

EasySave Groupe 1



CESI 
ÉCOLE D'INGÉNIEURS

Alban Clavo
Matheo Pinget
Evan Joasson
Jonas Mionnet



Contexte	1
Gestion de projet	2
Versioning	3
Architecture	4
Diagrammes UML	5
Version 1	6
Version 2	7
Version 3	8
Documentation	9
Conclusion	10

Contexte

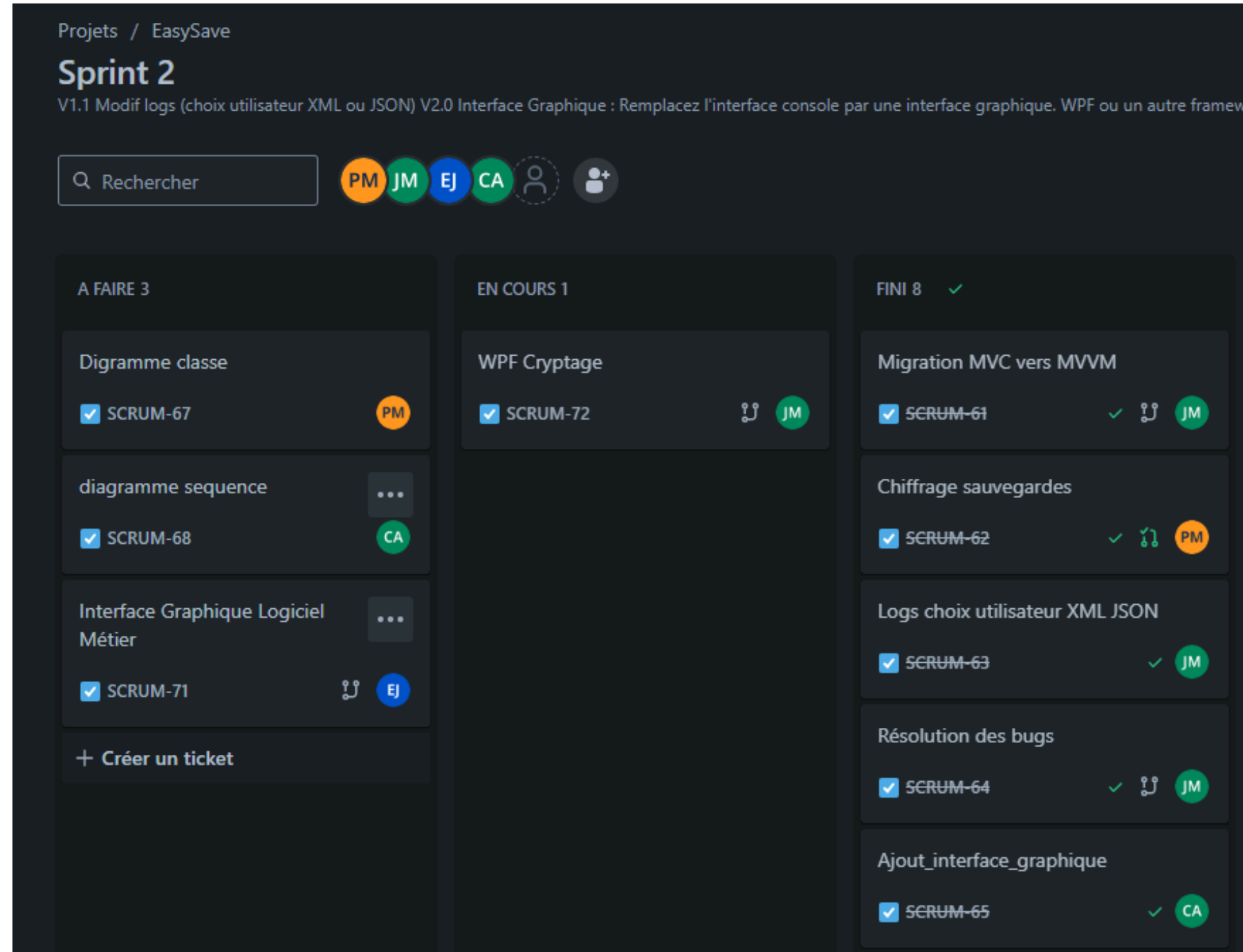
Objectifs du projet :

- **développer un logiciel de sauvegarde performant et maintenable**

Enjeux :

- **Intégration dans la suite ProSoft et respect des standards**
- **Facilité d'utilisation et expérience utilisateur optimisée**
- **Maintenance et évolutivité du logiciel**

Gestion de Projet



1

Suivi structuré des tâches

- Organisation des tâches sous forme de tickets
- Attribution des tâches aux membres de l'équipe
- Définition des priorités et des échéances

2

Gestion Agile efficace

- Supporte les méthodes Scrum et Kanban
- Planification et suivi des sprints
- Adaptabilité aux évolutions du projet

3

Collaboration et transparence

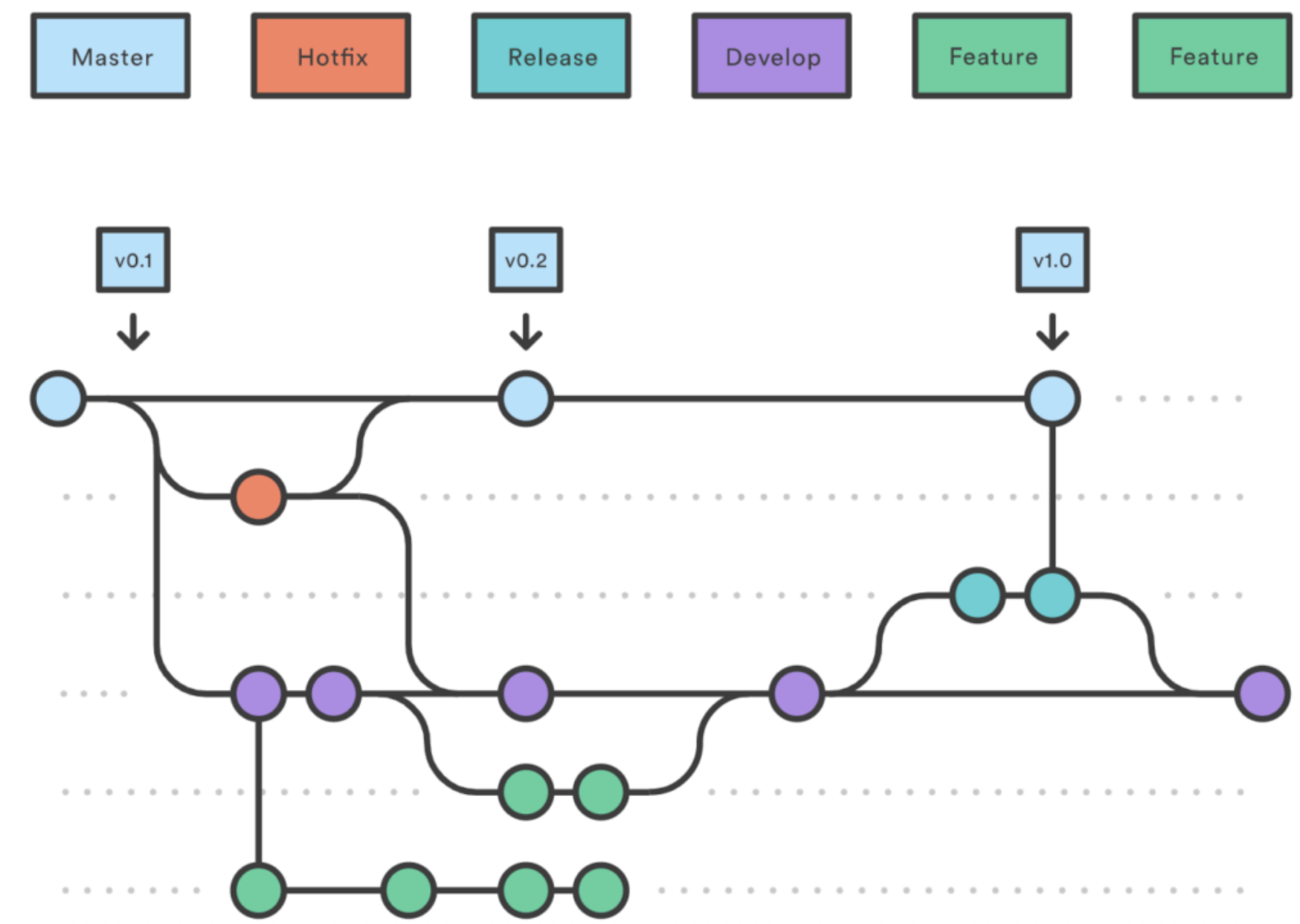
- Visibilité en temps réel de l'avancement des tâches
- Centralisation des informations et des discussions
- Intégration avec GitHub, Confluence et Slack

4

Suivi des bugs et évolutions

- Identification et traçabilité des anomalies
- Création de backlogs et roadmaps
- Génération de rapports et indicateurs de performance

Versioning



1

Organiser notre workflow de développement

- Création de branches dédiées pour chaque nouvelle fonctionnalité (feature branches)
- Utilisation de la branche dev pour centraliser le code à tester avant de le déployer en production

2

Suivre l'avancement du code au quotidien

- Intégration des modifications des développeurs via des merges dans les branches appropriées
- Gestion des versions grâce aux tags (par exemple v1.0) pour suivre les différentes étapes du développement

3

Faciliter la collaboration et la coordination

- Poussée des modifications dans des branches de feature spécifiques, permettant une meilleure séparation des tâches
- Utilisation des merges pour combiner les efforts de l'équipe et assurer la cohésion du code

4

Gérer les tests et la stabilité

- Tests effectués dans la branche dev avant la mise en production
- Fusion des modifications dans la branche release pour préparer le code en vue d'un déploiement stable



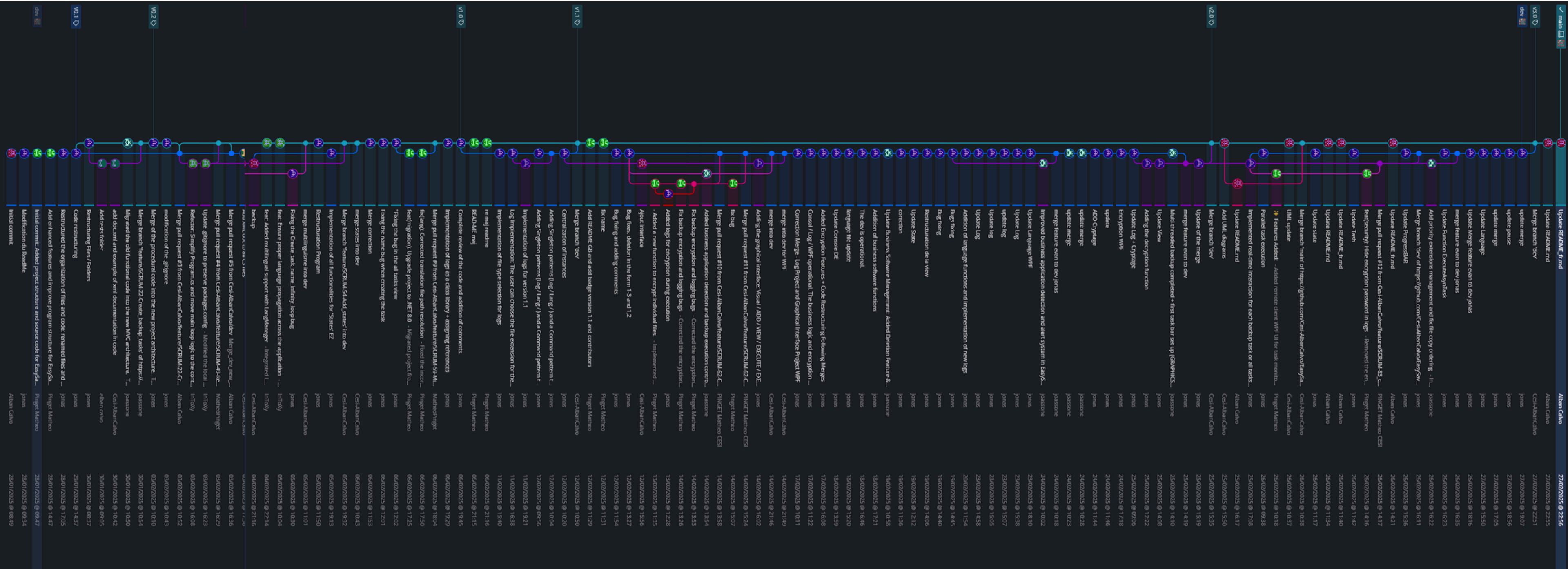
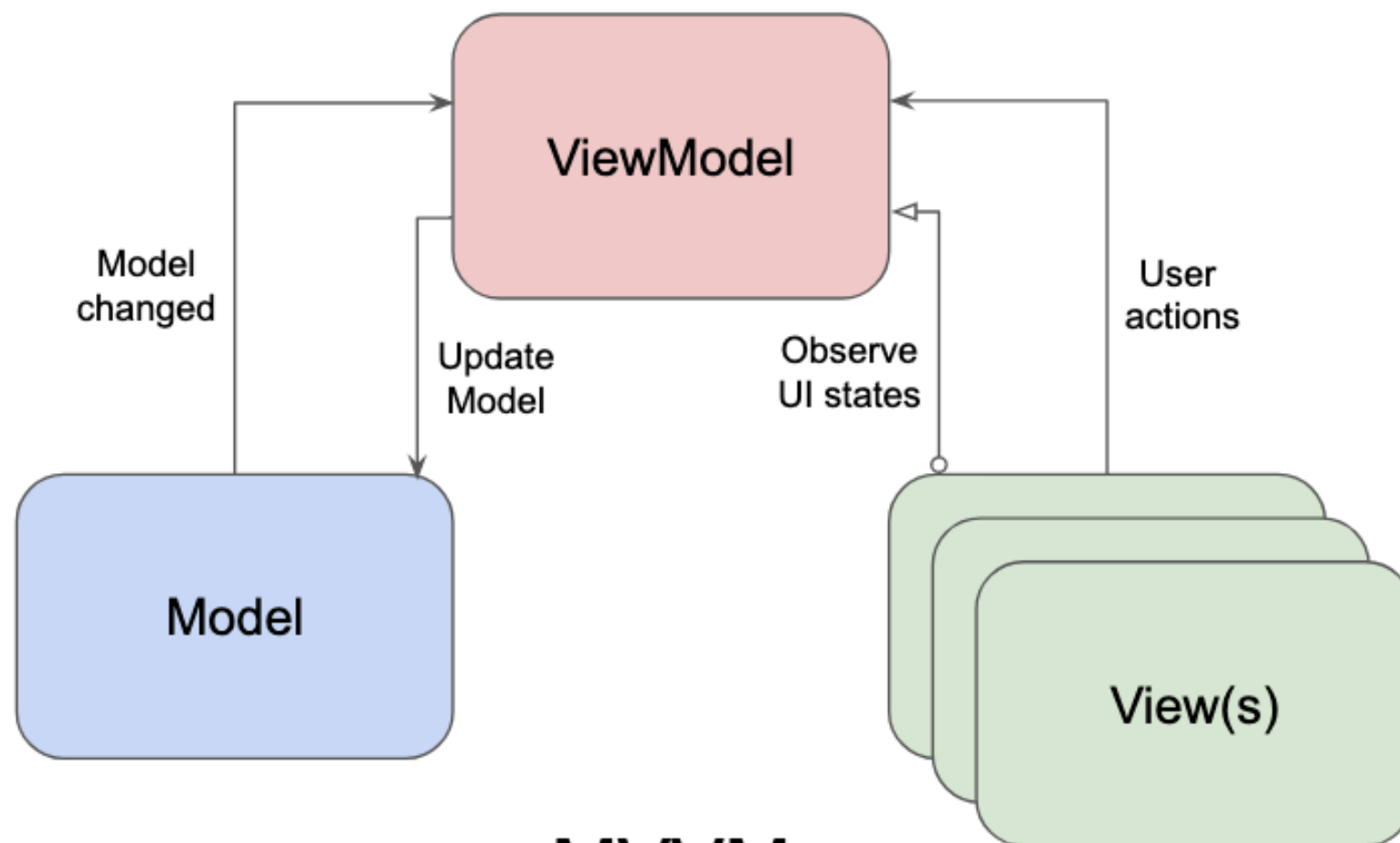


Figure 1 : Arborescence git

Architecture



MVVM

1

Séparation des responsabilités

- Divise le code en **Model (Données)**, **View (Interface)** et **ViewModel (Gestion des interactions)**.
- Facilite la **compréhension et la maintenance** du projet.

2

Facilite les tests et la maintenance

- La logique métier (Model) est **indépendante de l'interface graphique** (View), ce qui **simplifie les tests unitaires**.
- Le ViewModel permet de **modifier l'interface sans impacter la logique**.

3

Évolutivité et modularité

- Ajout de **nouvelles fonctionnalités** sans toucher au cœur du projet.
- Permet de **réutiliser du code facilement** entre plusieurs interfaces (View).

4

Gérer les tests et la stabilité

- Structure claire pour les **grands projets**.
- Séparation entre **les actions utilisateur et la gestion des données**.

Diagrammes UML

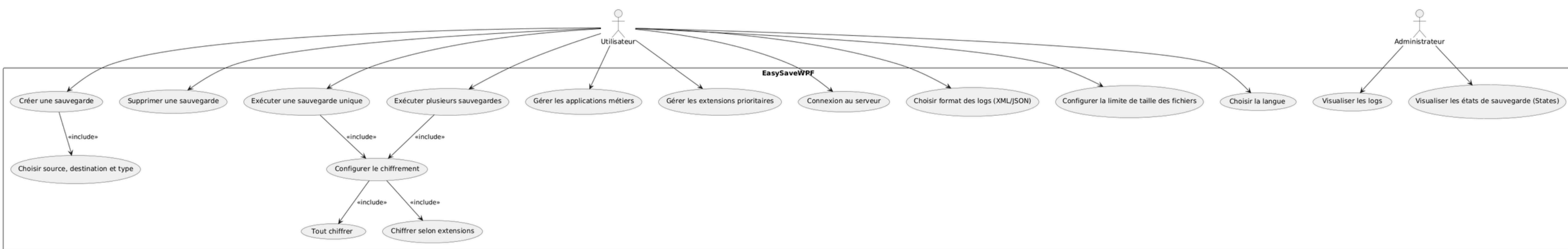


Figure 2 : Diagramme de cas d'utilisation

Diagrammes UML

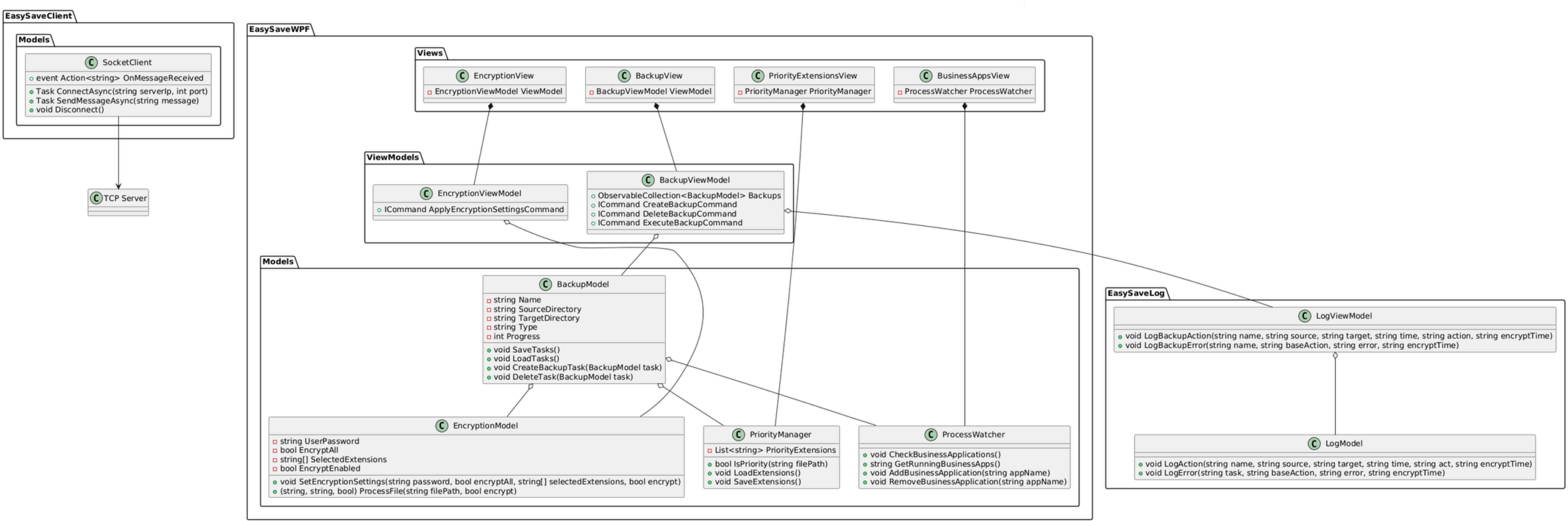


Figure 3 : Diagramme de classe

Version 1.x

Contraintes

- C#
- .NET 8
- Persistance des données (JSON)
- DLL pour les logs

Attentes

- Interface CLI
- Sauvegarder un dossier
- Choix langue FR/EN
- Log
- State
- Format Logs (JSON/XML)(1.1)
- Actions sur les sauvegardes :
 - Créer
 - Supprimer
 - Lister
 - Exécuter

```

/$$$$$$$$$
|$$______/
|$$_
|$$$$$$ /$$$$$$$ /$$$$$$$ /$$$ /$$$
|$$$$$$ |_____$$/ $$_ /$$_ /$$_ /$$$$$$$
|$$_ /$$$$$$$ |$$$$$$$ |$$$ |$$$ \_____$$/ /$$$$$$$ \$$$$$/ |$$$$$$$
|$$_ /$$_ $$_ \_____$$/ |$$$ |$$$ /$$_ \$$$ /$$_ $$/ |$$$ /
|$$$$$$$$ |$$$$$$$ /$$$$$$$ /$$$$$$$ /$$$$$$$ /$$$$$$$ \$$$ / |$$$$$$$
|_____/ \_____/ |_____/ \_____$$/ \_____/ \_____/ \_____/
| $$$ | $$$
| $$$$$$/
| \_____/

```

1. Créer une tâche de sauvegarde
2. Exécuter une tâche de sauvegarde
3. Exécuter toutes les tâches de sauvegarde
4. Voir toutes les tâches de sauvegarde
5. Supprimer une tâche de sauvegarde
6. Changer le Format des Log
7. Quitter

Selectionnez une option: |

Version 2.x

Name

Source

...

Destination

...

Type

▼

Add

View

Language

▼

Apply

Log file type

▼

Apply

business software

Execute all

Attentes

- Interface graphique (WPF)
- Nombre de travaux illimités
- Chiffrement SHA256
- Détection de logiciel métier
 - Si un logiciel métier est détecté, les sauvegardes ne sont pas lancées

Version 3.x

Attentes

- Sauvegardes en parallèle
- Sauvegardes de fichiers prioritaires (extensions)
- Limiter la taille de fichiers transmis en même temps
- Suivi et interaction :
 - Pause
 - Relance
 - Arrêt
- Pause temporaire si logiciel métier détecté
- Console déportée avec des sockets

EasySave

Name

Save_10

Source

C:\Users\Matheo\Downloads\SRC

...

Destination

C:\Users\Matheo\Downloads\DST

...

Type

Full

▼

Add

View

Language

▼

Apply

Log file type

▼

Apply

business software

Decrypt Folder

Execute all

Prioritary Extensions

Pause

Stop

Name	Source	Destination	Type	Progression	Actions
Save_10	C:\Users\Matheo\Doi	C:\Users\Matheo\Doi	Full		● ✗
Save_1	C:\Users\Matheo\Doi	C:\Users\Matheo\Doi	Full		● ✗

Remote Console

127.0.0.1

Connect

Retrieve Tasks

Task Name	Source	Destination
Save_7	C:\Users\Matheo\Downloz	C:\Users\Matheo\Downloz
Save_8	C:\Users\Matheo\Downloz	C:\Users\Matheo\Downloz
Save_9	C:\Users\Matheo\Downloz	C:\Users\Matheo\Downloz
Save_10	C:\Users\Matheo\Downloz	C:\Users\Matheo\Downloz
Save_1	C:\Users\Matheo\Downloz	C:\Users\Matheo\Downloz
Save_2	C:\Users\Matheo\Downloz	C:\Users\Matheo\Downloz

Execute Selected Task

[CLIENT] Execution request sent for Save_1 with encryption: NO_ENCRYPT

Documentation

Documentation disponible sur : <https://groupe1-easysave.alwaysdata.net/index.html>

Documentation utilisateur : <https://github.com/Cesi-AlbanCalvo/EasySave#user-manual---version-20>

Points d'amélioration

- Gestion avancée de la bande passante
- Sauvegarde cloud et stockage hybride
- Interface utilisateur améliorée
- Sécurité renforcée
- Gestion des erreurs et résilience

CONCLUSION