Usability changes in a post-EOA world

Cindy Chau
Founding Designer, Phylax

Greg Markou
Co-Founder, Sprinter & ChainSafe

Dapp UX today is constrained by the concept of what an account is.

Dapp UX today is defined by single signature signing

Non-upgradeable

Entire account gets connected

Must have native token to send transactions

Seed phrase storage

No native batching

No transaction rules

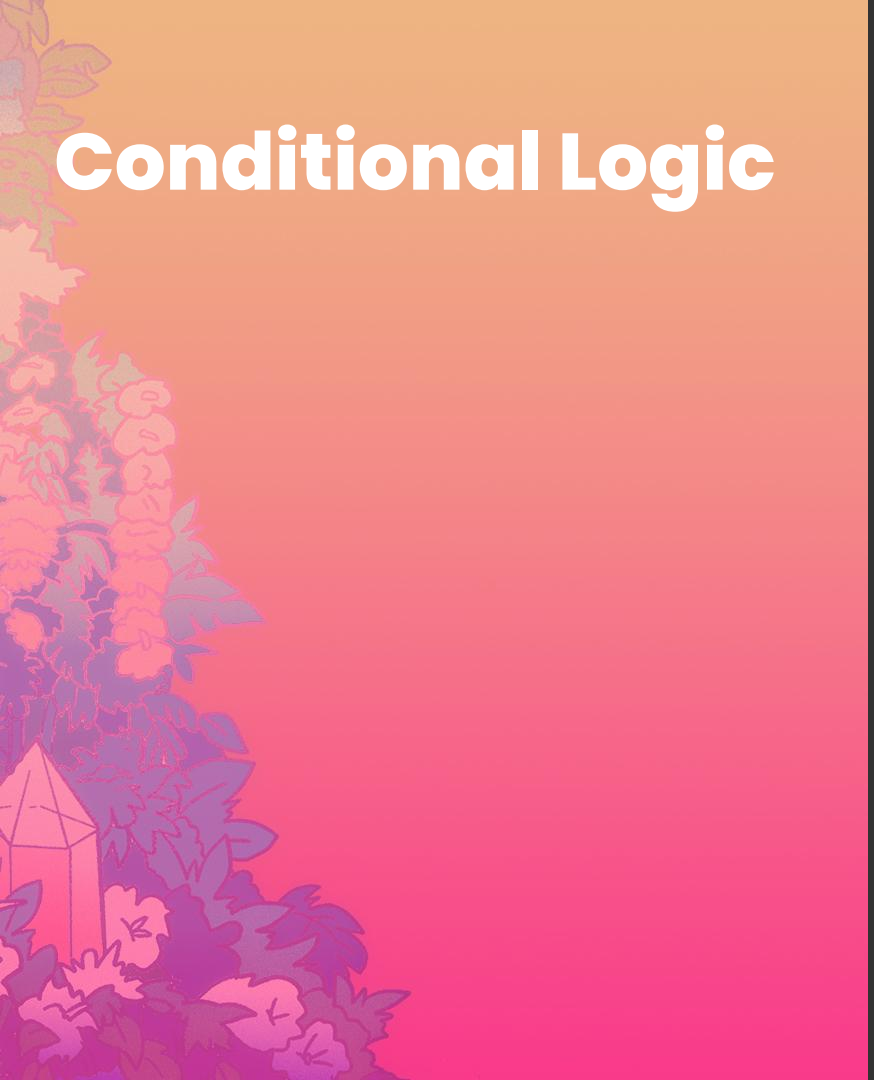
Common pitfalls that account abstraction can solve for

Enhanced security, smooth onboarding, convergent applications

“Appification” of dapps

What greater programmability and security on smart contract wallets can look like

Conditional Logic



← Back to portfolio selection

Add Portfolio Security

Add an escape hatch, define disaster recovery

Portfolio Selection | Security Settings | Review & Pay

Add recovery strategies

Unwind all positions
Auto-remove exposure if liquidity pool drains more thna\

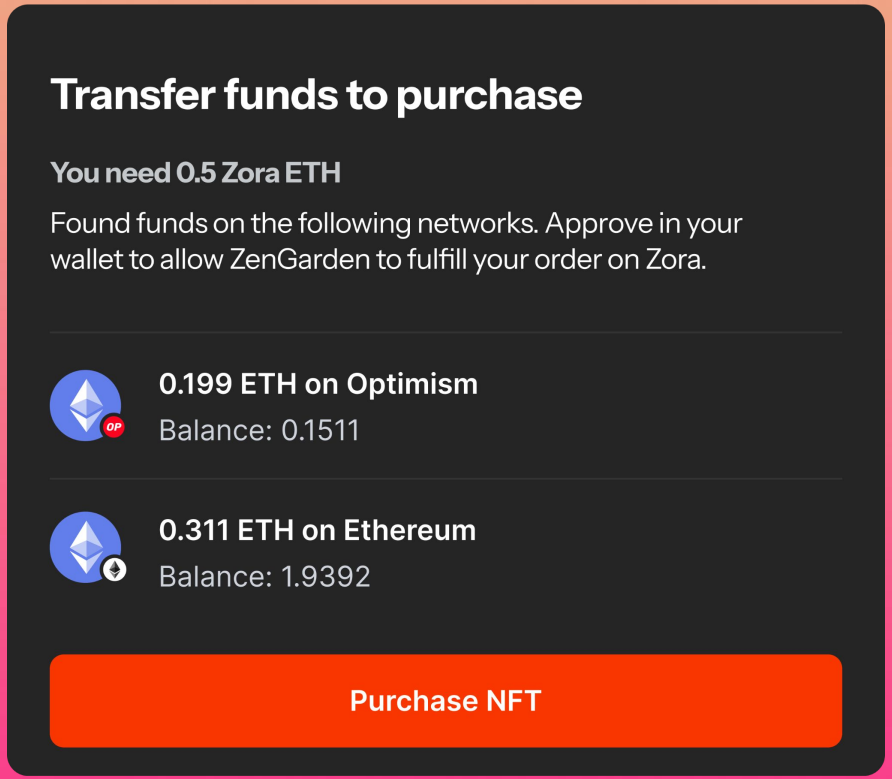
Do not unwind, revoke all permissions
Keep positions but remove dapp permissions

Watch on

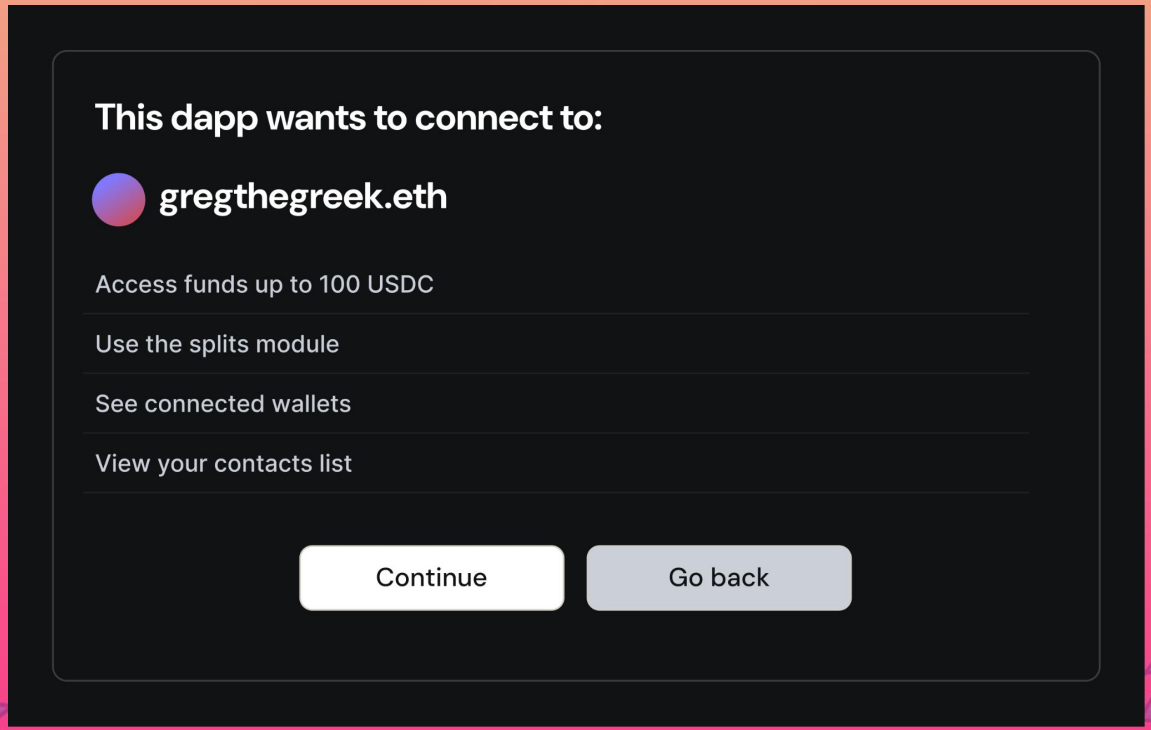
Optimism	Base	Mainnet	Based
Free	Free	Free	Coming soon Free

$25/mo
Estimated Payment

Cross-chain calls & aggregation



Modular authentication to allow user to define the dapp's level of access



Better blending of offchain and onchain and cross-chain components

New solutions = new friction

UX Enhancement	New user friction & risks
Fewer approvals	Account and funds need to migrate; new address!
Enhanced programmability	EOA is still initial signer for it
Gas abstraction	4337-compatibility before 7702 = not gas optimized
Account Recovery	Unfamiliar user patterns
Modular Authentication or Session Keys	Increased complexity for implementing account methods; you've issued a new wallet for the user

How do you build the best possible ground-zero dapp experience using smart accounts?

Surely it differs by use case...

Doesn’t have to.

The improvements that will get us there

CAIP-275

ERC-4337

EIP-7702

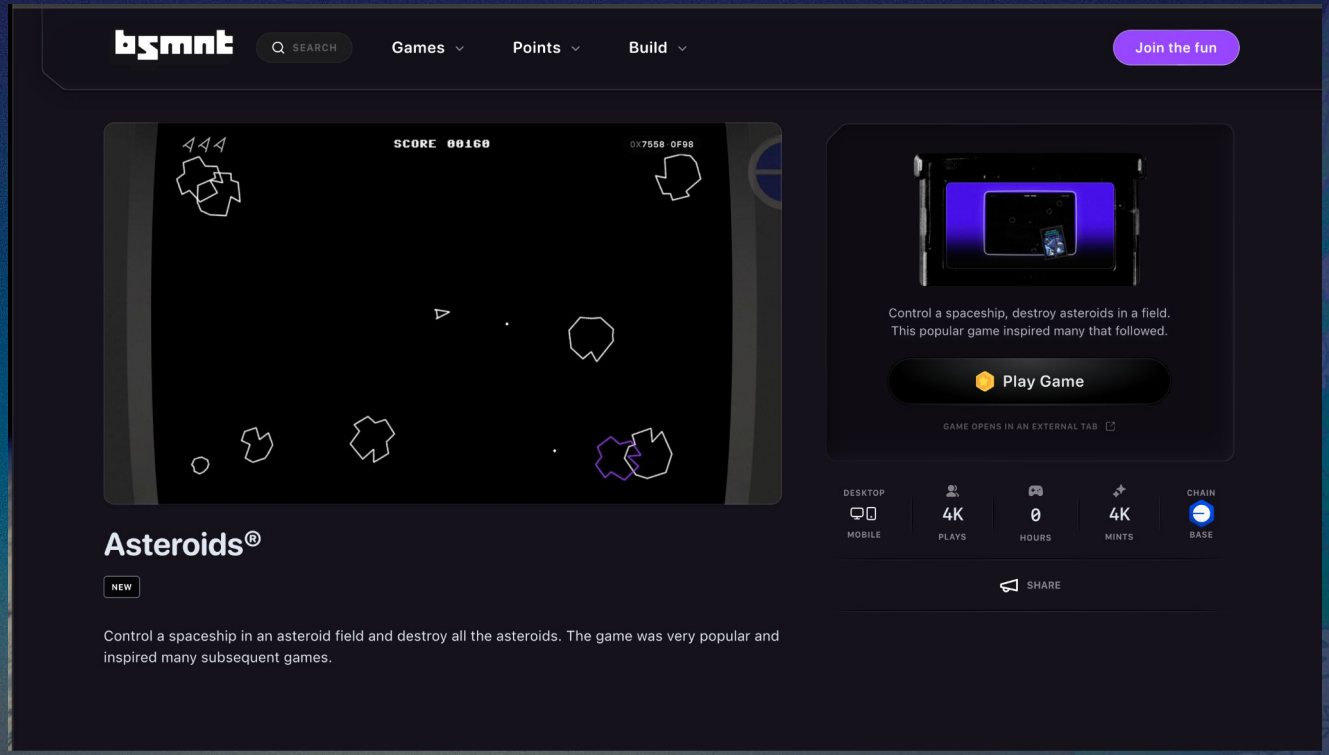
EIP-7755

EIP-7715

EIP-7676

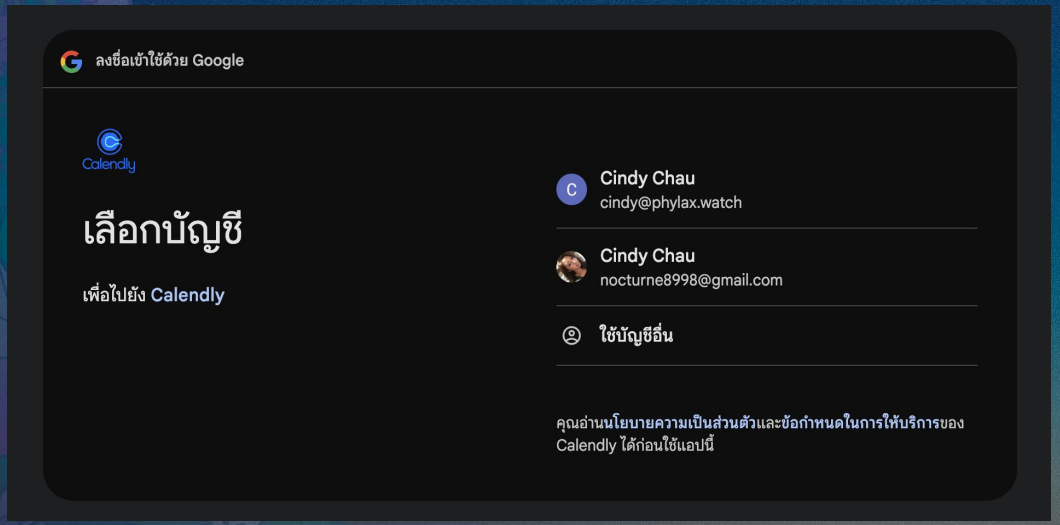
RIP-7575

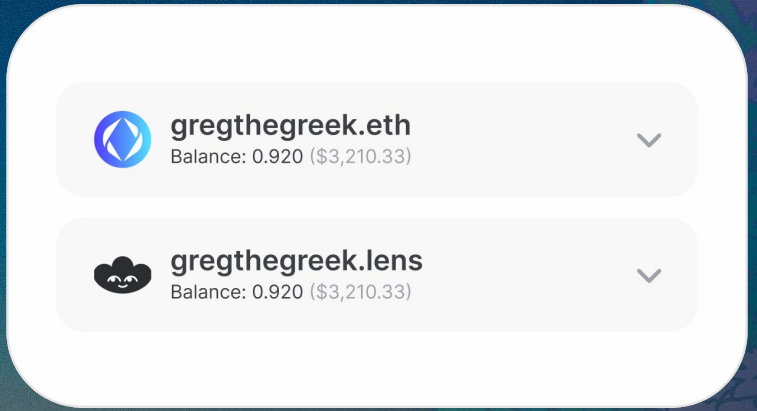
1. Let the user browse the app without logging in first with a silent embedded wallet.



2. Deliver an OAuth experience to let the user sign in with the account they want.

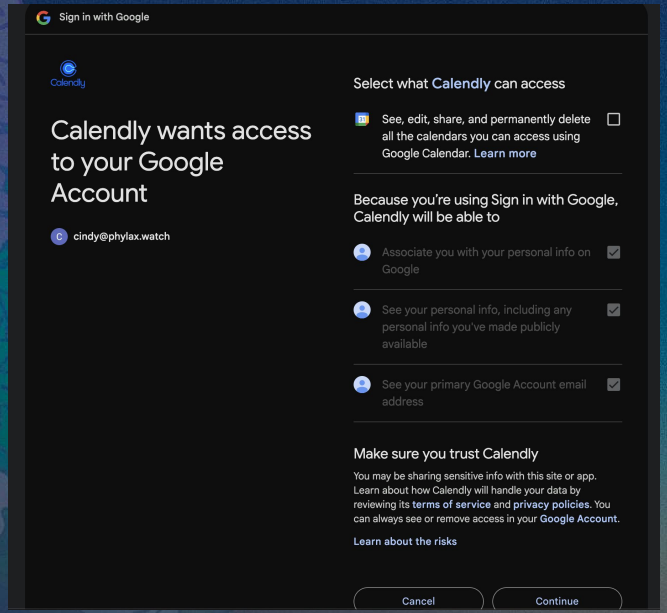
CAIP-275: Discovery
EIP-7555: OAuth

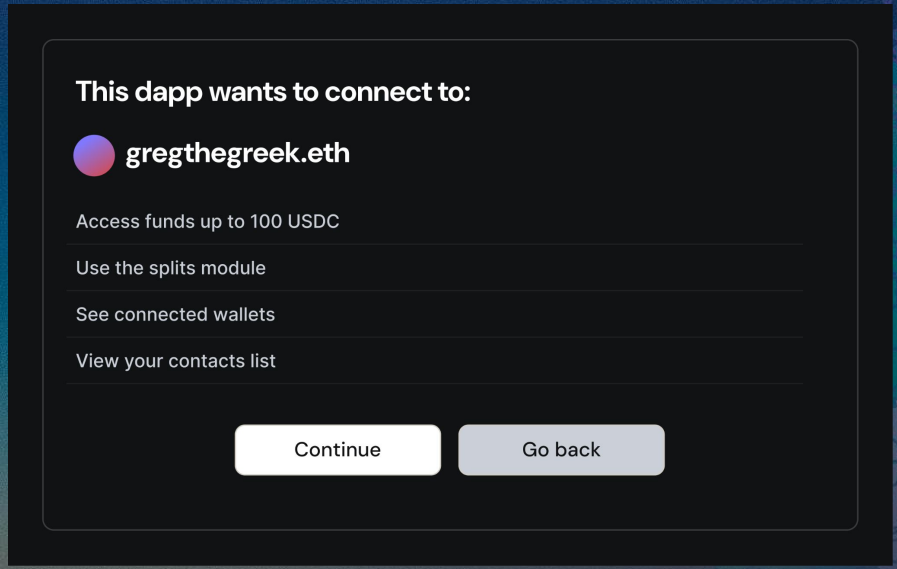




3. Let the user decide what the app gets to do with it

EIP-7715: Permissions Controls
Authorize a spend limit, session key creation

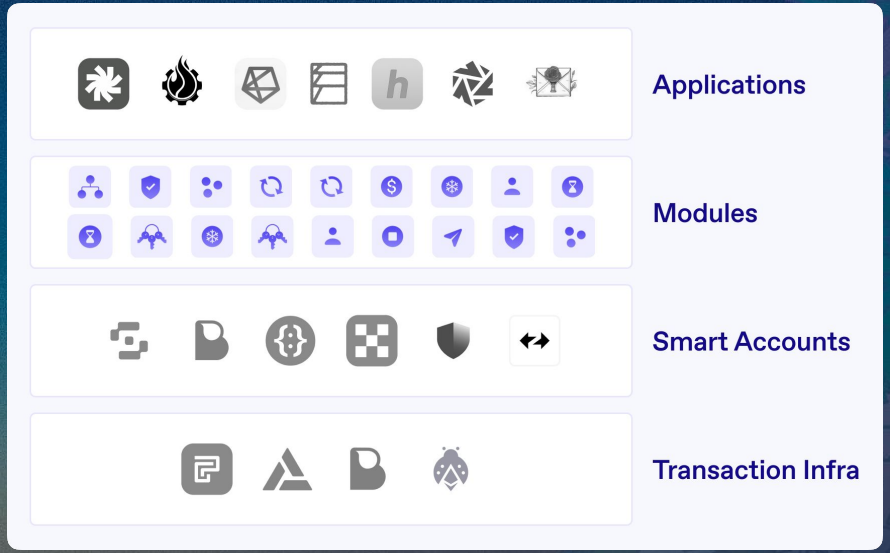




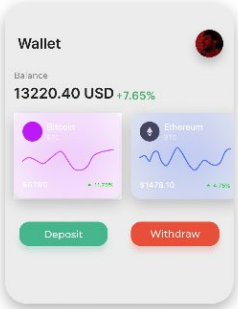
4. Greater programmability and personalized app experiences

EIP-7579: Minimal Modular Smart Accounts

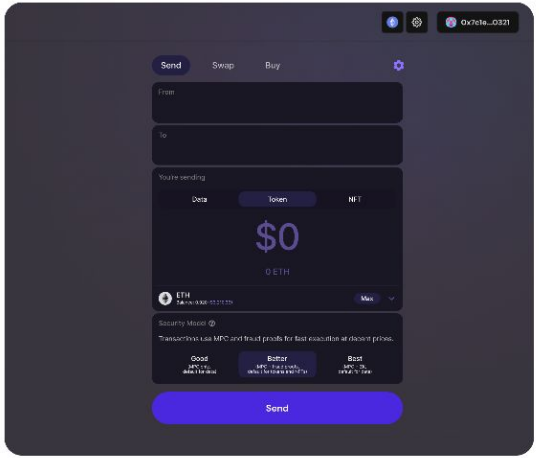
- Fewer approvals, more functionality
- Bundle transactions and complex logic



The Endgame

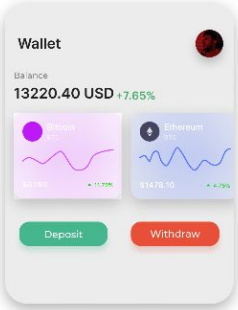


Wallet



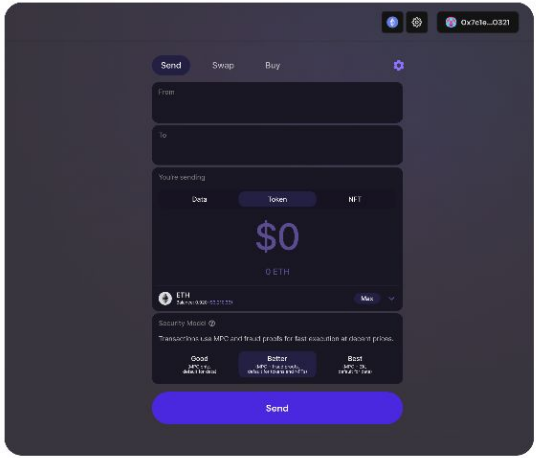
Dapp

The Endgame



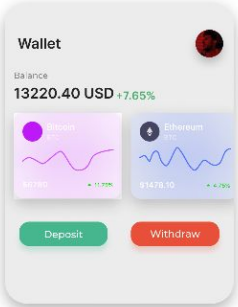
Wallet

ERC-4337



Dapp

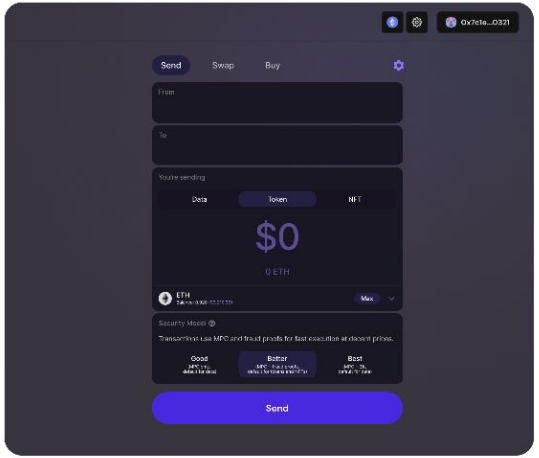
The Endgame



Wallet

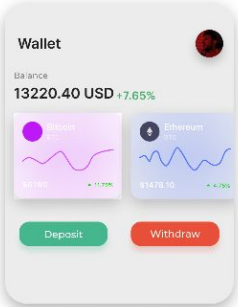
ERC-4337

Passkey



Dapp

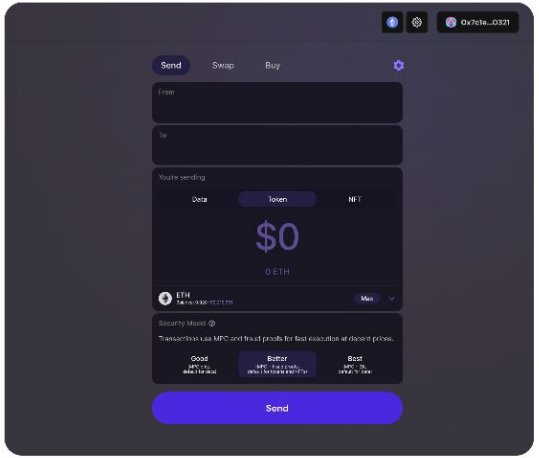
The Endgame



Wallet

ERC-4337

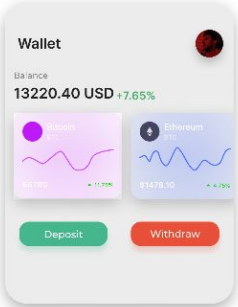
Passkey



Dapp

Embedded Wallet

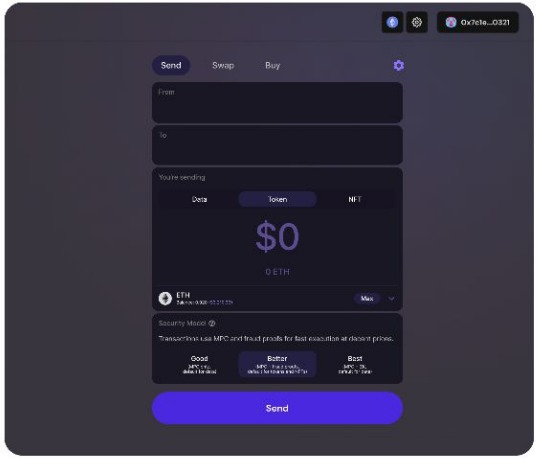
The Endgame



Wallet

ERC-4337

Passkey



Dapp

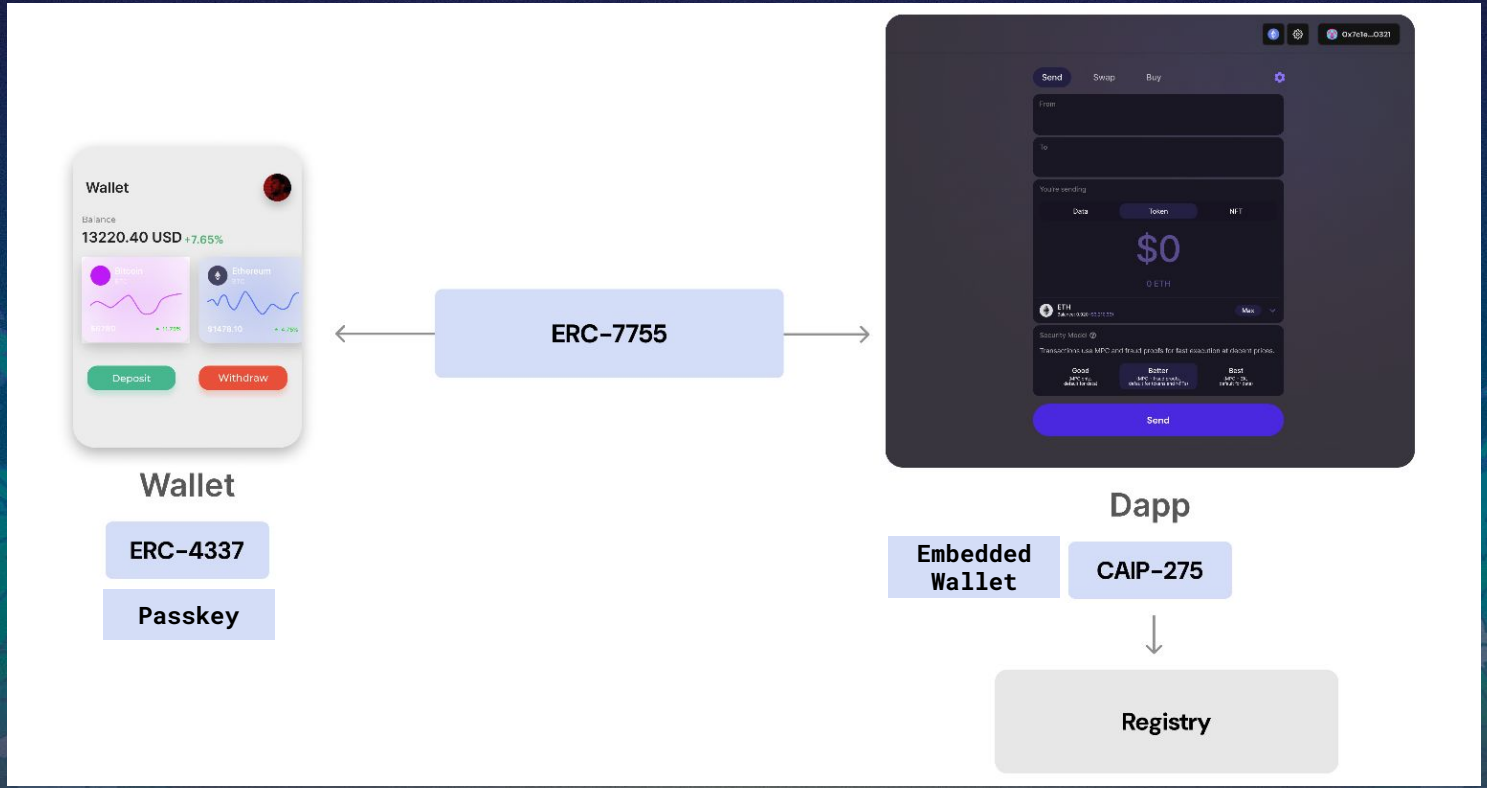
Embedded Wallet

CAIP-275

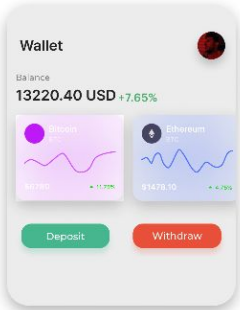
↓

Registry

The Endgame


Wallet
Balance
13220.40 USD +7.65%
Deposit
Withdraw
ERC-7755
Send
Swap
Buy
From
To
You're sending
Data
Token
NFT
$0
0 ETH
ETH
Max
Security Model
Good
Better
Best
Send
Wallet
ERC-4337
Passkey
Dapp
Embedded Wallet
CAIP-275
Registry

The Endgame



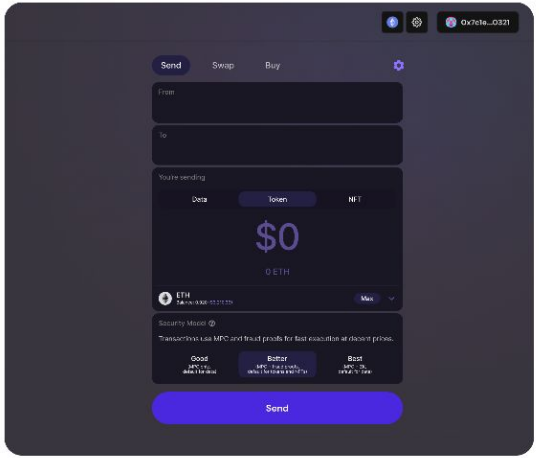
Wallet

ERC-4337

Passkey

ERC-7715

ERC-7755



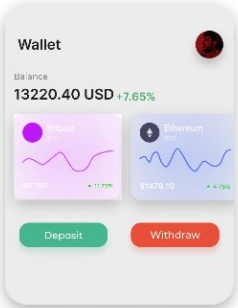
Dapp

Embedded Wallet

CAIP-275

Registry

The Endgame



Wallet

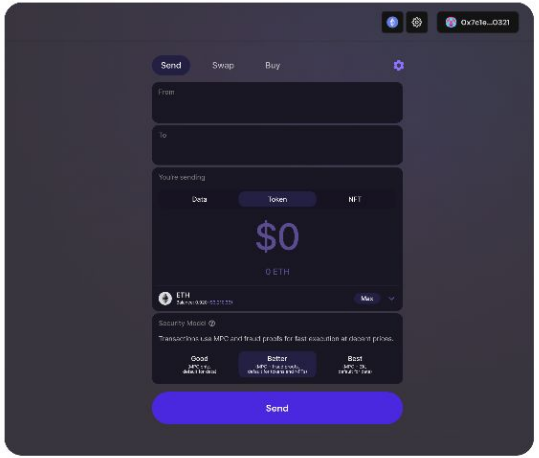
ERC-4337

Passkey

ERC-7579

ERC-7715

ERC-7755



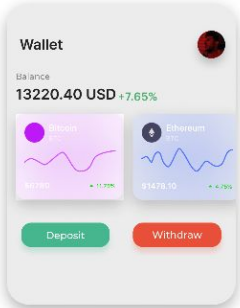
Dapp

Embedded Wallet

CAIP-275

Registry

The Endgame



Wallet

ERC-4337

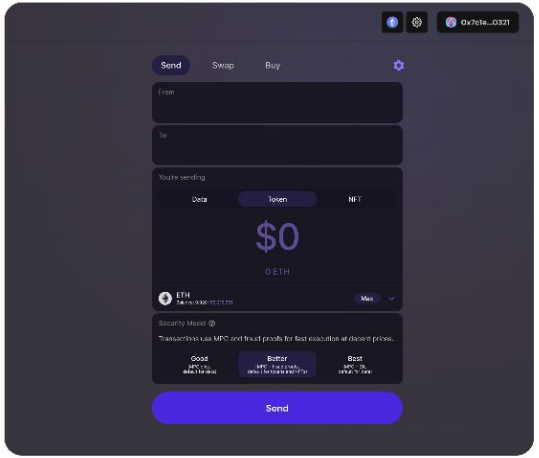
Passkey

ERC-7579

ERC-7715

ERC-7811

ERC-7755



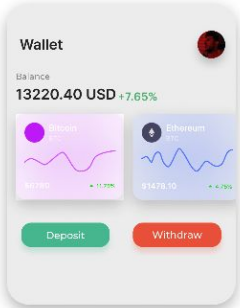
Dapp

Embedded Wallet

CAIP-275

Registry

The Endgame



Wallet

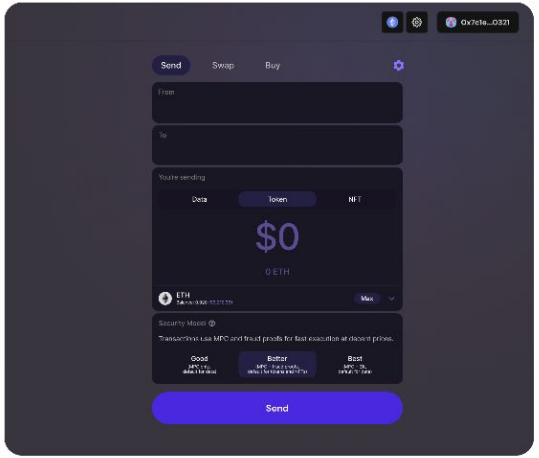
ERC-4337

Passkey

ERC-7579 | ERC-xxx

ERC-7715 | ERC-7811

ERC-7755



Dapp

Embedded Wallet

CAIP-275

Registry

The time for greater ground-zero dapp experience is here.

Let’s keep talking!

Cindy Chau
@ascii_g1rl

Greg Markou
@GregTheGreek