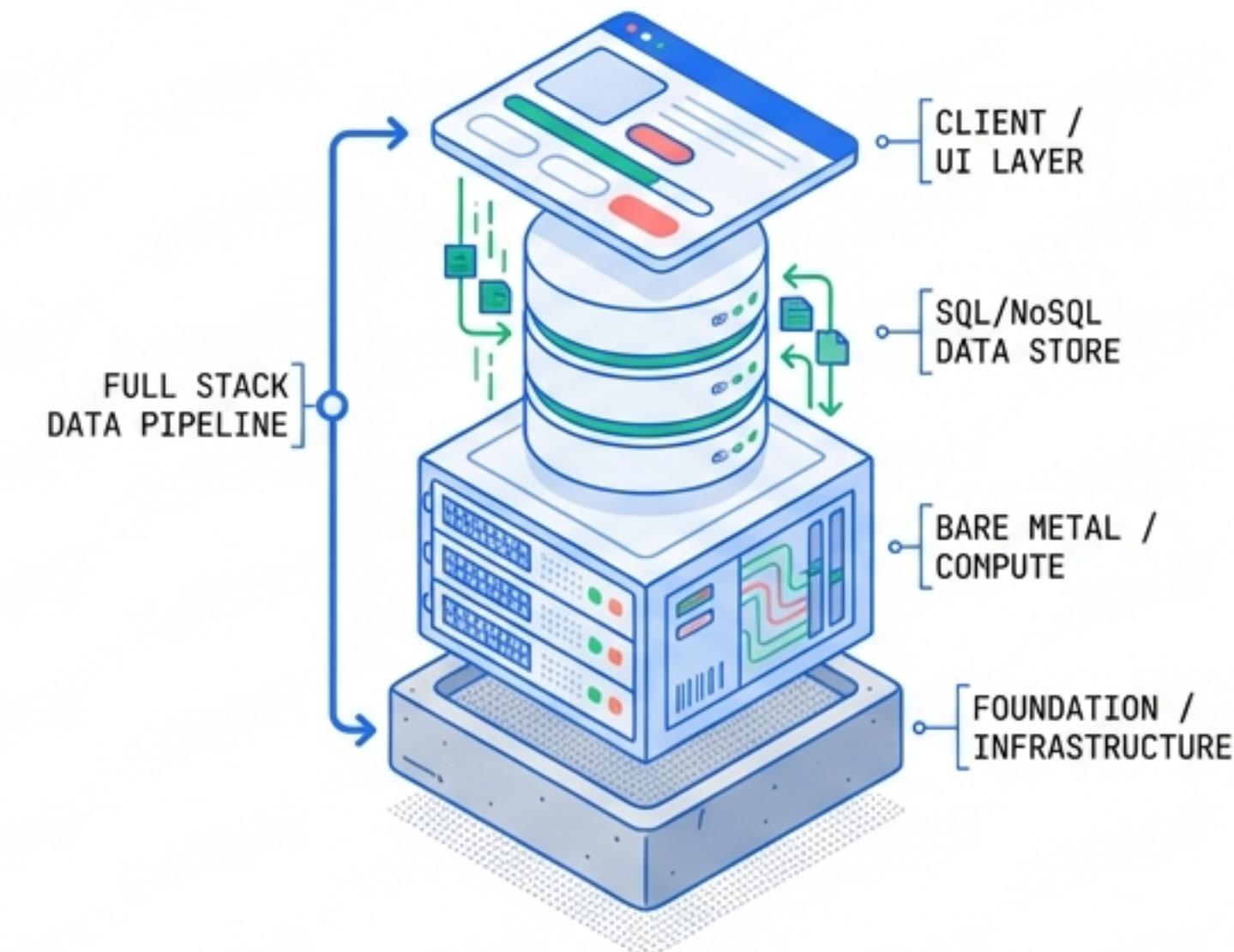


The System Design Architect's Handbook

A visual curriculum: From bare metal to distributed scale.



Module 0: The Engineering Knowledge Base

The industry leaders defining modern system design.



Netflix TechBlog



Uber Blog



Cloudflare Blog



Engineering at
Meta



LinkedIn
Engineering



Discord Blog



AWS
Architecture



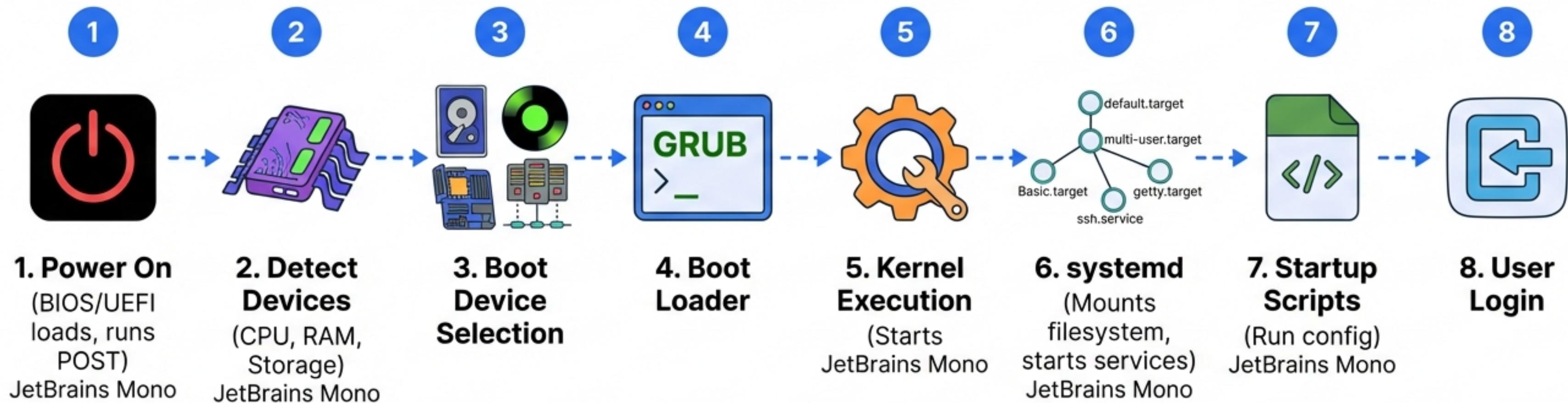
Slack
Engineering



Stripe Blog

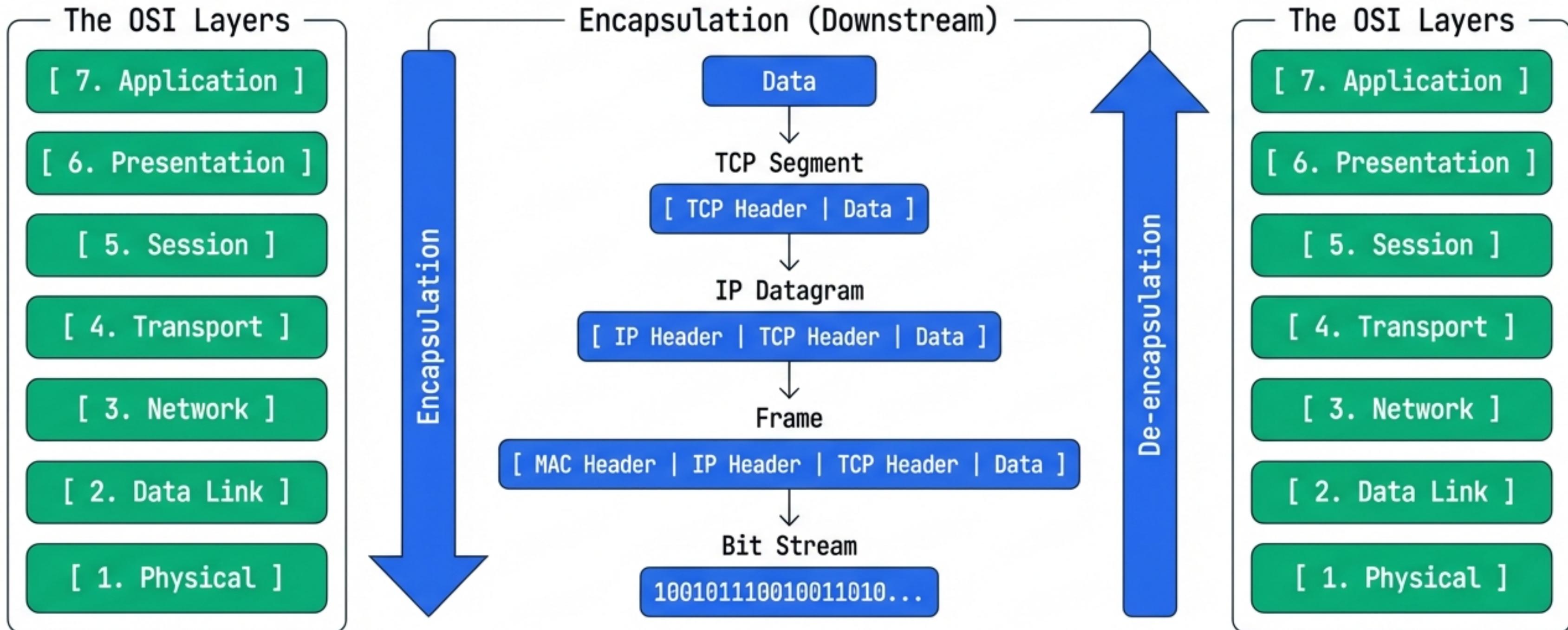
Module 1: Infrastructure Foundations

The Linux Boot Process



Module 1: Infrastructure Foundations

The OSI Model & Data Encapsulation



Module 1: Infrastructure Foundations

8 Popular Network Protocols

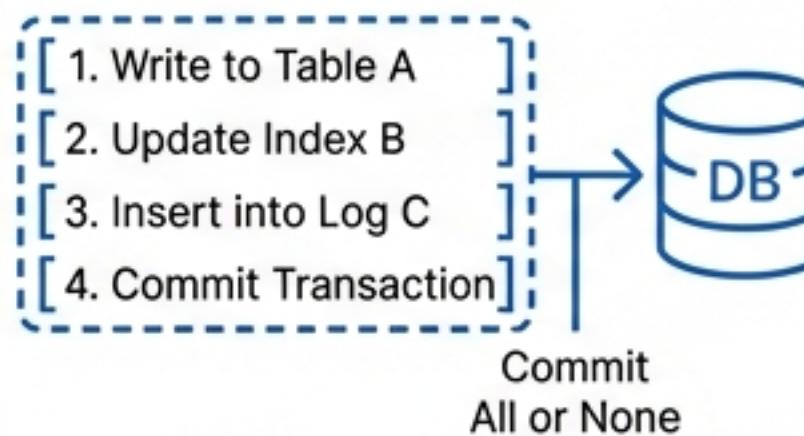
-  **HTTP** Client-Server web foundation.
-  **HTTPS** Secure HTTP with encryption.
-  **HTTP/3 (QUIC)** UDP-based, faster response.
-  **WebSocket** Full-duplex, real-time push.
-  **TCP** Reliable, **connection-based** delivery.
-  **UDP** **Connectionless**, fast, packet-loss tolerant.
-  **SMTP** Simple Mail Transfer Protocol.
-  **FTP** File Transfer Protocol.

Module 2: The Data Layer

Transactional Integrity (ACID)

Atomicity

All or nothing. If any part fails, the entire transaction rolls back.



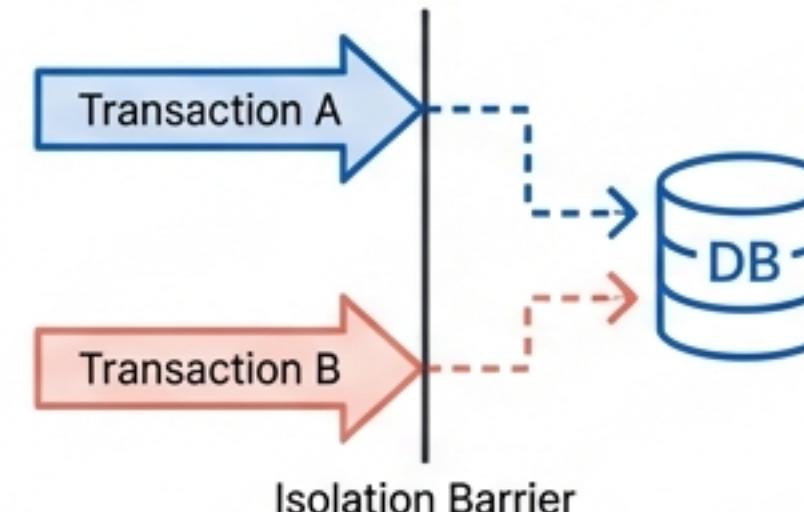
Consistency

Preserving invariants. Data written must be valid according to defined rules.



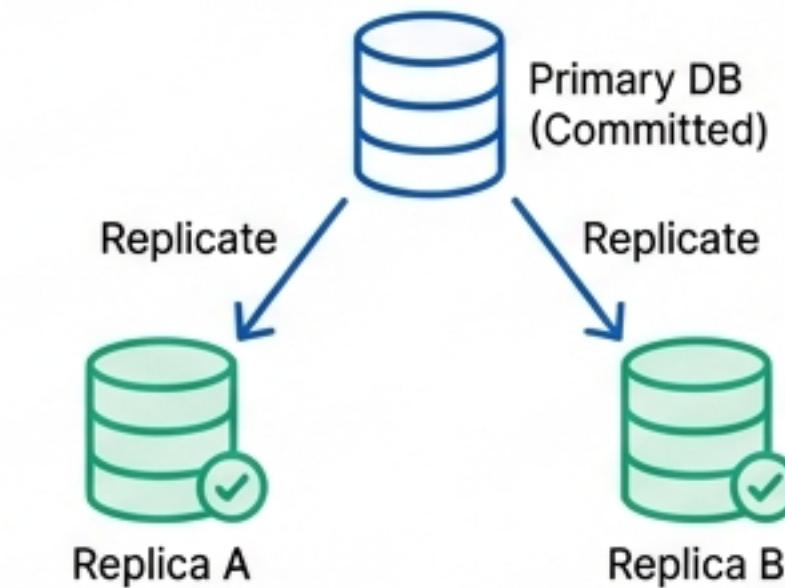
Isolation

Concurrent transactions do not interfere. Serializability is the strictest form.



Durability

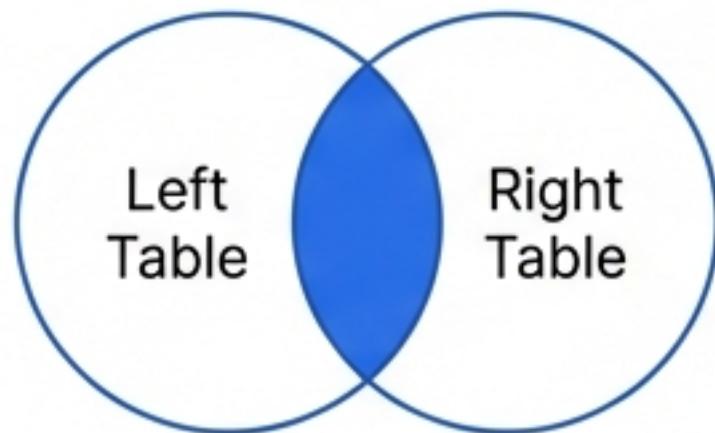
Persistence. Once committed, data survives system failure.



Module 2: The Data Layer

SQL Joins Visualized

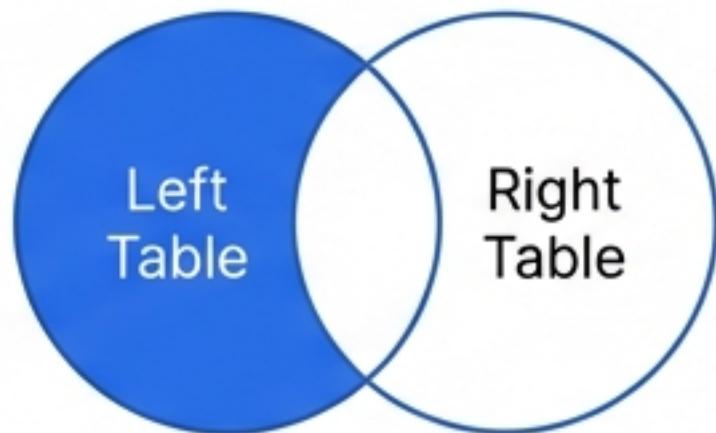
INNER JOIN



Returns only matching rows in both tables.

ID	Value
123	A
124	B

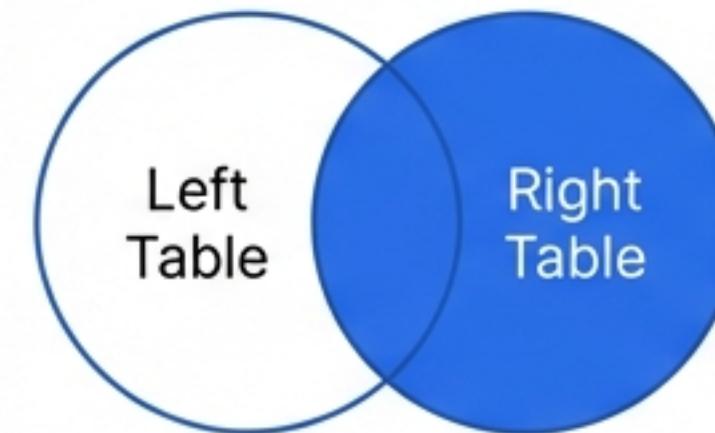
LEFT JOIN



Returns all records from the Left table + matching from Right.

L.ID	L.Val	R.Val
123	A	X
124	B	Y
125	C	NULL

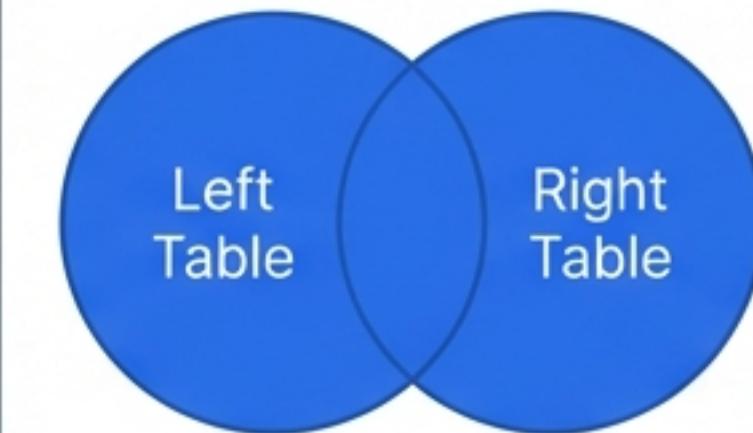
RIGHT JOIN



Returns all records from the Right table + matching from Left.

R.ID	R.Val	L.Val
123	X	A
124	Y	B
126	Z	NULL

FULL OUTER JOIN

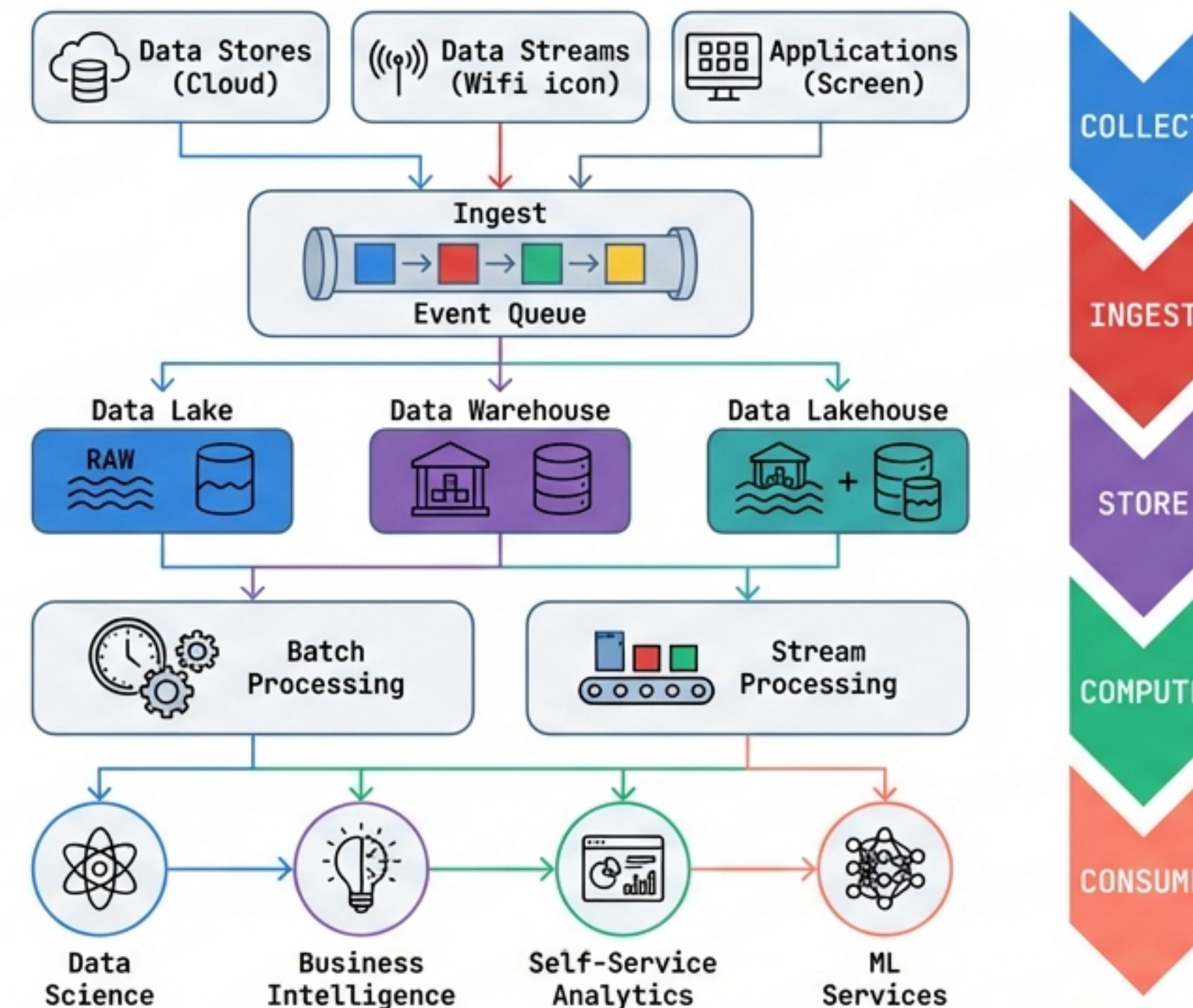


Returns all records where there is a match in either table.

L.ID	L.Val	R.ID	R.Val
123	A	123	X
124	B	124	Y
125	C	NULL	NULL
NULL	NULL	126	Z

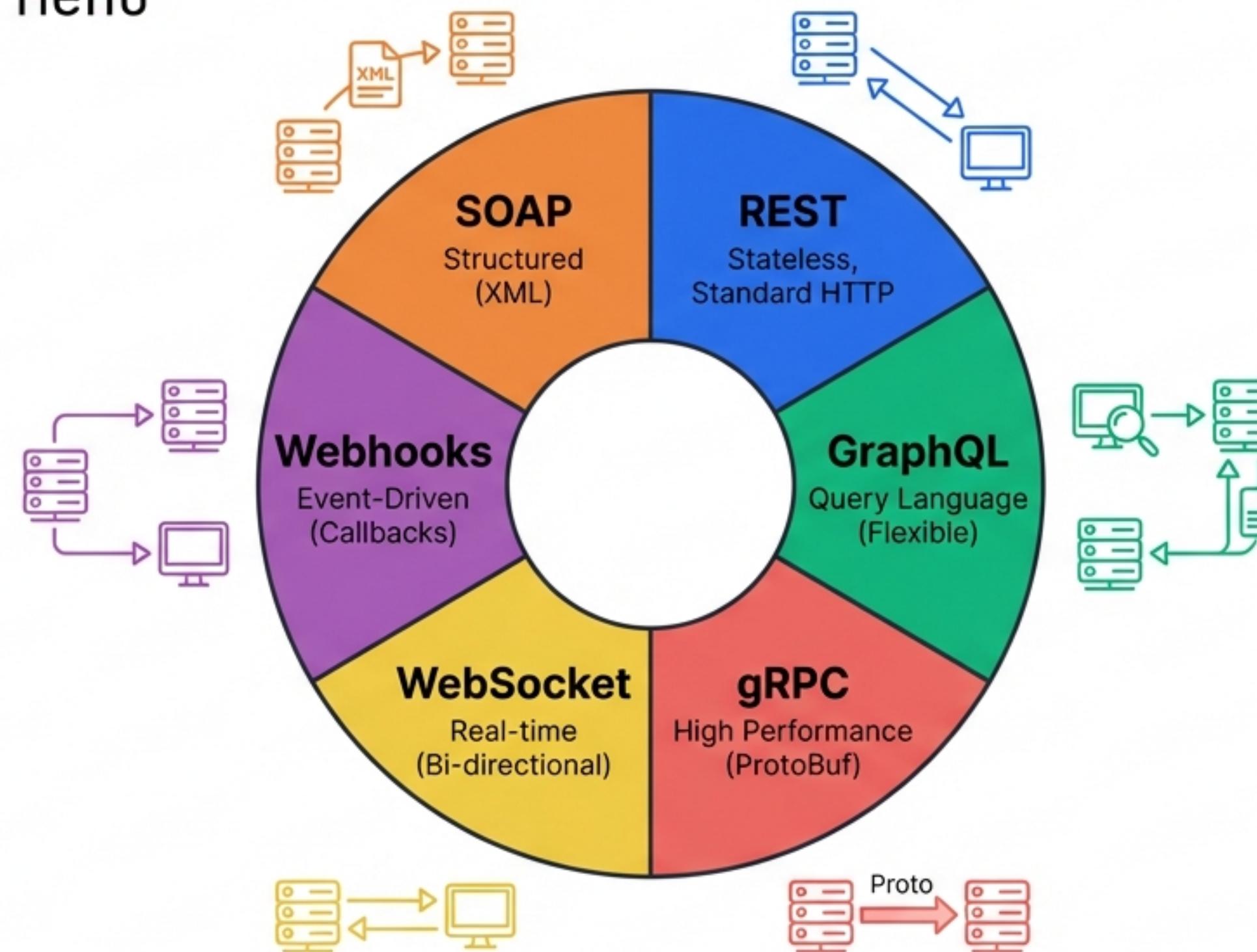
Module 2: The Data Layer

Data Pipelines Overview



Module 3: API Design & Communication

The Protocol Menu

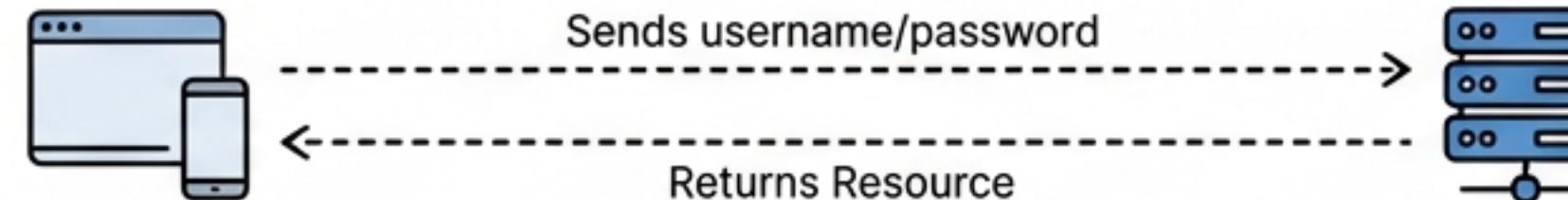


Module 3: API Design & Communication

REST API Authentication Methods



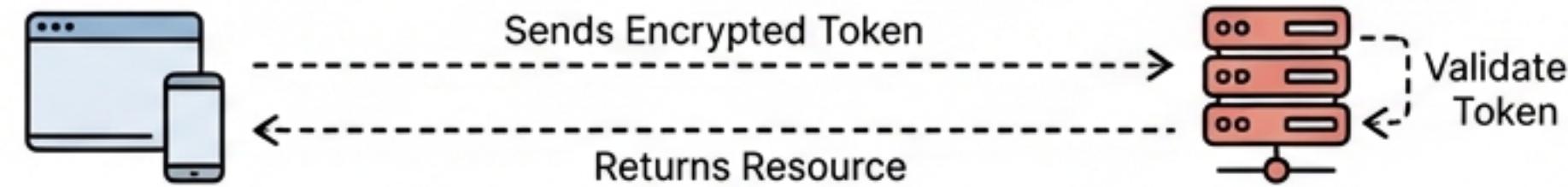
Basic Authentication



Best For:
Simple / Internal



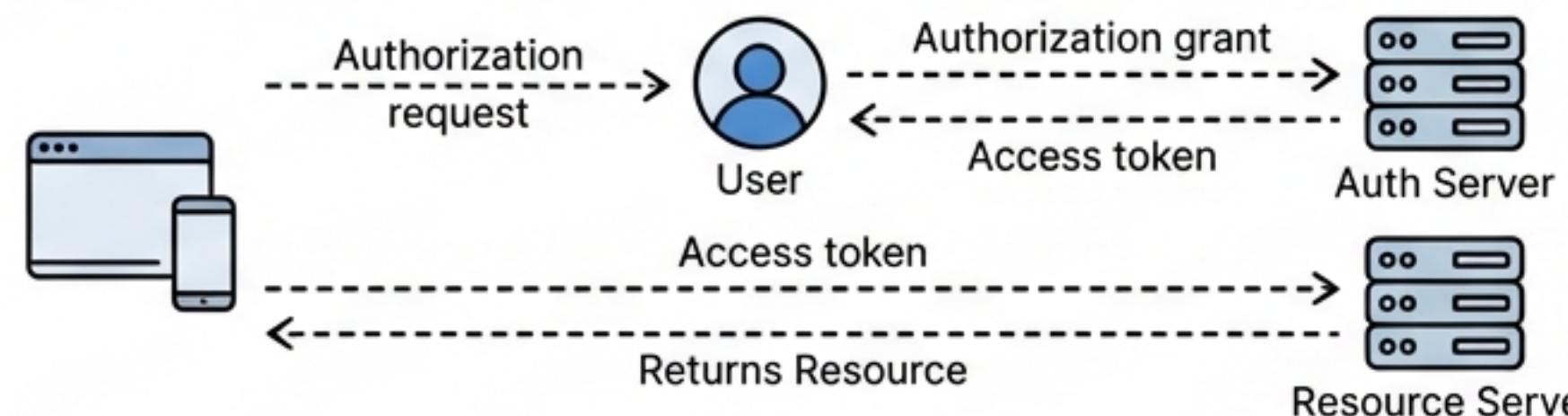
Token Authentication



Best For:
SPAs / Mobile Apps



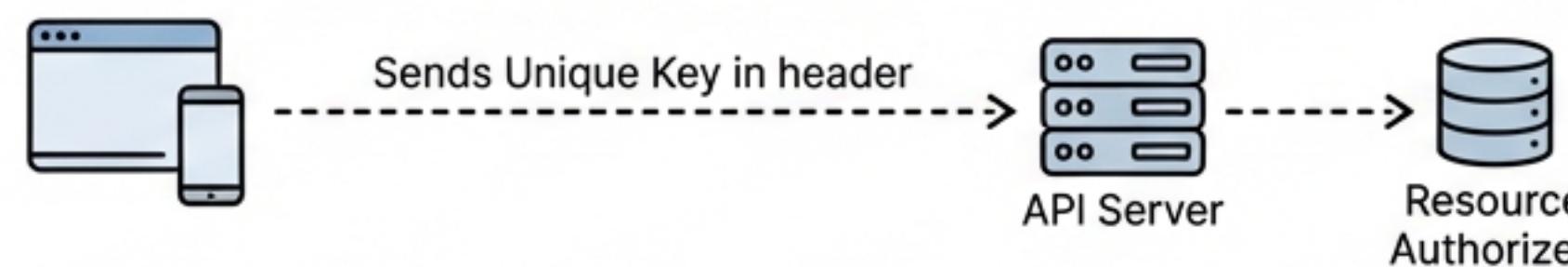
OAuth Authentication



Best For:
Third-party Access /
Social Login



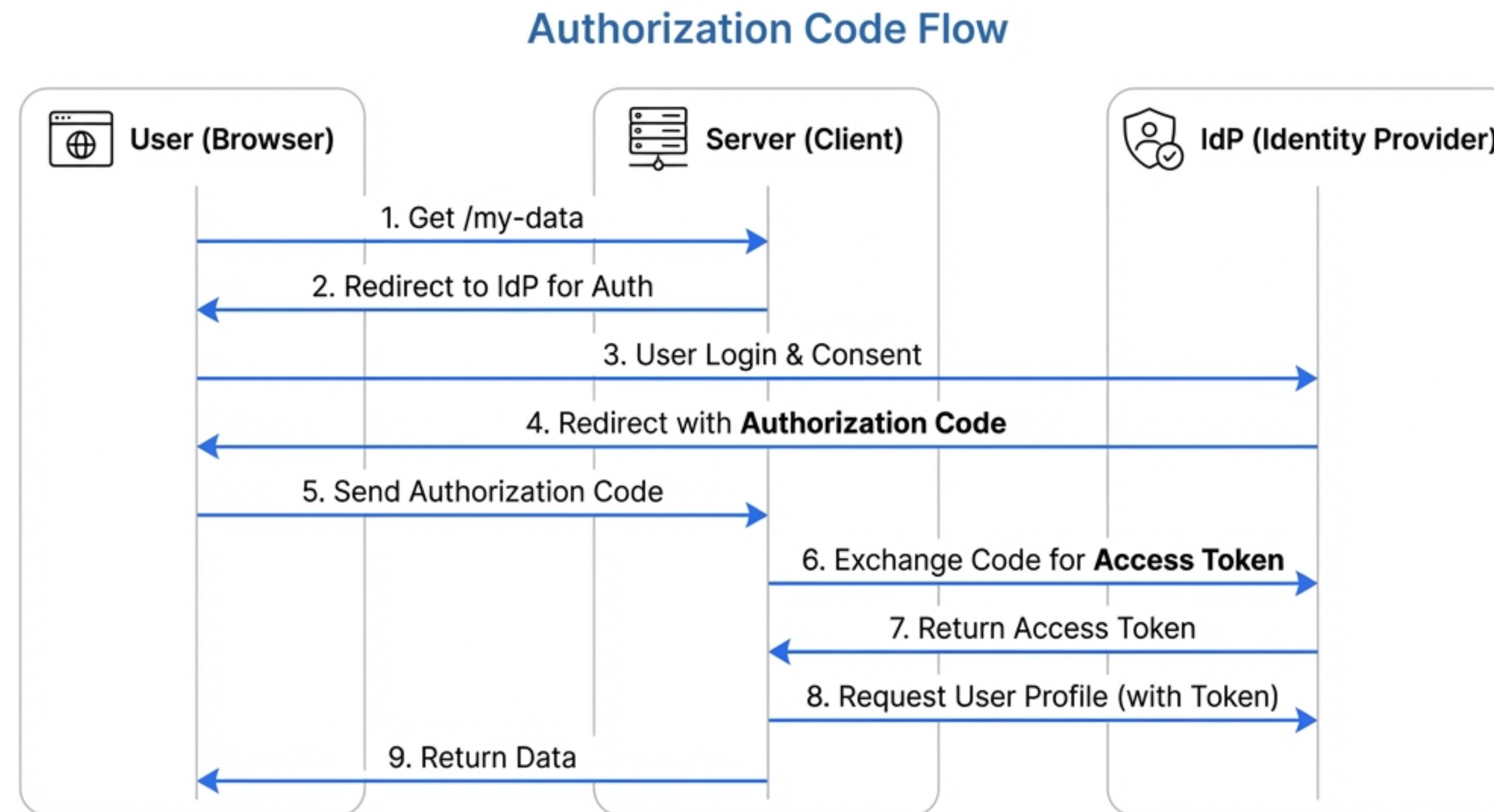
API Key Authentication



Best For:
Project Throttling / S2S

Module 3: API Design & Communication

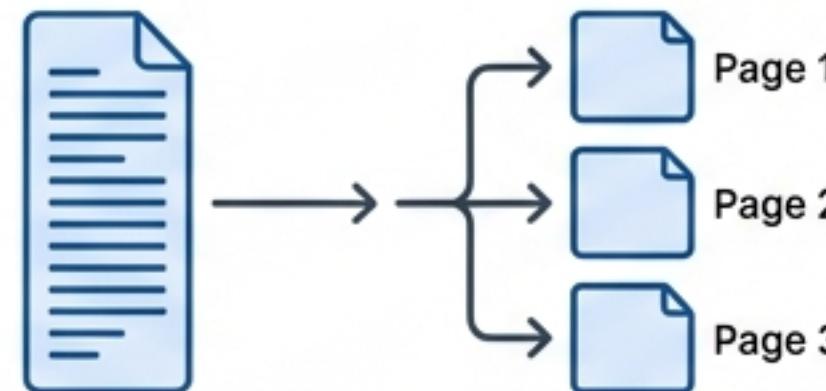
OAuth 2.0 Deep Dive: Authorization Code Flow



Module 3: API Design & Communication

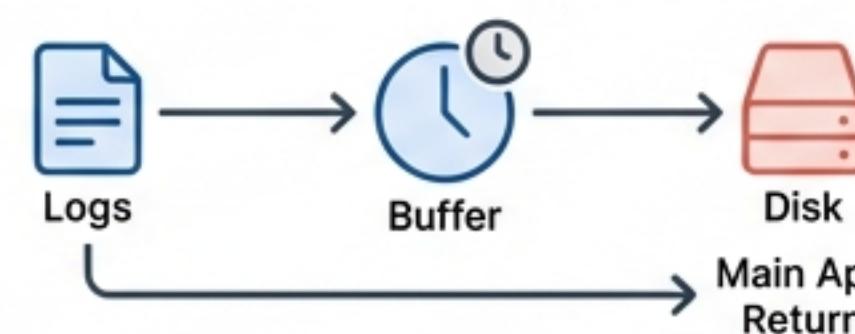
5 Ways to Improve API Performance

1. Pagination



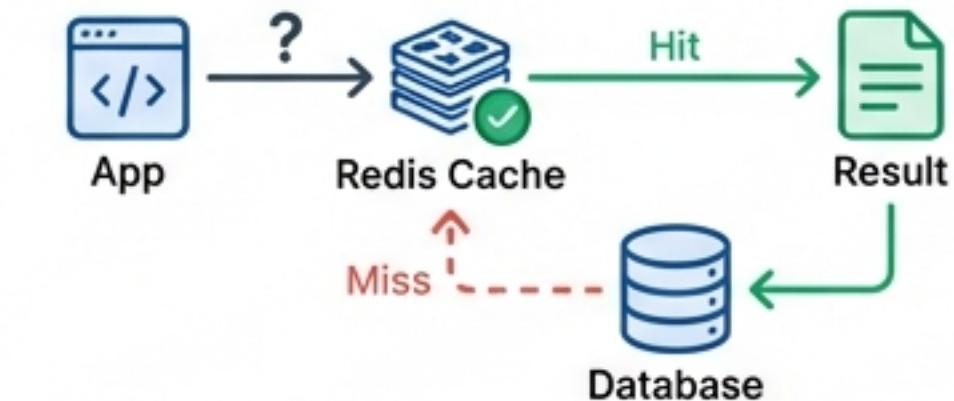
Stream large results in pages to reduce payload size.

2. Async Logging



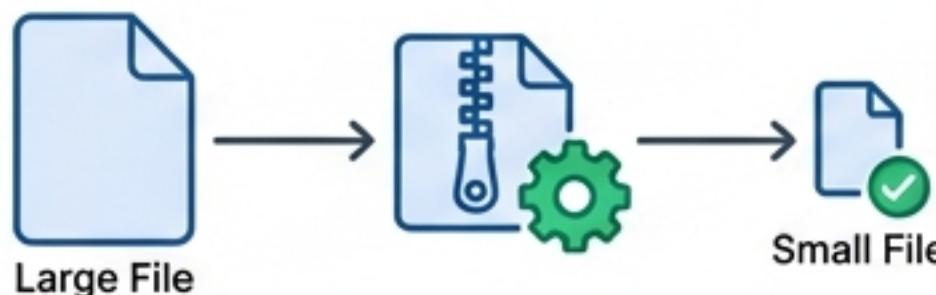
Use lock-free buffers to avoid disk I/O blocking.

3. Caching



Store frequently used data in-memory (Redis).

4. Payload Compression

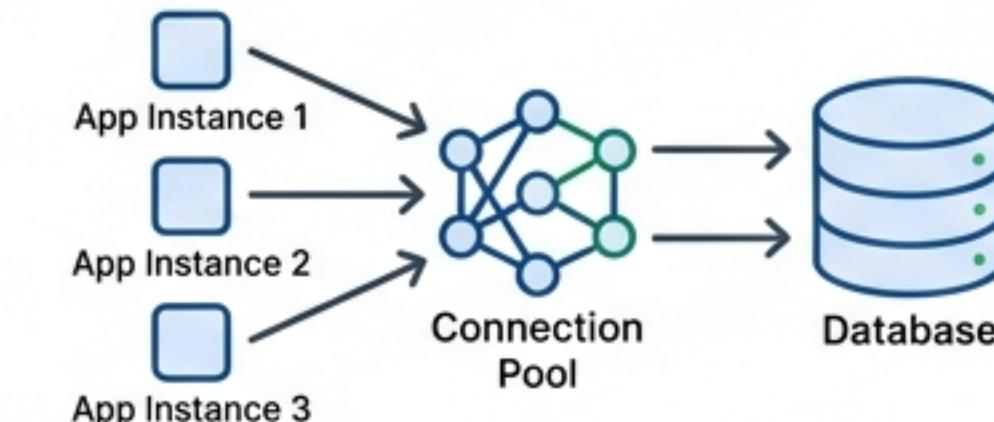


Gzip requests and responses to speed up transmission.



Gzip requests and responses to speed up transmission.

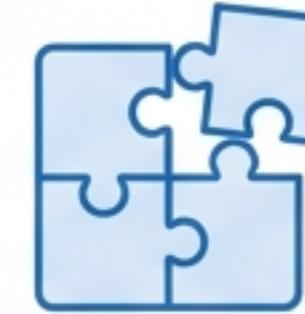
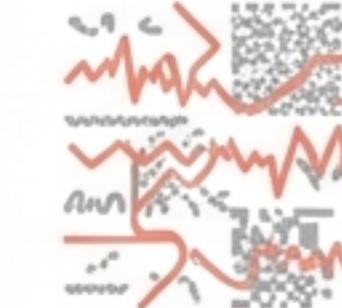
5. Connection Pooling



Reuse open DB connections to avoid handshake overhead.

Module 3: API Design & Communication

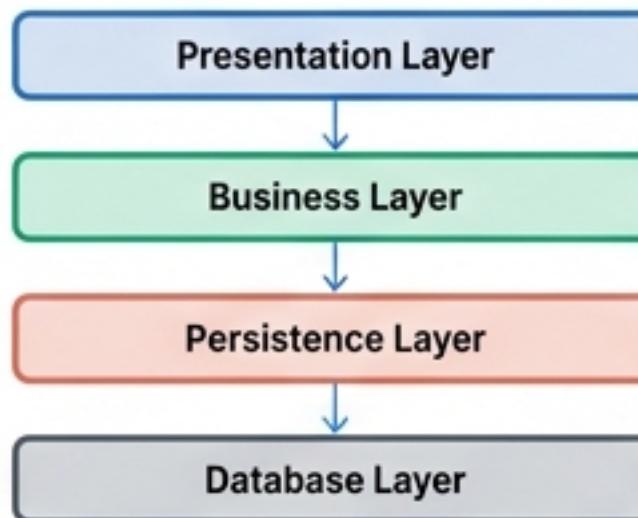
9 Types of API Testing

 Smoke Basic validation.	 Functional Meets requirements.	 Integration End-to-end flow.
 Regression Did new code break old features?	 Load Capacity testing.	 Stress Breaking point.
 Security External threats.	 UI Interaction test.	 Fuzz Invalid input injection.

Module 4: Architectural Patterns

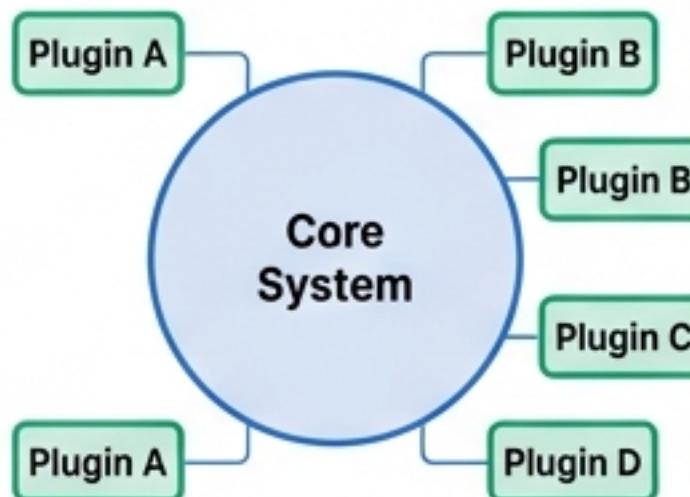
Top Architectural Styles

1. Layered (N-tier)



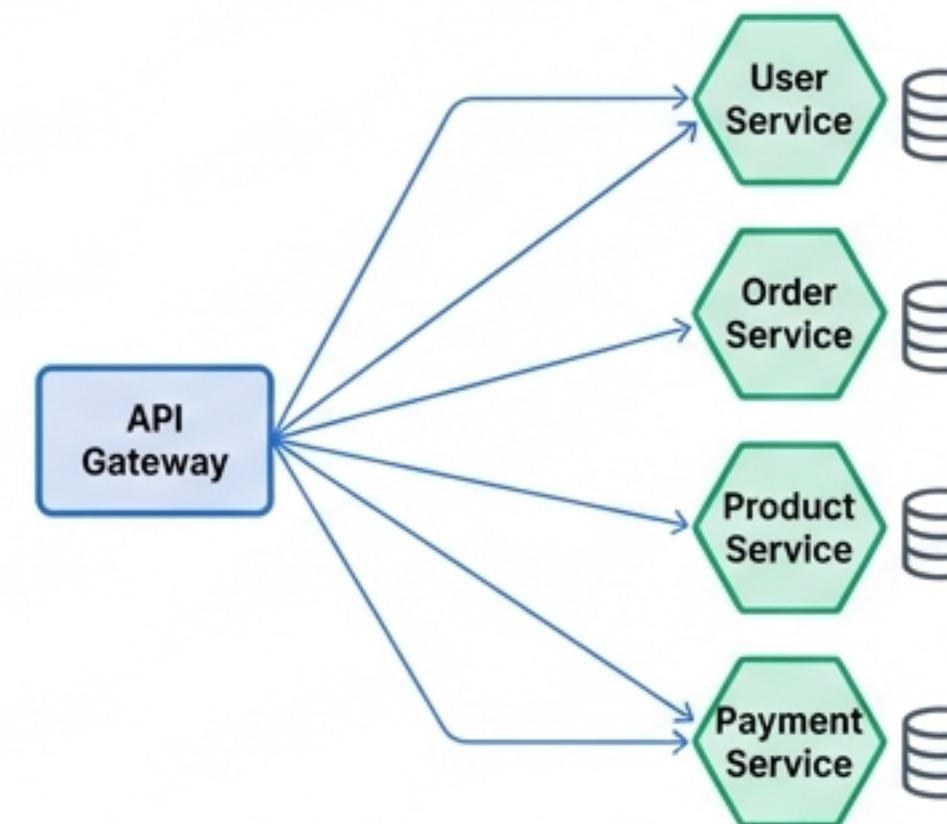
Separates software into logical layers for maintainability.

4. Microkernel



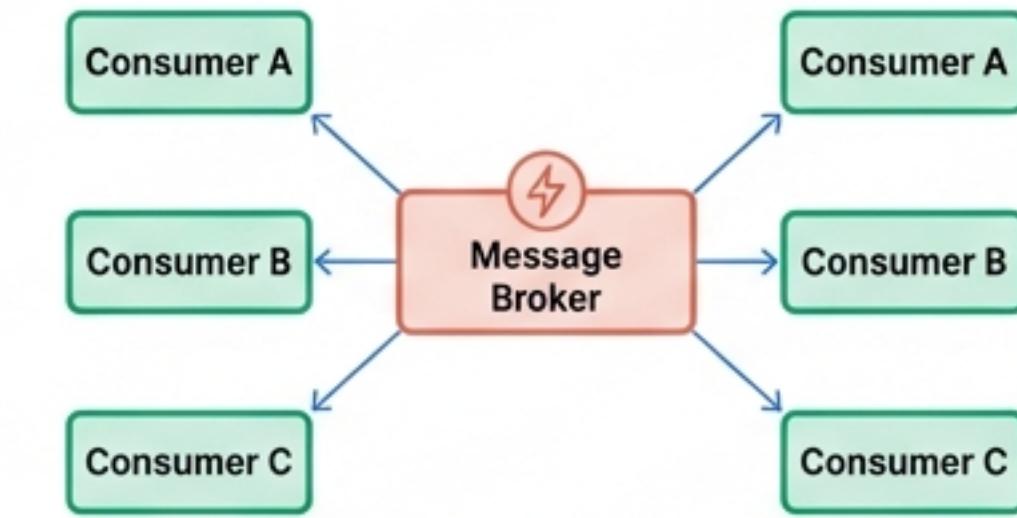
Separates a minimal core from extended functionality plugins.

2. Microservices



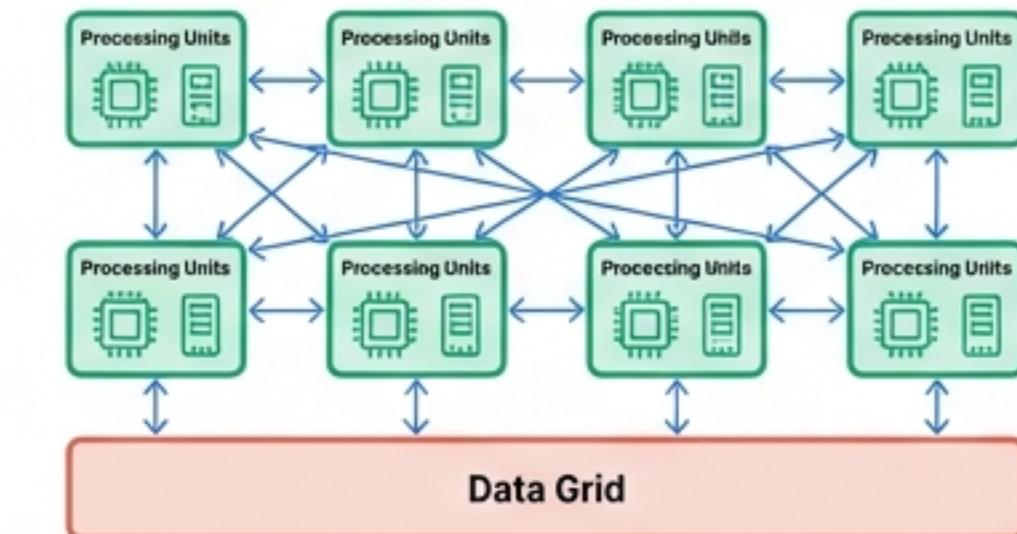
Decomposes application into small, independent services.

3. Event-Driven



Promotes production, detection, consumption of events.

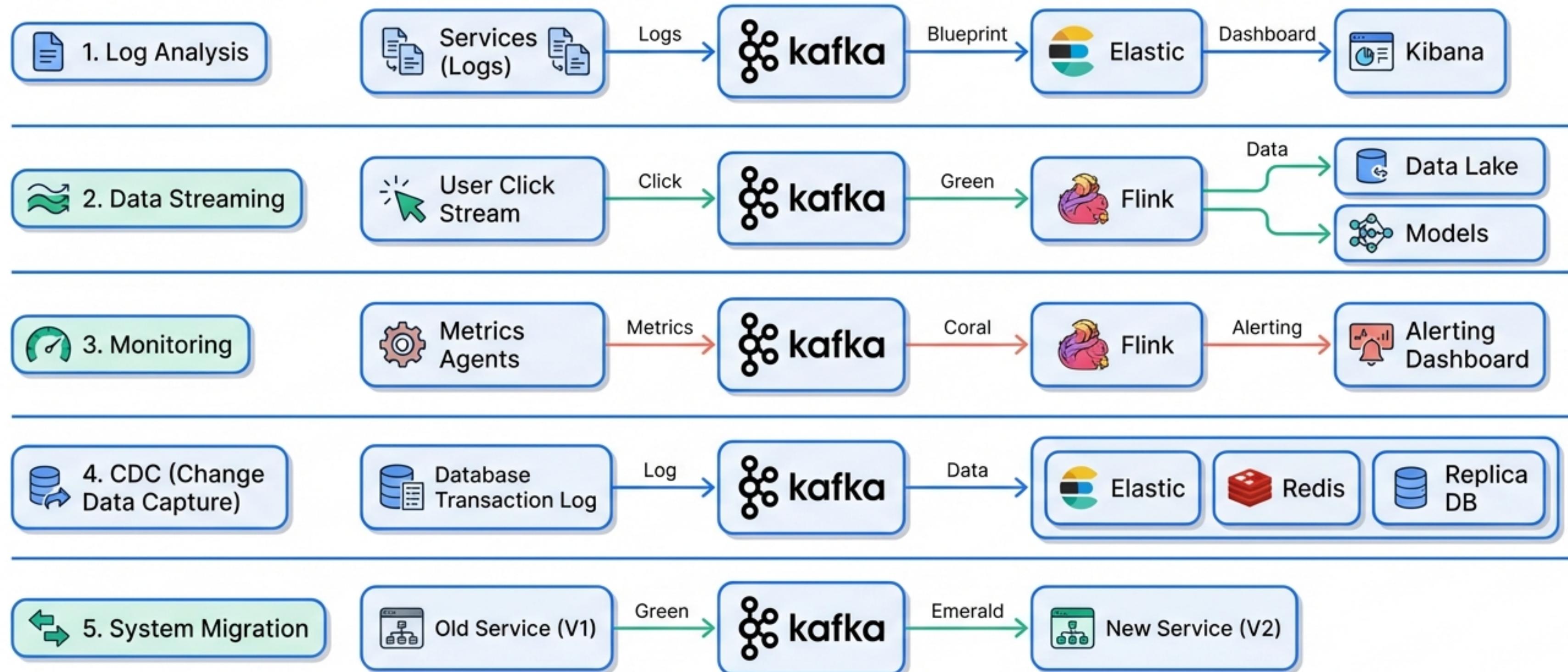
5. Space-Based



Uses processing units and a data grid for high scalability.

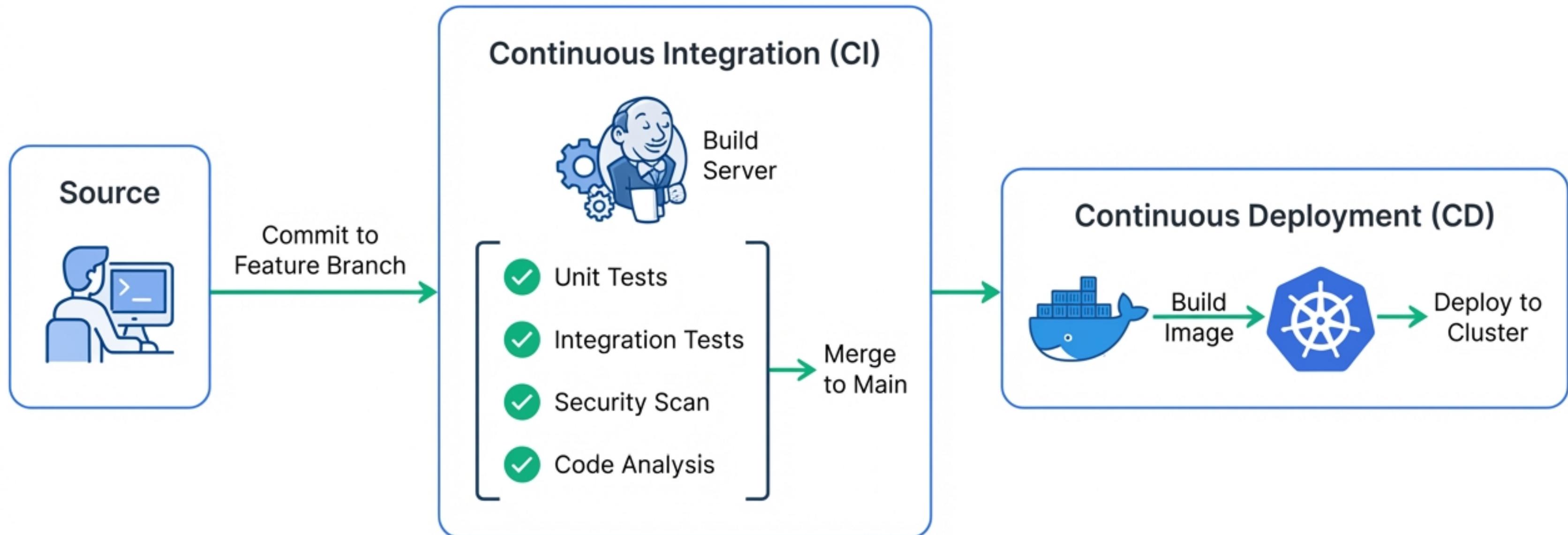
Module 4: Scalability

Apache Kafka Use Cases



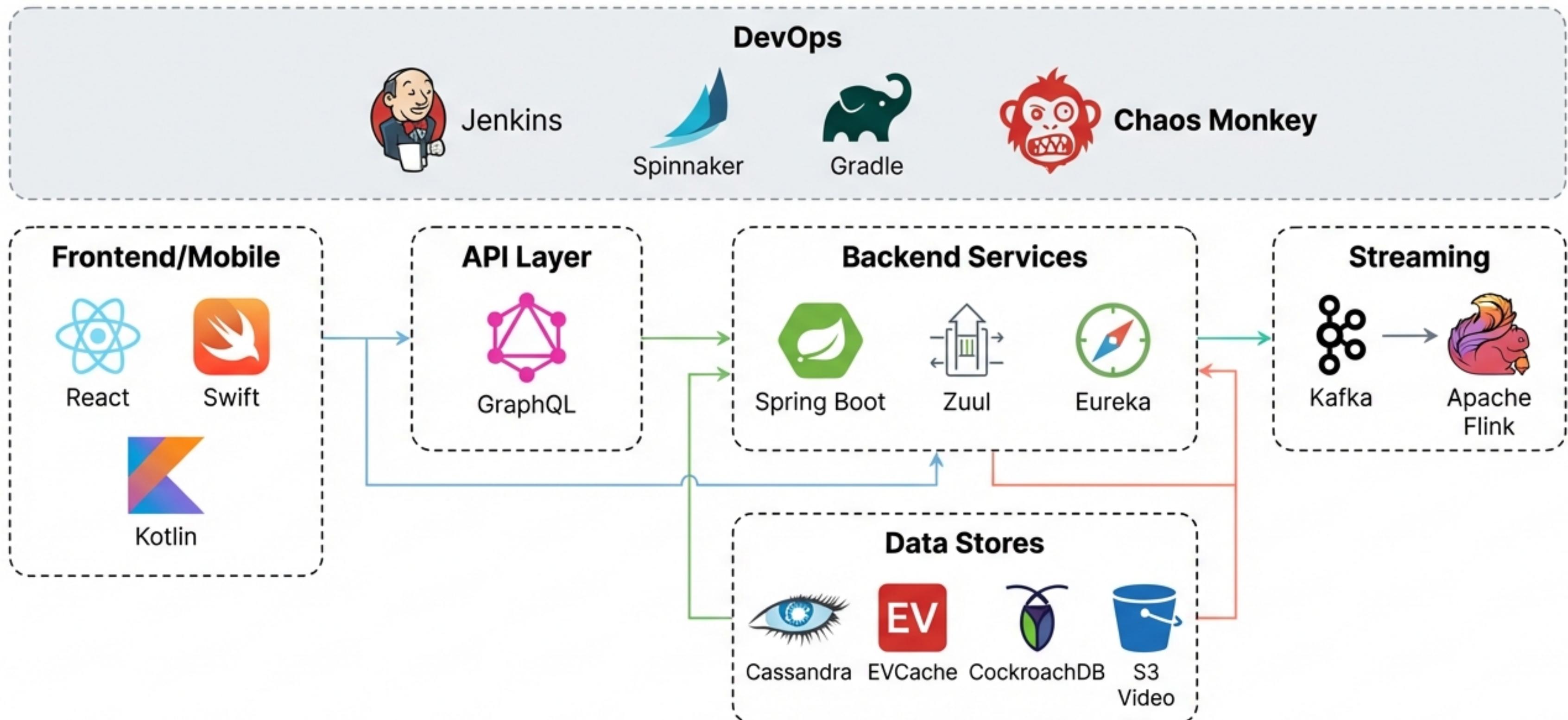
Module 5: Delivery & Operations

The CI/CD Pipeline



Module 6: Case Study

The Netflix Tech Stack



The Full-Stack Journey

Summary & Checklist

“System Design is not about mastering one tool, but understanding the flow of data through the entire ecosystem.”



Infrastructure: Linux Boot & OSI Model.



Data Layer: ACID Transactions & SQL Joins.



Communication: API Protocols & OAuth 2.0.



Architecture: Microservices & Kafka Streaming.



Operations: CI/CD Pipelines.