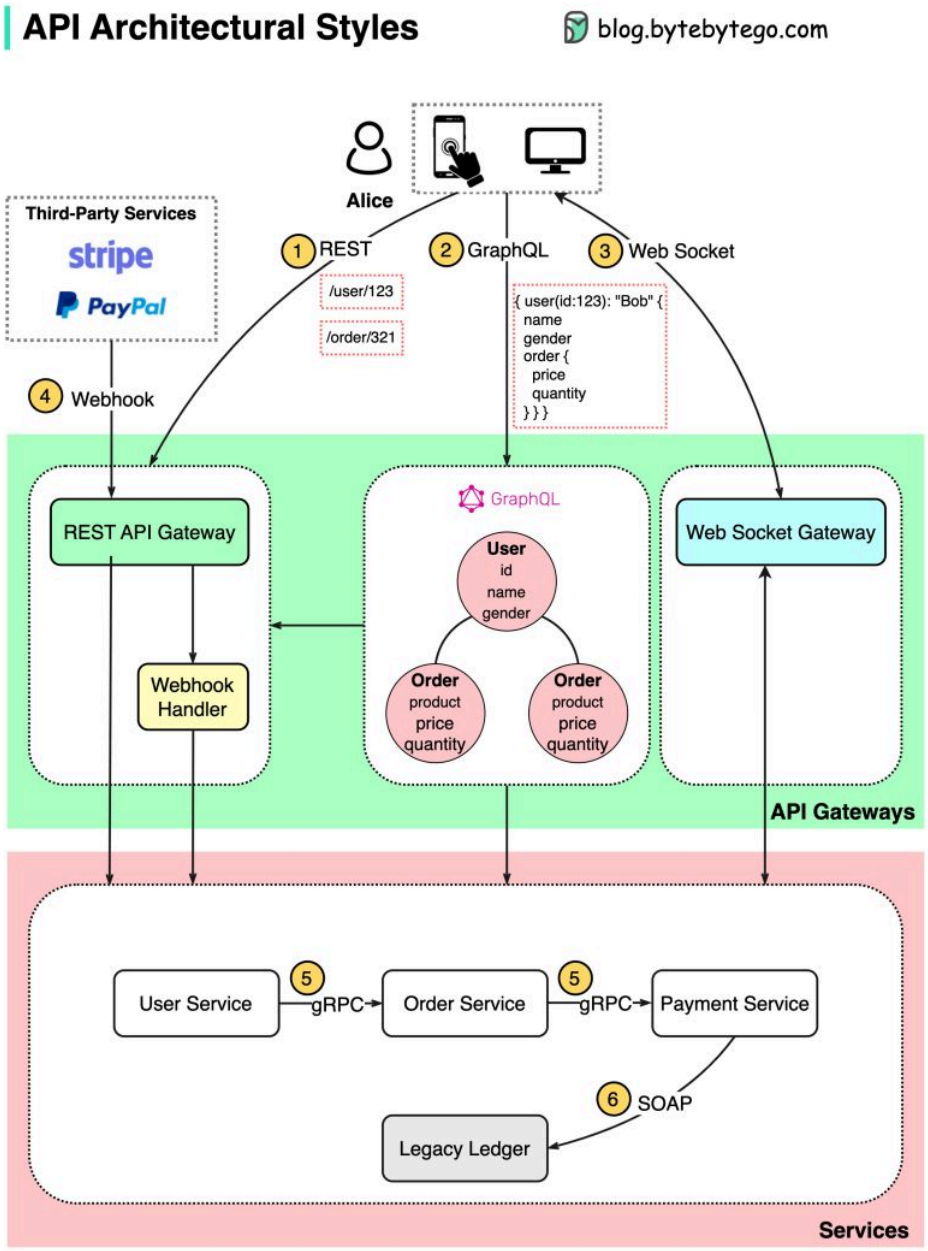
What are the API architectural styles?

The diagram below shows the common API architectural styles in one picture.



1. REST
   Proposed in 2000, REST is the most used style. It is often used between front-end clients and back-end services. It is compliant with 6 architectural constraints. The payload format can be JSON, XML, HTML, or plain text.

2. GraphQL

GraphQL was proposed in 2015 by Meta. It provides a schema and type system, suitable for complex systems where the relationships between entities are graph-like. For example, in the diagram below, GraphQL can retrieve user and order information in one call, while in REST this needs multiple calls.

GraphQL is not a replacement for REST. It can be built upon existing REST services.

3. Web socket
   Web socket is a protocol that provides full-duplex communications over TCP. The clients establish web sockets to receive real-time updates from the back-end services. Unlike REST, which always “pulls” data, web socket enables data to be “pushed”.

4. Webhook
   Webhooks are usually used by third-party asynchronous API calls. In the diagram below, for example, we use Stripe or Paypal for payment channels and register a webhook for payment results. When a third-party payment service is done, it notifies the payment service if the payment is successful or failed. Webhook calls are usually part of the system’s state machine.

5. gRPC
   Released in 2016, gRPC is used for communications among microservices. gRPC library handles encoding/decoding and data transmission.

6. SOAP
   SOAP stands for Simple Object Access Protocol. Its payload is XML only, suitable for communications between internal systems.

👉 Over to you: What API architectural styles have you used?