* **Only one Rustboard can be connected to the robot at a time.** If two dashboards could be connected, which one should data be sent to? What if both dashboards have nodes with the same ID but different types? Data transfer becomes a nightmare.
* **When a client connects for the first time ever, the server and client must agree on a client id.** That way, if that client connects again after the robot has modified node values, it will recognize the client and be able to update the dashboard to the correct values.
* When a client reconnects to the server and the server wants to update some node values that have been locally modified**, the server must first check to make sure that the node in question is of the same type**.
* **When a connected client’s dashboard is modified, it must update the server version of the dashboard** with these modifications.
* Rustboard programs support multiple dashboard layouts. **A server that is connected to and communicating with a Rustboard must be able to update any layout, whether or not it is the “focused” layout that the user can currently see.**
* When a dashboard client connects, it sends a message:
  + “messageType”: “connection info”
  + “time”: // the UTC time of the client’s device
* **Two layouts that are part of the same Rustboard can have nodes with identical types and ids.**
* **Nodes can be accessed either by just their id or by their id and layout name. If a server attempts to modify two nodes with the same type and id at the same time, both of them will be updated.**
* **All Rustboards will be stored on the control hub. If they have not been used for a certain period of time, they will be deleted.**
* **When the Rustboard server is instantiated, it will not load each rustboard dataset into a Rustboard instance, because this would take far too much memory. Instead, metadata about the rustboards will be loaded into memory (the name of the rustboard, the )**
* **TODO:** When someone using the server-side code tries to grab values from an input, for example, the code might run into problems. What if multiple layouts have the same input with different values? The client-side code needs to link these inputs so that any change to an input in one layout results in a change to the identical input. In fact, **any nodes of the same id and type on the client side will be considered as the same node.**
* **The concept of a rustboard layout will be foreign to the Rustboard. A “layout” is simply an extension of, as well as the smallest visible unit of, a Rustboard.**
* **TODO: the client application should not allow nodes with the same id unless those nodes are of the same type**