Topic 5 - Archival Storage

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.

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:

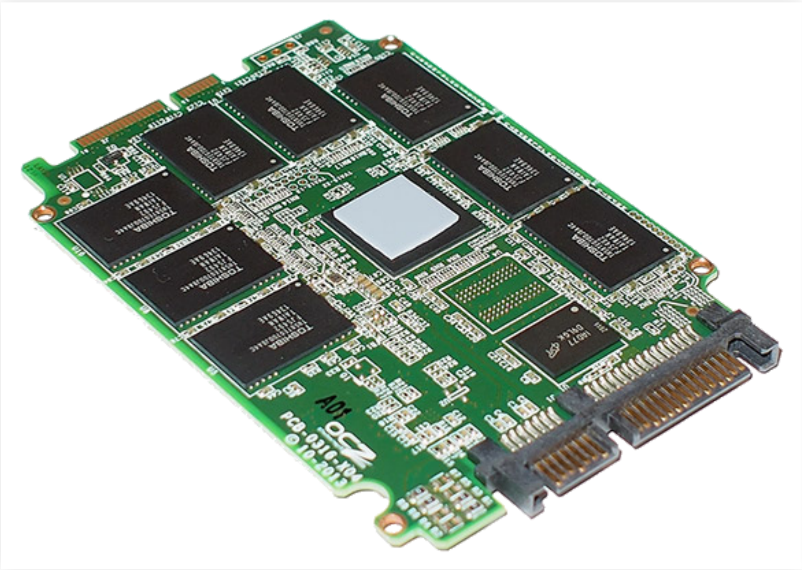
- Read: 2 generations down
- Write: 1 generation down

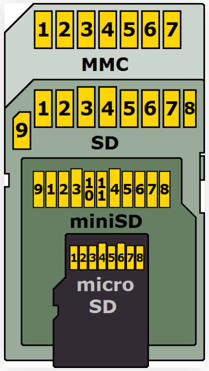
Example: LTO-6 drive should be able to read LTO 6,5,4 - but write only 6 and 5.

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:

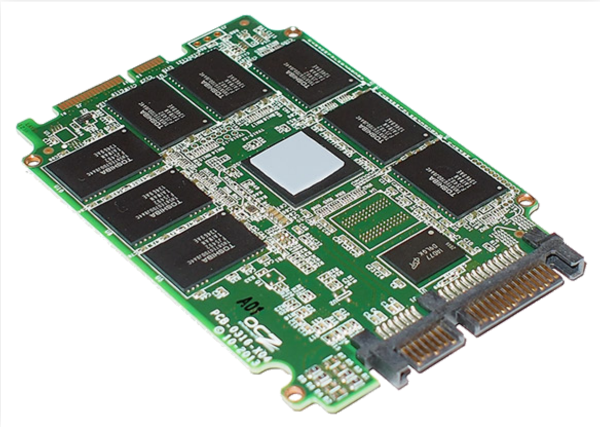
- cost/size ratio: getting cheaper, but still more expensive than HDD
- No moving parts = no mechanical wear
- SD cards: almost no electronics in the carrier
- Speed: Can be extremely fast

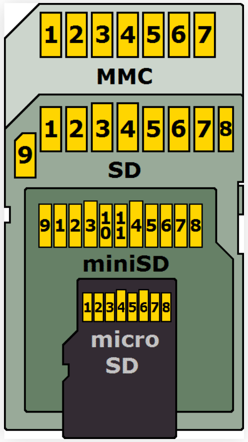
Cons:

- 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.

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)

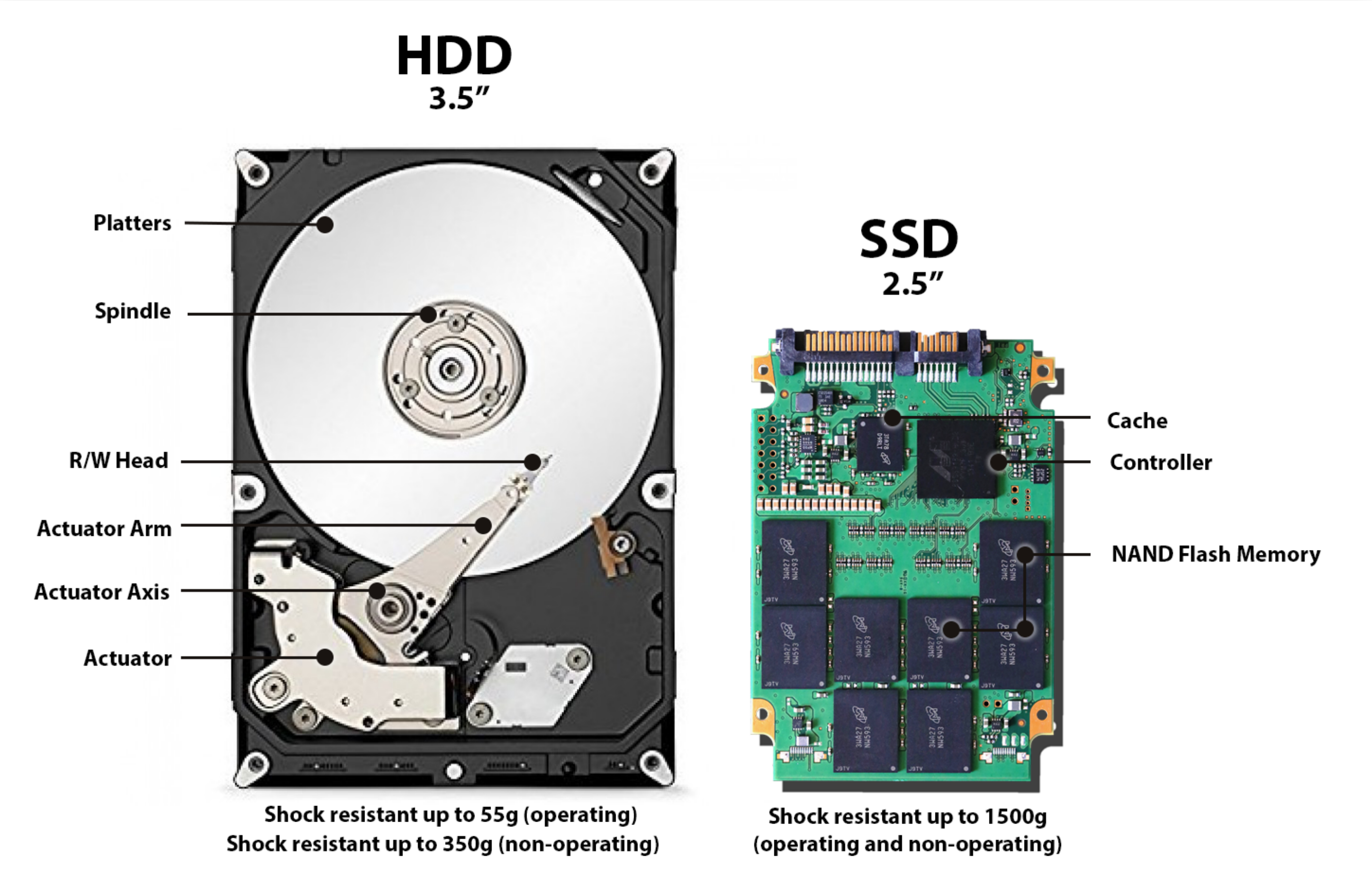
It's usually good to mix storage media. For example for different use cases:

- 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


HDD
3.5"
Platters
Spindle
R/W Head
Actuator Arm
Actuator Axis
Actuator
Shock resistant up to 55g (operating)
Shock resistant up to 350g (non-operating)
SSD
2.5"
Cache
Controller
NAND Flash Memory
Shock resistant up to 1500g
(operating and non-operating)

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

Storage Types

External Hard Disks

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

Network Attached Storage (NAS)

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

Data Tape


C7976A 6.25TB MP RW
H P 3 4 5 6 L6
Ready
Drive Error
Tape Error
Clean
Encryption

Data Tape

Ideal for:

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

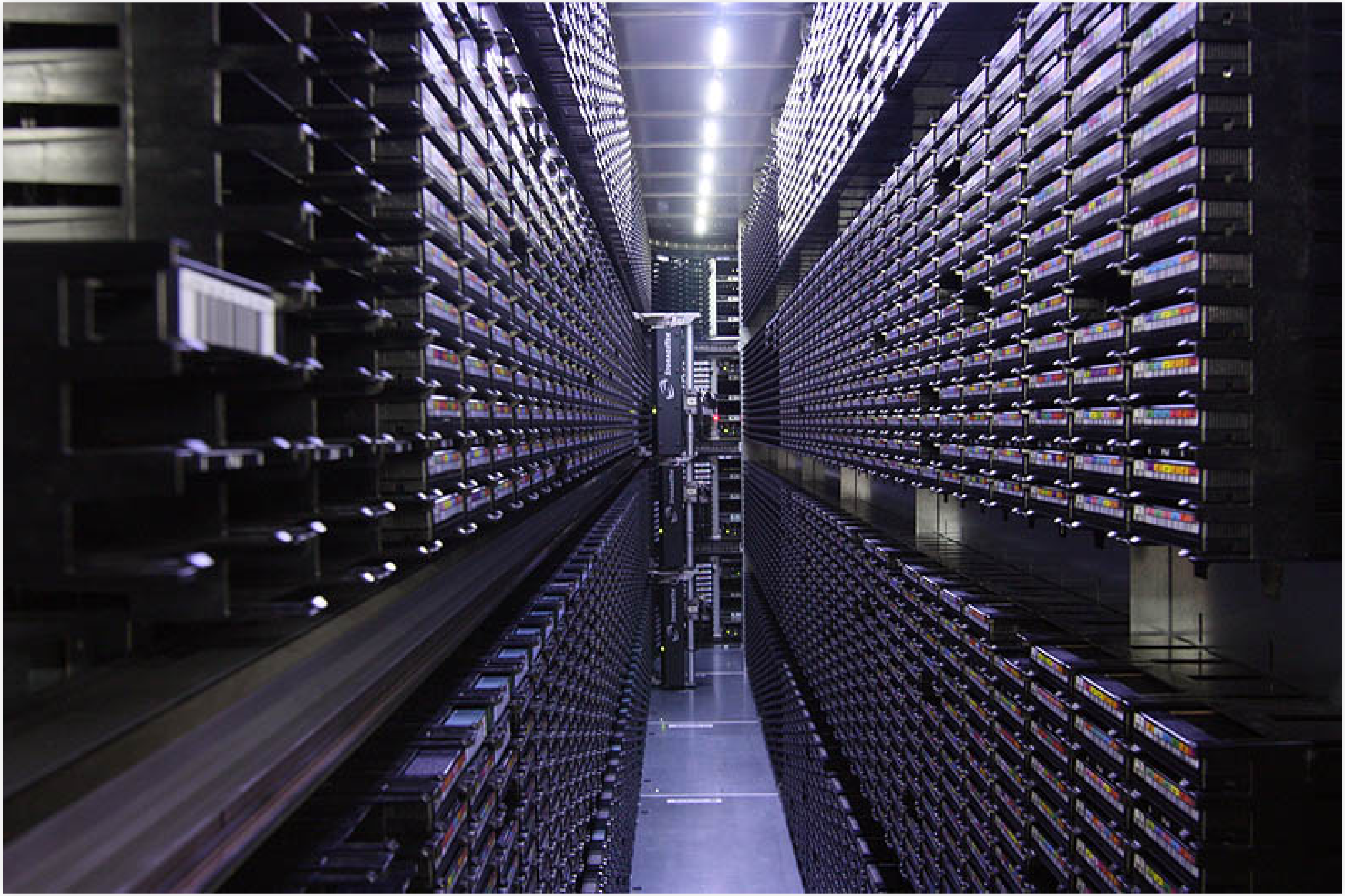
Advantages:

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

Disadvantages:

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

Data Tape Library (Robot)



The Cloud


There is NO CLOUD, just
other people's computers
fsfe.org

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)

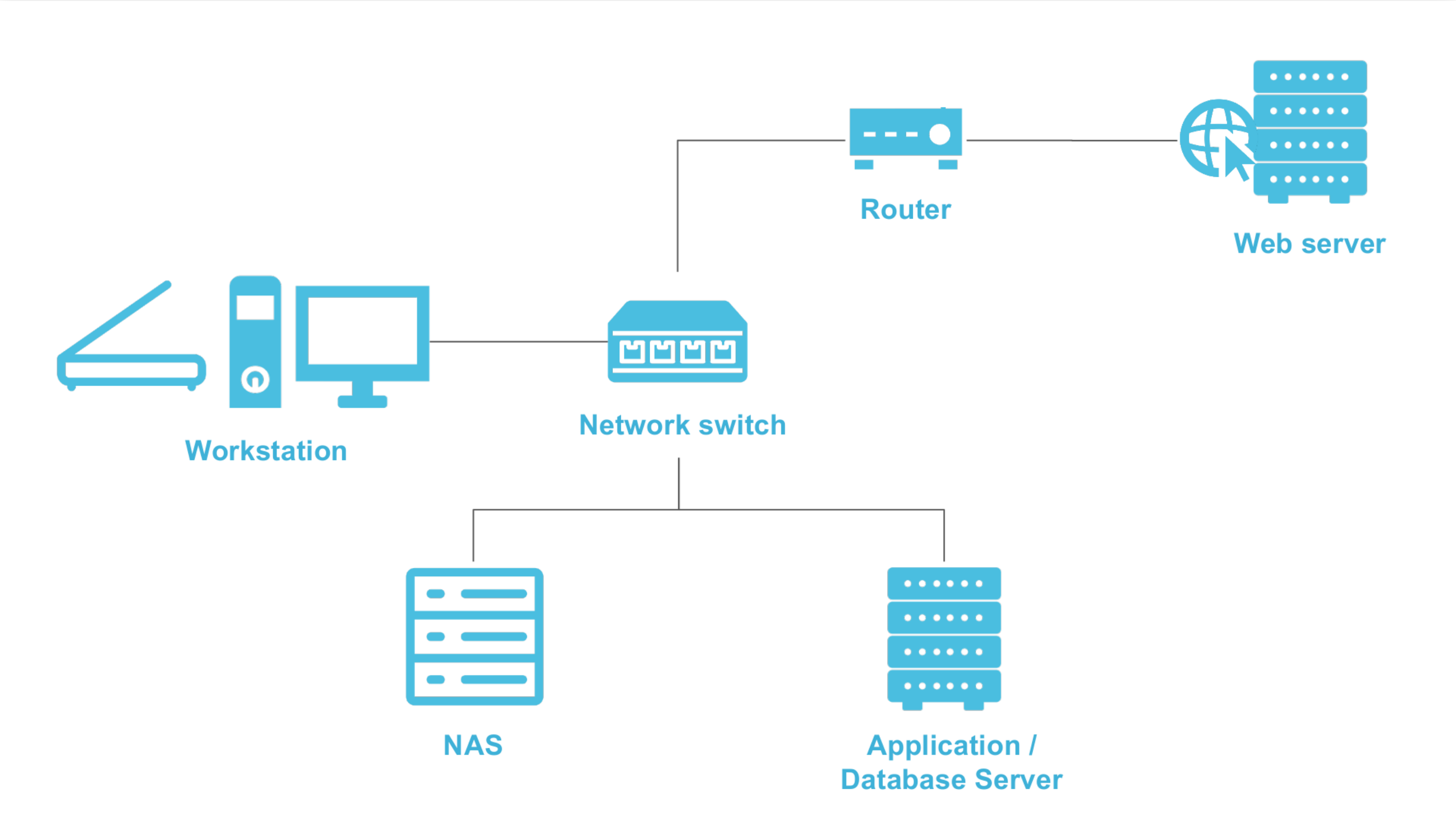
Advantages:

* 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

Networks



Router
Web server
Workstation
Network switch
NAS
Application /
Database Server

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!


"Onions have layers, Ogres have layers. You get it? We both have layers!"

Speaker notes

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

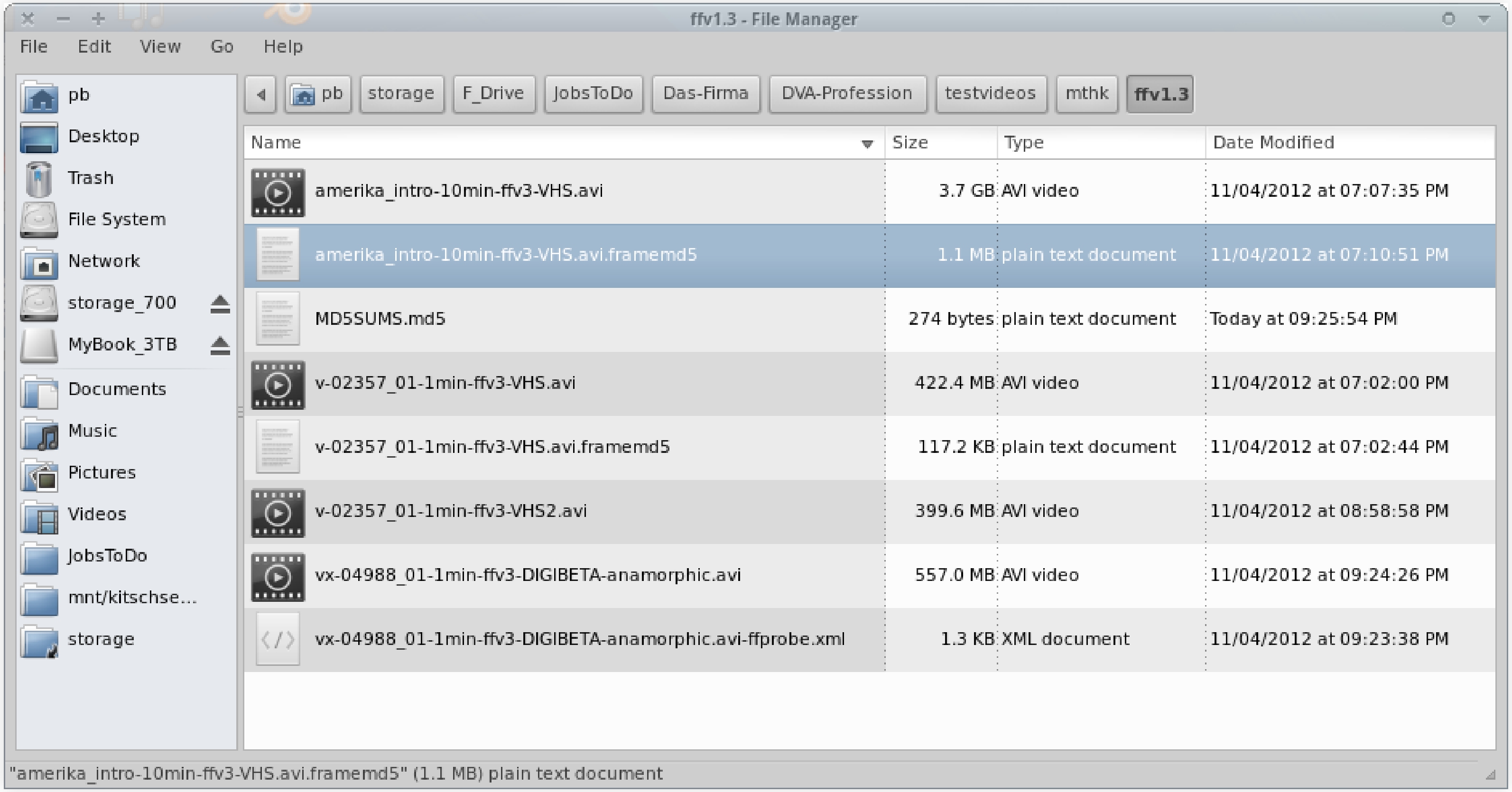
- Application required to read/write data
- Operating system
- Hardware it runs on
- Filesystem
- Drive
- Physical carrier (medium)

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



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 "comfort" 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, vendors may write data in their own way on LTO tapes. This allows them to implement some non-standard features that often provide more "comfort" 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.

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

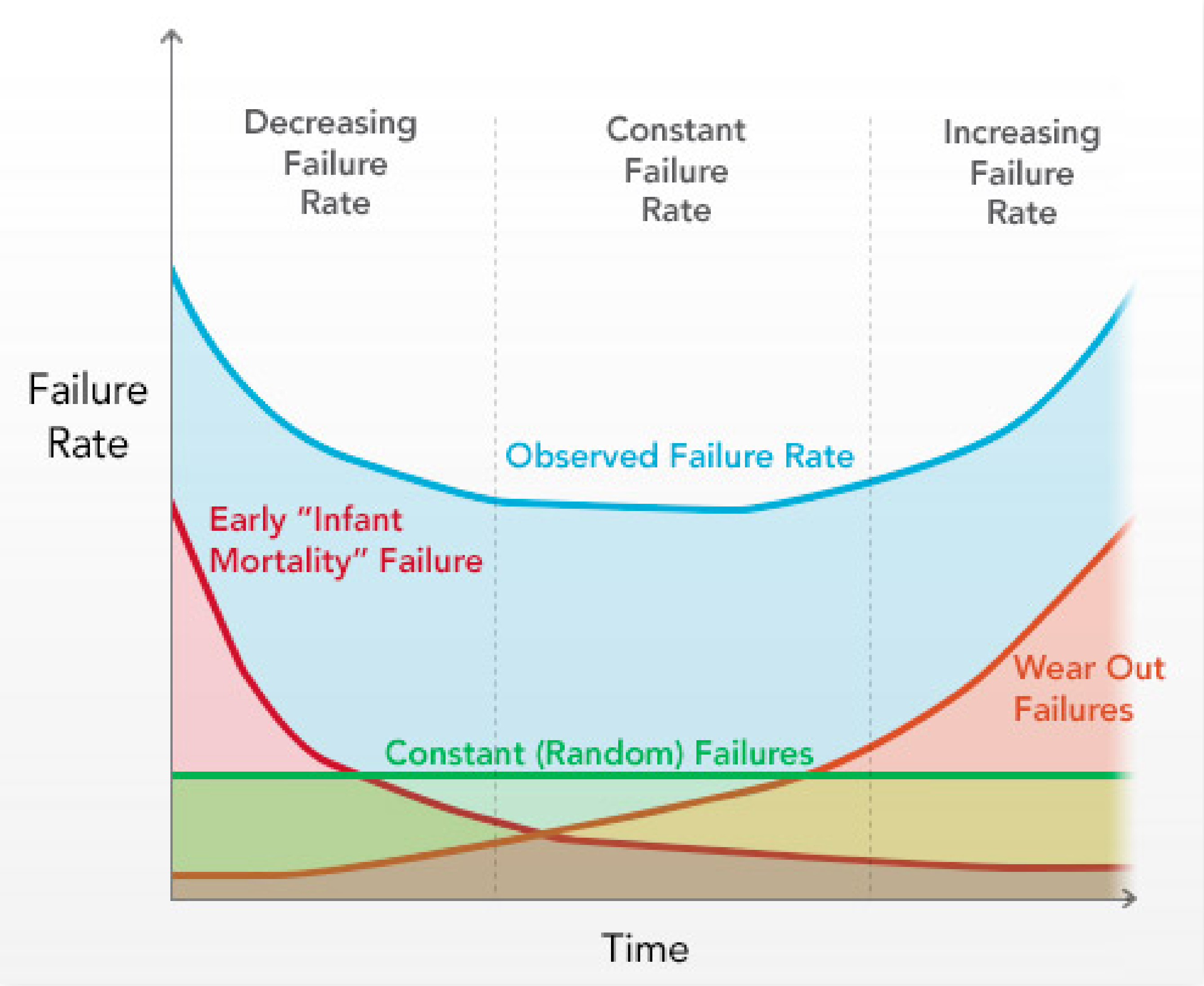
- Keep at least **three** copies of your data.
- Store **two** backup copies on different devices or storage media.
- Keep at least **one** backup copy offsite.

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


Decreasing Failure Rate
Constant Failure Rate
Increasing Failure Rate
Failure Rate
Observed Failure Rate
Early "Infant Mortality" Failure
Wear Out Failures
Constant (Random) Failures
Time

The Story of ToyStory2

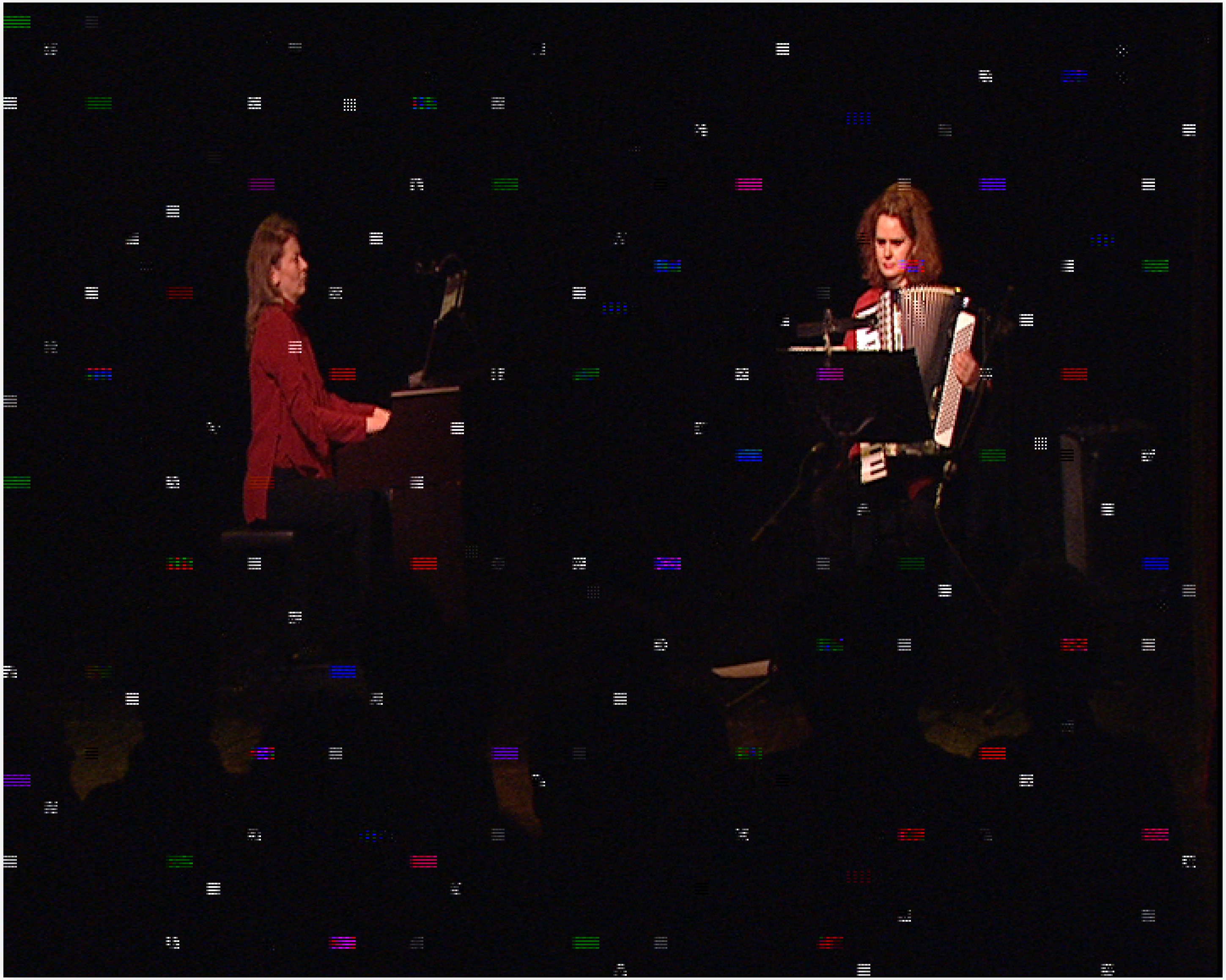
https://www.youtube.com/watch?v=8dhp_20j0Ys

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

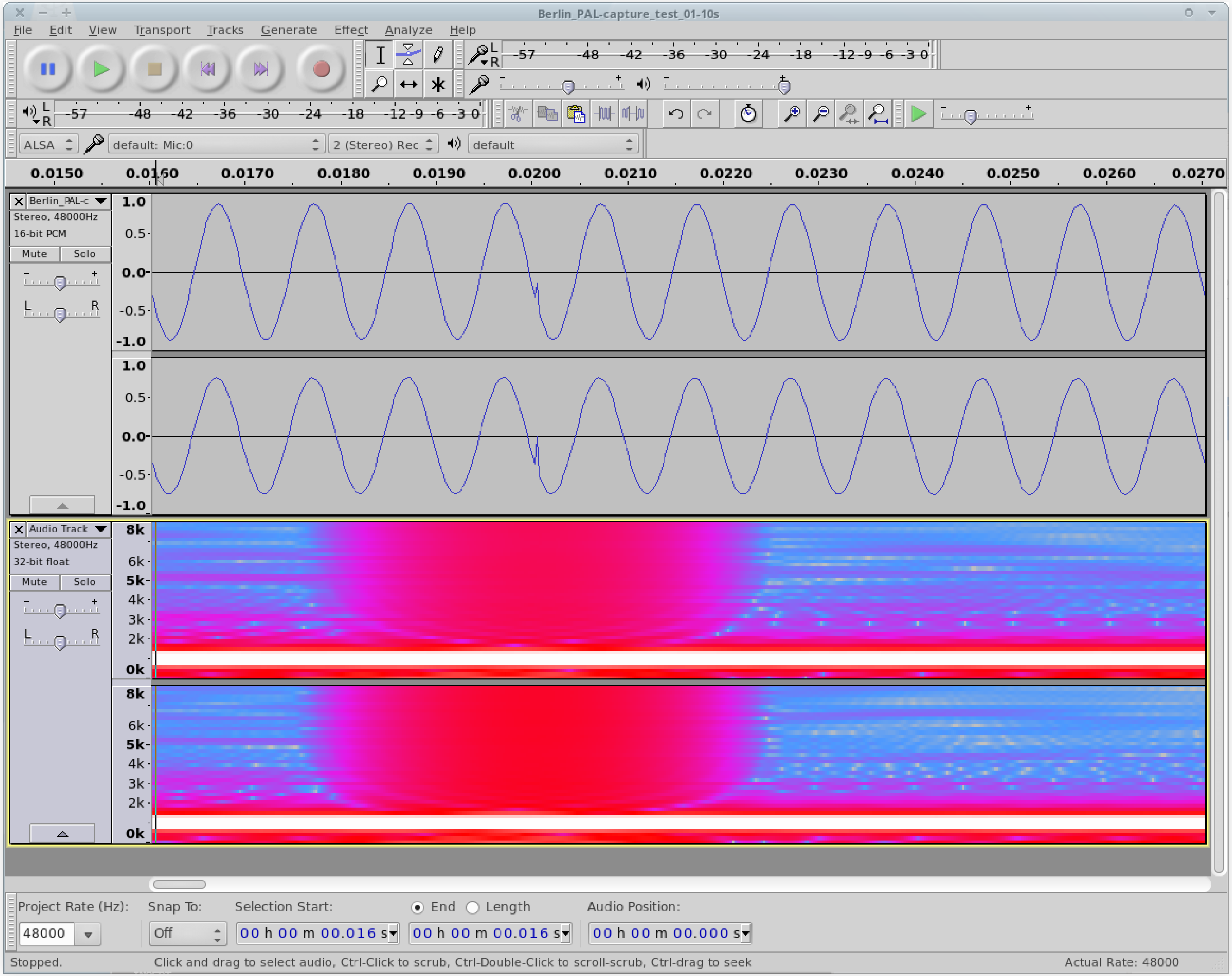


Speaker notes

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


Berlin_PAL-capture_test_01-10s
File Edit View Transport Tracks Generate Effect Analyze Help
ALSA
default: Mic:0
2 (Stereo) Rec
default
Berlin_PAL-c
Stereo, 48000Hz
16-bit PCM
Mute Solo
Audio Track
Stereo, 48000Hz
32-bit float
Project Rate (Hz):
48000
Snap To:
Off
Selection Start:
00 h 00 m 00.016 s
End Length
00 h 00 m 00.016 s
Audio Position:
00 h 00 m 00.000 s
Stopped.
Click and drag to select audio, Ctrl-Click to scrub, Ctrl-Double-Click to scroll-scrub, Ctrl-drag to seek
Actual Rate: 48000

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 = 0b1**1**111111
- 0xBF = 0b1**0**111111

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*

Data scrubbing, Fixity checking

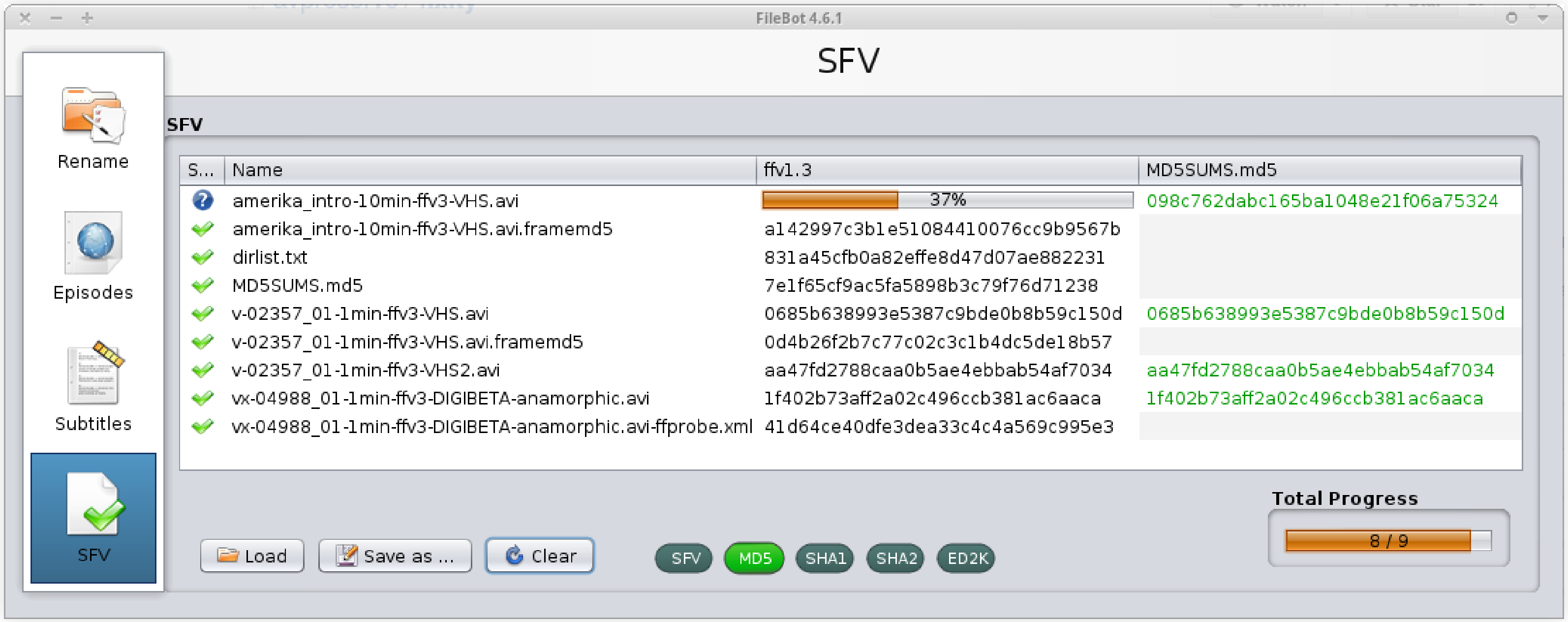
How do you know your data is intact?

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.

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!

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.

Comments?

Questions?

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)