Orchestration de ressources réseaux basée sur la technologie blockchains

Encadrante : *Maria POTOP-BUTUCARU*

Étudiants : *Mbainaissem Narcisse MEDION, Imene ASSOUS, Rita BENZAKOUR, Imane ELAZZOUZI*

Table des matières

Liste des abréviations

TEE	*Trusted Execution Environment*
DKG	Distributed Key Generation
IAS	Intel Attestation Service
SDN	*Software-Defined Networking*
NFV	Network Function Virtualization
TLS	*Transport Layer Security*
QoS/QoE	Quality of Service / Quality of Experience
HSM	*Hardware Security Module*
SGX	Software Guard Extensions *(Intel)*
SEV	*Secure Encrypted Virtualization* (AMD)
MPC	Multi-Party Computation
BLS	*Boneh–Lynn–Shacham* (signatures)
MIR	Mathématiques Informatique et Recherche *(bibliothèque)*
EVM	Ethereum Virtual Machine
dApp	Application décentralisée

Liste des figures

1. Introduction

1.1. Contexte du projet

Le développement rapide de la virtualisation et des technologies associées, telles que le Software-Defined Networking (SDN) et la Network Function Virtualization (NFV), a un impact considérable sur l'évolution des réseaux, notamment dans le contexte des réseaux de cinquième génération (5G). Ces paradigmes permettent une flexibilité accrue dans la gestion des ressources réseau, en permettant de concevoir et d'adapter des protocoles et des fonctions réseau à divers cas d'étude. L'émergence de réseaux de plus en plus programmables permet aux opérateurs de répondre de manière dynamique et automatisée aux besoins du système, en fonction des variations des états du réseau. Cela améliore la performance et l'efficience des réseaux modernes.

Ce projet se concentre sur l'étude des solutions d'orchestration de ressources dans des réseaux multi-domaines. Nous mettrons un accent particulier sur l'utilisation de la blockchain pour surmonter les limitations des solutions actuelles et fournir une solution plus sécurisée et plus privée.

1.2. Problématique générale

L'orchestration des ressources dans des réseaux multi-domaines est un défi majeur, particulièrement lorsqu'il s'agit de garantir la confidentialité, la sécurité, et la performance du réseau tout en assurant la scalabilité du système. Les architectures traditionnelles, bien qu'efficaces dans des environnements simples, présentent plusieurs limites dans des contextes plus complexes. Elles nécessitent souvent une confiance totale dans les entités centrales de gestion, ce qui constitue une faiblesse majeure en termes de sécurité et de transparence. Dans ce cadre, la question centrale est la suivante : ***comment assurer une orchestration efficace, sécurisée et confidentielle des ressources réseau entre domaines multiples, sans exposer d'informations sensibles ni dépendre d'une autorité centrale ?***

2. Cahier des charges

2.1. Objectifs

Ce projet a pour objectif de radiographier l'état de l'art des solutions décentralisées d'orchestration des ressources réseau utilisant les technologies modernes telles que l'architecture NetChain basée sur la blockchain et l'environnement d'exécution de confiance (TEE). On vise à identifier des pistes pour surmonter les limites des approches traditionnelles, notamment le manque de confidentialité, l'absence d'équité et les risques liés aux autorités centralisées et ainsi proposer une architecture adaptée aux besoins des opérateurs réseau pour le partage d'informations sensibles et la garantie de QoS/QoE dans des environnements distribués et dynamiques.

2.2. Contraintes Techniques

Contraintes liées à l'architecture — L'architecture proposée doit rester strictement modulaire, de façon à pouvoir accueillir sans refonte majeure de nouvelles fonctions ou de nouveaux services réseau, et elle doit s'appuyer sur une infrastructure distribuée pour le traitement des transactions blockchain comme pour la gestion des ressources, afin d'assurer la résilience globale du système et la sécurisation des données.

Contraintes liées à la sécurité — Le système doit protéger la confidentialité des échanges entre opérateurs tout en maintenant une gestion rigoureuse des ressources ; il doit donc offrir une résistance explicite contre les attaques Sybil, les dénis de service (DoS) et les interceptions de type man-in-the-middle, tout en intégrant des mécanismes d'authentification forte et de contrôle d'accès strict.

Contraintes liées à l'interopérabilité — La solution doit rester pleinement compatible avec les technologies existantes, notamment SDN et NFV, afin de garantir une interopérabilité fluide avec les infrastructures hétérogènes des différents opérateurs de réseau.

2.3. Délimitations du périmètre du projet

Ce projet s'inscrit dans le cadre de l'unité d'enseignement « Projets annuels du parcours RES » et est encadré par Madame Maria POTOP-BUTUCARU, professeure et chercheuse. Il se concentre uniquement sur l'orchestration des ressources réseau dans un environnement multi-domaines en utilisant la blockchain pour résoudre les problèmes de sécurité et de confidentialité

2.4. Résultats attendus

Les résultats attendus comprennent, premièrement, **une architecture décentralisée** pleinement fonctionnelle, capable d'orchestrer en toute sécurité les ressources réseau entre plusieurs opérateurs tout en préservant la confidentialité des données ; deuxièmement, **un protocole de consensus CoNet** opérationnel intégrant des mécanismes de validation qui n'exposent aucune information sensible ; et, troisièmement, **une solution réellement scalable**, apte à absorber un nombre croissant de transactions réseau tout en maintenant des performances optimisées en termes de latence et de bande passante.

2.5. Livrables

Les livrables comprennent :

- Un rapport détaillé de la solution d'orchestration de ressources, avec une description complète de l'architecture, du fonctionnement du protocole CoNet et des technologies utilisées (blockchain, TEE),

- Le code source des smart contracts utilisés pour l'orchestration et la gestion des transactions réseau,

- Un ensemble de tests et de résultats de performance, incluant des simulations et des analyses sur la scalabilité et la sécurité de la solution.

3. État de l'art

3.1. Concepts clés

La 5G

La 5G représente la cinquième génération des réseaux mobiles, offrant des débits de données plus rapides, une latence réduite et la possibilité de connecter un nombre bien plus élevé d'appareils en simultané par rapport à la 4G. En utilisant de nouvelles fréquences radio, notamment les ondes millimétriques, elle permet des vitesses de téléchargement et de téléversement nettement supérieures (jusqu'à 10 Gbps) et une latence qui peut descendre sous les 1 milliseconde.

Avantages et défis de la 5G dans l'orchestration des réseaux :

- Avantages : Permet de créer des réseaux programmables et dynamiques, pouvant être adaptés en fonction des besoins spécifiques des utilisateurs (par exemple, pour des services ultra-rapides ou des applications Internet of Things (IoT)).

- Défis : La gestion de ces réseaux hautement dynamiques et multi-domaines devient complexe, avec un besoin d'orchestration avancée pour optimiser les ressources tout en garantissant la sécurité et la confidentialité.

Applications spécifiques de la 5G dans les réseaux multi-domaines :

La 5G permet de segmenter le réseau en tranches (slices), chacune étant adaptée à des besoins spécifiques. Par exemple, une tranche peut être dédiée aux véhicules autonomes, tandis qu'une autre pourrait être utilisée pour des applications médicales sensibles, nécessitant une faible latence et une haute sécurité. Dans un environnement multi-domaine, ces tranches doivent être efficacement orchestrées pour garantir leur performance et leur intégrité.

Network Slicing

Principe du Network Slicing:

Le Network Slicing est un concept clé dans les réseaux de cinquième génération (5G), qui permet de découper un réseau physique en plusieurs tranches virtuelles (slices), chacune étant une sous-réseau logique optimisé pour des besoins spécifiques. Chaque tranche peut être configurée pour des objectifs de performance, de latence et de capacité particuliers, selon les exigences des

utilisateurs ou des applications.

Chaque slice est indépendante et peut être personnalisée pour un usage particulier, qu'il s'agisse de services à faible latence (comme les véhicules autonomes) ou de services à haute bande passante (comme la vidéo en haute définition). Le network slicing permet également aux opérateurs de gérer les ressources du réseau de manière plus dynamique, de réduire les coûts opérationnels, et d'optimiser l'utilisation des infrastructures réseau existantes.

Rôle du Network Slicing dans l'orchestration des réseaux multi-domaines :

Dans un environnement multi-domaine, chaque slice peut traverser plusieurs domaines de gestion, chacun pouvant appartenir à des opérateurs différents. L'orchestration des slices réseau entre ces différents domaines nécessite une gestion coordonnée des ressources et des politiques de QoS (Qualité de Service). Le Network Slicing simplifie l'intégration de services multiples tout en garantissant que chaque service possède les ressources réseau nécessaires pour fonctionner de manière optimale, en isolant les différents types de trafic.

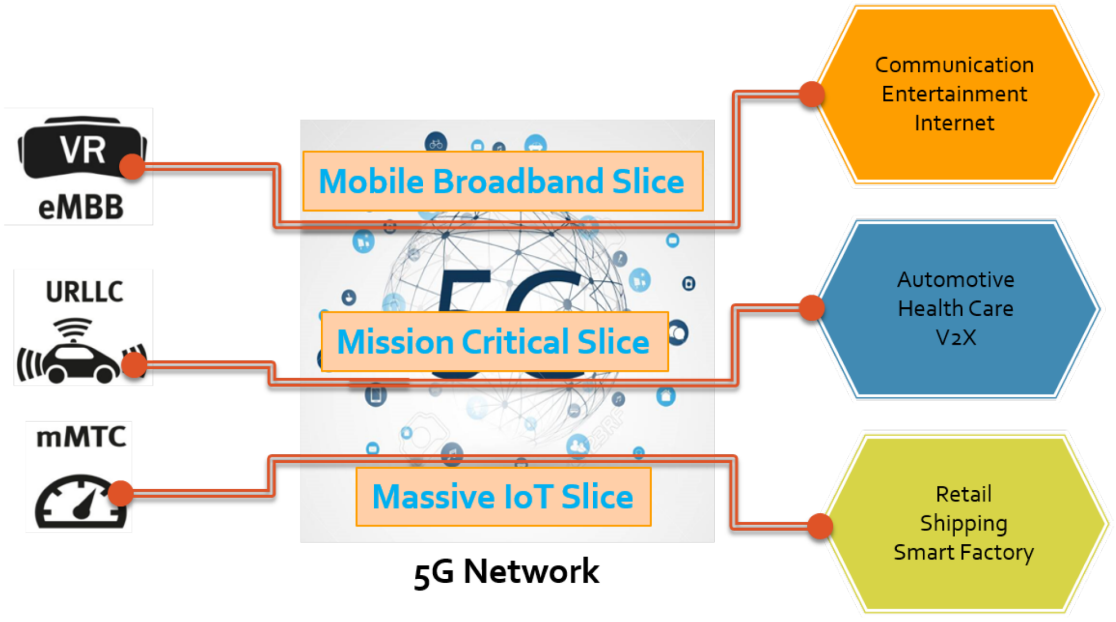

Figure 1: Architecture des tranches réseau dans un réseau 5G

Réseaux Multi-Domaines

Un réseau multi-domaine désigne un réseau composé de plusieurs domaines distincts, où chaque domaine peut appartenir à un opérateur ou une infrastructure différente. Dans un réseau 5G, par exemple, différents opérateurs ou fournisseurs de services peuvent être impliqués dans l'orchestration des tranches réseau (slices). Ces tranches peuvent être adaptées à des besoins spécifiques (comme la basse latence pour les véhicules autonomes ou la haute capacité pour la vidéo en haute définition).

Chaque domaine (par exemple, un opérateur spécifique ou un fournisseur de services cloud) gère ses propres ressources, et l'orchestre indépendamment des autres domaines, tout en collaborant

dans un environnement global pour offrir des services aux utilisateurs finaux.

SDN (Software-Defined Networking)

Principe du SDN et son fonctionnement :

Le SDN est une approche qui permet de séparer la gestion du réseau de son infrastructure physique. L'objectif est de rendre les réseaux plus flexibles, programmables et plus faciles à configurer à l'aide de contrôleurs centraux. Ce contrôle centralisé permet de gérer le flux de données entre les équipements du réseau (commutateurs, routeurs) via un logiciel.

Rôle du SDN dans l'orchestration des réseaux multi-domaines :

Le SDN assure l'interopérabilité entre différents opérateurs de réseaux et technologies.

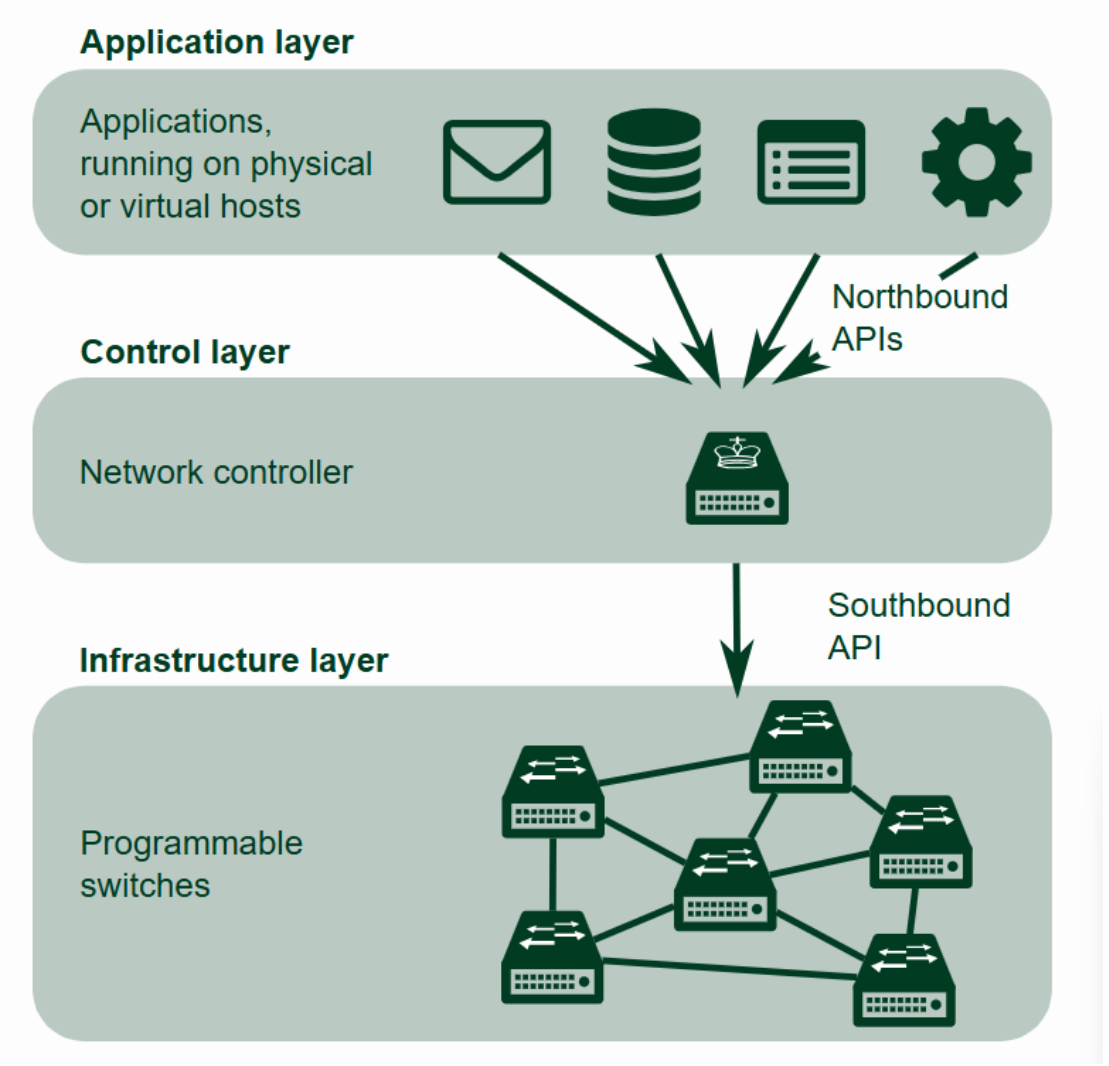

Figure 2: Architecture d'un réseau SDN (Software-Defined Networking)

NFV (Network Functions Virtualization)

Principe du NFV et son fonctionnement :

Le NFV permet de virtualiser les fonctions réseau, telles que les pares-feux, les routeurs, et les équilibreurs de charge, afin de les exécuter sur des serveurs standards et non sur des équipements physiques dédiés. Cela permet de réduire les coûts et d'améliorer la flexibilité et l'agilité du réseau.

Rôle du NFV dans l'orchestration des réseaux multi-domaines :

Le NFV permet d'intégrer des fonctions réseau virtuelles dans une architecture multi-domaines, facilitant l'adaptation du réseau en fonction des demandes spécifiques de chaque domaine. Il contribue à l'automatisation et à la flexibilité de l'orchestration.

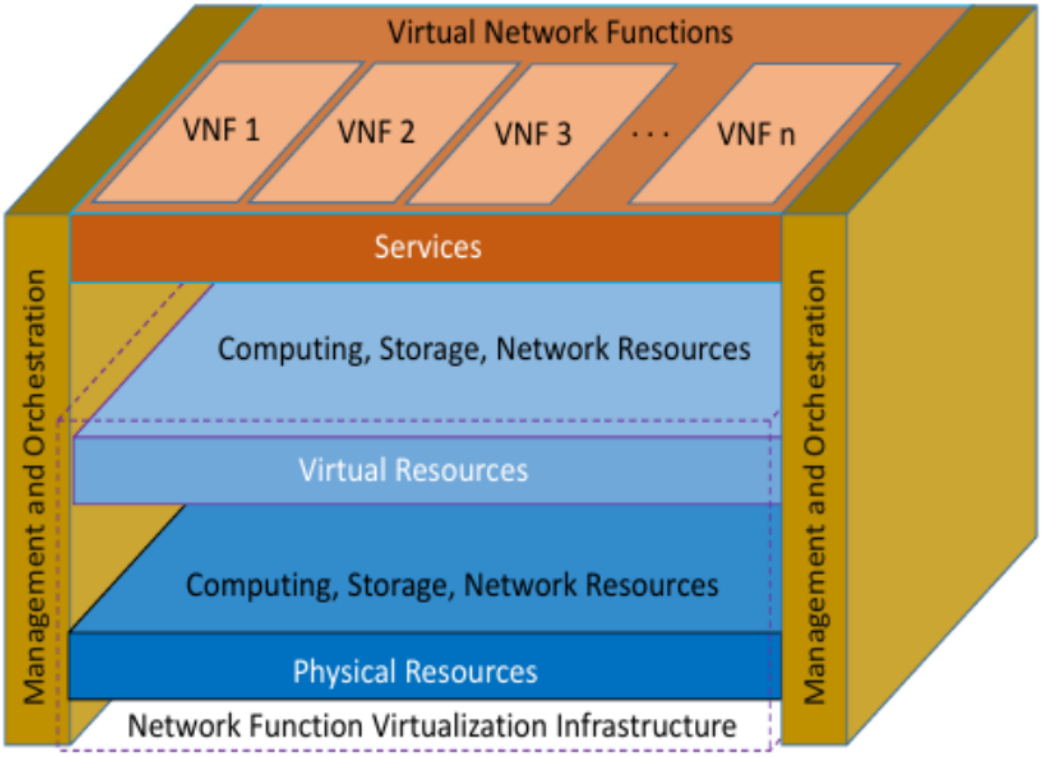

Figure 3: Architecture des fonctions réseau virtualisées (NFV)

Blockchain

La blockchain est une technologie décentralisée et sécurisée qui stocke les données de manière immuable dans des chaînes de blocs. Elle repose sur des mécanismes de consensus pour garantir la confiance sans intermédiaire et permet l'intégration de contrats intelligents pour automatiser les interactions. Dans le contexte de l'orchestration de tranches réseau multi-domaines, elle assure la confidentialité, la transparence et la résilience, bien qu'elle pose des défis de scalabilité et de performance selon son type (publique ou privée).

TEE (Trusted Execution Environments)

Les TEE sont essentiels pour garantir la confidentialité et la sécurité dans les architectures blockchain et multi-domaines. Ils permettent de traiter des données sensibles sans les divulguer, tout en protégeant les clés cryptographiques et les algorithmes critiques. Bien qu'efficaces pour sécuriser les systèmes, les TEE dépendent de matériels spécifiques et restent vulnérables à certaines attaques, posant des défis en matière de transparence et d'interopérabilité.

Rôle des TEE dans l'orchestration des réseaux multi-domaines :

Les TEE peuvent être utilisés dans des solutions d'orchestration décentralisées, permettant aux opérateurs de traiter des informations sensibles sans compromettre leur confidentialité. Dans les environnements multi-domaines, où des informations doivent être partagées entre plusieurs entités, l'utilisation de TEE assure la protection et l'intégrité des données, tout en permettant une gestion sécurisée des ressources réseau.

Figure 4: Schéma illustrant un Trusted Execution Environment (TEE)[1]

DKG

Le Distributed Key Generation (DKG) est un protocole cryptographique crucial pour les systèmes décentralisés, permettant de générer et gérer des clés de manière sécurisée sans autorité centrale. Il protège les données sensibles et garantit l'intégrité des échanges dans les environnements multi-domaines en distribuant les parts de clé entre les nœuds. Bien qu'exigeant en ressources, le DKG offre résilience et sécurité, ce qui le rend essentiel pour des systèmes comme les réseaux 5G et les architectures blockchain.

Rôle du DKG dans l'orchestration des réseaux multi-domaines :

Dans l'orchestration des ressources réseau multi-domaines, le DKG peut être utilisé pour gérer les clés cryptographiques nécessaires au chiffrement et à la validation des transactions. L'utilisation de DKG dans un environnement blockchain permet d'assurer que le partage de la clé secrète pour

l’orchestration et la gestion des ressources est sécurisé et qu'aucune entité unique n’a accès à l’intégralité des informations. Cela permet d’assurer la résilience et la sécurisation des échanges entre les différents domaines.

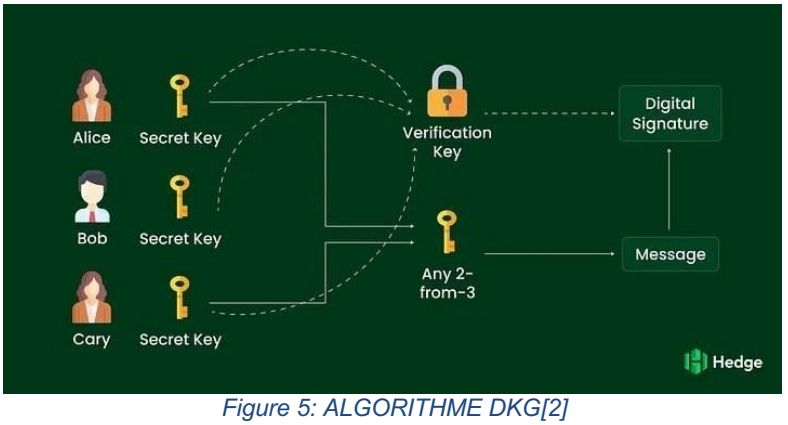

Figure 5: ALGORITHME DKG[2]

3.2. Solutions Traditionnelles

3.2.1. *La problématique des réseaux multi-domaines*

Défis des architectures traditionnelles dans les réseaux multi-domaines :

L’orchestration des ressources dans des **réseaux multi-domaines** représente un défi majeur. Dans les architectures traditionnelles, les ressources de plusieurs domaines différents (par exemple, opérateurs différents ou infrastructures géographiquement réparties) doivent être gérées de manière cohérente et optimisée. Cependant, les architectures **centralisées** qui dominent actuellement ne parviennent pas toujours à relever ces défis, notamment pour les raisons suivantes :

- **Isolation des données** : Dans une architecture traditionnelle, les opérateurs de réseaux doivent souvent partager des données sensibles, ce qui peut entraîner des **problèmes de sécurité** et de **confidentialité**. L'absence de mécanismes fiables de gestion de la confidentialité peut exposer ces données à des fuites ou des accès non autorisés.

- **Manque d’interopérabilité** : Les différents domaines de gestion des réseaux utilisent souvent des technologies incompatibles, ce qui crée des obstacles à l'intégration et à la coordination efficace des ressources réseau.

- **Risque de point de défaillance unique** : Les solutions centralisées sont vulnérables aux pannes du serveur ou aux attaques ciblant la **gestion centralisée**, ce qui affecte l’ensemble du réseau.

Manque de confidentialité et de sécurité dans les réseaux multi-domaines :

Les solutions centralisées présentent un risque élevé en matière de sécurité et de confidentialité des données. L’information circulant entre différents opérateurs de réseau est souvent exposée à des **attaques potentielles**, telles que des **attaques de type Man-in-the-Middle (MITM)** ou des **fuites de données**. Il est donc essentiel de garantir des **protocoles de sécurité renforcés** pour assurer la protection des informations

échangées entre les différents opérateurs, tout en permettant une gestion **dynamique et flexible** des ressources.

3.2.2. *Solutions traditionnelles d'orchestration de réseaux*

La littérature distingue trois stratégies dominantes pour l'orchestration des slices dans un environnement multi-domaine ; chacune présente, à des degrés divers, des lacunes devenues rédhibitoires dans un contexte 5G / 6G où la confidentialité, la disponibilité et l'élasticité sont désormais des exigences non négociables.

Premièrement, **l'architecture centralisée** confie à une instance unique – souvent qualifiée de *broker* – la coordination des ressources et des services entre opérateurs. Cette centralisation simplifie la supervision globale, mais elle oblige chaque acteur à divulguer ses paramètres sensibles (matrices de trafic, coûts, capacités), introduit un point de défaillance / de compromission unique et ouvre la porte à de possibles manipulations de l'allocation des slices.

Deuxièmement, **les solutions fondées sur le calcul multipartite sécurisé (MPC)** entendent préserver la confidentialité en répartissant le calcul entre les participants sans qu'aucun n'accède aux données brutes des autres. Si la garantie de secret est théoriquement élevée, le coût algorithmique l'est tout autant : la latence et la consommation de bande passante deviennent rapidement prohibitives dès que l'on souhaite orchestrer un grand nombre de domaines ou opérer en quasi-temps réel, deux conditions intrinsèques à l'orchestration de slices.

Troisièmement, **les cadres de type X-MANO** proposent un modèle d'information fédéré dans lequel chaque domaine transmet ses données à un gestionnaire externe censé en préserver la confidentialité. Cette confiance déléguée améliore la protection vis-à-vis des pairs, mais elle réintroduit une dépendance forte à l'égard d'une autorité centrale ; cette entité, si elle est défaillante ou compromise, compromet l'ensemble du système et annule de facto la promesse de résilience.

En définitive, qu'elle s'appuie sur un contrôleur central, sur des primitives MPC ou sur une fédération de type X-MANO, l'orchestration multi-domaine actuelle échoue à concilier simultanément confidentialité, robustesse et passage à l'échelle. Ces limitations justifient l'exploration de solutions décentralisées, résilientes par construction – par exemple celles combinant blockchain, consensus léger et enclaves TEE – afin de répondre aux exigences opérationnelles des réseaux modernes.

3.3. Solutions Basées sur la Blockchain

3.3.1. *Introduction à la Blockchain dans les réseaux*

La technologie **blockchain** constitue une réponse crédible aux défis de la gestion des ressources et de la confidentialité dans les réseaux multi-domaines. Contrairement aux approches fondées sur un orchestrateur central, elle repose sur un registre distribué où chaque transaction est validée collectivement, horodatée et rendue immuable ; l'absence d'autorité unique élimine le point de défaillance central, tandis que la cryptographie assure l'intégrité et la traçabilité des échanges. En

outre, l'automatisation par contrats intelligents permet de supprimer nombre d'intermédiaires et, partant, de simplifier la gestion et de réduire les coûts opérationnels.

Plusieurs travaux se sont appuyés sur des plateformes de référence :

- **Ethereum** offre un écosystème mature et une sécurité éprouvée (Proof of Work puis Proof of Stake), mais la congestion du réseau et la variabilité des frais de transaction nuisent à la prévisibilité des performances lorsqu'il s'agit d'orchestrer des ressources en temps quasi réel.

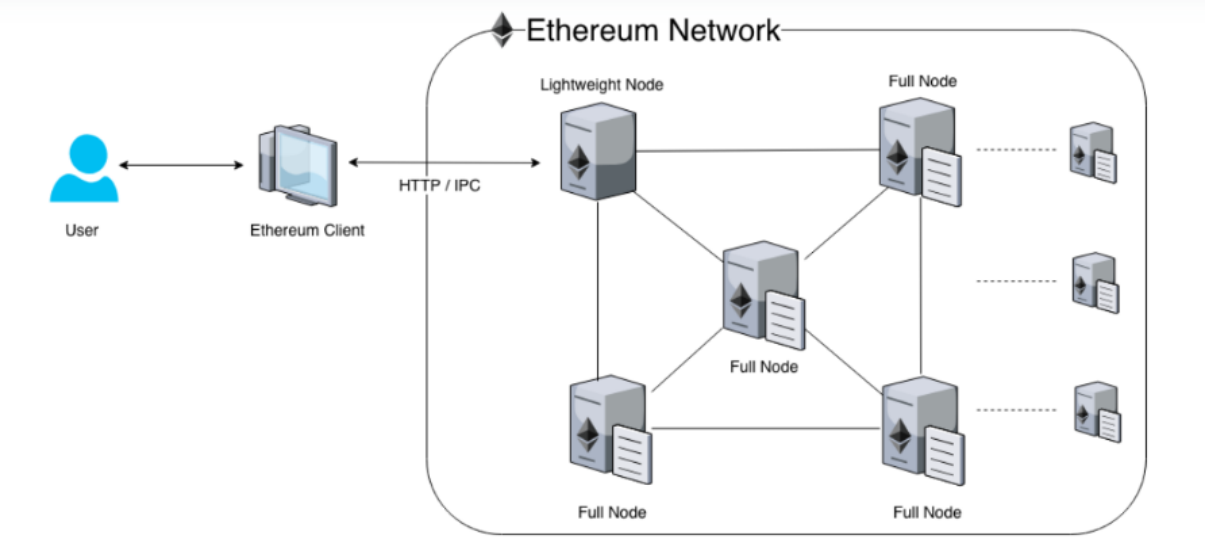

Figure 6: Blockchain Ethereum

- **Hyperledger Fabric** propose un modèle permissionné et modulaire adapté aux environnements industriels ; toutefois, le consensus Raft, fréquemment retenu, voit son débit décliner dès que le nombre de nœuds croît significativement, ce qui limite la scalabilité exigée par un contexte multi-domaine étendu.

Ces contraintes de performance et d'élasticité justifient l'exploration d'architectures plus légères, **NetChain** s'inscrit dans cette perspective en combinant un protocole de consensus optimisé avec des mécanismes de préservation de la confidentialité mieux adaptés aux exigences opérationnelles des réseaux modernes.

3.3.2. *Solution NetChain*

NetChain se présente comme un système d'orchestration de tranches réseau multi-domaines fondé sur le couple **blockchain + Trusted Execution Environments (TEE)**, combinaison destinée à préserver la confidentialité des opérateurs tout en garantissant l'intégrité des décisions d'allocation.

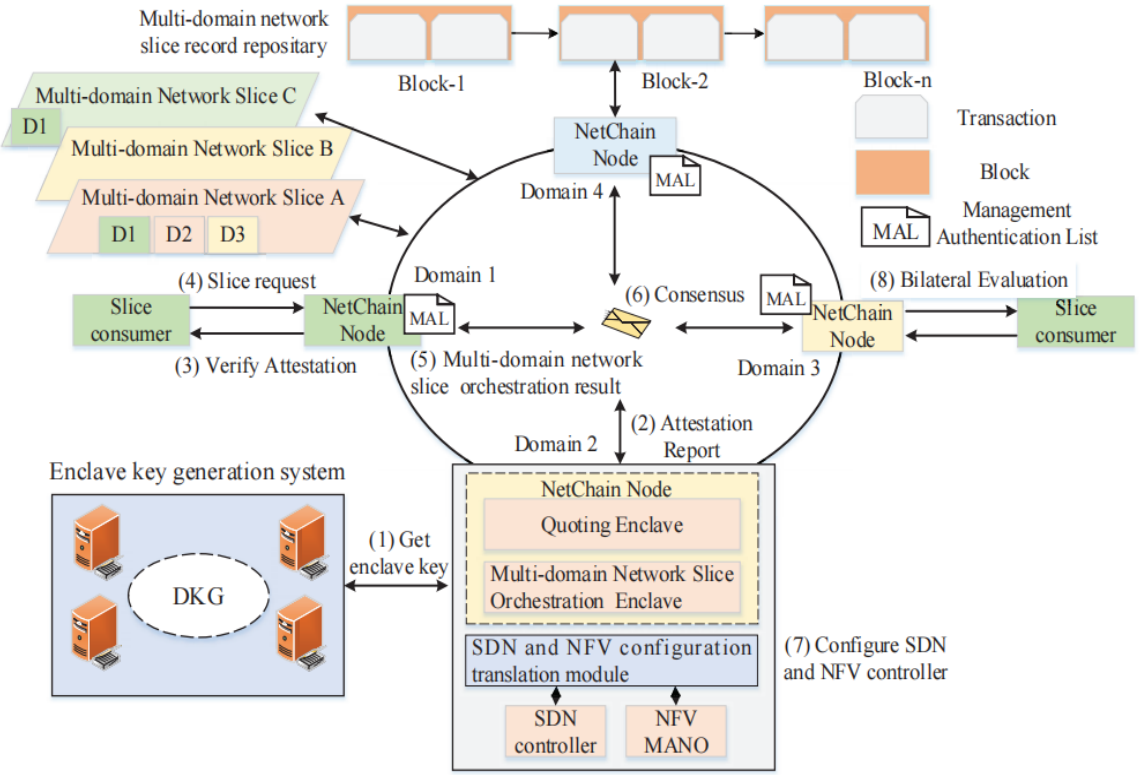

Figure 7: Architecture de Netchain

Avant de présenter les étapes de l'architecture, il est essentiel d'examiner plus en détail le nœud NetChain, cœur de l'architecture, qui joue un rôle central dans l'orchestration sécurisée des tranches réseau multi-domaines grâce à ses composants interconnectés.

Le nœud NetChain constitue le pivot de l'architecture : c'est lui qui orchestre les tranches réseau s'étendant sur plusieurs domaines tout en veillant, dans un environnement décentralisé, à la sécurité, à la confidentialité et à l'intégrité des échanges. Sa structure s'appuie sur quatre sous-ensembles coopérants. D'abord, la **Quoting Enclave** élabore la preuve cryptographique attestant que le code exécuté en enclave est bien authentique ; ce *quote* d'attestation, signé par l'Intel Attestation Service grâce à une clef EPID, garantit que le binaire n'a subi aucune altération et publie, avec la paire de clefs correspondante, l'empreinte de l'état matériel et logiciel. Ensuite, la **Network Slice Orchestration Enclave** exécute l'algorithme d'allocation multi-domaine : à partir de l'état courant du réseau, elle calcule la répartition des ressources nécessaires, chiffre le résultat (propriétaire de la tranche, cycle de vie, capacités à réserver) puis le diffuse pour validation collective avant de l'inscrire sur la blockchain ; en parallèle, elle génère les artefacts de configuration à destination des plans SDN et NFV. Pour que ces décisions reposent sur une vision cohérente du système, la Network **Status Enclave** maintient un registre sécurisé de l'état des ressources de chaque domaine et l'actualise après chaque consensus, offrant ainsi à l'enclave d'orchestration un accès constant à des données fiables. Enfin, le ledger blockchain conserve l'ensemble des artefacts : preuves d'attestation, décisions d'orchestration chiffrées et instantanés chiffrés de disponibilité (calcul, stockage, bande passante) ; ce registre distribué fournit l'historique immuable des opérations. Les configurations produites sont finalement traduites par le **module**

SDN/NFV vers les formats attendus par les contrôleurs et orchestrateurs MANO, qui procédent au déploiement effectif des fonctions et des ressources au sein de chaque domaine administratif.

Le cheminement d'une requête se déroule comme suit :

1. **Demande et attestation de slice.** Le consommateur formule une requête d'accès à une ressource ; cette requête est immédiatement scellée et attestée dans les enclaves TEE de chacun des domaines concernés, de sorte que son authenticité et l'intégrité de ses paramètres puissent être vérifiées avant tout traitement ultérieur.
2. **Validation et consensus.** Les nœuds NetChain agrégeront ensuite l'ensemble des demandes et les soumettront au protocole de consensus **CoNet** ; cette étape produit un verdict unique sur la validité des transactions inter-domaines.
3. **Orchestration multi-domaine.** La transaction validée est propagée aux contrôleurs SDN et NFV afin de provisionner la topologie ainsi que les fonctions réseau requises pour la tranche.
4. **Évaluation bilatérale.** Une évaluation ex-post confronte les engagements de service à la qualité réellement observée, garantissant ainsi que l'allocation demeure optimale et équitable pour les parties.
5. **Inscription on-Chain.** Les métadonnées afférentes (empreintes, accusés de réception, notes d'évaluation) sont finalement regroupées dans un bloc et ancrées dans la blockchain, ce qui assure traçabilité et immuabilité de l'ensemble du processus.

Le protocole de consensus CoNet :

CoNet dérive de l'algorithme **Practical Byzantine Fault Tolerance (PBFT)** : il en conserve la tolérance à $f = \lfloor (n - 1) / 3 \rfloor$ nœuds arbitraires, mais introduit deux adaptations majeures destinées aux environnements multi-domaine :

- **Sélection aléatoire des validateurs.** À chaque tour, l'ensemble des nœuds éligibles est échantillonné aléatoirement, proportionnellement à leur contribution effective (capacité ou disponibilité), ce qui répartit la charge et réduit les risques de collusion.
- **Signature collective BLS.** Les validateurs forment une signature agrégée de type Boneh–Lynn–Shacham ; la taille constante de cette signature simplifie la vérification par les observateurs externes et abaisse la latence de finalisation.

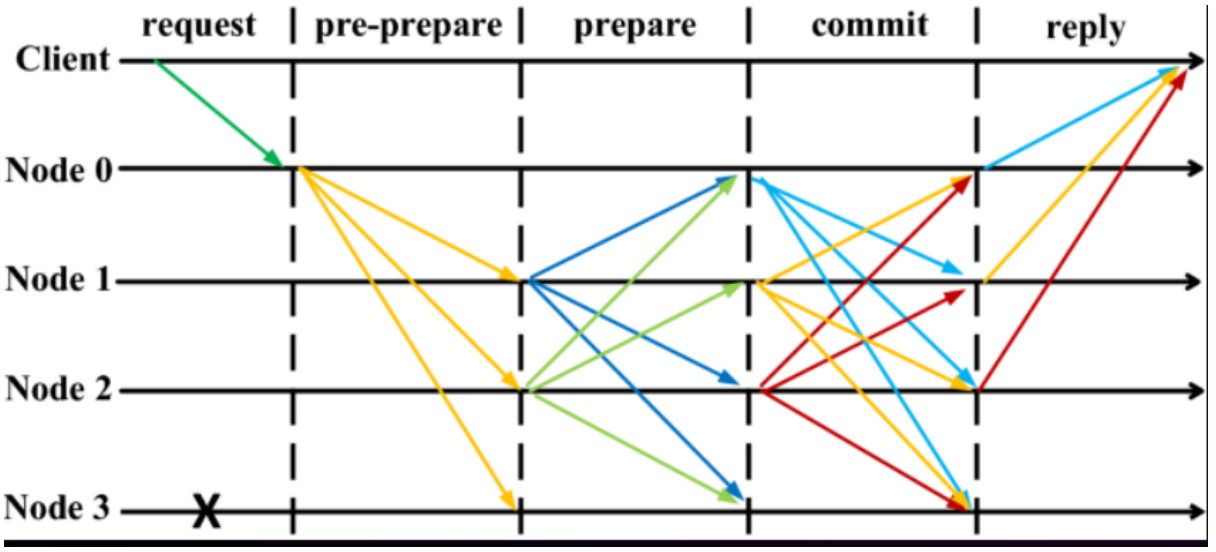

Figure 8: Algorithme PBFT

Le déroulement opérationnel s'effectue en quatre phases : sélection aléatoire d'un leader et des validateurs ; proposition d'un bloc regroupant les transactions d'orchestration ; validation par vote et génération de la signature collective ; diffusion puis ancrage définitif du bloc sur le registre.

Bénéfices attendus :

En agrégeant l'exécution protégée des TEE, la notarisation blockchain et l'efficacité de CoNet, NetChain ambitionne :

- **Une latence de validation réduite**, compatible avec les exigences temps-réel des réseaux 5G ;
- **Une sécurité renforcée**, la combinaison PBFT + TEE rendant l'attaque Sybil ou la falsification de blocs économiquement dissuasive ;
- **Une montée en charge linéaire** sur le nombre de transactions grâce à la signature BLS et à la répartition aléatoire des validateurs ;
- **Une efficience énergétique** très supérieure aux mécanismes de type Proof-of-Work, condition sine qua non pour un déploiement à grande échelle.

Ces caractéristiques positionnent NetChain comme une alternative plus adaptée aux impératifs d'orchestration multi-domaine que les implémentations reposant sur Ethereum ou Hyperledger Fabric, dont la scalabilité reste déjà sous contrainte dans les cas d'usage temps-réel.

Évaluation Bilatérale dans NetChain :

L'évaluation bilatérale est un mécanisme essentiel dans NetChain pour garantir l'équité, la qualité de service (QoS) et la qualité de l'expérience (QoE) lors de l'orchestration des tranches réseau multi-domaines. Ce processus repose sur la théorie des jeux et vise à limiter les comportements malveillants, comme la surévaluation des ressources par les fournisseurs de réseau.

Fonctionnement :

- Lors de l'évaluation, le fournisseur de réseau et le consommateur de tranche échangent des évaluations qui influencent directement leur réputation et leur crédibilité.

- Des ajustements sont effectués en fonction des écarts de capacité entre les ressources fournies et perçues.

Ce mécanisme favorise un équilibre de Nash dans lequel les deux parties maximisent leur bénéfice tout en réduisant les conflits.

4. Plan de développement

4.1. Introduction

Cette section détaille les approches adoptées pour atteindre les objectifs fixés, reposant sur une planification rigoureuse et une organisation efficace des tâches. Elle met également en lumière les méthodes employées pour explorer et identifier les solutions récentes d'orchestration des tranches de réseau multi-domaines, en analysant leurs performances, leurs limites et leur pertinence dans le contexte du projet.

4.2. Diagramme de Gantt

Le plan d'avancement de notre projet est représenté dans le diagramme de Gantt ci- dessous. Ce diagramme est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes tâches qui constituent notre projet en indiquant la plage temporelle à laquelle ces tâches doivent être effectuées.

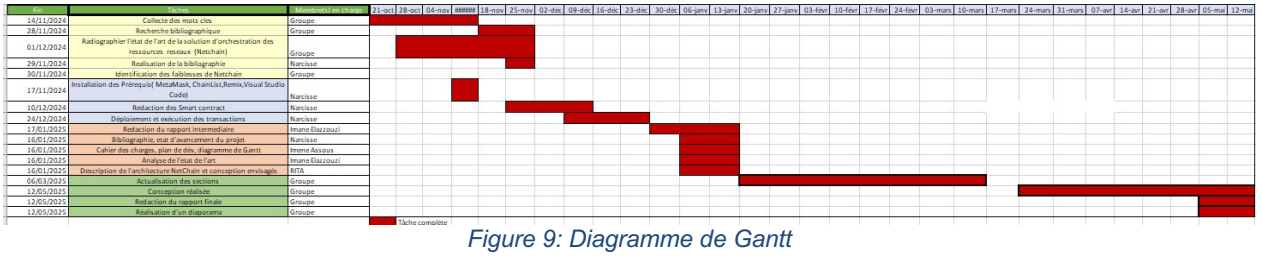

Figure 9: Diagramme de Gantt

4.3. Répartition des tâches

Pour mener à bien ce projet, nous avons adopté une approche structurée. Cette organisation a permis à l'équipe de travailler efficacement tout en respectant les délais impartis. Voici les principales étapes et la répartition des responsabilités :

4.3.1. *Collecte des mots-clés*

Cette phase initiale a permis d'identifier les termes techniques et concepts clés liés au projet. Ces mots-clés ont été utilisés pour orienter la recherche bibliographique et structurer les travaux. Ils incluent : blockchain, TEE, SDN, NFV, consensus décentralisé, QoS/QoE, et orchestration

multi-domaine.

4.3.2. *Recherche bibliographique*

Afin de détailler et analyser les choix du projet nous avons effectué une revue de littérature approfondie pour rassembler des sources académiques et des articles pertinents sur les technologies comme SDN, NFV, blockchain, et TEE.

4.3.3. *Radiographier l'état de l'art*

Cette étape a consisté à analyser les solutions actuelles en matière d'orchestration réseau. Les architectures centralisées et décentralisées ont été comparées, et leurs limites en termes de confidentialité, équité, et scalabilité ont été identifiées. Ces observations orienteront la conception de la solution proposée.

4.3.4. *Réalisation de la bibliographie*

Afin de faciliter les citations dans le rapport, nous avons structuré et formaté les références bibliographiques en suivant un style de citation standard. Cette démarche a permis de constituer une base documentaire claire et cohérente, rassemblant les sources académiques et techniques pertinentes.

4.3.5. *Identification des faiblesses de NetChain*

L'analyse de NetChain a mis en évidence des faiblesses majeures, notamment une dépendance aux plateformes blockchain tierces, des risques de sécurité et de scalabilité, ainsi qu'un partage transparent des informations compromettant la confidentialité. De plus, ses performances sont réduites en raison des coûts élevés liés aux mécanismes de consensus et aux environnements TEE.

4.3.6. *Installation des prérequis*

Avant le déploiement des transactions, nous avons d'abord procédé à la mise en place des outils nécessaires, notamment MetaMask, ChainList, Remix, et Visual Studio Code, afin de préparer l'environnement technique requis pour le bon déroulement du projet. Cette étape a assuré une base solide pour le développement et les tests des transactions.

4.3.7. *Rédaction des Smart Contracts*

Pour gérer les transactions et les interactions nécessaires à l'orchestration réseau, nous avons écrit les contrats intelligents en Solidity. Cette étape essentielle a établi les fondations techniques nécessaires pour réaliser des tests sur les réseaux blockchain, garantissant ainsi la validation et la fiabilité des mécanismes développés.

4.3.8. *Déploiement et Exécution des Transactions*

Les smart contracts ont été déployés sur le réseau testnet Ethereum Sepolia. Cette phase a permis de vérifier leur bon fonctionnement, d'identifier d'éventuelles anomalies et d'apporter des

ajustements afin d'améliorer leur efficacité et leur conformité aux exigences techniques du projet.

4.3.9. *Rédaction du Rapport final*

La rédaction du rapport final s'est faite de manière linéaire et structurée. Une distinction nette des sections a été faite entre les membres de l'équipe, chaque membre s'est ensuite concentré sur une partie spécifique du rapport. Chaque étudiant a rédigé sa section au fur et à mesure de l'évolution des tests et des résultats obtenus, les données et les connaissances techniques qui ont été développées au cours des phases précédentes ont toutes été écrites. Il se compose d'un cahier des charges, d'un plan de développement, d'une bibliographie réalisée grâce au tutorat de la MIR, d'une analyse, d'une conception envisagée et d'un état d'avancement du projet.

5. Implémentation

Ce chapitre décrit la matérialisation logicielle du prototype NetChain : l'environnement de développement retenu, la logique interne des contrats intelligents, le protocole de consensus CoNet, puis la chaîne de déploiement employé jusqu'aux premiers essais publics.

5.1. Environnement de développement

Le code est écrit en **Solidity 0.8.21** à l'aide de **Remix IDE v0.33.0**. Les tests unitaires s'exécutent localement sur **Ganache-CLI** afin de bénéficier de cycles courts de compilation-exécution. Lorsqu'une couverture de tests supérieure à 80 % est atteinte, les binaires EVM sont migrés sur le testnet **Sepolia** via un script *ethers.js* piloté par **MetaMask**. Ce triplet Remix / Ganache / Sepolia garantit à la fois reproductibilité (outils open-source) et représentativité (réseau public).

5.2. Contrats intelligents et logique métier

L'implémentation s'appuie sur cinq contrats autonomes, chacun associé à une responsabilité précise :

- **AttestationRegistry** assure la racine de confiance. Lorsqu'un opérateur rejoint NetChain, son enclave SGX transmet le hachage du rapport IAS ; le contrat conserve ce hachage avec la clé publique et l'identifiant de domaine, de sorte que toute transaction ultérieure puisse vérifier l'authenticité du nœud signataire.

- **SliceRequestManager** pilote le cycle de vie d'une requête de tranche réseau. Sa fonction submitRequest() crée une structure *SliceRequest* contenant l'ensemble des paramètres – ressources, contraintes de QoS, domaine demandeur, horodatage – puis publie l'événement *SliceSubmitted*. Les transitions d'état (*Pending*, *Validated*, *Provisioned*) sont gérées par updateStatus().

- **ConsensusLog** journalise le protocole CoNet. Lorsqu'un leader propose un bloc, submitProposal() inscrit l'empreinte dans la chaîne ; chaque validateur envoie ensuite son vote via voteOnBlock(). Quand le quorum est atteint, le contrat émet *BlockFinalised*, rendant la décision irrévocable.

- **ReputationManager** agrège les évaluations bilatérales échangées après exécution d'une slice. Chaque partie attribue une note numérique à son homologue ; ces scores, cumulés dans le contrat, pondèrent la probabilité qu'un nœud soit tiré au sort comme validateur au tour suivant.

- **ResourceLedger** maintient l'état courant des ressources proclamées par chaque domaine (capacité, bande passante, latence résiduelle). Avant toute allocation, l'orchestrateur hors-chaîne invoque checkAvailability() ; le contrat refuse la transaction si la demande excède les quotas, prévenant ainsi la sur-provision.

Tous les contrats émettent des événements structurés

(Par exemple *NodeRegistered*, *SliceSubmitted*, *BlockFinalised*) afin que le composant hors chaîne puisse reconstruire l'état global sans appels view coûteux.

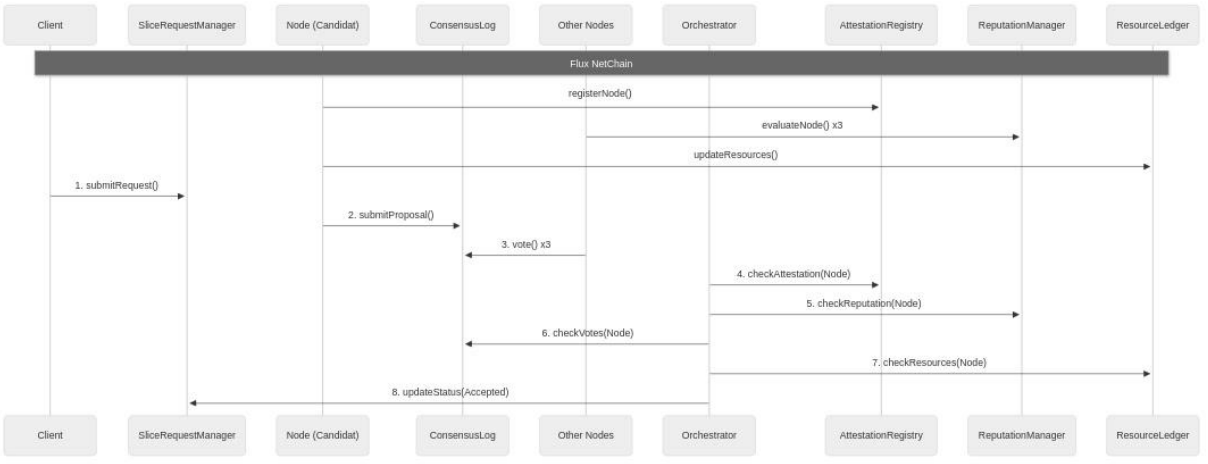

Figure 10: Enchaînement des appels dans NetChain lors de la création d'une slice réseau

5.3. Chaîne de déploiement

Le pipeline suit cinq étapes : (1) rédaction de chaque contrat dans un fichier .sol commenté NatSpec ; (2) compilation avec l'optimiseur Remix (200 passes) ; (3) exécution des tests sur

Ganache (sept nœuds simulés) ; (4) déploiement vers Sepolia par un script *deploy.js* qui renseigne les adresses dans deployments.json ; (5) injection de ces adresses dans l'orchestrateur SDN/NFV, lequel appelle les fonctions publiques via JSON-RPC. L'automatisation CI/CD complète (Hardhat + GitHub Actions) constitue un travail immédiat.

6. Résultats de l'implémentation

6.1. Tests et évaluation des performances

Les essais menés sur un environnement de machines virtuelles ont confirmé la réactivité et la stabilité du prototype : l'ajout de nouveaux nœuds s'effectue sans délai perceptible, les demandes de tranche réseau sont rapidement validées, et l'ensemble de la procédure se déroule de façon fluide, sans surcharge notable du système. Dans l'ensemble, le comportement observé est conforme aux attentes formulées au chapitre 2, ce qui atteste de la faisabilité pratique de l'architecture retenue et de son aptitude à être déployée dans un cadre opérationnel.

6.2. Limitations rencontrées et remèdes envisagés

• Surcharge **ponctuelle de Sepolia**: la latence augmente de 35 % aux heures de pointe. → Déploiement d'un testnet privé *Proof-of-Authority* pour les essais de charge.

• Collisions **d'écriture dans `ResourceLedger`**: plusieurs domaines publient simultanément. → Ajout d'un *mutex*optimiste (timestamp + nonce) et réflexion sur un sharding par type de ressource.

• Chaîne **CI/CD manuelle**: risque d'erreur humaine. → Mise en place de Hardhat + GitHub Actions (lint Slither, tests, déploiement).

• Boucle **SDN/NFV incomplète**: la démonstration reste hors ligne. → Intégration de l'API REST d'ONOS et d'OpenStack Tacker pour un provisioning de VNFs en temps réel.

7. Conclusion et perspectives

Le prototype NetChain valide la faisabilité d'une orchestration de tranches réseau multi-domaines fondée sur la combinaison **blockchain + TEE + consensus CoNet**. Les premières mesures confirment une latence compatible 5G, un coût transactionnel maîtrisé et une traçabilité exhaustive.

Les perspectives du projet se concentrent sur trois volets : d'abord, intégrer le prototype NetChain à un contrôleur SDN et à un orchestrateur NFV afin d'illustrer concrètement la création et la gestion d'une slice de bout en bout ; ensuite, déployer un petit réseau de test privé pour évaluer la latence et le coût de transaction lorsque le nombre de domaines augmente ; enfin, automatiser la chaîne de développement (compilation, tests et vérification de sécurité).

Bibliographie

1. F. Poltronieri, L. Campioni, R. Lenzi, A. Morelli, N. Suri, and M. Tortonesi, "Secure multi- domain information sharing in tactical networks," *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Los Angeles, CA, USA, 2018, pp. 1–6. https://doi.org/10.1109/MILCOM.2018.8599693
2. R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016. https://doi.org/10.1109/COMST.2015.2477041
3. P. Rost et al., "Mobile network architecture evolution toward 5G," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 84–91, May 2016. https://doi.org/10.1109/MCOM.2016.7470940
4. X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017. https://doi.org/10.1109/MCOM.2017.1600951
5. N. F. S. De Sousa, D. A. L. Perez, R. V. Rosa, M. A. S. Santos, and C. E. Rothenberg, "Network service orchestration: A survey," *Comput. Commun.*, vols. 142–143, pp. 69–94, Jun. 2019. https://doi.org/10.1016/j.comcom.2019.04.008
6. T. Taleb, I. Afolabi, K. Samdanis, and F. Z. Yousaf, "On multi-domain network slicing orchestration architecture and federated resource control," *IEEE Netw.*, vol. 33, no. 5, pp. 242–252, Sep./Oct. 2019. https://doi.org/10.1109/MNET.2018.1800267
7. T. Mano, T. Inoue, D. Ikarashi, K. Hamada, K. Mizutani, and O. Akashi, "Efficient virtual network optimization across multiple domains without revealing private information," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 477–488, Sep. 2016. https://doi.org/10.1109/ICCCN.2014.6911811
8. R. V. Rosa and C. E. Rothenberg, "Blockchain-based decentralized applications for multiple administrative domain networking," *IEEE Commun. Stand. Mag.*, vol. 2, no. 3, pp. 29–37, Sep. 2018. https://doi.org/10.1109/MCOMSTD.2018.1800015
9. G. Wood et al., "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project, Zug, Switzerland, Yellow Paper, vol. 151, pp. 1–32, 2014.

10. [3] G. He et al., "NetChain: A Blockchain-Enabled Privacy-Preserving Multi-Domain Network Slice Orchestration Architecture," IEEE Trans. Network Service Manag., vol.19, no.1, pp. 188–202, Mar. 2022. https://doi.org/10.1109/TNSM.2021.3110057

11. E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," Proc. 13th EuroSys Conf., 2018, p. 30. https://doi.org/10.1145/3190508.3190538

12. R. Cheng et al., "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts," *Proc. IEEE Eur. Symp. Security Privacy (EuroS&P)*, Stockholm, Sweden, 2019, pp. 185–200. https://doi.org/10.1109/MSEC.2020.2976984

13. A. Kate and I. Goldberg, "Distributed key generation for the Internet," *Proc. 29th IEEE Int. Conf. Distrib. Comput. Syst.*, Montreal, QC, Canada, 2009, pp. 119–128. https://doi.org/10.1109/ICDCS.2009.21

14. S. Bano et al., "SoK: Consensus in the age of blockchains," *Proc. 1st ACM Conf. Adv. Financ. Technol.*, 2019, pp. 183–198. https://doi.org/10.1145/3318041.3355458

15. Bilel Zaghdoudi, "Rapport Smart Contract : Guide de déploiement", document pédagogique fourni dans le cadre du projet de Master 1, Sorbonne Université, Octobre 2024.

16. [1]Sumsung Developers, “Understanding keystore architecture” https://developer.samsung.com/blockchain/keystore/understanding-keystore/keystore-architecture.html.[Consulté: Jan. 17, 2025].

17. [2]Coinmonks, "Distributed Key Generation (DKG) Techniques": https://medium.com/coinmonks/distributed-key-generation-dkg-techniques-bf1fe2581838. [Consulté: Jan. 17, 2025].

Annexe A

Mécanismes de consensus :

1. Proof of Work (PoW)

Principe de fonctionnement :
Le Proof of Work (PoW) est un mécanisme de consensus utilisé par certaines blockchains, comme Bitcoin. Dans ce modèle, les participants, appelés mineurs, doivent résoudre des problèmes mathématiques complexes (casse-têtes) pour ajouter un bloc de transactions à la blockchain. Ces problèmes sont difficiles à résoudre, mais une fois la solution trouvée, elle est facile à vérifier par le réseau.

Comment ça marche :

- Les mineurs tentent de résoudre un problème cryptographique par des calculs.
- Une fois la solution trouvée, le mineur ajoute un bloc à la blockchain, et le reste du réseau valide ce bloc.
- En récompense, les mineurs reçoivent des cryptomonnaies (par exemple, des bitcoins).

2. Proof of Stake (PoS)

Principe de fonctionnement :
Le Proof of Stake (PoS) est une alternative au Proof of Work (PoW). Dans PoS, au lieu de résoudre des problèmes mathématiques, les validateurs sont choisis en fonction de la quantité de cryptomonnaie qu'ils possèdent et sont prêts à "mettre en jeu" (c'est-à-dire à bloquer temporairement). Plus un participant détient de cryptomonnaie, plus il a de chances d'être choisi pour valider un bloc.

Comment ça marche :

- Les validateurs bloquent une certaine quantité de cryptomonnaie (leur stake) comme gage de leur engagement.
- Si un validateur est choisi pour valider un bloc, il vérifie les transactions et les ajoute à la blockchain.

- En récompense, les validateurs reçoivent des frais de transaction ou des cryptomonnaies supplémentaires.

Avantages :

- Moins de consommation énergétique : Contrairement à PoW, PoS ne nécessite pas de calculs complexes.

- Scalabilité améliorée : PoS permet un meilleur débit de transactions, avec une validation plus rapide.

Limites :

- Centralisation possible : Les grands détenteurs de cryptomonnaie peuvent avoir une influence disproportionnée sur la validation des transactions.

- Sécurité relative : Bien que PoS soit plus économe en énergie, il peut être plus vulnérable aux attaques si les acteurs malveillants contrôlent une grande partie des tokens.

3. Raft - Algorithme de Consensus

Raft est un algorithme de consensus utilisé dans les systèmes distribués pour assurer la cohérence et la synchronisation des données à travers plusieurs nœuds. Contrairement à d'autres algorithmes de consensus comme Paxos, Raft a été conçu pour être plus compréhensible et facile à implémenter tout en garantissant la fiabilité et la consistance des systèmes distribués.

Fonctionnement :

- Les nœuds sont organisés en leader et followers. Le leader est responsable des écritures et de la répartition des tâches.

- En cas de panne du leader, un processus d'élection est lancé pour en choisir un nouveau.

- Les followers répliquent les données et suivent les instructions du leader.

- Le processus de log réplication assure que les entrées du journal sont copiées de manière cohérente sur tous les nœuds.

Applications :
Raft est largement utilisé dans des systèmes comme les bases de données distribuées (etcd, Consul) ainsi que dans des systèmes de gestion de configuration et de services distribués, où la consistance et la tolérance aux pannes sont essentielles.

Annexe B

Outils de développement :

1. MetaMask

MetaMask est un porte-monnaie numérique pour interagir avec des dApps (applications décentralisées) et gérer des cryptomonnaies sur des réseaux compatibles avec Ethereum.

Fonctionnalités :

- Stocke et gère des clés privées pour interagir avec des réseaux blockchain.
- Permet aux utilisateurs de connecter facilement leurs portefeuilles à des dApps via le navigateur.

2. ChainList

ChainList est une plateforme permettant d'ajouter facilement des réseaux blockchain dans MetaMask. Elle permet aux utilisateurs d'ajouter des réseaux Ethereum et autres chaînes compatibles avec Ethereum en quelques clics.

Fonctionnalités :

- Permet de découvrir et d'ajouter des réseaux blockchain dans MetaMask via des liens directs.
- Facilite l'intégration avec des réseaux comme Binance Smart Chain, Polygon, etc.

3. Remix

Remix est un IDE en ligne pour le développement de smart contracts en Solidity et leur déploiement sur Ethereum.

Fonctionnalités :

- Fournit un éditeur Solidity pour écrire des smart contracts.
- Inclut un compilateur pour générer le bytecode nécessaire pour déployer les contrats sur la blockchain Ethereum.

- Offre des outils pour tester et déployer les contrats sur des réseaux de test ou sur la blockchain principale.

4. Visual Studio Code

Visual Studio Code (VS Code) est un éditeur de code source développé par Microsoft, largement utilisé pour de nombreux langages, y compris Solidity pour la programmation de smart contracts.

Fonctionnalités :

- Prise en charge de nombreux langages comme JavaScript, Python, C++, et Solidity.
- Extensions pour l'intégration de blockchain et d'outils de développement décentralisé.
- Débogage intégré et gestion des versions avec Git.

5. Solidity 0.8.21

Solidity est le langage de programmation principalement utilisé pour écrire des smart contracts (contrats intelligents) sur la blockchain Ethereum. La version 0.8.21 fait référence à une version spécifique de Solidity, qui comporte des améliorations et des correctifs.
Chaque version de Solidity introduit des mises à jour dans le langage, notamment pour la gestion de la sécurité et de l'efficacité du code.

6. Ganache-CLI

Ganache-CLI est une version en ligne de commande de Ganache, un simulateur de Blockchain Ethereum local. Il permet de simuler un réseau Ethereum pour tester des smart contracts de manière rapide et sans frais. Ganache-CLI permet une émulation locale de la blockchain pour tester des contrats avant de les déployer sur un réseau réel, comme Ethereum Mainnet ou un testnet.

7. Binaires EVM

Les binaires EVM font référence aux fichiers compilés des smart contracts écrits en Solidity qui sont compatibles avec la Ethereum Virtual Machine (EVM). L'EVM est l'environnement d'exécution des smart contracts sur la blockchain Ethereum, et les binaires EVM sont le code exécutable qui est déployé et exécuté sur la blockchain.

8. Testnet Sepolia

Le testnet Sepolia est un réseau de test d'Ethereum utilisé pour tester des contrats intelligents dans un environnement similaire au réseau principal Ethereum (Mainnet) sans risquer des fonds réels. Sepolia permet de simuler des transactions et des interactions sur la blockchain avant de déployer des contrats sur le réseau principal.

9. Script ethers.js

ethers.js est une bibliothèque JavaScript légère et puissante utilisée pour interagir avec la blockchain Ethereum. Elle permet de manipuler des smart contracts, de gérer des transaction et d'interagir avec des portefeuilles Ethereum, notamment à travers des outils comme MetaMask. Un script ethers.js est un programme qui utilise cette bibliothèque pour effectuer des actions sur la blockchain Ethereum, telles que la migration de binaires EVM sur un réseau comme Sepolia.

10. Hardhat :

Hardhat est un environnement de développement pour Ethereum qui permet de tester, déployer et déboguer des smart contracts. Il offre des outils d'automatisation pour les processus de compilation, de test, et de déploiement de contrats.

11. GitHub Actions :

GitHub Actions est un service d'automatisation qui permet de définir des workflows CI/CDsur GitHub. Il permet d'intégrer des tests, de déployer du code, et d'automatiser des processus à chaque modification du code source.

12. Slither :

Slither est un outil d'analyse statique pour smart contracts Solidity. Il permet de détecter des vulnérabilités et des erreurs dans les contrats intelligents avant leur déploiement.

13. JSON-RPC :

JSON-RPC est un protocole permettant de faire des appels remote procedure calls (RPC) en utilisant le format JSON. Il est utilisé pour interagir avec des nœuds Ethereum et pour envoyer des transactions ou des requêtes vers la blockchain.

14. CI/CD :

CI/CD est une méthode qui permet de tester et de déployer automatiquement les applications, garantissant ainsi des mises à jour fiables et rapides.

15. API REST :

API REST (Representational State Transfer) est une architecture de communication qui permet l'interaction entre les systèmes informatiques en utilisant les protocoles HTTP. Les API REST sont souvent utilisées pour permettre la communication entre des applications web ou des services.

16. ONOS :

ONOS (Open Network Operating System) est un système d'exploitation réseau open-source conçu pour la gestion des réseaux SDN (Software-Defined Networking). Il fournit une plateforme pour l'orchestration des réseaux et permet une gestion flexible des ressources.

17. OpenStack Tacker :

OpenStack Tacker est un service OpenStack qui permet l'orchestration et le provisionnement des fonctions réseau virtuelles (VNFs). Il permet aux opérateurs de gérer les ressources réseau de manière flexible et automatisée dans des environnements cloud, notamment pour les réseaux de cinquième génération (5G).

18. VNFs (Virtual Network Functions) :

VNFs sont des fonctions réseau qui sont virtualisées et peuvent être exécutées sur des machines virtuelles ou des containers. Elles sont une alternative aux équipements réseau matériels traditionnels, permettant une plus grande flexibilité et évolutivité dans les réseaux.