Service Manager :: System Service Descriptor (from ECG)

Status	IN PROGRESS
Completion (%)	80%
Owner	Sergeev, Alexey (A.)

1. Nested Pages

- ACL Permission Denial Warning Analysis (from ECG)

2. Introduction

Service Manager uses data provided in a descriptor file to create Services, FTCP Cpmmand Hand;ers and ACL databases to

1. Make FTCP command handlers discoverable by FCI
2. Allow SOA clients to communicate to each other.

3. Descriptor content

Descriptor XM schema description.

3.1. Service ID

Service ID must be an FNV wise unique string. Service developers are supposed to add service IDs to a database to make then discoverable by other developers.

3.2. Service Group ID

Service group ID attribute allows to group services to allow consumers to use a single <soa-service-request> element to request FULL access to all services in the group.

Example:

FCI declares "FNV_ECG_FCI" service group, FCI test app requests access to the group.

Service group ID should be an FNV wise unique string.

Services from different components cannot belong to the same group.

3.3. FCI Client

Example

FCI client "id" should match "uniqueServiceId" parameter of FciClient::createInstance() API.

Command "id" should not appear in different descriptors more than once across all ECUs.

3.4. IPPT Client

Example

The IPPT client ID must match the "featureId" of IpptClient constructor.

3.5. Access Type

Service Manager currently do not support "RPC" and "DATA" access type. It ignores the attribute and grants "FULL" service access. The feature support is coming soon. Descriptor can declare different access types.

3.6. Consumer "id" attribute vs Consumer "endpoint" element.

"id" attribute and "endpoint" element are mutually exclusive. A descriptor cannot contain both. If "id" attribute explicitly provided, Service Manager generates a topic: ""SERVICES/RESPONSE/**ECU_ID**/**UID**/**CONSUMER_ID**".

where.

ECU_ID is an id of the consumer's ECU

UID is a user id of the consumer's client (process)

CONSUMER_ID is the "id" attribute value

This topic pattern allows to reduce a probability of duplicate topics.

"id" attribute can be leveraged if

1. Consumer topic follows the pattern
2. Consumer was created using the next SOA API. "id" attribute value should match the "consumerId" parameter value.

```
        /**
         * Factory method for creating a SoaConsumer object
         * @param messageManager a shared pointer to a SoaMessageManager instance
         * @param serviceDirectoryManager a shared pointer to a SoaServiceDirectoryManager instance
         *                                 can be a nullptr if the consumer does not need access to Service
Manager Services
         *                                 Supported Service Manager APIs:
         *                                   getCommandHandlers
         *                                   getCommandHandlersAsync
         *                                 Not supported Service Manager APIS
         *                                   requestServiceStatusCached
         *                                   requestServiceStatusAsync
         *                                   requestServiceStatus
         *                                   subscribeToServiceStatusUpdateAsync
         *                                   subscribeToServiceStatusUpdate
         *                                   unsubscribeToServiceStatusUpdateAsync
         *                                   unsubscribeToServiceStatusUpdate
         *                                   startService
         *                                   startServiceAsync
         * @param consumerId a consumer ID. Must be unique string for the component
         * @return  a shared pointer to a new SoaConsumer instance
         *          a nullptr if
         *              1. messageManager is a nullptr
         *              2. serviceDirectoryManager is a nullptr
         *              3. consumerId is an empty string
         */
        static SharedPtr createSoaConsumer(
                const SoaMessageManager::SharedPtr& messageManager,
                const SoaServiceDirectoryManager::SharedPtr& serviceDirectoryManager,
                const std::string& consumerId);
```

3.7. Shared Services

3.8. SERVICE_MANAGER Service request

FCI Test App requests access to SERVICE_MANAGER service.

Component developers should do that if they create an instance of SoaServiceDirectoryManager to pass it to SoaConsumer::createSoaConsumer(...) facrtory method.

However latest SOA code allows to pass a nullptr instead of a SoaServiceDirectoryManager instance if a component do not need to access SERVICE_MANAGER service.

Looks like now only FCI uses SERVICE_MANAGER service to discover FTCP command handlers.

If you do not create a SoaServiceDirectoryManager instance in your componenet code you do not need to request access to SERVICE_MANAGER service.

4. Descriptor Example

4.1. Service Manager Descriptor

Example of platform component providing services.

Provides two RPC services:

1. SERVICE_MANAGER_LIFECYCLE
2. SERVICE_MANAGER

```
<?xml version="1.0" encoding="utf-8"?>
<platform-client-descriptor  uid="61" version="1.0.0">
  <services>
    <service id="SERVICE_MANAGER_LIFECYCLE">
      <request-endpoint>SERVICES/REQUEST/FNV/SOA/SERVICEMANAGER/SERVICELIFECYCLE</request-endpoint>
    </service>
    <service id="SERVICE_MANAGER">
      <request-endpoint>SERVICES/REQUEST/FNV/SOA/SERVICEMANAGER</request-endpoint>
    </service>
  </services>
</platform-client-descriptor>
```

4.2. FCI Descriptor

Another one example of platform component providing services.and registering consumers.

```
<?xml version="1.0" encoding="utf-8"?>
<platform-client-descriptor uid="190" version="1.0.0">
  <services>
    <service id="FNV_ECG_FCI_GENSERVICE" group-id="FNV_ECG_FCI">
      <request-endpoint>SERVICES/REQUEST/FNV/FCI/GENSERVICE</request-endpoint>
      <data-endpoints>
        <data-endpoint>SERVICES/DATA/FNV/FCI/BROADCAST</data-endpoint>
        <data-endpoint>SERVICES/DATA/FNV/FCI/SMS_BROADCAST</data-endpoint>
      </data-endpoints>
    </service>
    <service id="FNV_ECG_FCI_FTCP_RESPONSE" group-id="FNV_ECG_FCI">
      <request-endpoint>SERVICES/REQUEST/FNV/FCI/FTCP/RESPONSE</request-endpoint>
    </service>
    <service id="FNV_ECG_FCI_FTCP_ALERT" group-id="FNV_ECG_FCI">
      <request-endpoint>SERVICES/REQUEST/FNV/FCI/FTCP/ALERT</request-endpoint>
    </service>
    <service id="FNV_ECG_FCI_FTCP_QUERY" group-id="FNV_ECG_FCI">
      <request-endpoint>SERVICES/REQUEST/FNV/FCI/FTCP/QUERY</request-endpoint>
    </service>
  </services>
  <soa-permissions>
    <soa-consumer-permissions>
      <endpoint>SERVICES/RESPONSE/FNV/FCI/CONSUMER</endpoint>
      <soa-service-requests>
        <soa-service-request service-id="SERVICE_MANAGER" access="RPC"/>
      </soa-service-requests>
    </soa-consumer-permissions>
    <soa-consumer-permissions>
      <endpoint>SERVICES/RESPONSE/FNV/FCI/SERVICEDIR</endpoint>
      <soa-service-requests>
        <soa-service-request service-id="SERVICE_MANAGER" access="RPC"/>
      </soa-service-requests>
    </soa-consumer-permissions>
  </soa-permissions>
</platform-client-descriptor>
```

FCI provides four services:

1. FNV_ECG_FCI_GENSERVICE, listening for requests on "SERVICES/REQUEST/FNV/FCI/GENSERVICE" topic and broadcasting data to "SERVICES/DATA/FNV/FCI/BROADCAST" and "SERVICES/DATA/FNV/FCI/SMS_BROADCAST" topics.
2. FNV_ECG_FCI_FTCP_RESPONSE
3. FNV_ECG_FCI_FTCP_ALERT
4. FNV_ECG_FCI_FTCP_QUERY

FCI has a consumer requesting assess to Service Manager "SERVICE_MANAGER" service declared in Service Manager Descriptor. Similar consumer should be included in the most of the component descriptors. The topic is a topic used to create a SoaServiceDirectoryManager instance.

```
SoaClientEndpoint::SharedPtr sdmEndpoint = SoaMessageManager::createClientEndpoint( "SERVICES/RESPONSE/FNV/FCI
/SERVICEDIR" );
SoaServiceDirectoryManager::SharedPtr svcDirMan = SoaServiceDirectoryManager::create( messageManager,
sdmEndpoint );
```

4.3. FCI Test App Descriptor

Example of an application or a platform component descriptor providing FTCP command handler.

Sample SOA Clent Descriptor

```
<?xml version="1.0" encoding="utf-8"?>
<platform-client-descriptor uid="1021" version="1.0.0">
  <fci-clients>
    <fci-client id="FCITESTAPP">
      <ftcp-commands>
        <command id="DummyCommand"/>
      </ftcp-commands>
    </fci-client>
  </fci-clients>
  <soa-permissions>
    <soa-consumer-permissions>
      <endpoint>SERVICES/RESPONSE/FNV/FCITESTAPP/CONSUMER</endpoint>
      <soa-service-requests>
                <soa-service-group-request group-id="FNV_ECG_FCI"/>
      </soa-service-requests>
    </soa-consumer-permissions>
  </soa-permissions>
</platform-client-descriptor>
```

The descriptor contains:

1. A "FCITESTAPP" FTCP command handler.
2. An FCI service consumer: "SERVICES/RESPONSE/FNV/FCITESTAPP/CONSUMER"

NOTE 1. fci client id is an id currently used to register ftcp commands with FCI API.

5. Deploying a descriptor

5.1. ECG

1. Create a descriptor file and add it to "fnv/soa-descriptor" repo under "descriptors" directory.
    a. For consistency name the descriptor file *<component-name>-descr.xml*
2. Ad a soft link to the descriptor to "ecg" directory.
3. Whitelist the descriptor in system_build.xml using service manager uid(61) and gid(61). For other ECU's ensure that the file is included in the ECU image.

    ```
    <add_file src="etc/soa-sm/descriptors/soa-sm-descr.xml" tgt="etc/soa-sm/descriptors/soa-sm-descr.xml"
    perms="440" uid="61" gid="61"/>
    ```

4. Add Sergeev, Alexey (A.) and Liang, Su (S.) as a reviews for pull request.
5. **Required**: Ensure all service manager "DescriptorsTest" tests have passed before merging the PR.

Hardware test. Should be completed locally before starting a PR.

1. Get the latest ECG bundle from jenkins
2. Copy your descriptor to "etc/soa-sm/descriptors/" on target
3. Use "client db_recovery" EDT command. Service manager will remove existing database and create a new one.
4. Ensure Service Manger executed DB recovery without errors: in fdp log search for "ClientDataManager: registerEcgDescriptors: ECG database update success" message, application: "serviceManager".

5.2. Other ECUs

Services

1. Create a descriptor file and add it to "fnv/soa-descriptor" repo under "descriptors" directory.
    a. For consistency name the descriptor file *<component-name>-descr.xml*
2. Add a soft link to the descriptor to "ecu" directory, where ecu is an ecu name. Example "sync" for SYNC ecu.
3. Descriptor should be added to the ecu build. Please consult ecu build owner on how to proceed with the deployment.

Apps managed by ALM on SYNC

This feature is not working as 2019/11/15

1. Append the soa-descriptor to the end of the app descriptor (e.g. buildroot/swpart/sync4/sync4-alm-pkg/alm-pkg-Garmin-App/descriptor.xml)

Testing new descriptors.

Descriptors for platform ECU client are registered at run-time by ECU gateway. Every time when SYNC gateway connects to SOA, it tries to register SYNC descriptors with Service Manager (SM). That is a SOA API. The payload contains SYNC build version and descriptors. If the version does not match the one persisted in SM database, SM updates the database.

If a test build has the same version as previously registered, new descriptor will not be registered or updated. To test successfully developers need to clean up SM database.

There are two reliable ways to clean up SM database:

1. Reflash ECG. Please be noted that the database is stored in data partition: "/data/config/soa-sm/acl-db". Some "loadProduct" options do not remove data partition.
2. Use SM EDT command to trigger database recovery. SM will delete the database and create a new one.

Please capture ECG log while SYNC gateway registers descriptors. That will help to debug issues.

Follow the steps to update apply your descriptor update:

1. Flash ECU with the updated build
2. Disconnect or turn off ECU
3. Clean up SM database on ECG (pick one from the list above)
4. Start capturing ECG logs
5. Connect or turn on ECU
6. Wait for some time to allow gateway to register descriptors

Service Manager automatically creates database entries to allow FCI FCI Client and IPPT IPPT Client transactions, if <fci-clients> or <ippt-clients> field exists in the descriptor.

5.3. SOA internal

Control Data topic

Service Manager automatically allows any component to publish/subscribe to all subtopics of "SERVICES/DATA/CONTROL/#".

Control Data topic usage

Service status updates