Building a security appliance based on FreeBSD

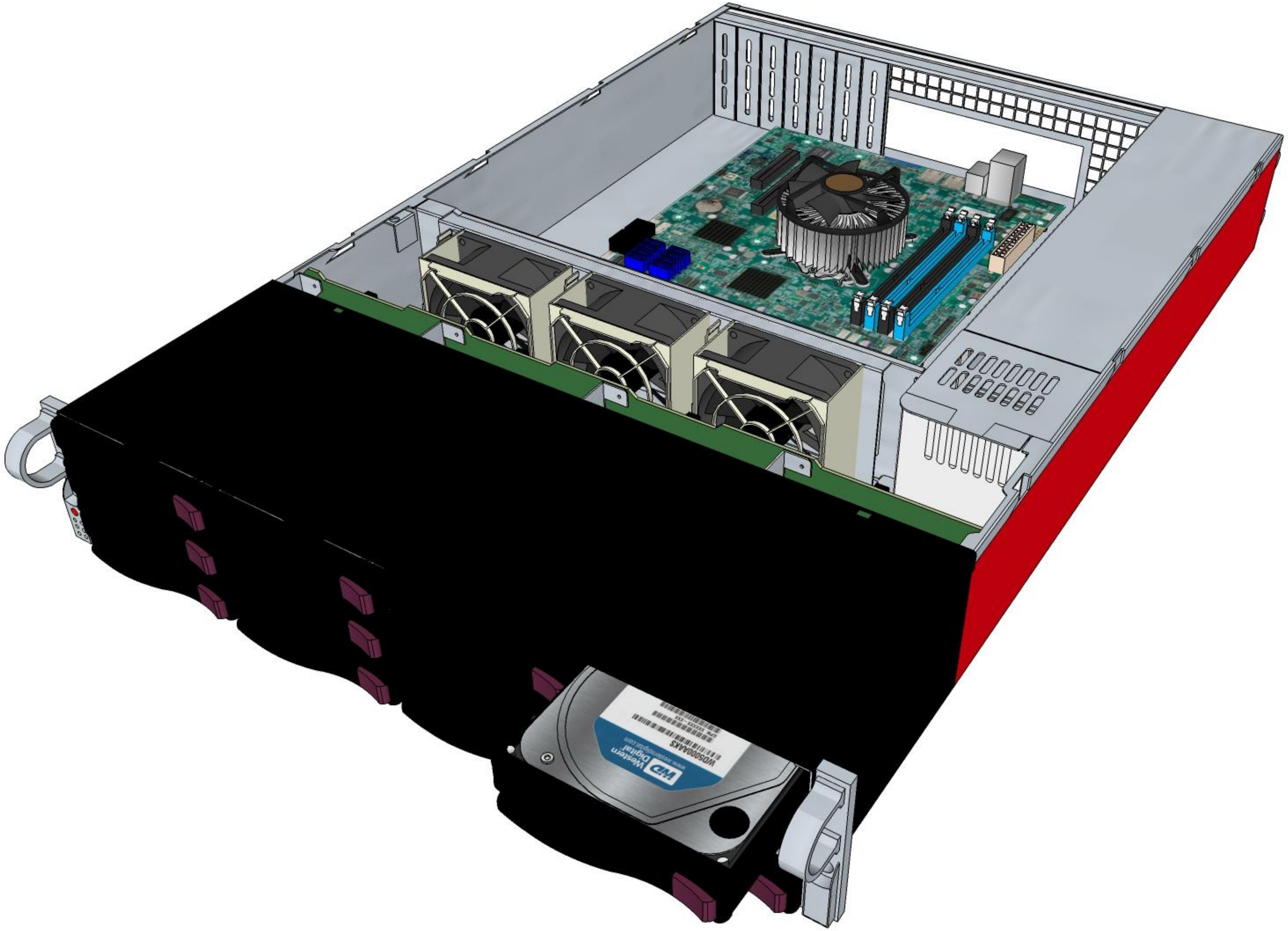
Mariusz Zaborski

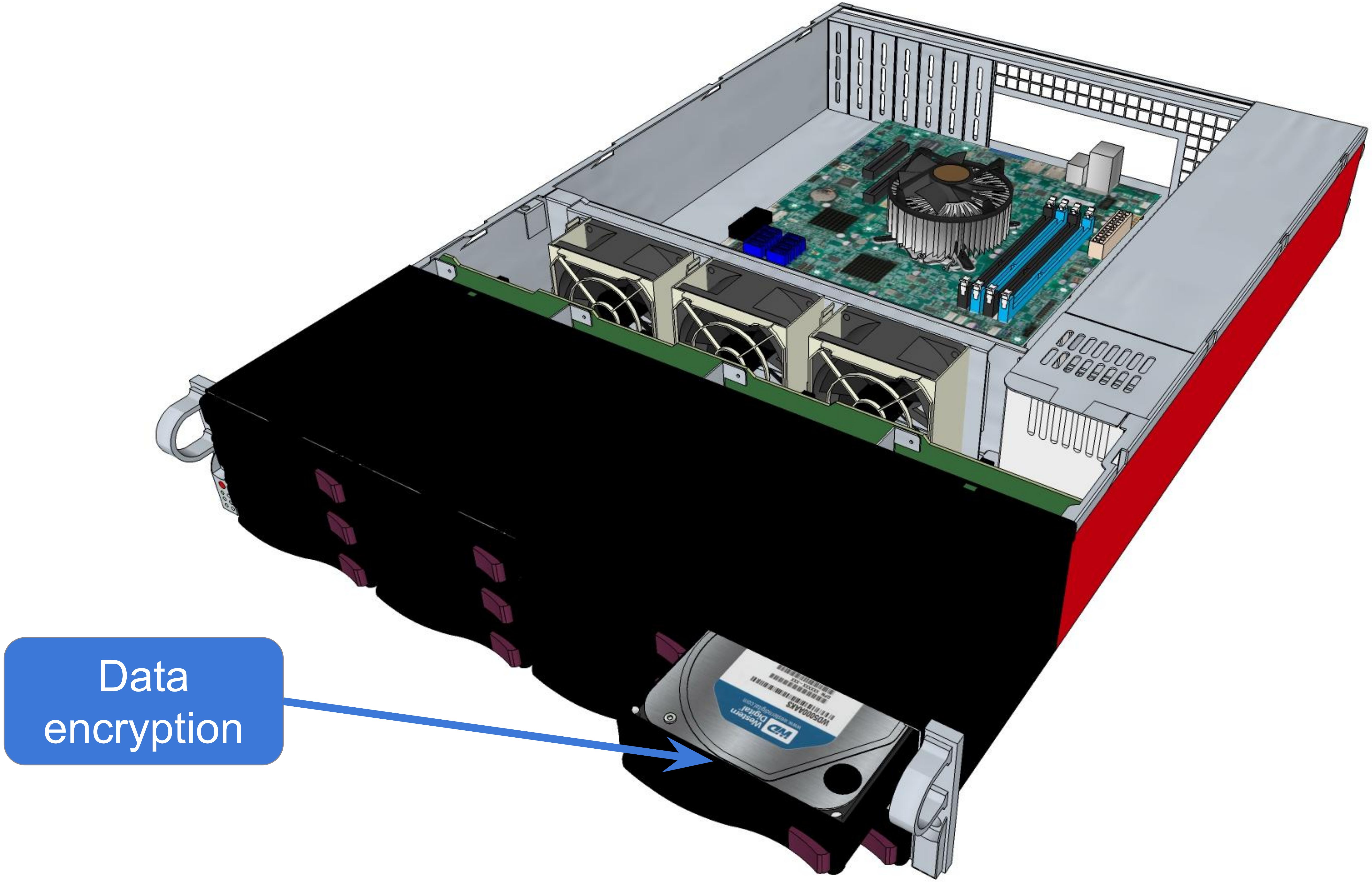
✉ m.zaborski@fudosecurity.com

✉ oshogbo@FreeBSD.org

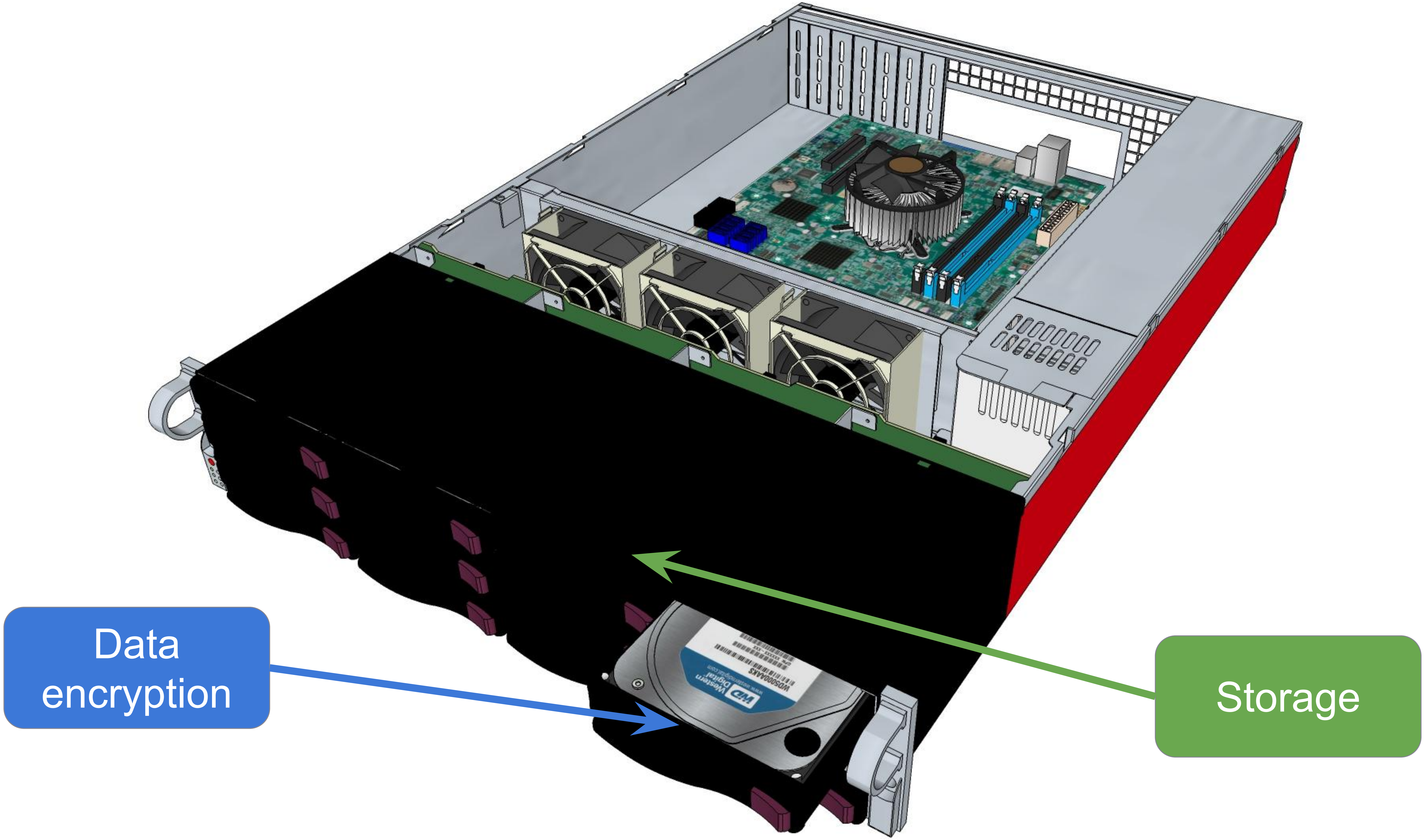
🌐 https://oshogbo.vexillium.org

🐦 @oshogbovx


WD
Western Digital
WD5000AAKS

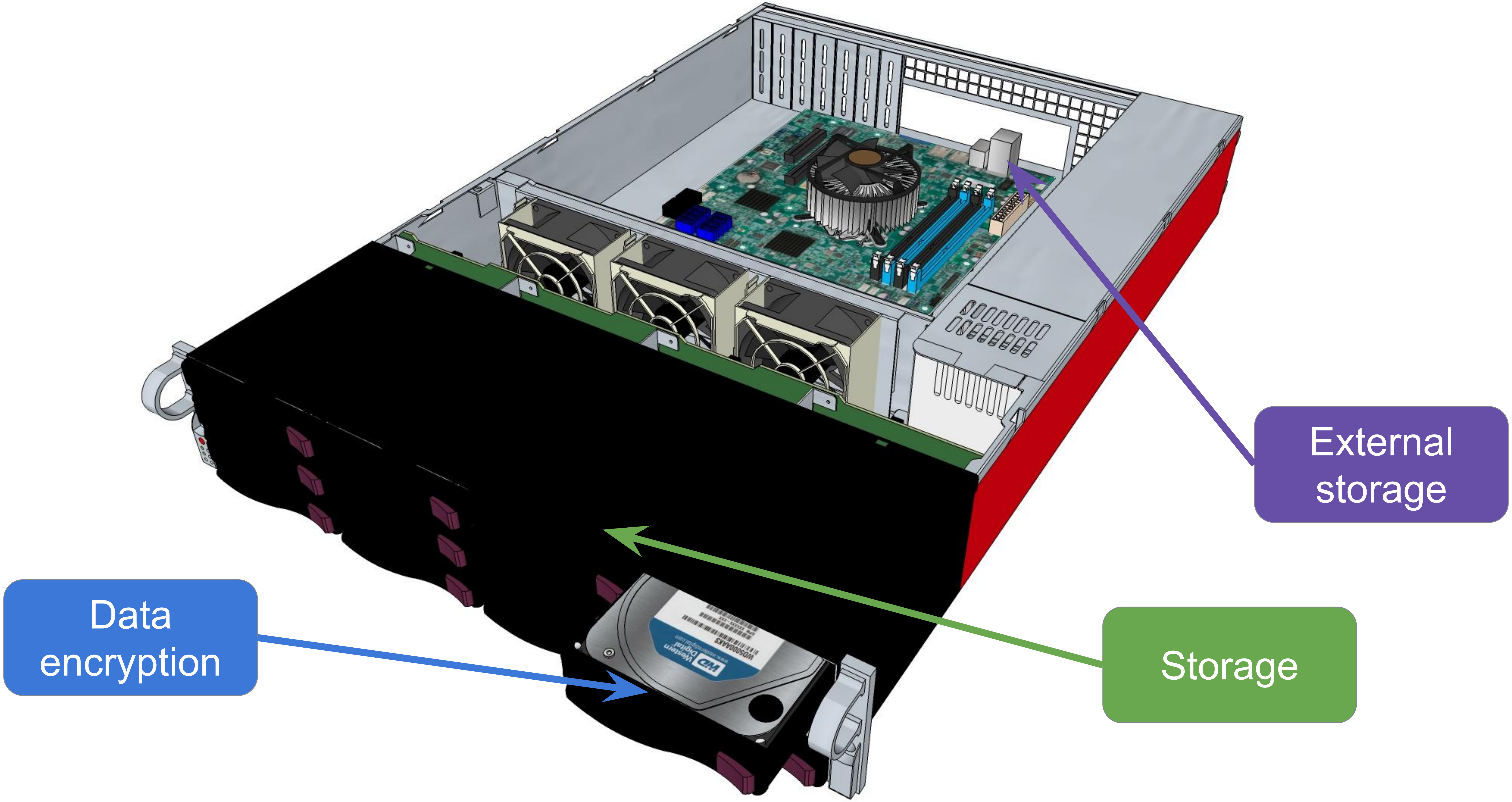


Data encryption



Data encryption

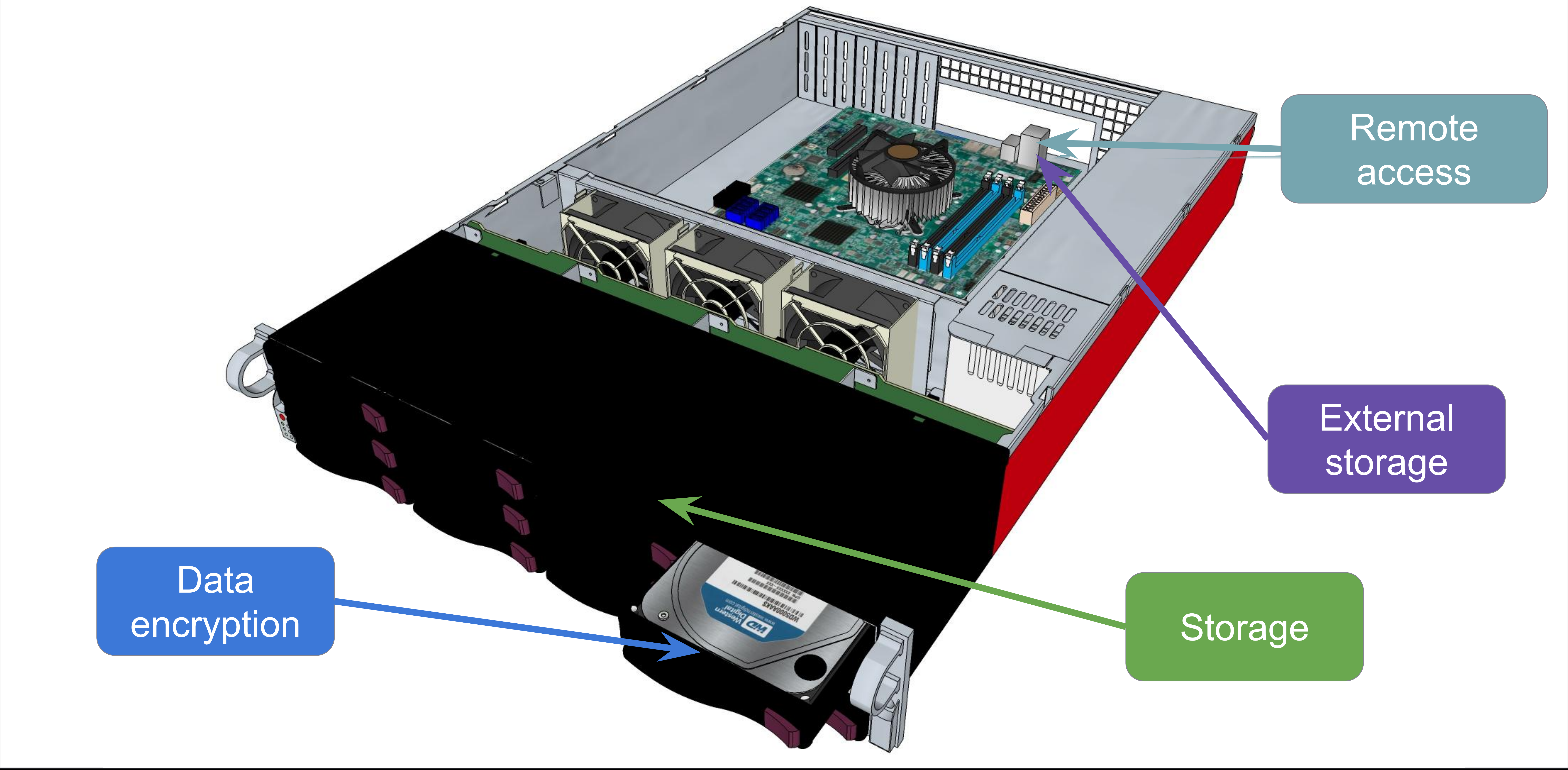
Storage

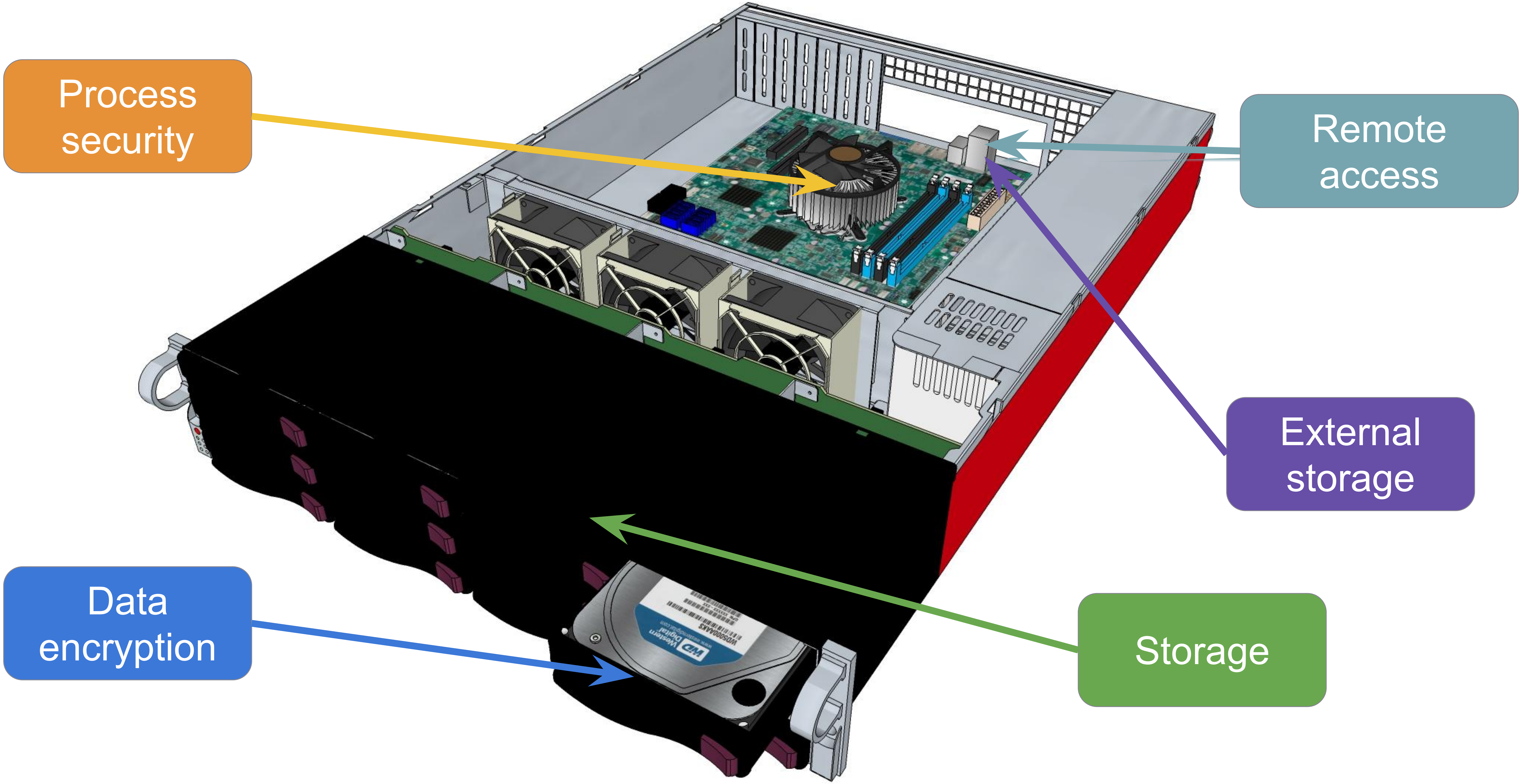


External storage

Storage

Data encryption


Remote access
External storage
Storage
Data encryption



Process security

Remote access

External storage

Storage

Data encryption

Data Encryption

Data Encryption

- GBDE
- GELI
- native ZFS encryption

GBDE - Geom Based Disk Encryption

- FreeBSD 5.0
- AES-CBC 128bits
- Different key for each write
  - CPU overhead
  - disk space overhead

GELI

- Many cryptographic algorithms
  - AES-XTS
  - AES-CBC
  - Blowfish-CBC
  - Camellia-CBC
  - 3DES-CBC
- Integrity verification (HMAC)
- Don’t have such overheads like GDBE
- One-time key

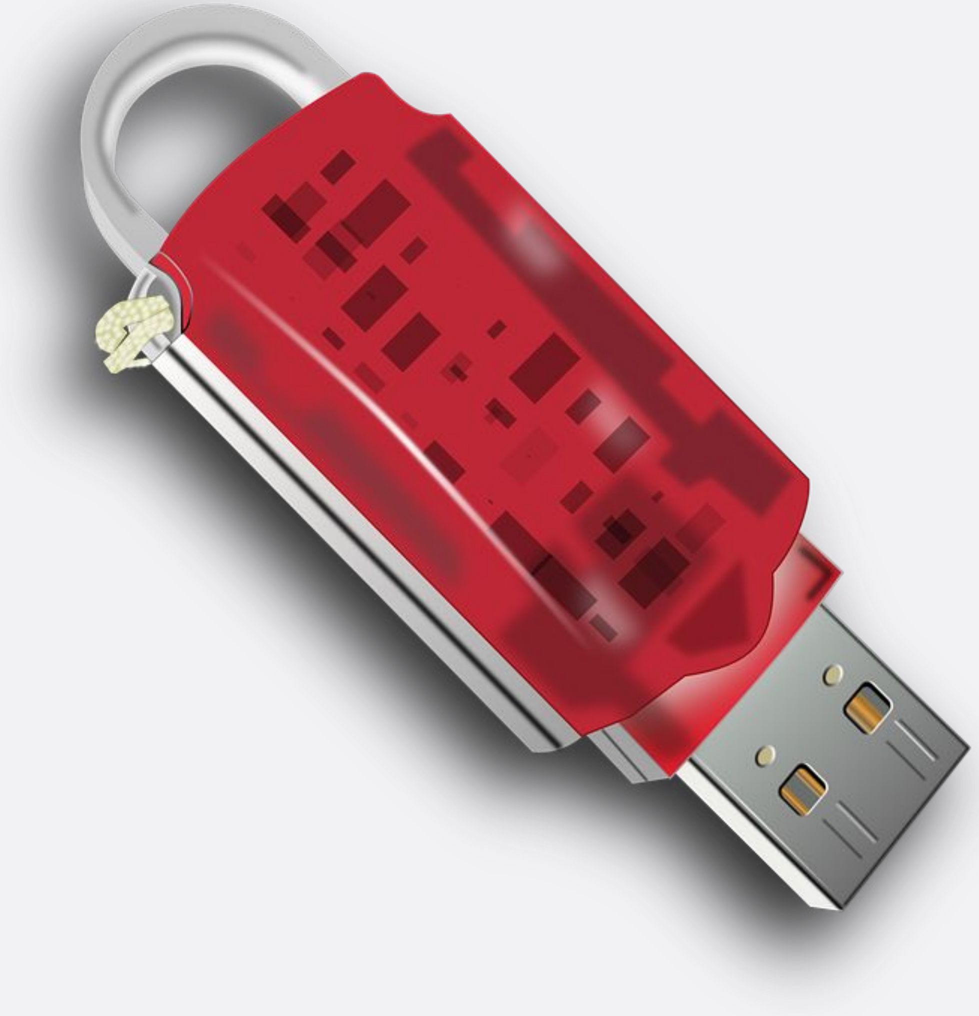
Keeping encryption key

Appliance:

- Use memstick
- Need only during boot
- Initialize during first boot

VM:

- Use passphrase
- Use no encryption



Storage

Storage

- ZFS
- UFS

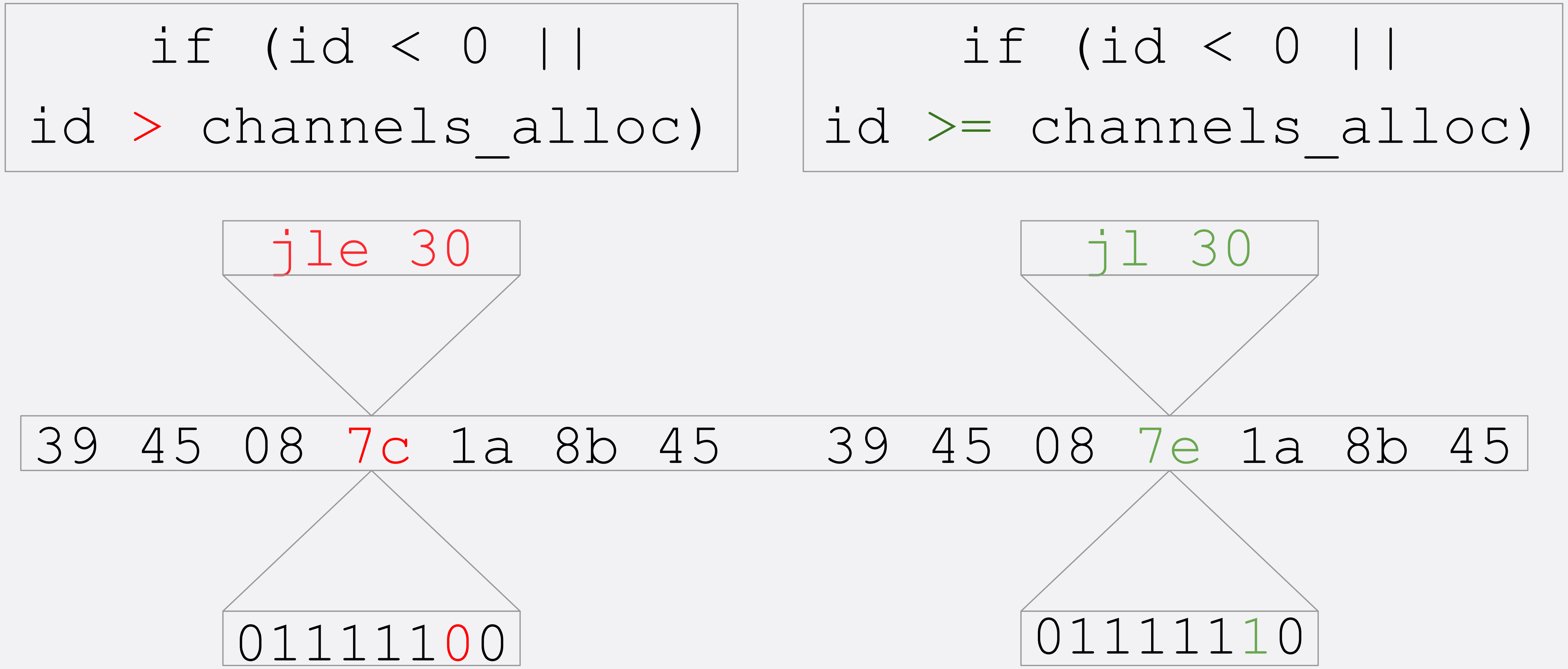
ZFS

- checksums
- snapshots
- compression
- RAIDZ

OpenZFS

ZFS - checksum

- fletcher2
- fletcher4
- sha256
- sha512
- skein



```
    if (id < 0 ||
id > channels_alloc)
```

```
jle 30
```

```
39 45 08 7c 1a 8b 45
```

```
01111100
```

```
    if (id < 0 ||
id >= channels_alloc)
```

```
jl 30
```

```
39 45 08 7e 1a 8b 45
```

```
01111110
```

ZFS - compression

- GZIP
- lz4
- ZSTD

```
# zfs list -o name,compression,compressratio
NAME                  COMPRESS   RATIO
data/data/local/dumps     lz4   16.20x
data/tmp                  lz4    1.00x
data/var/crash            lz4   11.17x
```

ZFS - compression

- GZIP
- lz4
- ZSTD

```
# zfs list -o name,compression,compressratio
NAME                       COMPRESS   RATIO
data/data/local/dumps         lz4    16.20x
data/tmp                      lz4     1.00x
data/var/crash                lz4    11.17x
```

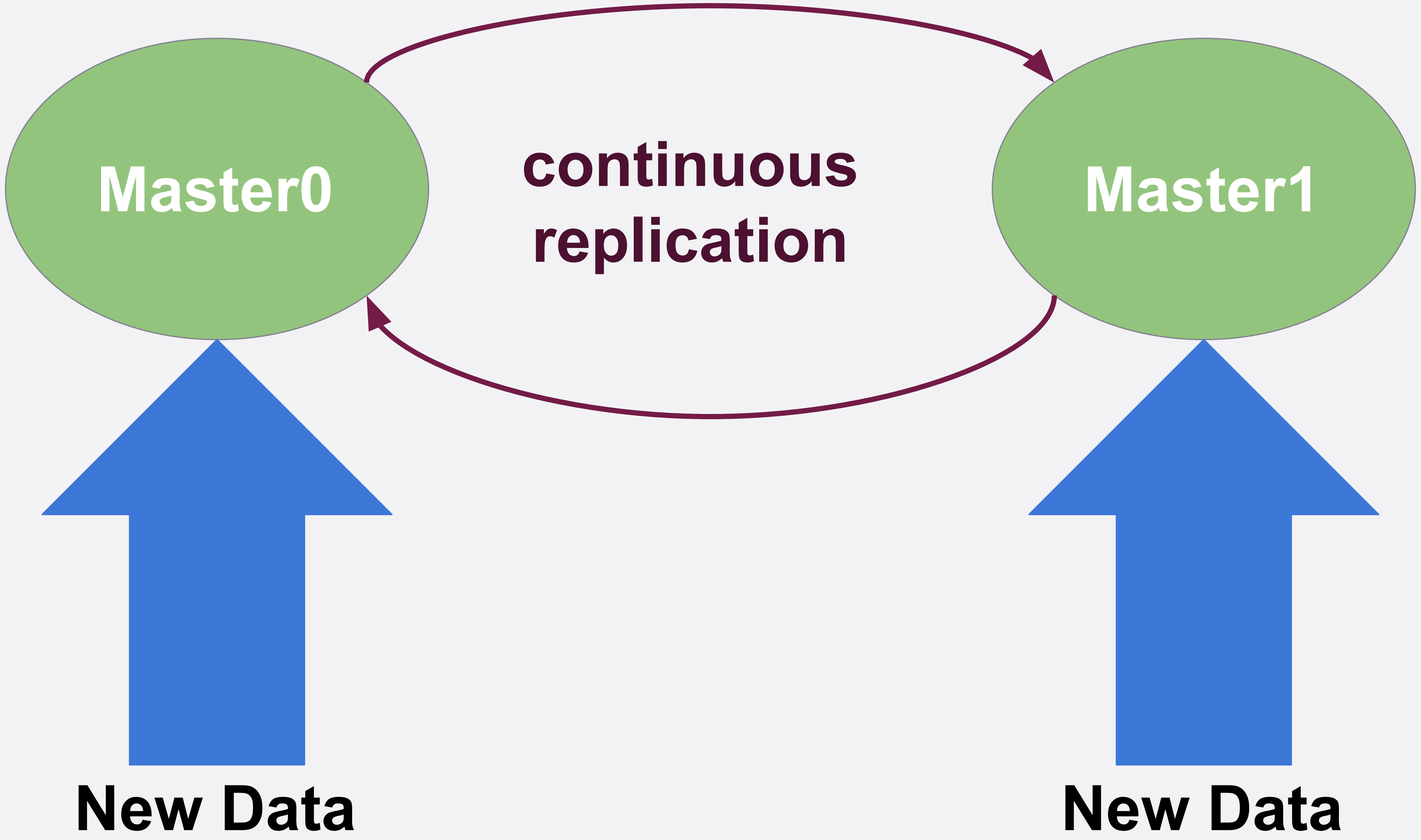
Problem: What if customer want to backup the data?

ZFS - snapshots

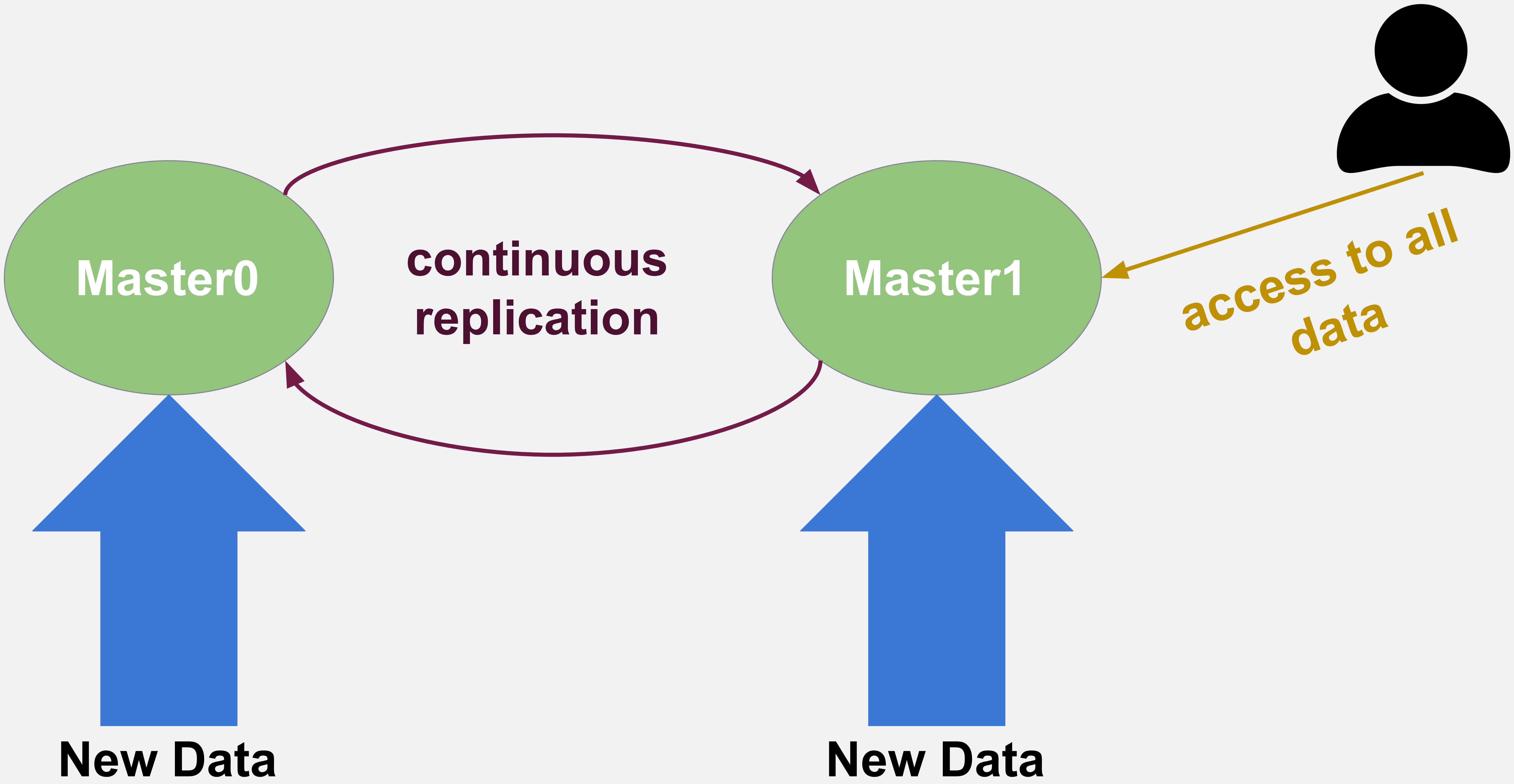
A **snapshot** is a read-only copy of a file system or volume. Snapshots can be created almost instantly, and they initially consume o additional disk space within the pool. However, as data within the active dataset changes, the snapshot consumes disk space by continuing to reference the old data, thus preventing the disk space from being freed.

https://docs.oracle.com/cd/E23824_01/html/821-1448/gbciq.html

Snapshots - cluster multi-master



Snapshots - cluster multi-master



Snapshots - cluster multi-master

```
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
data                      135G  7.93T   192K   /
data/data                 135G  7.93T  2.82M   /data
data/data/12345678/dumps  192K  7.93T   192K   /data/12345678/dumps
data/data/local/dumps    7.27G  7.93T  7.27G   /data/local/dumps
```

Snapshots - cluster multi-master

```
# zfs list -t snapshot
NAME                                     USED  AVAIL  REFER MOUNTPOINT
data/data/12345678/dumps@20180130051939     0      -    192k  -
data/data/local/dumps@20180130051934        0      -   7.27G  -
data/data/local/dumps@20180130052038        0      -    192k  -
```

ZFS sending & receiving snapshots

Before r317414:

- ZFS decompress FS to send
- manual compress FS to reduce latency
- sending over SSH
- manual decompress FS received over SSH
- ZFS compress FS which was received

After r317414:

- ZFS FS send over SSH
- ZFS FS receive over SSH

Downside of using ZFS snapshots

- Data loss after rollback
- Can’t rollback ZFS changes
- Snapshots can take a lot of space on cluster multi-master

Downside of using ZFS snapshots

- Data loss after rollback
- Can't rollback ZFS changes

checkpoints

- Snapshots can take a lot of space on cluster multi-master

bookmarks

OpenZFS

Downsides of ZFS

- Not enough RAM to import pool
- No full disk encryption
- If something will go very wrong we still want to be able to do something
- What about factory reset?

Read only file system - UFS

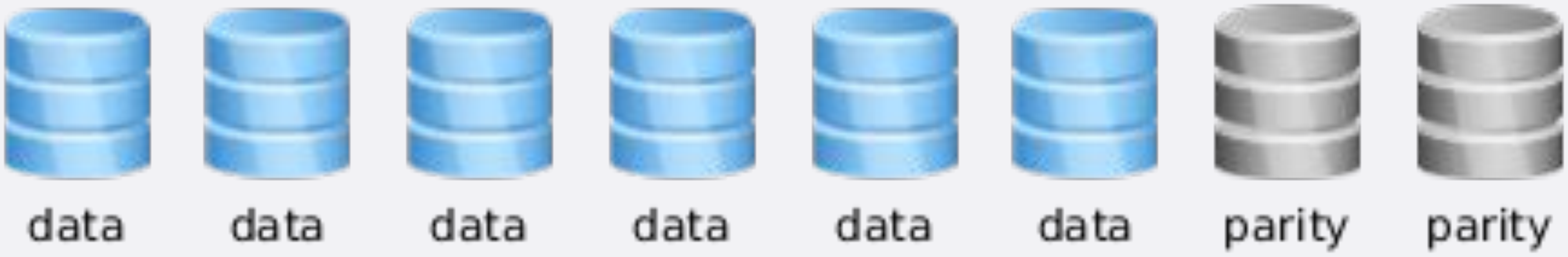
- GELI&ZFS for customer data

- Contains read-only operating system
- Data are not encrypted
- If something goes wrong we can still boot from it
- Try to reflect some ZFS promises

Read only file system - UFS

```
# gpart show -l ada0
=>      40  11721045101  ada0  GPT  (5.5T)
        40          128     1  boot0  (64K)
      168      8388608      2  system0-0  [bootme]  (4.0G)
  8388776      8388608      3  system1-0  (4.0G)
 16777384      8388608      4  system2-0  (4.0G)
 25165992     16572416      5  swap0  (7.9G)
 41738408  11679306727      6  data0  (5.4T)
```

RAIDZ2



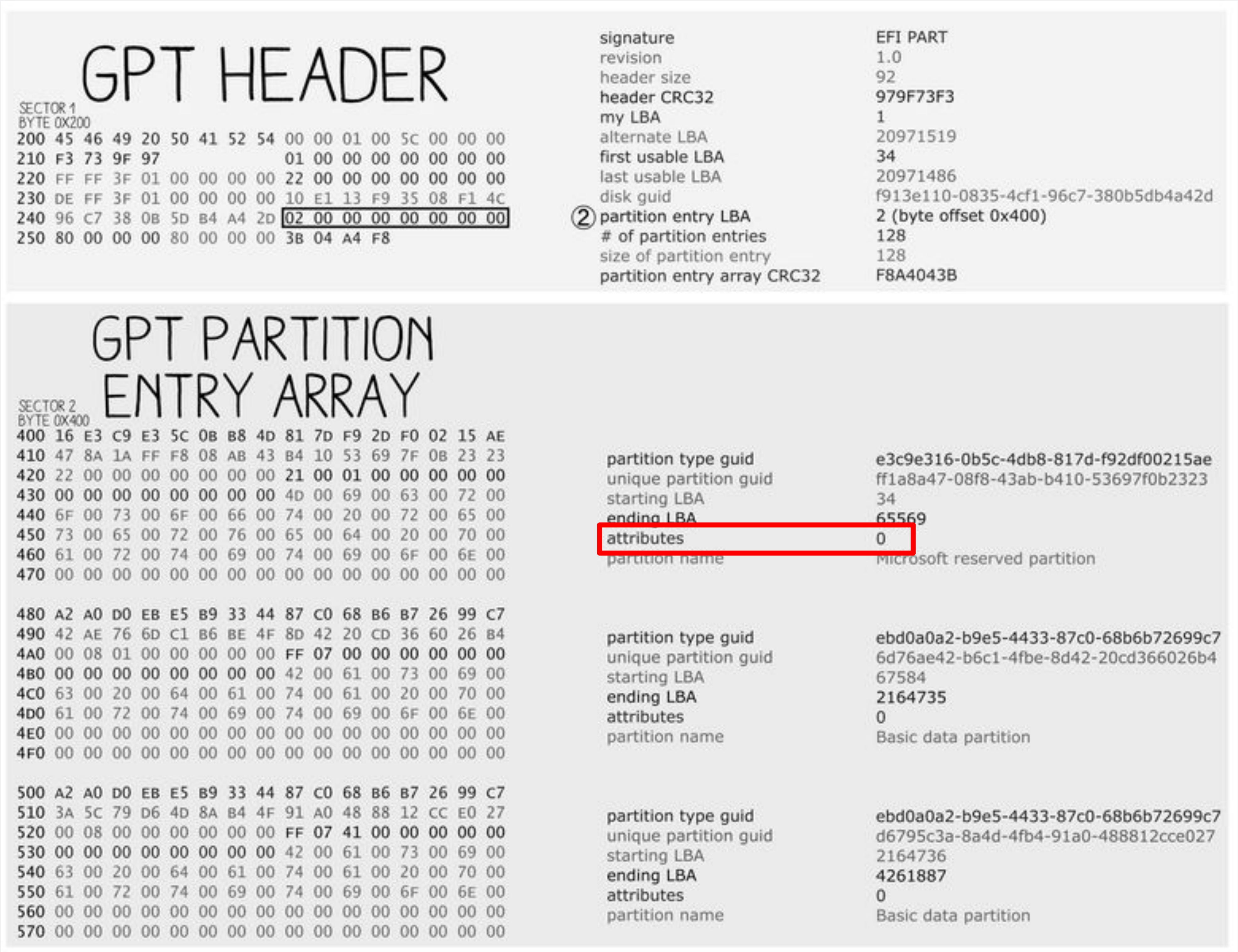
Reflect RAIDZ2 with UFS

```
         Name    Status  Components
mirror/system0  COMPLETE gpt/system0-0  (ACTIVE)
                         gpt/system0-1  (ACTIVE)
                         gpt/system0-2  (ACTIVE)
                         gpt/system0-3  (ACTIVE)
                         gpt/system0-4  (ACTIVE)
                         gpt/system0-5  (ACTIVE)
```

Reflect RAIDZ2 with SWAP

```
       Name     Status  Components
mirror/swap0  COMPLETE gpt/swap1  (ACTIVE)
                       gpt/swap2  (ACTIVE)
                       gpt/swap0  (ACTIVE)
mirror/swap1  COMPLETE gpt/swap3  (ACTIVE)
                       gpt/swap4  (ACTIVE)
                       gpt/swap5  (ACTIVE)
```

Upgrade steps



GPT HEADER

```
SECTOR 1
BYTE 0X200
200 45 46 49 20 50 41 52 54 00 00 01 00 5C 00 00 00
210 F3 73 9F 97             01 00 00 00 00 00 00 00
220 FF FF 3F 01 00 00 00 00 22 00 00 00 00 00 00 00
230 DE FF 3F 01 00 00 00 00 10 E1 13 F9 35 08 F1 4C
240 96 C7 38 0B 5D B4 A4 2D 02 00 00 00 00 00 00 00
250 80 00 00 00 80 00 00 00 3B 04 A4 F8
```

signature	EFI PART
revision	1.0
header size	92
header CRC32	979F73F3
my LBA	1
alternate LBA	20971519
first usable LBA	34
last usable LBA	20971486
disk guid	f913e110-0835-4cf1-96c7-380b5db4a42d
② partition entry LBA	2 (byte offset 0x400)
# of partition entries	128
size of partition entry	128
partition entry array CRC32	F8A4043B

GPT PARTITION ENTRY ARRAY

```
SECTOR 2
BYTE 0X400
400 16 E3 C9 E3 5C 0B B8 4D 81 7D F9 2D F0 02 15 AE
410 47 8A 1A FF F8 08 AB 43 B4 10 53 69 7F 0B 23 23
420 22 00 00 00 00 00 00 00 21 00 01 00 00 00 00 00
430 00 00 00 00 00 00 00 00 4D 00 69 00 63 00 72 00
440 6F 00 73 00 6F 00 66 00 74 00 20 00 72 00 65 00
450 73 00 65 00 72 00 76 00 65 00 64 00 20 00 70 00
460 61 00 72 00 74 00 69 00 74 00 69 00 6F 00 6E 00
470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

480 A2 A0 D0 EB E5 B9 33 44 87 C0 68 B6 B7 26 99 C7
490 42 AE 76 6D C1 B6 BE 4F 8D 42 20 CD 36 60 26 B4
4A0 00 08 01 00 00 00 00 00 FF 07 00 00 00 00 00 00
4B0 00 00 00 00 00 00 00 00 42 00 61 00 73 00 69 00
4C0 63 00 20 00 64 00 61 00 74 00 61 00 20 00 70 00
4D0 61 00 72 00 74 00 69 00 74 00 69 00 6F 00 6E 00
4E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

500 A2 A0 D0 EB E5 B9 33 44 87 C0 68 B6 B7 26 99 C7
510 3A 5C 79 D6 4D 8A B4 4F 91 A0 48 88 12 CC E0 27
520 00 08 00 00 00 00 00 00 FF 07 41 00 00 00 00 00
530 00 00 00 00 00 00 00 00 42 00 61 00 73 00 69 00
540 63 00 20 00 64 00 61 00 74 00 61 00 20 00 70 00
550 61 00 72 00 74 00 69 00 74 00 69 00 6F 00 6E 00
560 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
570 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

partition type guid	e3c9e316-0b5c-4db8-817d-f92df00215ae
unique partition guid	ff1a8a47-08f8-43ab-b410-53697f0b2323
starting LBA	34
ending LBA	65569
attributes	0
partition name	Microsoft reserved partition

partition type guid	ebd0a0a2-b9e5-4433-87c0-68b6b72699c7
unique partition guid	6d76ae42-b6c1-4fbe-8d42-20cd366026b4
starting LBA	67584
ending LBA	2164735
attributes	0
partition name	Basic data partition

partition type guid	ebd0a0a2-b9e5-4433-87c0-68b6b72699c7
unique partition guid	d6795c3a-8a4d-4fb4-91a0-488812cce027
starting LBA	2164736
ending LBA	4261887
attributes	0
partition name	Basic data partition

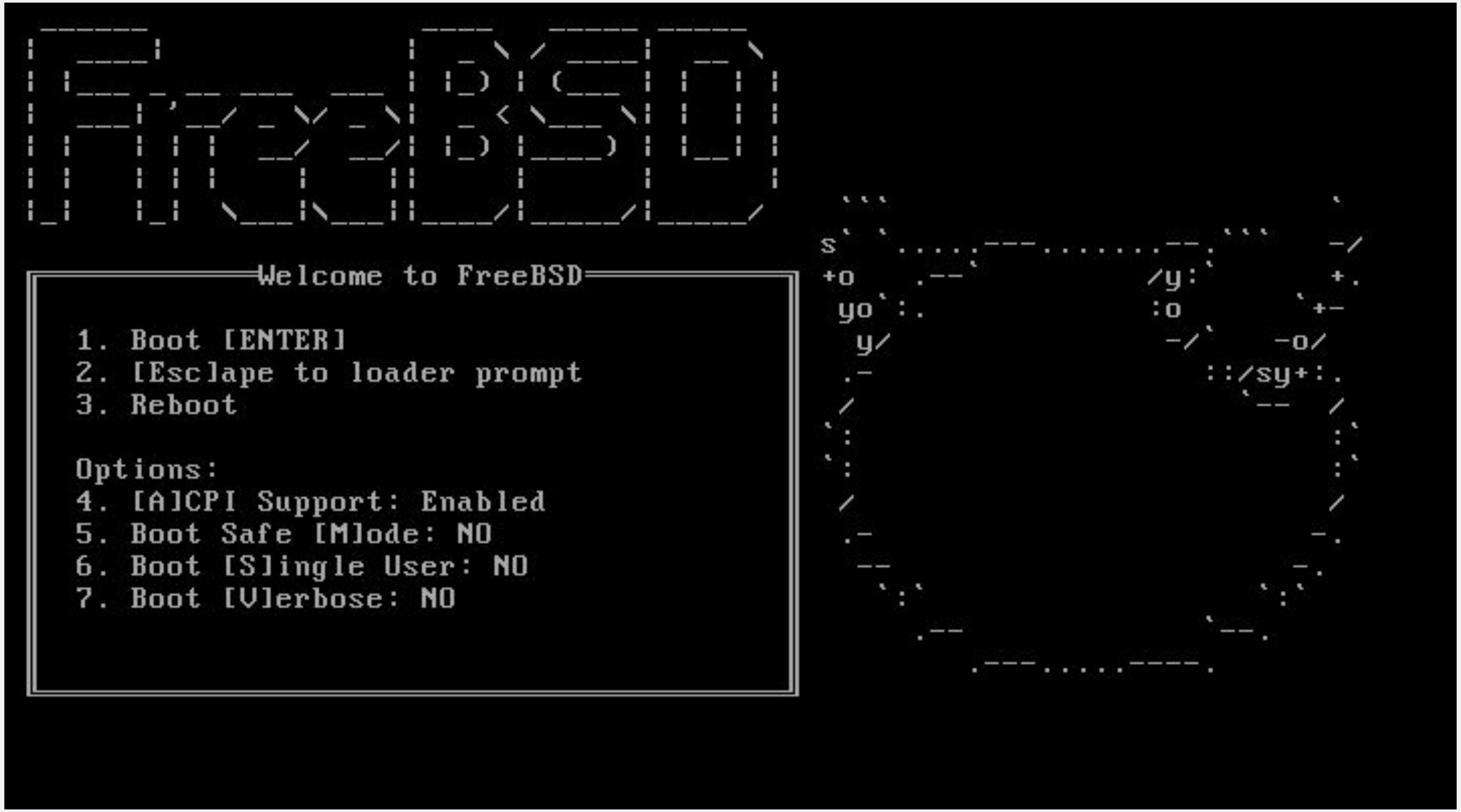
Upgrade steps - Boot from system0

```
# gpart show -l ada0
=>         40  11721045101  ada0  GPT  (5.5T)
           40          128     1  boot0  (64K)
          168      8388608     2  system0-0  [bootme]  (4.0G)
      8388776      8388608     3  system1-0  (4.0G)
     16777384      8388608     4  system2-0  (4.0G)
     25165992     16572416     5  swap0  (7.9G)
     41738408  11679306727     6  data0  (5.4T)
```

Upgrade steps - override system1 and set bootonce

```
# gpart show -l ada0
=>        40  11721045101    ada0  GPT  (5.5T)
          40          128       1  boot0  (64K)
         168      8388608       2  system0-0  [bootme]  (4.0G)
     8388776      8388608       3  system1-0  [bootonce, bootme]  (4.0G)
    16777384      8388608       4  system2-0  (4.0G)
    25165992     16572416       5  swap0  (7.9G)
    41738408  11679306727       6  data0  (5.4T)
```

Upgrade steps - reboot



```
Welcome to FreeBSD

 1. Boot [ENTER]
 2. [Esc]ape to loader prompt
 3. Reboot

 Options:
 4. [A]CPI Support: Enabled
 5. Boot Safe [M]ode: NO
 6. Boot [S]ingle User: NO
 7. Boot [V]erbose: NO
```

Upgrade steps - bootloader removes bootme

```
# gpart show -l ada0
=>       40  11721045101  ada0  GPT  (5.5T)
         40          128     1  boot0  (64K)
        168      8388608     2  system0-0  [bootme]  (4.0G)
    8388776      8388608     3  system1-0  [bootonce]  (4.0G)
   16777384      8388608     4  system2-0  (4.0G)
   25165992     16572416     5  swap0  (7.9G)
   41738408  11679306727     6  data0  (5.4T)
```

Upgrade steps

- Create zfs snapshot

- Upgrade error accrued

- Reboot machine

Upgrade steps - boot from partition with bootme

```
# gpart show -l ada0
=>     40  11721045101   ada0  GPT  (5.5T)
       40          128        1  boot0   (64K)
      168      8388608        2  system0-0  [bootme]  (4.0G)
  8388776      8388608        3  system1-0  [bootonce]  (4.0G)
 16777384      8388608        4  system2-0  (4.0G)
 25165992     16572416        5  swap0  (7.9G)
 41738408  11679306727        6  data0  (5.4T)
```

Upgrade steps - rollback

```
# zfs rollback -R data@upgrade
# gpart show -l ada0
=>        40  11721045101   ada0  GPT  (5.5T)
          40          128        1  boot0  (64K)
         168      8388608        2  system0-0  [bootme]  (4.0G)
     8388776      8388608        3  system1-0  [bootfailed]  (4.0G)
    16777384      8388608        4  system2-0  (4.0G)
    25165992     16572416        5  swap0  (7.9G)
    41738408  11679306727        6  data0  (5.4T)
```

Upgrade steps - upgrade succeeded

```
# gpart show -l ada0
=>        40  11721045101  ada0  GPT  (5.5T)
          40          128     1  boot0  (64K)
         168      8388608     2  system0-0  [bootme]  (4.0G)
     8388776      8388608     3  system1-0  [bootonce]  (4.0G)
    16777384      8388608     4  system2-0  (4.0G)
    25165992     16572416     5  swap0  (7.9G)
    41738408  11679306727     6  data0  (5.4T)
```

Upgrade steps - upgrade succeeded

```
# gpart show -l ada0
=>         40  11721045101   ada0  GPT  (5.5T)
           40          128      1  boot0  (64K)
          168      8388608      2  system0-0  (4.0G)
      8388776      8388608      3  system1-0  [bootme]  (4.0G)
     16777384      8388608      4  system2-0  (4.0G)
     25165992     16572416      5  swap0  (7.9G)
     41738408  11679306727      6  data0  (5.4T)
```

Hot plug

```
notify 20 {
  match "system" "DEVFS";
  match "type" "CREATE";
  match "cdev" "^ada[0-9]+$";
  action "/usr/local/bin/newdisk $cdev";
};
```

External Storage

External storage

- NFS
- iscsi
- SAN over FC

External storage - NFS

- NFS
- iscsi
- SAN over FC

- No encryption
- No authorization
- Is it corporate solution?
- Able to mount on multiple machines

External storage - iscsi

- NFS
- iscsi
- SAN over FC

- Encryption
- Authorization
- Is it corporate solution?
- Not able to mount on multiple machines

External storage - SAN over FC

- NFS
- iscsi
- SAN over FC

- Encryption
- Authorization
- It is a corporate solution
- Not able to mount on multiple machines

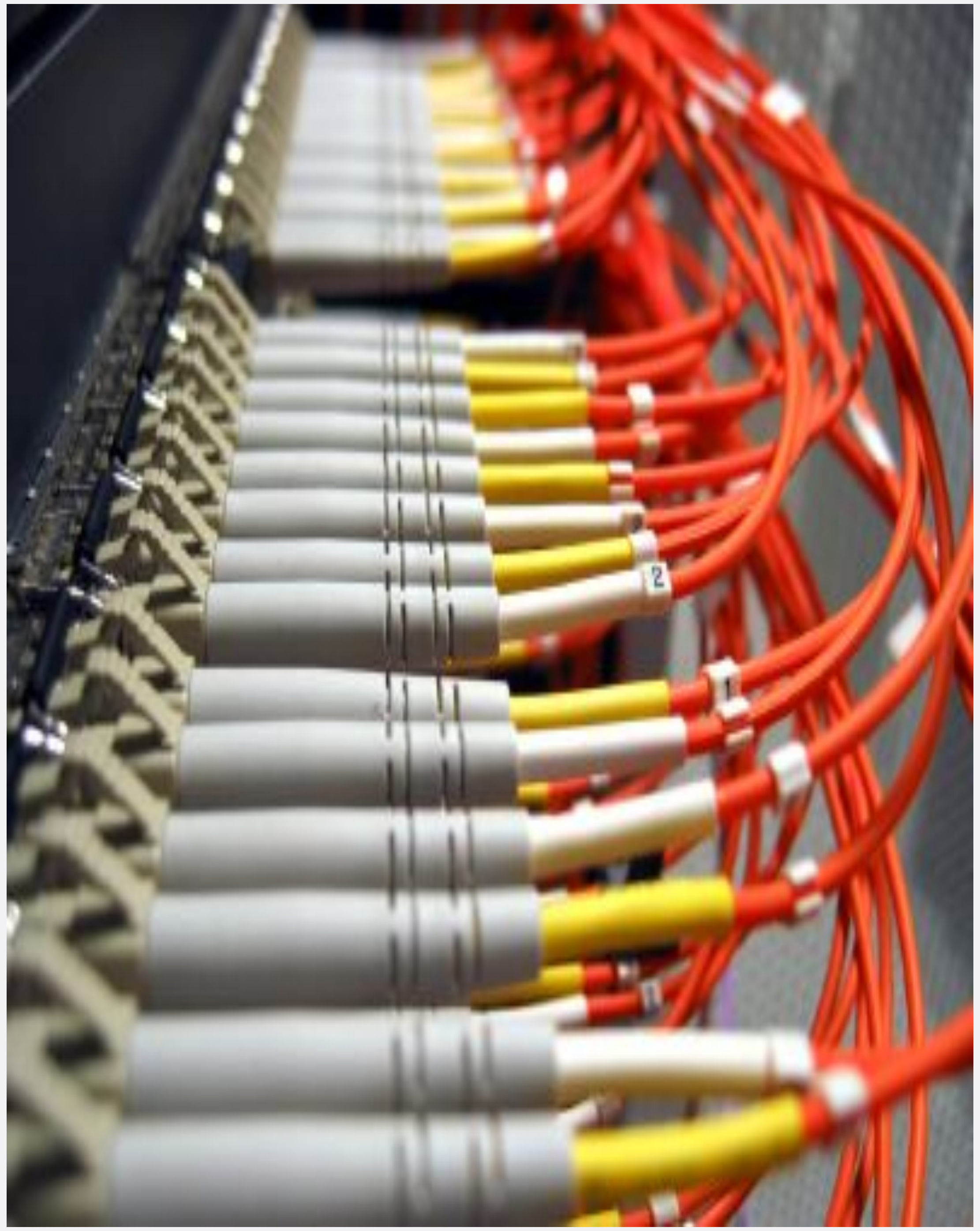
External storage - SAN over FC

- NFS
- iscsi
- SAN over FC with GELI

- Encryption
- Authorization
- It is a corporate solution
- Not able to mount on multiple machines

Redundancy

- Use at least two FC cards
- Combine multiple luns with gmultipath



Remote access

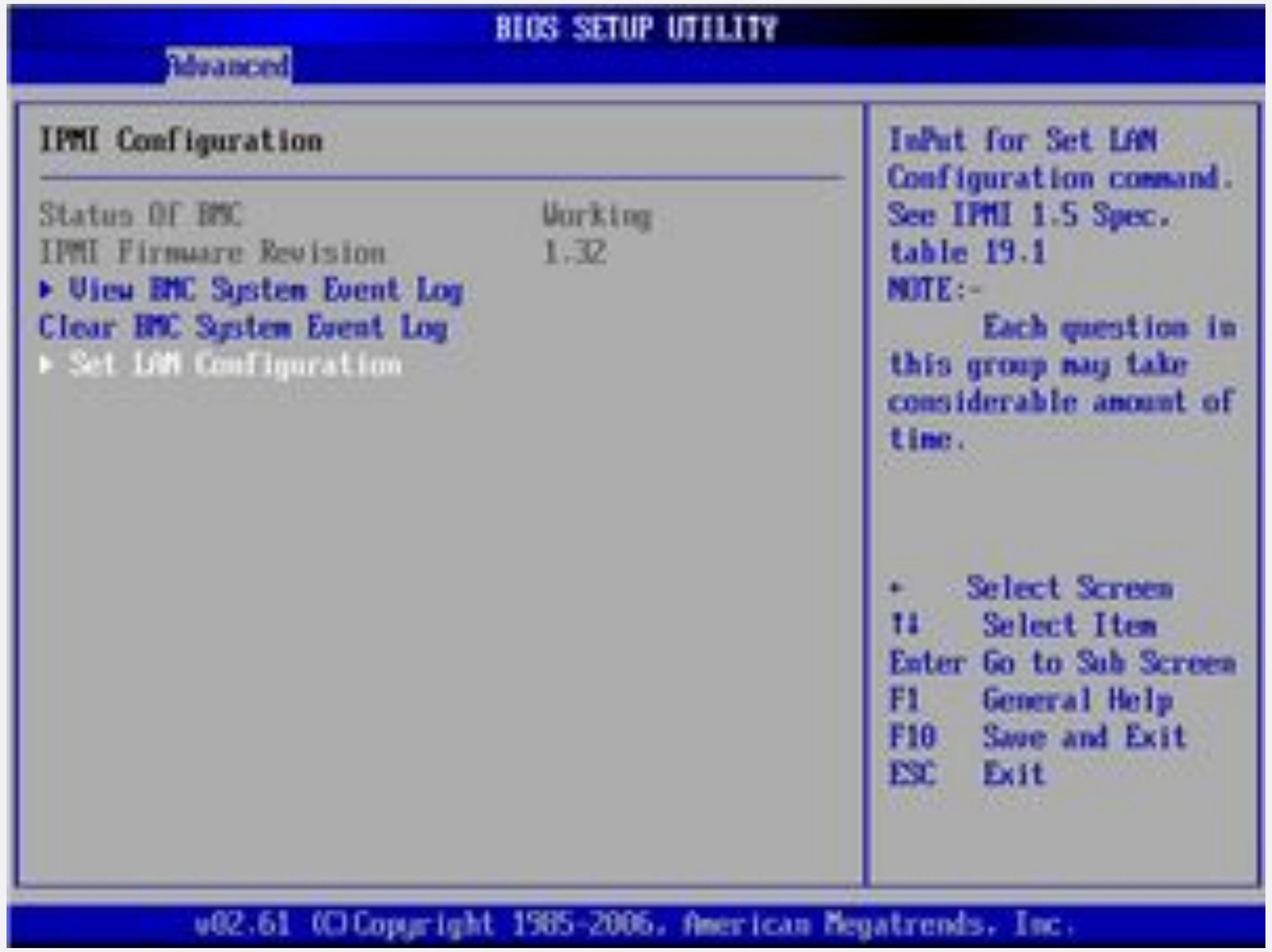
Access the box

- Through SSH
- We don’t want to customize our builds per client
- In theory we can have an key per client
- SSH keys
  - Hard to hijack
  - What if your engineer change the job?
  - We have to be in customer network

Access the box - exotic

- IPMI
- Some video conference (like webex)

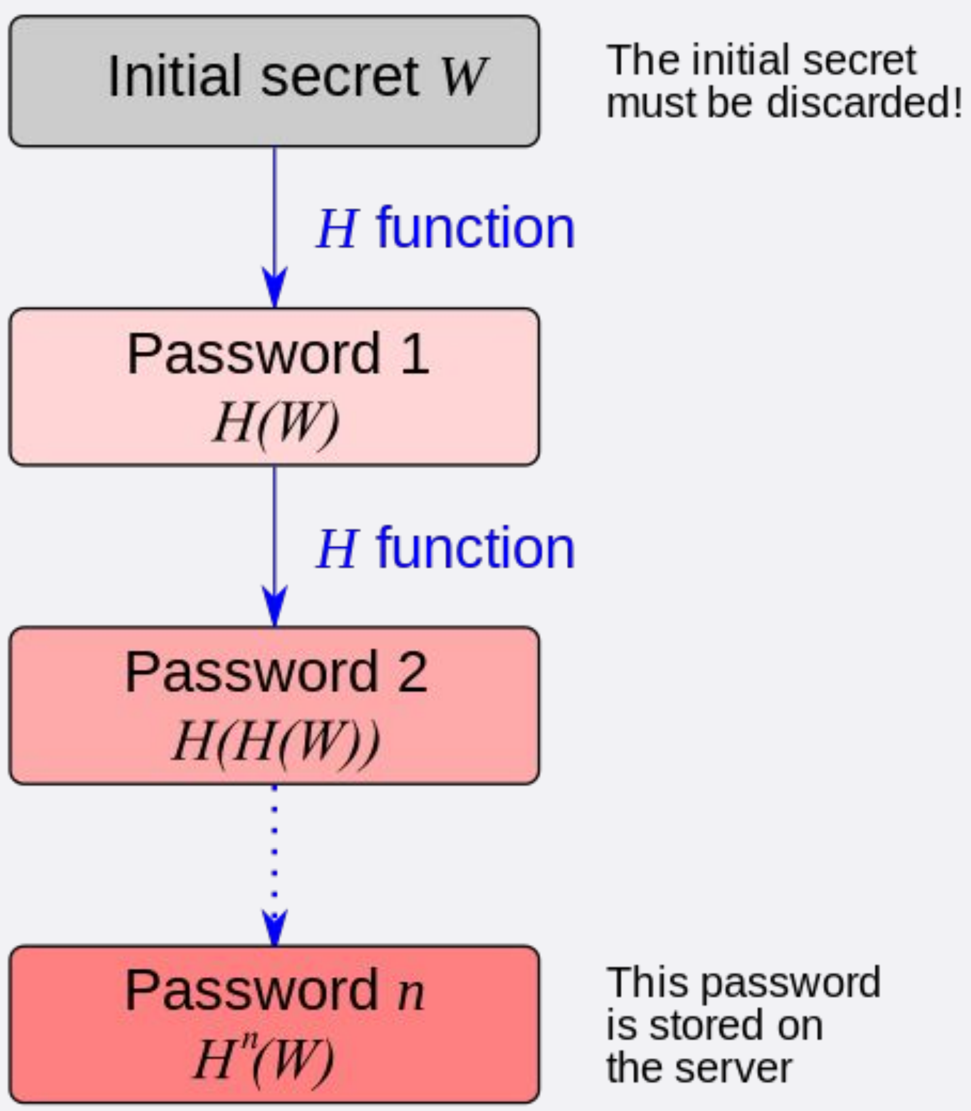
- No SSH keys
  - So maybe password after all?
  - But password is easy to hijack
  - What if yours enginner change the job?



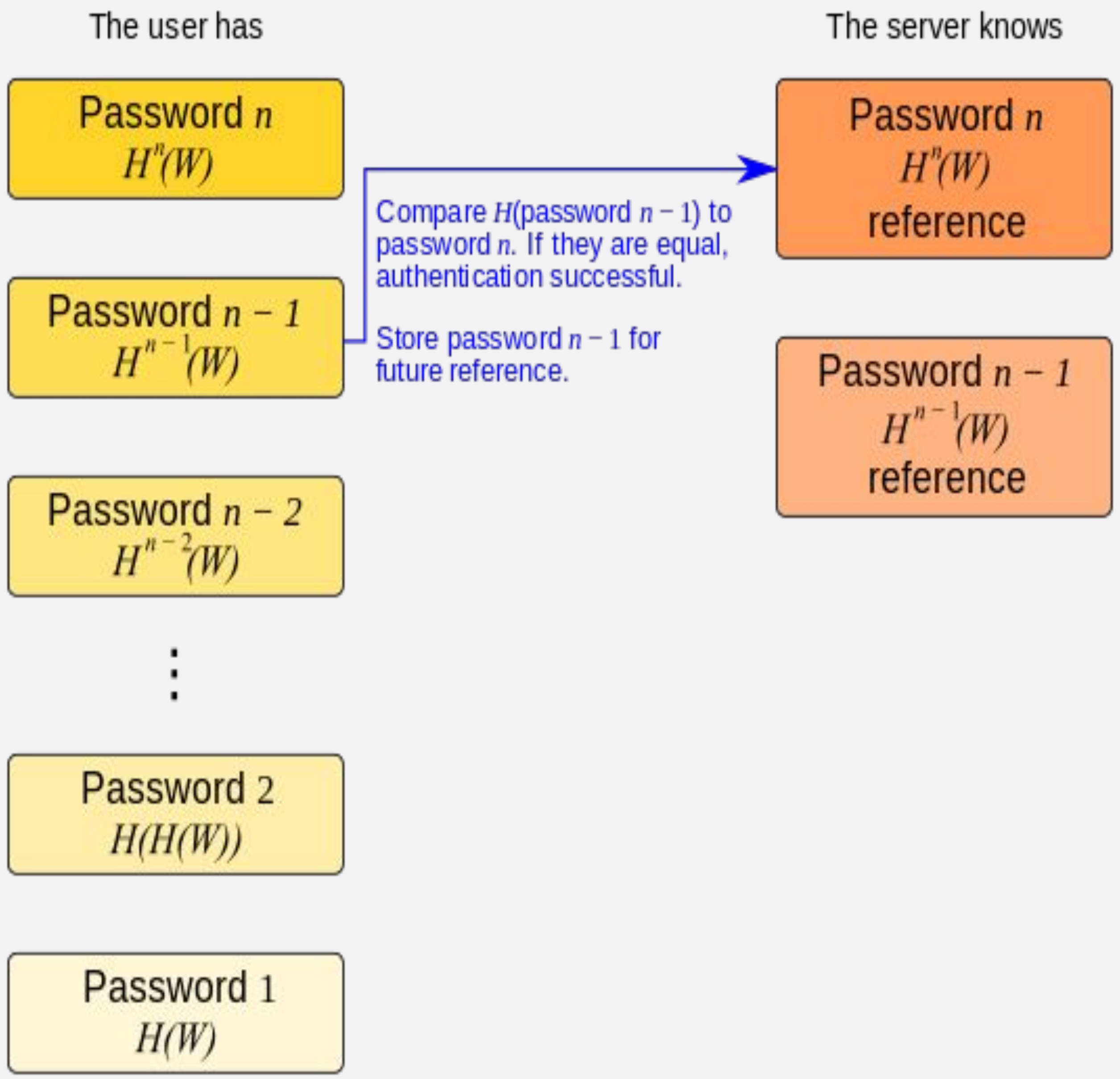
```
                              BIOS SETUP UTILITY
        Advanced

 IPMI Configuration                                  InPut for Set LAN
                                                     Configuration command.
 Status Of BMC                    Working            See IPMI 1.5 Spec.
 IPMI Firmware Revision           1.32               table 19.1
 ▶ View BMC System Event Log                         NOTE:-
 Clear BMC System Event Log                                Each question in
 ▶ Set LAN Configuration                             this group may take
                                                     considerable amount of
                                                     time.

                                                     ←     Select Screen
                                                     ↑↓    Select Item
                                                     Enter Go to Sub Screen
                                                     F1    General Help
                                                     F10   Save and Exit
                                                     ESC   Exit

            v02.61 (C)Copyright 1985-2006, American Megatrends, Inc.
```

Implementing S/Key (whlkey)

S/KEY password generation



S/KEY authentication



Implementing S/Key (whlkey)

- We configure it as:
  - 50 keys per day
  - The key length is 16 chars
  - Key is rotated every day
- Unified password:
  - O == 0, I == l, etc.
- The secret is stored in some trusted machine
- The engineer can only get keys for this week

Process security

Basic problem

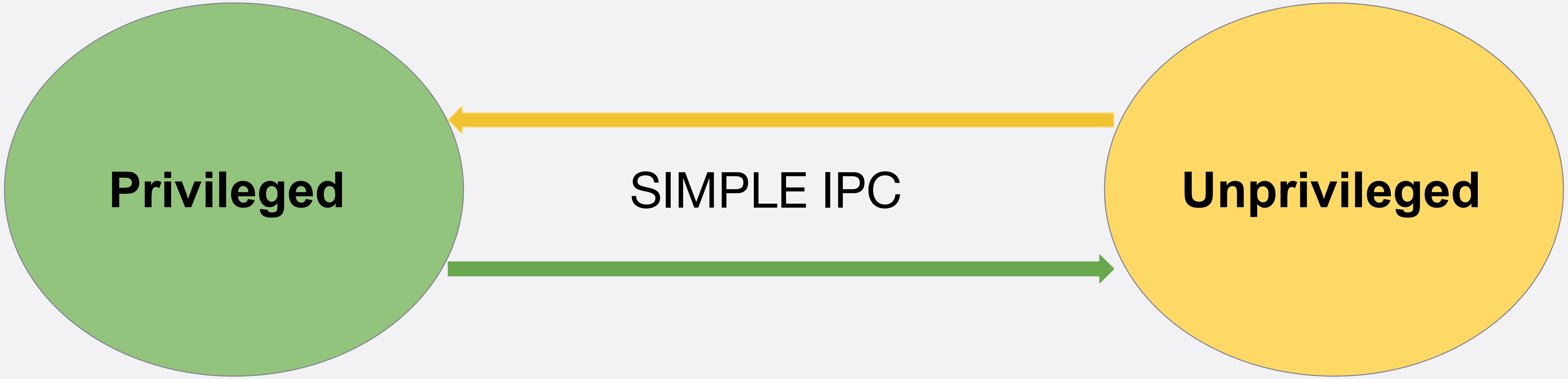
- You can’t build everything from scratch
- You can’t audit everything
- Who you really trust?

Basic problem

- You can’t build everything from scratch
- You can’t audit everything
- Who you really trust?

Security stops where the trust begins

Privileged separation



- Reduce TCB
- Simple communication

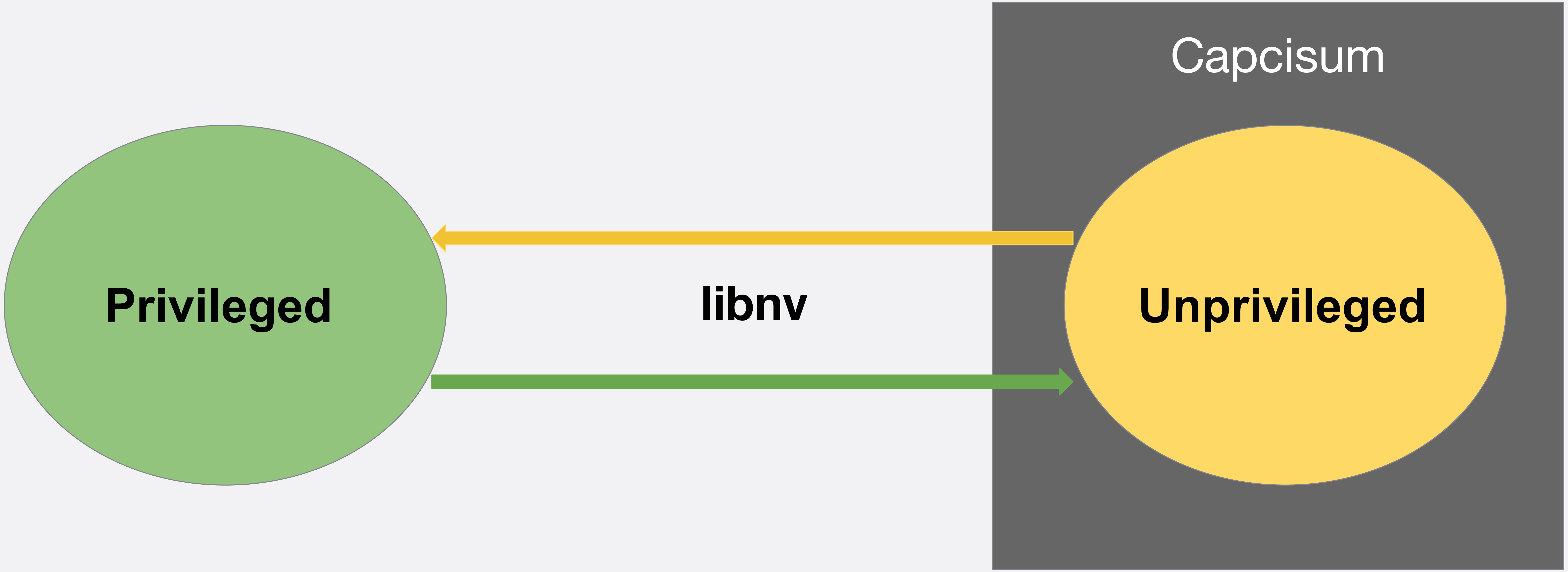
Privileged process

- Have access to:
  - DB
  - Storage
  - Network
- Authenticate unprivileged process
- Extend capabilities of unprivileged process

Unprivileged process

- Have access to storage by single FD
- Have access to network by single/two FD
- Implements complicate logic
- Is sending a simple commands asking privileged process
- Limited RAM
- Limited CPU time

Privileged separation with FreeBSD



Capsicum

- tight sandboxing (cap_enter(2))
- capability rights (cap_rights_limit(2))

Libnv

- nvlist_create
- nvlist_add_${type}
- nvlist_get_${type}
- nvlist_take_${type}
- nvlist_move_${type}
- nvlist_send
- nvlist_recv
- nvlist_destoy

- Types:
  - string
  - number
  - bool
  - nvlist
  - descriptor
  - binary
  - array

Privileged separation - is it hard?

OpenSSL

tesseract

leptonica

OpenSSH

libNTLM

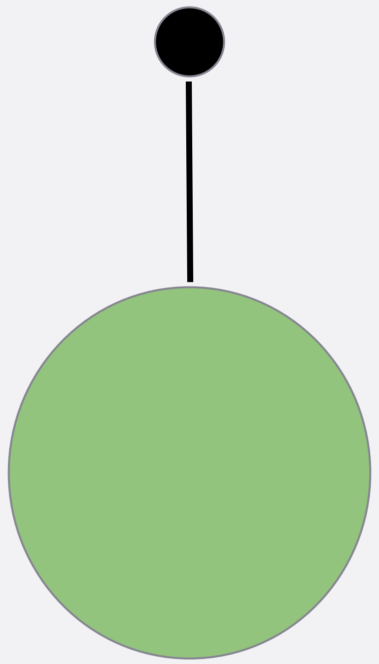
FreeRDP

FreeTDS

libX11

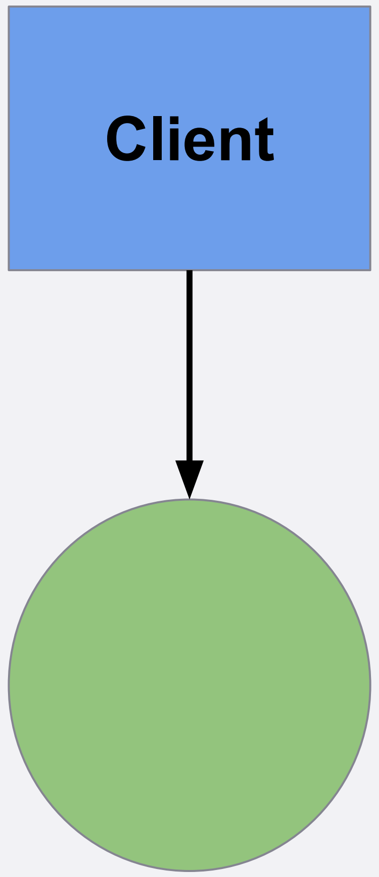
freetype

Privileged separation - network daemon



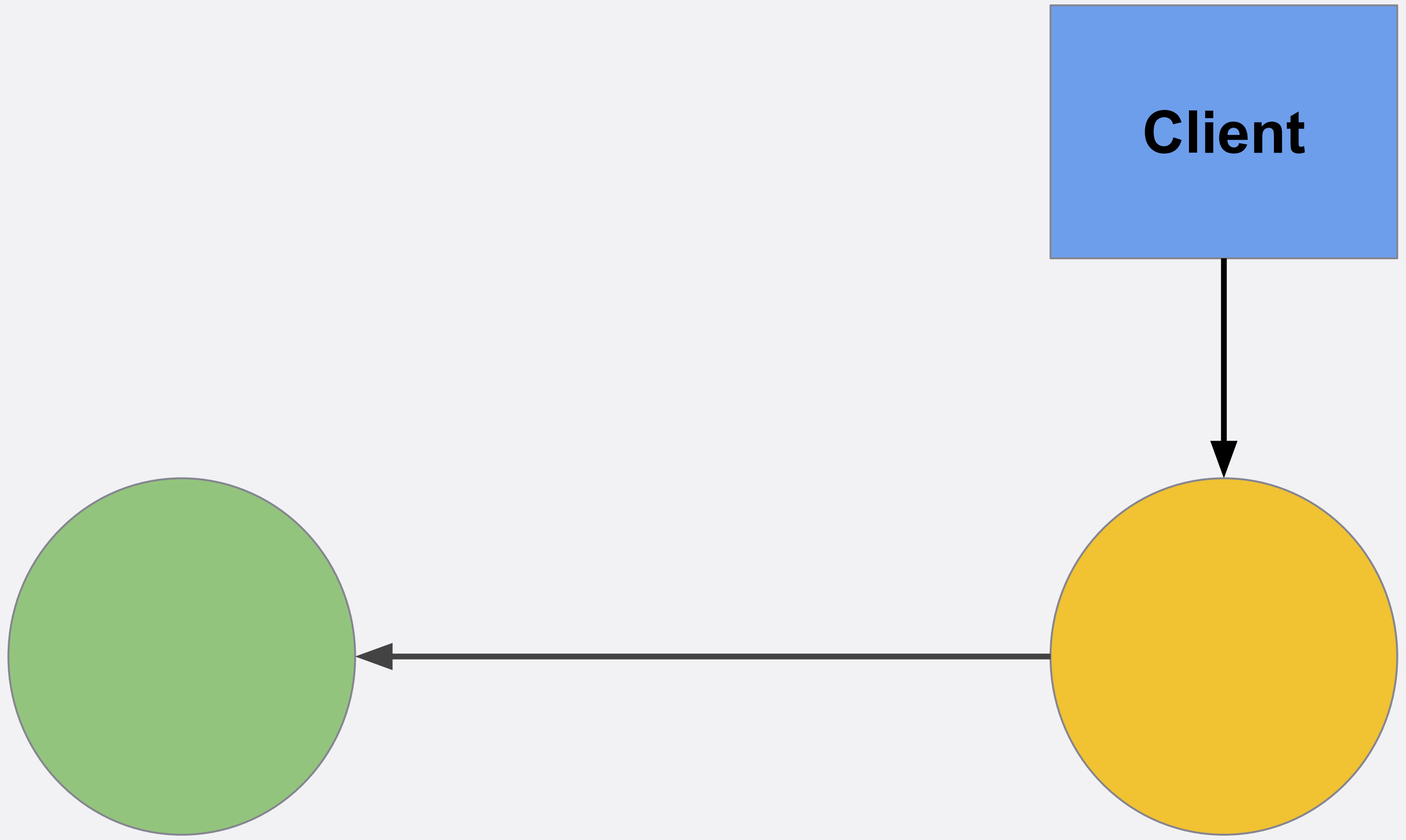
- Privileged process is waiting for connection

Privileged separation - network daemon



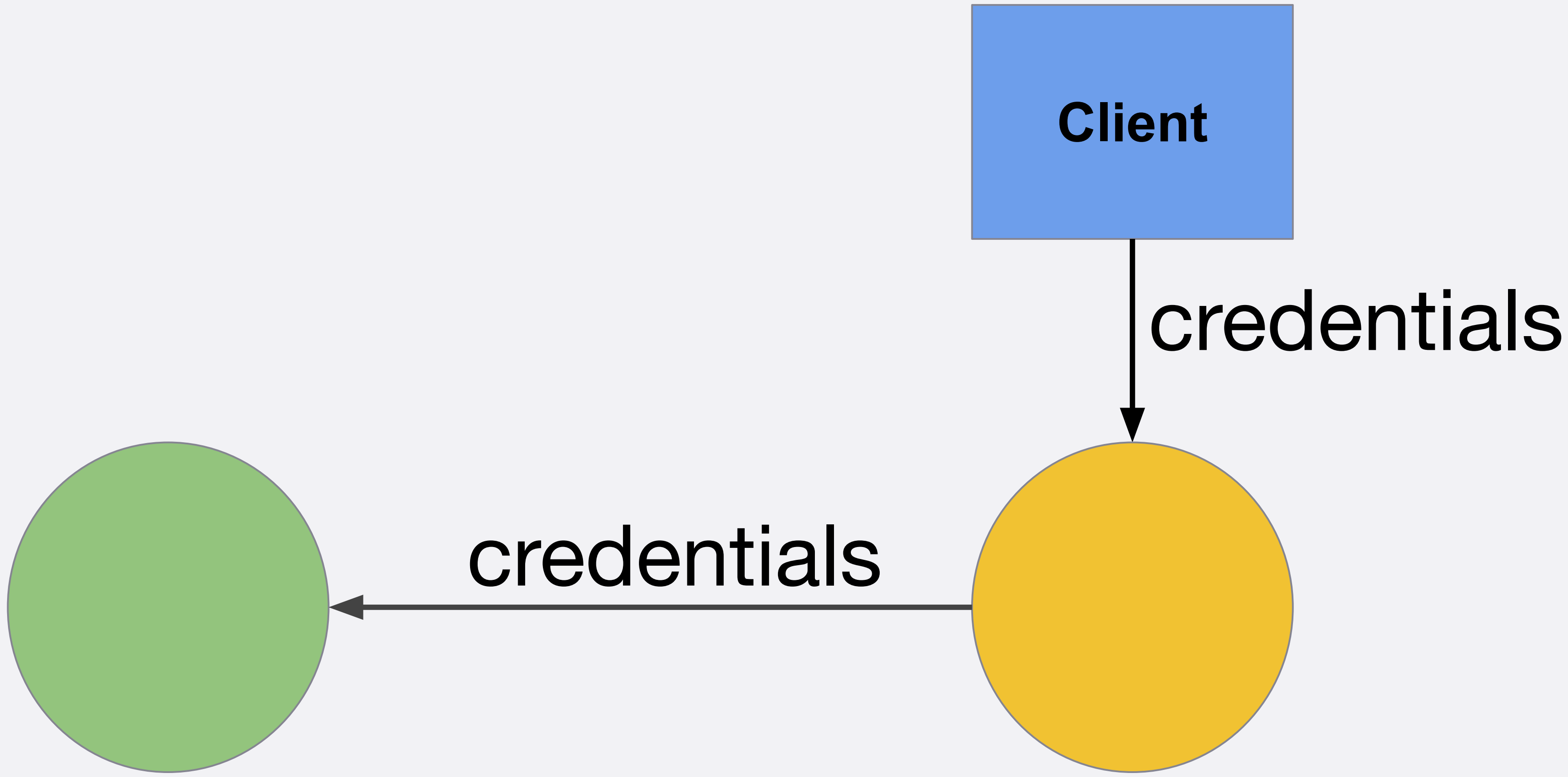
- Privileged process is waiting for connection
- New connection from client

Privileged separation - network daemon



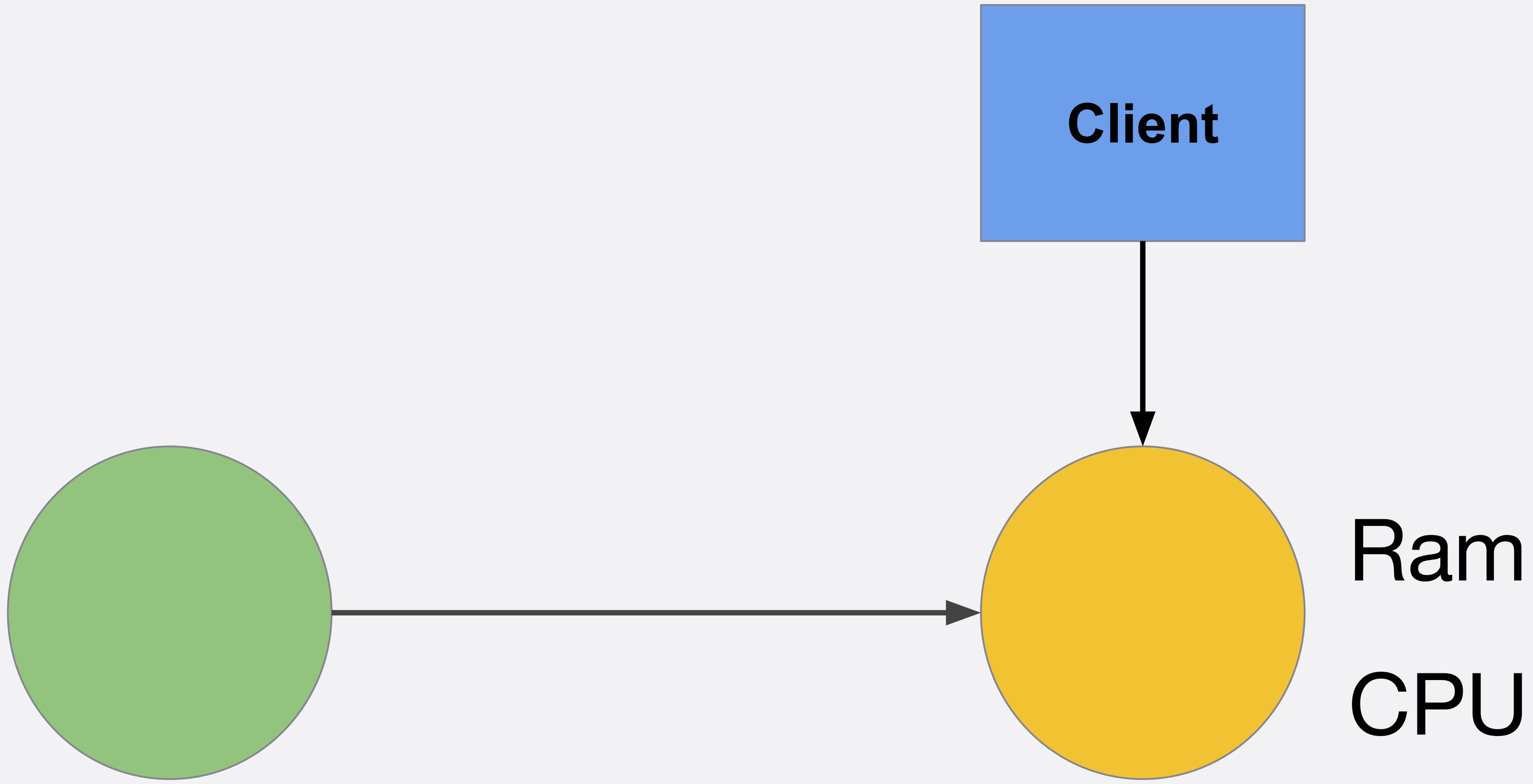
- Privileged process is waiting for connection
- New connection from client
- Fork and create unprivileged process

Privileged separation - network daemon



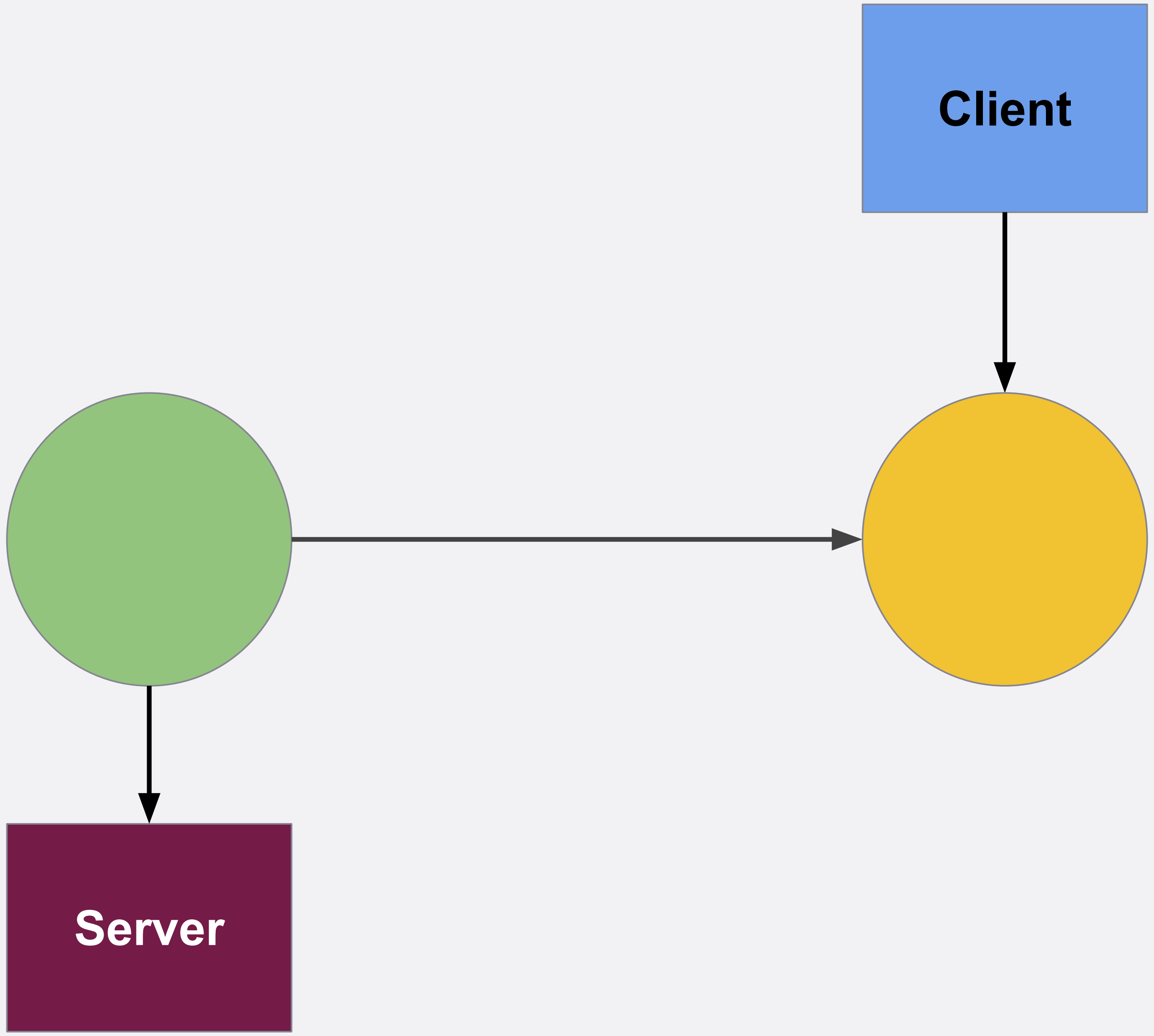
- Privileged process is waiting for connection
- New connection from client
- Fork and create unprivileged process
- Client is authenticating

Privileged separation - network daemon



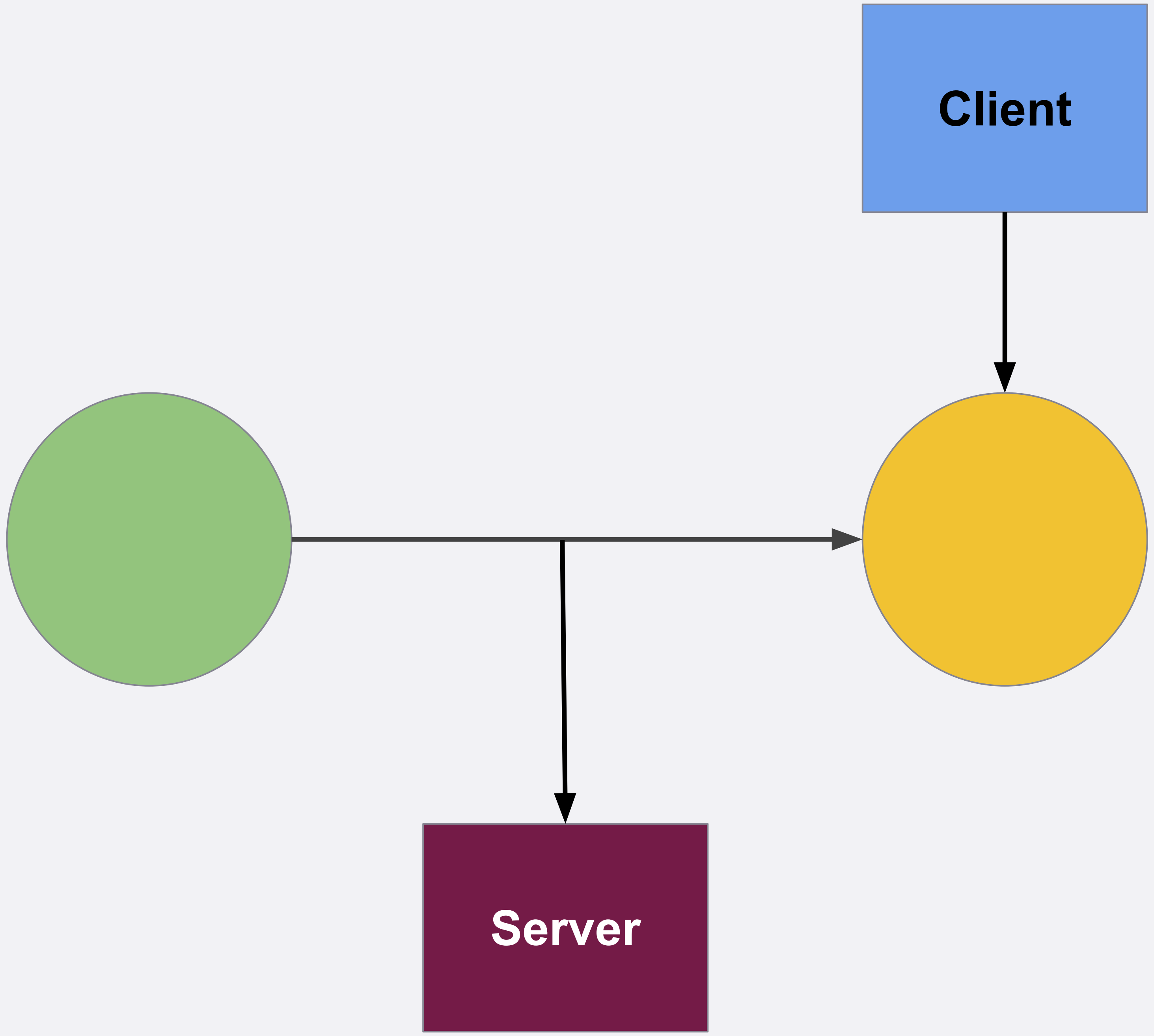
- New connection from client
- Fork and create unprivileged process
- Client is authenticating
- Privilegiat process is raising unprivileged process limits

Privileged separation - network daemon



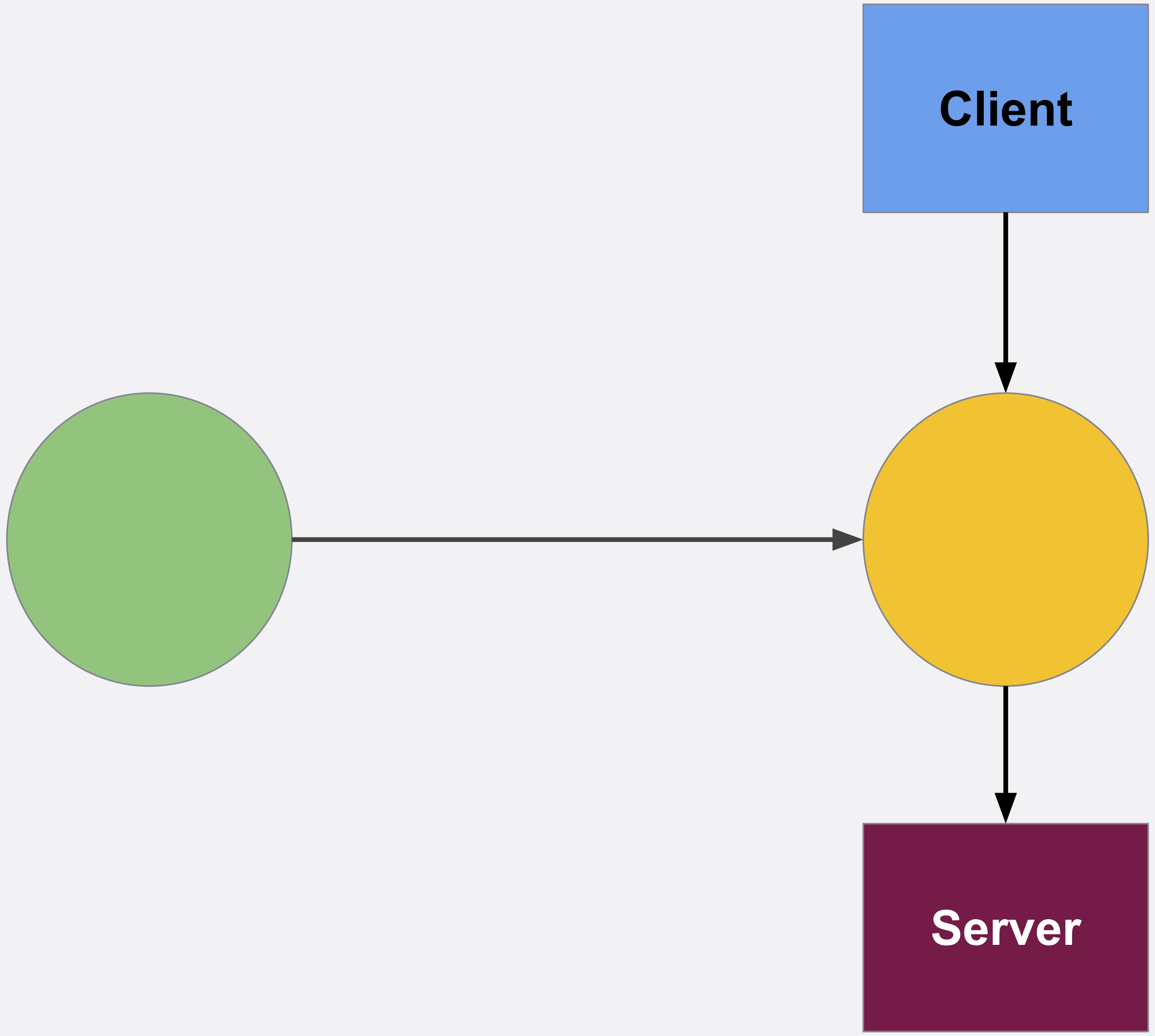
- Fork and create unprivileged process
- Client is authenticating
- Privilegiat process is raising unprivileged process limits
- Creating connection to the server

Privileged separation - network daemon



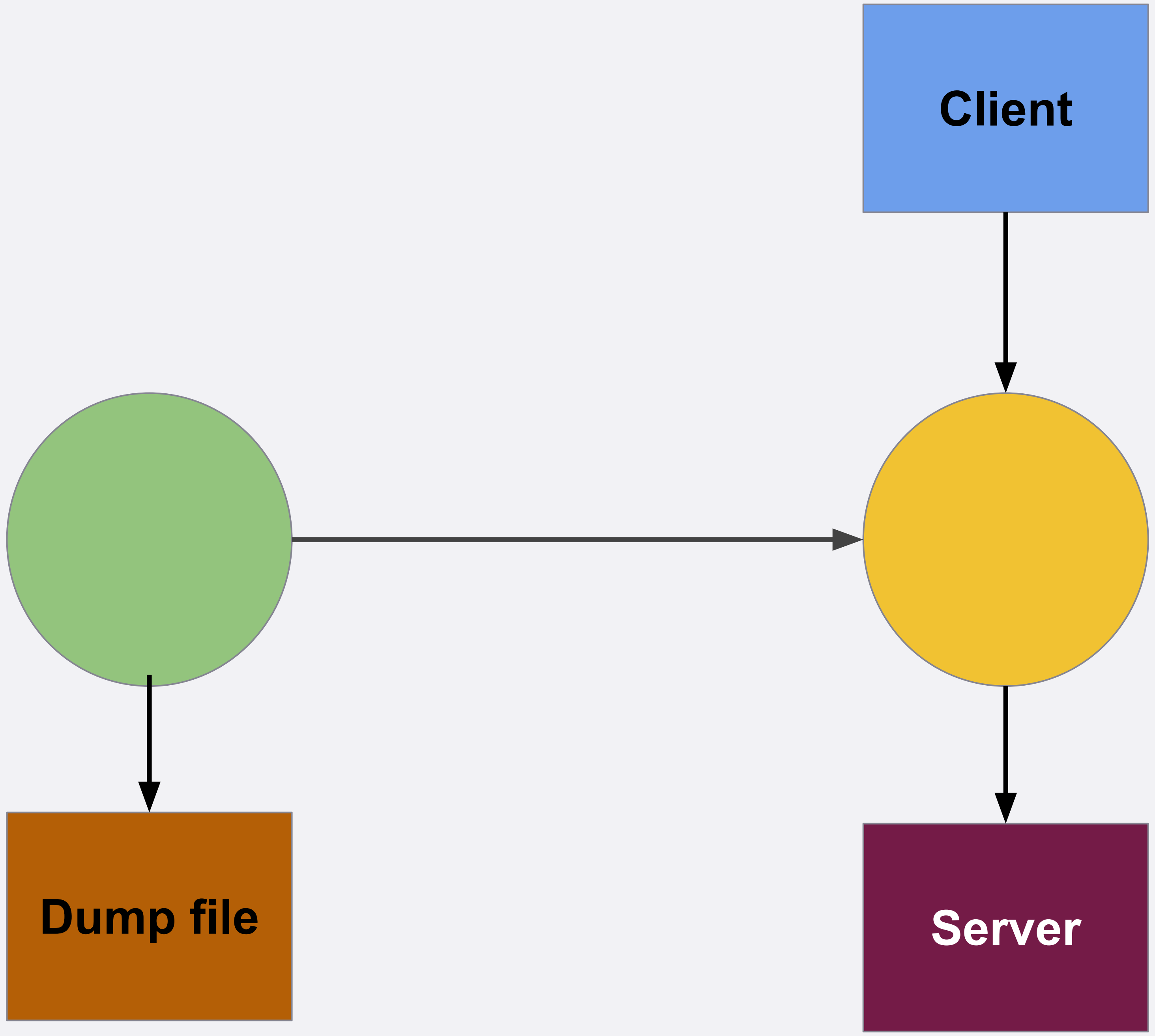
- Client is authenticating
- Privilegiat process is raising unprivileged process limits
- Creating connection to the server
- Pass connection to unprivileged process

Privileged separation - network daemon



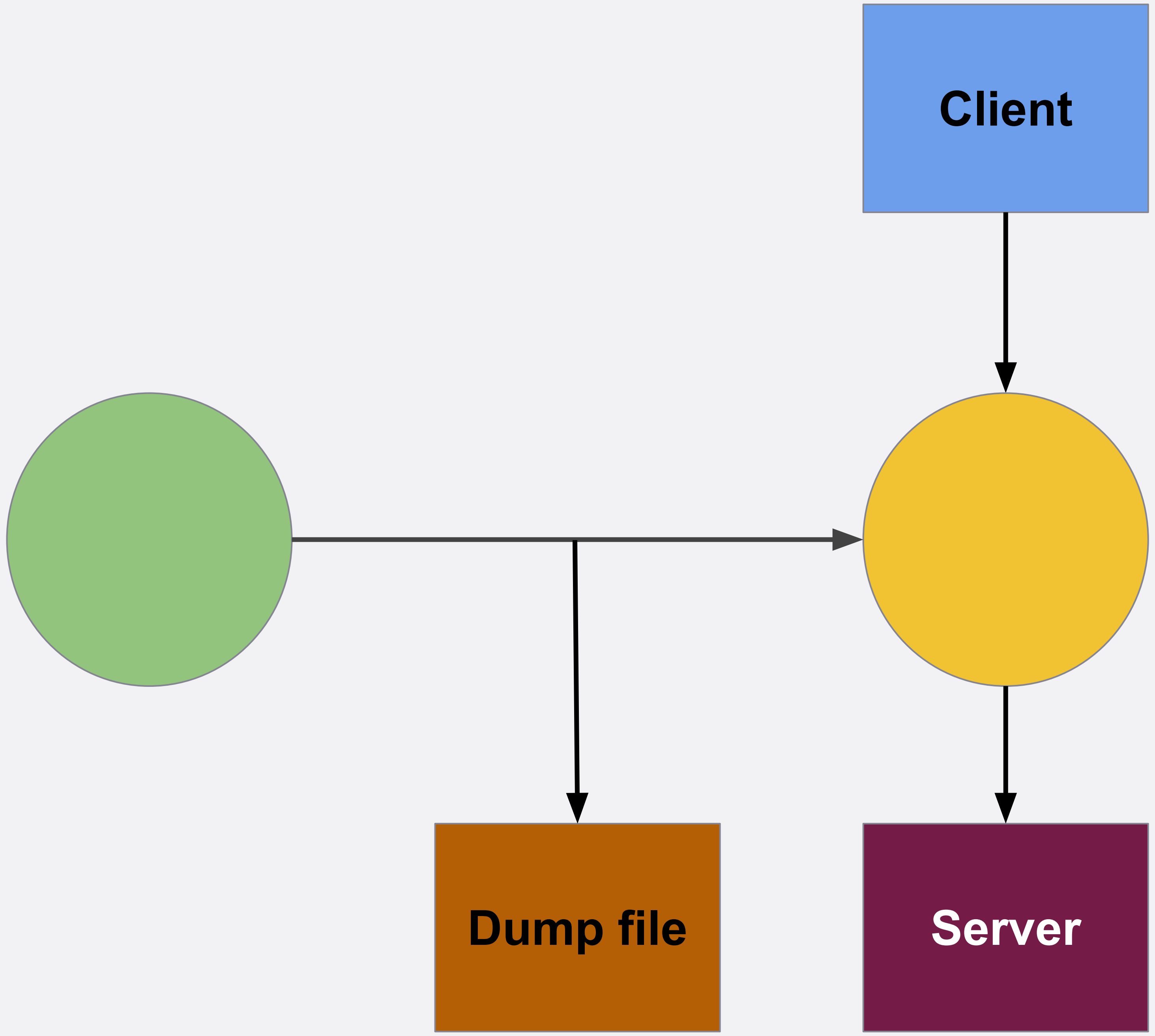
- Client is authenticating
- Privilegiat process is raising unprivileged process limits
- Creating connection to the server
- Pass connection to unprivileged process

Privileged separation - network daemon



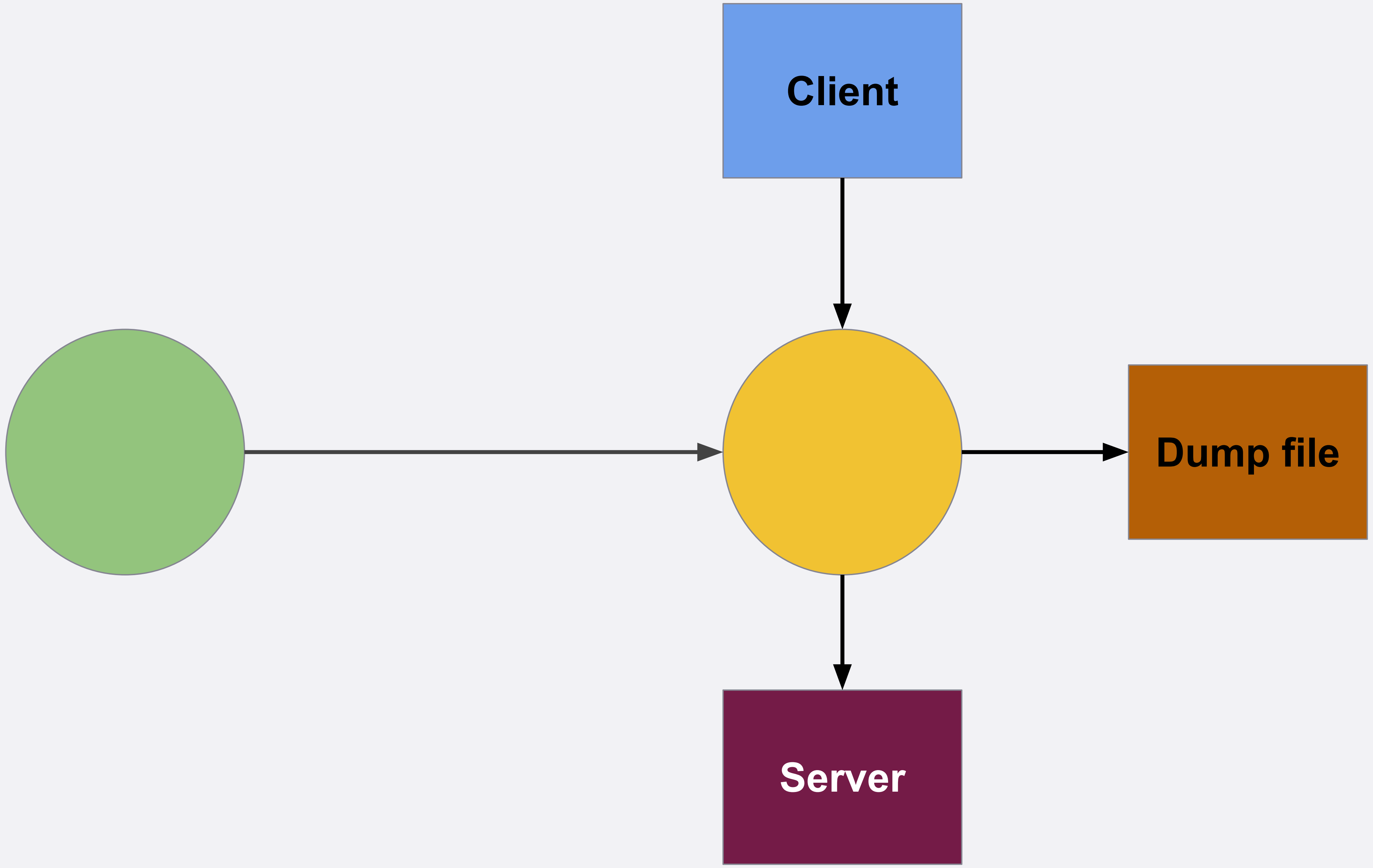
- Privilegiat process is raising unprivileged process limits
- Creating connection to the server
- Pass connection to unprivileged process
- Create a dump file

Privileged separation - network daemon



- Privilegiat process is raising unprivileged process limits
- Creating connection to the server
- Pass connection to unprivileged process
- Create a dump file
- Pass dump file

Privileged separation - network daemon



- Privilegiat process is raising unprivileged process limits
- Creating connection to the server
- Pass connection to unprivileged process
- Create a dump file
- Pass dump file

Other methods

- Jails
- CloudABI

nuxi

Thank you!

Mariusz Zaborski

✉ m.zaborski@fudosecurity.com

✉ oshogbo@FreeBSD.org

🌐 https://oshogbo.vexillium.org

@oshogbovx