



**ALLSEEN  
ALLIANCE**

# **Core Working Group**

**February 11, 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.



# AllSeen API Change Process

Brian Rockwell (MSFT)

# Non-breaking Changes

## Backwards Compatibility

For any given release N, changes to APIs should be verified not to break clients built with any major release version between N and N-2. After a total of four releases since deprecation is initiated, the API can be removed.

Example: For a change to the version N API, the following would apply: (released bits in SDKs)

- N: **change is made**: No breakage
- N+1: No breakage
- N+2: **Technically unsupported (may be broken)**
- N+3: Technically unsupported (may be broken)
- N+4: API removed

# Breaking Changes

Communication, Coordination, Lead Time

DL: allseen-breaking-changes

When introducing breaking changes, the following processes steps need to be taken ahead of time:

1. Coordination with Working Group to pre-approve the change
2. If pre-approved: Send email the [TBD: not yet created] “AllSeen Breaking Changes” DL (public) outlining the suggested change, request for feedback.
3. The TSC will wait a minimum of 5 business days between the breaking email is sent and the TSC vote to approve/reject the change.
4. Send email the [TBD: not yet created] “AllSeen Breaking Changes” DL (public) outlining the results of the TSC vote.
5. The code with breaking change can be merged.

# Breaking Changes

## Annotating/Depecrating APIs

Removal of APIs requires TSC approval. If approved, any APIs to be removed need to be marked “Deprecated” (more information: [here](#)). Deprecated APIs shall be supported for two official releases, and can then be removed after 4 official releases.

Example: For the deprecation of an API in the N release, the following would apply:

- N: Marked deprecated, supported
- N+1: Marked deprecated, supported
- N+2: Marked deprecated, not supported
- N+3: Marked deprecated, not supported
- N+4: Removed

# Testing Changes

In order to ensure that appropriate fixes are submitted with a breaking change, the following validation should be performed:

- Ensure that all related tests (including dependent frameworks) are run by Jenkins as a part of the Gerrit build.
- Submit code to Gerrit and verify test results from build (only tests in Jenkins will apply)
- End to end testing for interoperability (allseen\core\test\tcl\tc-sc-interaction) between any related Standard Client and Thin Core components (Ex: Thin Core producer with Standard Core consumer), router node and leaf node for:
  - Current codebase.
  - Previous AllJoyn releases (in supported window of releases).
  - All mandatory language bindings (C and C++)
- Committer will determine if any additional tests need to be run.



# Testing Changes

## Ownership of fixes

When regression to behavior that was in a release has been introduced, the submitter making the breaking change(s) should submit fixes for any related breaking changes in the same submission. The working group can help coordinate fixes as appropriate.

# Open discussion

- Arvind: Timeline for 16.04 MSFT targeting last feature merge 3/18 – that would be when branch is cut
  - Bug fixes only after that, with the bar raising in towards the release
- Dan: proposal to list StartManagement/EndManagement
  - Feedback has been positive.
  - Next step is to get IRB, and update HLD
  - IRB link: <https://git.allseenalliance.org/gerrit/#/c/4283/>



# 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)