

# Наблюдаемость IT ландшафта на базе open source решений и стандартов



**MTC**

Ты знаешь, что можешь!

## О себе

# Бочаров Филипп

Ведущий разработчик в МТС

Занимаюсь разработкой платформы Наблюдаемости. Помогаю продуктовым командам сделать работу сложных распределенных систем понятной и прозрачной.



# Появление проблемы



Перевод крупной CRM + Billing системы с монолита на SOA архитектуру.

Более 100 сервисов, более 1000 типов процессов, 50 млн. экземпляров процессов в день

Сложно проводить диагностику процессов:

- Отладка на машине разработчика невозможна
- Сотни разрозненных лог-файлов
- Долго искать логи конкретного процесса

# Где мои логи?

Разные  
форматы

Десятки  
серверов

200+ продуктовых  
команд

Тех. поддержка

В каких  
сервисах искать?

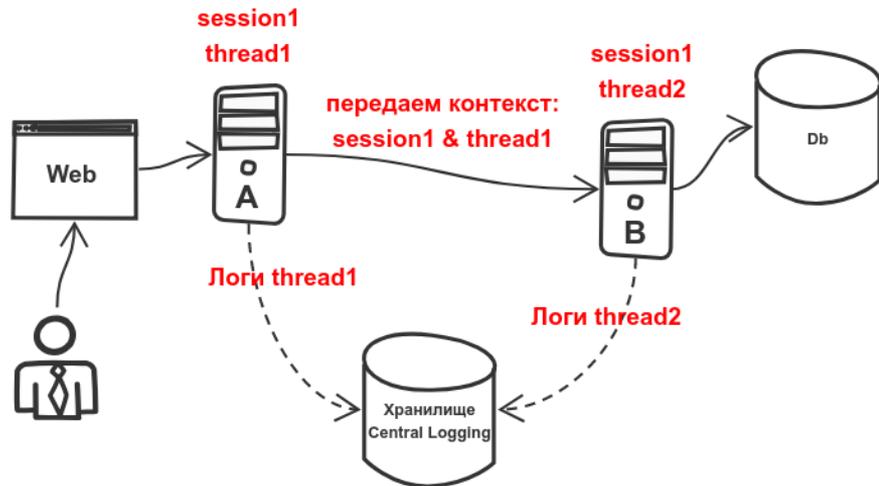
Логи не привязаны  
к процессу



Ты знаешь, что можешь!

# in-house решение

Разработали Central Logging - систему трассировки и централизованного логирования.



# Результаты

- Снижение трудозатрат на 1 баг на 30%
- Закончился “пинг-понг” багов между командами
- Де-факто стандарт для разработчиков, тестирования и тех. поддержки



# НОВЫЕ ВЫЗОВЫ

- Множество новых цифровых продуктов: телемедицина, Smart Farming итд... (200+ продуктовых команд)
- Гетерогенность ландшафта – продукты используют разные языки, платформы и фреймворки
- Продукты образуют экосистемы и взаимодействуют между собой

Нужно обеспечить end-to-end наблюдаемость

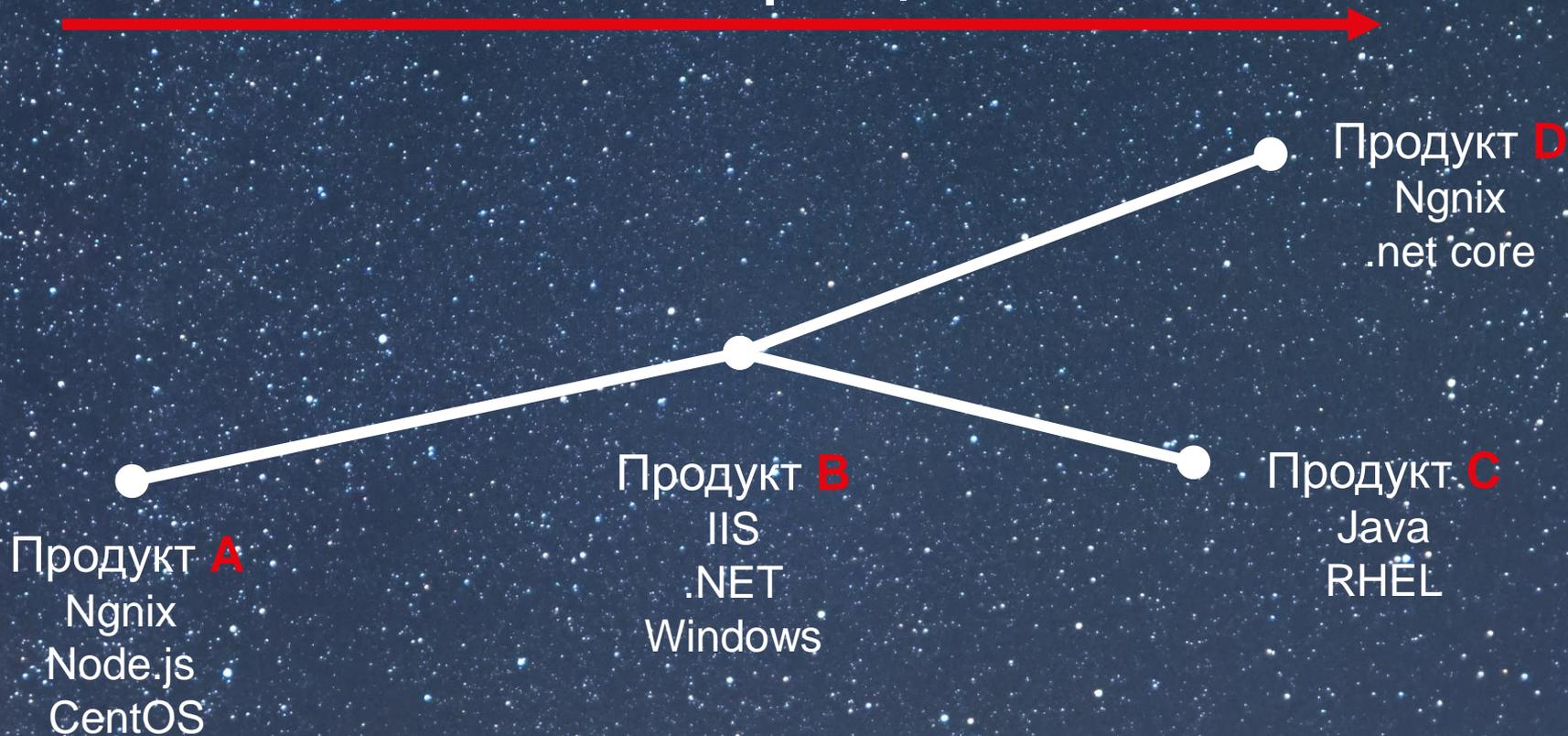
IT ландшафта



**МТС**

Ты знаешь, что можешь!

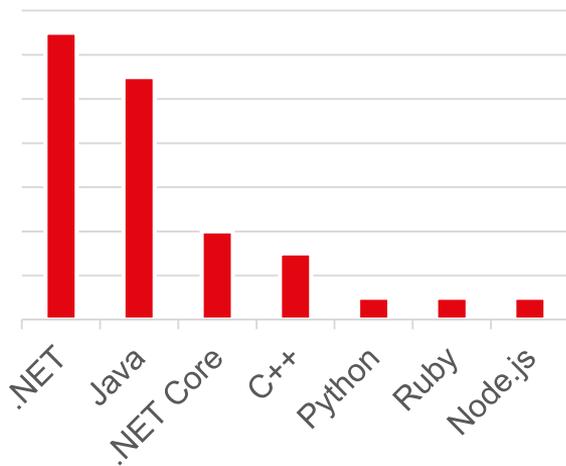
# Бизнес процесс



**MTC**

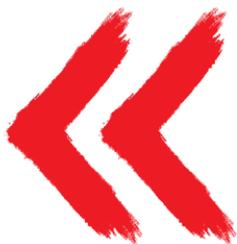
Ты знаешь, что можешь!

# Пробуем применить Central Logging



Провели исследование IT ландшафта:

Наши продукты разработаны на разных языках и платформах. Используют разное связующее ПО и фреймворки.

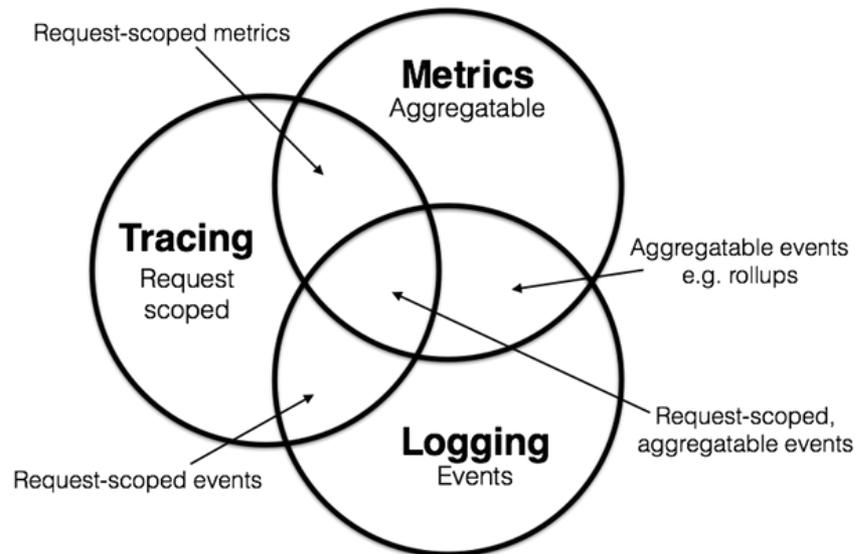


Силами одной команды не получится охватить все технологические стеки на нашем IT ландшафте



# Наблюдаемость

Это свойство системы, показывающее, можно ли по выходным данным определить внутреннее состояние системы и процессов в ней.



# Диагностика процессов



## Домен наблюдаемости

Метрики продукта

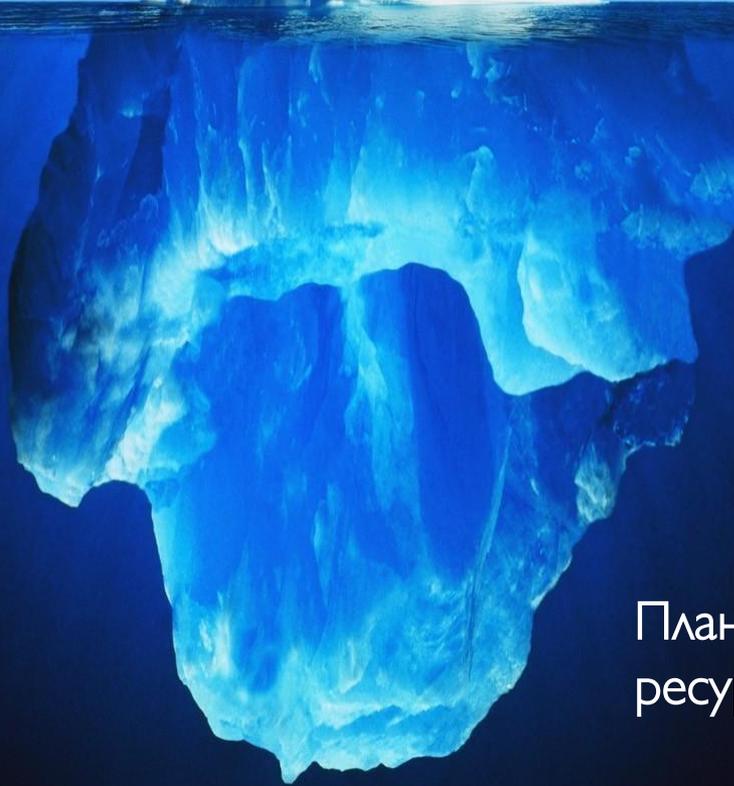
Карта IT ландшафта

SLA процессов

Оценка влияния инцидента

Предотвращение инцидента

Планирование ресурсов



Ты знаешь, что можешь!

# На рынке уже есть решения

## Инвазивные

Observability as a code

Разработчик ответственен за наблюдаемость и размечает свой код с помощью API

Примеры: Open Zipkin, Jaeger



## Неинвазивные

Коробочные решения

На машину устанавливается агент, автоматически инструментирующий сервисы

Примеры: AppDynamics, New Relic

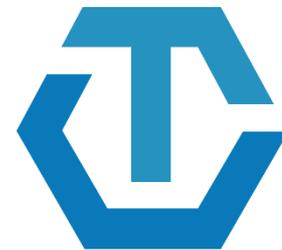
# The OpenTracing project

Это открытый стандарт и независимый от поставщика API для распределенной трассировки.

Open Source, Apache 2.0

C#, Java, C++, PHP, Python, Ruby, Objective-C, Go, JavaScript, ...

Поддерживаемые трассировщики: Jaeger, LightStep, Instana, DataDog



OPENTRACING

**Единый API и модель данных для всех платформ и продуктов**



**MTC**

Ты знаешь, что можешь!

# Выбор системы трассировки

Смотрели: Jaeger, Open Zipkin, Appdash, LightStep, Instana, DataDog

## Jaeger

- Совместимость с Open Zipkin
- Sampling: Probabilistic, Rate, Remote
- Лучшая поддержка OpenTracing
- Интеграционное тестирование клиентов разных платформ



## Open Zipkin

- Долше на рынке
- Больше звезд на Github

### Вывод:

1. Нет явного победителя – обе системы хороши!
2. Мы выбрали Jaeger из-за совместимости с Open Zipkin и продвинутого сэмплирования.



**МТС**

Ты знаешь, что можешь!

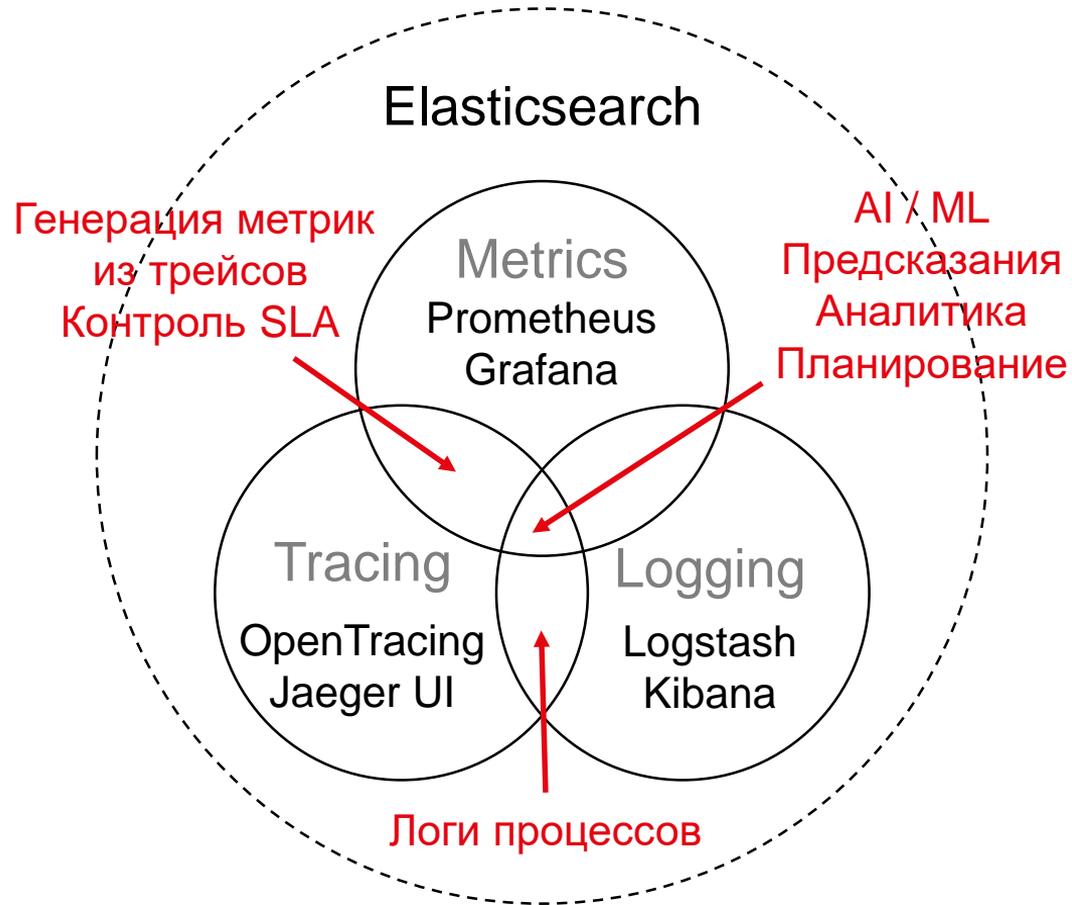
# Собираем все вместе

Elasticsearch – как единое хранилище данных

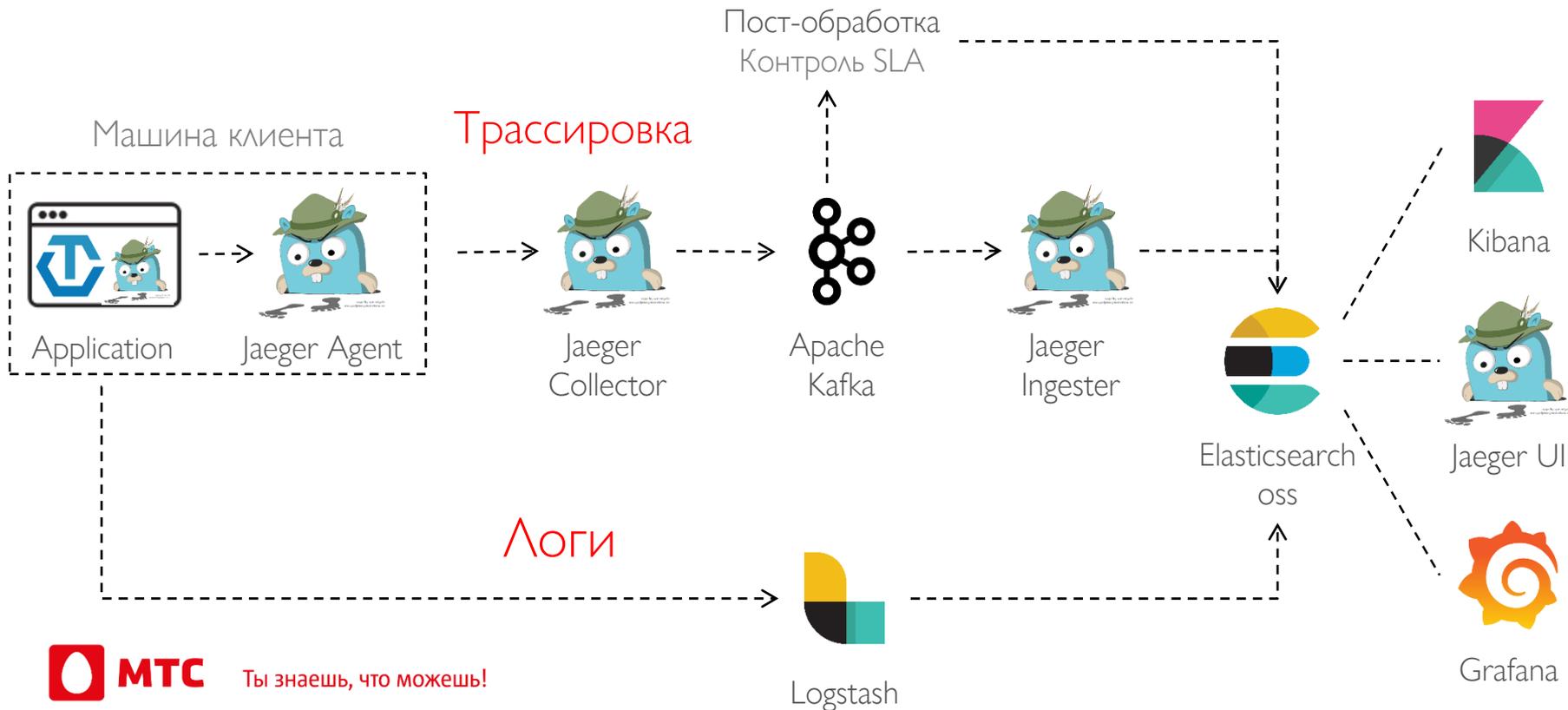
Для пользователя:  
Jaeger UI + Kibana + Grafana

Связь всех типов данных открывает новые возможности:

- Предсказание аварий
- Планирование ресурсов
- Оценка влияния инцидента
- ...



# Архитектура платформы Наблюдаемости



# Почему этого недостаточно?



Хочется получить единое рабочее место, которое может “всё”



У нас десятки мегабайт логов на один процесс. В Jaeger UI неудобно просматривать такое количество логов. Нет подсветки XML и JSON.



У нас в логах могут быть персональные данные клиента. Мы не хотим, чтобы другие команды их видели...



Open Source инструменты требуют доработки и кастомизации с учетом требований стейкхолдеров



**МТС**

Ты знаешь, что можешь!

# Как мы можем помочь сообществу?

## Открыть свои `opentracing-contrib*` пакеты

При подключении команды исследуем ее стек и разрабатываем для нее пакет, если его еще нет. Пакеты публикуем на GitHub и регистрируем в <https://opentracing.io/registry>

Уже готовы: пакеты инструментирования WCF и ASP.NET (.net)

## Делиться кейсами использования

У МТС есть свои кейсы, отличные от Red Hat и Uber. Обратная связь от нас и обсуждение этих кейсов позволит open source компонентам стать взрослее.



**МТС**

Ты знаешь, что можешь!

# Преимущество – снижение трудозатрат

## за счет повышения качества кода

- Code review от мейнтейнеров
- Больше пользователей – быстрее отловим баги

## за счет общего решения задачи

- Если кейс нужен всем - вместо частного решения, дорабатываем open source систему

## за счет поддержки кода сообществом

- Затраты на исправление багов и внедрение новых фич распределяются на все сообщество



# Выводы

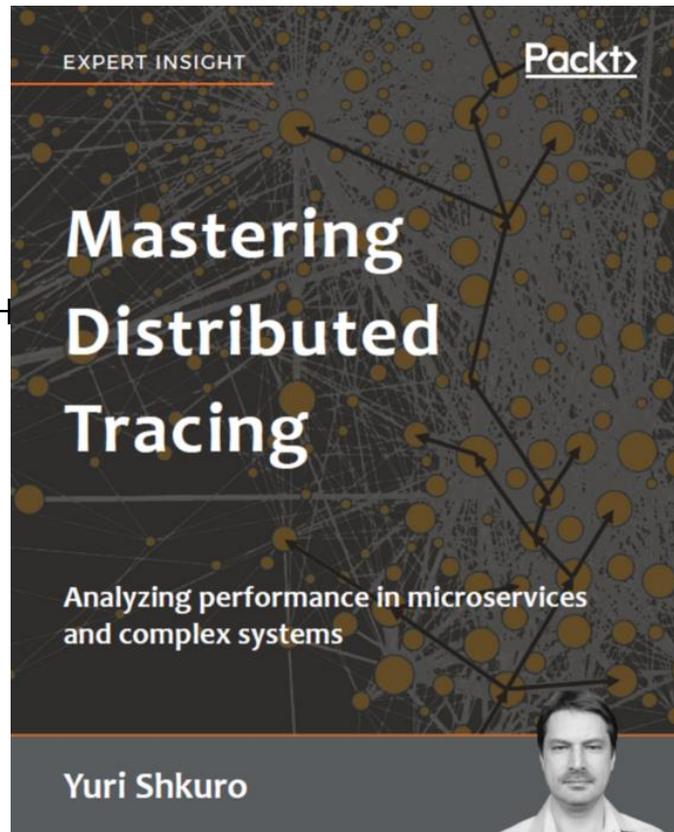
1. Наблюдаемость IT ландшафта дает новые, удивительные возможности всем участникам
2. Уже сейчас из Open Source компонентов можно собрать решение, адаптированное для вашего IT ландшафта
3. Сообщество идет по пути создания стандартов в области наблюдаемости. Прежде чем писать свой “велосипед” - изучите мировой опыт

## Полезные ссылки

<https://openapm.io/> - собери свою платформу Наблюдаемости

<https://opentracing.io/registry> - инструментирован ли твой фреймворк?

<https://www.cncf.io/> - The Cloud Native Computing Foundation.  
Инкубатор oss решений для микросервисной архитектуры.



Спасибо!