Implementation of the file system layer in HelenOS

XXXII. Conference EurOpen.CZ

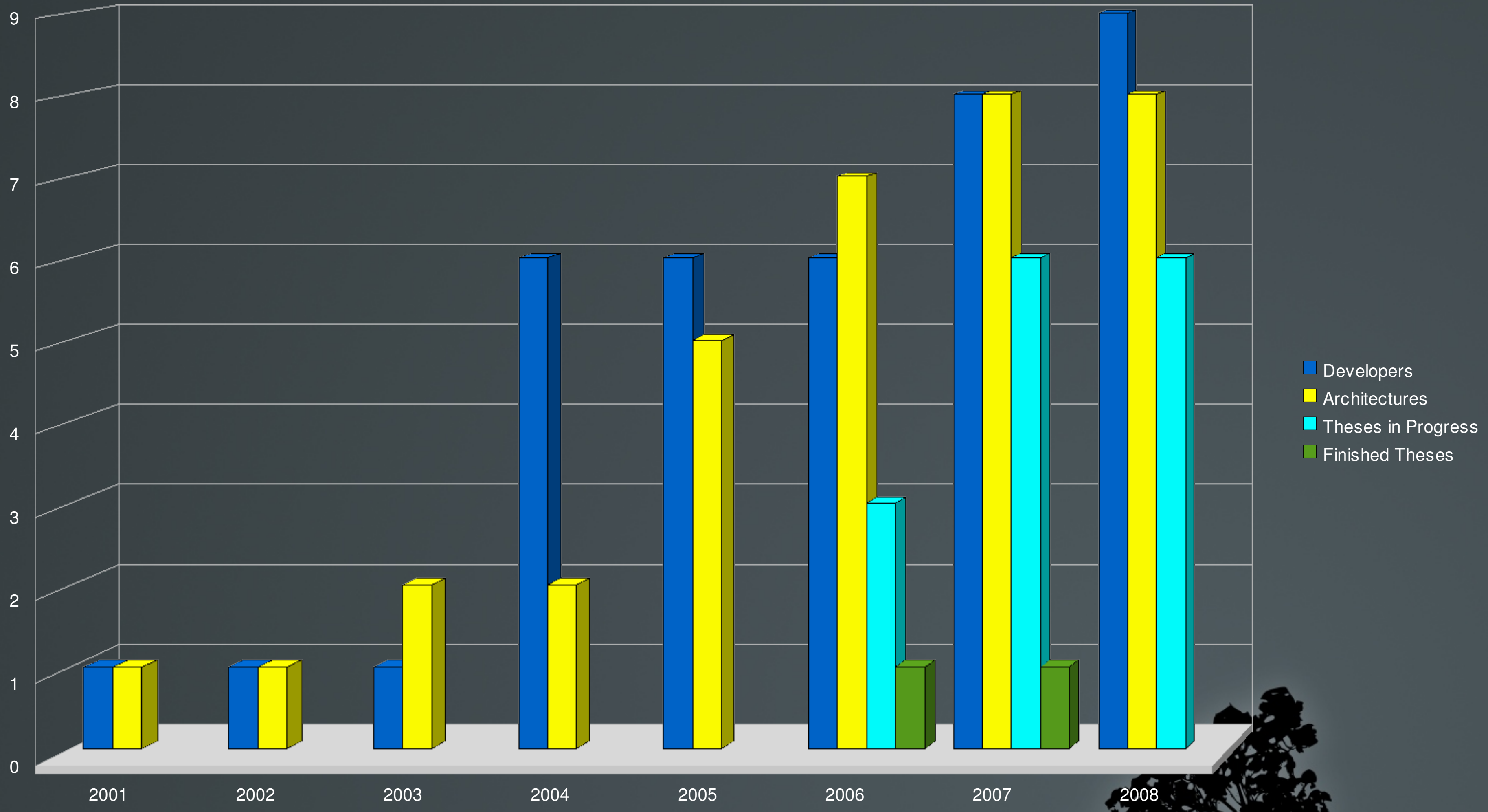
Jakub Jermar

May 21, 2008

HelenOS basic facts

- http://www.helenos.eu
- Experimental general purpose operating system
- Microkernel and userspace libraries and services
- Incomplete, under development
  - Lack of file system support
    - Major barrier preventing adoption

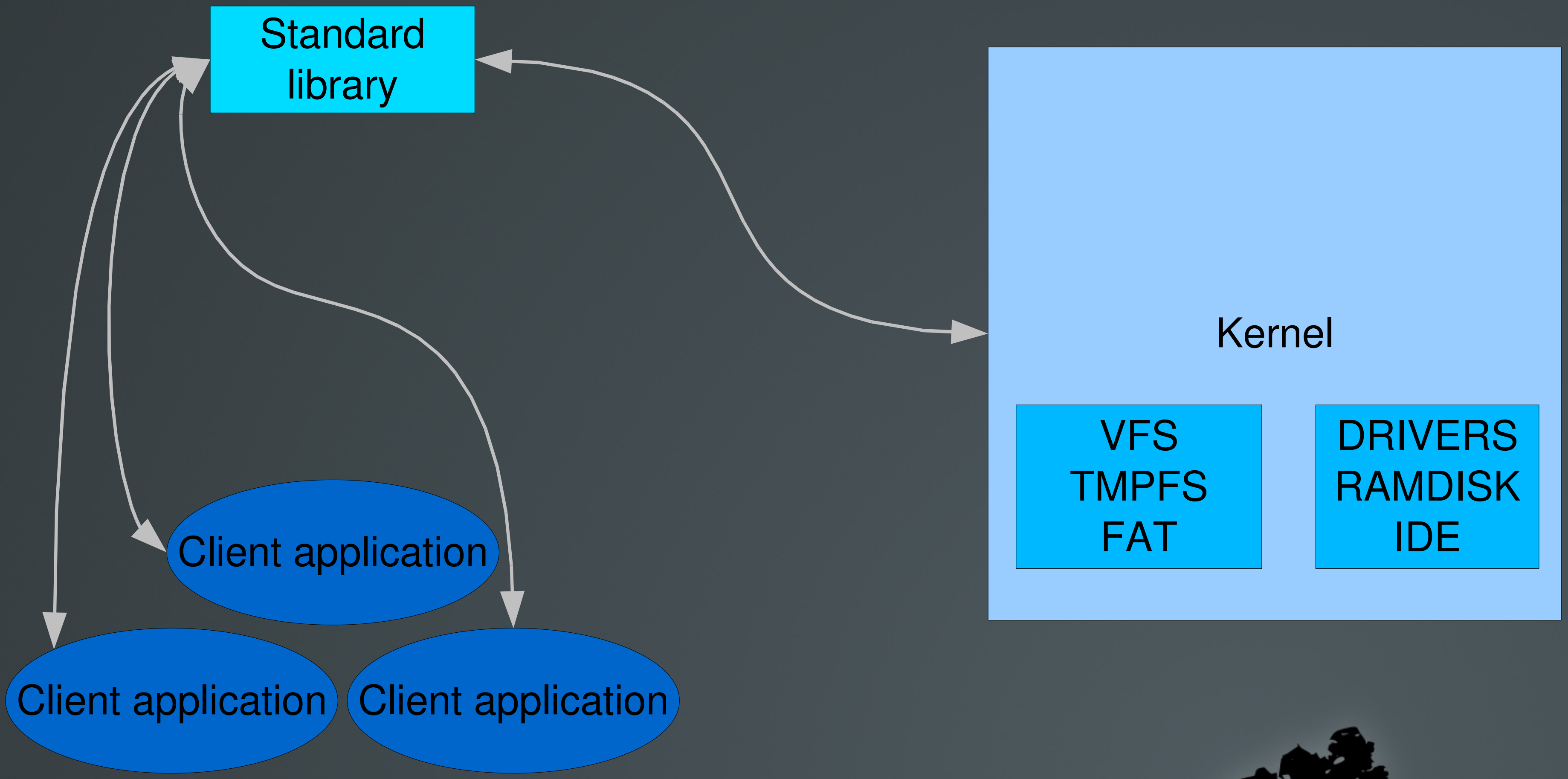
Project history



File systems vs. Monolithic kernels

- Well understood topic
  - Several well-known implementations
- Everything runs in one address space
  - VFS polymorphism via structures with function pointers
  - Function calls
  - Execution in kernel

Big picture: monolithic kernels



Standard library

Client application

Client application

Client application

Kernel

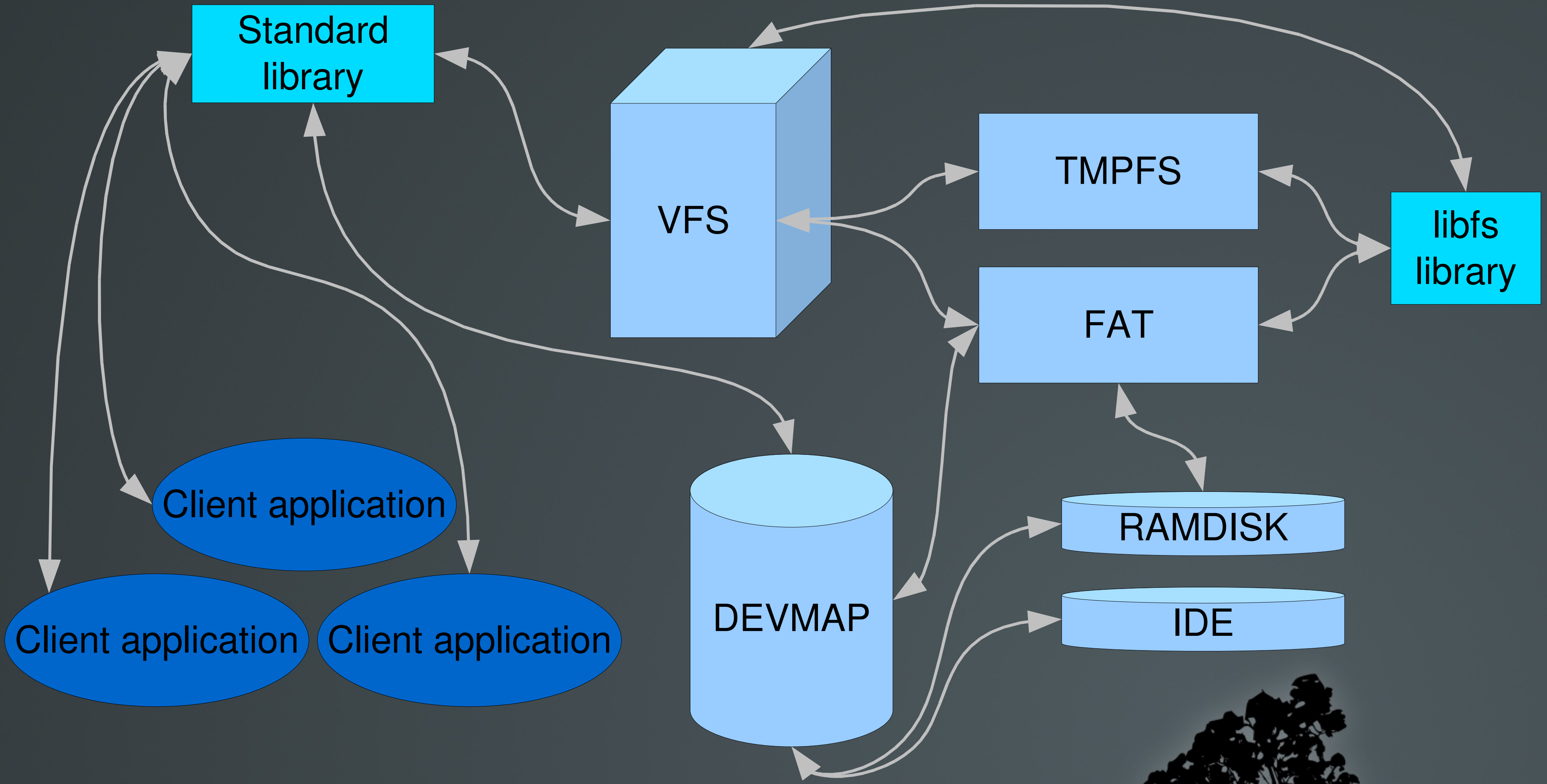
VFS
TMPFS
FAT

DRIVERS
RAMDISK
IDE

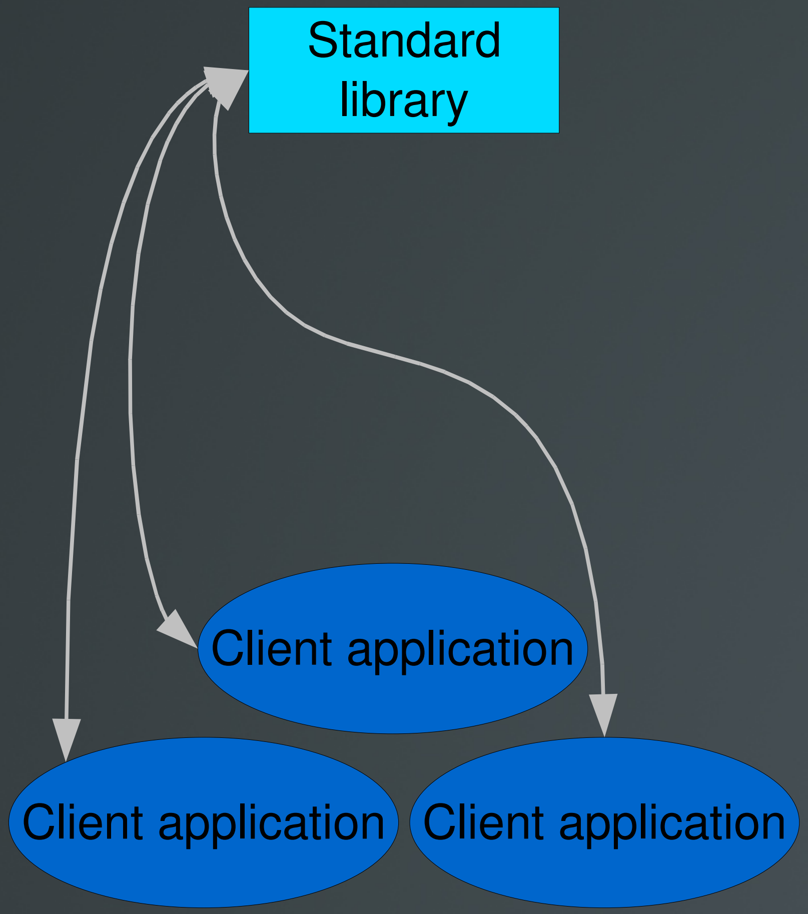
File systems vs. Microkernels

- Well understood topic?
- Problems:
  - What does the big picture (should) look like
    - No common address space
    - Breaking functionality into separate entities
    - Execution in userspace
  - How do the separate entities communicate?
    - IPC messages
    - Memory sharing
    - Data copying

Big picture: HelenOS



Standard library



Standard library

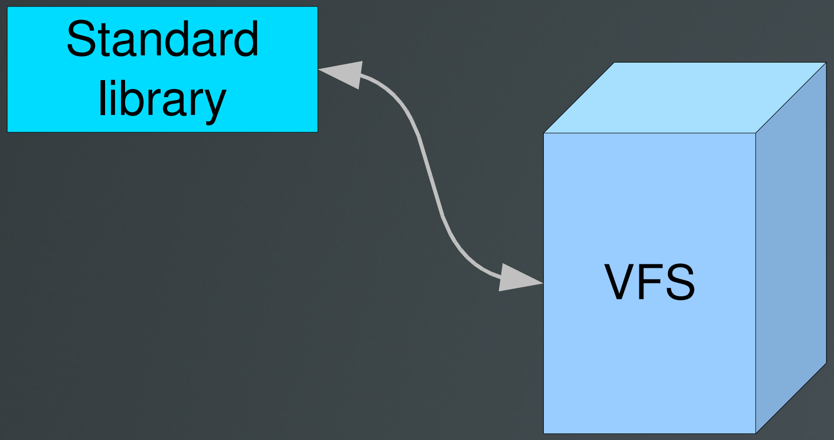
Client application

Client application

Client application

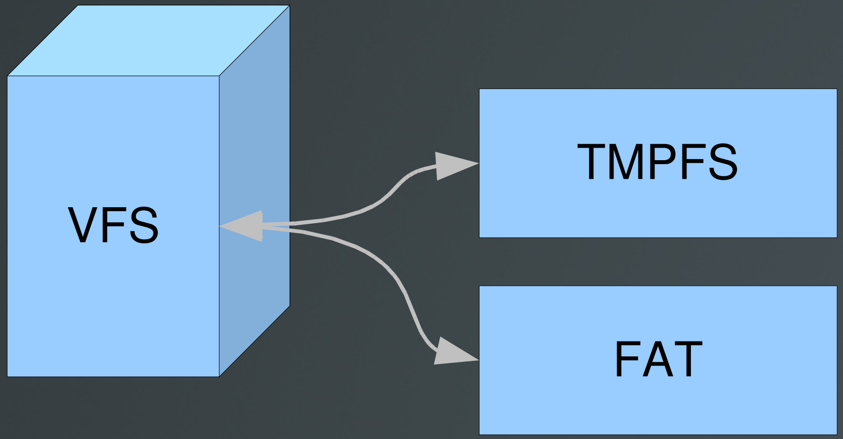
- Applications use a ”subset” of POSIX calls
- The library translates some of these calls to VFS calls
  - Directly
  - Wraps around others
- Relative to absolute paths

VFS frontend



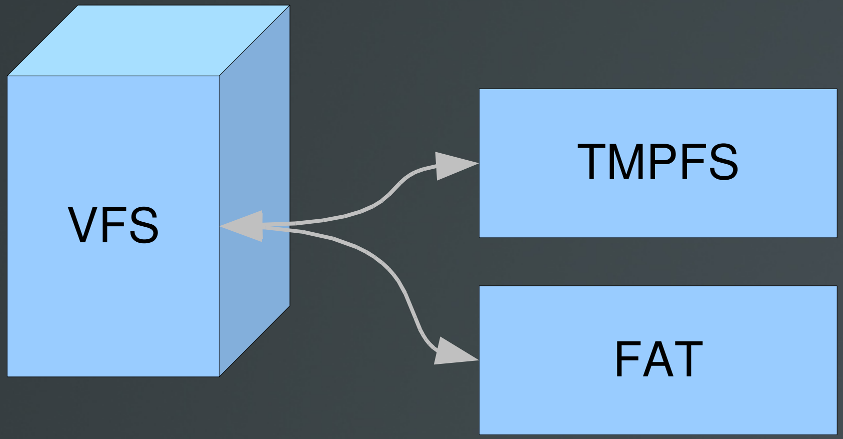
- VFS nodes
- Open files per client
- Path canonization
- Reference counting
- Synchronization
- Multiplexes requests

VFS backend



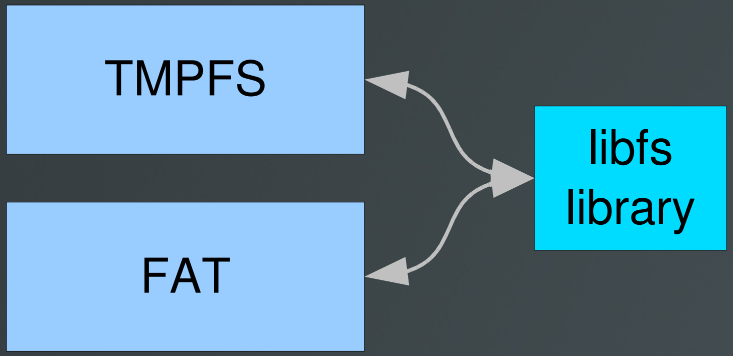
- Registry of individual FS
- Pathname Lookup Buffer
  - VFS shares PLB read-write
  - FS share PLB read-only
- VFS output protocol
  - All output VFS methods
  - All FS must understand it
  - VFS polymorphism

Individual FS servers



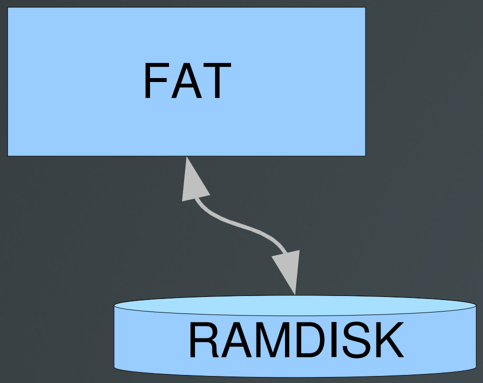
- Implement the output VFS protocol
- File system's logic
- Cache some data/metadata in memory
- Understand VFS triplets
  - (fs_handle, dev_handle, index)

libfs library



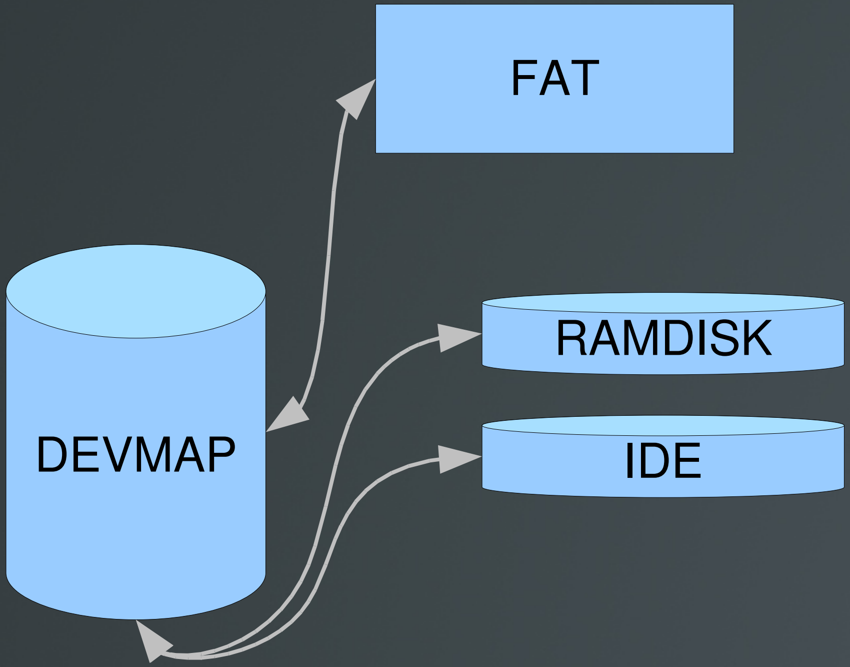
- FS registration code
- libfs_lookup
  - Template for VFS_LOOKUP
- libfs_ops_t
  - Virtual methods needed by libfs_lookup

Block device drivers



- All block devices required to implement the same protocol
- FS doesn't care what block device it is mounted on
- FS learns about the device and its driver via DEVMAP

DEVMAP



- Registry of block device drivers and block device instances
- Maps device handles to connections

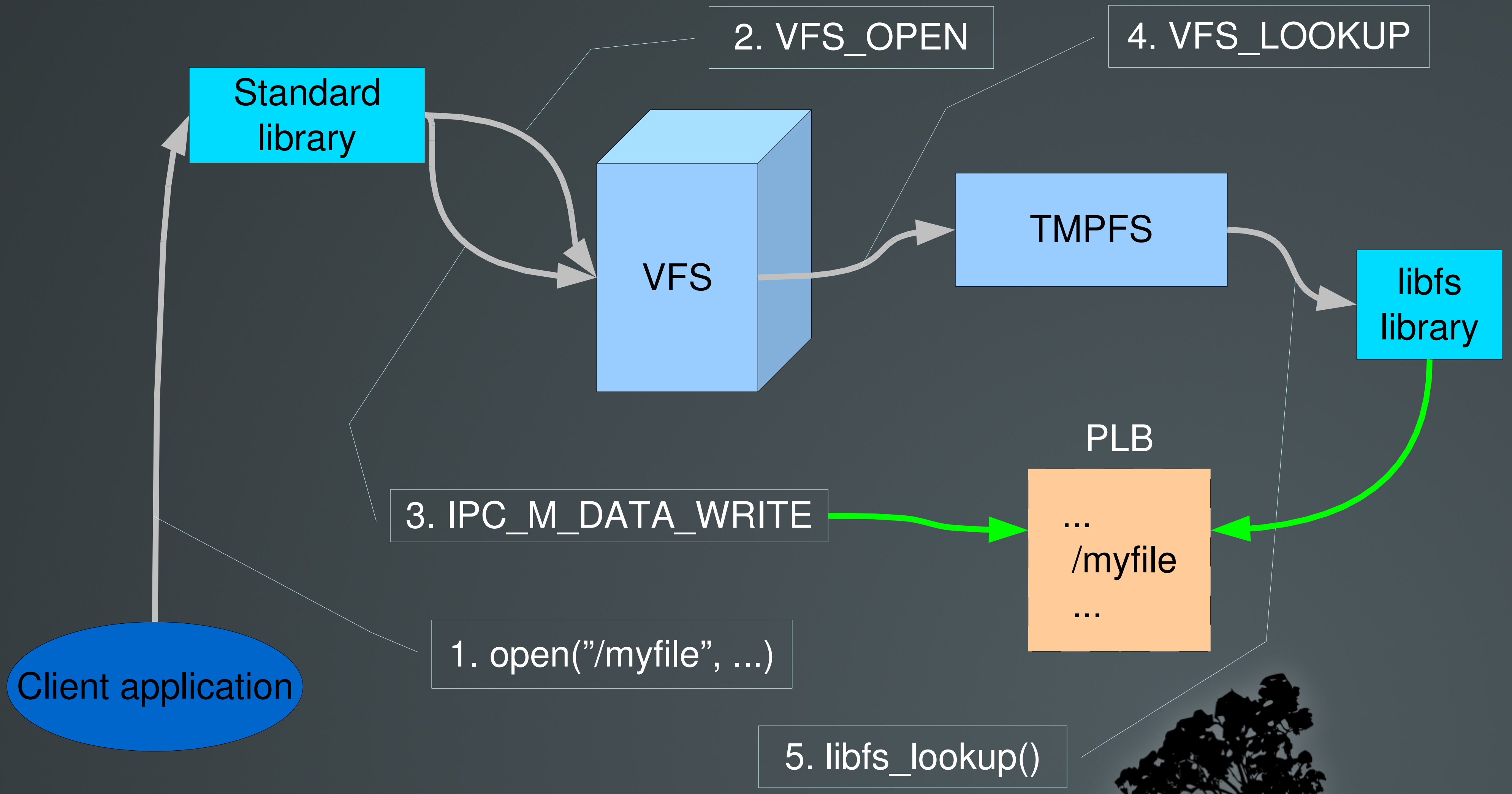
File system synchronization

- Mostly in VFS
  - VFS nodes have contents RW lock
  - Namespace RW lock
  - No locking for table of open files
    - Per-fibril data a.k.a. TLS
- Less synchronization in individual FS servers

Means of communication

- Short IPC messages (method + up to 5 arguments)
  - Both requests and answers
- Memory sharing and data copying integrated in IPC
  - Parties negotiate over IPC; kernel carries out the action
  - Combo messages
- Short and combo messages can be forwarded
  - Memory shared/data copied only between the endpoints
  - Sender masquerading

Communication example: open()



VFS + Standard library

- Fairly complete, but still evolving
  - mount(), open(), read(), write(), lseek(), close(), mkdir(), unlink(), rename()
  - opendir(), readdir(), rewinddir(), closedir()
  - getcwd(), chdir()
- Missing
  - stat(), unmount(), ...
  - mmap()

TMPFS

- Both metadata and data live in virtual memory
- No on-disk format
- No block-device needed
- Contents lost on reset
- Implementation complete
- Testing purposes

FAT

- FAT16
- Work in progress
- Not that easy as it might seem
  - Simple on-disk layout
  - Non-existence of stable and unique node indices
- Evolution of translation layer that provides stable unique indices

Perspective

- Finishing FAT
  - Needed for loading programs from disk
  - Needed for non-root mounts
- Evolving block device drivers
- Block cache
- More FS implementations
- Improving IPC mechanism for copying data
- Swapping to/from file system

http://www.helenos.eu

Jakub Jermar