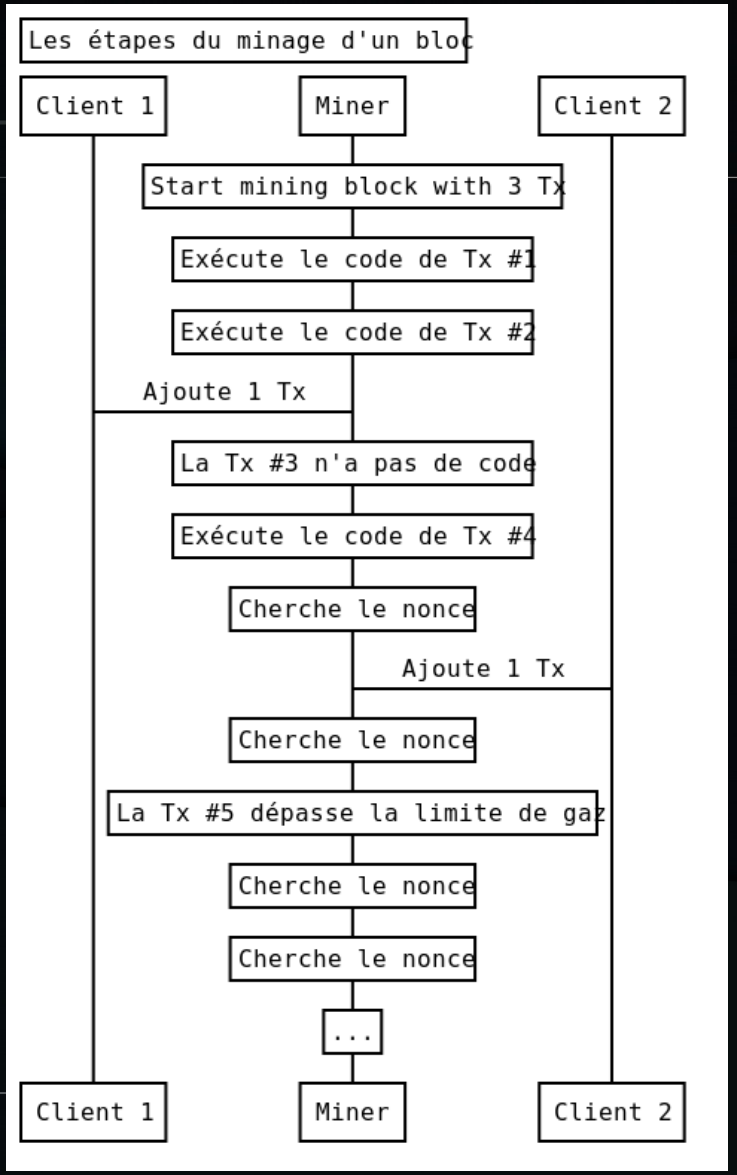
Le Futur d’Ethereum

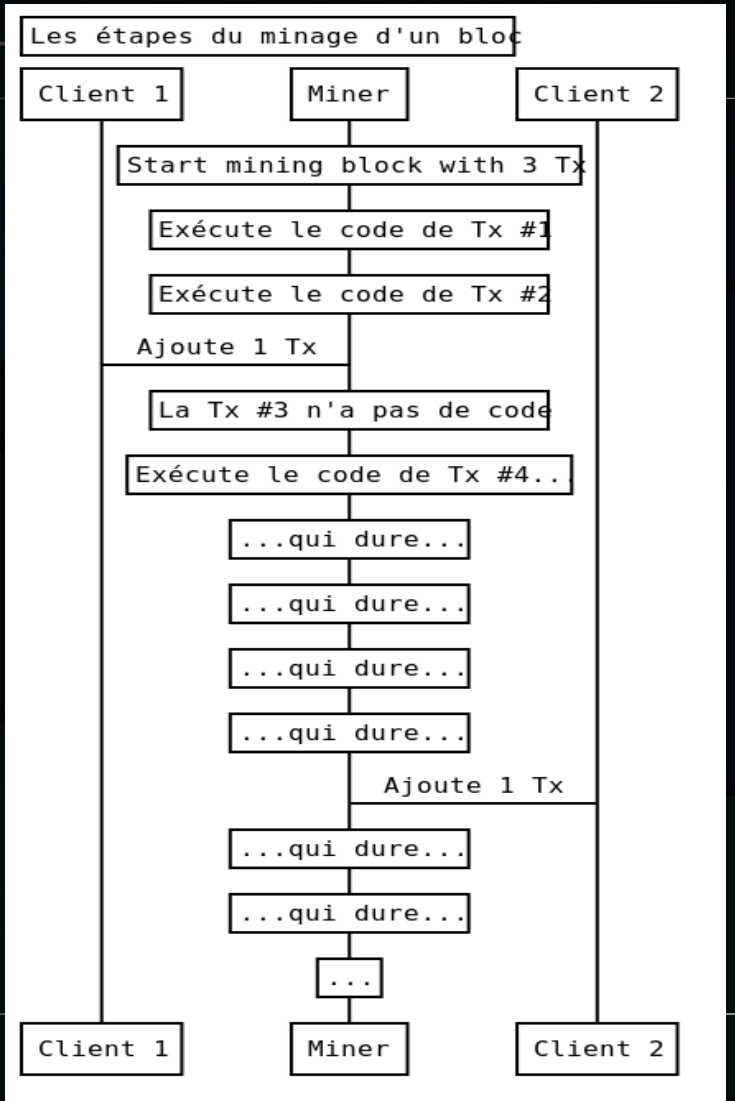
Minage d’un bloc

- Exécuter le code des transactions
- Vérifier que la consommation de gaz est inférieure à la limite
- Finaliser le bloc



Longs calculs

- Si un calcul est très long, les autres transactions sont retardées
- Pas de parallélisme



Proof of Work

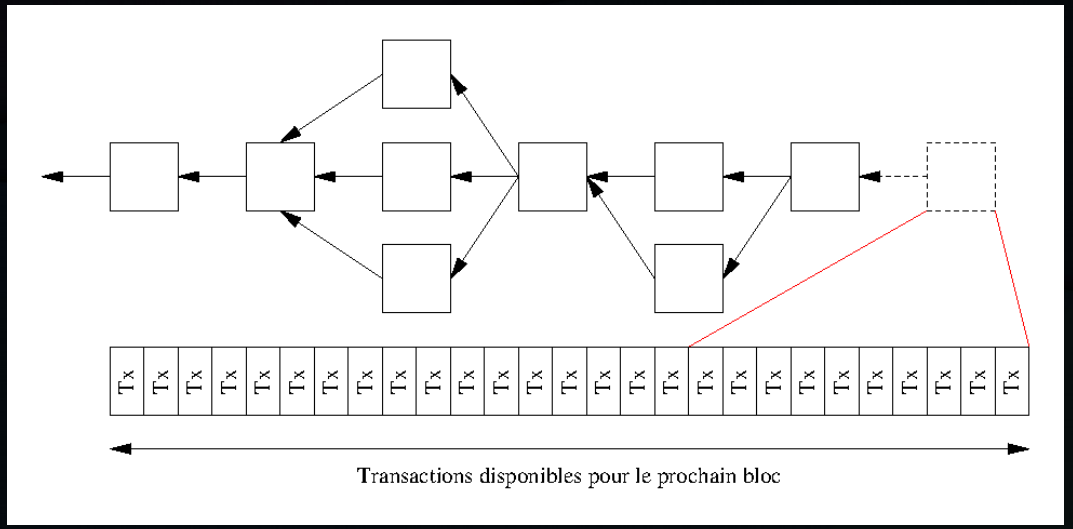
- Lotterie: beaucoup jouent, un seul gagne
- Énergie utilisée en pure perte pour la plupart des acteurs
- Ces problèmes se reflètent dans le prix du gaz

Stockage

- ~5GB rien que pour l’état des contrats
- 200GB+ pour la chaîne
- On ne paye qu’une fois pour toutes (les 0s sont moins cher)
- Synchronisation difficile

Trop de Transactions

- Peu de blocs proposés à la fois
- Il faut payer le gaz plus cher ou bien attendre son tour



Méthodes proposées

- Clients léger
- PoS
- Sharding
- Truebit
- eWASM
- Polkadot
- EIP-648
- Location d’espace
- ...

Client léger

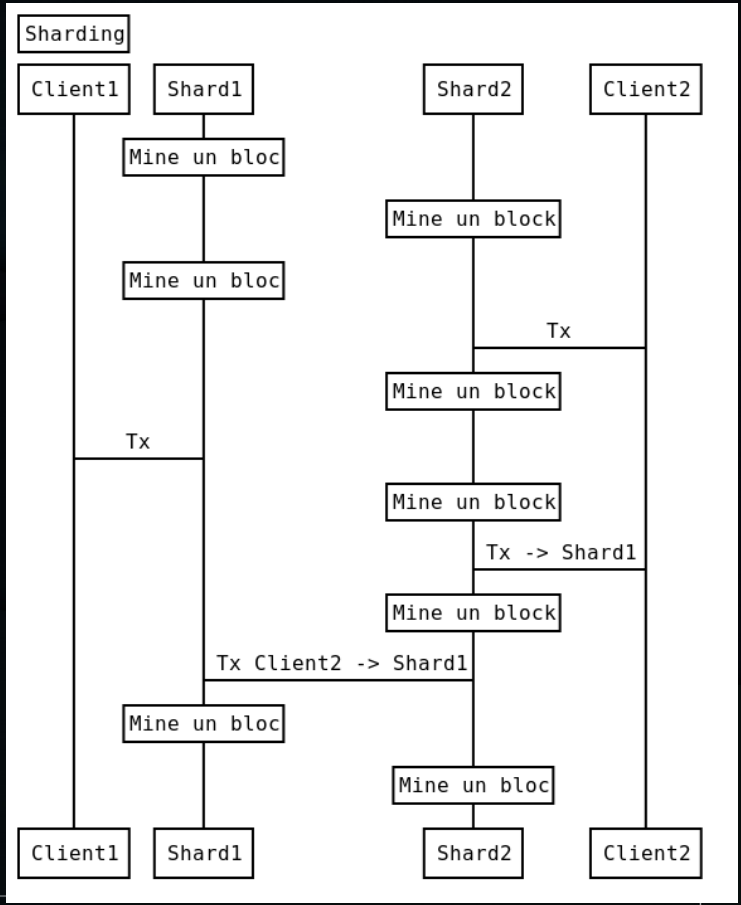
- ☺ Pas besoin de télécharger la blockchain en entier
- ☺ Au moins deux clients en développement (EF + Status)
- ☹ Besoin de faire confiance à un tiers
- ☹ Difficile de miner

Casper - PoS

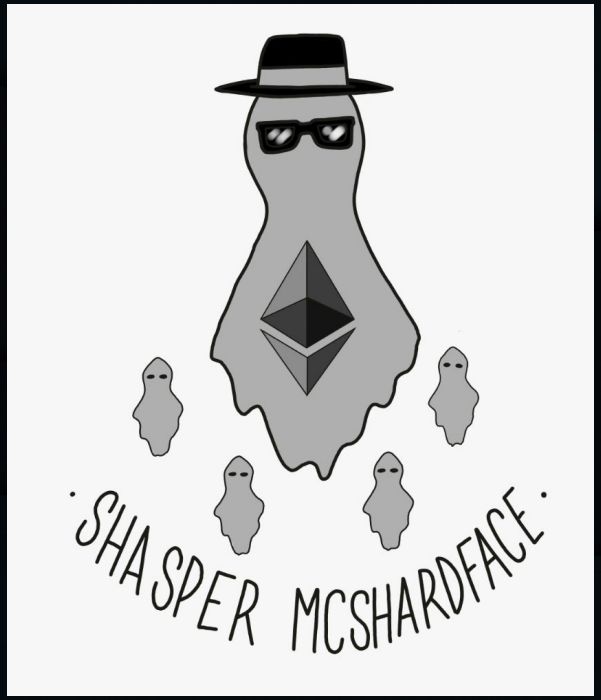
- ☺ Une seule machine choisie pour créer un bloc
- 😐 Probabilité d’être choisi $\propto$ valeur mise en jeu
- 😐 Plusieurs propositions (Vlad, Vitalik, …), attendu depuis des années...
- ☹ ...et abandonné la semaine dernière

Sharding

- Idée: diviser le traffic sur 512 sous-chaînes
- Possible spécialisation de chaque chaîne
- Coûteux de passer d’une chaîne à l’autre
- Permet d’experimenter d’autres technologies

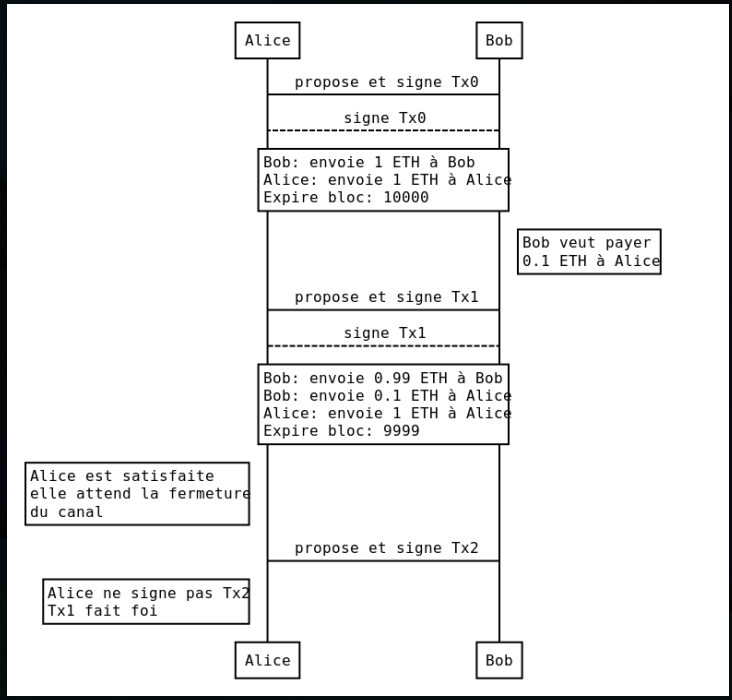


Casper + Sharding = “Shasper”



Raiden

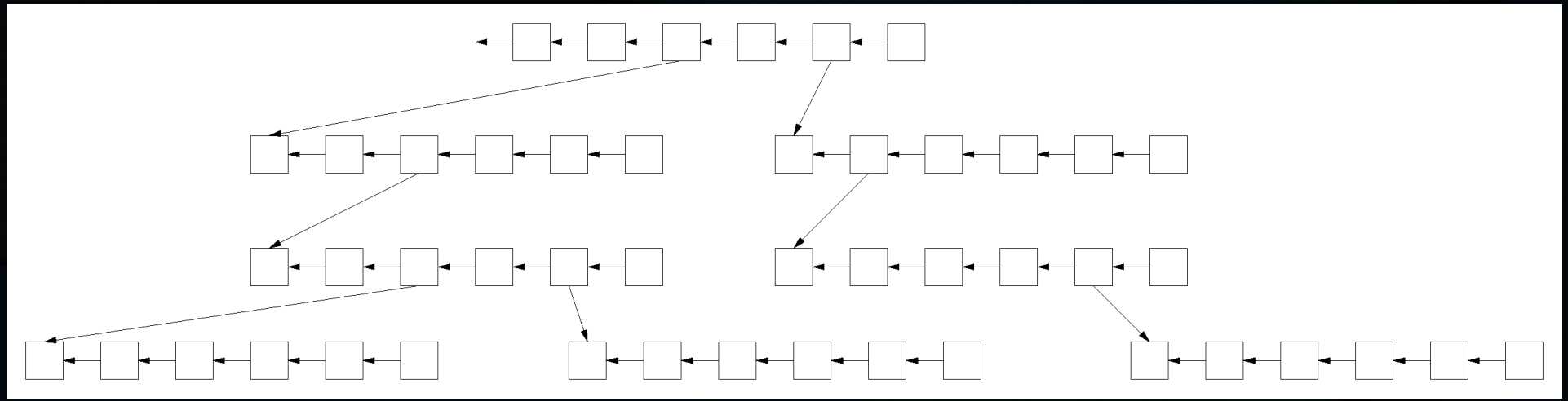
- Semblable à lightning
- Échange de transactions pré-signées = *channel*
- Équilibrage des comptes lors de la fermeture du canal



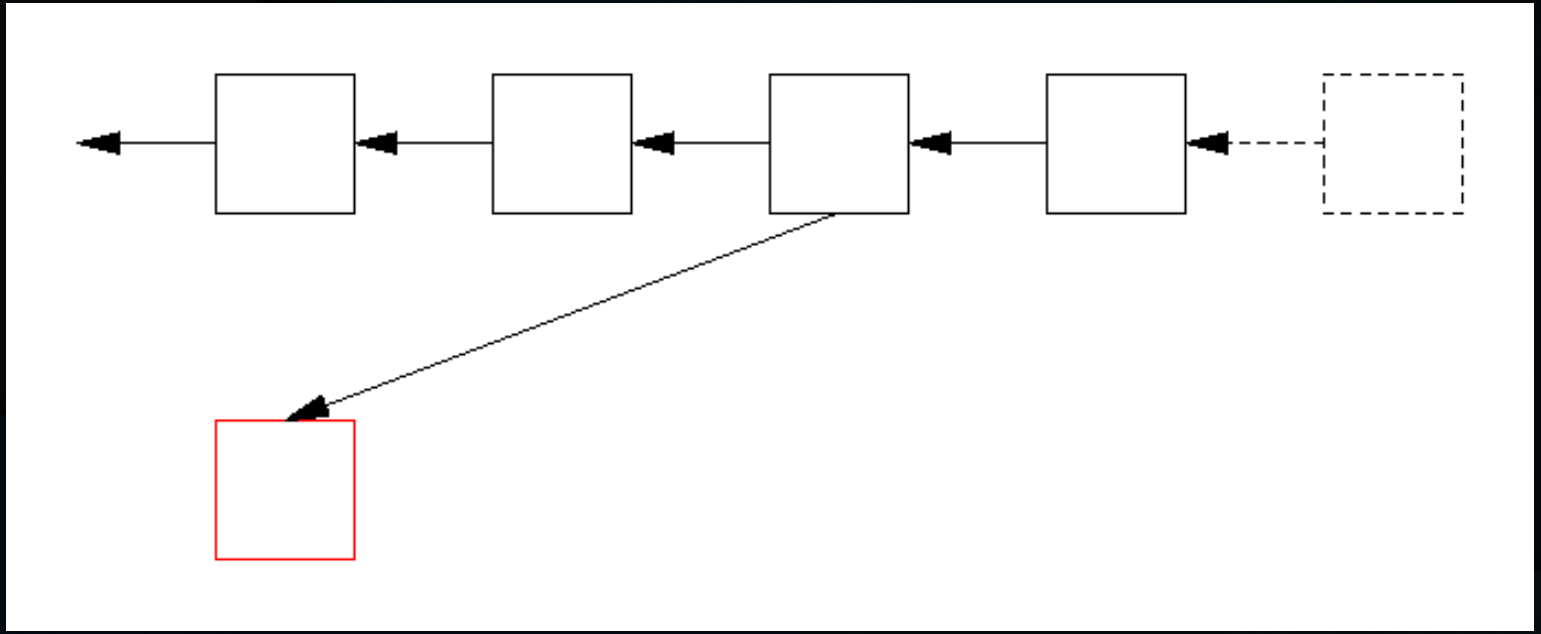
Plasma

- Arbre de blockchains
- Les feuilles sont rapides et peu coûteuses
- La chaîne racine est Ethereum
- Chaque noeud est garant des noeuds inférieurs
- Idée que le gestionnaire de chaîne à tout à perdre si la chaîne est détruite.

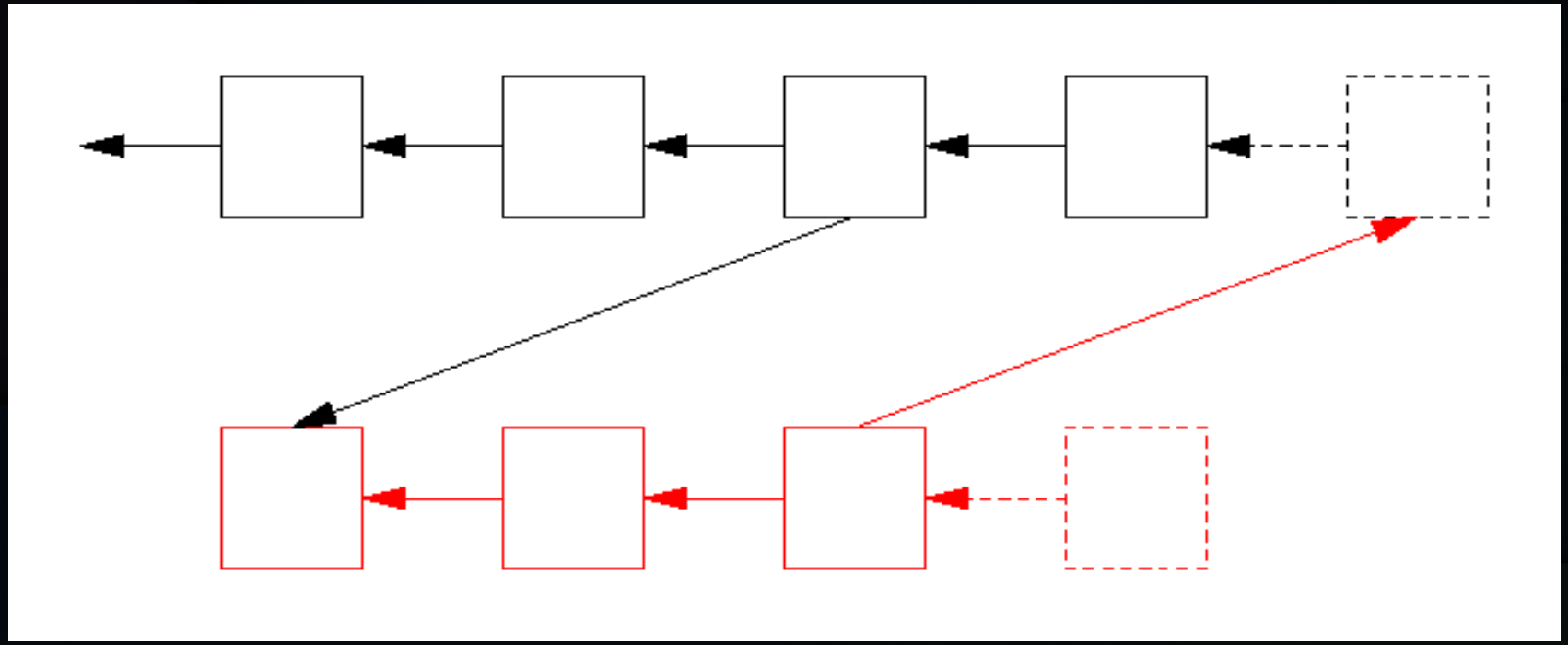
Plasma – Hiérarchie de chaînes



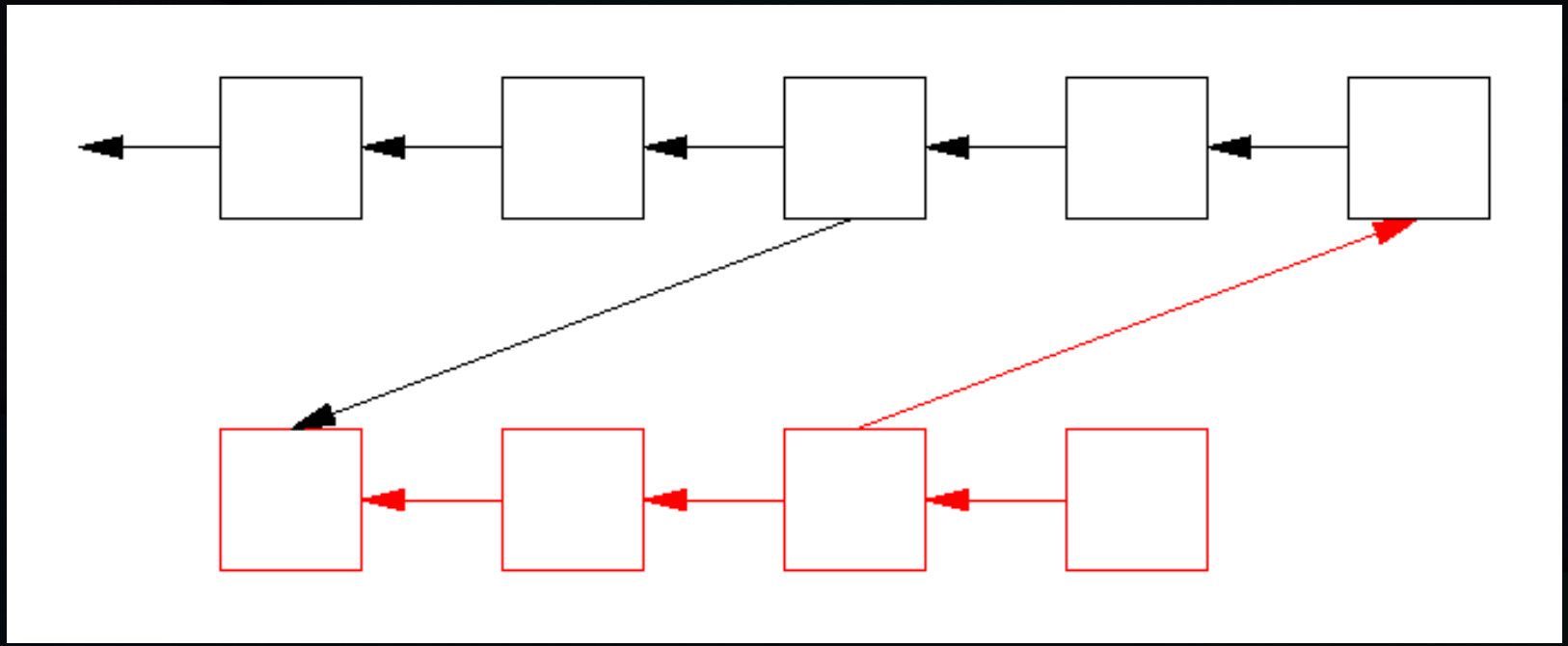
Plasma – Création d’une sous-chaîne



Plasma – Preuve de validité



Plasma – ajout de bloc

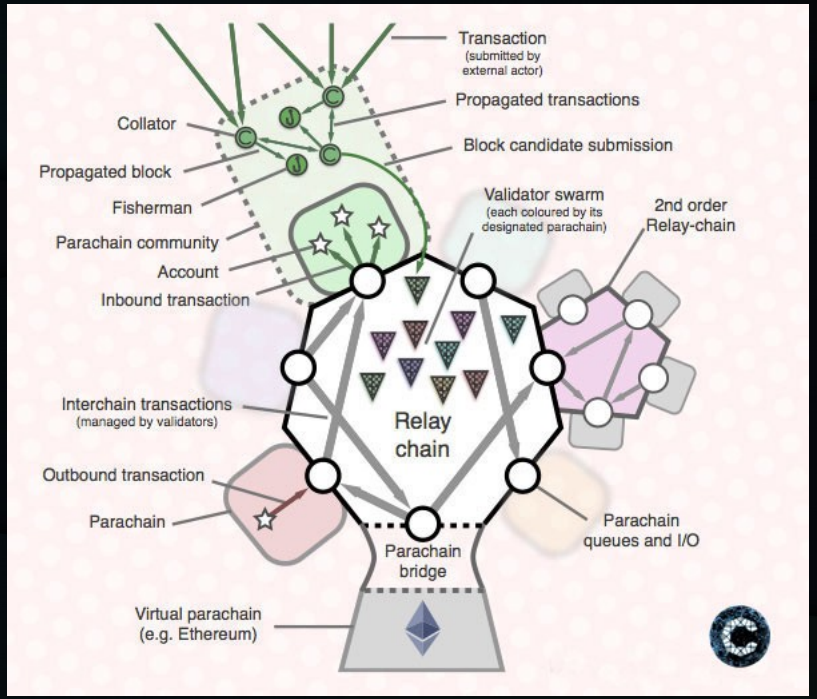


Truebit: Calcul hors-chaîne

- Idée: seulement un petit nombre de machines exécutent les contrats
- Contrôllé par un smart contract sur Ethereum
- Sommes mises en jeu pour garantir la bonne foi des participants
- Contrôle de la bonne exécution des contrats (zero-knowledge proofs)
- Fausses erreurs pour vérifier l’honnêteté de tous les acteurs

Polkadot

- “Internet of Blockchains”
- Connection entre différentes chaînes par une “chaîne-relai”



eWASM

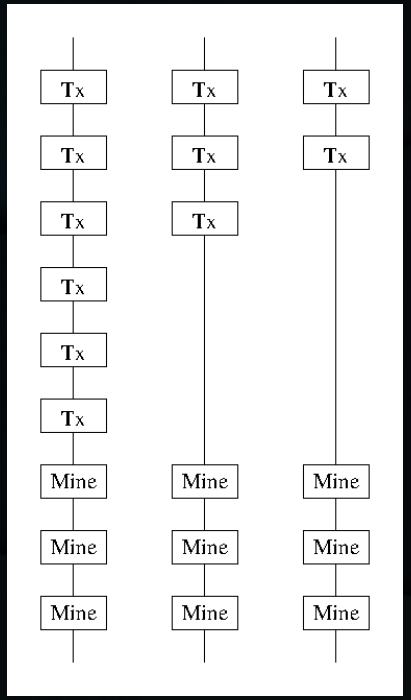
- WebAssembly – un format d’exécutables pour le web
- Les instructions EVM sont remplacées par des appels de fonctions
- Algorithmes écrits dans un seul langage et disponible pour tous (EOS, Polkadot, etc…)

eWASM

```rust
extern crate ethereum_bn128;
extern crate ewasm_api;
extern crate parity_bytes as bytes;

use bytes::BytesRef;

#[no_mangle]
pub extern "C" fn main() {
    // NOTE: no need to validate the input length as bn128_add will behave like EVM1.0 calldatacopy
    // add keep imaginary zeroes.

    ewasm_api::consume_gas(500);

    let input = ewasm_api::calldata_acquire();

    let mut output = vec![0u8; 64];
    match ethereum_bn128::bn128_add(&input[..], &mut BytesRef::Fixed(&mut output[..])) {
        Ok(_) => {
            ewasm_api::finish_data(&output);
        }
        Err(_) => {
            ewasm_api::revert();
        }
    }
}
```

eWASM – Exécution parallèle

- La plupart des contrats sont indépendants
- Exécution des contrats en parallèle
- Problèmes déjà résolus par les OS



Questions?

github+gitter+twitter:
@gballet