



# Meta Transactions

powered by OpenZeppelin Defender

**[zpl.in/defender-workshop](https://zpl.in/defender-workshop)**

**Santiago Palladino**

[santiago@openzeppelin.com](mailto:santiago@openzeppelin.com)

 [@smpalladino](https://twitter.com/smpalladino)



Our mission is to protect  
the open economy

OpenZeppelin is a software company that  
provides **security audits** and **products** for  
decentralized systems.

Projects from any size -from new startups to  
established organizations- trust OpenZeppelin to  
build, inspect and connect to the open economy.



# Security, Reliability and Risk Management

OpenZeppelin provides a complete suite of **security and reliability products** to build, manage, and inspect all aspects of software development and operations for Ethereum projects.



# Getting started

Let's talk user onboarding

*“**40%** of people abandon a website that takes more than **3 seconds** to load”*

# The user onboarding journey

- Navigate to the app
- Go to the browser store
- Install a new extension
- Accept terms & conditions
- Write down 12 words
- Re-enter them for verification
- Setup a passphrase
- Copy their new address
- Search for an exchange
- Create an account
- Choose a different password
- Accept terms & conditions
- Verify email account
- Enter personal data
- Upload proof of identity
- Send a wire transfer
- Await for funds to be accredited
- Purchase ether needed for transaction
- Take their ether out of the exchange
- Start using the app

## Create an account

## Purchase funds

Users need funds to interact with a contract  
to pay for gas fees

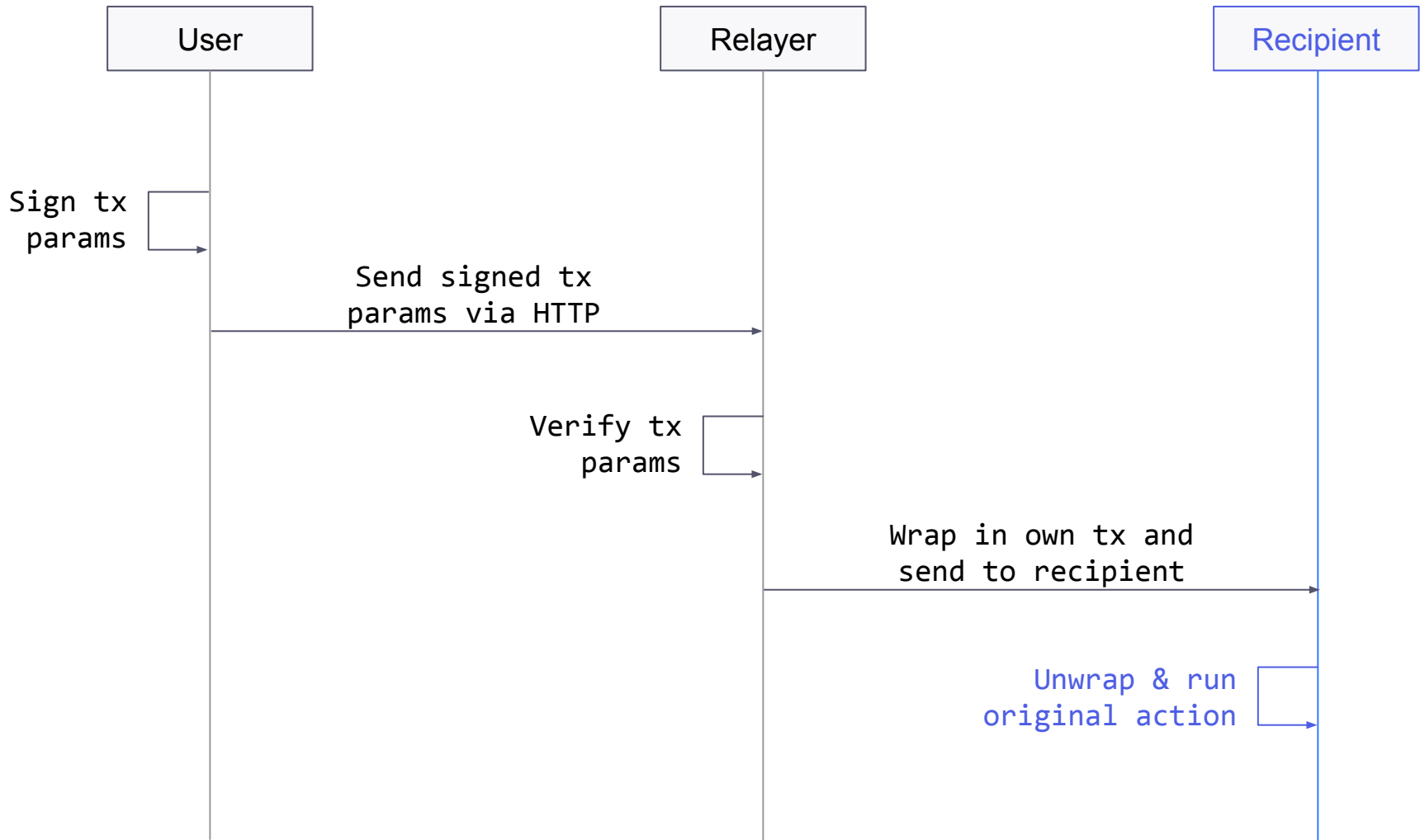
# Enter meta-transactions

Decoupling the *sender* from the *payer* of the fees



## How do meta-transactions work?

1. **User** signs a request and sends it to a **relayer**
2. **Relayer** wraps it in a tx and sends it to a **contract**
3. **Contract** unwraps the tx and executes on behalf of user



## Benefits of meta-txs

- User doesn't need funds to start using app
- Especially valuable for sidechains
- Enables usage of app-generated wallets

# Implementation

Using OpenZeppelin Contracts & Defender

User

Relayer

Recipient

**MetaMask**

EIP712 signTypedData

**OpenZeppelin Defender**

Autotasks webhooks

Relayer

**OpenZeppelin Contracts**

MinimalForwarder

EIP2771 Recipient

Send signed tx  
params via HTTP

Verify  
params

Wrap in own tx and  
send to recipient

Unwrap & run  
original action

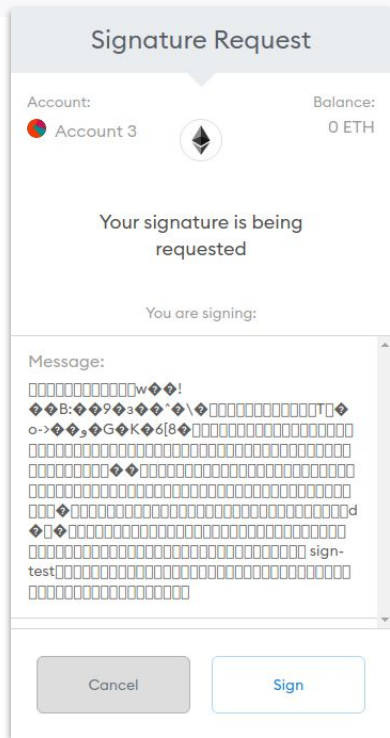
Sign  
params

## User signature

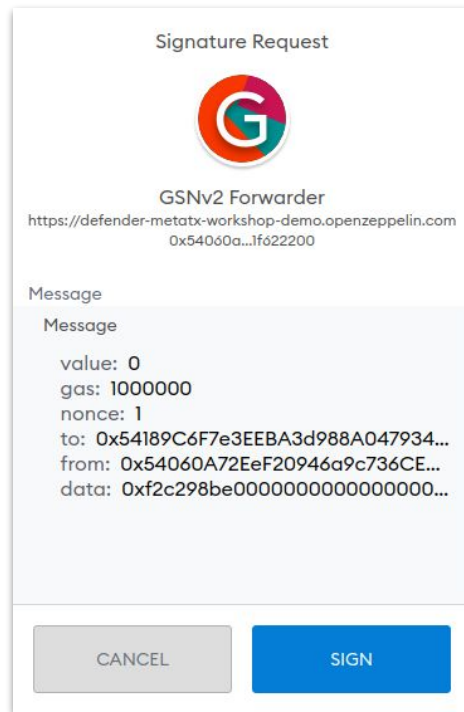
recipient  
data  
nonce  
domain\_sep

application contract address  
function and args to execute  
prevents replay attacks  
prevents replay attacks cross-forwarders

## Signing plain messages vs typed data



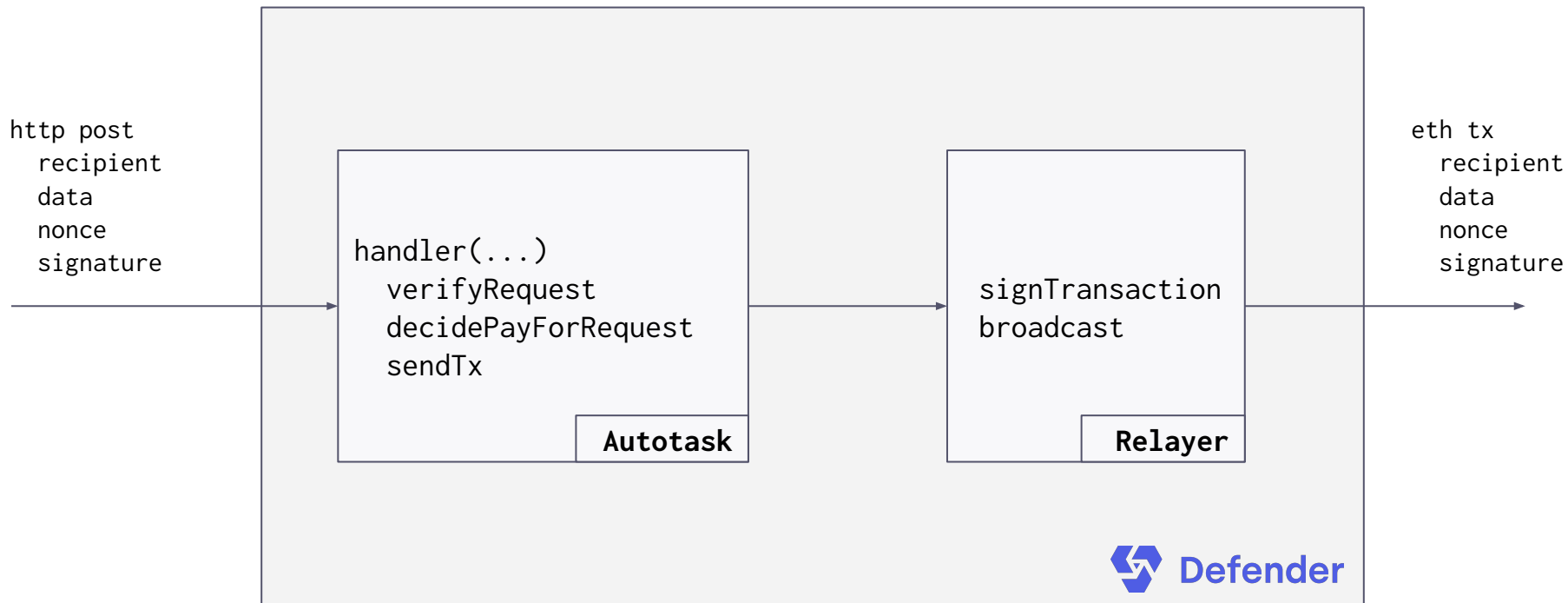
## Sign Message



## Sign Typed Data (EIP 712)

<https://openzeppelin.com>

# Relay

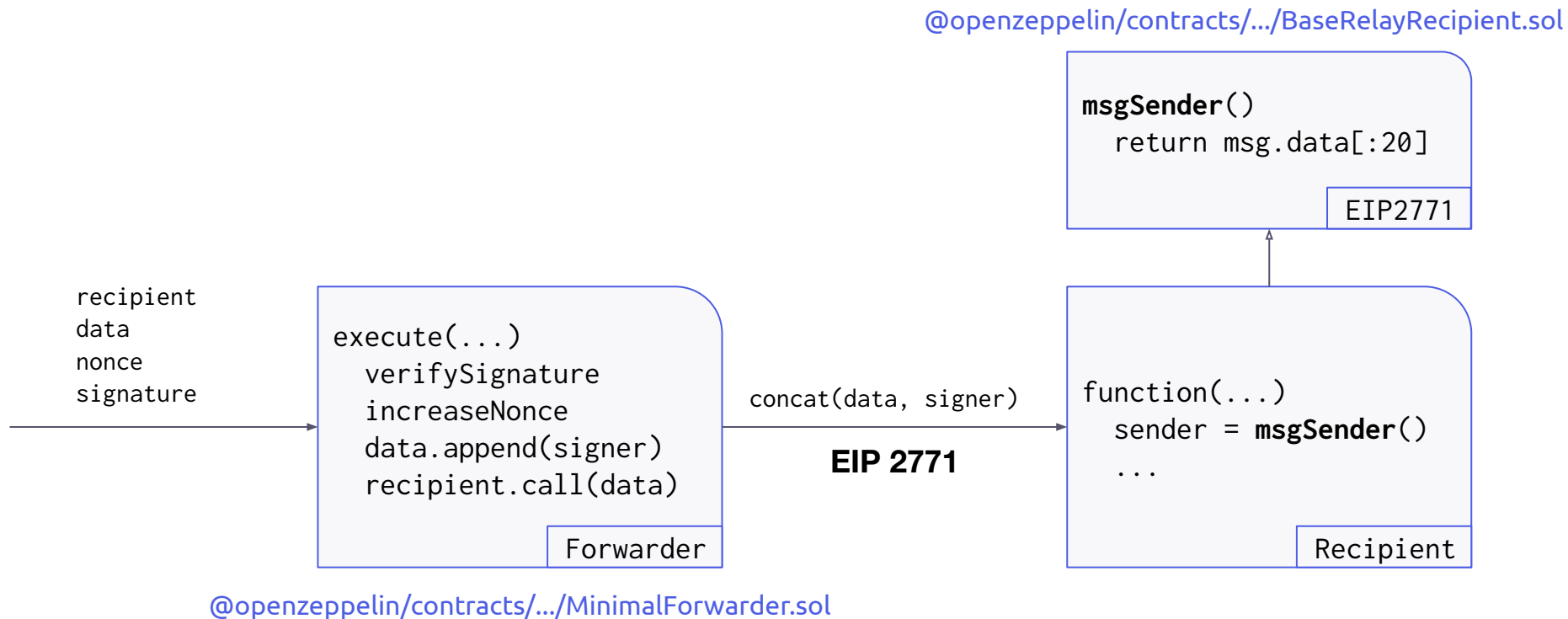




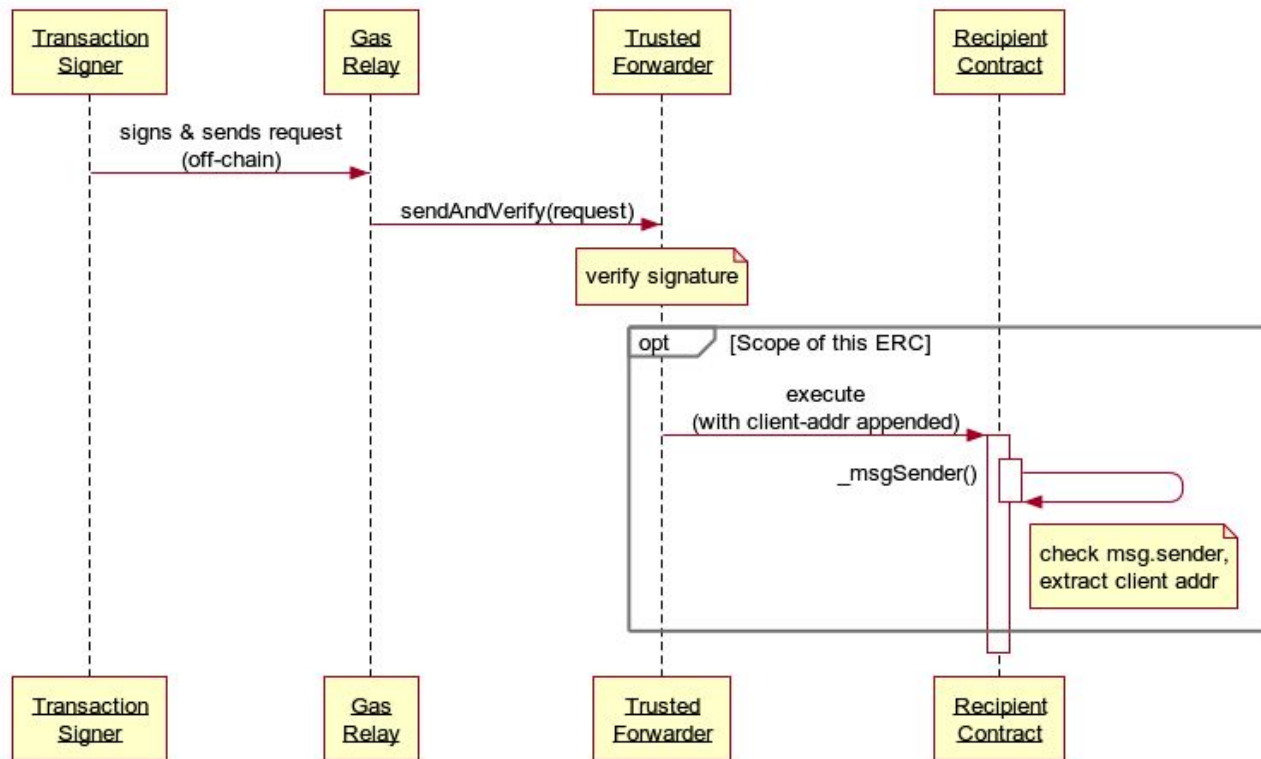
## Why use a Defender Relayer?

- Securely stores the signing key
- Manages nonces
- Determines best gas price
- Automatically resubmits txs
- High availability through multiple providers

# Contracts



# EIP2771



# Demo Time

Hands-on with the code

## Recap

- **User** signs meta-tx request and sends to webhook
- **Autotask** receives and validates request
- **Relayer** wraps request in a tx, signs it, and sends it
- **Forwarder** contract validates signature and forwards call
- **Registry** contract processes call as if sent by the signer

**defender.openzeppelin.com**

**docs.openzeppelin.com**


**forum.openzeppelin.com**




# Thank you!

## Learn more

[openzeppelin.com/defender](https://openzeppelin.com/defender)  
[forum.openzeppelin.com](https://forum.openzeppelin.com)  
[docs.openzeppelin.com](https://docs.openzeppelin.com)



## Contact

 @smpalladino  
[santiago@openzeppelin.com](mailto:santiago@openzeppelin.com)

