# Analysis of Alternatives

**Spikes Analysed:**

1. Johnny-Five and Socket.io using Node.js
2. BoneScript and Google Firebase using Node.js

**Background Information:**

Bonescript: A library of Node.js specifically for Beaglebone. It provides functions for hardware interaction. (beaglebone.org, 2016)

Johnny-Five: A javascript library for the Robotics and IoT platform. (Bailey, 2013) It is mainly developed for Arduino but has io packages that make it compatible with other microcontrollers.

Socket.io: A javascript library that is an event based bi directional communication layer for web applications. (Kelleher, 2014)

Google Firebase: Firebase is a realtime, cloud hosted database. The data is stored as a JSON and is synchronized to every connected client. (Google Firebase, 2017)

**Analysis Criteria:**

1. Latency of connection between client and serve
2. Learning curve for the toolchain
3. Support and documentation
4. Interoperability between toolchains

**Outcomes:**

Bonescript:

* The learning curve for this toolchain was relatively low as it uses simple concepts to deal with the hardware. The functions are also named in such a way that they are intuitive and can be easily understood, for example, digitalWrite()
* The documentation for bonescript is very complete and has many examples which makes it easy to use.
* Bonescript being built specifically for beaglebone, has no issues with the beaglebone. It also has no known compatability issues with Firebase or Socket.io.

Johnny-Five:

* The learning curve for this toolchain was relatively steep as it required the understanding of event based programming but seeing as our team has this understanding, it was not too difficult.
* The documentation for Johnny-Five and the examples given were mostly written for Arduino making it slightly more tedious as one must translate that for the beaglebone. Certain functions were also deprecated but still appear in the examples which could cause confusion.
* Using the beaglebone.io made johnny-five compatible with the beaglebone and there are no known compatibility issues with Socket.io or Firebase.

Socket.io:

* The latency using Socket.io was lesser compared to when using Firebase. Although it was a negligible amount.
* The learning curve for this toolchain was steep as our team did not understand the purpose of socket.io. However, after understanding its purpose, implementation was relatively easy.
* Support for socket.io is average as the documentation is complete but there are few examples that can be used as reference.
* There are no known compatibility issues with Johnny-Five or Bonescript.

Google Firebase:

* The latency using Firebase was slightly more compared to socket.io presumably because the data had to be updated to the database and only then could the client read. However, the difference is of a negligible amount.
* The learning curve for this toolchain was not too steep as the functions were very intuitively named. The structure for the database however is a JSON structure which would require some research for inexperienced users.
* The documentation for Firebase is the best out of all the toolchains explored as not only do they offer sample code and documentation but they also offer short videos explaining the samples of code.
* There are no known compatibility issues with Johnny-Five and Bonescript.

**Recommendation:**

Our team would recommend using a combination of Johnny-Five for hardware communication as the high-level nature of Johnny-Five would make it more usable for larger projects such as the morse code decoder. Since our team found it relatively simple to translate the examples for Arduino to Beaglebone, we found that the current support is sufficient. As there are no known compatibility issues, it would work well with either Firebase or Socket.io.

We selected Google Firebase for our client-server communication because should the project require scalability, Firebase would be able to provide those services without the developers having to program everything from scratch. The documentation for Firebase being so complete, would be useful when developing the project as it would aid our team in learning new functions to implement our project.

Nodejs is the preferred language as both our team members have worked with javascript in our past few projects, this makes the language familiar and still recent in our minds which would further reduce the learning curve when developing the next projct.

# **Bibliography**

Bailey, D. (2013, July 16). *JavaScript Powered Arduino with Johnny-Five*. Retrieved from Safari: https://www.safaribooksonline.com/blog/2013/07/16/javascript-powered-arduino-with-johnny-five/

beaglebone.org. (2016, February 10). *Bonescript*. Retrieved from beaglebone.org: http://beagleboard.org/Support/BoneScript

Google Firebase. (2017, April 13). *Firebase Realtime Database*. Retrieved from Firebase: https://firebase.google.com/docs/database/

Kelleher, F. (2014, August 11). *Understanding Socket.IO*. Retrieved from NodeSource: https://nodesource.com/blog/understanding-socketio/