2021

От одной большой ETL-джобы до процесса проведения экспериментов над дата-пайплайнами

Спикер: Артем Юдовин

Агенда

- 🙋 Знакомство
- 🪄 Зачем нужны эксперименты в pipeline?
- 🖐️ 5 шагов к экспериментам
- 🤟 Демо: три эксперимента

Привет!

Я Артем Юдовин

twitter / GitHub — @ayudovin

5+ лет работаю в Big Data направлении. Люблю open-source проекты. Tech Lead команды data engineers в продуктовой компании Profitero.

Кто мы?

Profitero — продуктовая IT-компания, разрабатывающая один из ведущих мировых сервисов e-commerce аналитики. Помогаем различным брендам продавать онлайн: собираем информацию у ритейлеров, анализируем и выдаём в понятном виде.

2010
Год основания

4000+
Брендов

600+
Ритейлеров

Контекст

1

Постоянно необходимо улучшать методики обработки данных

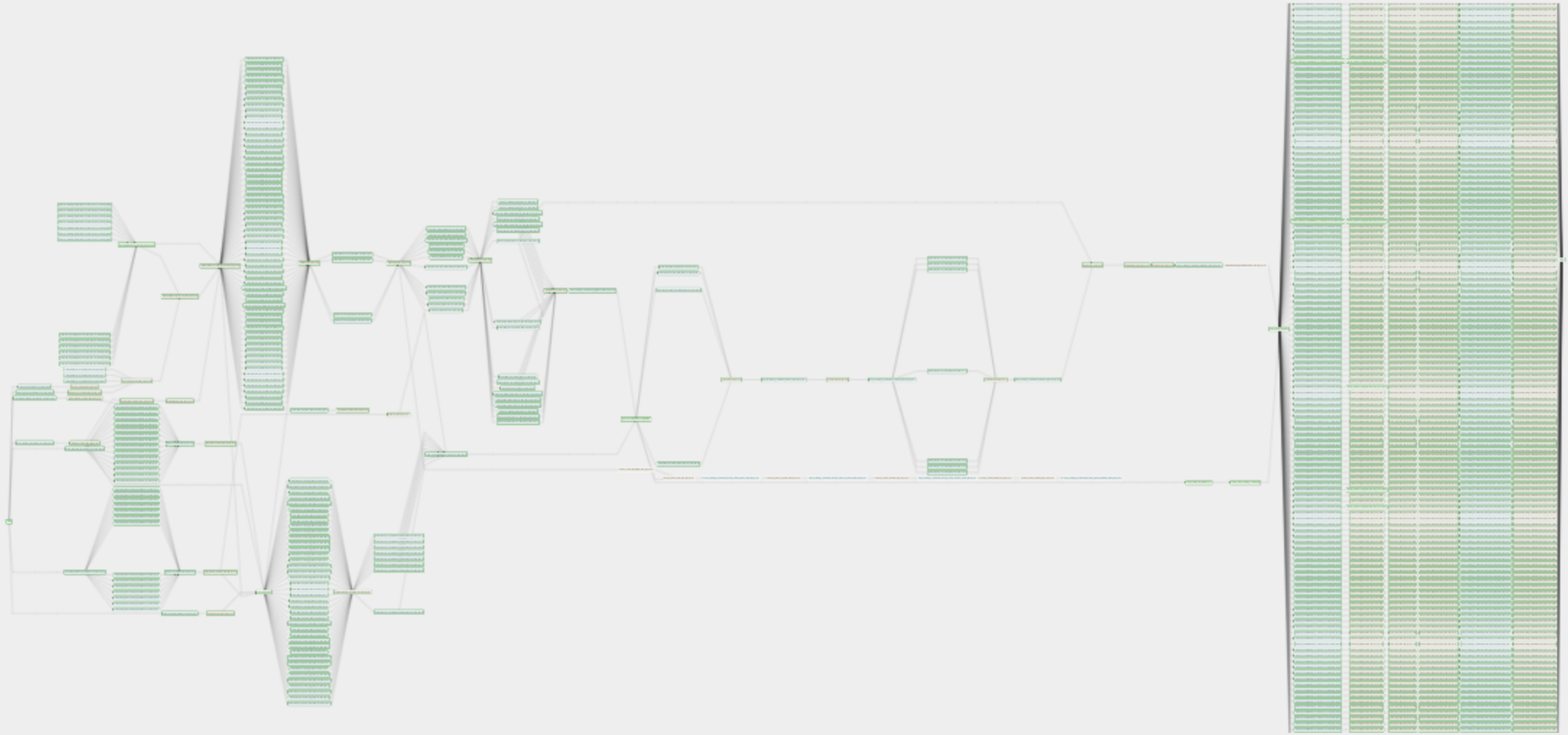
2

Многие решение принимаются не на основе метрик качества

3

Всегда есть риск ухудшить данные клиента

Главная проблема



Концепция

🪄 Эксперименты

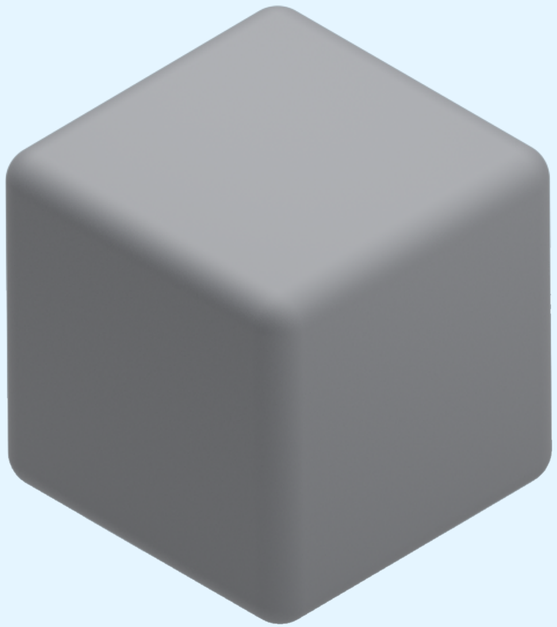
— это изменение методики обработки данных и проверка финальных результатов с расчетом метрик качества.

Эксперименты

1. Как организовать процесс проведения экспериментов?
2. Как сделать этот процесс простым?
3. Как проводить больше одного эксперимента?

С чего все начиналось

Есть входные данные



Есть выходные данные

⏳Время

Сложности

- Сложно применять какие-либо изменения
- Сложно предсказать влияние этих изменений на финальный импакт
- Сложно разобраться, что происходит внутри
- Нет контроля над происходящим
- Нет визуализации

Staging

Диктуем свои условия

1. Готовим свое окружение

2. Разворачиваем на нем pipeline

3. Применяем изменения

4. Смотрим на результат

Недостатки

Не все так гладко

Один эксперимент за раз

Неконтролируемый процесс (только входные и выходные данные)

Нет визуализации

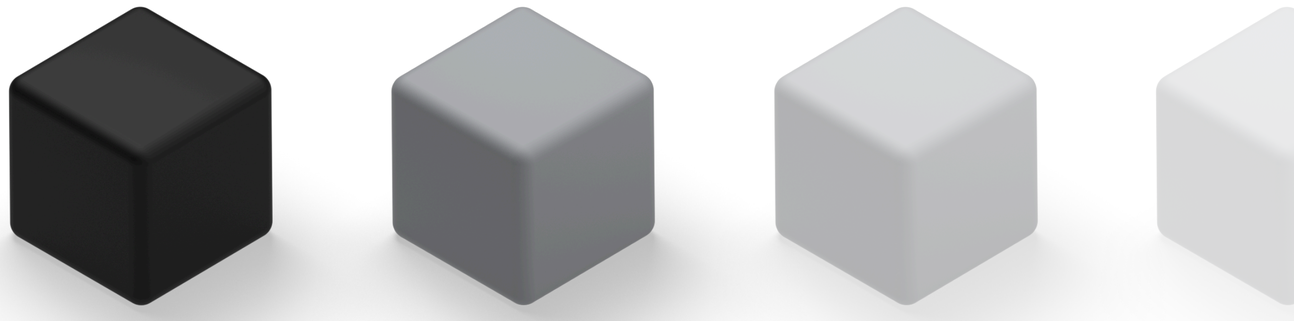
Долго ждать результатов

Долго генерировать новую версию для эксперимента

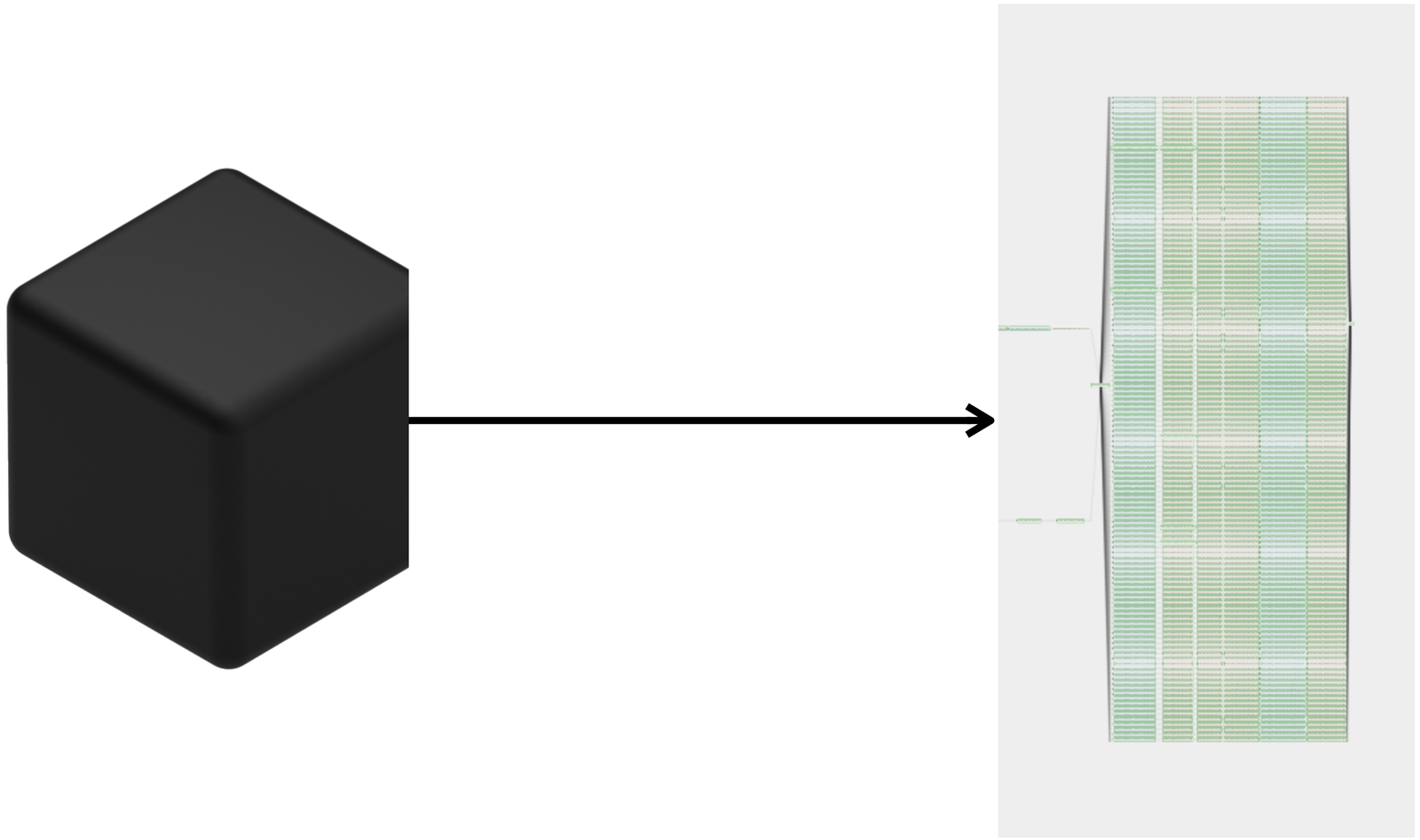
Неудобно сравнивать эксперименты между собой

Нельзя пересчитать только измененную ветку

Интерактивность

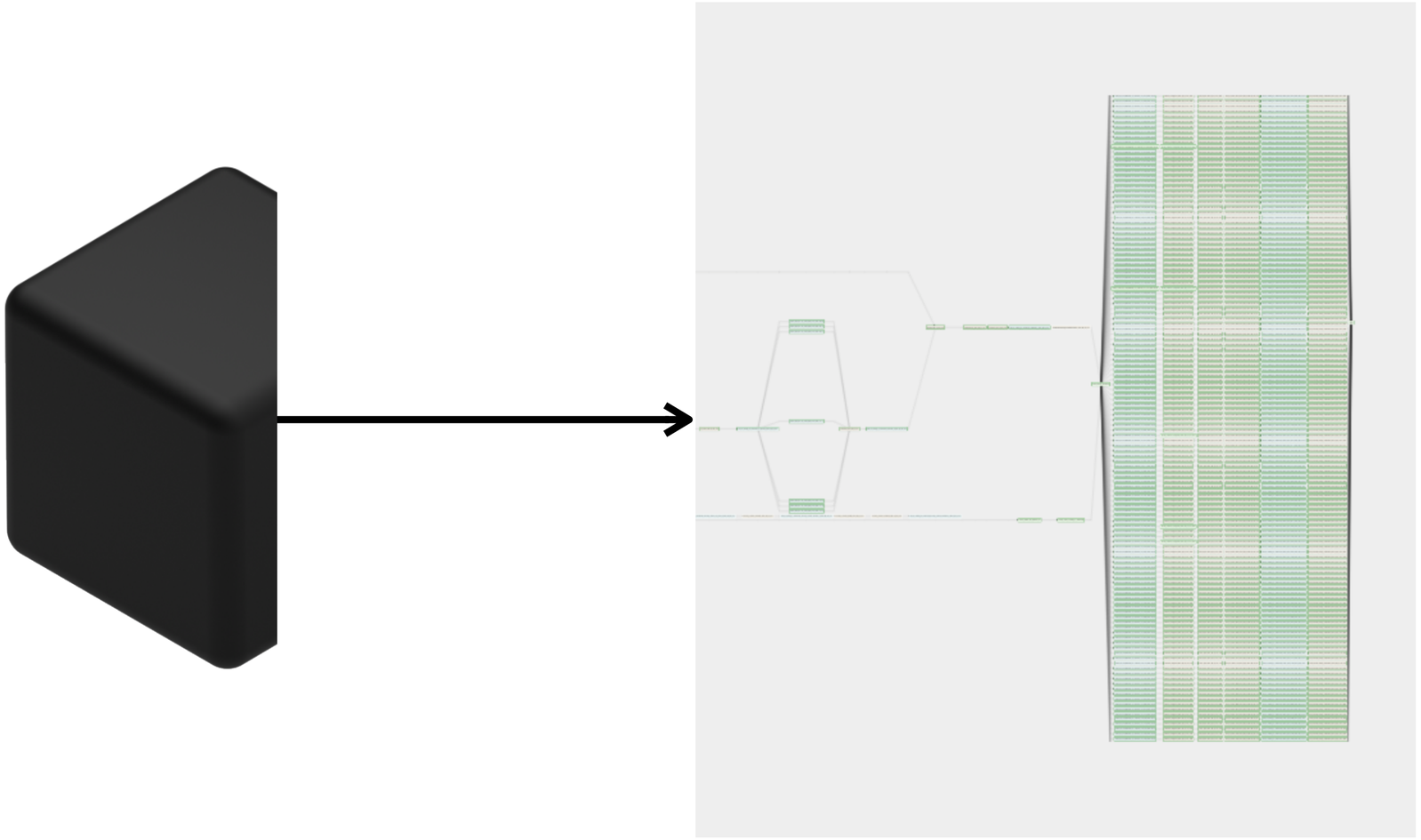


Больше прозрачности и контроля!



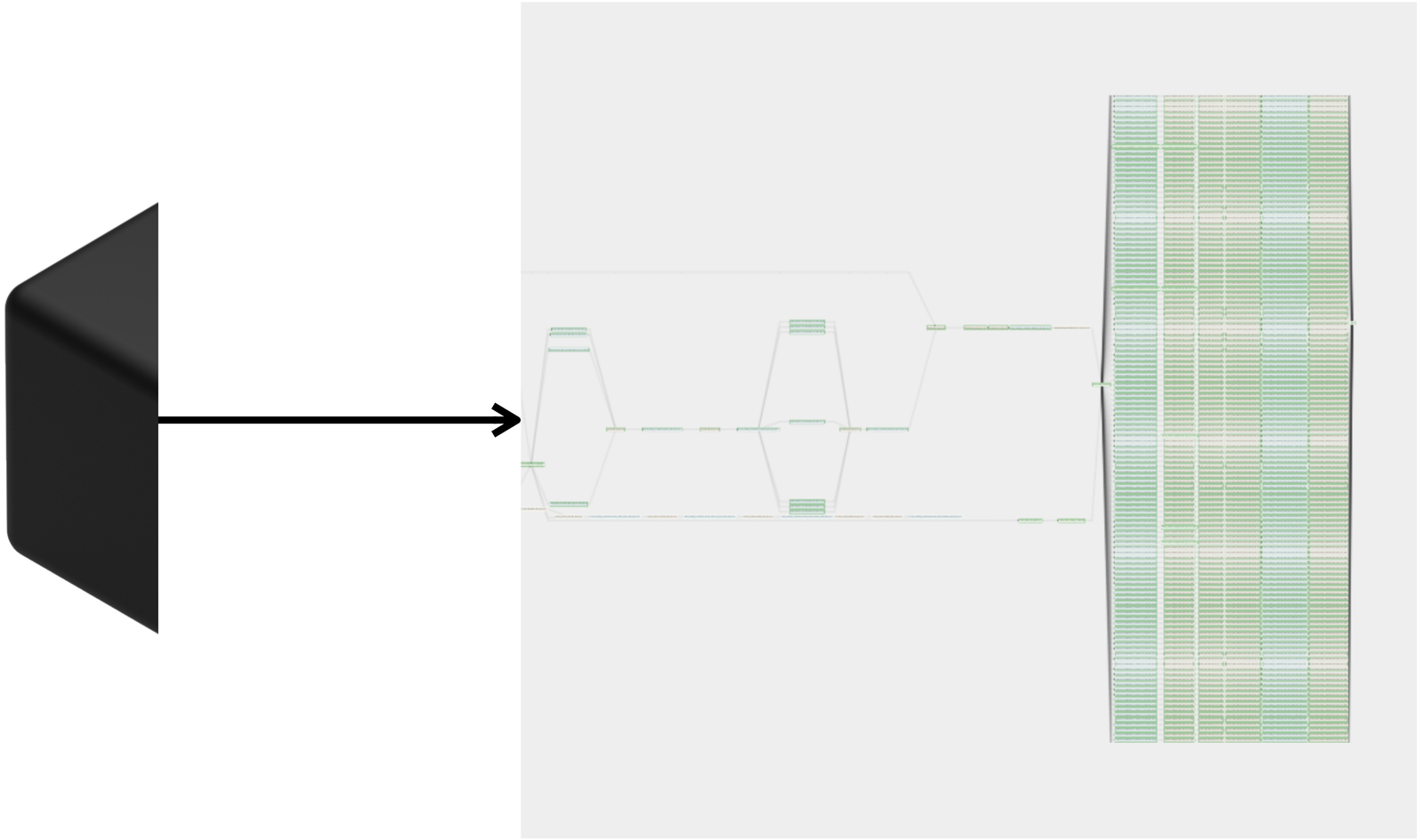
Еще!

Apache Airflow



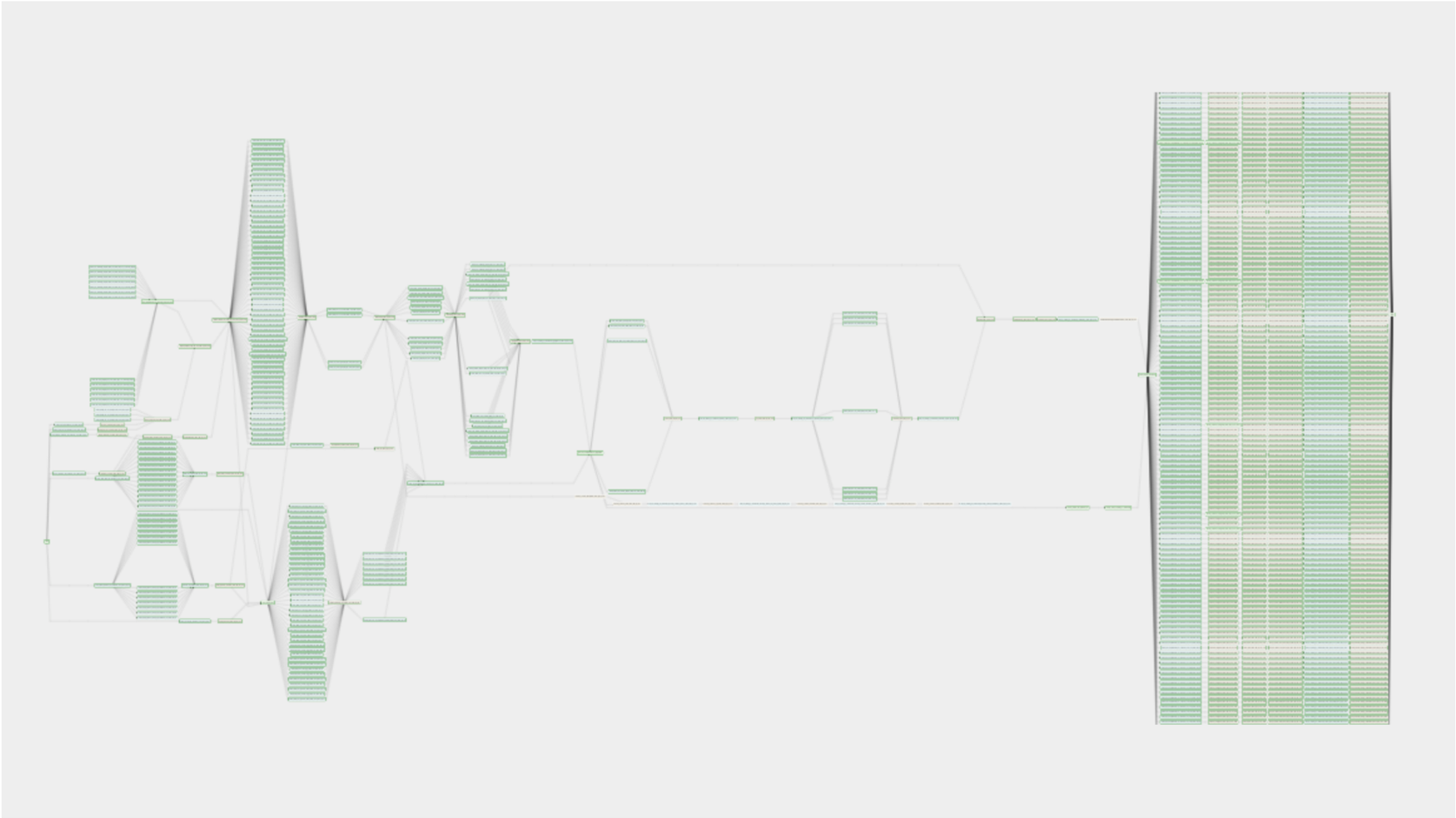
Еще!

Apache Airflow



Идеально!

Apache Airflow



Интерактивность

Спасибо тебе!

- Визуализация
- Контроль над pipeline
- Можно пересчитывать только измененную ветку

Недостатки

Не все так гладко

Один эксперимент за раз

~~Неконтролируемый процесс (только входные и выходные данные)~~

~~Нет визуализации~~

~~Долго ждать результатов~~

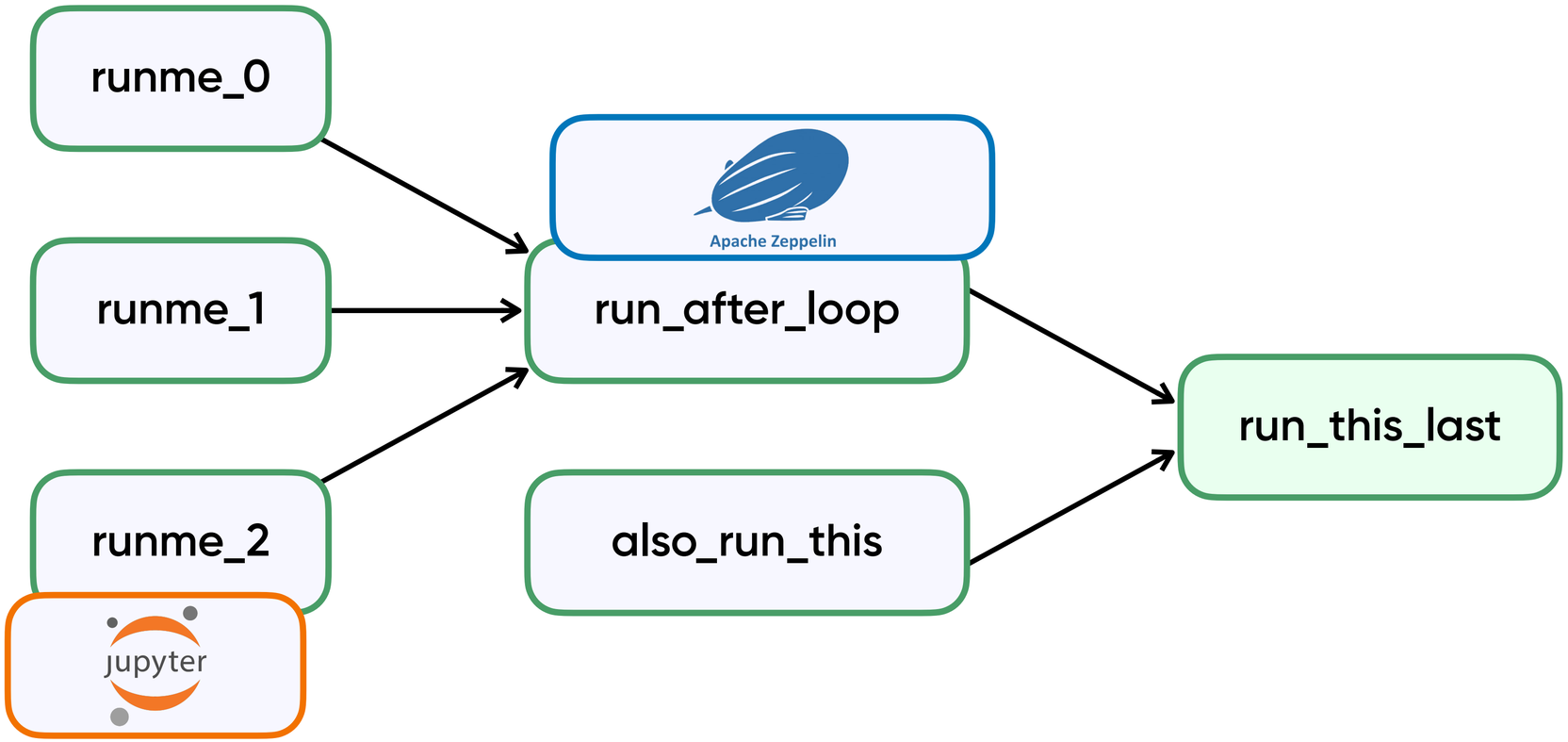
Долго генерировать новую версию для эксперимента

Неудобно сравнивать эксперименты между собой

~~Нельзя пересчитать только измененную ветку~~

Подлог

1. Генерируем нашу новую snapshot версию джобы
2. Кладем версию в Nexus
3. Обновляем версию в конфигурации
4. Перезапускаем нужный бранч


runme_0
runme_1
runme_2
jupyter
Apache Zeppelin
run_after_loop
also_run_this
run_this_last

Подлог

Ввели в заблуждение

jupyter

Apache Zeppelin

- Быстро подменяем существующий код
- Можем заменить полностью оператор на то, что в ноутбуке
- Можем заменить только часть методики

Недостатки

Не все так гладко

Один эксперимент за раз

Увеличиваются требования к коду?

Должна быть возможность подтянуть код как библиотеку?

Неудобно сравнивать эксперименты между собой

Изоляция

Эксперимент 1

Эксперимент 2

Эксперимент 3

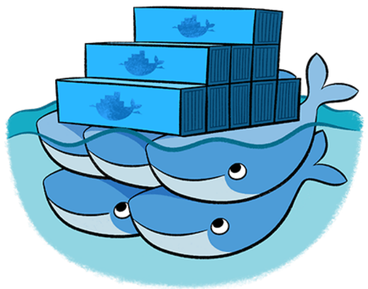
1. Часто идей больше, чем одна

2. Не знаем, какую идею лучше отправить в production

3. Нужно ли дорабатывать идею?

4. А можно посчитать импакт нескольких идей и показать клиенту, чтобы он выбрал что лучше?

Контейнезируй и оркестрируй!



Staging hardware

isolated environment

isolated environment

isolated environment

isolated environment

isolated environment

Больше, чем одна идея!

Мы все можем! Ну, почти.

- Мы можем производить больше одного эксперимента
- Можем проводить различные POCs
- Можем использовать для тестирования
- Можем предоставить необходимый инвайромент для outside staff

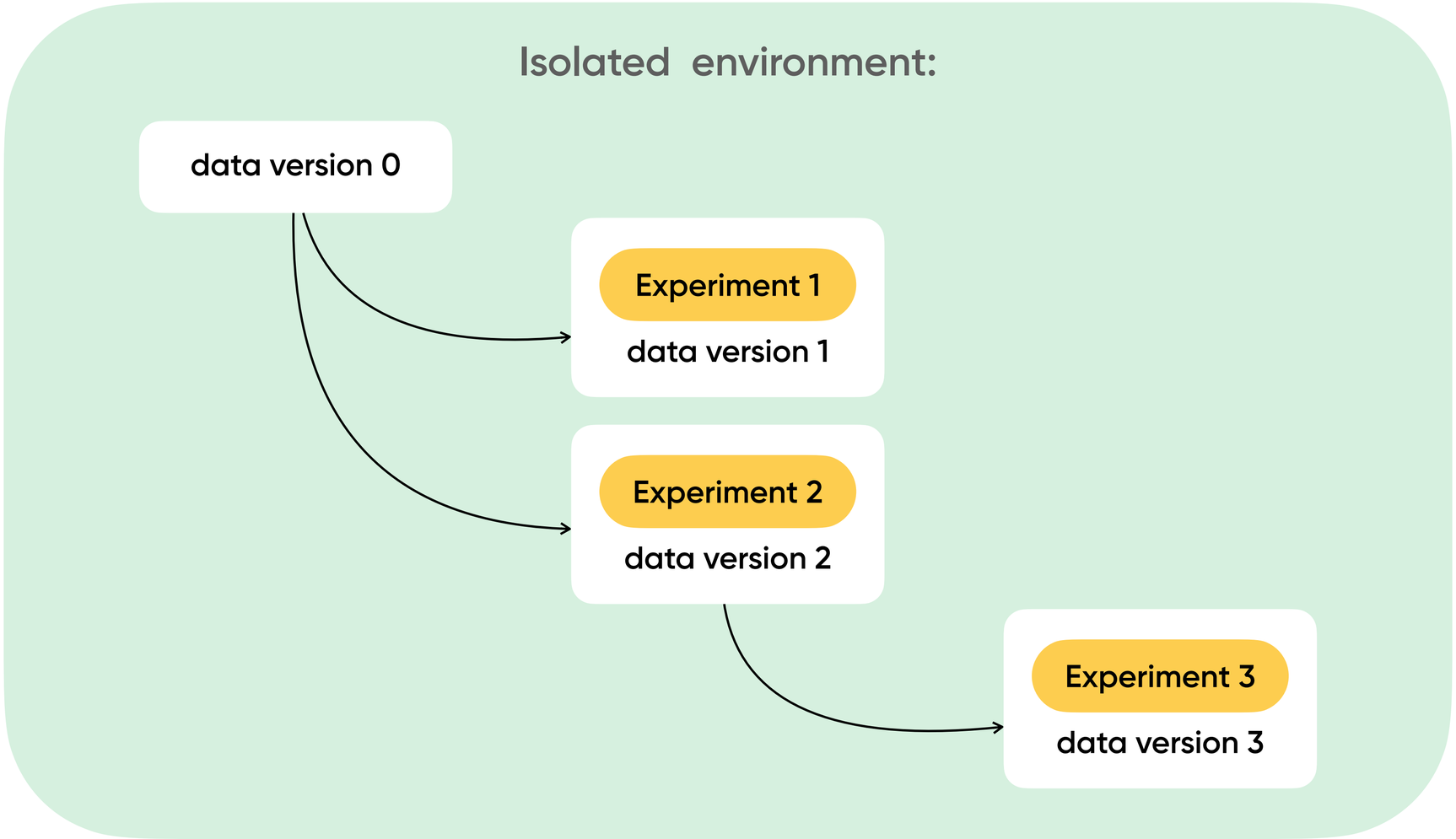
Недостатки

Не все так гладко

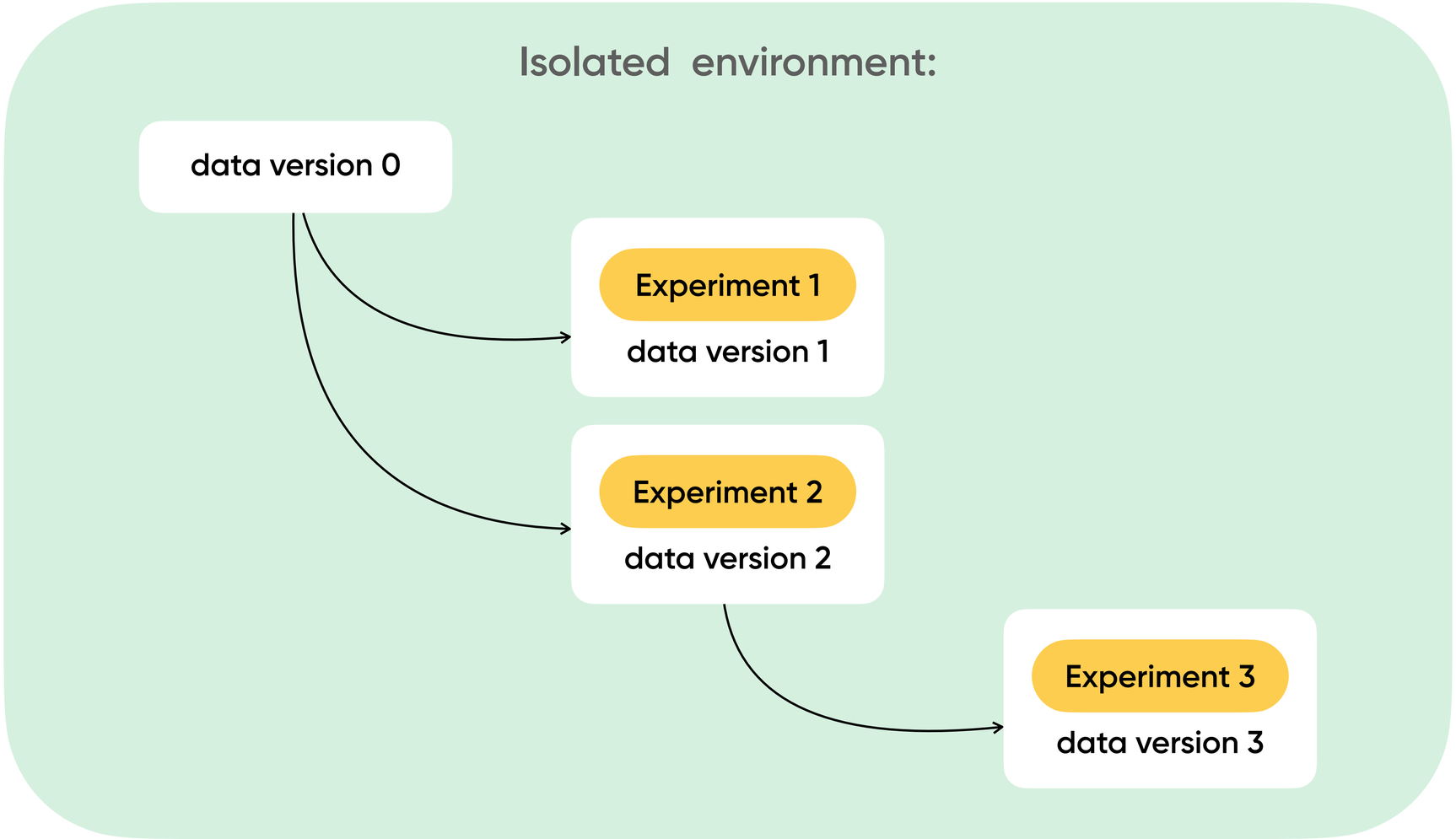
Как правило, один эксперимент
— один инвайромент

Неудобно сравнивать
эксперименты между собой

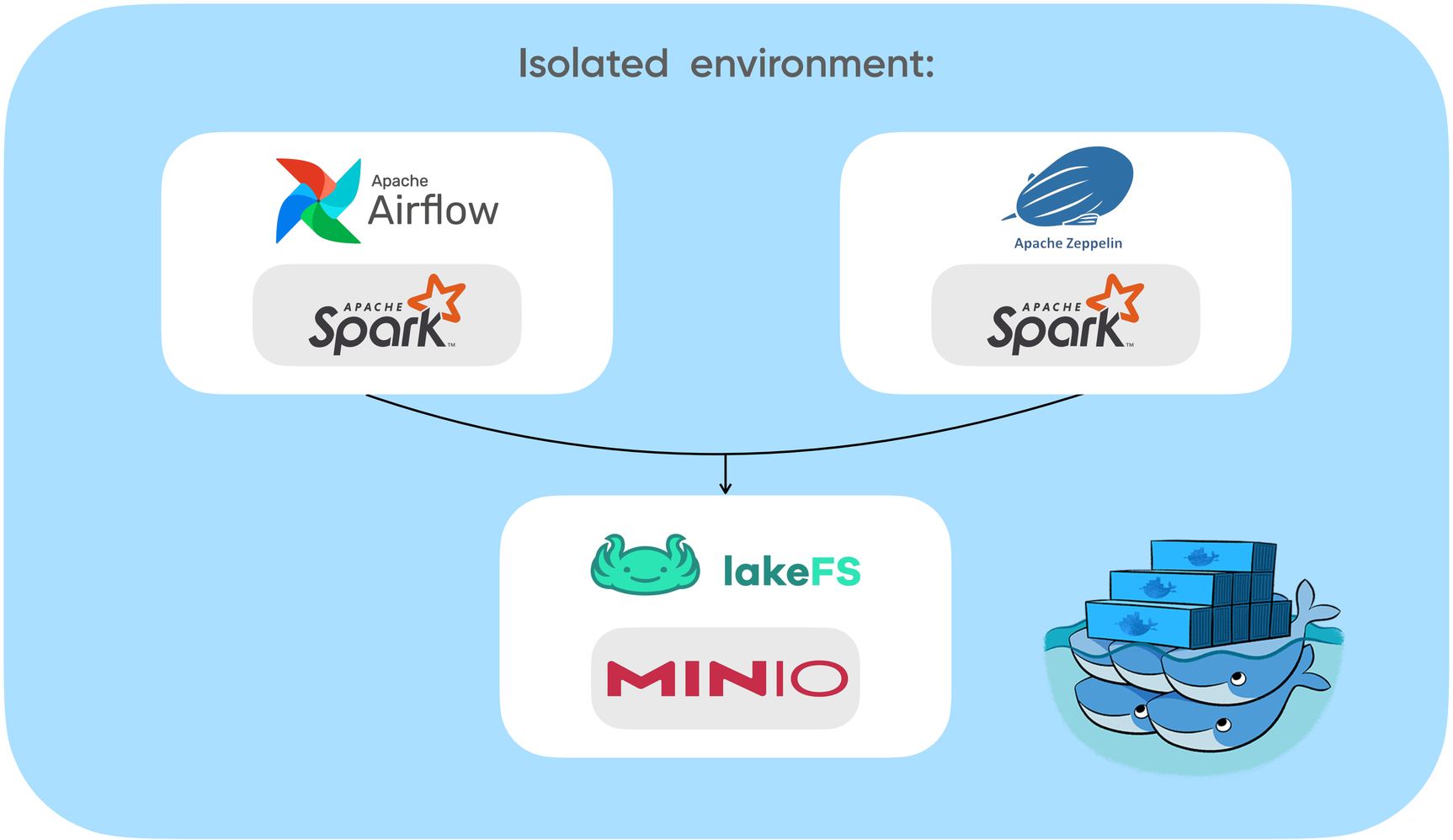
Версионирование данных



Как это может нам помочь?



Работай с данными, как git с твоим репозиторием



Версионирование данных

Git — наше все!

- На одном инвайроменте мы можем производить больше одного эксперимента

- Можем удобно анализировать и сравнивать результаты между собой

Недостатки

Не все так гладко

Зависит от выбранной технологии 🤷

Результат


Apache
Airflow
Apache Spark
Apache Zeppelin
Apache Spark
lakeFS
MINIO

- Почему так долго разворачивать изолированное окружение?
- А можно развернуть +100500 изолированных окружений?
- А если я не хочу разворачивать все сервисы в изолированном окружении?

Demo

https://github.com/ayudovin/data-pipeline-experiments

Что мы будем делать?

- Попробуем предсказать количество проданных продуктов на основе его ранга

- Попробуем улучшить MAE различными изменениями (можно вставить формулу?)

☝️

Поднимем все необходимое окружение:

Minio, LakeFS, Apache Zeppelin, Apache Airflow, Apache Spark, Docker

✌️

Проведем несколько экспериментов:

1. Новая версия

2. Переопределим метод в ноутбуке

3. Заменить оператор на ноутбук

🤟

Сравним результаты

Всем спасибо!

💙 В главных ролях:

Minio
https://min.io/

LakeFS
https://lakefs.io/

Jupyter
https://jupyter.org/

Docker
https://www.docker.com/

Apache Airflow
https://airflow.apache.org/

Apache Spark
https://spark.apache.org/

Apache Zeppelin
https://zeppelin.apache.org/