Public-Private Hybrid Rollups

The Next Ethereum Frontier

Aztec Labs

Privacy is a Best Practice

- “Hey organizations, we solved scaling, look at Ethereum again”

Privacy is a Best Practice

- “Hey organizations, we solved scaling, look at Ethereum again”
- They do the same toy experiments they always have

Privacy is a Best Practice

- “Hey organizations, we solved scaling, look at Ethereum again”
- They do the same toy experiments they always have
- Real business stays where there is real privacy

What Does Privacy Mean For Ethereum?

“Someone did something to some state in some function of some contract”

“Something Happened”

Ethereum Needs Privacy

- Ethereum has to balance innovation speed and scope on layer 1

Ethereum Needs Privacy

- Ethereum has to balance innovation speed and scope on layer 1
- We need to lead the way with private rollups

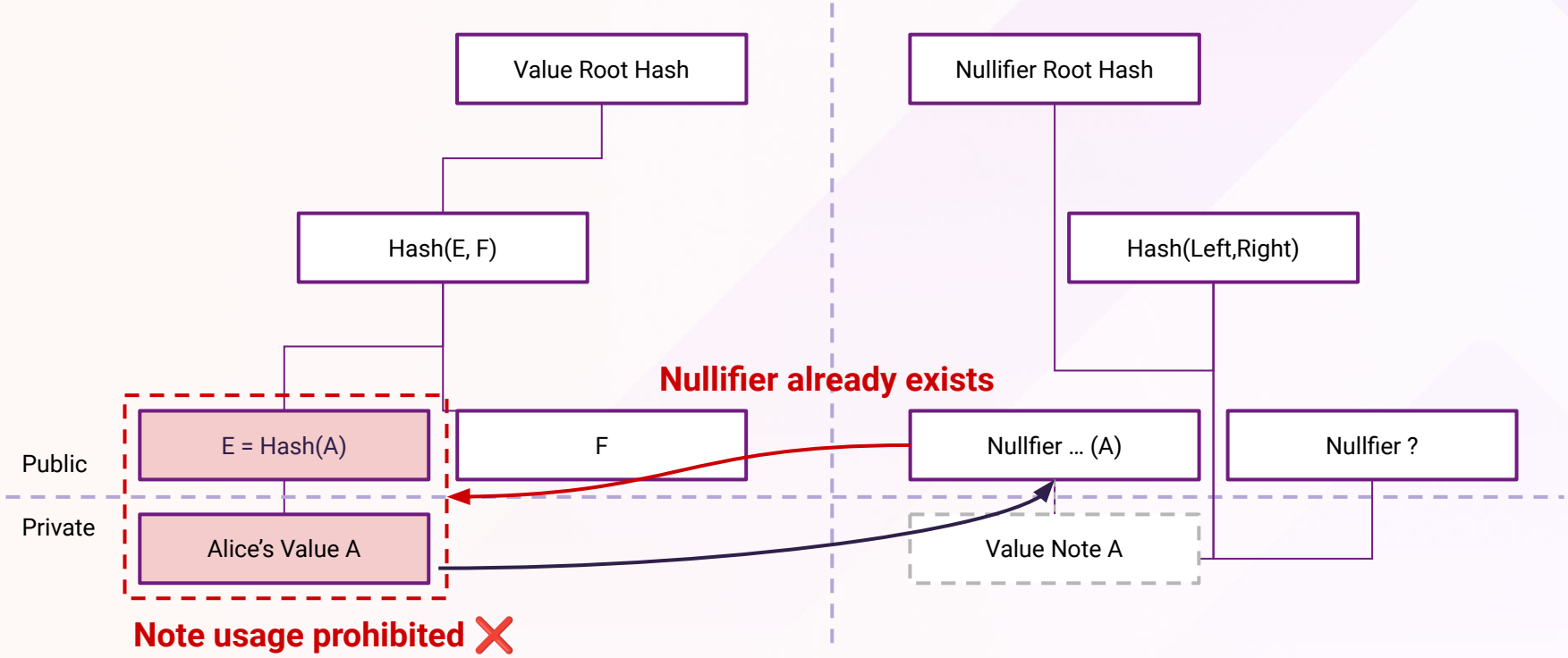
Quite Caveat

- This is a technical talk - I won’t be talking about which jurisdiction to worry about when it comes to compliant privacy.

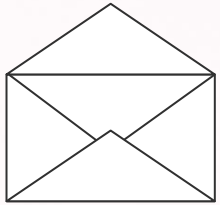
- We have our **amazing crypto-native legal team** here talking about how to navigate real-world deployments!


AZTEC
PRIVACY
EVM
imgflip.com

Bitcoin-style Notes + Nullifiers + ZK!



What's In A Private Note?



```
struct CardNote {
    strength: u32,
    points: u32,
    owner: Field,
    randomness: Field,
    // note header
    contract_address: AztecAddress,
    uniqueness_nonce: Field,
    storage_slot: Field,
}
```

What Does Public Mean?

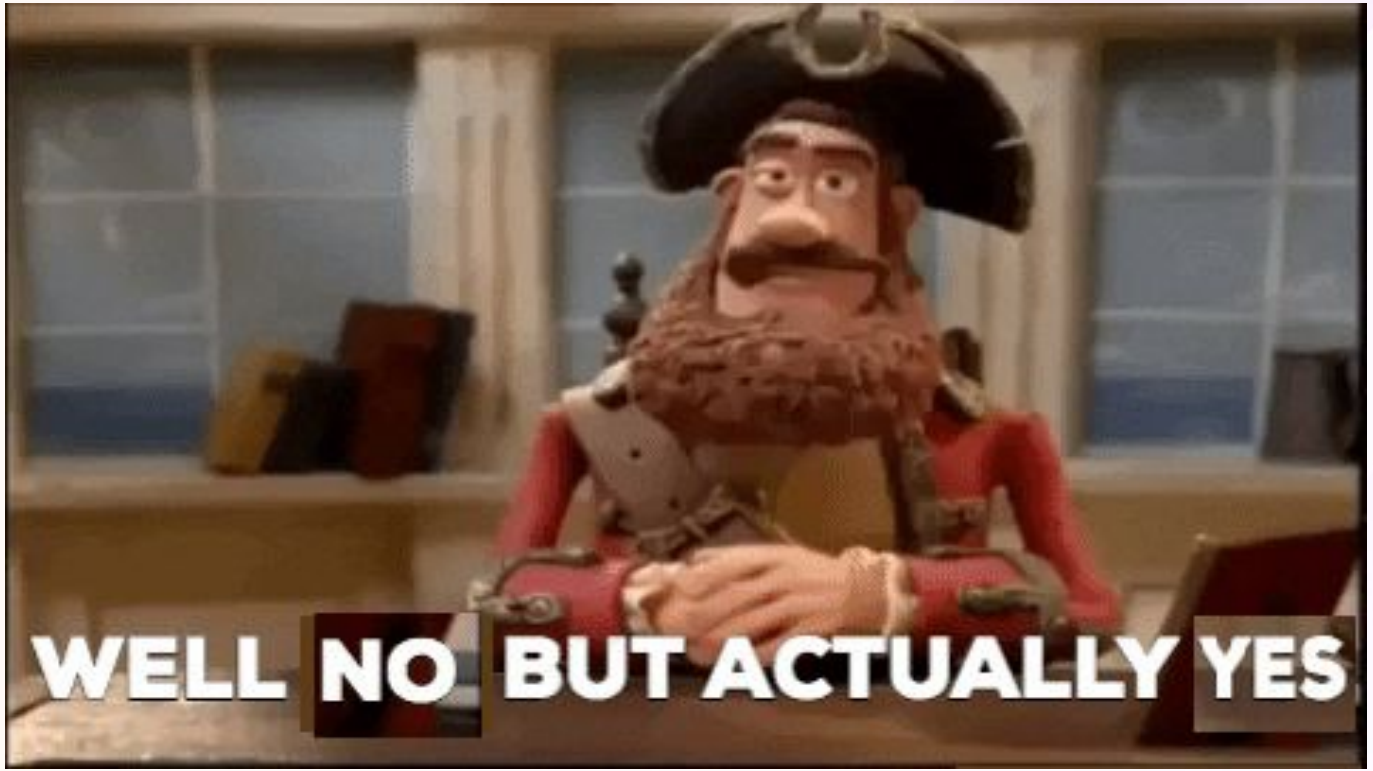
- For this talk - anything that doesn’t neatly fit into “something happened”

What Does Public Mean?

- For this talk - anything that doesn't neatly fit into "something happened"

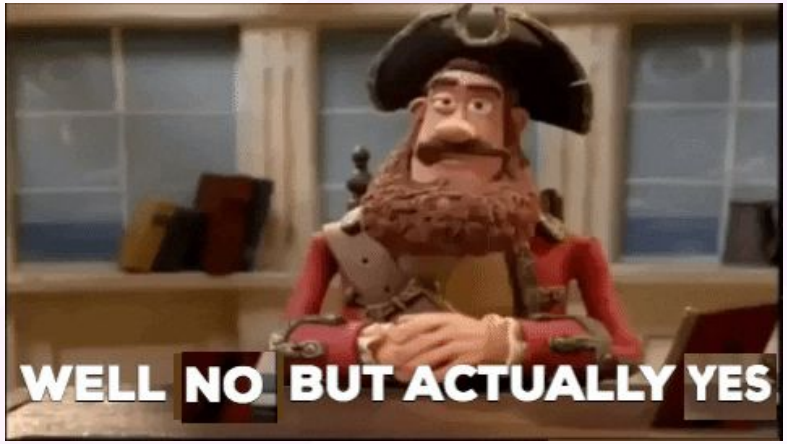
- This means that **anyone other than the parties transacting** gleans ANY information beyond

Can Everything Be Private?

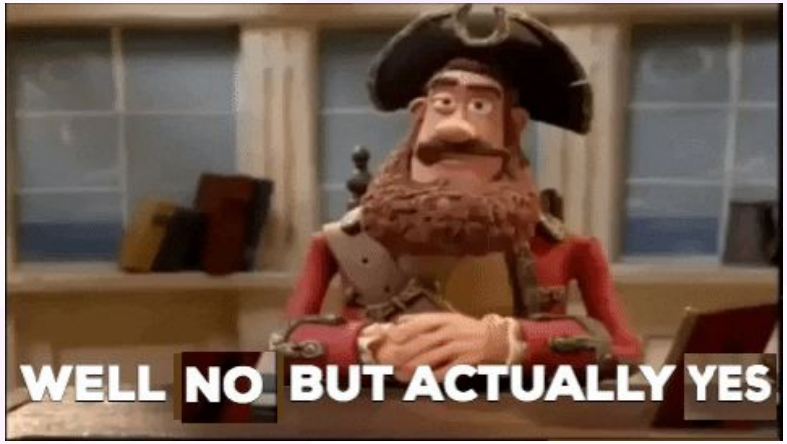


Can Everything Be Private?

- **No** - if we take the approach of notes and nullifiers and ZK, it lends itself easily to peer-to-peer but not multiparty e.g. uniswap



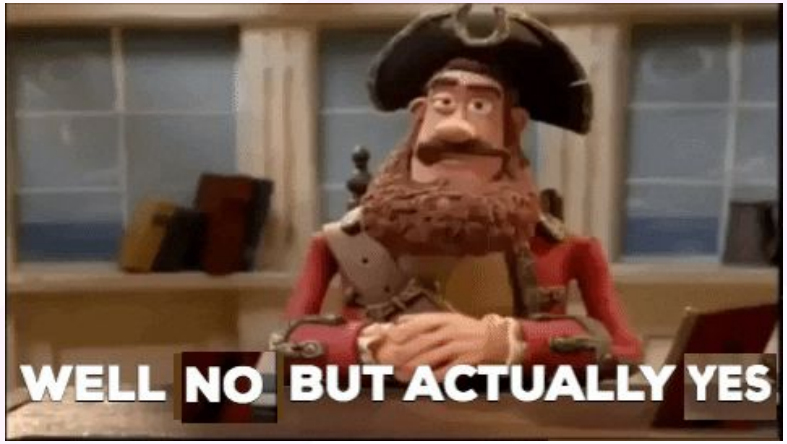
Can Everything Be Private?



- **No** - if we take the approach of notes and nullifiers and ZK, it lends itself easily to peer-to-peer but not multiparty e.g. uniswap

- **But** - we can still have private entry points to public shared state, relaxing ‘something happened’ to be ‘someone did this specific trade, but we don’t know who’

Can Everything Be Private?



- **No** - if we take the approach of notes and nullifiers and ZK, it lends itself easily to peer-to-peer but not multiparty e.g. uniswap

- **But** - we can still have private entry points to public shared state, relaxing ‘something happened’ to be ‘someone did this specific trade, but we don’t know who’

- **Actually yes** - if we look outside of just ZK, we can get privacy for the ‘last mile’, (mostly) preserving the property ‘something happened’

A Hard *No*

- The chain is being tracked more than ever...

- No privacy compromises user safety and business edge

A Hard *No*

- Really there is no hard no for holistic privacy on **any** chain

A Hard *No*

- Really there is no hard no for holistic privacy on **any** chain

- That being said, even these so-called ZK rollups - looking at you, ZKSync - make basic privacy hard, and holistic privacy very hard!

A Nice *But*

A Nice *But*

- In Aztec, identity is always private via private account abstraction

A Nice *But*

- In Aztec, identity is always private via private account abstraction

- Aztec uses the same smart contract language/framework in both public and private functions of a given contract

A Nice *But*

- In Aztec, identity is always private via private account abstraction

- Aztec uses the same smart contract language/framework in both public and private functions of a given contract

- Transactions can be a mix of private and public, sharing state. The whole thing either succeeds or reverts

A Nice *But*

```
fn sum(x : u32, y : u32) -> u32 {
  x + y
}
```

Noir

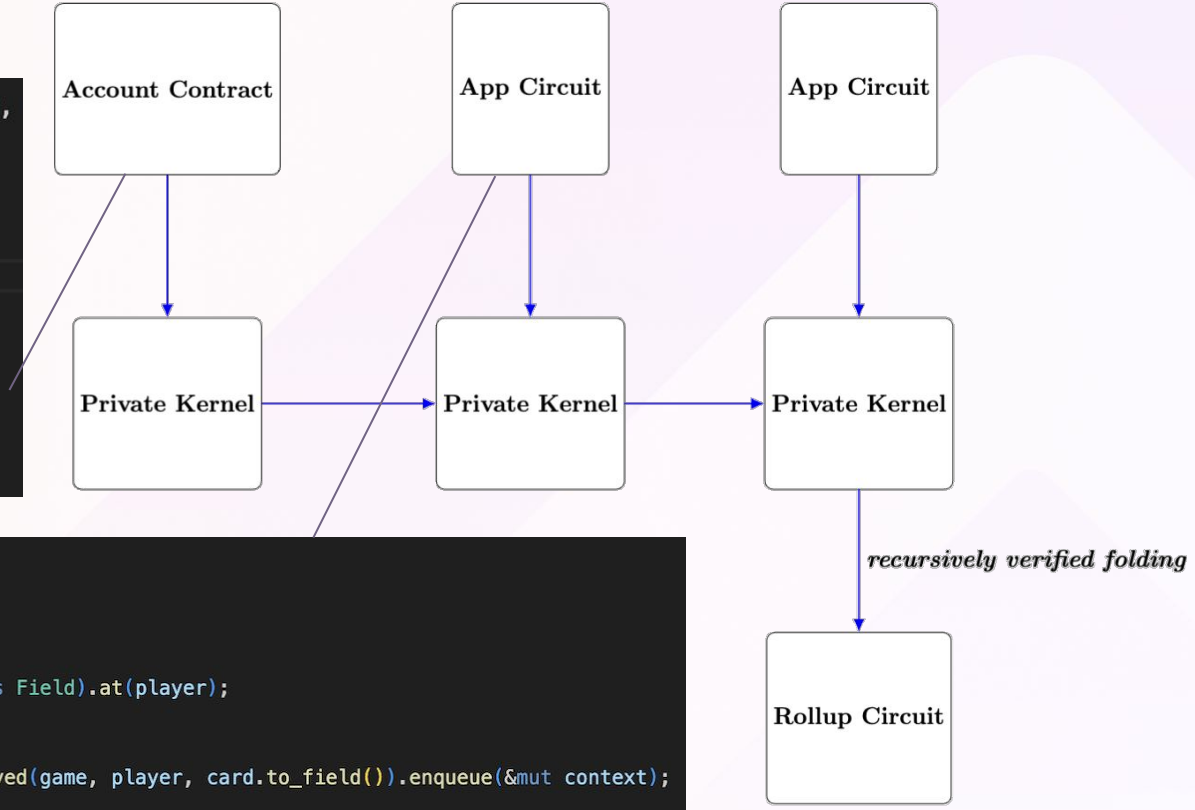
```
fn sum(x : u32, y : u32) -> u32 {
  x + y
}
```

A Nice *But*

```
pub fn entrypoint(self, app_payload: AppPayload,
    let valid_fn = self.is_valid_impl;

    let fee_hash = fee_payload.hash();
    assert(valid_fn(self.context, fee_hash));
    fee_payload.execute_calls(self.context);
    self.context.end_setup();

    let app_hash = app_payload.hash();
    assert(valid_fn(self.context, app_hash));
    app_payload.execute_calls(self.context);
}
```




```
#[aztec(private)]
fn play_card(game: u32, card: Card) {
    let player = context.msg_sender();

    let mut game_deck = storage.game_decks.at(game as Field).at(player);
    game_deck.remove_cards([card]);

    CardGame::at(context.this_address()).on_card_played(game, player, card.to_field()).enqueue(&mut context);
}
```

Getting To *Yes*

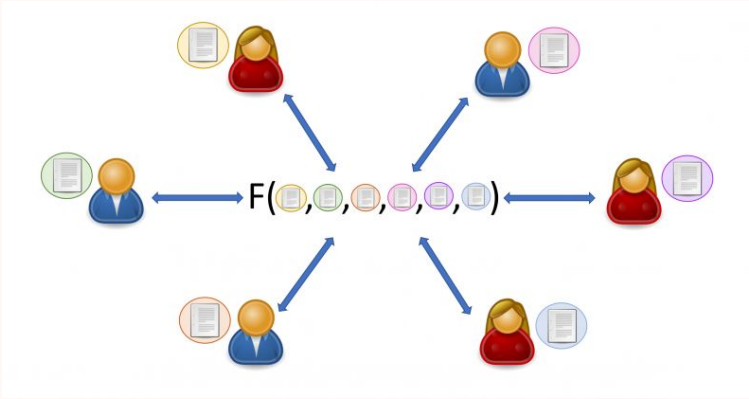
- Aztec makes public state easy to access from private computation as it is a practical way to greatly improve privacy status quo

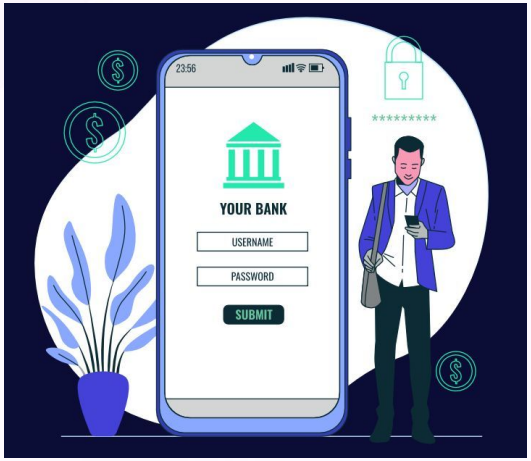
Getting To *Yes*

- Aztec makes public state easy to access from private computation as it is a practical way to greatly improve privacy status quo

- The gold standard though is to do everything with privacy primitives - the chain being purely “something happened”

Getting To *Yes*





Thank you for listening!

- Discord → Discord.Aztec.Network
- Research → Forum.Aztec.Network
- Documentation → Docs.Aztec.Network

Follow me on 𝕏
@digitalsagell

Aztec Labs

Aztec Labs