General Flow of Events processed by Axelar Network

This section will provide a high-level understanding of the end-to-end flow by which Axelar network processes cross-chain messages. It includes:

- How messages are picked up by Axelar on external connected chains.
- How messages are delivered to the destination chains.
- How Axelar validators and cryptography work to securely route and transfer messages.
- An introduction to the underlying relayer services that provide a smooth user experience.

Background

Before understanding the flow of events processed by Axelar, it's important to understand Axelar's core features and tech stack.

Axelar is a proof-of-stake blockchain network built using Cosmos SDK. It has its own permissionless set of validators, which constantly produce blocks containing transactions. This validator set is used by the network to securely process cross-chain messages through voting on the legitimacy of all messages. To understand more about the setup, take a look at the Axelar Tech Stack Diagram.

Axelar delivers secure cross-chain communication between separate blockchain networks. This could involve making cross-chain token transfers, calling a smart contract on another chain or passing general, arbitrary messages across chains. While each feature's exact processing flow varies, it can be simplified down to a cross-chain "message" that the network processes.

Cross-chain General Message Passing requests

Axelar enables dApp builders to add cross-chain functionality to their applications, through a suite of APIs. These APIs allow a dApp user to send a message containing arbitrary data cross-chain, from the source chain to the destination chain. Thus, a dApp may call a smart contract on the destination chain, and attach general data information to be used as input for the smart contract call. To do this, the dApp interacts with the Axelar Gateway, an Axelar-operated installation on the source chain, which contains the functions and logic needed for initiating and processing any cross-chain messages.

Gateways

Once a cross-chain message is initiated by a dApp user, its first stop is to interact with an Axelar Gateway. On each chain connected to Axelar network, a Gateway is deployed. On EVM chains, it is a smart contract address. On Cosmos and other non-EVM chains, it is an application with logic and the ability to communicate with Axelar network. This Gateway is used to receive messages from a connected dApp and send them into the Axelar network for routing to any connected chain.

The Gateway is controlled by a key, which is held jointly by all Axelar validators. This is accomplished through a multiparty cryptography scheme, where the key is divided into many pieces, called key shares. Each validator holds many key shares, and the amount of shares is dictated by the amount AXL tokens staked with the validator.

Simply put, Gateways enable communication between Axelar and its connected chains, and they have two main functions. On the source chain where the message originates, the Gateway enables initiating cross-chain message requests. On the destination chain where the message is received, the Gateway enables the message to be executed and completes the cross-chain protocol.

Message processing and relayers

Axelar network must be aware of events happening and cross-chain communication requests being submitted on external supported chains. This function is handled by relayers. Once a Gateway receives a message, it will generate an event. Axelar runs relayer services which observe all connected chains, and these relayers will pick up the event and submit it to the Axelar network for processing. These relayer services are a free, operational convenience Axelar provides, and can be run by anyone who wishes to create and use their own relayer service instead.

Message validation

Since anyone can run their own relayer services and submit events to Axelar network (see the previous section on message processing and relayers), all events received by Axelar must first go through validation, to ensure the submitted events are correct and trustworthy. In many blockchain ecosystems, trust comes from the network validators, who are incentivized to behave honestly through consensus protocols, such as proof-of-stake. These validators are already incentivized by Axelar network's proof-of-stake consensus model to create blocks and approve transactions, so it is a natural extension to have the validators participate in consensus and vote on the truthfulness of cross-chain events being submitted.

Once the event sent by an external chain is received by Axelar, a poll is generated and Axelar validators vote on the poll. In order to vote on the legitimacy of a message, Axelar validators need to verify the submitted event. Each validator runs their own node for the source chain where the incoming event originated, and by querying the RPC endpoint of their source-chain node, validators can check if the submitted event was observed on their source-chain node. If the event is found, the validator votes in the poll to approve the message as legitimate.

As Axelar grows and connects more and more chains, it becomes increasingly restrictive to require every Axelar validator to run a node for every chain supported by the network. Instead, Axelar validators are incentivized to run nodes for as many supported chains as possible, through increased staking rewards, based on the number of chains they support. By having Axelar's decentralized, proof-of-stake validator network vote on the legitimacy of each cross-chain message, the network is able to process and route them to the destination chains securely. Only verified events are translated into actions on the destination chain, and events not verified by Axelar validators are discarded. Afterwards, the event is validated by the Axelar network, processed by the network consensus protocol and recorded in a block. The cross-chain message is routed into a queue of messages for the destination chain, and ready to be sent to the destination-chain Gateway.

By using proof-of-stake consensus to secure cross-chain messages, additional assumptions in security are avoided, as most of the Axelar-connected networks use the same proof-of-stake consensus model to validate their transactions.

Submitting a message to the destination chain

After the message is approved by Axelar network validators, it needs to be authorized. The Gateway can only allow actions on the external chain if the number of validators holding key shares who authorize the action reaches a set threshold.

Once a message is authorized, anyone can take this signed message and submit it to the destination chain for processing. Similar to the relayer services described earlier, which observe external chain events and bring them into Axelar network, another set of relayer services monitors outgoing transaction queues containing cross-chain messages that are already approved and signed, and periodically submits these transactions to external chains. Just as for incoming messages, these outbound Axelar relayer services are a free, operational convenience Axelar provides, and can be run by anyone who wishes to create and use their own relayer service instead.

After a relayer sends it out, a Gateway on the destination chain receives the approved message, where its payload is marked as approved by Axelar. The Gateway stores the contract call approval along with a hash of the payload, representing the legitimacy of this cross-chain message. The approved payload can now be executed by anyone at any time.

Gas and executor services

Once the cross-chain payload is marked as approved by the Axelar Gateway contract, it needs to be sent to the destination dApp or contract and executed, in order to perform a function call or token transfer. This execution requires tokens of the destination chain to cover gas fee costs. The gas could be paid in a few different ways. One option is to have the Web3 user pay the fee manually for their own cross-chain message. This makes for an inconvenient user experience, as it requires the user to acquire destination-chain and source-chain tokens. Axelar offers Gas and Executor services to help make this easier.

Axelar creates and deploys a smart contract called the Gas Receiver to all connected EVM chains. (General Message Passing for Cosmos chains is currently under development and this may enable Gas-Receiver-like services to be deployed there, as well.) This Gas Receiver allows the user to pay for all transaction fees required throughout the entire journey of their cross-chain message in a one-time payment on the source chain, in the source chain's native token. The Gas Receiver estimates the total gas cost required across source chain, Axelar network and destination chain, and converts source-chain tokens (or any token Axelar supports) into AXL, destination-chain tokens and any other required currencies. For more details on this process, see the diagram below and read thisblog post providing a high-level description of the AXL token's uses and benefits.

Monitoring and recovery

Once the message is executed on the destination chain, its cross-chain journey is complete. However, there may be factors that prevent this completion at various stages of the process. These might include a gas limit set too low, or contract logic that is incompatible with a function called on the destination chain. In order to mitigate against these obstructions, Axelar provides various tools fordebugging ,monitoring and recovery .

For example, while a cross-chain message is being processed by Axelar, a user or developer can follow and observe its journey. The Axelarscan explorer has a General Message Passingtracking tool built for this purpose. The user can enter their source-chain sender address or transaction hash into the Axelarscan search bar to find their cross-chain message. This will bring up a detailed page containing information about the message, and its current status. If there are any issues with the transaction, this page will also suggest possible fixes. Visit thislink for more information on how to use this monitoring and recovery tool.