*Reviewer: 1   
Comments to the Author   
In general - while the aim of your system is admirable I cannot really see how you are helping the implementer put data into your system. When you say you are compatible across multiple systems - do you simply mean that the user can write a client that just reads the data off the system and inserts the data into your database with some SQL? I get the point but it's not really an innovation. If you have a REST, SOAP or socket library that you use then detail it or if you have future plans for an interface then I'd explain that. I assume your file parsing library just reads in a static format? On the data presentation - concentrate on the plotting library tie in with the back end rather than saying the user can get data out with SQL and SELECT clauses - if the users knows SQL they'll say 'it's just SQL' and if they don't they won't understand it. I'd really try to give more concrete 'meat' to the system and if you intend to do things put in a 'futures' section to describe where you are going.*

TODO

*Specifics   
  
\* I assume you import file format is fixed - maybe give an example of this?*

TODO

*\* Do you have a framework to allow the user to input data into your system?*

TODO

*\* I assume you are using 'standard' Python and not Python 3K*

TODO

*\* Reference NumPy and SciPy in your biblio*

TODO

*\* With the continuous measurement and a local queue to hold data (both good ideas) do you have a framework for this and/or do you plan one in the future?*

TODO

*\* You have a big body of Python code - maybe break it into pseudo code with comments to explain what you are doing*

TODO

*\* When you say non-centralized systems - give an example even if it is just a case of Excel files floating around on desktops!*

TODO

*\* You explain your iof MySQL but you could have equally used Postgres for that so why didn't you? If it is just a flip of a coin - that's cool!*

TODO

*\* You says that a MySQL instance needs aa high performance server - unless you are uploading massive quantitites of data then you don't. Teh same of storage capacity - unless you have massive BLOBs you won't need that much storage. If you focus on storage talk about RAIDs - in layman's language obvioulsy!*

TODO

*\* When you say standard server software do you just mean a MySQL server?*

TODO

*\* Giving access to all the tables in a database is a \_really\_ bad idea - what about security; one client can affect the data for every other client.*

TODO

*\* ODBC is a bit dated; why not use a direct access and encrypt the username/password on the system. Also having a different username/password will require the DBA to give new passwords to every user.*

TODO

*\* If every client needs to make a table that means that each client needs to make a table. If I have 300 clients they each need their own table? Why not have a client table to identify the data?*

TODO

*\* You are using a weirdly formatted database; the metdata should be normalised out - why didn't you do this. There are valid reasons; look for NoSQL stuff but at the moment it just looks like a weird design.*

TODO

*\* On the data extarction - you're just describing SQL cut this section right down*

TODO

*\* You list possible DBMSs; include MS SQL - even if you don't liek it it is a very popular system!*

TODO

*\* The visualisation system is very interesting - write mreo about this and show examples of swapping out the plotters. Maybe talk about having setups which don't require files to be written on the server directly?*

TODO

*\* Are you just using the plotting library and raw HTML/CSS on the browser client side - if you are using DOJO or jQuery add it in and reference it.*

TODO