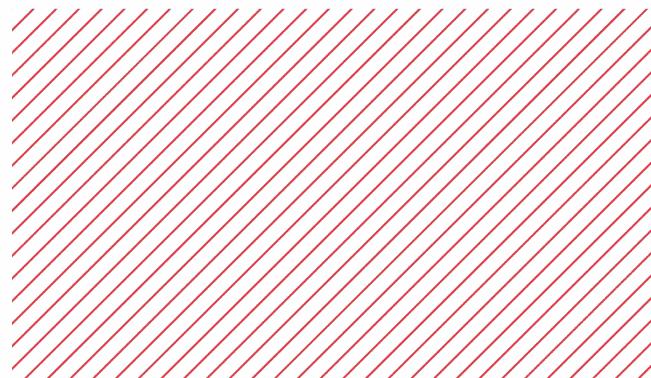


академия  
больших  
данных



# Машинное обучение в продакшене

Михаил Марюфич, MLE



**Что делаю?**

**Разрабатываю** ML решения полного цикла

**Выкатываю** их в продакшен

**Улучшаю** инфраструктуру для ML

**Что сделал?**

Различные реко-пайплайны

Поиск дубликатов в Юле

Распознавание текста в ОК

Распознавание документов для Ситимобил

**Автоматизация обучения и выкатки моделей**

**Образование:**

**МатМех СПбГУ**

**Computer Science Center** по направлению Data Science и

Software Engineering

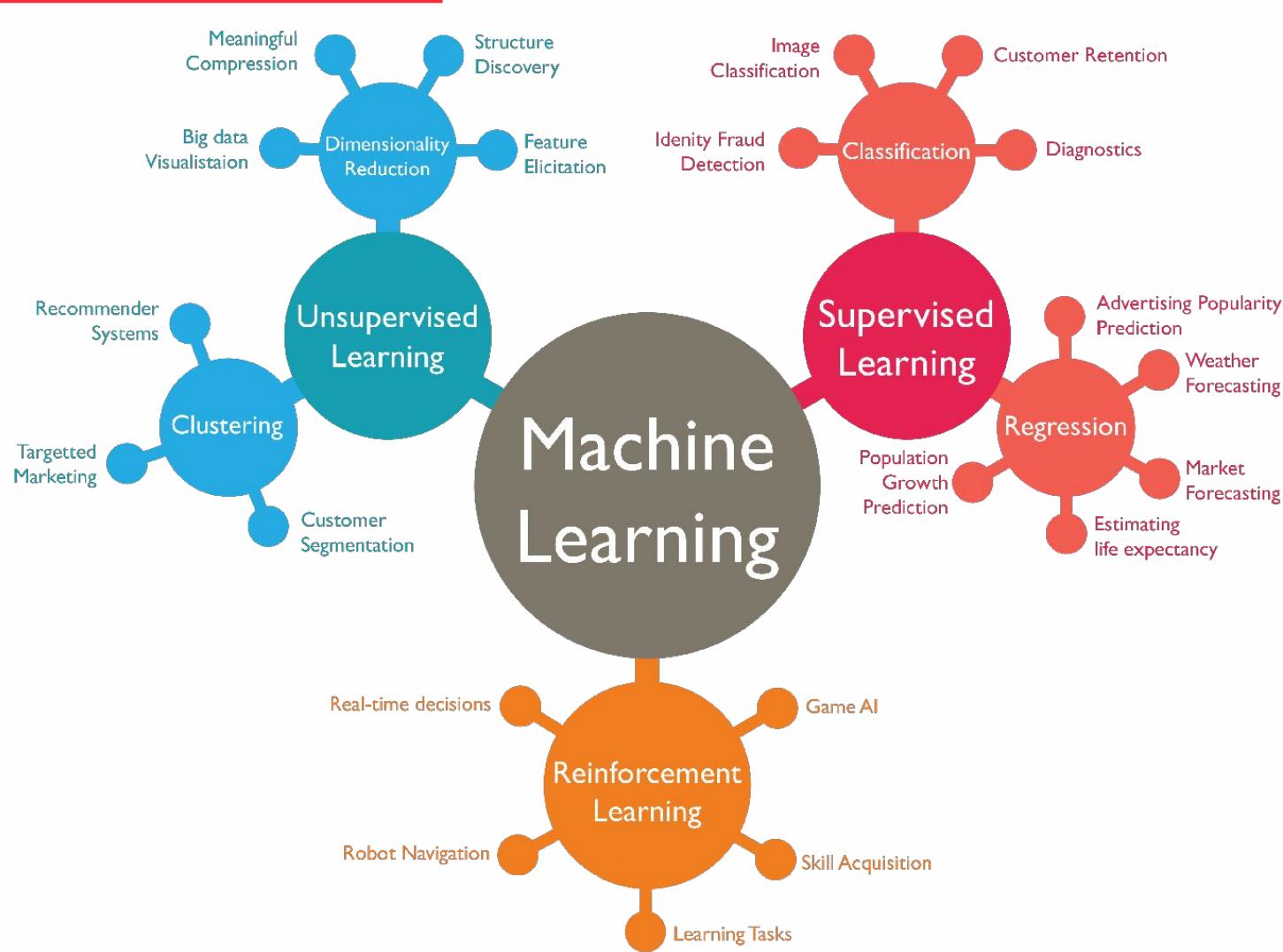
<https://compscicenter.ru/students/872/>



**Михаил Марюфич**  
**ML Engineer**

# Что такое ML?

# ML - это про науку



# Для чего компании внедряют ML?



# Модель должна быть в проде!



# Infra - ЭТО ВАЖНО

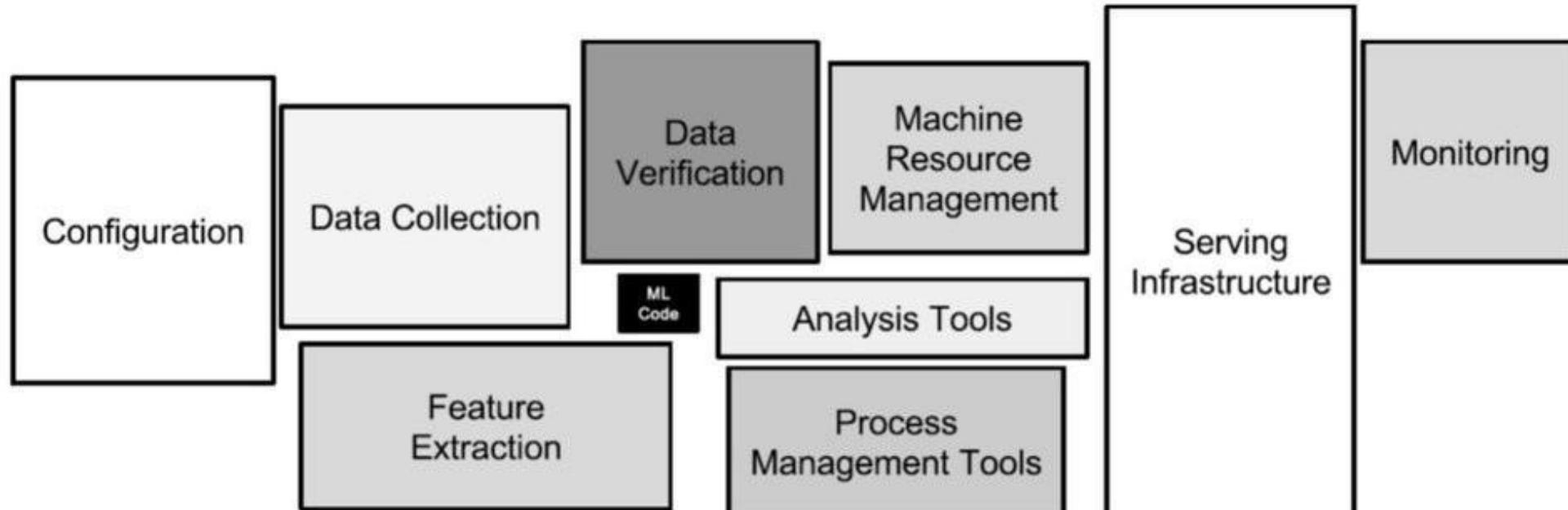


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

VALOHAI NEWSLETTER  
JANUARY 2021

HAPPY NEW YEAR 😊

2021 – The Year of MLOps

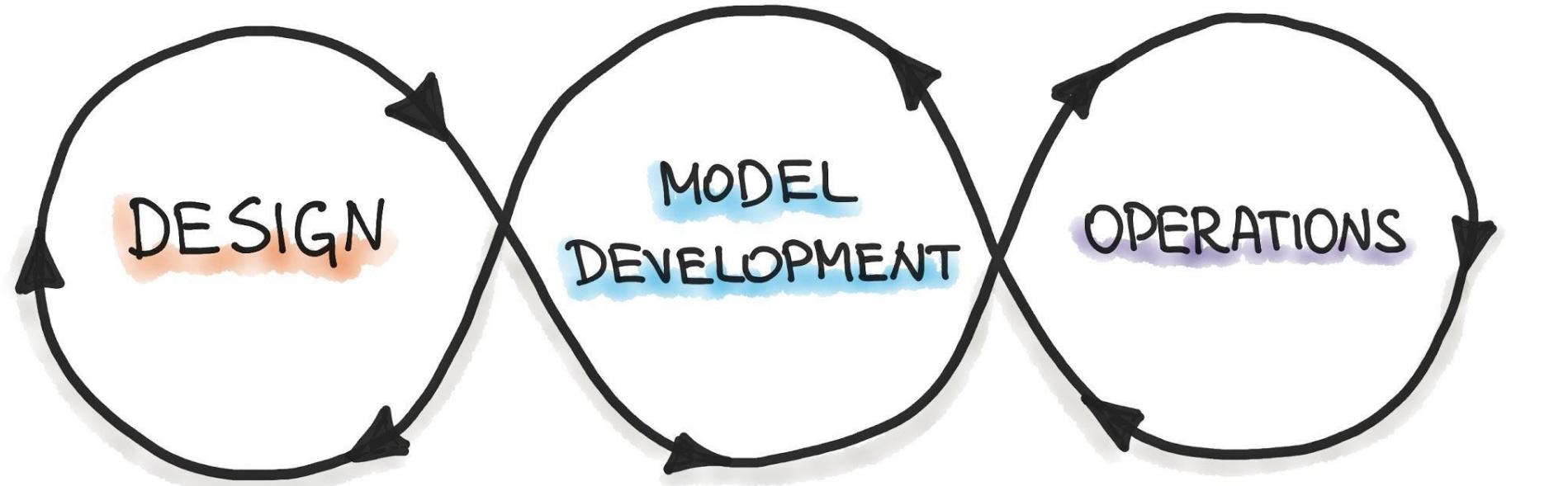


2020



2021

# MLOps



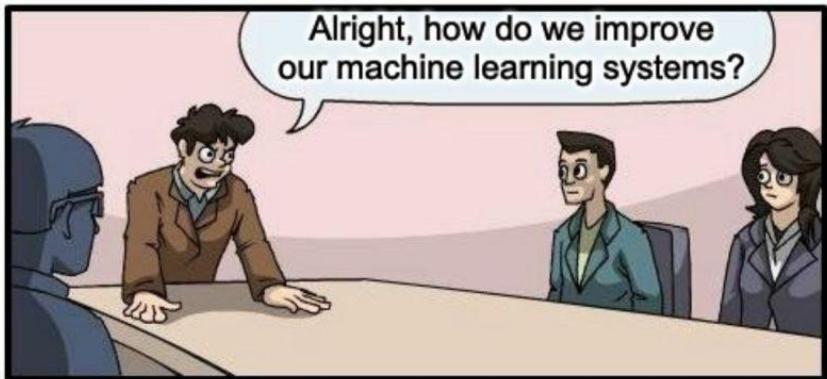
- Requirements Engineering
- ML Use-Cases Priorization
- Data Availability Check

- Data Engineering
- ML Model Engineering
- Model Testing & Validation

- ML Model Deployment
- CI/CD Pipelines
- Monitoring & Triggering

Давайте знакомиться!

# Заполняем опрос!



SCAN ME

<https://ok.me/RZqgX>

# О курсе

---

Проходит каждый вторник в 19-00

По формату лекции + демки (попробуем практики, но как пойдет)

По наполнению -- планируются лекции + демонстрации(лайв кодинг, демо инструментов и тд), но возможны и практики(с активным участием, посмотрим как пойдет этот формат).

На курсе будет 5 обязательных домашних работ. За все это дело можно будет набрать 100 баллов, критерии будут при выдаче (ближайшая на следующем занятии)

На 5 - 85 баллов

На 4 - 65 баллов

На 3 - 45 баллов

+ могут быть необязательные активности, которые также будут оцениваться(все гибко).

Приступаем к делу =)

```
background-color: #F5F5F5;
text-shadow: 0px -1px 0px #EAEAEA;
filter: dropshadow(color:#EAEAEA);
color:#777;

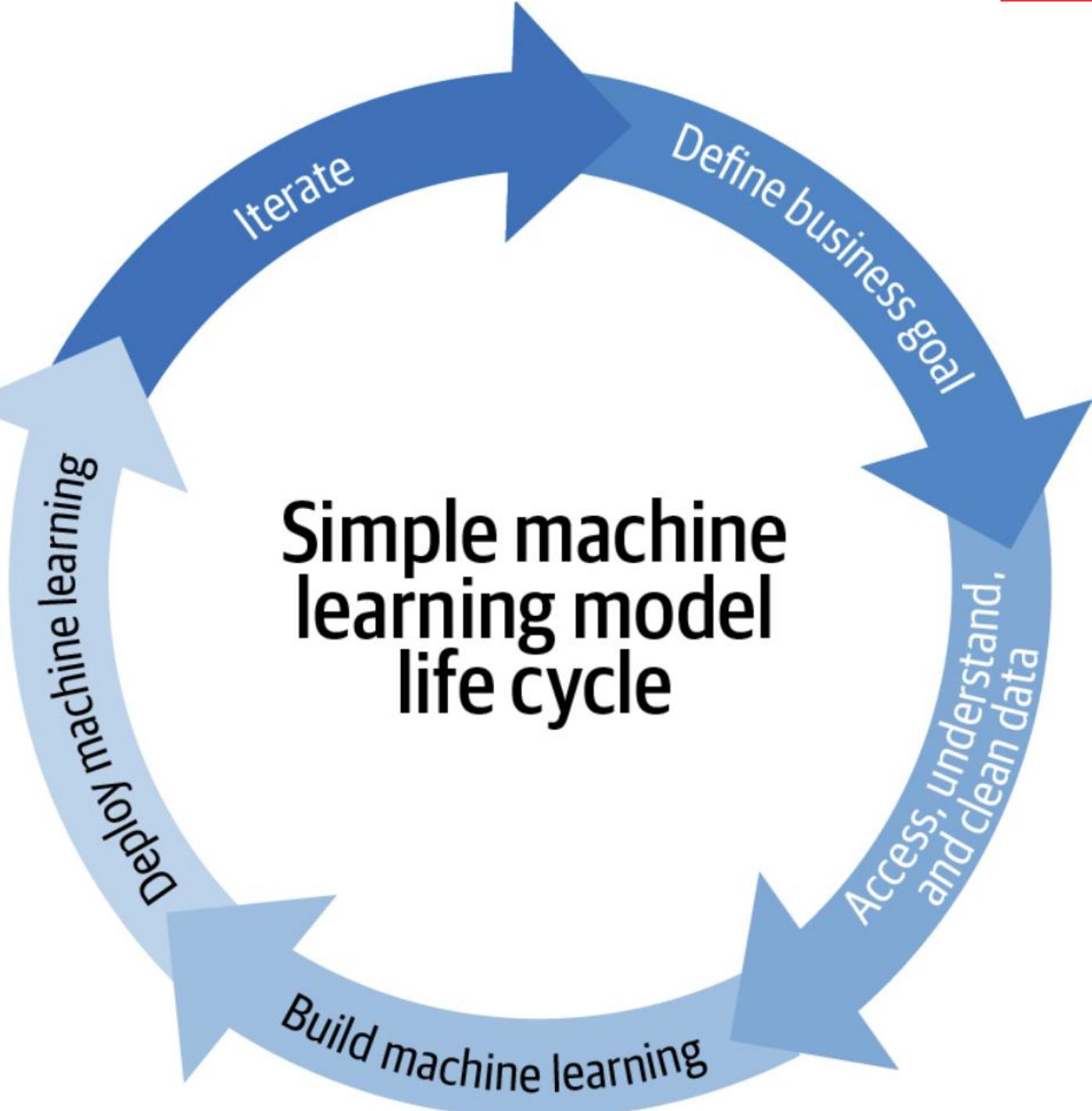
}
header #main-navigation ul li span:hover,
box-shadow: 0px 0px 1px #EAEAEA;
-webkit-box-shadow: 0px 0px 2px #EAEAEA;
moz-box-shadow: 0px 0px 1px #EAEAEA;
background-color:#F9F9F9;
active span,
li span.dashboard,
..../img/dashboard.png
ul li span.dashboard,
finaldashboard.html
finaldashboard.html
```



# План занятия

---

- 1) Обсуждаем различные этапы жизненного цикла ML
  - а) Постановка целей
  - б) Работа с данными
  - в) Тренировка моделей
  - г) Деплой
  - д) Мониторинг
  - е) Итерация
- 2) Роли в ДС
- 3) GIT



Основные этапы  
разработки модели

Определяем цели



# Примеры “целей”

---

- 1) Сделать нейронную сеть, которая будет детектировать спам.
- 2) Внедрить в продакшен 100 моделей
- 3) Разобраться, как работает BERT



# Примеры целей

---

- 1) Снизить количество людей, которые видят спам до 1%
  
- 1) Снизить количество людей, которые не возвращают большие кредиты(более 1 млн рублей до 0.05% от всех, кому вы выдаем такие.
  
- 2) Повысить среднее время пользования нашим сервисом до 100 минут в день.

# Цели (SMART)

---

S M A R T



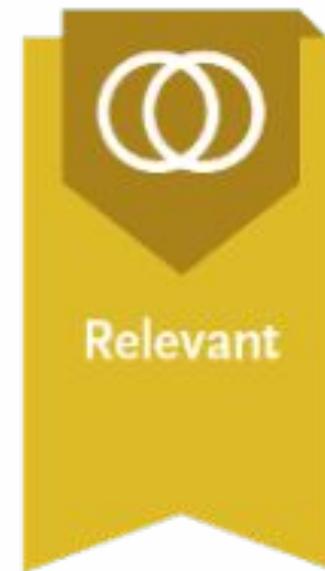
Specific



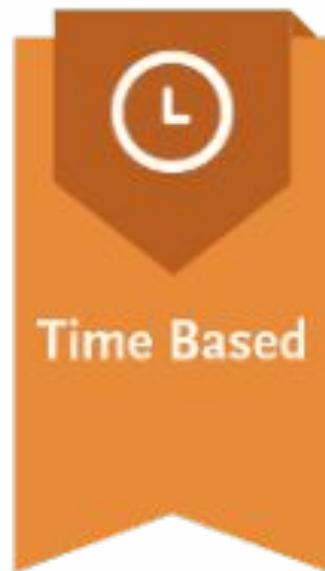
Measurable



Attainable

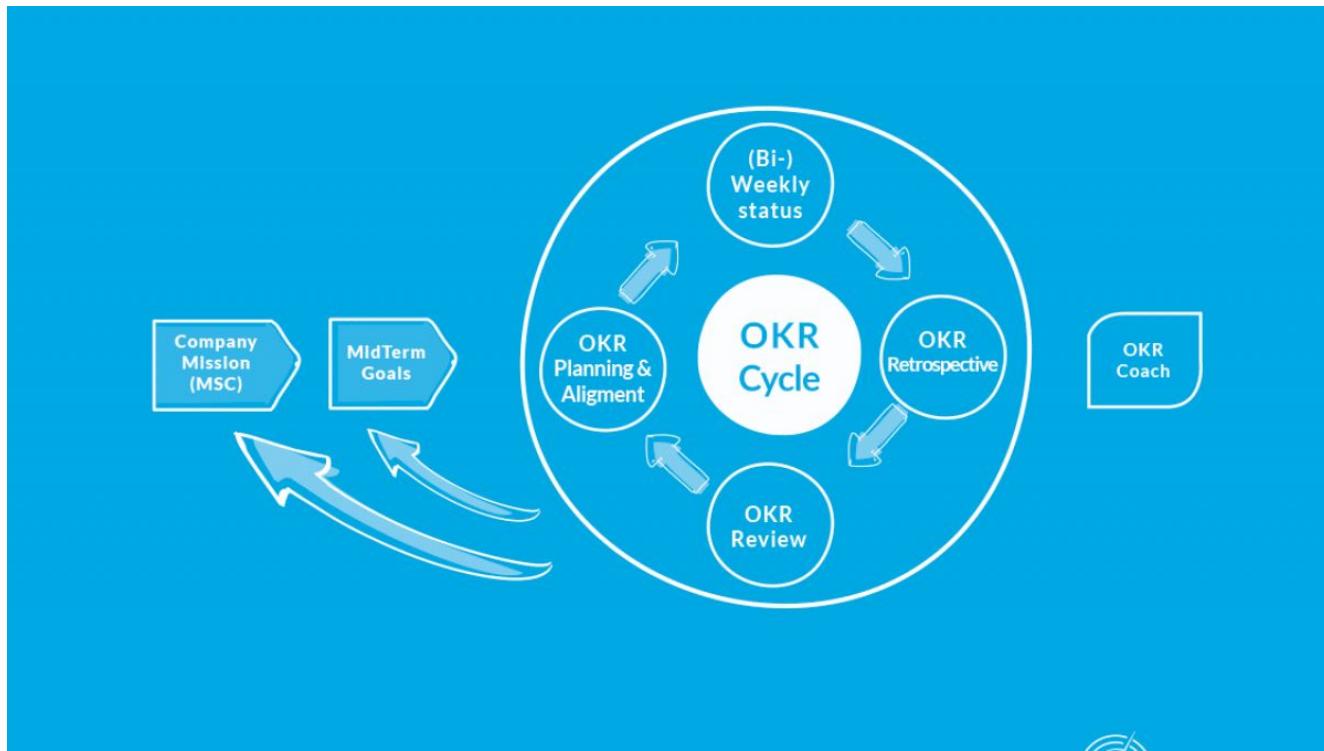


Relevant



Time Based

# OKR (objectives and key result)



Иерархическая структура целей, от главных ключевых целей для организации до целей конкретных сотрудников.

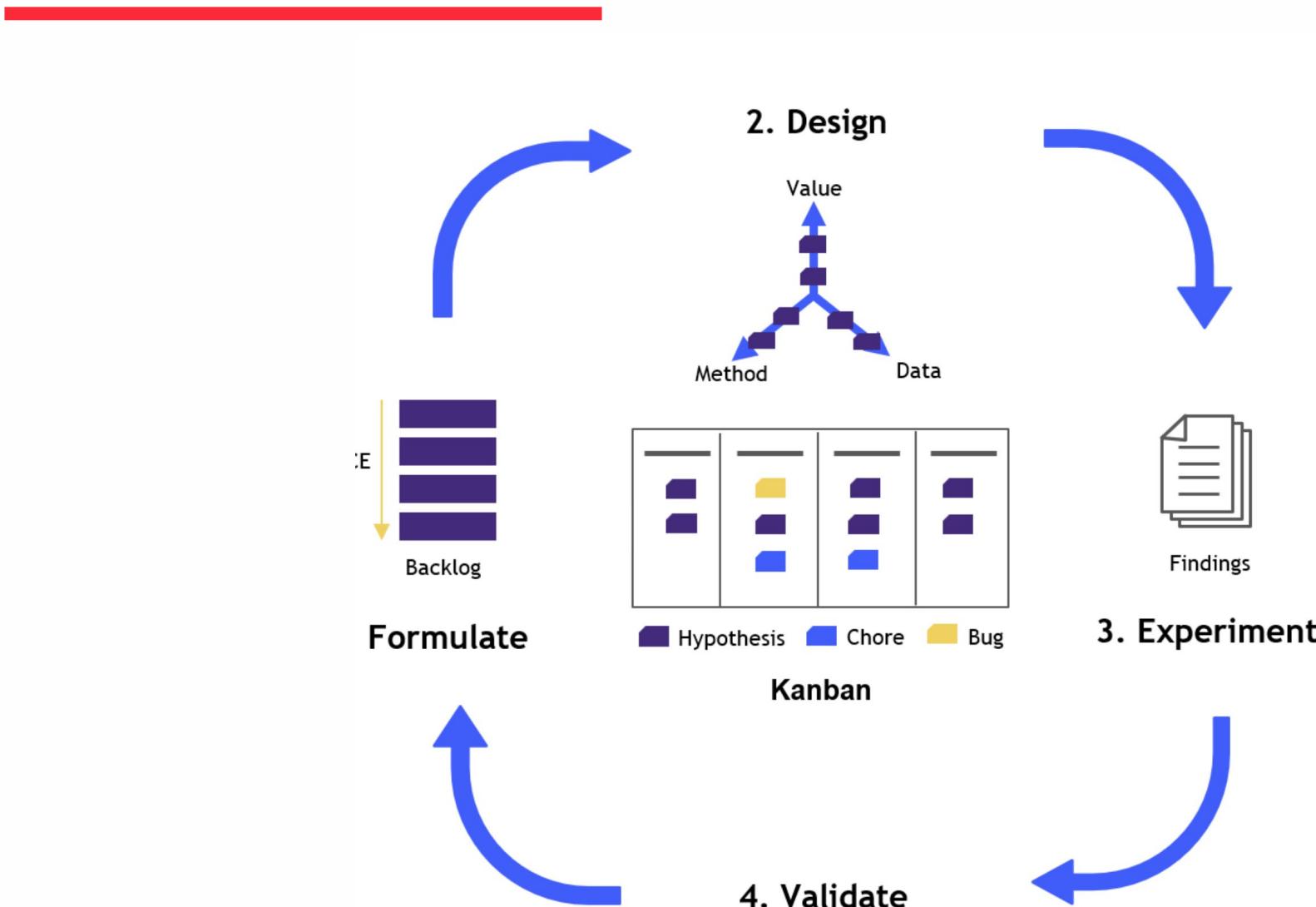
Все, что мы делаем — ведет к достижению ключевых целей

# Доклады по OKR/проверке гипотез

---



# Lean Machine Learning





# ИТОГИ

---

1. Не делаем то, что делать не надо
2. Используем SMART для формулировки целей
3. Думаем об измерении бизнес метрик

# Работа с данными



# Зачем нам данные?

---

Учим на них модели

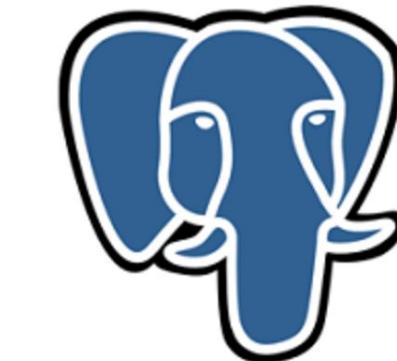
Ищем в них инсайты

Делаем по ним предсказания

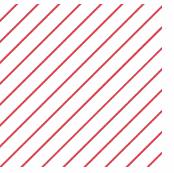
Оцениваем по ним результат

# Data Store - данные должны где-то храниться

---



Postgre<sup>SQL</sup>



# Вопросы к данным

---

Какие релевантные датасеты доступны?

Достаточно ли точны и надежны данные?

Как получить доступ к этим данным?

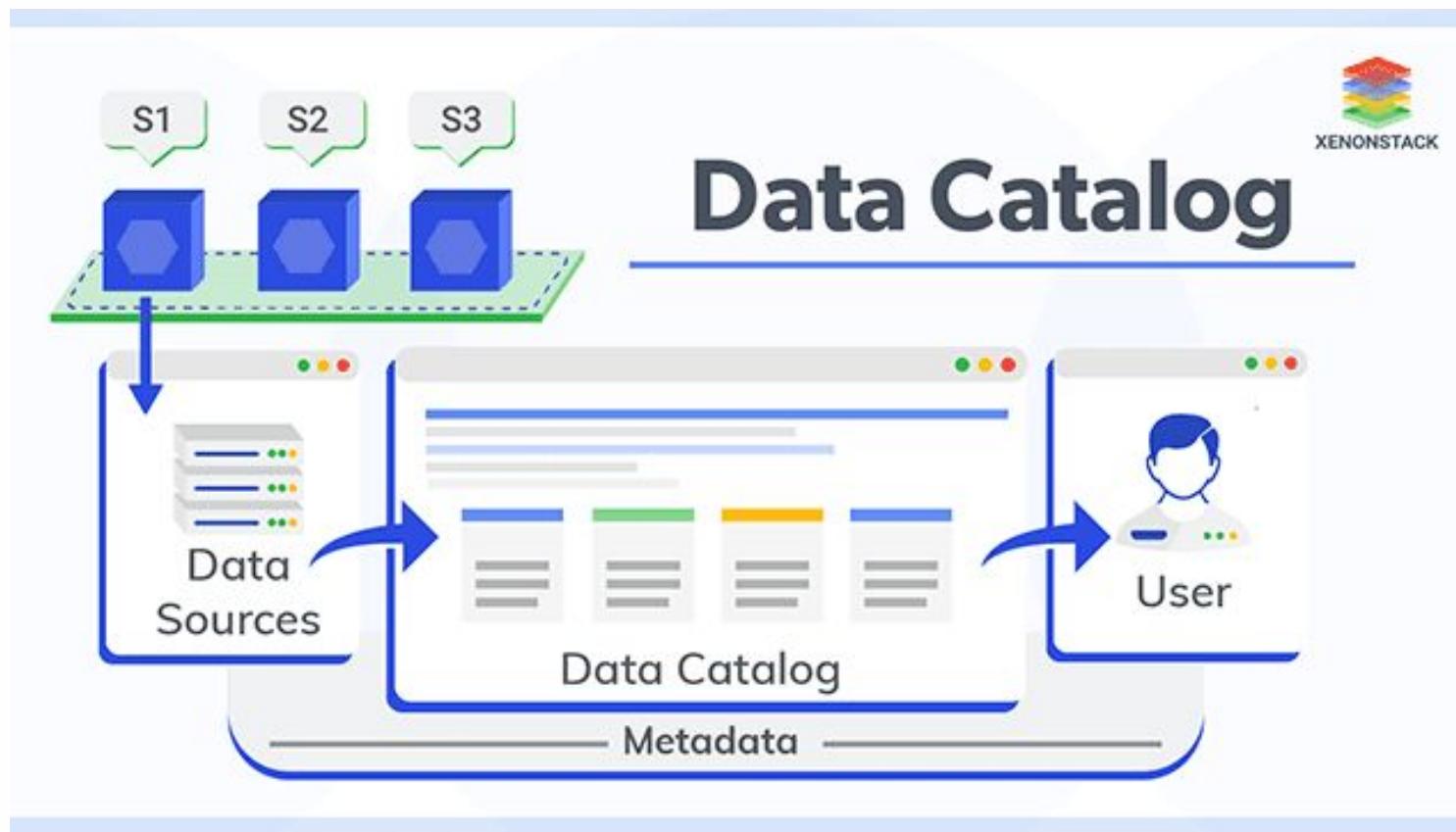
Какие фичи можно получить при джойне?

Часто ли обновляются данные?

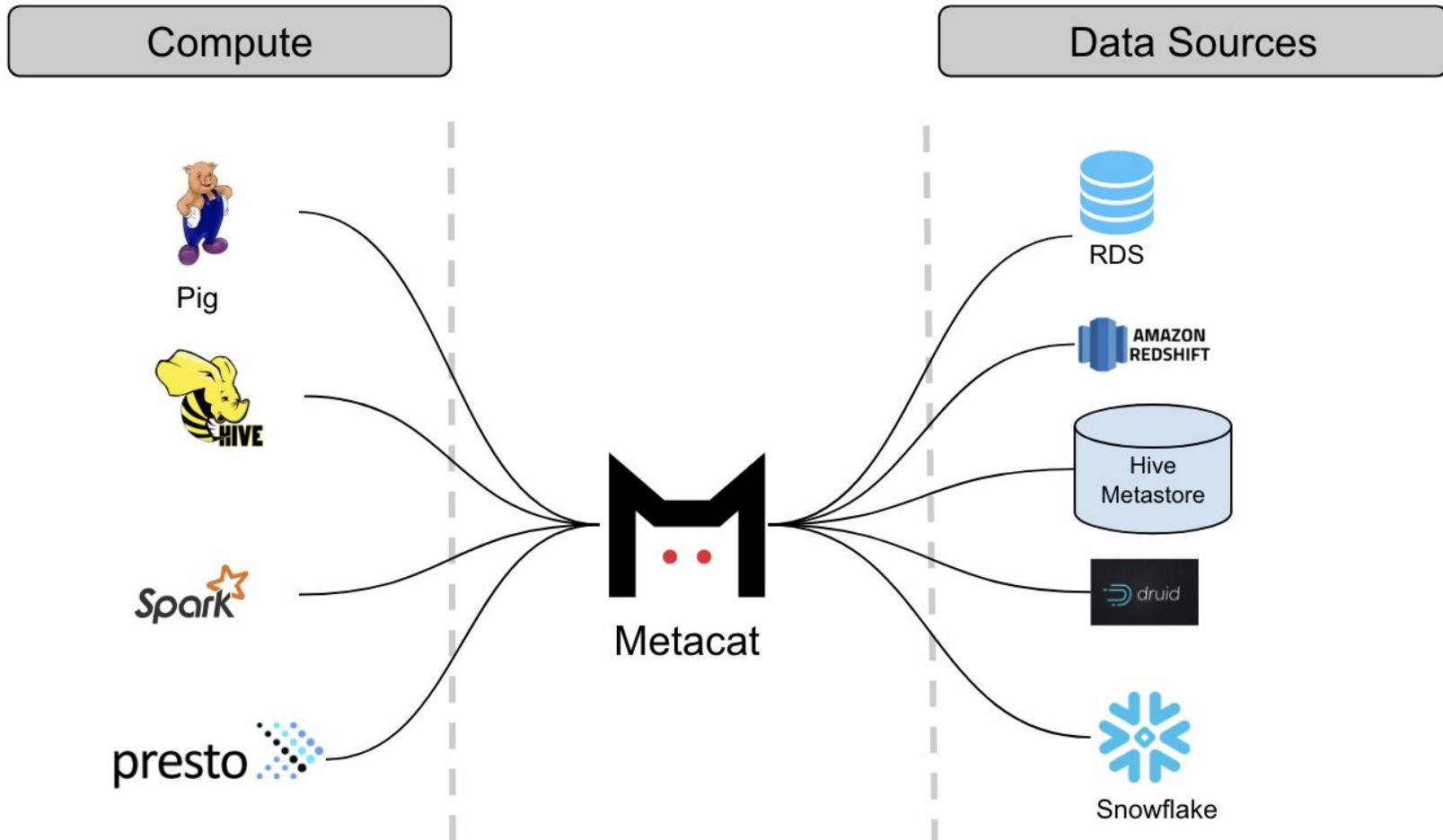
Можно ли будет получать эти фичи в realtime?

# Data Catalog

---

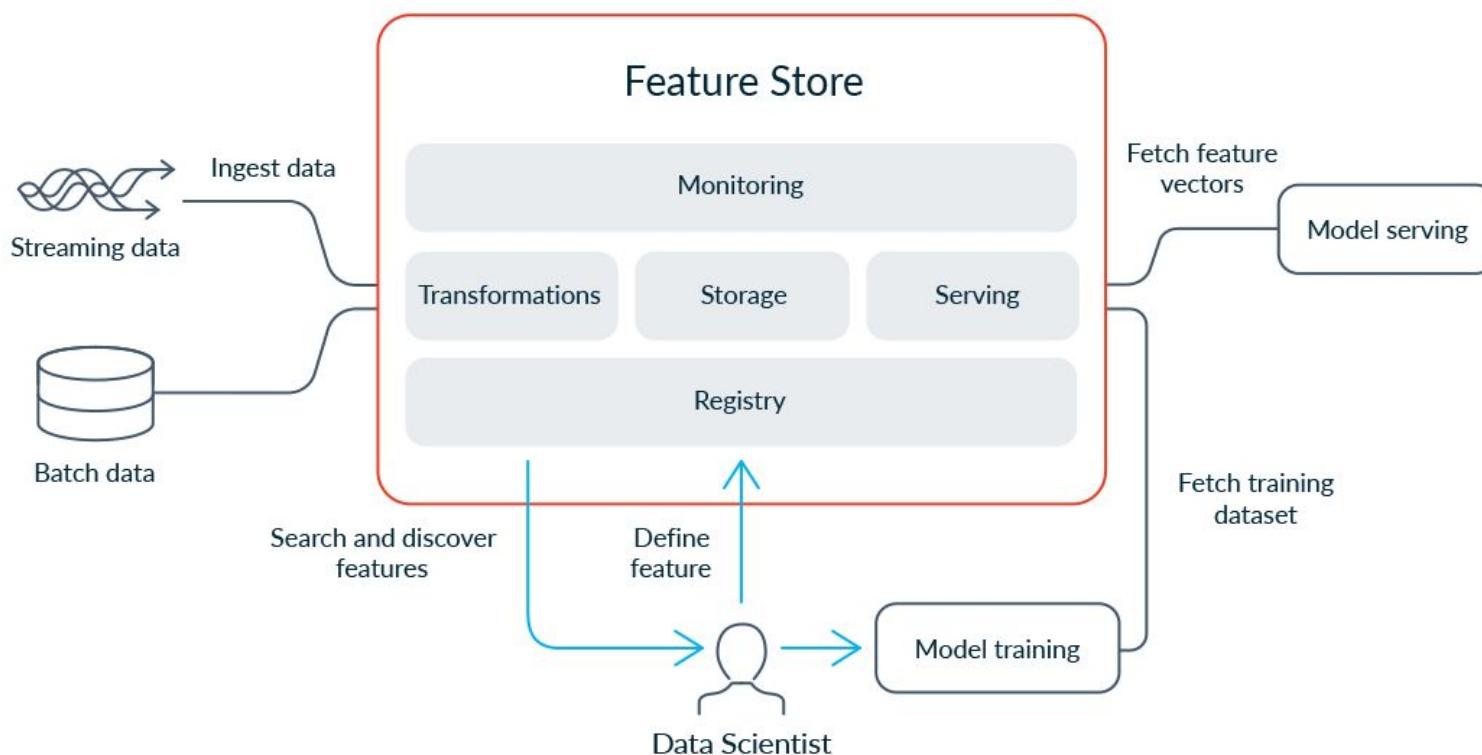


# Data Catalog



# Feature Store

Доступ к фичам в реал-тайме





# ИТОГИ

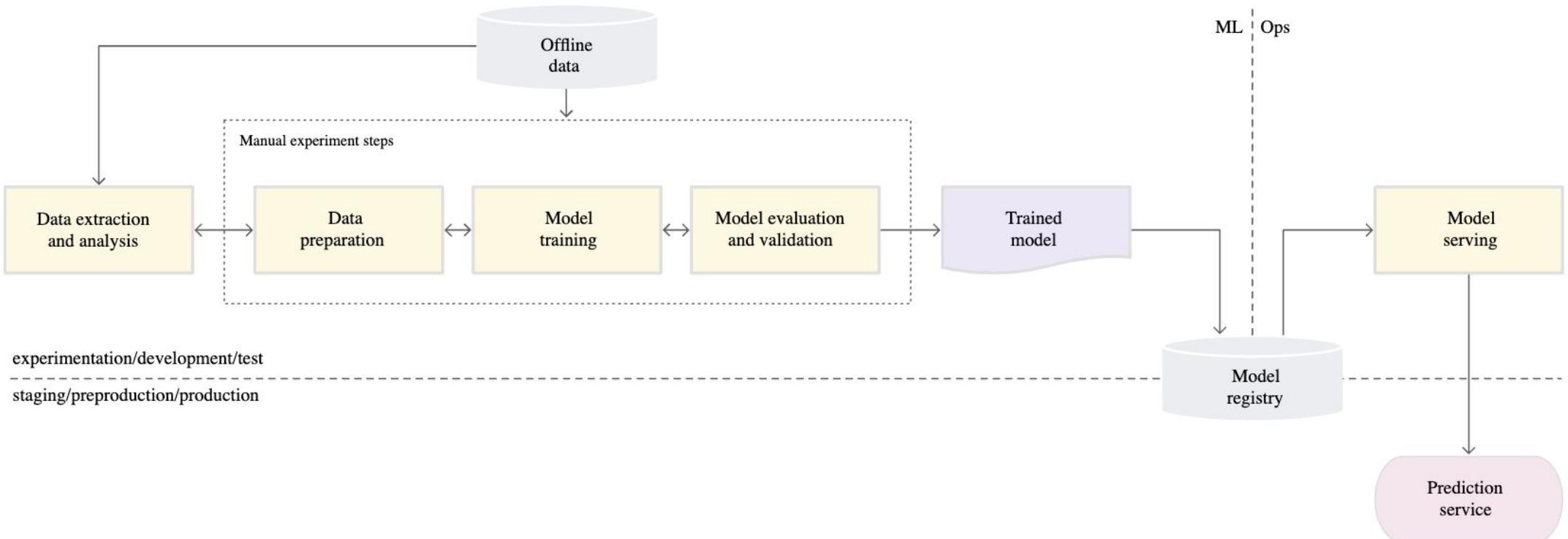
---

1. Данные храним =)
2. Стремимся уметь отвечать на вышеизложенные вопросы

Тренировка и оценка качества  
моделей.

# MLOPS Level 0

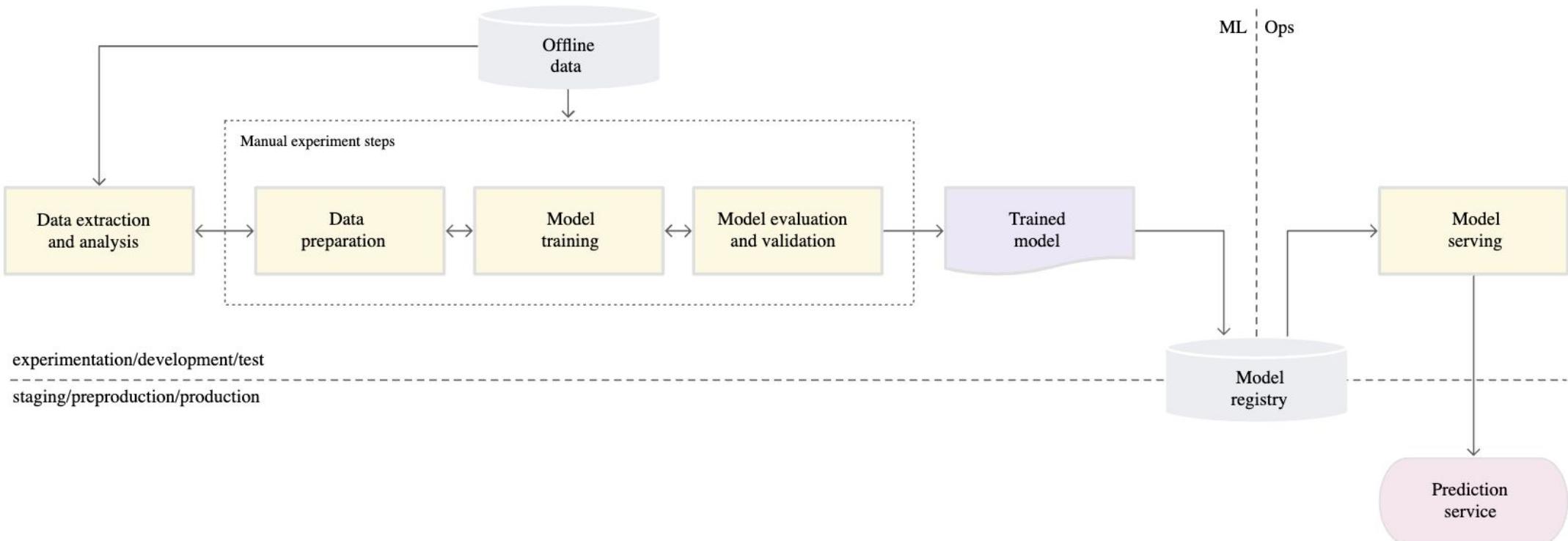
DS делает модели



# MLOPS Level 0

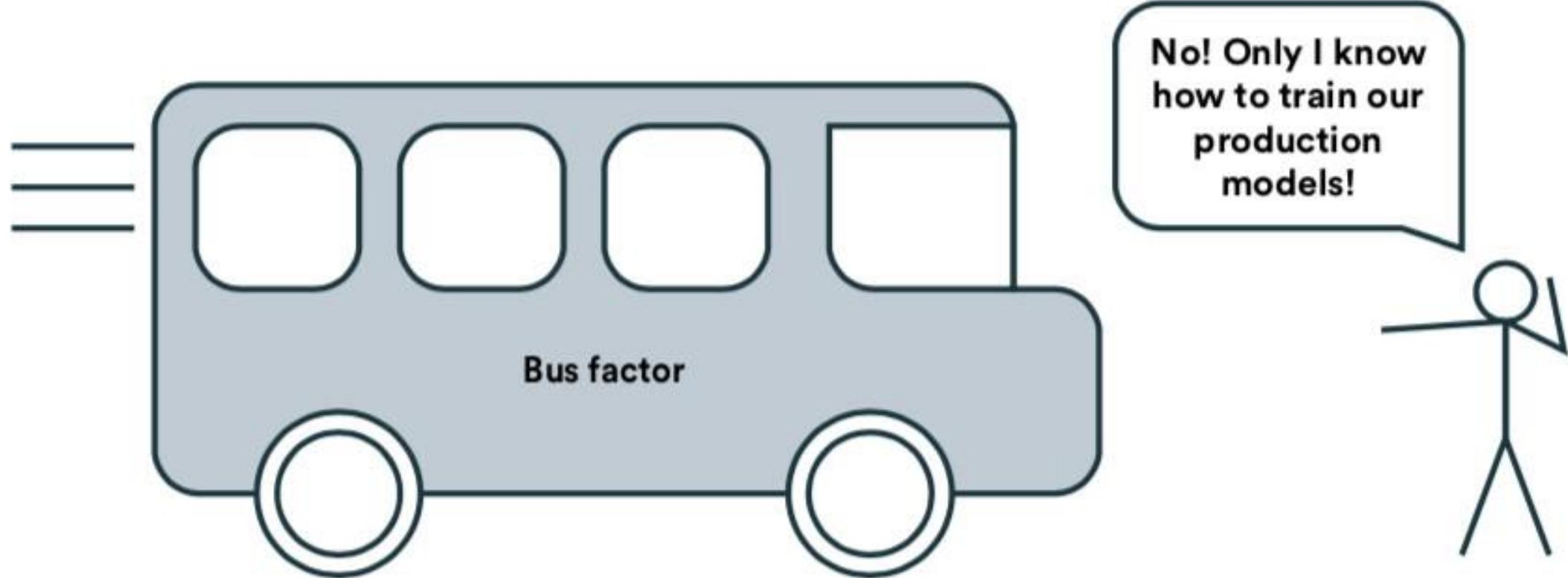


DS делает модели



# BUS FACTOR

---





# Проблемы

---

- Потери данных
- Потери кода
- Потери знаний о том, как обучать и выкапывать



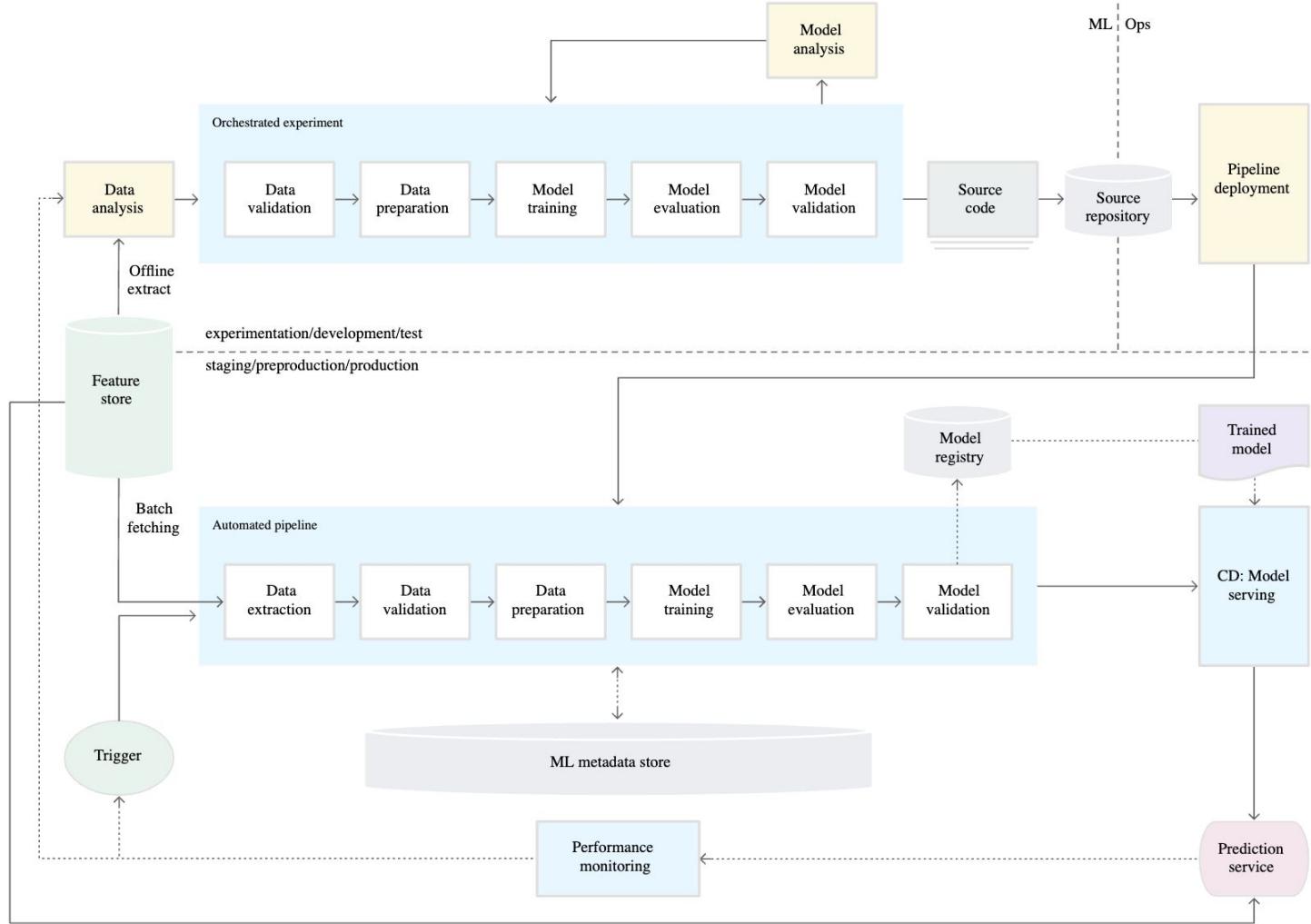
# Training-Serving Skew

---

- Наши данные не такие, как на проде
- Фичи получаются не тем же способом, что на проде
- DS не думает о том, что на проде
- Теряется способ воспроизвести модель

# MLOPS Level 1

DS делает пайплины,  
которые делают модели



# Что поможет реализовать?

---



Версионирование  
кода



Шедулинг и  
оркестрация



Версионирование  
данных



Трекинг экспериментов



# ИТОГИ

---

1. Нужно писать переиспользуемый, версионируемый, тестируемый код
2. Нужно осознание того, что модель — это не цель

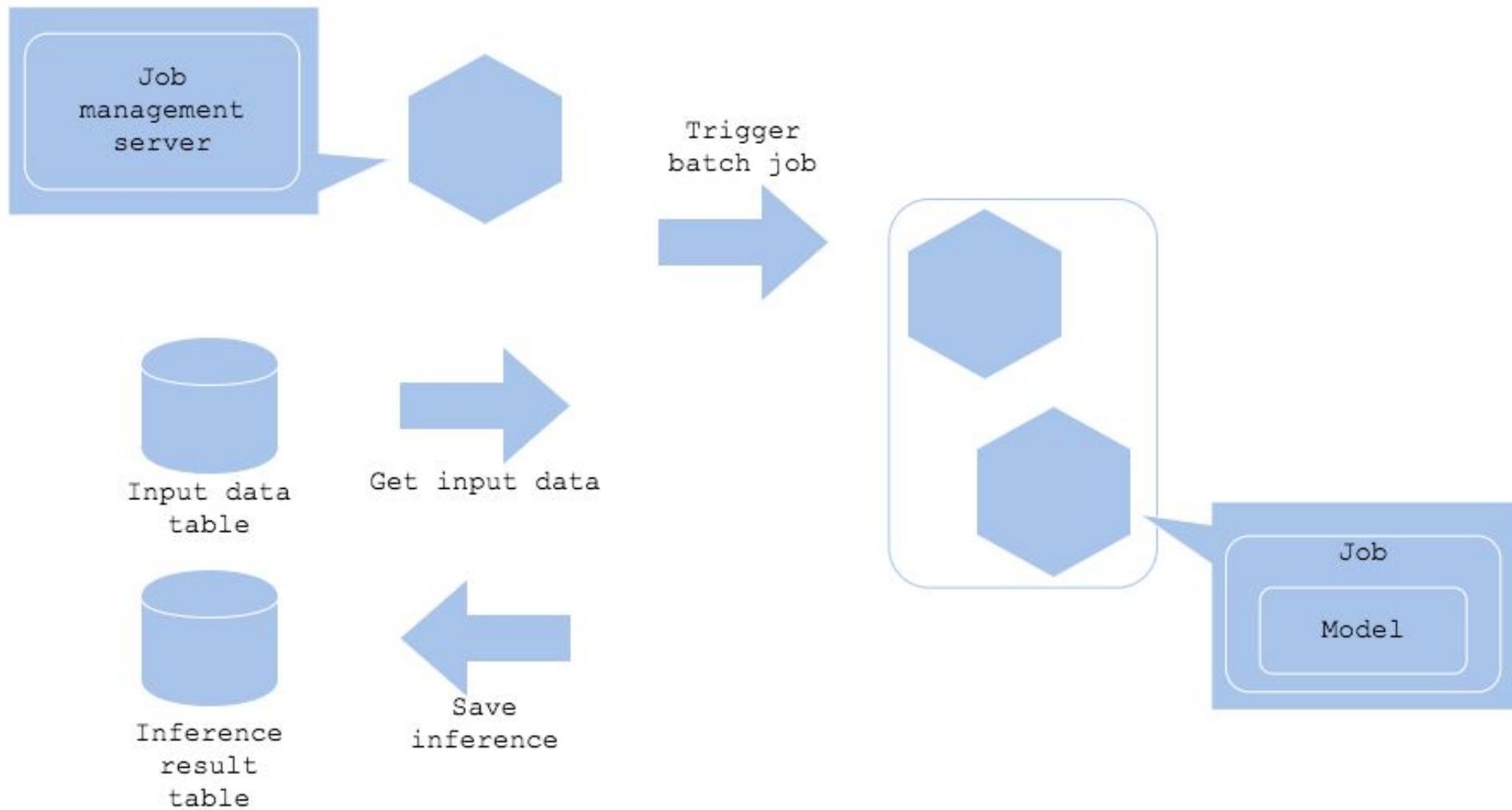
# Использование моделей

# Классический банковский ML

Index	ПОЛ	ВОЗРАСТ	ЗП	PREDICT
0	М	23	300 к/сек	1
1	М	33	45 000 р/мес	0.5
2	М	34	15 000 р/мес	0.1
3	Ж	55	55 000 р/мес	0.7

AI →

# Пакетный паттерн





# Когда нужно использовать?

---

- Если вам **НЕ** нужно получать результат прогноза в реальном времени или почти в реальном времени.
- Для массивной обработки данных
- Когда для корректной работы продакшена достаточно запускать PREDICTION по расписанию(раз в сутки, в час, в 10 минут)



# Что поможет реализовать?

---



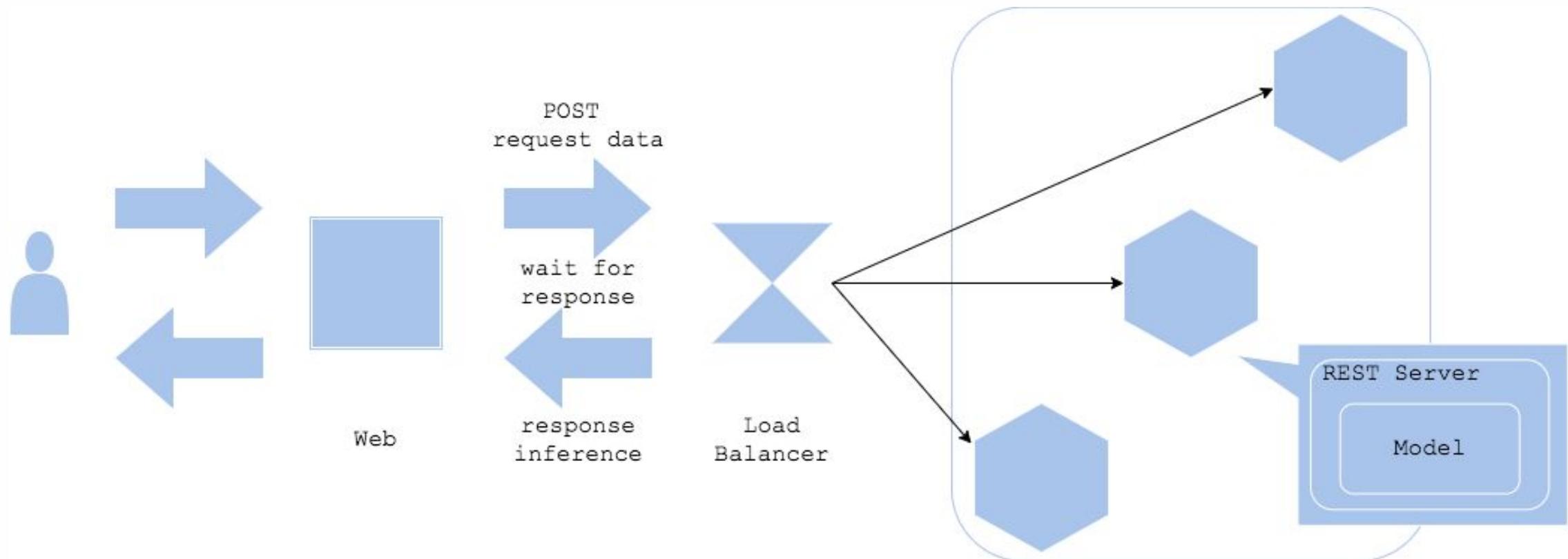
Apache  
**Airflow**

Загружаешь фотку - получаешь ответ

---



# Синхронный паттерн



[https://github.com/mercari/ml-system-design-pattern/blob/master/Serving-patterns/Synchronous-pattern/design\\_en.md](https://github.com/mercari/ml-system-design-pattern/blob/master/Serving-patterns/Synchronous-pattern/design_en.md)



# Синхронный паттерн

---

- когда вам нужен результат предсказания для того, чтобы сделать следующий шаг

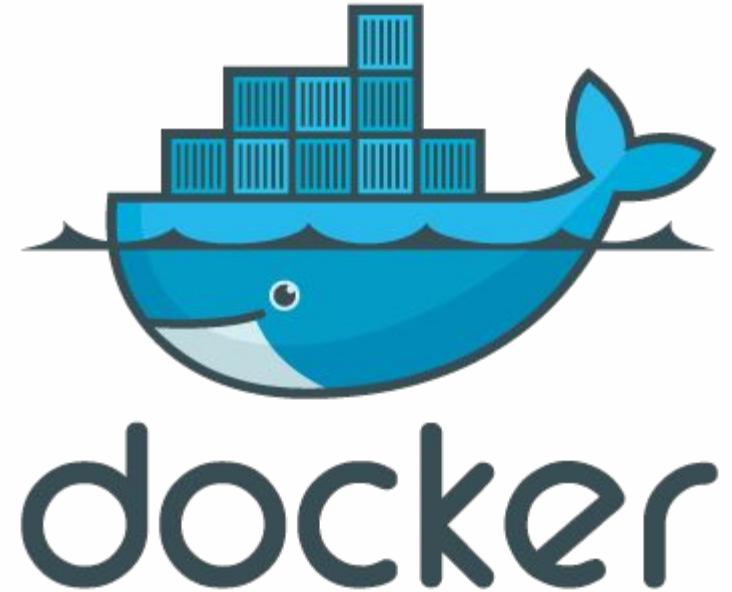
ПЛЮСЫ	МИНУСЫ
Очень просто реализовать	Скорость предсказания будет бутылочным горлышком производительности вашей системы
Как правило, низкое latency	Если предсказание слишком долгое, то нужно сделать так, чтобы пользователь этого не замечал

[https://github.com/mercari/ml-system-design-pattern/blob/master/Serving-patterns/Synchronous-pattern/design\\_en.md](https://github.com/mercari/ml-system-design-pattern/blob/master/Serving-patterns/Synchronous-pattern/design_en.md)



# Что поможет реализовать?

---



# Удаление спам сообщений в Instagram



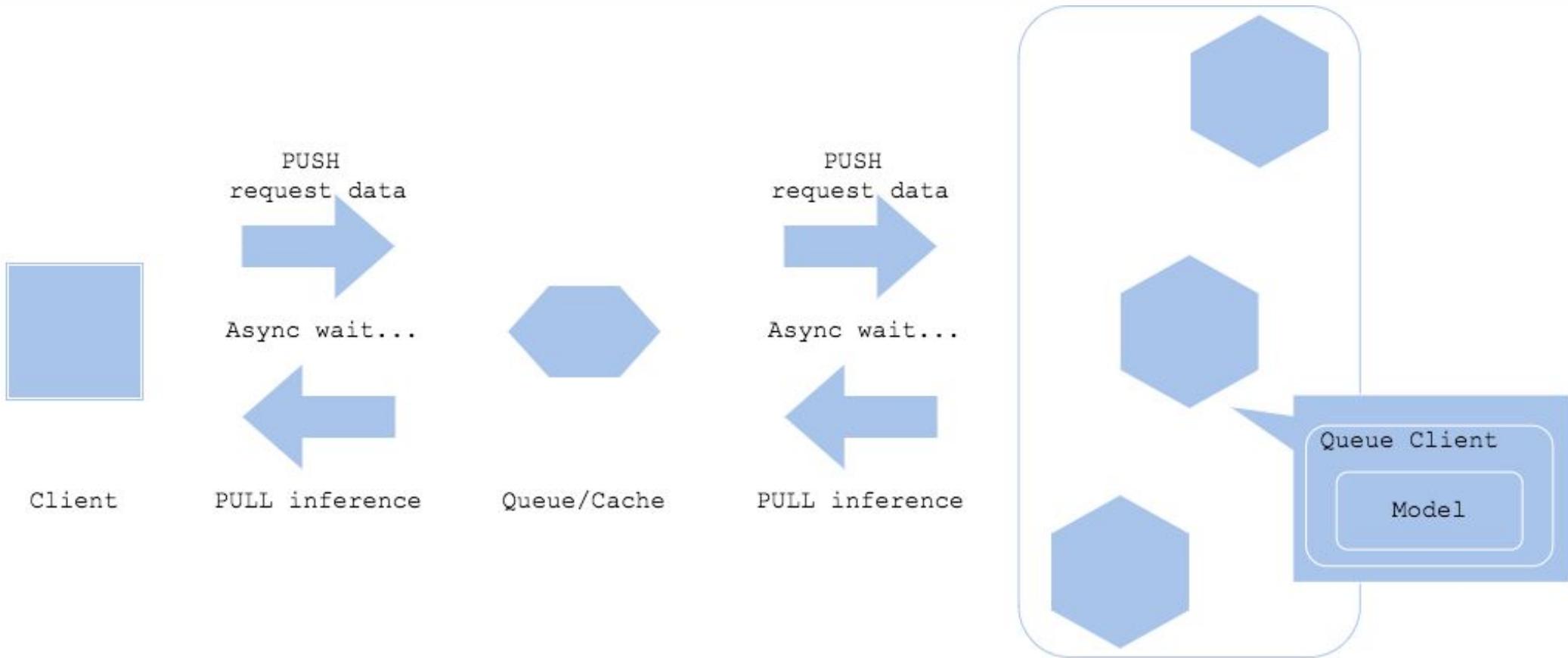
pavelvolyaofficial • Подписаться

irina\_almazova\_proff Классные вы ❤️  
\_777777755 Шоколад и мармелад ❤️  
anikeeva600 Вы классные!!! Любите  
друг друга.... И будьте счастливы!!!  
iiodas ❤️  
sweetypresentbouquet Очень  
красивая пара 😊 ❤️  
nastyawol ❤️❤️❤️⭐⭐⭐  
marinavictory ❤️  
cornershop\_777 🔥🔥🔥  
**expressbeautykiev Привет!**  
Подпишитесь на нашу страницу и Вы  
окунетесь в атмосферу  
совершенства, красоты и  
привлекательности. 💁‍♀️蝶

master\_school\_socchi СЧАСТЬЯ ВАМ  
РЕБЯТА ❤️❤️❤️

273 933 отметок "Нравится"  
17 ЯНВАРЯ  
Добавьте комментарий... ⋮

# Асинхронный(near realtime) паттерн



[https://github.com/mercari/ml-system-design-pattern/blob/master/Serving-patterns/Asynchronous-pattern/design\\_en.md](https://github.com/mercari/ml-system-design-pattern/blob/master/Serving-patterns/Asynchronous-pattern/design_en.md)



# Асинхронный(near realtime) паттерн

---

- когда вам не нужен результат прямо сейчас

ПЛЮСЫ	МИНУСЫ
Можно отделить бизнес логику и логику предсказания	Как правило, не подходит для использования в реальном времени.
Более высокая(по сравнению с синхронным) пропускная способность	Нужны очереди/кеши, etc
Вы не заблокированы временем предсказания	

[https://github.com/mercari/ml-system-design-pattern/blob/master/Serving-patterns/Asynchronous-pattern/design\\_en.md](https://github.com/mercari/ml-system-design-pattern/blob/master/Serving-patterns/Asynchronous-pattern/design_en.md)



# Что поможет реализовать?

---





# ИТОГИ

---

1. Есть несколько сценариев использования ML моделей
2. Нужно уметь выбирать нужный в зависимости от ограничений, целей
3. Реализуются по разному

# Деплой

# Куда деплоить?

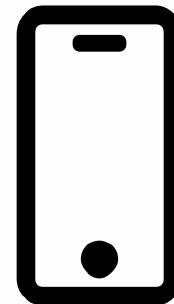
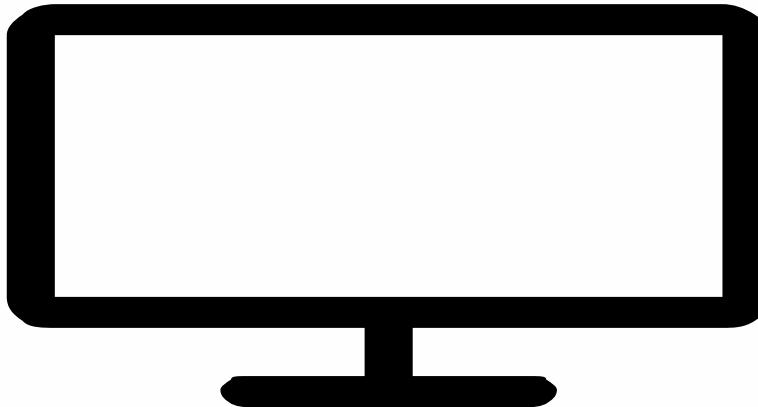
---



**kubernetes**



Google Cloud

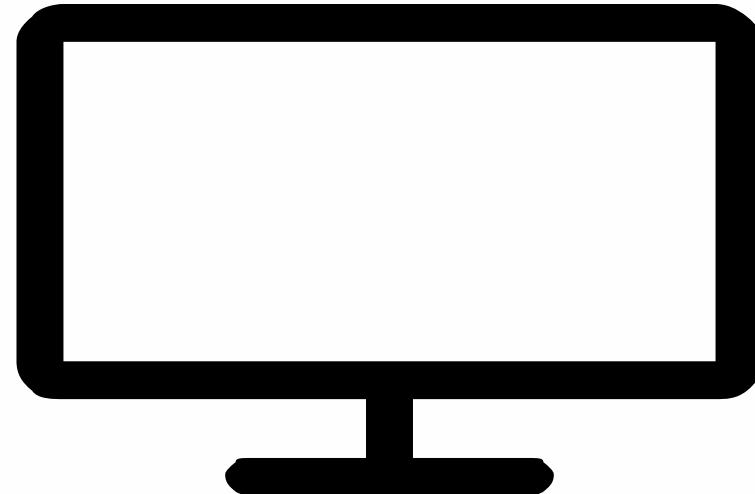
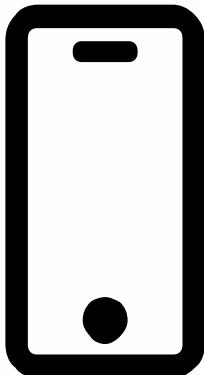




# Embedded

---

Использование модели встроено в приложение





## Embedded: характеристики

---

Нет сетевой задержки  
можно запустить на девайсе  
Сложно масштабировать  
независимо от приложения



# Model as a service

---

Использование модели вынесено в отдельный сервис



## Model as a service: характеристики

---

Сетевой задержка

Можно масштабировать  
независимо от приложения



## Масштабирование приложения

---

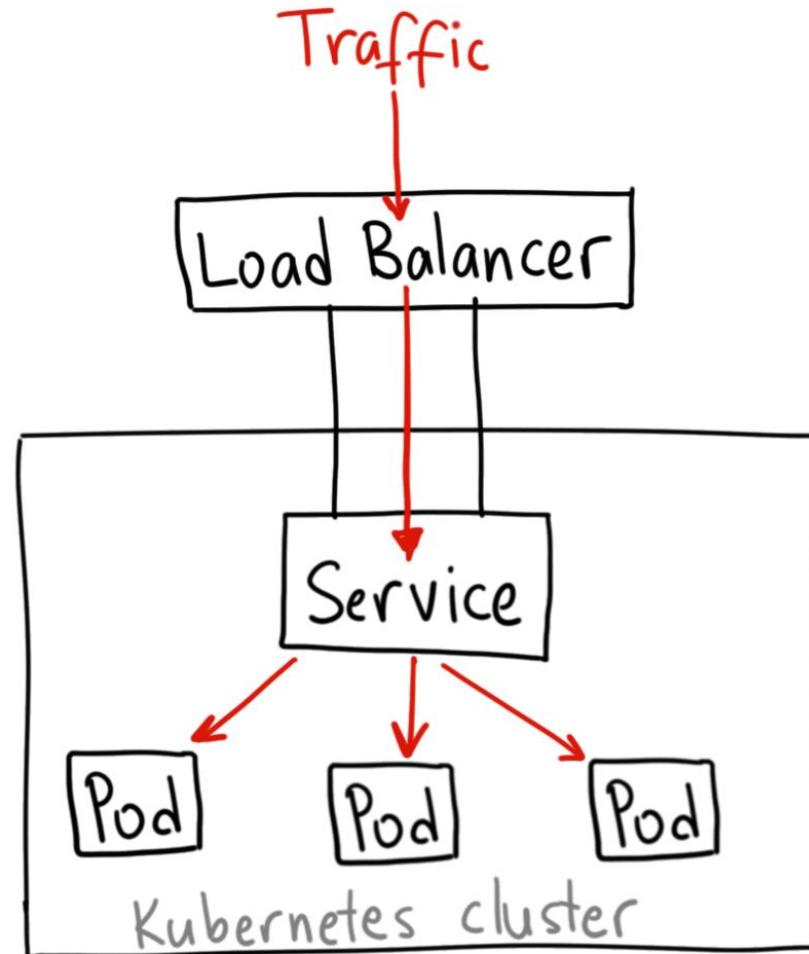
Сетевая задержка

Независимый релизный цикл

Можно масштабировать  
независимо от приложения

# Масштабирование

---



# UPDATE модели

---



V1.1 -> V1.2

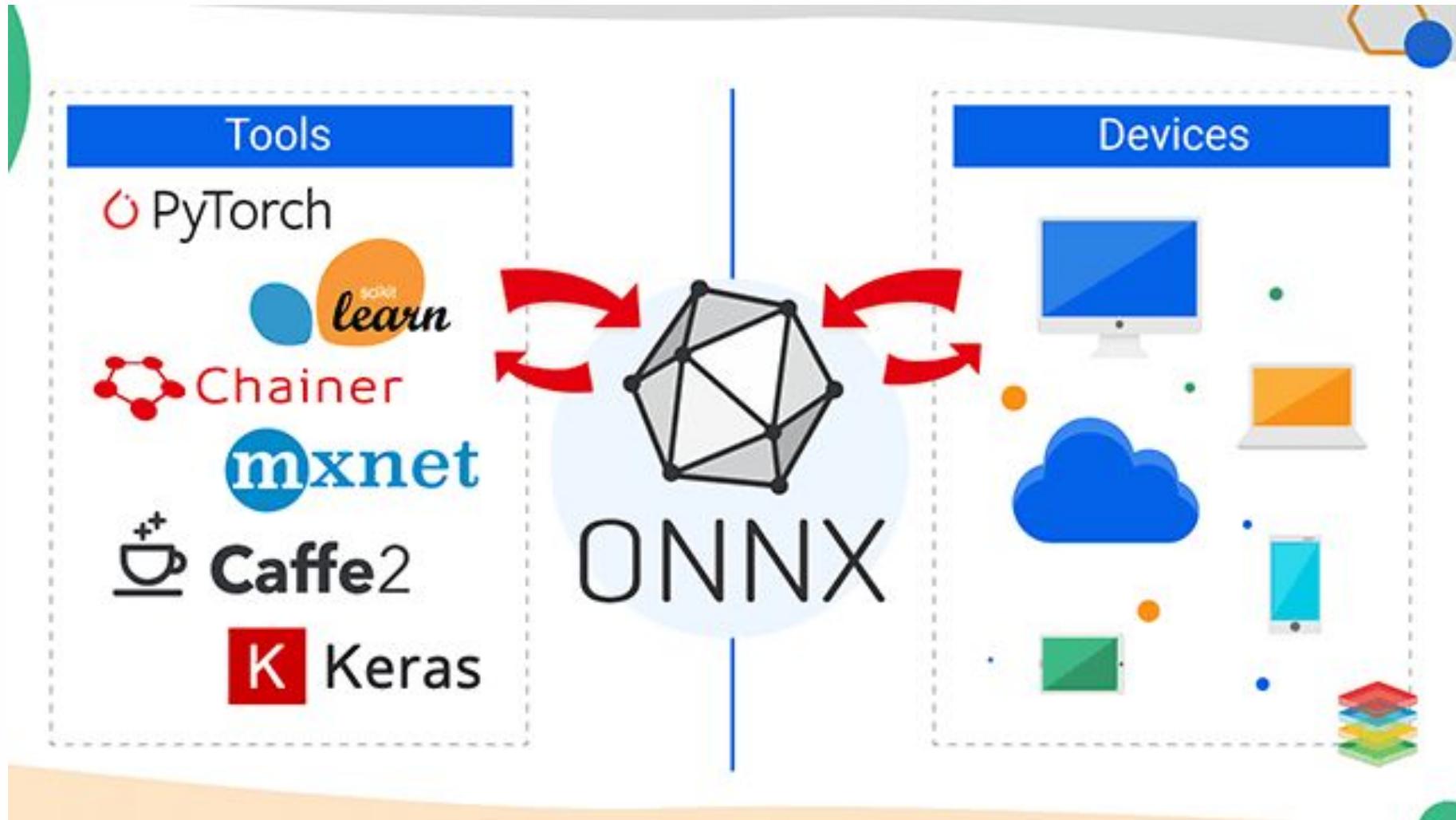
# Model as a code

---

```
if ($x[27] < 0.0735105276) {  
    if ($x[27] < 0.0153422952) {  
        if ($x[24] < 2.27298903) {  
            if ($x[27] < 0.00678618671) {  
                if ($x[24] < 1.7399652) {  
                    if ($x[2] < 1.82783937) {  
                        return 63;  
                    return 64;  
                if ($x[7] < 0.0550374165) {  
                    return 65;  
                return 66;  
            }  
        }  
    }  
}
```

# Model as a data

---





# Итоги

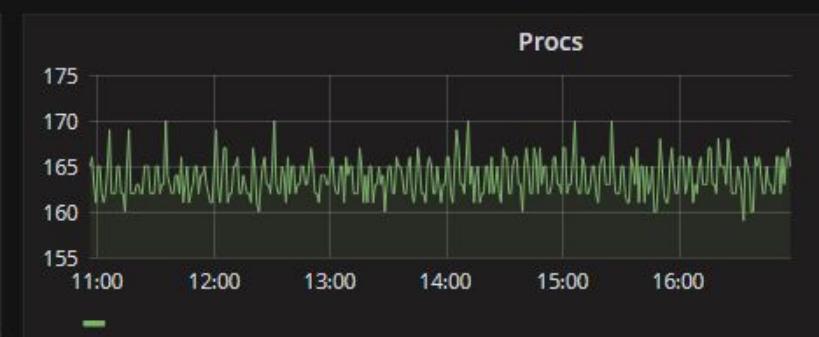
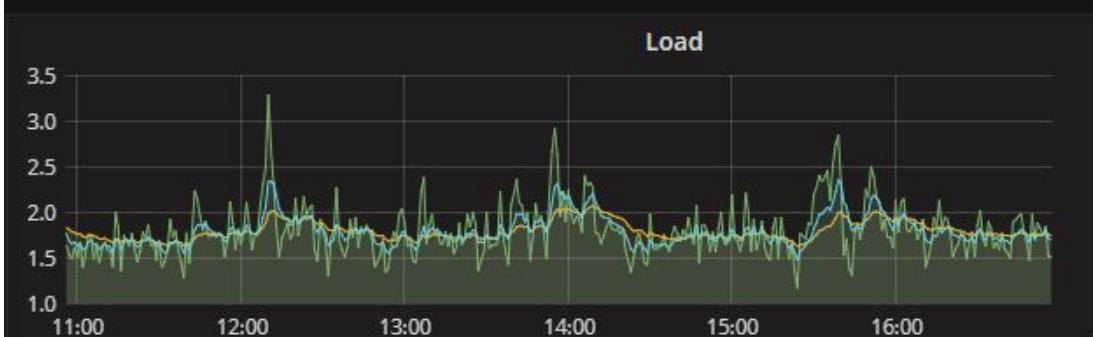
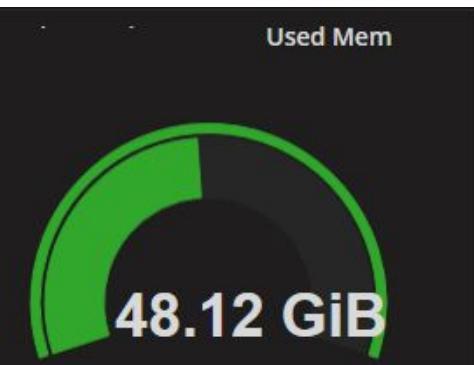
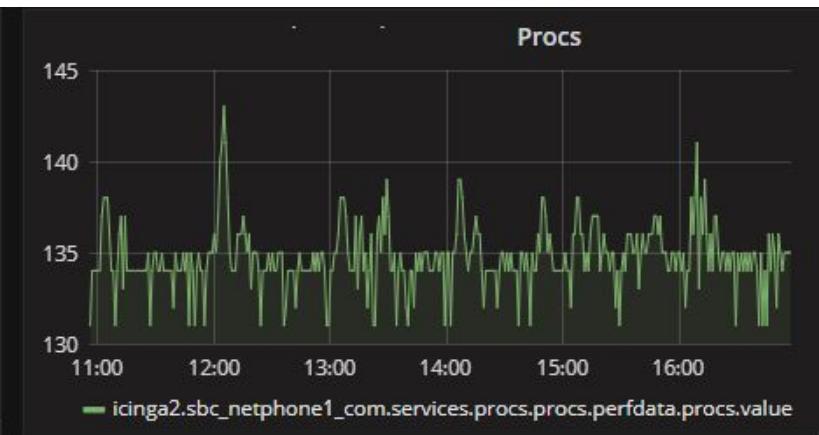
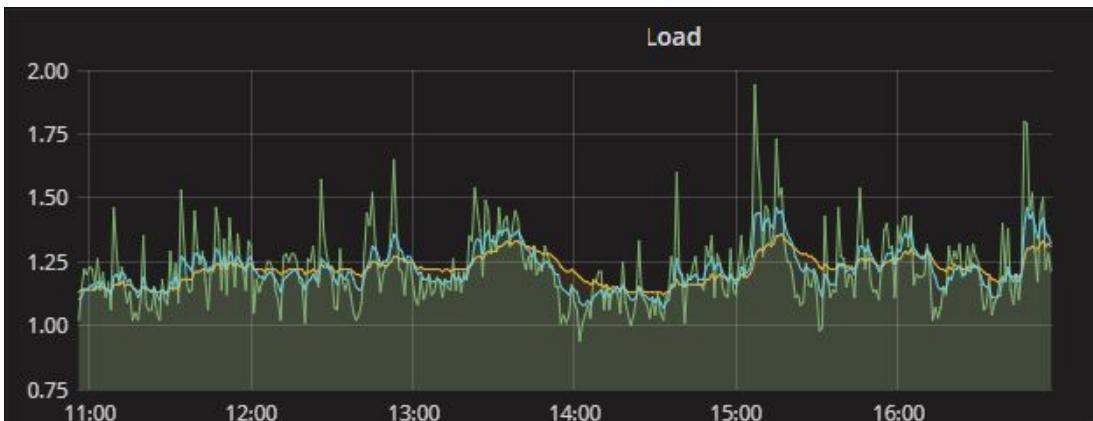
---

В зависимости от сценария использования мы будем деплоить в разные окружения

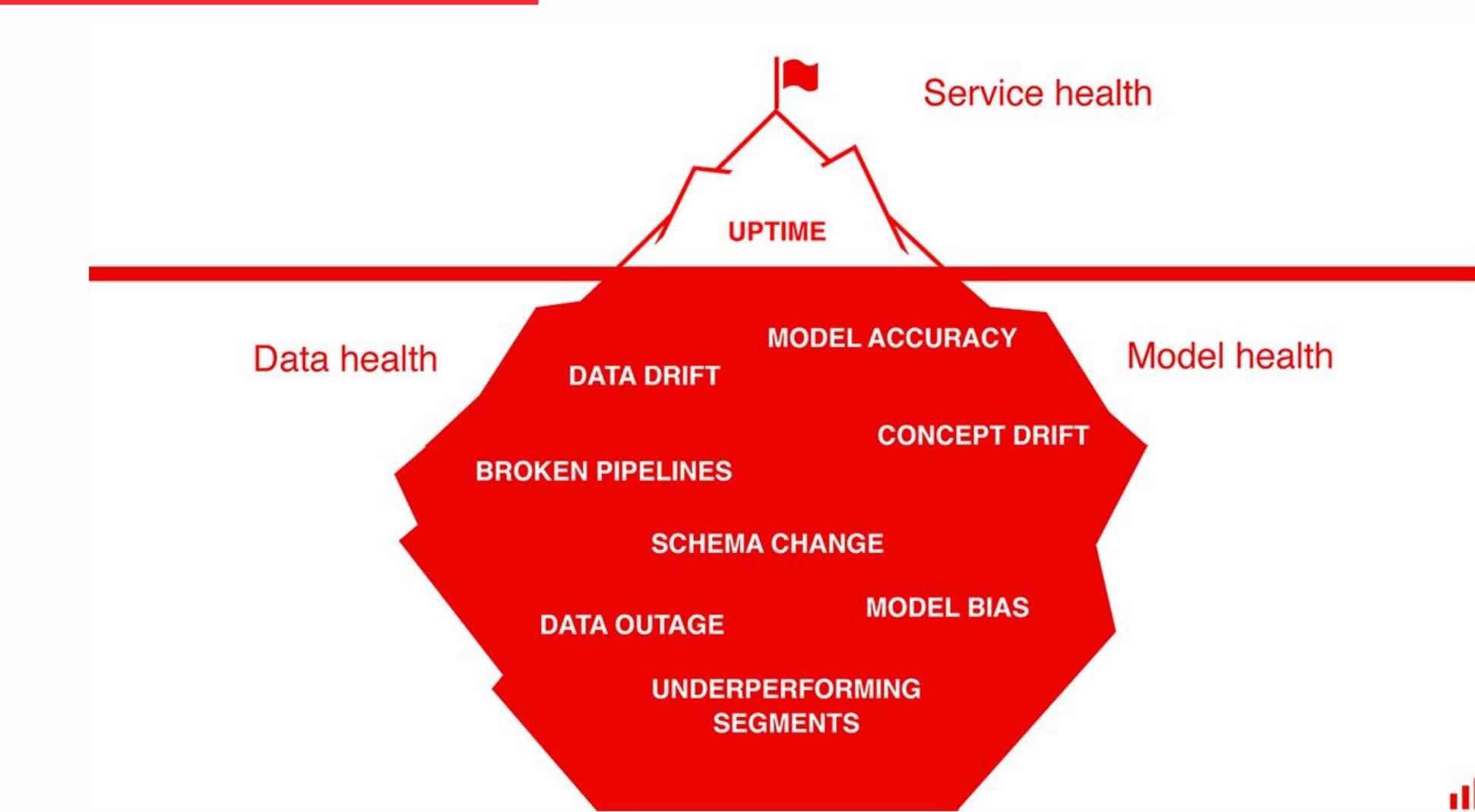
# Мониторинг

# Технические метрики

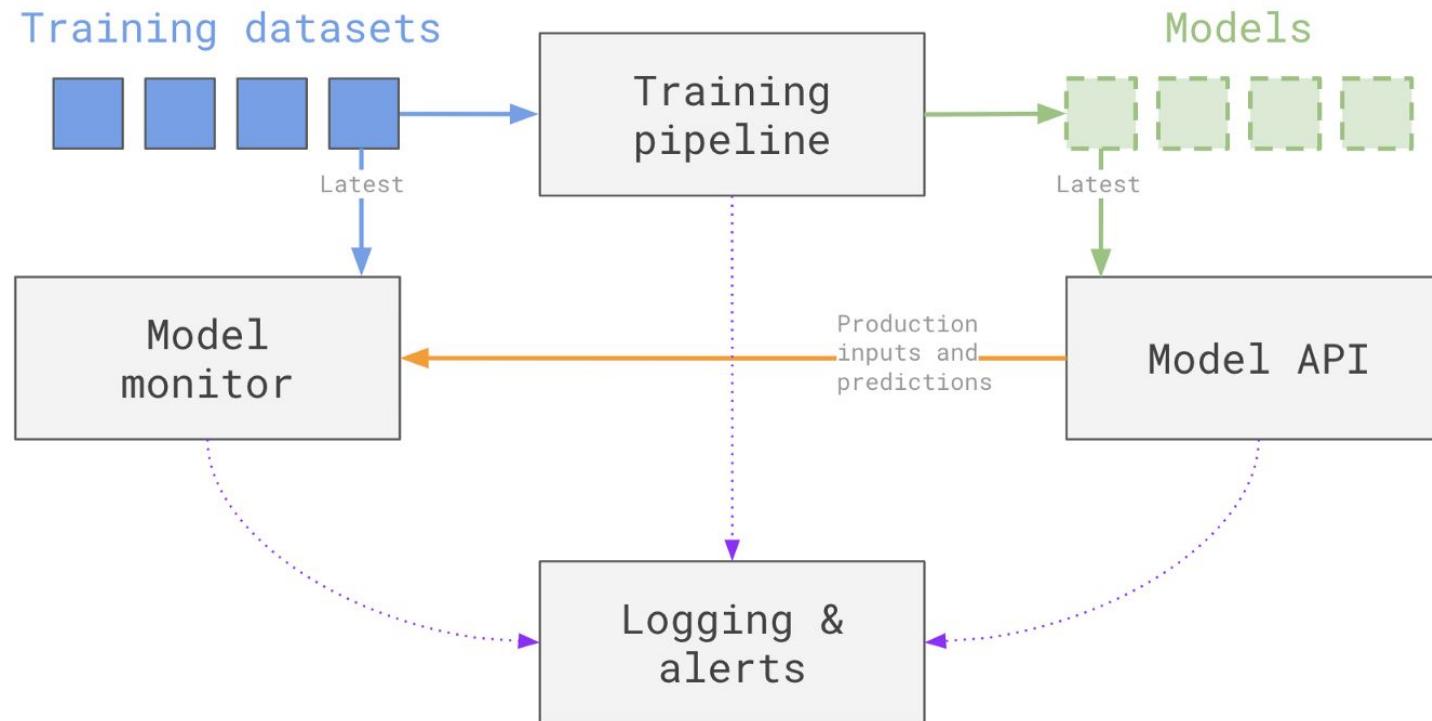
cput, gput, query per seconds, errors



# ML specific

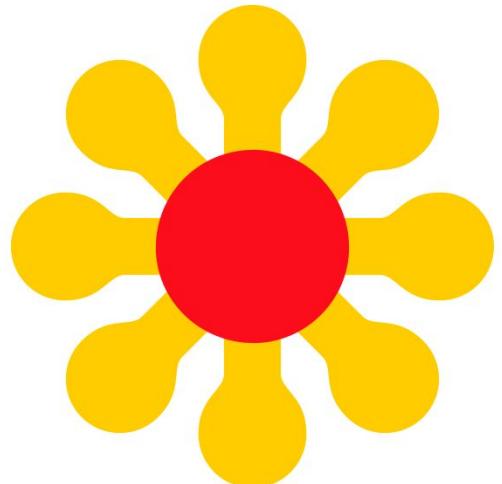


# Monitoring Scheme



# Что поможет реализовать?

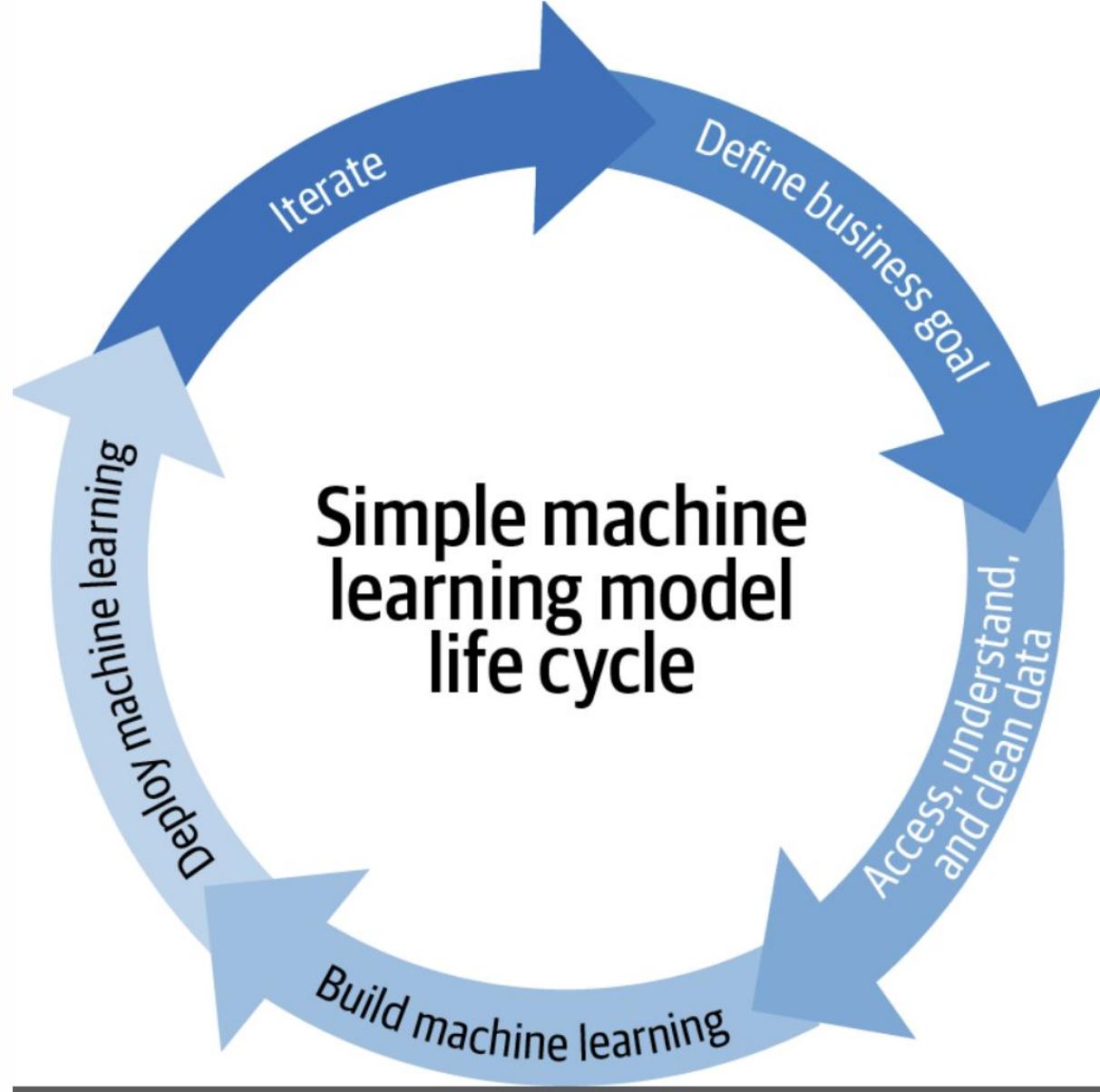
---



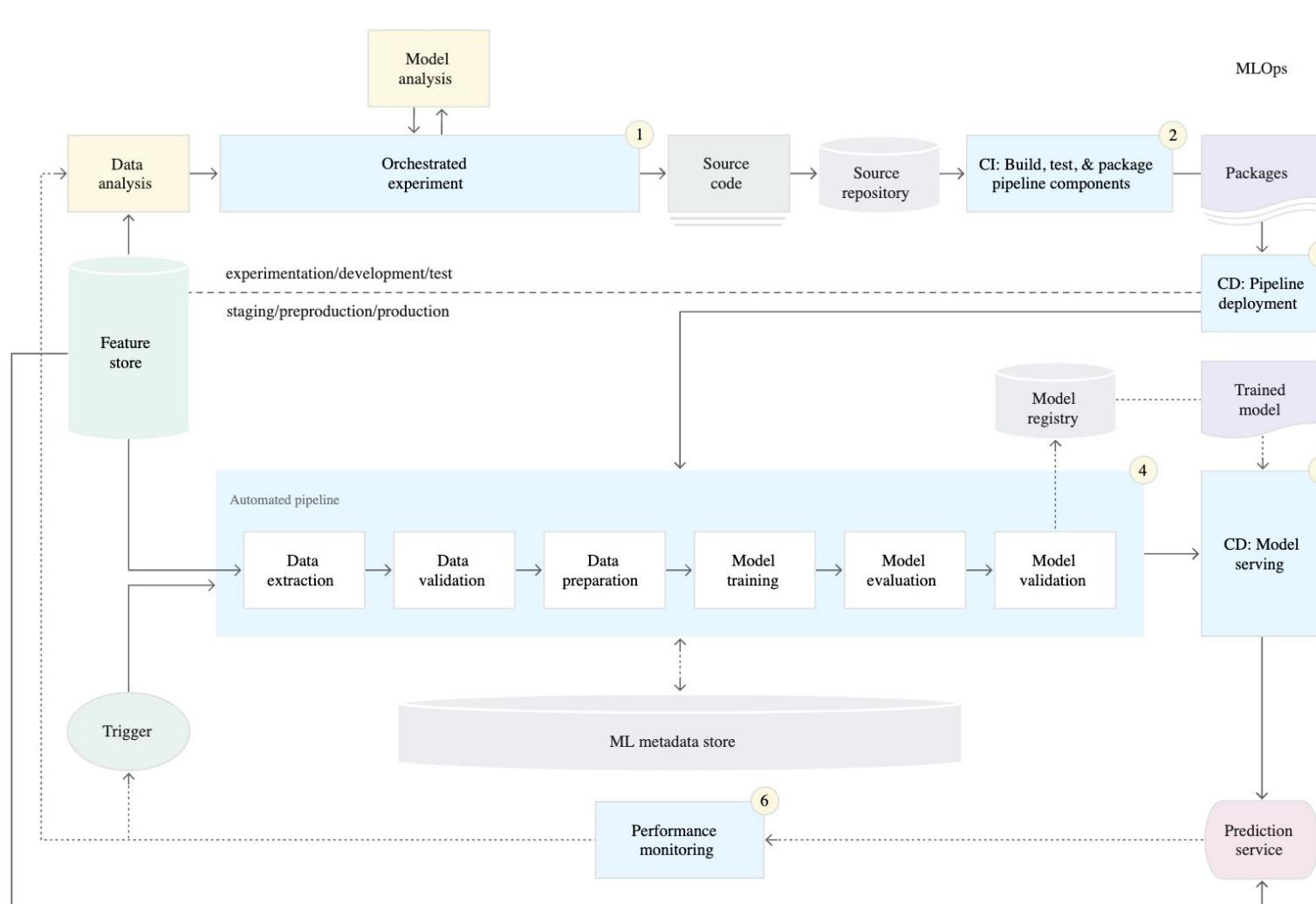
# Итерация

# Модель разработана не раз и навсегда

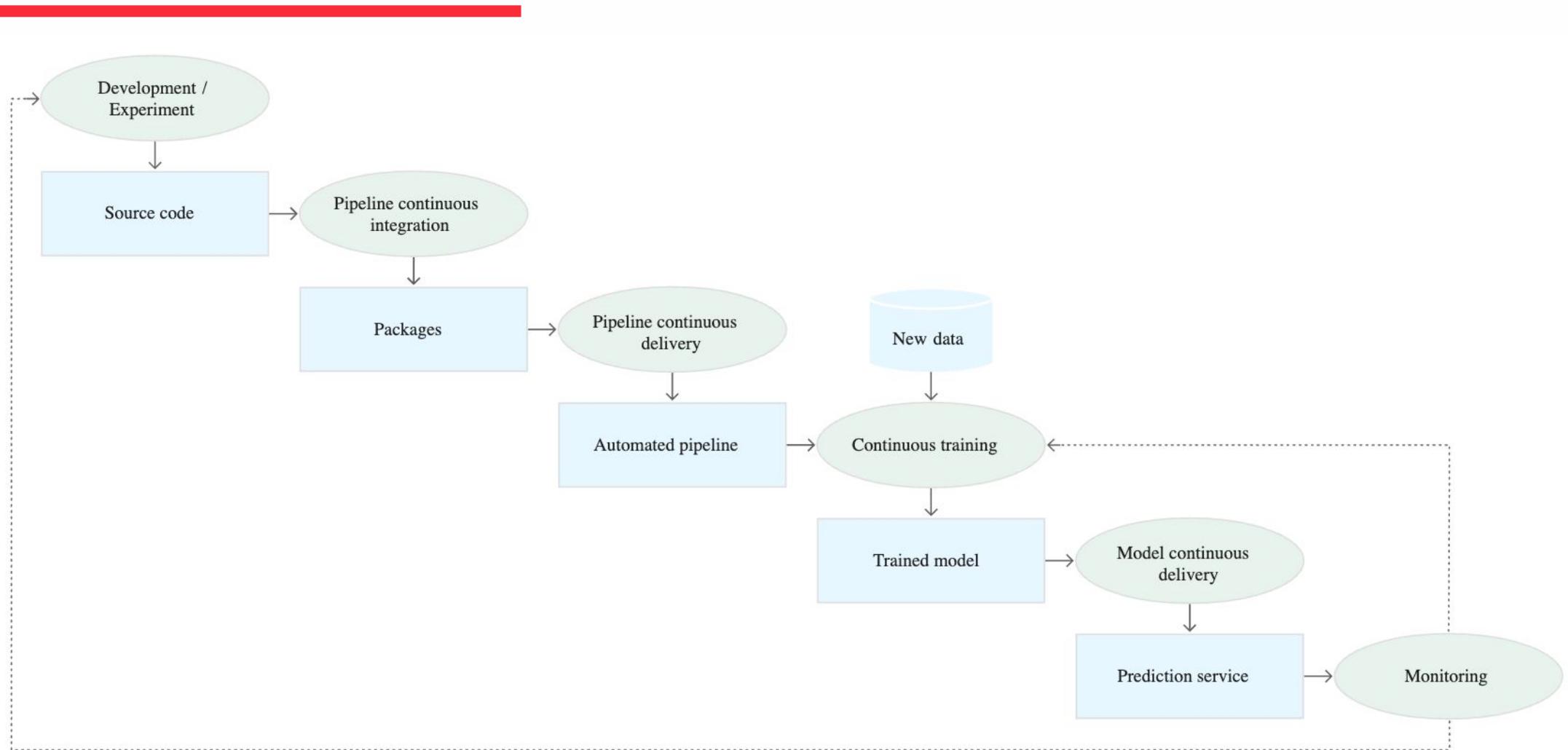
---



# MLOPS: level 2



# Автоматизация помогает двигаться быстрее



# Что поможет реализовать?

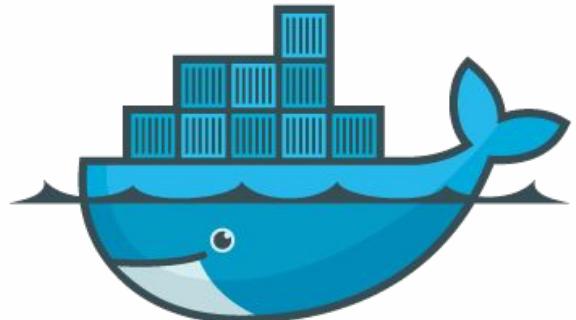
---



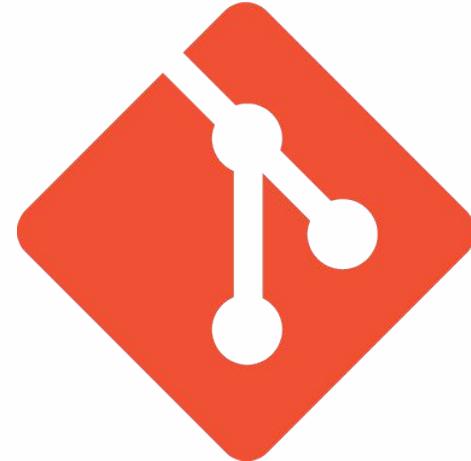
Jenkins



kubernetes



docker

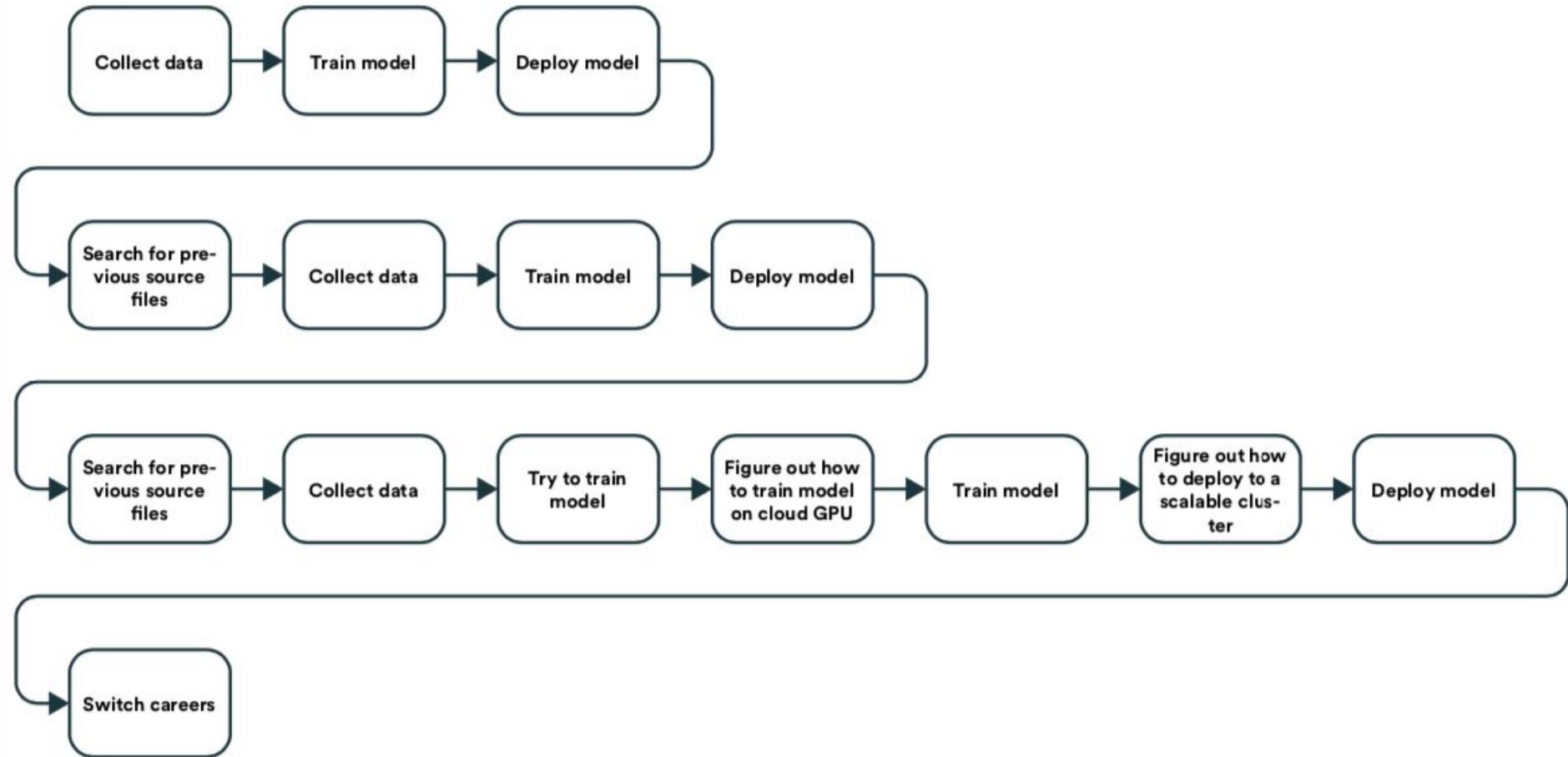


git

# История 2-х компаний

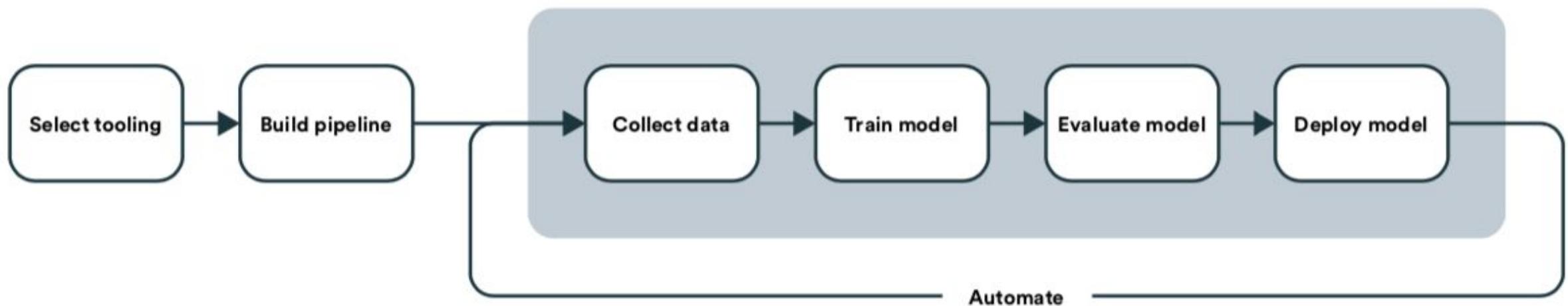
# Company 1

---



# Company 2

---





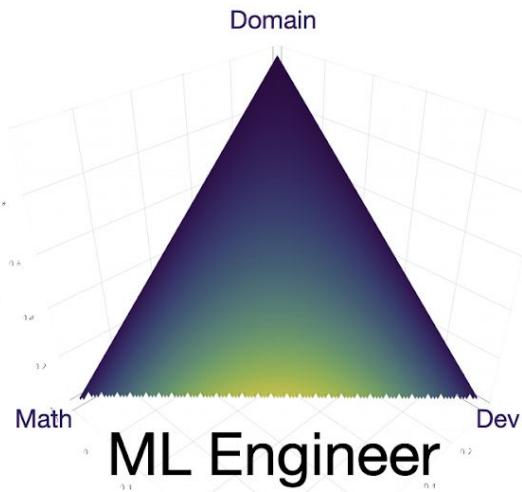
# Итоги

---

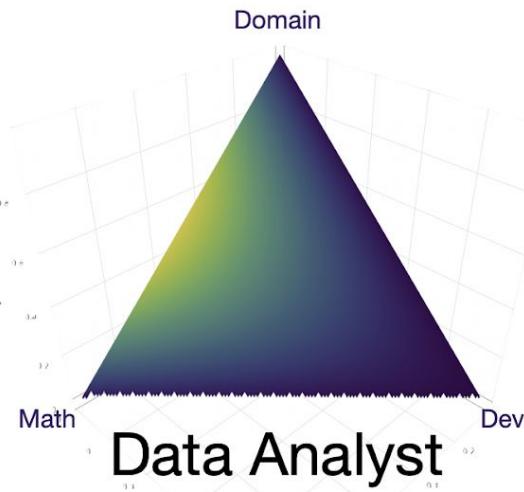
- Двигаться быстро важно
- Но вложиться в инфра/процесс тоже важно

# Роли в ML

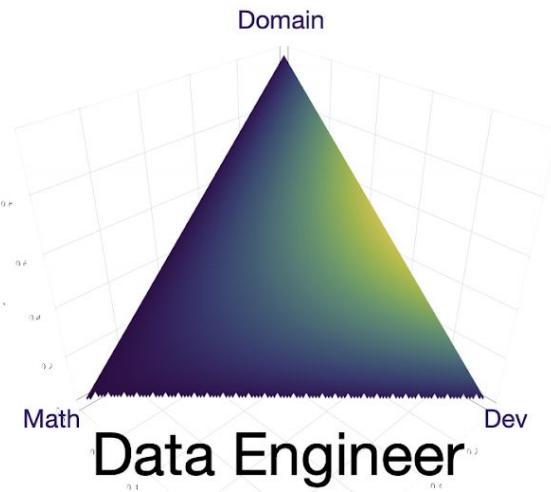
# Роли в ML



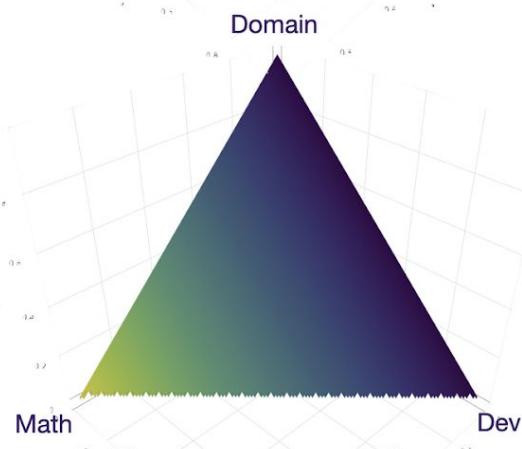
**ML Engineer**



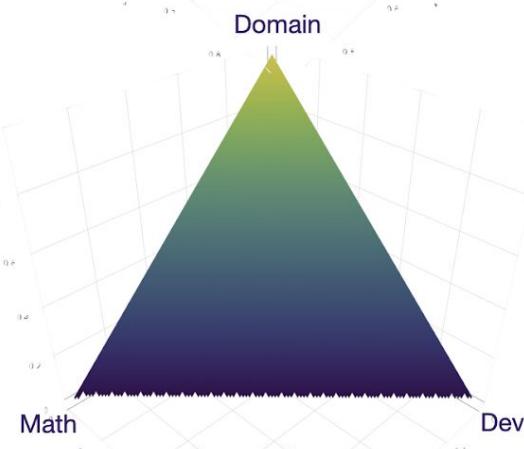
**Data Analyst**



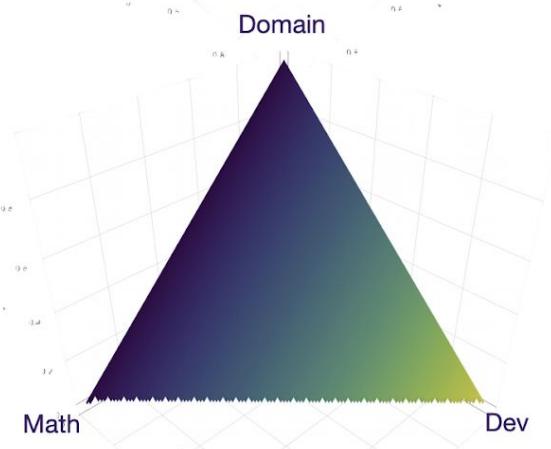
**Data Engineer**



**ML Researcher**

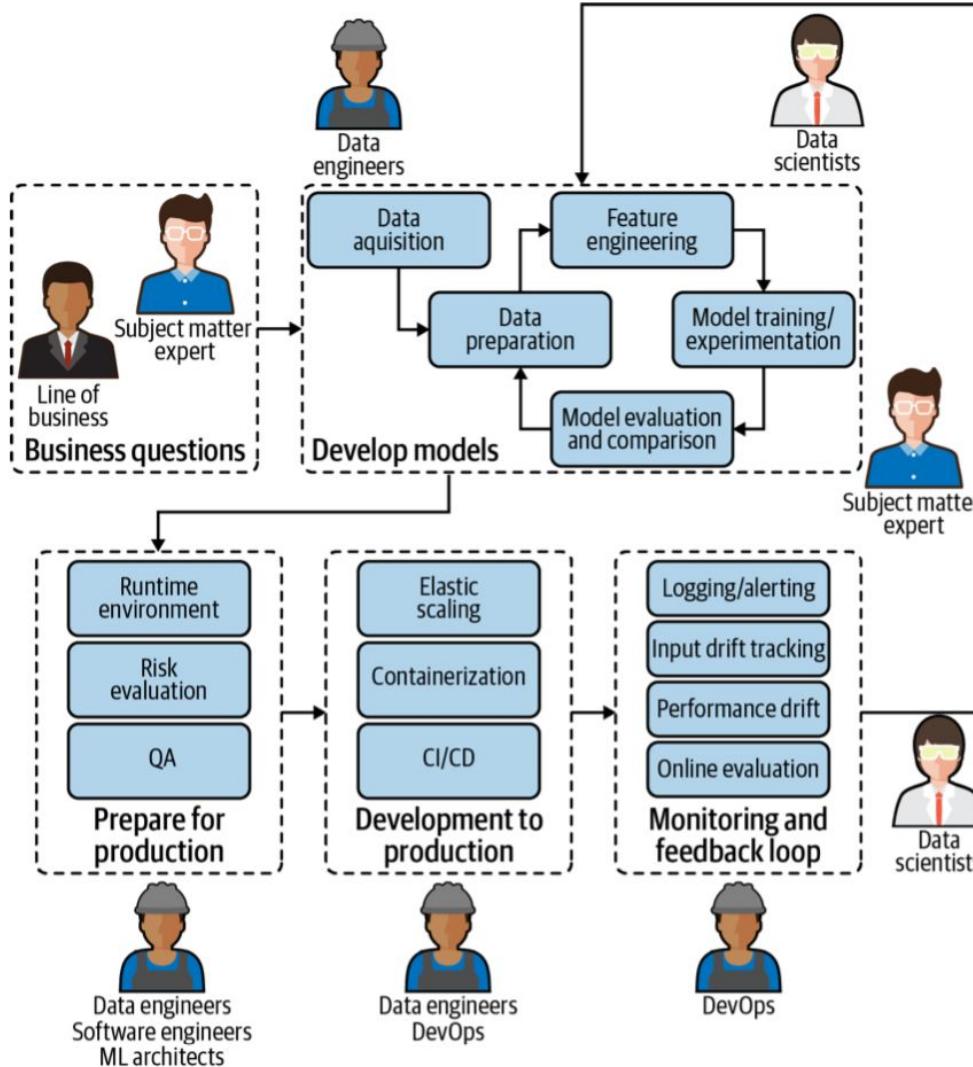


**Analyst**



**Devops**

# Enterprise ML





# Data Scientist/ML Engineer

---

Разработку модели

За то, что модель готова к эксплуатации

Развертывание модели +-

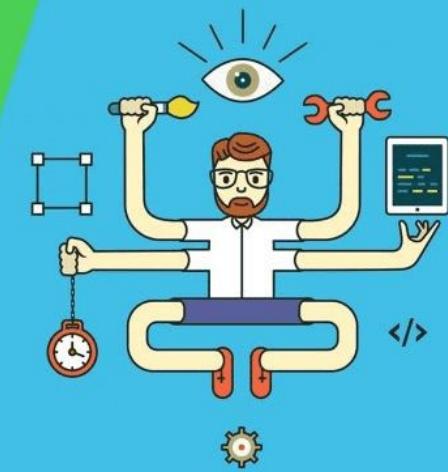
Оценка качества модели онлайн-оффлайн

Итеративное улучшение моделей

# Data Engineer

---

**BIG  
DATA  
ENGINEER**



Создание платформы данных

Разработку конкретных источников для  
моделей

Оптимизацию производительности data  
pipelines

# Software Engineer

---



Встраивание моделек в продукт

# DEVOPS

---

Создание платформы для развертывания и мониторинга приложений ( в том числе и моделей)

Создание CICD пайплайнов для компонентов ML систем





# ИТОГИ

---

1. Люди всякие нужны, люди всякие важны
2. Data Scientist должен иметь возможность доставлять результаты своего труда САМ
3. Переписывать за DS код тренировки/инференса моделей — антипаттерн

# Be T-shaped



Mindset



CI/CD Pipeline



Frontend



Cloud Platform



Backend API



Container



Research



Machine Learning



Math

<https://towardsdatascience.com/t-shaped-skills-builder-guide-in-2020-for-end-to-end-data-scientist-33d2652511b0>