# CryptoCuties: A New Family of Non-Fungible Virtual Assets

PRIMES Conference October 17

Rachel Chen
Mentor: Jules Drean

# Our Origins

## NFTs

Established way of virtual ownership using **blockchain**

- Ownership is verifiable by tracking down the blockchain
- Without need for third party verification! Wow!

**CryptoKitties**

First NFTs minted using Ethereum's ERC-721 token standard

$300k

# What's wrong with NFTs?

**Problems with current NFTs**

- A user's private key is proof-of-ownership over virtual asset by creating digital signature. And that's all you get.
- Relies on human agreement that your signature actually represents ownership before you get any "rights" over a virtual asset
- Ultimately no technical reinforcement tying signature to ownership

**Idea:** What can we do better? → Our motivation

**Find a possible solution for a stronger sense of virtual ownership?**

- Limits capabilities to people who own illegitimate versions of virtual assets (ex: copies)
- Technical reinforcement of virtual ownership: Inherent "rights"

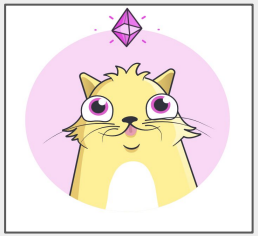
# Introducing CryptoCuties: Our Version of Virtual Assets

- These virtual assets are associated with a **proof of validity** and an **audible history** of any interaction or changes done to the CryptoCutie.
- Only **authorized hardware** **can create a proof of validity** for the interactions and modifications of a specific CryptoCutie.
- CryptoCuties need to be **regularly updated** to stay valid.

**Tamagotchi**

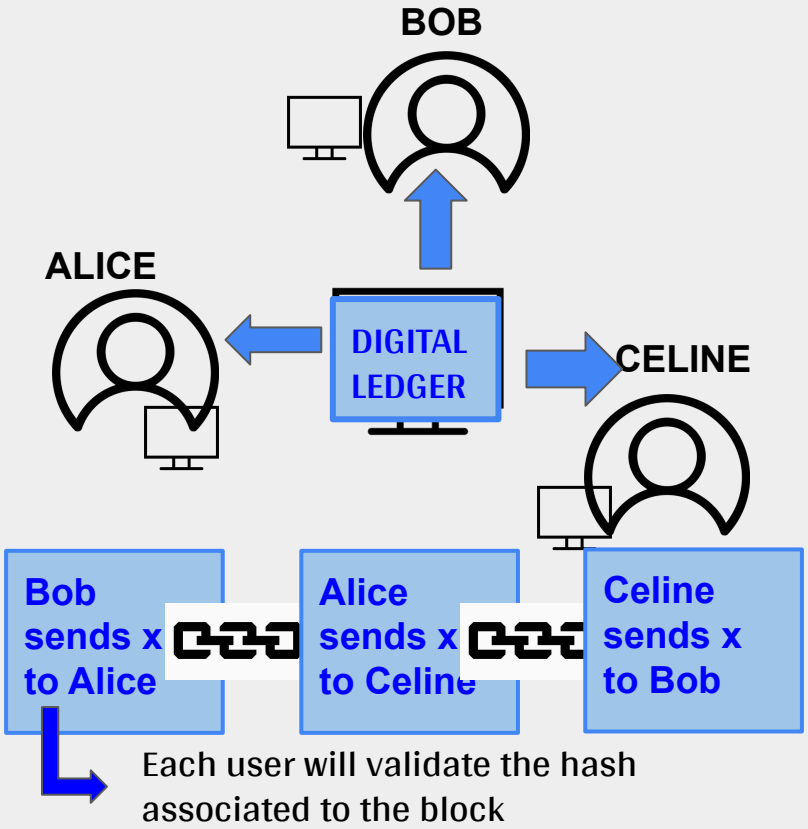
+

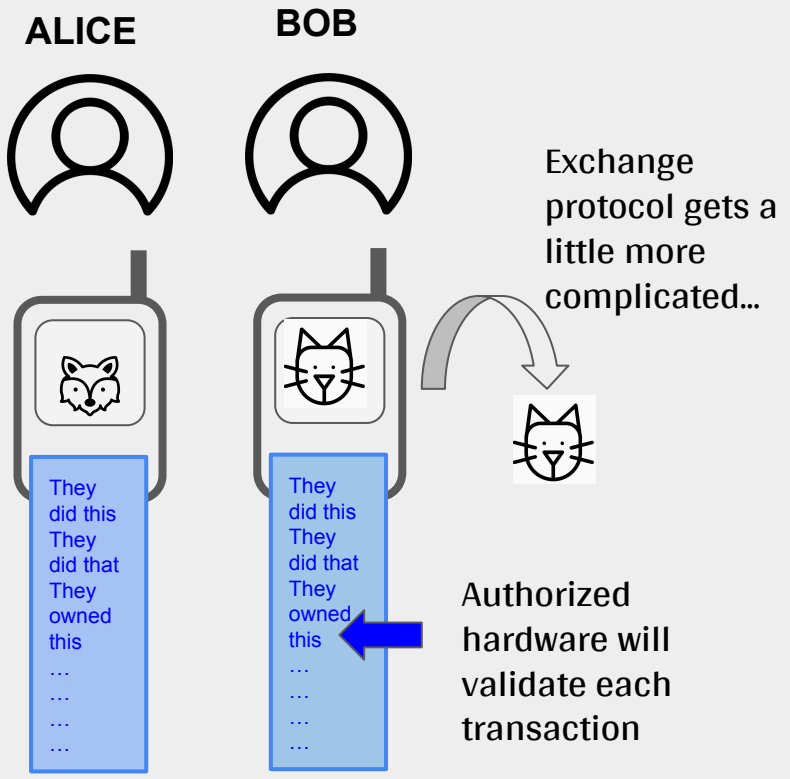
**CryptoKitties (NFTs)**



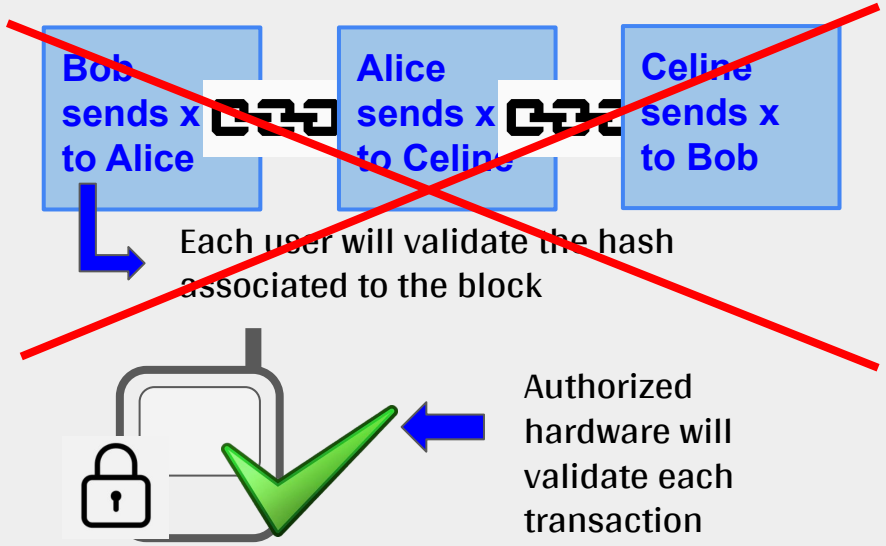
Hardware +
Virtual Assets

# Comparison: Blockchain Vs. CryptoCutie



# Motivation




## Using Trusted Hardware To Replace the Use of Blockchain

- Replaces the need for global consensus by placing trust into secure hardware

## Local History vs. Global Shared Ledger

- Replacing global history with local history embedded with the virtual asset
- No dependency on third party or block chain

# Security Guarantees

**1 Non-forgeability:** Cannot create a valid CryptoCutie with fake history


**2 Non-forkability:** Cannot copy an existing CryptoCutie and make it legitimate.

**3 “Liveliness”:** Cannot be killed intentionally by a malicious remote adversary.

**4 Singularity:** There can only be one valid copy of a CryptoCutie owned by one device at the same time.

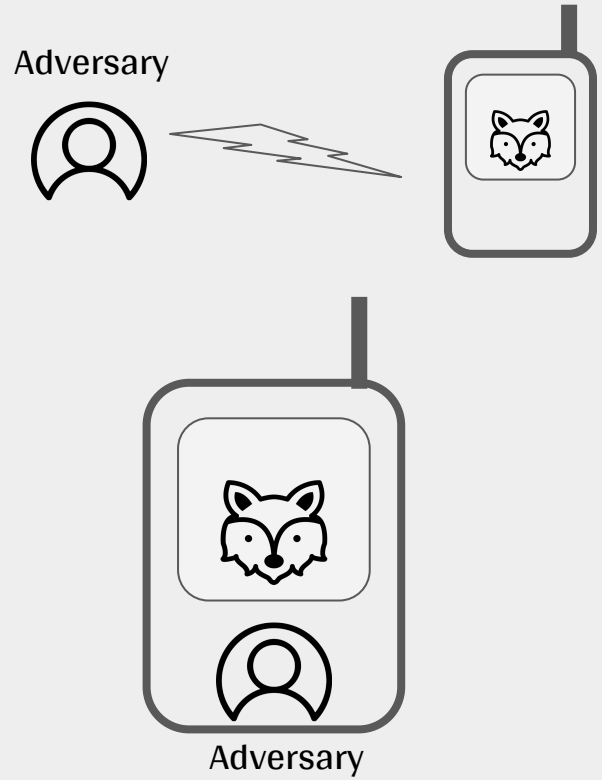
# Threat Model

**Remote (thus weaker) Attacker:** someone on the network that doesn't have physical access to the device

- GOAL: All security guarantees hold

**Local (strong) Attacker:** someone that has physical access to the device, can run arbitrary code on it etc

- GOAL: Non Fungibility holds, but not liveliness (that's ok)



# **Implementation:** What do we want?

**Assume: Minimal Requirements**

- Pair of keys with certification signed by trusted manufacturer
- Small tamper-proof storage to store keys
- Remote attestation protocol

**Cryptographic Link**

1) Cryptographic binding between the Crypto Cutie and a single piece of trusted hardware

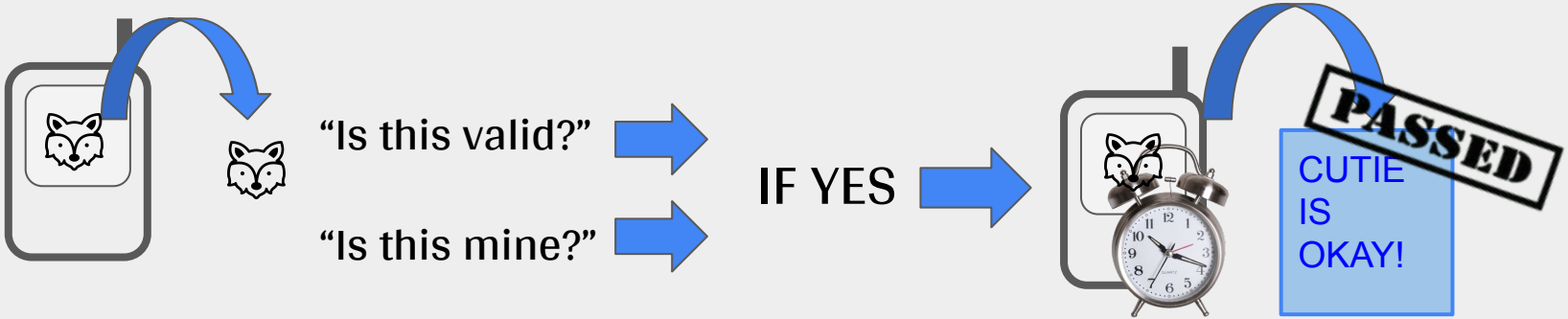
2) This binding represents ownership of the CryptoCutie

3) Other trusted hardware cannot generate a valid proof of ownership and liveliness for a given CryptoCutie

- Prevents copies existing on other hardware.

# **Deeper into implementation:** Sustenance Requirement

**“Heartbeat” Mechanism → <u>Prevents offline duplications</u>**




If the **interval** between two proofs is too long, then the CryptoCutie is made INVALID. WHICH MEANS...

Without constant proof of validity by trusted hardware, the CryptoCutie is considered invalid. Can perish if left alone or stored offline.

# Time Synchronization

We need time synchronization for heartbeat mechanism to work.

**We can let owner choose their own clocks:** It's against her interests to change the time because then her CryptoCutie will become invalid.

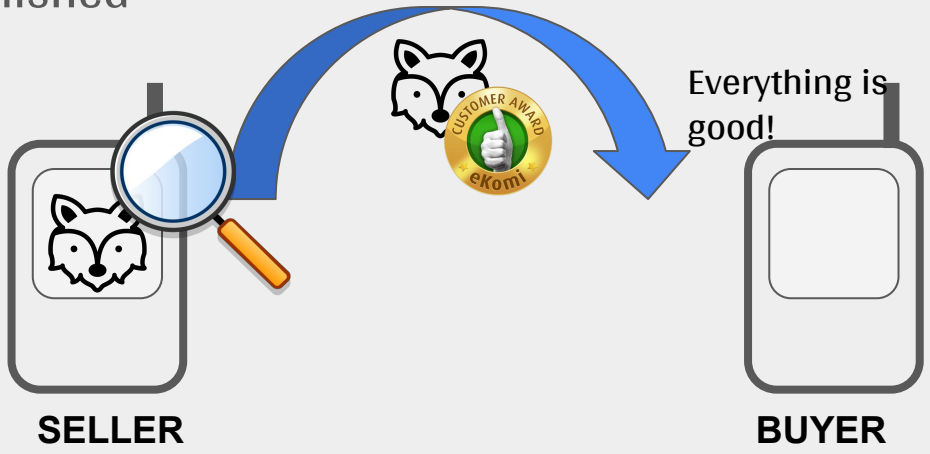
# **Deeper into implementation:** Exchange Protocol

Problem: Heartbeat requirement → How to exchange without CryptoCuties becoming invalid?

Solution: “Frozen” state - Extends allowed interval value between two heartbeats until exchange is finished

WE NEED REMOTE ATTESTATION which will verify

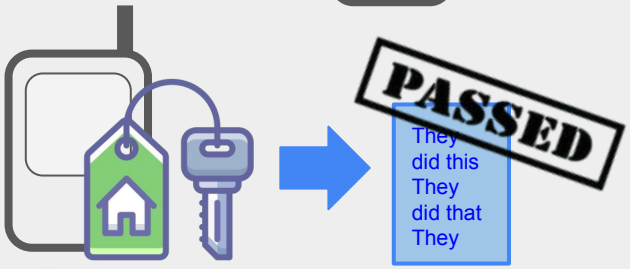
- Selling to the buyer
- Only to that one buyer
- CryptoCutie is valid



# How We Make Any Duplications Invalid

#1: **Duplication stored on same hardware**: Trusted hardware can check.

#2: **Duplication stored on different hardware**:
Cryptographic binding prevents unauthorized hardware from generating valid proof of liveliness and ownership.




#3: **Duplication stored offline on hardware**: Heartbeat and Time Sync Requirement

#4: **Duplication of frozen state from exchange protocol**: Code attestation verifies the freezing protocol and ensures exchange of valid CryptoCutie.

# Future Work

- **What are our minimal requirements?** Looking at TPM 2.0 Capabilities
  - Virtual Monotonic Counter: Counter than cannot be turned back, only incremented and read
- **Attestation: Turning interactive proof to non-interactive proof** (in order to keep our protocol offline)

# Acknowledgements


- Mentor Jules Drean
- MIT PRIMES
- My family