

Skillbox

# Observability

Анти-паттерны  
Будущее мониторинга

Андрей Гордиенков

Solution Architect

ABAX

# На прошлом уроке

- Что такое трассировка событий
- Как организована трассировка
- Инструментарий

# На этом уроке

- Анти-паттерны
- Ожидания от будущего

# Анти-паттерны

- Одержимость инструментами
- Мониторинг на полной ставке
- Мониторинг для галочки
- Информационный костыль
- Ручная конфигурация



# Одержимость инструментами

Постоянное внедрение и отмена инструментов для обеспечения Наблюдаемости в системе.

- Идеального инструмента не существует. Характеристика Наблюдаемости – это культура в компании.
- Распознавайте карго-культ и маркетинг конференций.
- Иногда действительно надо сделать свой инструмент для мониторинга
- Единая отправная точка (не панель управления или инструмент)

# Мониторинг на полной ставке

Оператор мониторинга – не профессия

- Все в какой-то мере должны быть вовлечены в процесс мониторинга.
- Сервис не готов к реальному использованию, если он не инструментирован.



# Мониторинг для галочки

Когда мониторинг только для формальной отчетности регулирующим органам.

- Разберитесь, что в вашем случае означает «работающий сервис».
- Метрики ОС малополезны для уведомлений
- Сбор метрик должен быть частым, максимальный интервал 60 секунд.



# Информационный костыль

Средства мониторинга добавляются для обнаружения ошибок, вместо их исправления.

- Надо исправлять причины ошибок, а не создавать уведомления вокруг них, для перезапуска сервиса.





# Ручная конфигурация

Ручное добавление компонентов к системе мониторинга – не ваш выбор.

- Мониторинг облачных систем оперирует с целыми классами различных подсистем.
- Новые сервисы и компоненты должны автоматически встраиваться в систему мониторинга.





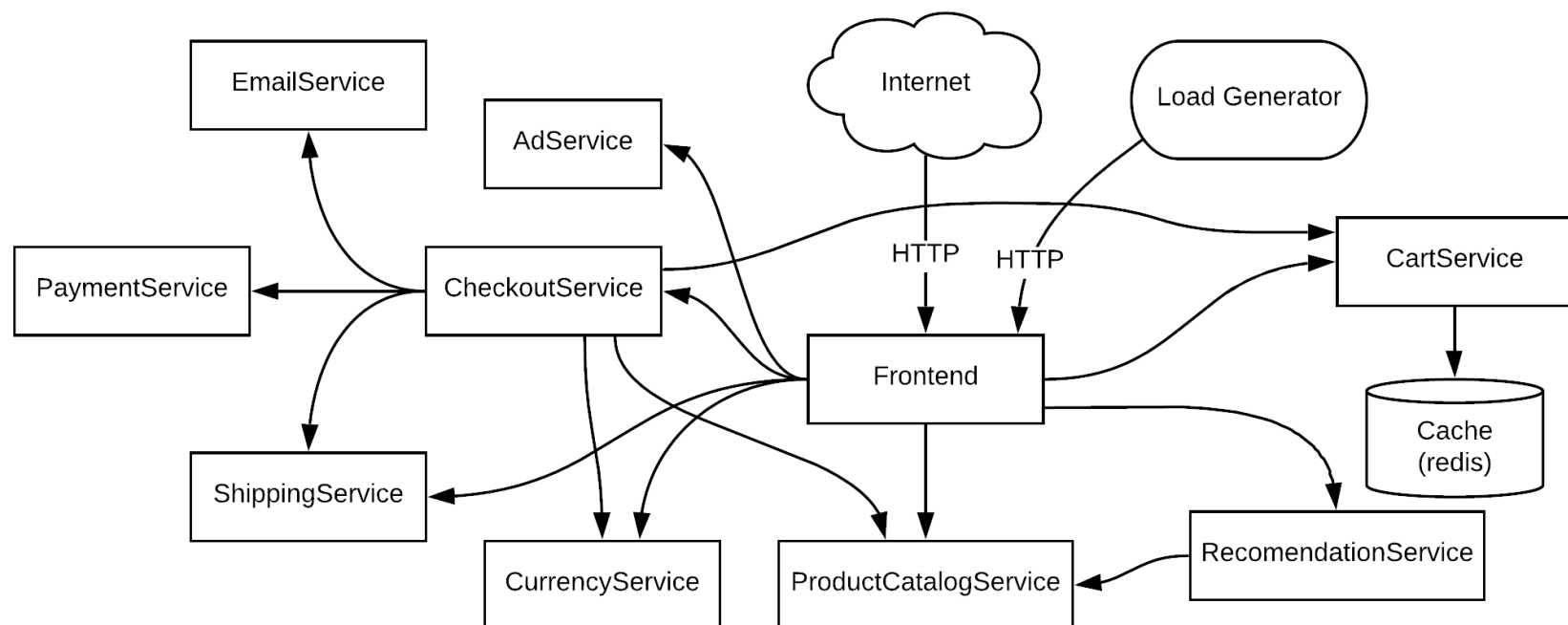
# Будущее

- Трассировку можно сделать более интеллектуальным на основе исторических данных.
- Подготовка специальных представлений для более удобной проверки гипотез во время разборки инцидентов.
- Глубокая интеграция с CI/CD.
- Кросс-отслеживание технических и бизнес данных, для понимания характера изменений.

# Пример

## *Hipster Store*

11 сервисов на различных языках демонстрируют пример использования Kubernetes/GKE, Istio, Stackdriver, gRPC и OpenCensus.



<https://github.com/googlecloudplatform/microservices-demo>

# FAQ

*Можно ли хранить трассировку для всего в любой БД?*

На ваш страх и риск:

- Очень высокая мощность данных
- Большие требования по пропускной способности

InfluxDB, Cassandra, Mongo значительно лучше для этого подходят, чем MySQL, Postgress. Но все зависит от траффика и объема данных.

# FAQ

*Мы уже логируем много всего, этого разве недостаточно?*

Если ваши логи можно сгруппировать относительно ID  
изначального запроса, то это уже очень хороший старт.

# Выводы

- Мониторинг начинается на уровне пользователя, а не информационных сервисов и систем.
- Обеспечение характеристики Наблюдаемости – ответственность команды разработки.
- Продумывайте совместимость инструментов мониторинга.

Цель Наблюдаемости заключается не в сборе логов, метрик или трассировок. Она заключается в создании культуры проектирования, основанной на фактах и обратной связи, а затем в распространении этой культуры во всей организации.

# Итоги модуля

- Наблюдаемость – это состояние системы и культура команды проекта.
- Наблюдаемость дает возможность понимания и разбора инцидентов в сложных системах.
- Подбирайте инструментарий под свои нужды. Не всё описанное в маркетинговом буклете требуется для обеспечения характеристики наблюдаемости.
- Обращайте внимание на возможность масштабирования инструментов, производительность и стоимость использования.