We need to identify potential protocols to use for sending the raw data received by the phone service to the server. Initial ideas just based on what I've seen before are HTTP and MQTT. We need to figure out what the options are, and what the pros and cons are for each. In particular, we should at least answer the following:

* What available protocols are there?
* What are the pros and cons of each?
* What is typical for IoT type data?
* HTTP and HTTPS and HTTP/2
  + Request-response protocol
  + Depending on version can be secured or unsecured
  + Is stateless, doesn’t need info about senders/receivers over multiple requests.
  + Probably simplest to use and develop with.
  + [OkHttp](https://square.github.io/okhttp/)
    - HTTP & HTTP/2 client for android
* [CoAP](http://coap.technology/) (Constrained App Protocol)
  + Defined by [RFC7252](https://tools.ietf.org/html/rfc7252)
  + Intended for resource-constrained devices.
  + Easily interfaces to HTTP, very similar.
  + Provides request/response model between endpoints
  + Is secured
  + [Californium](http://www.eclipse.org/californium/)
    - Java implementation of CoAP
    - Targeted for back-end communicating with IoT
    - Provides API for RESTful Web services
    - “Outperforms HTTP servers”
  + [SPITFIREFOX](https://github.com/okleine/spitfirefox).
    - Android implementation of nCoAP
    - nCoap is based on asynchronous and event-driven network application framework [Netty](http://netty.io).
* [MQTT](http://mqtt.org/) (MQ Telemetry Transport)
  + M2M/IoT connectivity protocol.
  + Light weight publish/subscribe message transport.
    - ‘sensor’ publishes to ‘broker’ which sends data to ‘subscribers’
  + Designed for high volume/low message size.
  + Can set QoS for messages.
  + Two-way comm channel
  + For a sustained connection MQQT will be more battery efficient
    - <http://stephendnicholas.com/posts/power-profiling-mqtt-vs-https>
  + [Moquette](https://github.com/andsel/moquette)
    - MQTT broker for android
    - Uses Netty for encoding/decoding.
    - Can set QoS
* [Websocket](https://websocket.org/about.html)
  + Provides full duplex over single connection.
  + Compatible with HTTP, uses ports 80/443
  + Can be secured

<https://www.postscapes.com/internet-of-things-protocols/>

* Useful site with a lot of defintions/links for IoT related stuff. Lists things in 8 layers. Most pertinent being 3. Comms/Transport and 5. Data Protocols