




Confidential Compute In Decentralized Networks

\$whoami

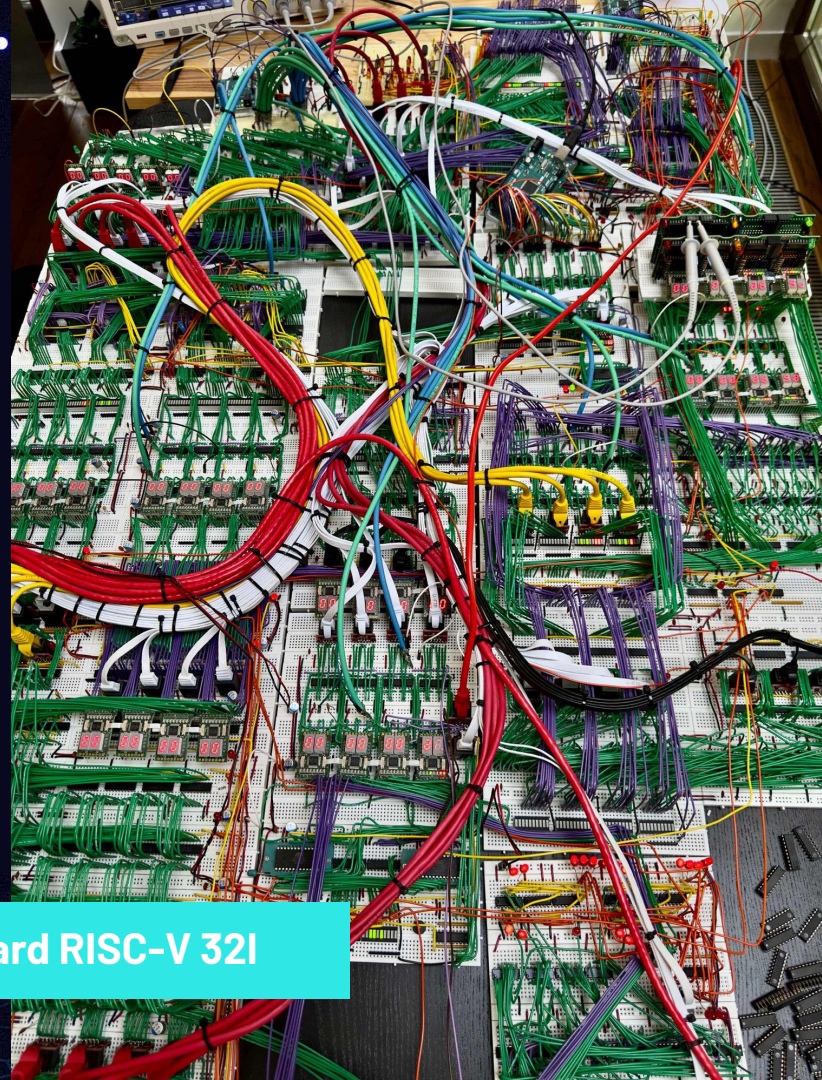
Magik6k

Łukasz Magiera

- Co-Founder **Curio Storage Inc.** 
- Previously contributing to **IPFS / Filecoin**
- Building weird **CPU**s because I can



74xx BreadBoard RISC-V 32I





What Am I Going To Talk About!

1. Decentralized Network Introduction
2. Compute on (un)trusted hardware
3. Confidential Storage for DSN
4. Attestations for Decentralized Networks
5. Attestation Runtime
6. Q&A



Decentralized Network Introduction

01

What's the big idea in Decentralized Networks



1. "I want to retrieve X from Y" -> "I want to retrieve X"
2. "I want to store X with Y" -> "I want to store X"
3. "I want to compute X with Y" -> "I want to compute X"



Decentralized Network Aims

A GOOD Decentralized Network SHOULD:

- Allowed Client to buy services from “The Network” without individual Client-Provider trust-based relationships.
- Allow anyone with sufficient hardware to become a Provider and sell their service to “The Network”

Decentralized Networks without TEE



Storage

- Solved multiple times in multiple ways, generally with constructions periodically checking some merkle-proofs, possibly compressed with ZK-SNARKS
- Still in some (more enterprise) cases could use better guarantees



Retrieval

- Attempts were made, but generally there are fundamental limitations on what can be proven about retrieval between byzantine entities.



Compute

- There are existing networks, but not yet close to what's needed for "web2 scale"

What Confidential Compute Can Do In This Area

- Decentralized systems typically operate under the assumption that the only trustworthy elements are the protocol rules enforced by the software each user runs.
- The code run by other protocol participants is susceptible to tampering.



TEEs make it possible to trust the code being run by other parties.



Broad Themes

- Proving / Attestation of Entity Parameters
- Improving Data Confidentiality
- Running Compute Jobs on untrusted providers





Compute on (un)Trusted Providers

02

Running Compute Jobs on Untrusted Providers

Aka “Just let anyone run isolated VMs/Containers on your HW”



Running Compute Jobs on Untrusted Providers

1. Get an image to run
2. Start a Confidential VM
 - a. Give it **RAM**, **Cores**, **Network**, Storage, etc.
3. Get paid for used resources.

Running Compute Jobs on Untrusted Providers

- Network, Storage
 - Can mostly measure parameters with a second resource-parameter-attesting VM
- RAM
 - The VM can know what memory pages it has allocated.
 - Can it know things like memory speed?
- Cores
 - Need some mechanism to know cpu time allocation
 - Also cpu frequency, threads, caches, power limit (incl thermal throttling), etc.
- Etc.
 - GPUs
 - Time
 - Some jobs require accurate trustable time source



Improving Data Confidentiality

03

Decentralized Storage Networks Today

Today data is stored with untrusted providers directly as sent by clients

- Encryption can be applied but is often not enough
- ACLs can be added to the system, but by definition are best-effort - enforced by untrusted providers
- Data deletion is not possible to prove

DSNs today only strongly guarantee data storage, other services are best-effort.

Decentralized *Confidential* Storage Networks



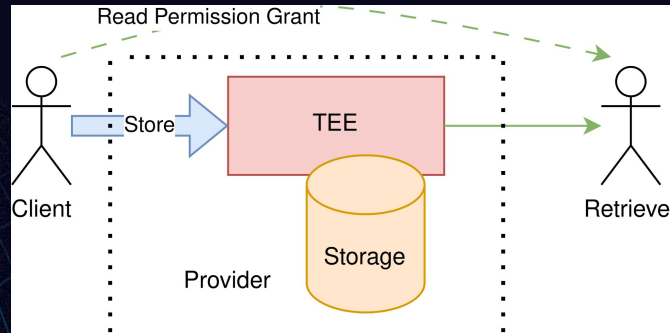
With TEEs it is possible to do things which otherwise (and often have been formally proven to be) impossible!



Decentralized Confidential Storage Networks

With TEEs it should be possible to:

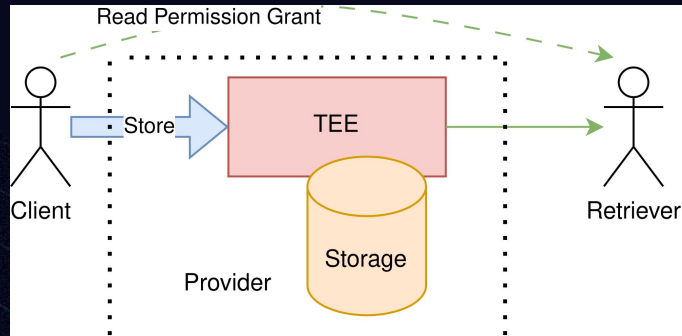
- Retain cryptographic storage durability/replication guarantees of today's DSNs
- Make the actual data invisible to Providers
 - Layered at-rest encryption (Client + TEE)
 - Storage-Level ACL - Provider can't just read client data



Decentralized Confidential Storage Networks

With TEEs it should be possible to:

- Support stronger ACLs
 - TEEs would only send data to authenticated clients.
- ZK-SNARK Proofs could be computed in TEEs with GPUs
 - Or only minimal information needed to compute the proof would be released by the TEE



Decentralized Confidential Storage Networks

Data Deletion should be possible to attest

- If storage media is talking directly with the TEE, the TEE can attest that the data:
 - Was only stored on that storage media, wasn't copied or tampered with
 - Can attest that requests to delete data have actually deleted data from the physical drives



Attestations for Decentralized Networks

**0
4**

Attestations for Decentralized Networks

Fairly Simple: Prove that X (public, private) is true

Attest(Public, Private) -> Attestation

Verify(Public, Attestation) -> True/False



Attestations for Decentralized Networks

Main Challenge: Unclear, Catastrophically-Breakable Security Bound

Attestations for Decentralized Networks

Main Challenge: Unclear, Catastrophically-Breakable Security Bound



How the conversations with engineers in this space typically goes:

> Hey I think we could do X with CC.

< Oh absolutely not, I've seen secure enclaves be broken over and over, [...]

> [30 mins of explaining, giving examples, reassuring about limited assumptions]

< Ok maybe there is some value.

Attestations for Decentralized Networks

Main Challenge: Unclear, Catastrophically-Breakable Security Bound

- Hardware can be, and routinely is broken
- Developers have to assume that there is a finite \$\$\$ amount where
 - An individual attestation can be forged
 - An individual secure VM can be compromised
 - An individual machine enclave can be compromised
 - A whole line of hardware can be compromised
 - That \$\$\$ amount generally gets lower with time
- There exist actors who must be ultimately trusted



Attestations for Decentralized Networks

Main Challenge: Unclear, Catastrophically-Breakable Security Bound

THIS IS OK!

Attestations for Decentralized Networks

Main Challenge: Unclear, Catastrophically-Breakable Security Bound

Do the obvious:

- Don't concentrate too much trust in each VM
 - E.g. don't use a single VM to run an entire ledger
 - Don't put network-global secrets in every VM
- Obfuscate high-value targets
 - Make it very hard for Providers to know which VMs / Machines are running the most "interesting" workloads
- Generally decrease any attack rewards and increase attack costs

Attestations for Decentralized Networks

Main Challenge: Unclear, Catastrophically-Breakable Security Bound

- The reputation can be improved:
 - Open Source / Source Visible cryptography related hardware
 - Show that it's impossible to leak the actual root key
 - Show that firmware can't ever access the real root key
 - Same with CC-related parts of firmware
 - Make it easy to audit, have robust public bug bounty program
 - At least give third-party researchers/auditors access to sources
 - Share audit reports widely



Attestations for Decentralized Networks

1. Possible to create attestation-proofs for very large computations
2. Possible to attest to ongoing parameters of a system

Attestations for Decentralized Networks

1. Possible to create attestation-proofs for very large computations
2. Possible to attest to ongoing parameters of a system

Example: Proof-of-data-integrity

A computation ingests some raw data along with a random challenge. The output is an attestation with the hash of the data next to the challenge.

This proves that whoever ran the computation had to have every single bit of the data to produce the resulting hash. Should easily scale to Exabytes of checking throughput per day across a network. Primarily useful for enterprise storage needs



Attestations for Decentralized Networks

1. Possible to create attestation-proofs for very large computations
2. Possible to attest to ongoing parameters of a system

Example: Attested queries.

Networks like Filecoin tend to work with content-addressed data, which is already self-certifying. With CC it's possible to implement data manipulation/query primitives which don't require the Client to compute anything without losing verifiability.

Attestations for Decentralized Networks

1. Possible to create attestation-proofs for very large computations
2. Possible to attest to ongoing parameters of a system

Example: Attested Retrieval

- Stronger: VM which attests data exchange between two secure enclaves took place (at some speed, with some latencies)
- Weaker: VM which attests that data was sent to some AS (with some parameters)



Attestations for Decentralized Networks

1. Possible to create attestation-proofs for very large computations
2. Possible to attest to ongoing parameters of a system

Example: Attested Disk Access Parameters

- Simple VM measures requests to data which is supposed to be available in the local data center. It can then create attestations about throughput and latencies



Attestations for Decentralized Networks

1. Possible to create attestation-proofs for very large computations
2. Possible to attest to ongoing parameters of a system
3. **Ledger Use Cases**

Example: Off-Chain smart contracts

- Trusted VMs can run code which syncs a chain, and is able to interact with it in automated ways

Attestations for Decentralized Networks

1. Possible to create attestation-proofs for very large computations
2. Possible to attest to ongoing parameters of a system
3. **Ledger Use Cases**

Example: Off-Chain smart contracts

- Trusted VMs can run code which syncs a chain, and is able to interact with it in automated ways (e.g. send some amount of money after receiving weights for an ML model which is able to solve some challenge)

Note: Highly limited by uncertain security bound of secure enclaves

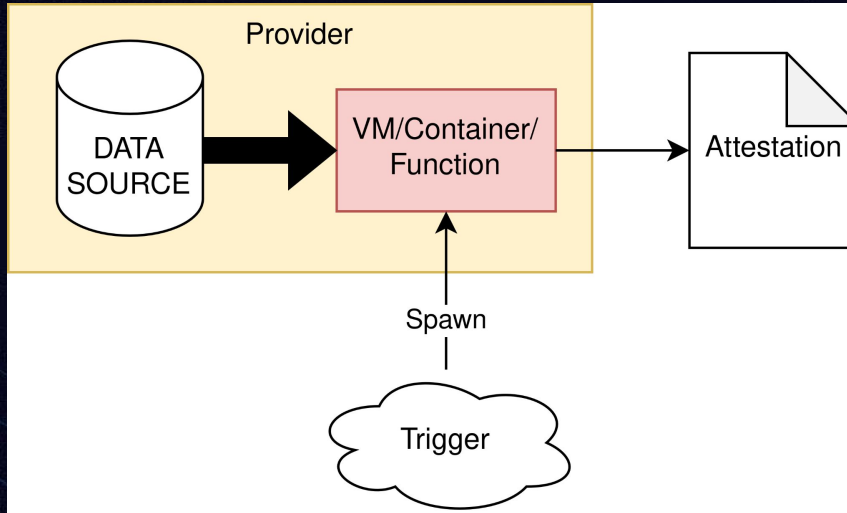


Attestation Runtime

05

What is Needed in The Space

- A way to run (open-source, audited) simple computations
 - I.e. not whole applications
 - Persistent storage not really necessary
- A convenient way to verify various attestations in on-chain applications



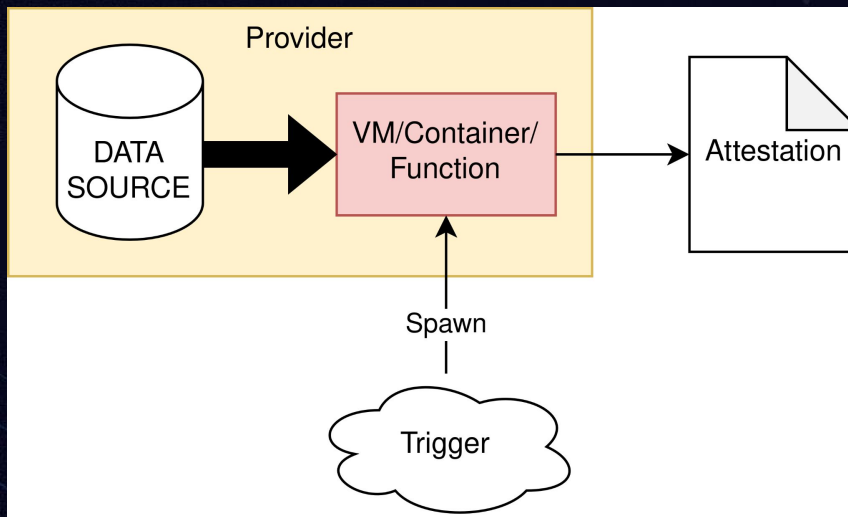
What is Needed in The Space

- A way to run (open-source, audited) simple computations
 - I.e. not whole applications
 - Persistent storage not really necessary
- A convenient way to verify various attestations in on-chain applications
- High throughput IO, able to use high performance cpu features
 - Churn through data at 100 Gbps + per machine.



What is Needed in The Space

- Function Runtime?
- Input/Passthrough Data, Output Attestation(s)



What is Needed in The Space

- Enarx - Capable enough in the attestation department; Wasmtime is also really fast, but likely not fast enough
- CoCo / Kata - Closest to what is needed, but still focused around longer lived workloads?
 - CoCo + Function runtime? Not aware of any which works well together
- Ecosystem seems quite big, but with distinct goals. Discoverability is hard.

What is Needed in The Space

- Need a way to verify various attestations in blockchain applications
- Quite simple but has to be done by someone
- Set of smart contracts implementing PKI, small framework for working with SNP/TDX/etc. attestations

What is Needed in The Space

- Need a way to verify various attestations in blockchain applications
- Quite simple but has to be done by someone
- Set of smart contracts implementing PKI, small framework for working with SNP/TDX/etc. attestations
- Marlin/Oyster seem to have done most of this already
 - Others I'm not aware of?

What is Needed in The Space



- There is some learning curve, things seem rather mysterious at first
 - Majority of developers don't even realize that modern server CPUs have this magical capability.
- Most components seem to exist individually
 - There is just a bit of glue that's missing
- Hackers!
 - Everything in this space is Open Source!



Q&A

06

magik@curiostorage.org