СЛЁРМ Обучение для инженеров и технических лидеров

Дата-инженер

Проблемы и решения при построении архитектуры

Николай Марков

Цели урока. Что вы узнаете:

1. Как не попасть в болото и как из него выбраться
2. Как найти, что где лежит
3. Немного про MLOps и зачем он нужен
4. Занудные советы, очевидные и не очень

Как выбраться из болота

Помните этот слайд про Data Lake?

- Все данные компании в сыром виде агрегируются в **одно общее место**
- В некоторых компаниях в качестве Data Lake используют **структурированные БД**, но это, скорее, исключение
- Обычно под Data Lake понимается **объектное хранилище**, которое хранит данные просто в виде **бинарных блобов**
- С помощью использования специализированных форматов (**Parquet**, **ORC** etc.) можно получить **ускорение запросов** поверх данных в озере

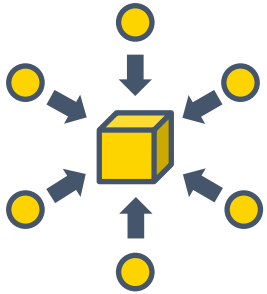
Выделим важные моменты

- Все данные компании в сыром виде агрегируются в **одно общее место**
- В некоторых компаниях в качестве Data Lake используют **структурированные БД**, но это, скорее, исключение
- Обычно под Data Lake понимается **объектное хранилище**, которое хранит данные просто в виде **бинарных блобов**
- С помощью использования специализированных форматов (**Parquet**, **ORC** etc.) можно получить **ускорение запросов** поверх данных в озере

Есть риск получить Data Swamp

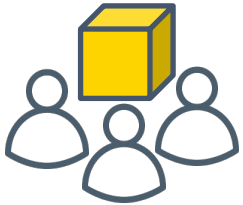
- **Одно общее место** → все поддерживает одна команда и зашивается с этим
- **Бинарные блобы** → куча файлов с какими-то колонками, и никто не знает, с какими
- **Структурированные БД** → нам точно никогда не понадобится хранить картинки, видео и pdf. Зуб даем!
- **Ускорение запросов** → кроме даталейка нам ничего не нужно, обойдемся и так

Одно общее место



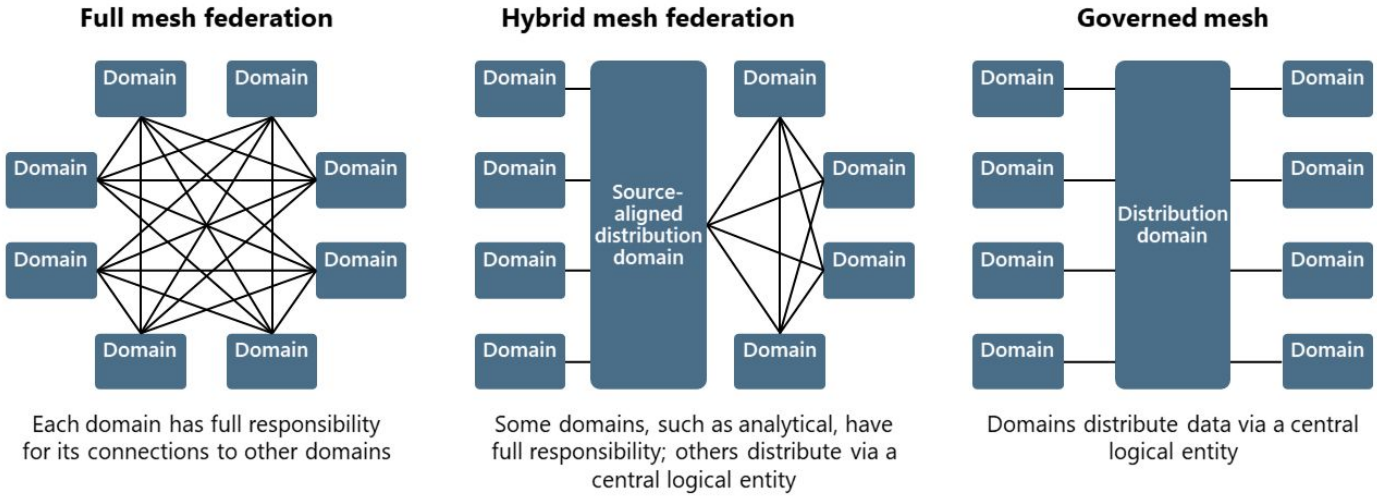
Одно центральное место
! = одна команда поддержки

Мы здесь собрались анализировать данные, а не просто собирать



Данные отдела – это его актив внутри компании

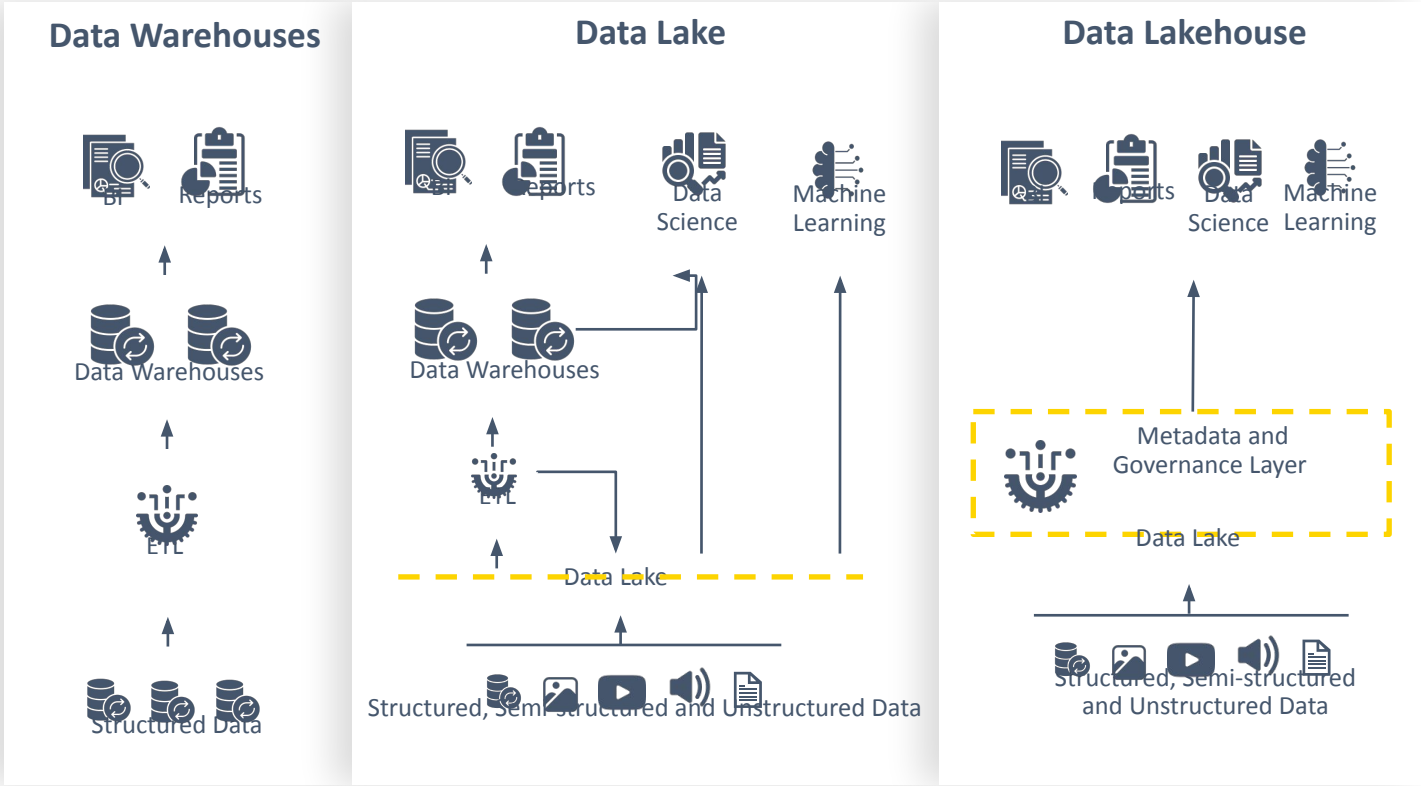
Потенциальное решение — Data Mesh



Бинарные блобы и ускорение запросов

- Построение современных структурированных хранилищ часто требует применения **методологии разделения на бизнес-объекты (Snowflake, Data Vault, Anchor)**
- Это влечет за собой необходимость продвинутой **поддержки JOIN'ов** (объединений датасетов по ключам) и прочих **оптимизаций**, например, **вакууминга**
- Объектные хранилища специально построены так, чтобы не смотреть в содержимое файлов, поэтому как объединения, так и индексирование **становятся сложными и неэффективными в реализации**
- А что делать со **Stream Processing**?

Потенциальное решение — Data Lakehouse



Гибридные решения

Как найти,
что где лежит

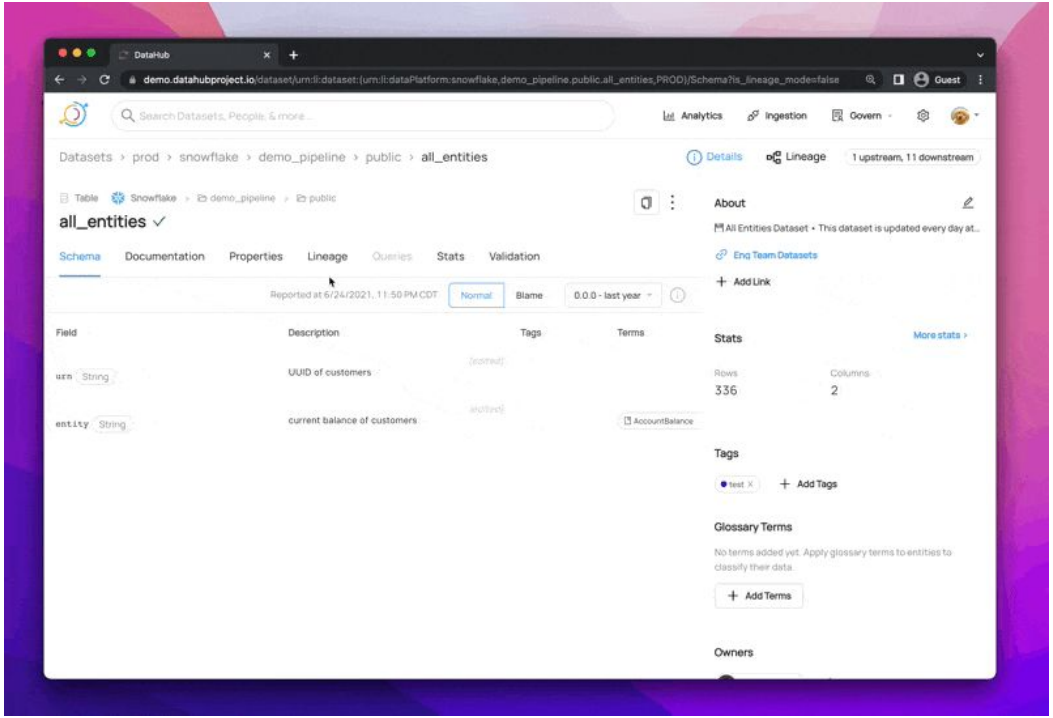
Knowledge Sharing

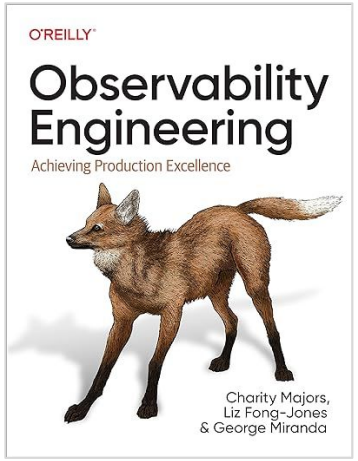
В компаниях часто есть Confluence или что-то еще Wiki-образное

Каждая команда ведет доки самостоятельно

На вопрос «из какого датасета какие колонки взять» ответ часто «эээ…»

Data Observability & Data Catalogs

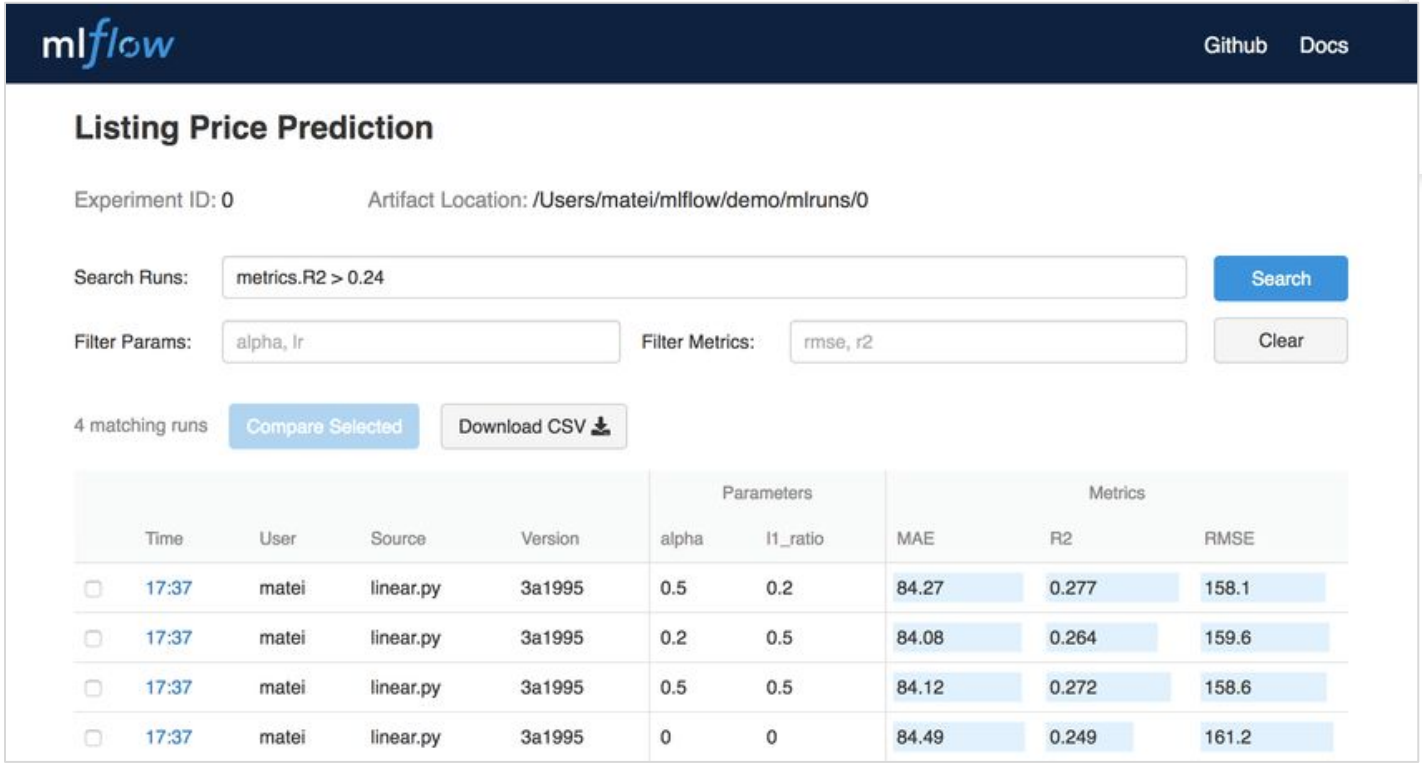




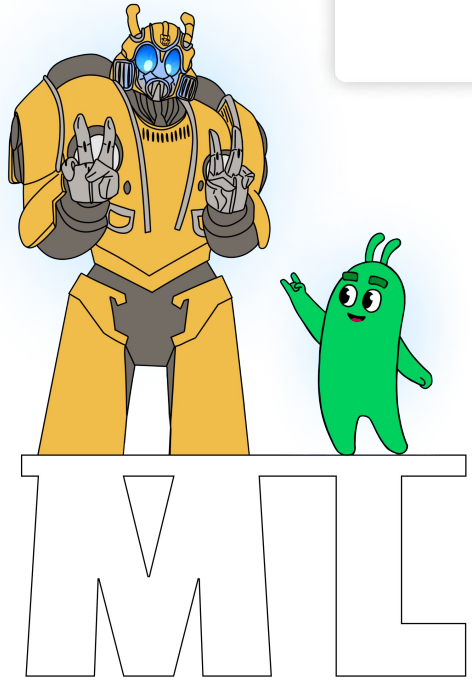
[Reading on Data Observability]

Data Lineage

История экспериментов



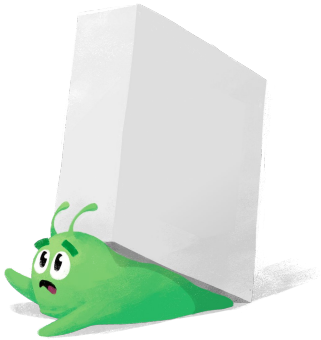
Немного про MLOps



Методология разработки моделей

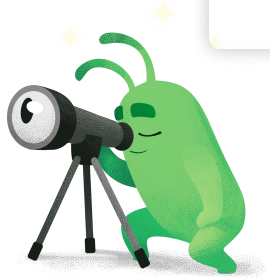
- DevOps – это **не человек** и **не профессия**, это **набор методологий**, направленных на **быстрый выпуск** продукта
- Data Science – это больше про **быстрые эксперименты**, чем про разработку стабильного продукта
- Соответственно, практики MLOps – это про **удобную работу с моделями машинного обучения и постановку экспериментов**
- Удивительно, как много проблем можно решить, **поговорив с людьми классически через рот**

Контейнеры — ваши друзья



Академические библиотеки часто представляют собой ад с зависимостями

Писать Dockerfile’ы на порядок проще, чем собирать пакеты



Удобно версионировать

Советы

- Сделайте **шаблонный проект**, например, с использованием Cookiecutter
- **ChatOps** в корпоративном мессенджере может сработать
- Не надо ждать, что Data Scientist’ы будут писать качественный код. **Это не их работа**
- Лучше предоставьте им инструменты, которыми будет **удобно пользоваться**
- Никогда, ни при каких условиях **не запускайте в продакшене** код **напрямую из Jupyter-ноутбуков**

Итоги. О чем поговорили:

1. Слоистая архитектура в том числе помогает избежать проблем с «застаиванием» данных
2. Большая часть проблем — не технические, а человеческие
3. Закрытость процесса и отсутствие источников информации губят производительность и мешают работать бизнесу
4. Простой и универсальный инструментарий — залог спокойного сна дата-инженера

Спасибо
за внимание!