Fast withdrawals for optimistic rollups

Jeremy Clark

MADIBA security research group

UNIVERSITÉ Concordia UNIVERSITY

GINA CODY SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

Funding & Partners:

NSERC CRSNG

Raymond Chabot Grant Thornton

catallaxy

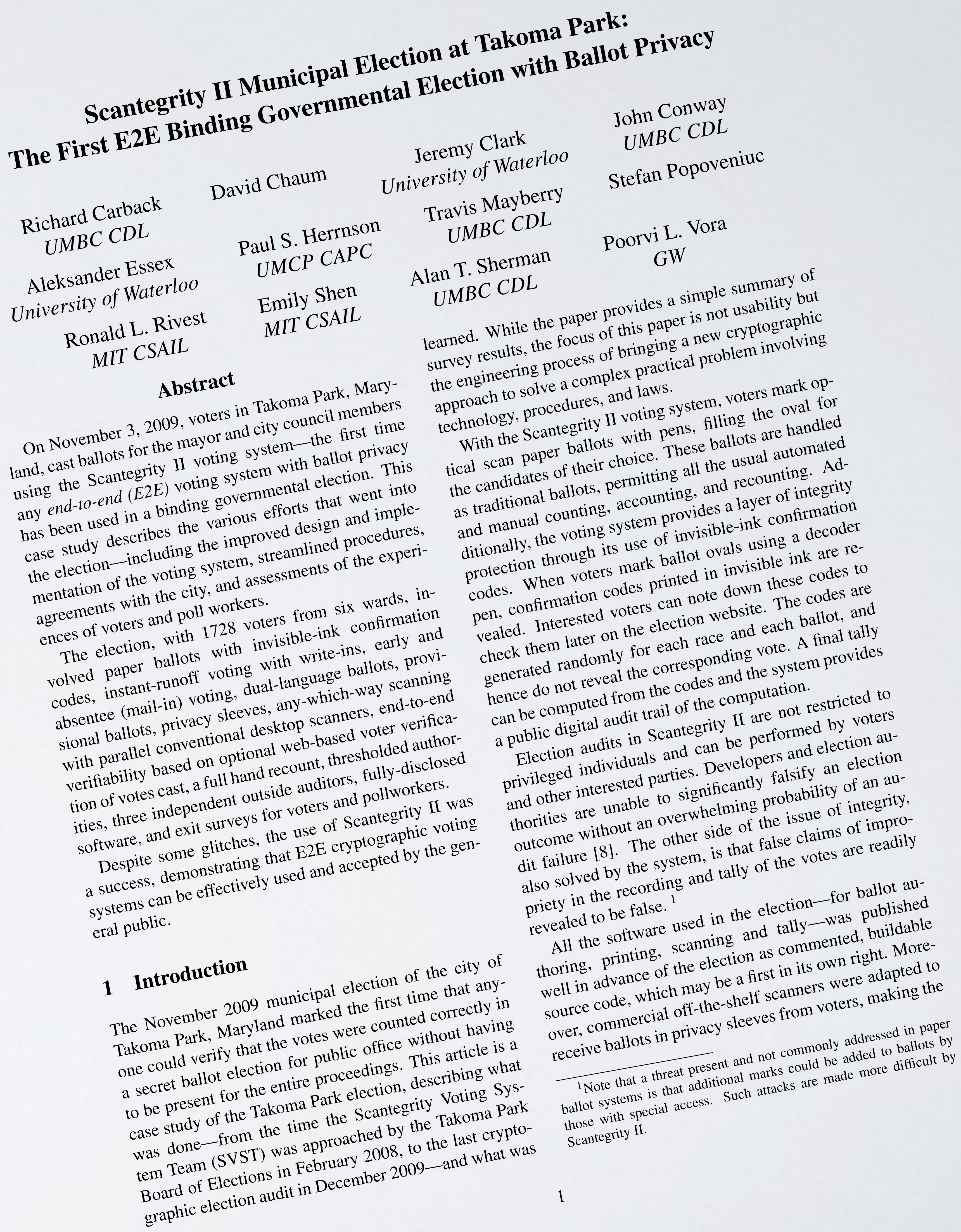
AUTORITÉ DES MARCHÉS FINANCIERS

Fonds de recherche Nature et technologies Québec

Office of the Privacy Commissioner of Canada

Cryptographic Voting

End-to-end verifiable voting
Coercion-resistance
Human-votable
Liquid democracy
Blockchain “real” voting (skeptic)
DAO voting



Scantegrity II Municipal Election at Takoma Park: The First E2E Binding Governmental Election with Ballot Privacy

Richard Carback, *UMBC CDL*
David Chaum
Jeremy Clark, *University of Waterloo*
John Conway, *UMBC CDL*
Aleksander Essex, *University of Waterloo*
Paul S. Herrnson, *UMCP CAPC*
Travis Mayberry, *UMBC CDL*
Stefan Popoveniuc
Ronald L. Rivest, *MIT CSAIL*
Emily Shen, *MIT CSAIL*
Alan T. Sherman, *UMBC CDL*
Poorvi L. Vora, *GW*

Abstract

On November 3, 2009, voters in Takoma Park, Maryland, cast ballots for the mayor and city council members using the Scantegrity II voting system—the first time any *end-to-end (E2E)* voting system with ballot privacy has been used in a binding governmental election. This case study describes the various efforts that went into the election—including the improved design and implementation of the voting system, streamlined procedures, agreements with the city, and assessments of the experiences of voters and poll workers.

The election, with 1728 voters from six wards, involved paper ballots with invisible-ink confirmation codes, instant-runoff voting with write-ins, early and absentee (mail-in) voting, dual-language ballots, provisional ballots, privacy sleeves, any-which-way scanning with parallel conventional desktop scanners, end-to-end verifiability based on optional web-based voter verification of votes cast, a full hand recount, thresholded authorities, three independent outside auditors, fully-disclosed software, and exit surveys for voters and pollworkers.

Despite some glitches, the use of Scantegrity II was a success, demonstrating that E2E cryptographic voting systems can be effectively used and accepted by the general public.

1 Introduction

The November 2009 municipal election of the city of Takoma Park, Maryland marked the first time that anyone could verify that the votes were counted correctly in a secret ballot election for public office without having to be present for the entire proceedings. This article is a case study of the Takoma Park election, describing what was done—from the time the Scantegrity Voting System Team (SVST) was approached by the Takoma Park Board of Elections in February 2008, to the last cryptographic election audit in December 2009—and what was learned. While the paper provides a simple summary of survey results, the focus of this paper is not usability but the engineering process of bringing a new cryptographic approach to solve a complex practical problem involving technology, procedures, and laws.

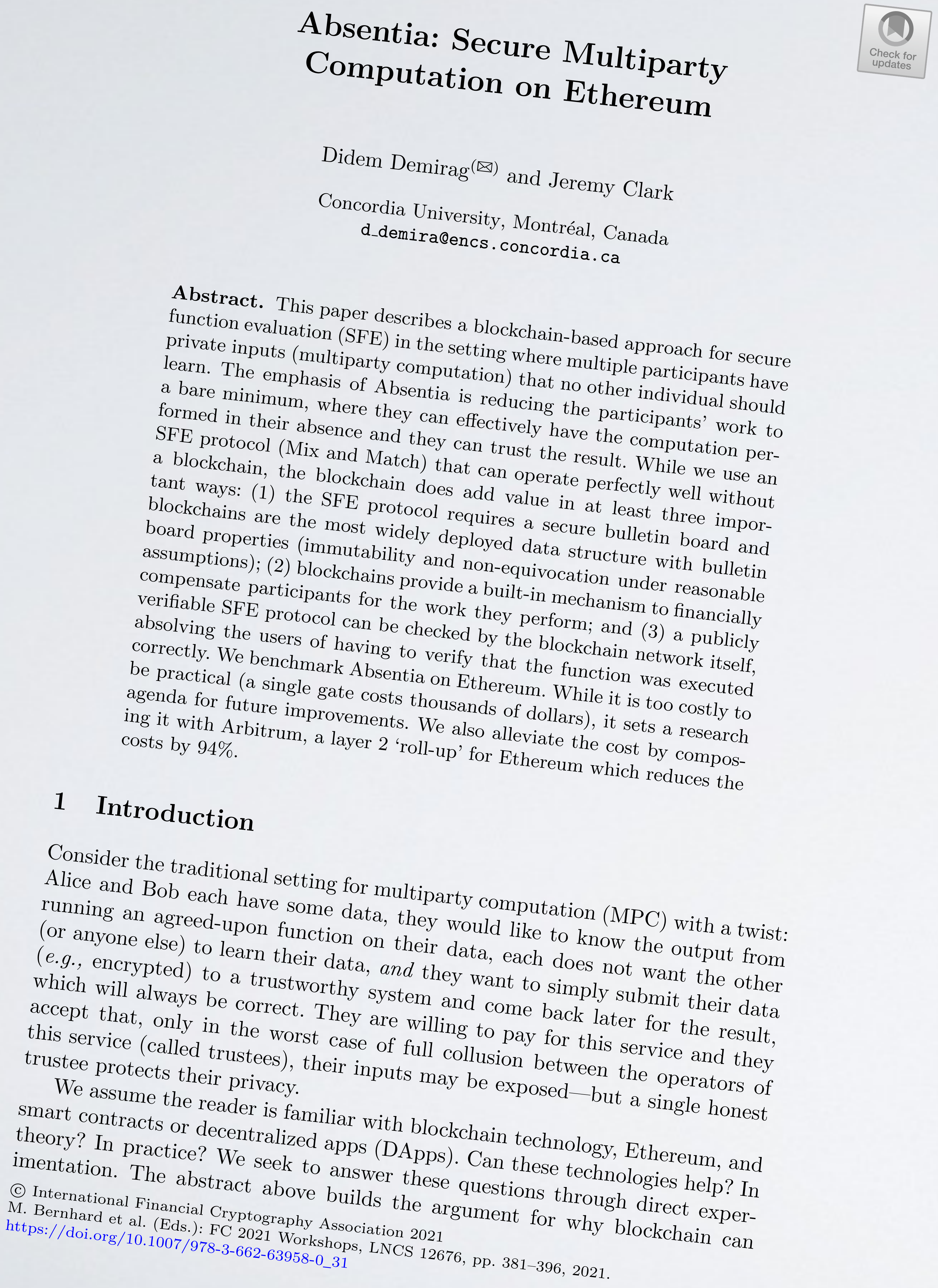
With the Scantegrity II voting system, voters mark optical scan paper ballots with pens, filling the oval for the candidates of their choice. These ballots are handled as traditional ballots, permitting all the usual automated and manual counting, accounting, and recounting. Additionally, the voting system provides a layer of integrity protection through its use of invisible-ink confirmation codes. When voters mark ballot ovals using a decoder pen, confirmation codes printed in invisible ink are revealed. Interested voters can note down these codes to check them later on the election website. The codes are generated randomly for each race and each ballot, and hence do not reveal the corresponding vote. A final tally can be computed from the codes and the system provides a public digital audit trail of the computation.

Election audits in Scantegrity II are not restricted to privileged individuals and can be performed by voters and other interested parties. Developers and election authorities are unable to significantly falsify an election outcome without an overwhelming probability of an audit failure [8]. The other side of the issue of integrity, also solved by the system, is that false claims of impropriety in the recording and tally of the votes are readily revealed to be false. [1]

All the software used in the election—for ballot authoring, printing, scanning and tally—was published well in advance of the election as commented, buildable source code, which may be a first in its own right. Moreover, commercial off-the-shelf scanners were adapted to receive ballots in privacy sleeves from voters, making the

[1] Note that a threat present and not commonly addressed in paper ballot systems is that additional marks could be added to ballots by those with special access. Such attacks are made more difficult by Scantegrity II.

Decentralized Apps (DApps)



Absentia: Secure Multiparty Computation on Ethereum

Check for updates

Didem Demirag(✉) and Jeremy Clark

Concordia University, Montréal, Canada
d_demira@encs.concordia.ca

Abstract. This paper describes a blockchain-based approach for secure function evaluation (SFE) in the setting where multiple participants have private inputs (multiparty computation) that no other individual should learn. The emphasis of Absentia is reducing the participants' work to a bare minimum, where they can effectively have the computation performed in their absence and they can trust the result. While we use an SFE protocol (Mix and Match) that can operate perfectly well without a blockchain, the blockchain does add value in at least three important ways: (1) the SFE protocol requires a secure bulletin board and blockchains are the most widely deployed data structure with bulletin board properties (immutability and non-equivocation under reasonable assumptions); (2) blockchains provide a built-in mechanism to financially compensate participants for the work they perform; and (3) a publicly verifiable SFE protocol can be checked by the blockchain network itself, absolving the users of having to verify that the function was executed correctly. We benchmark Absentia on Ethereum. While it is too costly to be practical (a single gate costs thousands of dollars), it sets a research agenda for future improvements. We also alleviate the cost by composing it with Arbitrum, a layer 2 'roll-up' for Ethereum which reduces the costs by 94%.

1 Introduction

Consider the traditional setting for multiparty computation (MPC) with a twist: Alice and Bob each have some data, they would like to know the output from running an agreed-upon function on their data, each does not want the other (or anyone else) to learn their data, *and* they want to simply submit their data (*e.g.*, encrypted) to a trustworthy system and come back later for the result, which will always be correct. They are willing to pay for this service and they accept that, only in the worst case of full collusion between the operators of this service (called trustees), their inputs may be exposed—but a single honest trustee protects their privacy.

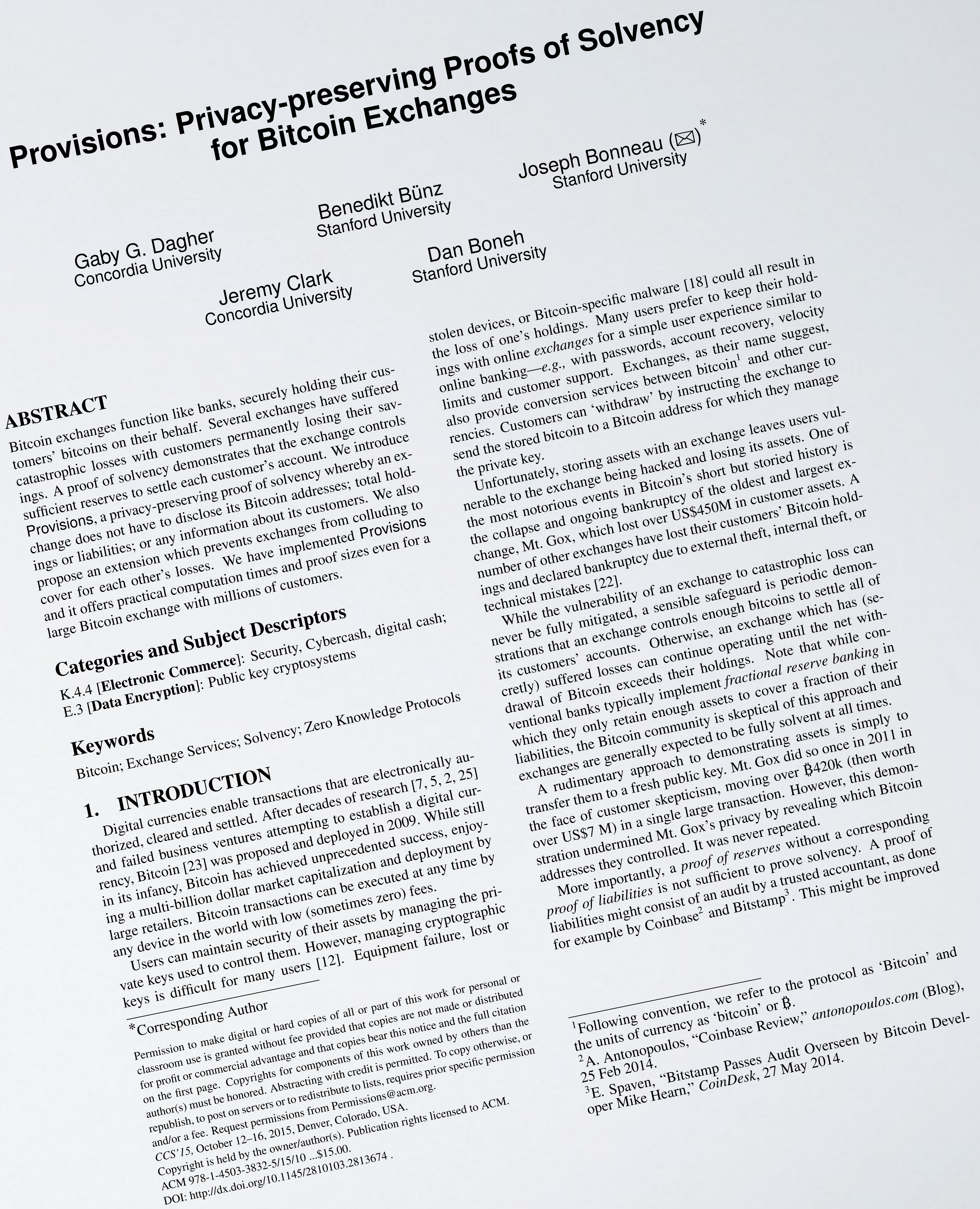
We assume the reader is familiar with blockchain technology, Ethereum, and smart contracts or decentralized apps (DApps). Can these technologies help? In theory? In practice? We seek to answer these questions through direct experimentation. The abstract above builds the argument for why blockchain can

M. Bernhard et al. (Eds.): FC 2021 Workshops, LNCS 12676, pp. 381–396, 2021.
https://doi.org/10.1007/978-3-662-63958-0_31

Multi-party computation
On-chain order books
DeFi (stablecoins, lending, derivatives)
Prediction markets
Web certificates

Blockchain protocols & attacks

Proofs of solvency
Upgradability of contracts
Multiple withdrawal attack (ERC20)
Cryptojacking



Provisions: Privacy-preserving Proofs of Solvency for Bitcoin Exchanges

Gaby G. Dagher
Concordia University

Jeremy Clark
Concordia University

Benedikt Bünz
Stanford University

Dan Boneh
Stanford University

Joseph Bonneau (✉)*
Stanford University

ABSTRACT

Bitcoin exchanges function like banks, securely holding their customers' bitcoins on their behalf. Several exchanges have suffered catastrophic losses with customers permanently losing their savings. A proof of solvency demonstrates that the exchange controls sufficient reserves to settle each customer's account. We introduce Provisions, a privacy-preserving proof of solvency whereby an exchange does not have to disclose its Bitcoin addresses; total holdings or liabilities; or any information about its customers. We also propose an extension which prevents exchanges from colluding to cover for each other's losses. We have implemented Provisions and it offers practical computation times and proof sizes even for a large Bitcoin exchange with millions of customers.

Categories and Subject Descriptors

K.4.4 [**Electronic Commerce**]: Security, Cybercash, digital cash;
E.3 [**Data Encryption**]: Public key cryptosystems

Keywords

Bitcoin; Exchange Services; Solvency; Zero Knowledge Protocols

1. INTRODUCTION

Digital currencies enable transactions that are electronically authorized, cleared and settled. After decades of research [7, 5, 2, 25] and failed business ventures attempting to establish a digital currency, Bitcoin [23] was proposed and deployed in 2009. While still in its infancy, Bitcoin has achieved unprecedented success, enjoying a multi-billion dollar market capitalization and deployment by large retailers. Bitcoin transactions can be executed at any time by any device in the world with low (sometimes zero) fees.

Users can maintain security of their assets by managing the private keys used to control them. However, managing cryptographic keys is difficult for many users [12]. Equipment failure, lost or stolen devices, or Bitcoin-specific malware [18] could all result in the loss of one's holdings. Many users prefer to keep their holdings with online *exchanges* for a simple user experience similar to online banking—*e.g.*, with passwords, account recovery, velocity limits and customer support. Exchanges, as their name suggest, also provide conversion services between bitcoin[1] and other currencies. Customers can 'withdraw' by instructing the exchange to send the stored bitcoin to a Bitcoin address for which they manage the private key.

Unfortunately, storing assets with an exchange leaves users vulnerable to the exchange being hacked and losing its assets. One of the most notorious events in Bitcoin's short but storied history is the collapse and ongoing bankruptcy of the oldest and largest exchange, Mt. Gox, which lost over US$450M in customer assets. A number of other exchanges have lost their customers' Bitcoin holdings and declared bankruptcy due to external theft, internal theft, or technical mistakes [22].

While the vulnerability of an exchange to catastrophic loss can never be fully mitigated, a sensible safeguard is periodic demonstrations that an exchange controls enough bitcoins to settle all of its customers' accounts. Otherwise, an exchange which has (secretly) suffered losses can continue operating until the net withdrawal of Bitcoin exceeds their holdings. Note that while conventional banks typically implement *fractional reserve banking* in which they only retain enough assets to cover a fraction of their liabilities, the Bitcoin community is skeptical of this approach and exchanges are generally expected to be fully solvent at all times.

A rudimentary approach to demonstrating assets is simply to transfer them to a fresh public key. Mt. Gox did so once in 2011 in the face of customer skepticism, moving over ฿420k (then worth over US$7 M) in a single large transaction. However, this demonstration undermined Mt. Gox's privacy by revealing which Bitcoin addresses they controlled. It was never repeated.

More importantly, a *proof of reserves* without a corresponding *proof of liabilities* is not sufficient to prove solvency. A proof of liabilities might consist of an audit by a trusted accountant, as done for example by Coinbase[2] and Bitstamp[3]. This might be improved

*Corresponding Author

CCS'15, October 12–16, 2015, Denver, Colorado, USA.

ACM 978-1-4503-3832-5/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2810103.2813674 .

[1]Following convention, we refer to the protocol as 'Bitcoin' and the units of currency as 'bitcoin' or ฿.
[2]A. Antonopoulos, "Coinbase Review," *antonopoulos.com* (Blog), 25 Feb 2014.
[3]E. Spaven, "Bitstamp Passes Audit Overseen by Bitcoin Developer Mike Hearn," *CoinDesk*, 27 May 2014.

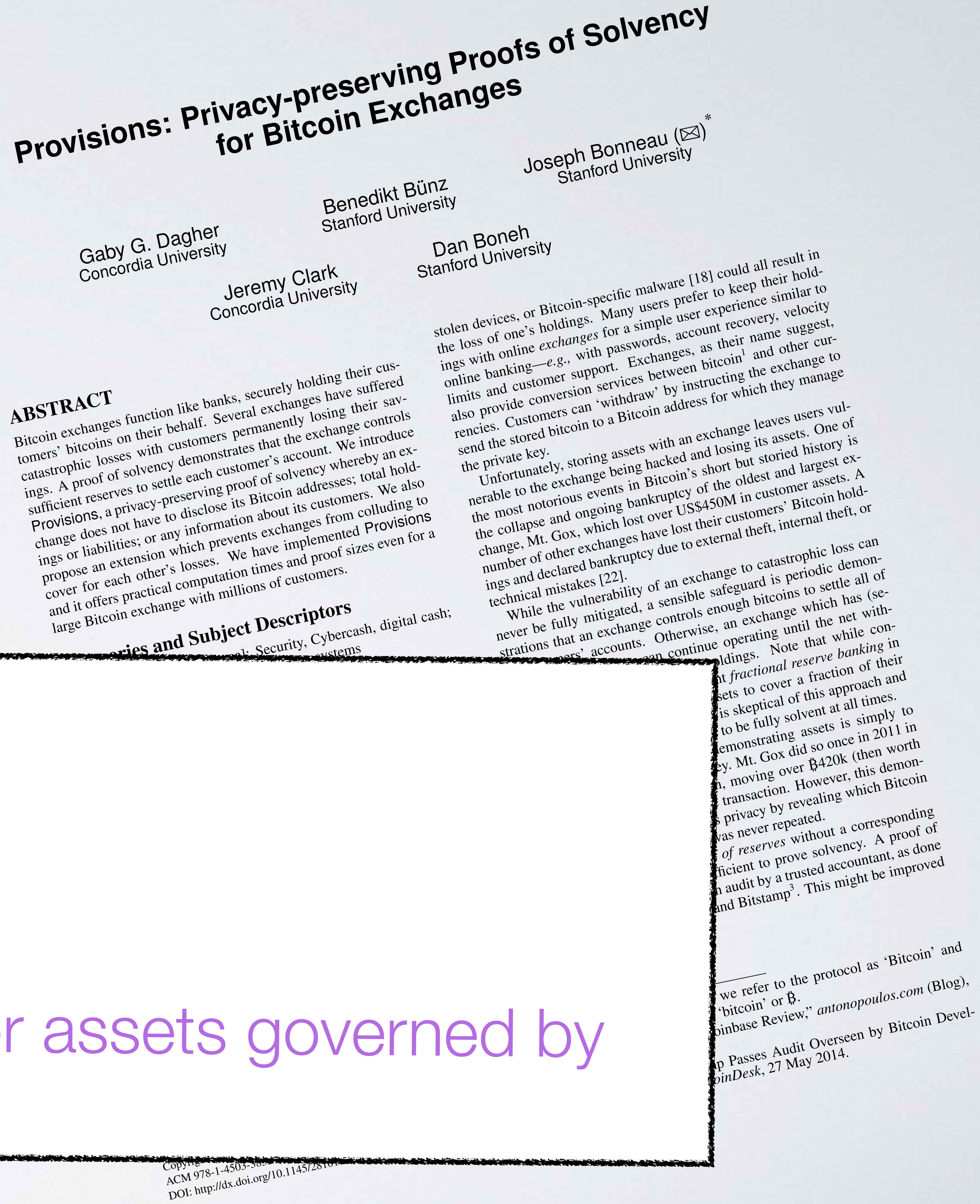
Blockchain protocols & attacks

Proofs of solvency

Upgradability of contracts

Multiple withdrawal attack (ERC20)

Cryptojacking


Provisions: Privacy-preserving Proofs of Solvency for Bitcoin Exchanges

Gaby G. Dagher, Concordia University
Benedikt Bünz, Stanford University
Joseph Bonneau, Stanford University
Jeremy Clark, Concordia University
Dan Boneh, Stanford University

ABSTRACT

Bitcoin exchanges function like banks, securely holding their customers' bitcoins on their behalf. Several exchanges have suffered catastrophic losses with customers permanently losing their savings. A proof of solvency demonstrates that the exchange controls sufficient reserves to settle each customer's account. We introduce Provisions, a privacy-preserving proof of solvency whereby an exchange does not have to disclose its Bitcoin addresses; total holdings or liabilities; or any information about its customers. We also propose an extension which prevents exchanges from colluding to cover for each other's losses. We have implemented Provisions and it offers practical computation times and proof sizes even for a large Bitcoin exchange with millions of customers.

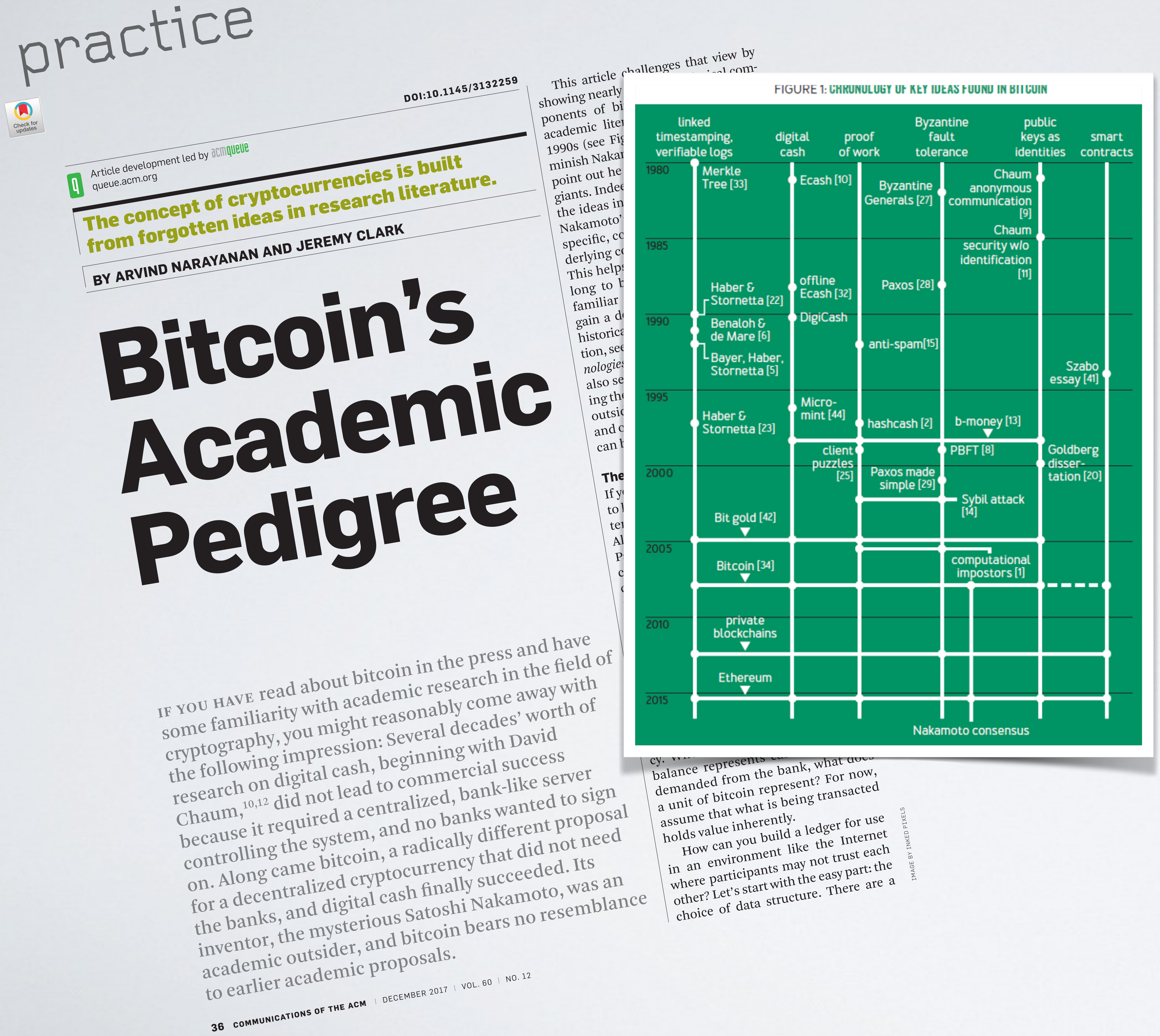
stolen devices, or Bitcoin-specific malware [18] could all result in the loss of one's holdings. Many users prefer to keep their holdings with online *exchanges* for a simple user experience similar to online banking—*e.g.*, with passwords, account recovery, velocity limits and customer support. Exchanges, as their name suggest, also provide conversion services between bitcoin[1] and other currencies. Customers can 'withdraw' by instructing the exchange to send the stored bitcoin to a Bitcoin address for which they manage the private key.

Unfortunately, storing assets with an exchange leaves users vulnerable to the exchange being hacked and losing its assets. One of the most notorious events in Bitcoin's short but storied history is the collapse and ongoing bankruptcy of the oldest and largest exchange, Mt. Gox, which lost over US$450M in customer assets. A number of other exchanges have lost their customers' Bitcoin holdings and declared bankruptcy due to external theft, internal theft, or technical mistakes [22].

Proofs of solvency:

* standardization
* most are broken [K. Chalkias]
* many prove liabilities but not assets
* custom zk-SNARK (IOP) is unnatural for assets governed by ECDSA keys (non-pairing groups)

Systemization of Knowledge (SoKs)

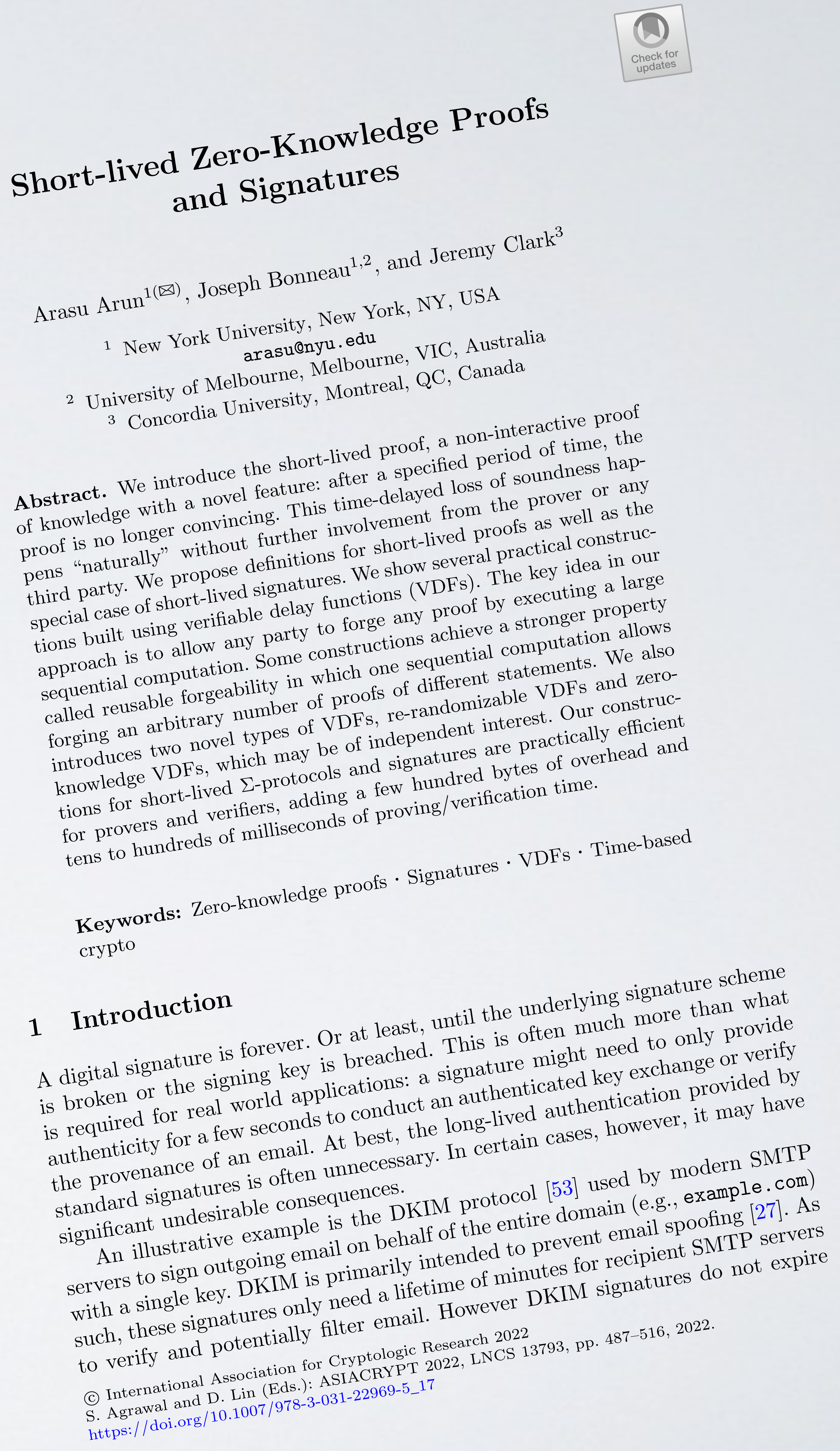


- Bitcoin history
- Front-running
- Stablecoins
- Oracles
- Auditing crypto-assets

- HTTPS
- Encrypted email

Time-based cryptography

Carbon dating protocols
Random beacons
Short-lived signatures and ZKPs
One-time programs (TEE-based)



Short-lived Zero-Knowledge Proofs and Signatures

Arasu Arun[1(✉)], Joseph Bonneau[1,2], and Jeremy Clark[3]

[1] New York University, New York, NY, USA
arasu@nyu.edu
[2] University of Melbourne, Melbourne, VIC, Australia
[3] Concordia University, Montreal, QC, Canada

Abstract. We introduce the short-lived proof, a non-interactive proof of knowledge with a novel feature: after a specified period of time, the proof is no longer convincing. This time-delayed loss of soundness happens "naturally" without further involvement from the prover or any third party. We propose definitions for short-lived proofs as well as the special case of short-lived signatures. We show several practical constructions built using verifiable delay functions (VDFs). The key idea in our approach is to allow any party to forge any proof by executing a large sequential computation. Some constructions achieve a stronger property called reusable forgeability in which one sequential computation allows forging an arbitrary number of proofs of different statements. We also introduces two novel types of VDFs, re-randomizable VDFs and zero-knowledge VDFs, which may be of independent interest. Our constructions for short-lived Σ-protocols and signatures are practically efficient for provers and verifiers, adding a few hundred bytes of overhead and tens to hundreds of milliseconds of proving/verification time.

Keywords: Zero-knowledge proofs · Signatures · VDFs · Time-based crypto

1 Introduction

A digital signature is forever. Or at least, until the underlying signature scheme is broken or the signing key is breached. This is often much more than what is required for real world applications: a signature might need to only provide authenticity for a few seconds to conduct an authenticated key exchange or verify the provenance of an email. At best, the long-lived authentication provided by standard signatures is often unnecessary. In certain cases, however, it may have significant undesirable consequences.

An illustrative example is the DKIM protocol [53] used by modern SMTP servers to sign outgoing email on behalf of the entire domain (e.g., example.com) with a single key. DKIM is primarily intended to prevent email spoofing [27]. As such, these signatures only need a lifetime of minutes for recipient SMTP servers to verify and potentially filter email. However DKIM signatures do not expire

S. Agrawal and D. Lin (Eds.): ASIACRYPT 2022, LNCS 13793, pp. 487–516, 2022.
https://doi.org/10.1007/978-3-031-22969-5_17

Central Bank Digital Currencies (CBDC)

Balancing Privacy and Auditability

DOI:10.1145/3579316

Article development led by acmqueue
queue.acm.org

Now is the time to shape what future payment flows will reveal about you.

BY RAPHAEL AUER, RAINER BÖHME, JEREMY CLARK, AND DIDEM DEMIRAG

Mapping the Privacy Landscape for Central Bank Digital Currencies

PAYMENT RECORDS PAINT a detailed picture of an individual's behavior. They reveal wealth, health, and interests, but individuals do not want the burden of deciding which are sensitive or private.[1] Central banks are exploring options to digitize cash. As of January 2023, 27 of the 38 member states of the Organization for Economic Cooperation and Development (OECD) have announced retail central bank digital currency (CBDC) research and projects.[a]

The issue of privacy needs to move center stage. Decades of work on privacy-enhancing technologies have highlighted that privacy does not come for free, it is easy to get wrong, and it is imperative to design before deployment.

CBDC has been discussed in policy reports, academic papers, and public media through lenses such as monetary policy,[6] impact on the financial system,[2] and technology.[3] Almost all of these documents flag the importance of privacy, but many lack in-depth discussion or concrete design choices. Figure 1 shows the uptake of privacy in the CBDC literature: While the question is raised, significant treatment is still rare. An exception is recent academic papers (shown in the top right corner of the figure), which are generally written by computer scientists. These papers offer specific solutions to include in the privacy design landscape.

Policymakers may shy away from papers with cryptographic equations that mention Alice and Bob. While there are exceptions,[9] the gap in concrete privacy solutions in policy reports is puzzling, as economists have argued that CBDC could make an essential difference in providing privacy in digital payments.[10] It is popular for authors of these reports to point out the tension between privacy and law enforcement; reiterate that it requires a solution; and ultimately punt to government officials, legislators, the judiciary, or public opinion to solve it. Occasionally, technical solutions are prescribed (for example, blockchains, cryptography, zero-knowledge proofs) without adequate operational details or even precision about exactly what data is protected from whom. The number of distinct stakeholders, combined with the technical challenges, has stalled progress toward deploying retail CBDC.

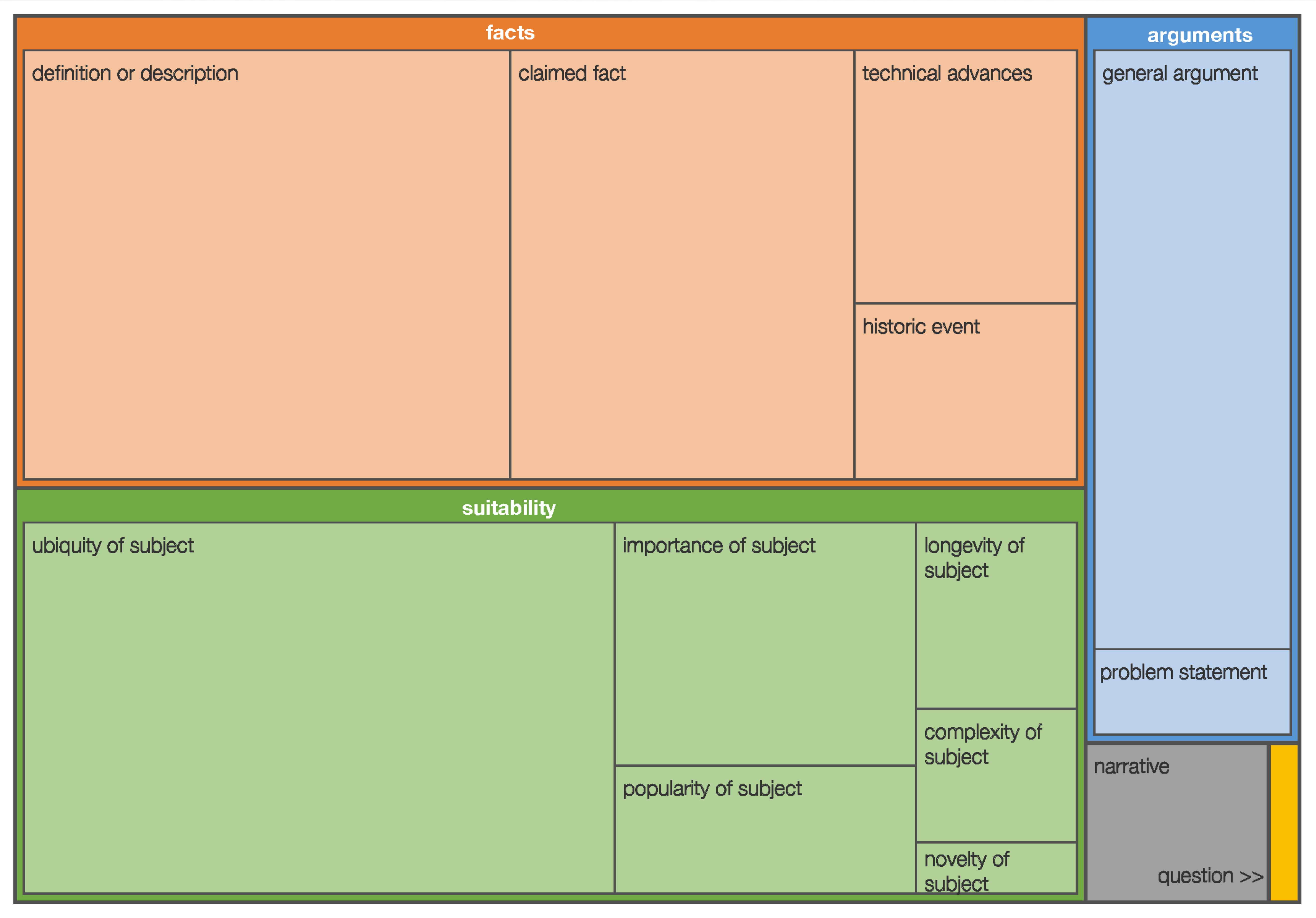
One step forward is understanding who the key stakeholders are and what their interests are in payment records. Knowledge of conflicting interests is helpful for developing requirements and narrowing the range of technical solutions. This article contributes to the literature by identifying three stakeholder groups—privacy-conscious users, data holders, and law enforcement—and exploring their conflicts at a high level.

A main insight is that nuanced da-

IMAGE BY ANDRIJ BORYS ASSOCIATES, USING SHUTTERSTOCK

a See the January 2023 dataset update at https://www.bis.org/publ/work880.htm

Fun Projects



Opening sentences of ~400 USENIX Security papers

Opening Sentences in Academic Writing

How Security Researchers Defeat the Blinking Cursor

Didem Demirag
Concordia University
Montreal, QC, Canada
d_demira@encs.concordia.ca

Jeremy Clark
Concordia University
Montreal, QC, Canada
j.clark@concordia.ca

ABSTRACT

Traditionally, education in computer science focuses on stakeholders like teachers, undergraduate students, and employers. However researchers also educate themselves about recent results and new subject matters. An important vehicle in this informal, self-education process is reading peer-reviewed academic papers—papers that are also used in the curriculum of graduate-level research courses. Technical writing skills are important in this domain, as well as engaging the reader with interesting text. This paper is a study of academic writing. We study in depth the first sentence used by researchers in opening their academic papers and how this sentence operates to draw the reader in. We use a corpus of 379 papers from a top-tier cybersecurity conference and use qualitative analysis (coding from grounded theory) to create a taxonomy of 5 general types and 14 sub-types of opening sentences. In this paper, we define and illustrate each type through examples, and reflect on what we learned about writing after examining all of these sentences.

CCS CONCEPTS

• **Social and professional topics** → **Informal education**; **Computational thinking**.

KEYWORDS

Scientific Writing; Education; Cybersecurity

ACM Reference Format:
Didem Demirag and Jeremy Clark. 2022. Opening Sentences in Academic Writing: How Security Researchers Defeat the Blinking Cursor. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2022), March 3–5, 2022, Providence, RI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3478431.3499378

1 INTRODUCTORY REMARKS

What makes the writing style of an academic paper stand out? Strong technical writing is partially founded on *SIGCSE* research dating back to the 1990s on how to move writing from the English department into computer science [27, 36, 55, 62]. Technical writing is now considered a necessary communication skill in our curriculum (*cf. ACM/IEEE Computing Curricula 2020* [15, 16]).

In a departure from many *SIGCSE* papers, educators in our work are not teachers in a classroom but are authors of academic papers who transfer knowledge to their readers. A well-written paper can reach a wide audience beyond conference attendees and others who benefit from direct communication with the authors. Further, academic papers are used in the classroom, particularly in courses offered to graduate students. For a paper to be effective, it must first catch the attention of the course instructor and then engage the interest of the students. While most researchers are, or were, students, and writing is included in modern curricula, writing research papers is an advanced form of writing considered too ambitious for teaching to undergraduates [30]. Therefore, crafting academic papers is often self-taught and/or passed from supervisors to students through mentorship [48].

Writing tools and systems help produce competent text, but tend not to enhance elegance and style. If writing style is difficult to teach and analyze, what can be done? In this paper, we build a 'zoo' of writing samples (conceptualized by Miro [48] at *SIGCSE'11*), placed in a taxonomy we develop, to exhibit different writing styles for writers and readers to study and learn from. The specific lessons drawn from viewing our zoo are meant to be subjective, and depend on the viewer's personal context. A secondary contribution of our paper is our methodology for building a zoo. We hope to see this applied to other domains of writing.

Our methodology is empirical and positive. To illustrate what we mean, consider a research paper that designs and tests a pedological tool in an educational setting. By contrast, an *empirical* paper might survey a set of courses from around the world. Often before a *normative* approach (*i.e.*, what ought to be done) can be formulated, it is instructive to first consider a *positive* approach (what is being done). Our zoo is not a curation of 'good' writing samples (beyond being acceptable for publication in a top-tier conference) but offers a large set of samples that are carefully organized by our interpretation of what the writers intended to convey.

The Opening Sentence. We believe it is fruitful to study the writing style of academic computer science articles at different levels of granularity. As a general trade-off, short writing samples enable the study of a large number of samples, while longer writing samples enable a broader representation of the writing style. In this work, we choose to look at a large number of very short samples—each only a single sentence. In choosing which sentence to study from each paper, the first sentence is an intuitive candidate. The opening sentence of a paper needs to be bold, convey the importance of the subject of the paper, and hook the reader. The novelist Stephen

ACM ISBN 978-1-4503-9070-5/22/03…$15.00
https://doi.org/10.1145/3478431.3499378

Fast withdrawals from optimistic rollups

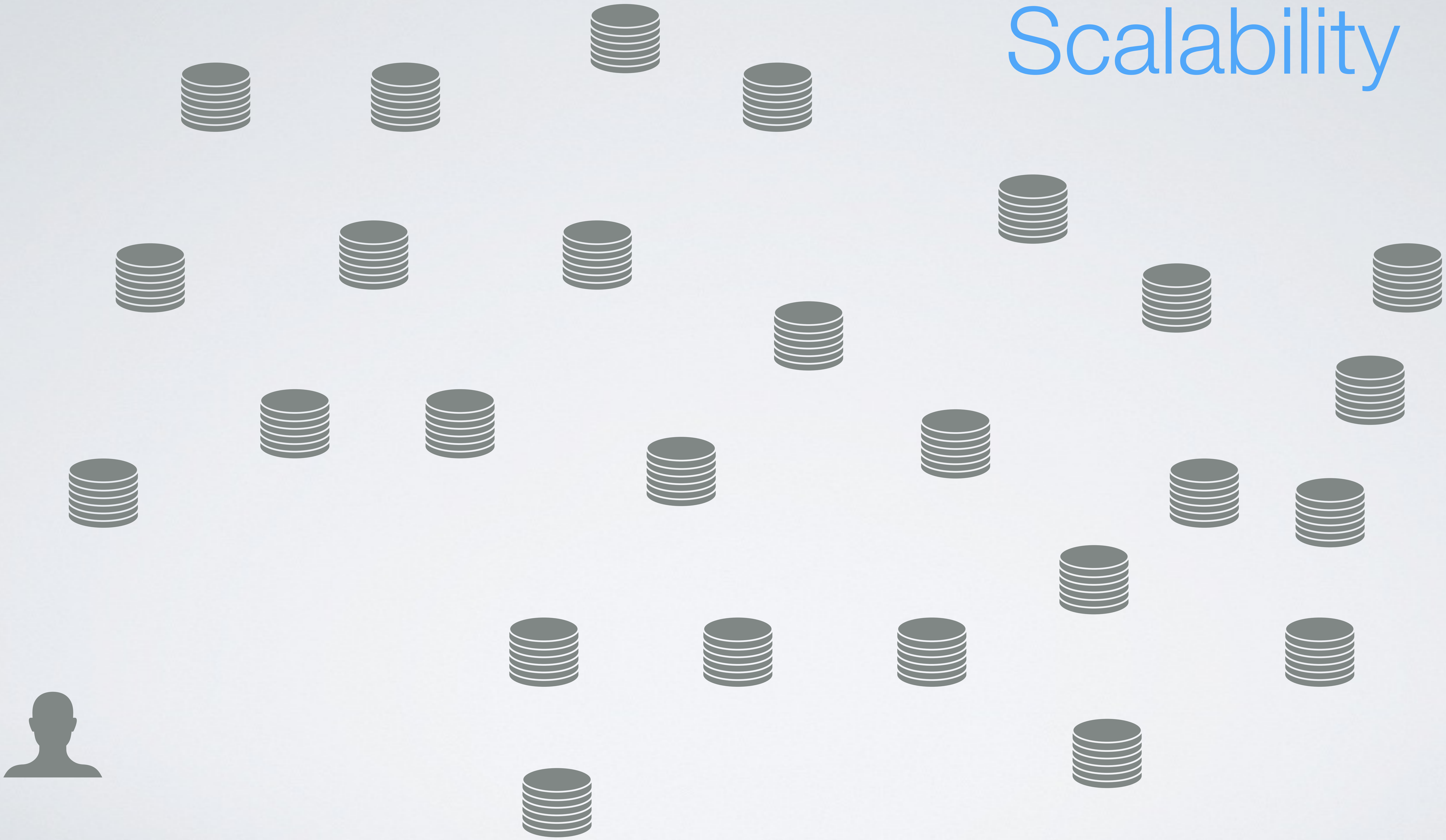
MAHSA MOOSAVI, CONCORDIA UNIVERSITY / OFFCHAIN LABS

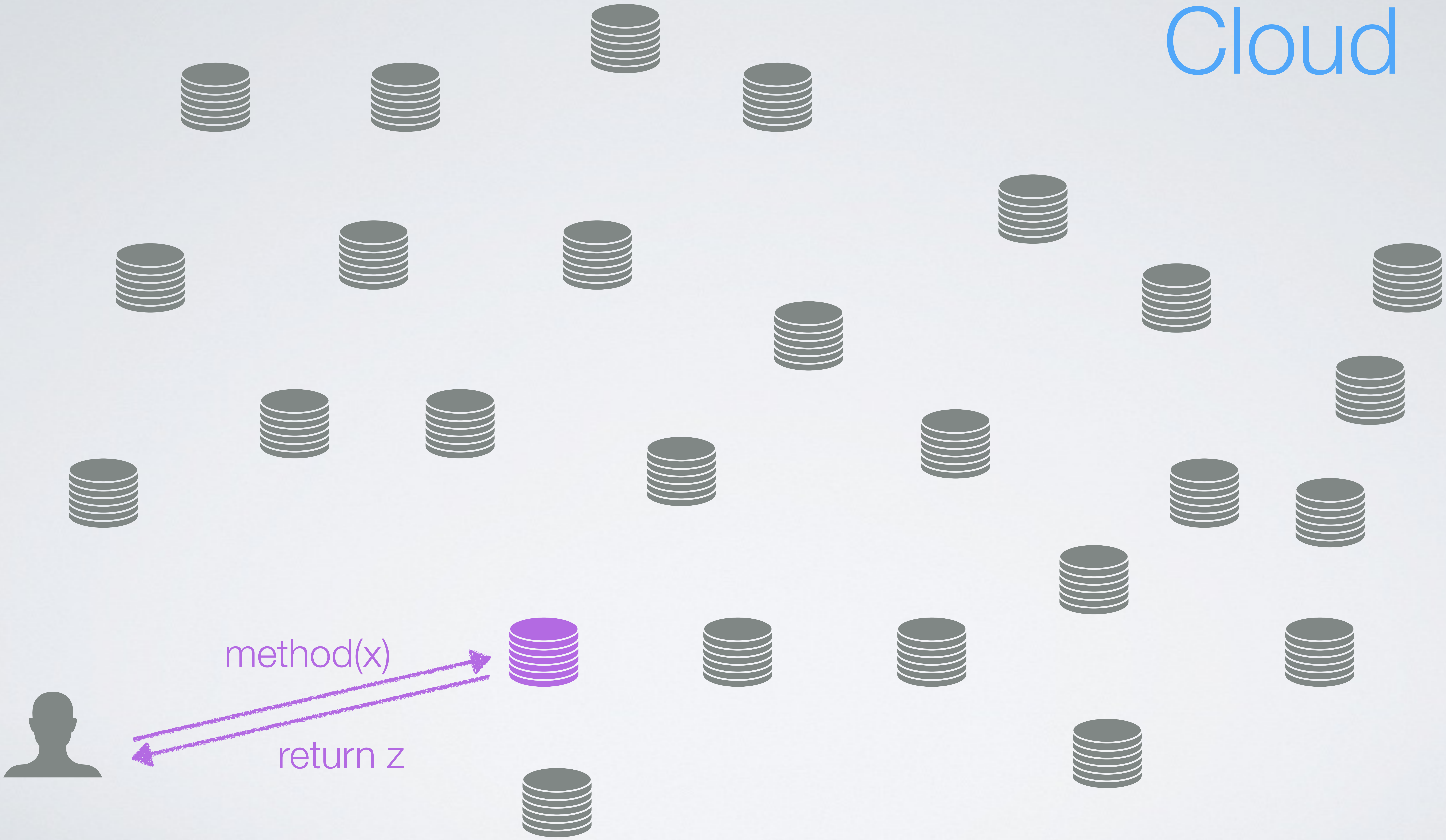
MEHDI SALEHI, OFFCHAIN LABS

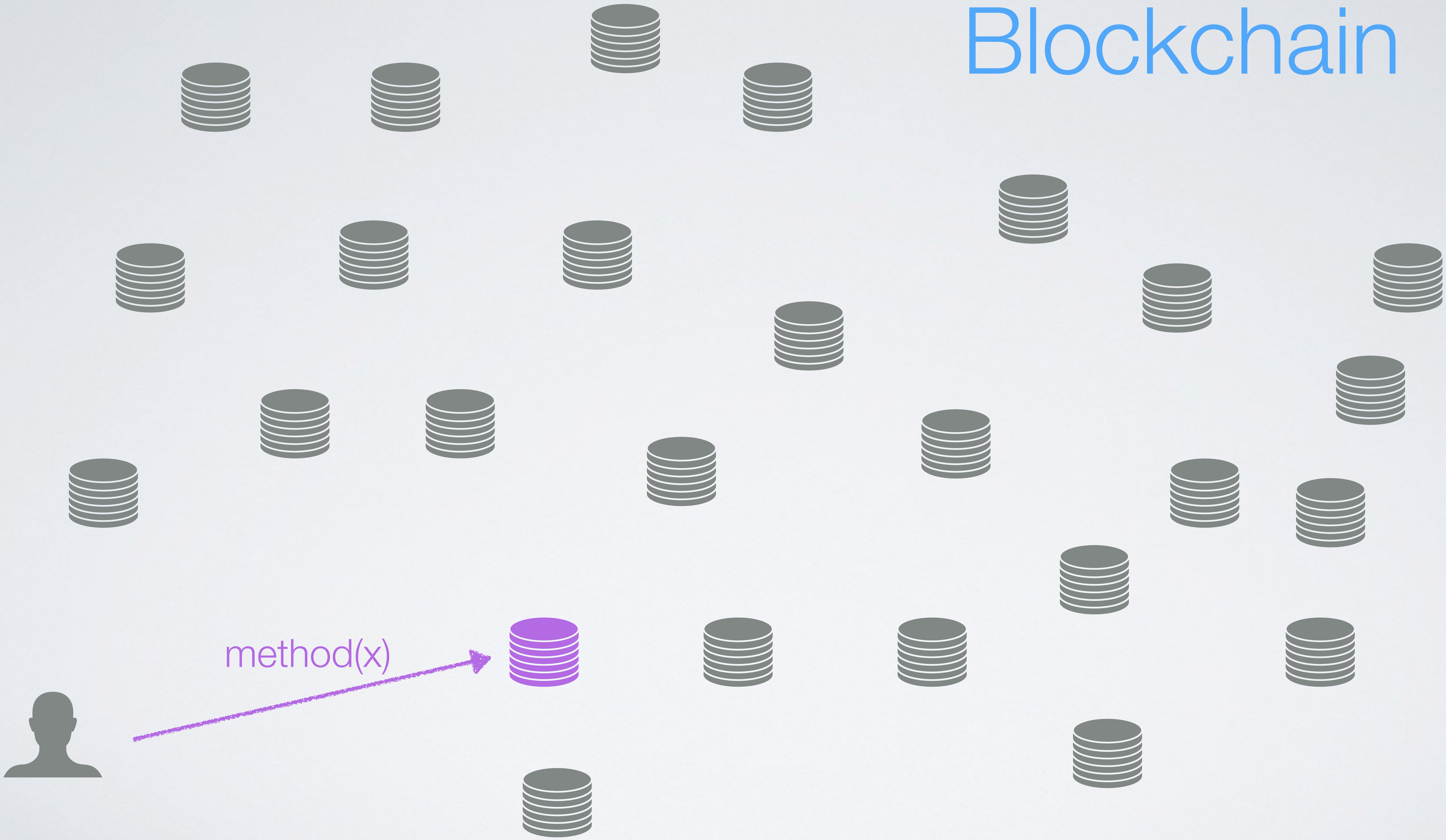
DANIEL GOLDMAN, OFFCHAIN LABS

JEREMY CLARK, CONCORDIA UNIVERSITY

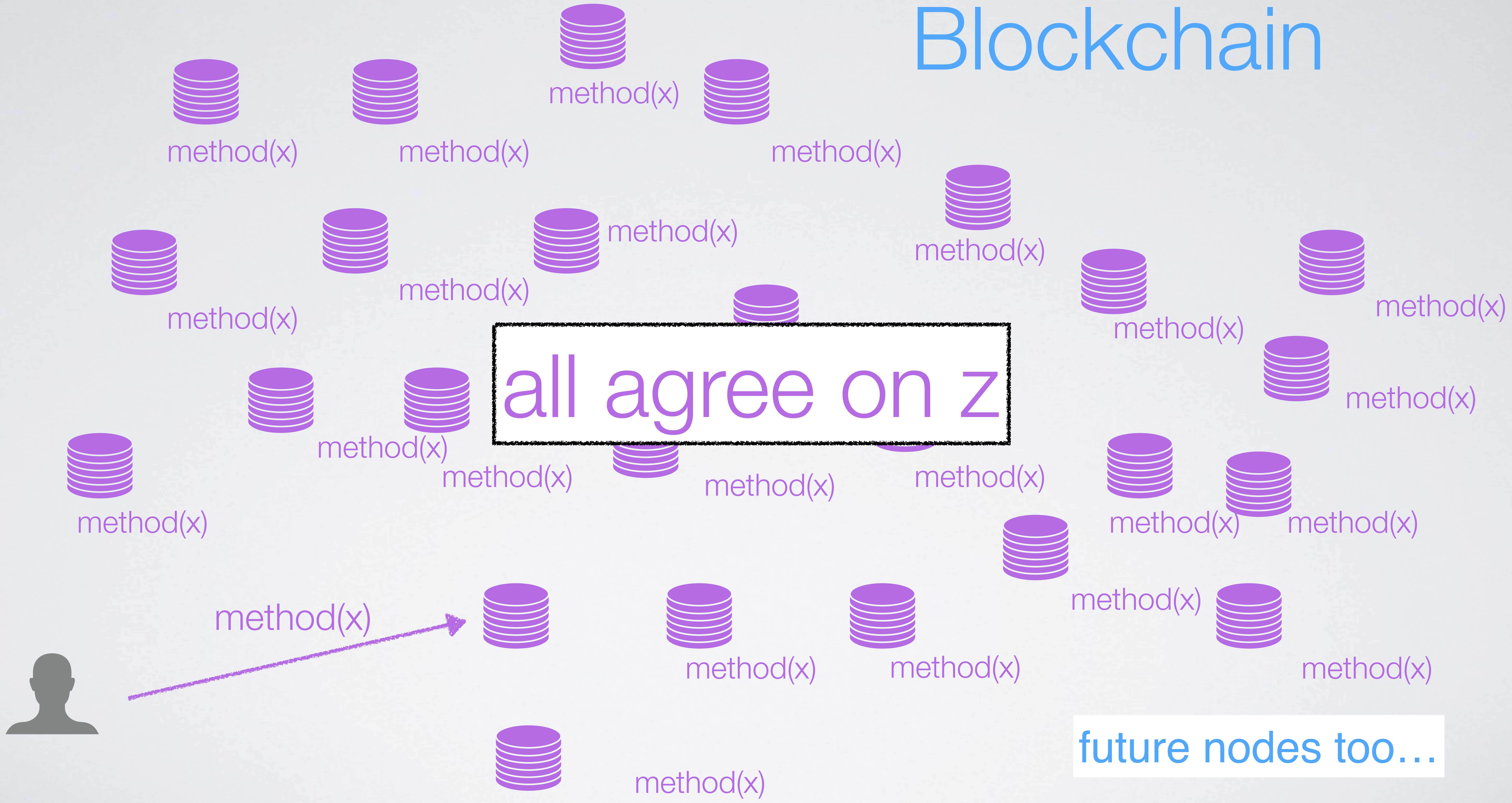
Scalability



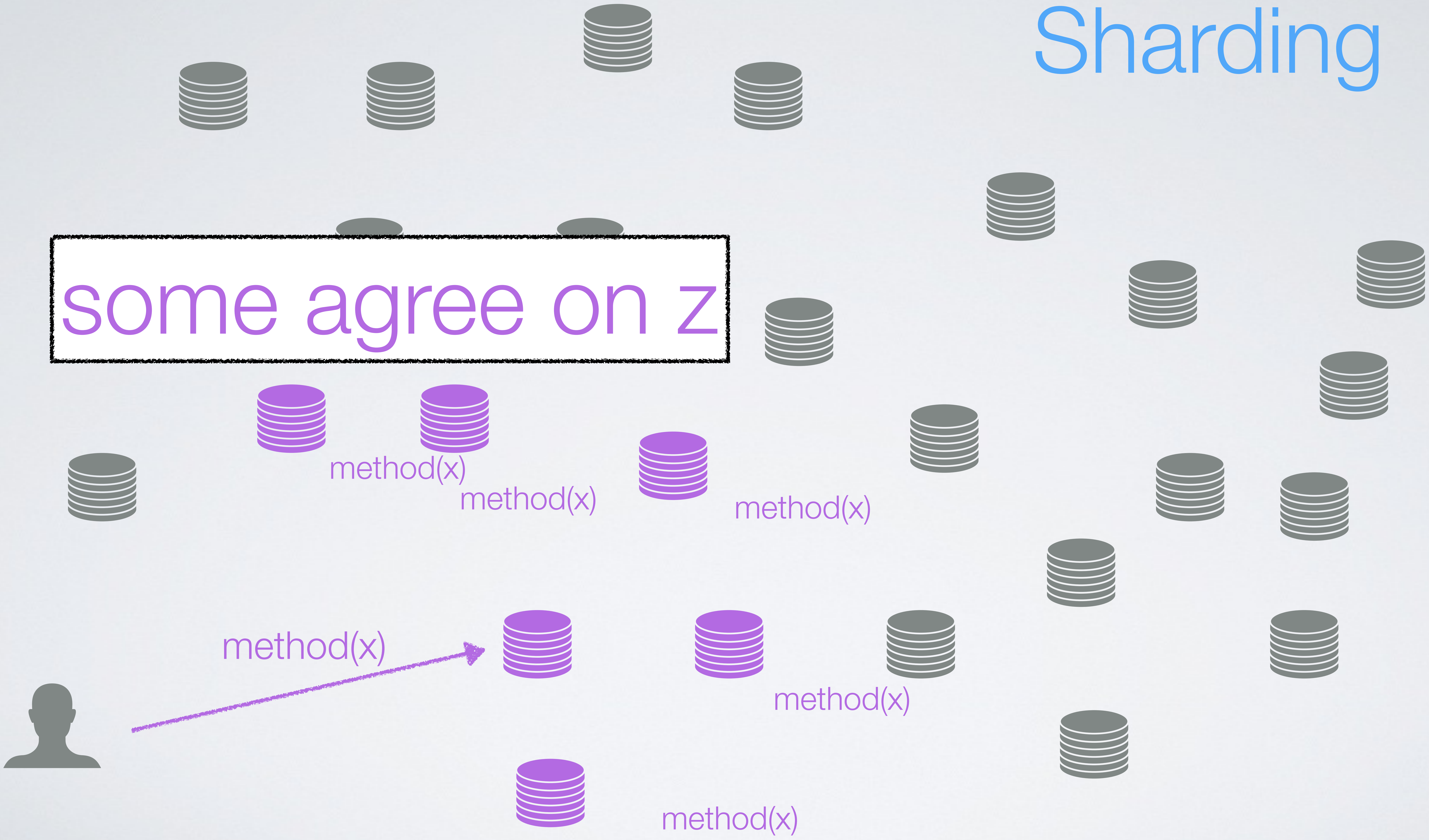

Cloud
method(x)
return z


Blockchain
method(x)

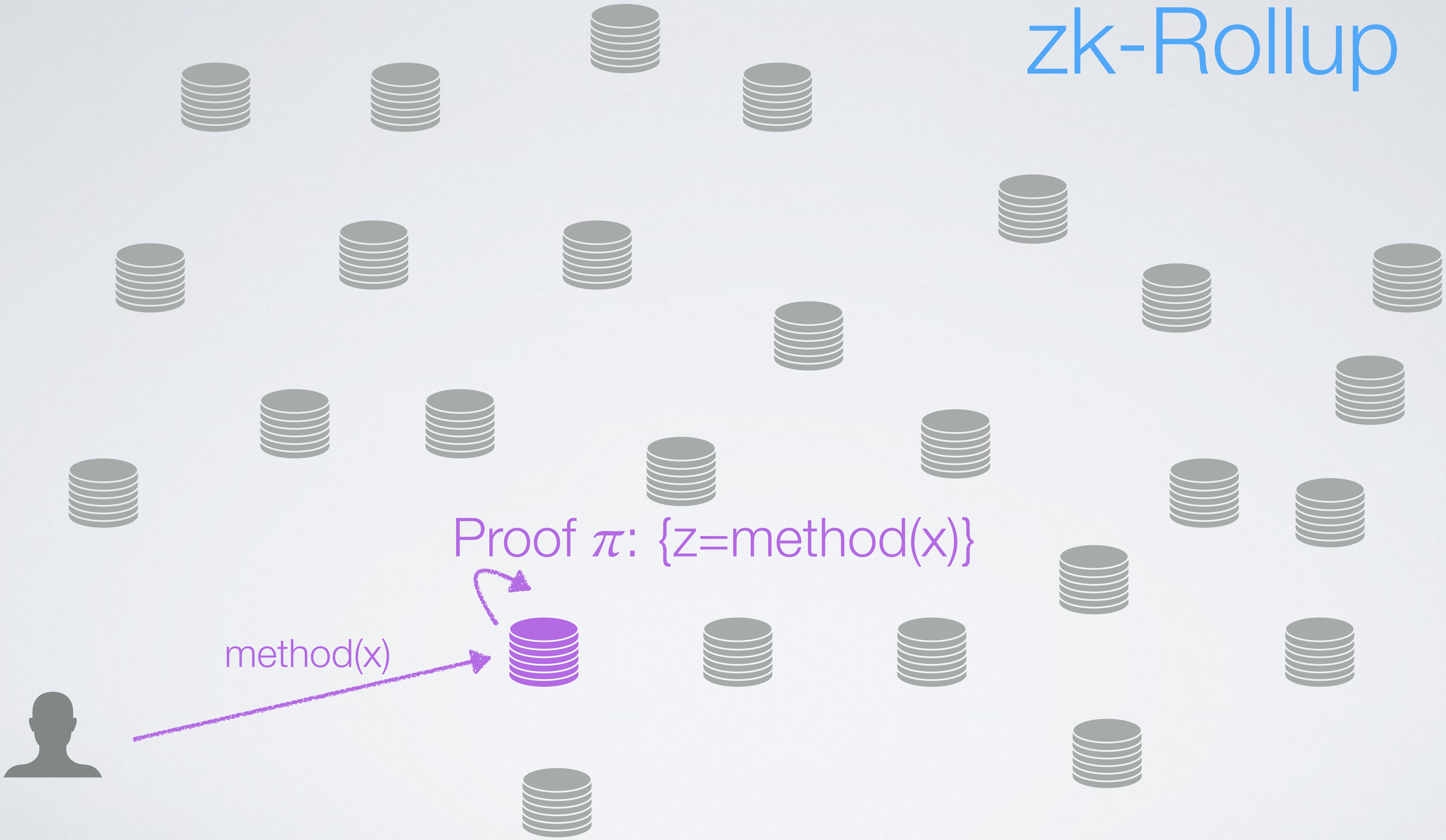
Blockchain


method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
all agree on z
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)
future nodes too…

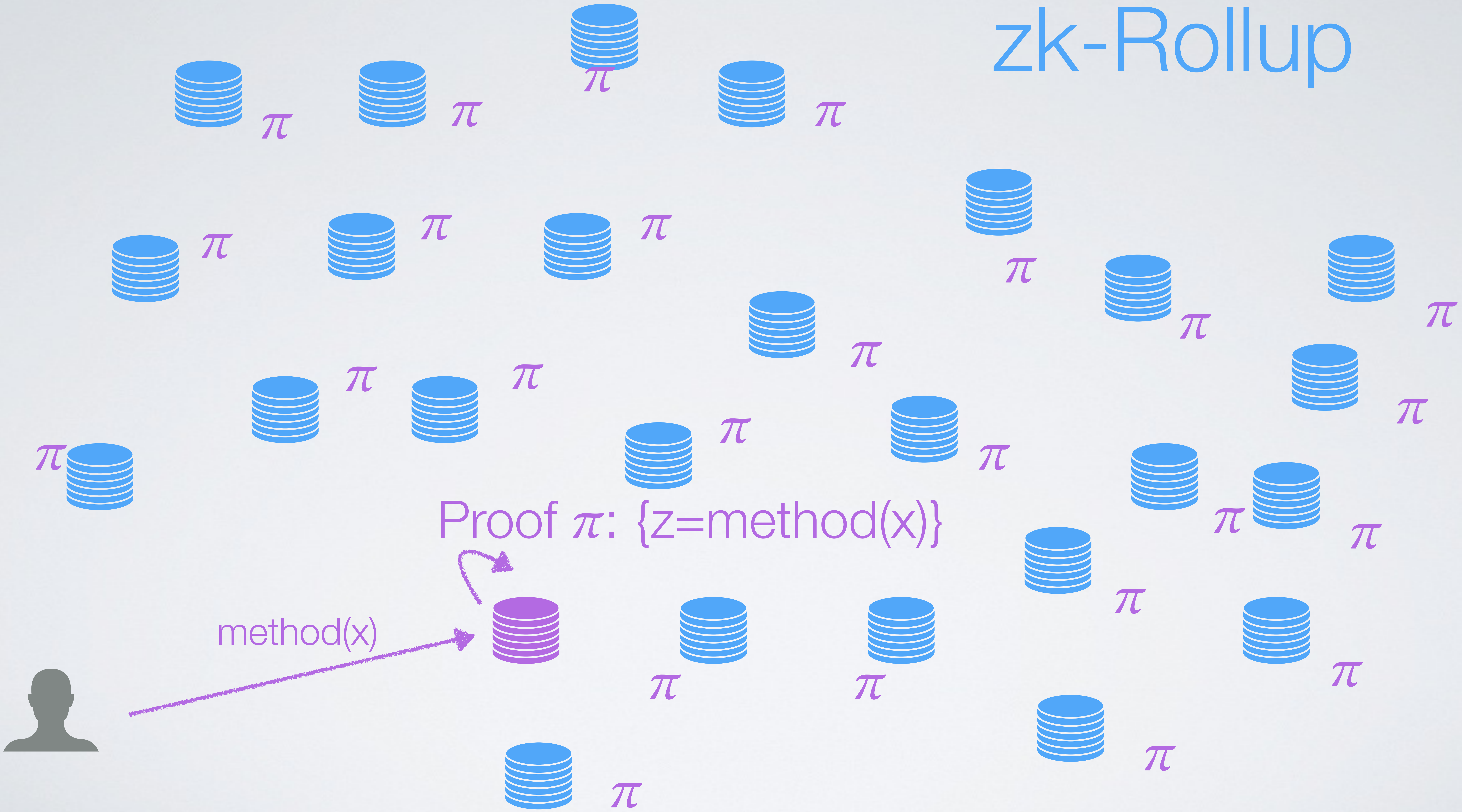
Sharding


some agree on z
method(x)
method(x)
method(x)
method(x)
method(x)
method(x)

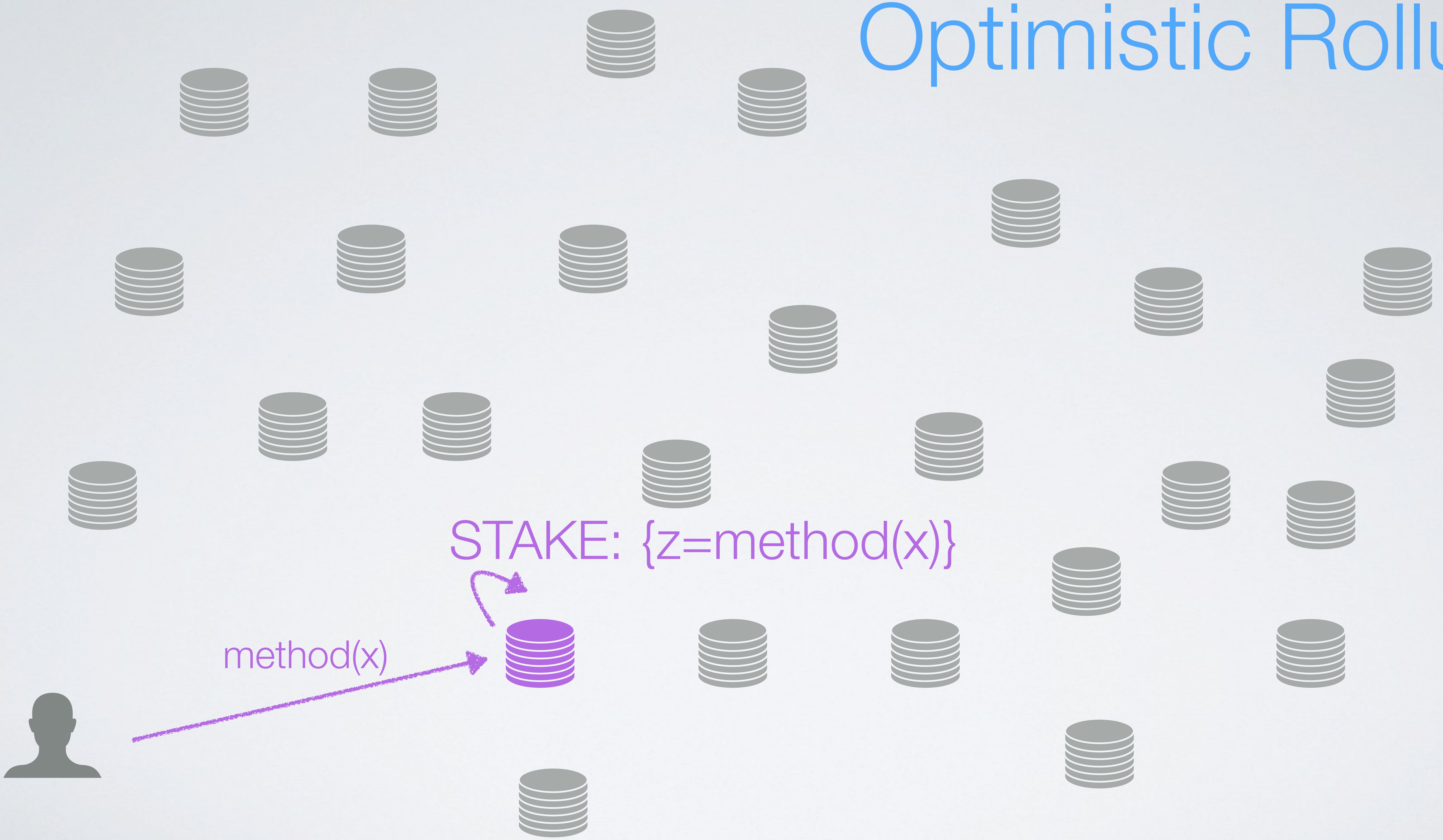
zk-Rollup



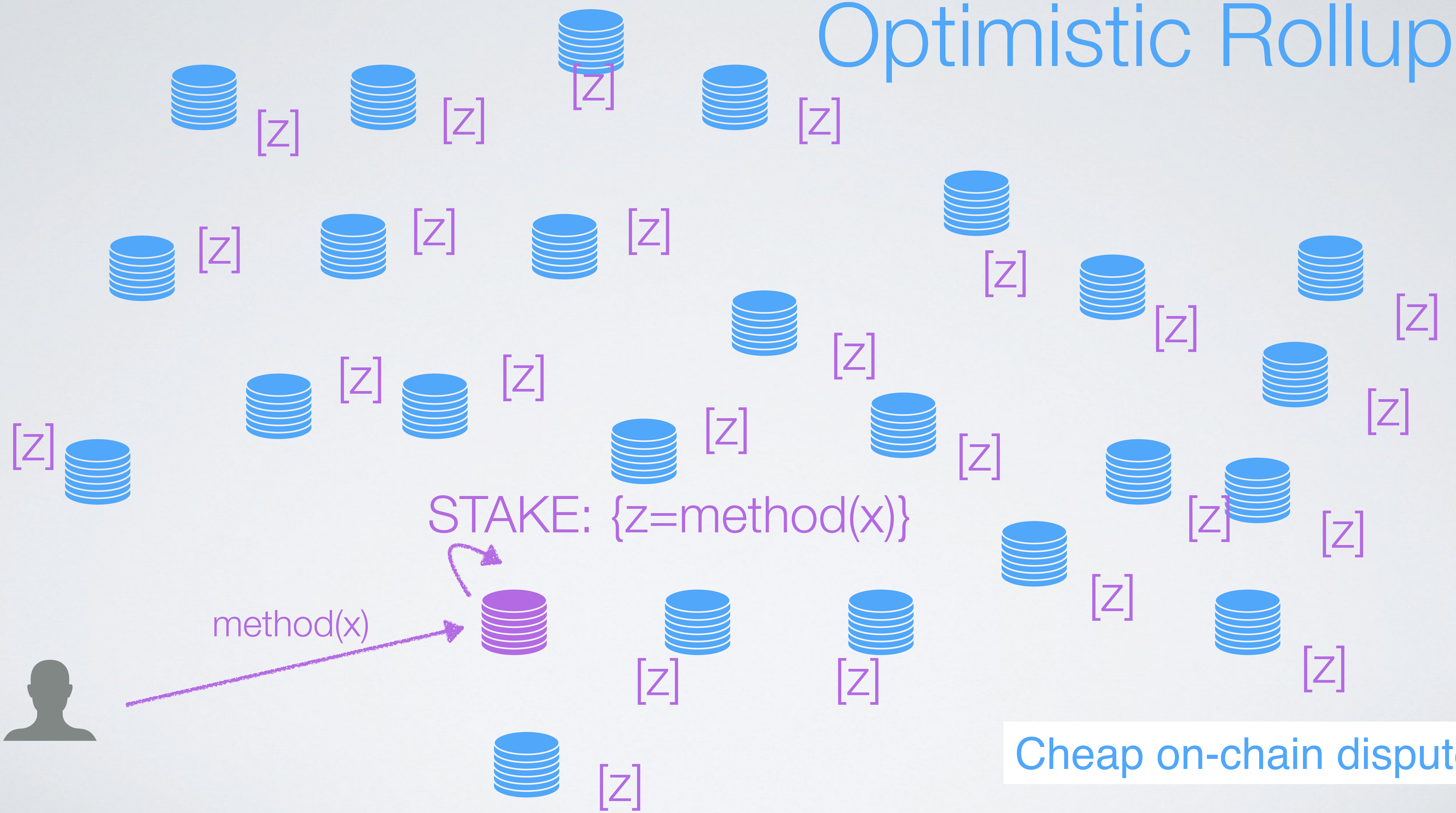
zk-Rollup



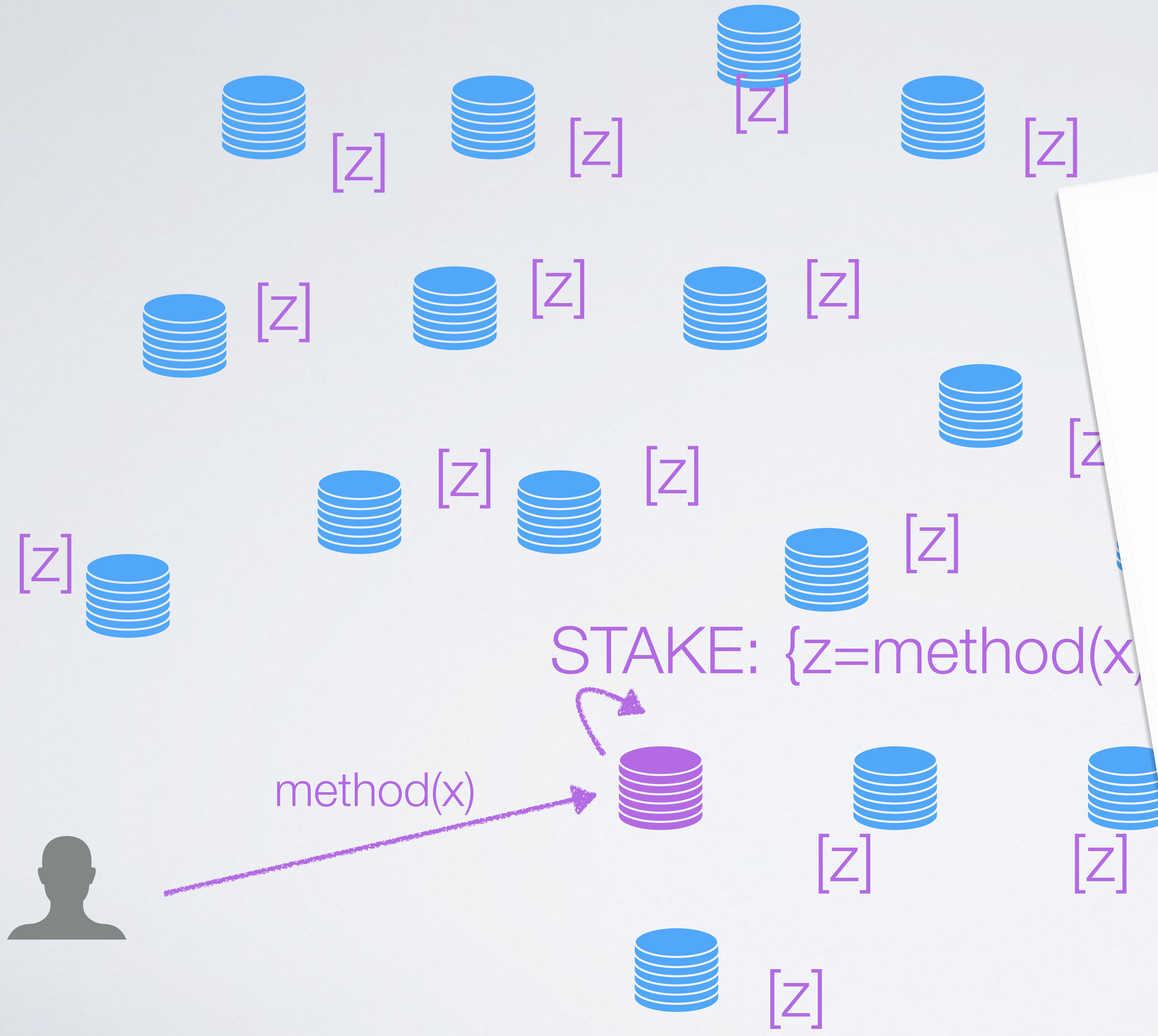
Optimistic Rollup

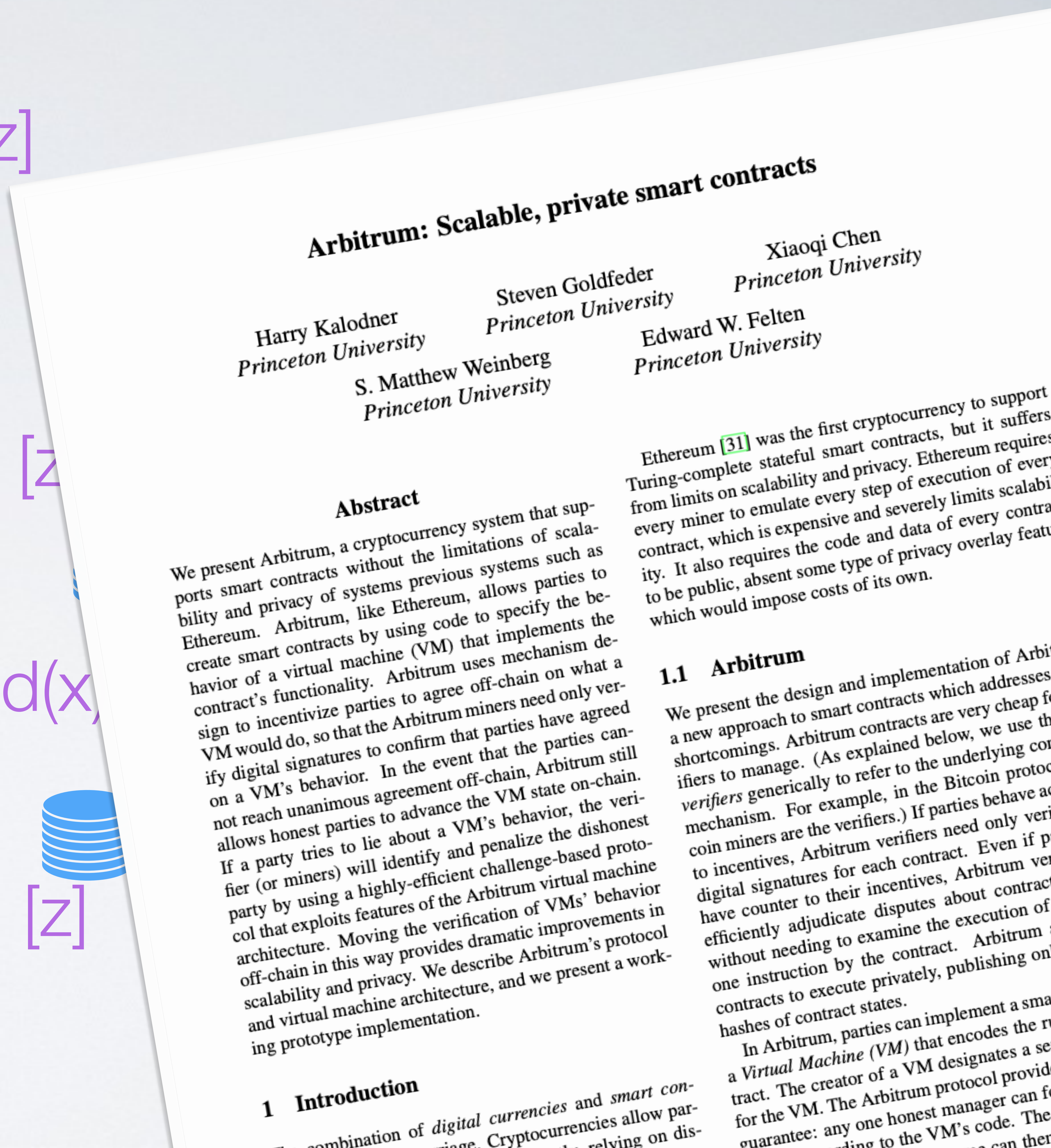


Optimistic Rollup



Cheap on-chain disputes




Arbitrum: Scalable, private smart contracts

Harry Kalodner
Princeton University

Steven Goldfeder
Princeton University

Xiaoqi Chen
Princeton University

S. Matthew Weinberg
Princeton University

Edward W. Felten
Princeton University

Abstract

We present Arbitrum, a cryptocurrency system that supports smart contracts without the limitations of scalability and privacy of systems previous systems such as Ethereum. Arbitrum, like Ethereum, allows parties to create smart contracts by using code to specify the behavior of a virtual machine (VM) that implements the contract's functionality. Arbitrum uses mechanism design to incentivize parties to agree off-chain on what a VM would do, so that the Arbitrum miners need only verify digital signatures to confirm that parties have agreed on a VM's behavior. In the event that the parties cannot reach unanimous agreement off-chain, Arbitrum still allows honest parties to advance the VM state on-chain. If a party tries to lie about a VM's behavior, the verifier (or miners) will identify and penalize the dishonest party by using a highly-efficient challenge-based protocol that exploits features of the Arbitrum virtual machine architecture. Moving the verification of VMs' behavior off-chain in this way provides dramatic improvements in scalability and privacy. We describe Arbitrum's protocol and virtual machine architecture, and we present a working prototype implementation.

1 Introduction

1.1 Arbitrum

CoinDesk

Technology

Layer 2 Network Arbitrum Surpasses Ethereum in Daily Transactions

Arbitrum's dominance continues to grow in the first quarter of 2023 as the number of unique addresses on Arbitrum reaches an all-time high.

By Sage D. Young

Feb 22, 2023 at 1:00 a.m.

Updated Feb 22, 2023 at 10:34 a.m.

Active projects | Upcoming projects | Archived projects

#	NAME	RISKS	TECHNOLOGY	STAGE	PURPOSE	TVL	MKT SHARE
1	Arbitrum One		Optimistic Rollup	STAGE 1	Universal	$5.94B ▲ 9.52%	60.81%
2	OP Mainnet		Optimistic Rollup OP	STAGE 0	Universal	$2.20B ▲ 7.12%	22.60%
3	zkSync Era		ZK Rollup	STAGE 0	Universal	$645M ▲ 22.18%	6.61%
4	dYdX		ZK Rollup	STAGE 1	Exchange	$351M ▲ 5.08%	3.60%
5	Immutable X		Validium	n/a	NFT, Exchange	$101M ▲ 16.53%	1.04%
6	Metis Andromeda		Optimistic Chain OP	n/a	Universal	$99.00M ▲ 5.14%	1.01%
7	Loopring		ZK Rollup	STAGE 0	Tokens, NFTs, AMM	$98.03M ▲ 8.57%	1.00%
8	zkSync Lite		ZK Rollup	STAGE 1	Payments, Tokens	$80.93M ▼ 7.05%	0.83%
9	Starknet		ZK Rollup	STAGE 0	Universal	$67.49M ▲ 7.15%	0.69%
10	Polygon zkEVM		ZK Rollup	STAGE 0	Universal	$42.70M ▲ 12.35%	0.44%

Transaction output is correct

1. I believe it because I ran it myself (de facto)
2. I believe it because it was signed by lots of people (PoA)
3. I believe it because it someone put a lot of PoW into extending it (SPV)
4. I believe it because I checked a mathematical proof (zk-rollup)
5. I believe it because someone staked on it being correct (optimistic rollup)

Transaction output is correct

1. I believe it because I ran it myself (de facto)
2. I believe it because it was signed by lots of people (PoA)
3. I believe it because it someone put a lot of PoW into extending it (SPV)
4. I believe it because I checked a mathematical proof (zk-rollup)
5. I believe it because someone staked on it being correct (optimistic rollup)

- Optimistic rollups are 5 (+2 in practice)
- They are “optimistic” because you still need to do (1-4) if there is a dispute
- Idea: make spurious disputes costly so they do not occur: all assertions require ETH
- Idea: make disputes have low gas costs: dispute is narrowed to one OPCODE in the execution path

Optimistic Rollup

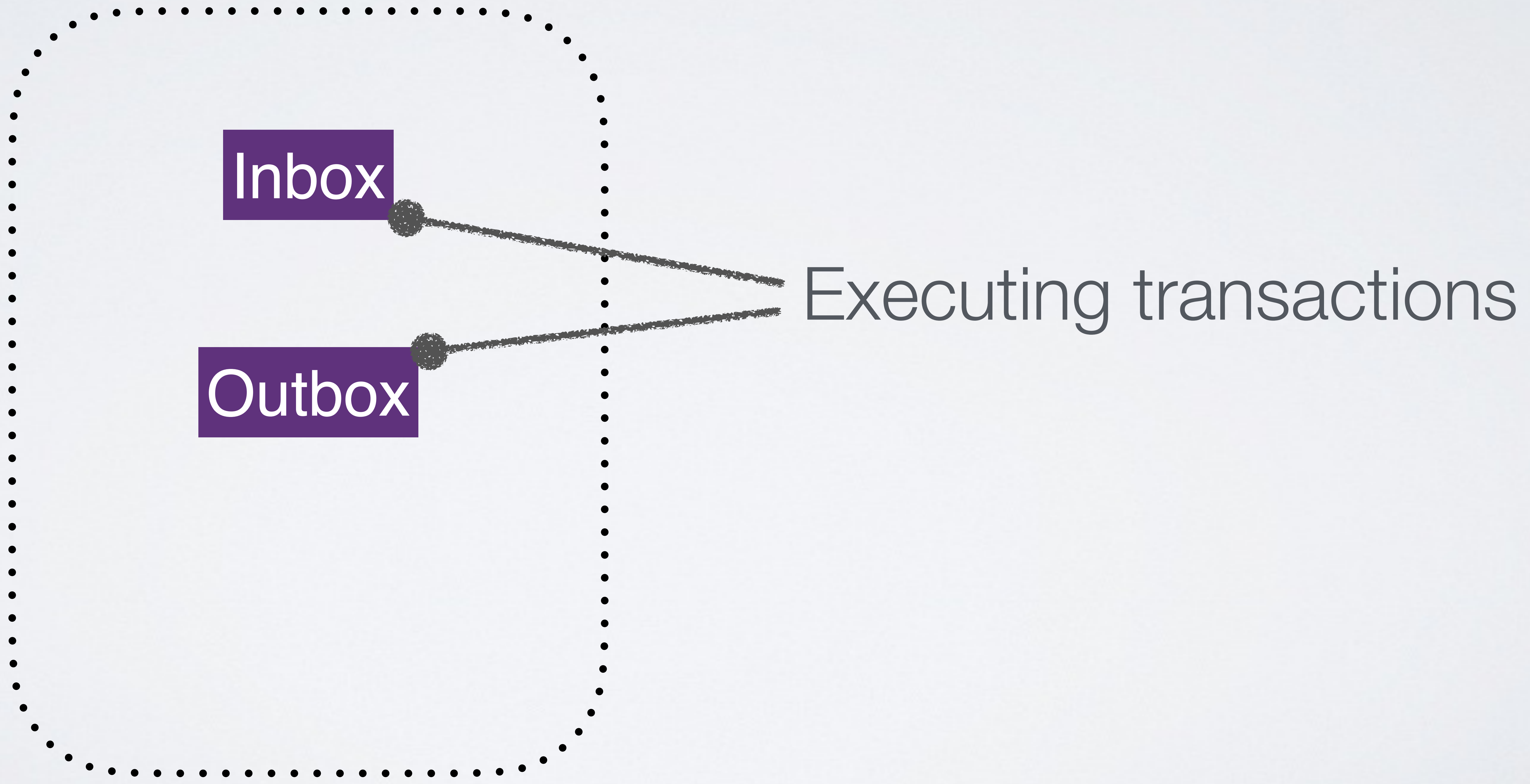
On-Chain (L1): Ethereum

Off-Chain (L2): ArbOS

Optimistic Rollup

On-Chain (L1): Ethereum

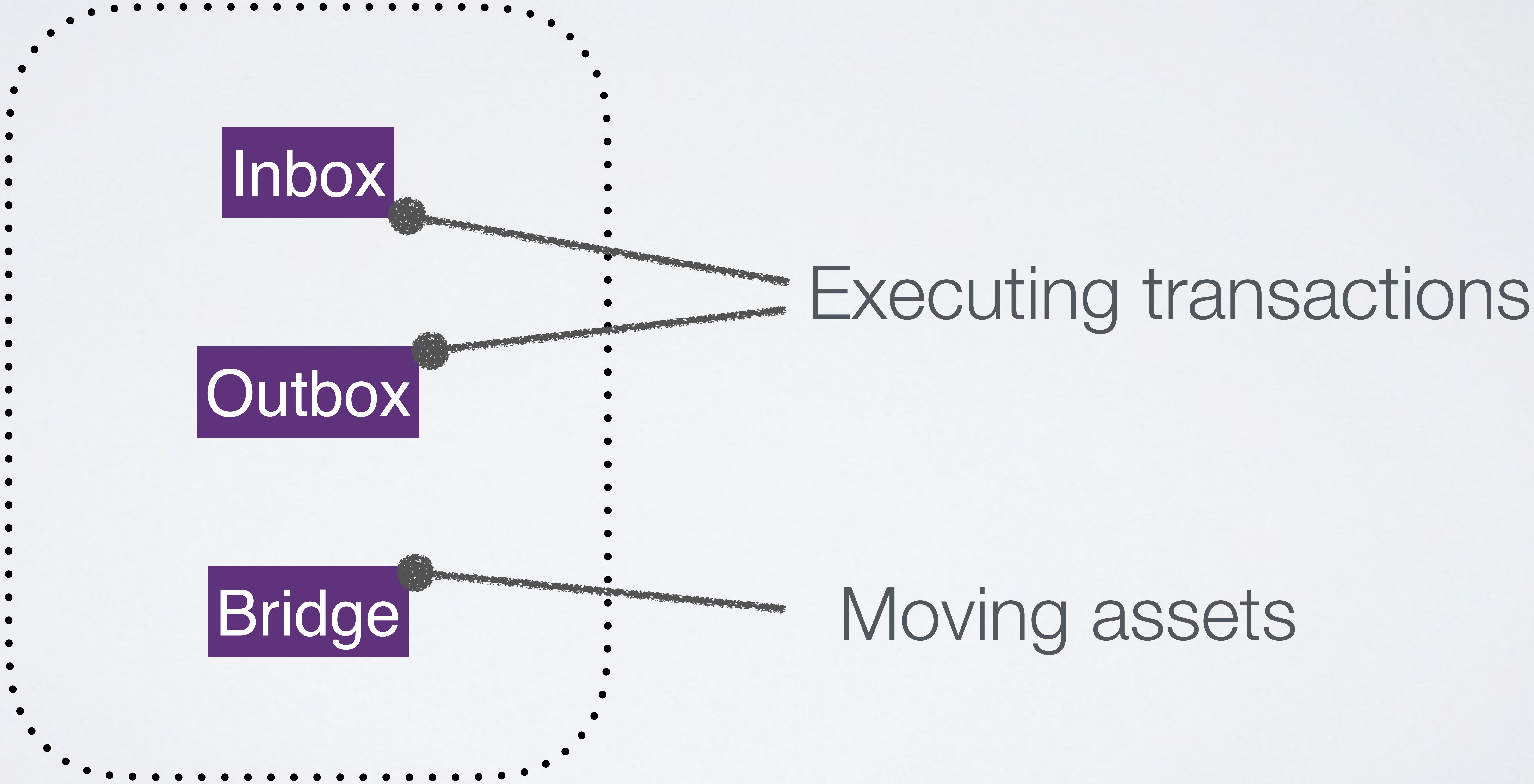
Off-Chain (L2): ArbOS



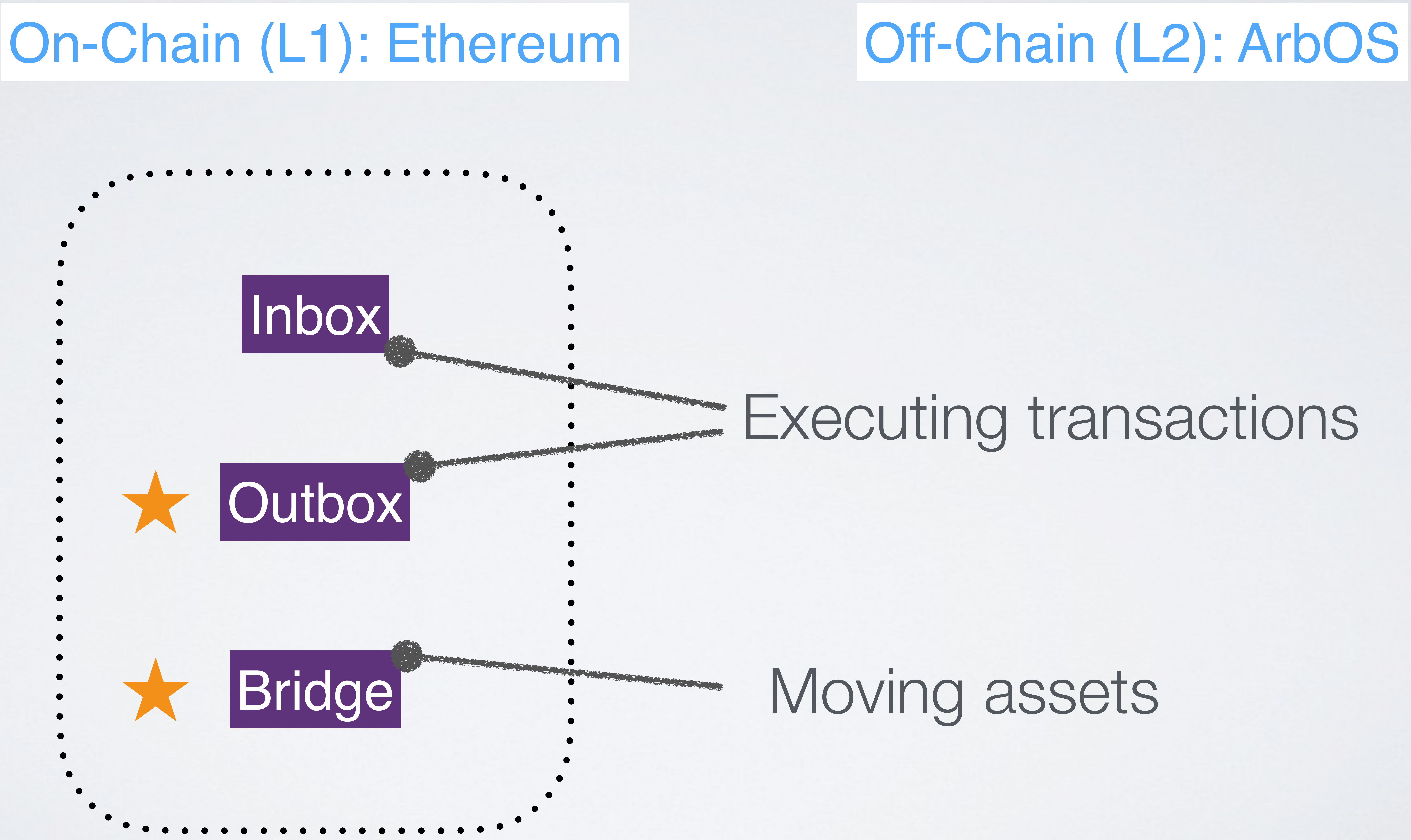
Optimistic Rollup

On-Chain (L1): Ethereum

Off-Chain (L2): ArbOS



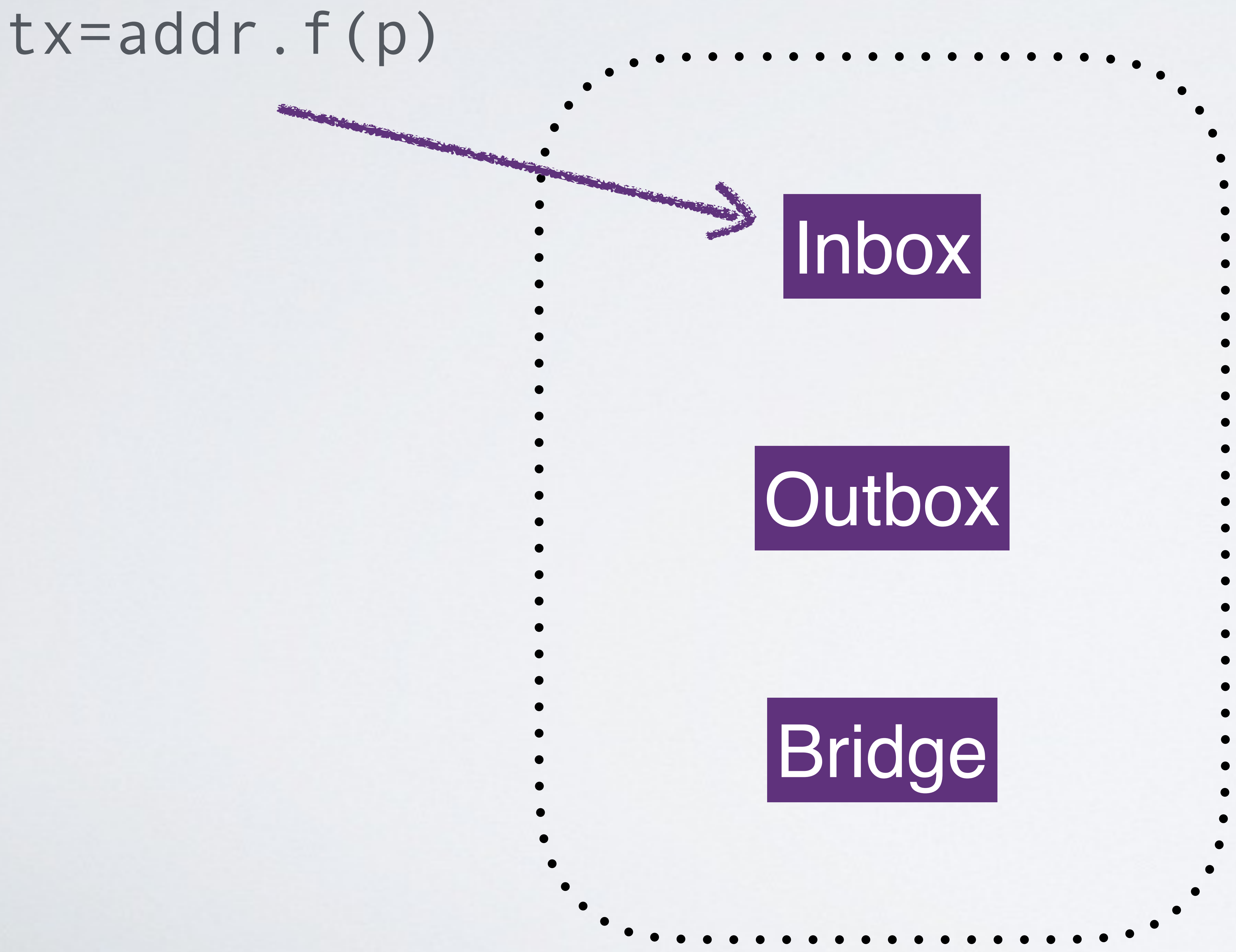
Optimistic Rollup



Optimistic Rollup

On-Chain (L1): Ethereum

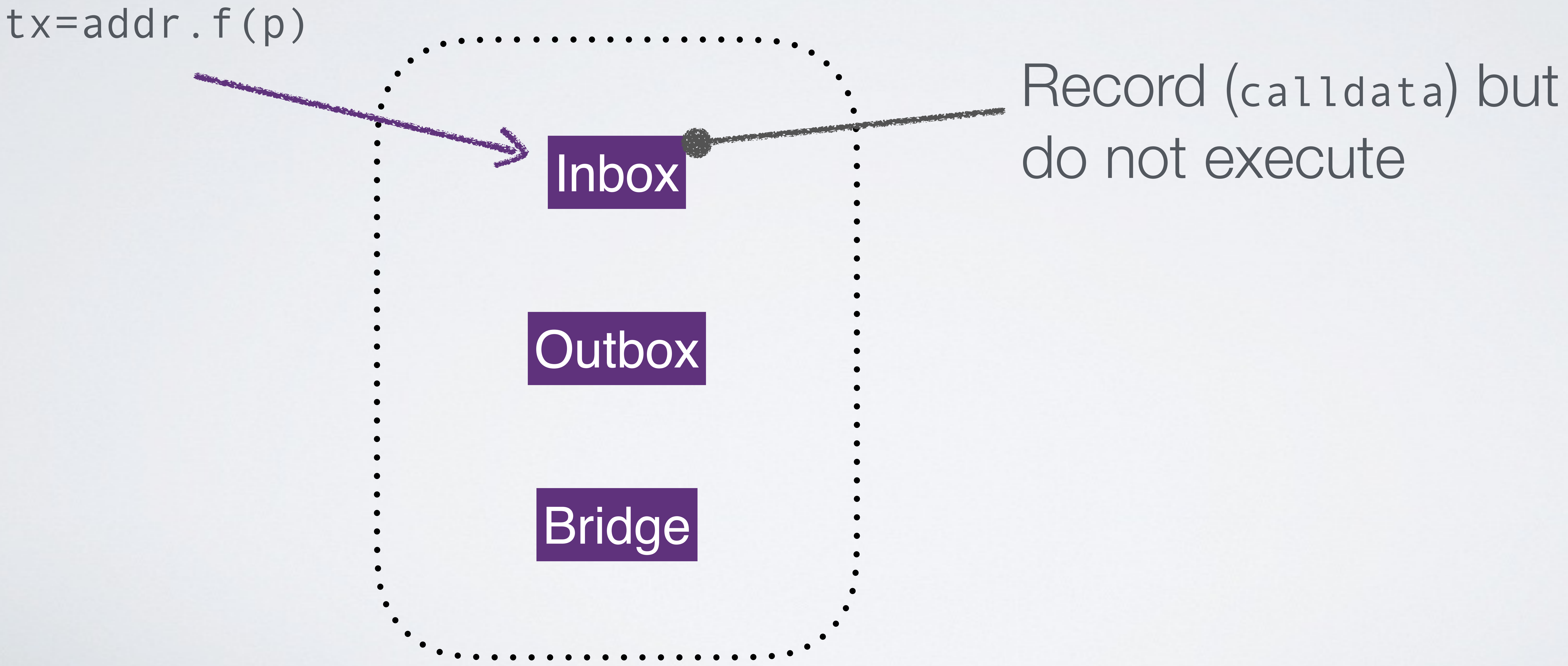
Off-Chain (L2): ArbOS



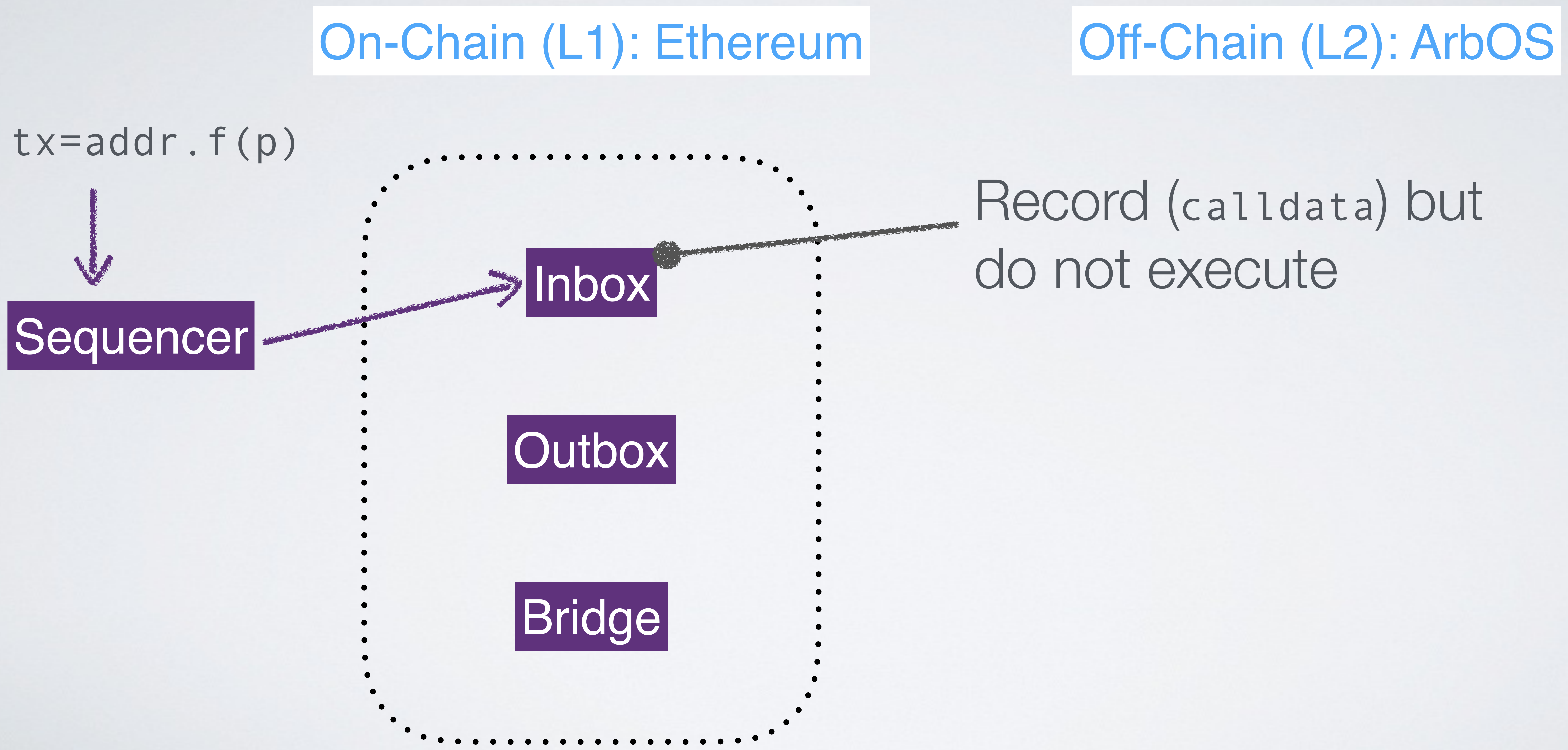
Optimistic Rollup

On-Chain (L1): Ethereum

Off-Chain (L2): ArbOS



Optimistic Rollup



Optimistic Rollup

On-Chain (L1): Ethereum

Off-Chain (L2): ArbOS

`tx=addr.f(p)`

Sequencer

Inbox

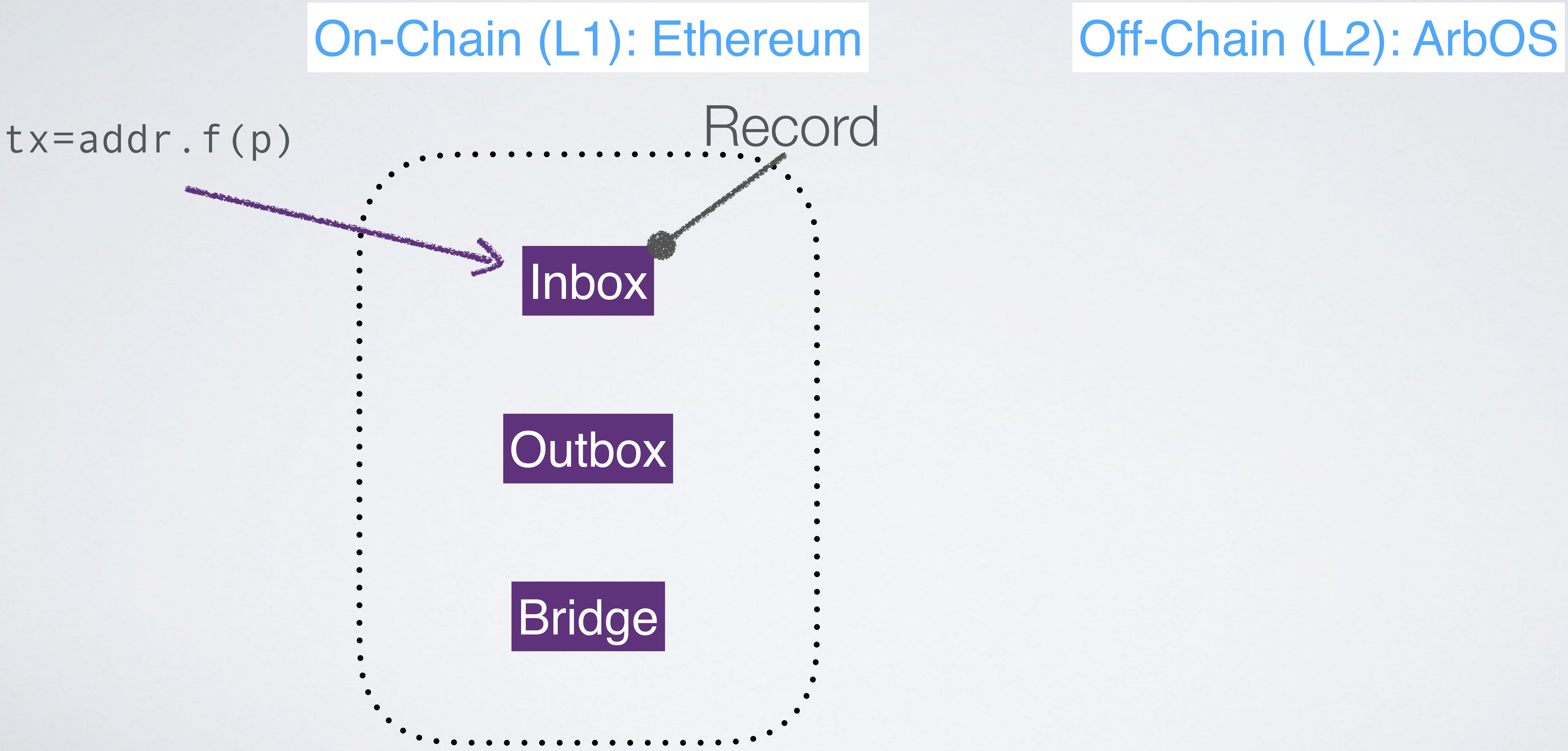
Outbox

Bridge

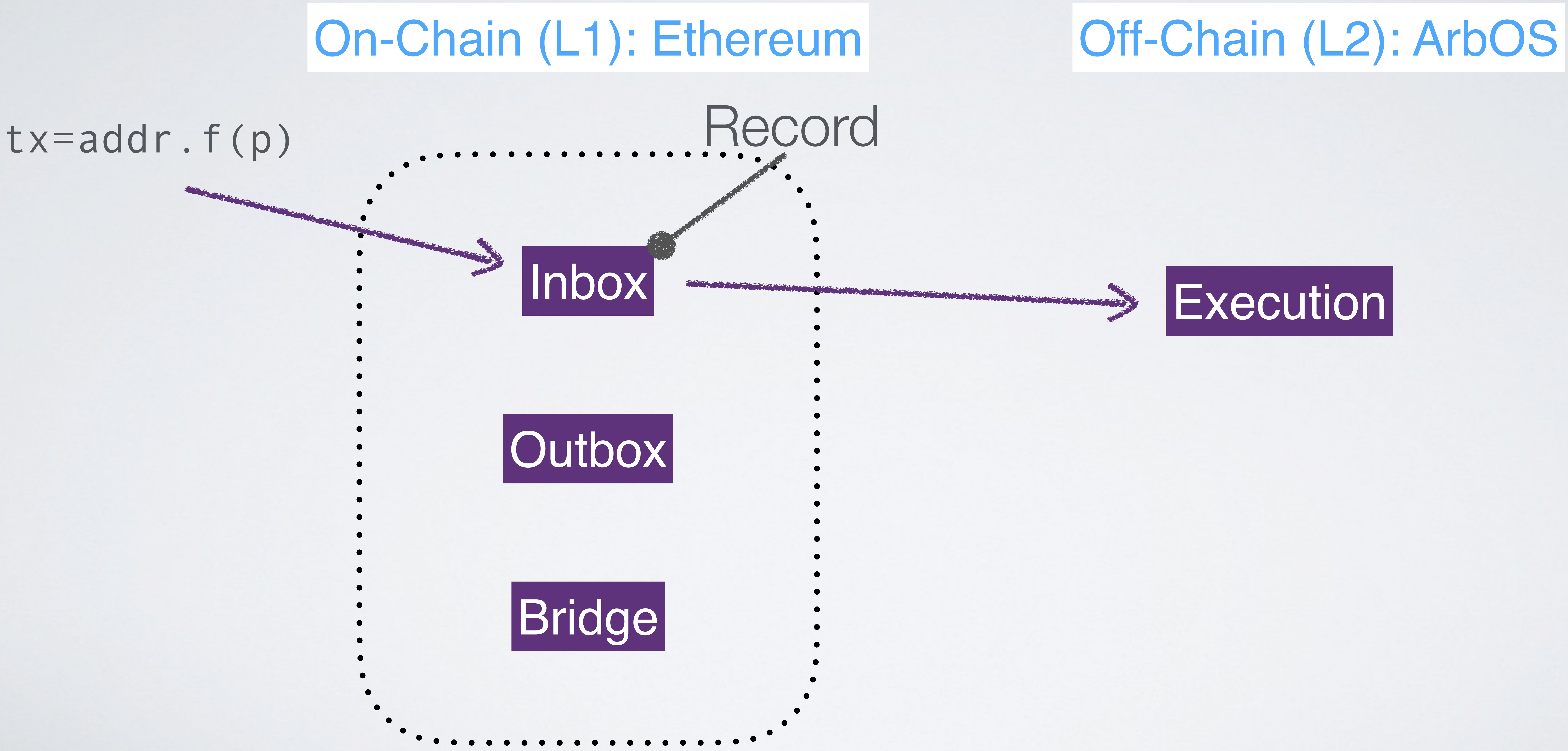
Record (`calldata`) but do not execute

Gasless
Pack
Amortize gas
MEV

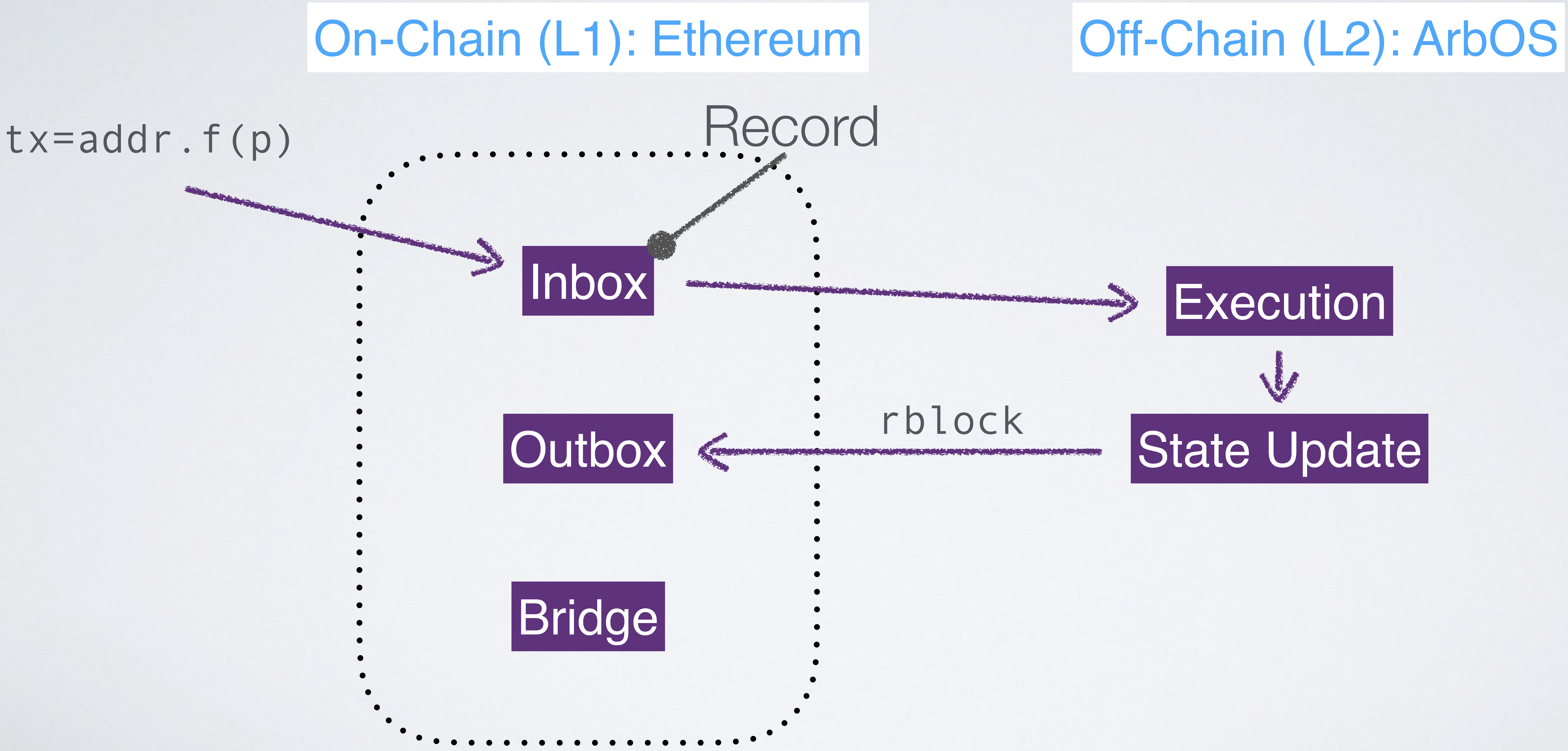
Optimistic Rollup



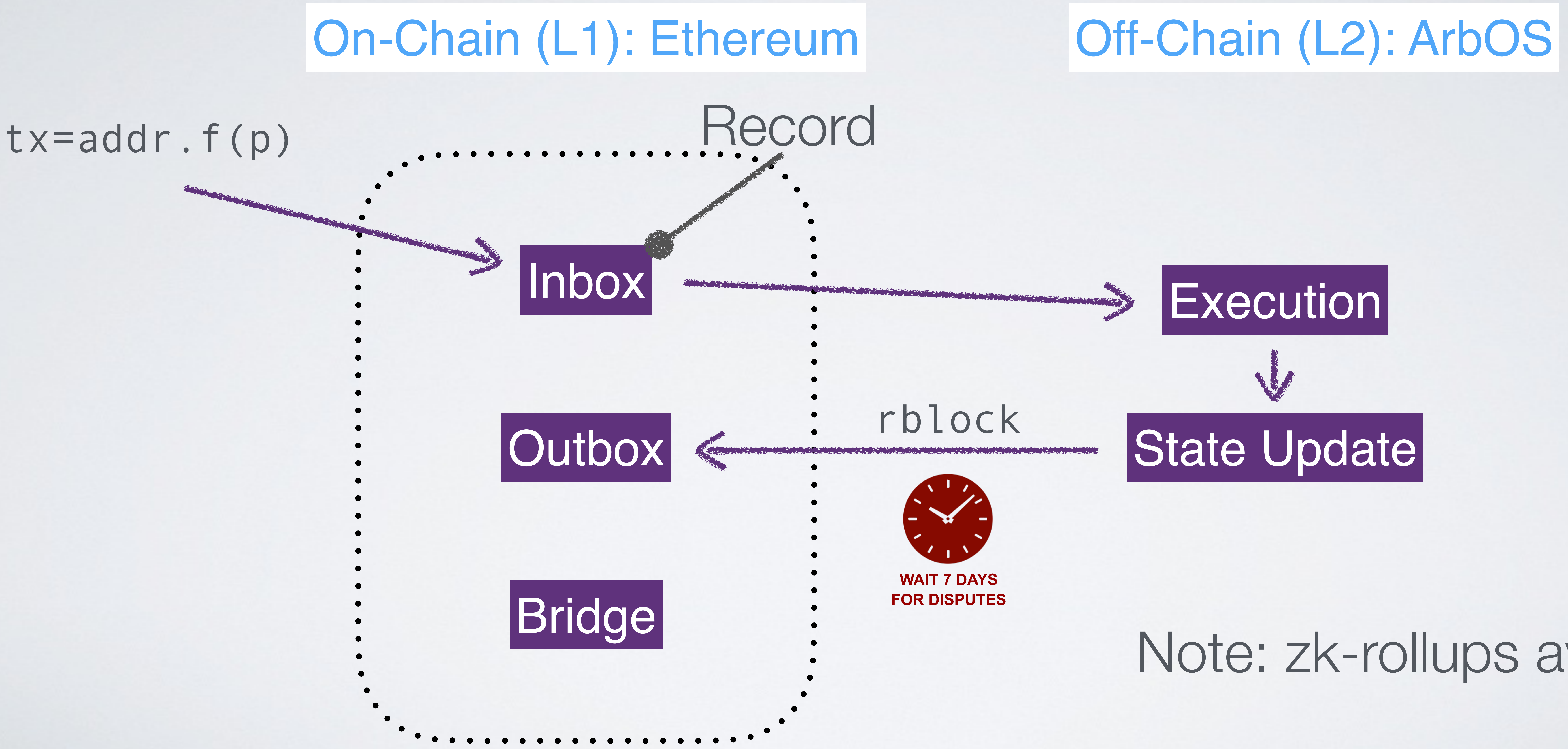
Optimistic Rollup



Optimistic Rollup

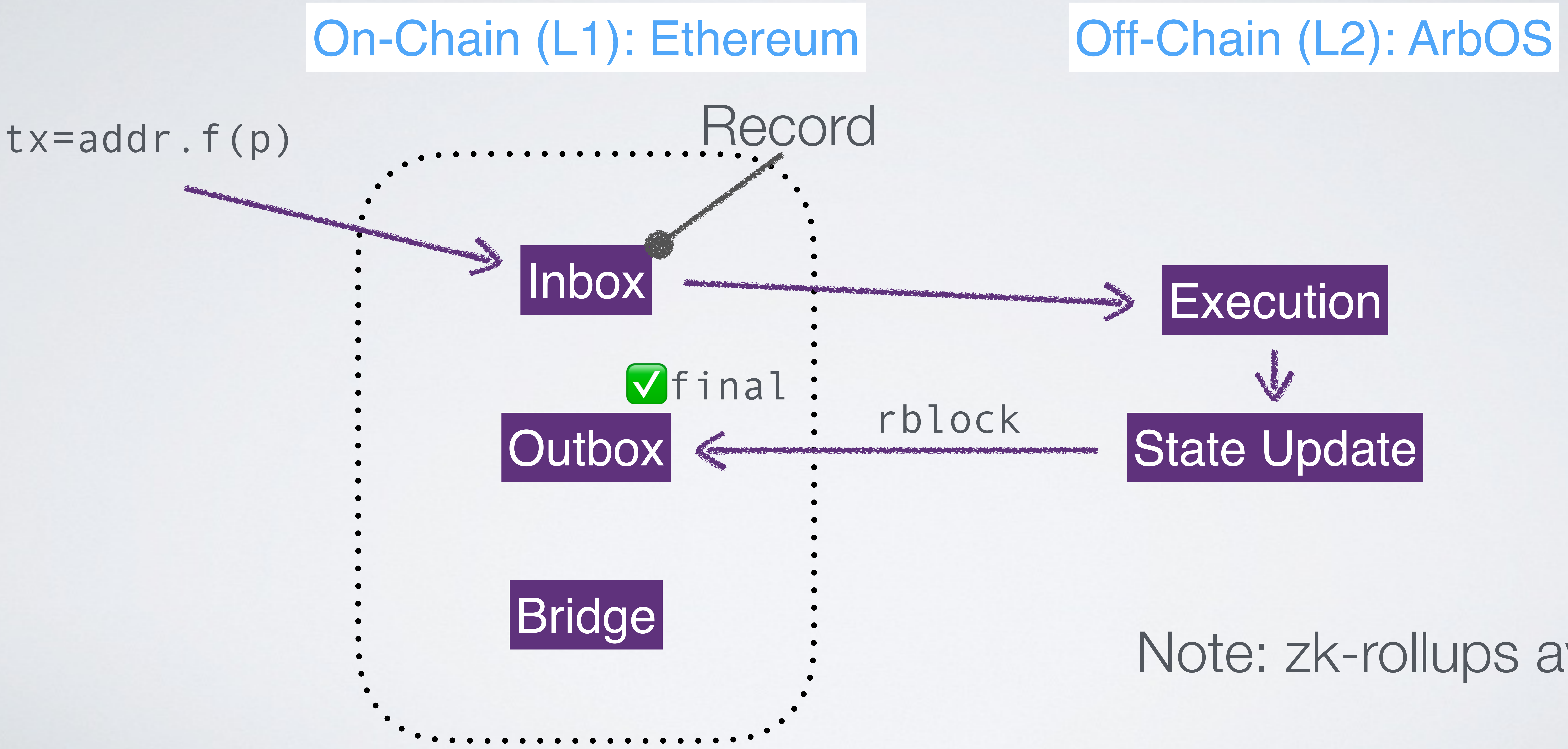


Optimistic Rollup



Note: zk-rollups avoid

Optimistic Rollup



Note: zk-rollups avoid

Bridge: Deposit

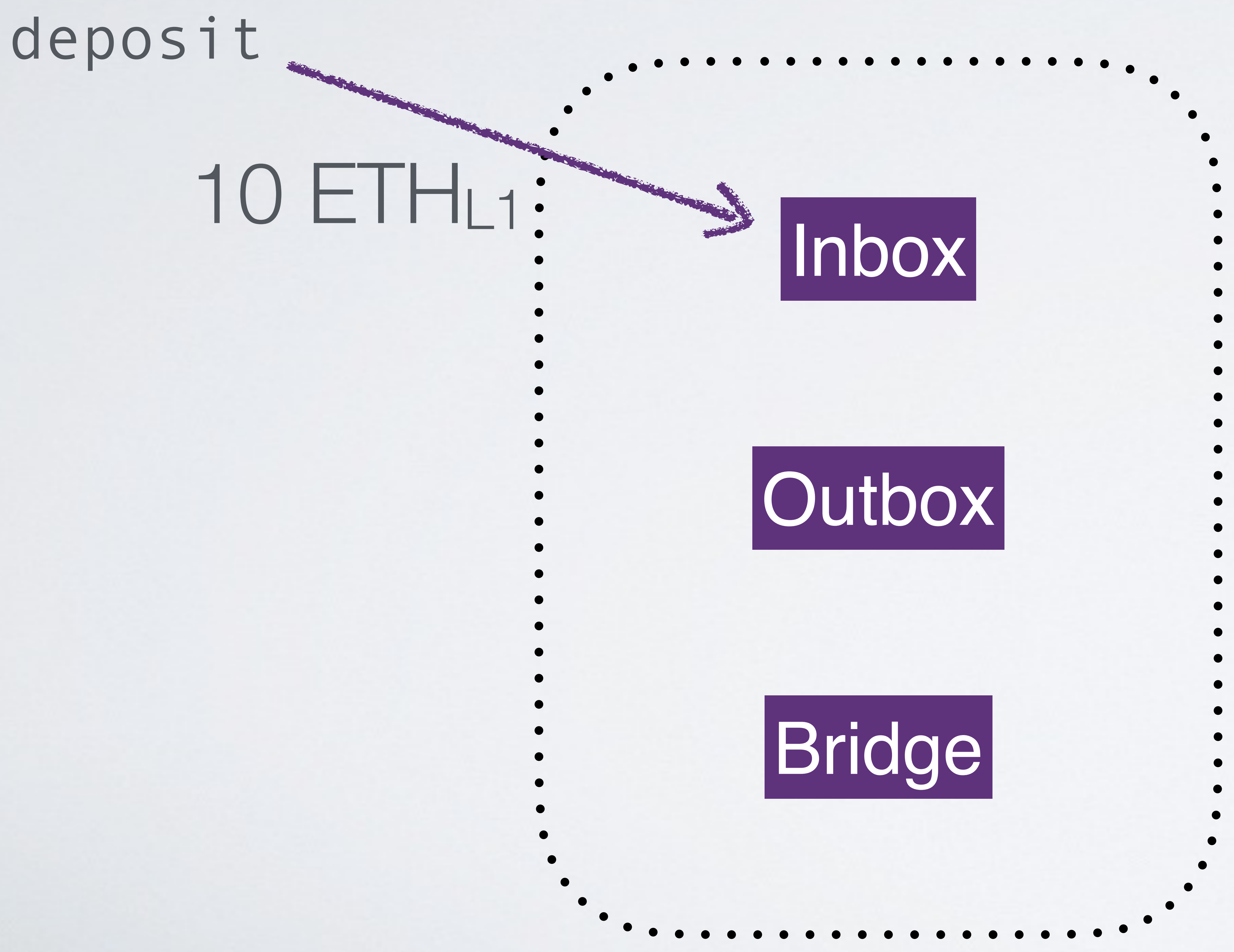
On-Chain (L1): Ethereum

Off-Chain (L2): ArbOS

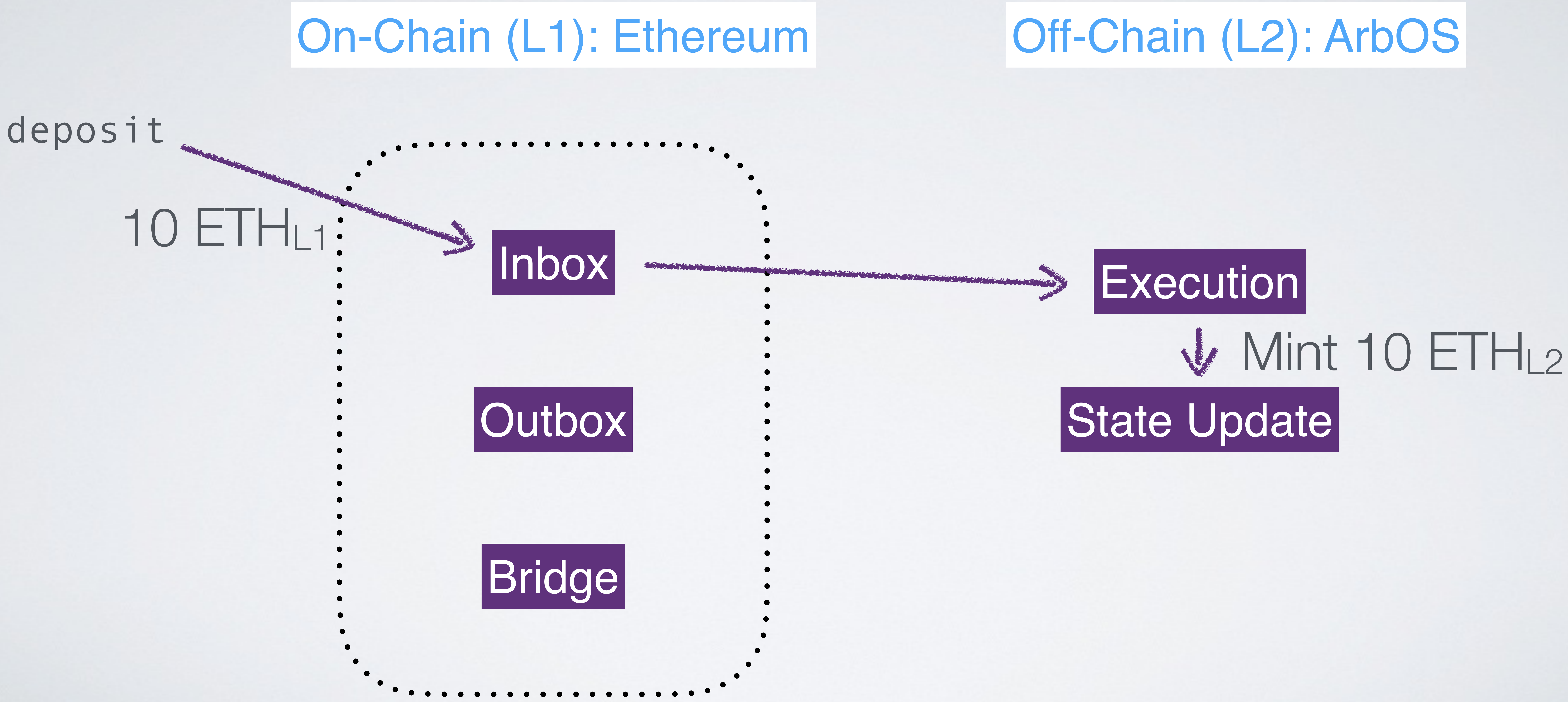
Bridge: Deposit

On-Chain (L1): Ethereum

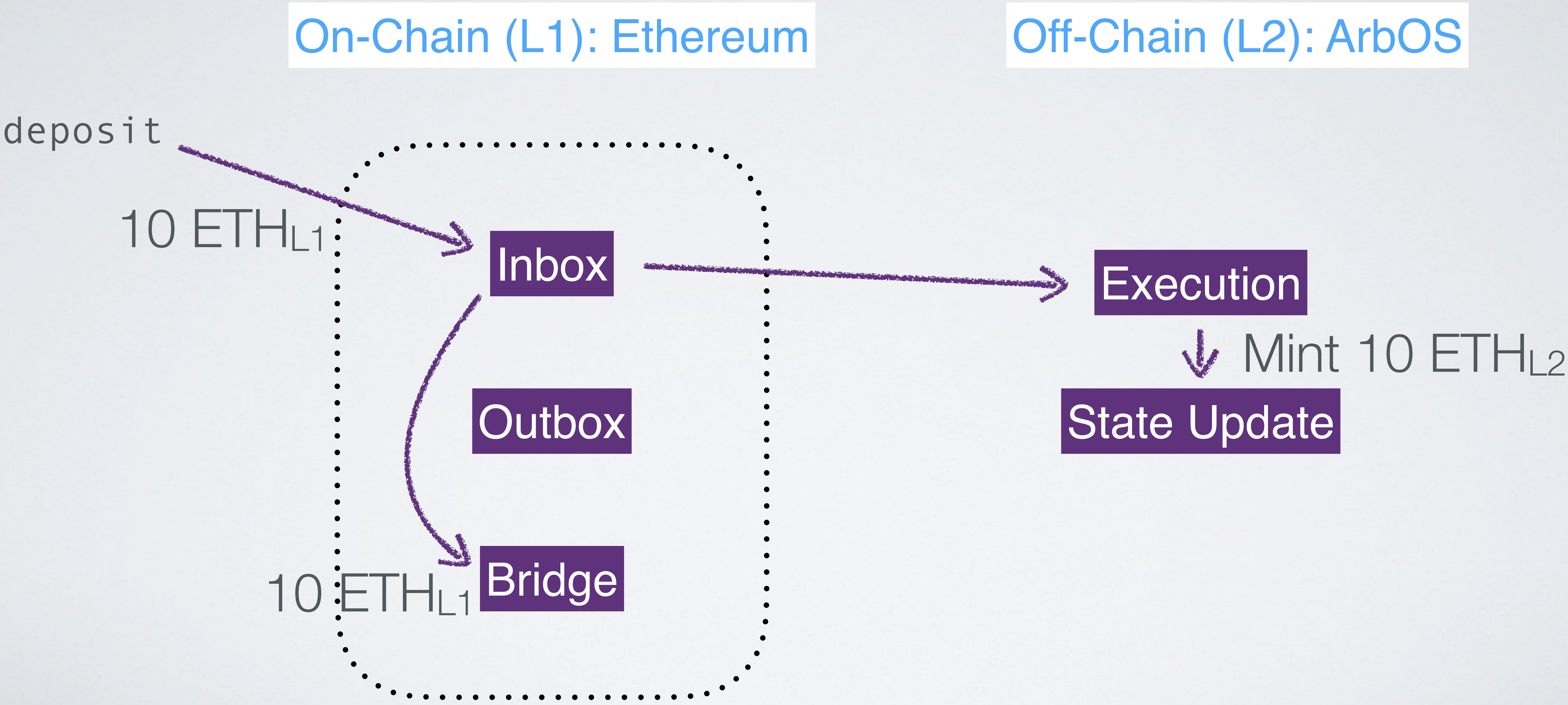
Off-Chain (L2): ArbOS



Bridge: Deposit



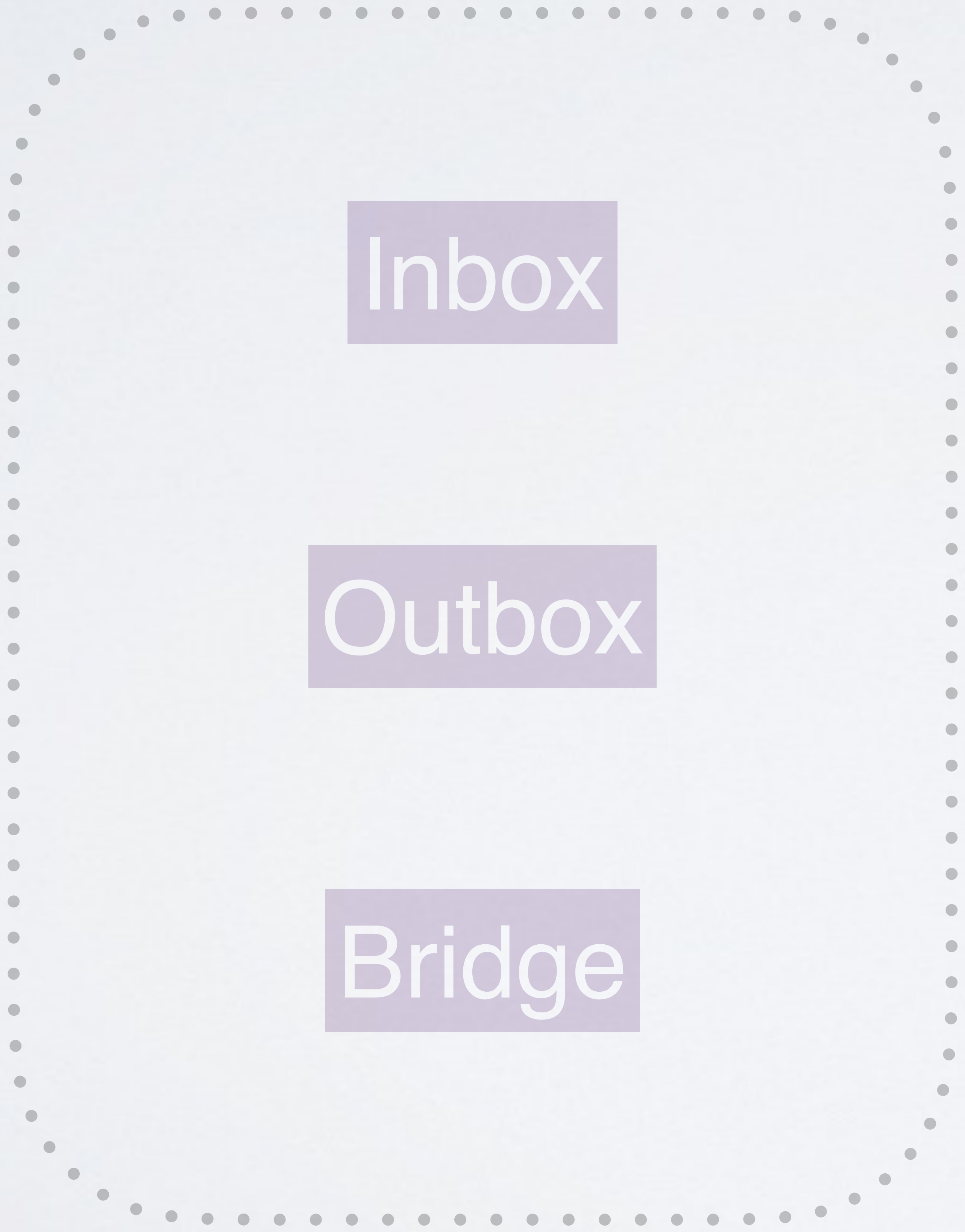
Bridge: Deposit



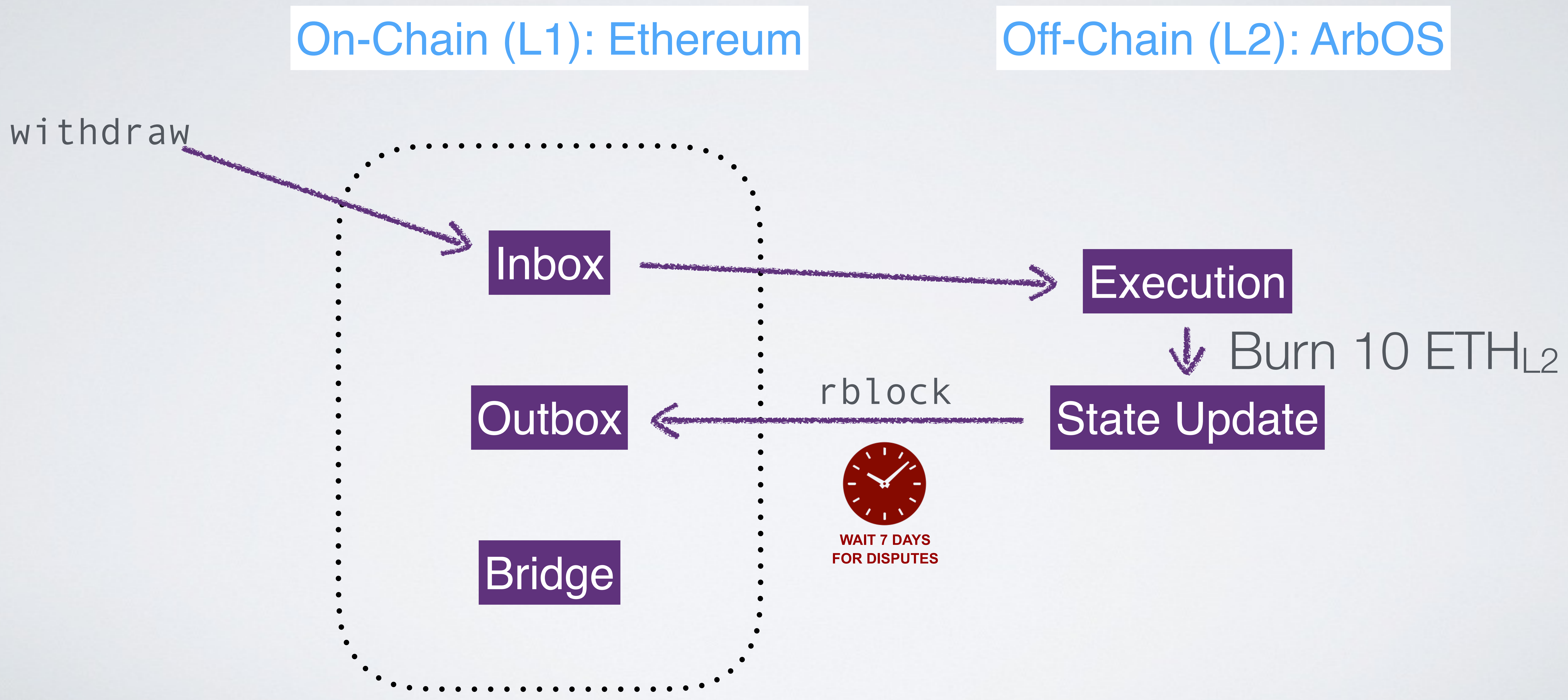
Bridge: Withdraw

On-Chain (L1): Ethereum

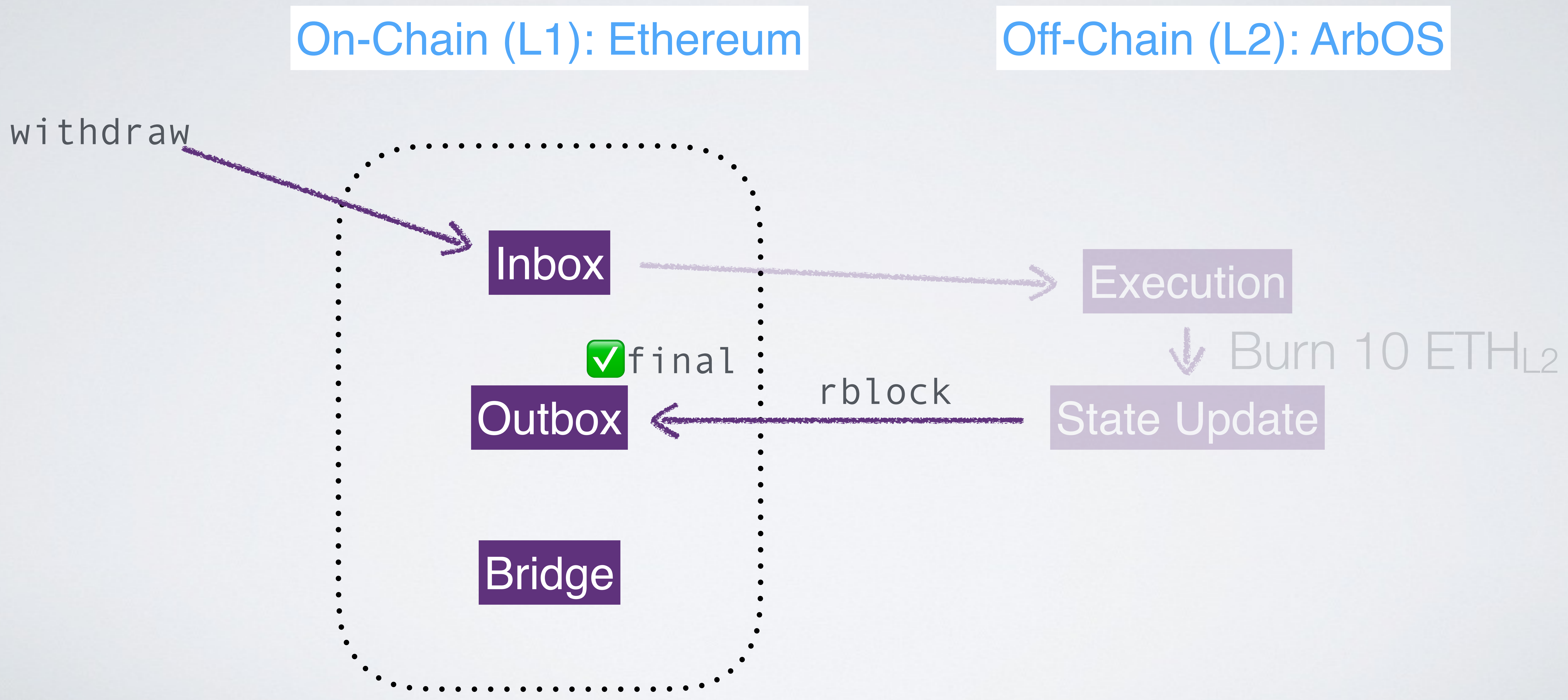
Off-Chain (L2): ArbOS



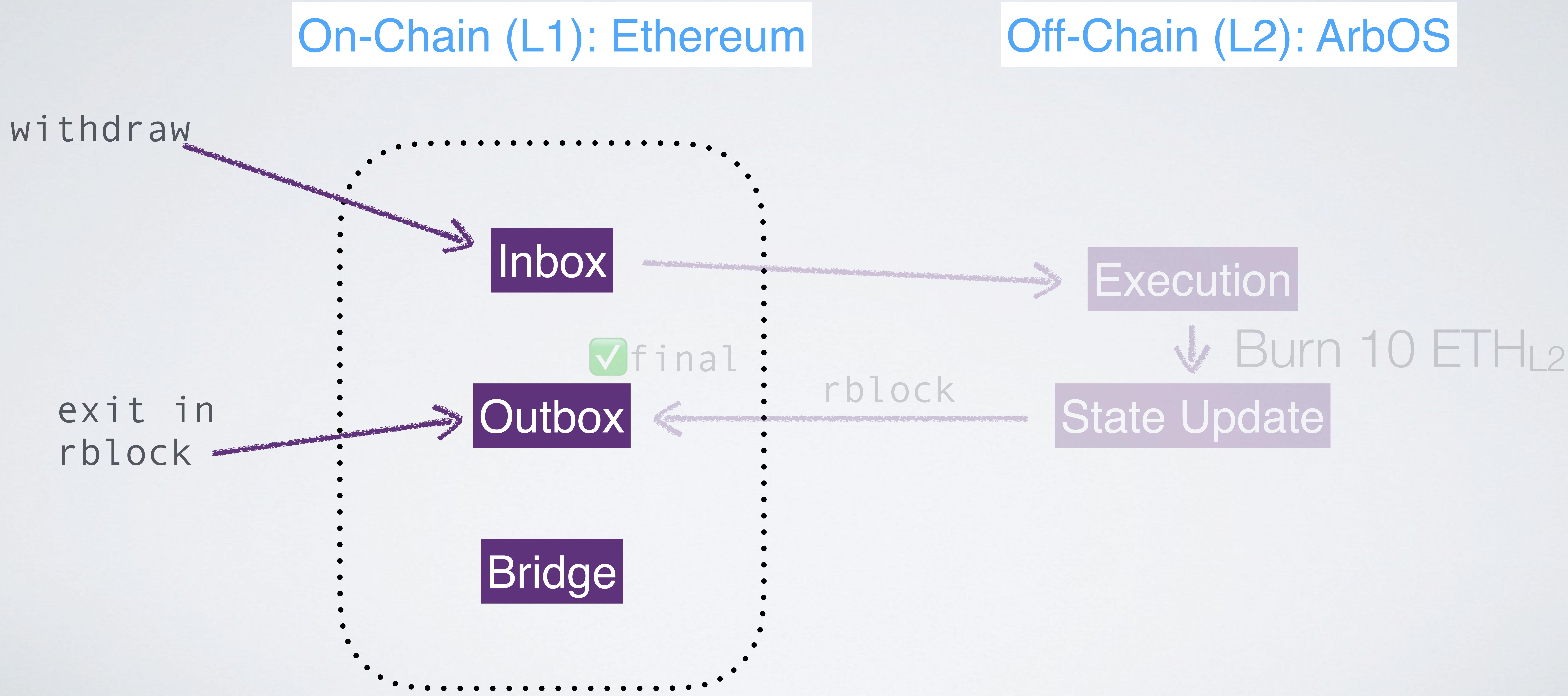
Bridge: Withdraw



Bridge: Withdraw



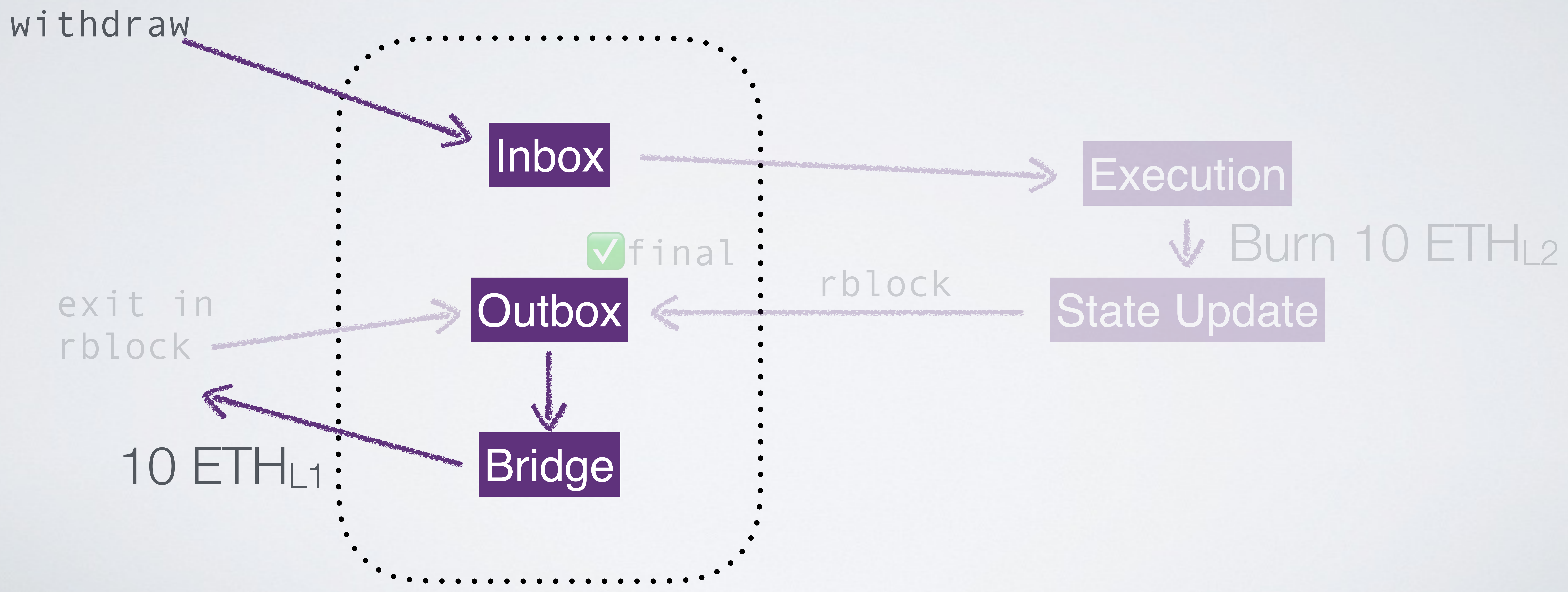
Bridge: Withdraw



Bridge: Withdraw

On-Chain (L1): Ethereum

Off-Chain (L2): ArbOS



Bridge: Withdraw

On-Chain (L1): Ethereum

Off-Chain (L2): ArbOS

Works for any ERC20 token

The problem

- Withdrawals take 7 days (at least… disputes could make longer)
  - 0 disputes yet
- Shorten 7 days but it is complicated -> hard to see it being minutes / hours

The problem

- Withdrawals take 7 days (at least... disputes could make longer)
  - 0 disputes yet
- Shorten 7 days but it is complicated -> hard to see it being minutes / hours

- I just sent a transaction, will it be finalized?

The problem

- Withdrawals take 7 days (at least… disputes could make longer)
  - 0 disputes yet
- Shorten 7 days but it is complicated -> hard to see it being minutes / hours

- I just sent a transaction, will it be finalized?
  - In layer 1 mempool: maybe, maybe not

The problem

- Withdrawals take 7 days (at least… disputes could make longer)
  - 0 disputes yet
- Shorten 7 days but it is complicated -> hard to see it being minutes / hours

- I just sent a transaction, will it be finalized?
  - In layer 1 mempool: maybe, maybe not
  - In layer 2 rblock: if rblock is valid, it must finalize (already sequenced in inbox)
    - “Eventual finality”

The problem

- Withdrawals take 7 days (at least… disputes could make longer)
  - 0 disputes yet
- Shorten 7 days but it is complicated -> hard to see it being minutes / hours

- I just sent a transaction, will it be finalized?
  - In layer 1 mempool: maybe, maybe not
  - In layer 2 rblock: if rblock is valid, it must finalize (already sequenced in inbox)
    - “Eventual finality”

- My exit is sitting in an rblock, it *will* come out in 7 days, maybe I can sell it?
  - “Tradeable exits”

Some other solutions

- **Any exchange: $ETH_{L1} \leftrightarrow ETH_{L2}$**

Some other solutions

- **Any exchange: $ETH_{L1} \leftrightarrow ETH_{L2}$**

Type	*Example*	No trusted third party	Within an L1 transaction	Within an L2 rollup	No griefing	No free option	Opt-in anytime	L2-to-L2
Normal Exit (baseline)	Arbitrum	●			●	●		
Centralized	Coinbase		●	●	●	●		●
HTLC Swaps	Celer	●	○	●				●
Conditional Transfers	StarkEx	●	●	●				
Bridge Tokens	Hop	○	●	●		●		●
Tradeable Exits	This Work	●	∼	●	●	●	●	
Hedged Tradeable Exits	This Work	●	∼	●	●	●	●	

Table 1 Comparing alternatives for fast withdrawals from optimistic rollups for liquid and fungible tokens where ● satisfies the property fully, ○ partially satisfies the property, and no dot means the property is not satisfied. For our work, ∼ means we propose how to fully achieve the property but do not by default (see caveats in Section 6.1).

Some other solutions

- **Any exchange: $ETH_{L1} \leftrightarrow ETH_{L2}$**

Type	*Example*	No trusted third party	Within an L1 transaction	Within an L2 rollup	No griefing	No free option	Opt-in anytime	L2-to-L2
Normal Exit (baseline)	Arbitrum	•			•	•		
Centralized	Coinbase		•	•	•	•		•
HTLC Swaps	Celer	•	○	•				•
Conditional Transfers	StarkEx	•	•	•				
Bridge Tokens	Hop	○	•	•		•		•
Tradeable Exits	This Work	•	∼	•	•	•	•	
Hedged Tradeable Exits	This Work	•	∼	•	•	•	•	

Table 1 Comparing alternatives for fast withdrawals from optimistic rollups for liquid and fungible tokens where • satisfies the property fully, ○ partially satisfies the property, and no dot means the property is not satisfied. For our work, ∼ means we propose how to fully achieve the property but do not by default (see caveats in Section 6.1).

Decentralized

Some other solutions

- **Any exchange: $ETH_{L1} \leftrightarrow ETH_{L2}$**

Type	*Example*	No trusted third party	Within an L1 transaction	Within an L2 rollup	No griefing	No free option	Opt-in anytime	L2-to-L2
Normal Exit (baseline)	Arbitrum	•			•	•		
Centralized	Coinbase		•	•	•	•		•
HTLC Swaps	Celer	•	○	•				•
Conditional Transfers	StarkEx	•	•	•				
Bridge Tokens	Hop	○	•	•		•		•
Tradeable Exits	This Work	•	∼	•	•	•	•	
Hedged Tradeable Exits	This Work	•	∼	•	•	•	•	

Table 1 Comparing alternatives for fast withdrawals from optimistic rollups for liquid and fungible tokens where • satisfies the property fully, ○ partially satisfies the property, and no dot means the property is not satisfied. For our work, ∼ means we propose how to fully achieve the property but do not by default (see caveats in Section 6.1).

Fast

Some other solutions

- **Any exchange: $ETH_{L1} \leftrightarrow ETH_{L2}$**

Type	*Example*	No trusted third party	Within an L1 transaction	Within an L2 rollup	No griefing	No free option	Opt-in anytime	L2-to-L2
Normal Exit (baseline)	Arbitrum	•			•	•		
Centralized	Coinbase		•	•	•	•		•
HTLC Swaps	Celer	•	○	•				•
Conditional Transfers	StarkEx	•	•	•				
Bridge Tokens	Hop	○	•	•		•		•
Tradeable Exits	This Work	•	∼	•	•	•	•	
Hedged Tradeable Exits	This Work	•	∼	•	•	•	•	

Table 1 Comparing alternatives for fast withdrawals from optimistic rollups for liquid and fungible tokens where • satisfies the property fully, ○ partially satisfies the property, and no dot means the property is not satisfied. For our work, ∼ means we propose how to fully achieve the property but do not by default (see caveats in Section 6.1).

Atomic swap bug:
One party can selectively delay or abort

Some other solutions

- **Any exchange: $ETH_{L1} \leftrightarrow ETH_{L2}$**

Type	*Example*	No trusted third party	Within an L1 transaction	Within an L2 rollup	No griefing	No free option	Opt-in anytime	L2-to-L2
Normal Exit (baseline)	Arbitrum	•			•	•		
Centralized	Coinbase		•	•	•	•		•
HTLC Swaps	Celer	•	○	•				•
Conditional Transfers	StarkEx	•	•	•				
Bridge Tokens	Hop	○	•	•		•		•
Tradeable Exits	This Work	•	∼	•	•	•	•	
Hedged Tradeable Exits	This Work	•	∼	•	•	•	•	

Table 1 Comparing alternatives for fast withdrawals from optimistic rollups for liquid and fungible tokens where • satisfies the property fully, ○ partially satisfies the property, and no dot means the property is not satisfied. For our work, ∼ means we propose how to fully achieve the property but do not by default (see caveats in Section 6.1).

Oops, you already withdrew

Some other solutions

- **Any exchange: $ETH_{L1} \leftrightarrow ETH_{L2}$**

Type	*Example*	No trusted third party	Within an L1 transaction	Within an L2 rollup	No griefing	No free option	Opt-in anytime	L2-to-L2
Normal Exit (baseline)	Arbitrum	•			•	•		
Centralized	Coinbase		•	•	•	•		•
HTLC Swaps	Celer	•	○	•				•
Conditional Transfers	StarkEx	•	•	•				
Bridge Tokens	Hop	○	•	•		•		•
Tradeable Exits	This Work	•	∼	•	•	•	•	
Hedged Tradeable Exits	This Work	•	∼	•	•	•	•	

Table 1 Comparing alternatives for fast withdrawals from optimistic rollups for liquid and fungible tokens where • satisfies the property fully, ○ partially satisfies the property, and no dot means the property is not satisfied. For our work, ∼ means we propose how to fully achieve the property but do not by default (see caveats in Section 6.1).

We are not trying to compete w/ Hop

Tradeable exits

- Alice has an exit for 10 ETH_{L2} that is pending for 7 days
  - Give Alice a claim as a transferable token: 10 ETH_{xx}
- Bob has 10 ETH_{L1}
- Alice and Bob swap

Tradeable exits

- Alice has an exit for 10 ETH_{L2} that is pending for 7 days
  - Give Alice a claim as a transferable token: 10 ETH_{xx}
- Bob has 10 ETH_{L1}
- Alice and Bob swap

- Remaining problems:
  - Bob needs to be an Arbitrum validator to believe the 10 ETH_{xx} will finalize
  - 10 ETH_{L2} > 10 ETH_{xx}
    - Price ETH_{xx}?
  - Implementation

Tradeable exits

- **Bob needs to be an Arbitrum validator to believe the 10 ETH_{xx} will finalize**
  - **Insurance: covers withdraw amount if rblock does not finalize**
  - **Insurance is a safe bet for any Arbitrum validator (eventual finality)**
  - **Set up as a simple prediction market**

le exits

- Bob needs to be an Arb ... ill finalize
  - Insurance: covers wi ...
  - Insurance is a safe be ... ality)
  - Set up as a simple pre ...

On Decentralizing Prediction Markets and Order Books

Jeremy Clark[1], Joseph Bonneau[2], Edward W. Felten[2], Joshua A. Kroll[2], Andrew Miller[3], and Arvind Narayanan[2]

[1] Concordia University
[2] Princeton University
[3] University of Maryland

Abstract. We propose techniques for decentralizing prediction markets and order books, utilizing Bitcoin's security model and consensus mechanism. Decentralization of prediction markets offers several key advantages over a centralized market: no single entity governs over the market, all transactions are transparent in the block chain, and anybody can participate pseudonymously to either open a new market or place bets in an existing one. We provide trust agility: each market has its own specified arbiter and users can choose to interact in markets that rely on the arbiters they trust. We also provide a transparent, decentralized order book that enables order execution on the block chain in the presence of potentially malicious miners.

1 Introductory Remarks

Bitcoin has demonstrated that achieving consensus in a decentralized network is practical. This has stimulated research on applying Bitcoin-esque consensus mechanisms to new applications (*e.g.*, DNS through Namecoin,[4] timestamping through CommitCoin [10], and smart contracts through Ethereum[5]). In this paper, we consider application of Bitcoin's principles to prediction markets.

A prediction market (PM) enables forecasts about uncertain future events to be forged into financial instruments that can be traded (bought, sold, shorted, *etc.*) until the uncertainty of the event is resolved. In several common forecasting scenarios, PMs have demonstrated lower error than polls, expert opinions, and statistical inference [2]. Thus an open and transparent PM not only serves its traders, it serves any stakeholder in the outcome by providing useful forecasting information through prices.

Whenever discussing the application of Bitcoin to a new technology or service, its important to distinguish exactly what is meant. For example, a "Bitcoin-based prediction market" could mean at least three different things: (1) adding Bitcoin-related contracts (*e.g.*, the future Bitcoin/USD exchange rate) to a traditional centralized PM, (2) converting the underlying currency of a centralized prediction market to Bitcoin, or (3) applying the design principles of Bitcoin to decentralize the functionality and governance of a PM.

Of the three interpretations, approach (1) is not a research contribution. Approach (2) inherits most of the properties of a traditional PM: Opening markets for new future events is subject to a commitment by the PM host to determine the outcome, virtually any trading rules can be implemented, and trade settlement and clearing can be automated if money is held in trading accounts. In addition, by denominating the PM in Bitcoin, approach (2) enables easy electronic deposits and withdrawals from trading accounts, and can add a level of anonymity. An example of approach (2) is Predictious.[6]

This set of properties is a desirable starting point but we see several ways it can be improved through approach (3). Thus, our contribution is a novel PM design that enables:

- **A Decentralized Clearing/Settlement Service.** Fully automated settlement and clearing of trades without escrowing funds to a trusted straight through processor (STP).
- **A Decentralized Order Matching Service.** Fully automated matching of orders in a built-in call market, plus full support for external centralized exchanges.

[4] http://namecoin.info
[5] http://www.ethereum.org
[6] https://www.predictious.com

WEIS 2014

Tradeable exits

- Bob needs to be an Arbitrum validator to believe the 10 ETH_{XX} will finalize
  - Insurance: covers withdraw amount if rblock does not finalize
  - Insurance is a safe bet for any Arbitrum validator (eventual finality)
  - Set up as a simple prediction market

- Trader deposits 10 ETH and is given two kinds of shares:
  - 10 $FAIL_{PM}$: redeemable for 10 ETH if rblock fails
  - 10 $FINAL_{PM}$: redeemable for 10 ETH if rblock succeeds

Tradeable exits

- Bob needs to be an Arbitrum validator to believe the 10 ETH_{xx} will finalize
  - Insurance: covers withdraw amount if rblock does not finalize
  - Insurance is a safe bet for any Arbitrum validator (eventual finality)
  - Set up as a simple prediction market

- Trader deposits 10 ETH and is given two kinds of shares:
  - 10 $FAIL_{PM}$: redeemable for 10 ETH if rblock fails
  - 10 $FINAL_{PM}$: redeemable for 10 ETH if rblock succeeds
    - Wait!!! Isn’t 10 ETH_{xx} and 10 $FINAL_{PM}$ the same thing?
    - Close for ETH_{xx} and not DAI_{xx} or ARB_{xx} or $TOKEN_{xx}$

Tradeable exits

- Bob needs to be an Arbitrum validator to believe the 10 ETH_{xx} will finalize
  - Insurance: covers withdraw amount if rblock does not finalize
  - Insurance is a safe bet for any Arbitrum validator (eventual finality)
  - Set up as a simple prediction market

- Trader deposits 10 ETH and is given two kinds of shares:
  - 10 $FAIL_{PM}$: redeemable for 10 ETH if rblock fails
  - 10 $FINAL_{PM}$: redeemable for 10 ETH if rblock succeeds
  - Shares can sold independent of each other
  - Shares can be returned as a pair to redeem 10 ETH before market closes
  - No oracle needed: rblock success in on-chain (implementation -> expose it)

Tradeable exits

- 10 ETH_{L1} > 10 ETH_{xx}
  - 10 ETH now > 10 ETH locked for 7 days

Tradeable exits

- 10 ETH_{L1} > 10 ETH_{xx}
  - 10 ETH now > 10 ETH locked for 7 days
- Price ETH_{xx} like a futures contract with one difference: buyer of ETH_{xx} pays today
  - Spot price of ETH_{L1} ✅
  - Time to expiration: ETH_{xx} will approach ETH_{L1} ✅
  - Differences in storage costs between ETH_{xx} and ETH_{L1} ❌
  - Differences in yield/interest between ETH_{xx} and ETH_{L1} ✅
  - Exchange rate risks ❌
  - Delivery cost: gas cost to resolve the exit ✅
  - Settlement risk: probability that rblock does not finalize ✅

$$F_0 = (S_0 + U - D) \cdot e^{(-r+y)\Delta t} \cdot R$$

withdraw window remains: 6 days
rblock fails: 1 in a billion
bad rblock undetected: 1 in a million
validation software wrong: 1 in a million
exit in gas: 0.008 ETH_{L1}
APY on ETH_{L1}: 0.2%
APY on ETH_{XX}: 0%

Pay 99.665 ETH_{L1} for 100 ETH_{XX}

$$F_0 = (S_0 + U - D) \cdot e^{(-r+y)\Delta t} \cdot R$$

withdraw window remains: 6 days
rblock fails: 1 in a billion
bad rblock undetected: 1 in a million
validation software wrong: 1 in a million
exit in gas: 0.008 ETH_{L1}
APY on ETH_{L1}: 0.2%
APY on ETH_{XX}: 0%

Pay 99.665 ETH_{L1} for 100 ETH_{XX}
Pay 0.04186 ETH_{L1} for 0.5 ETH_{XX}

Implementation

- Arbitrum Nitro
  - Outbox: Track ownership of exit and enable transfers
    - Transfers: 48,798 Gwei (first time is 85K)
    - Exit: 91,418 Gwei
  - Outbox: Expose rblock status (pending/confirmed) to make it prediction market friendly
  - Bridge: Send to right address
- Prediction market: in-progress (hook into Gnosis/Augur)
- A bunch of engineering details omitted

`https://github.com/MadibaGroup/nitro/tree/fast-withdrawals`

Summary

- Summary: Alice swaps {10 ETH_{XX} ,10 $FAIL_{PM}$} with Bob for ~10 ETH_{L1}
  - Hedged tradable exit
  - Anyone can receive ETH_{XX}, including a smart contract (no Arbitrum awareness)
  - Works for any token but assumes "insurance" is in ETH
  - Implementable


Q
@PulpSpy
pulpspy.com

Legality and Regulation

- Illicit uses: monitored by law enforcement agency (cybercrimes)
- Taxation: CRA guidelines (capital gain)
- Financial tracking: FINTRAC guidelines (MSB)
- Securities law: AMF guidelines and sandbox
- Accounting standards: No IFRS standards yet (convention: intangible asset)