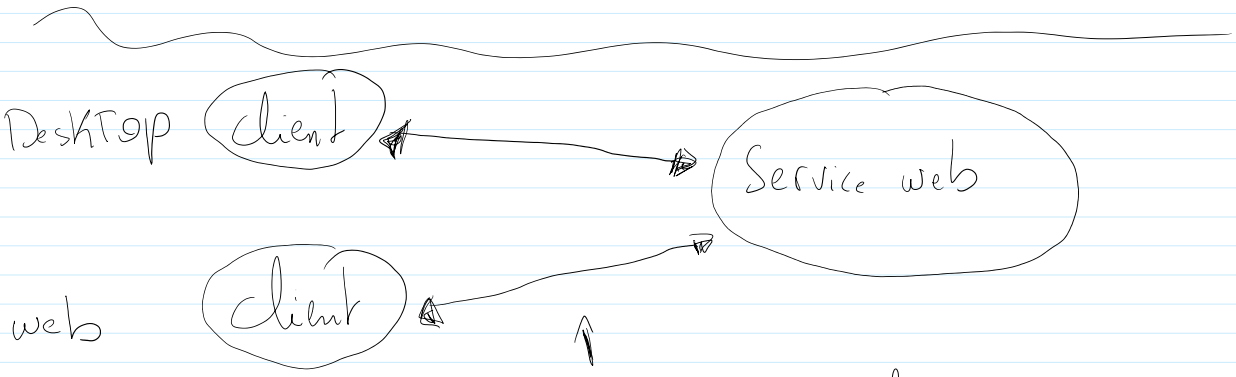
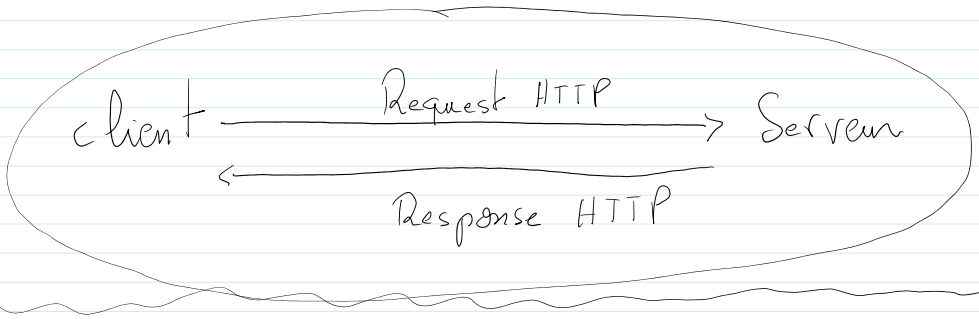
Client - Serveur

Front end /coté client	Back end /coté serveur	B.D.
• Desktop : C/C++, java, C#, js (HTML CSS) ...	- PHP - Js (Node.js) - JEE (java) - python - C++	• Relationnelle : MySql, oracle, SQL/Server • No Sql : Mongodb, firebase
• Mobile : 1) Android : java, Kotlin, xml... 2) Ios : objective C & swift 3) cross-platform : js, C# (HTML, CSS)		
• web : js (HTML et CSS)		





Plan

1)- mise à niveau (Revision)
- Protocole HTTP ✓
- js sans contexte

2)- js (Browser) → manipulation du DOM
→ Ajax

3)- web service → Node.js
→ Spring boot

I) Mise à niveau :

1) Protocole HTTP :

* deux acteurs : client & Serveur
⇒ c'est le client qui initie la communication

client —① envoi Request→ Serveur ②Traitement
←③ envoi Response

* infos nécessaire pour envoyer une request :
- Method (Get, Post, Delete ...)
  - GET: lire
  - POST: insérer
  - Delete: supprimer
  - PUT: modifier
- url → nom de domaine (DNS) / addresse Ip
- Port : Http ⇒ 80
  Https ⇒ 443
  ⇒ c'est obligatoire d'explicitement spécifier le port si ≠ default
- Path/ressource

⇒ Method protocole url : port / path
Http / Https

exemple : www.google.com →
Method ⇒ GET
protocole ⇒ Https
url ⇒ www.google.com
port ⇒ 443
path ⇒ /

* Format du Request :

method path version HTTP

Headers { clé : valeur
:

deux sauts de ligne

body / payload

* Format du Response :

version statuscode statusText

Headers { clé : valeur
:

deux sauts de ligne

body Response

I) - 2) js sans contexte

⇒ Langage de Programmation
orienté objet
interpreté