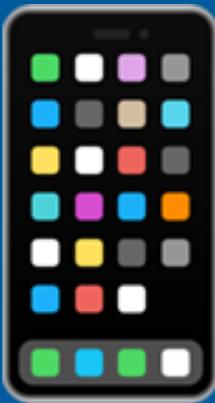


# NATIVE MOBILE ETHEREUM DAPPS

PAUL COWGILL



TASIT

# OUTLINE

WHY

WHAT ABOUT CENSORSHIP?

HOW

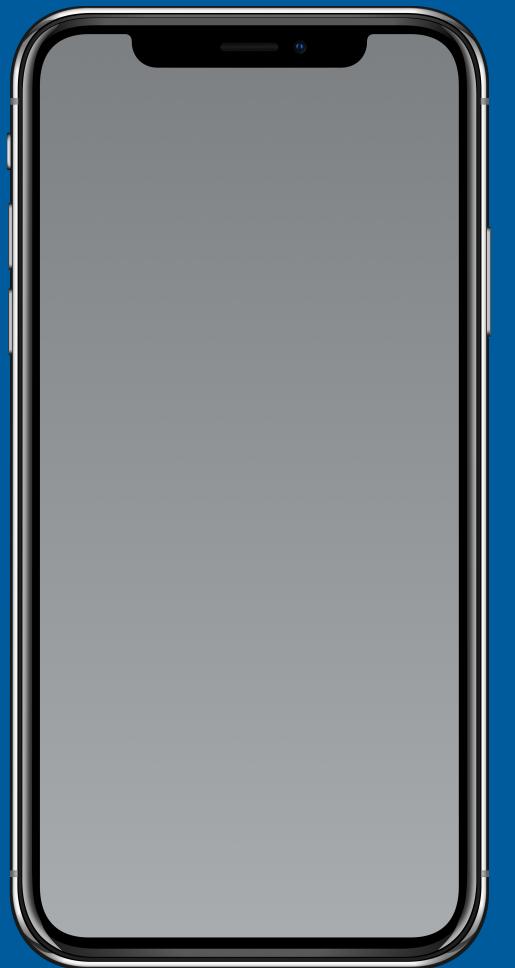
# WHO AM I?

@PAULCOWGILL

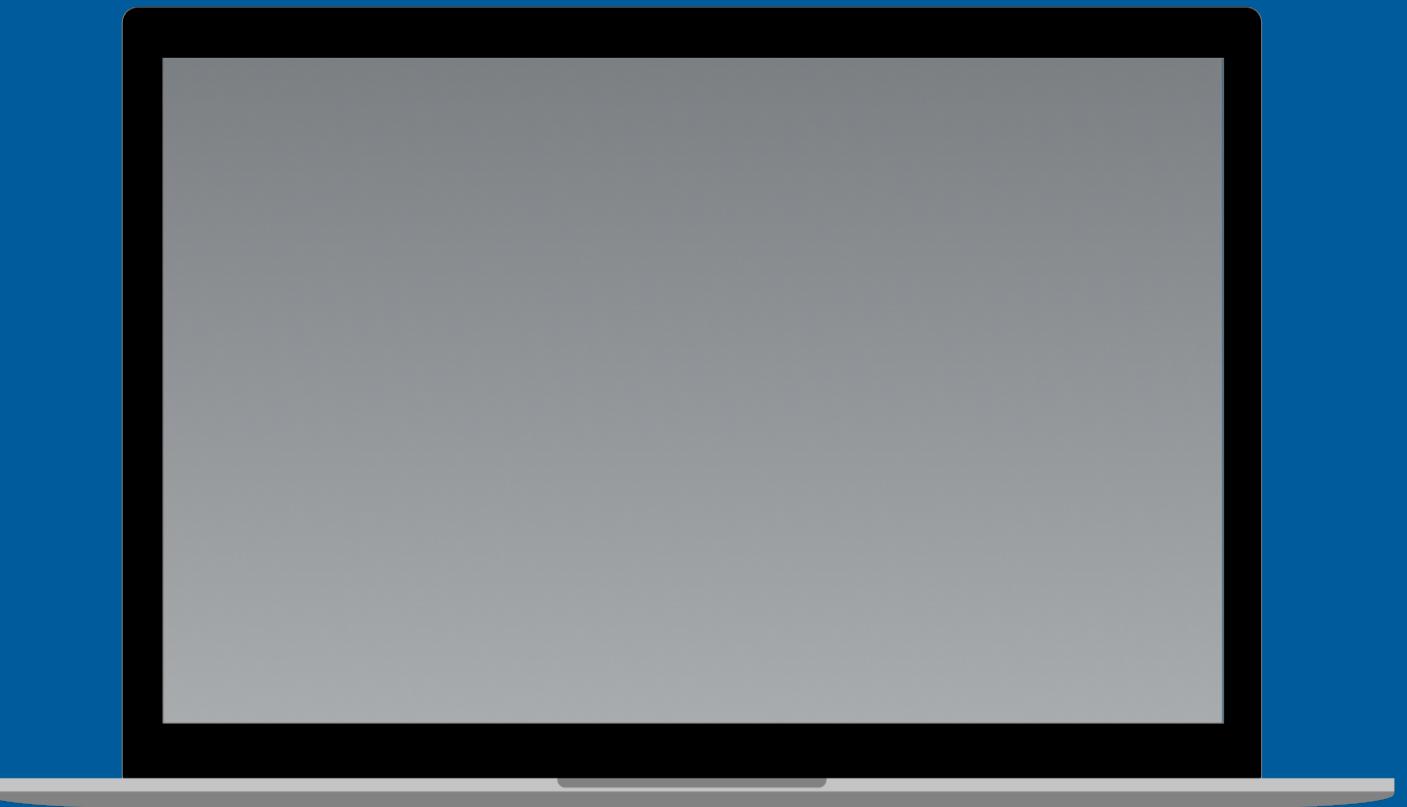


# WHY

# PHONES



# LAPTOPS



# NATIVE VS. WEB

NATIVE MOBILE APPS ARE THE CURRENT WINNER IN WEB 2.

A SLIGHTLY LARGER SWATH OF **DISENGAGED USERS** MIGHT TRY A  
MOBILE WEB APP FIRST.

BUT THE 'STICKY' **POWER USERS** PREFER NATIVE MOBILE APPS.

# WHEN TO USE NATIVE MOBILE?

- YOUNG AND/OR GLOBAL USER BASE
- HIGH FREQUENCY OR REAL TIME
- LOW(ISSH) STAKES-ALTHOUGH THIS IS CHANGING
  - BIOMETRIC AUTH
- CONVENIENT. RELIABLE. LOCAL DATA PERSISTENCE
  - USES CAMERA OR GPS

**WHAT ABOUT  
CENSORSHIP?**

# MAKE IT IMPOSSIBLE

USE CONTRACT-BASED ACCOUNTS (OR WALLETCONNECT)

# SCENARIOS WITH REACT NATIVE

APPLE TAKEDOWN?  
WORKS ON ANDROID

GOOGLE TAKEDOWN?  
WORKS ON IOS

GONE FROM BOTH PLATFORMS?  
react-native-web OR SHARING CODE WITH A REACT  
CODEBASE

**BIGTECHCO POLICIES CHANGE ALONG  
WITH THE ZEITGEIST.**

# HOW

# BUILDING NATIVE MOBILE DAPPS

SIMPLE UX WITH NO DECENTRALIZATION TRADEOFF

**IF YOUR PROJECT DISAPPEARS ONE DAY, THE USER SHOULD BE VERY DISAPPOINTED BUT ABLE TO USE THE CONTRACTS IN ANOTHER WAY.**

LET THE USER ARBITRARILY UPGRADE  
THE SECURITY WITH HARDWARE  
WALLETS, DEDICATED 'WALLETS', ETC.,  
BUT BURY THOSE OPTIONS IN AN  
ADVANCED MENU.

# WHAT IS NEEDED TO DO THIS WELL?

## ONBOARDING

- › IN-DAPP FIAT ONRAMPS
- › IN-DAPP WALLETS (BURNER AND CONTRACT-BASED)

## ONGOING UX

- › ENS
- › THE GRAPH
- › OPTIMISTIC UI
- › META-TRANSACTIONS

## MOBILE STUFF

- › DEEP LINKING
- › BIOMETRIC AUTH
- › PUSH NOTIFICATIONS



# ONBOARDING

# IN-DAPP BURNER WALLETS

TO ATTRACT THE NEXT WAVE OF ETHEREUM USERS...

...WE SHOULD ASSUME THAT USERS DON'T HAVE A WALLET YET.

# A SINGLE, FLUID UX

```
import { Account } from "tasit-sdk";
const burnerWallet = await Account.create();
const { address } = burnerWallet;
console.log(address); // '0x...'
// ...
```

# ANOTHER NEW ACCOUNT?

FOR NEW USERS. THEY DON'T HAVE A PREEXISTING ACCOUNT ANYWAY. AND THIS COULD EVOLVE INTO THEIR MAIN ACCOUNT.

A NEW ACCOUNT IS BETTER FOR OPSEC AND ACCOUNTING REASONS.

# CONTRACT-BASED ACCOUNTS PT. 1

```
const { address: burnerWalletAddress } = burnerWallet;
const username = "paul";
const contractKind = "gnosis-safe"; // | "argent" | "abridged" | "authereum" | ...

const create2ContractBasedAccount = await Account.upgrade({
  account: burnerWalletAddress,
  username: username,
  contractKind: contractKind
});
```

# CONTRACT-BASED ACCOUNTS PT. 2

```
console.log(create2ContractBasedAccount.address); // Send it an ERC20 token!  
const contractBasedAccount = await create2ContractBasedAccount.deploy();  
console.log(contractBasedAccount.ensName); // paul.{NAMEOFTHISDAPP}.eth
```

# 100 ACCOUNTS PER USER

(SOME UPGRADED INTO CONTRACTS, SOME NOT)

INVESTMENT. SAVINGS. GAMING. WORK. HOBBIES. SIDE PROJECTS.  
SOCIAL. ...

**SHOULD WE WIRE THIS UP WITH A  
'PARENT' CONTRACT-BASED ACCOUNT?**

# IN-DAPP FIAT ONRAMPS



**RampConnect**  
@RampConnect

RampConnect: A single Javascript library for all fiat on-ramp providers  
[@RampNetwork](#) [@sendwyre](#) [@MoonPayHQ](#)  
[@carbon\\_money](#) [@SimplexCC](#)

Coming soon...

1 2:15 PM - Dec 12, 2019

[See RampConnect's other Tweets](#)

# ONGOING UX

# OPTIMISTIC UI PT. 1

```
import { Contract } from "tasit-sdk";
const { NFT } = Contract;

const contractAddress = "0x0E86...333";
const contract = new NFT(contractAddress);

// No await!!!
const action = contract.safeTransferFrom(/*...*/);
```

# OPTIMISTIC UI PT. 2

```
action.on("error", errorListener);
action.on("oneConfirmation", goodSignListener);
action.on("enoughConfirmations", successListener);
action.send() // broadcast

// Do optimistic UI updates immediately, while making sure
// to update the UI again when there are enough
// confirmations for your use case
// ...
```



**WAIT, BUT I LIKE AWAIT.**

# THE GRAPH

```
const action = contract.safeTransferfrom(/*...*/);
action.on("error", errorListener);
action.on("presentInTheGraph", successListener);
action.send() // broadcast
// ...
```

(AND OF COURSE MORE STANDARD THINGS LIKE QUERYING FOR A LOT OF INDEXED EVENTS)

# ENS

SENDING

SOMEBODYELSE.ETH

(ETHERS.JS UNDER THE HOOD)

RECEIVING

PAUL.UNISWAP.ETH

PAUL.GITCOIN.ETH

PAUL.TASIT.ETH

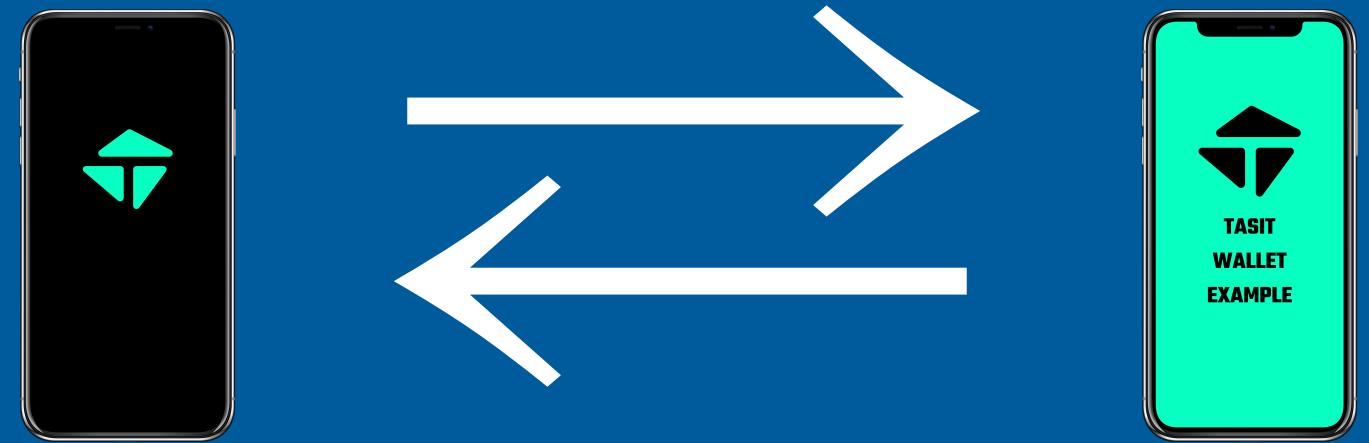
# META-TRANSACTIONS

```
const action = contract.myFavoriteMethod(/*...*/);
action.on("error", errorListener);
action.on("enoughConfirmations", successListener);
action.sendForFree(); // meta-tx broadcast
```

```
// ...
```

**MOBILE-  
SPECIFIC  
FEATURES**

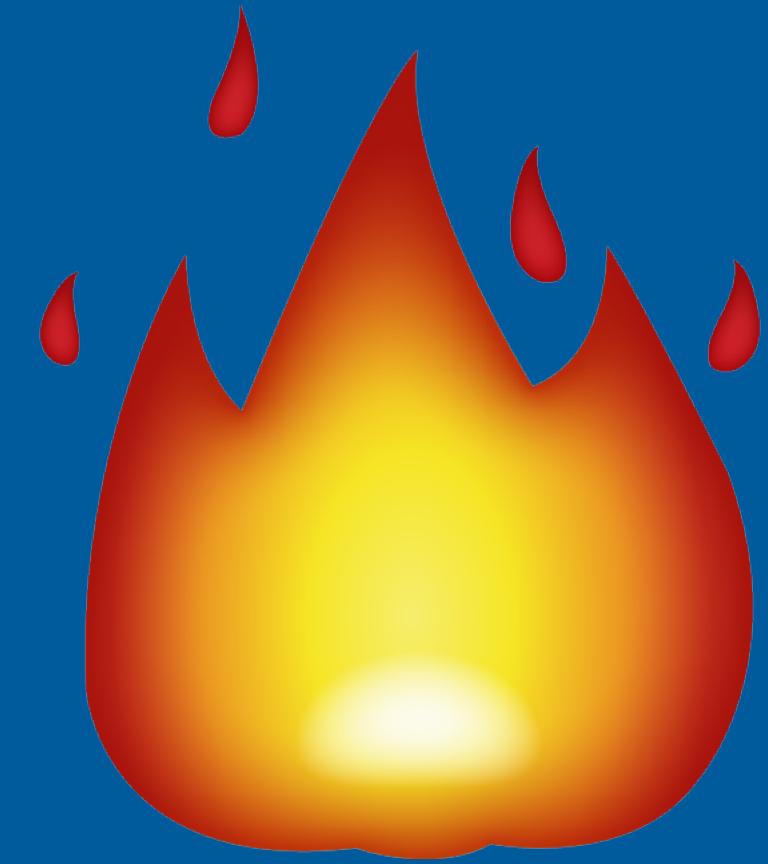
# DEEPLINKING



# BIOMETRIC AUTH & PUSH NOTIFICATIONS

{COMING SOON}

# BOILING THE OCEAN?



**DON'T USE TOO MANY BROWSER APIs  
IN YOUR JS SDK.**

# SCALING AND PRIVACY

# SCALING AND PRIVACY

L2

STATE CHANNELS, (OPTIMISTIC | ZK) ROLLUP

ETH2

PRIVACY

# WRAPPING UP

# DAPPS ARE USEFUL!

DAPPS **ALREADY** DELIVER SIGNIFICANT VALUE AND 'A-HA' MOMENTS TO CRYPTO-NATIVE 'INNOVATORS'.

THESE APPS WOULD BE USEFUL FOR MAINSTREAM PEOPLE TOO IF PACKAGED CORRECTLY.

# UX IS THE PROBLEM

GREAT WEB 2 PRODUCT TEAMS CAN FIX THIS  
IF THEY'RE GIVEN THE RIGHT TOOLS.

# WHERE TO GO FROM HERE

WE'RE ACTIVELY DOING **USER INTERVIEWS** AND **DEVELOPER  
INTERVIEWS** TO SHAPE WHICH FEATURES TO PRIORITIZE.

@PAULCOWGILL

# THANKS



**ecosystem  
support  
program**





WEBSITE

TASIT.IO



TWITTER

@TASITPROJECT



MEDIUM

MEDIUM.COM/TASIT



GITHUB

GITHUB.COM/  
TASITLABS



DISCORD

BIT.LY/TASIT-  
DISCORD



FEATURES

FEEDBACK.TASIT.IO



DOCS

DOCS.TASIT.IO



TELEGRAM

T.ME/TASITPROJECT