

Введение в мониторинг. Системы мониторинга.

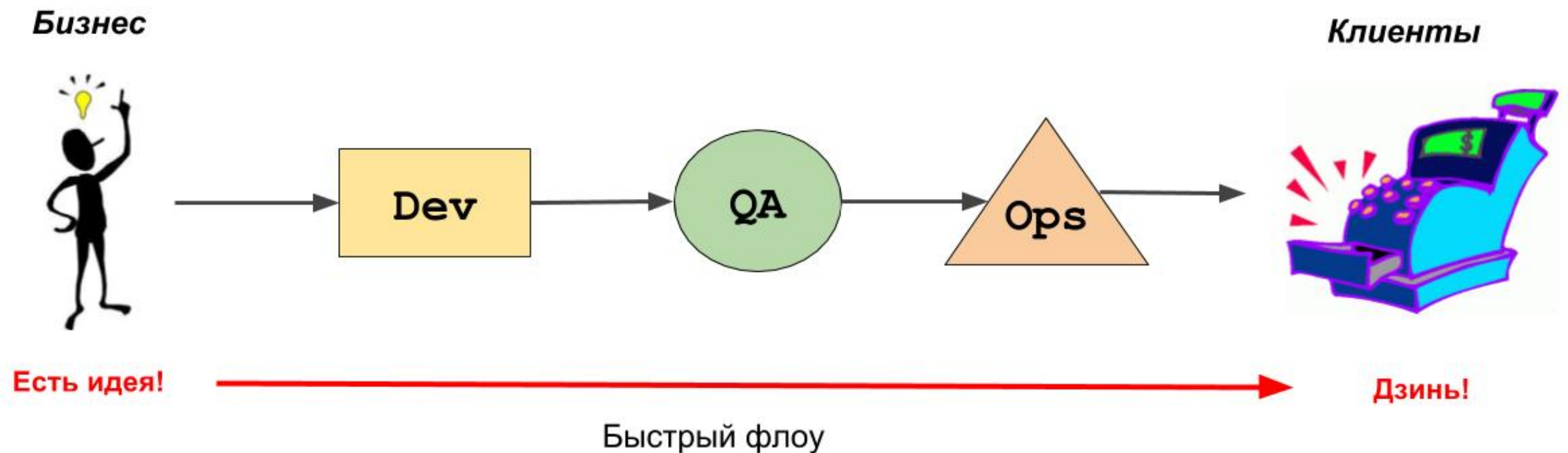
План

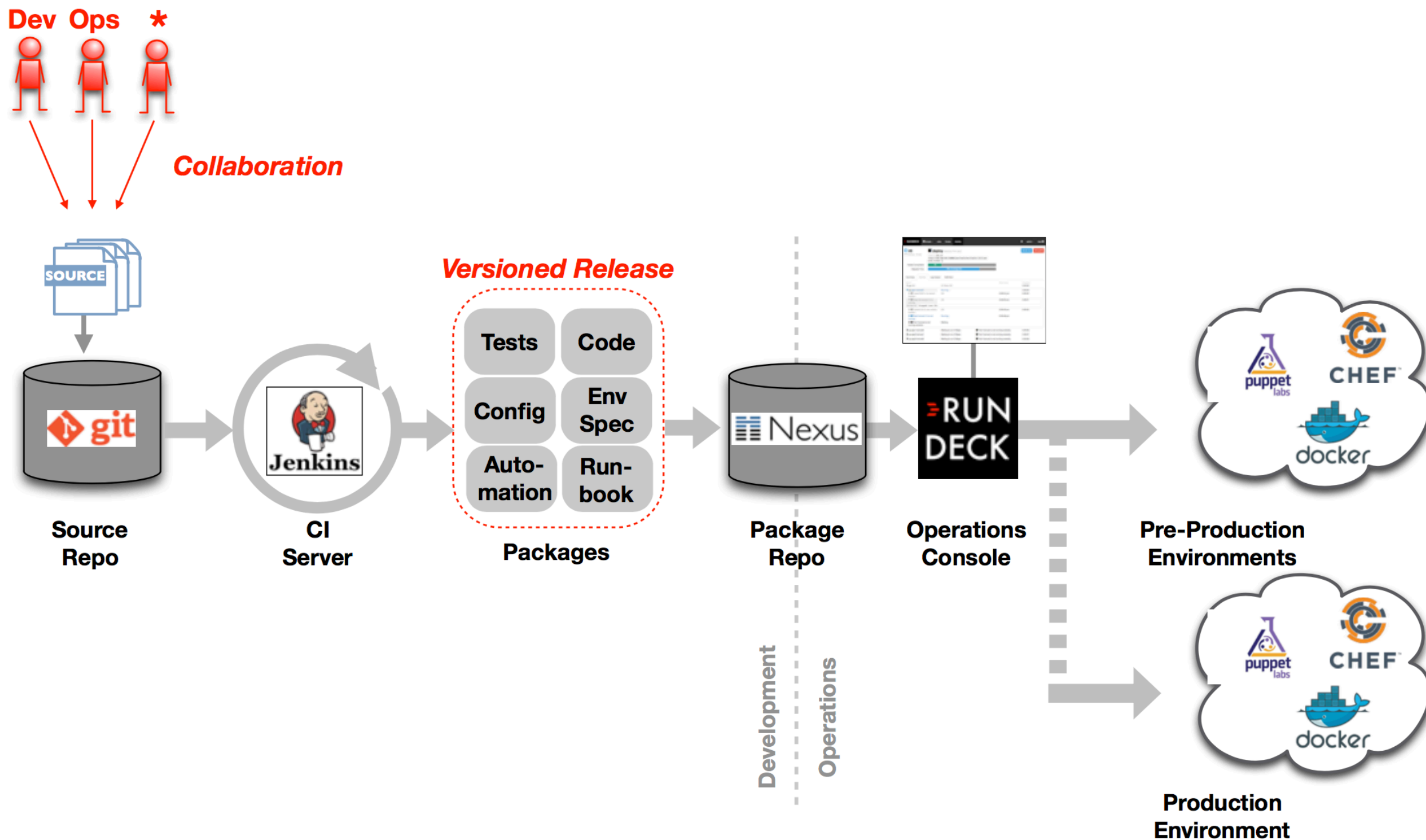
- Три пути DevOps
- Мониторинг
- Модели мониторинга
- Принципы работы систем мониторинга
- Prometheus

Три пути DevOps

Первый путь

- Обеспечить поток работы слева на право



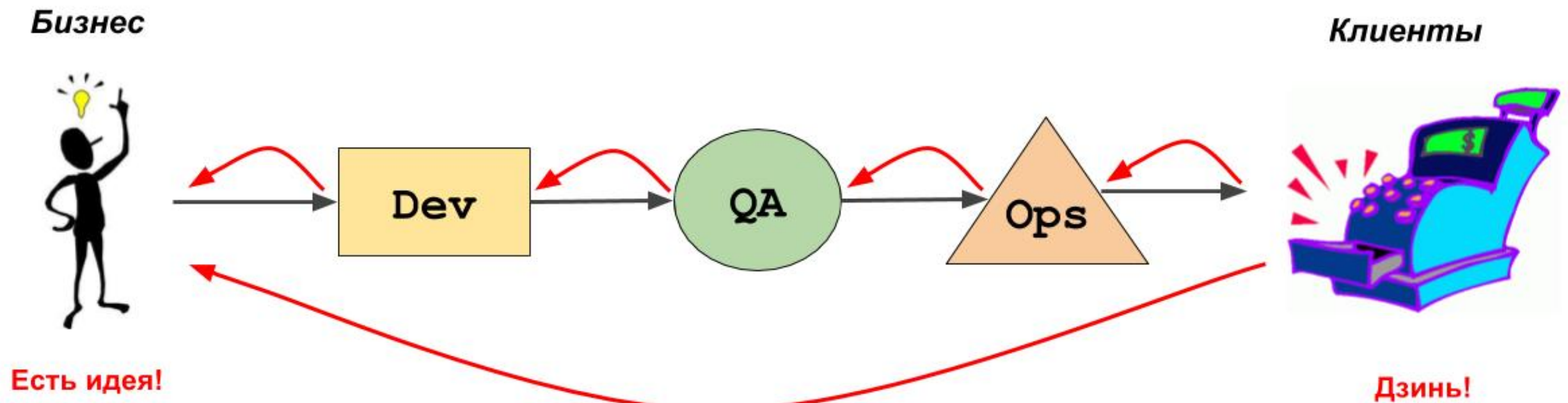


Что дальше?

- ...
- ...
- ...

Второй путь

- Обеспечить быструю обратную связь справа налево



Плюсы быстрой обратной связи

- Повышает качество продукта
- Позволяет оперативно реагировать на значимые события/изменения
- Повышается скорость обучения внутри организации

Примеры быстрой обратной связи

- Автоматизация сборки кода и прогона тестов
- ChatOps
- Мониторинг окружений и работы приложения
- Мониторинг поведения пользователей

Третий путь

- Создание культуры постоянного эксперимента и обучения внутри организации

Мониторинг

Зачем?

Надежность имеет значение

Никто не будет пользоваться сайтом, если сайт:

- не работает
- работает медленно
- работает с ошибками

Надежность работы продукта становится залогом его успеха.

Анализ поведения пользователей

Без анализа поведения пользователей:

- нельзя понять успеха бизнес идей
- нет понимания потребностей пользователей

На рынке выигрывает тот, кто умеет понимать потребности пользователей и реализовывать их в кратчайшие сроки.

Нужен процесс, который ПОЗВОЛИТ...

- Обеспечить контроль над работой наших сервисов
 - Отслеживание состояния и качество работы сервисов и инфраструктуры
 - Система оповещения о проблемах и потенциальных проблемах
- Обеспечить механизм оценки работы бизнес идей

Мониторинг

- совокупность инструментов и практик, позволяющая осуществлять контроль над работой IT систем и давать оценку качества работы этих систем

Мониторинг как сервис

Может быть:

- outsource (ITSumma)
- SaaS (New Relic, DataDog, okmeter)
- собственный сервис (Zabbix, Prometheus, TICK)

Должен удовлетворять потребности своих клиентов:

- dev
- ops
- business

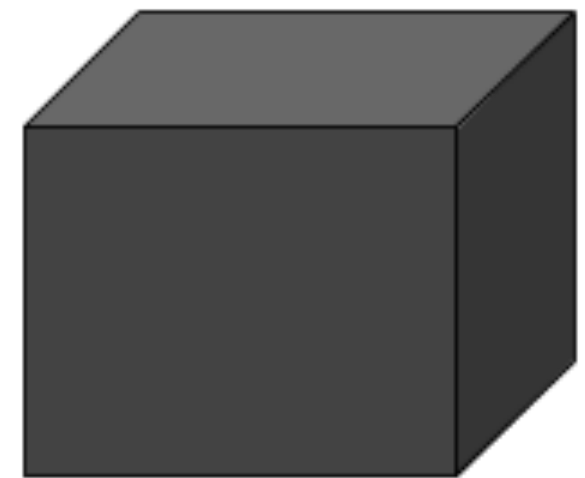
Что мониторить?

- Инфраструктура
- Работа и состояние поддерживающих сервисов
- Процесс деплоя
- Работа и состояние приложения
- Бизнес метрики, поведение пользователей

Модели мониторинга

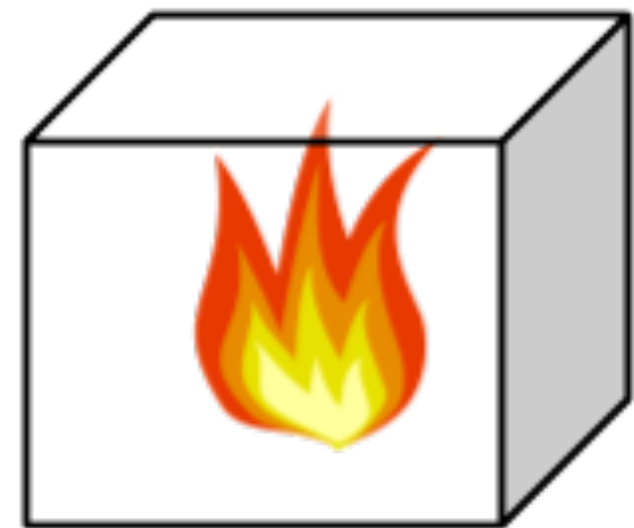
Blackbox мониторинг

- Мониторинг извне с точки зрения пользователя
- Не видим, как работает система внутри
- Примеры: ping, HTTP запрос



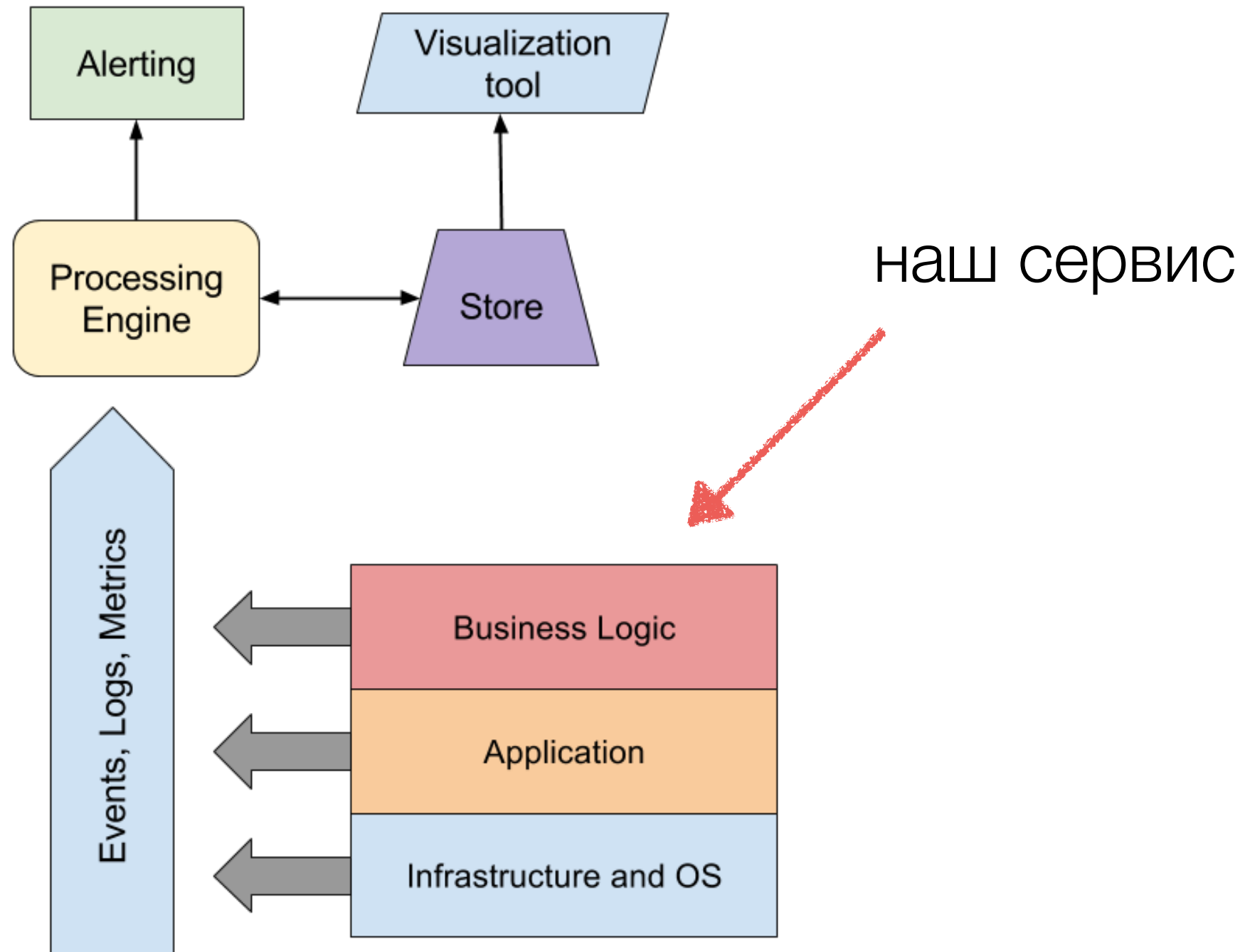
Whitebox мониторинг

- Дополняет модель blackbox
- Мониторинг на основе информации о внутренней работе системы
- Примеры: CPU, время запроса к БД



Системы мониторинга

Архитектура



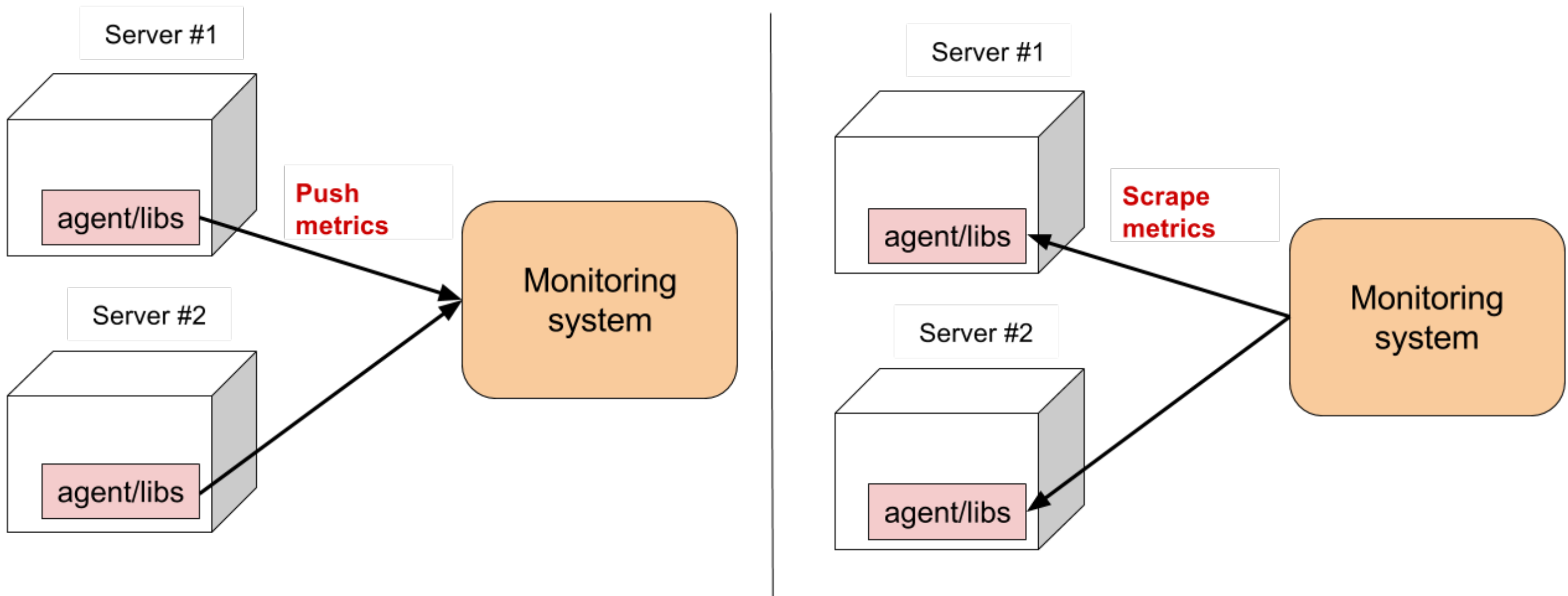
Основные компоненты

- Средства сбора и отправки данных: агенты, API libs
- Хранилище
- Инструмент обработки данных
- Инструмент визуализации и анализа
- Система оповещения

Примеры систем мониторинга

- Nagios
- Sensu
- Riemann
- TICK
- Zabbix
- Prometheus
- etc

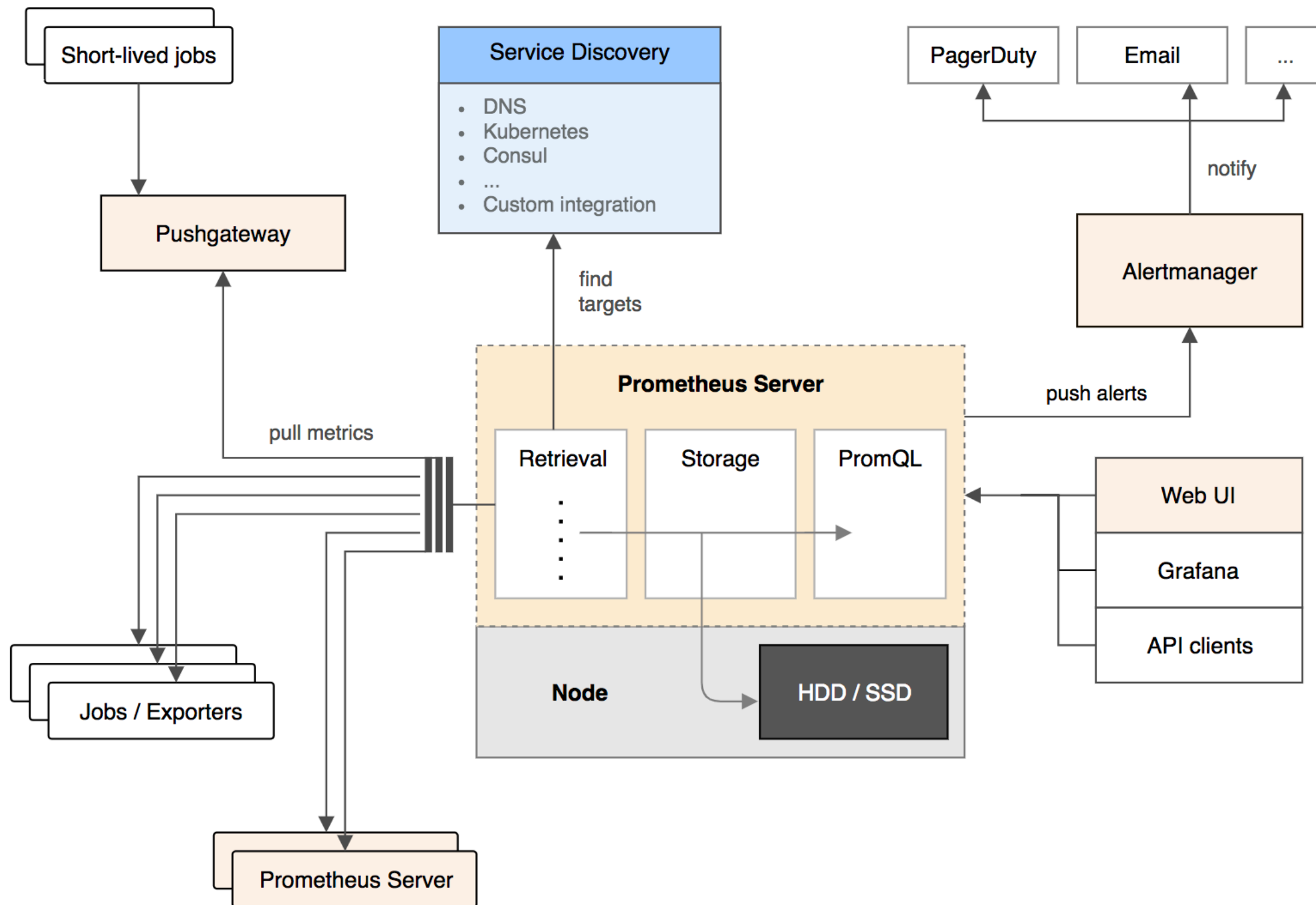
Push vs Pull



Prometheus



- Развитие проекта началось в 2012
- Создан на основе Borgmon, бывшими работниками Google
- Open source
- Написан на Go
- Whitebox, Pull система



Jobs & targets

- Targets (endpoint) - источник для сбора метрик
- Группы источников объединяют в jobs

Exporter

- Программа, которая делает метрики доступными для сбора Прометеем
- Дает возможность конвертировать метрики в нужный для Прометея формат
- Используется когда нельзя поменять код приложения
- Примеры: PostgreSQL, RabbitMQ, Nginx, Node exporter, cAdvisor

Конфигурация

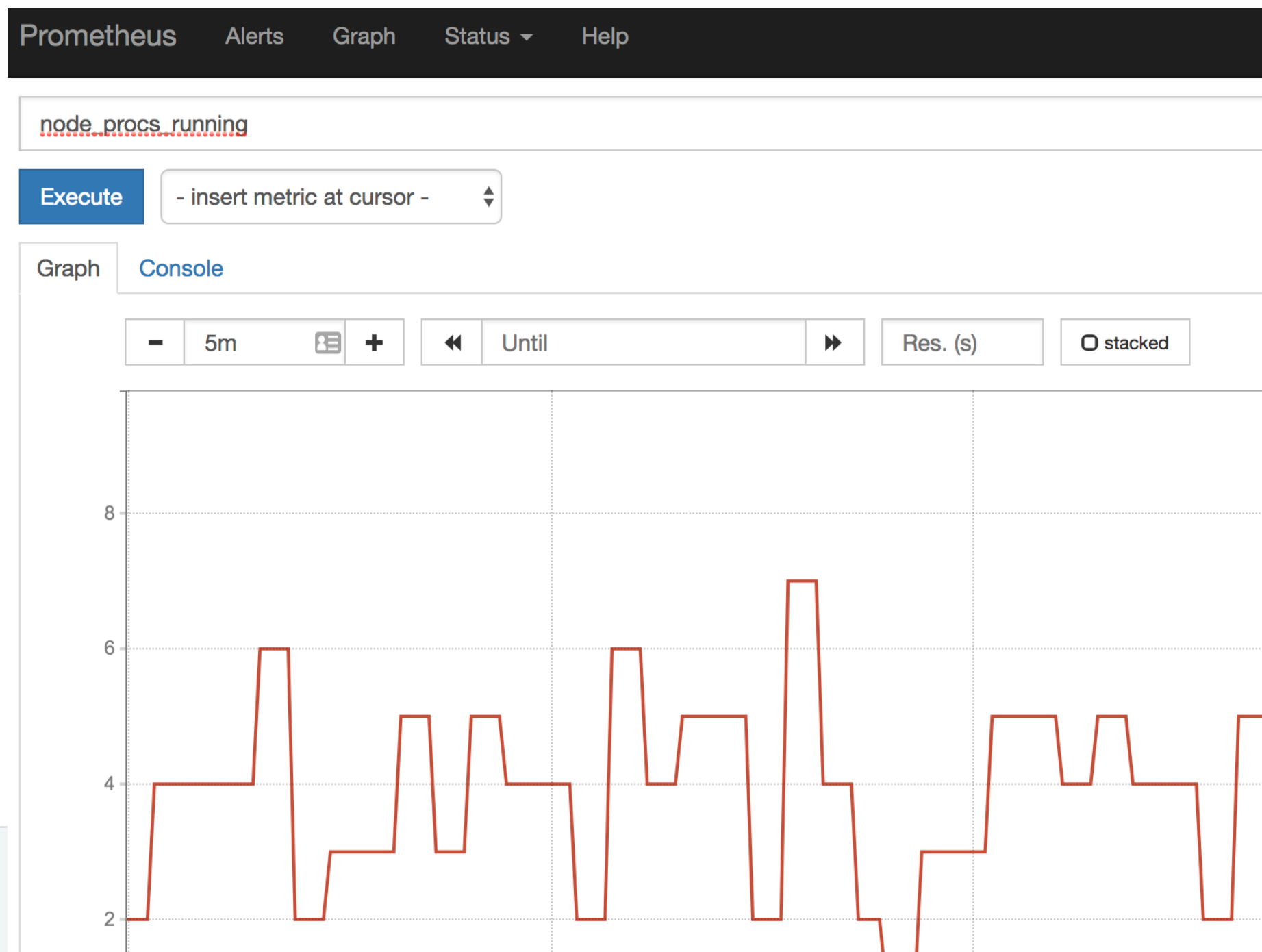
prometheus.yml

```
global:
  scrape_interval: "5s"

scrape_configs:
  - job_name: "prometheus"
    static_configs:
      - targets:
        - "localhost:9090"

  - job_name: "reddit"
    static_configs:
      - targets:
        - "172.18.0.12:9292"
```

Пользовательский интерфейс



To be continued...