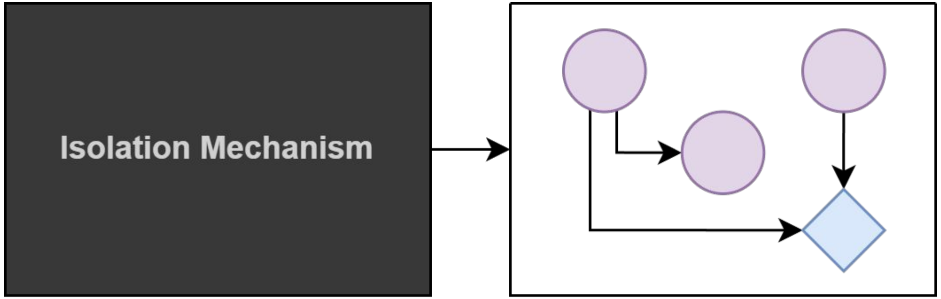
CellulOS: Building an OS on seL4

Arya Stevinson, Linh Pham

Advised by: Margo Seltzer, Reto Achermann, Aastha Mehta
Grads: Sid Agrawal, Shaurya Patel

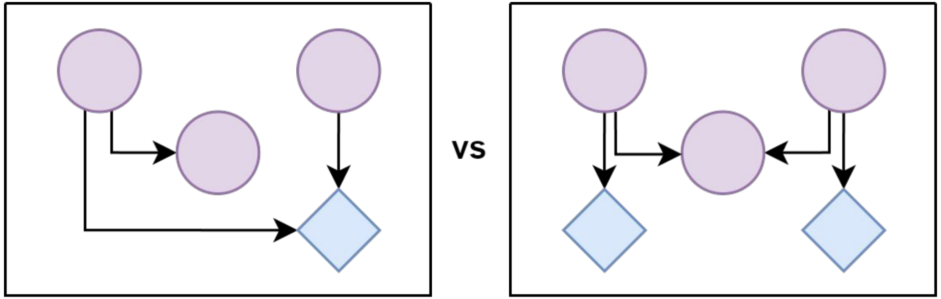
The Big Picture

1. Modelling



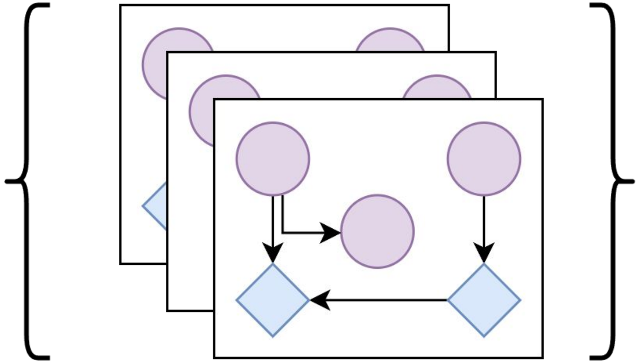
Isolation Mechanism

2. Comparing



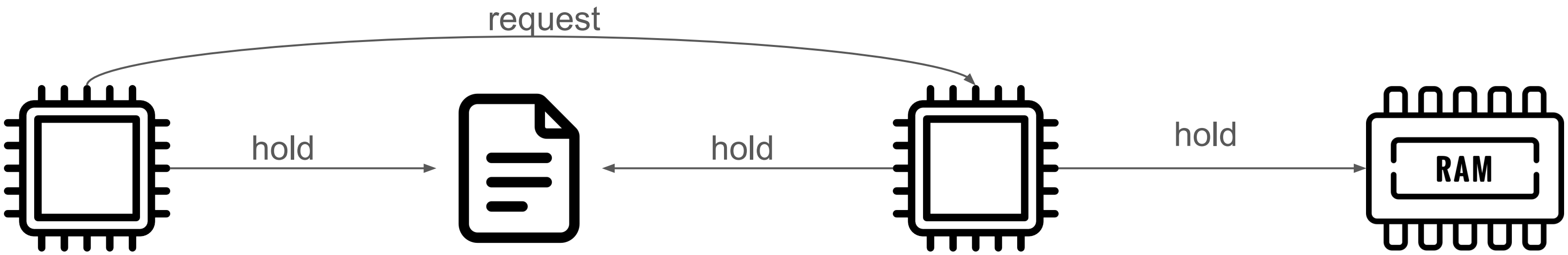
vs

3. Enumerating & Exploring



What do we need from a custom OS?

1. Ability to extract the **model state**



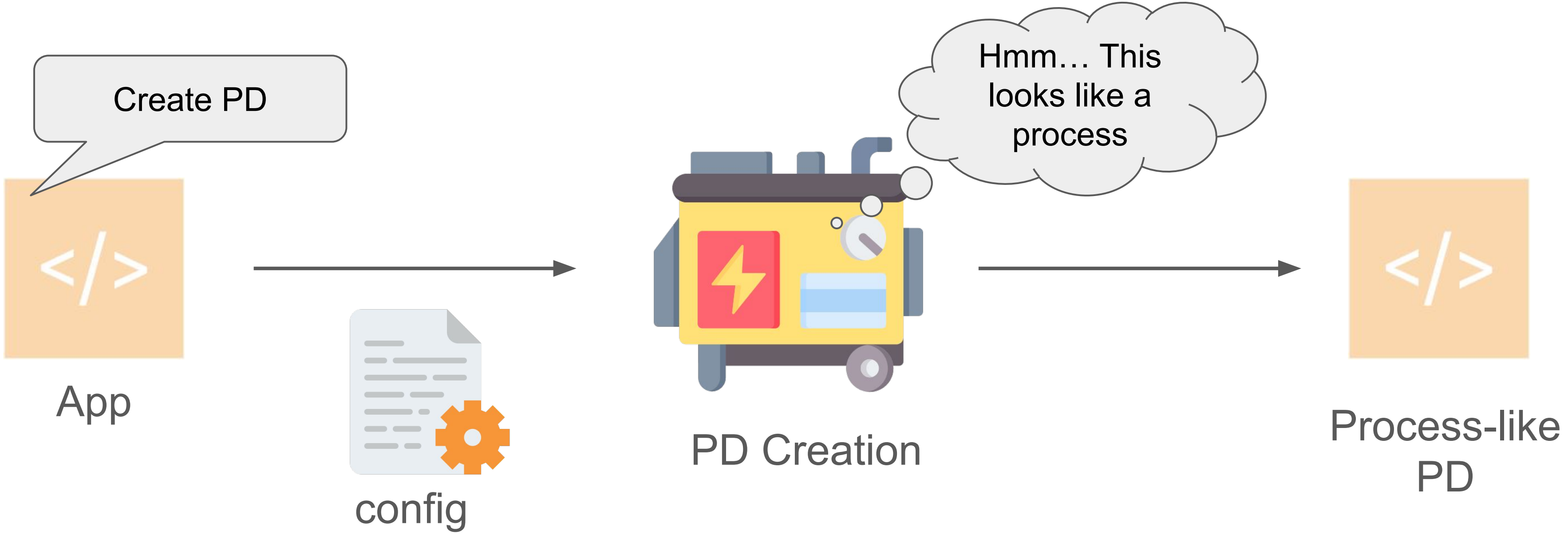
Track and/or extract:

- resources and their dependencies
- active entities (protection domains), their resources, and their dependencies

What do we need from a custom OS?

A protection domain (PD) is simply an active entity in the system that runs some code

2. Ability to build isolation abstractions in a unified way



Create PD

App

config

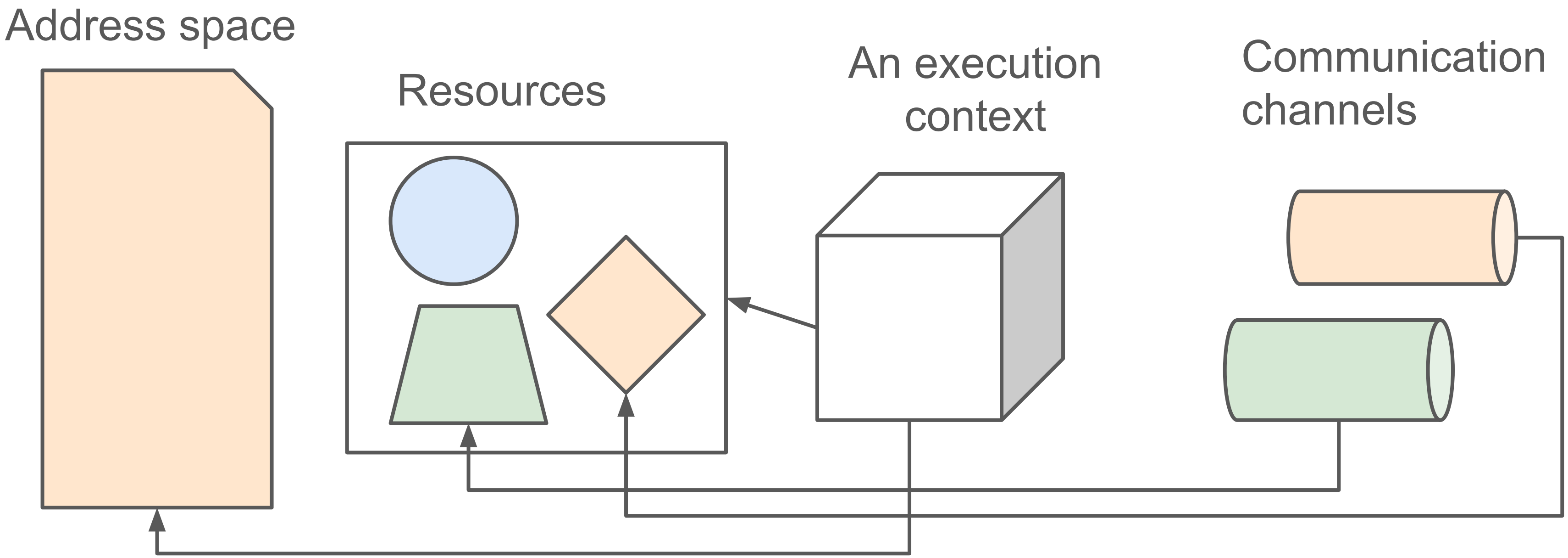
PD Creation

Hmm… This looks like a process

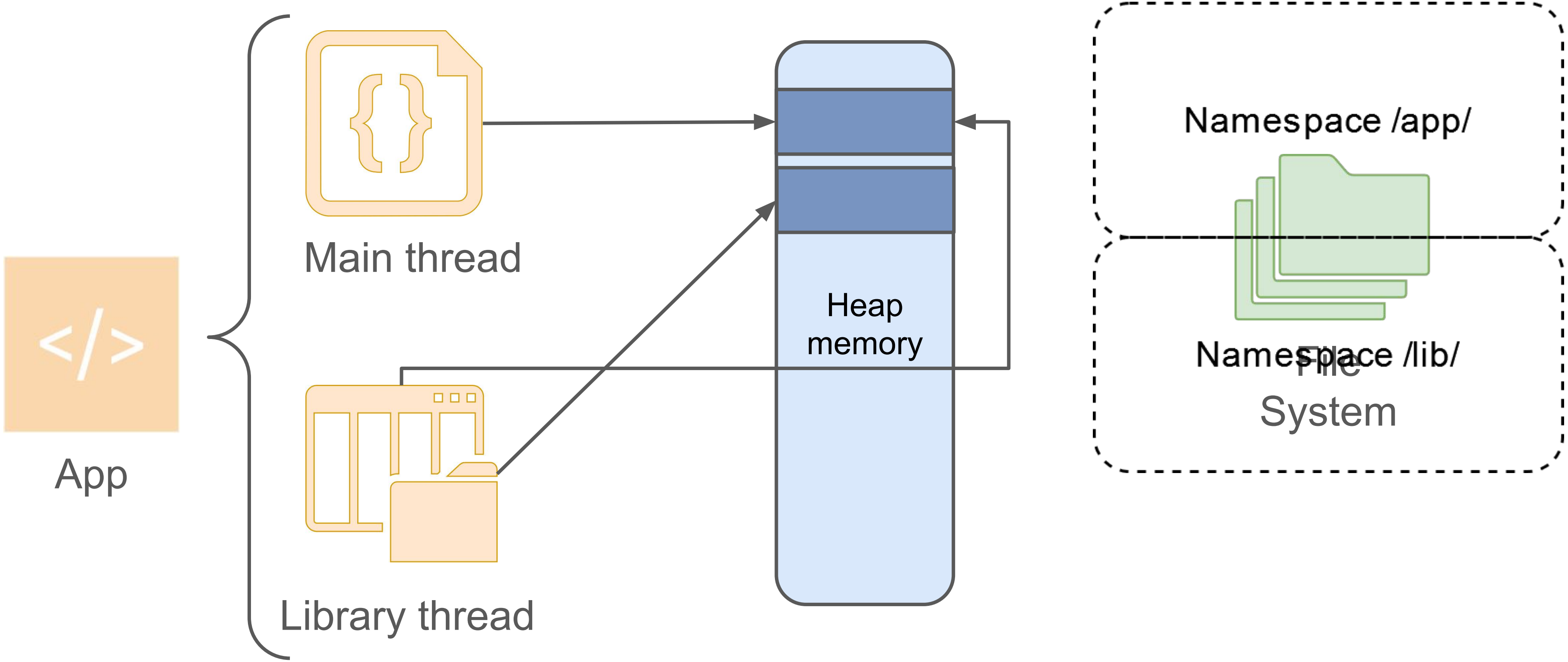
Process-like PD

Minimum specification of a Configuration

The core components required for any PD to run:

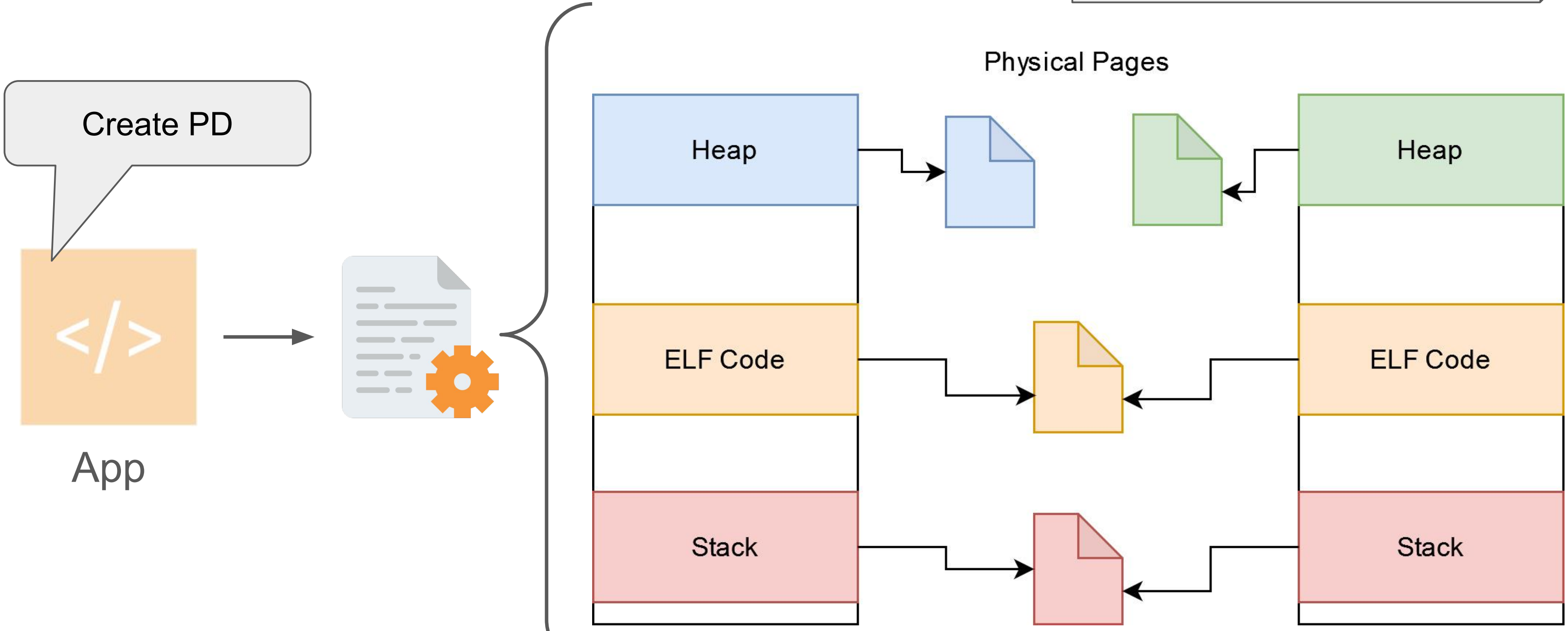


PDs for specialized use-cases



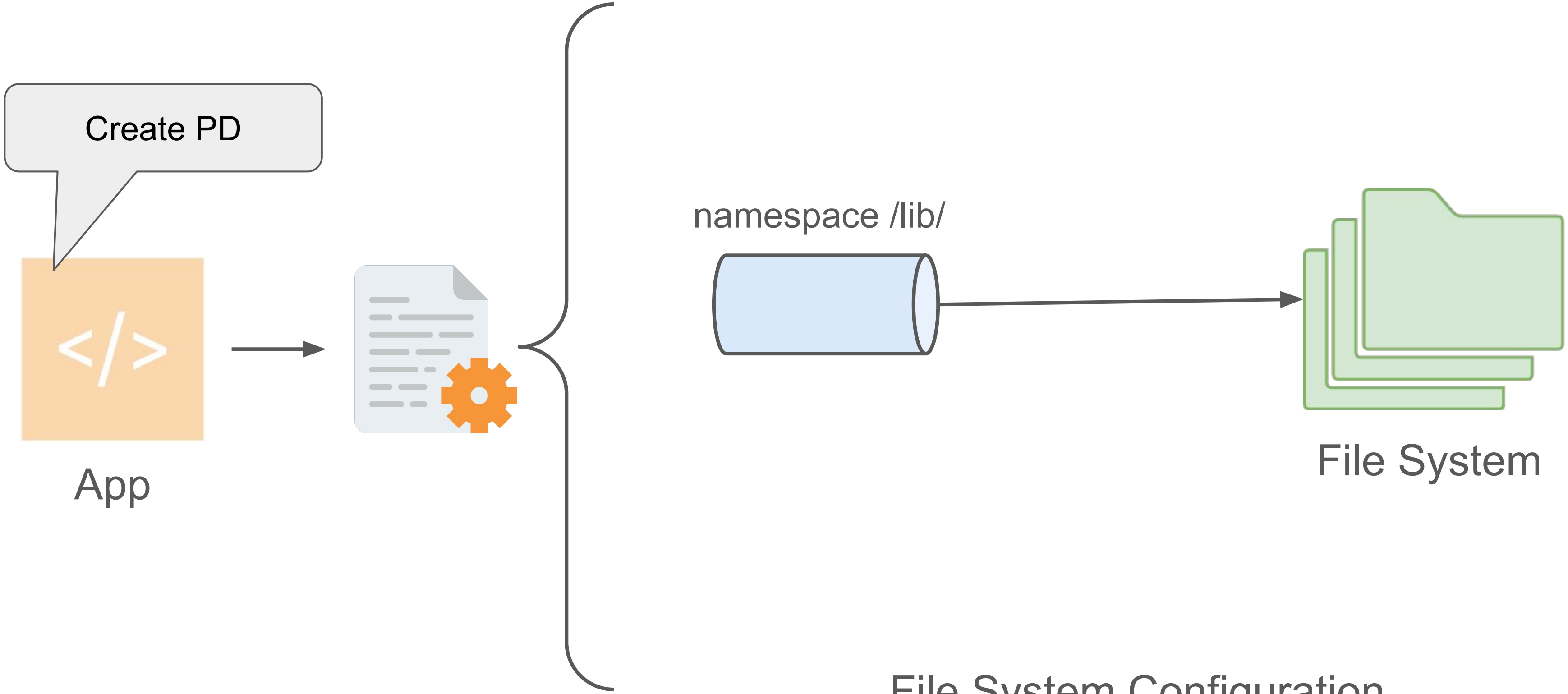
Special configs for special PDs

Configuration is always in reference to another PD



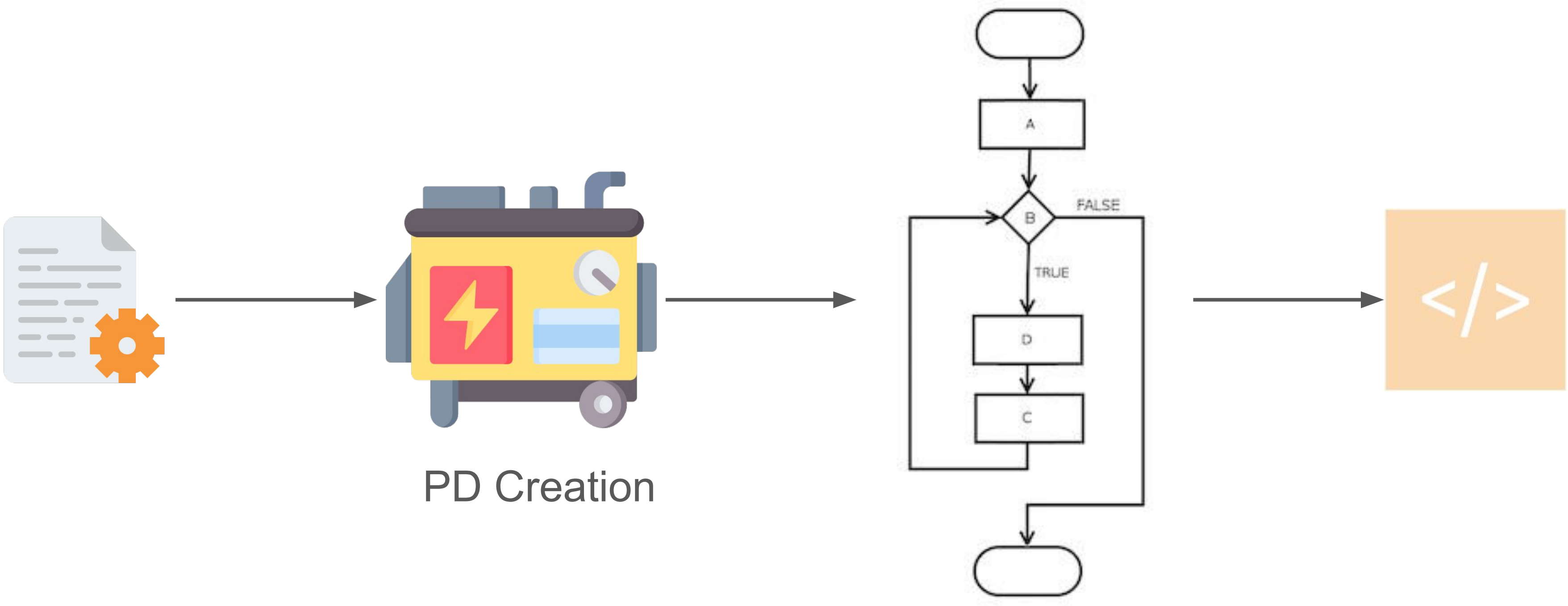
Address Space Configuration

Special configs for special PDs

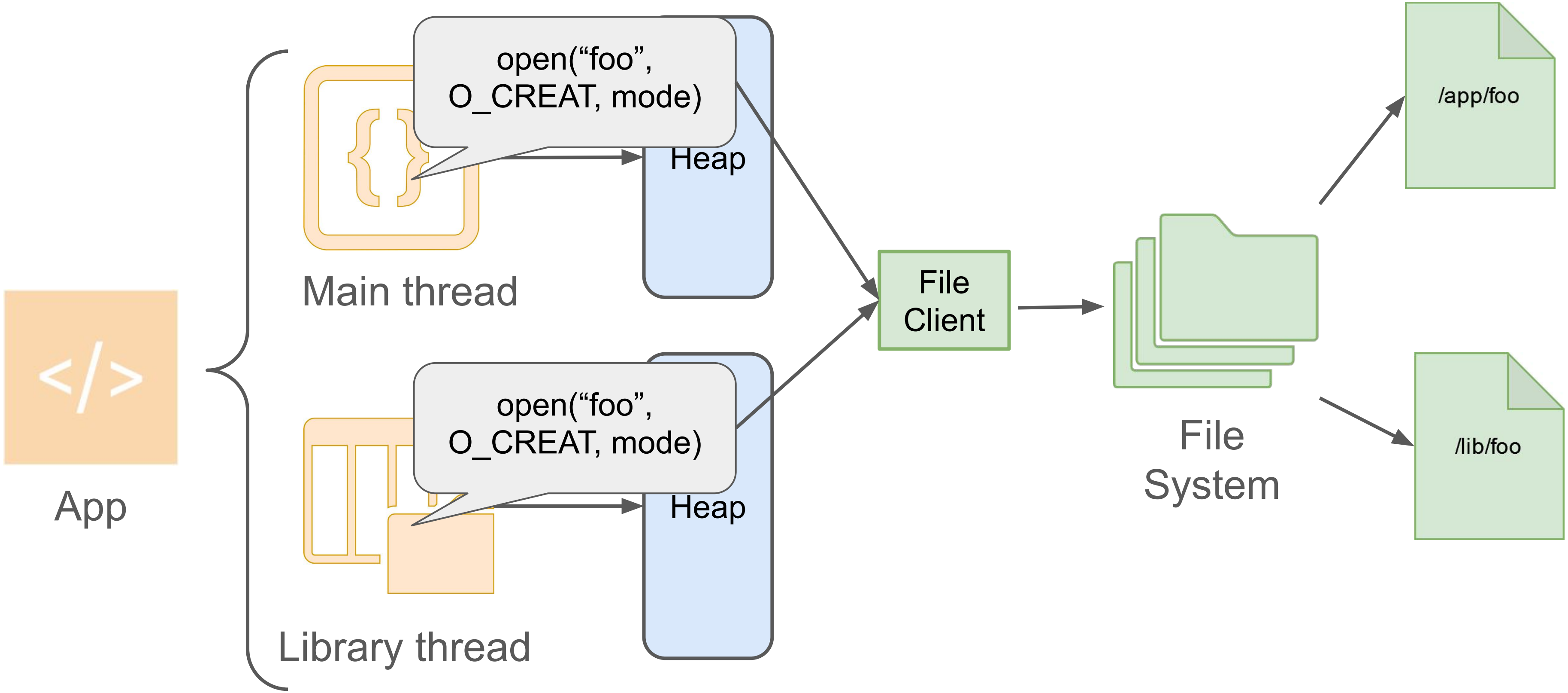


File System Configuration

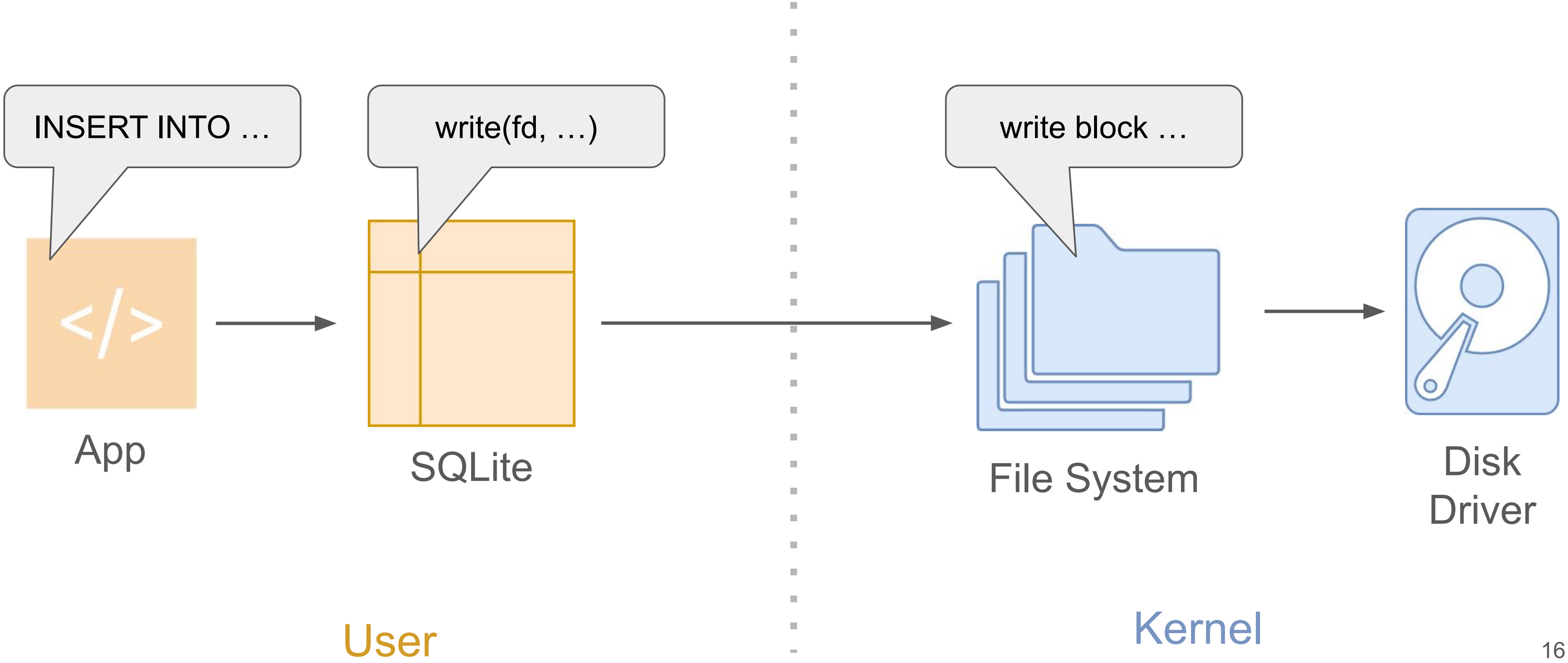
Setting up PDs based on Configuration



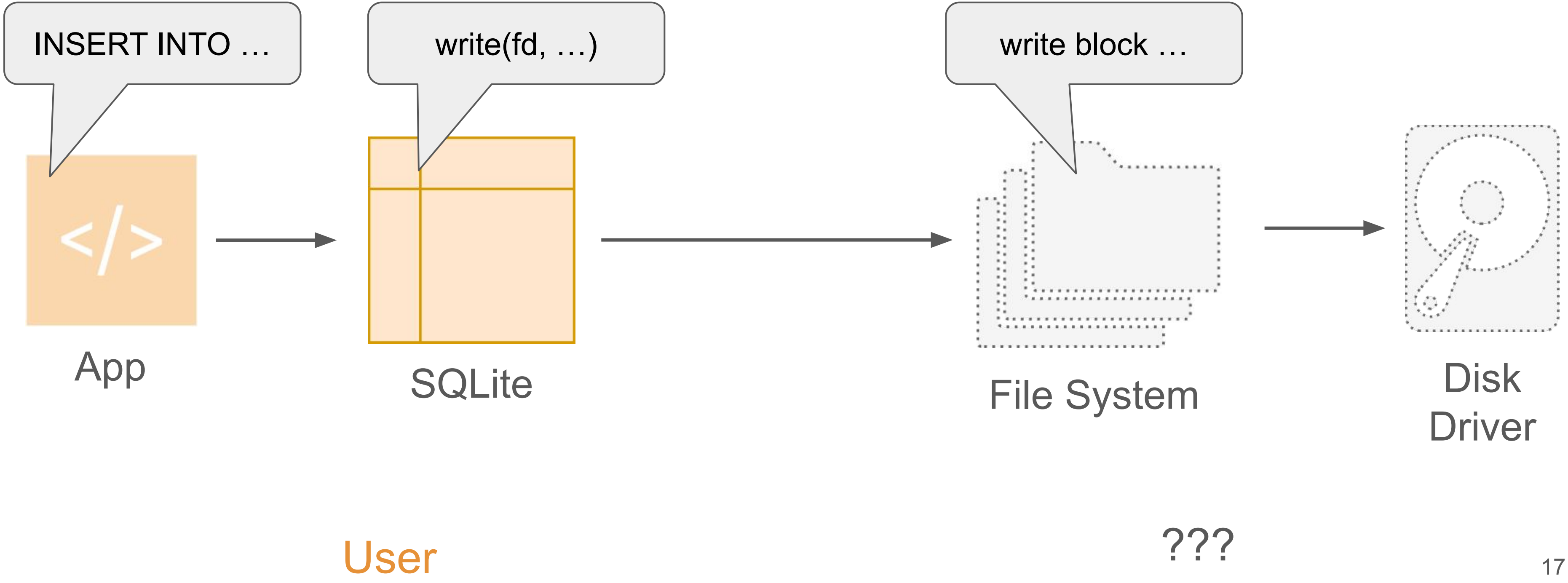
Specialized PD in action



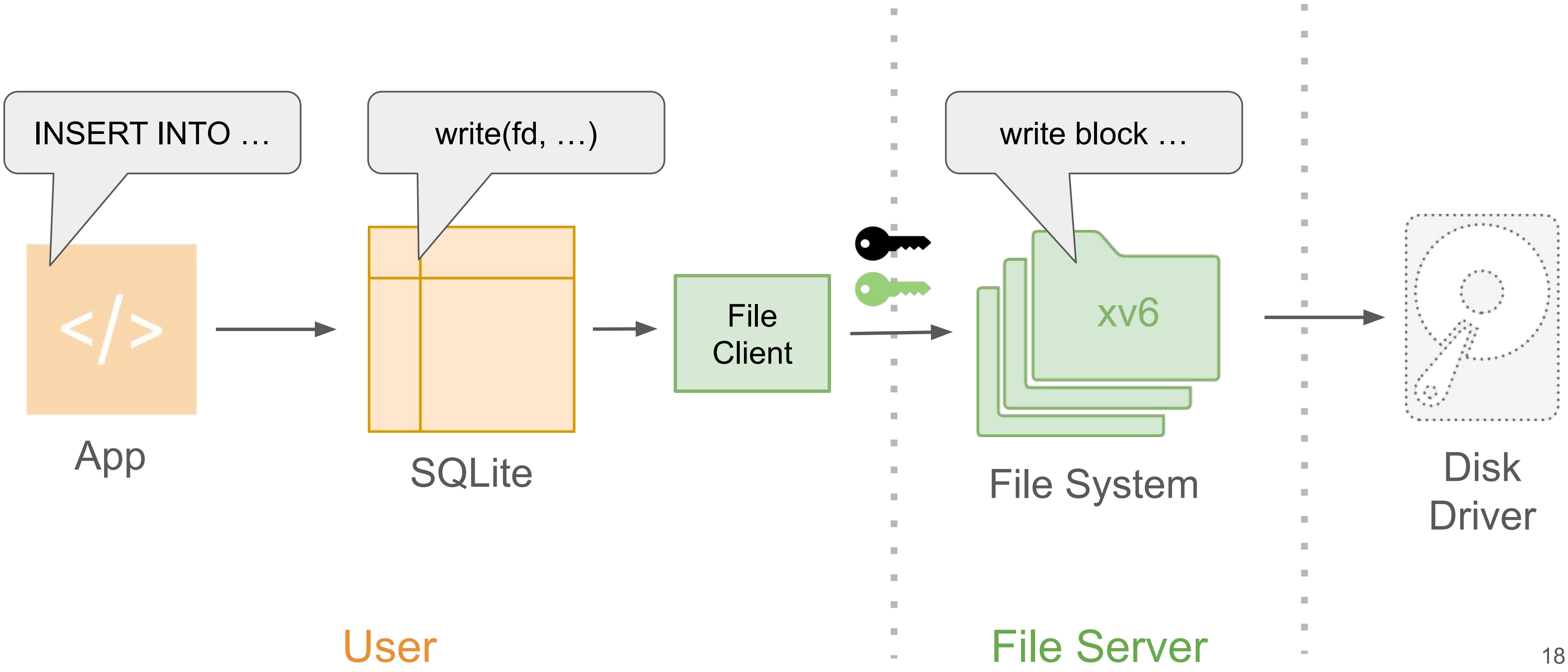
A Useful OS Runs Programs (it’s kind of the point)



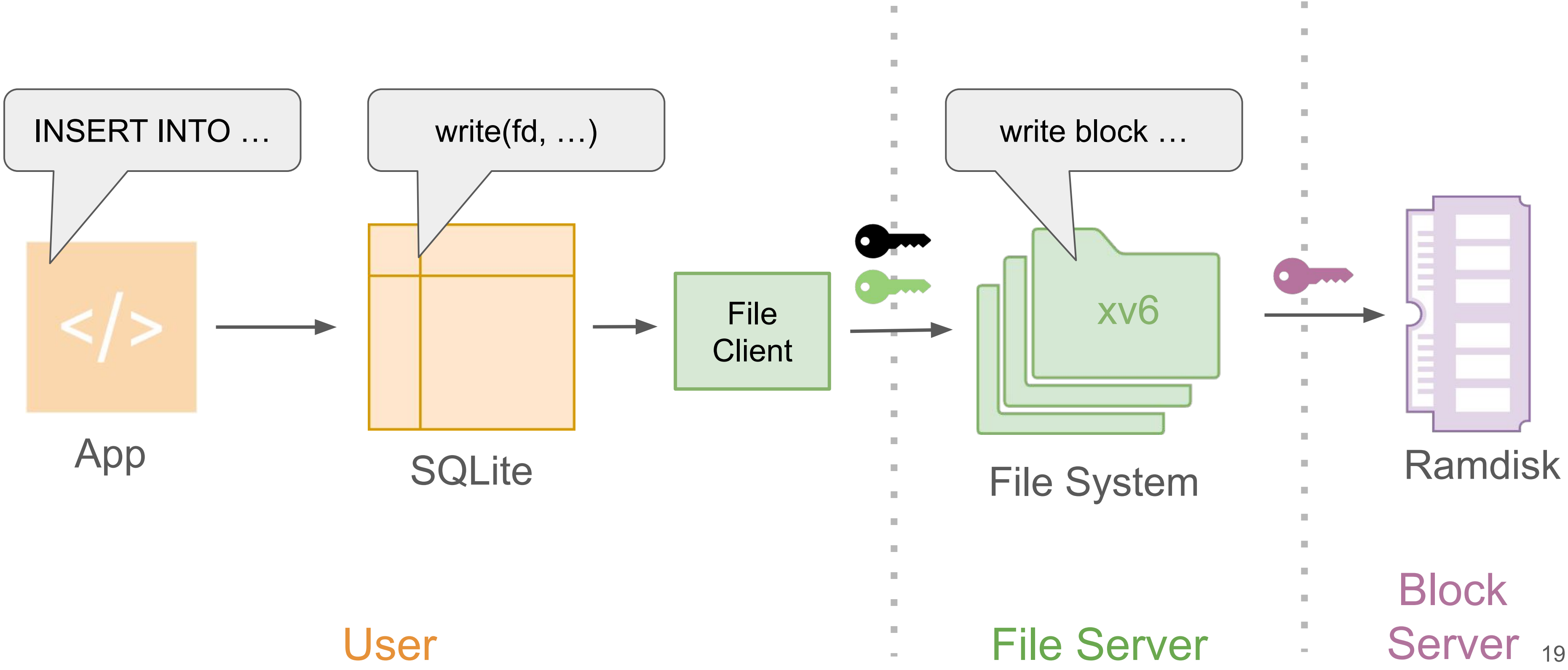
A Useful OS Runs Programs (but seL4 doesn't do much for us)



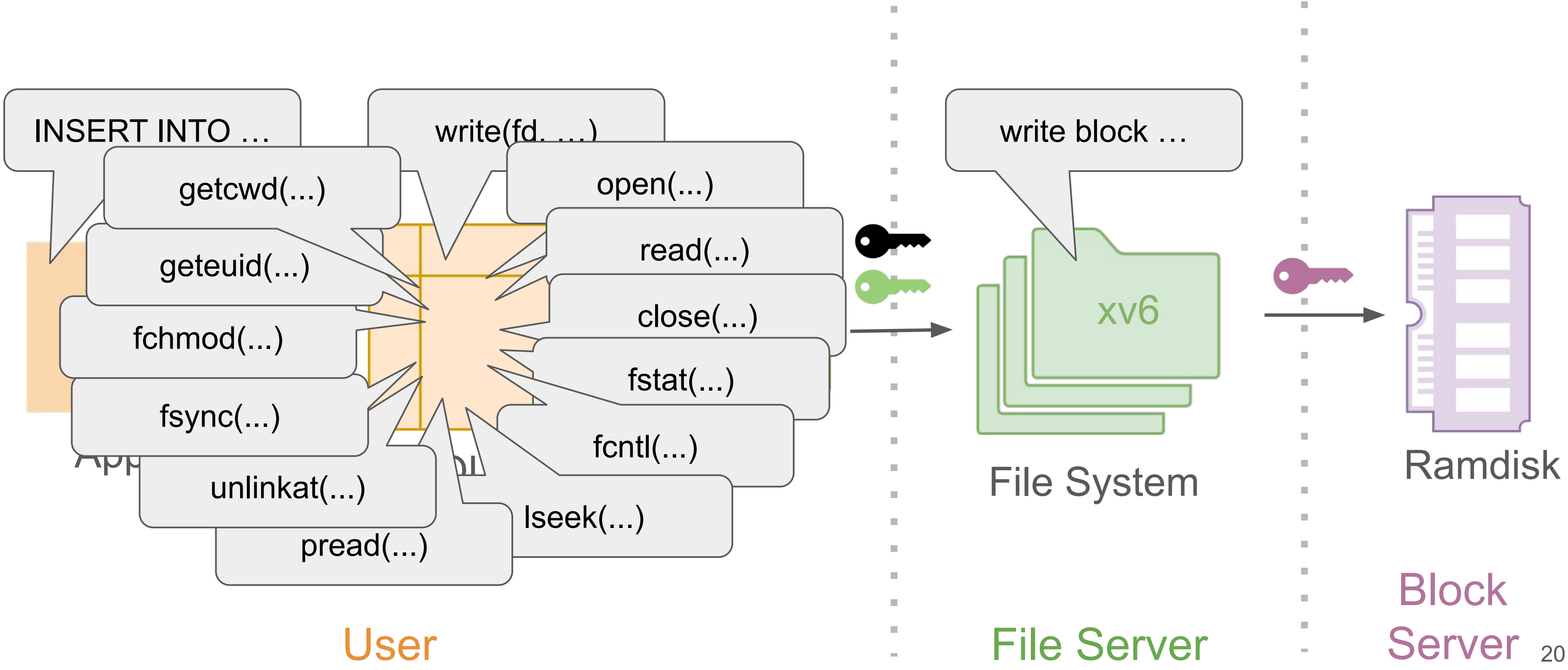
A Useful OS Runs Programs (in the microkernel way)

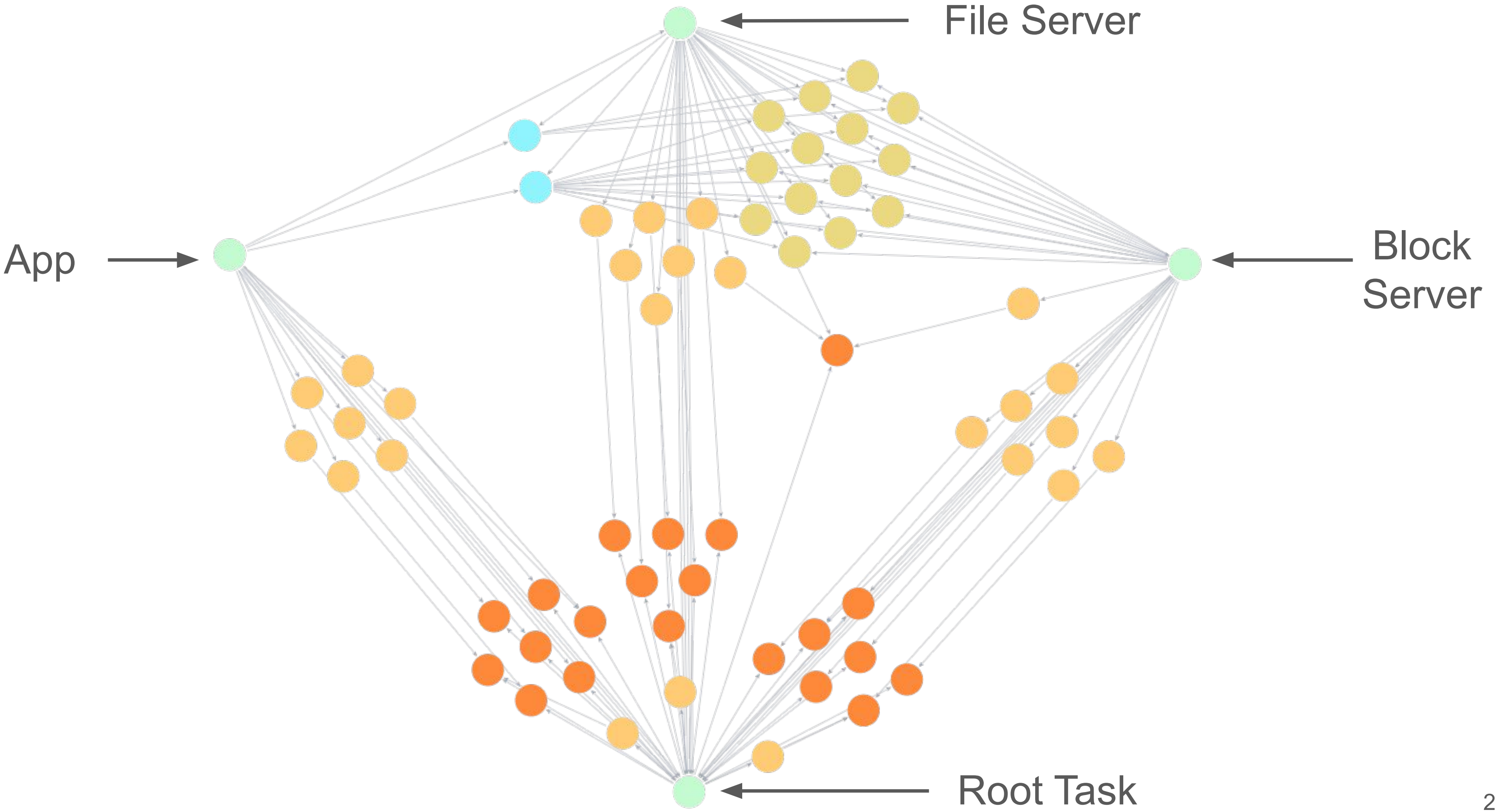


A Useful OS Runs Programs (in the microkernel way)



A Useful OS Runs Programs (in the microkernel way)



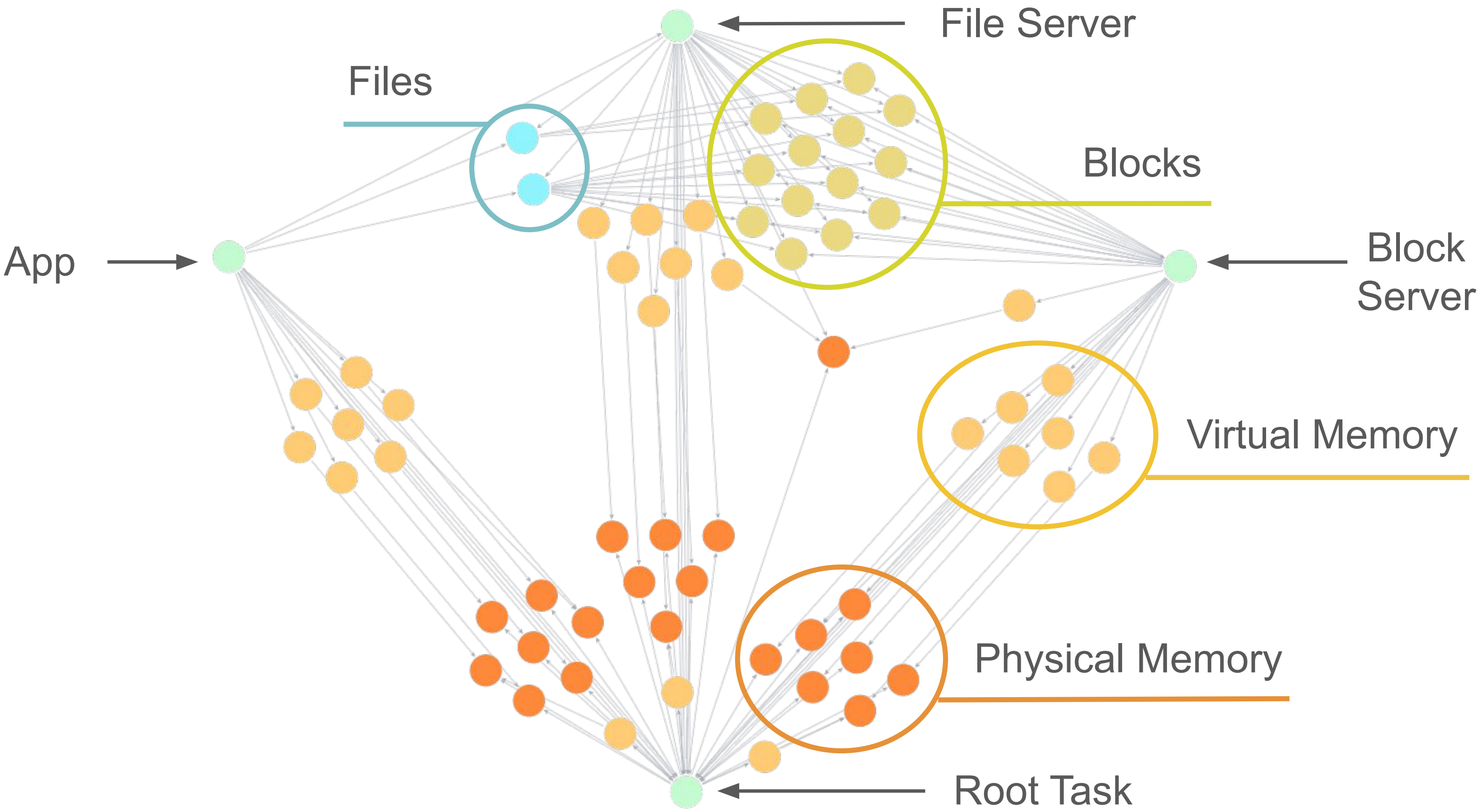


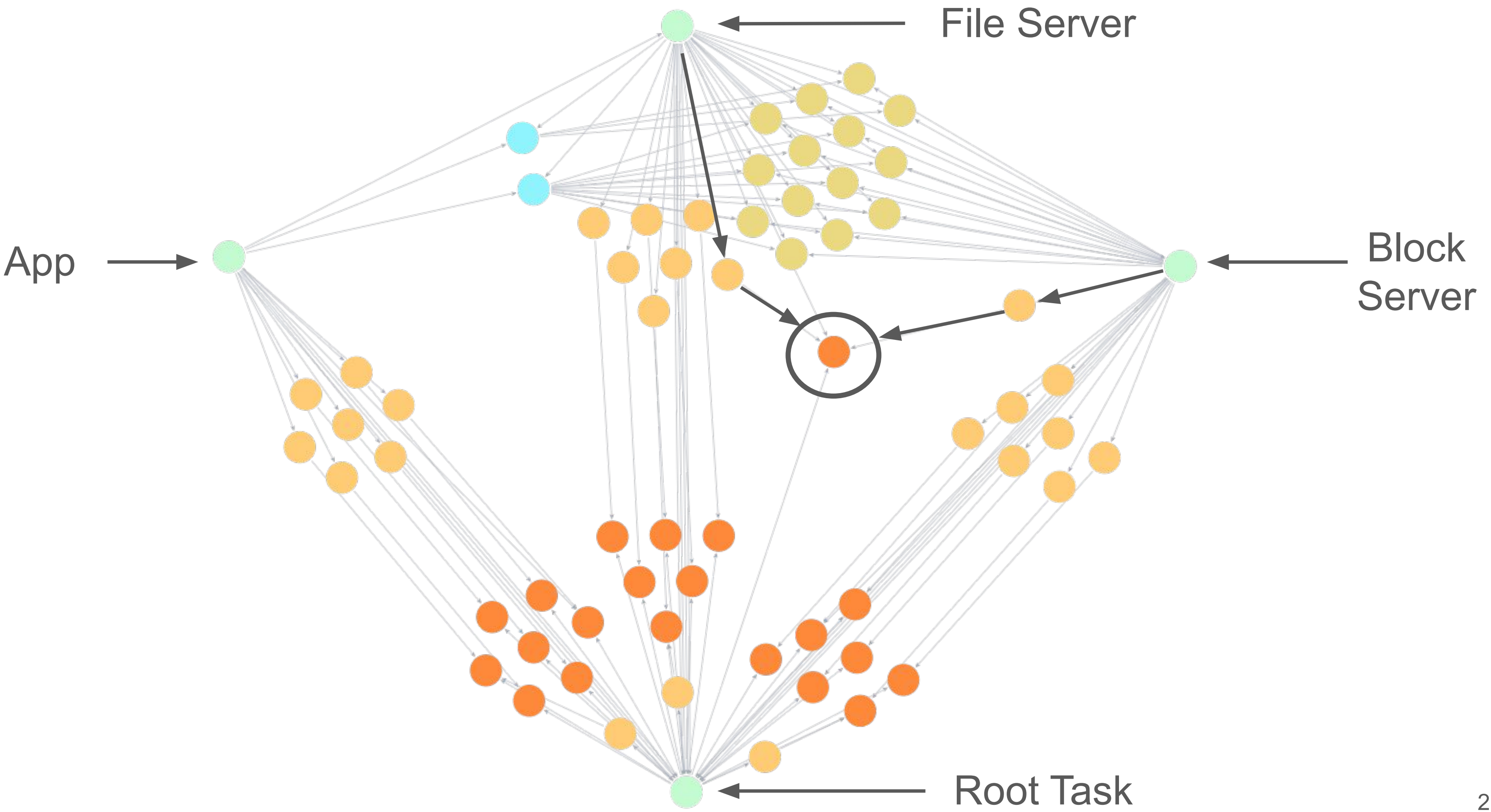
File Server

App

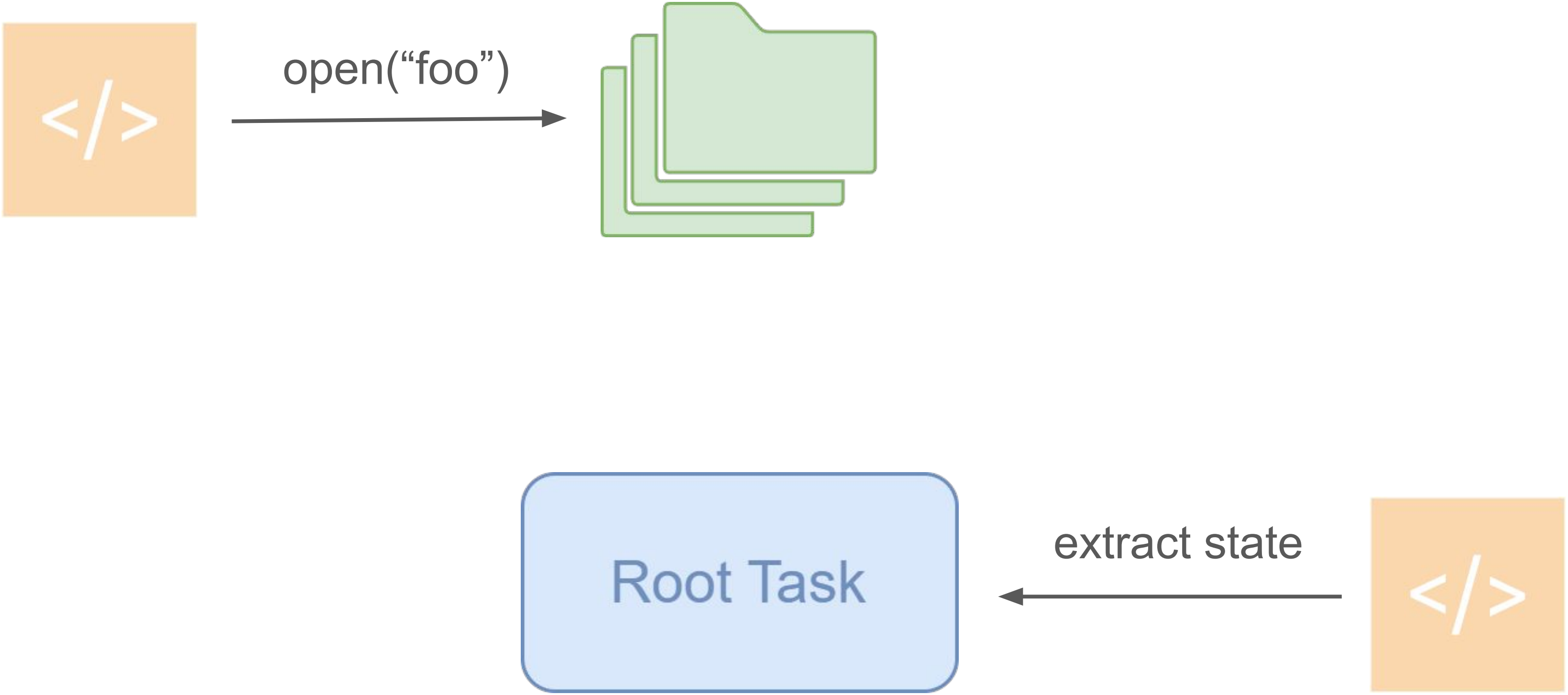
Block Server

Root Task


File Server
Files
Blocks
App
Block Server
Virtual Memory
Physical Memory
Root Task


File Server
App
Block Server
Root Task

An Interesting Challenge: Communication & Deadlock

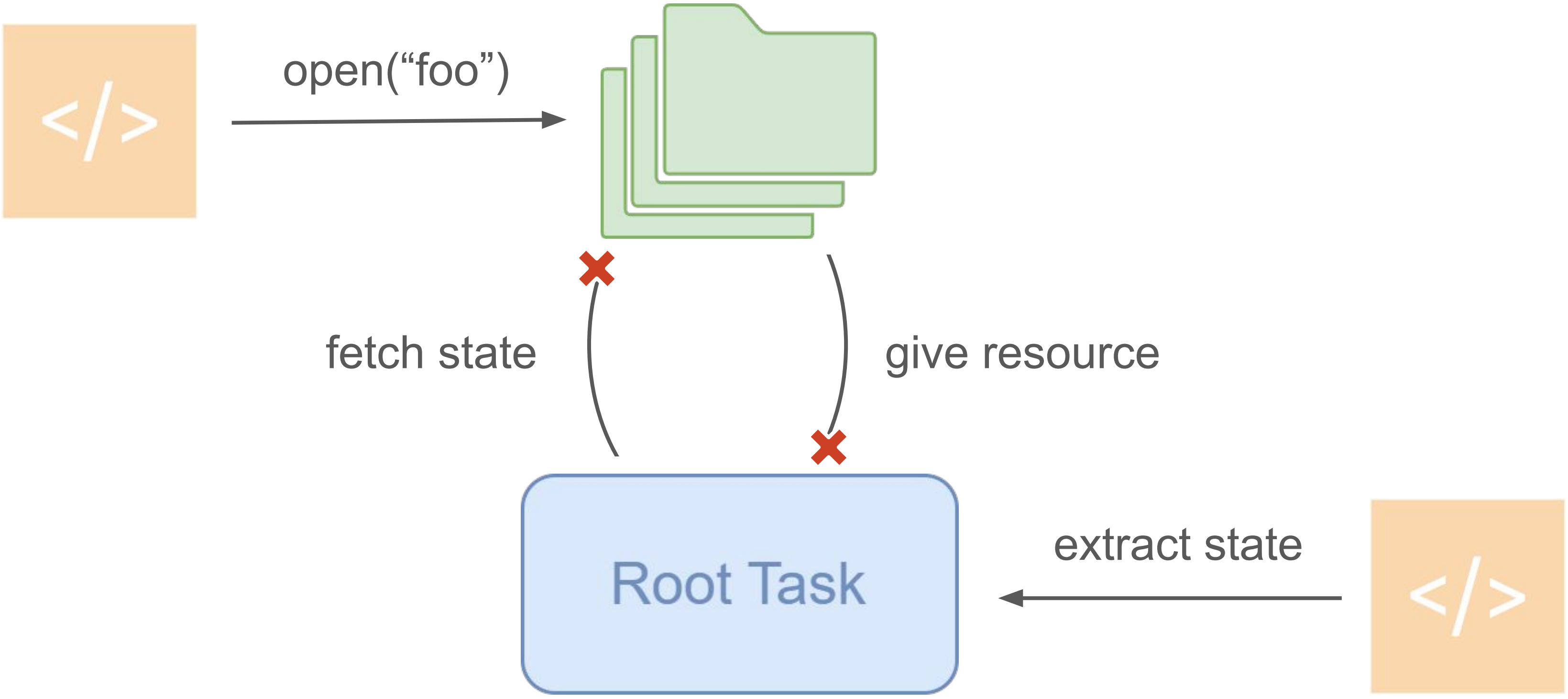


open("foo")

Root Task

extract state

An Interesting Challenge: Communication & Deadlock



open("foo")

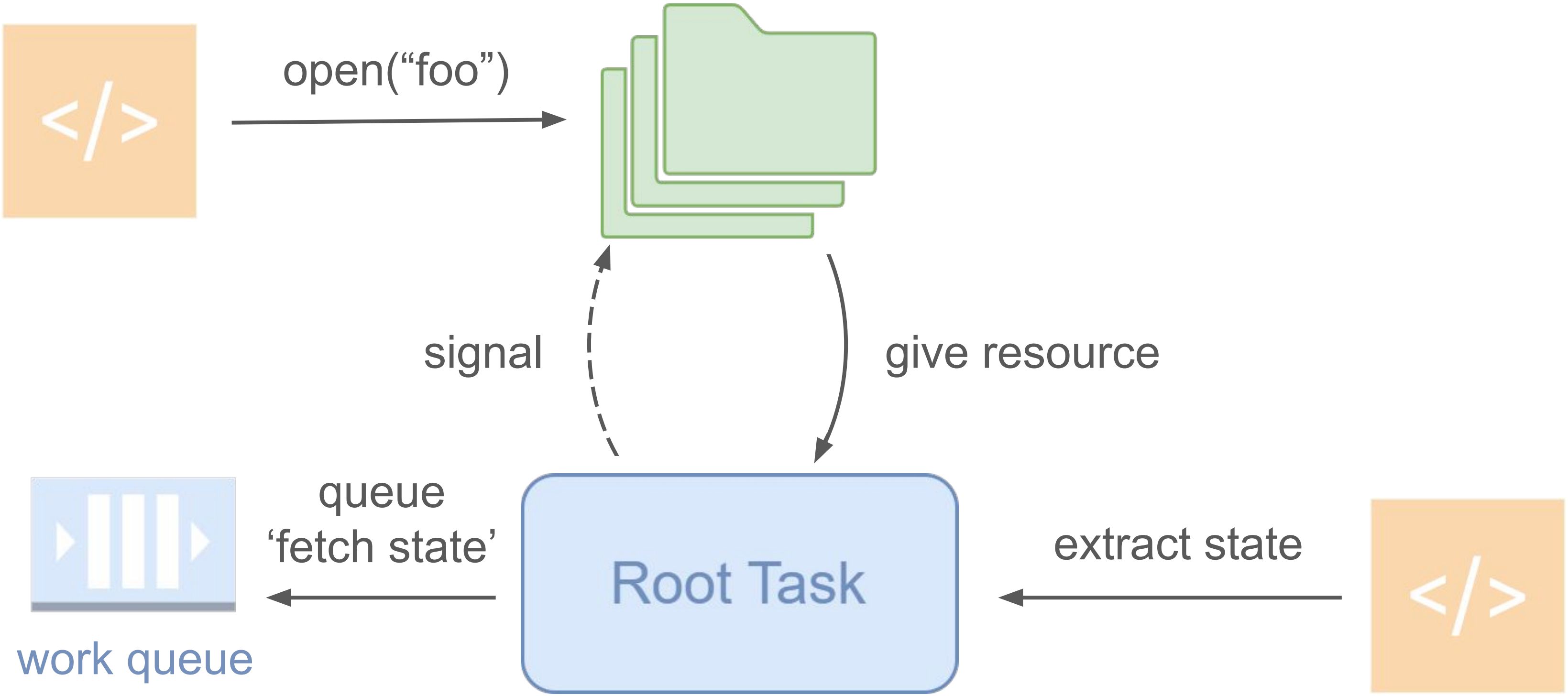
fetch state

give resource

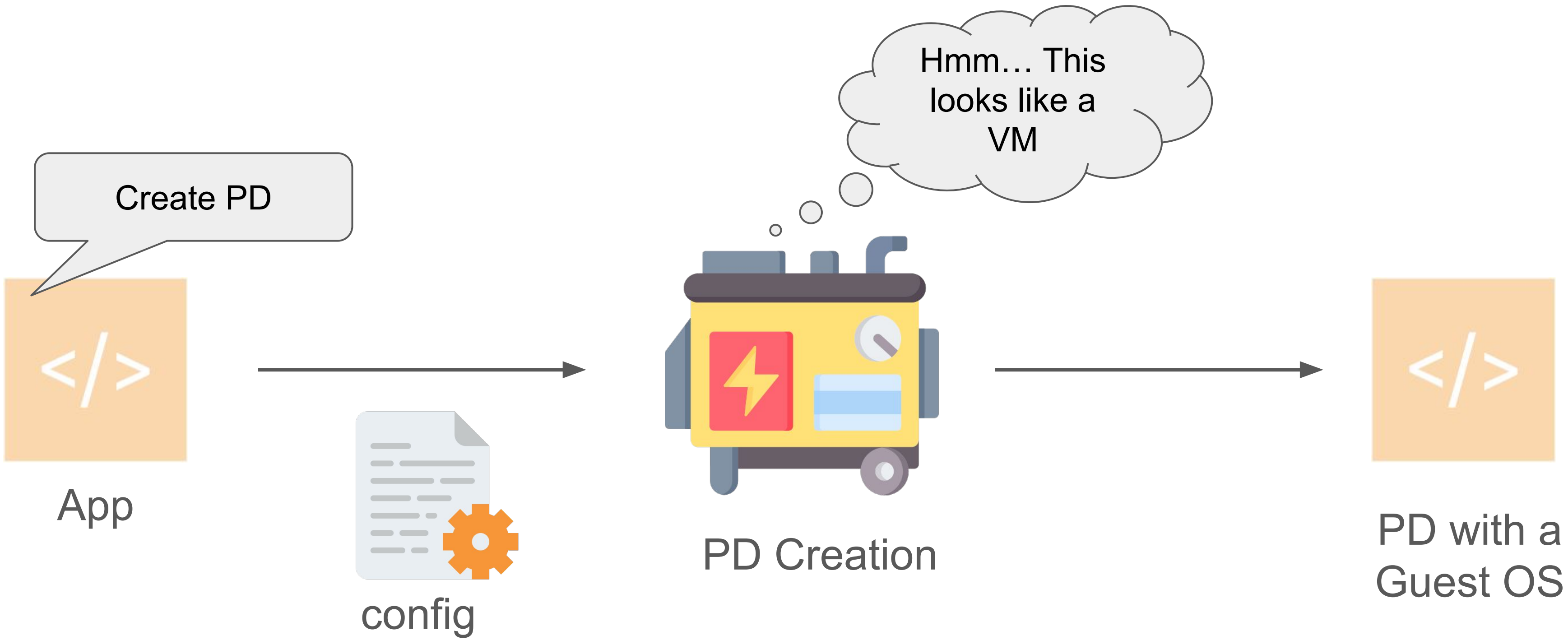
Root Task

extract state

An Interesting Challenge: Communication & Deadlock



An Interesting Challenge: VMM



Create PD

App

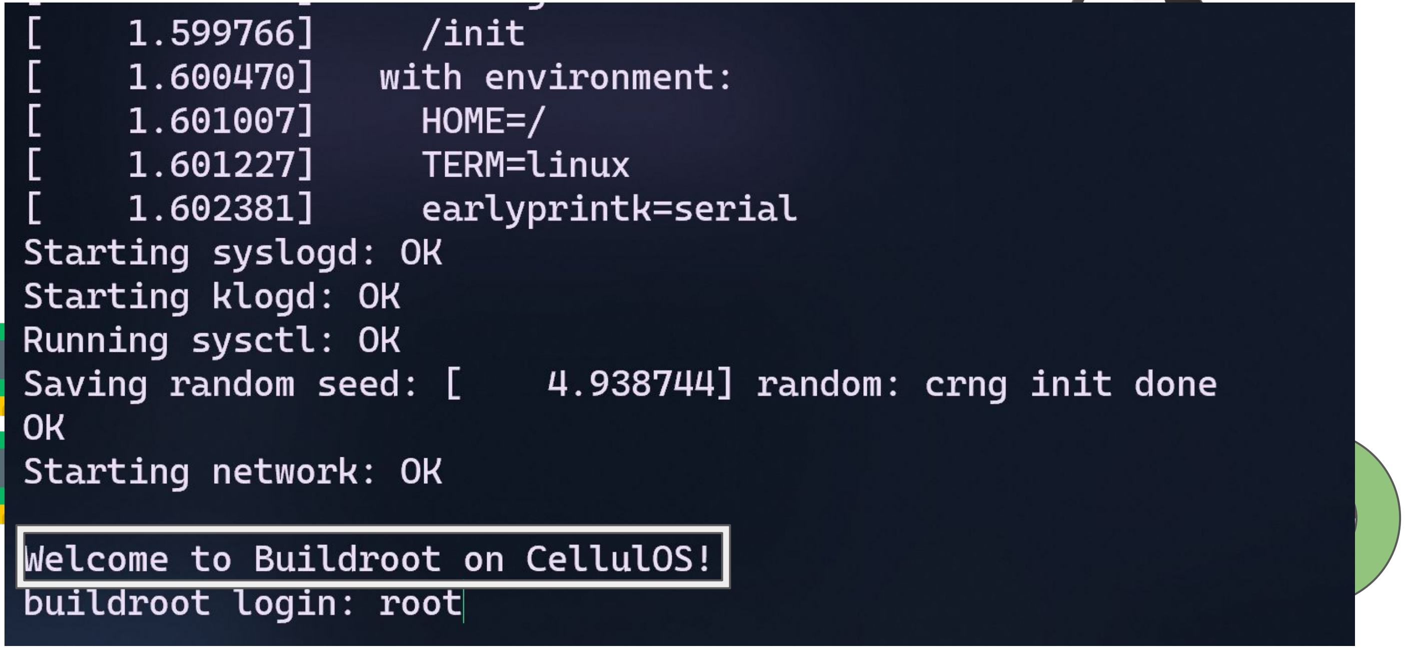
config

Hmm… This looks like a VM

PD Creation

PD with a Guest OS

An Frustrating Challenge: VMM



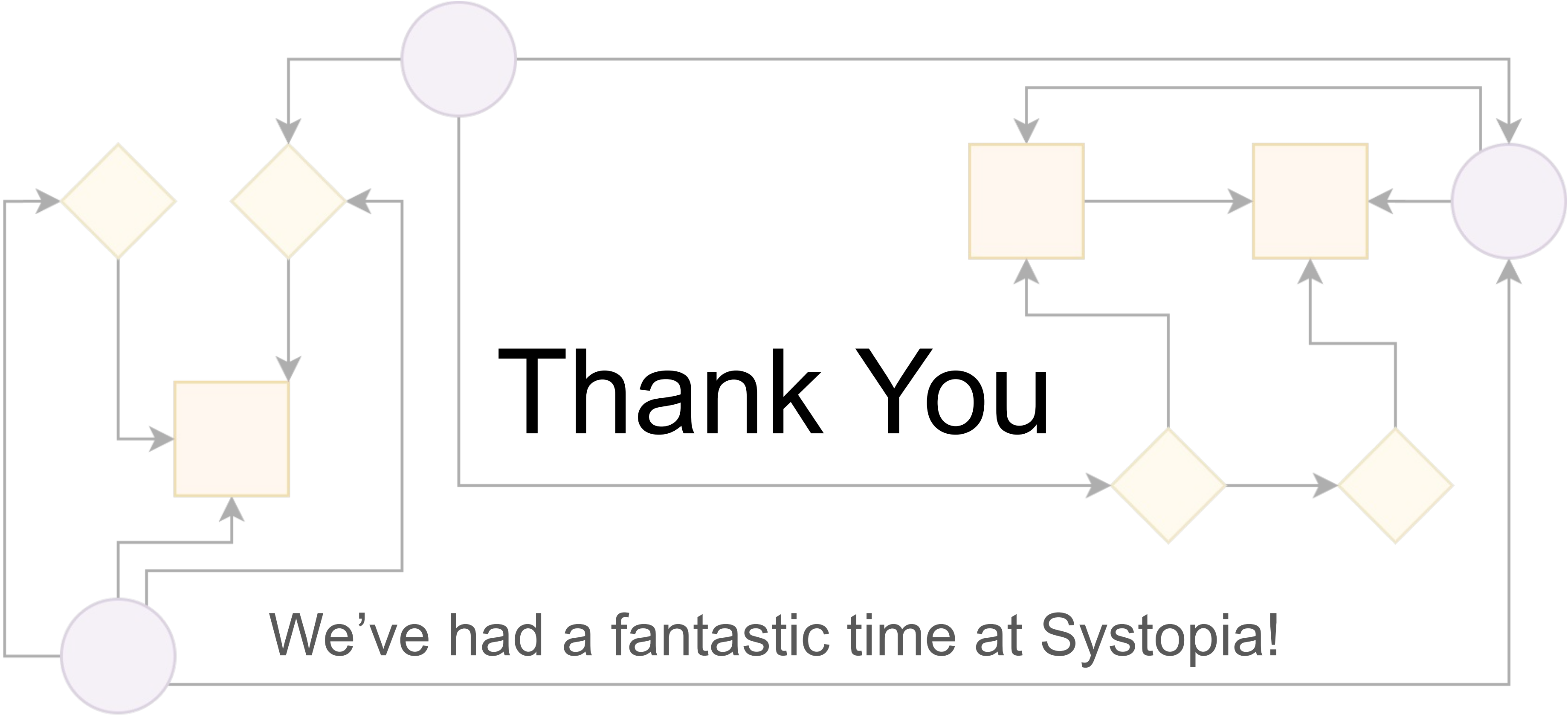
```
[    1.599766]     /init
[    1.600470]   with environment:
[    1.601007]     HOME=/
[    1.601227]     TERM=linux
[    1.602381]     earlyprintk=serial
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
Saving random seed: [    4.938744] random: crng init done
OK
Starting network: OK

Welcome to Buildroot on CellulOS!
buildroot login: root
```

Other Things We Did

- Implementing the Root Task
- RPC mechanism
- Resource cleanup
  - Flexible cleanup policies
- CellulOS model state workflow
  - Neo4j scripts & docker container
  - Metrics calculations
- Model state extraction from Linux /proc
- Wiki
- Lines of code:
  - C: 58k
  - Python: 2k

Programming for a Microkernel

Programming for Linux



Thank You

We’ve had a fantastic time at Systopia!

Wiki: https://cellulosdocs.readthedocs.io/en/cellulos/