A Java API for unifying ad-hoc Wifi networking

Peter Banis, Klaus Cipi, Michael Kolar, Robert Olsen

Faculty Sponsor: Dr. Marius Silaghi

Milestone 1 (October 1)

- Decide between developing on MacOs, Windows, or Linux
- Install and configure Android Studio
- Decide whether to use Eclipse or IntelliJ for an IDE
- Install and connect to the SVN repository provided by Dr. Silaghi
- Successfully install the API
- Run an example program with the API (to verify correctness of install and behavior)
- Create Requirements Document
- Create Design Document
- Create Test Plan

Milestone 1 Progress (1/2)

Task	Completion %	Peter	Klaus	Michael	Robert	To Do
Decide between developing on MacOS, Windows, or Linux	100%	Discussion	Discussion	Discussion	Discussion	None
Decide whether to use Eclipse or IntelliJ as an IDE	100%	Discussion	Discussion	Discussion	Discussion	None
Install and configure Android Studio	100%	Do	Do	Do	Do	None
Install and connect to the SVN repository provided by Dr. Silaghi	100%	Do	Do	Do	Do	None

Milestone 1 Progress (2/2)

Task	Completion %	Peter	Klaus	Michael	Robert	To Do
Successfully install the API	100%	Linux	MacOS	Windows	Android	None
Run an example program with the API	100%	Do	Do	Do	Do	None
Create Requirements Document	100%	20%	50%	20%	10%	None
Create Design Document	100%	10%	50%	20%	20%	None
Create Test Plan	100%	90%	0%	10%	0%	None

IntelliJ > Eclipse (Prove it!)

Feature	IntelliJ	Eclipse
Easier Debugging	To evaluate expression -> hover over expression with mouse then press hotkey	To evaluate expression -> highlight entire expression
Superior Auto-Complete	Unfinished word -> Use context clues (type, location, etc) to guess what is being typed accurately	Unfinished word -> show user every possible class/method/variable name!
Relevant Variable Names	Suggest to the user unused names that seem natural to a human programmer based on method name, type, and value	Null

Requirements Document (1/2)

Responsibilities of the API

- Support for Windows 7 & 10, MacOS 10.5+, Linux, and Android 4+
- Account for both P2P and ad-hoc connections
- Use P2P where supported by hardware
  - Use ad-hoc otherwise, required as a “bare minimum”
  - Android devices must support P2P to be compatible
- Communication over UDP and TCP/IP
- Take advantage of Android’s simultaneous internet and ad-hoc connections

Requirements Document (2/2)

Installation Verification Tool

- Inform the user of missing/corrupt files
- Report problems due to insufficient permissions (what and where)
- Subset of tools that attempt to correct both
- Warn user of device incompatibility
- Quick, detailed feedback for issues

Design Document (1/2)

- Stakeholders’ Perspectives
  - API Providers: People who design, build, and distribute API
    - Directly affect API
  - API Consumers (developers): Software developers who integrate the API into their applications
    - Directly interact with API
  - API Costumer (non-developers): People who pay for the API’s use (Usually Companies).
    - Indirectly benefit from API
  - API-End Users: People who use (and hopefully enjoy!) the applications that are based on the API
    - Indirectly benefit from API

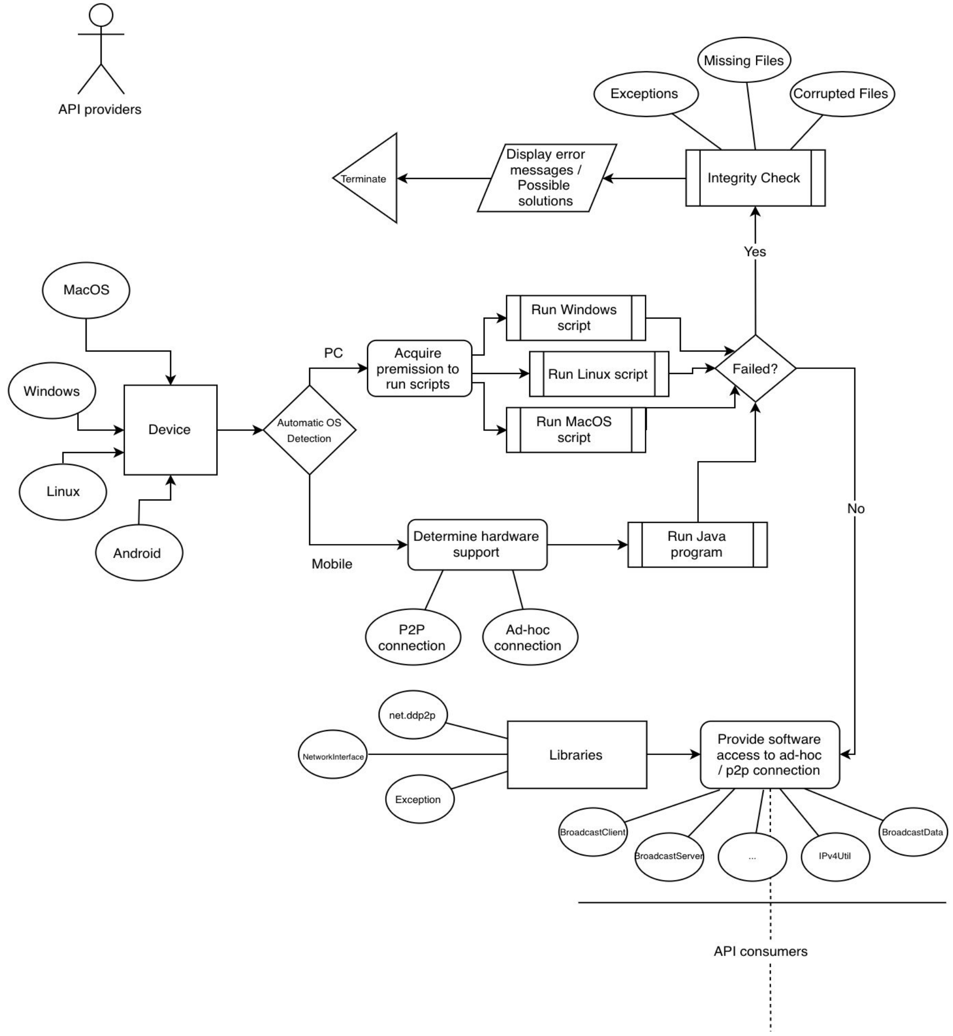
Design Document (2/2)

Design Concerns

- Responsiveness
  - Ability of a system to complete tasks in a given time
- Reliability
  - The ability to transfer data with full integrity
- Functionality
  - The set of functions associated with a piece of software

API Providers Viewpoint

(Sample Diagram)



Test Document (1/2)

- Code written for a milestone is tested under the same milestone
- Modifications to code will require retesting
- Testing done across all four platforms
- API tested for ad-hoc, P2P, and intercommunication
- Validation tested for file, permission, and compatibility checking
- Ad-hoc testing will be directed by Peter and Klaus
- P2P testing will be directed by Michael and Robert

Test Document (2/2)

- Documentation formatting
    - Date of testing
    - Name of tester
    - Names of code verifiers
    - Approval of test director
- Test performance
    - Given input
    - Expected vs actual output
    - Comments on unusual/inconsistent behavior

Milestone 2 (October 29)

- Integrity check and installation validation (IT&D)
- Automatic OS detection (IT&D)

Questions?