On the Care and Feeding of Storage Devices

Benjamin Brewster

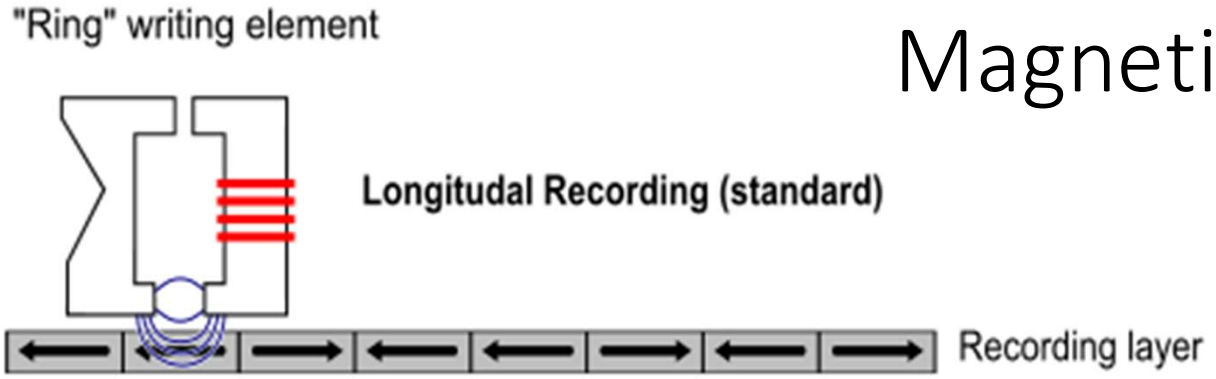
Why You Need to Care

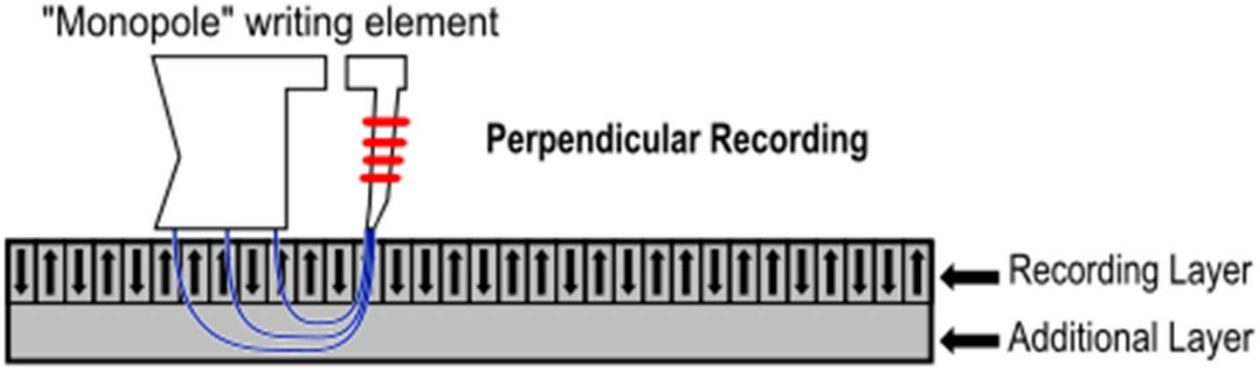
- Because someday you'll have to:
  - Compare them and make a wise purchasing decision - pick the right tool for the job!
  - Convince your boss you need an SSD in your work computer
  - Format a hard drive before you install an OS
  - Choose a file system format for a flash drive usable on Windows, MacOS, and Linux
  - Defragment a hard drive
  - Teach people not to abuse hard drives (they can break by being dropped only as little as an inch!)

Hard Disk Drives

- A Hard Disk Drive (HDD) holds data magnetically on a spinning set of platters, read and written to by ultra-precise heads on the end of a movable arm
- Modern drives position the read/write heads only a few nanometers above the platters, meaning that any foreign objects such as smoke or fingerprints will cause the head to collide with the platter and/or object, causing a “head crash”, destroying the platter and head
- HDDs are extremely sensitive to physical shock, such as drops
- The arm moves via an actuator known as a “voice coil”, which is the same technology used to move a speaker up and down based on varying voltage

Magnetic Data Recording



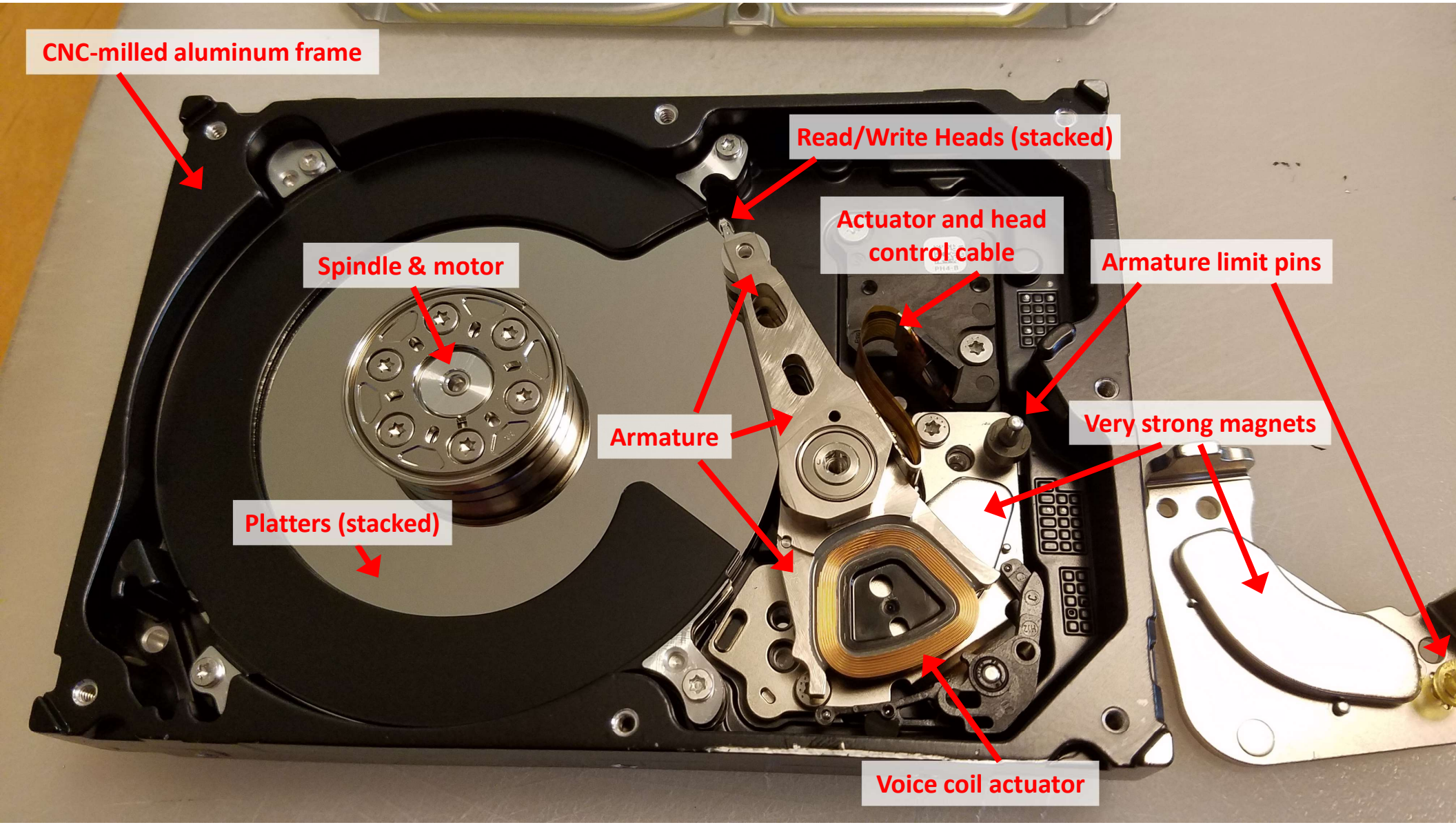


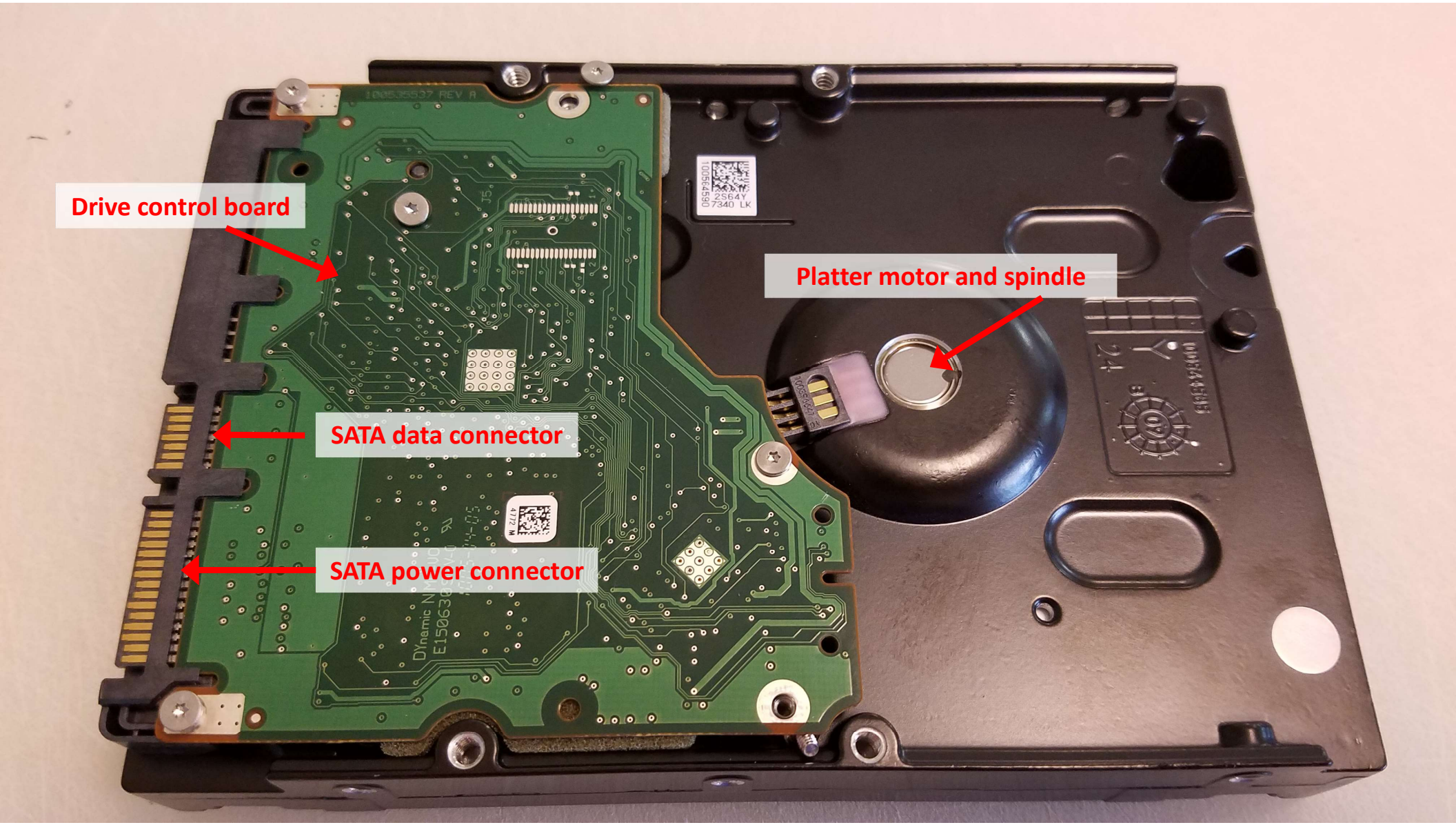
Luca Cassioli, Public Domain

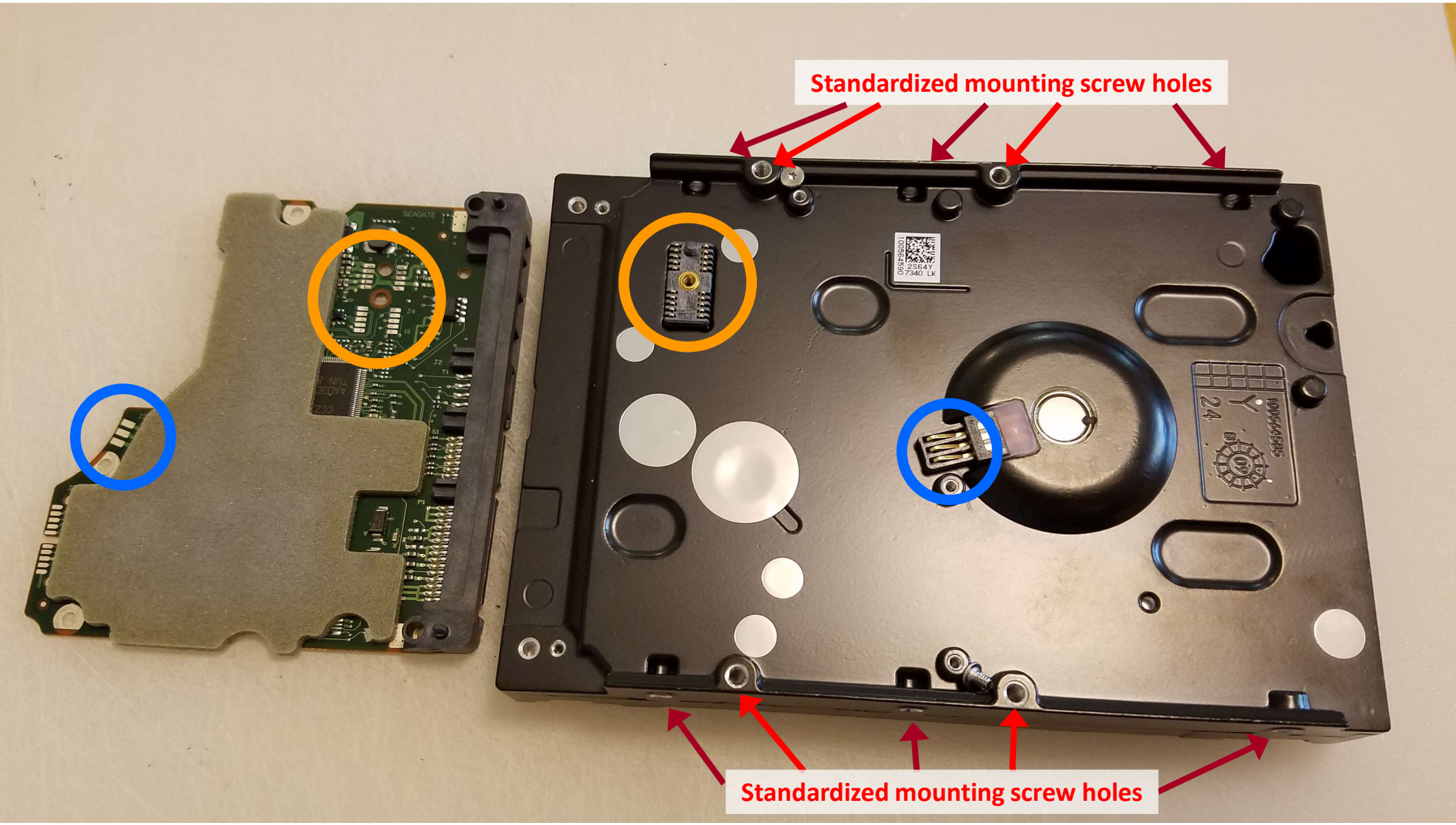
- Estimated data density limits:
- Longitudinal:
  - 100-200 Gb/sq. in
- Perpendicular:
  - 1000 Gb/sq. in

HDD Guts

- Our sample drive is:
  - Seagate OEM made for Apple
  - 1 TB (1000 GB)
  - 7200 RPM


CNC-milled aluminum frame
Read/Write Heads (stacked)
Actuator and head control cable
Armature limit pins
Spindle & motor
Armature
Very strong magnets
Platters (stacked)
Voice coil actuator


Drive control board
Platter motor and spindle
SATA data connector
SATA power connector


Standardized mounting screw holes
Standardized mounting screw holes

Solid-State Drives

- A Solid-state drive (SSD) stores data on flash memory
- Since any data cell can be accessed in fixed time, the seek speed is constant; reading any file, from any previous internal state, takes the same amount of time
- Read and write times are (far) faster than HDDs
- No moving parts, much lighter, smaller, not as shock sensitive, silent, can operate in vacuums, can be moved while being in use, uses generally less power, not susceptible to magnetic fields…
- … but much more expensive

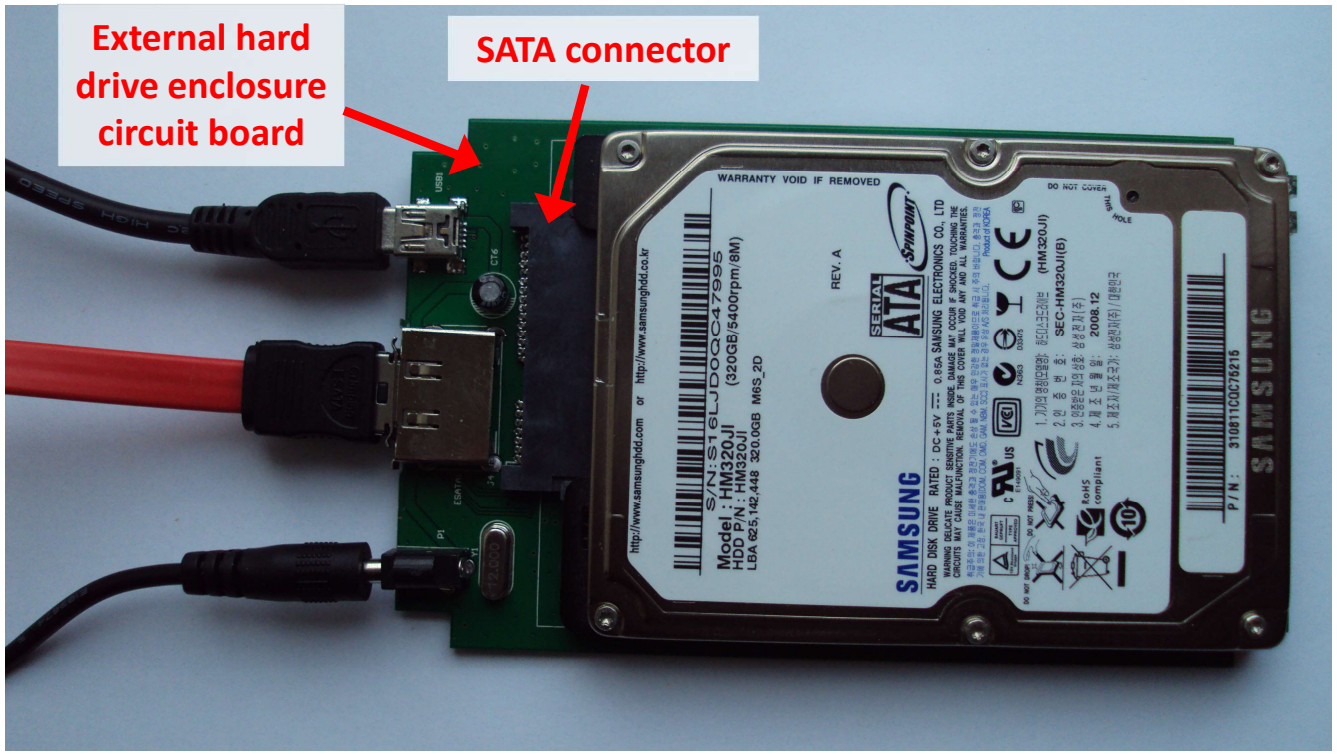
Form Factors

- The names of the form factors come from the size of the platters
- SSDs look the same, but don't have the obvious platter spindle

- "3.5 inch"
  - 4" x 1" x 5.75"
- aka Desktop

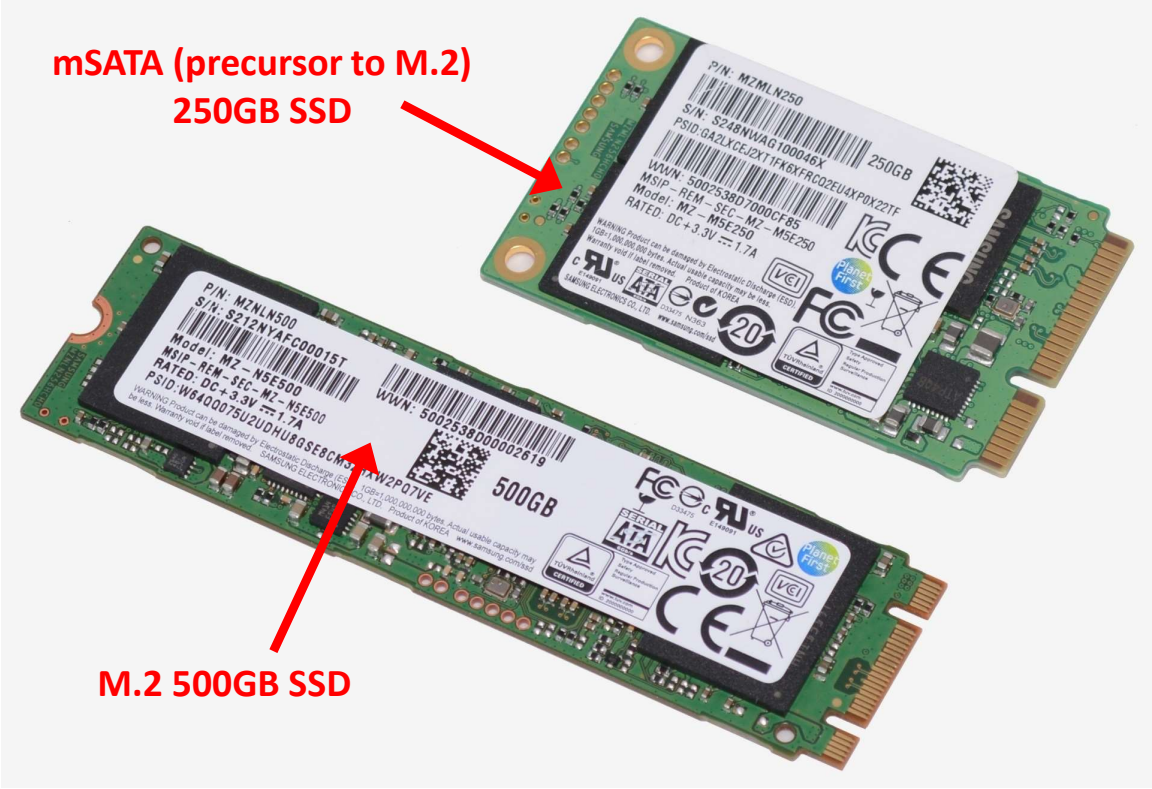
Form Factors

- “2.5 inch” :: 2.75” x 0.275-0.75” x 3.945” :: aka Laptop



Form Factors

- M.2 is a case-less, PCB board only-standard that supports widths from 12 to 30mm, and lengths from 30 to 110mm
- Hardware is built to accept one or more of these form factors

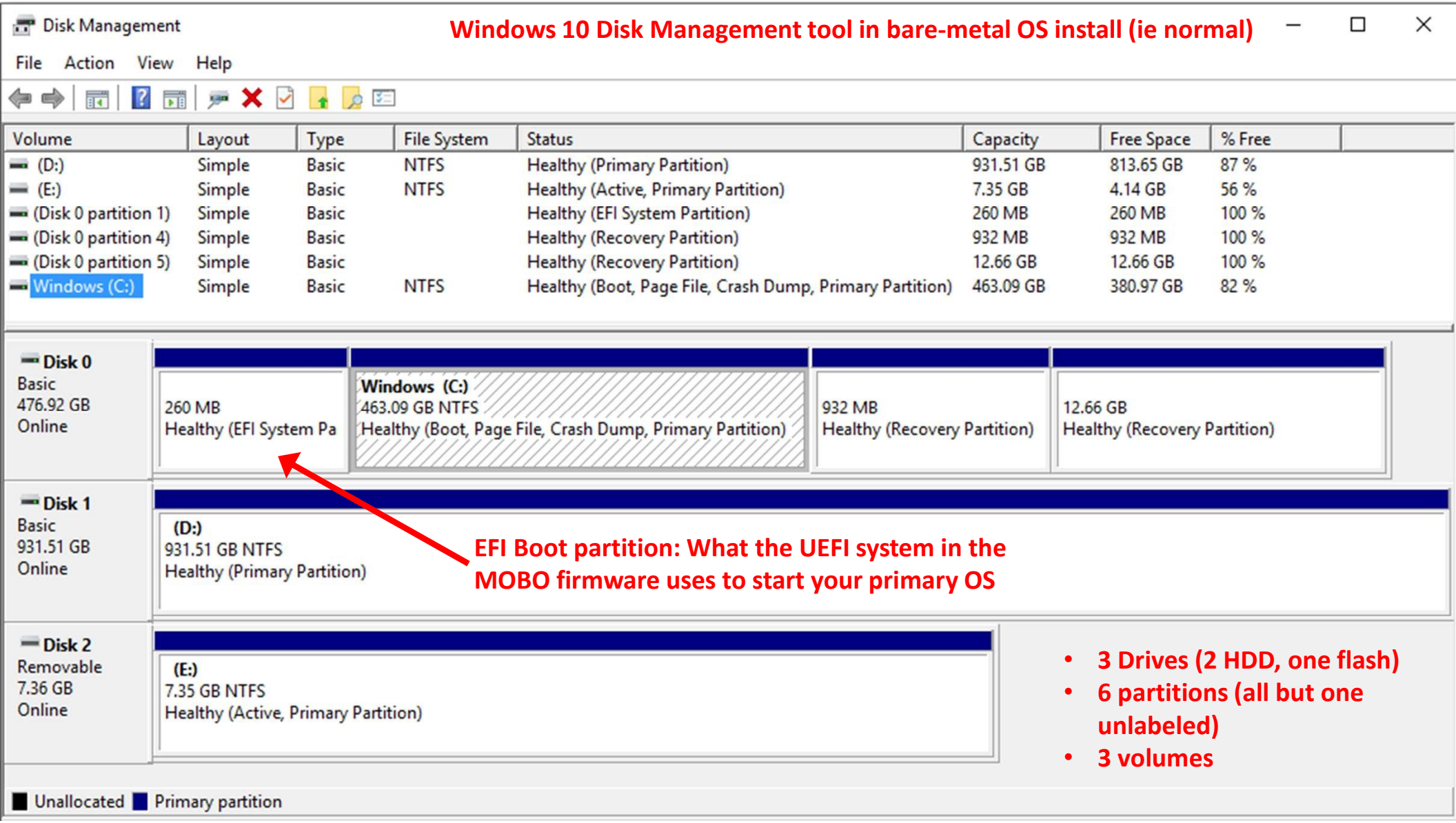


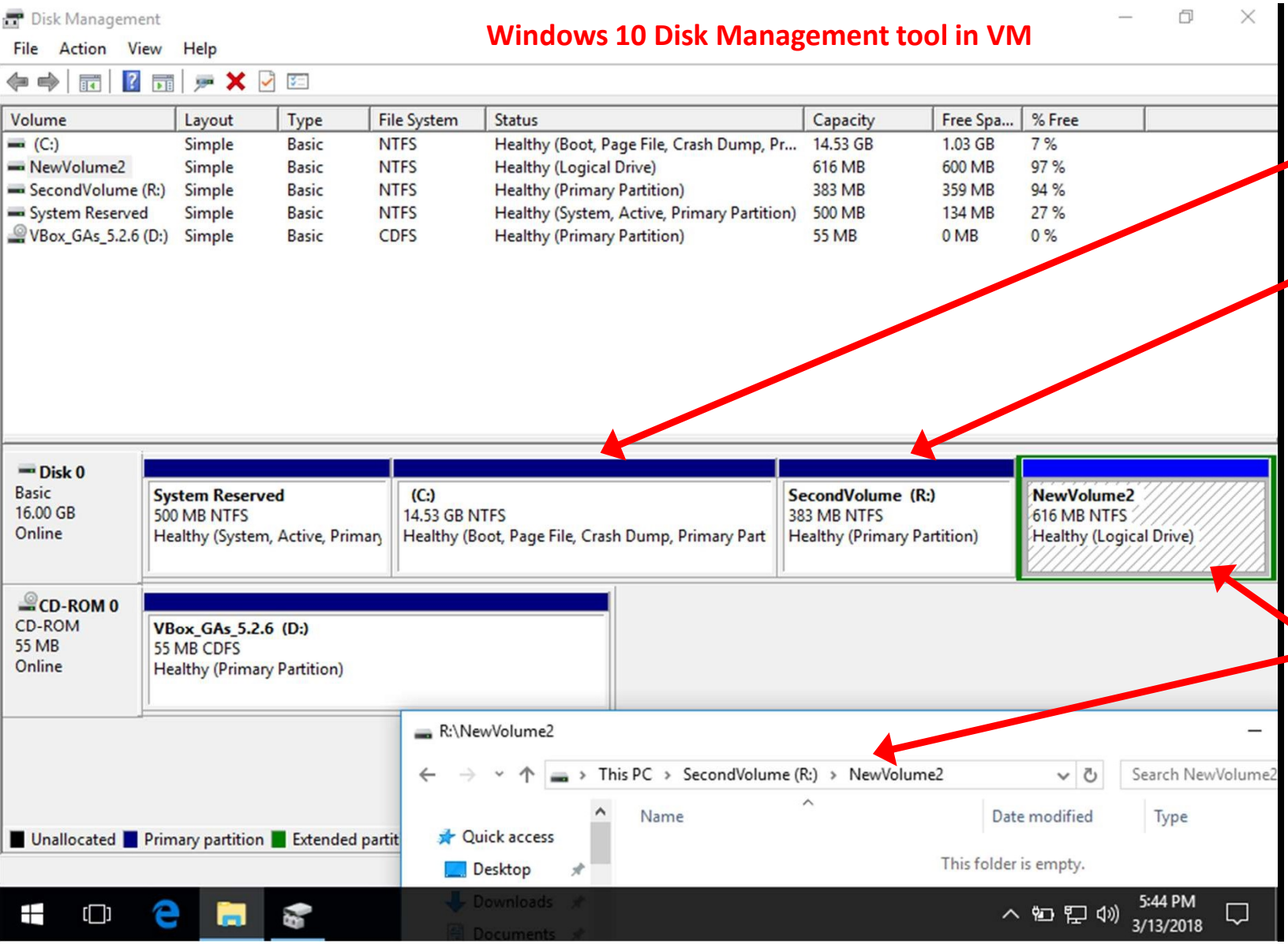
Failures

- HDDs typically fail because of mechanical degradation of the magnetic substrate (possibly from a head crash), spindle motor failure, power component failure, or physical shock
- SSDs typically fail because of power component failure, or too many erase operations on too many blocks over the lifetime of the drive
  - Algorithms (TRIM, wear leveling) spread out the operations, periodically moving data to less-used areas of the drive, to prevent one area from becoming over-used too soon
- Expected life-times of all types of constant-use drives are difficult to estimate; around 2-9% of all drives fail per year during the first three years
- Most manufacturers typically set consumer-grade drive warranties at (only) 3-5 years!

Partitions, Tables, and Volumes

- Hard drives are logically organized as follows:
- Partition: A designated, specific area of a drive, managed as a specific physical entity by the OS
- File System: A system of organizing the representation (naming), addressing, metadata, and storage of files and folders within a partition, including integrity features, permissions/security, and size limits (can be different for each partition)
- Volume: A file storage location inside a file system, consisting of a root directory in which all files and directories are hierarchically found
  - Multiple volumes can be created inside a partition
- Mount Point: the particular folder inside the booted operating system where a volume's files can be found
  - Volumes can be mounted inside an existing volume, causing the new volume to have a non-root filepath from the perspective of the *other* volumes
- How about some examples?


Disk Management
Windows 10 Disk Management tool in bare-metal OS install (ie normal)
File Action View Help
Volume Layout Type File System Status Capacity Free Space % Free
(D:) Simple Basic NTFS Healthy (Primary Partition) 931.51 GB 813.65 GB 87 %
(E:) Simple Basic NTFS Healthy (Active, Primary Partition) 7.35 GB 4.14 GB 56 %
(Disk 0 partition 1) Simple Basic Healthy (EFI System Partition) 260 MB 260 MB 100 %
(Disk 0 partition 4) Simple Basic Healthy (Recovery Partition) 932 MB 932 MB 100 %
(Disk 0 partition 5) Simple Basic Healthy (Recovery Partition) 12.66 GB 12.66 GB 100 %
Windows (C:) Simple Basic NTFS Healthy (Boot, Page File, Crash Dump, Primary Partition) 463.09 GB 380.97 GB 82 %
Disk 0
Basic
476.92 GB
Online
260 MB
Healthy (EFI System Pa
Windows (C:)
463.09 GB NTFS
Healthy (Boot, Page File, Crash Dump, Primary Partition)
932 MB
Healthy (Recovery Partition)
12.66 GB
Healthy (Recovery Partition)
Disk 1
Basic
931.51 GB
Online
(D:)
931.51 GB NTFS
Healthy (Primary Partition)
EFI Boot partition: What the UEFI system in the MOBO firmware uses to start your primary OS
Disk 2
Removable
7.36 GB
Online
(E:)
7.35 GB NTFS
Healthy (Active, Primary Partition)
• 3 Drives (2 HDD, one flash)
• 6 partitions (all but one unlabeled)
• 3 volumes
Unallocated
Primary partition


Disk Management
File Action View Help
Windows 10 Disk Management tool in VM
Volume Layout Type File System Status Capacity Free Spa... % Free
(C:) Simple Basic NTFS Healthy (Boot, Page File, Crash Dump, Pr... 14.53 GB 1.03 GB 7 %
NewVolume2 Simple Basic NTFS Healthy (Logical Drive) 616 MB 600 MB 97 %
SecondVolume (R:) Simple Basic NTFS Healthy (Primary Partition) 383 MB 359 MB 94 %
System Reserved Simple Basic NTFS Healthy (System, Active, Primary Partition) 500 MB 134 MB 27 %
VBox_GAs_5.2.6 (D:) Simple Basic CDFS Healthy (Primary Partition) 55 MB 0 MB 0 %
Disk 0
Basic
16.00 GB
Online
System Reserved
500 MB NTFS
Healthy (System, Active, Primary
(C:)
14.53 GB NTFS
Healthy (Boot, Page File, Crash Dump, Primary Part
SecondVolume (R:)
383 MB NTFS
Healthy (Primary Partition)
NewVolume2
616 MB NTFS
Healthy (Logical Drive)
CD-ROM 0
CD-ROM
55 MB
Online
VBox_GAs_5.2.6 (D:)
55 MB CDFS
Healthy (Primary Partition)
Unallocated
Primary partition
Extended partit
R:\NewVolume2
This PC › SecondVolume (R:) › NewVolume2
Search NewVolume2
Quick access
Desktop
Downloads
Documents
Name
Date modified
Type
This folder is empty.
5:44 PM
3/13/2018
Unlabeled partition with volume “C:”
Volume “R:” on partition “SecondVolume”
Volume named “NewVolume2” inside volume “R:” on partition SecondVolume

Identifying Linux Mount Points and Volumes

```
$ cat /proc/mounts | grep "stak/users"
128.193.40.232:/stak_users/data /nfs/stak/users nfs
rw,nosuid,relatime,vers=3,rsize=262144,wsize=262144,namlen=255,hard,proto=tcp,port=2054,ti
meo=600,retrans=2,sec=sys,mountaddr=128.193.40.232,mountvers=3,mountport=5004,mountproto=t
cp,local_lock=none,addr=128.193.40.232 0 0
```

```
$ lsblk
NAME                  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                     8:0    0  100G  0 disk
└─sda1                  8:1    0  700M  0 part /boot
sdb                     8:16   0  5.6T  0 disk
└─sdb1                  8:17   0  5.6T  0 part
  ├─vg_el7-lv_root    253:0    0    2G  0 lvm  /
  ├─vg_el7-lv_swap    253:1    0 31.4G  0 lvm  [SWAP]
  ├─vg_el7-lv_usr     253:2    0   20G  0 lvm  /usr
  ├─vg_el7-lv_scratch 253:3    0  5.6T  0 lvm  /scratch
  ├─vg_el7-lv_private 253:4    0    1G  0 lvm  /private
  ├─vg_el7-lv_var     253:5    0    8G  0 lvm  /var
  └─vg_el7-lv_tmp     253:6    0    8G  0 lvm  /tmp
```

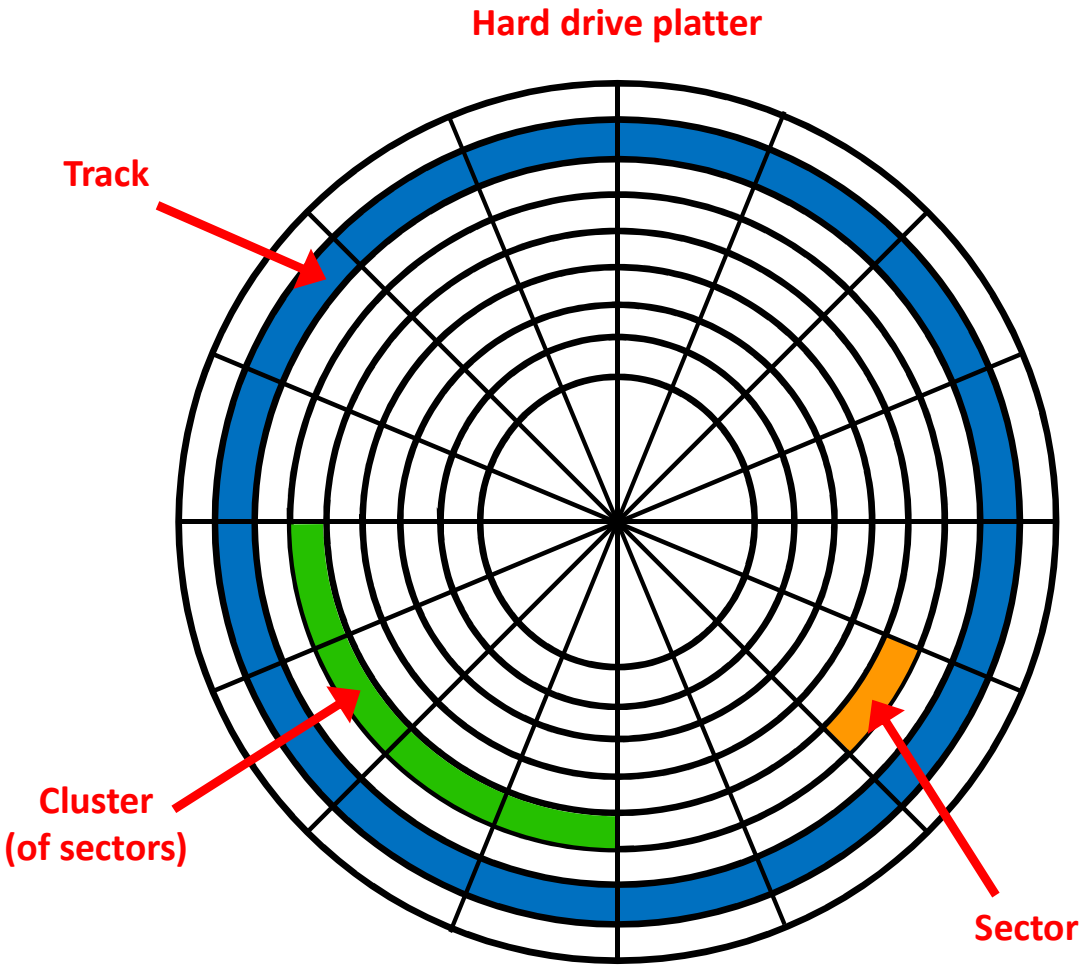
Types:

- disk = drive
- part = partition
- lvm = Logical Volume Manager, a virtual partition aka volume

These volumes all have mountpoints in /

Sectors and Clusters

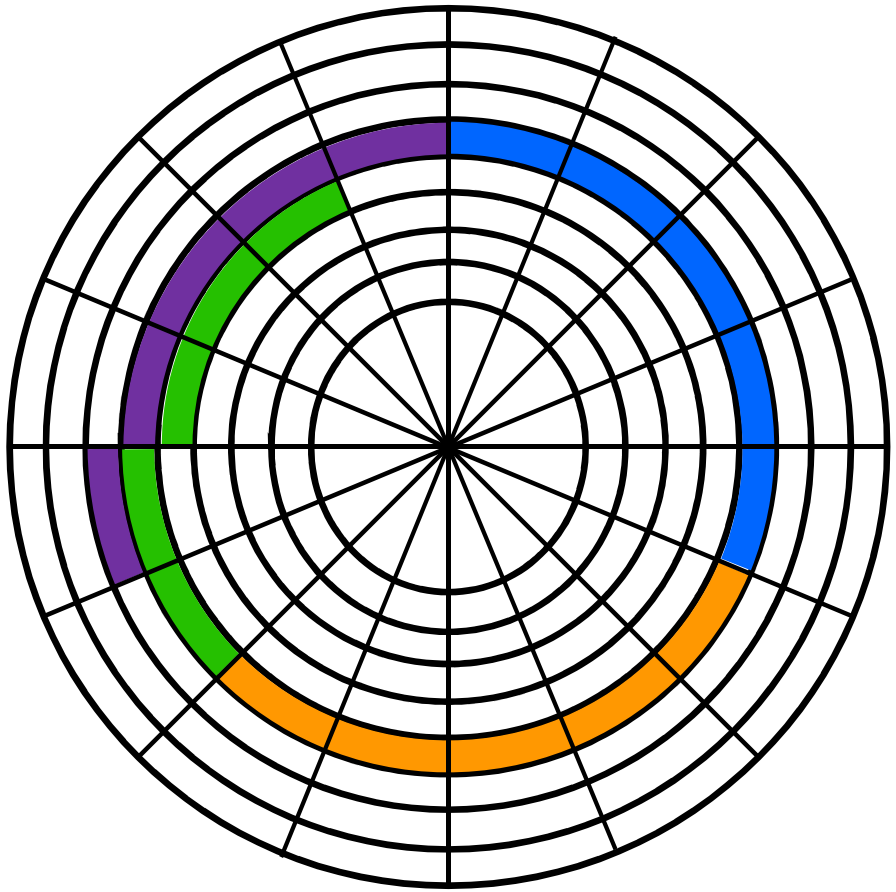
- **Sectors** are the minimum storage unit of a drive, and have been various sizes; today usually 4,096 bytes (the *Advanced Format*)
- A **cluster** (aka allocation unit) is the smallest amount of space that the file system will use to store a file
- The size of a cluster is specified when the drive is formatted
- A drive formatted with 4KB clusters (1 sector) will use an entire 4K cluster for a 10-byte file; the wasted space is called **slack space**
- A larger cluster size reduces fragmentation, but increases slack space



Sectors and Clusters

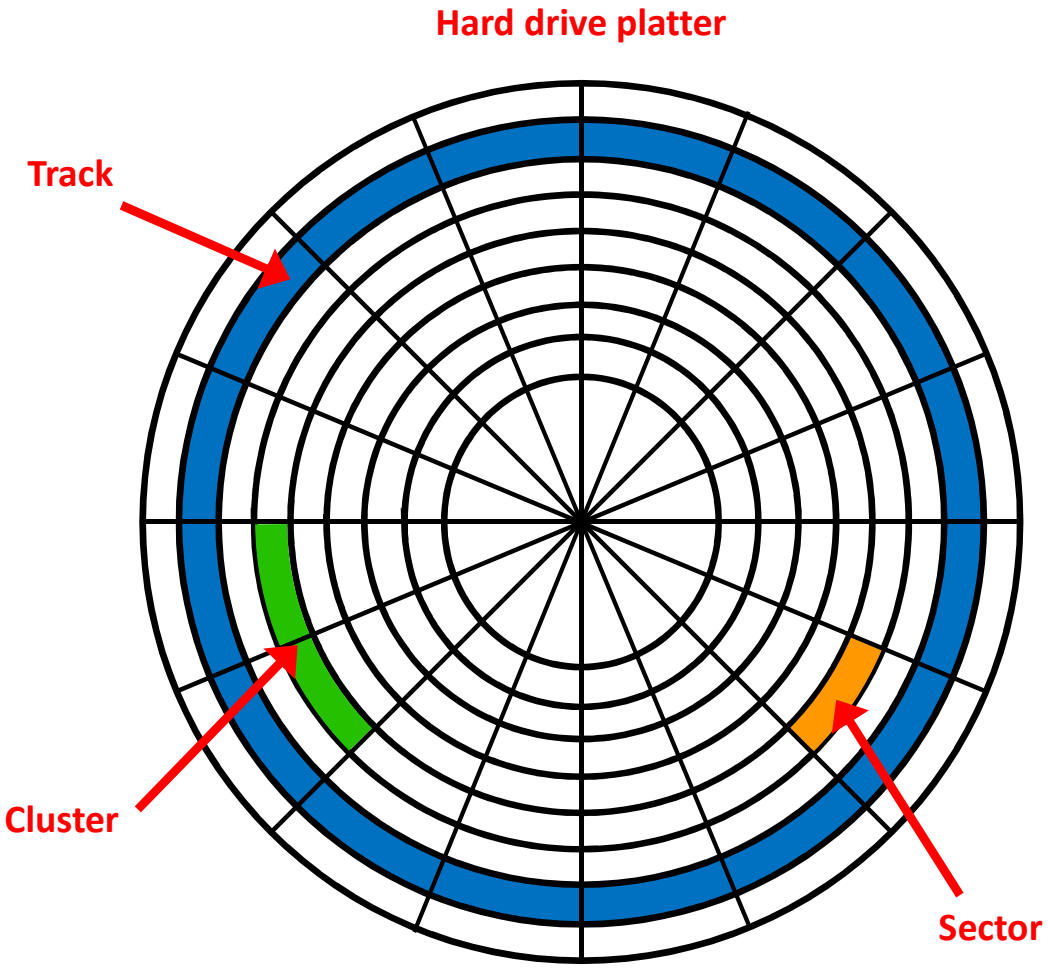
- Clusters do not have to be *physically* contiguous to still be *logically* contiguous
- A track boundary might be in the middle of a cluster

- Here:
  - 4,096 byte sectors
  - 20,480 byte clusters
  - 65,636 byte tracks



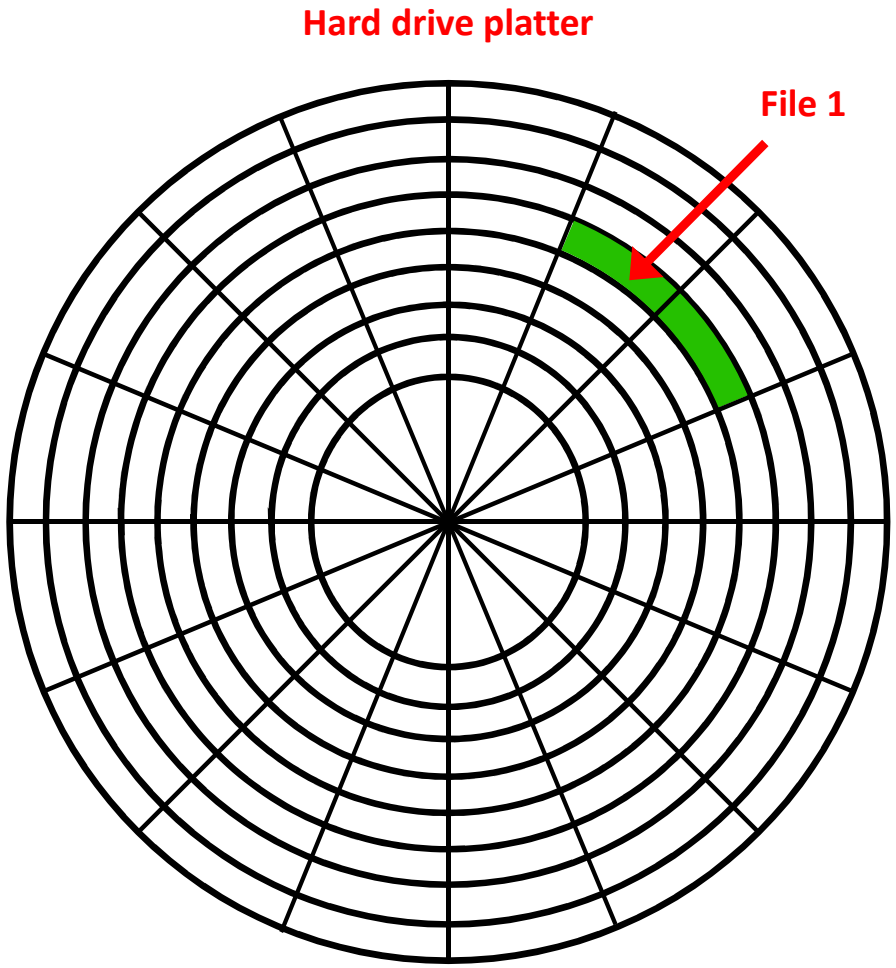
Fragmentation

- Files that are stored on a disk can become fragmented over time, causing increased latency to read and write operations, due to waiting for the platter to rotate to the read/write heads
- For this example:
  - 4,096 byte sectors
  - 8,192 byte clusters
  - 65,636 byte tracks



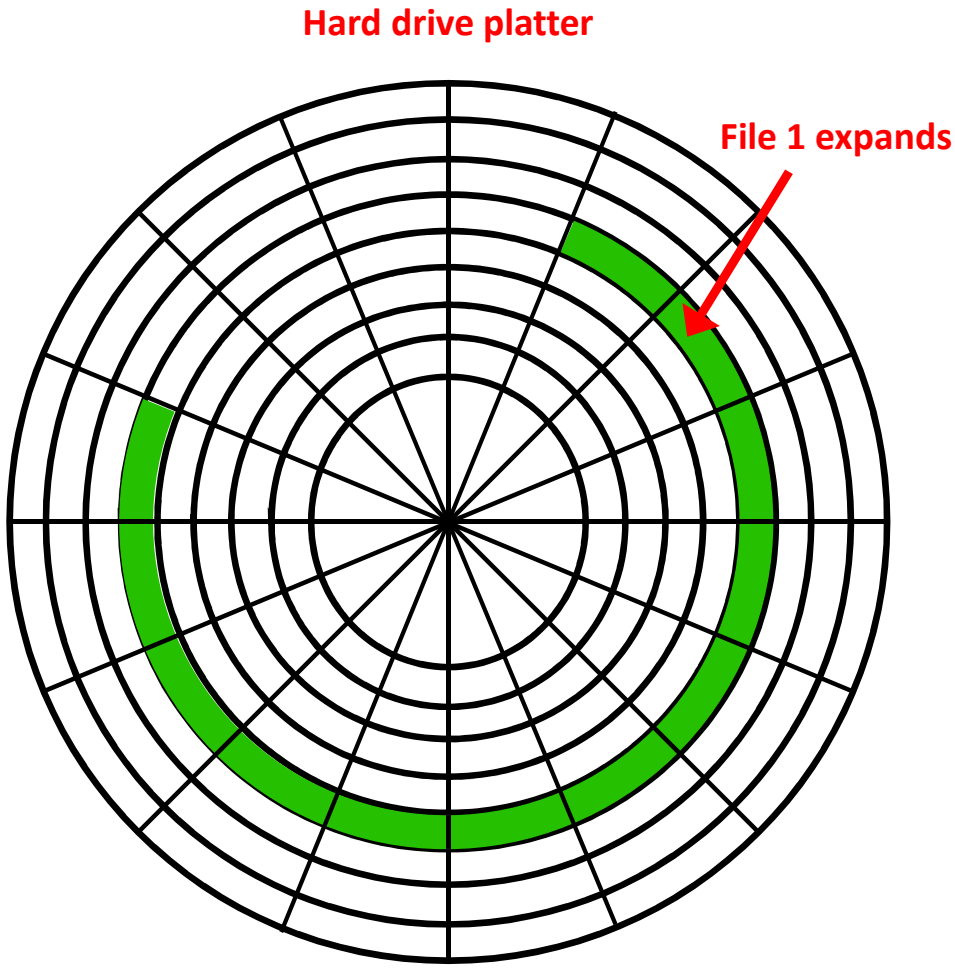
Fragmentation

- Files that are stored on a disk can become fragmented over time, causing increased latency to read and write operations, due to waiting for the platter to rotate to the read/write heads
- For this example:
  - 4,096 byte sectors
  - 8,192 byte clusters
  - 65,636 byte tracks



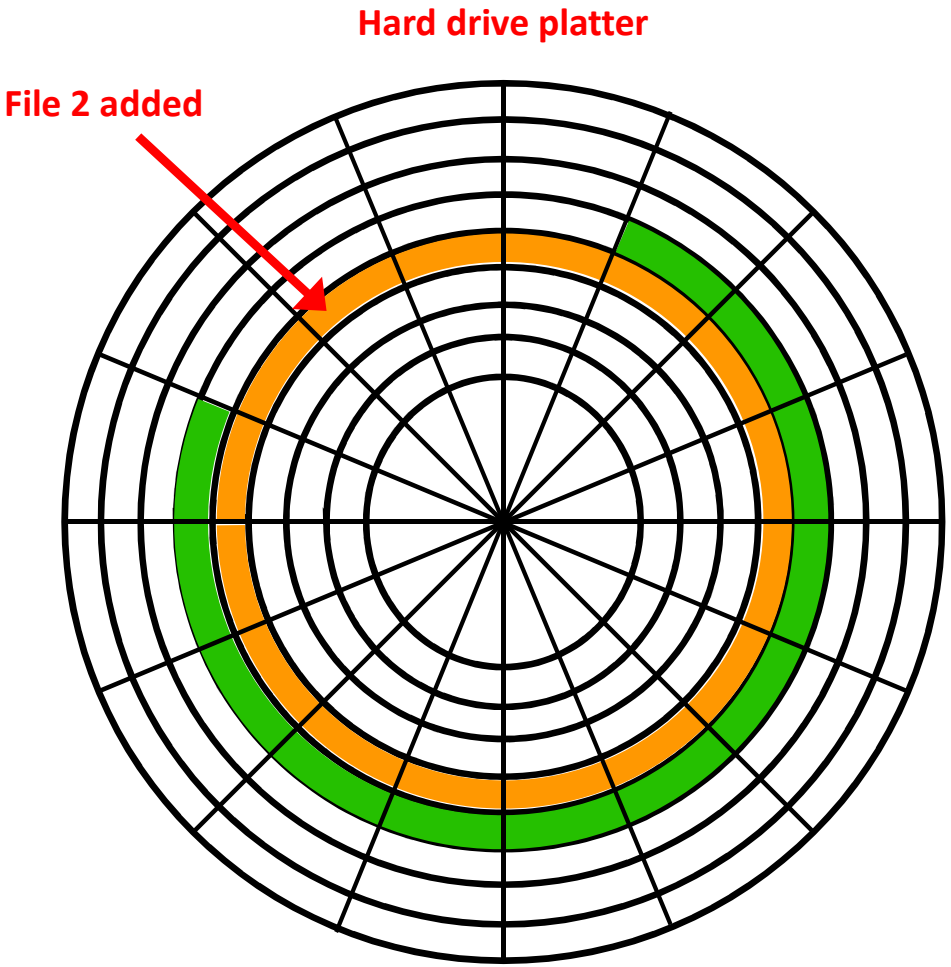
Fragmentation

- Files that are stored on a disk can become fragmented over time, causing increased latency to read and write operations, due to waiting for the platter to rotate to the read/write heads
- For this example:
  - 4,096 byte sectors
  - 8,192 byte clusters
  - 65,636 byte tracks



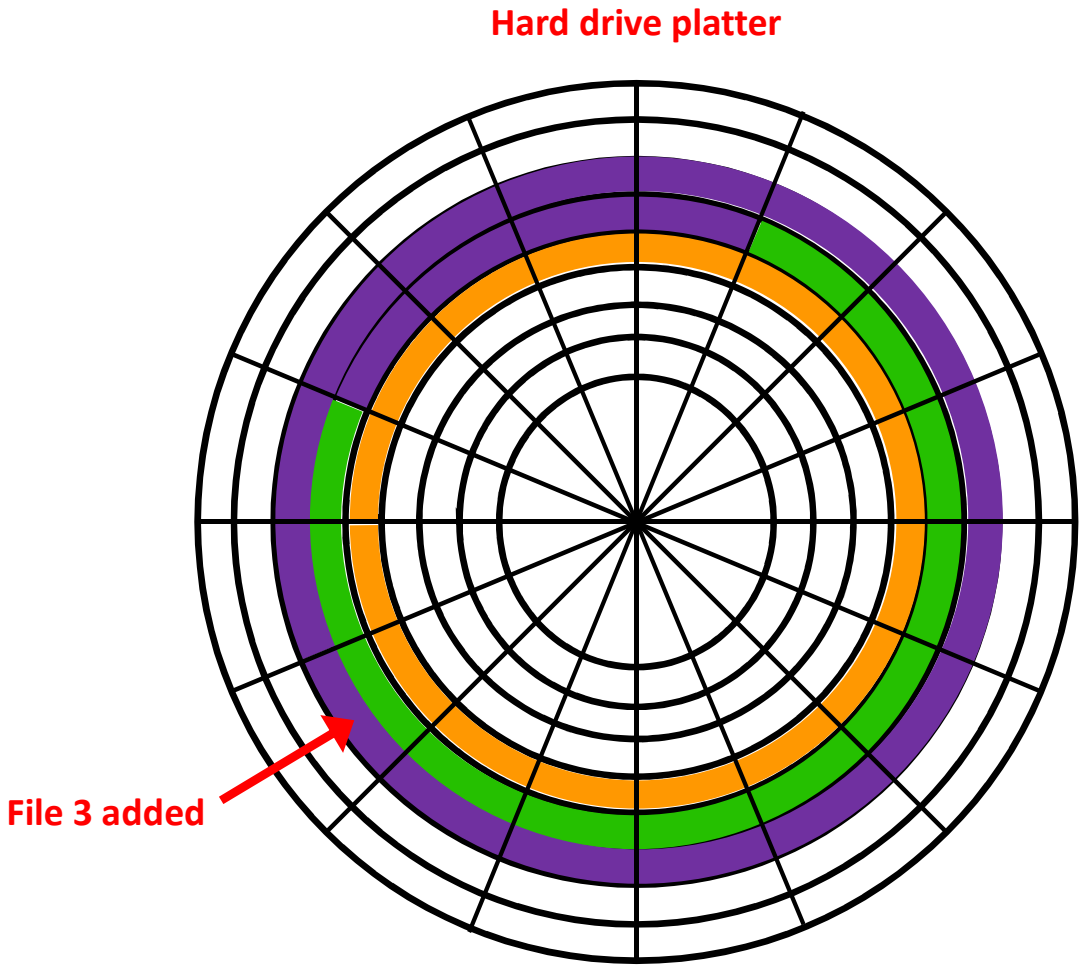
Fragmentation

- Files that are stored on a disk can become fragmented over time, causing increased latency to read and write operations, due to waiting for the platter to rotate to the read/write heads
- For this example:
  - 4,096 byte sectors
  - 8,192 byte clusters
  - 65,636 byte tracks



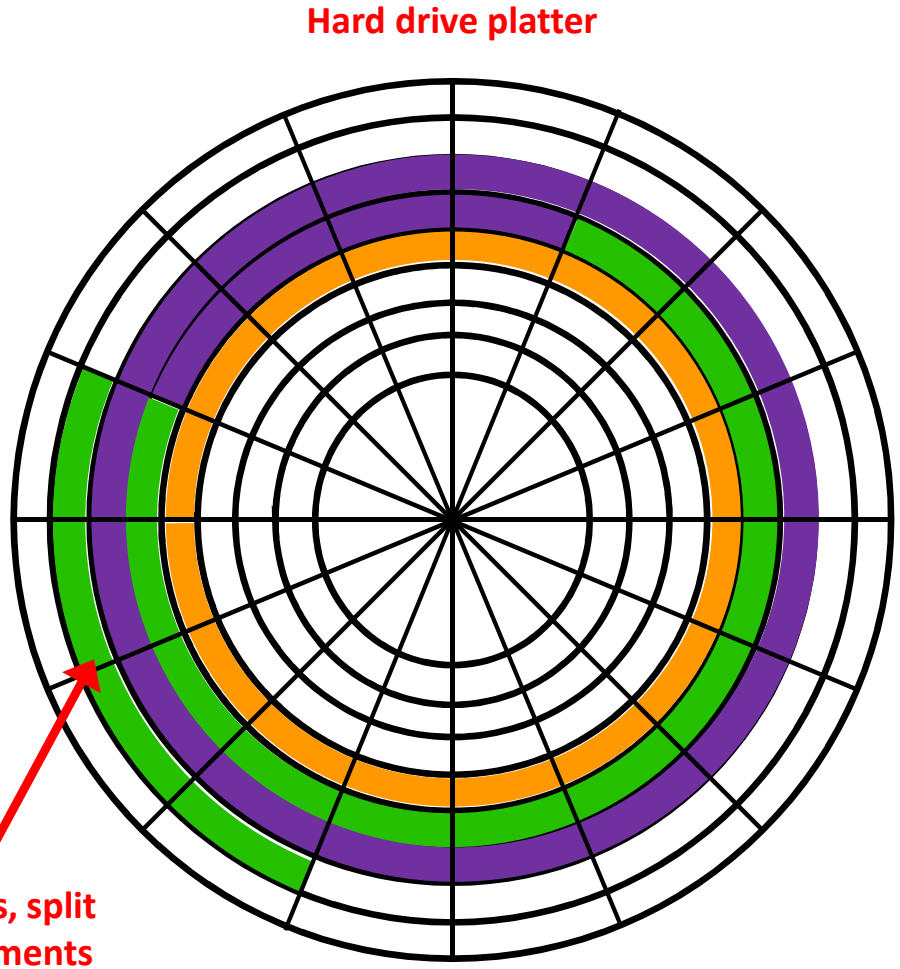
Fragmentation

- Files that are stored on a disk can become fragmented over time, causing increased latency to read and write operations, due to waiting for the platter to rotate to the read/write heads
- For this example:
  - 4,096 byte sectors
  - 8,192 byte clusters
  - 65,636 byte tracks



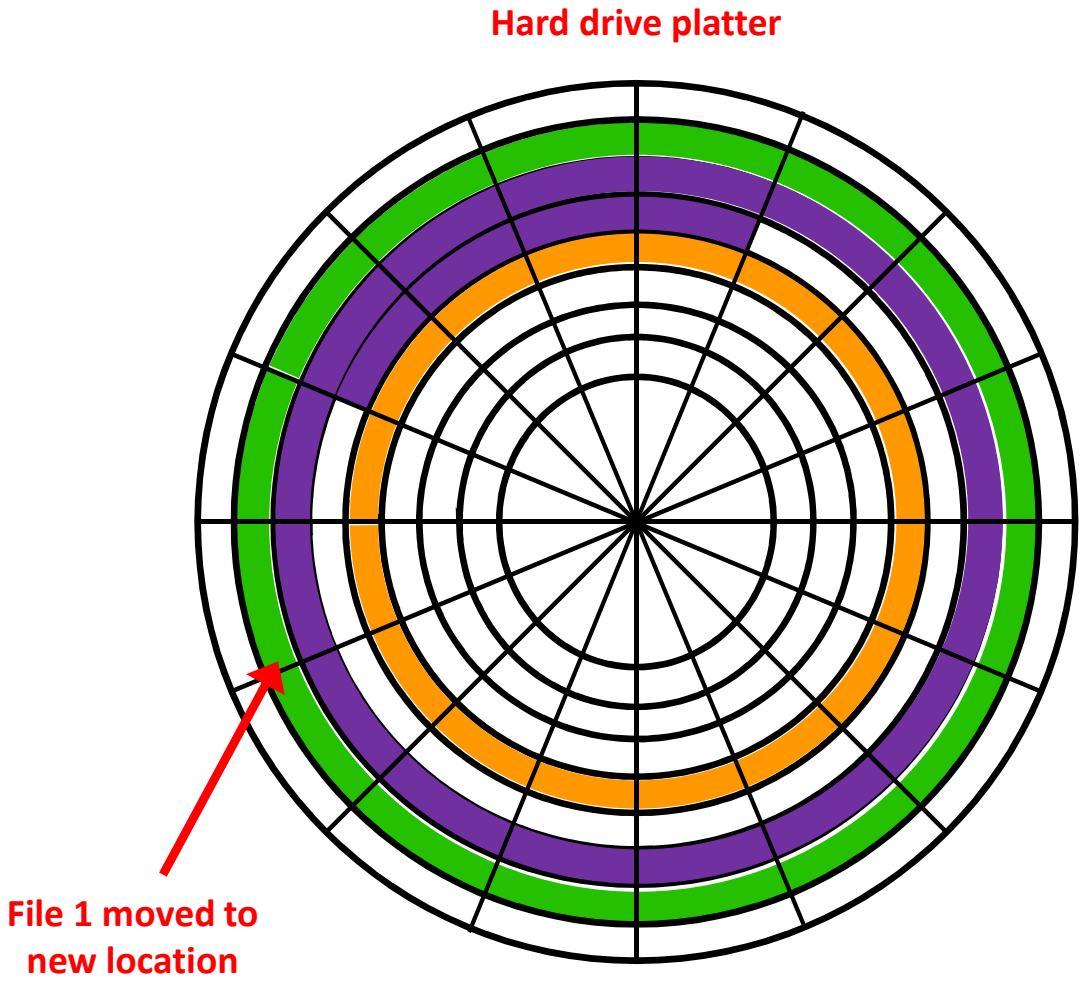
Fragmentation

- Files that are stored on a disk can become fragmented over time, causing increased latency to read and write operations, due to waiting for the platter to rotate to the read/write heads
- For this example:
  - 4,096 byte sectors
  - 8,192 byte clusters
  - 65,636 byte tracks



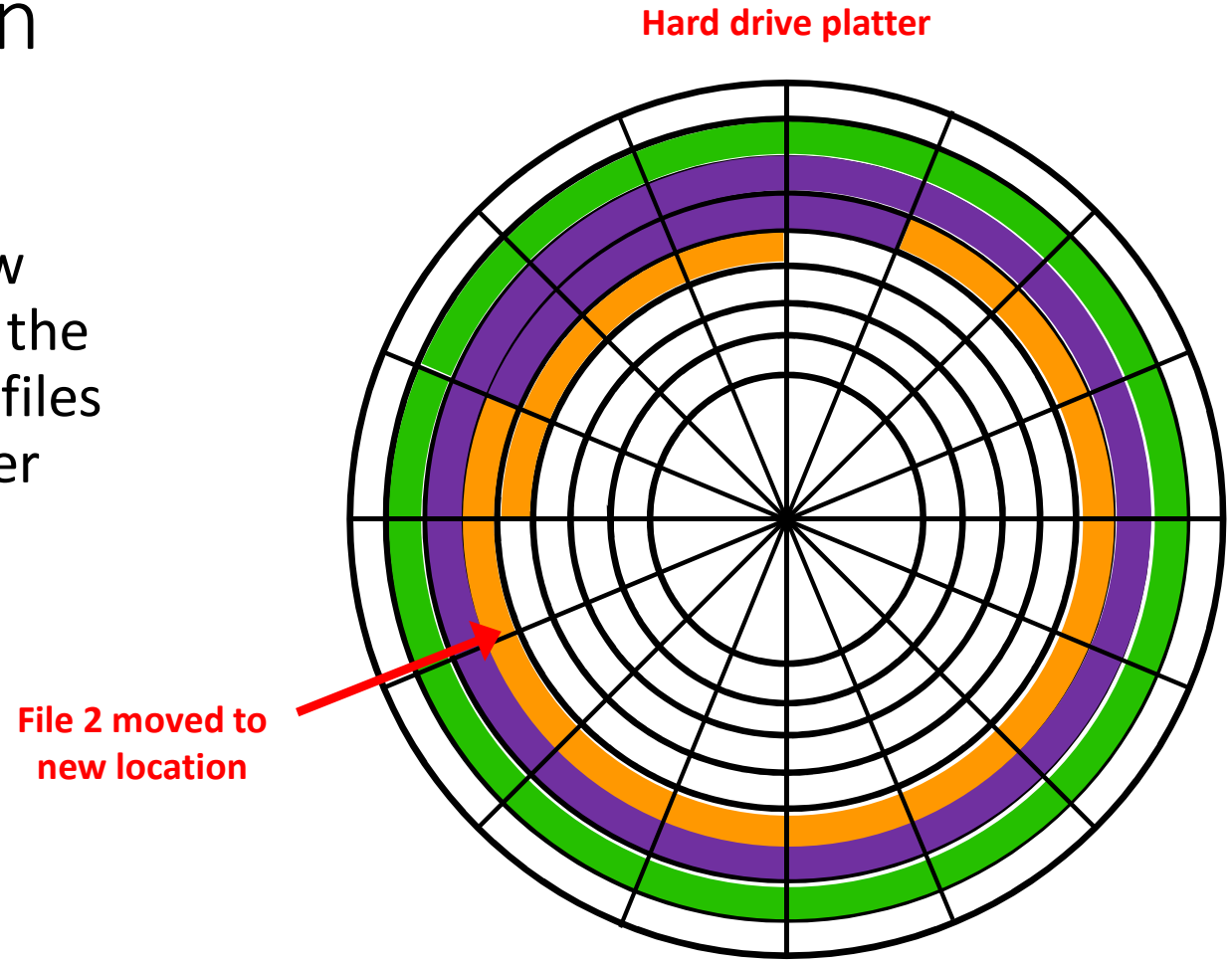
Defragmentation

- Defragmentation moves files into contiguous physical locations for faster read/write access



Defragmentation

- Inner holes will also be compacted to both slow down fragmentation in the future, and position all files closer together for faster read/write access



Defragmentation Final Thoughts

- SSDs do not need to be defragged: they fragment so slowly due to algorithms put in place to prevent this from happening, that it's not worth the write use on the drive
- When Windows "defrags" an SSD, it's merely doing a more aggressive TRIM garbage collection on the drive
- Starting with Windows 8, Windows automatically defrags HDDs on a schedule - right when you're trying to do something important

Disk Formatting

- Formatting a partition in Windows is done from File Explorer: Right-click on a lettered volume (which is on it's own partition), select Format; can choose:
  - Capacity (possibly leaving some parts of the drive unformatted and not part of the primary partition)
  - File System (NTFS, FAT(32) exFAT, etc.)
  - Allocation Unit size (aka block/cluster: 4096, 8192, 16 KB, 32 KB, 64 KB)
  - A disk label or name: MUSIC

- Formatting a partition in Linux allows the same choices:

```
$ mkfs -t ext3 -b 1024 -L MUSIC /dev/sda1
```

Blocks

- The term **block** can refer to many things depending on context:
  - A sector on a disk
  - A chunk of data in a transmission, either specified because of a manual setting, or perhaps related to the amount of data handled at once by an underlying driver, software buffer, or hardware buffer
  - The amount of data a “block device” processes at once, which may or may not be related to the sector or cluster size on the disk it gets that data from; could be the amount specified in the previous bullet point
  - An arbitrarily-sized unit used to track file size in UNIX, e.g. 512 bytes
- **Logical block addressing** (LBA) is the name of the scheme used to give every block/sector on a hard drive a single number (0, 1, 2, etc.) regardless of its physical position

File Systems in More Detail

- An OS cannot read a filesystem unless it has drivers to do so
- Filesystems are frequently designed by OS developers to meet the needs of the OS
- A Format is synonymous with File System

Drive Wiping

- Deleting a file merely marks the area on the drive as free, but the data is still there: many tools can extract that data
- Even formatting the drive may not be enough!
- Data can be securely deleted from a HDD only by writing data over the entire drive, but are you sure you've gotten the entire drive?
- Despite reports to the contrary, files that have been overwritten cannot be read from an HDD, even with Magnetic Force Microscopy
- SSDs occasionally have bugs in their delete/wipe commands that leave data intact
- The only sure way to wipe is to destroy the platters or memory chips: shredding, drills, hammers, disassembly and big magnets, etc.

Conclusion + HDD Speaker!

- Understanding the details behind file systems and disk drive management allows you to create storage areas that meet the specified requirements
- Examples and standard file location descriptions coming next week

- Laptop to amp to hard drive = speaker!