



**ALLSEEN
ALLIANCE**

Technical Steering Meeting

July 12, 2016



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. Approve minutes from previous meeting
2. SDPX – do we want to do this?
3. Smart Spaces 16.10 planning update
4. Base Services Developer update



SPDX: do we move forward?

What is SPDX?

Standard:

- A standard format for communicating the licenses and copyrights associated with software packages

Guiding principles:

- Human and machine readable
- Focus on capturing facts; avoid interpretations

Vision:

- To help reduce redundant work in determining software license information and facilitate compliance

What would we need to do?

- Update source to include the SPDX license information
- Created the SPDX documents that describe our packages
- Like require the used of paid resources to make this happen
- **Is this something the TSC would like to do?**



Smart Spaces

- 16.10 planning

Our 16.10 goals from the TSC

- **Common Device Models**

- Data models for Light, Lock, Switch, Door Sensor, Window Sensor, Smart Plug, Speaker, Water Sensor, Smoke Detector, Motion Sensor
- Device simulator
- Full platform support (iOS, Android, Linux, OpenWRT, Windows 10)

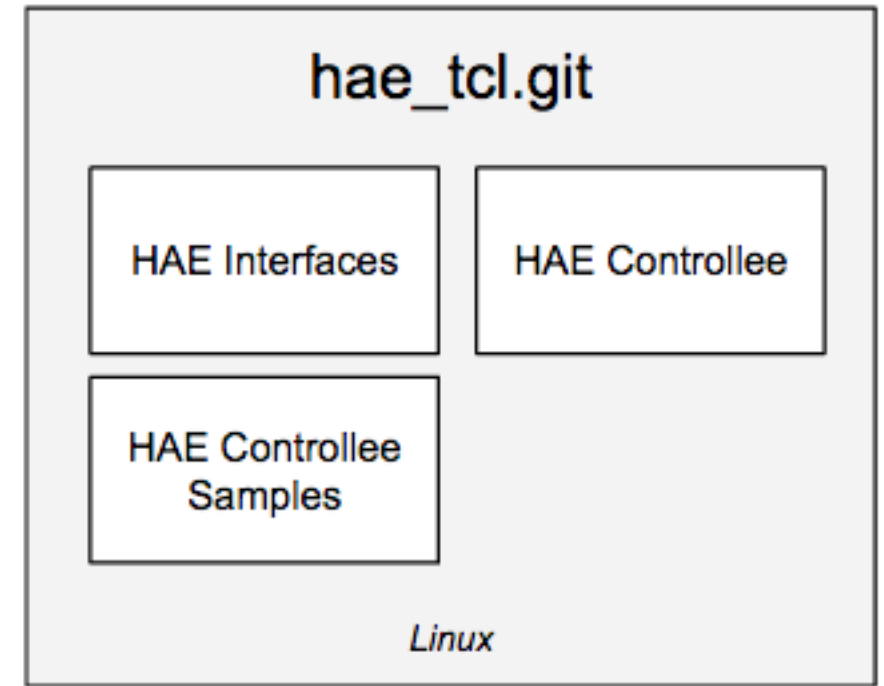
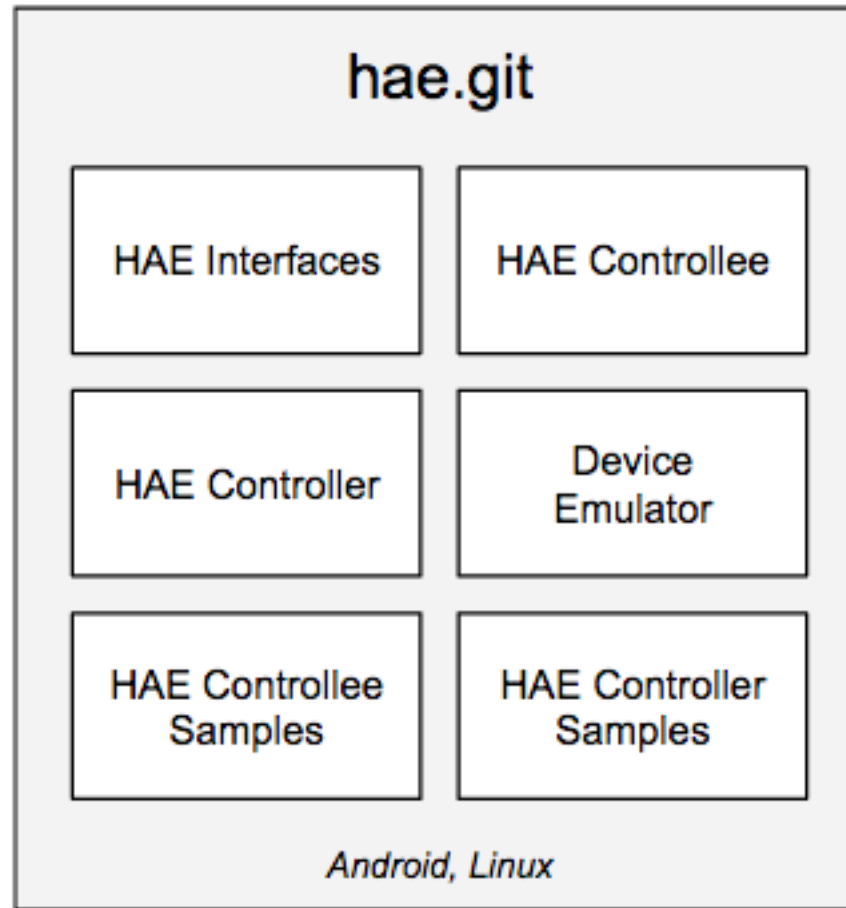
- **(Stretch) Common Controller**

- Groups
- Presets

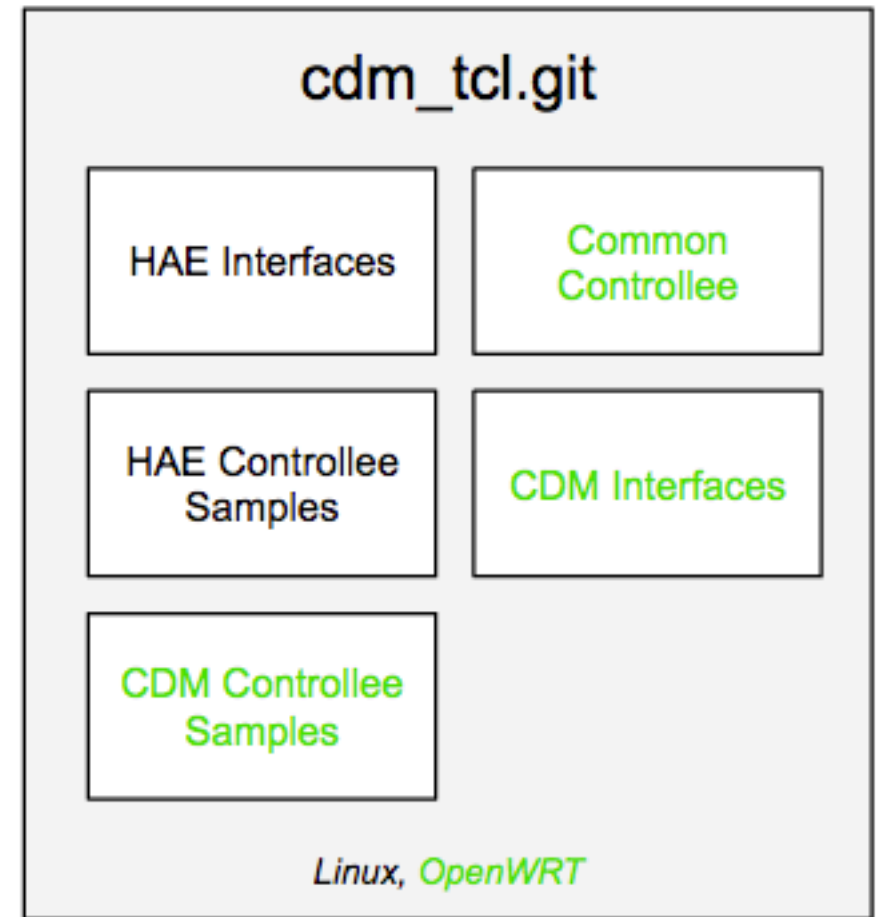
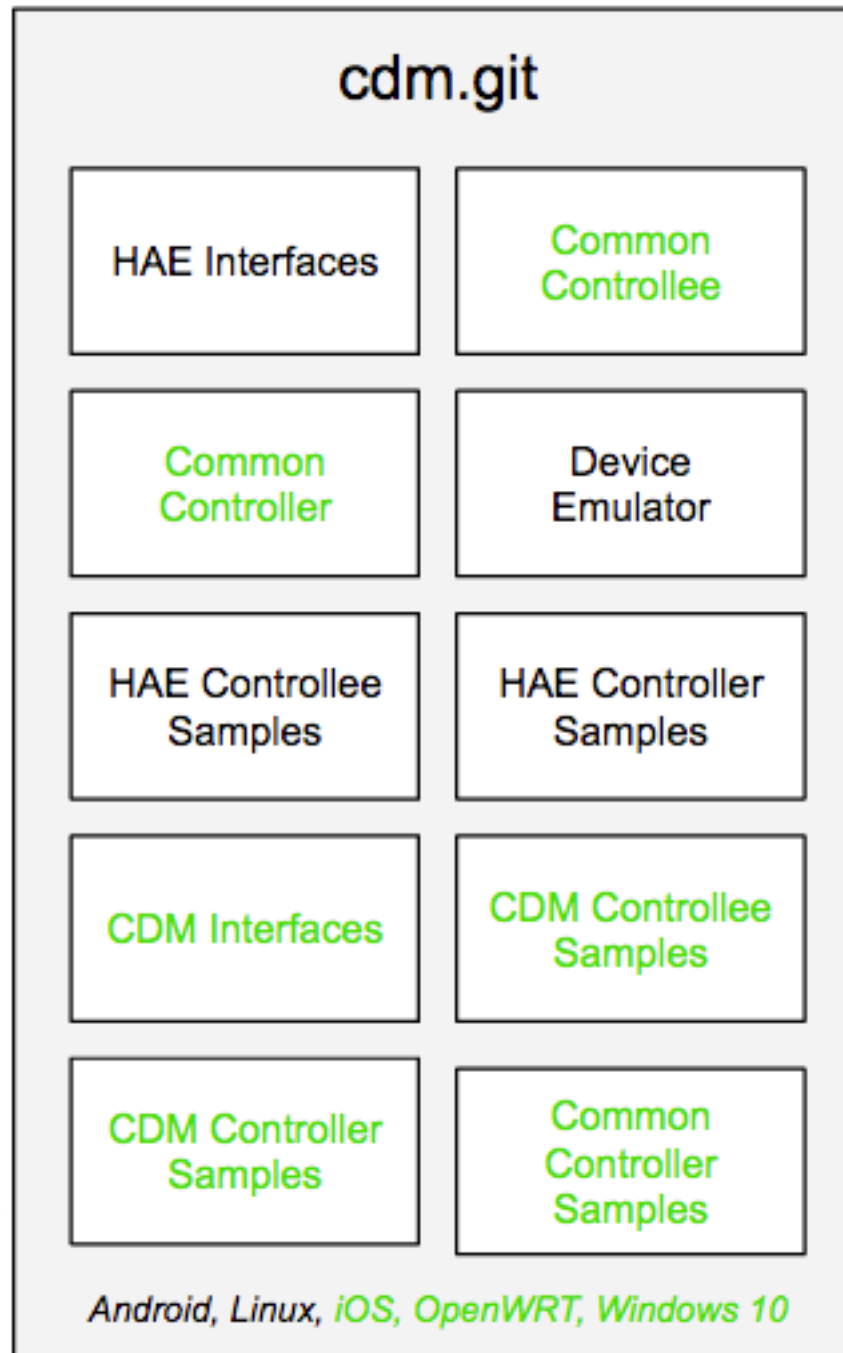
Premise

- We love the work that has been done in HAE and want to leverage it
- We want to avoid duplicating any effort
- We want it to be simple for developers to use great features in AllJoyn and for it to not be scattered around in different places

HAE currently



HAE merges with Common Device Models



Notes

- Merge HAE and Common Devices Models and use AllSeen funded developers to develop it into the premier Smart Spaces solution
 - Current suggestion is that we keep the name Common Device Models. **Does anyone on the TSC have a better suggestion?**
- One standard Common Controller and Common Controllee but with optional components
 - We intend to extend this project to include Common Controller so that there is less overhead from multiple projects and more centralization for developers. **Does anyone on the TSC object to this?**
- We keep the great work done already in HAE and just build on it
- We add full platform support (iOS, OpenWRT, Windows 10)
- Utilize code generation wherever possible to minimize maintenance work
- Further verticals (automotive, industrial etc) can be added to the project in the future

Common Controller components

- Controls
 - Automatic interface generation based on semantic information from device interface
 - Based on the HAE Controller that currently exists for Android
- Presets
 - A set of stored actions (or property writes) for devices that can be triggered as an action
- Groups
 - Logically group devices
 - Messages sent to the group will only be received and actioned by devices that respond to them
- Living Scenarios
 - Set of predefined events that can be configured to trigger presets

Common Controller syncing

- In a perfect situation you would have multiple Common Controllers on the networks, they would stay in sync and a leader would be designated
- In order to reduce initial scope we aim to only support a single instance on the network
- We will provide a management API that developers can use to inspect and modify each Common Controller instance so they can layer their own syncing or coordination on top
- This can then eventually evolve into the full fledged solution

Semantic Annotations

- One thing we aim to add more of in the CDM interfaces and hopefully HAE interfaces is semantic annotations
- They can be used to communicate semantic information about events, actions and properties eg.
 - Whether an element is intended for use by a user or machine:
 - `<annotation name="org.alljoyn.smartspaces.usage" value="User"/>`
 - Specifying human readable names for true and false boolean values:
 - `<annotation name="org.alljoyn.smartspaces.false" value="Off"/>`
 - `<annotation name="org.alljoyn.smartspaces.true" value="On"/>`
 - Many more exciting possibilities
- Allows great user experiences to be built on top of AllJoyn without over specifying
 - For example this could support App UX just as well as Voice UX or Augmented Reality UX

Subsequent Deprecations for the Future

- **Lighting Service Framework**

- Light interface will be replaced by the Light interface in CDM Interfaces
 - We aim to try for backwards compatibility
- Groups, Presets could be achieved in Common Controller components

- **Control Panel**

- Could be achieved better by the work done in HAE Controller that will be extended into the Controls component of Common Controller

Next Steps

- Two Bulls has already started work based on this roadmap since HAE is in agreement.
- Next TSC meeting I will propose some votes to the TSC to do any structural changes to projects or increase their scope if needed. **Marcello/Brett what will be required for this?**
- HAE will continue to release 16.04 with Inhwon Choi as the lead but we aim to rename the project before release.
- Post release Two Bulls will become the maintainer of the project and merge in our 16.10 work.
- Two Bulls will continue to maintain Base Services in Common Frameworks WG



Base Services

16.04

- June Progress Report and July Plan

Resourcing

- Aaron Vernon (Project Manager) ~**15%**
- Daniel Stroud (Software Architect) ~**15%**
- Jarrod Moldrich (Lead Software Engineer) was **100%** now **0%** (finished up end of June)
- Nathaniel Garbutt (Lead Software Engineer) **100%** (back from paternity leave)
- James Lee (Software Engineer) **80%**
- Jayden Ivanovic (Software Engineer) **100%**
- Cameron Harvey (Software Engineer) **100%**
- Brian Hollister (Software Engineer) **100%**
- Ben Kersten (Software Engineer) consulting as required
- Brett Rogers (Technical QA) **100%**

June Actuals

	Total Days
Product Owner	
Aaron Vernon	4
	4
Developers	
Ben Kersten	0
Brian Hollister	21
Cameron Harvey	20.5
Daniel Stroud	2
Jarrold Moldrich	20
Jayden Ivanovic	17.5
James Lee	14
	95
QA	
Brett Rogers	20
	20
	119

June - Summary of work

- **42%** - Code cleanup and deprecation, essentially paying down technical debt in Base Services for standard and thin core.
- **20%** - Planning the work we need to do for 16.10 which involved a lot of investigation of HAE and LSF. Our planning has been done in conjunction with Inhwon and HAE.
- **12%** - Various small issues that were either identified as high priority or more efficient to fix than to leave alone
- **10%** - Onboarding maintenance, bugs that were identified when we lifted the bar on testing for the onboarding samples
- **8%** - Improving documentation of Base Services
- **8%** - Build server and systems improvements

June 2016 - Jira Issues update



- **58 issues** are new
- **18 issues** are in progress
- **74 issues** have been resolved
- There are currently **61 open issues**

July 2016 - Plan

- We have the support of HAE for our 16.10 plan and so we are working on 16.10 whilst they are finalizing 16.04
 - As per our Common Device Models plan we will start with Light, Lock and Switch interfaces and aim to have them ready for August
 - We don't anticipate that all 5 developers will be required on Common Device Models so some may start working on Common Controller



Thank you

Follow us on  

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