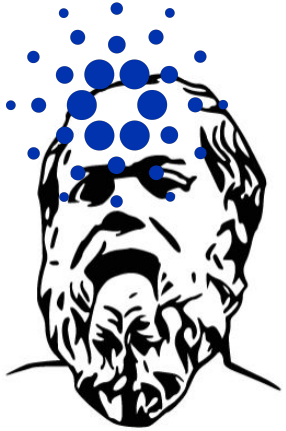
zokrada



Bringing zk-SNARKs to Cardano

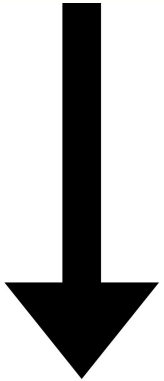
ZK Tools for ETH are here & deployed

- ZK Proofs on Cardano not researched a lot
- No support in any Smart Contract language

```
1  def main(private field a, field b) {
2      assert(a * a == b);
3      return;
4  }
```

Meanwhile

- ETH tooling abundant
- Can write abstract tools and compile to Solidity Contract

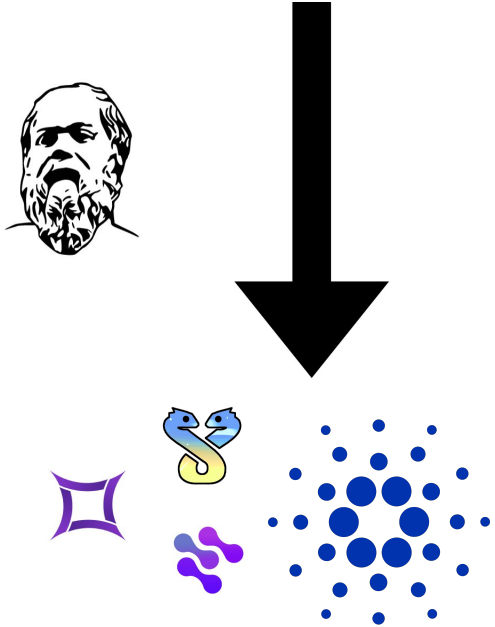


ethereum

zkADA

- Leverage existing tooling to export verifiers on Cardano
- Compatible with any Smart Contract Language Through UPLC ABI

```
1  def main(private field a, field b) {
2      assert(a * a == b);
3      return;
4  }
```



Feasible!

- Prototype for export already verified

  → Works for ZK proofs in OpShin

- However too expensive when entirely validating without EC Primitives

Steps to completion

- Needs cheap primitives for ECC
- Can benefit from a UPLC-level library standard

- CIPs
  - CIP 381: BL17-381
  - CIP XXX: BN254
  - CIP XXX: UPLC ABI

Opportunities for Cardano

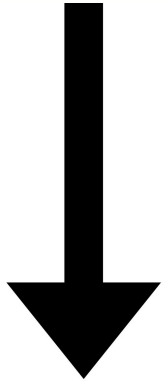
- ZK Enthusiasts can start building immediately
- Can transfer knowledge from existing implementations and build on top

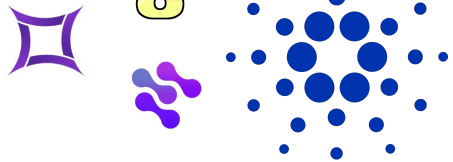
- Will attract more experts to research rollups etc.

Summary: zokrada

- Brings ZK Proofs to Cardano
- Leverages existing technology and attracts Brainpower
- Is implementable in the short timeframe

```
1  def main(private field a, field b) {
2      assert(a * a == b);
3      return;
4  }
```





Check out the prototype:
https://github.com/nielstron/zkada