Requirements for defining a PREMIS standard for Fedora

Open Questions

- How will the PREMIS be used?
  - support standard external policy audit tool
  - harvest and index for curatorial queries
  - generate a preservation view of objects
- Do we want to define a standard dissemination of PREMIS or a standard storage for PREMIS? (contract vs. implementation)
  - If both, pick one to start with
- Which parts of the PREMIS schema must be stored as such and which parts may be generated on the fly, from Fedora and other system data?
  - Will we be making suggestions for tools which can automatically add to the PREMIS datastream based on certain Fedora-instantiated events?
- How do Fedora objects, datastreams and versions map onto PREMIS object types?
  - How do the relationships between these get encoded, is there a PREMIS vocabulary for these relationships?
  - We will need to draw lines between where Fedora successfully generates an audit trail itself and shouldn't be replicated, and where it doesn't.
- Do the requirements of the object/storage element indicate a need to retrieve a set of standard metadata from the low level storage module?
- Will we use PID for objectIdentifier? If so do we want to require a universally unique PID (UUID) to better support future migration of stored PREMIS?
- will we be encoding events that happen **before** ingestion into a Fedora repository?
  - Example: checksum taken at transfer time to compare with Fedora checksum
  - Example: any other information that needs to be stored (standard trusted digital repository checklist stuff, such as manifestations created), which describe actions taken to stabilize the object will create manifestations before ingestion
  - Which of these are events (e.g. "used Acrobat to create a PDF/A") as opposed to technical metadata (e.g. "this PDF/A. was created using Acrobat")?
- will we be making suggestions for what to put in the very vague technical metadata chunk of PREMIS?