Large Scale Systems
CS 410 / 510

Lecture 15:

Consensus across the Globe

Suyash Gupta

Assistant Professor

Distopia Labs and ONRG

Dept. of Computer Science

(E) suyash@uoregon.edu

(W) gupta-suyash.github.io

UNIVERSITY OF OREGON

Assignment 4 is Out!

- Assignment 4 is due on **June 2, 2025** at **11:59pm PST**.
- Please start working with your groups.

Presentations

- Presentation slots are out!
- Each group will present their MiniSpanner in their respective time slots.
- Presentation Format:
  - Each group will get **15-16 min** to present their progress.
  - **5 min** for a working demo
  - **10 min** for Q/A.
  - Every student should present for around 4 minutes.
  - Every student should state their contributions.
  - Based on Q/A and presentation quality, grades will be decided for each student.

Reading Material

- For this class and next lecture:
  - Read **Chapters 3-4** from **Fault-Tolerant Distributed Transactions on Blockchain**.

Last Class

- Last class we looked at:

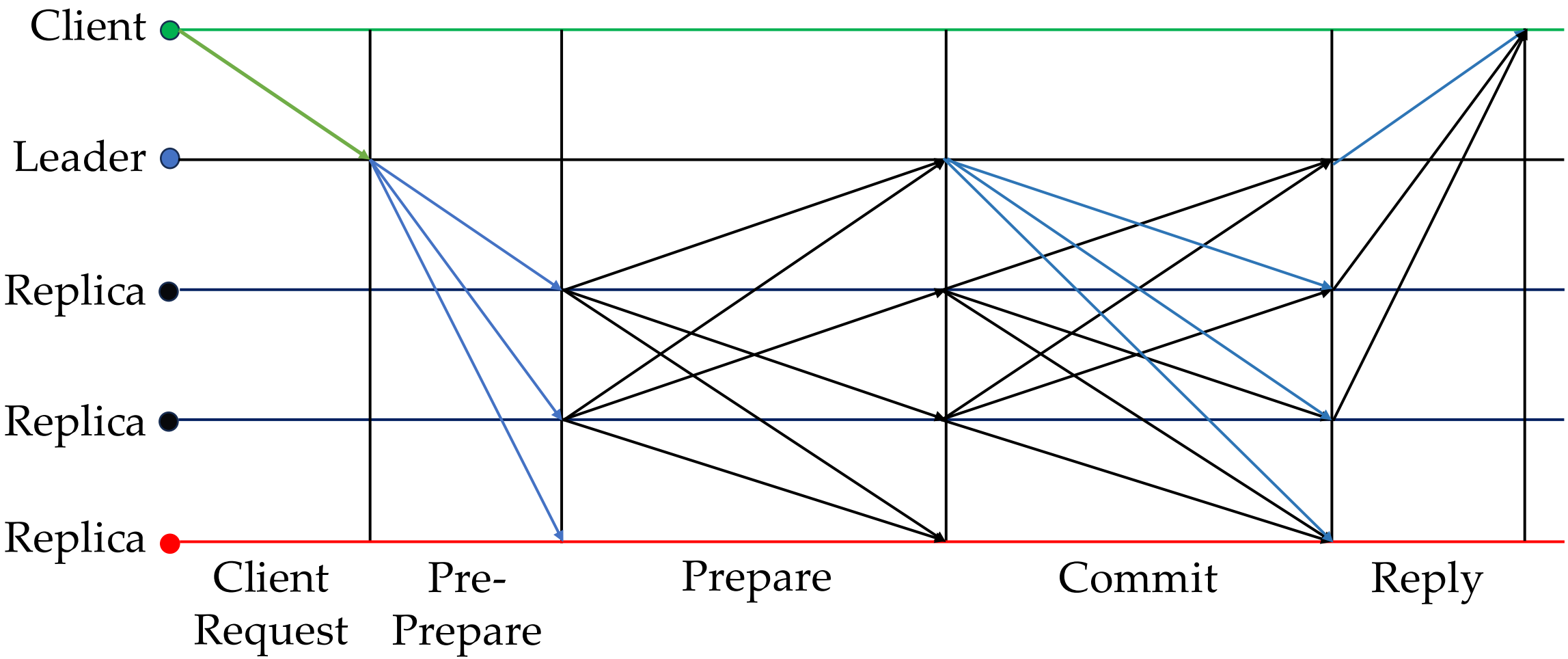
- Parallelizing Consensus
- RCC

Optimizing PBFT

- Today, our goal is to continue optimizing PBFT.

- Until now, we added to PBFT:
  - Pipelining
  - Out-of-Order Message Processing
  - Speculation
  - Parallelism

PBFT Protocol

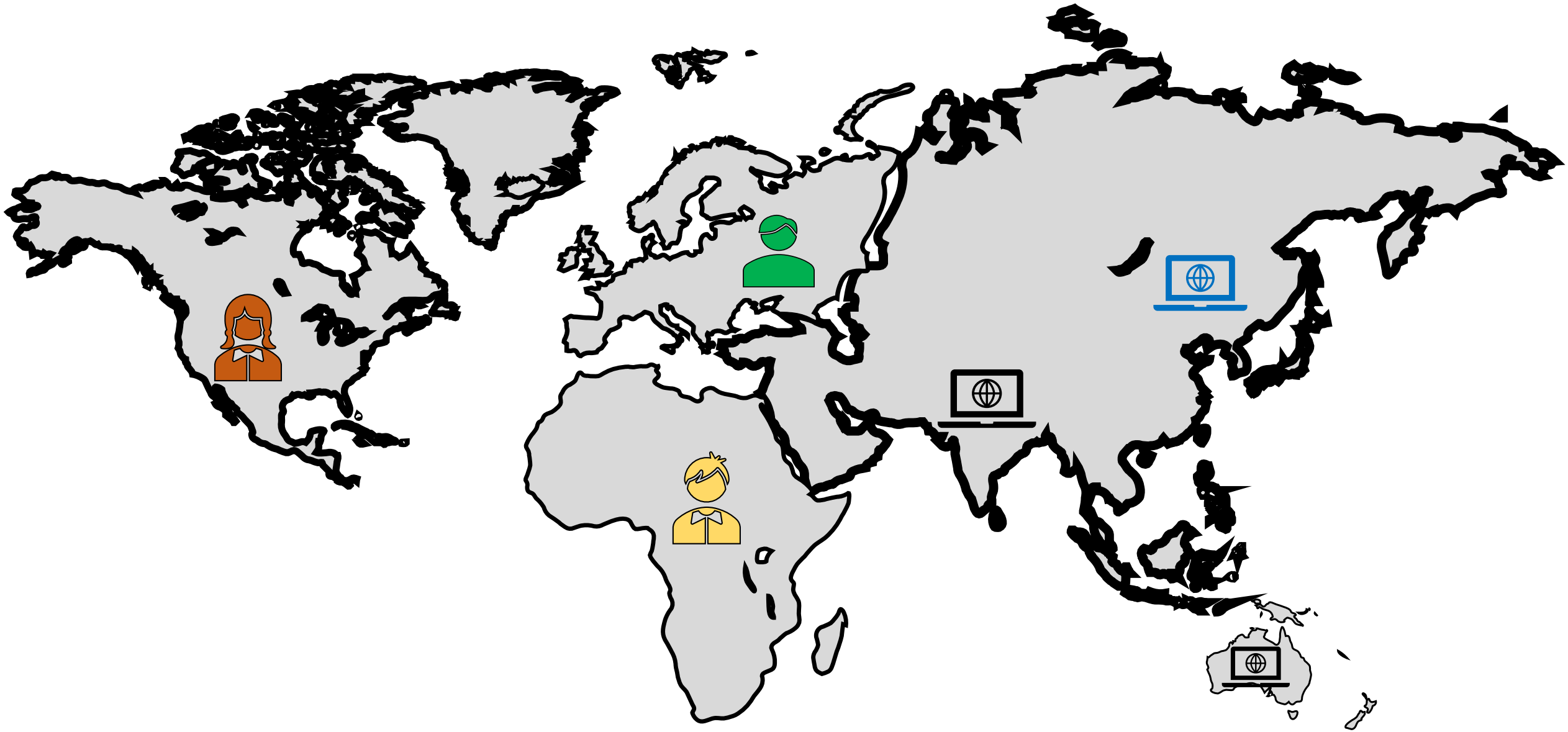


What can we do next?

What can we do next?

- We can apply clustering in consensus to reduce some costs.

Where are replicas located?



Impact of Replicas Deployment

- What is the impact of running consensus among these replicas that are located all over the globe?

Impact of Replicas Deployment

- What is the impact of running consensus among these replicas that are located all over the globe?
- High Latency
- Low throughput

Impact of Replicas Deployment

An experiment showing cost of deploying replicas across the globe on Google Cloud.

	Ping round-trip times (ms)						*Bandwidth* (Mbit/s)					
	O	*I*	*M*	*B*	*T*	*S*	*O*	*I*	*M*	*B*	*T*	*S*
Oregon (*O*)	≤ 1	38	65	136	118	161	7998	669	371	194	188	136
Iowa (*I*)		≤ 1	33	98	153	172		10004	752	243	144	120
Montreal (*M*)			< 1	82	186	202			7977	283	111	102
Belgium (*B*)				≤ 1	252	270				9728	79	66
Taiwan (*T*)					≤ 1	137					7998	160
Sydney (*S*)												

High!

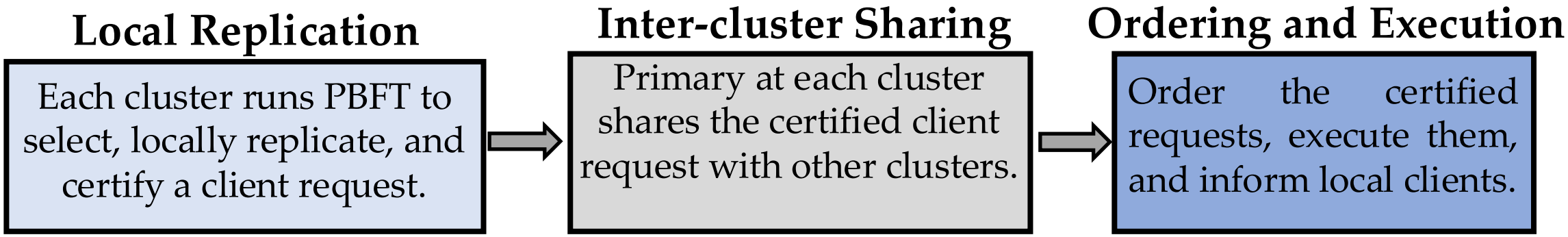
Low!

Geo-scale BFT Consensus

- **GeoBFT [VLDB'20]** reduces the cost of PBFT consensus by clustering replicas.

- System is fully-replicated, but all clusters work independently.

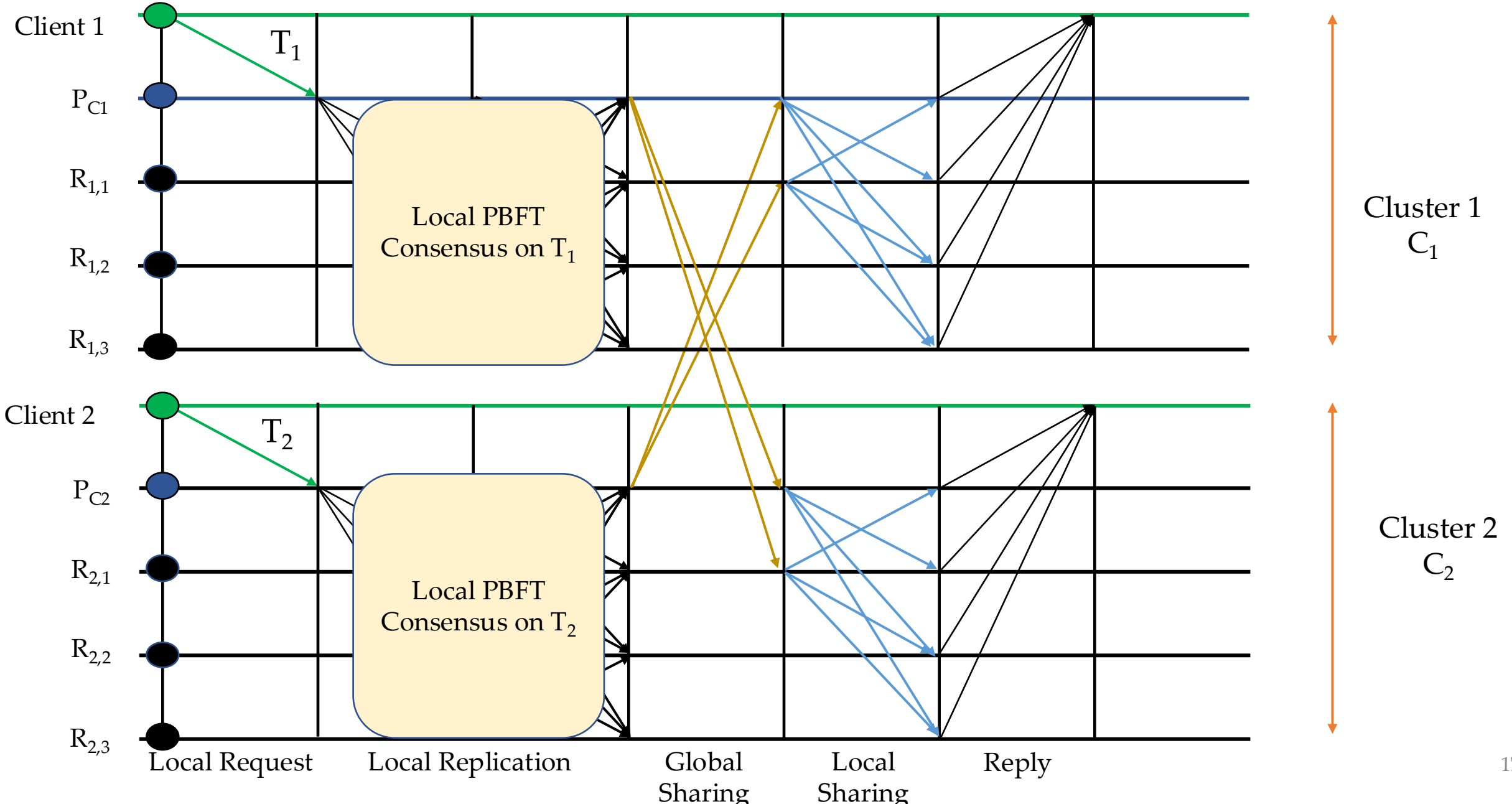
GeoBFT Protocol Overview



GeoBFT Protocol Overview

- **Goals of GeoBFT:**
  - Take Topology into consideration.
  - Run Parallel Consensus at each cluster.
  - Ensure cheap inter-cluster communication.

GeoBFT Protocol Overview

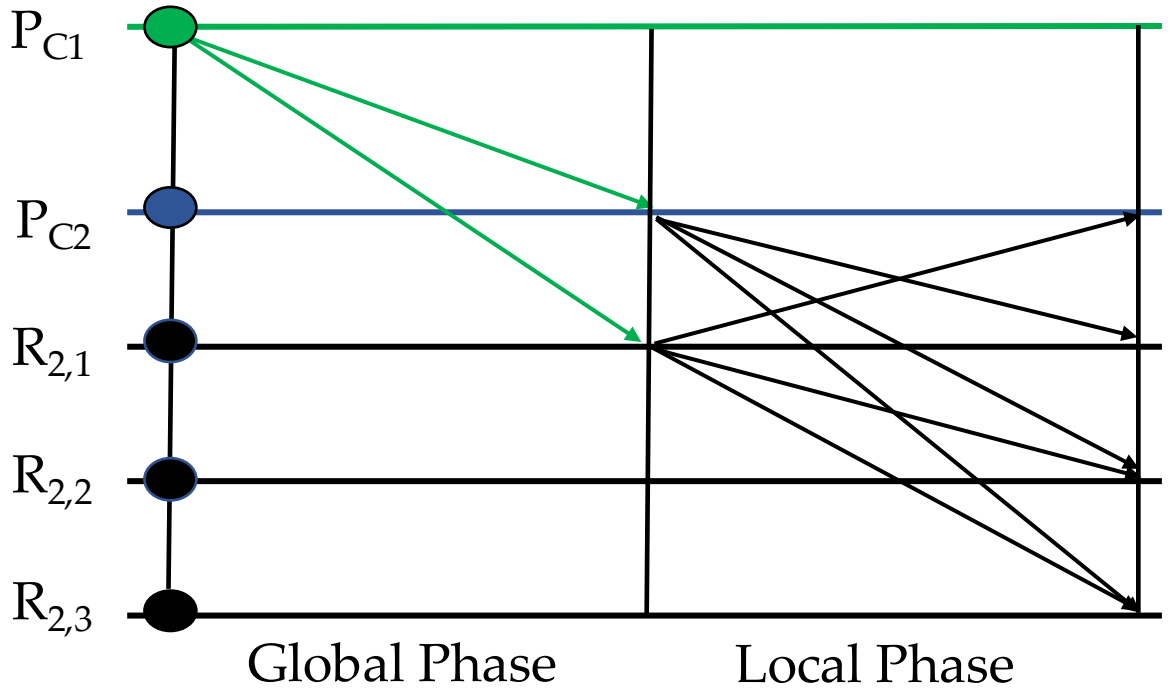


Local Replication

- Run **any** consensus protocol of your choice → PBFT, PoE
- Ensure **each cluster has n = 3f+1** replicas, where at most **f** replicas are Byzantine.
- Once a cluster has committed its request, it needs to **exchange requests** with other clusters → Full replication.

Inter-Cluster Communication

Inter-Cluster Communication



The Primary P_{C1} sends a certificate that includes the client request and commit messages from n-f replicas of Cluster C_1 .

Inter-Cluster Communication

- Make the leader create a certificate, which proves that **n-f** replicas committed the request.
- Leder sends the certificate to **f+1** replicas.
- The **f+1** replicas forward the received certificate to everyone in their cluster.

Ordering and Execution

Ordering and Execution

- GeoBFT orders requests deterministically.
- For $\mathbf{i < j}$, requests of Cluster $\mathbf{C_i}$ are executed before requests of cluster $\mathbf{C_j}$.
- For example: requests of $\mathbf{C_1}$ are executed before $\mathbf{C_2}$.

Challenges for GeoBFT

Challenges for GeoBFT

- Replication Factor.
- Malicious leader.
- Lack of Requests.
- Cluster Failure and Sizes.

Replication Factor

Replication Factor

- PBFT's replication factor → **n = 3f+1 replicas**.
- GeoBFT requires **n = 3f+1 replicas per cluster**.
- Assume 4 clusters and each with 4 replicas:
  - PBFT → n = 16, f = 5.
  - GeoBFT → Maximum 4 failures and at most one failure per cluster.

Malicious Leader

Malicious Leader

- What if the malicious leader impact inter-cluster communication.:
  - Does not send certificate to all clusters.
  - Does not send certificate to f+1 replicas in each cluster.

Malicious Leader

- What if the malicious leader impact inter-cluster communication.:
  - Does not send certificate to all clusters.
  - Does not send certificate to f+1 replicas in each cluster.

- **Detect** the malicious behavior
  - Epoch based design like in RCC.

- **Request replacement** of the malicious leader by broadcasting the request to every replica in the cluster of malicious leader.

Lack of Requests

- Not every cluster will always have some request to send?

Lack of Requests

- Not every cluster will always have some request to send?

- Send a **No-op certificate** → certificate created with the help of other replicas.

Cluster Size and Failure

- Do all clusters need to be of same size?

Cluster Size and Failure

- Do all clusters need to be of same size?
  - No, clusters can have different sizes.

Cluster Size and Failure

- Do all clusters need to be of same size?
  - No, clusters can have different sizes.

- Can a full cluster fail?

Cluster Size and Failure

- Do all clusters need to be of same size?
  - No, clusters can have different sizes.

- Can a full cluster fail?
  - GeoBFT cannot handle a cluster failure.

What else can we do next to GeoBFT?

What else can we do next to GeoBFT?

- How about we apply the logic of sharding?

- Each shard manages distinct data items.

- Two types of transactions:
  - Intra-shard → Support parallel processing.
  - Inter-shard → Require communication among the shards.

RingBFT Consensus

- **RingBFT [EDBT'22]** reduces the cost of PBFT consensus by sharding replicas.

- System is no longer fully-replicated.

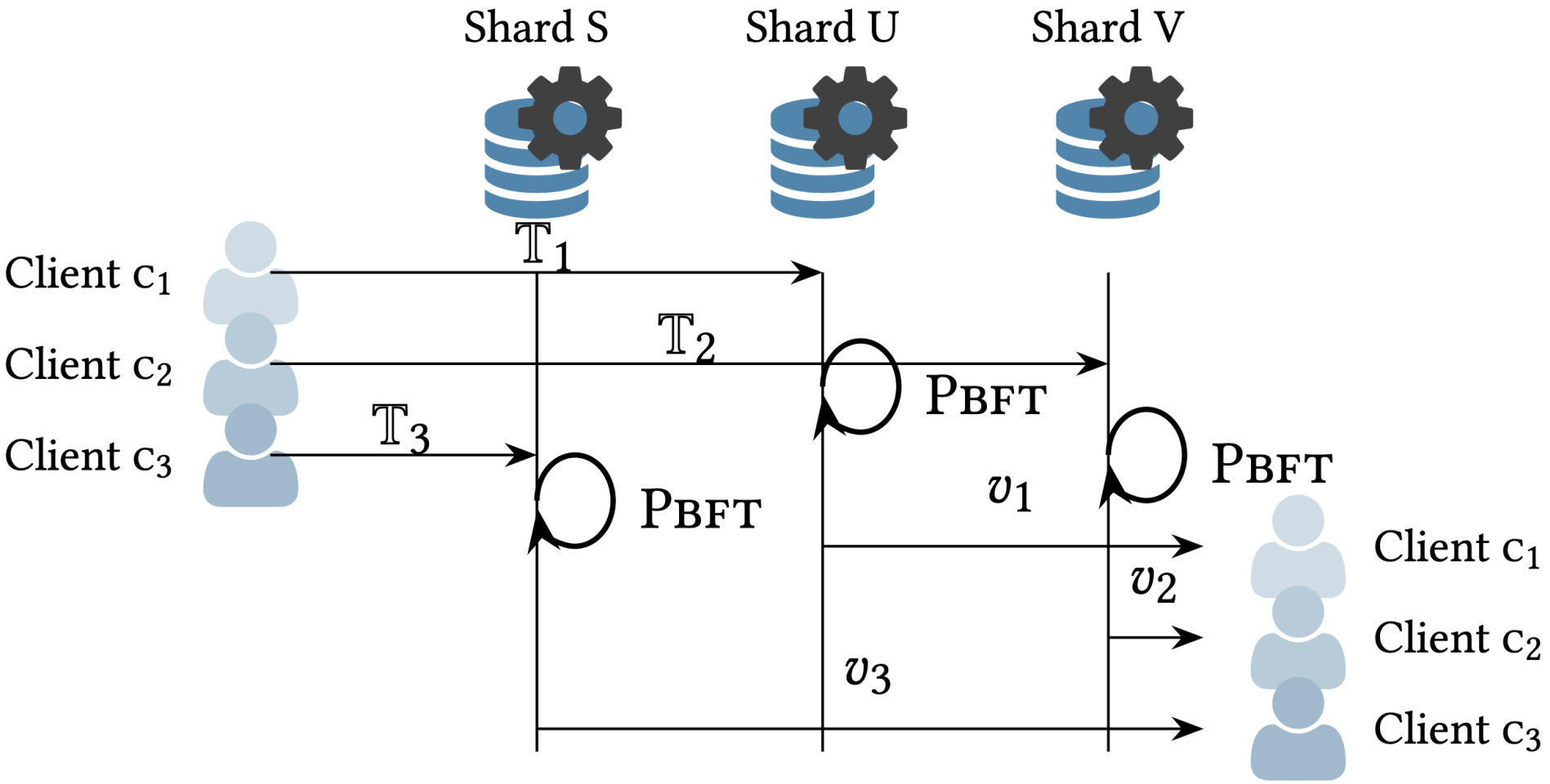
RingBFT Consensus

- **RingBFT [EDBT'22]** reduces the cost of PBFT consensus by sharding replicas.

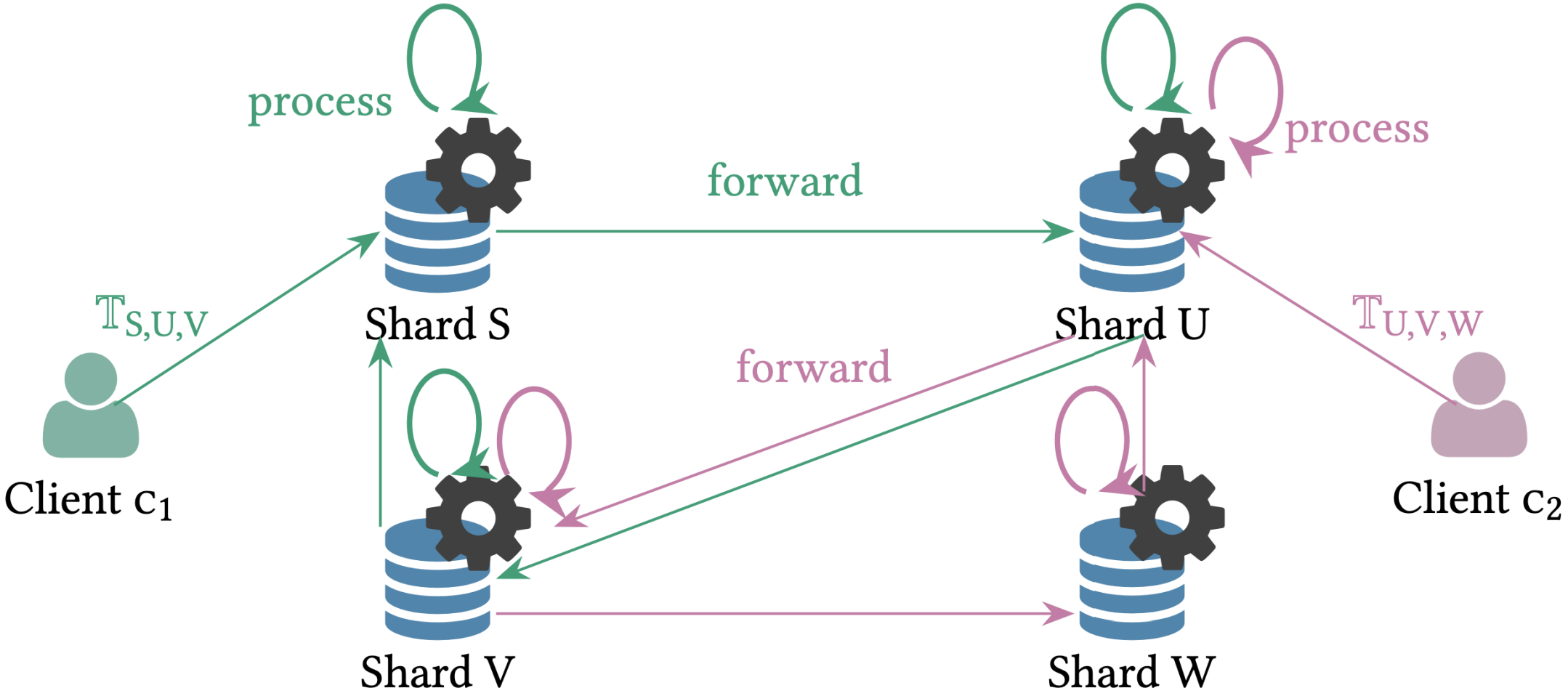
- System is no longer fully-replicated.

- RingBFT suggests going around the shards in a ring with processing the transaction at each step.
  - Order the transaction locally.
  - If a inter-shard transaction → forward the Read-Write sets and updates to the next shard in the ring.
  - The direction of ring never changes.

RingBFT Intra-Shard Consensus



RingBFT Inter-Shard Consensus



RingBFT Inter-Shard Consensus

- The goal is to make as many few rotations around the ring as possible.
- Each rotation does: Process, Forward, and Re-transmit.
- The Ring order is fixed!