

# ***AllJoyn™ Analytics Service Framework 1.0 Interface Definition***

***November 9, 2014***

---

This work is licensed under a Creative Commons Attribution 4.0 International License.

<http://creativecommons.org/licenses/by/4.0/>

Any and all source code included in this work is licensed under the ISC License per the AllSeen Alliance IP Policy.

<https://allseenalliance.org/allseen/ip-policy>

---

# Contents

---

|  |          |
|--|----------|
| <b>1 Introduction.....</b>                               | <b>4</b> |
| 1.1 Purpose .....  | 4        |
| 1.2 Scope .....  | 4        |
| 1.3 Release history .....                                | 4        |
| 1.4 References .....                                     | 4        |
| <b>2 Specification .....</b>                             | <b>5</b> |
| 2.1 Overview .....                                       | 5        |
| 2.2 Discovery .....                                      | 7        |
| 2.3 Discovery call flows .....                           | 7        |
| 2.3.1 Typical discovery flow .....                       | 7        |
| 2.3.2 Non-typical discovery flow.....                    | 7        |
| 2.4 AnalyticsEventAgent interface .....                  | 8        |
| 2.4.1 AnalyticsEventAgent call flow .....                | 8        |
| 2.4.2 Interface name .....                               | 9        |
| 2.4.3 Properties .....                                   | 9        |
| 2.4.4 Methods .....                                      | 10       |
| 2.4.5 AllJoyn AnalyticsEventAgent Introspection XML..... | 11       |

# Figures

Figure 1. Analytics service architecture within the AllJoyn framework..... 5

Figure 2. Analytics app and agent running on same physical device. .... 6

Figure 3. Application with analytics core linked in..... 6

Figure 4. Typical discovery call flow (client discovers a service app) ..... 7

Figure 5. Non-typical discovery call flow ..... 8

Figure 6. AnalyticsEventAgent call flow. .... 9

# 1 Introduction

---

## 1.1 Purpose

This document describes the specification of the AllJoyn™ Analytics Service Interface. This interface is intended to provide a standard, vendor-agnostic method of sending analytics information from devices on the AllJoyn proximal network to analytics services in the cloud.

## 1.2 Scope

This document is targeted to the developers for AllJoyn applications.

## 1.3 Release history

| Release version | What changed   |
|-----------------|--|
| 14.12           | ■ AnalyticsEventAgent interface version 1 was added. |

## 1.4 References

The following are reference documents found on the AllSeen Alliance web site's Docs and Downloads section.

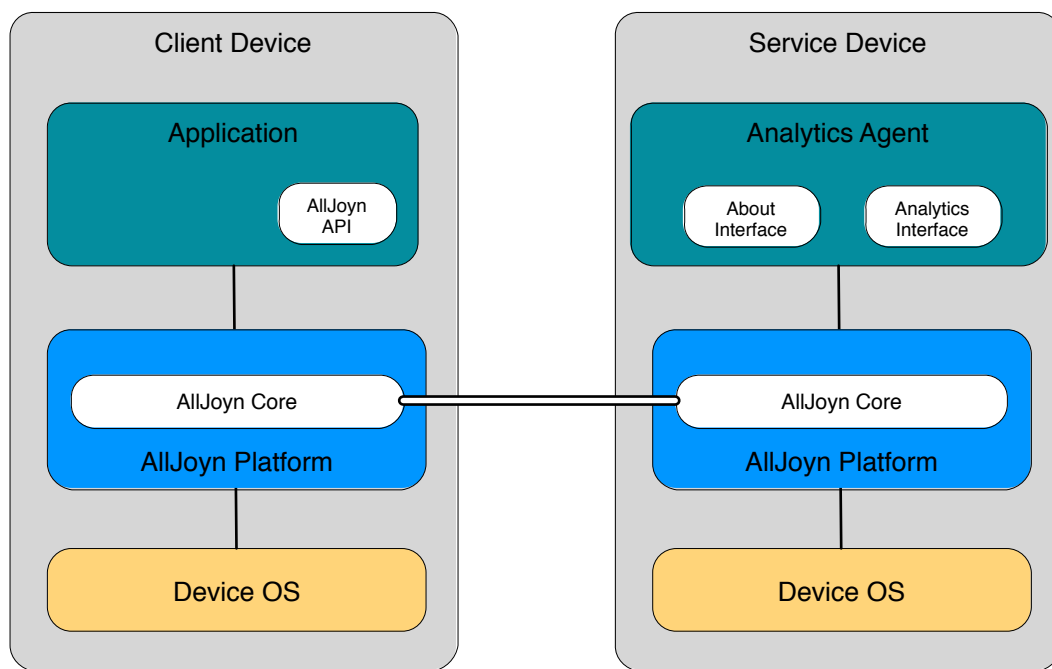
- [Introduction to the AllJoyn™ Framework](#)
- [Introduction to AllJoyn™ Thin Library](#)

## 2 Specification

---

### 2.1 Overview

An agent implement the analytics interface, possibly as an AllJoyn™ Gateway Agent. It is expected that each device manufacturer would deploy an agent specific to its devices. The agent may run on the same physical device as the client application, or on a separate device, depending on the manufacturer's requirements. Figure 1 shows the relationship between an application and the analytics agent when running on two separate devices. Figure 2 illustrates the same configuration, but with client and service running on the same physical device. Figure 3 shows the analytics core linked into the application, with AllJoyn messages replaced with direct API calls.



**Figure 1. Analytics service architecture within the AllJoyn framework.**

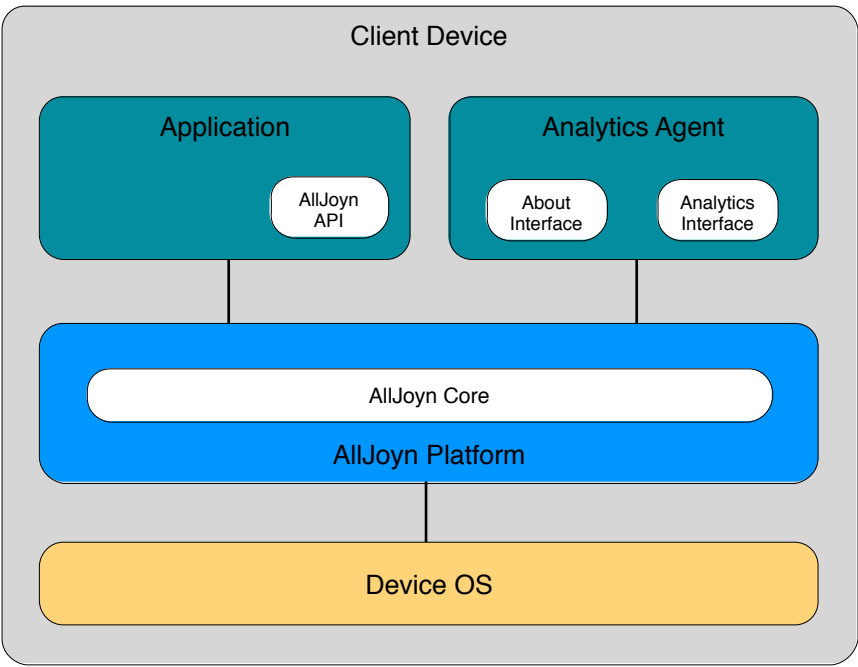


Figure 2. Analytics app and agent running on same physical device.

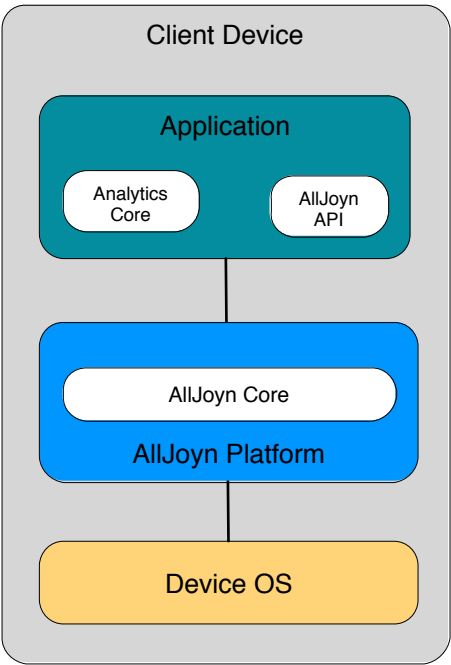


Figure 3. Application with analytics core linked in.

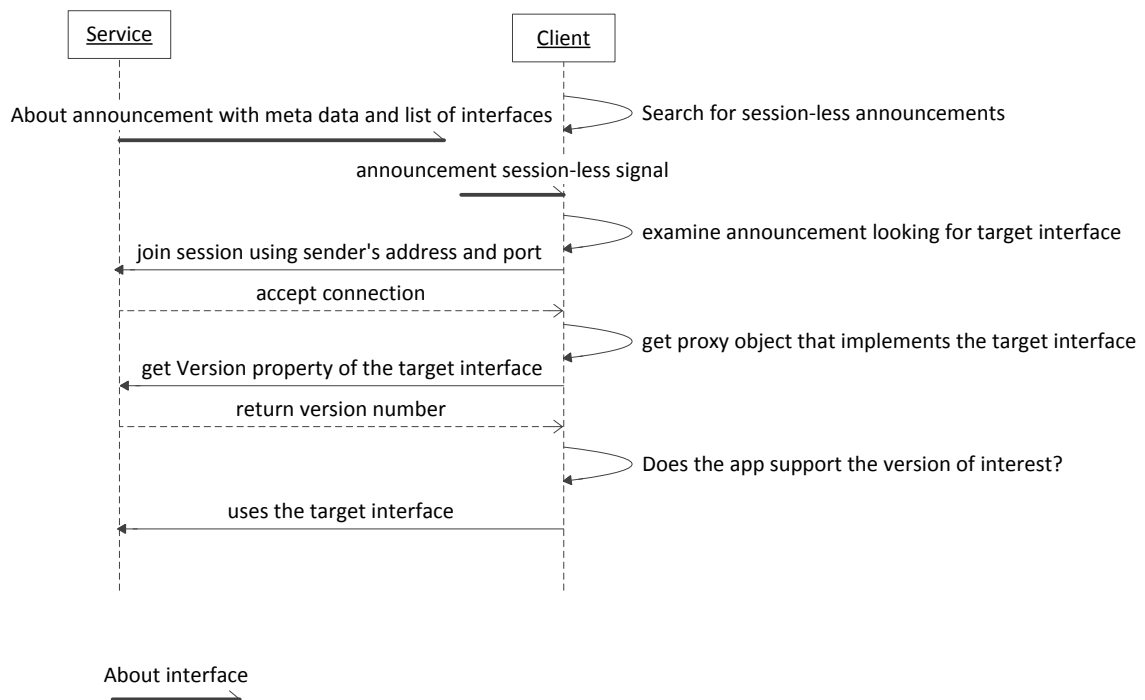
## 2.2 Discovery

A client can discover the agent 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 interfaces to allow the client to determine whether the app provides functionality of interest.

## 2.3 Discovery call flows

### 2.3.1 Typical discovery flow

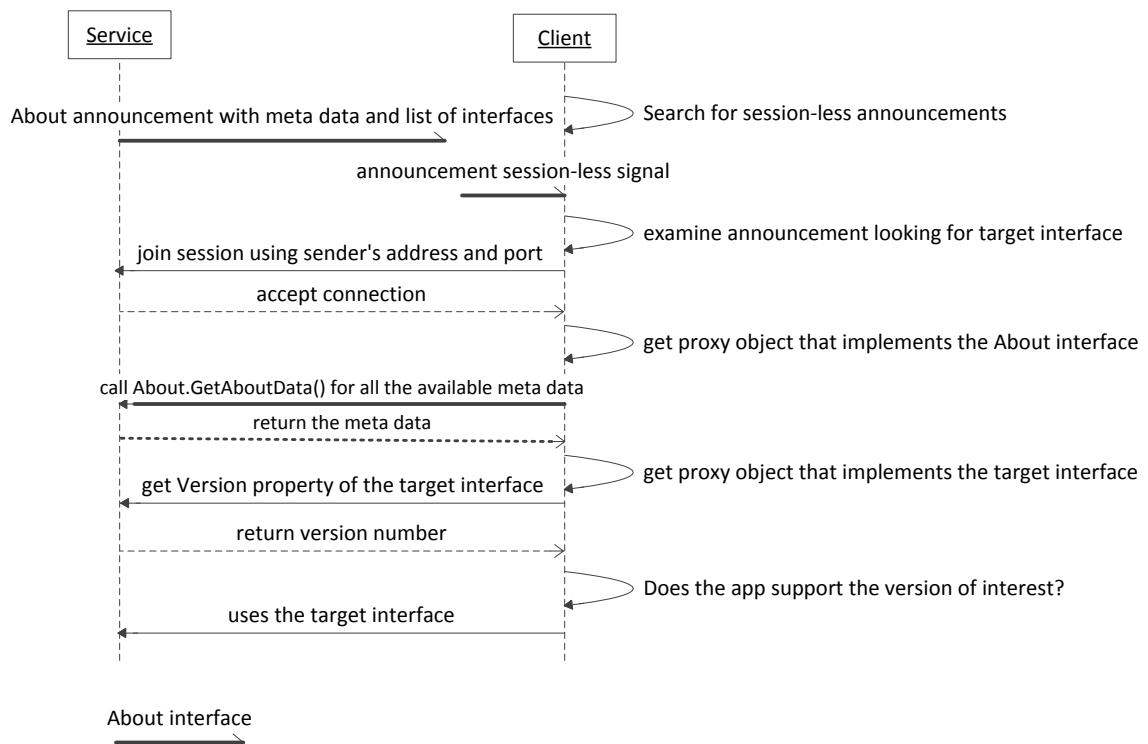
Figure 4 illustrates a typical call flow for a client to discover a service app. The client merely relies on the sessionless announcement to decide whether to connect to the service app to use its service offering.



**Figure 4. Typical discovery call flow (client discovers a service app)**

### 2.3.2 Non-typical discovery flow

Figure 5 illustrates a call flow for a client to discover a service app and make a request for more detailed information.



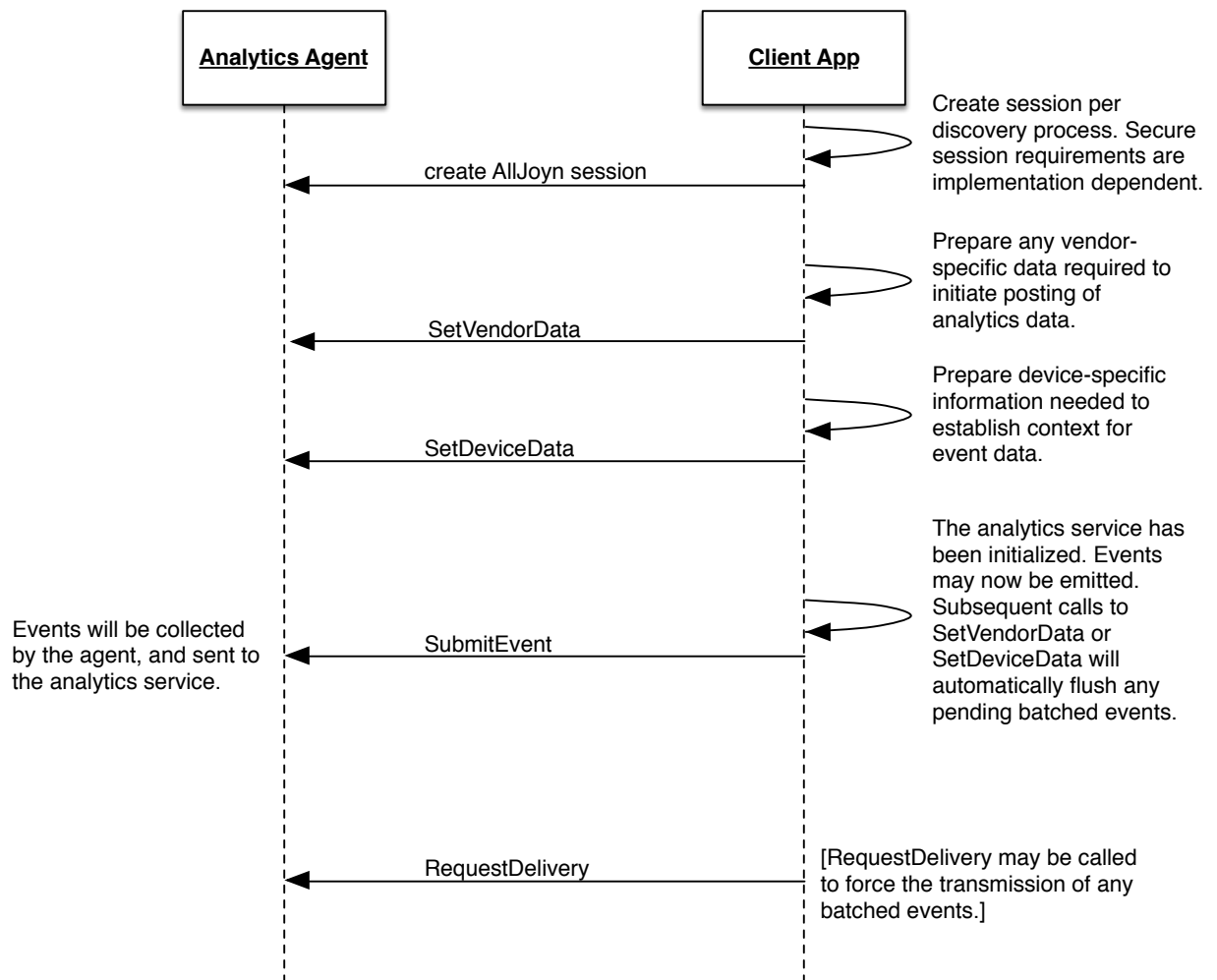
**Figure 5. Non-typical discovery call flow**

## 2.4 AnalyticsEventAgent interface

### 2.4.1 AnalyticsEventAgent call flow

Figure 6 illustrates a typical call flow for the AnalyticsEventAgent interface.





**Figure 6. AnalyticsEventAgent call flow.**

## 2.4.2 Interface name

| Interface name                  | Version | Secured |
|---------------------------------|---------|---------|
| org.alljoyn.AnalyticsEventAgent | 1       | Yes.    |

## 2.4.3 Properties

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

## 2.4.4 Methods

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

### 2.4.4.1 SetDeviceData

#### Inputs

| Parameter name | Signature | List of values           | Description  |
|----------------|-----------|--------------------------|--|
| values         | a{sv}     | List of key value pairs. | Values to be logically included with all events. Vendor-specific requirements may cause errors to be thrown.<br><br>Example: Model version number, device serial number. |

#### Description

SetDeviceData is used to provide contextual information for events being generated by the device. These typically would include the fields described in [Table 2](#) of the *AllJoyn About Feature 1.0 Interface Specification*.

### 2.4.4.2 SetVendorData

#### Inputs

| Parameter name | Signature | List of values           | Description  |
|----------------|-----------|--------------------------|--|
| values         | a{sv}     | List of key value pairs. | Vendor-specific settings related to the functionality of the vendor's service. This could include API keys, upload protocol settings, etc. |

#### Description

Allow the device to set vendor-specific values related to the functionality of the vendor's service.

### 2.4.4.3 RequestDelivery

#### Description

Request batched data to be flushed to the vendor's cloud service.

## 2.4.4.4 SubmitEvent

### Inputs

| Parameter name | Signature | List of values           | Description  |
|----------------|-----------|--------------------------|--|
| name           | s         | Event name               | The name of the application event.   |
| timestamp      | t         | Unix epoch timestamp     | Time at which the event occurred. This may be set to zero if the device does not have a real-time clock. |
| sequence       | u         | Unsigned integer         | A monotonically increasing value providing a relative sequencing of events.                              |
| values         | a{sv}     | List of key value pairs. | Event attributes.  |

### Description

Log an event to the analytics service.

## 2.4.5 AllJoyn AnalyticsEventAgent Introspection XML

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

```
<node name="/Analytics"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspect.xsd">
  <interface name="org.alljoyn.AnalyticsEventAgent">
    <!-- set values to be logically included with all events.
      vendor-specific validator can cause errors to be thrown.
      Example: Model version number, device serial number.
    -->
    <method name="SetDeviceData">
      <arg name="values" type="a{sv}" direction="in"/>
    </method>

    <!-- set analytics-vendor-specific data for this device.
      API keys, upload protocol information, etc.
    -->
    <method name="SetVendorData">
      <arg name="values" type="a{sv}" direction="in"/>
    </method>
```

```
<!-- request flush of any batched data to cloud. -->
<method name="RequestDelivery"></method>

<method name="SubmitEvent">
  <!-- event type -->
    <arg name="name" type="s" direction="in"/>

  <!-- epoch timestamp or 0 -->
    <arg name="timestamp" type="t" direction="in"/>

  <!-- sequence number, or 0 -->
    <arg name="sequence" type="u" direction="in"/>

  <!-- array of field names and values-->
    <arg name="values" type="a{sv}" direction="in"/>
</method>
<annotation name="org.alljoyn.Bus.Secure" value="on"/>
</interface>
</node>
```