Blockchain Interoperability

André Augusto, INESC-ID, Técnico Lisboa
Rafael Belchior, INESC-ID, Blockdaemon

DAY 5 - February 14th, 2025

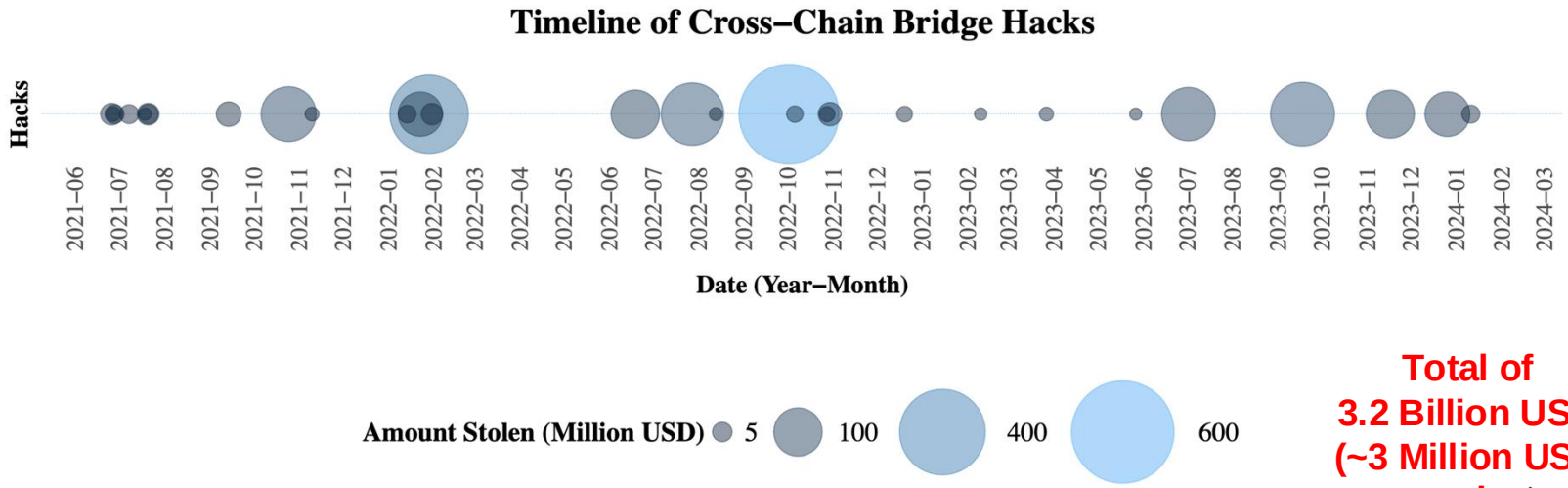
BIG – enhancing the research and innovation potential of Técnico - Lisbon through **B**lockchain technologies and design **I**nnovation for social **G**ood
Grant agreement: 952226

Why should we **study Interoperability Mechanisms** in Blockchain?

+3.2 BILLION USD

Stolen from cross-chain bridges since 2021

A Recurrent Problem…

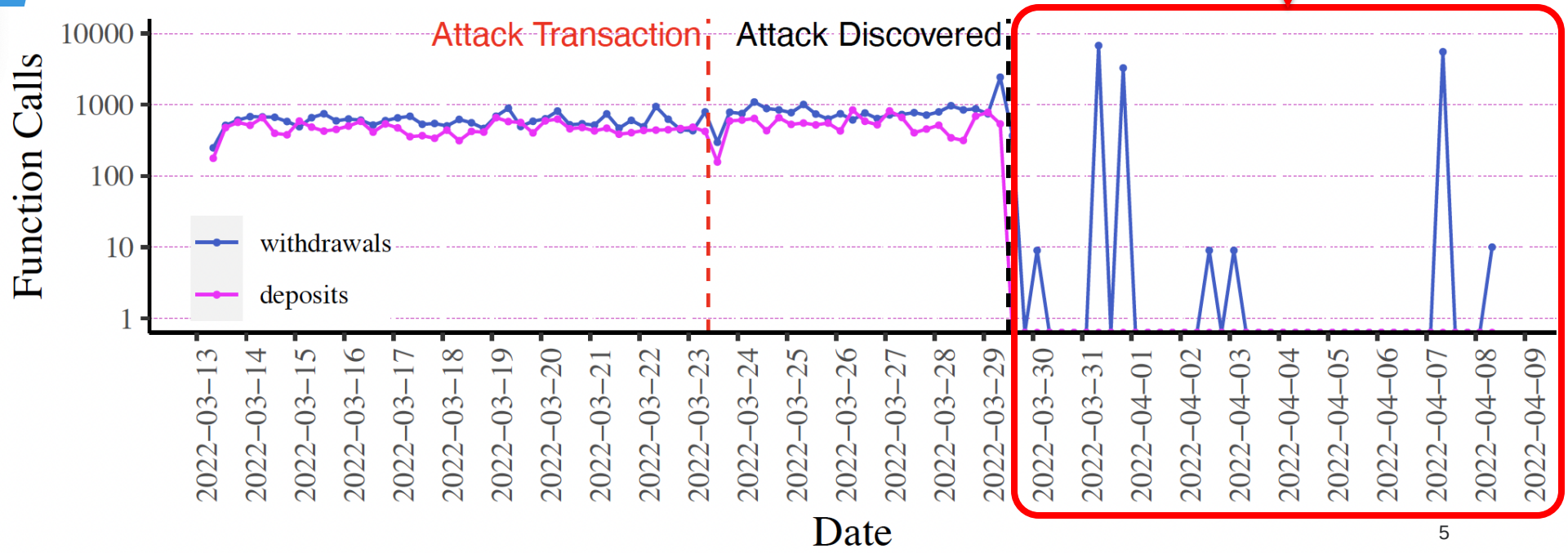


Total of 3.2 Billion USD (~3 Million USD per day)

Augusto, R. Belchior, M. Correia, A. Vasconcelos, L. Zhang and T. Hardjono, "SoK: Security and Privacy of Blockchain Interoperability," 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2024, pp. 3840-3865,

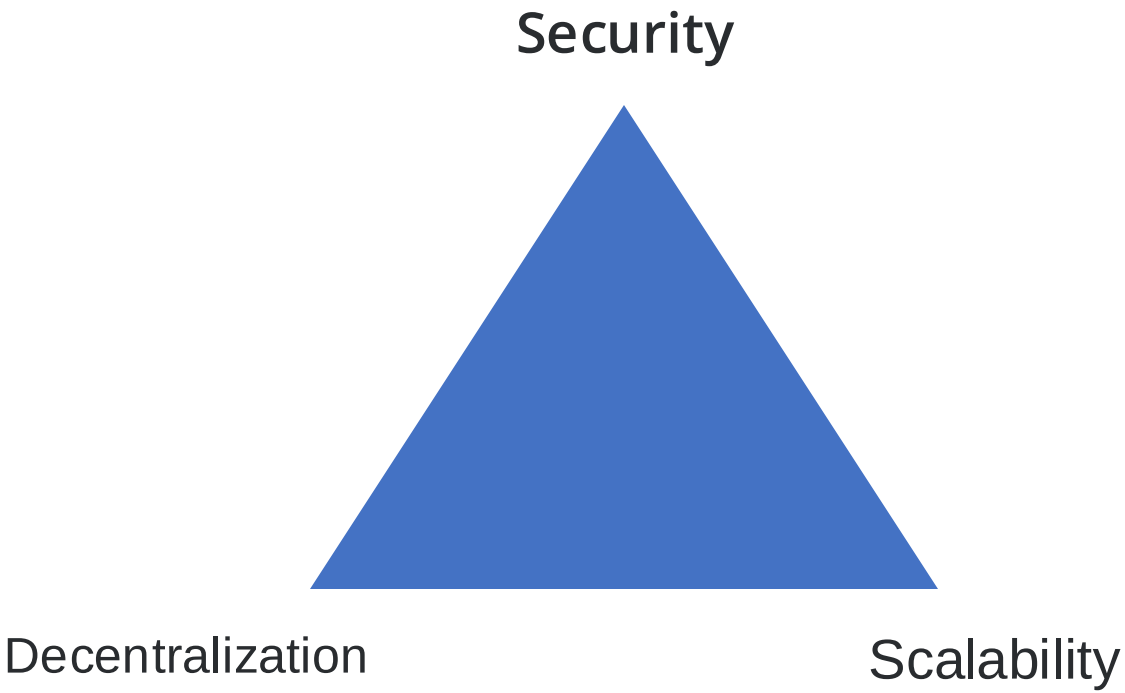
The Consequences

Decreases the flow of funds to other blockchains, and consequently to dApps deployed there



Why is Blockchain Interoperability needed?

The Blockchain Trilemma

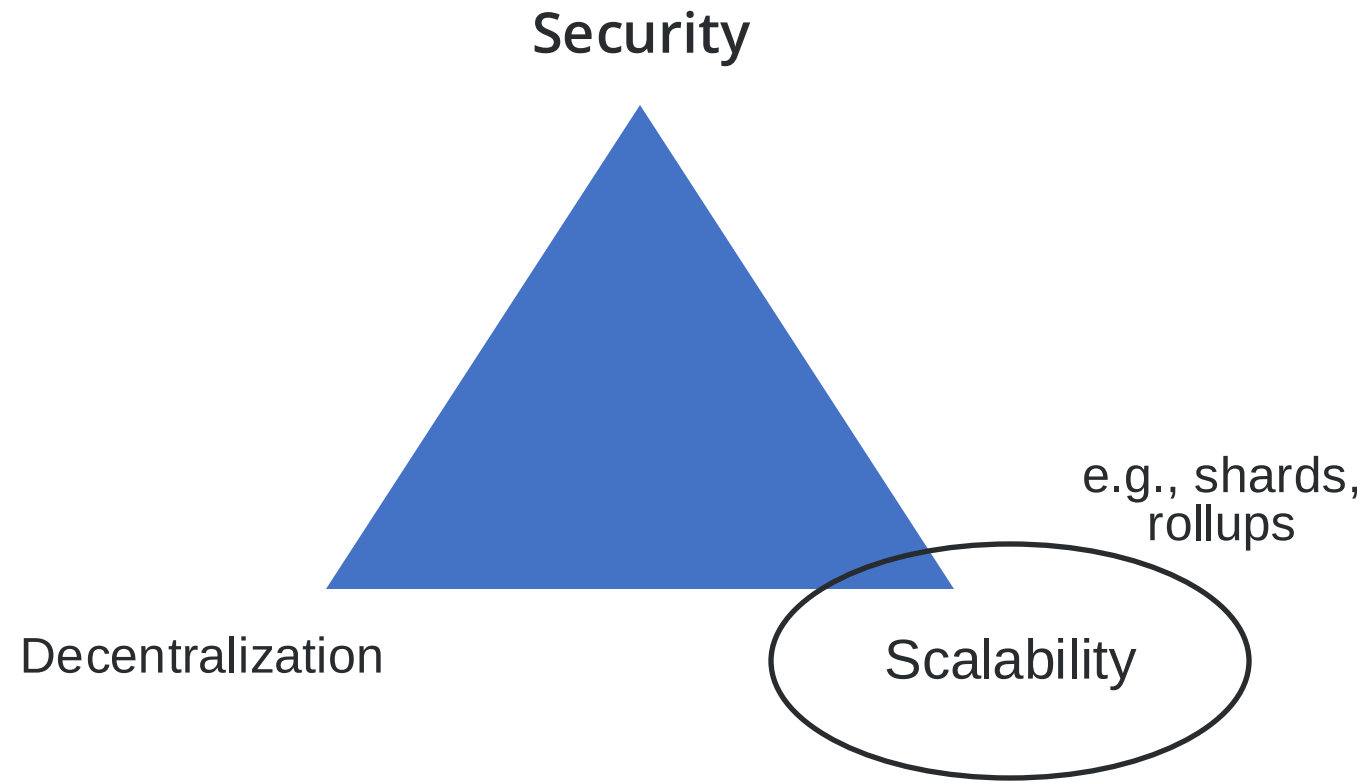


Connect Different Ecosystems

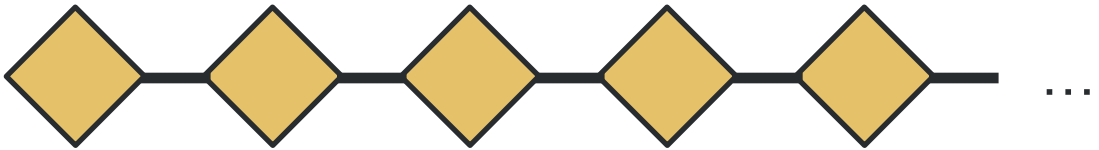
#	Coin		Price	1h	24h	7d	24h Volume	Market Cap
1	Bitcoin BTC	Buy	$68,389.59	0.3%	1.0%	3.3%	$13,663,947,240	$1,347,551,070,887
2	Ethereum ETH	Buy	$3,841.93	0.3%	2.7%	25.1%	$12,470,262,478	$461,028,760,320
4	BNB BNB	Buy	$598.73	0.0%	0.4%	4.4%	$429,266,868	$92,207,772,331
5	Solana SOL	Buy	$162.62	0.5%	2.7%	4.3%	$2,032,605,702	$72,884,782,988
10	Toncoin TON	Buy	$6.32	0.5%	1.1%	0.1%	$124,426,607	$21,956,332,740
11	Cardano ADA	Buy	$0.4578	0.2%	0.4%	2.1%	$200,474,590	$16,169,141,159
12	Avalanche AVAX	Buy	$36.77	0.1%	3.2%	2.5%	$232,265,046	$14,427,629,064

1	Polygon MATIC	Monero XMR
2	Immutable X IMX	Arweave AR
3	Mantle MNT	Sui SUI
4	Stacks STX	Injective INJ
5	ARBITRUM ARB	Fantom FTM
6	Synthetix Network SNX	
7	StarkNet Token STRK	
8	Metis Token METIS	

The Blockchain Trilemma

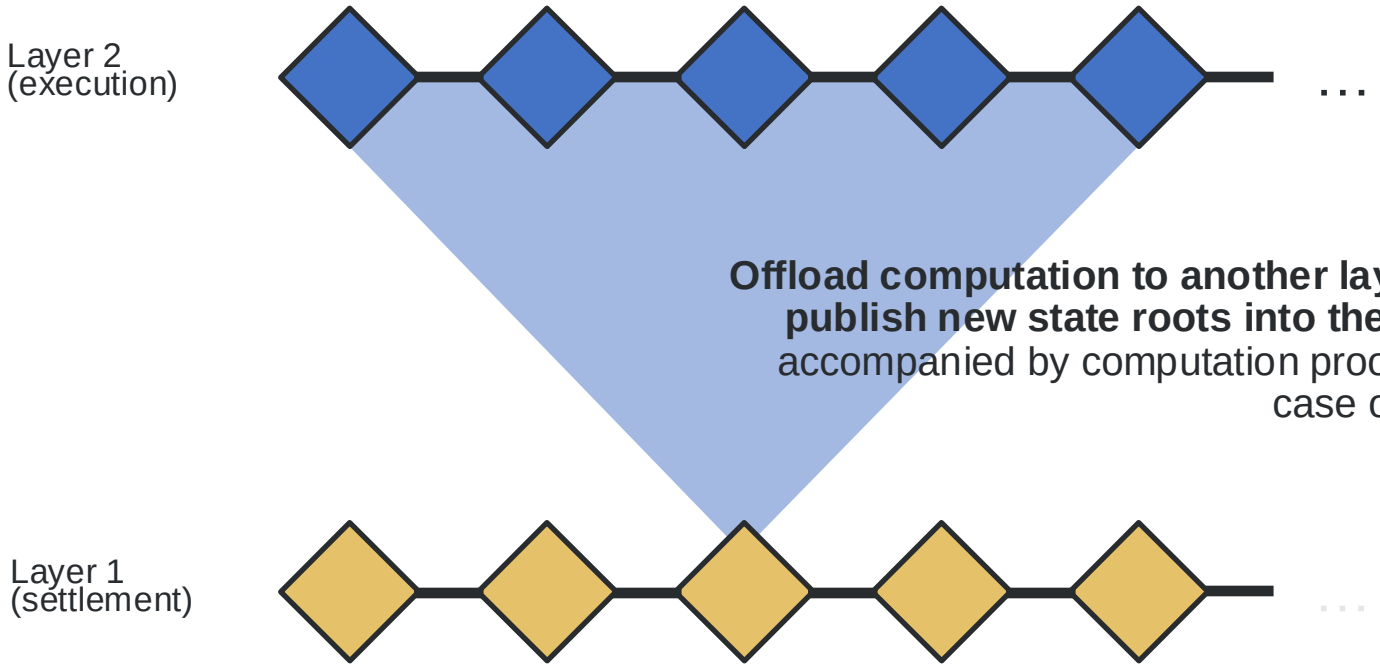


The Scalability Problem of Blockchains



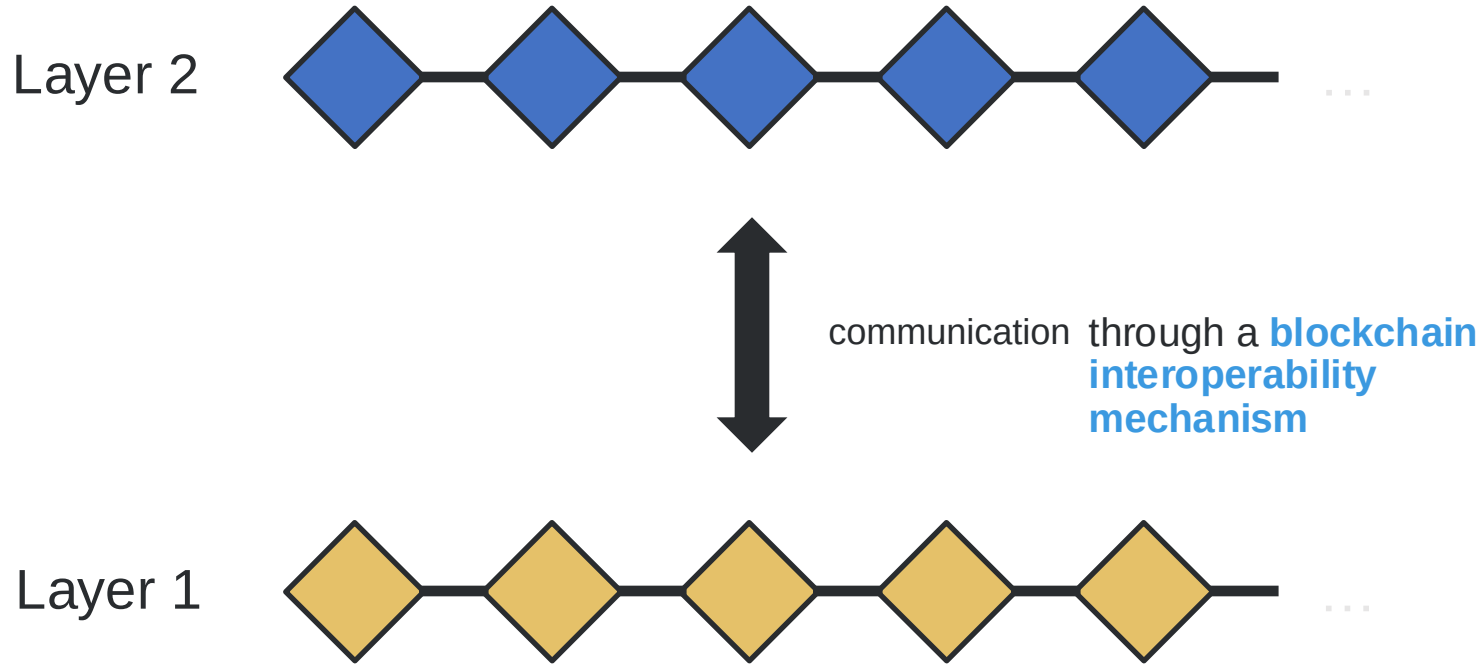
Limited number of transactions in each block
High transaction fees
…

Scaling Blockchains

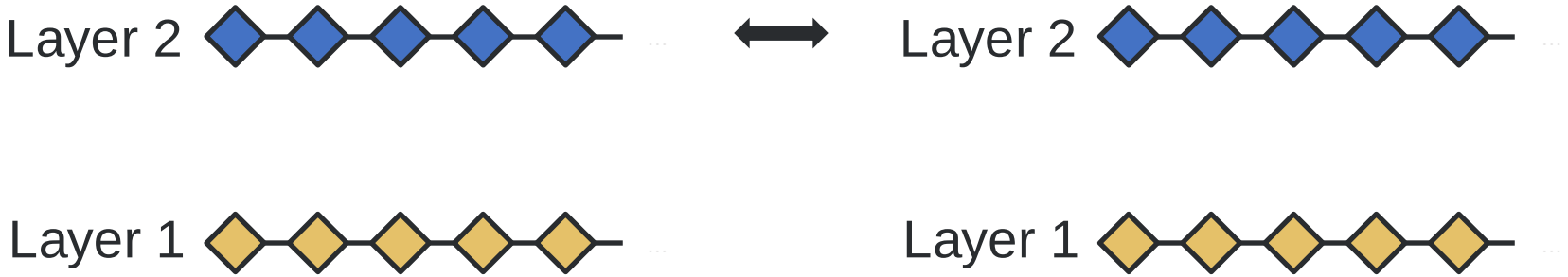


Offload computation to another layer (L2) and publish new state roots into the L1. May be accompanied by computation proofs (as in the case of zk-rollups)

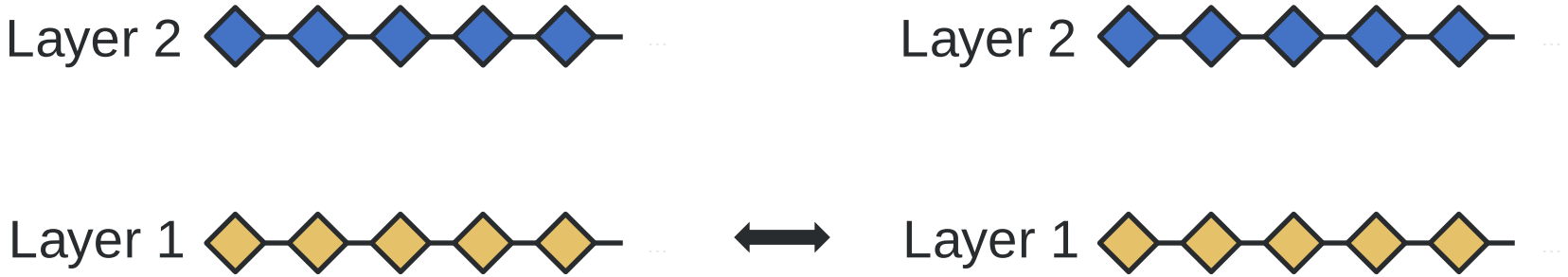
Scaling Blockchains



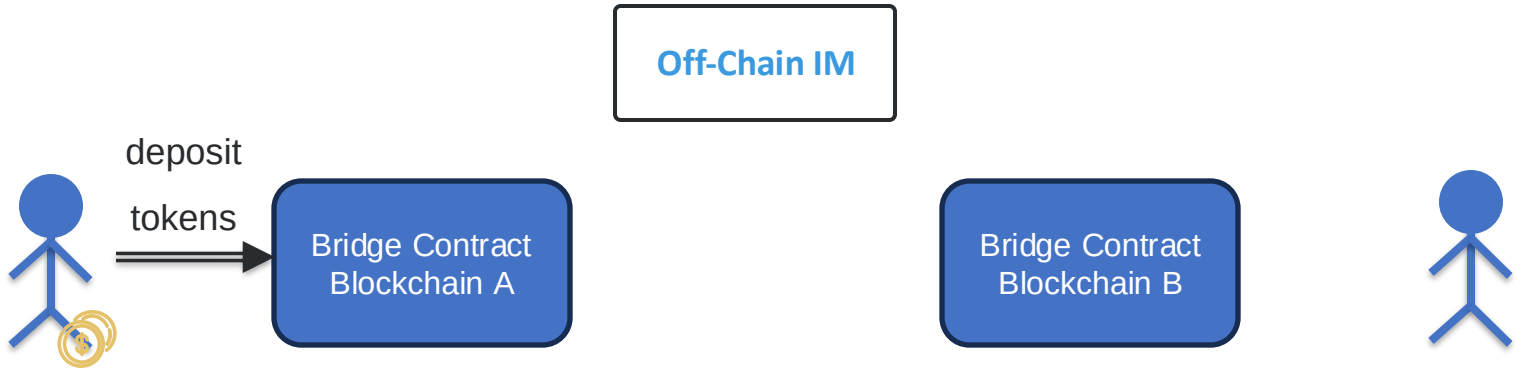
What about connecting L2s?



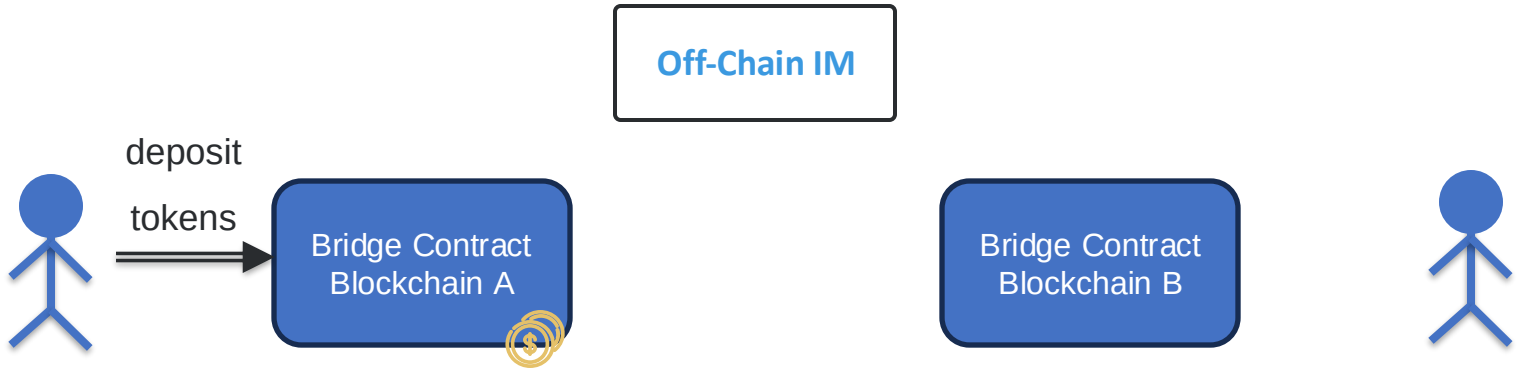
What about connecting L1s?



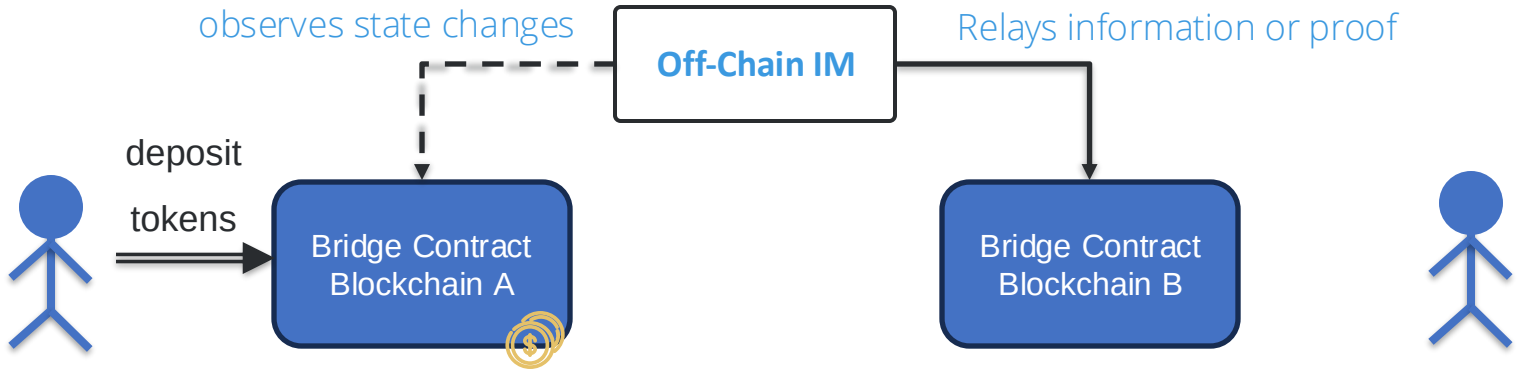
Example: how does a token bridge work?



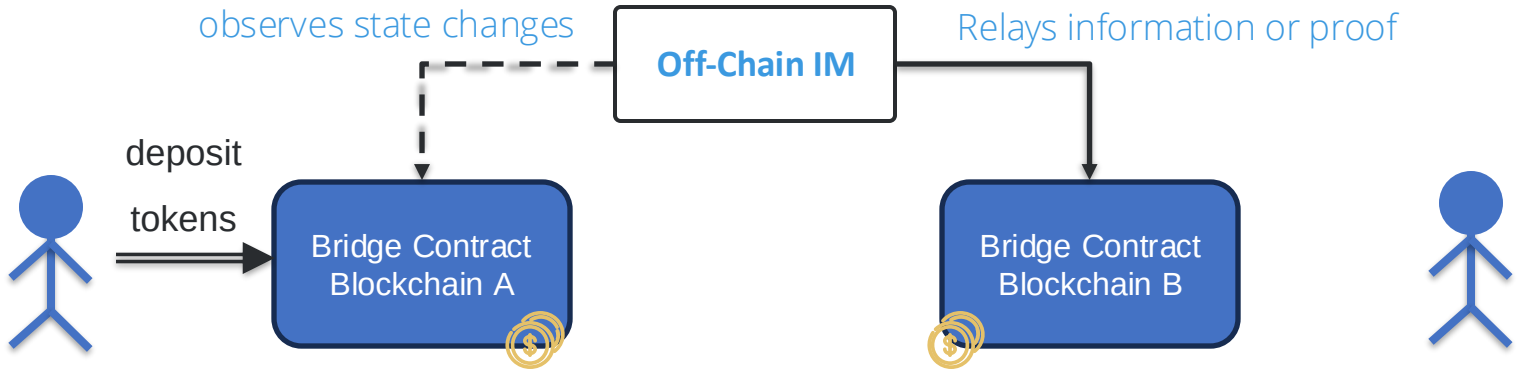
Example: how does a token bridge work?



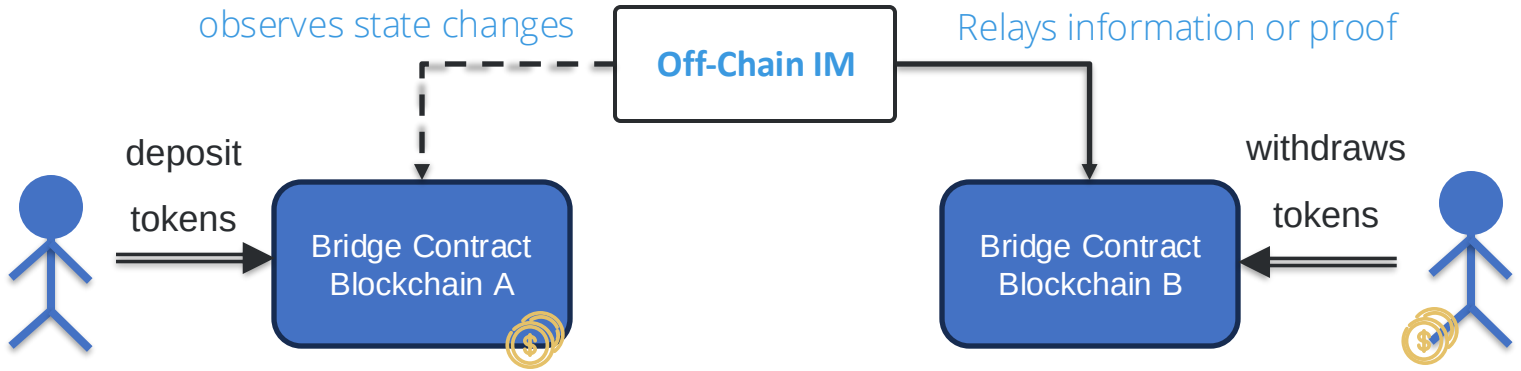
Example: how does a token bridge work?



Example: how does a token bridge work?



Example: how does a token bridge work?



There are multiple modes:

- Lock-mint (in the diagram)
- Burn-mint
- Lock-unlock

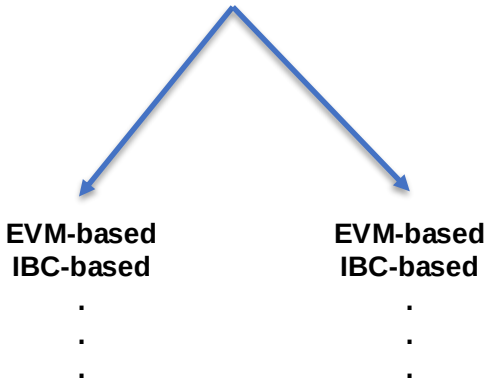
Blockchain Interoperability

"the ability of a source blockchain to change the state of a target blockchain (or vice-versa), enabled by cross-chain or cross-blockchain transactions, spanning across a composition of homogeneous and heterogeneous blockchain systems"

Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. 2021. A Survey on Blockchain Interoperability: Past, Present, and Future Trends. ACM Comput. Surv. 54, 8, Article 168 (November 2022), 41 pages. https://doi.org/10.1145/3471140

Blockchain Interoperability

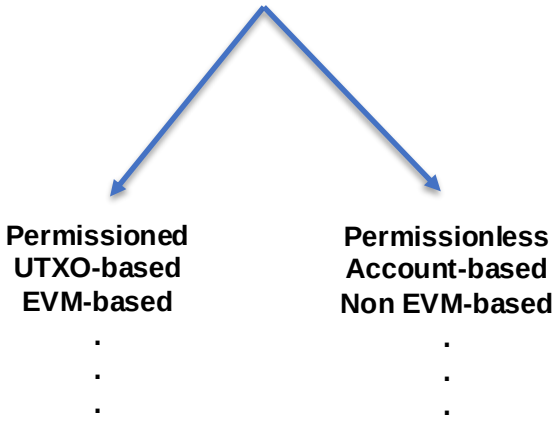
"the ability of a source blockchain to change the state of a target blockchain (or vice-versa), enabled by cross-chain or cross-blockchain transactions, spanning across a composition of homogeneous and heterogeneous blockchain systems"



Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. 2021. A Survey on Blockchain Interoperability: Past, Present, and Future Trends. ACM Comput. Surv. 54, 8, Article 168 (November 2022), 41 pages. https://doi.org/10.1145/3471140

Blockchain Interoperability

"the ability of a source blockchain to change the state of a target blockchain (or vice-versa), enabled by cross-chain or cross-blockchain transactions, spanning across a composition of homogeneous and heterogeneous blockchain systems"



Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. 2021. A Survey on Blockchain Interoperability: Past, Present, and Future Trends. ACM Comput. Surv. 54, 8, Article 168 (November 2022), 41 pages. https://doi.org/10.1145/3471140

In a Nutshell... Interoperability:

Enables connectivity between **Homogeneous** or **Heterogeneous platforms**

Reduces liquidity fragmentation across DeFi protocols in multiple blockchains (L1s or L2s)

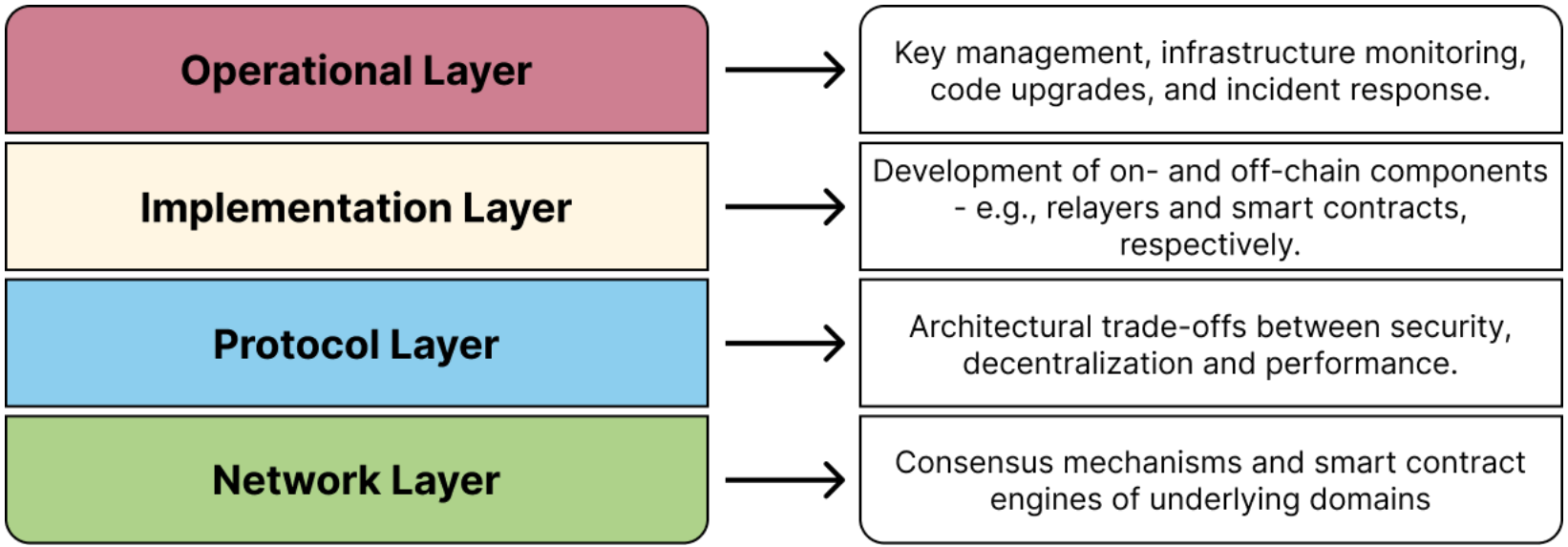
The Core Idea: Enables the **seamless flow of ~~assets and data~~** across platforms

The Core Idea: Enables the **seamless flow of value** across platforms

Outline

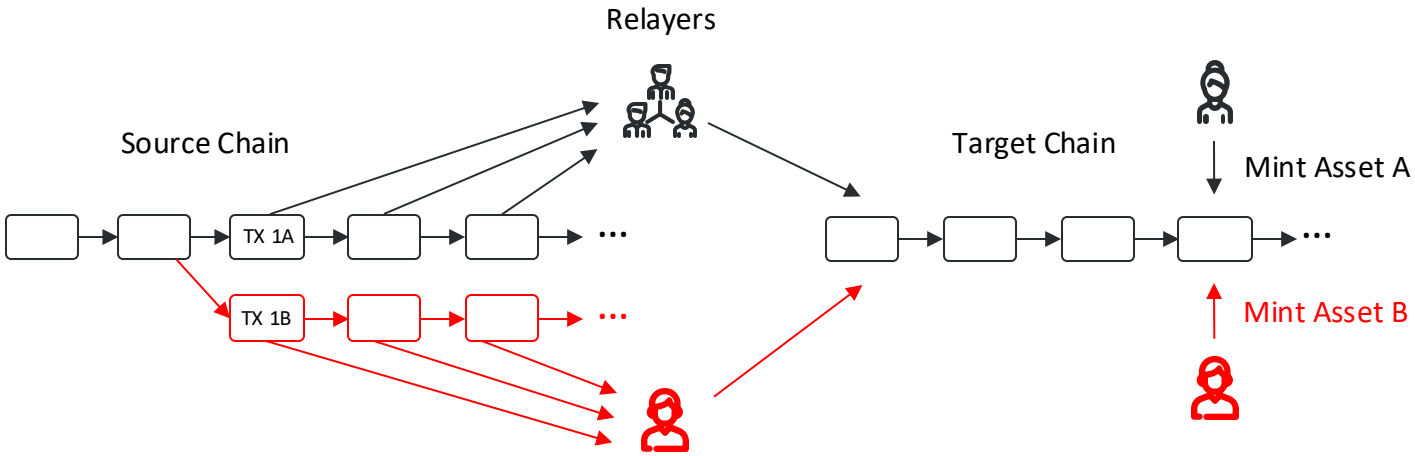
- ~~Motivation (Why?, How?, What?)~~
- Blockchain Interoperability and Interoperability Mechanisms
- Security and Privacy of Interoperability Mechanisms
- Securing interoperability solutions: Hephaestus and XChainWatcher
- Future Research Directions

Building Blocks to Make It Work



A. Augusto, et al., "SoK: Security and Privacy of Blockchain Interoperability," in 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2024 pp. 3840-3865. doi: 10.1109/SP54263.2024.00255

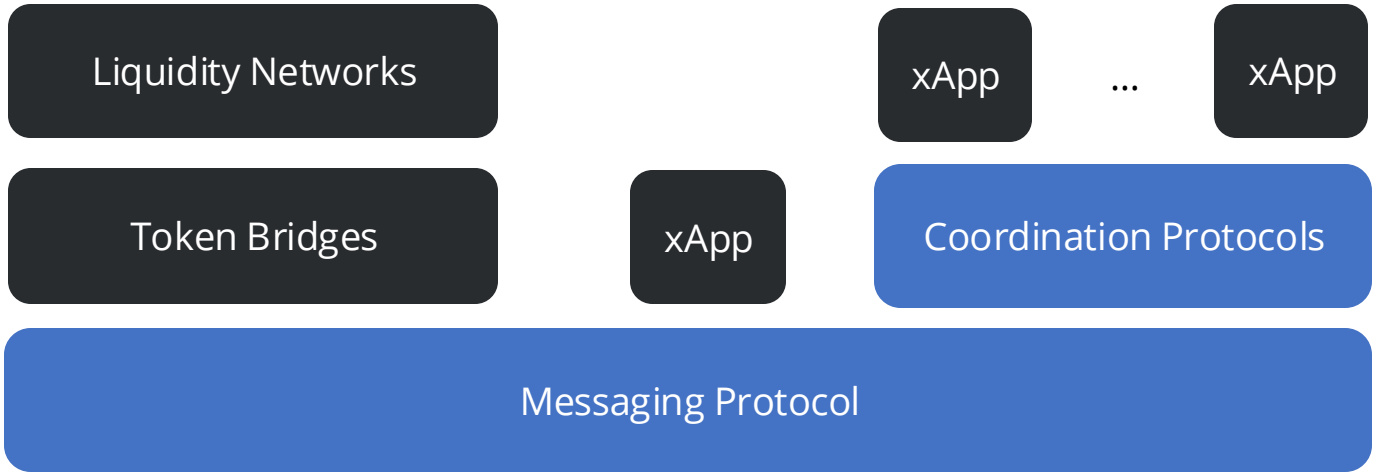
Example: the importance of the network layer



TX 1A — Lock Asset A

TX 1B — Lock Asset B

The Protocol Layer



Source: Abebe, E., Robinson, P., Chand, A., Murdock, M., & Hyland-Wood, D. Crosschain Risk Framework. https://crosschainriskframework.github.io/

Blockchain Interoperability

Architectures

Centralization

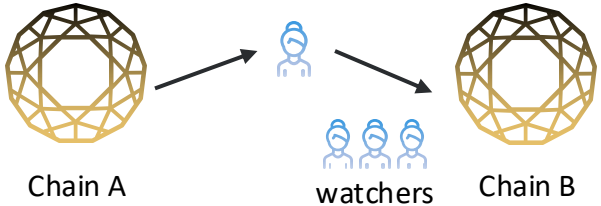
Trusted Computation

Permissionless Network

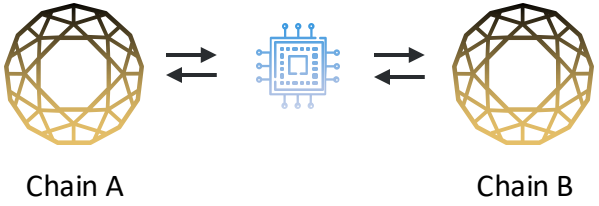
Permissioned Network

Architectures

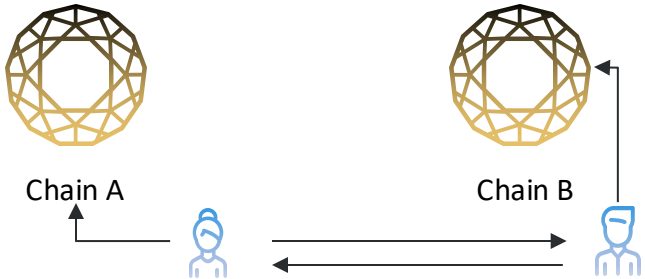
Fraud Proofs



Validity Proofs (e.g., SNARKs)



Hash and Time Locks



and more…

Example: The Secure Asset Transfer Protocol (SATP) Model

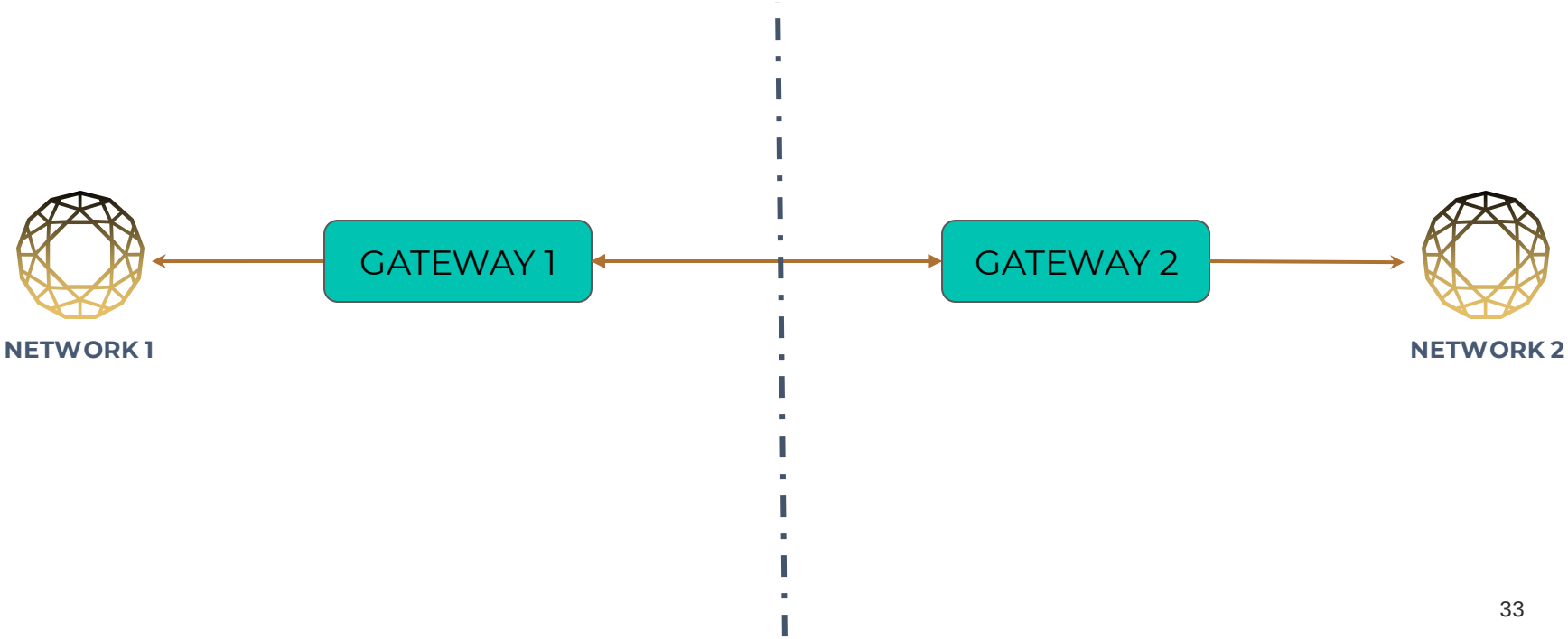
MIT Connection Science
the technology of innovation

Example: The Secure Asset Transfer Protocol (SATP) Model

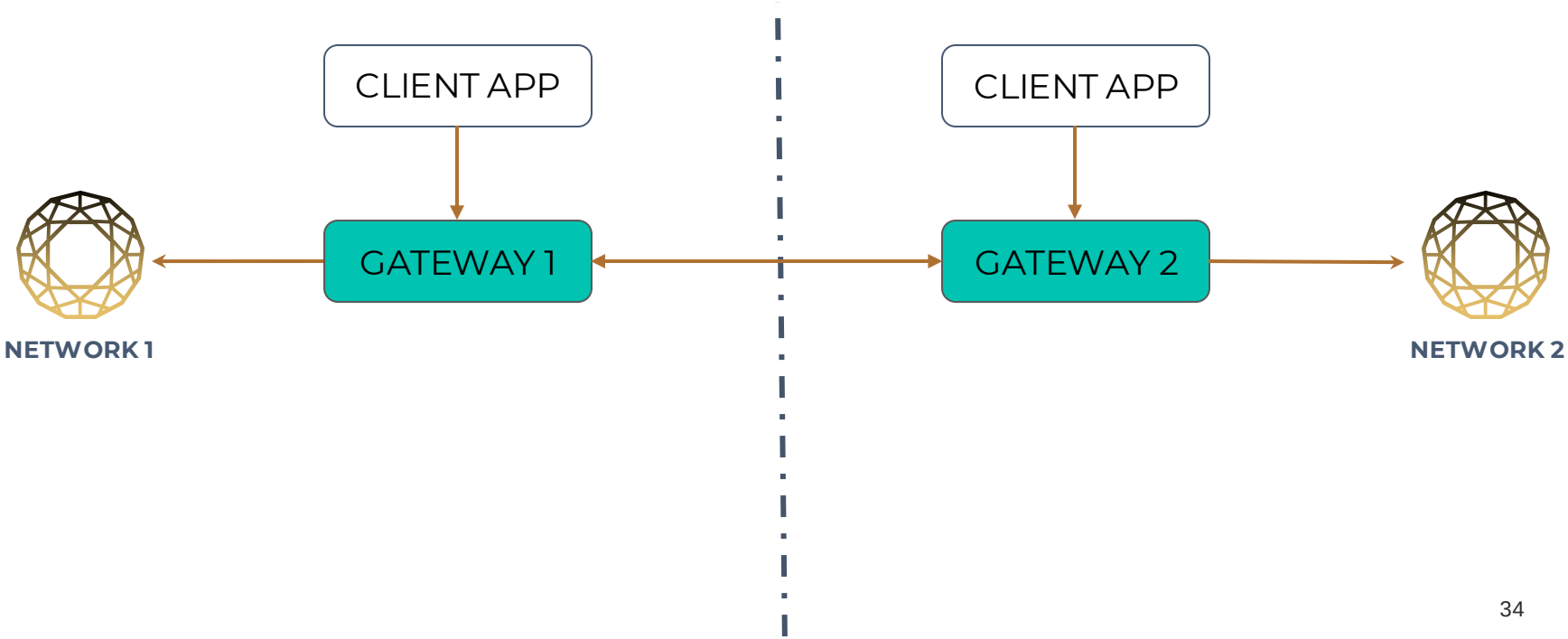
NETWORK 1

NETWORK 2

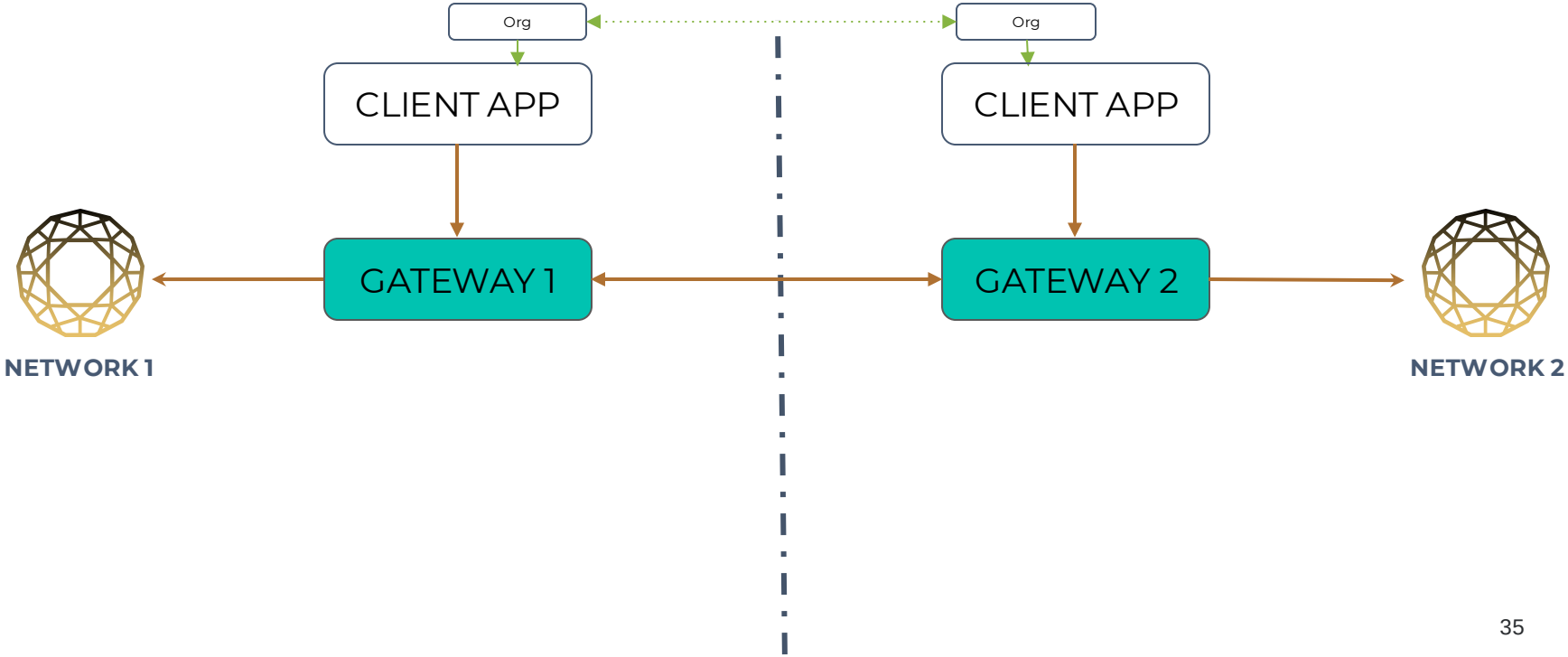
Example: The Secure Asset Transfer Protocol (SATP)



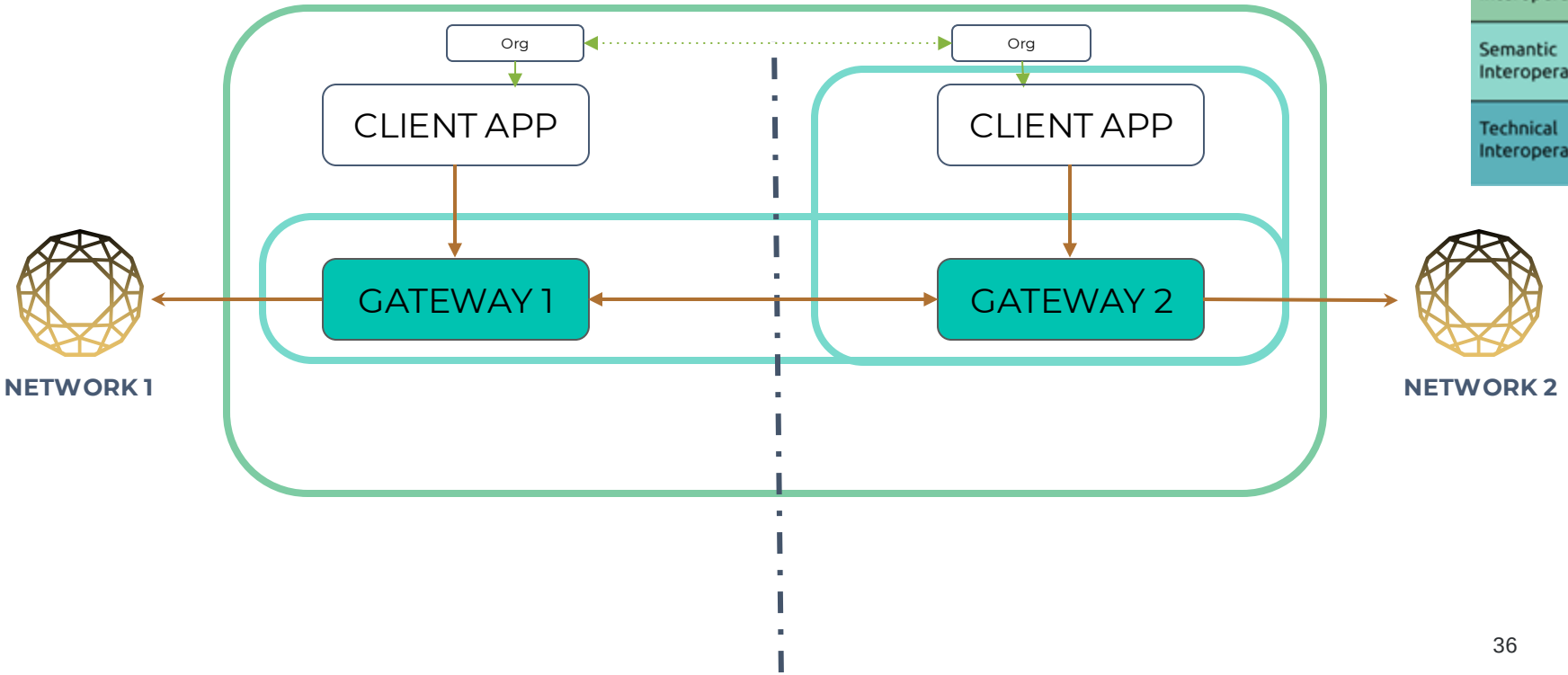
Example: The Secure Asset Transfer Protocol (SATP)



Example: The Secure Asset Transfer Protocol (SATP)



Example: The Secure Asset Transfer Protocol (SATP)



Legal Interoperability

Organizational Interoperability

Semantic Interoperability

Technical Interoperability

Secure Asset Transfer Protocol (SATP)

37

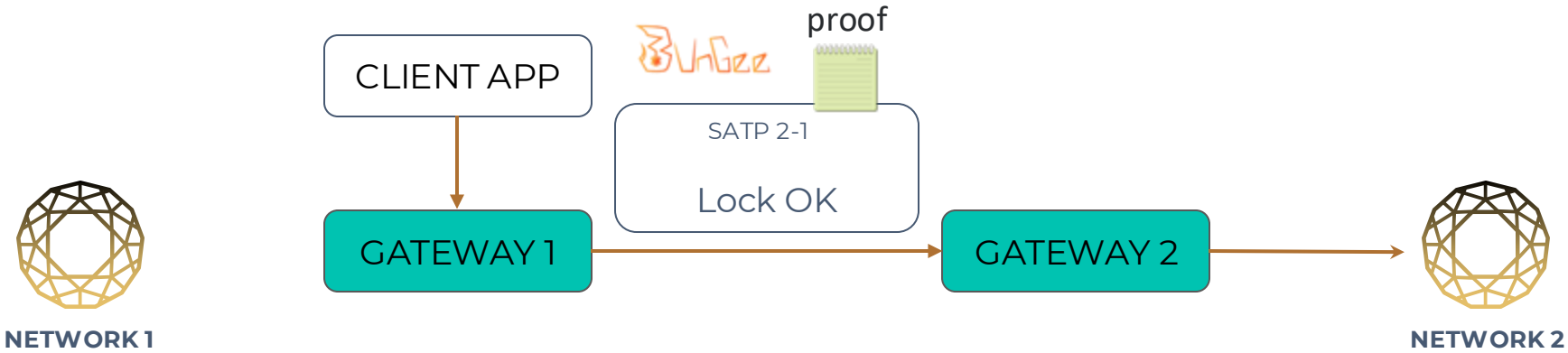
Secure Asset Transfer Protocol (SATP)

38

Secure Asset Transfer Protocol (SATP)

39

Secure Asset Transfer Protocol (SATP)

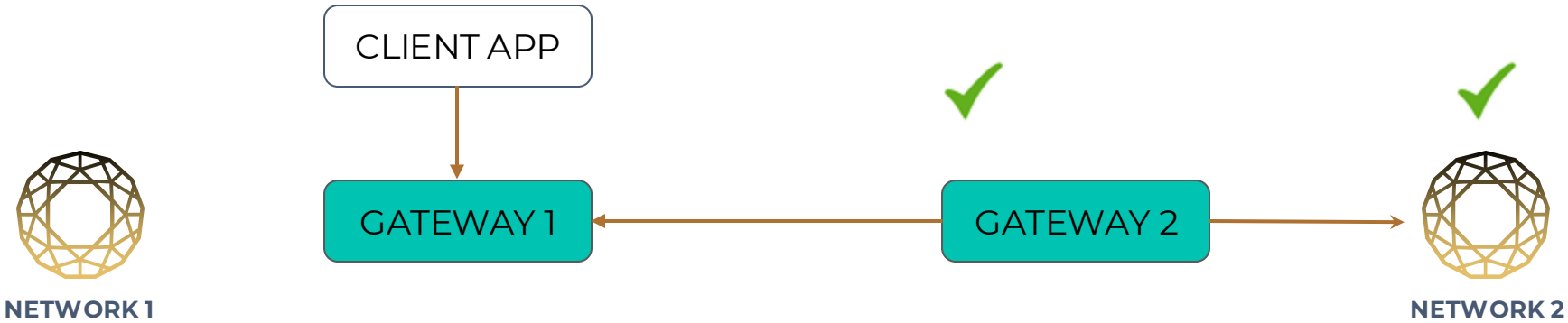


Proving Systems:

SNARK-based
Merkle Proofs
...

40

Secure Asset Transfer Protocol (SATP)

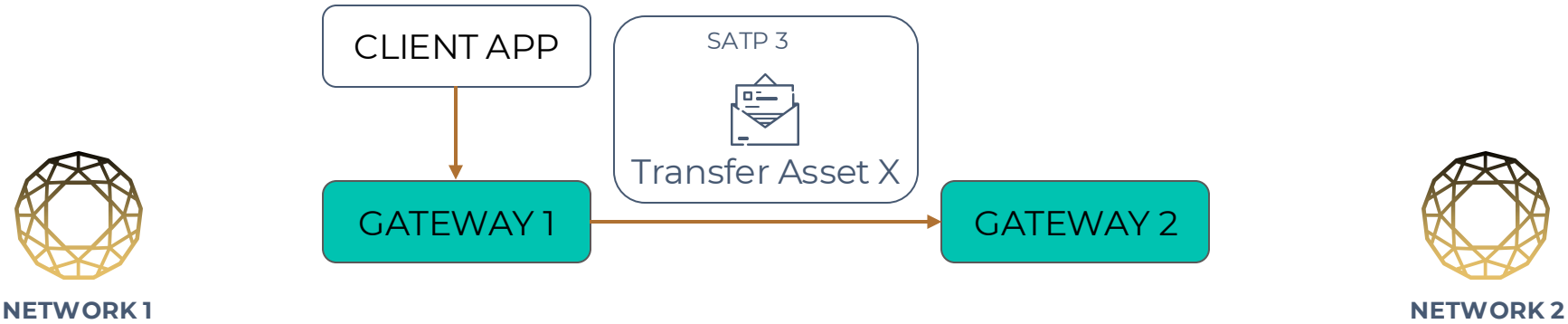


41

Secure Asset Transfer Protocol (SATP)

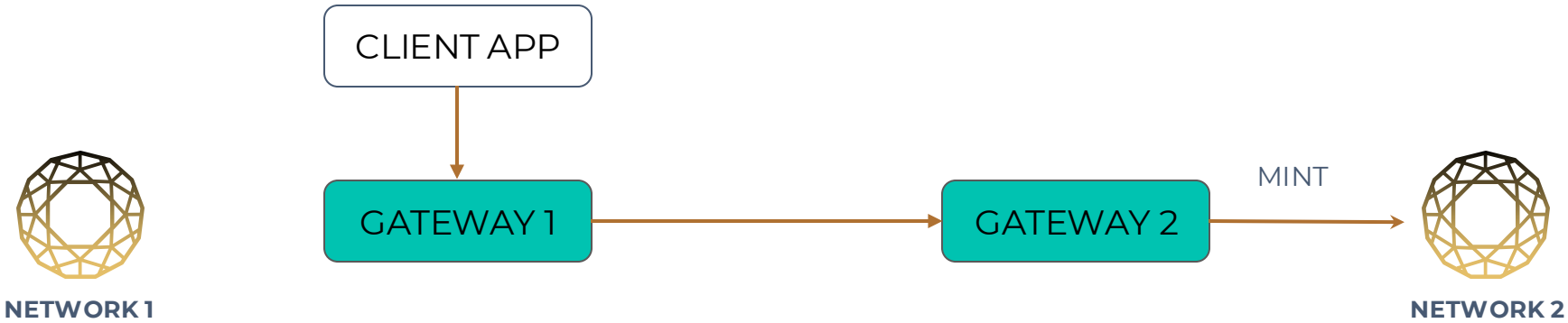
42

Secure Asset Transfer Protocol (SATP)



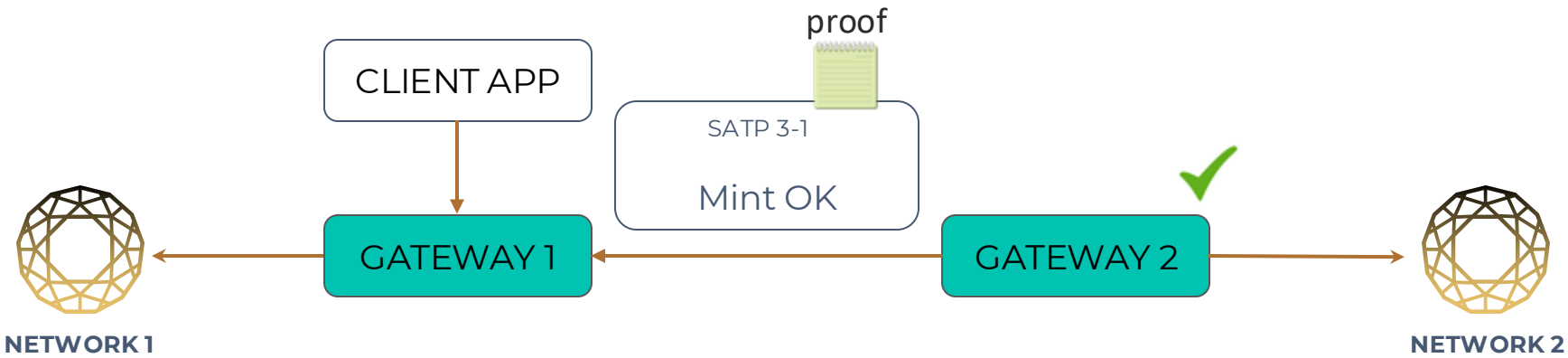
43

Secure Asset Transfer Protocol (SATP)



44

Secure Asset Transfer Protocol (SATP)

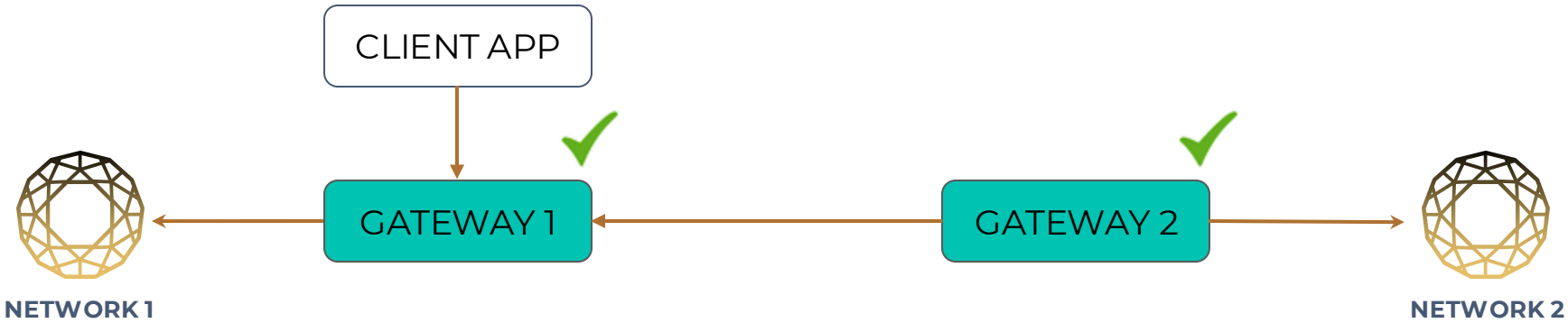


Proving Systems:

SNARK-based
Merkle Proofs
...

45

Secure Asset Transfer Protocol (SATP)



46

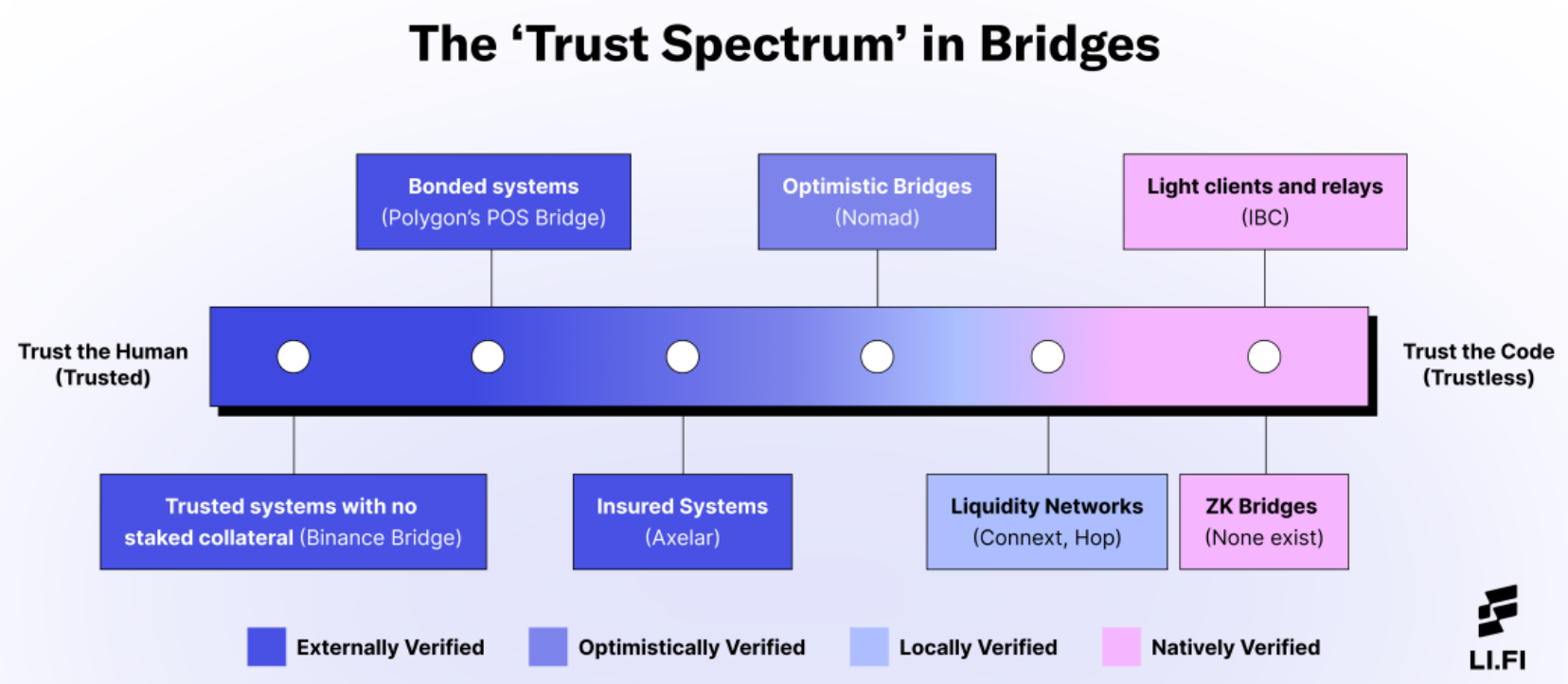
Outline

- ~~Motivation (Why?, How?, What?)~~
- ~~Blockchain Interoperability and Interoperability Mechanisms~~
- Security and Privacy of Interoperability Mechanisms
- Securing interoperability solutions: Hephaestus and XChainWatcher
- Future Research Directions

How to classify IMs based on security guarantees?

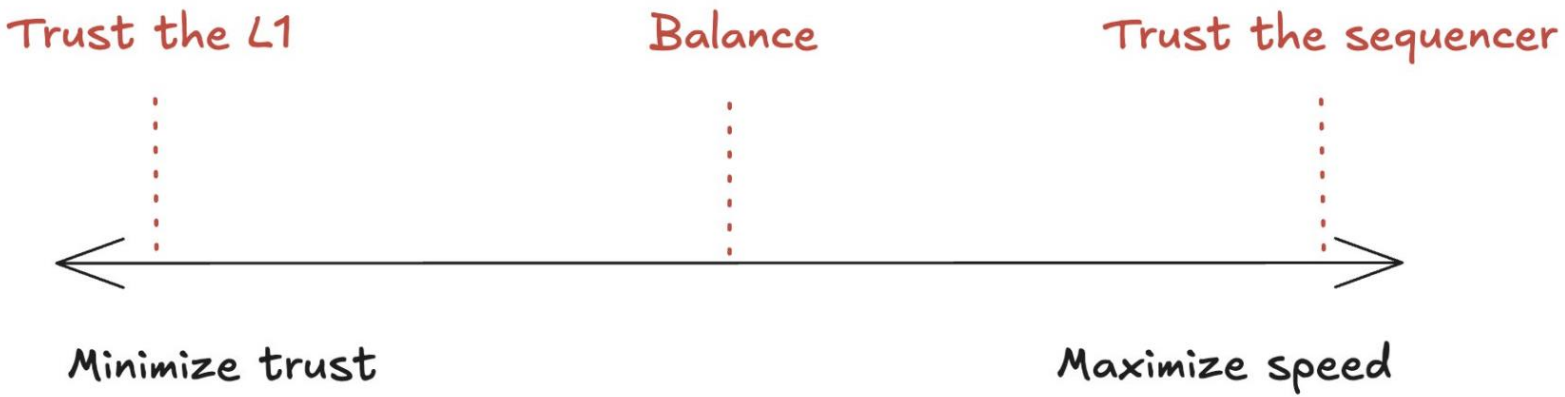
*“There exists no asynchronous **cross-chain communication protocol** tolerant against misbehaving nodes without a **trusted** third **party**.”*

Trust spectrum



Source: https://blog.li.fi/li-fi-with-bridges-trust-is-a-spectrum-354cd5a1a6d8

Trust spectrum (Rollups)



Source: https://x.com/brobobo_bo_bobo/status/1860004523101814956

Is the *Trust Spectrum* Enough? NO

So…what does a secure interoperability solution look like?

A set of properties

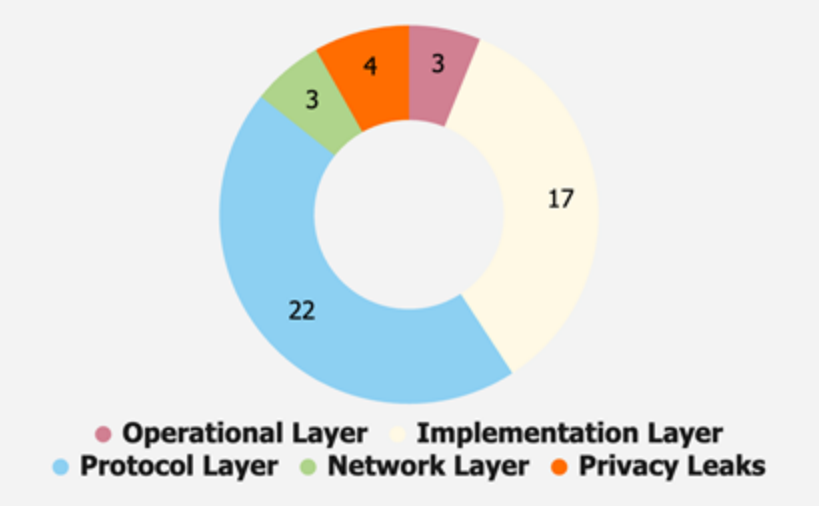
Integrity — of the system, data, and assets

Accountability — of participants for integrity breach attempts

Availability — of system to process cross-chain transactions

A. Augusto, et al., "SoK: Security and Privacy of Blockchain Interoperability," in 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2024 pp. 3840-3865. doi: 10.1109/SP54263.2024.00255

Vulnerabilities in Interoperability



CROSS-CHAIN SYSTEMS. THE COLORED CIRCLE DENOTES THE LAYER WHERE IT CAN BE FOUND (CF. SECTION 3.1).

Vulnerability/Leak	Mitigations
$\mathcal{V}_1$ Honest mining assumption [45]	$\mathcal{M}_1$-$\mathcal{M}_5$
$\mathcal{V}_2$ Absence of identity verification [45], [71], [72]	$\mathcal{M}_8$-$\mathcal{M}_{11}$
$\mathcal{V}_3$ Network isolation [38], [45], [62], [77]	$\mathcal{M}_6$,$\mathcal{M}_7$
$\mathcal{V}_4$ Outdated light client state [45], [53], [150]	$\mathcal{M}_{16}$
$\mathcal{V}_5$ Wrong main chain identification [6], [45], [77]	$\mathcal{M}_{18}$
$\mathcal{V}_6$ Incorrect event verification [151]–[154]	$\mathcal{M}_{12}$-$\mathcal{M}_{14}$
$\mathcal{V}_7$ Acceptance of invalid consensus proofs [155]	$\mathcal{M}_{15}$
$\mathcal{V}_8$ Absence of chain identification [156]	$\mathcal{M}_4$
$\mathcal{V}_9$ Submission of repeated inclusion proofs [21], [45], [77], [157]	$\mathcal{M}_{17}$
$\mathcal{V}_{10}$ Counterfeiting assets [45], [77], [158]	$\mathcal{M}_{19}$-$\mathcal{M}_{23}$
$\mathcal{V}_{11}$ Involuntary timelock expiry [63], [85]	$\mathcal{M}_{29}$-$\mathcal{M}_{30}$
$\mathcal{V}_{12}$ Unset withdrawal limits [156], [159]	$\mathcal{M}_{69}$
$\mathcal{V}_{13}$ Action withhold [58], [61], [80], [86], [86], [94], [160]	$\mathcal{M}_8$,$\mathcal{M}_{27}$,$\mathcal{M}_{28}$
$\mathcal{V}_{14}$ Unspecified gas limit [161]	$\mathcal{M}_{65}$
$\mathcal{V}_{15}$ Resource exhaustion [45], [55], [57], [60], [65], [69]	$\mathcal{M}_{48}$-$\mathcal{M}_{50}$
$\mathcal{V}_{16}$ Single point of failure [156], [162]	$\mathcal{M}_7$,$\mathcal{M}_{32}$,$\mathcal{M}_{47}$
$\mathcal{V}_{17}$ Publicly identifiable operators [74]	$\mathcal{M}_{44}$-$\mathcal{M}_{46}$
$\mathcal{V}_{18}$ Misaligned incentive mechanisms [38], [60], [65], [122]	$\mathcal{M}_{23}$,$\mathcal{M}_{31}$-$\mathcal{M}_{34}$

Attacks in Cross-Chain Bridges

Project Information		General Attack		Mapping to Theoretical Vulnerabilities					
Name & Ref	SA	Date	Amount	$\mathcal{V}_{44}$	$\mathcal{V}_{43}$	$\mathcal{V}_{28}$	$\mathcal{V}_{27}$	$\mathcal{V}_{24}$	$\mathcal{V}_{6}$
[218] Ronin	$\mathcal{SA}_{22}$	Mar 2022	624M	✓	✓	✗	✗	✗	✗
[219] PolyBridge #1	$\mathcal{SA}_{22}$	Aug 2021	611M	✗	✓	✓	✗	✗	✗
[220] BNB	$\mathcal{SA}_{11}$	Oct 2022	566M	✗	✗	✗	✗	✓	✗
[123] Wormhole	$\mathcal{SA}_{22}$	Feb 2022	326M	✗	✗	✓	✗	✓	✗
[221] Nomad	$\mathcal{SA}_{33}$	Aug 2022	190M	✗	✗	✗	✗	✓	✗
[222] BXH	$\mathcal{SA}_{11}$	Oct 2021	139M	✓	✓	✗	✗	✗	✗
[223] Multichain #2	$\mathcal{SA}_{22}$	Jul 2023	126M	✓†	✓†	✗	✗	✗	✗
[224] Harmony	$\mathcal{SA}_{22}$	Jun 2022	100M	✓	✓	✗	✗	✗	✗
[225] Qubit	$\mathcal{SA}_{11}$	Jan 2022	80M	✗	✗	✗	✓	✓	✗
[226] pNetwork	$\mathcal{SA}_{33}$	Sep 2021	13M	✗	✗	✗	✗	✗	✓
[227] Thorchain #3	$\mathcal{SA}_{21}$	Jul 2021	8M	✗	✗	✗	✗	✗	✓
[223] Anyswap	$\mathcal{SA}_{22}$	Jul 2021	8M	✗	✓	✗	✗	✗	✗
[227] Thorchain #2	$\mathcal{SA}_{21}$	Jul 2021	5M	✗	✗	✗	✗	✓	✓
[219] PolyBridge #2	$\mathcal{SA}_{22}$	Jul 2023	4.4M	✗	✓	✗	✗	✗	✗
[228] Meter	$\mathcal{SA}_{22}$	Jul 2021	4.4M	✗	✗	✗	✗	✓	✗
[229] Chainswap	$\mathcal{SA}_{22}$	Jul 2021	4.4M	✗	✗	✓	✗	✓	✗
[223] Multichain #1	$\mathcal{SA}_{22}$	Jan 2022	3M	✗	✗	✗	✓	✓	✗
[227] Thorchain #1	$\mathcal{SA}_{21}$	Jun 2021	140K	✗	✗	✗	✗	✗	✓
Summary		**07/21 - 07/23**	**2.9B**	**22%**	**39%**	**17%**	**11%**	**44%**	**22%**

A. Augusto, et al., "SoK: Security and Privacy of Blockchain Interoperability," in 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2024 pp. 3840-3865. doi: 10.1109/SP54263.2024.00255

Vulnerabilities Behind

Project Information		General Attack Information					Incident Resp		Where		Mapping to Theoretical Vulnerabilities					
Name & Ref	SA	Date	Amount	AT	Txs	Mix	DT	CT	VL	EL	$\mathcal{V}_{44}$	$\mathcal{V}_{43}$	$\mathcal{V}_{28}$	$\mathcal{V}_{27}$	$\mathcal{V}_{24}$	$\mathcal{V}_{6}$

Physical Infrastructure Backdoors

Bad key Management

Dead code

Unsafe Third-party software

Lack of access control

Incorrect event verification

~66% used a *Permissioned Network* as Architecture

What about Privacy?

Privacy Brings Additional Challenges

- What do
- A
- I
- What if w design a system that guarantees all these properties?

Well…

OFAC

OFFICE OF FOREIGN ASSETS CONTROL

The Obvious Example

…and it “only” provides the unlinkability of transactions in one blockchain

Interesting Connection with Bridge Attacks

Name & Ref	Date	Amount	Mix
[193] Ronin	Mar 2022	624M	◑
[194] PolyBridge #1	Aug 2021	611M	○
[195] BNB	Oct 2022	566M	◐
[108] Wormhole	Feb 2022	326M	◐
[196] Nomad	Aug 2022	190M	◑
[197] BXH	Oct 2021	139M	◐
[198] Multichain #2	Jul 2023	126M	○
[199] Harmony	Jun 2022	100M	◑
[200] Qubit	Jan 2022	80M	◑
[201] pNetwork	Sep 2021	13M	○
[202] Thorchain #3	Jul 2021	8M	◑
[198] Anyswap	Jul 2021	8M	◑
[202] Thorchain #2	Jul 2021	5M	◑
[194] PolyBridge #2	Jul 2023	4.4M	○
[203] Meter	Jul 2021	4.4M	◑
[204] Chainswap	Jul 2021	4.4M	●
[198] Multichain #1	Jan 2022	3M	●
[202] Thorchain #1	Jun 2021	140K	◑

14 out of 18
used Transaction Mixers,
mainly Tornado Cash

Usage of Mixers (Mix)
- ○ Not used
- ◐ Before the attack
- ◑ After the attack
- ● Before and after the attack

Would a cross-chain protocol with the same level of privacy **be sanctioned**?

SANCTIONED

Explore the notion of ***Revokable Privacy***. Is it possible to **guarantee these properties** if and only **if there is no misbehavior**?

Outline

- ~~Motivation (Why?, How?, What?)~~
- ~~Blockchain Interoperability and Interoperability Mechanisms~~
- ~~Security and Privacy of Interoperability Mechanisms~~
- Securing interoperability solutions: Hephaestus and XChainWatcher
- Future Research Directions

A Prominent Problem

TABLE 5. CLASSIFICATION OF MOST PROFITABLE CROSS-CHAIN BRIDGE HACKS GROUPED BY USD. THE CELLS WITH THE VULNERABILITY NUMBER ARE FILLED WITH THE COLOR ACCORD WE ADD A "SUMMARY" ROW THAT AGGREGATES INFORMATION. SPECIFICALLY, WE USE CELL EACH VULNERABILITY WAS FOUND.

Project Information		General Attack Information					Incident Resp	
Name & Ref	SA	Date	Amount	AT	Txs	Mix	DT	CT
[218] Ronin	SA_{22}	Mar 2022	624M	■	○	◑	6d	●
[219] PolyBridge #1	SA_{22}	Aug 2021	611M	□	◔	○	–	◔
[220] BNB	SA_{11}	Oct 2022	566M	■	◔	◐	–	◑
[123] Wormhole	SA_{22}	Feb 2022	326M	■	○	◐	–	○
[221] Nomad	SA_{33}	Aug 2022	190M	◧	◕	◑	–	◔
[222] BXH	SA_{11}	Oct 2021	139M	■	○	◐	–	◑
[223] Multichain #2	SA_{22}	Jul 2023	126M	■	○	○	–	◑
[224] Harmony	SA_{22}	Jun 2022	100M	■	◔	◑	–	◕
[225] Qubit	SA_{11}	Jan 2022	80M	■	◔	◑	–	◔
[226] pNetwork	SA_{33}	Sep 2021	13M	■	◔	○	13m	◔
[227] Thorchain #3	SA_{21}	Jul 2021	8M	■	○	◑	–	–
[223] Anyswap	SA_{22}	Jul 2021	8M	■	○	◑	–	◕
[227] Thorchain #2	SA_{21}	Jul 2021	5M	■	◕	◑	–	◑
[219] PolyBridge #2	SA_{22}	Jul 2023	4.4M	■	◑	○	7h	◕
[228] Meter	SA_{22}	Jul 2021	4.4M	■	○	◑	–	◔
[229] Chainswap	SA_{22}	Jul 2021	4.4M	■	●	●	–	◑
[223] Multichain #1	SA_{22}	Jan 2022	3M	◧	–	●	–	◕
[227] Thorchain #1	SA_{21}	Jun 2021	140K	■	–	◑	5m	–
Summary		**07/21 - 07/23**	**2.9B**					

Communication Time (CT)
- ○]0; 2] hours
- ◔]2; 4] hours
- ◑]4; 6] hours
- ◕]6; 24] hours
- ● >= 6 days

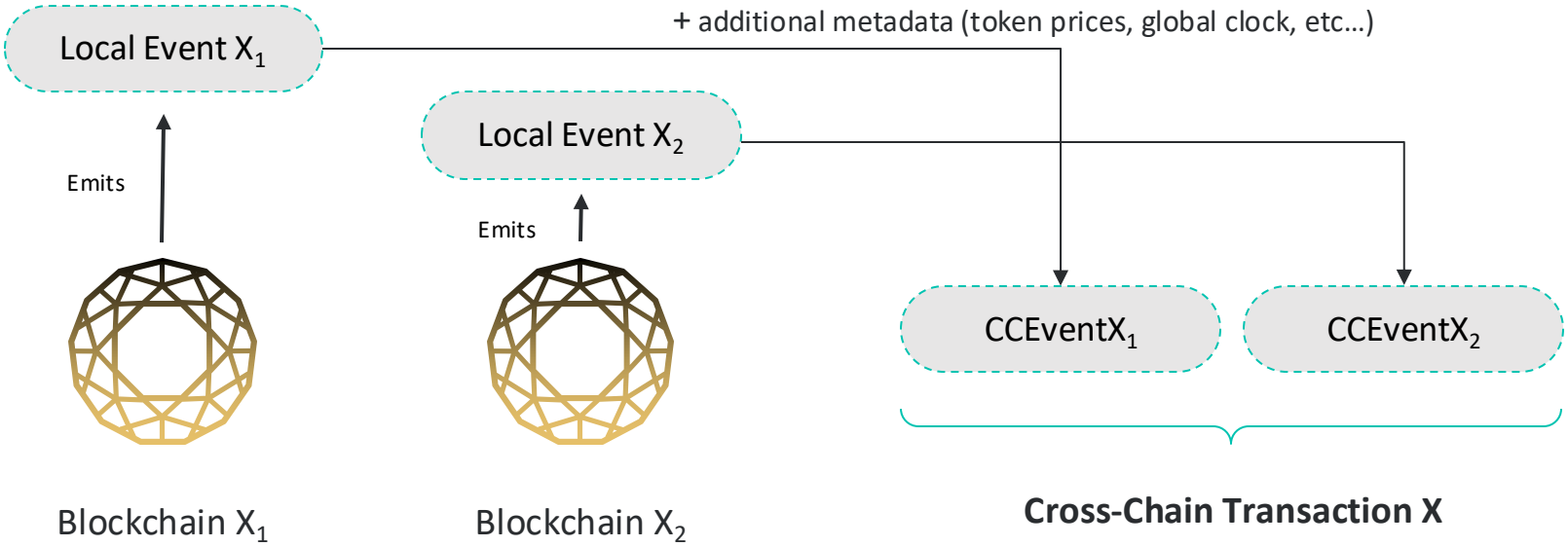
Attacks stole between 140K USD and ~620M USD

Defi Protocol LI.FI Struck by $11M Exploit

The exploit is reported to be related to the LI.FI bridge.

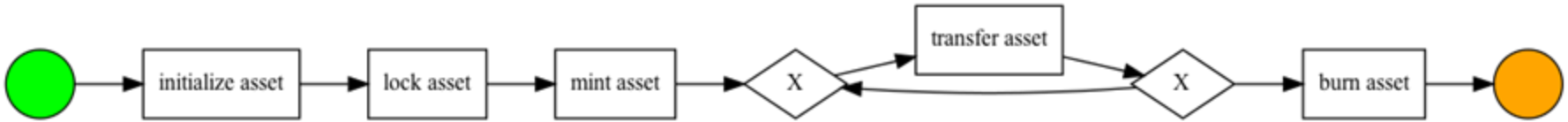
By Oliver Knight · Jul 16, 2024 at 2:30 p.m. · Updated Jul 16, 2024 at 8:45 p.m.

The Solution: Cross-Chain Modelling

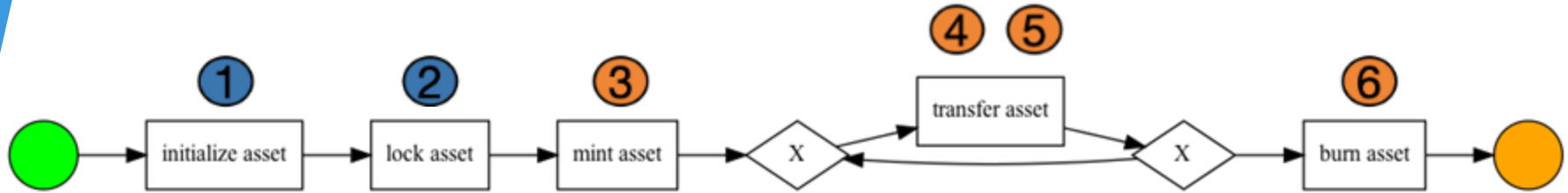


Belchior, R., Somogyvari, P., Pfannschmidt, J., Vasconcelos, A., & Correia, M. (2023). Hephaestus: Modeling, analysis, and performance evaluation of cross-chain transactions. *IEEE Transactions on Reliability*.

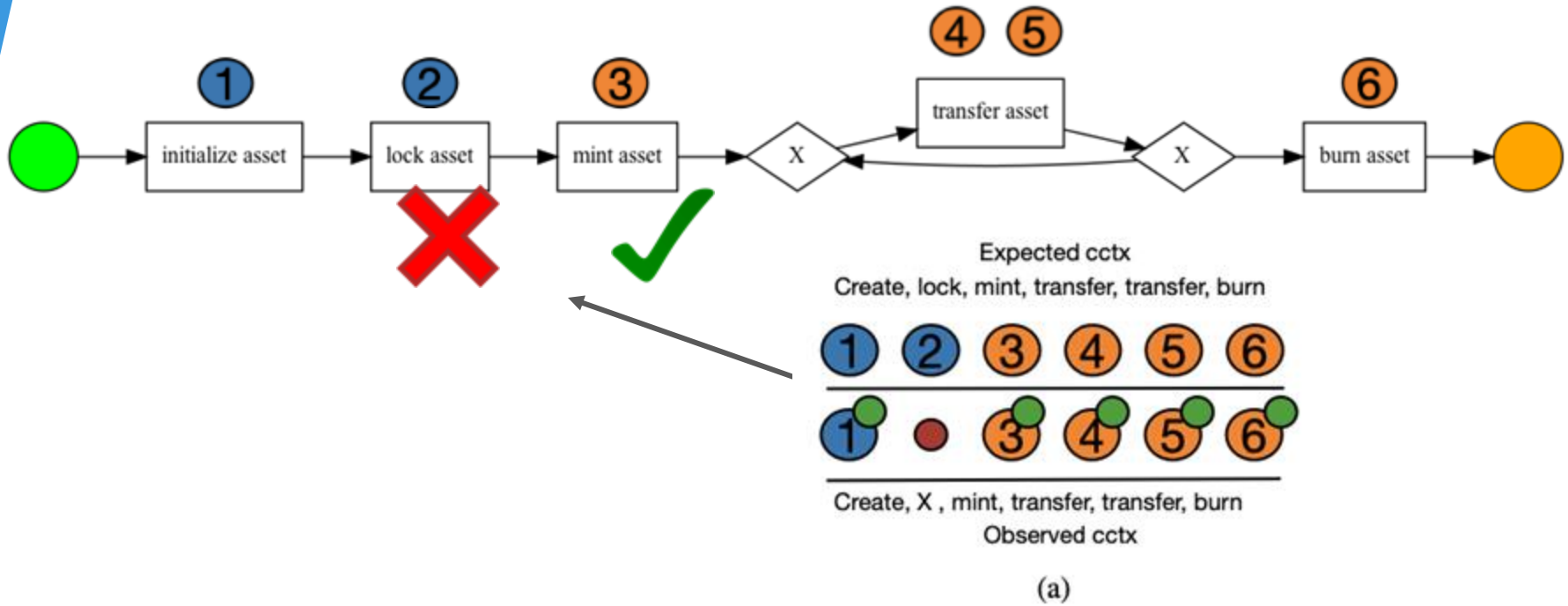
Key Idea (burn-mint model)



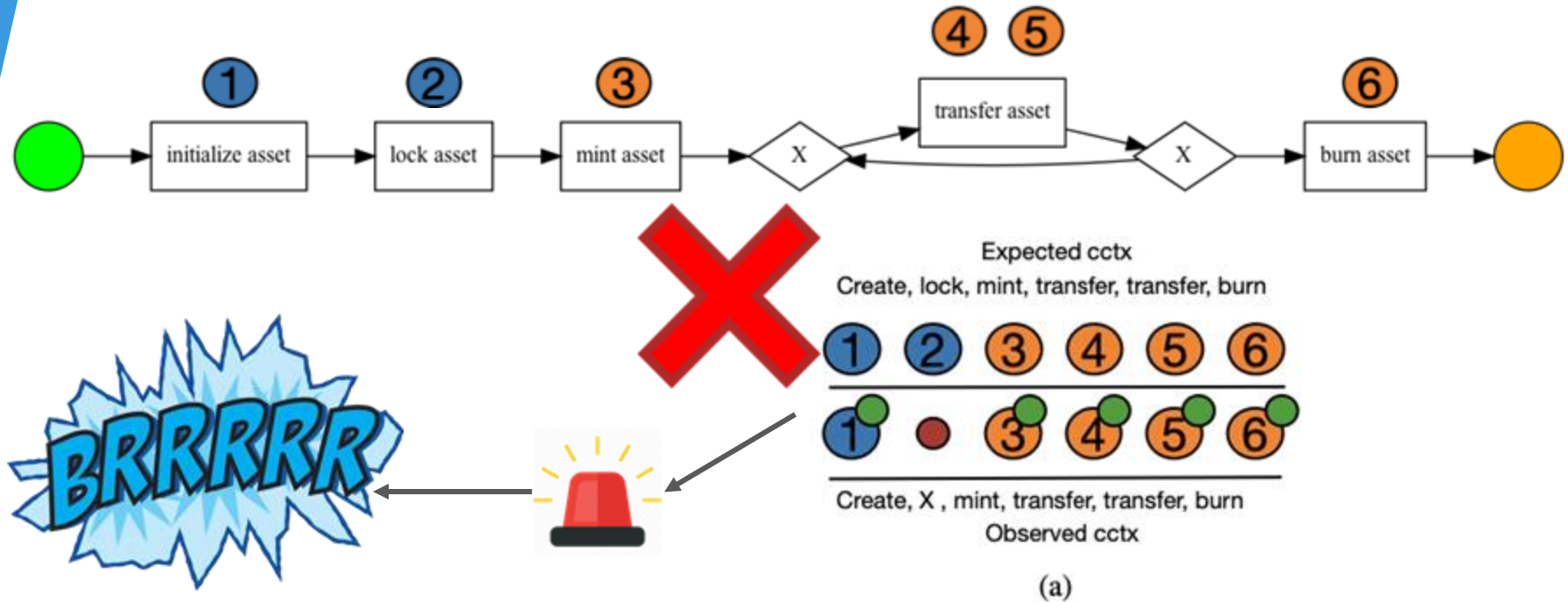
Key Idea (burn-mint model)



Key Idea (burn-mint model)



Key Idea (burn-mint model)



Algorithm 1: `Cross-Chain State Update.`
Creation of a cross-chain state from a set of *ccevents*

Input: Set of events $\mathcal{E}$
Input: State update algorithm `createCCState`
Input: Cross-chain rules $\mathcal{R}$
Input: Cross-chain state $\mathcal{S}$
Output: Upon success returns cross-chain state $\mathcal{S}$, and a *SYNC MOVE* 🟢

1 **require** `verifySatisfability`$(e, \mathcal{R}, \mathcal{S})$ `// Returns tuple (event,MOVE ON LOG` 🔴`) if event do not conform to the rules, cross-chain state is invalid.`
2 **foreach** $e \in \mathcal{E}$ **do**
3 `// For each event in retrieved event set`
4 **if** $\nexists \mathcal{S}[e.caseID]$ **then**
5 `// each cross-chain state key is indexed by case ID.`
6 $cc =$ `populateCCTX`$(\mathcal{S}[e.caseID], e)$
7 **end if**
8 **else**
9 $cc =$ `updateCCTx`$(\mathcal{S}[e.caseID], e)$
10 **end if**
11 $\mathcal{S} = \mathcal{S} \cup cc$
12 $\mathcal{S}' =$ `createCCState`$(\mathcal{S}, e.caseID)$
`// Calculates updated ccstate, algorithm is parametrizable`
13 **end foreach**
14 **return** ($\mathcal{S}'$, SYNC MOVE 🟢)

Our state is at position X. Each time a tx happens, we update the state

Algorithm 1: Cross-Chain State Update.
Creation of a cross-chain state from a set of *ccevents*

Input: Set of events $\mathcal{E}$
Input: State update algorithm `createCCState`
Input: Cross-chain rules $\mathcal{R}$
Input: Cross-chain state $\mathcal{S}$
Output: Upon success returns cross-chain state $\mathcal{S}$, and a *SYNC MOVE*

1 **require** `verifySatisfability`$(e, \mathcal{R}, \mathcal{S})$ // Returns tuple (event, *MOVE ON LOG*) if event do not conform to the rules, cross-chain state is invalid.
2 **foreach** $e \in \mathcal{E}$ **do**
3 // For each event in retrieved event set
4 **if** $\nexists \mathcal{S}[e.caseID]$ **then**
5 // each cross-chain state key is indexed by case ID.
6 $cc =$ `populateCCTX`$(\mathcal{S}[e.caseID], e)$
7 **end if**
8 **else**
9 $cc =$ `updateCCTx`$(\mathcal{S}[e.caseID], e)$
10 **end if**
11 $\mathcal{S} = \mathcal{S} \cup cc$
12 $\mathcal{S}' =$ `createCCState`$(\mathcal{S}, e.caseID)$
// Calculates updated ccstate, algorithm is parametrizable
13 **end foreach**
14 **return** ($\mathcal{S}'$, SYNC MOVE)

Our state is at position X. Each time a tx happens, we update the state

Check for non-modelled behavior

Algorithm 1: Cross-Chain State Update.
Creation of a cross-chain state from a set of *ccevents*

Input: Set of events $\mathcal{E}$
Input: State update algorithm `createCCState`
Input: Cross-chain rules $\mathcal{R}$
Input: Cross-chain state $\mathcal{S}$
Output: Upon success returns cross-chain state $\mathcal{S}$, and a *SYNC MOVE*

require $\texttt{verifySatisfability}(e, \mathcal{R}, \mathcal{S})$ // Returns tuple (event, *MOVE ON LOG*) if event do not conform to the rules, cross-chain state is invalid.
foreach $e \in \mathcal{E}$ **do**
  // For each event in retrieved event set
  if $\nexists \mathcal{S}[e.caseID]$ **then**
    // each cross-chain state key is indexed by case ID.
    $cc = \texttt{populateCCTX}(\mathcal{S}[e.caseID], e)$
  end if
  else
    $cc = \texttt{updateCCTx}(\mathcal{S}[e.caseID], e)$
  end if
  $\mathcal{S} = \mathcal{S} \cup cc$
  $\mathcal{S}' = \texttt{createCCState}(\mathcal{S}, e.caseID)$
  // Calculates updated ccstate, algorithm is parametrizable
end foreach
return ($\mathcal{S}'$, SYNC MOVE)

Our state is at position X. Each time a tx happens, we update the state

Check for non-modelled behavior

Update the state of the cross-chain model

Algorithm 1: Cross-Chain State Update.
Creation of a cross-chain state from a set of *ccevents*

Input: Set of events $\mathcal{E}$
Input: State update algorithm `createCCState`
Input: Cross-chain rules $\mathcal{R}$
Input: Cross-chain state $\mathcal{S}$
Output: Upon success returns cross-chain state $\mathcal{S}$, and a *SYNC MOVE*

require `verifySatisfability`$(e, \mathcal{R}, \mathcal{S})$ // Returns tuple (event, *MOVE ON LOG*) if event do not conform to the rules, cross-chain state is invalid.
foreach $e \in \mathcal{E}$ **do**
  // For each event in retrieved event set
  if $\nexists \mathcal{S}[e.caseID]$ **then**
    // each cross-chain state key is indexed by case ID.
    $cc =$ `populateCCTX`$(\mathcal{S}[e.caseID], e)$
  end if
  else
    $cc =$ `updateCCTx`$(\mathcal{S}[e.caseID], e)$
  end if
  $\mathcal{S} = \mathcal{S} \cup cc$
  $\mathcal{S}' =$ `createCCState`$(\mathcal{S}, e.caseID)$
  // Calculates updated ccstate, algorithm is parametrizable
end foreach
return ($\mathcal{S}'$, SYNC MOVE)

Our state is at position X. Each time a tx happens, we update the state

Check for non-modelled behavior

Update the state of the cross-chain model

Valid move = cross-chain rules are being respected

Capabilities of a Cross-Chain Model

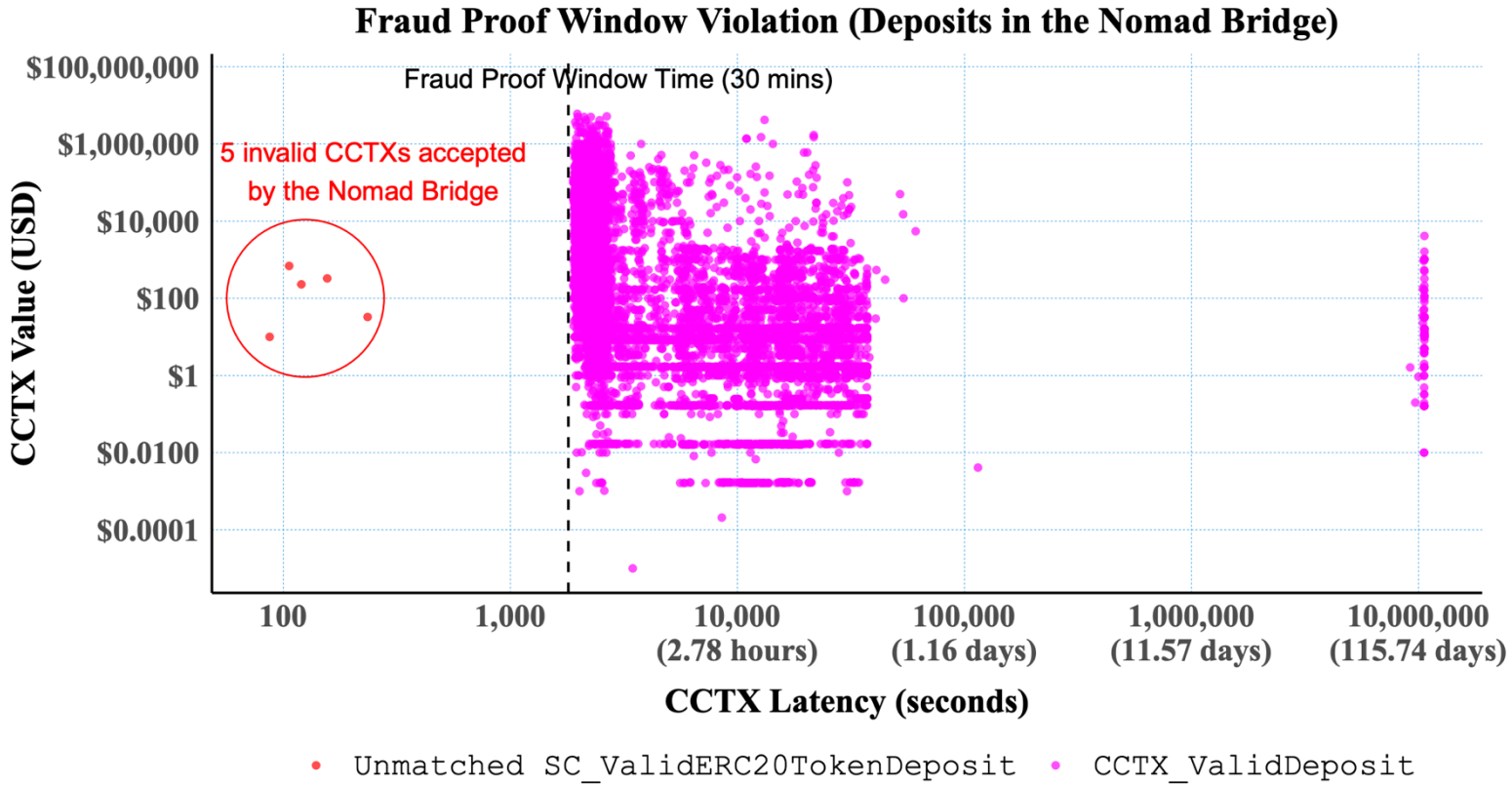
Finding anomalies in cross-chain protocols through **cross-chain rules**.

Example: defining what a valid deposit of tokens should look like

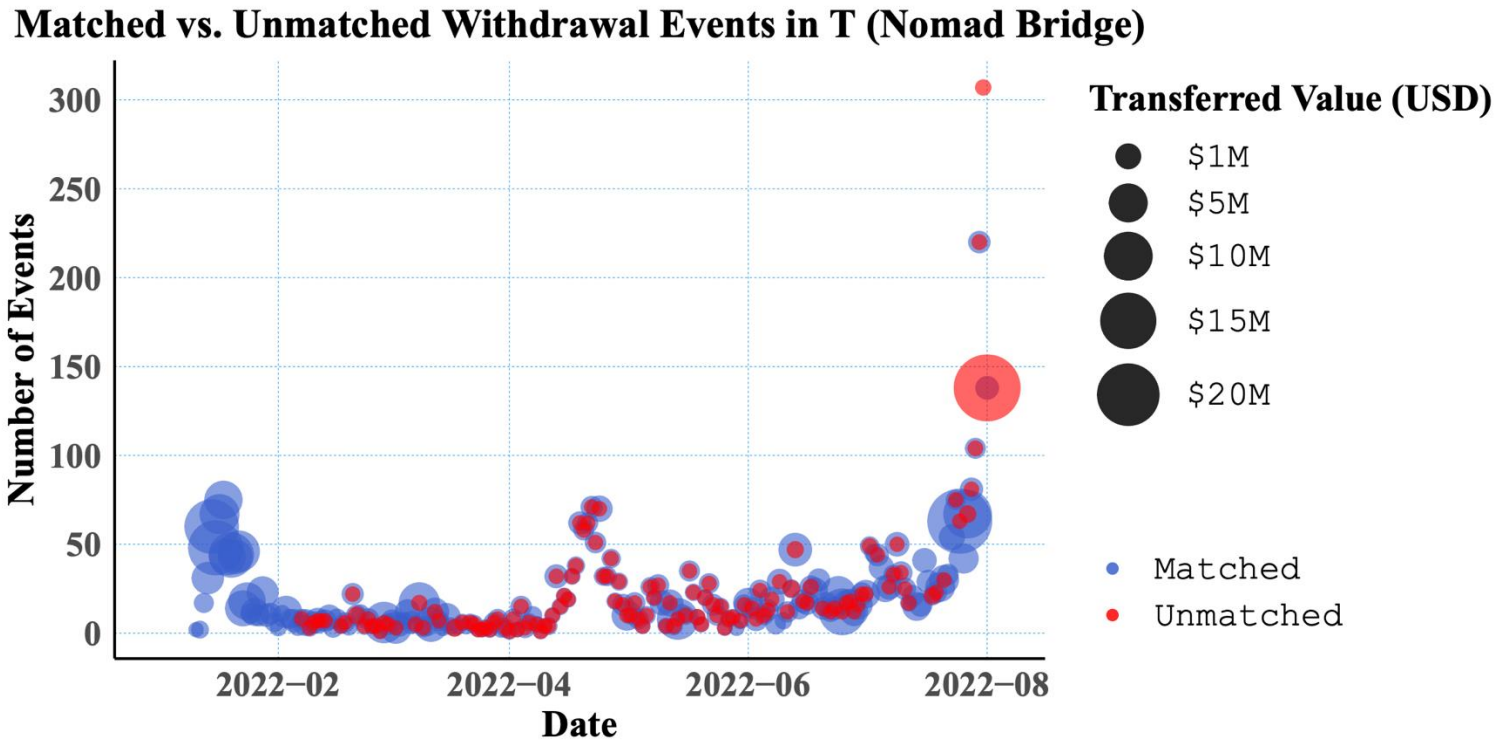
```
// Rule 4 (D)
CCTX_ValidDeposit(orig_chain_id, orig_timestamp, orig_tx_hash, dst_chain_id, dst_tim
     orig_token, dst_token, sender, benef, amount) :-
   TC_ValidERC20TokenDeposit(dst_timestamp, dst_tx_hash, deposit_id, benef, dst_toke
   (
     SC_ValidERC20TokenDeposit(orig_timestamp, orig_tx_hash, deposit_id, sender, _,
     orig_chain_id, dst_chain_id, _, amount) ;
     SC_ValidNativeTokenDeposit(orig_timestamp, orig_tx_hash, deposit_id, sender, _,
     orig_chain_id, dst_chain_id, _, amount)
   ),
   cctx_finality(orig_chain_id, orig_chain_finality),
   orig_timestamp + orig_chain_finality < dst_timestamp.
```

Augusto, A., Belchior, R., Pfannschmidt, J., Vasconcelos, A., & Correia, M. (2024). XChainWatcher: Monitoring and Identifying Attacks in Cross-Chain Bridges. *arXiv preprint arXiv:2410.02029*.

Anomaly 1


Fraud Proof Window Violation (Deposits in the Nomad Bridge)
Fraud Proof Window Time (30 mins)
5 invalid CCTXs accepted by the Nomad Bridge
CCTX Value (USD)
$100,000,000
$1,000,000
$10,000
$100
$1
$0.0100
$0.0001
100
1,000
10,000 (2.78 hours)
100,000 (1.16 days)
1,000,000 (11.57 days)
10,000,000 (115.74 days)
CCTX Latency (seconds)
Unmatched SC_ValidERC20TokenDeposit
CCTX_ValidDeposit

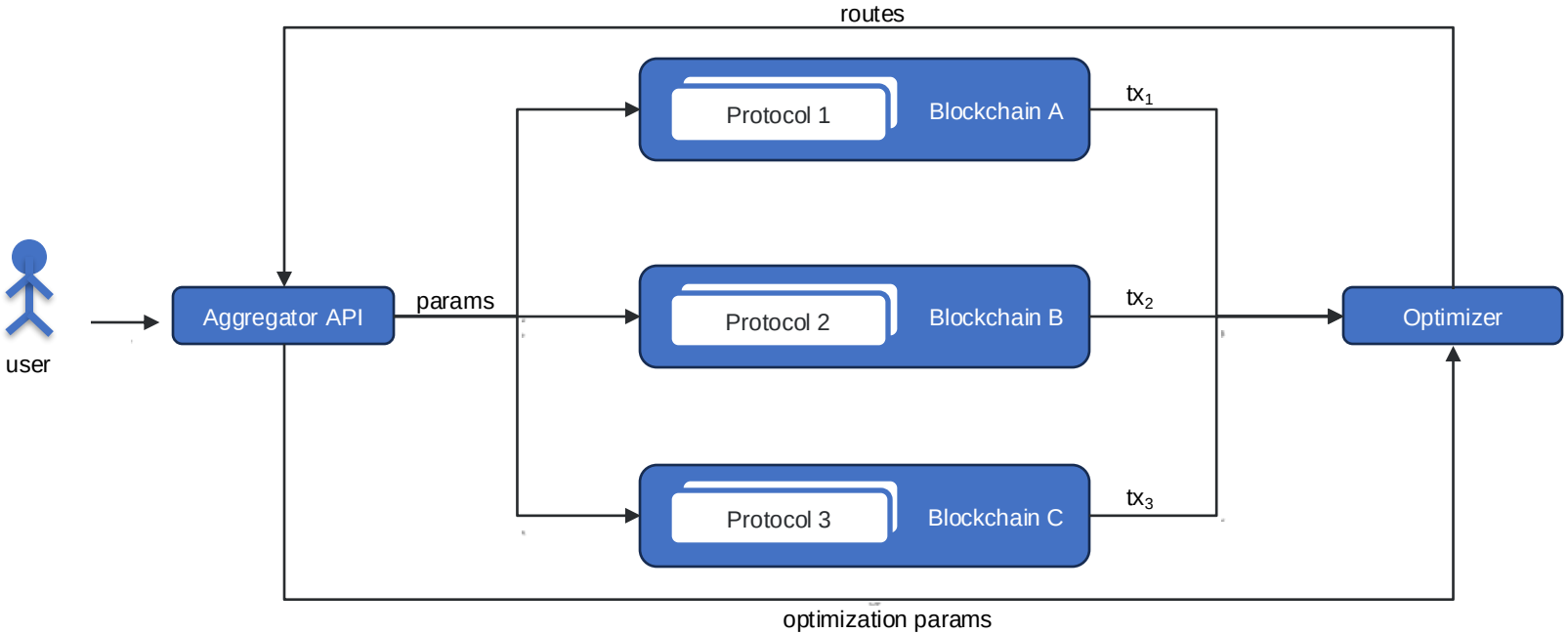
Anomaly 2


Matched vs. Unmatched Withdrawal Events in T (Nomad Bridge)
Number of Events
300
250
200
150
100
50
0
2022–02
2022–04
2022–06
2022–08
Date
Transferred Value (USD)
$1M
$5M
$10M
$15M
$20M
Matched
Unmatched

Outline

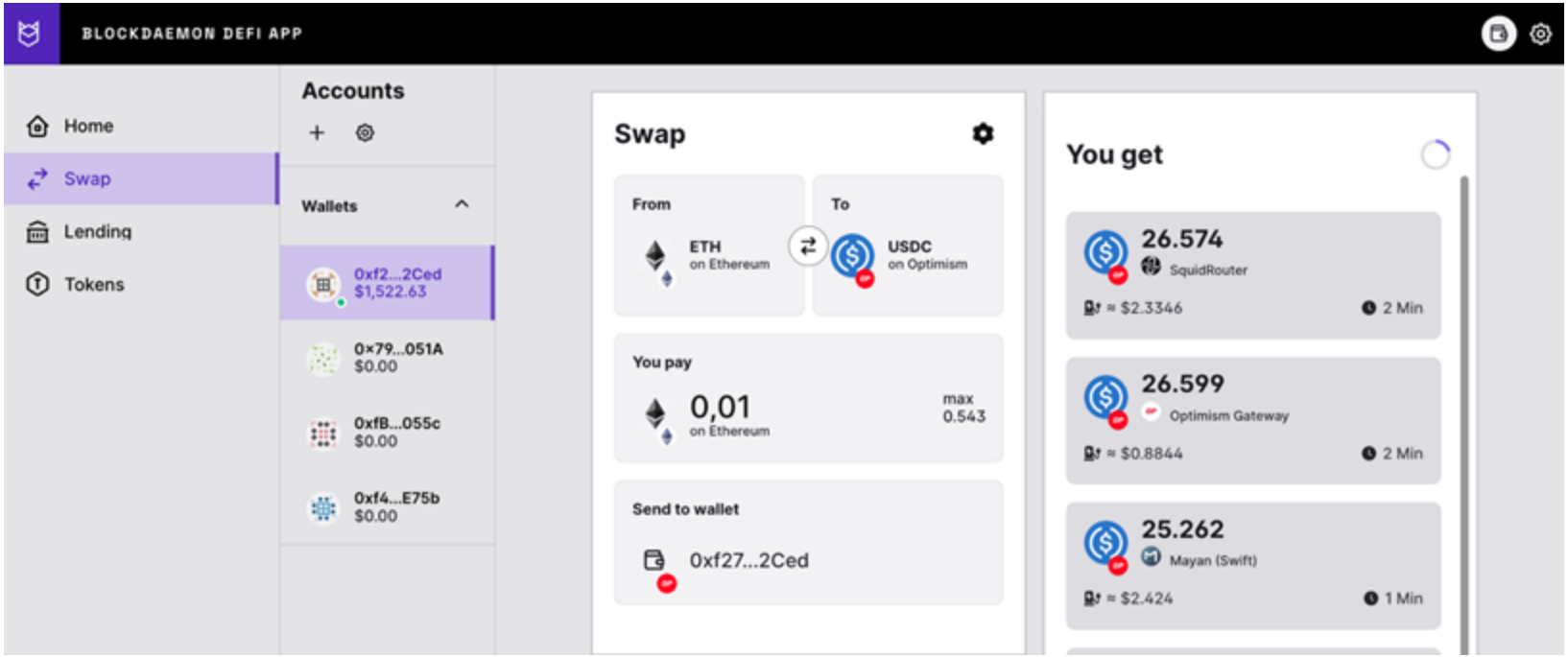
- ~~Motivation (Why?, How?, What?)~~
- ~~Blockchain Interoperability and Interoperability Mechanisms~~
- ~~Security and Privacy of Interoperability Mechanisms~~
- ~~Securing interoperability solutions: Hephaestus and XChainWatcher~~
- Future Research Directions

Bridge Aggregators

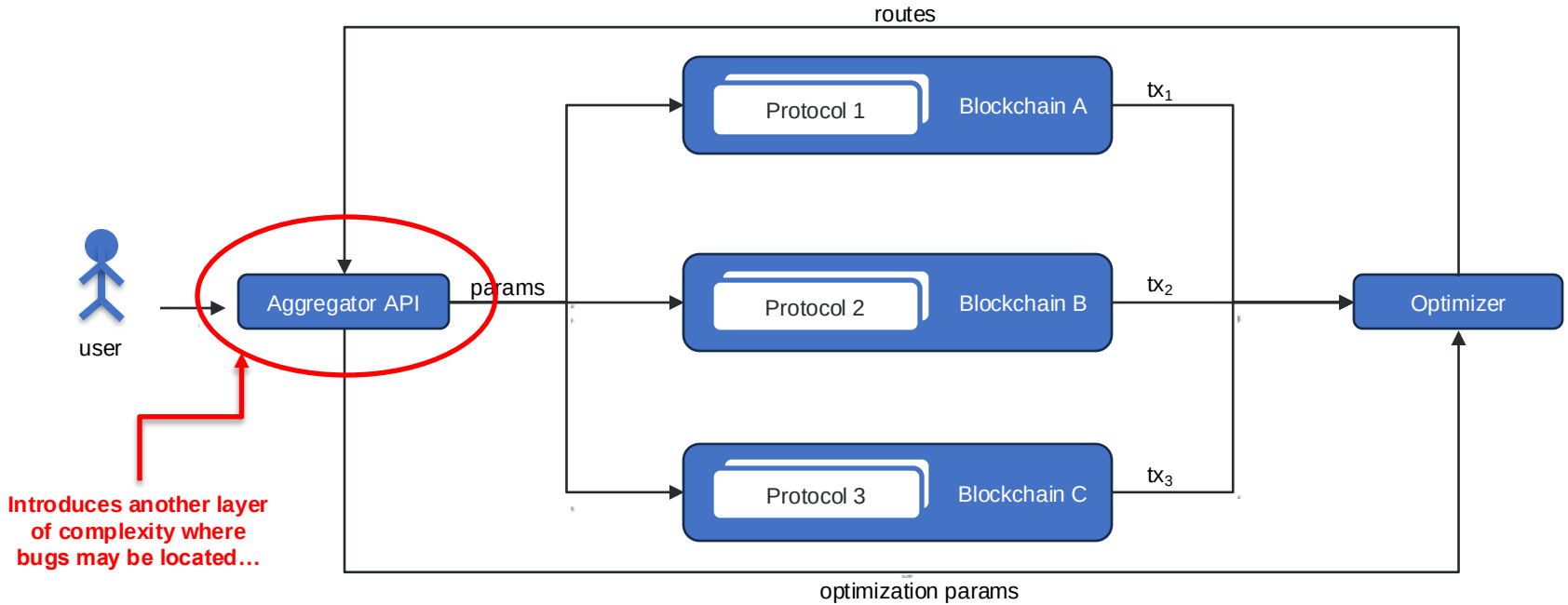


Subramanian, Shankar, et al. "Benchmarking blockchain bridge aggregators." *2024 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2024.

Bridge Aggregators (Example)



Bridge Aggregators

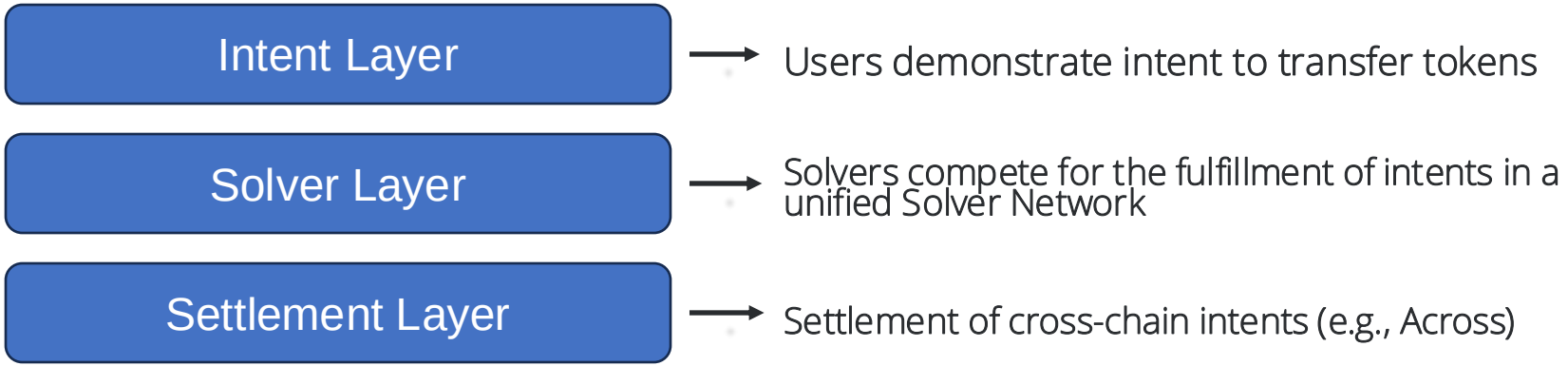


Subramanian, Shankar, et al. "Benchmarking blockchain bridge aggregators." *2024 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2024.

ERC-7683 Cross-Chain Intents

Focus on user experience, fulfilling immediately users' orders

Shift risk to a 'Network of Solvers'



Current Interoperability Challenges

Weak monitoring of
cross-chain solutions

Layer 2s are
majorly **centralized**

Sometimes **large time windows** to
withdraw funds (e.g., 7 days)

Awful user experience when
interacting **with cross-chain protocols**

Standardization Efforts

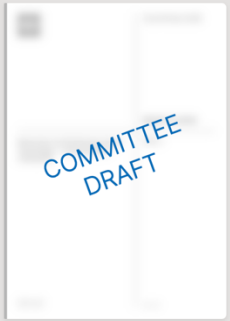
EEA Distributed Ledger Technology Interoperability Specification Version 1.0

EEA Publication 19 September 2024

ENTERPRISE ETHEREUM ALLIANCE

This Version:

https://entethalliance.org/specs/dlt-interop/v1/



ISO/CD TS 23516

Blockchain and Distributed Ledger Technology — Interoperability Framework

Under development

A draft is being reviewed by the committee.

Understudied Interoperability Layers

https://interoperable-europe.ec.europa.eu/collection/nifo-national-interoperability-framework-observatory/solution/eif-toolbox/6-interoperability-layers

Materials for further studying

Hyperledger Cacti workshop (3h) - https://www.youtube.com/watch?v=TM-dnP2yzRM&t=4410s

DLT Interoperation: Implementing IETF Secure Asset Transfer Protocol in Hyperledger Cacti: https://www.youtube.com/watch?v=hmkK2lxhhFw

R. Belchior et al., “A Brief History of Blockchain Interoperability” Communications of the ACM (CACM), 2024 - https://dl.acm.org/doi/pdf/10.1145/3648607

M. Hargreaves et al., “Secure Asset Transfer Protocol (SATP)”, Internet Engineering Task Force Internet Draft draft-ietf-satp-core-04, May 2024 - IETF draft

References

Belchior, R., Riley, L., Hardjono, T., Vasconcelos, A., & Correia, M. (2023). Do you need a distributed ledger technology interoperability solution?. *Distributed Ledger Technologies: Research and Practice*, *2*(1), 1-37.

Belchior, R., Vasconcelos, A., Correia, M., & Hardjono, T. (2022). Hermes: Fault-tolerant middleware for blockchain interoperability. Future Generation Computer Systems, 129, 236-251.

Belchior, R., Dimov, D., Karadjov, Z., Pfannschmidt, J., Vasconcelos, A., & Correia, M. (2023). Harmonia: Securing cross-chain applications using zero-knowledge proofs. Authorea Preprints.

Belchior, R., Somogyvari, P., Pfannschmidt, J., Vasconcelos, A., & Correia, M. (2023). Hephaestus: Modeling, analysis, and performance evaluation of cross-chain transactions. *IEEE Transactions on Reliability*.

Belchior, R., Vasconcelos, A., Guerreiro, S., & Correia, M. (2021). A survey on blockchain interoperability: Past, present, and future trends. Acm Computing Surveys (CSUR), 54(8), 1-41.

Augusto, A., Belchior, R., Correia, M., Vasconcelos, A., Zhang, L., & Hardjono, T. (2024, May). Sok: Security and privacy of blockchain interoperability. In *2024 IEEE Symposium on Security and Privacy (SP)* (pp. 3840-3865). IEEE.

Augusto, A., Belchior, R., Pfannschmidt, J., Vasconcelos, A., & Correia, M. (2024). XChainWatcher: Monitoring and Identifying Attacks in Cross-Chain Bridges. *arXiv preprint arXiv:2410.02029*.

Subramanian, S., Augusto, A., Belchior, R., Vasconcelos, A., & Correia, M. (2024, August). Benchmarking blockchain bridge aggregators. In 2024 IEEE International Conference on Blockchain (Blockchain) (pp. 37-45). IEEE.

Appendix

Interoperability can take multiple forms

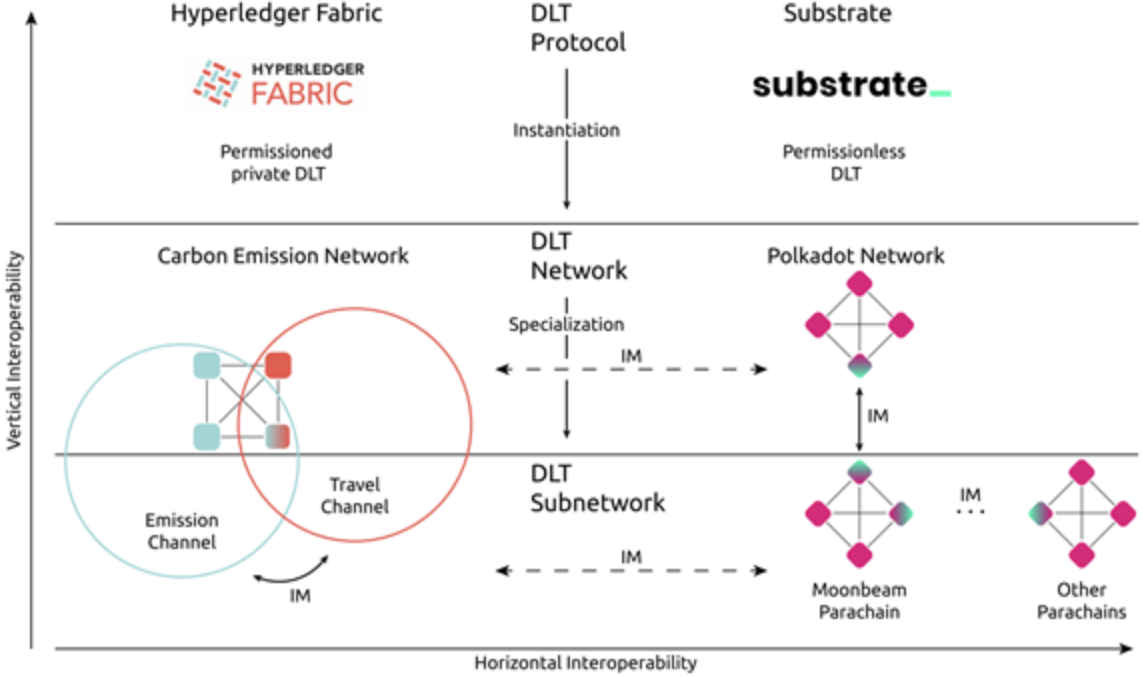
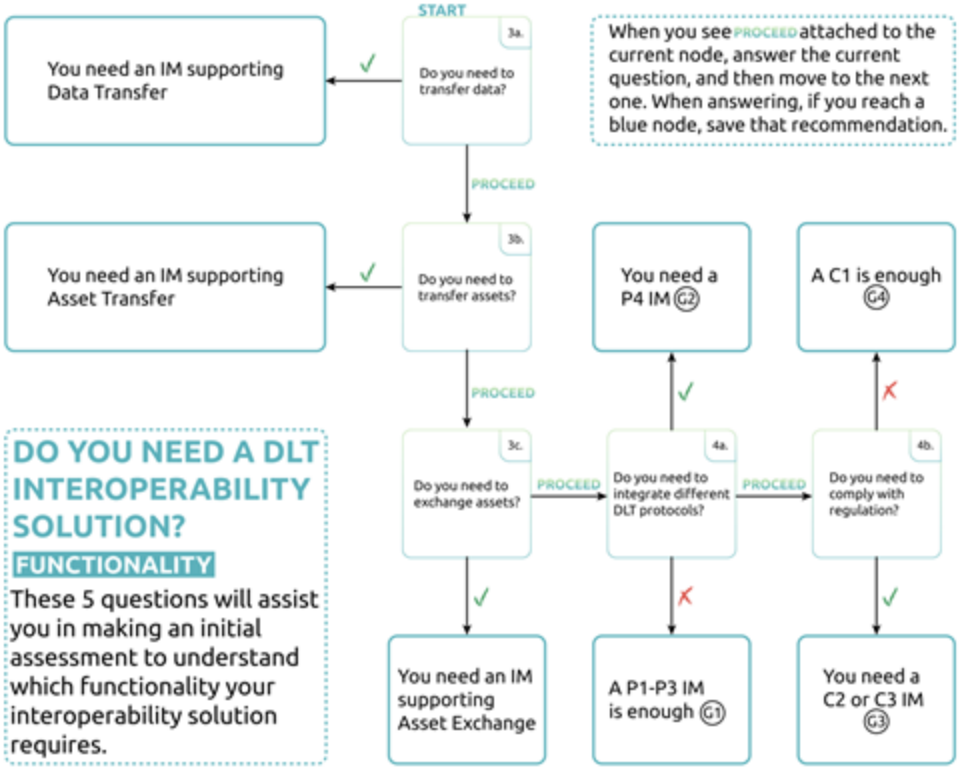
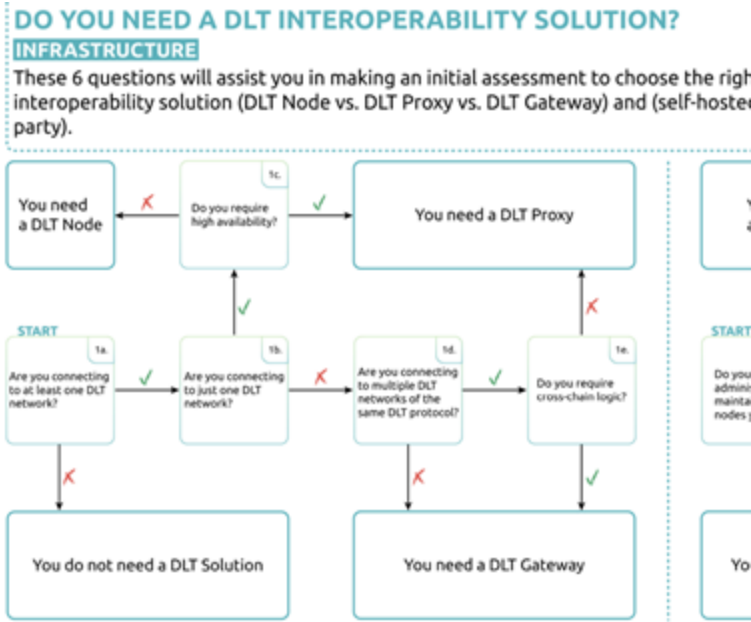


Fig. 2. DLT protocols, networks, and subnetworks.

Do you need an interoperability solution?





Interoperability Assessment

Table 3. DLT Interoperability Solution Assessment

Potentiality Assessment (PA)	**Score (0–4)**
P1: Interoperation within the same DLT network, same subnetworks	☐
P2: Interoperation within the same DLT network, different subnetworks	☐
P3: Interoperation within different DLT networks	☐
P4: Interoperation within different DLT protocols	☐
Compatibility Assessment (CA)	**Score (0–3)**
C1: Provides semantic-level interoperability (shared protocols)	☐
C2: Provides organization-level interoperability (shared agreements)	☐
C3: Provides legal-level interoperability (follow regulations)	☐
Performance Assessment (PeA)	**Score (0–3)**
PE1: Provides acceptable cross-chain transaction end-to-end latency/throughput	☐
PE2: Provides acceptable cross-chain transaction end-to-end cost	☐
PE3: Complies with desirable energetic consumption goals	☐
PA + CA + PeA	**Total (0–10):**

Interoperability assessment is divided into PE, CA, and PeA assessments. A higher score corresponds to a more interoperable solution.

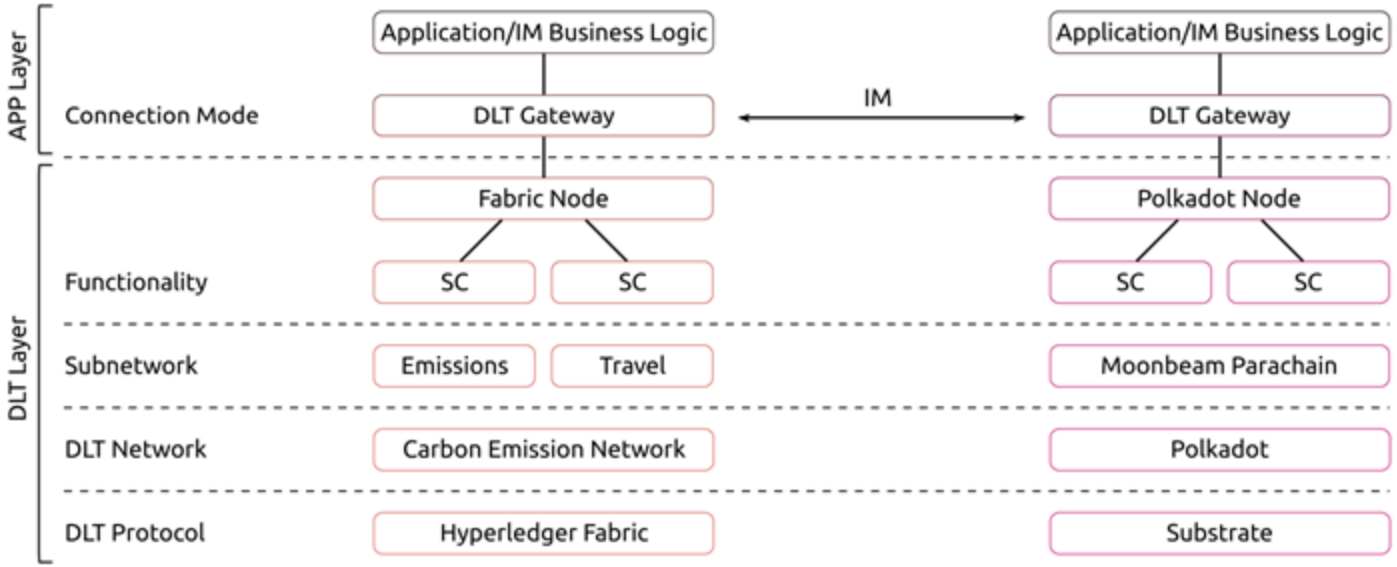
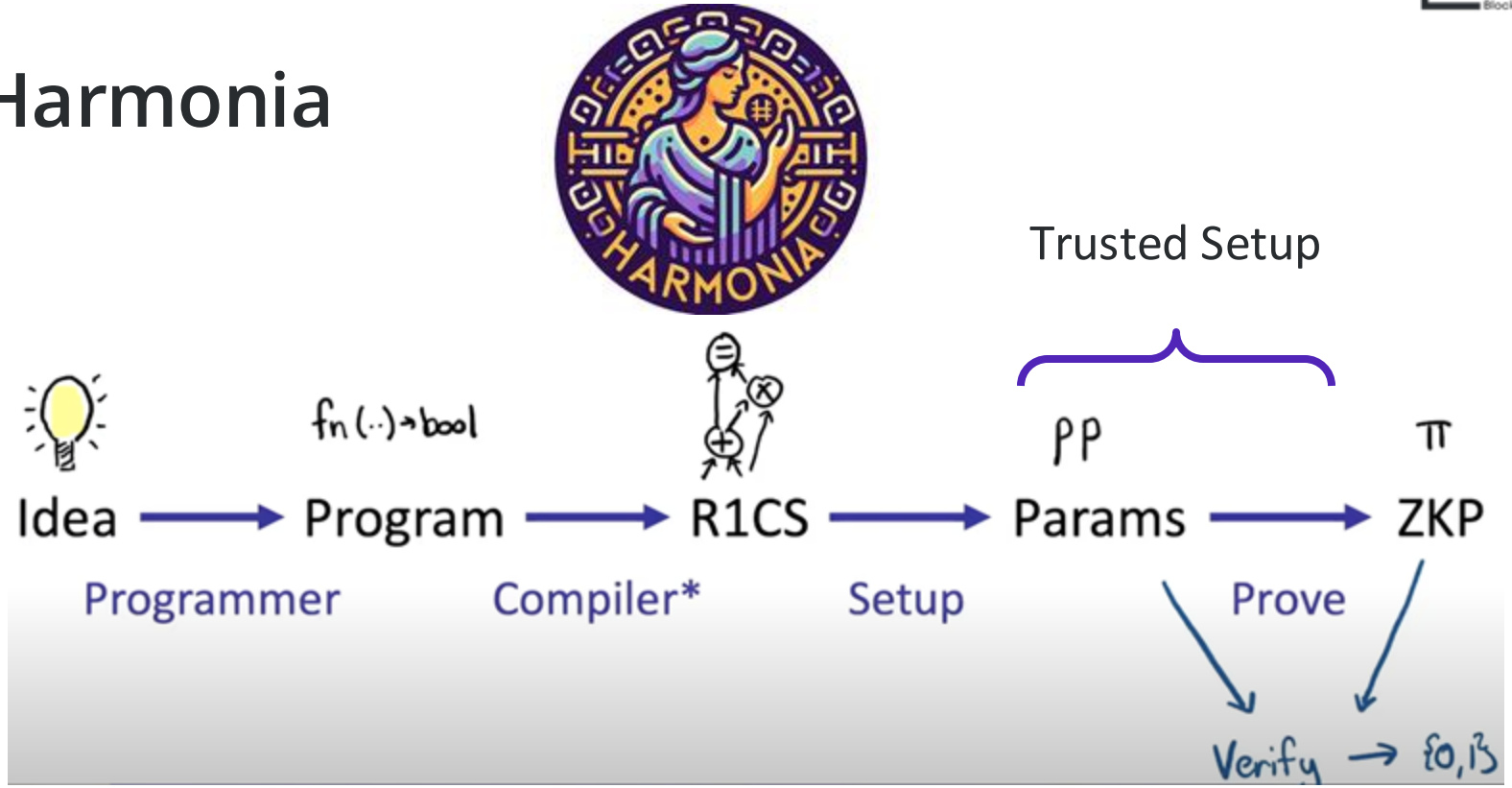


Fig. 12. Example of vertical interoperation in a Hyperledger Fabric network and the Polkadot network. Horizontal interoperability can be achieved via an IM using, for example, a DLT gateway.

Harmonia

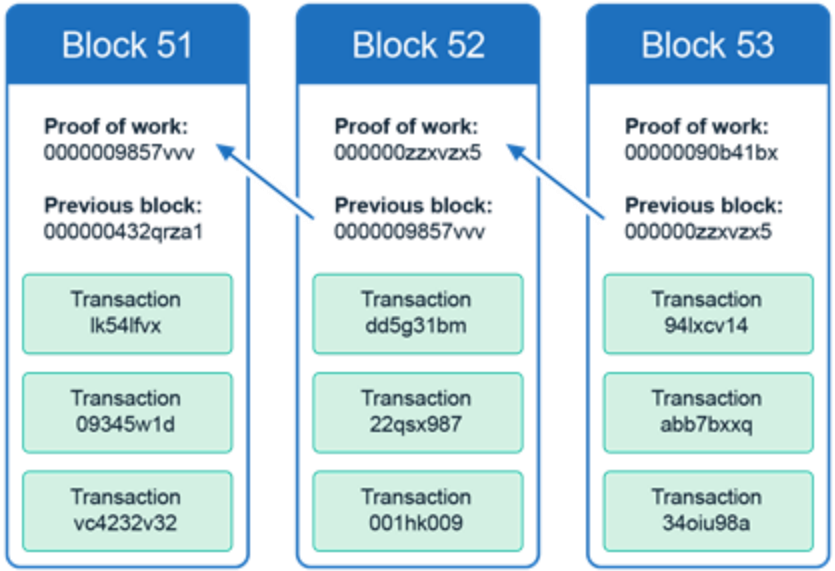


source : https://www.youtube.com/watch?v=UpRSaG6iuks

Key Idea

PART 1

1. ***Obtain block*** of interest from the source chain
2. Prove that the current block is valid according to a set of rules (**generate ZKP**)
3. Valid block gives us a ***provably valid block root -> R***

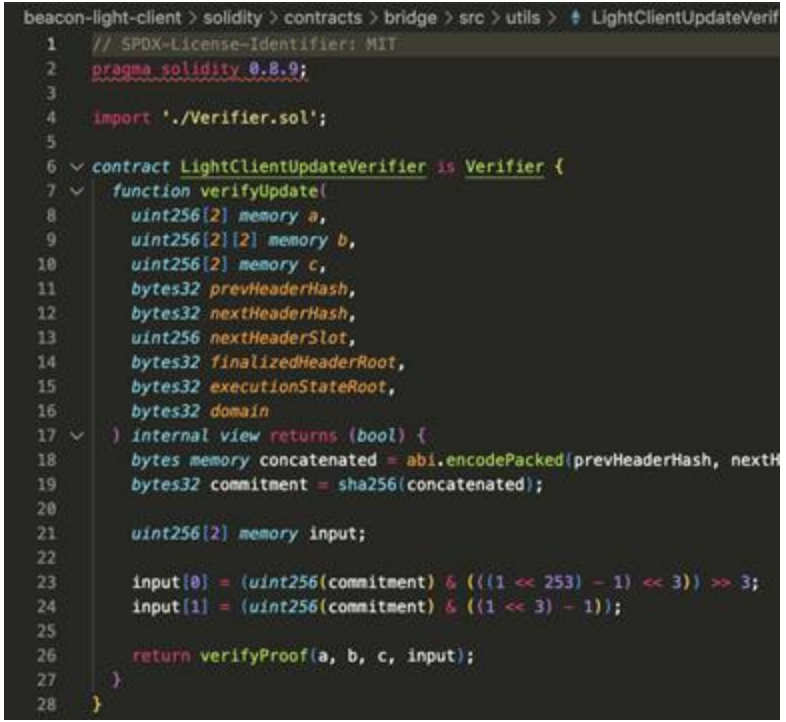


Key Idea

PART 2

1. Create a ***Merkle proof*** on the source chain
2. ***Propagate that proof to target chain*** via a relayer
3. **Verify** the merkle proof using the provably valid block root ***(Part 1, R)***

PART 3 (*business logic, Part 1 + Part 2)*



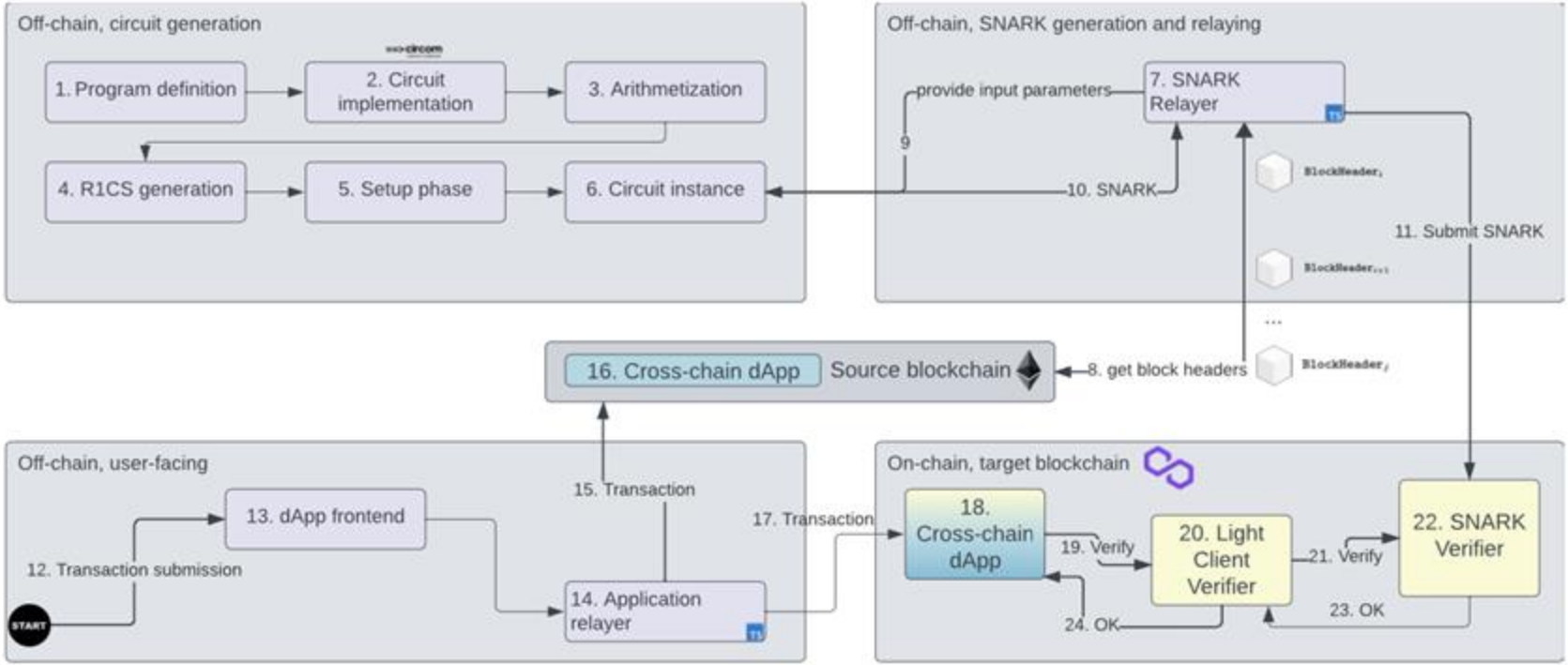


Fig. 2: Architecture of Harmonia. Off-chain components are represented by ▭. Destination chain components ▭. The cross-chain contract is represented by ▭.

R1CS is generated by circom.

We feed the circuit into the zero knowledge proof system setup algorithm (generates public parameters). P

roofs are generated by RapidSNARK upon a certain input (previous beacon block header and a new light client update data structure).

Such proofs can be sent as a transaction to the Verifier.sol smart contract, that, upon verification, updates our light client state..

beacon-light-client > solidity > contracts > bridge > src > utils > LightClientUpdateVerif

```
1   // SPDX-License-Identifier: MIT
2   pragma solidity 0.8.9;
3
4   import './Verifier.sol';
5
6   contract LightClientUpdateVerifier is Verifier {
7     function verifyUpdate(
8       uint256[2] memory a,
9       uint256[2][2] memory b,
10      uint256[2] memory c,
11      bytes32 prevHeaderHash,
12      bytes32 nextHeaderHash,
13      uint256 nextHeaderSlot,
14      bytes32 finalizedHeaderRoot,
15      bytes32 executionStateRoot,
16      bytes32 domain
17    ) internal view returns (bool) {
18      bytes memory concatenated = abi.encodePacked(prevHeaderHash, nextH
19      bytes32 commitment = sha256(concatenated);
20
21      uint256[2] memory input;
22
23      input[0] = (uint256(commitment) & (((1 << 253) - 1) << 3)) >> 3;
24      input[1] = (uint256(commitment) & ((1 << 3) - 1));
25
26      return verifyProof(a, b, c, input);
27    }
28  }
```

Harmonia - Zero Knowledge Proof Generation

The program is "process a light client update by applying all light client syncing rules"

(e.g., merkle branch of finality header is valid, merkle branch of next sync committee is valid, validate aggregated BLS signatures from the light sync committee, compute certain sync committee period at slot X).

Full set of rules here: https://github.com/ethereum/consensus-specs/blob/dev/specs/altair/light-client/sync-protocol.md

if the operator can supply a proof of correct execution for a particular update, one can be certain that all the rules were followed and that the end state is indeed what the operator claims, namely:

-execution state root
-finalized state root
-optimistic state root

Blockchain Interoperability

André Augusto, INESC-ID, Técnico Lisboa
Rafael Belchior, INESC-ID, Blockdaemon

DAY 5 - February 14th, 2025

BIG – enhancing the research and innovation potential of Técnico - Lisbon through **B**lockchain technologies and design **I**nnovation for social **G**ood
Grant agreement: 952226