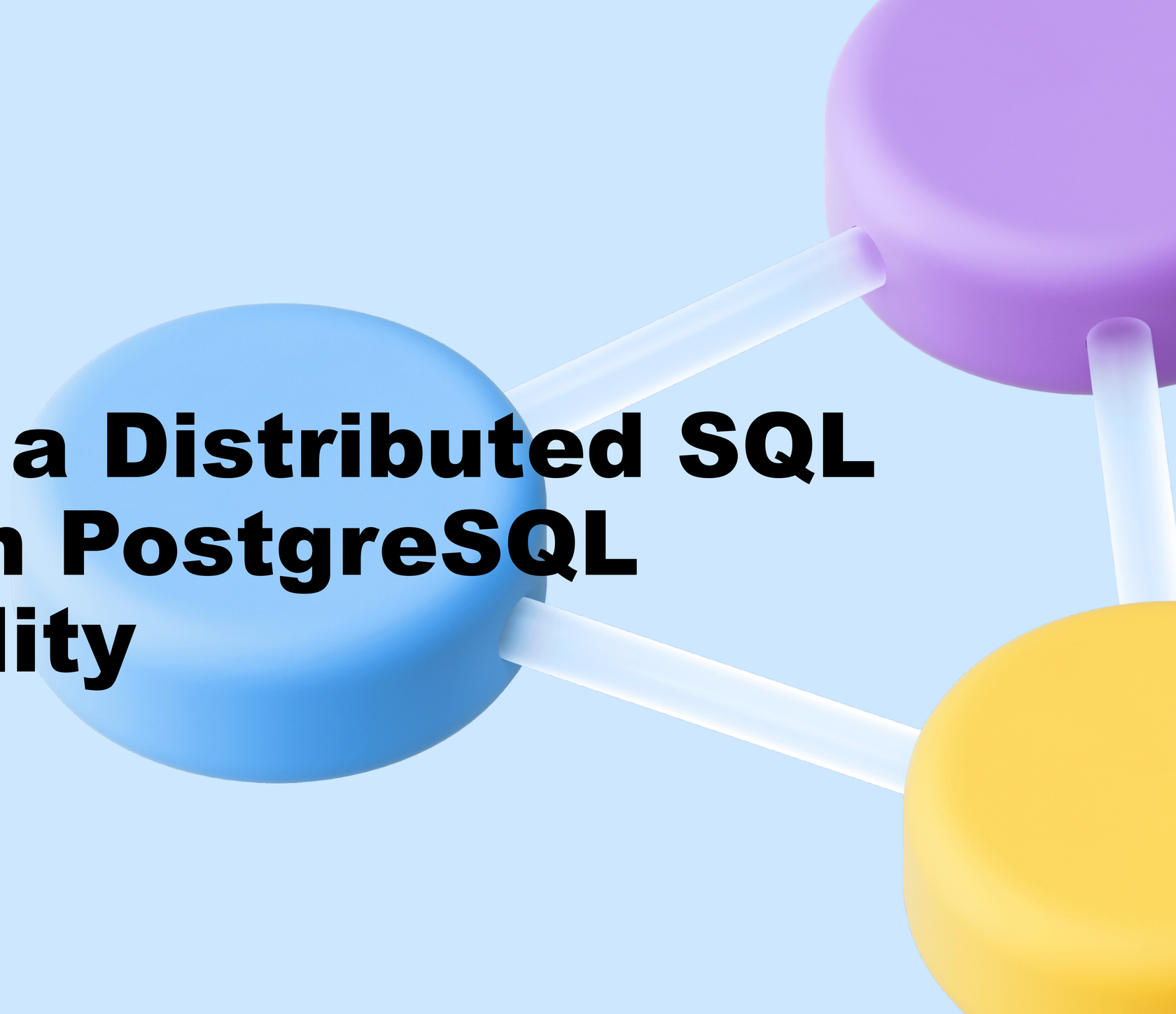
YDB

YDB: extending a Distributed SQL DBMS with PostgreSQL compatibility



Ivan Blinkov
VP, Product and Open-Source

Ivan Blinkov

- Over a decade of experience specifically in database management systems (DBMS)
- Talked with countless DBMS users and stakeholders to understand how and why they ended up with a specific solution
- Worked on a handful of DBMS products, including two open-source ones:

ClickHouse YDB

Agenda

1

Why YDB needs PostgreSQL compatibility?

2

Approaches to making a DBMS PostgreSQL-compatible

3

YDB’s approach and lessons learned

Why YDB needs PostgreSQL compatibility?

YDB: Open-Source Distributed SQL Database

Mission critical

- Designed for services with 24×7 uptime requirements
- Serializable consistency
- Adapts to workloads
- Security features

Highly available

- Survives AZ plus rack failure without human intervention
- Seamless upgrades
- Self-healing
- Smart SDKs

Data platform

- Tables
- Topics (persistent queues)
- Analytics
- Federated queries
- Multitenancy

Typical YDB use cases

- Finance
- E-commerce
- Ride-hailing
- Advertisement
- Logistics
- AI services

What can be built with YDB?

Transactional workloads (OLTP)

- Millions of transactions per second
- Two SQL dialects: YQL and PostgreSQL-compatible

Analytical workloads (OLAP)

- Real-time reporting
- Data warehousing
- AI features

Streaming workloads (persistent queues)

- Exactly once or at least once
- gRPC API or Kafka-compatible API

Summary of YDB history

2014	Started as an in-house infrastructure technology
2020	Provided as a managed cloud service
2022	Published to open-source under Apache 2.0 license
2023	Started working on PostgreSQL compatibility
2024	Added PostgreSQL compatibility

YDB’s PostgreSQL compatibility goals

1

Allow to reuse existing PostgreSQL open-source toolset and ecosystem

2

Reuse YDB’s distributed query engine for:

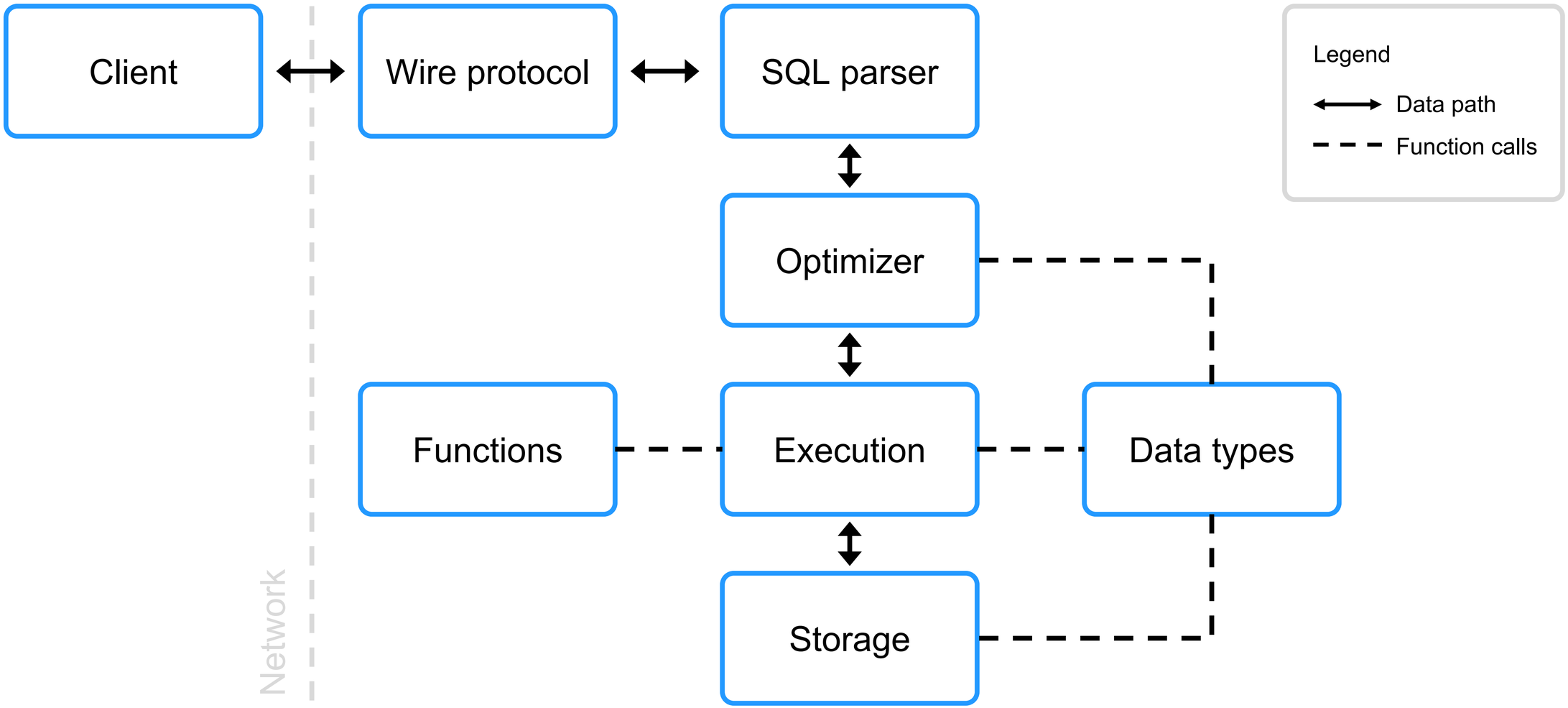
- High availability
- Strong consistency
- Scalability limited by budget

3

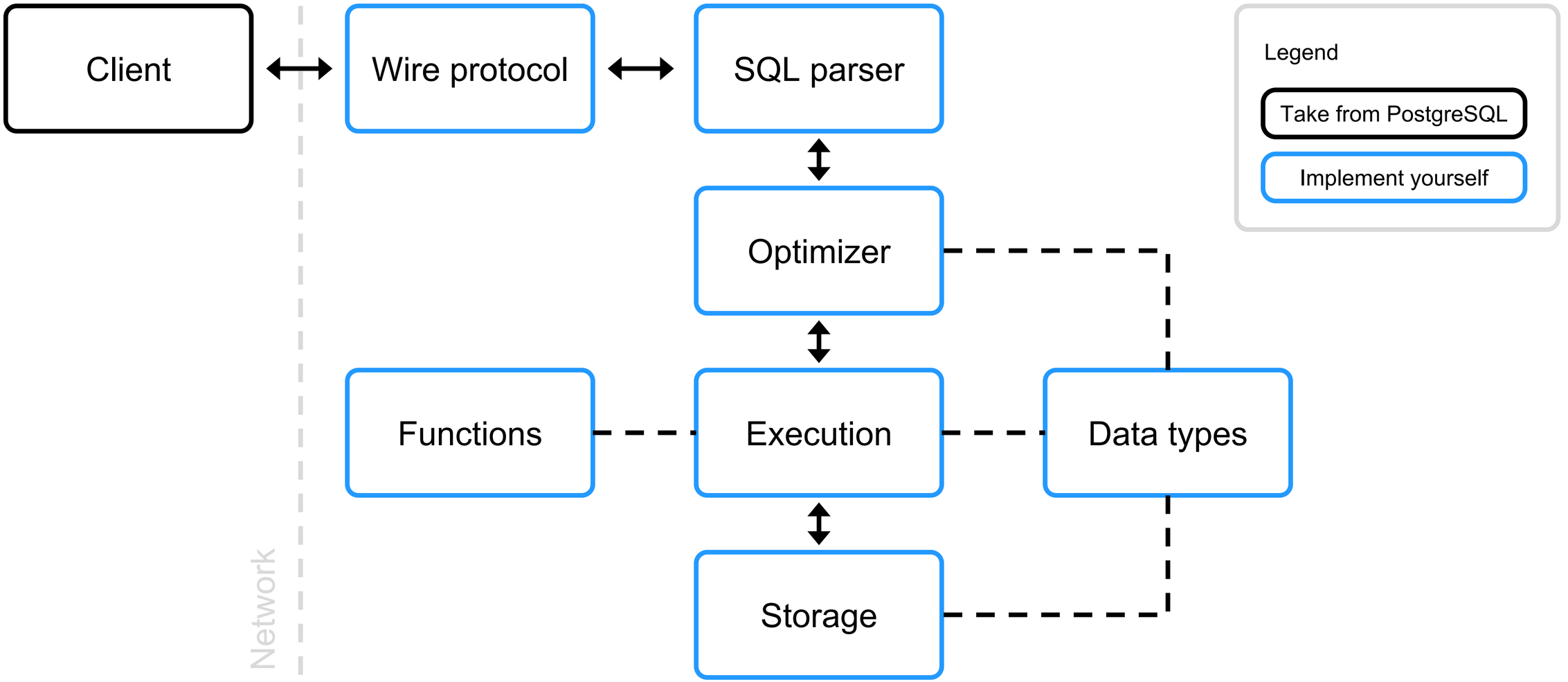
Interoperability between YDB and PostgreSQL layers

Possible approaches to making a database technology PostgreSQL-compatible

How most SQL DBMS process queries?



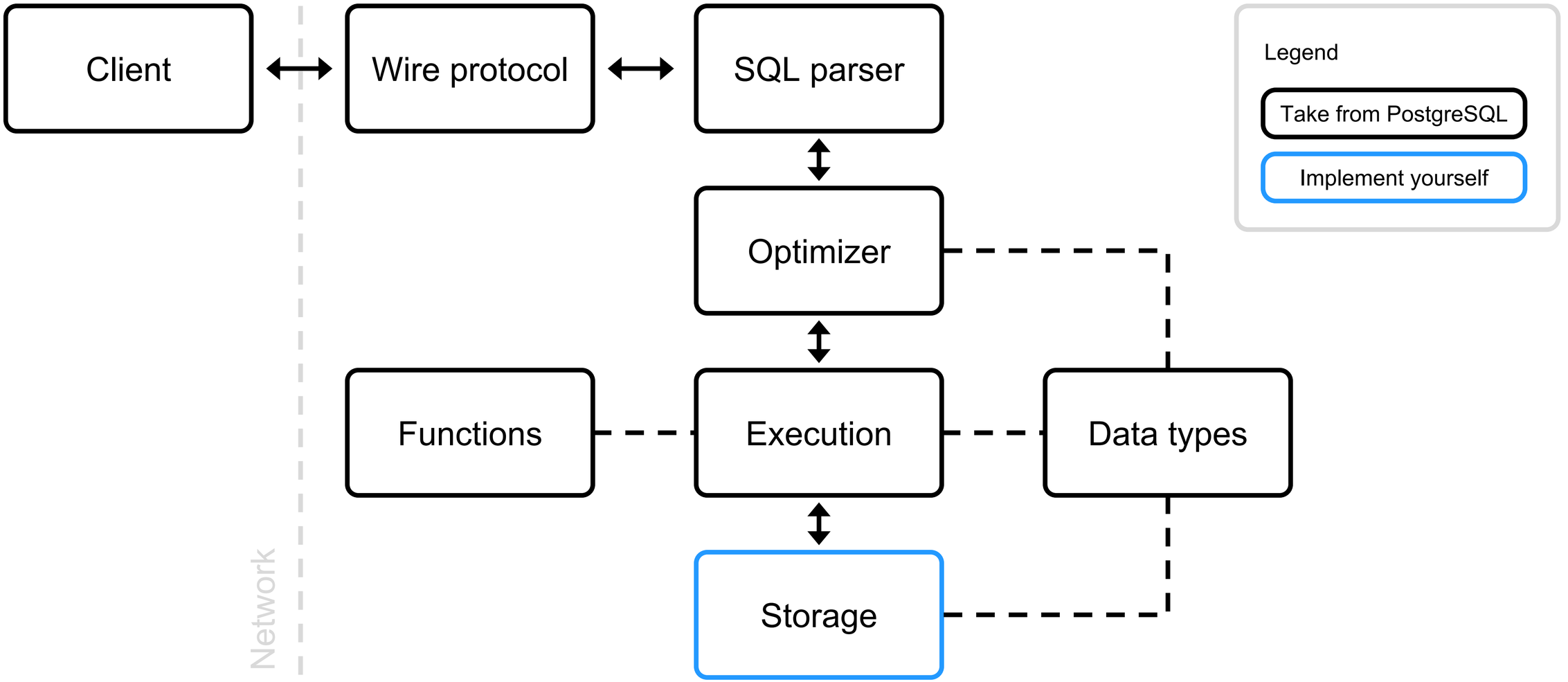
Implement everything from scratch



Implement everything from scratch

Advantages	Disadvantages
Design implementation for distributed environment	Hard to mimic all corner cases
Freedom to optimize algorithms	A lot of work to reimplement all current and future features

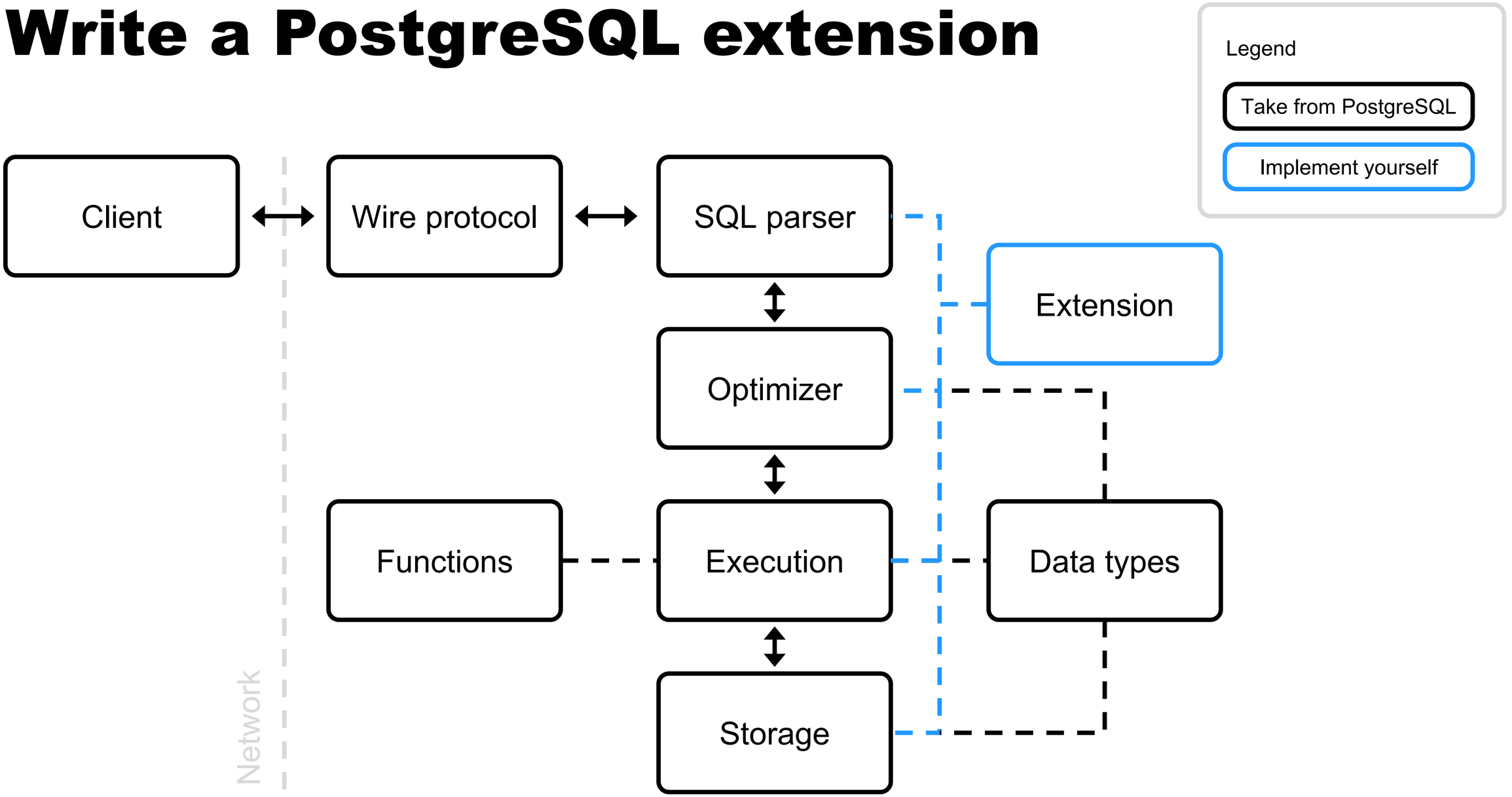
Use full PostgreSQL runtime



Use full PostgreSQL runtime

Advantages	Disadvantages
Best runtime compatibility	Need to main a PostgreSQL fork
Relatively easy PostgreSQL release upgrade	Limited by original runtime capabilities
Partial extension support	

Write a PostgreSQL extension

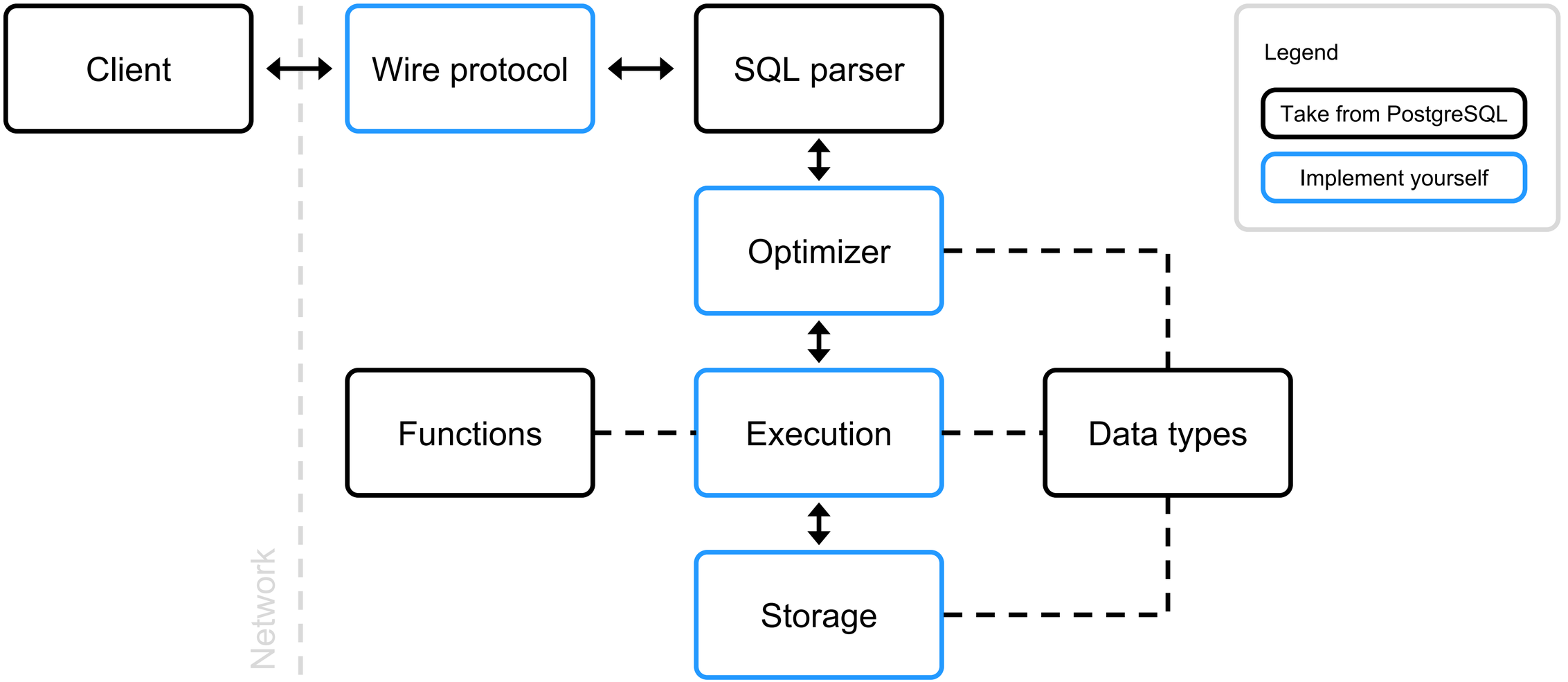


Write a PostgreSQL extension

Advantages	Disadvantages
Uses native extension API	Limited extension points
Easy PostgreSQL release upgrades	Limited by PostgreSQL runtime capabilities

YDB’s approach and lessons learned

YDB's approach: best of both worlds



YDB’s approach: best of both worlds

Advantages	Disadvantages
Keep YDB’s key properties	A lot of work for the integration
High level of PostgreSQL compatibility due to code reuse	Moderate complexity of PostgreSQL release upgrades
Interoperability between PostgreSQL and YDB	

Lessons learned

1. Every SQL implementation is different
2. Only PostgreSQL is 100% compatible with PostgreSQL
3. Settle on goals and trade-off expectations early
4. Testing compatibility and its coverage is crucial:
    - PostgreSQL regression tests
    - Documentation based tests
    - Drivers integration tests
    - Real applications

YDB is 100% open-source

Permissive Apache 2.0 License for:

- Core platform is built from scratch in C++
- SDKs in Java, Python, Go, Rust, Node.js, PHP, etc.
- Documentation in Markdown

https://github.com/ydb-platform/ydb

Contributors are welcome!

YDB

Thank you!

https://ydb.tech

YDB highlights:

- Strong consistency
- Resilience and self-healing
- Elastic scalability
- PostgreSQL compatibility
- Various workloads
- 100% open-source under Apache 2.0