

# Privacy Enhanced Distributed Ledger Technology with Hyperledger Fabric

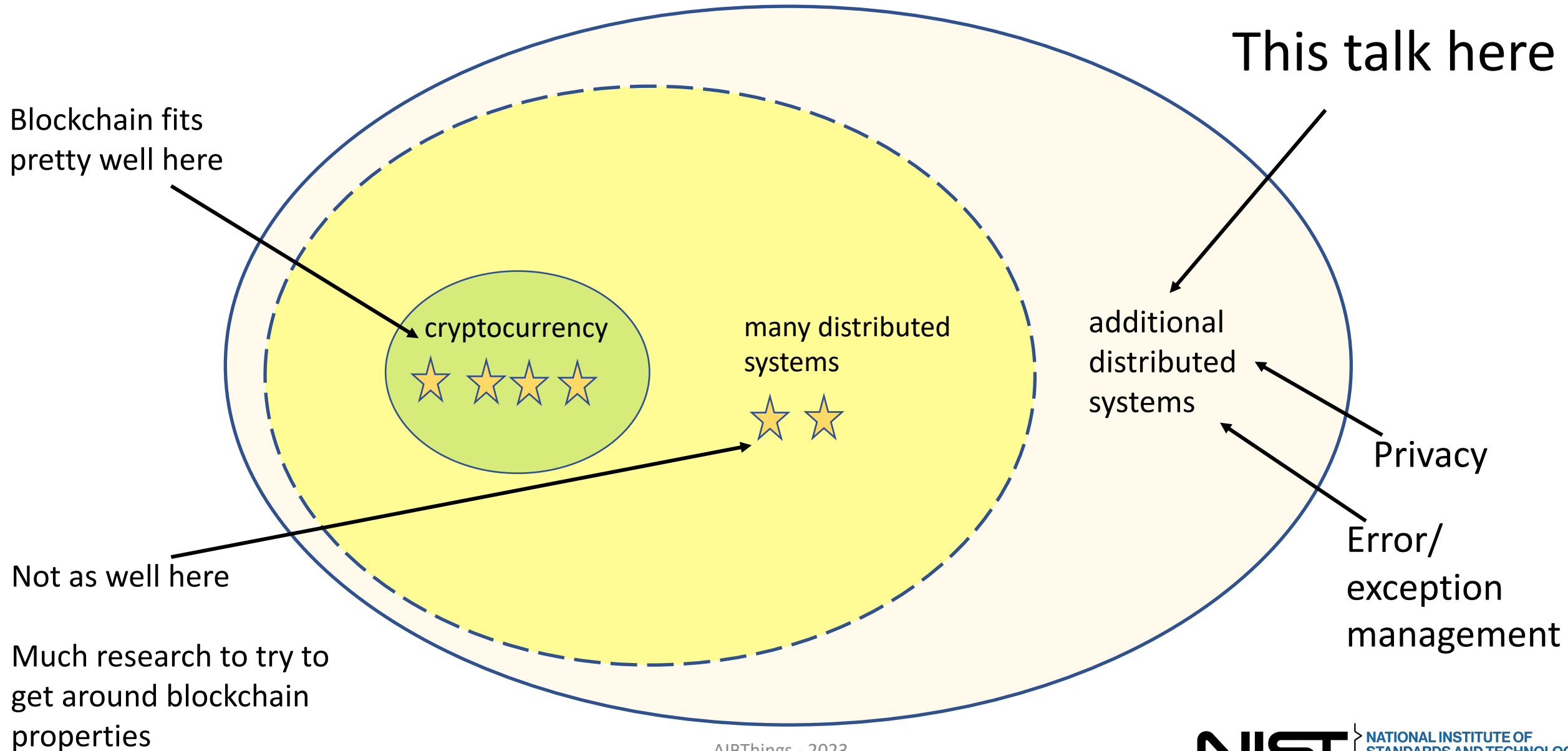
IEEE International Conference on Artificial  
Intelligence, Blockchain, and Internet of Things  
Sept 17, 2023

Rick Kuhn  
US National Institute of Standards and Technology  
kuhn@nist.gov

# Key Points – why listen to this talk?

- Blockchain has valuable properties, but conflicts with privacy and exception management – “immutable” - deletion impossible
  - ➔ Sometimes we don't need blockchain, just some blockchain features
- Data structure called *blockmatrix* provides distributed trust, integrity protection of blockchain, but allows controlled edits for privacy or corrections
- Drop-in compatibility for Hyperledger Fabric applications
  - ➔ Released and available

# Market, range of applications for DLT



# Why use redactable DLT 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
  - *any personal* data "concerning an identified or identifiable natural person"
  - includes pseudo-anonymized data linkable to person
  - US states adopting similar privacy rules, including California and Virginia

# What's been tried to solve blockchain/privacy conflict?

- Don't put personal data on blockchain – but pseudo-anonymized data are still considered personal; Financial transactions are obviously personal data
- Encrypt data and destroy key to delete – but data must be secure for decades (e.g., DES replaced in only 17 years)
- Chameleon hash function – non-standard cryptography
- Off-chain storage of sensitive data – what if on-chain index to off-chain data is also sensitive?

# Many blockchain applications don't need blockchain, just some blockchain features

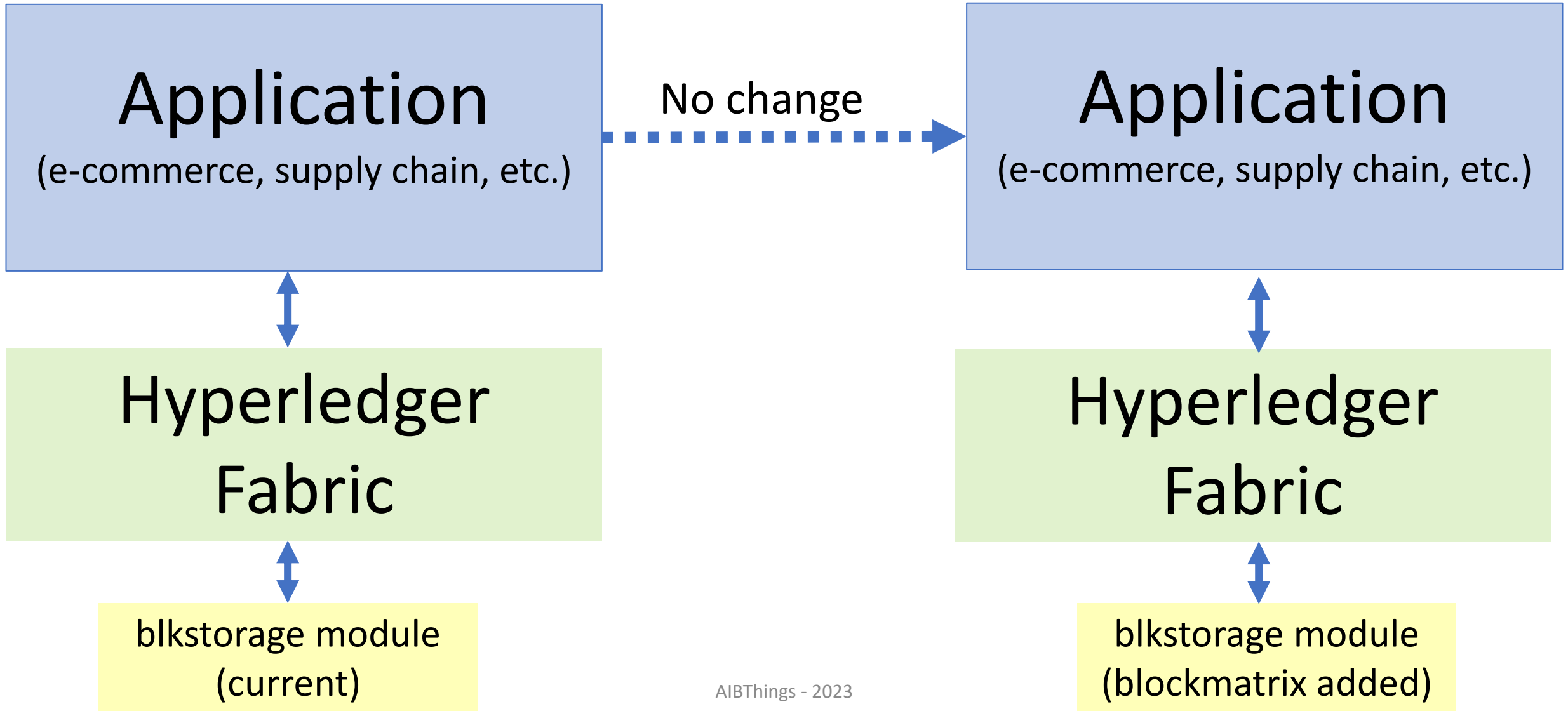
**Datablock matrix** –two hash values per block instead of linked chain

Blockchain -> distributed trust, integrity protection, immutability

Datablock matrix -> distributed trust, integrity protection, editable

- Open source
- Incorporated into Next Gen Access Control
- NOT to replace blockchain, to provide alternative tools for distributed system design
- Hyperledger Fabric component available

# Compatible with Hyperledger applications



# 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 <sub>0,-</sub>
1						H <sub>1,-</sub>
2						H <sub>2,-</sub>
3			X			H <sub>3,-</sub>
4						H <sub>4,-</sub>
	H <sub>-,0</sub>	H <sub>-,1</sub>	H <sub>-,2</sub>	H <sub>-,3</sub>	H <sub>-,4</sub>	

Figure 1. Block matrix



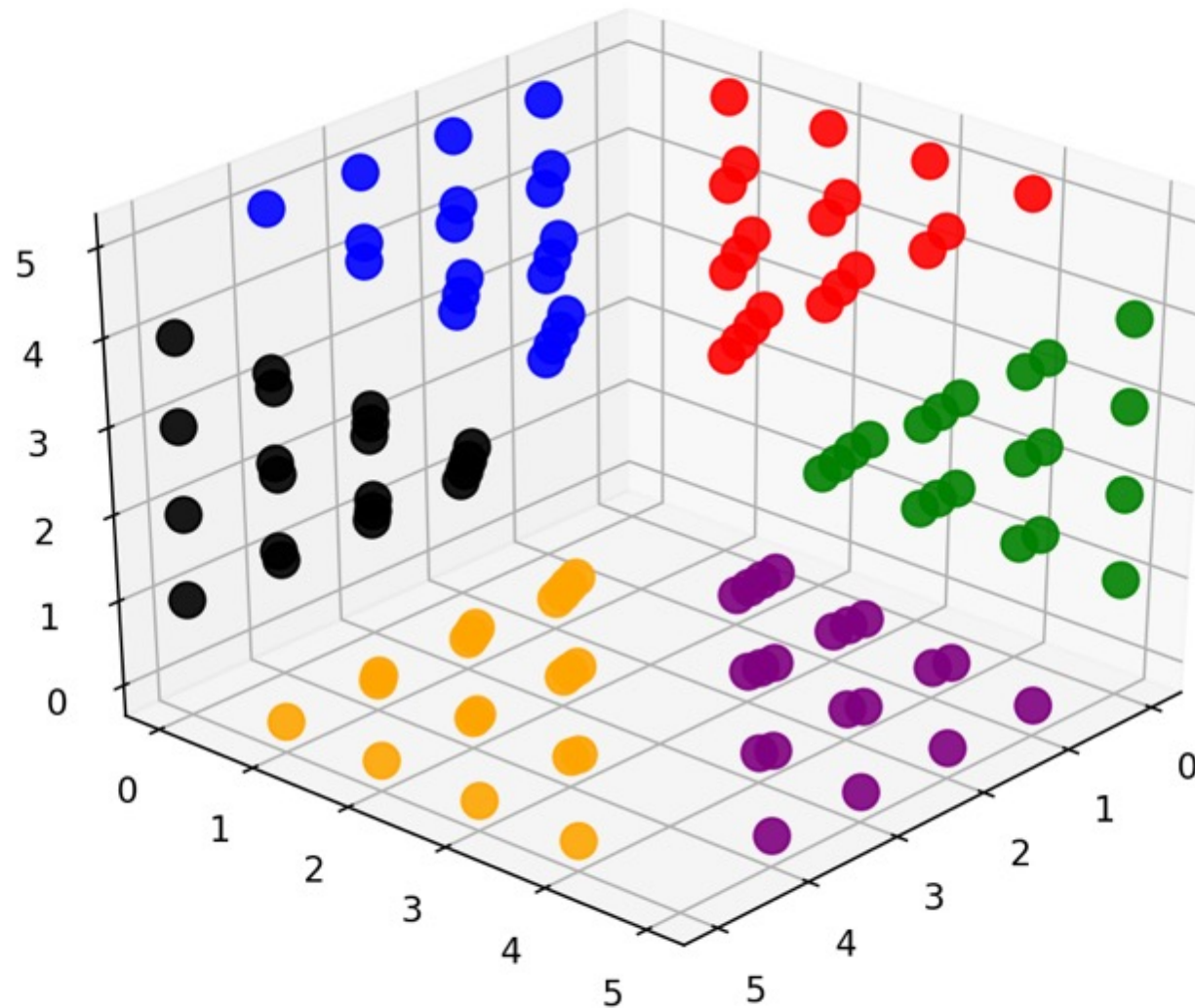
# How does this work?

- 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.

# Structure can be extended to multiple dimensions

- Block dispersal for 3 dimensions
- Location in sectors 0..5 according to  $b \bmod 6$  for block  $b$



# 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

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?

YES

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

YES

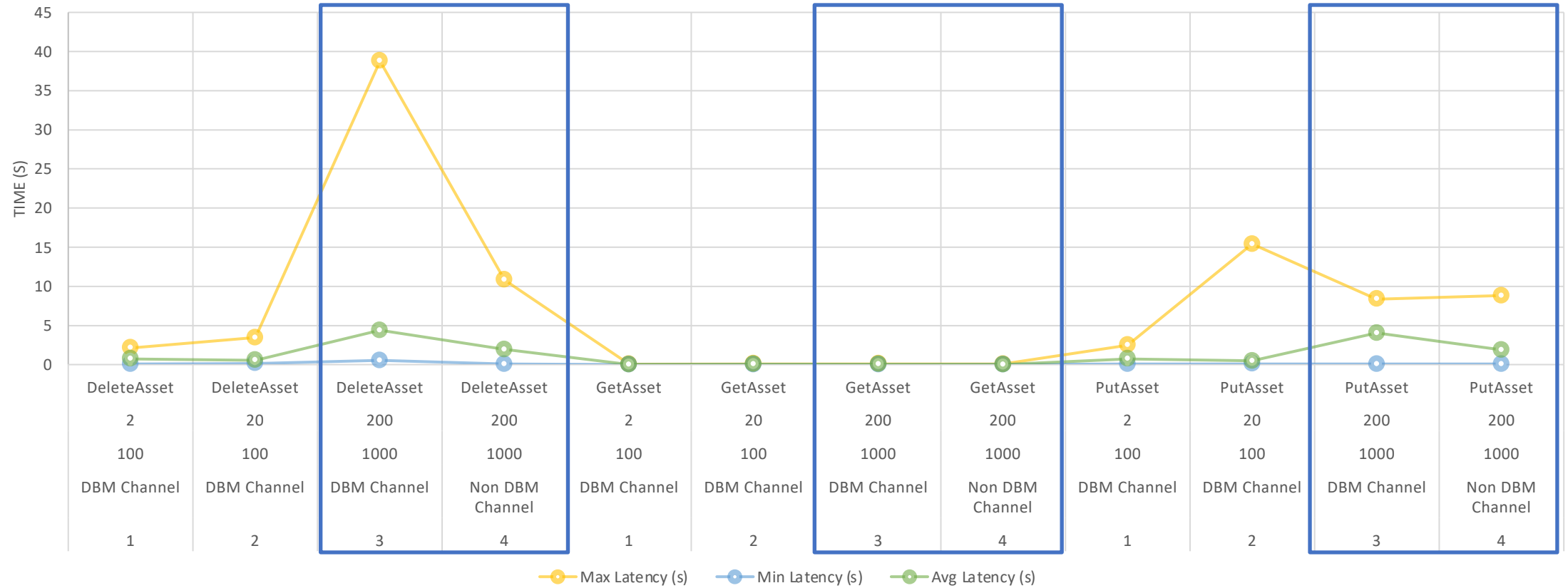
You may have a useful data block matrix use case

# Hyperledger blockmatrix implementation

- Designed to use existing API as closely as possible
  - add blocks in same manner as adding to blockchain
- Blockmatrix is configurable by channel (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
- RED Ledger = Redactable Enhanced Distributed Ledger
- <https://csrc.nist.gov/projects/redactable-distributed-ledger>

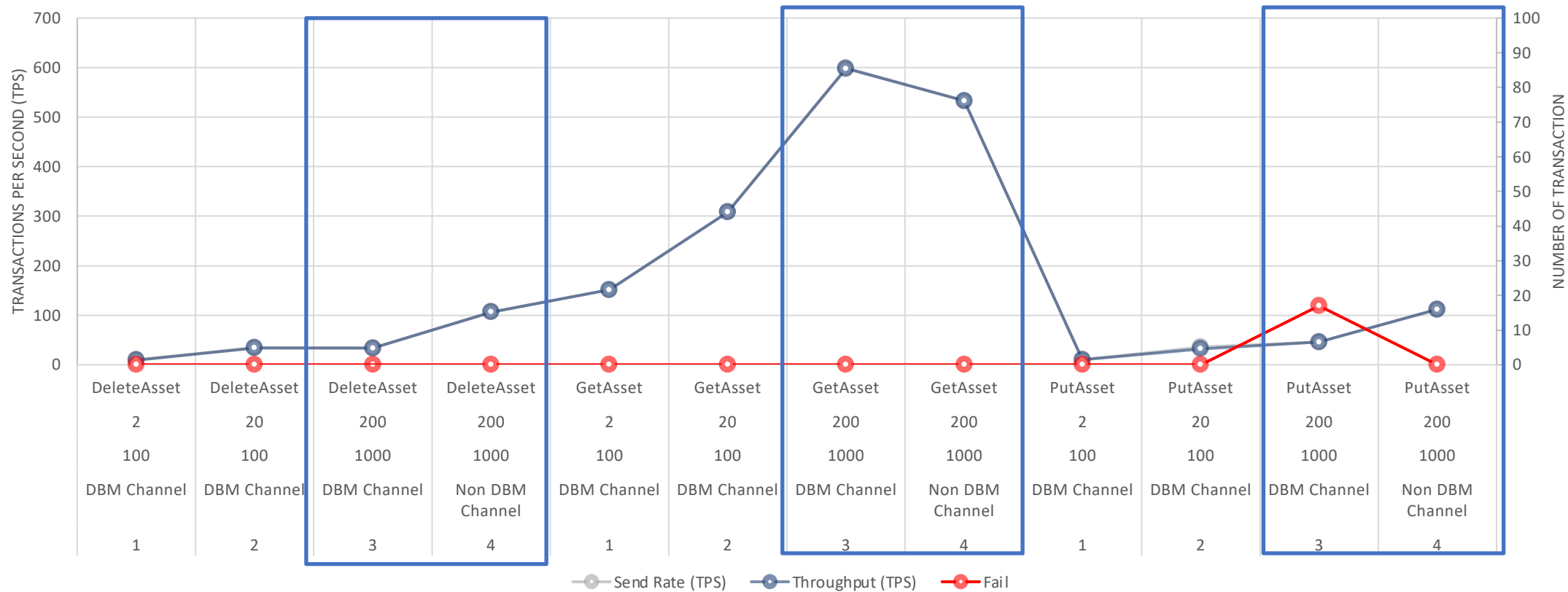
# Latency

Impact of the number of assets and transaction loads on Latency



# Throughput

Impact of the number of assets and transaction loads on Failure and Send Rate/Throughputs



# More Information

## Foundation:

- 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/2022/05/20/data-structure-for-integrity-protection-with-erasure-capability/final>

## Applications:

- Roberts, J. D., Defranco, J. F., & Kuhn, D. R. (2023). [Data Block Matrix and Hyperledger Implementation: Extending Distributed Ledger Technology for Privacy Requirements](#). *ACM Distributed Ledger Technologies: Research and Practice*, 2(2), 1-11.

## 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
- Gokhan Kocak, Asena, Inc.