Data ~ Mass

- hard to move: inertia
- tends to clump: fewer & larger aggregations with time
- needs to be preserved (classically - else energy is exchanged)

Outline

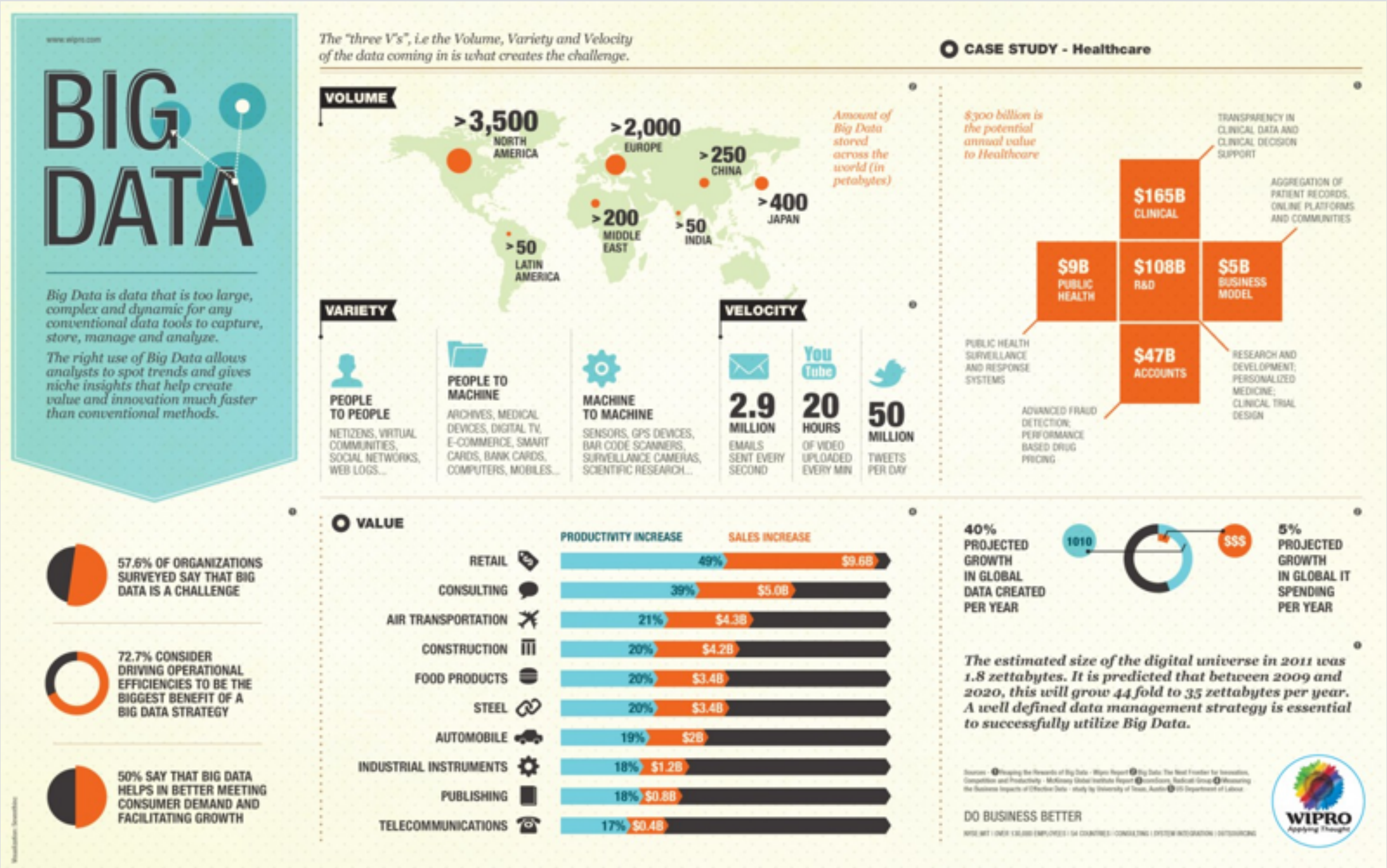
- Big Data (and Analytics)
  - New market for methods used in science since decades
  - but potentially also new methods, which can be applied in science
- Storage
  - Media and Organisation
- Data
  - Structure and Access
- New approaches and technologies
  - Impact on science data management

Big Data

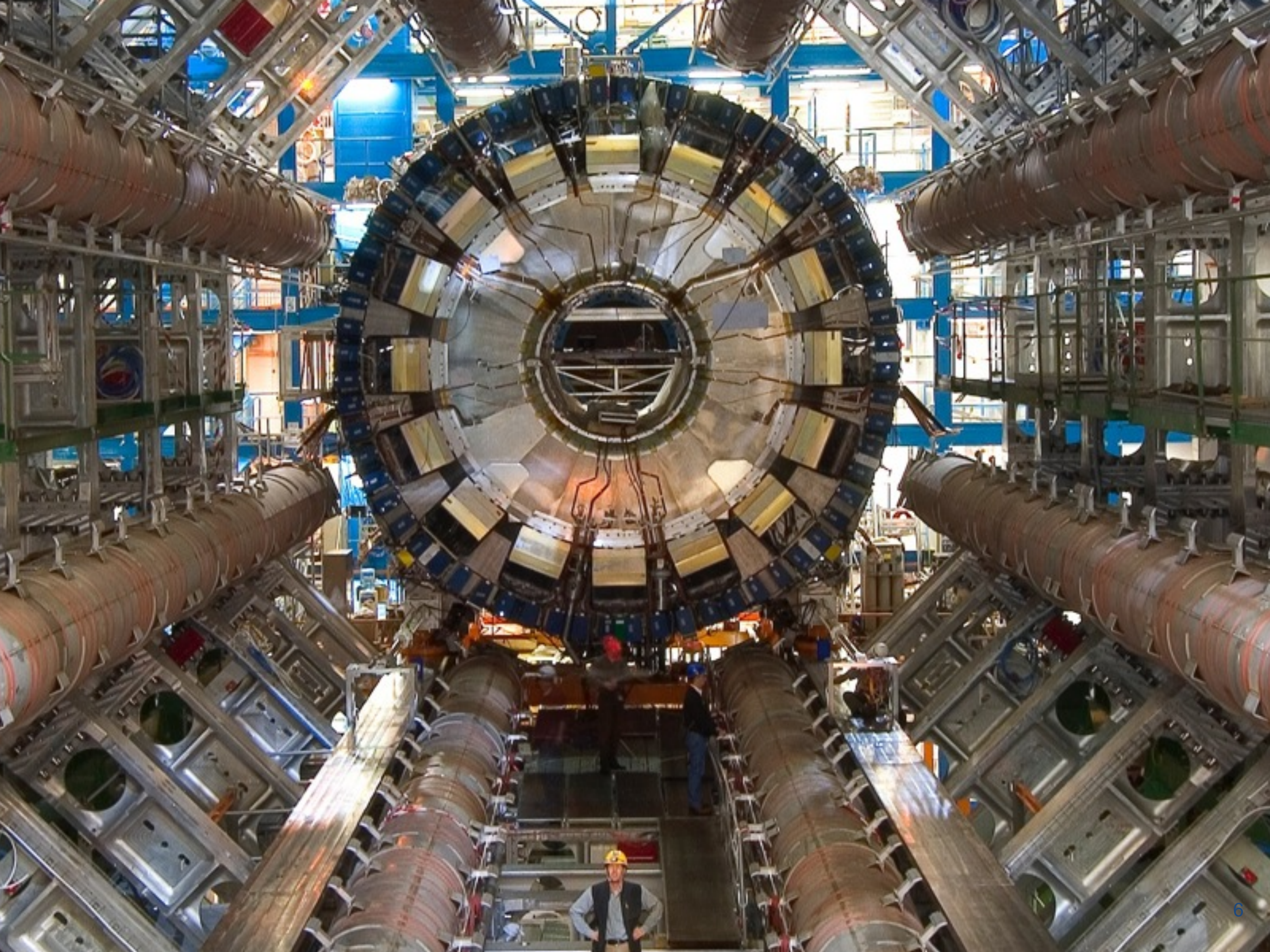
- 35 zettabytes stored by 2020 (worldwide)
  - growing exponentially
- Why? Because …
  - it is technically possible
    - Moore’s & Kryder’s law
    - data volume is proportional to budget
  - it is commercially relevant
    - to digital service providers
    - to marketing

In 60 seconds…

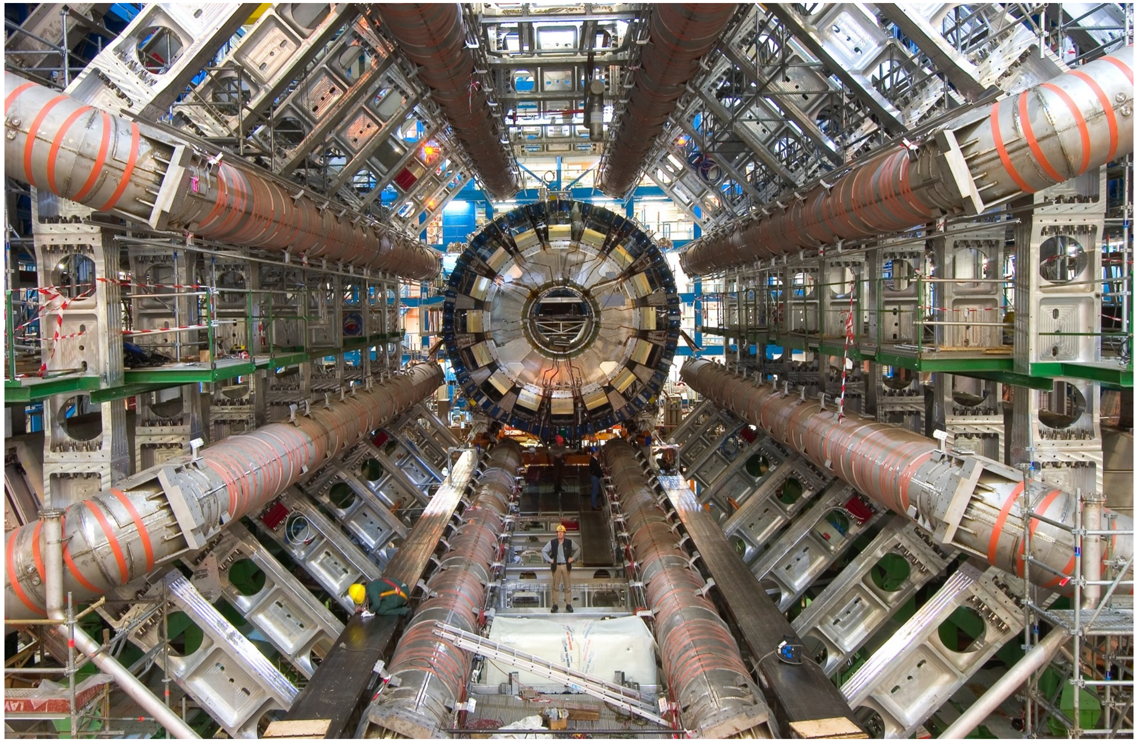
source: www.qmee.com



www.wipro.com
BIG DATA
Big Data is data that is too large, complex and dynamic for any conventional data tools to capture, store, manage and analyze.
The right use of Big Data allows analysts to spot trends and gives niche insights that help create value and innovation much faster than conventional methods.
The "three V's", i.e the Volume, Variety and Velocity of the data coming in is what creates the challenge.
VOLUME
>3,500 NORTH AMERICA
>2,000 EUROPE
>250 CHINA
>400 JAPAN
>200 MIDDLE EAST
>50 INDIA
>50 LATIN AMERICA
Amount of Big Data stored across the world (in petabytes)
VARIETY
PEOPLE TO PEOPLE
NETIZENS, VIRTUAL COMMUNITIES, SOCIAL NETWORKS, WEB LOGS...
PEOPLE TO MACHINE
ARCHIVES, MEDICAL DEVICES, DIGITAL TV, E-COMMERCE, SMART CARDS, BANK CARDS, COMPUTERS, MOBILES...
MACHINE TO MACHINE
SENSORS, GPS DEVICES, BAR CODE SCANNERS, SURVEILLANCE CAMERAS, SCIENTIFIC RESEARCH...
VELOCITY
2.9 MILLION EMAILS SENT EVERY SECOND
You Tube
20 HOURS OF VIDEO UPLOADED EVERY MIN
50 MILLION TWEETS PER DAY
CASE STUDY - Healthcare
$300 billion is the potential annual value to Healthcare
$165B CLINICAL
$9B PUBLIC HEALTH
$108B R&D
$5B BUSINESS MODEL
$47B ACCOUNTS
TRANSPARENCY IN CLINICAL DATA AND CLINICAL DECISION SUPPORT
AGGREGATION OF PATIENT RECORDS, ONLINE PLATFORMS AND COMMUNITIES
PUBLIC HEALTH SURVEILLANCE AND RESPONSE SYSTEMS
RESEARCH AND DEVELOPMENT; PERSONALIZED MEDICINE; CLINICAL TRIAL DESIGN
ADVANCED FRAUD DETECTION; PERFORMANCE BASED DRUG PRICING
57.6% OF ORGANIZATIONS SURVEYED SAY THAT BIG DATA IS A CHALLENGE
72.7% CONSIDER DRIVING OPERATIONAL EFFICIENCIES TO BE THE BIGGEST BENEFIT OF A BIG DATA STRATEGY
50% SAY THAT BIG DATA HELPS IN BETTER MEETING CONSUMER DEMAND AND FACILITATING GROWTH
VALUE
PRODUCTIVITY INCREASE
SALES INCREASE
RETAIL 49% $9.6B
CONSULTING 39% $5.0B
AIR TRANSPORTATION 21% $4.3B
CONSTRUCTION 20% $4.2B
FOOD PRODUCTS 20% $3.4B
STEEL 20% $3.4B
AUTOMOBILE 19% $2B
INDUSTRIAL INSTRUMENTS 18% $1.2B
PUBLISHING 18% $0.8B
TELECOMMUNICATIONS 17% $0.4B
40% PROJECTED GROWTH IN GLOBAL DATA CREATED PER YEAR
1010
$$$
5% PROJECTED GROWTH IN GLOBAL IT SPENDING PER YEAR
The estimated size of the digital universe in 2011 was 1.8 zettabytes. It is predicted that between 2009 and 2020, this will grow 44 fold to 35 zettabytes per year. A well defined data management strategy is essential to successfully utilize Big Data.
DO BUSINESS BETTER
WIPRO
Applying Thought



The ATLAS experiment



7000 tons, 150 million sensors

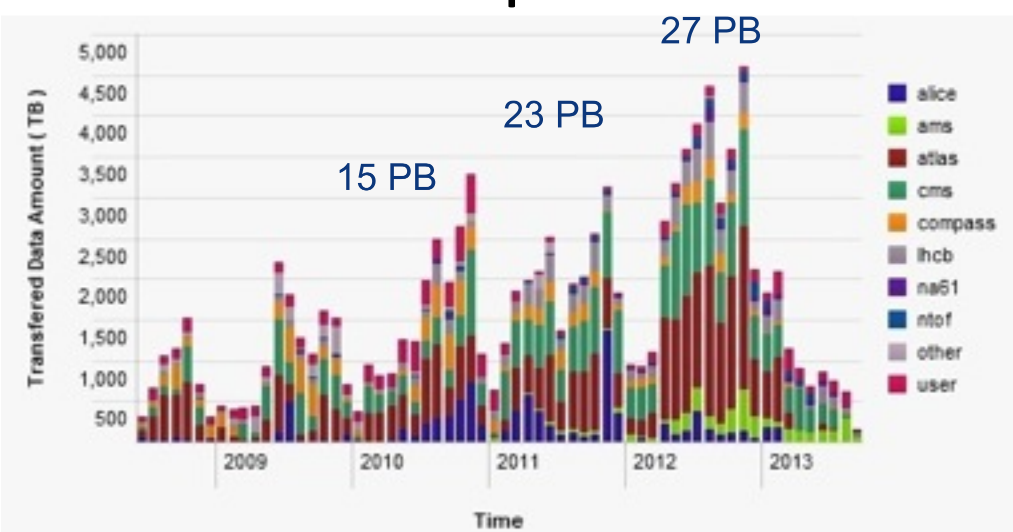
generating data 40 millions times per second

i.e. a petabyte/s

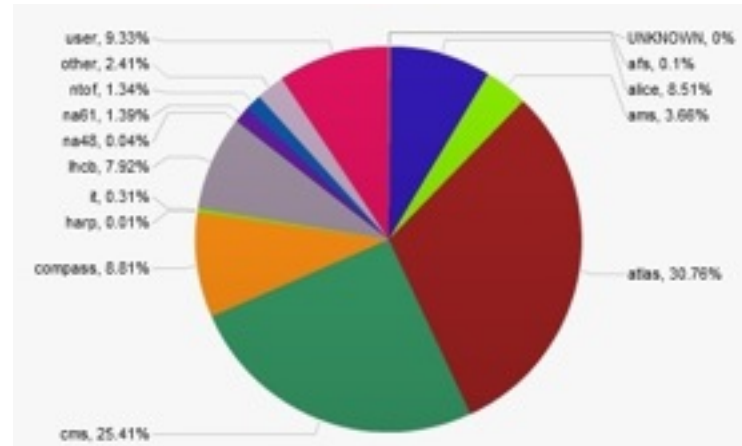
CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Data 2008-2013

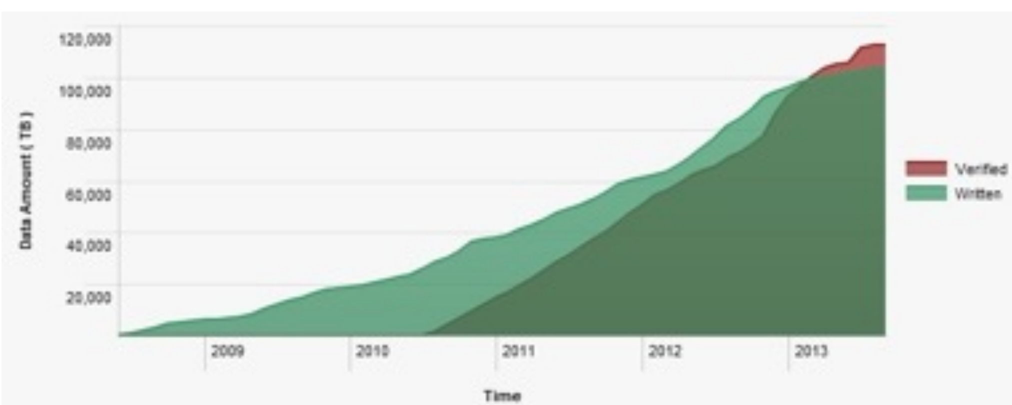
CERN Tape Writes



Tape Usage Breakdown

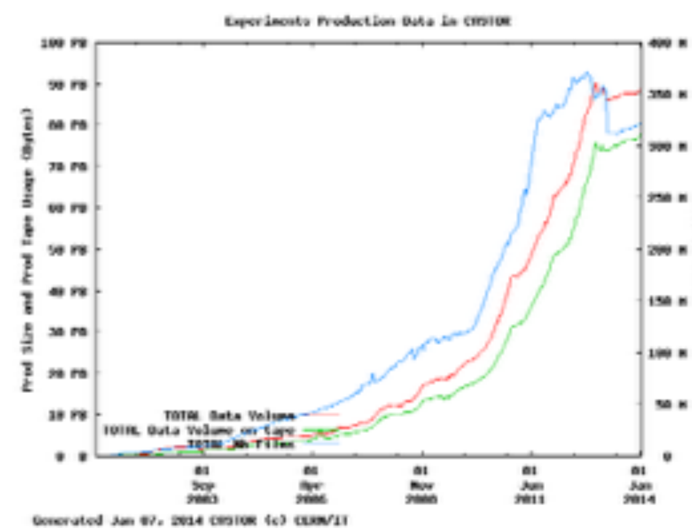


CERN Tape Verification



Data Loss: ~65 GB over 69 tapes
Duration: ~2.5 years

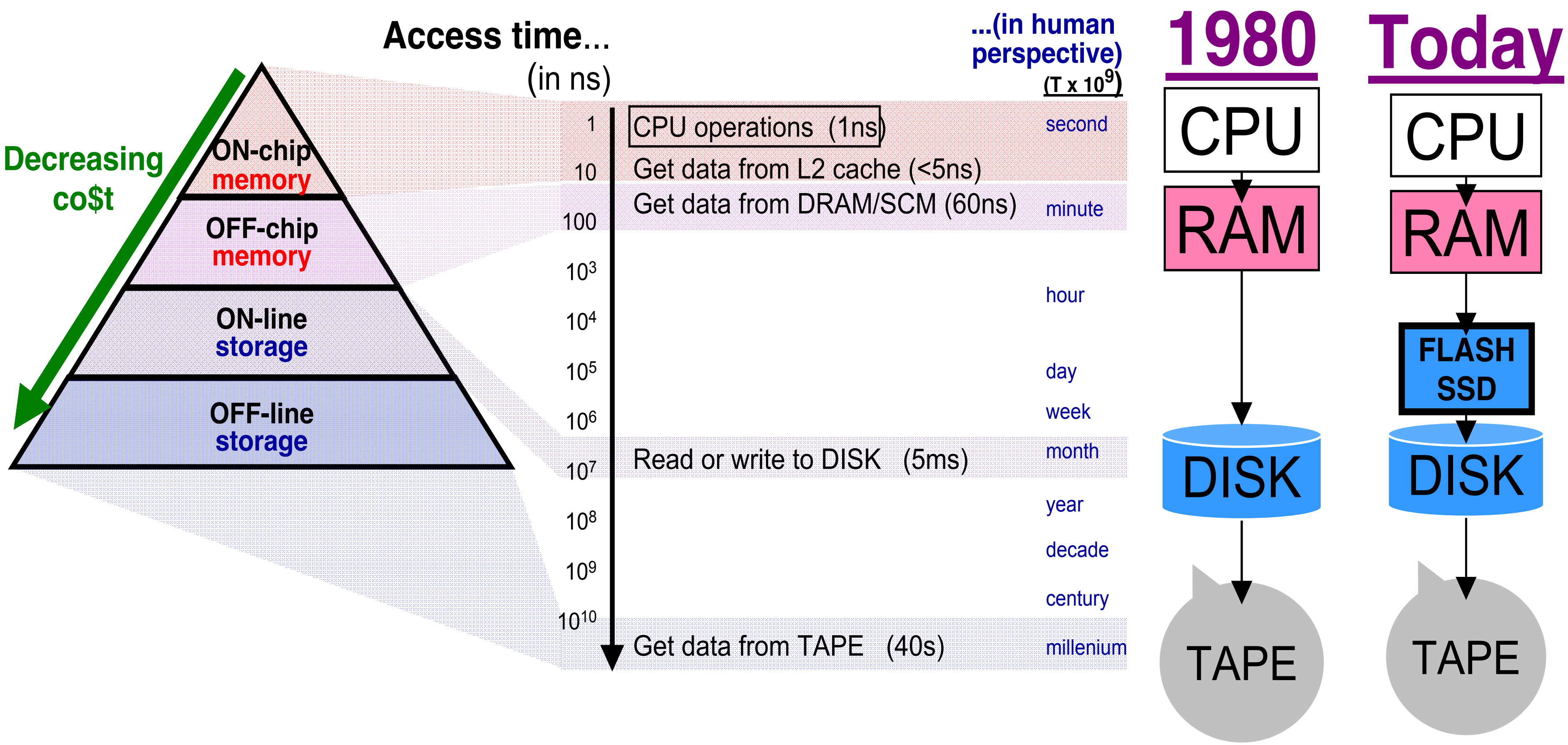
CERN Tape Archive



Storage Media and Organisation

I think Silicon Valley was misnamed. If you look back at the dollars shipped in products in the last decade, there has been more revenue from magnetic disks than from silicon. They ought to rename the place Iron Oxide Valley.

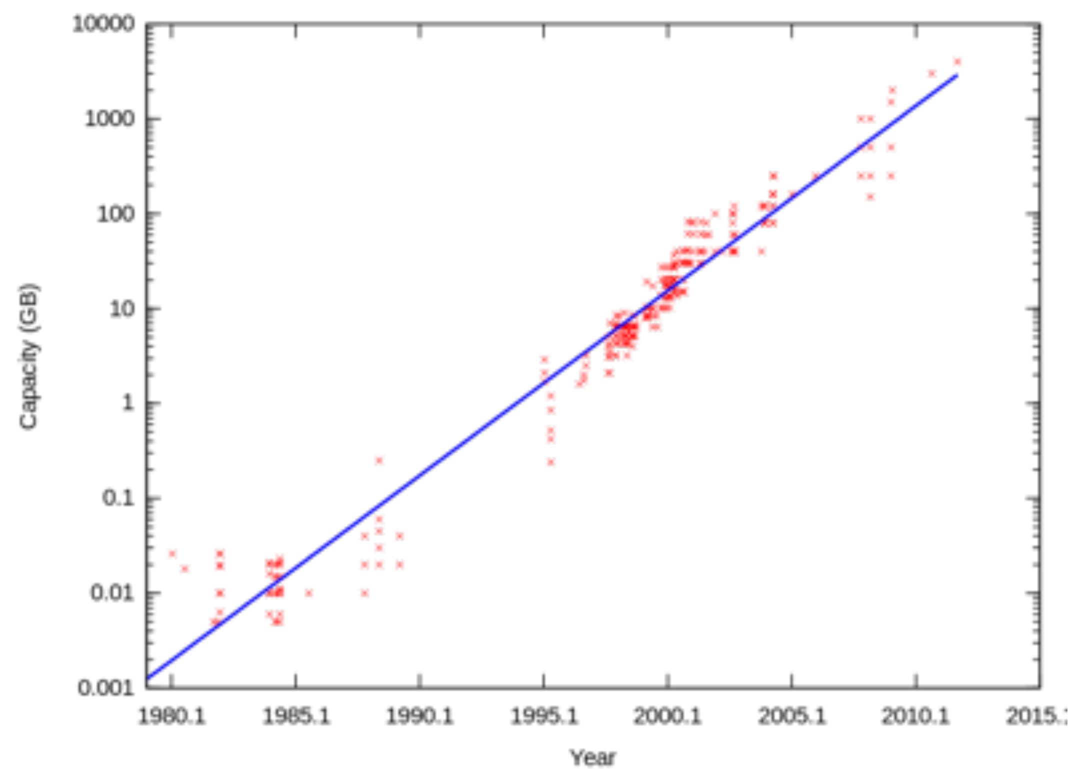
Al Hoagland, pioneer of magnetic disks (1982)



picture adapted from: "Storage class memory", IBM Almaden research centre, 2013

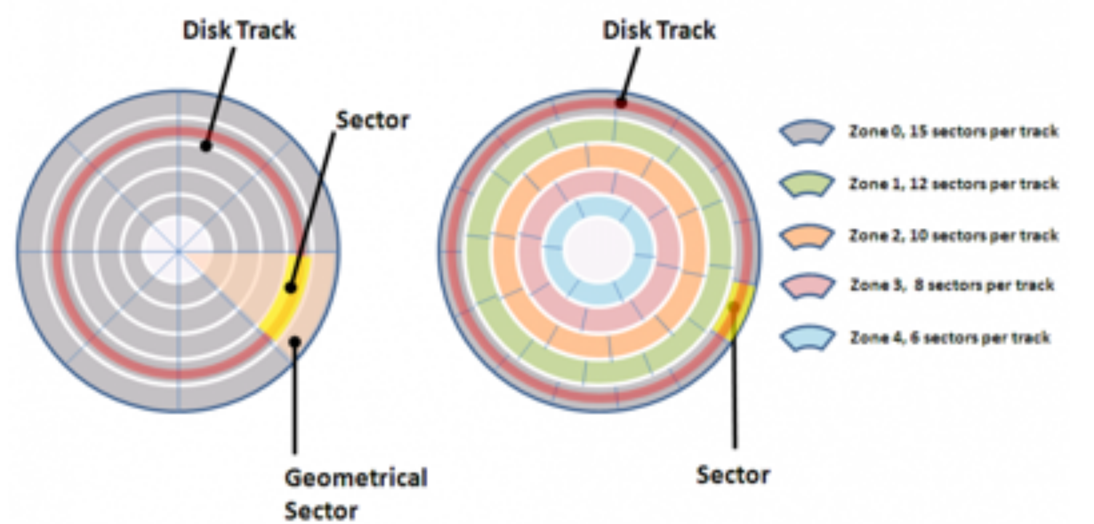
Magnetic Disk

- Kryder's “law” (better observation)
  - magnetic disk areal storage density doubles every 13 months
  - compare to Moore’s “law”: silicon performance doubles “only” every 18 months (combination of density ~~and speed~~)
- Storage volume outperformed CPU
  - or in other words: stored data volume is “cooling down”
  - or relevance of stored data is shrinking



Volume and IOPS

- Storage access time is governed mainly by two components
  - seek time - positioning time of the read head
    - eg 3-10 ms (average)
  - rotational delay of the disk
    - eg 7200rpm disk: 4.2 ms
- Both evolved due to mechanical constraints only over a “small” range - O(10)
- …but storage density has been growing exponentially.



Sequential vs random access

- How does the simple mechanics of rotating disks affect different access patterns?
  - read time = seek time + rotational latency + transfer time
    - sequential: few seeks and rotational waits with long transfers
    - random: one seek and wait per I/O => O(10-100) slower

> The secret to making disks fast is to treat them like tape
> (John Ousterhout)

> Tape is Dead, Disk is Tape,
> Flash is Disk, RAM Locality is King
> (Jim Gray)

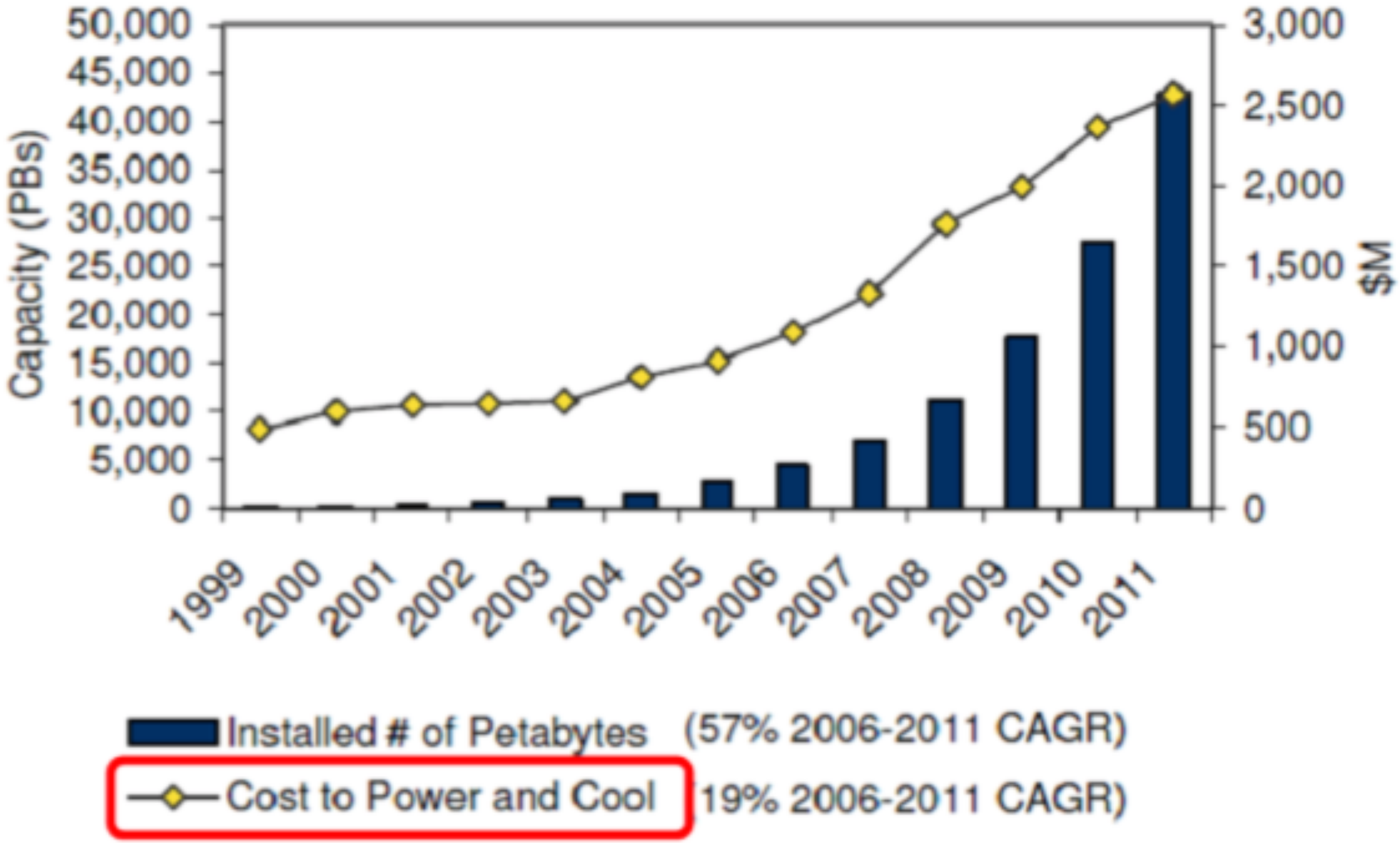
- Gap between sequential and random access is large and increasing with density
  - many concurrent sequential clients sharing storage create random pattern
- Real disks and operating systems try to reorder outstanding I/O requests
  - if the applications can pass multiple request in advance!
- For many database and analysis applications only the lower random rate (or IOPS/s) is relevant
  - and single client benchmarks fail to deliver good performance estimates

Disk Geometry: A "forced" Virtualisation

- Initially : physical addressing
  - #cylinder, #sector, #head
  - file systems optimised (re-sorted) block access for minimal number of seeks and seek distance
- Disk volume growth ran into software (BIOS) constraints
  - Disks had to "pretend" a fake geometry to fit in
  - and obsoleted the now counter-productive geometry optimisations
- Today's addressing method
  - Logical Block Addressing - LBA
  - carries only limited information about physical layout

20 MB IBM PC drive (1984)

Storage System Power & Cooling Cost Trend



SNIA IDC June 2008 – 'The Real Costs to Power and Cool the world's external storage'

Power Consumption

- Storage systems account often for 40% of power consumption
  - magnetic disks have improved, but still show relatively low power efficiency (defined as: power consumed per work done)

- empirically:

$$\text{Power} \approx \text{Diameter}^{4.6} \times \text{RPM}^{2.8} \times \text{Number of platters}$$

=> disks shrink and don't increase in rotational speed

Tuning for special needs…

- "Short stroking"
  - leave inner, slower part of disk unused for applications which need IOPS rather than transfer speed or volume
  - eg transactional databases / random access workloads:
    1TB drive with 12 ms access time : 200 IOPS with 100MB/s can turn into
    100 GB with 6 ms access and 300 IOPS with 200MB/s
- Use "free" inner part of the disk for other "cold" data
  - eg infrequently used backups, redundant replicas

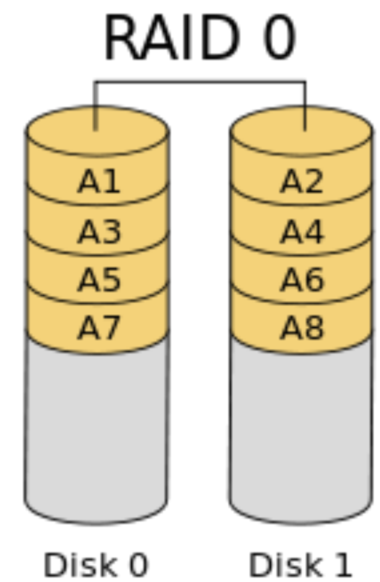
- More generally these two basic ideas can be combined to
  - "chunk-up" <u>all</u> data and spread it randomly over many disks
  - used by CEPH, Hadoop FS, {EOS} and many others

Media Aggregation

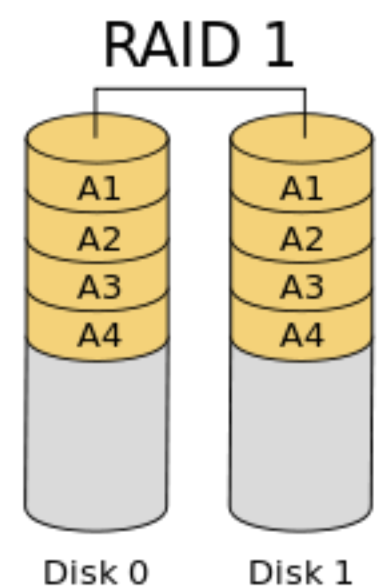
- Goals:
  - virtualise / cluster / federate many individual drive units into a single larger logical unit
  - provide more performance than a single drive
  - provide a larger reliability than the one of a single unit
- Redundant Array of Inexpensive Disks (RAID)
  - sometimes inexpensive => independent
  - initially implemented in dedicated disk controllers and disk arrays - later as pure software module

(Simple) RAID Levels

- RAID 0 - Striping (to n stripes)
  - *failure rate r* and capacity c unchanged
  - potentially: n • disk throughput
  - fault tolerance: none

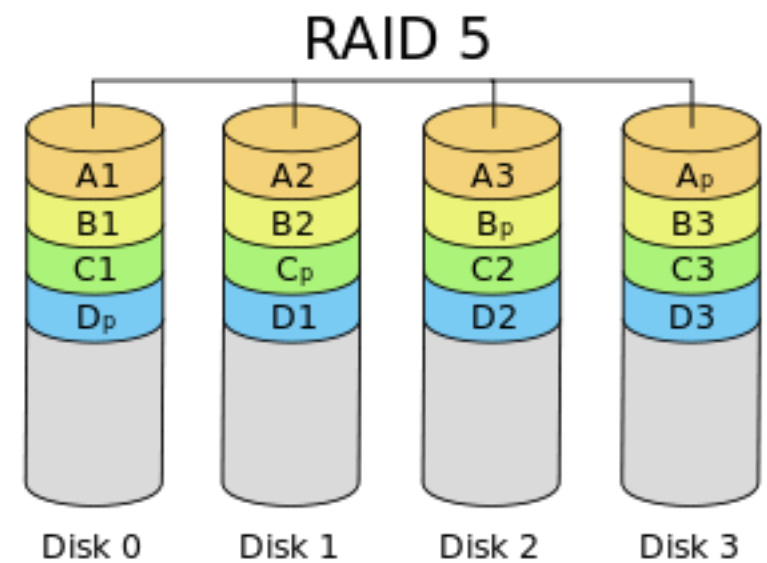


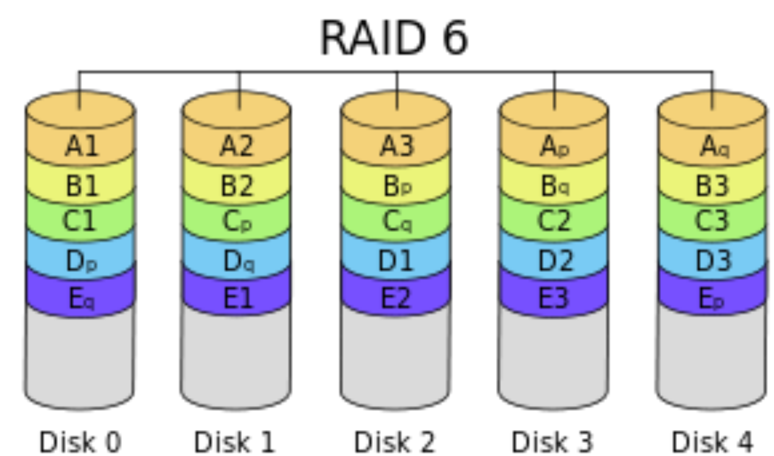
- RAID 1 - Mirroring (to n copies)
  - failure rate = $1-(1-r)^n$
    (assuming independence!)
  - capacity = 1/n • c
  - potentially: n • disk throughput
  - fault tolerance = n -1 drives



More Advanced RAID

- RAID 5 - block striping with distributed parity
  - capacity = $(1-1/n) \cdot c$
  - failure rate = $1 - (1-r)^n - nr(1-r)^{n-1}$
  - fault tolerance = 1 drive
- RAID 6 - adds orthogonal parity
  - fault tolerance = 2 drives





RAID Issues

- Assumption of independent drive errors does not hold
  - eg during recovery
  - drives often share also other common failure sources (power supplies, fans, network etc)
- Drive capacity increase and localised (=long) recovery result in probability for additional fault during recovery => data loss
- Many large scale systems went away from simple drive level RAID aggregation
  - but use the same concept on a higher level (see later slides)

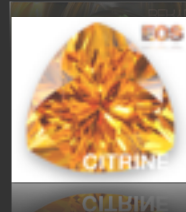
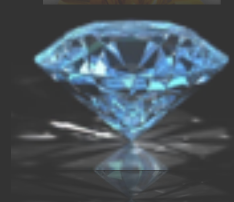
EOS is the CERN disk-only file storage for [non-] LHC derived data targeting physics analysis use cases.

5 instances - **25.000** disks - **60** PB storage space - **200** Mio files

Service in production since 2012 - 1-year availability including scheduled downtimes 99.5%.

External instances at FNAL, SASKE, SINICA, JINR, UNAM

Releases

Beryl		production version
Citrine		in preparation
Diamond		next generation storage bundle

EOS

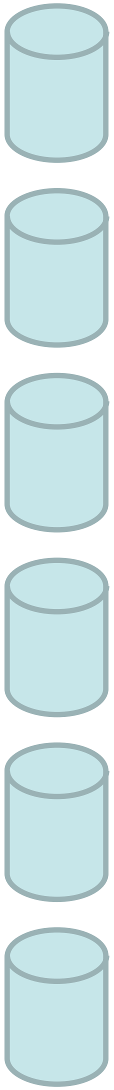
Disk Storage @ CERN

Deployment Simplifications and Development Targets

- Follow trend in many other large storage systems
  - server, controller, disk, file system failures need to be transparently absorbed by storage s/w
  - key functionality: file level replication and rebalancing
- Decouple h/w failures from data accessibility
  - data stays available (for some time at reduced performance) after a failure
  - could change current approach wrt h/w lifecycle
- Fine grained redundancy options on top of a standardised h/w setup
  - eg choose redundancy level (and hence the storage overhead) for individual data rather than globally
- Support bulk deployment operations like retirement and migration building on lower level rebalancing
  - eg retire tens of servers at end of warranty period

EOS approach to redundancy

- EOS uses JBOD disk devices for storage
  - redundancy added on s/w layer
- Using “sets” of N *independent* disk devices
  - Current configuration uses N=6
- Each file / directory / pool can be configured to replicate files M times (with M < N)
  - For example, M=3 every file is written 3 times on 3 *independent* disks out of the 6 available

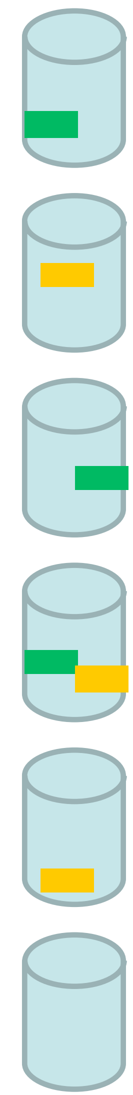


- On client reads:
  - any of the file replicas can be used
  - load is spread across many disks to achieve high throughput
  - more efficient than mirrored disks, and much better than RAID-5 or RAID-6

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

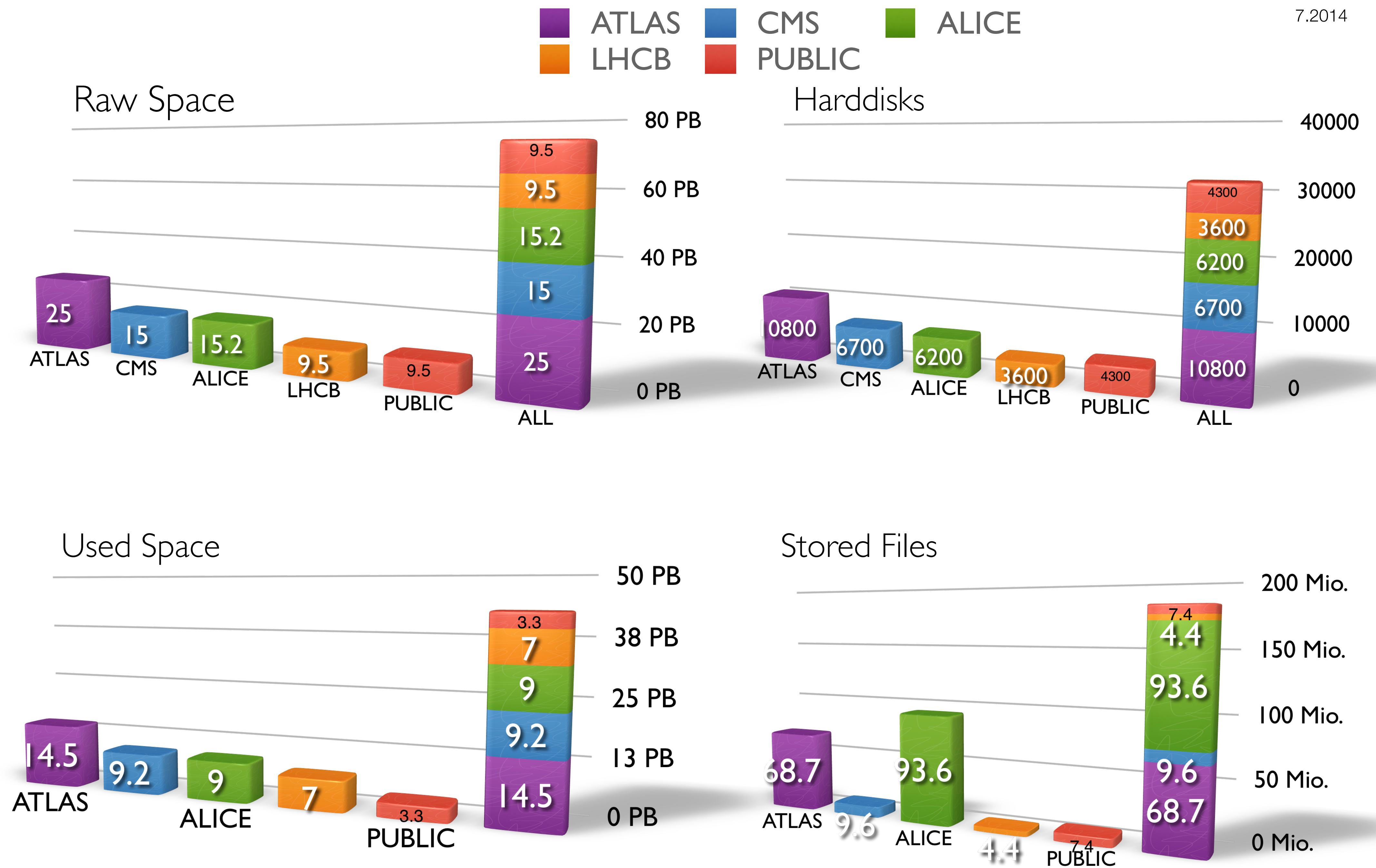
EOS approach to redundancy

- EOS uses JBOD disk devices for storage
  - redundancy added on s/w layer
- Using “sets” of N *independent* disk devices
  - Current configuration uses N=6
- Each file / directory / pool can be configured to replicate files M times (with M < N)
  - For example, M=3 every file is written 3 times on 3 *independent* disks out of the 6 available
- On client reads:
  - any of the file replicas can be used
  - load is spread across many disks to achieve high throughput
  - more efficient than mirrored disks, and much better than RAID-5 or RAID-6

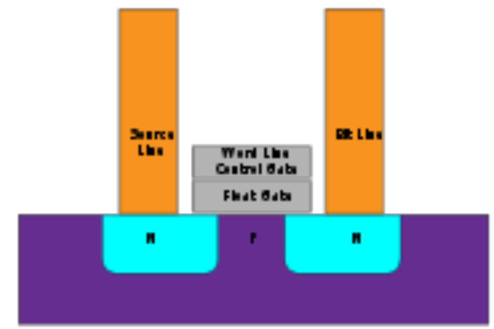


EOS Deployment at CERN

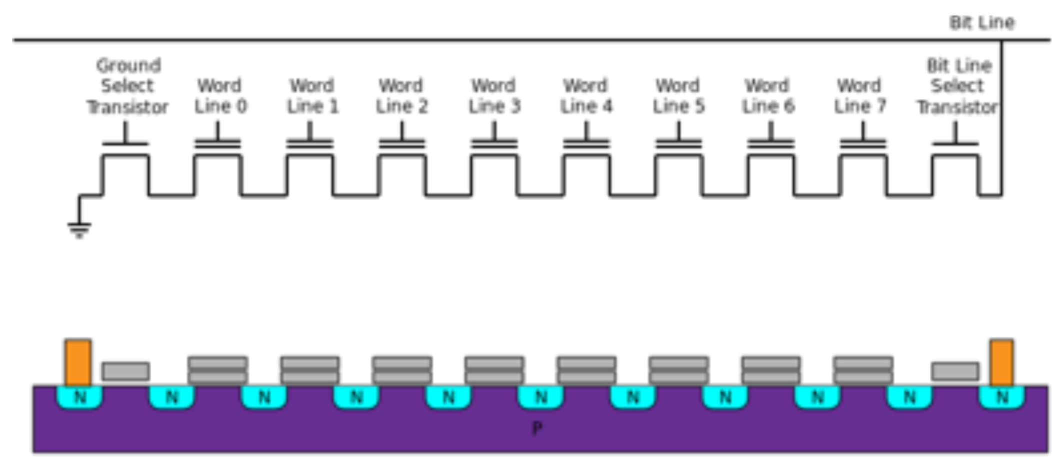
7.2014



Flash Memory

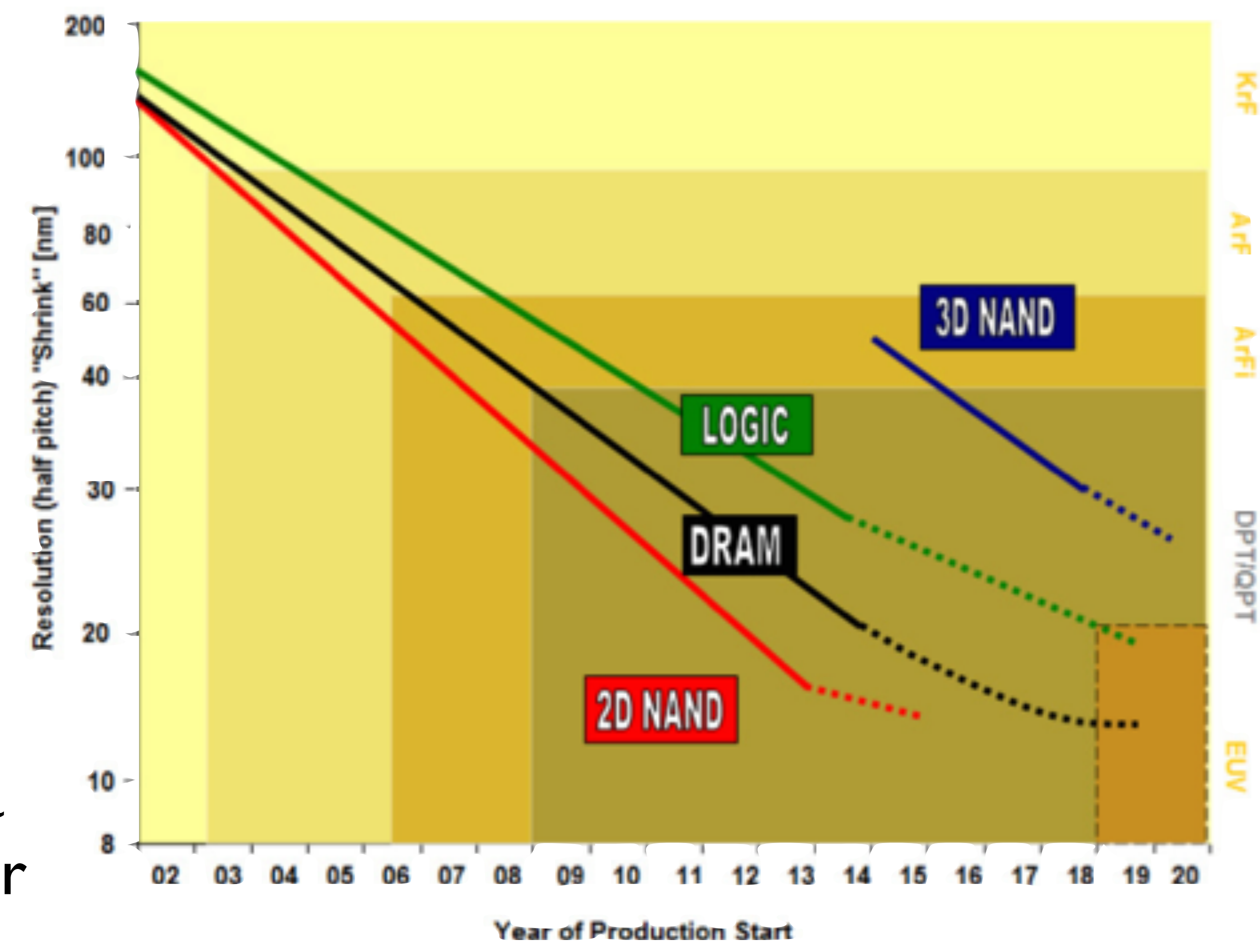


- Memory cell based on "floating gate" MOSFET transistors (Toshiba ~1980)
  - insulated floating gate traps electrons
  - if present, their field shields field from control gate
  - may store single (SLC) or multiple levels (MLC) per cell for ~ years
- Writing and erasure via tunnel effect
- Used widely as USB sticks, SD cards, mobile devices to SSDs



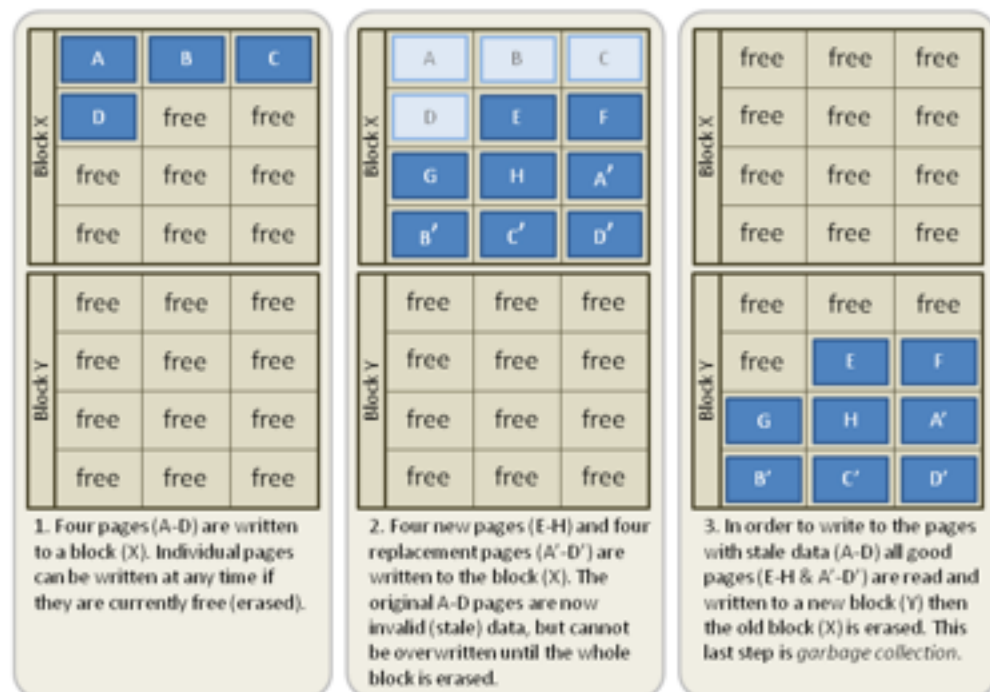
Flash: Basic Properties

- Density ~ Moore's law
- no moving parts
- power efficient
- small form factor
- limited endurance
  - usually 5-100 k erase/write cycles
  - complex internal data management and wear levelling



Flash: unexpected side-effects

- asymmetric read/write performance
- write amplification : factor between user data and resulting flash memory changes
- block recycling : large internal trafic limits client transfers
- past writes influence future performance :
  eg benchmarks on new SSD have only limited value
- limited durability (!= endurance)



SSD vs HDD

- SSD is less well defined - fragmented market
  - Large (factor 20) spread in performance (and price)
  - Several orders of magnitude more IOPS/s
    - current consumer SSDs reach 100k IOPS/s
  - Still O(10) higher price/GB
  - Better power efficiency - in particular for idle storage
- Still a niche solution in a data centre context
  - “Hot” transactional logs from databases or storage system metadata
- BUT - all the mobile market is going there
  - and the server market fraction is decreasing…

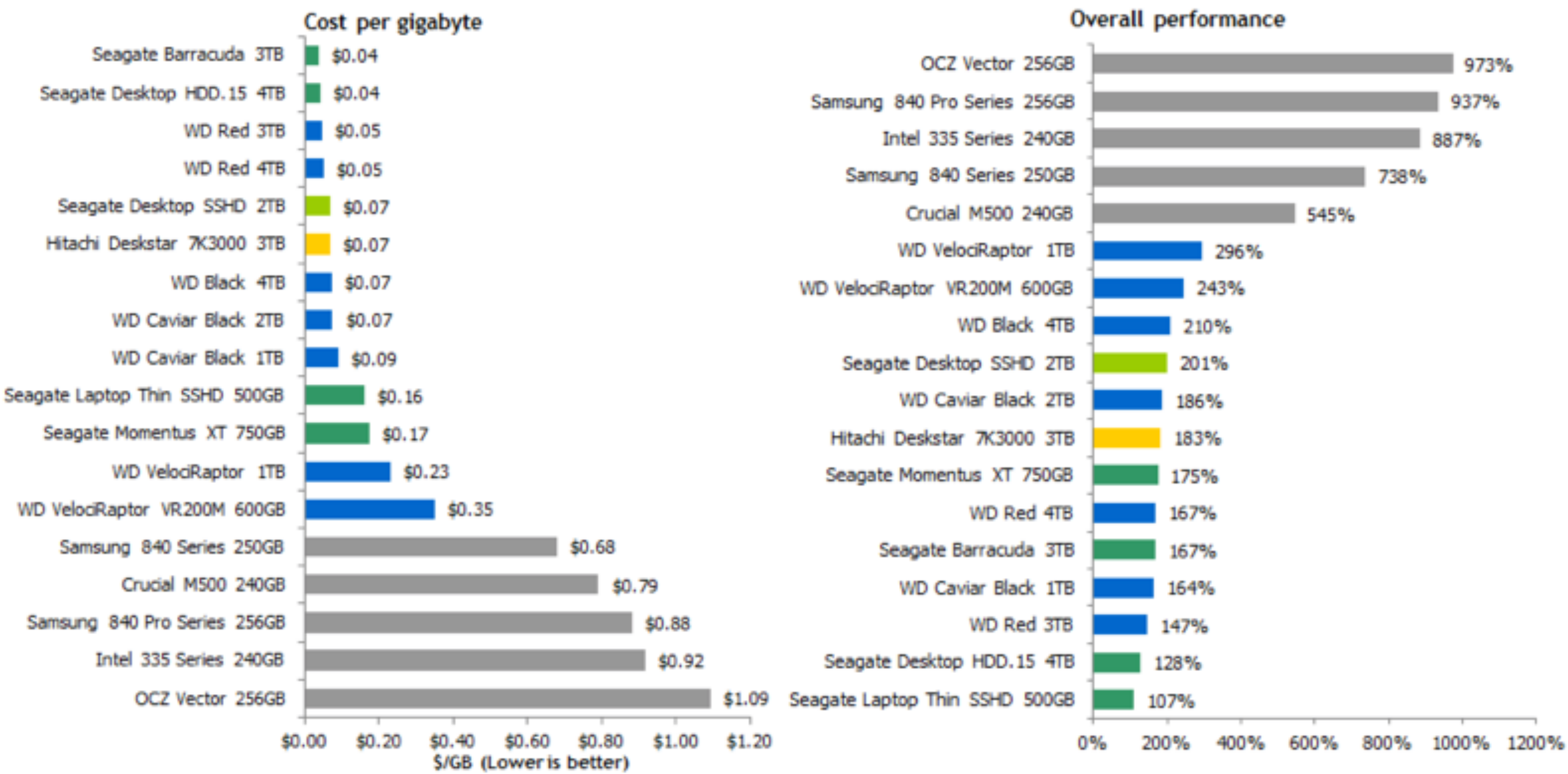
Hybrid Disks

SSD + HDD = SSHD

- eg 8GB Flash cache embedded with 2TB HDD
  - OS agnostic
  - laptop / desktop
  - consumer type workloads, eg speed up booting a machine

- Software options to combine SSD & HDD in a more flexible way - eg:
  - FusionDrive (Apple): caching & tiering
  - ZFS: filesystem cache extension
  - Linux: several tiering projects (eg MyLinear/GreenDM)

A few examples: Performance and $/GB



source: http://techreport.com

Tape

Why Tape is Poised for a George Foreman-Like Comeback

Posted by David Vellante in Compliance, Data Protection, Storage, Wikibon on June 24, 2014

Tape is Dead, Not!

The combination of tape and flash will yield much better performance and substantially lower cost than spinning disk. This statement will prove true for long-term data retention use cases storing large data objects. The implications of this forecast are: 1) Tape is relevant in this age of Big Data; 2) Certain tape markets may actually show growth again; 3) Spinning disk is getting squeezed from the top by flash and from below by a disk/tape mashup we call "flape."

Spinning Disk: Slow and Getting Slower

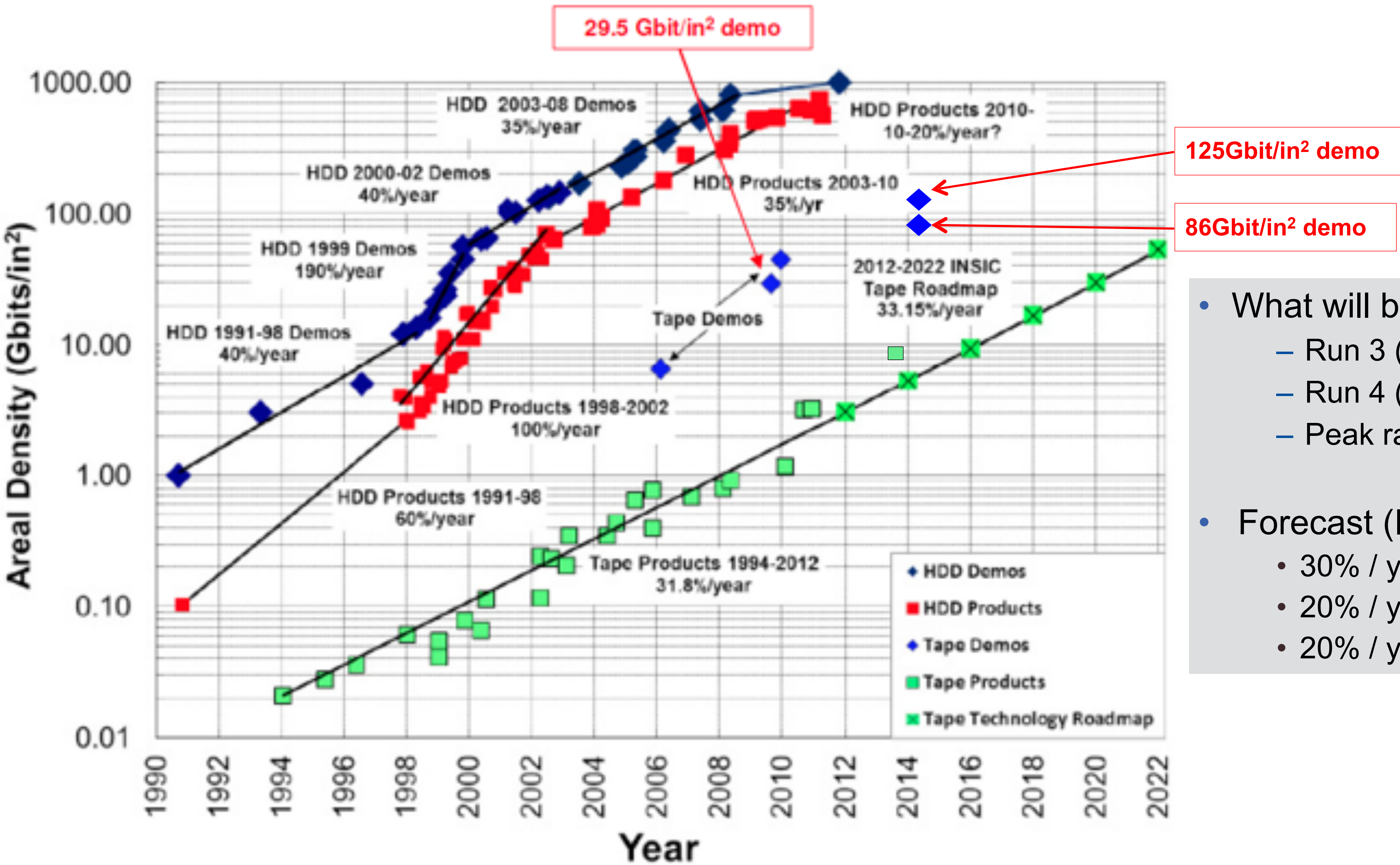
...continue reading the full post

Data Protection, Storage, Tape

7 Comments

source: http://wikibon.org/blog/

Tape vs Disk - Areal Storage Density

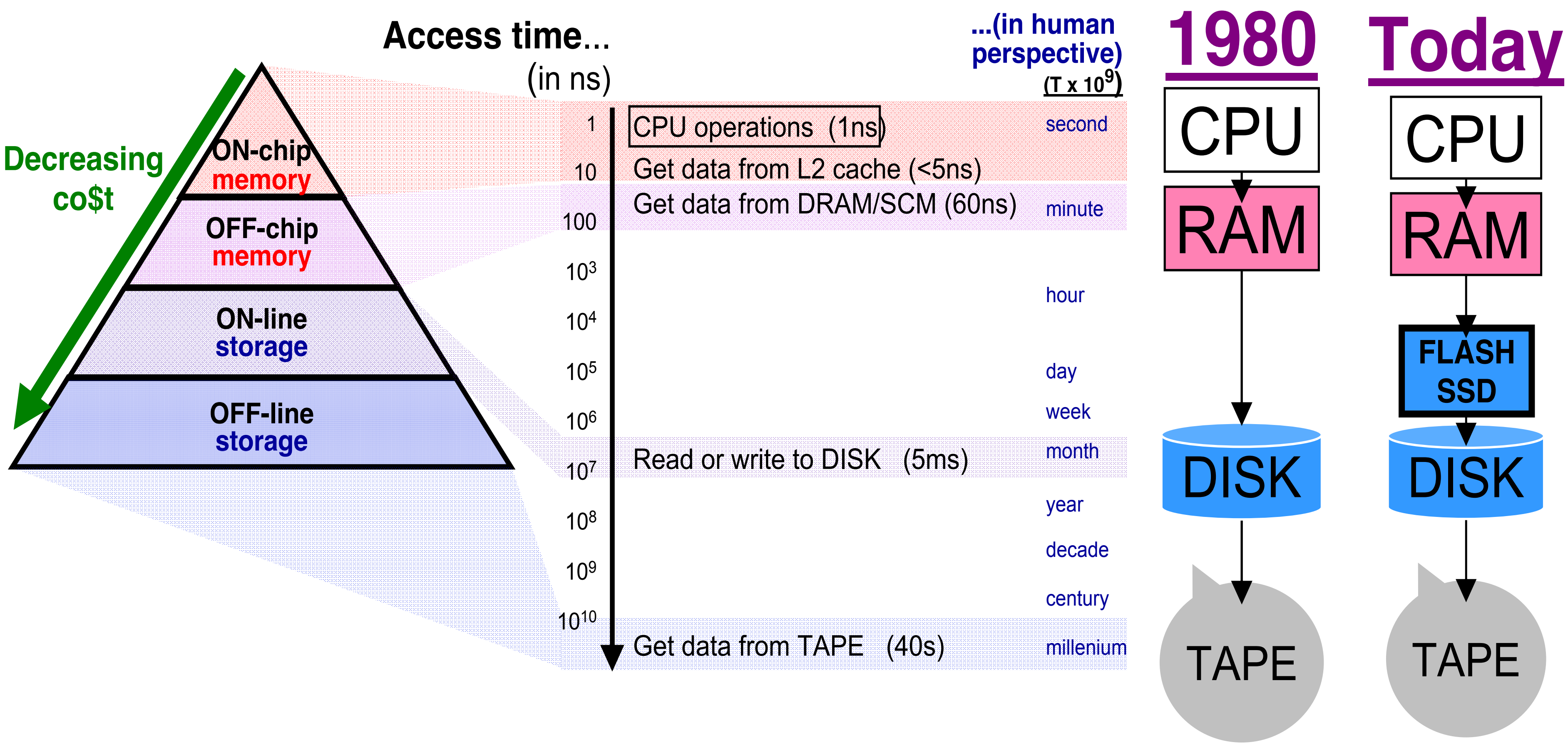


(Source: INSIC 2012-2022 International Magnetic Tape Storage Roadmap)

- What will be required for LHC?
  - Run 3 ('20-'22): ~150PB/yr
  - Run 4 ('23-'29): ~600PB/yr
  - Peak rates of ~80GB/s
- Forecast (INSIC Roadmap)
  - 30% / yr tape capacity per $
  - 20% / yr bandwidth increase
  - 20% / yr disk capacity per $

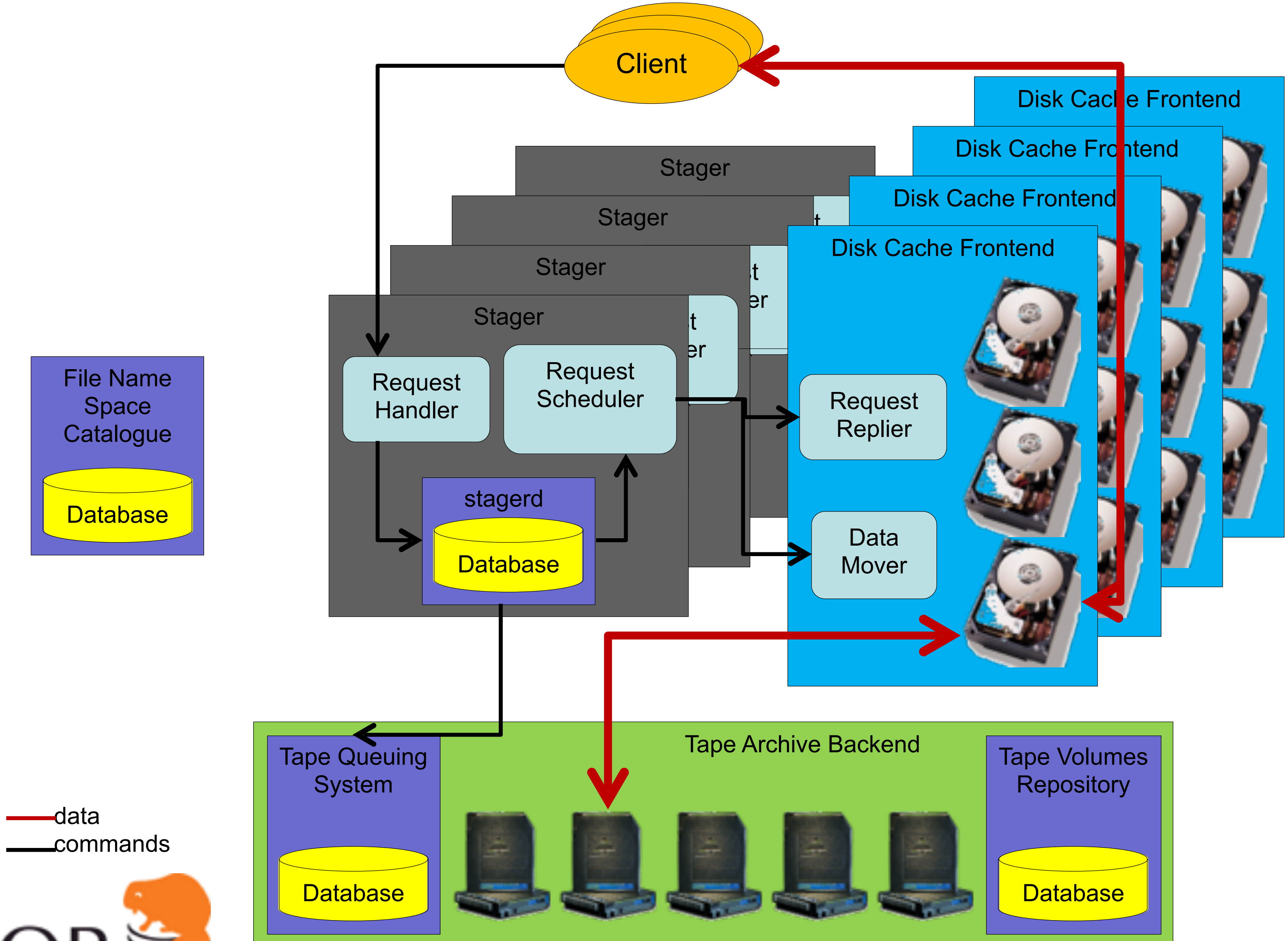
Tape Advantages

- Lowest cost per GB
- Lowest power consumption per GB
- High sequential rate per drive: 250 MB/s
  - increasing faster than disk
- Few vendors, but stable market and continued evolution
- Tape is at one end of the storage media chain with a clear focus on its strength: $/GB



source: storage class memory, IBM Almaden research centre

CASTOR Architecture



CASTOR
CERN Advanced STORage manager

credits: G. Cancio - CERN

Archive Integrity

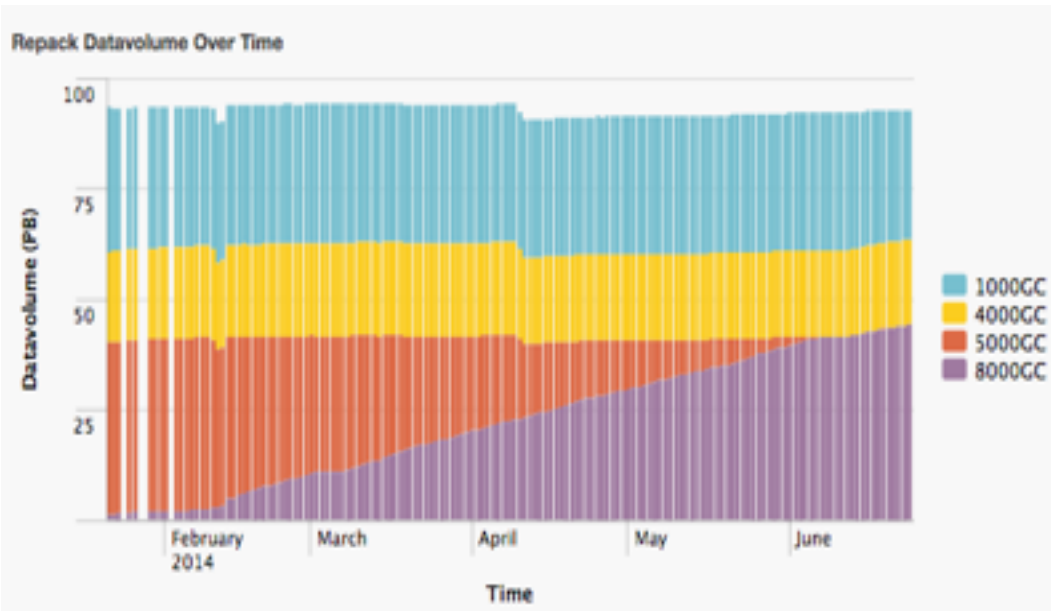
- Scientific data archives often outlive the projects which create them
  - need to insure data integrity also when active user community moved on
- Bit-level preservation
  - regular read and metadata consistency testing on all data
    - “Scrubbing”
  - opportunistically (low priority) with otherwise unused tape drives
- More detail: Data Preservation lectures later this week

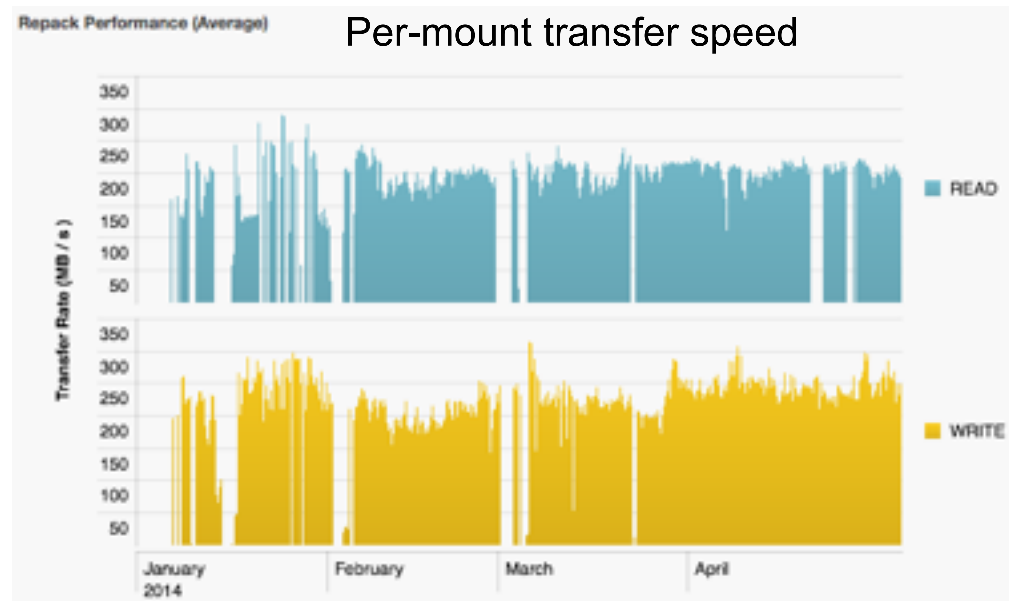
Tape Media and Drive Evolution

- Tape media capacity/$ increases by 30% per year
  - for enterprise class also via a drive firmware/media updates
- At CERN all archived data is migrated every 2-3 years
  - from last generation drive/media to next
  - additional assurance of data integrity
  - free archive space for incoming data from LHC and other experiments
- Causes similar I/O load as one LHC experiment !
  - and runs for about one year

Current “Repack” Exercise at CERN

- Repacking ~2PB / week
  - sustained ~3.4GB/s with 16 drives; ~206 MB/s avg per drive, write peaks up to 8.6GB/s
  - No surprises, since data was pre-verified



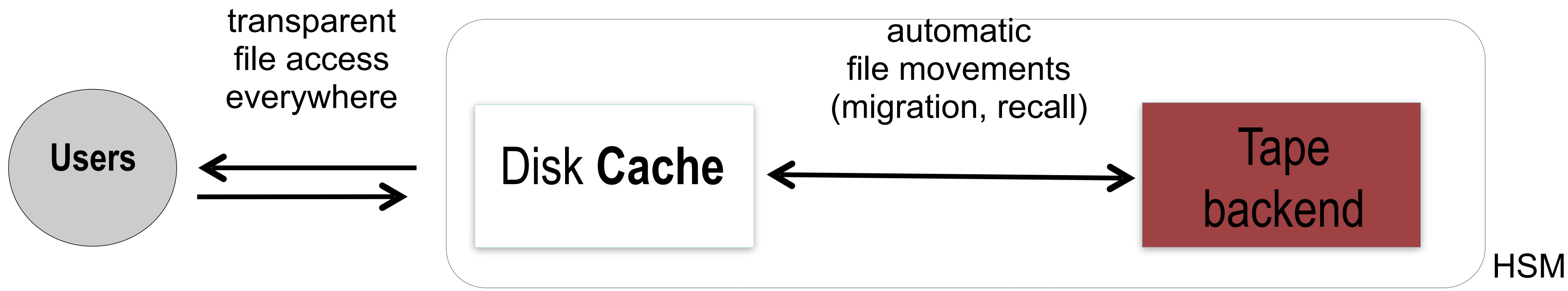


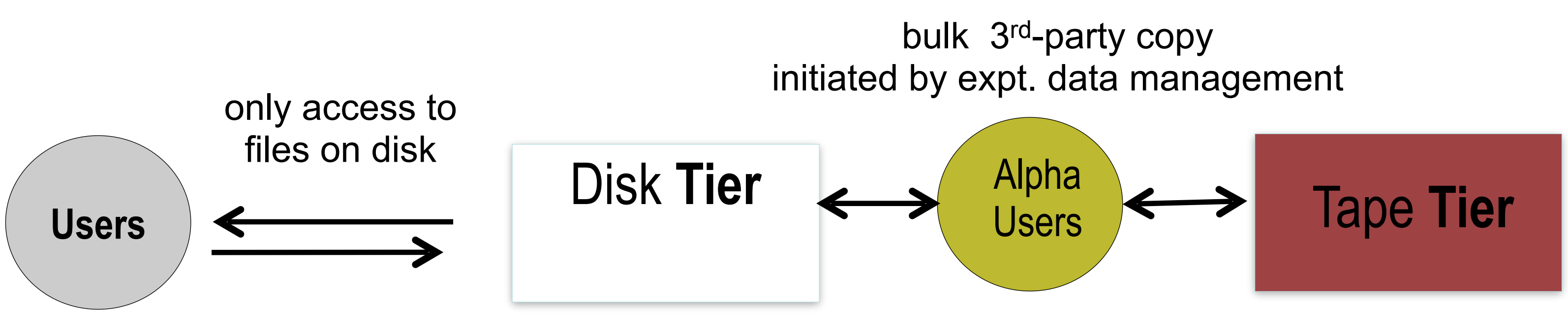
- With this performance repack could complete Q4 2014
  – next drive generation expected for Q4 2014 -> ~20PB to be done in Q1 2015
- Important validation for CASTOR tape + stager software stack
  – Confidence for physics use cases with high data rate
  – Eg LHC Run2 Pb-Pb data rates (~10GB/s): OK

credits: G. Cancio - CERN

Hierarchical vs Tiered Storage

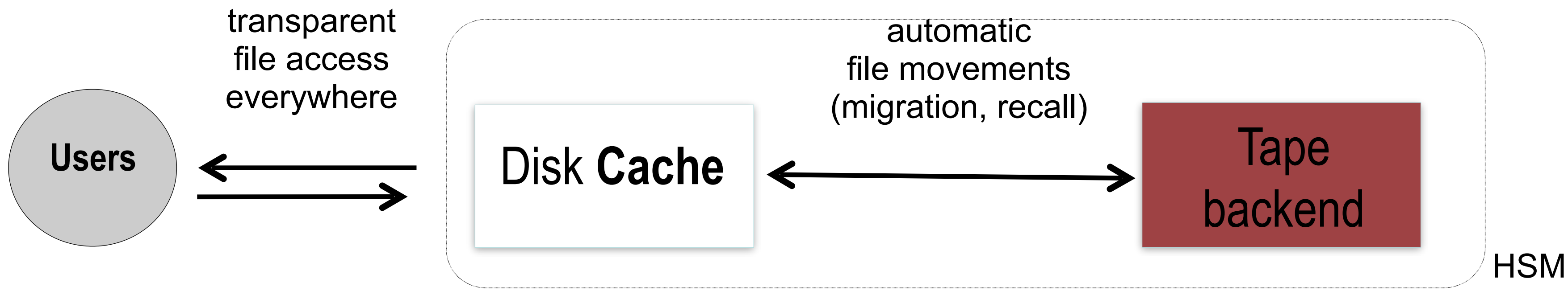
Move away from "transparent", file/user based HSM





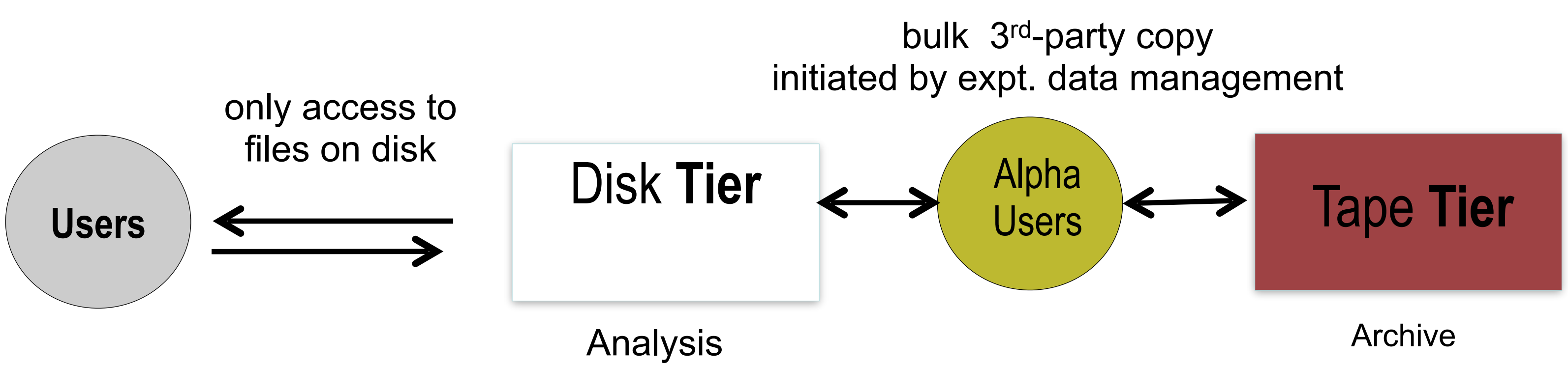
Hierarchical vs Tiered Storage

Move away from “transparent”, file/user based HSM



Model change from HSM to more loosely coupled Data Tiers

- Separate Analysis from other Use Cases
- Introduce a new (decoupled) system for random-access data analysis
- Tape access limited to privileged users who manage the disk pools
  - Data “management” is better done by the data owner (experiment) who has upfront knowledge about data campaigns, access patterns and relative resource priorities



CASTOR tape sw evolution

Investigated alternatives to (parts of) CASTOR software stack

- Amazon Glacier: potential as simple tape front-end interface
  - “stripped down S3” WS-based interface; minimal metadata and operations
  - .. but in reality, coupled to S3 infrastructure; key functionality missing from API (redirection support, no staging concept, etc) ; modest interest from Amazon to share knowledge with CERN
- LTFS: abstraction layer (POSIX) on top of complex tape I/O
  - Shipped by IBM and Oracle; being adopted by film industry
  - High complexity and low maturity, incompatible with present ANSI format, diverging (and non-OSS) extensions for library management

Strategy: re-engineer rather than replace CASTOR tape layer

- Replace CASTOR tape server codebase
  - Code aged (20+ years) , full of legacy OS/hardware, exotic tape formats and pre-CASTOR support
  - Replace 10+ daemons and executables by two: tape mounting and serving
  - Extensions such as Logical Block Protection and Ceph client support
- Review CASTOR drive queue / volume management services
  - Provide a single integrated service, take better into account reduced number of higher-capacity tapes
  - Avoid drive write starvation problems, better load-balancing, allow for pre-emptive scheduling (ie user vs verification jobs)

Tape Market evolution (1)

- New tape drives and media released or in pipeline

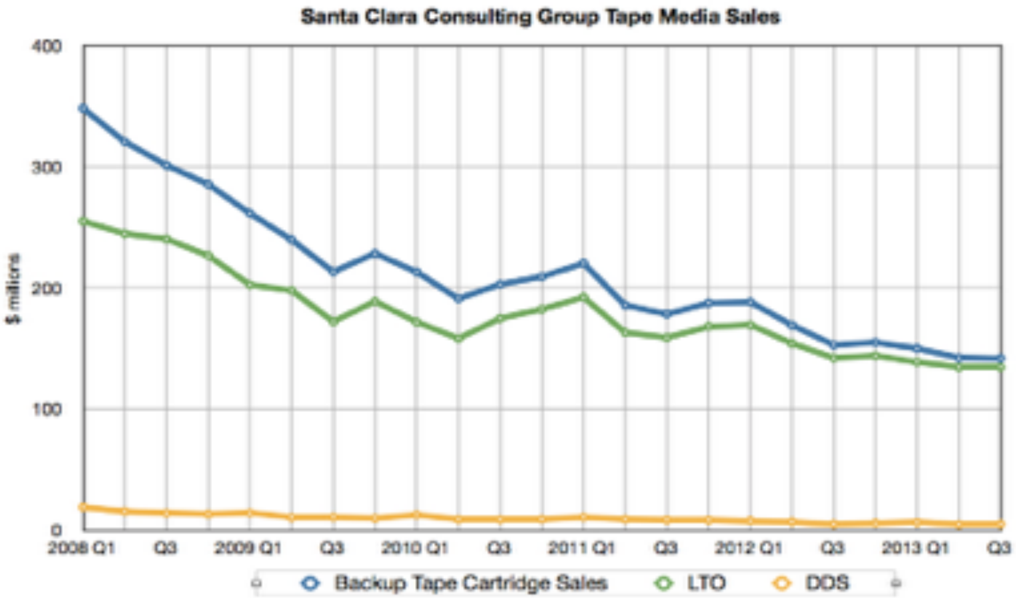
Vendor	Name	Capacity	Speed	Type	Date
IBM	***TS1140***	***4TB***	***240MB/s***	***Enterprise***	***06/2011***
LTO(*)	LTO-6	2.5TB	160MB/s	Commodity	12/2012
Oracle	***T10000D***	***8.5TB***	***252MB/s***	***Enterprise***	***09/2013***
IBM	???	???	???	Enterprise	???

- R&D and Roadmaps for further evolution
  - Change from MP to BaFe media allowing finer particles and magnetisation
    - 45Gb/in^2 demo (~50TB tape) – announced 5/2010
    - 85.9Gb/in^2 demo by IBM/Fuji (~154TB tape) – announced 5/2014
  - Sony demonstration 4/2014: 125Gb/in^2 (~185TB) with sputtered CoPtCr
    - Cost of media production could be a concern
  - LTO Roadmap: LTO-7: 6.4TB (~2015), LTO-8: 12.8TB (~2018?)
  - Next enterprise drives generation? 2017? 15-20TB? (~2017)
  - Little / no improvements in tape loading/positioning

(*) : IBM/HP/Quantum (drives); Fuji/Maxell/TDK/Sony (media)

Tape Market evolution (2)

- Commodity tape market is consolidating
  - LTO market share is > 90%; but market shrinking by ~5-10% / year (~600M$ / yr in 2013)
  - Small/medium sized backups go now to disk
  - TDK & Maxell stopping tape media production; other commodity formats (DAT/DDS, DLT, etc) frozen
  - LTO capacity increase slower (~27% / year compared to ~40% / year for enterprise)

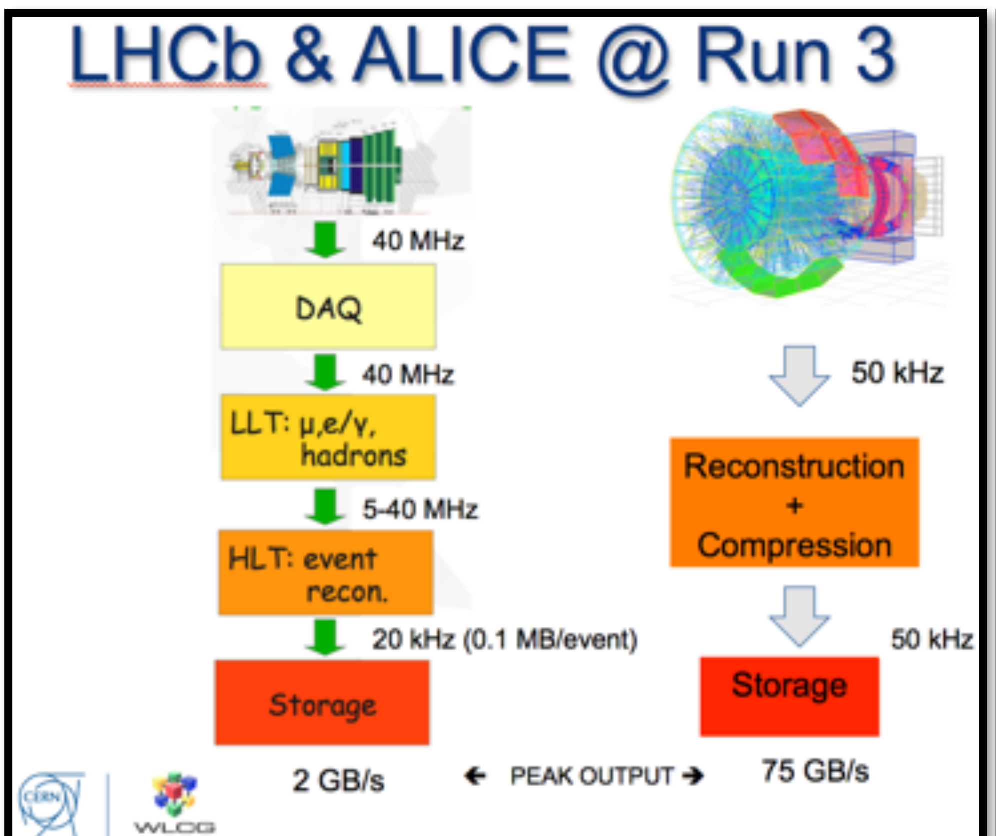


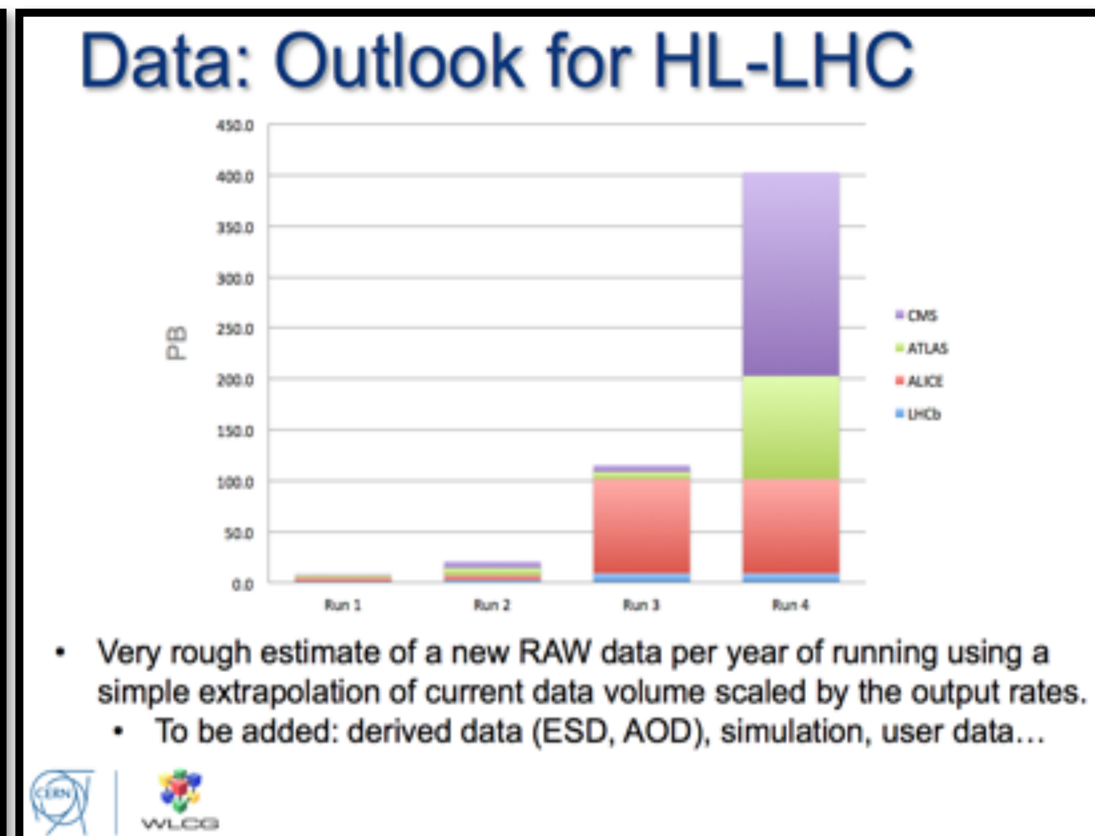
- Enterprise tape is a profitable, growing (but niche) market
  - Large-scale archive market where infrastructure investment pays off
    - e.g. Google (O(10)EB), Amazon(?)), scientific (SKA – up to 1EB/yr), ISP's, etc
  - Sufficient to drive tape research and production?
  - Competition from spun-down disk archive services ie Evault LTS2 (Seagate)

LHC archiving needs - some model calculations

LHC archiving needs - some model calculations

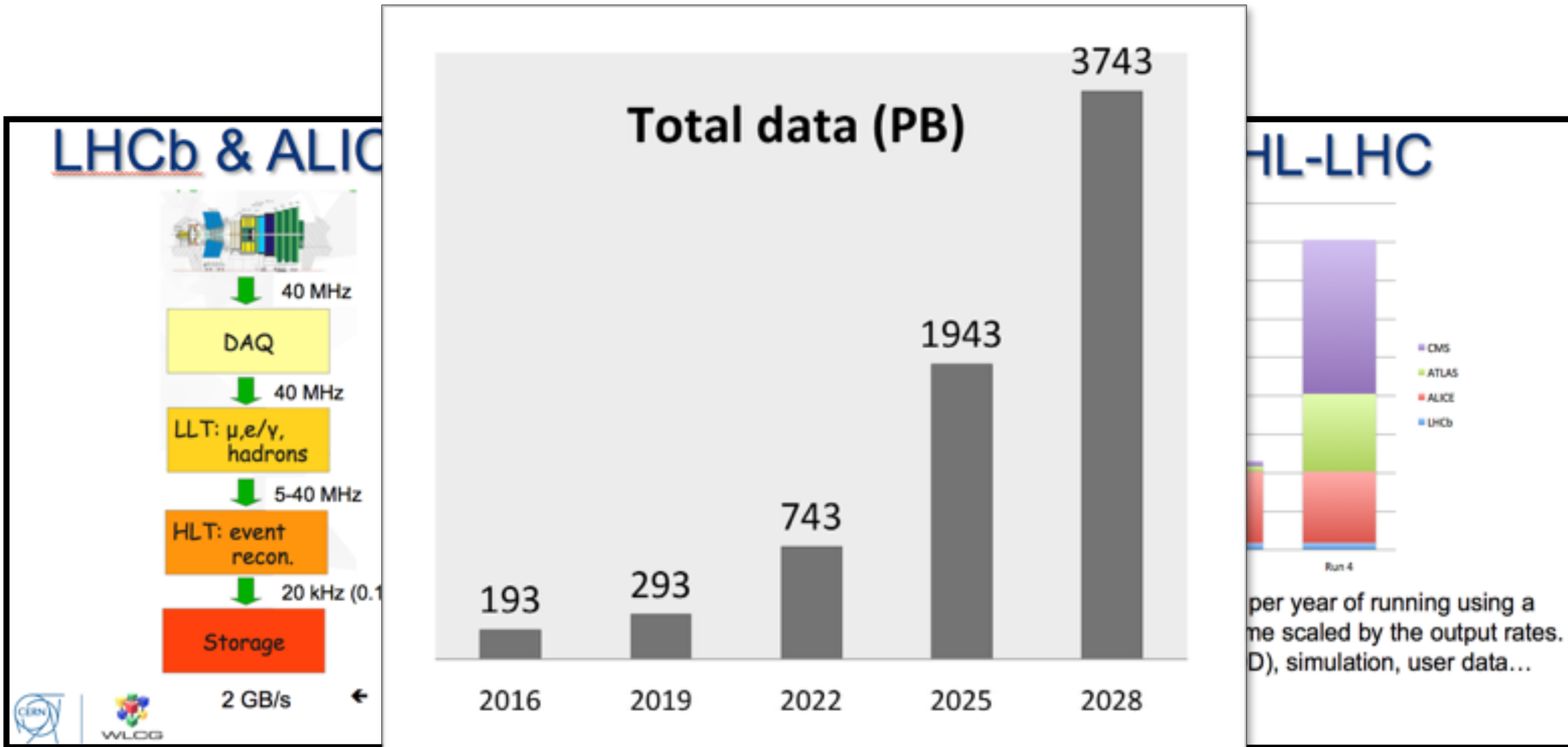
- Beyond 2018?
  - Run 3 (2020-2022): ~150PB/year
  - Run 4 (2023-2029): ~600PB/year
  - Peak rates of ~80GB/s





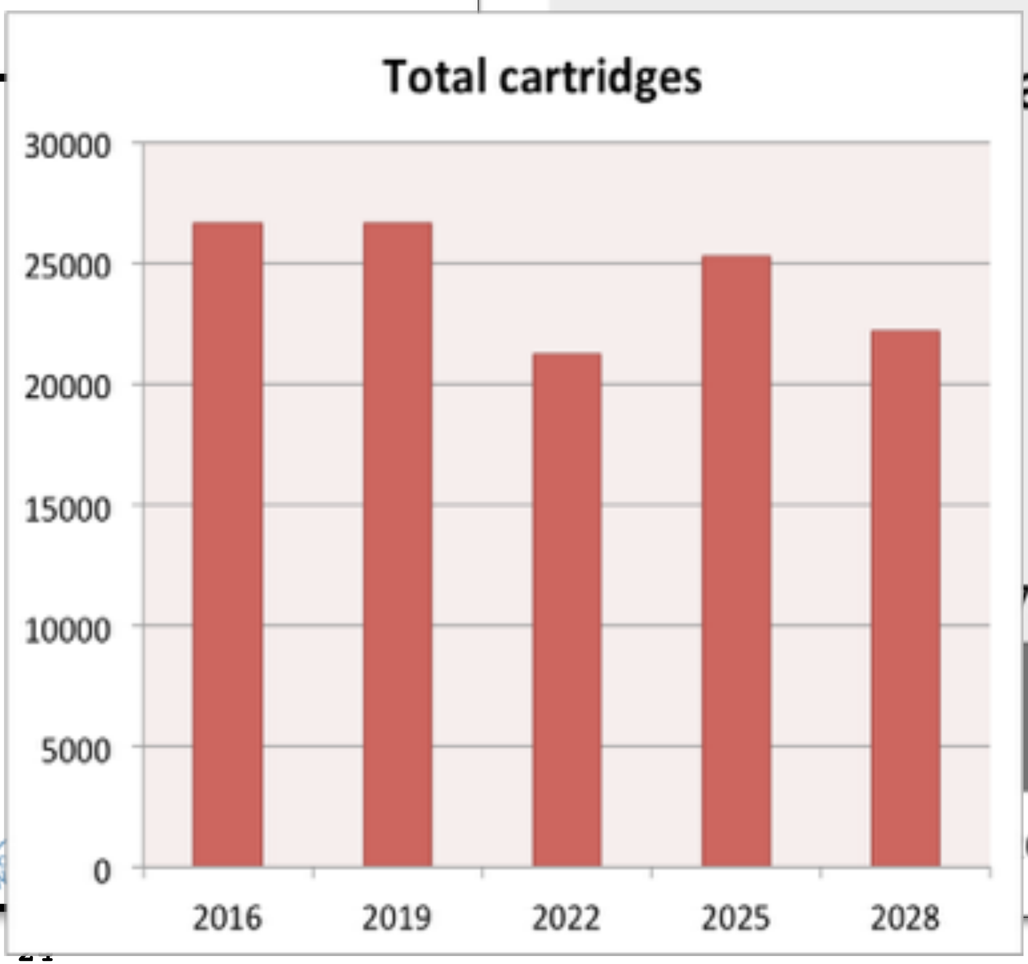
LHC archiving needs - some model calculations

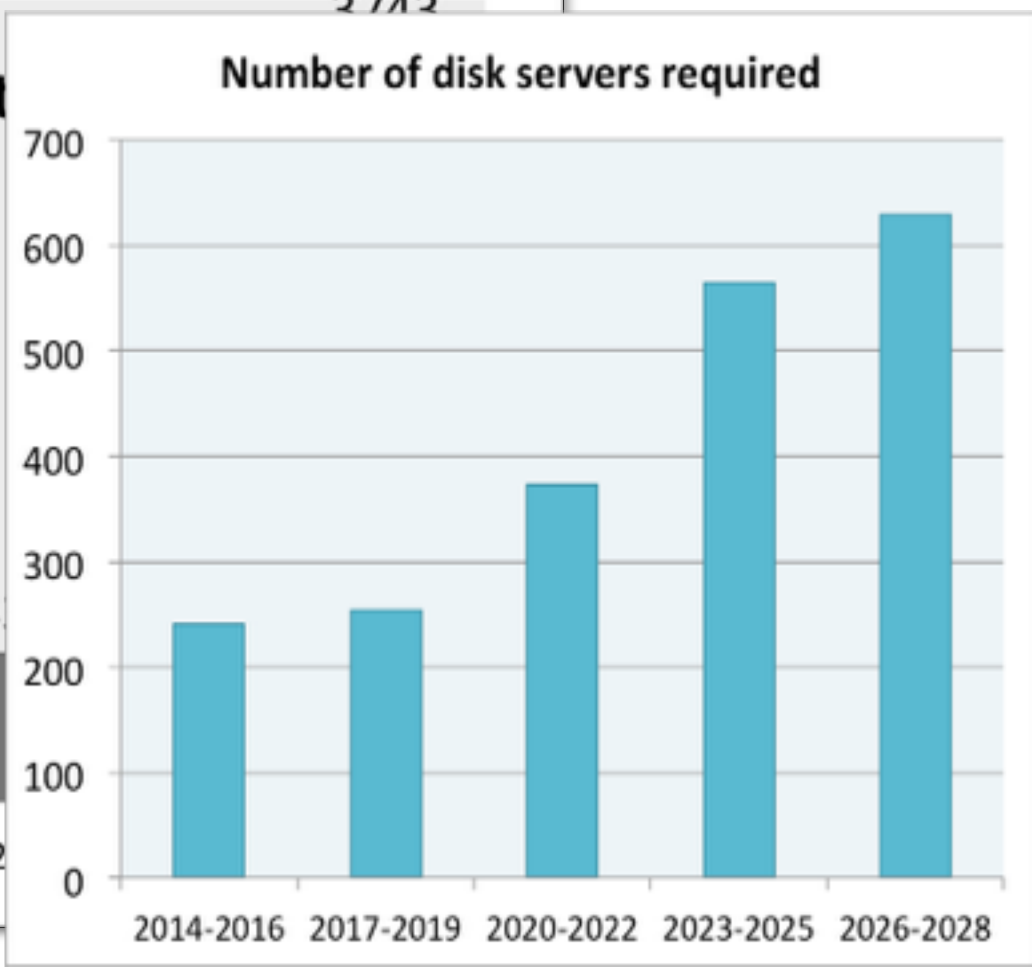
- Beyond 2018?
  - Run 3 (2020-2022): ~150PB/year
  - Run 4 (2023-2029): ~600PB/year
  - Peak rates of ~80GB/s



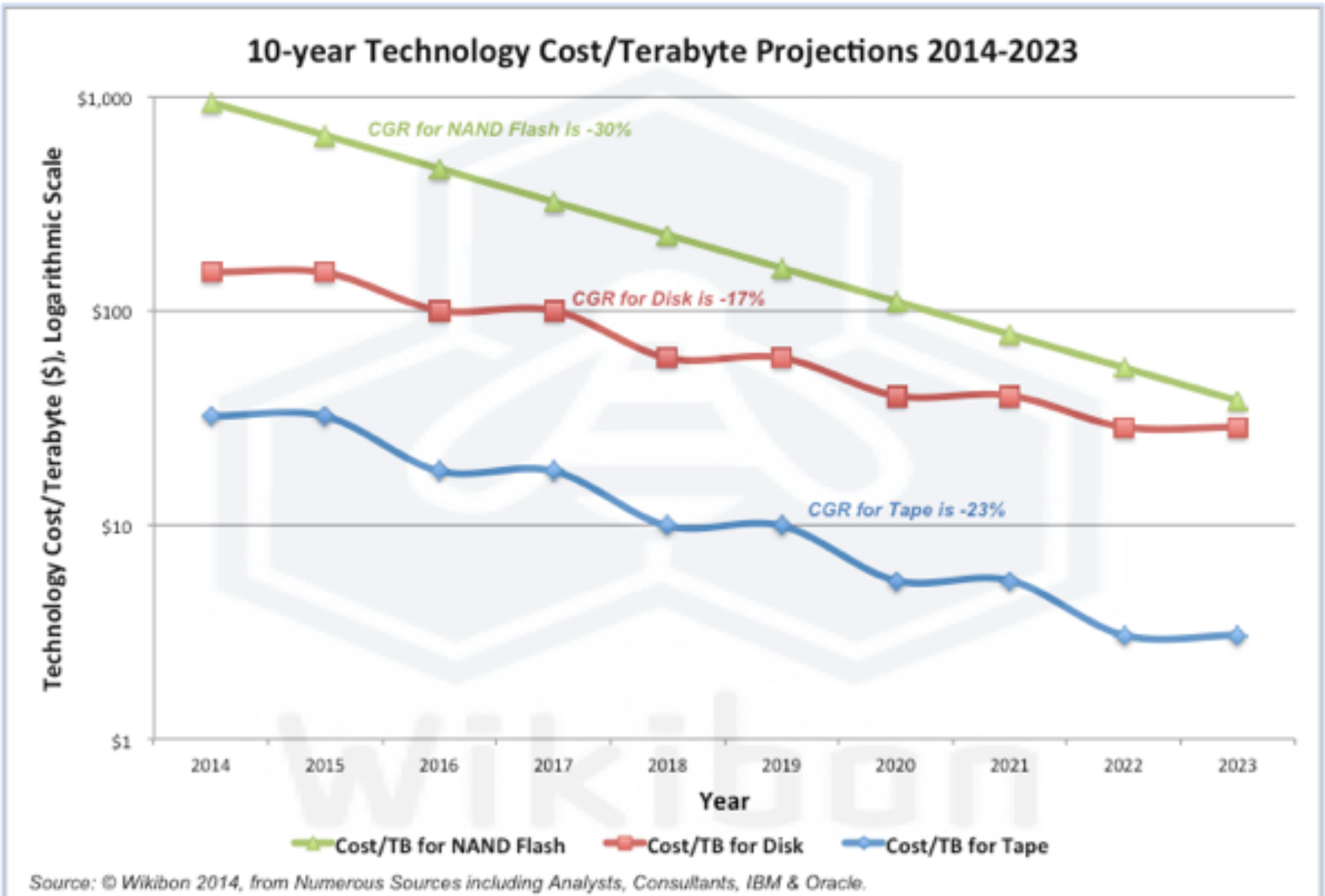
LHC archiving needs - some model calculations

- Beyond 2018?
  - Run 3 (2020-2022): ~150PB/year
  - Run 4 (2023-2029): ~600PB/year
  - Peak rates of ~80GB/s





A Price Prediction

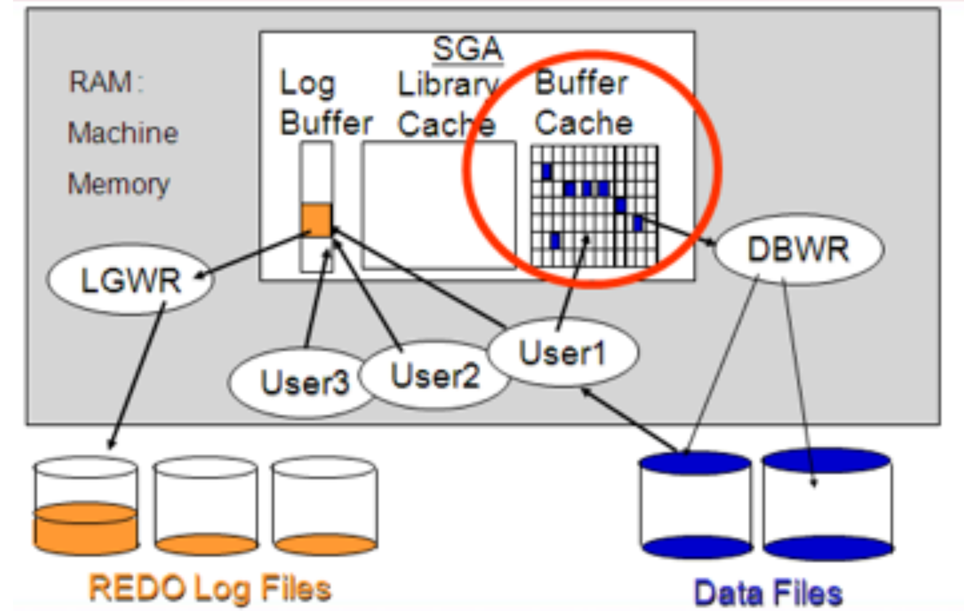


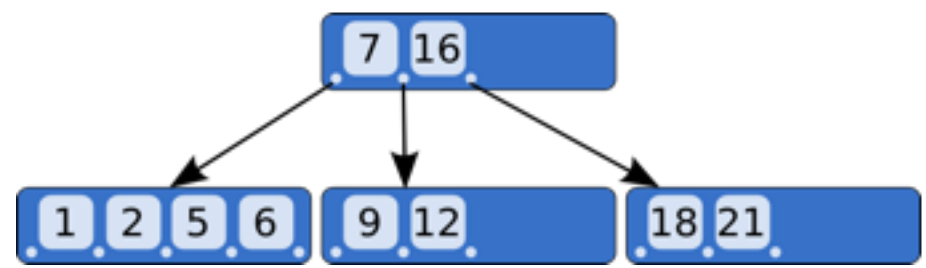
Organising Data

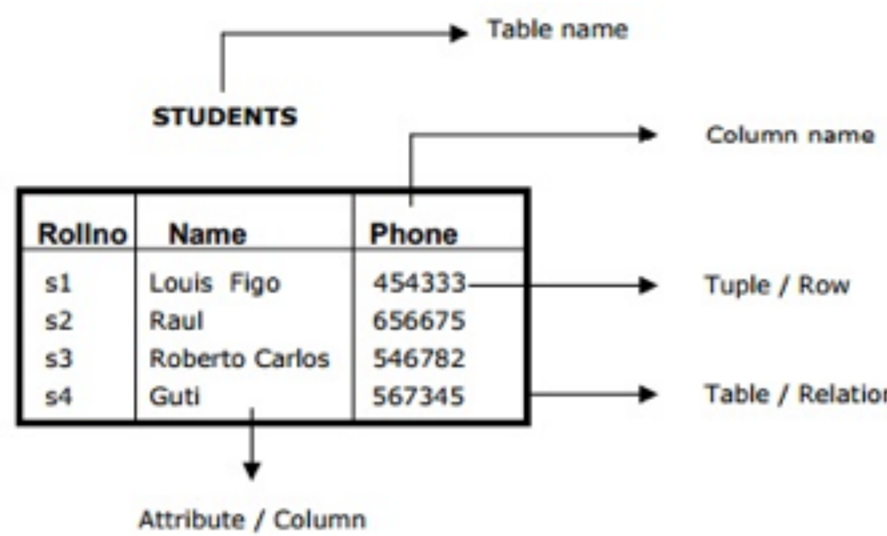
Databases

- Consistent data changes for concurrent users
  - ACID transactions
- Indexed (fast) access to disk-resident data by key
  - eg Bayer-Trees
- Structured Query Language
  - constraint tabular data model
  - or binding to general development language

- All three main functions under increasing pressure from simpler (aka more specialized) solutions
  - ACID scaling & transactional development skills
  - Increased memory availability allows to access data faster than B-Trees
  - Constraints of tabular model for some problems

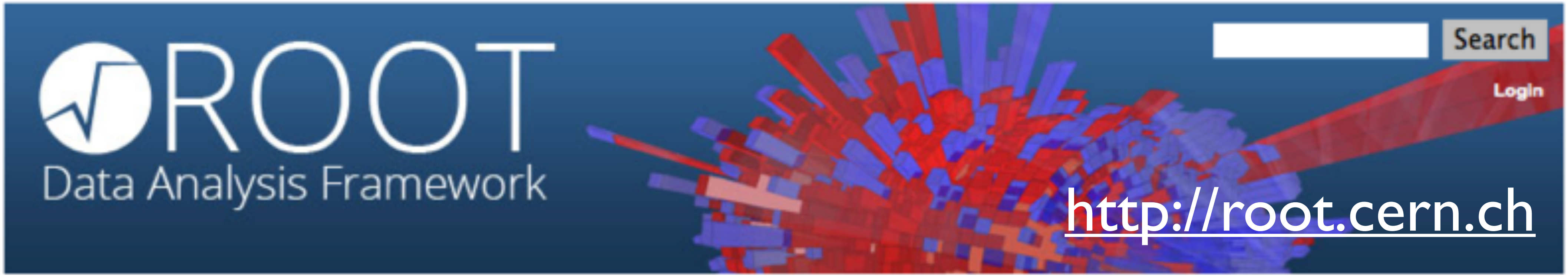






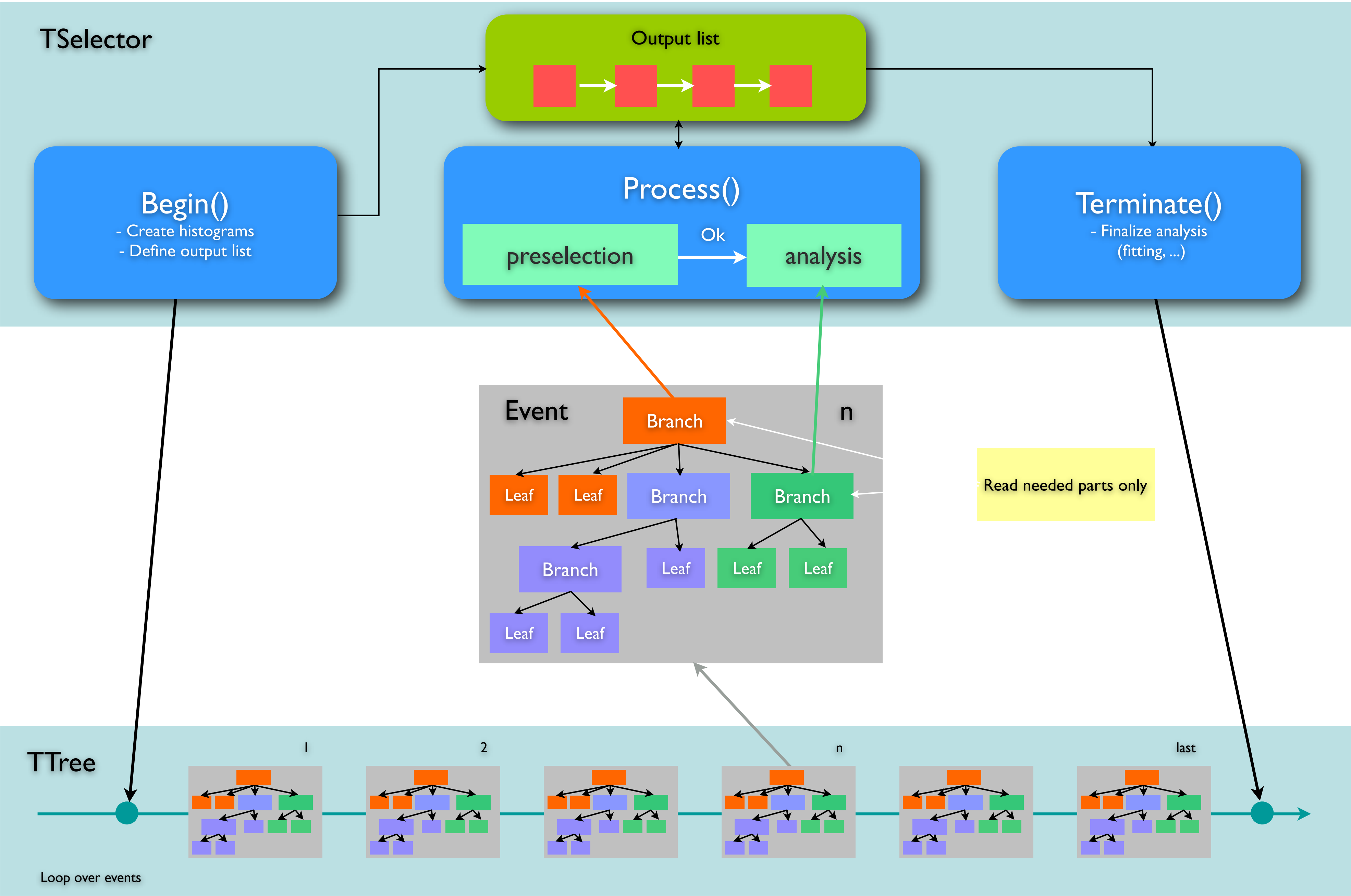
How to store/retrieve LHC data models? A short history…

- 1st Try - All data in an commercial Object Database (1995)
  - good match for complex data model and C++ language integration
  - used at PB scale for BaBar experiment at SLAC
  - but the market predicted by many analysts did not materialise!
- 2nd Try - All data in a relational DB - object relational mapping (1999)
  - Scale of deployment was far for from being proven
  - Users code in C++ and rejected data model definition in SQL
- Hybrid between RDBMS and structured files (from 2001)
  - Relational DBs for transactional management of meta data (TB scale)
    - File/dataset meta data, conditions, calibration, provenance, work flow
    - via DB abstraction (plugins: Oracle, MySQL, SQLite, Frontier/SQUID)
    - see XLDB 2007 talk for details
- Home-grown persistency framework ROOT (180PB today)
  - Uses C++ "introspection" to store/retrieve networks of C++ objects
  - Configurable column-store for efficient sparse reading

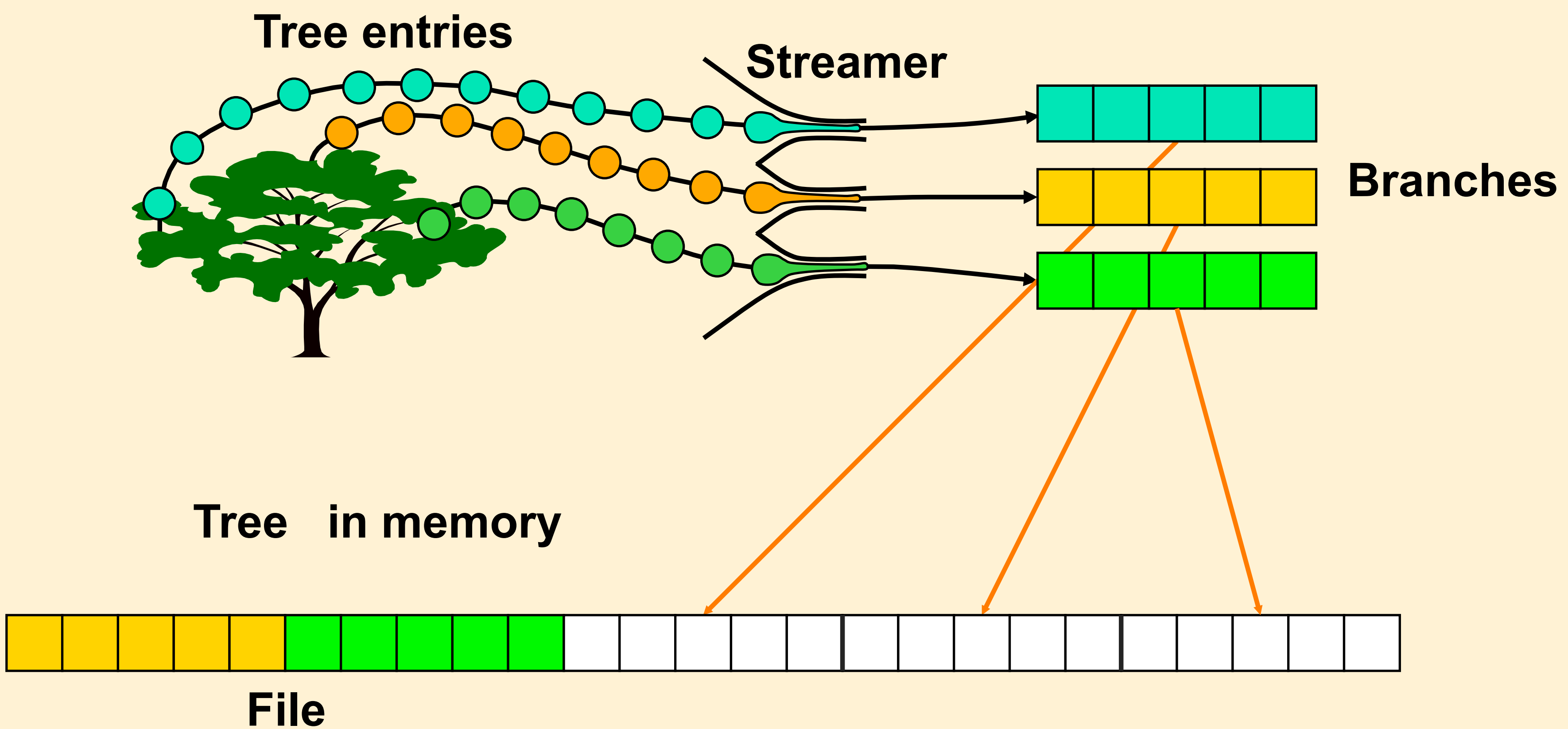


- Scalable, efficient, machine independent format
- Orthogonal to object model
  - Persistency does not dictate object model
- Based on object serialization to a buffer
- Automatic schema evolution (backward and forward compatibility)
- Object versioning
- Compression
- Easily tunable granularity and clustering
- Remote access
  - HTTP, HDFS, Amazon S3, CloudFront and Google Storage
- Self describing file format (stores reflection information)
- ROOT I/O is used to store all LHC data (actually all HEP data)

Processing a TTree

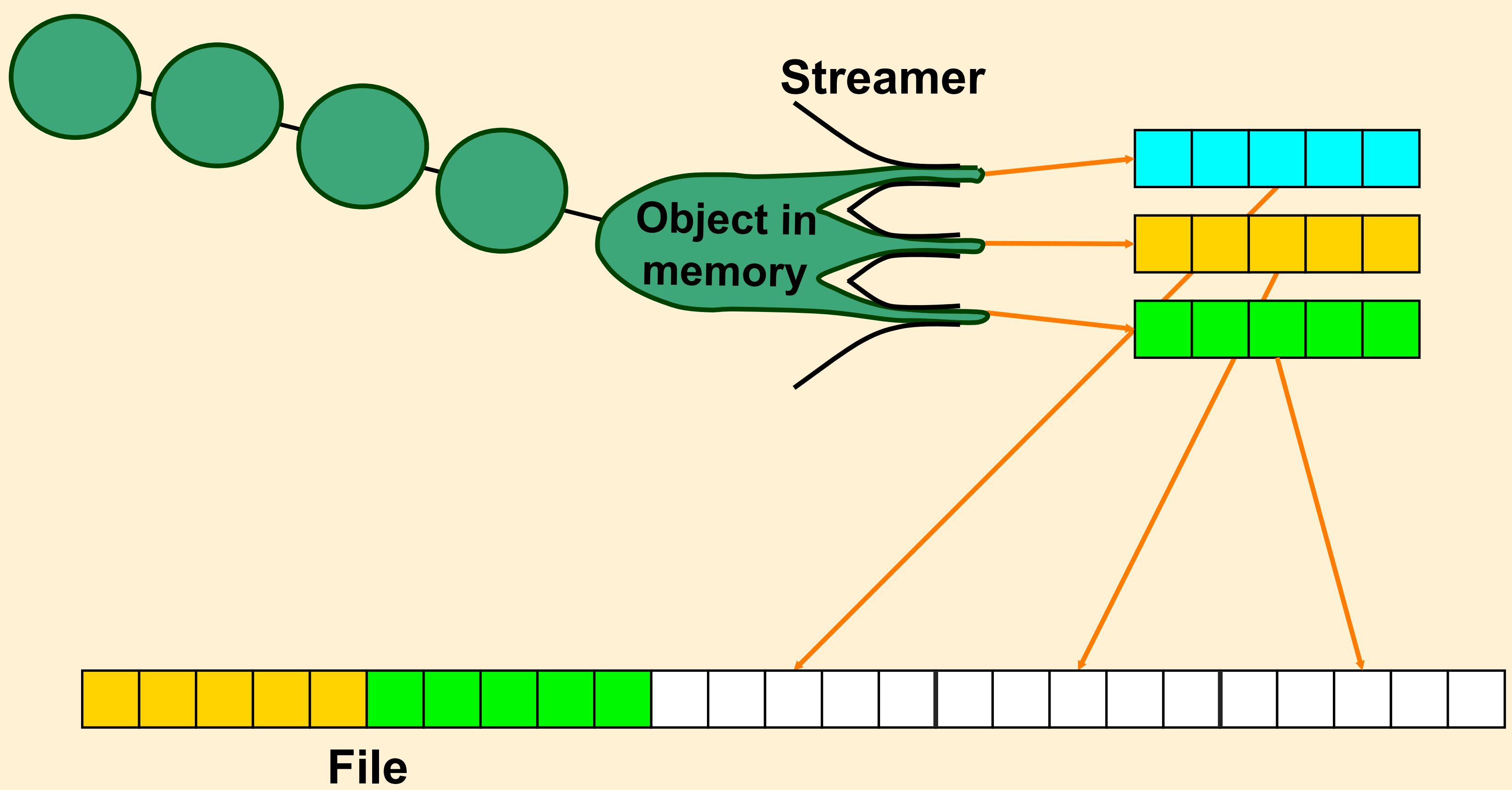


TTree - Clustering per Object



TTree = container for an arbitrary set of independent event trees

TTree - Clustering per Attribute



tuneable: mix of row, column storage is possible within an object tree

Michael Hausenblas - Chief Data Engineer @ MapR

in his bog at https://medium.com/large-scale-data-processing/3da34e59f123

[…]

I was flabbergasted and went like: OMG, there is a group of people who have been doing this for almost 20 years now. While I think the Google engineers deserve the credits for the engineering innovations they introduced in their 2010 paper on Dremel I also believe Fons and his team deserve at least the same attention and credit.

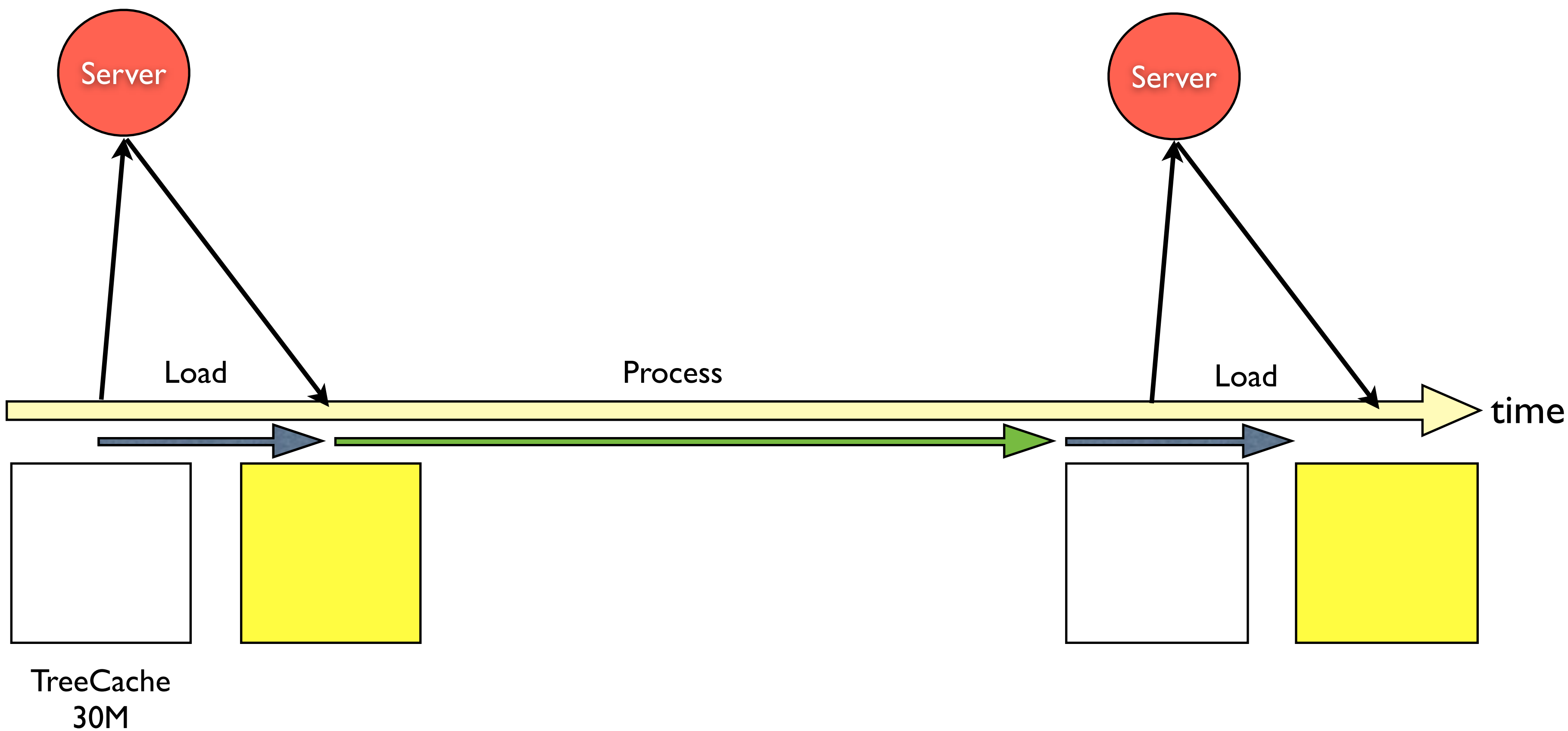
[…]

Data Access Optimisations - Vector Reads and Client Cache

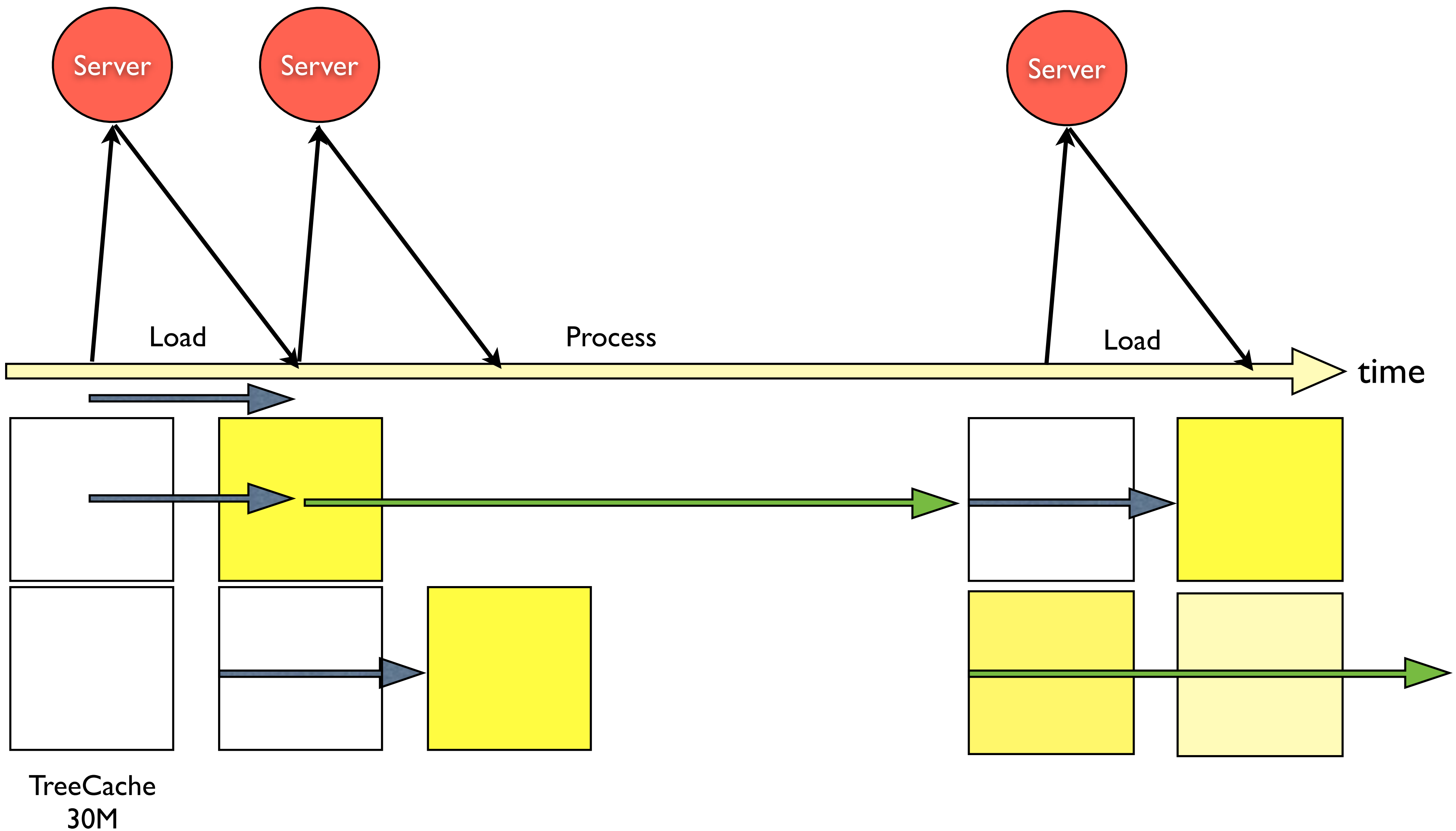
- I/O requests from analysis jobs are often small and not strictly consecutive, due to
  - selective reads
  - reads from several parallel branches within the file
- Consequence: random I/O
  - 1: many network round trips
  - 2: trivial read-ahead is counter-productive, protocol level read-ahead is not adequate
  - 3: often limited by IOPS/s rather than throughput
- Client side cache helps with 2) and 3)
- Vector-reads (eg ROOT TreeCache) with 1)
- Cache provider & user: consistent assumptions!
  - cache logic can/should exploit application knowledge about upcoming data requests => in ROOT

Optimisations for WAN

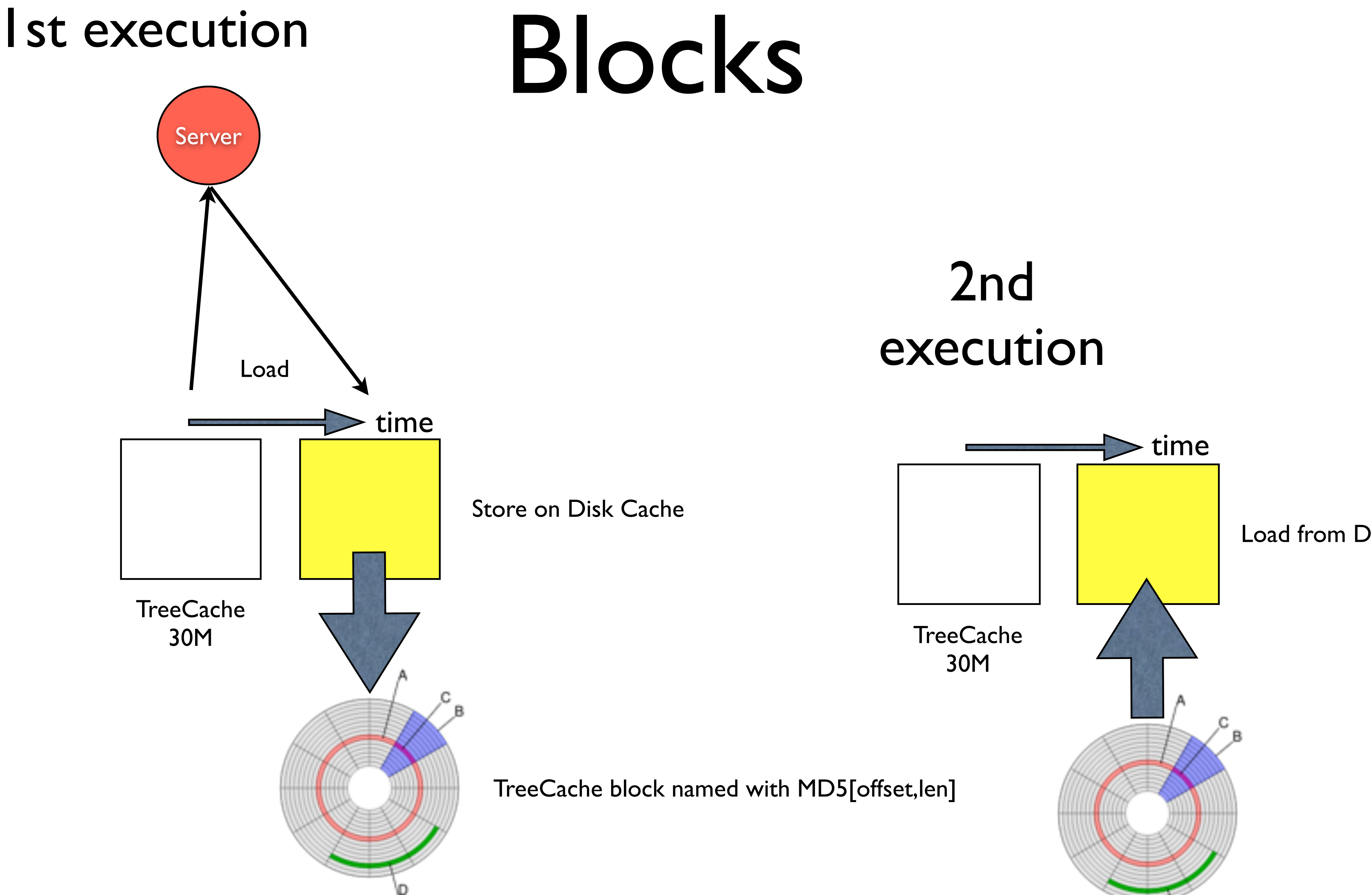
ROOT Tree processing was initially synchronous



Asynchronous Pre-Fetching



Caching of TreeCache Blocks



Data Access Protocols

Authentication, Authorisation, Accounting

- Certificate (X509) scalability vs shared secrets
  - implication for user access protocols: session management
  - scaling implication for service
    - eg: EOS is using shared secrets internally and is moving to scalable authentication front-end nodes
- Site integration
  - mapping to Kerberos and E-groups
- basic ACL's do not always allow to describe required functionality
  - eg allow to create files but not to delete/update
- More complex ACL's need education and regular checks to spot unintended use

XRootD

Collaboration between SLAC, CERN, Duke Univ., UCSD and JINR

1) Framework for scalable storage servers

- modular & mature code base with origins back in 90’
- used heavily & extended for very large installations in EOS

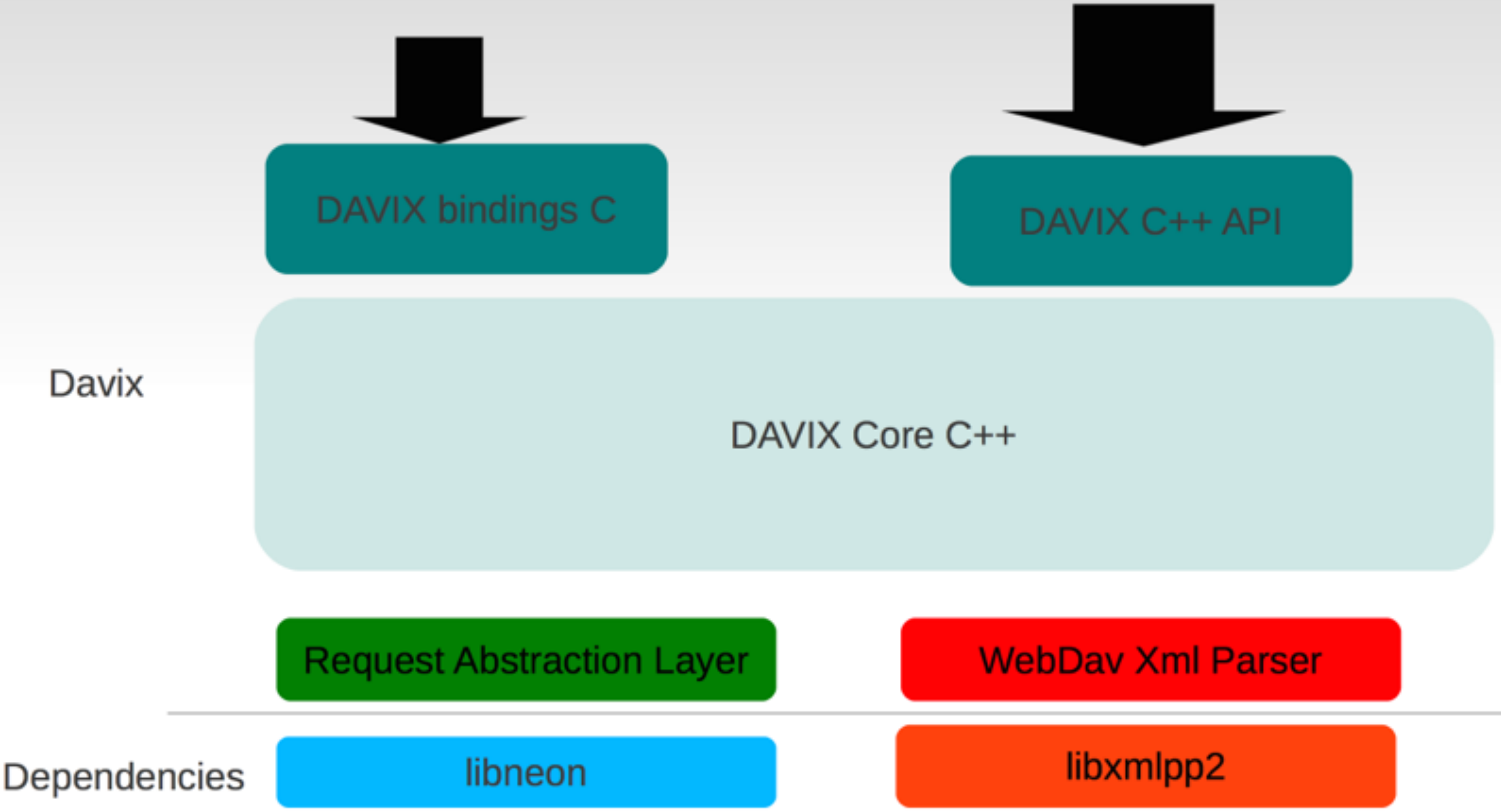
2) Rich storage access protocol

- session, management redirection, pluggable authentication, support for multi-threading etc..
- widely used in HEP community
- recently extended with http protocol translation

HTTP, WebDAV and S3

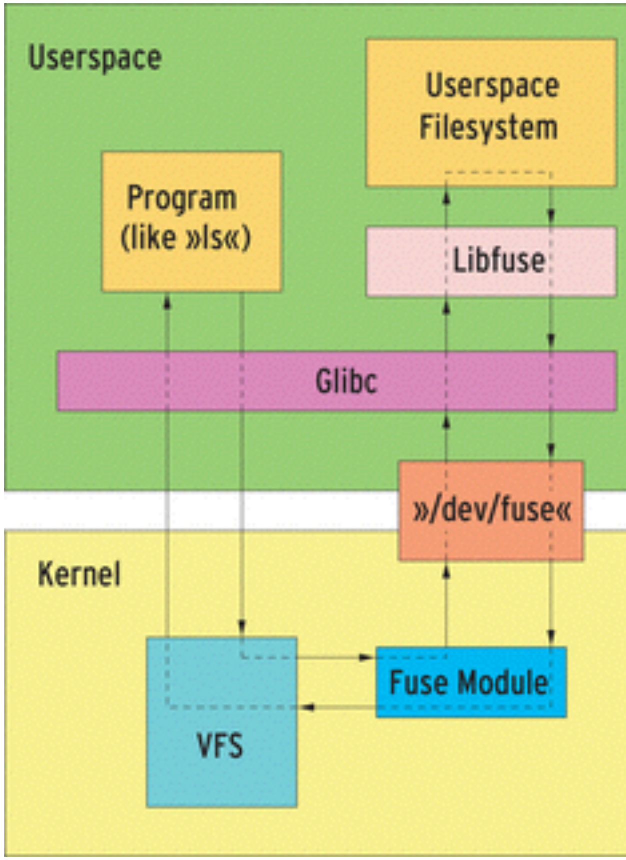
- Goal
  - can we profit more from the technology stack developed for large scale commercial deployments?
  - can we provide services to other communities based on these protocols?
- Challenge
  - semantics for non-trivial operations (other than put, get) turns out to vary between different available tools
  - redirection semantics, authentication/authorisation with (our) X509 infrastructure still requires HEP specific modifications
- Promising developments
  - basic I/O performance has been shown to reach similar performance
  - Davix as a quite complete http/webdav/S3 client is available as reference implementation

Davix Architecture :



Fuse and NFS

- Goal
  - can we access all files as a “normal” local files
  - eg from applications w/o support I/O protocol plugins
  - eg communities w/o control over the source code of their apps
- Fuse - client side file system plugin translating to other network protocols
  - eg XRoot, ssh, webdav
- Challenge
  - going through a POSIX interface on the client side may loose the application knowledge about what data is likely to be read next (vector-read)
- NFS - well defined semantics and commercial backing from larger storage companies
  - dCache and DPM implementations
  - not a lot of traction yet inside the HEP community



Organising Files

"...the results of countless computations can be kept "on file" and taken out again. Such a "file" now exists in a "memory" tube developed at RCA Laboratories. Electronically it retains figures fed into calculating machines, holds them in storage while it memorizes new ones - speeds intelligent solutions through mazes of mathematics."

RCA (Radio Corporation of America) advertisement in 1950

File Names and other Meta Data

- Early on, POSIX has standardised the semantics of file systems
  - interoperability for all applications
- Hierarchical namespace
  - to organise larger numbers of files in directories
- What meta data is kept by the system automatically
  - eg: size, owner, access time, modification time
- Access semantics
  - eg what happens when two processes access the same file
  - what meta data is used for grating access, calculating quota etc.

Posix semantics and scalability

- modification & access time
- rename & re-write & append
- concurrent modifications & locking

- Result:
  - many larger scale systems today break POSIX - in slightly different ways…
  - Interoperability was lost again.

The prefix "meta": always in for some confusion?

The word "**metaphysics**" derives from the Greek words μετά (*metá*, "beyond", "upon" or "after") and φυσικά (*physiká*, "physics").[7] […] The editor of Aristotle's works, Andronicus of Rhodes, is thought to have placed the books on first philosophy right after another work, *Physics*, and called them τὰ μετὰ τὰ φυσικὰ βιβλία (*ta meta ta physika biblia*) or "the books that come after the [books on] physics". This was misread by Latin scholiasts, who thought it meant "the science of what is beyond the physical".

Metadata is "data about data". The term is ambiguous, as it is used for two fundamentally different concepts (types). **Structural metadata** is about the design and specification of data structures and is more properly called "data about the containers of data"; **descriptive metadata**, on the other hand, is about individual instances of application data, the data content.

source: Wikipedia

Access by hierarchical position or by description?

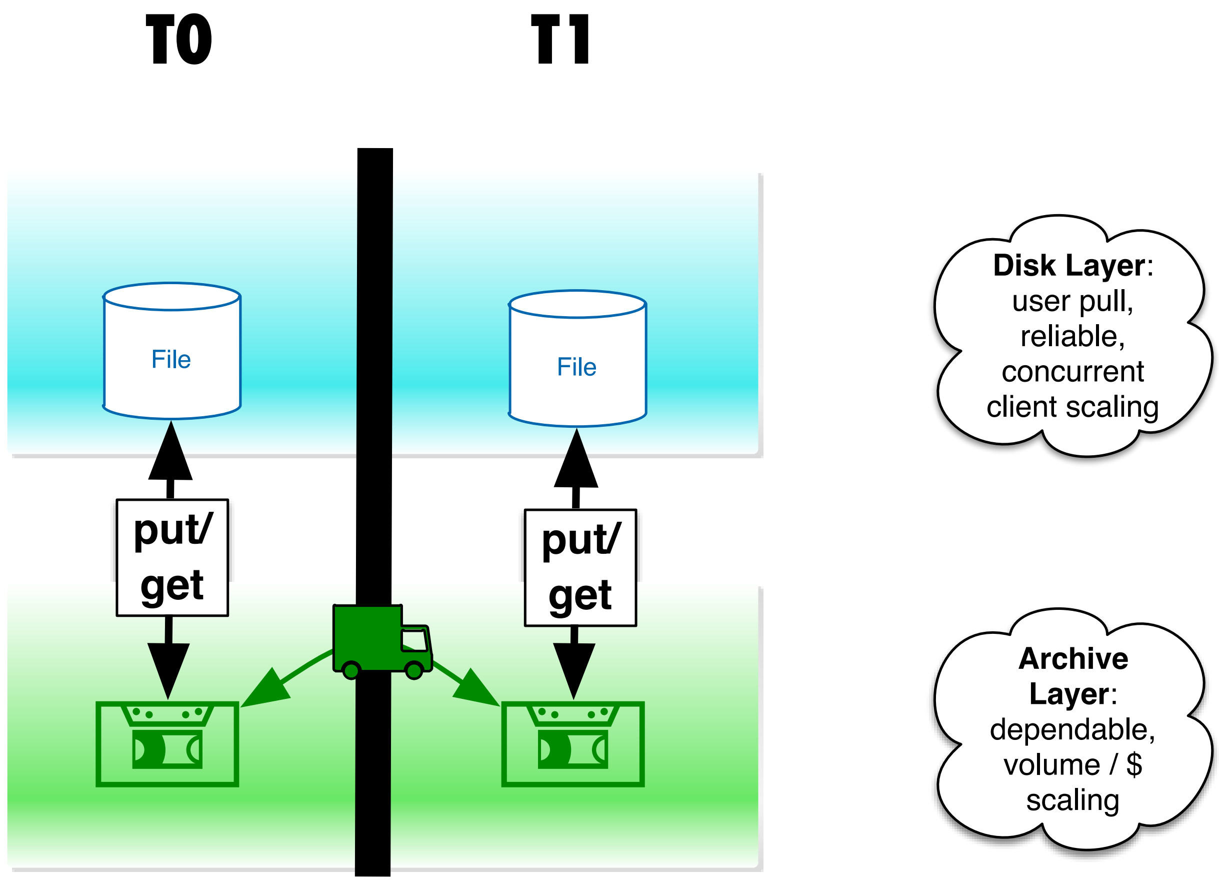
- Domain specific meta data
  - is different for each domain
- Searching + Tagging
  - Email, mp3, photos: folders vs search
  - URLs - bookmarks vs google
- Essentially
  - underlying file names become irrelevant
  - access is done via an meta-data overlay structure which implements popular search use cases
  - similar systems in-place/planned for LHC
  - Impact: the (file) storage system looses information about the context (remember the loss of geometry info on ancient disks?)
- In large systems it will be beneficial to tie the overlay more closely to data layer

Grids, Clouds and beyond…

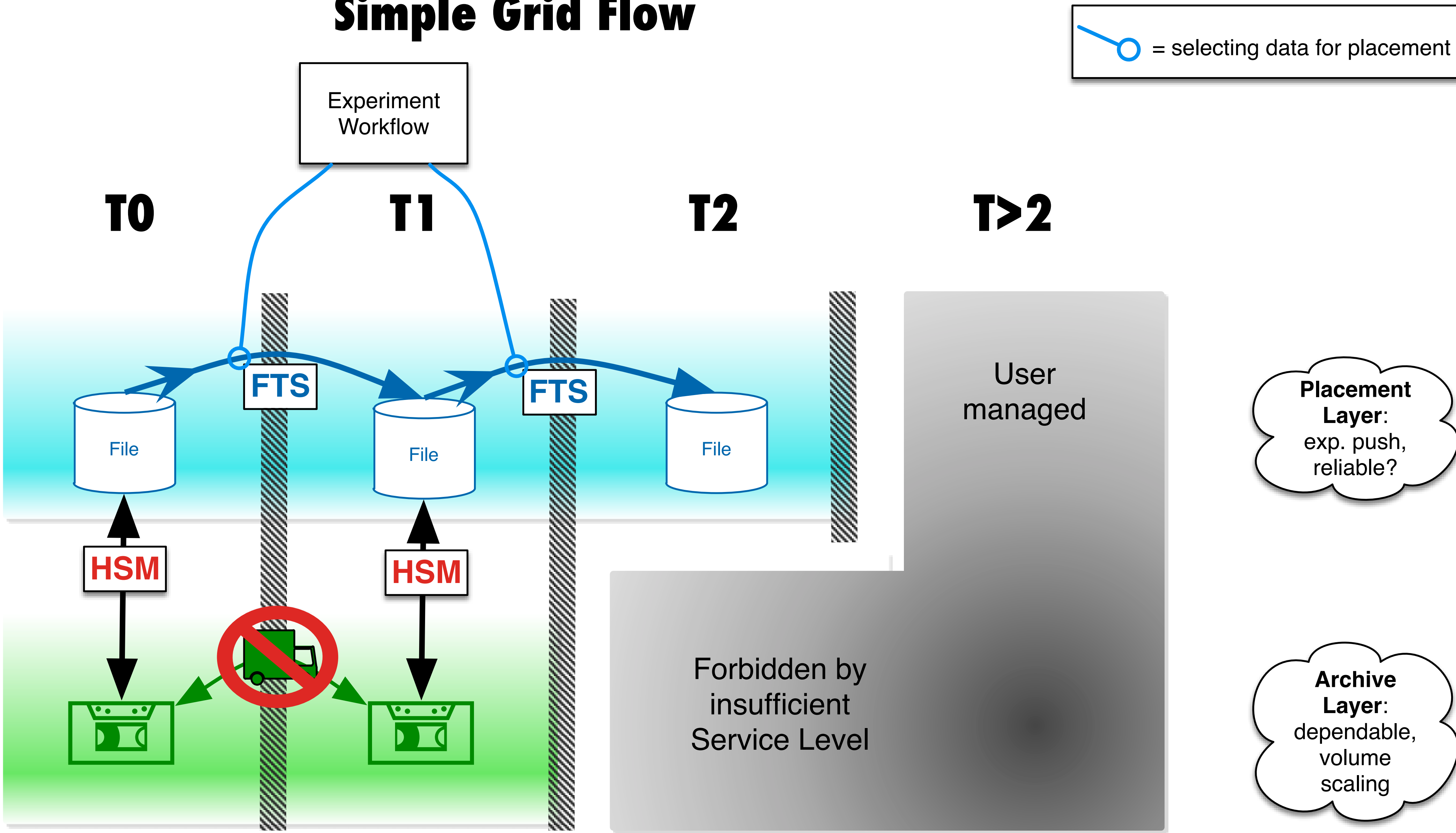
Thermodynamic facts can often be explained by viewing macroscopic objects as assemblies of very many microscopic or atomic objects that obey Hamiltonian dynamics. The microscopic or atomic objects exist in species, the objects of each species being all alike. Because of this likeness, statistical methods can be used to account for the macroscopic properties of the thermodynamic system in terms of the properties of the microscopic species.

http://www.mongodb-is-web-scale.com

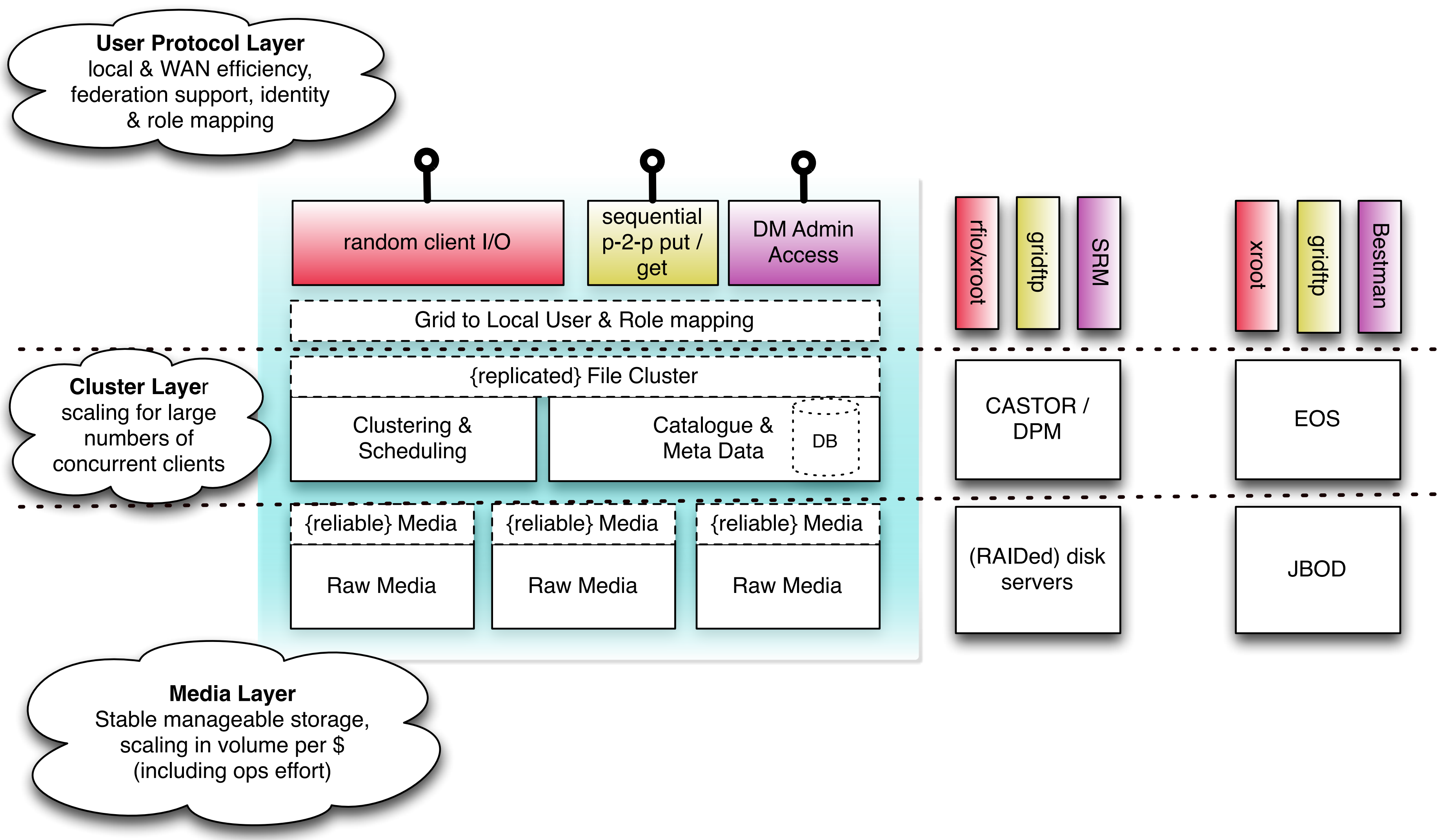
Once upon a time..



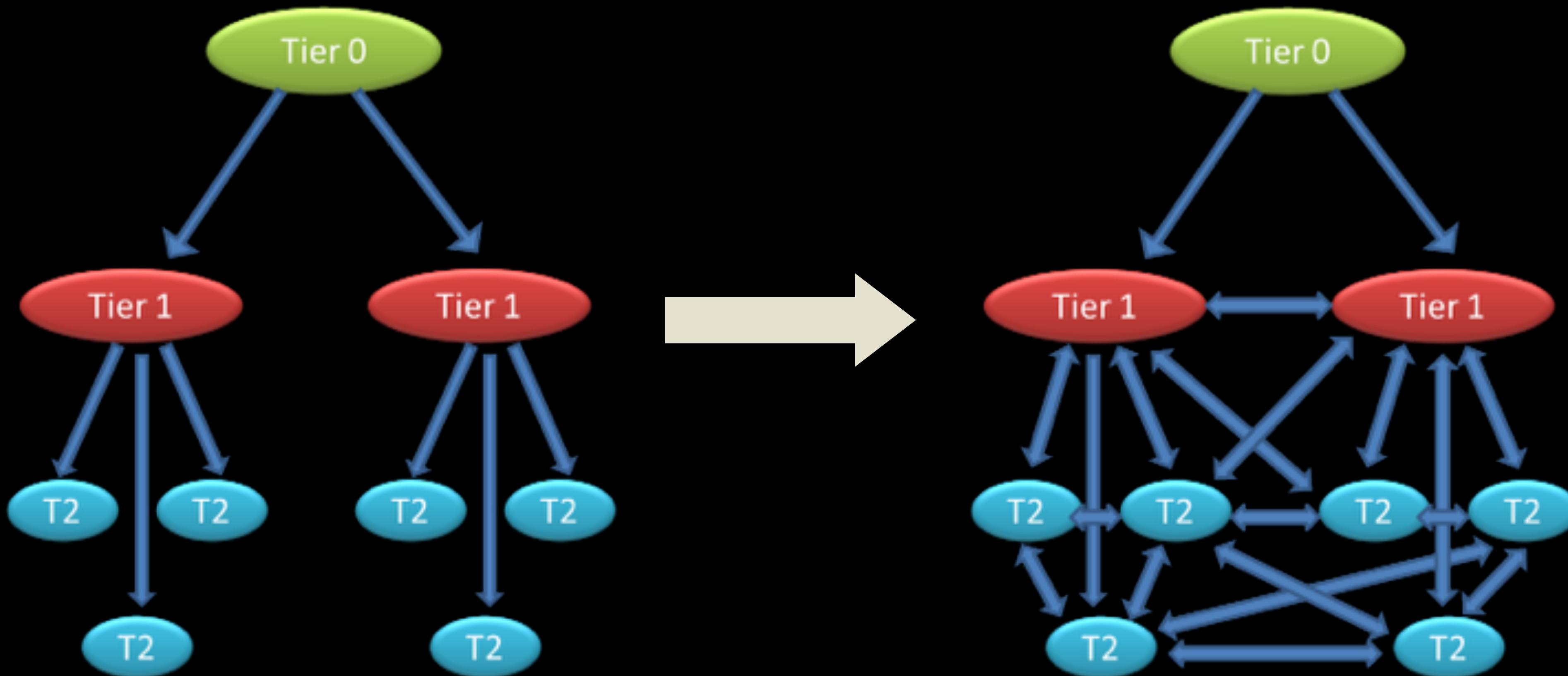
Simple Grid Flow



Structure of a Grid Storage Element



Computing model evolution



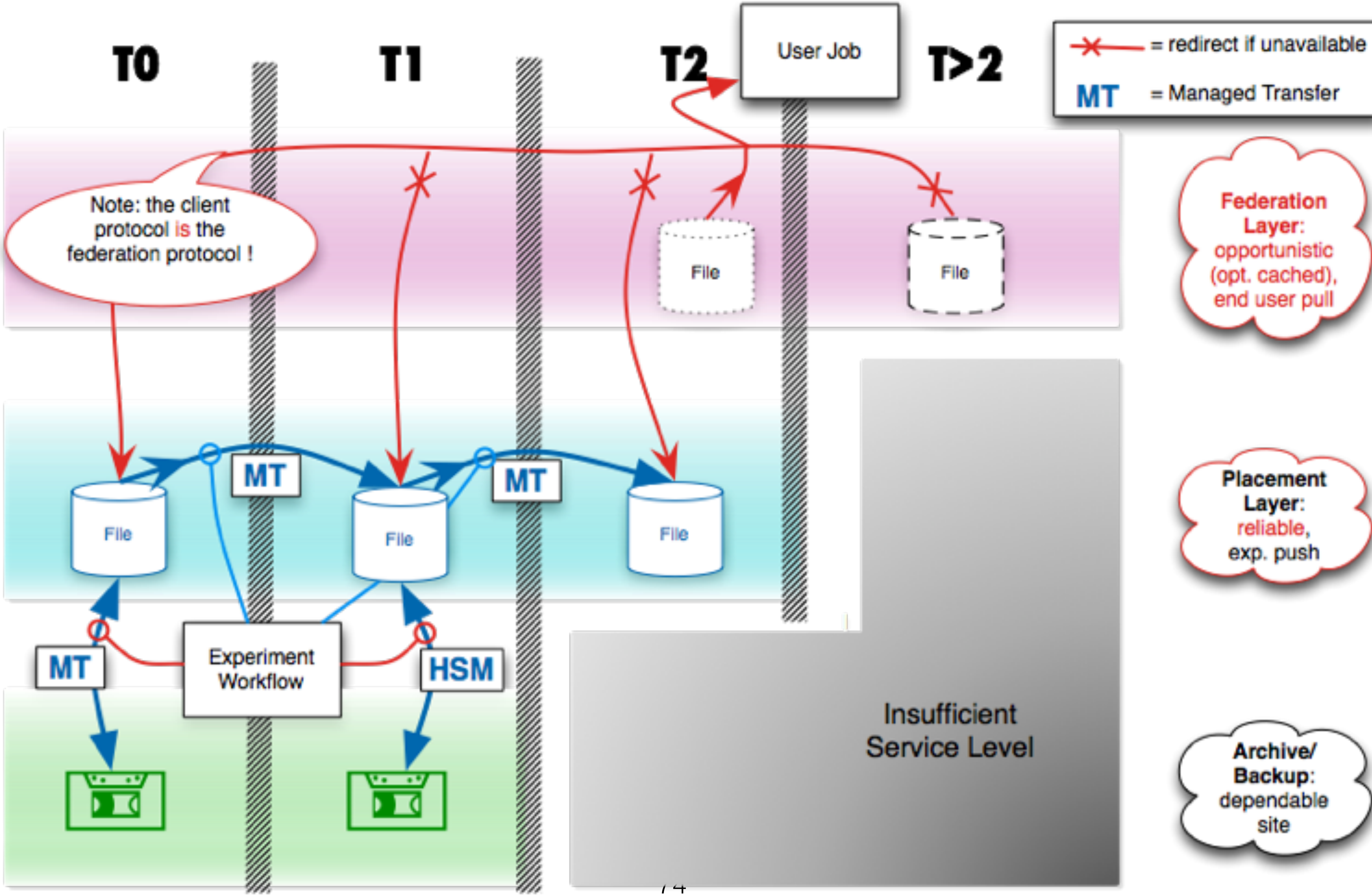
Strict Hierarchy (Control)

Mesh (additional Flexibility)

Federating Data

- Goal: hide internal complexity of the storage system from
  - science end user
  - {experiment data support team}

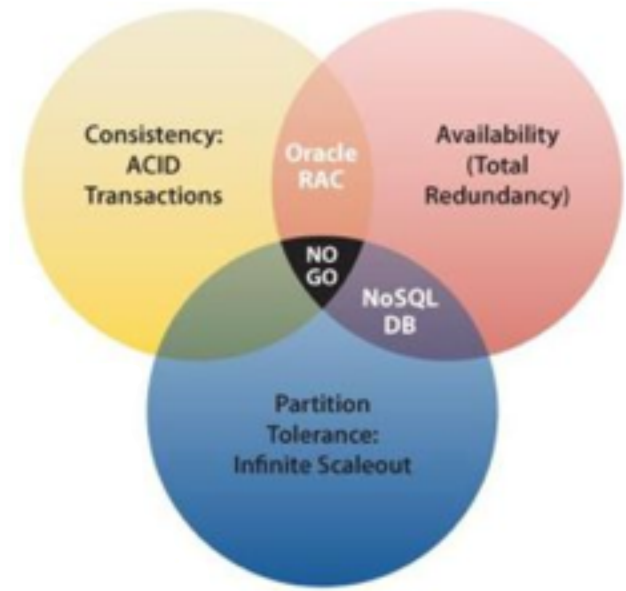
Data Placement and Federation



Cloud Storage

CAP Theorem

- The CAP theorem (Brewer, 2000) states that any networked shared-data system can have at most two of three desirable properties
  - consistency (C) equivalent to having a single up-to-date copy of the data
  - high availability (A) of that data (for updates)
  - tolerance to network partitions (P)
- "two of three" should rather be seen as exclusion of all three at the same time
  - This means
    - distributed ACID databases can not exist
    - but eventual consistency can



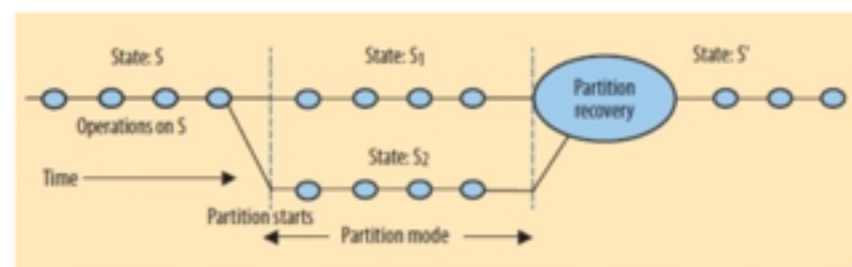
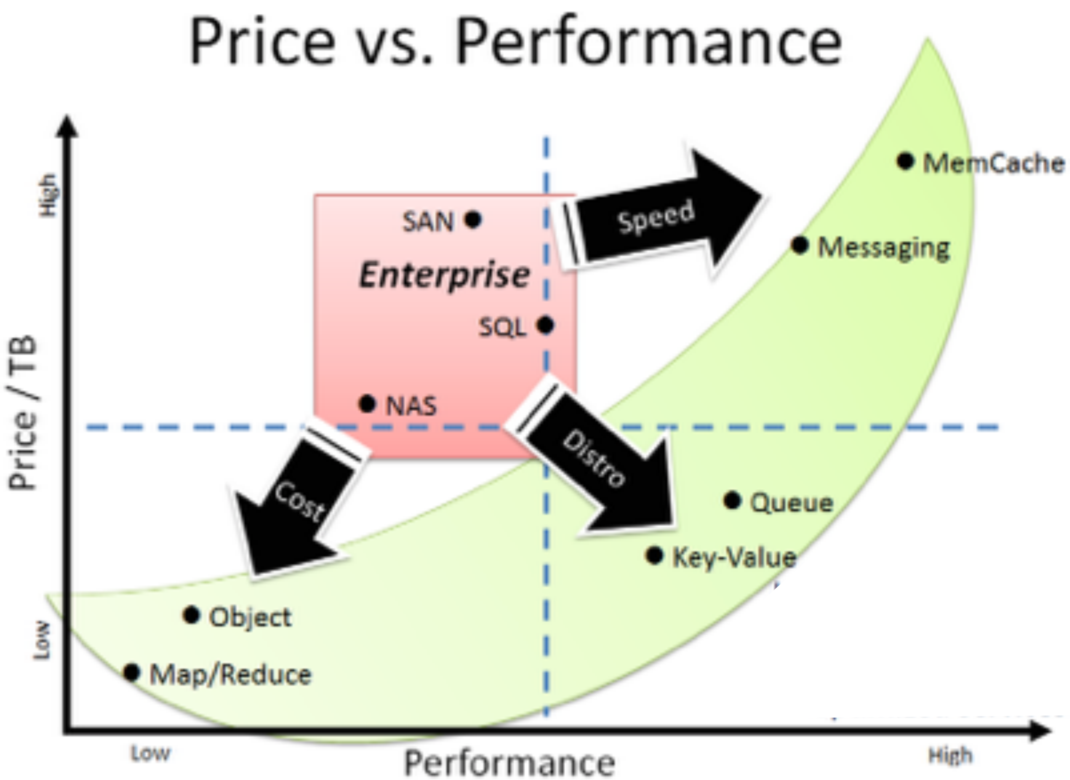


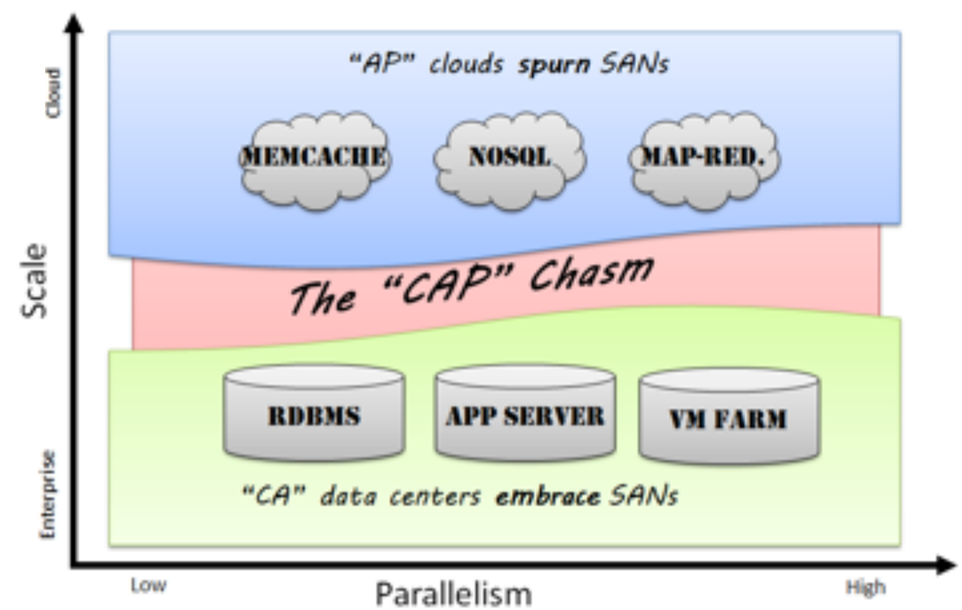
Figure 1. The state starts out consistent and remains so until a partition starts. To stay available, both sides enter partition mode and continue to execute operations, creating concurrent states S_1 and S_2, which are inconsistent. When the partition ends, the truth becomes clear and partition recovery starts. During recovery, the system merges S_1 and S_2 into a consistent state S' and also compensates for any mistakes made during the partition.

Cloud Storage



Cloud storage breaks one-size-fits-all model into optimized services

1. Legacy applications tried to eliminate faults to achieve Consistency with **physically redundant scale up** designs.
2. Cloud applications assume faults to achieve Partitioning Tolerance with **logically redundant scale out** design.



source: http://robhirschfeld.com/category/development/cap-theorem/

S3 is not Posix

- **S3 is focused on simple file storage and transfer**
  - Posix was designed for a single machine
- **Amazon S3 has 1T objects**
  - 50x larger than largest NFS
- **No seek/write**
  - No updates. Change a byte, write the whole file or DB
- **Tail: no**
- **Partitions: Buckets**
- **Linked files: Not present**
- **Directory: Simulated**
- **atime: Not present**
- **Rest API focused on reading/writing objects**

NFS Version 3 Operation	SPECsfs2008
LOOKUP	24%
READ	18%
WRITE	10%
GETATTR	26%
READLINK	1%
READDIR	1%
CREATE	1%
REMOVE	1%
FSSTAT	1%
SETATTR	4%
READDIRPLUS	2%
ACCESS	11%
COMMIT	NA

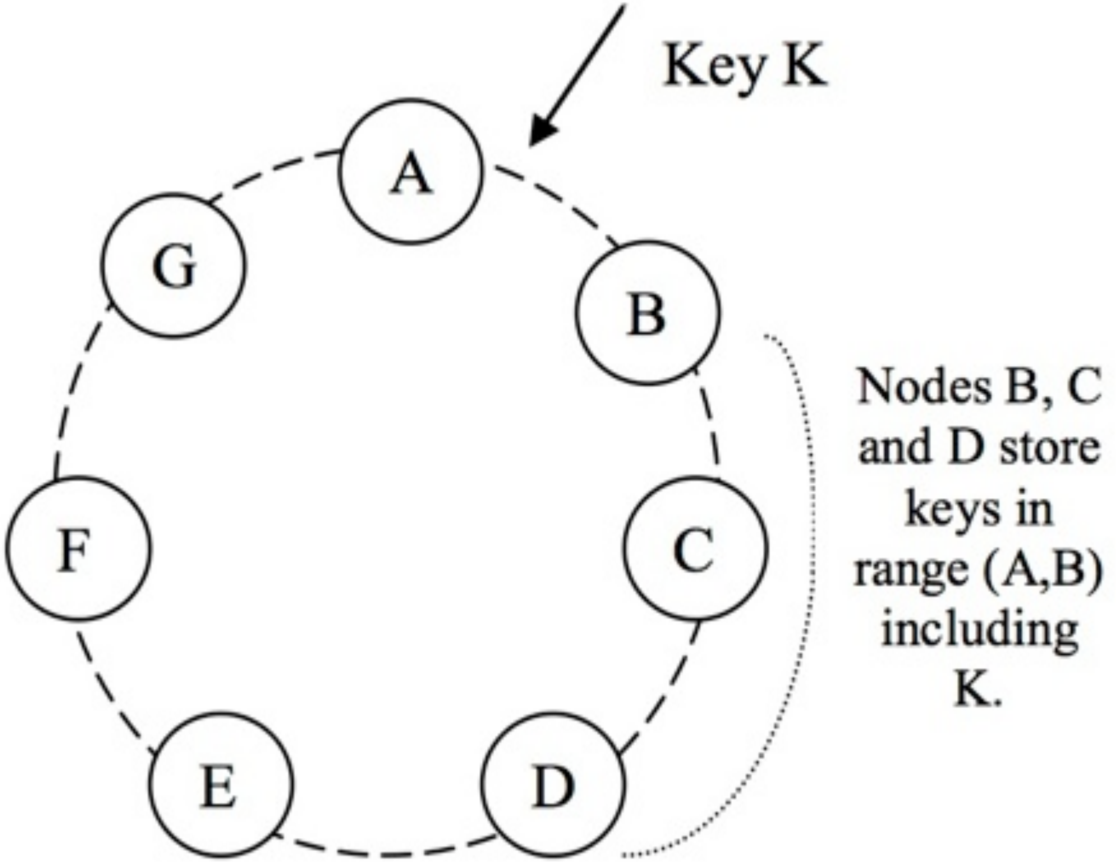
source: James Hughes, CERN computing seminar

Amazon Dynamo - Distributed Hash Tables

Simple API

- **Sharded by hash of the Key**
  - Data = get (Key)
  - put (Key, Data)
  - delete (Key)

Dynamo Concept

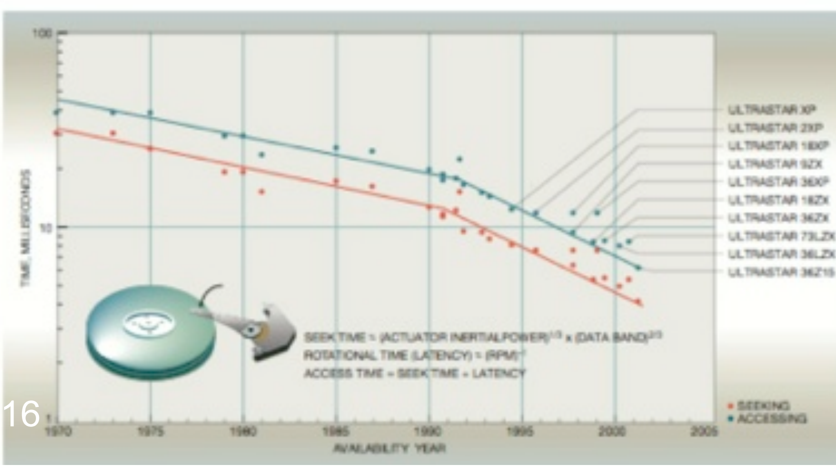


source: James Hughes, CERN computing seminar

Can we make it inexpensive?

- **Disk is the lowest cost storage at this time**
- **Consumer or Enterprise drives? Customer's choice**
  - Enterprise drives are 200% Cost, 133% performance
  - Google and CMU measured reliability as equal
- **Disk drive performance has not increased over the years**
  - 10^1 reduction in performance in 30 years
  - 10^6 increase in processor speed over 30 years
- **Can we use simple cell phone processors?**
  - Distributed RAM and Flash
- **Strict 1:1 reliability**
  - One disk, one processor, **One failure mode**
- **Fail in place**

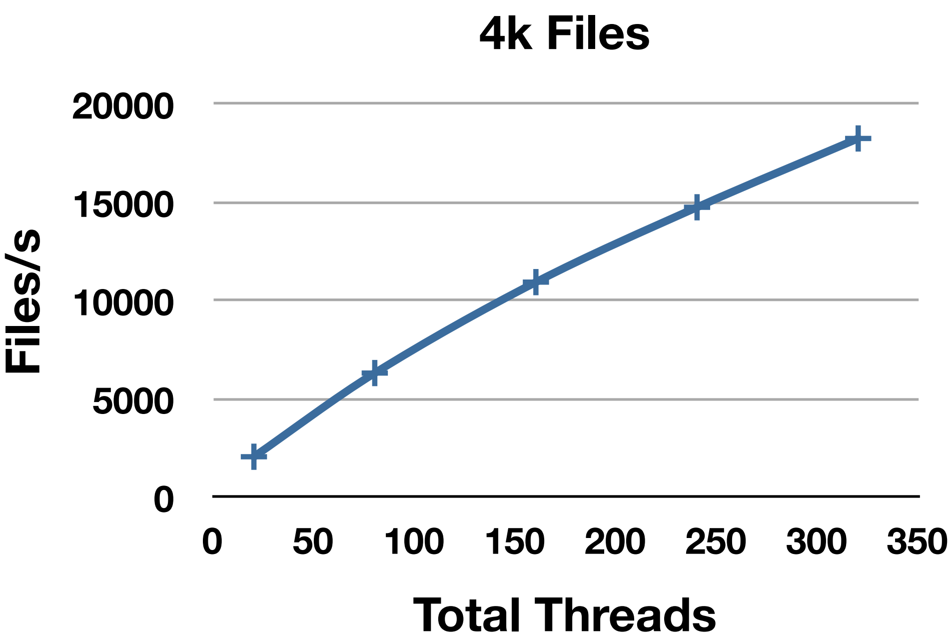
Figure 11 Disk drive access/seek times

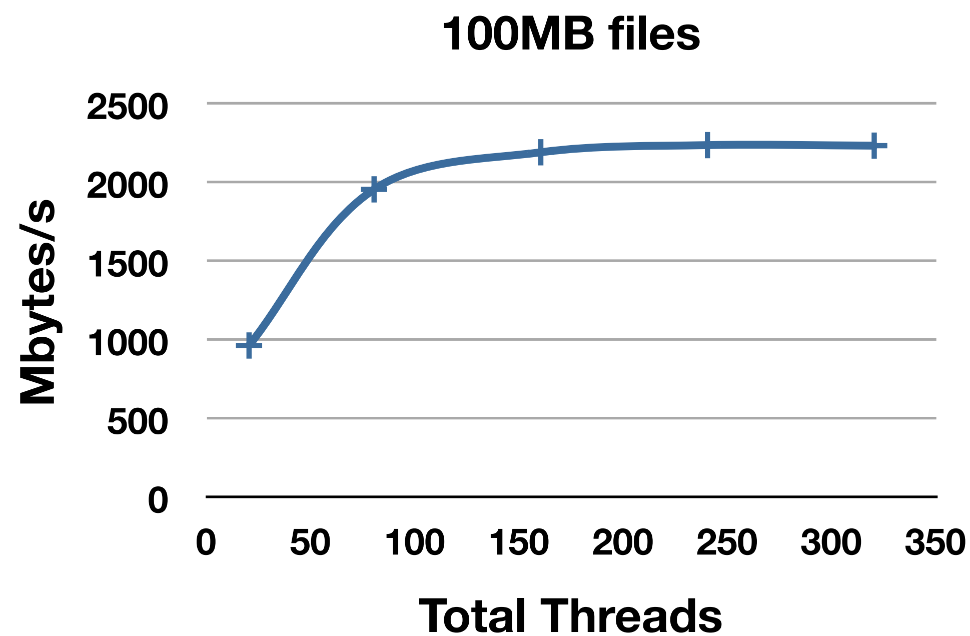


source: James Hughes, CERN computing seminar

Performance Results from CERN

- **Aggregate performance, 760TB system, 20 clients**



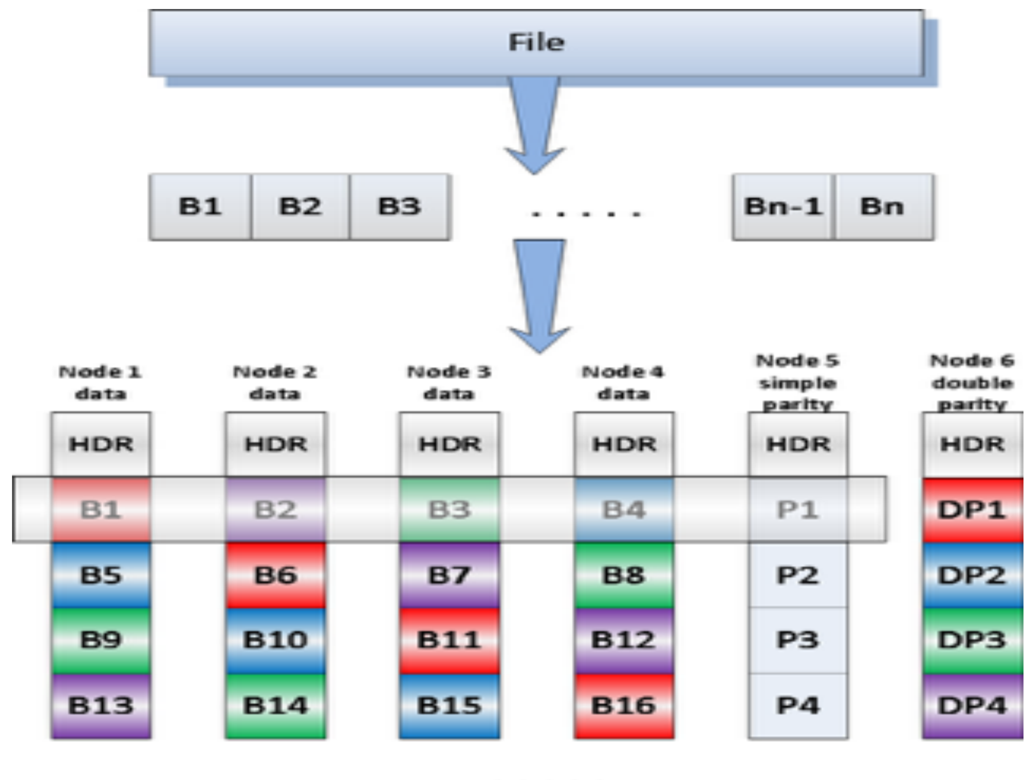


source: James Hughes, CERN computing seminar

On the test-bench

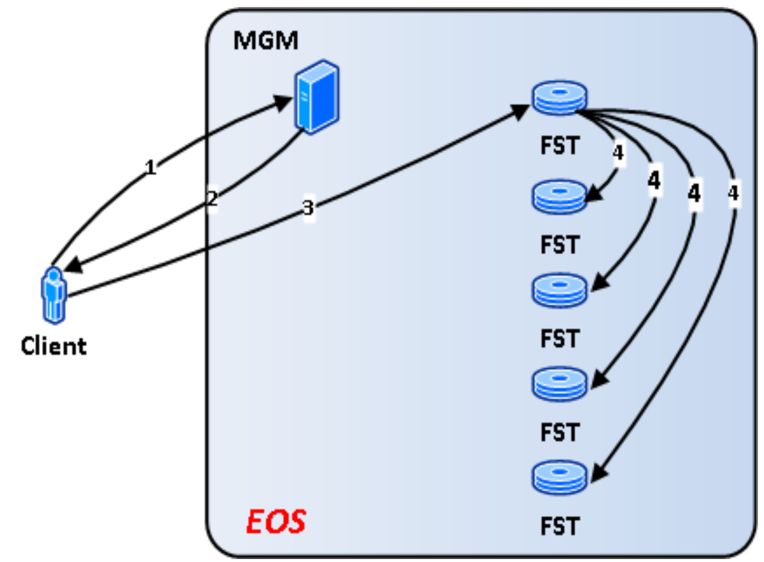
RAIN Storage

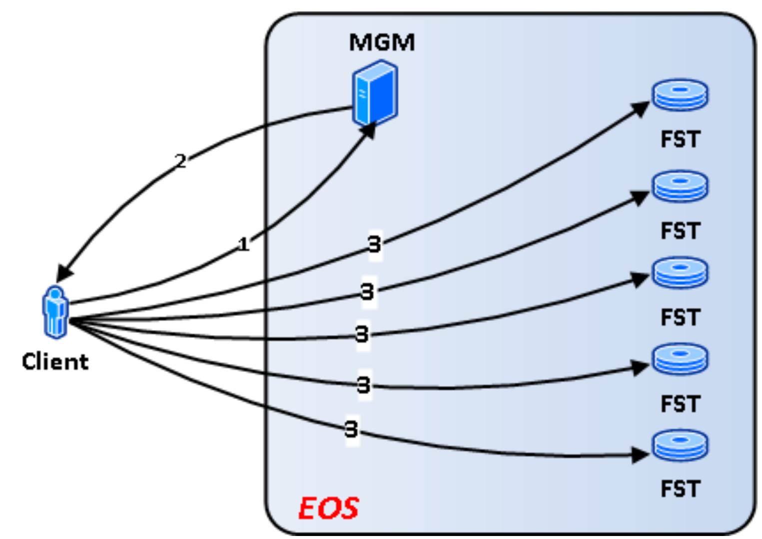
- Redundant Array of Inexpensive Nodes
  (Caltech/NASA-JPL 2001, Dell 2004)
- Aggregating on network node- instead of disk-level



RAIN Reading Modes

- Gateway mode
  - one of the storage nodes performs aggregation
  - potential bottleneck for throughput
- Parallel mode
  - client performs aggregation
  - larger number of client to storage connections





POW Slides

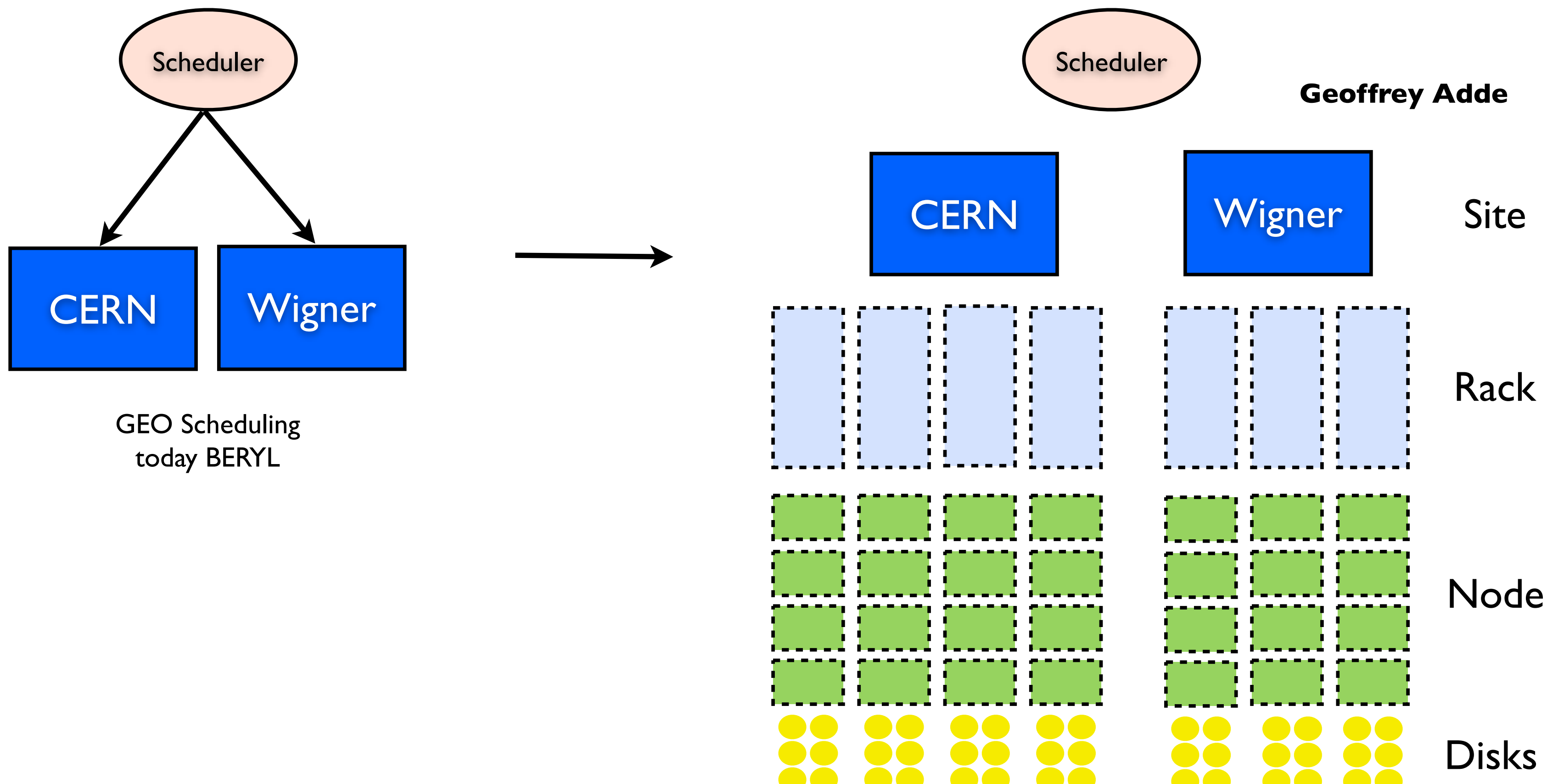
midterm

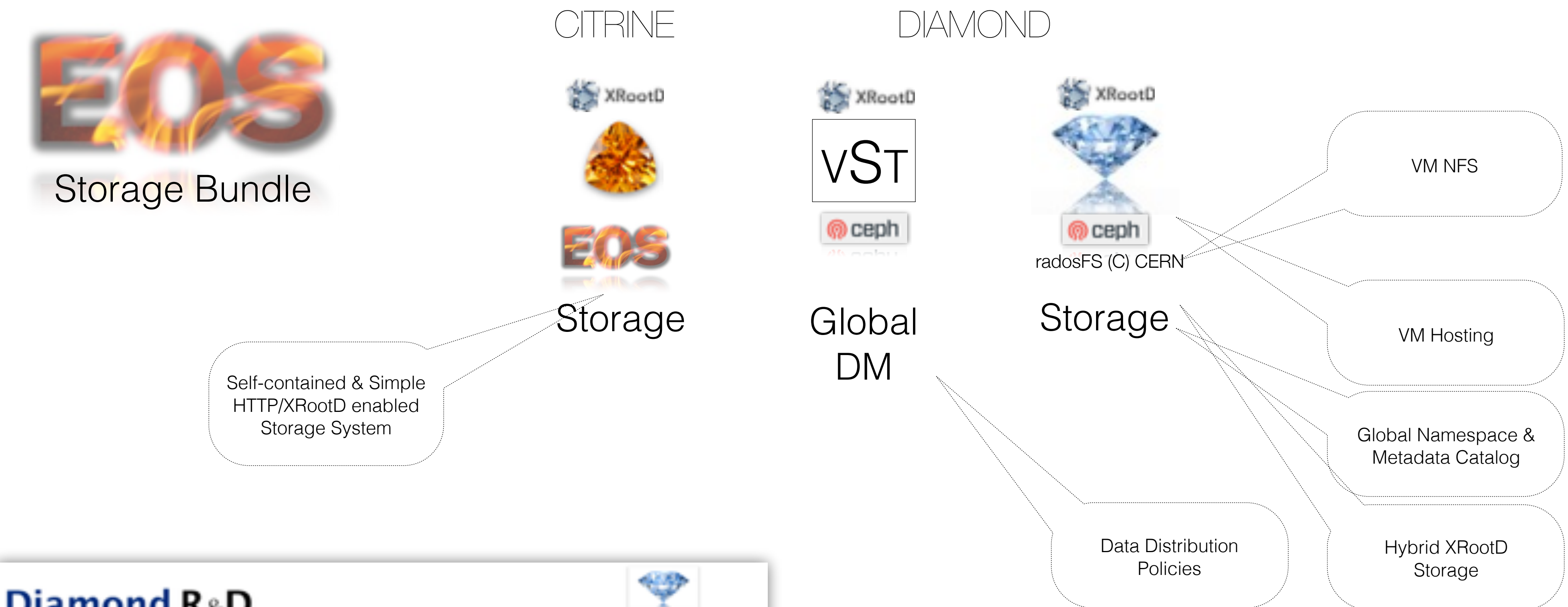
1 year

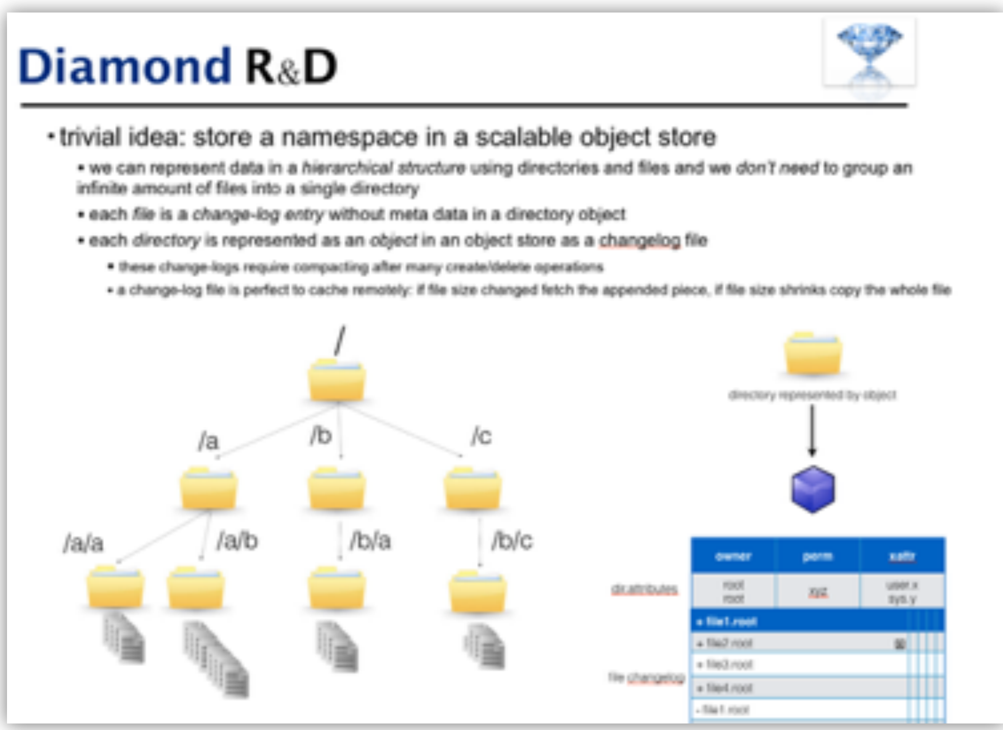
EOS CITRINE

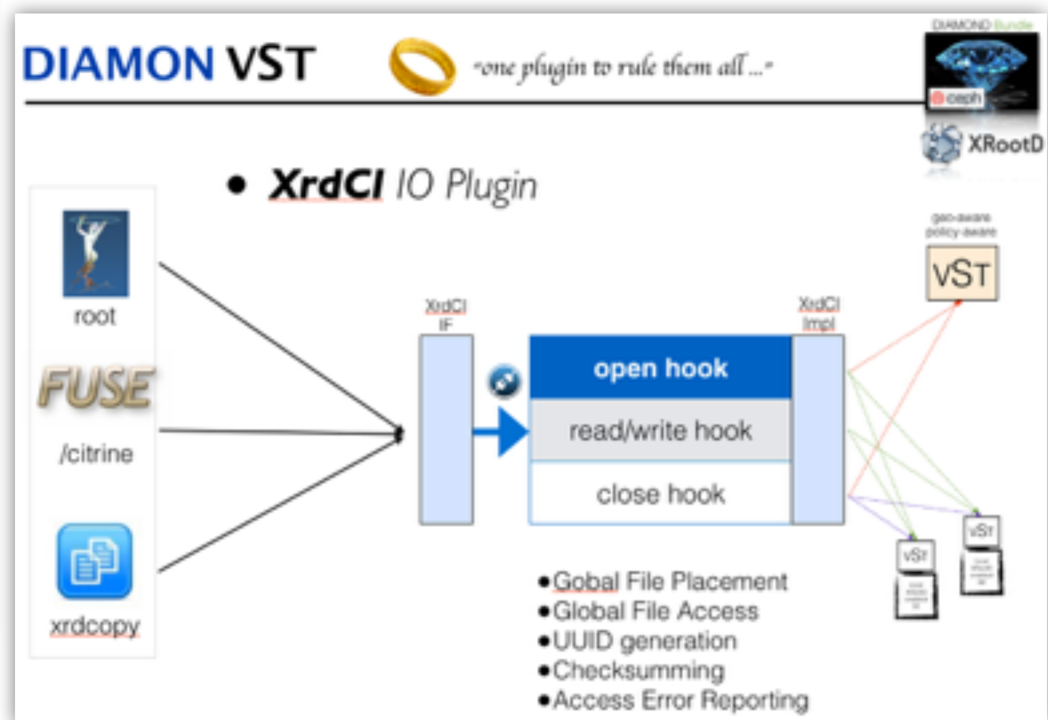
"reduce failure modes & improve data access efficiency"

- Topology-aware placement & scheduling



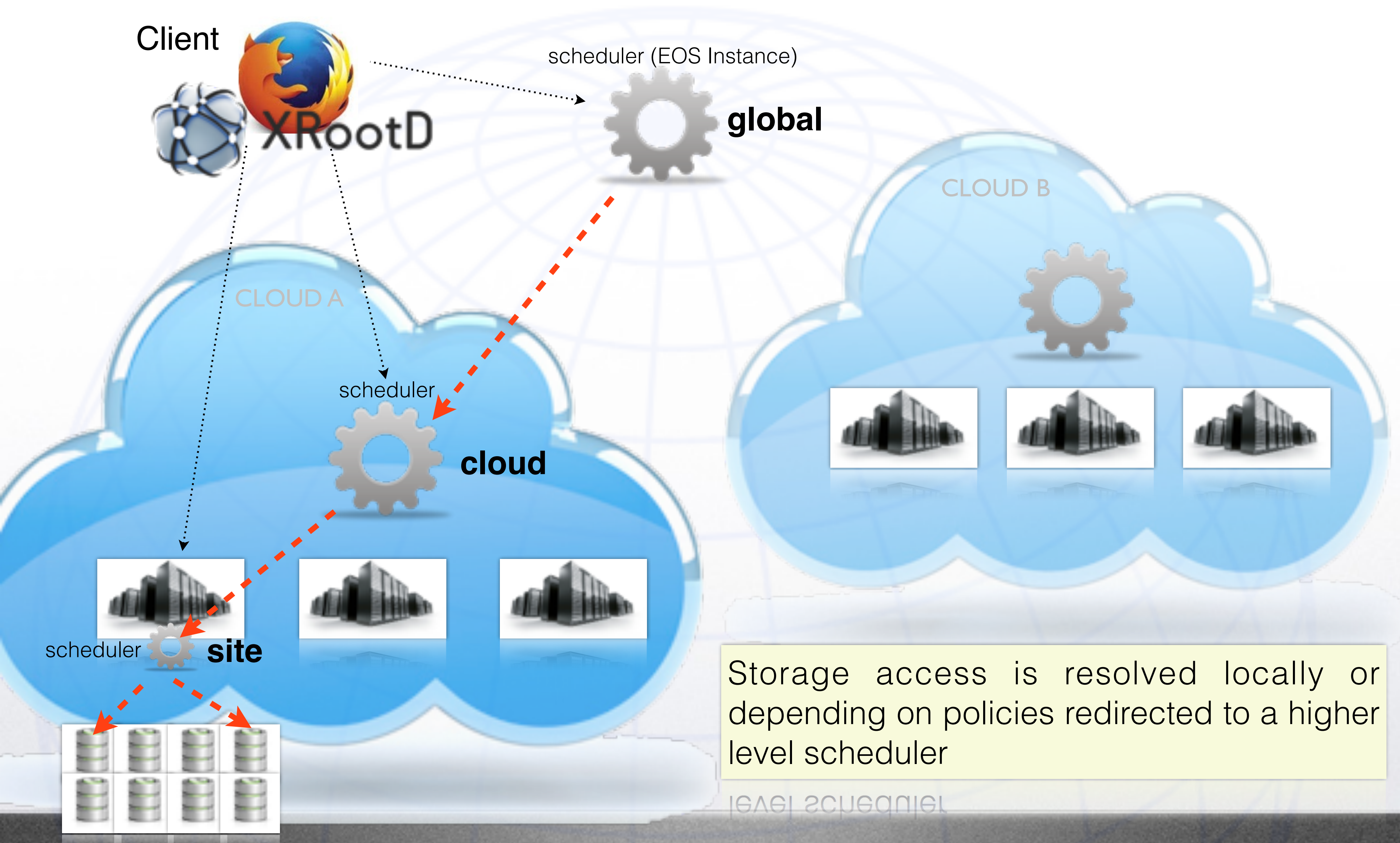

EOS
Storage Bundle
CITRINE
DIAMOND
XRootD
EOS
Storage
Self-contained & Simple
HTTP/XRootD enabled
Storage System
XRootD
VST
ceph
Global
DM
XRootD
ceph
radosFS (C) CERN
Storage
VM NFS
VM Hosting
Global Namespace &
Metadata Catalog
Data Distribution
Policies
Hybrid XRootD
Storage


Diamond R&D
• trivial idea: store a namespace in a scalable object store
• we can represent data in a hierarchical structure using directories and files and we don't need to group an infinite amount of files into a single directory
• each file is a change-log entry without meta data in a directory object
• each directory is represented as an object in an object store as a changelog file
• these change-logs require compacting after many create/delete operations
• a change-log file is perfect to cache remotely: if file size changed fetch the appended piece, if file size shrinks copy the whole file
/
/a
/b
/c
/a/a
/a/b
/b/a
/b/c
directory represented by object
owner
perm
xattr
root
xyz
file1.root


DIAMON VST
"one plugin to rule them all ..."
XRootD
• XrdCl IO Plugin
root
FUSE
/citrine
xrdcopy
XrdCl IF
XrdCl impl
open hook
read/write hook
close hook
VST
•Gobal File Placement
•Global File Access
•UUID generation
•Checksumming
•Access Error Reporting

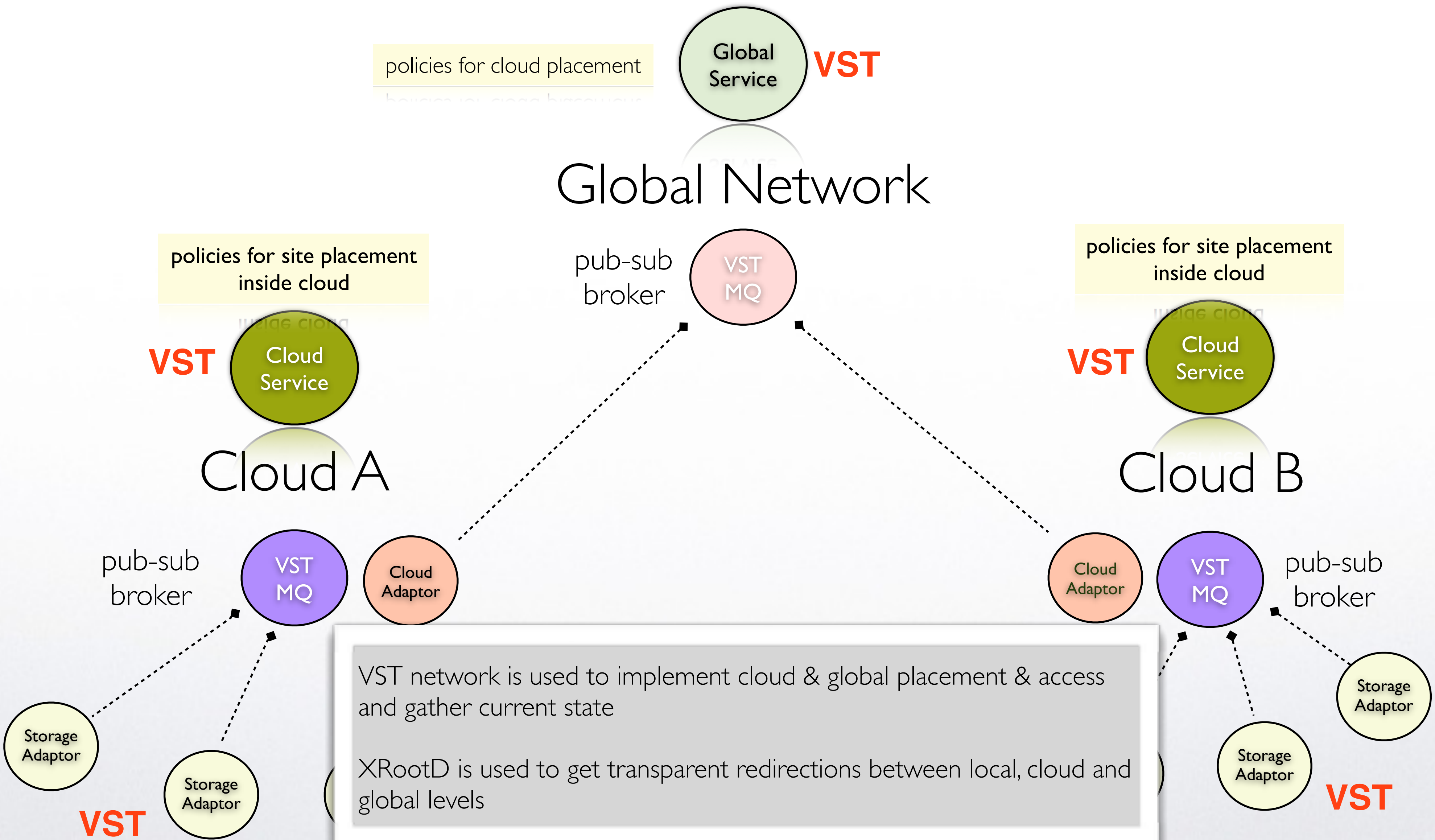
Clustering Sites

Virtual Storage Cloud



Storage access is resolved locally or depending on policies redirected to a higher level scheduler

Clustering Clouds

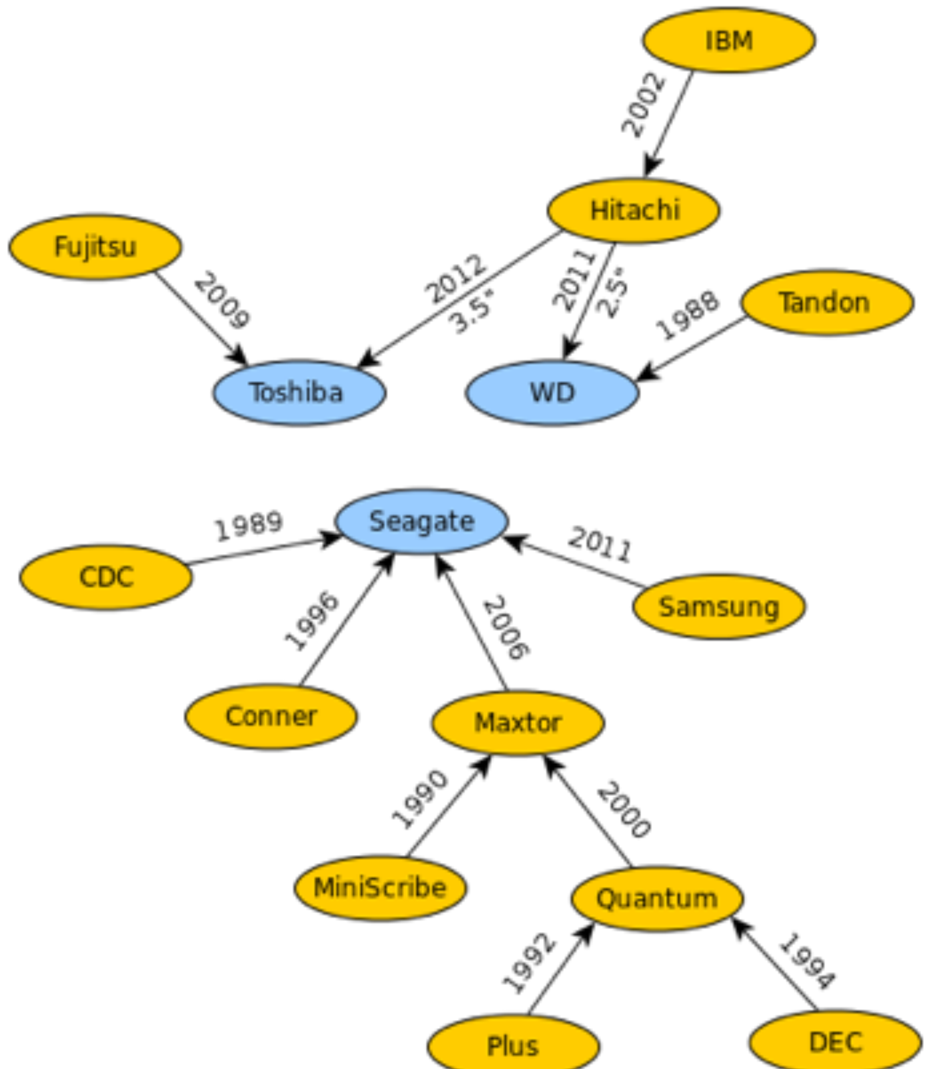


VST network is used to implement cloud & global placement & access and gather current state

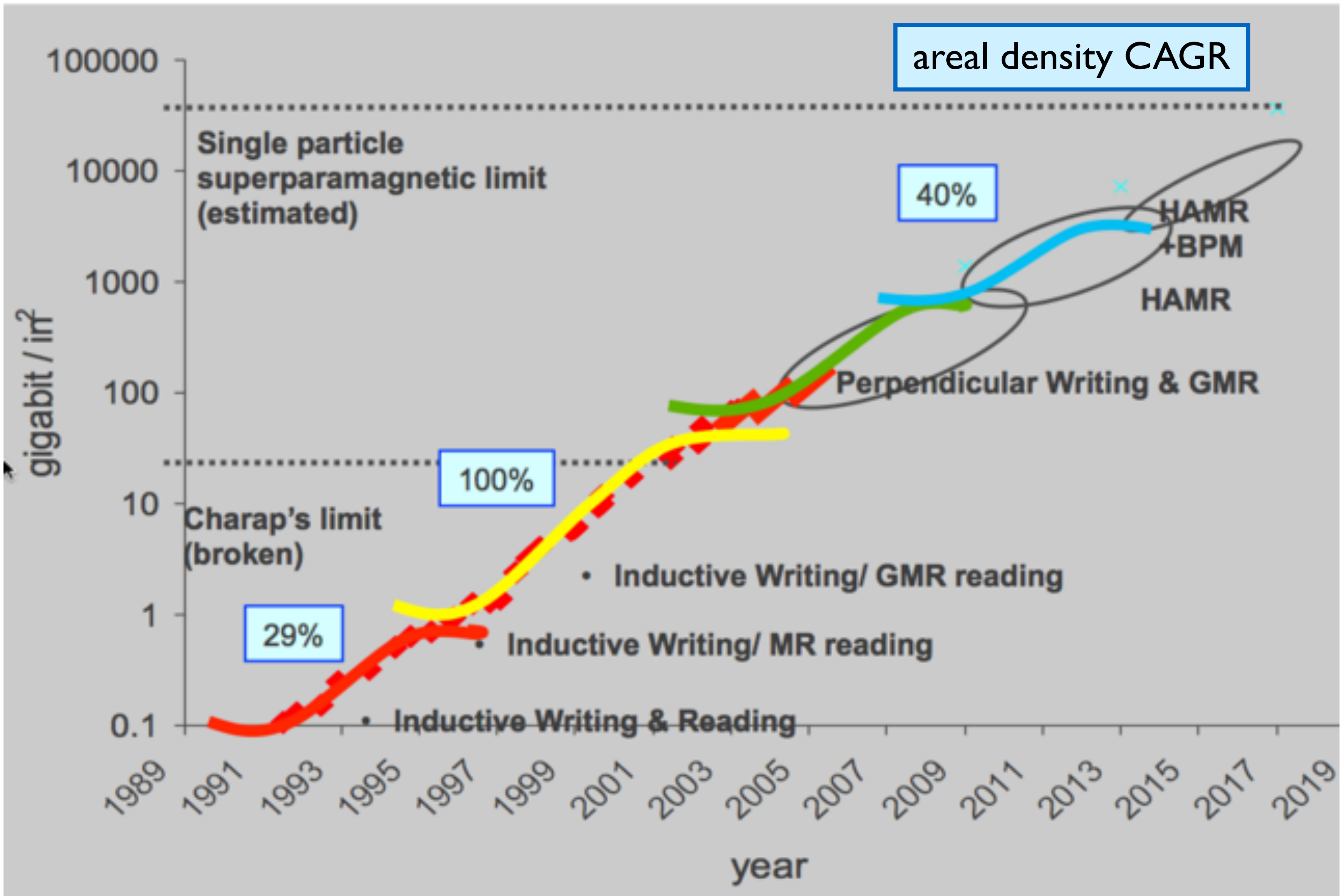
XRootD is used to get transparent redirections between local, cloud and global levels

Back to the basics…

Disk Market Consolidation

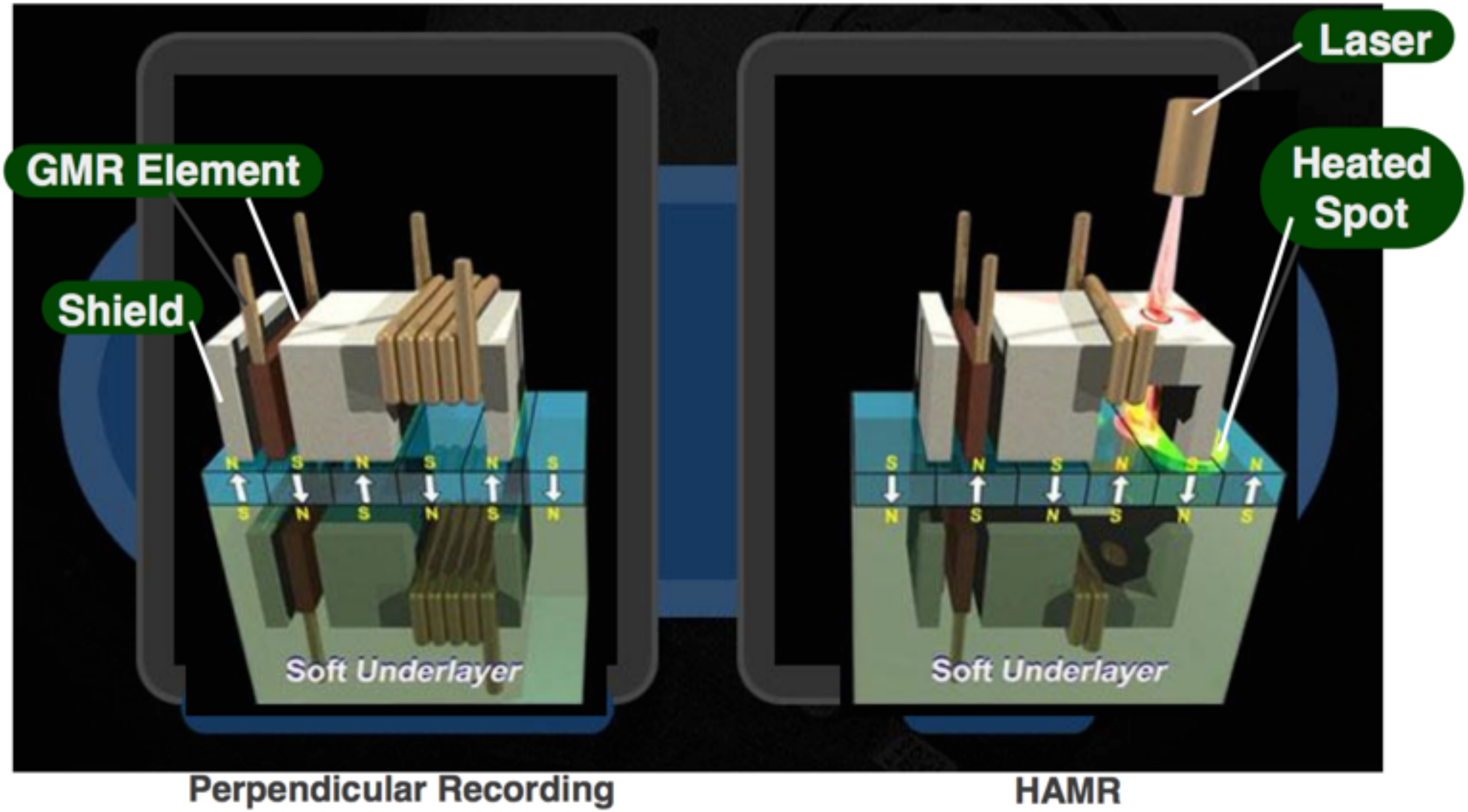


Does Kryder's law still hold? What's next for disk storage?



source: HDD Opportunities & Challenges, Now to 2020, Dave Anderson, Seagate

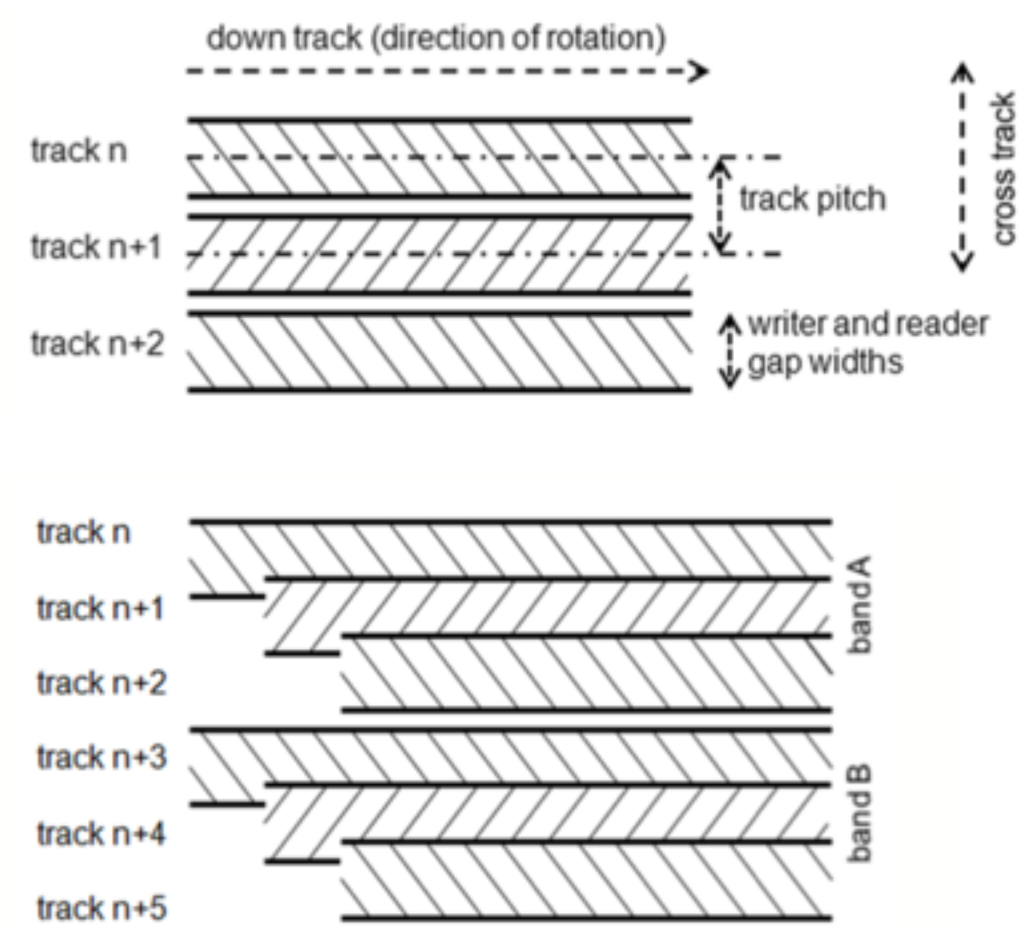
Heat Assisted Magnetic Recording (HAMR)



source: Future Materials Research in Data Storage, Mark H. Kryder

Shingled Recording

- Shingled Media
  - wide write head
  - narrow read head
- Result
  - continued density increase
  - write amplification within a band

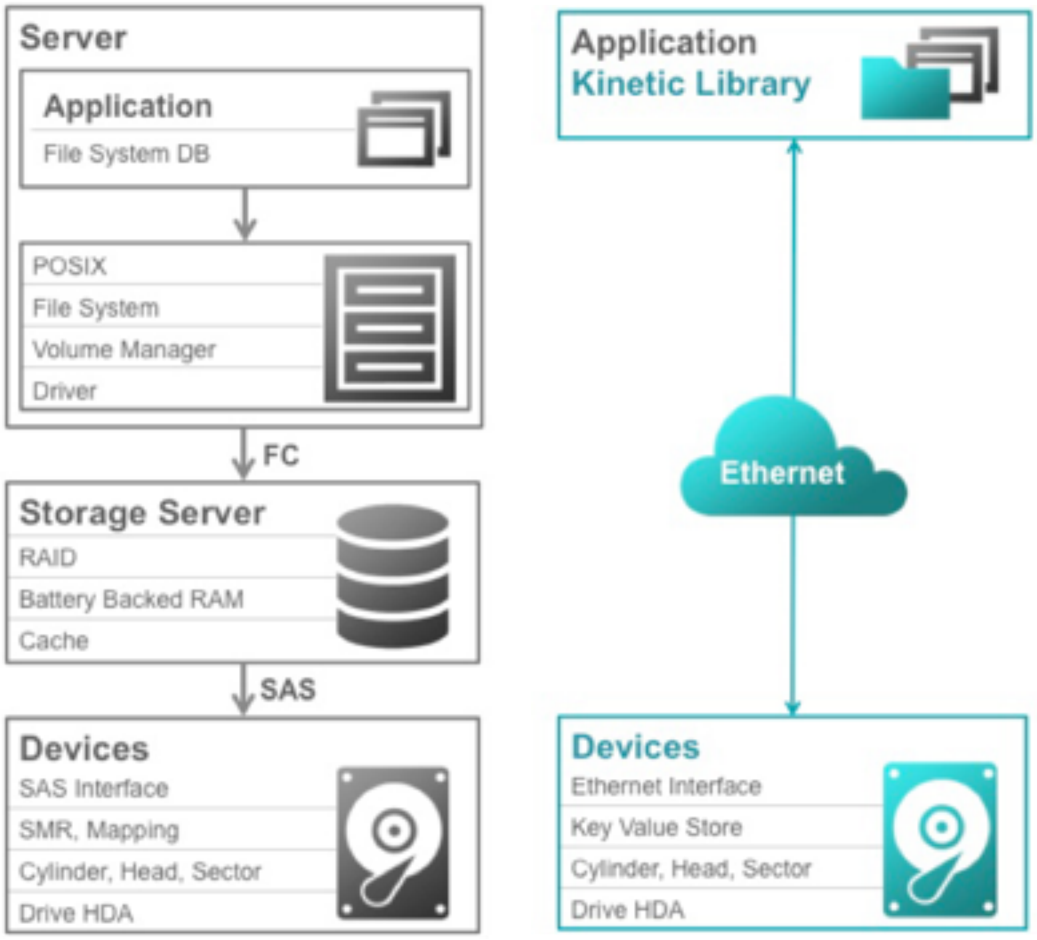


Impact of Shingled Recording

- Gap between Read and Write performance increases
  - need to check eg if meta data mixing with data is still feasible
- Market / Application Impact
  - differentiation into several types of disks?
    - emulation traditional disk
    - explicit management by application
    - constraint semantics (object disk)
- Not clear yet
  - which types will reach a market share & price that makes them attractive for science applications
  - how the constrained semantics can be mapped to science workflows
- => R&D area in CERN openlab

Object Disk

- Each disk talks object storage protocol over TCP
  - replication/failover with other disks in a networked disk cluster
  - open access library for app development
- Other vendors are (re-)evaluating this approach
- Why now?
  - shingled disk technology comes with natural match to semantic constraints: eg no data/ metadata updates
- Early stage with several open questions
  - port price for disk network / price gain via reduced server CPU?
  - standardisation of protocol/semantics to allow app development at low risk of vendor binding?

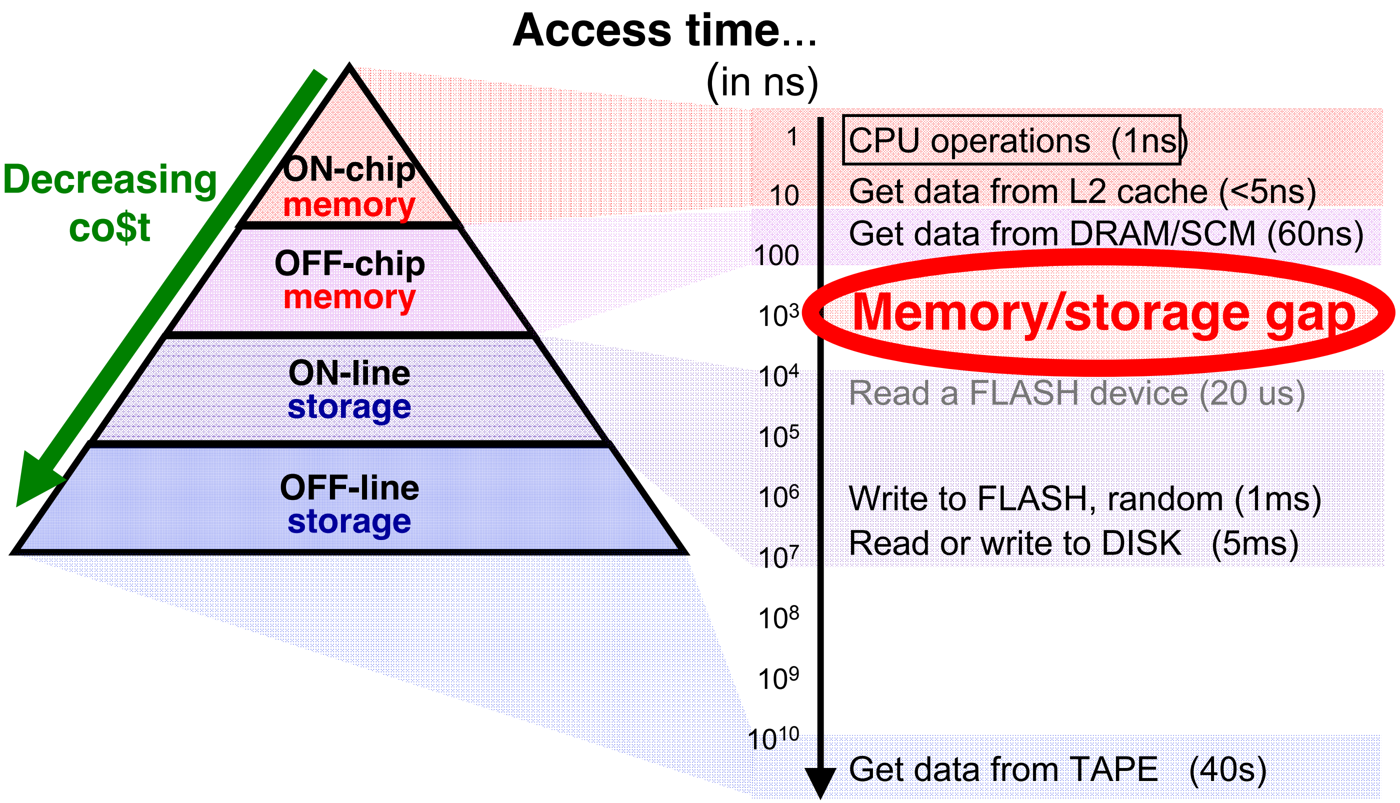


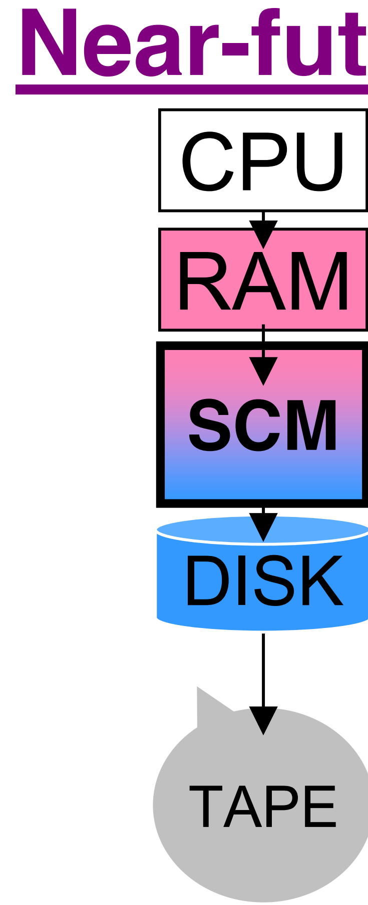
CEPH - Object Storage

- CEPH
  - redundant object store with client side calculated placement decision (CRUSH)
  - RADOS - native access
    - S3 / Swift via gateway -> scalability impact?
  - additional consolidation possibilities for sites
    - block storage (eg for VMs) used in AI project
    - CEPH file system
      - not yet supported - but “almost awesome”
- Interest from several projects to evaluate
  - CASTOR: match high-speed tape drives to “slow” disk cache for migration/recall

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Problem (& opportunity): The access-time gap between memory & storage





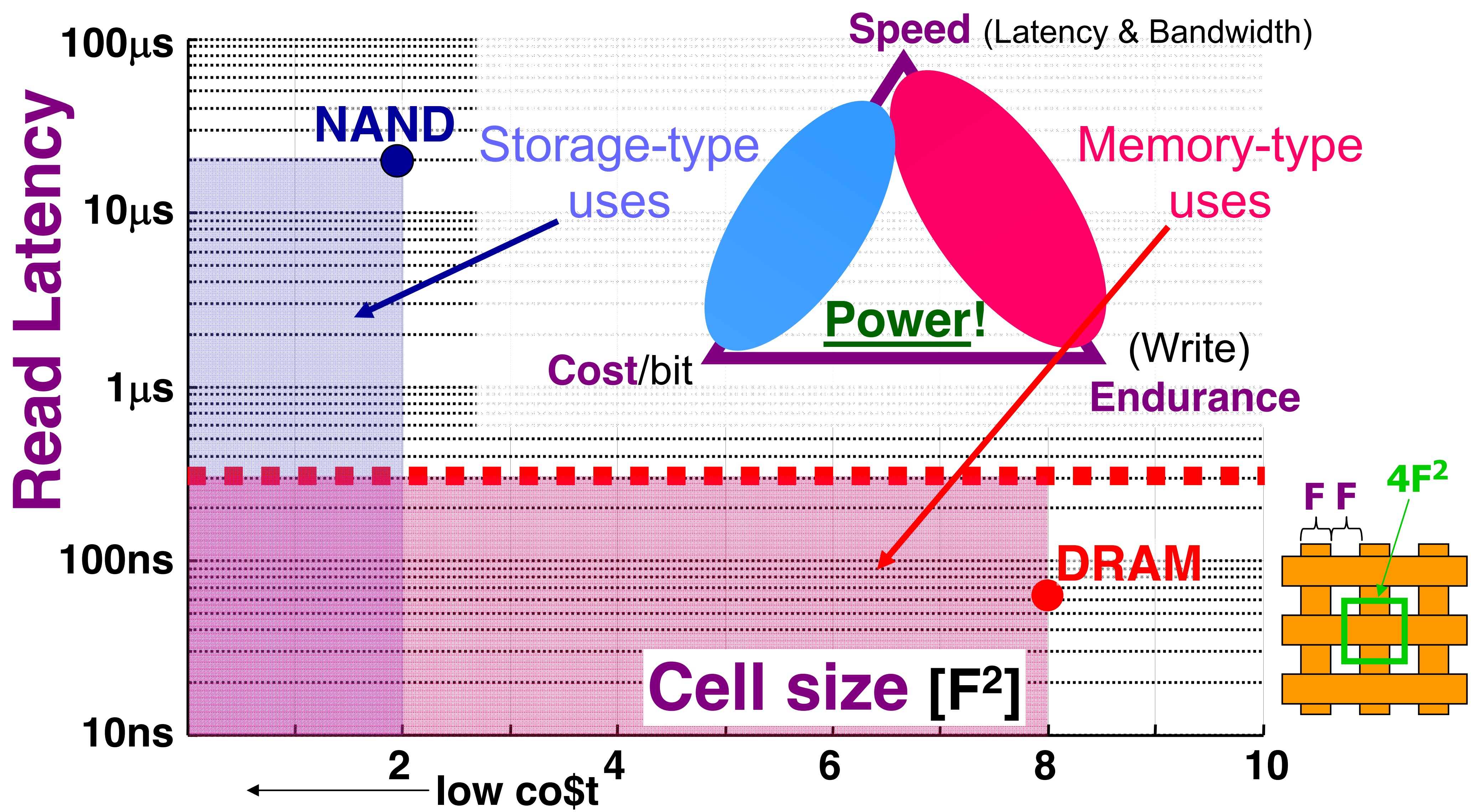
Research into new solid-state non-volatile memory candidates
– originally motivated by finding a "successor" for NAND Flash –
has opened up several interesting ways to change the memory/storage hierarchy…

1) **Embedded** Non-Volatile Memory – low-density, fast ON-chip NVM
2) **Embedded** Storage – low density, slower ON-chip storage

3) M-type **Storage Class Memory** – **high-density**, fast OFF- (or ON*)-chip NVM
4) S-type **Storage Class Memory** – **high-density**, very-near-ON-line storage

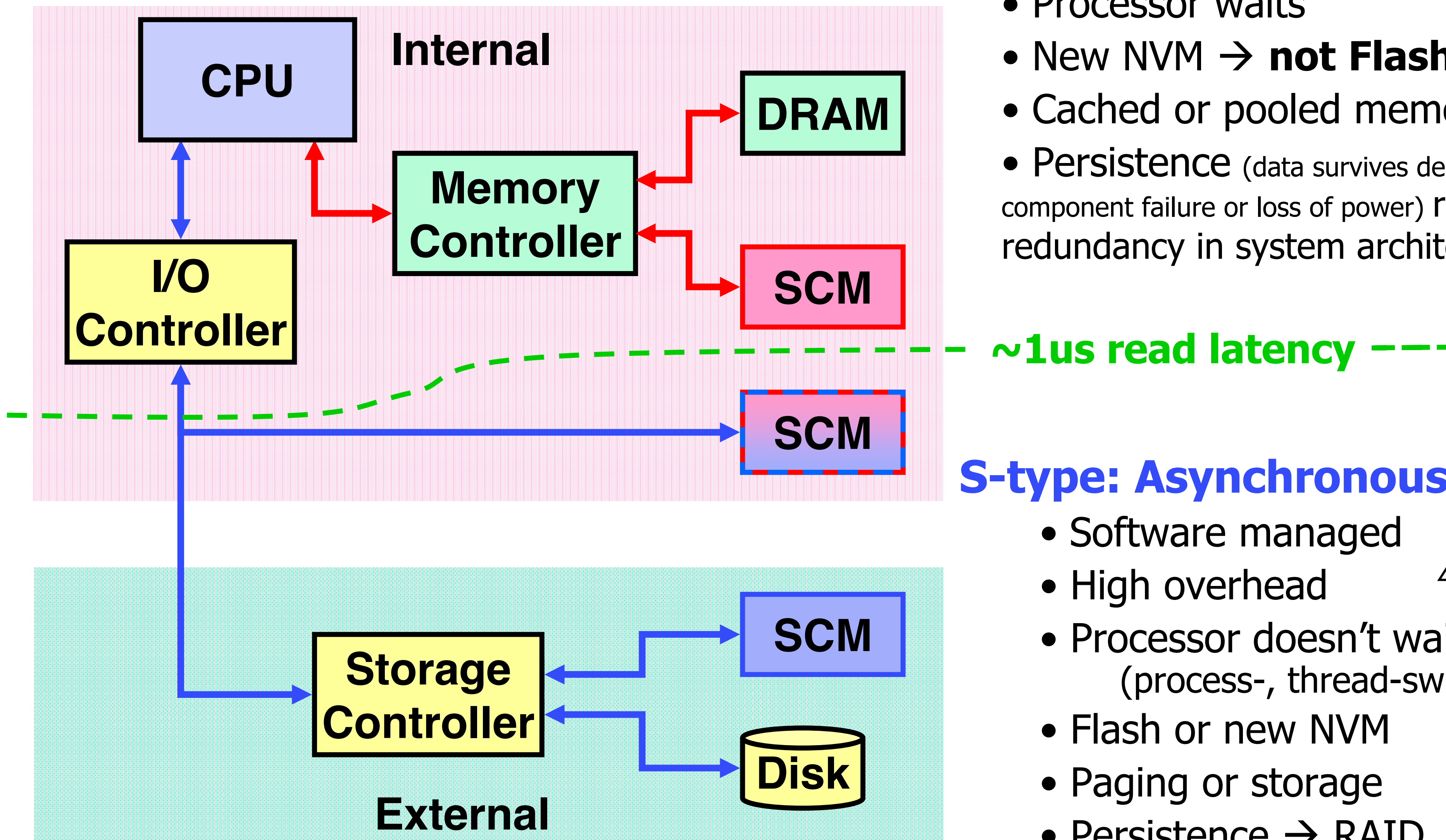
* ON-chip using 3-D packaging

Storage-type vs. memory-type Storage Class Memory



The cost basis of semiconductor processing is well understood – the paths to higher density are
1) shrinking the minimum lithographic pitch **F**, and 2) storing **more bits PER 4F²**

S-type vs. M-type SCM



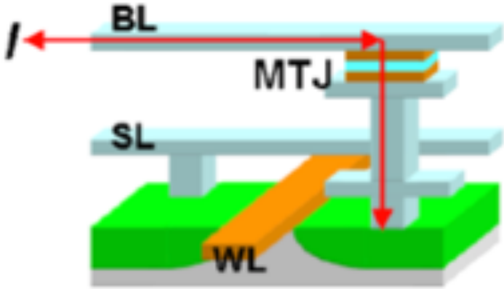
M-type: Synchronous

- Hardware managed
- Low overhead
- Processor waits
- New NVM → **not Flash**
- Cached or pooled memory
- Persistence (data survives despite component failure or loss of power) requires redundancy in system architecture

S-type: Asynchronous

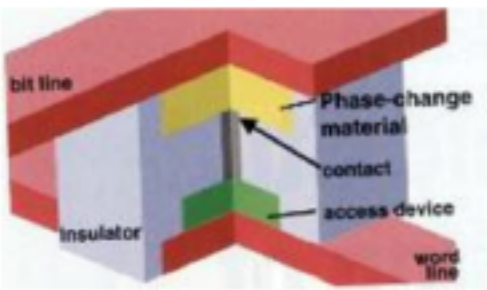
- Software managed
- High overhead
- Processor doesn't wait, (process-, thread-switching)
- Flash or new NVM
- Paging or storage
- Persistence → RAID

STT-MRAM



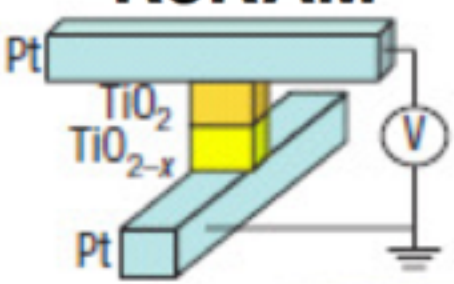
- High speed operation and non-volatility
- Main contender for DRAM replacement
- Eliminating DRAM refresh is a latency, bandwidth & power opportunity for STT-MRAM
- Complicated MTJ stacking structure, Yield challenge
- High temperature process & Low resistance ratio
- Margin Challenges, Soft errors
- 1x nm scaling and cost competitiveness??

PCM



- Most mature amongst emerging memory candidates – low density PCM in production for NOR replacement
- Drift challenges with high density PCM, Stuck Faults – reliability challenge
- Active Power, write current & latency – power/thermal challenges, too slow to work as main memory
- Scaling vs Thermal disturbance ??

ReRAM



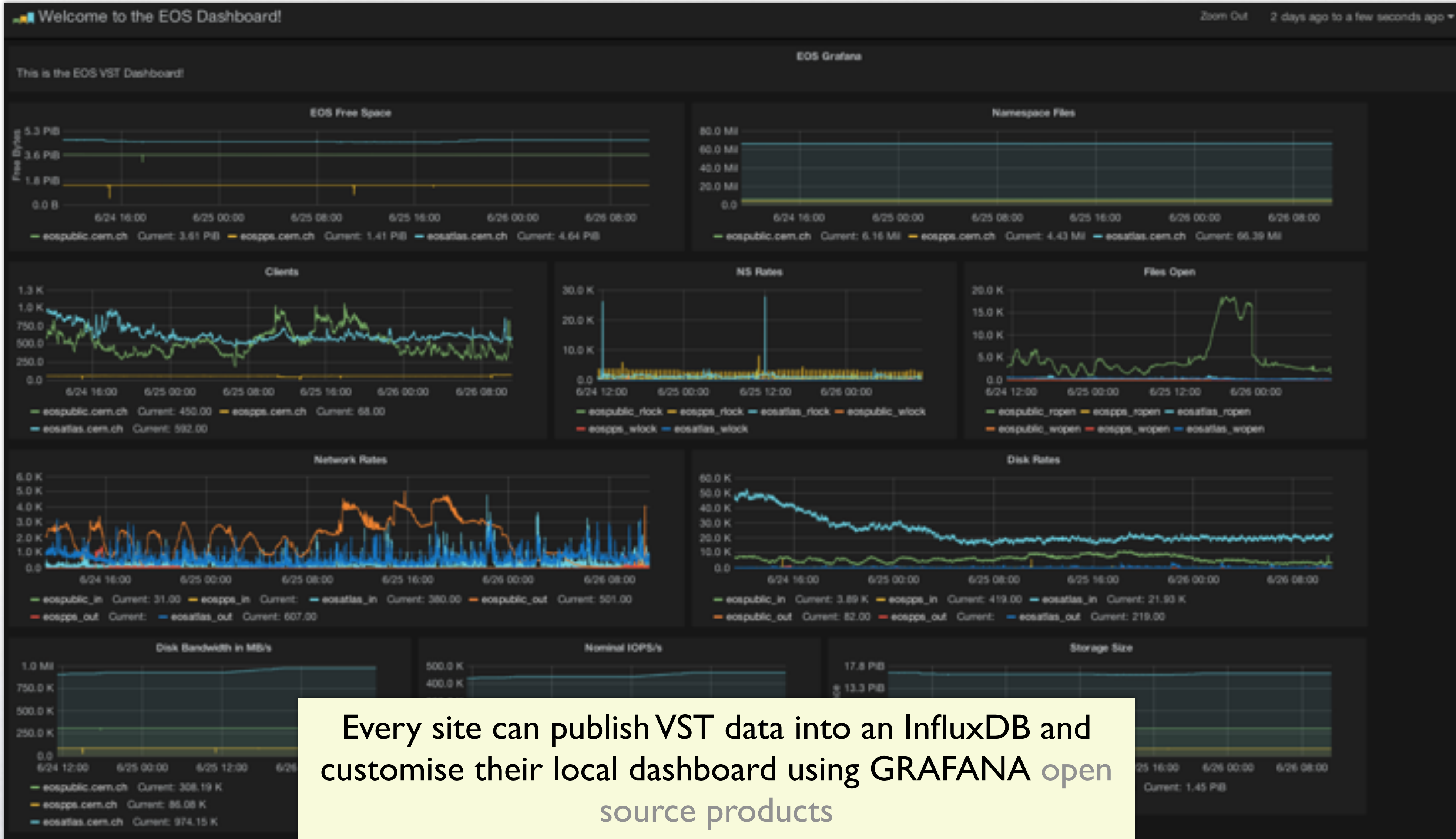
- Very simple materials and structure
- Fast access, moderate endurance and low power
- Various and unclear switching mechanisms
- Large cell-to-cell variability
- EUV needed vs 3D NAND
- Stacking required for high density – manufacturing & yield challenges??

Last but not least…

Storage “monitoring”

- Many different use cases with different requirements
- Operational monitoring
  - Is the system behaving as expected?
    - component status, error frequencies
    - Any reason to alert operational staff
  - Are the users behaving as expected?
    - is the resource consumption inline with expectations and experiment priorities?
- Longer term analysis
  - is the replication factor of a files suitable?
  - is the size of a storage system adequate for associated CPU resources?
  - is the relative investments in tape, disk, network and CPU cost optimal?
- Both areas use same metrics collection but use very different methods to process them

GRAFANA VST Dashboard



Every site can publish VST data into an InfluxDB and customise their local dashboard using GRAFANA open source products

Some existing or planned cache components

CERN IT Department

Where	What	Why	Who	How	Size	Lifetime	Accessed
Disk Server	FS cache	reduce repeated disk IO	OS/VM	pull	GB RAM	hours	kHz
Site (managed)	File Placement (SE + Catalog)	push popular data to avoid transfer I/O wait	content: exp storage: site	push	10-100 TB (disk)	months	10-100Hz
Site (unmanaged)	Proxy/CDN (eg SQUID, Xroot proxy, {Event Proxy})	reduce latency for repeat reads increase bandwitdh via tree hierarchy	storage: site optionally: exp push	pull	10TB??	weeks/months	10-100Hz
	may come with file/block/{event} granule - efficiency depends on popular fraction of cache granule						
Worker Node	Async read-ahead	increase CPU/IO overlap	job	async pull	GB (RAM)	job lifetime	<Hz
	persistent version of above	reduce repeat reads between jobs (eg user laptop case)	user	pull	10 GB (disk)	weeks?	<Hz
	FS cache for file:// access or WN download	reduce repeated disk/ net IO	OS/VM	pull	GB RAM	hours	100 Hz
Process	TTreeCache	reduce network/disk round-trips	root + exp framework	pull	10-100 GB (RAM)	job lifetime	<Hz
	usage currently different between experiments and partially implemented in exp frameworks						

Ideally we would look at this with an overall throughput-increase/$ perspective
- but we still miss a lot of analytics to get there

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

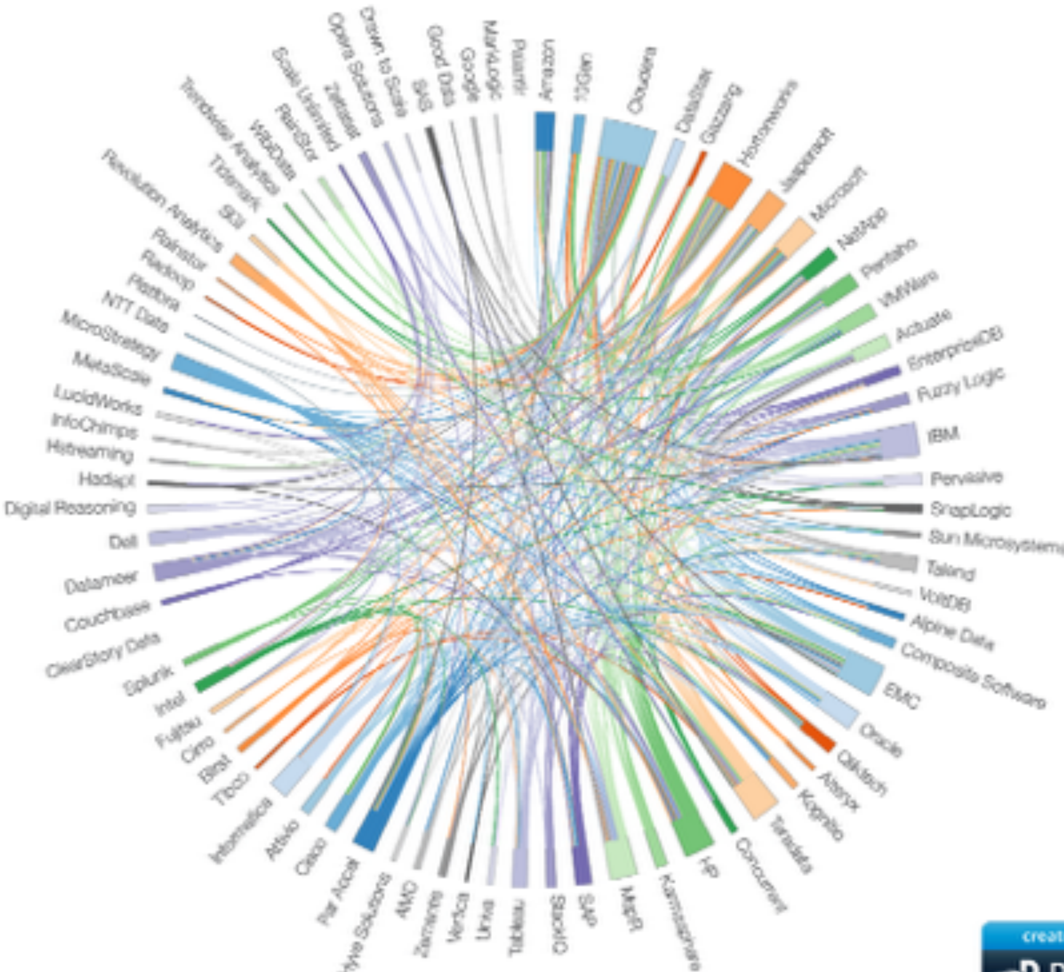
Analytics for Storage

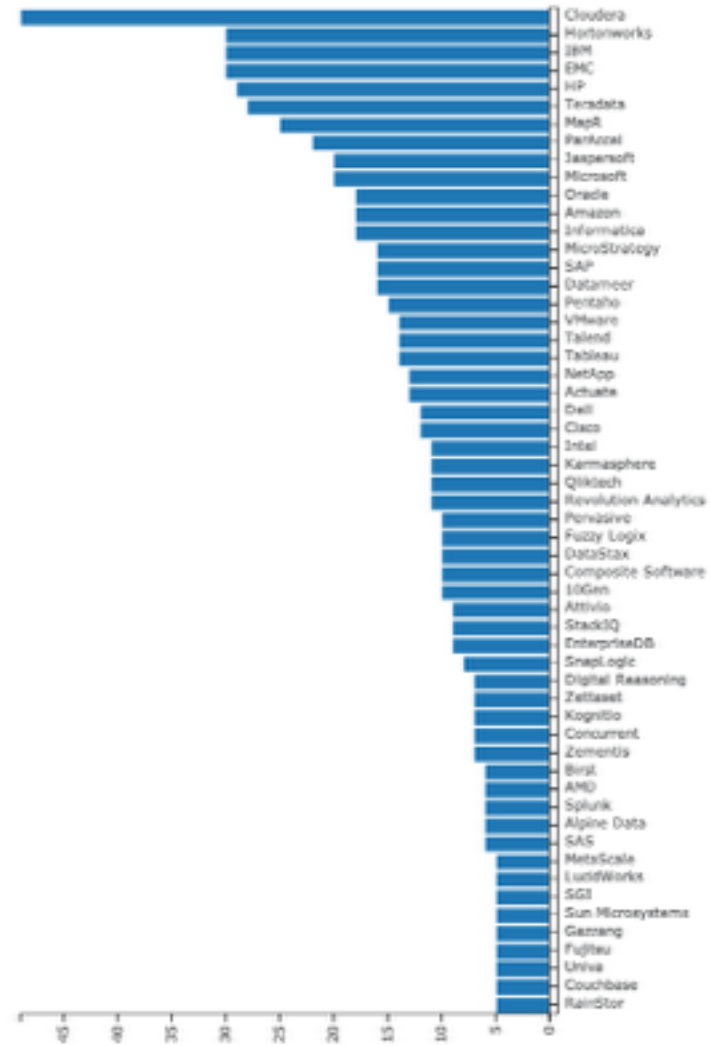
- Now that we have a large scale system with a significant amount of usage and performance data collected, we can try to apply the same statistical methods and modelling to this data as we use for "Physics"
- The input data more complicated as many additional input metrics are needed
  - cpu & memory utilisation, location, hardware type and virtualisation
  - many of them are only available in log files and dispersed databases
- We finally have a standard "Big Data" problem similar to many of the commercial Big Data names
  - luckily on a much smaller scale than our physics data

HADOOP ECOSYSTEM

Who has the most connections/partners?

data from January 2013



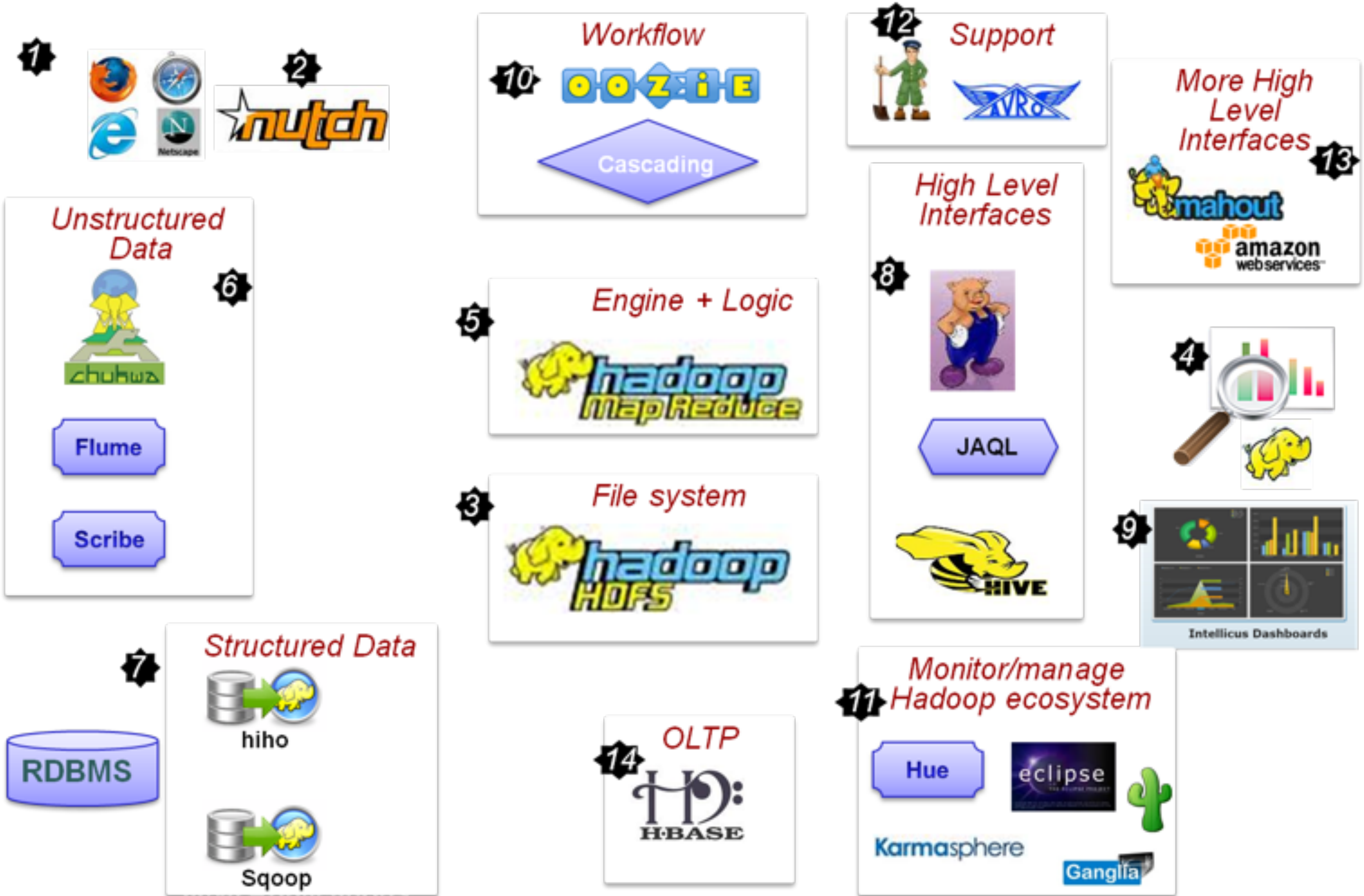


created using Datameer

Hadoop

- Hadoop is not just storage
  - but a complete processing infrastructure
    - with generalised resource management
- Parallel local access
  - Map Reduce
  - PIG/Latin
- Consistency constrained (scalable) database features
  - HBase
  - Spark
- Significant interest for analytics from IT and experiments
- We will see how far we get on the other side of the gardner hype curve.

Hadoop Ecosystem Map (2010 - outdated)



source: http://indoos.wordpress.com/2010/08/16/hadoop-ecosystem-world-map/

Summary

- Storage systems played and will play a crucial role in HEP and in increasing number of other sciences
- The evolution of the base technologies has allowed us follow the ever increasing demands from the science community
  - physical limitations seem to allow continuation for the foreseeable future
  - new storage technologies are likely to appear during the LHC program and may change the way we develop science software and workflows
- After a design and early deployment phase storage and computing may enter a quantitative optimisation phase to review existing and upcoming system architectures