iRODS & OSG

Tanya Levshina

Known Problems

The OSG doesn't provide efficient means to manage public storage supplied by OSG sites. These are the common issues raised by the multiple VOs, such as Engage, SBGrid, Dzero, NEES, SCEC and others:

- Most of the sites do not support dynamic allocation of storage and do not have tools for automatic management.
- VOs that rely on opportunistic storage have difficulties finding an appropriate storage, verifying its availability and monitoring its utilization.
- VO needs tools to manage metadata, OSG doesn't offer any common solution.
- The involvement of a Production Manager, Site Admins and VO support personnel is required to allocate or rescind storage space.

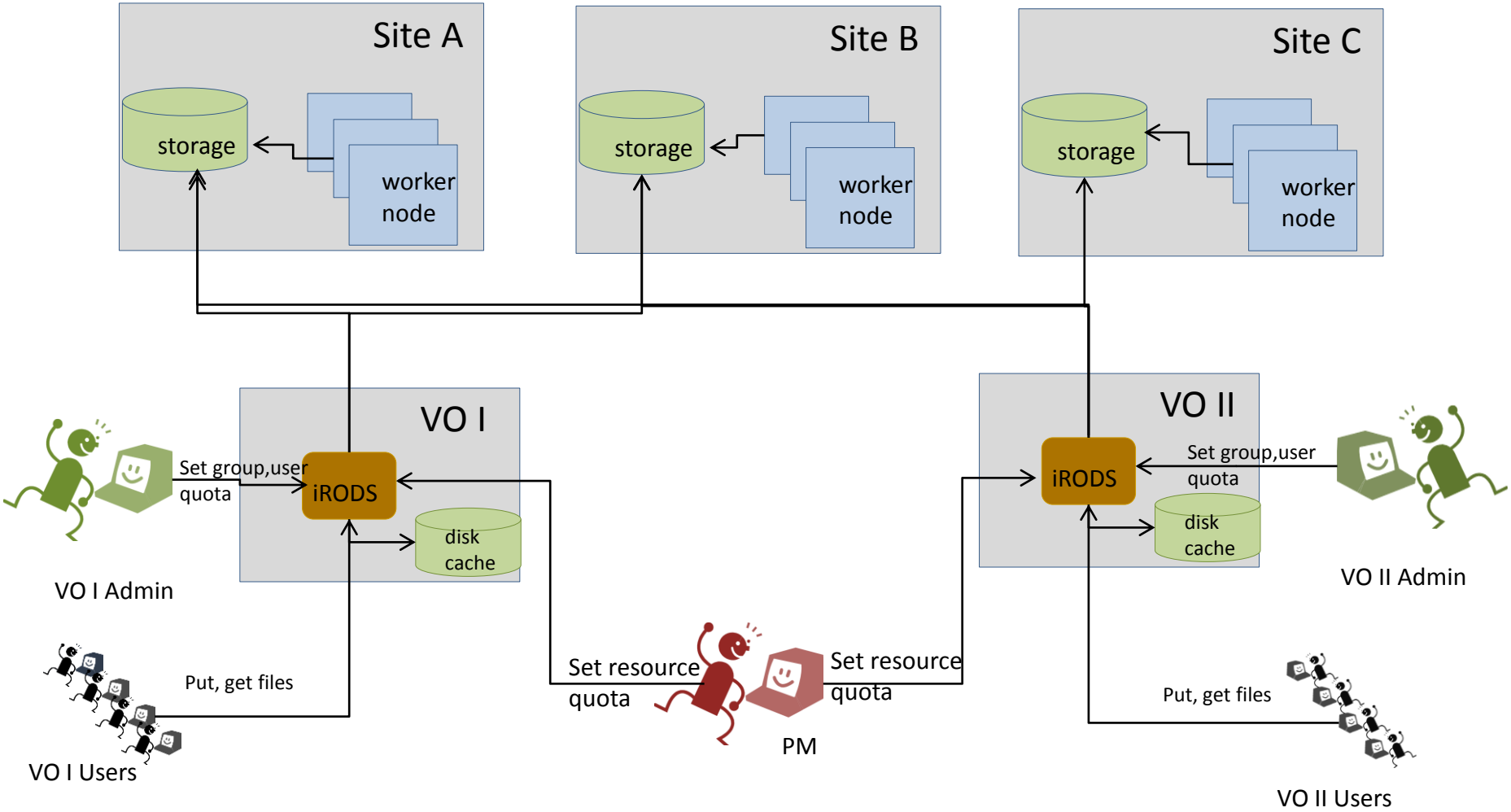
Requirements

- Allow OSG managed public storage allocation across all the participating sites.
- Allow VOs to sub-allocate storage per groups/users.
- Ease the burden of site selection for reads/writes.
- Allow policy-based and manual dataset replication.
- Provide automatic cleanup of expired files.

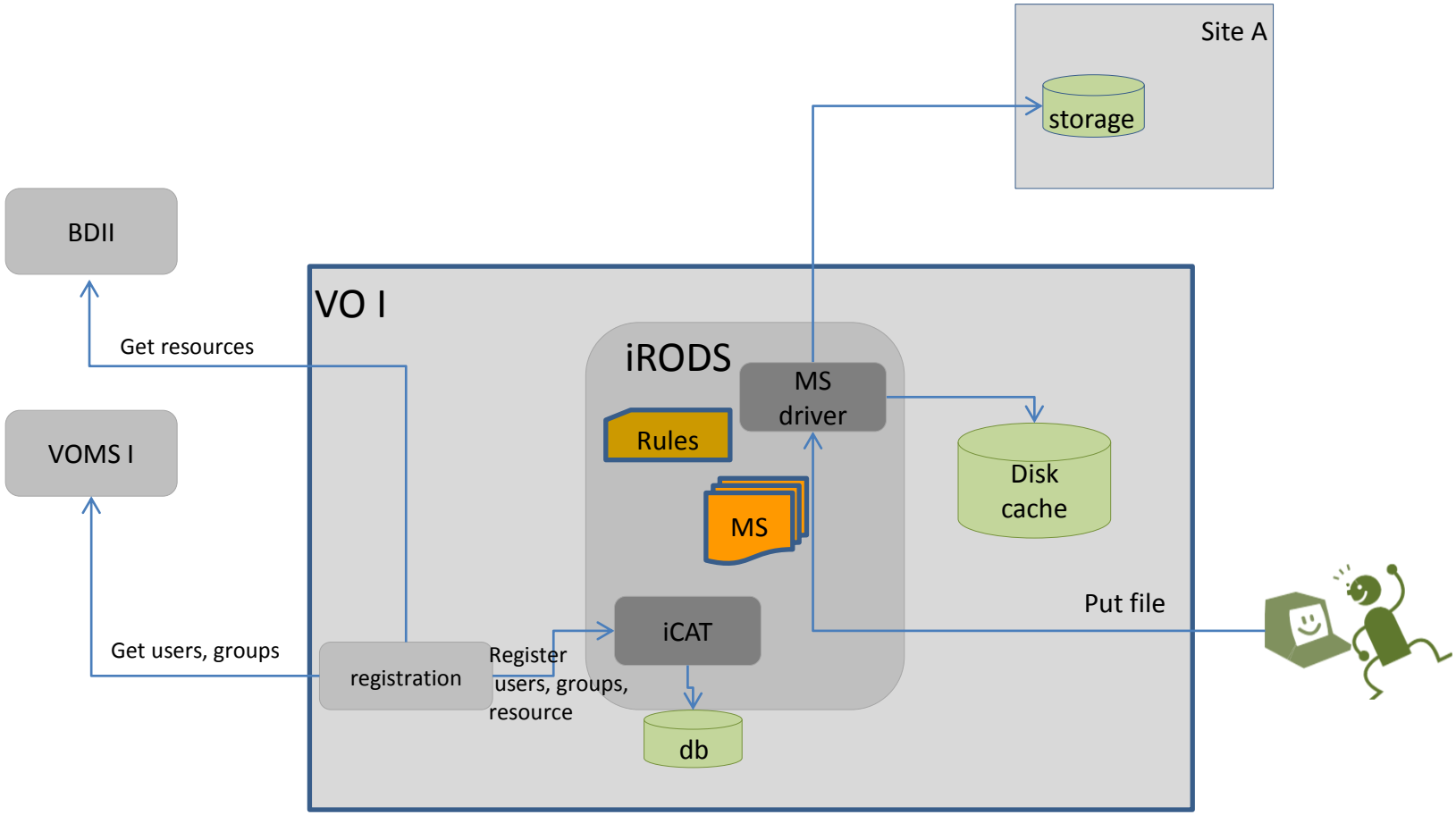
iRODS

- The Integrated Rule-Oriented Data System (iRODS) implements the policy-based data management framework.
  - handles various objects (resources, collections and files)
  - each object has a set of properties (metadata) associated with it
  - properties are enforced by polices (set of Rules)
  - various set of rules (actions/micro-services) that are controlled by policies are executed when object is accessed.
- iRODS performs transfers by
  - using implementation specific protocol to access POSIX compliant resources
  - using an external driver to Mass Storage. The driver should implement "put" and "get" methods to transfer entire files. File transfer is performed in two steps (disk cache is needed)
- The Metadata Catalog (iCAT) stores complete state information about the system in a database.

High Level Design

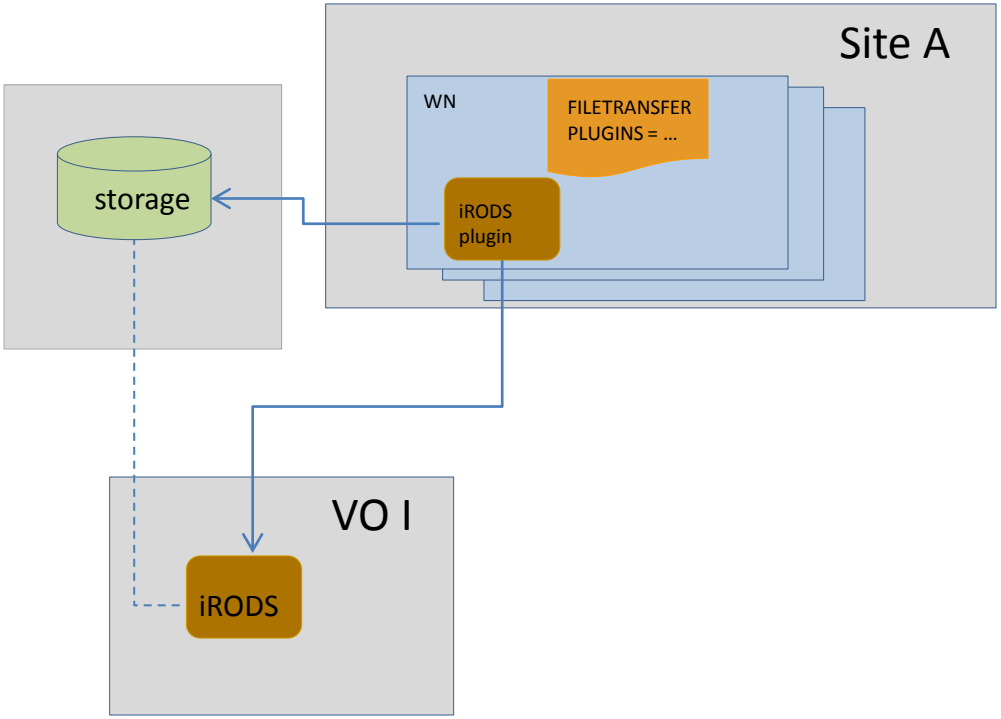


iRODS on VO Submission Node



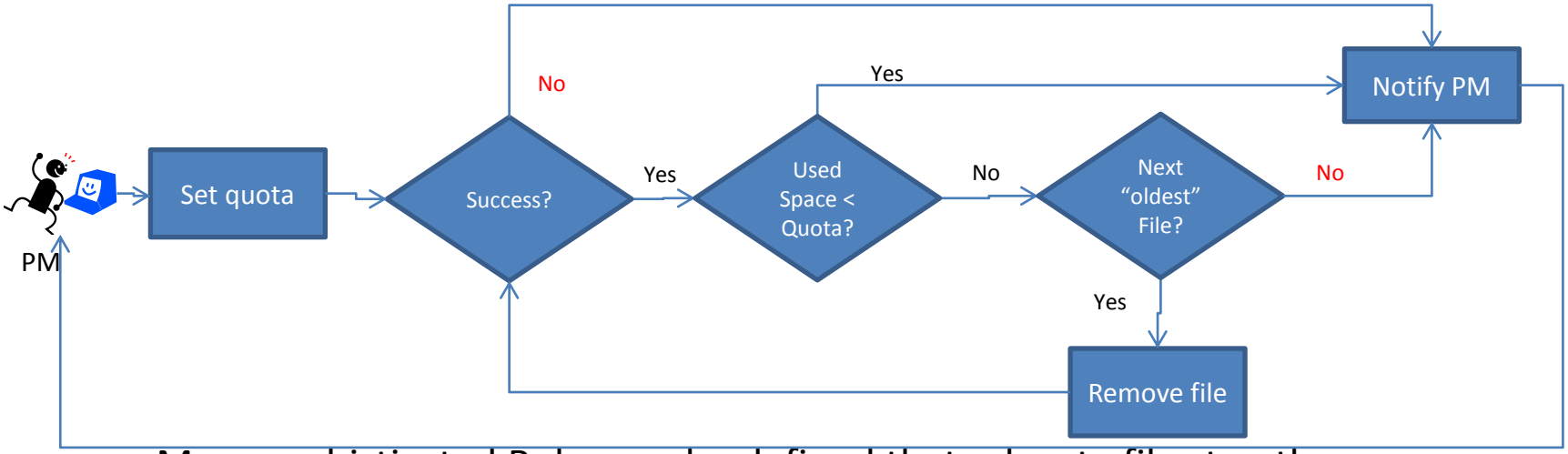
Grid Jobs and iRODS

- A user's job that is running on a worker node needs to upload file to some storage registered with IRODS.
- If this action is done outside iRODS framework then periodical storage -> iRODS sync is needed
- In some cases condor transfer plugin mechanism can be used :



iRODS and a Production Manager

- A Production Manager is responsible for storage allocation
  - has access to all VOs iRODS.
  - handles quota allocation using iadmin commands
  - sets/changes quota per resource, VO
  - queries information about quota, resources, and space utilization
  - disables resource (draining)
- Example: a PM decreases quota for a particular resource for a VO. iRODS invokes a simple Rule that triggers the following actions:



More sophisticated Rules can be defined that relocate files to other resource before starting files removal.

iRODS and VO

- Deployment:
  - iRODS is running on a VO-owned machine.
  - iRODS is configured with a OSG storage plugin.
  - iRODS node has additional disk space for staging data to the OSG.
- VO Admin actions:
  - sub-allocates storage (manages quota per user groups or users)
  - maintains policies that support data managing according Public Storage utilization agreement
  - queries information relevant to quotas, space availability, space usage per user/group
- User actions:
  - pre-stages file via iRODS (two hops: local cache, remote storage)
  - Uploads files from worker node to a durable storage (two hops or one hop and sync)
  - Download files to the worker node directly from storage (contact iRODS if LFN – PFN is needed)

iRODS and OSG Sites

- provides a separate storage allocation for iRODS enabled VOs – the “public storage”
- provides authorization setting suitable for VO iRODS to manage VO user’s data
- allows to run storage-iRODS synchronization (can be a VO grid job)

Pros

- iRODS has to be deployed only by VOs, minimal burden for Sites.
- Allows OSG to manage inter-VO storage allocations
- Allows a VO Admin to sub-allocate resources to VO group and/or users.
- Can reuse policies developed by the iRODS community
- Provides means for metadata management (LFN->PFN mapping, number of replicas, resource utilization, quotas)
- Can be expanded to include more iRODS servers (zone, federation)

Cons

- Relies on cooperation between OSG and VO Admins
- Requires development:
  - External driver for OSG storage
  - OSG Storage->iRODS synchronization
  - Condor transfer plugin for iRODS
- Set new requirements on a VO-owned machine:
  - A VO should have a dedicated node with sizeable disk cache (~TBs)
- File staging is done in two steps (client -> iRODS staging -> public storage)

End Goals

- OSG Sites should be able to delegate public storage management to the OSG.
- By using OSG public storage, data management costs for VOs should go down.
- Possible solution for the distributed data authorization issues is to use “authorization token” approach (ALICE - AliEn). In this case all proxies accessing public storage are signed by OSG, and include file access authorization in the VOMS attributes.

References

- iRODS Home Page
  - https://www.irods.org/index.php/IRODS:Data_Grids,_Digital_Libraries,_Persistent_Archives,_and_Real-time_Data_Systems
- OSG Storage Group iRODS evaluation(by Ted Hesselroth):
  - https://osg-docdb.opensciencegrid.org:440/cgi-bin/RetrieveFile?docid=967
  - https://twiki.grid.iu.edu/bin/view/Storage/FederatingStorageElementsForOpportunisticUse
- Condor transfer plugin example (Parag Mhashilkar)
  - https://cdcvs.fnal.gov/redmine/documents/250
- Authorization of Data Access in Distributed Storage Systems (Derek Feichtinger, Paul Scherrer, Andreas J. Peters)
  - http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1542739