

Project Title

A Java API for unifying ad-hoc Wifi networking

Team Members

Klaus Cipi - kcipi2015@my.fit.edu

Peter Banis - pbanis2015@my.fit.edu

Michael Kolar - mkolar2015@my.fit.edu

Robert Olsen - olsenr2015@my.fit.edu

Faculty Sponsor

Dr. Marius Silaghi - msilaghi@fit.edu

Client

Dr. Marius Silaghi - Associate Professor at College of Engineering & Science, FIT

Meetings with Sponsor/Client

Friday, January 11

Goals and Motivation

Our goal is to complete the Android compatible API for Ad-hoc and P2P networking, and to create an integrity check file that will verify that the API is properly installed. The API will allow easy connectivity between all main operating systems (Android, Mac OS, Windows, Linux). Currently, no unified API for Ad-hoc and P2P networking exists, which creates a serious deficiency in what can be done for users that are based in multiple ecosystems. The unifying nature of the Java API will allow for easy cross-platform communication. This will enable easier transfer of files and other information between devices, which will be convenient for their users. Ad-hoc networks will also allow different devices to connect to each other. This can allow users to give each other internet access, to create a distributed network or even form an intranet without being limited by what physical devices users have

Key Features

- Integrity check/installation validation: The system must ensure that all components have been installed correctly. This is not limited to only ensuring the correct files exist in the correct directories, but also that all files have needed permissions(for example, a script might lack an execute permission which prevents further progress). The system will recognize when such a situation occurs, and if it cannot be fixed by the system alone, the system will provide useful feedback to the user so they may correct the error.
- Automatic OS detection: A single Java API package will include support for all major platforms like Android, Windows, Linux, and MacOS. The API will be able to automatically detect the platform that the developer is working on and provide the appropriate support for it. Furthermore, during the installation of any application that will use this API, only scripts for the proper platform will run.

- Better utilization of Android's native hardware capability: We plan on taking advantage of Android's native hardware capability for simultaneous dedicated internet connect and ad-hoc connection. Users will find this convenient since they won't need a second adapter to use both connections.

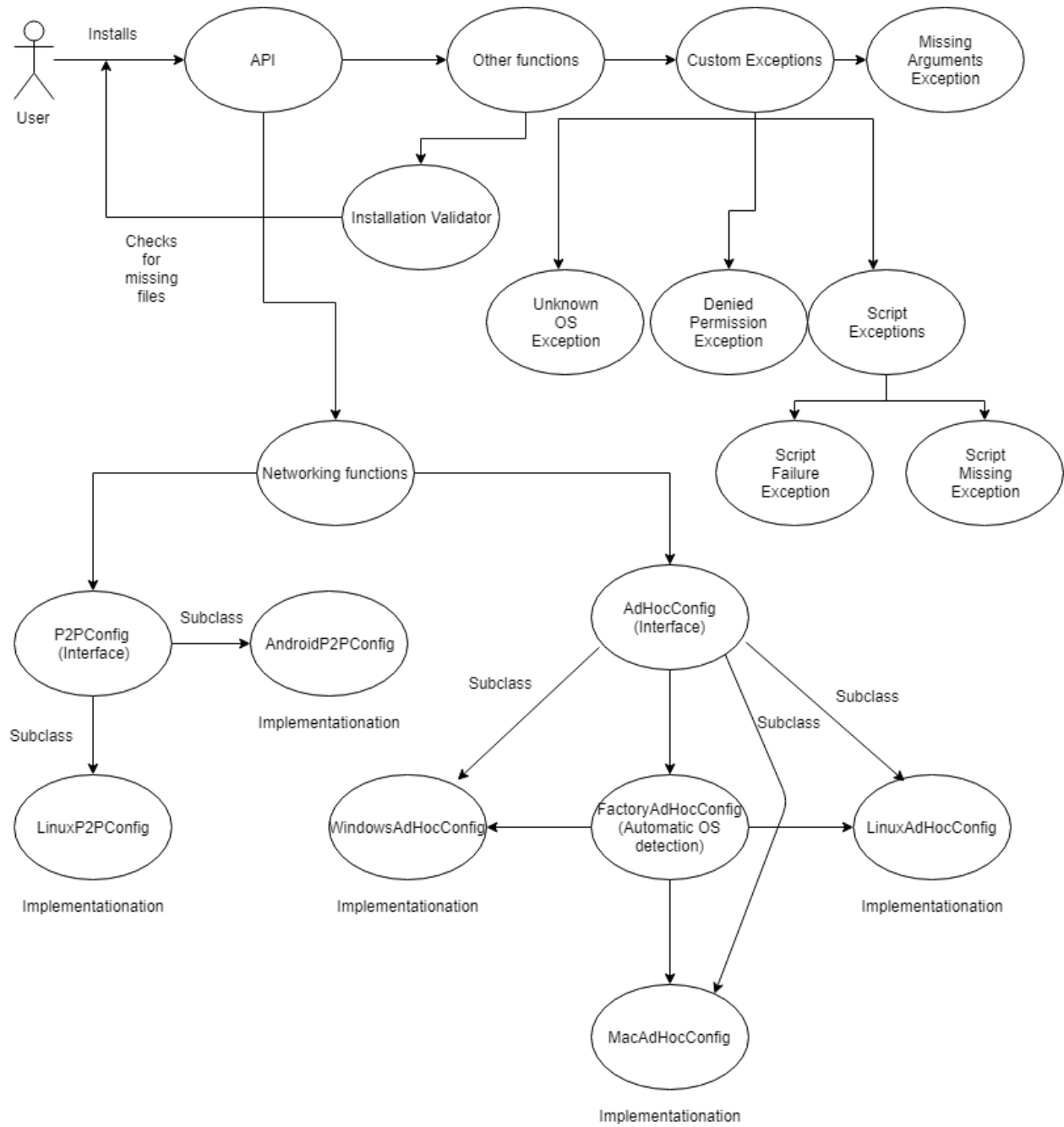
Novel Feature

Creating ability to use Android devices' dedicated ad-hoc wifi connection. We are not aware of software that currently takes advantage of this feature.

Technical Challenges

- Quickly identifying all peers on network to enable communication. Attempts to use broadcasting in the past have been met with some difficulty and will require additional care.
- Since we have yet to work with Android, our team is still unfamiliar with its development processes.
- Android does not support ad-hoc mode, so it cannot be used to host an ad-hoc network. This means that all networks started by Android devices will have to be P2P networks.

System Architecture Design Diagram



Progress Summary

Module/feature	Completion %	To Do
Automatic OS Detection	100%	None
API Logging	100%	None
OS Configuration Classes	100%	None
Ad-Hoc Support	90%	Method to detect other devices on a network; DHCP determined to be untenable as solution
Direct P2P Support	20%	Linux and Android only; Mac/Windows do not have accessible tooling for DirectP2P support

Milestone 4 - Due February 11, 2019

- Complete DirectP2P support
- Complete DHCP option for Ad-Hoc networks
- Implement Android P2P support

Milestone 5 - Due March 18, 2019

- Create Showcase Poster
- Ensure that devices on a network can be automatically detected (no more verbally asking for the other IP address)
- Ensure each OS (now including Android) can connect via its preferred connection type to each other OS.

Milestone 6 - Due April 15, 2019

- Create User Manual
- Create Demo Video

Task Matrix for Milestone 4

Task	Peter	Michael	Klaus	Robert
Complete DirectP2P support	30%	25%	20%	25%
Complete Detect Devices on Network	25%	25%	25%	25%
Implement Android P2P support	30%	17%	18%	35%

Description of each planned task for Milestone 4

- Complete DirectP2P support: The only platform that fully supports P2P is Linux, so during the next milestone we will only enable this support on Linux. We will work on writing scripts for each OS, if P2P is possible, and provide a way for developers to identify, join or create Wifi direct networks. On Windows and Mac, the API methods to identify availability of a DirectP2P network will inform the user that the connection type is not supported. After a connection is established, our API will provide sockets for the developers to exchange data.
- Complete Detect Devices on network: Currently we rely on a combination of static IPs and a feature of the 802.11 standard which specifies a device to default to an IP in the 169.254.X.Y range when attempting to get an IP via DHCP and there is no DHCP server available. Enabling DHCP support has undergone initial experimentation without success. The current operating assumption is static IP assignment from scripts in this range is workable and solves the issue of IP allotment. We need a method to determine all the peers on the network, as this is not reliably supported at the protocol level. Our current design approach is pinging all the IP's in the range and then returning a list of all the IPs that respond. This is the method we will use going forward.
- Implement Android P2P support: The Android API already has built in support for P2P networking through the WifiP2PManager class. This and related classes will enable us to progress quickly through the implementation of Android support.

Approval from Faculty Sponsor

- "I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones."
- Signature: _____ Date: _____