

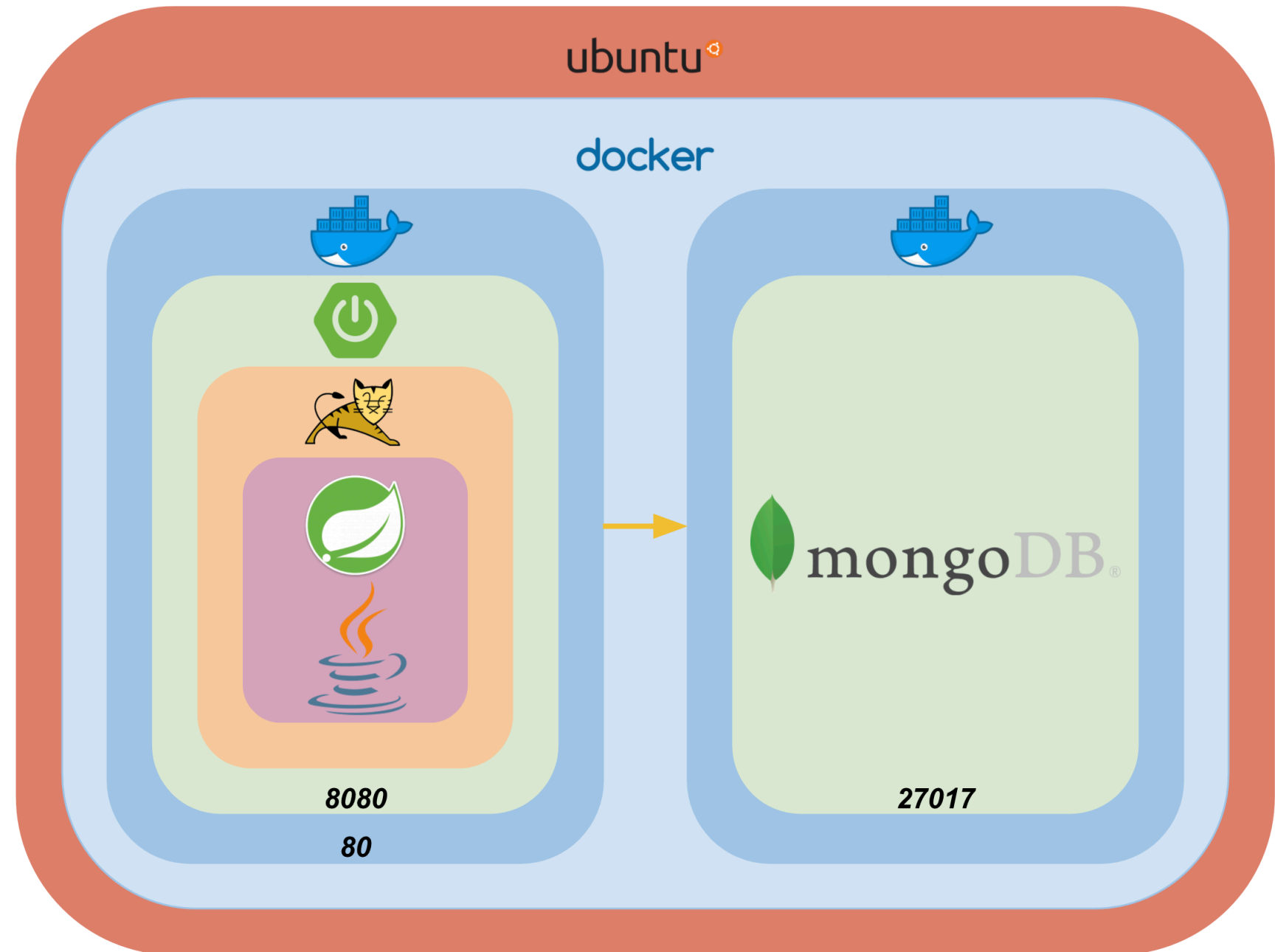


# Projet Cloud - Polytech 5A

Présenté par L. DELORME, N. KASPRZYK, F. PARIS et T. PRIEUR-DREVON

# Architecture globale

- Serveur Ubuntu
  - Service Docker
    - 2 containers Docker
      - Application (API)
        - MongoDB
      - I lien Docker
        - Application ⇒ MongoDB
    - 2 redirections de port Docker
      - Application (80 ⇒ 8080)
      - MongoDB (27017 ⇒ 27017)

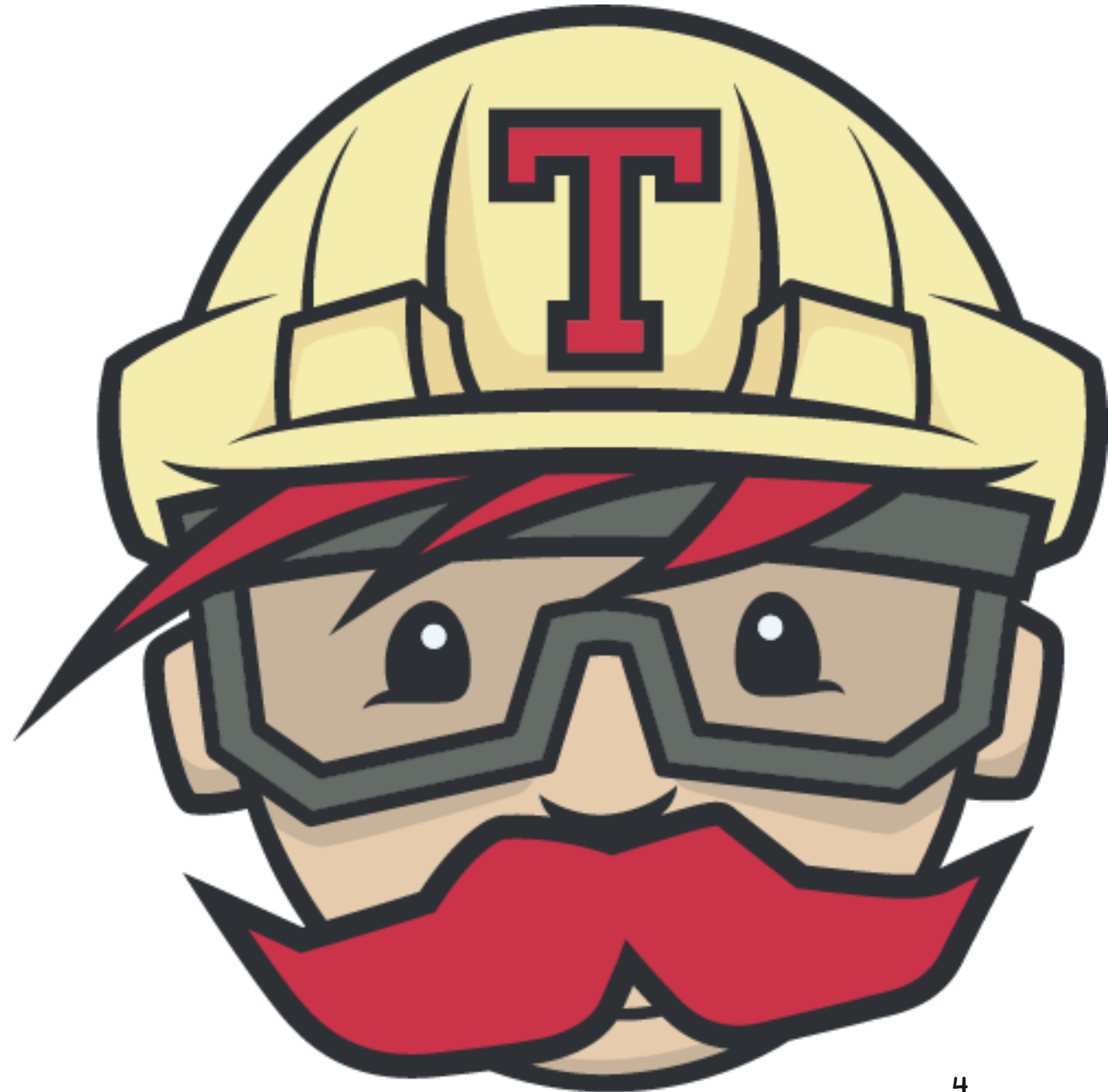


# Choix techniques

Thématique	Technologie(s) utilisée(s)	Alternative(s) ?	Avantage(s)	Inconvénient(s)	Pertinent ?
Structure de l'API	SpringBoot	NodeJS	Robuste et très performant	"Lourd" à mettre en place	✓
Sérialisation & désérialization du contenu	Jackson	JSON-B, Gson	Personnalisation puissante et intégration native à SpringBoot	Trop lourd pour de petits projets	✗
Getters & setters des entités	Lombok	IDEs	Rien à faire !	-	✓
BDD	MongoDB	MySQL, PostgreSQL & Oracle	Simple, flexible et évolutif (passage à l'échelle)	Aucune contrainte et normalisation des données	✓
Wrapper MongoDB	Jongo	Spring Data MongoDB	Excellente encapsulation du driver MongoDB	Moteur des requêtes à définir	✓ ✗
Tests	JUnit, AssertJ & Mockito	TestNG	Écriture des tests simple et mocks configurables	-	✓
Continuous Integration	Travis-CI	Jenkins & Circle-CI	Gratuit, builds illimités & intégration de services tiers	Builds parfois "longs"	✓
Continuous Deployment	Docker	Heroku & Ansible	Portabilité des containers & déploiements automatiques	-	✓

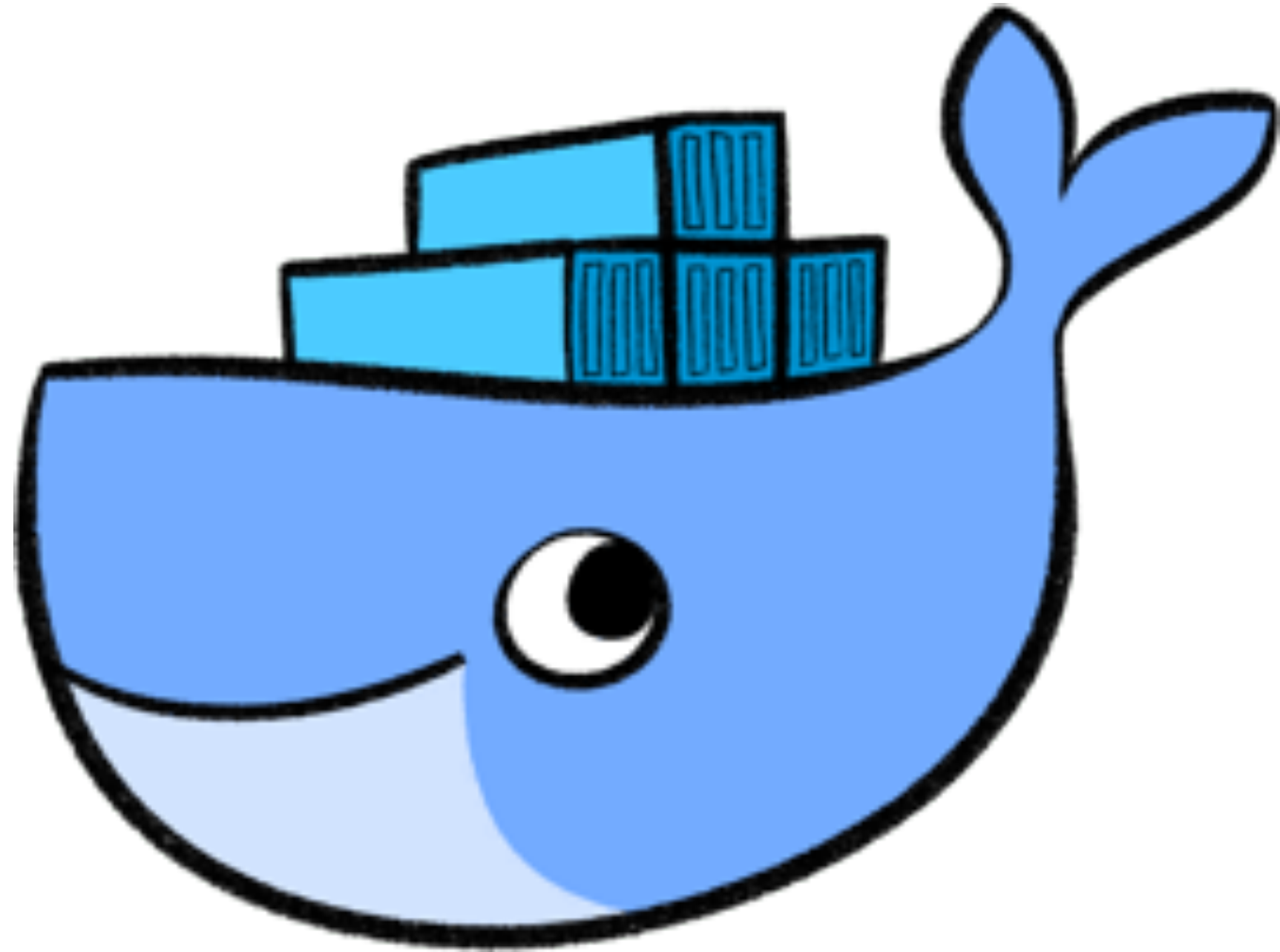
# Pourquoi le Continuous Integration ?

- Compilation et exécution des tests automatiquement à chaque
  - Commit
  - Pull Request
  - Nouvelle branche
- Notification de l'état du dernier build
- Détection et correction des problèmes d'intégration au fur et à mesure
- **Version dev' toujours fonctionnelle pour une démonstration**



# Pourquoi le Continuous Deployment ?

- Prend les fonctionnalités validées durant le CI
- Déploiement automatique de la solution dans un environnement type de production
- QA peut tester simplement sans manipulation technique
- **Déploiement très rapide des nouvelles versions en production**

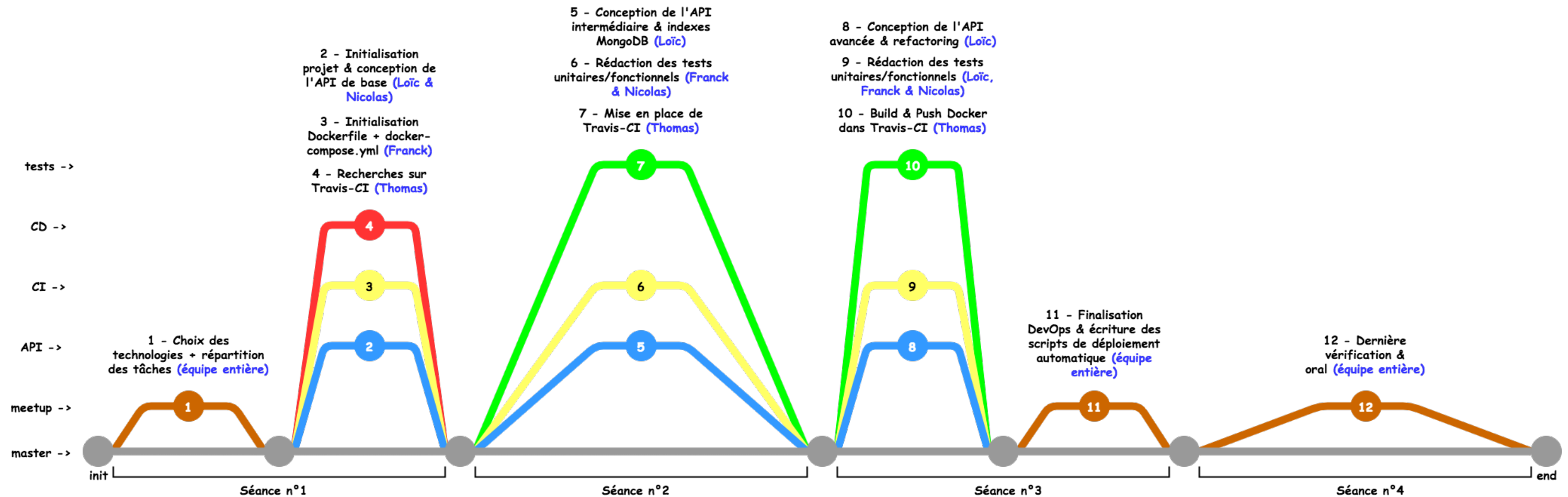




# Démonstration



# Organisation du groupe au fil du temps



# Difficultés rencontrées

- MongoDB
  - Écriture des requêtes
  - Gestion des types de données (dates et positions)
- Jackson
  - Sérialisation des BigDecimal non fonctionnelle
- Tests client
  - Comportements parfois aléatoires
  - Surprises...



# Retours sur le projet

- Projet très intéressant
  - Liberté des choix techniques
  - Plusieurs aspects (développement et DevOps)
  - Mise en situation du métier d'ingénieur
- Propositions d'amélioration
  - Tests client
  - Pas d'accès intégral au serveur (firewall, etc...)
  - Pas assez de DevOps ( $\approx 20\%$  du temps total)
  - Pas d'apport sur l'administration du serveur

# Remerciements

→ M. Antoine Caron, encadrant du projet

