Transaction Processing in Blockchain Sharding

Ramesh Adhikari
PhD Student

Supervised By
Dr. Costas Busch

School of Computer and Cyber Sciences
Augusta University
Augusta, GA, USA

September 16, 2024

Overview

- **Papers We Survey**
- **Introduction to Blockchain**
  - What is Blockchain?
  - Scalability Issues in Blockchains
- **Sharding as a Solution**
  - How Sharding Addresses Scalability Issues
- **Focus Areas: Transaction Processing in Blockchain Sharding**
  - Intra-Shard Transactions Processing
  - Cross-Shard Transactions Processing

Papers We Survey

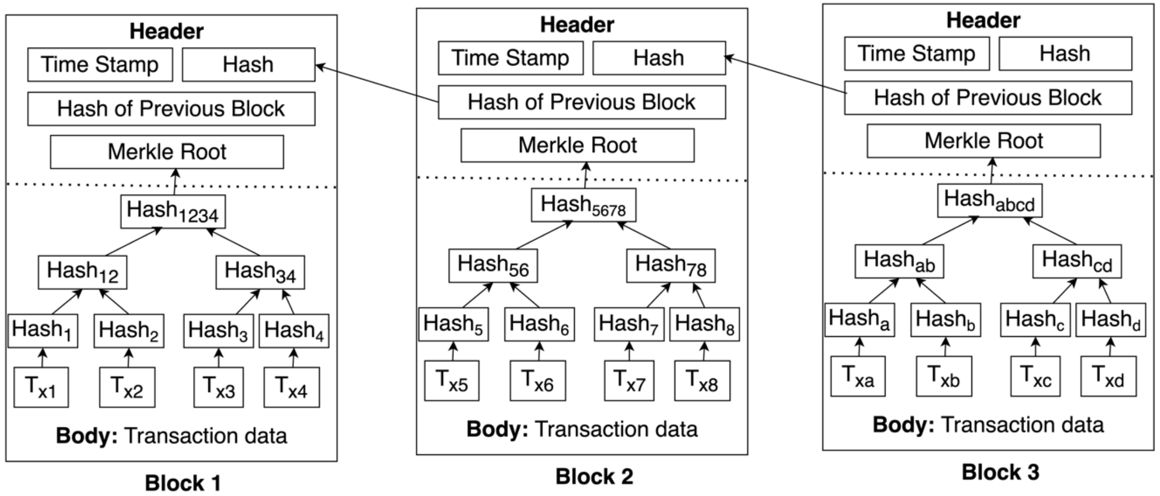
Papers We Survey

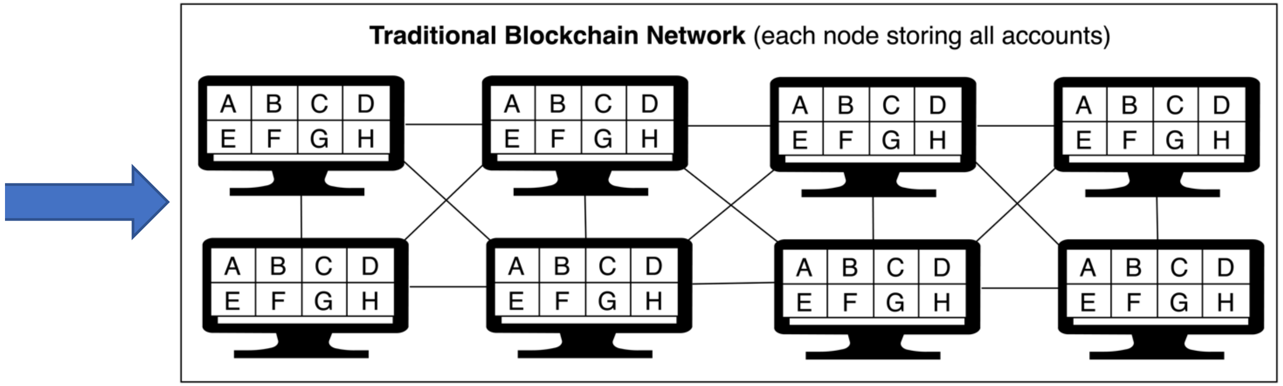
- We survey 11 blockchain sharding papers
  - Focus on **transaction processing**
- Identify problems
- Propose future research directions

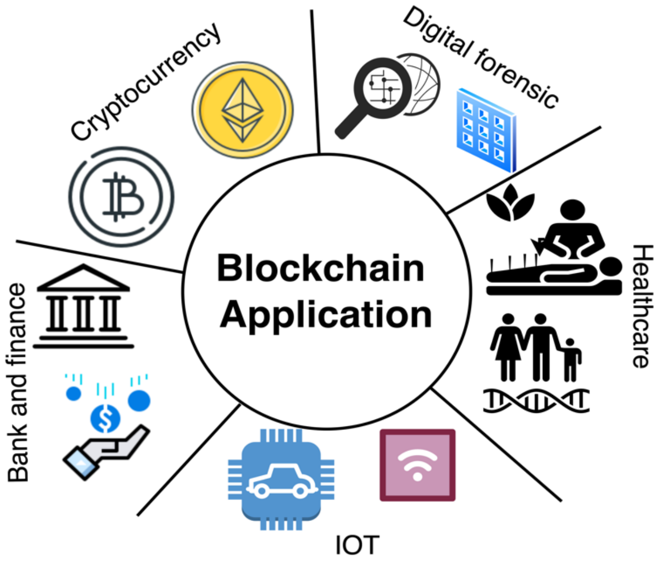
ELASTICO [1] (2016)	GriDB [7] (2023)
RapidChain [2] (2018)	LB-Chain [8] (2023)
Pyramid [3] (2021)	TxAllo [9] (2023)
Meepo [4] (2021)	X-Shard [10] (2024)
ByShard [5] (2021)	Estuary [11] (2024)
Service-Aware [6] (2022)	

What is Blockchain?

What is Blockchain?







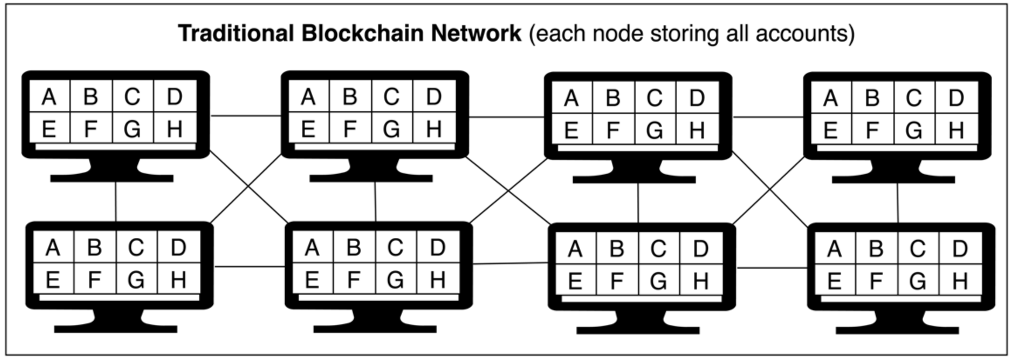
Features

- ✓ Decentralized
- ✓ Immutable
- ✓ Fault Tolerant
- ✓ Transparent
- ✓ Enhanced Security

Issues in Blockchain

Issues in Blockchain

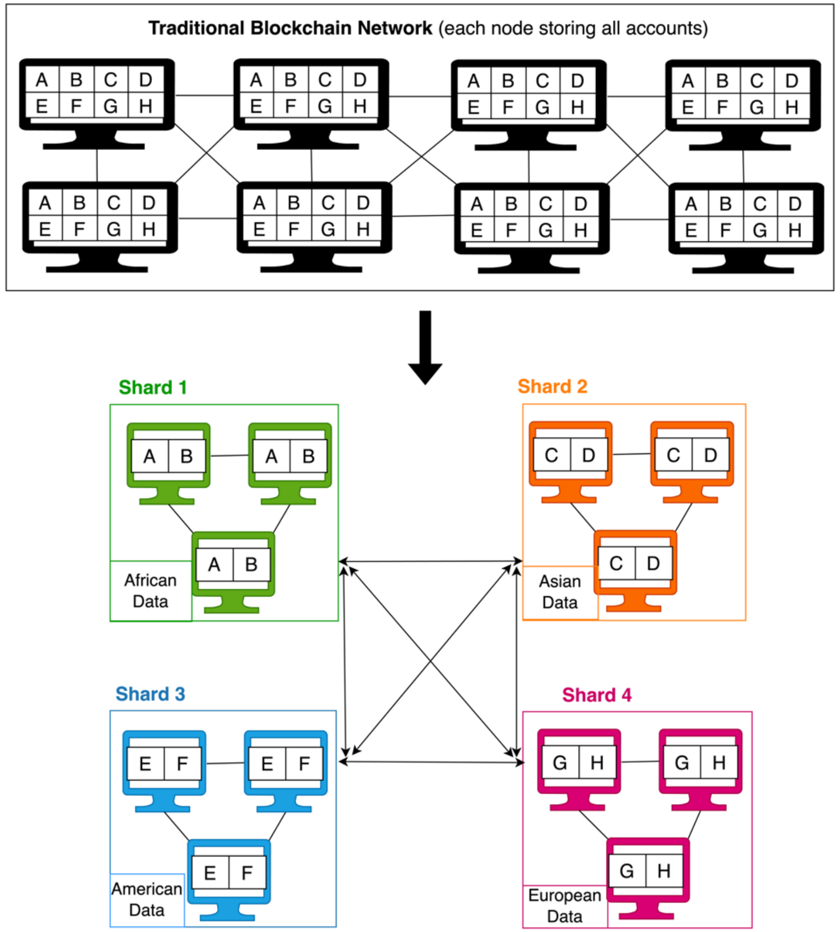
- All nodes need to agree on the validity of transactions
- Hence every node must store all transactions
- All nodes need to reach consensus to append block
- Scalability issue: limited throughput, and long confirmation time



Application	Txn per second (TPS) (Throughput)	Average Txn (Block) confirmation time (Latency)
	3-7	60 Minutes
ethereum	15-20	3 Minutes
VISA	24000	Instantly

Sharding as a Solution

Sharding as a Solution



Sharded Blockchain each storing part of accounts where A, B,C,D,E,F are accounts

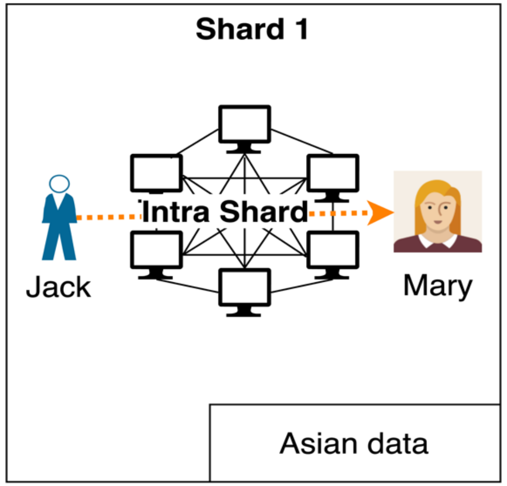
Advantages

- ✓ **Scalability**: Handles more transactions as network grows.
- ✓ **Increase throughput**: Parallel transaction processing.
- ✓ **Efficiency**: Reduces storage, communication, and computing complexity.

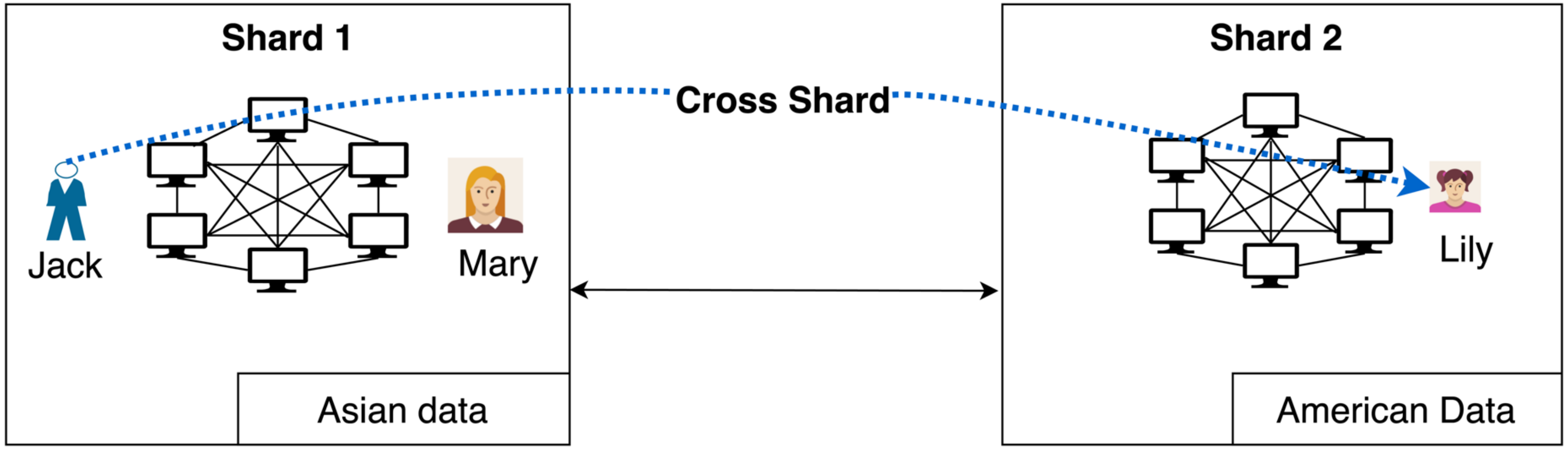
Transaction Processing in Blockchain Sharding

Transaction Processing in Blockchain Sharding

- Intra-Shard Transactions Processing



- Cross-Shard Transactions Processing

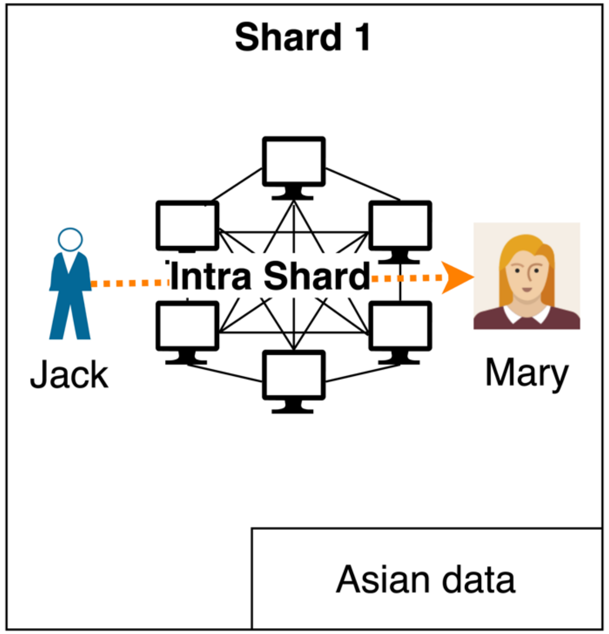


Intra-Shard Transaction Processing

Intra-shard Transaction Processing Protocol

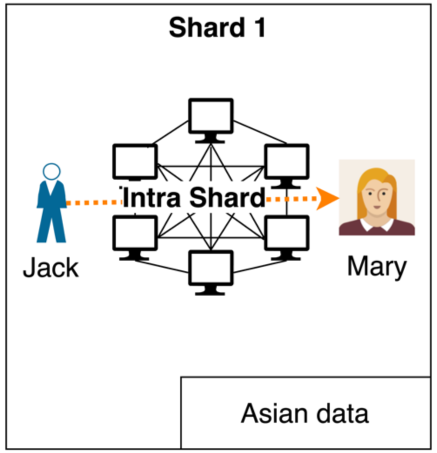
- PBFT Consensus Algorithm
  - Commonly used in many sharding papers
  - Examples: ELASTICO [1], ByShard [5], X-Shard [10], Estuary [11]

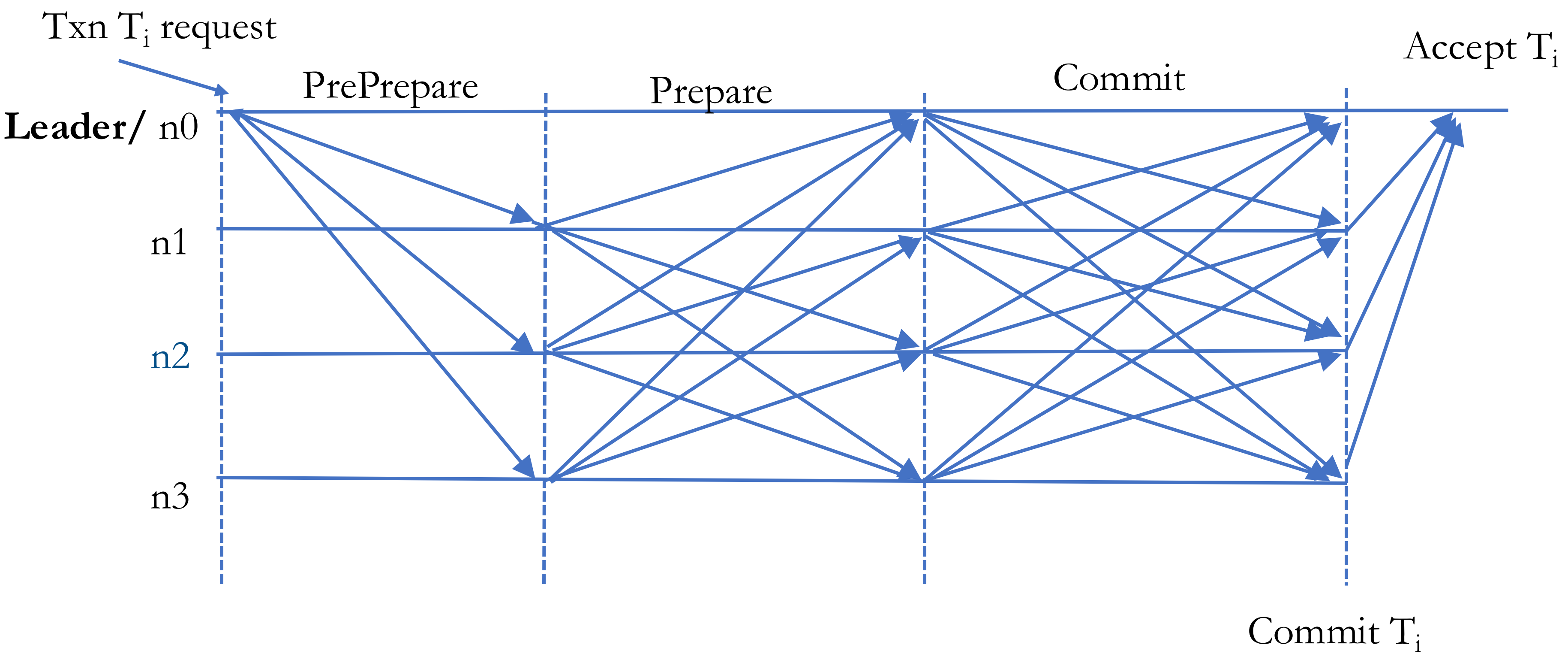
- Variants of PBFT Consensus Algorithm
  - Some papers used variants such as Sync PBFT and Fast PBFT
  - Examples: RapidChain [2], Service-Aware Dynamic Sharding [6].



PBFT Consensus Algorithm

Used by: ELASTICO [1], ByShard [5], X-Shard [10], Estuary [11]



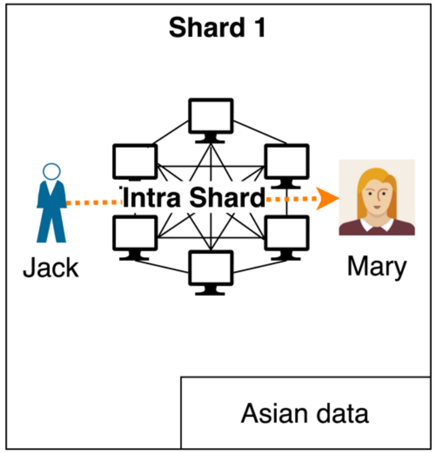


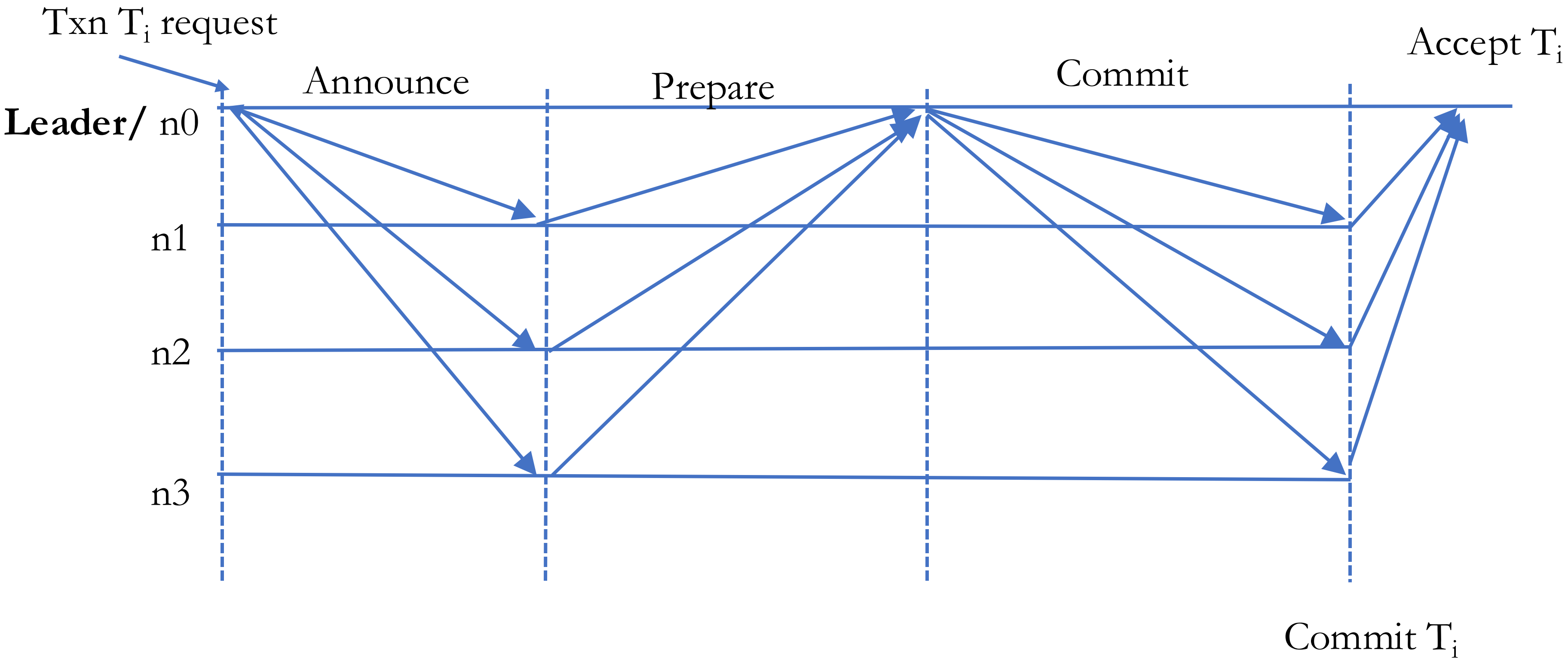
Complexity $O(m^2)$

$\boldsymbol{m}$ is the number of nodes within a shard

Fast PBFT Algorithm

Used by: Service-Aware Dynamic Sharding [6]





- Complexity $O(m)$
- But nodes needs to be always online and in sync
- Single point of Failure at Leader

Summary of Intra-shard Consensus Protocols

Sharding Protocol	Intra-shard consensus protocol		
	Algorithm	Complexity	Fault Tolerance
ELASTICO [1]	PBFT [12]	$O(m^2)$	33 %
ByShard [3]	PBFT [12]	$O(m^2)$	33 %
X-Shard [10]	PBFT [12]	$O(m^2)$	33 %
Estuary [11]	PBFT [12]	$O(m^2)$	33 %
RapidChain [2]	Sync PBFT [14]	$O(m^2)$	50 %
Service-Aware [6]	Fast PBFT [13]	$O(m)$	33 %

m is the number of nodes within a shard

- Fast BFT [13] requires nodes to be always online and in sync with the consensus progress. However, this will not be true in a real-world scenario
- And single point of Failure at leader

Problems and Future works

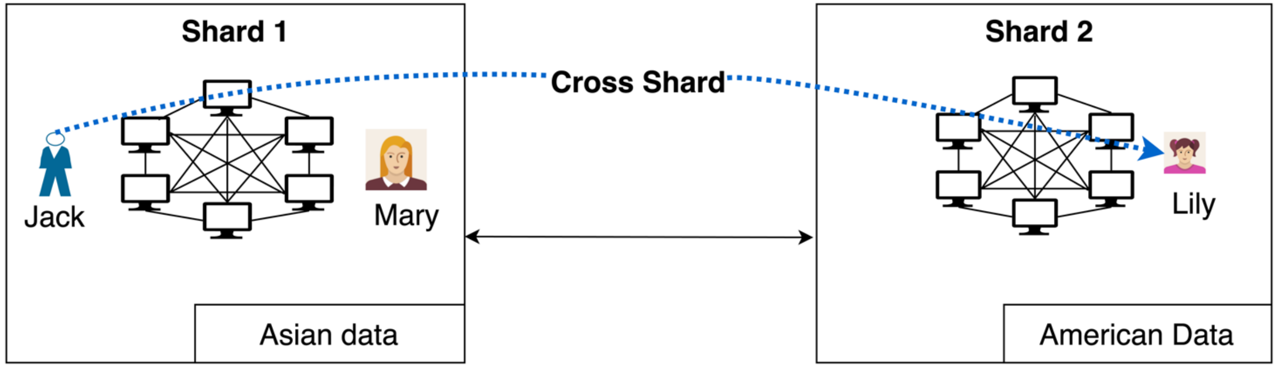
- **Communication Overhead**
  - **Issue:** PBFT [12] consensus has high communication costs $O(m^2)$, especially with more nodes
  - **Future Work:** Reduce communication complexity within shards

- **Risk of Malicious Shards**
  - **Issue:** Risk of adversary-controlled shards
  - **Future Work:** Develop methods to detect, restore, and replace malicious shards through the actions of **honest shards** (or **backup shards)**

Cross-Shard Communication

Cross-Shard Communication Protocol

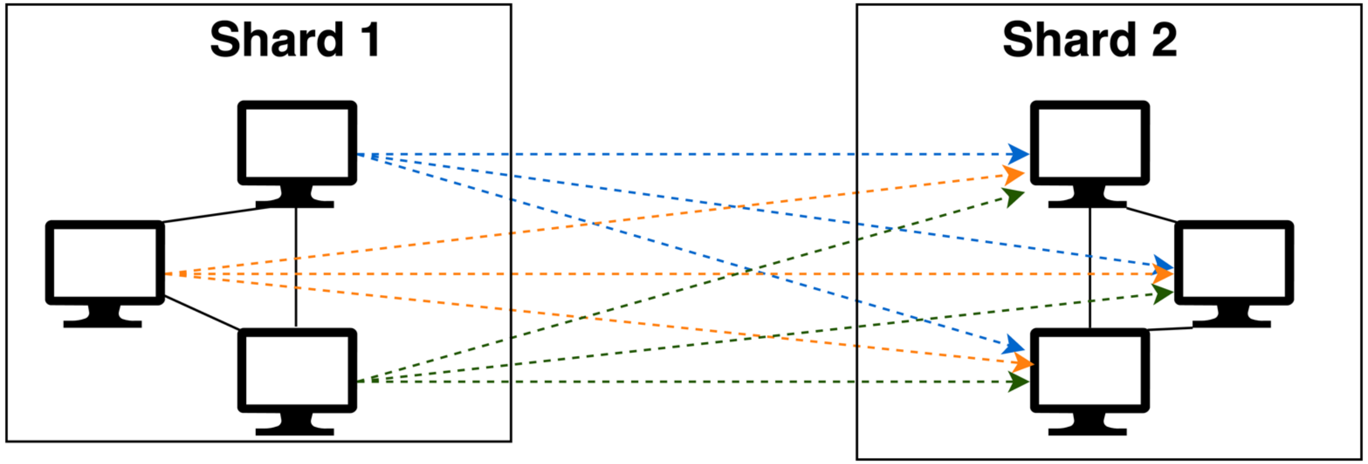
- Broadcast-Based Protocol
  - **Applied In**: Elastico [1], Pyramid [3]
- Cluster Sending Protocol
  - **Applied In**: Byshard [5]



Broadcast-Based Protocol

Used in Elastico [1], Pyramid [3]

- Operates with Byzantine failures
- Use a consensus protocol (PBFT) to agree on a value
- Messages are broadcasted from one shard to another shard
  - Ensure at least one non-faulty node receives message



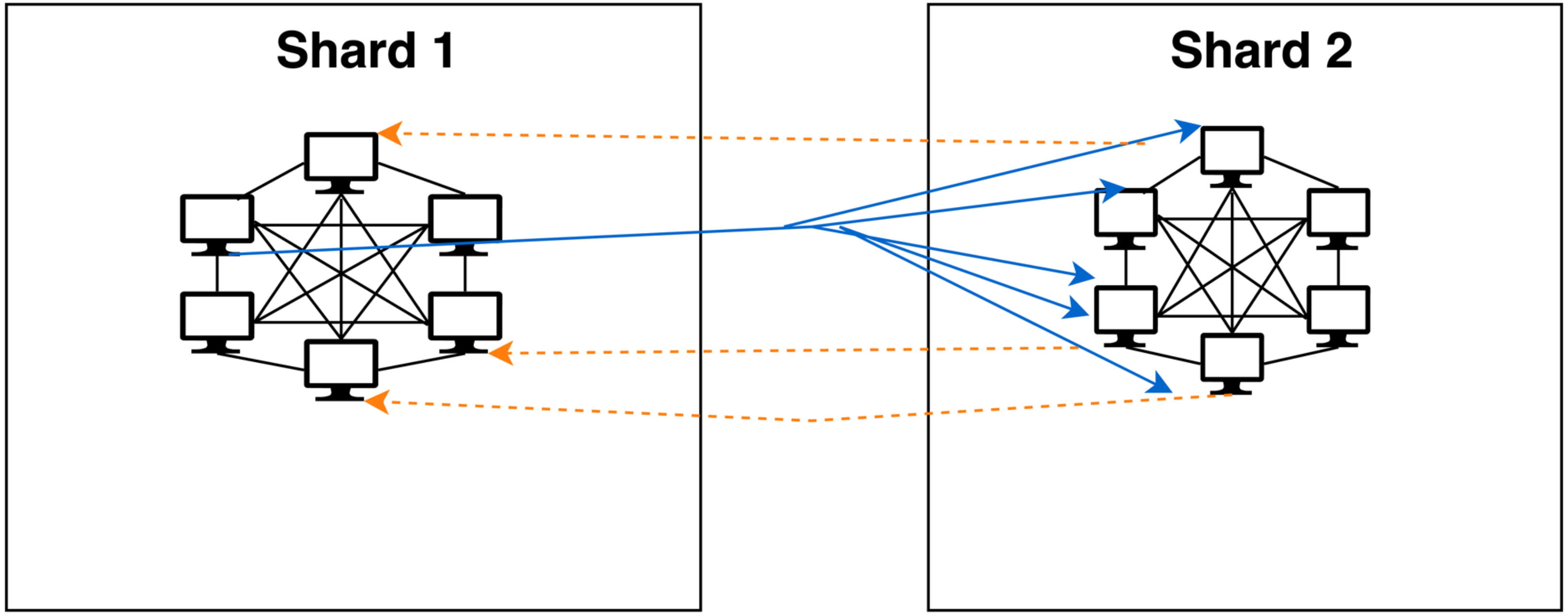
Complexity $O(m^2)$

m is number of nodes within a shard

Cluster Sending Protocol

Used in Byshard [5]

- All honest nodes from S1 agree on message using PBFT before sending
- All honest nodes in the receiving shard receive the message
- The sending shard receives confirmation of message receipt



Problems and Future works

- **Communication Complexity**
  - **Issue:** Broadcast-based protocols (Elastico [1], Pyramid [3]) have high communication costs $O(m^2)$
  - **Future Work:** Develop cross-shard communication protocols with lower complexity

- **Risk of Malicious Leaders**
  - **Issue:** Single leader nodes can act maliciously (GriDB [7], Byshard [5], RapidChain[2]), disrupting shard communication
  - **Future Work:** Focus on electing honest leaders, detecting malicious ones, and enabling quick recovery

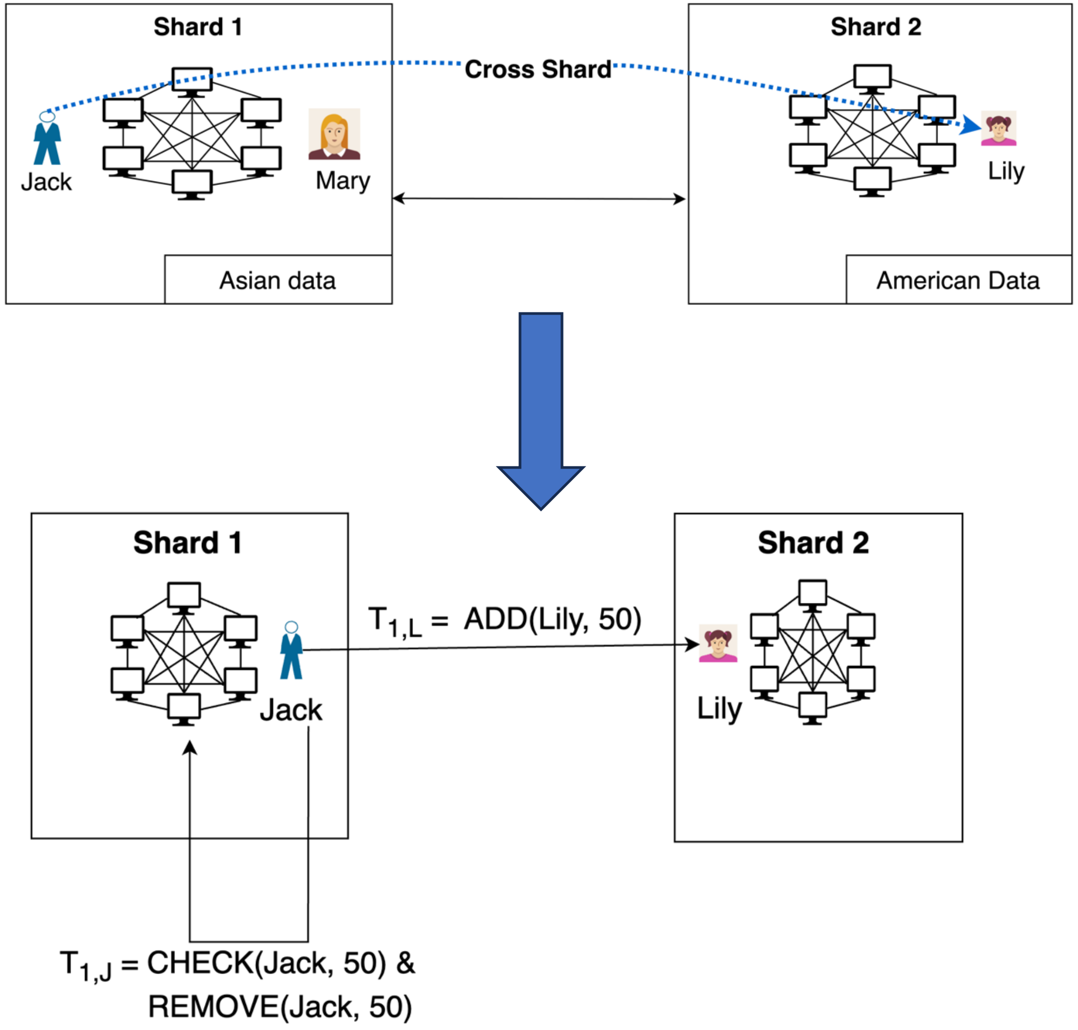
Cross-shard Transaction Processing

Cross-shard Transaction Processing

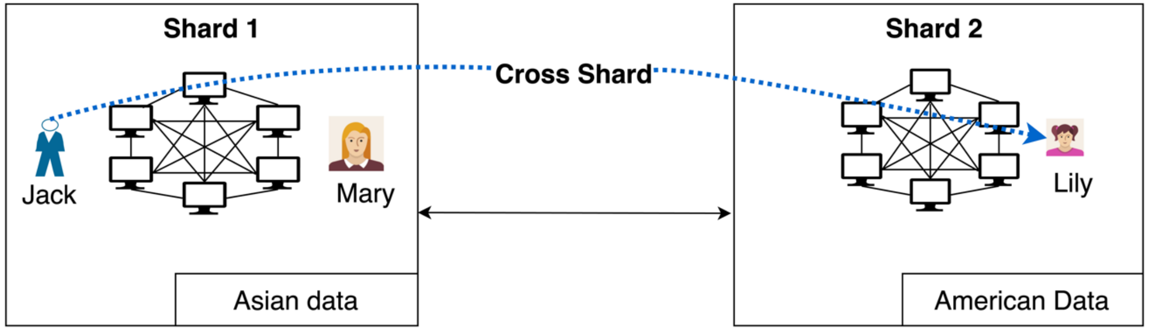
- **Basic Idea:** Split transaction into sub-transactions and send to respective shards for processing

T_1= Send \$50 From **Jack** account to **Lily** account

Ensure atomic and consistent commits in each shard



Cross-shard Transaction Processing Technique



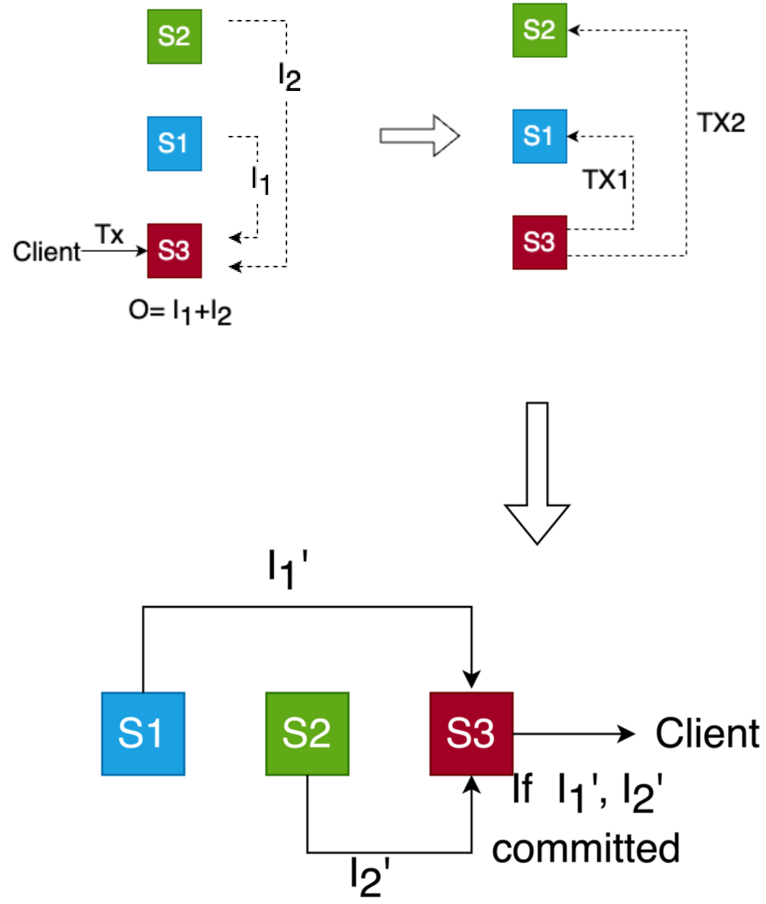
- Transaction Split and Confirmation Approach
  - **Applied in:** Rapidchain [2]
- Two-Phase Commit Based approach
  - **Applied in:** ByShard [5], Service-Aware[6], Estuary [11]
- Overlap Shard Approach
  - **Applied in:** Pyramid [3]
- Dynamic Sharding
  - **Applied in:** Service-Aware [6], LB-Chain [8], TxAllo [9], X-shard [10], Estuary [11]

Transaction Split and Confirmation Approach

- Suppose client Tx submitted to Shard S3
- Tx consists of two inputs, I_1 (from S1) and I_2 (from S2), and one output, O (in S3)

Used in RapidChain [2]

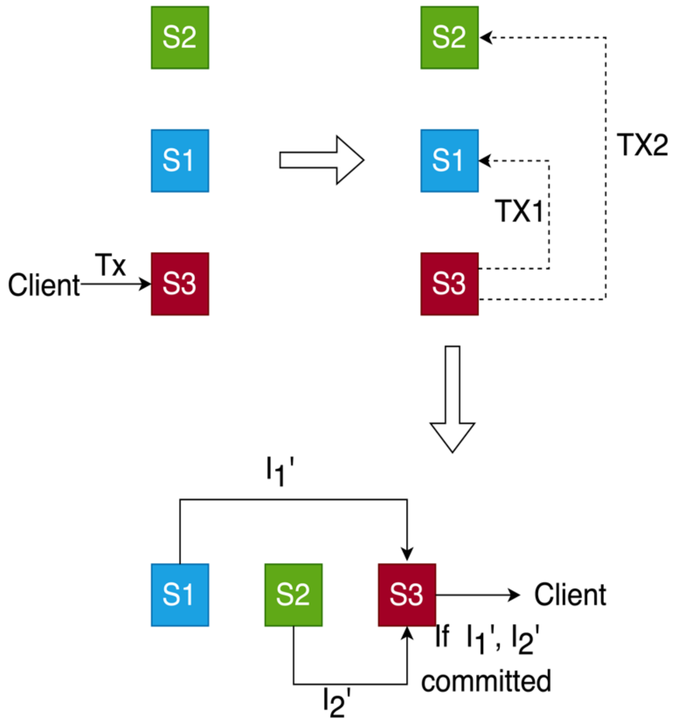
- Leader of S3
  - Split Tx into three subtransactions:
    - **Tx1:** $< I_1, I_1' >$ (Shard S1)
    - **Tx2:** $< I_2, I_2' >$ (Shard S2)
    - **Tx3:** $< (I_1' + I_2'), O >$ (Shard S3)
  - Send Tx1 to Shard S1 and Tx2 to Shard S2
- Shard S1 and S2, Commit Tx1 and Tx2 to their ledgers



- Final Steps:
  - If Tx1 and Tx2 are committed in S1 and S2
    - S1 and S2 send I'_1 and I'_2 to S3
  - Tx3: $< (I_1' + I_2'), O >$ is committed in S3

Problems and Future works

- Lack of Atomicity:
  - **Issue**: Tx split into Tx1, Tx2, and Tx3 if Tx1 **fails** in shard S1 but Tx2 **succeeds** in S2 can destroy atomicity of transaction
  - **Future Work**: Develop methods to ensure atomicity and isolation property of transaction



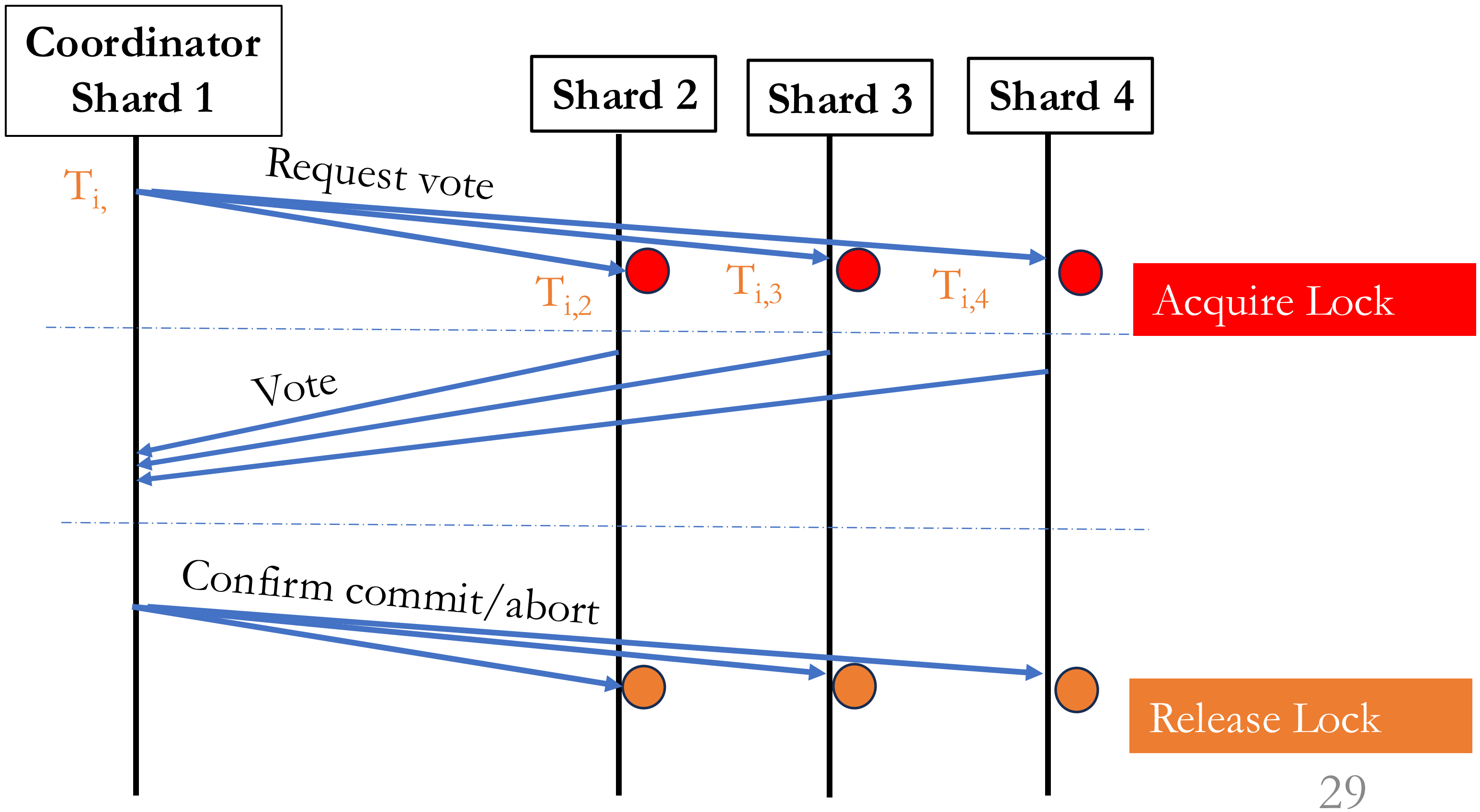
- Lack of Multi-output Support:
  - **Issue:** Only handles multi-input, single-output transactions. (Smart contract required multi output)
  - **Future work:** Design approaches to handle multi-input, multi-output cross-shard transactions

Two-Phase Commit Protocol

Used in ByShard [5], Service-Aware[6], Estuary [11]

- **Two-Phase Commit:** Ensures atomic decisions on transaction commitment
- **Two-Phase Locking:** Provides concurrency control

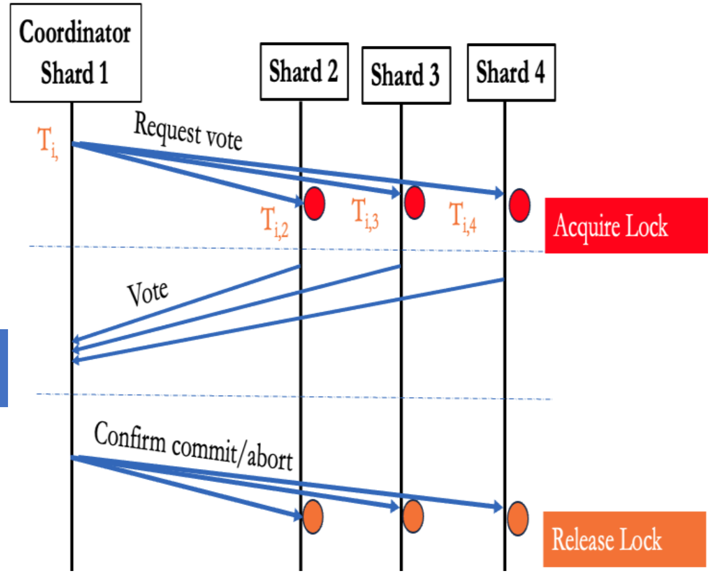
- Suppose there is Transaction T_i which access accounts in Shard 2, 3, 4
- Coordinator shard split transaction into subtransacitons as $T_{i,2}$ $T_{i,3}$ $T_{i,4}$ and send to respective shard



Problems and Future works

- Account Locking
  - **Issue:** Locking accounts for concurrency control can lead to performance issues and deadlocks if not managed properly
  - **Future Work:** Explore lock-free transaction methods

We provide Lockless Blockchain Sharding with Multiversion Control [15] (SIROCCO 2023)

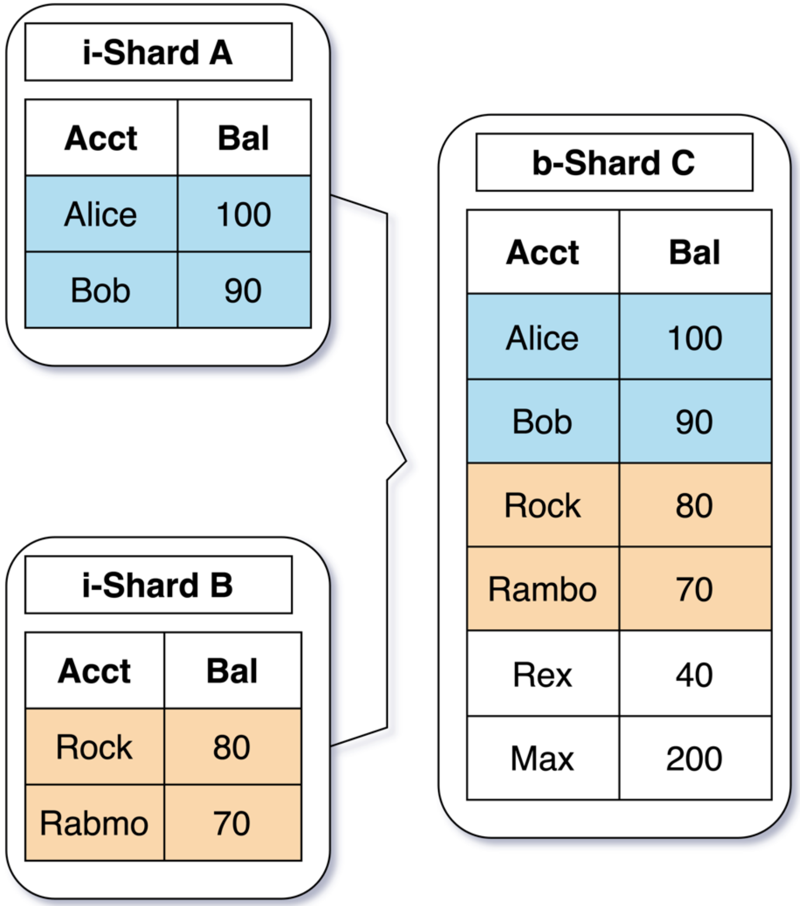


- High Communication and consensus Costs
  - **Issue:** The need for extensive back-and-forth communication increases overhead for consistent commitment
  - **Future Work:** Explore new approach to reduce communication and consensus costs

Overlap Shard Approach

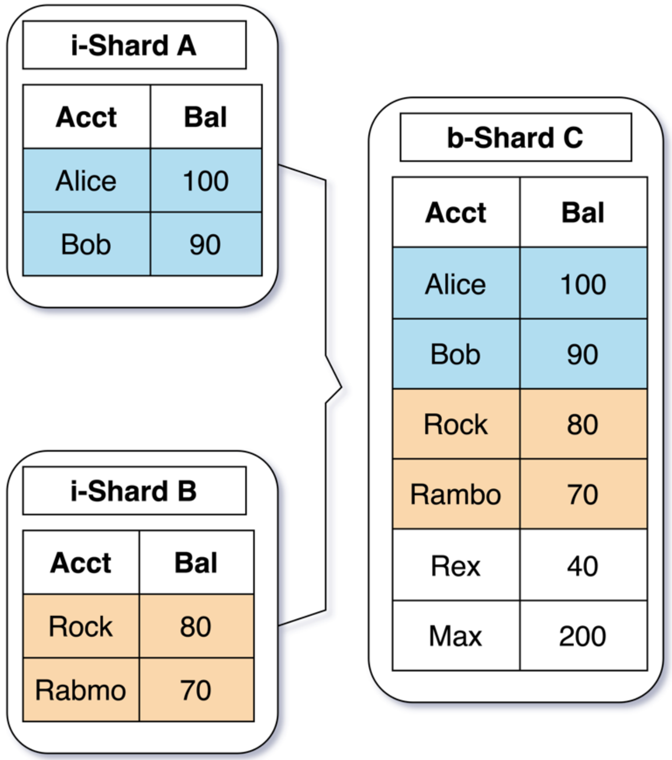
Used in Pyramid [3]

- Some of the shard holds others shards state information
- Cross-shard blocks are proposed by a b-shard (which has other i-shard state information)
- i-shards verify transactions and send accept/reject messages
- Accepted blocks are committed across shards



Problems and Future works

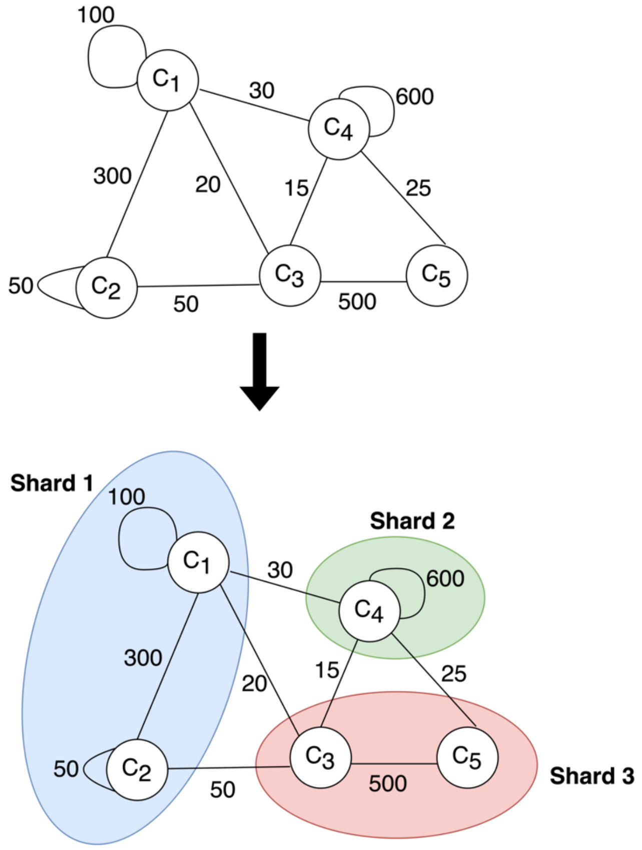
- **Storage Overhead**
  - **Issue:** Storing additional state information in shards leads to higher storage requirements
  - **Future Work:** Find methods to reduce storage overhead while maintaining consistency
- **Efficient Consensus Needed**
  - **Issue:** Overlapping shards require advanced consensus protocols for accurate state updates
  - **Future Work:** Propose consensus protocols to enhance efficiency and maintain consistency in state updates



Dynamic Sharding

Used in Service-Aware [6], LB-Chain [8], TxAllo [9], X-shard [10]

- Goal:
  - Minimize Cross-Shard Transactions
  - Dynamically migrating accounts and their states between shards



- Techniques:
  - Graph-Based Analysis:
    - Construct transaction-account(state) graphs.
    - Identify heavily interconnected accounts.
    - The weight represents the number of transactions
  - Machine Learning: LB-Chain [8], TxAllo [9]
    - Predict future transaction by analyzing history of transaction pattern for optimal shard allocations

Problem and Future works:

- **Inaccurate Transaction Prediction**
  - **Issue:** Machine learning models may fail to accurately predict transaction patterns
  - **Future Work:** Enhance predictive models to improve shard allocation accuracy

- **High Migration Costs**
  - **Issue:** Migrating accounts between shards can create significant overhead and congestion
  - **Future Work:** Develop strategies to minimize migration cost and network congestion

- **Challenges in Consistent Migration**
  - **Issue:** Achieving atomic and consistent state migration across shards is complex
  - **Future Work:** Investigate efficient methods for maintaining consistency and atomicity during state migration

Summary of Problems and Future Directions

Topics	Problems	Future Research Directions
Intra-Shard Transaction Processing	• **Communication Overhead:** PBFT consensus has high communication costs, especially with more nodes.	• Develop the intra-shard consensus protocol with minimum communication complexity within shards.
	• **Risk of Malicious Shards:** Risk of adversary-controlled shards.	• Develop methods to detect, restore, and replace malicious shards through the actions of **honest shards**.
Cross-Shard Communication	• **Risk of Malicious Leaders:** Single leader nodes can act maliciously, disrupting shard communication	• Focus on electing honest leaders, detecting malicious ones, and enabling quick recovery.
	• **Communication Complexity:** E.g. Broadcast-based protocols have high communication costs $O(m^2)$	• Develop cross-shard communication protocols with lower complexity
Cross-Shard Transaction Processing	• **Atomicity and Isolation Issues**: Difficulties in ensuring transaction properties.	• Develop techniques to ensure reliable transaction atomicity and isolation with low complexity.
	• **High Communication Costs:** Lock based approach overhead with back and forth communication for consistent commitment	• Explore new approach to reduce communication costs
	• **Costly Account Migration In Dynamic Sharding:** Migrating accounts between shards can create significant overhead and congestion. If we migrate account we need to	• Develop strategies to minimize migration overhead and network congestion.

Research progress and services

- Published two papers
    - Lockless Blockchain Sharding with Multiversion Control (SIROCCO 2023)
        - The 30th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2023) ,in Madrid, Spain (June 2023)
    - Stable Blockchain Sharding under Adversarial Transaction Generation (SPAA 2024)
        - The 36th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2024), in Nantes, France. (June 2024)

- Currently working on three papers
    - Fast Transaction Scheduling in Blockchain Sharding
    - Transaction Scheduling in Fog-Cloud computing
    - Stable Blockchain Sharding (Journal version)

- Review 17 papers
    - 5 Journal papers
        - IEEE Transactions on Network and Service Management (2022), Transactions on Mobile Computing (2024), Blockchain: Research and Applications (2024), Journal of Network and Computer Applications (2024)
    - 12 conference papers
        - Blockchain 2023, Blockchain 2024, PODC 2024, SIROCCO 2024, SIGMIS CPR 2024

Reviewer: IEEE Transactions on Green Communications and Networking

References

- [1] Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., & Saxena, P. (2016, October). A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (pp. 17-30).
- [2] Zamani, M., Movahedi, M., & Raykova, M. (2018, October). Rapidchain: Scaling blockchain via full sharding. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security* (pp. 931-948).
- [3] Hong, Z., Guo, S., Li, P., & Chen, W. (2021, May). Pyramid: A layered sharding blockchain system. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications* (pp. 1-10). IEEE.
- [4] Zheng, P., Xu, Q., Zheng, Z., Zhou, Z., Yan, Y., & Zhang, H. (2021, April). Meepo: Sharded consortium blockchain. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)* (pp. 1847-1852). IEEE.
- [5] Hellings, J., & Sadoghi, M. (2021). Byshard: Sharding in a byzantine environment. *Proceedings of the VLDB Endowment*, *14*(11), 2230-2243.
- [6] Set, S. K., & Park, G. S. (2022). Service-aware dynamic sharding approach for scalable blockchain. *IEEE Transactions on Services Computing*, *16*(4), 2954-2969.
- [7] Hong, Z., Guo, S., Zhou, E., Chen, W., Huang, H., & Zomaya, A. (2024). GriDB: scaling blockchain database via sharding and off-chain cross-shard mechanism. *arXiv preprint arXiv:2407.03750*.

References

- [8] Li, M., Wang, W., & Zhang, J. (2023). LB-Chain: Load-balanced and low-latency blockchain sharding via account migration. *IEEE Transactions on Parallel and Distributed Systems*, *34*(10), 2797-2810.
- [9] Zhang, Y., Pan, S., & Yu, J. (2023, April). Txallo: Dynamic transaction allocation in sharded blockchain systems. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)* (pp. 721-733). IEEE.
- [10] Xu, J., Ming, Y., Wu, Z., Wang, C., & Jia, X. (2024). X-Shard: Optimistic Cross-Shard Transaction Processing for Sharding-Based Blockchains. *IEEE Transactions on Parallel and Distributed Systems*.
- [11] Jia, L., Liu, Y., Wang, K., & Sun, Y. (2024). Estuary: A Low Cross-Shard Blockchain Sharding Protocol Based on State Splitting. *IEEE Transactions on Parallel and Distributed Systems.*
- [12] Castro, M., & Liskov, B. (1999, February). Practical byzantine fault tolerance. In *OsDI* (Vol. 99, No. 1999, pp. 173-186).
- [13] Fast bft consensus, 2022. [Online]. Available: https://docs.harmony. one/home/general/technology/consensus
- [14] Abraham, I., Devadas, S., Nayak, K., & Ren, L. (2017). Brief announcement: Practical synchronous byzantine consensus.
- [15] Adhikari, R., & Busch, C. (2023, May). Lockless blockchain sharding with multiversion control. In *International Colloquium on Structural Information and Communication Complexity* (pp. 112-131). Cham: Springer Nature Switzerland.
- [16] Jin, D., Yu, Z., Jiao, P., Pan, S., He, D., Wu, J., ... & Zhang, W. (2021). A survey of community detection approaches: From statistical modeling to deep learning. *IEEE Transactions on Knowledge and Data Engineering*, *35*(2), 1149-1170.
- [17] Blondel Vincent, D., Jean-Loup, G., Renaud, L., & Etienne, L. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, *10*(2008), P10008.

Thank you!

Questions?