Time: 2 hours

Ryan is learning RUST to work with spawning node js server and communication and such with that

Procrastination worked in our favor for finishing the project plan v2 early

Ryan made the message format for the particles and connection management class, we looked over it and discussed additional details about blocking and nonblocking. All messages should use this format to help with extensibility. The atomic layer also needs a signature and end point so particles can talk directly to the atomic layer. Streams need to be unidirectional for security b/c particle A and B can then communicate without going through atomic layer.

Nat and Dan made an unnamed document (naughty naughty) about Particle to Atomic communication

Alex had internal dialogue externalized

Connection manager needs to be created, basically a struc. Nathan will attempt, once again working with Dan to get this working.

Progress Reports

Alex: Issues with setup, need to 'npm install foundation-sites' for pretty UI. Using redux on node is confusing, working on getting that running. Ryan will send me code to help getting testing running. So Alex is not being silly in not finding a good solution, Ryan had many struggles too.

Nat: particle API is mostly done, finishing up iterative event. Worked with Dan to get this done.

Ryan: See above, also made more components

Dan: See Nat, Encryption work

Let us think about demo over break and this week.

Ryan and Dan had a long conversation before the meeting about service discovery and message types. Their solution helps to fix the multiple message issues.

Broadcast event go to all of a type of particle. Network layer then sends the X events to all particles. Ask Ryan and Dan for more info as needed.

Need to synchronize clocks on handshake bc Pi do not have an inherit clock

Ryan has to ninja away for 5 minutes at the end of the meeting.

========================

After class

Add logging feature to track message paths. Track message ID when it is passed

**What to do for demo...**

1 File manager

2 File system resources

1 Application to request file

Make the file an image, load into UI

Request comes in for file to manager, file manager asks file system for file, manager sends a stream back to application

Dan and Nat will make the file particle + file manager

Ryan and Alex will make the UI Particle