Yogesh Patil • 3rd+
Full-Stack Developer @TCS | Java | Spring Boot | React | Next.js | SQL | NoSQL | 2x GCP Certifie...
3d •

Follow

SOAP vs REST vs GraphQL vs gRPC

From tightly coupled protocols to flexible, scalable, and developer-friendly frameworks – APIs have come a long way! Here's a quick timeline of how API architectures have evolved:

📌 CORBA (1980s) – Distributed object systems, complex but powerful

📌 SOAP (1998) – XML-based, standardized enterprise messaging

📌 REST (2000) – Simplicity and scalability through stateless communication

📌 JSON-RPC (2005) – Lightweight and developer-friendly

📌 GraphQL (2015) – Query exactly what you need, nothing more

📌 gRPC (2015) – High-performance RPC with HTTP/2 and protobuf

The diagram compares the evolution of API architectural styles. Each style has its own approach to data exchange, learning curve, and ideal use cases.

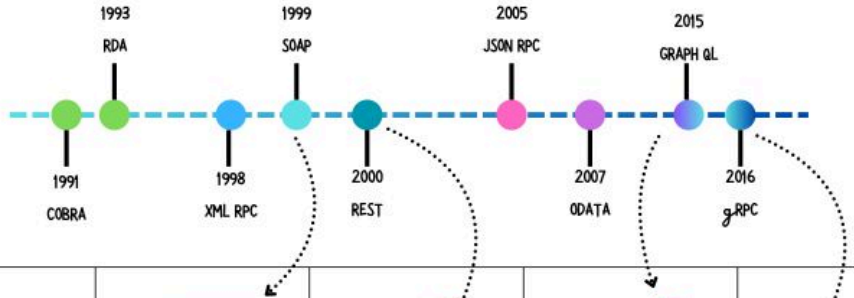
Explore the diagram to see how each style is best suited for different use cases, from legacy systems to modern architectures.

👉 What's your go-to API style in 2025? REST, GraphQL, gRPC, or something new?

#APIs #SoftwareArchitecture #TechTimeline #GraphQL #RESTAPI #gRPC #SOAP #CORBA #DeveloperJourney #CloudEngineering #SystemDesign #LinkedInTech

API Architectural Styles Comparison

source : Yogesh Patil



Feature	SOAP	REST	GraphQL	gRPC
Protocol	HTTP, SMTP, TCP	HTTP	HTTP	HTTP/2
Message Format	XML only	XML, JSON, HTML, Plain Text	JSON	Protobuf (binary)
Design Style	RPC (Remote Procedure Call)	Resource-based (stateless)	Query-based	Strict RPC
Schema Definition	WSDL	No strict schema	Strongly typed schema	Protobuf definition file (.proto)
Best For	Enterprise integrations, legacy	Public APIs, CRUD apps	Mobile apps, complex queries	Microservices, internal APIs
Security Standards	WS-Security	HTTPS, OAuth, JWT	HTTPS, custom	SSL/TLS, OAuth