Dissecting BFT Consensus: In Trusted Components we Trust!

Suyash Gupta

UC Berkeley

Sajjad Rahnama

ExpoLab
UC Davis

Shubham Pandey

ExpoLab
UC Davis

Natacha Crooks

UC Berkeley

Mohammad Sadoghi

Expolab
UC Davis

Why Should this Talk Interest you?

Bad News

Trusted Hardware

cannot be used

to efficiently reduce replication

factor of BFT protocols to 2f+1.

Good News

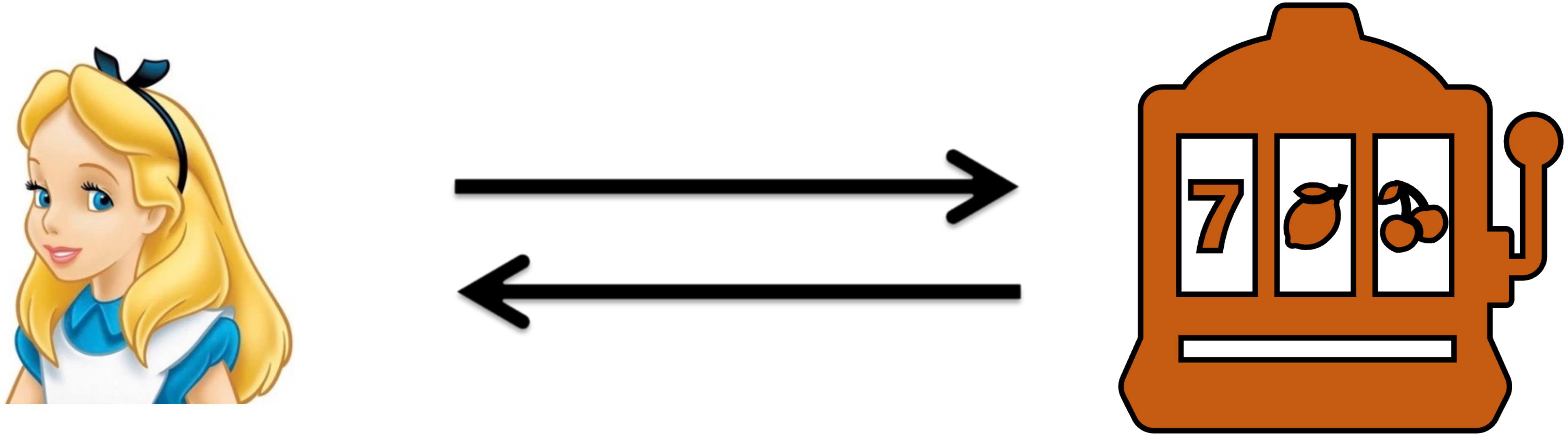
Trusted Hardware

can be used

to design more efficient and

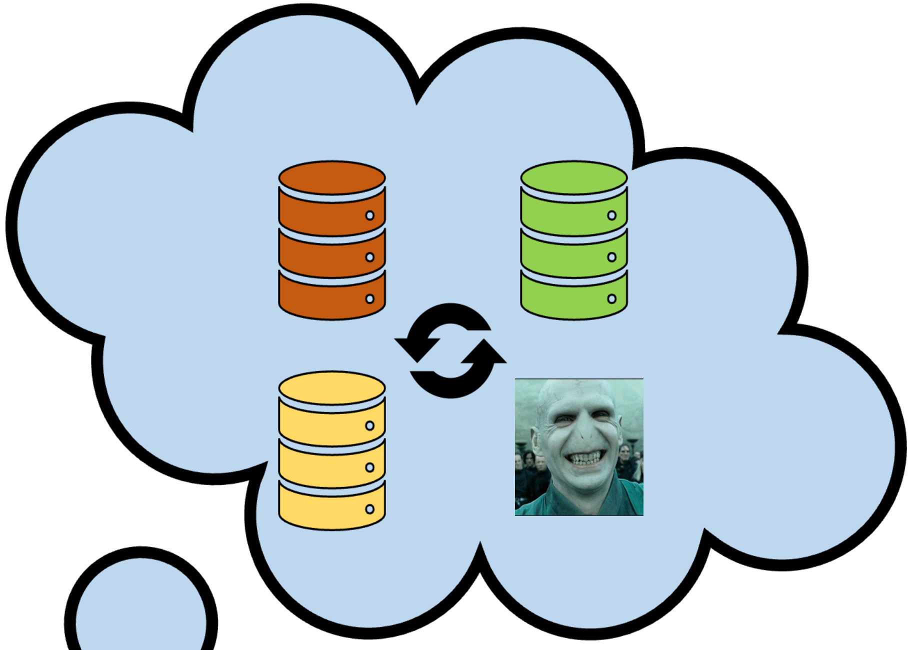
scalable 3f+1 BFT protocols.

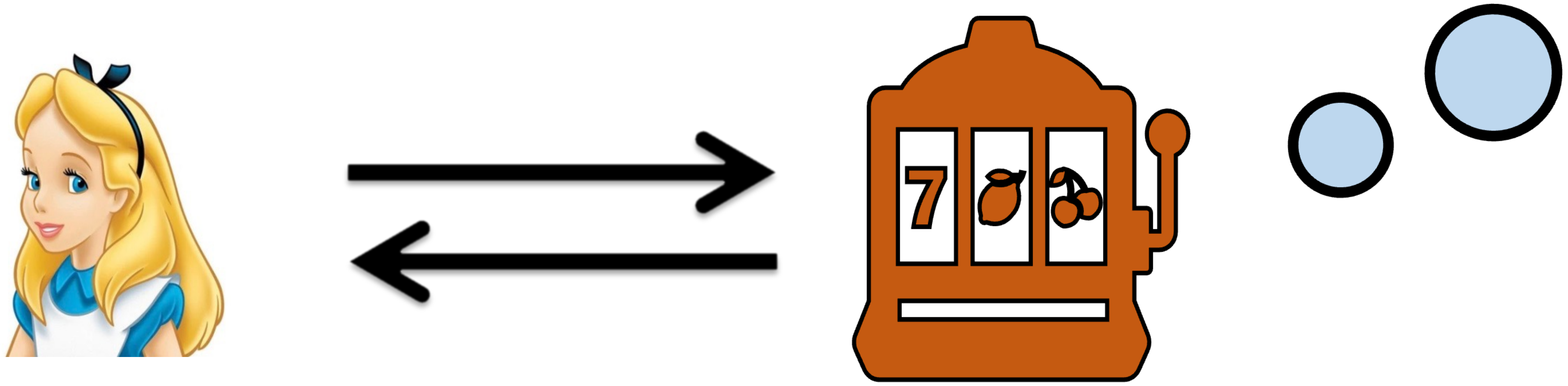
Replicated State Machine



Replicated State Machine

- **Safety** → Consistent log of operations.
- **Liveness** → Replicas should make progress.
- **Responsiveness** → Client should receive response.



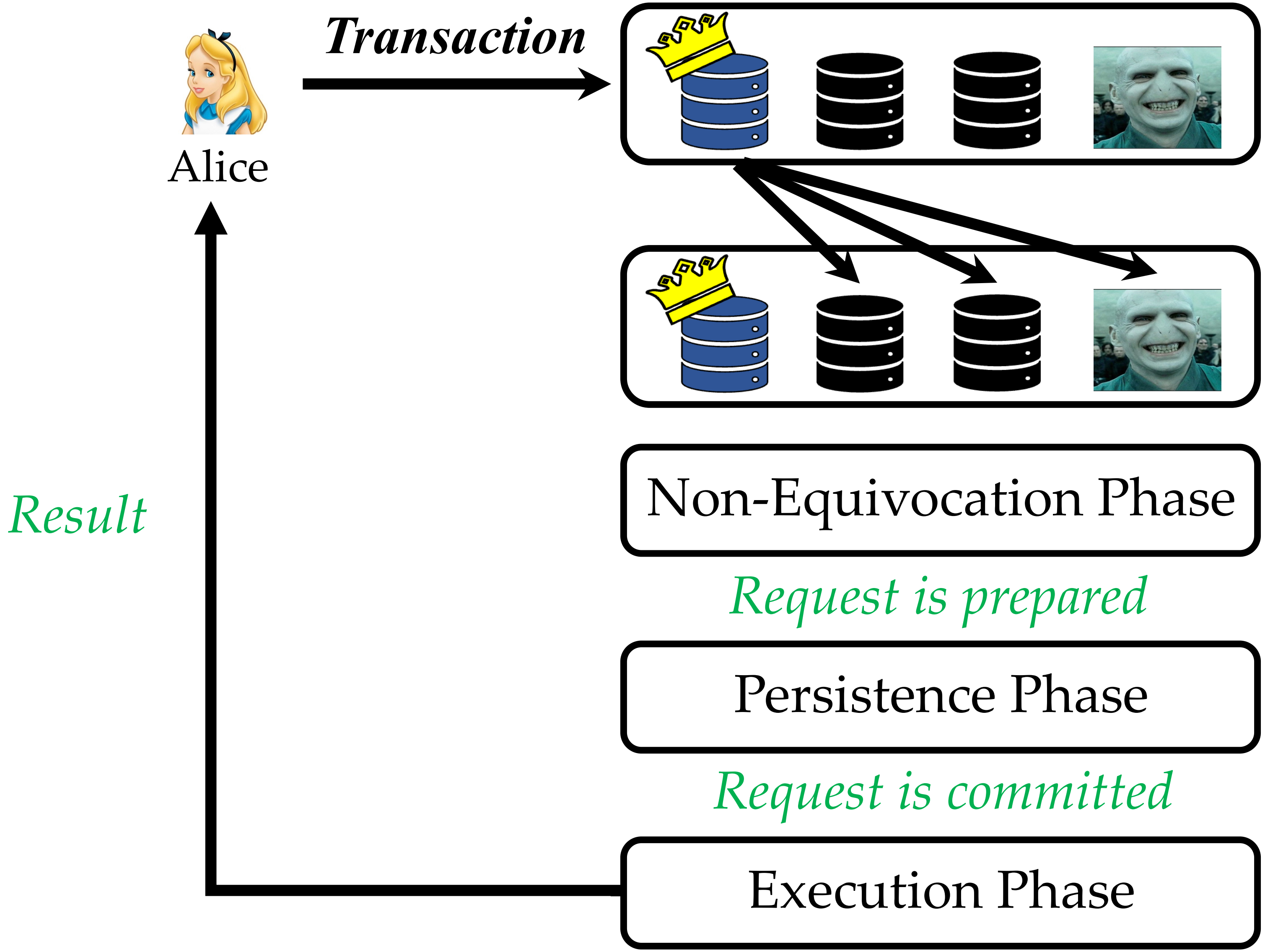


Byzantine Fault Tolerant RSM

n replicas & at most f byzantine → **n >= 3f+1**

Run Byzantine Fault Tolerant (BFT) Consensus

Byzantine Fault Tolerance Consensus



n = 3f+1 replicas

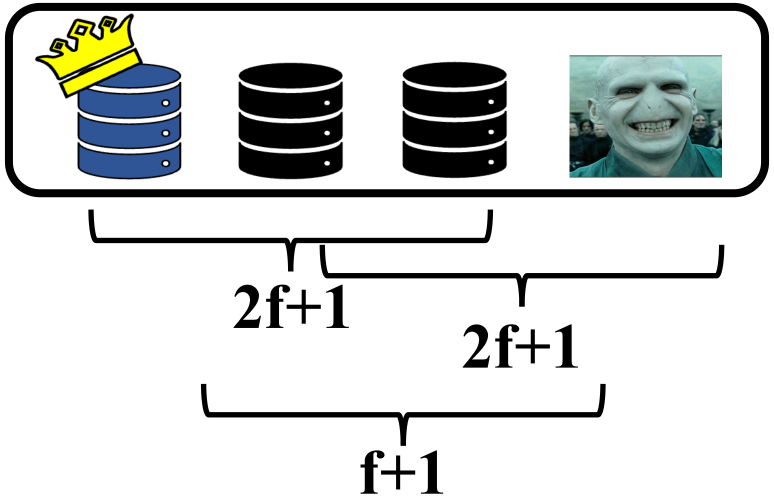
Non-Equivocation

Create a ***Prepare Quorum***:

No two prepare quorums can exist for

different transactions at the same sequence number.

Every quorum **needs to intersect** in at least one honest replica.



Persistence

If a new leader is elected,

RSM should ensure that

previously committed requests persist.

Execution

Client needs **f+1** matching responses.

Ensures execution by **one honest** replica.

Proof of request commitment **not sufficient**.

The Ugly Side of BFT

Crash Fault Tolerant Systems

2f+1 replicas

Byzantine Fault Tolerant Systems

3f+1 replicas

Equivocation is root cause of higher replication factor

Maybe Trusted Hardware Can help?

Keystone

intel SGX

Trusted Byzantine Fault-Tolerance Consensus

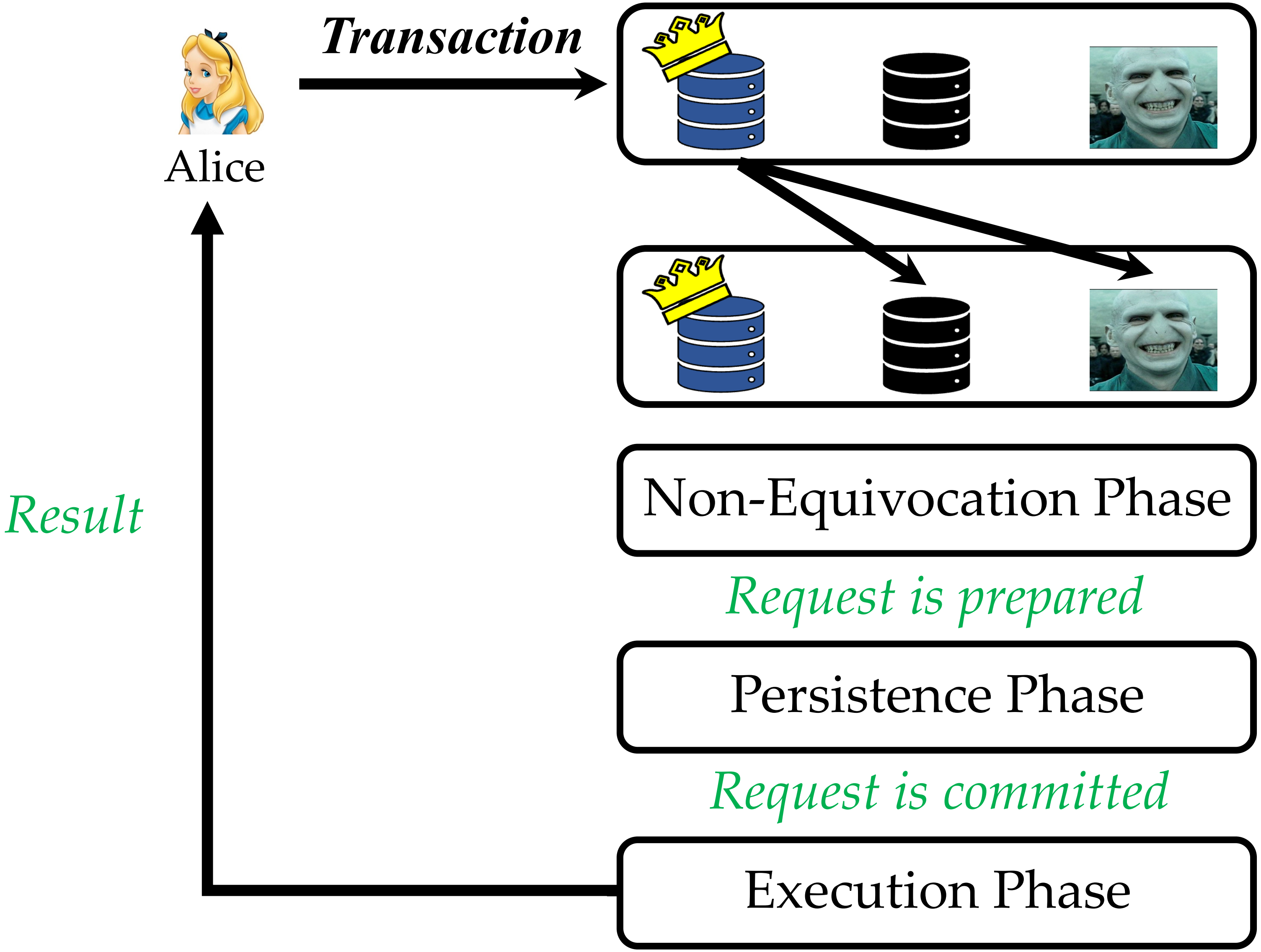
Trusted component *attest* order of each transaction.

Replicas cannot equivocate.

A2M, TrInc, MinBFT, MinZZ, CheapBFT, Hotstuff-M, Damysus

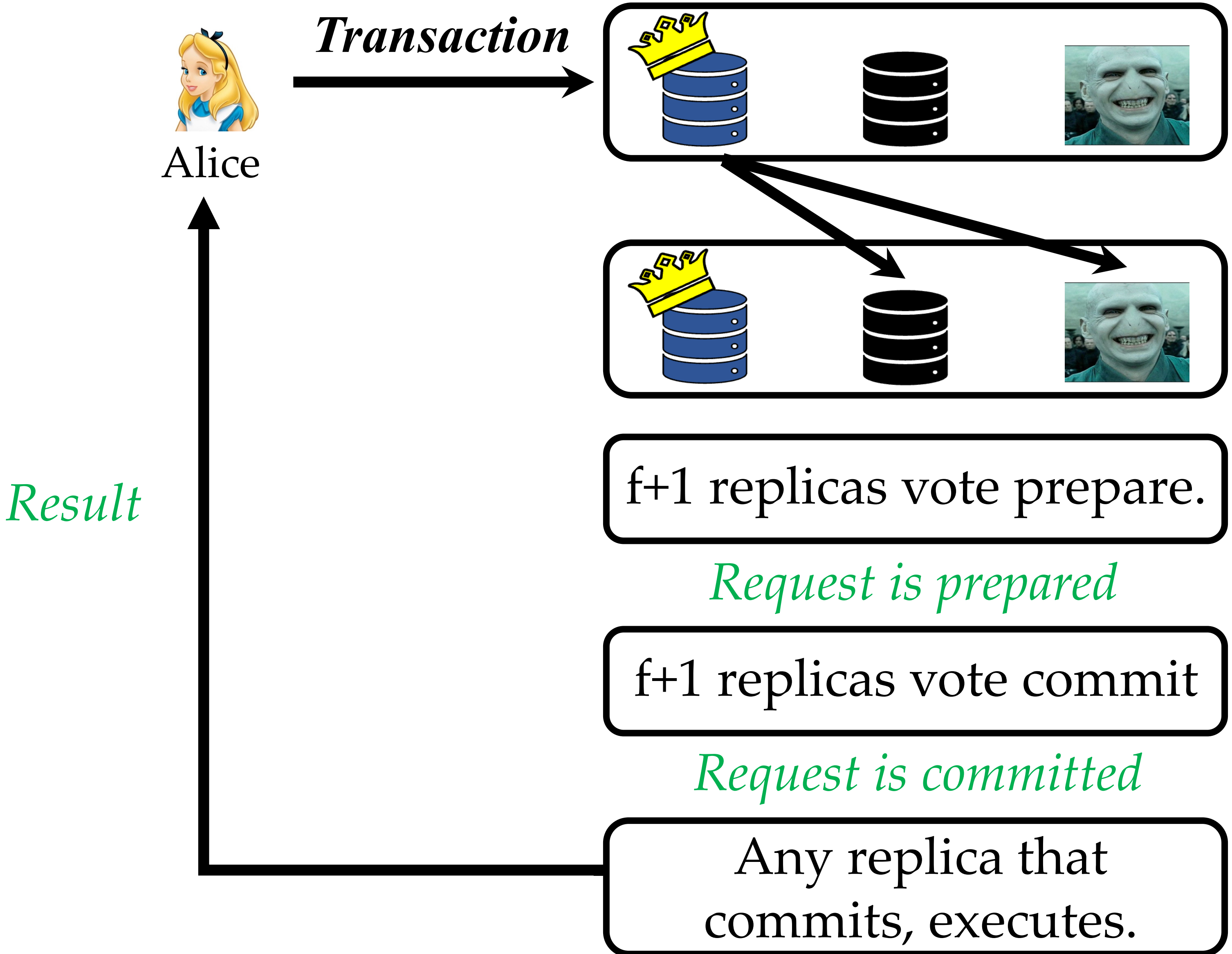
Trust-BFT protocols → 2f+1 enough for safety

Trust-Byzantine Fault Tolerance Consensus



$n = 2f{+}1$ replicas

Trust-Byzantine Fault Tolerance Consensus



So Are We Done?

Unfortunately No!

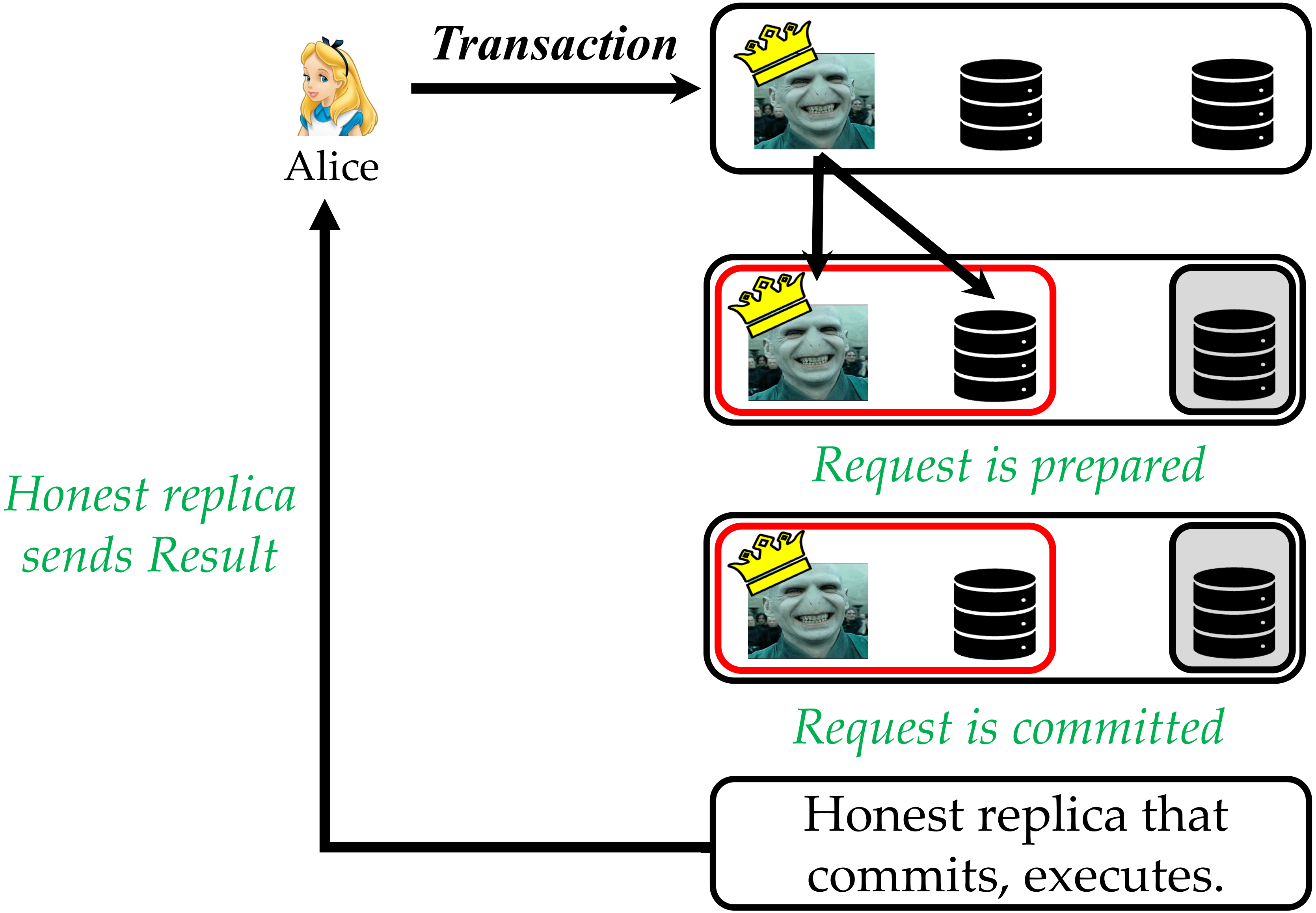
Hidden Pitfalls with Trust-BFT Protocols

- **Algorithmic Pitfall**
  - Limited Responsiveness
  - Loss of Safety under Rollbacks
  - Lack of Parallelism

- **Measurement Pitfall**
  - Instead of focusing on *reducing* replication → Focus on *increasing* Throughput per Machine.

Limited Responsiveness

Quorum Size = f+1 = 2

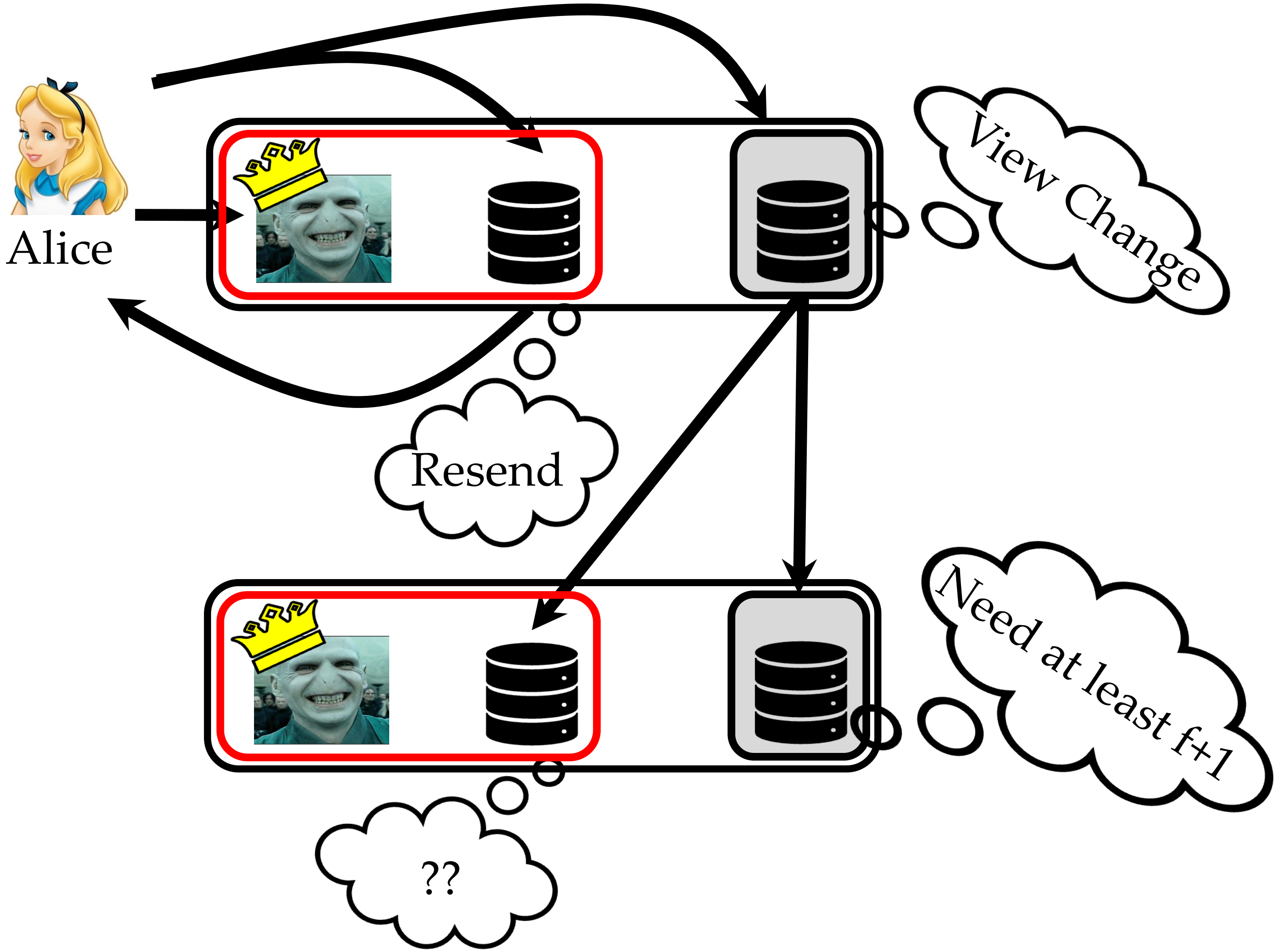


Alice Stuck!

Alice needs **f+1 = 2** matching responses.

Alice receives only 1 response.

No progress for Alice



Lack of Parallelism

- Every message sent requires an attestation bound to specific sequence number.
- Replicas cannot run consensus on two transactions in parallel!
- **We show** that despite 2f+1 replicas, Trusted-BFT protocols are slower than BFT.

Loss of Safety under Rollbacks

- Trusted Enclaves can be rollbacked!
  - On enclave rollback, safety cannot be guaranteed.

- Possible Solution? Make use of TPMs or persistent counters!
  - Too slow → 180ms per access.
  - Very few writes → TPMs allow at most 1 million writes.
  - Trust-BFT protocols require O(n) accesses per consensus phase.

Solution →

FlexiTrust Protocols

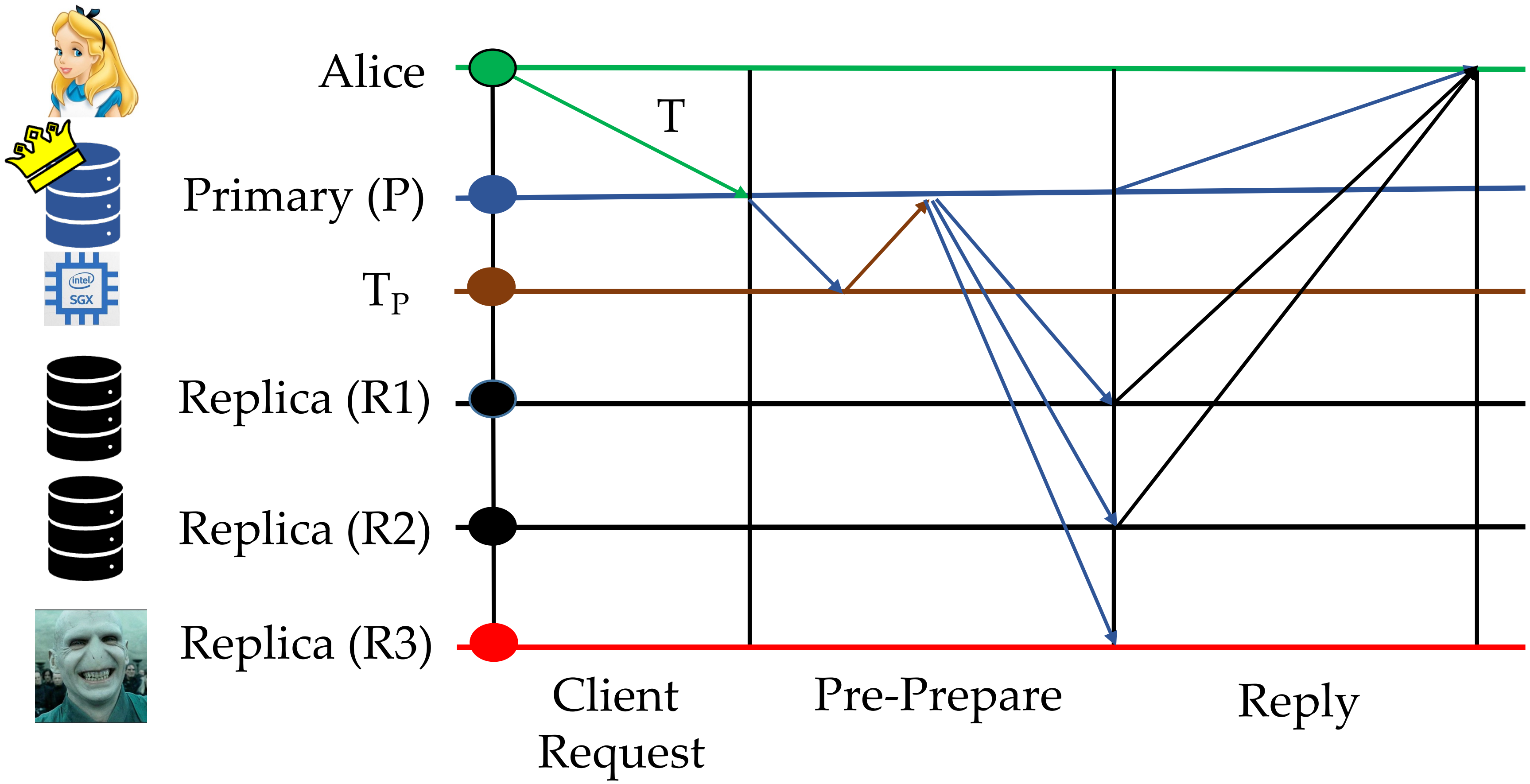
- A novel suite of protocols.
- Guarantee both liveness and responsiveness.
- Require access to trusted component only once per consensus.
  - Employing TPMs to avoid enclave rollbacks is now much less expensive!

Magical Ingredients behind
FlexiTrust Protocols

- Switch back to replication factor 3f+1.
  - Larger Quorums guarantee responsiveness.

- Trusted hardware accessed only by the primary before sending proposal.
  - Guarantees non-equivocation.
  - Permits replicas to participate in multiple consensus invocations in parallel.
  - Helps to reduce phases and communication.

Flexi-ZZ Protocol!



Single phase, Linear, Handles f failures, Only needs Trusted counters.

Evaluation on ResilientDB*

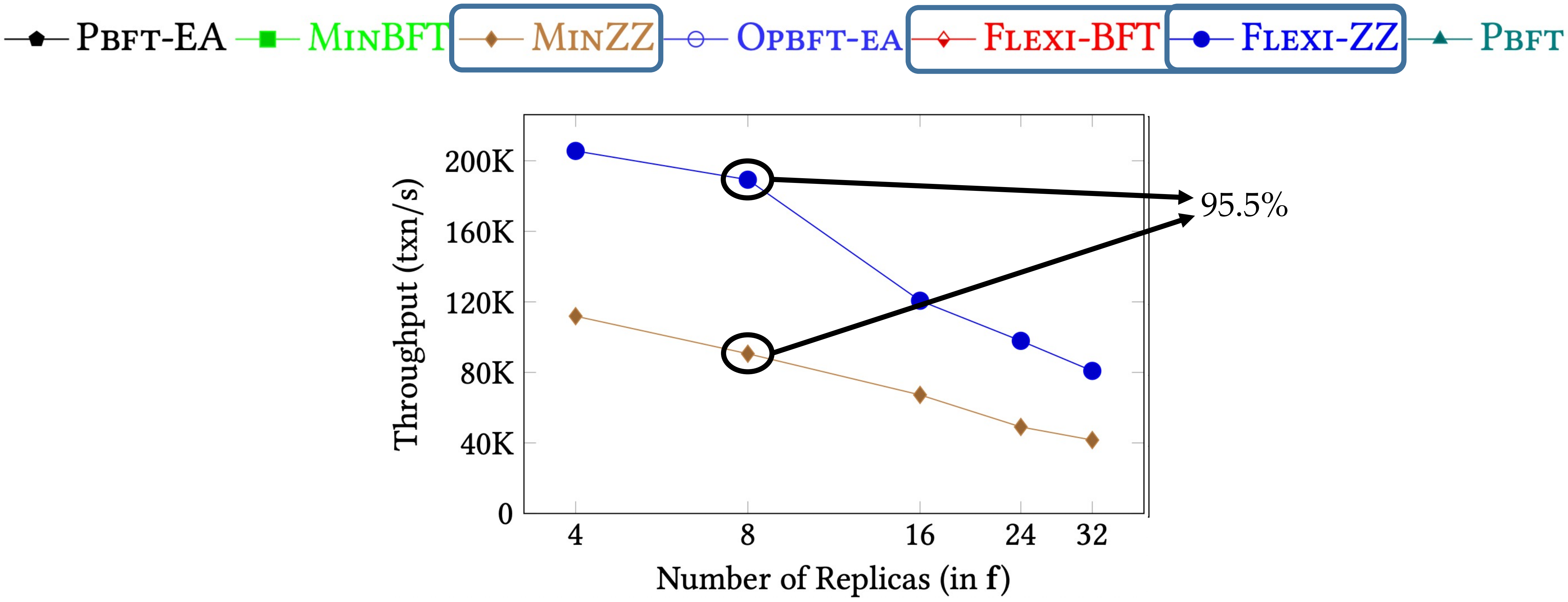
*https://resilientdb.com/

Throughput per Machine

Replicas (in f)	Total Replicas (in **n**)		Protocols	
	FLEXI-ZZ	MINZZ	FLEXI-ZZ	MINZZ
4	13	9	15813	12431
8	25	17	7570	5329
16	49	33	2462	2038
24	73	49	1341	1002
32	97	63	834	640

- MinZZ → Single phase like FlexiZZ but n >= 2f+1.
- For these experiments, we deployed up to 80k clients.

Scalability



Number of replicas (f=8)

- N = 17 → PBFT-EA, MinBFT, MinZZ, OPBFT-EA
- N = 25 → PBFT, FlexiBFT, FlexiZZ

➢ Conclusions:

- Simply reducing replication will not yield higher throughput.
- Existing Trust-BFT protocols limit responsiveness and scalability.
- **FlexiTrust** protocols advocate meaningful application of BFT consensus.

➢ Reach me:

- **Twitter:** suyash_sg
- **Email:** suyash.gupta@berkeley.edu
- **Web:** https://gupta-suyash.github.io/