#FIXNORDICS

NORDIC TRADING CONFERENCE 2025

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

FIX NORDIC TRADING CONFERENCE 2025

#FIXNORDICS

Orchestra in Action – Part 1

Community Tools and Core Concepts

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

FIX NORDIC TRADING CONFERENCE 2025

Orchestra in Action – Part 1

Community Tools and Core Concepts

Hanno Klein
FIX Technical Director & Global Technical Committee Co-Chair **FIX Trading Community**
Founder & Senior Standards Advisor
FIXdom

Patrick Lucas
Founder
Atomic Wire

Martin Swanson
Founder
Atomic Wire

Introduction to Atomic Wire

- A FinTech based in London and Berlin, we joined the FIX Trading Community in 2023 and are actively involved in the Orchestra subcommittee, contributing to FIX open-source projects.
- We developed community tools for Orchestra because they proved valuable for working with the modern data stack:
  - **Orchestra Build Tools** (with examples on GitHub)
  - **Orchestra Hub** (Alpha)
  - **Orchimate** (1.0)

Simple streaming query

“How many agency orders were executed in the past 1 hour?”

Answering even a simple question like this involves a lot of heavy lifting

- Diverse encodings
- User customization
- Ambiguity
- Versioning & compatibility
- Interoperability
- Ecosystem

Orchestra offers features that let us address these challenges

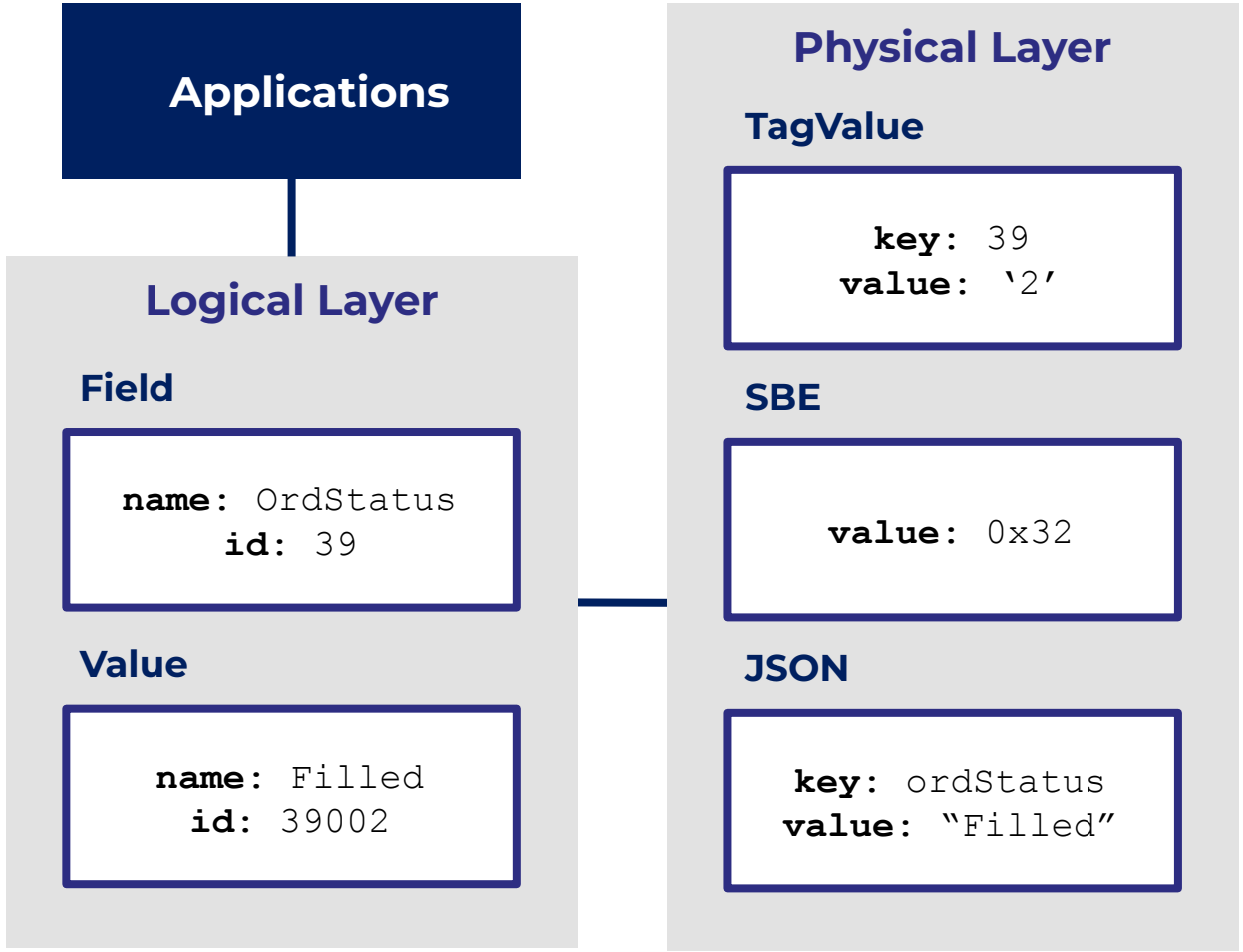
- Encoding abstraction
- Semantic inheritance (custom specifications)
- Message variants (scenarios)
- Append-only versioning (pedigree)
- Multi-protocol capability

> **INFO**
>
> Orchestra is more than just an interface definition language—it provides a framework for building **encoding-agnostic applications** and **evolving data** in a controlled manner.

Encoding abstraction separates the underlying data encoding formats from the application logic

- Data dictionary elements are defined **independently** from over-the-wire formats.
- Each element has a logical `name` and `id` **mapped** to physical encodings.

> **TIP**
>
> Now we can build **generic applications** that support any encoding



Semantic inheritance allows the creation of custom specifications while maintaining consistency

- All data dictionary elements are **uniquely identifiable**.
- Reference specifications, such as FIX, offer a global data dictionary for a specific namespace.
- Users can create **custom specifications** that inherit elements from reference specifications.
- This provides a formal mechanism for ensuring **semantic equivalence** across specifications, reducing duplication, and maintaining consistency.

Message variants remove ambiguity and allow us to tailor specifications for specific use cases

- FIX messages can serve multiple purposes
  - e.g. ExecutionReport has at least seven uses
- A variant is a **restricted version** of a global message definition, tailored for specific use cases.
- Implemented in Orchestra as **scenarios**, which are applicable to any type of data dictionary element
  - e.g. Instrument component per asset class
- Eliminates ambiguity and can reduce complexity in the application layer.

Append-only versioning lets us track, evolve, and maintain data compatibility

- Maintain **version history** in a single specification
- Add, update, or deprecate data elements without deleting or reassigning unique `name` or `id`.
- Enables **data compatibility checking**
- Simplifies upgrades and lets you **evolve data** in a controlled manner

Multi-protocol capability provides a consistent approach and enhances interoperability

- Orchestra is **not restricted to FIX**; it offers a flexible approach for any data protocol, for instance:
  - Standard FIX and FIX customizations
  - Native market data feeds (binary)
  - FINRA CAT reporting (JSON)
- **ISO 20022** base specification is in development
- Internal **proprietary** formats can also be supported
- Provides a **generalizable** approach for any data dictionary-based protocol

An ecosystem of community tools to build, discover and explore Orchestra specifications

- **Orchestra Build Tools**
  - Integrate Orchestra into your CI/CD workflows
  - Uses FIX open source CLIs where possible
  - Generate build artefacts (docs, encoding schemas, code)
  - Lots of executable examples available on GitHub
- **Orchestra Hub**
  - Central repository for discovering specs
  - Access specs programmatically (simplify upgrades)
  - Supports versioning
- **Orchimate**
  - Search and explore Orchestra specs
  - Dynamically load your own specs

#FIXNORDICS

Orchestra in Action – Part 2

Real-World Applications and Use Cases

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

FIX NORDIC TRADING CONFERENCE 2025

Orchestra in Action – Part 2

Real-World Applications and Use Cases

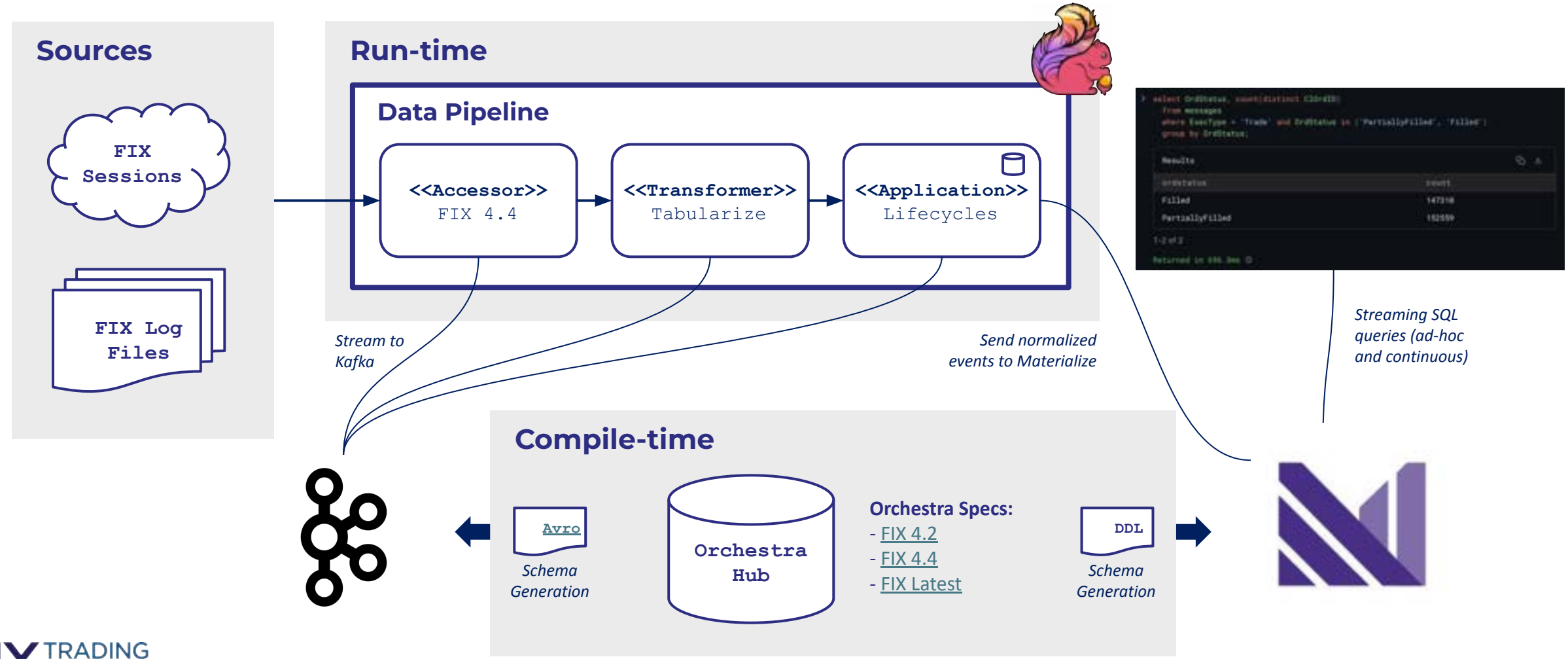
Hanno Klein
FIX Technical Director & Global Technical Committee Co-Chair **FIX Trading Community**
Founder & Senior Standards Advisor
FIXdom

Patrick Lucas
Founder
Atomic Wire

Martin Swanson
Founder
Atomic Wire

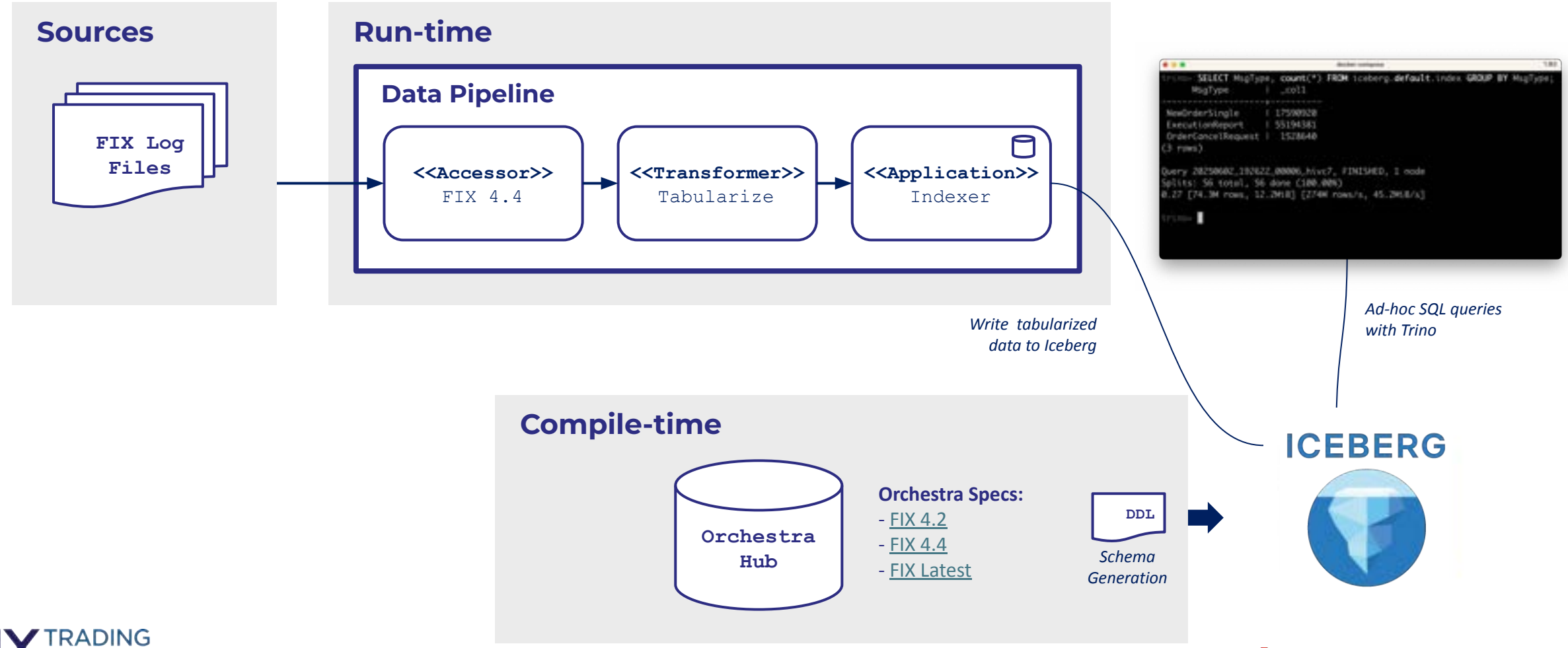
Example #1: FIX SQL

Real-time streaming SQL queries over FIX data streams



Example #2: FIX SQL

Data analytics over FIX archives with Apache Iceberg

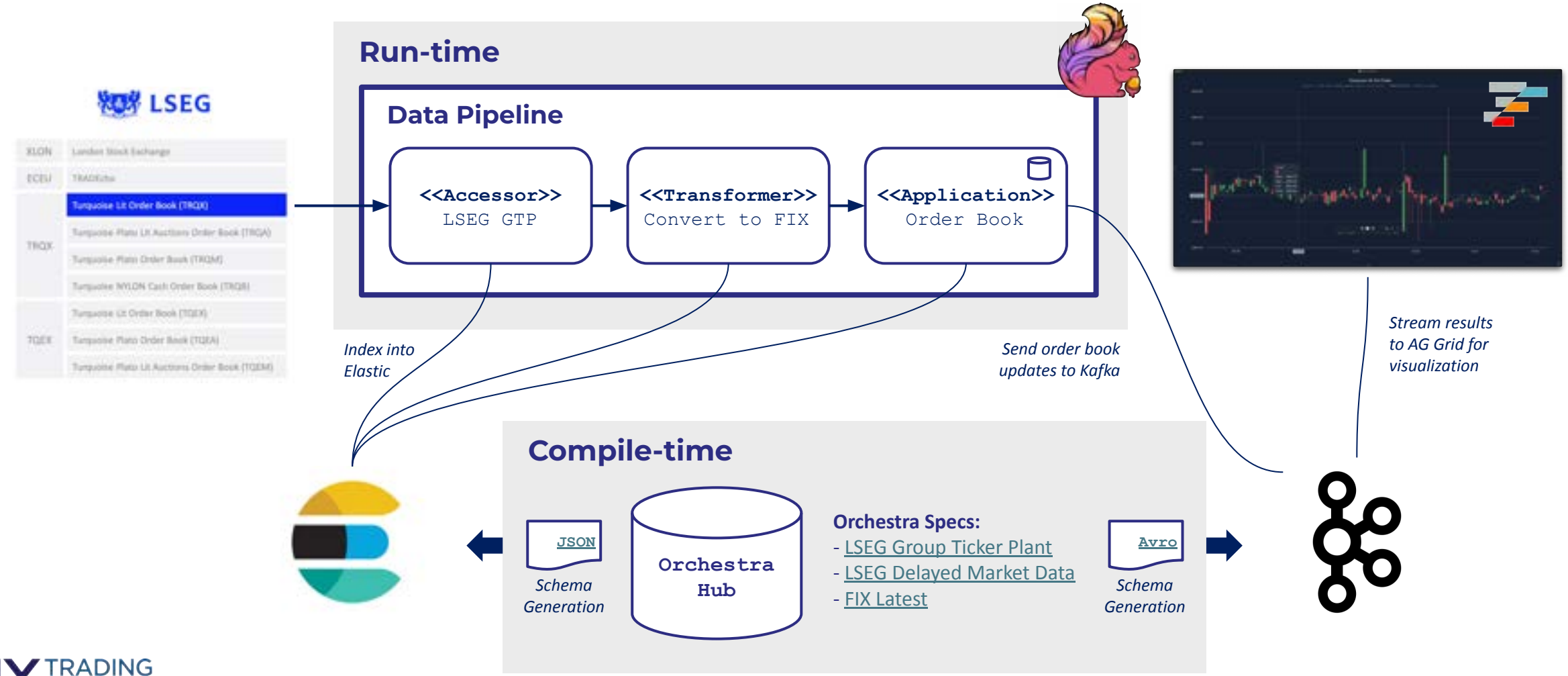


FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

atomicwire

Example #3: Order Book Updates

Real-time order book state computation with LSEG market data



atomicwire

Q&A

atomicwire.io/contact

FIX TRADING COMMUNITY™
INDUSTRY-DRIVEN • INDEPENDENT • NEUTRAL

FIX NORDIC TRADING CONFERENCE 2025