SRM Developers' Response to Enhancement Requests

Michael Ernst
DESY

SC3 Detailed Planning Workshop - CERN

Disclaimer

- Statements made in here are preliminary
  - Details, in particular priorities and schedules have to be negotiated with management according to overall priorities
- SE Types covered in this response
  - CASTOR (O. Barring, B. Couturier)
  - SRM/dCache (R. Kennedy, T. Perelmutov, J. Bakken)
  - DPM (J.-P. Baud)

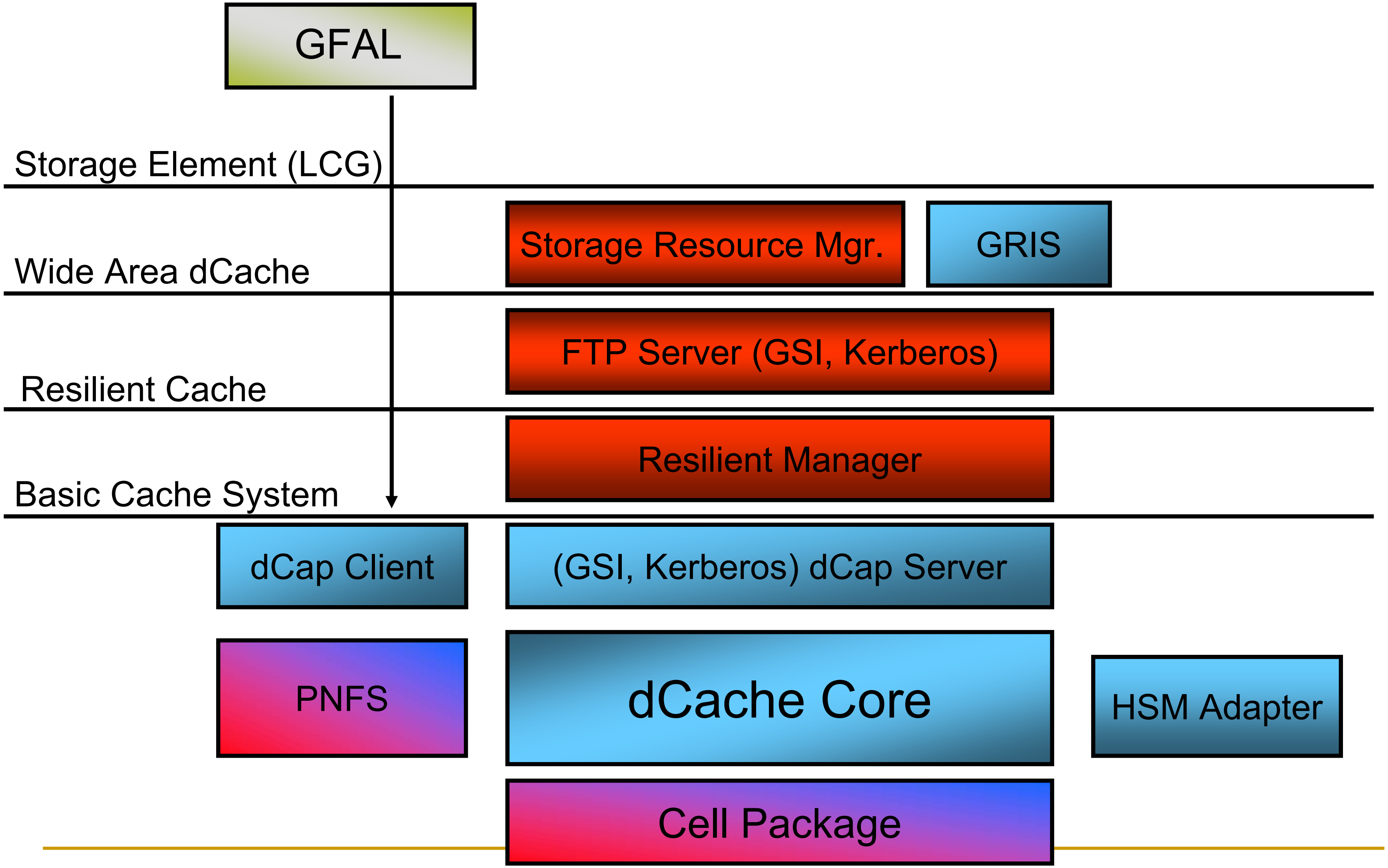
Remarks

- Agreement achieved by Baseline Services WG w.r.t. a common “LCG-SRM” set of functions that the experiments need is a big step forwards
- Provides solid goals the Developers can build on
- Communication between Experiments and Developers need improvement
  - WG formed during the LCG DM workshop was well attended by the Developers while the experiments where missing

Enhancements required by Experiments

- Timescales
  - Experiments: “Necessary to expose service well before start of SC4”

Example: dCache Functionality Layers



Enhancements required by Experiments

- Pin/Unpin Functionality
  - Pins have a lifetime
    - Operational issue when used by average user who could interfere w/ the rest of community, especially in an HSM environment w/ Migration capabilities
  - Implementation Plans
    - CASTOR: Could be implemented (STBN)
    - SRM/dCache: “Implicit (triggered by srmPrepareToGet) pins and unpins always part of SRM/dCache implementation
      We do not see a value in explicit pins/unpins, (when TURL is not requested), but we agreed to implement it anyway”
    - DPM: Implemented in current release.
      - Time can be specified (prepareToGet, separate command) or admin specified defaults (pool specific) can be used

Enhancements required by Experiments

2. Relative Path in SURLs
   - Requirement as we understand it
     - Client does not have to query information system for site specific information the server should know about – is that correct?
   - Developers have mixed feelings about this request
     - SRM/dCache Developer: We do not believe these are needed, SRMs do not support “current directory concept”
   - Implementation Plans
     - CASTOR: Could be implemented, would be a Hack (STBN)
     - SRM/dCache: Currently no plans to implement Relative Path
     - DPM: Not implemented yet, but could be made available on timescale of SC4

Enhancements required by Experiments

- ❸ Permission functions: All experiments would like the permissions to be based on roles & DN’s. SRM should be integrated with VOMS
  - (At least) two major issues
    - Standardization of VOMS functionality w.r.t. Authorization and respective Interfaces (LCAS, LCMAPS)
    - Implementation of this will largely depend on the support from the underlying storage system and availability of effort to implement it (i.e. support for ACLs vs. UNIX permissions only)
  - Implementation Plans
    - CASTOR: Fine as long as UNIX permissions were OK, much harder if ACLs were required. If ACLs, likely it would not fit SC4 schedule
    - SRM/dCache: W.r.t. SRM first version implemented & has been tested. More work required at level of underlying storage system
    - DPM: VOMS integration being implemented (LCAS/LCMAPS) using a method that will ensure compatibility w/ gLite
      - will be done the same way as has been done for GridFTP, gateKeeper
      - first release planned for July

Enhancements required by Experiments

- ❹ Directory functions (with the exception of “mv”)
  - Developers agree that this is useful functionality – with one caveat
    - (Some) Developers need more input as to why experiments need “ls” Is it more of a bookkeeping issue? If yes, why required to bother SE?
    - “ls” – Directory listing can easily create severe problems
      - Sheer data volume (w/ n*100k files/directory)
      - If “cookie model” have to maintain state / consistency
  - Implementation Plans
    - CASTOR: Can be done – reservations w.r.t. to “ls” (STBN)
    - SRM/dCache: In the process of implementing them
    - DPM: all implemented (“ls” supports single level only – no recursive lookup)

Enhancements required by Experiments

- Global Space Reservation
  - File based reservation part of the implementation since beginning
  - Allows a user to reserve space in advance
  - Reservation has a lifetime
    - Operational issue when used by average user who could interfere w/ the rest of community
  - SRM v2.1 Interface definition vague
    - No granularity (i.e. file size) attached to request
  - Developers view: Not useful in HSM environment with MSS
    - Useful in Disk-only environments (i.e. Tier2 Centers)
  - Implementation Plans
    - CASTOR: Could be implemented (STBN)
    - SRM/dCache: Working on implementation design according to use cases, prototype coded for SC2003
      - Static vs dynamic space allocation
      - Aiming for first implementation late Fall 2005
    - DPM: Implementation w/ limited functionalitty could be made available by start of SC4

Summary

- Status of Implementation of requested functionality varies
  - DPM most complete today
  - SRM/dCache and CASTOR developers are already working on some items or considering implementing them soon (priorities, schedules to be negotiated)
  - Likely not everything in all SEs available by Jan. 06
  - See no showstopper
- Good Communication needed …