ALLSEEN ALLIANCE

Core Working Group

July 02, 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. 15.08
   - Features
   - Schedule
   - Platforms
2. Proposal for Language Binding Policy for AllJoyn SCL
3. Deprecation of qcc::String in favor of std::string
4. Review Action Items
5. Post Mortem action items

15.08

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=11008
  - MSFT to scrub their features/tasks and discuss on 7/2
    - Will be reflected in JIRA at triage

Schedule

- July 24 Cut release branch
  - Merge branches
    - Sec 2.0 feature branch merged
    - Thin client reorg branch
    - Any additional branches?
  - All feature testing completed
  - Only regression testing remains
- August 28 Release
  - Completed regression testing one week prior
  - All blocking tickets closed
  - Most critical tickets closed
- Other milestones
  - Ticket deferral
    - Will firm on a date at next weeks call
  - Twice weekly triage July 20
    - 10am PT Monday and regular Thurs
- Proposal to move to end of Sept 15.09
  - MSFT, QCE &QEO in agreement
  - Need to present to TSC – Chris (QCE) to add slides for Monday's call

15.08 Platforms – Release Testing 1/2

- QCE testing commitment for 15.08
  - AllJoyn Standard Library:
    - Full regression test
      - Linux Ubuntu 14.04 LTS (64 bit)
      - Android Lollipop 5.0 (ARM)
      - OpenWRT Barrier Breaker (BB) branch
    - Smoke test
      - Android JellyBean 4.1 (ARM)
      - Android KitKat 4.4 (ARM)
      - OpenWRT Chaos Calmer (CC) branch
  - AllJoyn Thin Library:
    - Full regression test
      - Linux Ubuntu 14.04 LTS (64 bit)

15.08 Platforms – Release Testing 2/2

- Call for contributors to provide release testing
  - AllJoyn Standard Library:
    - Full regression test
    - Smoke test
  - AllJoyn Thin Library:
    - Full regression test
- Action: Close on platforms and SDKs for 15.08
  - Gavin (MSFT) will discuss at the next Core WG call regarding MSFT testing
    - Mathew (QCE) will ask about download stats on Core SDKs

15.08 SDKs

SDK	Toolchain Used
alljoyn-15.04.00-src.tar.gz	N/a - source
ajtcl-15.04.00-src.tar.gz	N/a - source
Core SDK - release (android)	Android NDK r10e (**note r10e is a change – no objection from Core WG**), Oracle Java 7
Core SDK - debug (android)	Android NDK r10e, Oracle Java 7
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)

Not included: 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

Proposal for Language Binding Policy for AllJoyn SCL
NOTE: Will discuss on 7/9 call

Arvind Padole
Microsoft

Guiding Principle

Ensure that one or more language bindings built on top of C++ API have feature parity with native C++ API

- The assurance of feature parity with native API will provide AllJoyn developers the flexibility of choosing alternate bindings where C++ API may not be the preferred option
- The language bindings that are thus intended by the alliance to have feature parity with the core C++ API will be referred to as **Mandatory Language bindings**
- All other bindings are **Optional Language Bindings**
  - Optional bindings are fully supported by the alliance except that they may not be as feature rich as the mandatory bindings

Contributing to AllJoyn Core SCL

- New AllJoyn Core contribution (feature or bug fix) that involves updates to C++ APIs must include corresponding updates to the mandatory language bindings including unit tests and samples as applicable
  - Specifically, such contribution will be considered incomplete and release blocking until all mandatory language bindings are updated to reach feature parity with the core C++ API
- Updates to mandatory bindings may be submitted separately from core C++ API. In such cases, updates to each mandatory language binding must be tracked using separate Jira tickets
- Updates to mandatory bindings may be contributed by two or more contributors who may be different from the contributor for original C++ API updates. In such cases, updates to each mandatory language binding must be tracked using separate Jira tickets assigned to the corresponding contributors
- If C++ API updates are merged before mandatory bindings, Jira tickets for mandatory binding updates must be marked as release blocking by setting the Severity to Blocking

Classifying a language binding as mandatory

- The alliance needs to have objective criteria for determining if a language binding should be part of the mandatory binding list. The proposed criteria is as follows

The language binding is broadly applicable and supported on all supported platforms

- Language bindings that currently meet the above criteria
  - C
  - Java (**Question** – is AllJoyn SCL for Java available on all platforms?)

Adding new language binding to the mandatory binding list

- The proposed addition is first evaluated and approved by the Core WG and is then sent for TSC approval
- TSC evaluates potential impact of the addition including impact to other WGs if any and makes the final determination
  - If Core WG supports a superset of the mandatory binding list for all WGs then adding a new mandatory binding to the Core WG list may not have implications on other WGs and thus may not need TSC review and approval

Removing a language binding from the mandatory binding list

- The proposed removal is first evaluated and approved by the Core WG and is then sent for TSC approval
- TSC evaluates potential impact of the removal including impact to other WGs if any and makes the final determination

Deprecation of qcc::String in favor of std::string

Deprecation of qcc::String in favor of std::string

- I would like to deprecate qcc::String in favor of std::string. The reasons for doing so include that it duplicates standard functionality, forces inefficiencies on programs that have to convert between qcc::String and std::string, and has been a source of bugs that could have been avoided if we used std::string.
- Deprecating qcc::String in favor of std::string will require TSC approval since it impacts the public API in AllJoyn Core. It will also take longer to finally remove since it provides a method, secure_clear, that is not provided by std::string that will need to be deprecated and removed first. The proposed process will be as follows:
  1. Deprecate qcc::String::secure_clear() - remove its usage from the one place where it is used (alljoyn_core/src/windows/KeyStoreListenerFactory.cc)
  2. Modify qcc::String to just be a wrapper for std::string
     - This would include introducing functionality to support mixing qcc::String and std::string. Hopefully a type conversion operator to convert qcc::String to std::string will solve most of the mixing problems.
     - The deprecated version of qcc::String::secure_clear() will need to use const_cast<> to cast away const from the point obtained from std::string::data() so that it can wipe the memory before calling string::clear(). Hopefully, compilers won't warn about casting const away.

Deprecation of qcc::String in favor of std::string

4. 4 major releases later, remove qcc::String::secure_clear()
5. Deprecate qcc::String - remove from public API usage
   - Also, change qcc::String to a typedef of std::string.
6. 4 major releases later remove qcc::String

- Once the code to convert qcc::String to std::string and back is in place then internal code can start to be changed to use std::string. Also, all new APIs would need to use std::string as well. By making qcc:String a wrapper around std::string, this allows for trivially easy conversion between the 2 that compilers can easily optimize out.
- The reason for deprecating qcc::String in 2 phases allows for other dependent projects more time to adjust to the change and it allows for std::string to be a drop-in replacement for qcc::String when qcc::String gets deprecated.
- Fortunately, all the method names in qcc::String are identical to those in std::string (with the exception of secure_clear).
  - (Note that the deprecated implementation of secure_clear() cannot use std::string::replace() when qcc::String is just a wrapper for std::string since the replace method will most likely create a new string rather than change the underlying memory which is the intent of secure_clear.)

- Action: Steve (QCE) to send summary of agreement to the Core WG

Action Items

Action Items (1/2)

- David (QCE) to lead team crafting testing proposals
  - Charters crafted for RFP committee and test policy committee and sent to mail list for review
- Marcello (QCE) to send IOS language binding proposal to WG mail list
  - Agreement is to stop supporting the language binding
  - Todd to see if Vlad wishes to take ownership of this binding
    - Vlad will fix bugs he cares about but not take ownership
  - Next step: Inform TSC
- Proposal to outline the process for changing APIs
  - Gavin (MSFT) will craft proposal for review by Core WG and then presented to TSC
  - Will review document by end of June
- Gavin (MSFT) to discuss features/tasks on 6/2
- 15.08a patch release?
  - Arvind/Gavin (MSFT) to communicate dates for items they want added
  - Kristof (QEO) to determine impact to QEO if 15.08 is moved out to accommodate additional items

Action Items (2/2)

- Gavin (MSFT) will discuss MSFT regression testing for 15.08
- Dan (MSFT) to submit proposal to Core WG mail list about possible deprecation of ajn::BusAttachment::Connect(connectspec) overload
  - Dan (MSFT) will start investigation in July
    - Dan (MSFT) to create JIRA ticket after investigation
- Marcello (QCE) to add components to JIRA for docs
  - Added on 6/29
- Steve (QCE) to send summary of agreement for Deprecation of qcc::String in favor of std::string to the Core WG mail list
- Chris (QCE) to create JIRA tickets for updating samples Gavin (MSFT) refactor toaster sample
  - This is the main ticket tracking the sample cleanup
    - https://jira.allseenalliance.org/browse/ASACORE-1528
    - Subtask created: https://jira.allseenalliance.org/browse/ASACORE-2095
      - Obtain list of samples QEO and MSFT wish to keep
        - » QEO has update the ticket with their request
      - Identify samples needed by test and remove remaining

Post Mortem Action Items

Post Mortem Action Items (1/2)

- C Bindings
  - Need clear ownership and responsibilities documented
  - Need better alignment on binding strategy all-up (AllSeen)
- Need to have Alliance running Windows 10 when RTM'd
- SCONS needs to support VS 2015
- Add ASADOCS for Core to the triage
- Check readme files as part of the release procedure
  - Core WG needs to decide who will own the specific readme files
- Set a project milestone to begin integration branch merges

24

Post Mortem Action Items (2/2)

- Add a way in JIRA to track compatibility issues and proposals
- Need to collaborate on stress/system testing
  - Add a track on system testing at F2F in October
- More robust automated testing
  - Investigation of occasional failures
  - Better feature/code coverage
  - Integration test scenario’s covering multiple features on multiple platforms
- Process for handling “Technical Debt”
- Track feature branches and add more spacing between feature branch merges
- Need more eyeballs on security for implementation and test
- Add 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

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