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- Organization is outdated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

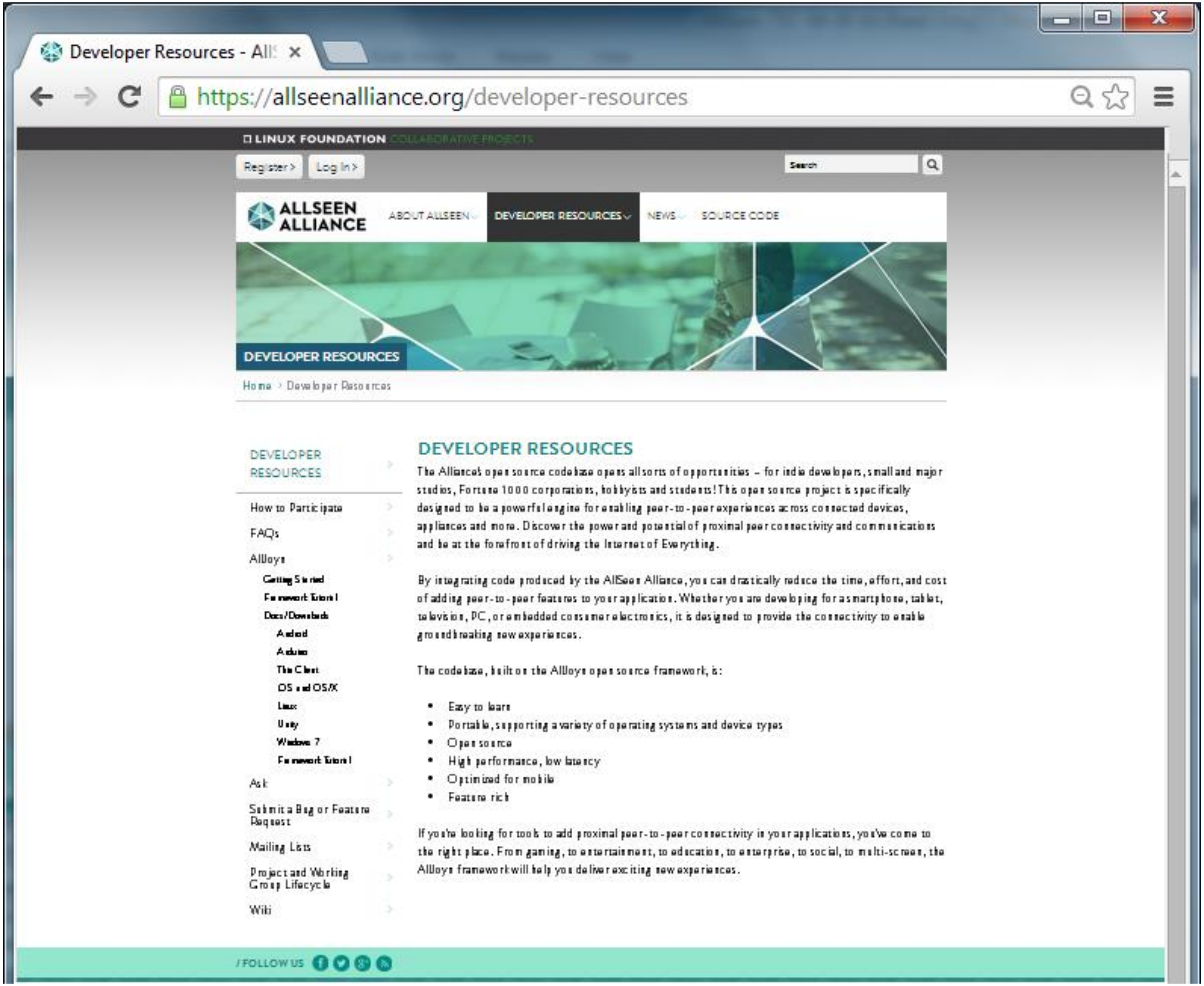
- Developers versus Contributors
  - those **using** versus **contributing** to the software
- Keep it simple
  - Start basic and incrementally introduce complexities
  - Goal: running in 5 minutes
- Web content, not PDFs
- Collaborative tool for easy, distributed, controlled updates

Two topics to discuss

- Content Organization
- Tool and Process

Content Organization Proposal

Current Developer Landing Page



DEVELOPER RESOURCES

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