Non-Fungible Objects (NFO): Hard-to-Counterfeit Virtual Assets Based On Trusted-Hardware

Rachel Chen

Mentor: Jules Drean

Plan

I. Discussion of our origins and motivation

II. Introduction to Non-Fungible Objects and our secure hardware

III. Presentation of our attestation mechanisms in detail

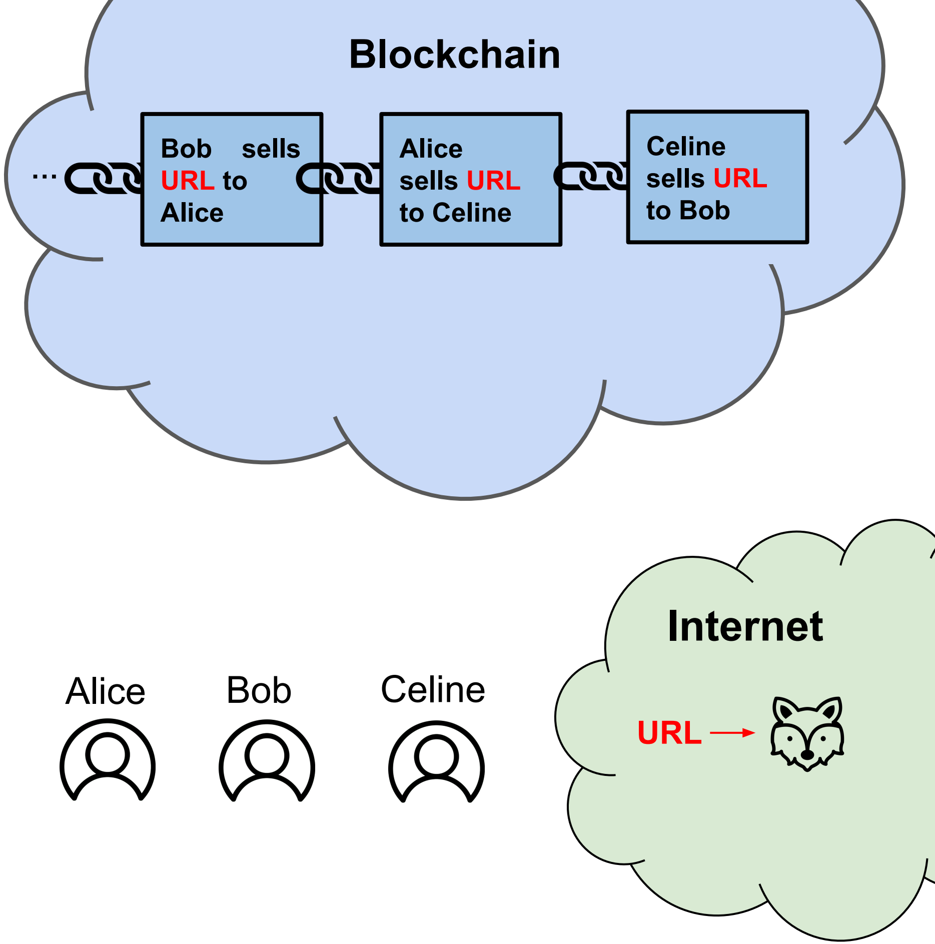
IV. Proposing an application

V. Displaying our prototype

VI. Final Remarks

Our Origins: “Non-Fungibility”
Virtual Ownership and the Blockchain

- Explosion of market of virtual goods
- Virtual ownership
- Non-Fungibility
- Current-day solution:
  Non-Fungible Tokens (NFTs)



Blockchain

…

Bob sells URL to Alice

Alice sells URL to Celine

Celine sells URL to Bob

Alice

Bob

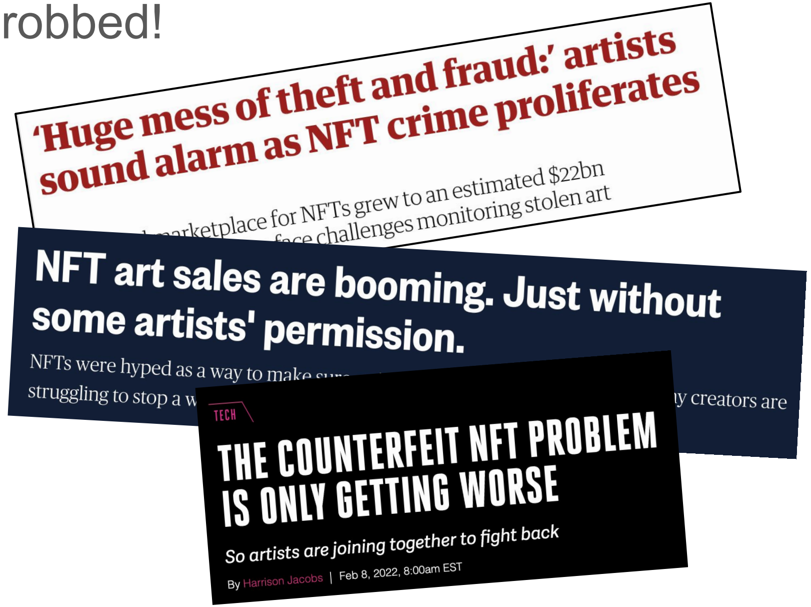
Celine

Internet

URL →

Non-Fungible Tokens Pitfalls

Lacks proof of authenticity: people get robbed!



'Huge mess of theft and fraud:' artists sound alarm as NFT crime proliferates

marketplace for NFTs grew to an estimated $22bn
face challenges monitoring stolen art

NFT art sales are booming. Just without some artists' permission.

NFTs were hyped as a way to make sur
struggling to stop a w
y creators are

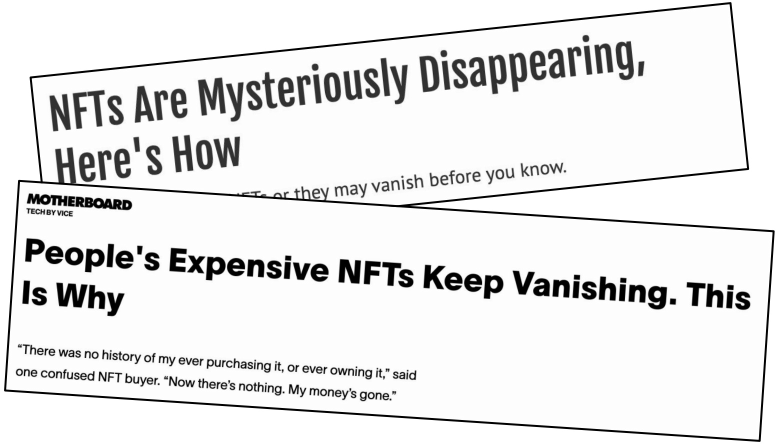
TECH

THE COUNTERFEIT NFT PROBLEM IS ONLY GETTING WORSE

So artists are joining together to fight back

By Harrison Jacobs | Feb 8, 2022, 8:00am EST

When exchanging NFTs, you only exchange a URL pointing to an image!



NFTs Are Mysteriously Disappearing, Here's How

FTs or they may vanish before you know.

MOTHERBOARD
TECH BY VICE

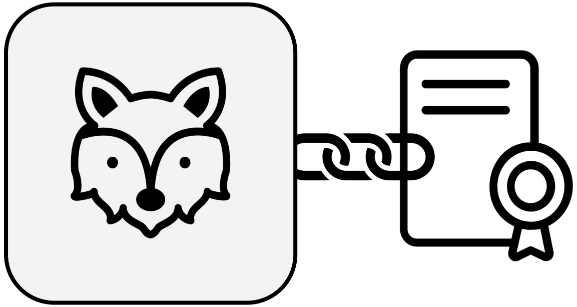
People's Expensive NFTs Keep Vanishing. This Is Why

"There was no history of my ever purchasing it, or ever owning it," said one confused NFT buyer. "Now there's nothing. My money's gone."

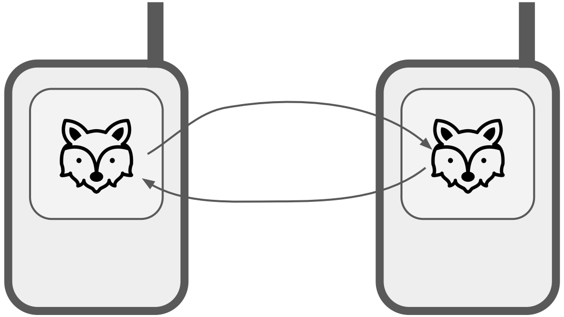
How do we tackle these problems?

We introduce **Non-Fungible Objects** (NFOs)

We rely on trusted hardware to attest any action on the NFO and create a certificate of authenticity.

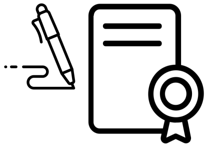


We don't use a blockchain. Most operations are done offline and the data moves with the ownership.

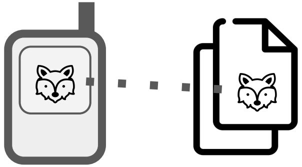


Security Guarantees

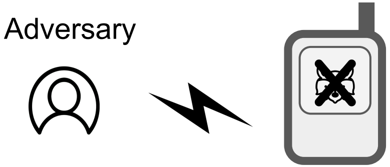
1 Non-forgeability: An attacker cannot create a valid NFO with a fake history



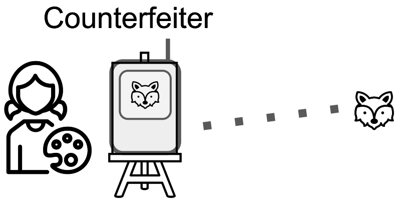
2 Non-forkability: An attacker cannot make a valid duplicate of an existing NFO



3 Liveliness: An attacker cannot kill a NFO remotely



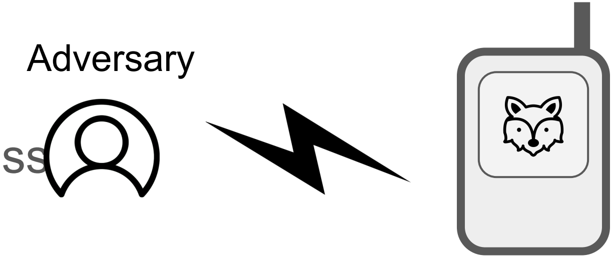
4 Authenticity: Reproducing a NFO is difficult: Requires a skilled counterfeiter



Threat Model

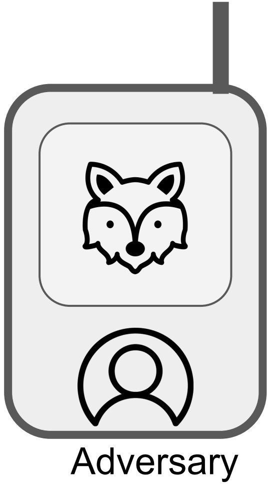
Remote (hence weaker) Attacker: an active network attacker without physical access to the device

- GOAL: All security guarantees hold

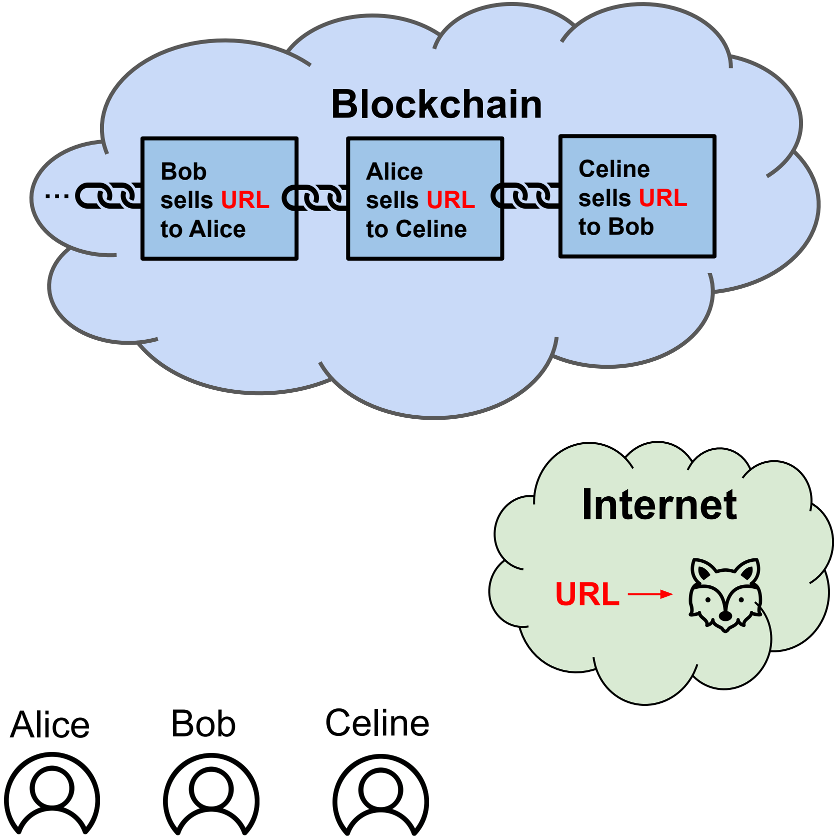


Local (strong) Attacker: someone that has physical access to the device, can run arbitrary code on it etc

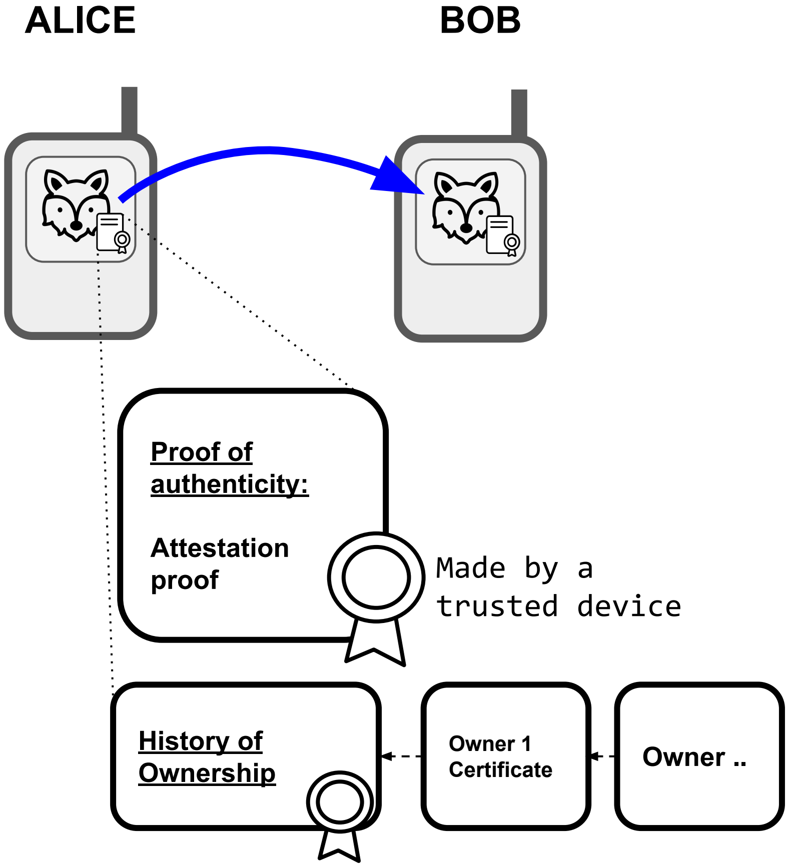
- GOAL: All security guarantees hold…
  expect liveliness, authenticity (and that's okay)



NFTs vs. NFOs



Blockchain
…
Bob sells URL to Alice
Alice sells URL to Celine
Celine sells URL to Bob
Internet
URL
Alice
Bob
Celine



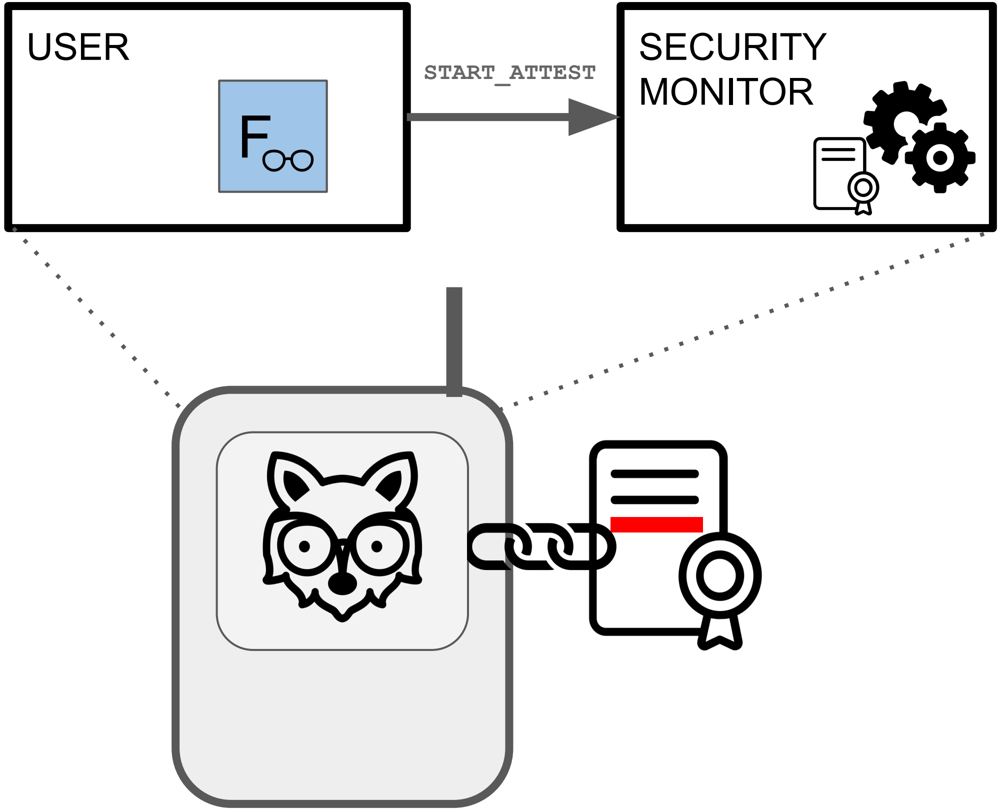
ALICE
BOB
Proof of authenticity:
Attestation proof
Made by a trusted device
History of Ownership
Owner 1 Certificate
Owner ..

Attestation Mechanism

Attestation data is created through this mechanism

Remote attestation: The host device attests the authenticity of the functions used to interact, edit or exchange the NFO.

`SYS_CALL` System calls are how the user interacts with the Security Monitor and the attestation mechanism

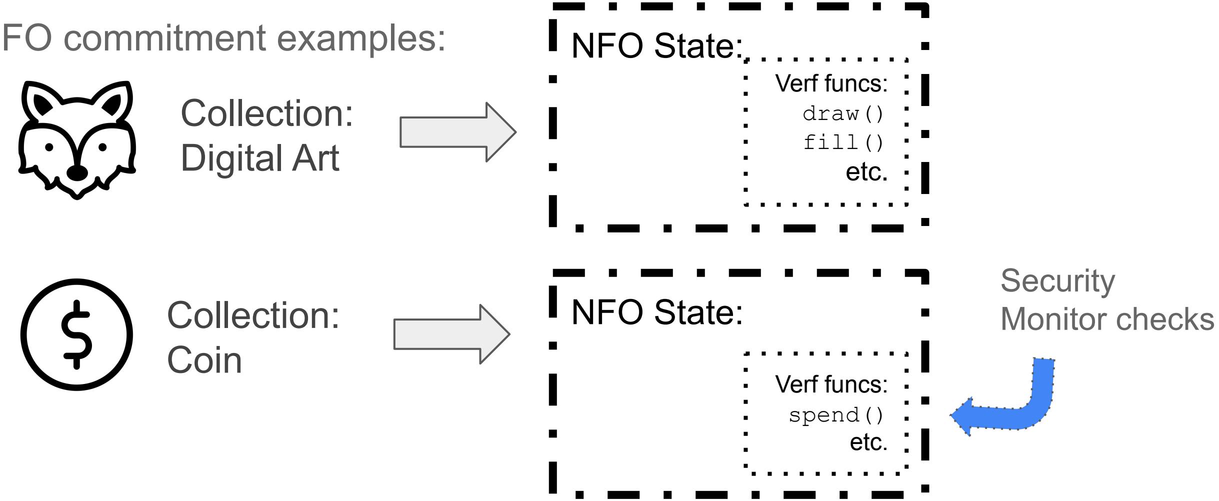


NFO collections

NFO collections are defined by “verifiable collections”

During the initialization of an NFO, the NFO will **commit** to a family of “verifiable functions” which define the “collection” the NFO belongs to.

NFO commitment examples:



Collection: Digital Art

NFO State:

Verf funcs:
`draw()`
`fill()`
etc.

Collection: Coin

NFO State:

Verf funcs:
`spend()`
etc.

Security Monitor checks

Exciting part:
Easy for devs to create and define their own NFO collections by creating their own family of valid funcs!

Invites creativity!

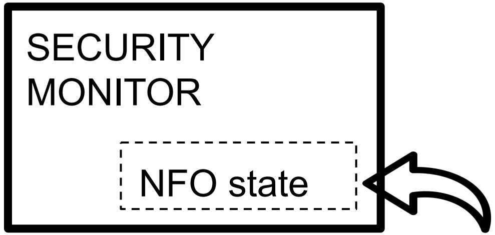
Preventing Duplications of NFOs:

On a single device

Goal: Preventing the creation of valid duplicates in order to uphold non-fungibility.

Problem: Offline duplications put back onto the device to be attested

Solution: Security monitor holds hash of NFO state.



Before you can alter NFO, security monitor checks if hash of the NFO matches the hash that is stored

<u>If hashes don’t match:</u>

An adversary could be using a duplicated NFO from a previous state

Result: NFO deemed invalid.

<u>If hashes match:</u>

User is using correct, most up-to-date NFO

Result: Function over NFO is attested

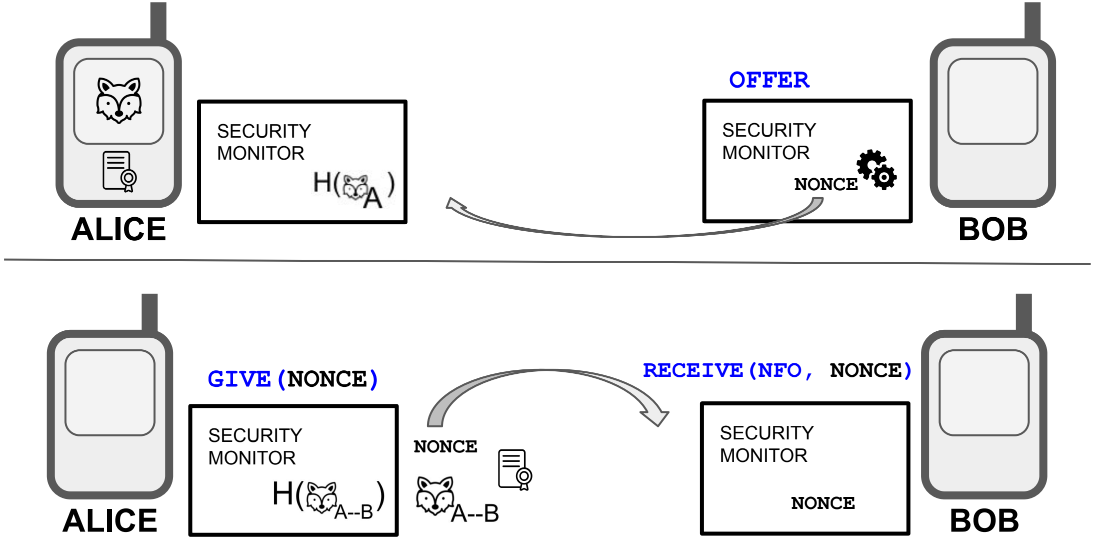
Preventing Duplications of NFOs:

Multiple devices (exchange)

Goal: Preventing the creation of valid duplicates in order to uphold non-fungibility.

Problem: Passing around duplications to different devices to be attested.

Solution: Security monitor + Special exchange protocol

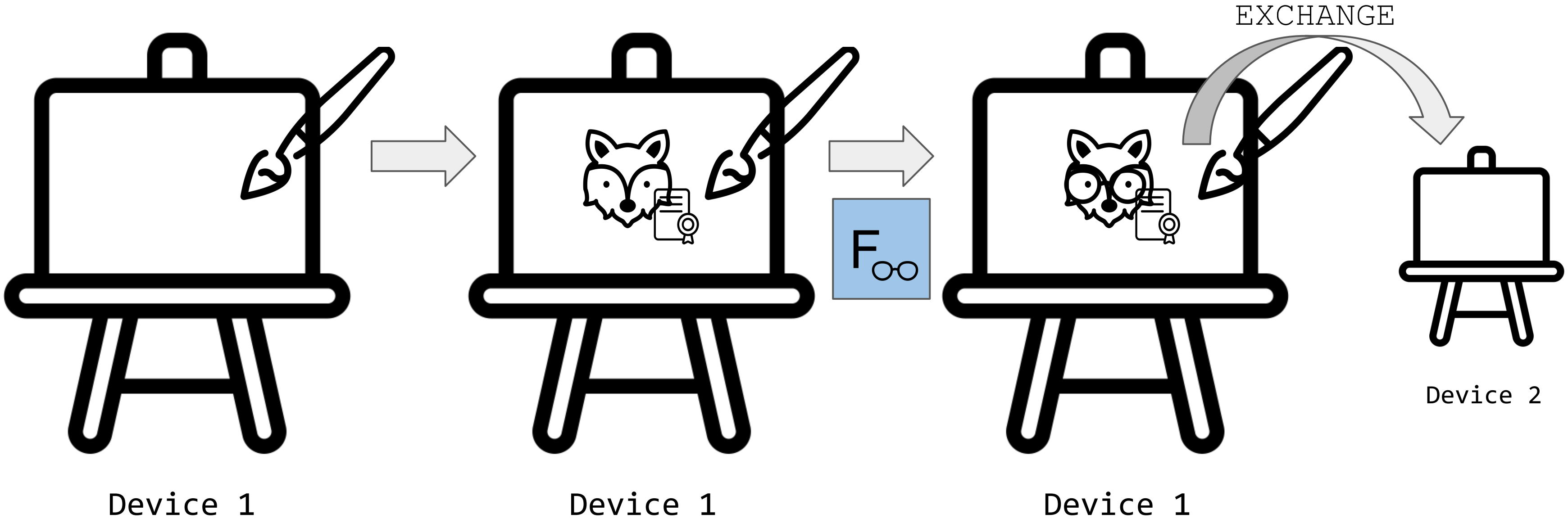


Alice can’t exchange or use any past duplicates

Application example: Digital art

Creating art on the device

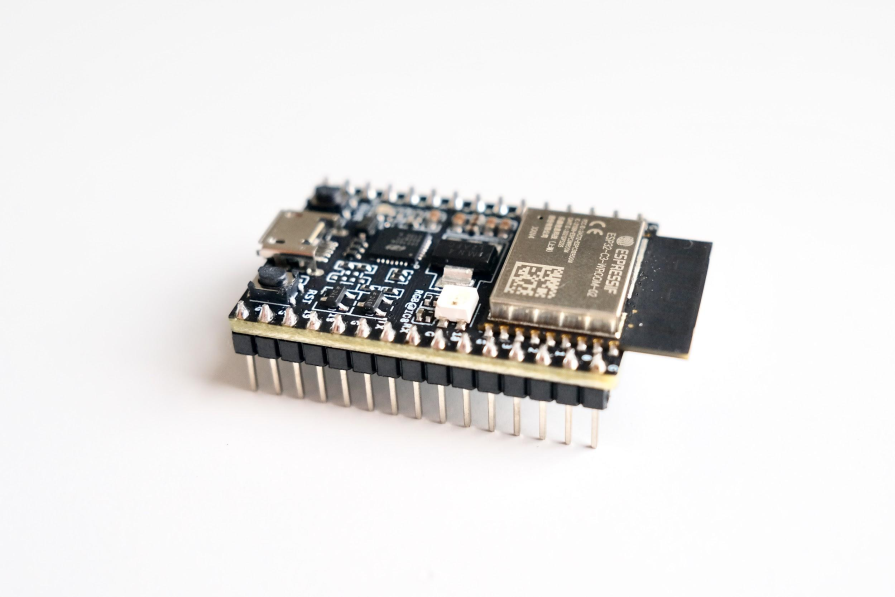
Each stroke on the canvas is equivalent as a function over the NFO which makes NFO art difficult to replicate.

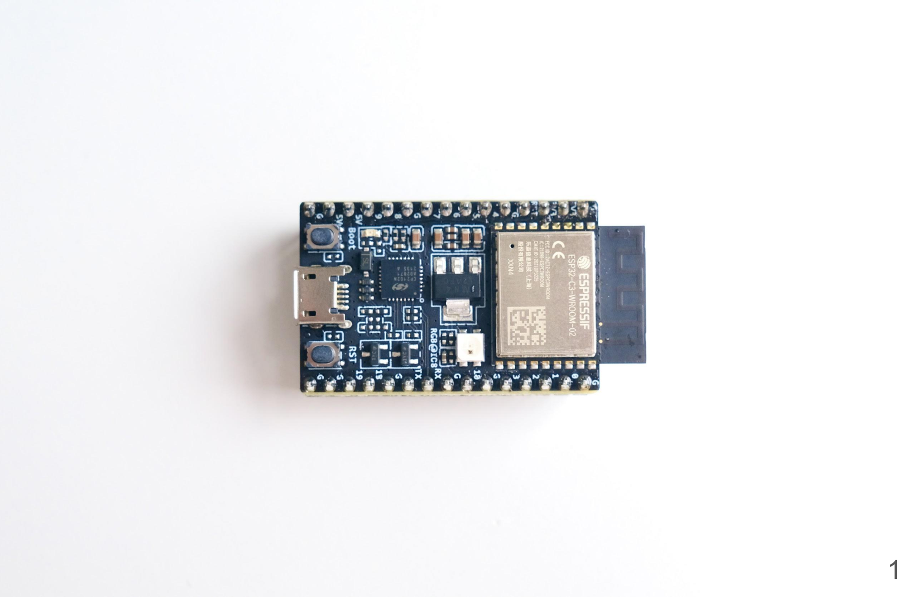


NFO hardware prototyping in the works

ESP32-c3

RISC-V





Other details in the paper that we haven’t covered

- **Security monitor**: Separation of untrusted and trusted modes on the hardware to execute attestation mechanism
- **Never powering off paradigm**: Keys are stored in volatile memory to be harder to steal off of device
- **Transition matrix**: Matrix of all transitions between verifiable functions of a NFO collection to inform the SM what is a valid transition or not
- What makes a secure verifiable function
- How to provision keys and attest that a device is valid
- Security and Trusted Hardware Assumptions
- Other applications (digital coin, video games)

Any Questions?

Acknowledgements

Jules Drean, my mentor, for his guidance, expertise, and contributions

Srini Devadas, for supporting and providing insightful feedback on my research

Slava Gerovitch, for organizing MIT PRIMES for high schoolers and support throughout my research progress

Thank you all!