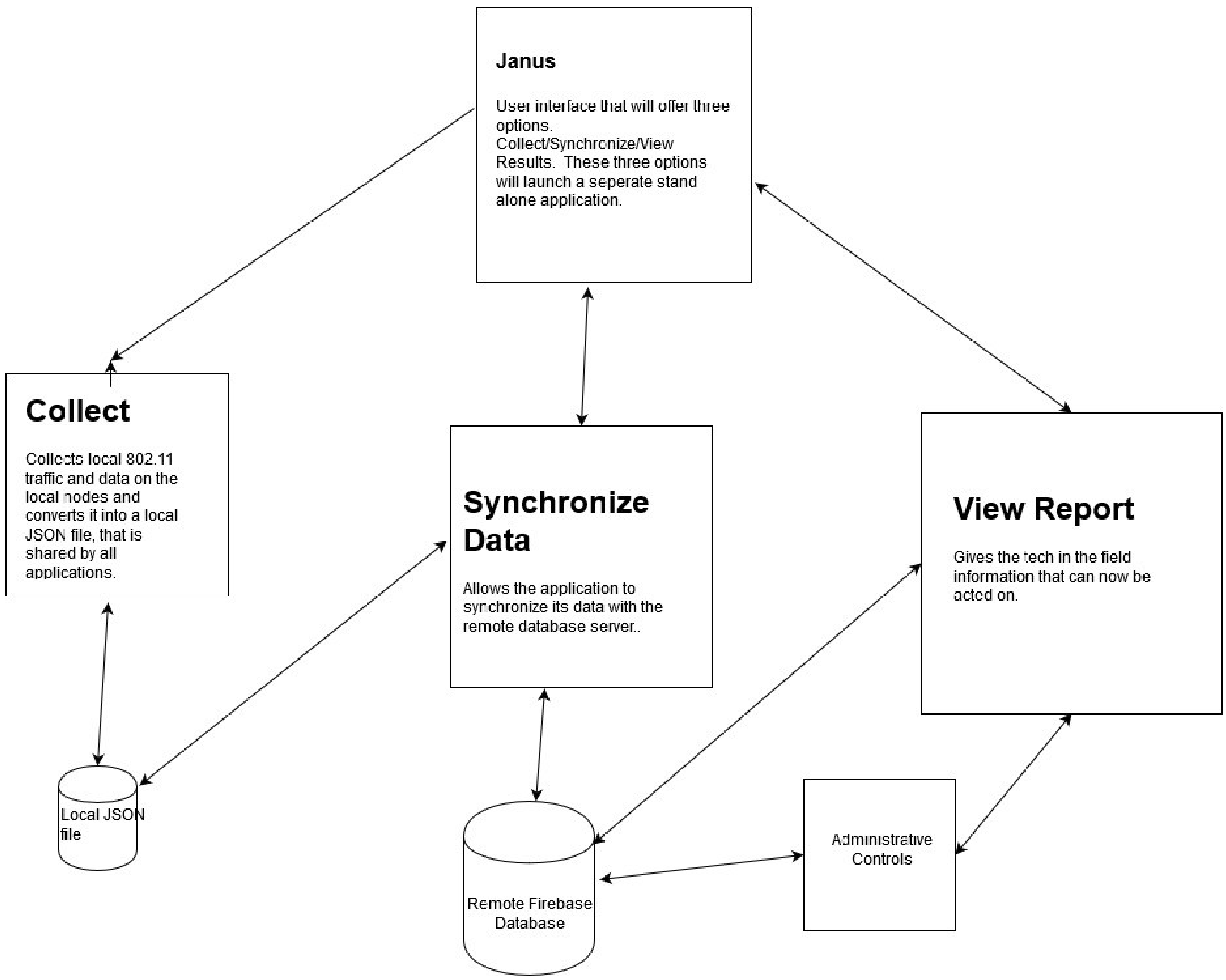
CS 477
Final Project
Architectural Report
Fall 2019

Christopher Wells
G-00260513

Overview:

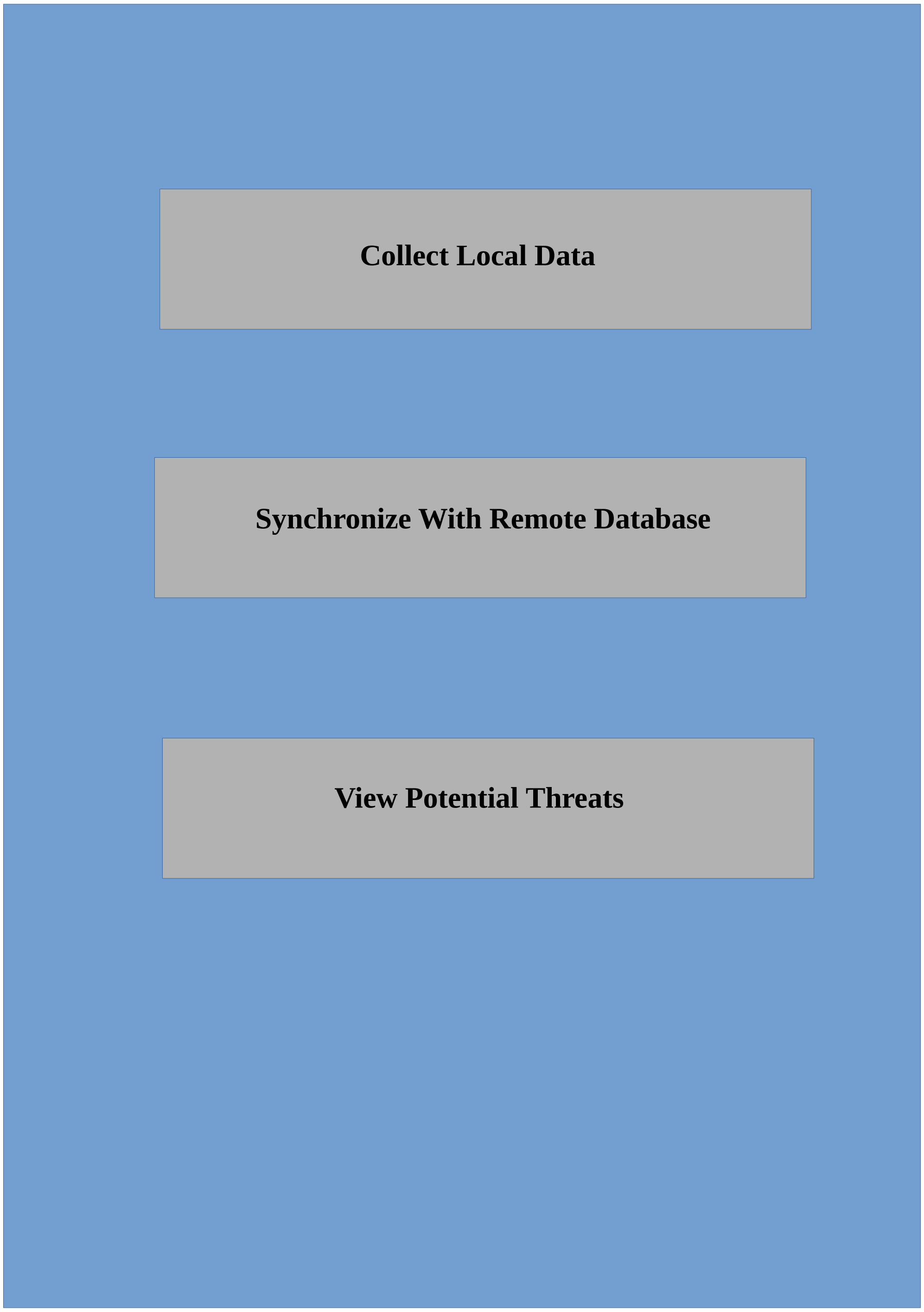
The design philosophy of this project is to divide and conquer. The project will be divided up into separate stand alone components. In this way the complexities of the larger project can be broken into manageable pieces. The main project will be broken into three general categories. The first being the collection of information. The second being the analysis of the information. The last component is going to be the processing of this information into actionable information.

Top Level Overview


Janus
User interface that will offer three options. Collect/Synchronize/View Results. These three options will launch a seperate stand alone application.
Collect
Collects local 802.11 traffic and data on the local nodes and converts it into a local JSON file, that is shared by all applications.
Synchronize Data
Allows the application to synchronize its data with the remote database server..
View Report
Gives the tech in the field information that can now be acted on.
Local JSON file
Remote Firebase Database
Administrative Controls

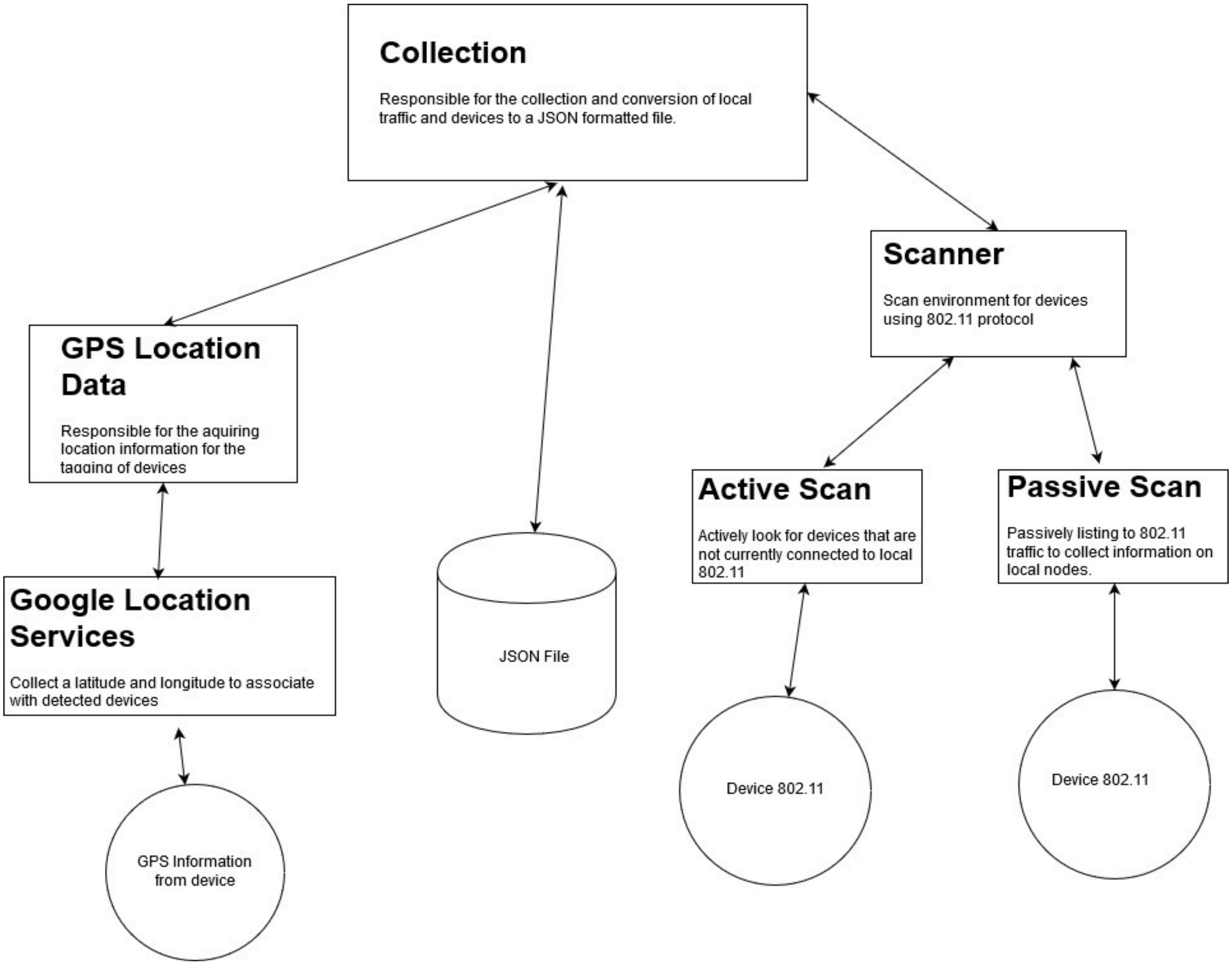
The modular design will allow for the easy replacement of a component, without the redesign of the entire project. The design will also allow for future growth. It is oped that this basic platform to be built upon and customized to fit environmental needs.

Janus User Interface



This application serves as a interface with the user to allow for the selection of the proper application to complete the task. There is not a large amount of code involved in the operation of this application.

Collection (802.11 protocol)


Collection
Responsible for the collection and conversion of local traffic and devices to a JSON formatted file.
Scanner
Scan environment for devices using 802.11 protocol
GPS Location Data
Responsible for the aquiring location information for the taaaina of devices
Google Location Services
Collect a latitude and longitude to associate with detected devices
GPS Information from device
JSON File
Active Scan
Actively look for devices that are not currently connected to local 802.11
Passive Scan
Passively listing to 802.11 traffic to collect information on local nodes.
Device 802.11
Device 802.11

Collection scans the environment for devices, and tags their traffic and ID. This data is collected and formatted in JSON format and stored in a common file for use by other components of the platform.
GPS Location Data will use the device GPS to pin a location that the data is being collected.
Active Scan, will attempt to ping addresses that seem inactive to see if there is a response. Look for sleeping devices. (Application would be to look for devices connected but inactive. The idea springs form a tablet that was once found in an office that no one knew about. It was running Android 2 and connected to the network, with no security.)
Passive Scan, puts the phone into passive mode and collects traffic in the area. This by far will be the primary method of collecting data.

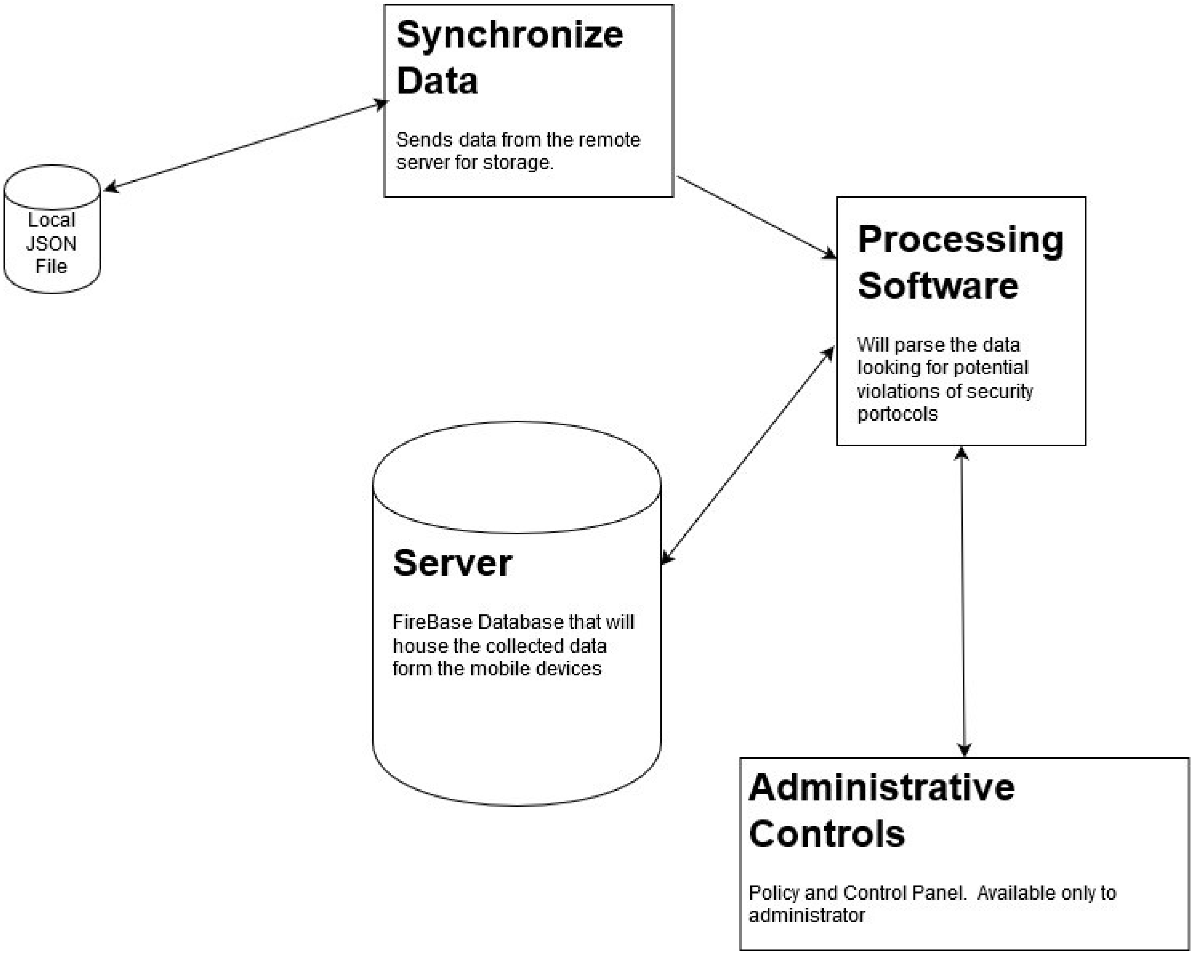
The stored data will be:

- Location:
  - Lattitude: Double
  - Logitude: Double
- Device:
  - Device ID: 64 bit, first 16bits will be zero for older devices
  - Destination IP: 32bits for the address that the device is talking to
  - Source IP: 32 bits
  - Date: Date
  - Time Start: Time
  - Time End:(Duration of the Communication to the site. Extremely short could be a sign of redirection to a Malware location)

Future Upgrades

The collection of cell traffic. This will allow the identification of devices connected to both the network and cellular data. This situation enables communication that are not filtered by the network firewall protocols. Another addition in the future will be a line of sight detection protocol. This will allow the user to look for signal scattering and know if the device is in a direct line of sight. The addition of deployable remote stations in in development for future additions. The architecture being based on the Arduino WiFI package.

Synchronize Data

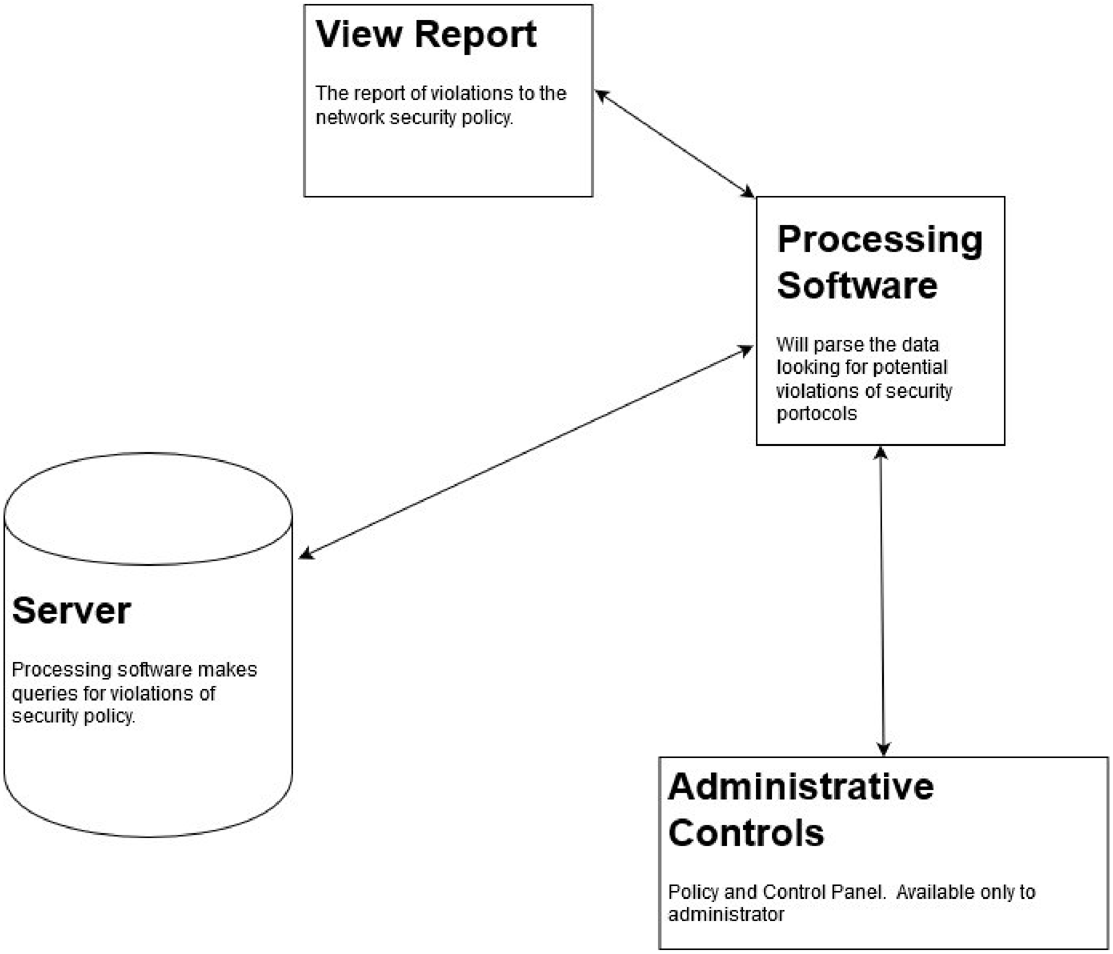

Synchronize Data
Sends data from the remote server for storage.
Local JSON File
Processing Software
Will parse the data looking for potential violations of security portocols
Server
FireBase Database that will house the collected data form the mobile devices
Administrative Controls
Policy and Control Panel. Available only to administrator

The goal is to be able to store the collected information and to process the data into something meaningful. It is this system that will need the strictest security policies, since it will have the repository of traffic data. The design principle will follow the “Least Privilege” model. The processing that will be implemented will add the owner and signature of the sites being visited. So for example:

www.gmu.edu 129.174.1.59 SSL Certificate Issued by InCommon RSA Server CA

The server can authenticate the IP as belonging to George Mason University. The certificates were updated in September. The site is not on a black list. This site would end up on a log just in case of an incident (Our version of the circular filing cabinet). The location would not be flagged.

View Report



View Report
The report of violations to the network security policy.
Processing Software
Will parse the data looking for potential violations of security portocols
Server
Processing software makes queries for violations of security policy.
Administrative Controls
Policy and Control Panel. Available only to administrator

After the data has been processed the report of violations can be requested by the user. The report will be of the form of date, location, MAC, IP Violation. Administrative controls will set the policies, and update information. The ideal situation is a live update on black list sites, and traffic monitoring. This first design is to be more of a proof of concept, and the data will be of a best effort form. In the future, modules will be replaced to improve the performance of the system.