ALLSEEN ALLIANCE

Technical Steering Meeting

April 29, 2014

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.

Reminder:

This call is being recorded

Agenda

- Approve minutes from last call
- Developer Documentation
  - Proposal for documentation revamp
- Gateway working group proposal
  - Presentation by Art Lancaster
- Data-driven API Proposal discussion
  - Working group or project under core or other WG?
  - Vote on approval to proceed

Developer Documentation Revamp Proposal

Wayne Lee

Overview

- Looking to revamp AllSeen Developer content starting in 14.06 timeframe

- Need agreement on:
  - – Content organization
  - – Tool and Process
  - – Ownership

What’s wrong with what we have?

- Organization is outdated
  - AllJoyn has grown, need reorg to more easily find content
- AllJoyn is multi-dimensional
  - Not enough intro, hand-holding content
- 100+ PDFs, OMG
  - Lots of repeated content (aka unnecessary noise)
  - Too much work to change one character
  - Developers prefer web content, not PDFs.
- Not set up for collaborative changes

High Level Concepts

- Developers versus Contributors
  - those **using** versus **contributing** to the software

- Keep it simple
  - Start basic and incrementally introduce complexities
  - Goal: running in 5 minutes

- Web content, not PDFs

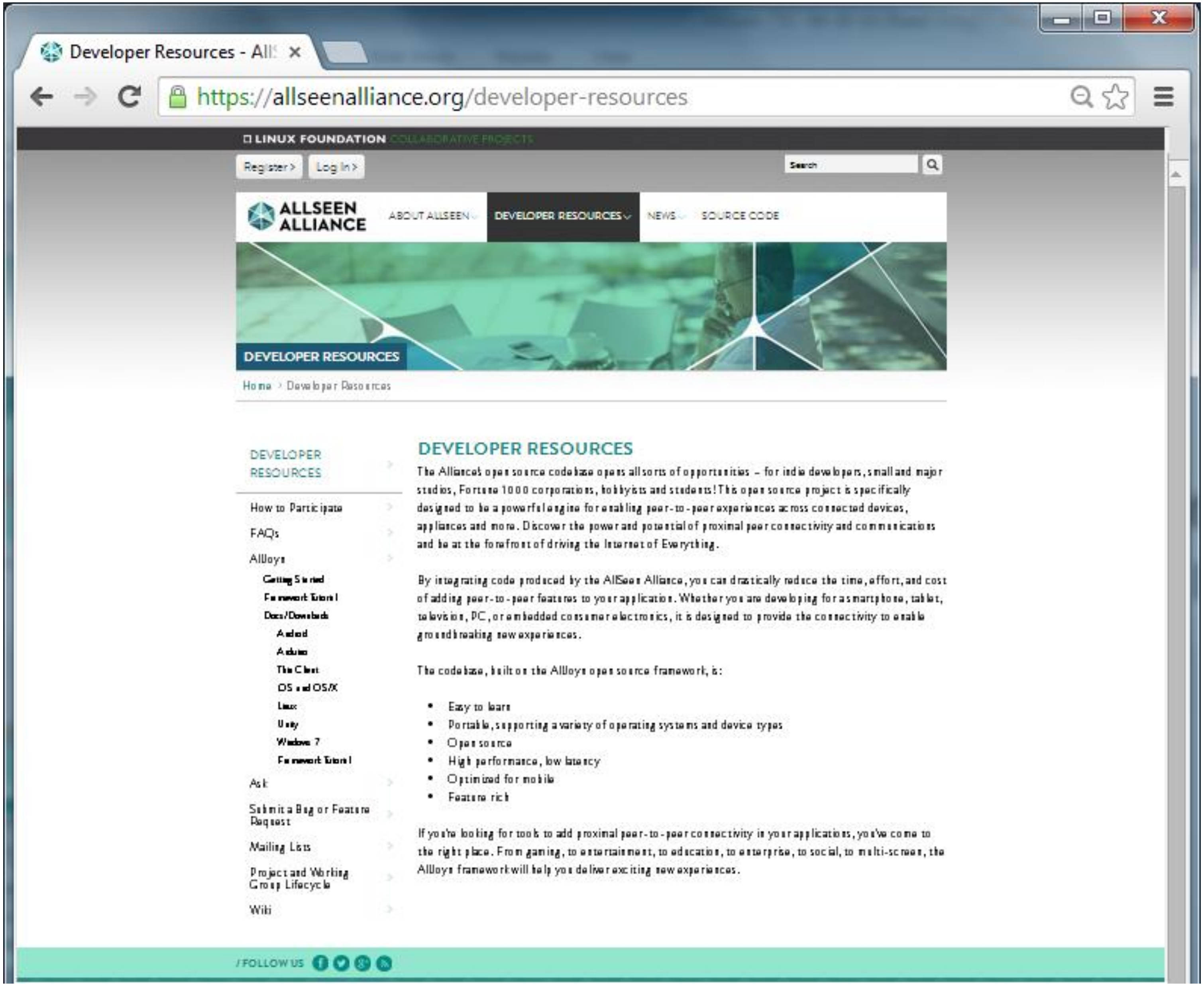
- Collaborative tool for easy, distributed, controlled updates

Two topics to discuss

- Content Organization
- Tool and Process

Content Organization Proposal

Current Developer Landing Page


Developer Resources - AllS
https://allseenalliance.org/developer-resources
LINUX FOUNDATION COLLABORATIVE PROJECTS
Register >
Log In >
Search
ALLSEEN ALLIANCE
ABOUT ALLSEEN
DEVELOPER RESOURCES
NEWS
SOURCE CODE
DEVELOPER RESOURCES
Home > Developer Resources

DEVELOPER RESOURCES

How to Participate

FAQs

AllJoyn

- Getting Started
- Framework Tutorial
- Docs/Downloads
  - Android
  - Arduino
  - Thin Client
  - OS and OS/X
  - Linux
  - Unity
  - Windows 7
  - Framework Tutorial

Ask

Submit a Bug or Feature Request

Mailing Lists

Project and Working Group Lifecycle

Wiki

DEVELOPER RESOURCES

The Alliance's open source codebase opens all sorts of opportunities – for indie developers, small and major studios, Fortune 1000 corporations, hobbyists and students! This open source project is specifically designed to be a powerful engine for enabling peer-to-peer experiences across connected devices, appliances and more. Discover the power and potential of proximal peer connectivity and communications and be at the forefront of driving the Internet of Everything.

By integrating code produced by the AllSeen Alliance, you can drastically reduce the time, effort, and cost of adding peer-to-peer features to your application. Whether you are developing for a smartphone, tablet, television, PC, or embedded consumer electronics, it is designed to provide the connectivity to enable groundbreaking new experiences.

The codebase, built on the AllJoyn open source framework, is:

- Easy to learn
- Portable, supporting a variety of operating systems and device types
- Open source
- High performance, low latency
- Optimized for mobile
- Feature rich

If you're looking for tools to add proximal peer-to-peer connectivity in your applications, you've come to the right place. From gaming, to entertainment, to education, to enterprise, to social, to multi-screen, the AllJoyn framework will help you deliver exciting new experiences.

Sections

#	Section	Description
1	Learn	This explains high level concepts, architecture, software components, protocol.
2	Download	This is where the developer goes to download and get the SDKs and source tarball. Explains which SDK to download
3	Develop	This has all sorts of information to develop. The meat of the content
4	Ask	Link to Ask Forums to ask questions and find answers
5	Contribute	Focused on how to contribute to the project

1. Learn

#	Section	Description
1.1	Intro	What is alljoyn, high level features, use cases
1.2	Architecture	High level architecture, define components, relationships (apps, thin, standard router, core, services)
1.3	Core Architecture	Core system overview and concepts. Includes subsections to cover flow, components, terms. Detailed system protocol definition
1.4	Service Framework Architecture	Describe each service, concepts, terms, describes how it works at the high level, interface definition
1.5	Glossary	

2. Download

#	Section	Description
2.0	Download	List of SDKs, tools to download, broken out by platforms. Info about each release. Links to older releases

3. Develop

#	Section	Description
3.1	Building	Info on how to build downloaded SDKs, source tarballs. Lists dependent tools.
3.2	Running Sample Apps	Describes how to run sample apps, command line arguments, which buttons to press, how to setup.
3.3	Tutorials	Tutorials to explain how to write a simple app, broken out into core/services. Walks through sample app.
3.4	API Usage Guides	For each service, provides high level description on how to use APIs and groups of APIs.
3.5	API Reference	Describes each API (auto-generated from code)
3.6	Debugging	Lists common issues, explains debugging techniques/tools.

4. Ask

#	Section	Description
4.0	Ask	Ask forum. StackOverflow like Q&A community supported site.

5. Contribute

#	Section	Description
5.1	Process	Describes contribution process, WG process, certification
5.2	Wiki	Wiki for contributors. Upcoming releases/plans. WIP WG info, APIs. Info on building from source. Other dynamic info for contributors.
5.3	Source	Link to git repos
5.4	Bugs & Feature Requests	Link to Jira change request system
5.5	Mailing Lists	List of mailing lists to participate in AllSeen development.

Tool and Process Proposals

Tool Concepts

1. Wiki
2. Drupal with workflow
3. Git + Drupal import

Tool #1: Wiki

- Use Wiki for everything
- Pros:
  - Available today, easy to use
- Cons:
  - Limited formatting, layout => not consistent with main site
  - Lack of approval process => decreased quality, organization over time

Tool #2: Drupal with workflow
(1 of 2)

- Add workflow module to existing Drupal
- Contributors adds edits
  - Changes goes to need-approval bucket
  - Select number of contributors (a few per WG)
- Approvals approves edits
  - Or provides feedback
  - Once approved, goes live
  - Select number of approvals (2-3 per WG)
- Philosophy: trust first, revoke later

Tool #2: Drupal with workflow (2 of 2)

- Pros
  - Approval system to ensure quality
  - Can easily preview proposed edits
- Cons
  - Needs further investigation for ease of use
  - Doc workflow different from source workflow

Tool #3: Git + Drupal Import

- Concept:
  - Keep content source in git
  - Tool imports to Drupal regularly

- Pro:
  - Approval workflow is familiar with contributors

- Con:
  - Need to find/build tool
  - Unsure if can preview edits
  - May have some/large development costs

ALLSEEN
ALLIANCE

Gateway Working Group Proposal Presentation

Art Lancaster

CTO Affinegy
April 29, 2014

Gateway Working Group Proposal Objectives

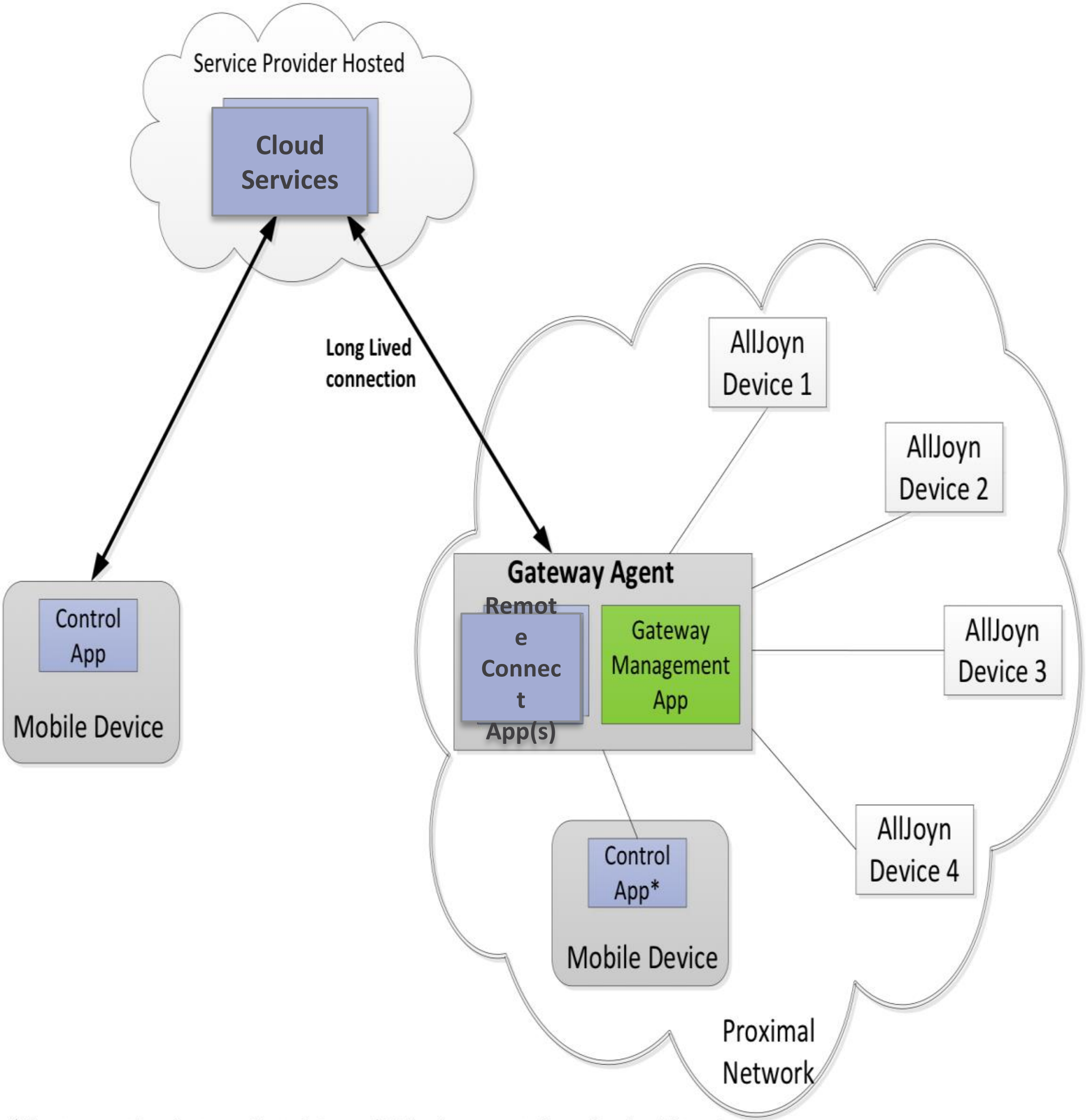
- Provide a standard and secure, remote access method for Alljoyn devices and applications
  - The remote access method should not require specific Internet firewall or port mapping configurations, so that it robustly supports most Internet connections
- Provide an extensible and standard means to connect AllJoyn devices to external/cloud services by providing:
  - A secure services discovery and provisioning node
  - Managed by the proximal network owner or with granted authority by a services provider
  - The external services to be provisioned through a hardened gateway node
  - Support multiple independent services providers – where the network owner has easy control of which are allowed to connect to the proximal network Alljoyn devices and applications.

Initial Contributors

- Affinegy
  - Art Lancaster, CTO – contributor and proposed as W.G. chair
  - Committers
    - Josh Spain, Director of Embedded Client Applications
    - Kevin Sandifer, Software Developer
    - Jim Howard, Sr. Software Developer
- Qualcomm
  - Shane Dewing, Senior Director Product Management – contributor
  - Committers
    - Tsahi Asher, Engineer Sr. Staff/Manager
    - Tali Messing, Engineer Sr.
    - Josh Hershberg, Engineer Sr. Staff

Overall Arch

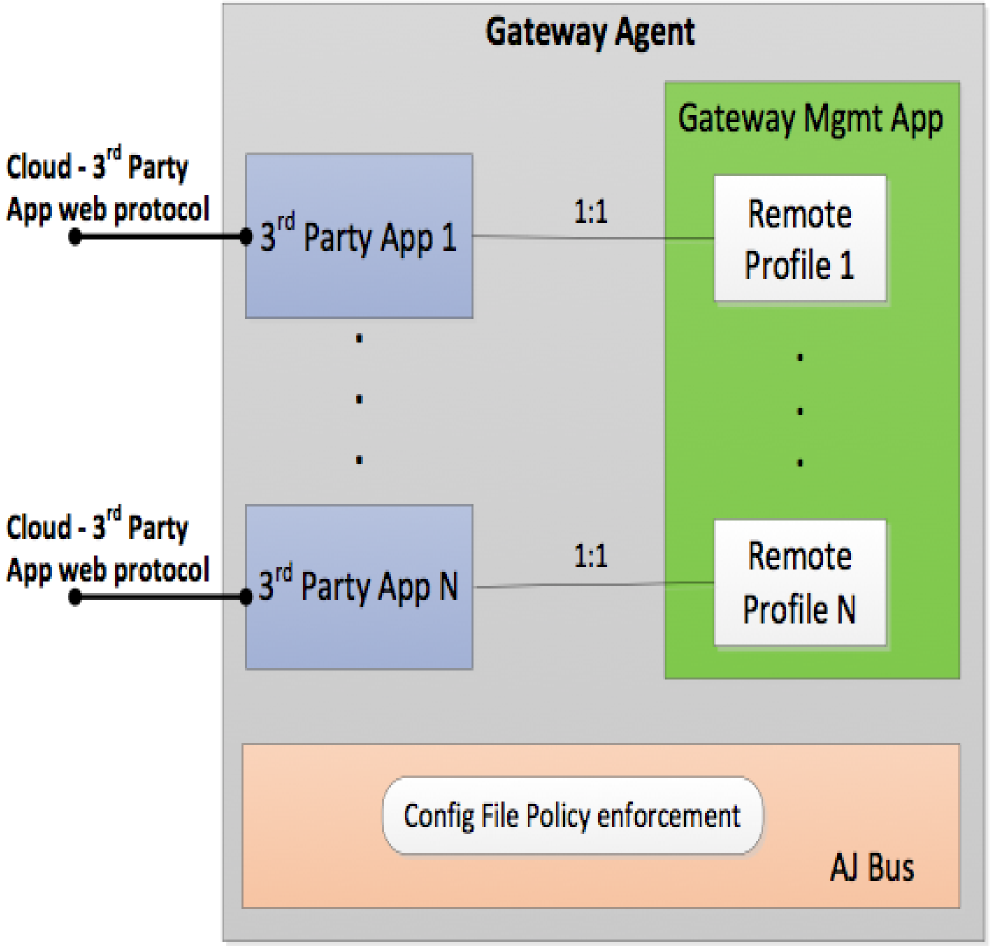
- Gateway Agent
  - Gateway Management App
  - Remote Connect App(s)
    - 3rd party apps connected to APIs of this W.G.
    - Reference implementation by this W.G. using existing standards for secure remote management and NAT traversal
- Control App
  - For the user to control the external applications



*Direct connections between Control App and AJ Devices are not shown for simplicity sake.

Remote Connection Profiles and Apps

- A Gateway Agent has a single Gateway Management app and one or more Remote Connection (third-party) apps.
- The Gateway Management app supports one or more remote profiles.
- Each remote profile gets associated with a single Remote Connection app.
- The architecture supports a 1:1 relationship between the remote profile and its Remote Connection app.
- The Gateway agent will include a secure means to manage the 3rd Party Apps installation, updates and deletion



Gateway Management Application

- AllJoyn profile management
  - Implements a secure AllJoyn Profile Management Interface org.alljoyn.GWAgent.ProfileMgmt. This is a generic interface independent of any service provider and exposes profile management functionality
- Secure interface for Remote Access (3rd party) Apps
  - Provides details for devices/apps/interfaces that have been configured for remote access to the third-party app and eventually to the cloud service
  - Since the Gateway Management app and third-party app are talking to the same AllJoyn bus, there is no need to establish an AllJoyn session between them
  - The Gateway Management app and third-party app register following WKNs with the AllJoyn bus and communicate with each other using these WKNs

Remote Access (3rd party) Applications

- Connection management with cloud service – persistent connection with protocol supported by the cloud service
- Data exchange with cloud service
  - Implements service provider-specific web protocol to exchange data with the cloud service for incoming and outgoing traffic to/from the proximal AllJoyn network.
  - This W.G. will provide a reference example implementation using standard implementations
    - XMPP / BOSH (XEP-0124) for efficient NAT traversal connections to AllJoyn devices/applications, extended with TR-069 for the addition of secure management control.
- Protocol translation between the external service’s protocol with the Alljoyn format
- AllJoyn device access
- Supports the standard AllJoyn API and Service Frameworks
- App access connection status to the Gateway Management App
- Exposing one or more services to Alljoyn devices in the proximal network.

Control Application

- An AllJoyn-enabled application developed by the service provider or an affiliated entity, or the gateway provider
- The specific functions vary by the application by typically include:
  - User authentication to the service or services supported
  - Alljoyn Gateway Agent discovery by app name (there can be more than one in a proximal network)
  - UI for profile functions and Profile configuration
- This W.G. will provide a Mobile SDK and Sample Control Application

Initial Contributions Upon W.G. Formation

- Qualcomm Connect Experiences
  - High level design document that details a proposed architecture and design for the Gateway Agent
- Affinegy
  - Embedded C-library SDK for its Gateway management client with secure remote cloud management and NAT traversal technology
  - Hosted cloud server developer access supporting this reference example implemenation

Project Plan

- Project creation, use cases and initial contributions and enhancements to core – on Gateway Working Group creation
- Aligned with an Alliance release for October 2014
  - Gateway Management App and new AllJoyn Interfaces
  - Mobile SDK and Sample Control App
  - SDK for Remote Access (3rd Party) App developers
  - Remote Access example reference implementation (XMPP/BOSH – TR-069) library

Thank you and Acknowledgements

- Special thanks to Shane Dewing and the QCE team for major contributions to the proposal and architecture design
- Thanks to TSC members

Data-driven API Proposal

Summary of contributions

R14.06

- Extend type system
- Marshaling improvements for C/C++
- Prototype data-driven C++ API

R14.10

- Data-driven API for C++, Android, and Thin library
- Language binding support for extended type system
- Documentation
  - best practices for data model definition
  - data-driven API reference guide
  - data-driven API user guide

Where does the project fit?

- Core WG
  - – Depends on core extensions to introspection
  - – Adds, extends, or replaces existing APIs
- Development tools WG
  - – Code generation is key component
  - – Project focus is enhancing developer experience
- New WG?
  - – Project scope is broad and ambitious
  - – Promotes a layered approach that maintains backwards compatibility

Question raised by membership

- Current type system uses XML
  - Based on original DBus introspection
  - Formalized with a more rigorous schema
  - Data driven API proposal extends this
- Input from membership (Ilya Dmitrichenko)
  - Why not JSON rather than XML
  - More compact and easier to parse
  - *Timing would be right to make this change*

Thank You.