Guardians

Guardian

Wormhole relies upon a set of distributed nodes which monitor state on several blockchains. In Wormhole these nodes are referred to as Guardians. The current guardian set can be seen in theWormhole Explorer .

It is the guardians role to observe messages and sign the corresponding payloads. Each guardian performs this step in isolation, later combining the resulting signatures with other guardians as a final step. The resulting collection of independent observations form a multisig which represents a proof that a state has been observed and agreed upon by a majority of the wormhole network. These multisigs are referred to as VAAs in Wormhole.

Guardian Network

The Guardian Network is designed to serve as Wormhole's oracle component, and the entire Wormhole ecosystem is founded on its technical underpinnings. It is the most critical element of the Wormhole ecosystem, and represents the single most important component to learn about if you want a deep understanding of Wormhole.

To understand not justhow the Guardian Network works, butwhy it works the way it does, let's first take a step back and go over the key design considerations. To become the best-in-class interoperability platform, there were five critical features Wormhole needed to have:

1. Decentralization
2.
    - Control of the network needs to be distributed amongst many parties.
3. Modularity
4.
    - Disparate parts of the ecosystem such as the oracle, relayer, applications, etc, should be kept as separate and modular as possible so they can be designed, modified and upgraded independently.
5. Chain Agnosticism
6.
    - Wormhole should be able to support not only EVM, but also chains like Solana, Algorand, Cosmos, and even platforms that haven't been created yet. It also should not have any one chain as a single point of failure.
7. Scalability
8.
    - Wormhole should be able to secure a large amount of value immediately and be able to handle the large transaction volume.
9. Upgradeability
10.
    - As the decentralized computing ecosystem evolves, Wormhole will need to be able to change the implementation of its existing modules without breaking integrators.
11.

Next, let's go into how Wormhole achieves these one at a time.

Decentralization

Decentralization is the biggest concern. Previous interoperability solutions have largely been entirely centralized, and even newer solutions utilizing things like adversarial relayers still tend to have single points of failure or collusion thresholds as low as 1 or 2.

When designing a decentralized oracle network, the first option to consider is likely a Proof-of-Stake (PoS) system-but this turns out to be a suboptimal solution. PoS is designed for blockchain consensus in smart-contract enabled environments, so it's less suitable when the network is verifying the output of many blockchains and not supporting its own smart contracts. While it looks appealing from a decentralization perspective, the network security remains unclear, and it can make some other outlined goals more difficult to achieve. Let's explore other options.

The next option would be to rush straight for the finish line and use zero-knowledge proofs to secure the network. This would be a good solution from a decentralization perspective, as it's literally trustless. However, zero-knowledge proofs are still a nascent technology and it's not really feasible to verify them on-chain, especially on chains with limited computational environments. That means a form of multisig will be needed to secure the network.

If we step back and look at the current De-Fi landscape, most of the top blockchains are secured by the same handful of validator companies. Currently, there are a limited number of companies in the world with the skills and capital to run top-notch validator companies.

If a protocol could unite a large number of those validator companies into a purpose-built consensus mechanism that's optimized for chain interoperability, that design would likely be more performant and secure than a network bootstrapped by a tokenomics model. Assuming the validators would be on board, how many could Wormhole realistically utilize?

If Wormhole were to use threshold signatures, the answer would basically be 'as many as are willing to participate.' However, threshold signatures have spotty support across the blockchain world, meaning it would be difficult and expensive to verify the signatures, ultimately limiting scalability and chain agnosticism. Thus, a t-schnorr multisig presents itself as the best option: cheap and well supported, despite the fact that its verification costs increases linearly with the number of signatures included.

All these things considered, 19 seems to be the maximum number and a good tradeoff. If 2/3 of the signatures are needed for consensus, then 13 signatures need to be verified on-chain, which remains reasonable from a gas-cost perspective.

Rather than securing the network with tokenomics, it is better to initially secure the network by involving robust companies which are heavily invested in the success of De-Fi as a whole. The 19 Guardians are not anonymous or small--they are many of the largest and most widely-known validator companies in cryptocurrency. The current list of Guardians can be viewedhere

That's how we end up with the network of 19 Guardians, each with an equal stake and joined in a purpose-built Proof of Authority consensus mechanism. As threshold signatures become better supported, the Guardian set can expand, and once ZKPs are ubiquitous, the Guardian Network will become fully trustless.

With our perspective on Decentralization laid out, the remaining elements fall into place.

Modularity

The Guardian Network is robust and trustworthy by itself, so there's no need for components like the relayer to contribute to the security model. That makes Wormhole able to have simple components that are very good at the one thing they do. That way, Guardians only need to verify on-chain activity and produce VAAs while Relayers only need to interact with blockchains and deliver messages.

The signing scheme of the VAAs can be changed without affecting downstream users, and multiple relay mechanisms can exist independently. xAssets can be implemented purely at the application layer and cross chain applications can use whatever components suits them.

Chain Agnosticism

Today, Wormhole supports a wider range of ecosystems than any other interoperability protocol because it uses simple tech (t-schnorr signatures), an adaptable, heterogeneous relayer model, and a robust validator network.

Wormhole can expand to new ecosystems as quickly as a Core Contract can be developed for the smart contract runtime. Relayers don't need to be factored into the security model--they just need to be able to upload messages to the blockchain. The Guardians are able to observe every transaction on every chain, without taking shortcuts.

Scalability

Wormhole scales well, as demonstrated by its ability to handle huge TVL and transaction volume--even during tumultuous events.

The requirements for running a Guardian are relatively heavy, as they need to run a full node for every single blockchain in the ecosystem. This is another reason why a limited number of robust validator companies are beneficial for this design.

However, once all the full nodes are running, the actual computation and network overheads of the Guardian Network become lightweight. The performance of the blockchains themselves tends to be the bottleneck in Wormhole, rather than anything happening inside the Guardian Network.

Upgradeability

Over time, the Guardian Set can be expanded beyond 19 with the use of threshold signatures. A variety of relaying models will emerge, each with their own strengths and weaknesses. ZKPs can be used on chains where they are well supported. The cross chain application ecosystem will grow, and cross chain applications will become increasingly intermingled with each other. There are very few APIs in Wormhole, and most items are implementation details from the perspective of an integrator. This creates a clear pathway towards a fully trustlessness interoperability layer which spans the entirety of decentralized computing.

Last updated8 months ago