ALLSEEN ALLIANCE

Technical Steering Meeting

September 16, 2014

Antitrust Compliance Notice

- AllSeen Alliance meetings involve participation by industry competitors, and it is the intention of AllSeen Alliance to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of and not participate in any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at AllSeen Alliance meetings and in connection with AllSeen Alliance activities are described in the AllSeen Alliance Antitrust Policy. If you have questions about these matters, please contact your company counsel, or if you are a member of AllSeen Alliance, feel free to contact Lee Gesmer or Andrew Updegrove, of the firm of Gesmer Updegrove LLP, which provides legal counsel to AllSeen Alliance.

Reminder:
This call is being recorded

Agenda

1. Approve minutes from previous meeting

2. Security Manager project proposal

3. Version numbering clarification

4. Analytics project proposal
   - Discussion and vote

5. Progress Update of Smart Home Service Framework

Security Manager Project Proposal

Technicolor

Security Manager Project Proposal (1 of 2)

- Project Description:
  - As part of the Security 2.0 design, The Security Manager is a service that helps the user with key management and permission rules building. Using policy templates defined by application developer, the Security Manager builds the application manifest to let the end-user authorize which interactions the application can do. The security Manger is optional because the permissions can be installed directly into the application. Hence the proposal to consider it as a service and have the source code stored in a dedicated git repository.
  - See https://wiki.allseenalliance.org/_media/core/alljoyn_security_2.0_hld_rev1.2.docx

- Scope:
  - Security 2.0: claiming, manifest, permission management, policies

- Dependencies:
  - AllSeen Core for permission management and manifest enforcing

Security Manager Project Proposal (2 of 2)

- Project Name:
  - Security manager
  - Repo: core/securitymgr

- Proposed Working Group:
  - Part of AllSeen Core WG

- Committers and Contributors:
  - Committers: Gerrit Ruelens, Phil T. Nguyen

- Initial Contribution:
  - Initial contribution expected end w1438 when the repository is available

Version numbering clarification

Version Numbering

The recent release process change allows service projects to have multiple feature releases per core release. AllJoyn needs a versioning scheme that captures:

- Core dependency: <core-year>.<core-month>, e.g. 14.06

- Feature release level: For use by services with more than one feature release built on the same core version

- Bug fix versions not affecting incompatibility and certification

We will now append a letter for bug fix releases:

<core-year>.<core-month>.<feature><bugfix-letter>

Examples:

- 14.06.00 is an initial service release using core 14.06

- 14.06.00a is a bug fix release

- 14.06.00b is another bug fix release

- 14.06.01 is the next service feature release using core 14.06

Version Numbering

As before, bug fix releases cannot add features, modify APIs, or introduce incompatible behavior. They are only intended to address critical stability and security issues.

The AllJoyn core will only add features on major releases – 14.12, 15.04, etc.

Analytics Project Proposal

Tellient

Discussion points

- Vote on Analytics as a project

- Where should Analytics reside?
  - Smarthome WG?
  - New WG?

the problem.

There is a gap in the delivery of analytics for the Internet of Things.
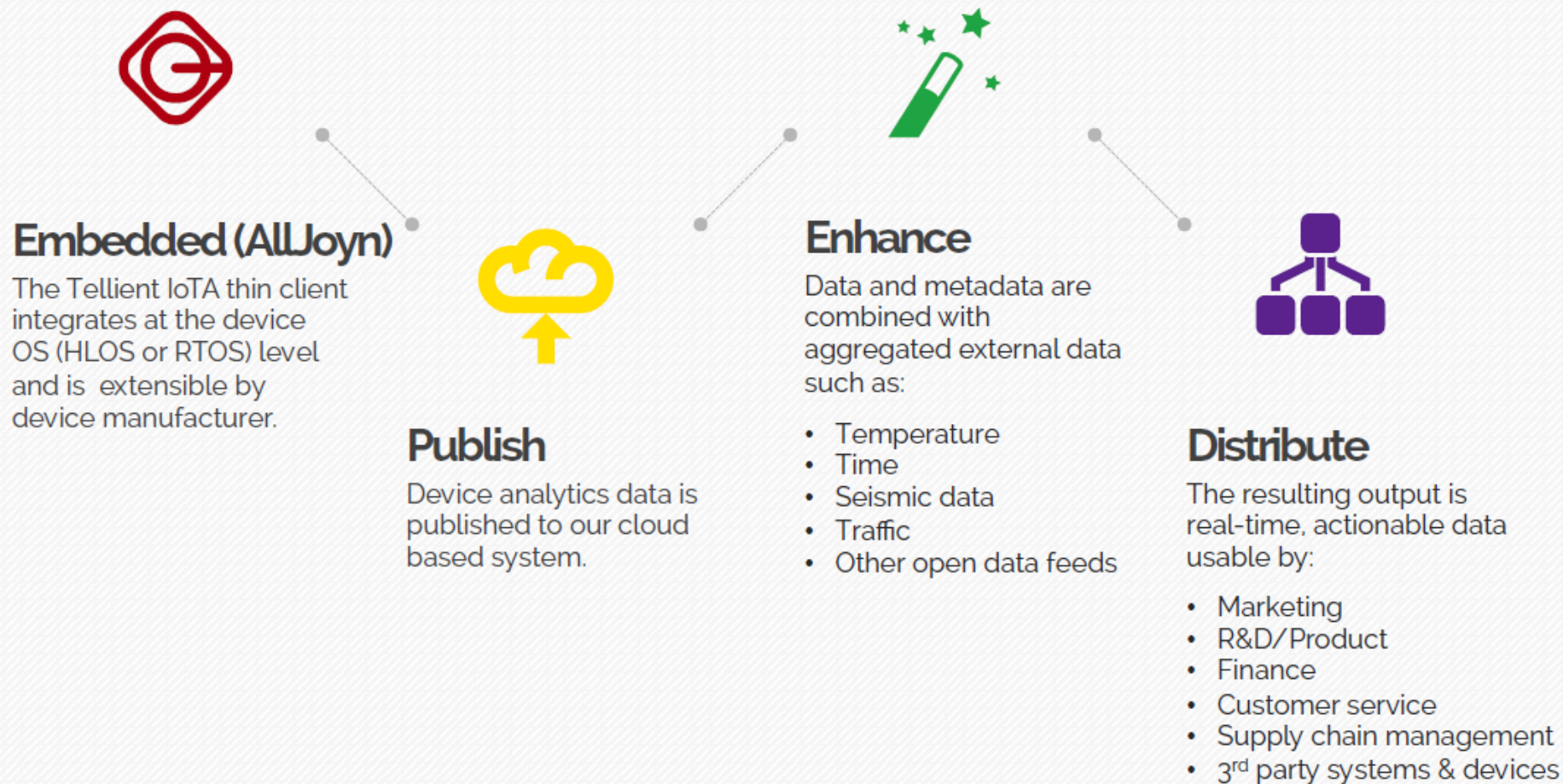
Device   Gateway   Analytics Engine   Visualization

The First Mile Problem

We provide an embedded client and a robust analytics solution specifically for connected devices.

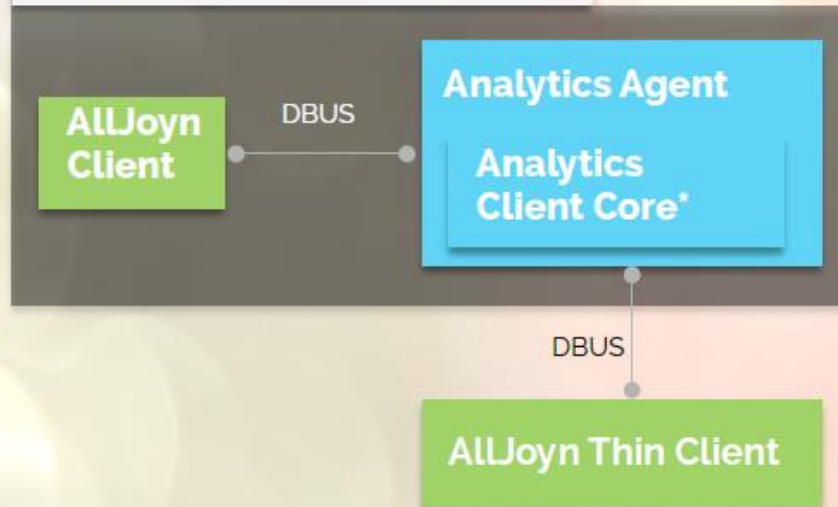Tellient

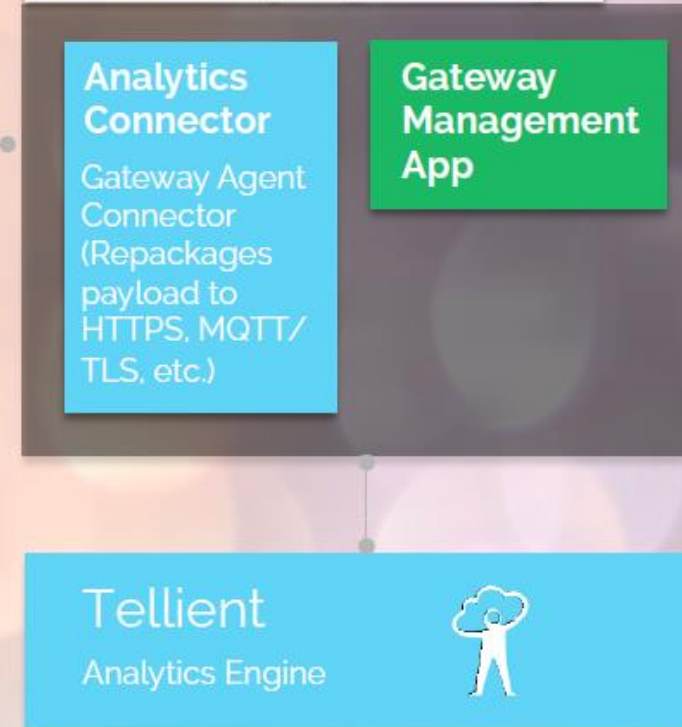# how it works.

Distributed and connected analytics.

Tellient

Embedded (AllJoyn)

The Tellient IoTA thin client integrates at the device OS (HLOS or RTOS) level and is extensible by device manufacturer.

Publish

Device analytics data is published to our cloud based system.

Enhance

Data and metadata are combined with aggregated external data such as:

- Temperature
- Time
- Seismic data
- Traffic
- Other open data feeds

Distribute

The resulting output is real-time, actionable data usable by:

- Marketing
- R&D/Product
- Finance
- Customer service
- Supply chain management
- 3rd party systems & devices

the solution.

AllJoyn Analytics system overview.

Tellient

the thing

AllJoyn Client — DBUS — **Analytics Agent**

Analytics Client Core*

DBUS

AllJoyn Thin Client

DBUS w/ protobuf payload

the AllJoyn gateway

Analytics Connector

Gateway Agent Connector (Repackages payload to HTTPS, MQTT/ TLS, etc.)

Gateway Management App

Tellient

Analytics Engine

*Analytics Client Core

- Custom protobuf implementation for collecting event data.
- Small footprint that could be embedded directly in AllJoyn clients, thick or thin, depending on requirements.

Language bindings proposal

Fon

SWIG

- Unique wrapper code to create the bindings
- Several languages support
- Actively maintained
- Same interface across languages

Manual

- Bindings maintained manually
- Development must support each target language separately
- If needed, creates overhead for Working Groups

fon

Progress Update of Smart Home Service Framework

Smart Home Service Objectives

The Smart Home service framework is a new AllJoyn service that enables the provision of connected-home services featuring centralized management for proximal home appliances.

Bedroom   Living room   Kitchen

The central smart home server
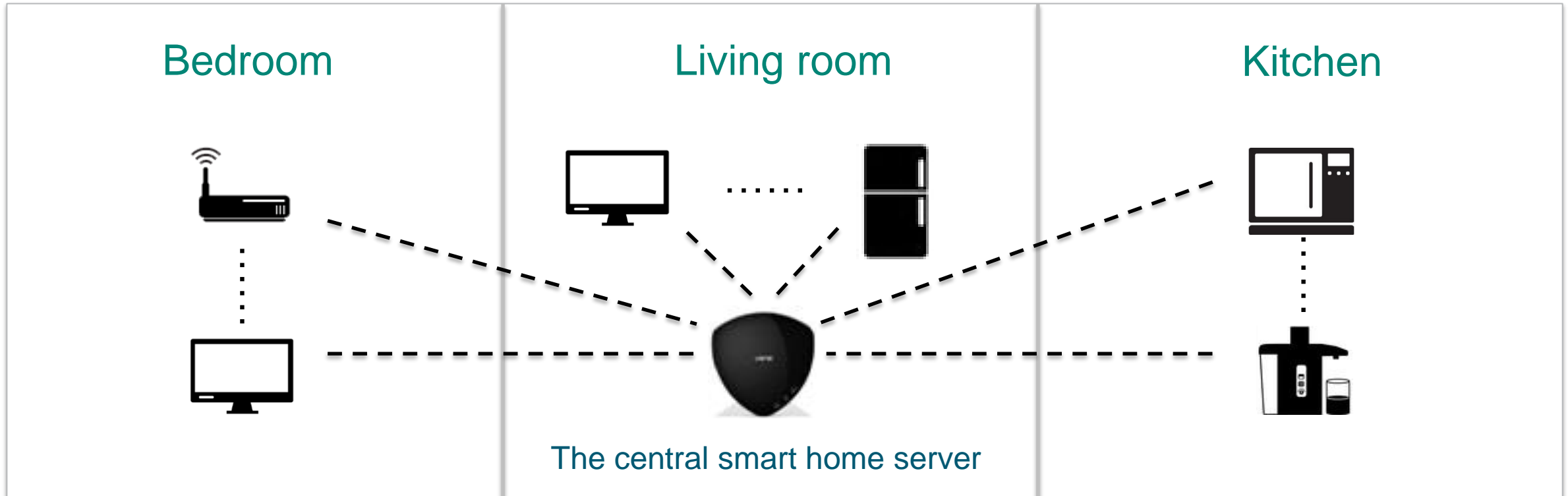
Smart Home Interface

Centralized Management Interface

- What it is
  - Manages home appliances and the user can control home appliances managed by the smart home server
  - The interface is standardized: "org.alljoyn.SmartHome.CentralizedManagement"

- Contains the following methods
  - ApplicationRegistration()
    - Register the appliance interfaces on the smart home server
  - ApplicationUnRegistration()
    - Unregister the appliance interfaces on the smart home server
  - DeviceHeartBeat()
    - Verify whether the appliance is online
  - Execute()
    - Call the appliance method by invoking the Execute() method provided by the smart home server

- Signal
  - ReturnValue
    - Send the data/value returned from the appliance method call to the application

Smart Home Interface

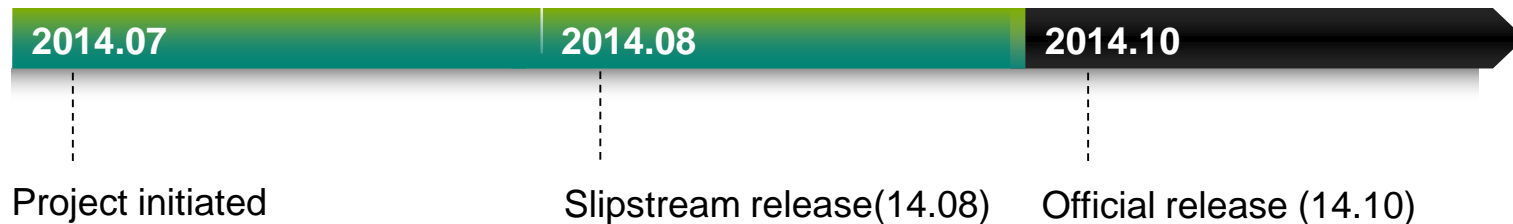## Centralized Management Feature as XML

```xml
<node name="/SmartHomeService"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspect.xsd">
  <interface name="org.alljoyn.SmartHome.CentralizedManagement">
    <property name="Version" type="q" access="read"/>
    <method name="ApplicationRegistration">
      <arg name="wellKnownName " type="s" direction="in"/>
      <arg name="uniqueName" type="s" direction="in"/>
      <arg name="deviceId" type="s" direction="in"/>
      <arg name="objectDescription" type="a(sov)" direction="in"/>
    </method>
    <method name="ApplianceUnRegistration">
      <arg name="deviceId" type="s" direction="in"/>
    </method>
    <method name="DeviceHeartBeat">
      <arg name="deviceId" type="s" direction="in"/>
    </method>
    <method name="Execute">
      <arg name="isReturn" type="b" direction="in"/>
      <arg name="deviceId" type="s" direction="in"/>
      <arg name="objectPath" type="o" direction="in"/>
      <arg name="methodName" type="s" direction="in"/>
      <arg name="arguments" type="v" direction="in"/>
    </method>
    <signal name="ReturnValue">
      <arg name="methodName" type="s"/>
      <arg name="ReturnStatus" type="s"/>
      <arg name="value" type="v"/>
    </signal>
  </interface>
</node>
```

Implementation Status

- Progress
  - Initial implementation finished on August 11
    - AllJoyn smart home server source
    - AllJoyn smart home client source

- Source code
  - https://git.allseenalliance.org/cgit/smarthome/homeservice.git
  - The initial code can be taken as beta release
  - The team is working on stability

Timeline for the initial release

- The project is dependent on the 14.06 release of AllJoyn Core and Base Service.

- The initial code is available at：

  https://git.allseenalliance.org/cgit/smarthome/homeservice.git

- The official release is targeted for October.

2014.07	2014.08	2014.10

Project initiated   Slipstream release(14.08)   Official release (14.10)

Thank You

Follow Us On

- For more information on AllSeen Alliance, visit us at: allseenalliance.org & allseenalliance.org/news/blogs