# API Class Notes

#### 2022-06-21

API are interfaces plus a protocol

- RPC - CORBA – JAVARPC

- SOAP

- Restful

- GraphQL

Principles

- Open close

- Imperative versus declarative

## The Speech Contract

APIs to a certain extent are intuitive

## Ancient APIs – Remote Procedure Calls

1960s – Programming libraries – APIs looked like function calls

* Produced a mindset that APIs HAD to look function calls
* Imperative interface – each item in an interface is a command

1980s – distributed systems

* Corba – RPC Remote Procedure Calls – C++ type apps

1990s – Java distributed systems Java-RPC RPC-IIOP

End of this approach – rise of web based apps.

## SOAP – Simple Object Protocol

* Still thinking imperatively
* RPC but over HTTP
* SOA – let’s connect all computing via a networking layer
* https://www.cs.usfca.edu/~parrt/course/601/lectures/programming.by.contract.html
* UUDI – service discovery
* WSDL – “wizzle” Web Service Definition Language” – XML
* SOAP – Simple object access protocl
* https://stoplight.io/api-types/soap-api
* Archaic but it is still around.
* XML as the primary tool – very difficult to work
* Tooling was bad – hard to work with
* Lots of boiler plate
* RPC – imperative progaming
* Aspect Oriented Programming
* Web service over HTTP
* No clear way to implement SOA design principles – talking at the service level – SOAP is code and the transition is hard.

## REST

* The explosion of the web
* Push applications out to end users.
  + They can’t be trained on how to use the applications
  + Apps had to be intuitive- Iconic
  + People do not think in terms of RPC, they think in terms of doing things to domain objects
* https://restfulapi.net/