OpenIO

Reference: OpenIO Core Solution Description (https://www.openio.io/resources/openio-core-solution-description)

Model and Primitives

Hierarchy: Namespace / Account / Container / Object

Namespace
: Gathers elements of physical infrastructure, hosts services, etc.

Account
: Don't need that in Hermes(?)

Container
: Unique name (in account), carries object management MD

Object
: BLOB, `PUT/GET/DELETE`, size limit configured at the namespace level, broken into chunks

- "... extensive use of filesystem extended attributes to store MD ..."
- 3-level MD hierarchy

"Consciousness"

- Secret sauce to solve the DPE problem
- Matchmaking between requests and services
- Balance between constraints on requests and quality of service

"The quality is calculated from metrics coming from the nodes themselves, they are shared among the grid of nodes to the *consciousness*. It is called the *score*. Through this feedback loop, each node knows in realtime what are the best nodes with the highest scores to handle a subsequent request."

The "consciousness feedback loop" figure is a form of the predictor/corrector loop.

Storage Policies

- Storage tier = class of hardware
- Storage policies are defined in namespaces
- Triplet: storage class, replication/redundancy, processing (on chunks!)
- Storage policies can be per container
- Object versioning available
