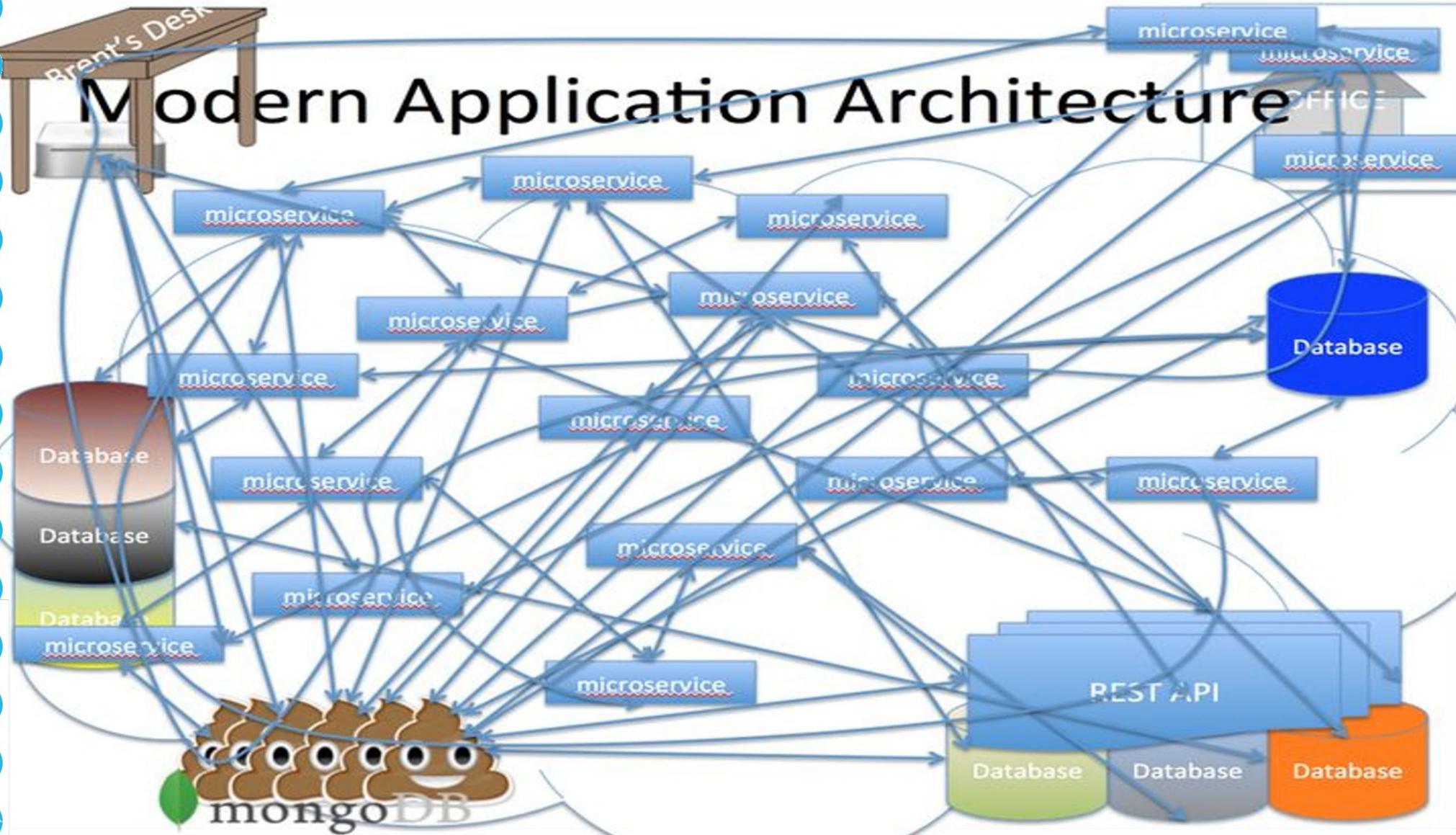




# Modern Application Architecture



# Окружения для деплоя



Компьютеры разработчиков



Армия клонов



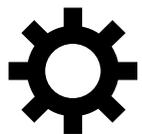
CI-сервер и тесты



Горизонтальное масштабирование



Staging



Production



Облако



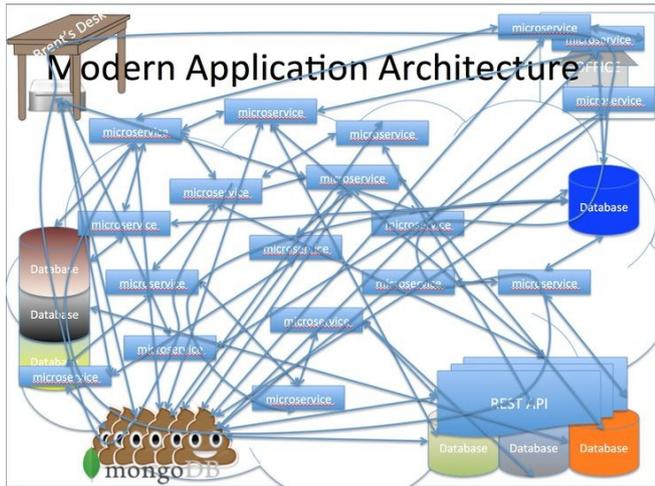
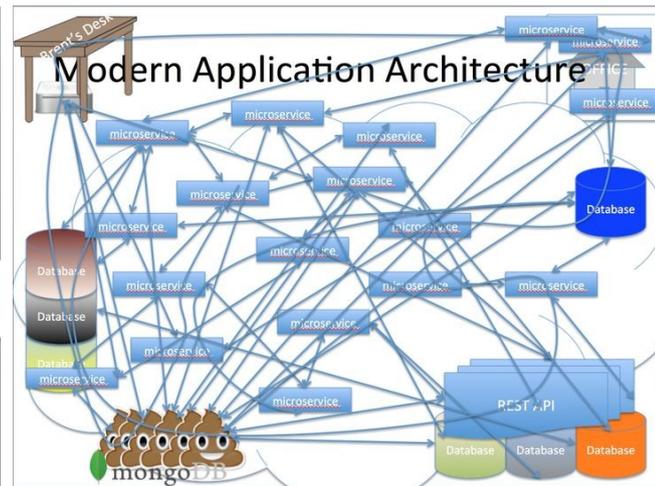
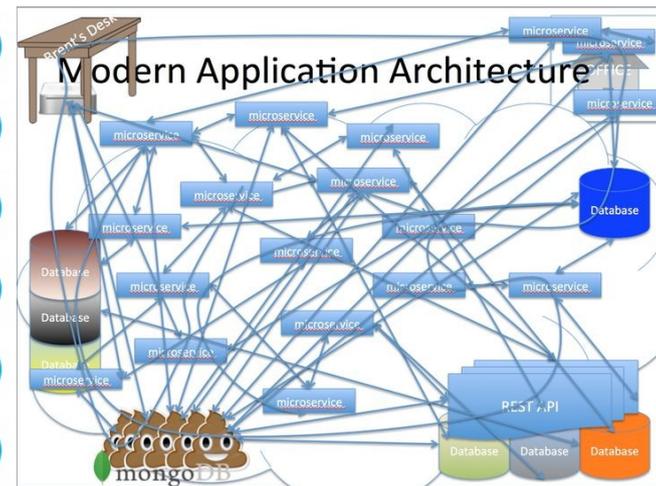
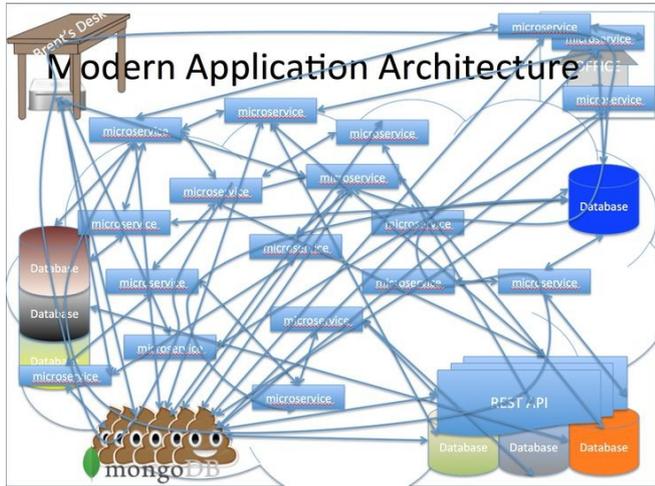
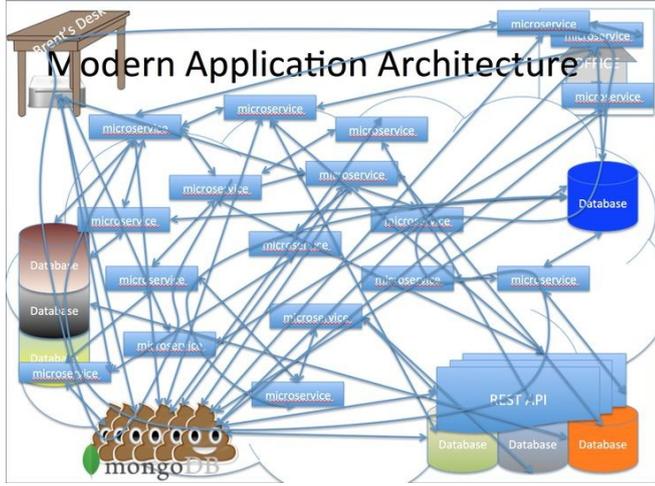
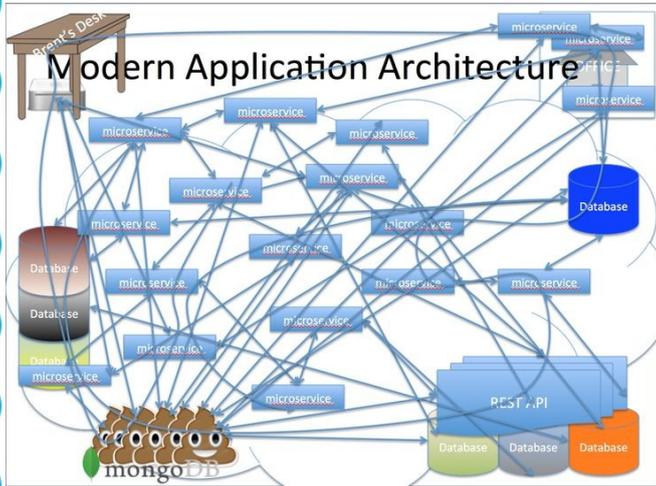
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
SQL 	?	?	?	?	?	?	?
SignalR 	?	?	?	?	?	?	?
Redis 	?	?	?	?	?	?	?
Queue 	?	?	?	?	?	?	?
Worker 	?	?	?	?	?	?	?
	Dev	CI Server	Staging	Production	Cluster	Multiple installations	Cloud



# Проблемы при деплое

- Сложность самого процесса из-за большого количества компонент и зависимостей между ними
- Никто полностью не понимает, как и что должно быть развёрнуто
- Устаревание документации
- Зависимости не всегда совместимы друг с другом
- Проблемы с откатом





# Проблемы при деплое

- Сложность самого процесса из-за большого количества компонент и зависимостей между ними
- Никто полностью не понимает, как и что должно быть развёрнуто
- Устаревание документации
- Зависимости не всегда совместимы друг с другом
- Проблемы с откатом

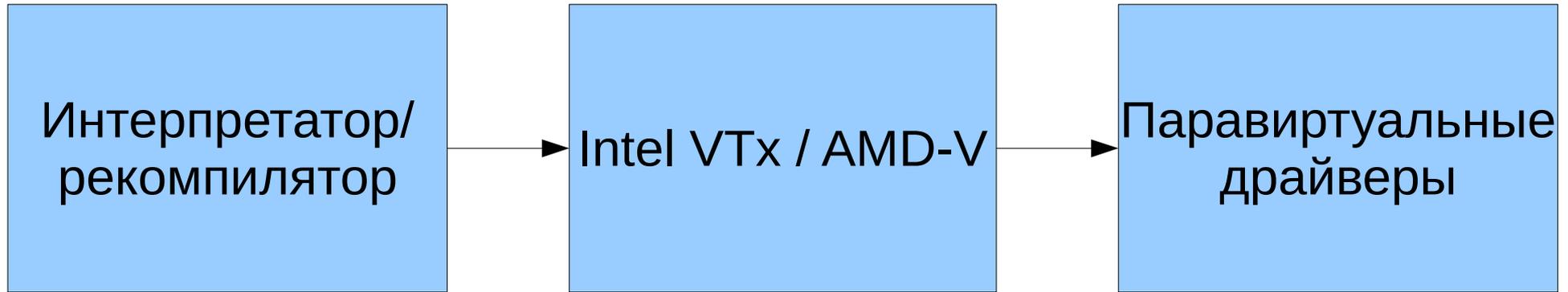


# Проблемы при решении с помощью VM

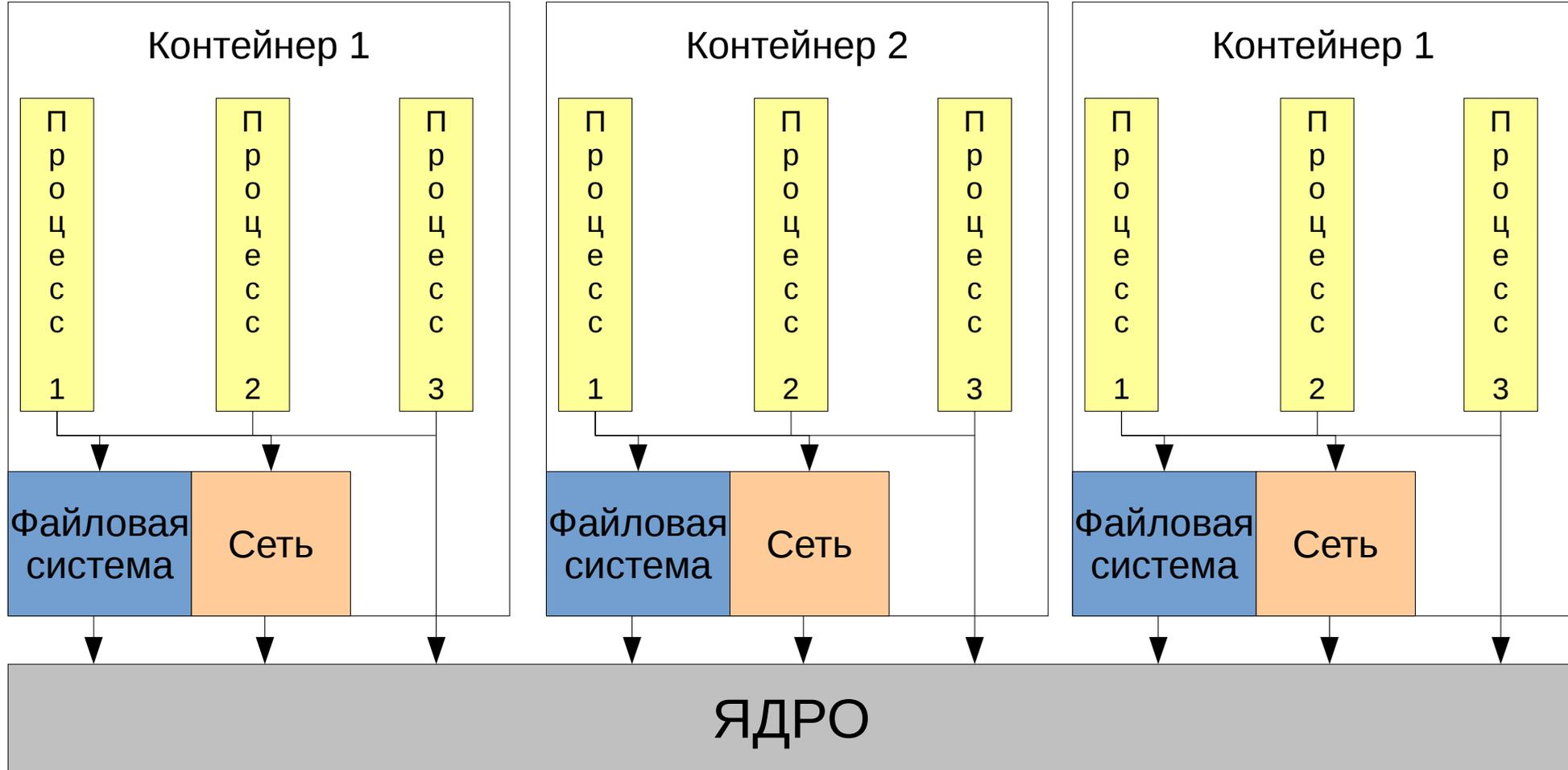
- Накладные расходы на виртуализацию (CPU, память)
- Каждый образ занимает много места, не относящегося к приложению
- Сложности при хранении конфигурации
- Сложно готовить образ
- Сложно управлять образами



# Эволюция VM



# Контейнерная виртуализация

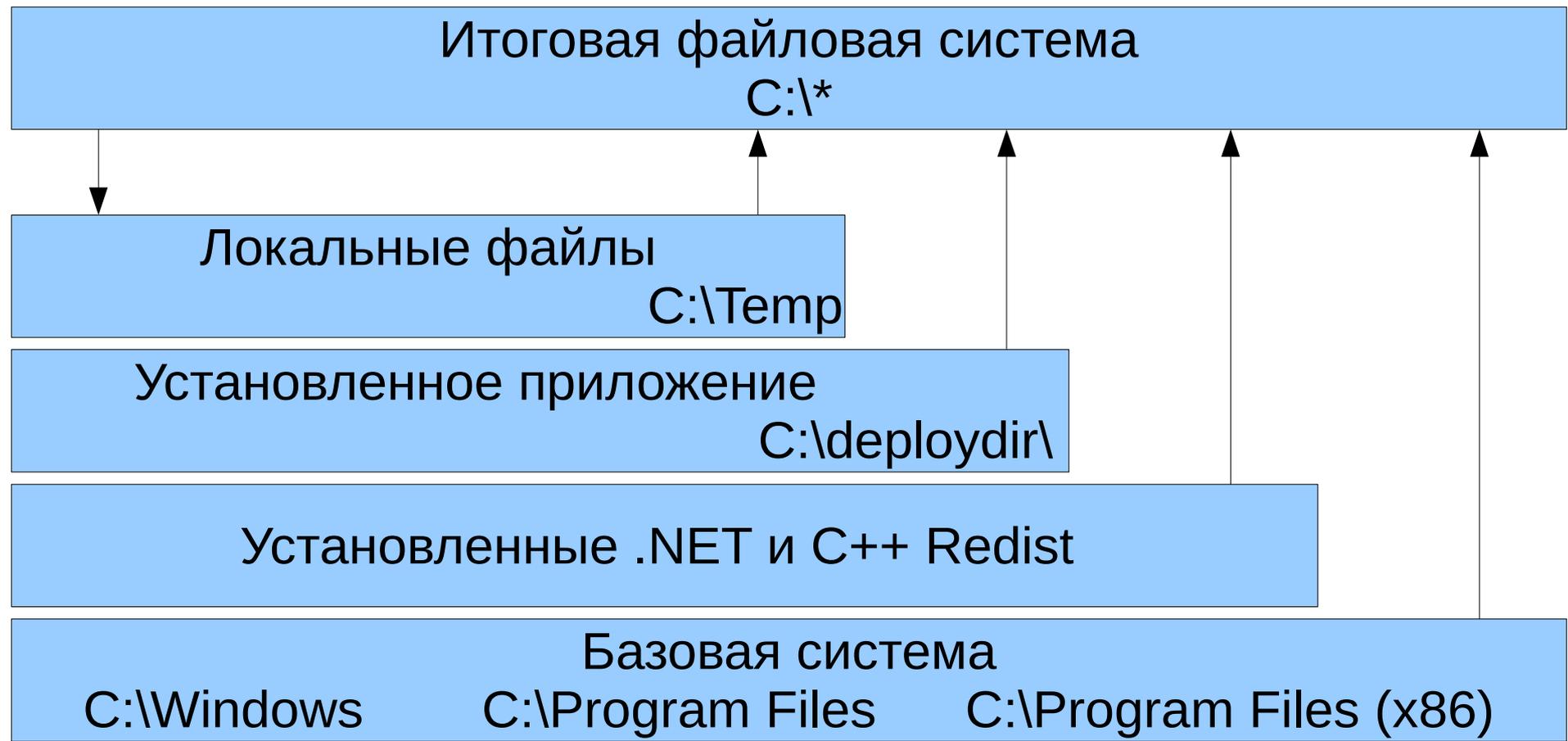


# Проблемы при решении с помощью VM

- Накладные расходы на виртуализацию (CPU, память)
- Каждый образ занимает много места, не относящегося к приложению
- Сложности при хранении конфигурации
- Сложно готовить образ
- Сложно управлять образами



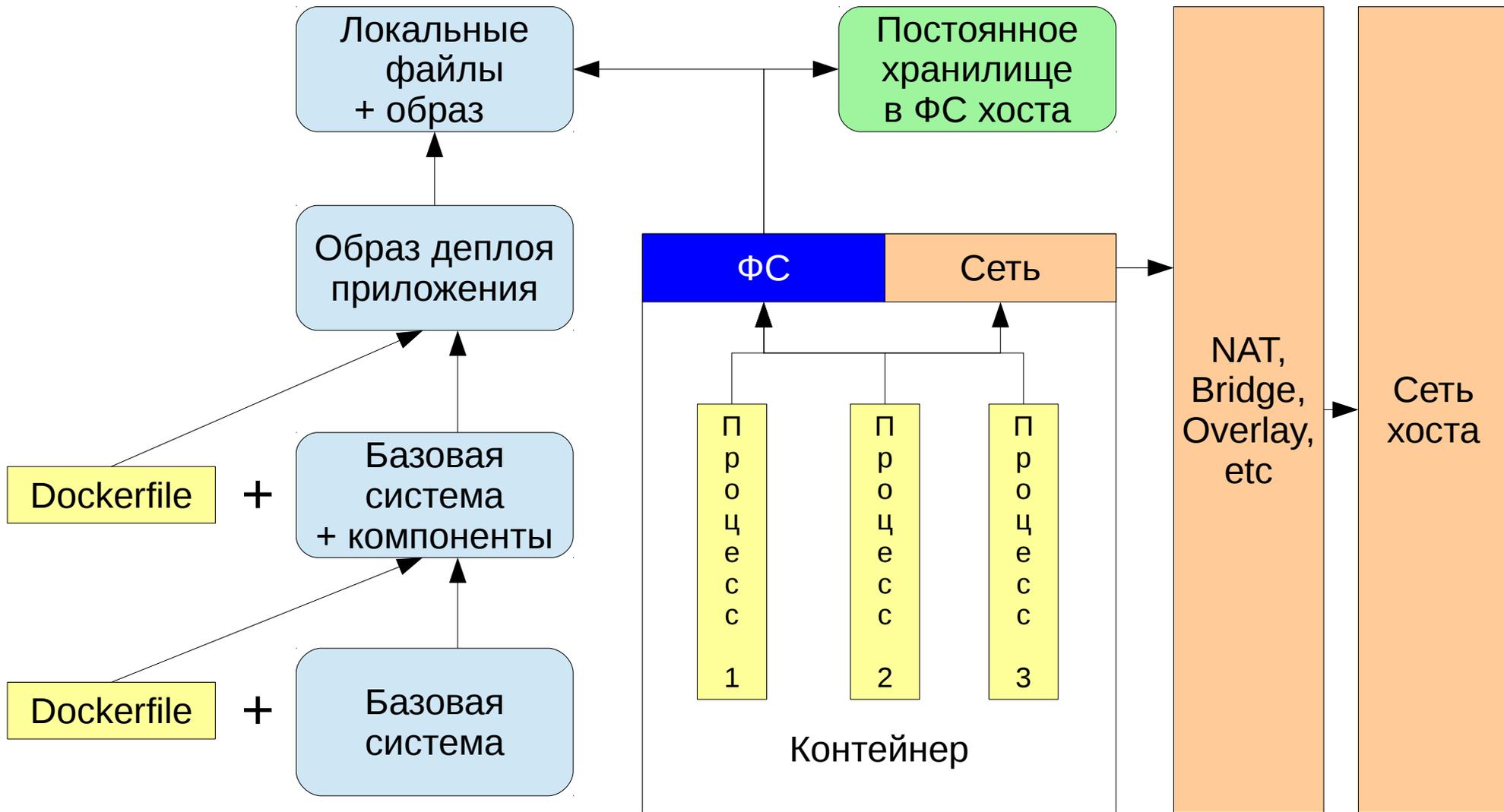
# Слой ФС



# Проблемы при решении с помощью VM

- Накладные расходы на виртуализацию (CPU, память)
- Каждый образ занимает много места, не относящегося к приложению
- Сложности при хранении конфигурации
- Сложно готовить образ
- Сложно управлять образами





# Демо

<http://github.com/kekekeks/dotnext2016spb>

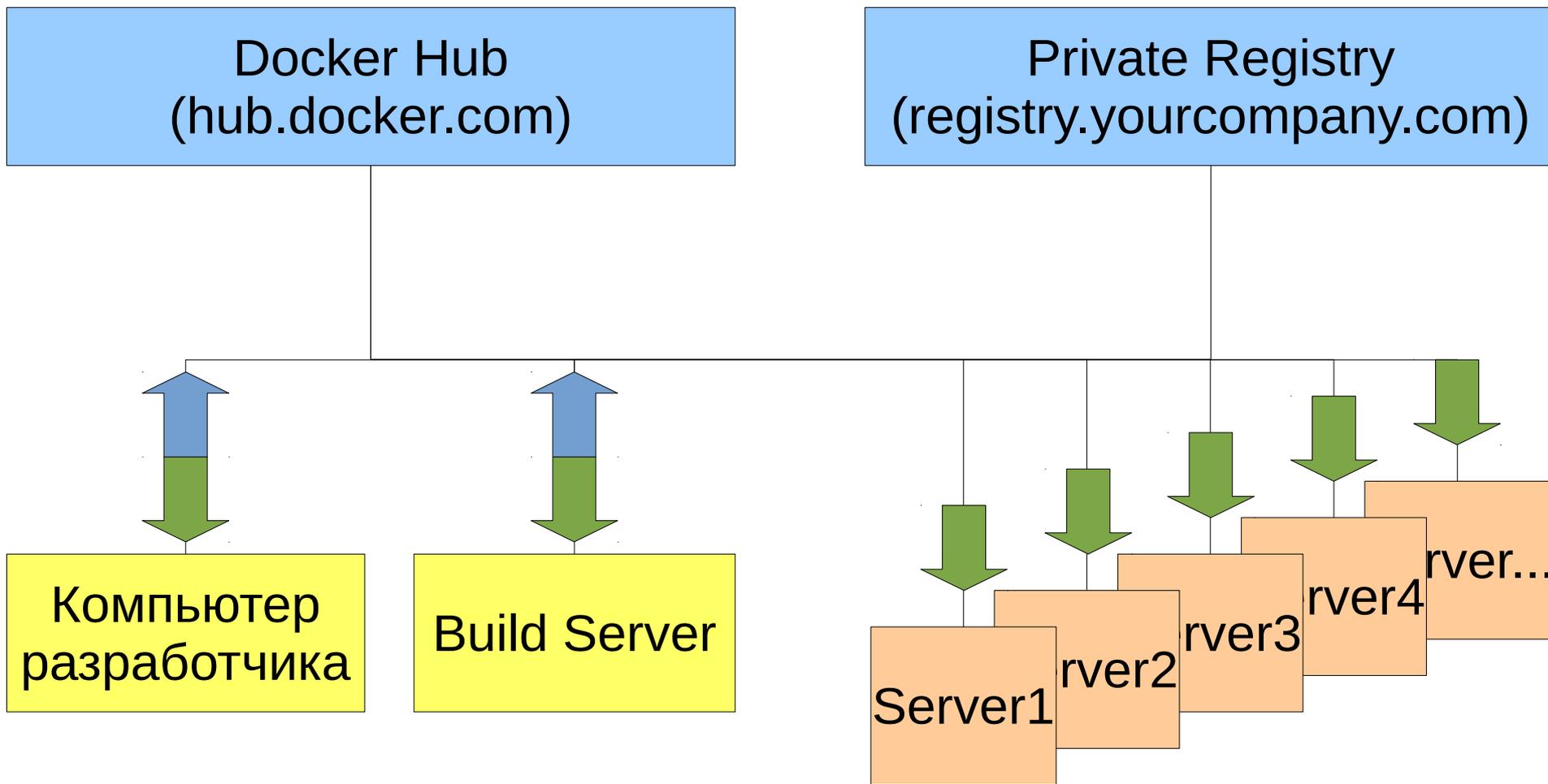


# Проблемы при решении с помощью VM

- Накладные расходы на виртуализацию (CPU, память)
- Каждый образ занимает много места, не относящегося к приложению
- Сложности при хранении конфигурации
- Сложно готовить образ
- Сложно управлять образами



# Docker Registry



# Проблемы при решении с помощью VM

- Накладные расходы на виртуализацию (CPU, память)
- Каждый образ занимает много места, не относящегося к приложению
- Сложности при хранении конфигурации
- Сложно готовить образ
- Сложно управлять образами



# Полезность разработчику

Один раз собранный образ можно запускать где угодно:

- Изолированное чистое выделенное и переносимое окружение для запуска приложения.
- Не надо думать о возможных проблемах совместимости с софтом, установленным на целевом сервере.
- Все серверы выглядят одинаково, т. к. находятся за пределами контейнера



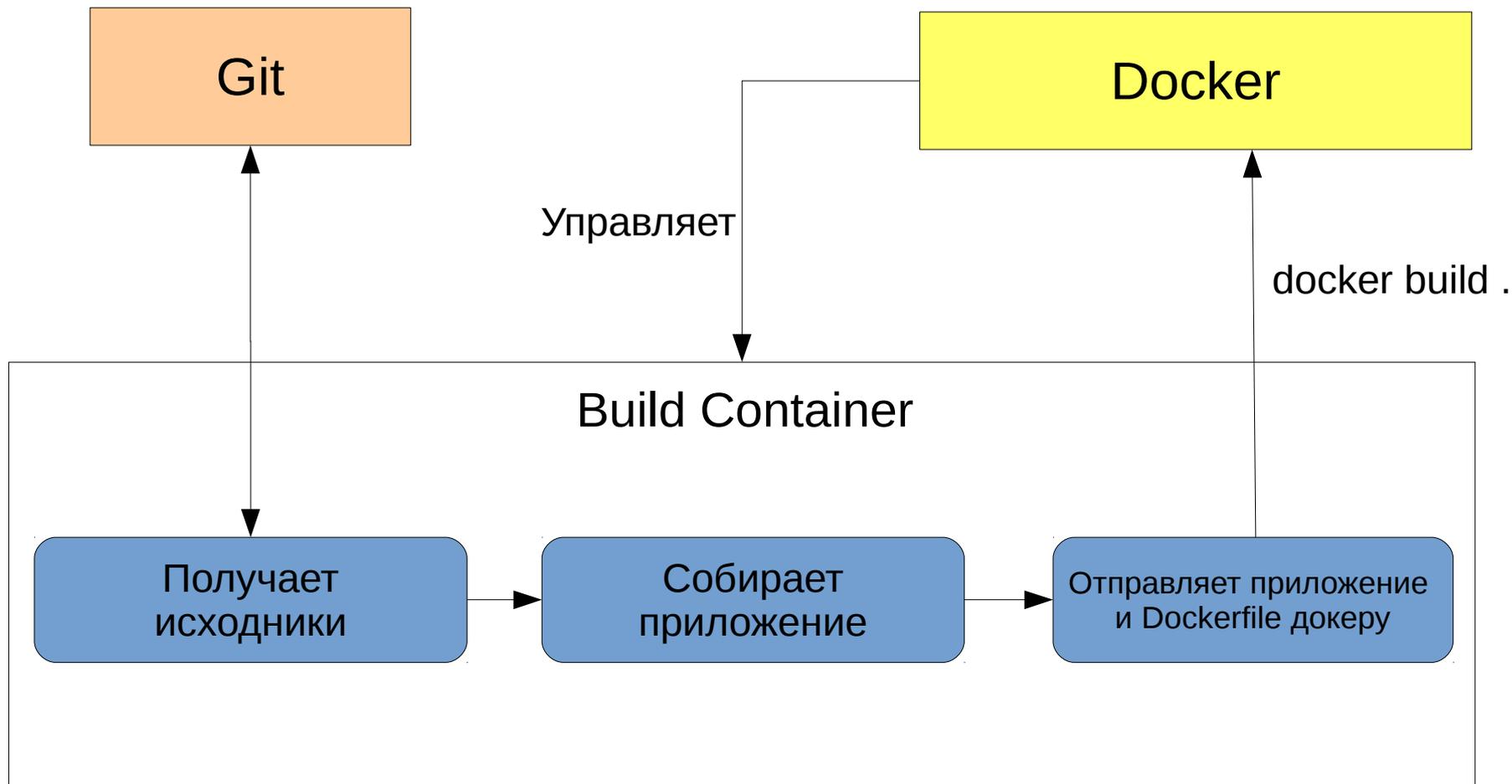
# Полезность администратору

На однажды настроенном сервере можно запускать что угодно:

- Приложения больше не могут между собой конфликтовать
- Не надо думать о трёхстраничной инструкции по установке всех необходимых зависимостей
- Все приложения выглядят одинаково: контейнеру нужен только конфиг и связь с сервисами и другими контейнерами.



## Ехал докер через докер...



# Докер, докер, докер, докер

							
							
SQL 							
SignalR 							
Redis 							
Queue 							
Worker 							



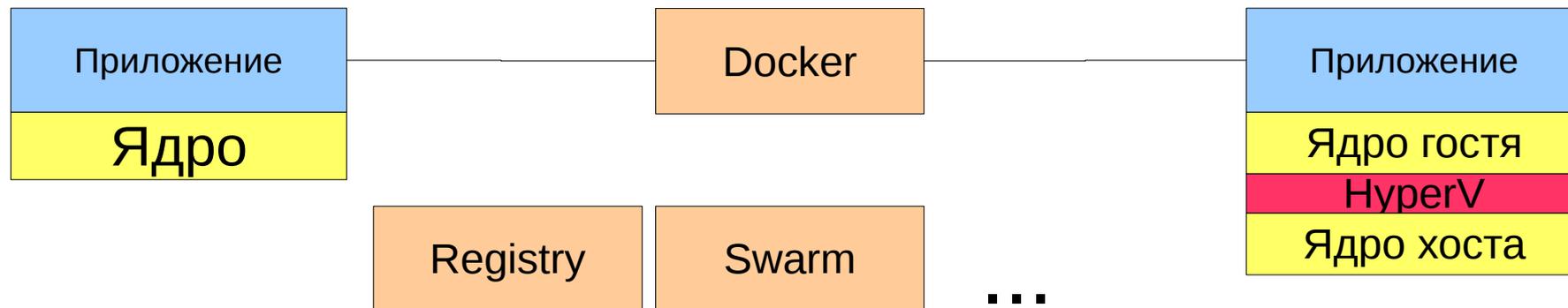
# Типы Windows-контейнеров

## Windows Server Container

- Windows Server Core
- Быстро запускается
- Требуется меньше ресурсов

## Hyper-V Container

- Windows Server Nano
- Быстро запускается
- Требуется меньше ресурсов



# Хранилища

## Директория в ФС хоста

- удобен доступ на редактирование со стороны хоста
- При присоединении используется пусть в ФС

## Volume container

- Volume доступен по имени, расположение данных контролируется Docker
- Файлы из контейнера автоматически копируются в volume, если та изначально пуста



# Docker Compose

version: '2'

services:

**web:**

image: web

ports:

- "5000:5000"

volumes:

- .:c:\data

depends\_on:

- **redis**

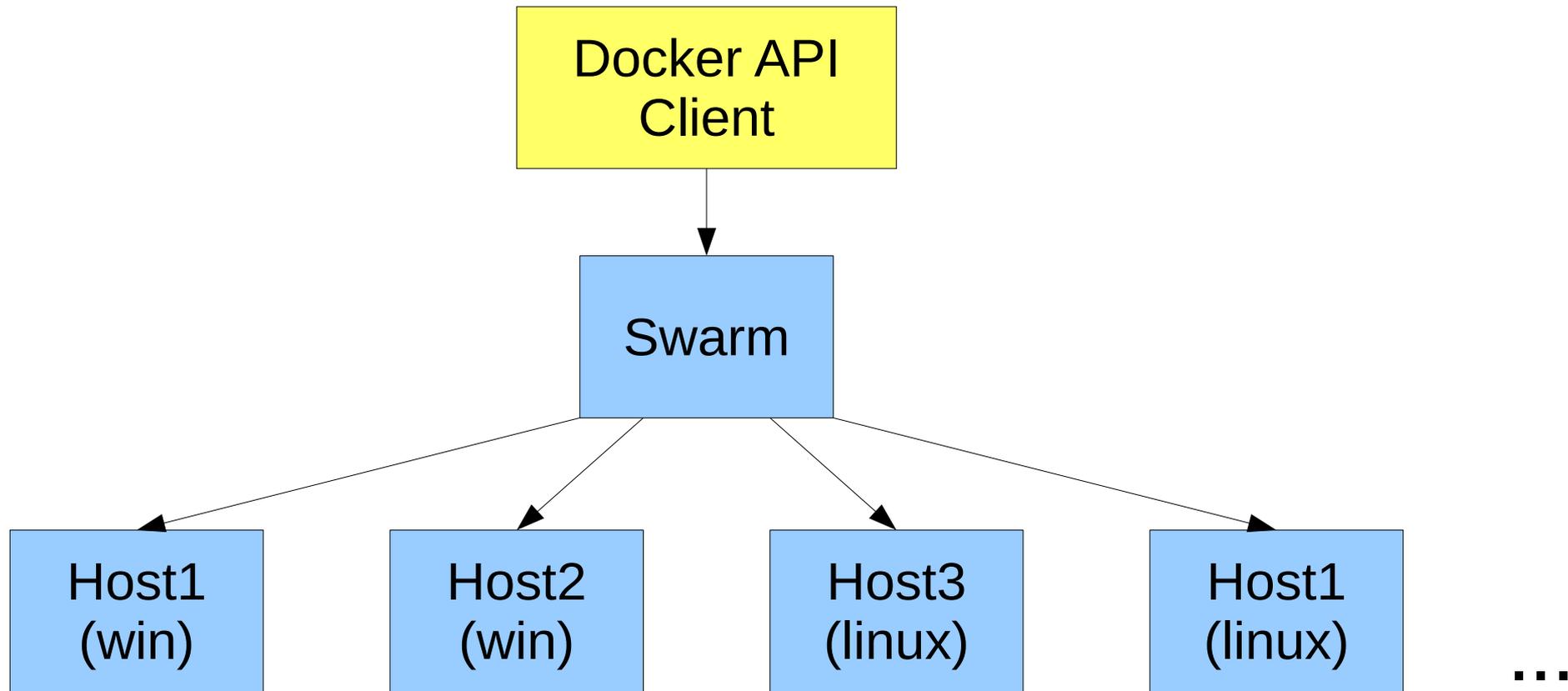
**redis:**

image: redis





# Docker Swarm

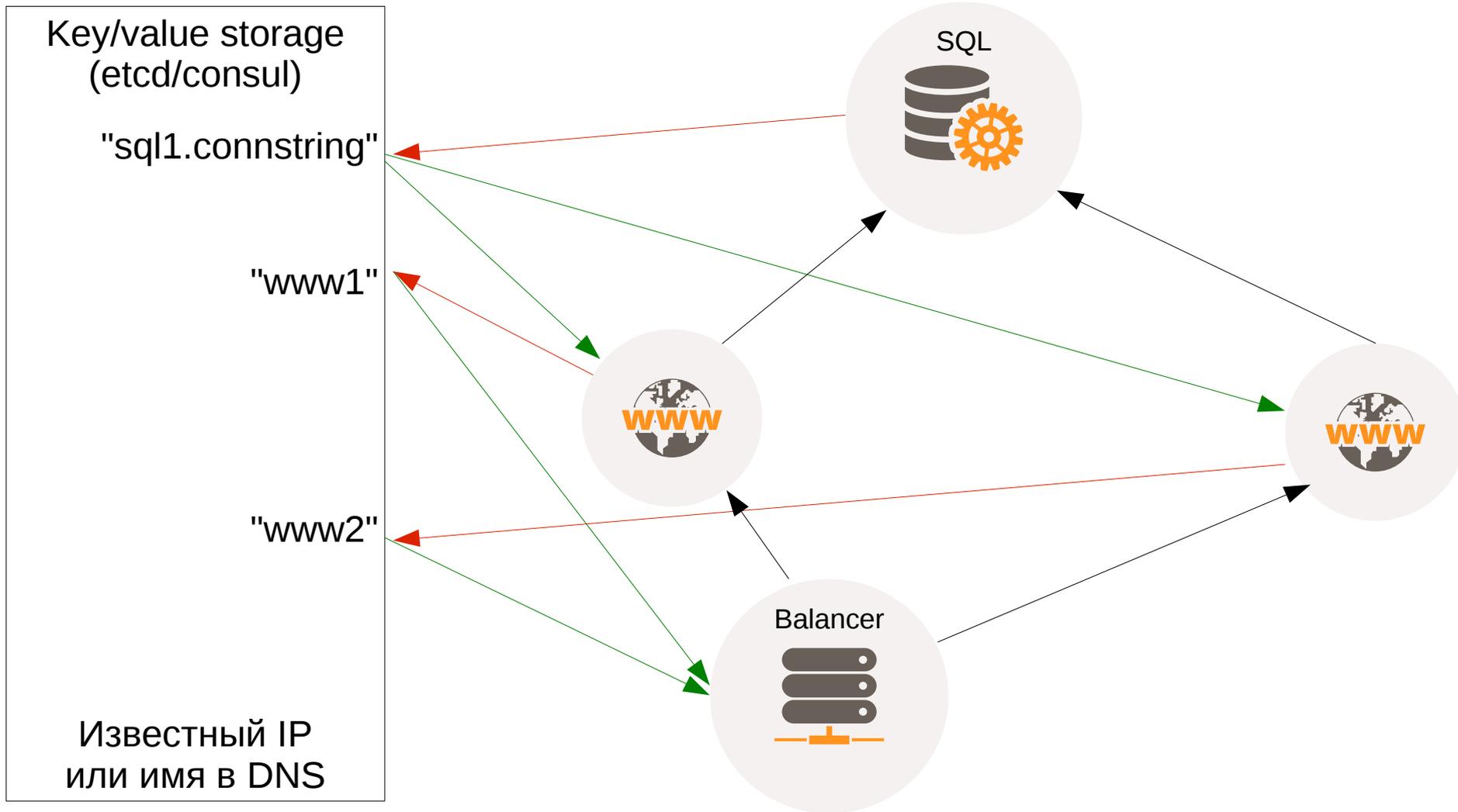


# Сети

- NAT (по-умолчанию) — контейнеры находятся в отдельной от внешнего мира сети, хост является шлюзом
- Bridge — контейнеры подключены к сетевому мосту и видны из внешней сети
- Internal — приватная сеть для группы контейнеров
- Overlay — приватная сеть для группы контейнеров на разных хостах (пока не доступно для Windows)



# Discovery



# Обновление софта в контейнерах

**Проблема:** софт и библиотеки в базовых контейнерах устаревают и их никто не обновляет.

**Решение:**

- Не использовать поддерживаемые «непонятно кем» базовые образы, только образы базовых систем или доверенных издателей (например образы от Microsoft, Canonical, RedHat)
- Пересобирать свои образы с часто используемым набором компонент на регулярной основе



Демо, полезные ссылки, презентация:

<https://github.com/kekekeks/dotnext2016spb>

