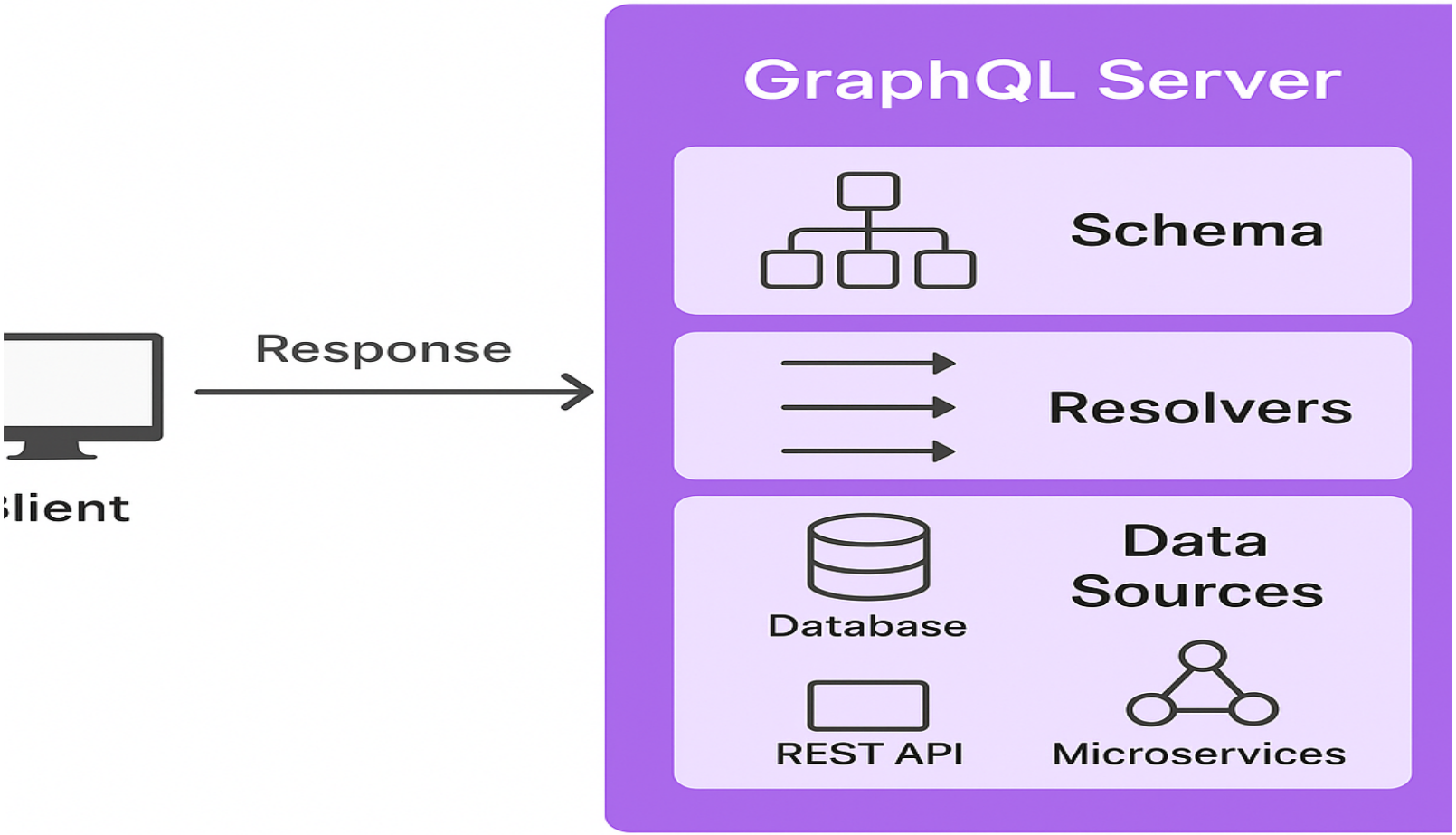
GraphQL: End-to-End Architecture Deep Dive

A conceptual and architectural guide with detailed architecture diagram and explanation.

GraphQL Deep Architecture Diagram


Response
lient
GraphQL Server
Schema
Resolvers
Data Sources
Database
REST API
Microservices

Architecture Explanation:
This diagram represents the core architecture of GraphQL in an enterprise environment. It highlights the request and response flow between the client and backend systems.

Client: Sends a GraphQL query specifying exactly the data required.
GraphQL Server: Processes the request through three main stages — Schema, Resolvers, and Data Sources.
- **Schema:** Defines the available data types and fields clients can query.
- **Resolvers:** Functions that handle data fetching for each field in the schema.
- **Data Sources:** Can include Databases, REST APIs, or Microservices. These provide the raw data requested.

The GraphQL Server validates, executes, and aggregates data into a single, efficient JSON response to the client. This architecture ensures flexibility, strong typing, and minimal over-fetching.

Summary

GraphQL provides a unified, schema-driven layer between clients and distributed data systems. Its resolver-based execution model enables modular scalability while maintaining strict type safety. This deep architecture model forms the foundation for advanced implementations such as GraphQL Federation and Edge GraphQL, enabling global-scale data orchestration with strong consistency guarantees.

Prepared for architectural reference — Landscape Edition.