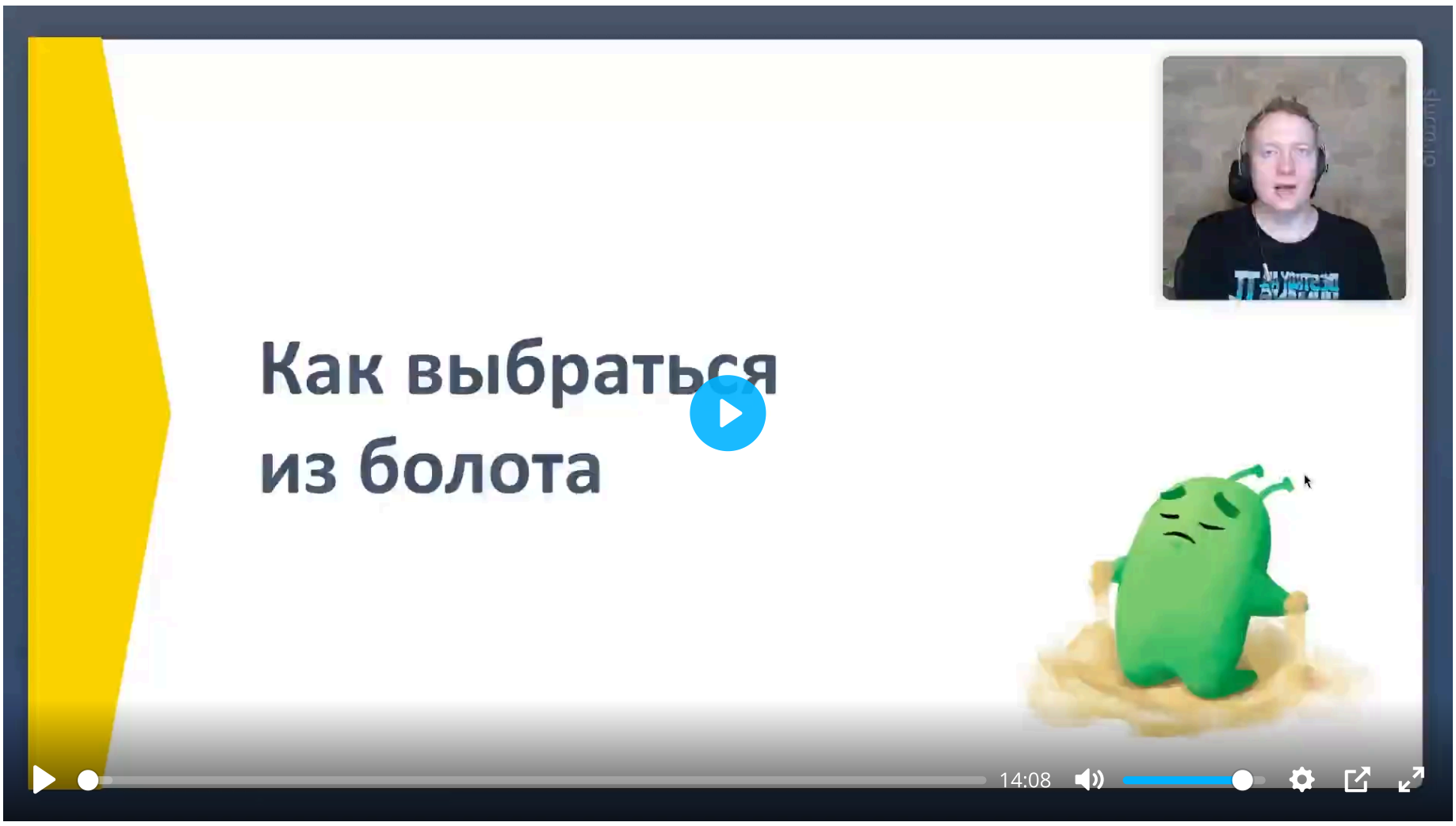
11.4 Частые проблемы при построении архитектуры и подходы к их решению



Тексовая расшифровка видео:

КАК ВЫБРАТЬСЯ ИЗ БОЛОТА

План:

- Выделим важные моменты;
- Есть риск получить Data Swamp;
- Одно общее место;
- Потенциальное решение – Data Mesh;
- Бинарные блобы и ускорение запросов;
- Потенциальное решение – Data Lakehouse;
- Гибридные решения.

Выделим важные моменты

Вспомним, о чем говорили ранее:

- 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.

Обратим более пристальное внимание на подчеркнутые слова:

- Во-первых, все сваливается в одно место;
- Во-вторых, кто-то использует структурированные БД, а кто-то –неструктурированные;
- В-третьих, в хранилищах есть бинарные блобы;
- В-четвертых, мы хотим ускорить запрос использованием специализированных форматов и т.д.

Есть риск получить Data Swamp

Попробуем перевести эти слова еще раз на русский язык:

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 понадобиться хранить картинки, видео и pdf. Зуб даем!

Ускорение запросов. Кроме даталейка нам ничего не нужно, обойдемся и так.

Одно общее место

Все это можно пытаться решать техническими способами, но нужно понимать, что в первую очередь проблема – сугубо человеческая. Все это решается словами.

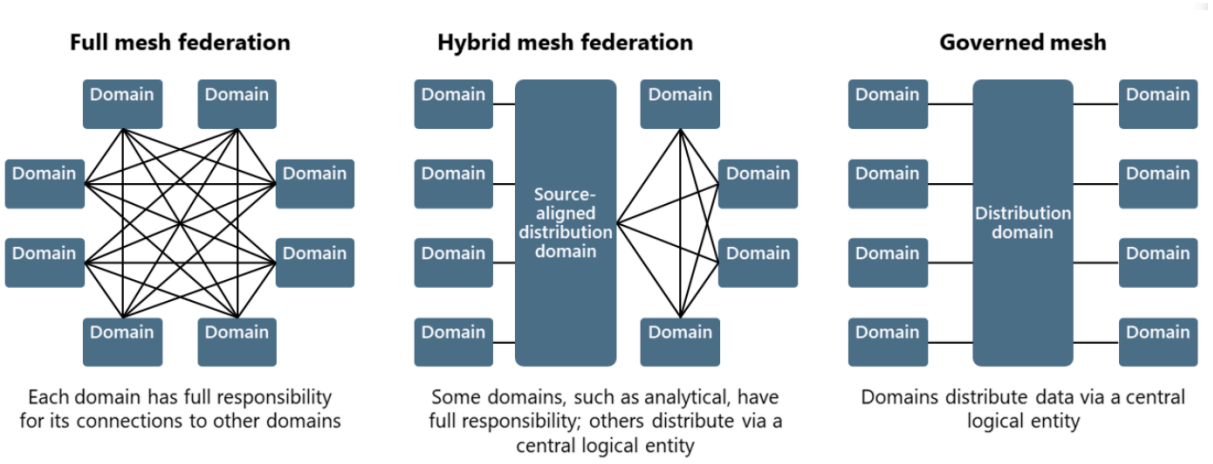
Что делать:

- Строго-настрого понять, что одно центральное место хранения ≠ одна команда поддержки.
- Нужно понимать, что мы не просто собираем данные, мы их еще и анализируем, чтобы решать какие-то бизнес-проблемы. **Собирать данные для сбора данных – бессмысленное занятие.**
- Помнить, что данные отдела – это его актив внутри компании.

Потенциальное решение – Data Mesh

Все это привело к концепции «Data Mesh».

Данные диаграммы – это наиболее разностороннее описание термина:



Идея достаточно простая: независимо от того, как данные расположены физически/технически, мы управляем ответственностью за них. У каждого датасета, который есть в системе, который выступает в роли источника для бизнес-инсайтов, должен быть владелец, к которому в случае возникновения проблем, должна быть возможность прийти и узнать причину возникшей проблемы. Поэтому получаются разные бизнес-домены.

Повторим: каждый отдел отвечает за те данные, которые он загружает в общее хранилище.

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.**

Вопрос: что делать со Stream Processing?

Ответ: в подобном подходе его, по сути, нет. Мы можем попытаться рядом «прикрутить» Spark или Flink и начать туда что-то закидывать. Однако, как объединять данные – не очень понятно. Отсюда появляется большое количество вопросов.

Потенциальное решение – Data Lakehouse

Исходя из того, что мы сказали выше, очевидным ответом будет следующее: возьмем структурированное хранилище и его накрутим, сделав полноценный Data warehouse. Однако стоит упомянуть о промежуточном хранилище «**Data Lakehouse**».

Data Lakehouse – это программный набор компонентов, который вы ставите рядом с Data Lake, и он начинает перетаскивать на себя и реализовывать часть задач, которые мы обсуждали ранее.

Гибридные решения

Вспомним проекты, о которых мы уже упоминали:

- Snowflake (работает в клаудах поверх S3-хранилищ);
- Delta Lake (работает поверх HDFS).

Если вы хотите чуть менее вендорные и чуть более опенсорсные проекты, то:

- Iceberg;
- Apache hudi;
- YTsaurus.

Как вам урок?

Изучил, далее