Building Composable Data Microservices with Apache Arrow

Bloomberg

Engineering

Community Over Code 2023
October 7, 2023

Peter Leicht, Engineering Team Lead
Shoumyo Chakravorti, Software Engineer
Bloomberg Indices Engineering

TechAtBloomberg.com

Who are we? (Bloomberg)

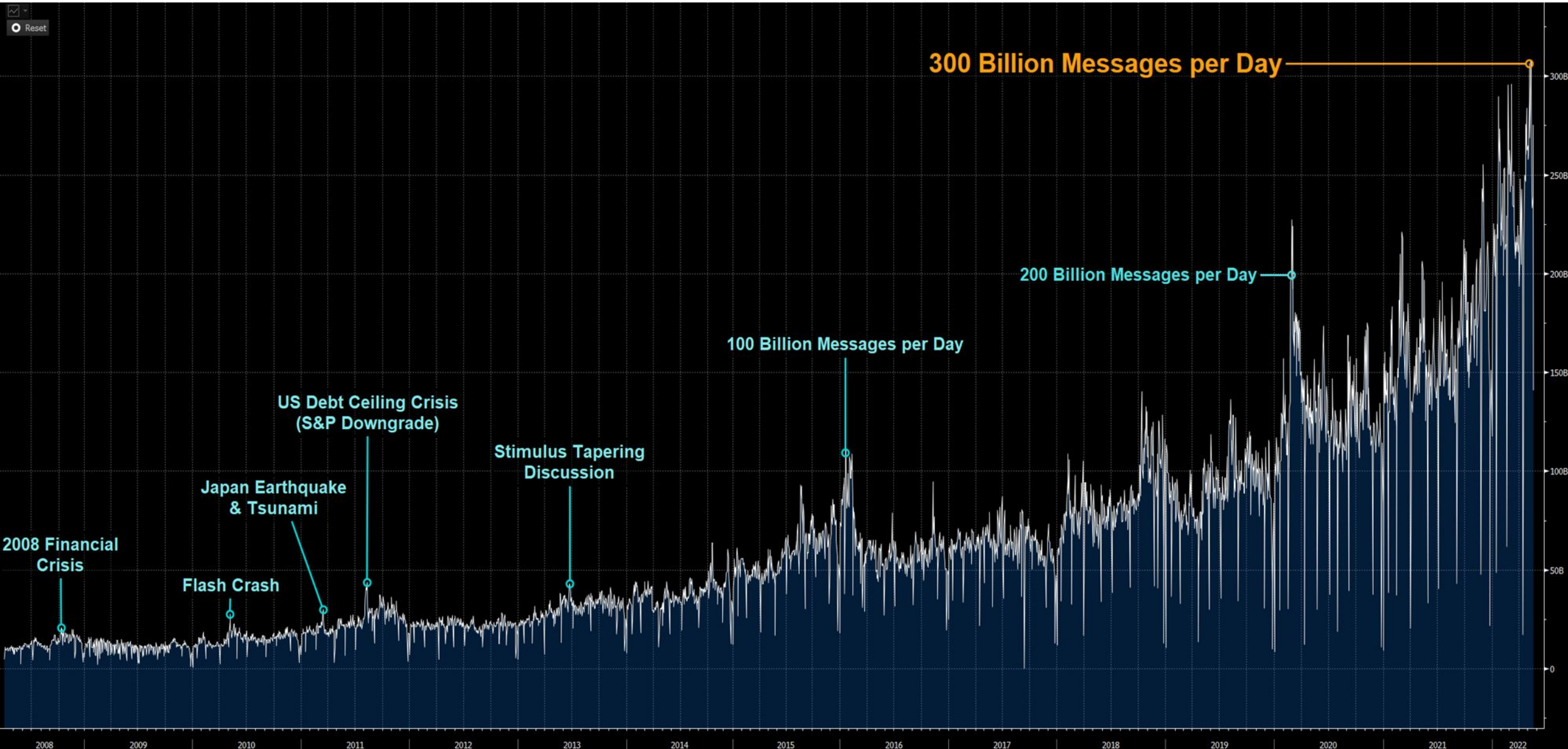
- 8,000+ software engineers
- Product areas:
  - **D**ata
  - **A**nalytics
  - **N**ews
  - **C**ommunication
  - **E**lectronic trading
- Lots of teams who need to process large datasets and publish analytics

TechAtBloomberg.com

Bloomberg

Engineering

Exponential Growth in Market Data Ingestion



Who are we? (Bloomberg Indices)

1. Ingest bond data

Daily bond data is acquired from various internal and external data sources into our data stores, after which it is cleaned, massaged, and enriched.

2. Compute indices

For each index, identify the members and their relative weights in the index, and compute index-level statistics.

3. Validate results

Product/Data/Operations validates any potentially anomalous data points.

4. Publish to clients

Once the data has been validated, the indices are published to the respective clients.

TechAtBloomberg.com

Bloomberg

Engineering

Who are we? (Bloomberg Indices)

- Fixed Income Indices
  - $50 Trillion dollar market value
  - Batch processing
  - Tight SLAs – clients are expecting reports ASAP

- Data scale
  - 200K+ bonds (rows), 1000s of columns of data per day
  - 25K+ indices produced each day
  - Up to 60-70K constituents (rows) per index

Bloomberg

Engineering

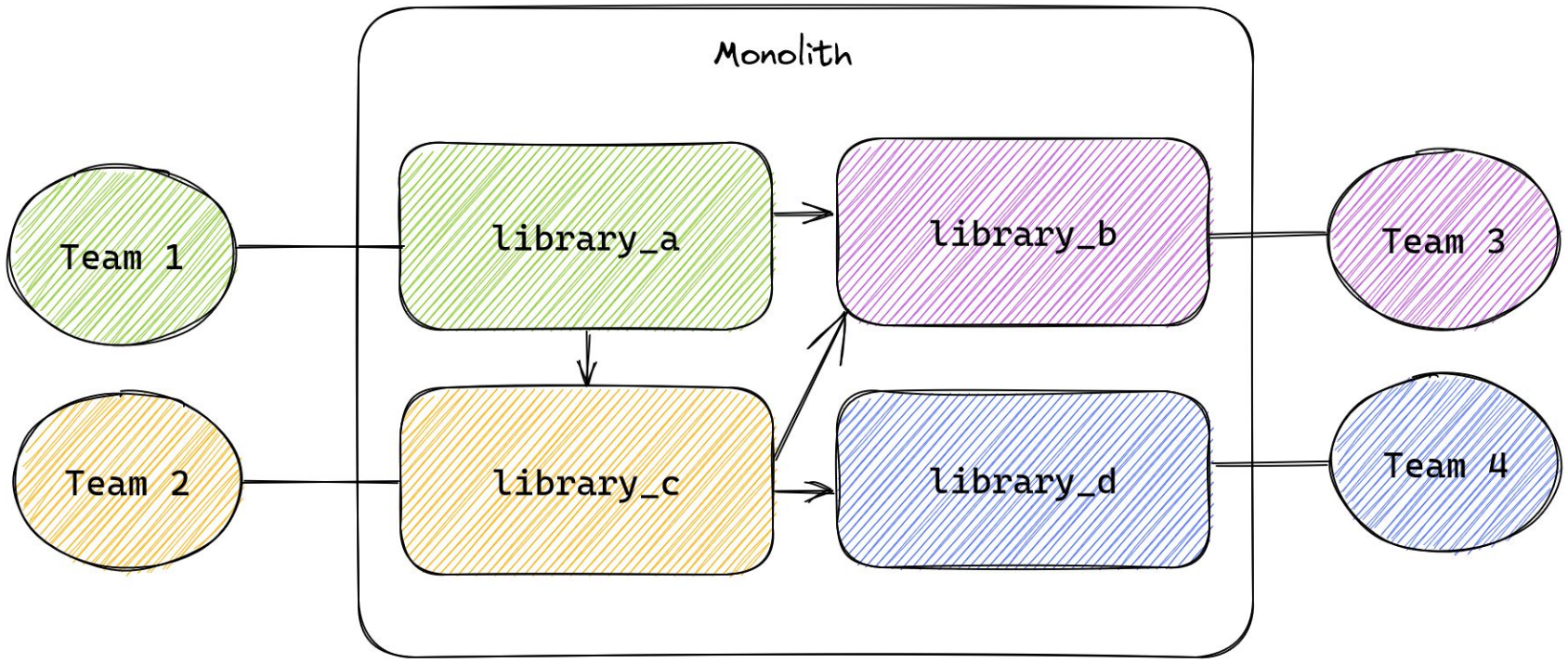
Why do we care about microservices?

TechAtBloomberg.com

Bloomberg

Engineering

Where did we come from?



TechAtBloomberg.com

Bloomberg

Engineering

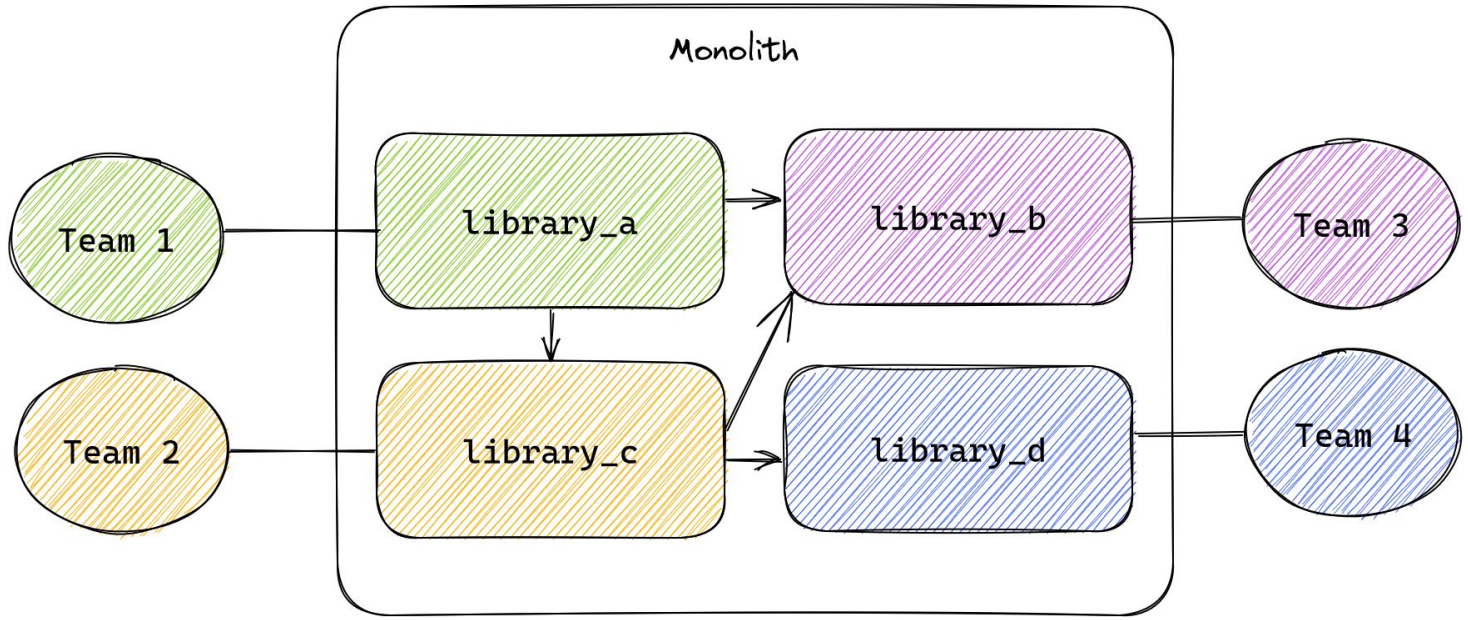
Monolith: Pros & Cons

😄 Pros

- Performance
- Programming practices

😔 Cons

- Coupling
- Impact radius
- Ownership
  - Who owns the Monolith?
- Release cadence
- Choice of languages
- Build processes

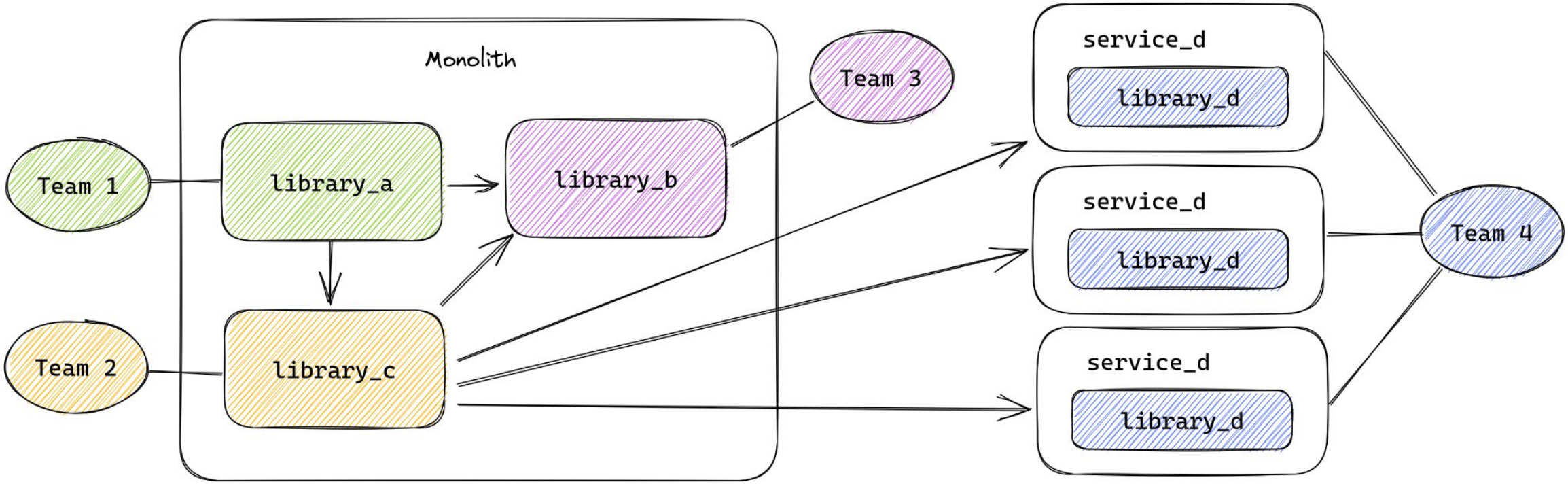


TechAtBloomberg.com

Bloomberg

Engineering

Splitting things out into services



TechAtBloomberg.com

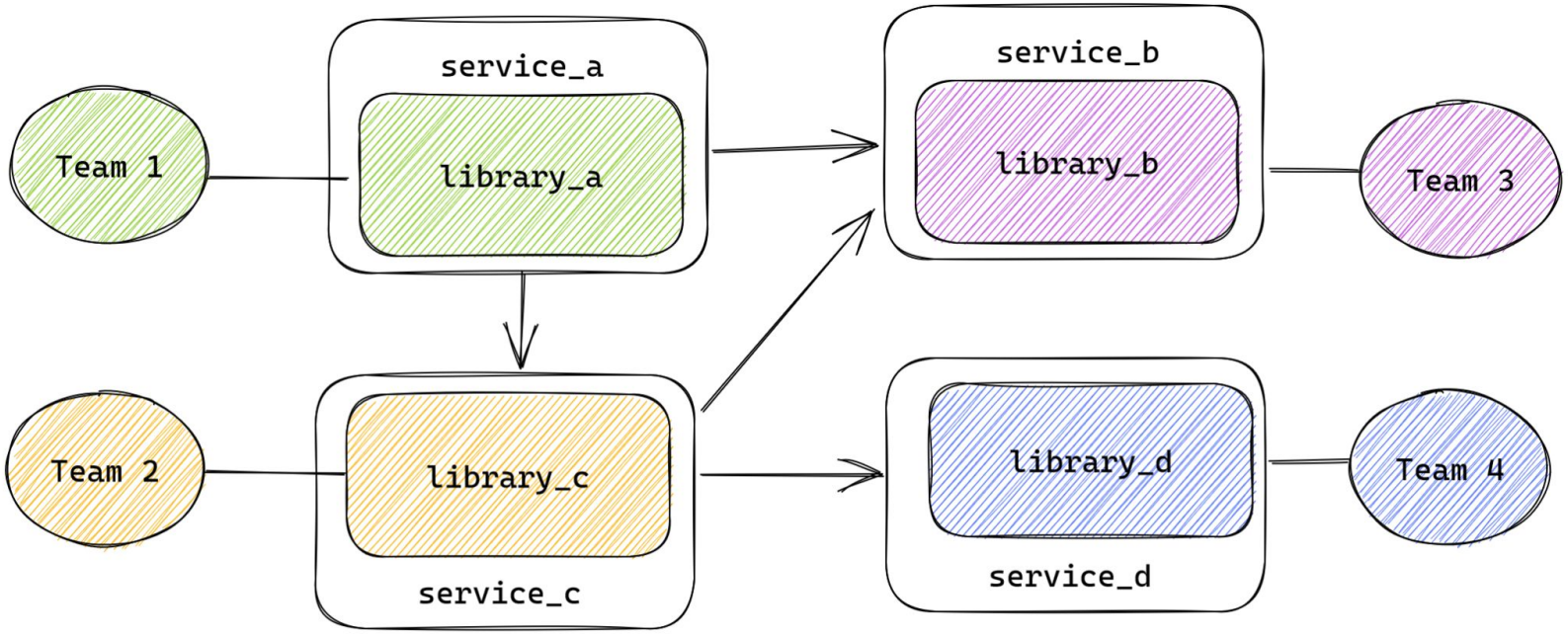
Bloomberg

Engineering

Going all the way

😄 Pros

- Decoupled
- Impact radius
- Ownership
- Release cadences
- Choice of language
- Build processes



TechAtBloomberg.com

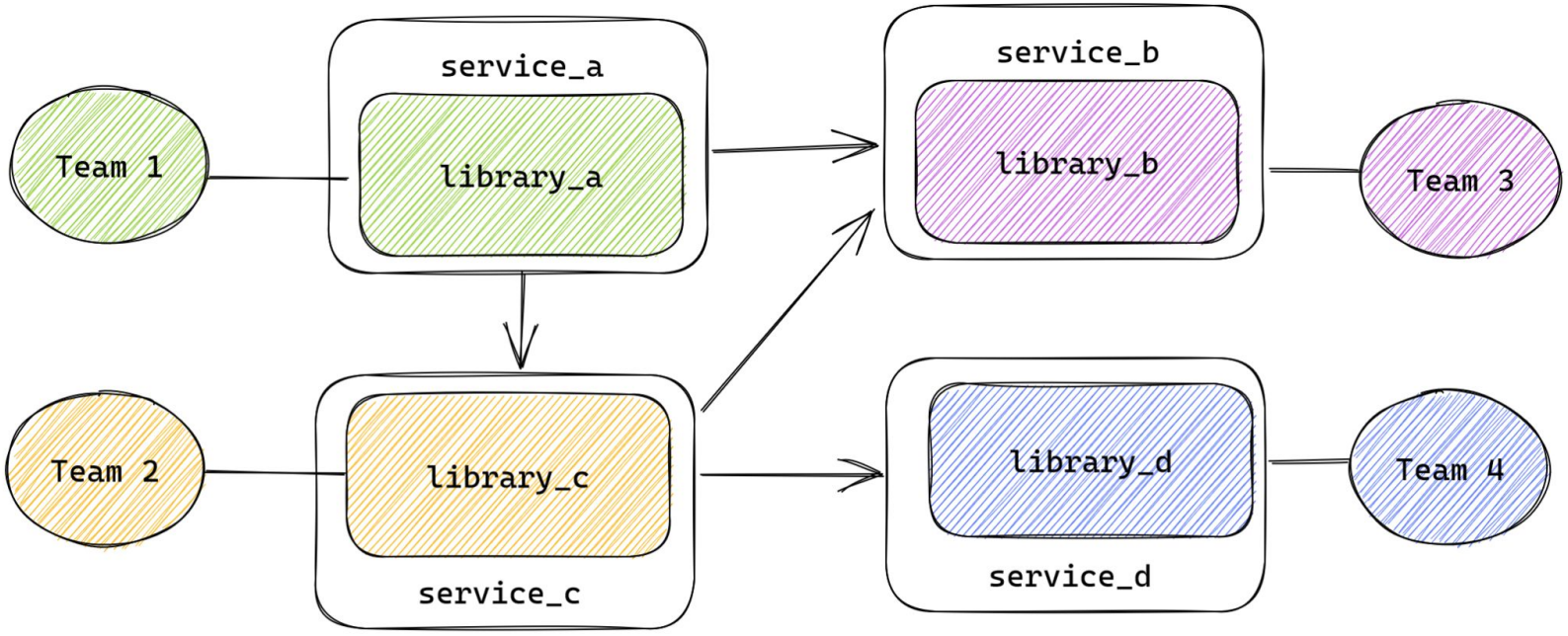
Bloomberg

Engineering

New complexities…

😔 Cons

- Performance
- Development costs



Bloomberg

Engineering

Problem 1: Performance

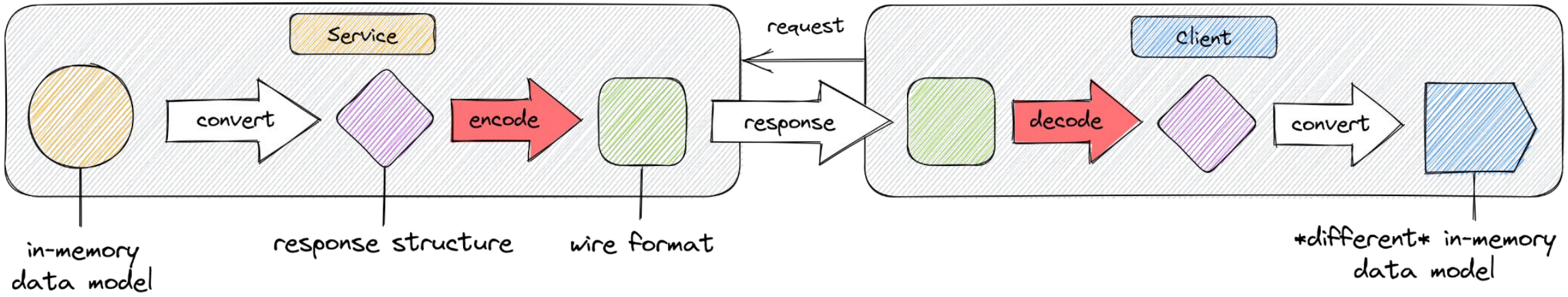
TechAtBloomberg.com

Bloomberg

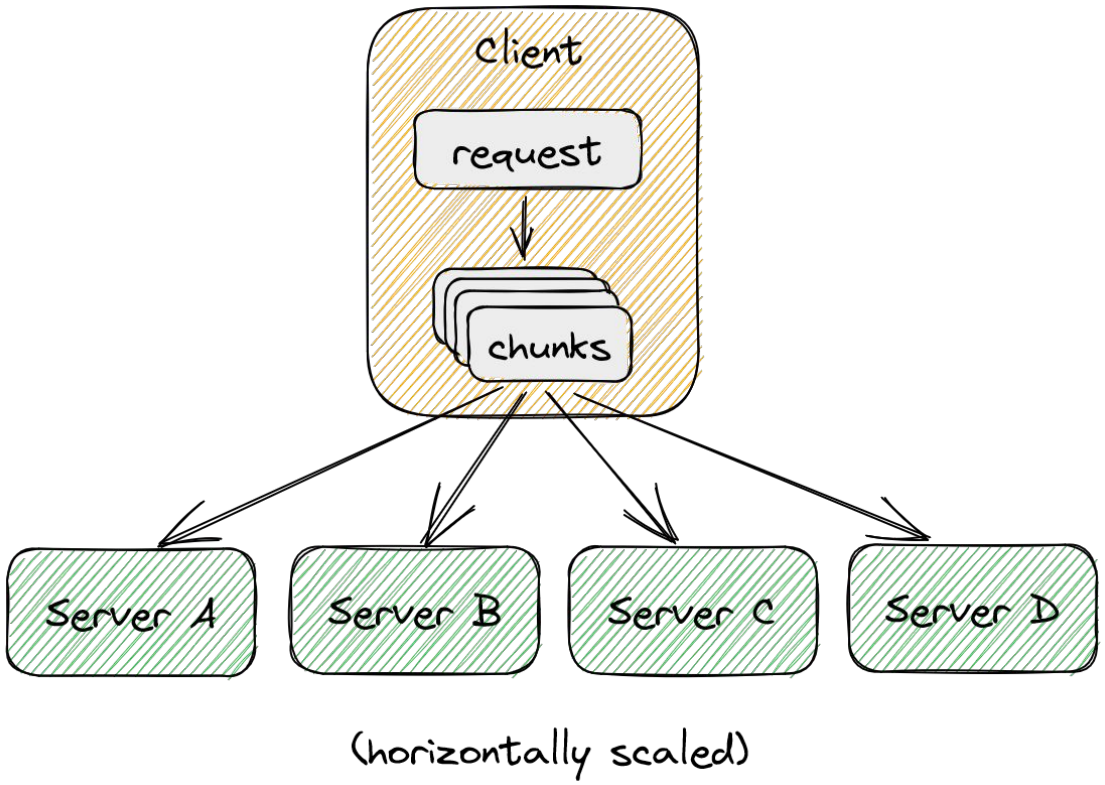
Engineering

Problem: CPU encoding and decoding

- CPU time spent encoding/decoding data for large requests/responses (~64MB)
  - C++ to C++: 50%
  - C++ to Python: 90%



Solution: Scaling out?



😄 Pros

- Latency

😔 Cons

- Band-Aid solution
  - Encode / decode still exists
  - Resource wastage
- Operational complexity

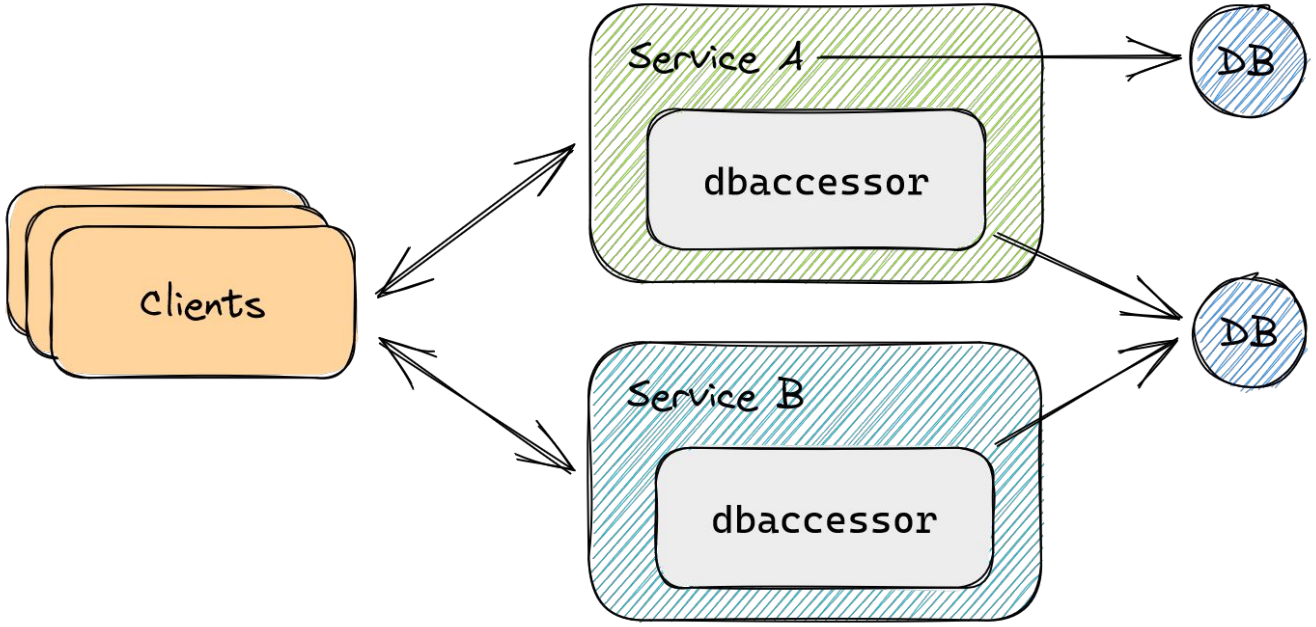
Solution: Libraries?

😄 Pros

- Latency
- No encoding/decoding

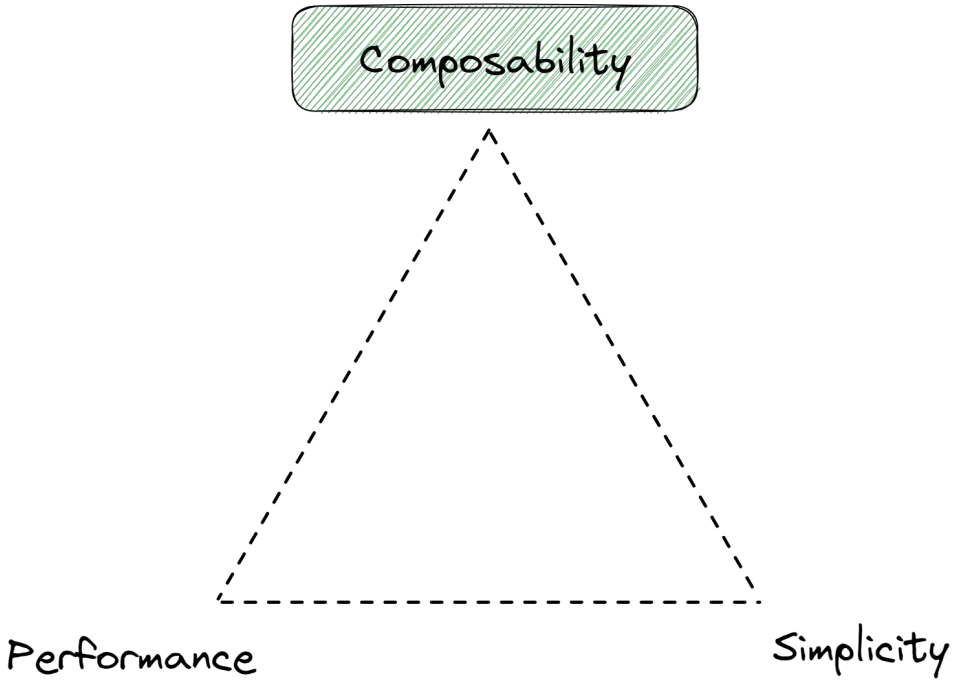
😔 Cons

- Becoming a monolith
- Multiple versions
- Access control



The “composability theorem”

- Must have composability
- Trading simplicity for performance



TechAtBloomberg.com

Bloomberg

Engineering

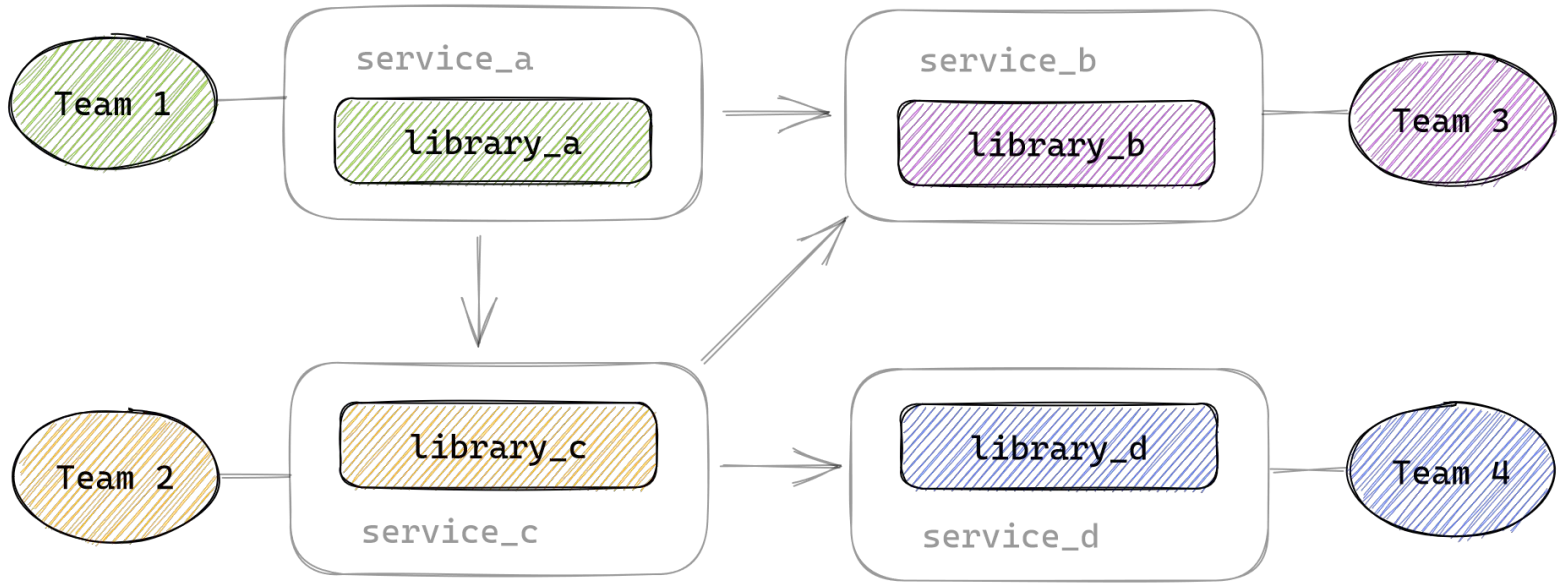
Problem 2: Fragmentation

TechAtBloomberg.com

Bloomberg

Engineering

Fragmentation is inevitable



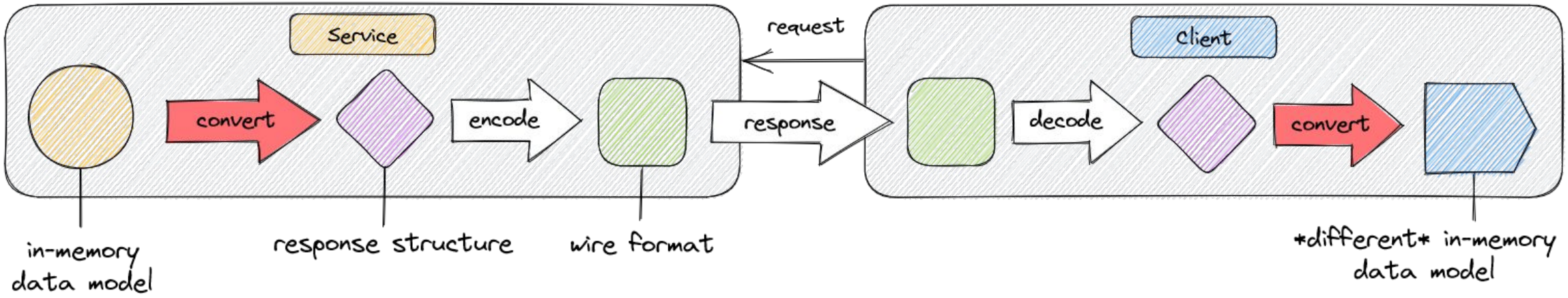
TechAtBloomberg.com

Bloomberg

Engineering

Problem: Lack of standardization

- Every service defines its own schema and internal data model
- Lots of time spent **designing schemas** for new systems
- Lots of time spent **connecting systems**



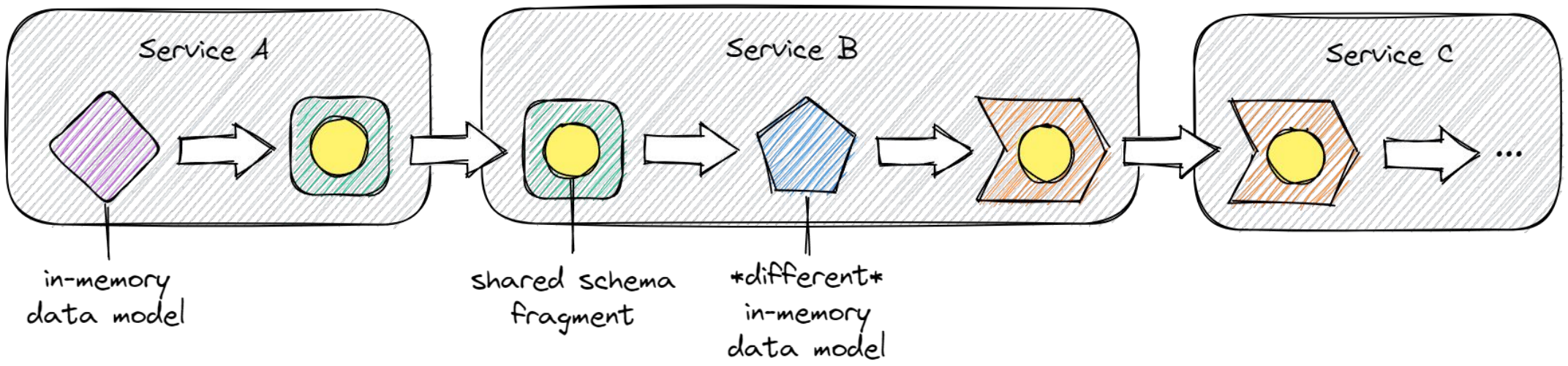
Solution: Shared data model?

😄 Pros

- Some consistency across service schemas

😔 Cons

- Increased schema complexity
  - Variant/union types
- In-memory format != on-wire format



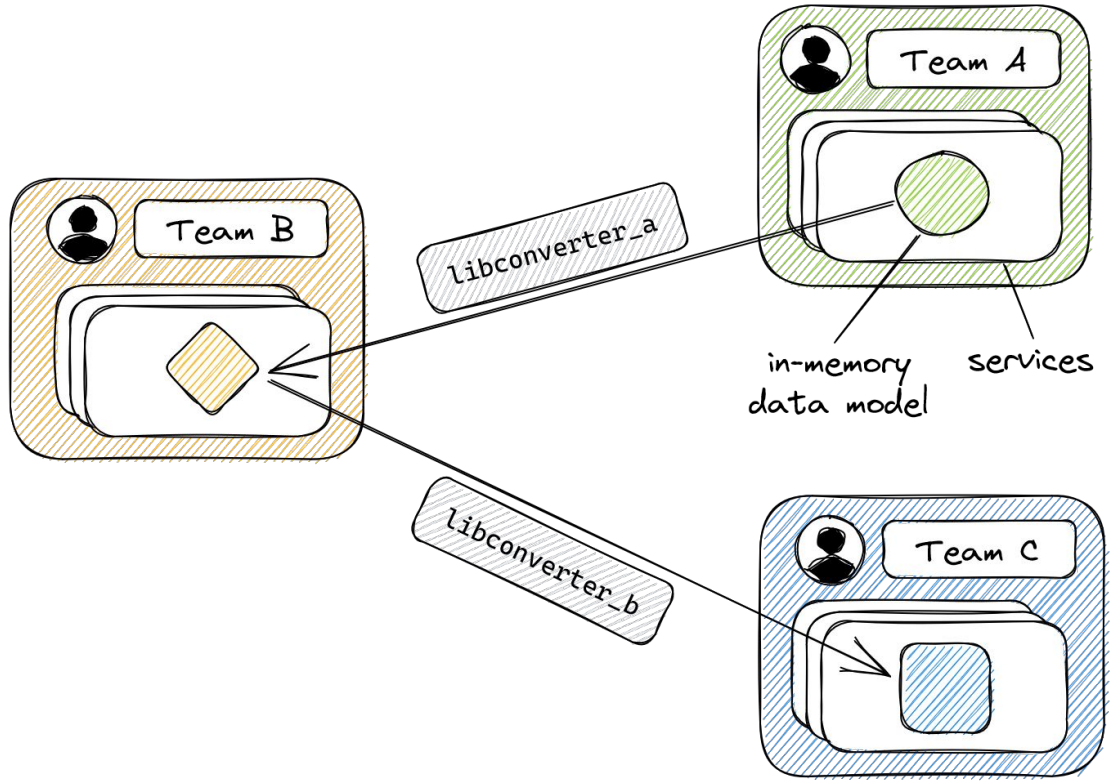
Solution: Organic localized standardization

😄 Pros

- *Some* reusability
- *Overall* less fragmentation

😔 Cons

- Local optimum
- Performance problems unsolved



In Summary

	Monolith	Services only	Services + Libraries	Services + shared data models and converters
Performance	✔️	❌	✔️	❌
Isolation	❌	✔️	⚠️	✔️
Maintainability	❌	❌	❌	⚠️

TechAtBloomberg.com

Bloomberg

Engineering

A theme emerges

- A good solution to composability would:
  - Standardize both in-memory and on-wire formats
  - Reduce the cost of integration
  - Not add unnecessary complexity
  - Not add significant performance penalties

In the meantime…

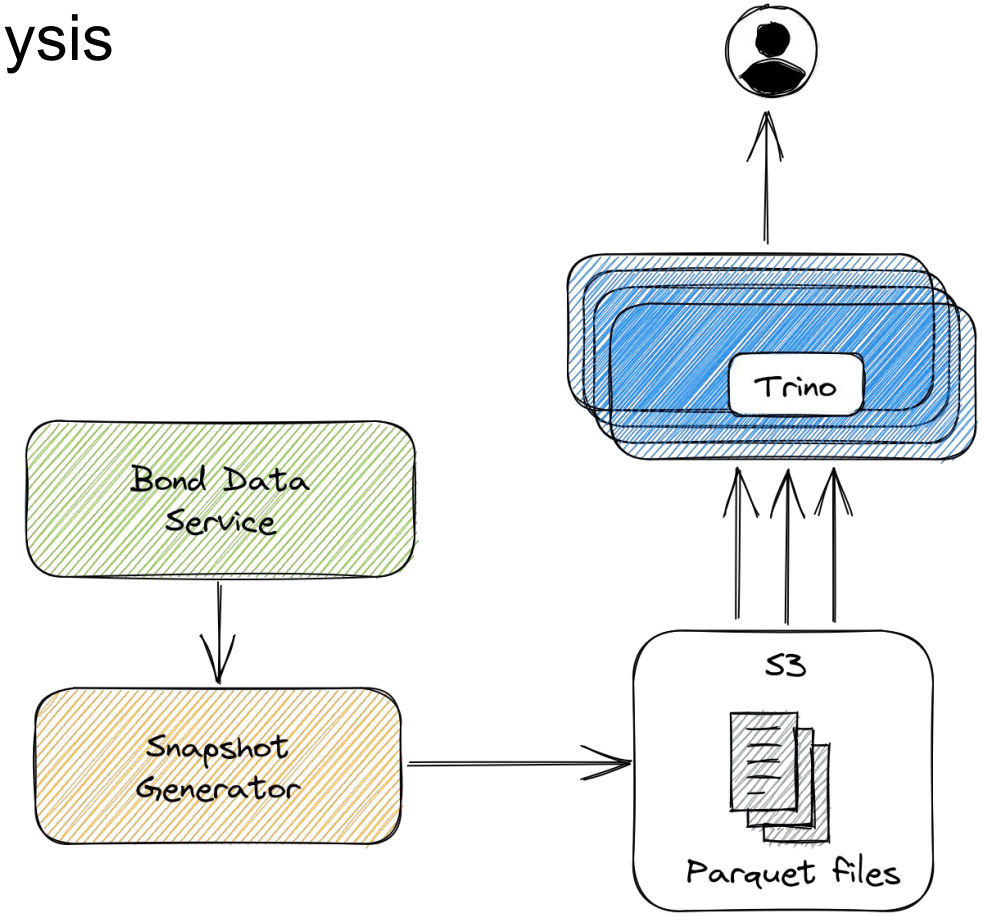
TechAtBloomberg.com

Bloomberg

Engineering

Offline analysis

- **Goal**: Using daily bond data for offline analysis
  - Compliance / troubleshooting
  - Product ideation
- Apache Parquet for analytics with Trino
- Apache Arrow as an intermediate format



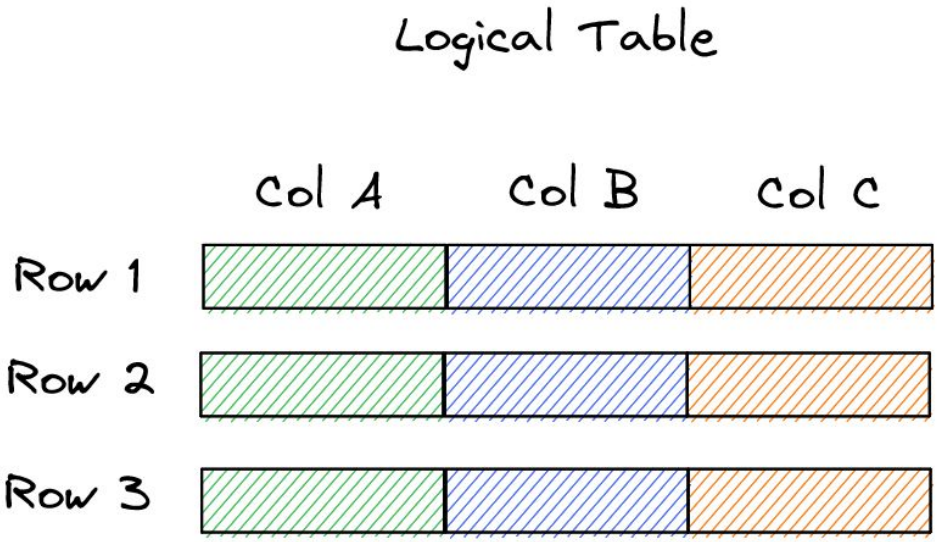
TechAtBloomberg.com

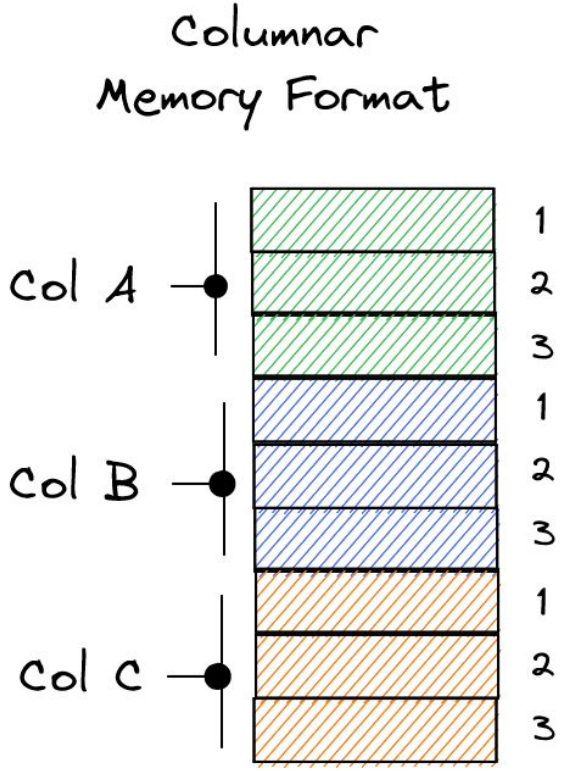
Bloomberg

Engineering

What is Apache Arrow?

- Tabular
- Columnar
- Fast data transfer
- IPC ≈ In-memory
- Cache friendly
- SIMD friendly





TechAtBloomberg.com

Bloomberg

Engineering

Example uses of Apache Arrow

Data Transfer

Compute Engines

Visualization

TechAtBloomberg.com

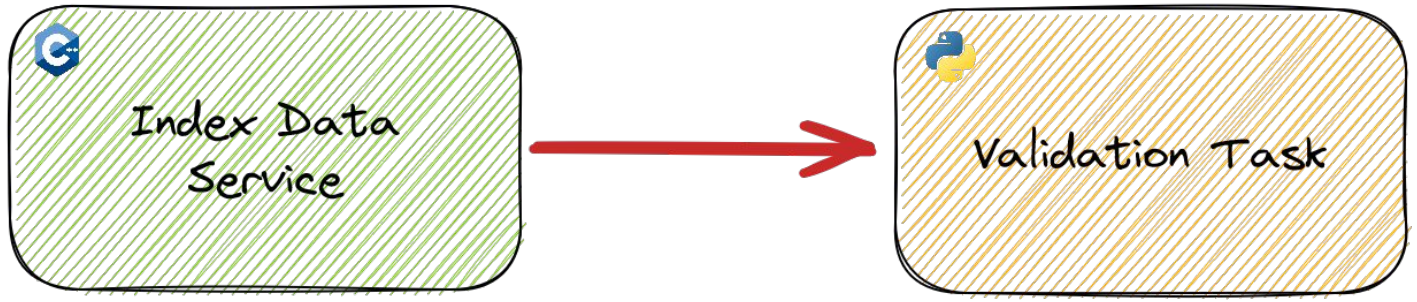
Bloomberg

Engineering

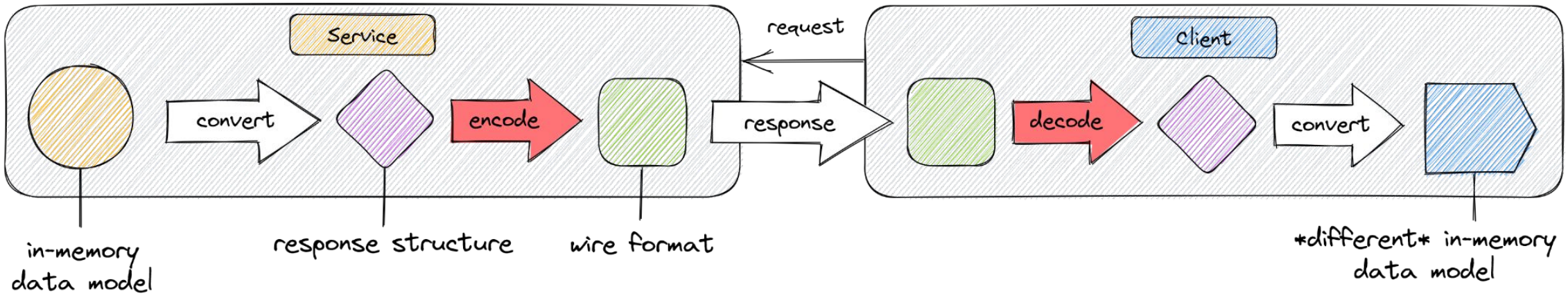
New use-case

- Python application for data validation / reconciliation

- Python app experiencing significant slowness
  - Culprit: **ser/de** was taking **90%** of the time!
  - Brute force was not an option…



Revisiting the status quo

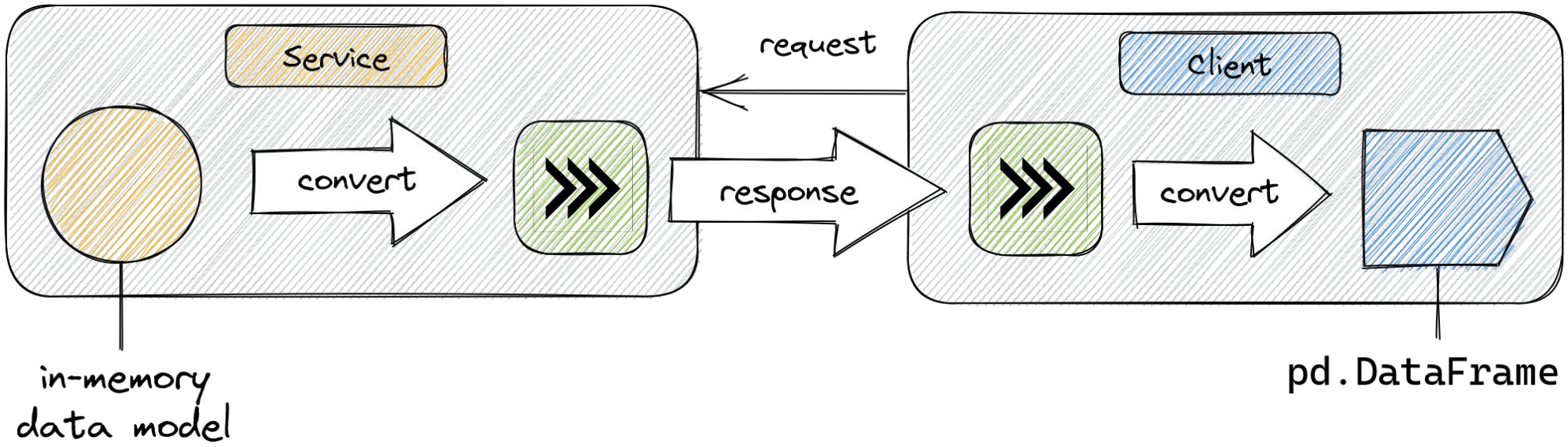


TechAtBloomberg.com

Bloomberg

Engineering

How about Arrow?



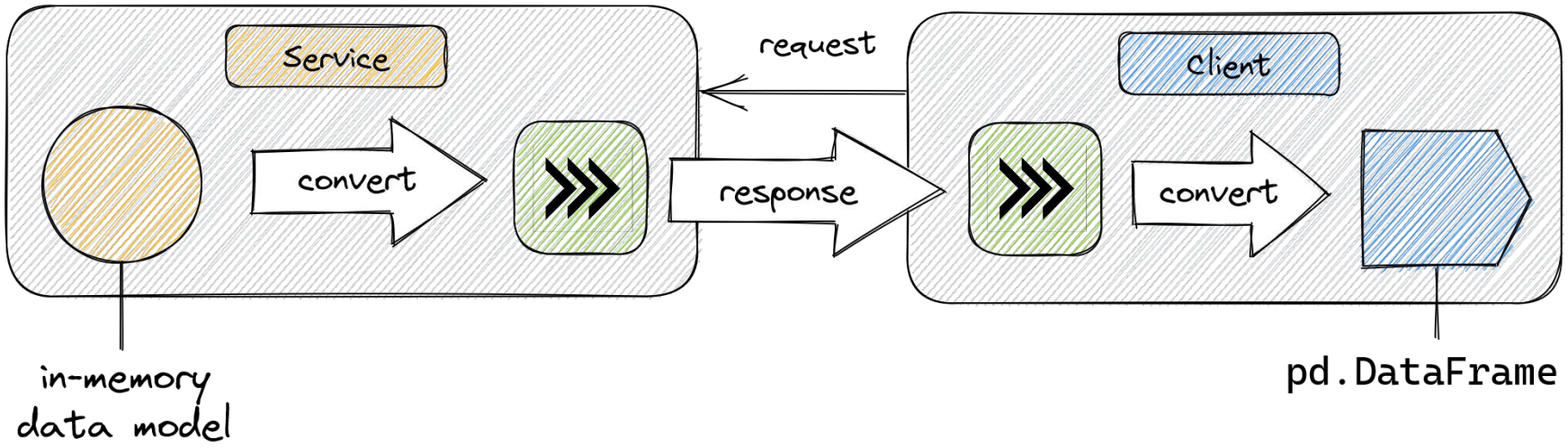
TechAtBloomberg.com

Bloomberg

Engineering

How did we do this?

- Integrate Arrow IPC with middleware
  - e.g., Arrow Flight
- Got back almost all of the **90%** ser/de time
- Just use pandas!



Bloomberg

Engineering

Adoption: slow but steady

- Headline features (in Indices)
  - For us, data transport is no longer a bottleneck!
  - Interoperability with pandas

- Growing usage across Bloomberg
  - In-memory data format for custom **analytics** engines
  - Faster **data transport** across applications

Can Apache Arrow solve our problems?

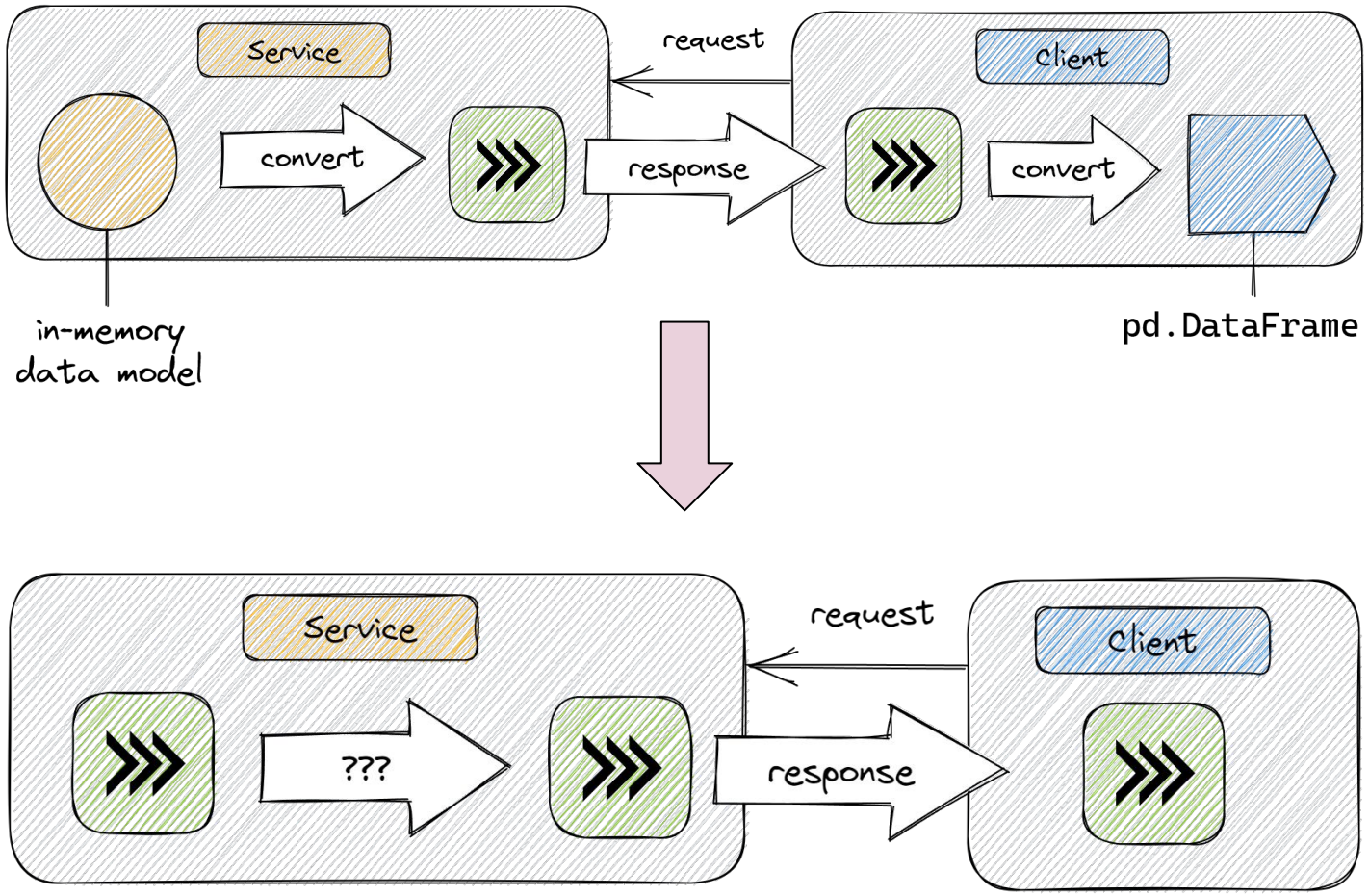
- Standardizes both in-memory and on-wire formats ✔️
- Does not add significant performance penalties ✔️
- Reduces the cost of integration? 🤔
- Does not add unnecessary complexity? 🤔

TechAtBloomberg.com

Bloomberg

Engineering

Taking it a step further



Bloomberg

Engineering

Kinds of business logic

- Enrichment
- Aggregation
- Validation
- Searching

bonds

SecId	Currency	Price
BBG001	USD	2.00
BBG002	GBP	9.50
BBG003	JPY	954.30
BBG004	USD	14.20

fx_rates

Currency	Rate
USD	1.00
JPY	149.30
GBP	0.83

bonds'

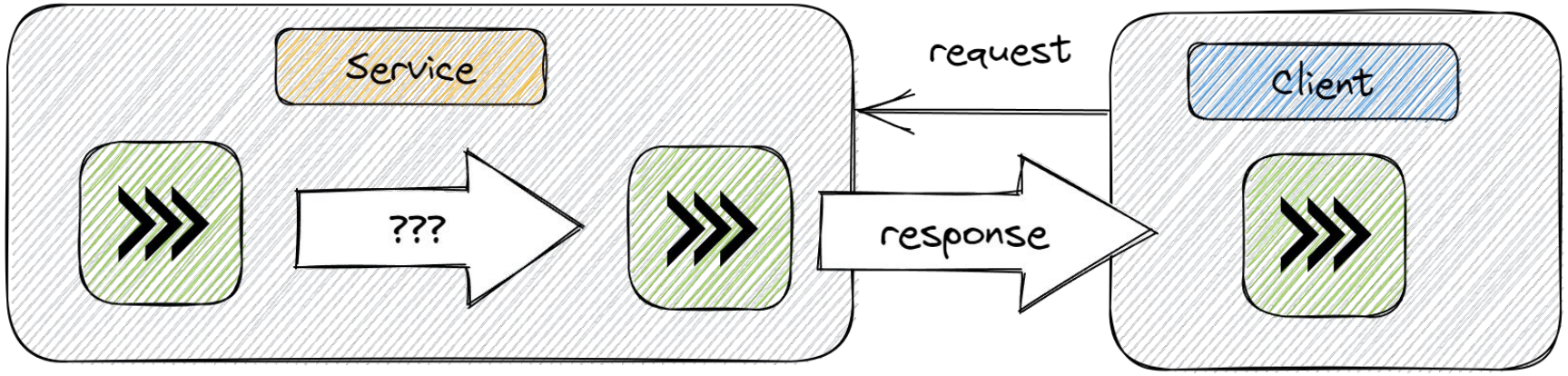
SecId	Price'
BBG001	2.00
BBG002	11.45
BBG003	6.39
BBG004	14.20

TechAtBloomberg.com

Bloomberg

Engineering

How can we do it on Apache Arrow?



In-Memory Analytics

TechAtBloomberg.com

Bloomberg

Engineering

Compute Engines

- In-process library
- Declarative
- Query Optimization
- Computation
  - Vectorization vs. JIT
- Some are Arrow-native

DATA FUSION

Operations supported by compute engines

- Filter
- Project
- Aggregate
- Join
- Sort
- Window functions
- Pivot tables
- Unnest
- …

TechAtBloomberg.com

Bloomberg

Engineering

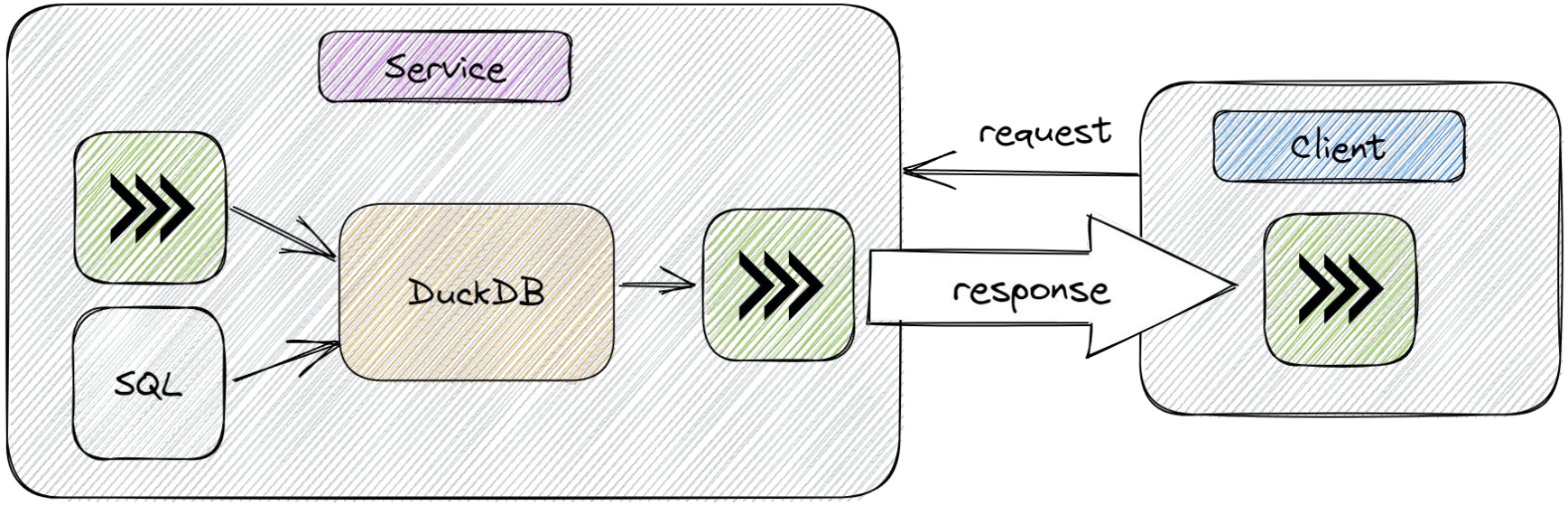
Representing our business logic

- Can we represent our logic *declaratively*?
- Enrichment → **Join / Project**
- Aggregation → **GroupBy / Agg**
- Validation → **Project / Filter**
- Searching → **Filter**

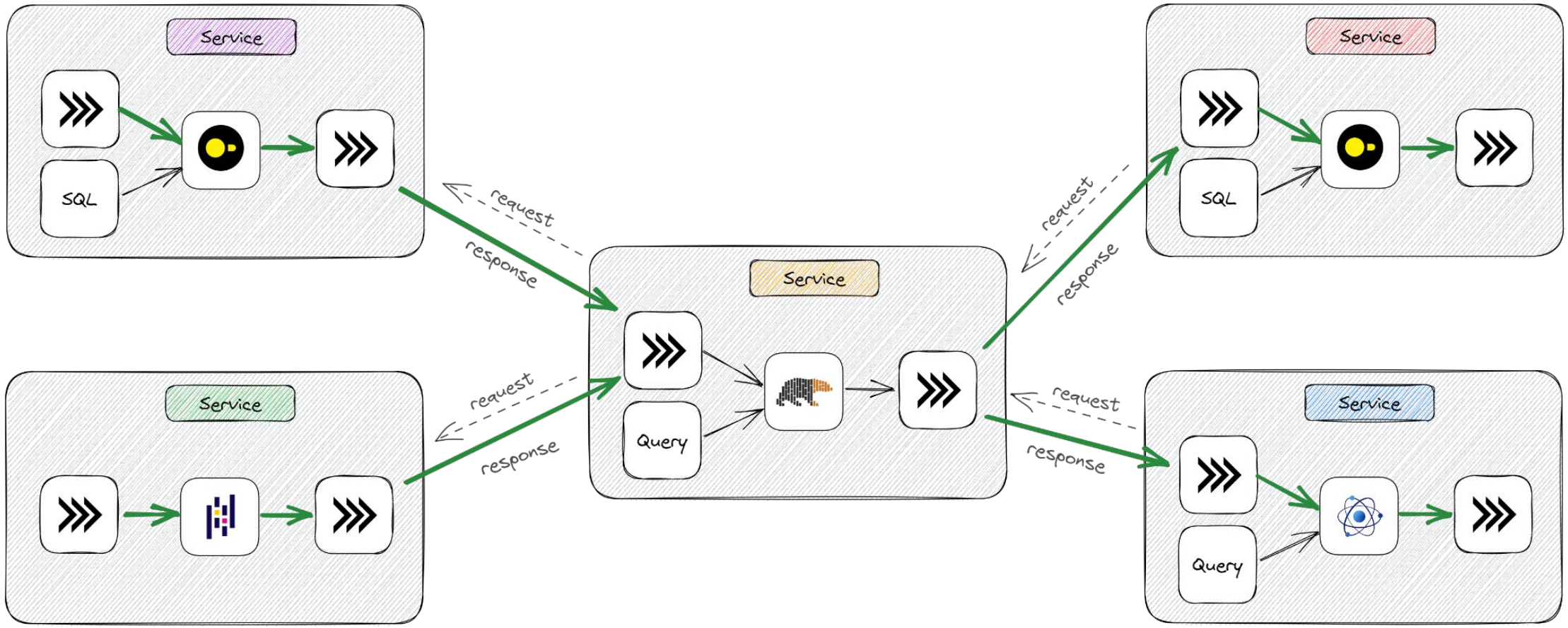
```
SELECT
    b.*,
   (b.Price * fx.Rate) AS NormalizedPrice
FROM bonds b
LEFT JOIN fx_rates fx ON b.Currency == fx.Currency
```


Adopting DuckDB

- C++ API + query optimizer + vectorized engine
- Supports Arrow input/output
- Observed **order of magnitude** performance improvement



Achieving greater composability



TechAtBloomberg.com

Bloomberg

Engineering

Learnings

TechAtBloomberg.com

Bloomberg

Engineering

Economies of scale

- Lower amortized cost to build new tooling
  - A **standard** is the foundation for generalizability
- e.g., Arrow integration with custom middleware
  - No need to encode and decode anymore
  - Every new user gets the tooling for free
  - Every new user encourages other callers to onboard as well
- Lots of newer open-source tooling built around Arrow
  - Driving developer costs down
- Performance gains open up additional design options

Learning from others

- Community-building helped us standardize and build the vision
- Couldn’t have thought of all use-cases in isolation
- Alignment across teams is necessary to standardize effectively

TechAtBloomberg.com

Bloomberg

Engineering

A shift in mindset

- Arrow is not…
  - *Just* an in-memory format or…
  - *Just* an on-the-wire format

- It's **both**
  - Selectively using Arrow for specific problems diminishes its value
    - e.g., Using Arrow for just data transport but not in-memory analytics
    - Still have to pay developer costs for implementing connector logic

Thank you!

We are hiring:

https://www.bloomberg.com/careers

Contact us:

- Peter Leicht: pleicht@bloomberg.net
- Shoumyo Chakravorti: schakravorti@bloomberg.net

Bloomberg

Engineering

TechAtBloomberg.com