Chain-of-Shards : chaînes privées imbriquées avec finalité publique sélective

Résumé

Chain-of-Shards part d'un problème simple : les blockchains publiques sont immuables, mais chaque transaction y consomme un espace soumis au marché du gas. Chain-of-Shards déplace le volume transactionnel dans des chaînes privées imbriquées, où l'application exécute à très haut débit et paie principalement le coût du compute. La chaîne produit ses preuves dans son domaine privé, selon son profil de déploiement, puis publie sur Ethereum les checkpoints qui doivent devenir interopérables, auditables ou consommables par un contrat. Le gas est ainsi réservé aux moments de portée publique : bridge, règlement ou audit.

1 Introduction

Le trilemme de la blockchain est souvent présenté comme l'un des principaux défis structurels des réseaux blockchain : il met en tension la sécurité, la scalabilité et la décentralisation. Bitcoin et Ethereum se tiennent du côté sûr et décentralisé : chacun peut les vérifier, mais ils restent lents et très coûteux. Un second groupe privilégie la rapidité et le faible coût. Les cas de BNB Smart Chain et de Base sont éclairants. Leur interopérabilité est souvent confondue avec la décentralisation, or ce sont deux choses différentes. Ces chaînes sont en grande partie exploitées par une seule entreprise ou un petit ensemble de validateurs. Aujourd'hui, en somme, « rapide et interopérable » signifie presque toujours « rapide et centralisé ».

La propriété la plus précieuse d'une blockchain publique est aussi sa ressource la plus coûteuse : rendre un état immuable. Sur Ethereum, chaque opération inscrite entre dans un marché du gas, donc dans une logique d'enchère, de congestion et de coût global. Chain-of-Shards conserve cette immuabilité pour les moments où elle apporte une valeur externe, tout en déplaçant l'exécution quotidienne dans des chaînes privées rapides et vérifiables.

Cette distinction rejoint les travaux récents sur les rollups, les validiums, les bridges zero-knowledge et la disponibilité des données [1–3]. Elle précise aussi un point central de l'interopérabilité : un fait prouvé hors chaîne acquiert une portée différente lorsqu'il devient lisible sur la couche publique par un contrat de destination. Nos récents travaux de recherche ont formalisé cette frontière avec les états *Local*, *Proved* et *Checkpointed* [4].

Chain-of-Shards est une nouvelle génération de blockchain. Un shard est une chaîne privée appartenant à une famille imbriquée, rattachée à la même couche d'ancrage. Ensemble, ces chaînes forment un tissu polycentrique : chacune conserve son profil de déploiement et sa cadence de publication. Une seule racine publique (seed) peut porter de vastes familles de chaînes privées. Chaque chaîne s'exécute de manière autonome, paie principalement le compute de son domaine (un coût prévisible et faible), puis publie sur Ethereum uniquement les hauteurs qui doivent devenir interopérables, auditables ou consommables par un contrat.

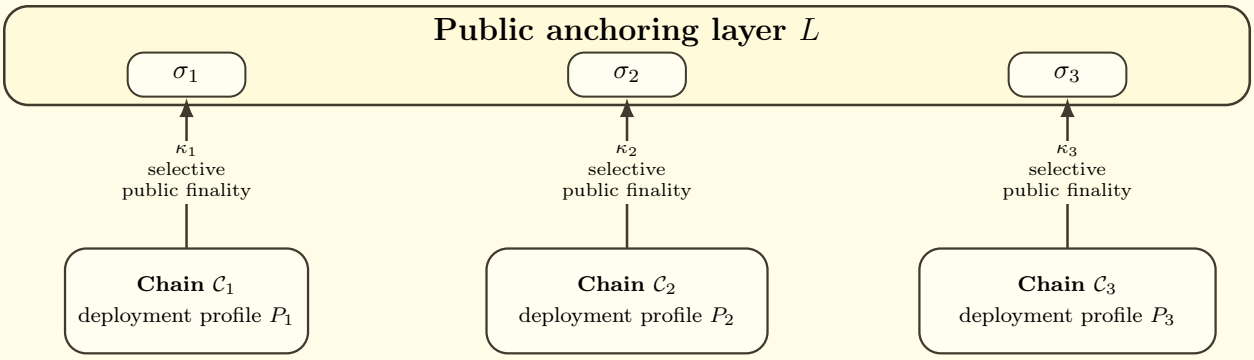

FIGURE 1 – Chaque chaîne part d'une seed publique, s'exécute dans son domaine privé et publie des checkpoints sur Ethereum lorsque certaines hauteurs doivent prendre une portée publique.

2 Définition d'une chaîne

Soit L une couche d'ancrage publique. Chaque chaîne imbriquée i part d'une seed publique :

$$\sigma_i = (id_i, g_i, \gamma_i, vk_i), \tag{1}$$

où id_i est l'identifiant de la chaîne, g_i l'engagement de genèse, γ_i la politique d'admissibilité inscrite dans la seed et vk_i la clé de vérification associée aux preuves de checkpoint. La politique γ_i engage le profil attendu, les validateurs, la règle de rotation, les délais de round et les paramètres qui rendent l'ordre local vérifiable.

La seed donne à la chaîne son origine publique. Elle fixe les paramètres qui permettent à un observateur, un validateur ou un contrat de rattacher une preuve à la bonne chaîne.

La chaîne qui en résulte est :

$$\mathcal{C}_i = (\sigma_i, P_i, \rho_i^0, \rho_i^1, \ldots), \tag{2}$$

avec le profil de déploiement réalisé :

$$P_i = (V_i, A_i, G_i, O_i). \tag{3}$$

Ici, V_i désigne les validateurs effectifs ; A_i décrit la réplication de l'état récupérable ou du matériel permettant de le reconstruire ; G_i précise les autorités de mise à jour ; O_i définit la règle d'ordonnancement locale.

Pour chaque chaîne, la seed ancre le profil de déploiement P_i dans une origine commune : identité, engagement de genèse, politique d'admissibilité et règles attendues de rotation et de vérification. Le profil P_i donne ensuite la lecture opérationnelle de cette configuration : qui valide, qui récupère, qui gouverne et qui ordonne. Chaque nœud repart de la même seed pour recalculer le validateur attendu à chaque round et vérifier que chaque bloc suit l'ordre annoncé.

Cette structure transforme la famille de chaînes en espace analysable. Une même couche d'ancrage porte alors des milliers de chaînes privées. Chacune conserve sa propre politique de publication. Certaines publient fréquemment pour servir des bridges ou des règlements rapides. D'autres accumulent des preuves privées et publient les hauteurs décisives selon leur politique. Toutes conservent une origine vérifiable et une relation de preuve identifiée.

L'engagement ρ_i^h résume l'état de la chaîne à la hauteur h. Il peut représenter une racine d'état, un engagement de journal, une sortie de calcul ou un digest applicatif. La précision importante tient au rattachement : chaque engagement appartient à une chaîne nommée, issue d'une seed publique, vérifiée sous une clé connue, puis éventuellement publiée sur L. Cette appartenance remplace les métadonnées de bridge isolées par une sémantique de chaîne complète.

3 *Zero-knowledge recursive state*

Le stade prouvé se lit comme un état zero-knowledge récursif. Pour une chaîne i et une hauteur h, l'engagement ρ_i^h décrit l'état courant, tandis que la preuve π_i^h établit la transition qui y mène. La récursion ajoute un lien explicite avec l'historique déjà prouvé : la preuve courante engage l'accumulateur de preuve précédent et produit un nouvel accumulateur.

On note a_i^h cet accumulateur récursif :

$$a_i^0 = \rho_i^0, \qquad a_i^h = \mathsf{Acc}(a_i^{h-1}, \rho_i^h, \pi_i^h).$$

Ici, Acc désigne une fonction d'accumulation cryptographique : elle combine l'accumulateur précédent, l'engagement courant et la preuve courante dans une valeur courte qui représente la continuité prouvée jusqu'à la hauteur h. En pratique, la preuve de la hauteur h vérifie la transition courante et confirme qu'elle prolonge l'accumulateur déjà accepté. L'observateur contrôle donc la dernière hauteur et la continuité de la chaîne, sans rejouer toute la séquence.

La relation de vérification peut alors se lire comme une vérification de hauteur et de continuité :

$$\mathsf{Verify}(vk_i, x_i^h, \pi_i^h) = 1.$$

Cette vérification confirme deux faits : la transition qui produit ρ_i^h est valide, et l'accumulateur a_i^h prolonge l'accumulateur précédent a_i^{h-1} sous la seed σ_i. Ce modèle de preuve met en évidence ce que l'observateur obtient : une preuve compacte de la transition courante et un lien cryptographique avec les preuves précédentes. L'état zero-knowledge récursif prépare ainsi la publication sélective. Tant que la hauteur reste privée, elle demeure vérifiable dans le domaine privé.

4 Les trois états

Nos récents travaux de recherche sur l'interopérabilité cross-chain ont formalisé trois états *Local*, *Proved* et *Checkpointed* [4]. Ces états forment une progression cumulative : un engagement local existe lorsque l'exécution le fixe. Il devient prouvé lorsqu'une preuve hors chaîne justifie la transition d'état. Il devient checkpointé lorsqu'une transaction inscrit la preuve et le rend consommable sur Ethereum.

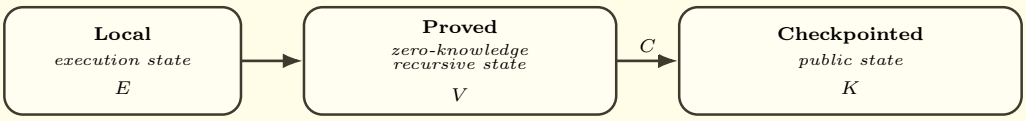

FIGURE 2 – Les trois états et la lecture opérationnelle $E \to V \to C \to K$.

Pour une hauteur h, la chaîne associe l'entrée publique x_i^h à une preuve π_i^h.

La vérification :

$$\mathsf{Verify}(vk_i, x_i^h, \pi_i^h) = 1$$

fait passer l'engagement au stade prouvé : le résultat local possède une justification cryptographique compacte.

Le checkpoint κ_i^h intervient lorsque cette hauteur doit être consommée sur L. Le checkpoint public porte trois éléments : l'identité de la chaîne, la hauteur et l'engagement prouvé. La preuve établit que le résultat est valide dans le domaine privé ; le checkpoint rend ce résultat utilisable dans le domaine public.

Cette architecture permet une gestion plus fine des coûts. La chaîne produit ses preuves dans son domaine privé. Les applications gardent leur cadence interne, puis publient les hauteurs nécessitant un ancrage public :

un jeu publiera une racine de tournoi, un système financier publiera un règlement, une chaîne d'entreprise publiera une preuve d'audit, un bridge publiera la hauteur utile à une sortie ou à un transfert inter-domaine.

5 *Checkpointability*

La *checkpointability* répond à une question pratique : la hauteur privée sur laquelle un participant agit serait-elle acceptée par le chemin public si elle était checkpointée maintenant ? Cette question a une valeur économique directe. La chaîne privée concentre l'exécution et la preuve dans son domaine ; Ethereum intervient lorsque la hauteur prend une portée publique. La *checkpointability* donne au participant le moyen de savoir, avant de payer le gas de publication, que la hauteur qu'il utilise possède un passage public admissible.

Nos récents travaux de recherche sur la *checkpointability* ont formalisé cette lecture avec quatre signaux : E pour la hauteur éligible, V pour la preuve vérifiée, C pour l'appel public accepté par simulation fidèle et K pour le checkpoint confirmé [5].

$$\mathbf{E} \longrightarrow \mathbf{V} \longrightarrow \mathbf{C} \longrightarrow \mathbf{K}$$

Un utilisateur qui agit sur une chaîne privée n'a pas besoin de publier immédiatement son action. Il lui suffit de vérifier la preuve. Si cette preuve est valide, il sait que cette hauteur pourra être ancrée publiquement à tout moment. Il peut donc continuer à opérer dans le domaine privé, reporter le paiement du gas, ou laisser un autre participant publier plus tard le checkpoint qui rendra l'état interopérable. Les actifs restent vérifiables dans le domaine privé, puis deviennent mobilisables sur Ethereum une fois le checkpoint correspondant confirmé.

La *checkpointability* donne à la chaîne privée un lien permanent avec Ethereum. Une hauteur prouvée peut rester privée tant que nécessaire, tout en étant publiable sur demande. C'est cette garantie de « publiabilité » qui permet d'opérer en confiance dans le domaine privé.

6 Immuabilité

Un engagement Chain-of-Shards suit un cycle lisible de bout en bout. L'application produit d'abord une transition dans son domaine privé. La chaîne calcule un nouvel engagement ρ_i^h, l'inscrit dans son historique local et rattache ce résultat à la seed σ_i. À ce stade, le fait existe pour les participants de la chaîne et s'insère dans la cadence locale définie par P_i.

Le cycle avance ensuite vers la preuve. Le validateur produit une preuve zero-knowledge, puis lie cette preuve à la bonne chaîne par la clé de vérification. La chaîne obtient alors un résultat prouvé : l'état local possède une justification cryptographique compacte. Dans une construction récursive, la preuve engage les preuves et engagements précédents, sans avoir besoin de tout rejouer. Cette étape porte la finalité locale. Elle donne aux validateurs et aux observateurs une manière de vérifier le résultat avec un coût de lecture très faible.

La publication constitue le troisième temps. La chaîne sélectionne une hauteur, prépare l'appel de checkpoint attendu, puis publie sur Ethereum. Le checkpoint transforme l'engagement prouvé en fait public. Un contrat de bridge, un système de règlement ou un auditeur peut alors consulter le checkpoint public, le rattacher à id_i, vérifier la preuve avec vk_i et consommer le résultat publié.

Une application peut rester rapide dans son domaine privé, garder ses preuves disponibles, puis rendre public le résultat qui porte une valeur externe. La racine publique (seed) devient le lieu de coordination, la chaîne privée reste le lieu de production, la preuve fait le lien entre les deux. Chain-of-Shards obtient ainsi une architecture où l'état circule par engagements, preuves et checkpoints.

7 Profils de déploiement

Le profil de déploiement $P_i = (V_i, A_i, G_i, O_i)$ rend la décentralisation observable.

TABLE 1 – Lecture opérationnelle du profil de déploiement.

Axe	Question rendue vérifiable
V_i validateurs	Qui exécute, prouve, signe et relaie les étapes de la chaîne.
A_i récupération	Où vivent l'état récupérable, les données de reconstruction et les preuves utiles.
G_i gouvernance	Qui autorise les mises à jour, rotations de clés, changements de paramètres et évolutions du chemin public.
O_i ordonnancement	Comment l'ordre local est produit, alterné, attesté et vérifié.

Le profil de déploiement P_i maintient séparés la récupération, la gouvernance et l'ordonnancement, chaîne par chaîne. Cette séparation transforme trois risques en trois surfaces d'attaque distinctes : la rétention de données, la gouvernance et la censure. Elle distingue deux systèmes en apparence identiques, mais qui se comportent très différemment une fois en service.

Certaines hauteurs peuvent rester significatives et prouvées localement sans devenir des faits publics. La finalité publique sélective devient donc un choix explicite : la chaîne décide quand son état doit compter publiquement et servir d'un domaine à l'autre. La cadence des preuves et la cadence des publications restent dissociées, ce qui donne aux applications un contrôle plus fin que l'habituel tout ou rien.

Le profil rend ces dimensions mesurables : l'indice de Herfindahl-Hirschmann quantifie la concentration des validateurs, tandis que le coefficient de Nakamoto mesure la robustesse à la censure. La décentralisation devient une mesure objectivable.

8 Défaillance de validateur

Une chaîne privée reste lisible même lorsque ses validateurs cessent de coopérer. Chain-of-Shards traite ce cas par construction : la seed σ_i reste publique et avec elle l'identité de la chaîne, son engagement de genèse et la clé de vérification vk_i. Un tiers peut donc rattacher une preuve existante à la bonne chaîne et la vérifier sans dépendre des validateurs V_i.

L'axe A_i du profil de déploiement rend visible la récupération de l'état. Un déploiement où les données de reconstruction restent sous le contrôle d'un seul validateur apparaît comme plus concentré ; un déploiement où elles sont répliquées hors de son contrôle apparaît comme plus robuste. La disponibilité de l'état devient ainsi une propriété lisible du profil, plutôt qu'une confiance implicite dans les validateurs courants.

La *checkpointability* borne ensuite le risque opérationnel. Toute hauteur déjà prouvée peut être publiée par un participant qui détient la preuve, l'engagement et l'appel public admissible. La défaillance ou la censure des validateurs actifs peut interrompre la production de nouveaux blocs ; l'historique prouvé conserve son intégrité et les hauteurs dont les preuves circulent gardent leur chemin d'ancrage public. La frontière du risque devient explicite : ce qui est prouvé et répliqué survit au validateur ; ce qui dépend d'une rétention unique apparaît dans le profil A_i avant l'usage.

Le point essentiel est simple : les validateurs actifs servent à faire avancer la chaîne, tandis que les hauteurs déjà prouvées vivent avec leurs preuves. Dès qu'un participant possède la preuve, l'engagement et l'appel public admissible, il peut faire reconnaître cette hauteur sur Ethereum. La seed identifie la bonne chaîne et A_i indique comment l'état se récupère. L'utilisateur sait ainsi distinguer l'exploitation courante de ce qui est déjà vérifiable et publiable.

9 Finalité

La finalité Chain-of-Shards progresse par niveaux. La chaîne privée produit des blocs sub-seconds ; leur finalité se renforce ensuite dans le temps, à mesure que les validateurs prouvent leur exécution par des preuves zero-knowledge. La seed reste le point de départ commun : elle porte l'identité, l'engagement de genèse et le

profil attendu, puis permet à chaque nœud de recalculer le validateur attendu, l'étape du round et la preuve à vérifier.

Finalité locale. Le bloc n est produit et devient visible dans la chaîne privée. Les nœuds l'acceptent lorsque le bloc suit l'ordre déterministe annoncé par la seed : bon round, bonne étape, bon validateur attendu et transition locale cohérente.

Finalité prouvée. La preuve zero-knowledge du bloc n arrive dans un bloc ultérieur $n + x$. Le validateur attendu pour cette étape produit la preuve, la signe et l'attache à l'état concerné. À ce niveau, le bloc n'est plus seulement visible : sa transition d'état est vérifiée par une preuve compacte, signée par le validateur.

Finalité différée. Un round complet donne à la chaîne privée sa finalité. Pour un round R dont le profil de déploiement fixe une rotation A, puis B, puis C, jusqu'à N, la séquence se lit ainsi :

$$\begin{aligned}
A :&\ \text{prouve et signe ;}\\
B :&\ \text{vérifie } A, \text{ puis prouve et signe un état qui engage } A;\\
C :&\ \text{vérifie } A+B, \text{ puis prouve et signe un état qui engage } A+B;\\
&\ \vdots\\
N :&\ \text{vérifie } A+B+\cdots+(N-1), \text{ puis prouve et signe un état qui engage } A+B+\cdots+N.
\end{aligned}$$

Chaque validateur vérifie que les étapes précédentes ont respecté la rotation attendue et les engagements déjà produits. Un bloc qui annonce un validateur inattendu ou engage un état inattendu est rejeté par les autres nœuds. À la fin du round, la chaîne possède plus qu'une preuve de transition isolée : elle possède une finalité différée, construite par accumulation de preuves et d'engagements récursifs.

Finalité publique. Une hauteur finalisée dans le domaine privé devient consommable sur Ethereum lorsqu'un checkpoint est inscrit et confirmé. Elle devient alors lisible et utilisable par un contrat, un bridge, un règlement ou un audit.

10 Où se situe Chain-of-Shards

TABLE 2 – Familles de travaux dans lesquelles s'inscrit Chain-of-Shards.

Famille de travaux	**Ce que ces travaux ont clarifié**	**Articulation dans Chain-of-Shards**
Rollups / validiums [1, 2]	Séparation entre exécution, preuve, disponibilité des données et règlement public.	Chaîne privée prouvée dont certaines hauteurs deviennent publiques par checkpoint choisi.
Validation côté client [6]	Vérification locale d'états ou d'actifs sans publication systématique.	Hauteurs privées prouvées, vérifiables dans leur domaine et mobilisables par checkpoint.
Sidechains privées ancrées [7, 8]	Valeur d'un domaine d'exécution dédié relié à une racine publique (seed).	Progression explicite entre état local, état prouvé et état checkpointé.
Séquenceurs et ordonnancement [9]	Importance de l'ordre, des rôles et de la résistance à la censure.	Profil P_i et finalité progressive par preuves zero-knowledge pour décrire qui valide, signe, relaie et vérifie.

Le Tableau 2 se lit comme une carte de continuités. Chain-of-Shards s'appuie sur les acquis des rollups, de la validation côté client, des sidechains ancrées et des travaux sur l'ordonnancement. Sa contribution consiste à réunir ces dimensions autour d'une unité explicite : la chaîne privée imbriquée, dont les hauteurs restent privées dans le flux courant et deviennent publiques par checkpoint choisi.

Cette composition se traduit par trois propriétés : premièrement, elle rattache chaque engagement à une seed. Deuxièmement, elle sépare la validité cryptographique de la portée publique. Troisièmement, elle rend le profil de déploiement visible comme partie intégrante de l'architecture de la chaîne. La finalité publique sélective apporte une granularité. Une application peut garder son exécution privée pendant que la chaîne prouve selon sa cadence interne, puis publier sur L les hauteurs pertinentes pour un bridge, une vérification externe, un audit ou un règlement.

11 Conclusion

Chain-of-Shards part d'un enjeu économique simple : toute transaction publiée sur Ethereum entre dans le marché du gas. Ce marché donne une sécurité et une vérifiabilité fortes, mais il devient coûteux lorsqu'il porte tout le volume applicatif. Chain-of-Shards déplace ce volume dans des chaînes privées rapides, où l'application paie principalement le compute et produit ses preuves à sa cadence.

La couche publique garde alors son rôle le plus précieux. Elle intervient lorsque le résultat doit sortir du domaine privé pour servir un bridge, un règlement, un audit, une sortie d'actifs ou un contrat sur Ethereum. Le checkpoint devient le moment de passage. Les gas fees accompagnent l'interopérabilité et les faits qui portent une valeur externe.

Cette séparation donne au modèle sa force : les transactions sont rapides dans le domaine privé, les preuves apportent une vérification cryptographique des transitions, la publication rend les hauteurs choisies interopérables. Chain-of-Shards permet ainsi de traiter un grand volume de transactions en quasi temps réel, tout en garantissant un haut niveau de décentralisation.

Références

[1] Louis Tremblay Thibault, Tom Sarry, and Abdelhakim Senhaji Hafid. Blockchain scaling using rollups : A comprehensive survey. *IEEE Access*, 10 :93039–93054, 2022.

[2] Stefanos Chaliasos, Denis Firsov, and Benjamin Livshits. Towards a formal foundation for blockchain ZK rollups. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 2714–2728, 2025.

[3] Muhammad Bin Saif, Sara Migliorini, and Fausto Spoto. A survey on data availability in layer 2 blockchain rollups : Open challenges and future improvements. *Future Internet*, 16(9) :315, 2024.

[4] Jonathan Oleszkiewicz and Babu Pillai. Cross-chain interoperability : Local, proved, and checkpointed states, 2026. Manuscript.

[5] Jonathan Oleszkiewicz. Checkpointability : Separating proof availability from publication liveness, 2026. Manuscript.

[6] Maxim Orlovsky. RGB I.0 : Scalable consensus for client-side validated smart contracts, 2025. IACR Cryptology ePrint Archive, Paper 2025/1400.

[7] Peter Robinson. Requirements for ethereum private sidechains, 2018. arXiv :1806.09834.

[8] Peter Robinson. The merits of using ethereum mainnet as a coordination blockchain for ethereum private sidechains. *The Knowledge Engineering Review*, 35 :e30, 2020.

[9] Shashank Motepalli, Luciano Freitas, and Benjamin Livshits. SoK : Decentralized sequencers for rollups, 2023. arXiv :2310.03616.