Quantum®

Your difference is in your data.™

Building a 100-year archive

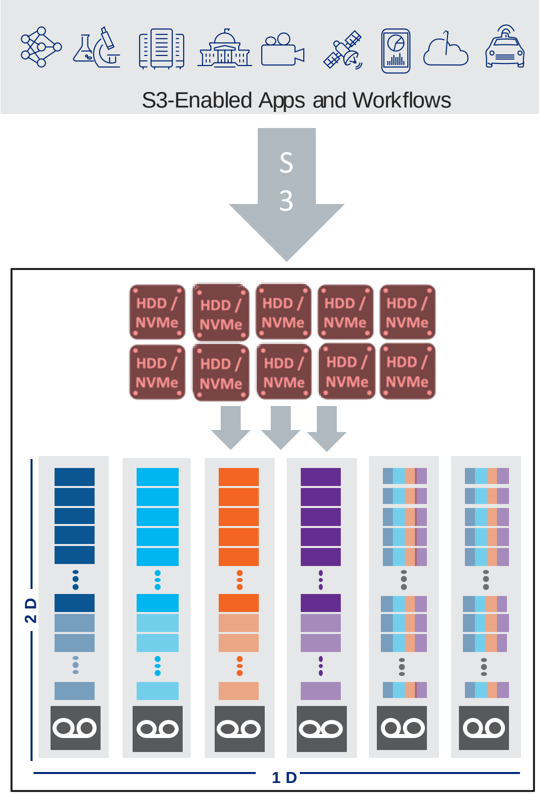
Stijn Blyweert & Glenn Heinle

stijn.blyweert@quantum.com
glenn.heinle@keepertech.com

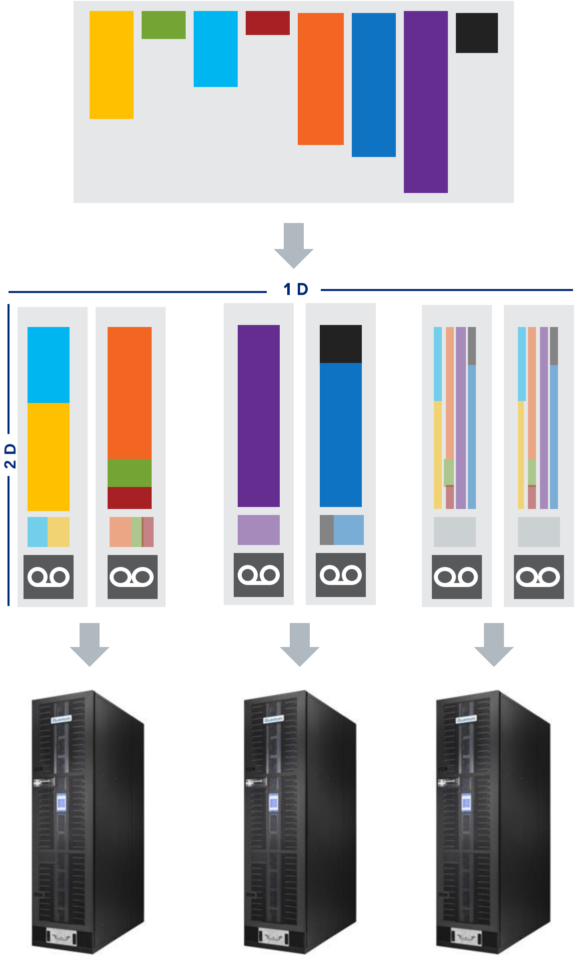
What requirements should shape a 100-year archive today?

- Use proven technology
- Technology Refresh
- High durability
- End-to-end integrity verification for reads and writes
- Ongoing verification
- Logging, Reporting, Auditing
- No Vendor Lock-In

ActiveScale Cold Storage



- Patented 2D Erasure Coding
  - Across-Tape Erasure Coding
    - Protect against multiple tape failures
  - Within-Tape Erasure Coding
    - Reduce read errors by factor 100 (4% overhead)
- Optimal performance
  - Read object from a single tape
  - Only large writes to tape
  - Restore queue
- Codesigned by Quantum's object storage and tape experts



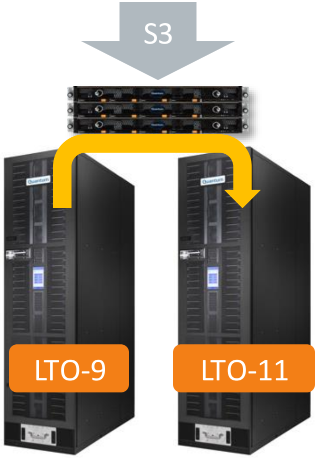
Use Proven Storage Technology

- Majority of archival storage is on **LTO Tape**
  - No good low-cost alternative available today
- But think about **migration** to newer technologies
  - New LTO generations, new storage technologies, …
  - Migration of old tapes can take a long time
    - Older tapes = more wear and tear

- Storage systems with built-in migration capability
  - Can do the **most efficient migrations**
  - Migration can happen transparently to end-users
    - Keep on using same credentials and permissions, same file/object names, …

ACTIVESCALE COLD STORAGE

- Migration is much easier due to Erasure Coding
  - Within-tape EC: reduce read errors by **factor 100**
- S3 frontend ensures transparent migration
  - Same S3 credentials, S3 Buckets, Object Names, …



High Durability at a Low Cost

30PB usable capacity	ASCS 2D Erasure Coding	Traditional Multi-copy Approach		
		1-Copy	2-Copy	3-Copy
Storage overhead	17.6%	0%	100%	200%
LTO-9 Tapes	1,996	1,667	3,333	5,000
Durability	15 Nines	3.2 Nines	6.4 Nines	9.7 Nines
Objects Lost Per 1B Objects	1 / 1M yr	603k / yr	363 / yr	1 / 4.6 yr

- Erasure Coding results in higher durability **and** lower cost
- Other durability-enhancing features
  - End-to-end data integrity checks for read and write
  - Ensure high-quality writes
  - Regular data integrity verification
  - Reduce tape wear-and-tear due to reading and writing

Scalar i7 RAPTOR
2000 slots

Logging, Reporting & Auditing

- Ensure admins know what happens with the data
- Need **visibility** into storage system operations
  - Creates, Deletes, New Versions (who, where, when)
  - Asset verifications (integrity checks)
  - Asset moves (due to technology updates)
- Traditional approach: **log files**
- But this has **scalability concerns**
  - On a large system there are billions of objects, resulting in very large log files
  - Investigate better approaches?

ACTIVESCALE COLD STORAGE

- Information is available in log files
  - Admins can get **read-only** access to these
- Alerts if data hasn't been verified on time

- Thinking about a more scalable approach
  - Make this information available via S3 API?
    - S3 Custom Metadata? Or custom S3 API extension?
  - Or via Event Notifications? (Kafka, …)
  - Community-wide agreement would be ideal

No Vendor Lock-in

- Making sure you can still access your data when something happens with your storage vendor

- Traditional approach: **LTFS**
- But LTFS has **technical limitations**
  - Causes more **wear-and-tear** of tapes
    - Due to separate metadata partition
    - Becomes a bigger and bigger issue with higher density media
  - It's **POSIX**, so it must be easy to use, right?
    - Looks easy, but that's deceiving
    - POSIX was **not designed** for high-latency storage

ACTIVESCALE COLD STORAGE

- We intentionally avoided LTFS
  - Due to its technical limitations
  - We designed a **more optimal** on-tape format
- How do we ensure there is no vendor lock-in?
  - We **share** our on-tape format with customers
    - There are no secrets in our on-tape format
  - We also **share** how to read data from tape without the S3 frontend
- S3 Glacier API = The Right Way to do S3 with Tape
  - S3 Glacier API is **designed for high-latency storage**
  - Most optimal way to do tape I/O
    - API allows **buffering & queueing** to optimize tape I/O

Quantum®

Your difference is in your data.