ALLSEEN ALLIANCE

Core Working Group

September 03, 2015

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.

Agenda

1. 15.09
   - Features
   - Schedule
   - Platforms
2. Platform Support Process Proposal
3. F2F Agenda
4. Review Action Items
5. Post Mortem action items

15.09

Schedule

- Sept 04
  - Regression testing complete
    - Not including Windows 10
- Sept 18
  - Windows 10 regression testing complete
    - Plan to start Sept 14
      - 1 week effort
- September 30 is the official software release
- Other milestones
  - Release branch
    - Cut on September 10
  - Ticket deferral
    - Defer Major and below during September 10 triage
    - Defer Critical during September 17 triage

Platform Support Process Proposal

Platform Support Process

- A supported platform is:
  - Actively maintained and tested prior to release
    - Platform test cases are documented in Test Link
  - Included in continuous integration (automated testing and builds)
  - Project committers will address issues that arise
- Current supported platforms are listed here:
  - https://allseenalliance.org/developers/download/supported-platforms
- Platforms are defined in terms of:
  - OS (with version), CPU architecture, toolchain (with version), API (where needed)
- Other platforms or variations may be included in the code base and still build and run, but are not verified. For example, older compiler versions or newer versions of the OS may work fine. Bug fixes and feature porting may be on a best-effort basis from committers or contributors.

Guidelines

- Platforms should be “released” versions (not beta) at least 1 month prior to the expected AllJoyn Core release.
  - Unless a platform is dropped, expectation is that support for a platform would continue to move forward as the platform evolves.
- Each platform requires a sponsor committer who takes responsibility for coordinating efforts for that platform.
- These issues should be considered well in advance in order to account for the long lead times to deploy automated builds and identify resources.

Timeline

- ~6 months prior to release:
  - Schedule a core WG meeting where platform changes will be discussed and disclose the meeting time in a TSC meeting.
  - Project maintainers with specific platform needs should follow the core mailing list for platform status and to participate in the selection process.
  - Contributors email platform change or addition proposals to allseen-core for preliminary discussion.
    - Include OS (with version), CPU architecture, toolchain (with version), API (where needed), core project (SC or TC). If this is an update to an existing platform, mention that.
- ~5 months prior to release: Core WG reviews platform proposals as part of a Core WG meeting. A majority of the project committers agree to the set of platforms and sponsor committers for the next release.
- >3 months prior to release: Publish Release Plan, including platform list. Present platforms in TSC meeting to ensure visibility, but there isn’t a TSC vote.

Documentation

- Core Release plan will list planned platform support
- The “Supported platforms” page (https://allseenalliance.org/developers/download/supported-platforms) is the authoritative list of supported platforms for recent releases
- This process will be documented on the wiki.

F2F Agenda

F2F Call for agenda items

- 2 separate meetings
  - Education for Alliance members
  - Discussion amongst Core WG members
- Education for Alliance members
  - What we have done so far (state of the union)
    - Scalability testing
    - Process definition
  - Tech overview of Security 2.0
    - Dave Thaler (MSFT) can support
  - 16.04 feature set
    - List of possible features
- Discussion amongst Core WG members
  - Agenda being worked

Action Items

Action Items

- Proposal to cancel Monday (Sept 7) triage meeting
  - US Holiday (Labor Day)
- Proposal to outline the process for changing APIs
  - Gavin (MSFT) will craft proposal for review by Core WG and then presented to TSC
  - Planned for September 03
- 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
  - Will be done middle of September
- Ry (LF) setting up a Windows 10 VM
  - Ry is has MSDN license
- Chris (QCE) to add Windows 10 AJ router discussion as agenda item week of Sept 7

Post Mortem Action Items

Post Mortem Action Items

- Ry (LF): Need to have Alliance running Windows 10 when RTM'd
- Check readme files as part of the release procedure
  - Core WG needs to decide who will own the specific readme files
  - Need to partition them out by platform
    - Marcello (QCE) will do an inventory to determine how to partition them out
- Set a project milestone to begin integration branch merges
- Add a way in JIRA to track compatibility issues and proposals
- Define process for handling “Technical Debt”
  - Example: Took a shortcut to make a release and not loosing track of this to fix the shortcut
- Track feature branches and add more spacing between feature branch merges
- Define process to require regression/unit tests for bugs

Discussion

- …

Thank You

Follow Us On

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

15.09 Platforms – Release Testing

- AllJoyn Standard Library:
  - Full regression test
    - QCE
      - Linux Ubuntu 14.04 LTS (64 bit)
      - Android Lollipop 5.0 (ARM)
      - OpenWRT Barrier Breaker (BB) branch
    - MSFT
      - Windows 10
  - Smoke test
    - QCE
      - Android JellyBean 4.1 (ARM)
      - Android KitKat 4.4 (ARM)
      - OpenWRT Chaos Calmer (CC) branch
- AllJoyn Thin Library:
  - Full regression test
    - QCE
      - Linux Ubuntu 14.04 LTS (64 bit)

15.09 SDKs

Included	
SDK	**Toolchain Used**
alljoyn-15.09.00-src.tar.gz	N/a - source
ajtcl-15.09.00-src.tar.gz	N/a - source
Core SDK - release (android)	Android NDK r10e, Oracle Java 7
Core SDK - debug (android)	Android NDK r10e, Oracle Java 7

Not Included	
SDK	**Toolchain Used**
Core SDK (osx/ios)	Unless a contributor is able to perform the platform tests then these binaries will not be distributed. Note: We will keep building it and include the OSX and iOS builds in the verification paths
Windows SDK (64-bit) (VS2012)	Visual Studio 2012, Oracle Java 7 (assumes contributor performs regression testing)
Windows SDK (32-bit) (VS2012)	Visual Studio 2012, Oracle Java 7 (assumes contributor performs regression testing)
Windows SDK (64-bit) (VS2013)	Visual Studio 2013, Oracle Java 7 (assumes contributor performs regression testing)
Windows SDK (32-bit) (VS2013)	Visual Studio 2013, Oracle Java 7 (assumes contributor performs regression testing)
Windows Thin Core SDK (zip file with source and stand alone router executable)	Visual Studio 2013 (assumes contributor verifies this works)

Backup: Supported bindings

- Existing bindings in Core
  - C++
  - Obj-C
  - Java
  - C
  - Javascript NPAPI

Language bindings discussion 5/7/15

- Need a formal policy to support language bindings
  - Came up at last TSC F2F
- SCL binding proposal
  - Required
    - C++ and C
  - Optional
    - Java, NPAPI, ObjC?
- TCL binding proposal
  - No proposal
  - Should we consider JavaScript?
- Need separate policy for platforms
  - May need to consider binding platform and language binding
- Next steps
  - Action: Marcello (QCE) to send proposal to the mail list

Notes from 14.12 Post Mortem

14.12 Post Mortem Improvement Items (1/3)

- Aligning date & the end game (lockdown) schedule of AllJoyn releases with release schedule of the contributing member companies if it happens to be in close proximity of AllJoyn release
  - **Action**: Arvind to send proposal to Core WG mail list
- Consistent and enforced definition/bar for code freeze
  - Need crisp definitions for "incremental bug bars“ (normal, tell, ask)
  - Need approval granularity (approval on merge)
  - Need process for how to deal with large last-minute changes
    - Suggestion – only high priority issues “ask” are added one week before release
  - **Action**:
    - Marcello to send current milestone definition to Core WG mail list
    - Gavin to send proposal to Core WG mail list based on Marcello’s email
- Need processes for breaking changes
  - Regarding protocol, API syntax, behavior
  - Mitigation: Proposed changes should be advertised
  - **Action**: Chris to add the process to this to the existing process draft

14.12 Post Mortem Improvement Items (2/3)

- Need agenda and slides 48 hours ahead of core WG meetings
  - Best effort to send slides by COB Friday
- Need notes from core WG meetings sent more consistently
  - Note: Linux foundation unable to assist
  - **Action**: Gavin to see if someone from Microsoft can assist
- For TSC: PR coordination for releases
  - More an issue for marketing committee
  - **Action**: Chris to discuss this with Philip
- Engage system test during feature testing
  - More members conducting system test is preferred
- Testing needs to be better distributed across members
- E2E testing is needed
  - **Action**: David and Arvind will make a proposal
- More frequently merge feature branches so that deltas can be kept to a reasonable minimum
  - **Action**: Chris to add process to the Wiki process draft

14.12 Post Mortem Improvement Items (3/3)

- May need more frequent but shorter Core WG meetings
  - WG meeting immediately following triage meeting for 30 minutes
  - **Action**: Chris to change Core WG status meetings to 30 minutes after the Thursday triage
- Increase frequency of triage meetings earlier in the process
  - **Action**: Chris to set up biweekly triage 2 weeks prior to branch date
    - For 15.04 it will begin week of March 9
- Define JIRA severity vs. priority process
  - **Action**: David & Arvind to set up discussions to craft proposal
- To be discussed at next meeting
  - JIRA label to identify contributing organization taking ownership of the item
  - Need process for managing the platform matrix
  - Revisit code style guidelines, rules, and enforcement

Features

- Major features (Committed for the release)
  - AJCORE-1393 Security 2.0 (MSFT, QEO, QCE)
  - AJCORE-1686 Commercialize UDP Transport for TC <-> RN connections (QCE)
- Full list of features
  - https://jira.allseenalliance.org/issues/?filter=11411