Redactable Distributed Ledger Technology for Hyperledger Fabric

Rick Kuhn

US National Institute of Standards and Technology

kuhn@nist.gov

INTERNATIONAL BLOCKCHAIN SUMMIT ISTANBUL 2022

Key Points – why listen to this talk?

- Blockchain has valuable properties, but conflicts with privacy and exception management – deletion impossible

  ➡ <u>Sometimes we don’t need blockchain</u>,
  just some blockchain features

- Data structure called *data block matrix* provides <u>distributed trust, integrity protection of blockchain</u>, but allows <u>controlled edits for privacy, corrections</u>

- Drop-in compatibility for Hyperledger Fabric applications

  ➡ Open source release Dec. 5, 2022

Comparison Summary

Blockchain

- Distributed trust
- Integrity protection through hashing
- Immutable records

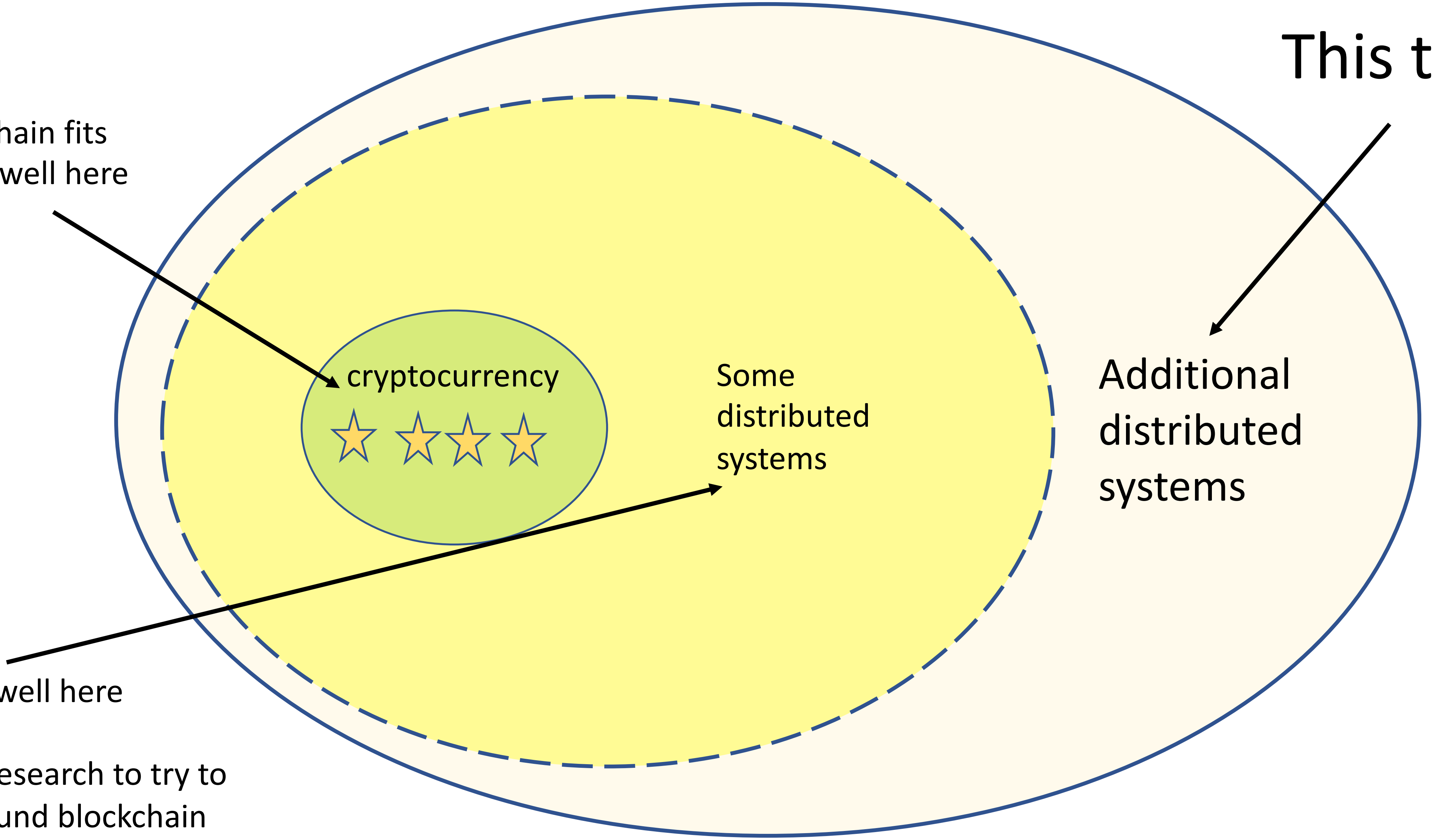
Data block matrix

- Distributed trust
- Integrity protection through hashing
- Redactable/editable records

Market, range of applications for DLT?

This talk here

Blockchain fits pretty well here



cryptocurrency ☆☆☆☆

Some distributed systems

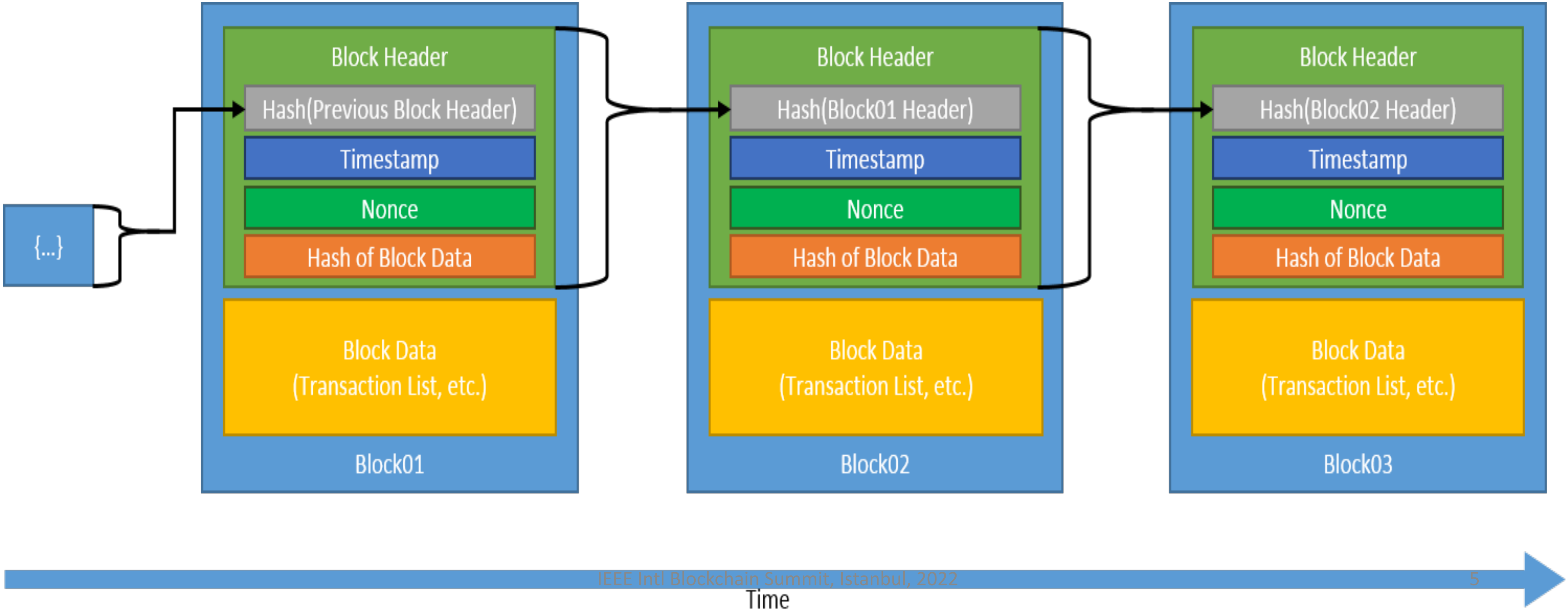
Additional distributed systems

Not as well here

Much research to try to get around blockchain properties

Structure of a Traditional Blockchain

Blockchain has been defined as "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way".



Why is immutability a problem for privacy?

- Permanence/immutability conflicts with ‘right to erasure’ privacy regulations
- Privacy rules such as European Union General Data Protection Regulation (GDPR) require that all information related to a particular person can be deleted at that person's request
  - *personal* data, defined as "any information concerning an identified or identifiable natural person" - data for which blockchains are designed
  - "Personal data which have undergone pseudonymisation, which could be attributed to a natural person by the use of additional information should be considered to be information on an identifiable natural person."
- US states adopting similar privacy rules – California, Virginia, Colorado
- US law enforcement also requires deletion of data in some cases

What’s been tried to solve blockchain/privacy conflict?

- Don’t put personal data on blockchain
  - Pseudo-anonymized data are still considered personal
  - Even if not directly tied to a person – dynamic IP address can be considered personal if it can be indirectly tied
  - Financial transactions are obviously personal data

- Encrypt data and destroy key to delete
  - Data must be secure permanently or for decades
  - Advancements in cryptography usually compromise old crypto – e.g., quantum computing puts current public key systems at risk

Redactable/editable blockchains

- Chameleon hash function - most common approach to providing editability
  - Intentionally generate a collision for the hash, given trapdoor or key, changing data but hash not disturbed
  - Standard blockchain for integrity protection
  - Requires specialized chameleon hash function
- Our approach, data block matrix
  - Dual hash list for integrity protection
  - Use standard hash function (SHA 256)
- Either may be best, depending on application requirements
  - *Tradeoffs like any other engineering problem*
  - Configurable option for Hyperledger Fabric

Public blockchain works for cryptocurrency
What about supply chain, logistics, etc.?

Cryptocurrency	Business, logistics, supply chain, e-commerce, etc.
1. Partial anonymity	ID required for contracts or government regulation
2. Public access/transparency	Controlled access
3. Small transaction size	Range of message sizes up to large documents, images
4. Immutable records	Changes and deletions, often required by law
5. Proof of work	Flexible consensus models
6. Block ordering guarantees	Timestamps often required
7. Decentralization	Same in many applications
8. Replication	Same in many applications
9. Data integrity guarantees	Same in many applications

Most
requirements
are different

What if we keep the useful blockchain features, but remove the immutability constraint?

Datablock matrix – uses two hash values per block instead of a linked chain

- Java or Go example code available as open source
- Incorporated into Next Gen Access Control – practical demo
- NOT to replace blockchain, to provide alternative tools for distributed system design
- Hyperledger Fabric component completed, available as open source Dec. 5, 2022

What are blockmatrix constraints and assumptions?

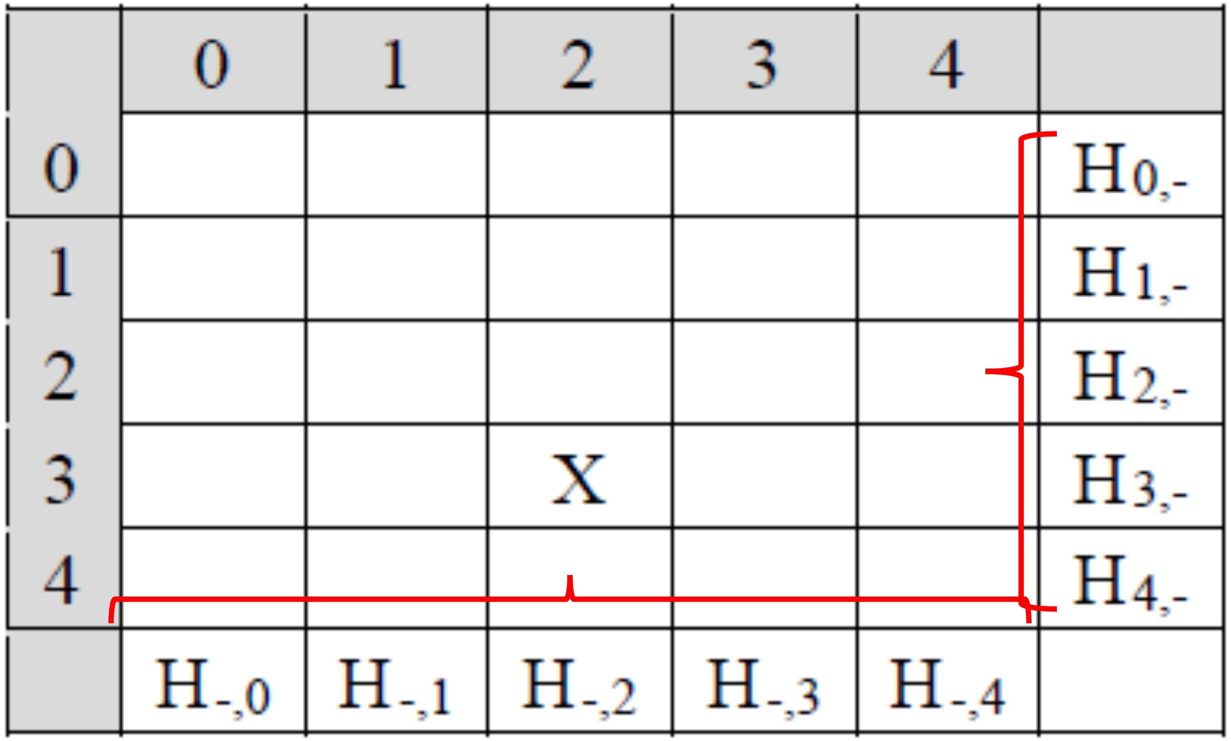
- Hash integrity protection must not be disrupted for blocks not deleted
- Ensure auditability and accountability – distributed trust
- Provide distributed consensus and guaranteed shared view

Designed for permissioned/private distributed ledger systems – such as supply chain, logistics
(not exciting as cryptocurrency but economic importance)

Datablock matrix data structure

- A data structure that provides integrity assurance using hash-linked records while also allowing the deletion of records

- Stores hashes of each row and column
- => each block within the matrix is protected by two hashes
- Suggested use for private/permissioned distributed ledger systems



	0	1	2	3	4	
0						$H_{0,-}$
1						$H_{1,-}$
2						$H_{2,-}$
3			X			$H_{3,-}$
4						$H_{4,-}$
	$H_{-,0}$	$H_{-,1}$	$H_{-,2}$	$H_{-,3}$	$H_{-,4}$	

Figure 1. Block matrix

How does this work?

- Suppose we want to delete block 12

- disrupts the hash values of $H_{3,-}$ for row 3 and $H_{-,2}$ and column 2
- blocks of row 3 are included in the hashes for columns 0, 1, 3, and 4
- blocks of column 2 are included in the hashes for rows 0, 1, 2, and 4

	0	1	2	3	4	
0	●	1	3	7	13	$H_{0,-}$
1	2	●	5	9	15	$H_{1,-}$
2	4	6	●	11	17	$H_{2,-}$
3	8	10	12	●	19	$H_{3,-}$
4	14	16	18	20	●	$H_{4,-}$
	$H_{-,0}$	$H_{-,1}$	$H_{-,2}$	$H_{-,3}$	$H_{-,4}$	etc.

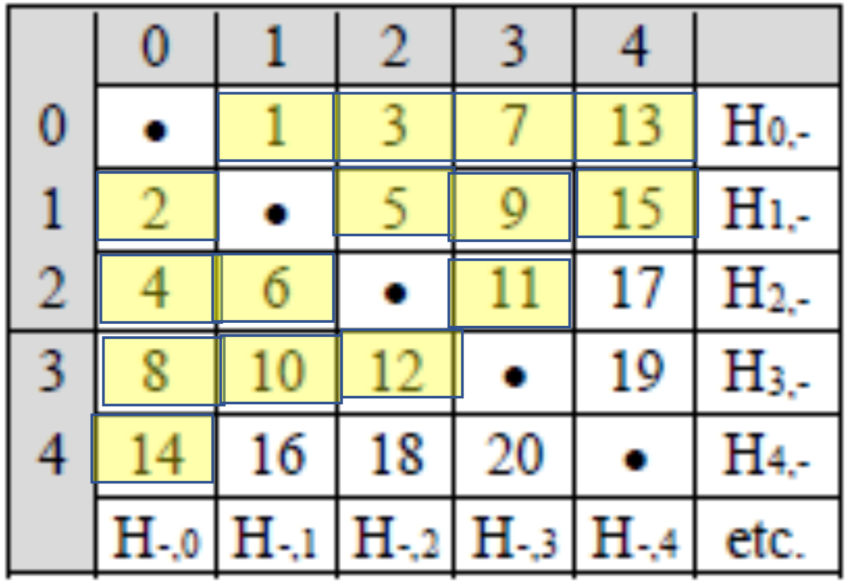
Datablock Matrix Population Algorithm

- Algorithm

```
while (new blocks) {//i,j = row, column indices
  if      (i==j) {add null block; i=0; j++; }
  else if (i<j) {add block(i,j); swap(i,j) }
  else if (i>j) {add block(i,j); j++; swap(i,j)}
}
```

- Basic algorithm is simple, many variations possible

- Block ordering provides desirable properties

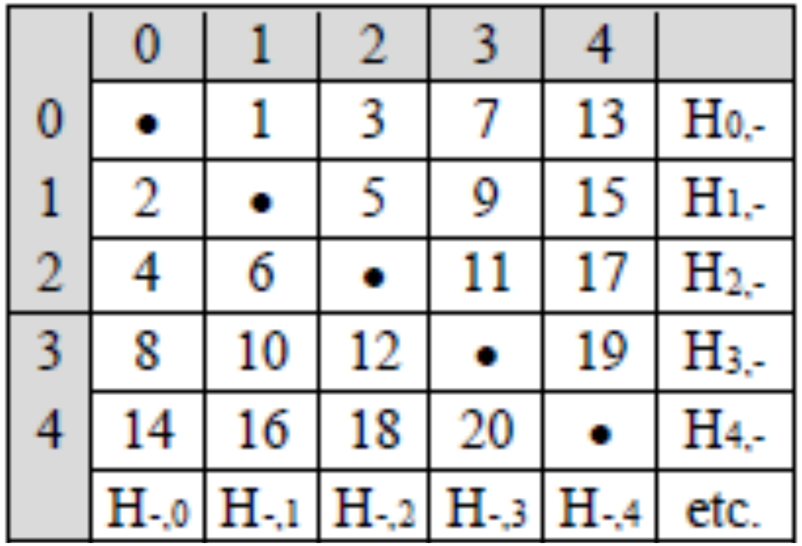


	0	1	2	3	4	
0	●	1	3	7	13	$H_{0,-}$
1	2	●	5	9	15	$H_{1,-}$
2	4	6	●	11	17	$H_{2,-}$
3	8	10	12	●	19	$H_{3,-}$
4	14	16	18	20	●	$H_{4,-}$
	$H_{-,0}$	$H_{-,1}$	$H_{-,2}$	$H_{-,3}$	$H_{-,4}$	etc.

Figure 2. Block matrix with numbered cells

Data Structure Properties

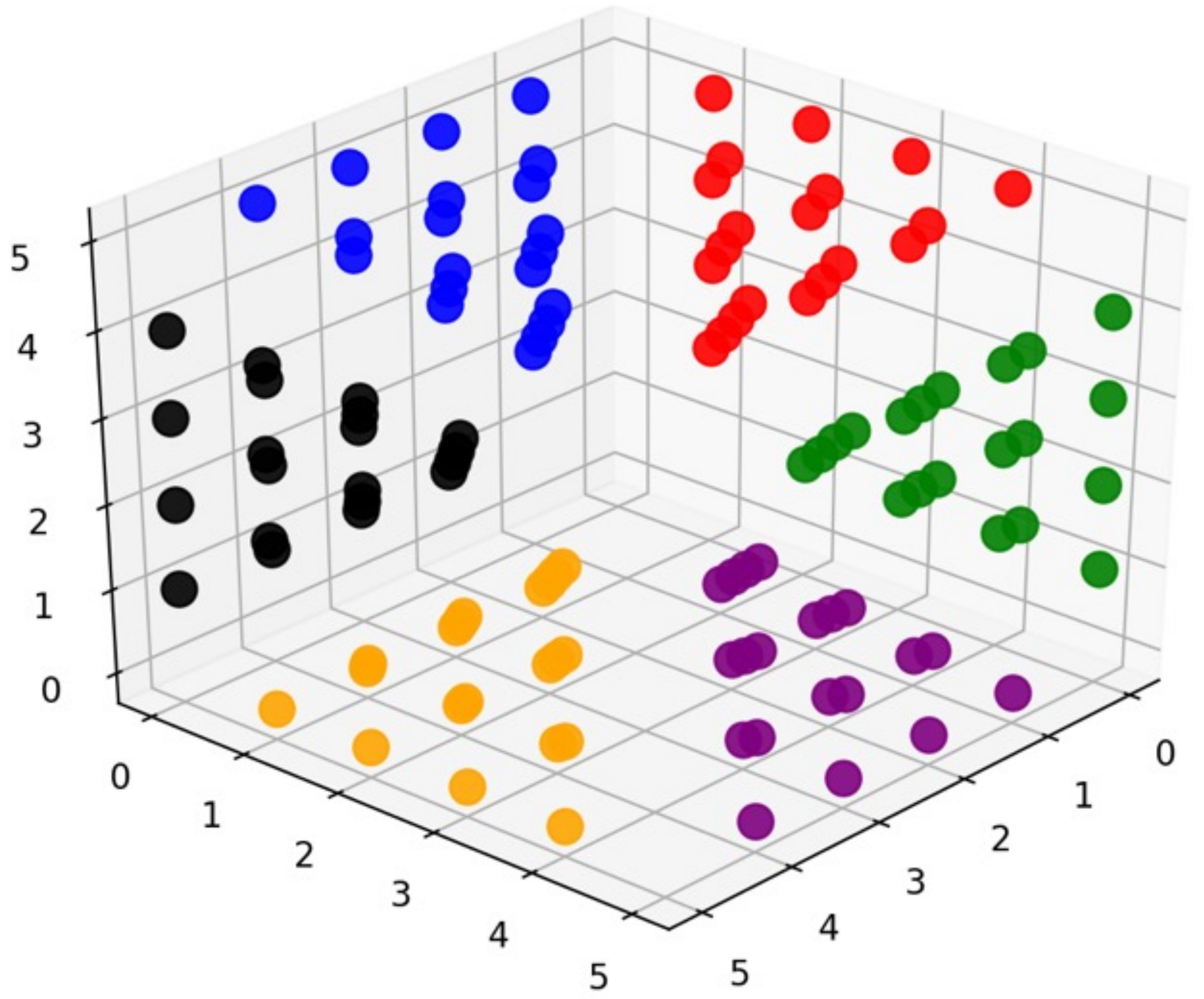
- *Balance*: upper half (above diagonal) contains at most one additional cell more than the lower half.
- *Hash sequence length*: number of blocks in a row or column hash proportional to $\sqrt{N}$ for a matrix with *N* blocks, by the balance property.
- *Number of blocks*: The total number of data blocks in the matrix is $k^2 - k$ for k rows/columns since the diagonal is null.
- *Block dispersal*: No consecutive blocks in same row or column, in sector 0 (below diagonal) or sector 1(above) for *b* mod 2 for block *b*



	0	1	2	3	4	
0	•	1	3	7	13	$H_{0,-}$
1	2	•	5	9	15	$H_{1,-}$
2	4	6	•	11	17	$H_{2,-}$
3	8	10	12	•	19	$H_{3,-}$
4	14	16	18	20	•	$H_{4,-}$
	$H_{-,0}$	$H_{-,1}$	$H_{-,2}$	$H_{-,3}$	$H_{-,4}$	etc.

Structure can be extended to multiple dimensions

- Block dispersal for 3 dimensions
- Location in sectors 0..5 according to *b* mod 6 for block *b*



So what?   Why use this data structure?

Again, many blockchain applications don’t need blockchain, just some features

Enlarge the market for blockchain

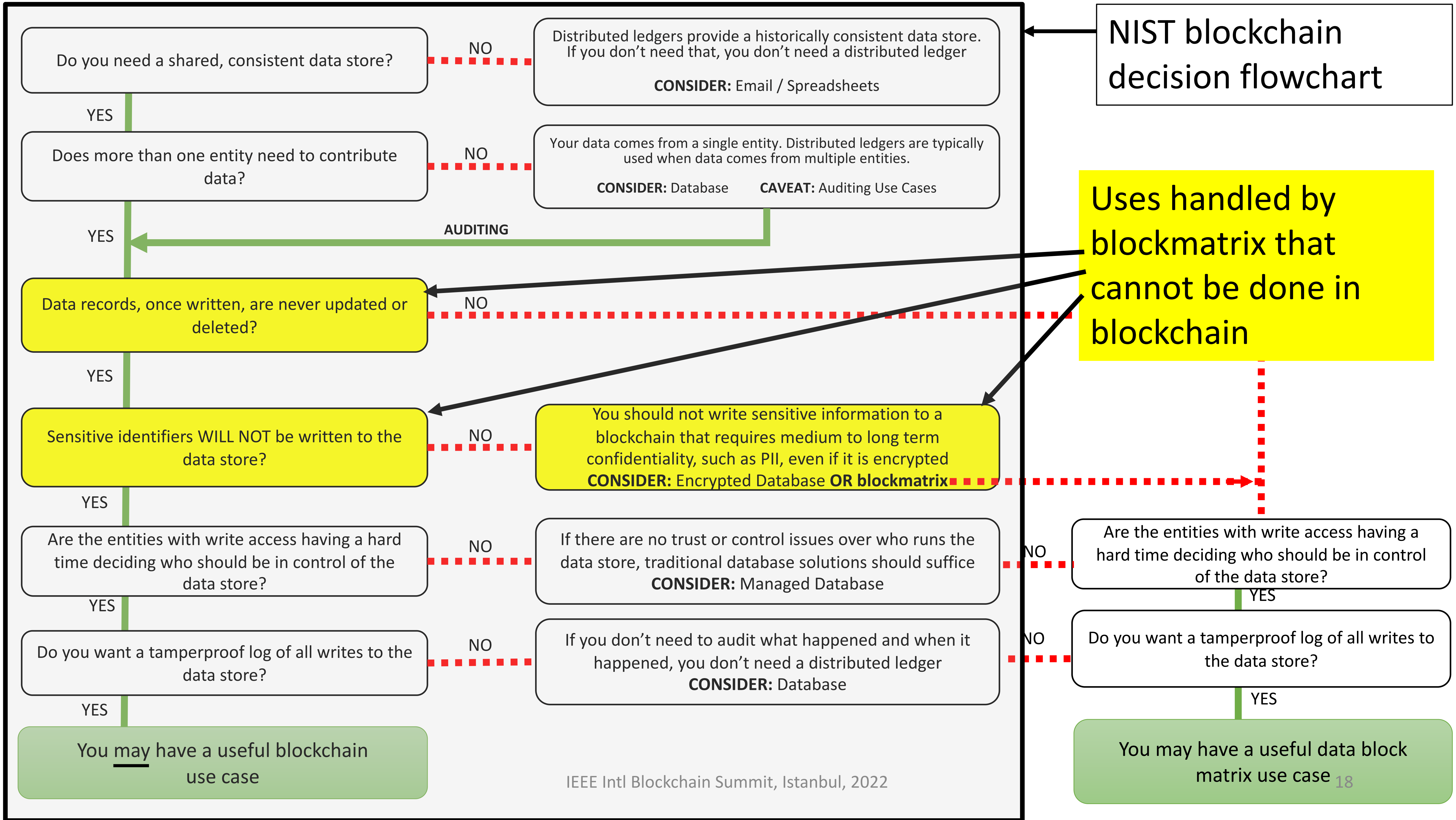
- Solve the conflict between blockchain and privacy regulations
- Allow for exception management

Replace network communication with local data

- You can obviously do this with conventional database functions, but
- New data structure adds integrity checks as in blockchain

Easy-to-use component for distributed database design

NIST blockchain decision flowchart



Do you need a shared, consistent data store?

NO

Distributed ledgers provide a historically consistent data store. If you don't need that, you don't need a distributed ledger

CONSIDER: Email / Spreadsheets

YES

Does more than one entity need to contribute data?

NO

Your data comes from a single entity. Distributed ledgers are typically used when data comes from multiple entities.

CONSIDER: Database **CAVEAT:** Auditing Use Cases

YES

AUDITING

Data records, once written, are never updated or deleted?

NO

YES

Sensitive identifiers WILL NOT be written to the data store?

NO

You should not write sensitive information to a blockchain that requires medium to long term confidentiality, such as PII, even if it is encrypted

CONSIDER: Encrypted Database **OR blockmatrix**

YES

Are the entities with write access having a hard time deciding who should be in control of the data store?

NO

If there are no trust or control issues over who runs the data store, traditional database solutions should suffice

CONSIDER: Managed Database

YES

Do you want a tamperproof log of all writes to the data store?

NO

If you don't need to audit what happened and when it happened, you don't need a distributed ledger

CONSIDER: Database

YES

You may have a useful blockchain use case

Uses handled by blockmatrix that cannot be done in blockchain

Are the entities with write access having a hard time deciding who should be in control of the data store?

NO

YES

Do you want a tamperproof log of all writes to the data store?

NO

YES

You may have a useful data block matrix use case

What about tech transfer?

- We won - NIST Technology Maturation Acceleration Program funding – for technology transfer and commercialization
- Integrating with Next Generation Database Access Control
- Patent approved – assures availability of technology
- Hyperledger component open source

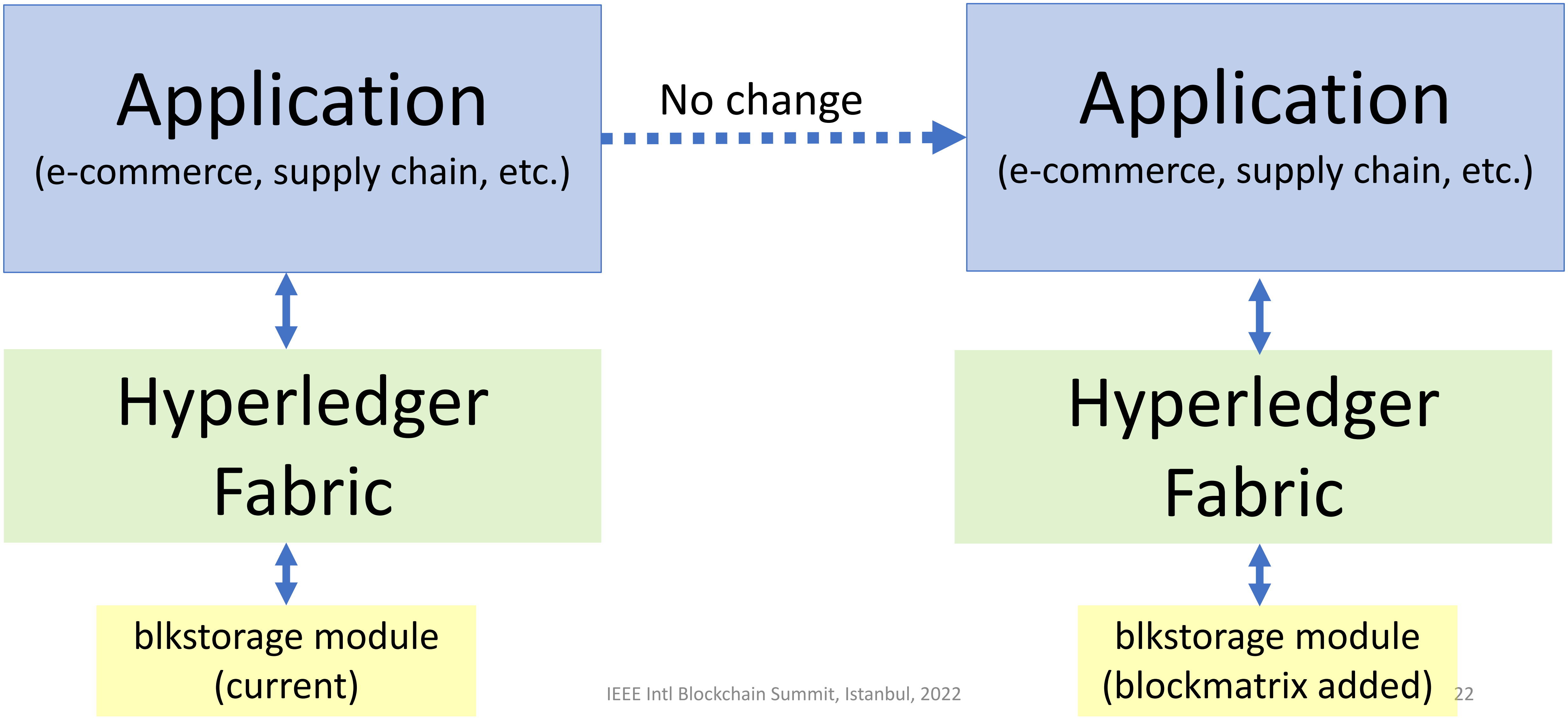
Hyperledger blockmatrix implementation

- Hyperledger is widely-used open source project started by IBM, Intel, and SAP
- Hyperledger Fabric - intended for large distributed systems
- Designed to use existing API as closely as possible – add blocks in same manner as adding to blockchain
- Open source release Dec. 5, 2022
- RED Ledger = Redactable Enhanced Distributed Ledger
- https://csrc.nist.gov/projects/redactable-distributed-ledger

Integration with Hyperledger Fabric

- Minimal code changes
- Changes primarily in <u>blkstorage</u> package, reducing potential for errors, easing future updates
- Blockmatrix is <u>configurable by channel</u> (private subnet)
- Configure to use conventional blockchain or blockmatrix
  - If a deployment uses two channels, one can be a blockchain and the other can be a blockmatrix

Compatible with current Hyperledger applications



Application

(e-commerce, supply chain, etc.)

No change

Application

(e-commerce, supply chain, etc.)

Hyperledger Fabric

Hyperledger Fabric

blkstorage module (current)

blkstorage module (blockmatrix added)

Hyperledger Integration Summary

- Blockmatrix implemented in Hyperledger Fabric, widely used for DLT functions
- Uses existing API for ease of application coding
- Minimal changes to Hyperledger code
- Potential applications include current uses of Hyperledger Fabric – e.g., supply chain and logistics, e-commerce, digital currency – adding privacy support

Future work - where do we go next?

- Support open source release
- Performance evaluation
  - Transaction rate
  - Data volume
- Demonstrate – clinical trials, logistics/supply chain

Other applications?

New European Central Bank report says Hyperledger Fabric fits needs of 'digital euro' – can blockmatrix help ?

More information:

- Kuhn, R., Yaga, D. and Voas, J., 2019. Rethinking Distributed Ledger Technology. *Computer*, *52*(2), pp.68-72.
- Kuhn, D. R. (2018). A Data Structure for Integrity Protection with Erasure Capability. https://csrc.nist.gov/publications/detail/white-paper/2018/05/31/data-structure-for-integrity-protection-with-erasure-capability/draft

Project sites with links to source code and publications

- https://csrc.nist.gov/Projects/enhanced-distributed-ledger-technology
- https://csrc.nist.gov/projects/redactable-distributed-ledger

Acknowledgements

- Josh Roberts, Jeff Voas, Dylan Yaga, Sylvain Chantreau, NIST
- Temur Saidkhodjaev, University of Maryland College Park
- Arsen Klyuev, Johns Hopkins University
- Gökhan Koçak, Asena, Inc.