Extending The Sleuth Kit and its Underlying Model for Pooled Storage File System Forensic Analysis

Fraunhofer Institute for Communication, Information Processing and Ergonomics

Jan-Niclas Hilgert* jan-niclas.hilgert@fkie.fraunhofer.de

Martin Lambertz martin.lambertz@fkie.fraunhofer.de

Daniel Plohmann daniel.plohmann@fkie.fraunhofer.de

Fraunhofer FKIE

Digital Forensic Analysis

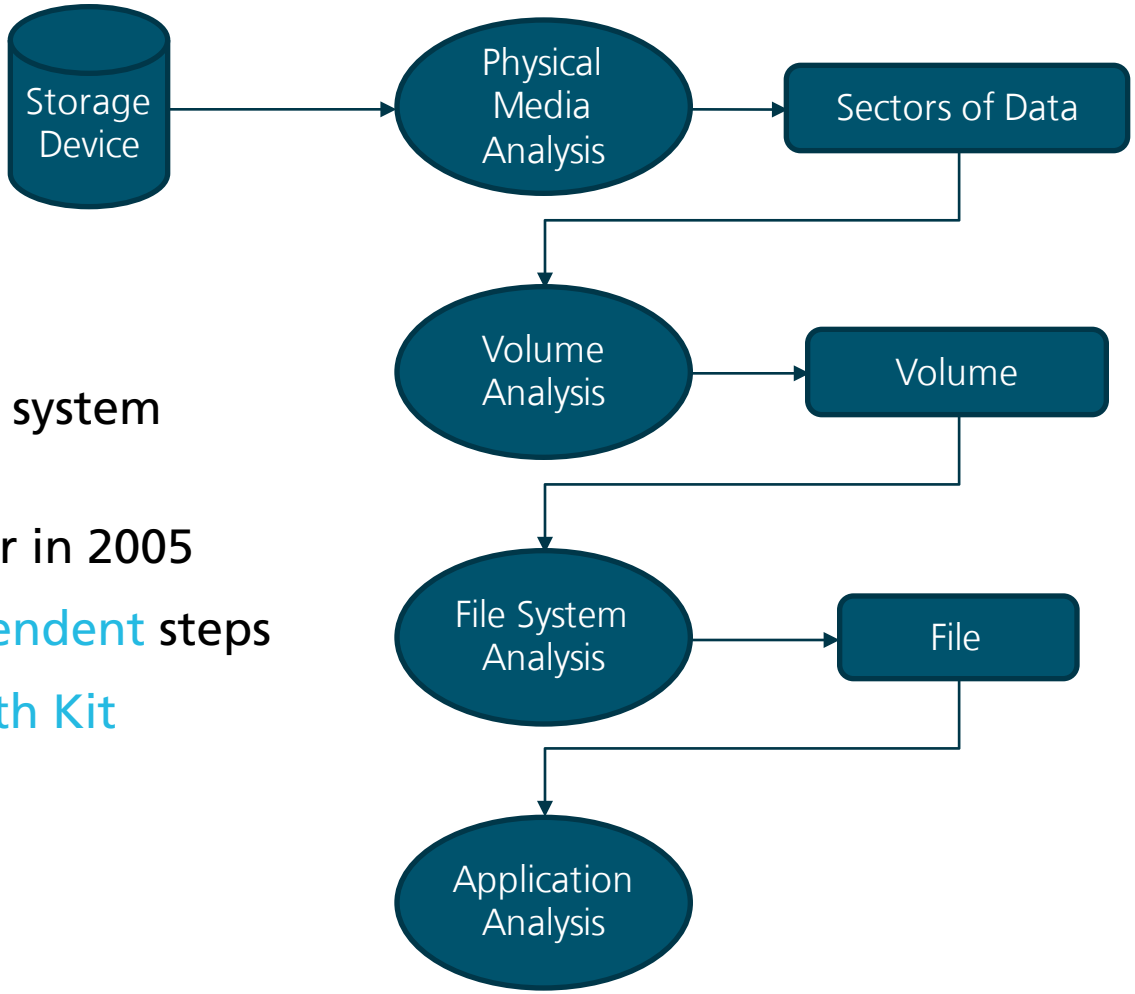
File Systems

- Define how data is read from and written to a storage device
- Utilize metadata to keep order of the data stored
- File systems differ in many aspects

- Extensive background knowledge is required for a forensic analysis
  - But not always existent

File System Forensic Analysis

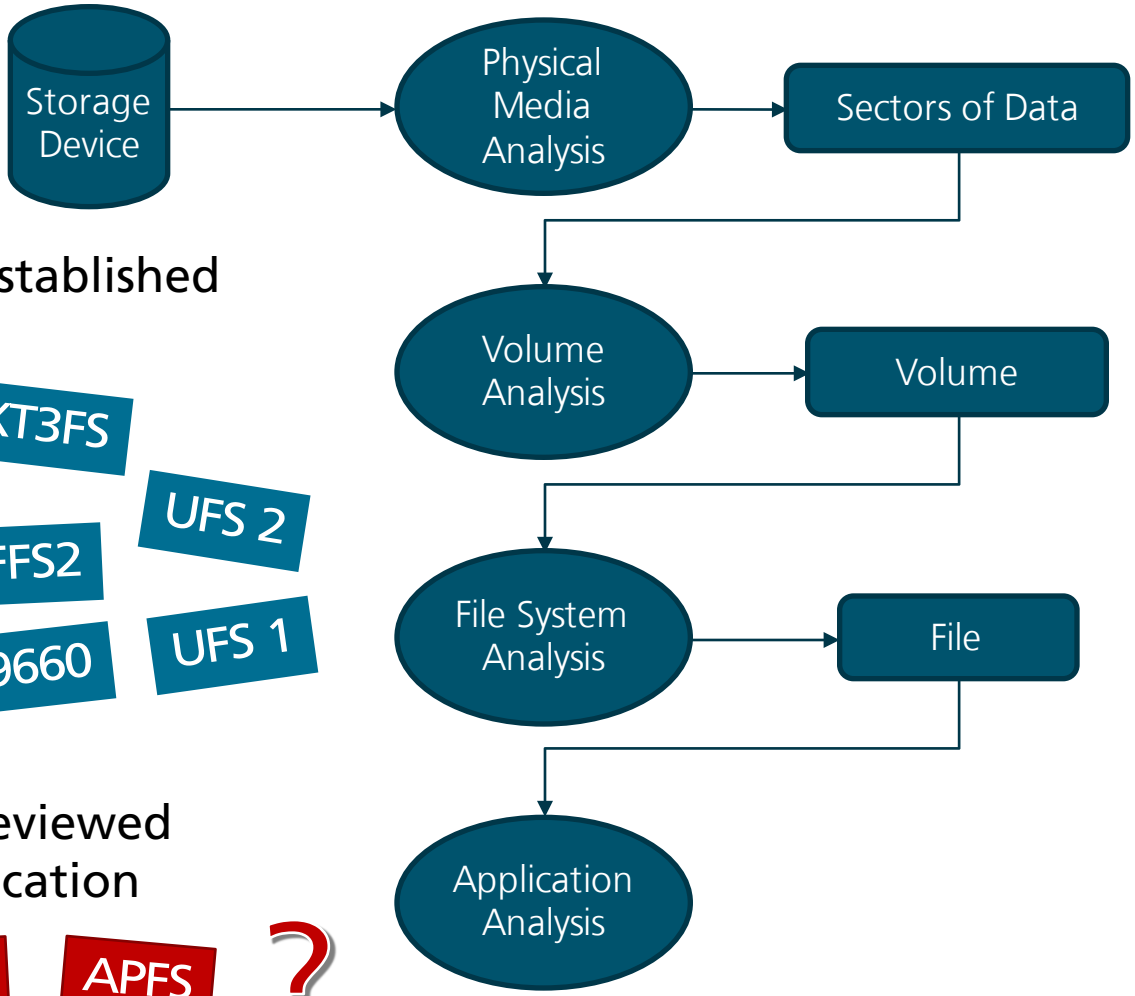
De-Facto Standard Model



- Universal model for a file system forensic analysis
- Presented by Brian Carrier in 2005
- Consists of four interdependent steps
- Implemented in The Sleuth Kit

File System Forensic Analysis

De-Facto Standard Model



- Works great on a lot of established file systems

EXT2FS
EXT3FS
ExFAT
UFS 2
NTFS
FAT
YAFFS2
HFS
UFS 1
Ext4
ISO 9660

- **Problem:** It hasn't been reviewed or changed since its publication

ZFS
BTRFS
ReFS
APFS
?

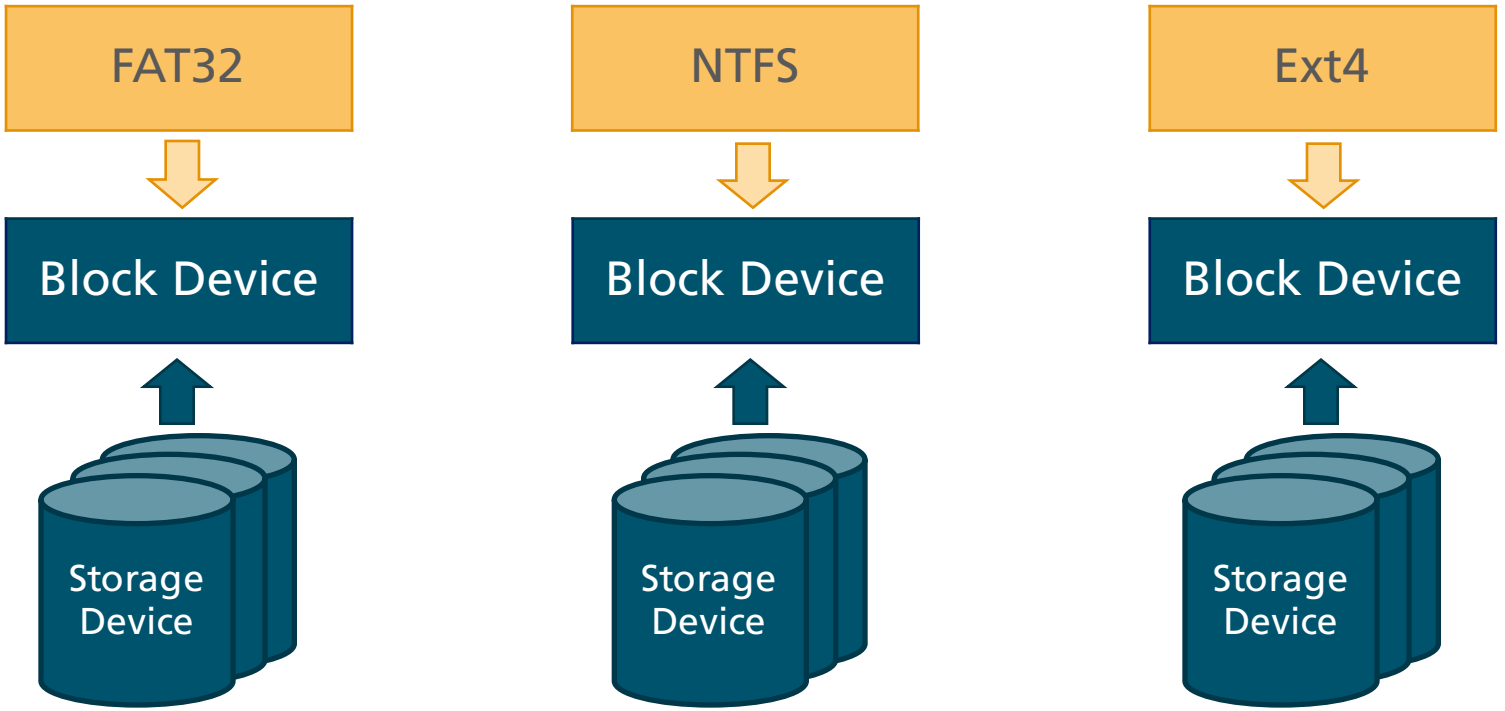
File System Forensic Analysis

Trying TSK on ZFS

```
jhilgert:ZPool$ fsstat disk1
Cannot determine file system type
```

Pooled Storage File Systems

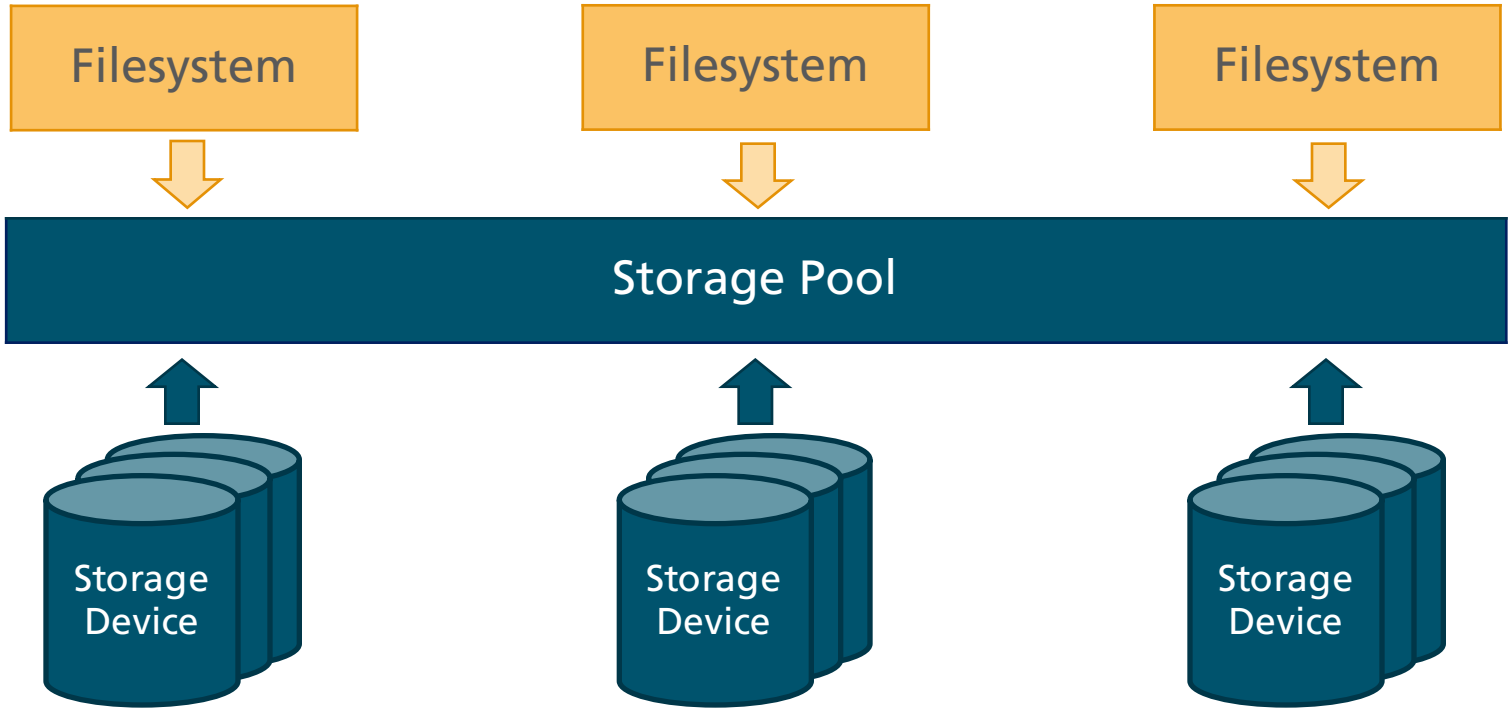
Old File System Mapping



- Storage devices (hard drives, solid state drives ...) are somehow used to create new block devices
- One file system is assigned to one of these block devices

Pooled Storage File Systems

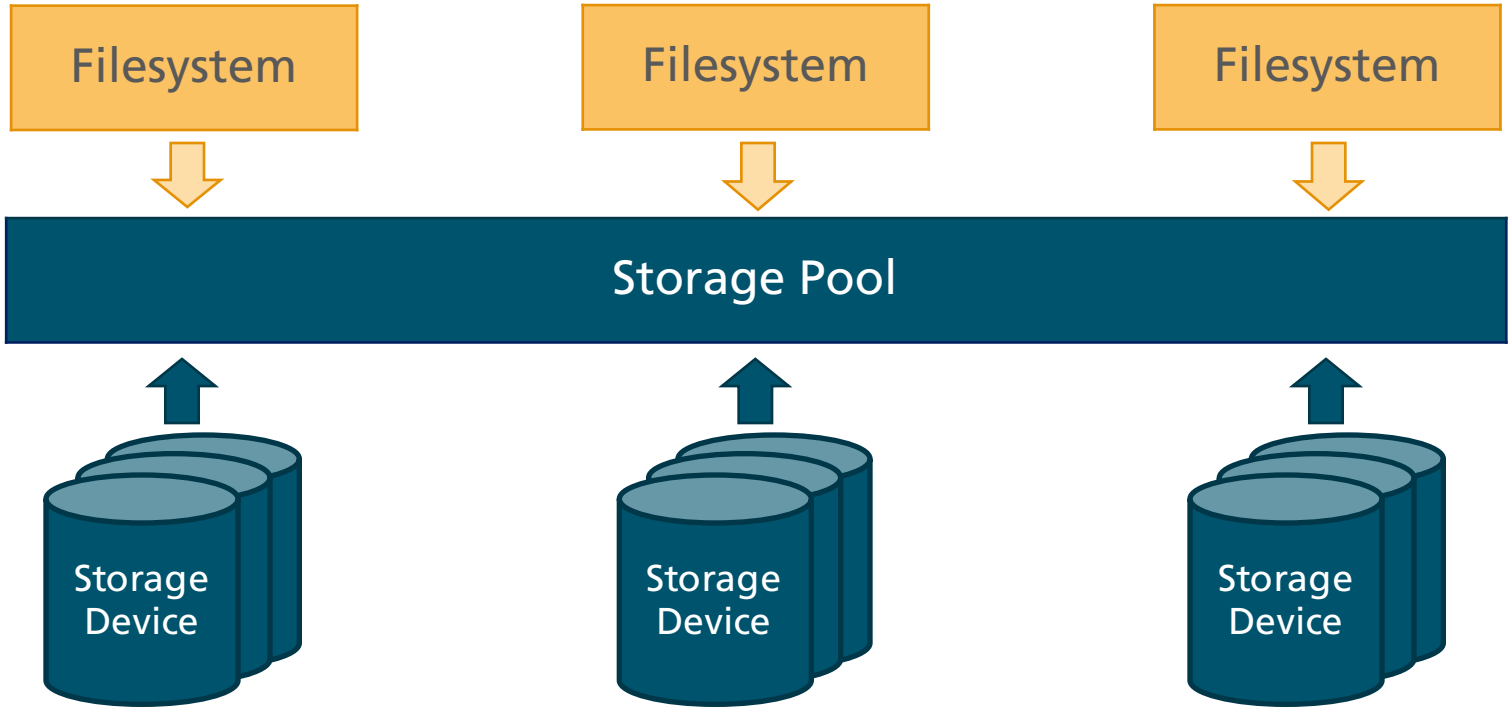
New Approach



- Storage Devices are combined to form a storage pool
- Its configuration depends completely on the implementation
- File systems share the available space from the storage pool

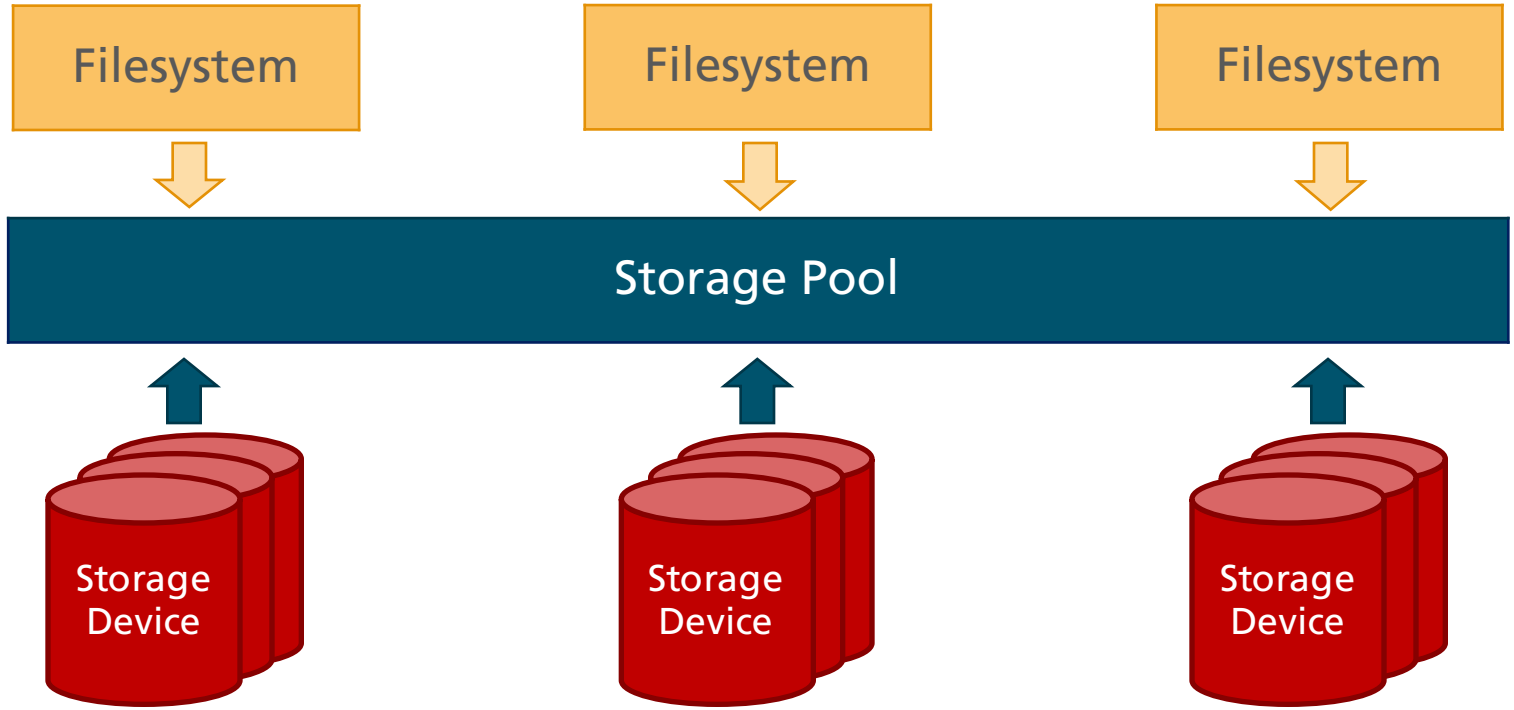
Extension of the Standard Model

Requirements – Detect Pool Members



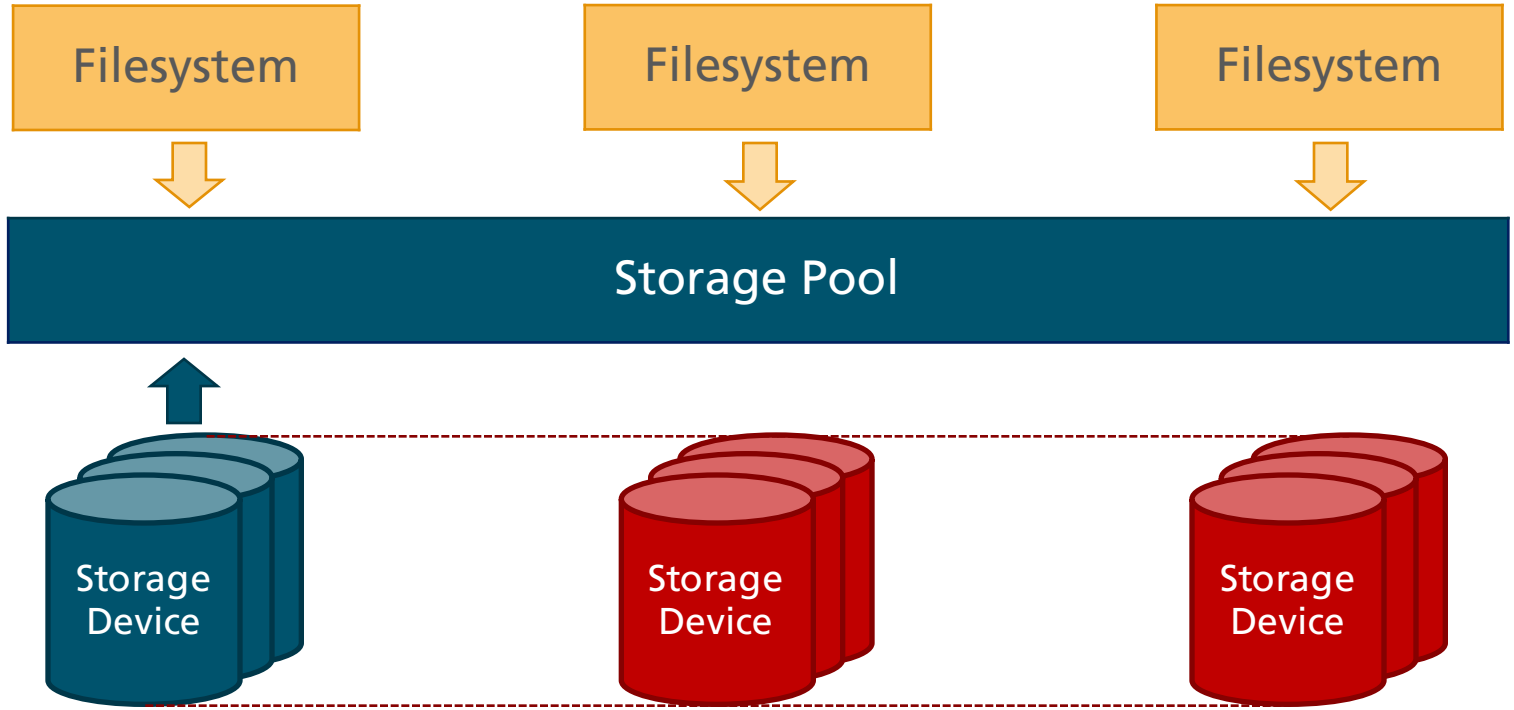
Extension of the Standard Model

Requirements – Detect Pool Members



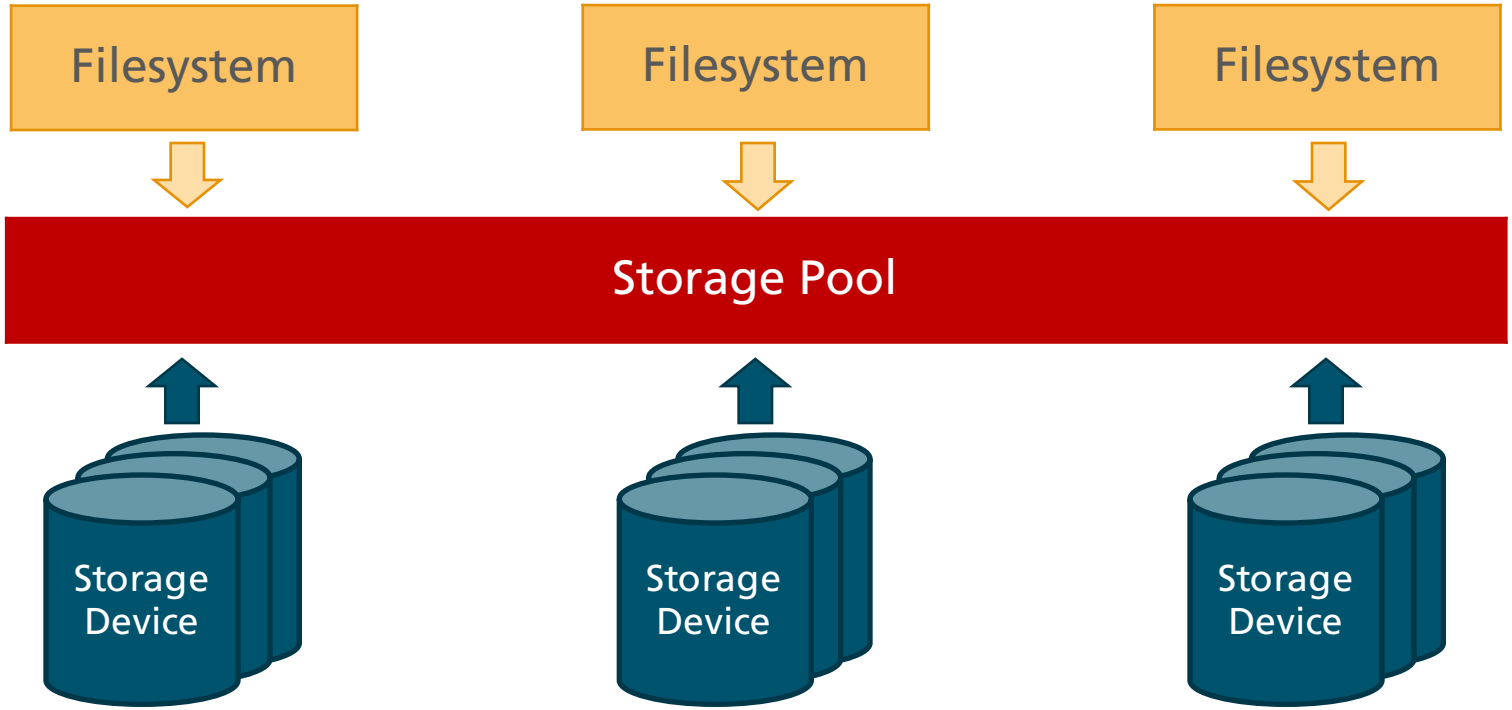
Extension of the Standard Model

Requirements – Detect Pool Configurations



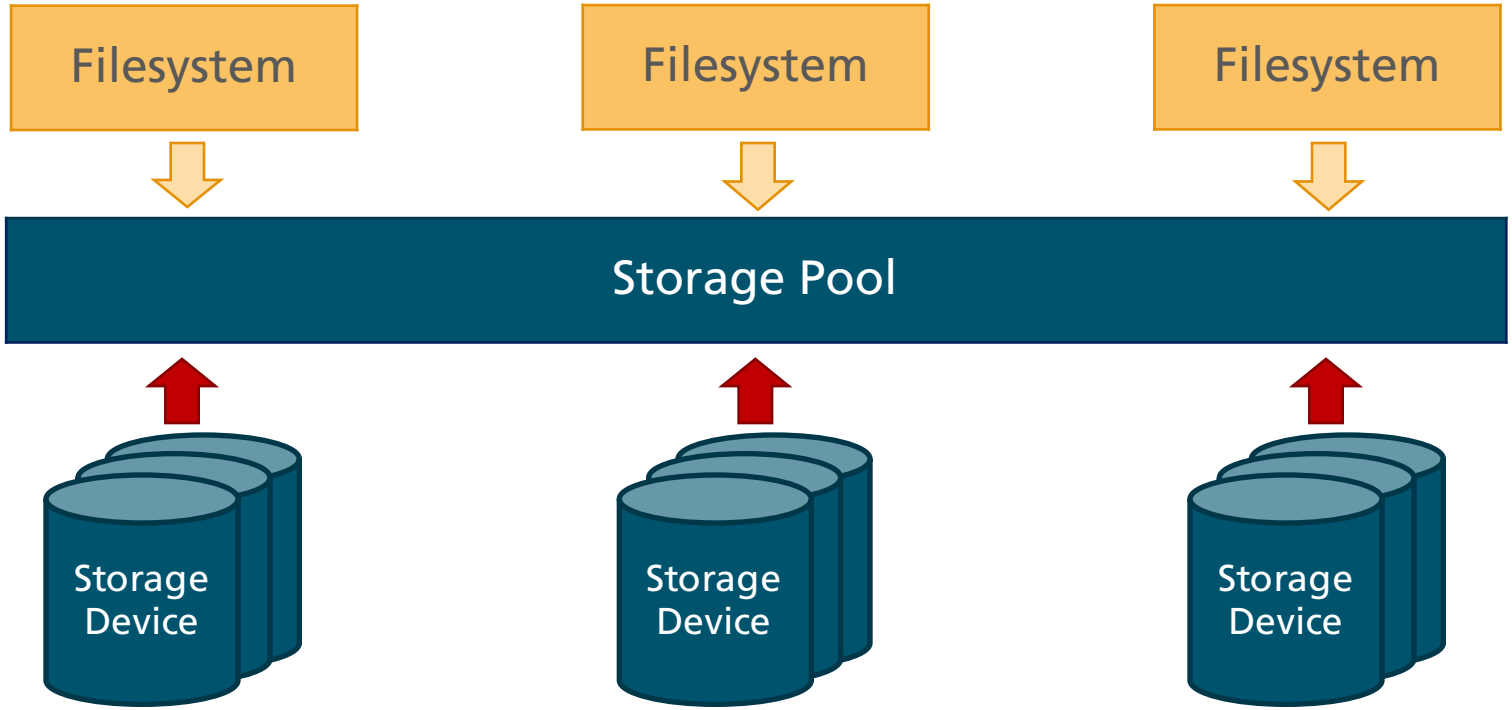
Extension of the Standard Model

Requirements – Analyze a Complete Pool



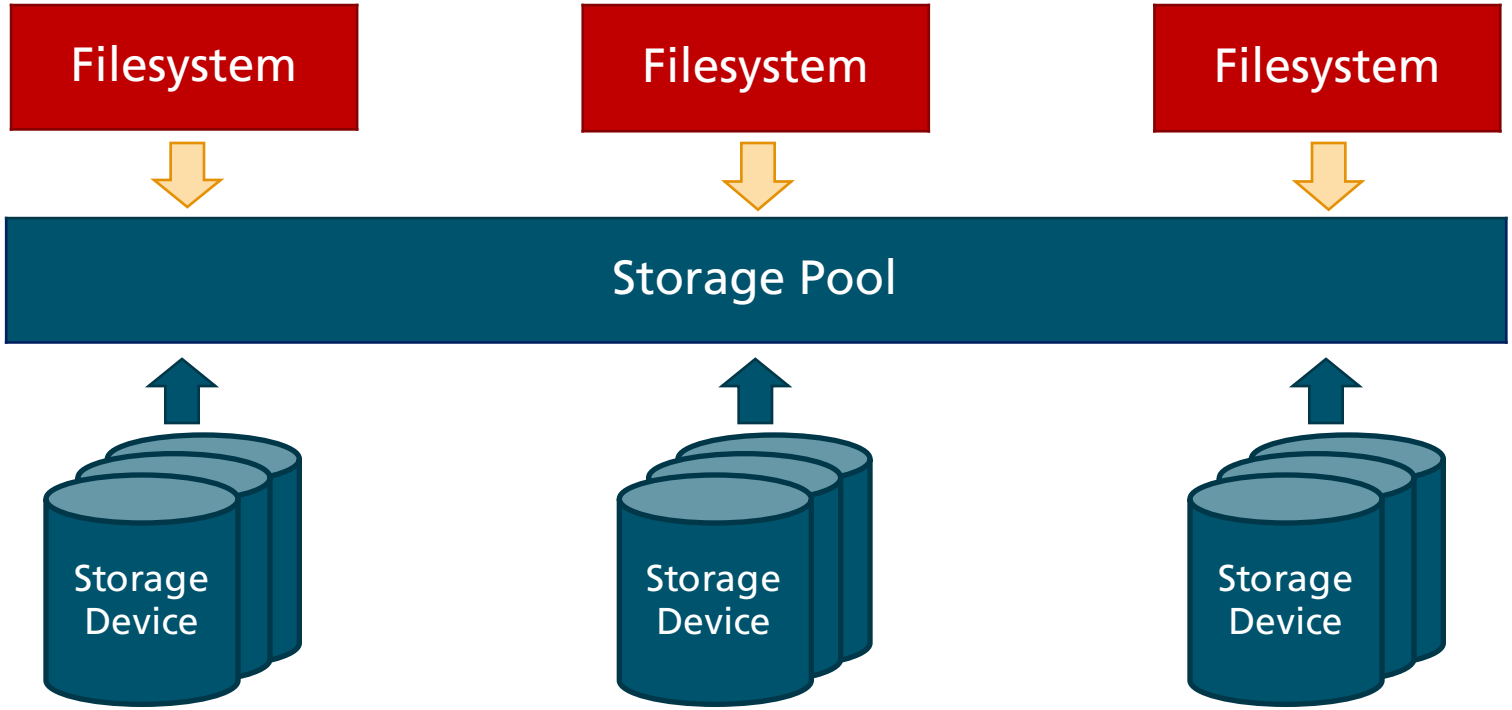
Extension of the Standard Model

Requirements – Access Correct Offsets on Physical Disks



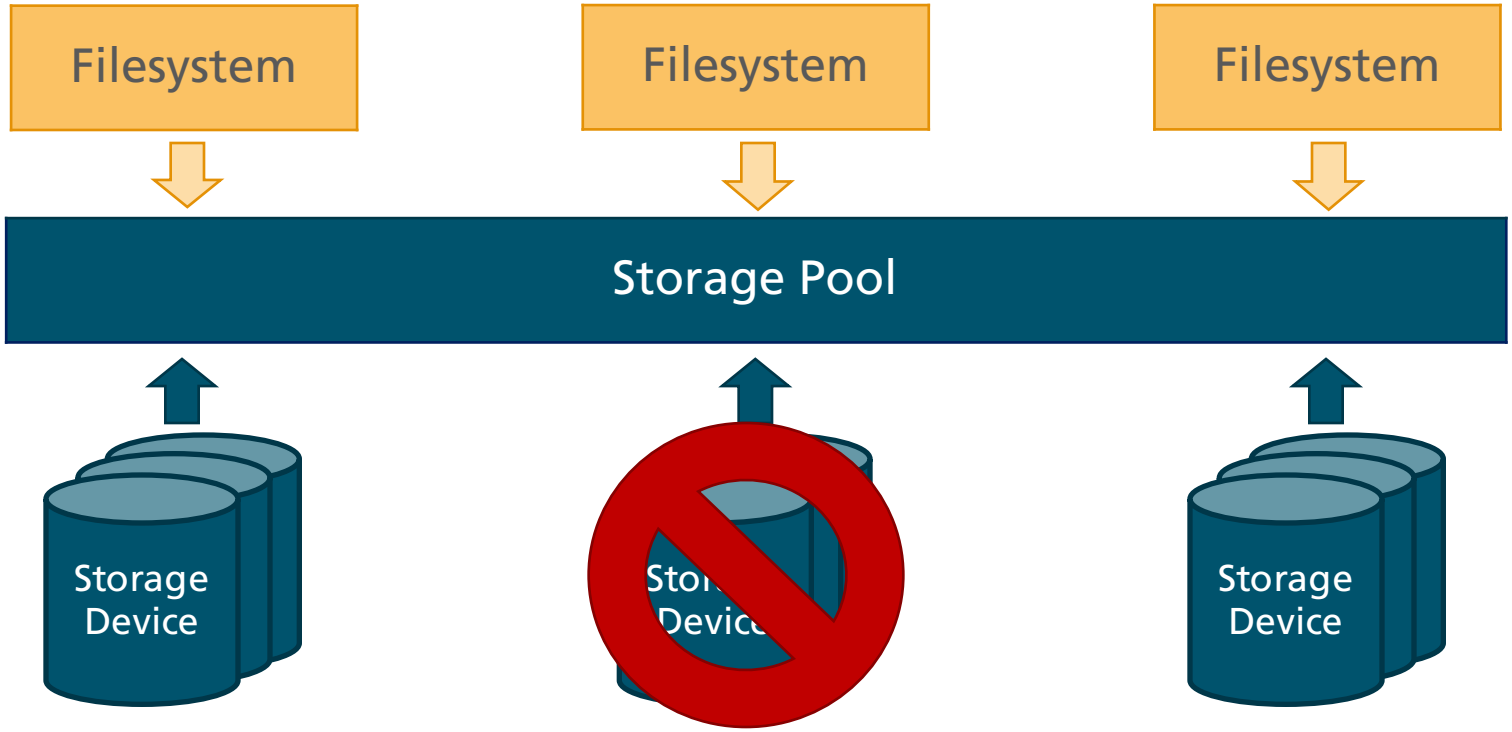
Extension of the Standard Model

Requirements – Access All Filesystem Data



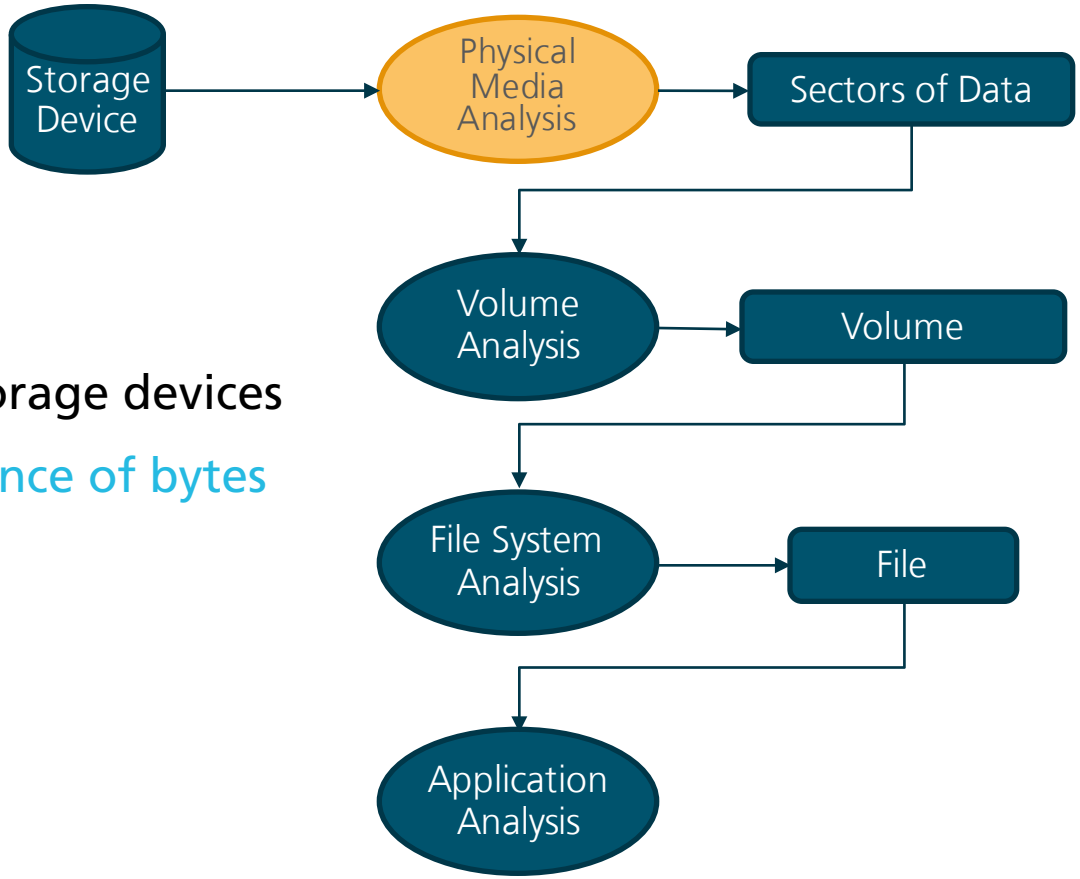
Extension of the Standard Model

Requirements – Deal with Incomplete Pools



File System Forensic Analysis

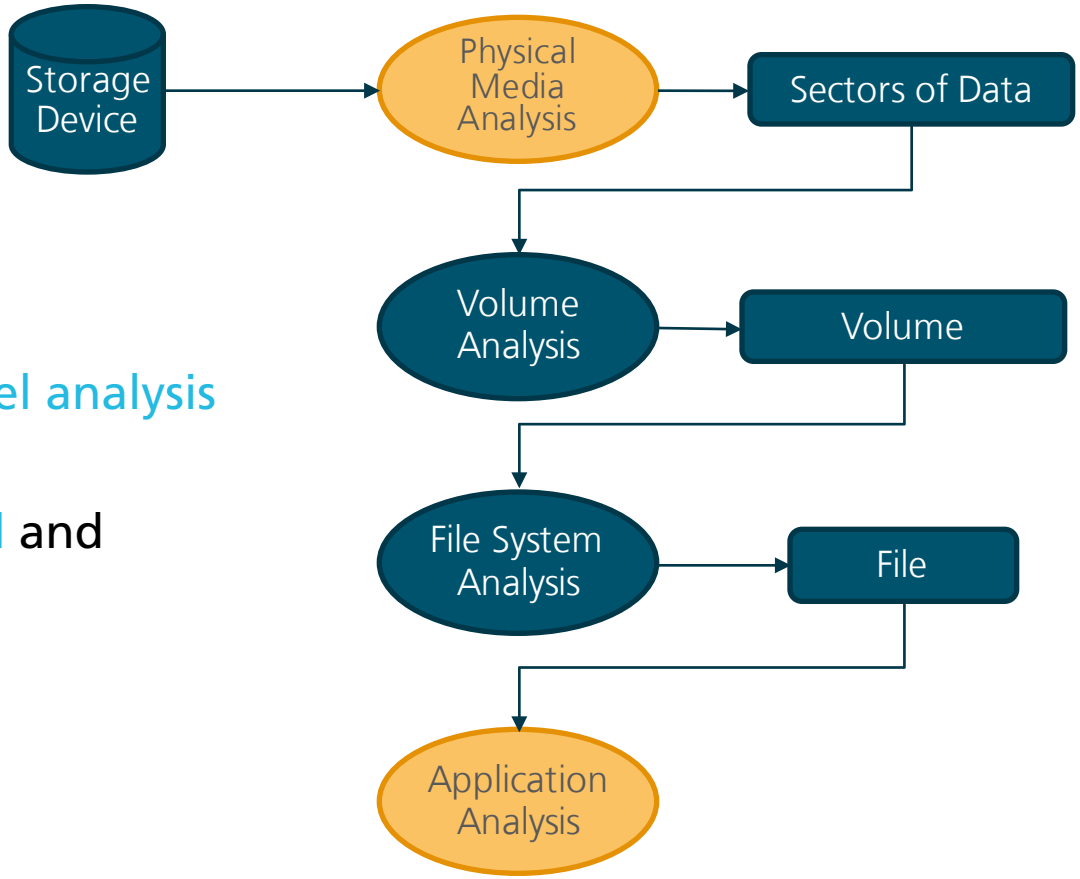
Phyiscal Media Analysis



- Acquisition of data from storage devices
- Data is only seen as a sequence of bytes
  - No change necessary

File System Forensic Analysis

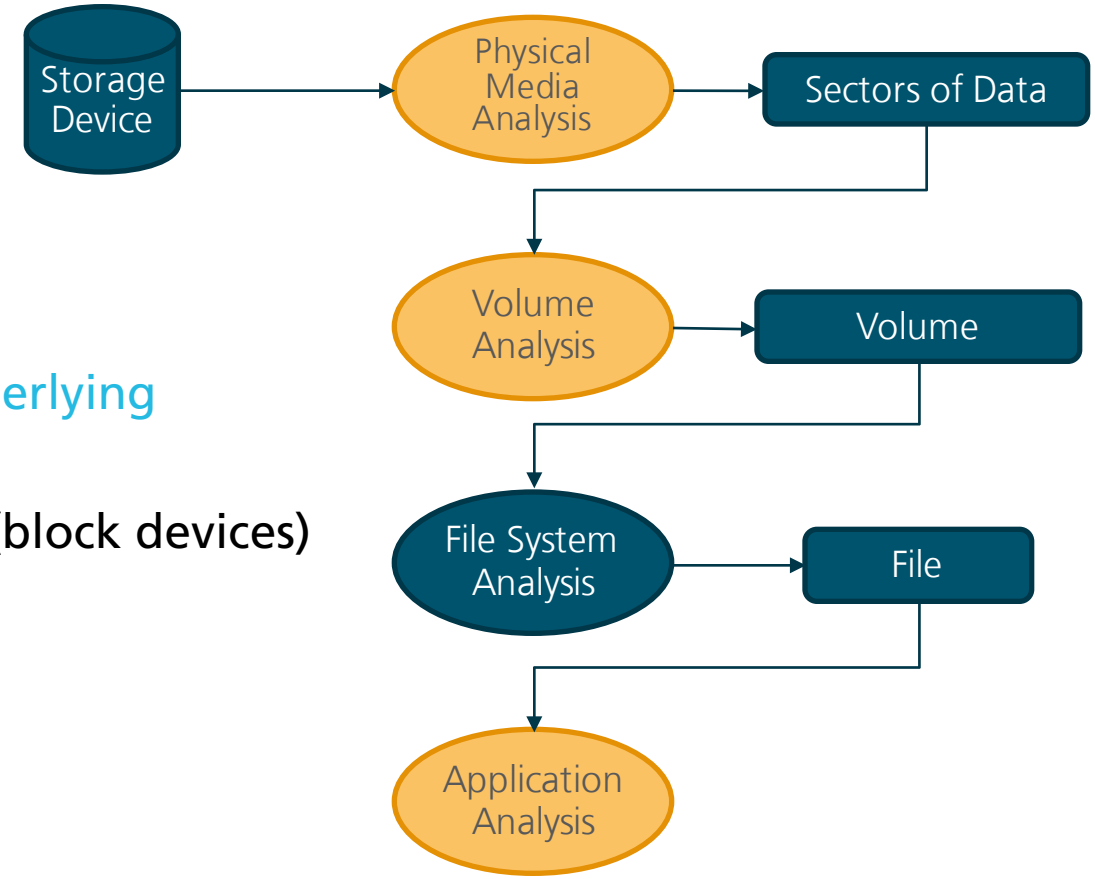
Application Analysis



- Performs an application-level analysis of the acquired files
- Works on already recovered and collected files
  - No change necessary

Extension of the Standard Model

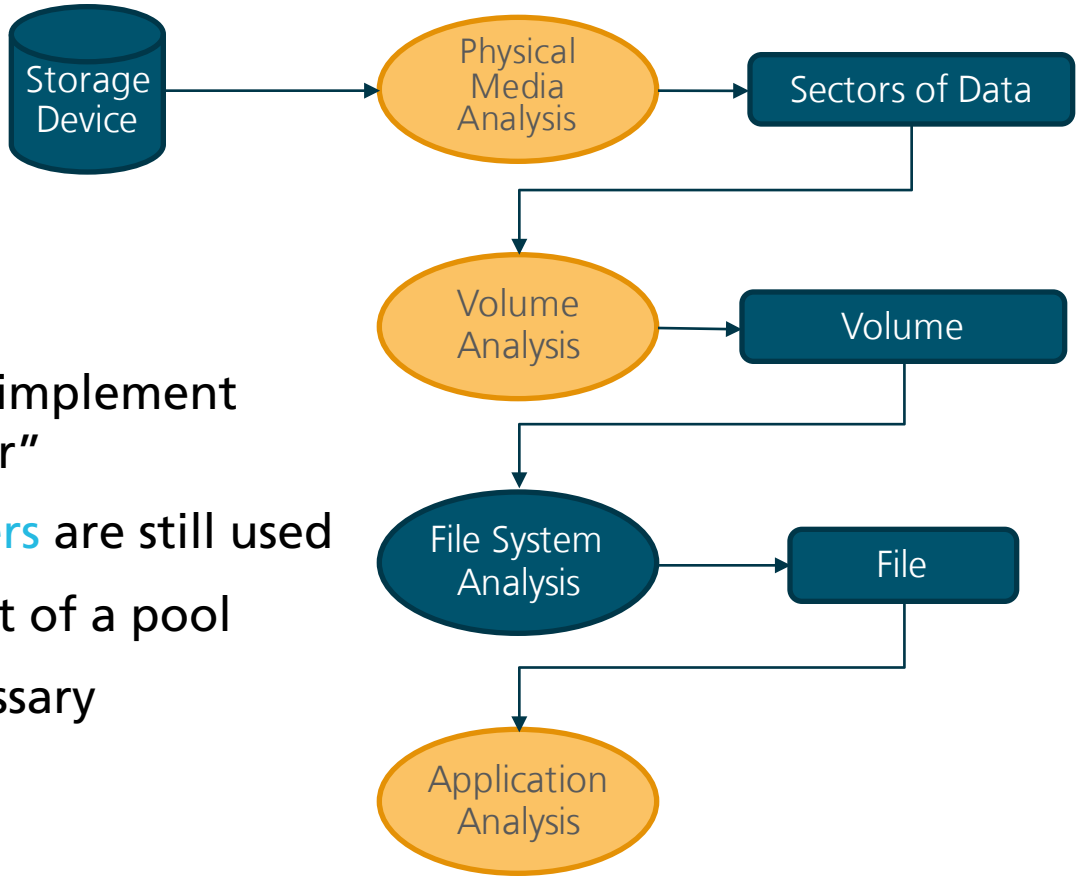
Volume Analysis



- Data is searched for its underlying volume structure
- Returns detected volumes (block devices)

Extension of the Standard Model

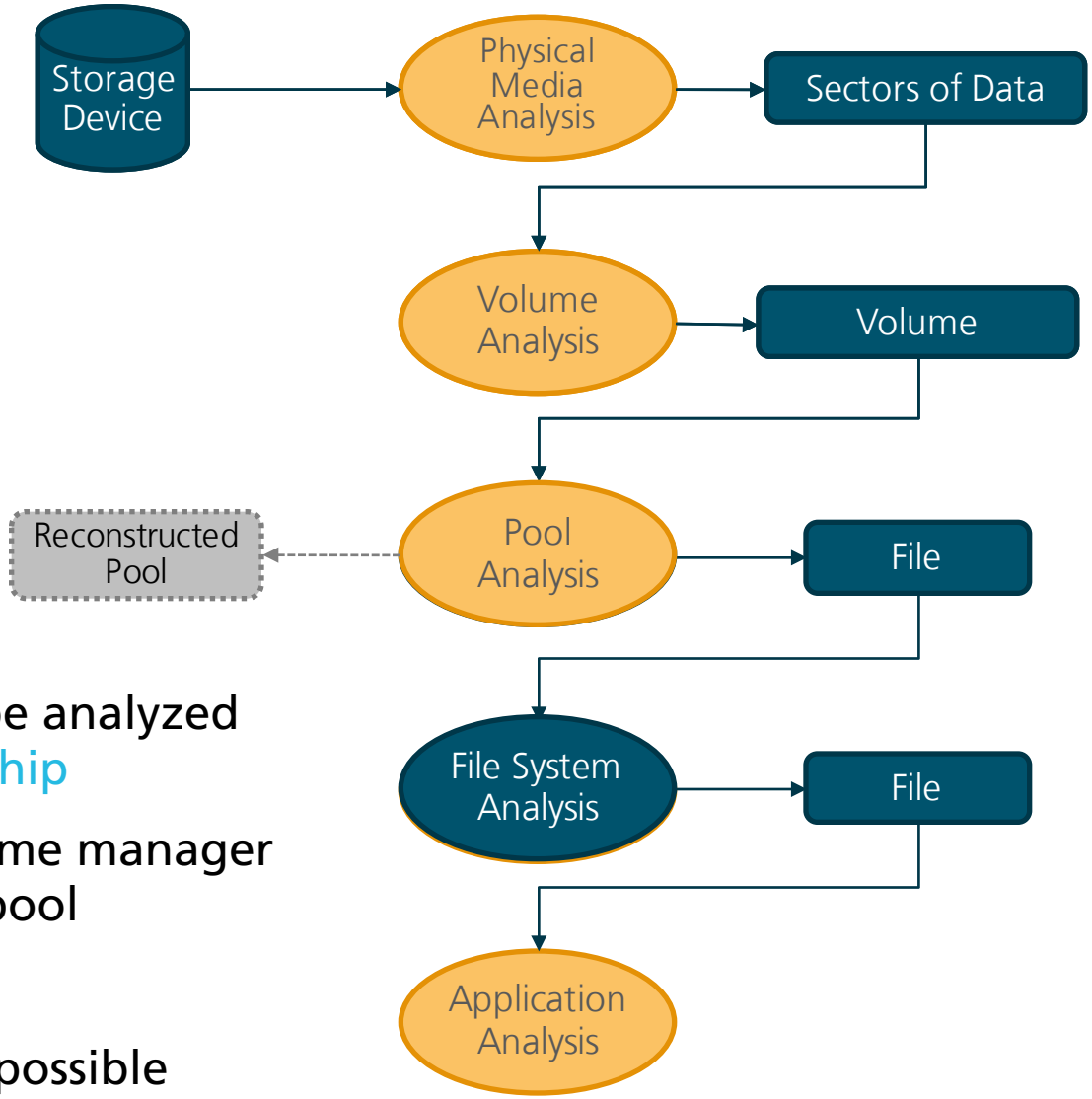
Volume Analysis



- Pooled storage file systems implement their own “volume manager”
- Established volume managers are still used
  - Volumes can also be part of a pool
- Volume analysis is still necessary

Extension of the Standard Model

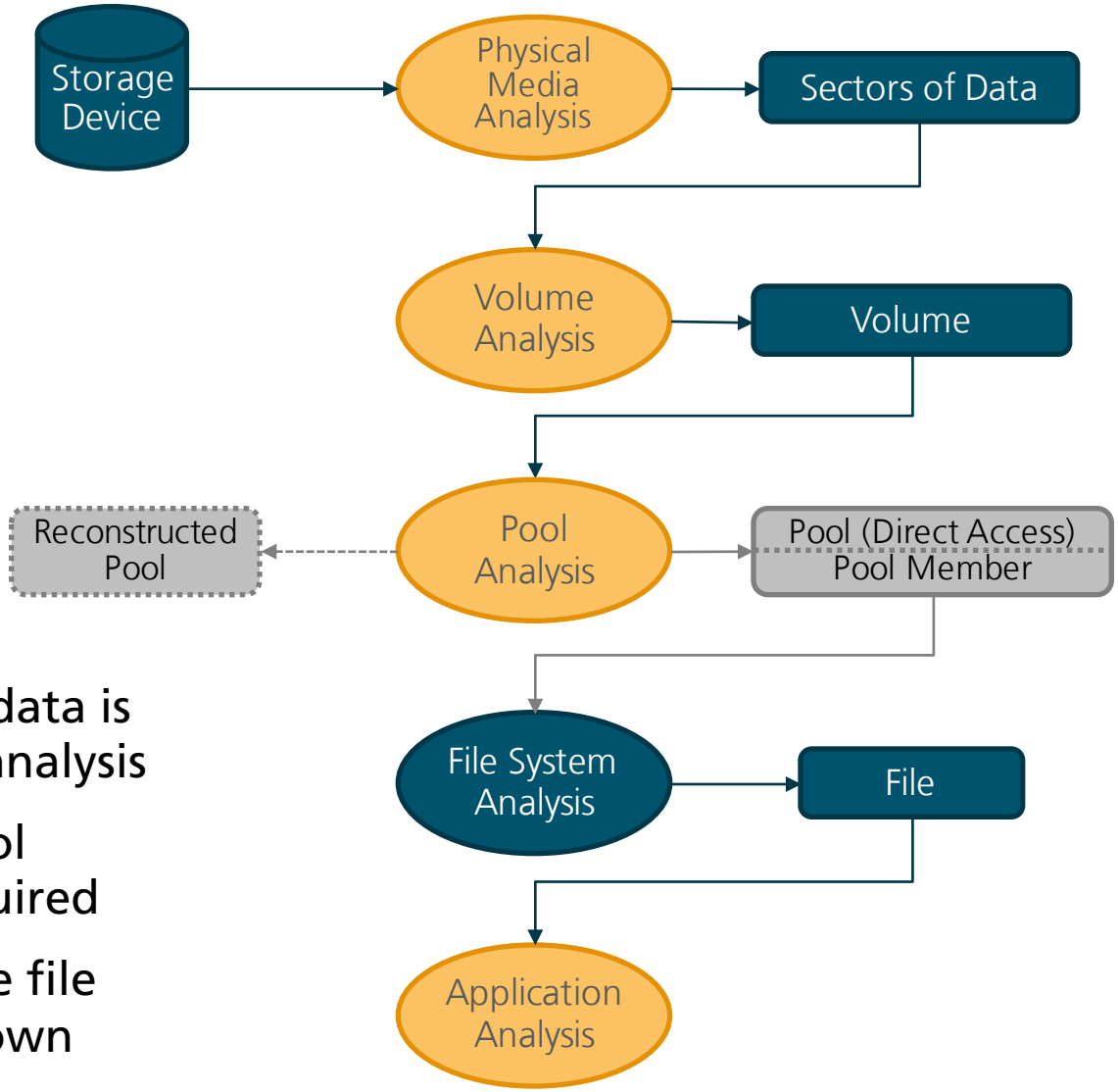
Pool Analysis



- Volumes or data need to be analyzed for possible pool membership
- Using the file systems volume manager results in a reconstructed pool
  - Only high-level access
  - No file system analysis possible

Extension of the Standard Model

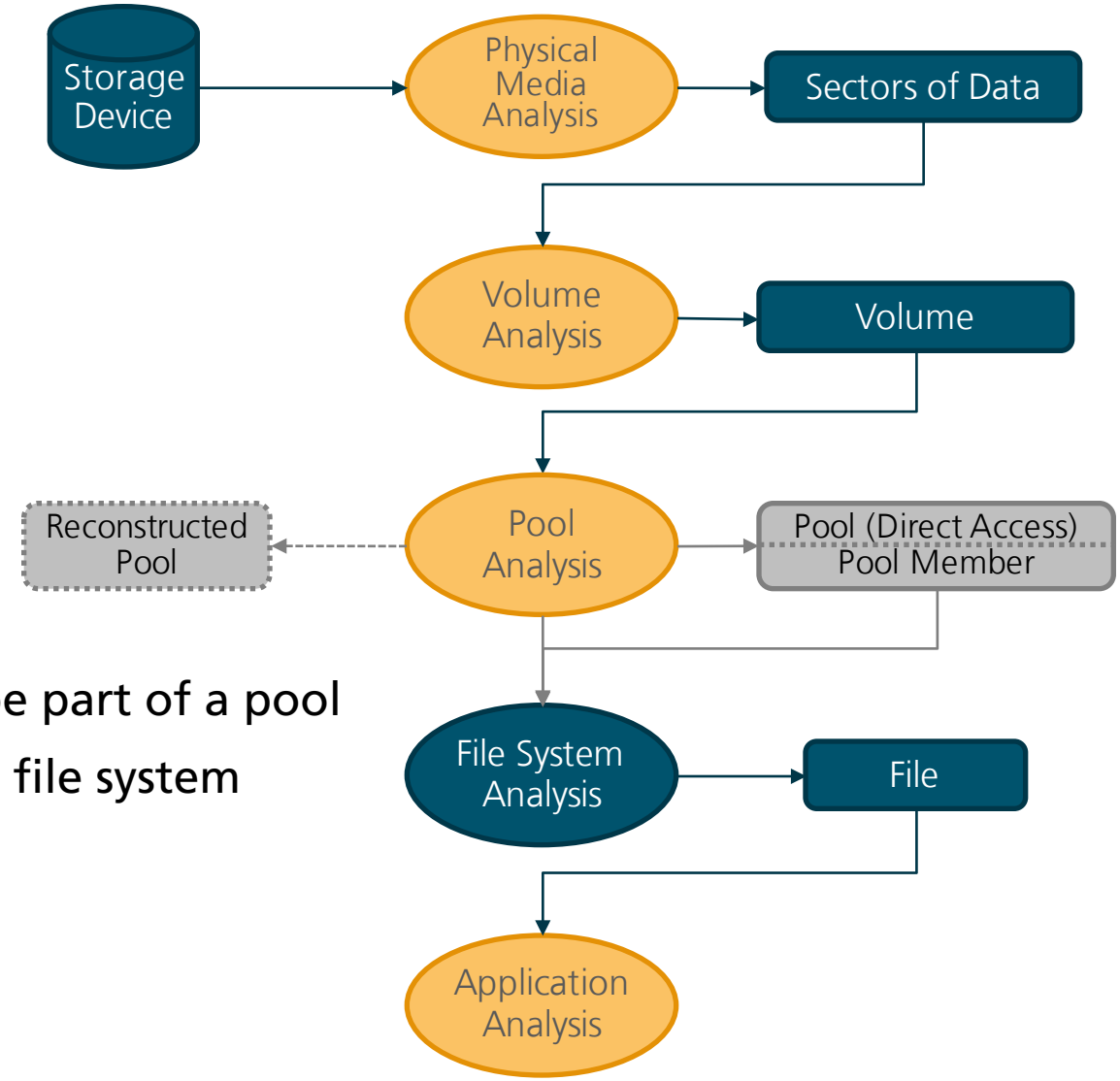
Pool Analysis



- File system data and metadata is essential for a file system analysis
  - Direct access to the pool and its members is required
  - Mapping schema of the file system needs to be known

Extension of the Standard Model

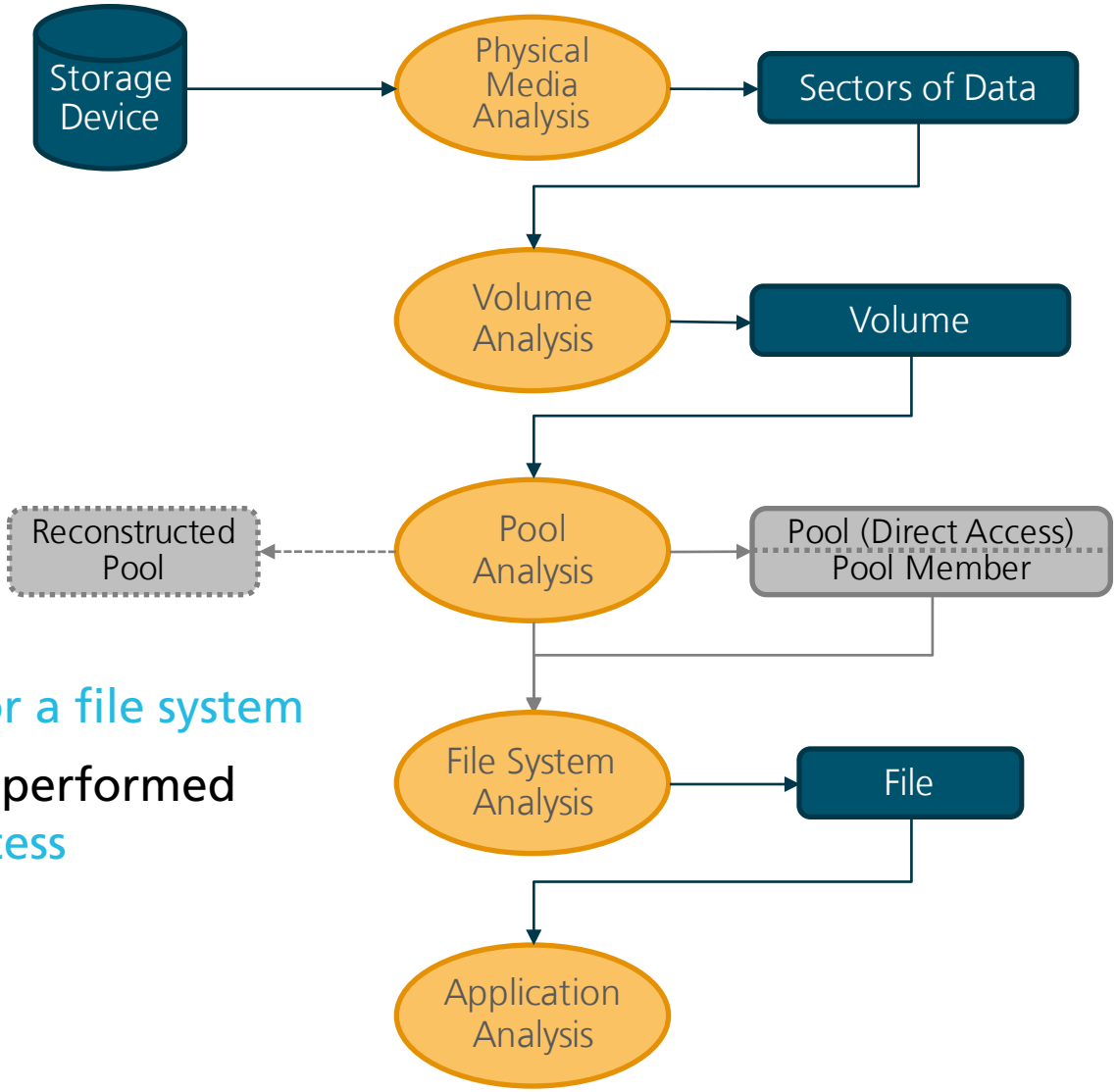
Pool Analysis



- Volumes do not need to be part of a pool
- **Important:** Pool analysis is file system dependent

Extension of the Standard Model

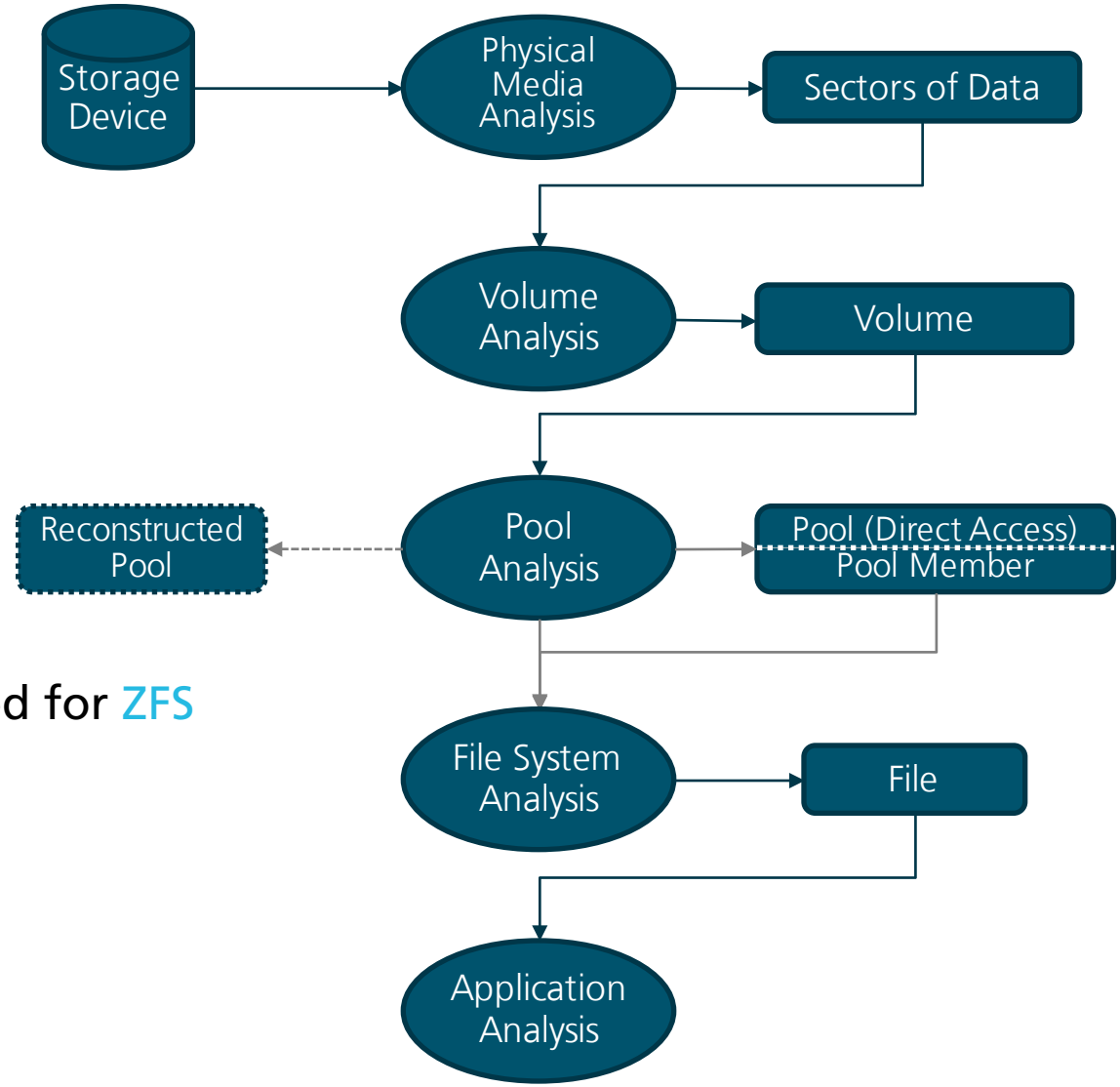
File System Analysis



- Each volume is searched for a file system
- File system analysis can be performed on the pool with direct access

Extension of the Standard Model

Extended Model



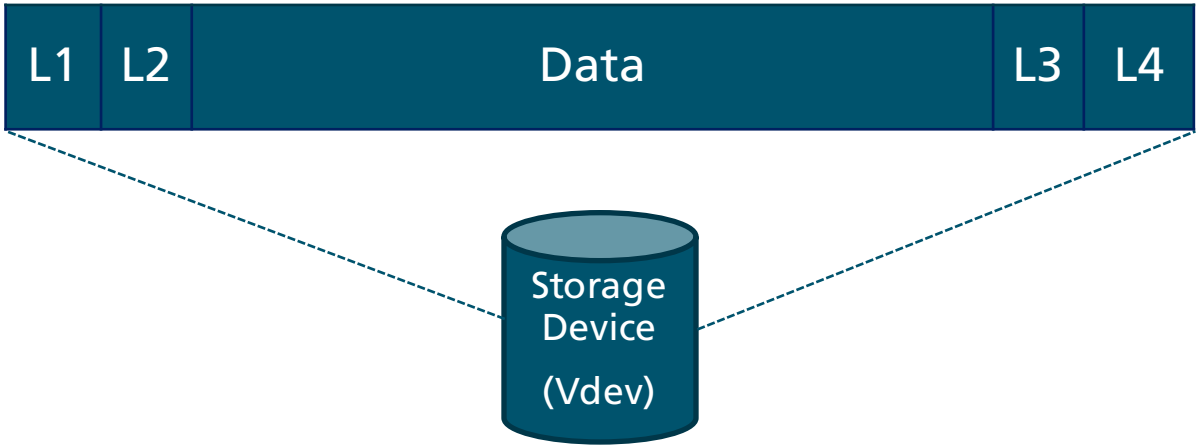
- Prototypically implemented for ZFS

Pooled Storage File Systems

ZFS – Volume Management

- First presented in 2003 (more than a decade ago)
- Available for major platforms like Solaris, FreeBSD, Linux, MacOS

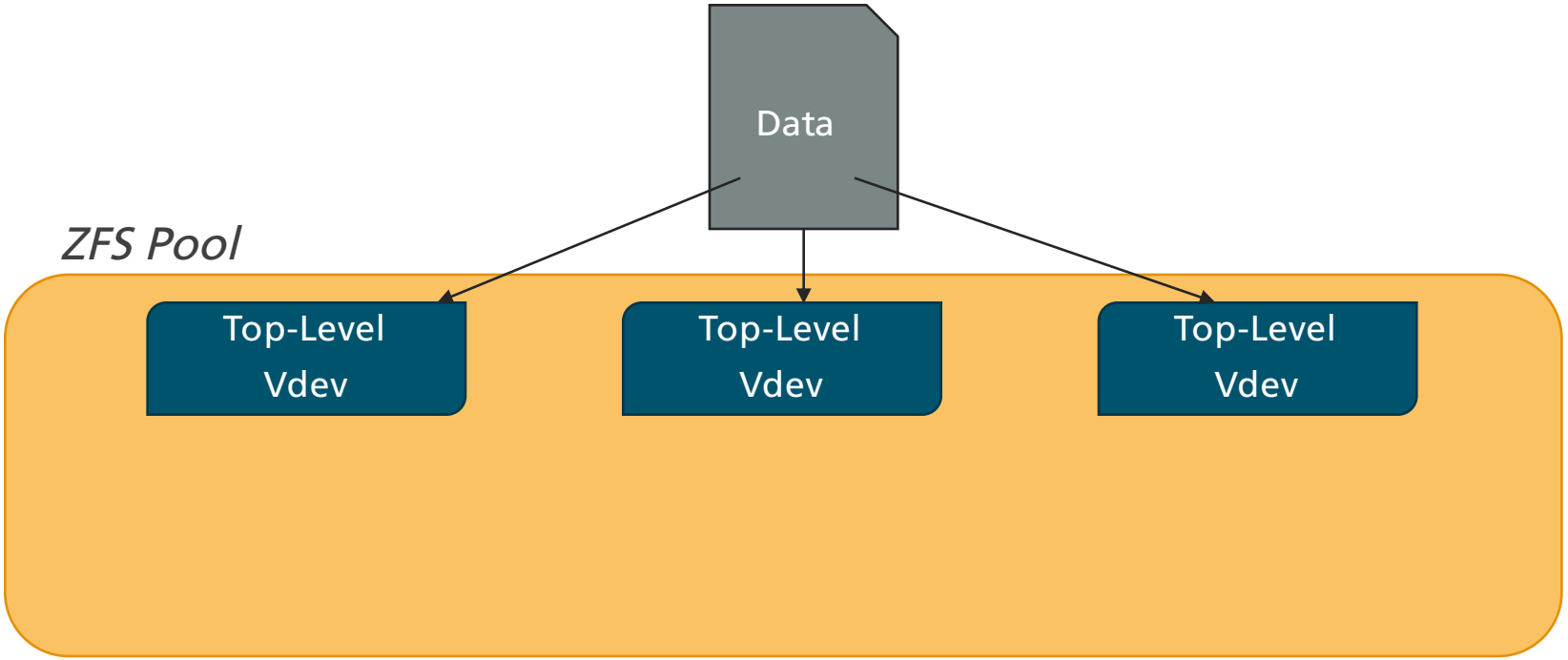
- ZFS combines multiple volumes into a storage pool
- Pool members are identified by four vdev-labels



Pooled Storage File Systems

ZFS – Volume Management

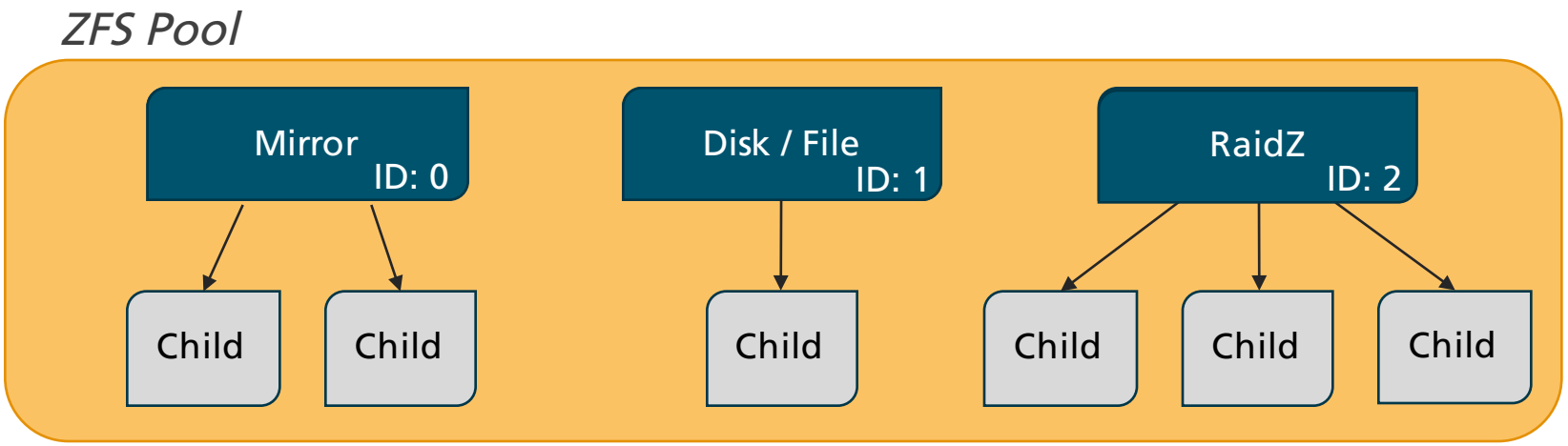
- Pools in ZFS consist of one or more top-level virtual devices (vdevs)
- Data is striped across all of these top-level vdevs



Pooled Storage File Systems

ZFS – Volume Management

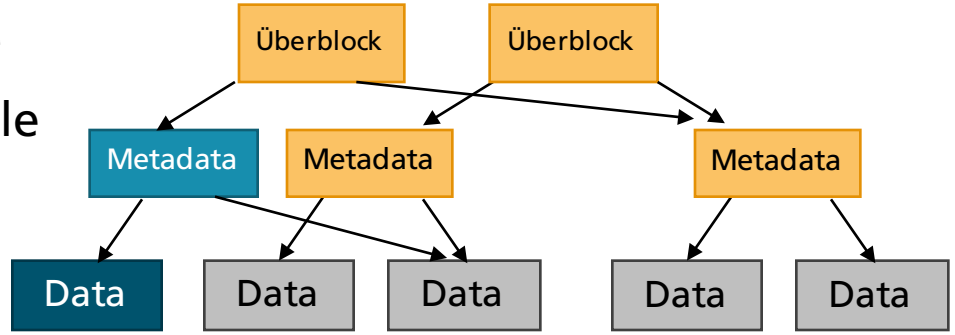
- Top-level vdevs are made up of children
- Different types of top-level vdevs are supported by ZFS
- Each top-level vdev is addressed using its ID



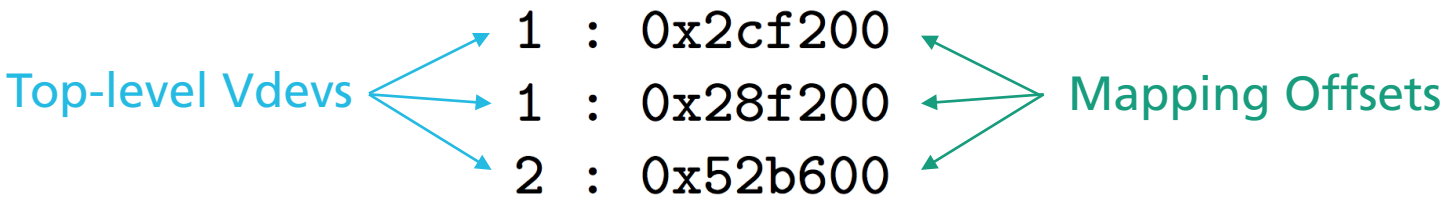
Pooled Storage File Systems

ZFS – General Structure

- Filesystem is represented in a tree-structure
- Überblock is the head of this tree
- Utilizes the copy-on-write principle

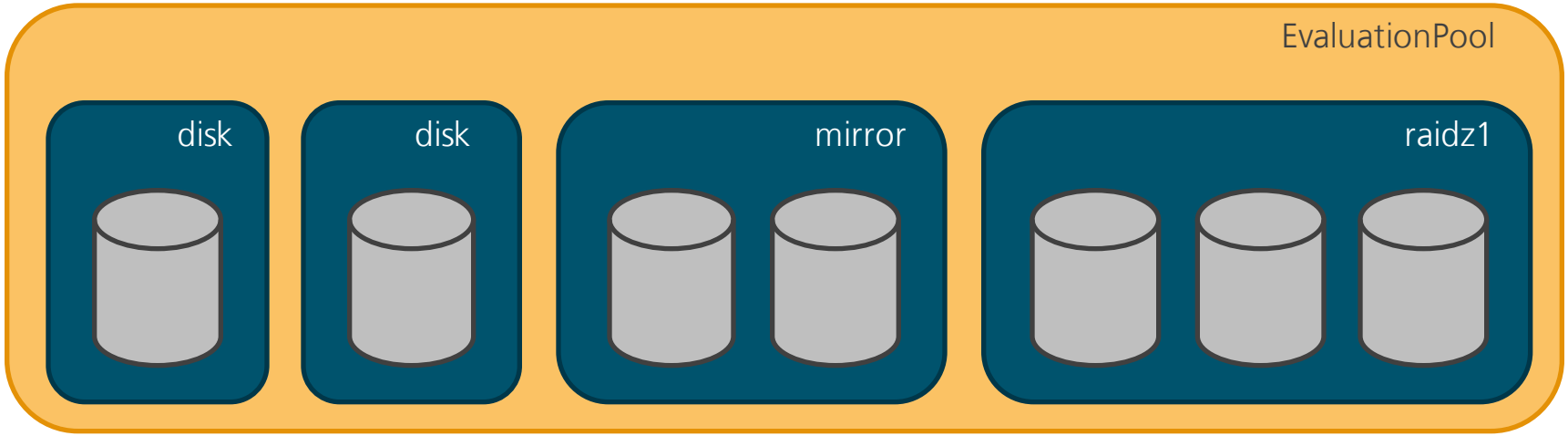


- Block pointers refer to variable-sized blocks of data
- Data is addressed by using Data Virtual Addresses (DVAs)



Evaluation
Evaluation Pool

- Use multiple numbers of disks in multiple configurations
  - Two simple disks
  - One mirror with two children
  - One raidz1 with three children



Contribution & Future Research

- Extension of The Sleuth Kit for pooled storage file systems
- Digital forensic analysis of ZFS
- Accessing older versions of the ZFS copy-on-write tree
- Dealing with incomplete pools

- Further investigations on other ZFS structures:
  - ZFS Intent Log (ZIL)
  - L2ARC
- Implementations for other pooled storage file systems
  - BTRFS, ReFS, APFS

Thanks for your attention!

https://github.com/fkie-cad/sleuthkit