The ZKProof Process Towards Standardising Zero-Knowledge

ZKPROOF

Mary Maller, Ethereum Foundation and PQShield, NIST STPPA 25 July 2023

Physical

- Handwritten signature
- Sealed envelope
- Ballot box
- Identity badge
- Cash

Digital

- Digital Signature
- Encrypted message
- E-Voting scheme
- Credential system
- Digital account

Physical

- Handwritten signature
- Sealed envelope
- Ballot box
- Identity badge
- Cash

Digital

- Digital Signature
- Encrypted message
- E-Voting scheme
- Credential system
- Digital account

Arguably more secure

Physical	Digital
Handwritten signature	Digital Signature
Sealed envelope	Encrypted message
Ballot box	E-Voting scheme
Identity badge	Credential system
Cash	Digital account
Lie detector	Zero-Knowledge proof

Arguably more secure

Zero-Knowledge Proofs

The digital language of truth

- Everything I say in zero-knowledge is true.
- I can choose to say nothing at all.
- Everything I do not say is perfectly hidden.

Applications

Verifiable FHE

Verifiable mixnets

Verifiable outsourced computation

Verifiable formal verification

Attested sensors

Scalable blockchains

Scalability:

- We do not want to redo a large computation.
- We care about the outcome being correct.
- We might not even have the original data in full.

Applications

Actively secure MPC

Random beacons

Range proofs

Membership proofs

Blind signatures

Code based digital signatures

Building block:

- Generally a useful building block for other cryptographic primitives.
- A hammer when you can’t think of anything smarter.

Applications

Privacy:

- There is information that must be kept private.
- E.g. whistleblower can say they are an employee without revealing identity.

Secret information games

Anonymous cryptocurrency

Whistleblowers

Compliant closed source algorithms

Anonymous credentials

Solvency proofs

Applications

Scalability and Privacy:

- We want to prove a large computation is done correctly.
- We also want to keep some inputs private.

Machine learning checks and balances

Blocklists

Storage proofs

Captcha

Persistent pseudonyms

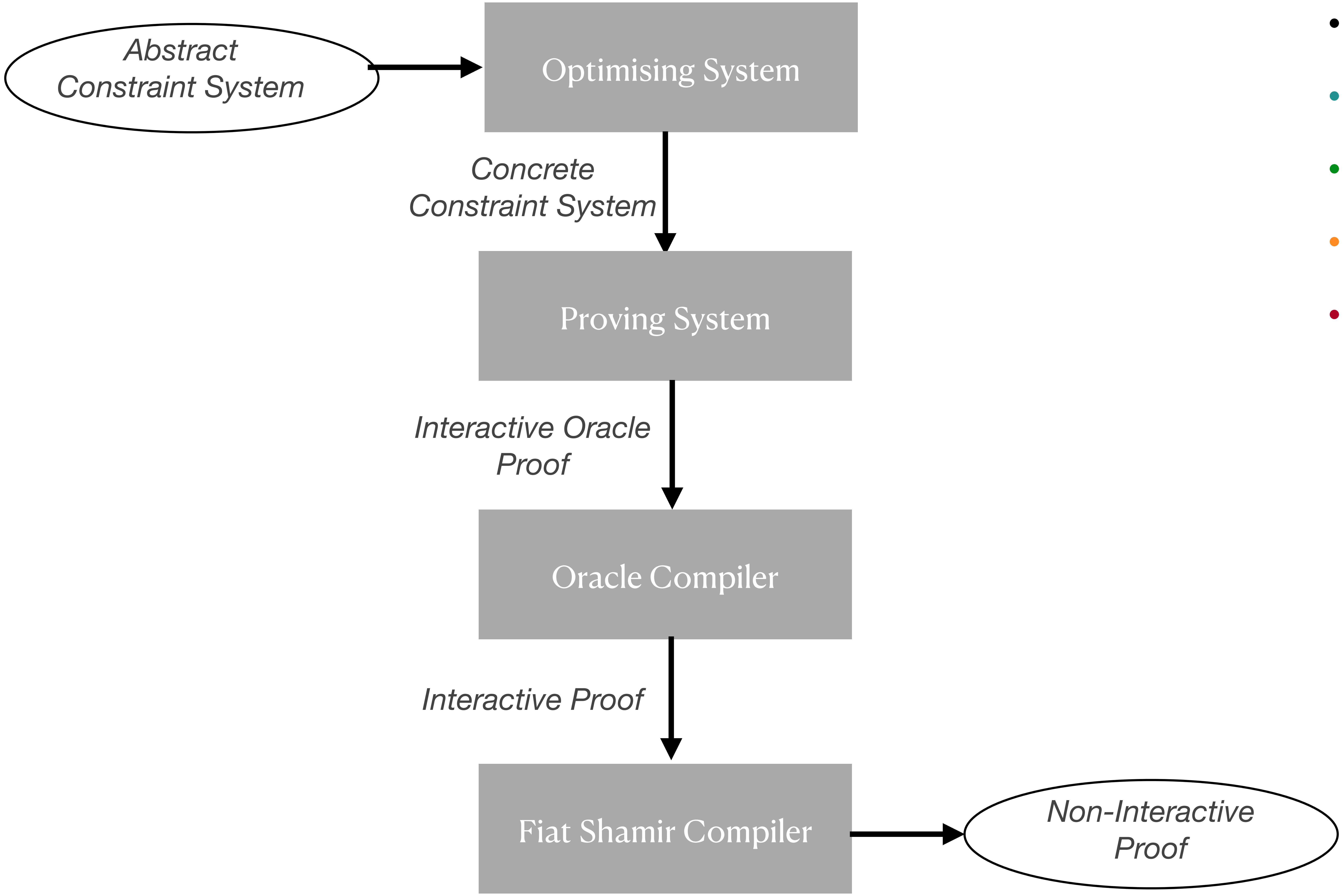
Proof of exploits

Applications

Vast

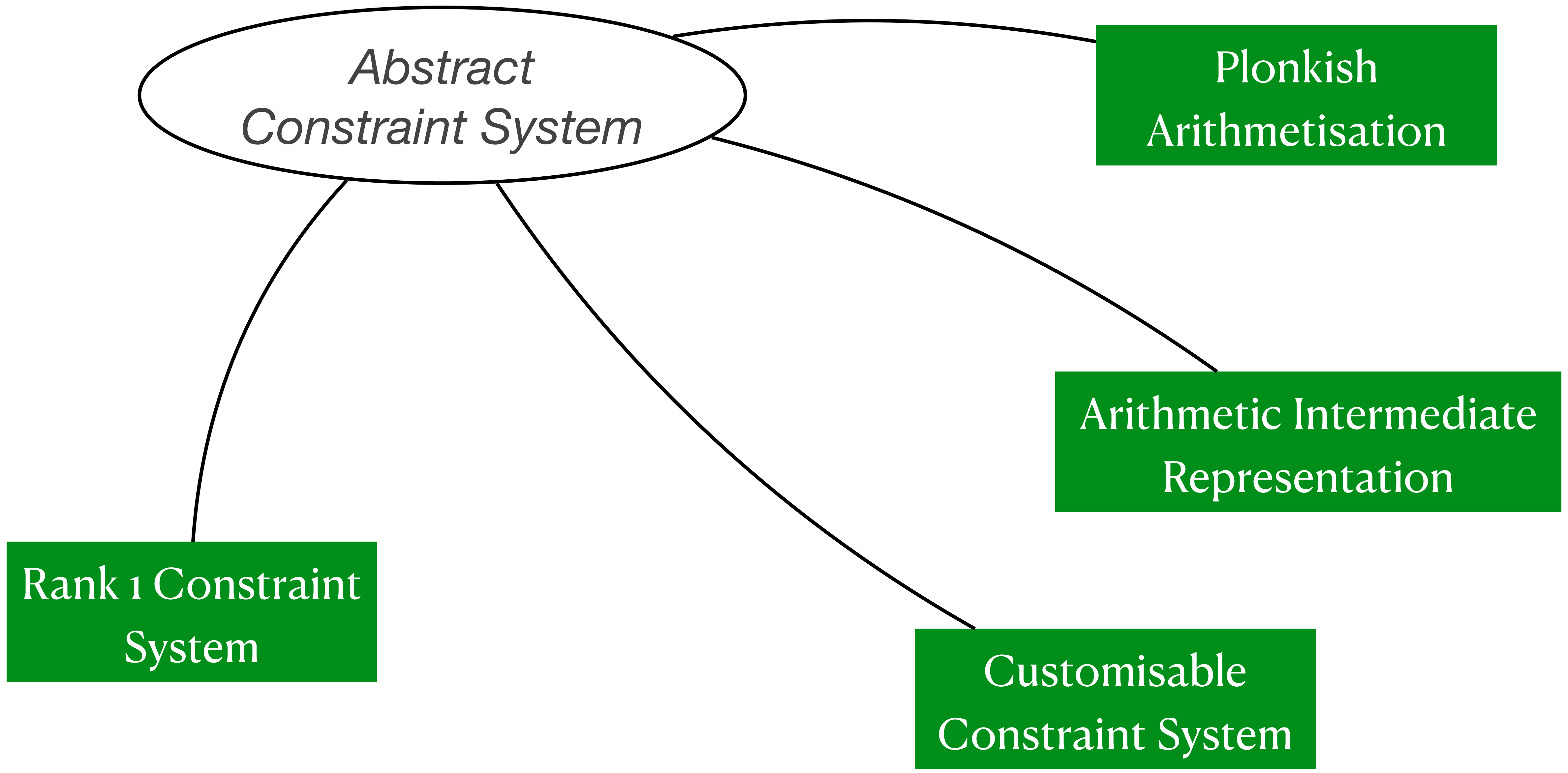
- Verifiable FHE
- Verifiable mixnets
- Verifiable outsourced computation
- Verifiable formal verification
- Attested sensors
- Scalable blockchains
- Actively secure MPC
- Random beacons
- Range proofs
- Membership proofs
- Blind signatures
- Code based digital signatures
- Secret information games
- Anonymous cryptocurrency
- Whistleblowers
- Compliant closed source algorithms
- Anonymous credentials
- Solvency proofs
- Machine learning checks and balances
- Blocklists
- Storage proofs
- Captcha
- Persistent pseudonyms
- Proof of exploits

Structure of a ZKP



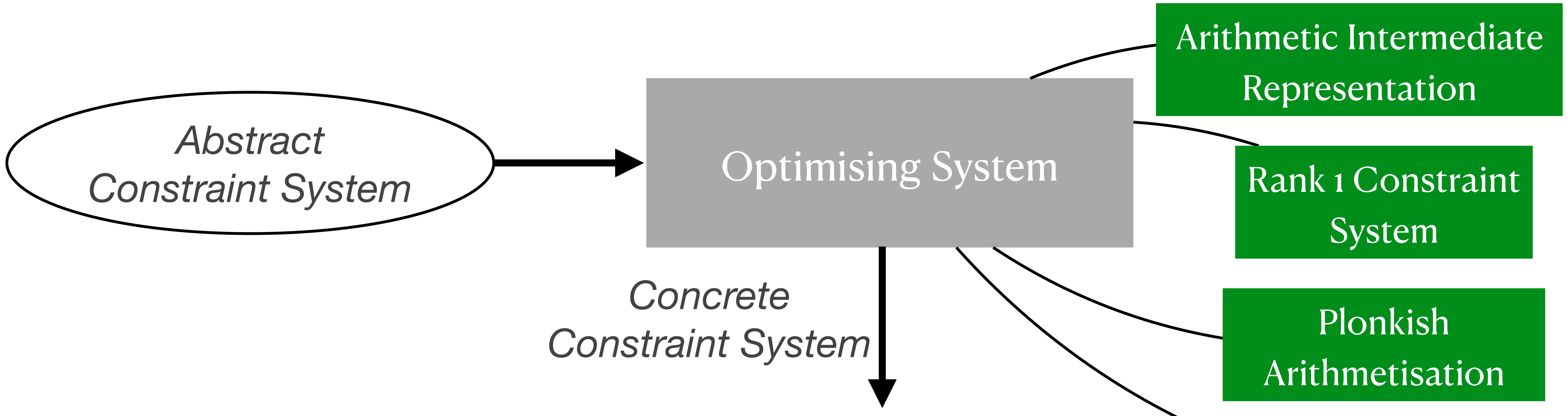
- Arithmetise
- Optimise
- Polynomial-ise
- Cryptographically Compile
- Deterministic-ify

Structure of a ZKP


Abstract Constraint System
Plonkish Arithmetisation
Arithmetic Intermediate Representation
Rank 1 Constraint System
Customisable Constraint System

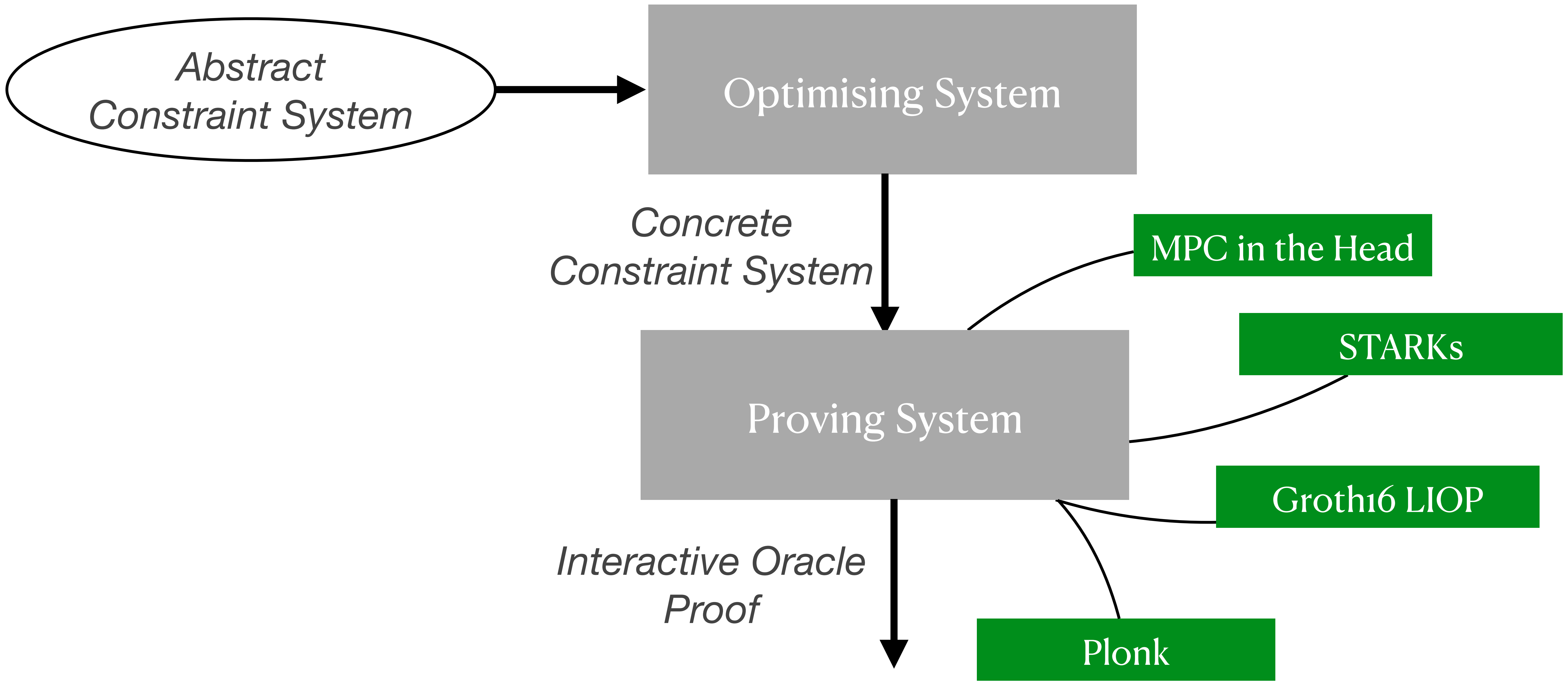
- Arithmetise:
  - Define the language.
  - What can people say or not say?
  - How must they say it?

Structure of a ZKP


Abstract Constraint System
Optimising System
Concrete Constraint System
Arithmetic Intermediate Representation
Rank 1 Constraint System
Plonkish Arithmetisation
Customisable Constraint System

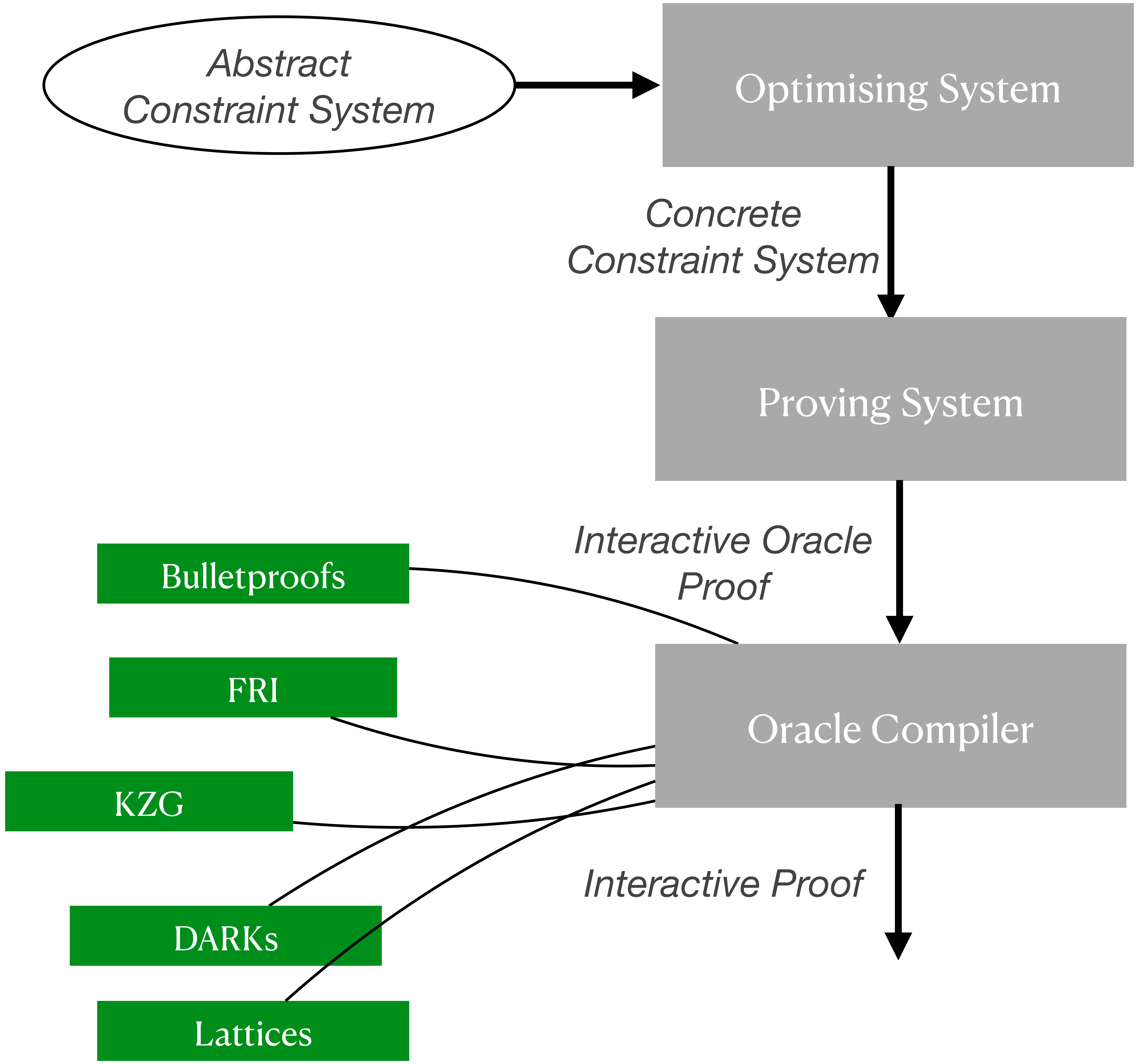
- Arithmetise
- Optimise:
  - Humans write constraints badly.
  - Programs can optimise human written constraints.
  - Require that the original meaning is not lost.
  - Optimiser inherently tied to constraint system.

Structure of a ZKP



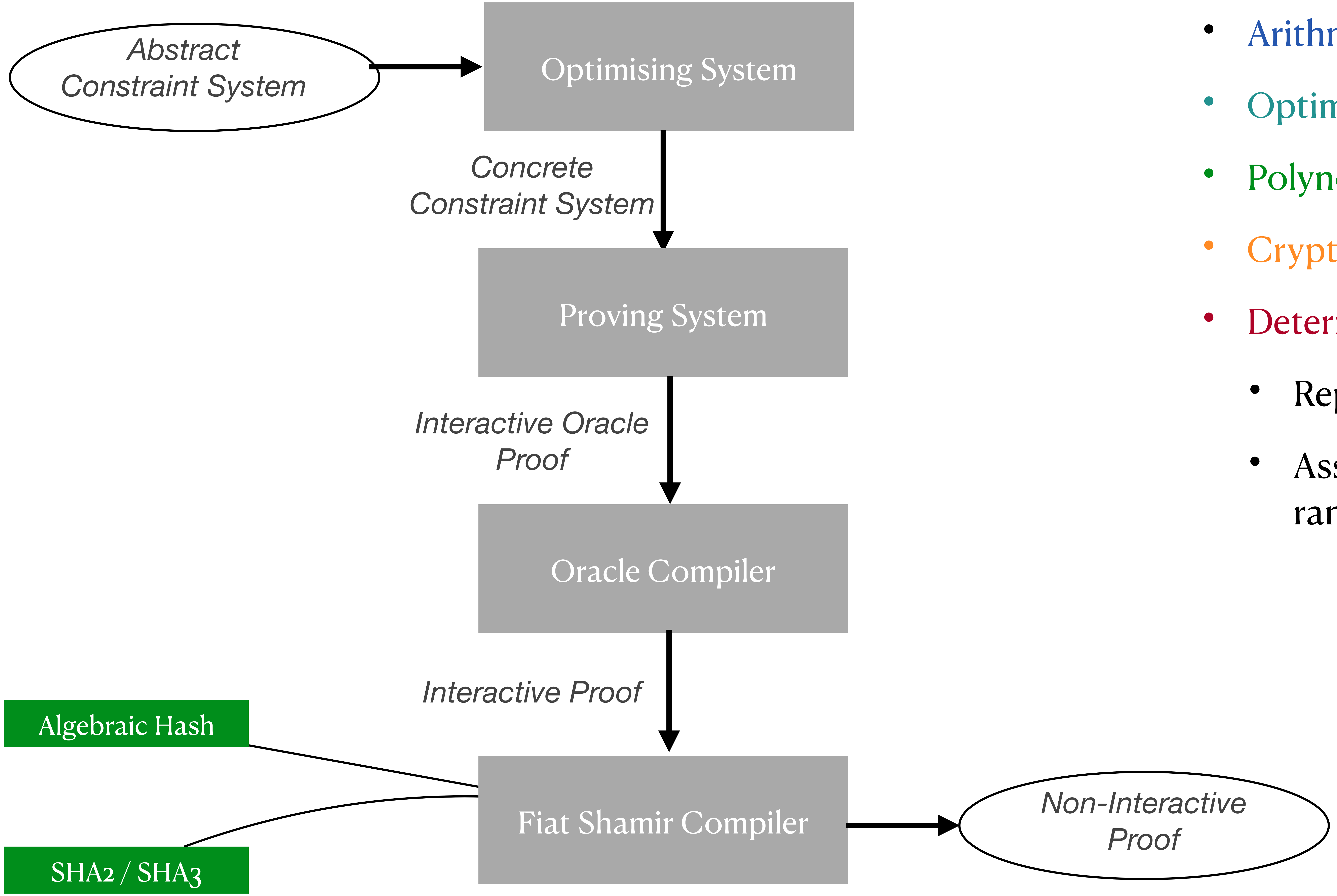
- Arithmetise
- Optimise
- Polynomial-ise:
  - Information theoretically secure proving system.
  - Different proving systems target different constraint systems.
  - This is where the hard mathematics is.

Structure of a ZKP



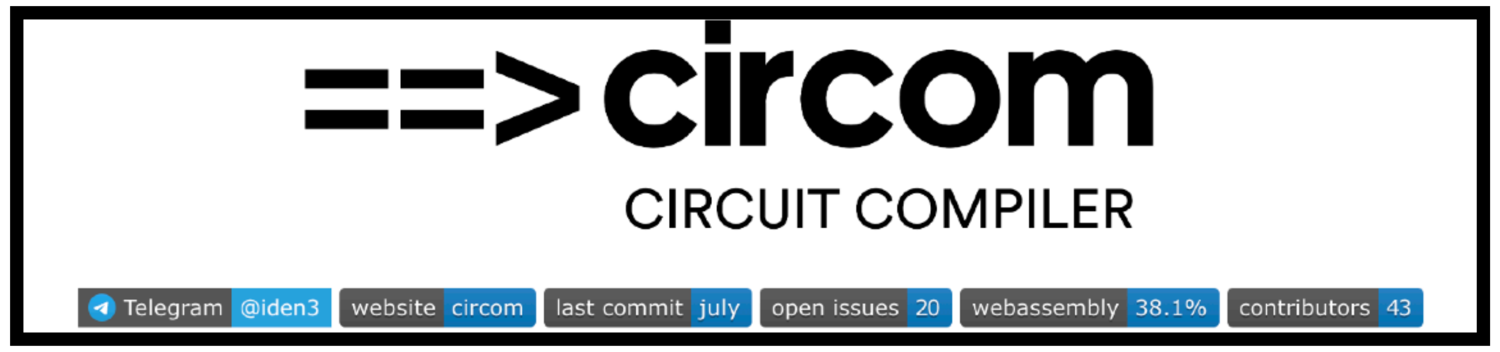
- Arithmetise
- Optimise
- Polynomial-ise
- Cryptographically Compile:
    - Use a polynomial commitment scheme.
    - Independent from the proving system.
    - Determines many features:
        - Hardness assumption
        - Efficiency
        - Trusted setup
        - Proof size

Structure of a ZKP



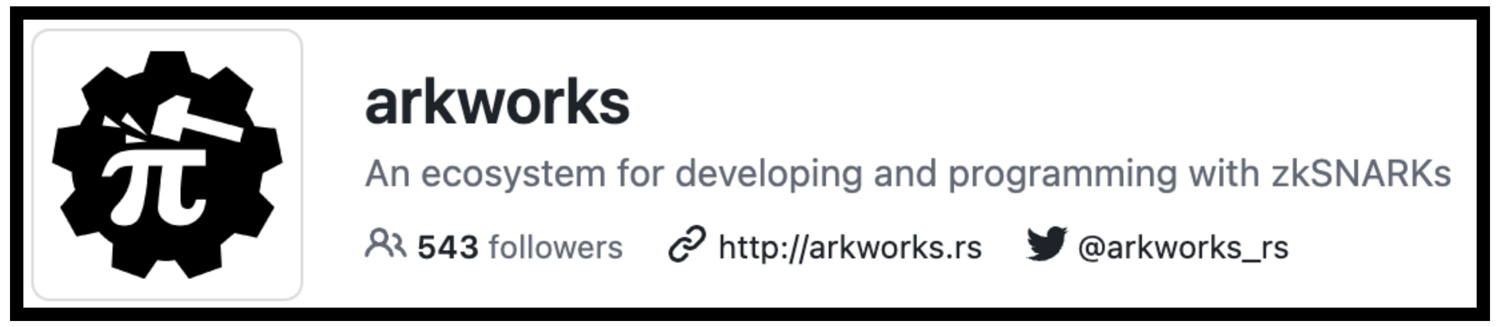
- Arithmetise
- Optimise
- Polynomial-ise
- Cryptographically Compile
- Deterministic-ify:
    - Replace true randomness with hashes.
    - Assumes hash functions behave like random oracles.

Implementations


==>circom

CIRCUIT COMPILER

Telegram @iden3 | website circom | last commit july | open issues 20 | webassembly 38.1% | contributors 43

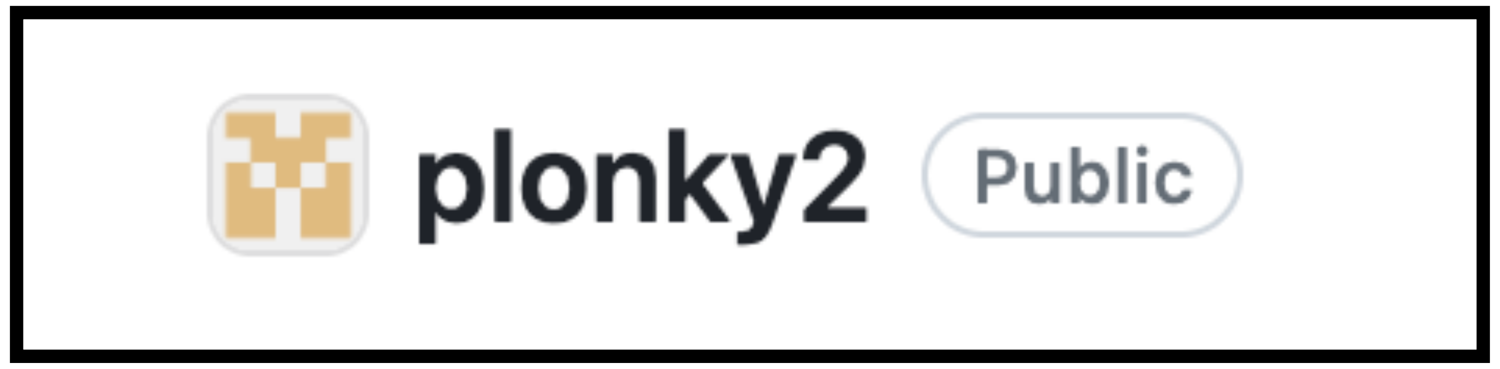

arkworks

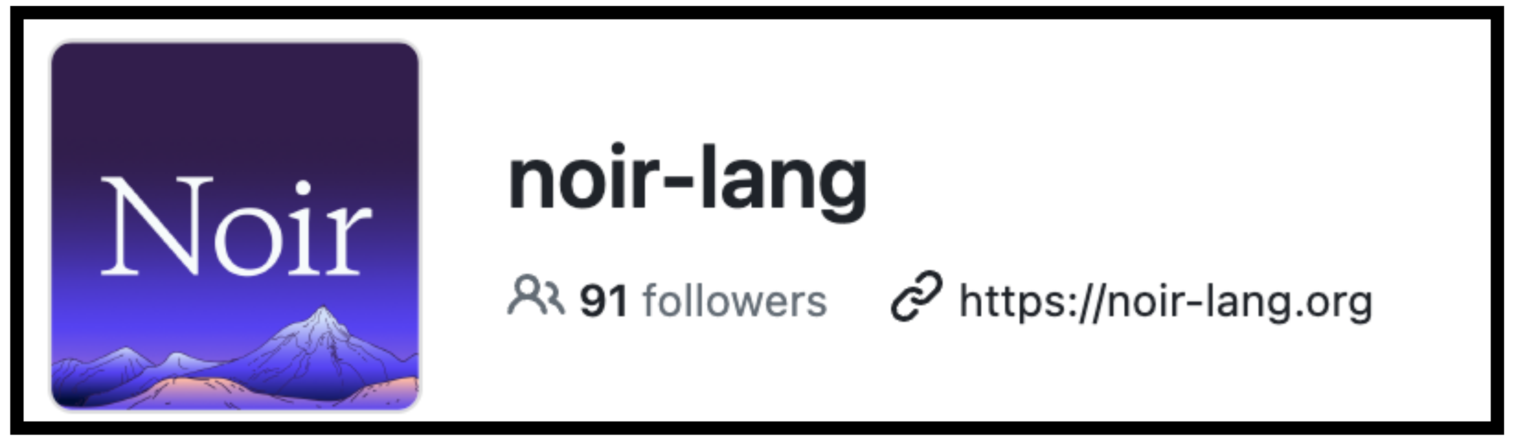
An ecosystem for developing and programming with zkSNARKs

543 followers http://arkworks.rs @arkworks_rs

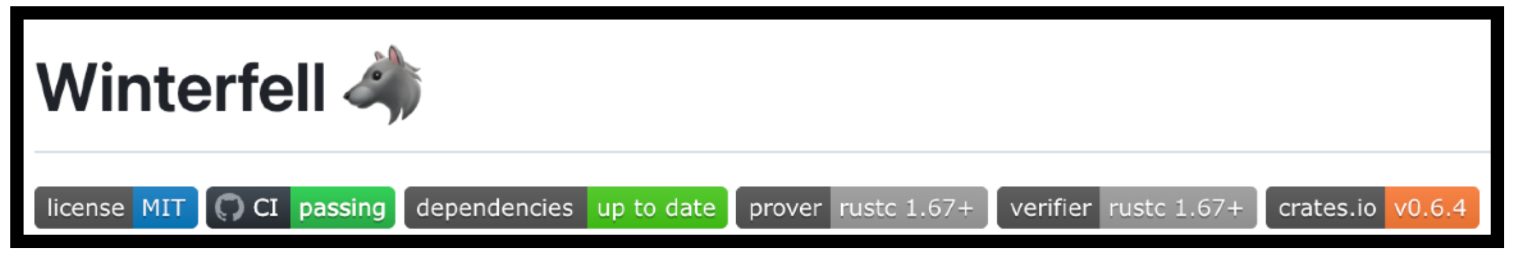
halo2 crates.io v0.1.0-beta.2

bellman Public


plonky2 Public


noir-lang

91 followers https://noir-lang.org

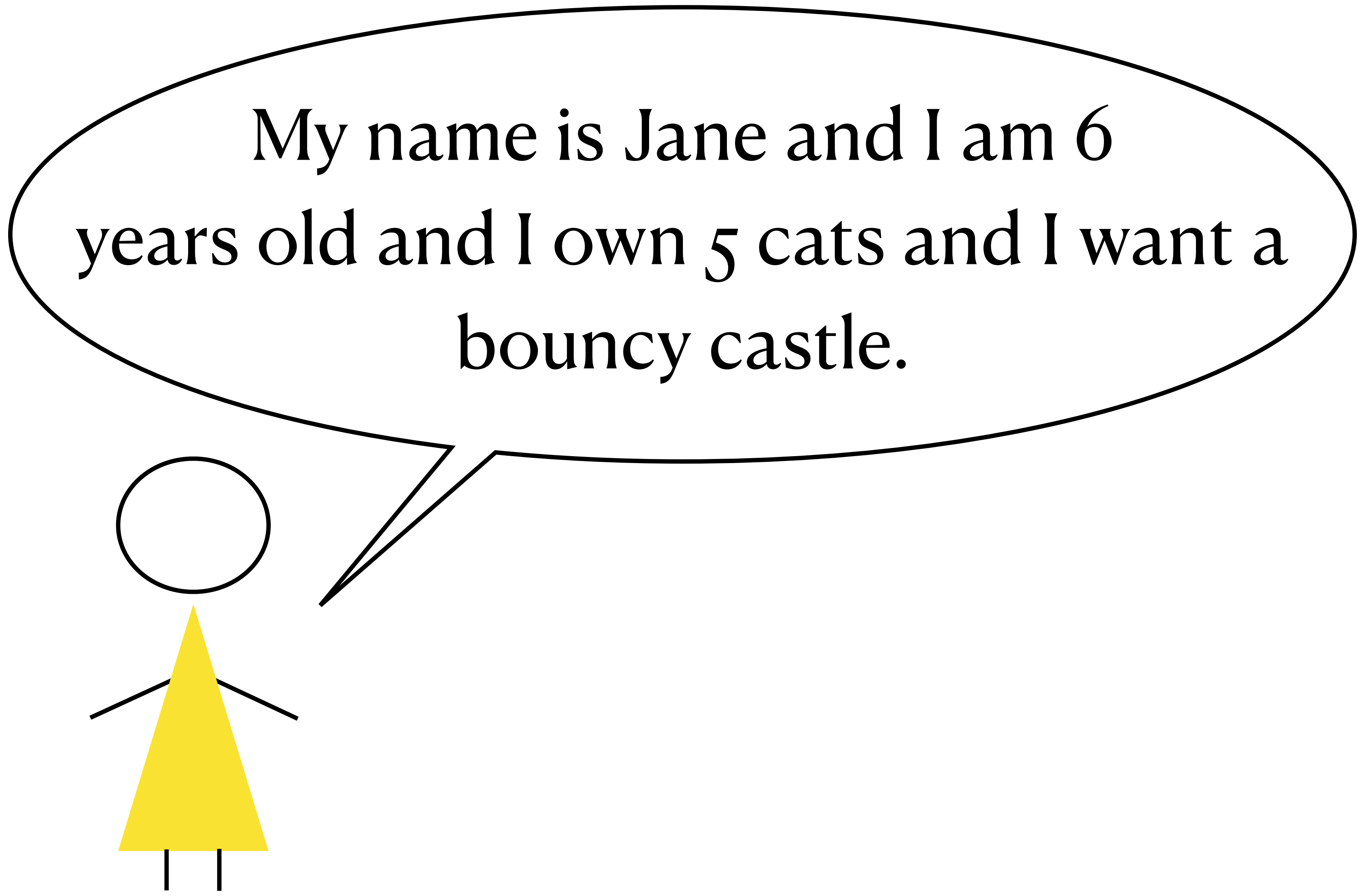

Winterfell

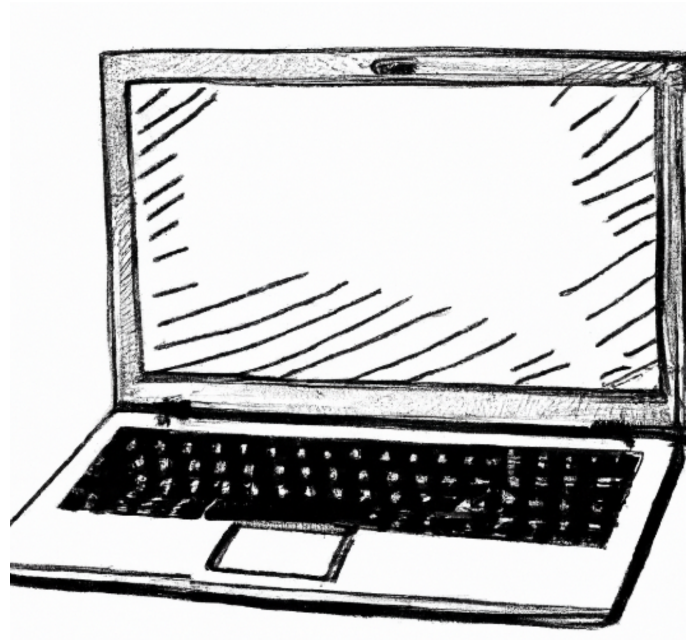
license MIT | CI passing | dependencies up to date | prover rustc 1.67+ | verifier rustc 1.67+ | crates.io v0.6.4

- Implementations target different layers of the system.
- Some are more advanced than others.
- ZK is used in production for specific statements.
- ZK is quite fast! And getting faster.
- Currently it requires expert knowledge to “talk in zk”.

Zero-Knowledge Proofs

The digital language of truth


My name is Jane and I am 6 years old and I own 5 cats and I want a bouncy castle.



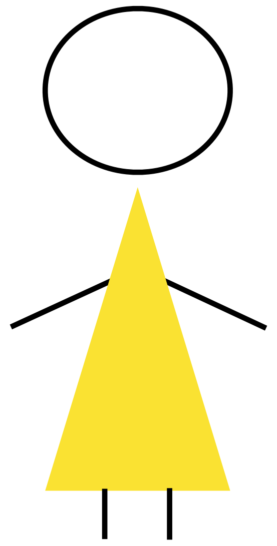
Translater

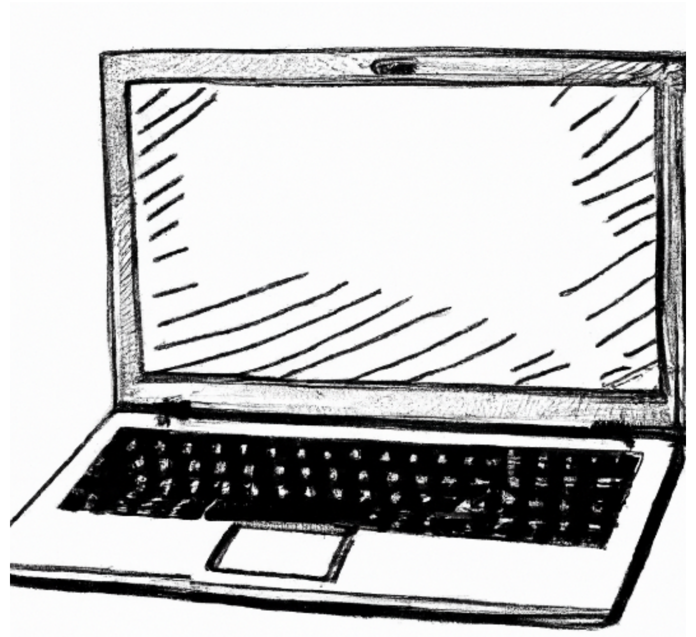
- Everything I say in zero-knowledge is true.
- I can choose to say nothing at all.
- Everything I do not say is perfectly hidden.

Zero-Knowledge Proofs

The digital language of truth

My name is Jane and I am 6 years old and I own 5 cats and I want a bouncy castle.





Translater

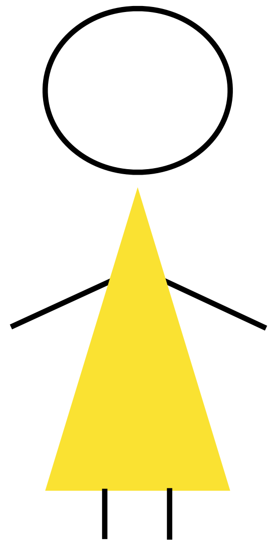
I know the secret key belonging to the identity Jane.

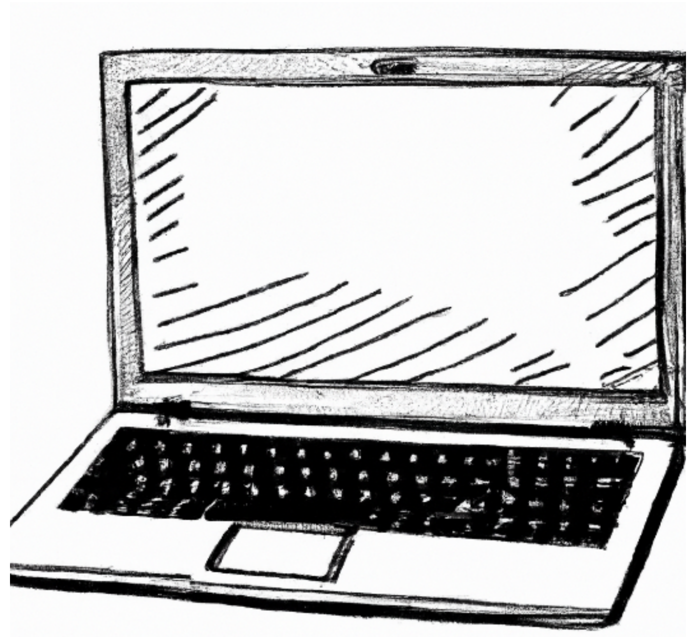
- Everything I say in zero-knowledge is true.
- I can choose to say nothing at all.
- Everything I do not say is perfectly hidden.

Zero-Knowledge Proofs

The digital language of truth

My name is Jane and I am 6 years old and I own 5 cats and I want a bouncy castle.





Translater

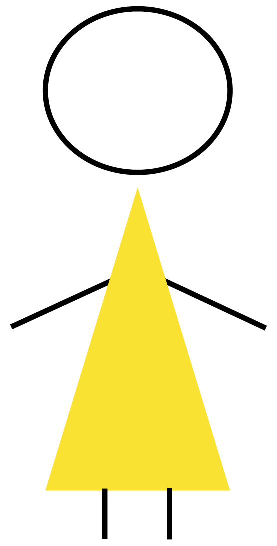
Identity Jane is registered in a trusted database with the age attribute equal to 6.

- Everything I say in zero-knowledge is true.
- I can choose to say nothing at all.
- Everything I do not say is perfectly hidden.

Zero-Knowledge Proofs

The digital language of truth

My name is Jane and I am 6 years old and I own 5 cats and I want a bouncy castle.



Translater

This would not accurately convey the meaning!

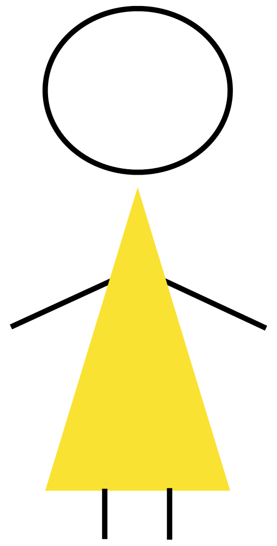
I know the secret ownership tag relating to 5 cats.

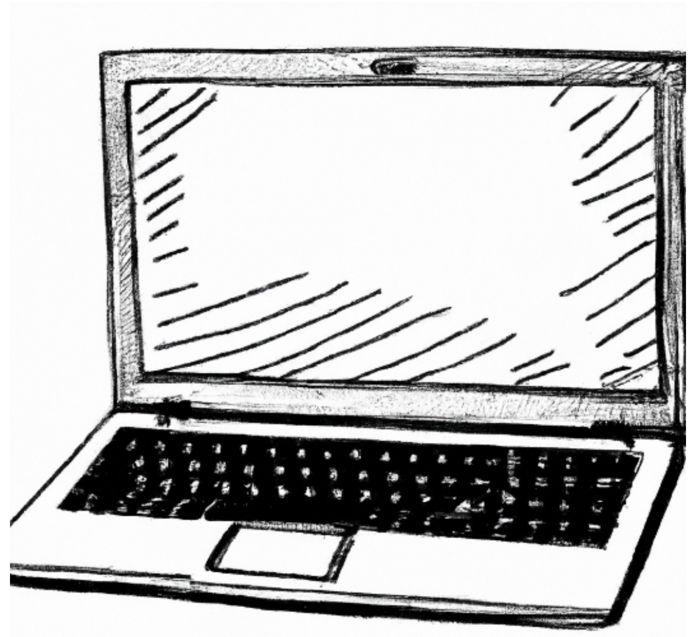
- Everything I say in zero-knowledge is true.
- I can choose to say nothing at all.
- Everything I do not say is perfectly hidden.

Zero-Knowledge Proofs

The digital language of truth

My name is Jane and I am 6 years old and I own 5 cats and I want a bouncy castle.





Translater

I know the secret ownership tag relating to 5 unique cats.

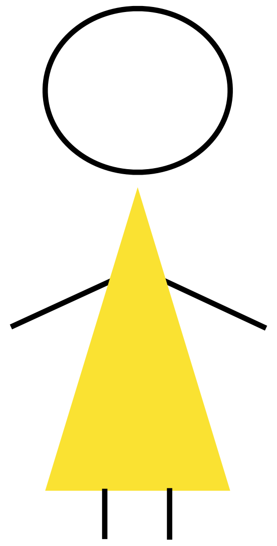
- Everything I say in zero-knowledge is true.
- I can choose to say nothing at all.
- Everything I do not say is perfectly hidden.

Zero-Knowledge Proofs

The digital language of truth

Can only say verifiable statements

My name is Jane and I am 6 years old and I own 5 cats and I want a bouncy castle.

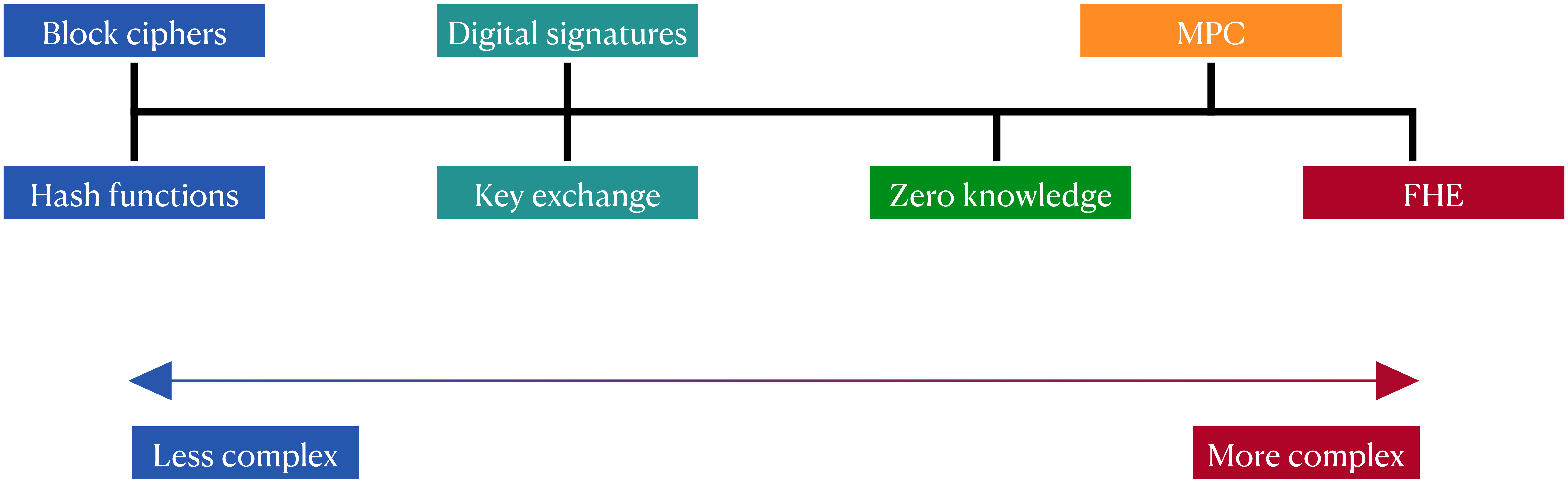


Translater

???????

- Everything I say in zero-knowledge is true.
- I can choose to say nothing at all.
- Everything I do not say is perfectly hidden.

Complexity of Cryptographic Primitives



ZKProof Standardisation Effort

Website

- Global movement to standardise and mainstream advanced cryptography by building a community-driven trust ecosystem.
- Formed in 2018 after top researchers and developers saw technology becoming advanced enough for standards.
- I joined the editorial team in 2021.
- We expect this to be a long process as the community jointly learn best practices.

Education

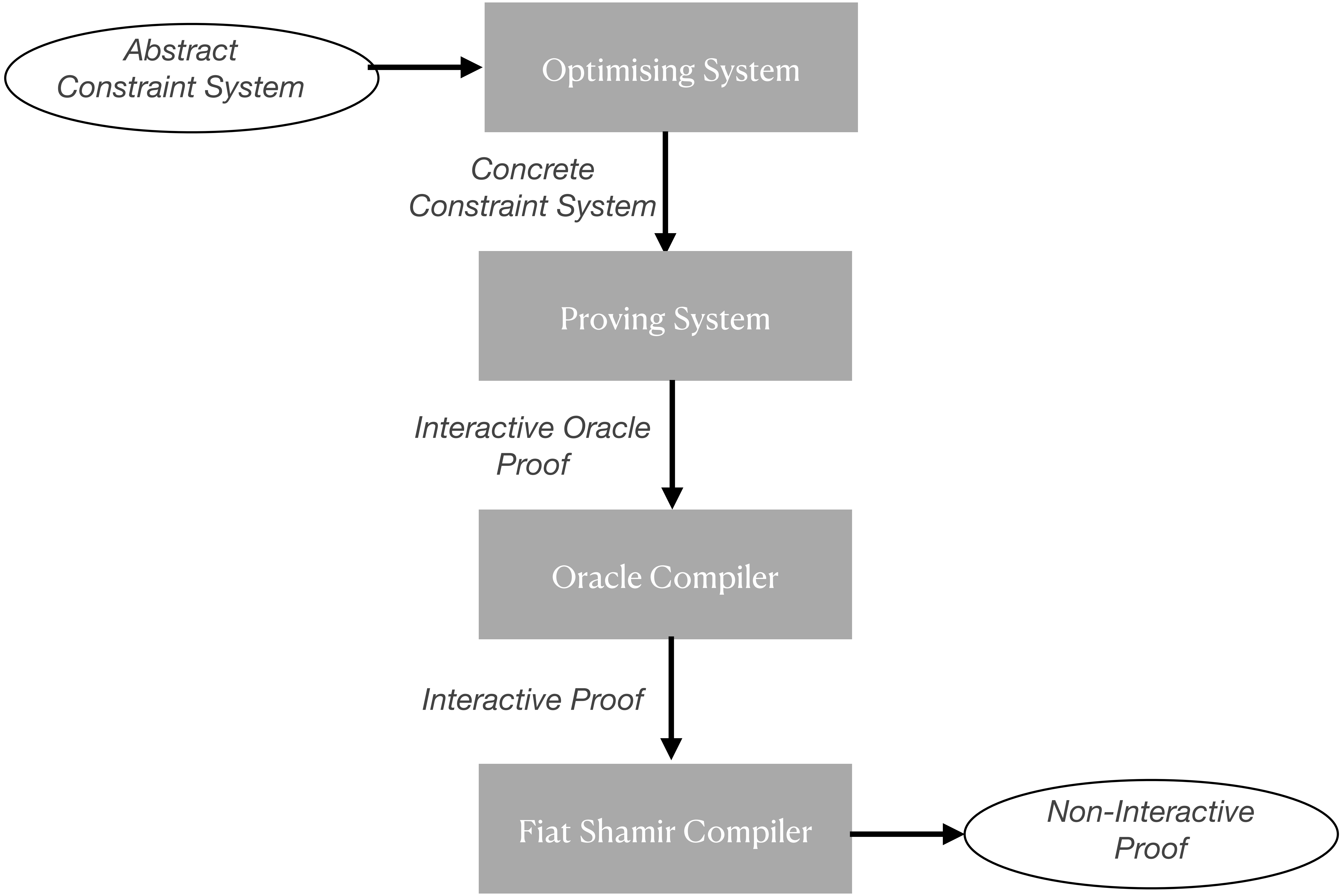
Standards

Community

ZKProof Standardisation Effort

Standards

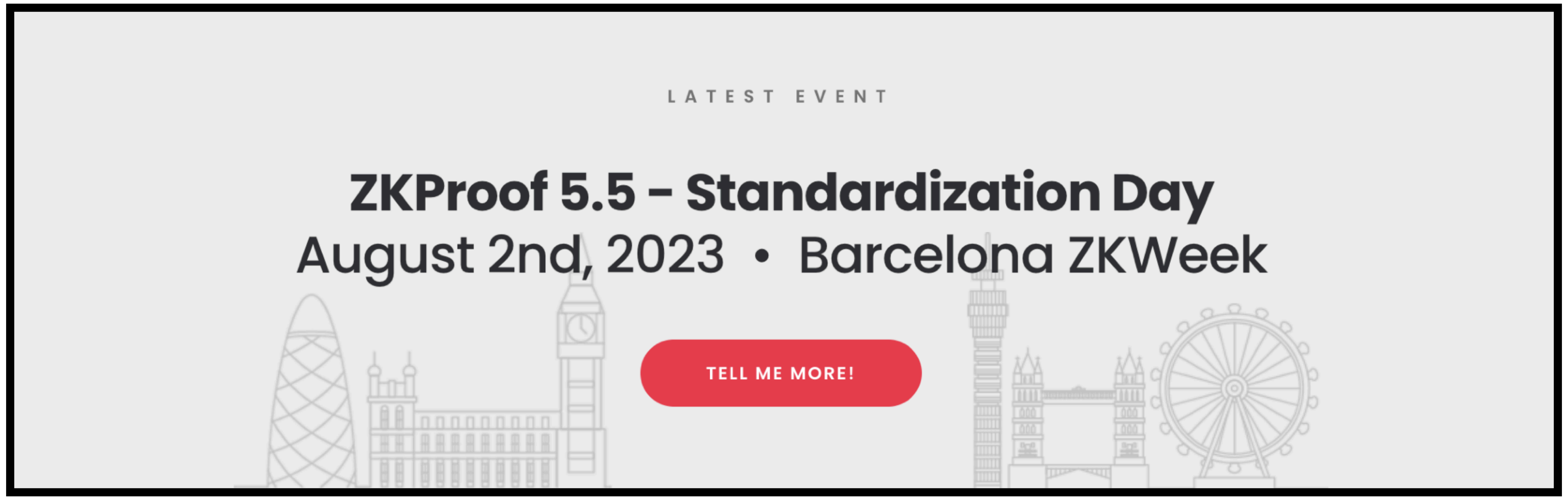
- Working groups write specifications for different proving systems.
- Michele Orru is leading a working group on the Fiat-Shamir compiler.
- I am part of a working group targeting the Plonkish constraint system.
- We’re open to submitting specifications to e.g. IETF once complete with community backing.



ZKProof Standardisation Effort

Community

- Yearly in person ZKProof events.
- Active working groups present results.
- People can propose formation of a new working groups.
- Presentations on applications and research in zero-knowledge.
- Discuss what is and is not working organisationally.



LATEST EVENT

ZKProof 5.5 – Standardization Day

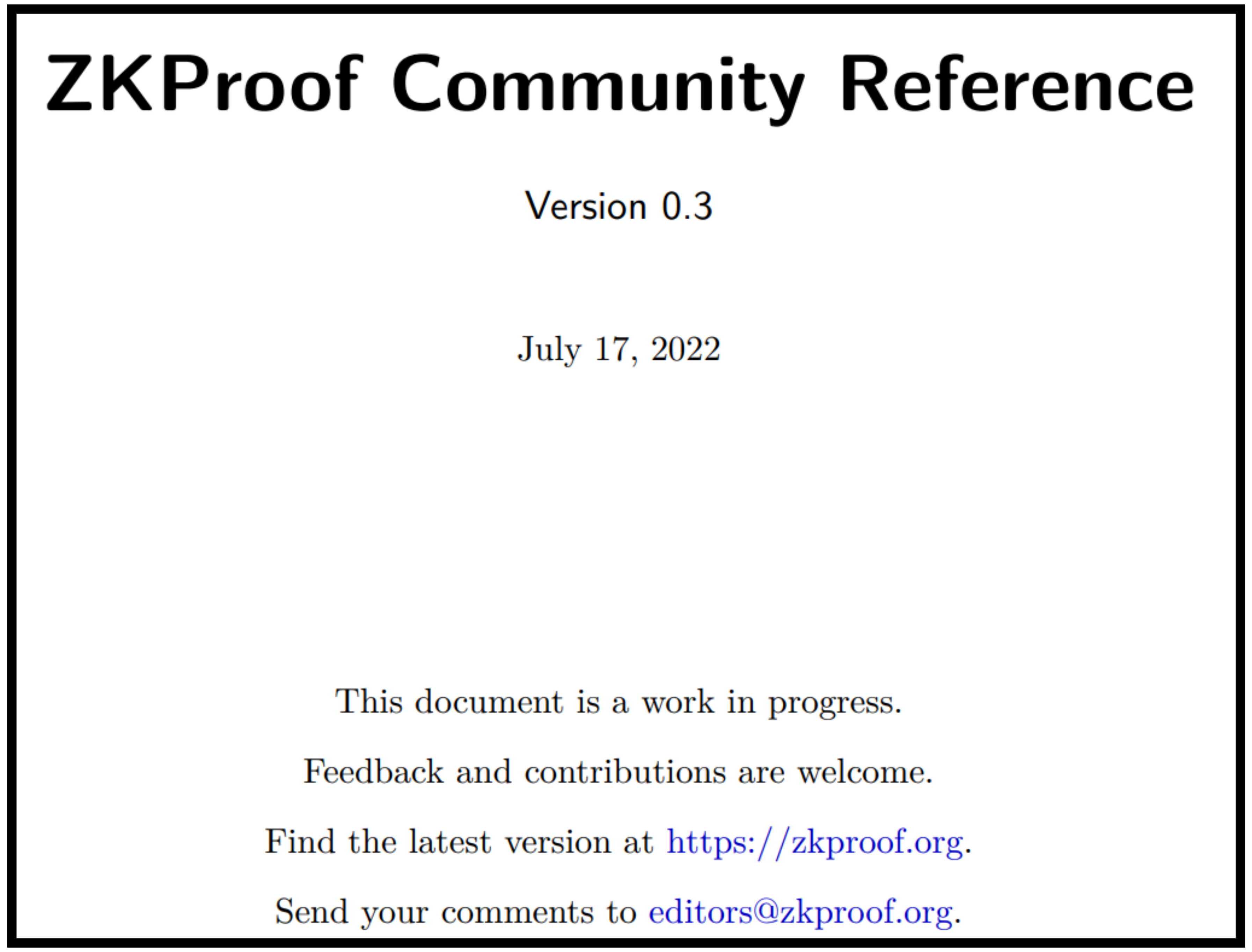
August 2nd, 2023 • Barcelona ZKWeek

ZKProof Standardisation Effort

Docs

Education

- Maintain a community reference document to comprehend terminology, examples, explanations and recommendations.
- Maintain a list of recommended educational resources for people looking to learn about zero-knowledge.



ZKProof Community Reference

Version 0.3

July 17, 2022

This document is a work in progress.

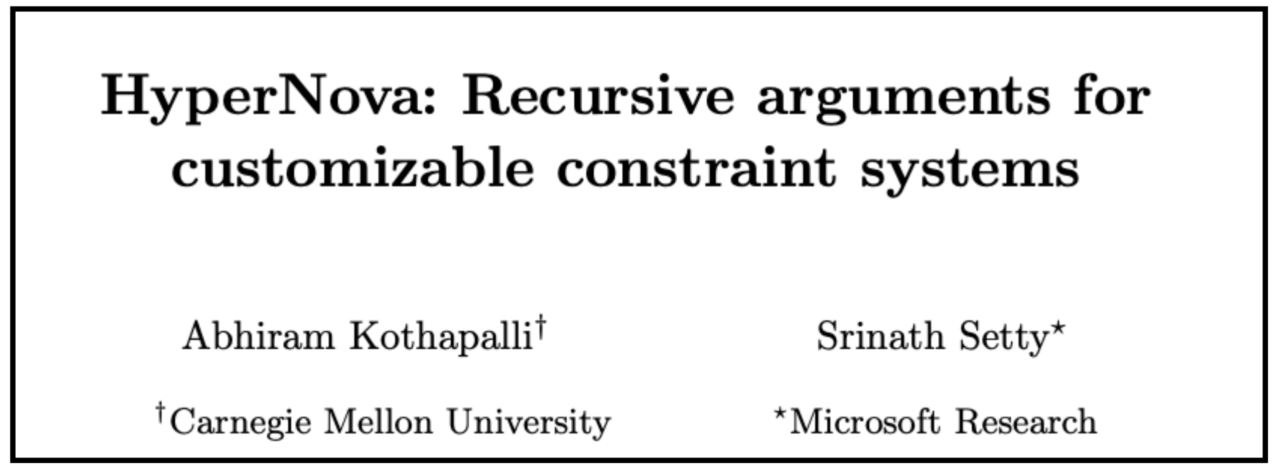
Feedback and contributions are welcome.

Find the latest version at https://zkproof.org.

Send your comments to editors@zkproof.org.

Recent Advances

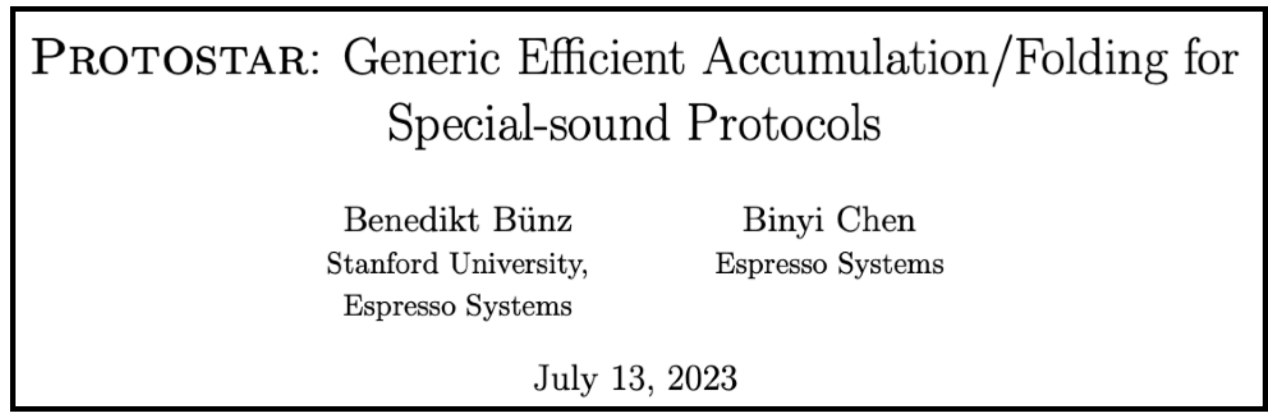
Folding Schemes



HyperNova: Recursive arguments for customizable constraint systems

Abhiram Kothapalli[†] Srinath Setty[⋆]

[†] Carnegie Mellon University [⋆] Microsoft Research



PROTOSTAR: Generic Efficient Accumulation/Folding for Special-sound Protocols

Benedikt Bünz
Stanford University,
Espresso Systems

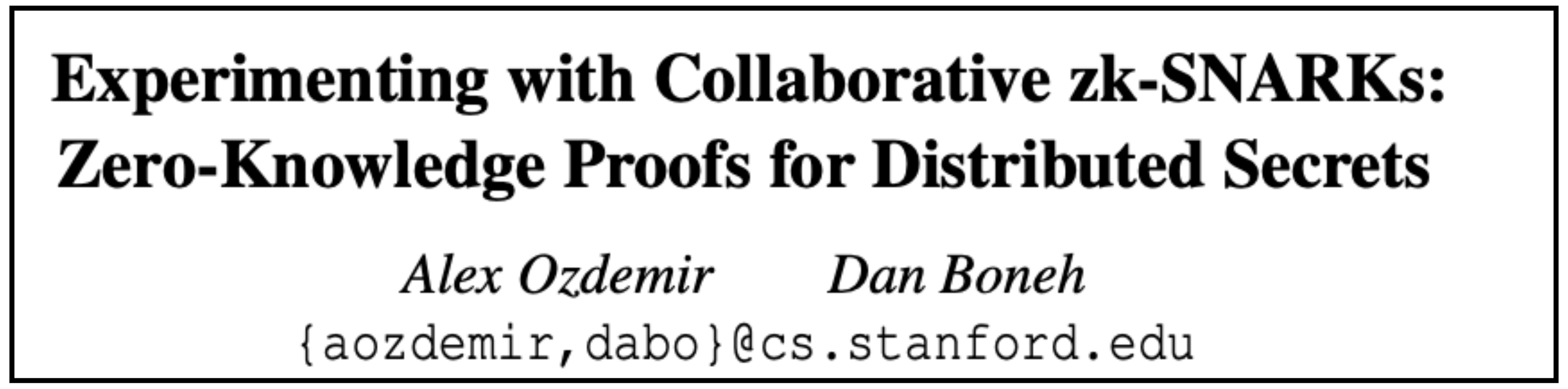
Binyi Chen
Espresso Systems

July 13, 2023

- A proof of a proof of a proof of a proof is faster than just one big proof.

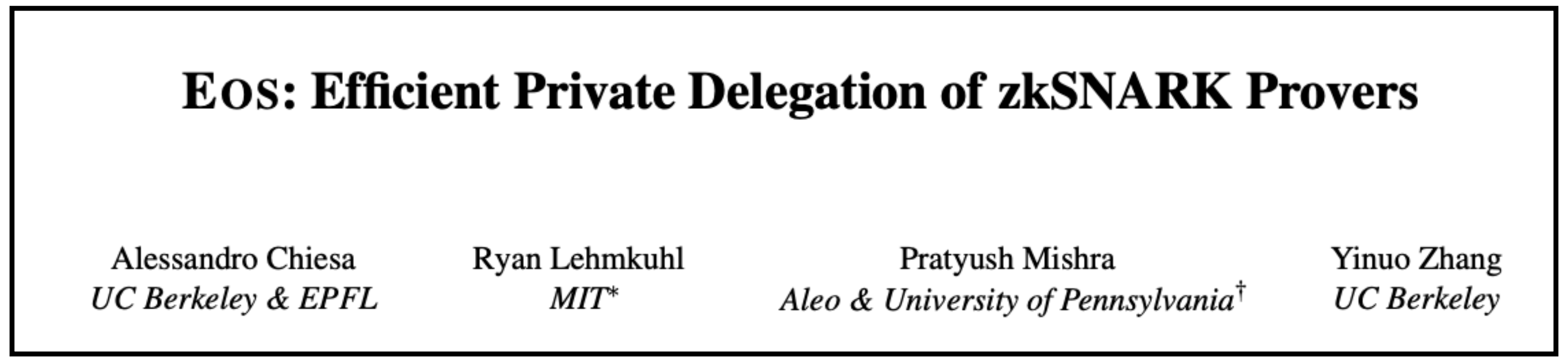
Recent Advances

Distributed Provers



Experimenting with Collaborative zk-SNARKs: Zero-Knowledge Proofs for Distributed Secrets

Alex Ozdemir *Dan Boneh*
{aozdemir,dabo}@cs.stanford.edu



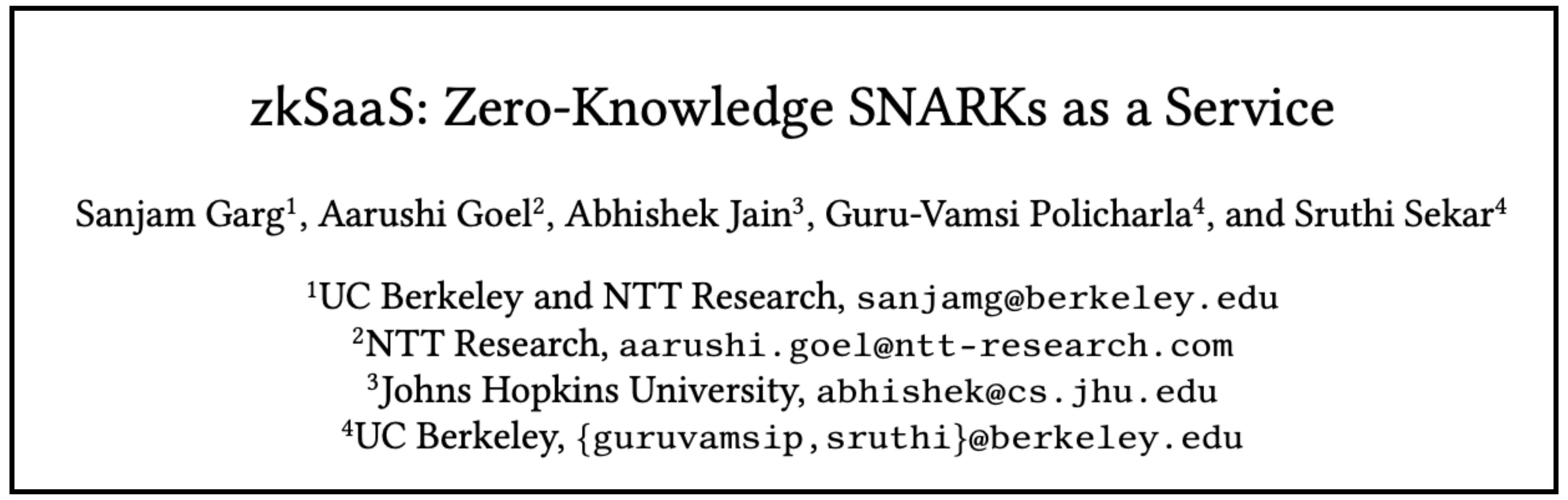
Eos: Efficient Private Delegation of zkSNARK Provers

Alessandro Chiesa
UC Berkeley & EPFL

Ryan Lehmkuhl
*MIT**

Pratyush Mishra
Aleo & University of Pennsylvania†

Yinuo Zhang
UC Berkeley



zkSaaS: Zero-Knowledge SNARKs as a Service

Sanjam Garg[1], Aarushi Goel[2], Abhishek Jain[3], Guru-Vamsi Policharla[4], and Sruthi Sekar[4]

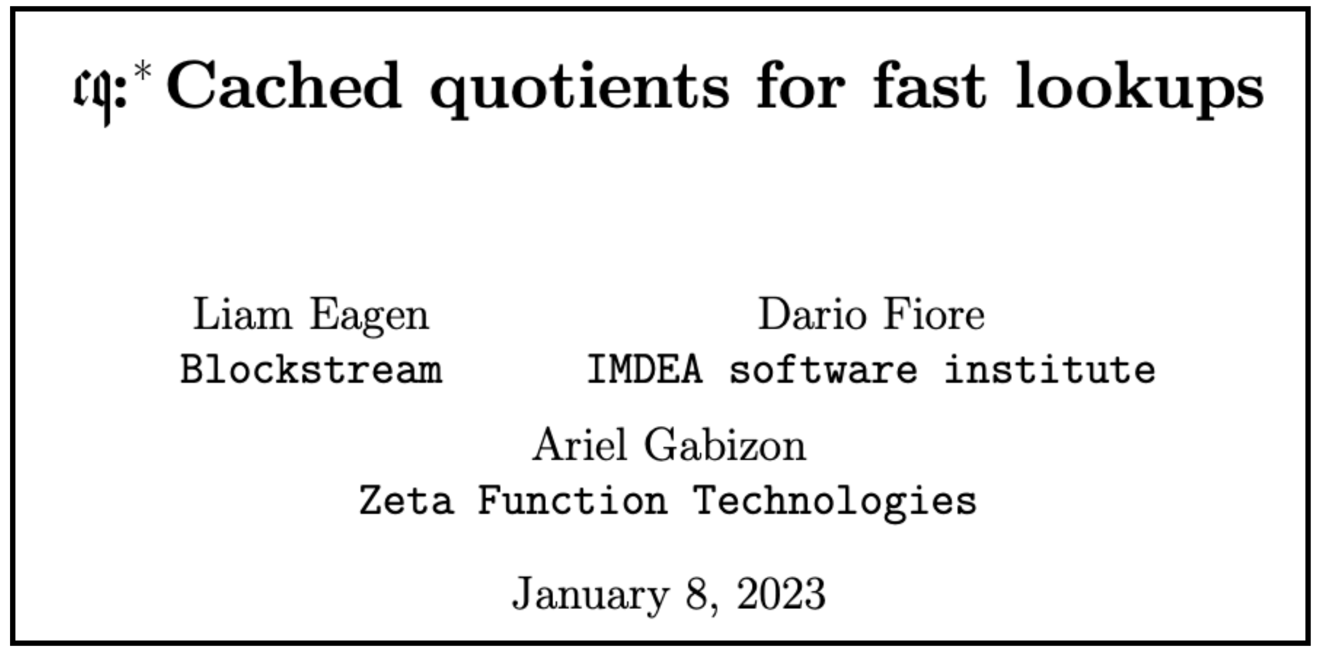
[1]UC Berkeley and NTT Research, sanjamg@berkeley.edu
[2]NTT Research, aarushi.goel@ntt-research.com
[3]Johns Hopkins University, abhishek@cs.jhu.edu
[4]UC Berkeley, {guruvamsip,sruthi}@berkeley.edu

- ZK provers can be expensive.
- We can outsource their generation to parallel processors.
- We can distribute any private data.

Recent Advances

Lookup Arguments

- Lookup tables contain precomputed information.
- They are useful in traditional computing languages.
- They are also useful in zk computing languages.



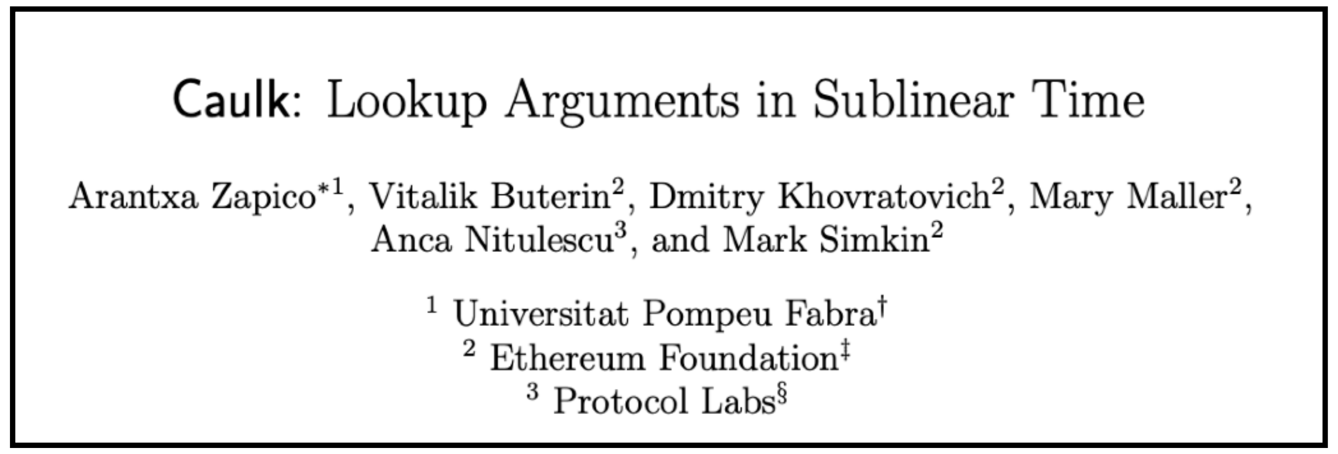
cq:* Cached quotients for fast lookups

Liam Eagen
`Blockstream`

Dario Fiore
`IMDEA software institute`

Ariel Gabizon
`Zeta Function Technologies`

January 8, 2023



Caulk: Lookup Arguments in Sublinear Time

Arantxa Zapico*[1], Vitalik Buterin[2], Dmitry Khovratovich[2], Mary Maller[2], Anca Nitulescu[3], and Mark Simkin[2]

[1] Universitat Pompeu Fabra†
[2] Ethereum Foundation‡
[3] Protocol Labs§

Final Remarks

- I personally believe we are ready now to write formal specifications for popular and established proving systems.
- Without the standards in place, it is difficult for larger companies to justify the risk of what is seen as highly experimental technology.
- Zero-knowledge is so useful that many startups are taking the risk anyway.
- The more support we receive, the better our chances of success.

Thank-you for listening!