
KubeCon
CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022
TiKV
TiKV
HELM

Armoring Cloud Native Workloads with LSM Superpowers

KubeCon CloudNativeCon North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022

Barun Acharya
Software Engineer
Accuknox

October 24-28, 2021

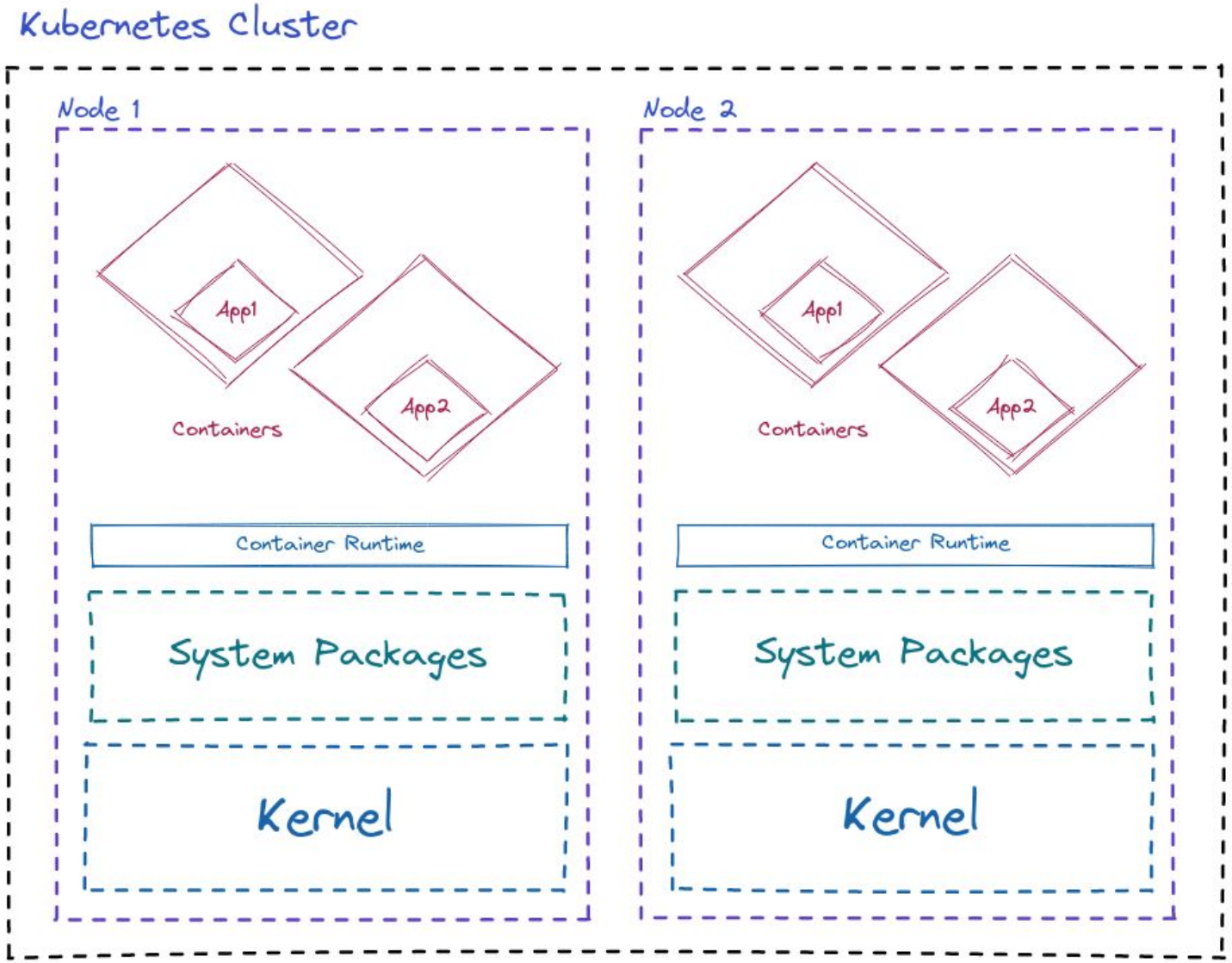
BUILDING FOR THE ROAD AHEAD

DETROIT 2022

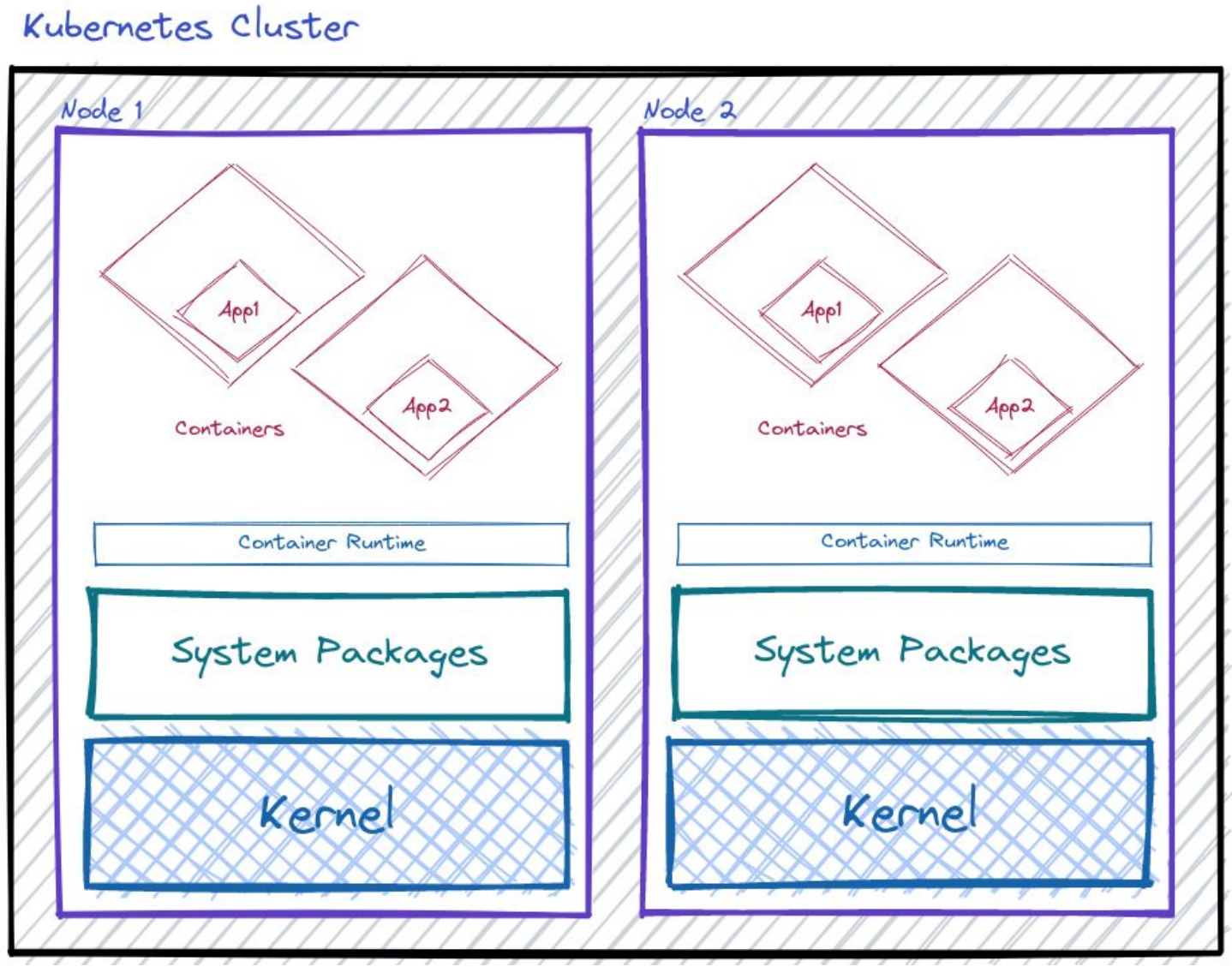
Armoring Cloud Native Workloads with LSM Superpowers

Barun Acharya, Accuknox

Cloud Native Workloads



Cloud Native Security

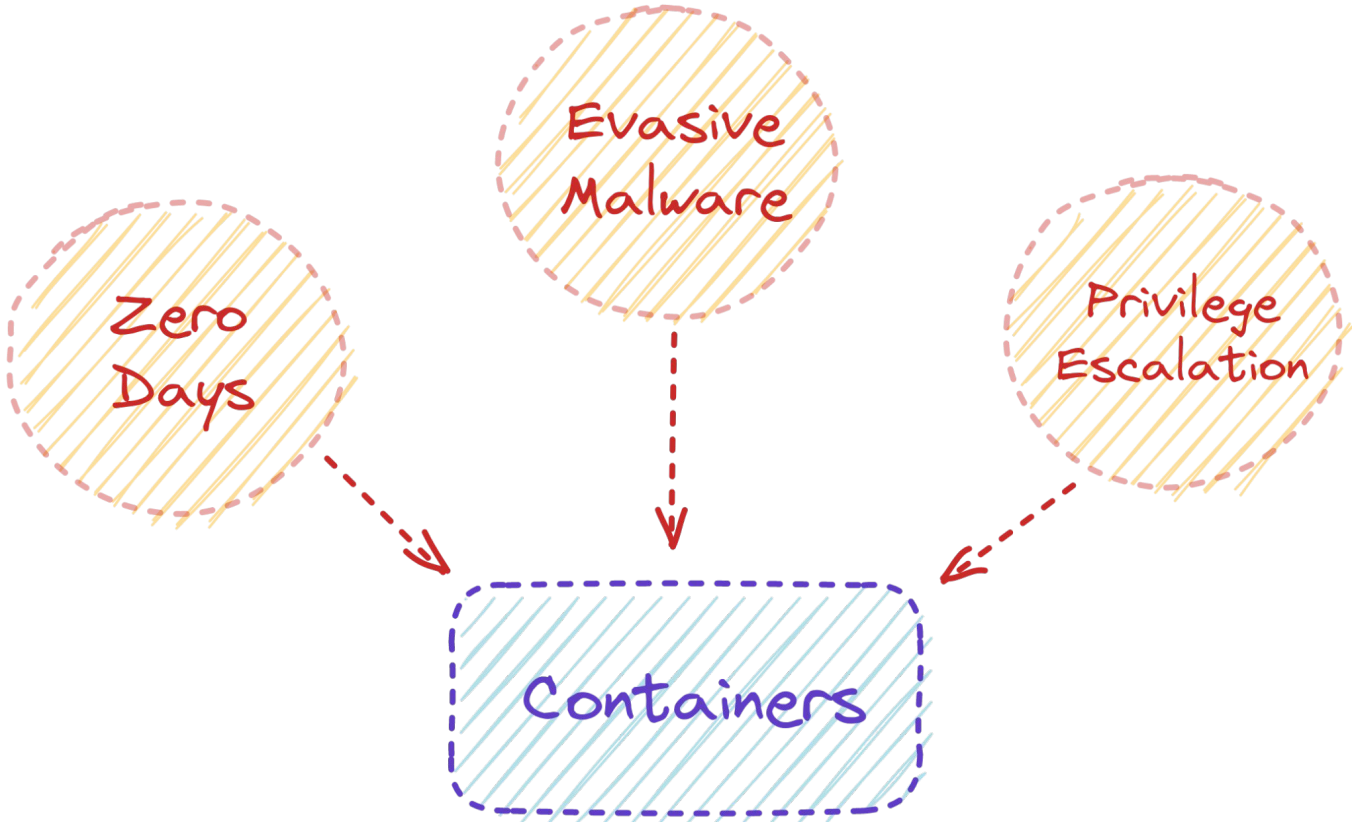


BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Container Security

Runtime Threats



How do we deal with vulnerabilities that manifest at Runtime?

We deal with them at Runtime

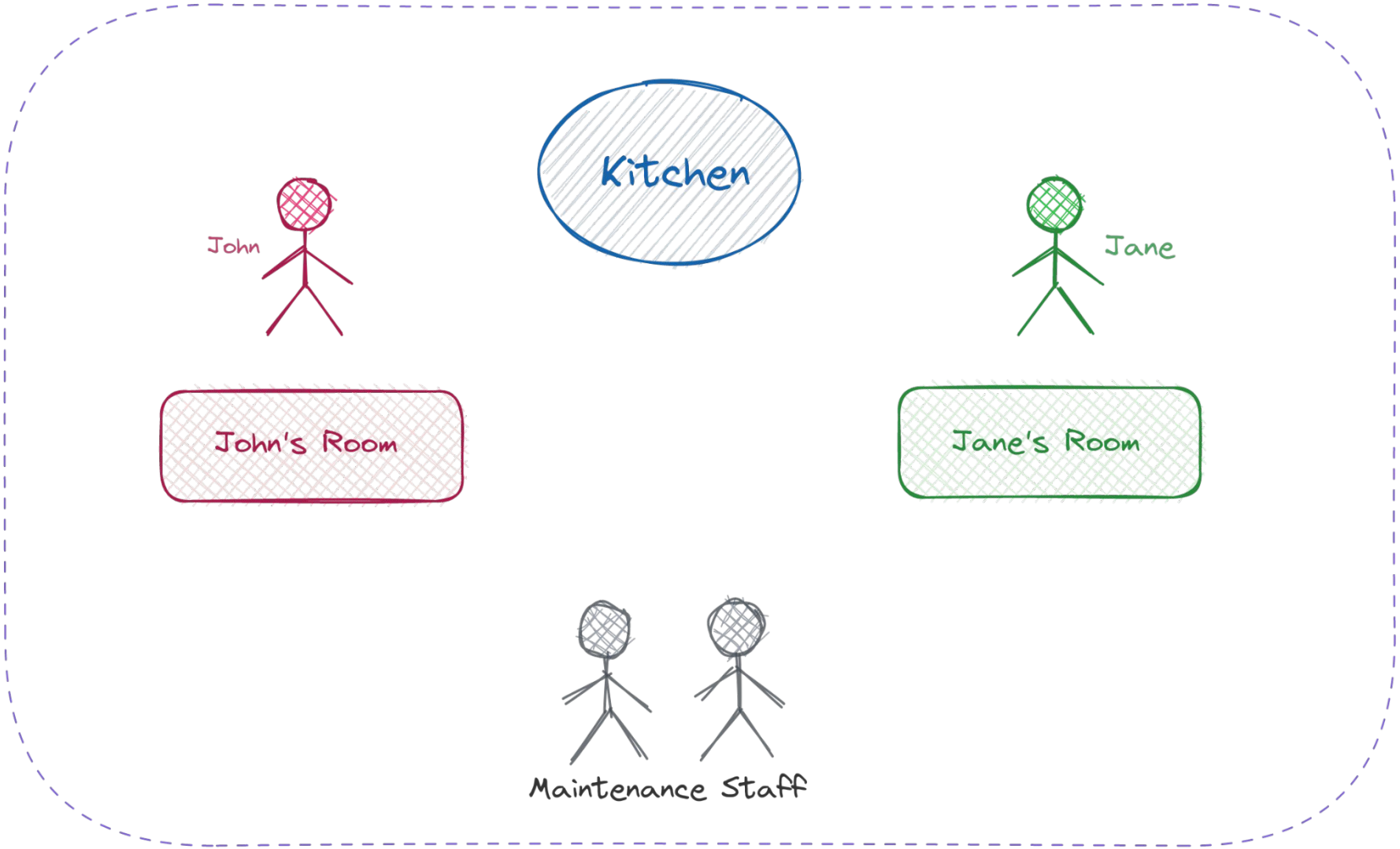
BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Container Runtime Security

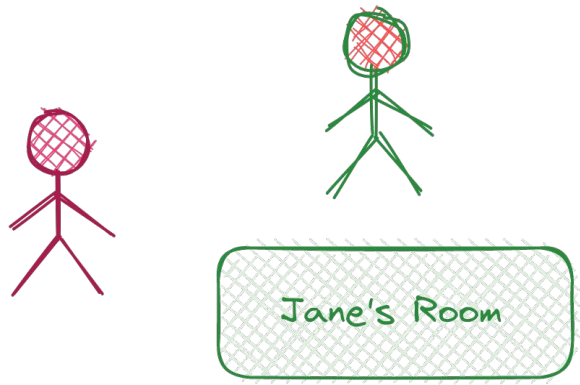
KubeCon | CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022

Let's imagine Containers as a Private Apartment

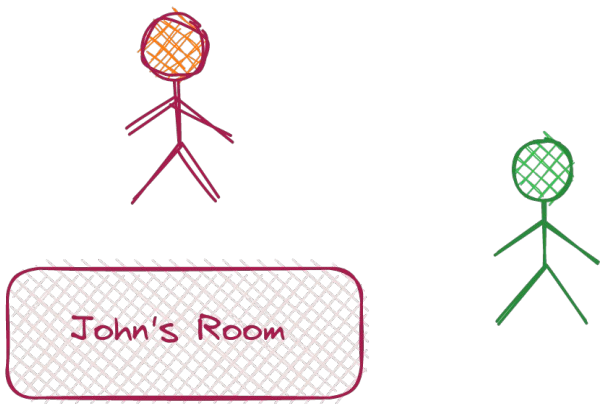

Kitchen
John
Jane
John's Room
Jane's Room
Maintenance Staff

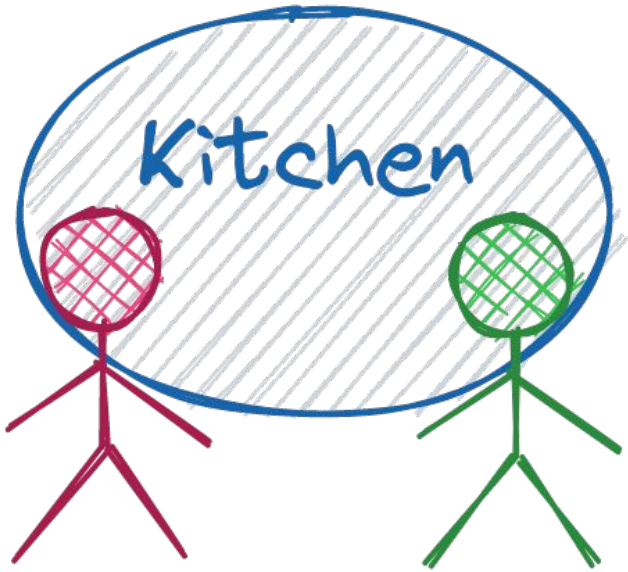
John and Jane value their Privacy

Jane doesn't like if John pokes into her Room

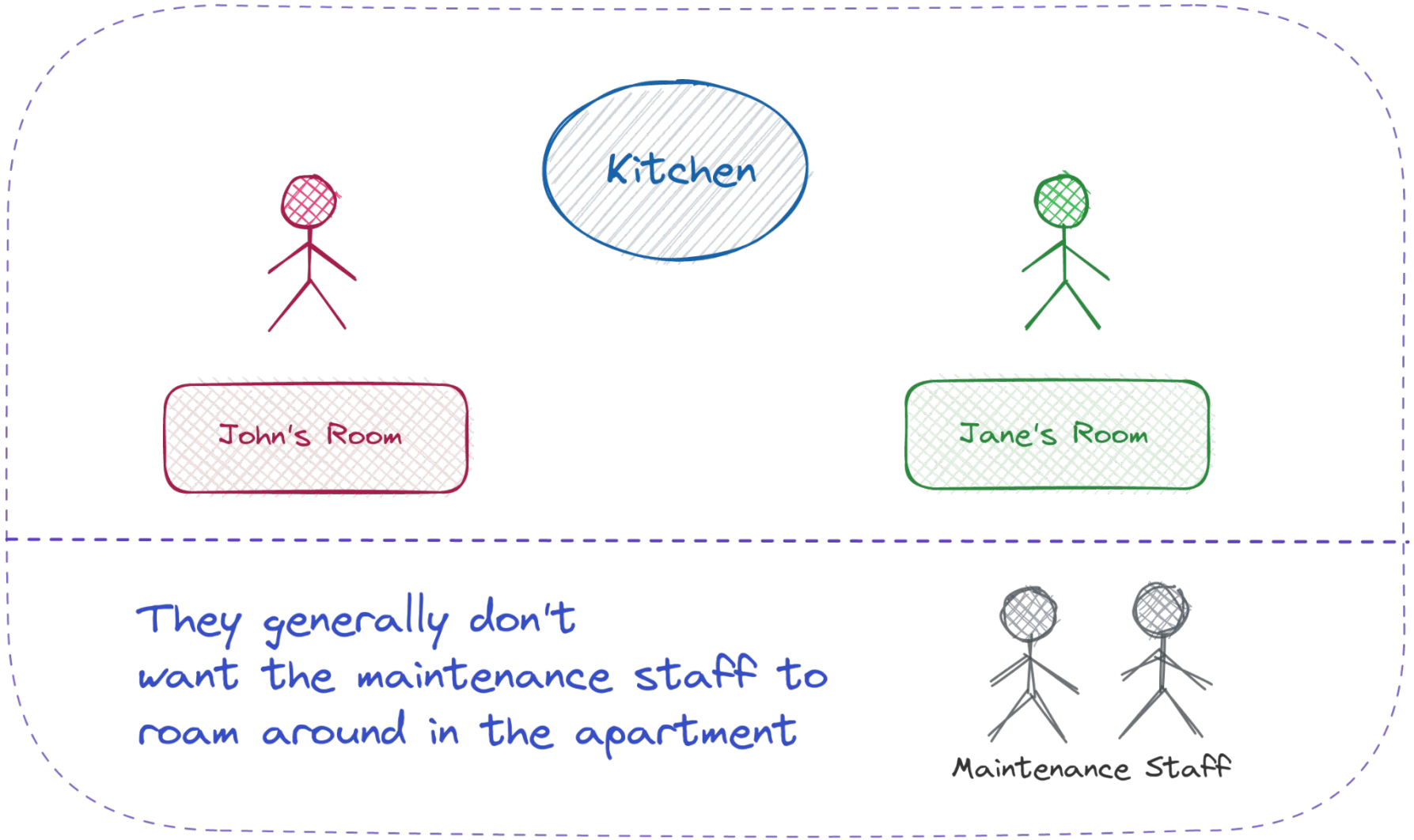


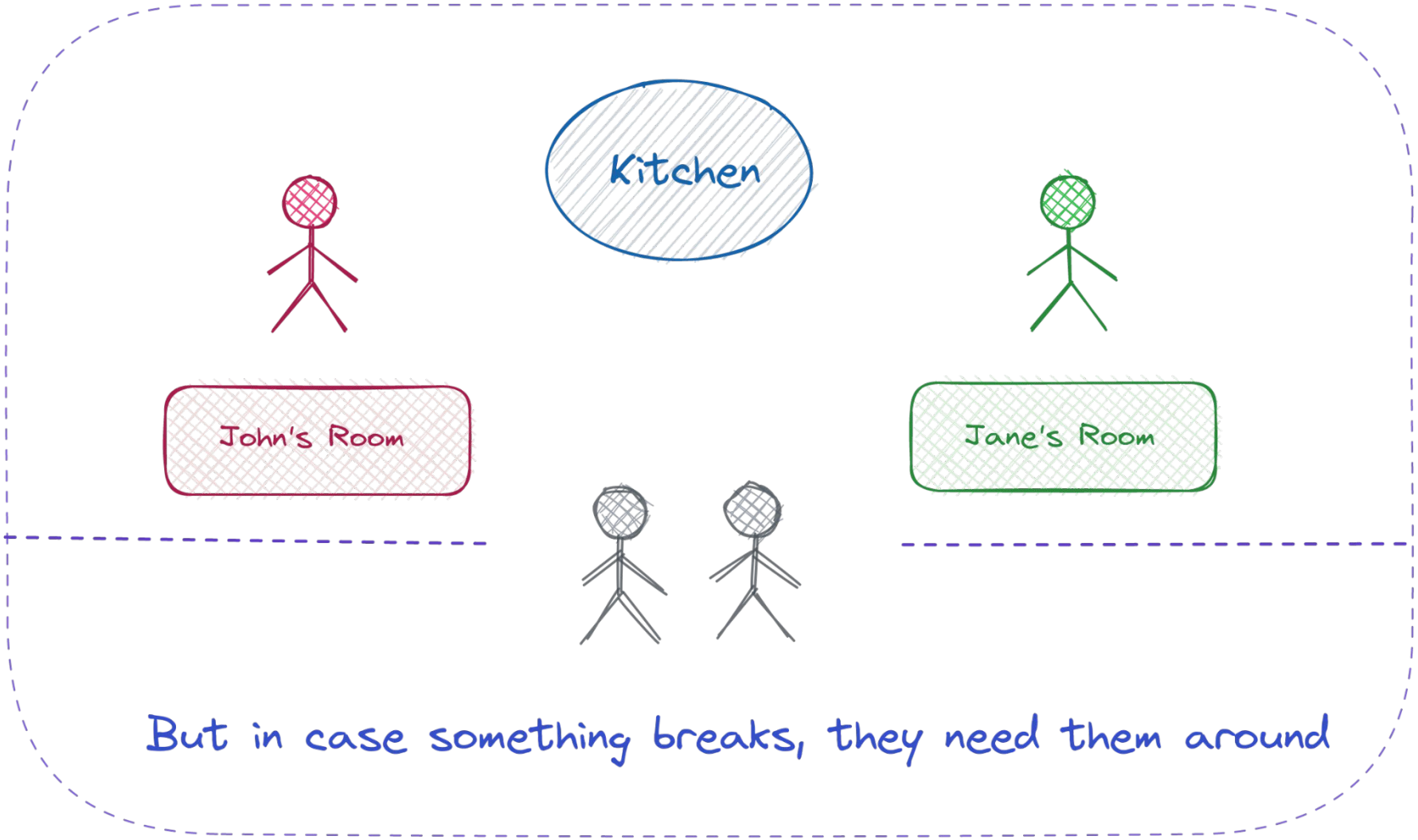
Neither does John if Jane does the same

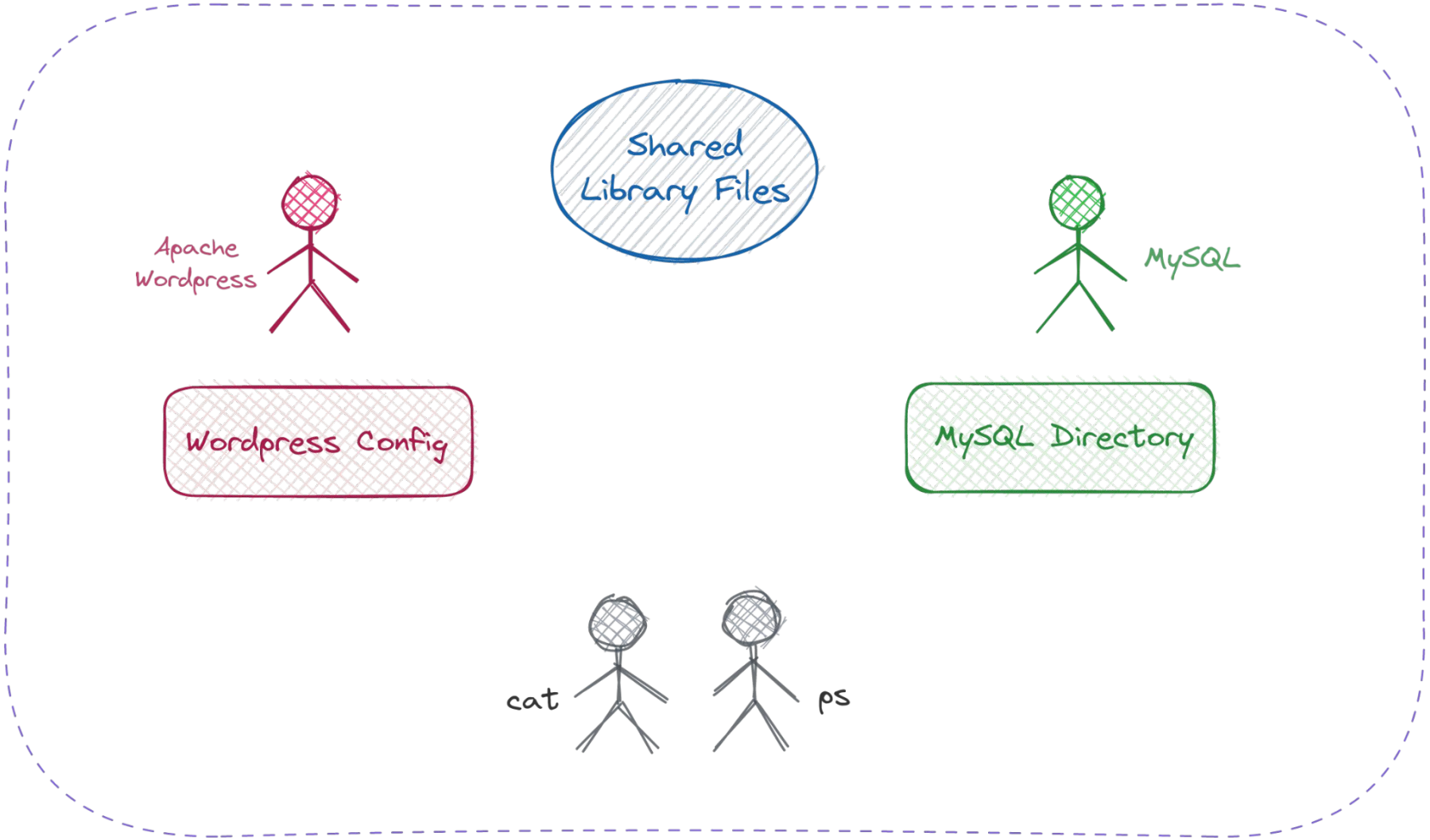


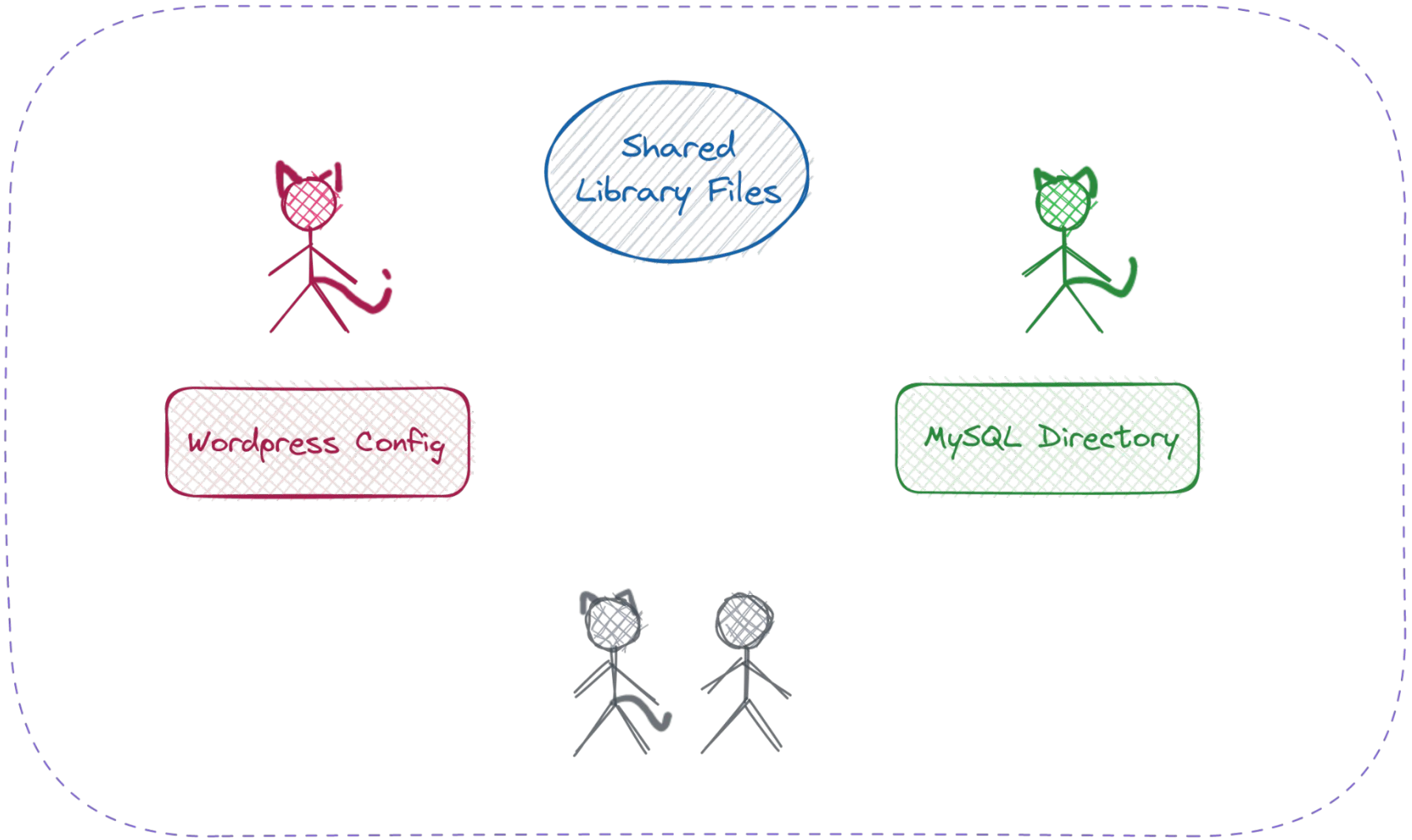


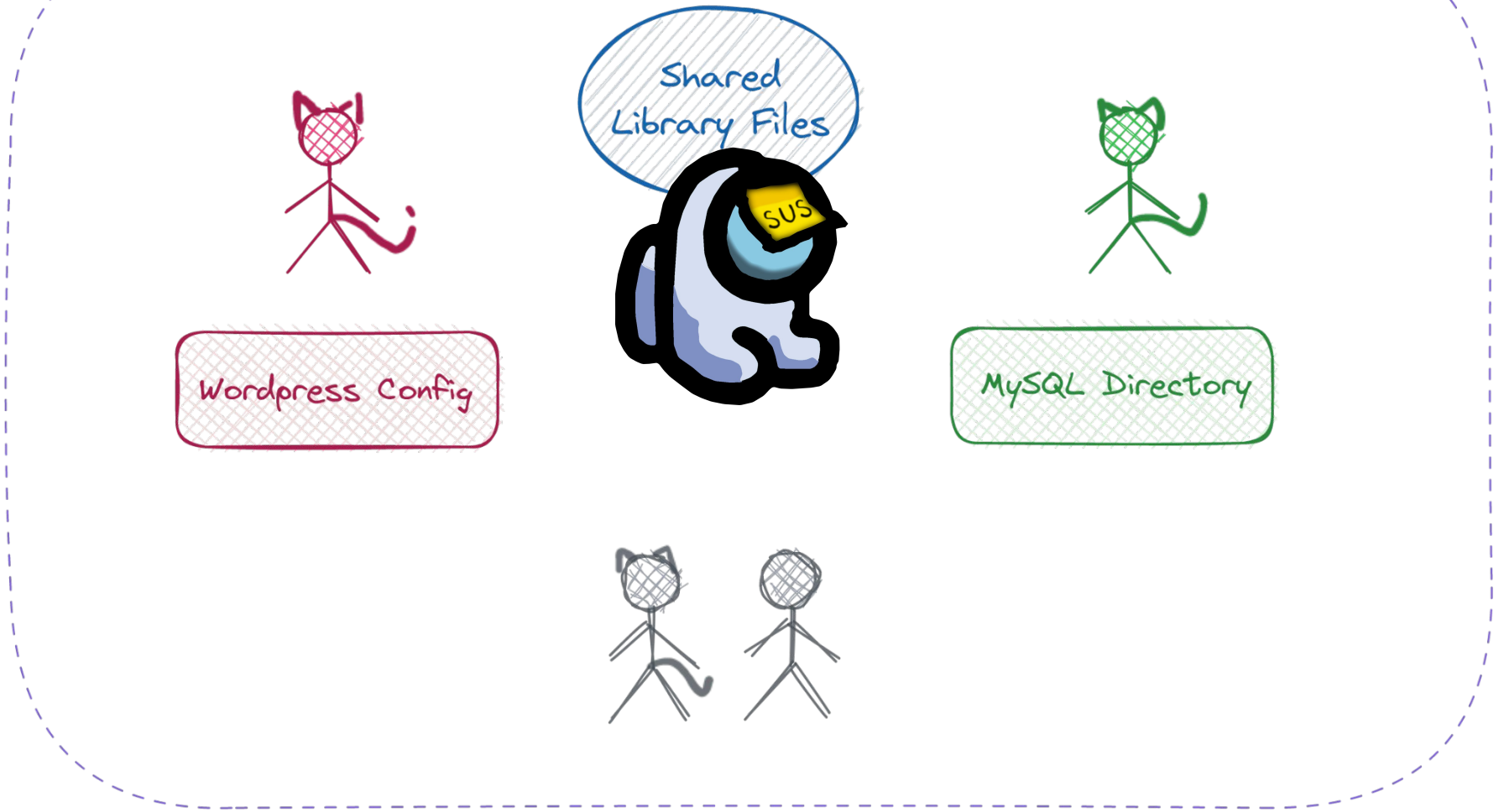
But they both like to hang around in the Kitchen


Kitchen
John's Room
Jane's Room
They generally don't want the maintenance staff to roam around in the apartment
Maintenance Staff


Kitchen
John's Room
Jane's Room
But in case something breaks, they need them around


Shared Library Files
Apache Wordpress
MySQL
Wordpress Config
MySQL Directory
cat
ps


Shared Library Files
Wordpress Config
MySQL Directory

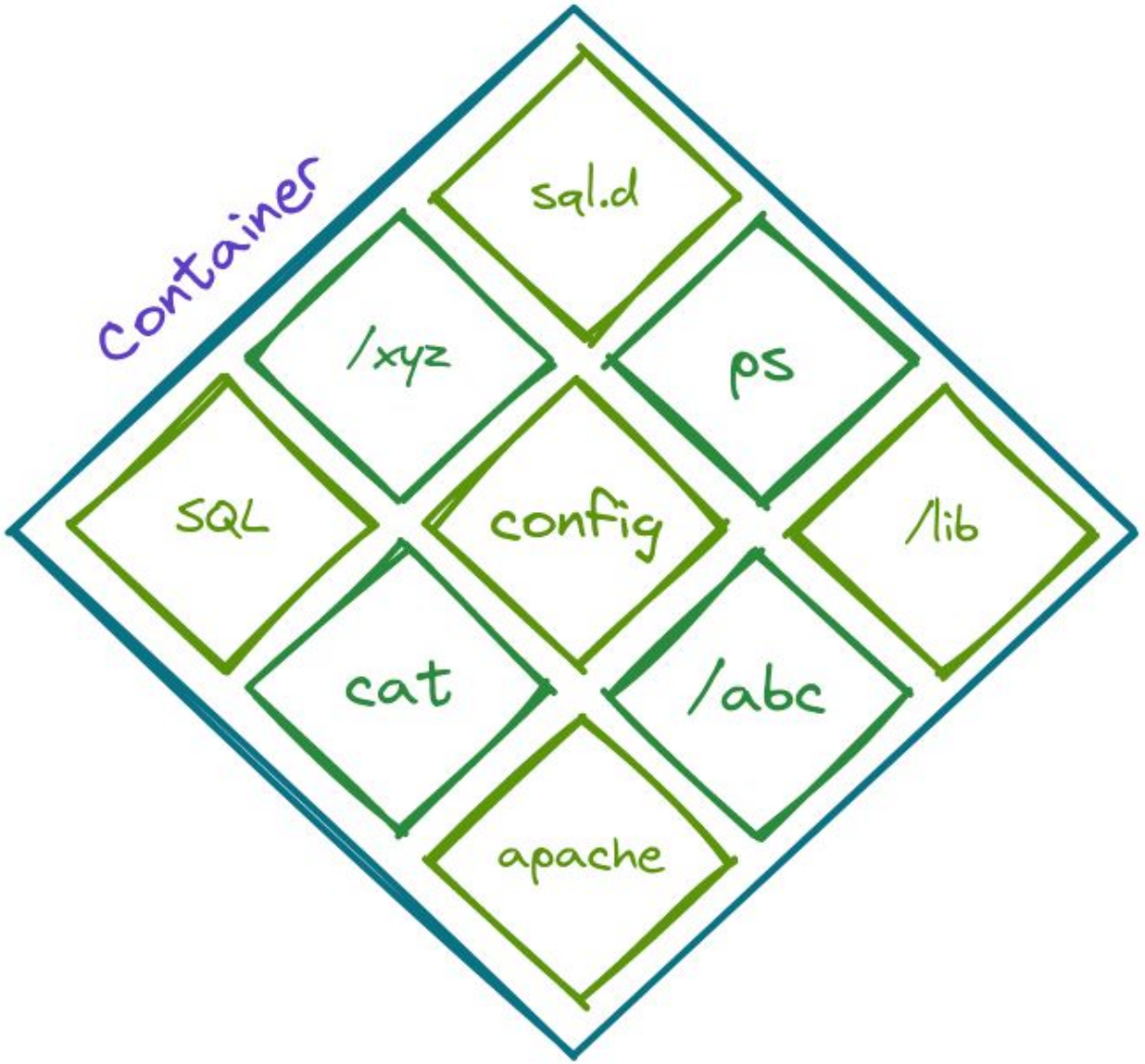

Shared Library Files
SUS
Wordpress Config
MySQL Directory

BUILDING FOR THE ROAD AHEAD

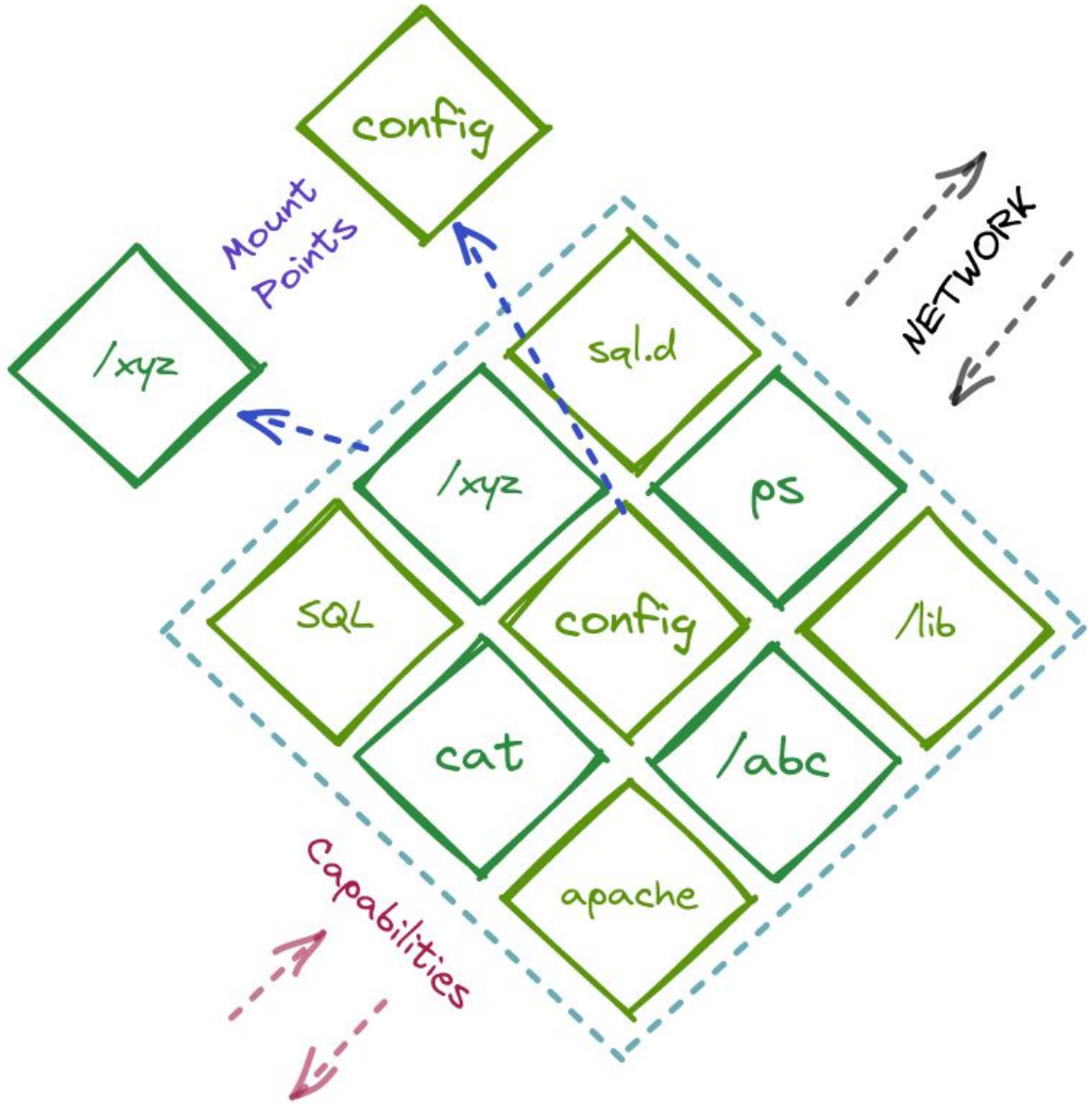
DETROIT 2022

Container Runtime Security

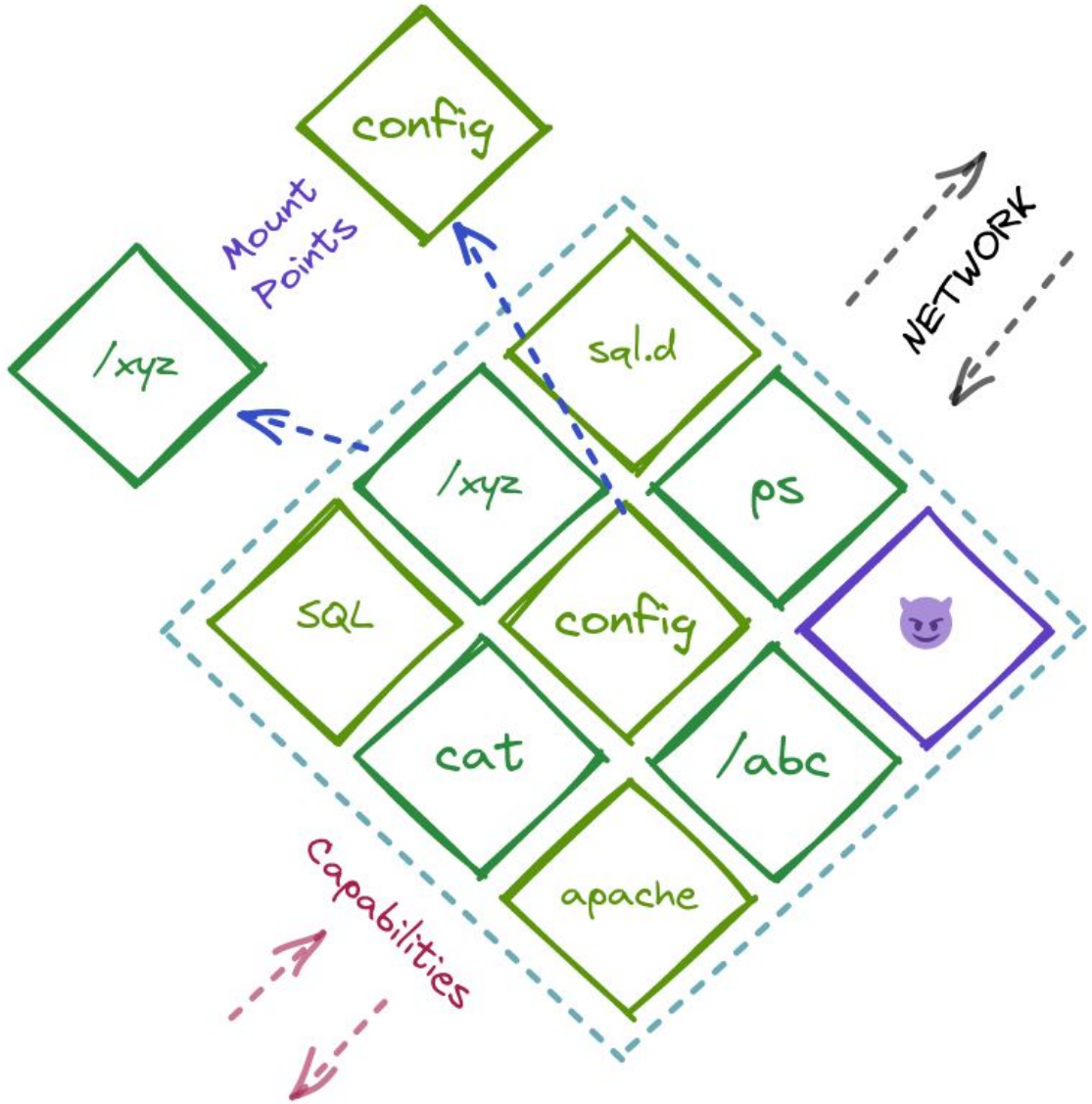
Peeking into the Container



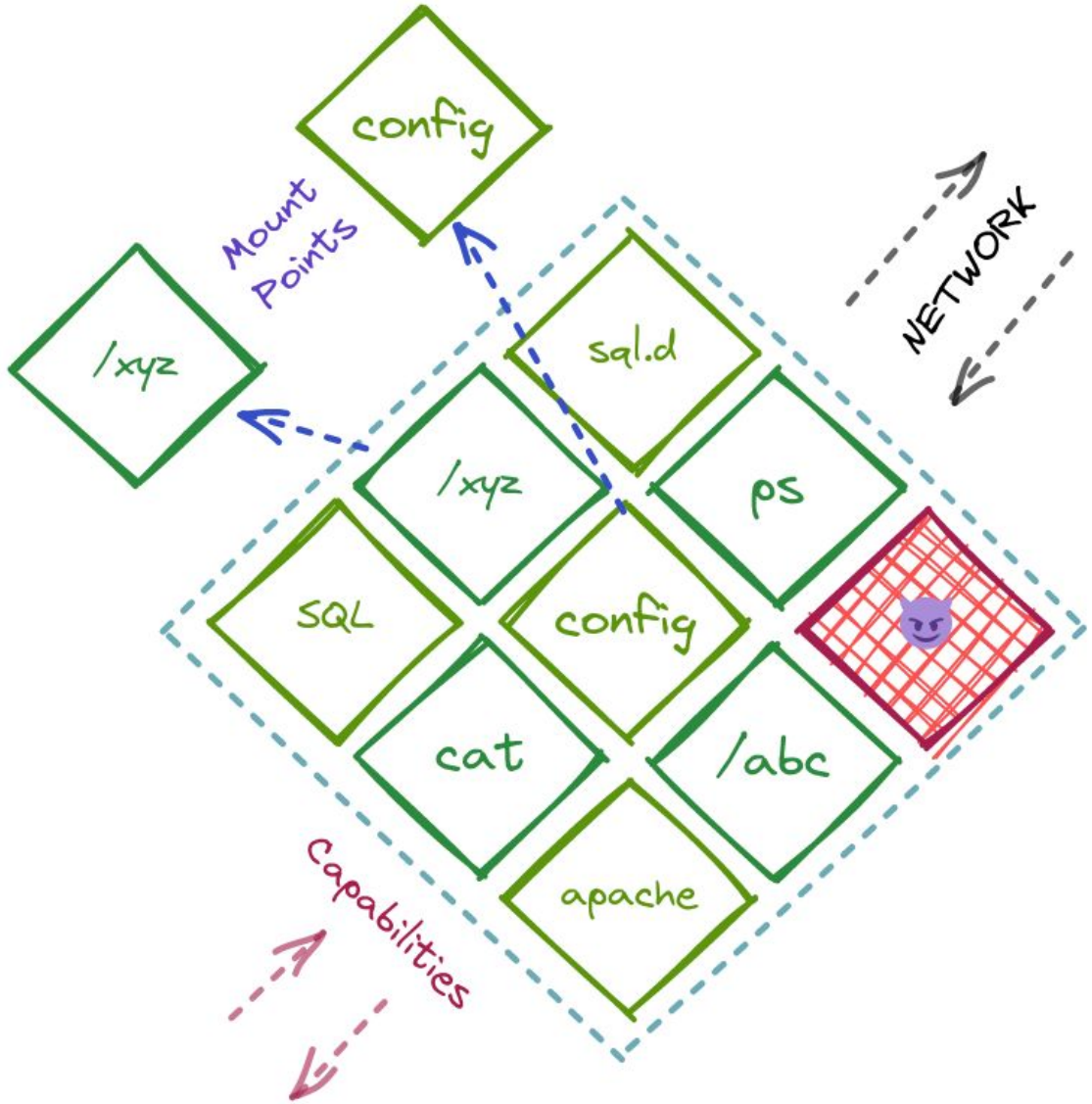
Peeking into the Container



Peeking into the Container



Need for Access Control inside Container

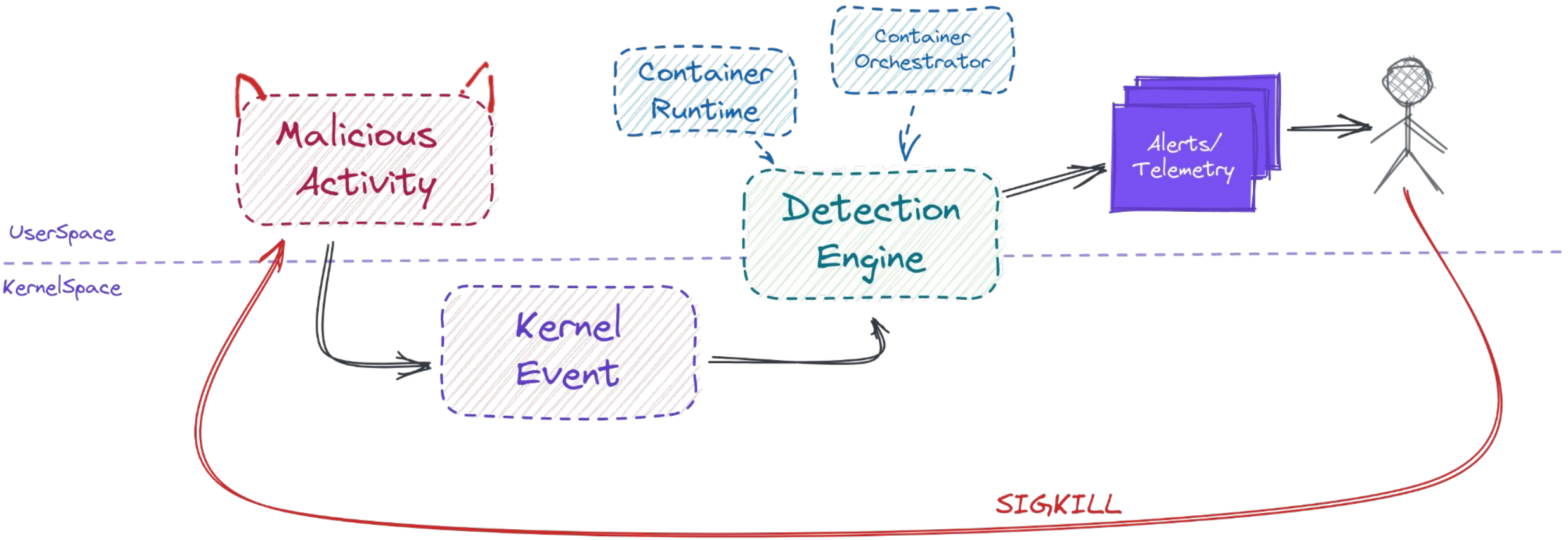


Runtime Enforcement

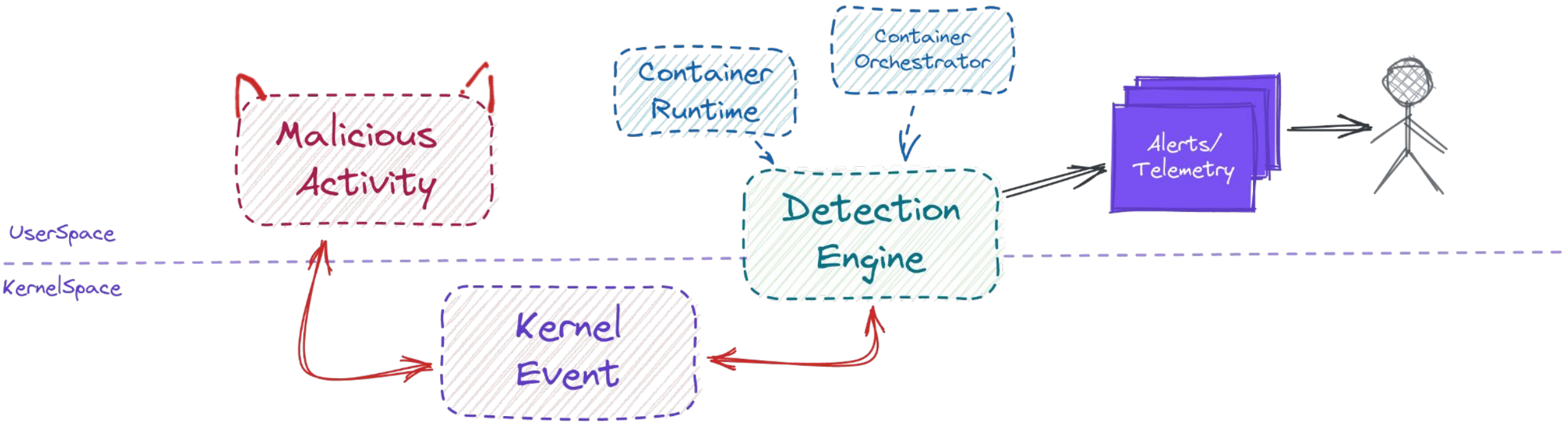
- ❑ **Detect** the Malicious Activity
- ❑ **Alert** the activity with context
- ❑ Take **Action** on the Activity

- ❑ **Detect** the Malicious Activity
- ❑ Take **Action** on the Activity
- ❑ **Alert** the activity with context

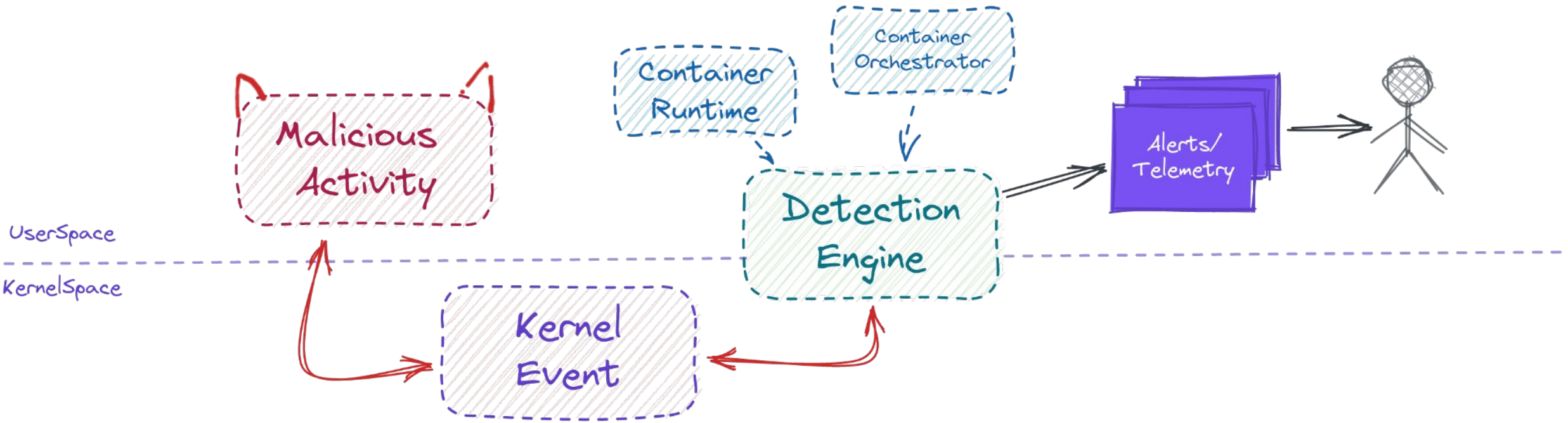
Async Enforcement



Inline Enforcement



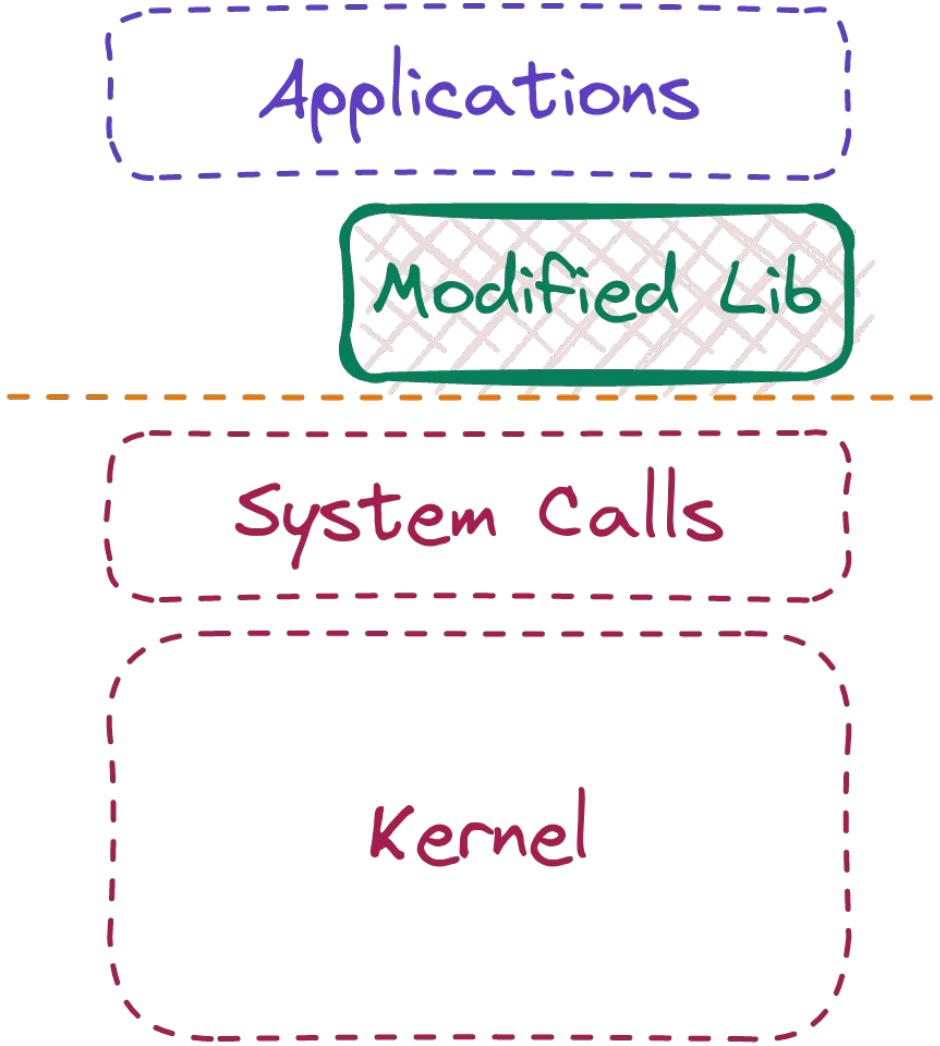
Inline Enforcement



Linux Enforcement Primitives

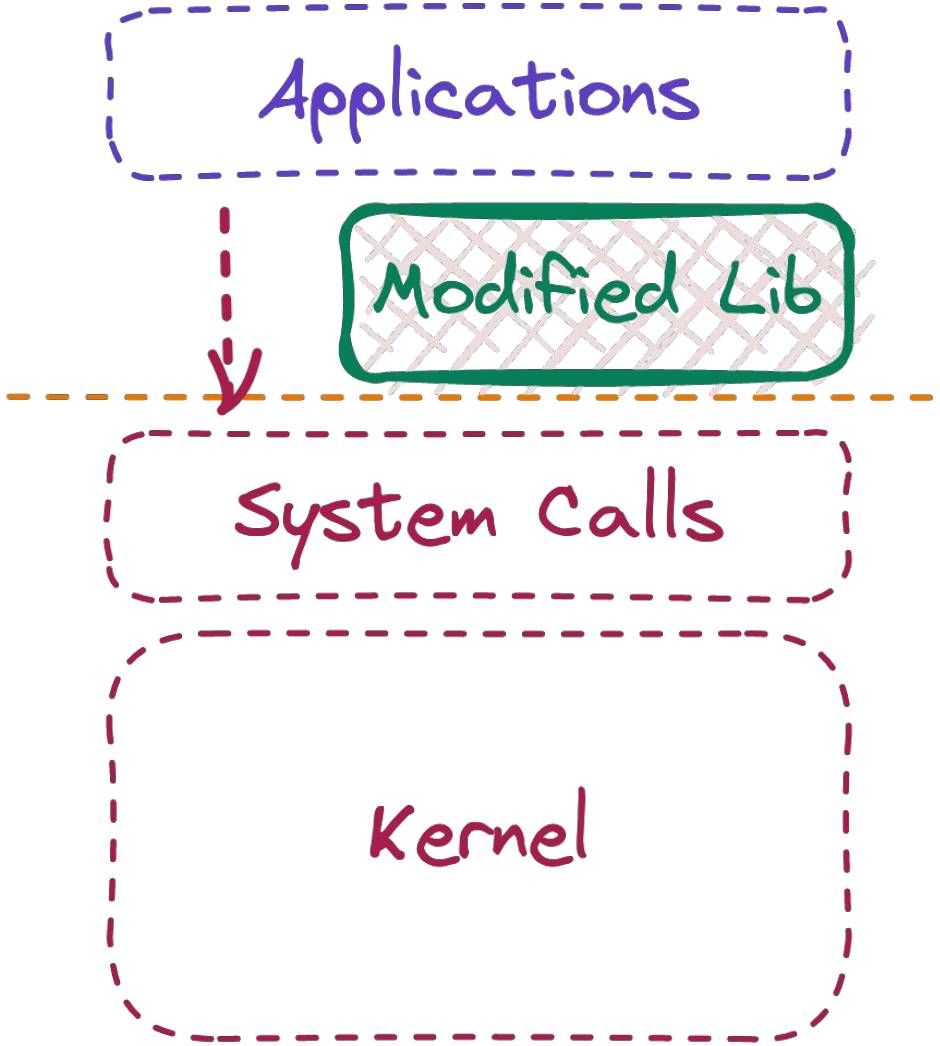
- LD_PRELOAD
- Seccomp (Secure Computing)
- Linux Security Modules

LD_PRELOAD



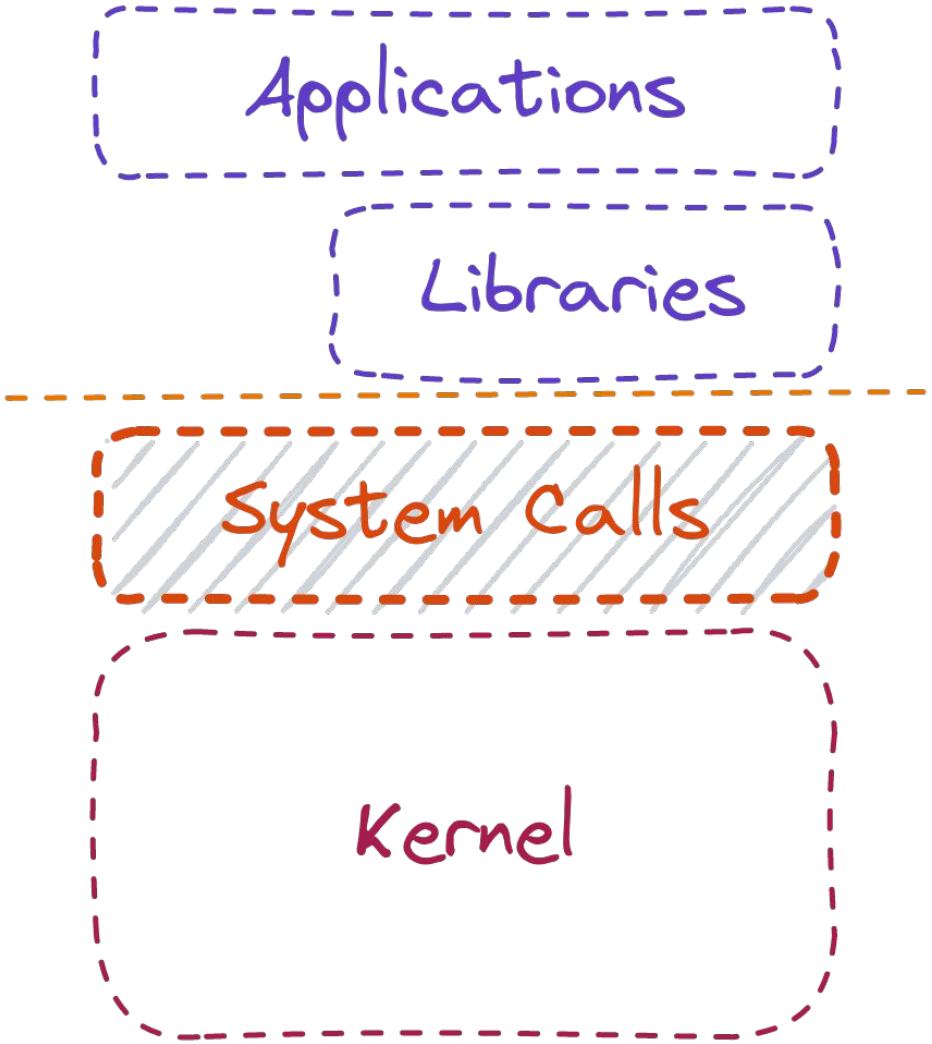
- Userspace Control
- Overrides Dynamic Libraries

LD_PRELOAD



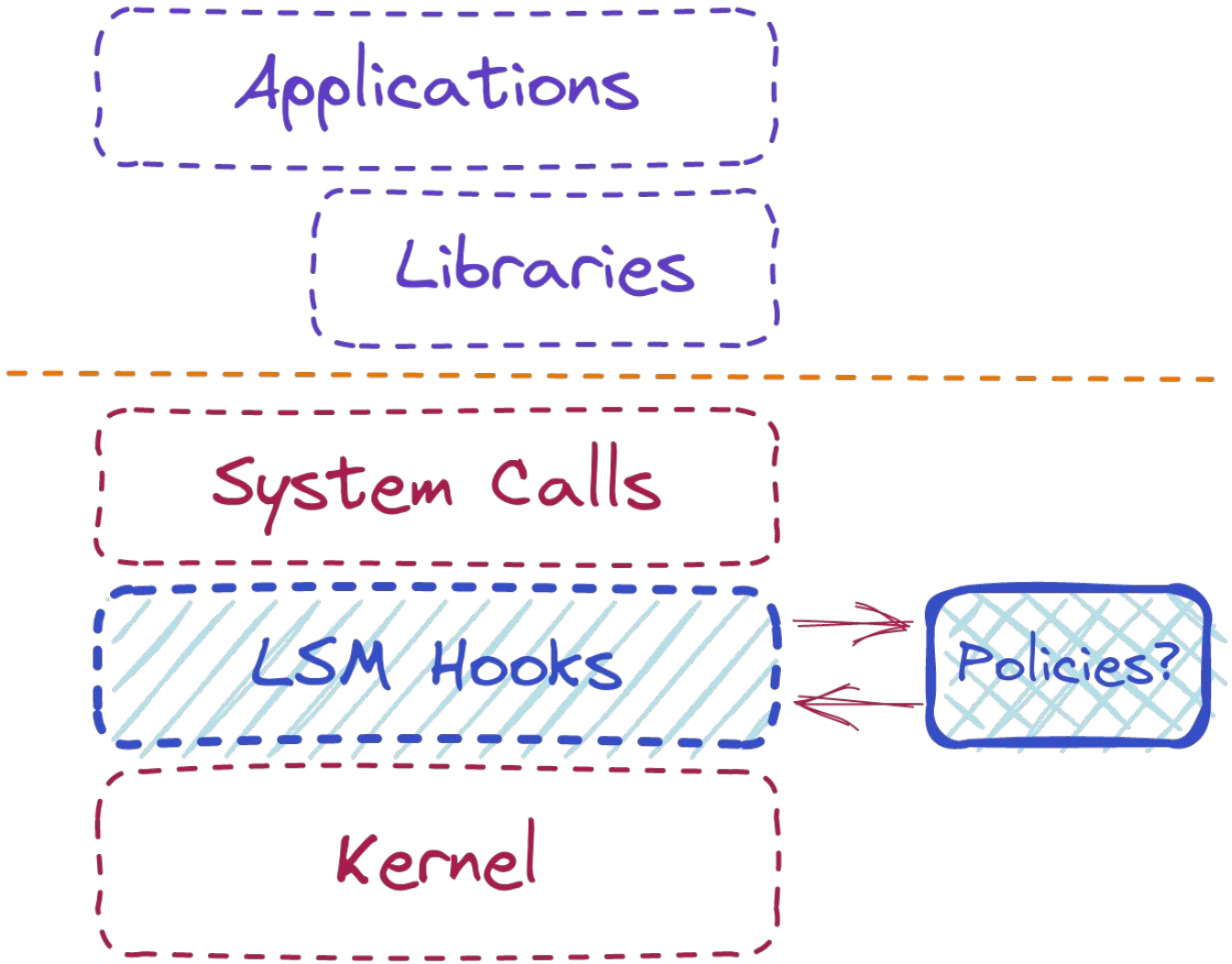
- Userspace Control
- Overrides Dynamic Libraries
- Attacker can invoke syscalls without going through libc at all

Seccomp



- Sandbox
- Docker Support
- Dynamic Enforcement not possible
- Limited Filtering

Linux Security Modules

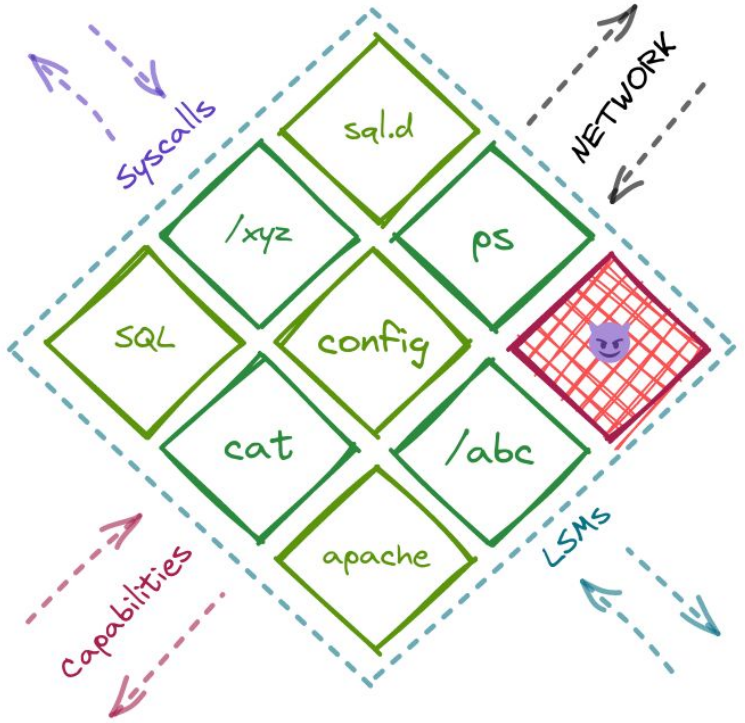


- AppArmor, SELinux, BPF LSM …
- Mature ecosystem of 20+ years
- Does not suffer from TOCTOU
- Not integrated with modern context
- Steep learning curve for policy language

How do we integrate these security primitives with **Cloud Native Workloads?**

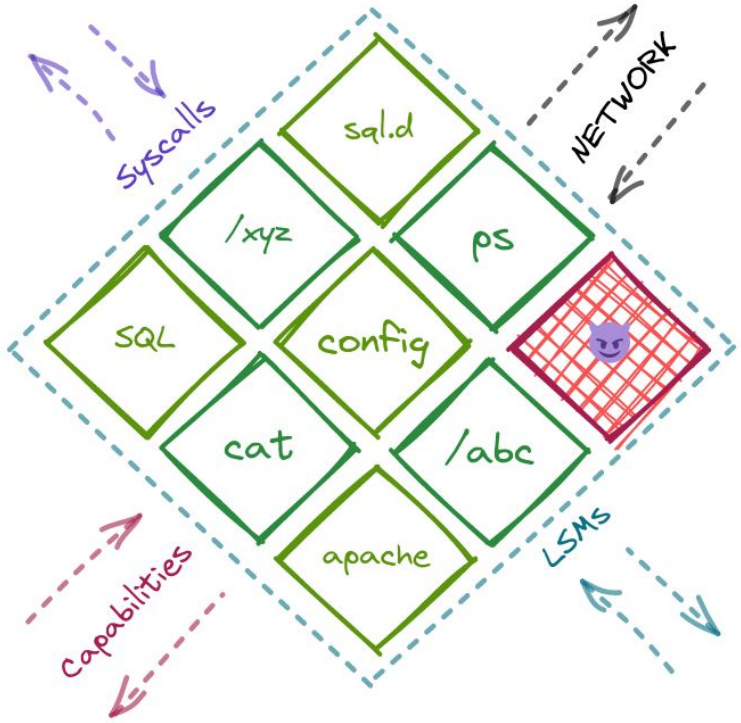
Pod Security Context

A security context defines privilege and access control settings for a Pod or Container. Security context settings include, but are not limited to Seccomp, Capabilities, AppArmor, SElinux…



Pod Security Context

A security context defines privilege and access control settings for a Pod or Container. Security context settings include, but are not limited to Seccomp, Capabilities, AppArmor, SElinux…



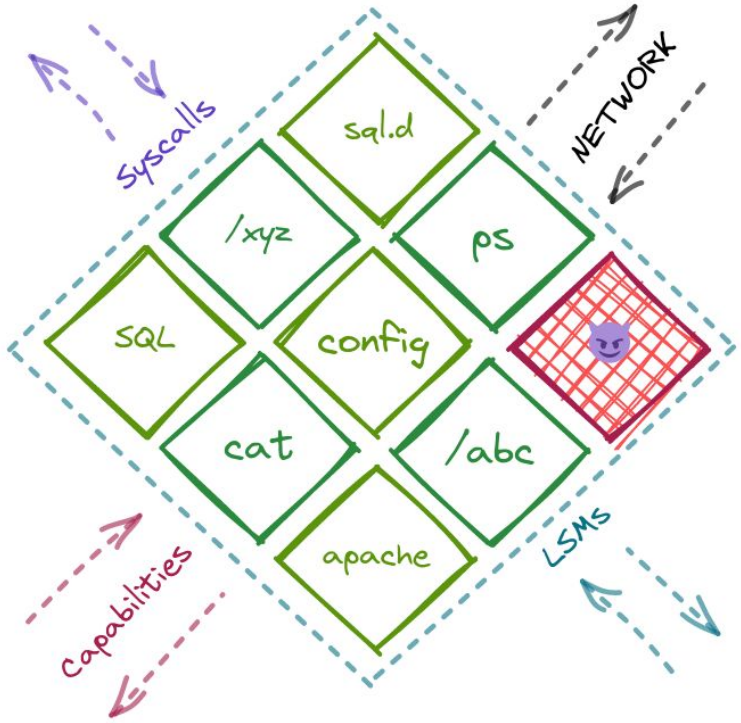

```yaml
securityContext:
  seccompProfile:
    type: Localhost
    localhostProfile: profiles/fine-grained.json
```

```yaml
securityContext:
  capabilities:
    add: ["NET_ADMIN", "SYS_TIME"]
```

```yaml
securityContext:
  seLinuxOptions:
    level: "s0:c123,c456"
```

Pod Security Context

A security context defines privilege and access control settings for a Pod or Container. Security context settings include, but are not limited to Seccomp, Capabilities, AppArmor, SElinux…




```yaml
securityContext:
  seccompProfile:
    type: Localhost
    localhostProfile: profiles/fine-grained.json
```

```yaml
securityContext:
  capabilities:
    add: ["NET_ADMIN", "SYS_TIME"]
```

```yaml
securityContext:
  seLinuxOptions:
    level: "s0:c123,c456"
```

They treat container as a single entity, and apply the security posture across the container

Pod Security Context

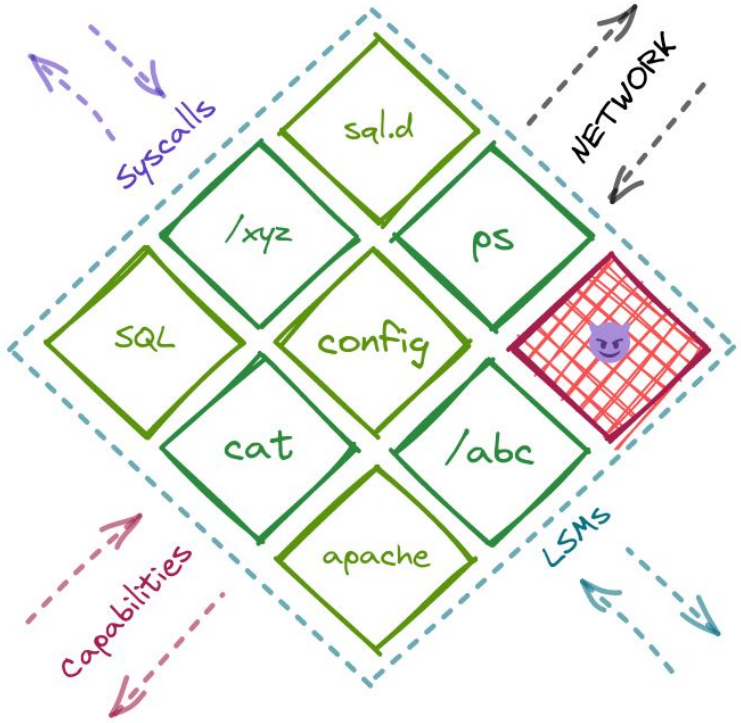
A security context defines privilege and access control settings for a Pod or Container. Security context settings include, but are not limited to Seccomp, Capabilities, AppArmor, SElinux…

```
container.apparmor.security.beta.kubernetes.io/<container_name>: <profile_ref>
```

```
→       /sys/**/ r,
→       /sys/devices/**/uevent r,
→       /sys/devices/system/**/meminfo r,
→       /sys/devices/pci[0-9]*/**/{config,revisio
→       /sys/devices/pci[0-9]*/**/{,subsystem_}{d

→       # Device access
→       /dev/ati/{,**} r,
→       /dev/dri/{,**} r,
→       /dev/tty rw,
→       audit deny /dev/{video,audio}* rwlkmx,
```

- Fine Grained Access Control
- Well Integrated with Modern Ecosystem
- Learning Curve
- AppArmor not available on all systems
- Audit Logs not enough context



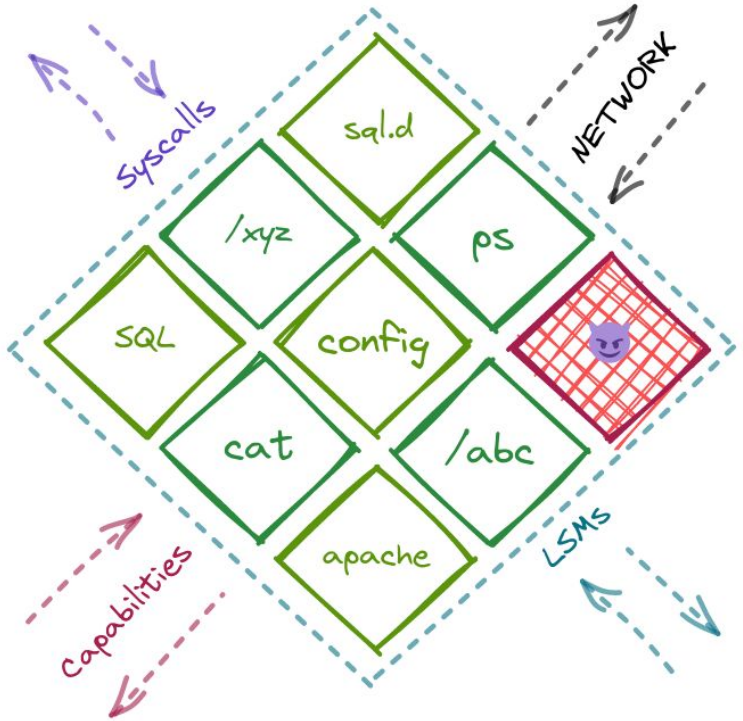
Pod Security Context

A security context defines privilege and access control settings for a Pod or Container. Security context settings include, but are not limited to Seccomp, Capabilities, AppArmor, SElinux…

```
container.apparmor.security.beta.kubernetes.io/<container_name>: <profile_ref>
```

```
type=AVC
msg=audit(1632826325.160:1463)
apparmor="DENIED"
operation="open"
profile=”apparmor_profile_1"
name=”./password"
pid=3875
comm=”cat"
requested_mask="ac"
denied_mask="ac"
fsuid=0
ouid=0
```

- Fine Grained Access Control
- Well Integrated with Modern Ecosystem
- Learning Curve
- AppArmor not available on
- all systems
- **Audit Logs/Alerts not enough context**



Problem Statement

- Access Control on Container Entities
- Declarative Way to Manage Policies for Access Control
- Inline Policy Enforcement
- Telemetry Events with Context

KubeArmor

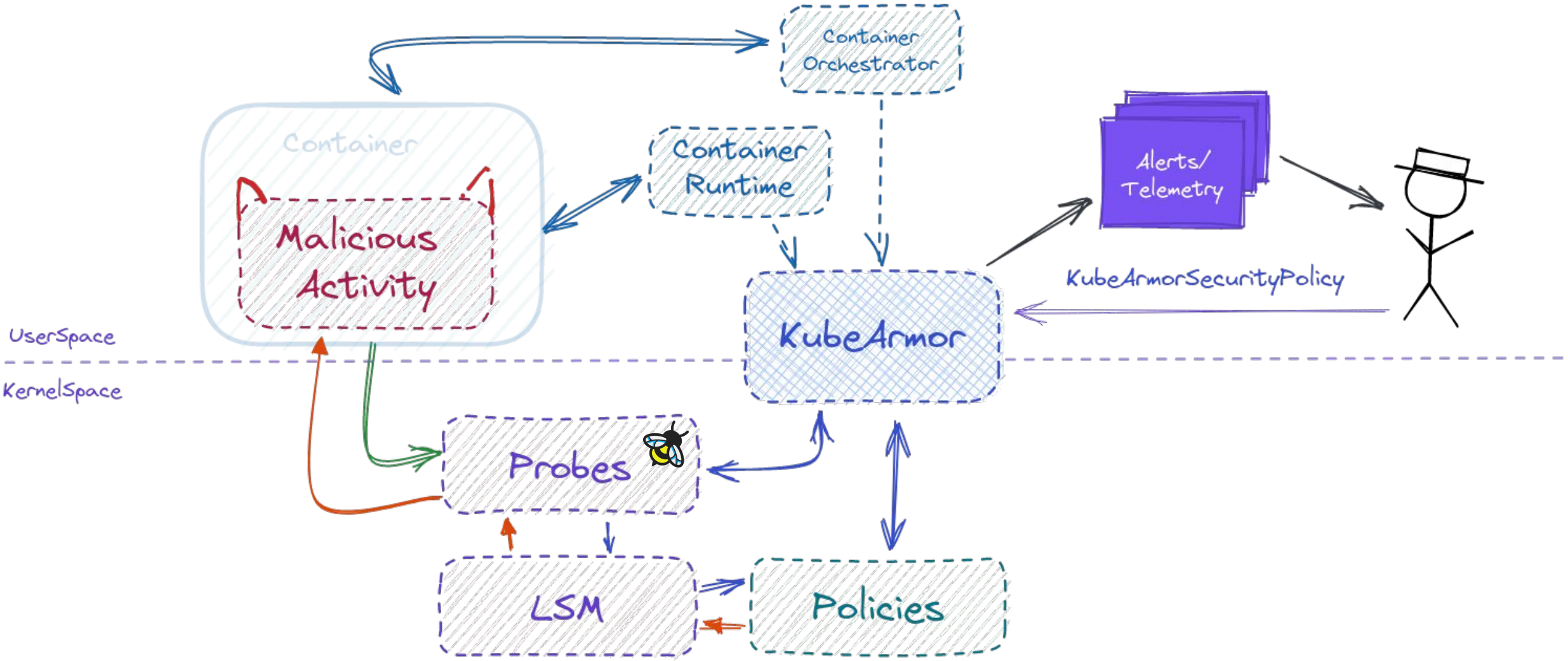
KubeCon | CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

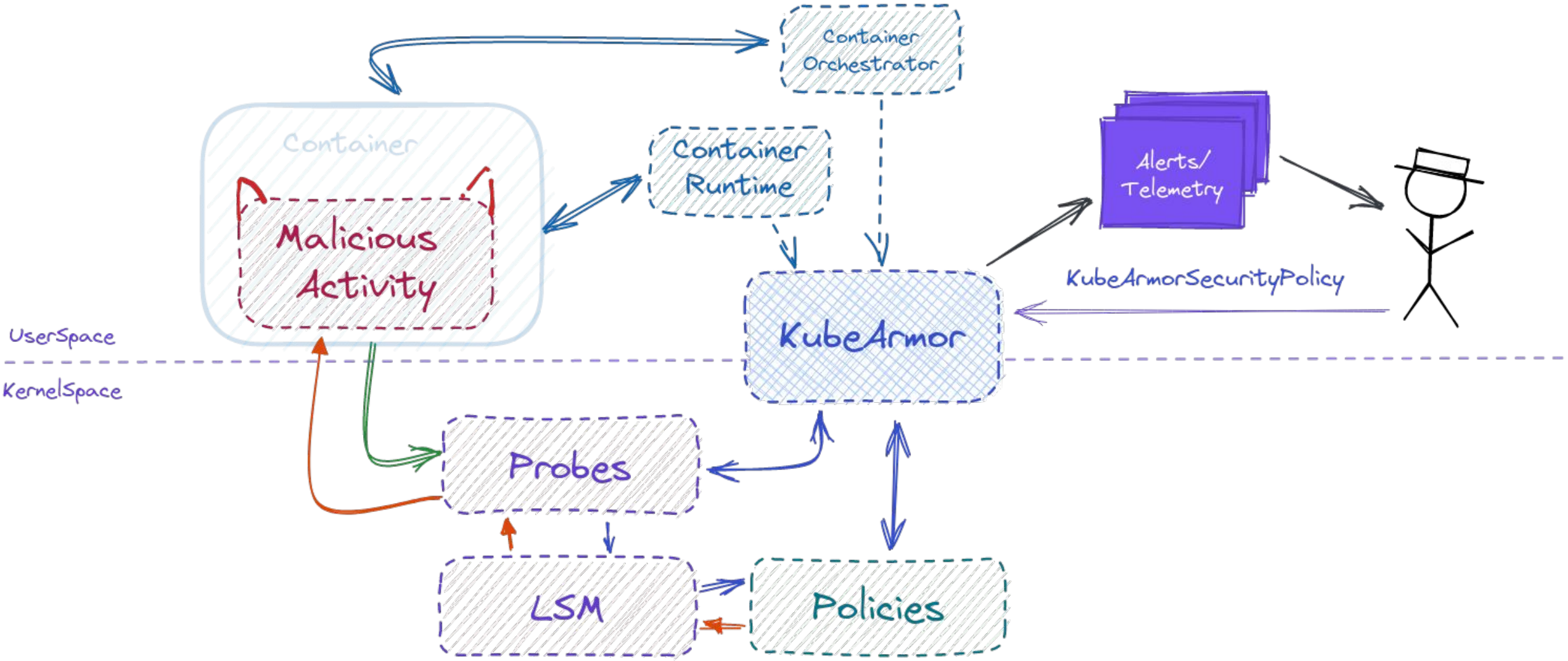
DETROIT 2022

KubeArmor is a cloud-native runtime security enforcement system that restricts the behavior (such as process execution, file access, and networking operations) of containers and nodes (VMs) at the system level.

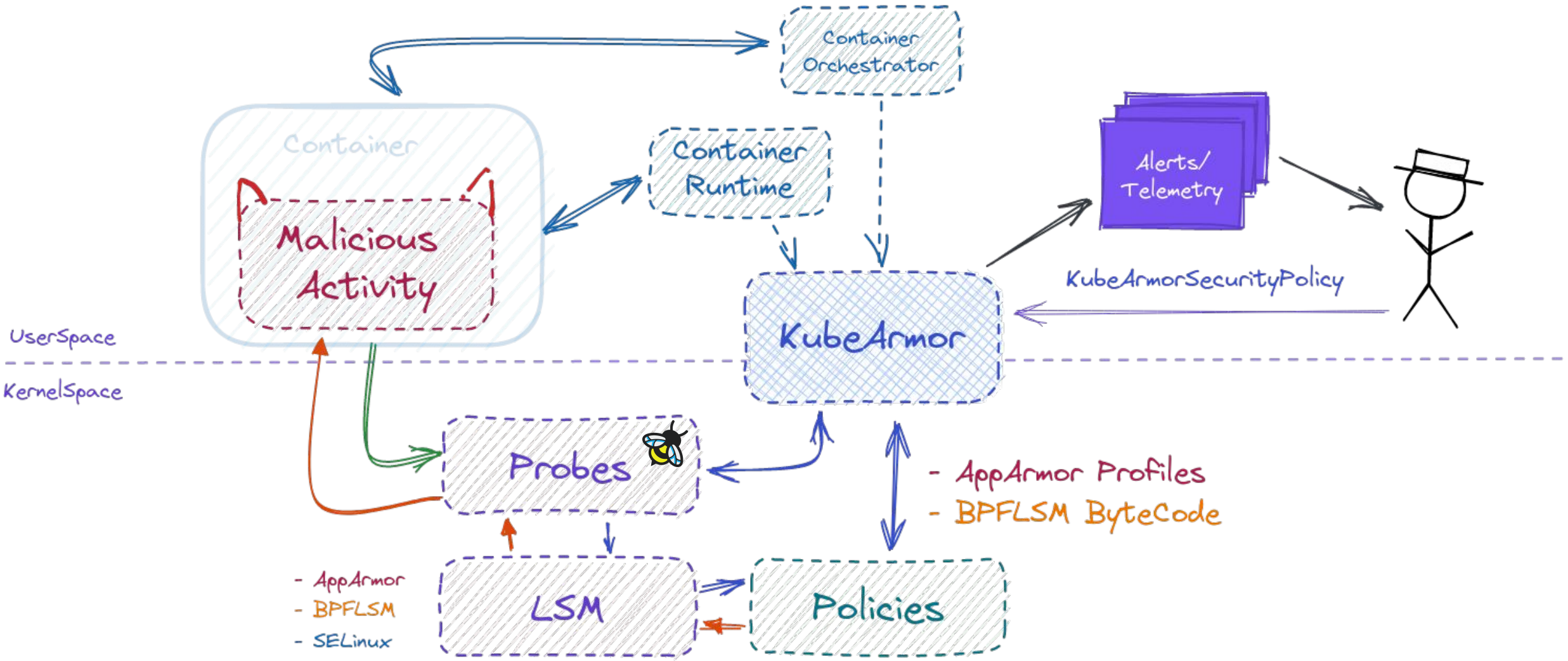
KubeArmor


Container Orchestrator
Container
Malicious Activity
Container Runtime
Alerts/ Telemetry
KubeArmorSecurityPolicy
KubeArmor
UserSpace
KernelSpace
Probes
LSM
Policies

KubeArmor

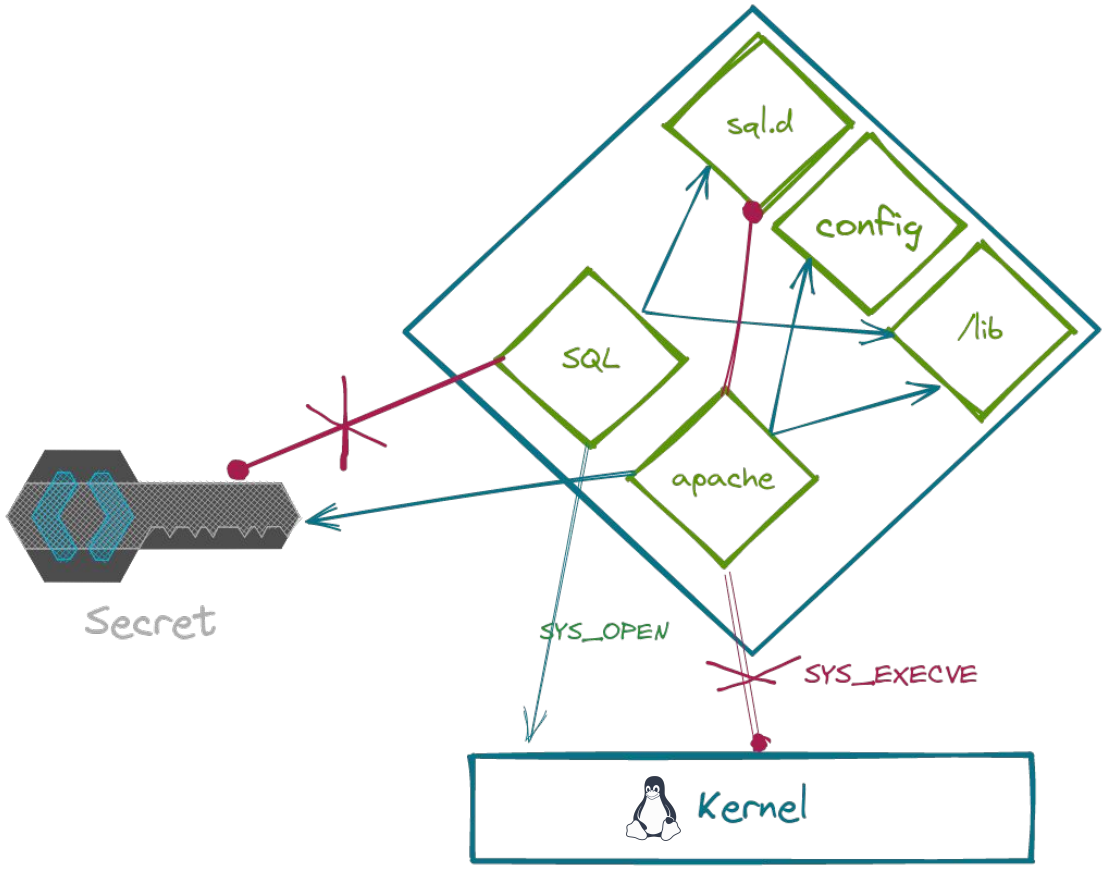

Container Orchestrator
Container
Malicious Activity
Container Runtime
Alerts/ Telemetry
KubeArmorSecurityPolicy
KubeArmor
UserSpace
KernelSpace
Probes
LSM
Policies

KubeArmor


Container Orchestrator
Container
Malicious Activity
Container Runtime
Alerts/ Telemetry
KubeArmorSecurityPolicy
UserSpace
KernelSpace
KubeArmor
Probes
- AppArmor Profiles
- BPFLSM ByteCode
- AppArmor
- BPFLSM
- SELinux
LSM
Policies

in Action

Demo Scenario



- Allow Specific Entities and Deny Rest
- Restrict App Access to Kernel Calls
- Lockdown Access to
  - K8s Secrets
  - Service Account Tokens
- Secure Pod Data

in Action

KubeArmor

- Access Control on Container Entities
- Declarative Way to Manage Policies for Access Control
- Inline Policy Enforcement
- Telemetry Events with Context

KubeCon | CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Thank You

https://kubearmor.io/

https://kubearmor.herokuapp.com/

https://barun.cc/

@daemon1024