Data & Storage Management TEGs
Summary of recommendations

Wahid Bhimji, Brian Bockelman,
Daniele Bonacorsi, Dirk Duellmann

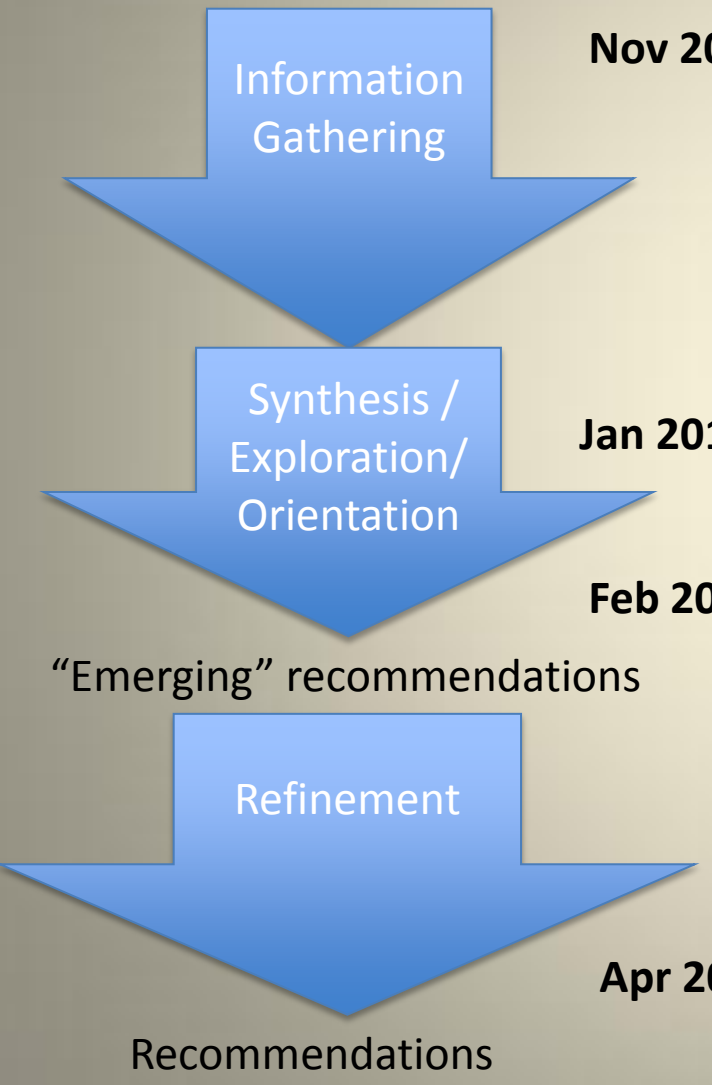
GDB, CERN

18th April 2012

WLCG Technical Evolution Groups (see also John's talk)

- "Reassess the implementation of the grid infrastructures that we use in the light of the experience with LHC data, and technology evolution…."
- Achieving **commonalities between experiments where possible, etc. etc.**
- Several groups – most relevant here are
  - Data Management (chairs: Brian Bockelman, Dirk Duellmann)
  - Storage Management (chairs Wahid, Daniele Bonacorsi)

Process



- Initial questionnaire
- Defined topics [TopicsDataStorageTEG]
- Soon Data / Storage TEG merged really..
- Questions to experiments:
  Experiment Presentations and Twikis
  [ALICE; ATLAS;CMS;LHCb]

- Storage Middleware presentations: 165687

- Face-to-face session for each topic
plus broader discussions.

Developed :

- Layer Diagram: Overarching picture
- Recommendations under each topic

See:

Final and draft report

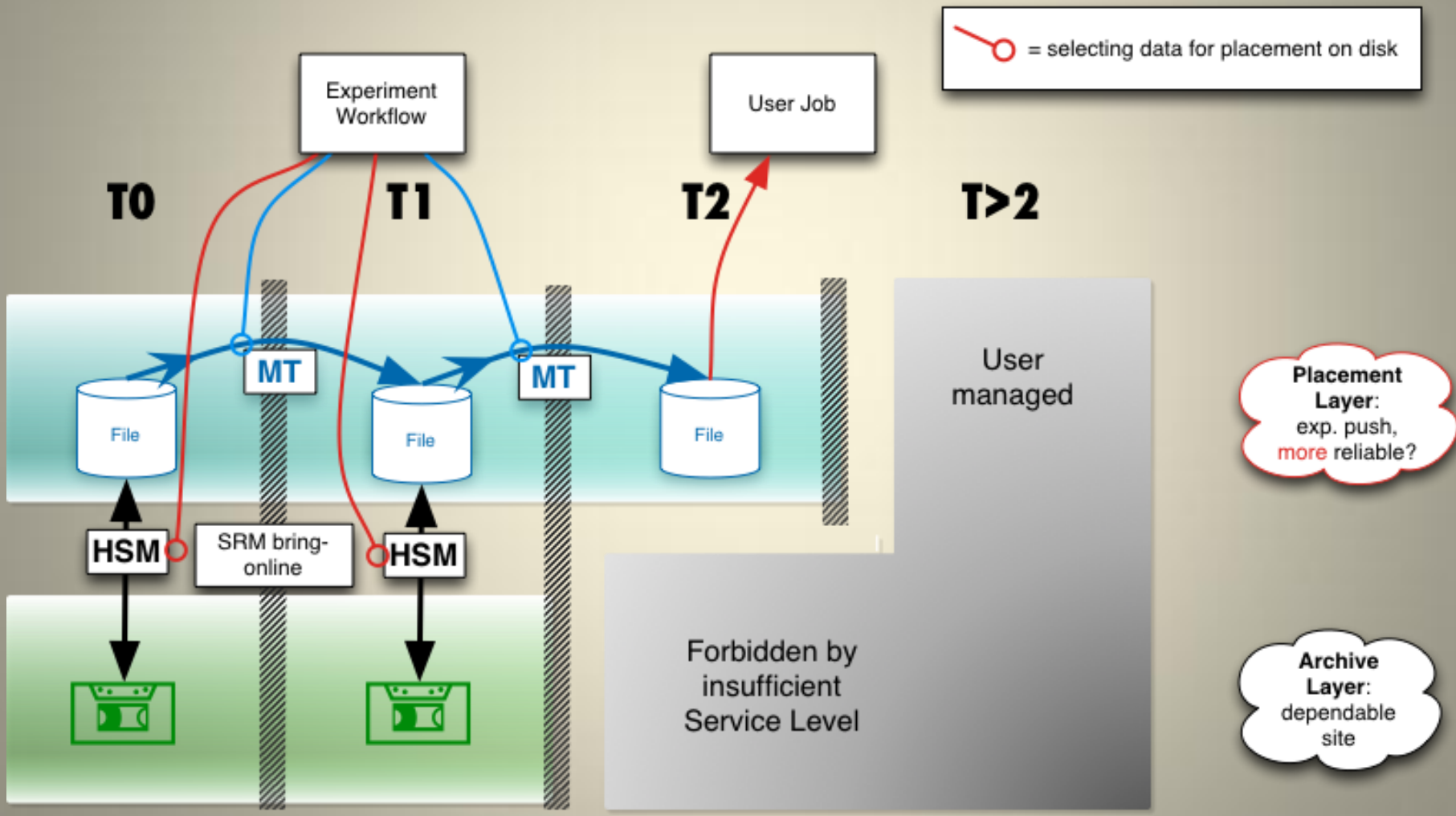
Layer Diagram

Data <-> Storage = Experiment <-> Site ??

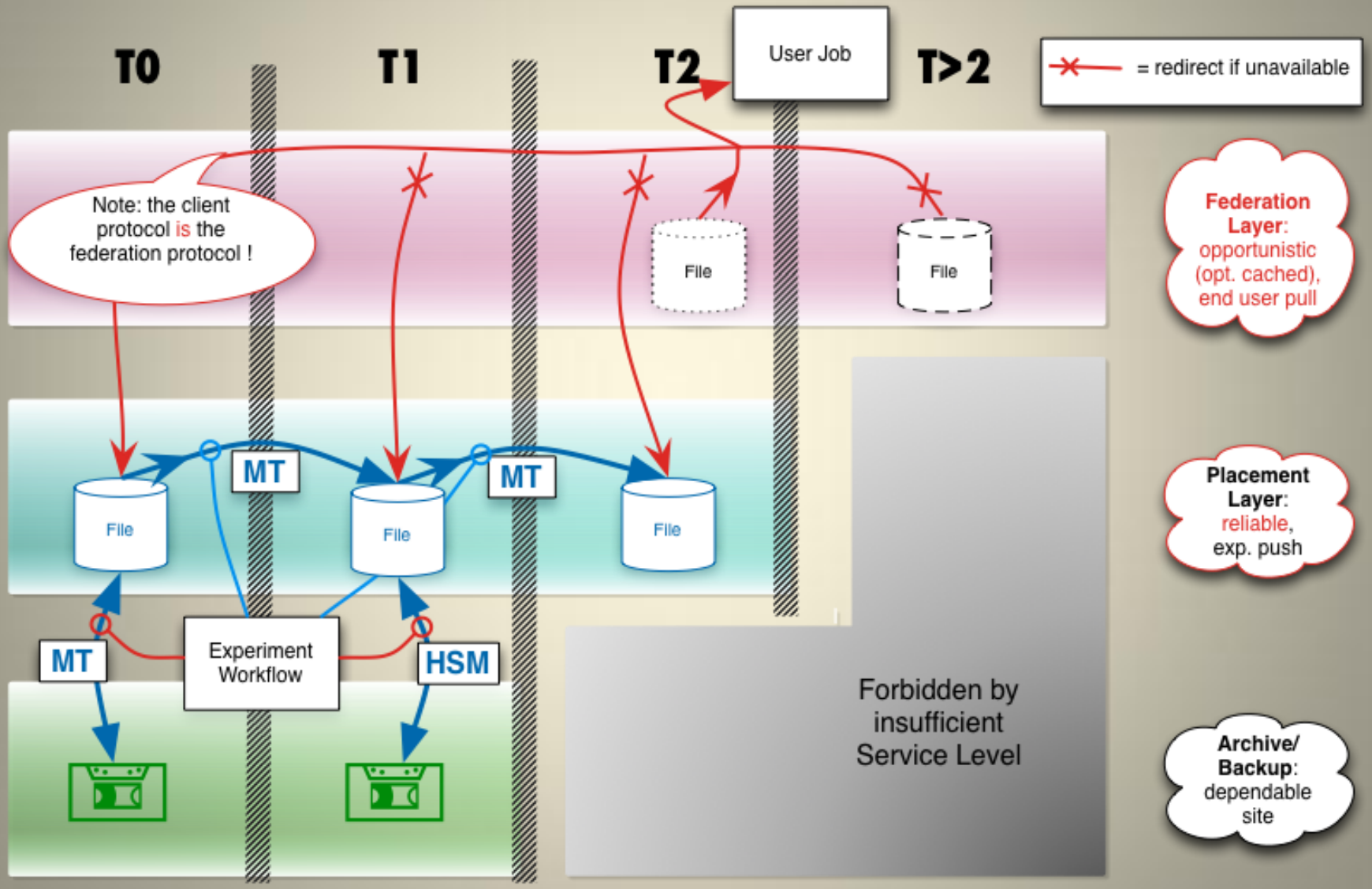
Certainly not the case now:….

Layer diagram to map out architecture and responsibilities

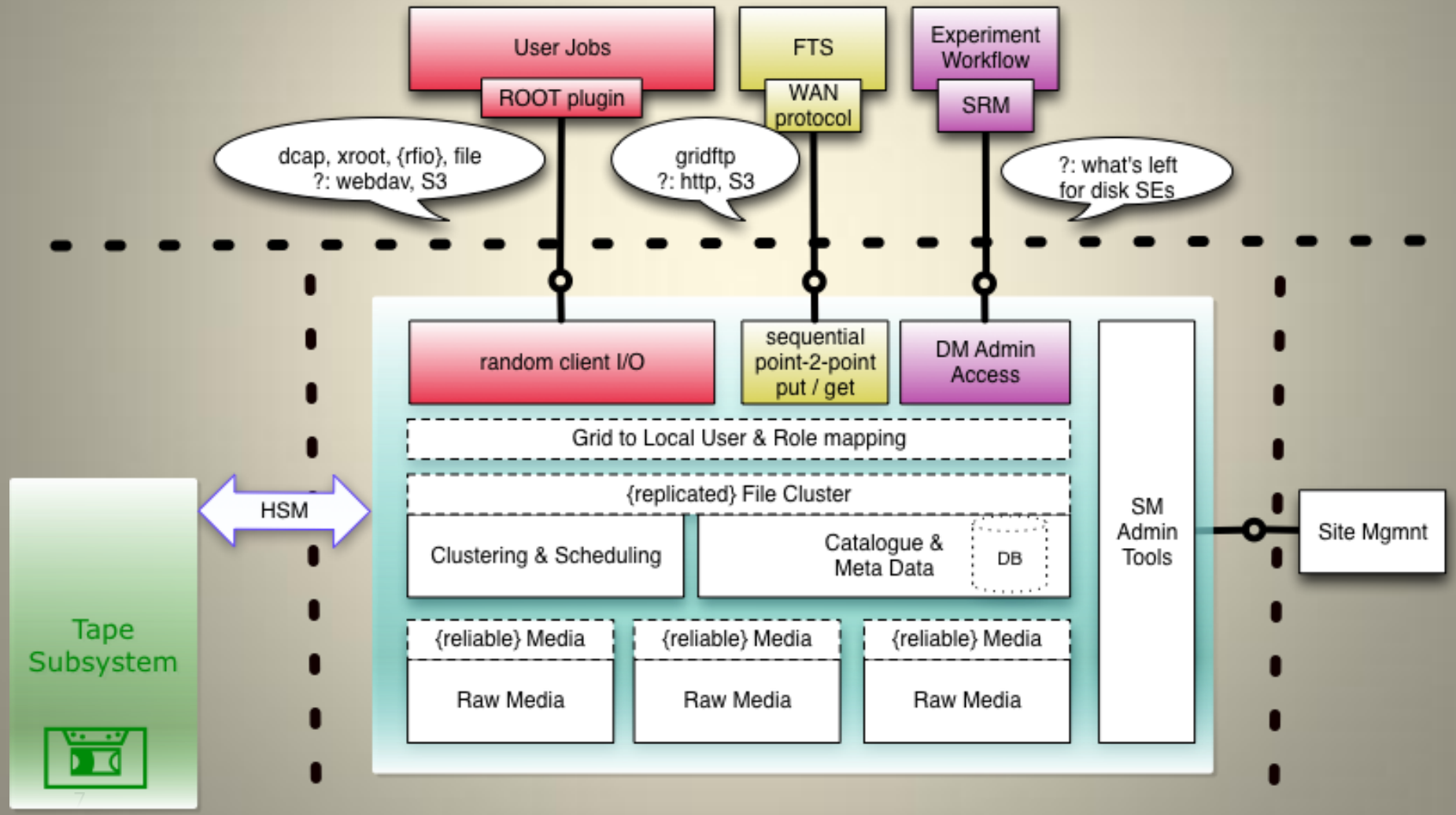
Data Placement



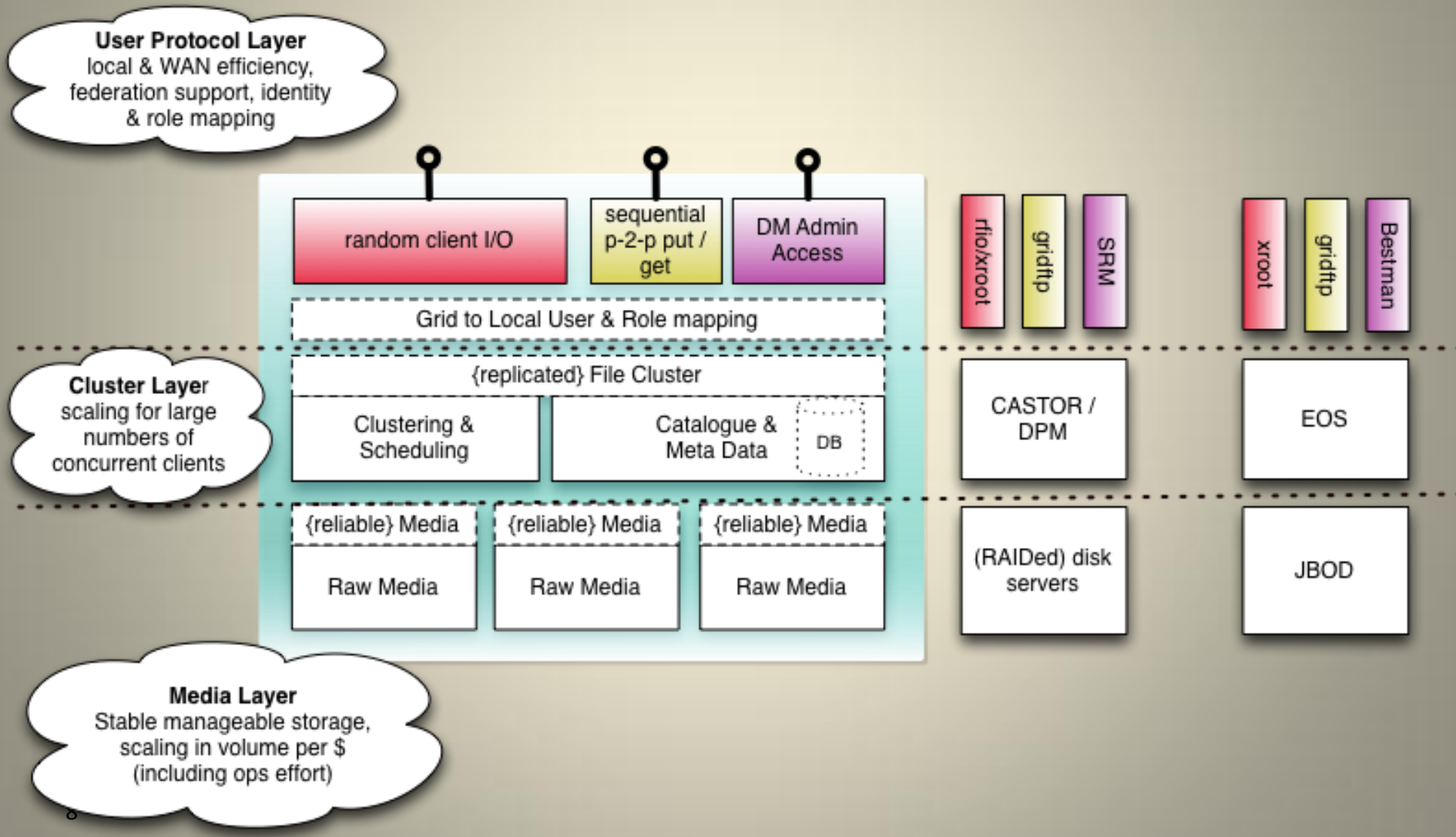
Placement with Federation



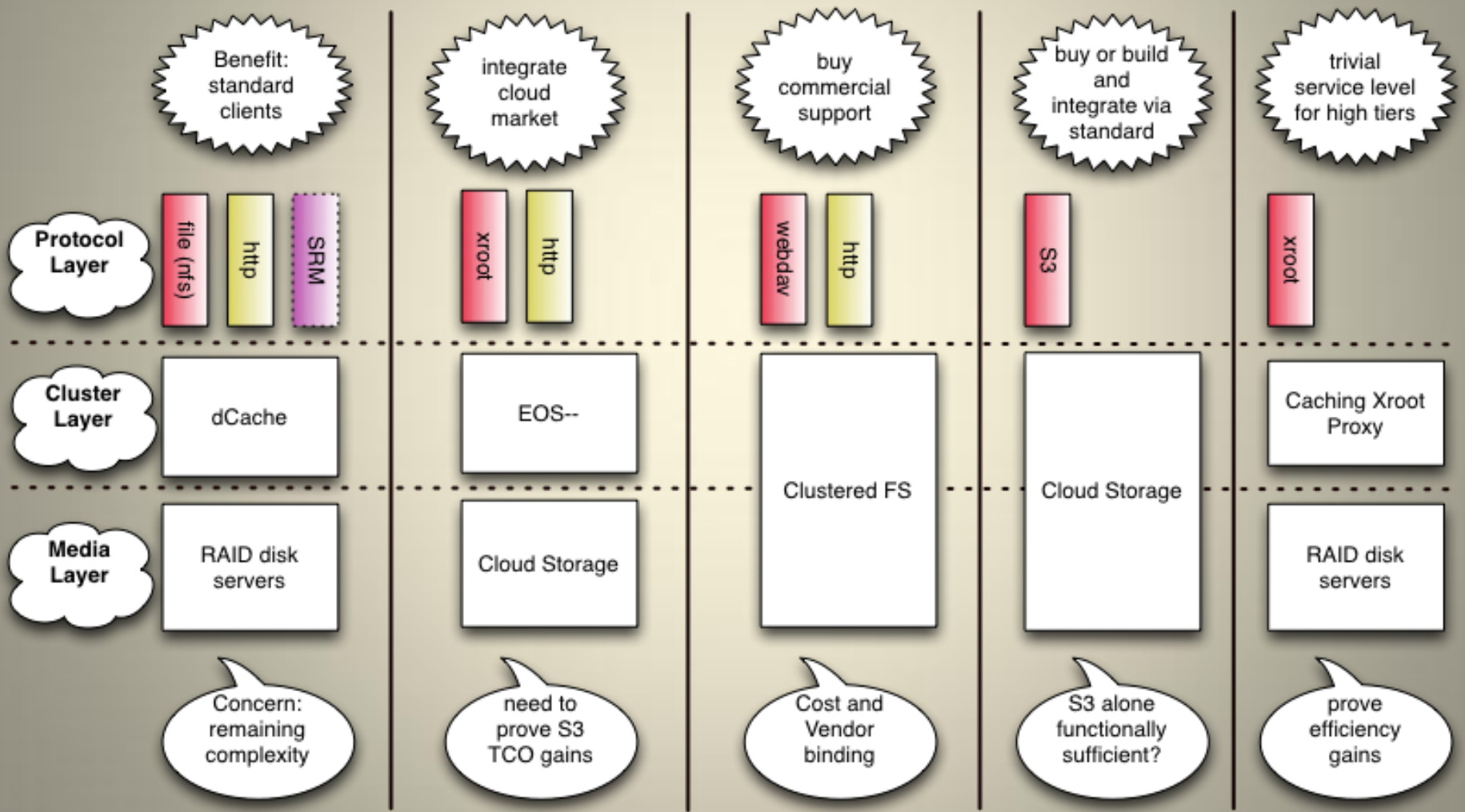
Storage Element Components



Examples of current SE's



Examples of (possible) future SE's



Recommendations and Observations

Placement & Federations

- Current option is only xrootd
  - Activity in http that should be supported (e.g. DPM)
  - (NFS 4.1 possible but not near happening for this)
  - R1: HTTP plugin to xrootd
- Activity in ALICE ; CMS; ATLAS
- All anticipate < 10 % of traffic this way
  - R2: Monitoring of federation network bandwidth
- Breakdown of what features experiments expect.
  - R3: Topical working groups on open questions

Topical Working Groups

- Launch working groups to follow up a list of technical topics in the context of the GDB:
  - Detailing the process of publishing new data into the read-only placement layer
  - Investigating a more strict separation of read-only and read-write data for increased scaling, stricter consistency guarantees and possibly definition of pure read-only cache implementations with reduced service levels (i.e. relevance for higher Tier sites).
  - Feasibility of moving a significant fraction of the current (read-only) data to world readable access avoiding the protocol overhead of fully authenticated protocols (assuming auditing to protect against denial of service attacks).
  - Investigating federation as repair mechanism of placed data; questions to answers would be:
    - Who initiates repair? Which inter-site trust relationship needs to be in place? How proactive is this repair? (e.g. regular site checksum scans or repair & redirect after checksum mismatch) How is the space accounting done? How do we address the repair of missing metadata?

Point-to-point Protocols

- GridFTP is ubiquitous and must be supported in medium term

  R4: gridFTP: use recent versions; exploit session reuse.

- Xrootd is currently used alternative:

  R5: Ensure xrootd well supported on all systems

- HTTP again a serious option (DPM<->dCache tests)

  R6: HTTP: continue tests; explore at scale

Managed Transfer (FTS)

- FTS is the only tool and used for more than transfer
  - Though experiments will go their own way if need be

R7: Update FTS3 workplan to include use of replicas; http transfers; staging from archive

R8: Cross-experiment test of FTS3 features

Management of Catalogues and Namespaces

R9: LFC not needed (by LHC) in med-term:

- Could be repurposed and useful tools (e.g. for consistency checking) should work with other catalogues

Also : Storage system quotas not needed (handled by experiment)

Separation of archives and disk pools/caches

- All experiments will split archive (tape) and cache (disk pools):
  - Atlas; LHCb; Alice already: CMS plan for this year

  R10: “HSM” still to be supported to manage disk buffer.
- A large separate disk pool managed through transfer offers advantages:
  - Performance: Lots of spindles.
  - Practicality: Need not be at same site.

  R11: FTS should support staging (see R7); Experiment workflows should support this transfer model

Storage Interfaces: SRM and Clouds

SRM:

- Ubiquitous;
- Needed in short-term buried in exp. frameworks;
- Practical advantages from common layer

BUT:

- Not all functions needed/implemented;
- Performance concerns;
- Industry not using (and developing alternatives);
- Experiment frameworks adapting for alternatives.

SRM: Looked at each functional component:

Which used: (see big table in report for details)

Functional Group	Usage Observation
Storage Capacity Management	For Space Management: Only space querying used (LHCb; ATLAS) (not dynamic reservation, moving between spaces etc.)
File Locality Management	For Service Classes: on medium term, spacetokens could be replaced by namespace endpoints (no orthogonality required)
	For Archives: bringOnline (and pinning) needed – no replacement.
Transfer protocol negotiation	Data access interface (get tURL from SURL): needed by LHCb: Alternatives exist: e.g algorithms or rule-based lookup
	Load balancing and backpressure: Needed but alternatives exist (and backpressure not imp. in SRM)
Transfer and Namespace	FTS and lcg-utils at least should support alternatives

Looked at alternatives:

Some used by WLCG currently (GridFTP ; xrootd)

Some in industry (S3; WebDav; CDMI)

Mapped to functions: (see big table in report for details)

Storage Interfaces: Recommendations

R12: Archive sites: maintain SRM as there's no replacement

- Non-archive no alternative yet for everything:
  - But experiments already looking at integrating

  R13: Working group should evaluate suitability targeting subset of used functions identified in report
    - Ensuring alternatives are scalable and supportable
    - must be supported by FTS and lcg_utils for interoperability

  R14: Working group should monitor and evaluate emerging developments in storage interfaces (e.g. Clouds) so experiments work together on long term solutions.

Storage Performance:

(Experiment I/O usage, LAN protocols, evolution of storage)

R15: Benchmarking and I/O requirement gathering

Develop benchmarks; Experiments forecast bandwidth IOPS and bandwidth needs; storage supports measurement of these.

R16: Protocol support and evolution

Experiments can use anything ROOT supports

But move towards fewer protocols and direct access supported.

ROOT; http direct access; and NFS4.1 should be developed

R17: I/O error management and resilience

Explicitly determine storage error types and ensure application handling

R18: Storage technology review

Incorporating vendors; spreading information between sites.

R19: High-throughput computing research

Not restricted to current data formats (ROOT);

Hadoop style processing or NextBigThing™

Storage Operations:

Site-run services: monitoring; accounting etc

R20: Site involvement in protocol and requirement evolution:

ie. site representatives on storage interface working group to ensure proposals are manageable by them

R21: Expectations on data availability. Handling of data losses

Experiments should state data loss expectations (in MoU) and reduce dependence on “cache” data.

Common site policies for data handling (examples in report)

R22: Improved activity monitoring:

Both popularity and access patterns

R23: Storage accounting

Support StAR accounting record

POOL Persistency

- Recently LHCb moved, so now ATLAS specific sw.
- Atlas also plan a move so:

R24: POOL development not required in medium term

Security

Separate document with Security TEG.

Areas that need attention in the near term:

R25: Removal of “backdoors” from CASTOR

R26: Checks of the actual permissions implemented by Storage Elements.

R27: Tackling the issues with data ownership listed in document (e.g. ex. VO members; files owned by VO rather than individual)