



# L'histoire d'une architecture émergente

- Qui permet de gagner des € dès le premier jour de développement.

On a **gagné** des centaines de milliers d'euros dès notre **première journée de développement.**

Ceci est l'histoire d'un **MVP vraiment minimaliste.**



# Emmanuel-Lin TOULEMONDE



Consultant chez Octo Technology  
Sujet d'intérêt : Data, Architecte, Delivery, Tech Leading



01

**Il était une fois ...**



# L'énergie des raffineries

Les raffineries consomment **beaucoup de gaz et d'électricité.**

Gaz : **3% de la consommation française**

Electricité : **0.5% de la consommation française**  
( $\frac{1}{3}$  centrale nucléaire)

**Coût : Plusieurs centaines de millions d'€/an**





# Acheteur énergie au raffinage

**Négocier et opérer** les contrats d'achat d'énergie.

Faire des **nominations**

Acheter de la **capacité**

**Quoi ?**

Prédire la consommation du lendemain

En cours de journée, si la raffinerie consomme trop de gaz, acheter un droit supplémentaire

**Pourquoi ?**

Les erreurs de nominations représentent un surcoût

Les dépassements impliquent des pénalités

**Enjeux ?**

Des centaines de milliers d'euros

**Existant ?**

Manuel

Rien

# VADER et la Digital Factory

## VADER : **VA**lorisation des **D**onnées **É**nergétiques du **R**affinage

### Objectif :

Nominer et acheter de la capacité supplémentaire automatiquement.



### Moyen :

Utiliser les données historiques jusqu'au **quasi temps-réel** pour alimenter un modèle de prévision.

### Qui :

La **Total Digital Factory**.

### Le calendrier :





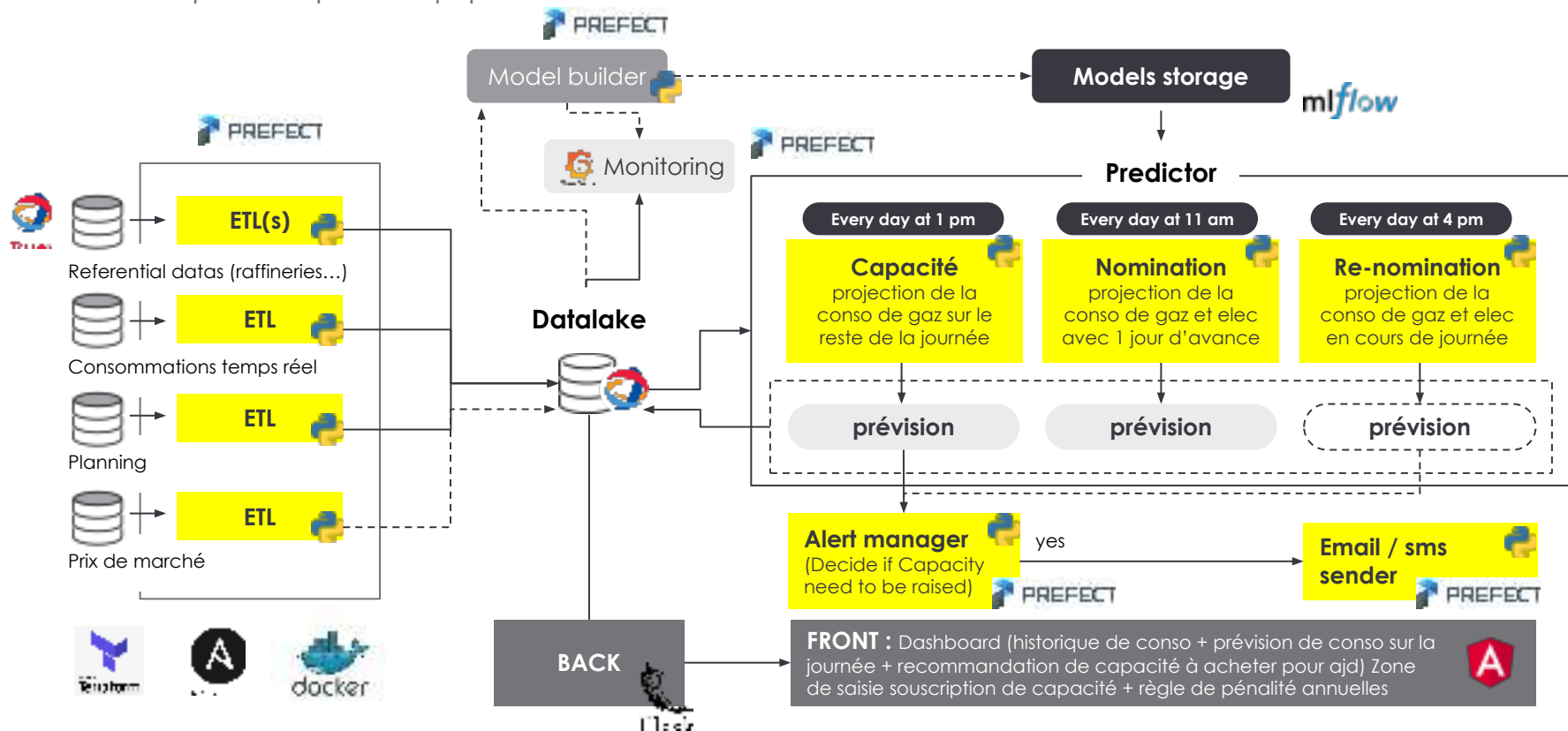
02

# Le plan



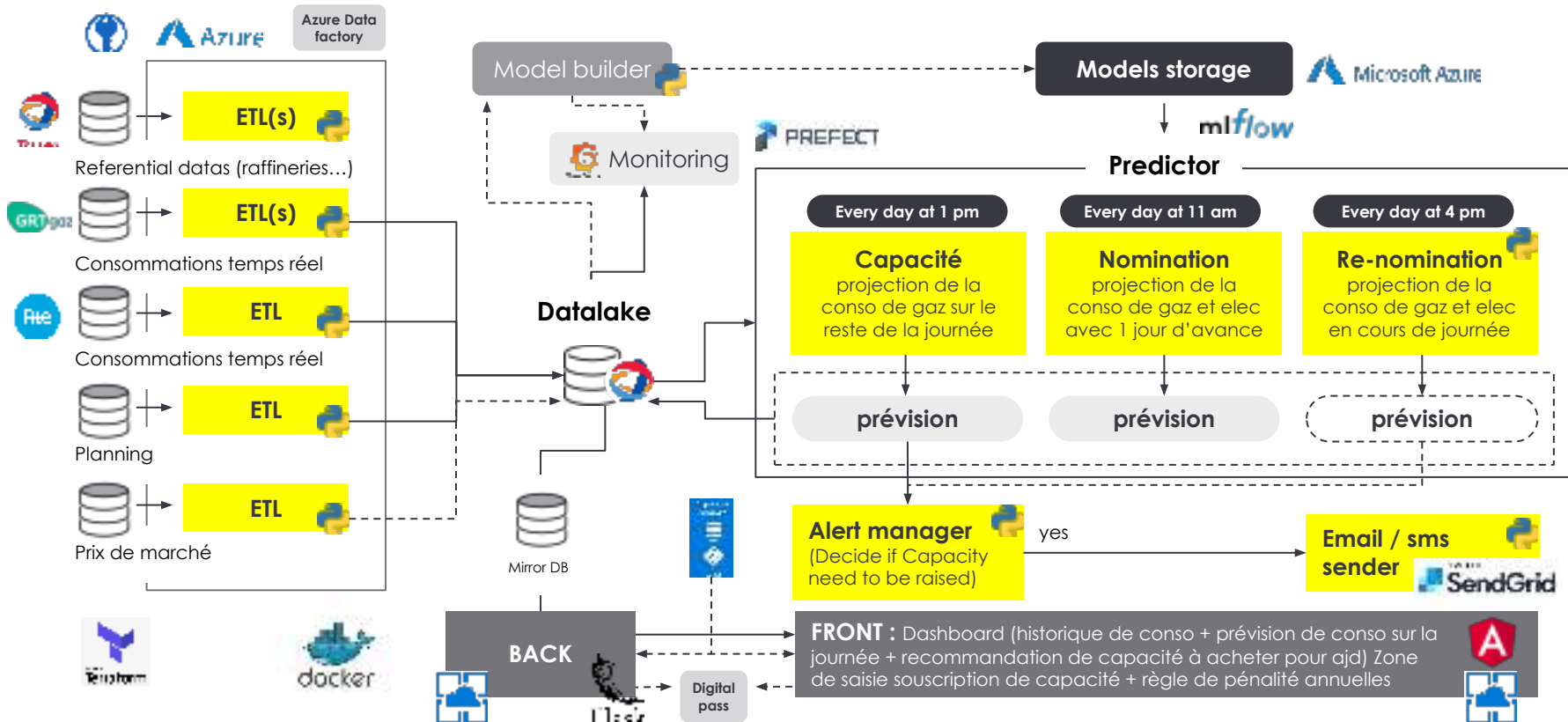
# Une architecture

Le schéma produit par l'équipe



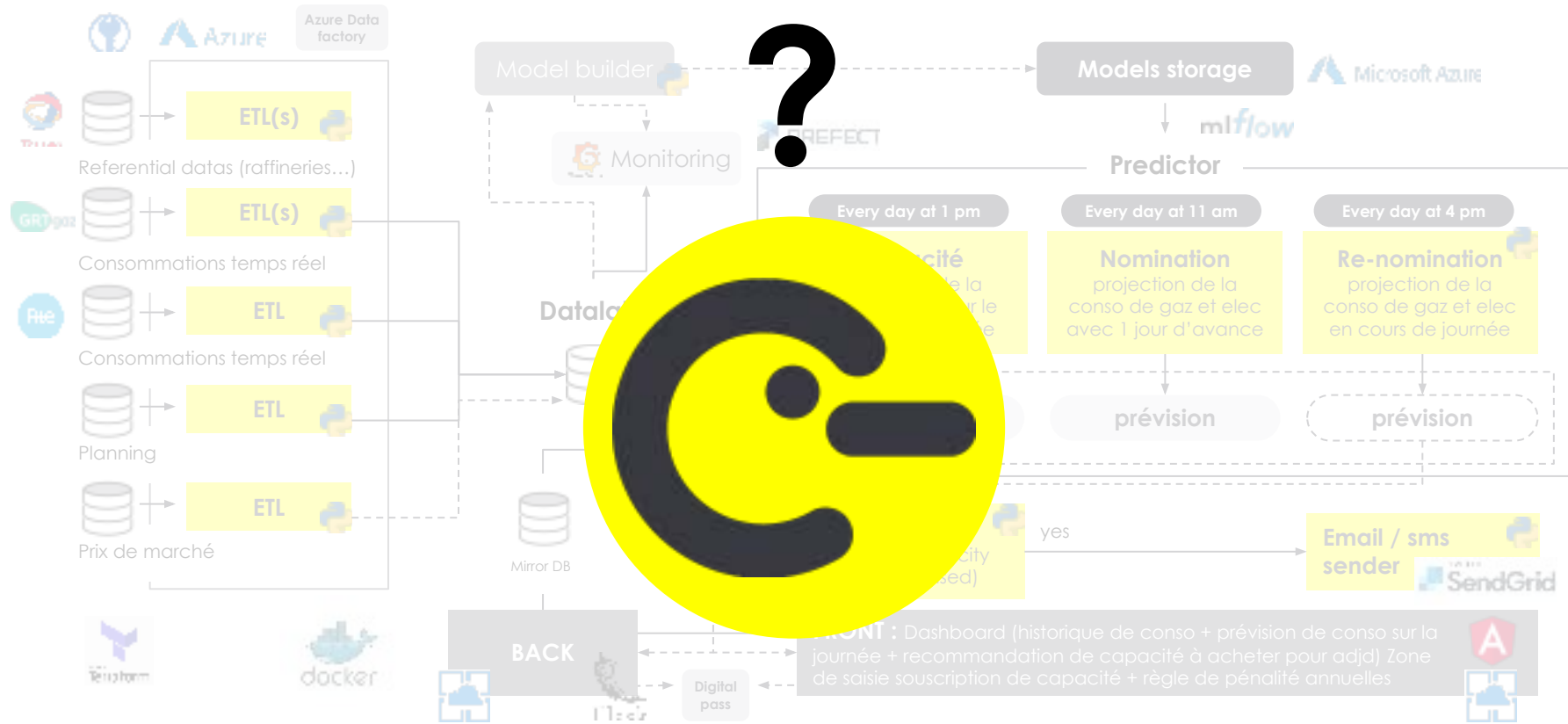
# Une architecture

## Une review



# Une architecture

## Le doute





03

# La réalité



# Découper en fonctionnalités et prioriser

Il y a **trois grandes fonctionnalités** :

- Capacité
- Nomination gaz
- Nomination électricité

Choix de la première fonctionnalité :  
**la capacité** car ils n'ont rien pour l'instant.

—  
**Critère de choix :**  
**Le plus simple pour faire mieux que l'existant.**  
—



# Construire la fonctionnalité capacité

Le *backlog* est long

Ce qu'il faut faire :

- **Intégrer les données de GRT Gaz**
  - Obtenir les droits sur le SFTP
  - Stocker les secrets
  - Tirer un flux du SFTP vers le data lake
  - Mettre les données dans une base de données
- **Faire un modèle de Machine Learning**
  - Faire de l'exploration
  - Construire une métrique
  - Faire de la préparation de données
  - Entraîner le modèle
  - Évaluer
  - Faire de l'A/B testing
  - Industrialiser
  - Mettre en production
- **Restituer les prédictions à l'utilisateur**
  - Faire un front
  - Faire un back
  - Intégrer une solution d'authentification



**Livraison dans 4 à 6 semaines**



# Construire la fonctionnalité capacité

Trouver plus simple

Ce qu'il faut faire :

- Intégrer les données de GRT Gaz
  - Obtenir les droits sur le SFTP
  - Stocker les secrets
  - Tirer un flux du SFTP vers le data lake
  - Mettre les données dans une base de données
- **Faire un modèle de Machine Learning**
  - ~~◦ Faire de l'exploration~~
  - ~~◦ Construire une métrique~~
  - ~~◦ Faire de la préparation de données~~
  - ~~◦ Entraîner le modèle~~
  - ~~◦ Evaluer~~
  - ~~◦ Faire de l'A/B testing~~
  - ~~◦ Industrialiser~~
  - ~~◦ Mettre en production~~
  - **Une règle métier (moyenne des 5 dernières heures \*24)**
- Restituer les prédictions à l'utilisateur
  - Faire un front
  - Faire un back
  - Intégrer une solution d'authentification



**Livraison dans 3 à 4 semaines**

**Critère de choix :**  
**Est-ce vraiment le moyen**  
**le plus simple de faire ?**



# Construire la fonctionnalité capacité

Trouver plus simple

Ce qu'il faut faire :

- Intégrer les données de GRT Gaz
  - Obtenir les droits sur le SFTP
  - Stocker les secrets
  - Tirer un flux du SFTP vers le data lake
  - Mettre les données dans une base de données
- Faire un modèle de Machine Learning
  - Faire de l'exploration
  - Construire une métrique
  - Faire de la préparation de données
  - Entraîner le modèle
  - Evaluer
  - Faire de l'A/B testing
  - Industrialiser
  - Mettre en production
  - Une règle métier (moyenne des 5 dernières heures \* 24)
- Restituer les prédictions à l'utilisateur
  - Faire un front
  - Faire un back
  - Intégrer une solution d'authentification
  - **Faire un screenshot et l'envoyer par teams**



**Livraison dans 2 semaines**

**Critère de choix :  
Est-ce vraiment le moyen  
le plus simple de faire ?**





# Construire la fonctionnalité capacité

Trouver plus simple, pourquoi pas manuellement

Ce qu'il faut faire :

- **Intégrer les données de GRT Gaz**
  - ~~Obtenir les droits sur le SFTP~~
  - ~~Stocker les secrets~~
  - ~~Tirer un flux du SFTP vers le data lake~~
  - ~~Mettre les données dans une base de données~~
  - **Extraire les données à la main**
- Faire un modèle de Machine Learning
  - ~~Faire de l'exploration~~
  - ~~Construire une métrique~~
  - ~~Faire de la préparation de données~~
  - ~~Entraîner le modèle~~
  - ~~Evaluer~~
  - ~~Faire de l'A/B testing~~
  - ~~Industrialiser~~
  - ~~Mettre en production~~
  - Une règle métier (moyenne des 5 dernières heures \* 24)
- Restituer les prédictions à l'utilisateur
  - ~~Faire un front~~
  - ~~Faire un back~~
  - ~~Intégrer une solution d'authentification~~
  - Faire un screenshot et l'envoyer par mail



**Livraison dans 1 journée**

**Critère de choix :**  
**Est-ce vraiment le moyen**  
**le plus simple de faire ?**

# On est en production manuelle dès le jour 1

ET [redacted] 14:02:11 (17/01/2021)  
[redacted] pas de problème niveau cjd :

|   | code   | name       | consumption 4 hours | projected_capacity | over_capacity |
|---|--------|------------|---------------------|--------------------|---------------|
| 0 | L18440 | 12081      | 867.608489          | 1000.000000        | False         |
| 1 | L18445 | Remande    | 75.087995           | 448.525795         | False         |
| 2 | L18580 | Granquatre | 31855.7577          | 3278.757522        | False         |
| 3 | L18716 | Deul       | 2805.932089         | 17348.477900       | False         |
| 4 | L18726 | Pesverce   | 226.670979          | 1308.475376        | False         |
| 5 | L18878 | Daupin     | 809.040332          | 1817.182722        | False         |
| 6 | L18955 | Reyde      | 500.171965          | 3806.531400        | False         |
| 7 | L18988 | guzn       | 1104.181081         | 3268.181484        | False         |
| 8 | L18995 | Carrey     | 45.781154           | 377.589364         | False         |
| 9 | L11025 | Parabre    | 3000.0000           | 8.000000           | False         |

10 tables from jous, N'arr, and David

ET [redacted] 1:53 PM - Edited  
[redacted] 10 jours pas de dépassement en vue :

|   | meter_number | projected_consumption | capacity | over_capacity |
|---|--------------|-----------------------|----------|---------------|
| 0 | L18442       | 3225.923585           | 4658.0   | False         |
| 1 | L18463       | 197.685341            | 4708.2   | False         |
| 2 | L18588       | 2847.422762           | 2892.0   | False         |
| 3 | L18716       | 28222.286898          | 28798.0  | False         |
| 4 | L18726       | 1455.750258           | 2432.0   | False         |
| 5 | L18878       | 2561.854449           | 6598.8   | False         |
| 6 | L18958       | 2191.779721           | 4998.0   | False         |
| 7 | L18989       | 6896.239636           | 11819.0  | False         |
| 8 | L18995       | 345.871539            | 598.0    | False         |
| 9 | L11025       | 424.152128            | 328.0    | True          |



04

# Une architecture émergente



# Architecture 0

Jour 1 : 11h

PC d'un dev



Predictor



Extract



JPEG

Utilisateur



Livrer de la  
valeur :

Avec une solution  
minimaliste



# Architecture 1

Jour 1 : 12h



Gagner du temps :

Versionner le code pour travailler en équipe

Cloud



PC d'un dev



Predictor

Utilisateur



Extract





# Architecture 2

Jour 2 sur 60

Cloud



PC d'un dev



Utilisateur



Livrer de la valeur :

Retirer une douleur utilisateur



# Architecture 3

Jour 4 sur 60

Cloud

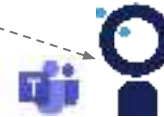


PC d'un dev



Predictor

Utilisateur



Gagner du temps :

Se donner confiance avec une CI



# Architecture 4

Jour 6 sur 60

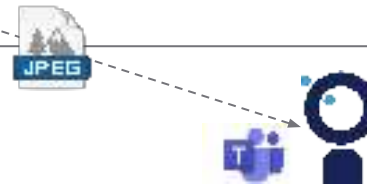
Cloud



PC d'un dev



Utilisateur



Gagner du temps :

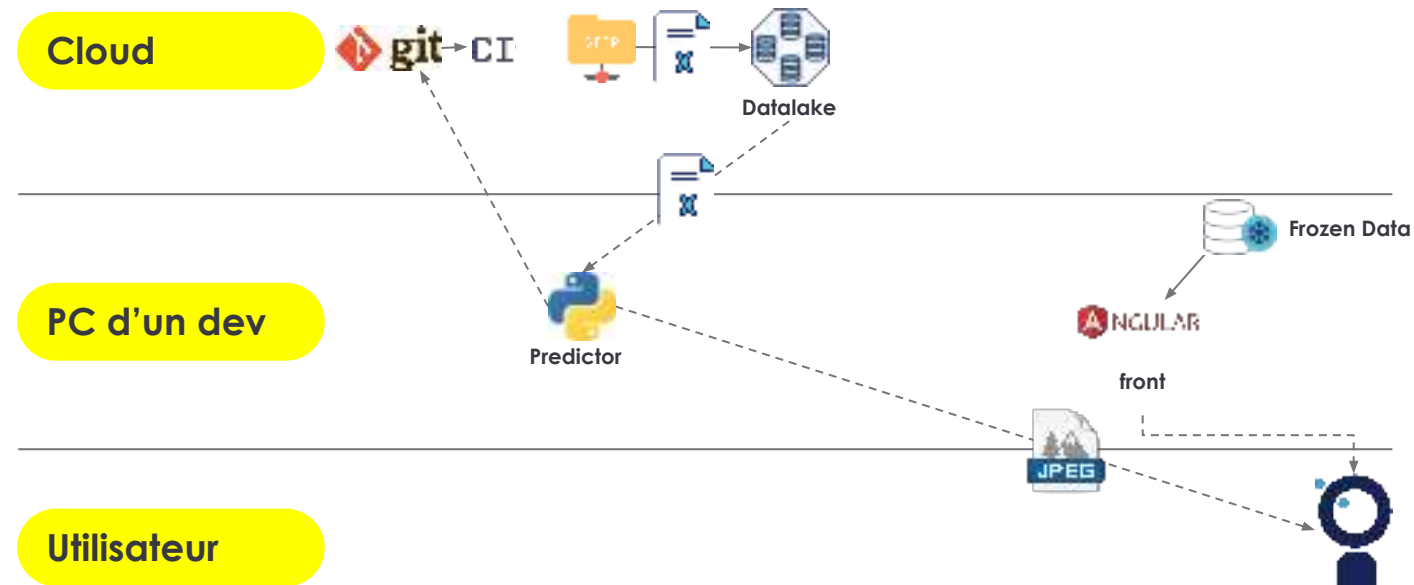
Faciliter une tâche pour les devs





# Architecture 5

Jour 7 sur 60



Gagner du temps :

Prendre du feedback rapide



# Architecture 6

Jour 8 sur 60

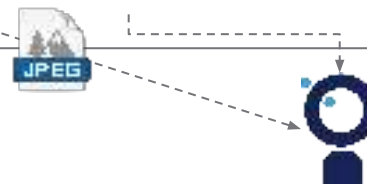
Cloud



PC d'un dev



Utilisateur



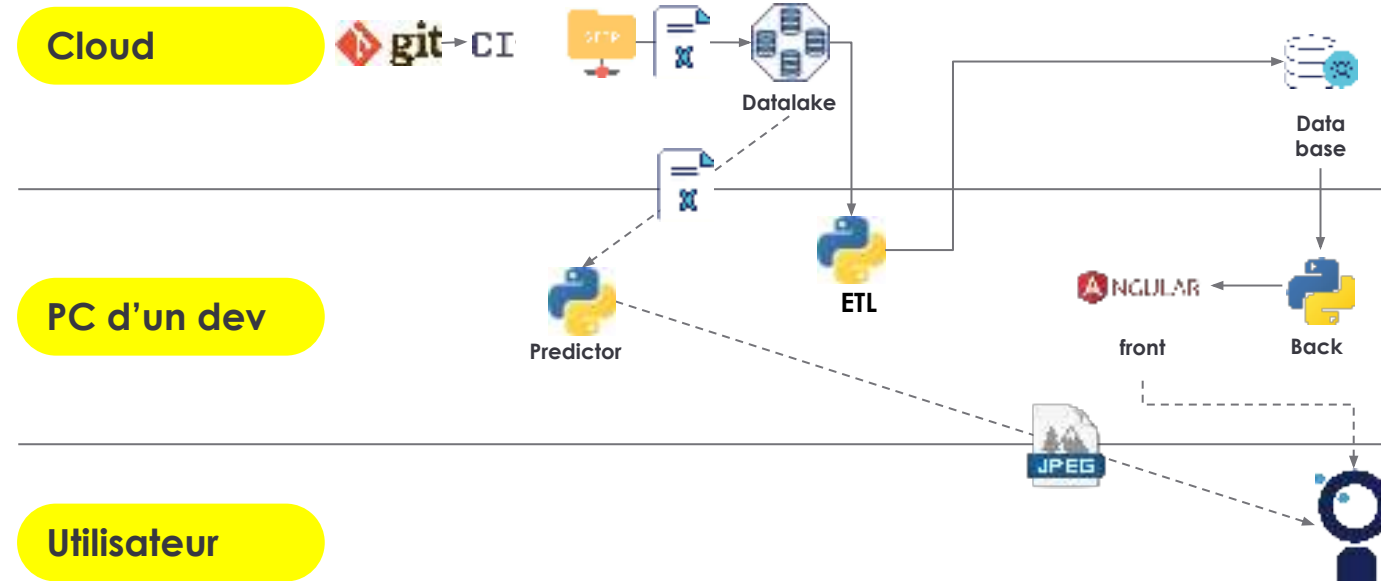
Gagner du temps :

Créer un BACK pour découpler



# Architecture 7

Jour 14 sur 60



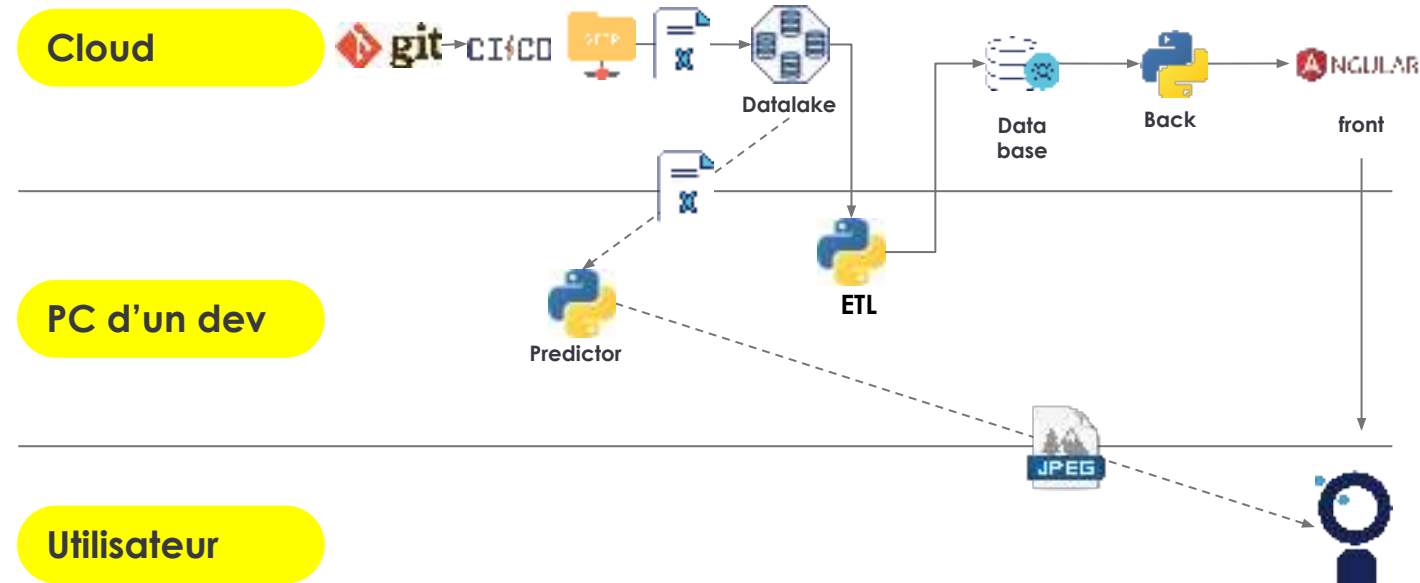
Apporter de la valeur :

Mettre des vraies données



# Architecture 8

Jour 20 sur 60



Apporter de la valeur :

Rendre le front accessible



# Architecture 9

Jour 25 sur 60

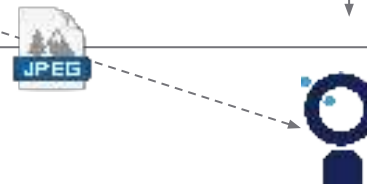
Cloud



PC d'un dev



Utilisateur



Gagner du temps et apporter de la valeur :

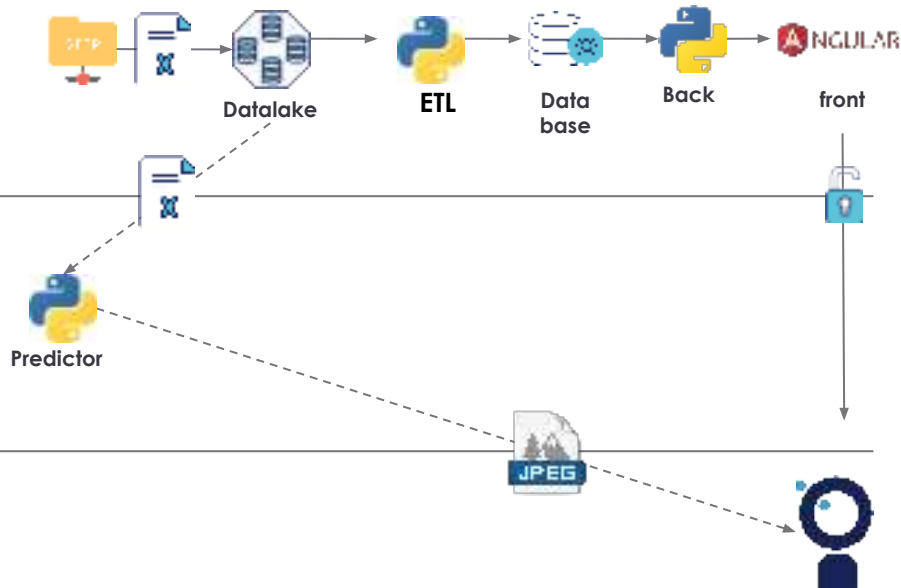
**Automatiser l'ingestion de données**



# Architecture 10

Jour 31 sur 60

Cloud



PC d'un dev

Utilisateur



Apporter de la valeur :

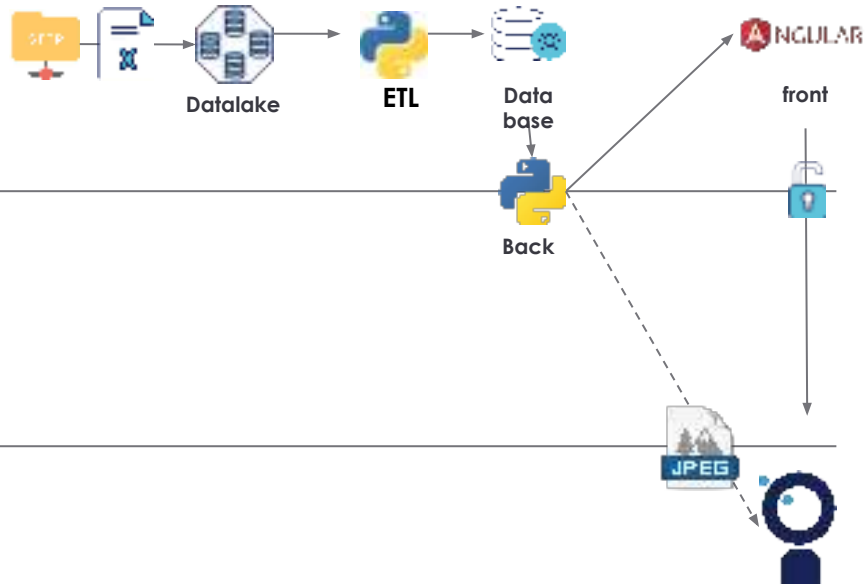
Rendre la solution pérenne avec une brique d'authentification



# Architecture 11

Jour 33 sur 60

Cloud



PC d'un dev

Utilisateur



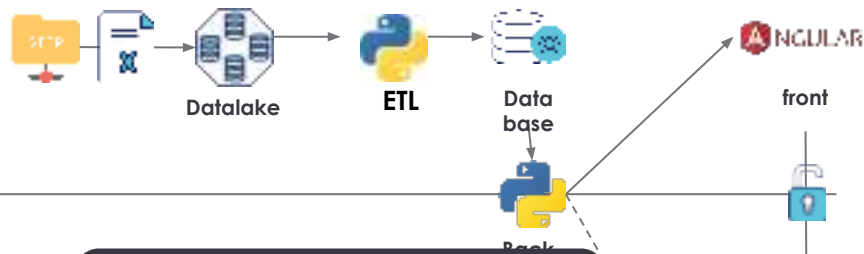
Gagner du temps :

Refactorer



# Architecture 12

Cloud



PC d'un dev

Utilisateur



Gagner du temps :

Ne pas résoudre de problème que l'on n'a pas encore rencontré

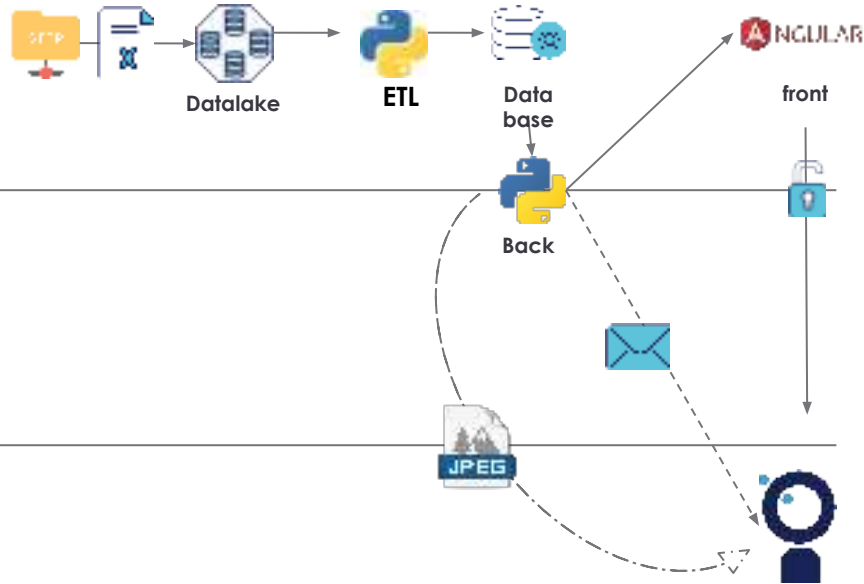




# Architecture 13

Jour 37 sur 60

Cloud



PC d'un dev

Utilisateur



Apporter de la valeur :

Ajouter la nomination



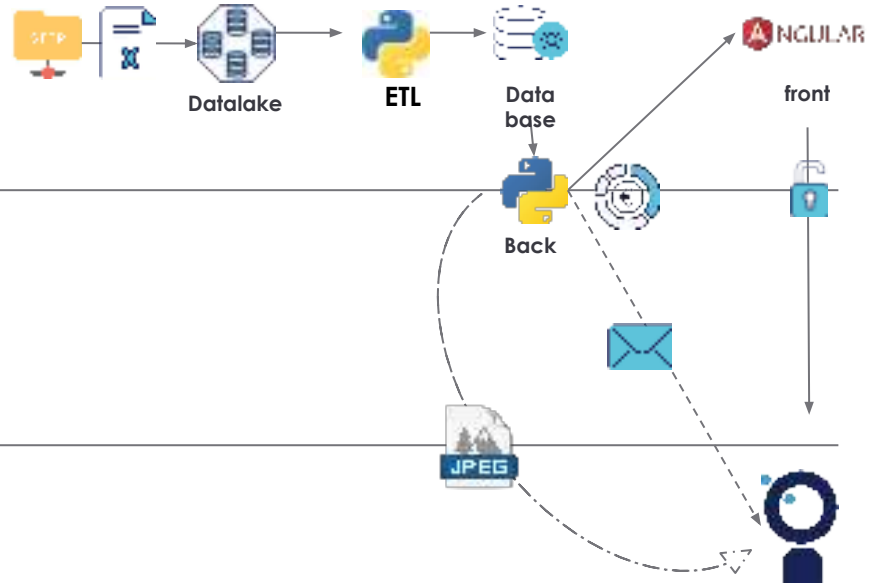
# Architecture 14

Jour 40 sur 60

Cloud

PC d'un dev

Utilisateur



Gagner du temps :

Automatiser la mesure de la valeur



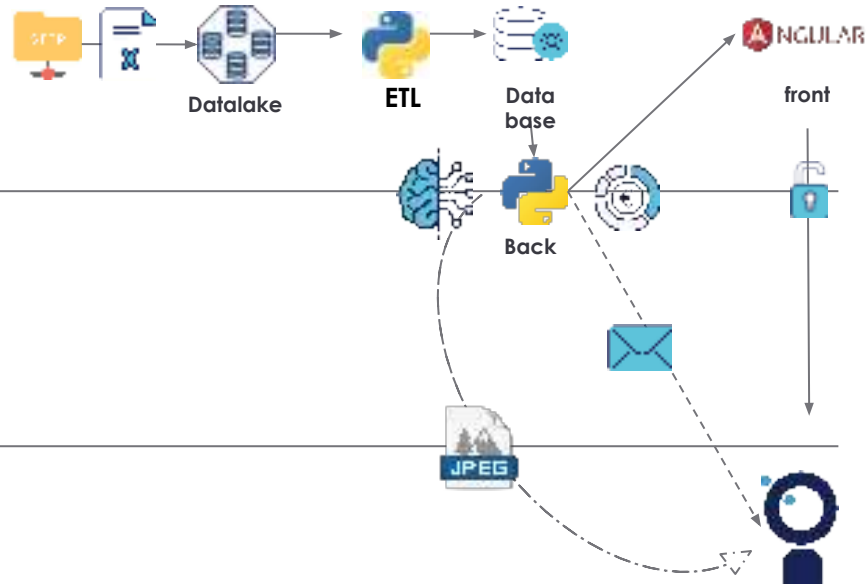
# Architecture 15

Jour 45 sur 60

Cloud

PC d'un dev

Utilisateur



Apporter de la valeur :

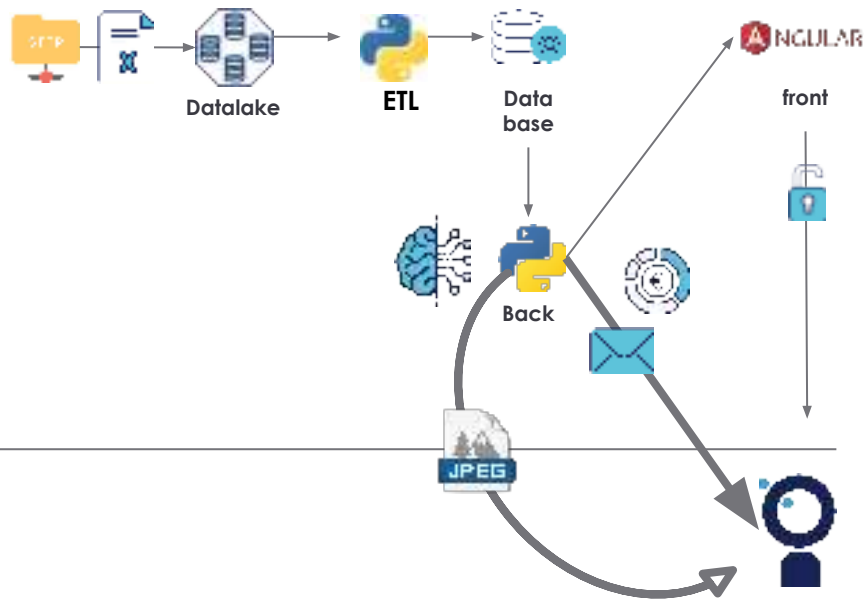
En mettant en place des modèles de *Machine Learning*



# Architecture 16

Jour 55 sur 60

Cloud



Utilisateur



Gagner du temps et apporter de la valeur :

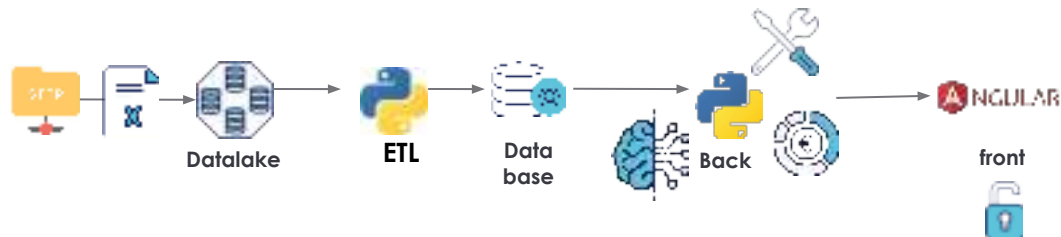
En automatisant les mails



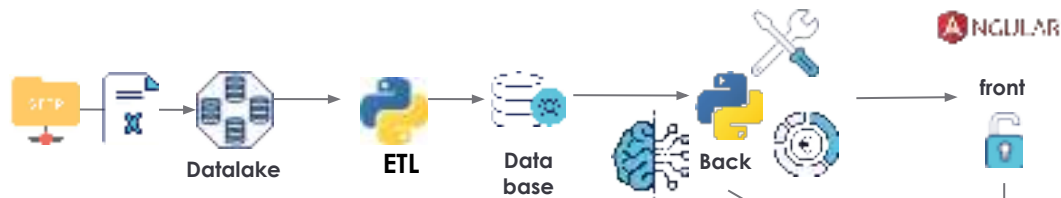
# Architecture 17

Jour 60

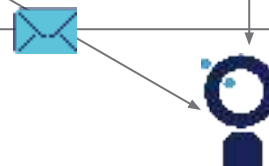
DEV



Prod



Utilisateur



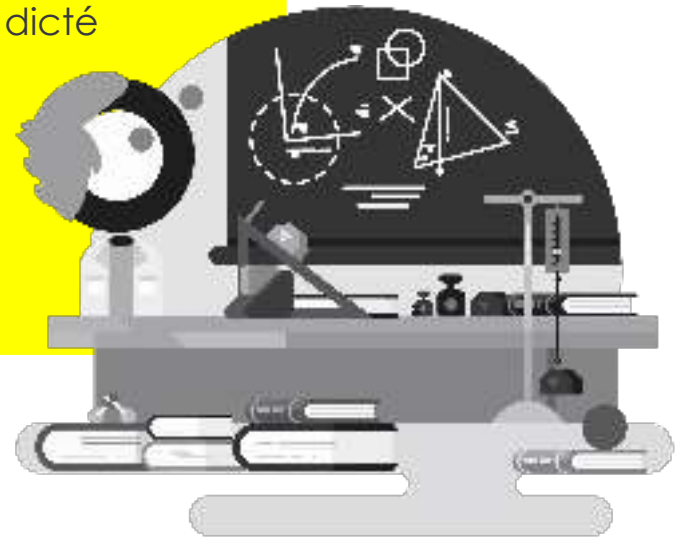
Apporter de la valeur :

Être plus stable en ayant 2 envs



# Architecture 18, 19, ... 50, .... 80, etc.

- Nous sommes capables de dessiner au moins un schéma d'architecture par jour.
- Chaque petit changement est un **choix conscient**, dicté par la réalité et non pas selon des hypothèses.
- Ces choix peuvent être documentés dans un *Architecture Decision Record*.





Voir cette architecture comme une vision pas comme quelque chose à implémenter.



« Une vision qui ne s'accompagne pas d'actions n'est qu'un rêve. Une action qui ne découle pas d'une vision c'est du temps perdu. Une vision suivie d'action peut changer le monde. » - Nelson Mandela



05

# Prise de recul





# Zoom sur la complexité de la simplicité

- Cela a l'air simple, mais ça l'est pas !
- Il faut se **“battre” contre** :
  - L'équipe de dev et soi même : pour ne pas faire de l'état de l'art
  - Le métier : qui veut une IA super puissante
  - L'IT : qui a des réflexes parfois surdimensionnés
  - Les architectes : qui veulent mettre tout l'outillage data
  - La culture : qui pardonne plus les échecs quand les outils sont complexes
- **Commencer simple c'est dur, continuer à faire simple encore plus.**
- Faire simple c'est ce qui prend **90% de l'énergie du tech lead**



---

**“Je vous écrit une longue lettre  
car je n'ai pas le temps d'en écrire  
une courte”.**

**- Blaise Pascal**

---



# Chaque élément d'architecture doit répondre à 2 objectifs



## Apporter de la valeur

Développer des fonctionnalités

Être utilisé dès le jour 1

Être centré utilisateur

Être pérenne

Commencer manuellement



## Gagner du temps

Faire simple

Automatiser au fur et à mesure

Mesurer

Se donner confiance

Avoir l'autonomie sur l'infra

# Les bénéfices de cette approche



## Côté utilisateurs

€ gagnés dès le jour 1

Confiance

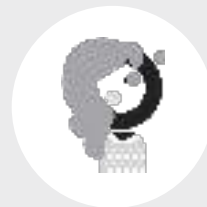
Boucle de  
feedback rapide



## Côté équipe

Satisfaction forte

Eviter un sprint 0 d'Ops



## Côté IT Total

Tous les choix  
d'architecture justifiés  
par un besoin

Minimum à maintenir



06

# La morale de l'histoire



# Takeaways

- **Faire simple** pour éviter le sur engineering et atteindre vos objectifs organisationnels plus vite.
- Être agile est un état d'esprit, **l'architecture émergente est l'incarnation de l'agilité en architecture logicielle.**
- Le **schéma d'architecture** du cadrage sert de **vision.**
- Commencer simple, puis faire des petites actions vers cette vision.



# DUCK CONF