



Software Engineering Conference Russia **2018**

October 12-13
Moscow

Как технология **serverless**
при создании **data science** проектов
МОЖЕТ ПОВЫСИТЬ УТИЛИЗАЦИЮ РЕСУРСОВ

Игорь Хапов
Владимир Алексеев

IBM

Содержание

- мы вообще о чем
- а что есть?
- зачем нам это надо?
- опыты: скрещиваем аналитику и FaaS
- GPU + FaaS = смотрим в будущее

КТО МЫ



Игорь Хапов

#ibm #лаборатория
#менеджер #разработчик
#kubernetes
#давайпортирую



Владимир Алексеев

#ibm #архитектор
#работасклиентом
#банкинг
#всеновое #хочувсезнать



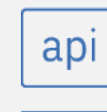
Что слышно с рынка

- У нас spark-кластер на 50 узлов. Уже на 100! Ой, а можно его оптимизировать?
- Data Science помешательство – нужна единая среда работы
- Мне бы jupyter, а еще Rstudio, да и TensorFlow не помешали бы, а кстати почему только TF..?
- Сколько-сколько за GPU? Это грабеж!
- А как этим управлять то теперь?

Хорошо, вот вам



Инструментарий



SPSS Modeler

Decision Optimization

Cognos Dashboards

Data Refinery

Model Builder

Watson API Tools

Machine Learning Runtimes



Deep Learning Runtimes



Инструментарий инфраструктуры

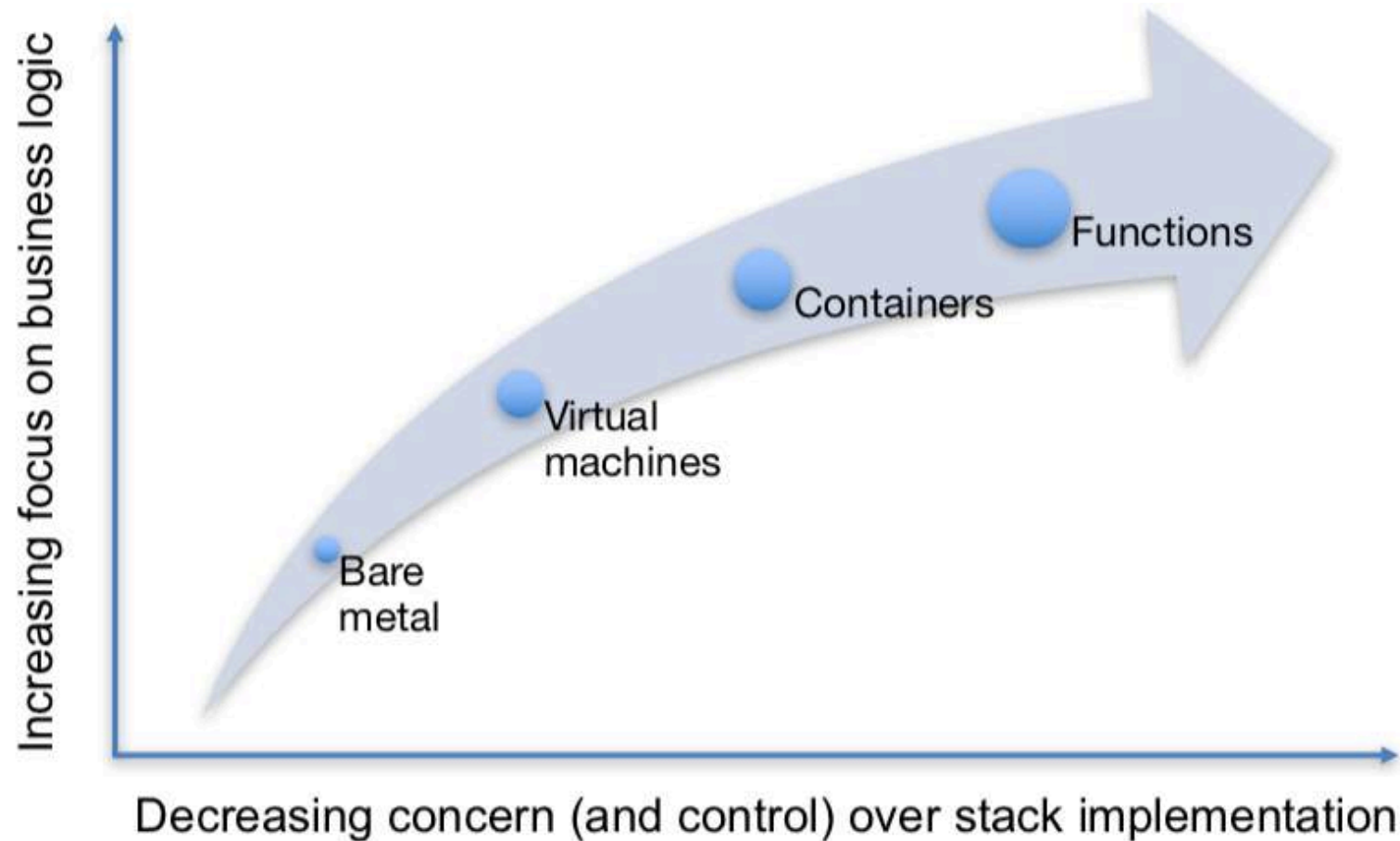


Контейнеры – ок, serverless – збс.



Serverless architectures are application designs that incorporate third-party “Backend as a Service” (BaaS) services, and/or that include custom code run in managed, ephemeral containers on a “Functions as a Service” (FaaS) platform.

<https://martinfowler.com/articles/serverless.html>



История развития



Mrp serverless



<https://github.com/anaibol/awesome-serverless>



<https://github.com/anaibol/awesome-serverless>

Hosting and code execution in the cloud

- **Amazon Lambda** - Run code without thinking about servers. Pay for only the compute time you consume.
- **Google Cloud Functions** - Lightweight, event-based, asynchronous compute solution that allows you to create small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment.
- **Azure Functions** - Listen and react to events across your stack.
- **Fission** - Fast, extensible, open source serverless functions on any Kubernetes cluster
- **IBM OpenWhisk** - Distributed compute service to execute application logic in response to events.
- **Kubeless** - is a Kubernetes-native serverless solution.
- **iron.io** - Isolates code and dependencies of individual tasks so they can be processed on demand.
- **weblab.io** - Microservices at your fingertips.
- **Peer5** - The serverless CDN. Limitless, affordable video delivery. More traffic means a stronger network.
- **StdLib** - Function as a service library and platform.
- **Auth0 Webtasks** - Run code with an HTTP call. No provisioning. No deployment.
- **Webscripts** - Scripting on the web. (Shutting down December 15, 2017)
- **APITools** - Troubleshoot, Modify, Monitor API traffic.
- **Surge** - Deploy static sites from the command line.
- **Netlify** - Generate & deploy static sites from git repositories.
- **Aerobatic** - Deploy static sites from the command line with plugins including password protection, CORS proxy, and more.
- **Effe** - a simple Open Source building block to emulate AWS Lambda.
- **Lever OS** - Serverless + Microservices = ♥
- **Now** - realtime node.js deployments
- **Brightwork** - Backend in a box for Developers
- **OpenComponents** - Serverless microservices for front-end components
- **Kloudbit** - Kloudbit helps developers build robust applications without the hassle of backend coding and server management.
- **Stackery** - Stackery enables teams of developers to design, deploy, and monitor their serverless infrastructure on cloud services providers like AWS

Frameworks

- **Apex** - Minimal AWS Lambda function manager with support for multiple languages including Nodejs, Golang, Python, Java, Rust and Clojure.
- **Up** - Deploy infinitely scalable serverless apps, apis, and sites in seconds.
- **Chalice** - Python serverless microframework from Amazon for AWS lambda
- **ClaudiaJS** - Deploy Node.js microservices to AWS easily.
- **DEEP** - Full-stack Web Framework for Cloud-Native Applications and Platforms using Microservices Architecture.
- **Gordon** - λ Gordon is a tool to create, wire and deploy AWS Lambdas using CloudFormation
- **Gestalt Framework** - Gestalt's Lambda Application SERver (LASER)™ for short, is a lambda service that supports running .Net, Javascript, Java, Scala, Ruby, and Python lambdas.
- **IronFunctions** - The Serverless Microservices platform
- **Kappa** - a command line tool that (hopefully) makes it easier to deploy, update, and test functions for AWS Lambda.
- **Lambda Forest** - Microframework that makes it easier to develop REST APIs using AWS Lambda Function and API Gateway.
- **Lambda Framework** - JAX-RS implementation for AWS Lambda.
- **lambda-registry** - A restify/expressjs like interface for aws lambda with api gateway event.
- **Lambdoku** - Heroku-like experience when using AWS Lambda
- **Python-λ** - A toolkit for developing and deploying serverless Python code in AWS Lambda
- **Serverless Framework** - Build and maintain web, mobile and IoT applications running on AWS Lambda, Azure Cloud Functions, IBM OpenWhisk, and Google Cloud Functions (formerly known as JAWS).
- **Shep** - A framework for building APIs using AWS API Gateway and Lambda
- **Sparta** - A framework that transforms a Go application into an AWS Lambda powered microservice.
- **SAM Local** - Is the AWS CLI tool for managing Serverless applications written with **AWS Serverless Application Model (SAM)**
- **Turtle** - library for building functional and actor-driven NodeJS apps on Lambda
- **Zappa** - Serverless Python WSGI with AWS Lambda + API Gateway.
- **λmbdify** - AWS Lambda automation and integration for Python
- **Squeezer Framework** - Event-driven APIs & Web apps on microservices, serverless.
- **Spring Cloud Function** - Java framework for doing Functions using Spring ecosystem
- **Fission Workflows** - Fast workflow-based function composition for serverless functions



Преимущества и недостатки использования serverless для data science проектов

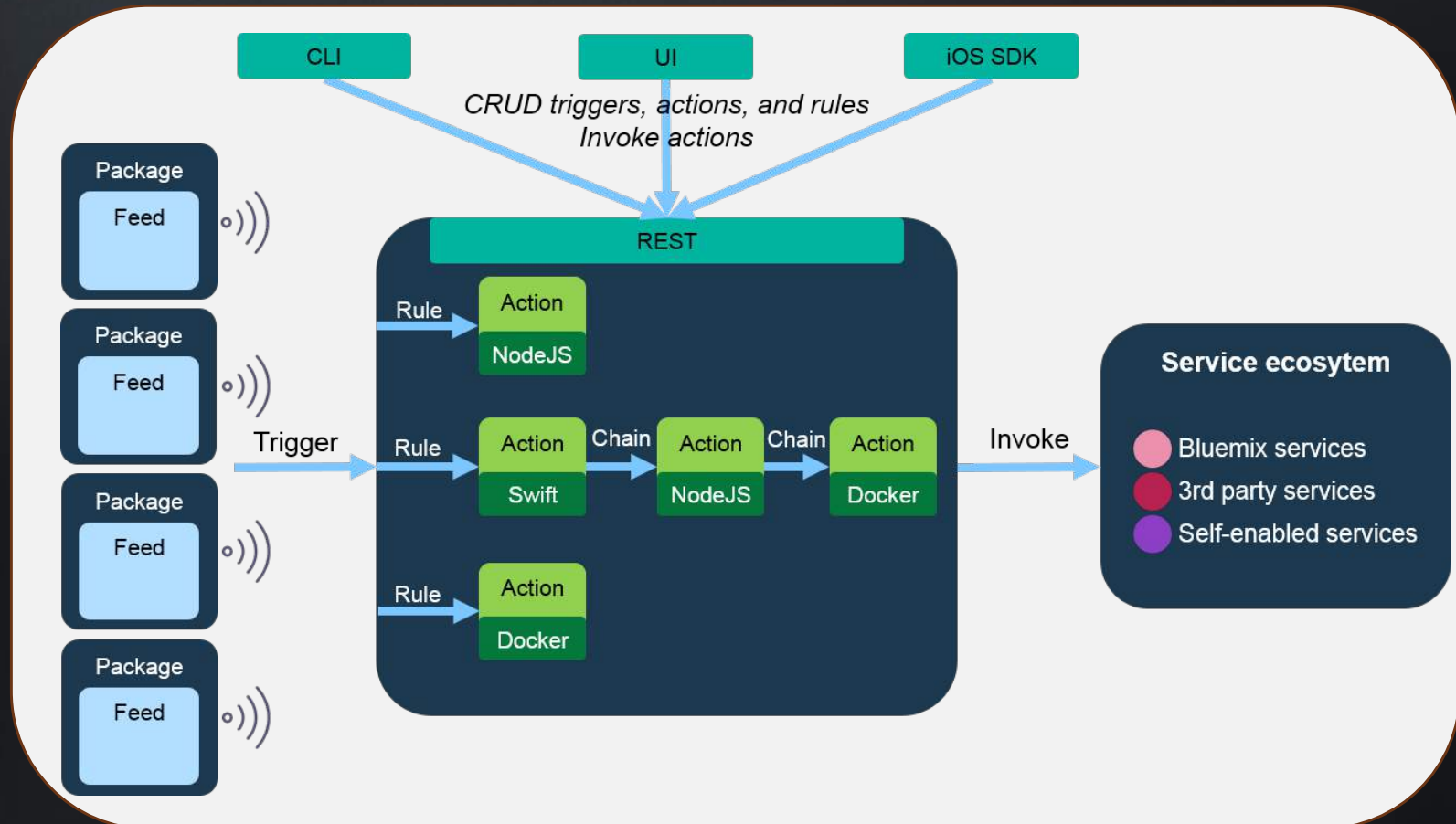
- +
 - Возможность использования одних и тех же экзекьюторов для разных приложений и пользователей
 - Возможность сохранения промежуточного результата удаленно
- - Использование встроенных scheduler-ов увеличивает latency запуска executors
 - Использование встроенных scheduler-ов для запуска заданий
 - Экзекьюторы взаимодействуют с scheduler для записи данных удаленно

Быстренько про OpenWhisk



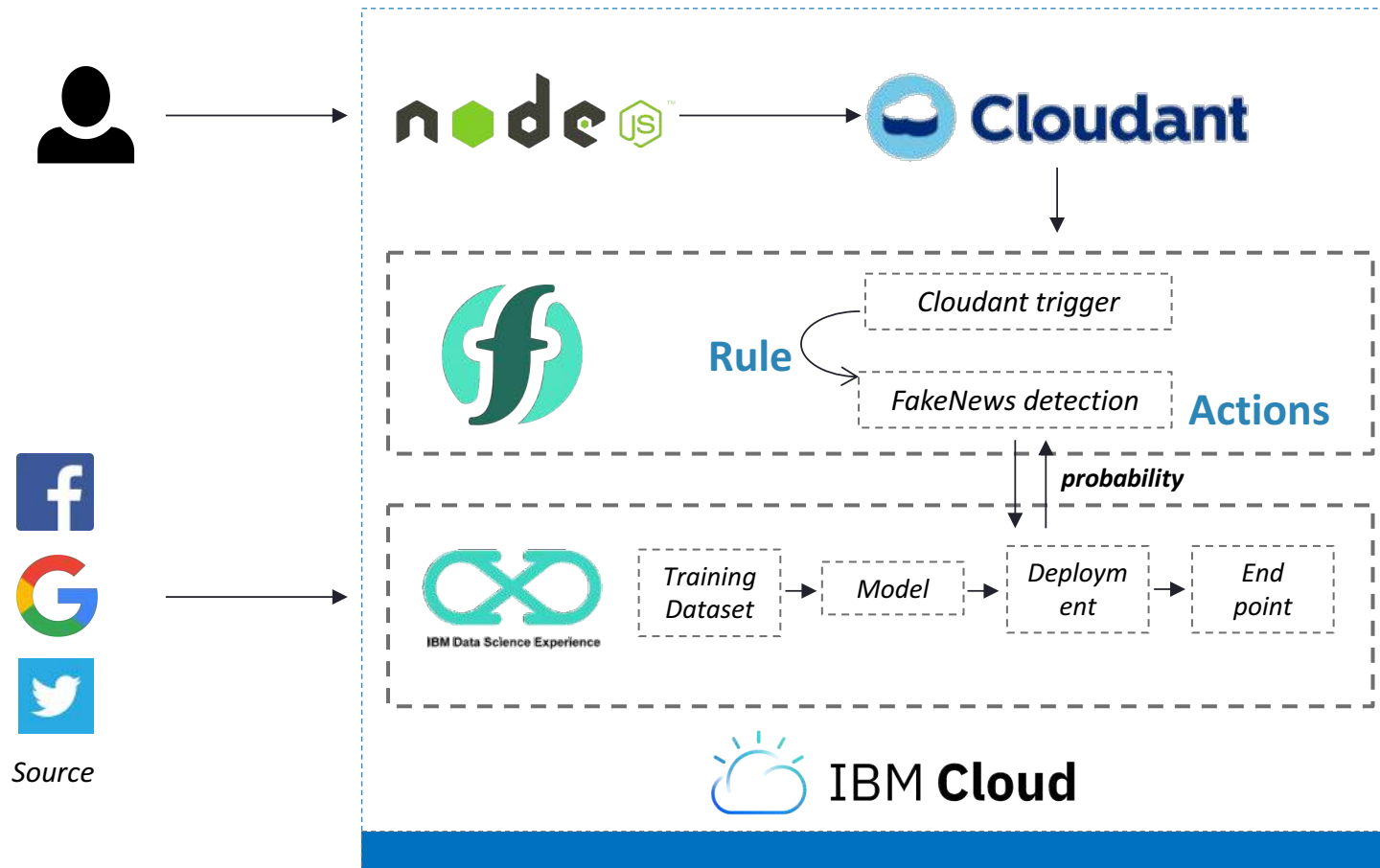
Причины выбора:

- возможность работы и в облаке, и on-premise
- наличие интеграции с другими сервисами в рамках единой платформы (IBM Cloud)
- Мы из IBM ;)

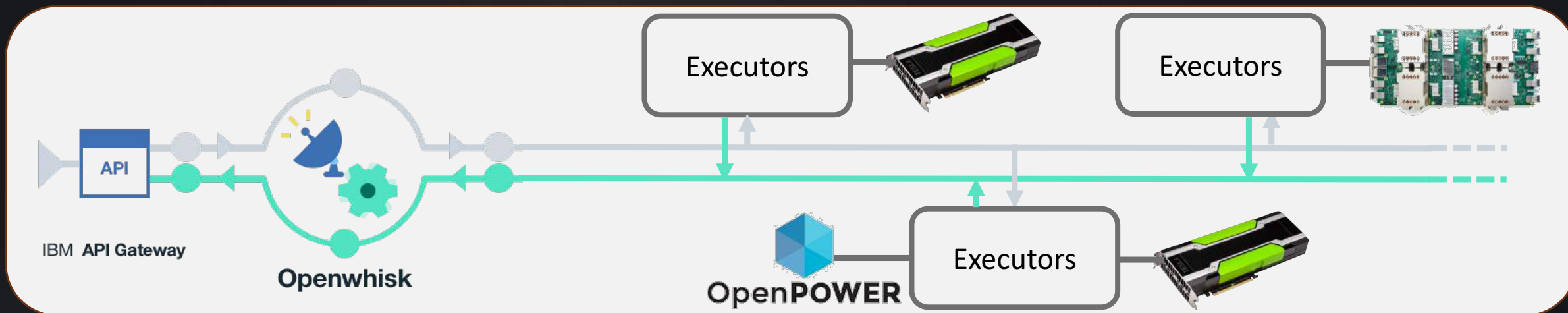
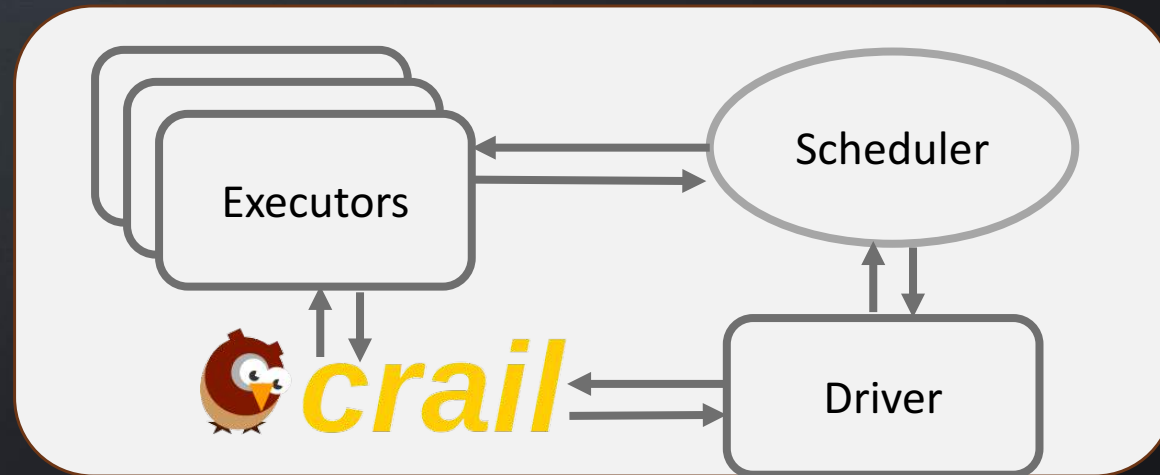
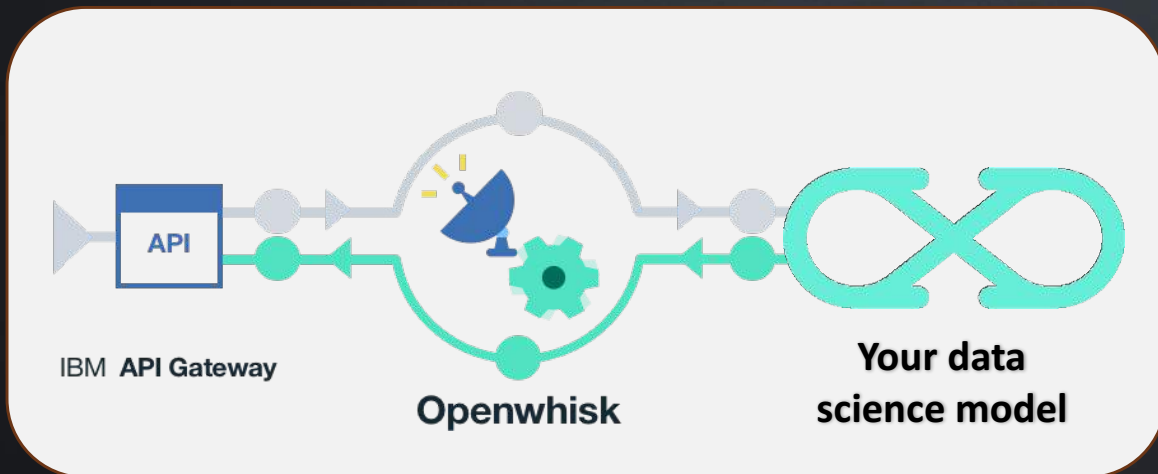


«Проба пера»

Задача – определение фейковых новостей для медиа-портала



Возможные варианты совмещения serverless и data science



Контакты



- Игорь Хапов
- <https://t.me/eecontainer>



- Владимир Алексеев
- <https://t.me/disruptiveit>

