ALLSEEN ALLIANCE

AllJoyn™ Core Interface Definition

Version 14.06 Update 1

October 28, 2014

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document defines the AllJoyn™ Core interfaces:

- About interface
- Events and Actions interface

The About interface is required by an application to provide the discovery mechanism for the service framework interfaces that it supports, as well as providing the basic identification information.

Events allow devices, AllJoyn nodes, to notify other AllJoyn nodes when something of significance occurs in the network. AllJoyn nodes may also expose Actions that can be invoked by other devices on the network. The power of events can be fully realized when there is a corresponding Action framework and Event picker application in the network that allows end users to program actions that should be taken when an event is sent.

1.2 Scope

This document is targeted to the developers for AllJoyn applications.

1.3 Release history

Release version	Date	What changed
14.02	2/28/2014	About interface version 1 was added.
14.06	6/30/2014	No updates
14.06 Update 1	10/28/2014	■ Updated the document title and Overview chapter title (changed Specification to Definition). ■ Combined the Events and Actions interface content with the About interface content to become the Core interface content. ■ Added the release version number to the document title for version tracking. ■ Added a note in the Definition Overview chapter to address the AllSeen Alliance Compliance and Certification program. ■ Added a Mandatory column for method and signal parameters to support the AllSeen Alliance Compliance and Certification program.

1.4 References

Except for RFCs, the following are reference documents found on the AllSeen Alliance web site's Docs/Downloads section.

- *AllJoyn™ Framework Tutorial*
- *Introduction to AllJoyn™ Thin Library*
- *RFC 5646 (Tags for Identifying Languages)*

1.5 Acronyms and terms

Acronuym/term	Definition
Action	A function performed by an AllJoyn-enabled device.
Action-receiving device	The device that performs an action.
Authoring app	Application that carries out the IFTTT rules.
Event	A message denoting that something has happened.
Event-emitting device	The device that sends the event.
Event Picker app	Application that lets end users program actions to take when an event is sent.
IFTTT	If This Then That. A logical construct that tests for a certain condition and then performs an action if it is "true".

2 Definition Overview

The About interface is to be implemented by an application on a target device. This interface allows the app to advertise itself so other apps can discover it. *Figure 1* illustrates the relationship between a client app and a service app.

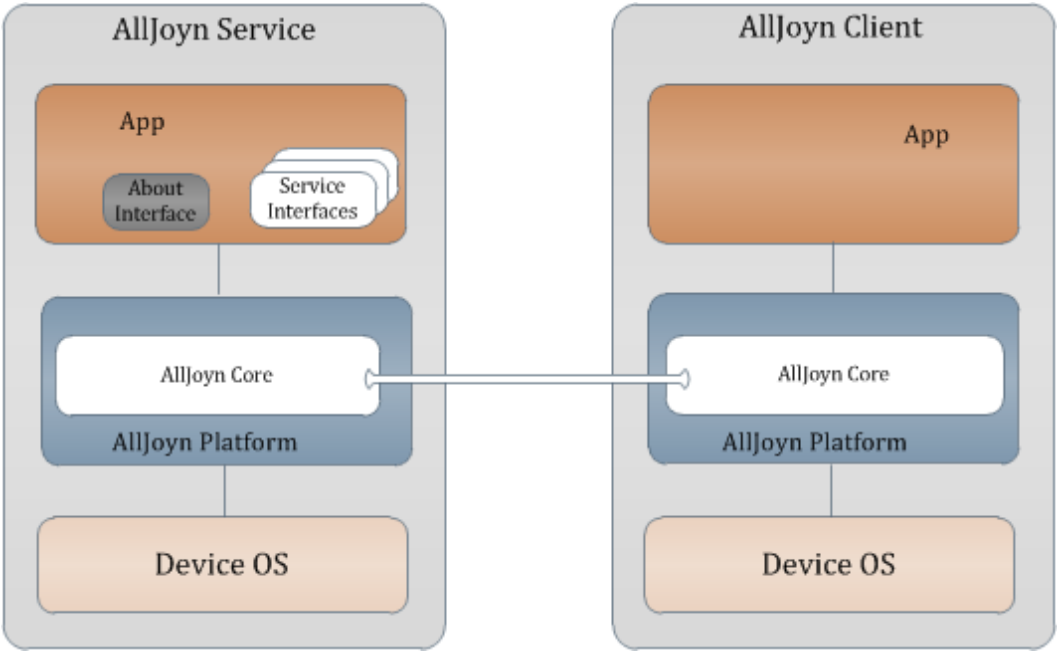

Figure 1: About feature architecture within the AllJoyn framework

The Events and Actions feature is part of the AllJoyn core, designed to enable creating If-This-Then-That (IFTTT)-based rules logic in the AllJoyn network.

- Events are used by AllJoyn devices/apps to notify other AllJoyn devices/apps when something of significance occurs in the network.
- Actions enable specific responses to AllJoyn events detected in the AllJoyn network. In this regard, Events and actions go hand-in-hand. An action is a way for making an application or device do something.

For example, an AllJoyn application can broadcast an event signifying that something has happened, such as movement that was detected by a motion detector. An AllJoyn application can receive this event and respond to it by taking a specific action, such as turning on the security camera.

Events are realized using AllJoyn sessionless signals, while actions are realized using AllJoyn methods. A description element is added to the AllJoyn introspect XML format to provide human readable text for the various events a device may emit and actions it may receive.

Figure 2 illustrates the context architecture for the Events and Actions feature. Events and actions are advertised in the Announcement signal using the org.allseen.Introspectable interface. Any advertised object supporting an event-emitting interface or action-receiving interface will include the org.allseen.Introspectable interface in the Announcement signal. The Event Picker App receives announcement signals from AllJoyn devices emitting events that can receive actions. The app introspects those devices to retrieve a human-readable description for all events and actions as part of the enhanced introspection XML data.

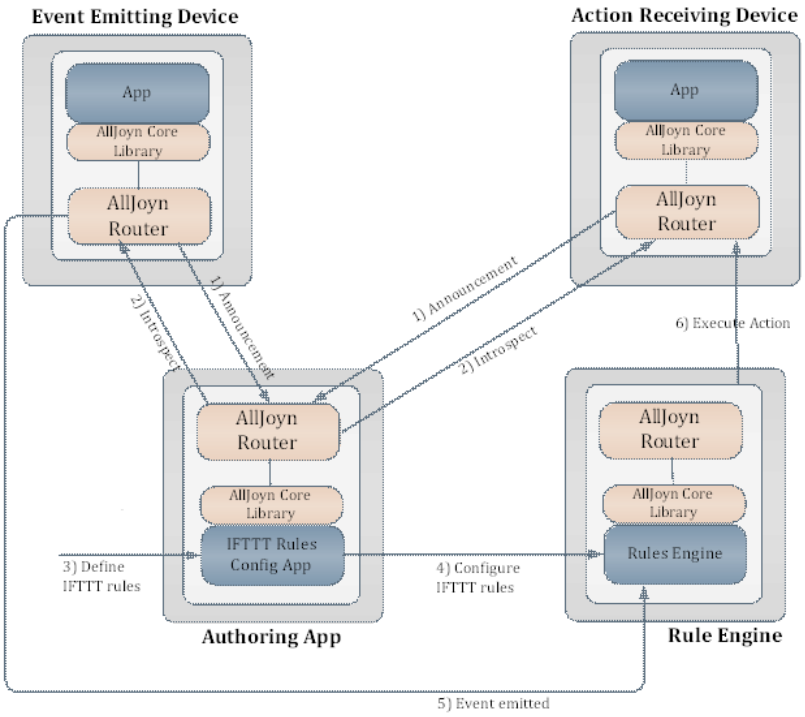

Figure 2: Events and Actions context architecture

The human-readable text description in the Event Picker app details can be presented to a user allowing the user to create IFTTT-based rules for automation in the AllJoyn network. These IFTTT rules get configured on a Rules Engine which can be on the same device or a different device than the Authoring app.

Note The Rules engine is beyond the scope of current design and its implementation is left to the ecosystem. The Rule Engine application detects when the event is emitted. Based on the configured IFTTT rules, it executes actions (method call) on the action-receiving devices.

Note All methods and signals are considered mandatory to support the AllSeen Alliance Compliance and Certification program. Individual parameters for a given method or signal may be considered mandatory or optional, and are specified accordingly in this document.

3 Discovery

A client can discover the app via an announcement which is a sessionless signal containing the basic app information like app name, device name, manufacturer, and model number. The announcement also contains the list of object paths and service framework interfaces to allow the client to determine whether the app provides functionality of interest.

In addition to the sessionless announcement, the About interface also provides the on-demand method calls to retrieve all the available metadata about the app that are not published in the announcement.

4 Discovery call flows

4.1 Typical discovery flow

Figure 3 illustrates a typical call flow for a client to discover a service app. The client merely relies on the sessionless announcement to decide whether to connect to the service app to use its service framework offering.

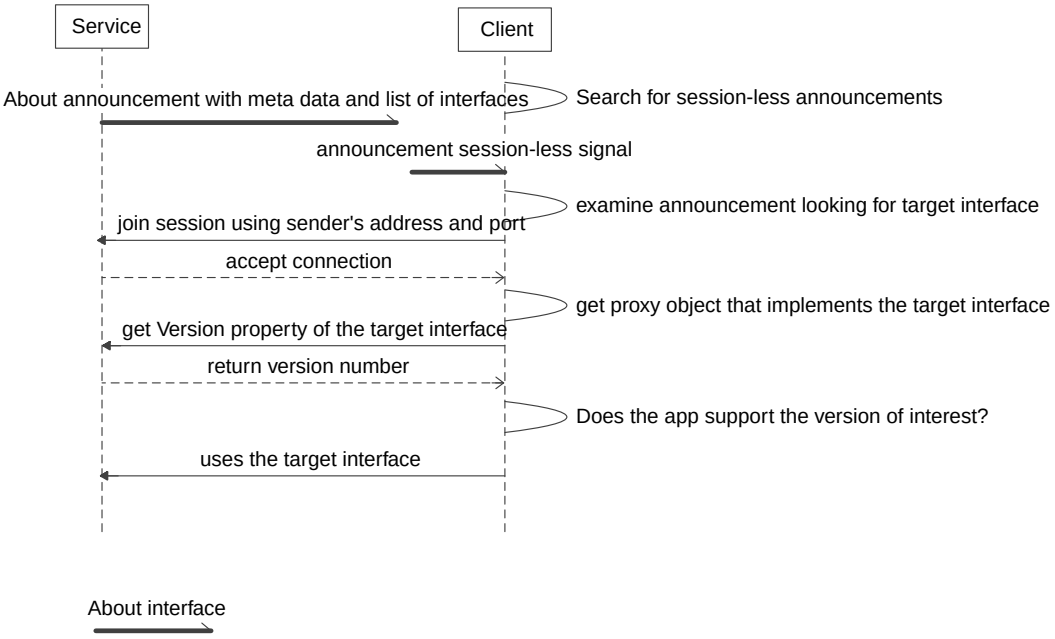

Figure 3: Typical discovery flow (client discovers a service app)

4.2 Nontypical discovery flow

Figure 4 illustrates a call flow for a client to discover a service app and make a request for more detailed information.

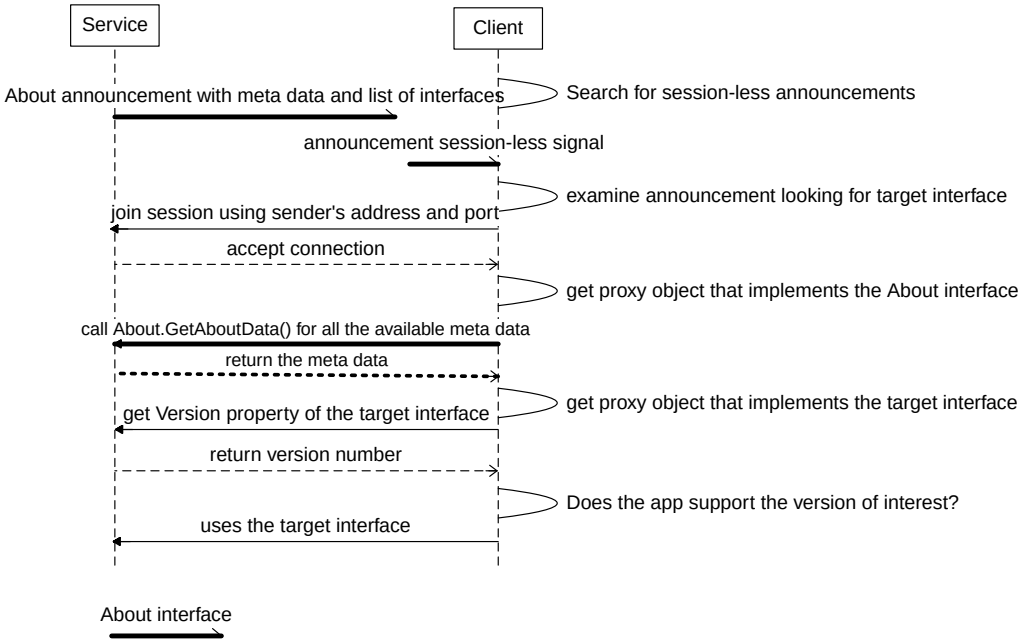

Figure 4: Nontypical discovery call flow

5 Error handling

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

Table 1 lists the possible errors raised by the About interface.

Table 1: About interface error handling

Error name	Error message
org.alljoyn.Error.LanguageNotSupported	Language not supported

6 About interface

6.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.About	1	no	/About

6.2 Properties

Property name	Signature	List of values	Writable	Description
Version	q	Positive integers	no	Interface version number

6.3 Methods

The following methods are exposed by a BusObject that implements the org.alljoyn.About interface.

6.3.1 GetAboutData

Inputs

Parameter name	Mandatory	Signature	List of values	Description
languageTag	yes	s	IETF language tags specified by RFC 5646	The desired language.

Output

Return signature	Mandatory	Description
a{sv}	yes	A dictionary of the available metadata fields. If language tag is not specified (i.e. “”), metadata fields based on default language are returned. If a language tag is not supported, the error org.alljoyn.Error.LanguageNotSupported is returned.

Description

Retrieve the list of available metadata fields based on the language tag.

Field information

Table 2 lists the names of the metadata fields. The fields with a yes value in the Announced column will also be published via the Announce signal. See *Signals* for more information.

Table 2: About interface data fields

Field name	Mandatory	Announced	Localized	Signature	Description
AppId	yes	yes	no	ay	The globally unique identifier for the application.
DefaultLanguage	yes	yes	no	s	The default language supported by the device. Specified as a IETF language tag listed in RFC 5646.
DeviceName	yes	yes	yes	s	Name of the device set by platform-specific means (such as Linux and Android).
DeviceId	yes	yes	no	s	Device identifier set by platform-specific means.
AppName	yes	yes	yes	s	Application name assigned by the app manufacturer (developer or the OEM).
Manufacturer	yes	yes	yes	s	The manufacturer's name of the app.
ModelNumber	yes	yes	no	s	The app model number.
SupportedLanguages	yes	no	no	as	List of supported languages.
Description	yes	no	yes	s	Detailed description expressed in language tags as in RFC 5646.
DateOfManufacture	no	no	no	s	Date of manufacture using format YYYY-MM-DD (known as XML DateTime format).
SoftwareVersion	yes	no	no	s	Software version of the app.
AJSoftwareVersion	yes	no	no	s	Current version of the AllJoyn SDK used by the application.
HardwareVersion	no	no	no	s	Hardware version of the device on which the app is running.
SupportUrl	no	no	no	s	Support URL (populated by the manufacturer).

6.3.2 GetObjectDescription

Inputs

None.

Output

Return signature	Mandatory	Description
a(oas)	yes	Return the array of object paths and the list of supported interfaces provided by each object.

Description

Retrieve the object paths and the list of all interfaces implemented by each of objects.

6.4 Signals

Signal name	Parameters			Sessionless	Description
Announce	**Name**	**Mandatory**	**Signature**	yes	This signal is used to announce the application information and the service framework interfaces that it supports. The following information is provided in the signal: ■ version -- Version number of the About interface. ■ port -- Session port. The app will listen on this port for incoming sessions. ■ objectDescription --Array of object paths and the list of supported interfaces provided by each object. ■ metaData -- Metadata. All the fields listed in *Table 2* with a yes value in the Announced column are provided in this signal.
	version	yes	q		
	port	yes	q		
	objectDescription	yes	a(oas)		
	metaData	yes	a{sv}		

6.5 AllJoyn Introspection XML

The following XML defines the org.alljoyn.About interface.

```
<node name="/About"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/
      introspec t.xsd">
     <interface name="org.alljoyn.About">
          <property name="Version" type="q" access="read"/>
          <method name="GetAboutData">
               <arg name="languageTag" type="s" direction="in"/>
              <arg name="aboutData" type="a{sv}" direction="out"/>
```

```
        </method>
        <method name="GetObjectDescription">
             <arg name="objectDescription" type="a(sas)"
              direction="out"/>
        </method>
        <signal name="Announce">
             <arg name="version" type="q"/>
             <arg name="port" type="q"/>
             <arg name="objectDescription" type="a(sas)"/>
             <arg name="metaData" type="a{sv}"/>
        </signal>
        </interface>
    </node>
```

7 Introspectable Interface

This chapter defines the Introspectable interface used by the Events and Actions feature.

7.1 Interface name

Interface name	Version	Secured	Object path
org.allseen.Introspectable	1	yes	Can be any object path

7.2 Properties

Property name	Signature	List of values	Writable	Description
Version	q	Positive integers	no	Interface version number

7.3 Methods

The following methods are exposed by a BusObject that implements the org.allseen.Introspectable interface.

7.3.1 GetDescriptionLanguages

Inputs

None.

Outputs

Return signature	Parameter name	Mandatory	Description
as	languageTags	yes	List of the languages in which this object has descriptions.

Description

Returns the aggregate of the languages for which this object has descriptions. For example, if an object implements two interfaces, X and Y - X has all of its members described in English (en) and French (fr) and Y has some descriptions in English (en) and Chinese (cn), this method will return ["en", "fr", "cn"]. The language tags will comply with IETF language tag standards.

7.3.2 IntrospectWithDescription

Inputs

Parameter name	Mandatory	Signature	List of values	Description
languageTag	yes	q		Requested language.

Output

Return signature	Parameter name	Mandatory	Description
s	data	yes	Returned introspection XML

Description

Returns the XML defined above with descriptions in the specified language (exact match only - no best match). If an element, e.g., method, does not have a description in that language, no description attribute is placed within the element.

7.4 AllJoyn introspection XML

The following XML defines the org.allseen.Introspectable interface.

```
<node name="/com/example/LightBulb">
   <description>Your lightbulb</description>
   <interface name="com.example.LightBulb">
      <description>Provides basic lighting functionality</description>
      <method name="ToggleSwitch">
         <description>Invoke this to toggle whether the light
         is on or off</description>
         <arg name="brightess" type="i" direction="in">
         <description>A value to specify how bright the bulb should
         shine</description>
         </arg>
      </method>
      <signal name="LightOn" sessionless=”true”>
         <description>Emitted when the light turns on</description>
      </signal>
      <signal name="LightOff" sessionless=”true”>
         <description>Emitted when the light turns off</description>
      </signal>
      <property name="LightState" type="y" access="read">
         <description>The current state of this light bulb</description>
      </property>
   </interface>
   <node name="child">
         <description>Some relevant description</description>
   </node>
</node>
```