

Dremio

The SQL Lakehouse Platform
Быстрые данные для всех

Повестка

✓ Основы

- Форматы: CSV vs Parquet
- Память: Arrow / Arrow Flight
- Открытая архитектура: озеро данных

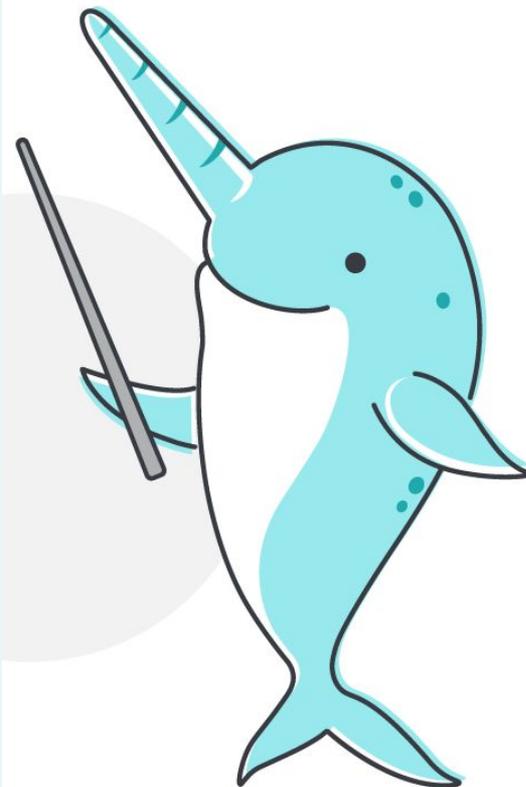
✓ Dremio

- Компания
- Демо

✓ Что дальше: Lakehouse

- Таблицы: Iceberg / Delta
- Репозиторий: Nessie

✓ QnA



Вступление



Viktor Kessler
Sr. Solutions Architect
viktor@dremio.com

<https://www.linkedin.com/in/viktor-kessler/>



Основы: Форматы Память



Строчно-ориентированно vs Колоночно-ориентированно

OLTP

для операций записи

Строчно-ориентированно



OLAP

для операций чтения

Колоночно-ориентированно



Строчно-ориентированно vs Колоночно-ориентированно

OLTP

OLAP

для операций записи

для операций чтения

Строчно-ориентированно

Колоночно-ориентированно

select name, city from customer

customer

id	name	age	city
1	Max	22	NY
2	Anna	19	NY
3	Eva	22	NY
4	Bil	22	NY

Строчно-ориентированно vs Колоночно-ориентированно

OLTP

OLAP

для операций записи

для операций чтения

Строчно-ориентированно

Колоночно-ориентированно

select name, city from customer

customer

id	name	age	city
1	Max	22	NY
2	Anna	19	NY
3	Eva	22	NY
4	Bil	22	NY

Memory

id	name	age	city
1	Max	22	NY
2	Anna	19	NY
3	Eva	22	NY
4	Bil	22	NY

Строчно-ориентированно vs Колоночно-ориентированно

OLTP

OLAP

для операций записи

для операций чтения

Строчно-ориентированно

Колоночно-ориентированно

select name, city from customer



customer

id	name	age	city
1	Max	22	NY
2	Anna	19	NY
3	Eva	22	NY
4	Bil	22	NY

Memory

id	name	age	city
1	Max	22	NY
2	Anna	19	NY
3	Eva	22	NY
4	Bil	22	NY

Memory

name	city
Max	NY
Anna	NY
Eva	NY
Bil	NY

Строчно-ориентированно vs Колоночно-ориентированно

OLTP

для операций записи

Строчно-ориентированно

OLAP

для операций чтения

Колоночно-ориентированно

**select name, city from customer
where age = 22**

customer

id	name	age	city
1	Max	22	NY
2	Anna	19	NY
3	Eva	22	NY
4	Bil	22	NY

Memory

id	name	age	city
1	Max	22	NY
2	Anna	19	NY
3	Eva	22	NY
4	Bil	22	NY

statistics: min/max age, footer

22

Max

NY

Eva

NY

Bil

NY

Строчно-ориентированно vs Колоночно-ориентированно

OLTP

OLAP

для операций записи

для операций чтения

Строчно-ориентированно

Колоночно-ориентированно

сжатие данных



customer

id	name	age	city
1	Max	22	NY
2	Anna	19	NY
3	Eva	22	NY
4	Bil	22	NY

on disk

id	name	age	city
1	Max	22	NY
2	Anna	19	NY
3	Eva	22	NY
4	Bil	22	NY

on disk

id	1	2	3	4
----	---	---	---	---

name	Max	Anna	Eva	Bil
------	-----	------	-----	-----

age	22	19
-----	----	----

city	NY
------	----

Apache Parquet

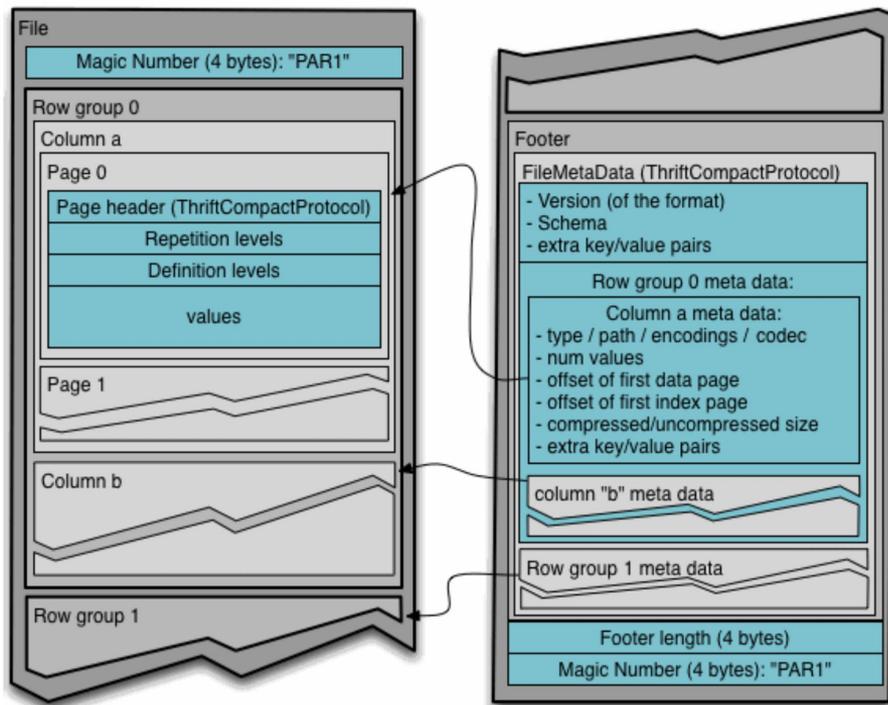


id	1	2	3	4
----	---	---	---	---

name	Max	Anna	Eva	Bil
------	-----	------	-----	-----

age	22	19
-----	----	----

city	NY
------	----



Apache Arrow vs proprietary



id	1	2	3	4
----	---	---	---	---

name	Max	Anna	Eva	Bil
------	-----	------	-----	-----

age	22	19
-----	----	----

city	NY
------	----

Apache Arrow vs proprietary



id	1	2	3	4
----	---	---	---	---

name	Max	Anna	Eva	Bil
------	-----	------	-----	-----

age	22	19
-----	----	----

city	NY
------	----



in Memory

id	name	age	city
1	Max	22	NY
2	Anna	19	NY
3	Eva	22	NY
4	Bil	22	NY

Apache Arrow vs proprietary



id	1	2	3	4
----	---	---	---	---

name	Max	Anna	Eva	Bil
------	-----	------	-----	-----

age	22	19
-----	----	----

city	NY
------	----

Read

A large black arrow pointing from the Parquet data representation to the Apache Arrow data representation.

in Memory

id	name	age	city
1	Max	22	NY
2	Anna	19	NY
3	Eva	22	NY
4	Bil	22	NY



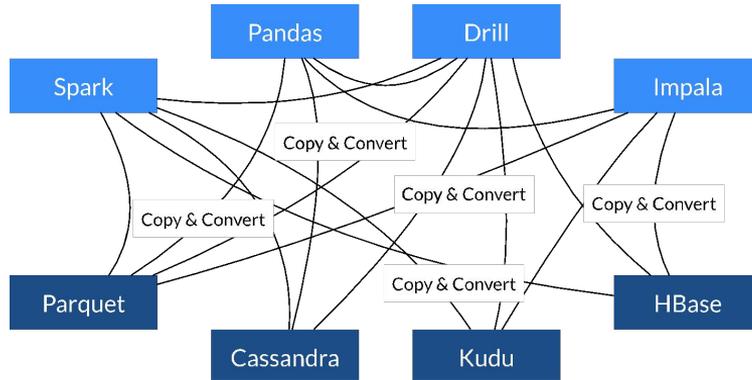
id	1	2	3	4
----	---	---	---	---

name	Max	Anna	Eva	Bil
------	-----	------	-----	-----

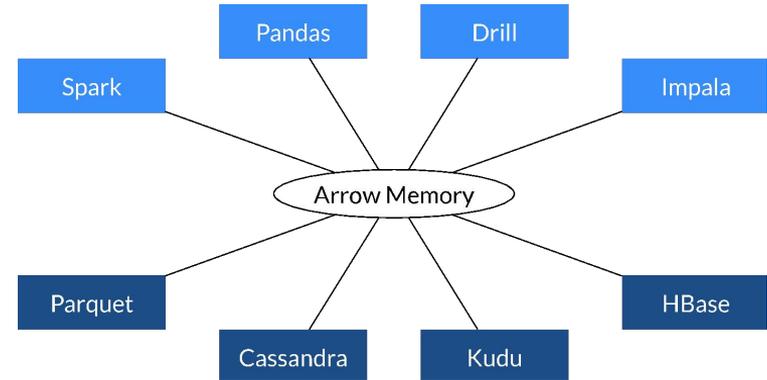
age	22	19
-----	----	----

city	NY
------	----

Apache Arrow vs proprietary



- Each system has its own internal memory format
- 70-80% CPU wasted on serialization and deserialization
- Similar functionality implemented in multiple projects

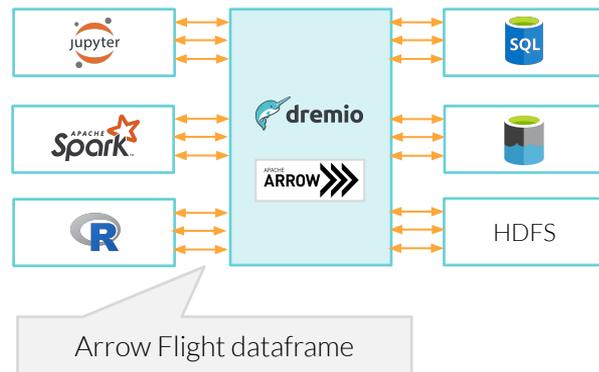


- All systems utilize the same memory format
- No overhead for cross-system communication
- Projects can share functionality (eg, Parquet-to-Arrow reader)

Arrow Flight

Протокол для высокопроизводительной передачи данных

- ODBC & JDBC создан в 1990-х годах и не предназначен для больших данных или данных масштаба озера данных.
- Новый стандарт для всех систем и единый клиент для Arrow для поддержки всех систем
- Arrow Flight - gRPC протокол для массивной параллельной обработки



Открытая архитектура: озеро данных

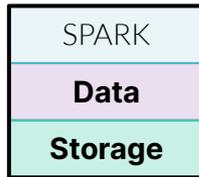
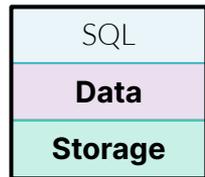


Новая архитектура данных

... - 2015

**Хранилище
данных
& Hadoop**

Совместное
размещение
вычисления &
хранения

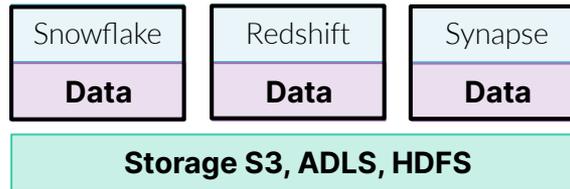
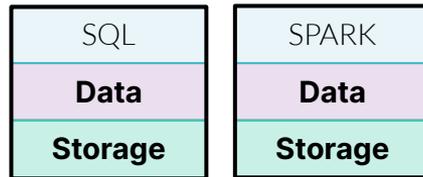


Новая архитектура данных

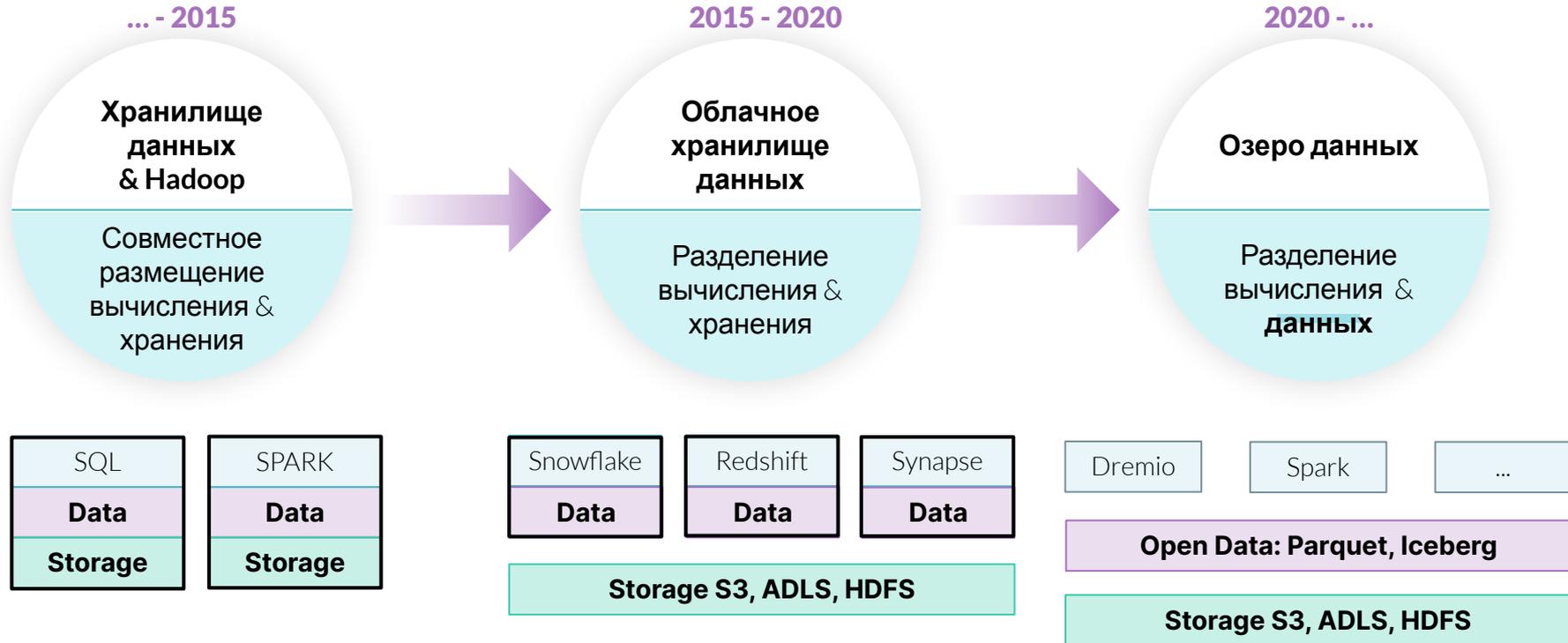
... - 2015



2015 - 2020



Новая архитектура данных



Гибкость сегодня и в будущем

Ваш выбор облачных услуг



Ваш выбор сервисов и инструментов



Избегайте
Lock-in
в облаке или
хранилище
данных



“With a data lake, data is stored in an *open format*, which makes it easier to work with different analytic services. Open format also makes it more likely for the data to be compatible with tools *that don't even exist yet*.”

Various roles in your organization, like data scientists, data engineers, application developers, and business analysts, can access data with *their choice* of analytic tools and frameworks.”

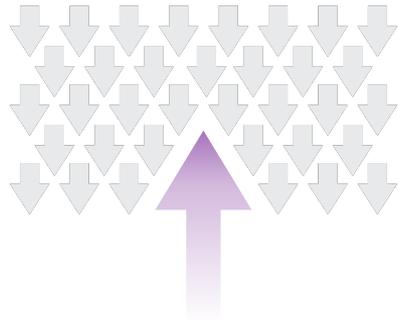


Werner Vogels
CTO, Amazon

Dremio



Dremio Business Overview



Unicorn Status

Over \$200M Raised in 2020

\$1B Valuation



Nessie

Open Source Innovation

Creator of Arrow, Arrow Flight & Nessie

20M Downloads/Month



10K+
attendees

Thought Leadership

Presenters of Subsurface Cloud
Data Lake Conference

Enterprise Customer Adoption



Fannie Mae



Boehringer
Ingelheim



Strategic Technology Partners

Dremio partners with innovative technology providers to deliver high-performing solutions for data lake analytics.



My Data is in the Lake!

Data Scientists



BI Users



Cloud Data
Lake Storage



or

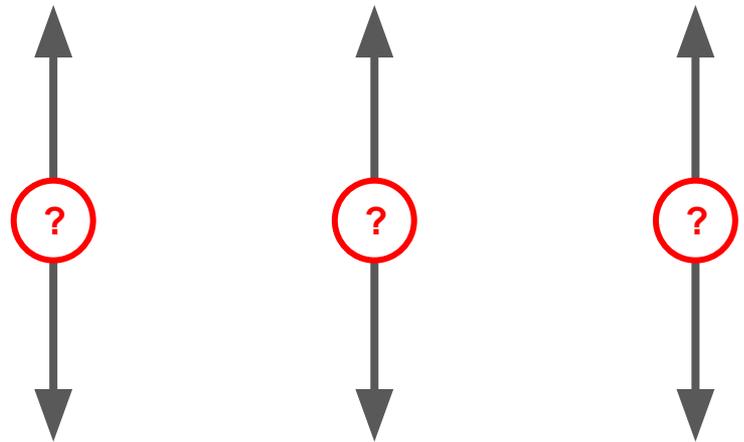


or

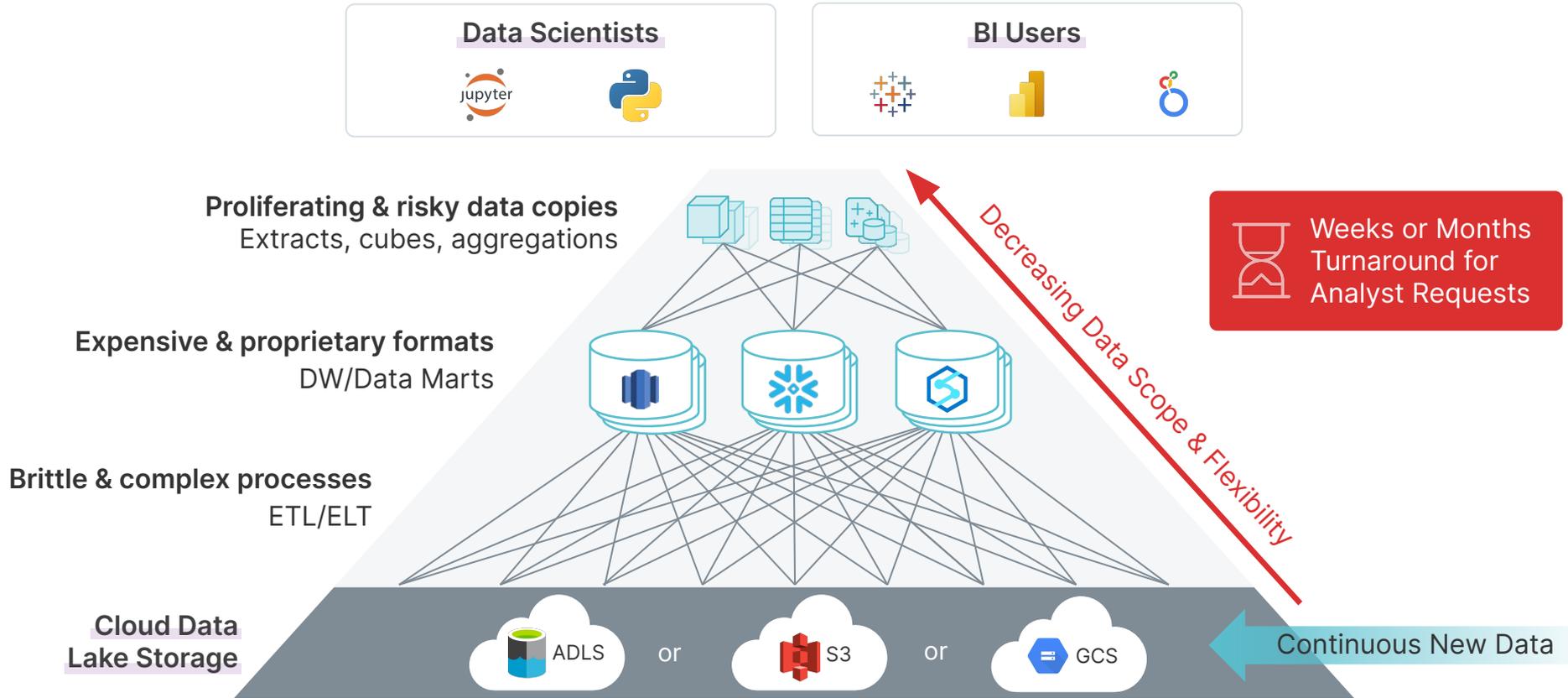


Continuous New Data

And how can I read it?



Pyramid of Doom



Упрощённая архитектура

Data Scientists



BI Users



Data Analyst Self-Service
Semantic Layer

BI & Ad-Hoc Performance
Query Accelerator

Query Data in Place
SQL Engine

Cloud Data
Lake Storage



or



or



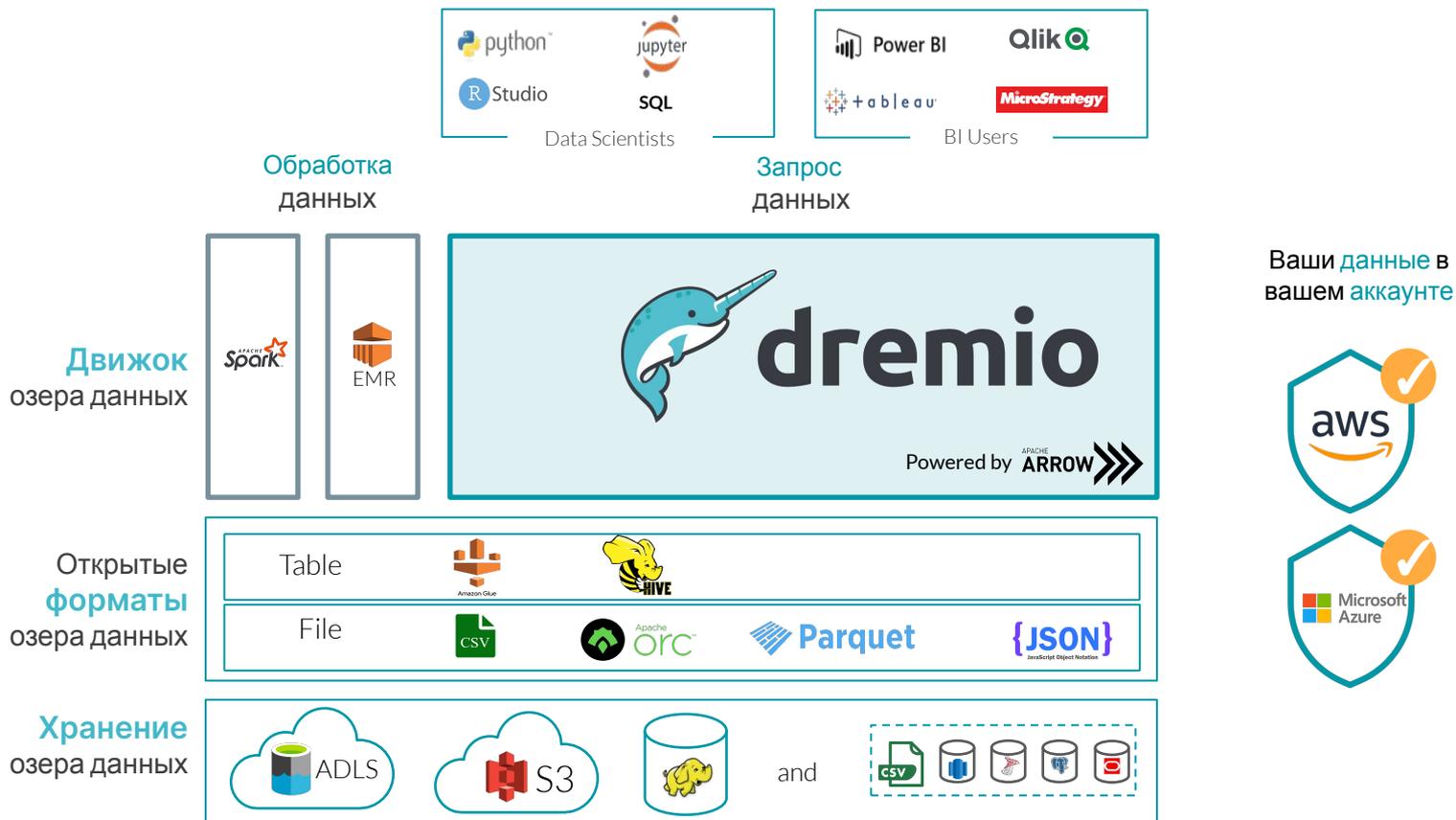
Full Data Available for Analysis



Hours or Days
Turnaround for
Analyst
Requests

Continuous New Data

Dremio обеспечивает открытую архитектуру для анализа данных



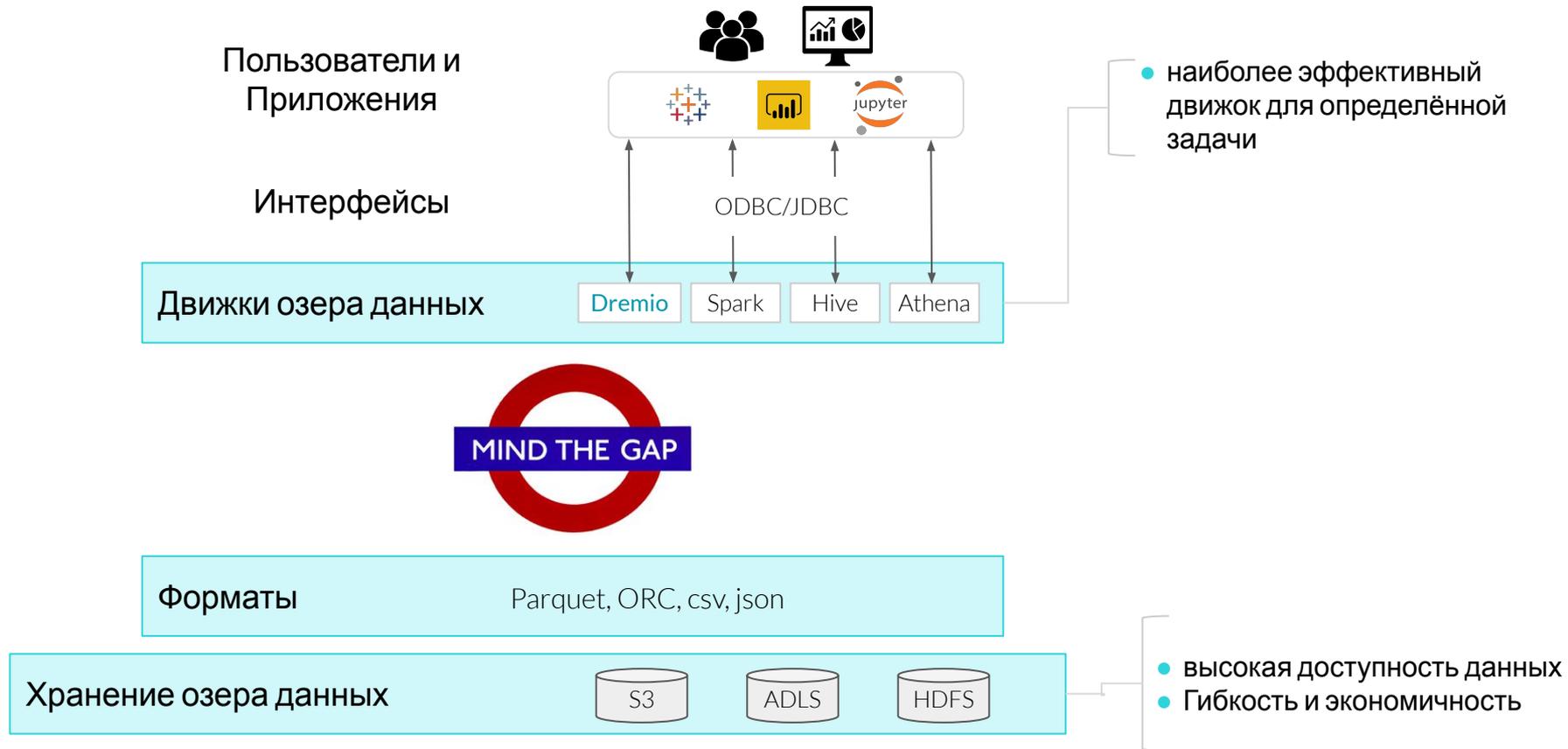
DEMO



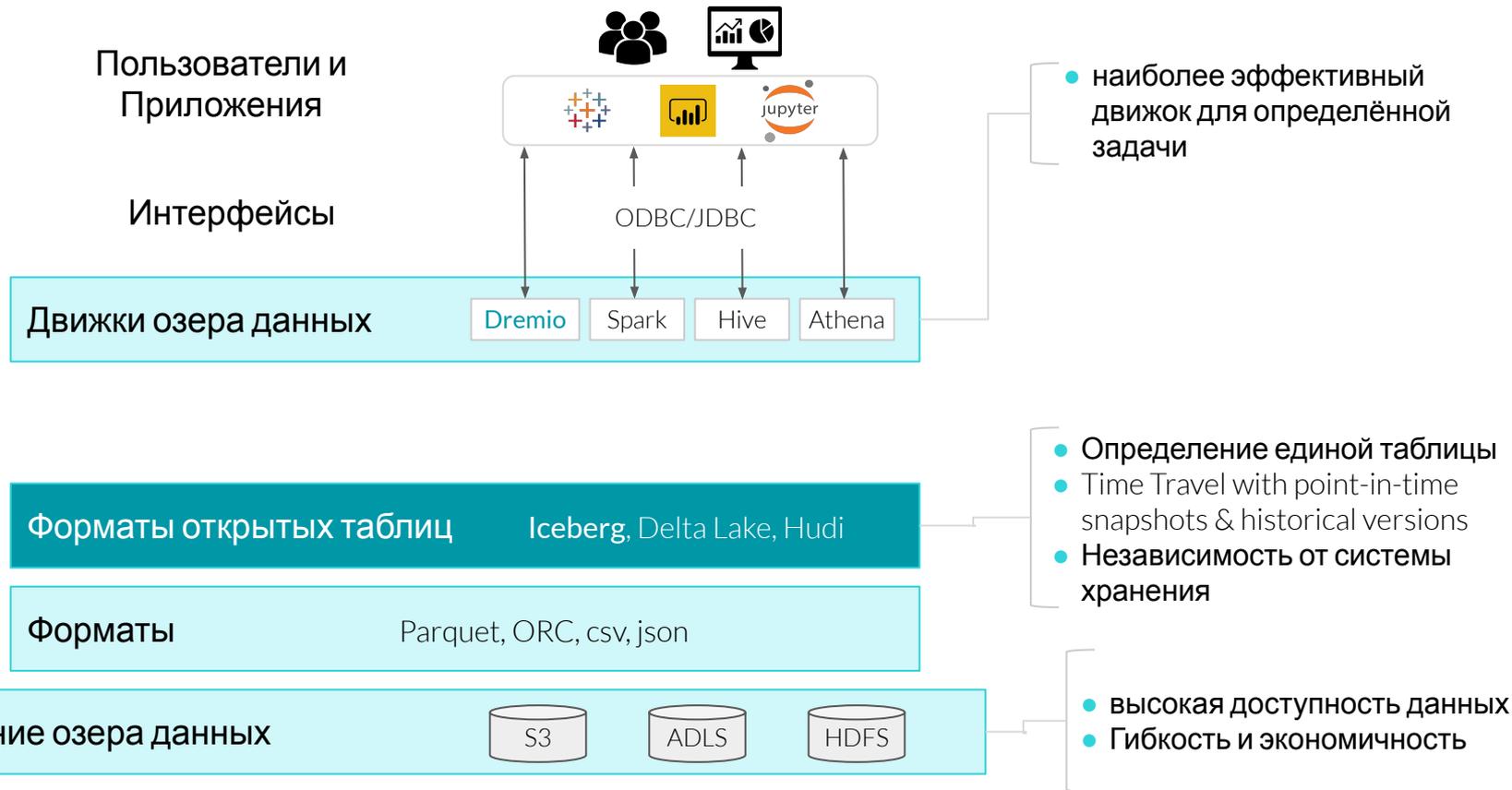
Что дальше?



Архитектура озера данных - до-2021



Архитектура озера данных - до-2021



Транзакций

Record-level data mutations with SQL DML

```
INSERT INTO t1 ...  
UPDATE t1 SET col1 = ...  
DELETE FROM t1 WHERE state = "CA"
```

Automatic partitioning

```
CREATE TABLE t1 PARTITIONED BY (month(date),  
bucket[1000](user))
```

Instant schema and partition evolution

```
ALTER TABLE t1 ADD/DROP/MODIFY/RENAME COLUMN c1 ...  
ALTER TABLE t1 ADD/DROP PARTITION FIELD ...
```

Time travel

```
SELECT * FROM t1 AT/BEFORE <timestamp>
```



- Created by Netflix, Apple and other big tech
- INSERT/UPDATE/DELETE with any engine
- Strong momentum in OSS community



- Created by Databricks
- INSERT/UPDATE with Spark, SELECT with any engine
- Primarily used in conjunction with Databricks

Ключевые новые функции уровня ДАННЫХ

Транзакций

Поддержка новых стандартов для форматов таблиц: Apache Iceberg и Delta Lake

Масштабирование и автоматизация

Источники Hive & Filesystem с неограниченными размерами таблиц, запросов и обновлениями метаданных почти в реальном времени

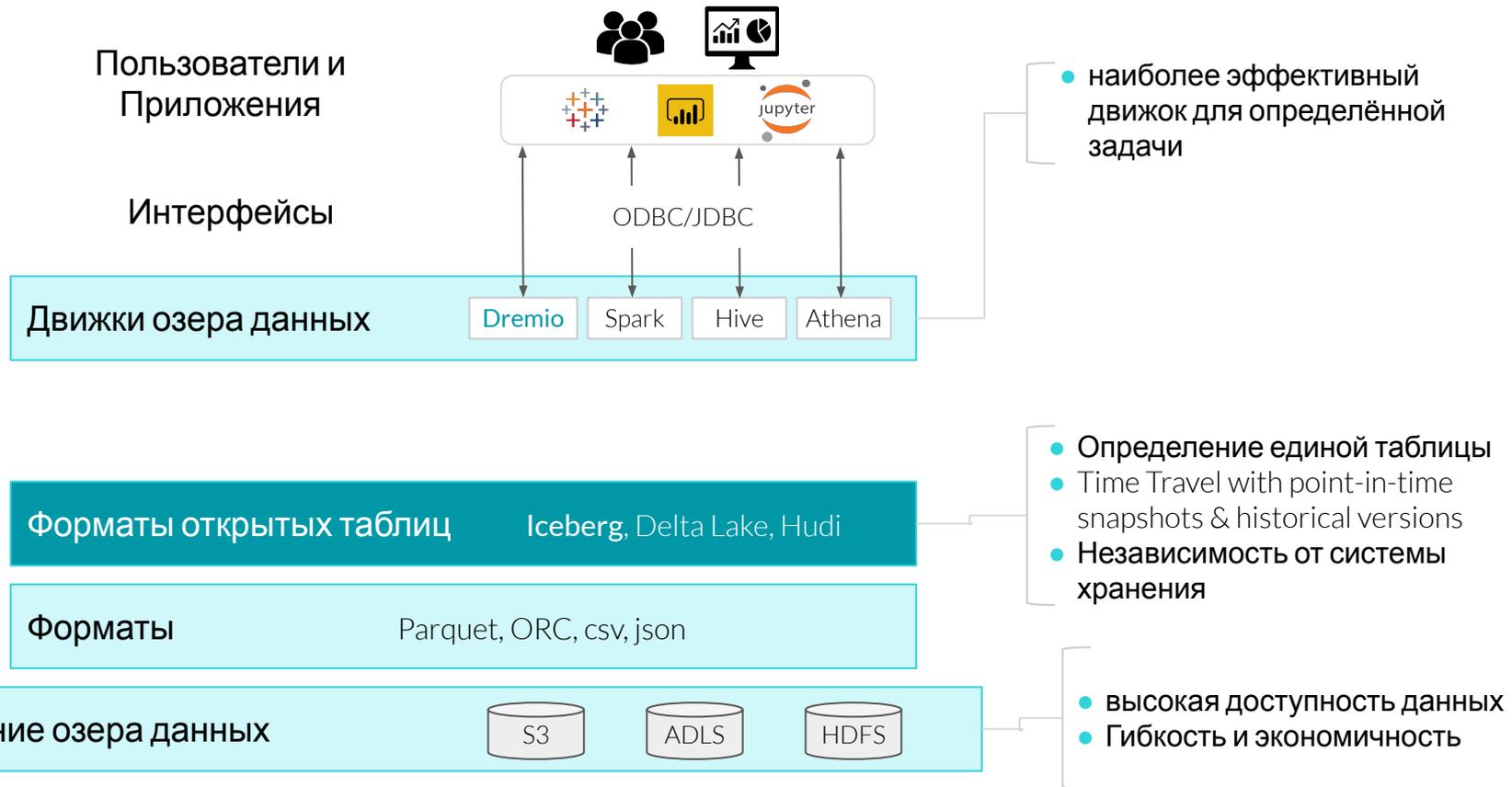
DML операции

ANSI SQL DML операции для закрытия функционального пробела в сравнение с хранилищем данными.
Dremio не только чтения данных а также для обновления

Внутренняя схема

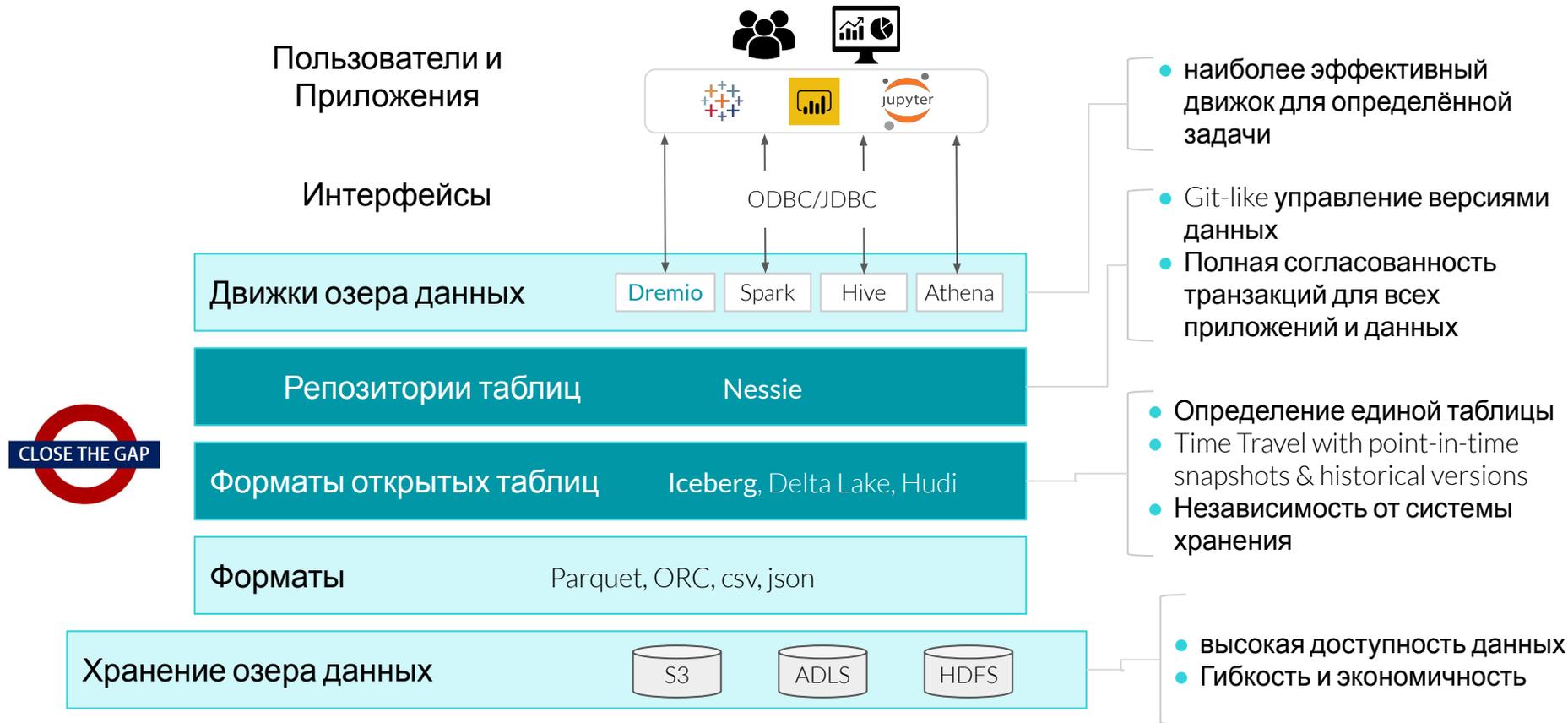
Возможность управлять определением схемы данных в источниках файловой системы.

Архитектура озера данных - до-2021



CLOSE THE GAP

Архитектура озера данных - до-2021



Ключевые новые функции уровня ДАННЫХ

Транзакций

Поддержка новых стандартов для форматов таблиц: Apache Iceberg и Delta Lake

Масштабирование и автоматизация

Источники Hive & Filesystem с неограниченными размерами таблиц, запросов и обновлениями метаданных почти в реальном времени

DML операции

ANSI SQL DML операции для закрытия функционального пробела в сравнение с хранилищем данными.
Dremio не только чтения данных а также для обновления

Внутренняя схема

Возможность управлять определением схемы данных в источниках файловой системы.

Git-like управления данными

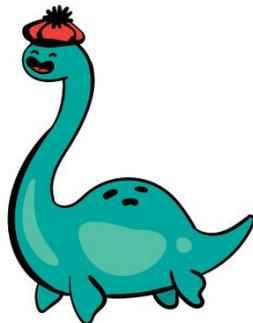
Новые возможности превосходящие функциональности хранилища данных

Nessie: A Git-like Experience for the Data Lake

Queries on branches/tags/times

- Reproducibility
- Compare datasets
- Compliance

```
USE BRANCH 'main'  
SELECT * FROM t1 // main implicit  
SELECT * FROM t1@etl  
SELECT * FROM t1 AT '2020-10-26'  
SELECT * FROM t1@etl '2020-10-26'
```

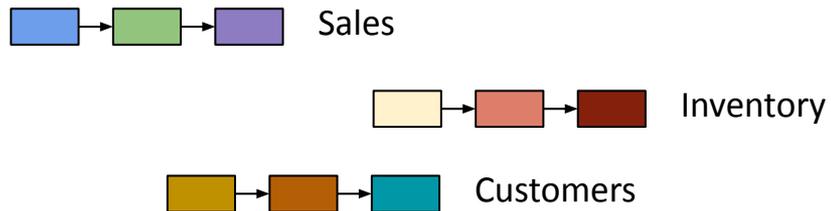


Multi-table & multi-engine transactions

- Multi-table consistency
- Experimentation
- Data promotion workflows (dev→stage→prod)

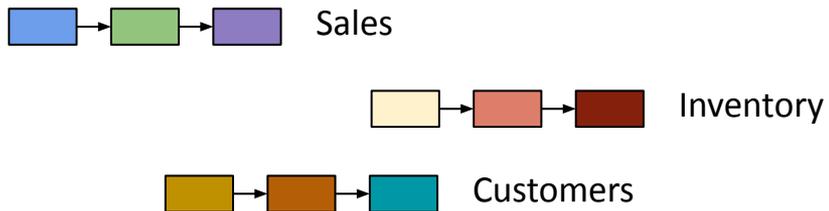
```
CREATE BRANCH etl  
[SPARK] Data Transformation customers  
[SPARK] Data Transformation orders  
[DREMIO] Reflection Refresh customers  
[DREMIO] Reflection Refresh orders  
USE BRANCH main  
MERGE BRANCH etl
```

Lakehouse без Nessie



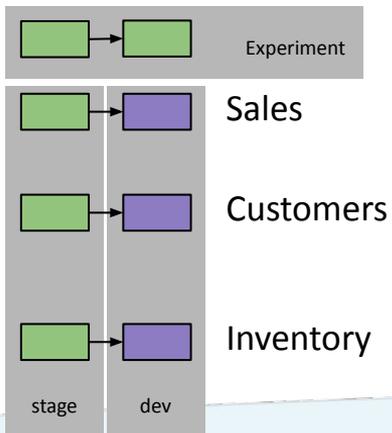
- несвязанные истории
- поддерживает только линейные изменения

Lakehouse **без** Nessie



- несвязанные истории
- поддерживает только линейные изменения

Lakehouse **с** Nessie



- связанные истории
- нелинейные изменения
- консистенция между таблицами
- кросс-табличные транзакции

Итог: Dremio SQL Lakehouse

Транзакции

- изменения на уровне строк с помощью SQL DML операции
- встроенная поддержка формата Таблиц Apache Iceberg и Delta Lake

Git-like контроль версионирования

- Data branching: верификация, multi-table
- Data version control: воспроизводимость и соответствие
- путешествие во времени (SELECT ... AS OF)

Автоматизация

- Оптимизация форматов
- сбор статистики
- индексация
- Валидация данных (RI)

Как начать?

<https://university.dremio.com/>

<https://www.dremio.com/deploy>

The screenshot shows the 'Deployment Options' page on the Dremio website. At the top, there is a navigation bar with the Dremio logo and links for Platform, Why Dremio, Customers, Partners, Learn, Company, and Contact Us. A 'Get Started' button is located in the top right corner. The main heading is 'Deployment Options'. Below it, a paragraph states: 'Dremio makes it really simple to run SQL queries directly against cloud data lake storage without making copies of your data. There are multiple deployment options to fit your needs.' There are three main sections: 'Dremio on Azure', 'Dremio on AWS', and 'Dremio On Prem'. Each section includes a brief description and a 'Start course' button. The 'Dremio on Azure' section lists 'Azure Marketplace' and 'AKS'. The 'Dremio on AWS' section lists 'AWS Marketplace' and 'EKS'. The 'Dremio On Prem' section lists 'Kubernetes', 'Docker', and 'YARN'. The page has a light blue background with decorative elements at the bottom.

The screenshot shows the 'Dremio University' website. At the top, there is a navigation bar with the Dremio logo and links for Explore courses, Docs, and Support. There are 'Register' and 'Sign in' buttons in the top right corner. The main heading is 'Dremio University'. Below it, a paragraph states: 'Free online courses to optimize your data lake'. There is a 'Register Now' button. The 'Course Highlights' section features three course cards: 'Dremio Fundamentals', 'Data Reflections', and 'Dremio for Data Consumers'. Each card includes an illustration, a brief description, and a 'Start course' button. The background is a teal color with a pattern of icons related to data and technology.

Community Edition

Single Deployment: `docker run -d -p 9047:9047 -p 31010:31010 -p 45678:45678 dremio/dremio-oss`

AWS Marketplace: <http://docs.dremio.com/deployment/aws/aws-edition-quickstart/>

Kubernetes Helm: https://github.com/dremio/dremio-cloud-tools/tree/master/charts/dremio_v2

QnA

viktor@dremio.com

<https://www.linkedin.com/in/viktor-kessler/>

