Proposal:
On Comparing Zero-Knowledge Proofs

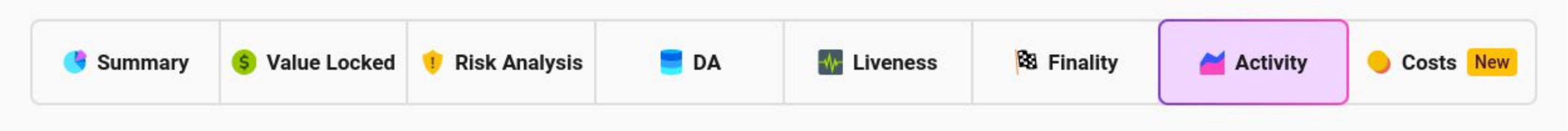
(Or: *it would be cool to have the L2BEAT of ZKPs!*)

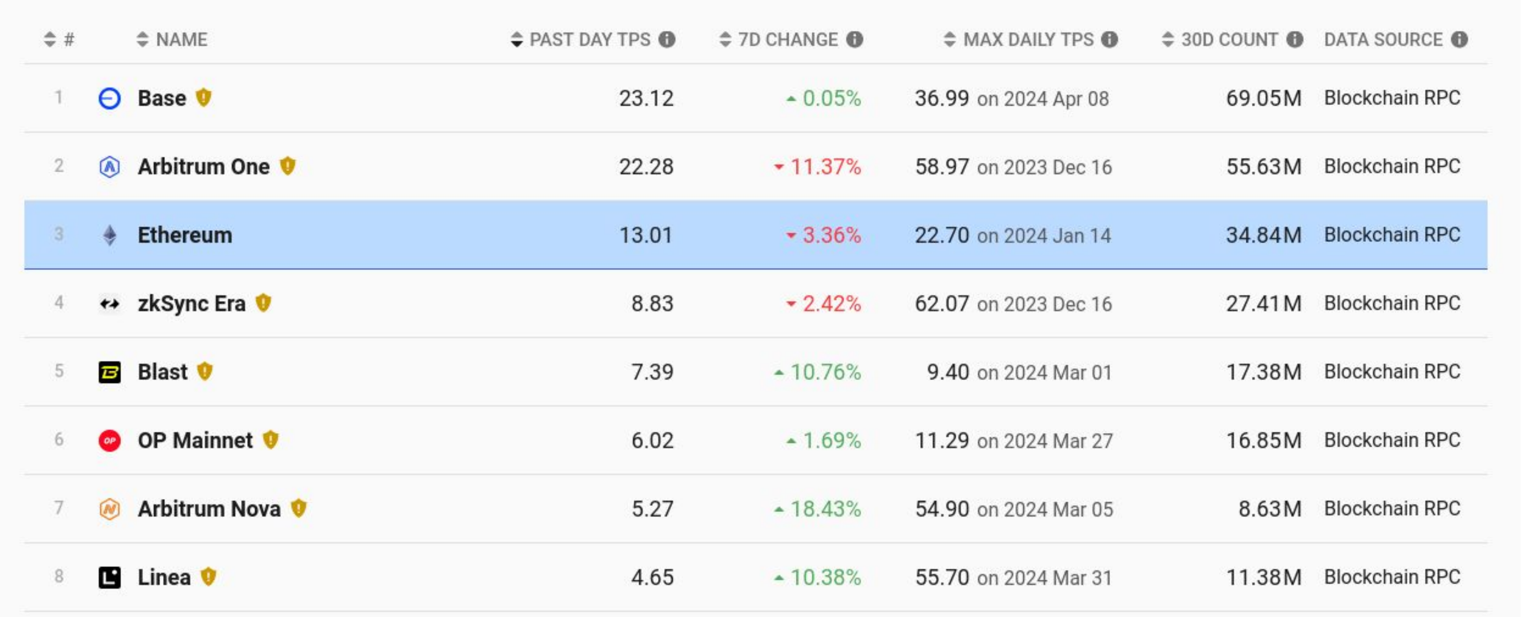
Matteo Campanelli and Marco Stronati

Matter Labs

ZKProof 2024

l2beat.com





#	NAME	PAST DAY TPS	7D CHANGE	MAX DAILY TPS	30D COUNT	DATA SOURCE
1	Base	23.12	▲ 0.05%	36.99 on 2024 Apr 08	69.05M	Blockchain RPC
2	Arbitrum One	22.28	▼ 11.37%	58.97 on 2023 Dec 16	55.63M	Blockchain RPC
3	Ethereum	13.01	▼ 3.36%	22.70 on 2024 Jan 14	34.84M	Blockchain RPC
4	zkSync Era	8.83	▼ 2.42%	62.07 on 2023 Dec 16	27.41M	Blockchain RPC
5	Blast	7.39	▲ 10.76%	9.40 on 2024 Mar 01	17.38M	Blockchain RPC
6	OP Mainnet	6.02	▲ 1.69%	11.29 on 2024 Mar 27	16.85M	Blockchain RPC
7	Arbitrum Nova	5.27	▲ 18.43%	54.90 on 2024 Mar 05	8.63M	Blockchain RPC
8	Linea	4.65	▲ 10.38%	55.70 on 2024 Mar 31	11.38M	Blockchain RPC

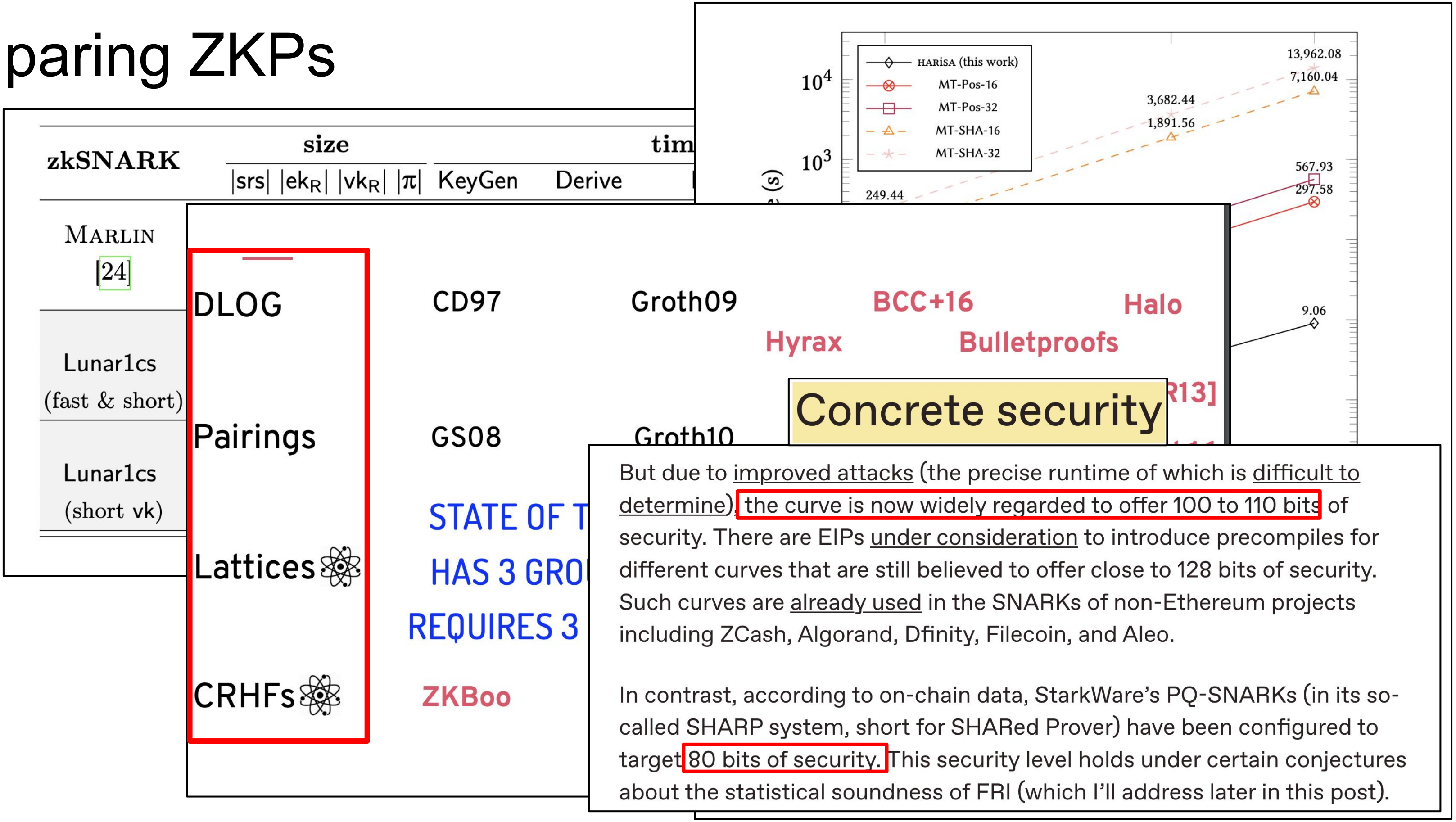
This Talk

- Why comparing ZKPs? And what do we mean by that?
- Previous Proposals and State of Affairs
  - Spoiler on the state of affairs: *meh*.
- Let’s Talk About *What Next*

Foreword

- This is an “opinion piece”
- Educated guesses on the state of affairs
- The ideas in here are just proposals
  - We want to hear your opinion and your better proposals

Comparing ZKPs



But due to improved attacks (the precise runtime of which is difficult to determine), the curve is now widely regarded to offer 100 to 110 bits of security. There are EIPs under consideration to introduce precompiles for different curves that are still believed to offer close to 128 bits of security. Such curves are already used in the SNARKs of non-Ethereum projects including ZCash, Algorand, Dfinity, Filecoin, and Aleo.

In contrast, according to on-chain data, StarkWare's PQ-SNARKs (in its so-called SHARP system, short for SHARed Prover) have been configured to target 80 bits of security. This security level holds under certain conjectures about the statistical soundness of FRI (which I'll address later in this post).

Sources: Lunar (Asiacrypt 21); HARiSA (CCS 22); Keynote by Sarah Meiklejohn @ Eurocrypt 2021; Justin Thaler's "SNARK Security and Performance"

Comparing ZKP ≈

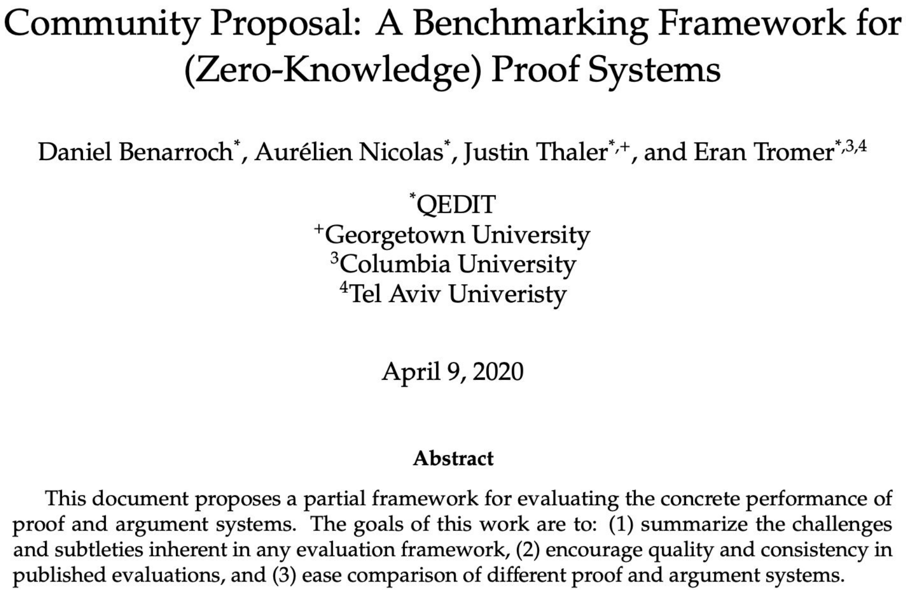
Blueprint (e.g. formal description)
Artifact

Efficiency
Security/Assumptions

Comparing ZKP Systems at a glance: why and for whom?

- **For researchers**
  - To *compare* their designs to existing ones
  - To *critically assess* claims of published and submitted works
- **For practitioners**
  - Who may be seeking out existing design/systems *to adopt*
- **For consumers of ZKP applications**
  - Who want to *compare providers* by performance, security, potential, etc.
  - (NB: major impact in rapidly expanding areas such as L2s based on validity proofs)

This is a hard problem, and not a new one ([BNTT20])



Community Proposal: A Benchmarking Framework for (Zero-Knowledge) Proof Systems

Daniel Benarroch[*], Aurélien Nicolas[*], Justin Thaler[*,+], and Eran Tromer[*,3,4]

[*]QEDIT
[+]Georgetown University
[3]Columbia University
[4]Tel Aviv Univeristy

April 9, 2020

Abstract

This document proposes a partial framework for evaluating the concrete performance of proof and argument systems. The goals of this work are to: (1) summarize the challenges and subtleties inherent in any evaluation framework, (2) encourage quality and consistency in published evaluations, and (3) ease comparison of different proof and argument systems.

The proposal in this talk is mostly orthogonal to [BNTT20]

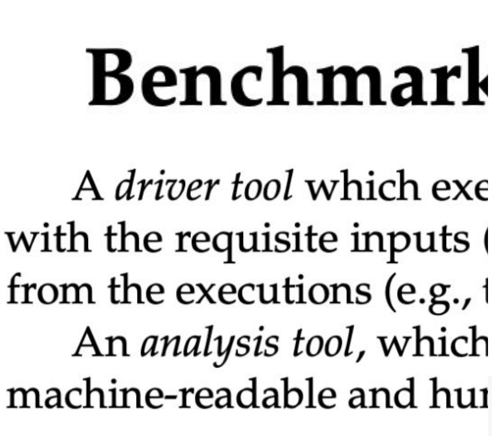
Image sources:
[BNTT20] and [BGKNT20] slides at 3rd ZKProof Workshop

Community Proposal: A Benchmarking Framework for (Zero-Knowledge) Proof Systems

Main Challenges

Ide
func

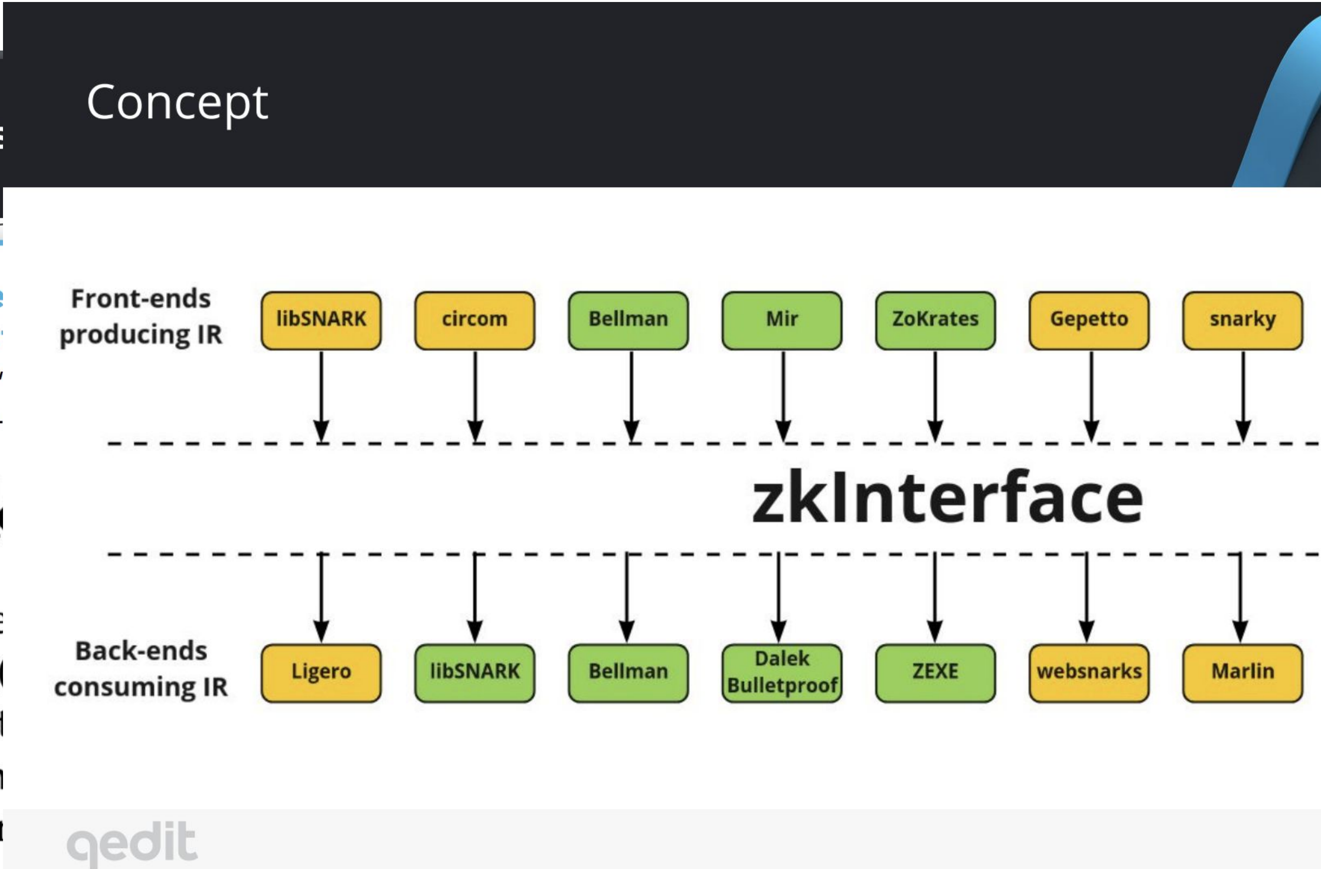
Identify "
functionalit



Benchmark

A *driver tool* which exe
with the requisite inputs (
from the executions (e.g., t

An *analysis tool*, which
machine-readable and hu



Concept

Front-ends producing IR

libSNARK, circom, Bellman, Mir, ZoKrates, Gepetto, snarky

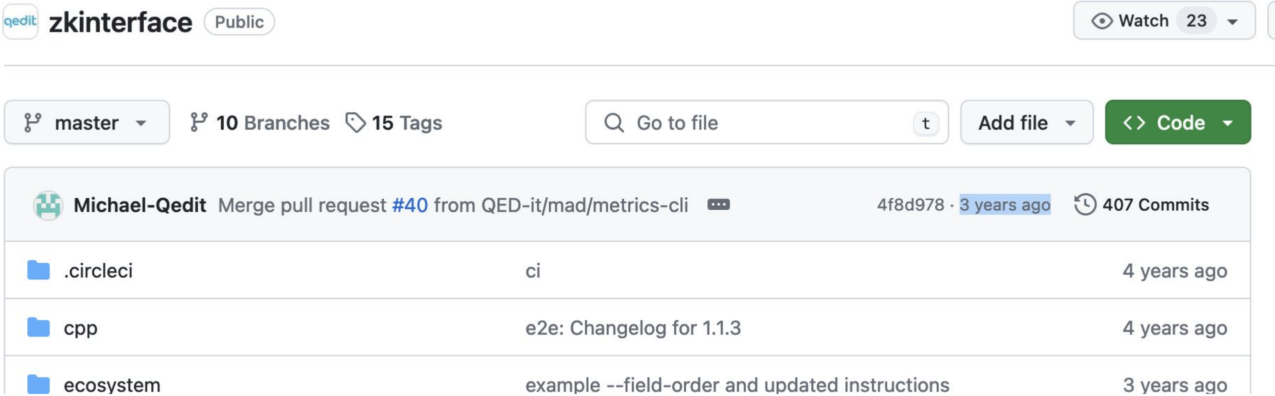
zkInterface

Back-ends consuming IR

Ligero, libSNARK, Bellman, Dalek Bulletproof, ZEXE, websnarks, Marlin

And then what happened??

Unfortunately, not much.



qedit zkinterface Public

Watch 23

master · 10 Branches · 15 Tags · Go to file t · Add file · Code

Michael-Qedit Merge pull request #40 from QED-it/mad/metrics-cli 4f8d978 · 3 years ago 407 Commits

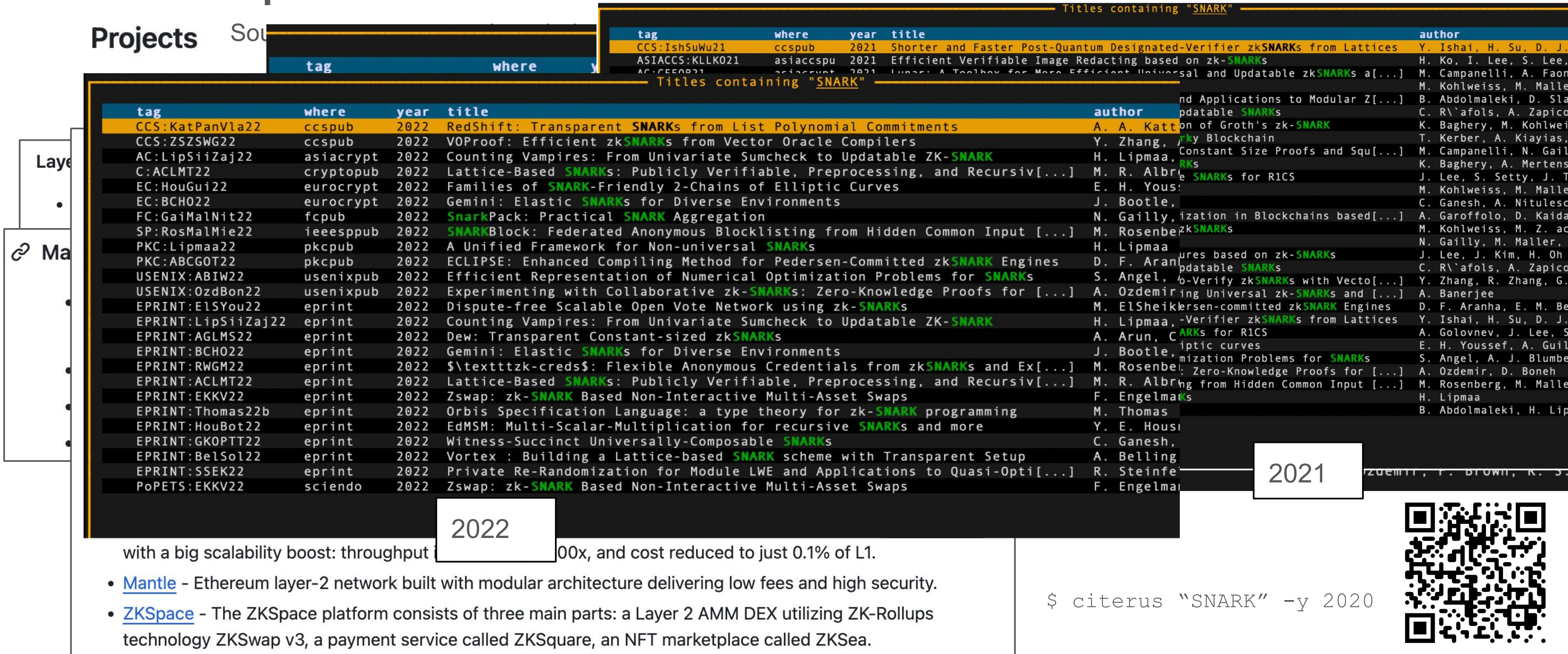
.circleci	ci	4 years ago
cpp	e2e: Changelog for 1.1.3	4 years ago
ecosystem	example --field-order and updated instructions	3 years ago

…which brings us to this talk.

And today this issue might be even more pressing

More adoptions

More constructions



Projects So

Laye

Ma

```
                                         Titles containing "SNARK"
tag                    where        year  title                                                                  author
CCS:KatPanVla22        ccspub       2022  RedShift: Transparent SNARKs from List Polynomial Commitments          A. A. Katt
CCS:ZSZSWG22           ccspub       2022  VOProof: Efficient zkSNARKs from Vector Oracle Compilers               Y. Zhang,
AC:LipSiiZaj22         asiacrypt    2022  Counting Vampires: From Univariate Sumcheck to Updatable ZK-SNARK     H. Lipmaa,
C:ACLMT22              cryptopub    2022  Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursiv[...]  M. R. Albr
EC:HouGui22            eurocrypt    2022  Families of SNARK-Friendly 2-Chains of Elliptic Curves                 E. H. Yous
EC:BCHO22              eurocrypt    2022  Gemini: Elastic SNARKs for Diverse Environments                        J. Bootle,
FC:GaiMalNit22         fcpub        2022  SnarkPack: Practical SNARK Aggregation                                 N. Gailly,
SP:RosMalMie22         ieeesppub    2022  SNARKBlock: Federated Anonymous Blocklisting from Hidden Common Input [...]  M. Rosenbe
PKC:Lipmaa22           pkcpub       2022  A Unified Framework for Non-universal SNARKs                           H. Lipmaa
PKC:ABCGOT22           pkcpub       2022  ECLIPSE: Enhanced Compiling Method for Pedersen-Committed zkSNARK Engines  D. F. Aran
USENIX:ABIW22          usenixpub    2022  Efficient Representation of Numerical Optimization Problems for SNARKs  S. Angel,
USENIX:OzdBon22        usenixpub    2022  Experimenting with Collaborative zk-SNARKs: Zero-Knowledge Proofs for [...]  A. Ozdemir
EPRINT:ElSYou22        eprint       2022  Dispute-free Scalable Open Vote Network using zk-SNARKs                M. ElSheik
EPRINT:LipSiiZaj22     eprint       2022  Counting Vampires: From Univariate Sumcheck to Updatable ZK-SNARK     H. Lipmaa,
EPRINT:AGLMS22         eprint       2022  Dew: Transparent Constant-sized zkSNARKs                               A. Arun, C
EPRINT:BCHO22          eprint       2022  Gemini: Elastic SNARKs for Diverse Environments                        J. Bootle,
EPRINT:RWGM22          eprint       2022  $\texttt zk-creds$: Flexible Anonymous Credentials from zkSNARKs and Ex[...]  M. Rosenbe
EPRINT:ACLMT22         eprint       2022  Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursiv[...]  M. R. Albr
EPRINT:EKKV22          eprint       2022  Zswap: zk-SNARK Based Non-Interactive Multi-Asset Swaps                F. Engelma
EPRINT:Thomas22b       eprint       2022  Orbis Specification Language: a type theory for zk-SNARK programming  M. Thomas
EPRINT:HouBot22        eprint       2022  EdMSM: Multi-Scalar-Multiplication for recursive SNARKs and more      Y. E. Hous
EPRINT:GKOPTT22        eprint       2022  Witness-Succinct Universally-Composable SNARKs                         C. Ganesh,
EPRINT:BelSol22        eprint       2022  Vortex : Building a Lattice-based SNARK scheme with Transparent Setup  A. Belling
EPRINT:SSEK22          eprint       2022  Private Re-Randomization for Module LWE and Applications to Quasi-Opti[...]  R. Steinfe
PoPETS:EKKV22          sciendo      2022  Zswap: zk-SNARK Based Non-Interactive Multi-Asset Swaps                F. Engelma
```

2022

```
                                         Titles containing "SNARK"
tag                    where        year  title                                                                  author
CCS:IshSuWu21          ccspub       2021  Shorter and Faster Post-Quantum Designated-Verifier zkSNARKs from Lattices  Y. Ishai, H. Su, D. J.
ASIACCS:KLLKO21        asiaccspu    2021  Efficient Verifiable Image Redacting based on zk-SNARKs               H. Ko, I. Lee, S. Lee,
                                          sal and Updatable zkSNARKs a[...]                                      M. Campanelli, A. Faon
                                                                                                                 M. Kohlweiss, M. Malle
                                          nd Applications to Modular Z[...]                                      B. Abdolmaleki, D. Sla
                                          pdatable SNARKs                                                        C. R\`afols, A. Zapico
                                          on of Groth's zk-SNARK                                                 K. Baghery, M. Kohlwei
                                          rky Blockchain                                                         T. Kerber, A. Kiayias,
                                          Constant Size Proofs and Squ[...]                                      M. Campanelli, N. Gail
                                          RKs                                                                    K. Baghery, A. Mertens
                                          e SNARKs for R1CS                                                      J. Lee, S. Setty, J. T
                                                                                                                 M. Kohlweiss, M. Malle
                                                                                                                 C. Ganesh, A. Nitulesc
                                          ization in Blockchains based[...]                                      A. Garoffolo, D. Kaida
                                          zkSNARKs                                                               M. Kohlweiss, M. Z. ac
                                                                                                                 N. Gailly, M. Maller,
                                          ures based on zk-SNARKs                                                J. Lee, J. Kim, H. Oh
                                          pdatable SNARKs                                                        C. R\`afols, A. Zapico
                                          o-Verify zkSNARKs with Vecto[...]                                      Y. Zhang, R. Zhang, G.
                                          ing Universal zk-SNARKs and [...]                                      A. Banerjee
                                          ersen-committed zkSNARK Engines                                        D. F. Aranha, E. M. Be
                                          -Verifier zkSNARKs from Lattices                                       Y. Ishai, H. Su, D. J.
                                          ARKs for R1CS                                                          A. Golovnev, J. Lee, S
                                          iptic curves                                                           E. H. Youssef, A. Guil
                                          mization Problems for SNARKs                                           S. Angel, A. J. Blumbe
                                          : Zero-Knowledge Proofs for [...]                                      A. Ozdemir, D. Boneh
                                          ng from Hidden Common Input [...]                                      M. Rosenberg, M. Malle
                                          Ks                                                                     H. Lipmaa
                                                                                                                 B. Abdolmaleki, H. Lip
```

2021

with a big scalability boost: throughput i[...]00x, and cost reduced to just 0.1% of L1.

- Mantle - Ethereum layer-2 network built with modular architecture delivering low fees and high security.
- ZKSpace - The ZKSpace platform consists of three main parts: a Layer 2 AMM DEX utilizing ZK-Rollups technology ZKSwap v3, a payment service called ZKSquare, an NFT marketplace called ZKSea.

```
$ citerus "SNARK" -y 2020
```

TL;DR of this Proposal

Call to arms:
Let’s have a system/process that allows us to compare ZKPs systems at a glance;

Ideal result is unfeasible =>
Let’s at least head towards *some ideal goal* while having *something imperfect* but good enough

A basic warm-up question to get to the ideal goal

We have a vague goal:

“Let’s have a system/process that allows us to compare ZKPs systems at a glance”

We have a vague notion of comparison:

Comparing ZKP ≈

- **Blueprint** (e.g. formal description)
- **Artifact**
- **Efficiency**
- **Security/Assumptions**

Q: how does the community (at large) already describe and assess ZKPs?

The “academia–industry divide”

(what is available? How is it assessed?)

“Academic” ZKP Constructions:

- Have explicit security statements and proofs, constructions
- Are peer-reviewed by other experts
- May lack usable implementation

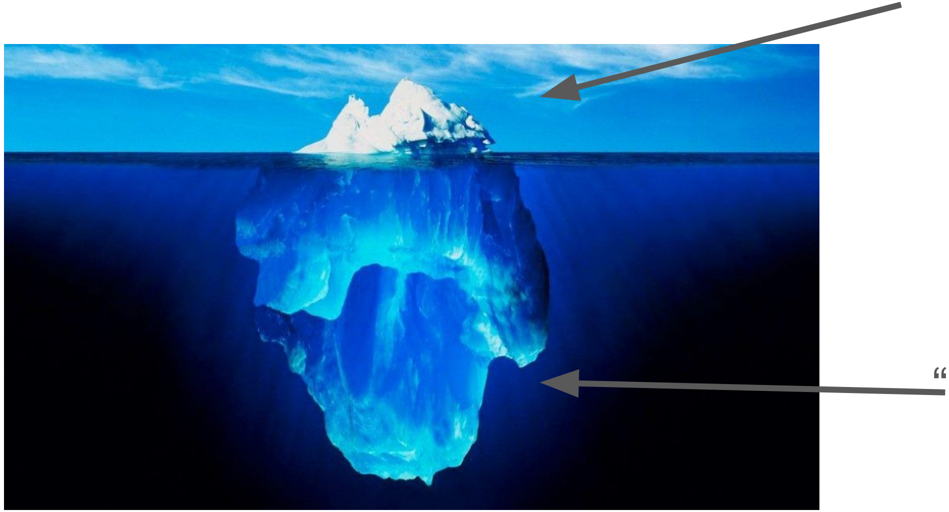
Deployed (“Industry”) ZKP Constructions:

- May be concretely used in practice
- Tend to be highly optimized
- Tend to be described informally in white papers; not peer-reviewed
- Rely on security audits with debatable effectiveness
  - Often enough, audits are not performed by experts in ZKPs

How does this inform us?

- *goal-wise:* should include bridging this gap
- *approach-wise*: meet in the middle; use strengths of two camps

The shape of the ideal goal

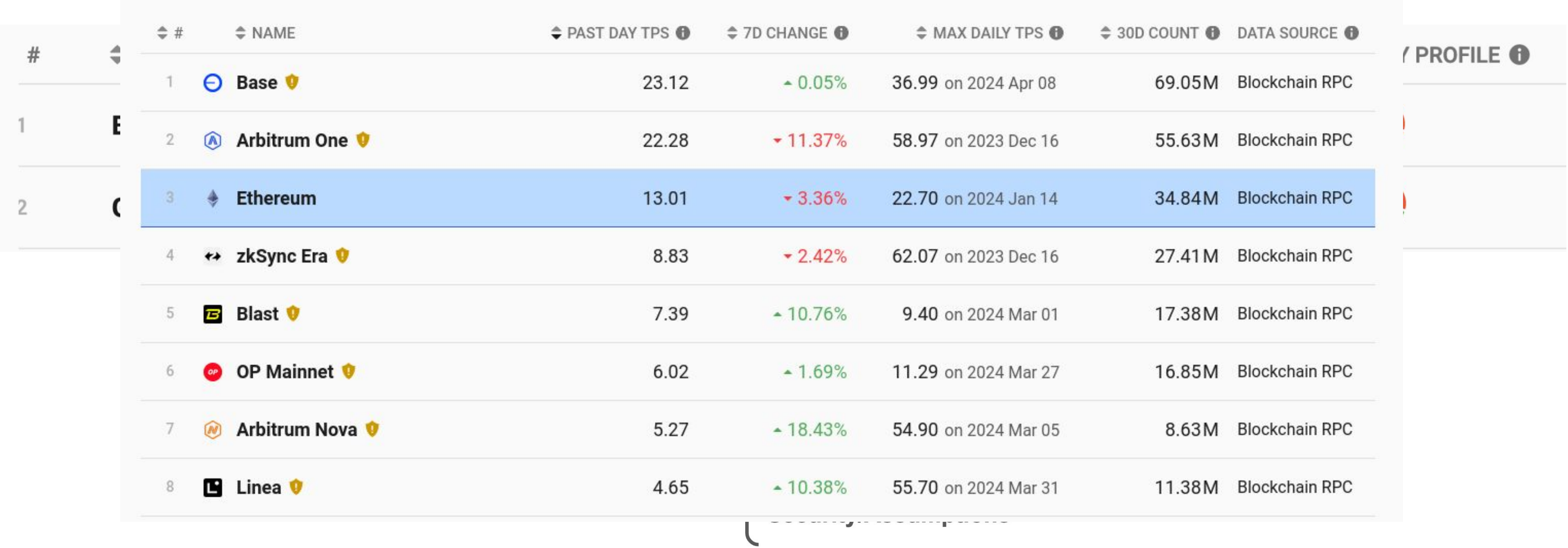


“What would this ideal goal *look like* *(on the surface)*?”

“What we need to make it work”

An “L2BEAT for ZKPs”

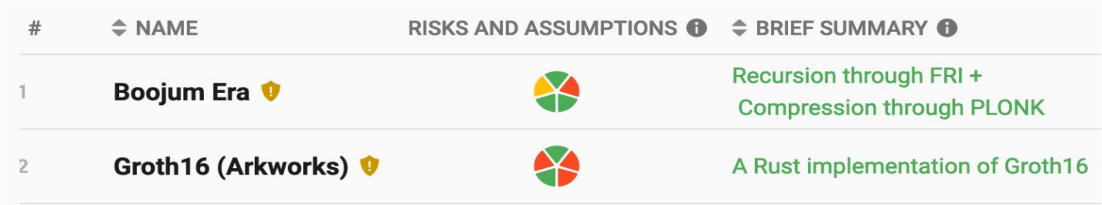
i.e., an homogeneous format for ZKP assessments + an interface for it



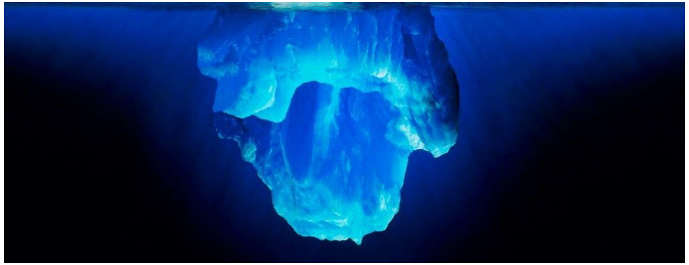
#	NAME	PAST DAY TPS	7D CHANGE	MAX DAILY TPS	30D COUNT	DATA SOURCE
1	Base	23.12	▲ 0.05%	36.99 on 2024 Apr 08	69.05M	Blockchain RPC
2	Arbitrum One	22.28	▼ 11.37%	58.97 on 2023 Dec 16	55.63M	Blockchain RPC
3	Ethereum	13.01	▼ 3.36%	22.70 on 2024 Jan 14	34.84M	Blockchain RPC
4	zkSync Era	8.83	▼ 2.42%	62.07 on 2023 Dec 16	27.41M	Blockchain RPC
5	Blast	7.39	▲ 10.76%	9.40 on 2024 Mar 01	17.38M	Blockchain RPC
6	OP Mainnet	6.02	▲ 1.69%	11.29 on 2024 Mar 27	16.85M	Blockchain RPC
7	Arbitrum Nova	5.27	▲ 18.43%	54.90 on 2024 Mar 05	8.63M	Blockchain RPC
8	Linea	4.65	▲ 10.38%	55.70 on 2024 Mar 31	11.38M	Blockchain RPC

Requirements (?)

- Should be *lean* (not a standardization process)
  - But should be an *assessment*
- Should *incentivize* participation
- Should 🎓 → ← 🏭



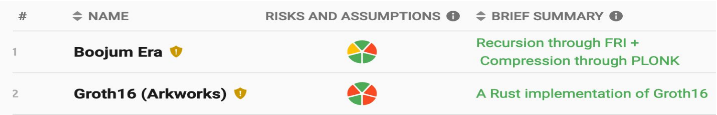
#	NAME	RISKS AND ASSUMPTIONS	BRIEF SUMMARY
1	**Boojum Era**	●	Recursion through FRI + Compression through PLONK
2	**Groth16 (Arkworks)**	●	A Rust implementation of Groth16



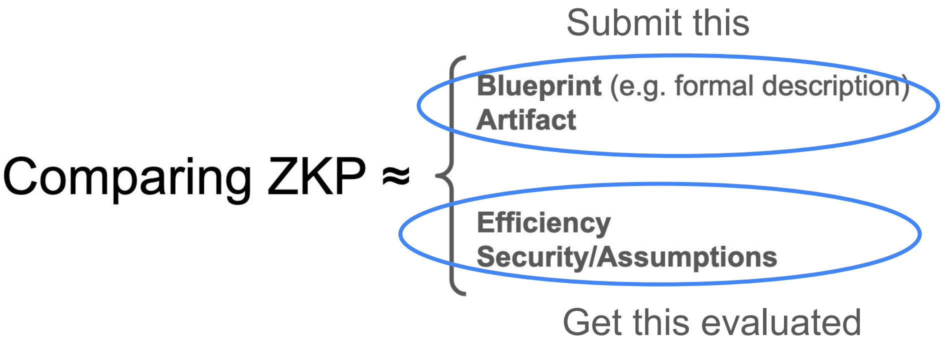
The underlying process

Proposal for a process: let’s run it like a “lean” conference

- **Submitted entries are peer-reviewed**
  - Frequent rolling deadlines
  - Submission = specs (as concrete as possible) + artifact.
- **How is this different from an academic conference?**
  - *Different requirements* (wrt: formality, related work, novelty, artifacts)
  - *Living creature*: accept re-submissions with incremental improvements; encourage reuse of code and specifications



#	NAME	RISKS AND ASSUMPTIONS	BRIEF SUMMARY
1	Boojum Era	●	Recursion through FRI + Compression through PLONK
2	Groth16 (Arkworks)	●	A Rust implementation of Groth16



Submit this

Comparing ZKP ≈

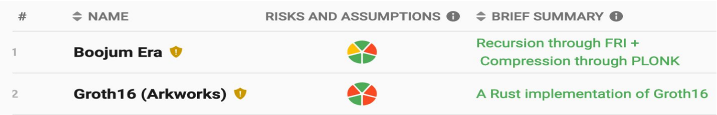
Blueprint (e.g. formal description)
Artifact

Efficiency
Security/Assumptions

Get this evaluated

Incentives: why would anyone jump in?

- **Visibility:**
  - Intuition: IF $PLATFORM_NAME is the place to be THEN people will try to there
  - possible if properly bootstrapped
  - **Further possibility for incentives:** anyone can make a submission, not necessarily authors of the original work
- **Who would review and why?**
  - Ideal reviewer profile *orbits* around “applied crypto expert”
  - Mixture of people from university and industry
    - A little bit like the ZKProof workshop



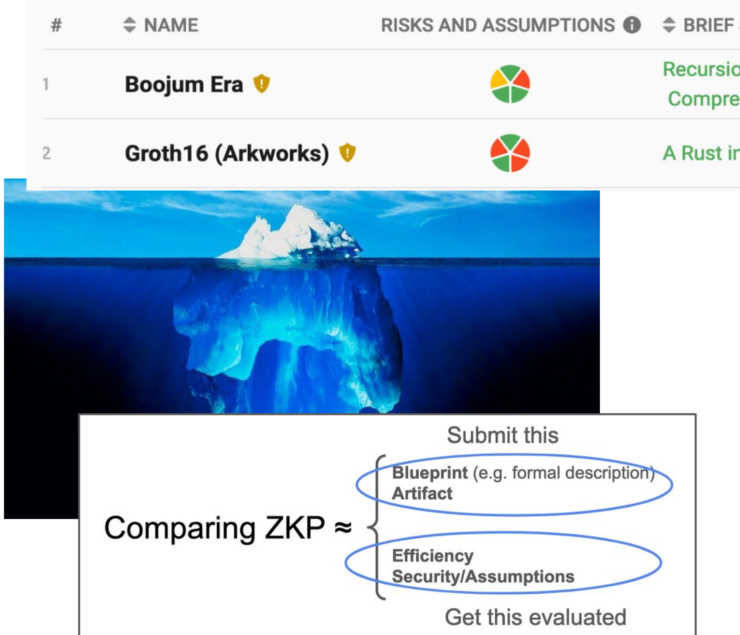
#	NAME	RISKS AND ASSUMPTIONS	BRIEF SUMMARY
1	Boojum Era	●	Recursion through FRI + Compression through PLONK
2	Groth16 (Arkworks)	●	A Rust implementation of Groth16

Expected questions (and other loose ends)

- Q: “*Doesn’t it look like a f*%$^@& nightmare to fairly compare ZKPs because…”*
    - *“... they could be efficient on setting A but not on setting B?”*
        - A: “Yes, true, but this is orthogonal to the main goal. First we’d settle on an imperfect set of comparison approaches.”
    - *“... they could be fast if parallelized but slow otherwise?”*
        - A: “Yes, true, but this is orthogonal to the main goal. First we’d settle on an imperfect set of comparison approaches.”
    - *“... they could be based on different intermediate representations and methodology?”*
        - A: “Yes, true, but this is orthogonal…”

Wrapping up

- Finding ways to compare ZKPs at a glance is important and urgent
- It could look like *this* (image), but other solutions are possible
- **Q:** If not this design, then which one?
- **Q:** If this design, then how to bootstrap it? (without being paralyzed by the unavoidable rough edges?)



#	⇕ NAME	RISKS AND ASSUMPTIONS ⓘ	⇕ BRIEF
1	**Boojum Era**	●	Recursio Compre
2	**Groth16 (Arkworks)**	●	A Rust ir

Submit this

Comparing ZKP ≈

Blueprint (e.g. formal description)
Artifact

Efficiency
Security/Assumptions

Get this evaluated

Thank you!

Contacts: {matteo,ms}@matterlabs.dev

Peer-reviewed submissions

benchmarks

+

specification

Artifact evaluation

- Benchmark on one widely available cloud instance
- Usual metrics (described in previous proposal)
- GPU only if widely available
- High-level black-box applications such as SHA256
  - Circuit implementation on top of Halo2+KZG
  - RiscV implementation on top of STARK wrapped in Groth16
- Some manual review wrt to specification

- Can we catch unsound circuits that cheat to be faster?
- What applications are relevant?

Review of specifications

Interface