

Configuration Interface Definition

Definition Overview

The Configuration service framework exposes device-specific methods such as restart and factory reset, device passcode, and device-specific settable attributes such as friendly name and default language. It is expected that OEM of the device would take this service framework and bundle it with a single application (system app). The enforcement of singleton instance of the Configuration service framework must be performed using explicit guidelines provided to OEMs and application developers regarding the usage of the Configuration service framework.

The following figure illustrates the relationship between software stack on the device hosting the AllJoyn™ service framework and the device hosting the AllJoyn client application.

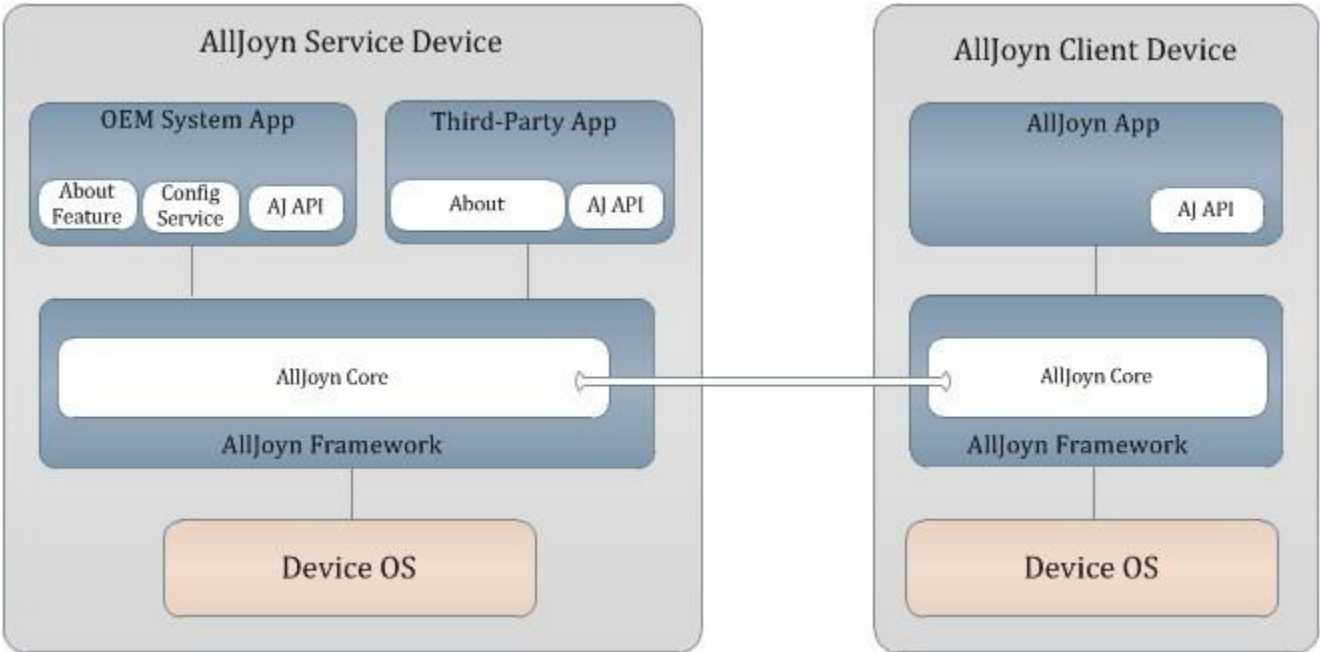


Figure: Configuration service framework architecture within the AllJoyn framework

The figure describes the scope of Configuration service framework and About feature in a multiple applications-per-device scenario. The following system behavior should be noted:

- The system application bundles the Configuration service framework and provides a remote mechanism to invoke device-specific configuration.
- It could be that OEMs provide equivalent (as exposed by the Configuration service framework) functionality via the local user interface.

NOTE: All methods and signals are considered mandatory to support the AllSeen Alliance Compliance and Certification program.

Typical Call Flows

This section highlights call flows that involve the Configuration service framework. The system app on the AllJoyn service framework device is involved in these call flows.

Device configuration change

The following figure illustrates a sample call flow where an Alljoyn app executing on an AllJoyn client device discovers the Configuration service framework via announcement and subsequently performs methods as specified in the Config interface to retrieve and update configuration data. See [Config Interface](#) for complete details.

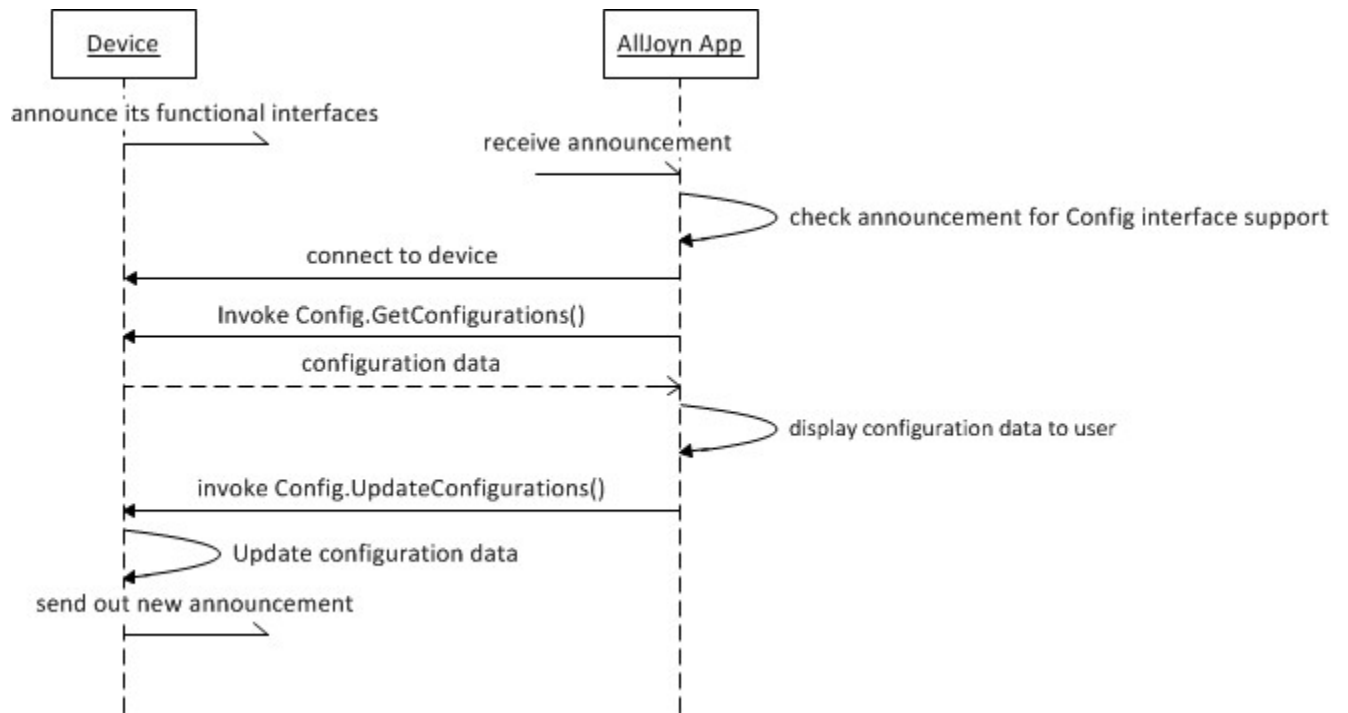


Figure: Device configuration change call flow

Factory reset

The following figure illustrates a sample call flow where an AllJoyn app executing on an AllJoyn client device discovers the Configuration service framework via announcement, and subsequently performs methods as specified in the Config interface to retrieve the configuration data and perform factory reset action if needed. See [Config Interface](#) for complete details.

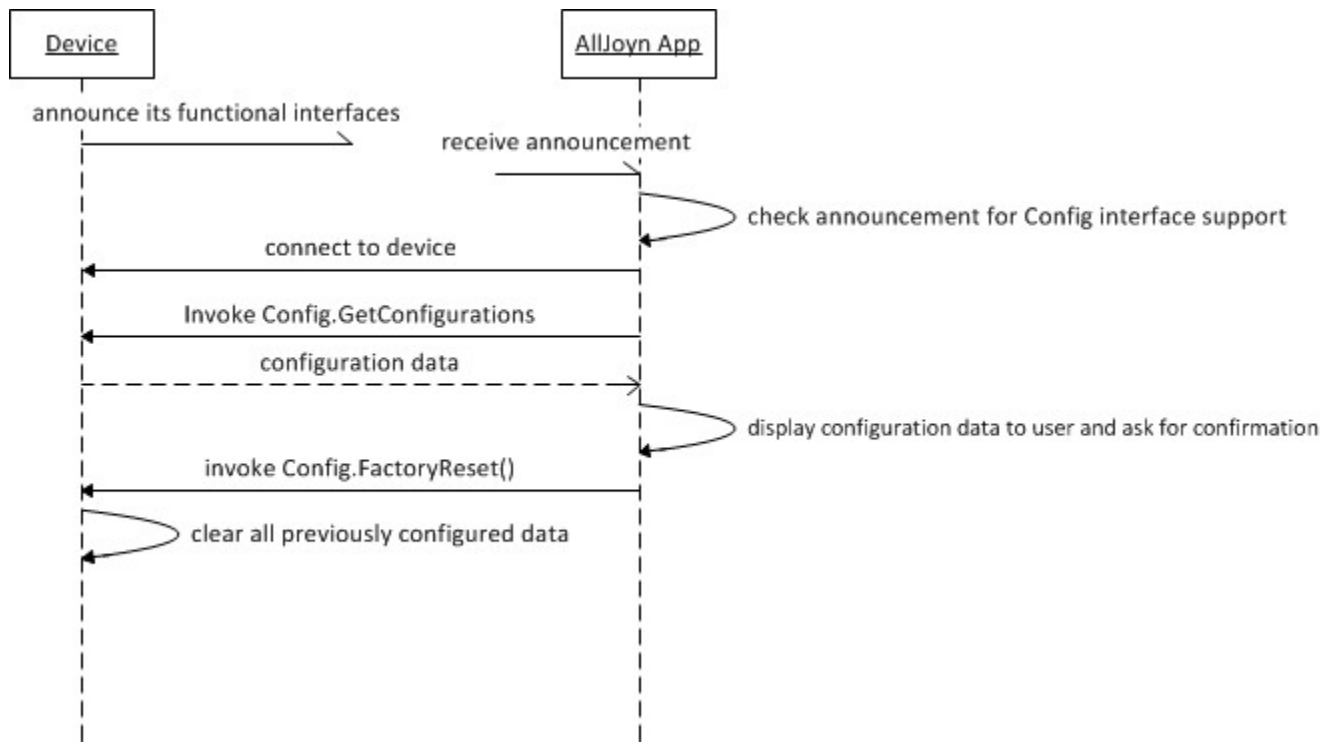


Figure: Device factory reset call flow

Error handling

The method calls in the Config interface use the AllJoyn error message handling feature (ER_BUS_REPLY_IS_ERROR_MESSAGE) to set the error name and error message.

Error name	
<code>org.alljoyn.Error.InvalidValue</code>	E
<code>org.alljoyn.Error.FeatureNotAvailable</code>	F
<code>org.alljoyn.Error.LanguageNotSupported</code>	T

Config Interface

Interface name

Interface name	Version
<code>org.alljoyn.Config</code>	1

Properties

Property name	Signature	List of values	Read/W
Version	<code>q</code>	Positive integers	Read-on

Methods

The following methods are exposed by the object that implements the `org.alljoyn.Config` interface.

`FactoryReset`

Message arguments

None.

Reply arguments

None.

Description

Directs the device to disconnect from the personal AP, clear all previously configured data, and start the softAP mode.

Error reply

Error	Description
<code>org.alljoyn.Error.FeatureNotAvailable</code>	Returned in the AllJoyn respo

`SetPasscode('say')`

Message arguments

Argument	Parameter name	Signature	List of values	Description
0	<code>daemonRealm</code>	<code>s</code>	N/A	Identifies the daemon's identity for secure framework.
1	<code>newPasscode</code>	<code>ay</code>	N/A	Passphrase that will be utilized for the secure framework.

Reply arguments

None.

Description

Updates the passcode to be used for the `org.alljoyn.Config` interface which is secure. The default passcode is 000000 until it is overwritten by `SetPasscode`.

```
a{sv} GetConfigurations('s')
```

Message arguments

Argument	Parameter name	Signature	List of values
0	<code>languageTag</code>	<code>s</code>	IETF language tags specified by RFC 5646.

Reply arguments

Argument	Parameter name	Return signature	List of values	Description
0	<code>configData</code>	<code>a{sv}</code>	N/A	Returns configuration fields in the form of Configuration map fields.

Description

Returns all the configurable fields specified within the scope of the Config interface. If language tag is not specified (i.e., ""), configuration fields based on the device's default language are returned.

##Error reply**

Error	Description
<code>org.alljoyn.Error.LanguageNotSupported</code> <code>UpdateConfigurations('sa{sv}')</code>	Returned if a lan

Message arguments

Argument	Parameter name	Signature	List of values
0	<code>languageTag</code>	<code>s</code>	IETF language tags specified by RFC 5646.
1	<code>configMap</code>	<code>a{sv}</code>	See Configuration map fields

Reply arguments

None.

Description

Provides a mechanism to update the configuration fields.

Error reply

Error	Description
<code>org.alljoyn.Error.InvalidValue</code>	Returned whenever there is an error in updating t contain the field name of the invalid field.
<code>org.alljoyn.Error.LanguageNotSupported</code> <code>ResetConfigurations('sas')</code>	Returned if a language tag is not supported by the

Message arguments

Argument	Parameter name	Signature	List of values
0	<code>languageTag</code>	<code>s</code>	IETF language tags specified by RFC 5646.
1	<code>fieldList</code>	<code>as</code>	N/A

Reply arguments

None.

Description

Provides a mechanism to reset (i.e., value is restored to factory default but the field itself is retained) values of configuration fields.

Error reply

Error	Description
<code>org.alljoyn.Error.InvalidValue</code>	Returned whenever there is an error related to field
<code>org.alljoyn.Error.LanguageNotSupported</code>	Returned if a language tag is not supported by the

Configuration map fields

The following table lists the known configuration fields that are part of the `configMap` parameter fields. The OEM or application developer can add additional fields.

Field name	Mandatory	Localized	Signature	Description
DefaultLanguage	yes	no	s	Default language supported by the device. If the parameter is not set as per the RFC, the error <code>org.alljoyn.Error.LanguageNotSupported</code> is returned. In this case, the default language on the device is used.
DeviceName	no	yes	s	Device name assigned by the user. The device

Introspection XML

```
<node name="/Config" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspect.xsd">
  <interface name="org.alljoyn.Config">
```



```
<property name="Version" type="q" access="read"/>
<method name="FactoryReset">
  <annotation name="org.freedesktop.DBus.Method.NoReply"
value="true"/>
</method>
<method name="Restart">
  <annotation name="org.freedesktop.DBus.Method.NoReply"
value="true"/>
</method>
<method name="SetPasscode">
  <arg name="daemonRealm" type="s" direction="in"/>
  <arg name="newPasscode" type="ay" direction="in"/>
</method>
<method name="GetConfigurations">
  <arg name="languageTag" type="s" direction="in"/>
  <arg name="configData" type="a{sv}" direction="out"/>
</method>
<method name="UpdateConfigurations">
  <arg name="languageTag" type="s" direction="in"/>
  <arg name="configMap" type="a{sv}" direction="in"/>
</method>
<method name="ResetConfigurations">
  <arg name="languageTag" type="s" direction="in"/>
  <arg name="fieldList" type="as" direction="in"/>
</method>
</interface>
</node>
```