Relay

What is Relaying?

Standard Transactions

In a standard Ethereum transaction, an ethereum user signs and sends the transaction themselves. This user controls the private key to an externally owned account (EOA) which they can use to sign a transaction and prove they have the right to spend the balance associated with that account address. For each transaction a user sends, there is an associated transaction fee, known as gas. Since Ethereum executes computation, each unit of computation has an associated gas cost, which deters malicious actors from overloading the network by requiring them to pay heavily for a potential attack. This is excellent news for Ethereum's security and helps keep the network consistent under load, but it comes at a hidden cost for onboarding new users.

Onboarding Issues

How does a new user start interacting with exciting on-chain applications like DeFi, NFTs, or gaming? They will always need the native token to pay for gas on every network, even if the network has very cheap gas fees like Polygon. This requires the user to open an account at a centralised exchange, go through KYC, and buy crypto using fiat. This can be quite a process, even for the most skilled of degens out there, and it can deter new users from being onboarded to a dApp by increasing the latency between their initial excitement and the time it takes to actually get started. This is where relaying comes in! A relayer can help solve these issues by sending a transaction on behalf of the user.

What is a Relayer?

We allow the user to send a transaction without a native token balance (it turns out relayers can be super nifty in loads of ways, for example, allowing a user who wants to swap a token to pay for the gas using the token being swapped!). Ideally, we would also like to still utilise the excellent security of a user signature, but for the transaction to be sent by a different EOA, one controlled by a relayer, who abstracts gas payment away from the user. This is a very import context shift to understand. We have shifted from a user signing and sending a transaction themselves, to a user signing a standardised message and passing that on to a relayer. This relayer will, first, verify the user's signature for security, and then pass their message along on-chain. Gelato Relay does exactly this by taking a user's message off-chain and subsequently building a meta-transaction which is executed on chain.

Meta Transactions and EIP-712

A meta transaction is a regular ethereum transaction which contains the actual message to be delivered on-chain to a target contract within itself, hence the term meta. The outer transaction helps facilitate the first on-chain call which is sent by a relayer. The call is forwarded to the target contract using an intermediate smart contract (Gelato Relay), which in turn forwards the call using the inner transaction to deliver the relayed message.

To achieve gasless transactions securely, Gelato Relay makes use of the EIP-712 standard. EIP-712 allows for a standardised way to sign and hash typed structured data. This means the user can sign a message using their wallet without incurring a gas cost or interacting with the chain at all, and this signature can be verified on-chain, by the relayer, facilitating a gasless transaction with security built in. This message will include important information such as the transaction signer address, the target contract address, and the calldata payload used to target a specific function.