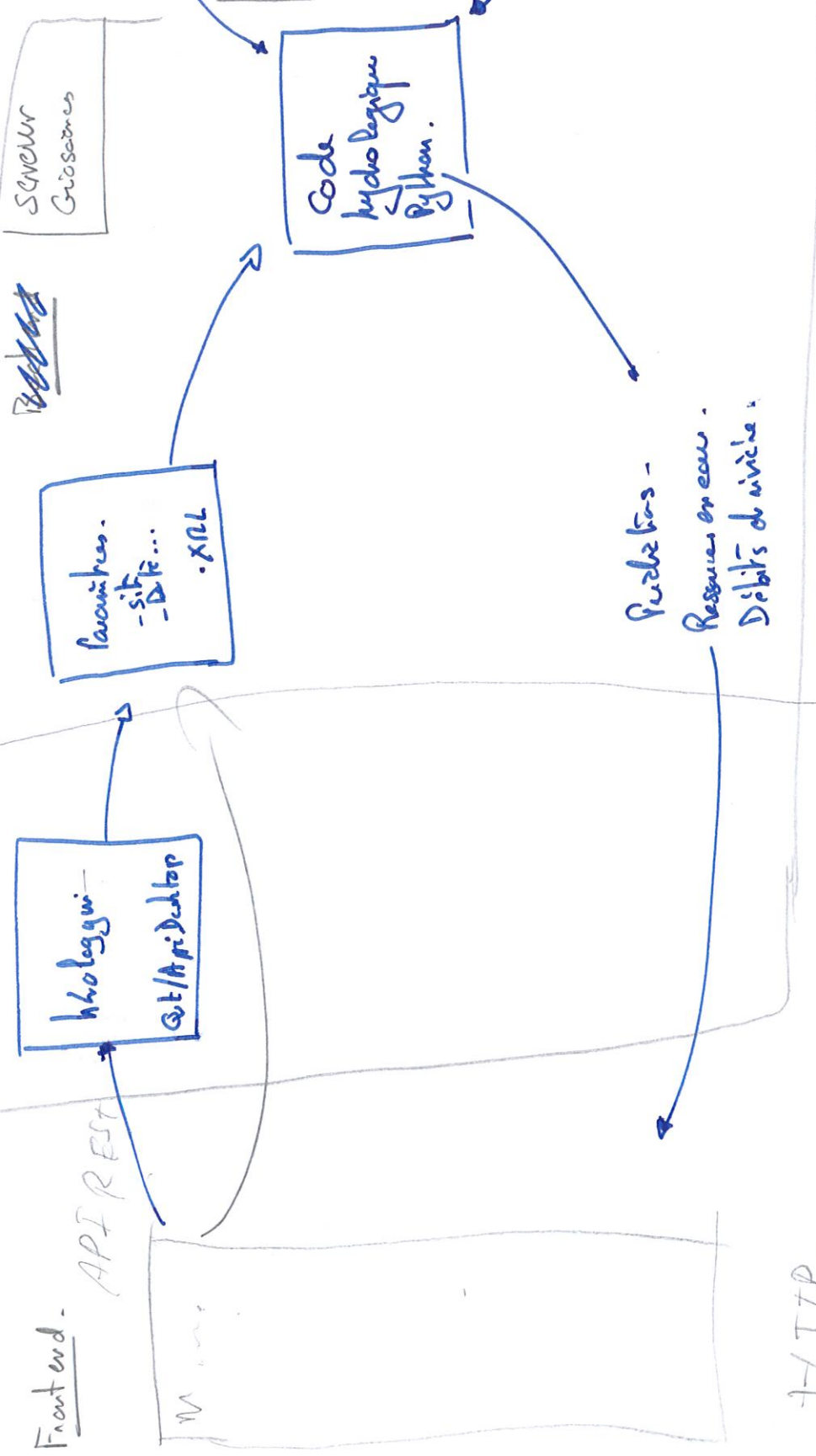


Utilisateur -
Frontend -

~~App~~ Frontend

Backend -

Web
Données Accessibles -



Champs Hydrologiques
- Paramètres
- Banque Hydro
.... - for

? champs Hydrologiques
.... - for

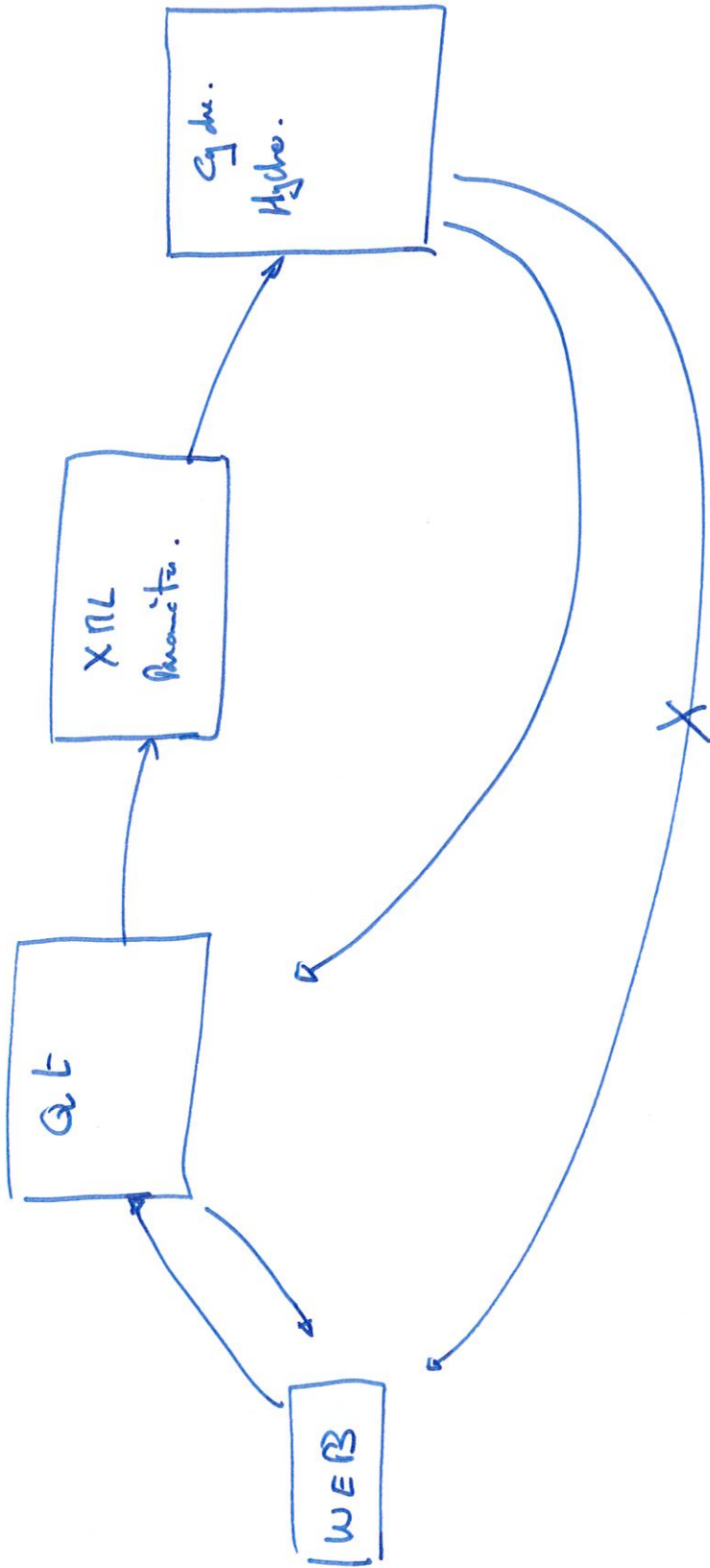
Productions -
Ressources en eau -
Débits d'irrigation

HTTP

GET - Read
POST - Write
DELETE

Formulaire

Liste



Py Age - 2 ans de developpement. 3000 lignes -

infant

Cydec (Python)

↑ structure
XML

batch

HydroNoddy (Python)

h2olab (C++) - 15 ans de developpement. $10^6/10^7$ lignes

cd1 cd2

fiction

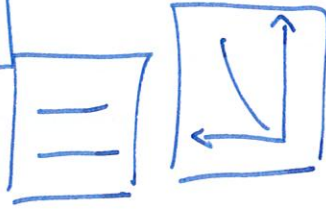
Facteur d'interface -

↑ interface
web

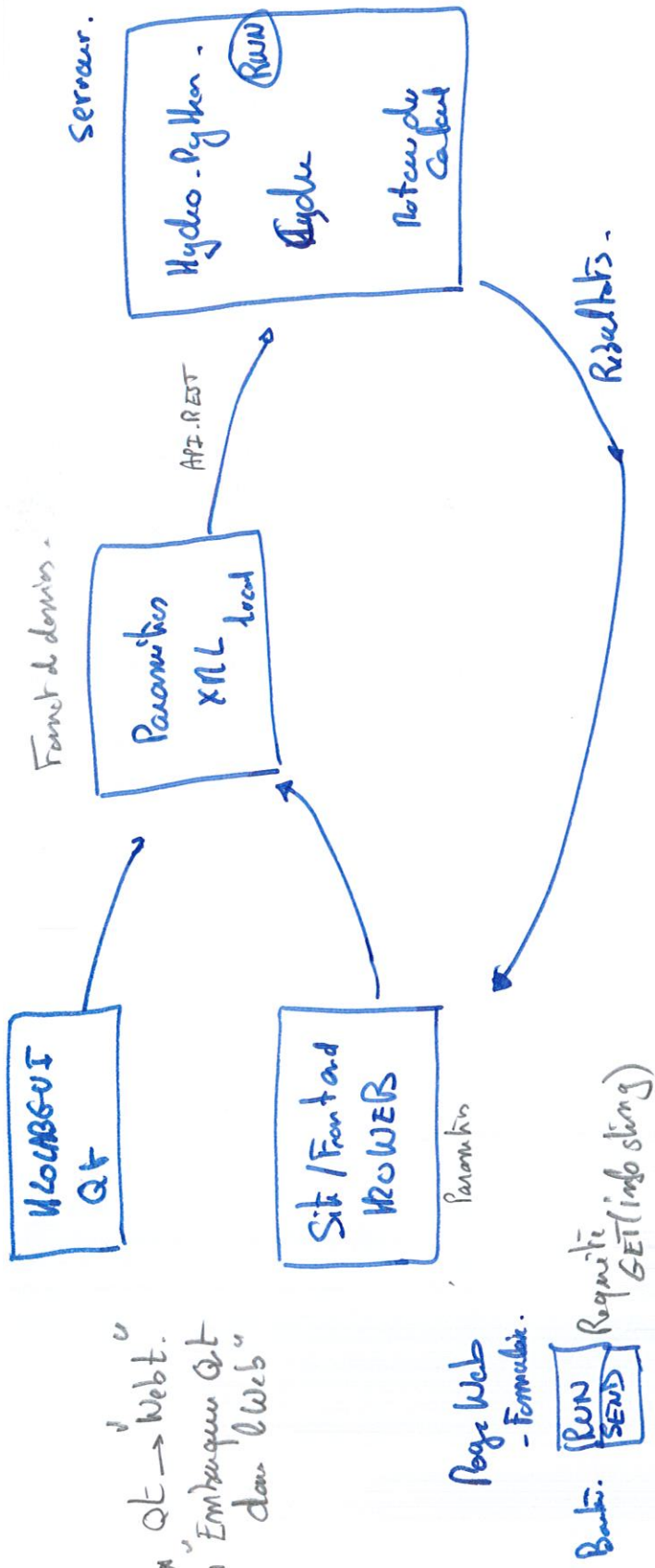
WEB

↑ interface
Qt

Qt
h2olabgui



* Qt → Web.
 # Embarquer Qt
 dans "Web"



Page Web
 - Formulaire.

But:

RUN	Requête
SEND	GET (info string)

(+) Avantages

- Standard dir vers API REST.

Back: Gyde.

Front: WROWEB

Formulaire: "Paramètres XNL"

- Gestion des cotes

(-) Inconvénient

- WOLABGUI à "redévelopper"

API - REST

Fonctions qui connectent le front au back.

FRONT

Page web (frontpage)

codé en html et en

javascript (instructions) → tous les dt types.

Angular : framework qui embarque
typescript, successeur de javascript
instructions avec la page.