

Skillbox

# Инфраструктура и среда исполнения

**Чернухин Максим**

Software Architect

АО «Альфа-Банк»

Skillbox

# OS, Bare Metal, Virtualization, Containers

**Чернухин Максим**

Software Architect

АО «Альфа-Банк»

# В прошлом модуле

- API Gateway
- Его функции и паттерны использования

# В этом модуле

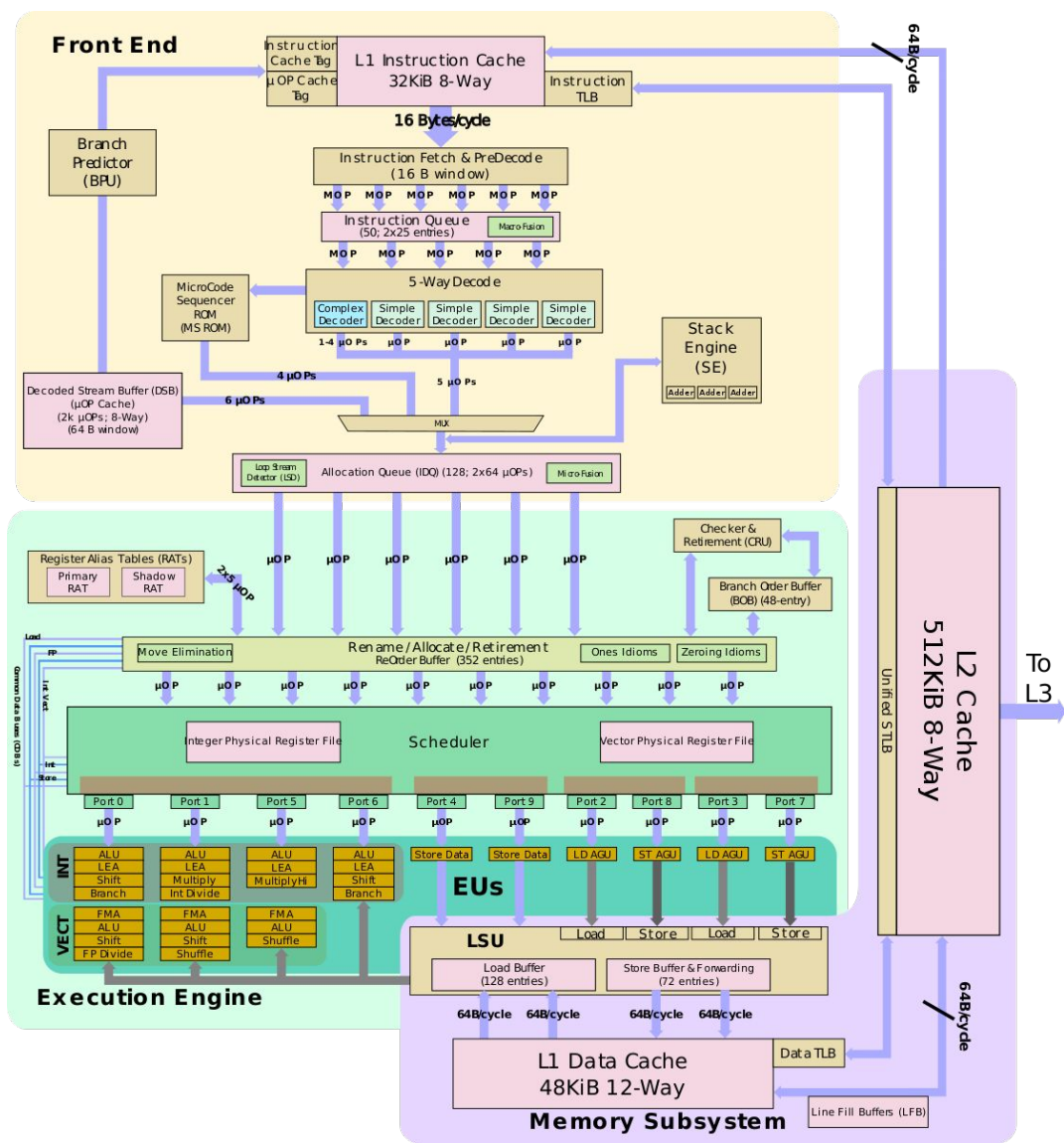
- Узнаем, что за собой скрывает понятие «инфраструктура»
- Узнаем, как происходит установка приложений
- Познакомимся с оркестрацией

# На этом уроке мы узнаем

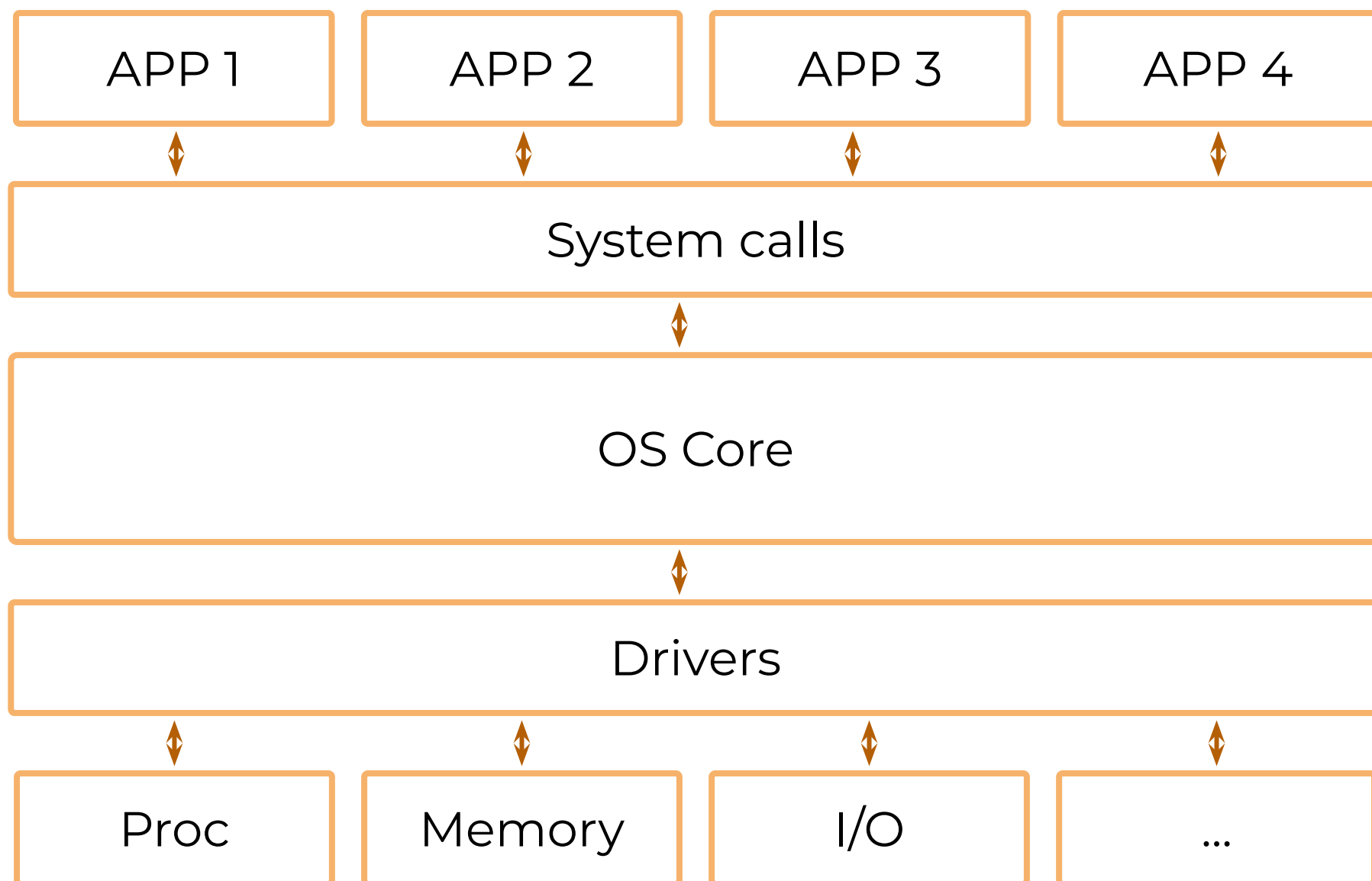
- Чем отличается VM от Docker-контейнера
- Что такое Bare Metal
- Зачем нужен гипервизор виртуальной машине

# Sunny Cove

## Microarchitectures

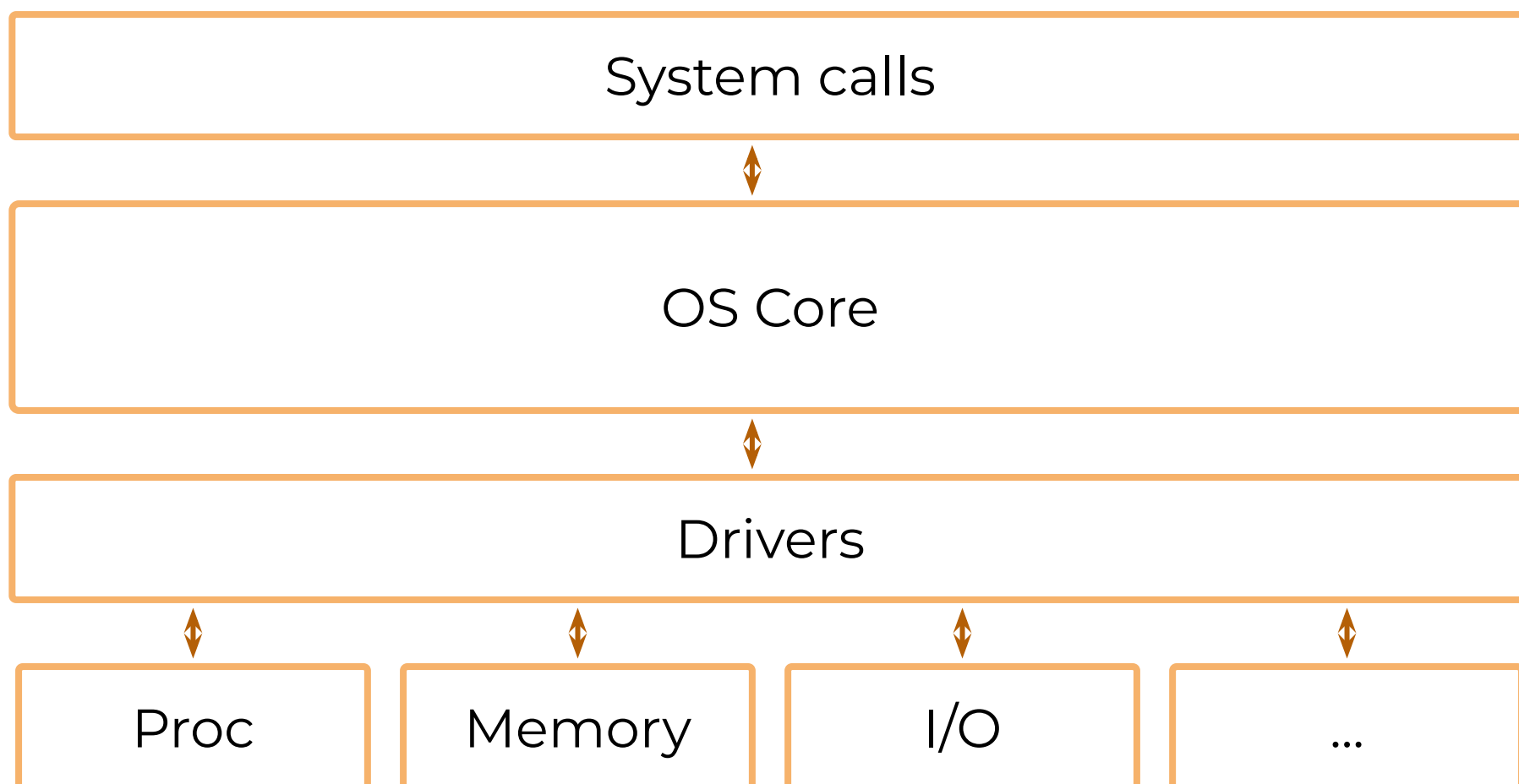


# Работа операционной системы



# Управление ресурсами

**Kernel space** — адресное пространство ядра ОС, в котором процессы имеют привилегированный доступ к ресурсам компьютера и другим процессам.



# Управление ресурсами

**User space** — адресное пространство, отведённое для пользовательских процессов (приложений), то есть не имеющих привилегированный доступ к ресурсам.

APP 1

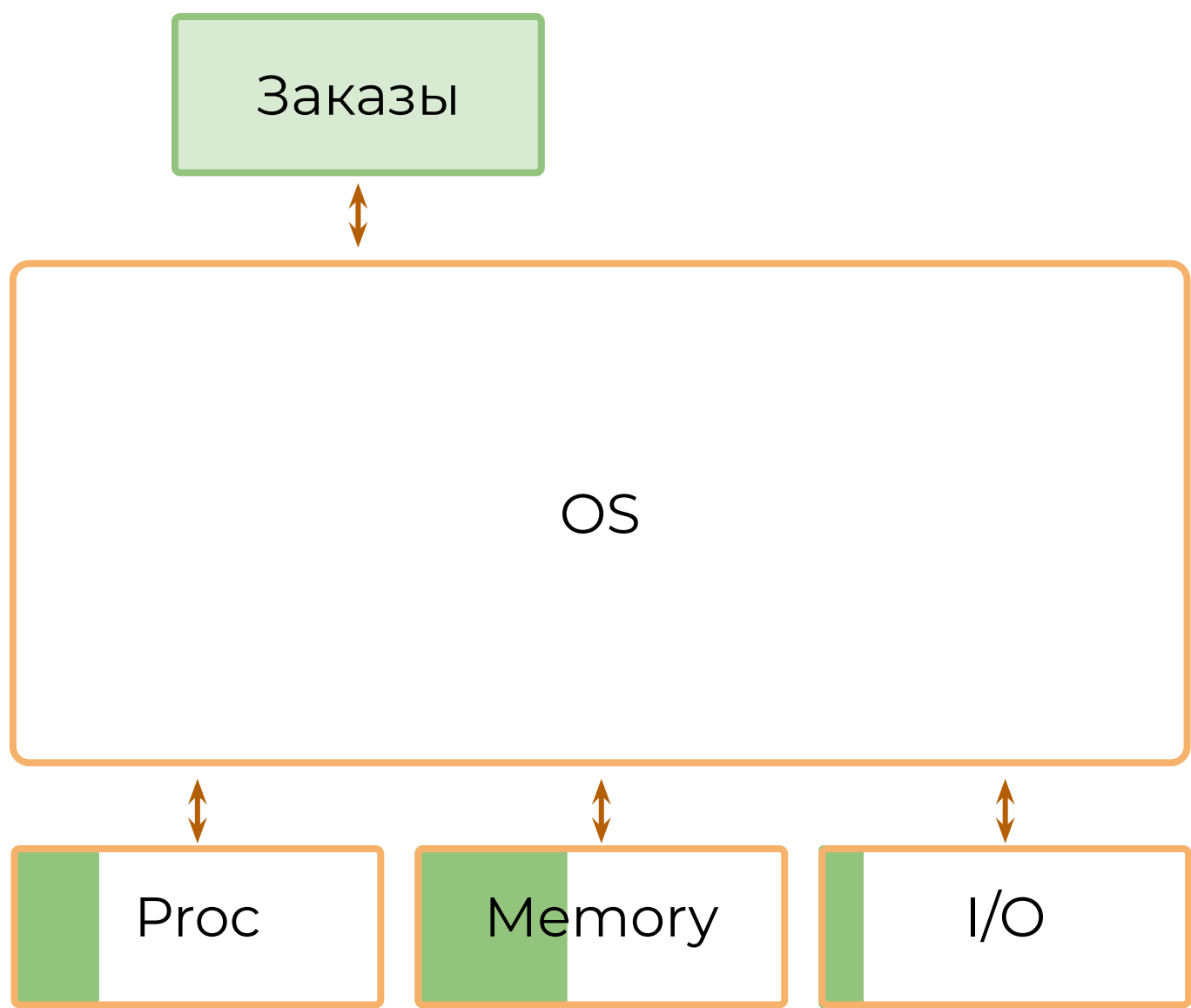
APP 2

APP 3

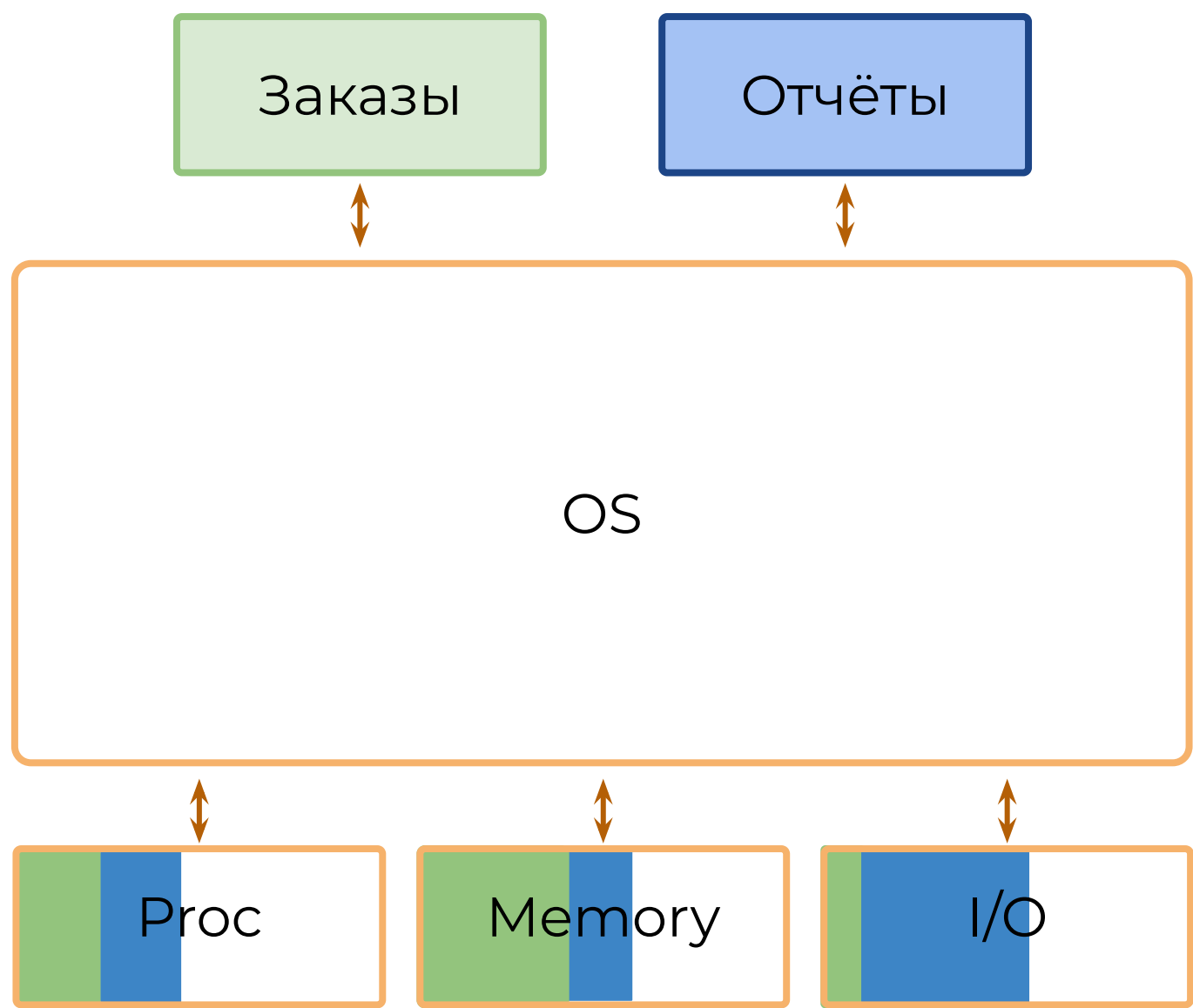
APP 4



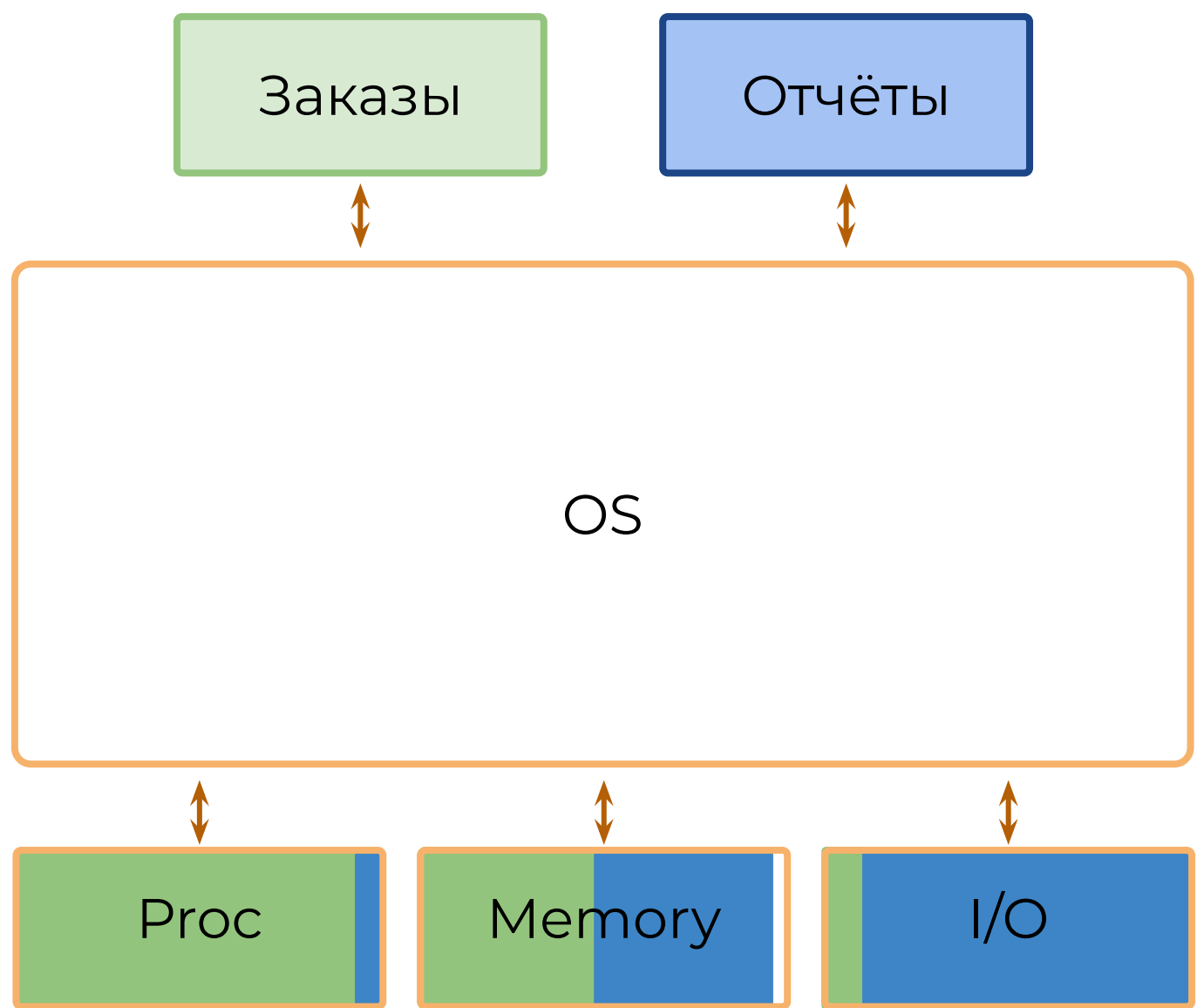
# Эксплуатация



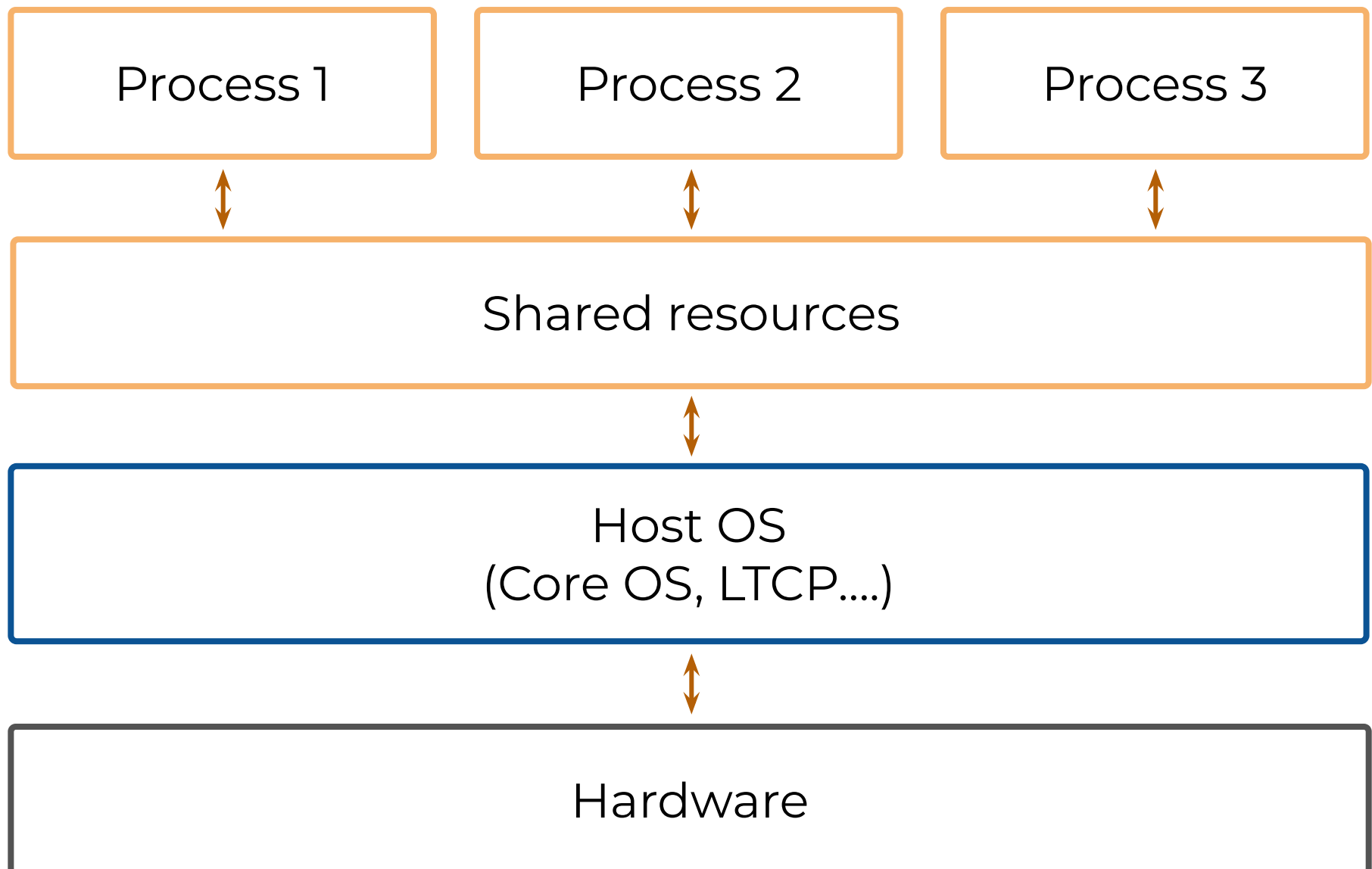
# Эксплуатация



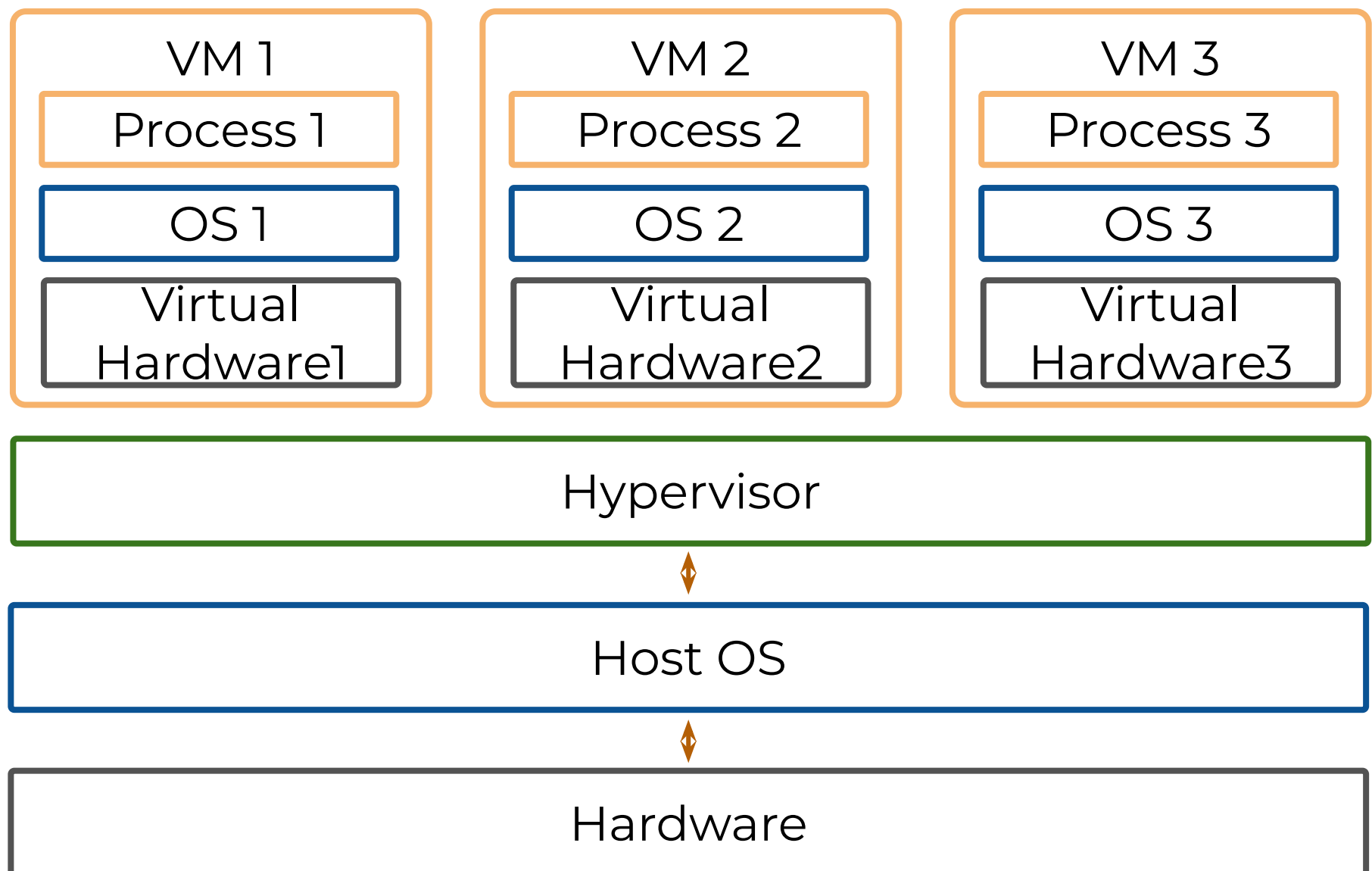
# Эксплуатация



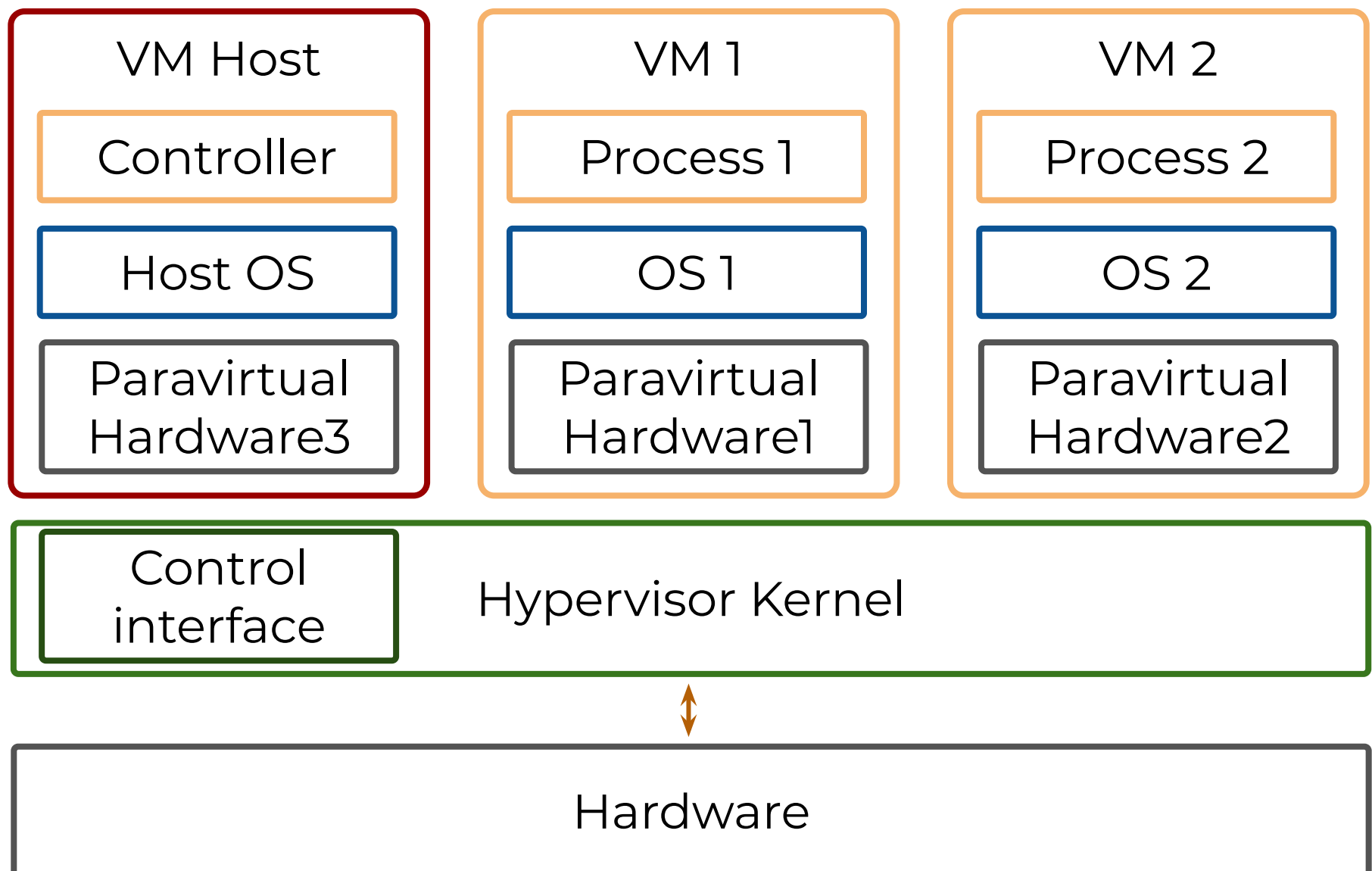
# Bare Metal



# Virtualization



# Paravirtualization



# Запись на диск

Virtualization

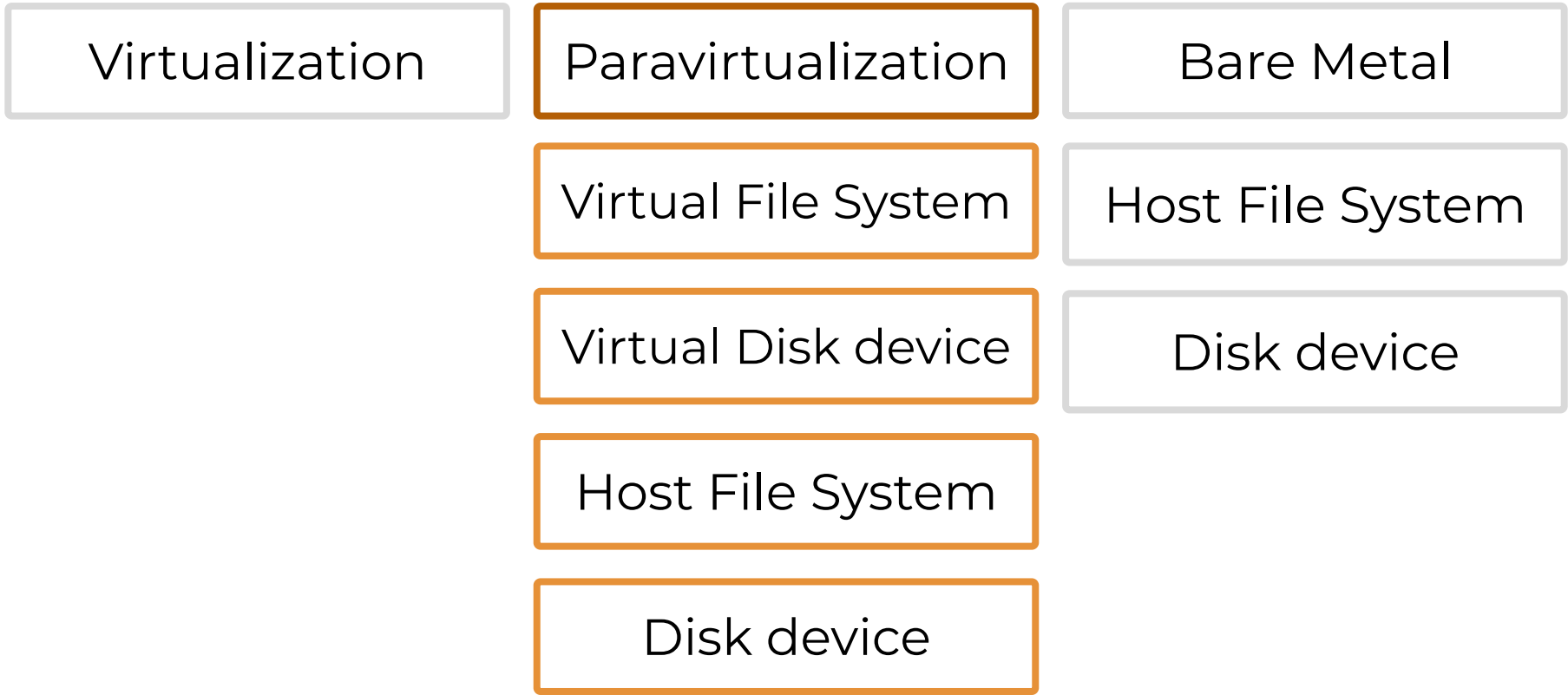
Paravirtualization

Bare Metal

Host File System

Disk device

# Запись на диск





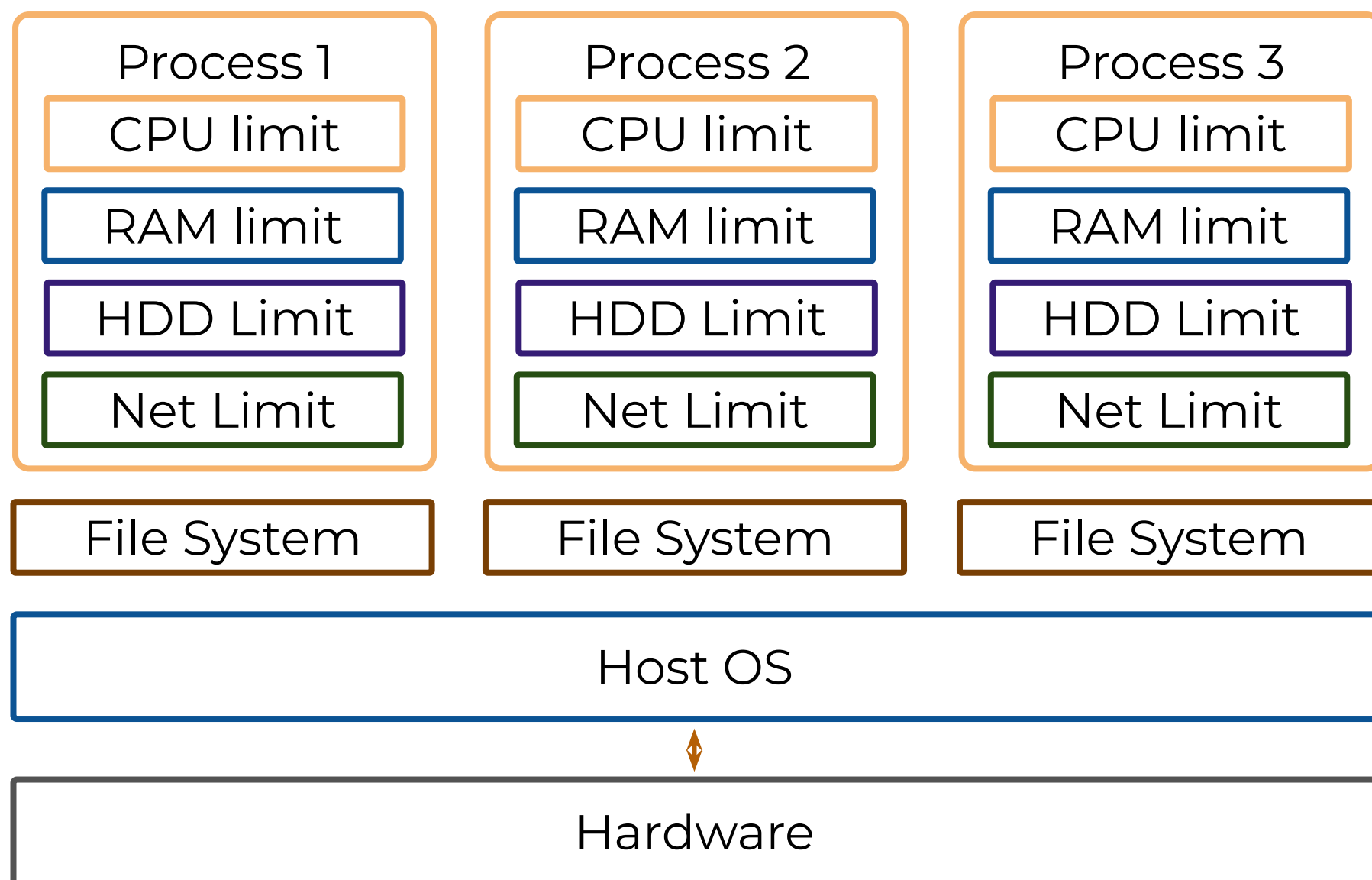
# Запись на диск

Virtualization	Paravirtualization	Bare Metal
Virtual File System	Virtual File System	Host File System
Virtual Disk device	Virtual Disk device	Disk device
Disk Emulation	Host File System	
Host File System	Disk device	
Disk device		

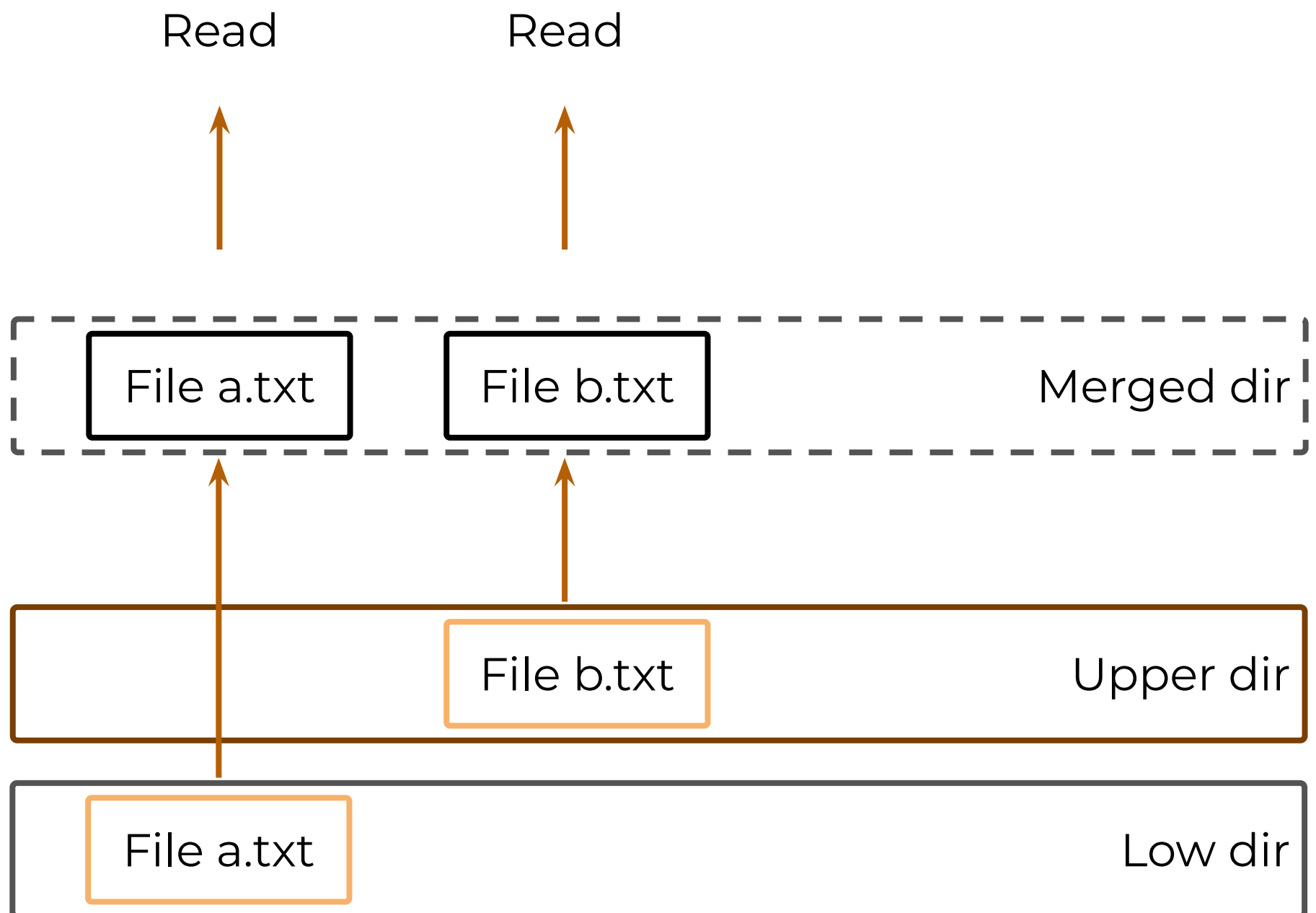
# Unix systems

- **Control group (cgroup)** — группа процессов, для которой механизмами ядра наложена изоляция и установлены ограничения на вычислительные ресурсы (процессорные, сетевые, ресурсы памяти, ресурсы ввода-вывода)
- **Namespaces** — это функция ядра Linux, позволяющая изолировать и виртуализировать глобальные системные ресурсы множества процессов (ID процессов, имена хостов, ID пользователей, доступ к сетям, межпроцессное взаимодействие и файловые системы)

# Resource sharing

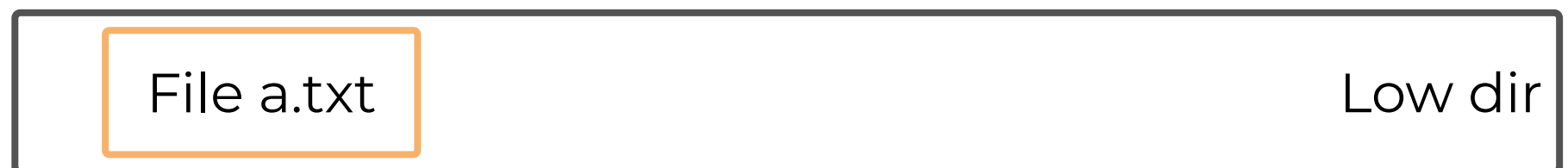
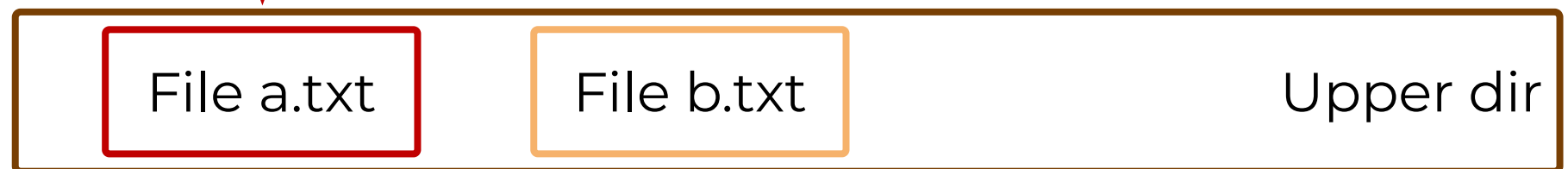


# Overlay Filesystem

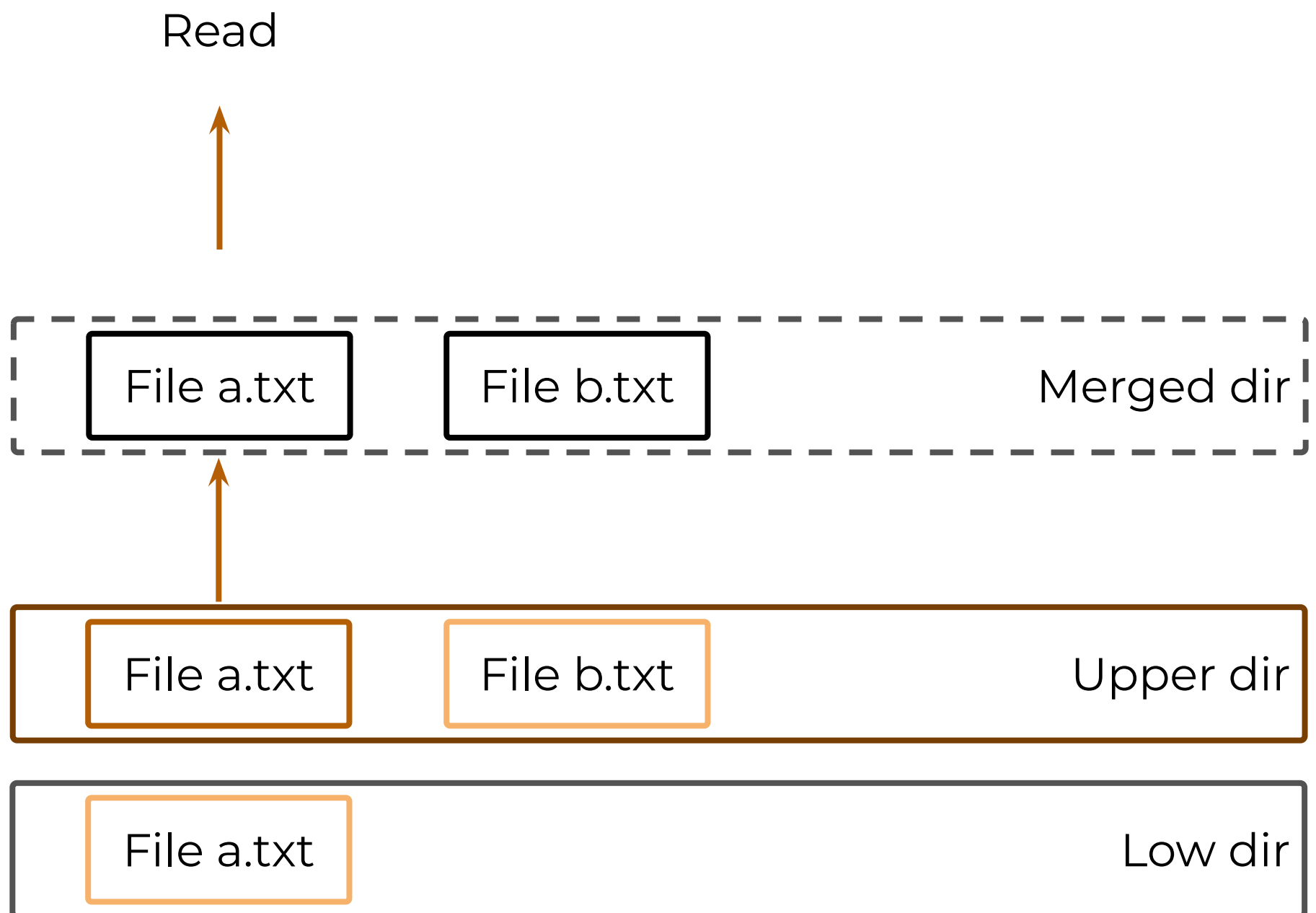


# Overlay Filesystem

Write

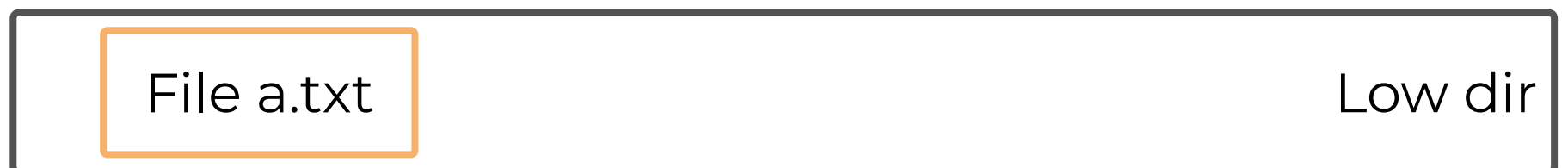
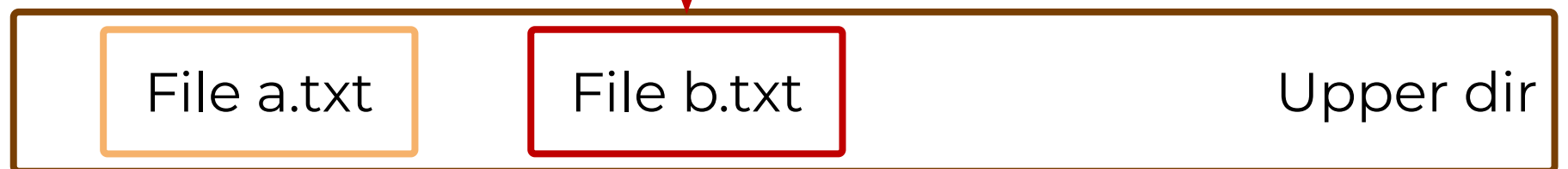
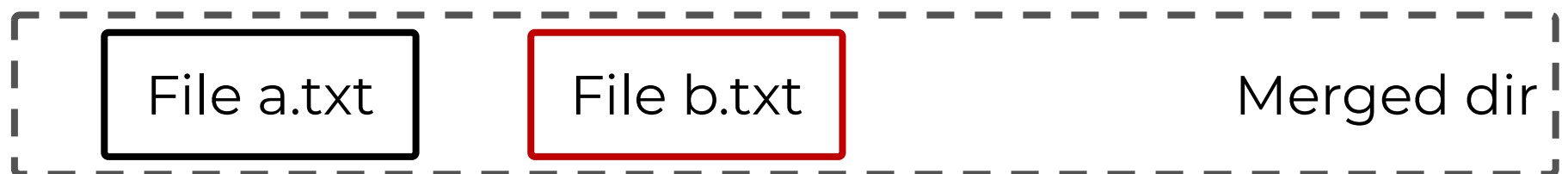


# Overlay Filesystem

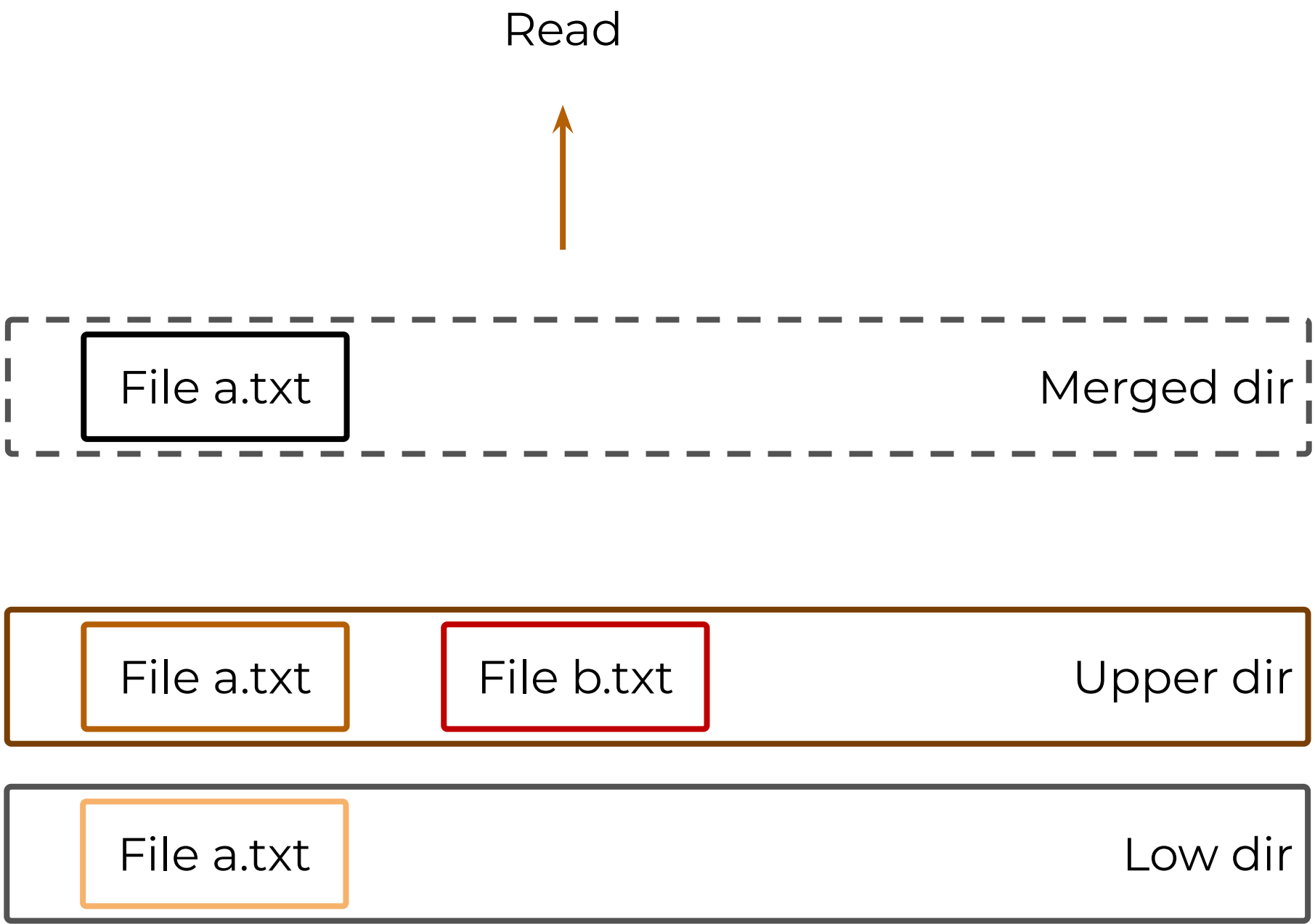


# Overlay Filesystem

Delete



# Overlay Filesystem

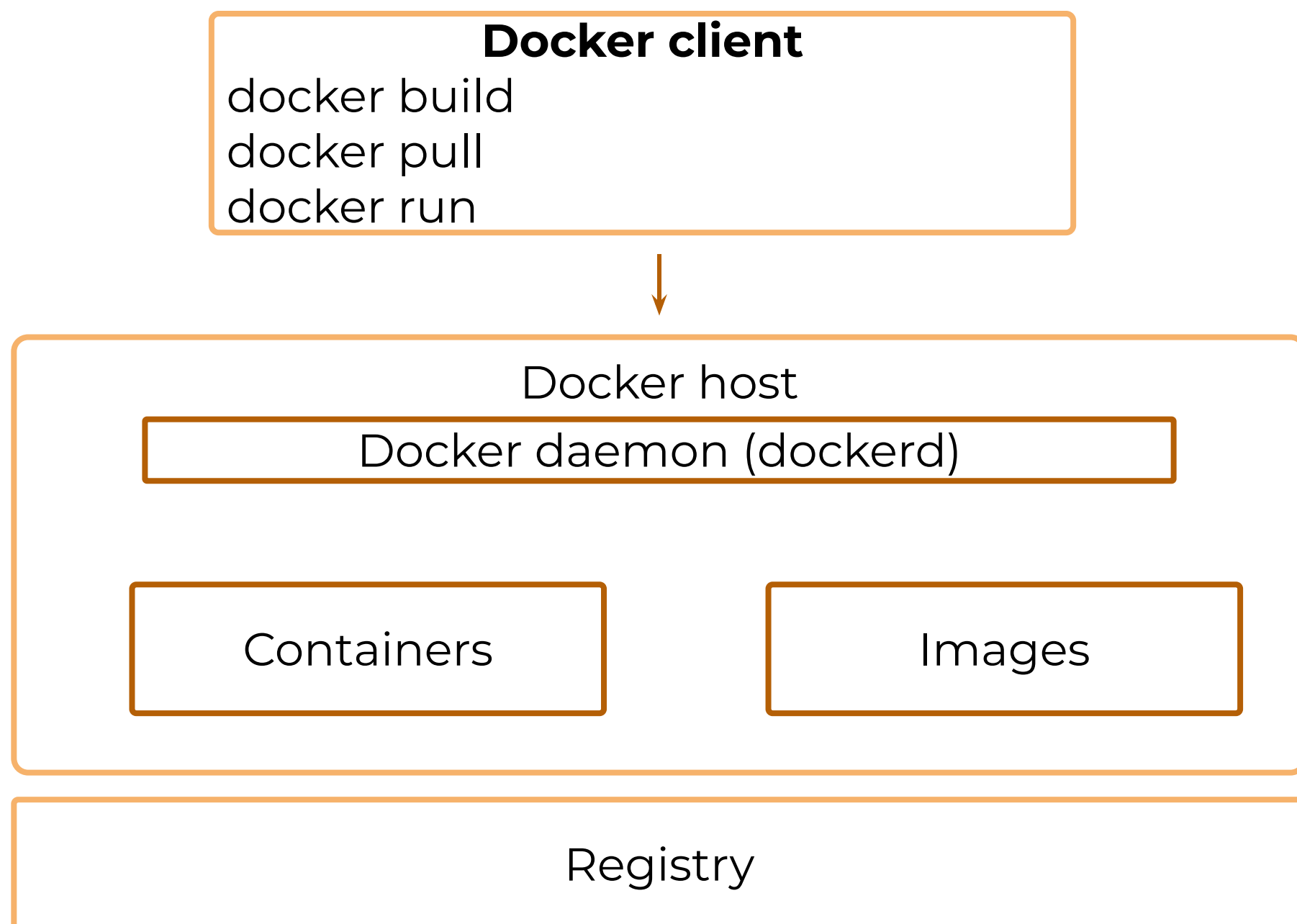




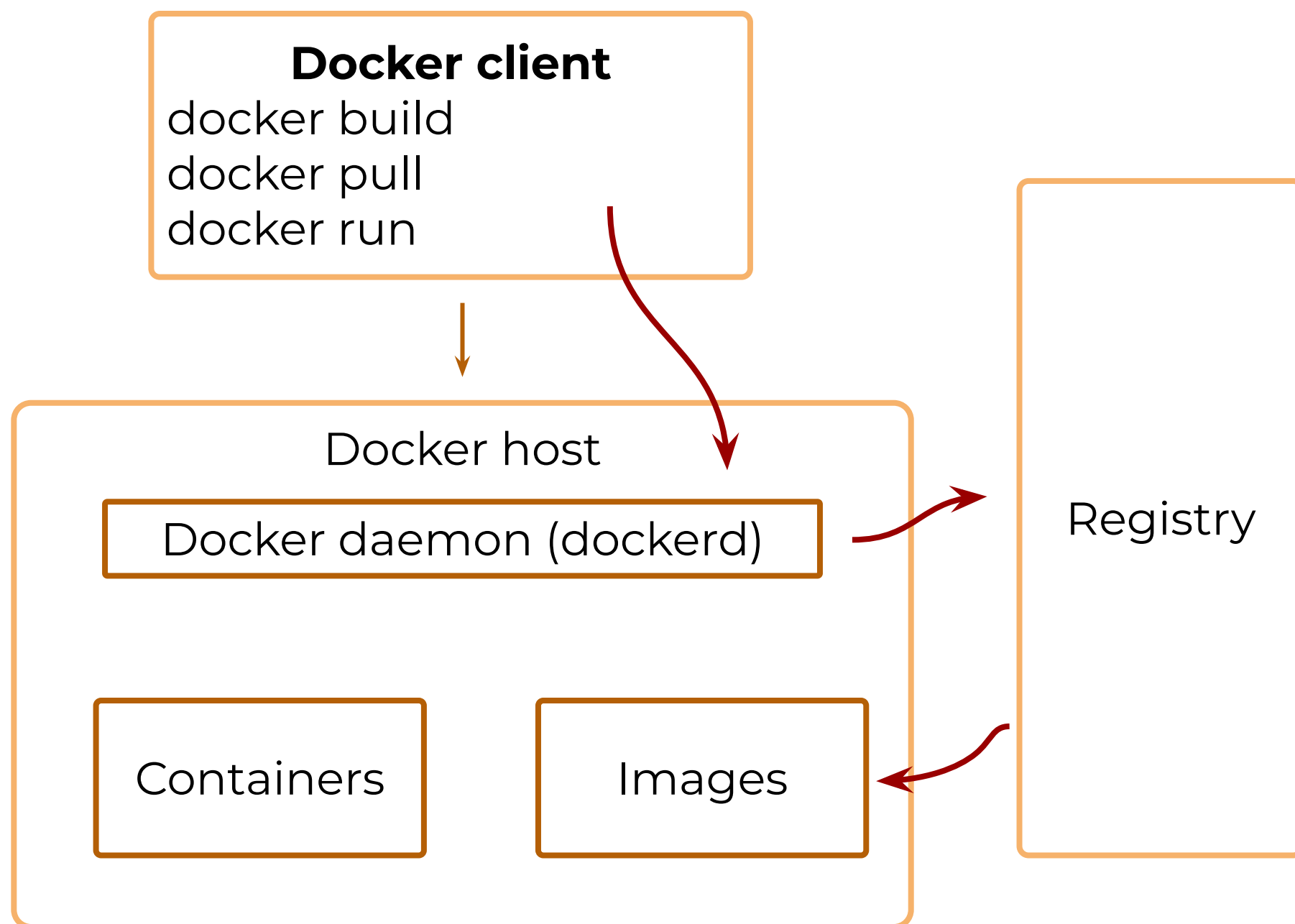
# Контейнеры

- **LXC, Rocket, Docker, OpenVZ, Yandex.Porto**
- **Плюсы:**
  - изоляция и управление процессами, хостами, доступ к сети и т. д.
  - лимит для сети, CPU, RAM, диска
- **Минусы:**
  - управление модулями ядра
  - сложности с виртуализацией VPN и видеокарты
  - сильная связь с Host OS

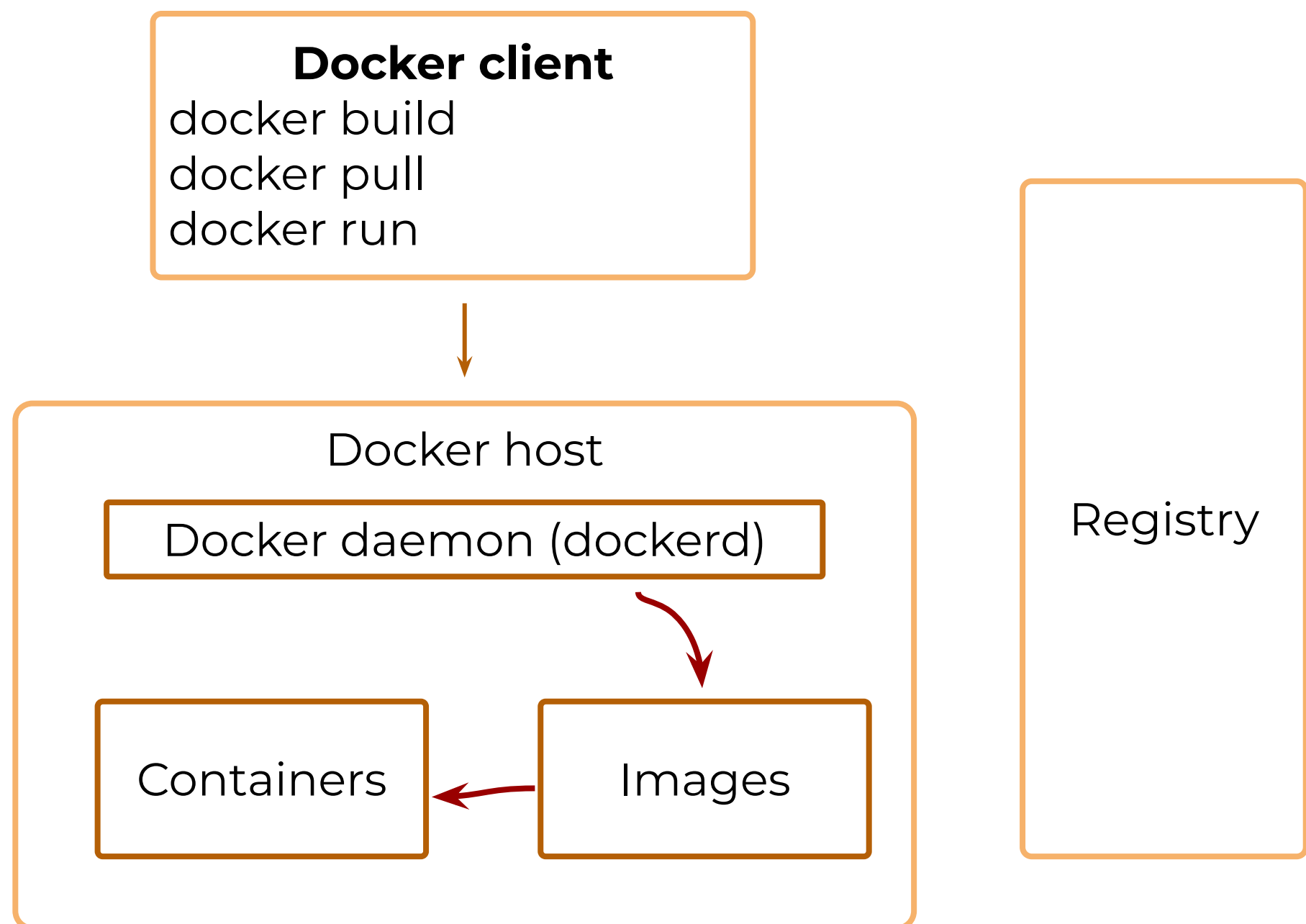
# Docker



# Docker



# Docker



# Выбор

- Важно выбирать тип использования ваших ресурсов исходя из:
  - требований по доступности
  - требований по времени восстановления
  - требований по времени масштабирования

# Итоги

- Мы познакомились с тем, как работают операционные системы
- Увидели, чем отличается виртуализация от контейнеров
- Познакомились с Docker и его файловой системой
- И поняли, что главное при выборе — это понимание бизнес-требований

# Что дальше?

- Поймём, что такое деплой приложений
- Познакомимся с видами установки приложений на production
- Узнаем, что такое CI/CD

Skillbox

**Спасибо  
за внимание!**