



UNIVERSITÉ SAINT JEAN  
SAINT JEAN INGÉNIEUR

# Cahier de Conception : DISK CLONER

Présenté par :

- HARVIS LANDRY FOTSEU FONKWA
- TCHINWA ISMAEL CLINTON

*Étudiants à Saint Jean Ingénieur*



Sous la supervision de :

- M. Emmanuel NGUIMBUS,
- Enseignant de SE à Saint Jean Ingénieur*

# PLAN



**01**

**PRESENTATION GENERALE**

**02**

**OBJECTIFS VISES**

**03**

**COMPOSANTES DU SYSTÈME**

**04**

**CONTRAINTES GENERALES DE  
CONCEPTION**

**05**

**DOCUMENTS DE RÉFÉRENCE**

**06**

**CONCEPTION  
GENERALE**

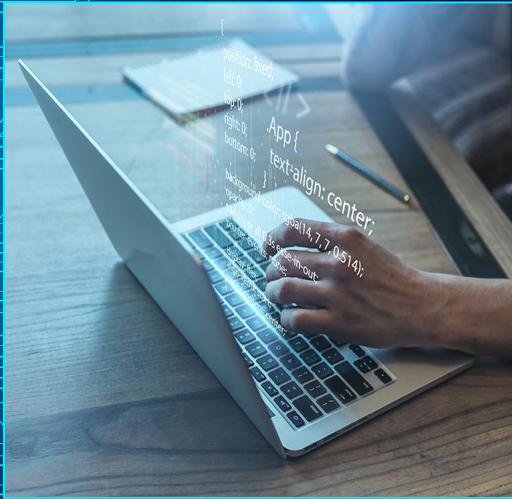
# 01

## PRESENTATION GENERALE

Disk Cloner



# INTRODUCTION



Le projet **Disk Cloner** répond aux défis de migration et de sécurisation des données sous **OpenSolaris**, un environnement exigeant une manipulation précise des structures de stockage bas niveau. Face aux risques d'erreurs liés aux outils en ligne de commande, ce projet propose une solution logicielle permettant de simplifier le clonage de disques et de partitions. En alliant la puissance des mécanismes **ZFS** à une **interface graphique** intuitive, le système garantit une duplication fiable, un suivi en temps réel de la progression et une sécurité accrue pour l'intégrité des infrastructures de stockage.



# PRESENTATION GENERALE DU PROJET

L'objectif principal est de **développer manuellement le moteur de copie** du logiciel (en recréant le fonctionnement de l'outil classique dd). Au lieu d'utiliser des programmes déjà existants, le projet vise à programmer directement la lecture et l'écriture des données sur le disque. Cette approche permet de contrôler précisément le transfert, de gérer les erreurs et de calculer en temps réel l'avancement de l'opération. Pour rendre l'outil accessible, une **interface graphique** sera créée afin d'éviter l'utilisation de lignes de commande complexes et de sécuriser les manipulations.

Ainsi, le projet se décompose en deux volets :

- **Programmation d'un moteur de copie sur mesure**, qui gère lui-même le déplacement des données bloc par bloc et assure que le disque ne soit pas accidentellement modifié s'il est déjà utilisé par l'ordinateur.

- **Développement d'une interface graphique visuelle**, permettant de lister les disques branchés, de lancer le clonage en un clic et de suivre la progression grâce à une barre d'état et des messages de confirmation clairs.





# 02

## OBJECTIFS VISES





# OBJECTIFS VISES

- **Maîtriser la gestion du stockage bas niveau** dans un environnement OpenSolaris grâce à la mise en œuvre d'un moteur de transfert propriétaire, capable de manipuler directement les flux de données binaires.
- **Développer un algorithme de clonage robuste**, permettant de remplacer l'outil standard dd par une solution programmée manuellement, offrant un contrôle total sur la lecture et l'écriture des blocs de données.
- **Garantir l'intégrité et la sécurité des données** via la détection automatique des partitions montées et l'intégration de verrous de sécurité pour empêcher toute opération destructive accidentelle sur le système hôte.
- **Simplifier l'administration des disques** par le développement d'une interface graphique intuitive, qui permet de piloter le moteur de copie et de visualiser la progression du transfert sans recourir à la ligne de commande.
- **Démontrer la faisabilité d'une solution de sauvegarde "low-level"**, capable de gérer les spécificités d'OpenSolaris (Slices et ZFS) dans un cadre académique ou professionnel de maintenance système.





# **COMPOSANTES DU SYSTÈME**

**03**



Acteurs internes	Roles
Moteur de copie (Moteur personnalisé)	<ul style="list-style-type: none"><li>• - Ouvre les périphériques de stockage en mode binaire (Raw Devices).</li><li>• - Réimplémente la logique de dd pour lire et écrire les données bloc par bloc.</li></ul>
Gestionnaire de Disques (Kernel Solaris)	<ul style="list-style-type: none"><li>• Interagit avec le noyau pour lister les disques physiques (/dev/rdisk/).</li><li>• - Analyse la structure des tranches (<b>Slices</b>) et identifie les partitions système.</li><li>• - Fournit les informations de taille et de formatage au système</li></ul>
Interface Graphique (UI)	<ul style="list-style-type: none"><li>- Fournit un accès simplifié à la sélection des sources et des destinations.</li><li>- Reçoit les métriques du moteur pour afficher la <b>progression</b> en temps réel.</li></ul>
Moniteur de Sécurité	<ul style="list-style-type: none"><li>- Vérifie l'état de montage des disques pour empêcher le clonage d'un disque utilisé.</li></ul>

# 04

## CONTRAINTES GÉNÉRALES D'UTILISATION



# CONTRAINTES DU SYSTEME



Catégories	Contraintes
Contraintes techniques	<ul style="list-style-type: none"><li>• Accès Bas Niveau (Raw I/O) : Obligation de passer par les fichiers spéciaux /dev/rdisk pour une copie bit à bit conforme à dd.</li><li>• Gestion des Buffers : Nécessité d'optimiser la taille de la mémoire tampon (buffer) pour équilibrer vitesse de transfert et consommation RAM.</li><li>• Spécificités OpenSolaris : Prise en charge des slices (partitions) Solaris et compatibilité avec les snapshots ZFS.</li><li>• Verrouillage des cibles : Empêcher l'écriture sur un système de fichiers monté pour éviter la corruption de données.</li></ul>
Contraintes sécuritaires	<ul style="list-style-type: none"><li>• Privilèges RBAC/Root : L'application doit impérativement s'exécuter avec des droits d'administrateur pour accéder aux drivers de disques.</li><li>• Irréversibilité des données : Obligation d'implémenter des barrières de confirmation critiques (ex: saisie de "CLONER") pour éviter l'effacement accidentel du disque système.</li><li>• Isolation : S'assurer que le processus de clonage n'interfère pas avec les autres services critiques du noyau Solaris.</li></ul>
Contraintes académiques et temporelles	<ul style="list-style-type: none"><li>• Maîtrise du fonctionnement de dd : Capacité à expliquer techniquement la différence entre une copie de fichiers et une copie de blocs.</li><li>• Qualité du code : Utilisation de langages adaptés au système (C/C++ ou scripts robustes) avec une gestion d'erreurs rigoureuse.</li><li>• Livrable documenté : Fournir les schémas d'architecture (UML) et les logs d'exécution comme preuve de succès.</li></ul>



# 05

## DOCUMENTS DE RÉFÉRENCE



# DOCUMENTS DE RÉFÉRENCE

Domaines	Références
Système OpenSolaris	Oracle Solaris Administration Guide (Devices and File Systems) OpenSolaris Bible (Nicholas Solter et al.)
Gestion du Stockage	ZFS Administration Guide (Oracle/Sun Microsystems) ZFS Send/Receive Stream Specification Standard POSIX (IEEE Std 1003.1) pour les entrées/sorties (read, write, open)
Architecture Système	Solaris Internals (Richard McDougall & Jim Mauro) RBAC Guide (Role-Based Access Control) pour Solaris PAM (Pluggable Authentication Modules) Documentation
Sécurité & Intégrité	NIST Special Publication 800-88 (Guidelines for Media Sanitization - pour comprendre l'écrasement de données)
Standards de Disques	SFF Standards (Small Form Factor) GPT (GUID Partition Table) vs VTOC (Volume Table of Contents - spécifique Solaris)

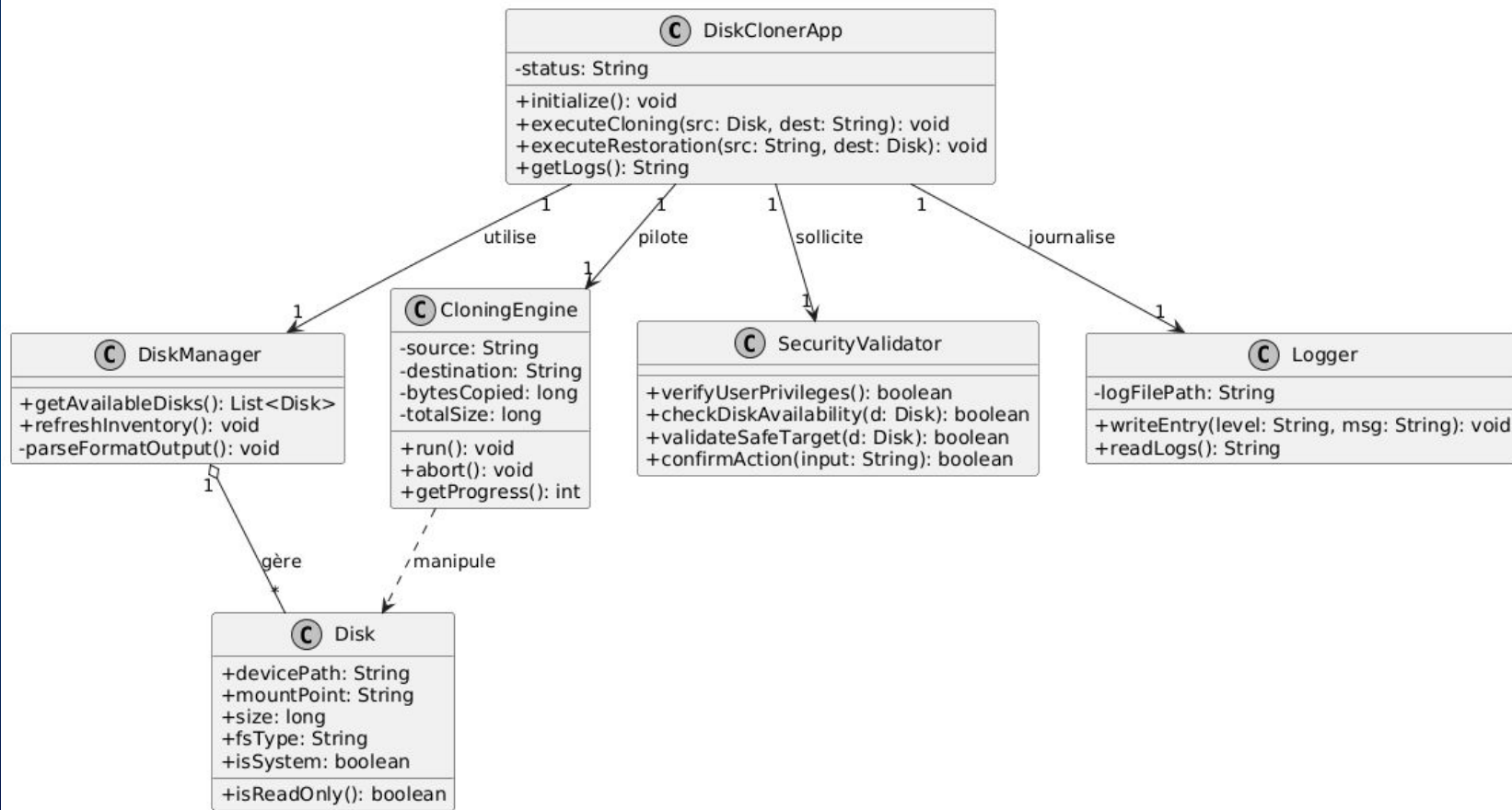
# 06

## CONCEPTION GENERALE



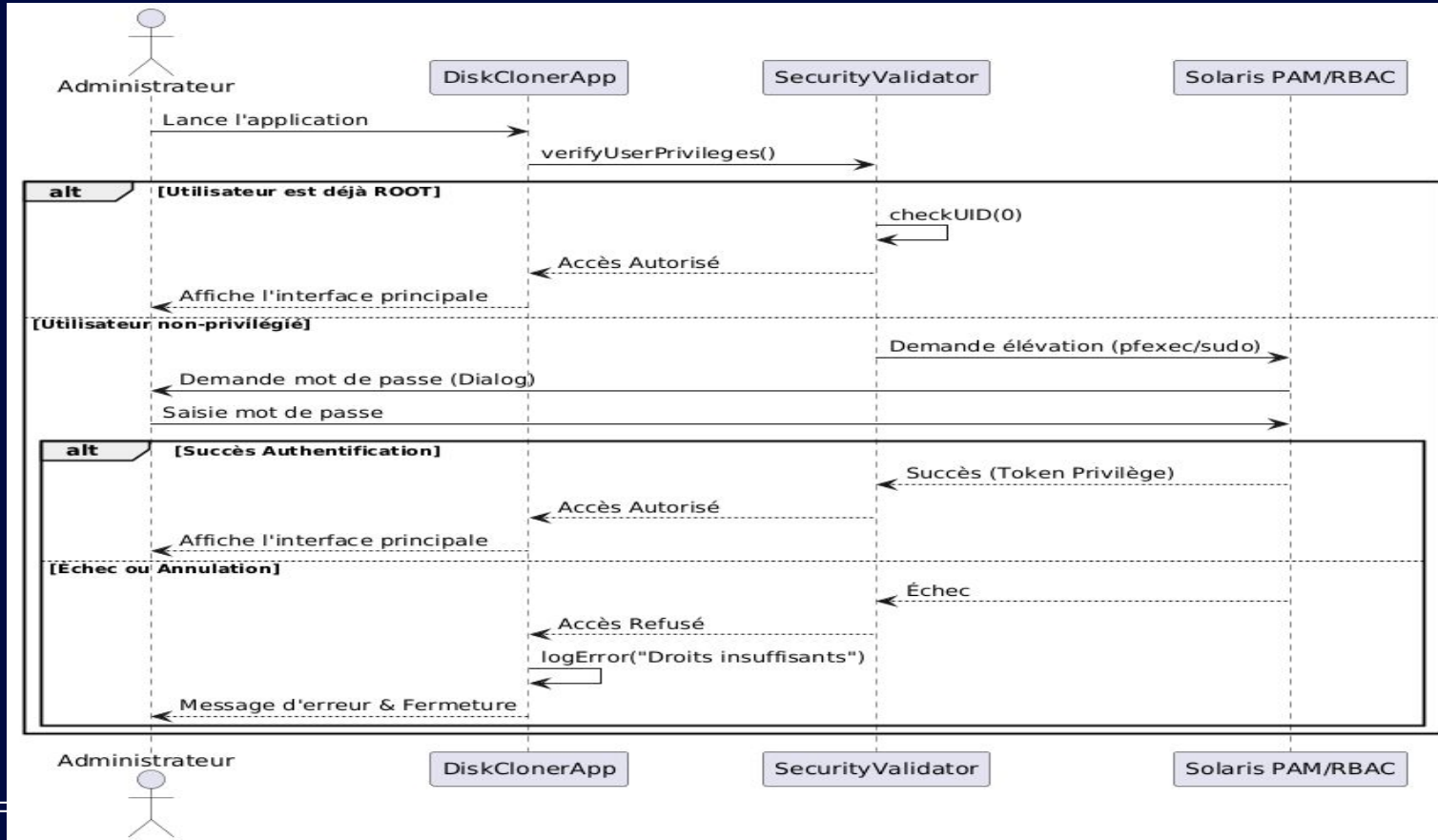


# DIAGRAMME DE CLASSE



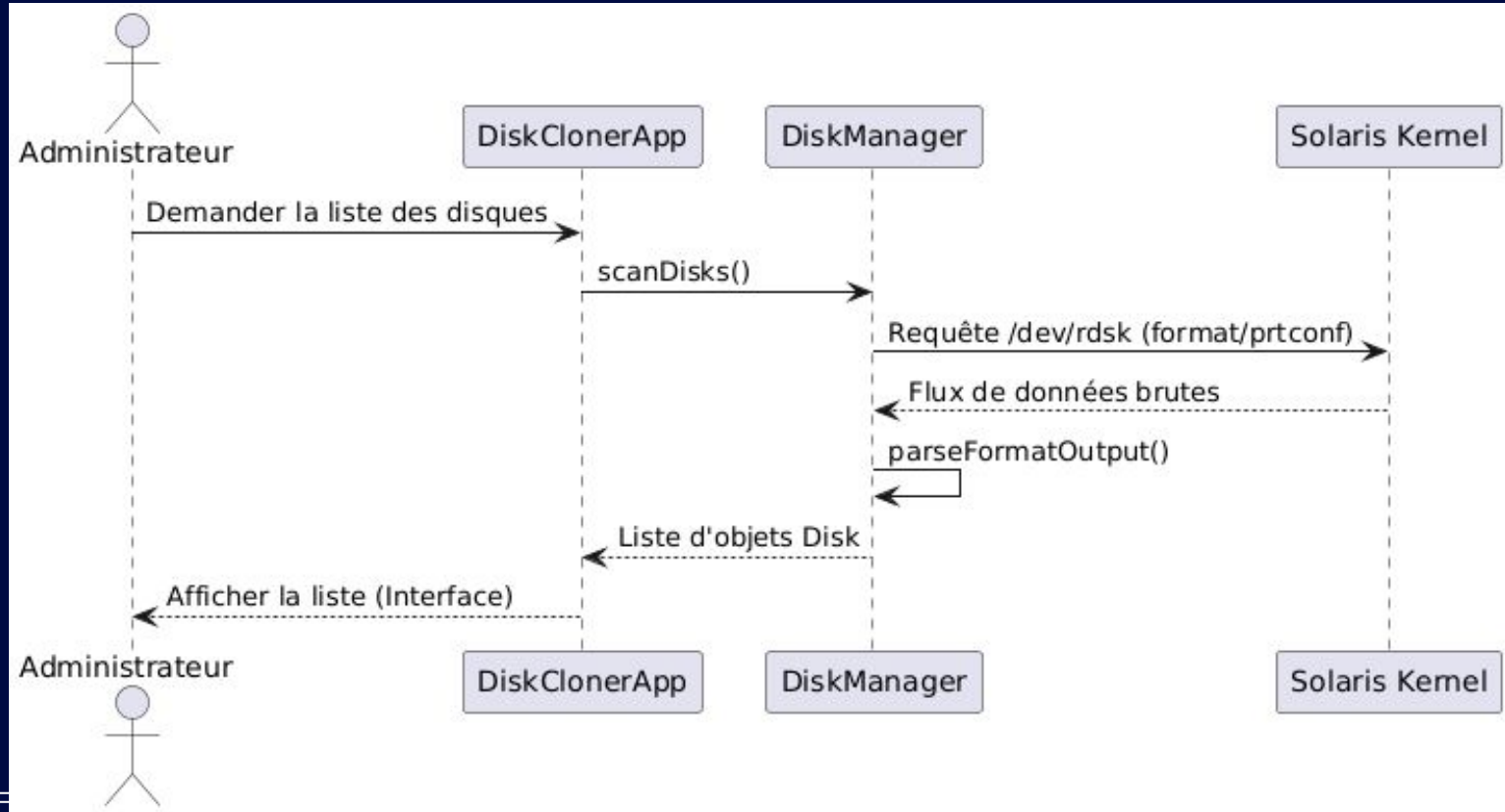
# PRINCIPAUX DIAGRAMMES DE SEQUENCE

## SEQUENCE DE CONNEXION



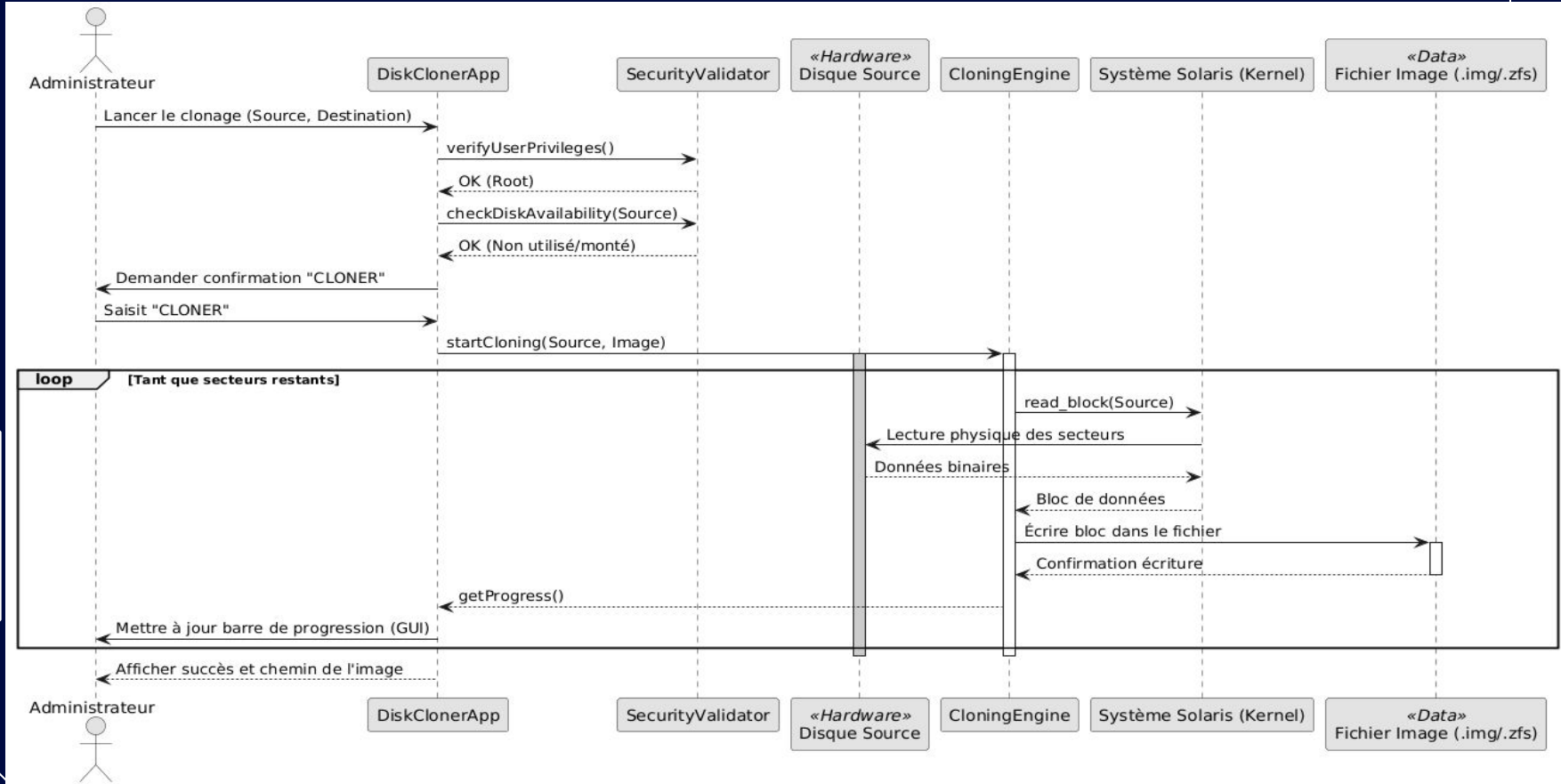
# PRINCIPAUX DIAGRAMMES DE SEQUENCE

## SEQUENCE Lister les disques et partitions



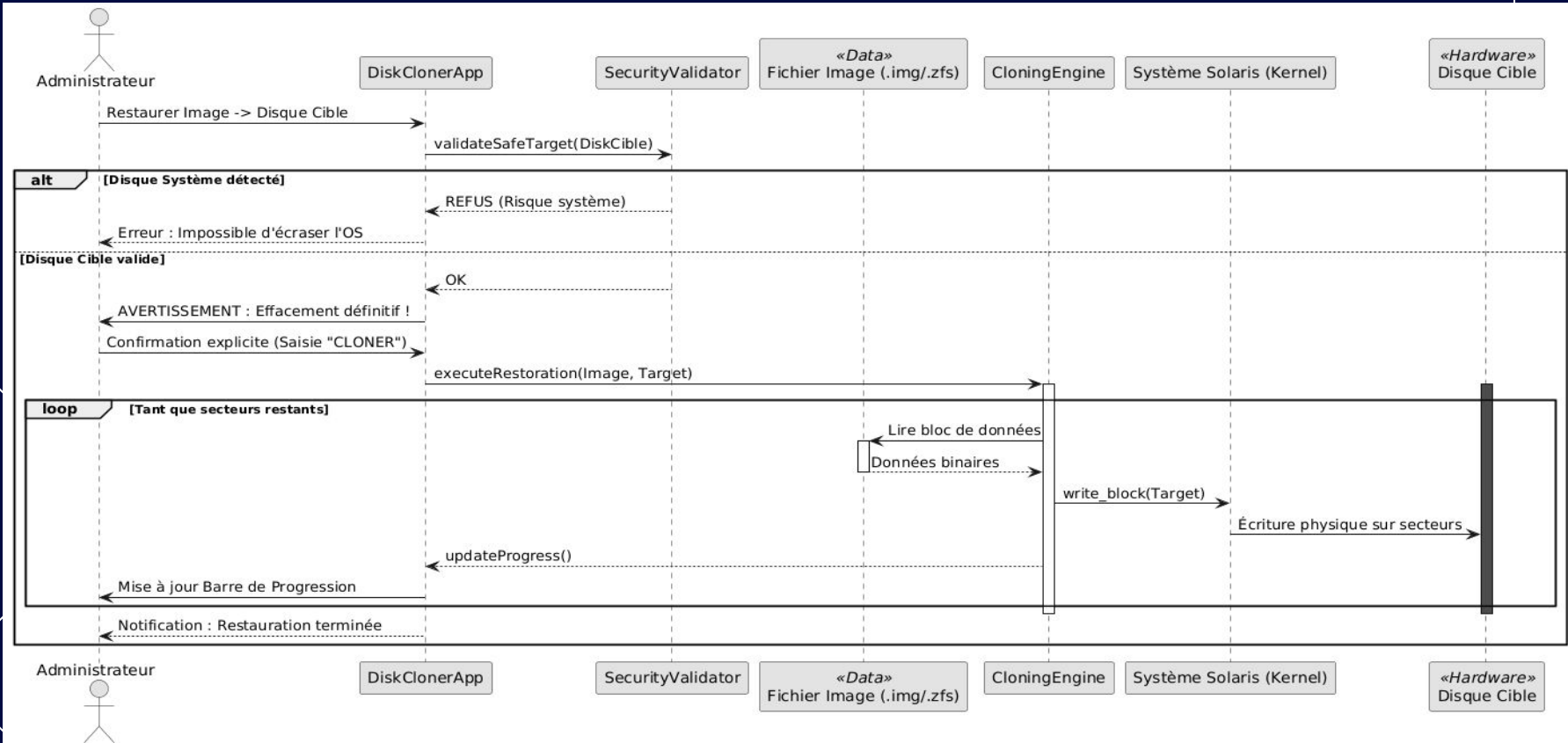
# PRINCIPAUX DIAGRAMMES DE SEQUENCE

## SEQUENCE Cloner un disque vers une image



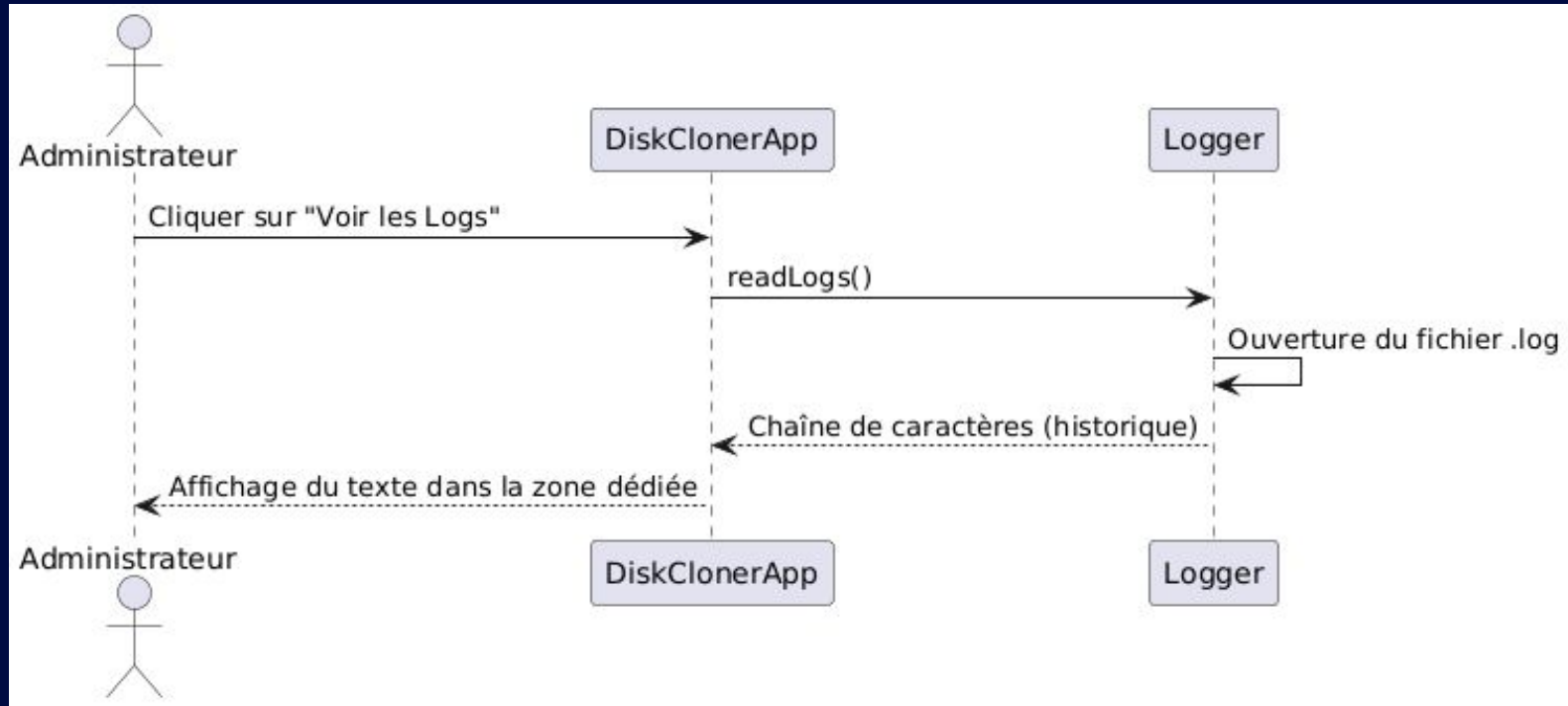
# PRINCIPAUX DIAGRAMMES DE SEQUENCE

## SEQUENCE Restaurer une image vers un disque



# PRINCIPAUX DIAGRAMMES DE SEQUENCE

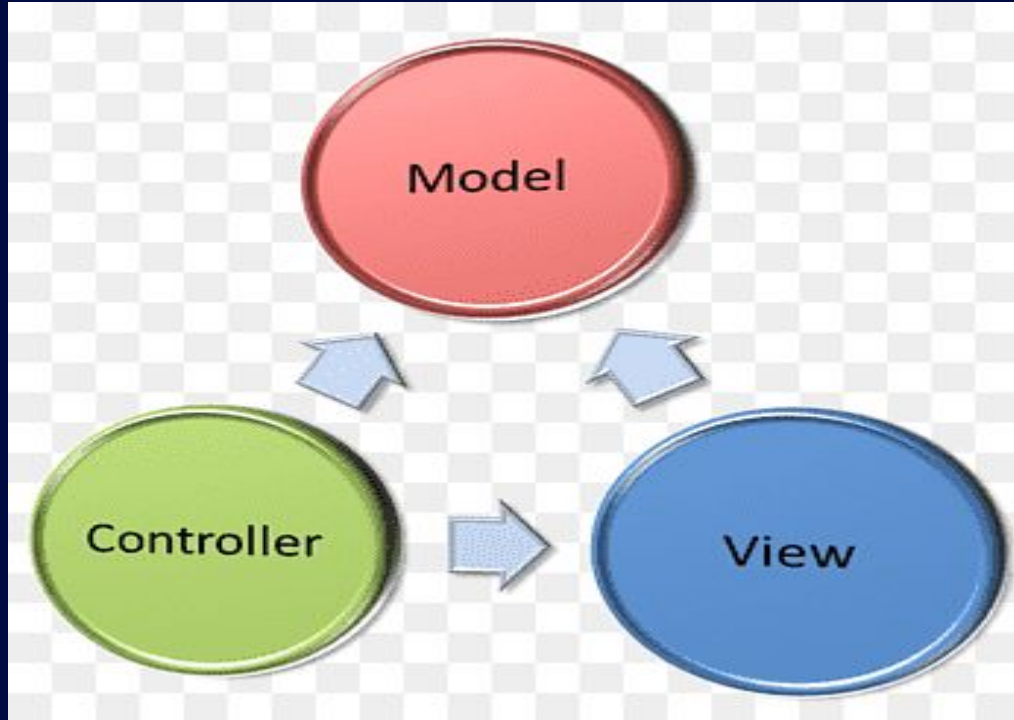
## SEQUENCE Consulter les journaux (Logs)



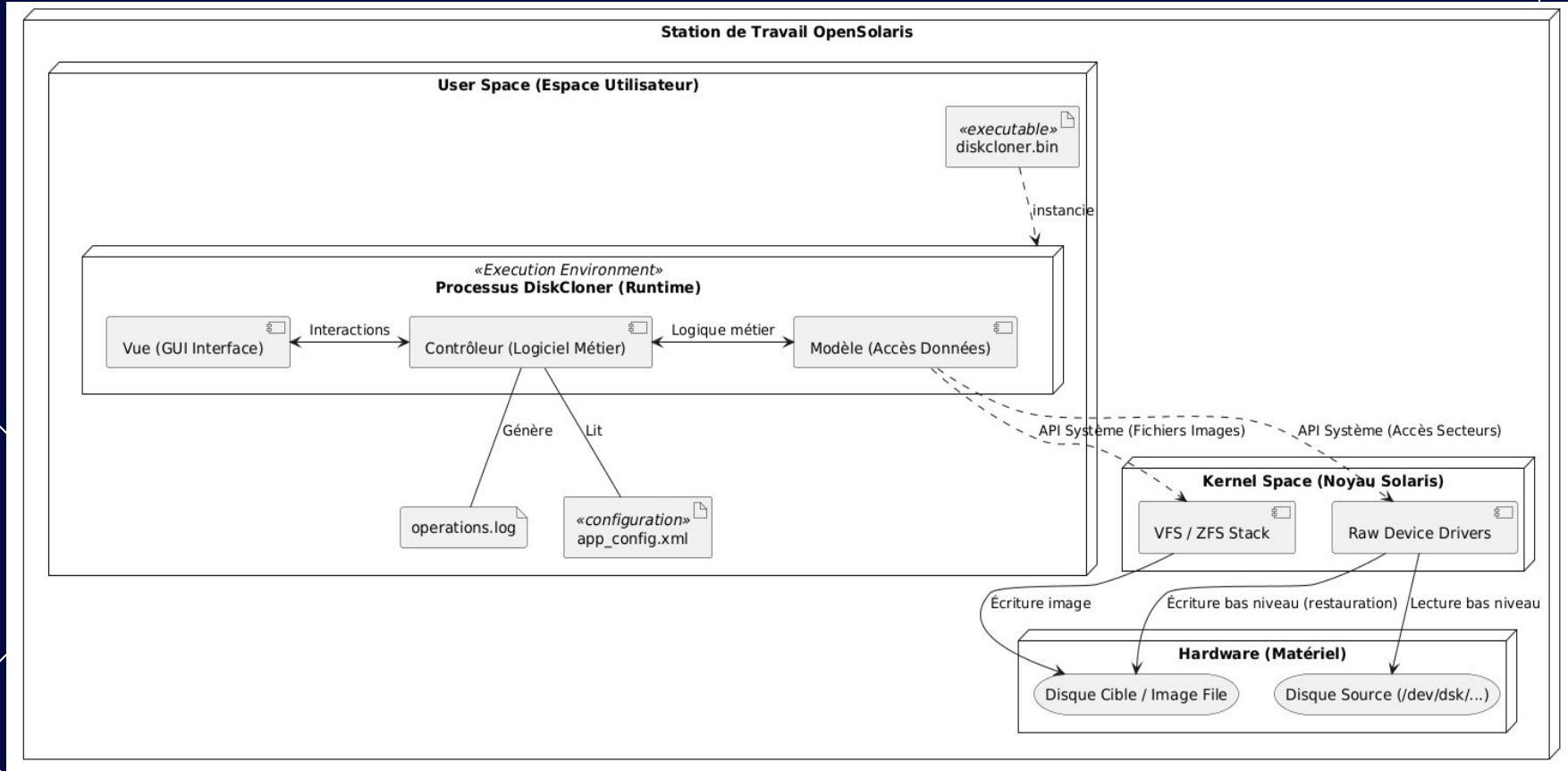


# PATRON DE CONCEPTION

## MVC (Modèle-Vue-Contrôleur)



# DIAGRAMME DE DEPLOIEMENT



# Thanks!

Any questions?

