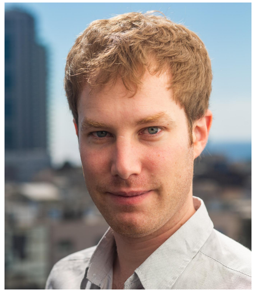
qedit

ZKProof.org

Amsterdam
2019-10-29

zkInterface

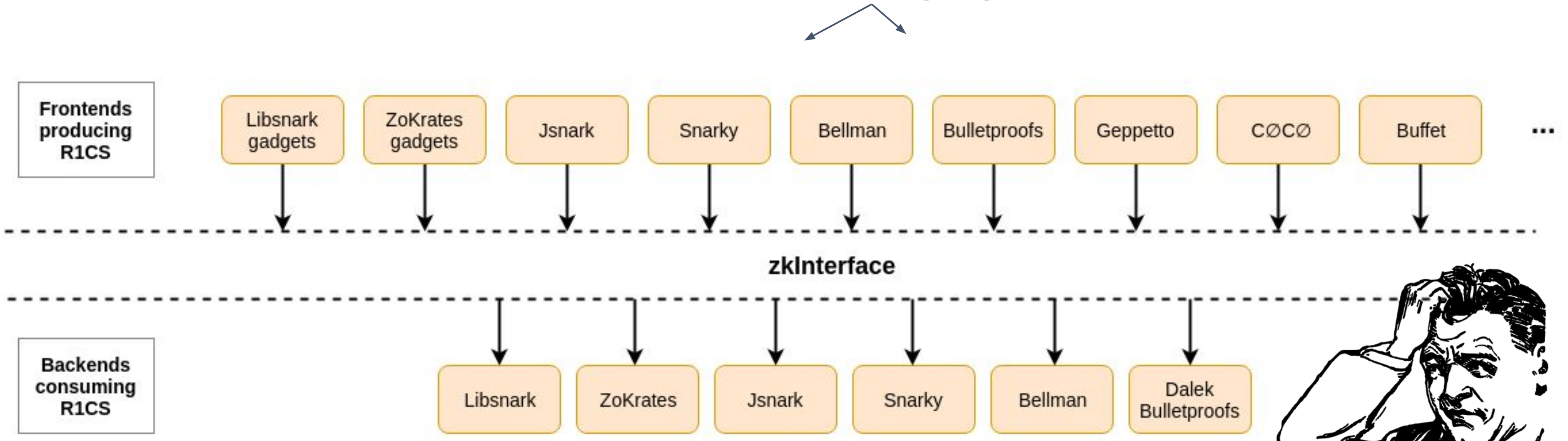
Zero-Knowledge Interoperability

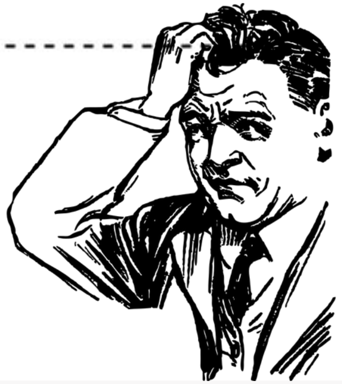


Aurélien Nicolas
Protocol Engineer

zkInterface Concept

Choose a **language**



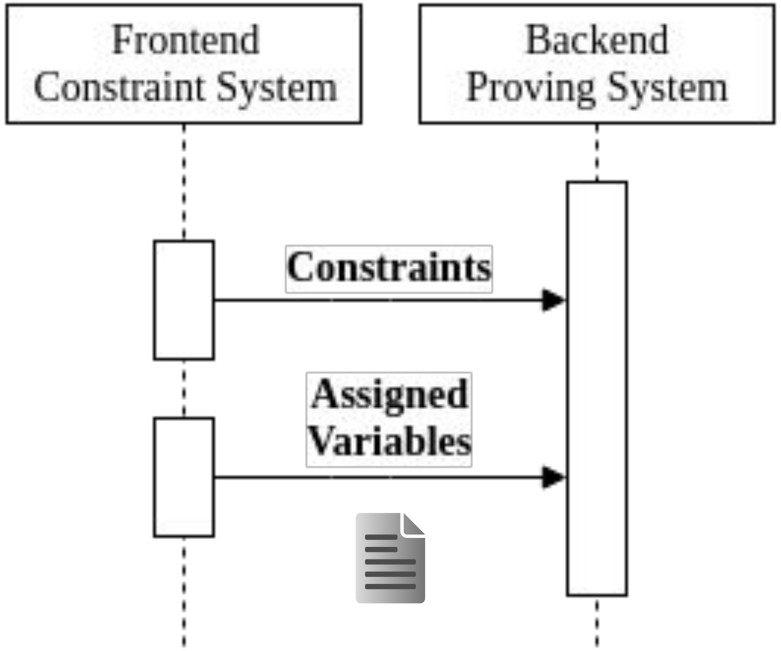


Choose a **proof system**

zkInterface Design

Based on **message passing**

- Describe the computation
- Pass witness values



zkInterface Design

- Compatible with varied approaches
- Extensible, performant
- Flexible cross-platform deployment

Spec & Implementations

github.com/QED-it/zkinterface

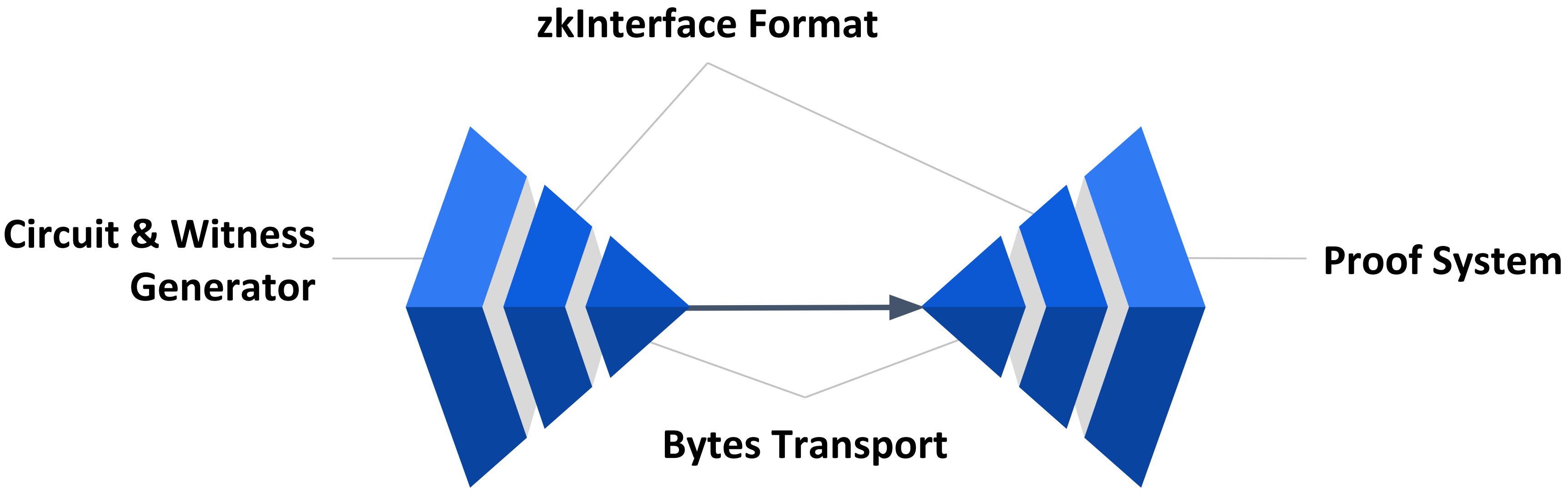
```
table R1CSConstraints {
   constraints    :[BilinearConstraint];
}

table BilinearConstraint {
   linear_combination_a :VariableValues;
   linear_combination_b :VariableValues;
   linear_combination_c :VariableValues;
}

table AssignedVariables {
   values :VariableValues;
}

…
```

Deployment


zkInterface Format
Circuit & Witness Generator
Proof System
Bytes Transport

Deployment

Move zkInterface messages between software components.

Rust / C++	Function calls
Webbrowser	Webassembly
Cross-platform	HTTP

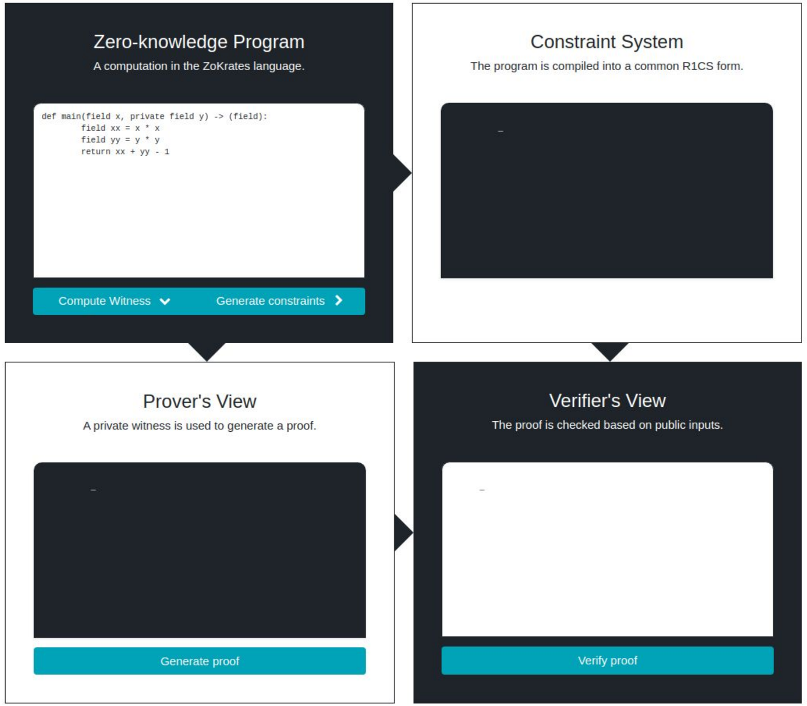
Use-case: Web Apps

ZK scripts in the browser

- A witness generator
- A proving system

Demo

qed-it.github.io/zkinterface-wasm-demo/



Zero-knowledge Program

A computation in the ZoKrates language.

```
def main(field x, private field y) -> (field):
        field xx = x * x
        field yy = y * y
        return xx + yy - 1
```

Constraint System

The program is compiled into a common R1CS form.

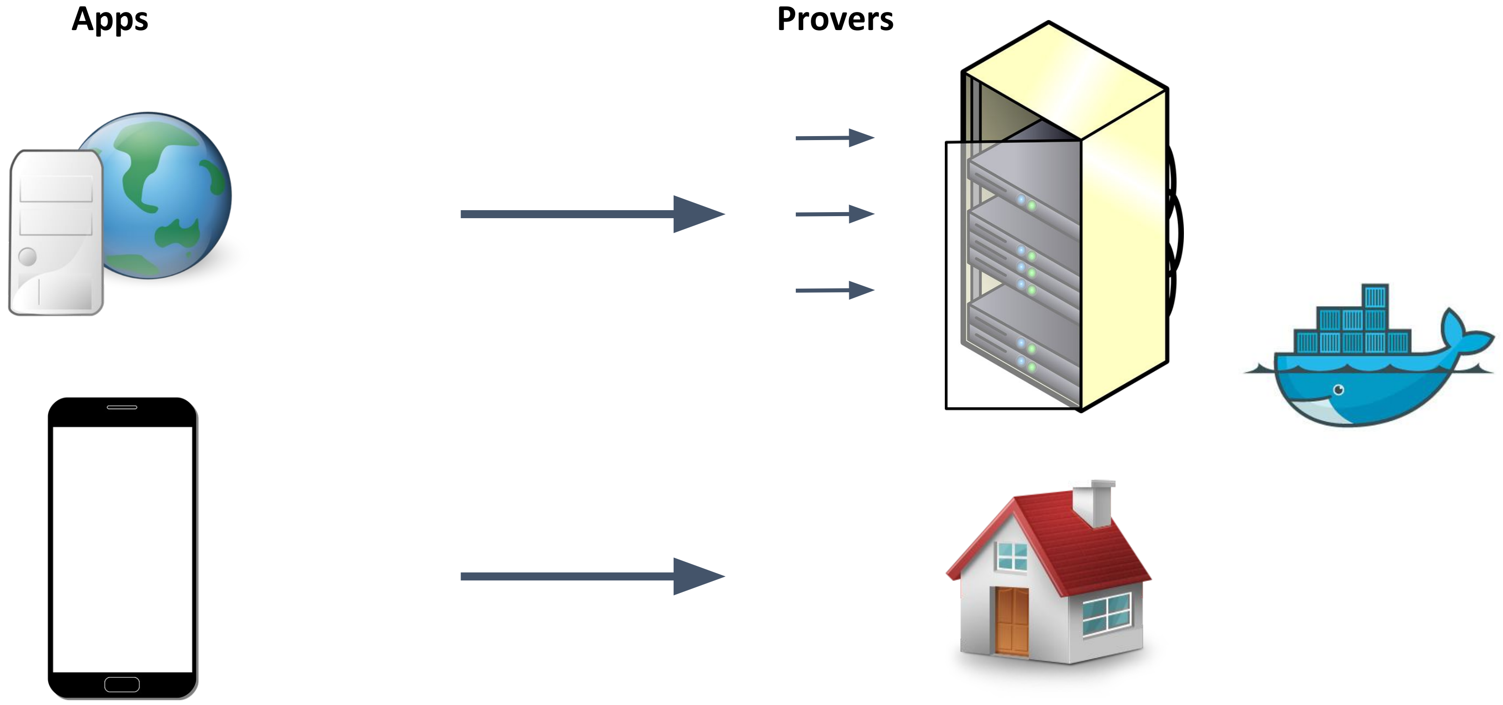
Prover's View

A private witness is used to generate a proof.

Verifier's View

The proof is checked based on public inputs.

Use-case: Proof-as-a-Service



Use-case: Tests & Benchmarks

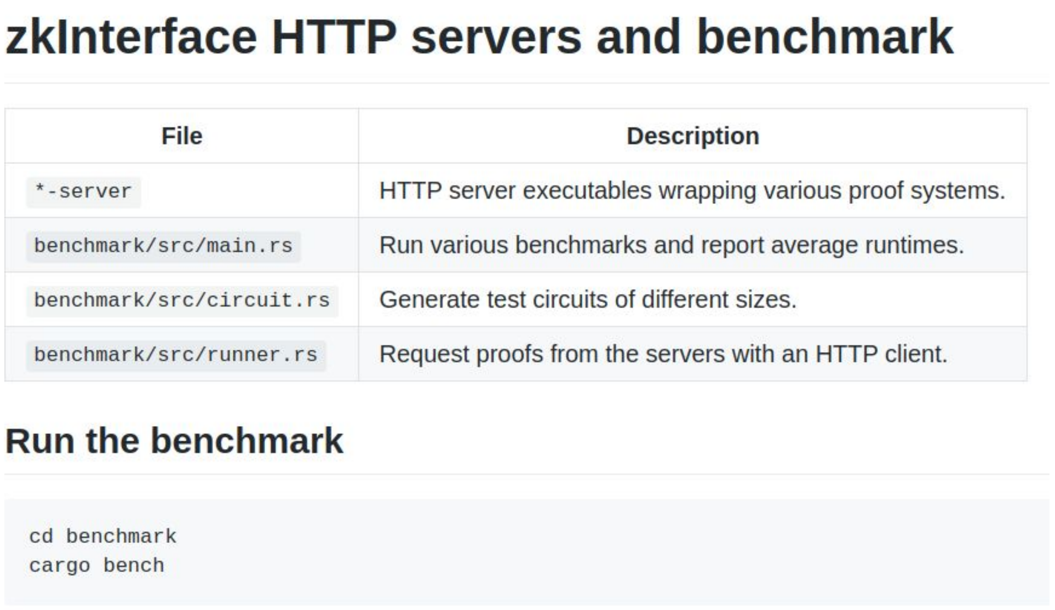
Compare provers,

circuit shapes,

hardware.

Code

github.com/QED-it/zkinterface-http



zkInterface HTTP servers and benchmark

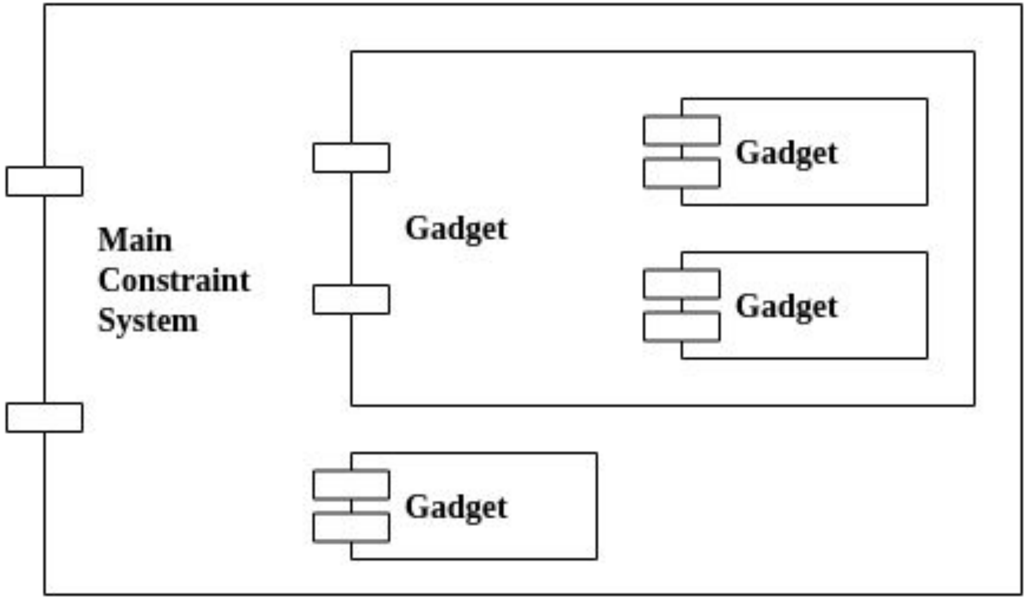
File	Description
`*-server`	HTTP server executables wrapping various proof systems.
`benchmark/src/main.rs`	Run various benchmarks and report average runtimes.
`benchmark/src/circuit.rs`	Generate test circuits of different sizes.
`benchmark/src/runner.rs`	Request proofs from the servers with an HTTP client.

Run the benchmark

```
cd benchmark
cargo bench
```

Use-case: Development Infrastructure

```
# install gadget_library
```



Status

Working Prototypes

Bellman Groth16, Dalek Bulletproofs.

ZoKrates, libSNARK, Mir.

Wishlist (in progress)

Circom / Websnark, ZEXE, MARLIN,

Setup MPC, …

Check our homepage

github.com/QED-it/zkinterface

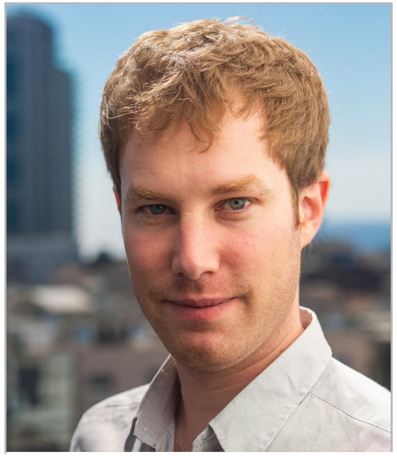
Propose

community.zkproof.org

Chat

Telegram zkInterface

Thank you!



Aurel Nicolas
QEDIT
aurel@qed-it.com

Daniel Benarroch
QEDIT
daniel@qed-it.com

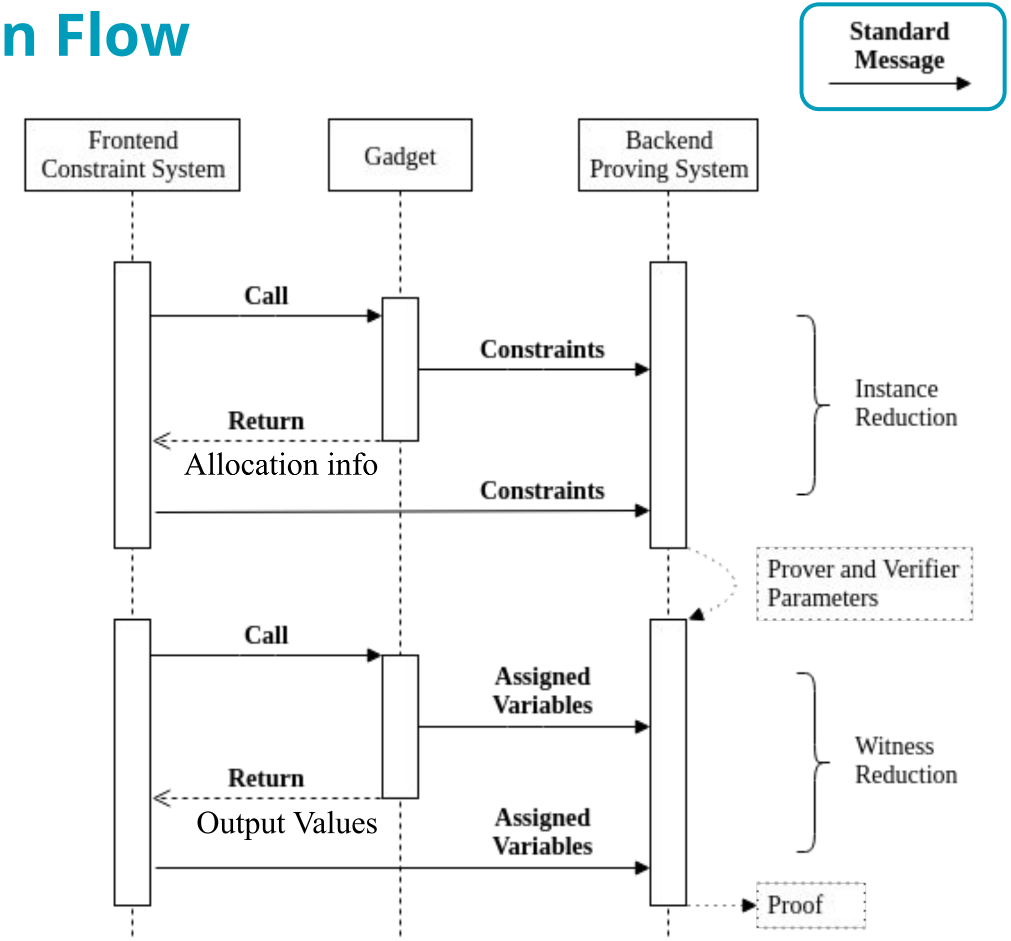
Eran Tromer
Columbia, TAU
eran@tromer.org

Ron Kahat
QEDIT
ron@qed-it.com

Special thanks to K. Gurkan, S. Deml, T. Schaeffer,
J. Eberhart, D. Lubarov, J. Baylina, O. Andreev, P. Mishra

Insert Slide Header

Composition Flow



The Proposal & Demo

- Specification: github.com/QED-it/zkinterface
- Messages definition
- Demo:
  - ZoKrates front-end
  - Bellman back-end

Using the standard

In frontends

- Support external gadgets
- Expose gadgets as a library
- Export R1CS/Witness

In backends

- Import R1CS/Witness

How to

- Use generated code (Rust, C++)
- Execute as Rust library, or C FFI, or processes & files

Future Work

Roadmap

- More Frontends (snarky, bellman)
- More Backends (bulletproofs, libsnark)

Extensions

- Executable packaging
- A type system for variables
- Other constraint systems (uniformity, boolean circuits, …)

Terminology

- **Frontend** = express constraints in a readable language
- **Backend** = cryptographic scheme to prove and verify
- **R1CS** = Rank 1 Constraint System
- **Gadget** = reusable fragment of R1CS
- **Instance** = the statement claimed (with respect to a fixed relation)
- **Witness** = secret evidence of the statement’s truth

Constraint system interoperability: goals

- Instance and witness formats
- Semantics, variable representation and mapping
- Witness reduction
- Gadgets interoperability
- Procedural interoperability

Desiderata

- Interoperability across frontend frameworks and programming languages.
- Ability to write gadgets that can be consumed by different frontends and backends.
- Minimize copying and duplication of data.
- The overhead of the R1CS construction and witness reduction should be low (and in particular, linear) compared to a native implementation of the same gadgets in existing frameworks.
- Expose details of the backend's interface that are necessary for performance (e.g., constraint system representation and algebraic fields).
- The approach can be extended to support constraint systems beyond R1CS.

Scope and limitations

In scope

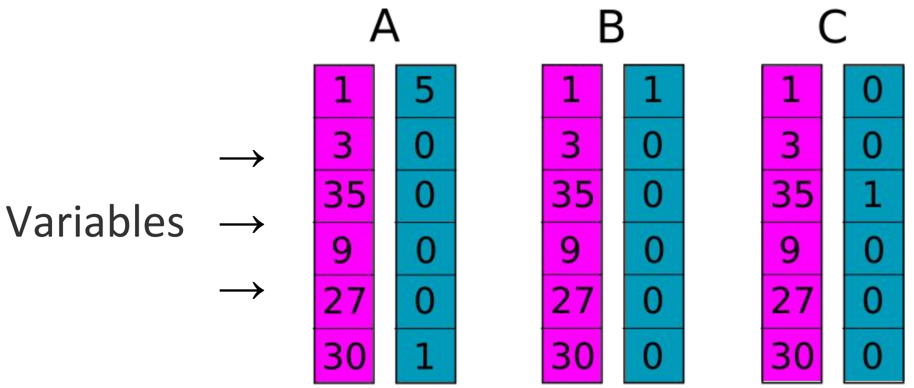
- Messages that the caller and callee exchange.
- Serialization of the messages.
- A protocol to build a constraint system from gadget composition.
- Technical recommendations for implementation.

Out of scope

- Backend interoperability
- Programming language and frontend frameworks
- Beyond R1CS
  - Other styles
  - Uniformity
- Packaging
- Typing

R1CS

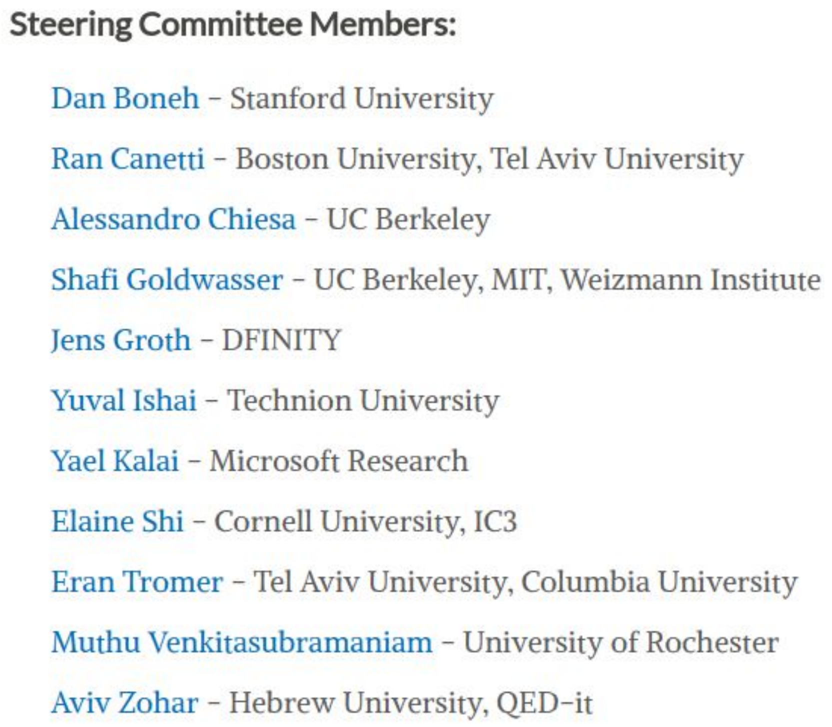
- Frontend → R1CS zkInterface → Backend
  a. Constraints (instance reduction) → Prover/Verifier Keys
  b. Witness (witness reduction) → Proof



35 * 1 - 35 = 0

ZKProof.org

- ZKProof is an open initiative to standardize zero knowledge proofs
  and bridge academia and industry
- 1st Standards Workshop generated 3 documents as guidelines
- 7 proposals and 30 talks and discussions
- Elliptic curve generation, commit&prove, interoperability.
- Discuss on community.zkproof.org



Steering Committee Members:

Dan Boneh – Stanford University
Ran Canetti – Boston University, Tel Aviv University
Alessandro Chiesa – UC Berkeley
Shafi Goldwasser – UC Berkeley, MIT, Weizmann Institute
Jens Groth – DFINITY
Yuval Ishai – Technion University
Yael Kalai – Microsoft Research
Elaine Shi – Cornell University, IC3
Eran Tromer – Tel Aviv University, Columbia University
Muthu Venkitasubramaniam – University of Rochester
Aviv Zohar – Hebrew University, QED-it

2nd ZKProof Workshop, April 10-12, Berkeley

PLATINUM SPONSORS

QEDit

Zcash Foundation

GOLD SPONSORS

BEAM

vmware®

SILVER SPONSORS

CPIIS CHECK POINT INSTITUTE FOR INFORMATION SECURITY

clearmatics

ethereum foundation

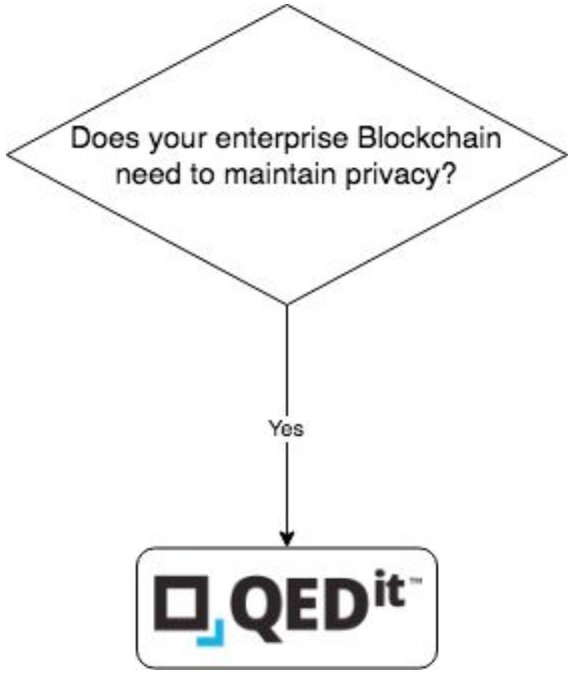
Protocol Labs

STRATUMN

SPECIAL CONTRIBUTORS

Microsoft

QEDIT



ZK for finance, regulation, supply chains.