Topic 5 - Archival Storage

Speaker notes

No notes on this slide.

Speaker notes

No notes on this slide.

Reality check

- “*Just storing files*” is not preservation
- Storage market focus may deviate from your use case
- There’s more than just *one right* solution…
- Mixing is usually a good idea.

Speaker notes

No notes on this slide.

Speaker notes

No notes on this slide.

Physical Media Types

Optical Disks

Speaker notes

Pros:

- Cost/size ratio: Relatively cheap (cheapest still?)
- Not susceptible to electro magnetic (dis)charge
- No mechanics/electronics in carrier - only in the drive

Cons:

- Error prone
- Suffers easily from material degradation (scratches, data layer falling off, light exposure, temperature, etc…)
- Becoming less and less common = less support, less choice, etc.

HDD: Hard Disk Drive

Speaker notes

Pros:

- Reasonable cost/size ratio
- (yet) cheaper than SSD
- well known, well supported

Cons:

- Mechanical wear (moving parts)
- Susceptible to electro magnetic discharge
- Electronics on carrier

Data Tape

Speaker notes

Pros:

- Cost/size ratio: Tape cartridge “cheaper” than HDD
- Tape preserves well (and there is much experience with dealing with tape degradation/aging from analogue)
- Supported in “tape libraries” (aka “tape robots”) for removing “tape-jockey” limitations.
- Larget capacity per cartridge than per HDD (yet)

Cons:

- Not common outside certain domains (large institutions, broadcast preservation, …)
- Therefore support for tapes very “specialized”
- Drives quite expensive (>3000 EUR)

LTO Generations:

LTO improvements include new LTO generations: Higher data density at same form factor. This allows the mechanics (of drives and robots) to stay compatible, with only requiring certain parts to be updated to support a new generation.

The LTO consortium guarantees a certain level of downwards compatibility. This used to be:

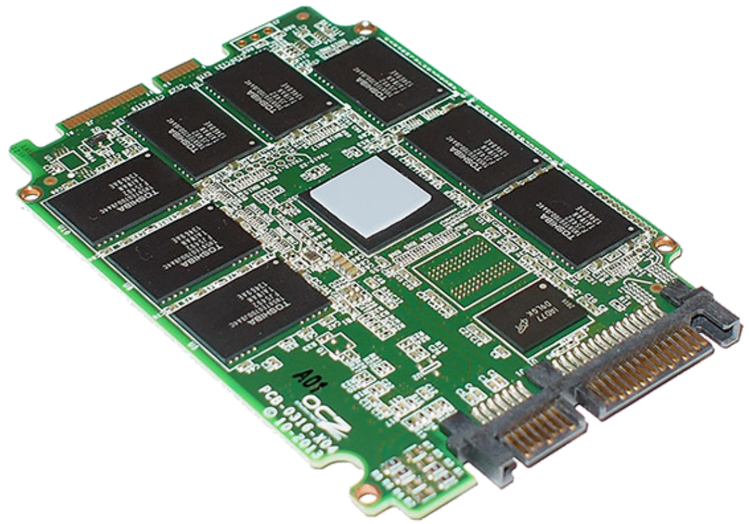
- Read: 2 generations down
- Write: 1 generation down

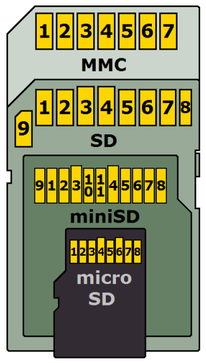
Example: LTO-6 drive should be able to read LTO 6,5,4 - but write only 6 and 5.

With LTO-8, this was reduced to 1 generation backwards compatibility, so an LTO-8 drive can not read LTO-6, only back to LTO-7

New LTO generation release: About every 2-3 years. This must be considered for migration!

Flash Memory





Speaker notes

Pros:

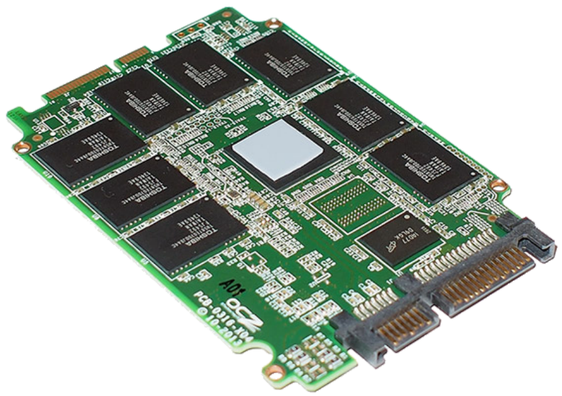
- cost/size ratio: getting cheaper, but still more expensive than HDD
- No moving parts = no mechanical wear
- SD cards: almost no electronics in the carrier
- Speed: Can be extremely fast

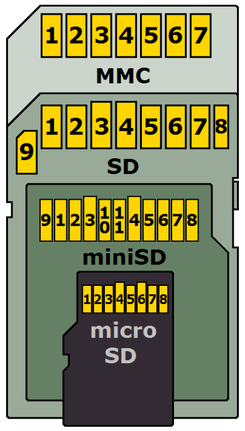
Cons:

- Currently most expensive (still)
- Lack of long-term experience
- Limited number of write/erase cycles per flash block (becoming less of an issue) =Should not be used for large number of write operations. But preservation storage case is mostly “read”.

More about Flash memory on Wikipedia. SD card image by Steve Meirowsky (Wikipedia) - Own work, CC BY-SA 3.0

Media Types: Overview





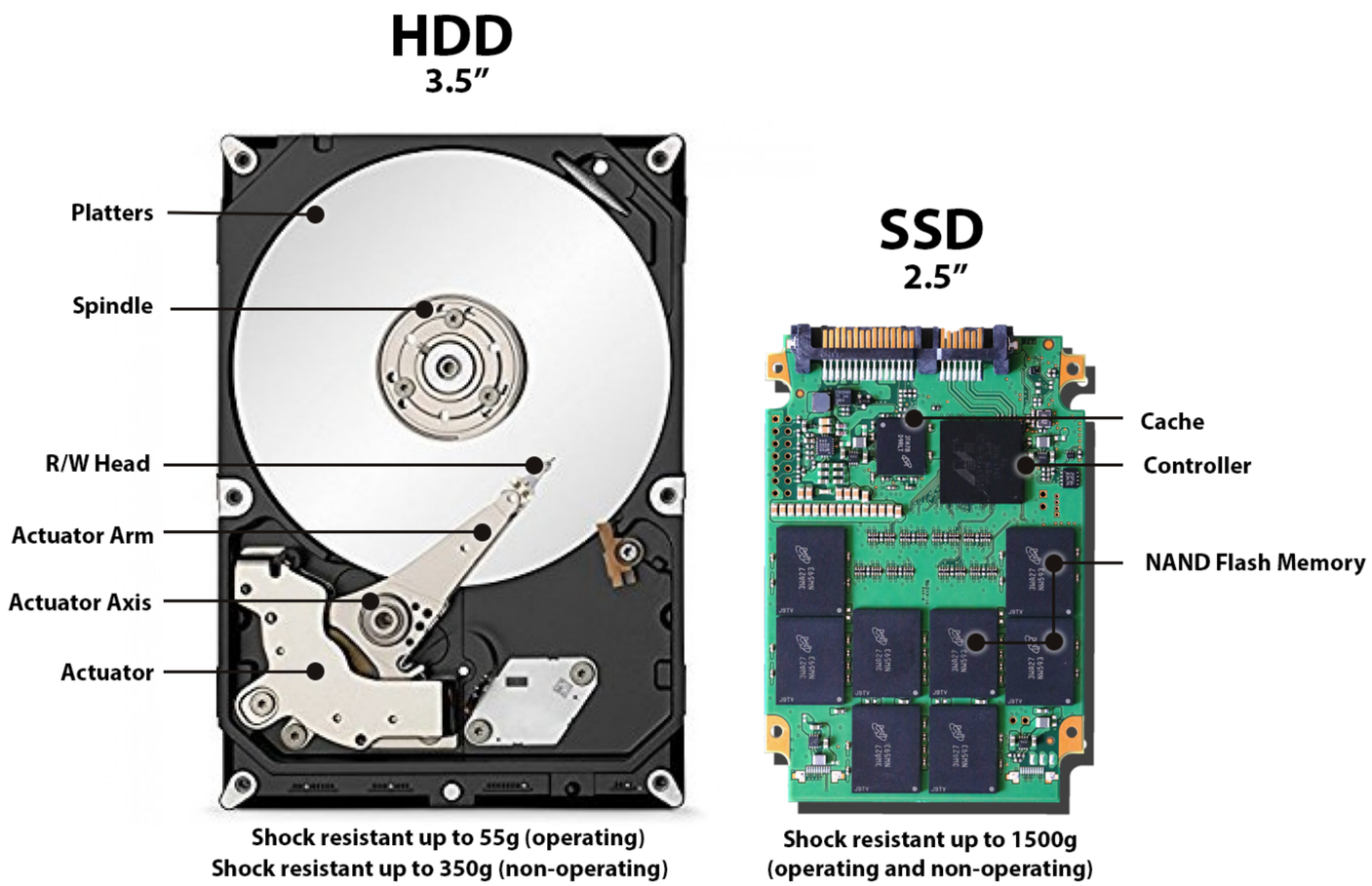
Speaker notes

- Data Access Speed?
  SSD > HDD > Tape > Optical
- Cost/size ratio?
  SSD > HDD > Tape > Optical
- Widespread?
  - SSD, HDD: Excellent!
  - Tape: almost exclusive professional use case
    Drives not that “easy” to get (for particular LTO generation)
  - Optical: Declining (except for “cold storage” in large data centers)

It’s usually good to mix storage media. For example for different use cases:

- preservation / backup:
  tape (affordable size, good preservation properties, less need for super-fast access)
- access: No need for long-term endurance, so e.g. optical would be sufficient.
- production: HDD-based storage for nearline files - tape for offline storage. Or: tape-robot with HDD cache (same concept, but fully automated)

HDD vs SSD



Speaker notes

HDD:

- (yet still) cheaper cost-per-data ratio
- moving parts = mechanical wear out
- Faster than tape, slower than SSD

SSD:

- Super fast!
- (yet still) more expensive than HDDs (equal size)
- NO moving parts
- Limited number of writes

Good to know:

- Higher data density = more impact of an error. Example: 1mm hole in CD vs DVD vs BluRay - or SD vs MicroSD, etc.
- HDD: The longer it is active, the shorter is lives.
- But: “*Hardware that lies, dies.*”
- Be aware if your HDD is “Shingled (SMR)” or not.

Speaker notes

Density:

- Think of a 1mm hole on an optical carrier: CD vs DVD vs BluRay
- All 3 have the same dimension, but different data density.
- When choosing a carrier: Do you really need it to be as small/dense as possible?

Hardware that lies, dies:

Got external HDDs stored on shelves?
You may want to power them up every once in a while (every 1-2 years or so), and copy the data off them before the hardware gets old. Because: Electronics fail (capacitors), lubricants harden out or movable parts get “lazy” or stuck.

Das Werkstatt Rule #2 (German): “Hardware die liegt, stirbt” Useful “Das Werkstatt” principles

Shingled HDD:

> *“The more time a shingled disk spends in band compaction, the more it is wasting resources.”*

Source: Classifying Data to Reduce Long-Term Data Movement in Shingled Write Disks This means, SMR disks are active when idle, causing them to potentially wear out faster.

Storage Types

Speaker notes

No notes on this slide.

External Hard Disks

Speaker notes

No notes on this slide.

External Hard Disks

Ideal for:

* Collections ≤ [size of largest HDD] * Accessed by only 1 computer at a time * Moving data * Quick access

Advantages:

* Relatively low cost (100-300 EUR) * Portable

Disadvantages

* Risky: Drives may fail * Backup is manual and easily out of sync * Access is limited

Speaker notes

No notes on this slide.

Network Attached Storage (NAS)

Speaker notes

No notes on this slide.

Network Attached Storage (NAS)

Ideal for:

* Collection < ca. 40 TB (if standalone). * Larger is possible, if clustered. * Quick access (incl. multiple networked users) * Networks with ≤ 1 GbE for large files (10 GbE for film) * Organizations with some IT support

Advantages:

* Multiple users can access in parallel * Relatively affordable (200-1000+ EUR)

Disadvantages:

* Requires heating / cooling * Potentially less secure (always on) * Less portable * Requires IT skills for problem solving

Speaker notes

No notes on this slide.

Data Tape

Speaker notes

No notes on this slide.

Data Tape

Ideal for:

* Collections from 10 TB to x PB * Collections that don’t need to be accessed in seconds * Back up scenarios

Advantages:

* Relatively low cost for tape stock * Scalable * Portable * Low failure rates

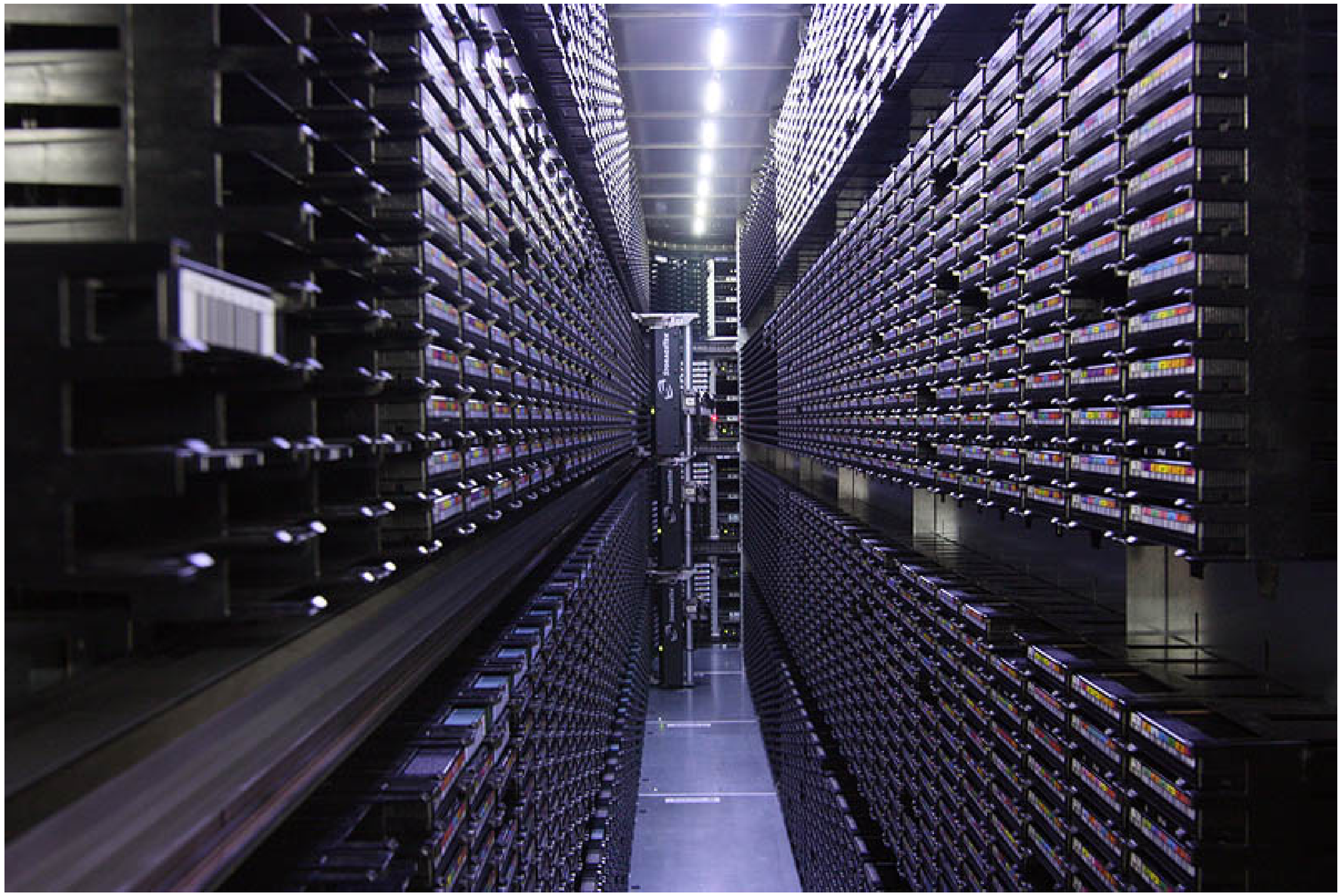
Disadvantages:

* Offline or nearline * Management & migration can be challenging * Proprietary tape filesystems

Speaker notes

No notes on this slide.

Data Tape Library (Robot)



Speaker notes

No notes on this slide.

LTO Generations

Imagine you find an old LTO tape…

- Not every drive can read every tape.
- New LTO generation release: ~every 2-3 years.
- < LTO-7: Read=2 gen. / Write=1 gen.
- BUT: LTO-8: Read/write=**1 gen.**!

Speaker notes

LTO improvements include new LTO generations: Higher data density at same form factor. This allows the mechanics (of drives and robots) to stay compatible, with only requiring certain parts to be updated to support a new generation.

The LTO consortium guarantees a certain level of downwards compatibility. This used to be:

- Read: 2 generations down
- Write: 1 generation down

Example: LTO-6 drive should be able to read LTO 6,5,4 - but write only 6 and 5.

With LTO-8, this was reduced to 1 generation backwards compatibility, so an LTO-8 drive can not read LTO-6, only back to LTO-7

New LTO generation release: About every 2-3 years. This must be considered for migration!

This also means that you might want to keep a few drives of different generations in house?

The Cloud

Speaker notes

The term “Cloud” in computing came from marketing, and describes nothing else than digital services that already existed for over 20 years - but now had a nicer GUI and focus on convenient usability.

There’s “cloud storage” (e.g. Owncloud/Nextcloud, Dropbox, etc) and “cloud computing/services” (GDocs, etc).

In this context we will only deal with “cloud storage”.

The Cloud

Ideal for:

* Collections from any size * Institutions with limited IT support * Collections that don’t need to be accessed immediately * Fast access to smaller resolution files (streaming)

Advantages:

* Only pay for what you use * Scalable * Reduces the day-to-day management needs * AV: Low-resolution access scenarios

Disadvantages:

* HTTP access can be very slow for large files * Requires careful planning to ensure the correct services are being purchased * Depends on Internet connection * Vendor migration/lock-in

Speaker notes

No notes on this slide.

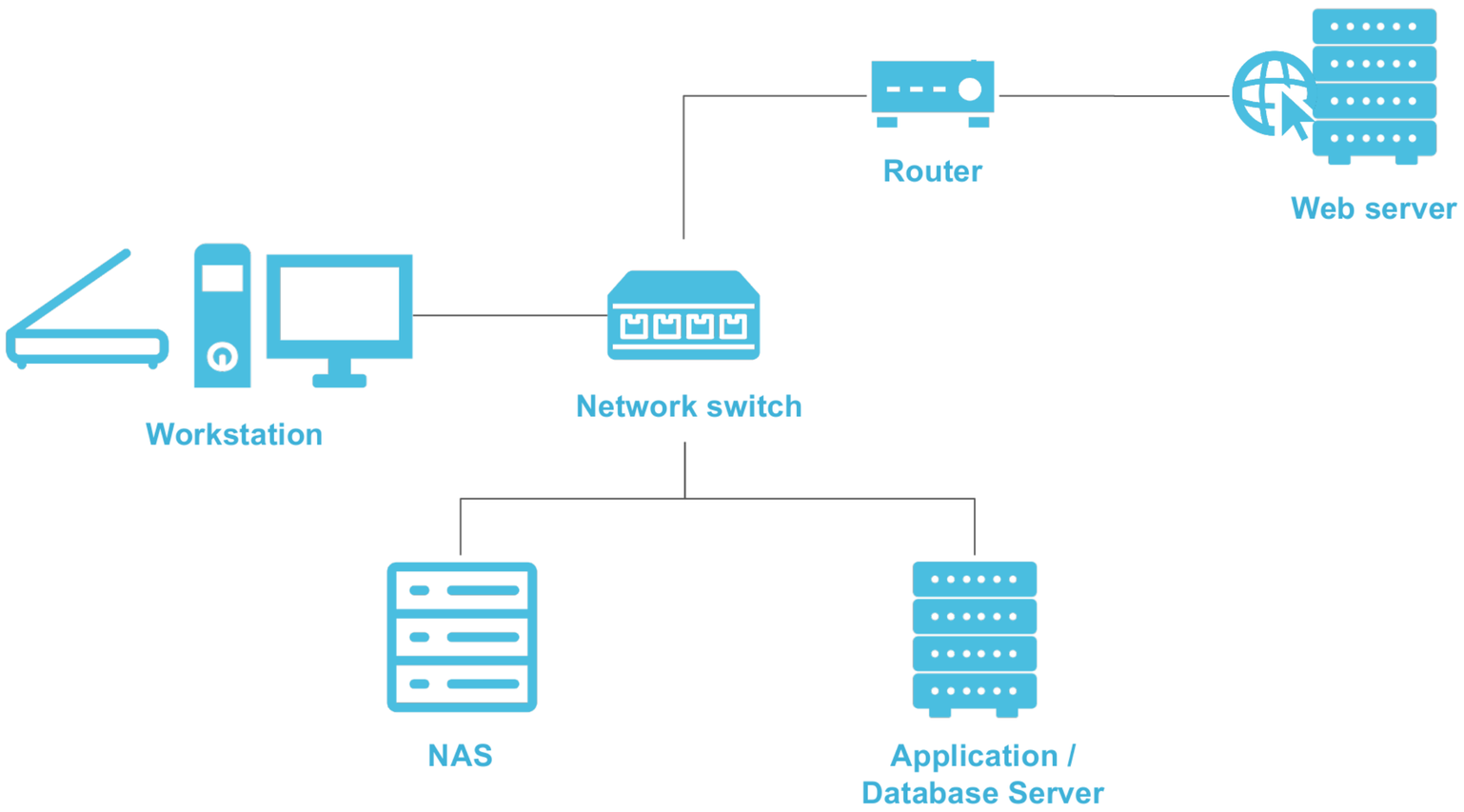
Storage Debate!

Speaker notes

Make 2 groups:

- One group for “all in-house”
- One group for “all external”
- Think of 5 “pros” why your scenario is a good choice.
- After 7 min. we’ll get back and debate.

Networks



Speaker notes

This is a simplified version of a network layout which might typically be present in preservation/ingest scenarios. It’s by no means complete, but should give a nice overview in which data resides where - and which data paths need to be travelled.

Consider: Layers!

Speaker notes

When considering a storage for long-term preservation, at least the following layers should be considered:

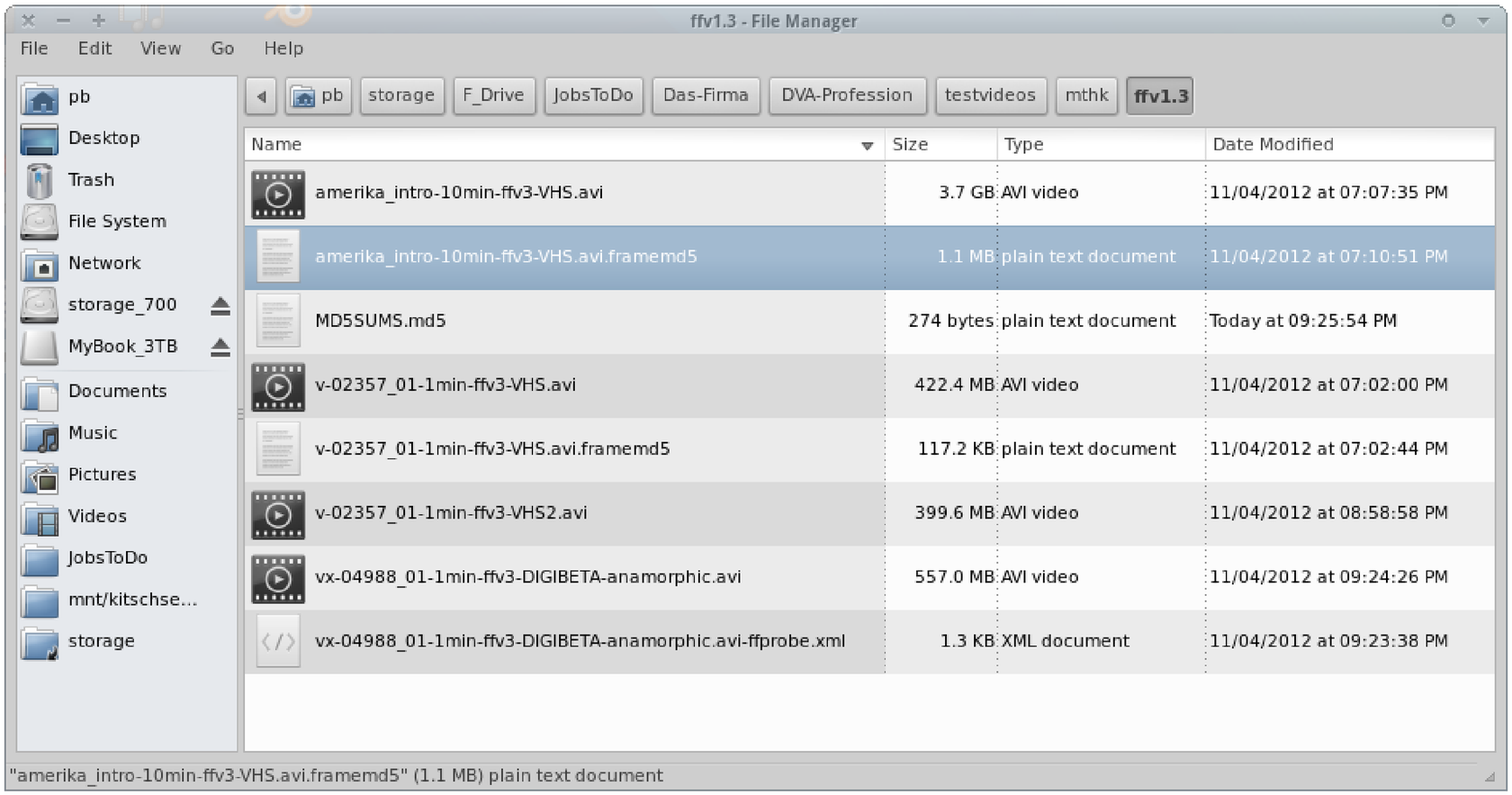
- Application required to read/write data
- Operating system
- Hardware it runs on
- Filesystem
- Drive
- Physical carrier (medium)

On top of that (but independent of the storage type chosen) are additional layers of the file itself:

- container format
- video format
- audio format
- metadata format
- …

Similar principles for selecting a file format apply to selecting a long-term storage solution. Especially: Consider getting out (=migration) before getting in - or while it is still running.

The File System



Speaker notes

No notes on this slide.

The File System?

- Raw storage = unstructured chain of bits
- File system (FS) = file/folder structure
- Different FS = different formats:
  - FAT{16,32} (Microsoft)
  - NTFS (Microsoft)
  - EXT{2,3,4} (Linux *)
  - ZFS (Sun *)
  - HFS+ (Apple)
  - LTFS (IBM *)

(*) Open formats

Speaker notes

FAT:

- FAT introduced in 1977.
- Widespread, well-known and documented
- Default for SD cards and embedded devices
- FAT32: 32bit limitation (max 4GB filesize, limited max. device size)
- No access permissions (=everyone can read/write everything)
- Very simple and straightforward filesystem
- Non-free / license fees by Microsoft.

NTFS:

- Introduced in 1993.
- max volume size (different implementations): 256 TiB (pre 2016), now 8 PiB (2019 or later)
- Non-free / license fees by Microsoft.

EXT:

- ext2 introduced in 1993.
- ext4 standard for Linux-based OS.
- ext4 and NTFS supported by digital cinema standard. (Because the projectors probably run Linux ;))
- max volume size: 1 EiB (1024 PiB)
- No licensing fees

ZFS: * Introduced in 2005 * More than “just a filesystem”: combines RAID (redundancy, handle harddisk failure) Logical Volume Management (LVM) and filesystem in one. * Integrated integrity checks * (recently) well known among IT administrators of unixoid systems. * Requires slightly more experience to handle (because of complexity. * Very interesting filesystem (also for preservation). Too much to describe here. You may want to read more on Wikipedia :)

HFS+: * HFS Introduced in 1985. * max volume size: 8 EB * Successor of “HFS” * Officially only supported on Apple OS

LTFS: * LTFS first prototype (Linux and Mac OS X) during 2008/2009 * Only for LTO tapes. * See separate page/slide on LTFS for details.

LTFS: Linear Tape File System

- Open specification = vendor neutral
- Better for preservation, but may not support “convenience” features.
- All implementations must:
  - Correctly read media that was compliant with any prior version.
  - Write media that is compliant with the version they claim compliance with.

Speaker notes

The Linear Tape File System (LTFS) is a good solution to avoid vendor lock-in by tape filesystems. Please make sure that you can read the data written onto the tapes under the following conditions:

- Different drive
- From different vendor
- With (FOSS) software (and/or from different vendor)

Otherwise, you won’t really know if you can access your data as technology- and vendor-neutral as desired.

By default, different applications may write data in their own way on LTO tapes. This allows them to implement some non-standard features that often provide more “convenience” by offering additional features. Using such vendor-specific features, instead of LTFS has the downside of reducing the possibilites to read that data under different conditions in the future.

Therefore it is advised to use LTFS for long term preservation use cases.

File system: Disaster relevant?

- Deleted files are still there (maybe fragmented).
- Different FS = different error resilience and recovery options.
- Does it scale? (moving files is when they're most vulnerable…)
- Tools/knowhow to deal with recovery of broken filesystems in your setups?
- Logical Volume Management (LVM) snapshots

Speaker notes

- Deleted files: Deleting files is just flagging them in the index and their disk space is marked as available again. The data is still where it was - until a new write operation (=creating new data) uses the same space and overwrites the old data.

  Related articles:

  - Get Your Data Back with Linux-Based Data Recovery Tools
  - [How to Install and Use TestDisk Data Recovery Tool in Linux](https://www.tecmint.com/install-testdisk-data-recovery-tool-in-linux/)
  - Scalpel: Extract lost files from disk/card image

- Different FS have different error resilience and recovery options: Go ask your IT guys! ZFS is pretty awesome, but "simple and well known" is also nice for recovery.

- FS recovery requires understanding the FS format. Got documentation? Source code?

- Logical Volume Management (LVM) Remember VM-snapshots? You can do the same on storage level (below filesystem): LVM Snapshots

- Does it scale?

  - Total number of files/folders?
  - Length of path?
  - Suitable for NAS/SAN/cluster?
  - FS vs Blocks vs Object-based?

Errors? Backup!

Speaker notes

Why Backup?

- Mechanical or electrical error
- Physical damage
- Aged technology (wear)
- Human error
- …

Storage Failure can result in:

- Total loss
- Data errors/corruption (bitrot)

The 3-2-1 Backup Rule

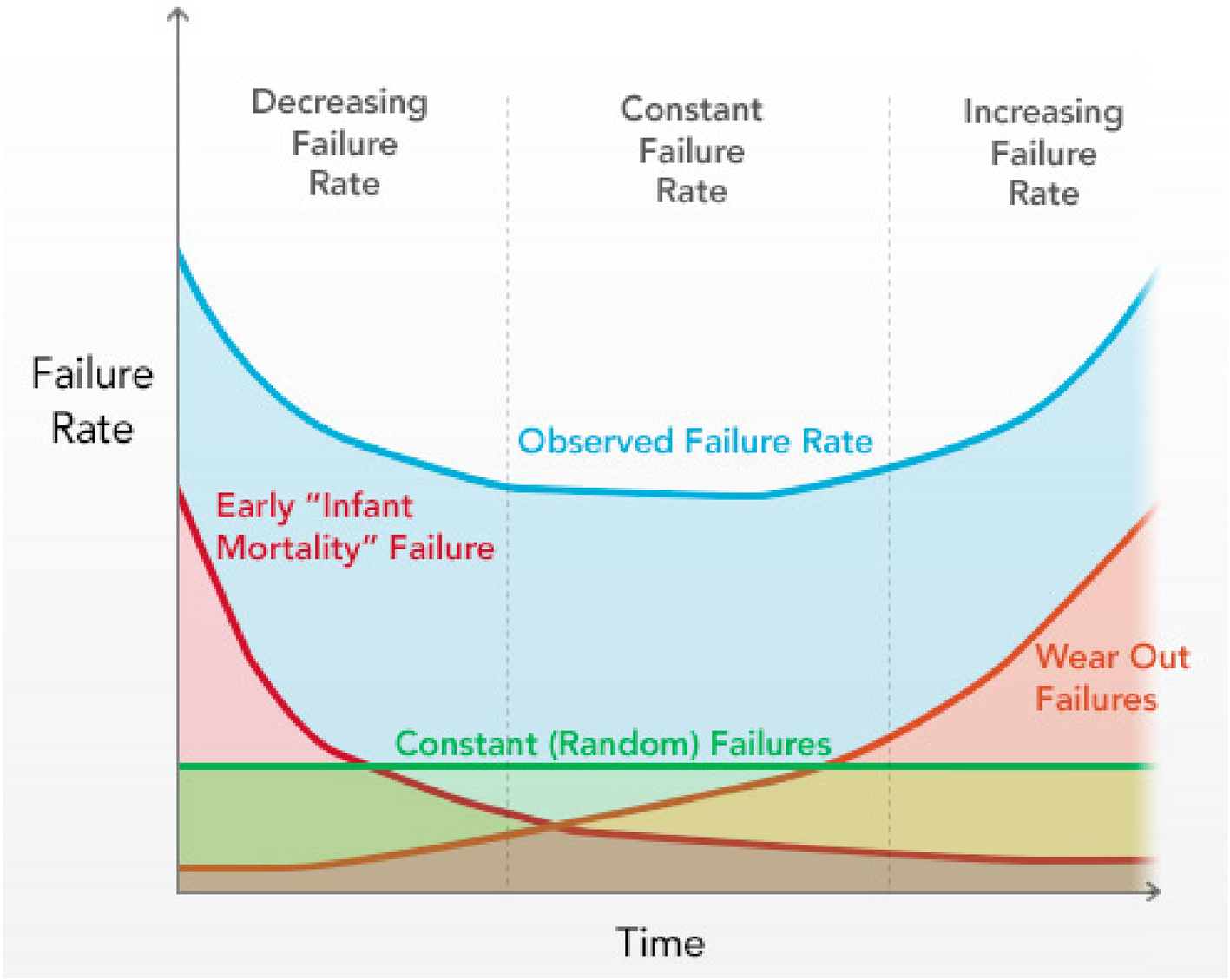
- Keep at least **three** copies of your data.
- Store **two** backup copies on different devices or storage media.
- Keep at least **one** backup copy offsite.

Speaker notes

That’s not a law, it’s a rule :) It’s good however to try to implement as much of it as you can.

If you can only do “2-1”, it’s better than not trying at all!

Statistics of HDD failure



Speaker notes

No notes on this slide.

The Story of ToyStory2

https://www.youtube.com/watch?v=8dhp_20j0Ys

Speaker notes

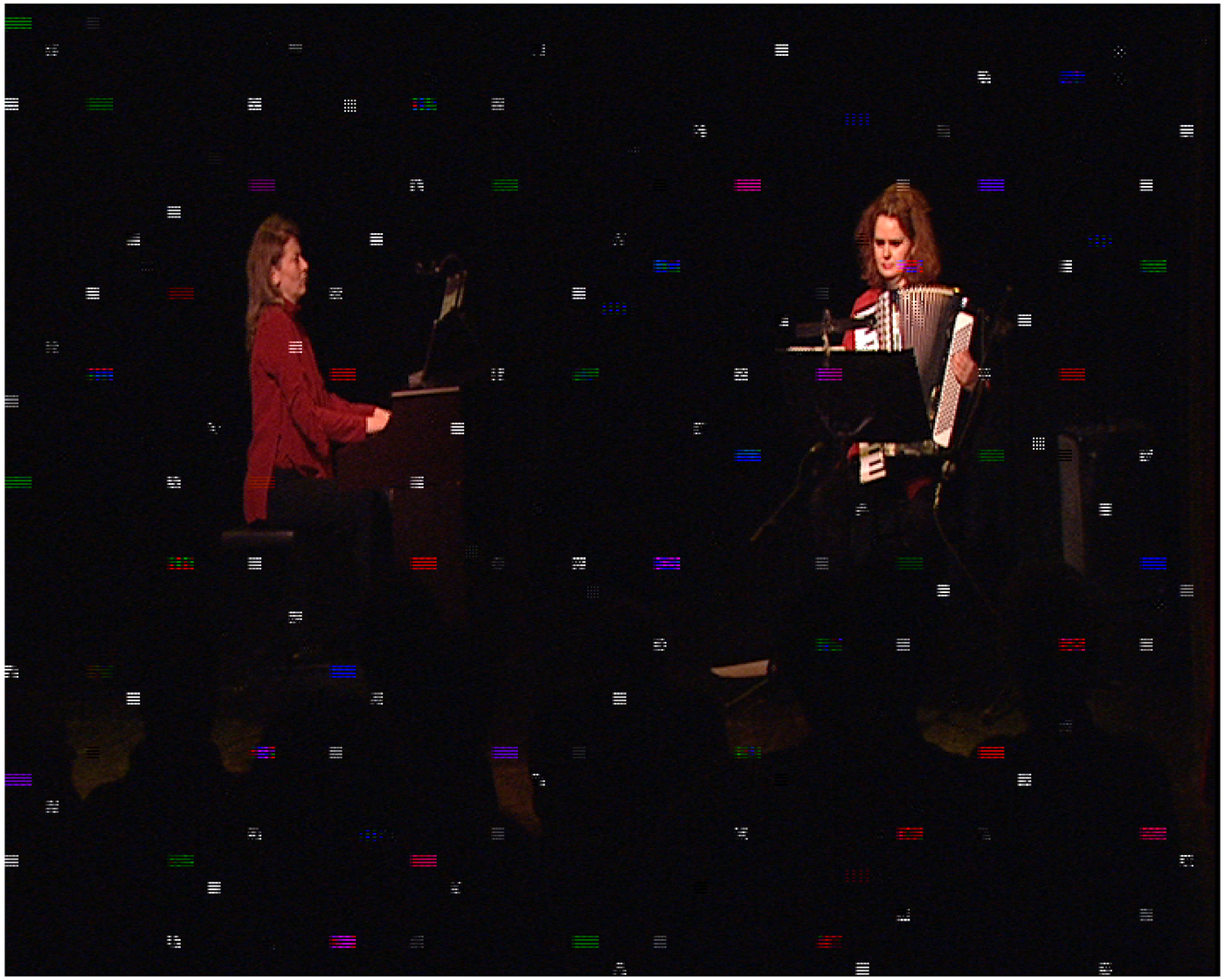
No notes on this slide.

Data errors

Speaker notes

The impact of individual broken (aka “flipped”) bits can be completely different: From absolutely unnoticable to complete inaccessibility - and everything in between.

Digibeta Dropouts

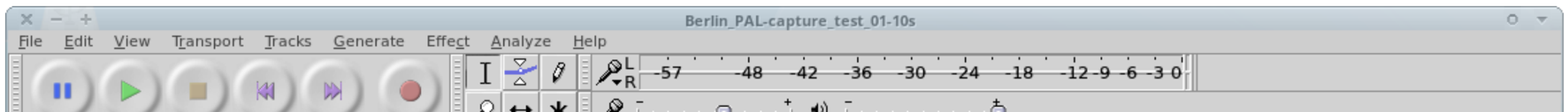


Speaker notes

That’s actually a “digital magnetic tape” storage failure. No folders/files, but a stream of digital data that represents the video image.

Also known as “Digibeta confetti”.

Audio Bit-Errors



Speaker notes

The spike in the sine wave is clearly indicating a wrong sample value at that position. This artefact is an audible, static-sounding, very short “click!”. It can go unnoticed if it only a handful of bits are broken over the whole duration of the recording.

Small Bit, Big Problem

JPEG header “signature” = FF D8 FF DB

- 0xFF = 0b11111111
- 0xBF = 0b10111111

Speaker notes

When a single bit, at a very wrong position is flipped/broken the impact can be great: Changing this very bit of a JPEG image makes it impossible to open it with most applications, as they are unable to identify it anymore.

Proof of concept: I’ve prepared a modified image file to “restore” using a hex editor to manually fix that bit.

Corruption / Bit rot

> *“Bit rot can be caused by a number of sources but the result is always the same – one or more bits in the file have changed, causing silent data corruption. The ‘silent’ part of the data corruption means that you don’t know it happened – all you know is that the data has changed (in essence it is now corrupt).” —* *Jeffrey B. Layton, Linux Magazine, June 2011*

Speaker notes

No notes on this slide.

Data scrubbing, Fixity checking

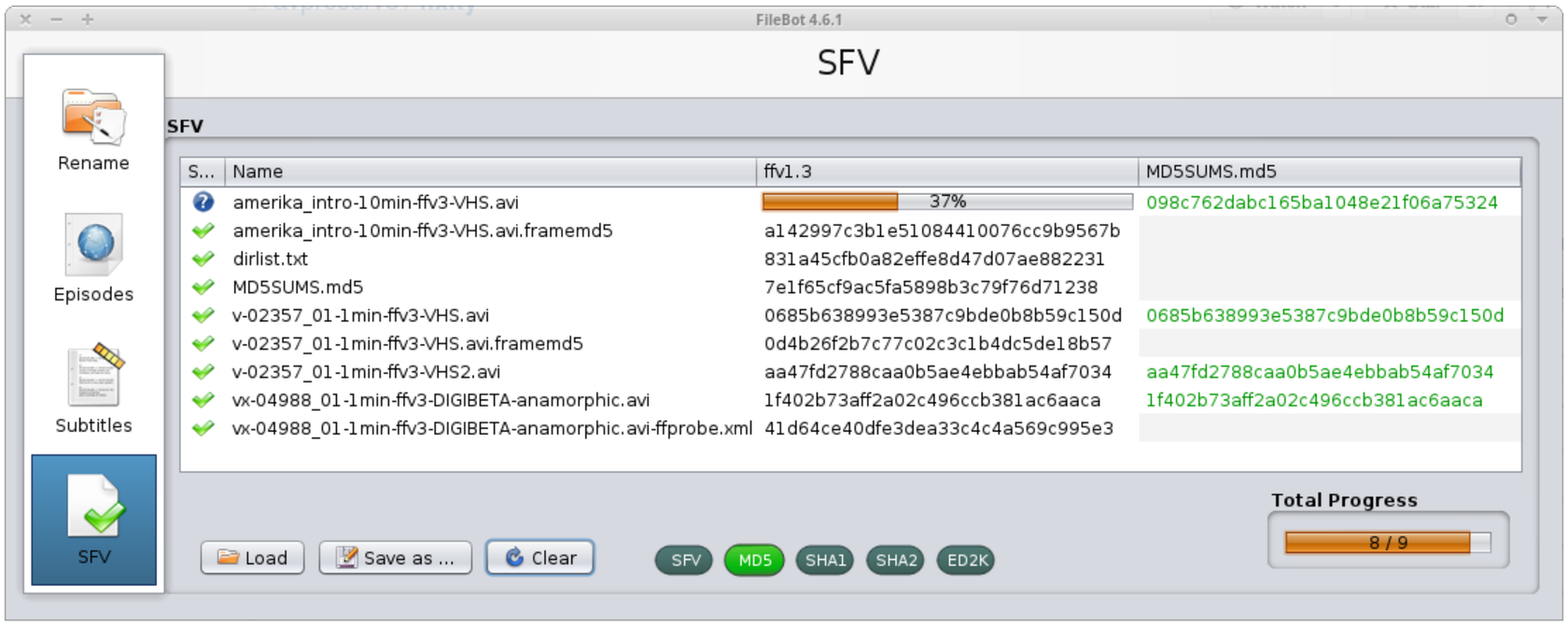
How do you know your data is intact?

Speaker notes

We’ve already created and used hashcodes in the ingest session. These fixity manifest files can be used to validate your storage data.

And don’t forget: Your backup needs to be validated (every now and then) too!

A check a day keeps the bitrot away…



Speaker notes

Daily fixity checks might be too often (unreasonable). You will want to find a timing sweetspot for when and how often to check your storage (and backups). Don't wait too long, and document the findings: issues as well as intact fixity checks, so you know "when" (and therefore maybe "why") a file got corrupted during its lifetime.

In this image you see just one example of a simple GUI that shows an ongoing calculation for the validation process of comparing the current data to existing hashes, previously stored in a plain text manifest file "MD5SUMS.md5".

In this example, only the media files initially had hashcodes.

Headcount

Hashcode manifest files can also be used to check if all files expected are present or additional ones exist that are unaccounted for.

Speaker notes

No notes on this slide.

Storage: Challenges / Risks

- Storage media failure
- Obsolescence
- Humans
- Catastrophes / War

Speaker notes

- Humans:
  - A “classic”: Accidential deletion
  - Not so common: Malicious intent
  - Catastrophes / War Geographically separate locations can help here. War: Should be avoided at all costs. Be creative!

RAID

Redundant Array of Inexpensive Disks

> *“[…] is a data storage virtualization technology that combines multiple physical disk drive components into one or more logical units for the purposes of data redundancy, performance improvement, or both.”*

Source: Wikipedia: RAID

Speaker notes

- Good ‘old’ friend: exists since before 1988
  - Here you see a classical hardware RAID chassis.
  - This is just one option.

RAID

- Different RAID levels:
  - Stripe = Fast, but dangerous!
  - Mirror = Perfect for OS system disks
  - RAID5/6 = 1 or 2 disks fault tolerance (parity bits)
- RAID is *not* a backup.

Speaker notes

Data replication is not a Backup: It is not unusual to discover an undesired modification or error only later on.

- Sync:
  - File changes (new, modification) are immediately replicated
  - Unwanted actions, too! (wrong modification or deletion)
- Async:
  - Allows you to go back in time.
  - Allows multiple copies/revisions of the same data.
  - “Long” time between new/modification and the copy.

Sad, but true:

- With > 4TB disks, RAID-6 may be insufficient…
- Long rebuild times may “burn out” the left-over disks.
- ZFS supports 3 disks parity, but … future?

Speaker notes

The algorithm RAID-6 uses to provide data redundancy for rebuilding in case of harddisks failing, may have come to an end with HDD capacities beyond about 4TB per-disk.

The reason for that is, that in case of a disk failure, the RAID mechanism reads the parity data from all remaining disks to reconstruct the one that “has gone”. With larger disks, the time for this rebuild to happen is getting longer. During that period, all remaining disks are constantly “stressed” until the required data has been read and the new disk rewritten.

Considering that disk failures more likely occur in RAID arrays that have been running for quite some time, it can be assumed that the other disks may be stressed too hard during that rebuild, causing other disks to fail due to the recovery - resulting in a complete data loss.

Cluster & replication

- Multiple copies on multiple servers (“cluster nodes”)
- Synchronized (replicated) automatically
- Off-site replication possible
- Replication is *not* a backup!
- What if a node goes down?
- How does “the cloud” do it?

Speaker notes

Where RAID provides failover redundancy for individual storage devices, a storage cluster may provide failover redundancy by means of whole PCs.

Each PC is called a “node” and is designed as a component that may fail, while the rest of the cluster can maintain their services. This is how cloud storage/service providers provide transparent failover setups.

Replication is not a backup.

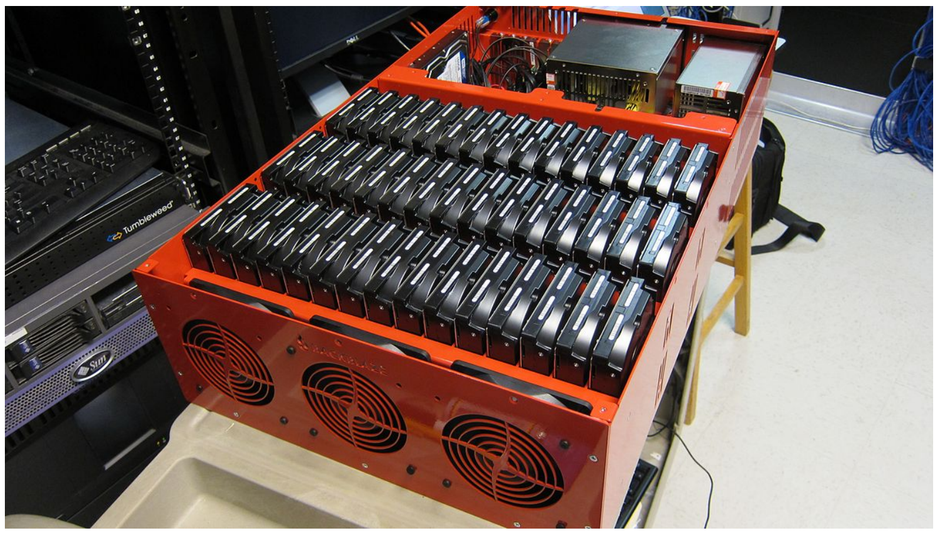
Replication mechanisms try to keep all their data in-sync as quickly and seamlessly as possible. This means that if you delete a file, it is usually not known by any replicating system if that was on purpose on in error.

Therefore, the replication mechanism happily will synchronize this change (=the deletion) to all of its nodes. Voila.

You get it, why this would be a bad backup option? ;)

Inside = The same stuff

- Combining a mix of most (if not all) the above mentioned technologies.
- Mostly unix-like operating systems (e.g. Linux).
- And lots of carriers.
- maintaining it = other people's problems.



Speaker notes

No notes on this slide.

So what if cluster nodes fail?

- Another node takes over.
- Seamlessly. Transparently.
- The faulty node is removed & replaced.
- Life continues.

Speaker notes

In case a node fails, another node jumps in to take over its job(s). The failing node is replaced in due time.
Comparable to RAID setups, but on another abstraction layer.

Of course they need some kind of “magic” (=algorithms, data protocols, mechanisms, applications, etc) to do so.

There are many different cluster operating systems and applications, some of

Erasure Coding

“protects data from multiple drives failure, unlike RAID or replication. For example, RAID6 can protect against two drive failure whereas in MinIO erasure code you can lose as many as half of drives and still the data remains safe.”

Source: MinIO Erasure Code Quickstart Guide

“Compared to data replication, erasure-coding approaches have better performance at reducing storage redundancy and data recovery bandwidth.”

Source: Reliability Assurance of Big Data in the Cloud (2015)

Speaker notes

Erasure coding is a different (algorithmic) approach to provide data redundancy to counteract device/node failures.

Quote from the Minio website:

“Erasure code is a mathematical algorithm to reconstruct missing or corrupted data. MinIO uses Reed-Solomon code to shard objects into variable data and parity blocks. For example, in a 12 drive setup, an object can be sharded to a variable number of data and parity blocks across all the drives - ranging from six data and six parity blocks to ten data and two parity blocks.

By default, MinIO shards the objects across N/2 data and N/2 parity drives. Though, you can use storage classes to use a custom configuration. We recommend N/2 data and parity blocks, as it ensures the best protection from drive failures.

In 12 drive example above, with MinIO server running in the default configuration, you can lose any of the six drives and still reconstruct the data reliably from the remaining drives.”

More math at: Wikipedia: Erasure Code

Quote from Wikipedia:

“Example: In RS (10, 4) code, which is used in Facebook for their HDFS,[4] 10 MB of user data is divided into ten 1MB blocks. Then, four additional 1 MB parity blocks are created to provide redundancy. This can tolerate up to 4 concurrent failures. The storage overhead here is 14/10 = 1.4X.

In the case of a fully replicated system, the 10 MB of user data will have to be replicated 4 times to tolerate up to 4 concurrent failures. The storage overhead in that case will be 50/10 = 5X.

This gives an idea of the lower storage overhead of erasure-coded storage compared to full replication and thus the attraction in today’s storage systems.”

Erasure Coding

- More parity possible than RAID
- Data can be spread across network cluster nodes.
- Does not suffer from rebuild “burn out”.
- More complex to set up & compute.
- Not that widely known/supported/used yet.

Speaker notes

Links:

- MinIO: Storage usage
- MinIO: Erasure Code Quickstart Guide
- Erasure coding vs RAID: Data protection in the cloud era (ComputerWeekly)
- RAID Vs. Erasure Coding (NetworkComputing)
- What is erasure coding? (VirtuService)
- Isilion.com: Erasure Coding (Isilon)
- Erasure Coding vs. RAID: Which Is Right and When? (ITProToday)
- Wikipedia: Erasure code

Object storage

- Files become “objects”.
- There are no folders (as we’re used to)
- There’s just an identifier.
- Arbitrary metadata may be assigned to each “object”.
- Scales infinitely (theoretically).
- Clusters of block storages with a clever abstraction layer.

Speaker notes

You may have been using object based storage on Google Drive or Amazon S3.

Pretty cool, but it handles very differently from any regular file/folder based storage! Please do your homework before deploying this.

It is still pretty new and if done wrong, it can cause more troubles/effort than a regular, hierarchical FS storage.

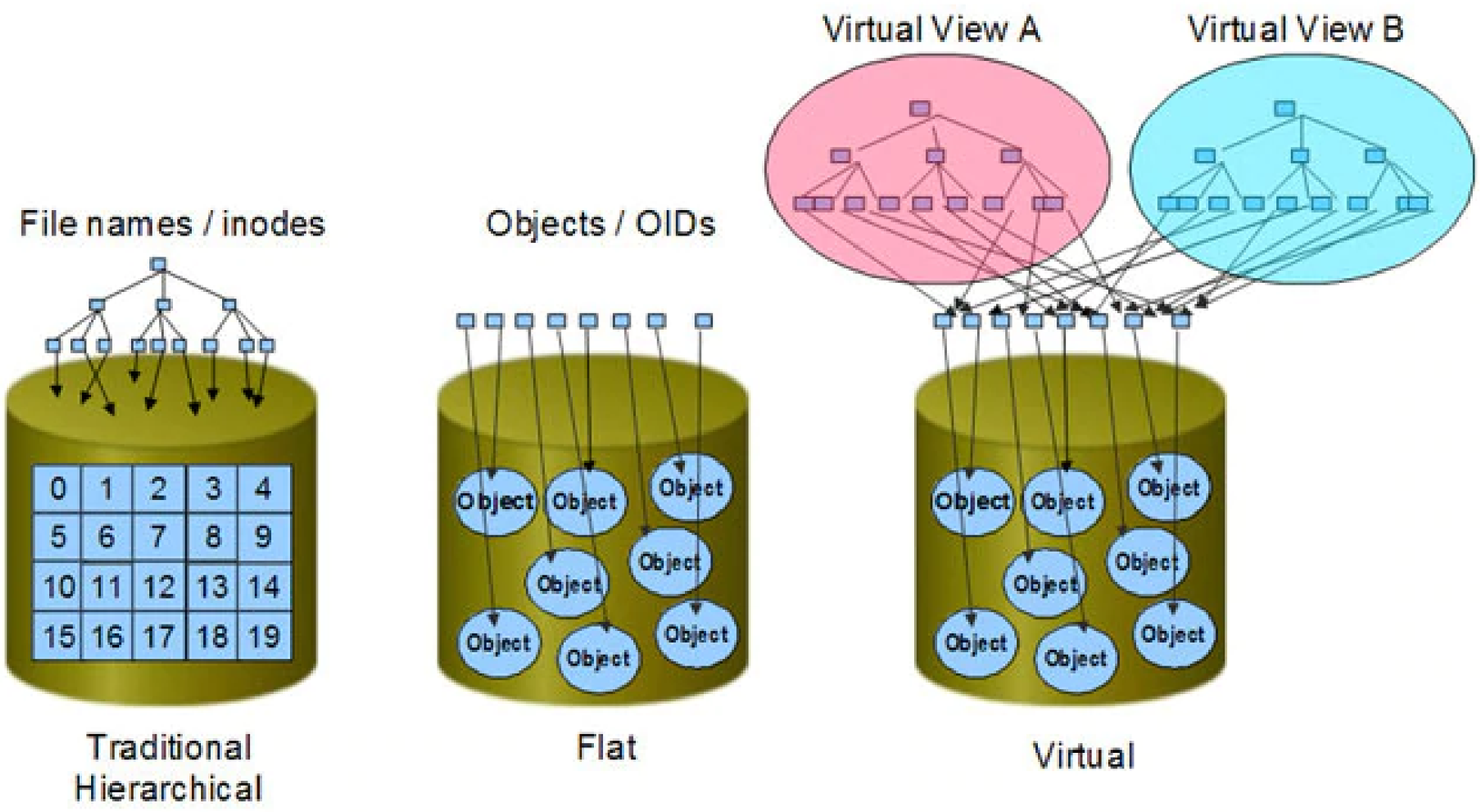
What is often left out by marketing:

- Underneith, it is still using block storage.
- Just with clever abstraction layer(s).
- Important: Your toolchain needs to be object-storage-aware! Regular, well-known tools cannot easily access object files as it used to be.
- I mention it here, because if done right, it scales.

I’m pretty sure, this is the future - and (at least large scale) storage will be going from classical (filesystem-based) to this (=object based) way.

Object storage

Beyond classical, hierarchical filesystems



Speaker notes

No notes on this slide.

Object storage = The future?

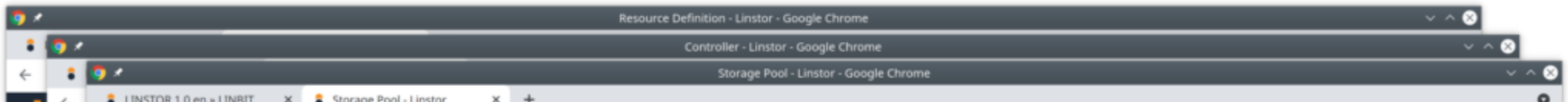
Object based storage may very likely replace hierarchical filesystems, but things need to be rewritten/adapted to properly support it.

It’s not yet plug-compatible with existing programs.

Speaker notes

No notes on this slide.

Software Defined Storage (SDS)



Speaker notes

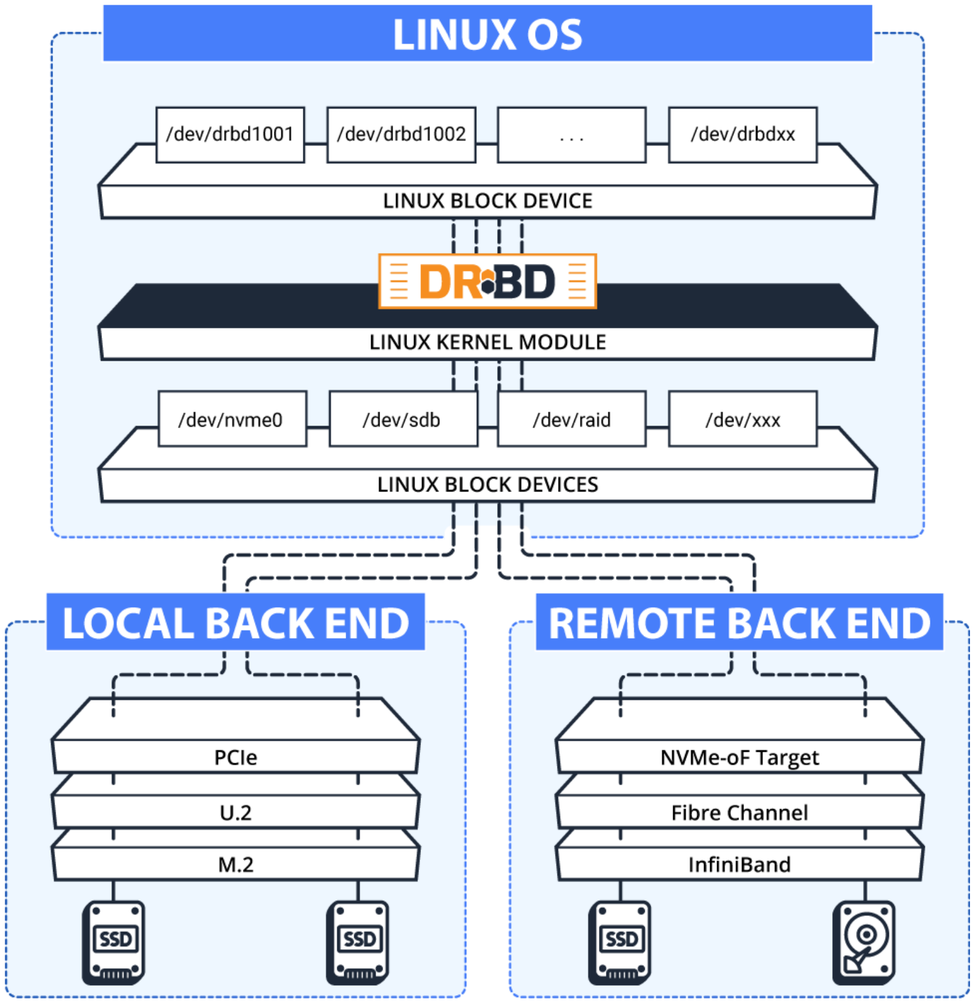
Quote from Wikipedia “Software-defined storage”:

“Software-defined storage (SDS) is a marketing term for computer data storage software for policy-based provisioning and management of data storage independent of the underlying hardware.

Software-defined storage typically includes a form of storage virtualization to separate the storage hardware from the software that manages it.

The software enabling a software-defined storage environment may also provide policy management for features such as data deduplication, replication, thin provisioning, snapshots and backup.”

Again: ~~Onions~~, eh Layers.



Speaker notes

Quote from Linbit Website:

> *“Since the software simply presents a block device, users do not need to use complex, shared file systems. Because we can use a single node with a filesystem on top of a block device, backing disk failure is transparent to the end user.”*

Large storage that provides flexibility/scalability, the required features (distribution, replication, backup, etc, etc) - and transparency for the applications accessing the files means stacking up more layers.

Even if it ain’t rocket science, it still is far from trivial (especially setting it up, maintaining and and debugging it).

The Cloud

- No internet = no service.
- Make sure your online bandwidth is sufficient.
- Have an exit plan (+contract?) for migration.
- What if their conditions change?
- Where is your data actually stored?
- Does it matter for you? (e.g. legally)
- Again: Consider mixing.

Speaker notes

Storing data in large data centers is very convenient. All the same challenges regarding large-scale storage still apply, but they’re someone else’s problem now :)

What’s still relevant for you though is:

- What happens if the provider changes their conditions?
- What happens in case of unstable political situations, war, etc?
- What if you want to migrate your stuff to another storage/provider?
- Does it matter (e.g. legally) if the provider could access your data?

Does it scale?

Speaker notes

The image shows one row of Backblaze storage pods in a computing center. This is just to give you an idea that:

- The Cloud literally is just a bunch of computers (and storage media inside them).
- Large volumes of data need large volumes of space. And electricity, knowhow and spare parts and staff.
- However, the underlying mechanisms and applications are designed to scale.

Does it scale?

- What if you run out of diskspace?
- Need to copy/move everything?
- Or can you simply add “more space”?
- How to know if everything is still “there” and intact?

Speaker notes

In the past with RAID, it was very common that enlarging the storage space meant: get a bigger RAID with bigger disks, then copy everything over and switch to the new storage.

With data sizes currently common for professional AV content, this approach is a very delicate one that puts your data in quite some danger, as the time where data is the most vulnerable is when it’s being moved.

More Storage Terms

- S.M.A.R.T. Self-Monitoring, Analysis and Reporting Technology
- NAS Network Attached Storage
- SAN Storage Area Network
- RAID Redundant Array of Inexpensive Disks
- Object Storage

Speaker notes

You don’t have to become a storage expert, but It’s good to have heard these terms and roughly know what they mean. This selection of terms is by faaar not complete, but these are quite likely to appear in a DLTP context.

Good practice

- At least 1 backup (=2 copies)
- Preferrably 2 backups (=3 copies)
- Geographic separate locations (with different threat profile)
- Mix storage media
- Migrate timely and with a plan (~5-7 years)
- Use the right tech for *your* needs
- Periodically check fixity of content **and** backup
- Work with IT to implement and maintain technology.

Speaker notes

No notes on this slide.

Comments?

Questions?

Speaker notes

No notes on this slide.

Links

- Difference between SSD and HDD
- Flash Storage Memory Guide
- SSD Lifespan
- Data degradation (Wikipedia)
- HDD vs SSD: What Does the Future for Storage Hold? (Backblaze)
- Hard Drive SMART Stats (Backblaze)
- How long do disk drives last? (Backblaze)
- Hard drive test data (Backblaze)

Speaker notes

No notes on this slide.