

Defragmenting Ethereum

Karl Floersch
Optimism



OPTIMISM

Defragmenting Ethereum



...picking up where we left off



What is Unipig?

A testnet demo of Uniswap running on optimistic rollup to demonstrate the UX improvements that are possible with layer 2.

Ethereum can scale today.



What is Unipig?

A testnet demo of Uniswap running on optimistic rollups to demonstrate the UX improvements that are possible at layer 2.

Ethereum can scale today.

Chapter 7: *We're in this Together*



What is Unipig?

A testnet demo of Uniswap running on optimistic rollups to demonstrate the UX improvements that are possible at layer 2.

Ethereum can scale today.

Chapter 7: We're in this Together



Chapter 8: Forming the Superstructure



Devcon 5

Devcon 6

Devcon 7

R&D

Optimistic rollups introduced

Proliferation

Superchain & OP Stack introduced
[come deploy!]

Interoperation (dev tools)

Interop tooling low hanging fruit

**Nothing is that new...
Nothing is that unexpected...**

The tech just slowly marches forward



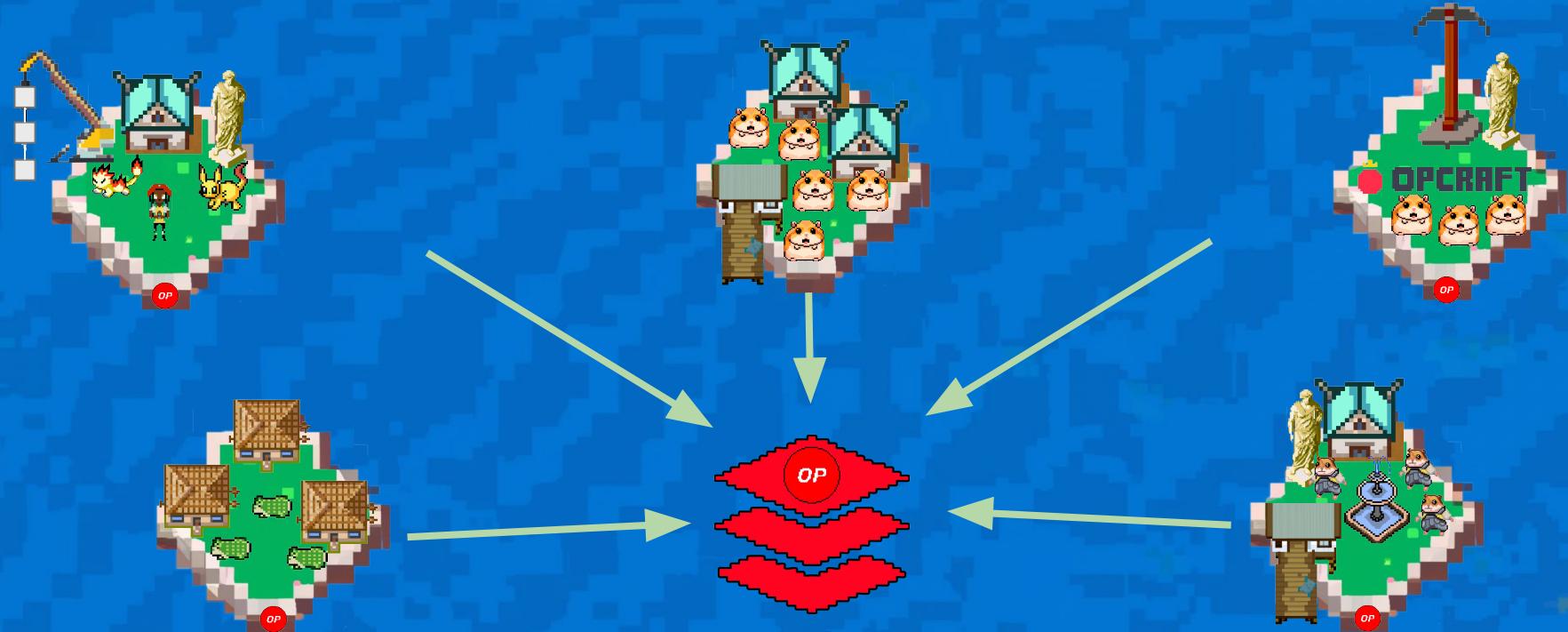




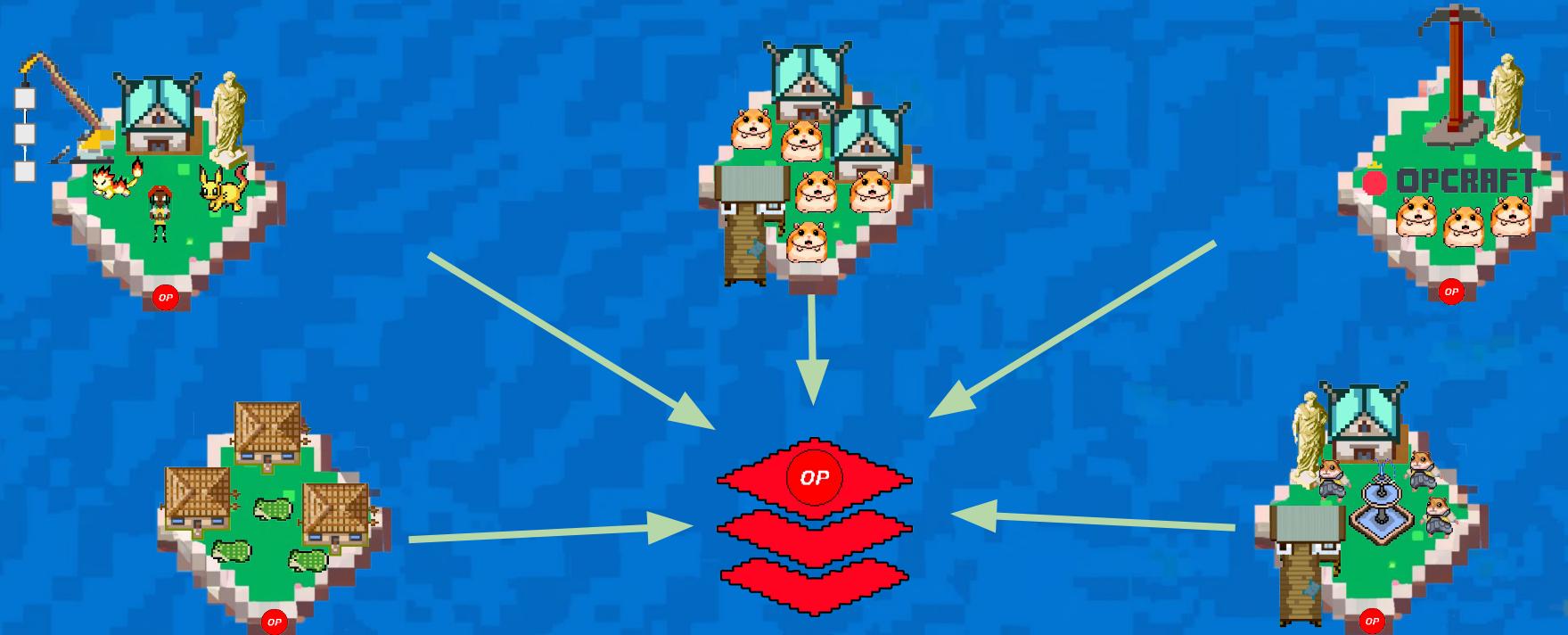
Chapter 8: Forming the Superstructure



Chapter 8: Forming the Superstructure



Chapter 8: Forming the Superstructure



...and cooperative op-chains will begin to form something new

Chapter 8: Forming the Superstructure



...and cooperative op-chains will begin to form something **new**

Chapter 8: Forming the Superstructure



...and cooperative op-chains will begin to form something **new**

Chapter 8: Forming the Superstructure



...and cooperative op-chains will begin to form something **new**

Chapter 8: Forming the Superstructure



Chapter 8: Forming the Superstructure



NOT multi-chain

NOT mono-chain





Devcon 5

R&D

Optimistic rollups
introduced

Devcon 6

Proliferation

Superchain & OP Stack
introduced
[come deploy!]

MEMES OF MEMECRAFT

Devcon

Devcon 6



R&D

Optimistic rollups
introduced

Proliferation

Superchain & OP Stack
introduced
[come deploy!]





1) Manifesto



Memes of Memecraft Memefesto

We the Memers of Memecraft, together in humanity's darkest hour, establish the following manifesto for the Mother of Memes—the Decentralized Autonomous Organization (DAO) overseeing the Memes of Memecraft (MoM).



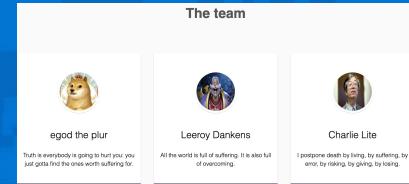
1) Manifesto



2) Team Page

Memes of Memecraft Memefesto

We the Memers of Memecraft, together in humanity's darkest hour, establish the following manifesto for the Mother of Memes—the Decentralized Autonomous Organization (DAO) overseeing the Memes of Memecraft (MoM).





1) Manifesto

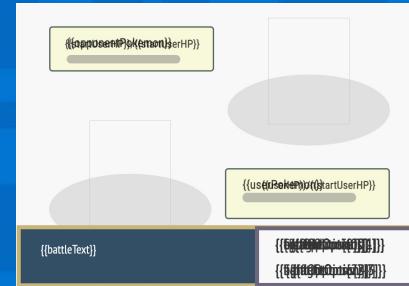
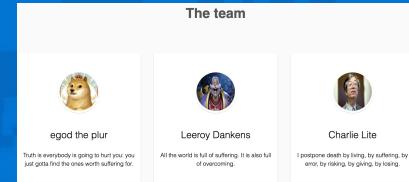


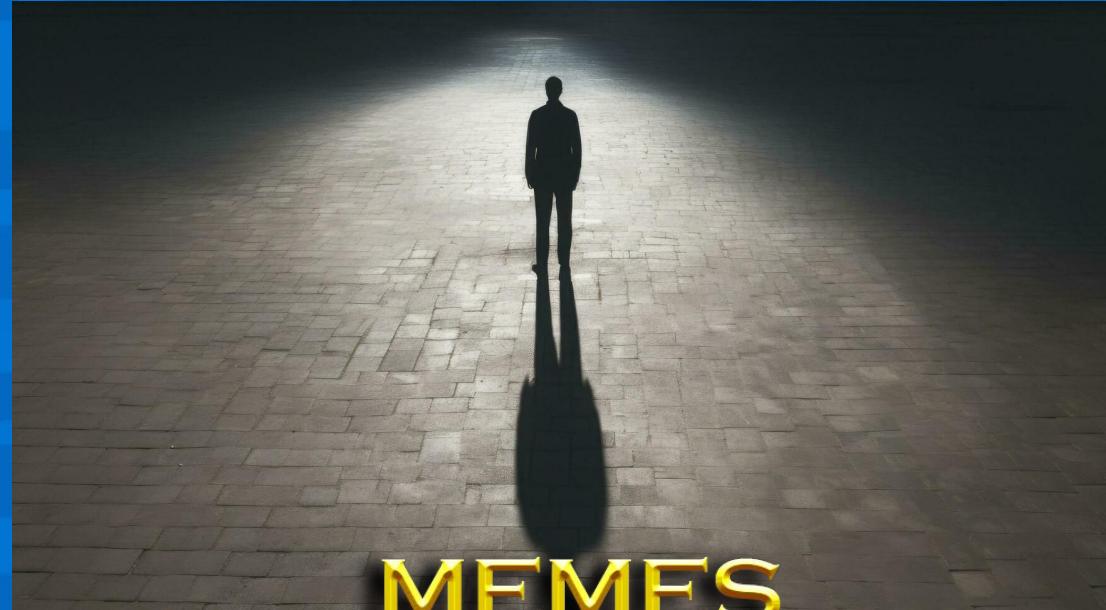
2) Team Page

3) An unusable app

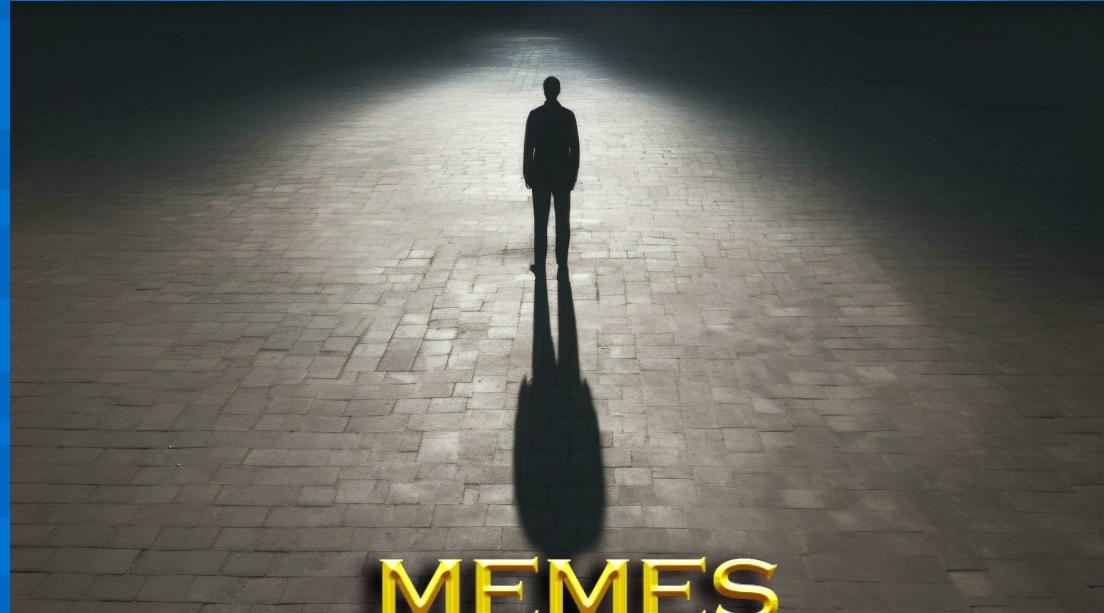
Memes of Memecraft Memefesto

We the Memers of Memecraft, together in humanity's darkest hour, establish the following manifesto for the Mother of Memes—the Decentralized Autonomous Organization (DAO) overseeing the Memes of Memecraft (MoM).





MEMES OF MEMECRAFT



MEMES OF MEMECRAFT

10 TPS, 10s latency,
NGMI

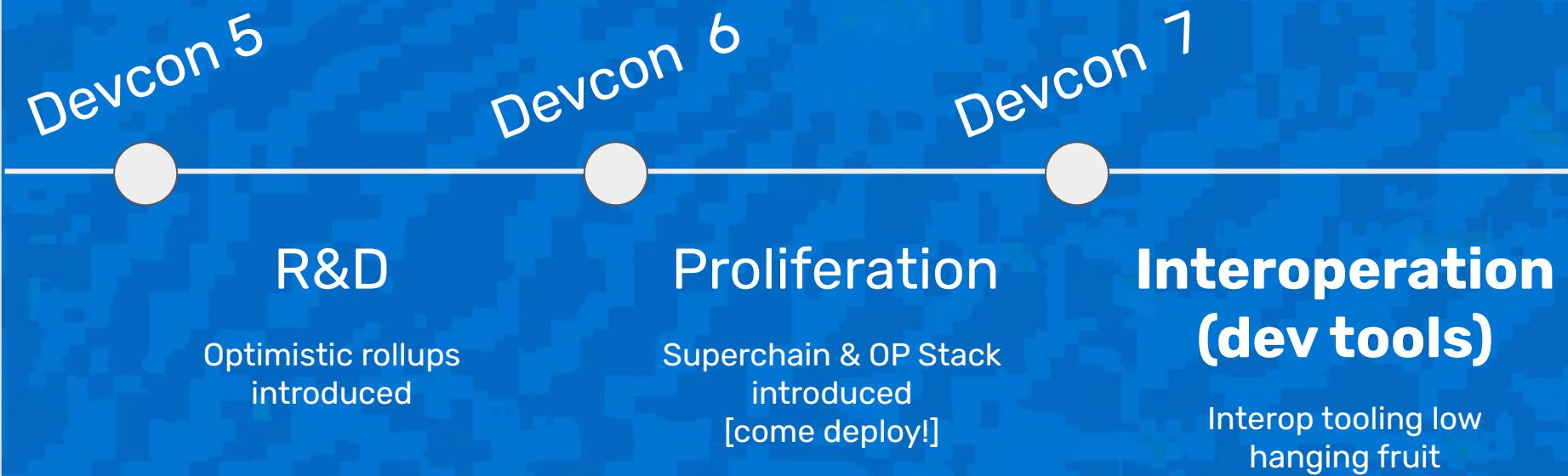




MEMES OF MEMECRAFT









Devcon 1

tion

Stack

!]

Interoperation (dev tools)

Interop tooling low
hanging fruit



The fruits are ready to pluck

Tried and true programming concepts can streamline UX

- Async await / promises
- Passkeys
- Sessions

No reason they can't be used in crypto



1. The **User** Journey
2. The **Developer** Journey
3. The **Tech**



1. The User Journey

2. The Developer Journey

3. The Tech

User Journey

1. Login
2. Mint
3. Battle
4. Brag

User Journey

1. Login
2. Mint
3. Battle
4. Brag

User Journey

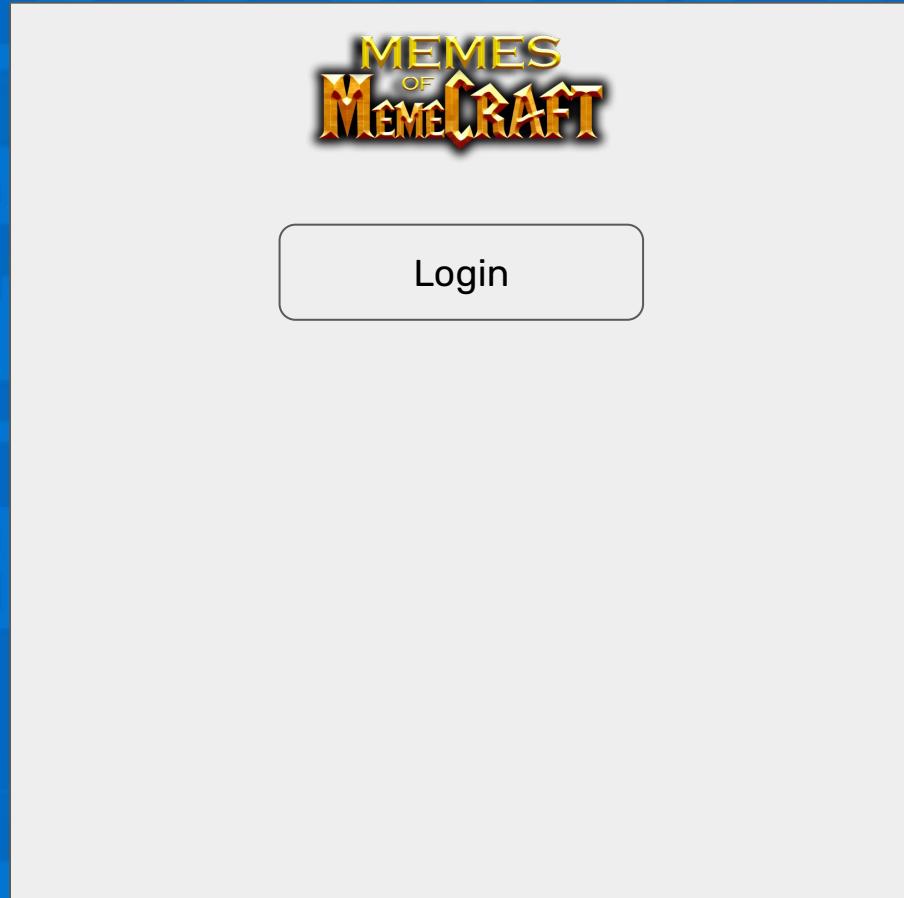
1. Login

- Familiar terms
- No seeds or PKs
- Only one signature

2. Mint

3. Battle

4. Brag



User Journey

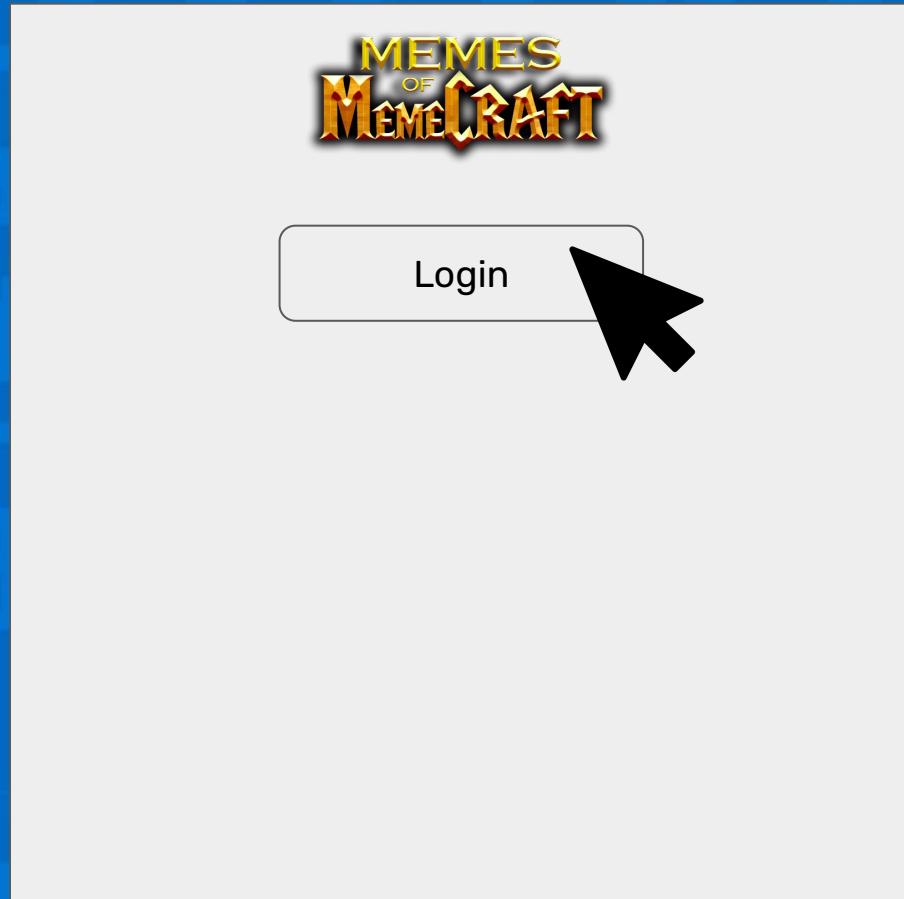
1. Login

- Familiar terms
- No seeds or PKs
- Only one signature

2. Mint

3. Battle

4. Brag



User Journey

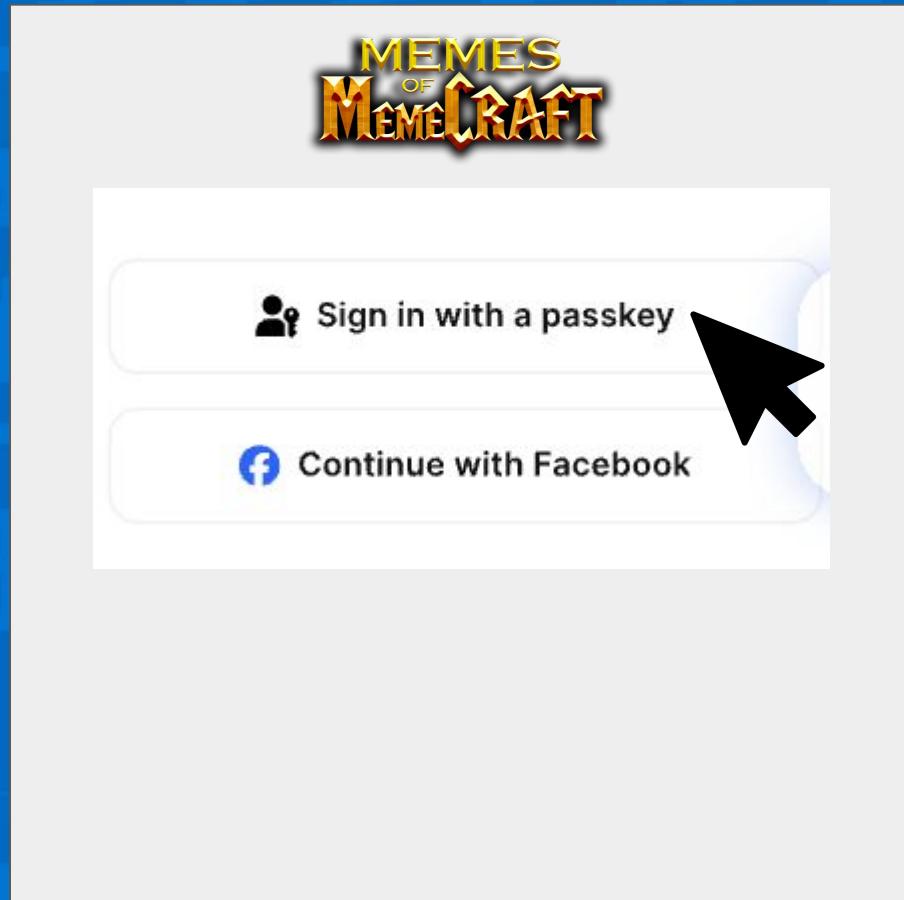
1. Login

- Familiar terms
- No seeds or PKs
- Only one signature

2. Mint

3. Battle

4. Brag



User Journey

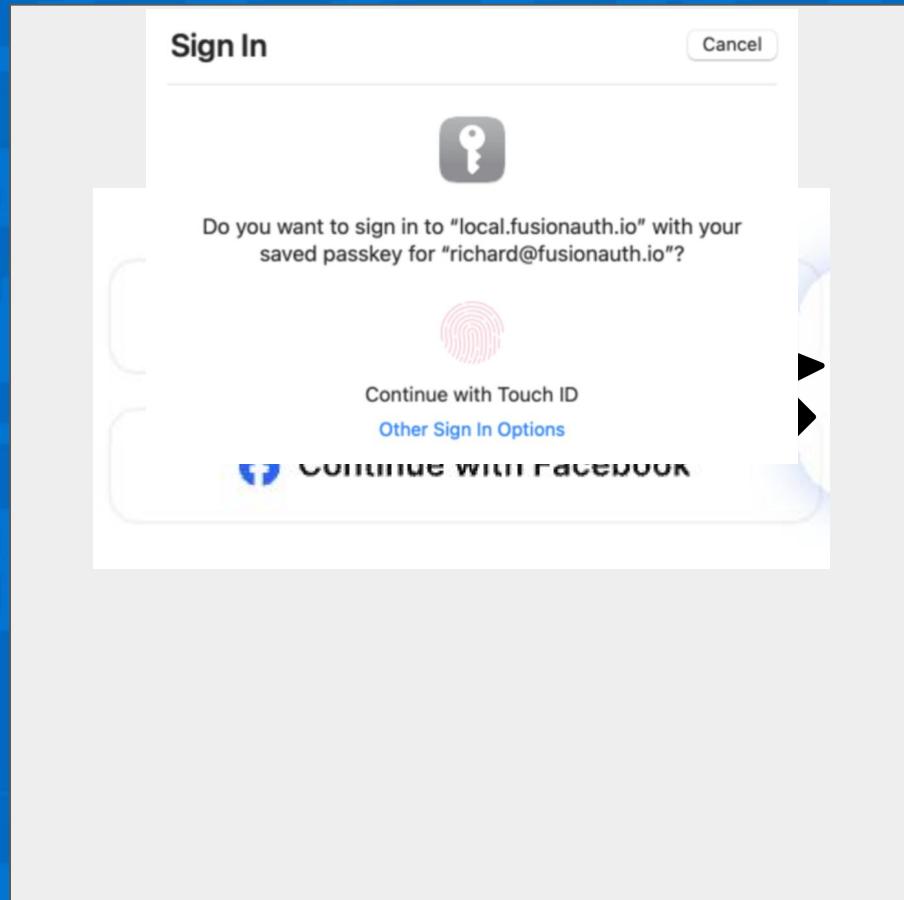
1. Login

- Familiar terms
- No seeds or PKs
- Only one signature

2. Mint

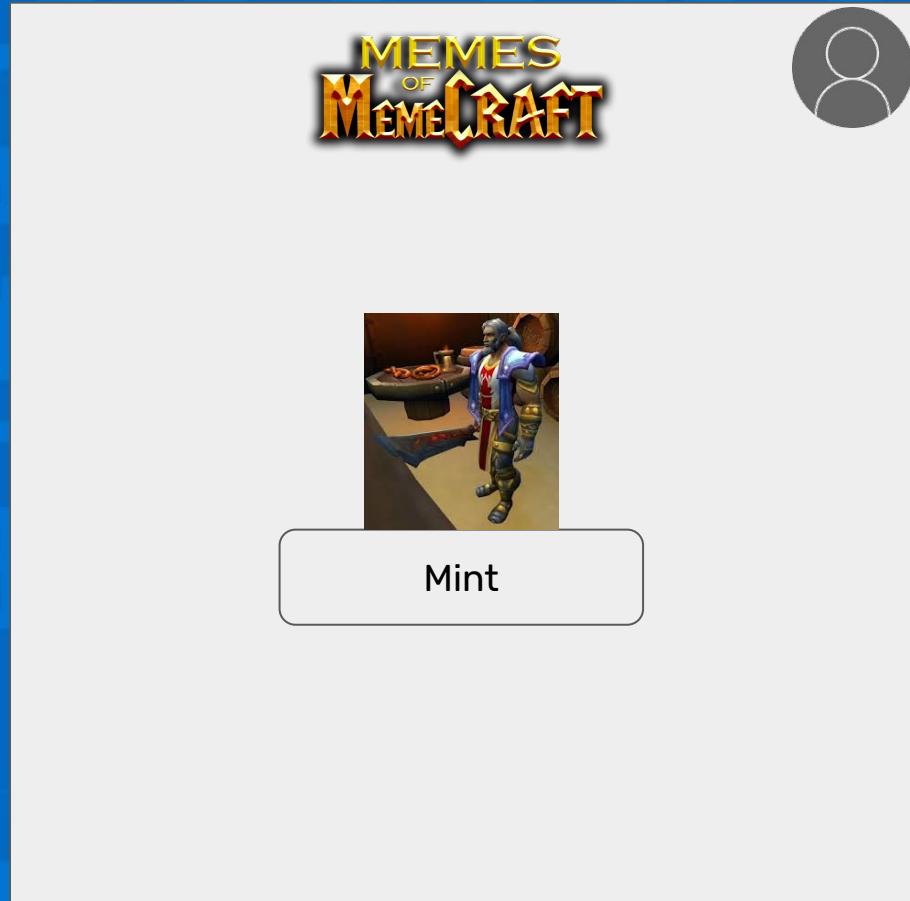
3. Battle

4. Brag



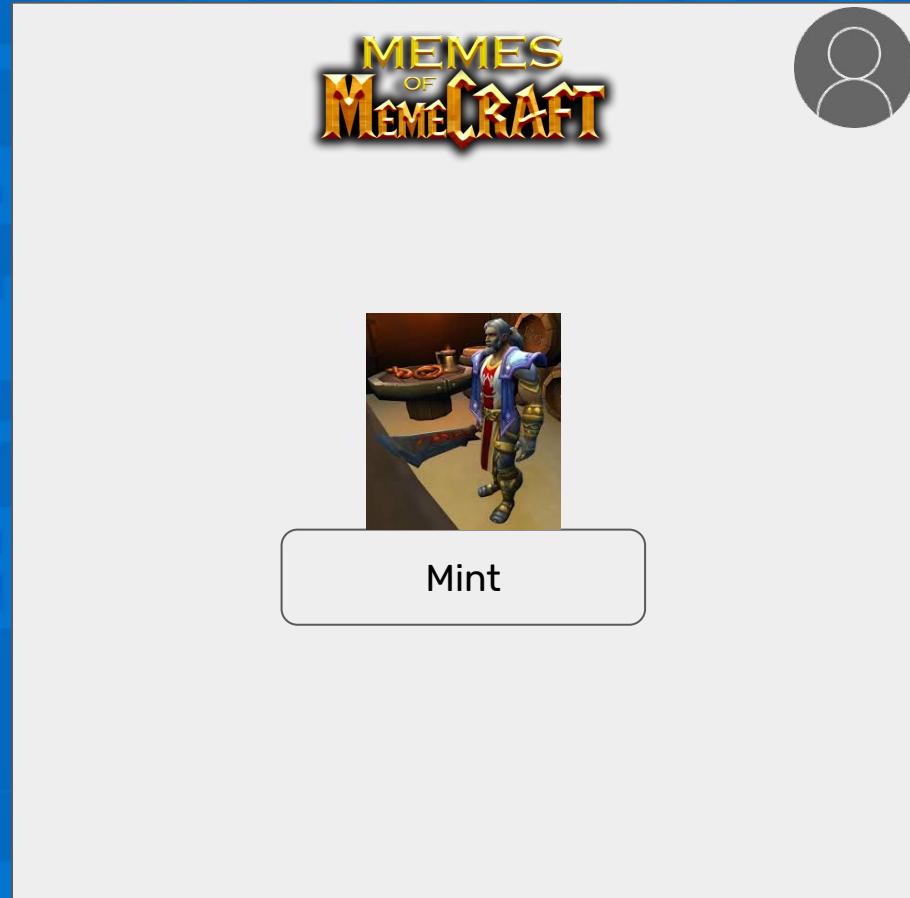
User Journey

1. Login
2. Mint
3. Battle
4. Brag



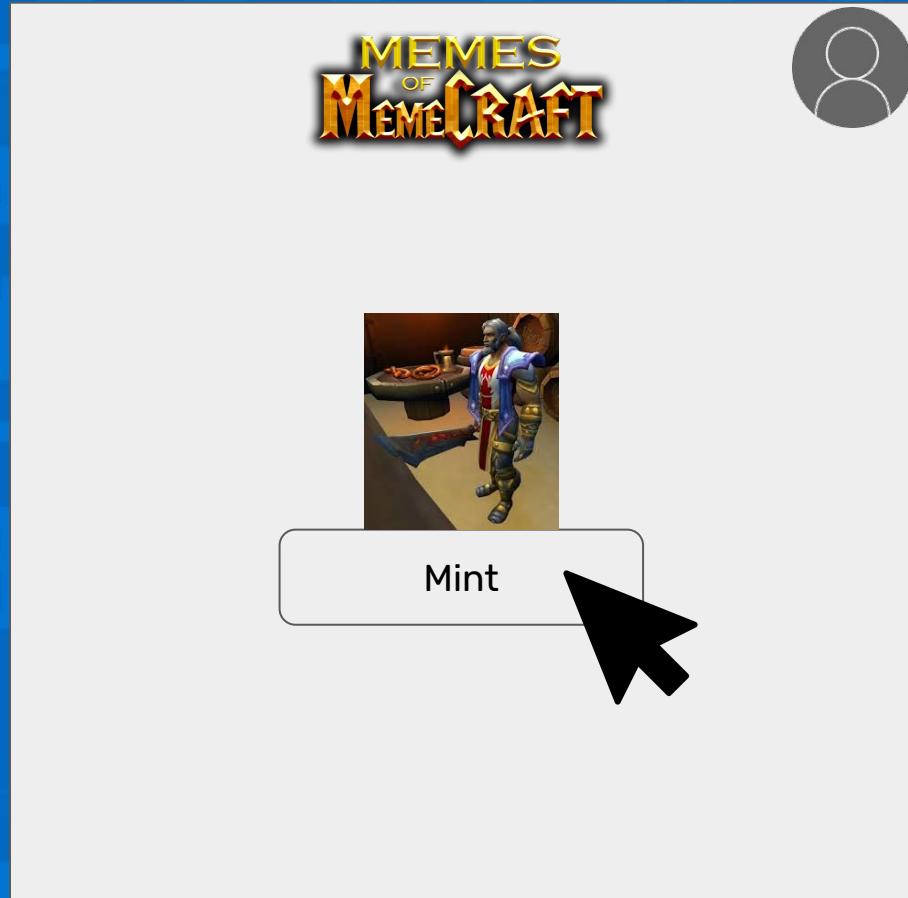
User Journey

1. Login
2. Mint
 - No signature required
 - Sub second latency
3. Battle
4. Brag



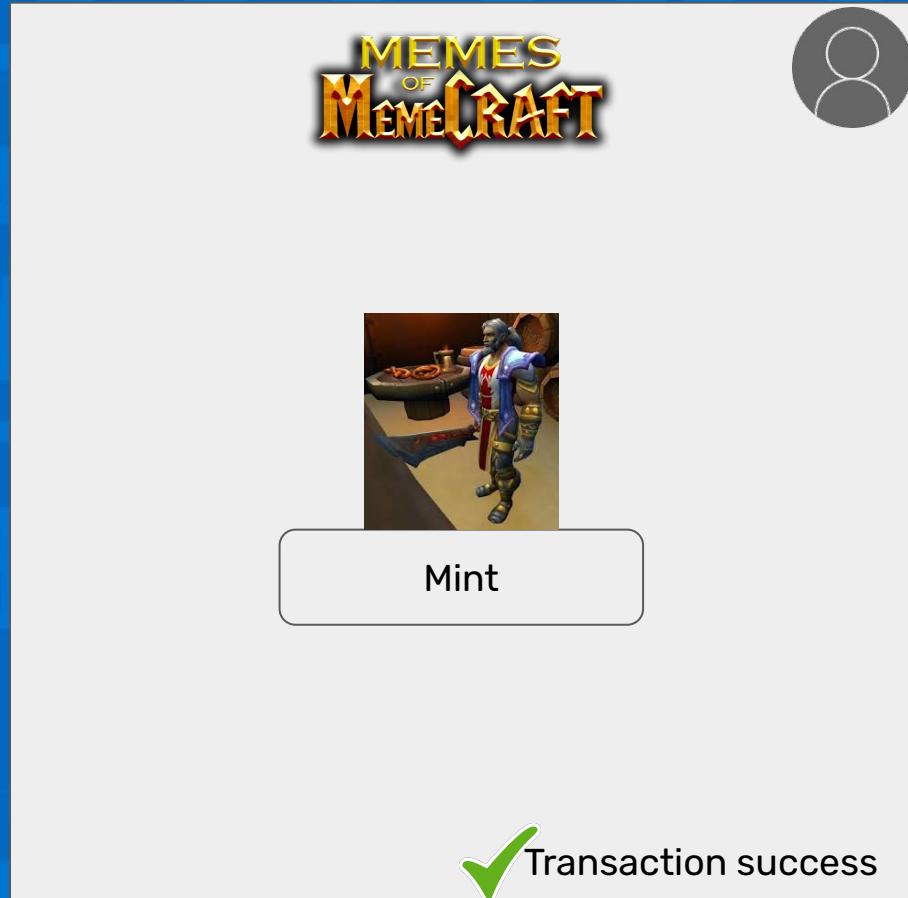
User Journey

1. Login
2. Mint
 - No signature required
 - Sub second latency
3. Battle
4. Brag



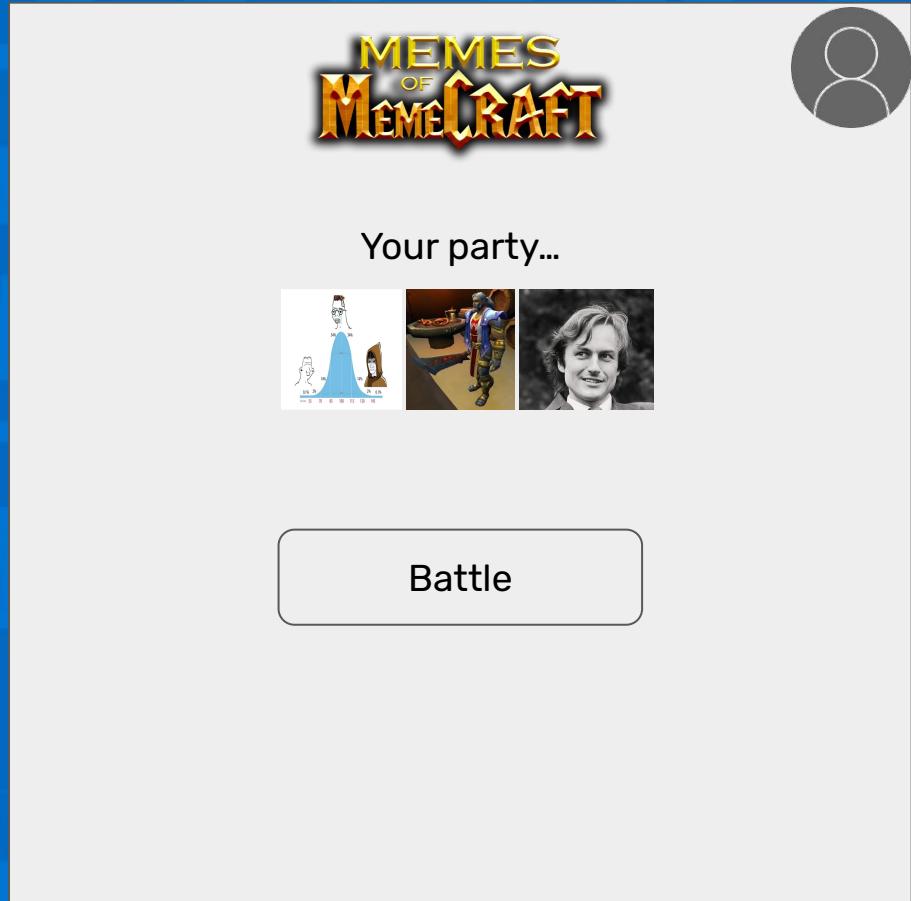
User Journey

1. Login
2. Mint
 - No signature required
 - Sub second latency
3. Battle
4. Brag



User Journey

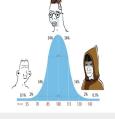
1. Login
2. Mint
3. Battle
4. Brag



The image shows a landing page for 'MEMES OF MEMECRAFT'. At the top right is a user icon. The title 'MEMES OF MEMECRAFT' is displayed in a stylized font. Below the title, the text 'Your party...' is followed by three small images: a cartoon character, a video game character, and a portrait of a man. A large button at the bottom right contains the word 'Battle'.

MEMES
OF
MEMECRAFT

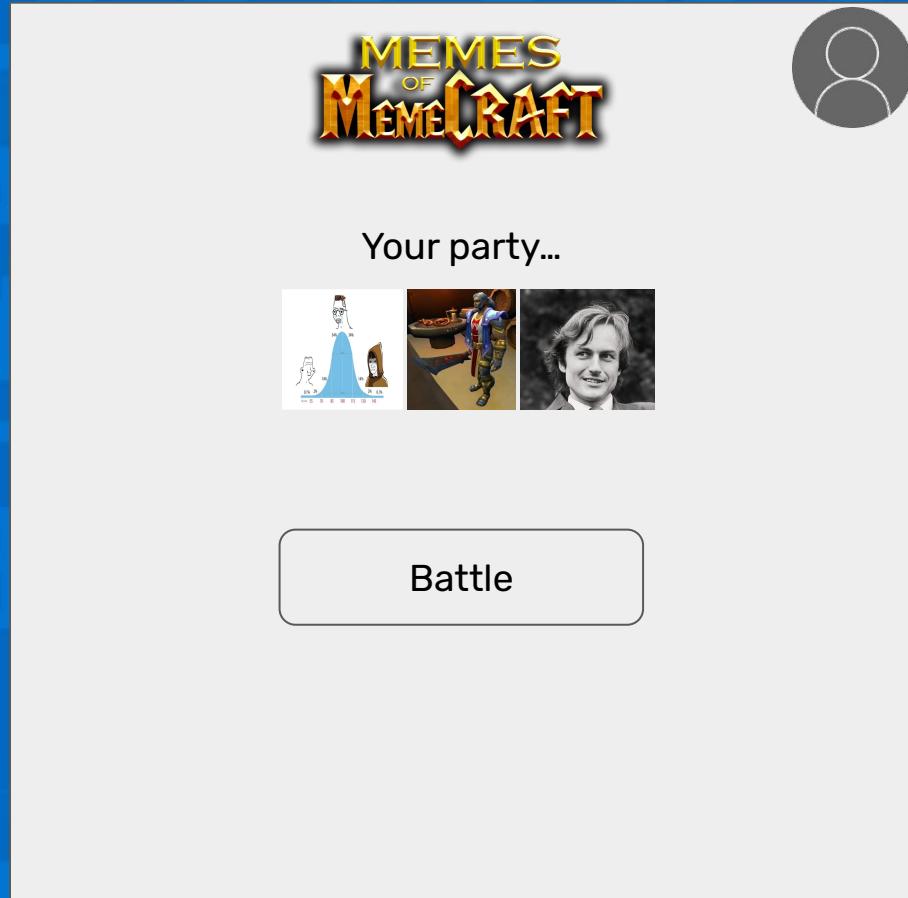
Your party...



Battle

User Journey

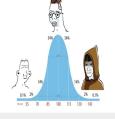
1. Login
2. Mint
3. Battle
4. Brag



The image shows a user interface for a game called "MEMES OF MEMECRAFT". At the top right is a circular profile icon. The title "MEMES OF MEMECRAFT" is displayed in a stylized font where "MEME" is written vertically. Below the title, the text "Your party..." is centered. Underneath this text are three small images: a cartoon character, a video game character, and a portrait of a man. At the bottom center is a large, rounded rectangular button with the word "Battle" in white text.

MEMES
OF
MEMECRAFT

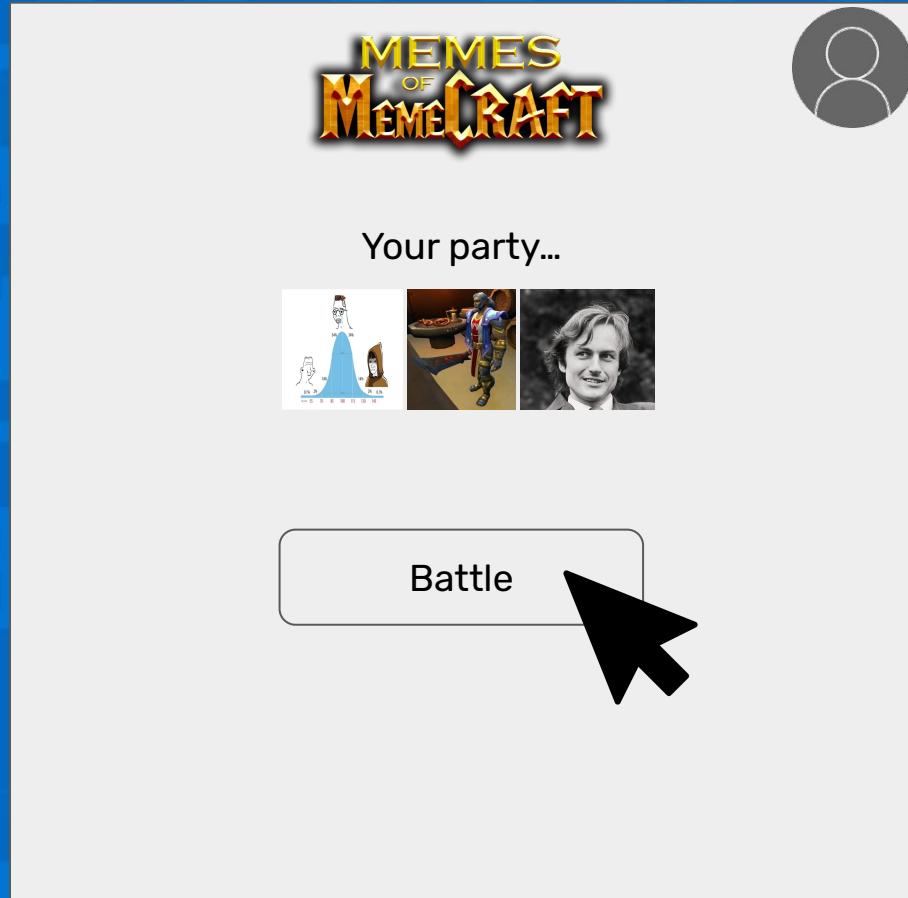
Your party...



Battle

User Journey

1. Login
2. Mint
3. Battle
4. Brag



User Journey

1. Login
2. Mint
3. Battle
4. Brag

The screenshot shows a mobile application interface with a white background. At the top right is a circular profile icon. In the center is the title "MEMES OF MEMECRAFT" in a stylized font. Below the title, the text "Your party..." is displayed. Underneath this text are three small images: a cartoon character, a video game character, and a portrait of a man. At the bottom left is a large rectangular button with the word "Battle" in the center. At the bottom right, the text "Matchmaking..." is visible.

MEMES
OF
MEMECRAFT

Your party...

Battle

Matchmaking...

User Journey

1. Login
2. Mint
3. Battle
4. Brag

The image shows a digital interface for the game "MEMES OF MEMECRAFT". At the top right is a circular profile icon. The title "MEMES OF MEMECRAFT" is displayed prominently. Below the title, the text "Your party..." is shown above three small images: a cartoon character, a video game character, and a portrait of a man. A large rectangular button labeled "Battle" is centered below these images. At the bottom of the screen, the text "Matchmaking... FOUND" is displayed.

Memes
OF
MEMECRAFT

Your party...

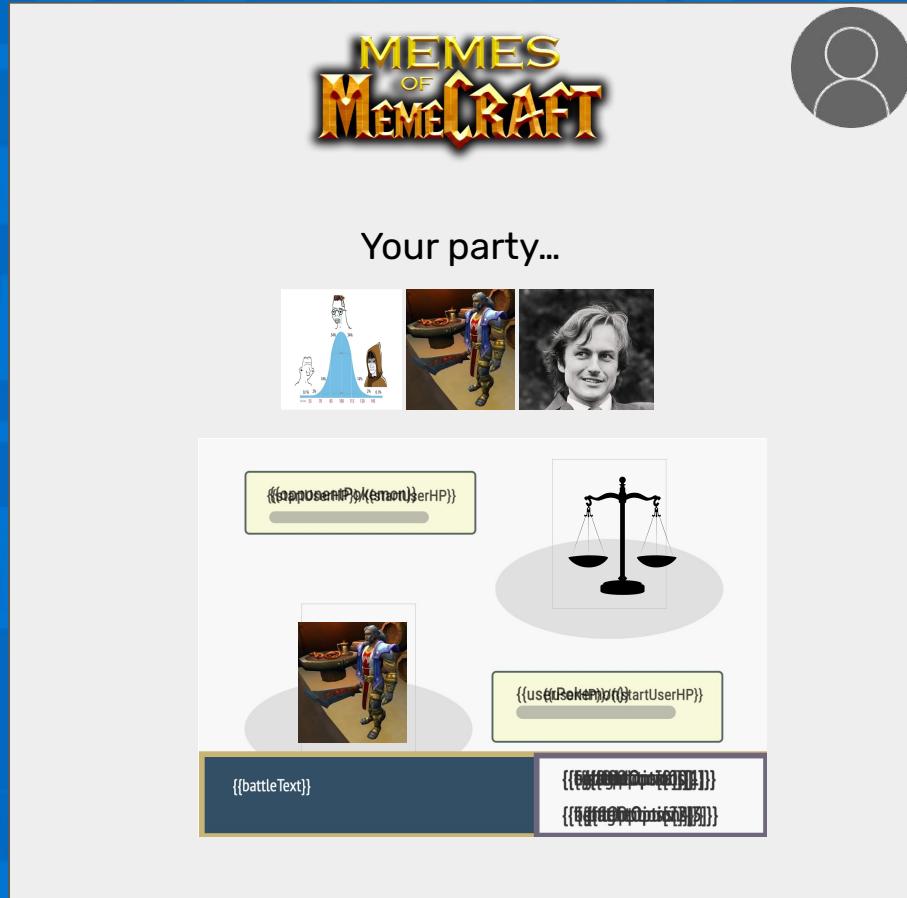
1% 10% 20% 30% 40%

Battle

Matchmaking... FOUND

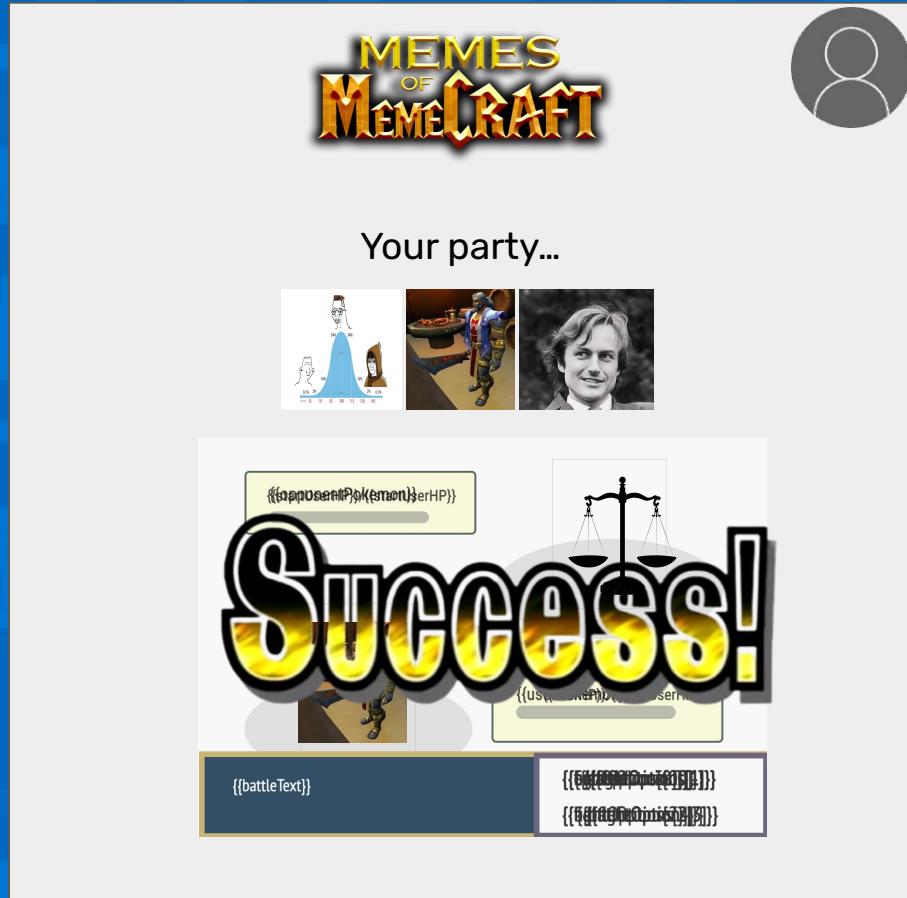
User Journey

1. Login
2. Mint
3. Battle
4. Brag



User Journey

1. Login
2. Mint
3. Battle
4. Brag



User Journey

1. Login
2. Mint
3. Battle
4. Brag

The screenshot shows a mobile application interface with a light gray background. At the top right is a circular profile icon with a stylized 'M' symbol. In the center is the logo "MEMES OF MEMECRAFT" with "MEMES OF" in a smaller font above "MEMECRAFT" in a large, bold, gold-colored font. Below the logo is a large blue Twitter bird icon. A white rounded rectangle contains the text "Share battle on memecaster!". Below this are three horizontal message cards, each with a user icon and text:

- I like to meme...
- MoM is sick
- wagmi

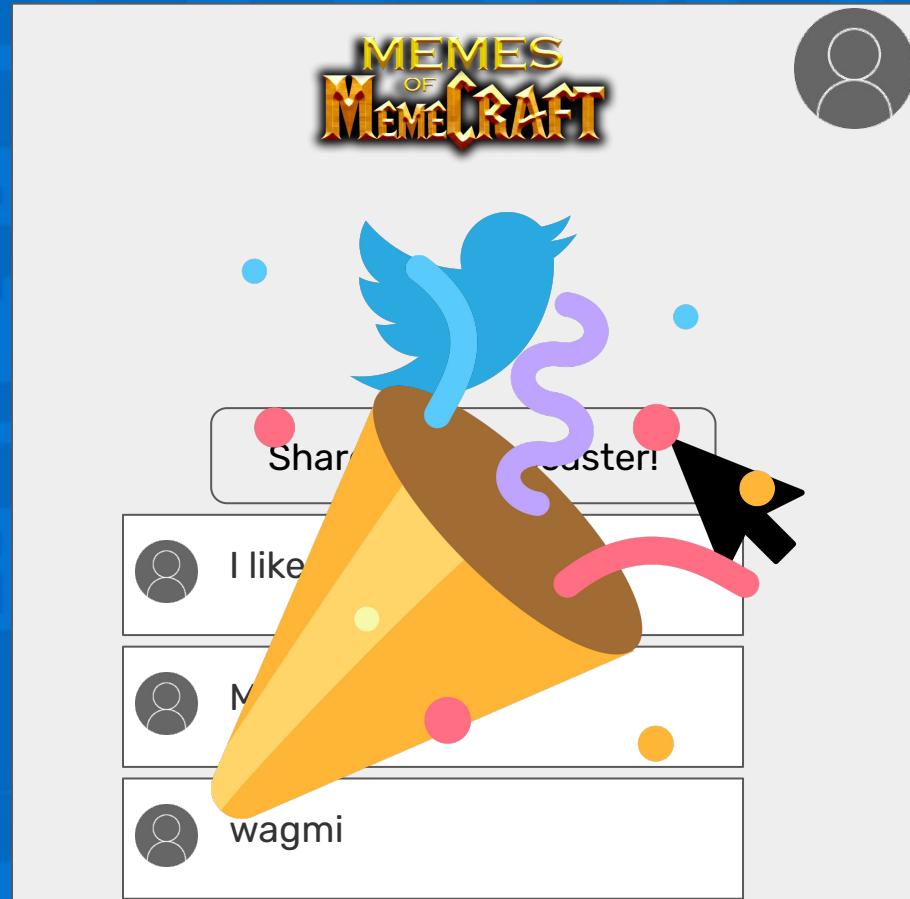
User Journey

1. Login
2. Mint
3. Battle
4. Brag



User Journey

1. Login
2. Mint
3. Battle
4. Brag



Simple right?



1. The **User** Journey
2. The **Developer** Journey
3. The **Tech**



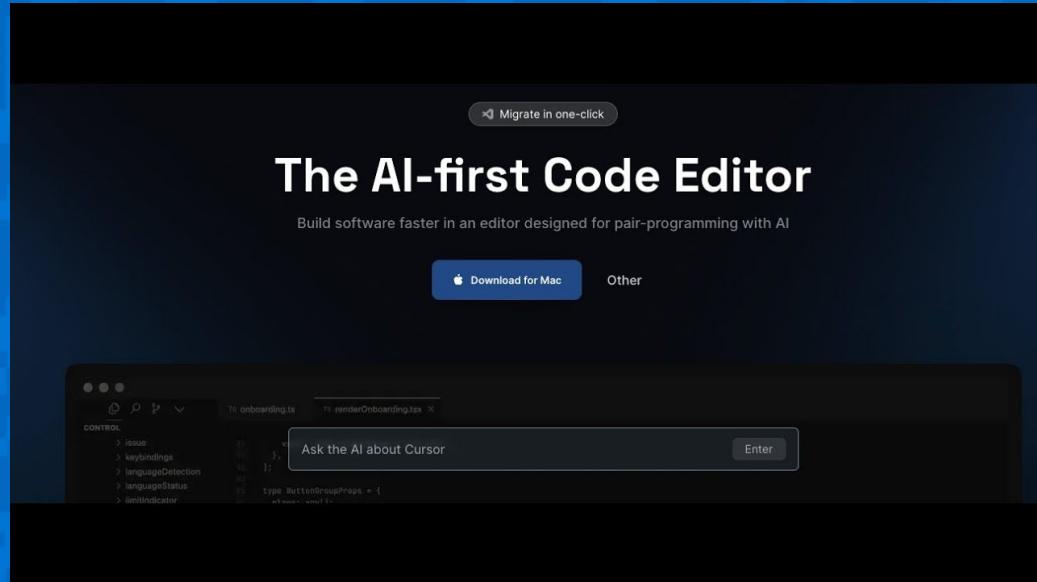
1. The **User Journey**
2. The **Developer Journey**
3. The **Tech**

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

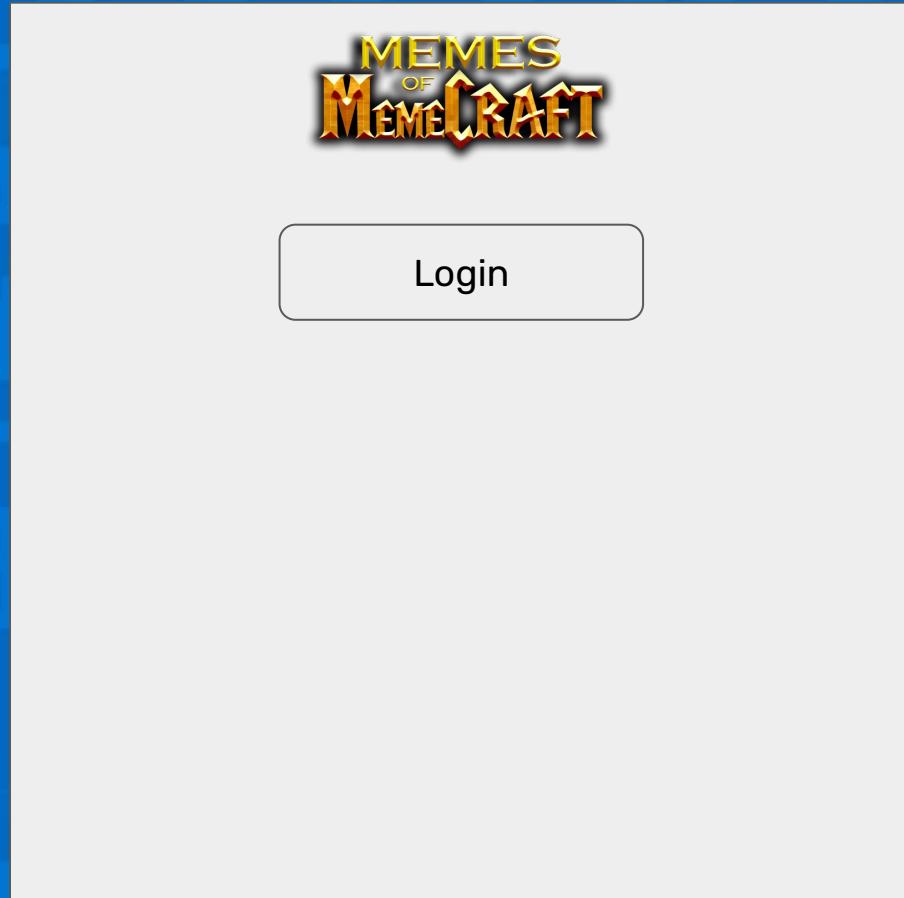


I'm going to simplify a bit... Imagine an

- AI code editor,
- in browser,
- auto refresh,
- solidity testing (if needed)

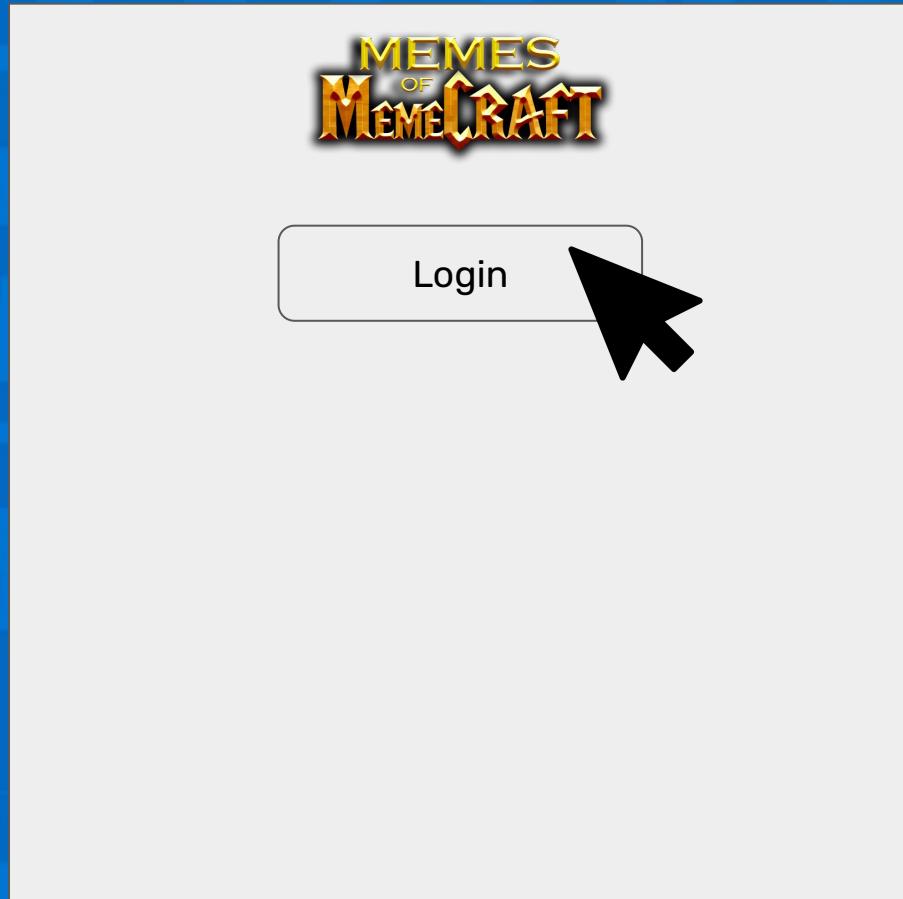
Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



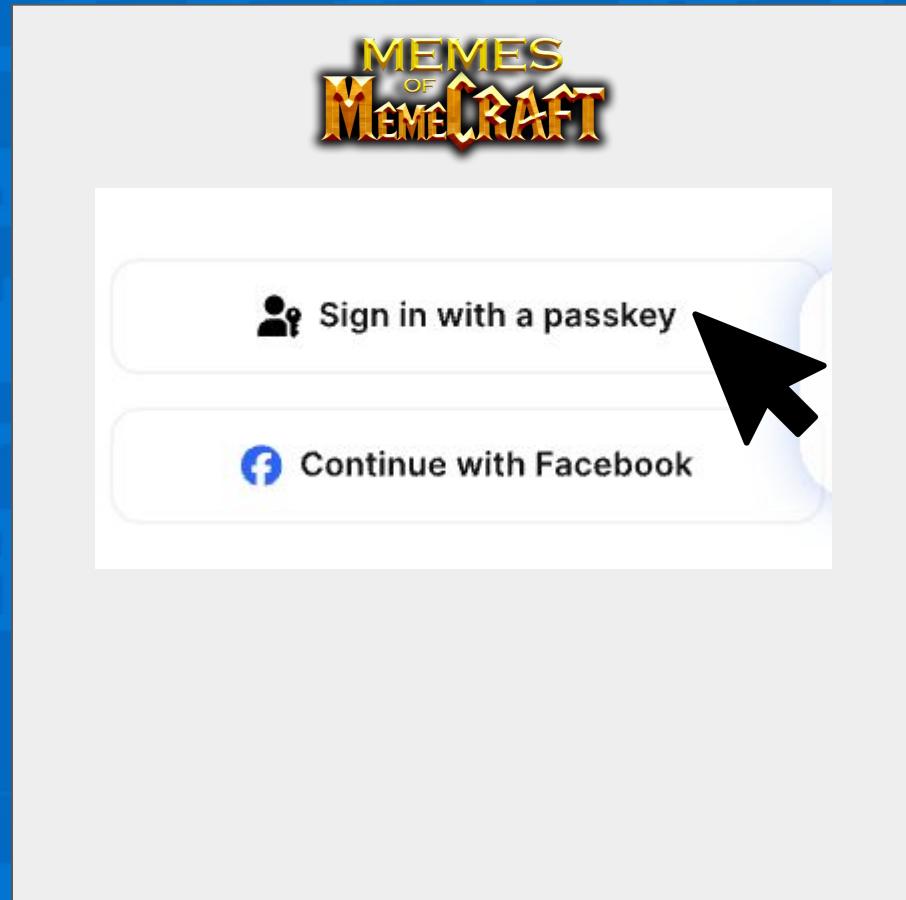
Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building Login*
2. *Building Mint*
3. *Building Battle*
4. *Building Brag*



Developer Journey

1. *Building Login*
2. *Building Mint*
3. *Building Battle*
4. *Building Brag*

```
const walletClient = createWalletClient({  
  account,  
  chain: customChain,  
  transport: http(rpcUrl)  
})
```

We need...

- Passkey login
- Smart wallet with sessions
- SuperRPC

All built into frontend libs

Developer Journey

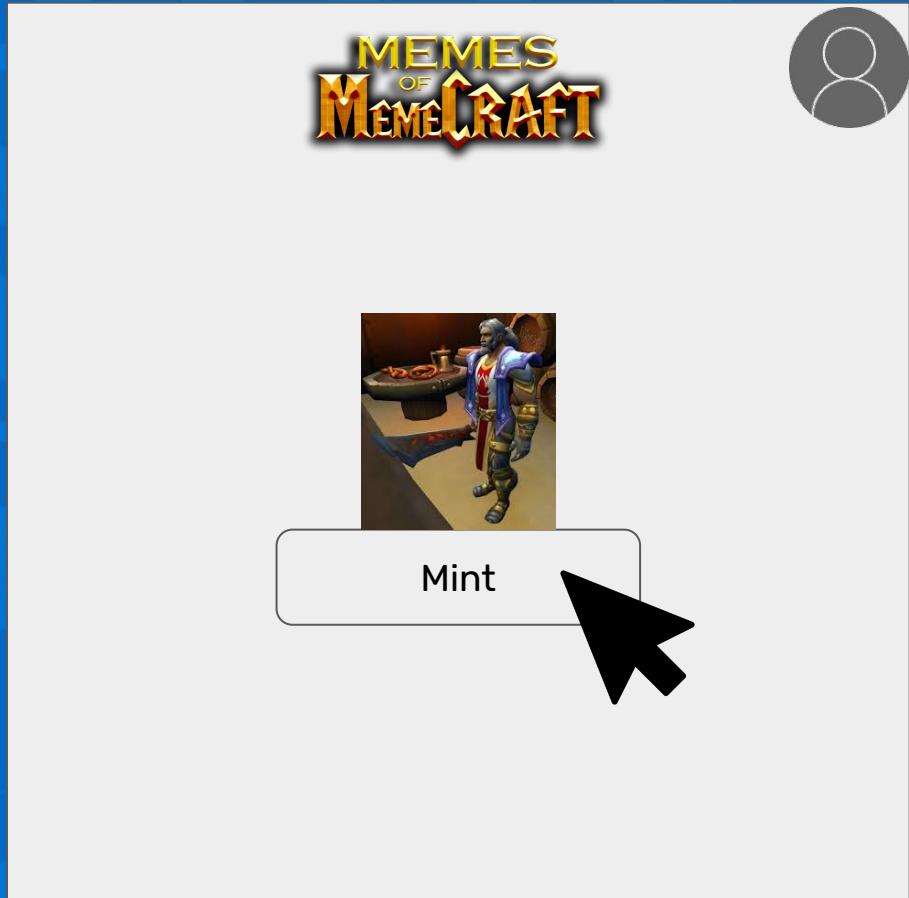
1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Mint

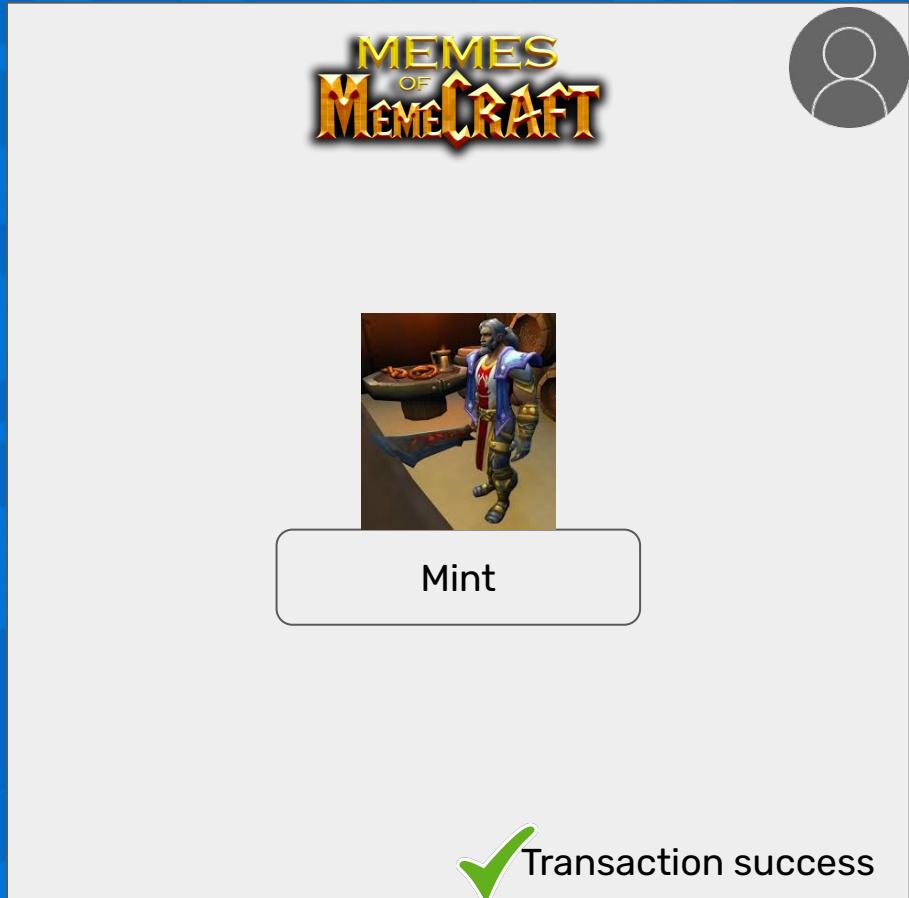
Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

Two steps:

1. Consolidate Assets
2. Write async contract

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

Two steps:

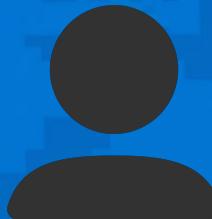
1. **Consolidate Assets**
2. Write async contract

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

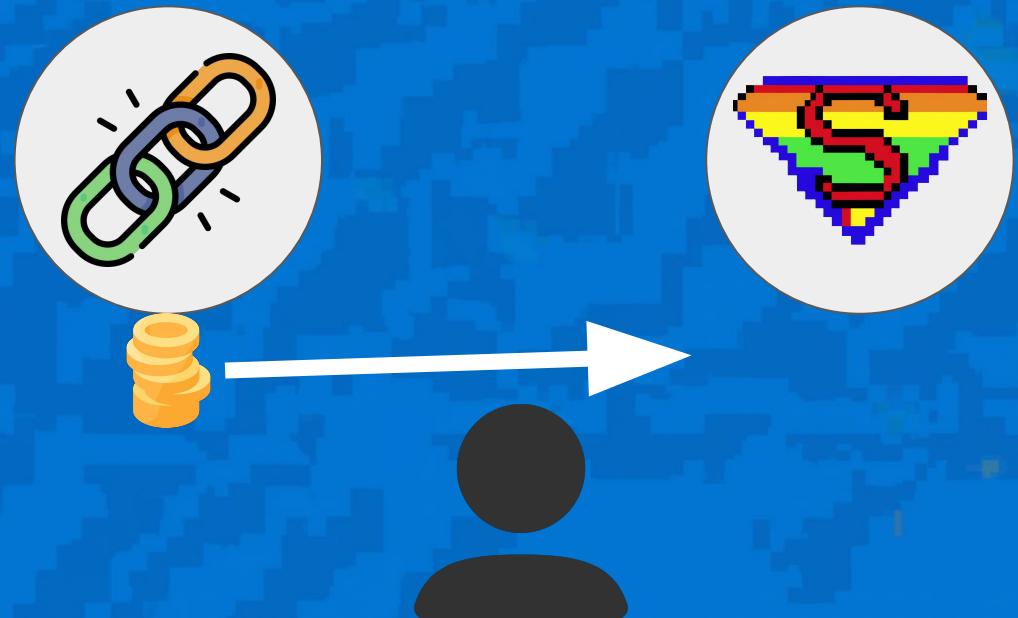
Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

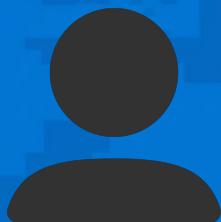


Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



7 days

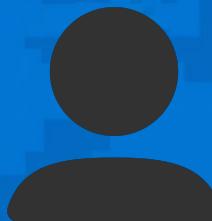


Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



7 days

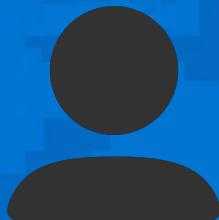


Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

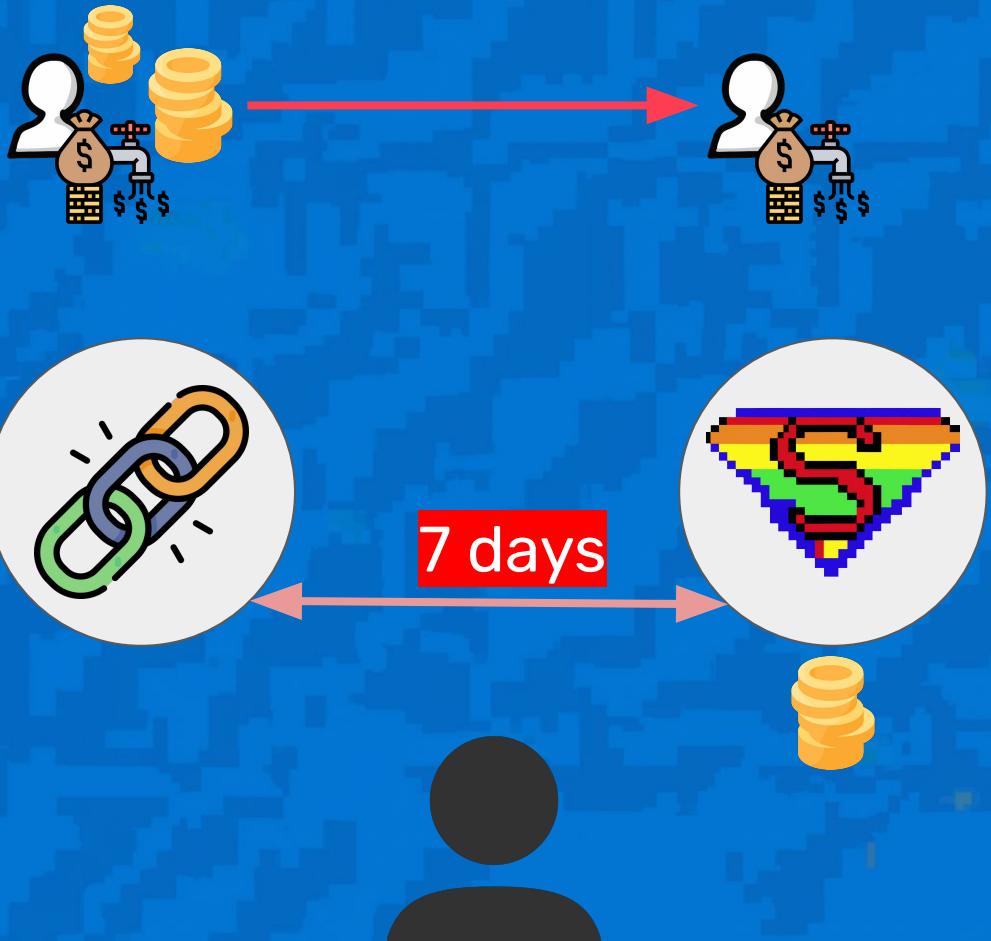


7 days



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

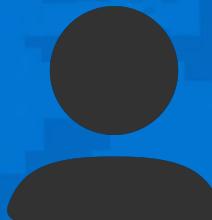


Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



7 days



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



feeeeess!



7 days



Instant!
(no matter the bridge)



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



feeeeess!



```
wallet.transferTokens(token, "*", chain_id, to)
```



7 days



Instant!
(no matter the bridge)

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

Two steps:

1. Consolidate Assets
2. Write async contract

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

Two steps:

1. Consolidate Assets
2. **Write async contract**

Developer Journey

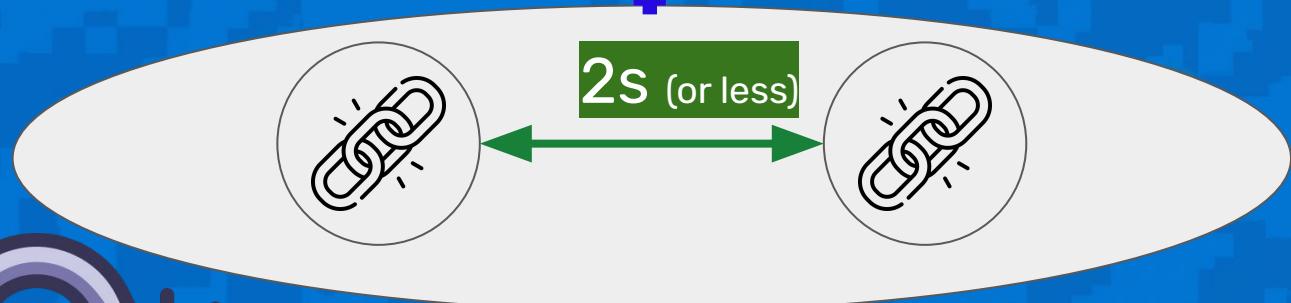
1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Superchain interop protocol
(x-chain reads)

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Superchain interop protocol
(x-chain reads)



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

```
function crossChainMemeMint() external {
    MemeMinter remoteInstance = getRemote(remoteChainId, address(this));
    Promise p = remoteInstance.mintMeme();
    p.then(this.registerMeme);
}
```

Imagine...

- Promise library
 - *Familiar async programming model*
- Async await syntax
 - *Super clean syntax (maybe too clean?)*
- Promise sponsorship
 - *No need to relay txs yourself*

...enabling developers to build & test
async smart contract systems

Developer Journey

1. *Building Login*
2. *Building Mint*
3. *Building Battle*
4. *Building Brag*

```
function crossChainMemeMint() external {
    MemeMinter remoteInstance = getRemote(remoteChainId, address(this));
    Promise p = remoteInstance.mintMeme();
    p.then(this.registerMeme);
}
```

Imagine...

- Promise library
 - *Familiar async programming model*
- Async await syntax



ben-chain / superchain-async

*...enabling developers to build & test
async smart contract systems*

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



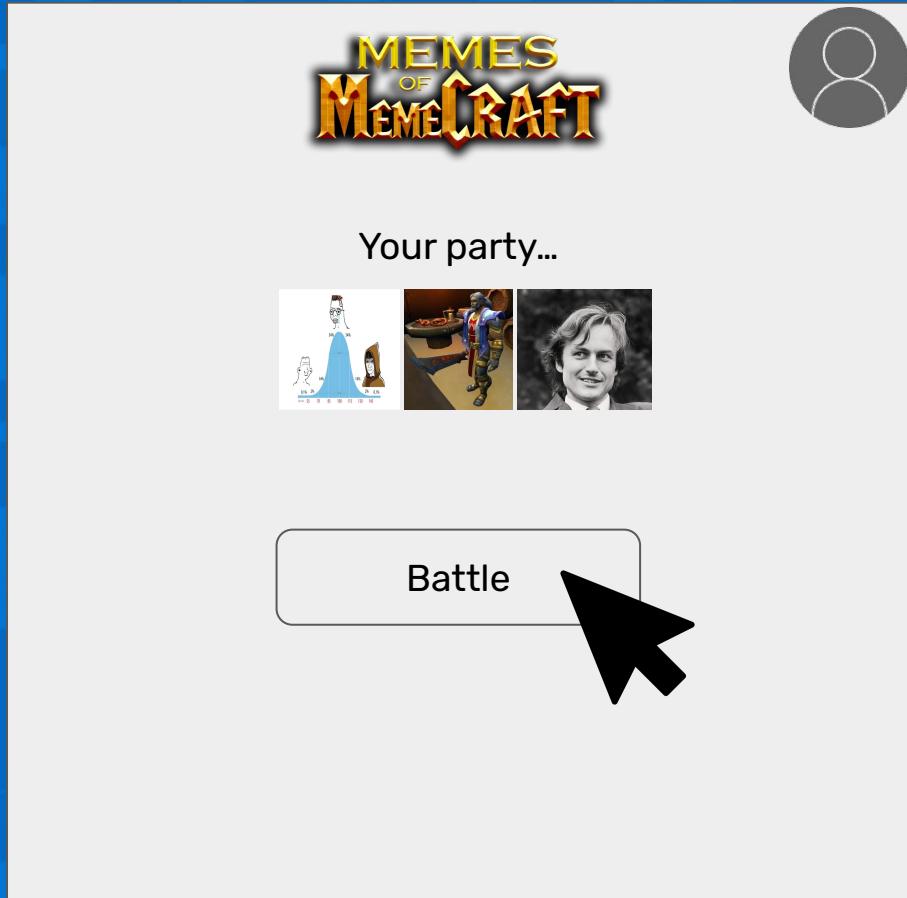
Your party...



Battle

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Your party...



Battle

Matchmaking...

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Your party...

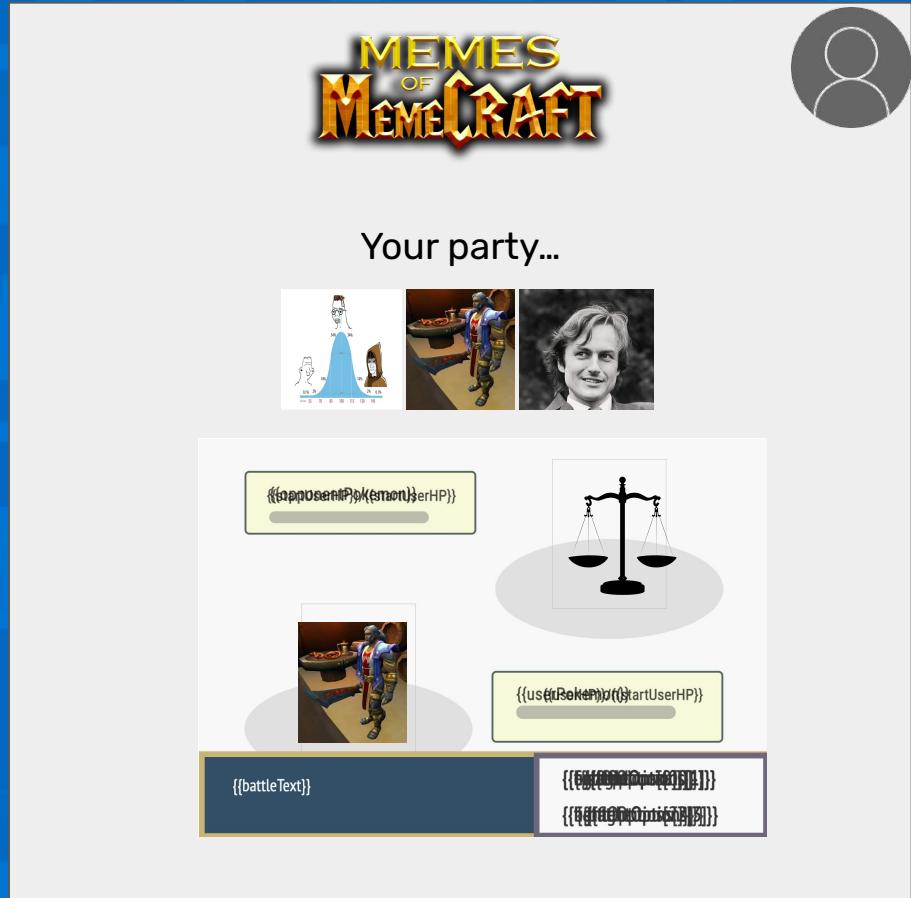


Battle

Matchmaking... FOUND

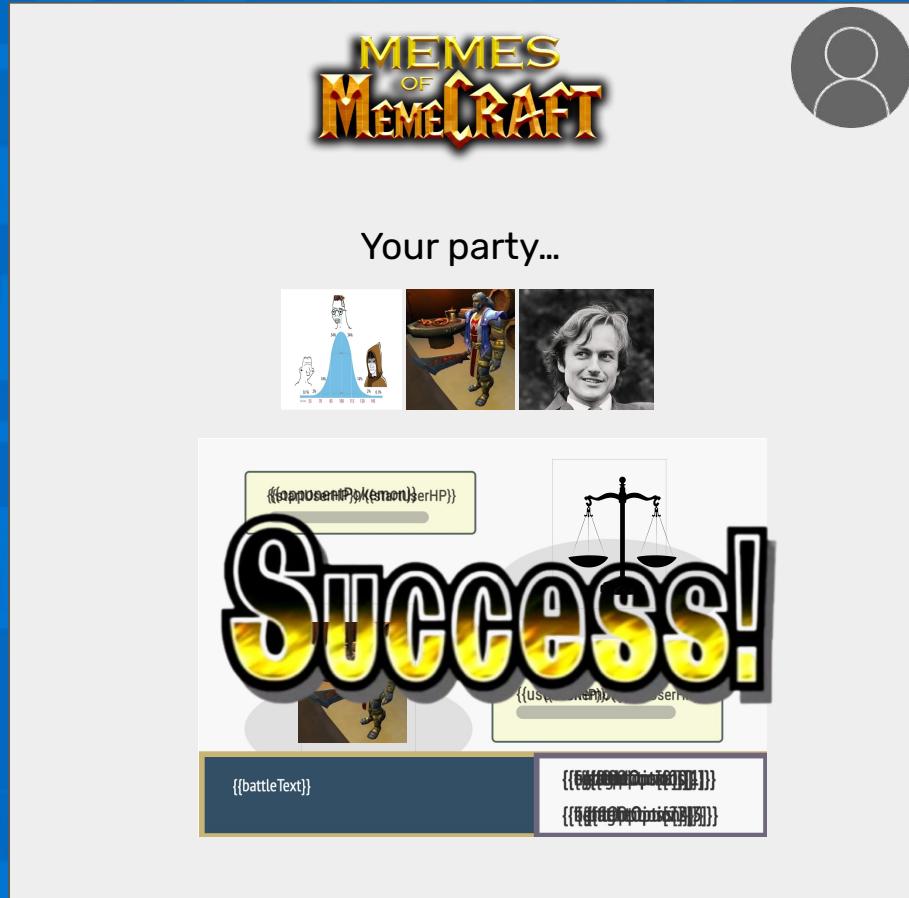
Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

Two steps:

1. Find low congestion chain
2. Install battle contracts

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

Two steps:

1. **Find low congestion chain**
2. Install battle contracts

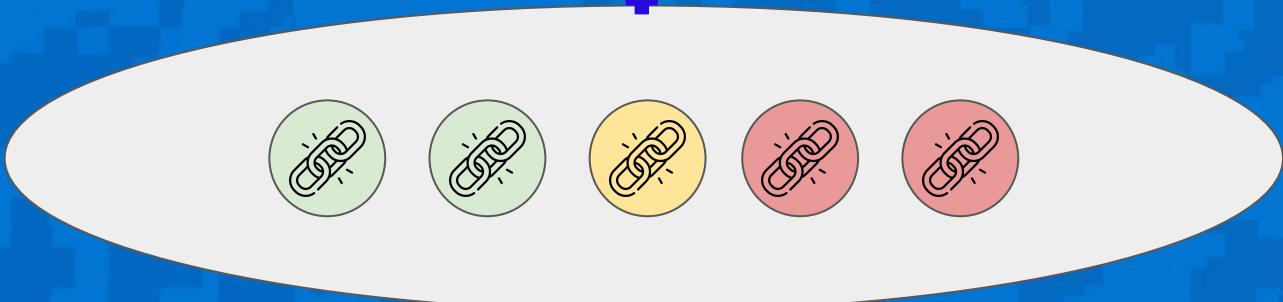
Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



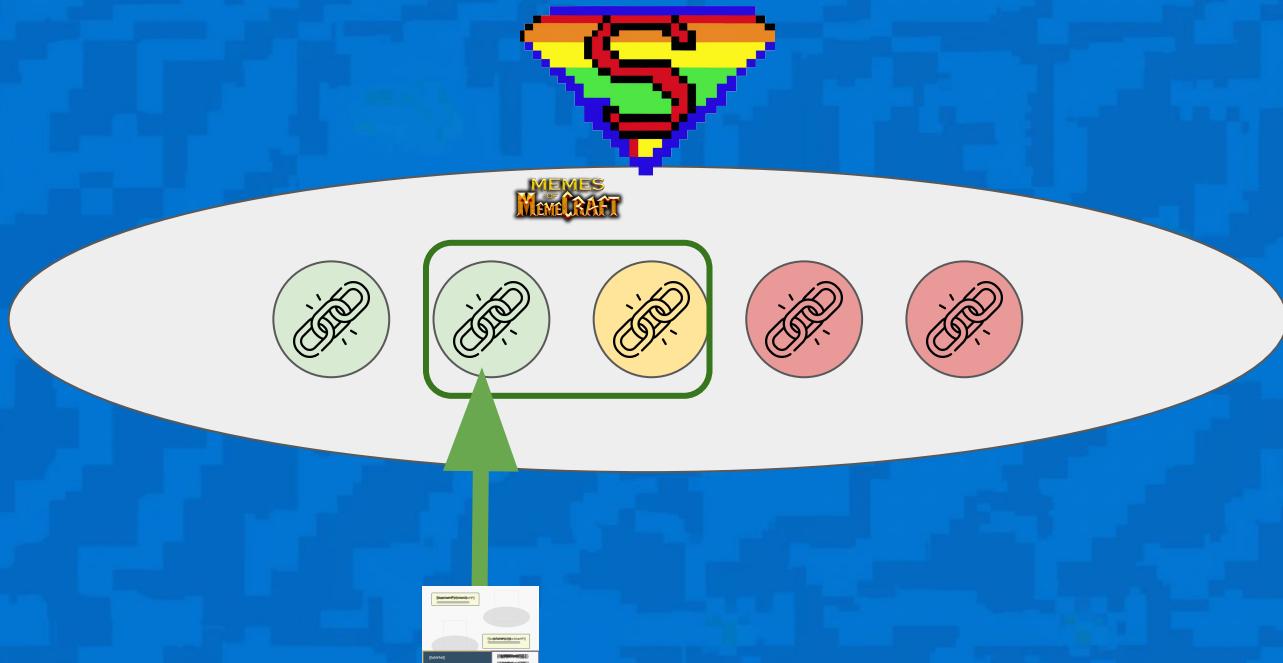
Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

Two steps:

1. Find low congestion chain
2. Install battle contracts

Developer Journey

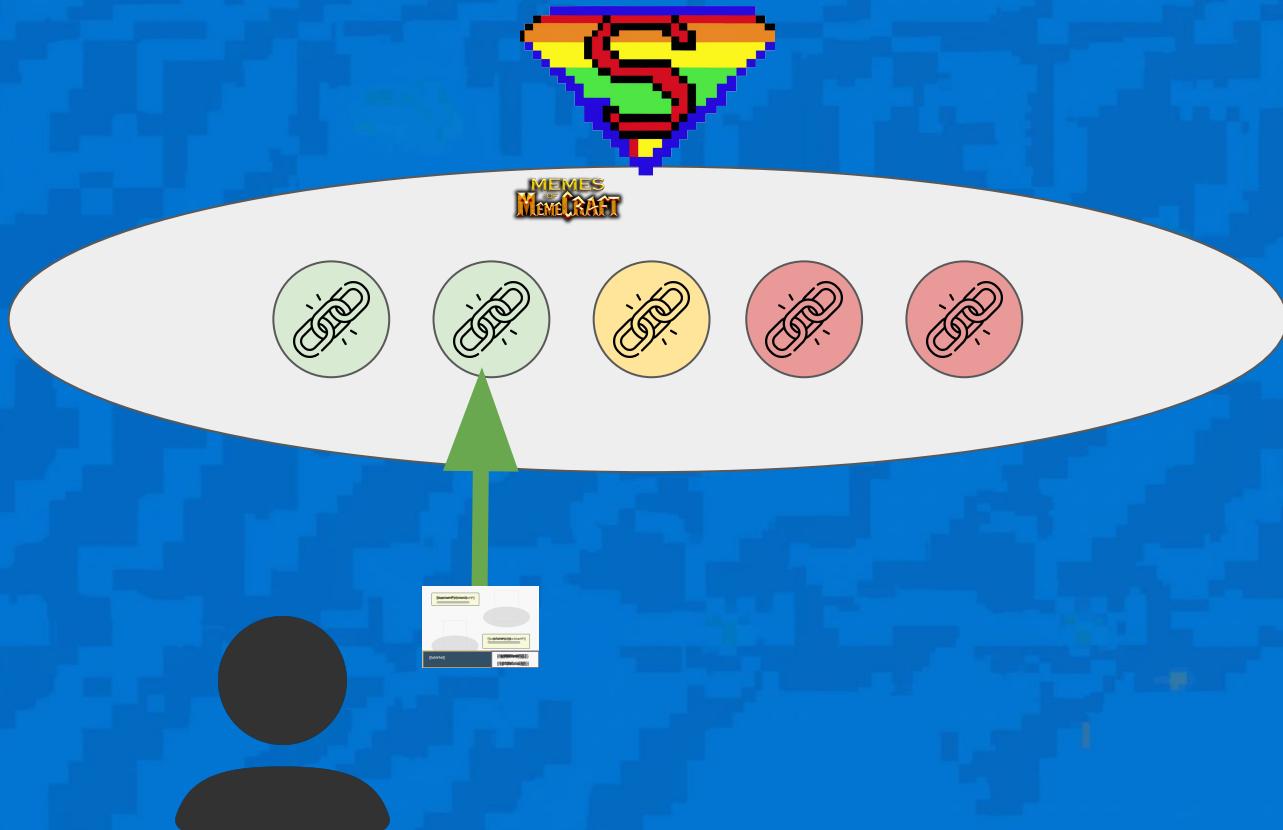
1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

Two steps:

1. Find low congestion chain
2. **Install battle contracts**

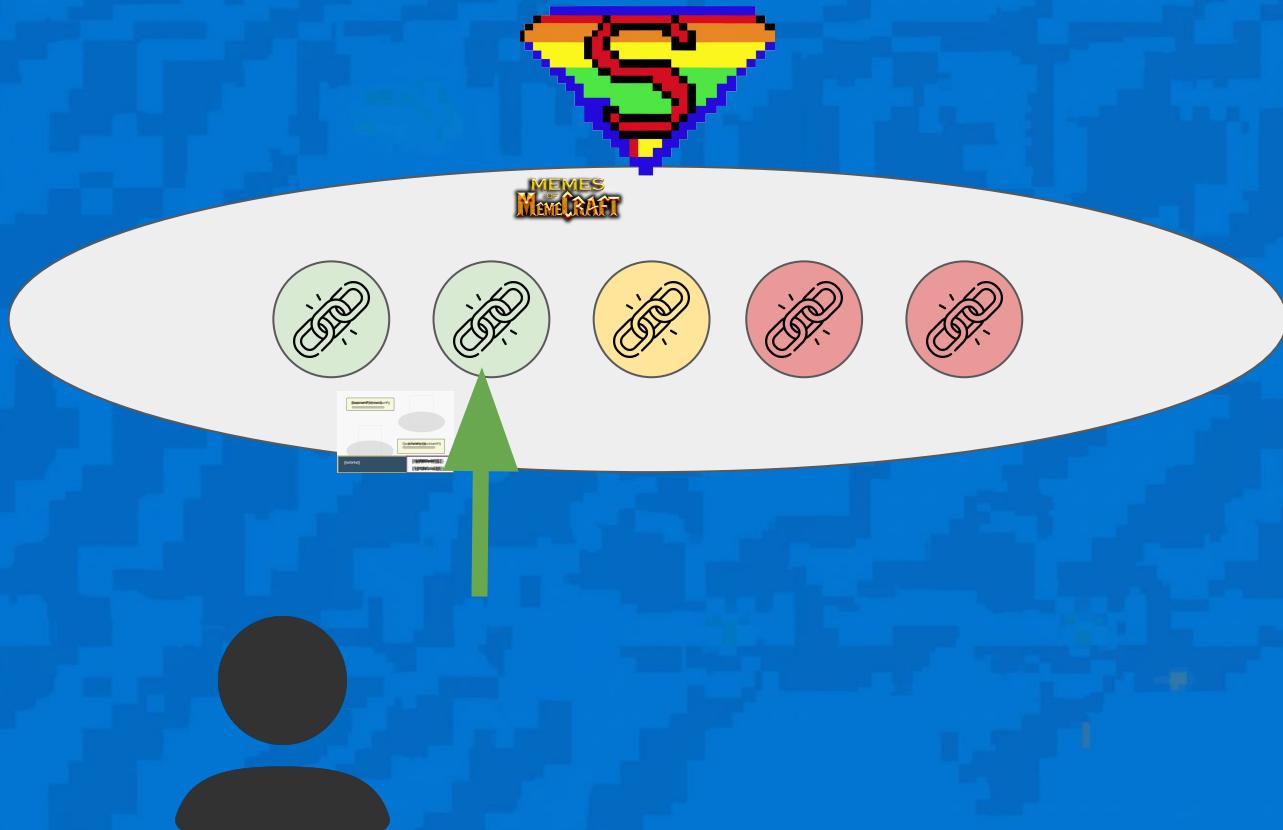
Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



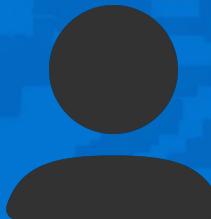
Permissionless
Deployments

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



✖️ Devs choose where to deploy



Permissionless
Deployments

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



✗ Devs choose where to deploy

✓ Users choose where to install

Permissionless Deployments

Developer Journey

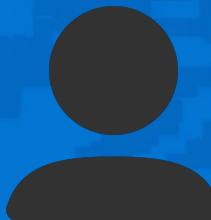
1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag

Also promises!



✗ Devs choose where to deploy

✓ Users choose where to install



Permissionless
Deployments

Developer Journey

1. *Building Login*
2. *Building Mint*
3. *Building Battle*
4. *Building Brag*

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Share battle on memecaster!

I like to meme...

MoM is sick

wagmi

Developer Journey

1. *Building* Login
2. *Building* Mint
3. *Building* Battle
4. *Building* Brag



Composability is



Developer Journey



1. *Building Login*
2. *Building Mint*
3. *Building Battle*
4. *Building Brag*

Developer Journey

1. *Building Login*
2. *Building Mint*
3. *Building Battle*
4. *Building Brag*



Mozilla Firefox



Netscape

```
<!-- HTML START -->
<header id="header-section" class="header-banner">...</header>
<section id="vision" class="am-sect">...</section>
<section id="manifesto" class="am-sect">
  <div class="container-width">
    <div class="am-container">flex
      
      <div class="am-content">
        <div class="am-title">Memes of Memecraft Memefesto </div>
        <div class="am-desc">
          ><p data-startline="3" data-endline="3" class="part c14470">...</p>
          ><p data-startline="5" data-endline="5" class="part c14479">...</p>
          ><p data-startline="7" data-endline="7" class="part c14487">...</p> == $0
          ><p data-startline="9" data-endline="9" class="part c14495">...</p>
          ><p data-startline="11" data-endline="11" class="part c14715">...</p>
          ><center class="c14723">...</center>
        </div>
      </div>
    </div>
  </div>
</section>
<section id="team" class="bdg-sect">...</section>
<footer class="footer-under">...</footer>
<!-- HTML END -->
```

Composability is



Composability is



Developer Journey

1. *Building Login*
2. *Building Mint*
3. *Building Battle*
4. *Building Brag*

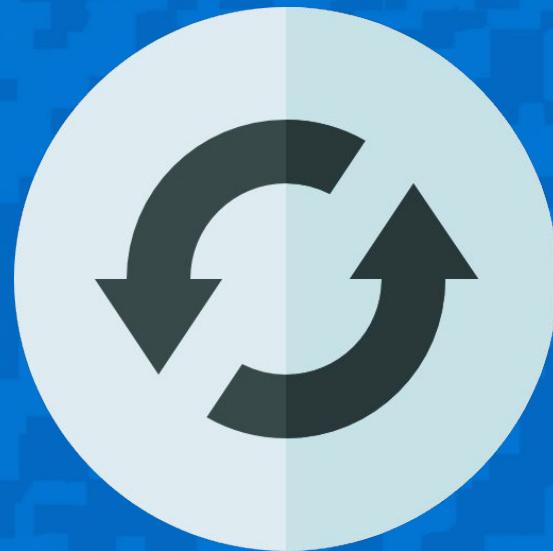
A screenshot of a web browser window displaying a page with a large amount of complex HTML code. The code includes various sections like 'header-section', 'vision', 'manifesto', and 'team', along with numerous

tags and other structural elements. A portion of the code is highlighted in blue, showing parts of a

tag.

Imagine...

- Onchain package manager?
 - Easily extend Twitter-like contracts
- Inspect element for smart contracts?
 - Onchain contract code verification
- Live code editor?
 - In browser auto deploy everything







1. The **User** Journey
2. The **Developer** Journey
3. The **Tech**



1. The **User** Journey
2. The **Developer** Journey
3. The Tech



WIP



Essentials



Dreams



WIP

- Cross chain reads/messaging
- Intents
- Chain specific addresses + onchain config
- Cross chain token standard



Essentials



Dreams



WIP

- **Cross chain reads/messaging**
- Intents
- Chain specific addresses + onchain config
- Cross chain token standard



Essentials



Dreams



WIP

- **Cross chain reads/messaging**
- Intents
- Chain specific addresses + onchain config
- Cross chain token standard



Essentials

Optimistic/speculative cross chain reads

Enables chains to optimistically read from remote chains before their proof has finalized.

Optimistic reads are subjectively finalized by any proof system – eg. ZKPs, TEEs, or even full sync



Dreams



WIP

- Cross chain reads/messaging
- Intents
- Chain specific addresses + onchain config
- Cross chain token standard



Essentials



Dreams



WIP

- Cross chain reads/messaging
- **Intents**
- Chain specific addresses + onchain config
- Cross chain token standard



Essentials



Dreams



WIP

- Cross chain reads/messaging
- **Intents**
- Chain specific addresses + onchain config
- Cross chain token standard



Essentials

ERC-7683



Dreams



WIP

- Cross chain reads/messaging
- Intents
- Chain specific addresses + onchain config
- Cross chain token standard



Essentials



Dreams



WIP

- Cross chain reads/messaging
- Intents
- **Chain specific addresses + onchain config**
- Cross chain token standard



Essentials



Dreams

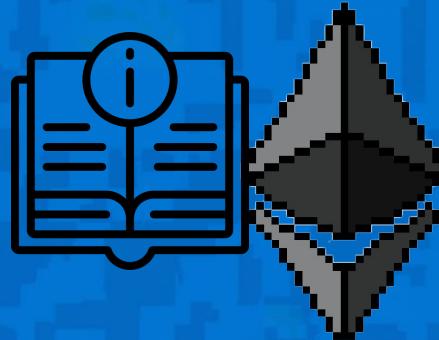


WIP

- Cross chain reads/messaging
- Intents
- **Chain specific addresses + onchain config**
- Cross chain token standard



Essentials



Dreams

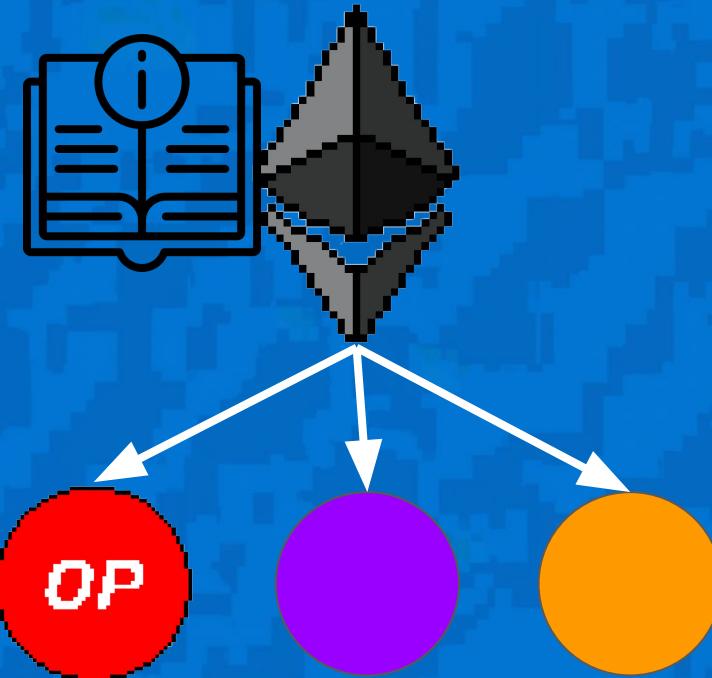


WIP

- Cross chain reads/messaging
- Intents
- **Chain specific addresses + onchain config**
- Cross chain token standard



Essentials



Dreams



WIP

- Cross chain reads/messaging
- Intents
- Chain specific addresses + onchain config
- **Cross chain token standard**



Essentials



Dreams



WIP

- Cross chain reads/messaging
- Intents
- Chain specific addresses + onchain config
- **Cross chain token standard**



Essentials



Dreams

ERC-7802: Crosschain Token Interface

ERCs token, cross-chain



WIP

- Cross chain reads/messaging
- Intents
- Chain specific addresses + onchain config
- Cross chain token standard



Essentials



Dreams



WIP

- Cross chain reads/messaging
- Intents
- Chain specific addresses + onchain config
- Cross chain token standard



Essentials

- Promise libraries & language integration
- Async contract upgrades
- CREATE2 deployment
- Passkey Wallets



Dreams



WIP

- Cross chain reads/messaging
- Intents
- Chain specific addresses + onchain config
- Cross chain token standard



Essentials

- Promise libraries & language integration
- Async contract upgrades
- CREATE2 deployment
- Passkey Wallets



Dreams

- Collaborative contract coding
- Alt-VMs
- Oracalize everything
- Mist browser?? lol
- ...IDK!!!



1. The **User** Journey
2. The **Developer** Journey
3. The **Tech**

OP



OP



OP



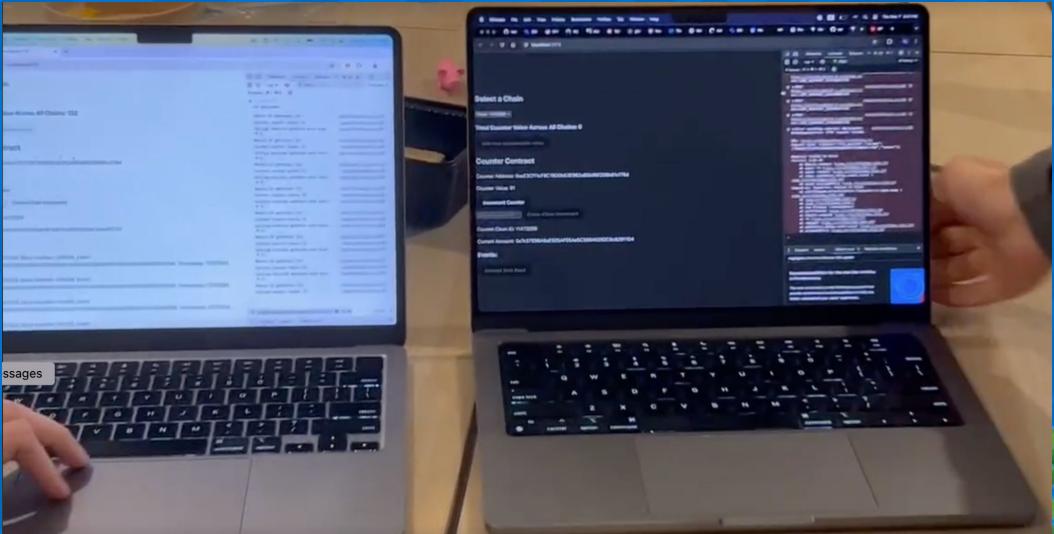
OP



OP

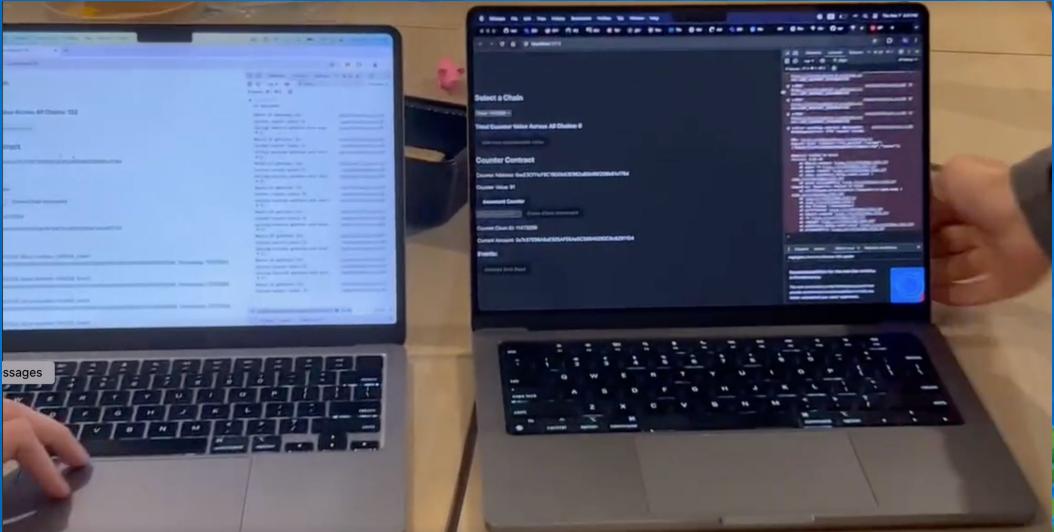


OP



2016 again!

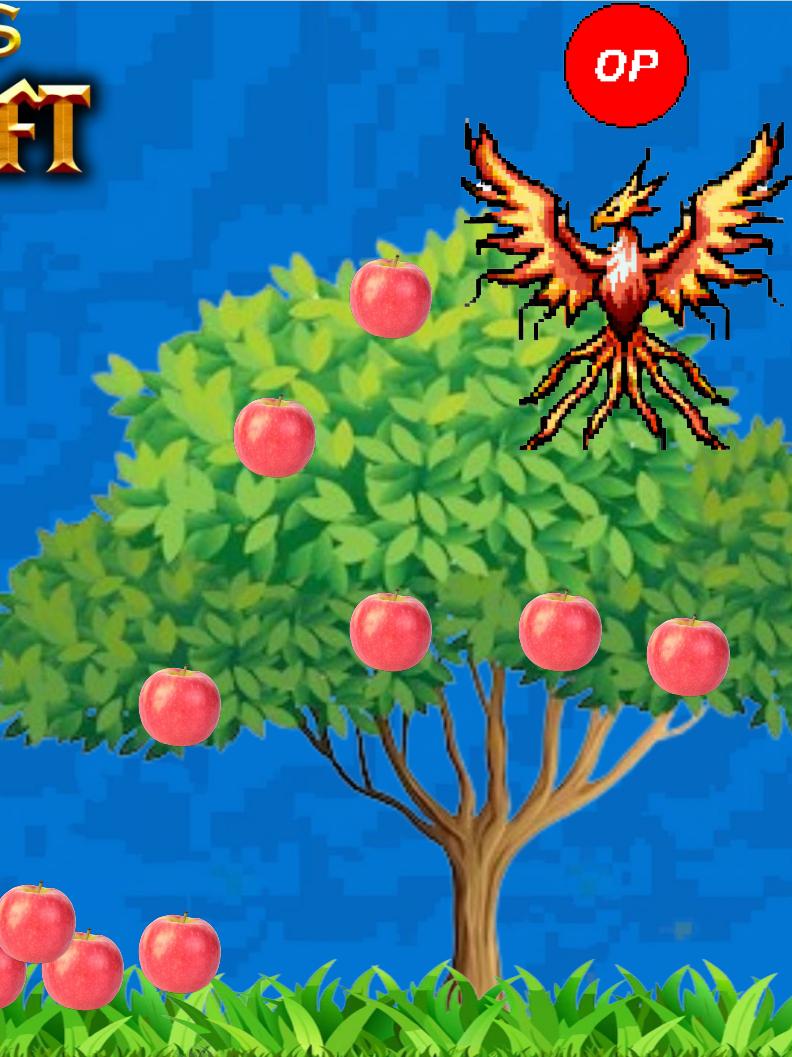
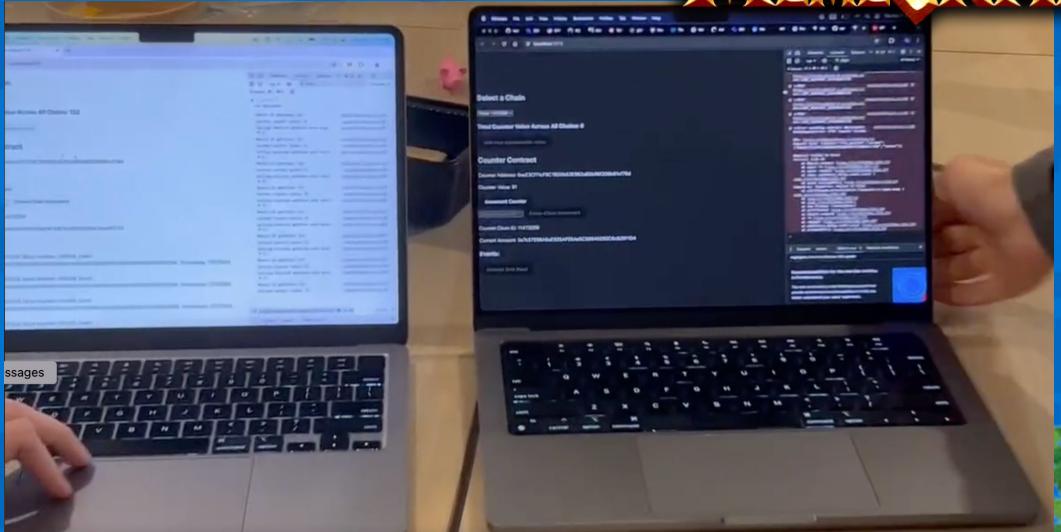
OP

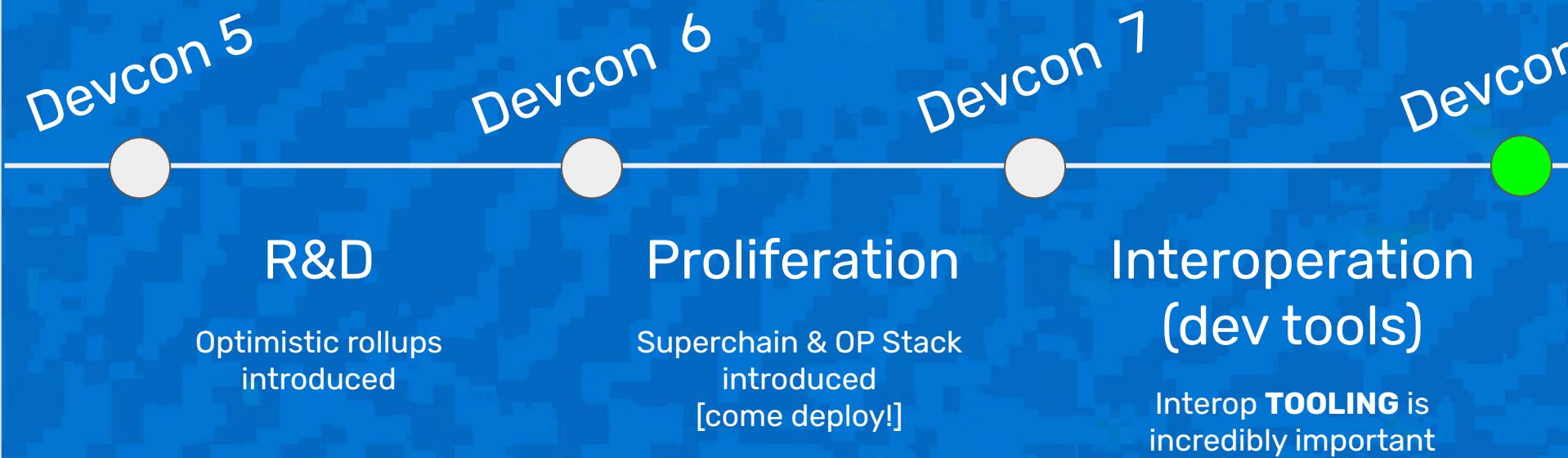


2016 again!

MEMES OF MEMECRAFT

OP





5

Devcon 6

Devcon 7

Devcon 8

R&D

timistic rollups
introduced

Proliferation

Superchain & OP Stack
introduced
[come deploy!]

Interoperation (dev tools)

Interop **TOOLING** is
incredibly important

Devcon 6

Devcon 1

Devcon 8

Proliferation

Superchain & OP Stack
introduced
[come deploy!]

Interoperation (dev tools)

Interop **TOOLING** is
incredibly important

Adoption (8 billion users)

Can't stop till global
adoption

We are still in the trenches



- SHIP interop

We are still in the trenches



- SHIP interop
- SCALE Eth DA

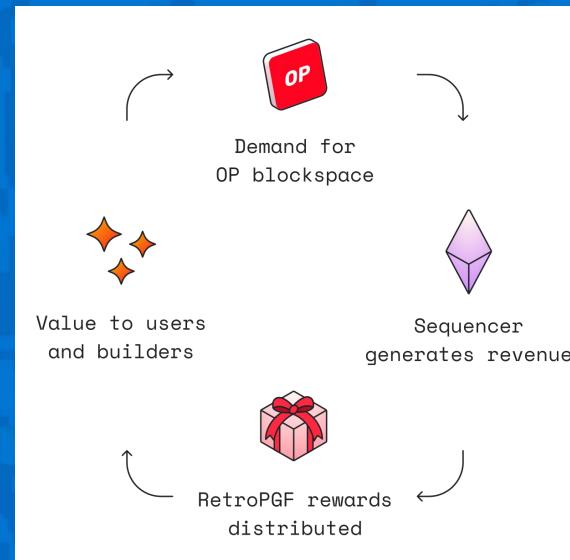
We are still in the trenches

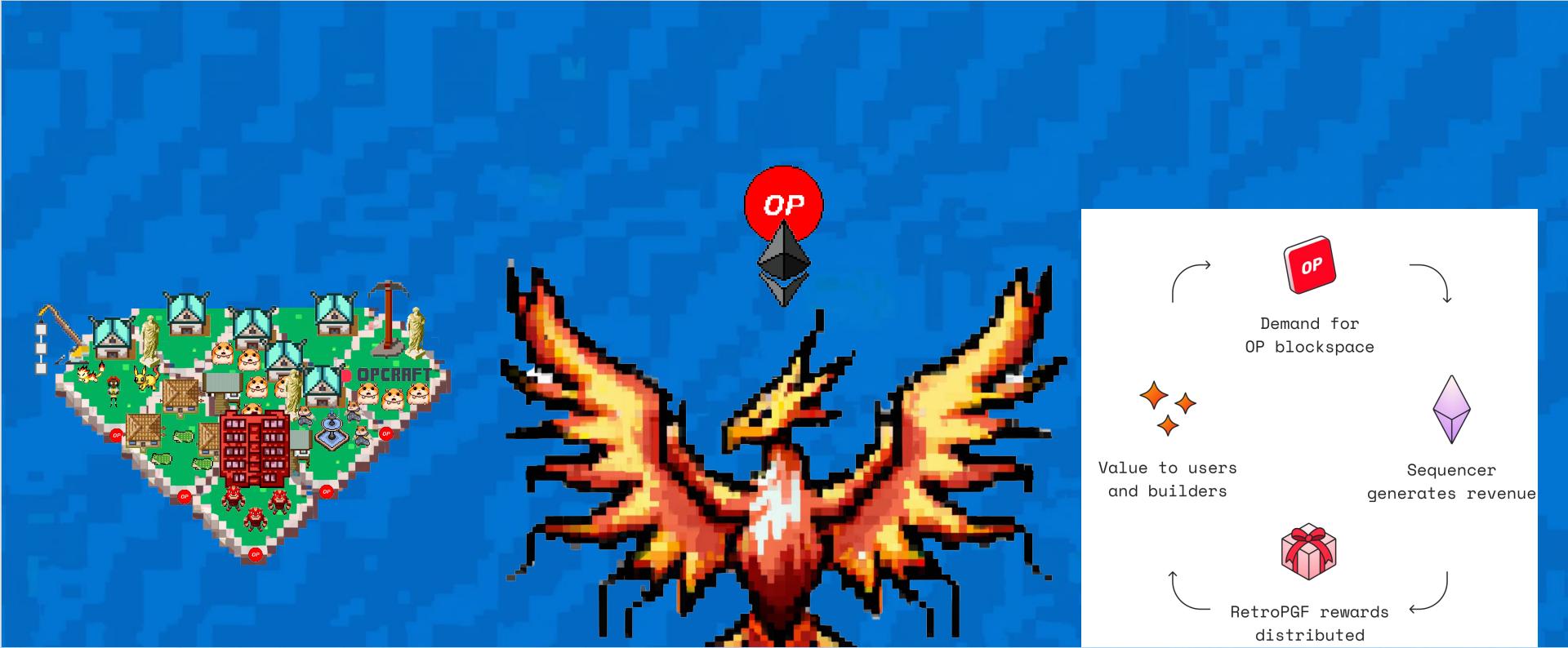


- SHIP interop
- SCALE Eth DA
- DOGFOOD the protocol

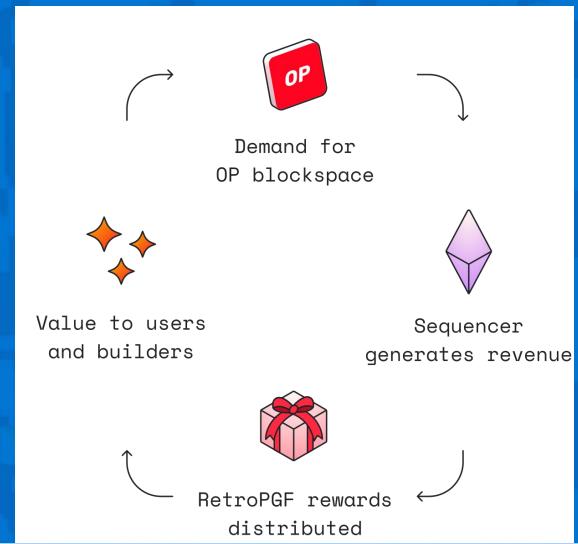


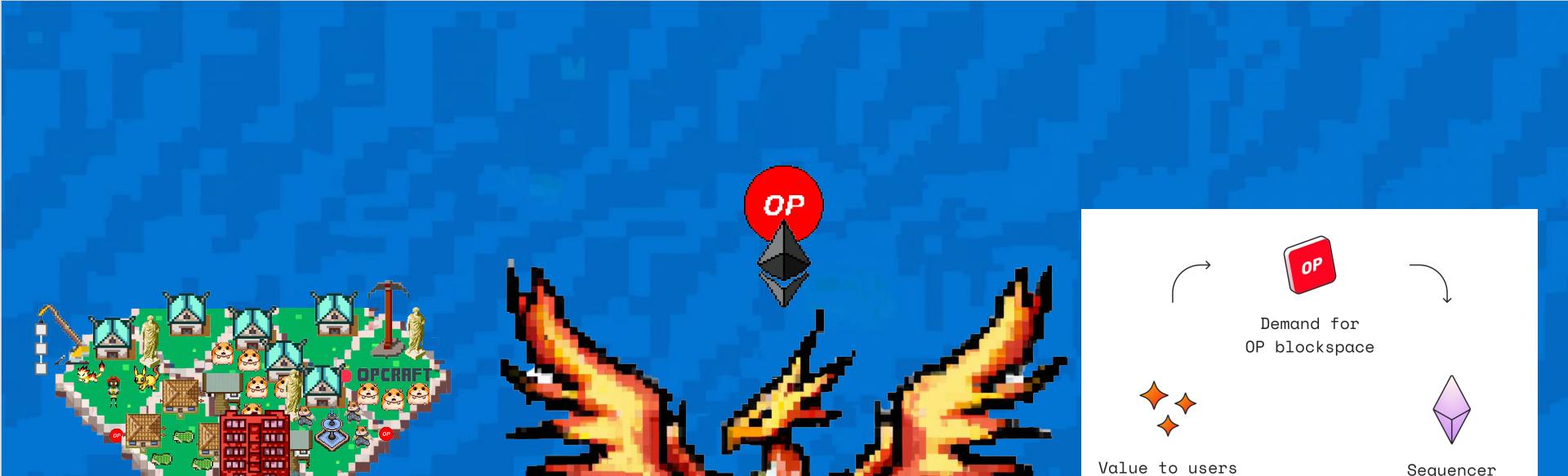






That's why:





That's why:

We're building the...

Optimism Collective

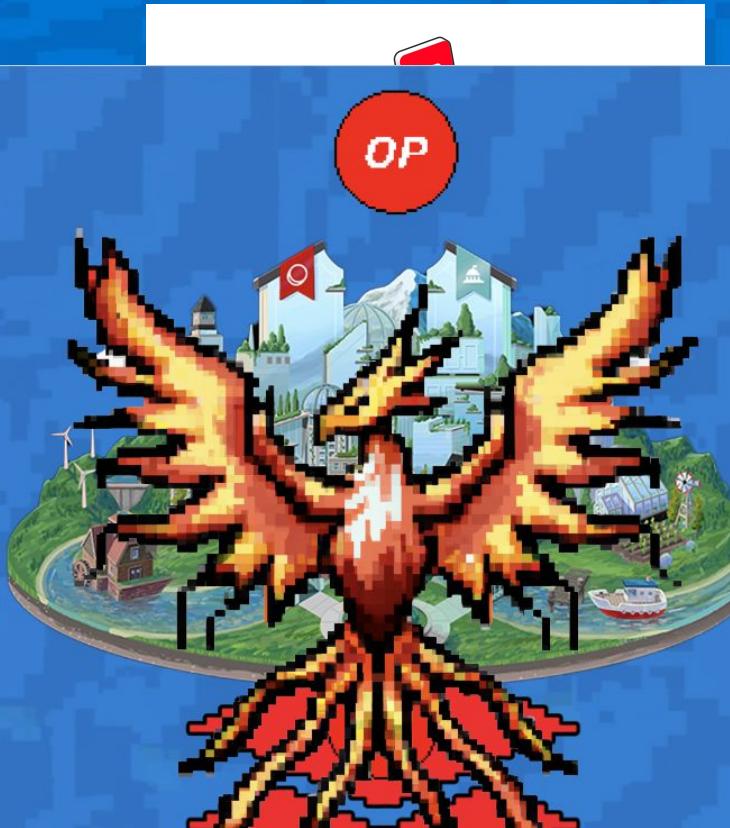


That's why:

We're building the...

 **Optimism Collective** 

Because together we can make
impact=profit and summon...



That's why:

We're building the...

Optimism Collective

Because together we can make
impact=profit and summon...

Ether's Phoenix



That's why:

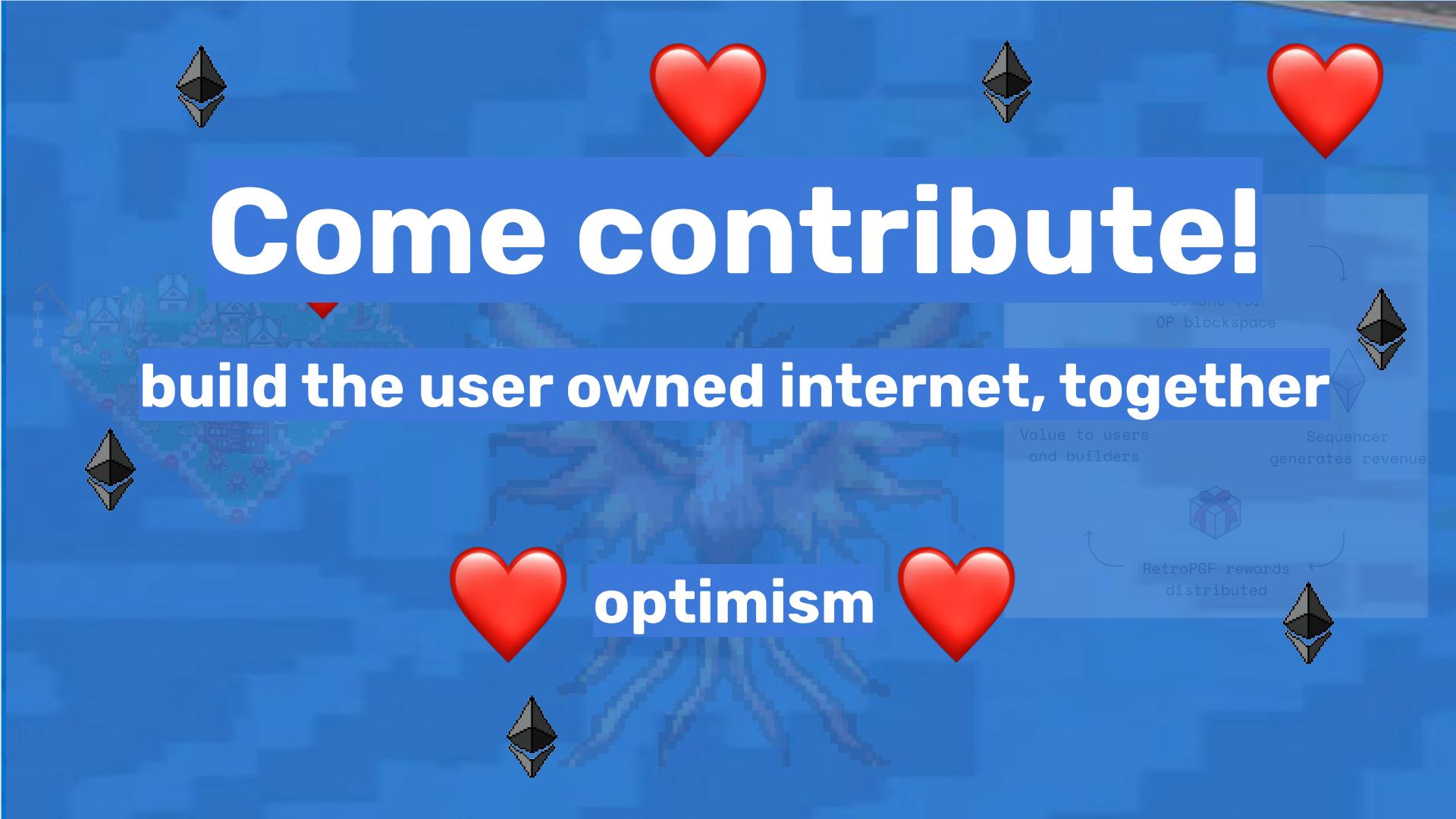
We're building the...

 **Optimism Collective** 

Because together we can make
impact=profit and summon...

 **Ether's Phoenix** 





Come contribute!

build the user owned internet, together

optimism



Value to users
and builders

Sequencer
generates revenue

RetroPGF rewards
distributed



Thank you!

Karl Floersch

optimism

<https://optimism.io/>