



Contship - Subnet - Proof of Concept and Design

Le blockchain pubbliche come Ethereum, Polygon e Solana offrono una rete pronta all'uso con costi di utilizzo bassi, ad eccezione delle commissioni di transazione. Tuttavia, la dipendenza da parametri predefiniti e dall'uptime della rete può essere un limite. Per i progetti che non richiedono parametri specializzati e necessitano di test rapidi o di messa in produzione rapida, le reti pubbliche possono essere vantaggiose, eliminando il lavoro di manutenzione e prevedendo solo costi di transazione.

Per organizzazioni come Contship, che adottano la tecnologia blockchain per la registrazione di eventi, la certificazione e i pagamenti tra aziende del gruppo, il servizio Subnet di Avalanche è una soluzione da considerare. I Subnet sono blockchain completamente personalizzabili, con permessi configurabili e caratteristiche dei nodi validatori. A differenza delle blockchain pubbliche, i Subnet offrono meno congestione, maggiore controllo e modularità. Tuttavia, questa soluzione prevede dei costi, come la manutenzione del server e la convalida dei nodi sulla mainnet di Avalanche, che richiede un minimo di 2000 AVAX.

Con questo PoC (Proof of Concept), l'obiettivo è dimostrare come un subnet Avalanche possa fornire la modularità che Contship cerca di ottenere e come potrebbe essere utile per il gruppo.

Accessi - Diritti amministratore

Questa prima fase è una delle più importanti perché determina chi può modificare le regole del subnet. Per una governance centralizzata che consenta di apportare modifiche velocemente e testarle rapidamente, è possibile assegnare i diritti esclusivamente a Contship. Tuttavia, per una migliore integrazione con i partner, sarebbe preferibile concedere loro anche i diritti amministrativi. L'obiettivo a lungo termine dovrebbe essere quello di avere un subnet decentralizzato nella governance, in modo che le regole del subnet possano essere modificate solo con un voto unanime.

Accessi - La smart contract pool

Questa "pool" servirebbe a raccogliere tutti gli smart contract sviluppati da Contship che sono stati testati e elaborati per avere un unico punto di riferimento per gli smart contract. Per creare questa pool, limitiamo i permessi di deployment degli smart contract a tutti gli utenti tranne i sviluppatori blockchain certificati da Contship. Ciò consente di mettere a disposizione sia delle aziende di Contship che dei partner una selezione di smart contract (ad esempio, certificazione di eventi, pagamenti automatici) che sono stati auditati da Contship. Inoltre, ciò consente di ridurre l'inquinamento del subnet e di avere solo gli smart contract desiderati pronti per l'uso.

Nodi - Localizzazione

Anche in questo caso, abbiamo due opzioni. La prima è di avere i nodi sui server di Contship. Questo ci consente di avere un controllo completo sui server e sul nodo, ma rende il subnet centralizzato nella localizzazione dei nodi. Il rischio maggiore sarebbe che, in caso di problema ai server di Contship, il subnet vada giù. Per un approccio più resiliente e decentralizzato, si consiglia di avere i nodi del subnet su servizi cloud come AWS (che ha una partnership con Avalanche e offre istanze EC2 con il nodo pronto all'uso, al costo di 250\$/mese per un singolo nodo). L'idea sarebbe quella di avere nodi gestiti da AWS in diversi paesi per decentralizzare il subnet e aumentare la resilienza della rete. Nel nostro design, implementeremo i nodi sui server di Contship.

Nodi - Requisiti Hardware

Con i subnet, è possibile definire i requisiti che i nodi devono soddisfare per poter convalidare il subnet. Un esempio di requisiti potrebbe essere l'hardware. Se il nodo è su un server con capacità hardware inferiori rispetto a quelle richieste dal subnet, potrebbe rischiare di rallentare la rete. Nel nostro caso, stabiliremo dei requisiti hardware che saranno quelli consigliati da Avalanche:


- CPU: equivalente di 8 vCPU AWS
- RAM: 16 GB
- Archiviazione: SSD da 1 TB
- Sistema operativo: Ubuntu 20.04 o MacOS >= 12

Ciò ci consentirà di avere nodi pronti per qualsiasi tipo di traffico di base sulla rete. Nel nostro caso, vogliamo una rete che si occupi della certificazione degli eventi e dei pagamenti, quindi non abbiamo bisogno di nodi più potenti. Nel caso in cui dovesse essere avviato un progetto che richieda più risorse, possiamo modificare i requisiti dei nodi per essere certi che i nodi possano supportare il subnet.

Nodi - Quantità

Il numero di nodi da definire dipende dal grado di resilienza desiderato per il nostro subnet. Se si utilizza un solo nodo, il rischio di mettere giù la rete è molto elevato, in quanto la rete dipende da quel singolo nodo. Avalanche consiglia un minimo di 5 nodi, ma questo approccio può risultare costoso. Per il nostro subnet, il minimo accettabile sarebbe di 2 nodi, mentre l'opzione ottimale sarebbe di 3 nodi. In questo modo, possiamo aumentare la resilienza della rete senza aggiungere costi non necessari.

Subnet - Fees

 Per questa parte del Design, utilizzeremo gli stessi parametri della blockchain principale di Avalanche, in quanto non abbiamo bisogno di un subnet per un uso specifico a questo momento.

Le fees sono un elemento fondamentale di qualsiasi blockchain, in quanto consentono il corretto funzionamento dell'algoritmo di consenso e della creazione dei blocchi.

Innanzitutto, possiamo configurare il `gasLimit`, ovvero il parametro che gestisce la quantità di computazione che può essere effettuata in un singolo blocco. Nel nostro caso, abbiamo impostato il `gasLimit` a `15.000.000`.

Successivamente, abbiamo il `targetBlockRate`, che determina la produzione di blocchi al secondo. Più il `targetBlockRate` è alto, più il base fee aumenterà e le transazioni saranno costose. Nel nostro caso, abbiamo impostato il `targetBlockRate` a `2`.

Poi c'è il `minBaseFee`, che stabilisce un costo minimo di transazione per qualsiasi transazione. Nel nostro caso, abbiamo impostato il `minBaseFee` a `60.000.000.000`.

Inoltre, abbiamo il `targetGas`, ovvero la quantità di gas che verrà consumata nell'arco di 10 secondi. Se la rete si accorge che le transaction fees sono molto diverse dal `targetGas`, la rete cercherà di stabilizzare il gas aggiustando il `baseFee`. Nel nostro caso, abbiamo impostato il `targetGas` a `10.000.000`.

Poi abbiamo il `baseFeeChangeDenominator`, che divide la differenza tra il `baseFee` e il `targetGas`. Se il denominatore è basso, il `baseFee` cambierà lentamente, mentre se è alto, il `baseFee` verrà aggiustato rapidamente. Nel nostro caso, abbiamo impostato il `baseFeeChangeDenominator` a `36`.

Inoltre, c'è il `minBlockGasCost`, ovvero il gas minimo richiesto dalla rete per la produzione di un blocco. Nel nostro caso, abbiamo impostato il `minBlockGasCost` a `0`.

Poi abbiamo il `maxBlockGasCost`, ovvero il gas massimo richiesto dalla rete per la produzione di un blocco. Nel nostro caso, abbiamo impostato il `maxBlockGasCost` a `5.000.000`.

Infine, abbiamo il `blockGasCostStep`, che determina il costo del gas per la produzione di un nuovo blocco in base al tempo trascorso tra il nuovo blocco e quello precedente. Nel nostro caso, abbiamo impostato il `blockGasCostStep` a `10.000`.



In futuro, si potranno modificare le fees per accelerare la produzione dei blocchi se si avrà bisogno di una rete più veloce o di una rete in grado di eseguire calcoli complessi. È importante tenere presente che maggiore è la velocità della rete, maggiori saranno le fees.

i Una rete senza fees non può funzionare correttamente, in quanto i blocchi richiedono gas per essere prodotti. Pertanto, è sempre necessario impostare un minimo nei parametri delle fees.

VM - Più scelte

Un subnet di Avalanche offre la possibilità di avere più blockchain contemporaneamente, ognuna con la propria virtual machine (VM) e le proprie caratteristiche. In particolare, la VM di base supportata è quella di Ethereum, che ci permette di utilizzare smart contract sviluppati con Solidity.

Tuttavia, nel caso in cui si voglia migliorare la velocità e la robustezza degli smart contract, è possibile valutare l'aggiunta di un'altra VM, come ad esempio quella di Solana, che supporta lo sviluppo di smart contract in Rust. In questo modo, si potrebbe adottare una strategia ibrida in cui una blockchain viene utilizzata per la certificazione e i pagamenti con smart contract Solidity, mentre l'altra blockchain viene utilizzata per eseguire azioni che richiedono maggiore velocità e robustezza con smart contract sviluppati in Rust. Nel nostro design attuale, implementiamo solo la VM di Ethereum.

Costi

Subnet 1 nodo	Cloud	On premise
2000 AVAX (78.200\$ con l'AVAX = 39.10\$)	0\$ mantenimento, 250\$ infrastruttura al mese	costi di mantenimento + costi dell'infrastruttura

Subnet Contship - Primo anno

Il design del subnet per il primo anno di test è stato progettato per essere semplice e funzionale. In particolare, il subnet sarà gestito esclusivamente da Contship, che manterrà i diritti amministrativi. La smart contract pool è già stata creata e solo i wallet autorizzati potranno deployare gli smart contract. I requisiti dei nodi rimangono quelli di base raccomandati da Avalanche, in quanto al momento non è necessario un subnet particolarmente potente. I nodi saranno ospitati sui server di Contship e saranno almeno 2 per garantire una certa resilienza. Le gas fees saranno quelle della mainnet di Avalanche, poiché non ci sono bisogni particolari da soddisfare, e il subnet avrà una sola VM, ovvero quella di Ethereum. In sintesi, il design del subnet è piuttosto centralizzato e pensato per una fase di test e per eseguire attività semplici come la certificazione di eventi e i sistemi di pagamento.

Subnet Contship - In Futuro

Per aumentare la fiducia dei partner e garantire una maggiore decentralizzazione del subnet, in futuro sarà importante passare da una configurazione di 2 nodi gestiti da Contship a una configurazione di 3 nodi, di cui 1 gestito da Contship e 2 gestiti da partner diversi. Inoltre, i diritti amministrativi dovrebbero essere distribuiti tra questi 3 attori per garantire che ogni cambiamento sulle regole del subnet venga effettuato in modo unanime e coordinato con i partner.

A seconda dei nuovi progetti, potremmo valutare una modifica delle fees per ottenere un subnet più veloce e in grado di gestire una maggiore computazione. Allo stesso modo, potremmo aumentare i requisiti dei nodi per garantire un subnet robusto e veloce.

Infine, potremmo valutare la creazione di più blockchain all'interno del subnet per avere delle blockchain configurate per usi specifici e ottimizzate per le esigenze dei diversi progetti.

Conclusione

In conclusione, il subnet di Contship è stato progettato per soddisfare le esigenze immediate dell'azienda, con un focus sulla certificazione di eventi e sui sistemi di pagamento. Il design iniziale prevede un subnet centralizzato gestito da Contship, con due nodi sui server dell'azienda e una VM di Ethereum.

Tuttavia, la visione a lungo termine prevede una maggiore decentralizzazione del subnet, con l'aggiunta di nodi gestiti da partner fidati e la distribuzione dei diritti amministrativi tra questi attori. Inoltre, il subnet potrebbe essere configurato per soddisfare le esigenze di nuovi progetti, con la possibilità di modificare le fees e i requisiti dei nodi per ottenere una rete più veloce e robusta.

Infine, la creazione di più blockchain all'interno del subnet potrebbe essere valutata per fornire configurazioni specifiche per diversi casi d'uso. In sintesi, il subnet di Contship è progettato per essere flessibile, scalabile e in grado di soddisfare le esigenze dell'azienda e dei suoi partner a lungo termine.