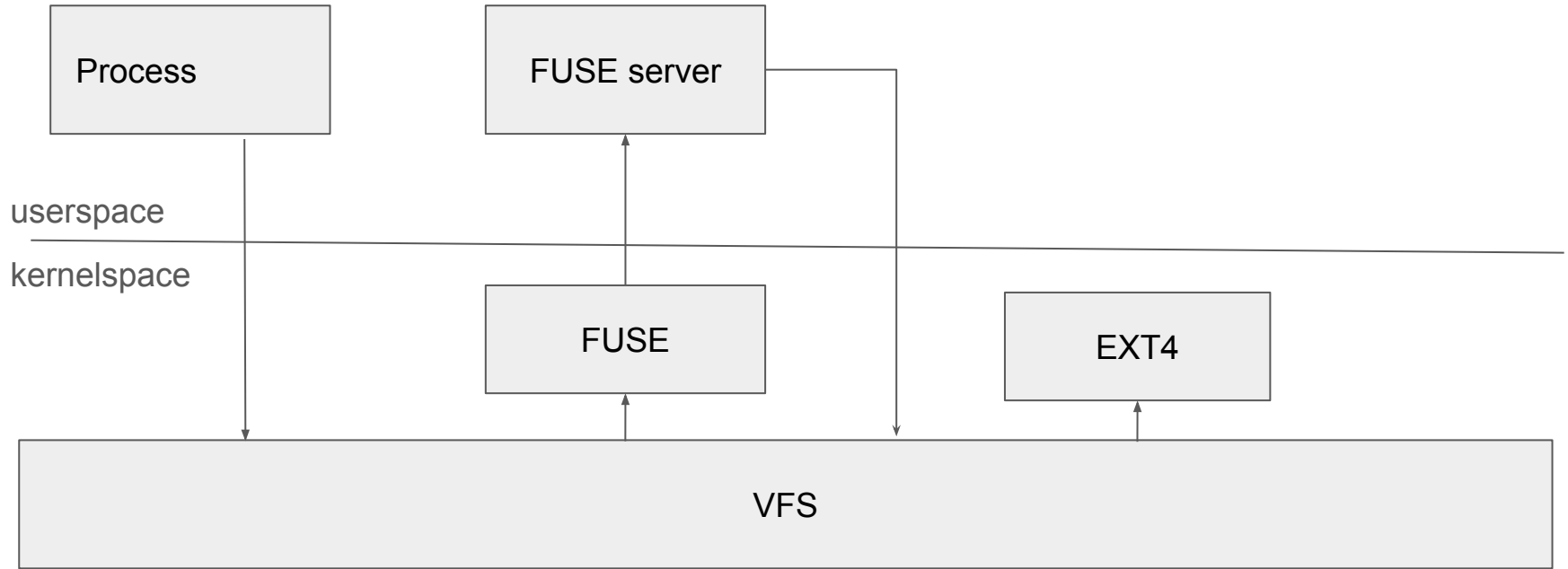


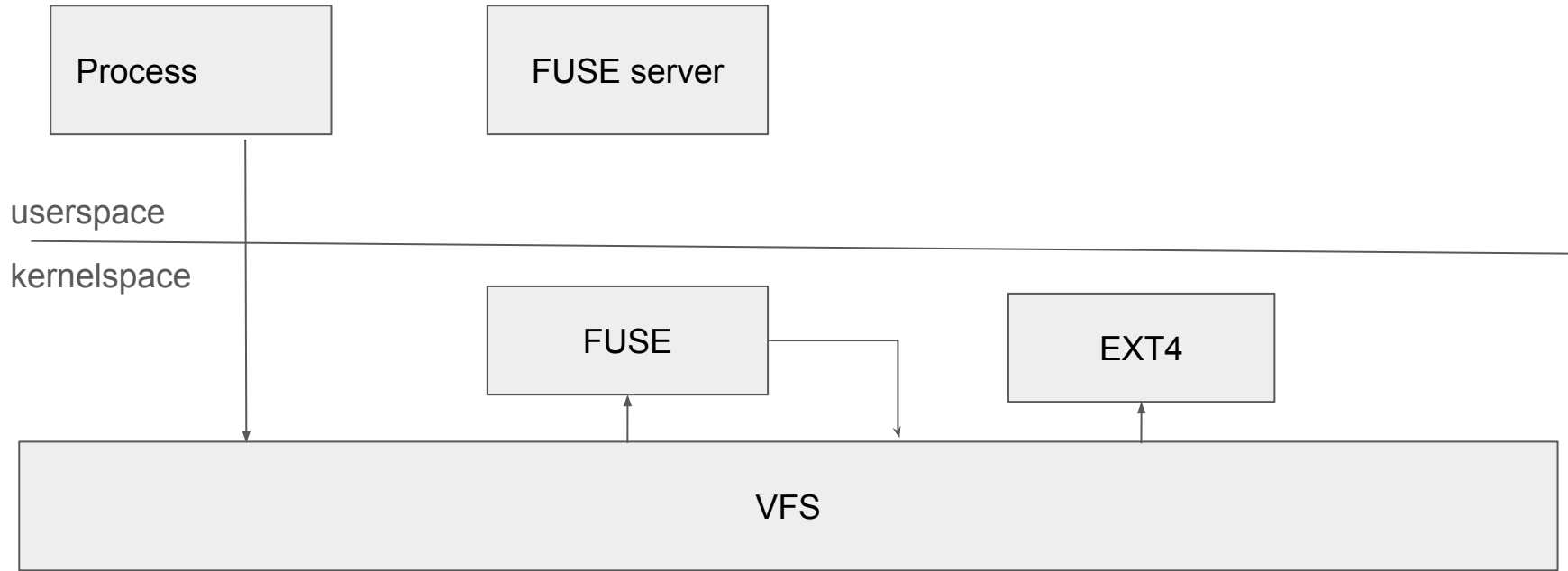
FUSE Kernel Developments

Passthrough Mode

Passthrough/loopback operation (w/o kernel support)



Passthrough/loopback operation (with kernel support)



Theory of operation

- Userspace provides a backing file to kernel
- Selected file system operations are performed directly on the backing file

Current version

- Implemented by Amir Goldstein (CTERA Networks)
- Only I/O is passed through
- Backing is established at open time

Raw API

```
backing_fd = open(backing_path, flags);
struct fuse_backing_map map = { .fd = backing_fd };
backing_id = ioctl(devfd, FUSE_DEV_IOC_BACKING_OPEN, &map);

struct fuse_open_out outarg;
outarg.open_flags |= FOPEN_PASSTHROUGH;
outarg.backing_id = backing_id;

ioctl(devfd, FUSE_DEV_IOC_BACKING_CLOSE, &backing_id);
```

WIP

- Getattr passthrough (Amir Goldstein)
 - Backing is established in FUSE_LOOKUP
- Fuse-bpf (Daniel Rosenberg, Paul Lawrence)
 - eBPF script installed into inode
 - Script is hooked into ops (most fs operations)
 - Script tells whether to use backing file or not
 - May modify arguments

CVMFS requirements

- Open/release passed through
- Split backing files

Questions?