

Distributed Tier1 scenarios

G. Donvito --- INFN-BARI

On behalf of the Distributed Storage Group

Outline

- * Why we need a distributed Tier1?
- * Pre-requisite and assumption
- * Few models:
 - * description and features
- * Few guidelines and suggestions
- * Timeline proposal

Why we need a distributed Tier1?

- * We have a grant from the MIUR in order to build a distributed computing facility for SuperB
- * This will give us the possibility to build on 4 sites a quite big farms strictly devoted to fulfills the SuperB requirements

Starting points

- We will have 3 (+1 smaller) centers in South Italy founded for SuperB
- We are building ~ “from scratch” so we can easily drive the technical solution
- The network among those site should be the most advanced that will be possible given the network technology
 - >10Gbps
- None of those centers already has tape libraries
- We will surely have other sites involved in SuperB computing
 - distributed world wide
 - With greater network latency

Open Questions

- T0 site is out of these 4 sites?
 - R: some of the proposed scenario fit with a T0 within these site, some other not
 - Could we build a lightweight T0 out of those 3-4 site with only reliable disk buffer and a good 40Gbps network connection?
- T1 should also provide “CAF” (something like an Express Analysis Facility)?
 - R: All the proposed scenario could address this problem, but it is important to know the answer from the beginning
- The data custodiality is a duty only for T0 or for T1 too?
 - It is strictly required to do “custodiality” with tapes at T1?
 - Is there the room to host tape library in each site?
 - We need to investigate on this item
- Is it foreseen to have other Tier1 in other country?
 - There will be a “full replication” of the data, or only a fraction of those?

Possible layouts

1. Split Data
 - a. Different datasets in different site
2. Replicated Data
 - a. Automatically replicated (with available sw)
 - b. Experiments tool driven (HEP community developed sw)
3. Split Features
 - a. T0, T1, CAF, etc

Split Data - Description

- * LHC model: already in production
- * The experiment split and associate data to each of the sites
 - * The association could be driven by physics requirements (community interest) or by computing requirements (size of datasets, processing time, etc)
- * All the sites are identical in terms of service
 - * Could be different in terms of size
- * Each site should run all the steps of the experiment workflow

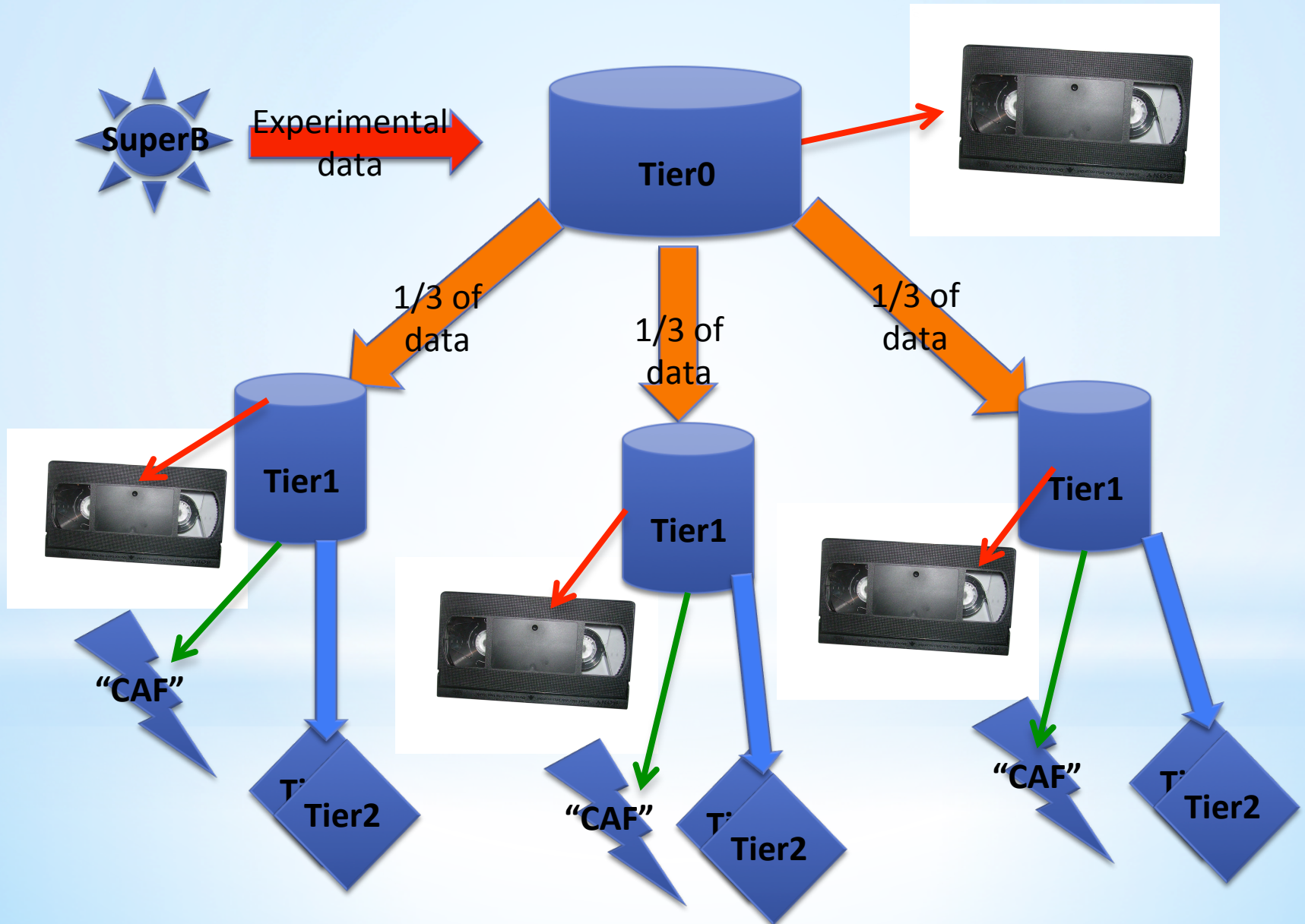
Split Data - Characteristics

- Each site could choose its own hw/sw solution
 - If the solution is “blessed by the experiment” (i.e.: there are the needed tools to exploit the solution)
- This model is easy to develop: no need to understand how/if it works
 - LHC is testing it deeply
- If a site is down, a not negligible fraction of the data is unavailable
- Each site should have (if required from computing model) the duty for its own data custodiality
 - This need a tape system on each site or (something similar)
 - This is costly!
- This will require a good data movement tool to move data from T0 to the T1
 - I guess we always need this

Split Data – Characteristics (2)

- *The network latency (and bandwidth) among sites it is not a problem
- *The data on each site should be accessed by jobs sent on the same site
 - *Remote access is surely less efficient
- *If we need a “CAF” facility in each Tier1 we need to arrange it somehow
 - *i.e.: Using a dedicated facility (storage, batch configuration, etc)
- *This model perfectly fits the cooperation with other(s) Tier1 in different country

* Split Data



Split Data – Technological option

- * CNAF model:
 - * GPFS + TSM (or **Lustre + HPSS**)
- * CERN model:
 - * **EOS + MSS** (staging has to be done somehow “manually”)
- * dCache infrastructure
- * Standard “Scalla” installation

Replicated Data – Description

- All the critical data are replicated in each site
 - We could assure “custodiality” without using tapes?
- Each site should have enough disk space to store all the critical data for the experiment
 - The less critical data, could have less than 3 copies
- All the sites are identical in terms of service
 - There could be a small difference in terms of size
- Each site could run all the steps of the experiment workflow
 - The job could be submitted where there are CPU available
 - Job scheduling is not data driven as each site has the data

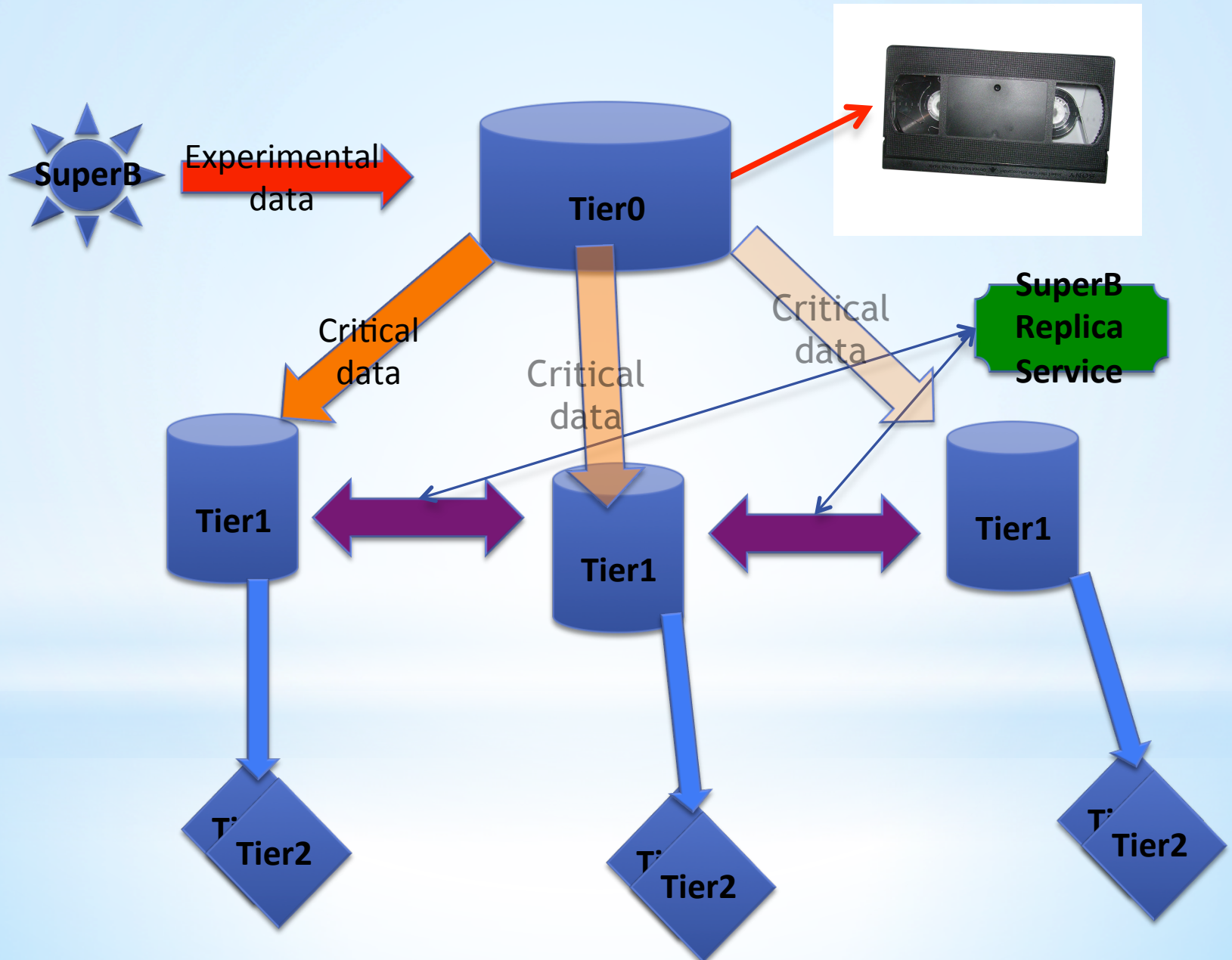
Replicated Data - Characteristics

- If a site is down (or overloaded), this do not affect at all the experiment community
- We can avoid having “high costly” and “difficult to manage” tape infrastructure
- This scenario will work far better if we can provide a good network bandwidth among sites
- It is important to understand how to keep in sync the 3 sites
 - There are basically 2 option:
 - “Experiment made” solution
 - Public available solution
- We need to understand how this fits with TCO of the storage solution (power and disks)
- Each site could be “disk only”
 - In principle there is no need to have a separate solution to implement a “CAF”

Replicated Data (exp made solution) - Characteristics

- * Each site could choose its own hw/sw implementation
- * The replication tool should be:
 - * resilient and well tested
 - * A failure in this system could cause a data loss
 - * Lightweight for the storage system
 - * The storage system could not be overloaded by “routine activities”
 - * Able to automatically deal with disk (and file-system) failure at each site
 - * You need also some cksum features

Replicated Data (exp made solution)



Replicated Data (exp made solution) - Technological option

- * CNAF model:

- * GPFS + TSM (or **Lustre + HPSS**)

- * CERN model:

- * **EOS + MSS** (staging has to be done somehow “manually”)

- * dCache infrastructure

- * Standard “Scalla” installation

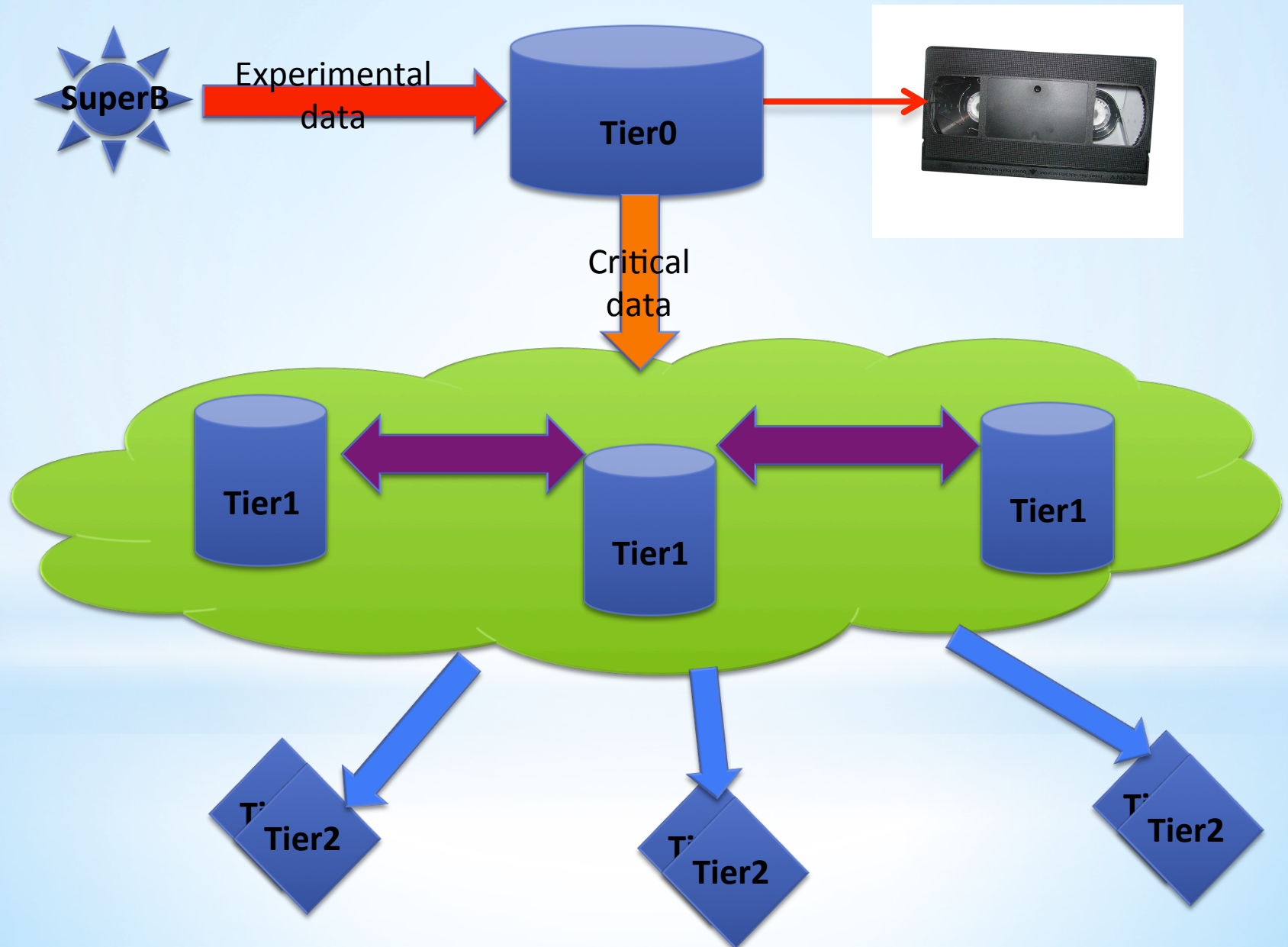
Replicated Data (public available solution) - Characteristics

- No need to maintain it
 - Widely tested and maybe less buggy!
 - If it is an open source solution we could only adapt to our specific needs
- The solution means having a single instance of the storage system distributed among the sites
 - A job scheduled in the site “A”, could both read data from “A” or from “B” in a transparent way
 - There is the need to replicate also metadata
 - For failover purposes
- Those solutions are always automatically replicating missing files
 - A job will never fail for a disk failure

Replicated Data (public available solution) - Characteristics

- It could be realized with a “low-cost” distributed disk-only hw solution
 - DAS or WN disks
- This solution could fit very well with a CDN: where other smaller site could be simple volatile disk cache
 - If we have another Tier1, this will easily be “yet another stream” going out from the Tier0
 - This will put far less load on the experiment DMS
- It is required that all the sites choose the same hw/sw solution

Replicated Data (public available solution)



Replicated Data (public available solution) - Technological option

- * Posix FileSystem:

- * GPFS (3 Data & Metadata Replicas)

- * Xrootd Based:

- * **EOS**

- * General Solution:

- * **Hadoop FS**

- * dCache infrastructure ??

- * Depends on the availability and reliability of the “ReplicaManager” feature

Split Features - Description

- Each site is specialized to do specific task
 - T0, T1, CAF etc
- Each site could have all the resource needed to do a specific “task” for the collaboration
- Both T0 and T1 should have a tape archive
 - CAF could be disk only
- An “experiment made” tool takes care of moving (staging) data among sites
- Each site should run only well defined steps of the experiment workflow
 - The job submission/match making is “definitely simple”

Split Features - Characteristics

- * Each site is free to choose hw/sw solution
- * If a site goes down at least one step of the chain for all the experiment is blocked
- * Each site has a dedicated infrastructure focused and build to do a specified task
 - * This will mean that is easy to reach a good efficiency
- * The network bandwidth and latency between sites is not a big problem

Split Features - Technological option

- * T0 / T1:

- * CNAF model:

- * GPFS + TSM (or **Lustre + HPSS**)

- * CERN model:

- * **EOS + CASTOR** (staging has to be done somehow “manually”)

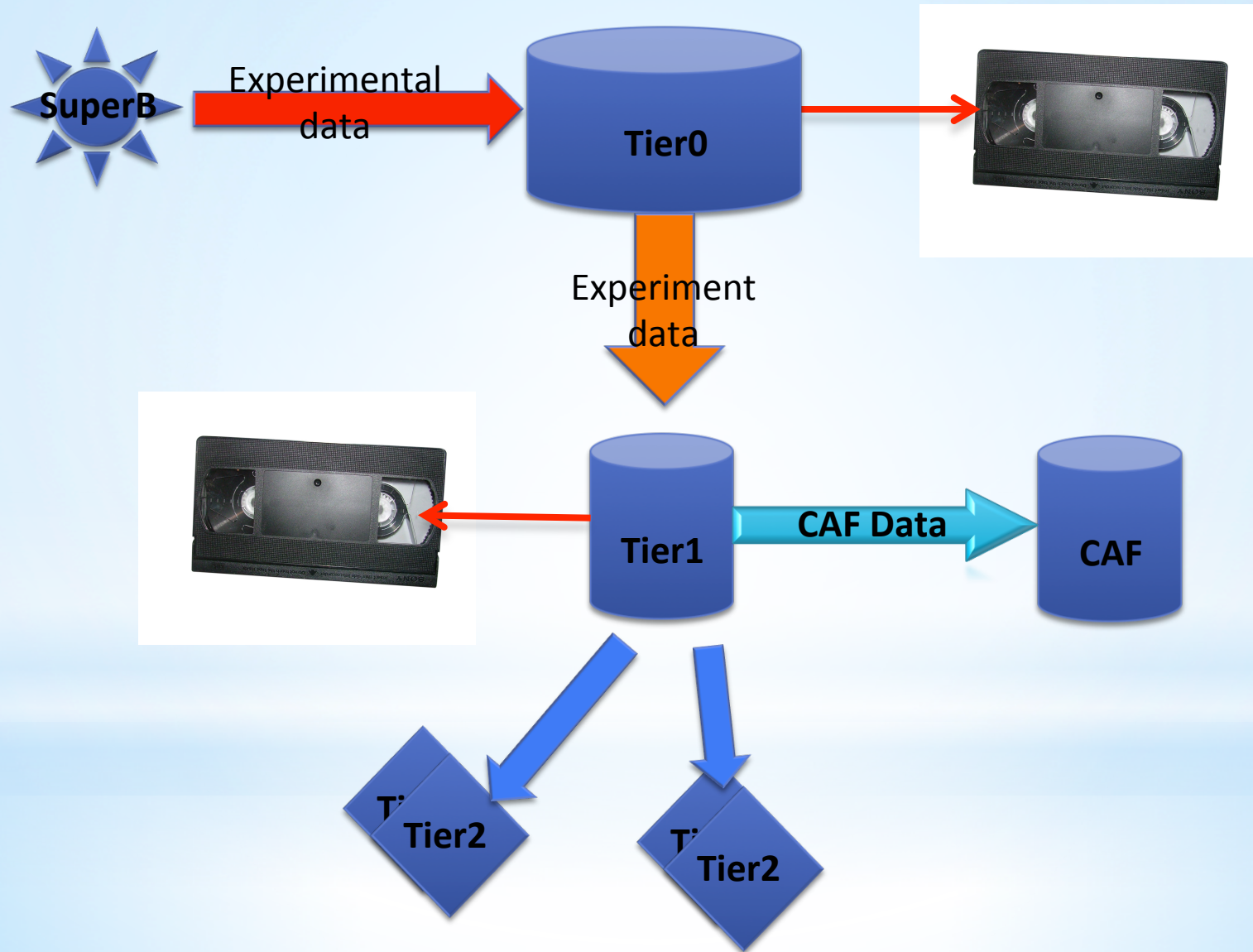
- * dCache infrastructure

- * CAF:

- * Lustre

- * EOS

Split Features



List of service

- *Data archival
- *Skimming
- *CAF
- *MC
- *Chaotic analysis ?
- *??

Summary

* Split Data:

- * Standard LHC model each: each T1 responsible for a fraction of data custodiality (a tape library in each site)
- * A site down -> a fraction of data *could* not be available

* Replicated Data:

- * No need to have tape: we have 3 copies of data on cheaper disks in each site
- * If a sites goes down the other can be used => no service disruption
- * At least two different solution here, but we should prefer “public-available solutions”

* Slit features:

- * Each site has a specific “tasks” to execute (Archive, skimming, CAF, etc)
- * Only one site need tape archive
- * If a site goes down a “function” for the experiment is stopped

* The user experience should be transparent to the layout implemented:

- * The gateway should take care of distribute jobs thanking care of the computing/ storage infrastructure and the jobs requirements

Timeline proposal

- End 2011 - Jan 2012
 - Find an agreement within the collaboration on the “Open Questions”
- Jan 2012
 - Find an agreement within the collaboration on a prioritized list of scenario which we are interested
- Feb 2012 start with technologies evaluation
 - At least two different solution for each scenario
 - At least first two scenarios
 - This could be done in parallel in different sites
- April 2012 start with distributed testbed at least with 2 sites
 - The “most interesting” scenario and solution
- May 2012 third site joining
- June 2012 testing “back-up” solution in the distributed testbed
- July 2012 Report to the collaboration

PON
approved

Agreement
on Scenarios

Testing
technical
solution

Report on
results

Computing
TDR

2011

2012

2013