# Web Services

There are many strategies for data exchange. Primarily SOAP REST GraphQL gRPC.

**Understanding APIs as data exchange mechanism**

Data exchange has been a critical aspect of enterprise architecture since a long time. With the standardization of network protocols, data exchange became digitized using various standard protocols. This also lead to standardization of data formats starting from XML to text based JSON and binary formats such as Protobuf.

Today's trend is to have clients interacting with an API layer representing the application on the server-side. With the rise of machine learning and artificial intelligence, service-to-service interaction facilitated by APIs will emerge as the Internet's principal activity.

However, while network communication and data structures have become more conventional over time, there is still variety among API formats. There is no "one ring to rule them all.”

Now one important question we need to ask here is how do I pick the best API format to meet the need at hand? The answer is that it's a matter of understanding the benefits and limitations of the given format.

The aim of this article to help understand benefits and trade-offs of most popular API formats.

# gRPC

This data exchange mechanism was developed by google and later made open source. It’s a specification that can be implemented in different languages hence providing us interoperability.

It uses Protocol Buffer Binary format

<https://www.redhat.com/architect/apis-soap-rest-graphql-grpc>

# API

* Automated way to expose data and operations between organizations.
* API should obey to following properties:
  + **Reusable:** Designed for multiple clients, properly documented
  + **Unique:** for defined set of business functions and perimeter
  + **SLA Aware:** describing its level of availability, responsiveness, usage limits, etc.
  + **Secured:**
  + **Interoperable:** API should be exposed using universal technical Standards, the technology which API uses internally is completely agnostic and has no impact on its exposition.

## OAS/Swagger specification:

* OAS/Swagger is de-facto standard definition format to describe REST APIs
* This is important for **api discover and adoption as it summarizes in one standardized technical document and the list of operations supported by the APIs, the formal requirements to invoke the APIs such as Scope (OAuth) and the data fields in the input and output.**