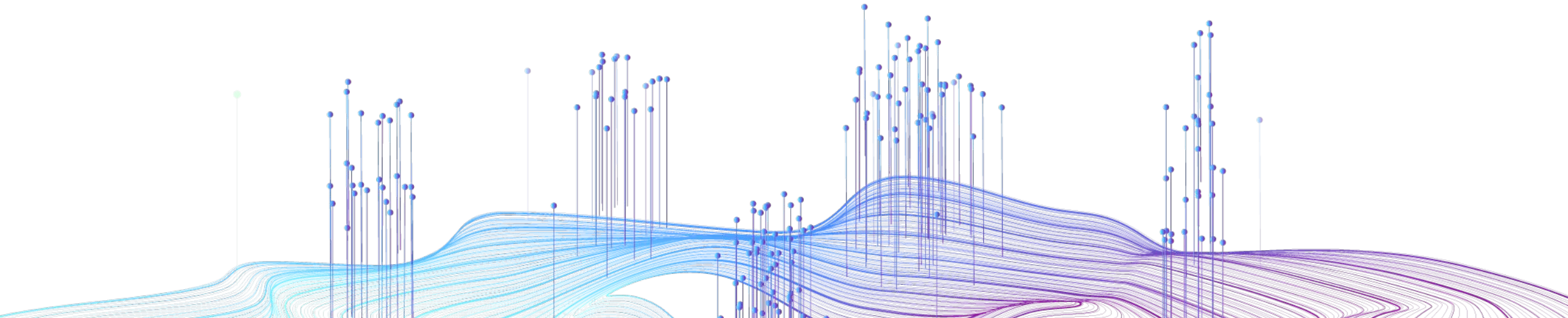
VOLTRON DATA

Apache Arrow: Open Source Standard Becomes Enterprise Necessity

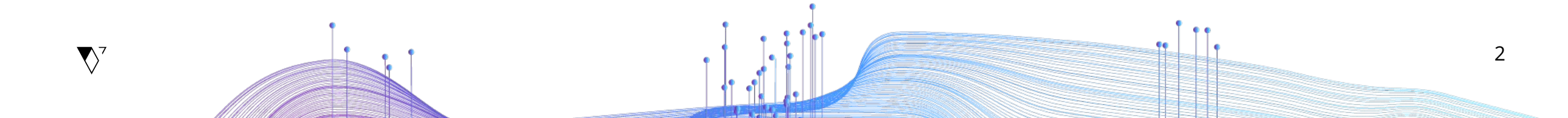
March 3, 2022
Subsurface Live Winter 2022

Wes McKinney
@wesmckinn



About Me

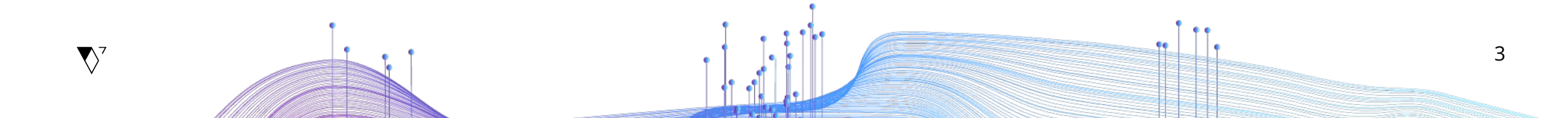
- **Apache Arrow** co-creator, PMC member
- **pandas** creator
- **Voltron Data** co-founder, CTO
- Previously at **Ursa Labs**, **Two Sigma**, **Cloudera**, **AQR**
- Author of ***Python for Data Analysis***



Apache Arrow

Multi-language toolbox for accelerated data interchange and in-memory processing

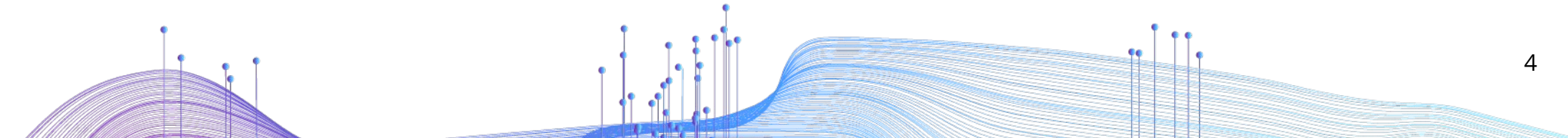
- Founded in **2016** by a group of open source developers
- Enables unification of **database** and **data science** tech stacks
- Thriving **user and developer community**
- Provides implementations or bindings in **twelve languages**
- Adopted by **numerous projects and products** in the ecosystem



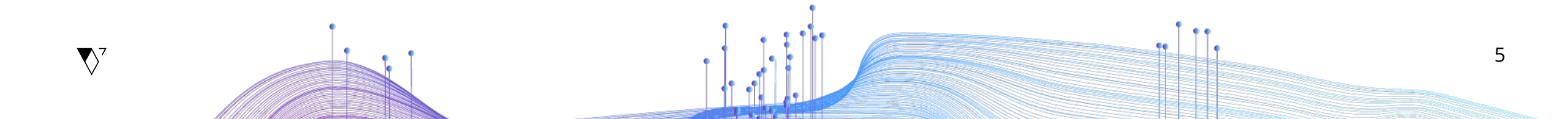
Voltron Data

- Creating a **unified foundation** for the future of analytical computing with **Apache Arrow**
- Founded in **2021**
- Raised **$110M** in seed and Series A funding
- Leading corporate **contributor to Arrow**

We're hiring!
voltrondata.com/careers



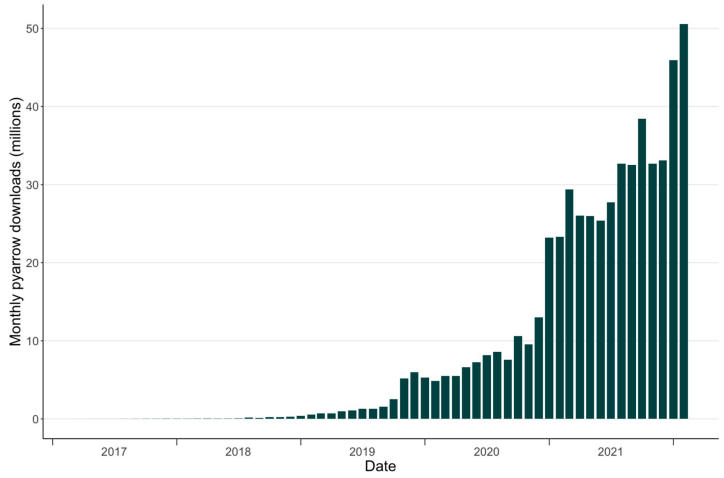
What's New in Apache Arrow



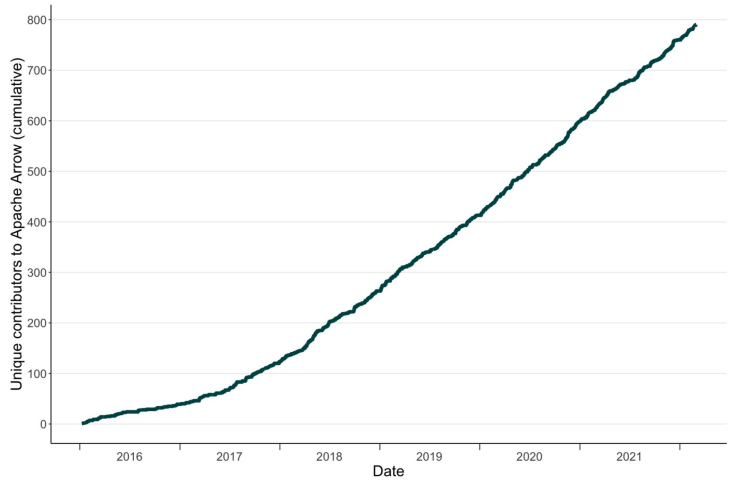
Project and Ecosystem Growth

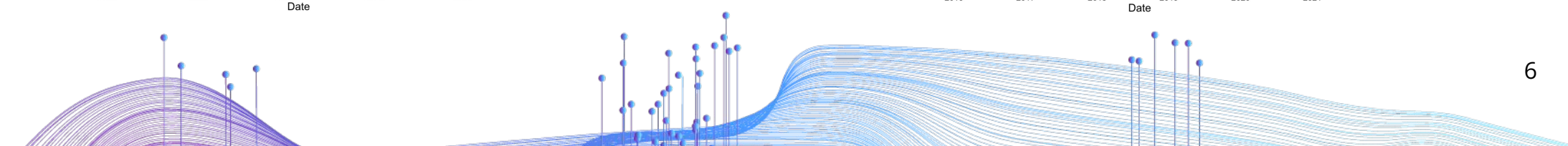
- Sustained growth in **users**, **contributors**, **applications**
- Arrow version **7.0.0** (**24th** major release) in February 2022

>50M monthly PyArrow downloads

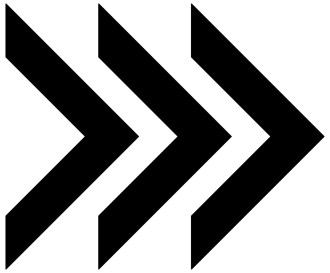


≈800 Arrow contributors



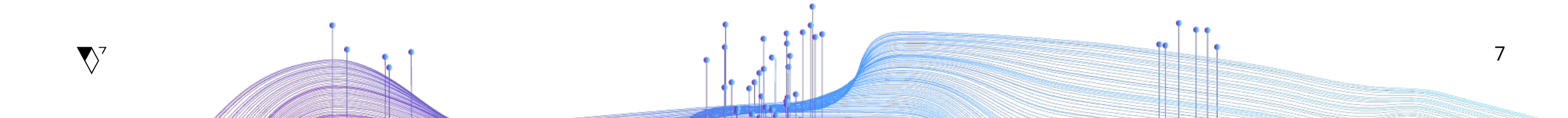


Arrow Flight SQL

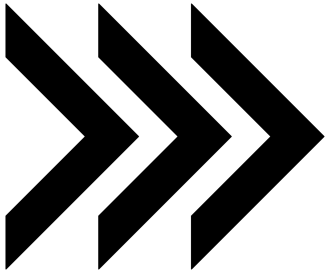


Next-generation standard for data access using SQL

- Adds SQL semantics to **Arrow Flight**
- Enables **ODBC/JDBC-style data access** at the speed of Flight
- Reduces implementation complexity for developers
- Shipped in **Arrow version 7.0.0** (February 2020)
  - **C++** and **Java** implementations available
  - Additional development and documentation is ongoing 🚧

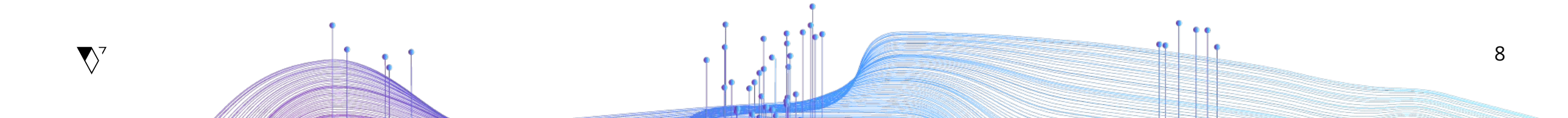


Arrow C++ Query Engine

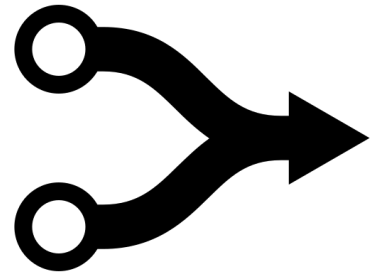


Work is ongoing to implement comprehensive query execution capabilities in the Arrow C++ library

- Common scalar functions ✔️
- Common grouped aggregate functions ✔️
- Common joins ✔️
- Performance and efficiency improvements 🚧
- Window functions 🚧
- Additional join types 🚧

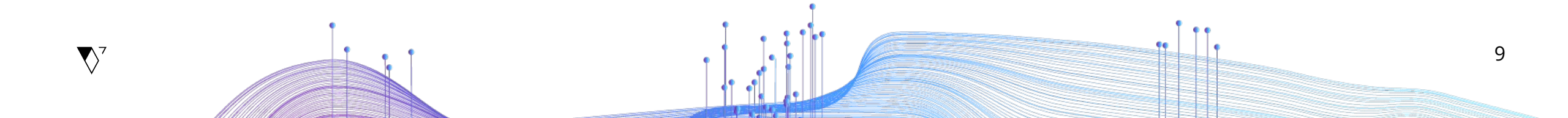


Substrait

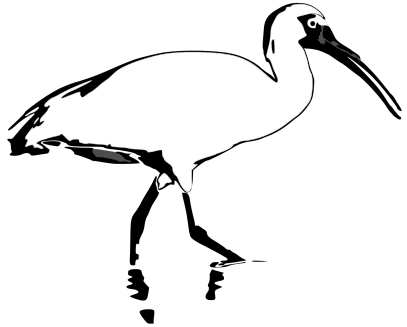


A cross-language, interoperable specification for data compute operations

- A standard, flexible way for **APIs** and **engines** to share the representation of analytics computations
  - Produced by APIs (**dplyr**, **Ibis**, **SQL**, ...)
  - Consumed by engines (**Arrow C++ engine**, **DuckDB**, ...)
- Work is underway to implement producers and consumers 🚧

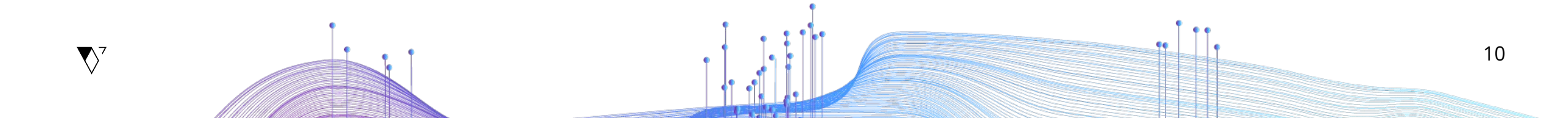


Ibis

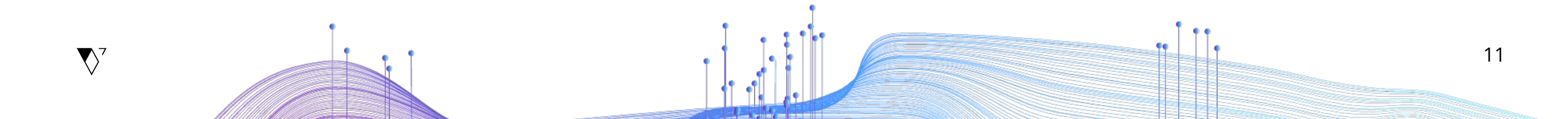


A high-level Python API for data analysis

- Fluent **pandas-like** syntax
- Can express virtually any SQL query
- Supports **modular backends** for querying systems including PostgreSQL, MySQL, SQLite, Impala, ClickHouse, BigQuery
- Work is underway to produce **Substrait** plans 🚧
  - Enabling Ibis queries to run on the **Arrow C++ engine**



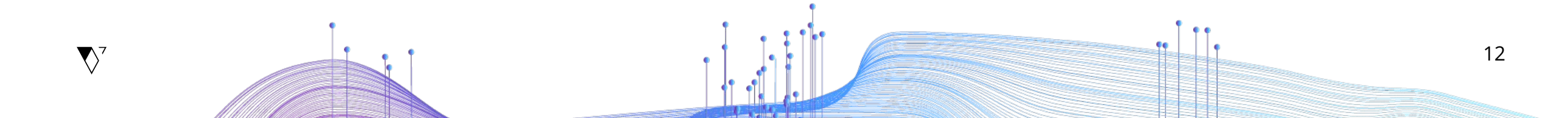
Challenges that Enterprises Are Solving with Apache Arrow



Challenge: Obstacles to Interoperability

- Enterprise data tools and workflows often incorporate multiple **languages**, **query engines**, **storage systems**, and **clouds**.

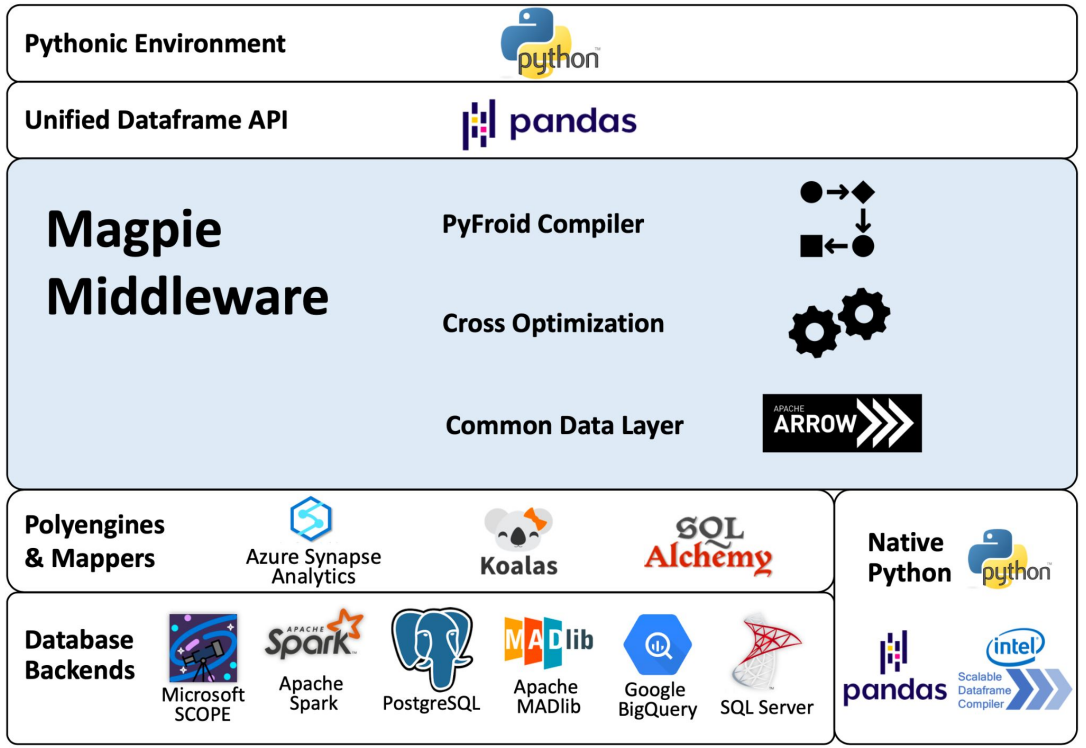
- Arrow provides **standard formats, interfaces, and frameworks** to enable efficient integration of these diverse components.



Case Study: Microsoft

Microsoft uses Arrow in their Magpie data science middleware

- Unifies multiple cloud and database backends into a single end user interface
- Stores intermediate results as **Arrow Tables**
- Uses **Arrow Flight** to move data between systems

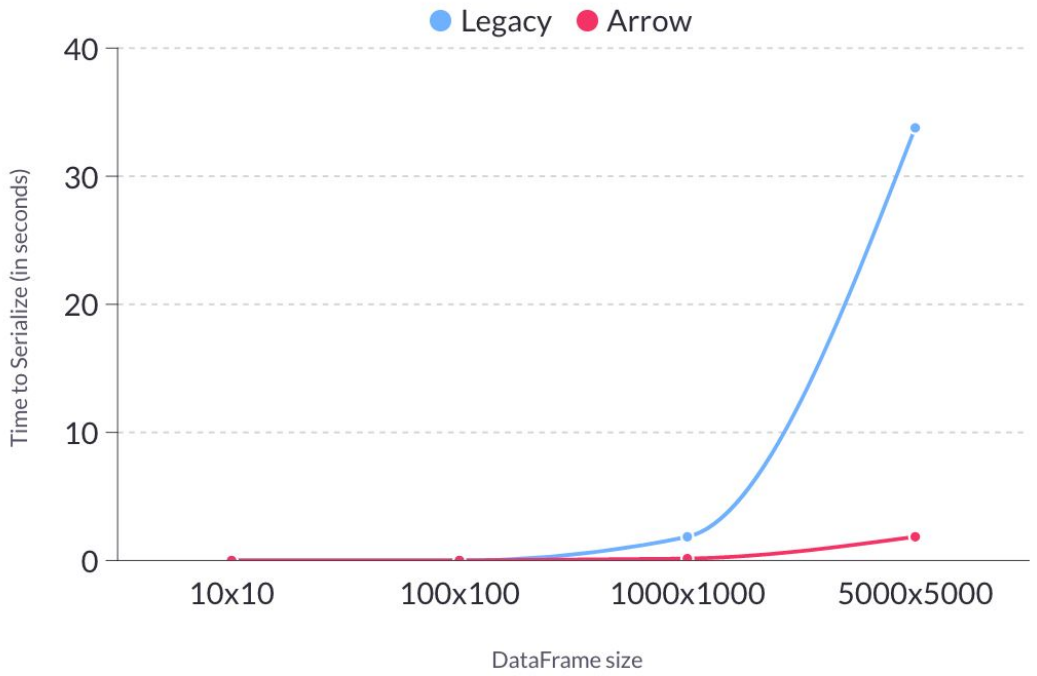


Source: http://cidrdb.org/cidr2021/papers/cidr2021_paper08.pdf

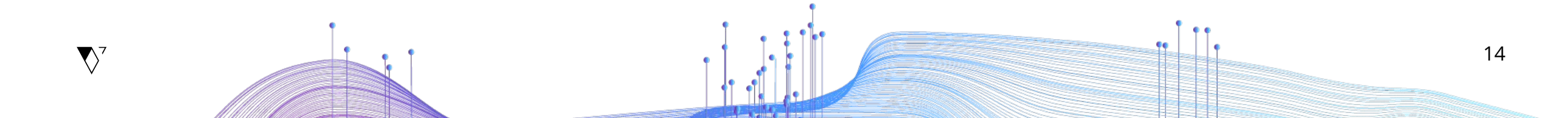
Case Study: Streamlit

Streamlit uses Arrow to move data from Python to JavaScript

- Reduces server–browser data transfer times
- Simplifies addition of new features to Streamlit
- Allowed deletion of >1000 lines of custom code



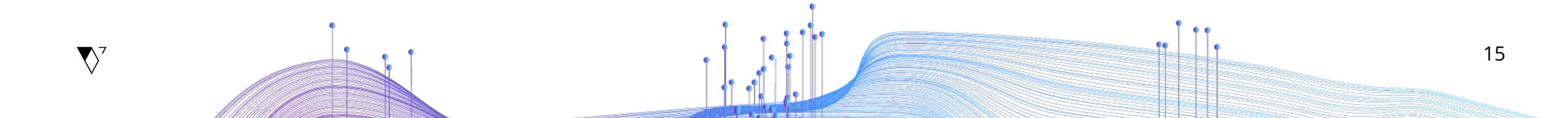
Source: https://blog.streamlit.io/all-in-on-apache-arrow/



Challenge: Slow Data Access Protocols

- Legacy data transport protocols are often slowed by **serialization and deserialization overhead** and other causes.

- The **Arrow columnar data format** and the **Arrow Flight framework** enable fast, serialization-free data transport.

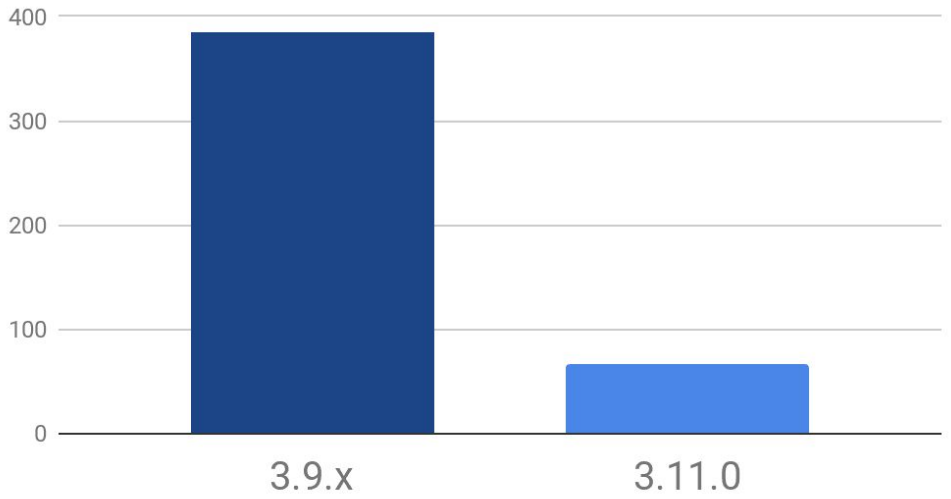


Case Study: Snowflake

Snowflake uses Arrow in its JDBC client, Python client, and Spark connector

- Eliminates serialization between columnar and row-oriented data formats
- Achieves **4 – 10x reduction** in fetch times

Figure 1. JDBC fetch performance benchmark for JDBC client version 3.9.x (without Arrow) versus 3.11.0 (with Arrow)

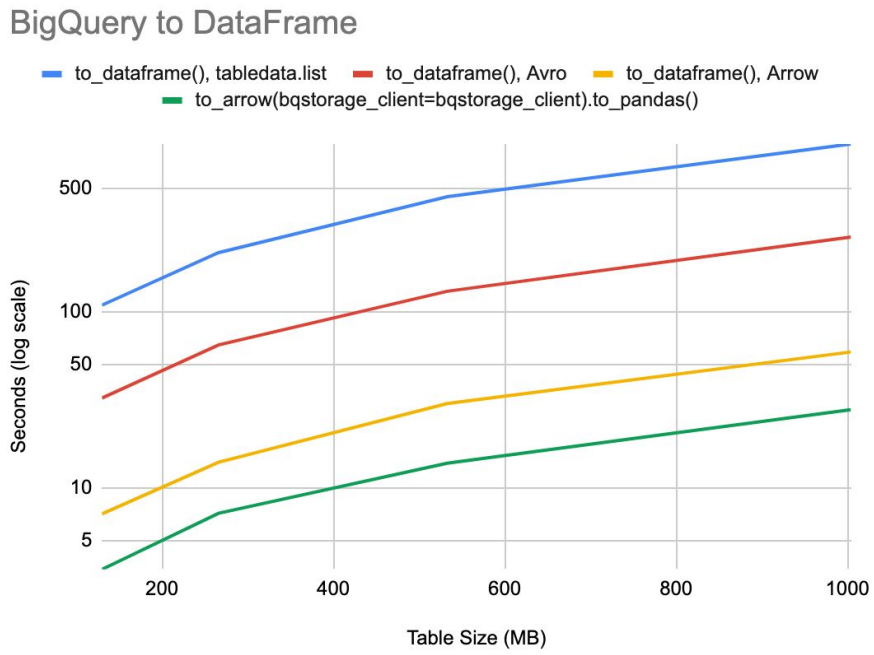


Source: https://www.snowflake.com/blog/fetching-query-results-from-snowflake-just-got-a-lot-faster-with-apache-arrow/

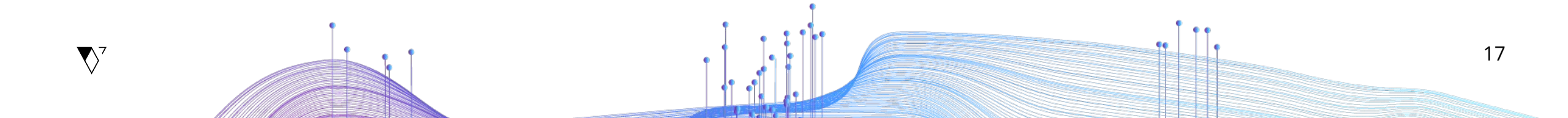
Case Study: Google Cloud Platform

Google Cloud uses Arrow in its BigQuery Storage API

- Eliminates or speeds serialization to multiple data formats
- Achieves **4 – 31x reduction** in download times
- Speedup is stable across data sizes



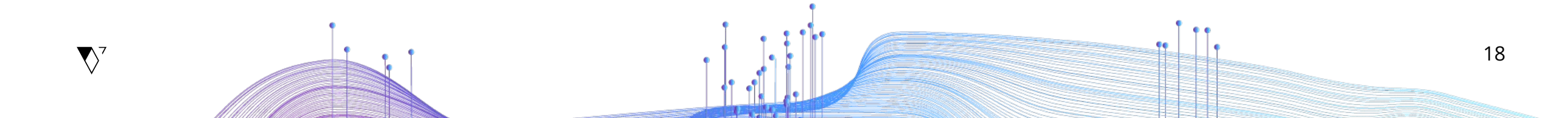
Source: https://medium.com/google-cloud/announcing-google-cloud-bigquery-version-1-17-0-1fc428512171



Challenge: Limits of JVM-Based Engines

- Computing engines implemented in JVM languages including Java and Scala often suffer from **performance bottlenecks**.

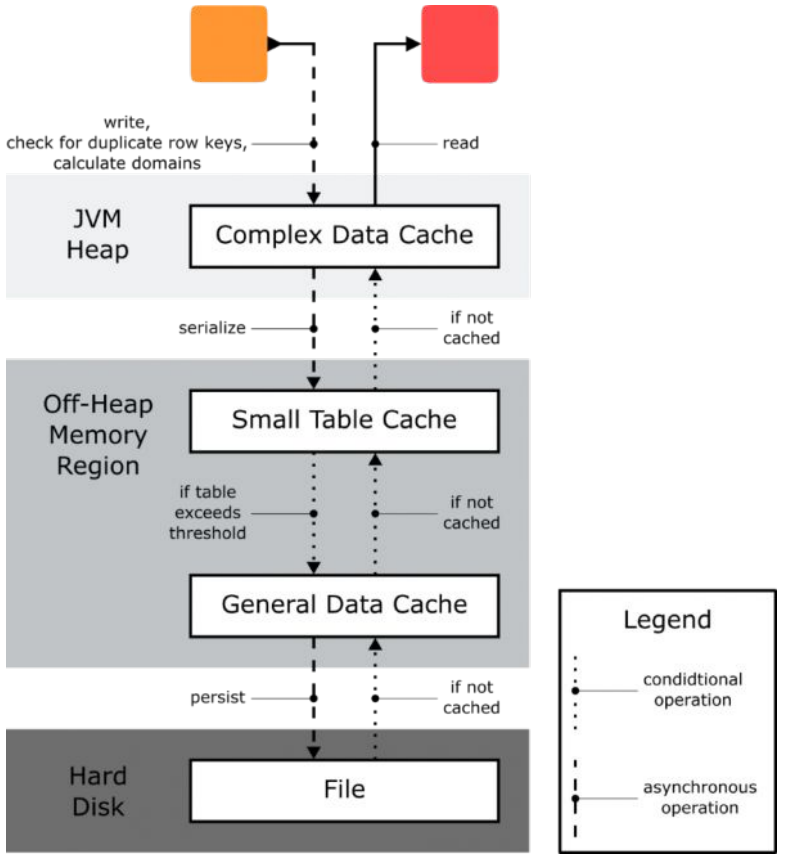
- Arrow columnar data structures can **accelerate JVM-based engines** and enable use of **pluggable high-performance components** implemented in lower-level languages like C++.



Case Study: KNIME

KNIME uses Arrow in its Columnar Table Backend

- Stores data more compactly in memory, improving performance
- Eliminates the need to create Java objects to represent table elements, reducing GC pressure
- Enables use of shared memory



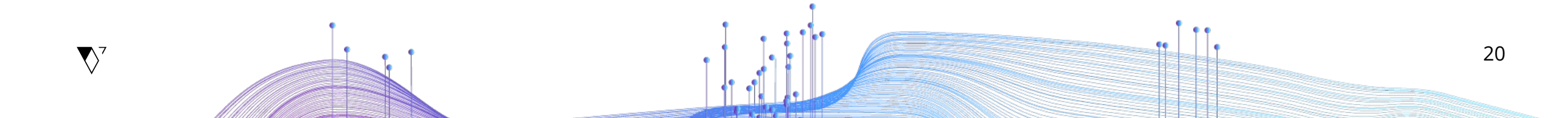
Source: https://www.knime.com/blog/improved-performance-with-new-table-backend

Case Study: Meta

Meta uses Arrow in its Velox C++ database acceleration library

- Improves the performance of **Spark** and **Presto** jobs
- Uses Arrow columnar memory layouts for most data types
- Enables SIMD vectorized expression evaluation

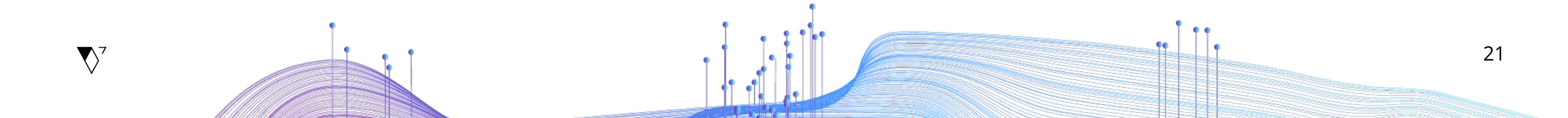
Source: https://github.com/facebookincubator/velox



Challenge: Embeddable Query Processing

- Enterprises want the flexibility to **embed in-memory analytical execution capabilities** directly in business applications.

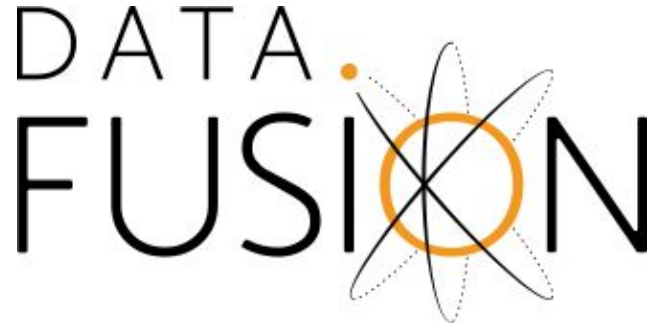
- By integrating with Arrow, embeddable engines can achieve excellent **performance**, **efficiency**, and **developer experience**.



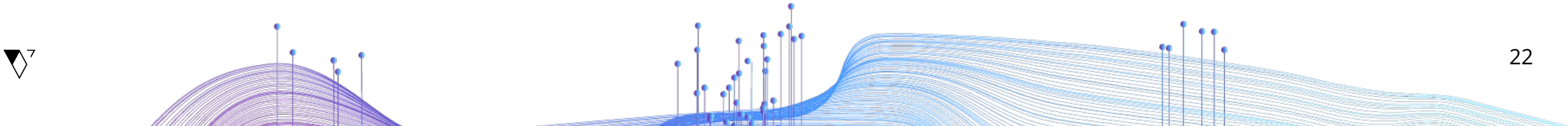
Case Study: DataFusion

Arrow DataFusion is extensible query execution framework

- Implemented as an **embeddable Rust library**
- Supports **distributed execution** through **Ballista**
- Uses **Arrow** as its **native in-memory format**
- Supports **SQL** and a **DataFrame API**
- Donated to the Apache Arrow project



Source: https://arrow.apache.org/datafusion

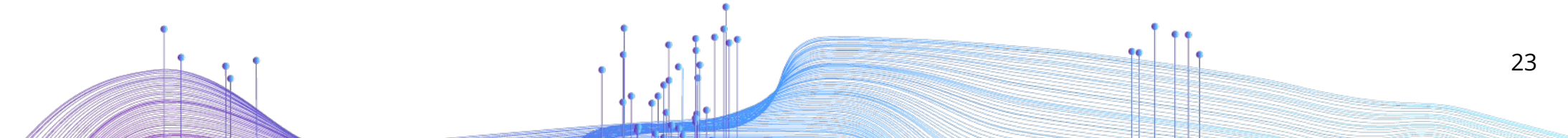


Case Study: DuckDB

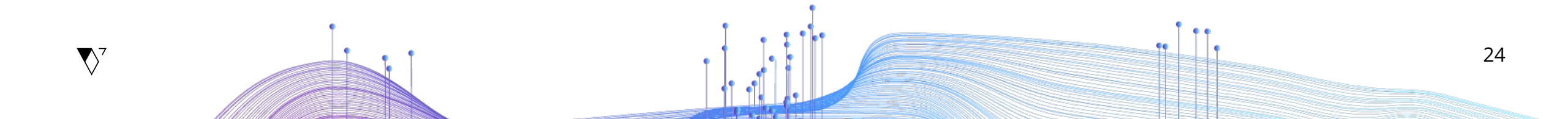
DuckDB is an in-process SQL OLAP database system

- Implemented as an **embeddable C++ library**
- Offers **zero-copy integration** with Arrow
- Can **push down filters and projections** into Arrow scans
- Interoperates with the Arrow **Python and R APIs**

Source: https://duckdb.org/2021/12/03/duck-arrow.html



How Voltron Data Supports Enterprise Applications of Apache Arrow



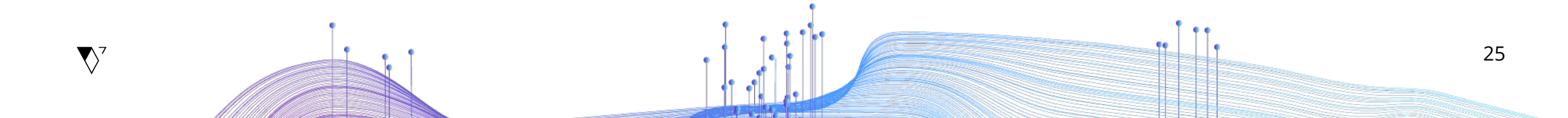
Enterprise Subscription for Arrow

- A focused set of services designed to **accelerate business success** with Apache Arrow
  - Professional support
  - Deployment assistance
  - Content and events
  - Private consultations
- Offered in **three editions** starting with **free**
- **Available now** (March 2022)
- Learn more and sign up at **voltrondata.com/subscription**

Trusted by:

RESEARCH

Open for Innovation KNIME



Thank you

Wes McKinney
@wesmckinn

Sign up:
voltrondata.com/subscription