

SOA. Разработка, Деплой и Мониторинг

Пулявин Артемий, VPoE in Unitemp



<http://www.devconf.ru>

План

- Разработка
- Инфраструктура
- Мониторинг
- Что дальше?

Вступление

Проблемы:

- стартап, рынок новый
- пивоты, две штуки
- постоянная проверка гипотез

Необходимо:

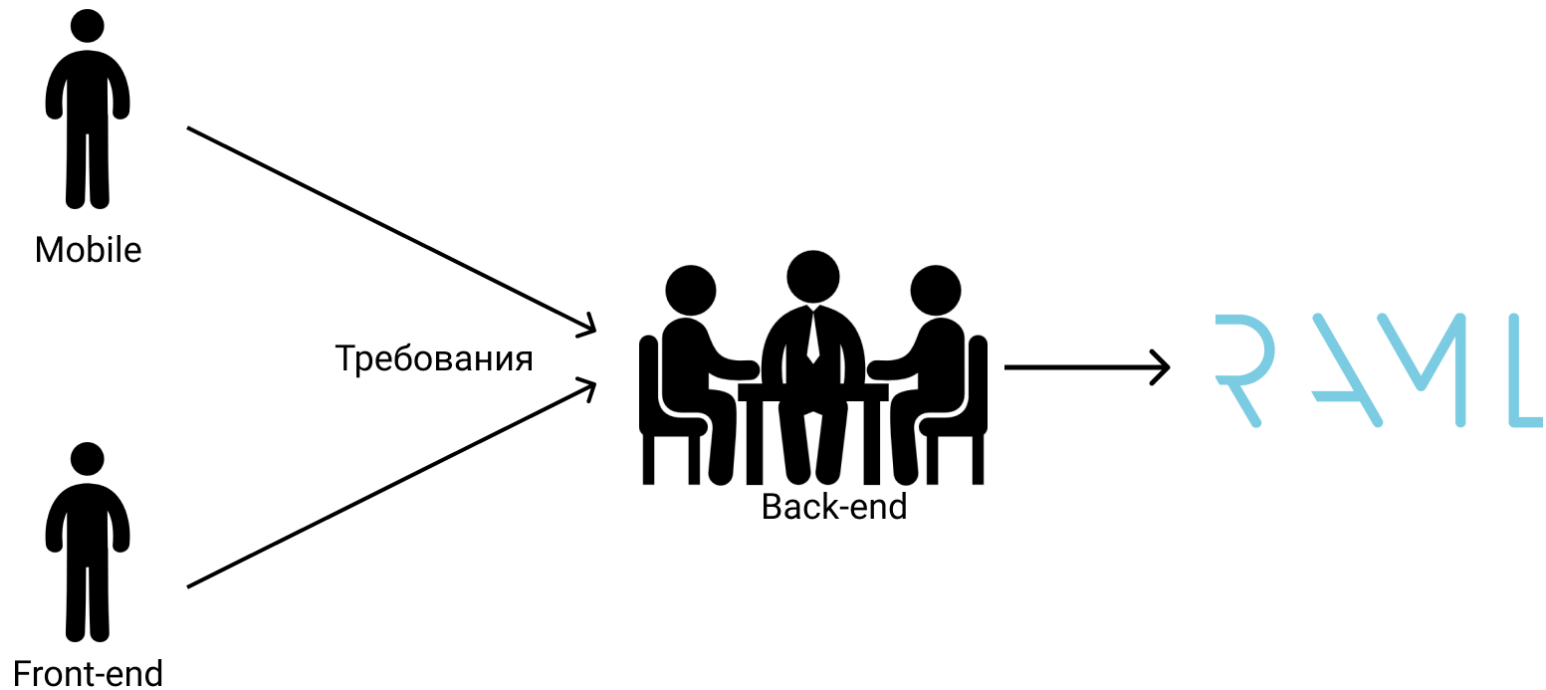
- включение и отключение функциональности
- замена и внедрение новой функциональности
- не обрасти легаси!

Service Oriented Architecture

Сервисы

- Notifier
- Documents
- Files
- Core
- ...

RestAPI



RestAPI

Группы роутов

- client
 - admin
 - server
-
- client и admin доступны извне и могут требовать авторизации
 - server авторизации не требует и находится за firewall

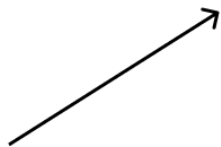
API Gateway



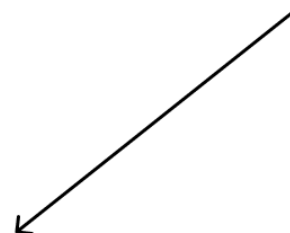
Команда №1



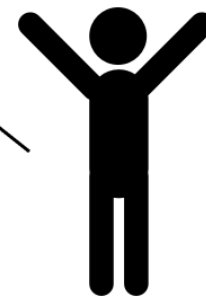
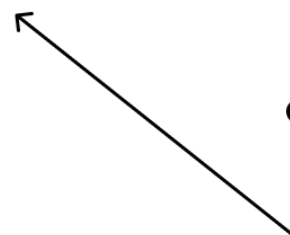
Команда №2



Outer API
Documentation

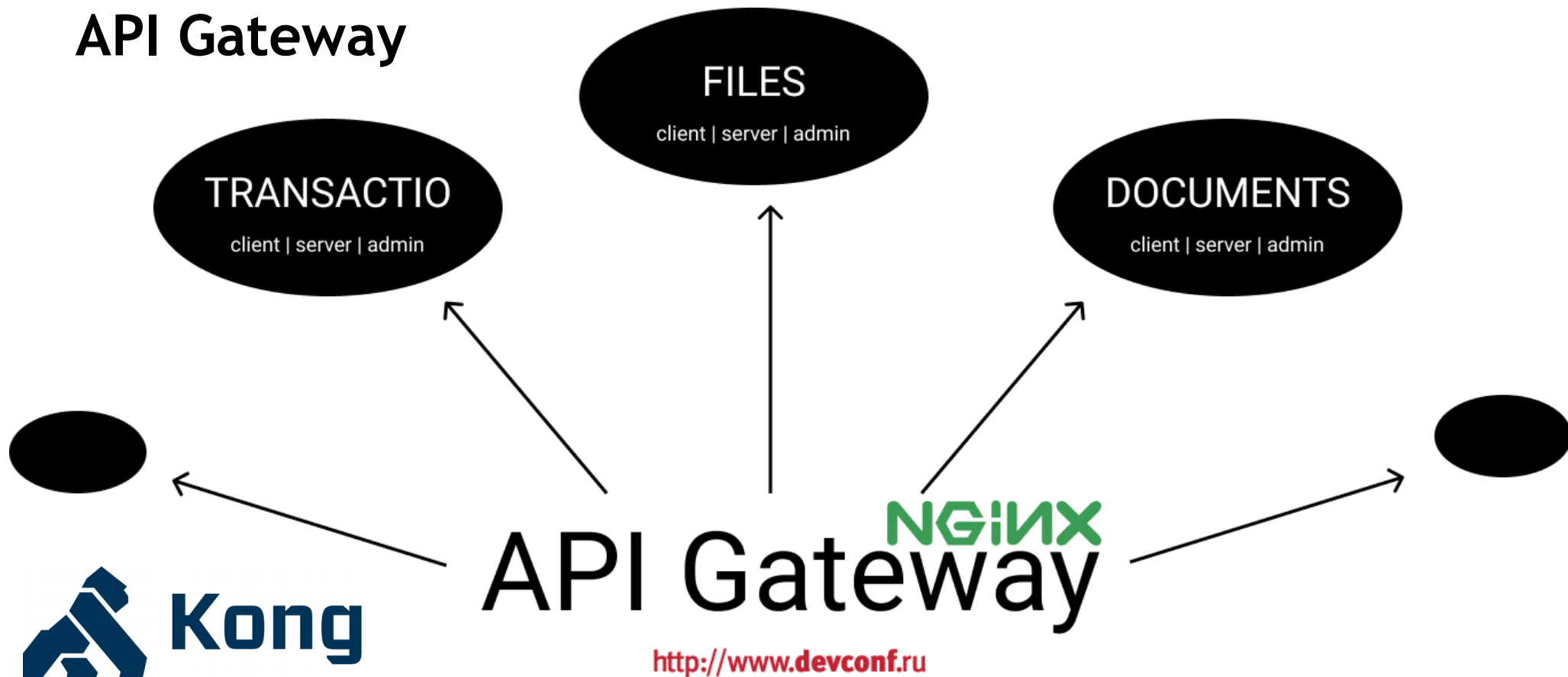


Front-end

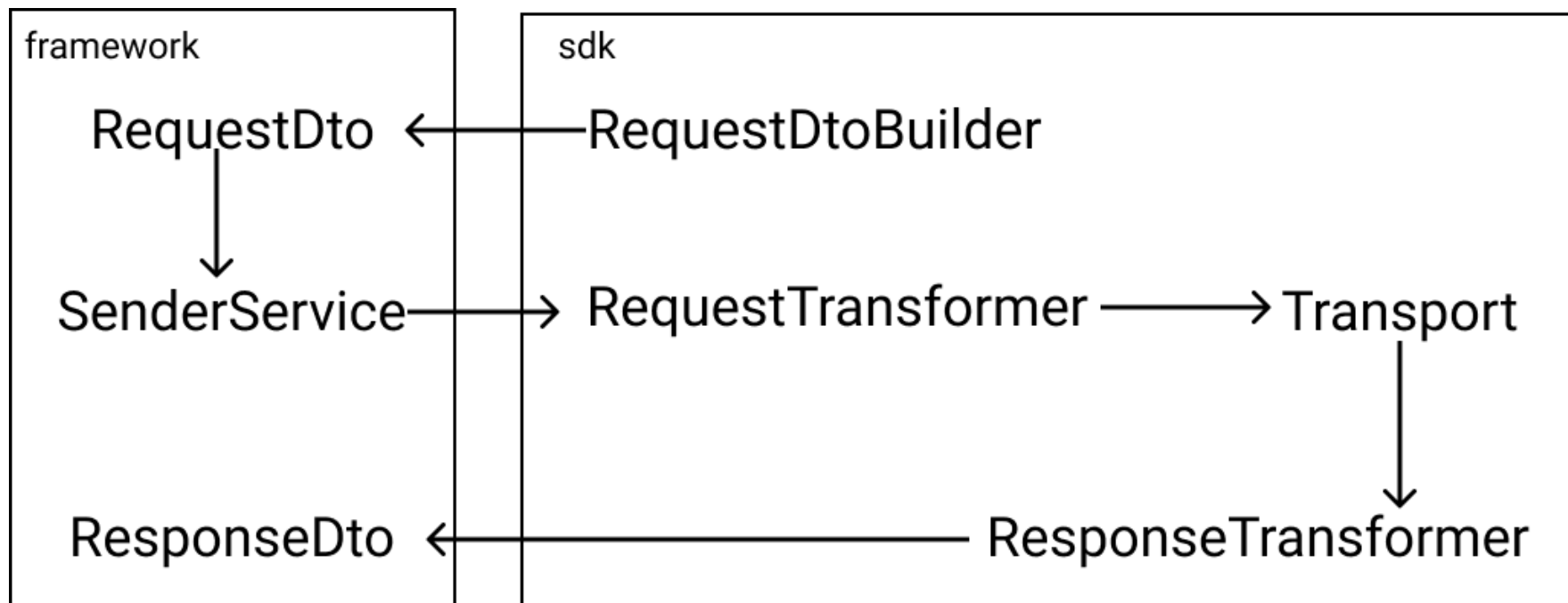


Mobile

API Gateway



SDK



Инфраструктура

- решаем задачу high availability

Из чего состоит любой сервис?

- web морда, обрабатывающая API-запросы
- регулярно выполняющие задачи, cron
- демоны, обрабатывающие очереди

Типы серверов

- web
- bot

каждый сервис имеет как минимум 2 web и 2 bot сервера для отказоустойчивости

Api Gateway

```
upstream notifier-upstream {  
    server notifier-web-01.unitemp.lcl;  
    server notifier-web-02.unitemp.lcl;  
}
```

Изолированность

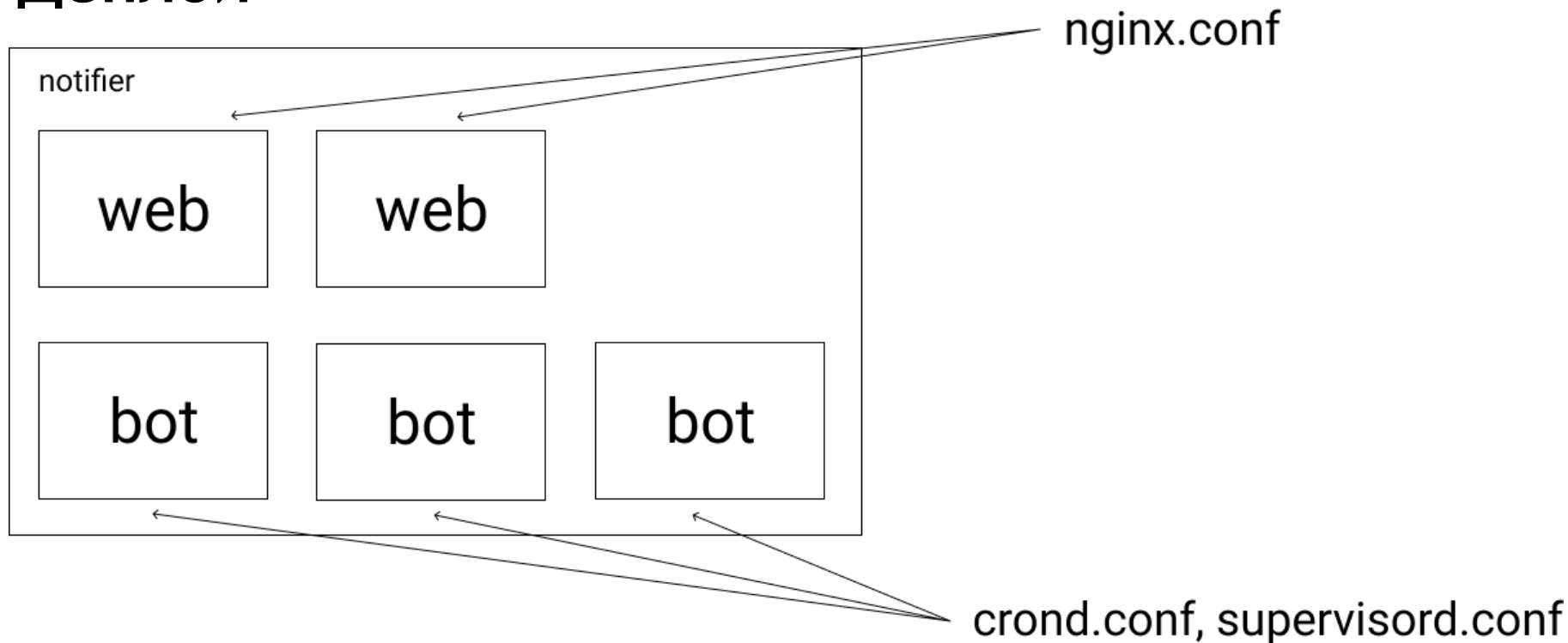
- на KVM не ходим!

Необходимо:

- деплой окружения
- деплой приложения

специальные Ansible playbooks для web- и bot- серверов

Деплой



Как конфигурировали раньше?

- способ первый, простой
- способ второй, просветление
- способ третий, полуавтоматизация

Первый и второй – плохо, третий – актуально

DSL?

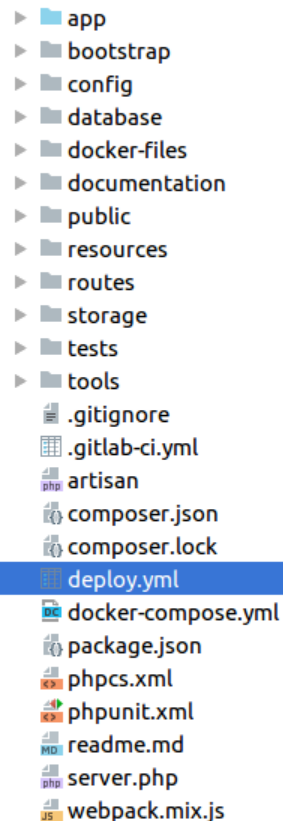
Плюсы DSL

- любой разработчик может в любой момент времени посмотреть что нужно для работы сервиса в той или иной ветке кода, а не дёргать DevOps'а с расспросами текущего состояния конфигов;
- devops не обременён никому не нужной работой.

deploy.yml

Поддерживаем описание для:

- nginx
- crontab
- supervisord



deploy.yml

```
supervisor:  
  - command: "{{ release_path }}/artisan queue:work --queue=core.invoice_notification --delay=3 --tries=5"  
    instances: 2  
    type: mirroring  
    name: invoice-notification  
  - command: "{{ release_path }}/artisan queue:work --queue=core.reports --delay=3"  
    instances: 10  
    type: balancing  
    weight: 2  
    name: reports
```

Типы:

- balancing (задача балансировки)
- mirroring (задача отказоустойчивости)

deploy.yml

crontab:

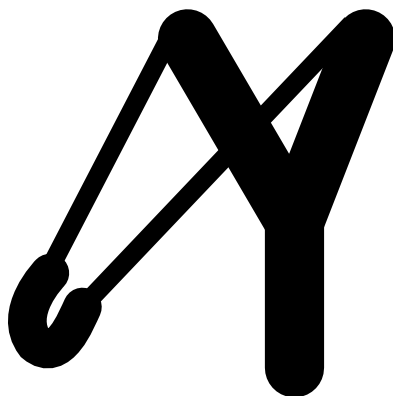
```
- command: "{{ release_path }}/artisan invoice:send"
  planning:
    minutes: "5"
    hours: "0"
    dom: "*"
    mon: "*"
    dow: "*"

- command: "{{ release_path }}/artisan recurrent:watcher"
  planning:
    minutes: "*/5"
    hours: "*"
    dom: "*"
    mon: "*"
    dow: "*"
```

аккуратно “размазывается” по всему кластеру

deploy.yml

```
nginx:  
  root: "{{ release_path }}/public"  
  index_files:  
    - "index.php"  
  locations:  
    - match: '/api/v1/'  
      configs:  
        proxy_pass: "https://api.unitemp.ru/api/v1/"
```

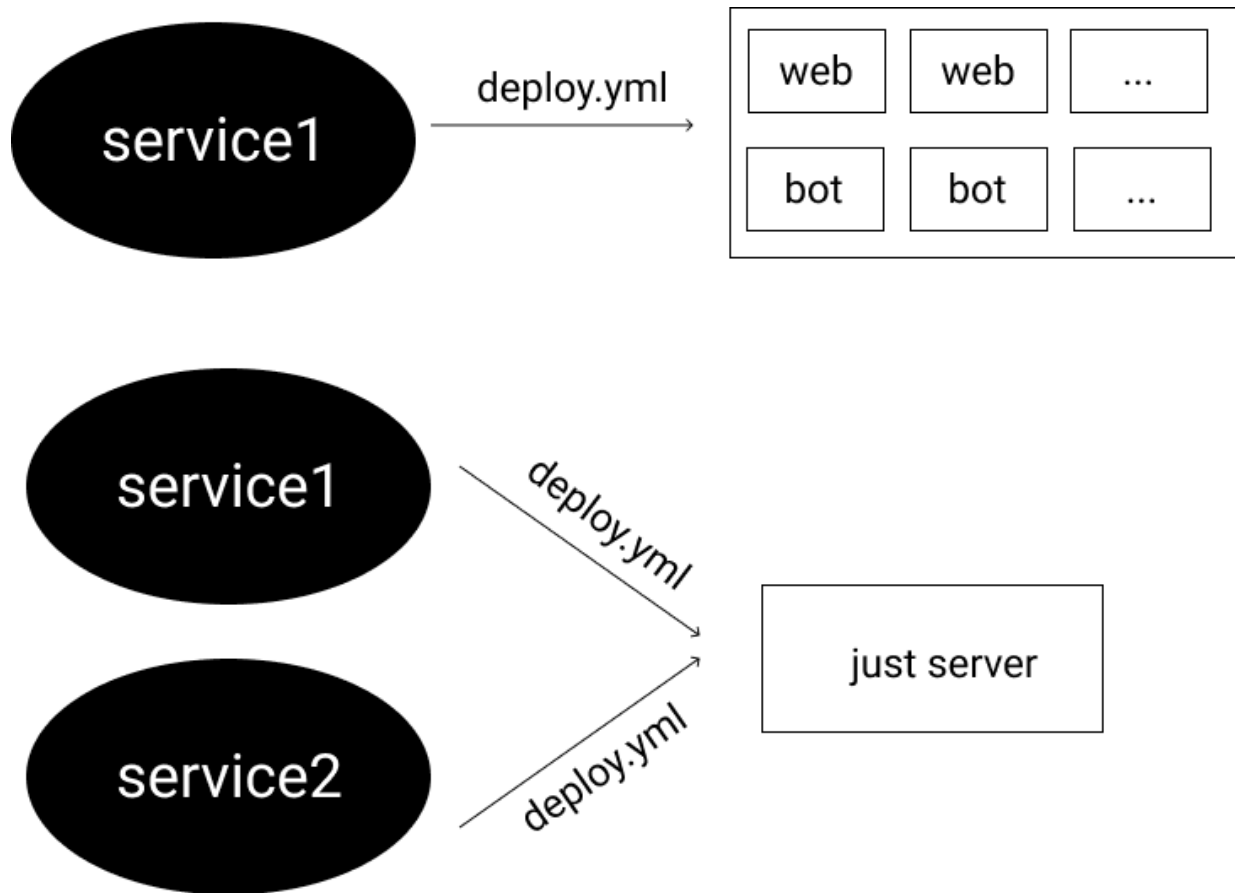
Catapult - continuous delivery platform

Catapult

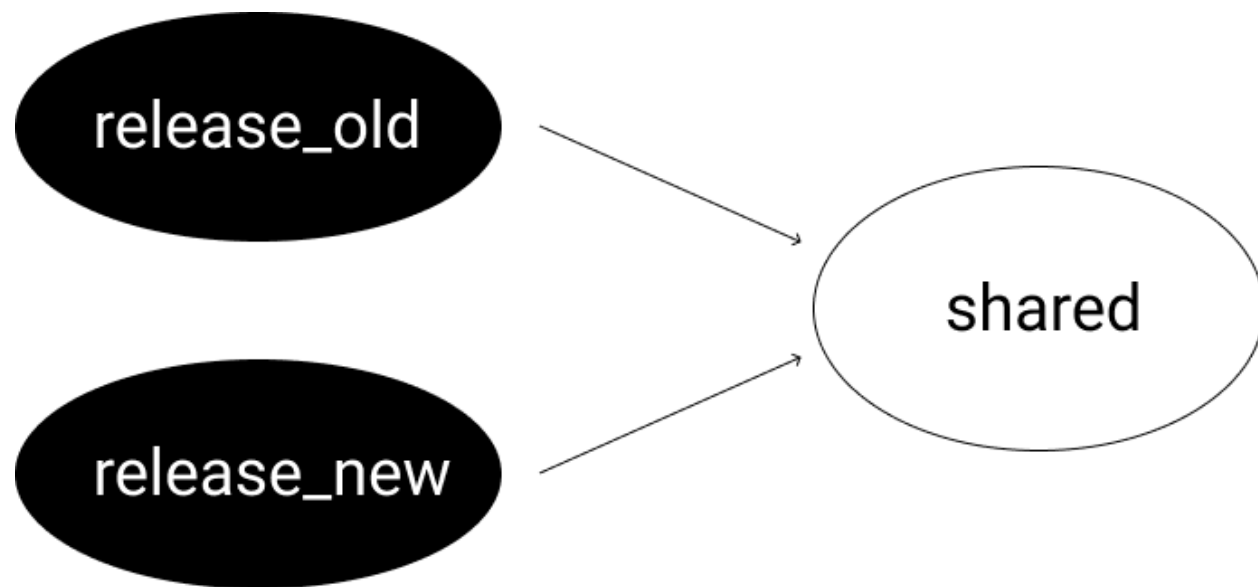
service: core
branch: master
place: prod



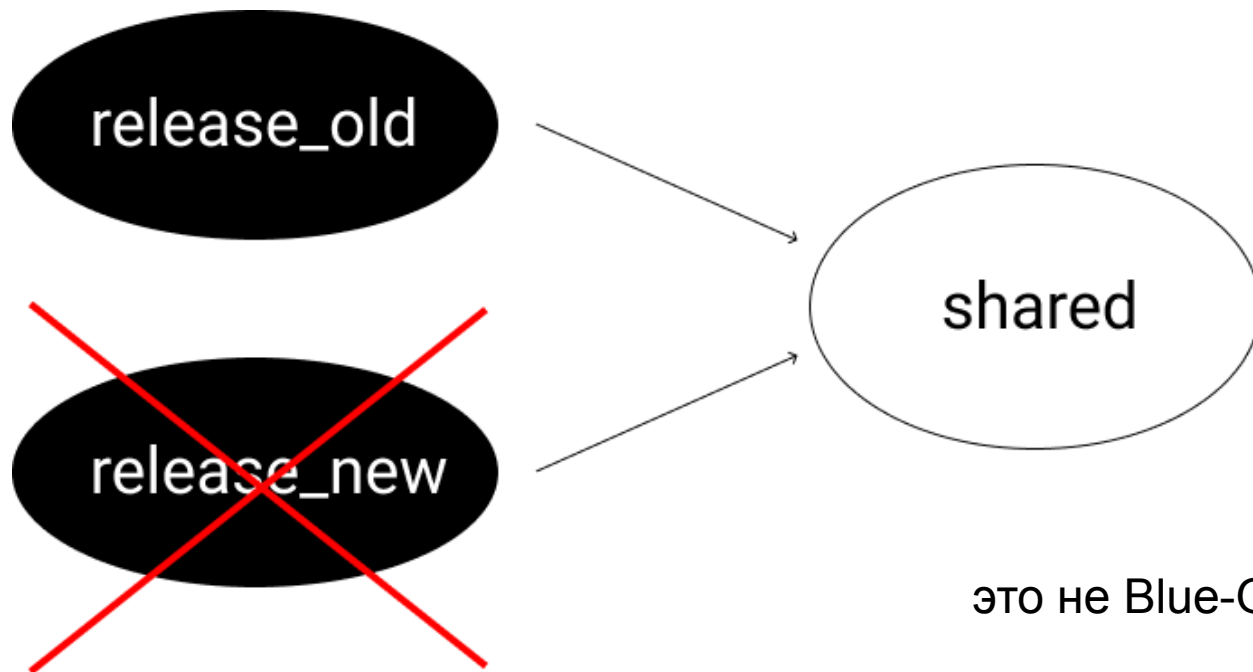
Catapult



Проблемы деплоеров



Откатились



это не Blue-Green deploy!

Этапы

- started
- lock
- configure
- build
- check
- release
- maintain
- install
- clean



Monitoring

Health-check

- какая часть системы сейчас не работает
- какая часть системы сейчас работает плохо

Prometheus vs. Zabbix

Prometheus

```
service_rabbitmq{service="notifier"} 53  
service_redis{service="notifier"} 13  
service_postgres{service="notifier"} 34  
service_self{service="notifier"} 1
```

Grafana



+ кастомные метрики для бизнеса

И так,

- окружение поднимаем через Ansible
- деплоим приложение через Catapult
- метрики снимаем через Prometheus

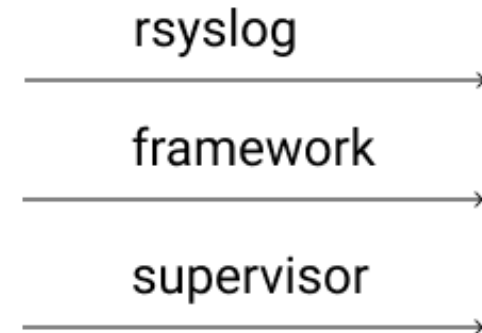
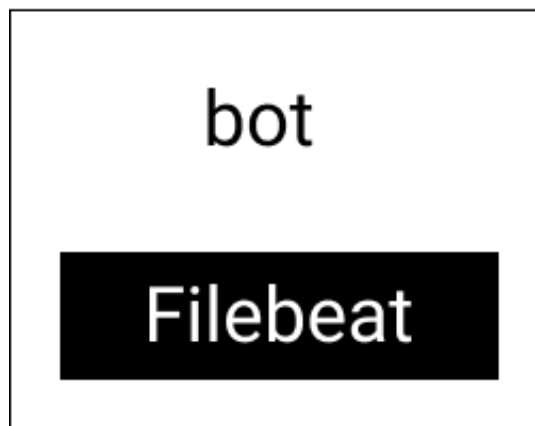
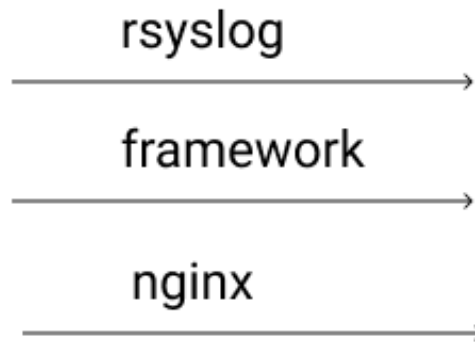
Остаётся...

Logging

Logging

- на web-серверах логи пишут nginx, php-fpm и сам сервис;
- на bot-серверах логи пишут supervisor и сам сервис;
- не плохо было бы ещё и syslog забирать.

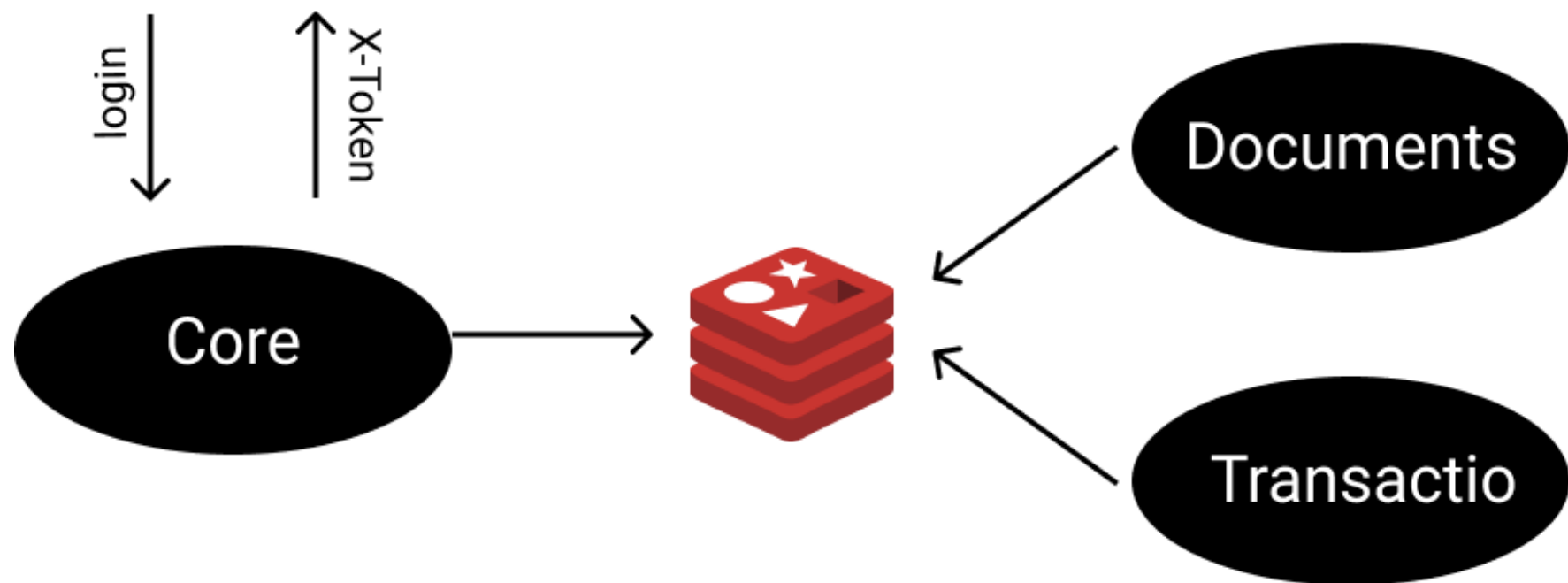
ELK

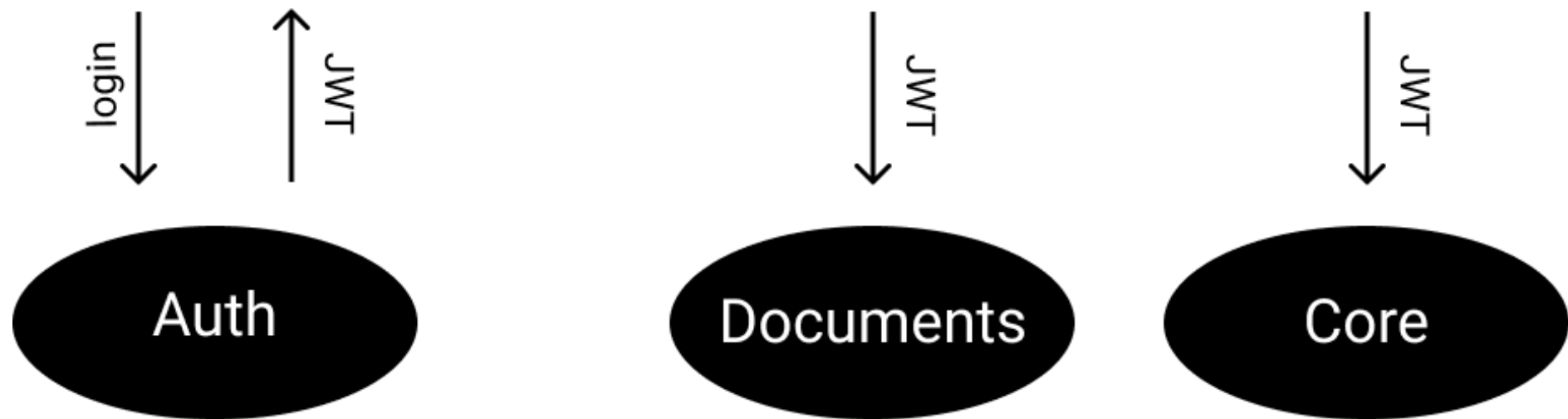


+ hash or stack trace

Что дальше?

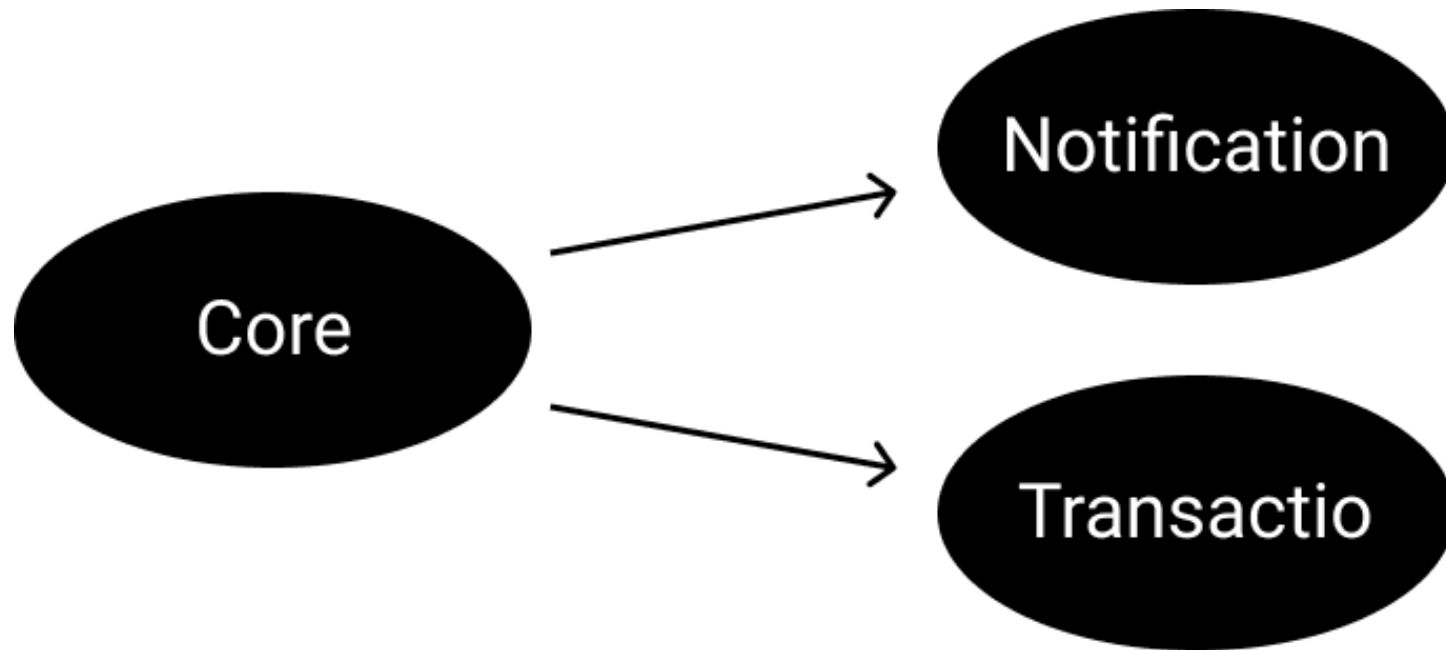
Auther Service



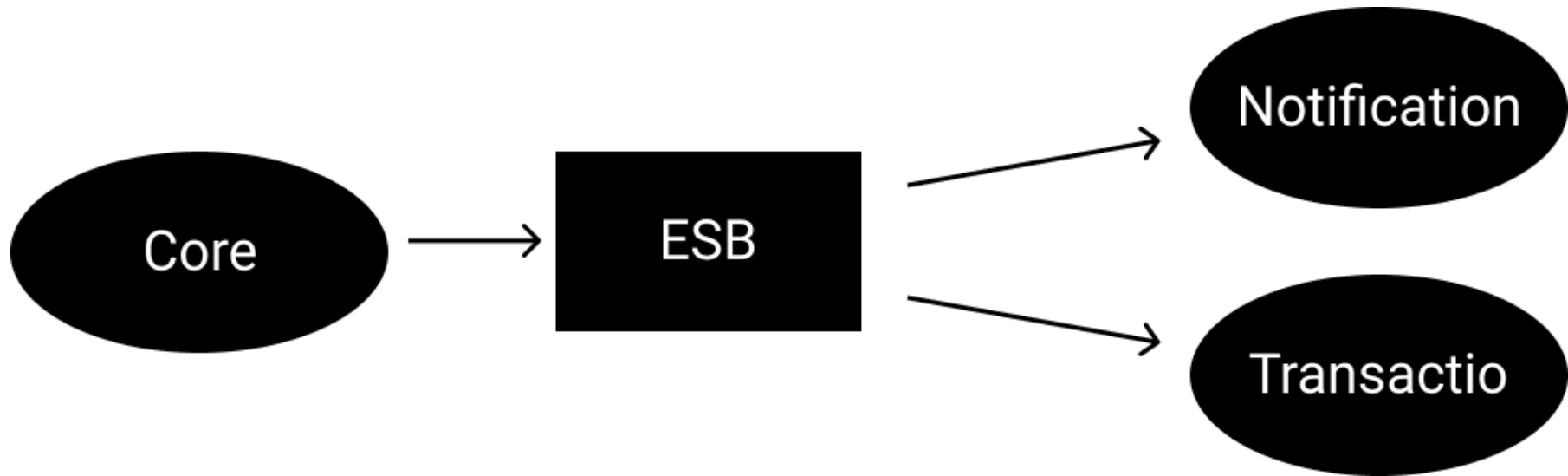


Enterprise Service Bus

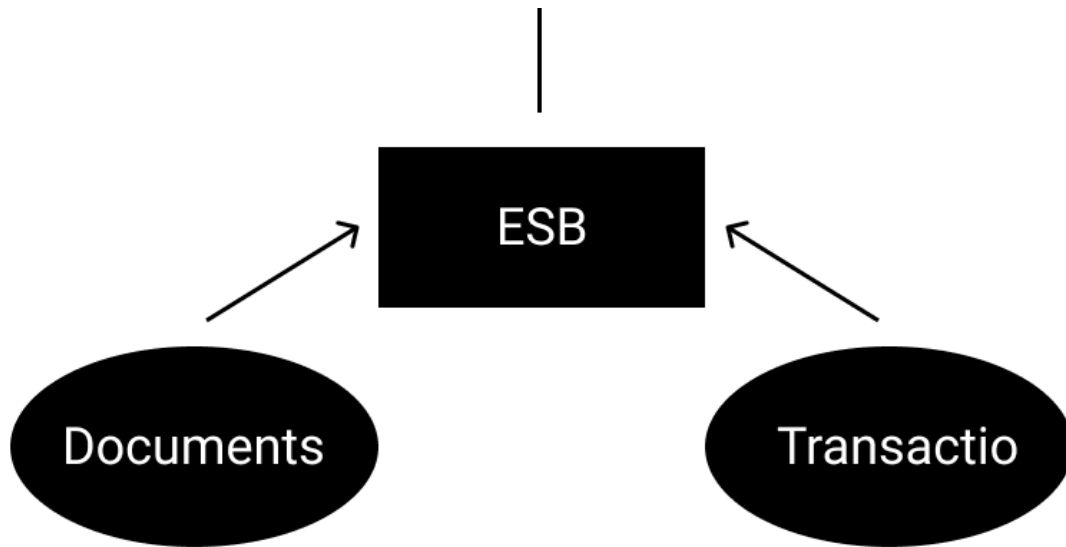
ESB



ESB



ESB



```
{  
  "passport": {  
    "surname": "Константинопольский",  
    "name": "Константин",  
    "patronymic": "Константинович"  
  }  
}
```

```
{  
  "address": "1Yu2BuptuZSiBWfr2Qy4aic6qEVnwPwrkHPEc",  
  "balance": 1000.00  
}
```


ESB

```
{  
  "address": "1Yu2BuptuZSiBwfr2Qy4aic6qEVnwPwrdkHPEc",  
  "balance": 1000.00,  
  "passport": {  
    "surname": "Константинопольский",  
    "name": "Константин",  
    "patronymic": "Константинович"  
  }  
}
```

API Gateway 

Вопросы?

Артемий Пулявин
artemiy@pulyavin.ru