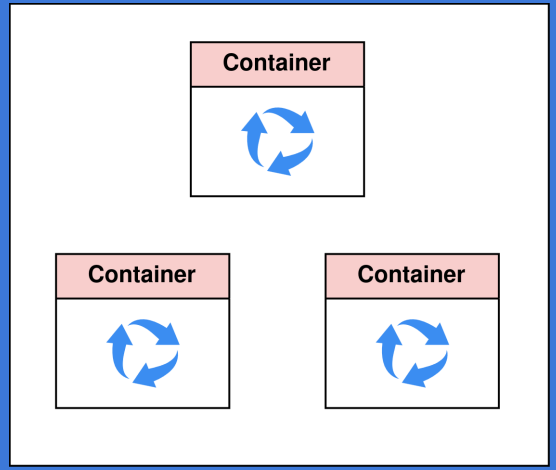
От DevOps к MLSecOps

Максим Рындин, ИСП РАН

Проблемы при ML-разработке

Исследования:

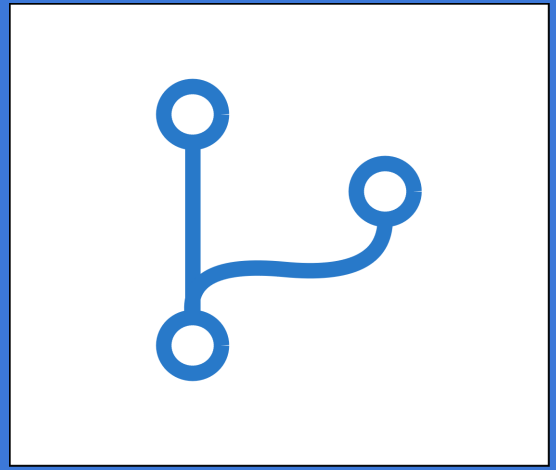
- воспроизводимость и изолированность экспериментов
- версионирование активов
- аналитика результатов
- управление ресурсами
- унификация процессов разработки в команде
- обеспечение доверия



Проблемы при ML-разработке

Исследования:

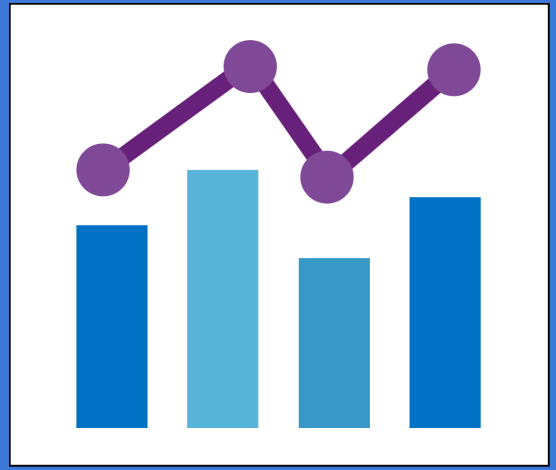
- воспроизводимость и изолированность экспериментов
- версионирование активов
- аналитика результатов
- управление ресурсами
- унификация процессов разработки в команде
- обеспечение доверия



Проблемы при ML-разработке

Исследования:

- воспроизводимость и изолированность экспериментов
- версионирование активов
- аналитика результатов
- управление ресурсами
- унификация процессов разработки в команде
- обеспечение доверия



Проблемы при ML-разработке

Исследования:

- воспроизводимость и изолированность экспериментов
- версионирование активов
- аналитика результатов
- управление ресурсами
- унификация процессов разработки в команде
- обеспечение доверия

Проблемы при ML-разработке

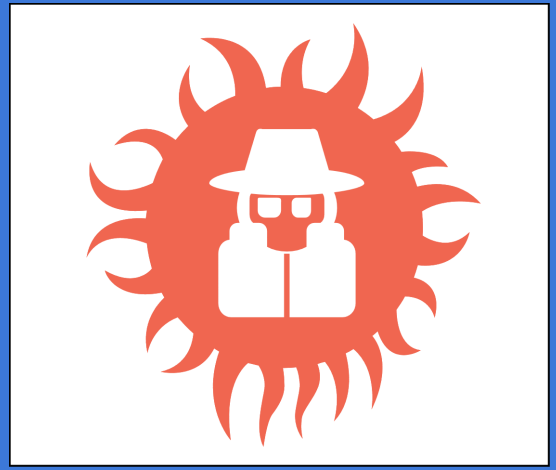
Исследования:

- воспроизводимость и изолированность экспериментов
- версионирование активов
- аналитика результатов
- управление ресурсами
- унификация процессов разработки в команде
- обеспечение доверия

Проблемы при ML-разработке

Исследования:

- воспроизводимость и изолированность экспериментов
- версионирование активов
- аналитика результатов
- управление ресурсами
- унификация процессов разработки в команде
- обеспечение доверия



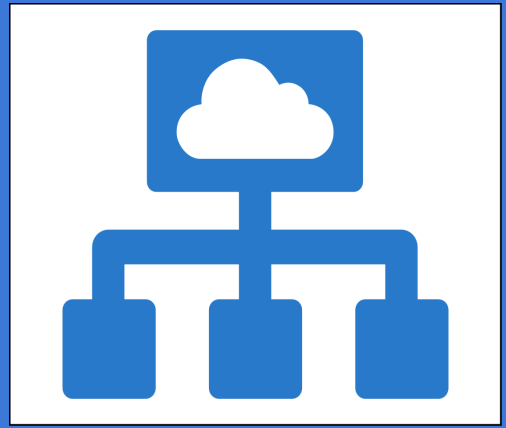
Проблемы при ML-разработке

Исследования:

- воспроизводимость и изолированность экспериментов
- версионирование активов
- аналитика результатов
- управление ресурсами
- унификация процессов разработки в команде
- обеспечение доверия

Исполнение:

- деплой, масштабирование
- обеспечение доверия
- мониторинг устаревания



Проблемы при ML-разработке

Исследования:

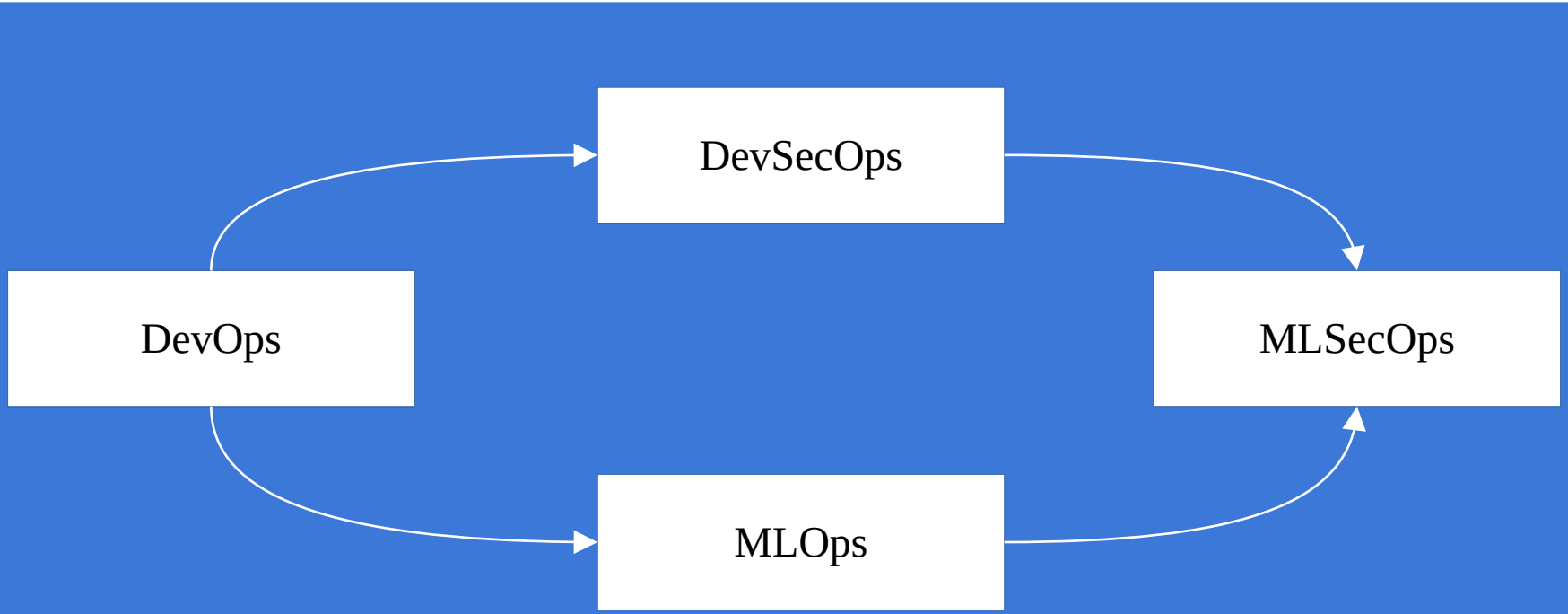
- воспроизводимость и изолированность экспериментов
- версионирование активов
- аналитика результатов
- управление ресурсами
- унификация процессов разработки в команде
- обеспечение доверия

Исполнение:

- деплой, масштабирование
- обеспечение доверия
- мониторинг устаревания

Автоматизация

Методологии



Инструменты

DevOps:

- инструменты для решения отдельных задач и платформы для всего списка задач
- хорошие программные абстракции для операций
- как следствие, вопросы доверия часто можно решить интеграцией специализированных инструментов в CI

Инструменты

DevOps:

- инструменты для решения отдельных задач и платформы для всего списка задач
- хорошие программные абстракции для операций
- как следствие, вопросы доверия часто можно решить интеграцией специализированных инструментов в CI

MLOps:

- инструменты для решения отдельных задач
- почти нет платформ для всего списка задач
- нет хороших программных абстракций, широкий стек

Инструменты

DevOps:

- инструменты для решения отдельных задач и платформы для всего списка задач
- хорошие программные абстракции для операций
- как следствие, вопросы доверия часто можно решить интеграцией специализированных инструментов в CI

MLOps:

- инструменты для решения отдельных задач
- почти нет платформ для всего списка задач
- нет хороших программных абстракций, широкий стек

MLSecOps:

- отдельные библиотеки, которые сложно использовать в пайплайнах, если у вас много моделей на разных стеках
- нет платформ и решений для комплексного решения задач

Задачи по обеспечению доверия

Общие:

- проверка публичных моделей на закладки в коде
- проверка цепочек поставки

Задачи по обеспечению доверия

Общие:

- проверка публичных моделей на закладки в коде
- проверка цепочек поставки

Обучение:

- обучение устойчивых к разному классу атак моделей
- проверка наборов данных на закладки, аномалии, очистка
- проверка моделей на закладки, очистка
- обучение интерпретируемых моделей

Задачи по обеспечению доверия

Общие:

- проверка публичных моделей на закладки в коде
- проверка цепочек поставки

Обучение:

- обучение устойчивых к разному классу атак моделей
- проверка наборов данных на закладки, аномалии, очистка
- проверка моделей на закладки, очистка
- обучение интерпретируемых моделей

Исполнение:

- бенчмарк предобученных моделей
- применение защит на основе пред- и пост- обработки
- противодействие краже, инверсии моделей, определению принадлежности
- противодействие краже данных из модели
- непрерывный мониторинг

Платформа ДИИ

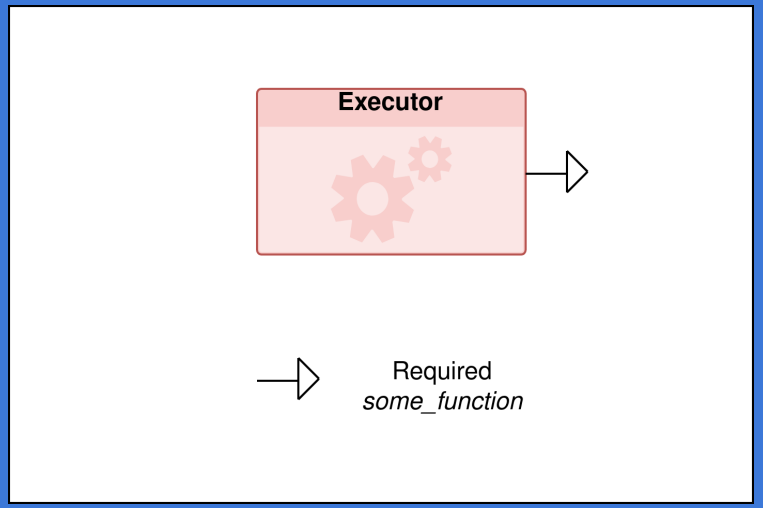
- удобство проведения множества экспериментов в команде
- инструменты сравнения результатов
- облачность
- эффективность внедрения, эксплуатации и поддержки моделей
- поддержка сценариев доверия и встраивание их в существующие процессы

Придумаем абстракции

- подавляющее число экспериментов с моделью стандартны
- train()
  - один цикл тренировки
  - HPO random search
  - HPO gridsearch
- большинство сценариев может быть сгруппированы в “классы”

Executor

- декларирует необходимые интерфейсы
- позволяет:
  - сохранять результаты
  - предоставлять данные модели
  - работать с несколькими моделями

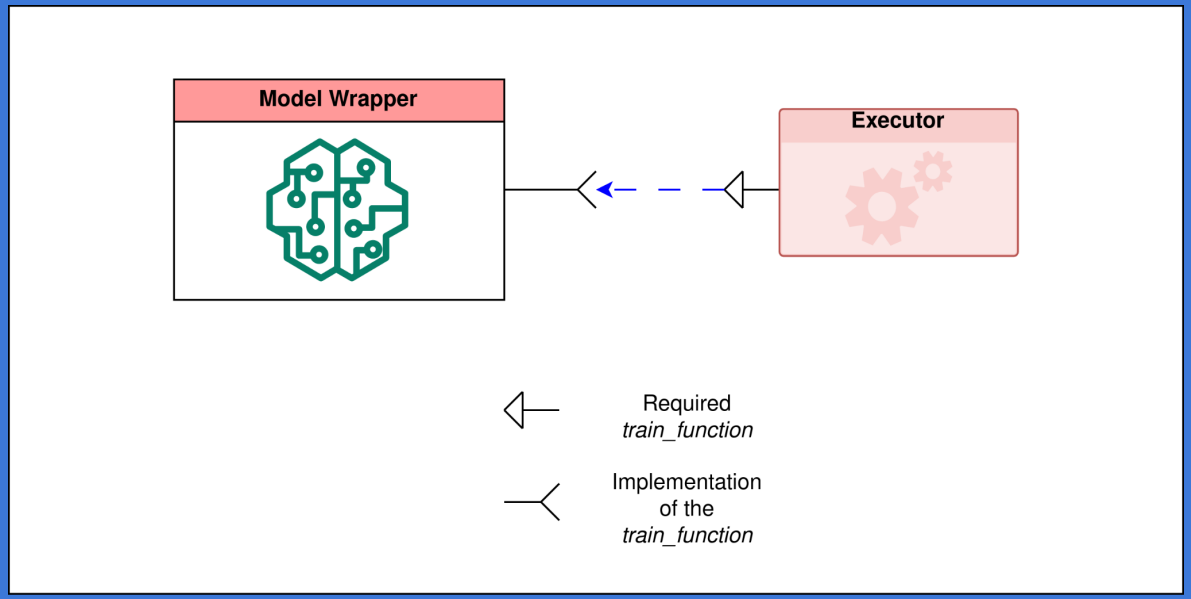


Model

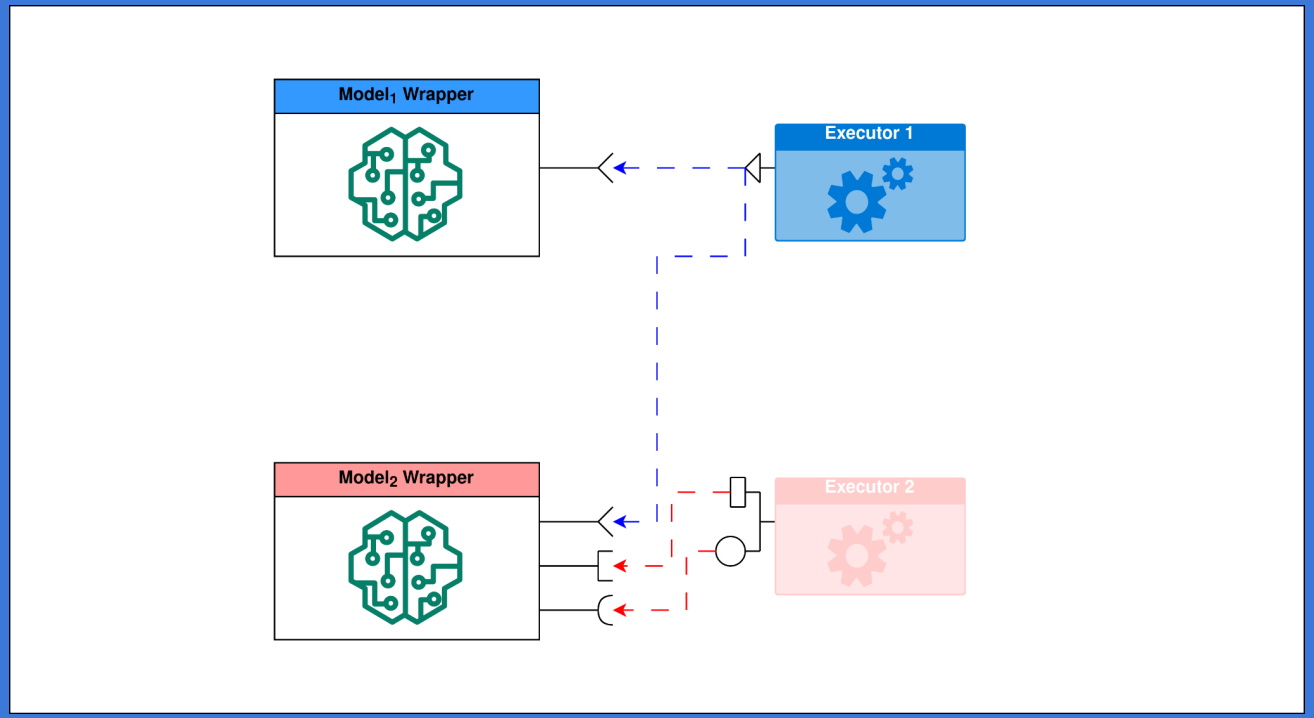
```python
class MyModel(TrainableModel):
    def __init__(self, ...):
        self.model = ...

    def train_function(self, num_epochs: int):
        ...
        for epoch in num_epochs:
            for input_data, target in self.dataset:
                ...
                ... = self.model(input_data)
                ...

            ...
            mlmanagement.log_metric("Accuracy", accuracy)

    def predict_function(self, input_batch):
        ...


MyModel(...).upload_model(registered_model_name='model')
```

- реализуем стандартизированные интерфейсы
- логируем метрики вызовом клиентской библиотеки
- загружаем код на сервер

Executor + Model

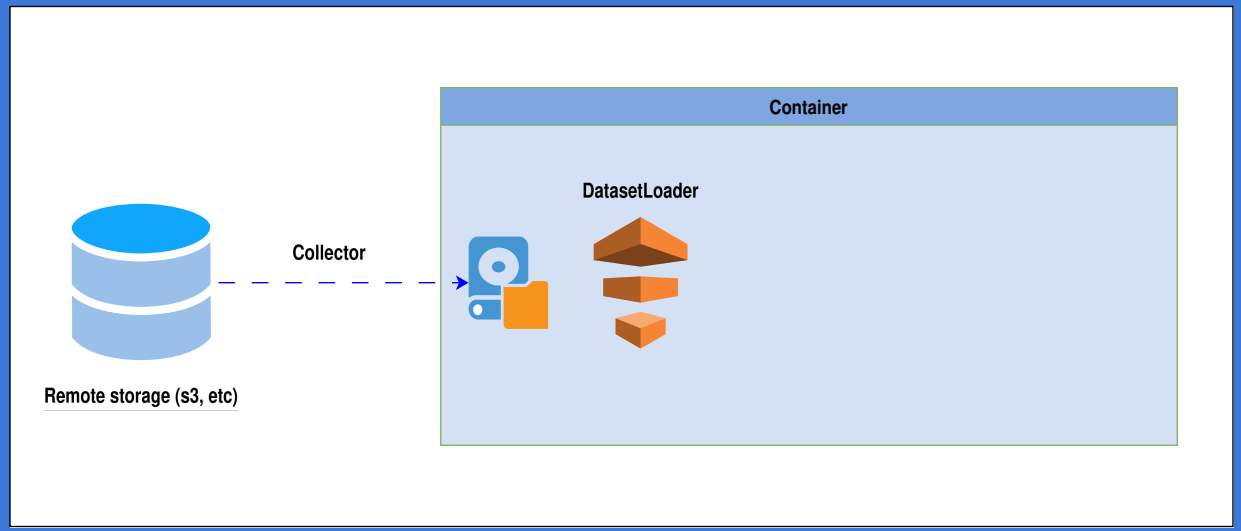


Executor + Model

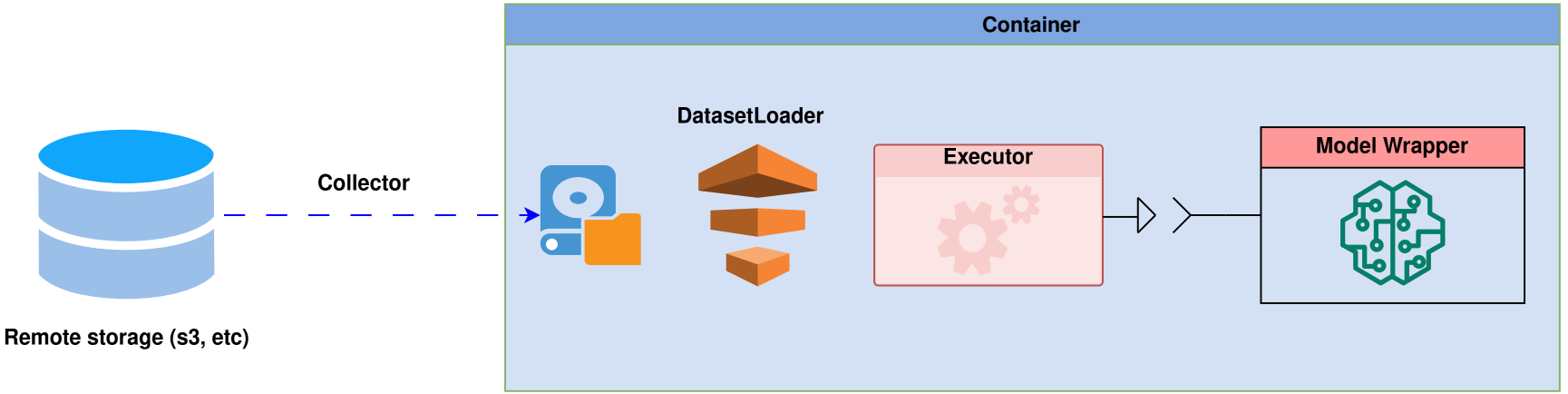


Data

- **Collector:** предоставление сырых данных в контейнер исполнения
- **DatasetLoader:** чтение сырых данных в формат, необходимый для модели



Pipeline


Container
DatasetLoader
Executor
Model Wrapper
Collector
Remote storage (s3, etc)

Inference

```
from ML_management.mlmanagement import load_model,
set_server_url

set_server_url("https://mlm.ispras.ru")
model = load_model("NEW_DEMO_MODEL", version=1)
model.predict_function(input_batch)
```

```
from ML_management.sdk import serve_model

model_url = serve_model(name, version, gpu)
```

- можно использовать полученную модель локально
- можно поднять Triton Inference Server для serving-a

Итоги

- облачная платформа
- версионирование моделей и данных
- хранение информации об экспериментах
- эффективное переиспользование кода
- поддержка сценариев доверия, непрерывного обучения и мониторинга
- простой serving

Демонстрация доступна на выставке!