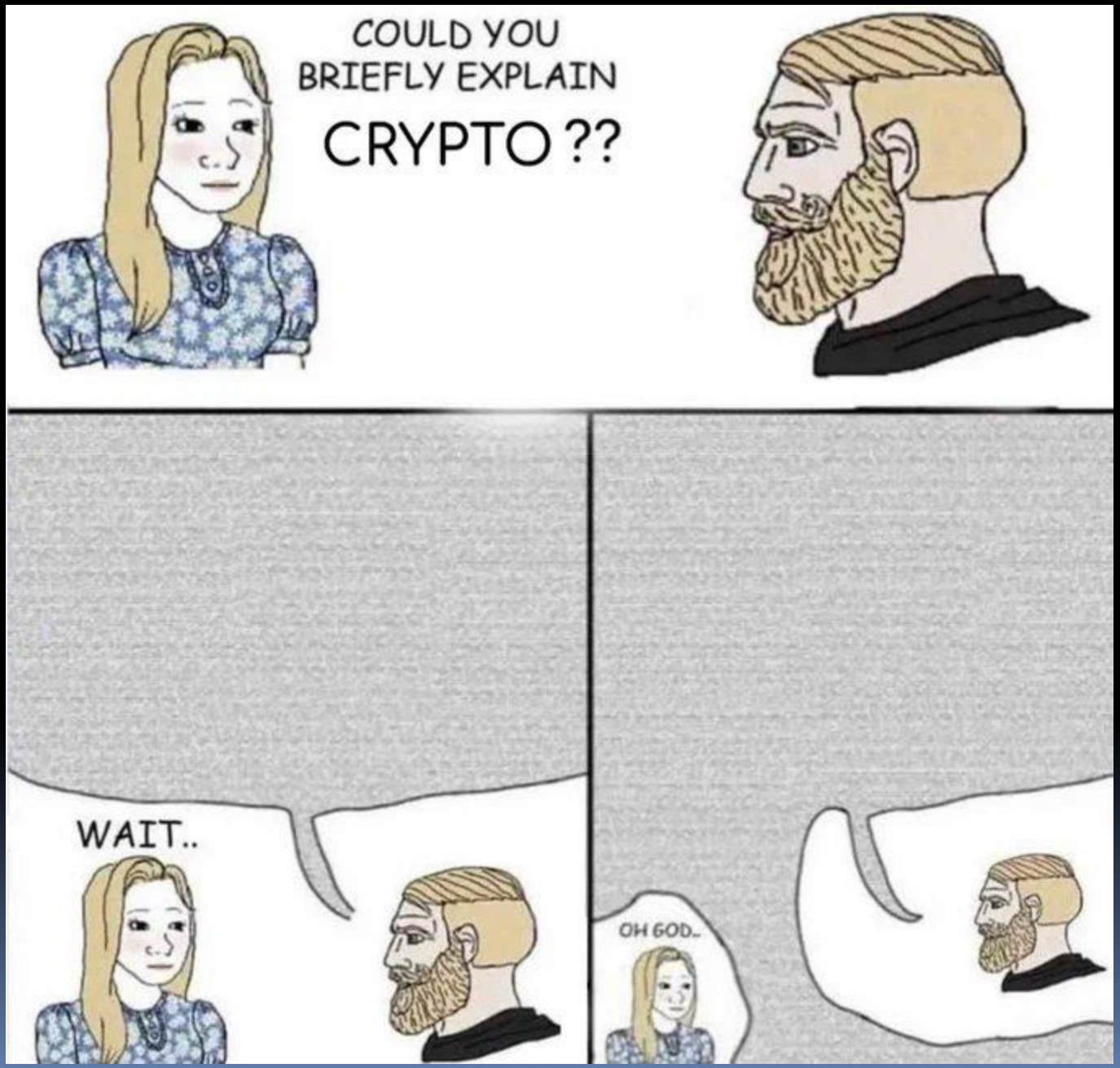
dr laurence e. day / @functiOnZer0; lse pp426 guest lecture

A Walk Through The Dark Forest:
Smart Contracts, Security, Privacy, and You


COULD YOU
BRIEFLY EXPLAIN
CRYPTO ??
WAIT..
OH GOD..

What You'll Sleep Through

- What *are* smart contracts?
- How do we *write* them?
- Why do we *want* them?

- Exploiting flawed smart contracts
- Best practices to mitigate risks

- Privacy tools and black-hat usage
- Immune systems for privacy tools

- Evolving issues: intents/base-level privacy

A Warning To The Wakeful

What I'm presenting is Ethereum-centric

There are several other blockchains which support smart contracts in different ways: e.g. Solana, Cardano, other stupid names

Principles/concerns the same throughout

Boring primer, *does* get more interesting!

What *Are* Smart Contracts?

Simply put, they're programs on blockchains

A common analogy is a vending machine:

- Put money in, press a button, get item

More specifically, a smart contract is a set of callable functions and a self-contained state stored at a given blockchain address

Nobody possesses the private key to a smart contract, but they *can* be owned by others

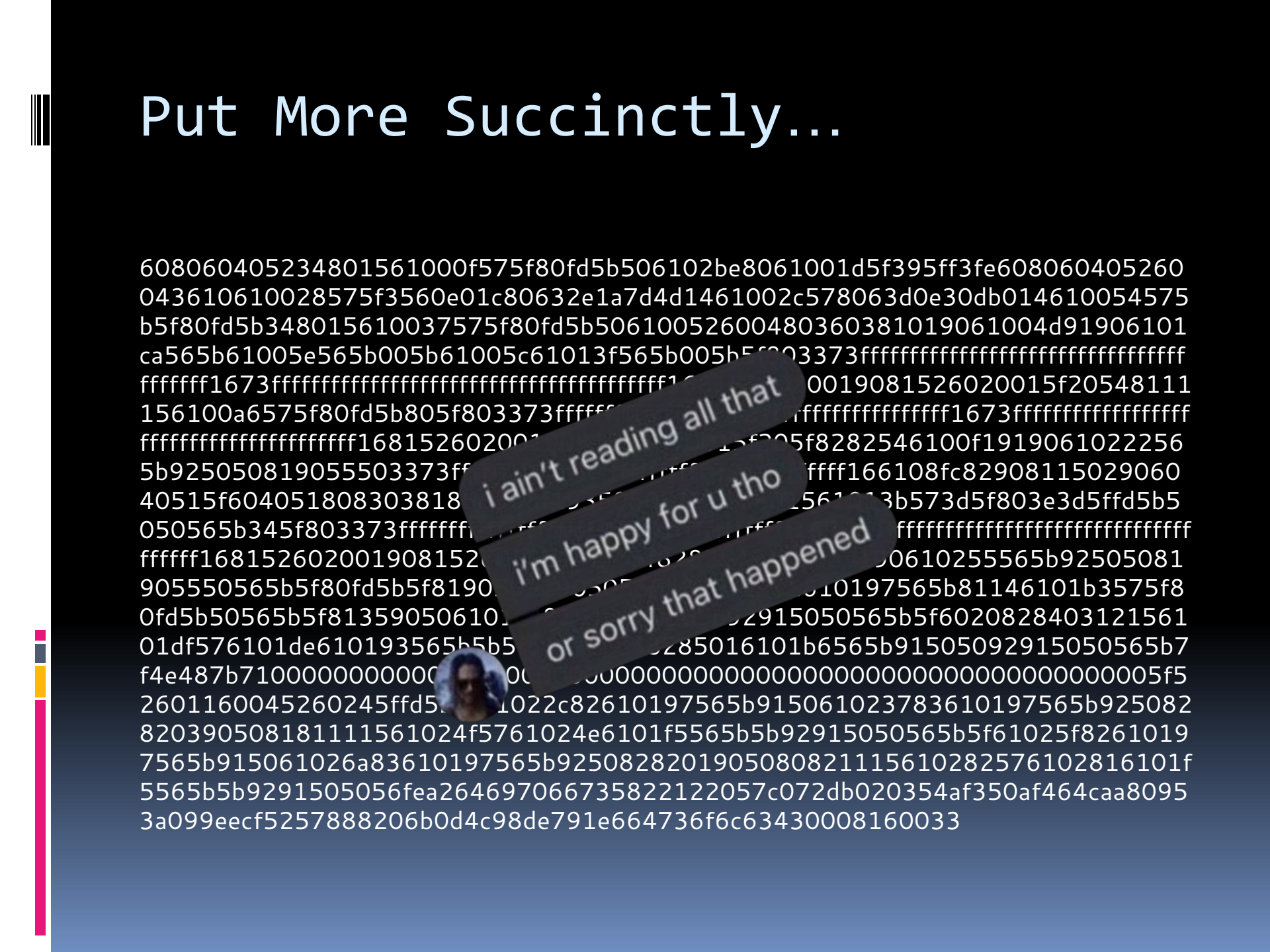
A Simple Example

```
pragma solidity =0.8.22

contract PiggyBank {

  mapping (address => uint) balances;

  function deposit() external payable {
    balances[msg.sender] += msg.value;
  }

  function withdraw(uint amount) external {
    require(amount <= balances[msg.sender]);
    balances[msg.sender] -= amount;
    payable(msg.sender).transfer(amount);
  }

}
```

State

Functions

Put More Succinctly…



What The *Hell* Was That?

The Ethereum blockchain exists in aggregate across all Ethereum nodes worldwide

It can be described as a *distributed state machine* rather than a ledger – that state just happens to include your balance

To determine how the state changes from one block to the next, it follows the *rules* of the Ethereum Virtual Machine (EVM)

The EVM In A Nutshell

The EVM is a collection of simple instructions that manipulate a stack (we won't dig in to this here)

These opcodes have incomprehensible names such as CREATE2, PUSH16, EXTCODEHASH, MLOAD and POP

Opcodes exist in the context of general computing (ARM, x86) – the EVM is a blockchain-specific set

That cursed jumble of hexadecimal you saw was just a series of these instructions that fetch, change and store *state* from one block to the next when called

Interfacing With The EVM

There are a few programming languages that allow you to write code that gets compiled into the EVM:

- Solidity, Vyper, Edge, Yul, Huff et al

You just saw an example of Solidity code

And might now be wondering "what does this have has to do with public policy?"

Aside: On Computability

Bitcoin makes use of a different low-level system called Script which handles BTC transactions (checks they're valid, etc)

But Script isn't 'Turing complete', so it can't represent arbitrary computation

A system that *is* Turing complete lets you run *any program you want*, rather than just punt Bitcoin from one address to another!

What *Are* Smart Contracts? Redux

Ethereum was designed as a global computer which is capable of performing precisely that computation which Bitcoin cannot

The EVM *is* Turing complete (*w/ constraints*), and it permits us to deploy any program we care to write to a new Ethereum address

As alluded to earlier – we (foolishly) termed these deployed programs smart contracts


AAH, SO IT'S LIKE THAT, HUH. I UNDERSTAND EVERYTHING NOW.
DOESN'T GET IT AT ALL

So What Do They Enable?

Any token (e.g. USDC) on Ethereum that isn't ETH itself is governed by smart contract

NFTs are issued by smart contracts that point at an URL for artwork (or, rarely, the art is directly generated on-chain)

Decentralised finance (DeFi) is *wholly* reliant on smart contract implementations

Focusing On DeFi

The vision of DeFi is that of offering financial infrastructure to the masses without relying on middlemen such as banks, brokerages or clearing-houses

Stablecoins are a good example lacking speculative volatility: tokens representing a fixed amount which can be minted, frozen or burned via their smart contract

DeFi: Finance's Kitchen Sink

We've gone a *lot* further though:

- Permissionless asset swaps (Uniswap),
- Overcollateralised lending markets (Euler),
- Oracles reporting real-life data (Chainlink),
- Looping leverage positions (Summer.fi),
- Flash loans to exploit arbitrage (Aave), etc etc

Your philosophy on the utility of blockchain may colour your vision of DeFi's usefulness, but we went ahead and did it anyway

Circling Back To Public Policy

Here are the facts on the ground as we go on:

- Ethereum is an open network, available to all
- *Anyone* can deploy *any program they want* if they pay the deployment cost in ETH
- *Anyone* can interact with *any program they want* if it isn't gated to certain addresses
- No registries, no constraints, no restrictions, and interactions often involve valuable assets


Impressive, very nice
Now let's see the consequences
for the human race

“My Name Is Ozymandias…”

Smart contracts are exploited *constantly*

US$3.48 billion stolen in DeFi (that we *know* of) between July 2021 – July 2022

I worked on one of the attacked protocols in that figure – US$16 million stolen

The largest heists tend to be of ‘bridges’ – used to move tokens between blockchains

How Does This Happen?

It's a common trope that every hack is an inside job done by developers: this is pretty far from reality, but *does* happen

It's *very* difficult to write bullet-proof code (any programmer will tell you this), especially for low-level systems: i.e. EVM

Constant vigilance is needed – concept of 'safe now and safe forever' does **not** apply

It’s All So Tiresome

Rules Written In Blood

I'm going to briefly cover two examples, one ancient and one recent

- The DAO Attack [June 2016]
- The Curve Vyper Attacks [July 2023]

They both involve an attack involving '**re-entrancy**' in different ways: it's the most vanilla type, but it pops up often

The DAO [June 2016]

```
function withdraw() public {

  …

  uint256 bal = balances[msg.sender];

  (bool sent, ) =
    msg.sender.call{value: bal}("");

  …

  balances[msg.sender] = 0;
}
```

This looks fine, right?

Amount stolen: US$60 million

For Forks Sake

This was the original re-entrancy attack: exploits attributable to abuse from running a function *again*, midway, before it's finished

Famously, this attack caused Ethereum to fork, and gave birth to the 'code is law' ideology as we see it applied to crypto

We started using re-entrancy guards in response to this, and considered it solved

Curve/Vyper Attacks [July 2023]

```
bool lockStatus;

function doStuff() nonReentrant {

    require(lockStatus == false);

    lockStatus = true;

    … the actual code for doStuff

    lockStatus = false;

}
```

Idea: if you've started `doStuff`, it must *complete* before you can use it again

Amount stolen: US$70 million

Vyper Attack: Root Cause

The code here was *100% correct*

The problem was that in the Vyper language, the way in which re-entrancy guards were *translated to the EVM* was incorrect: this was fixed *by mistake* without any fanfare

Bug lay in the shadows for over two years

Who's *responsible* for an incident like this?

Due to extreme market conditions, your money is gone.

Thank you for your patience and understanding.

Counteracting Hostility

The temptation is to throw one's hands up and say "right, this is all irredeemable"

Smart contracts are fraught with risk, but we're evolving as quickly as we get hit

Security works best at multiple points (e.g. wallet, contract, network, social layers)

Many practices now exist to help eliminate or mitigate issues that may lead to attacks

A Price For Every Palate

Depending on budget, we engage with:

- Gamified audit competitions
- Bug bounty programs
- Exploit insurance policies
- Whitehat counterattacks
- Embedded detection systems

Prevention is best, but if that's not on the table then we have to settle for cure

Not-So-Shadowy Super Coders

Blockchains aren't the black holes the media says they are – the whole *point* is that their activity is completely public

Blockchains themselves are *bad for crime*

Many addresses ultimately end up tied to exchanges (often with KYC policies), and more than one smart contract attacker has been identified by just checking the chain

The Dark Side of Privacy

As a response to this complete openness, systems were created that use zero knowledge (ZK) tech to break the chain

Tornado Cash is the most famous example of these – sanctioned by OFAC last August

Commonly used by Lazarus (North Korea) to mix exploited funds with clean ones or fund attacks in the first place

In Defence of Privacy

There are *many* reasons why you would want to hide what you spend money on

Equally many counter-arguments that tend to revolve around public probity

Financial privacy is a human right – "if you've got nothing to hide, you've got nothing to fear" is a panopticon mantra

A Rose By Any Other Name

New ZK mixers exist on Ethereum that have *the same functionality* as Tornado

Specifically, the ability to **prove** that your funds did *not* originate from any of a set of addresses known to be tainted/sanctioned

So, why isn't there a universal clampdown? Gray money hypothesis? Lack of size?

"You Think Darkness Is Your Ally?"

Another narrative that heats up depending on the current perception of the regulatory environment is *base-level privacy*

In systems adopting this (Aztec Connect), one cannot infer *any* information about interactions that they are not party to

Push too hard and crypto devs *will* move their brain trust and capital into the dark

Moving Towards The Future

Ethereum is slowly embracing a paradigm we refer to as 'intents-based'.

Compare:

"Walk to the nearest Tesco Express, walk in, pick up milk, pay for it and bring it back: here is £2 for the milk and a £0.50 tip"

With:

"I'll give £2.50 to the first person that brings me milk"

Living In Interesting Times

As the intents-based paradigm spreads in usage, likely to be several new vectors for attack: intents are smart contract reliant

The trusting relationship that intents gives rise to has plenty of room for new and exciting ways for people to get cleaned out

We'll learn and adapt: we always do

Conclusion

Smart contracts are *fascinating* – arbitrary programs operating in trustless, adverse environments are a gigantic nerd-snipe

They make for an environment where no one is ever bored, but everyone also dies of stress at the respectable age of 36

I am 35

Questions?