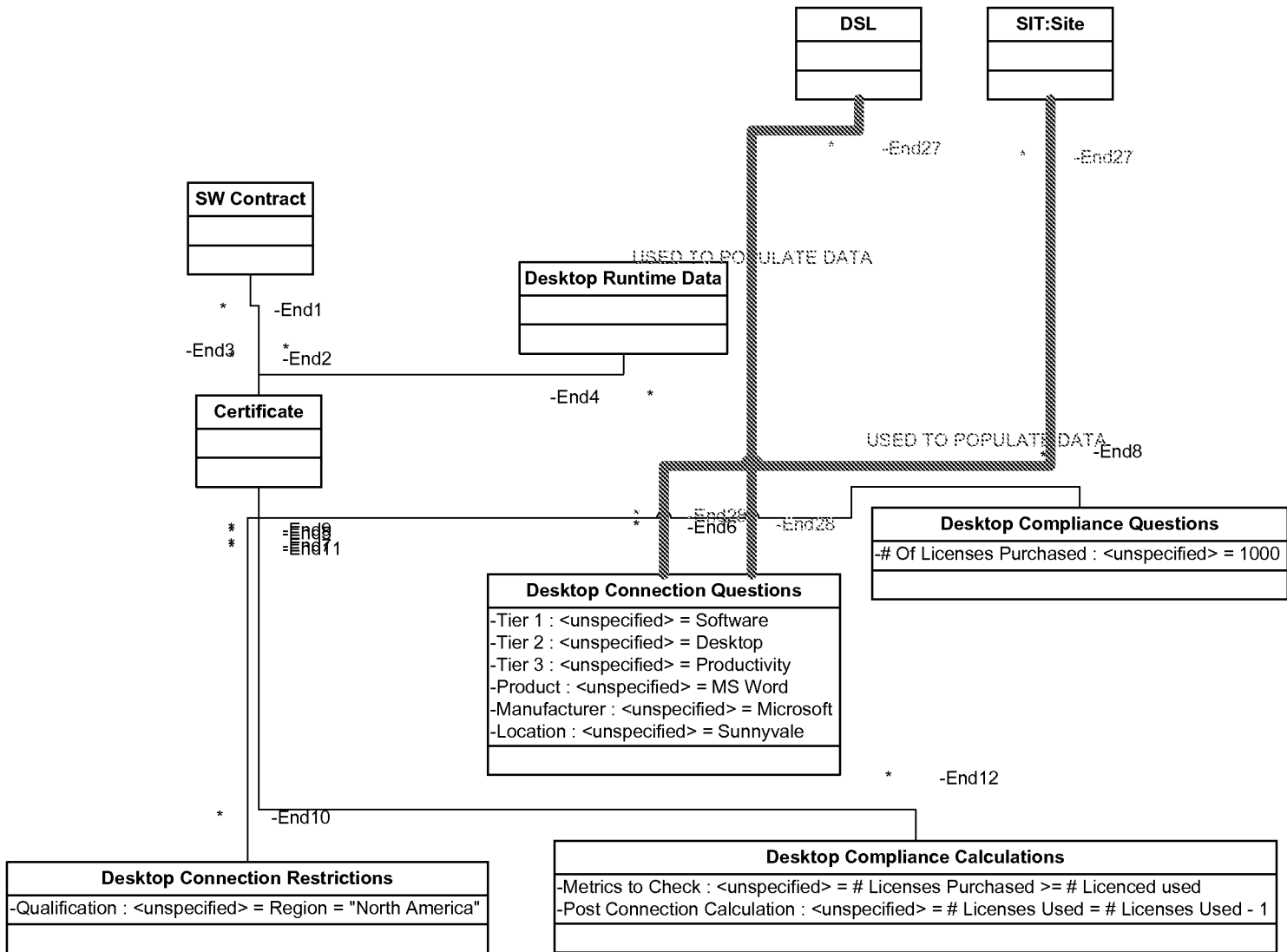
Step 4 – Licenses will be allocated to the connected contract runtime data forms via AR System workflow. Once a certificate is full, then the next certificate will be allocated.



Step 5 – If a new certificate is purchased for 50 licenses, it will need to fall under the same master certificate, and have sequence # higher than the first full certificate. This will be controlled via AR System workflow on the certificate forms.

There will also need to be AR System workflow that takes the Purchased # from the Certificate 3 Runtime form and use it to update the Master Runtime form data.





Exception processing
-Type Of Compliance
-Group to Notify
-Actions to perform

Configuration
-Type : <unspecified> = Desktop
-Connection Question form : <unspecified> = Desktop Connection Questions
-Connection Restriction Form : <unspecified> = Desktop Connection Restrictions
-Compliance Question Form : <unspecified> = Desktop Compliance Questions
-Compliance Calculation Form : <unspecified> = Desktop Compliance Calculations
-Runtime Tracking form : <unspecified> = Desktop Runtime Data

Field Mapping

Configuration	
-Type	
-Connection Question form	
-Restriction Question Form	
-Compliance Question Form	
-Compliance Calculation Form	
-Runtime Tracking form	

Field Mapping	
-Contract Form	
-Contract Field	
-CMDB Class	
-CMDB Attribute Field ID	

Base Contract	
-Contract Number	
-ID	
-Type	
-Status	
-Status Reason	
-Company	
-Org	
-Department	
-Supplier	
-Term	
-Notification Date	
-Termination Date	
-Purchase Date	
-Cost Center	
-Budget Code	
-Project Number	
-Accounting Code	
-Purchase Cost	
-Late Charge	
-Net Days Payable	
-Tax	
-Annual Fee	
-Termination Conditions	

Purchase Line Item	

Costs	

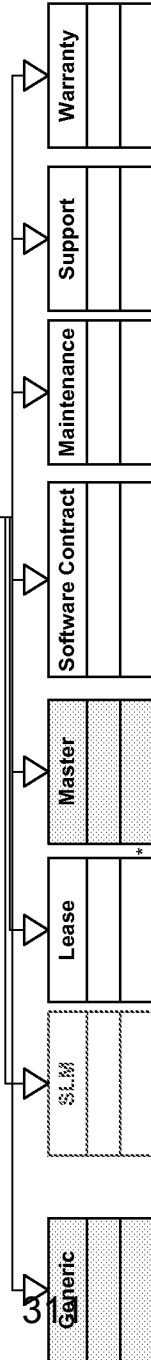
Schedules	
-Description Text Field	
-Attachment Field	

Terms and Conditions	
-Description Text Field	
-Attachment Field	

Rights Granted	
-Description Text Field	
-Attachment Field	

Exception processing	
-Type Of Compliance	
-Group to Notify	
-Actions to perform	

These forms are dynamic based on the configuration of the system. Each Type may have different forms to support it.



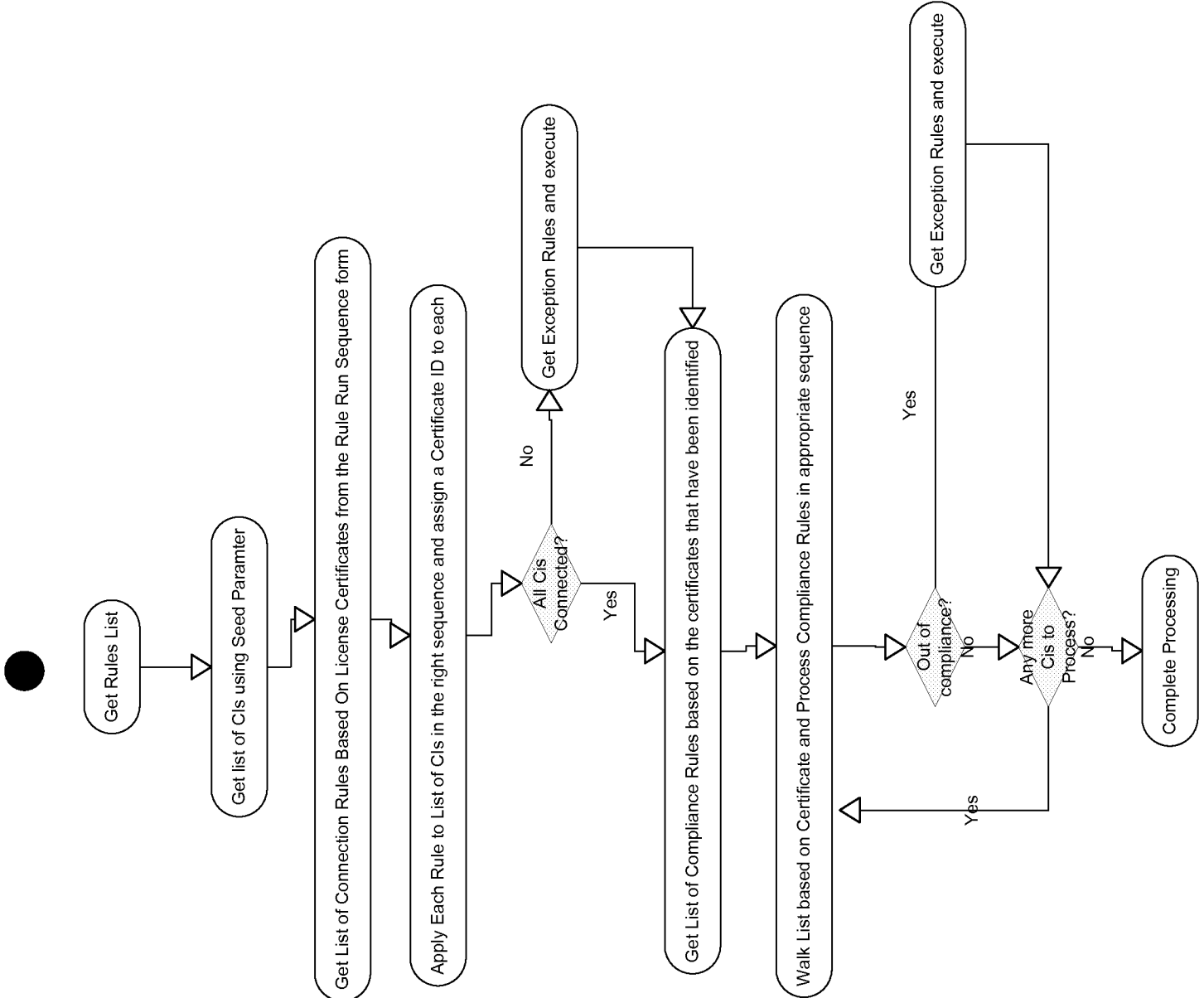
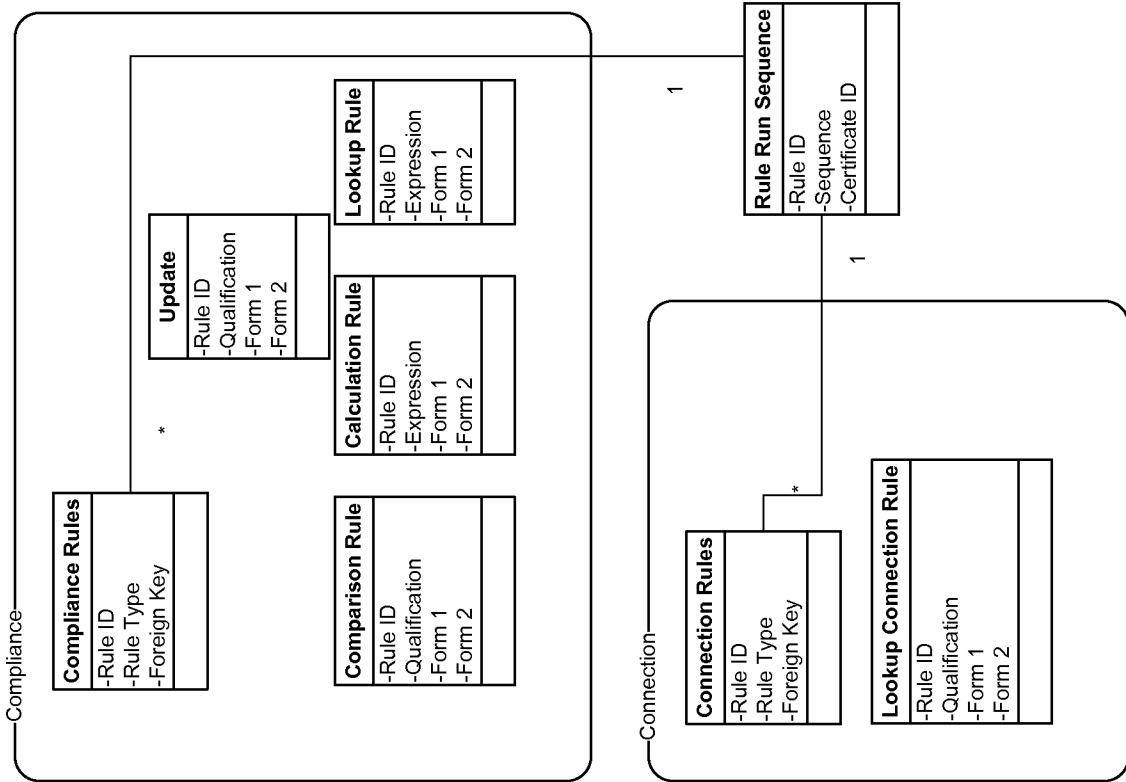
Certificate	
-Type	
-Licenses Purchased	
-PO ID	
-Manufacturer	
-Start Date	
-License key	
-Purchase Date	
-Expiration Date	

Compliance Questions	
-# Of Licenses Purchased	

Connection Questions	
-Tier 1	
-Tier 2	
-Tier 3	
-Product	
-Manufacturer	
-Location	

Compliance Calculations	
-Metrics to Check	
-Post Connection Calculation	

Connection Restrictions	
-Country	



Draft – Contract has never been executed but is going through the process of being executed

Valid Status Reasons:

- In Negotiation – Still negotiating contract
- Pending Signature – Negotiations and review completed but not signed
- Other?

Valid transitions:

- Executed – once executed, status in-effect
- Historical – if the contract was never signed, should be saved as historical with reason of never executed
- Delete – option to move contract right to Delete if it was never Active

Executed – Contract is executed and valid

Valid Status Reasons:

- Active – default status reason (all ok)
- Requires attention (?) – still valid but there has not been activity on the contract for some time (how much – config)
- Under re-negotiation – There is a possible change pending on the contract but still is valid
- Change Pending – there is a change to the contract that is pending – could be an early termination, addendum, amendment, etc...
- On Hold – still valid but no transactions should be performed against it
- Other?

Valid transitions:

- Historical – At any time, expiry date should be completed. Status reason is ?
- ENGINE IS CALLED TO RECALCULATE COMPLIANCE RULES

Historical – Contract is not valid

Valid Status Reasons:

- Expired – Contract has expired
- Terminated – Contract was terminated when effective
- Cancelled – Contract was cancelled before effective
- Other?

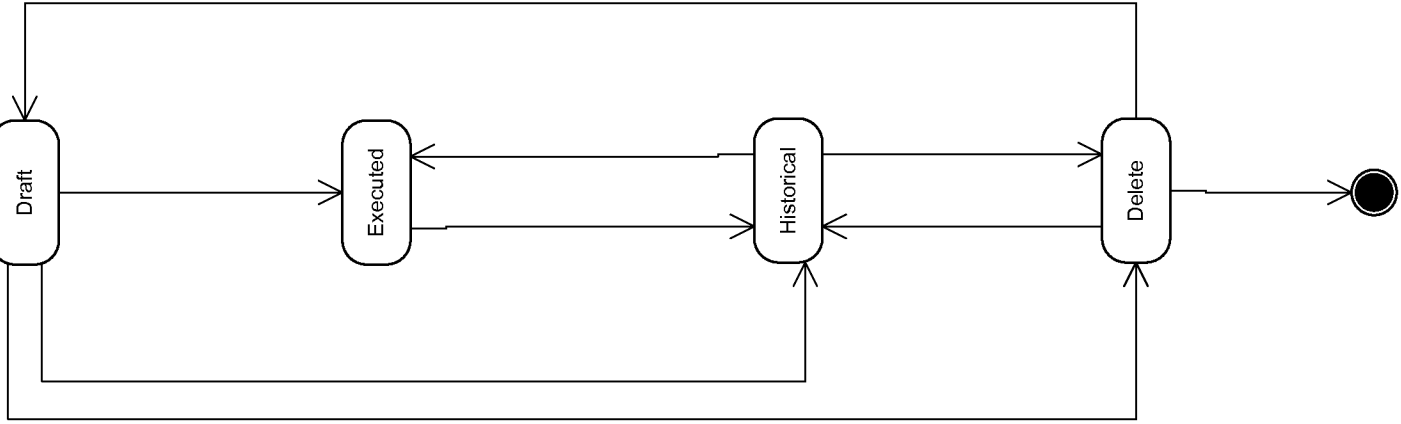
Valid transitions:

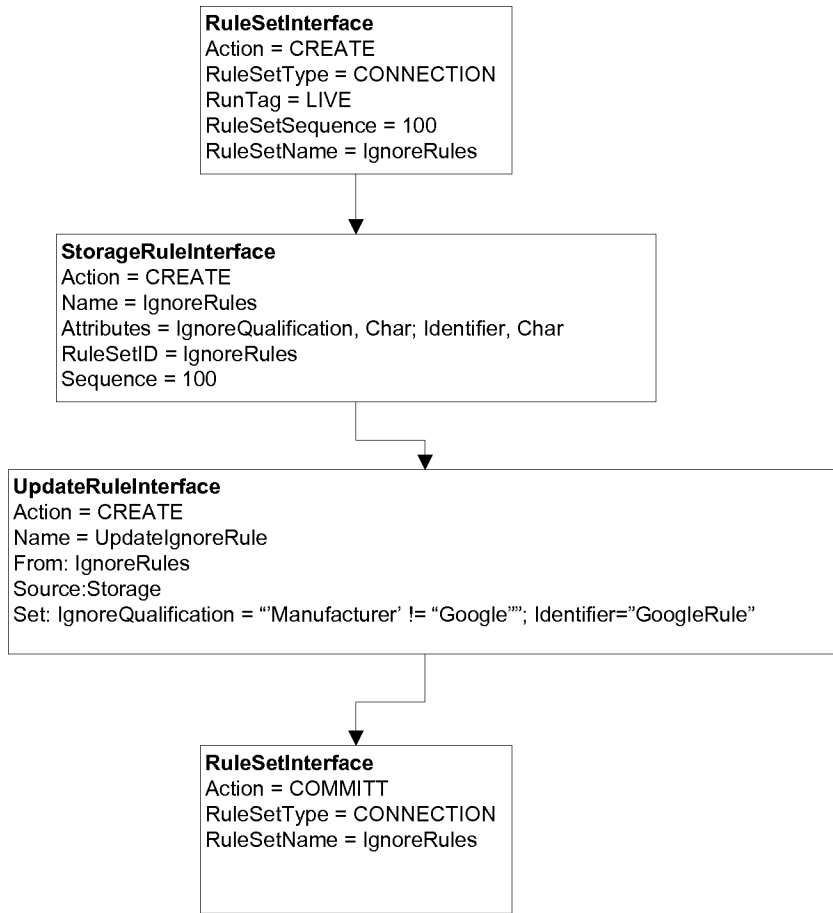
- In Effect – Only when status reason is expired or terminated, default status is ?
- Draft – Only when the status reason is Cancelled
- ENGINE IS CALLED TO RECALCULATE COMPLIANCE RULES

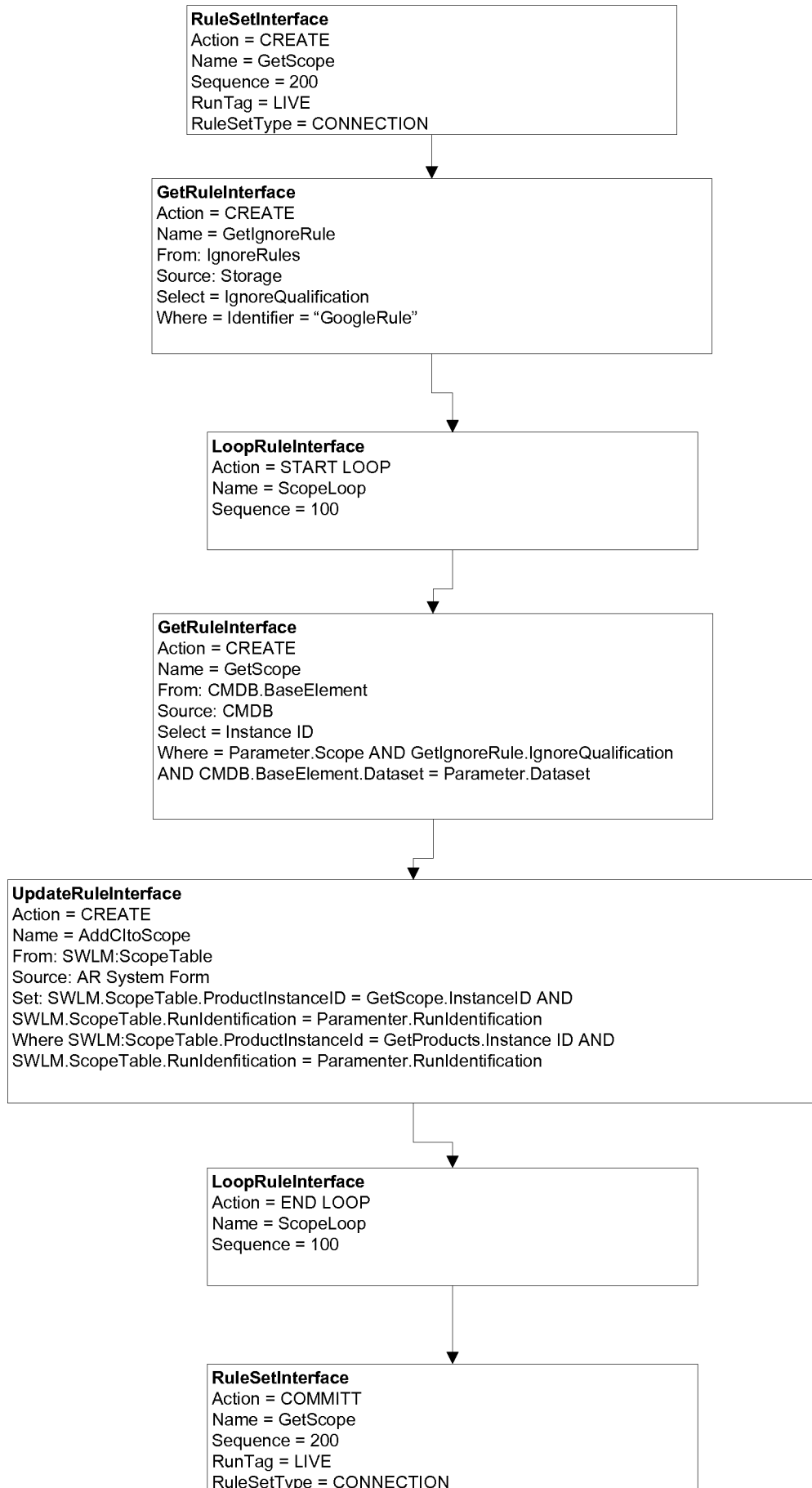
Delete – Contract is Scheduled for deletion

Valid Status Reasons:

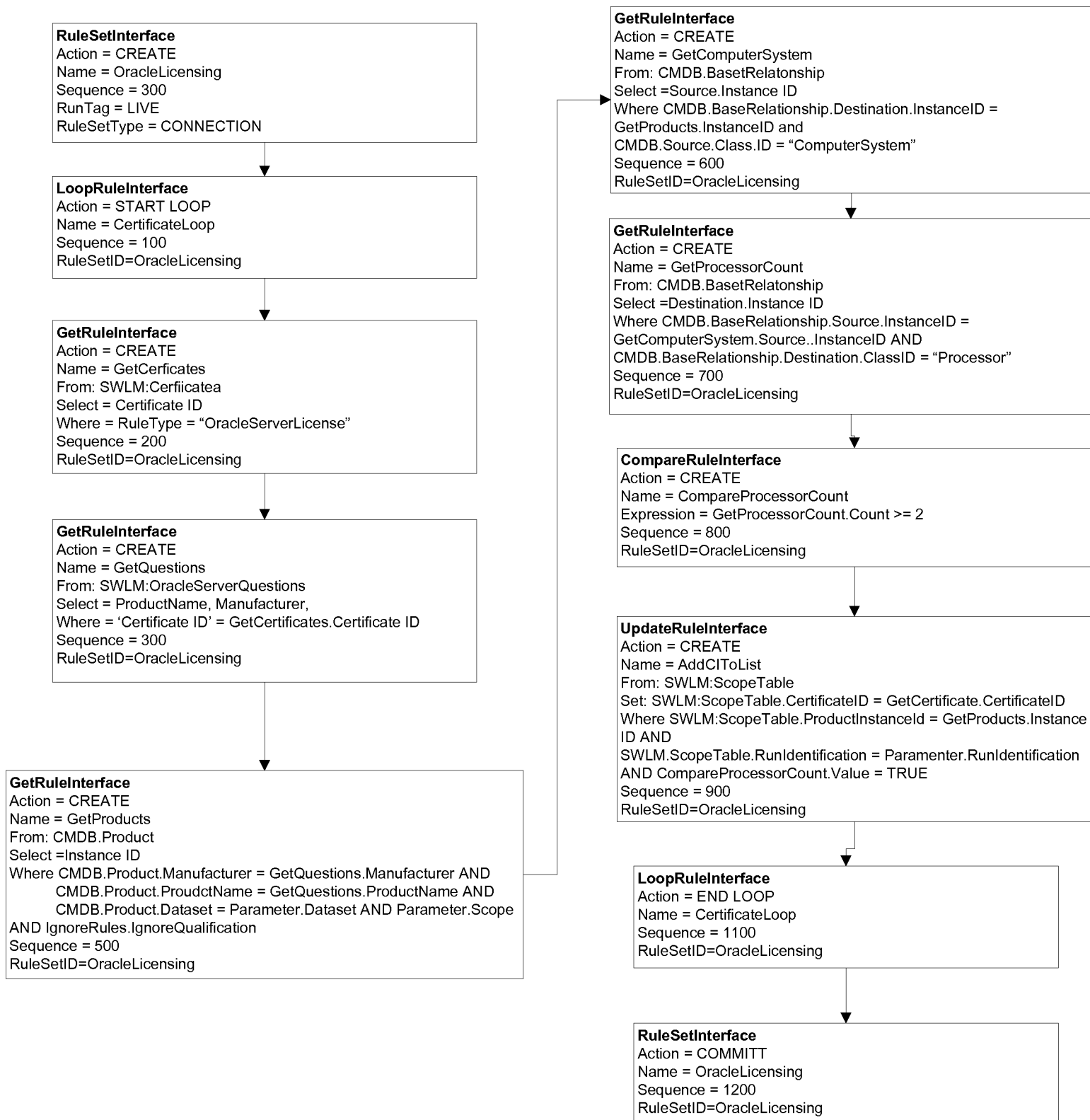
- Scheduled for Deletion – Contract is going to be removed on the next delete run
- Valid transitions:
- Back to previous state – either Historical or Draft







This is an example set of rules to determine licensing for Oracle. In this simple example the goal will be to get all computer systems hosting Oracle that have two Processors. The CMDB will be providing a query language that we should integrate with to make this query much easier. The details around the query language are still being worked out., Without the query language we would need to build a set of queries to get the data. This is represented below.



RuleSetInterface
Action = CREATE
Name = Committ
Sequence =
RunTag = LIVE
RuleSetType = CONNECTION

UpdateRuleInterface
Action = CREATE
Name = CommittData
From: SWLM:ScopeTable
Set: SWLM.ScopeTable.Process = TRUE
Where SWLM.ScopeTable.RunIdentification = Parameter.RunIdentification

UpdateRuleInterface
Action = CREATE
Name = CalculteMetrics
From: SWLM:MetricsCalculation
Set: SWLM.MetricsCalculation.Process = CONNECTIONS AND
SWLM.MetricsCalculation.RunIdentification = Parameter.RunIdentification

AR Workflow Processing on
SWLM:ScopeTable
TR.Process = TRUE

AR Workflow Processing on
SWLM:MetricsCalculation

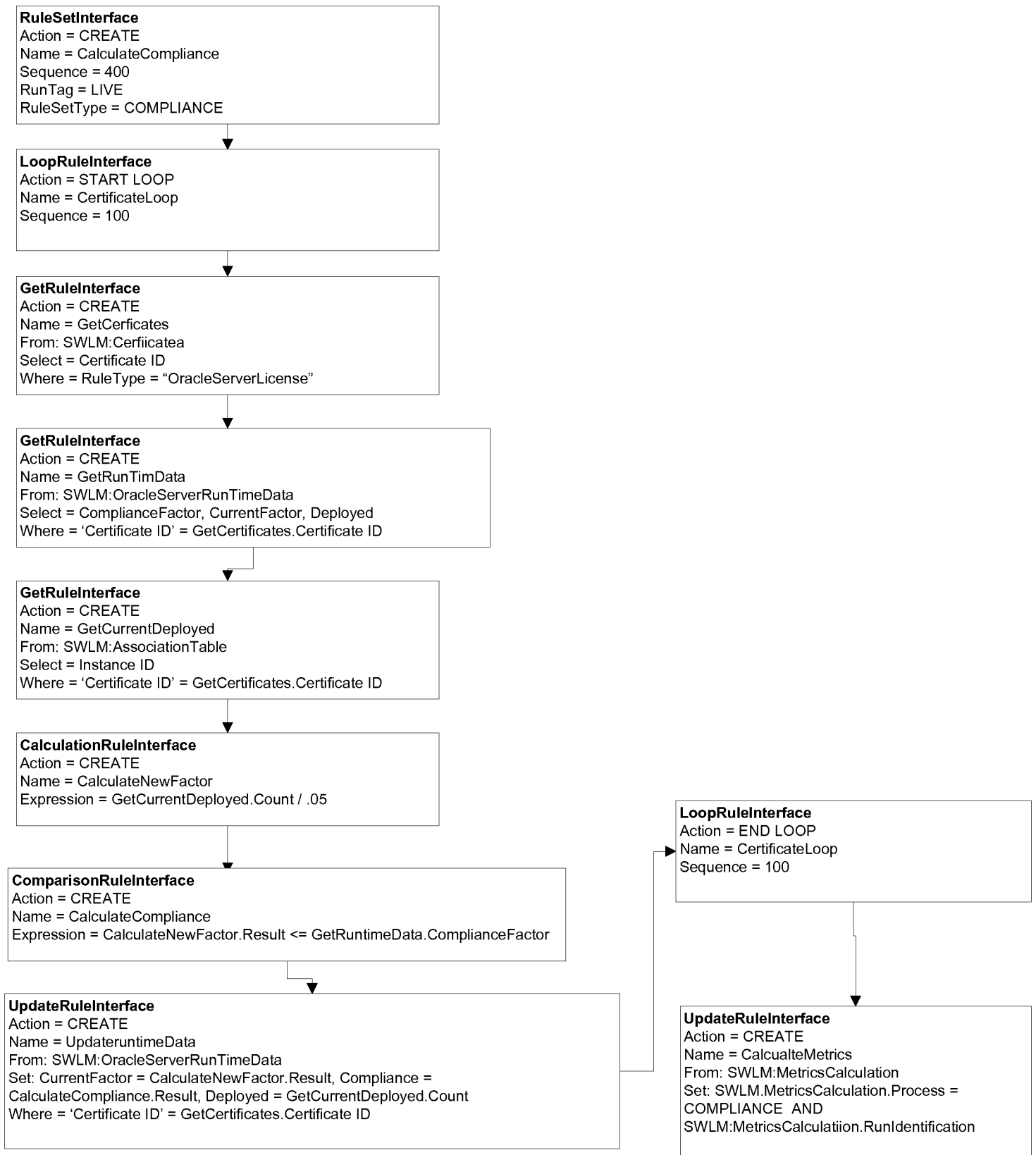
Filter SWLM:ProcessMultipleCertificates
Event: Modify
Form: SWLM:ScopeTable
RunIf: TR.Process = TRUE AND MultipleUpdates = TRUE
Actions
Push Fields to Exception Subsystem ro generate exception

Filter SWLM:NoCertificates
Event: Modify
Form: SWLM:ScopeTable
RunIf: TR.Process = TRUE AND CertificateID = \$NULL\$
Actions
Push Fields to Exception Subsystem ro generate exception

Filter SWLM:ProcessMetrics
Event: Modify
Calculate Data from SWLM:ScopeTable and Generate Events

Filter SWLM:CommittAssociation
Event: Modify
Form: SWLM:ScopeTable
RunIf: TR.Process = TRUE AND CertificateID != \$NULL\$
Actions
PushValueto Association Table

Filter SWLM:SetToProcessed
Event: Modify
Form: SWLM:ScopeTable
RunIf: TR.Process = TRUE
Actions
Set Fields to set Processed Flag = TRUE (Have escalation process which does
garbage collection periodically)



License Type

PerInstanceLicensing

Select Question Fields

Pick a Field

PerInstanceLicensing

Add >>

List of Fields

Get List of CIs

Form

PerInstanceLicensing

Qualification

PerInstanceLicensing

List Of Fields

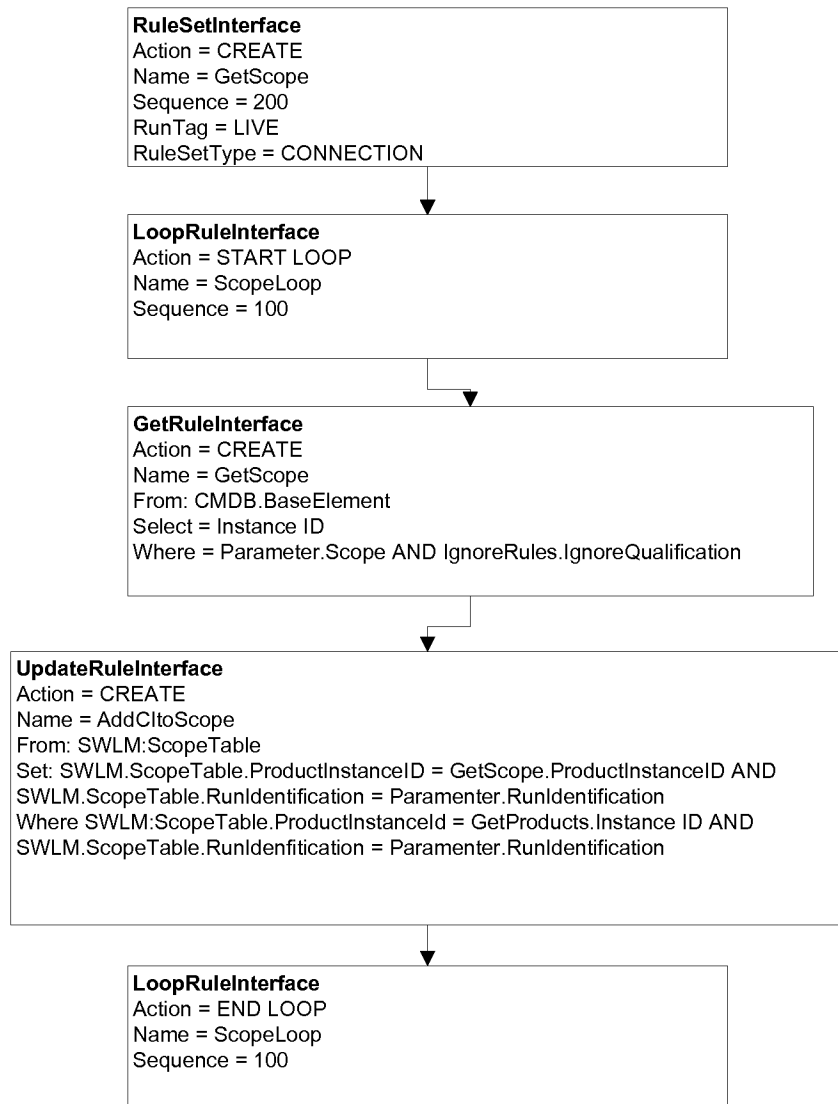
List of Existing Queries

Add New Query

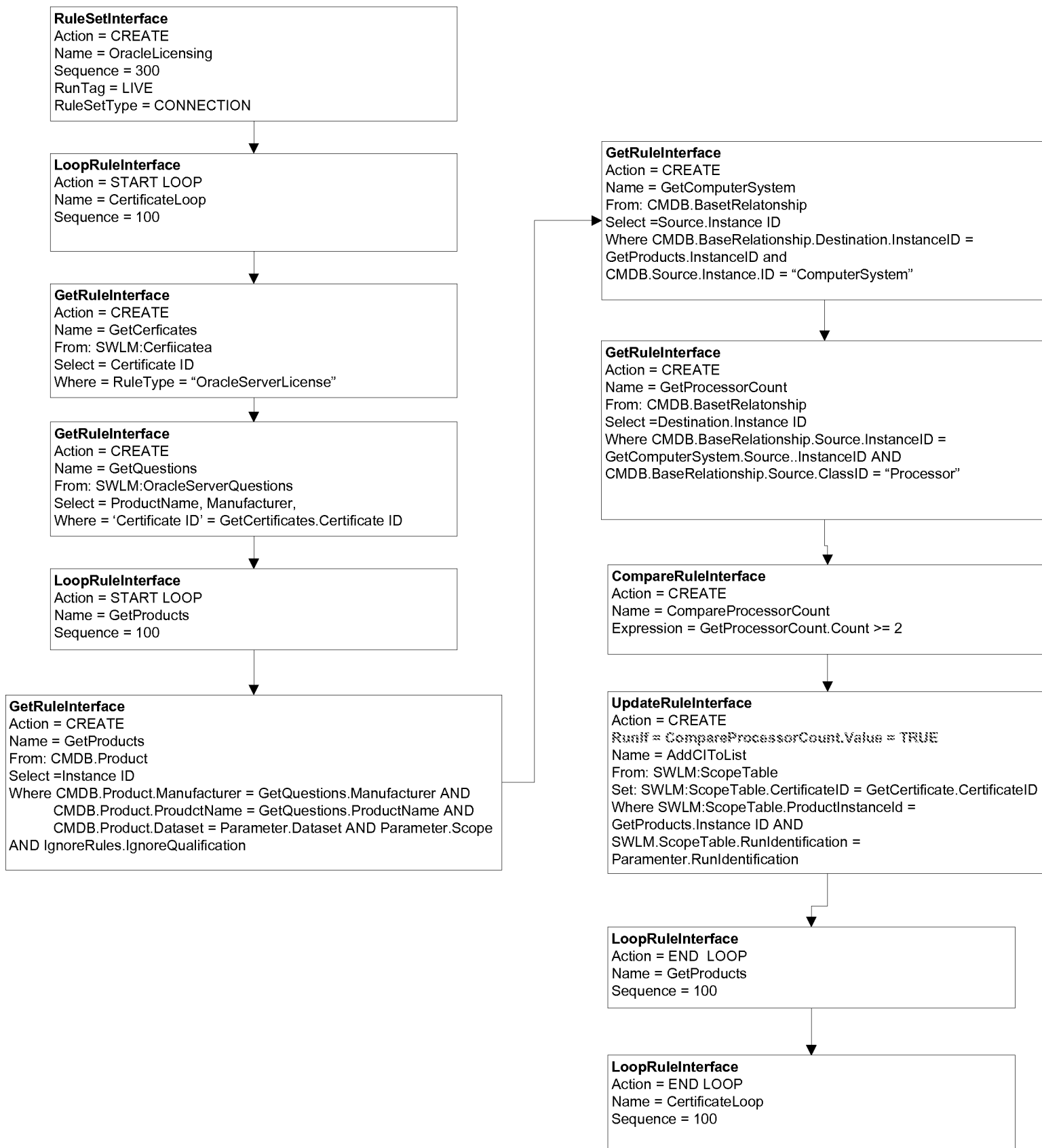
RuleSetInterface
Action = CREATE
RuleSetType = CONNECTION
RunTag = LIVE
RuleSetSequence = 100
RuleSetName = IgnoreRules



StorageRuleInterface
Action = CREATE
Name = IgnoreRules
Attributes = IgnoreQualification='240001003' = "Google"
Sequence = 100



This is an example set of rules to determine licensing for Oracle. In this simple example the goal will be to get all computer systems hosting Oracle that have two Processors. The CMDB will be providing a query language that we should integrate with to make this query much easier. The details around the query language are still being worked out., Without the query language we would need to build a set of queries to get the data. This is represented below.



RuleSetInterface
Action = CREATE
Name = Committ
Sequence = 400
RunTag = LIVE
RuleSetType = CONNECTION

UpdateRuleInterface
Action = CREATE
Name = CommittData
From: SWLM:ScopeTable
Set: SWLM.ScopeTable.Process = TRUE
Where SWLM.ScopeTable.RunIdentification = Parameter.RunIdentification

UpdateRuleInterface
Action = CREATE
Name = CalculteMetrics
From: SWLM:MetricsCalculation
Set: SWLM.MetricsCalculation.Process = CONNECTIONS AND
SWLM.MetricsCalculation.RunIdentification = Parameter.RunIdentification

AR Workflow Processing on
SWLM:ScopeTable
TR.Process = TRUE

AR Workflow Processing on
SWLM:MetricsCalculation

Filter SWLM:ProcessMultipleCertificates
Event: Modify
Form: SWLM:ScopeTable
Runlf: TR.Process = TRUE AND MultipleUpdates = TRUE
Actions
Push Fields to Exception Subsystem ro generate exception

Filter SWLM:ProcessMetrics
Event: Modify
Calculate Data from SWLM:ScopeTable and Generate Events

Filter SWLM:NoCertificates
Event: Modify
Form: SWLM:ScopeTable
Runlf: TR.Process = TRUE AND CertificateID = \$NULL\$
Actions
Push Fields to Exception Subsystem ro generate exception

Filter SWLM:CommittAssociation
Event: Modify
Form: SWLM:ScopeTable
Runlf: TR.Process = TRUE AND CertificateID != \$NULL\$
Actions
PushValueto Association Table

Filter SWLM:SetToProcessed
Event: Modify
Form: SWLM:ScopeTable
Runlf: TR.Process = TRUE
Actions
Set Fields to set Processed Flag = TRUE (Have escalation process which does
garbage collection periodically)

RuleSetInterface
Action = CREATE
Name = CalculateCompliance
Sequence = 400
RunTag = LIVE
RuleSetType = COMPLIANCE

LoopRuleInterface
Action = START LOOP
Name = CertificateLoop
Sequence = 100

GetRuleInterface
Action = CREATE
Name = GetCertificates
From: SWLM:Certificate
Select = Certificate ID
Where = RuleType = "OracleServerLicense"

GetRuleInterface
Action = CREATE
Name = GetRunTimeData
From: SWLM:OracleServerRunTimeData
Select = ComplianceFactor, CurrentFactor, Deployed
Where = 'Certificate ID' = GetCertificates.Certificate ID

GetRuleInterface
Action = CREATE
Name = GetCurrentDeployed
From: SWLM:AssociationTable
Select = Instance ID
Where = 'Certificate ID' = GetCertificates.Certificate ID

CalculationRuleInterface
Action = CREATE
Name = CalculateNewFactor
Expression = GetCurrentDeployed.Count / .05

ComparisonRuleInterface
Action = CREATE
Name = CalculateCompliance
Expression = CalculateNewFactor.Result <= GetRuntimeData.ComplianceFactor

UpdateRuleInterface
Action = CREATE
Name = UpdateruntimeData
From: SWLM:OracleServerRunTimeData
Set: CurrentFactor = CalculateNewFactor.Result, Compliance = CalculateCompliance.Result, Deployed = GetCurrentDeployed.Count
Where = 'Certificate ID' = GetCertificates.Certificate ID

LoopRuleInterface
Action = END LOOP
Name = CertificateLoop
Sequence = 100

UpdateRuleInterface
Action = CREATE
Name = CalculateMetrics
From: SWLM:MetricsCalculation
Set: SWLM:MetricsCalculation.Process = COMPLIANCE AND SWLM:MetricsCalculation.RunIdentification

License Type

PerInstanceLicensing

Select Question Fields

Pick a Field

PerInstanceLicensing

Add >>

List of Fields

Get List of CIs

Form

PerInstanceLicensing

Qualification

PerInstanceLicensing

List Of Fields

List of Existing Queries

Add New Query

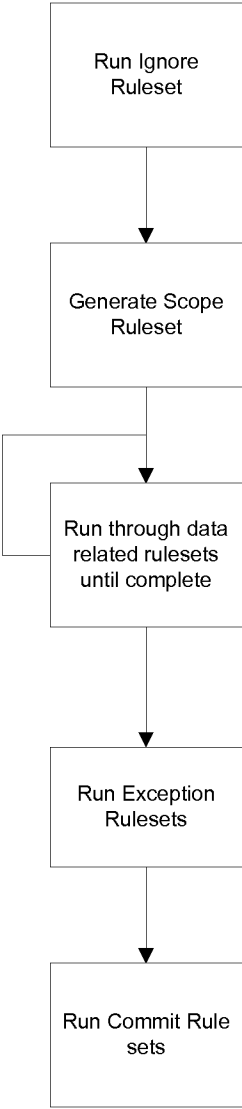
Engine Rule Processing

The assumptions here are that there will be multiple rule sets, and each rule set will have a sequence. The processing that will be done is the following:

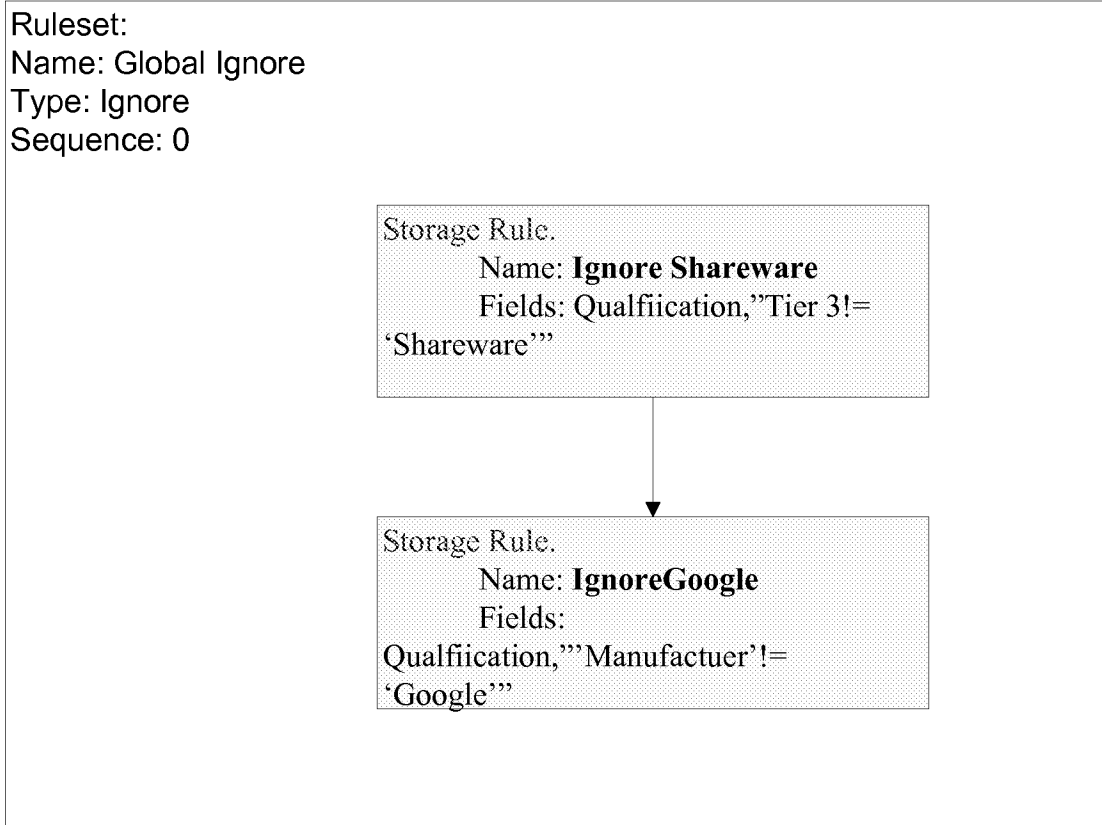
Connection Rules

- There will be a rule set that is run to apply the ignore rules qualification to memory.
- There will be a rule that is built dynamically based on the data passed in to start the engine. This rule will define the scope qualification.
- There will be a rule set that will take the scope and ignore rules from memory and make use of them to query the CMDB to gather the list of CIs that will be evaluated. This list of CIs will be committed to memory.
- After all of the setup rules have run, the engine will process the rules sets for each of the different types of licenses.
- Each ruleset will need to take the results and find the corresponding item in memory and then tag that CI with the certificate ID.
- After all the rule sets have been run, there will be a rule set for dealing with exceptions. This ruleset would iterate through the list of CIs in memory, and any which have multiple certificates related, would get pushed to the exception subsystem.
- Finally there will be a ruleset to commit the relationships to the database. This will walk the list of CIs in memory, and push the data to the relationships table.

High Level Process Flow For Connection Rules



Set of rules which provide the qualifications to be used for ignore purposes.



Scope Rules are dynamically built as a result of passed in parameters... if that is the case then a question is do we really need a separate rule, or can this be just a memory variable

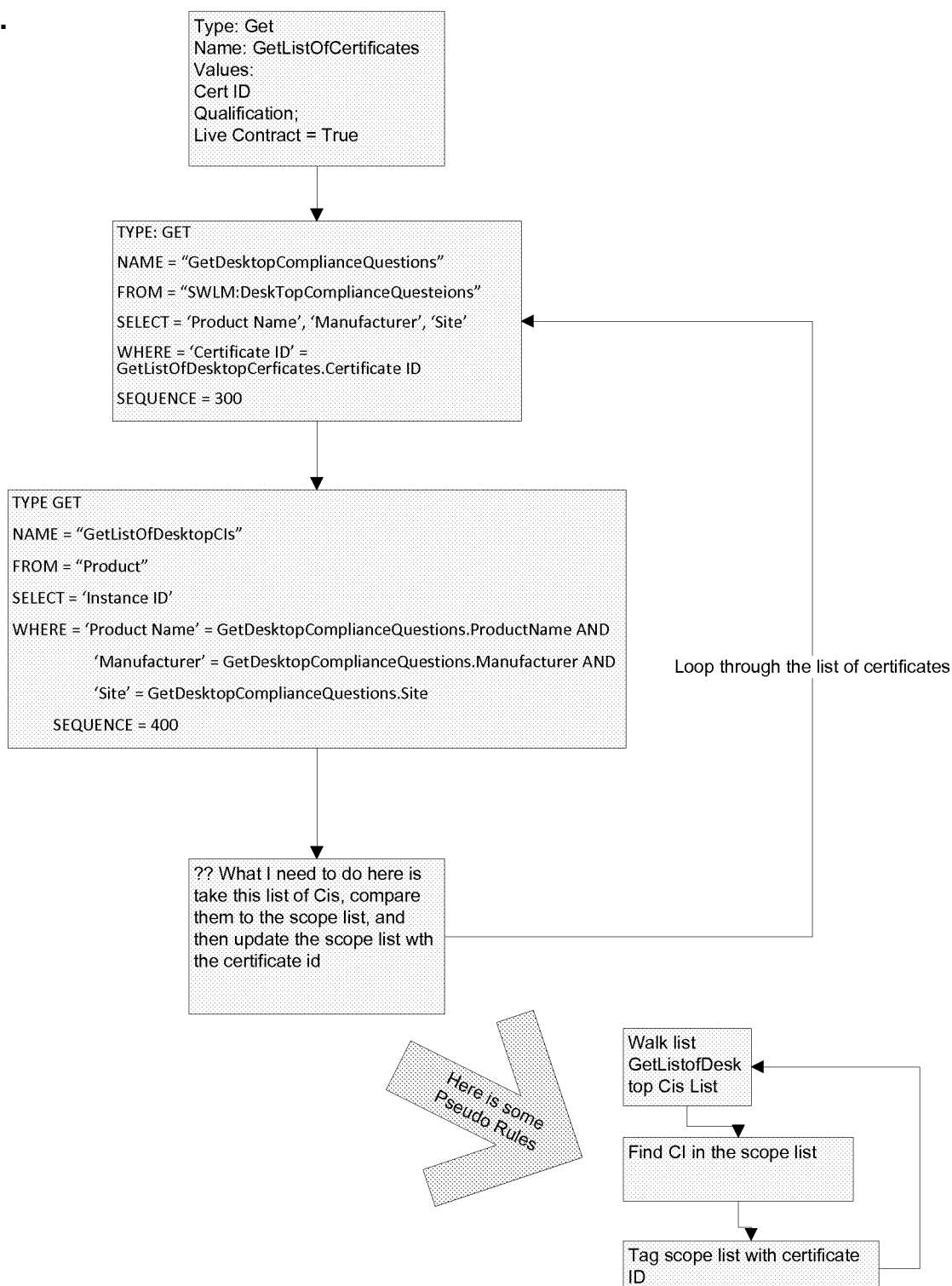
In the rule below Parameters.Scope is a qualification that is passed into the engine.

Ruleset:
Name: Scope
Type: Scope
Sequence: 0

Storage Rule.
Name: **Scope**
Fields:
Qualfication,Parameters.Scope

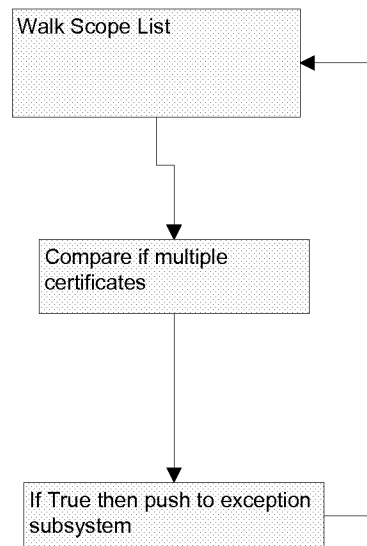
The data connection rulesets are the rule sets which do the work of walking through the certificates, and getting lists of CIs which can be connected to certificates.

If we have a status value which indicates a live contract, we can do the grouping. The way this would work is that on the certificate there would be an attribute which would indicate which contracts can be attached to. Any certificate not in a grouping that is active will be set so that you can attach to it. If a certificate is in a grouping, then only one of the certificate can be attached to. What will happen is that we will have a compliance rule that will set this flag once a certificate has met it's compliance.

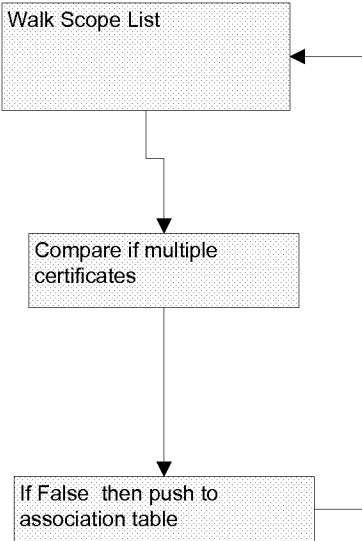


For exception handling we will need to have a way to walk the memory list, and then any records which have multiple certificates associated with them would be pushed to the exception subsystem

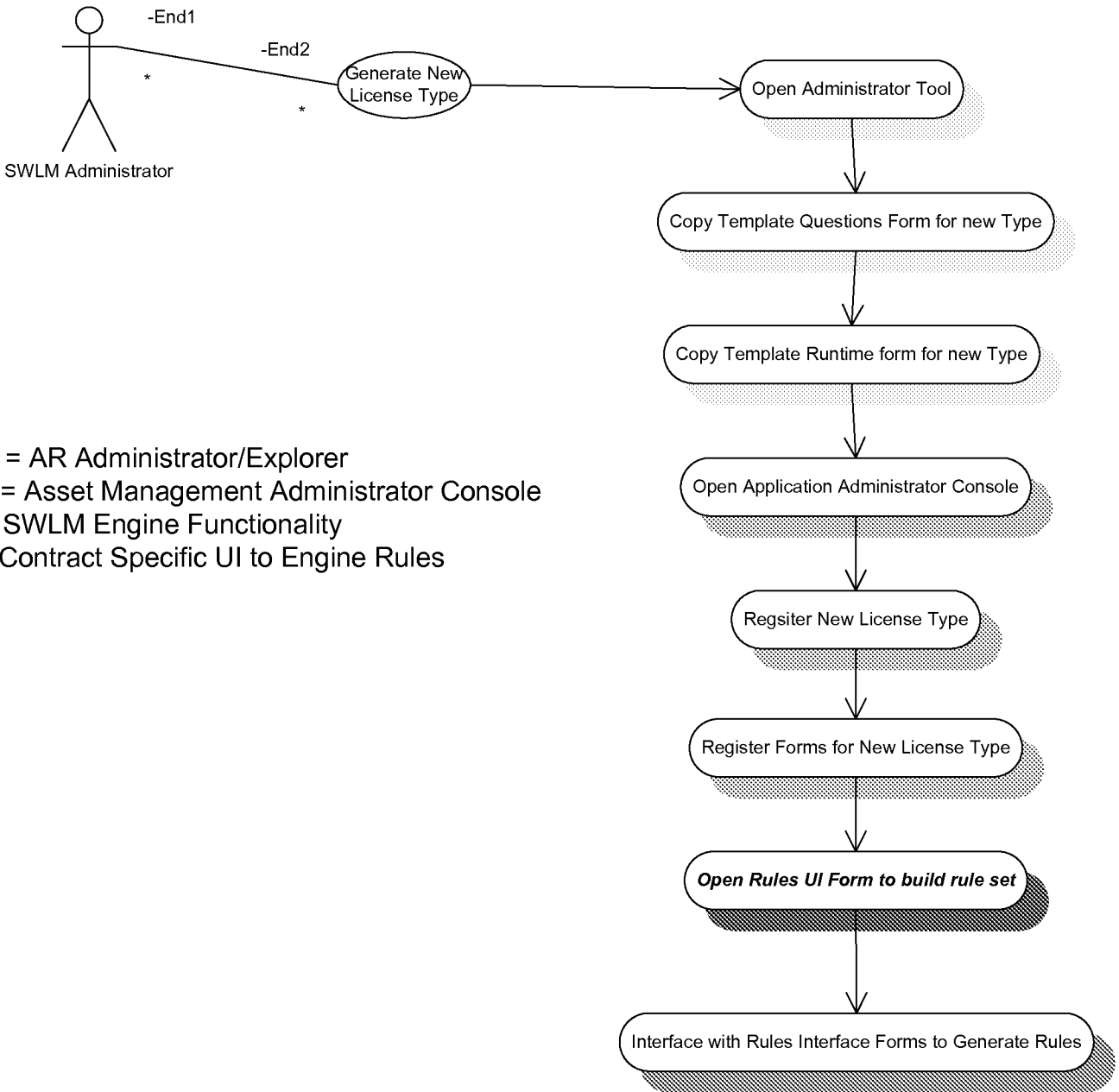
We need to have some type of rule which can read a list from storage.

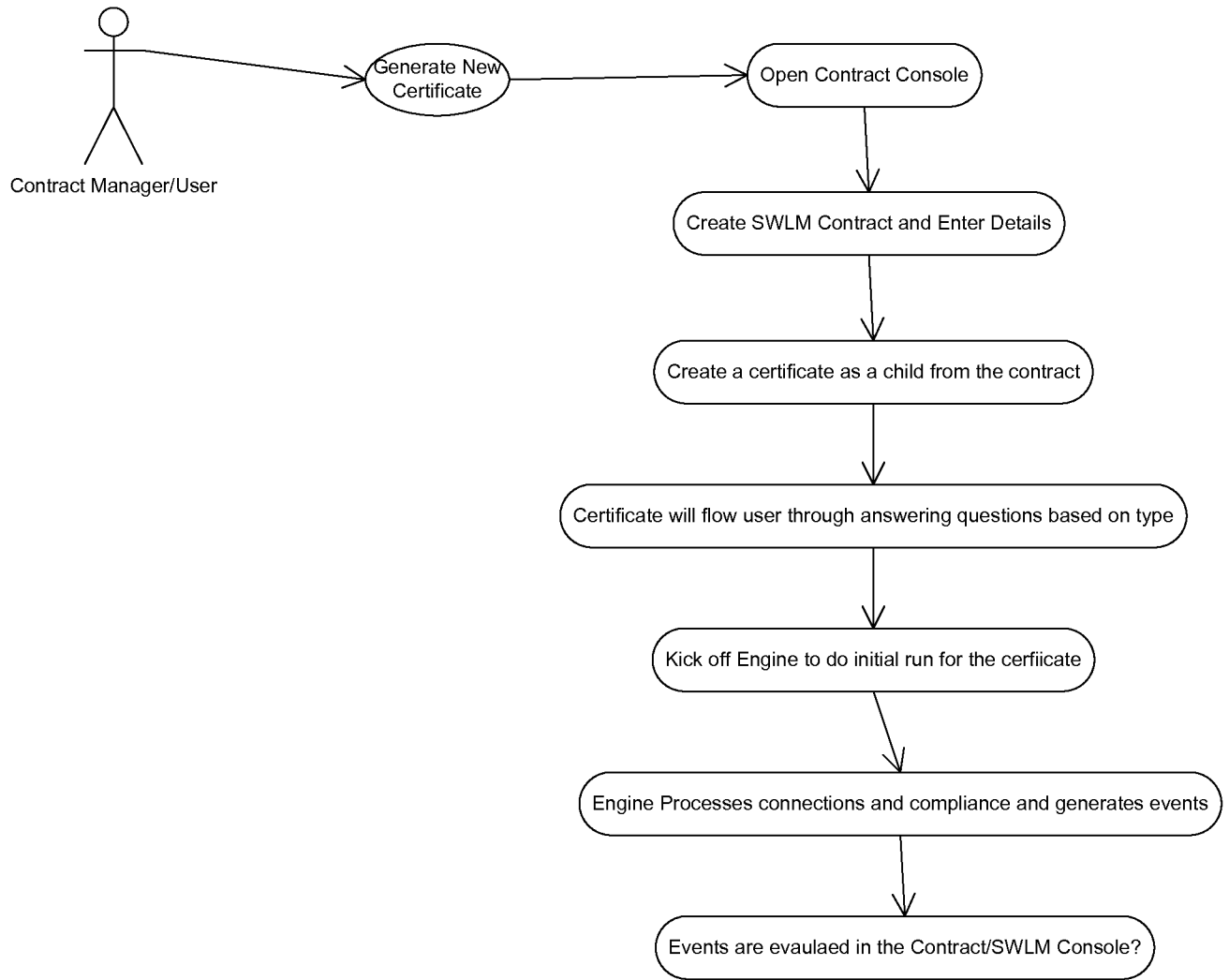


Similar to exceptions, we will need to walk through the storage list, and do a push rule to create/update/delete the relationships.



New License Type Use Case





Software Certificate

General Data

<input type="text" value="Enter Text"/>	<input type="text" value="Enter Text"/>
<input type="text" value="Enter Text"/>	<input type="text" value="Enter Text"/>

Questions

<input type="text" value="Enter Text"/>	<input type="text" value="Enter Text"/>
<input type="text" value="Enter Text"/>	
<input type="text" value="Enter Text"/>	

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	A Method and System for Configuration Management Database Software License Compliance
First Named Inventor/Applicant Name:	Thomas Adrian
Filer:	William M. Hubbard/Melissa Mosby
Attorney Docket Number:	149-0215PUS

Filed as Large Entity

Provisional Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Provisional application filing	1005	1	220	220

Pages:

Prov. Appl Size fee per 50 sheets >100	1085	3	270	810
--	------	---	-----	-----

Claims:

Miscellaneous-Filing:

Petition:

Patent-Appeals-and-Interference:

Post-Allowance-and-Post-Issuance:

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				1030

Electronic Acknowledgement Receipt

EFS ID:	5074024
Application Number:	61165505
International Application Number:	
Confirmation Number:	3848
Title of Invention:	A Method and System for Configuration Management Database Software License Compliance
First Named Inventor/Applicant Name:	Thomas Adrian
Customer Number:	29855
Filer:	William M. Hubbard/Melissa Mosby
Filer Authorized By:	William M. Hubbard
Attorney Docket Number:	149-0215PUS
Receipt Date:	31-MAR-2009
Filing Date:	
Time Stamp:	23:49:07
Application Type:	Provisional

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$1030
RAM confirmation Number	5301
Deposit Account	501922
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part / Zip	Pages (if appl.)
338	ServiceNow's Exhibit No. 1007				

1	Provisional Cover Sheet (SB16)	ProvisionalCoverSheet.pdf	32690 140acb67e24025f29b104bfe583bae9fa9e7b	no	3
Warnings:					
This is not a USPTO supplied Provisional Cover Sheet SB16 form.					
Information:					
2	Specification	ApplicationPart1.pdf	1778890 4eea530aa5cf6d9644604c59808c743011963ca4	no	94
Warnings:					
Information:					
3	Specification	ApplicationPart2.pdf	2129194 75baff91f3730694d7d92368c14d69ad4fa7ee94	no	163
Warnings:					
Information:					
4	Specification	ApplicationPart3.pdf	1148602 0bac485299b9714f2df37e89a758e2ca1d5ea40b	no	27
Warnings:					
Information:					
5	Specification	ApplicationPart4.pdf	347496 443af7085446348738acab5ea5dc68aa95182003	no	45
Warnings:					
Information:					
6	Fee Worksheet (PTO-06)	fee-info.pdf	31378 c780fa567464a790f4dc42472473eb27c63835d4	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			5468250		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.