Challenging data and workload management in CMS Computing with network-aware systems

D.Bonacorsi, T.Wildish

How can we make better use of the network to improve analysis throughput?

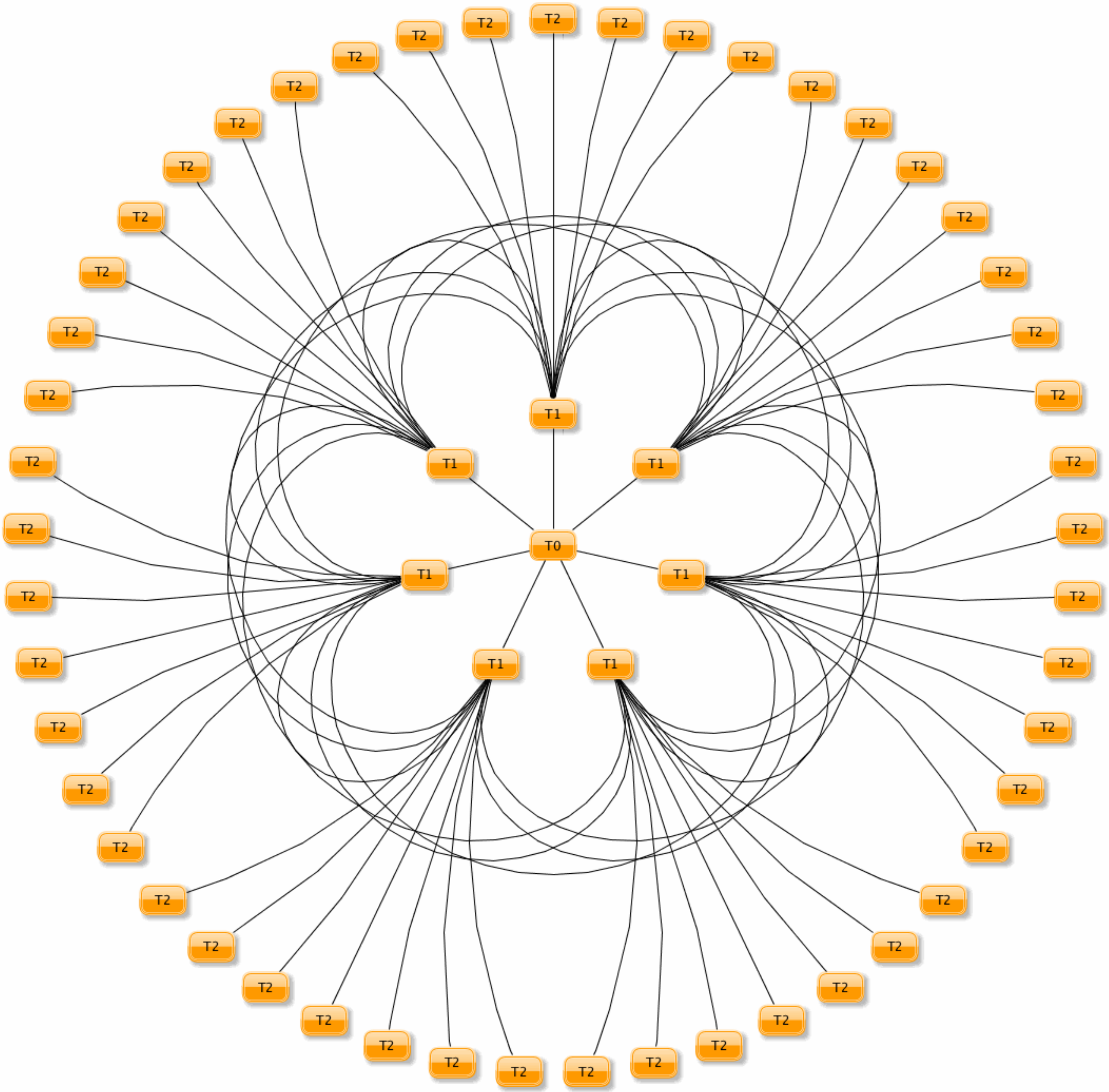
- A bit of history
- Present day reality
- Consequences of our reality
- Challenges (a.k.a problems)
- Solutions

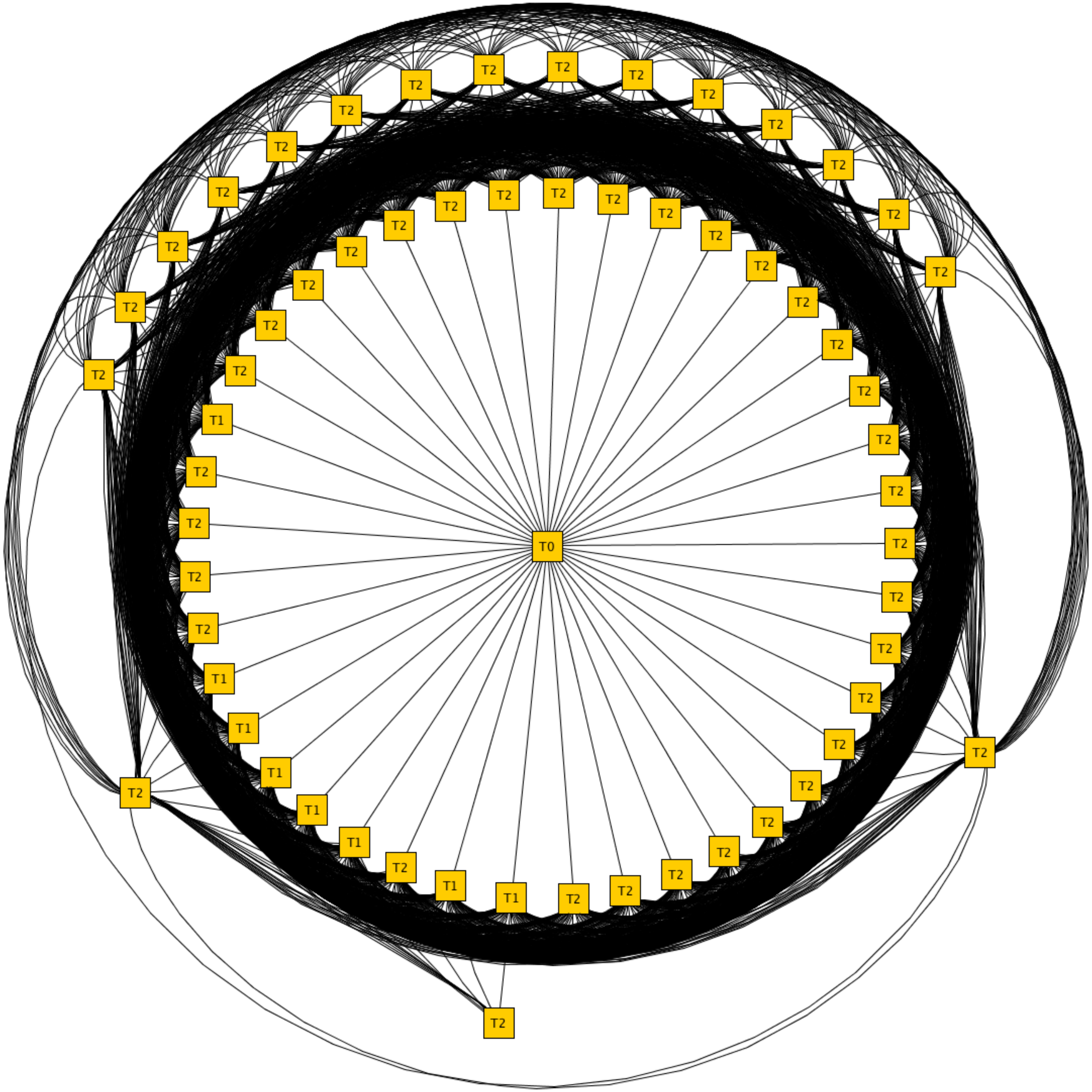
- PhEDEx is the CMS data-placement management tool[1] controls bulk data-flows
  - – Basic architecture is ten years old
  - – Routing and retry algorithms rather TCP-like
    - Keep network full, prefer to send stuff that flows easily
    - Rapid back-off, gentle retries
    - ‘Guaranteed’ delivery, but no concept of deadlines
  - – Prioritisation based on simple queue-ordering
  - – Slow to switch sources, assume good choices will remain, on average, good choices

[1] *The CMS Data Management System*, N.Magini

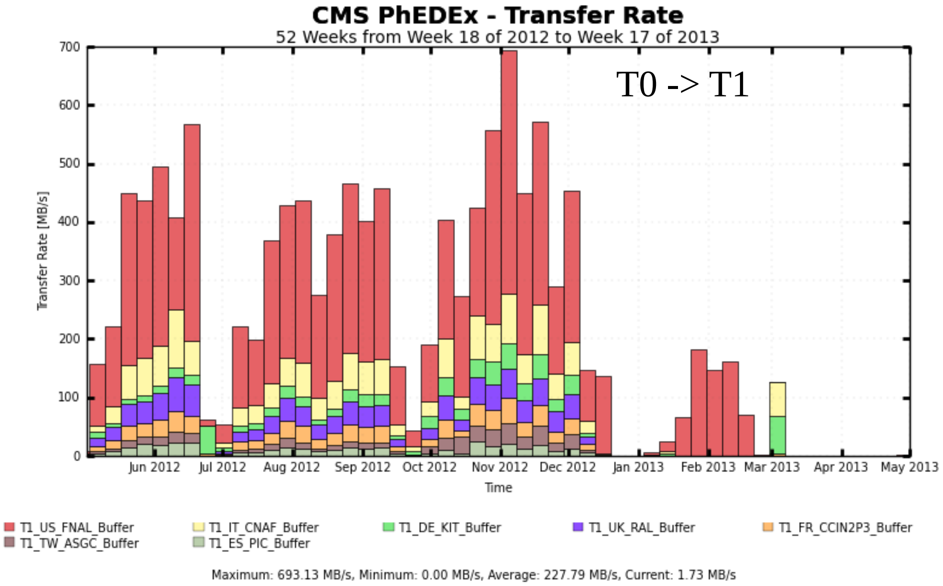
- Original computing model design decisions
  - – Did not trust the network
    - Assumed it would be the greatest source of failure
  - – Expected slow links, ~100 Mb/sec
    - Expect bandwidth to be a scarce resource
    - Optimise for efficient transfer of files, instead of datasets
  - – Move jobs to data, not data to jobs
  - – MONARC model
    - Strictly hierarchical, no T2->T2 or T2->T0
    - Minimal number of links to manage (~100)

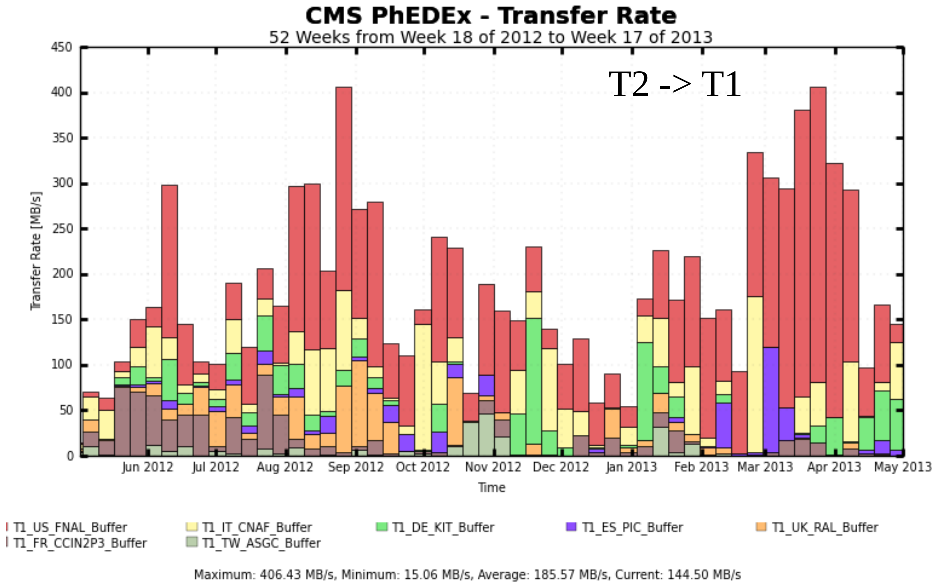
- Today
  - Network is most reliable component
    - Most errors at SE level or within a site, not between sites
  - 10 Gb/sec or more on many links
    - Bandwidth is abundant
    - We are far from saturating most links
  - Strict hierarchy relaxed, to say the least
    - Almost fully connected mesh
    - T2->T2 transfers significant

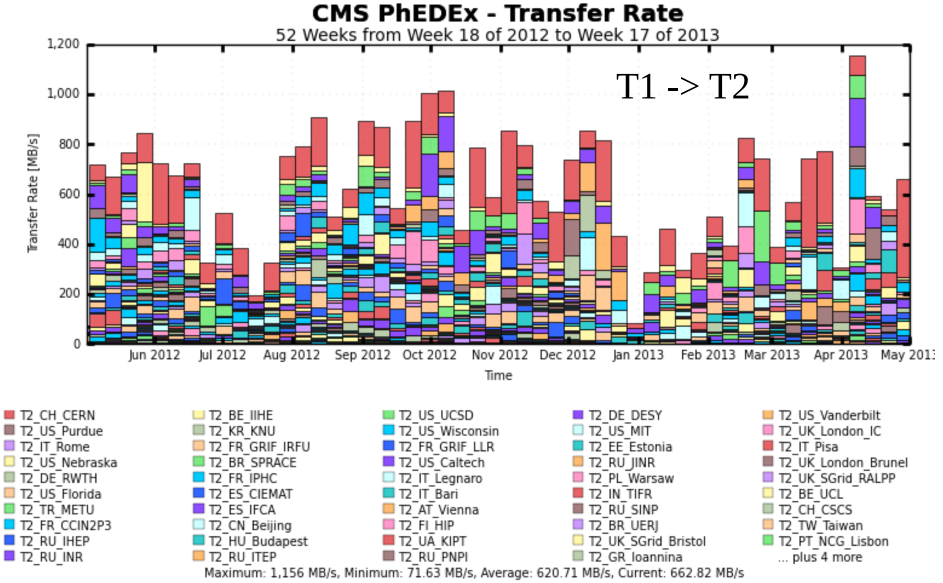

T2
T1
T0
T1
T2

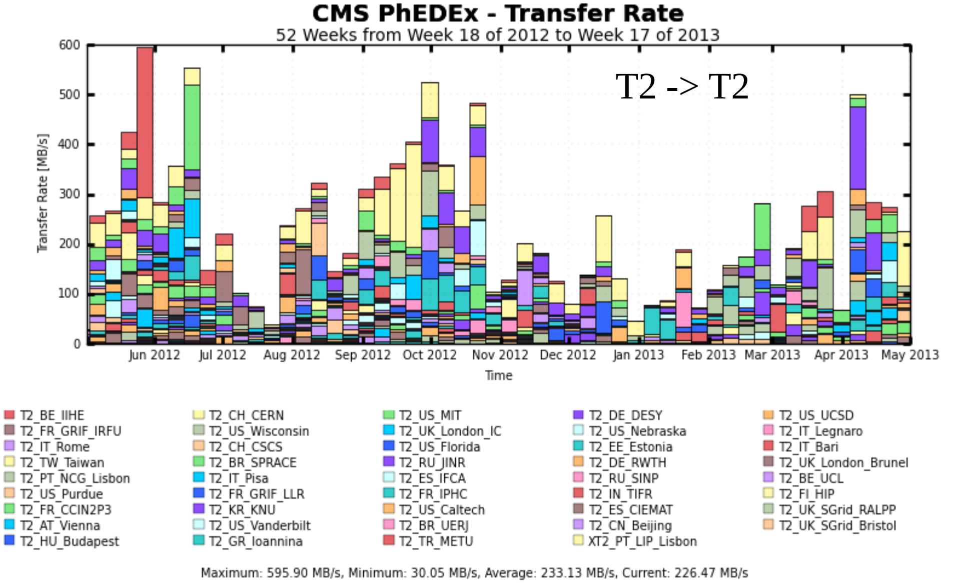

T0
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T1
T1
T2
T1
T1
T1
T1
T1
T2
T2
T2
T2
T1
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2
T2

Challenging CMS Computing with network-aware systems









Average rate last year	**Production**	**Debug**	**Total**
T0 -> T1	230 MB/sec	100 MB/sec	330 MB/sec
T2 -> T1	190	200	390
T1 -> T2	620	230	850
T2 -> T2	260	180	440
Total	1300	710	2010

Production instance is real data
Debug instance is for commissioning and link-tests
- 1/3 of total traffic is for knowledge of network state

Average rate ~ 2 GB/sec CMS-wide
- sustained over last 3 years.
- not b/w limited

- PhEDEx routing and retry algorithms do not make optimal use of network
  - Back-off and retry same link in case of failure, could fail-over to another source node or to AAA
- MONARC model largely displaced [1]
  - Much traffic between T2's, T3's, without T1 intermediaries
  - We don't have a model of this
    - Don't know why all these movements are requested
    - Don't know which movements are correlated

[1] *CMS Computing Model Evolution*, C.Grandi

- Current metrics
  - Time to transfer a dataset
    - PhEDEx-internal statistics, based on past performance
  - Time to complete a set of batch jobs
  - Amount of data on a site
    - Group quotas, number of replicas
- These are all dataflow or workflow centric
  - Related to use of hardware, not analysis throughput
  - Not easy to model analysis throughput!

- So back to the question:

How do we make better use of the network to improve analysis throughput?

- Near-term improvements
  - AAA - Any data, Any time, Anywhere [1,2]
    - Analysis fallback to xrootd/WAN access on failure
  - Data-popularity service
    - Identify idle data, automatically clean them
    - Identify hot data, automatically replicate them

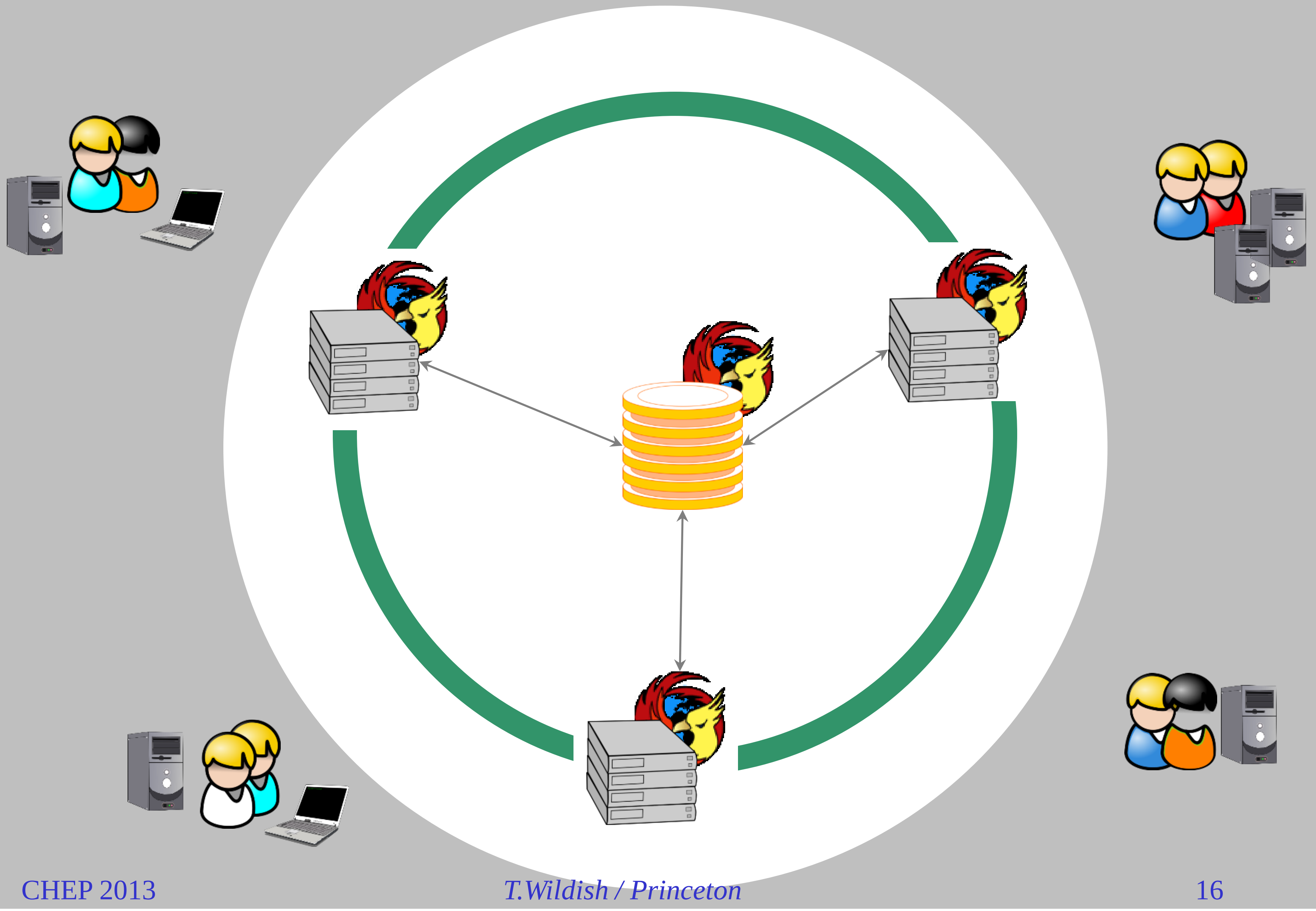
[1] *CMS use of a data-federation*, K.Bloom

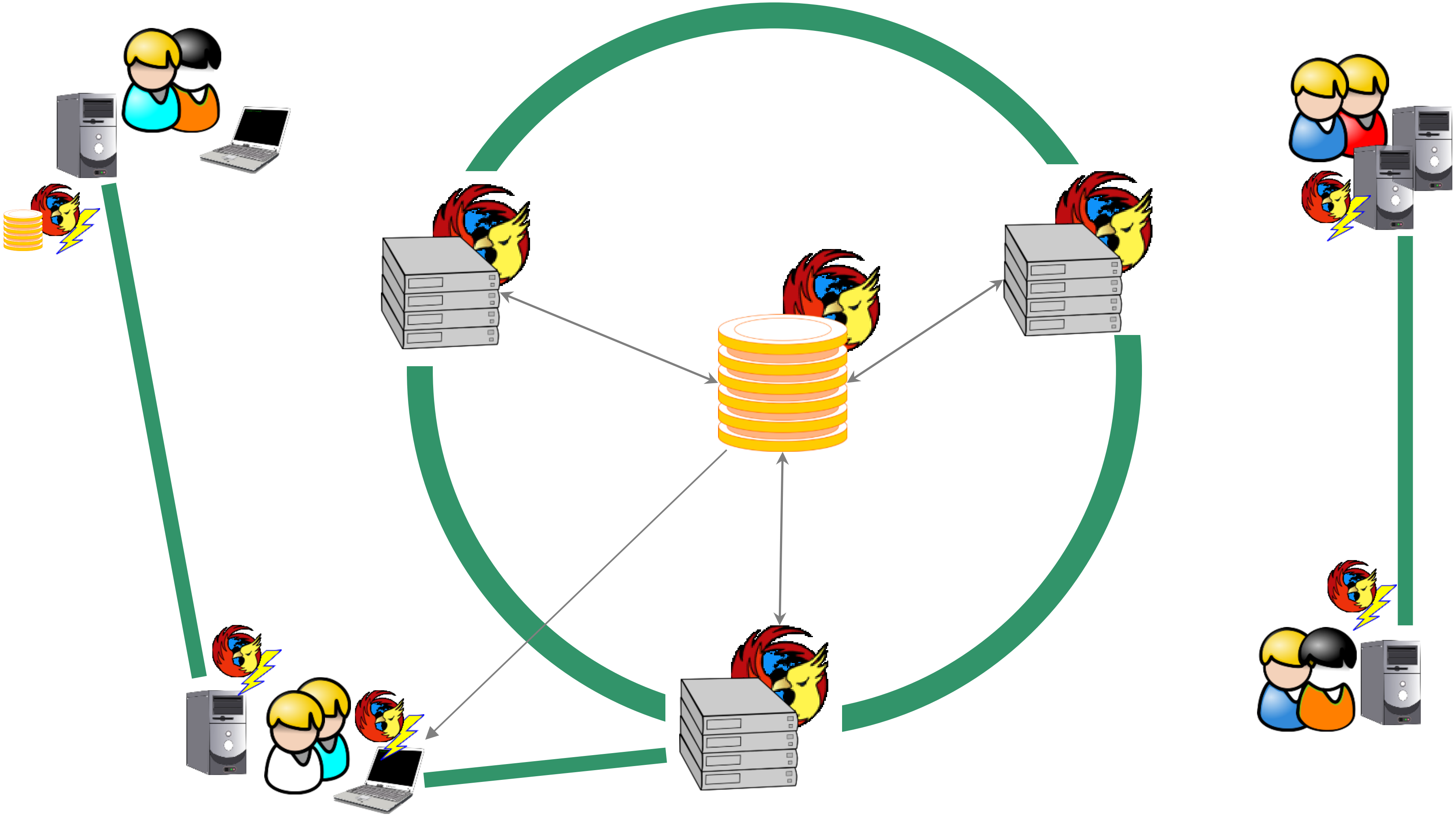
[2] *Xrootd, disk-based, caching-proxy for optimization of data-access, data-placement, and data-replication*, M.Tadel

- Longer term?
  - More dynamic and responsive data placement
  - Reduce latencies in PhEDEx (30 mins => 1 min)
    - Rapid switching of source replicas on failure
    - Emphasise block-completion over raw throughput
  - External sources of network-state (PerfSONAR, MonaLISA)
    - Awareness of overall network state, not just internal
  - Manage the network explicitly
    - Schedule use of bandwidth across PhEDEx, CMS, …
    - Reserve network resources, virtual circuits

- New possibilities
  - Improve knowledge of delivery-time for bulk data
    - Deterministic prediction of completion time
    - Co-schedule jobs with data placement
  - Respond to real network conditions, beyond the VO
  - JIT replication of datasets
    - Replicate with PhEDEx in response to traffic on AAA
    - Co-ordinate between PhEDEx & AAA on bandwidth use
  - Opportunistic storage/computing resources
    - Pump data in fast, use CPU, pump data out fast
    - Increases usability of non-CMS-controlled resources

- How?
  - Dual role of PhEDEx: data-store & data-movement
    - Refactor into CMS-managed data-store and a CDN
  - Data-store
    - Static repository of data. i.e. CMS-managed sites
    - Dynamic creation/deletion of sites, from SE’s to single machines, anything with a few TB of disk
    - Accept user-defined datasets, not just official data
  - CDN
    - Rapid movement of small volumes of data (~1TB)
    - Book network circuit for performance, determinism



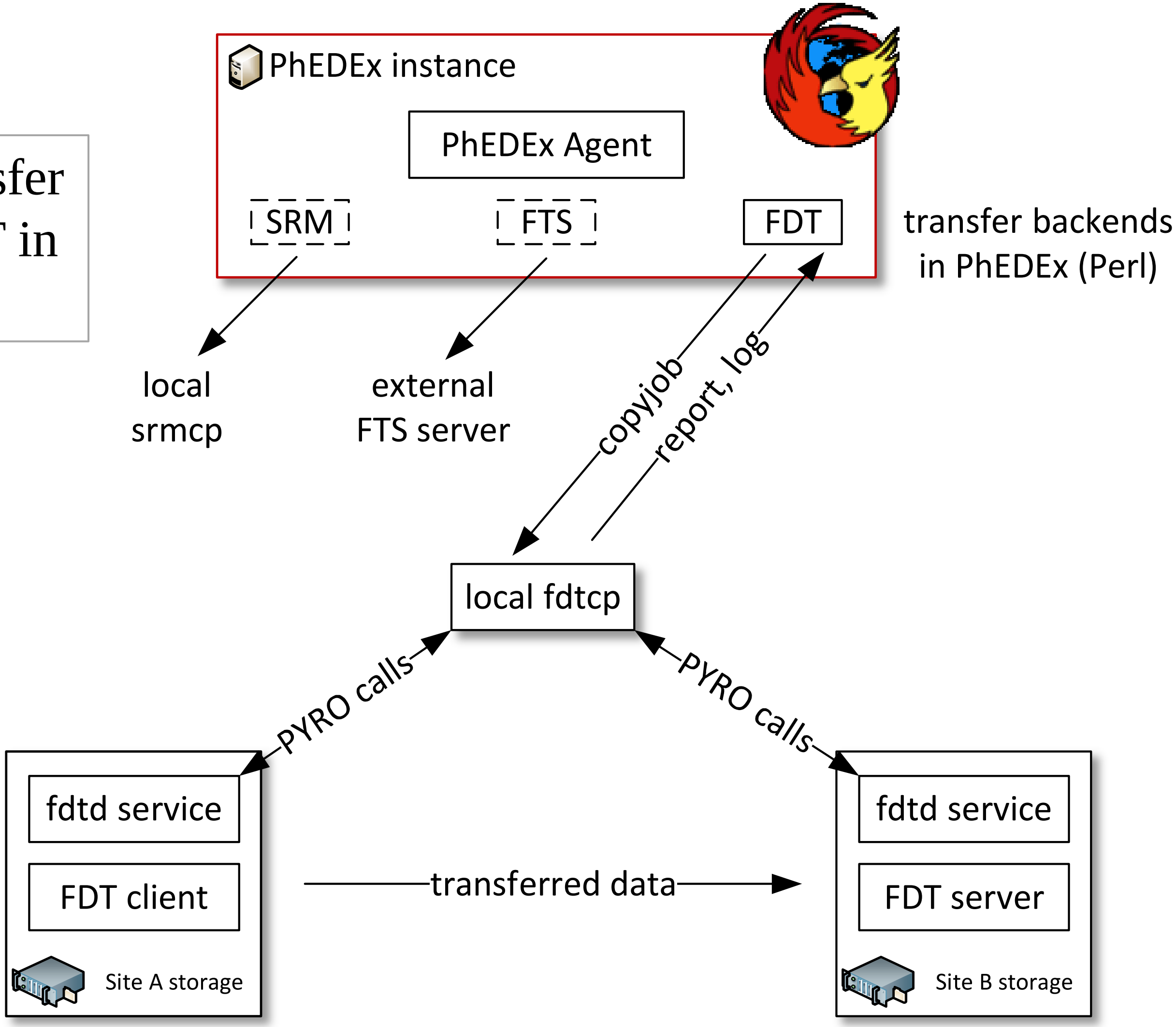


- Virtual circuits, PhEDEx and ANSE[1]
  - Several points of integration into PhEDEx
    - Per transfer-job: already exists with FDT
    - Per-destination, a circuit for all files from a given source
    - CMS-wide, book and manage circuits centrally
  - ANSE
    - Integrate network awareness into the software stacks of CMS (PhEDEx) - and ATLAS (PanDA)
    - Build on top of existing services (e.g. DYNES[2])
    - Part of LHCONE point-to-point demonstration

[1] *Integrating the Network into LHC Experiments: Update on the ANSE project*, A.Melo

[2] *Application Performance Evaluation and Recommendations for the DYNES*, S.McKee

Existing per-transfer circuits with FDT in in PhEDEx



- What’s missing?
  - – Network management in CMS
    - Network awareness/management in PhEDEx -> ANSE
    - Awareness and management for AAA or other activities?
    - Policy for use of circuits
      - – When is it worth/not worth using a circuit?
      - – API, concept of network budget, advance reservation…?
  - – Robust virtual circuits across the CMS VO
    - Multi-domain, many players, much work-in-progress[1]
  - – Integration, coordination of network/CPU/storage
    - Coherent use of network resources (paths, circuit duration, occupancy etc)

[1] *Advanced Networking for HEP, Research and Education in the LHC Era*, H.Newman

- Summary
  - A more responsive, deterministic, PhEDEx opens possibilities for CMS
    - Rapid replication can improve efficiency in many ways
    - Extend data-management to broader range of resources
  - Network awareness/control takes this to a new level
    - Knowing the network state allows smart routing decisions
    - Control of network allows greater flexibility and responsiveness, more determinism
    - Co-scheduling of CPU/network/storage
    - Lots of work to do, from the network fabric through the experiment applications to the management policies