Analysis of Alternatives

Criteria	**Implementation of Socket.IO in Python**	**Bonescript IoT library (Javascript) with the use of Socket.IO**	**Firebase real-time database**
Duration/Speed of signal transmission (fast/slow)	Performance similar to Bonescript IoT with Socket.IO	Fast	Firebase is fast but not as fast and straightforward in terms of its structure compared to Socket.IO since Socket.IO allows direct communication between Client and Server but if using Firebase we need to have an extra intermediate of database to handle the communication.
Difficulty to learn and Setup Complexity	Hard to learn compared to implementing Socket.IO with Bonescript. Requires the use of external library (python-socketio) for using Socket.IO in Python as Socket.IO is natively written in Javascript. Steeper learning curve as more tutorials and documenations available online are for setting up server and client applications in nodeJS environment (running javascript) as compared to Python. Writing server side script with Python is unconventional and relies heavily on the use of third party libraries. Also, not beginner friendly due to the native functions of beaglebone are more easily accessible with Bonescript.	Well-documented and many demo projects available on beagleboard.org makes learning easier for beginner. Easy to set up as the code can be run directly on the board using the node interpreter.	Easy to learn due to many online tutorials available. Testing by sending and receiving made easier as firebase is a real-time database. While learning is easy, setting up has been challenging. Firebase python module has to be downloaded before any Firebase code could work on Python.
Ease of maintenance	Require constant monitoring and more effort to maintain the codebase as it depends heavily on third party libraries which may introduce bugs from time to time.	High maintainability as Bonescript is specially designed and optimised for Beaglebone Board. Hence the bugs will be considerably less compared to other	Firebase database allows immediate changes to the database and hence modifications to the structure/data can be done more easily.

		third party libraries which may not be meant for using it with Beaglebone.	

Additional note about implementation of Socket.IO in Javascript and Python:

Javascript is favoured over Python mainly due to the abundance of resources online dedicated for Javascript in setting up Server and maintaining the device. It makes the learning curve for new members in the team much faster and increases the maintainability of the codebase in the long run. Also by choosing to implement in Javascript we would be able to eliminate the use of many third party libraries which would otherwise be needed in the implementation of Python.

Our team experienced many strange bugs and errors which only occur in Socket.IO of python implementation but similar code implementation was able to work fine in Javascript. This caused the team huge resources in troubleshooting these errors due to the use of third party libraries in order to support Socket.IO which is natively written in Javascript.

Hence , in our spike codebase, the implementation of Python into Beaglebone was not completed as we experimented with setting it up on a Laptop itself has shown that the implementation in Python is not worth the effort and such our team decided to focus on the implementation of Socket.IO in Javascript via the use of Bonescript. The screenshots for the errors and bugs are included in the spike codebase folder on GitHub.

Conclusion:

In addition to the above, we favour socket.io in Javascript over Firebase as Firebase would introduce an unnecessary layer of intermediate to the codebase and hence may introduce an unwanted delay in responsiveness.