**JAMScript Runtime API Design**

**Overview**

The runtime API is the interface exposed by the middleware to the JAMScript compiler. The JAMScript compiler translates the user program to either C or JavaScript that is embedded with calls to the API. The runtime API has two parts: JavaScript and C. The two parts are not symmetric because some functions are exclusively managed by one side. For instance, the data management component has many more API calls in the JavaScript side and few API calls in the C side.

**API for the C Runtime**

*Initialization of the Node*

Start the node

Wait for the node to bootstrap the initialization phase

Task table setup

*Remote Task Execution*

Call for task execution

Get results of the task execution

Wait for task execution without the results

*Sleeping or Waiting the Task Execution*

Go to sleep for some microseconds

Wait on a given semaphore

Signal a given semaphore

*Data Management*

Log data to a stream

Read data from a stream

**Example Code in the C Side**

To be filled in.

**API for the J Runtime**

*Initialization of the Node*

Start the node

Add the worker or start it

Add the scheduler or start it

Wait for the node to bootstrap the initialization phase

Add the task to the table

Add the condition to the table

Add the data definitions

*Remote Task Execution*

Call for task execution

Get results of the task execution

Wait for task execution without the results

*Sleeping or Waiting the Task Execution*

Go to sleep for some milliseconds

Wait on a given semaphore

Signal a given semaphore

*Data Management*

Log data to a stream

Read data from a stream

**Example Code in the J Side**

To be filled in.