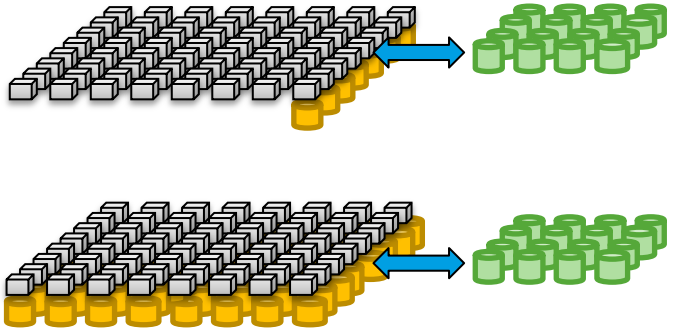
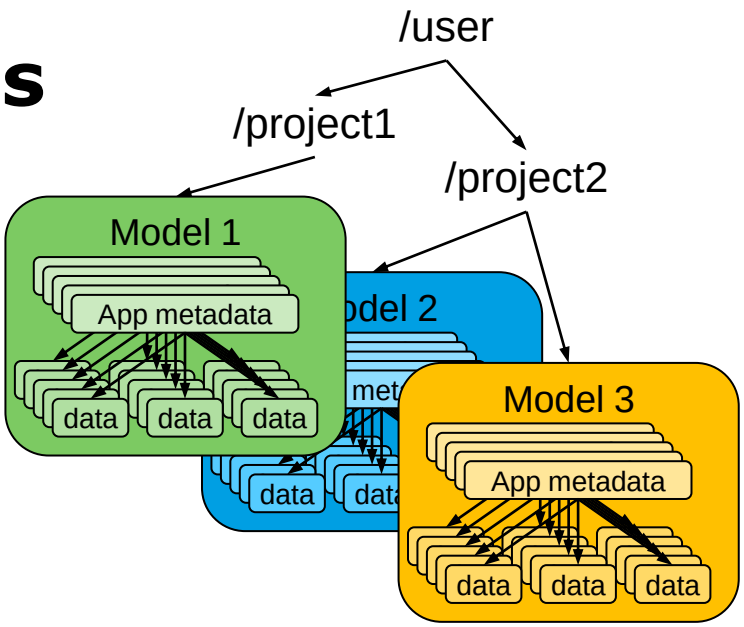
Exascale I/O challenges

Storage tiers

- Flash/NVRAM for peak performance (local)
- Disk for capacity (global)
- Fault management
  - Isolation between global/local storage
  - Timely notification / avoidance
  - Data integrity
- Pre-stage / post-drain
  - Scheduler integration
  - What to move and where to put it
- Edge v. ubiquitous storage
  - Post v. pre I/O function shipping
    - Filesystem v. memory extension / application objects
    - Resilience (checkpoint) strategy
  - Application & I/O comms interactions



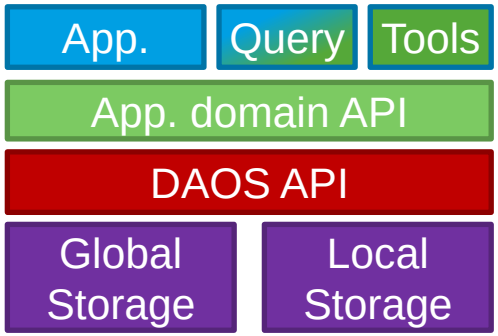
Exascale I/O challenges



Application data + metadata

- Explosive growth
  - Large, sophisticated models
  - Uncertainty Qualification
  - Billions – trillions of "Leaf" data objects
  - Complex analysis
- Filesystem namespace pollution
  - Keep filesystem namespace for storage management / administration
  - Separate namespace for application data + metadata
    - Distributed Application Object Storage (DAOS) containers
- Preserve model integrity in the face of all possible failures
  - Very large atomic, durable transactions
  - Integrity APIs at all levels of the I/O stack
- Search / query / analysis
  - Non-resident index maintenance & traversal / non-sequential data traversal
  - Move query processing to global storage
    - Same programming model as apps?

Exascale I/O challenges



Integrated computational model

- Multi-disciplinary requirements gathering
  - Application developers
  - High-level I/O library writers
  - Scheduling / workload management experts
  - Storage experts
- Compelling simulation / analysis platform
  - Programming & I/O models + runtime
    - Storage integral to workflow & resilience strategies
  - Fully expressive low-level application storage APIs
    - Full application access to virtualized hardware capabilities
      - Locality / concurrency / integrity
  - Mutiple application domain specific I/O models