Безопасная разработка при использовании open source компонентов

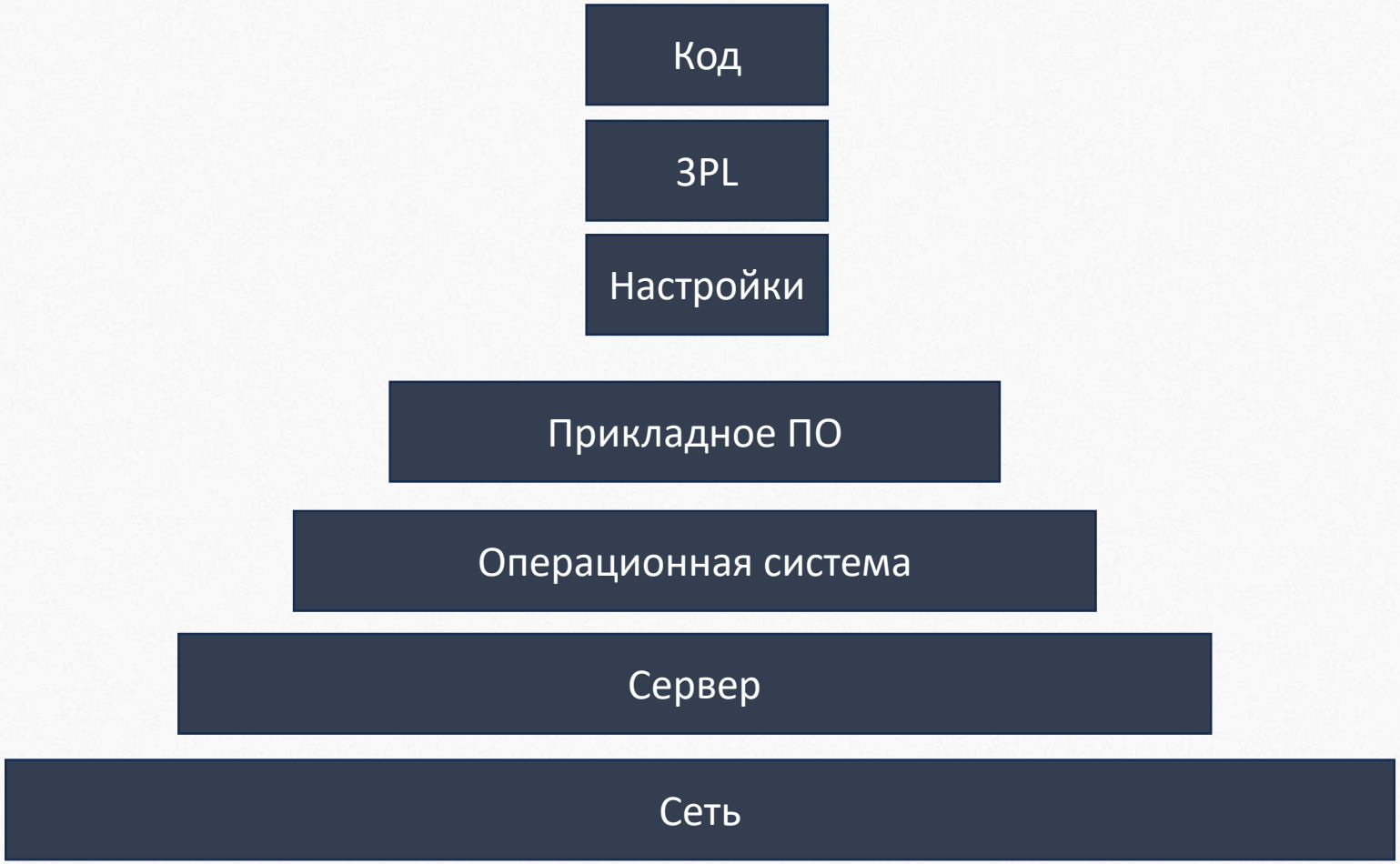
Крючков Константин
Swordfish Security
@twoks

bio

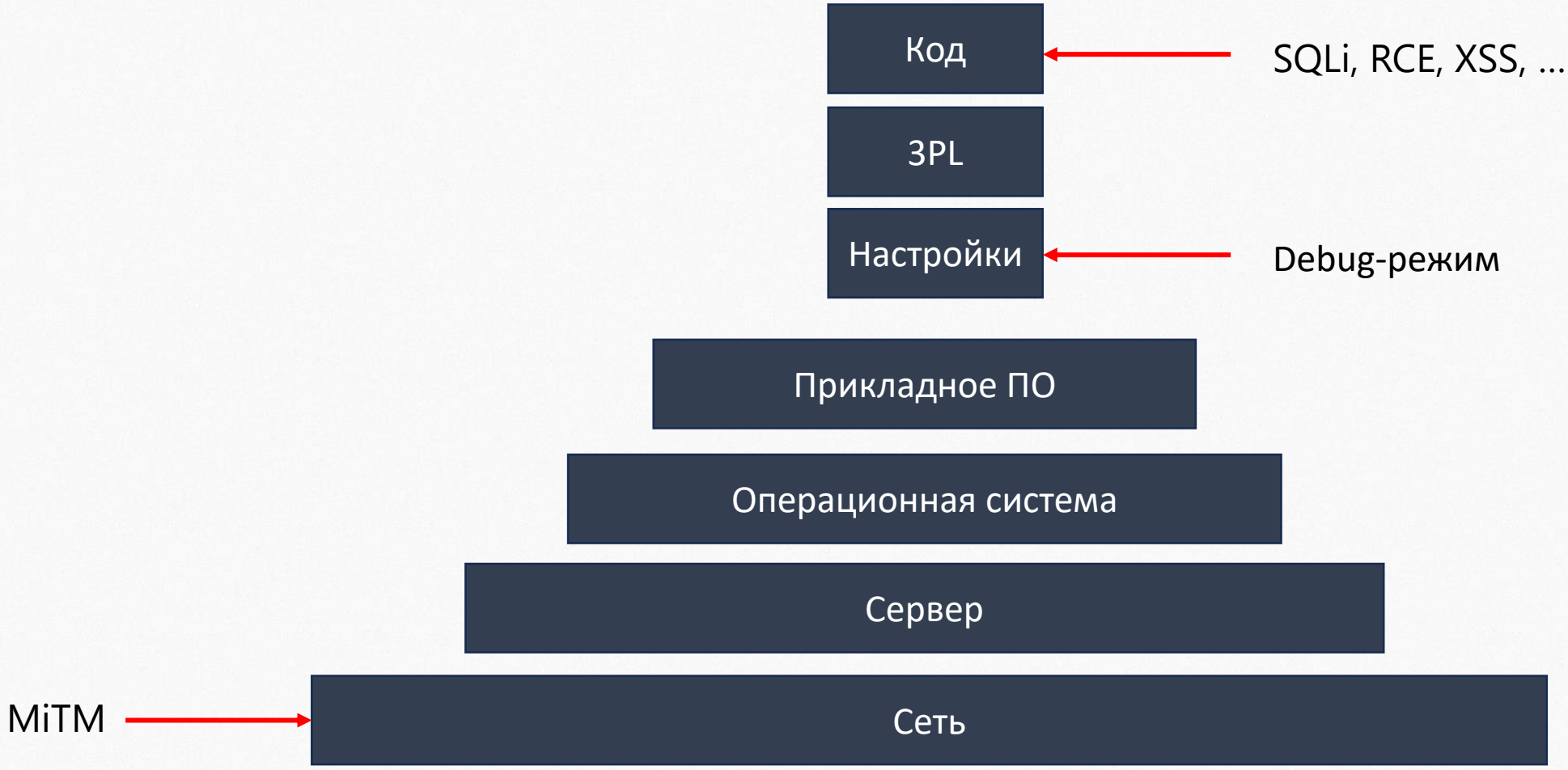
- Аудит
- Bug Bounty
- Триаж SAST/DAST/SCA
- DevSecOps
- Владелец продукта

APPSEC RACK

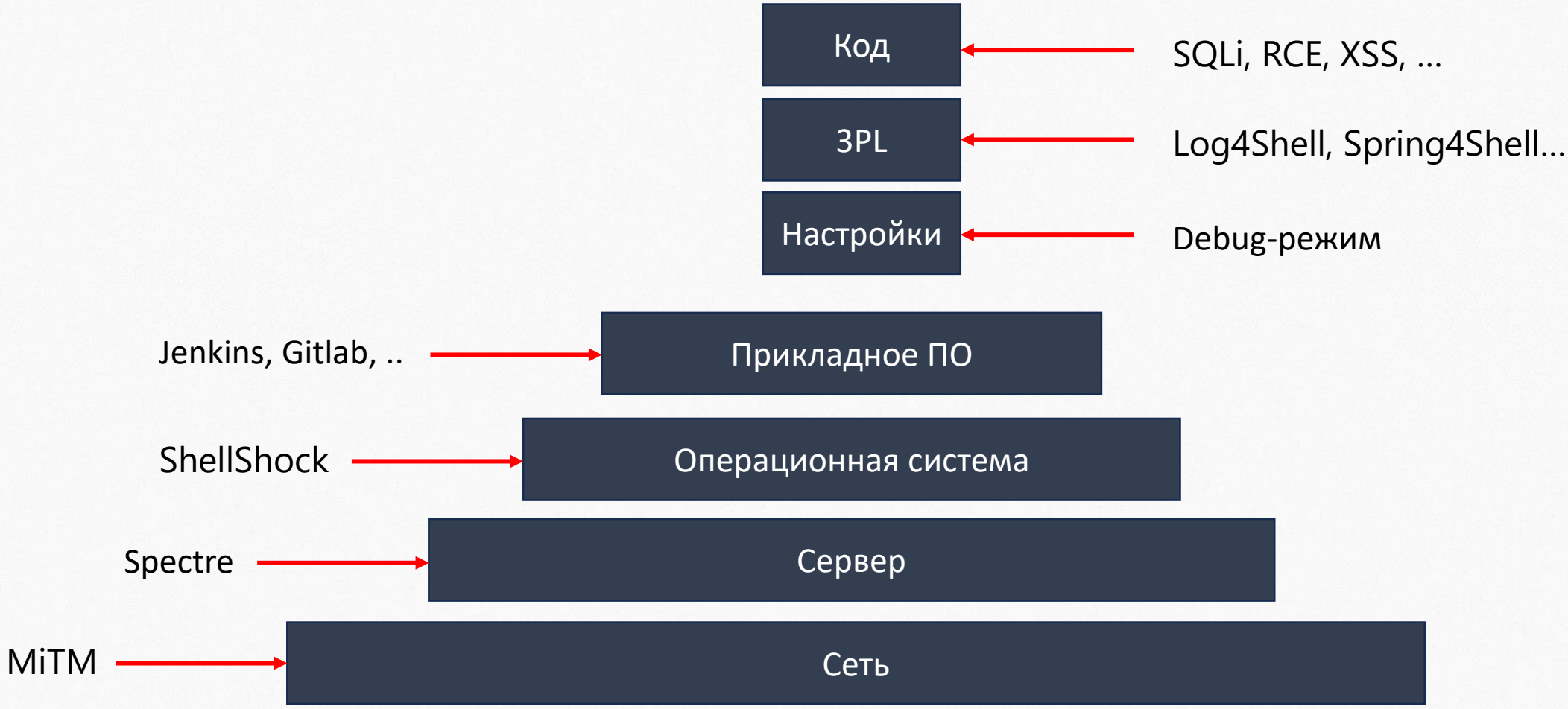
Архитектура приложений



Архитектура приложений



Архитектура приложений



Open Source

1. Возможность переиспользования

   Любой может проверить, модифицировать, улучшить и даже продать.

2. Уникальный феномен

   Цифровое волонтерство.
   Поддержка, уважение, признание, CV

3. Существующее сообщество и движение

   Очень большое

Open Source

Для компаний

Эффективность

Time-To-Market

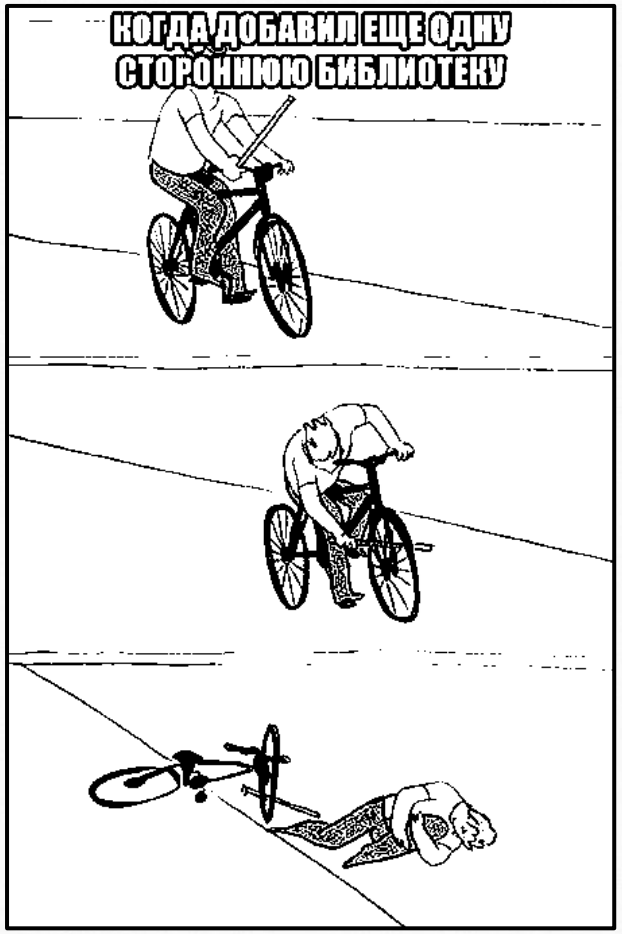
Open Source

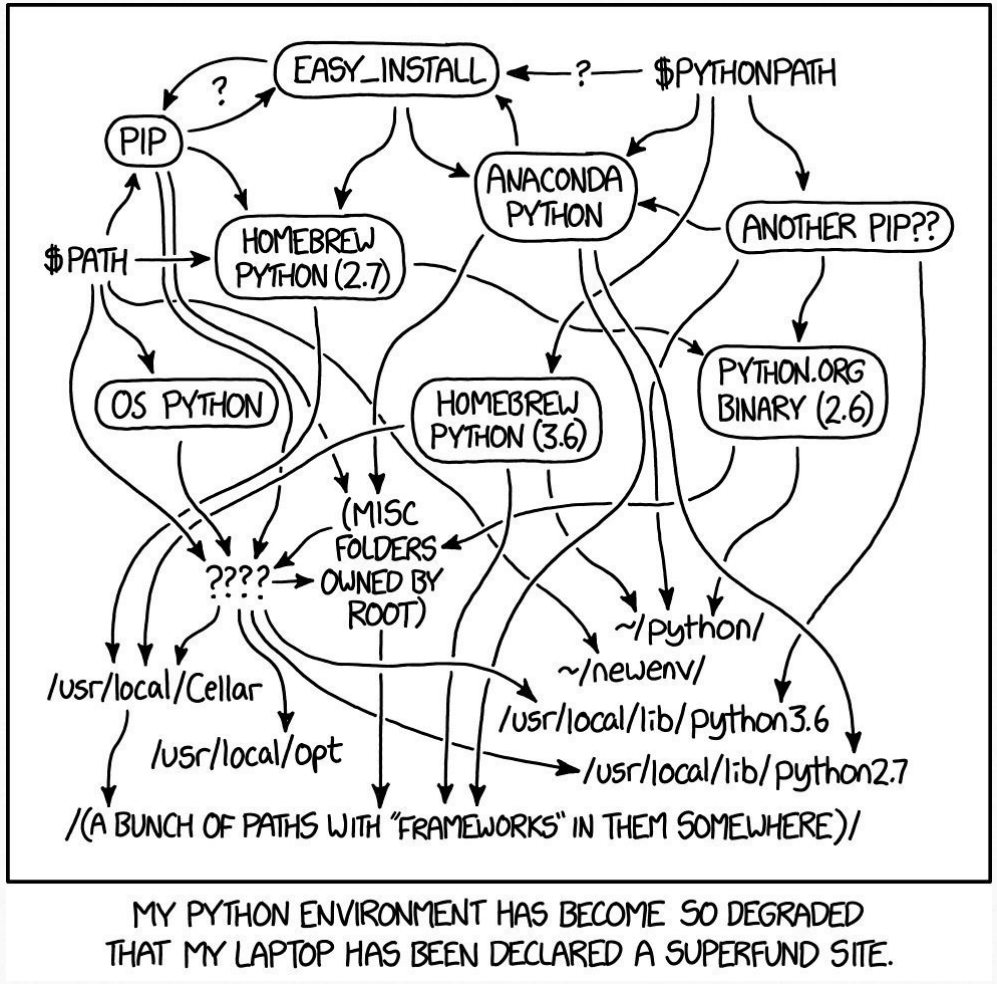
В наши дни разработка ПО невозможна без использования Open Source.

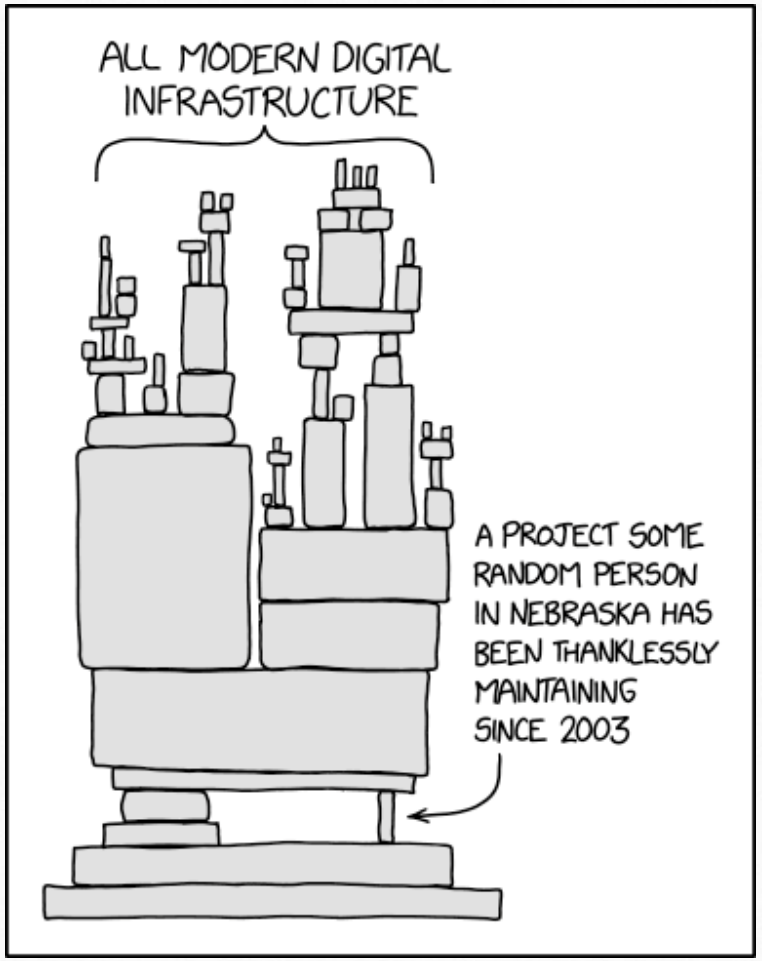
You can’t develop software anymore these days without doing open source.

Wolfgang Gehring, FOSS Ambassador
// Mercedes-Benz Tech Innovation

Open Source: опасность







Open Source: опасность

- Equifax (147M), SolarWinds (Supply Chain)
- event-stream (проблема с транзитивом)
- Dependency Confusion
- Log4Shell (Java Top 3)

- Maven Gate (2024)

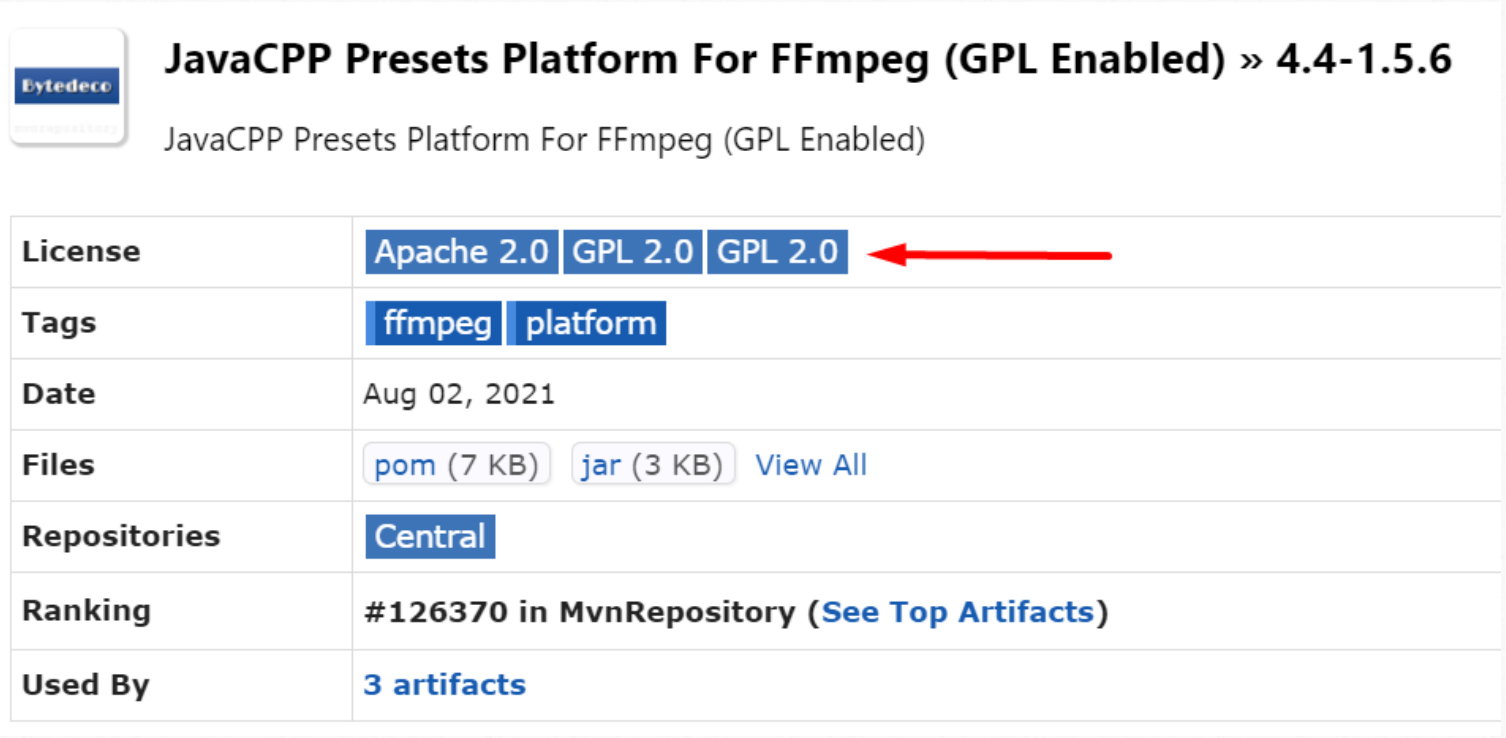
Open Source: безопасность

- Дизайн
- Загрузка
- Сборка
- Деплой
- Эксплуатация

0. Дизайн

- Определение бизнес-задачи
- Определение типа приложения
- Определение архитектуры
- Определение фреймворка

0. Дизайн



JavaCPP Presets Platform For FFmpeg (GPL Enabled) » 4.4-1.5.6

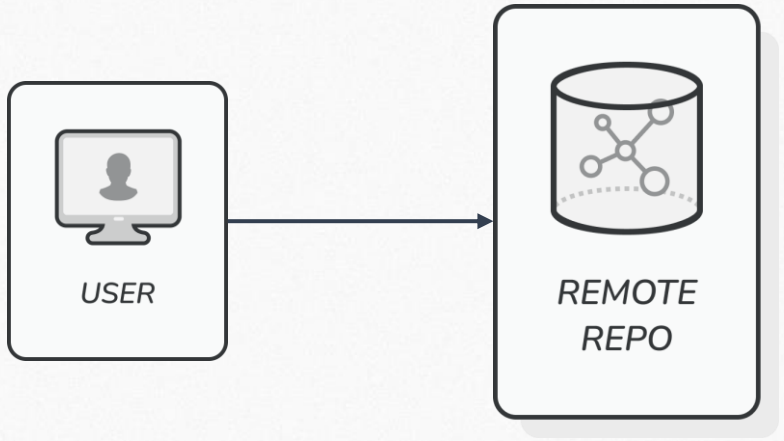
JavaCPP Presets Platform For FFmpeg (GPL Enabled)

License	Apache 2.0 GPL 2.0 GPL 2.0
Tags	ffmpeg platform
Date	Aug 02, 2021
Files	pom (7 KB) jar (3 KB) View All
Repositories	Central
Ranking	#126370 in MvnRepository (See Top Artifacts)
Used By	3 artifacts

- SPDX

1. Загрузка пакета

- Машина разработчика
- Машина агента CI/CD



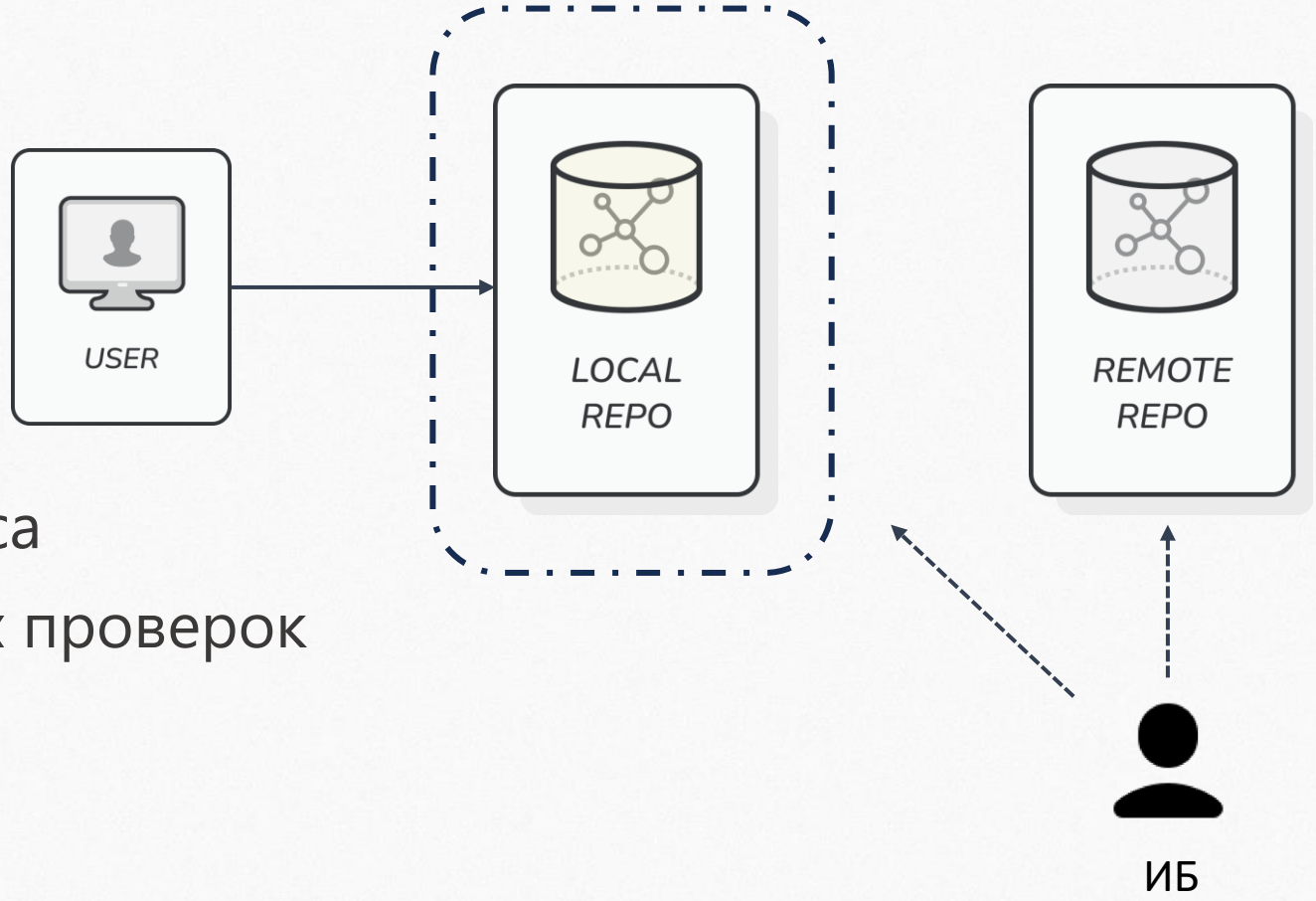
Maven Central

PYPI

NPM.js

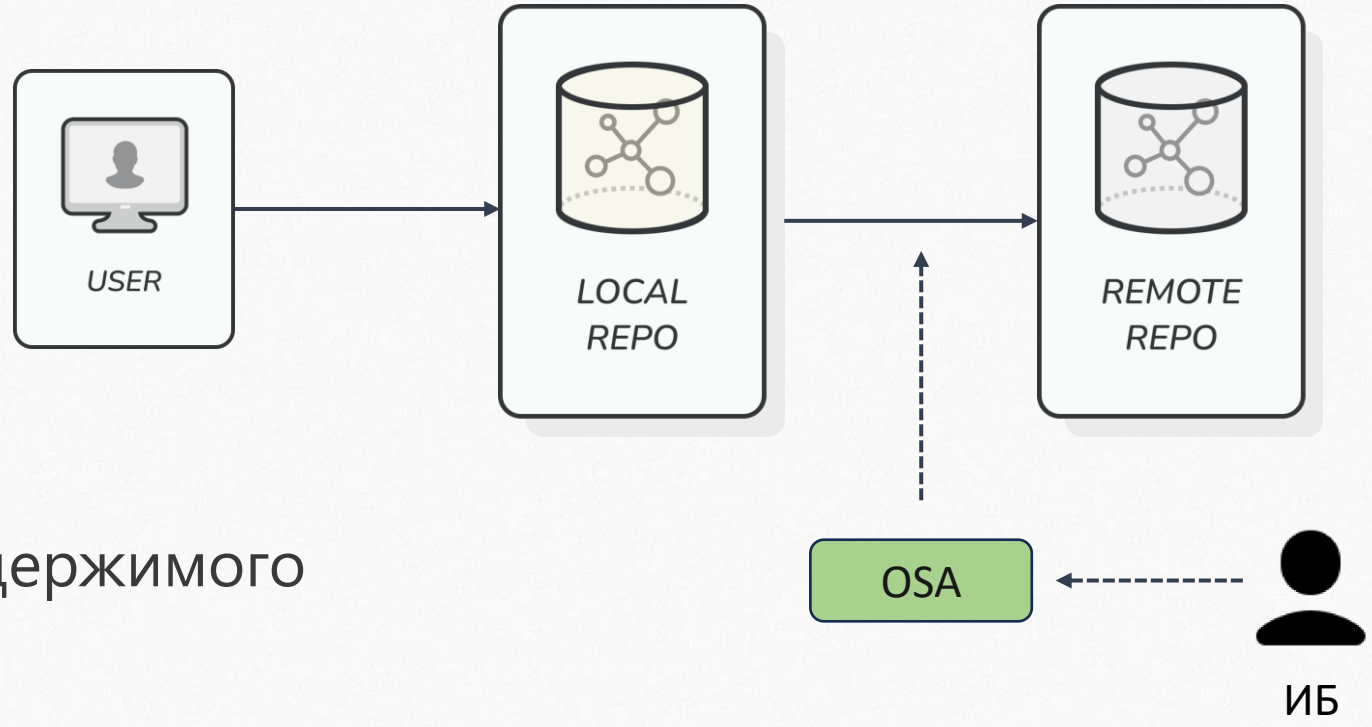
1. Загрузка пакета

- Безопасный репозиторий
- Публикация пакетов происходит в рамках организационного процесса и выполнения ряда ручных проверок



1. Загрузка пакета

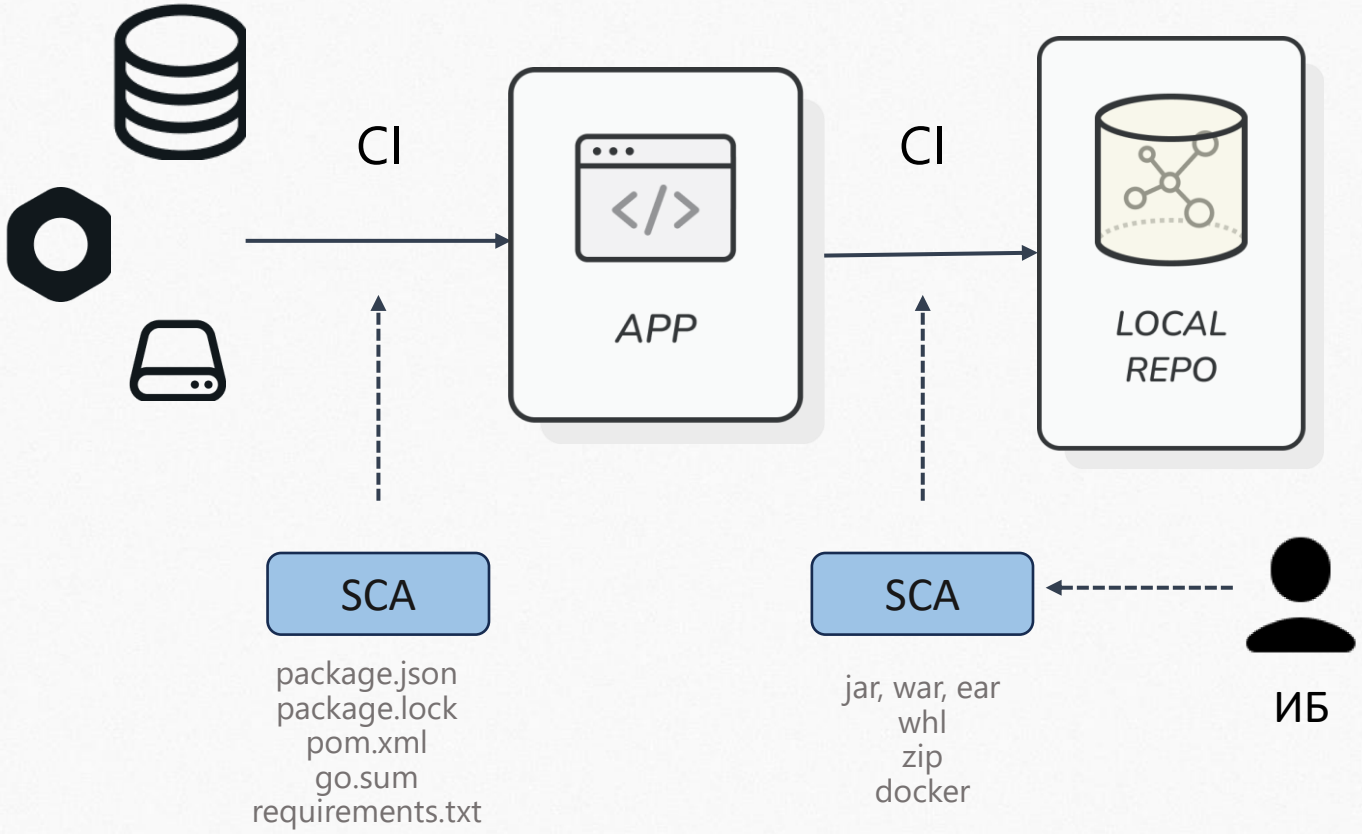
- Практика OSA (Open Source Analysis)
- Автоматическая проверка
- Наличие уязвимостей
- Наличие вредоносного содержимого
- Проблемные лицензии



2. Сборка

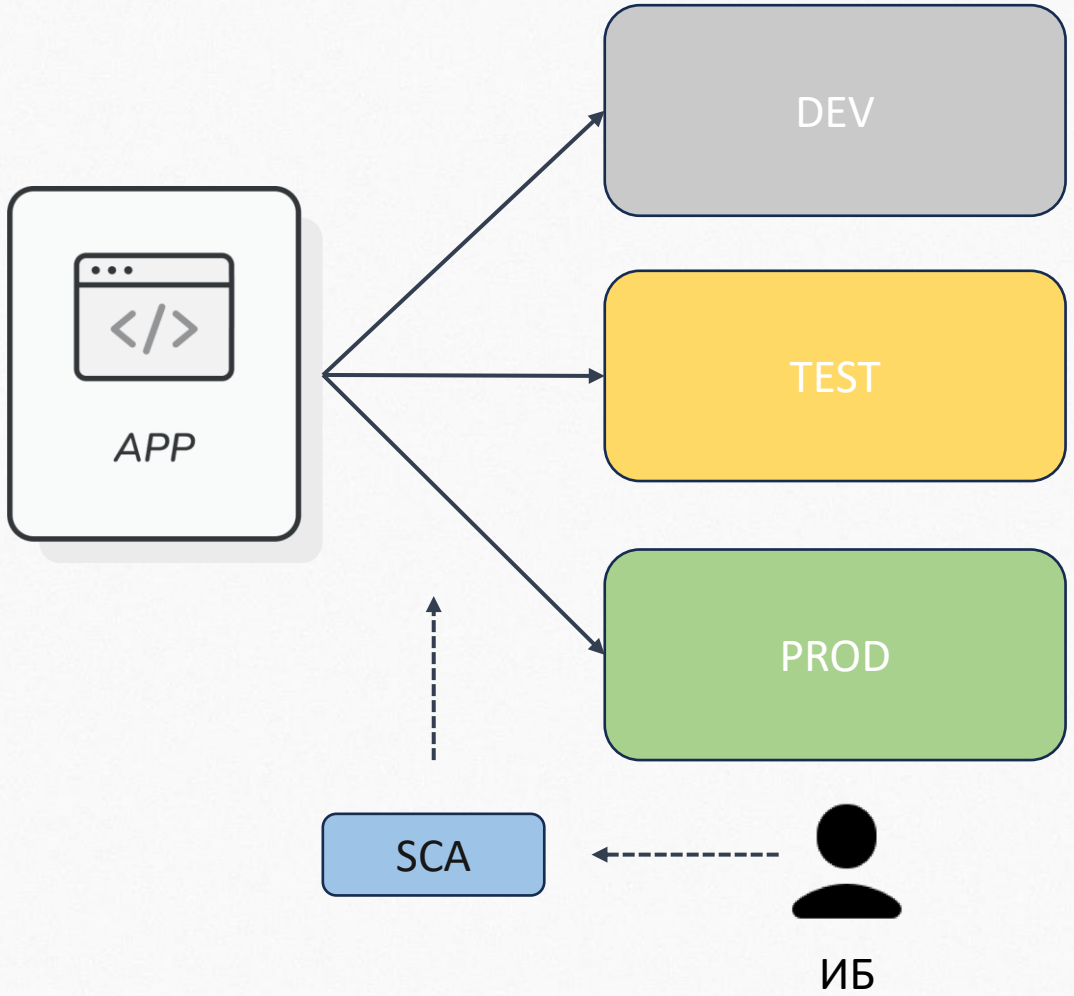
- Практика SCA (Software Composition Analysis)

- Анализ манифестов
- Анализ артефакта



3. Деплой

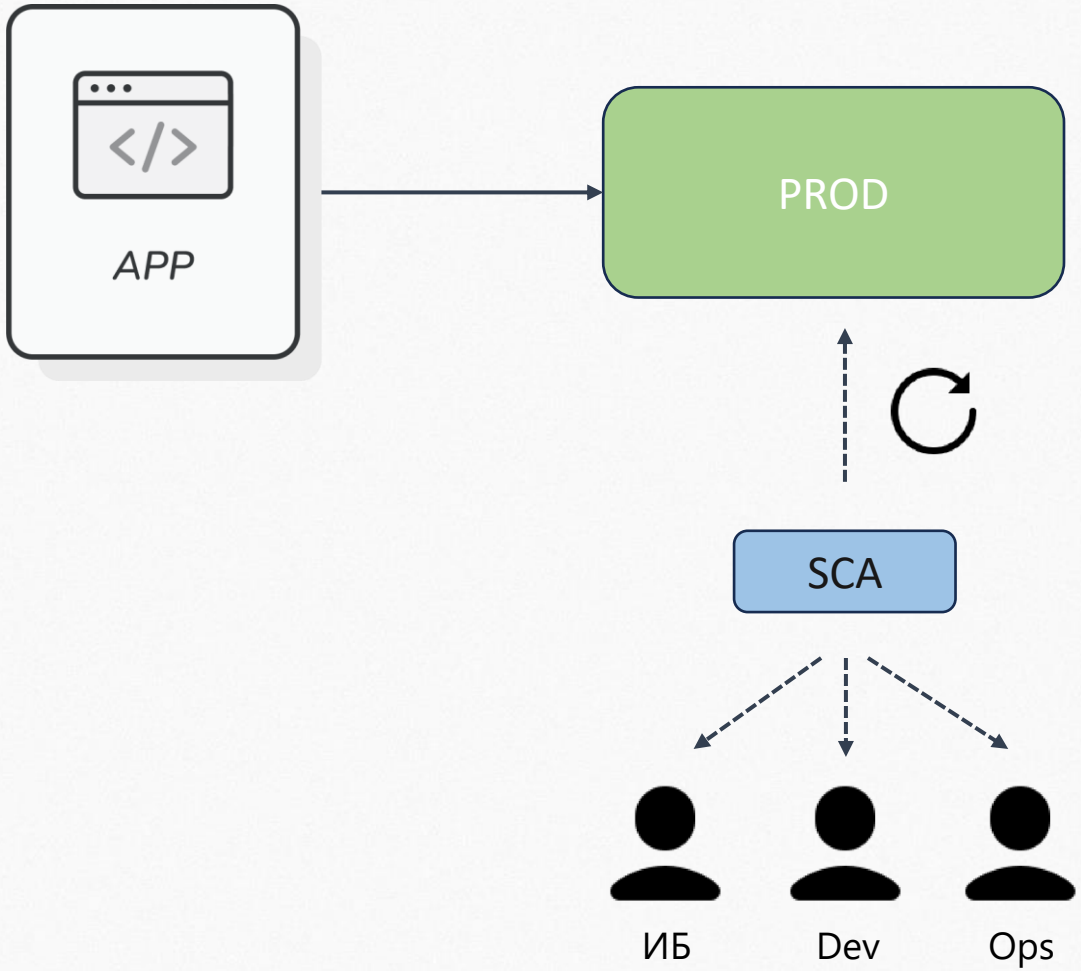
- Разные среды – разные настройки
- Проверка артефакта
  в пайплайне CD



4. Эксплуатация

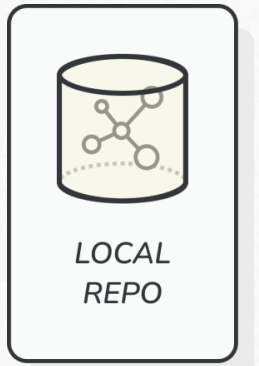
- Мониторинг 0-day

- Уведомление в Dev
- Уведомление в Ops
- Уведомление в ИБ



OSA vs SCA

Malware/Protestware



OSA

SCA

Malware/Protestware
Vulnerability
Monitoring

Как строить?

Open Source SCA

Dependency Check

Dependency Track

Trivy

npm audit

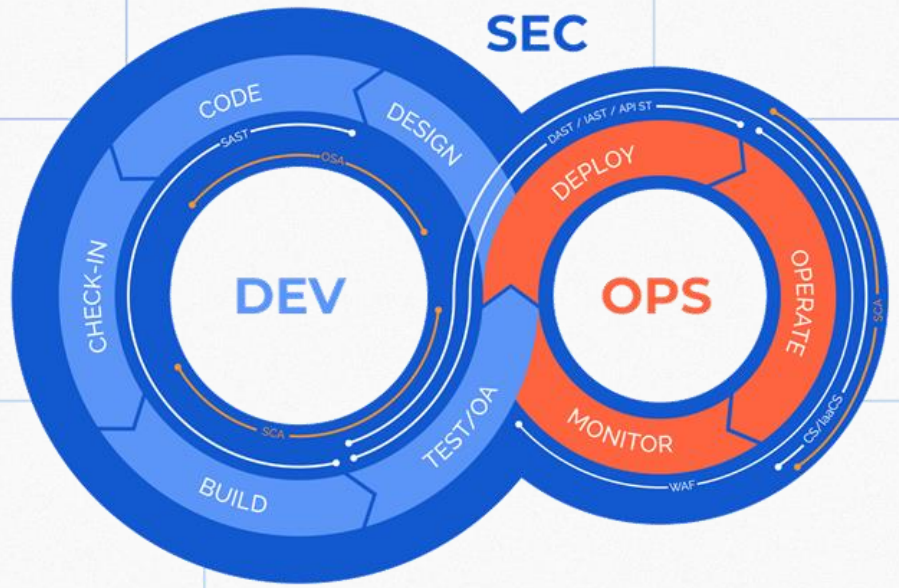
Commercial

В кулуарах

Спасибо!

Вопросы

DevSecOps



OSA

Анализ библиотек с открытым исходным кодом

Open-Source Analysis

DAST

Динамический анализ

Dynamic Application Security Testing

SAST

Статический анализ кода

Static Application Security Testing

MAST

Анализ защищенности мобильных приложений

Mobile Application Security Testing

SCA

Компонентный анализ ПО

Software Composition Analysis

CS

Анализ защищенности контейнеров

Container Security

ASOC

Оркестрация и Корреляция

Application Security Orchestration & Correlation