



**ALLSEEN  
ALLIANCE**

# **Core Working Group**

**February 18, 2016**

The contents of this document, and the interfaces described, and all the information herein, are the result of collaborative discussions by the Core Working Group. This summary documents the final consensus of the team.



**Reminder:  
This call is being  
recorded**

# 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.



# **Move Security Manager project into Core**

Dan Mihai (MSFT)

# Proposing to move Security Manager into Core project

- Main reasons:
- App developers using the SCL need a Security Manager to test the Security 2.0 behavior of their app. So, \*everyone\* who's a user of core/alljoyn should always be a user of securitymgr too – at least until commercial Security Managers will become available.
- It is not uncommon to break the securitymgr behavior when we change the core/alljoyn behavior. We should run the securitymgr tests when we change the SCL, and make sure these tests are still passing.
- Theoretically, at some point the Alliance might decide to make (some of) the securitymgr code part of the SCL. (I am not aware of such a need at this point)



# ASA Dev team update

Josh Spain (Affinegy)

# Affinegy - AllJoyn Core Contributions

- Ramping up team through March 2016
  - Full team available by April 1<sup>st</sup>
- Contributions already started

Areas	Contribution Begins (est.)	Full Team (est.)
Linux/OpenWrt SCL, TCL, C bindings	February 1 <sup>st</sup>	March 1 <sup>st</sup>
Android/Java Bindings	Late February	April 1 <sup>st</sup>
AllJoyn.js	Late February	March 1 <sup>st</sup>
iOS/Obj-C bindings	April 1 <sup>st</sup>	April 1 <sup>st</sup>

# Affinegy – AllJoyn Core Development Plan

- **16.04a**

- iOS 9
- OpenWrt
- AllJoyn.js

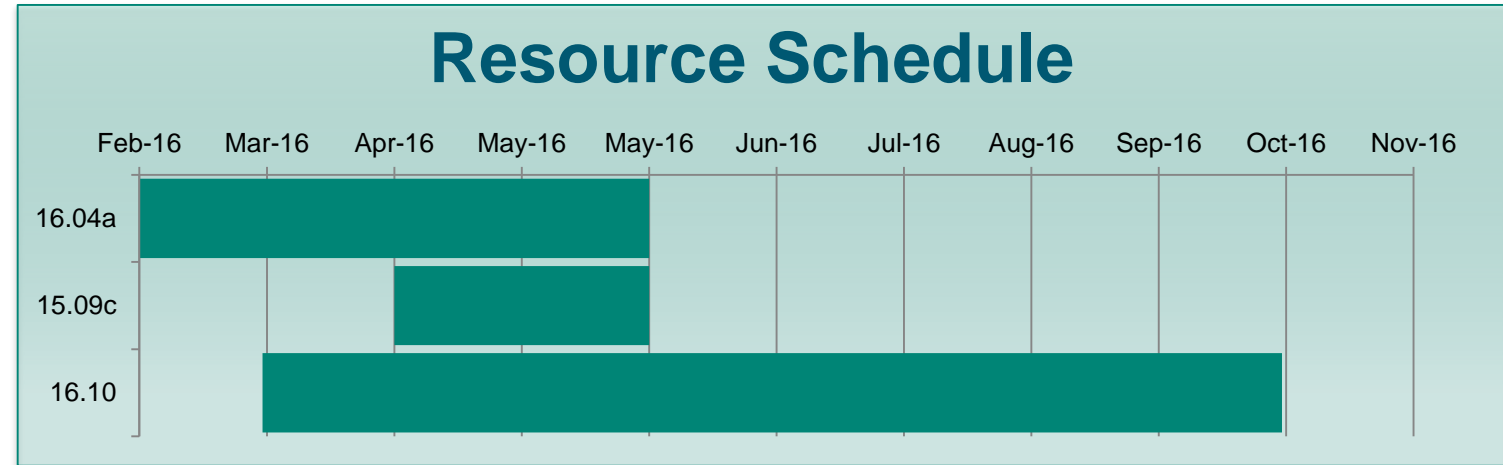
- **15.09c**

- iOS 9

- **16.10**

- Linux SCL/TCL
- C, Java, JavaScript, Obj-C bindings
- AllJoyn.js
- Android, iOS, OpenWrt

- *Security Manager?*







# Roadmap Prioritization

# Roadmap Prioritization

- TCL:
  - Beef up to be able to replace SCL
    - Layer to make API more dynamic (i.e. create AJ objects on the fly)
    - Make the Library re-entrant, this would likely included ability to handle multiple messages
  - Refactor interaction of ARDP with data path
  - Make mDNS packet assembly less fragile

# Roadmap Prioritization (II)

- Router
  - Use performance study to identify goals for scalability: **any updates?**
  - Modify IO architecture to make completely asynchronous
    - help reduce thread count
    - General scalability, performance improvement would be expected
  - IPv6 support
  - Optimize discovery+connect time
  - mDNS protocol compliance
  - Refactor SCL (**this would be shimming these SCL APIs on top of the TCL**)
  - Code sanity improvements – e.g. ASACORE-2235, static analysis, etc.
  - Sample code development
  - Key store redesign
  - Diagnosability improvements
  - Get router out of the data path



# Action Items

# Action Items

- Proposal to outline the process for changing APIs
  - Has been handed over to Brian Rockwell
    - will craft proposal for review by Core WG and then presented to TSC
  - Expectation that this will be reviewed with WG in January.
- Arvind (MSFT) to create Wiki process page linked off of Core WG wiki
  - Remove references to java from mandatory binding list for 16.04 timeframe
  - Hope to have something by the end of Jan.
- Ry (LF): Need to have Alliance running Windows 10 when RTM'd
  - Running the unit tests on the Windows 10 cannot seem to pass
    - Only one set of tests are failing, MSFT has worked with Ry.
    - Issue appears to be the combinations of Windows 10 and the version of VM ware
- MSFT: 16.04 work items, should have a sense by the end of Jan, or early Feb
  - Sec2.0 scenarios, working on C binding for Core Sec2.0

# Action Items

- Add a way in JIRA to track compatibility issues and proposals
  - Proposal is the use a label: “compatibility impact” in the Jira ticket
  - Committers should require for these sorts of change that the release notes be updated
  - **Marcello** to document this process in the Core Process wiki
- Define process to require regression/unit tests for bugs
  - **Marcello** to add description of Committers requiring unit test for bug fixes to Core Wiki
- Define process for handling “Technical Debt”
  - Example: Took a shortcut to make a release and not loosing track of this to fix the shortcut
  - Jira items to track when things are missing – e.g. feature added, but missing from Java binding
  - If shortcut is taken: could create a Jira task to track the problem



# Thank You

Follow Us On      

- For more information on AllSeen Alliance, visit us at: [allseenalliance.org](http://allseenalliance.org) & [allseenalliance.org/news/blogs](http://allseenalliance.org/news/blogs)