

RAIDAR design decomposition

#music/raidar

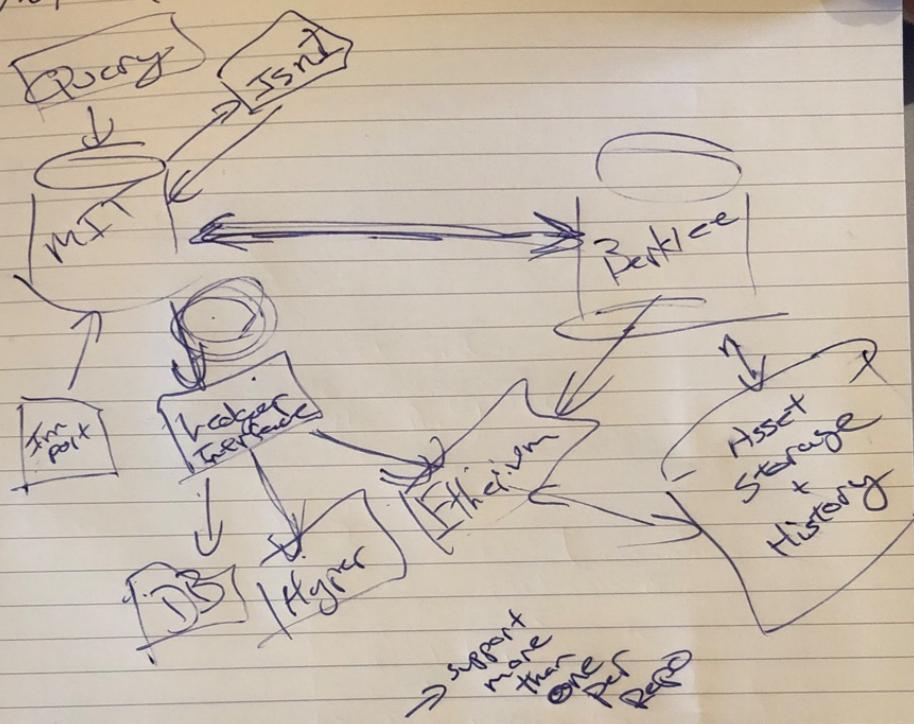
Actions

- admin setup for repo – ES

Functions & interfaces

- instances of repos
- repo-repo interface – does not assume full mesh connectivity
- repo-abstract ledger interface
 - behind that >1 ledger w/ its interface

3/15/2019 -- cont.



Abstract Ledger Interface

Data ~~import~~ Interface Input
Changes etc.

Query Interface

→ Pass ledger touches for flexibility

Repository to Repository Interface

Repository Info Interface
(~~not~~ trust make trust decisions)

Asset Storage + History

Common Lib of stuff / Defs

Repos Backend
admin/geo/rqz
Every Client

Data input client

↳ Bulk Loader

[Public/Private data]

indicate existence

RAIDAR Architecture paper initial thoughts

This paper describes:

- functional decomposition of RAIDAR
- key interfaces between major functions
- strategic requirements for functions and interfaces

See xxx for overall objectives for RAIDAR

Functional groups

RAIDAR provides two generalized services to its users:

- registering and updating:
 - works [examples][includes interim work products]
 - metadata about works [examples]
 - rights pertaining to the works [examples]
 - metadata about rights [examples]
 - in a later phase, rights licenses and metadata about rights licenses
 - provenance [define] for each of the above
- authoritative response to queries about the above.