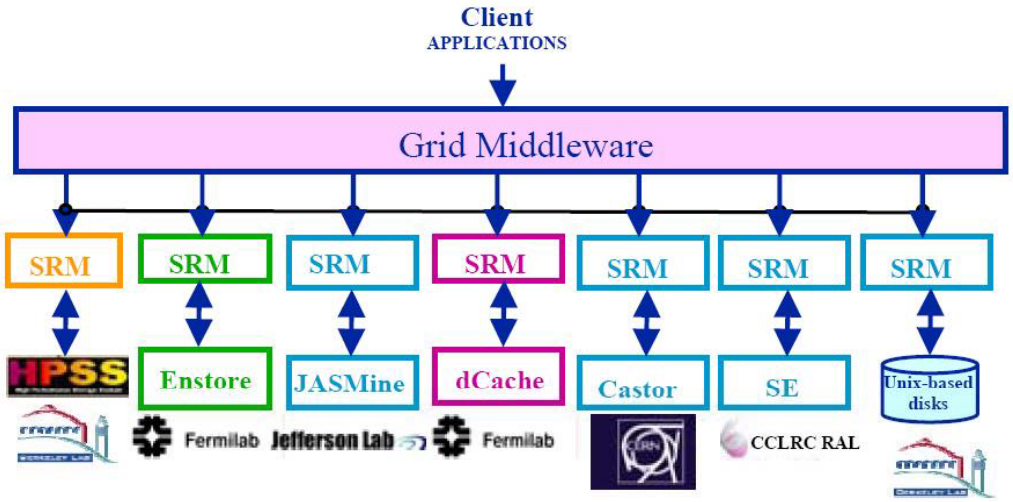
The Development of SRM interface for SRB

Fu-Ming Tsai
Academia Sinica Grid Computing
sary357@twgrid.org

What is SRM?



- SRM is an unique interface for accessing diffident backend storages for diffident middleware.
- Easy to develop applications to adapt different backend storages.
- Provide space and file management on the storage system.
- SRM is the web service interface and the implementation usually depends on the backend storage technology.

Why SRM(Storage resource management)?

- **Storage Elements (SE) can use different type of technologies**
  - CASTOR, dCache, DPM, BeStMan,...,etc.
  - DRM (Disk Resource Manager)/TRM (Tape Resource Manager) /HRM (Hierarchical Resource Manager)

- **Grid middleware needs to access files with an uniform interface**
  - Manage storage resources
  - Not a file transfer protocol

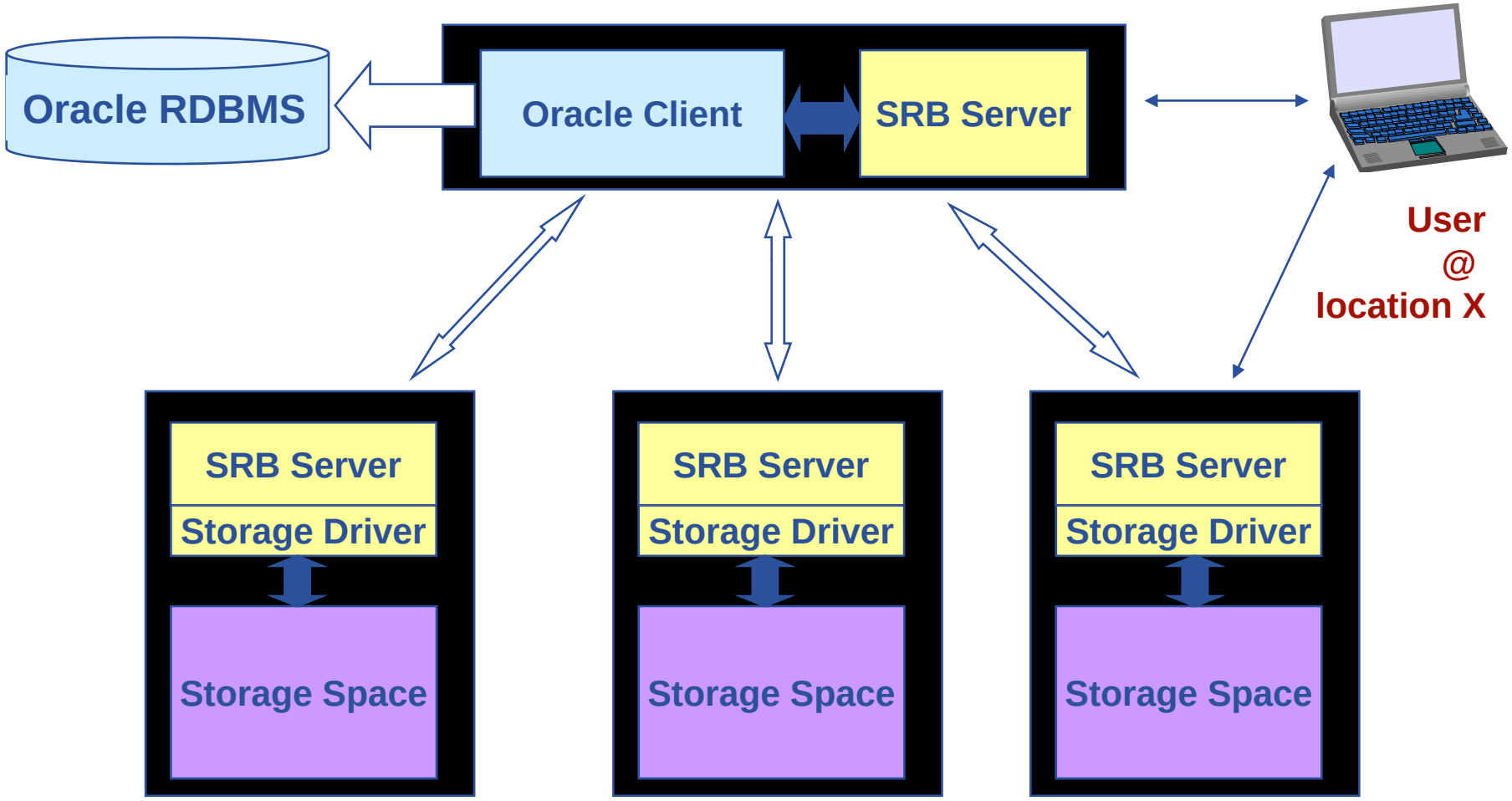
SRM features

- Provides space management
- Provides an uniform access interface
- Manages DRM/Tape/HRM
- Does not transfer files itself.
- Manage the life time of file

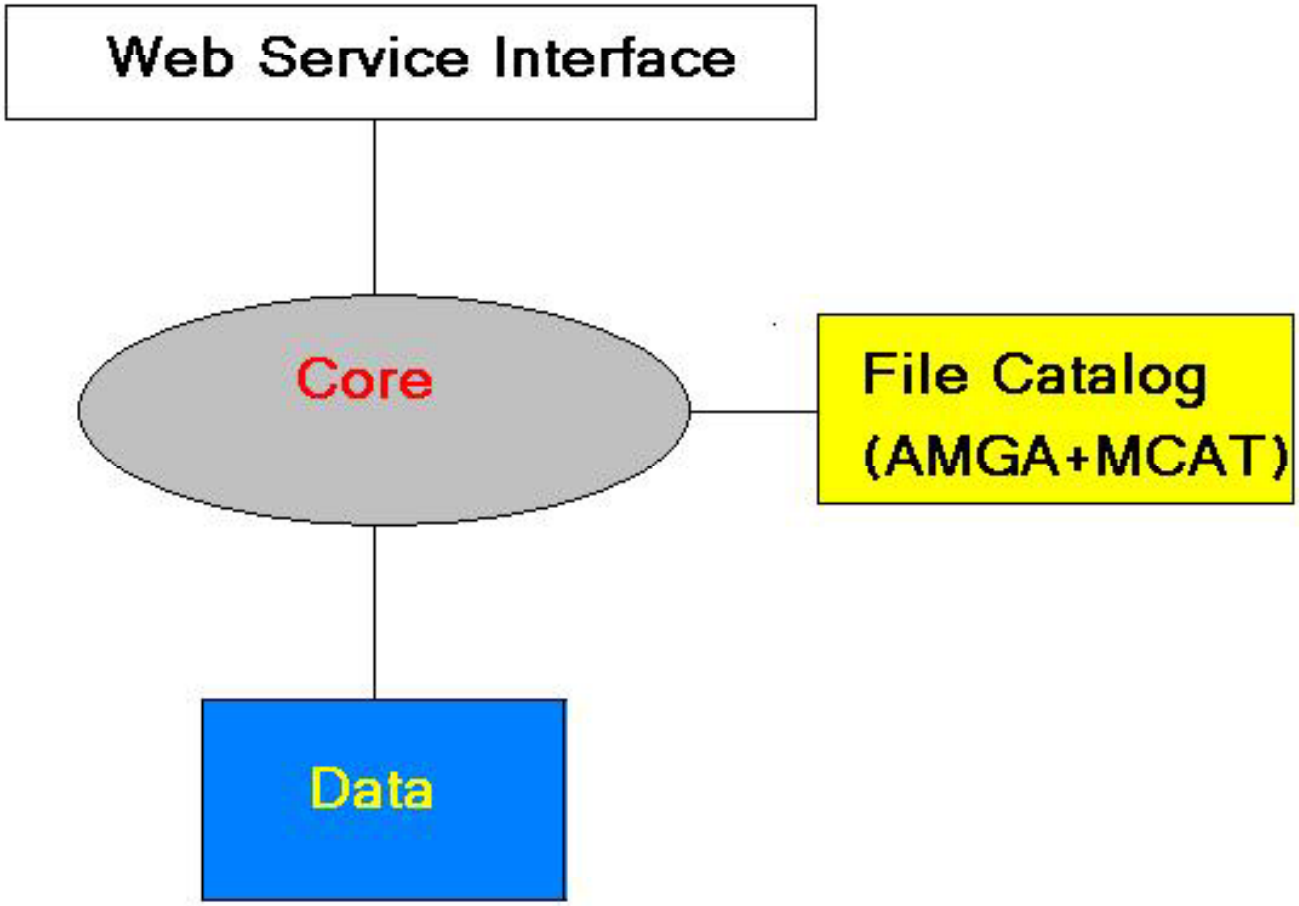
What is SRB (Storage Resource Broker) ?

- Developed by San Diego Supercomputer Center
- A distributed file management system (Data Grid), based on a client-server architecture.
- Allows users to access files seamlessly across a distributed environment, based upon their attributes rather than just their names or physical locations.
- It replicates, syncs, archives, and connects heterogeneous resources in a logical and abstracted manner.

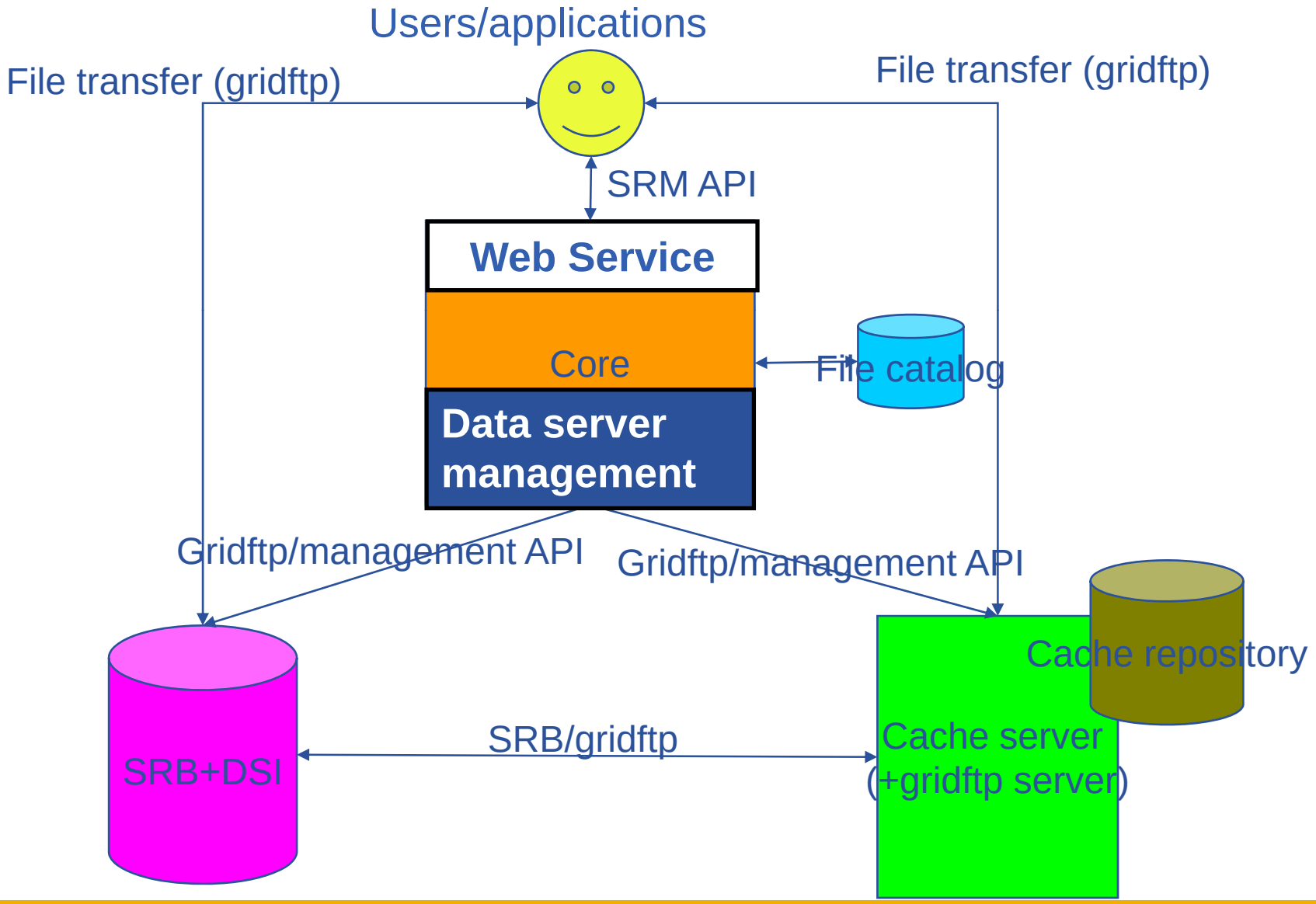
SRB Physical Structure



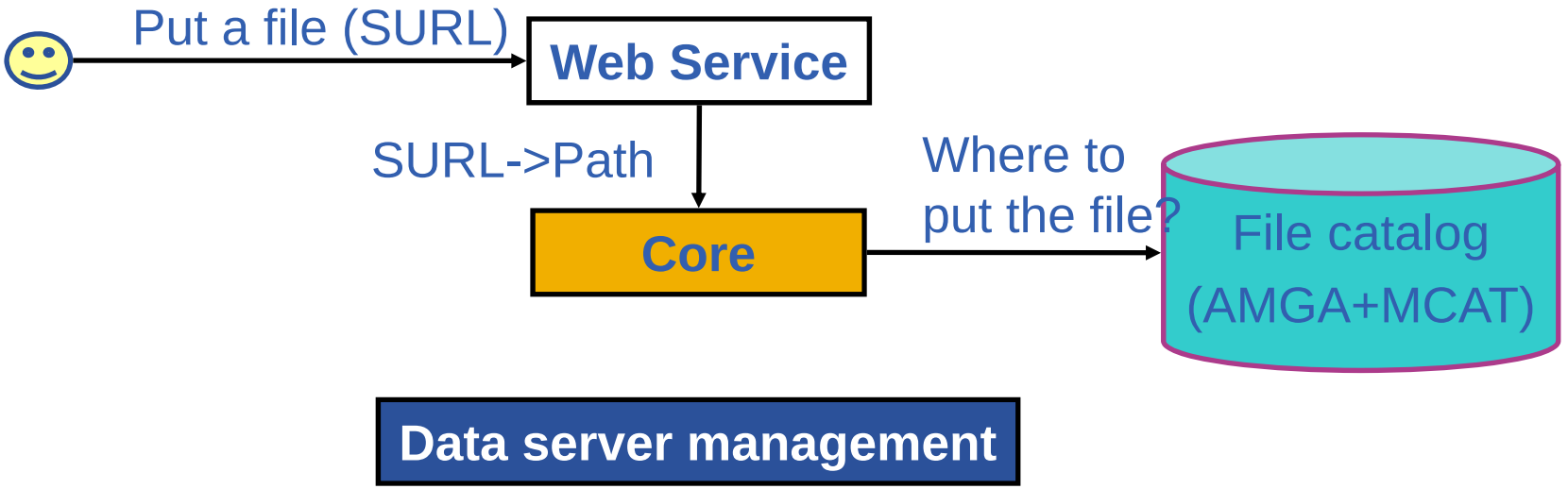
SRM for SRB – Our Implementation

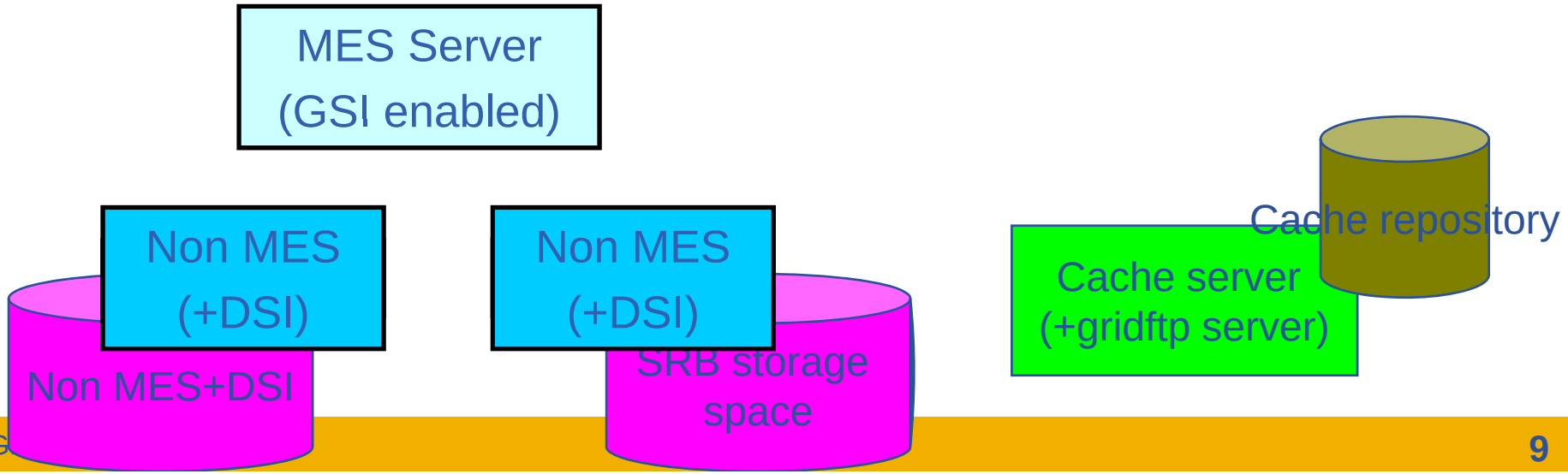


Architecture Overview

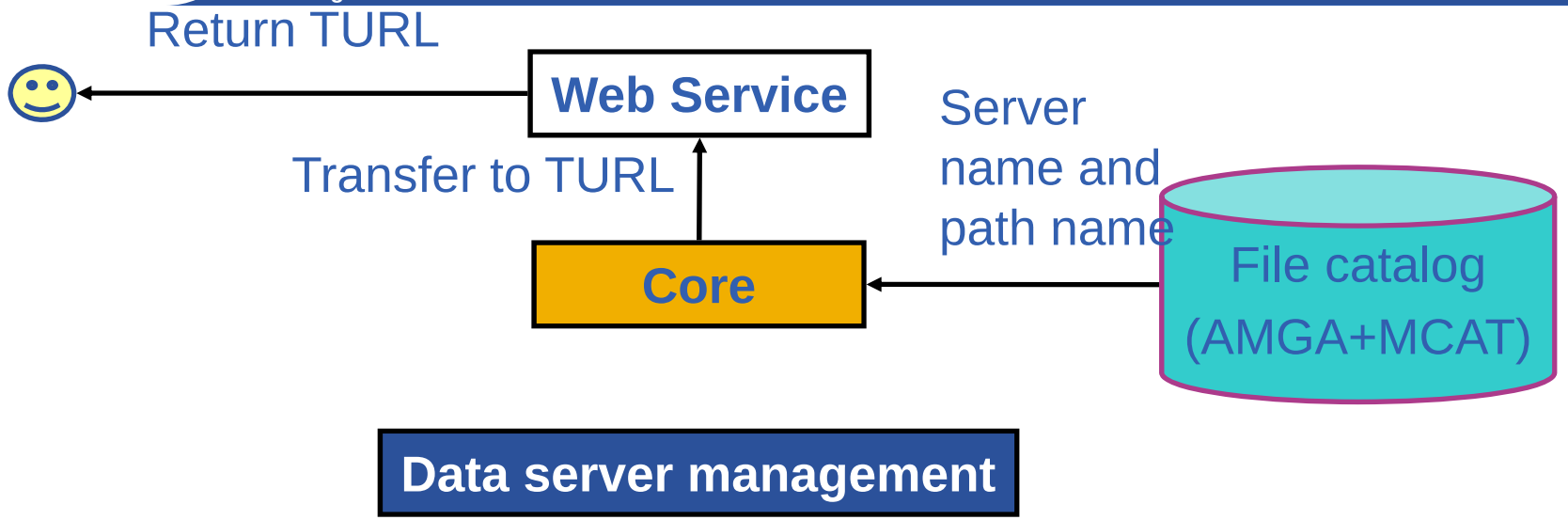


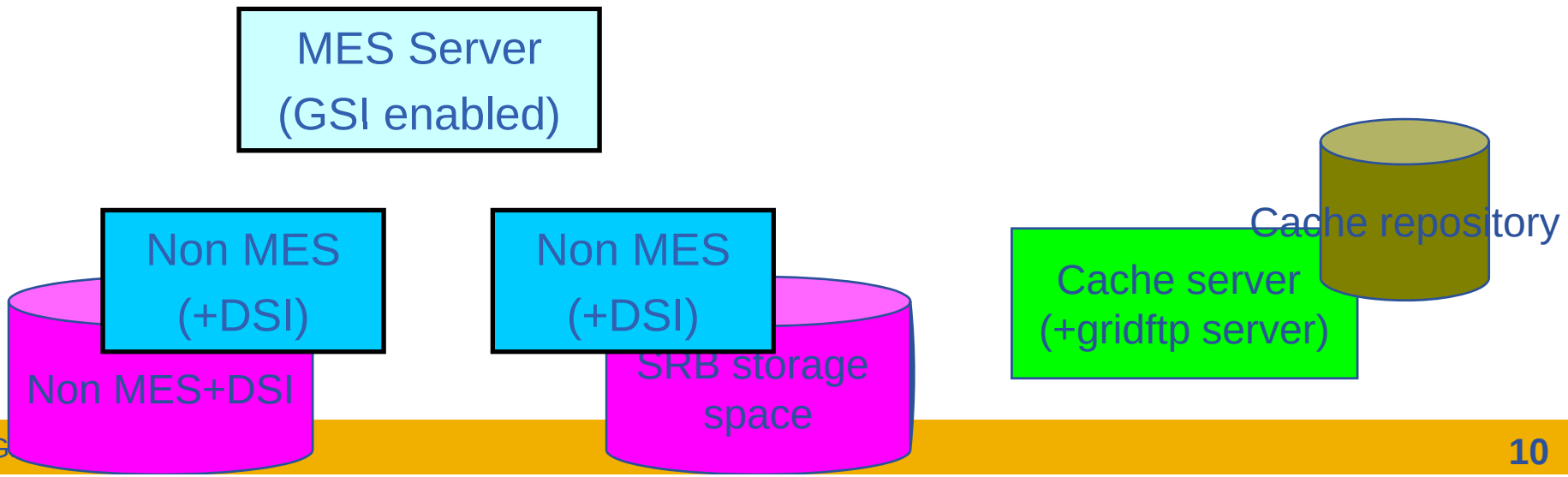
Architecture Overview (cont.)



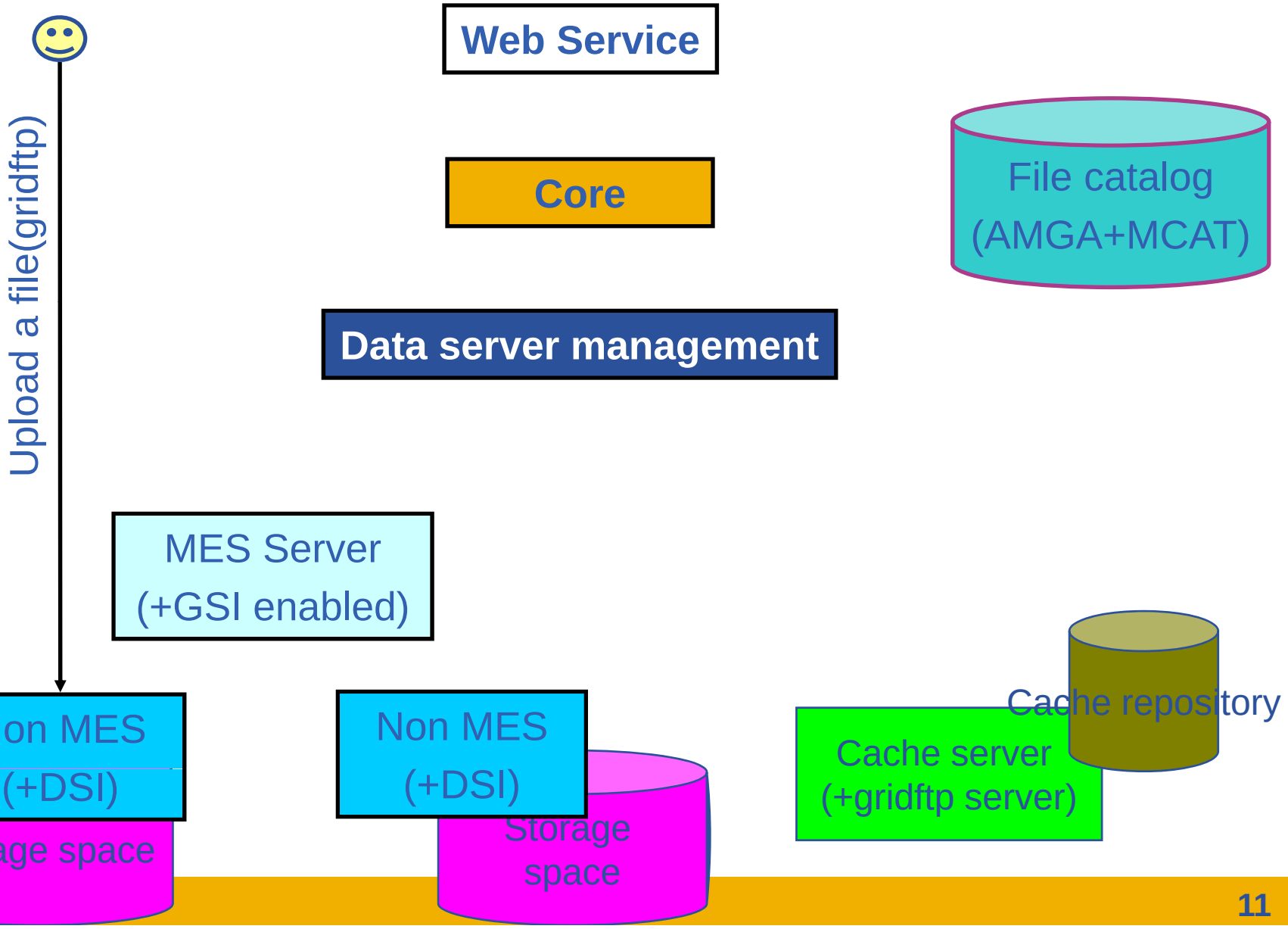


Architecture Overview (cont.)

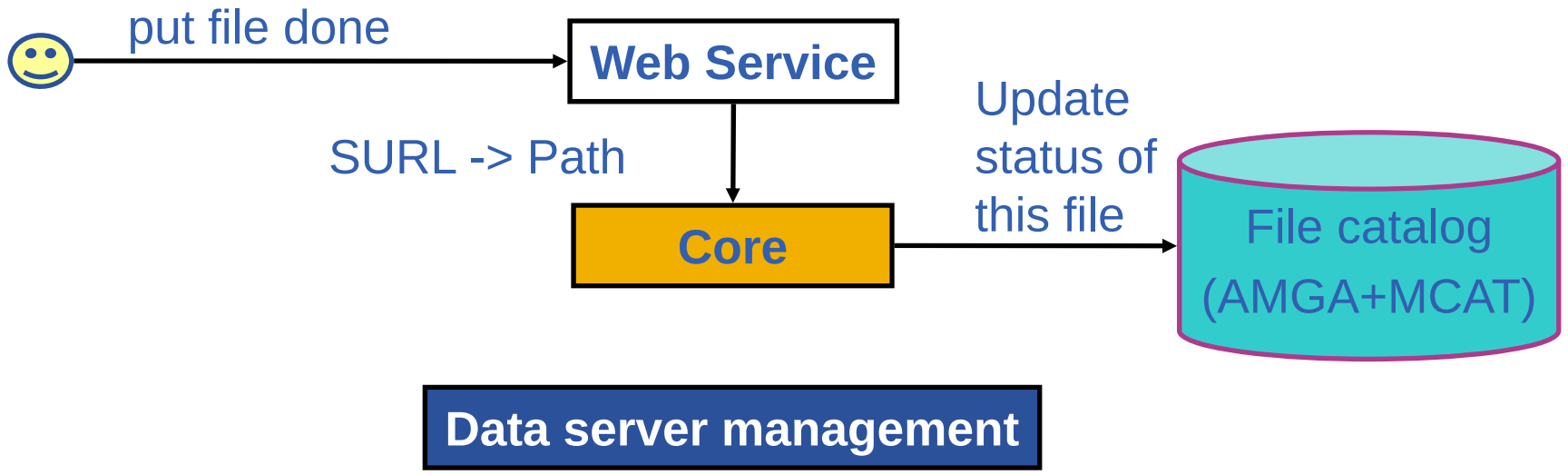


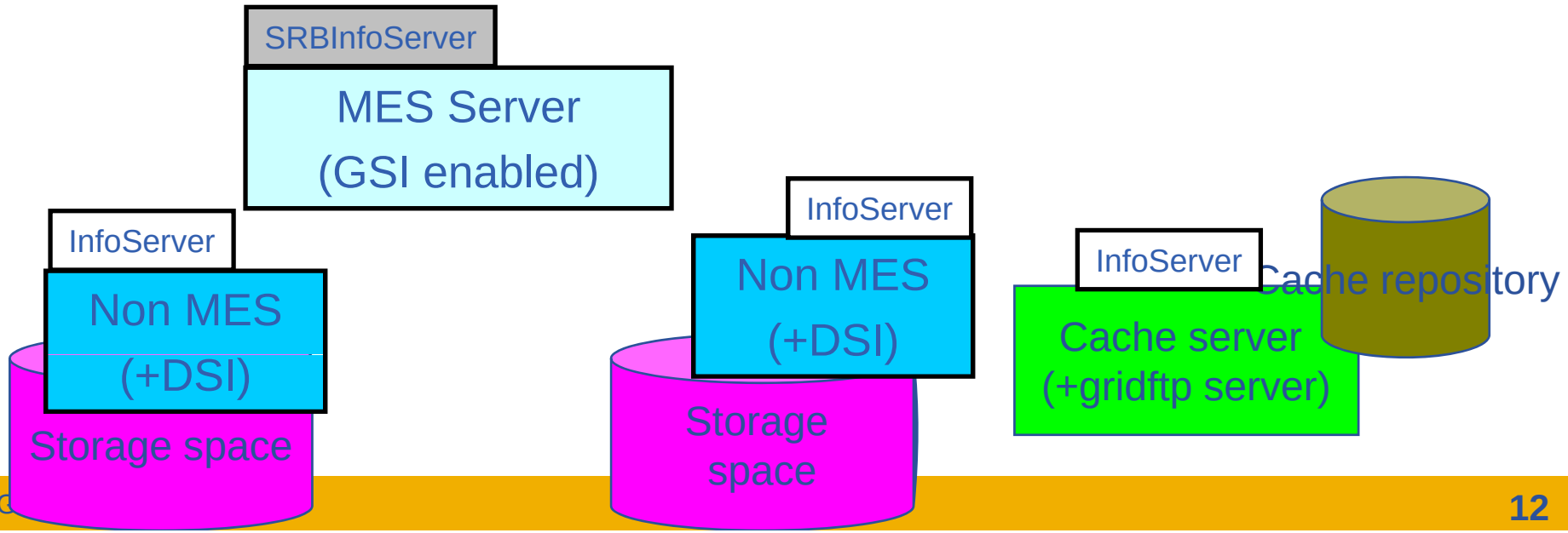


Architecture Overview (cont.)

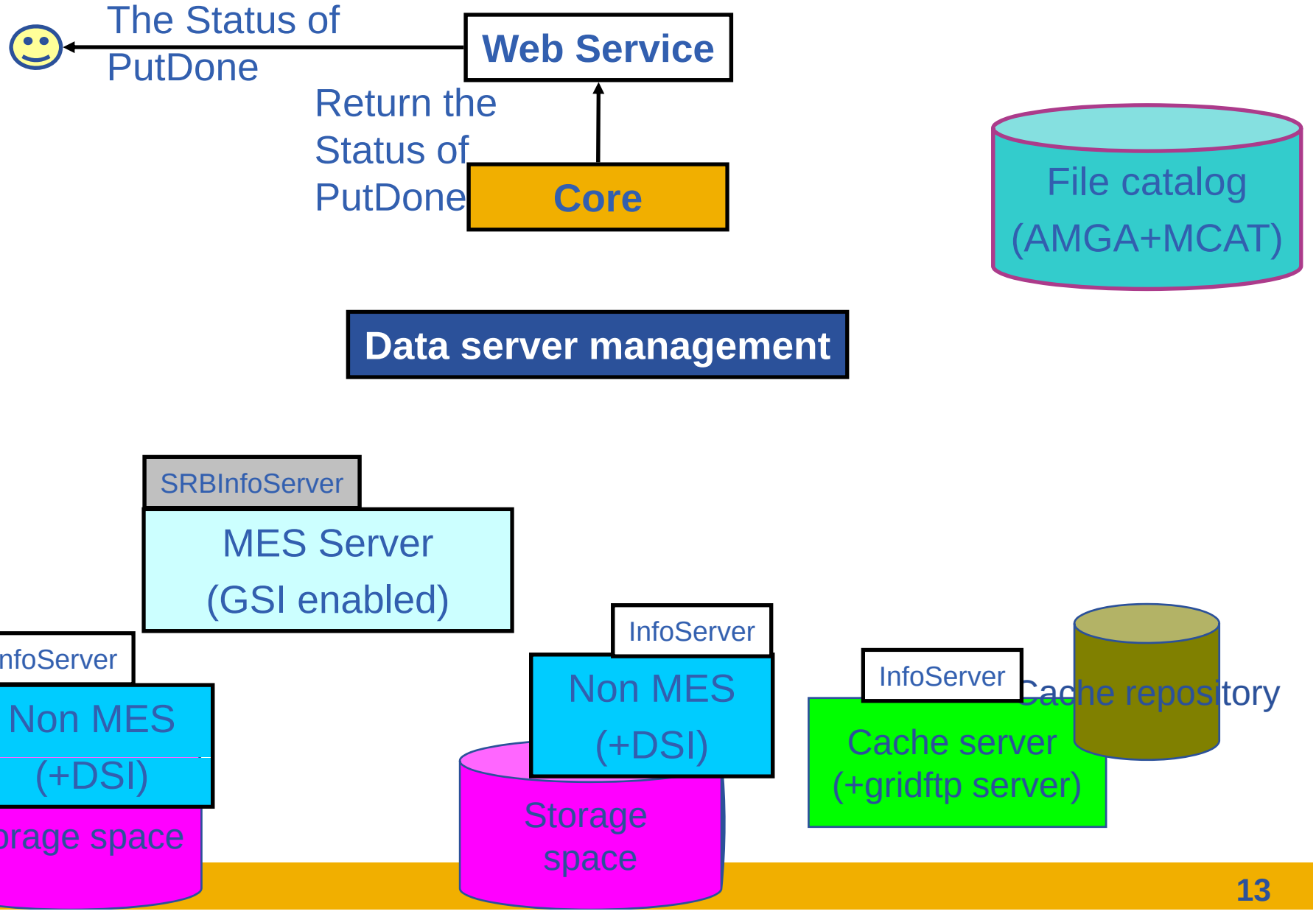


Architecture Overview (cont.)

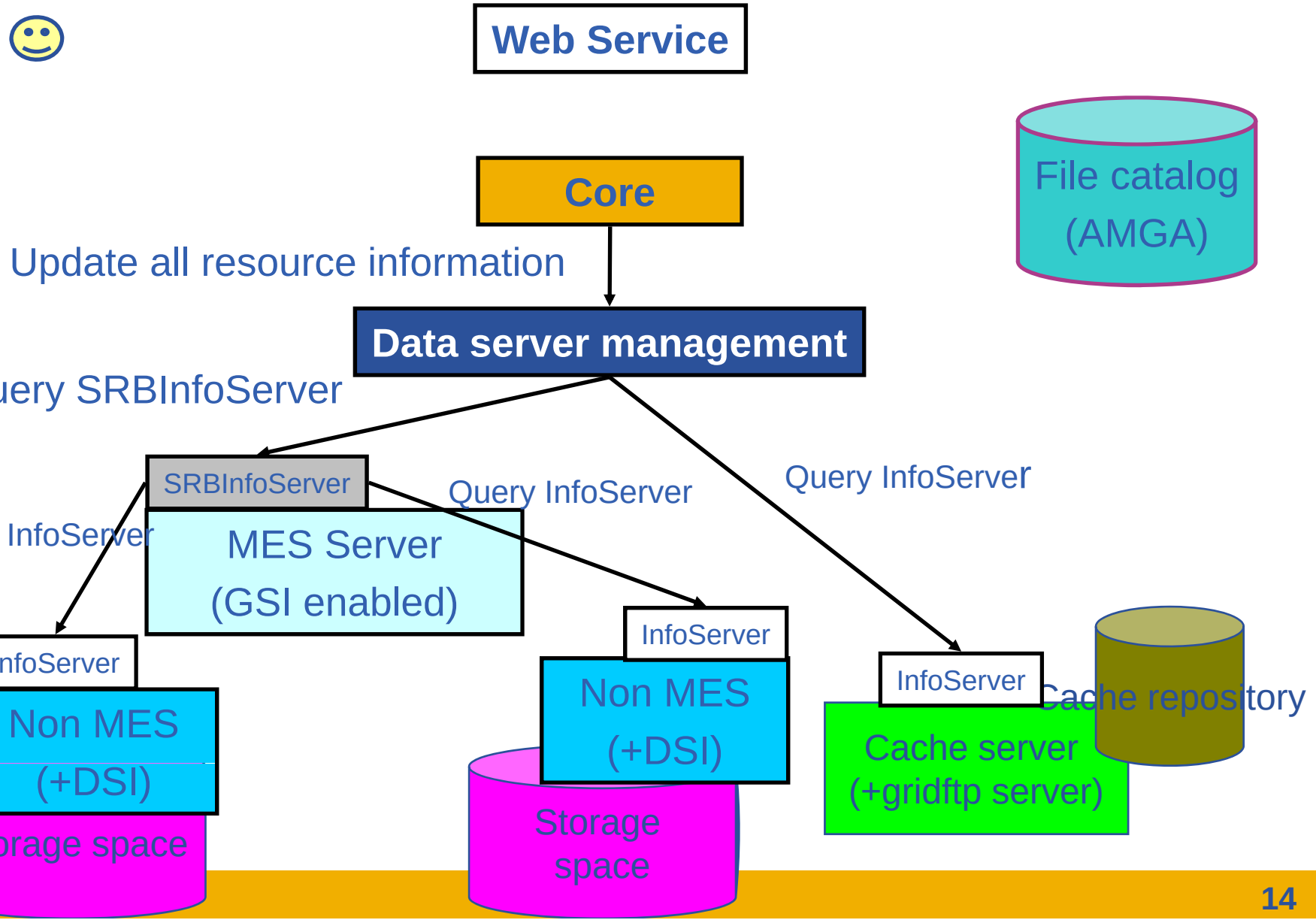




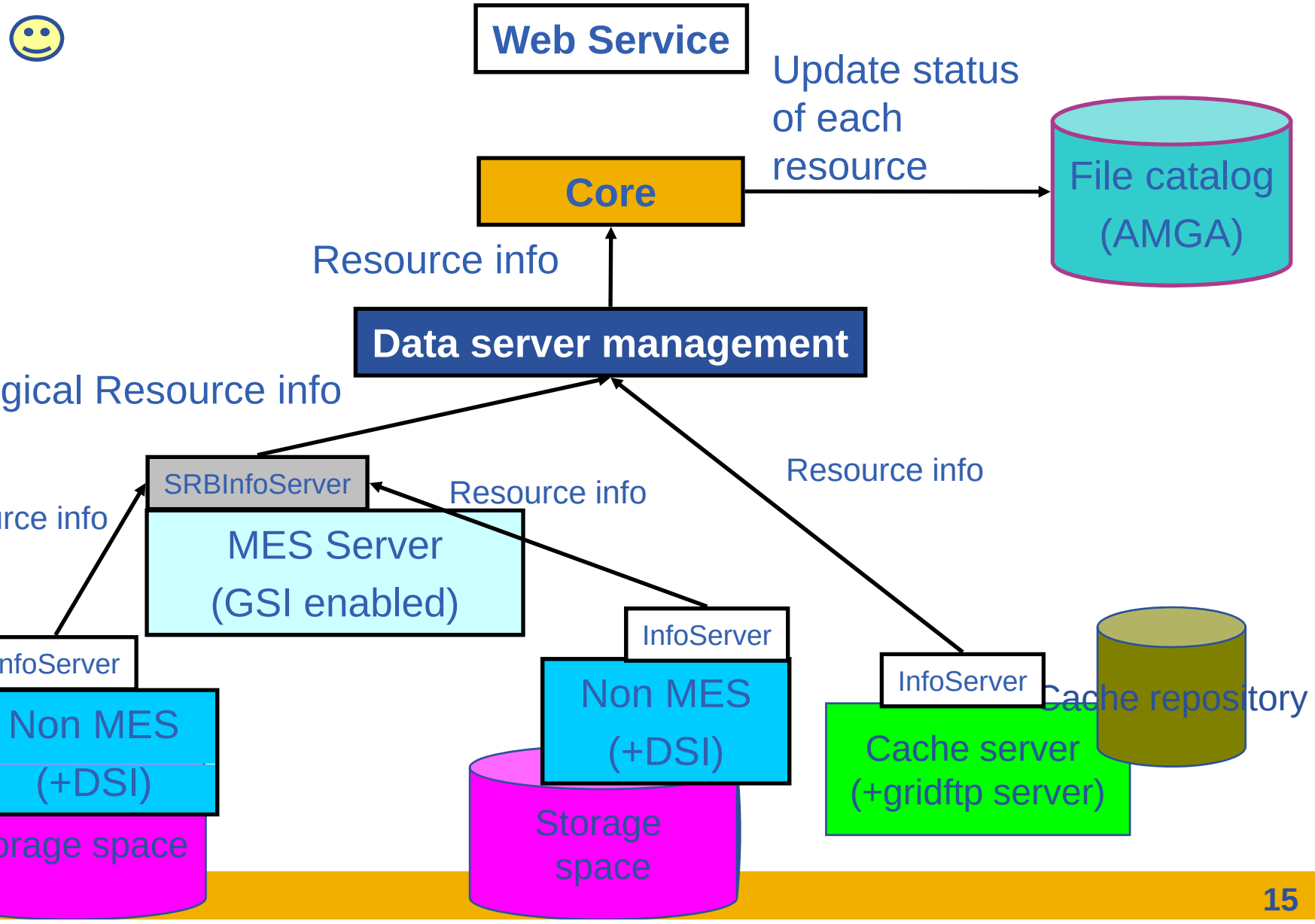
Architecture Overview (cont.)



Architecture Overview (cont.)



Architecture Overview (cont.)



Support Flexible File/Space Types

- **SRM system has a caching mechanism and has to take care of SRM issues like file lifetime, space management,…,etc.**
  - Volatile space
  - Durable space
  - Permanent space
- **In our implementation**
  - Volatile space & Durable space
    - In the temporary storage space.
  - Permanent space
    - In the SRB env.

Checking Disk Status

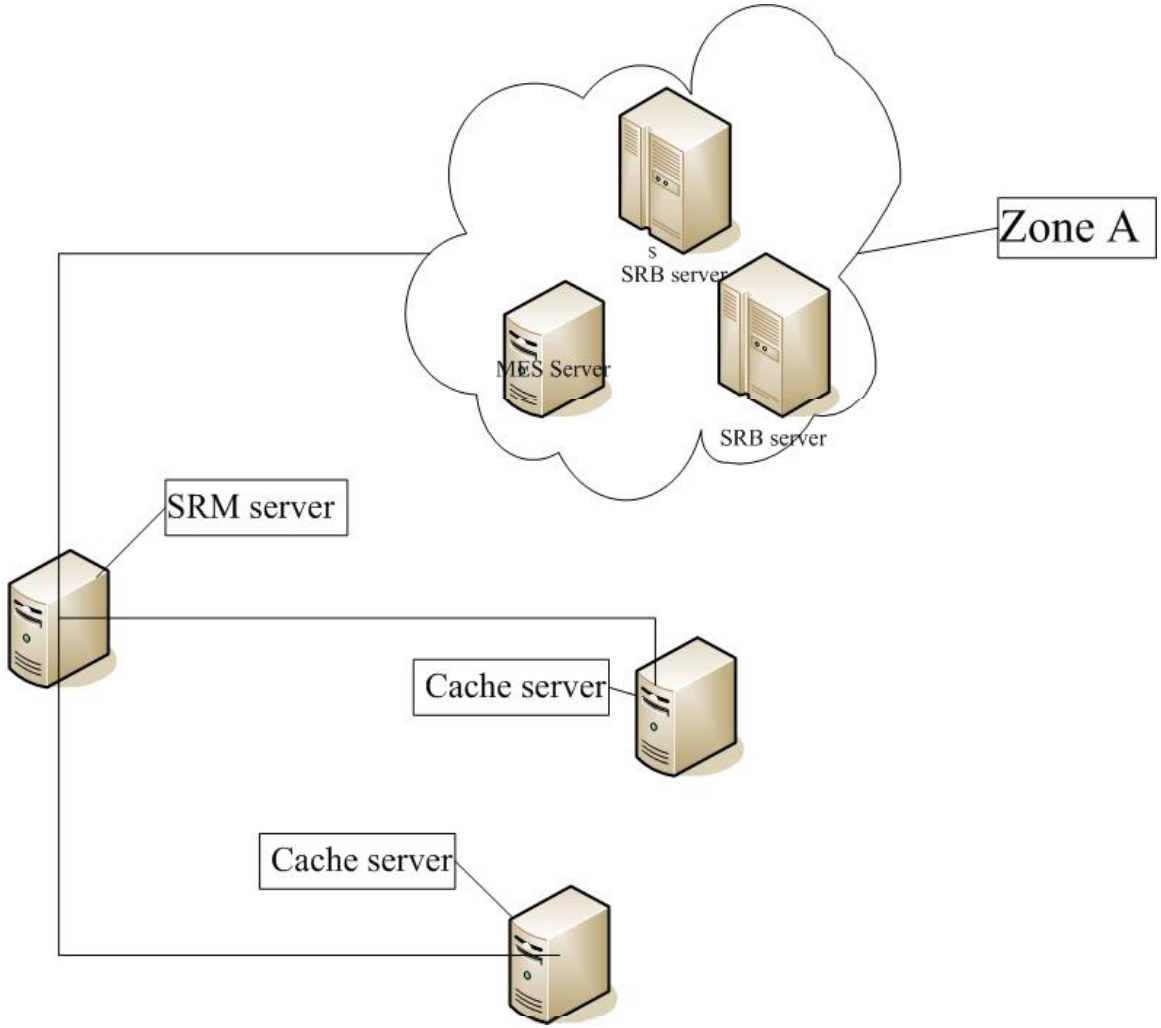
- **How to get the disk usage of the space?**
  - Need to know the free and used space on cache server and SRB server
  - SRB does not provide the mechanism to monitor resource usage
  - We need to know the usage
    - Space management
- **In our implementation**
  - InfoServer:
    - Deployed on cache server and non-Mcat enabled SRB server
  - SRBInfoServer:
    - Deployed on Mcat-enabled SRB server

Naming

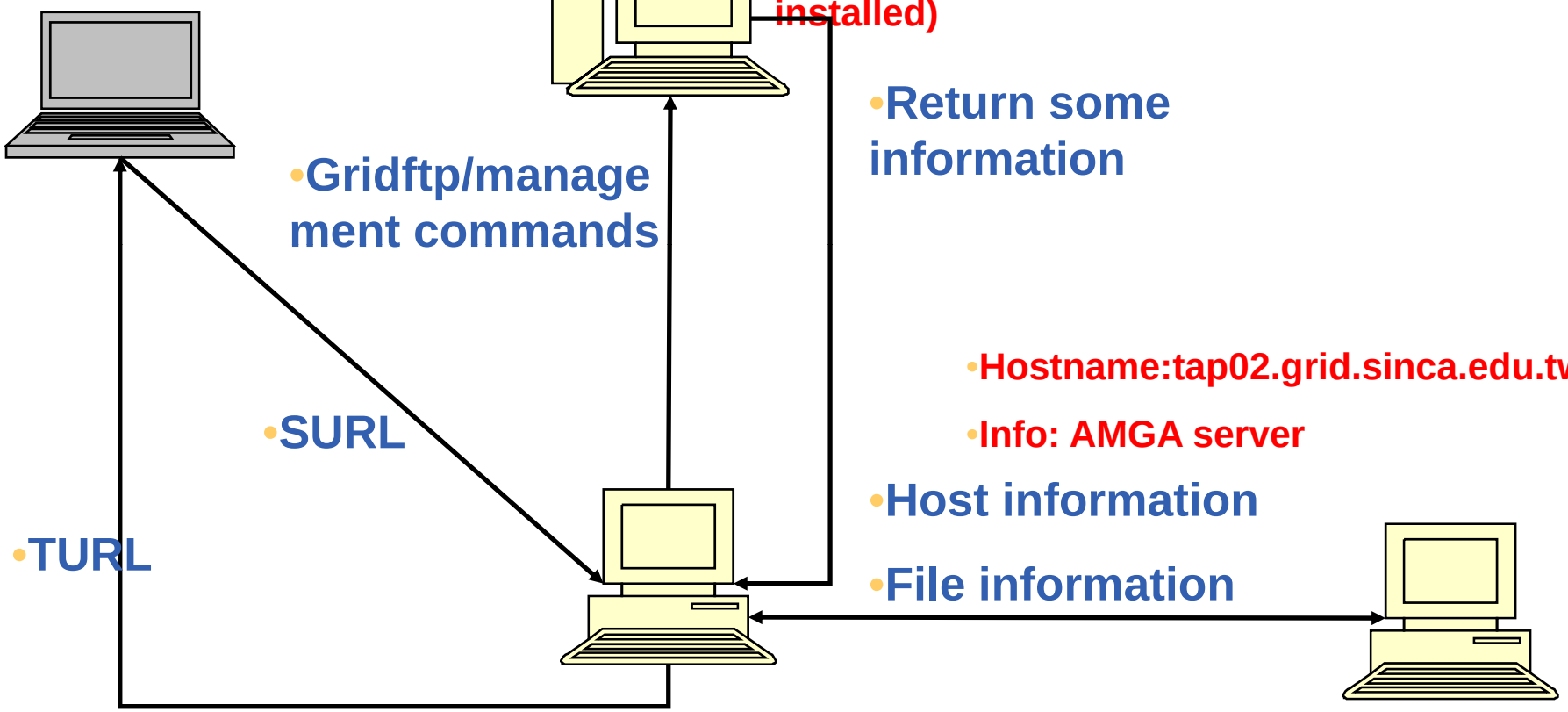
- **Mapping between SURL, TURL, and file name**
  - Different naming space on cache server and SRB server
    - /XXX/YYY/ZZZ: cache server
    - /Zone/home/UserAccount.UserDomain/: SRB server
    - How to map them??

- **In our implementation**
  - Use AMGA to provide uniform naming space
    - /XXX/YYY/ZZZ
  - How to map to TURL
    - On cache server
      - *Use the mounted point gotten from InfoServer and use this mounted point to store user's data, then reply TURL*
    - On SRB server
      - *Try to login and get user's home collection. Use user's home collection to store user's data, then reply TURL.*

Typical Use Case



Testbed deployment



Demo

- **Test the functions that we have implemented.**

- **Access data in**
  - Cache server
  - SRB server

Demo(1)

- **Test the following functions**
  - SrmPing
  - SrmGetTransferProtocols
  - SrmSetPermissiong
  - SrmCheckPermission
  - SrmGetPermission
  - SrmMkdir
  - SrmRmdir
  - SrmPrepareToPut
  - SrmPutDone
  - SrmPrepareToGet
  - SrmReleaseFiles
  - SrmRm

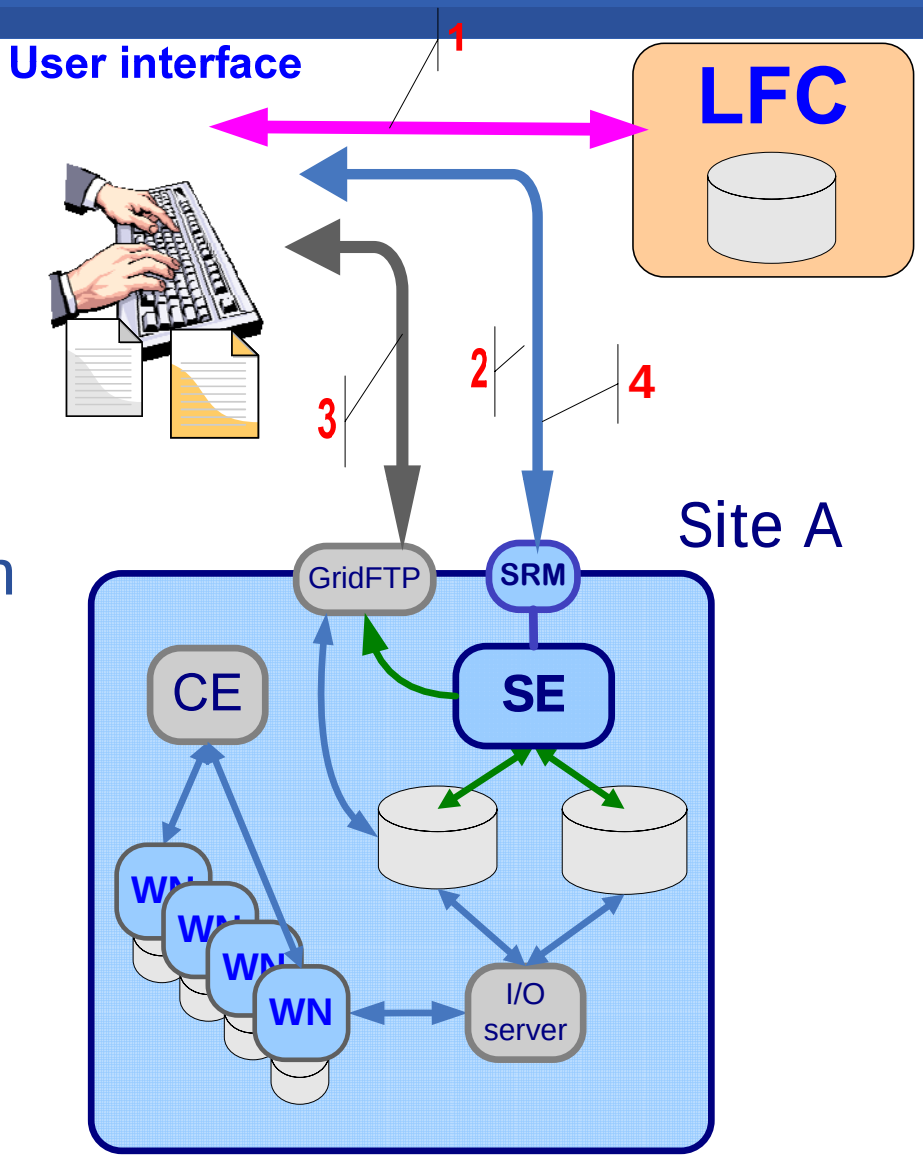
Demo(2)

- Put/get data in/from cache env through SRM
- Put/get data in/from SRB env through SRM

Demo(3)

User needs to store data in SE (from a UI)

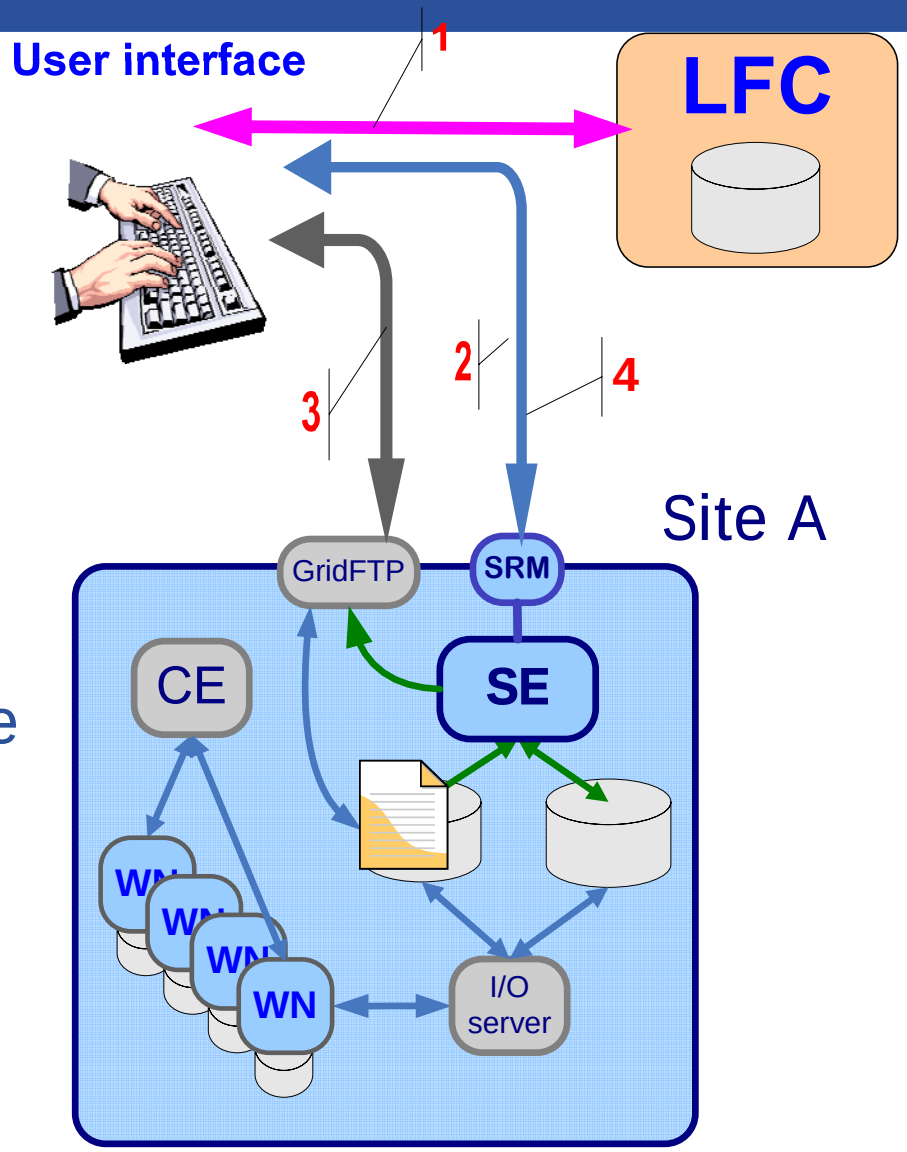
1. Create a new LFN entry in LFC, return a SURL.
2. srmPrepateToPut (SURL)
3. Transfer the file
4. srmPutDone (SURL)



Demo(3)

User needs to retrieve (onto the UI) data stored into SE

1. Query the file catalog to retrieve the SURL from the LFN.
2. srmPrepateToGet (SURL)
3. Transfer the file (read)
4. srmReleaseFile (SURL)



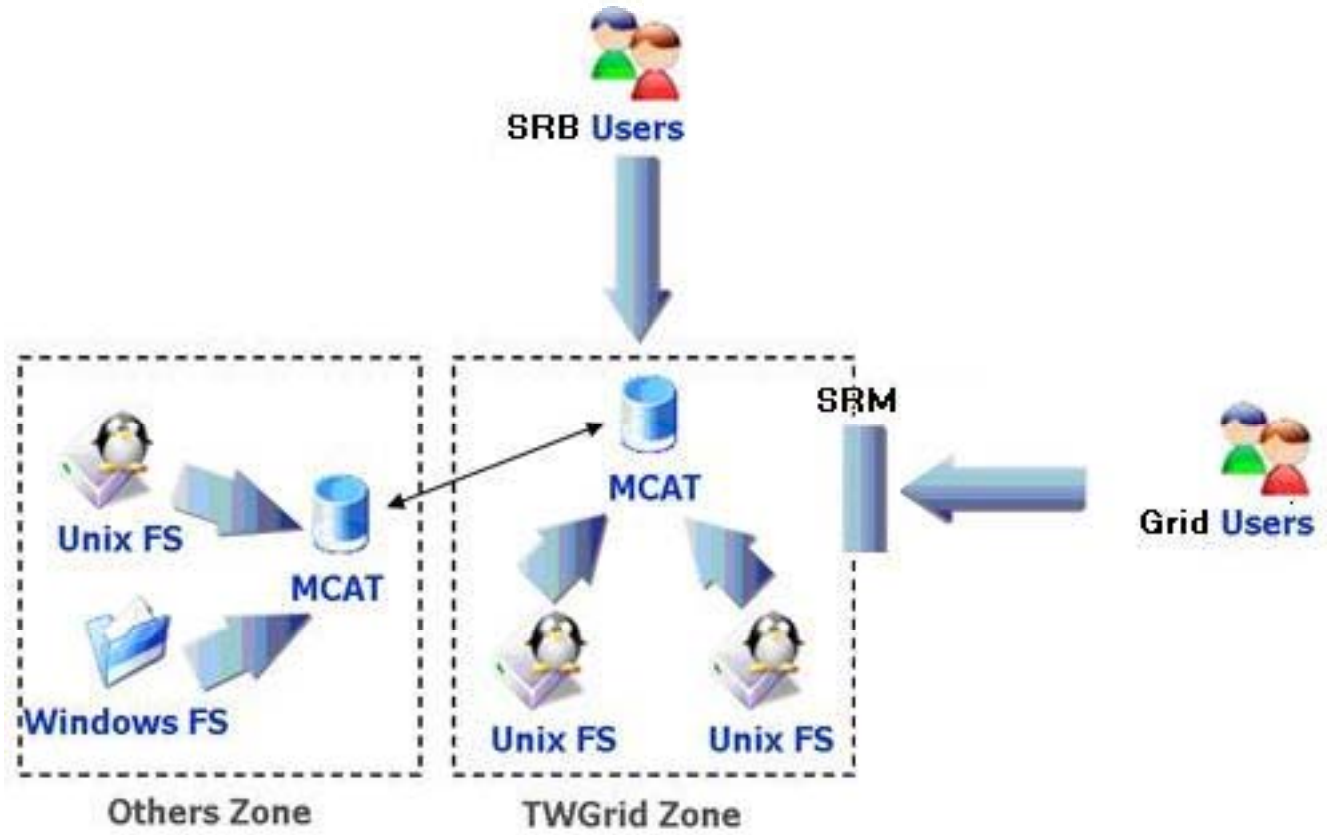
Progress

- **What we have done:**
  - We have finished the parts about authorizing users, web service interface (just interface), gridftp deployment, and SRB-DSI.
  - If the data is stored in cache server
    - Some functions like directory functions, permission functions, ..,etc has been implemented.
  - If the data is stored in SRB server
    - Permission functions are being implemented with Jargon API.
- **What we are doing:**
  - Still focusing on the part about integrating MCAT into our implementation.
    - Permission functions
    - Directory functions
    - And more...

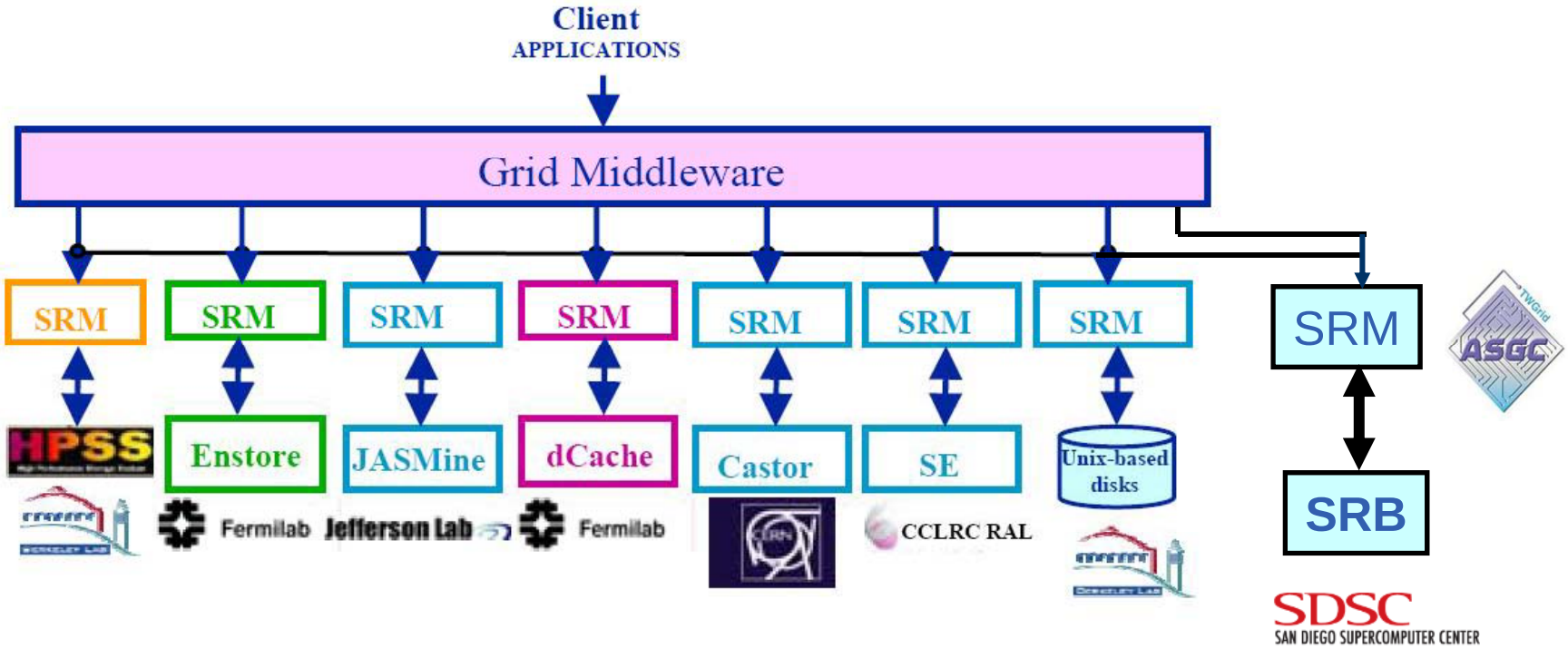
Future Plan

- **Short term**
  - Implement interface compatible SRM version 2.2.
- **Mid-term**
  - Interact and test with other data management systems such as Castor, dCache,...,etc.
- **Long-term**
  - Interoperate with gLite.
    - Hope glite users who join VO can access files and resources in SRB and vice versa.

Future Plan


SRB Users
SRM
MCAT
Unix FS
MCAT
Windows FS
Unix FS
Unix FS
Grid Users
Others Zone
TWGrid Zone

Future Plan


Client
APPLICATIONS
Grid Middleware
SRM
SRM
SRM
SRM
SRM
SRM
SRM
HPSS
Enstore
JASMine
dCache
Castor
SE
Unix-based disks
Fermilab
Jefferson Lab
Fermilab
CCLRC RAL
SRM
SRB
ASGC
SDSC
SAN DIEGO SUPERCOMPUTER CENTER

References

- **SRM-SRB interface project wiki page**
  - http://lists.grid.sinica.edu.tw/apwiki/SRM-SRB
- **SRM working group:**
  - http://sdm.lbl.gov/srm-wg/
- **SRB:**
  - http://www.sdsc.edu/srb/index.php/Main_Page
- **Globus:**
  - http://www.globus.org
- **CoG:**
  - http://wiki.cogkit.org/index.php/Main_Page
- **Axis:**
  - http://ws.apache.org/axis/

Thanks for your attention