

Some of the theoretical learning for Interview discussion in system Design

API Architectural style Comparison

There are many API architectural style, for ex:

SOAP, Rest, RPC etc. In addition to these styles

there are more like GraphQL etc.

	<u>SOAP</u>	<u>REST</u>	<u>RPC</u>	<u>GraphQL</u>
Format	<u>XML only</u>	<u>JSON</u> , HTML, XML, Plaintext	<u>JSON</u> , XML, Protobuffer, Thrift, Flatbuffer	<u>JSON</u>
Learning Curve	time required is high bcz of diffi culty level is high	Easy	Easy	Medium difficulty level
Community	Small	Large Community	Large Community	Growing
Use-cases	<ul style="list-style-type: none"> - Payment gateway - Identity mgmt. - CRM Solutions. - Financial & tele communication services - legacy system support. 	<ul style="list-style-type: none"> - Public API's - Simple resource driven apps. 	<ul style="list-style-type: none"> - Command & action oriented api's - high performance communication in massive micro-systems. 	<ul style="list-style-type: none"> - mobile api's - complex systems - micro services.

Ⓐ In general, whenever you have security on priority you always opt SOAP architectural style because it will help you by default.

Ⓑ Whenever you have to build public APIs and API is doing simple resource based communication b/w client & server, REST could be better choice.

Ⓒ RPC, could be better choice in scenario where performance matters most comparison to security & public openness. RPC could be focussed on closed system.

Ⓓ GraphQL is latest out of all other options and it is growing day & day because of handling very large complex system scenario.
