ZK-SecreC

Dan Bogdanov
Hendrik Eerikson
Peeter Laud
Markko Merzin
Härmel Nestra
Martin Pettai
Jaak Randmets
Raul-Martin Rebane
Kert Tali
Sandhra-Mirella Valdma

This research has been funded by the Defense Advanced Research Projects Agency (DARPA) under contract HR0011-20-C-0083. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This research has also been supported by European Regional Development Fund through the Estonian Centre of Excellence in ICT Research (EXITE).

CYBERNETICA

Why a DSL?

- Creating proofs from real-world statements must be easy for adoption of ZK
- Challenges for normal languages
  - Interleaving on- and off-circuit computation
  - Witness expansion side-channel

ZK-SecreC

- Rust-like syntax
- Real-life data sizes, formats
- Built for circuit computation
- Information flow labels
- Integrates well with backends

Information Flow

uint[N] $pre @prover

- Stage marks on/off circuit
- Domain marks privacy
- Highly polymorphic
- Branching on stage/domain

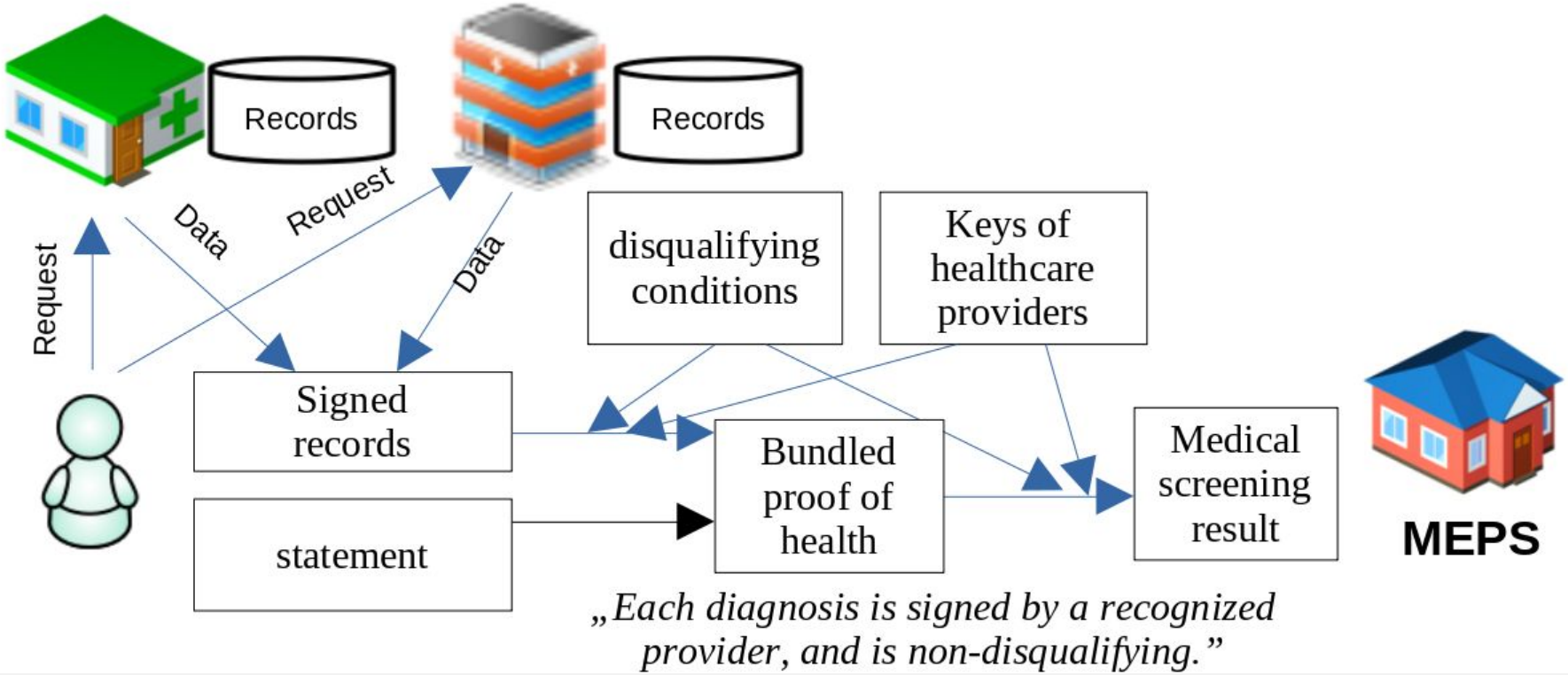
Example

```
pub fn sqrt_fixed[N : Nat, $S, @D](x : Fixed[N, $S, @D]) -> Fixed[N, $S, @D] {
 if (post $S) {
   let res = fixed_post(sqrt_fixed_pre(fixed_pre(x)));
   if (@prover <= @D) { check_coef_sqrt(x,res)}
   res
 } else {sqrt_fixed_pre(x)}
}
```

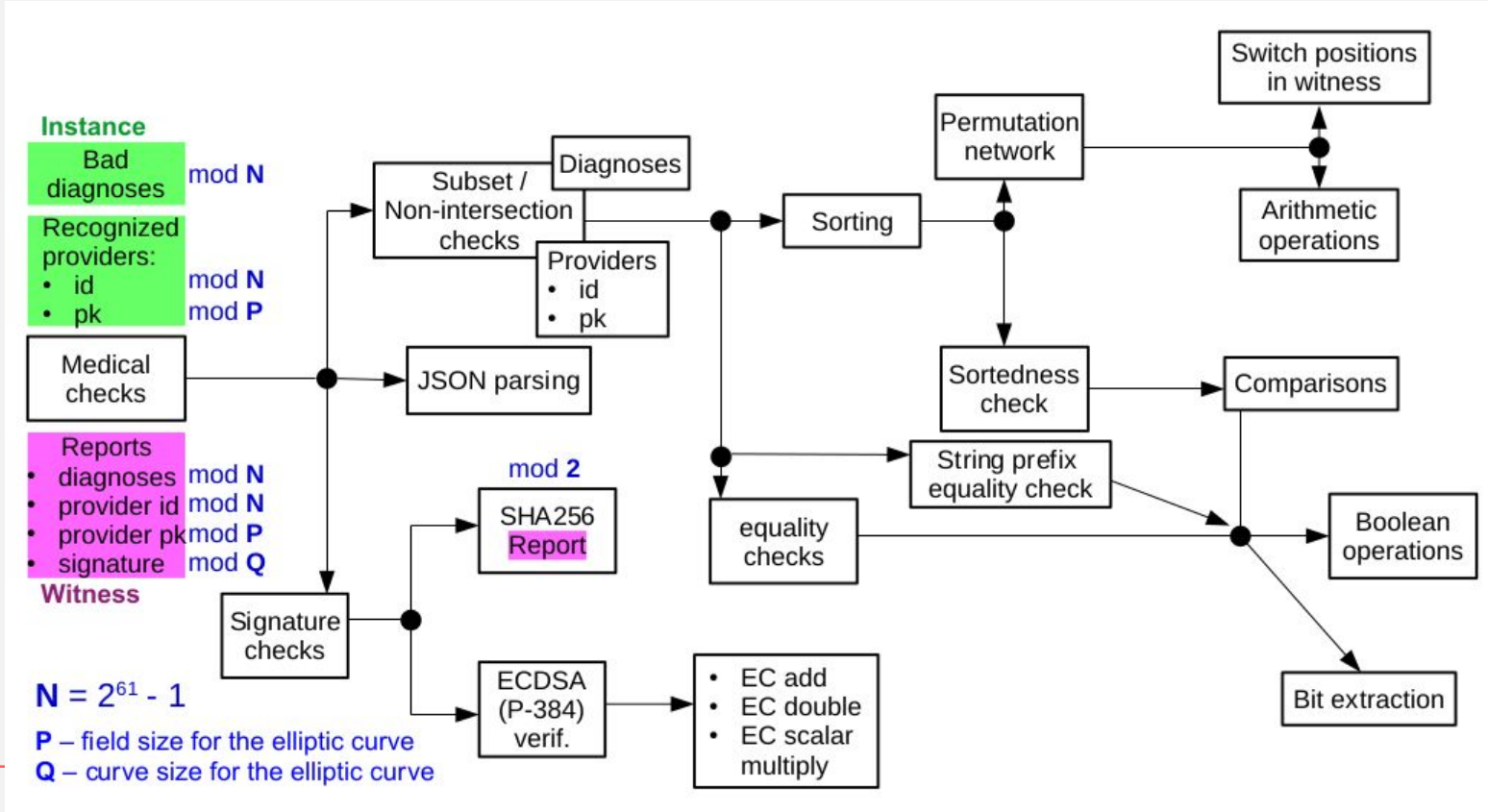
Standard Library

- We build example real-life use cases
  - Have a good idea? Tell me!
- Signing, Hashing
- Fractional Numbers
- Finite Automata, Parsing

Example: Medical Checks



Example: Medical Checks

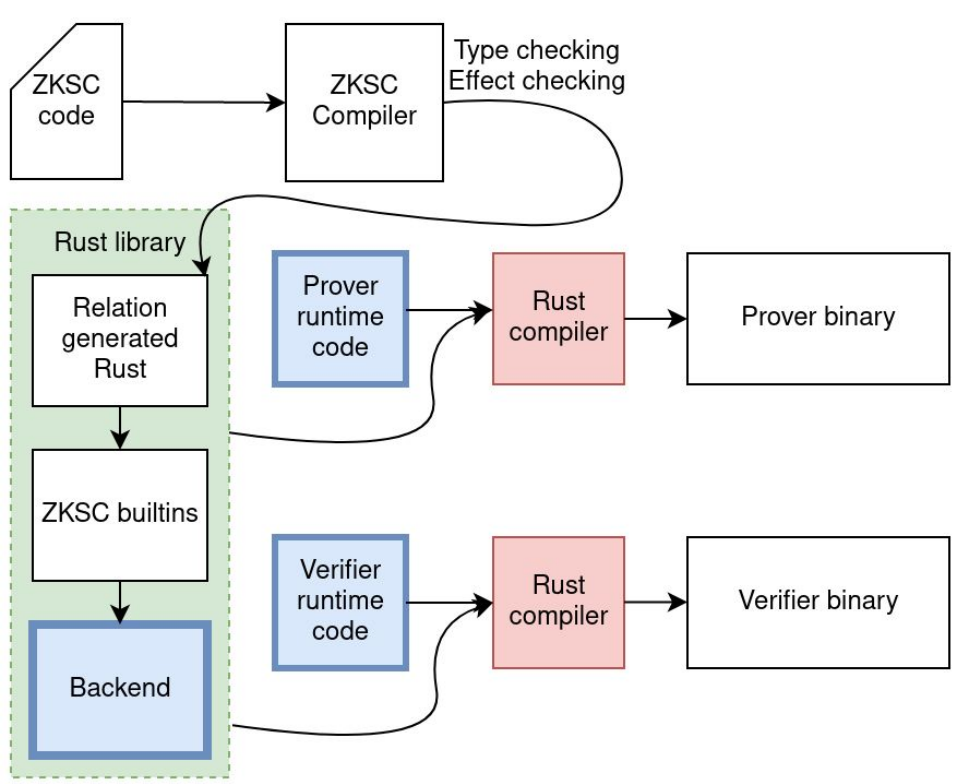


CYBERNETICA

Performance

Providers	Diagnoses per provider	Trusted Providers	Bad Diagnoses	Circuit compile time (s)
1	2	100	100	14
1	2	1000	1000	16
1	4	100	100	19
1	4	1000	1000	23

Pipeline


ZKSC code
ZKSC Compiler
Type checking
Effect checking
Rust library
Relation generated Rust
ZKSC builtins
Backend
Prover runtime code
Rust compiler
Prover binary
Verifier runtime code
Rust compiler
Verifier binary

Rust benefits

- Binary generates circuit, it is not the circuit
- Much more compact
- Circuit can scale to parameters
- Can avoid re-deployment
- WASM

Backends

- Circuit in standardized textual format
- EMP-toolkit in M61
- Diet Mac’n’Cheese integration
  - Prover in WASM, verifier in x86-64

Thank You!

Raul-Martin Rebane

cybernetica
CyberneticaAS
cybernetica_ee
Cybernetica