Issues with localization microscopy data handling

1. File Formats
* This actually breaks down into two separate issues. The diversity of file formats within which the localization data is saved and the diversity of ways that localization data is stored within those files.
* The types of localization data and the order in which they are saved can differ fairly substantially. Some localization software packages stop at saving the localized spot position and uncertainty of fit, some include more data about the signal and noise levels etc. There is no agreed upon set of fields or order in which those fields will be stored.
* There also isn't a single file format for Localization microscopy data at the moment. Most of them are different forms of tab-or comma-deliniated text files. Because of the number of localizations that must be saved from a single data set, these files end up being hundreds of MB to a few GB in size.
* Alternatives to text files include the binary '.tsf' file format which came from the Micro-manager localization plugin and is an option in ThunderSTORM for saving data. There is also an HDF5 based file format used by the The PYthon Microscopy Environment. QuickPALM also provides an option to save all the data in a '.tif' image. Of these options, HDF5 does have the advantage of wide adoption and ease of import into OMERO, plus existing tools for reading and writing to it. I am wary of trying to create a new standard though.
1. Handling localization data in OMERO
	* The raw data produced by a localization experiment is usually just a stack of image files. These are easily stored in OMERO. At the moment, the only way we have to store the processed data is to annotate an OMERO object with the localization file. I'm not sure how well that works, and it means the data has to be downloaded and then opened with a third party tool.
	* There also does not seem to be any way to connect the initial image stack to and images reconstructed from the associated localization data. It would be really useful to be able to click on an image stack and be able to see the processed 2D/3D images created from it and also stored in OMERO.
	* This allows for a scenario where an image stack is uploaded to OMERO after acquisition, a localization program then processes the data in the background, annotates the raw data with the results in some agreed-upon file format and then generates reconstructed images that are linked back to the original stack of images.
	* The other common scenario I'd expect is that the images are acquired and processed on one computer, then the stack, localization data and reconstructed images are all uploaded to OMERO.
2. Coming up with an independent data viewer
	* Once localization data has been obtained, it would be very useful to have an open source data viewer that could reconstruct both 2D and 3D localization data regardless of data file and output image files.
	* This means we would need some kind of input plugin system to allow it to read in all existing file formats and any new ones people come up with.
	* Ideally, there would also be processing plugins(maybe a scripting interface?) to allow people to include new means of correcting drift, filtering data, estimating resolution etc.
	* The ability to talk to OMERO could also be useful
	* The closest tool we have to fulfilling these requirements is palm-siever, which is a really useful tool, but doesn't appear to have been updated recently. It could be a great platform to start from though. There is another potential alternative called VISP, but as far as I am aware it isn't open source and doesn't, as far as I am aware, have a plugin interface.