𝟔 𝐌𝐮𝐬𝐭 𝐊𝐧𝐨𝐰 𝐀𝐏𝐈 𝐀𝐫𝐜𝐡𝐢𝐭𝐞𝐜𝐭𝐮𝐫𝐚𝐥 𝐃𝐞𝐬𝐢𝐠𝐧𝐬  
  
🟣 REST (Representational State Transfer):  
An architectural style for designing networked applications, using HTTP requests to access and use data.  
  
Use Cases:  
\* Building web services that are lightweight, maintainable, and scalable.  
\* Integrating different systems or technologies in a flexible manner.  
  
🟣 GraphQL:  
A query language for your API, allowing clients to request exactly the data they need.  
  
Use Cases:  
\* Developing applications that require real-time data updates and complex, nested queries.  
\* Enhancing the performance of mobile applications by reducing the need for multiple API requests.  
  
🟣 SOAP (Simple Object Access Protocol):  
A protocol for exchanging structured information in web services, using XML.  
  
Use Cases:  
\* Implementing web services with high security and transactional reliability.  
\* Enterprise-level web services where ACID compliance and robustness are necessary.  
  
🟣 gRPC (gRPC Remote Procedure Calls):  
A high-performance, open-source universal RPC framework developed by Google.  
  
Use Cases:  
\* Microservices architectures where efficient communication between services is crucial.  
\* Streaming data between clients and servers in real-time applications.  
  
🟣 WebSockets:  
A communication protocol providing full-duplex communication channels over a single TCP connection.  
  
Use Cases:  
\* Building real-time, interactive applications like chat systems and live sports updates.  
\* Implementing notifications and real-time feeds in web applications.  
  
🟣 MQTT (Message Queuing Telemetry Transport):  
A lightweight messaging protocol designed for low-bandwidth, high-latency or unreliable networks.  
  
Use Cases:  
\* Internet of Things (IoT) applications, where devices need to communicate under bandwidth constraints.  
\* Messaging in mobile applications where network bandwidth is limited or variable.  
  
Photo credit: Nelson Djalo  
  
Ref:  
1: REST vs GraphQL vs gRPC: <https://lnkd.in/gwhq6XCZ>  
2. 18 System Design Concepts Every Engineer Must Know: <https://lnkd.in/g-39VDgi>  
  
[#api](https://www.linkedin.com/feed/hashtag/?keywords=api&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7135888626613174273) [#grpc](https://www.linkedin.com/feed/hashtag/?keywords=grpc&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7135888626613174273) [#graphql](https://www.linkedin.com/feed/hashtag/?keywords=graphql&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7135888626613174273) [#restapi](https://www.linkedin.com/feed/hashtag/?keywords=restapi&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7135888626613174273) [#systemdesign](https://www.linkedin.com/feed/hashtag/?keywords=systemdesign&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7135888626613174273)

Activate to view larger image,

