DATA FORMAT

Data is sent from the client smartphones in a ASCII string which conforms to the JSON format. The overall organization of this method is an object containing the phone metadata and an array which contains various objects, each representing the data gathered from a specific scan initiated by the phone at a point in time. Thus, at a high level, we will have the following layout:

```
{
      “metadata”:METADATA OBJECT,
      “data”:[
            Scan information 1,
            Scan information 2,
            …
      ]
}
```

Our phone metadata comes from the Android Build API and includes various information about the phone such as brand, model, underlying hardware information, and the various pieces of operating system distance.

```
{
      //main board name
      “Board”:STRING,
      //system bootloader version
      “Bootloader”: STRING,
      //consumer visible brand of phone
      “Brand”: STRING,
      //name of the industrial design
      “Device”: STRING,
      //build id string that is consumer visible
      “Display”: STRING,
      //unique build identifying string
      “Fingerprint”: STRING,
      //name of hardware from /proc
      “Hardware”: STRING,
      //undocumented...
      “Host”: STRING,
      //either the changelist number or some other string
      “ID”: STRING,
      //name of hardware or product manufacturer
      “Manufacturer”: STRING,
      //user visible product name. May or may not be the familiar name
      “Model”: STRING,
      //name of product
```

```
    "Product": STRING,
    //hardware serial number
    "Serial": STRING,
    //list of tags identifying the build
    "Tags": STRING,
    //undocumented...
    "Time": NUMBER,
    //type of build, i.e. user or engineering
    "Type": STRING,
    //undocumented...
    "User": STRING,
    //version string of radio firmware
    "Radio": STRING,
    //development codename for build or "REL" for release builds
    "Codename": STRING,
    //string representing the user visible os release version
    "Release": STRING,
    //internal source code number, (e.g. git hash)
    "Incremental": STRING,
    //integer representing available SDK/API versions
    "SDKint": NUMBER
}
```

The information from the scans is encoded into its own unique JSON Object containing generic metadata about the scan (time and location it occurred), as well as the results of the scans themselves. Thus our "scan information" objects will have the following structure:

```
{
    //timestamp of when scan happened
    "timestamp":NUMBER,
    //GPS location of the scan
    "latitude":NUMBER,
    "longitude":NUMBER,
    //wifi scan results
    "wifi":ARRAY OF WIFI OBJECTS,
    //lte scan results
    "LTE":LTE OBJECT
}
```

Since there can be many different Wifi capable access points in the vicinity of any client smartphone, we will store the data collected from the wifi scans in an array. The data about each particular access point will be contained within its own JSON object with the following format:

```
{
      //this timestamp is how long the wifi AP has been up
      “timestamp”:NUMBER,
      “frequency”:NUMBER,
      “RSSI”:NUMBER,
      “BSSID”:STRING,
      “SSID”:STRING
}
```

We will only need a single JSON object to contain the data about the LTE information we gather from the phone, since the data is only collected about one specific entity, unlike the wifi. Thus, our object containing information about the LTE scan results is formatted like so:

```
{
      “NetworkType”:STRING,
      “ServiceProviderName”:STRING,
      //Signal Strength(RSRP) as a dBm value
      “dBm”:NUMBER,
      //LTE cell identity
      “CellID”:NUMBER,
      //LTE Mobile Country Code
      “MCC”:NUMBER,
      //LTE MObile Network Code
      “MNC”:NUMBER,
      //LTE Physical Cell Id
      “PCI”:NUMBER,
      //LTE Tracking Area Code
      “TAC”:NUMBER,
      //LTE Timing Advance value
      “TimingAdvance”:NUMBER,
      //LTE Reference Signal, Signal to Noise
      “RSSNR”:NUMBER,
      //LTE Channel Quality Indicator
      “CQI”:NUMBER,
      //LTE Reference Signal Received Quality
      “RSRQ”:NUMBER,
}
```

The following is an example of what this could look like in practice, slightly abbreviated for brevity:

```
{
      “metadata” : {
```

```
      “Board”:”APQ8064”,
      “Bootloader”:”1.5.7.0000”,
      “Brand”:”htc”,
      “Device”:”m7”,
      “Display”:”KOT49H release-keys”,
      “Fingerprint”:”htc/cingular_us/m7:4.4.2/KOT49H/336091.3:us
      er/release-keys”,
      “Hardware”:”m7”,
      “Host”:”ABM105”,
      “ID”:”KOT49H”,
      “Manufacturer”:”HTC”,
      “Model”:”HTC One”,
      “Product”:”cingular_us”,
      “Serial”:”HT34NW906112”,
      “Tags”:”release-keys”,
      “Time”:1419928358000,
      “Type”:”user”,
      “User”:”buildteam”,
      “Radio”:”4M.27.3218.14_10.37.1718.01L”,
      “Codename”:”REL”,
      “Incremental”:”336091.3”,
      “Release”:”4.4.2”,
      “SDKint”:19
},
“Data”:[
      {
            “timestamp”:#######,
            “latitude”:#######,
            “longitude”:######,
            “Wifi”: [
                  {
                        “timestamp”:########,
                        “frequency”:######,
                        “RSSI”:#####,
                        “BSSID”:”aaaaaa”,
                        “SSID”:”aaaaaa”
                  },
                  {
                        “timestamp”:#######,
                        “frequency”:######,
                        “RSSI”:#####,
                        “BSSID”:”bbbb”,
                        “SSID”:”bbbb”
```

```
                        },  …..
                    ],
                    “LTE”: {
                                “NetworkType”:”aaaa”,
                                “ServiceProviderName”:”bbbb”,
                                “dBm”:######,
                                “CellID”:######,
                                “MCC”:######,
                                “MNC”:######,
                                “PCI”:######,
                                “TAC”:######,
                                “TimingAdvance”:######,
                                “RSSNR”:#####,
                                “CQI”:#####,
                                “RSRP”:#####
                    }
            }, ….
      ]
}
```

SENDING THE DATA

Data will be sent from the client smartphones using basic TCP socket communication. These connections will only ever be initiated by the clients themselves, so the clients will not require any open listening ports.