

# CS 60002: Distributed Systems

## T1: Course Introduction

Department of Computer Science  
and Engineering



INDIAN INSTITUTE OF TECHNOLOGY  
KHARAGPUR



**Sandip Chakraborty**  
[sandipc@cse.iitkgp.ac.in](mailto:sandipc@cse.iitkgp.ac.in)

# Teaching Assistants



**Prasenjit Karmakar**

[prasenjitkarmakar52282@gmail.com](mailto:prasenjitkarmakar52282@gmail.com)



**Aritra Hota**

[aritra828207@gmail.com](mailto:aritra828207@gmail.com)



**Sourajit Bhattacharjee**

[sourajit.bhattacharjee@kgpian.iitkgp.ac.in](mailto:sourajit.bhattacharjee@kgpian.iitkgp.ac.in)

# Pre-requisite for this course

- Data Structures and Algorithms
- Operating Systems
- Computer Networks
- Database Systems

# Course Structure

- A blend of lecture-based study and research-based study with Programming Assignments / Short Projects
- **Lectures**
  - Discuss the fundamentals – algorithms and systems-aspect of the course
- **Term Projects**
  - Two types of term projects – **Research-based** and **Survey-based**
  - **Research-based:** Explore a recent research topic and come up with some experimental studies/new algorithms/new system designs, etc.
  - **Survey-based:** Read some (at least 3N papers for N-member groups) recent research papers on a topic assigned and come-up with a survey paper
  - Group size: 4-6 members
  - Fill up this Google form to submit your group information by **January 10, 2024**: <https://forms.gle/5eFSEcaqQ8fxQy698>

# Course Structure

- **Programming Assignments / Short Projects**

- Roughly four assignments during the semester
- Form a group of 3-4 students. Submit the group information through this Google form by 10th January 2024: <https://forms.gle/3BNEV2q4XdGBXkCN8>
- Create a private GitHub repo; collaborate among the group members to build the assignment solutions.
- You may discuss among your friends, TAs, instructors, but the code should be on your own
- We'll be having a **target deadline** for each of the assignments. You need to make the GitHub repo public by the target deadline and share it with the instructor and the TAs.
- You are allowed to edit your code after the target deadline; but we can evaluate it any day after this deadline – your marks for the assignment will be based on this evaluation only.
- There would be no extension of this target deadline

# Course Structure

- **Programming Assignments / Short Projects**

- Your assignment will be evaluated exactly once. We'll raise an issue on the GitHub if there is any comments because of which your marks are deducted. However, reevaluation will not be done.
- Your code should run on a standard Ubuntu 20.04 or later. You should include a README file in each submission, that should mention the test environment, sample inputs and outputs clearly.
- You'll not be entertained for any marks based on any changes made after the evaluation date, or any incomplete information in your README file.
- You should also include a Makefile to compile and run your code, and one or more sample inputs with which you have tested the code (under an input directory).
- In case of plagiarism, marks will be given only to the submission with the earliest last modification timestamp.
- Every member of the group should collaborate, which should get reflected in the commit logs.

# Grading

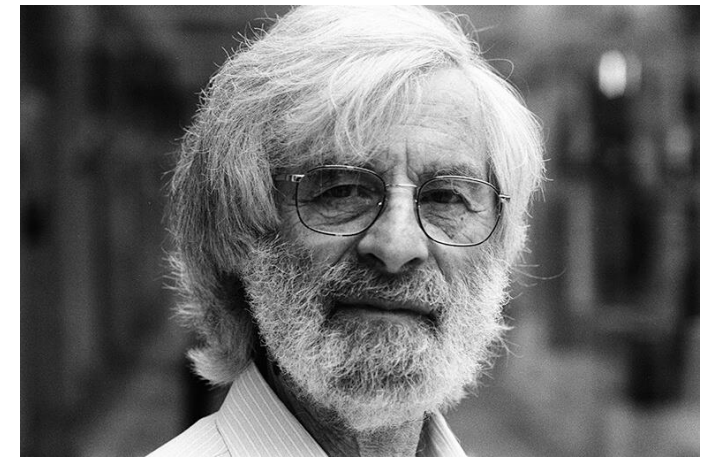
- Mid Sem: 25%
- End Sem: 30%
- Project: 20%
- Assignments: 20%
- Class Attendance: 5%



# What is a Distributed System?

*"A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable"*

-- Leslie Lamport



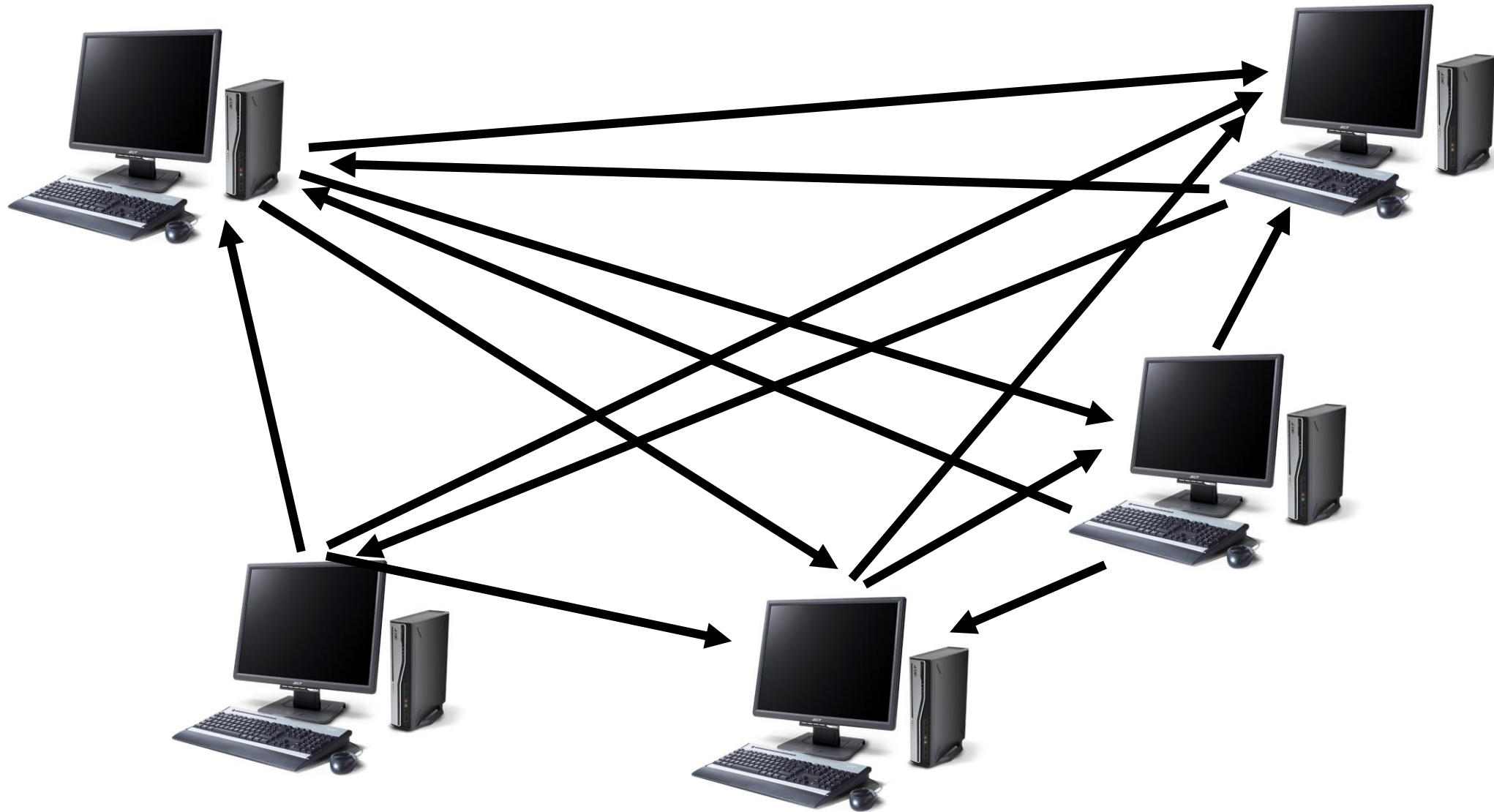


# What is a Distributed System?



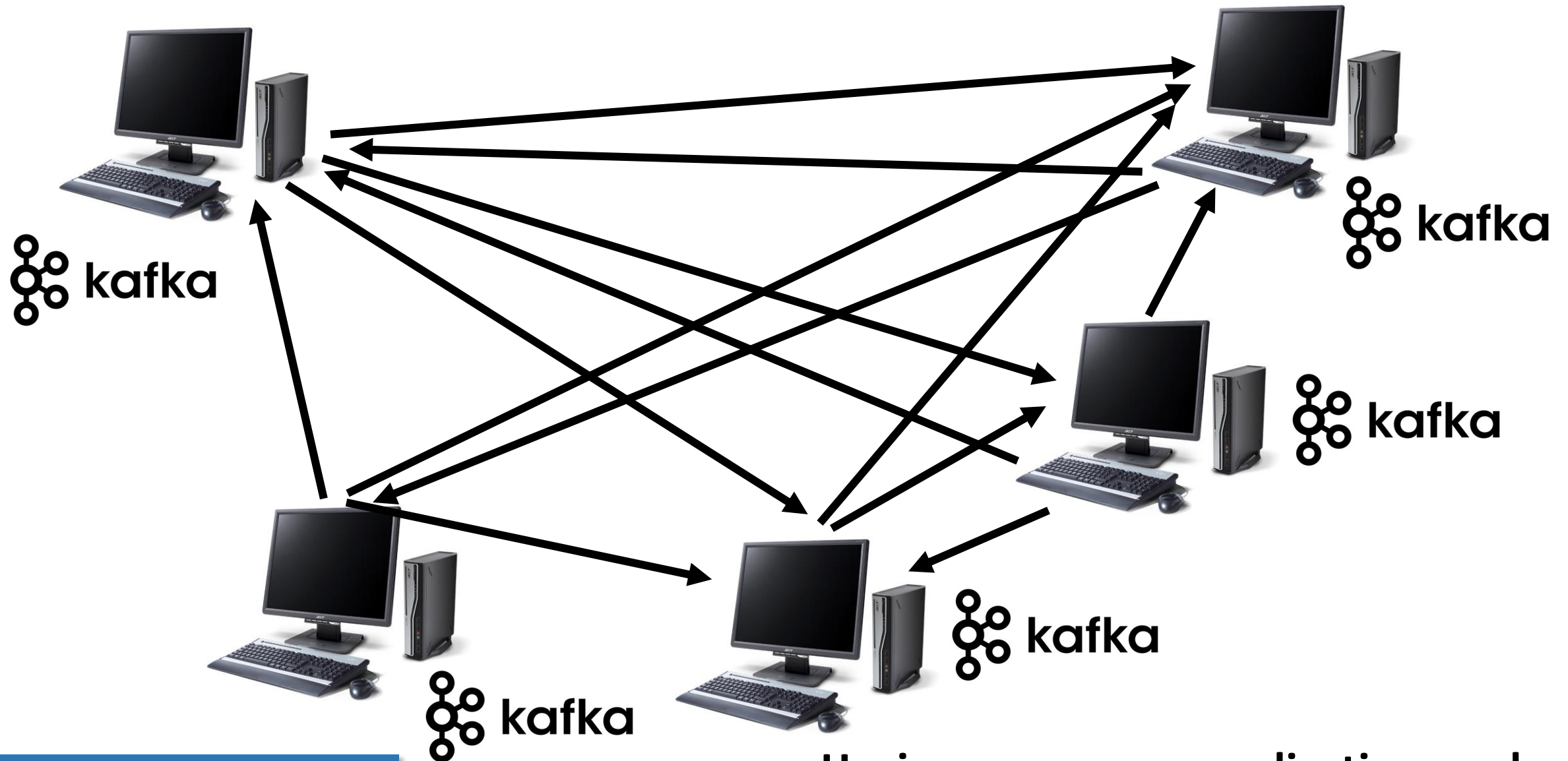
**Multiple Computers**

# What is a Distributed System?



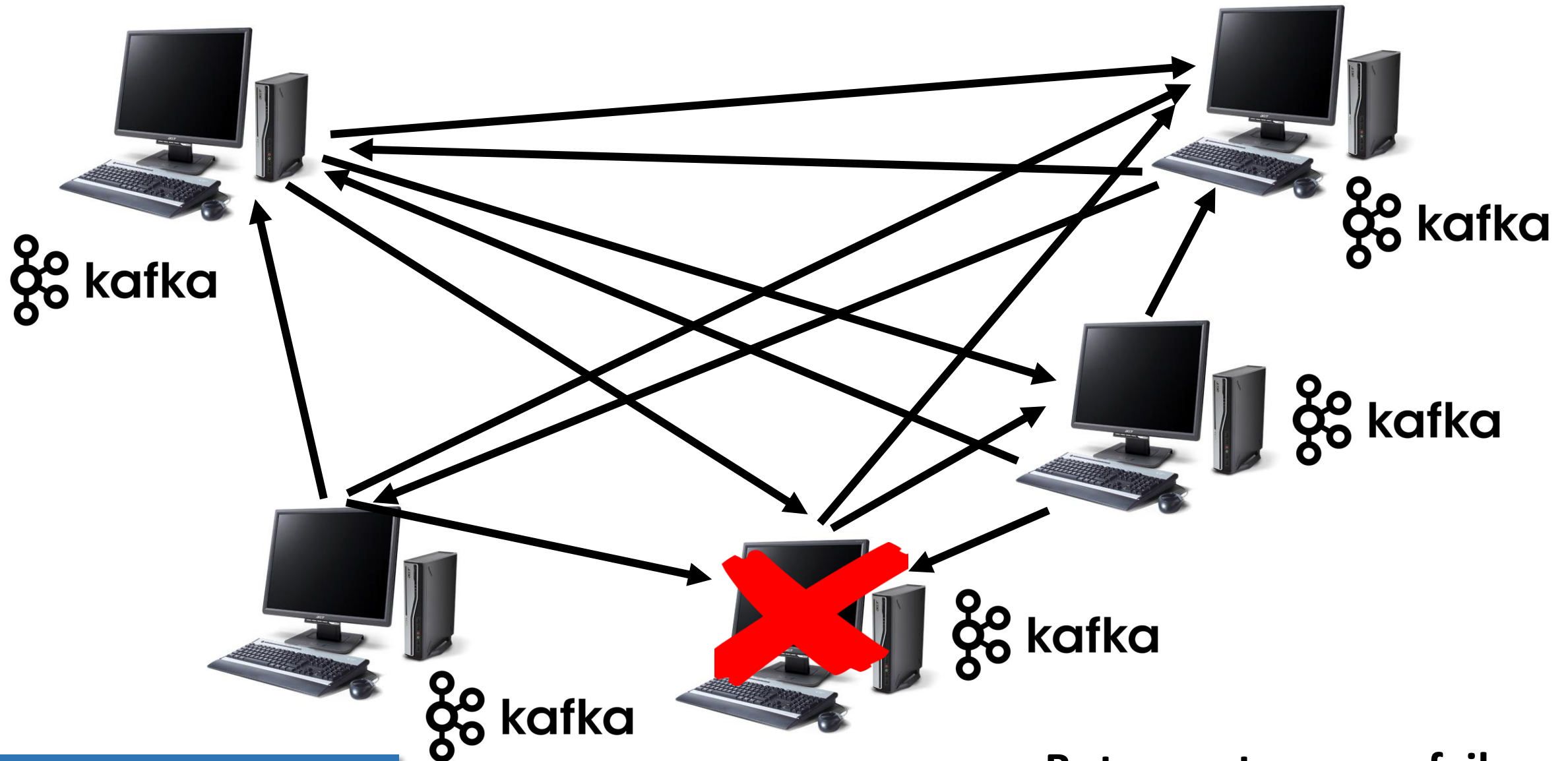
**Wants to talk to each other**

# What is a Distributed System?



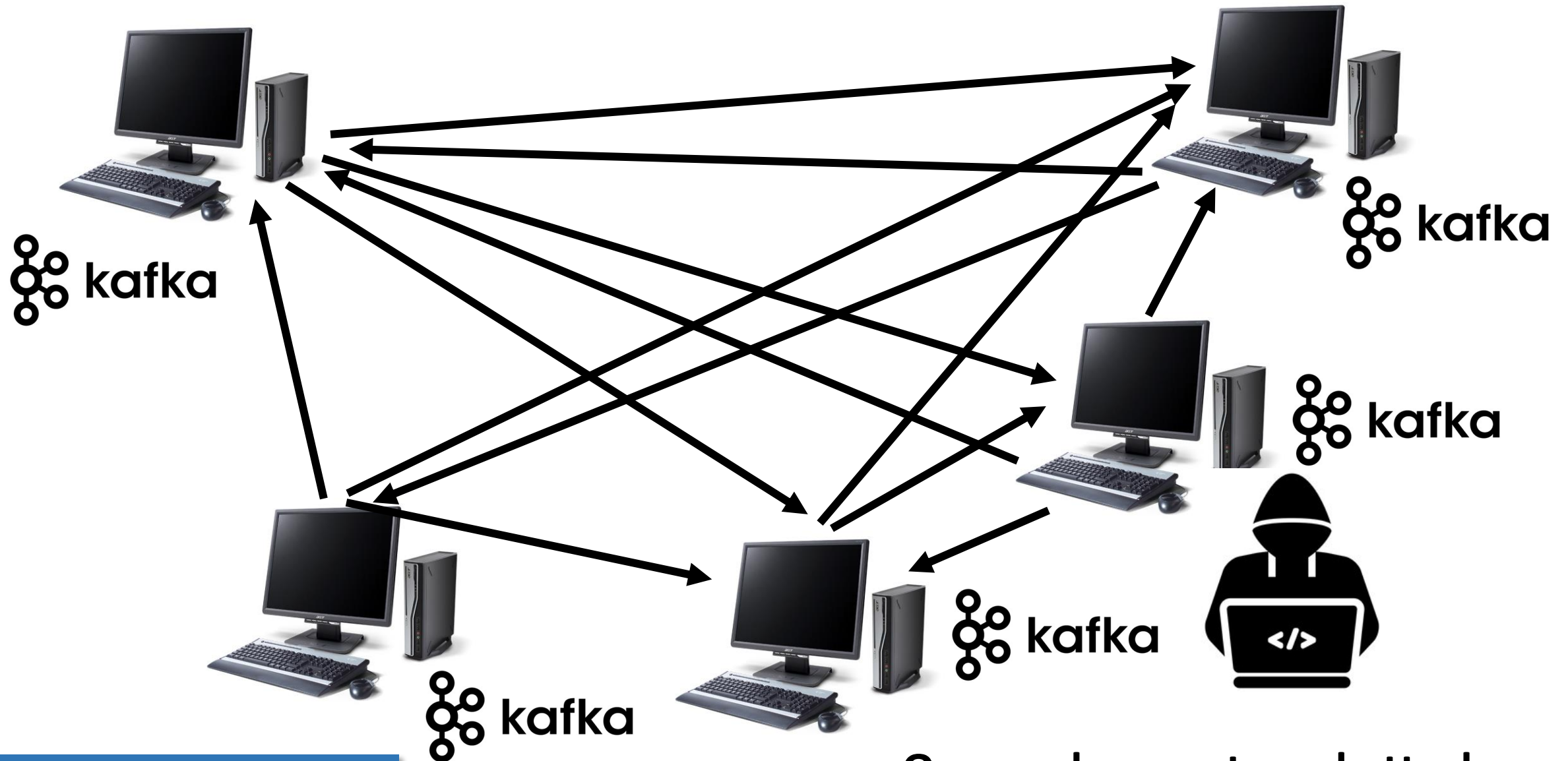
Having a common application goal ...

# What is a Distributed System?



But, a system may fail ...

# What is a Distributed System?



Or, may have external attacks ...

# Examples of Distributed Systems

- Almost every large system that you use ...

The Amazon logo, featuring the word "amazon" in a black, lowercase, sans-serif font, with a curved orange arrow underneath it pointing from the letter 'a' to the letter 'z'.The Flipkart logo, featuring the word "Flipkart" in a blue, lowercase, sans-serif font, with a yellow shopping bag icon containing a blue 'f' and a white swoosh to its right.The Netflix logo, featuring the word "NETFLIX" in a bold, red, uppercase, sans-serif font on a black rectangular background.The Facebook logo, featuring the word "facebook" in a white, lowercase, sans-serif font on a dark blue rectangular background.The Twitter logo, featuring the word "twitter" in a white, lowercase, sans-serif font, followed by a white bird icon, all on a light blue rectangular background.The Hotstar logo, featuring the word "hotstar" in a white, lowercase, sans-serif font on a dark blue rectangular background.The MakeMyTrip logo, featuring the word "make" in a dark blue, lowercase, sans-serif font, followed by a red square containing the word "my" in a white, cursive font, and then the word "trip" in a dark blue, lowercase, sans-serif font.





and ...



# Your IITKGP ERP



**Indian Institute of Technology  
Kharagpur**  
*Enterprise Resource Planning*



About SSL Certificate

[Sign In](#) [Sign Up](#) [Forgot Password](#)

Please enter following credentials for signing in. If you dont have any account click on Sign Up. Forgot your password/answer of security question? Click on Forgot Password.

**New admittees:** Please go through [this](#) document. It contains the detailed steps for creating ERP login, signing in and updating profile.

**Sign In**

[Important](#) [Message](#) [Quick Links](#)

**Students**


## Welcome Freshers

- **Registration** [Instruction \(All\)](#) | [Guideline\(UG\)](#) | [FAQ\(UG\)](#)
- **Admission** [ERP Procedure](#) | [UG](#) | [PG \(incl. JAM\)](#) | [RS](#) | [PREP](#)
- **Payment** [Instruction for semester fees payment](#)
- **Interest subvention** [Interest subvention](#) on Education Loan for Students' of IITs
- Central Timetable: [First Year Timetable](#) | [2021-2022 AUTUMN](#)
- **Medical Leave for Students** [Guide](#)
- **Notice** [Branch Change](#) | [Switchover\(Interdisciplinary\)](#) | [Switchover after 3rd year](#) | [Vertical Allocation](#) | [Summer Quarter Failure List](#)
- **PGDBA** [Curriculum](#)

## Directive to all students regarding registration, payment and other technical problems

For registration related problems, mail to [chandan.giri@adm.iitkgp.ac.in](mailto:chandan.giri@adm.iitkgp.ac.in) with a copy to [chairman.erp@adm.iitkgp.ac.in](mailto:chairman.erp@adm.iitkgp.ac.in).

For online payment related problem, [Preview](#) [nupamkh@iitkgp.ac.in](mailto:nupamkh@iitkgp.ac.in) with a copy to



Internet is also  
a Distributed  
System

---



# You have already learned quite a few distributed algos ...

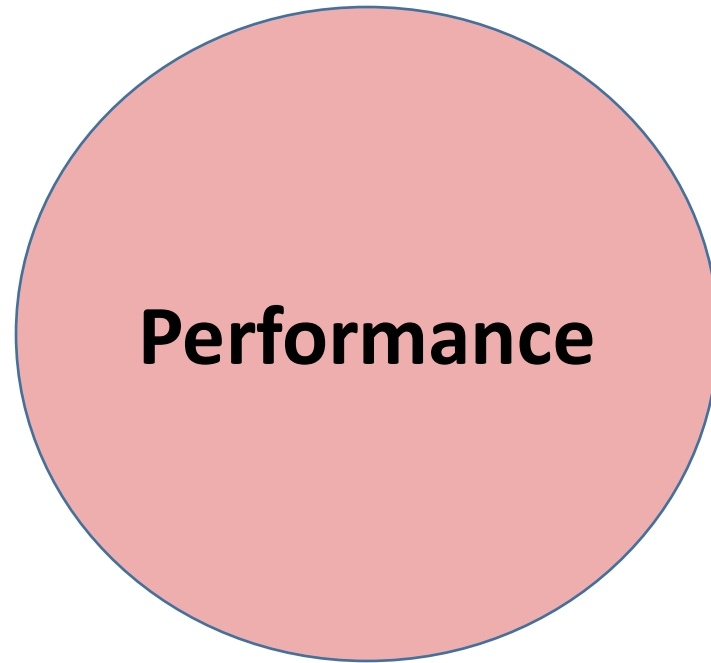
- Internet routing
- TCP congestion control
- Domain Name Systems
- Peer-to-Peer File Transfer (Have you used DC++ or Bittorrent?)
- ...
- The fundamental primitives behind such systems
  - Message passing
  - Shared memory

# What Additional Are You Going to Learn Here?

- It is difficult to satisfy certain properties simultaneously in a distributed system ...

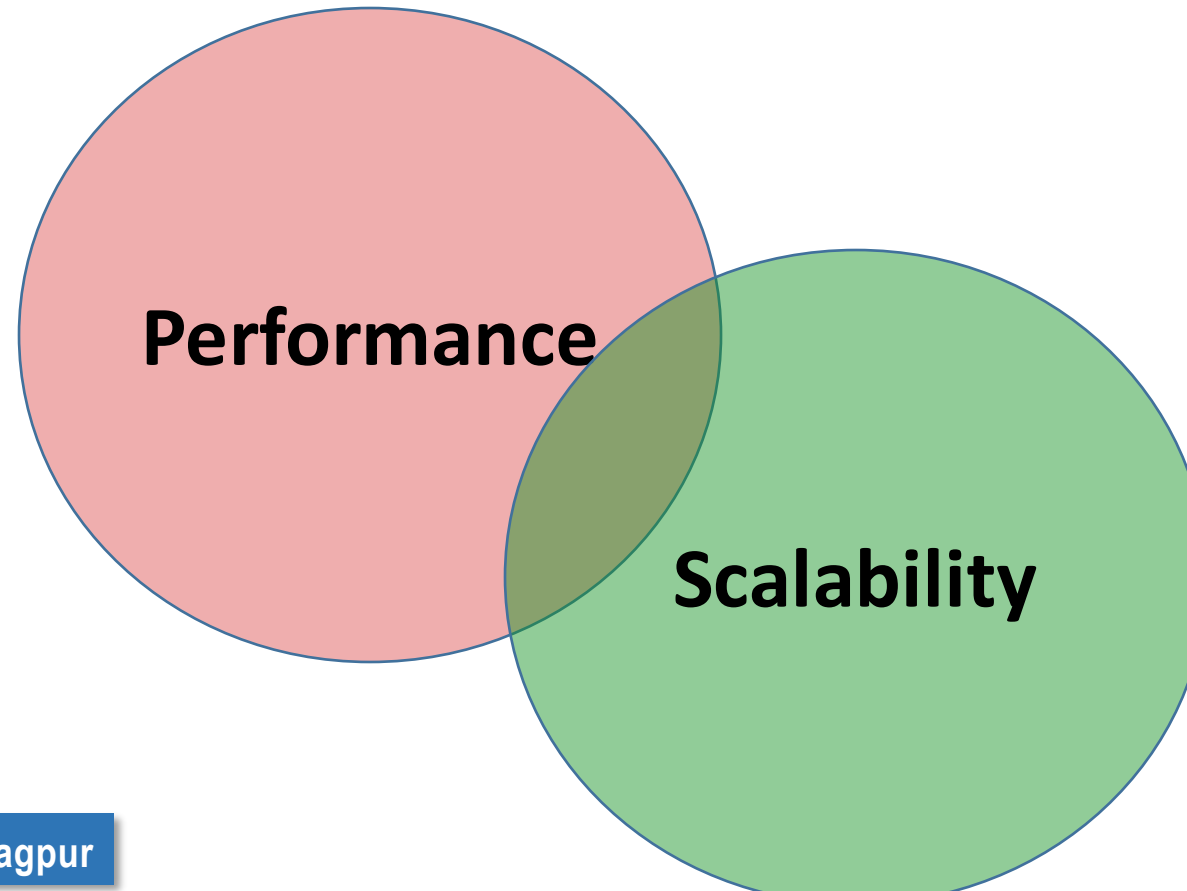
# What Additional Are You Going to Learn Here?

- It is difficult to satisfy certain properties simultaneously in a distributed system ...



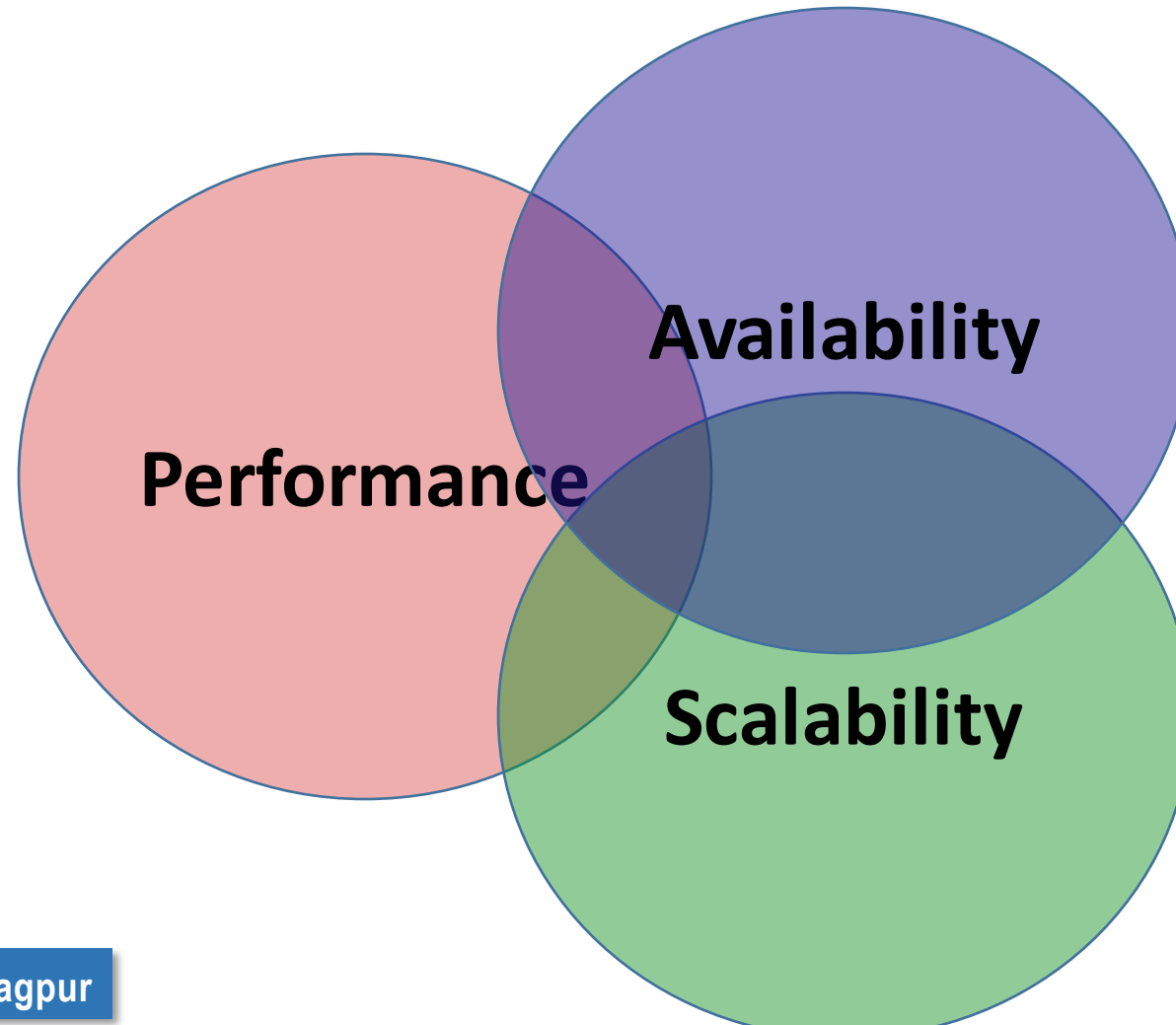
# What Additional Are You Going to Learn Here?

- It is difficult to satisfy certain properties simultaneously in a distributed system ...



# What Additional Are You Going to Learn Here?

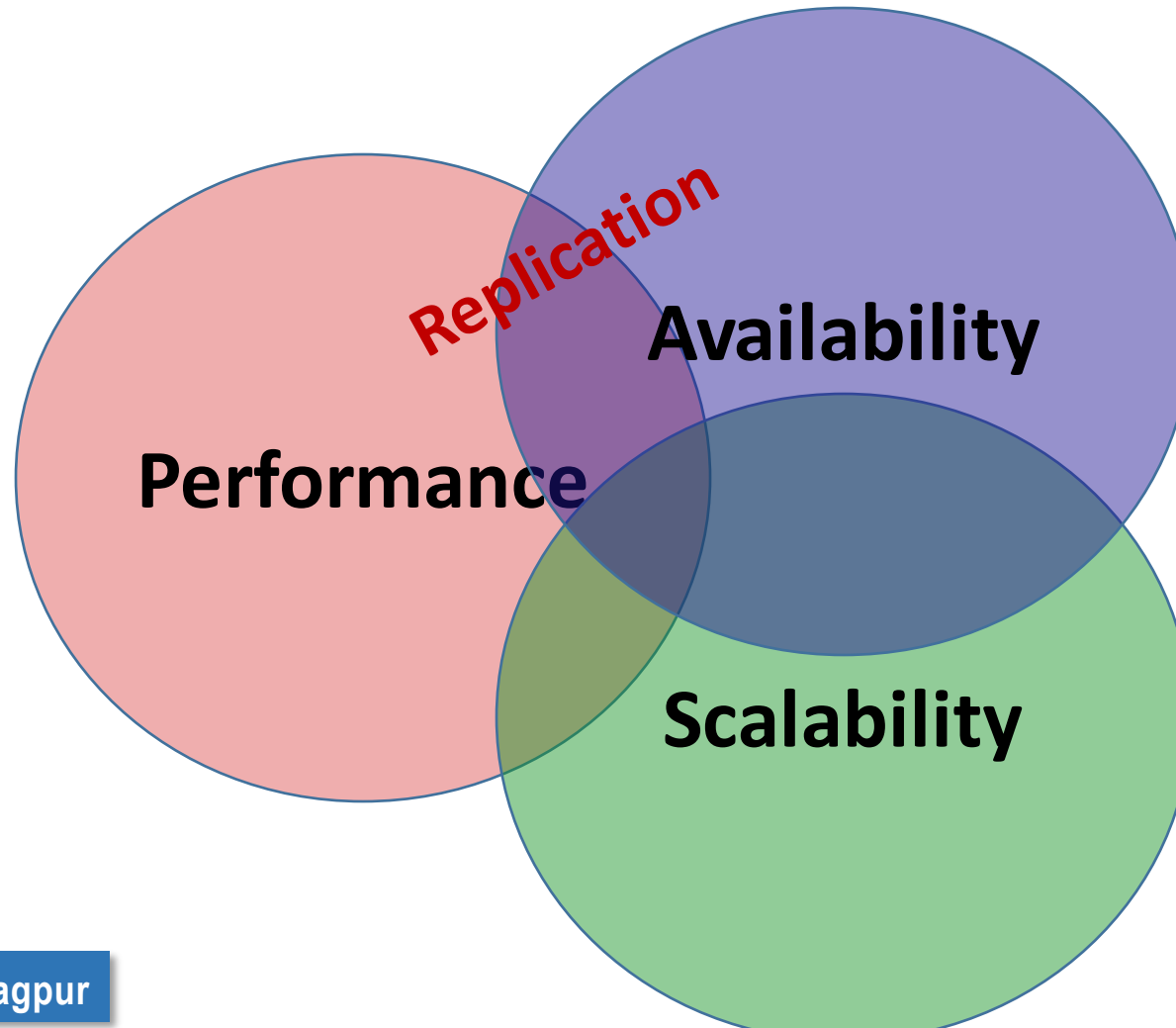
- It is difficult to satisfy certain properties simultaneously in a distributed system ...





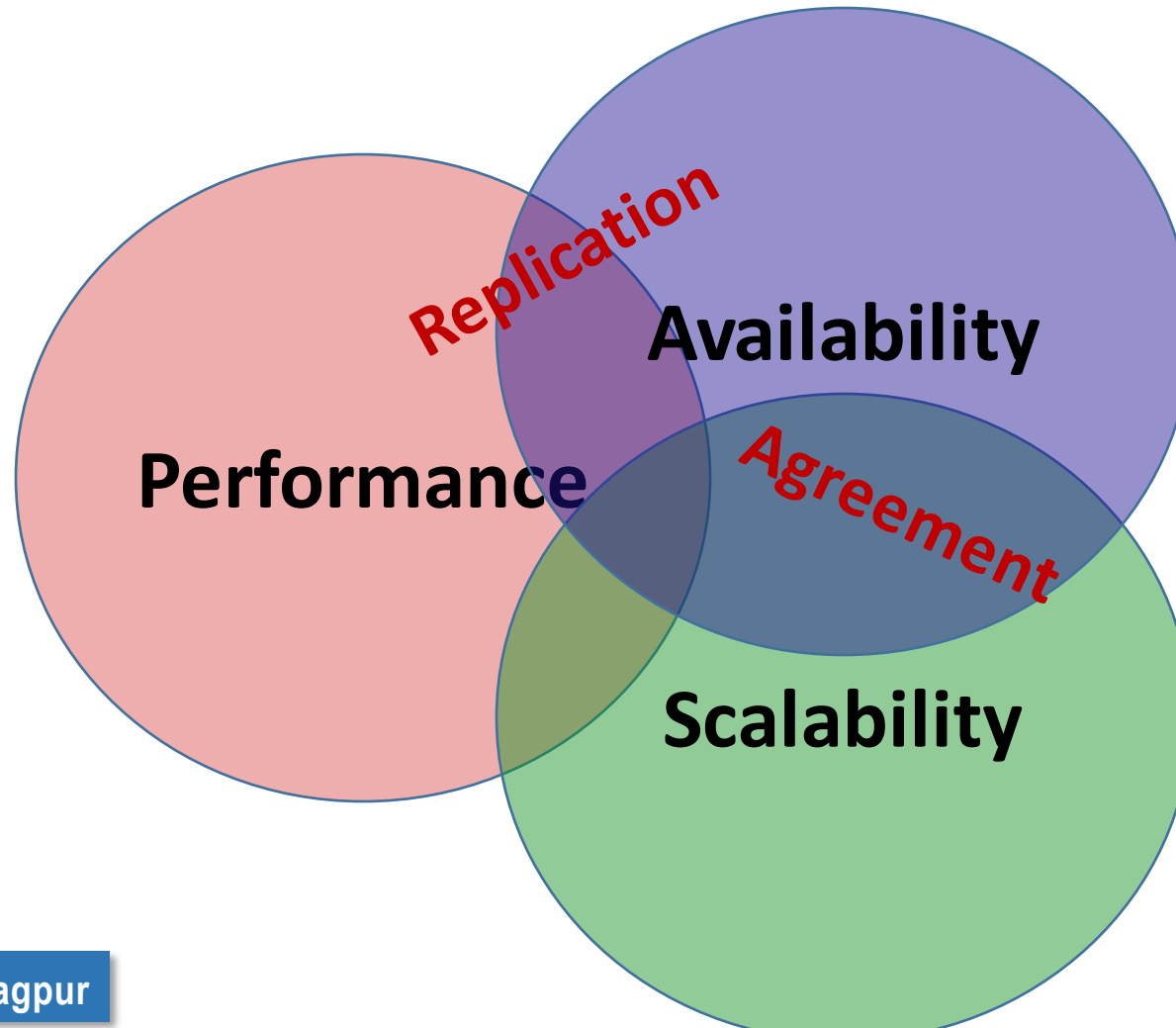
# What Additional Are You Going to Learn Here?

- It is difficult to satisfy certain properties simultaneously in a distributed system ...



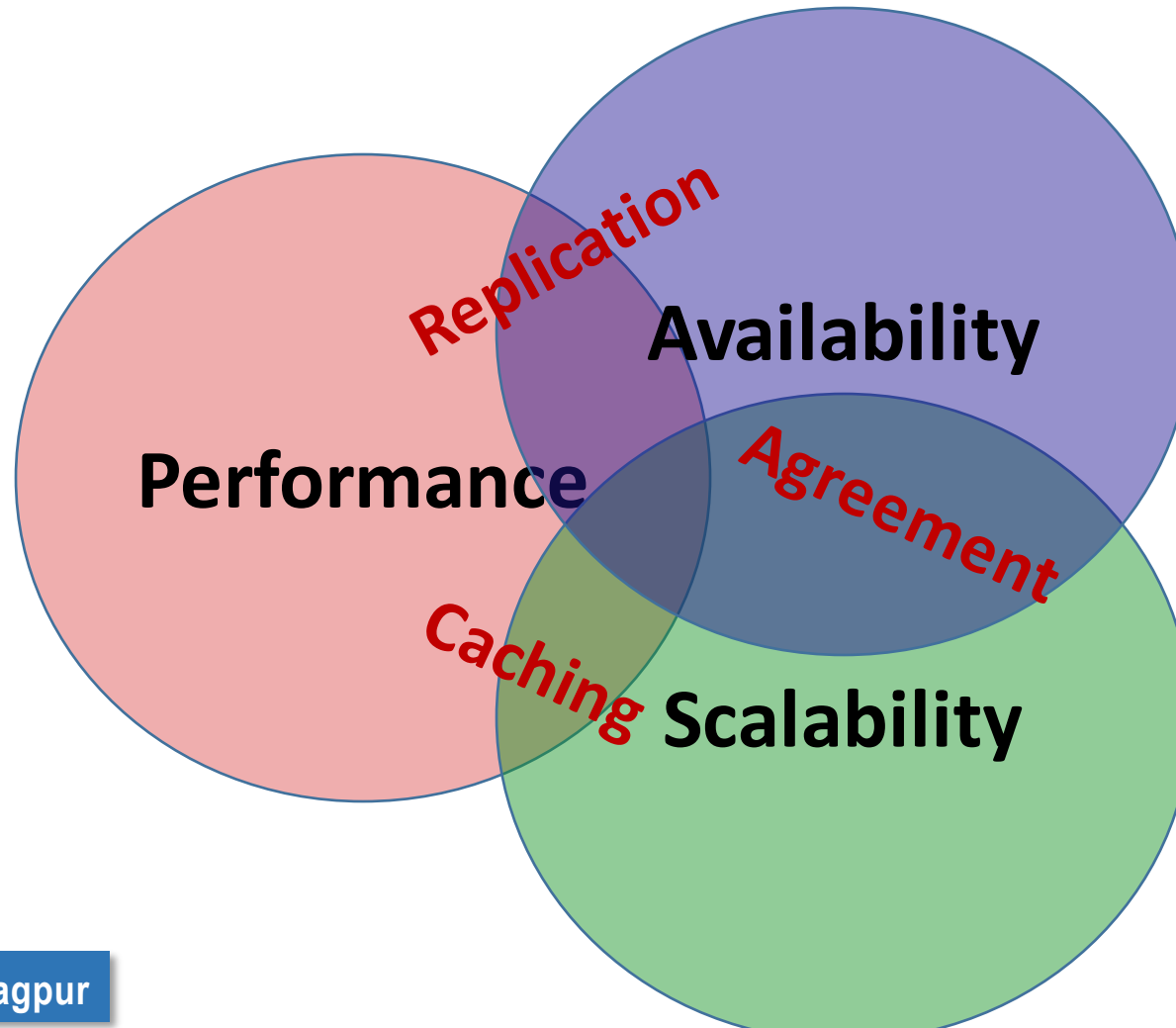
# What Additional Are You Going to Learn Here?

- It is difficult to satisfy certain properties simultaneously in a distributed system ...



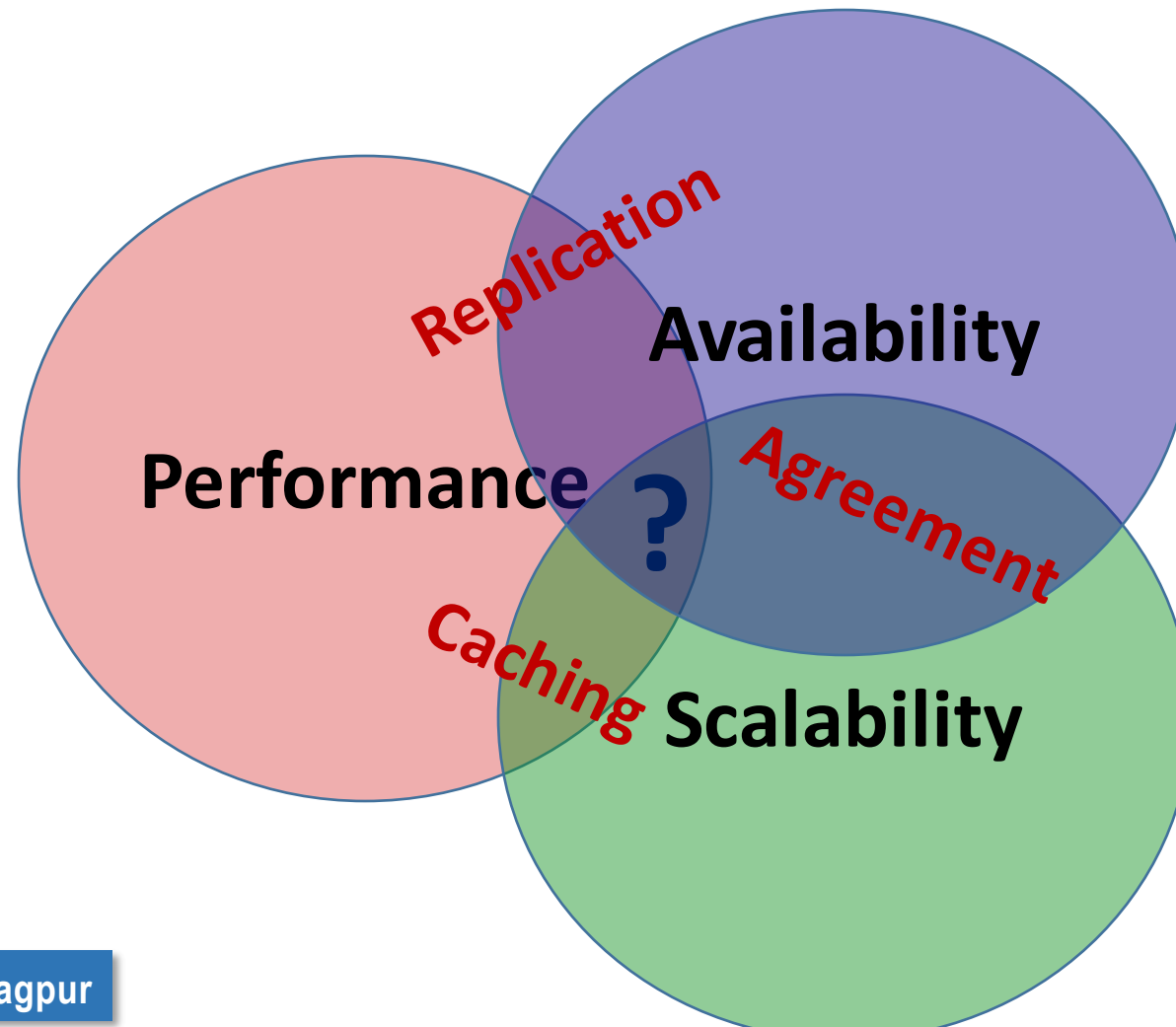
# What Additional Are You Going to Learn Here?

- It is difficult to satisfy certain properties simultaneously in a distributed system ...



# What Additional Are You Going to Learn Here?

- It is difficult to satisfy certain properties simultaneously in a distributed system ...

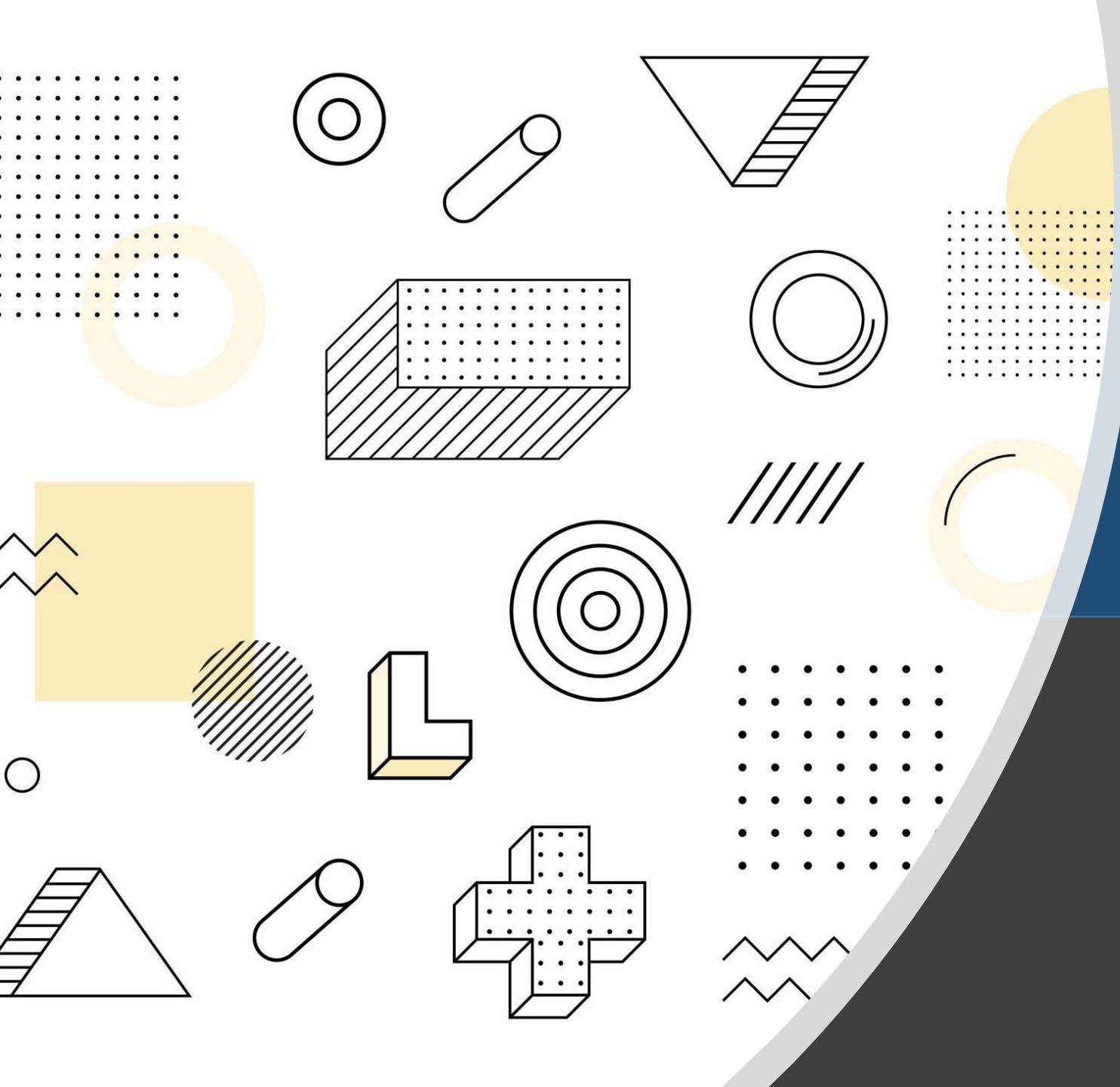


# Books and References

- van Steen and Tanenbaum, **Distributed Systems** (any edition)
  - Free e-book available: <https://www.distributed-systems.net/index.php/books/ds3/>
- Bacon and Harris, **Operating Systems: Concurrent and Distributed Software Design**, Addison-Wesley 2003
- A. D. Kshemkalyani and M. Singhal, **Distributed Algorithms: Principles, Algorithms, and Systems**
- We'll follow various papers and articles, will refer them during discussing different topics

# Some Conferences and Journals to Follow ...

- PODC
- DISC
- ICDCS
- OSDI/SOSP
- ASPLOS
- Usenix ATC
- IEEE Transactions on Parallel and Distributed Systems
- ACM Transactions on Computer Systems



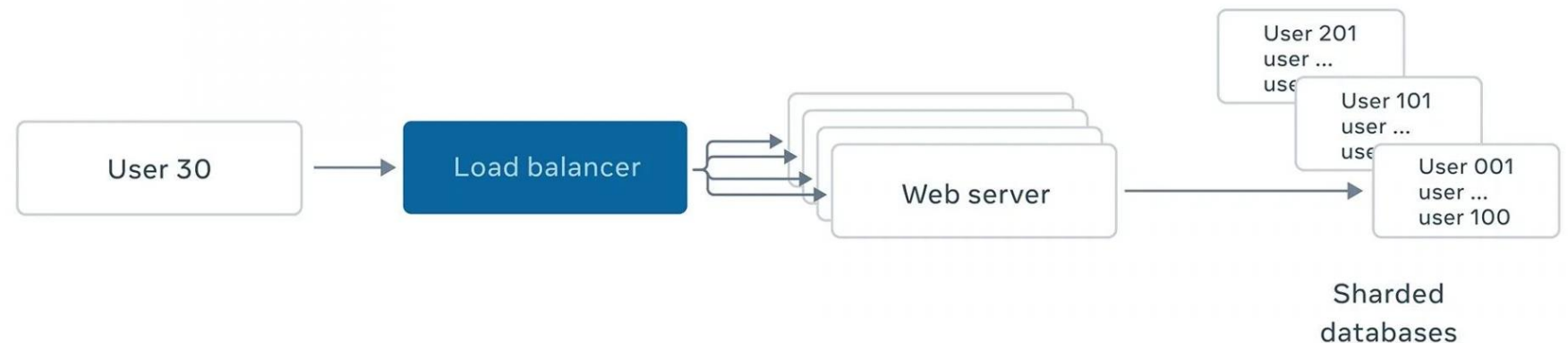
Some running  
systems that you  
might be using ....

But you don't  
know about...



# Facebook Shard Manager

- Data from billions of users, are stored in many databases
- **Sharing:** "a way to scale out services to support high throughput"
- Divide the data into shards and allocate servers for individual shards
  - Spread the load across different databases
  - Failure of shards (hardware or software failure)



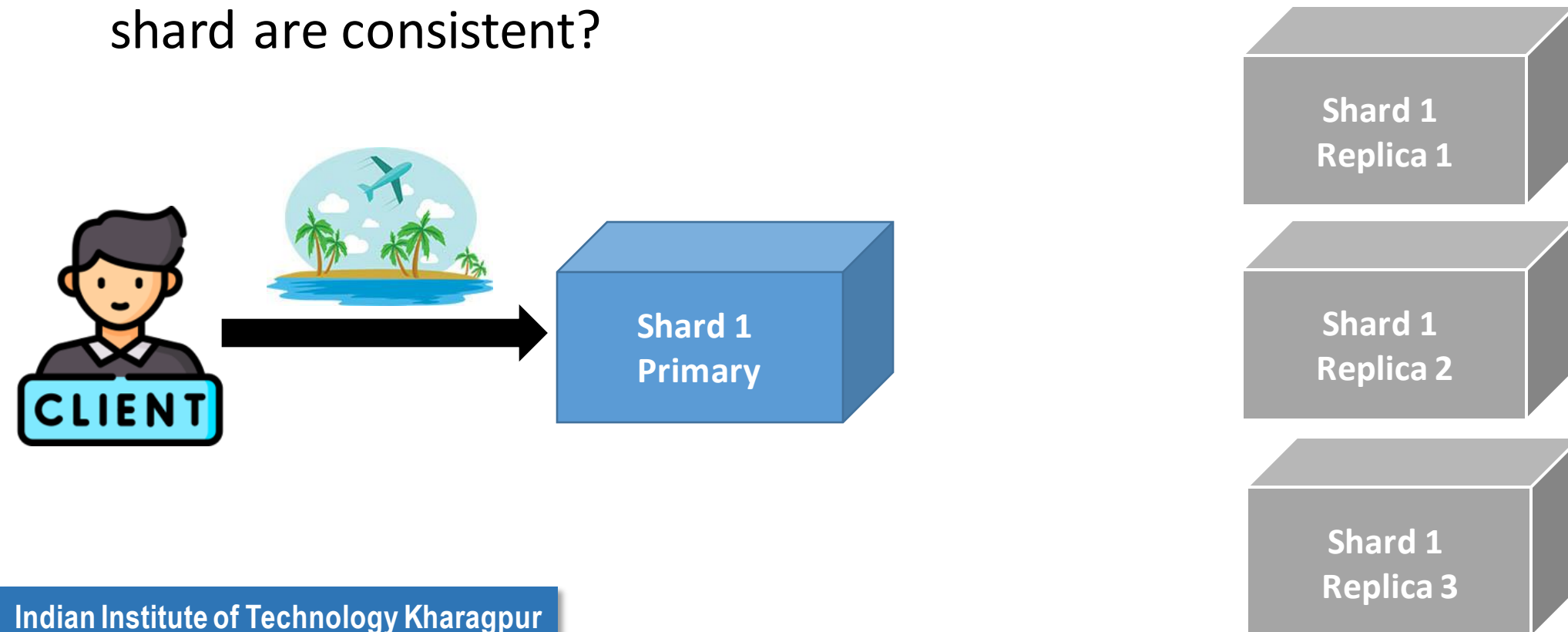
# Facebook Shard Manager

- Maintain multiple replicas for each shard
  - **Why?** Data can be rerouted from another shard when one shard fails
- **Challenge:** During data update, how do you ensure that all the replicas of a shard are consistent?



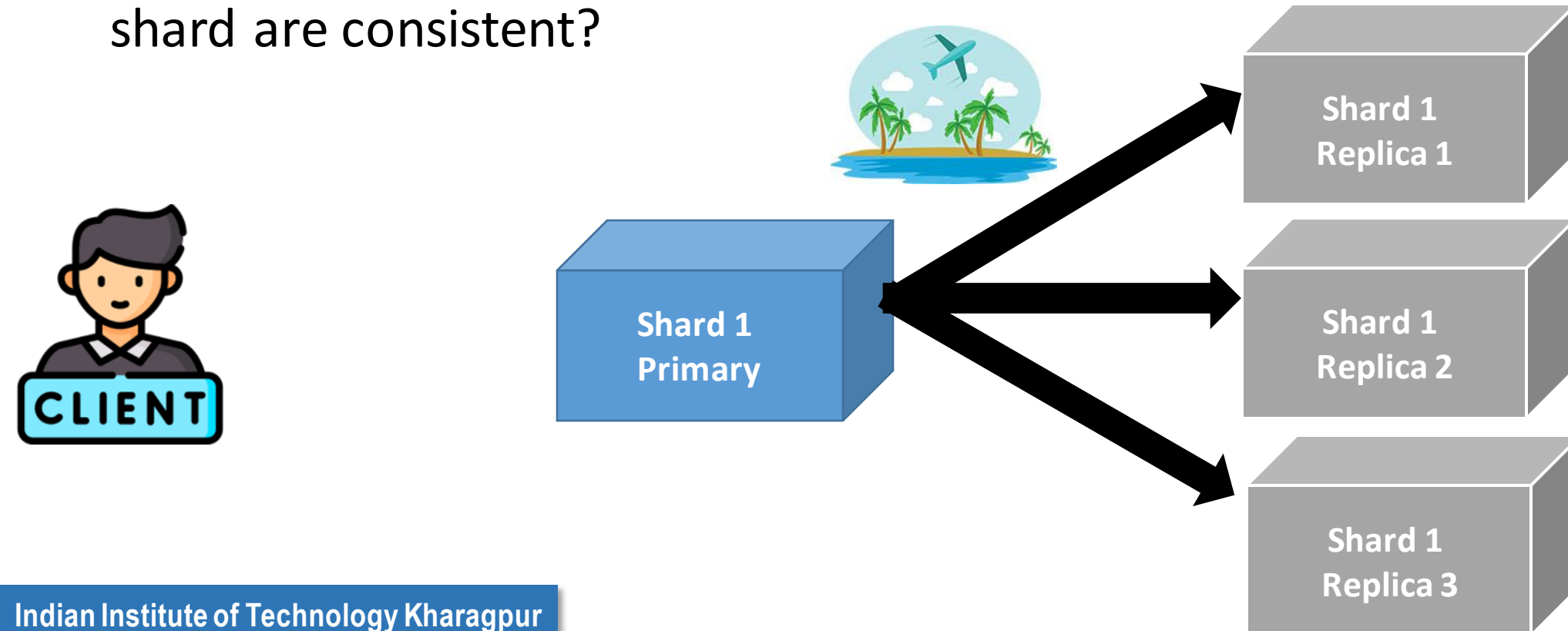
# Facebook Shard Manager

- Maintain multiple replicas for each shard
  - **Why?** Data can be rerouted from another shard when one shard fails
- **Challenge:** During data update, how do you ensure that all the replicas of a shard are consistent?



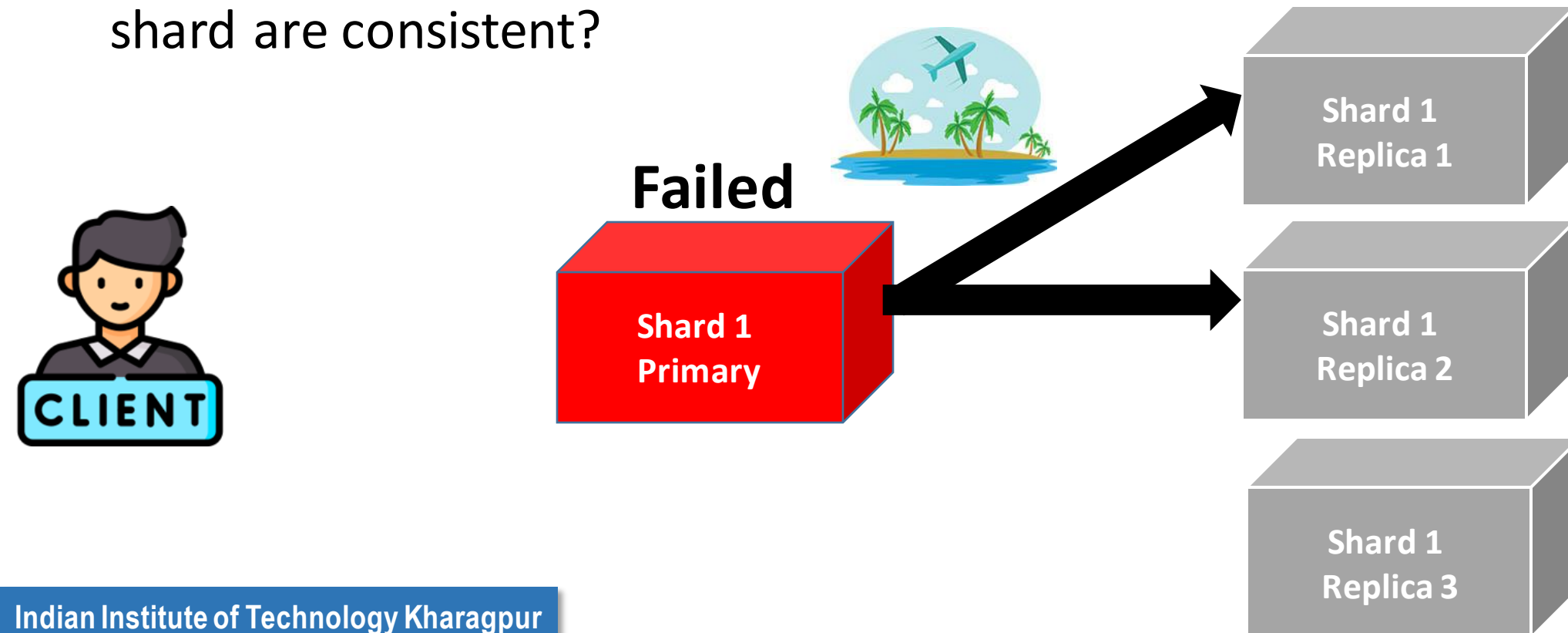
# Facebook Shard Manager

- Maintain multiple replicas for each shard
  - **Why?** Data can be rerouted from another shard when one shard fails
- **Challenge:** During data update, how do you ensure that all the replicas of a shard are consistent?



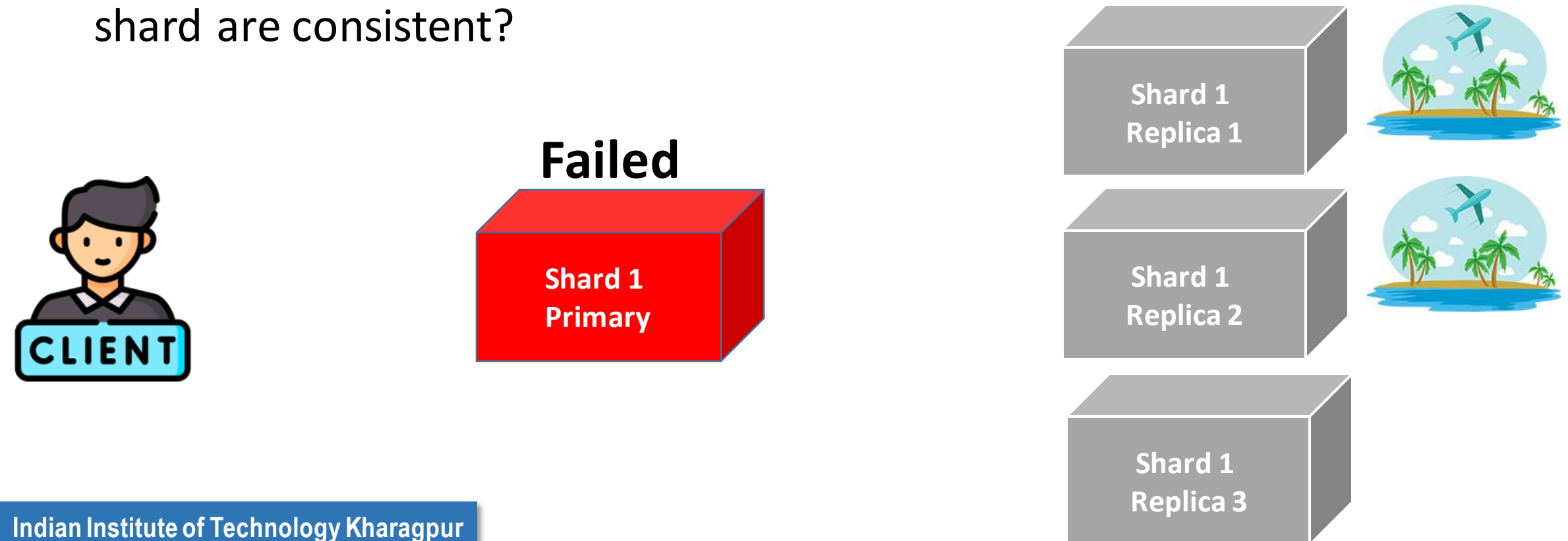
# Facebook Shard Manager

- Maintain multiple replicas for each shard
  - **Why?** Data can be rerouted from another shard when one shard fails
- **Challenge:** During data update, how do you ensure that all the replicas of a shard are consistent?



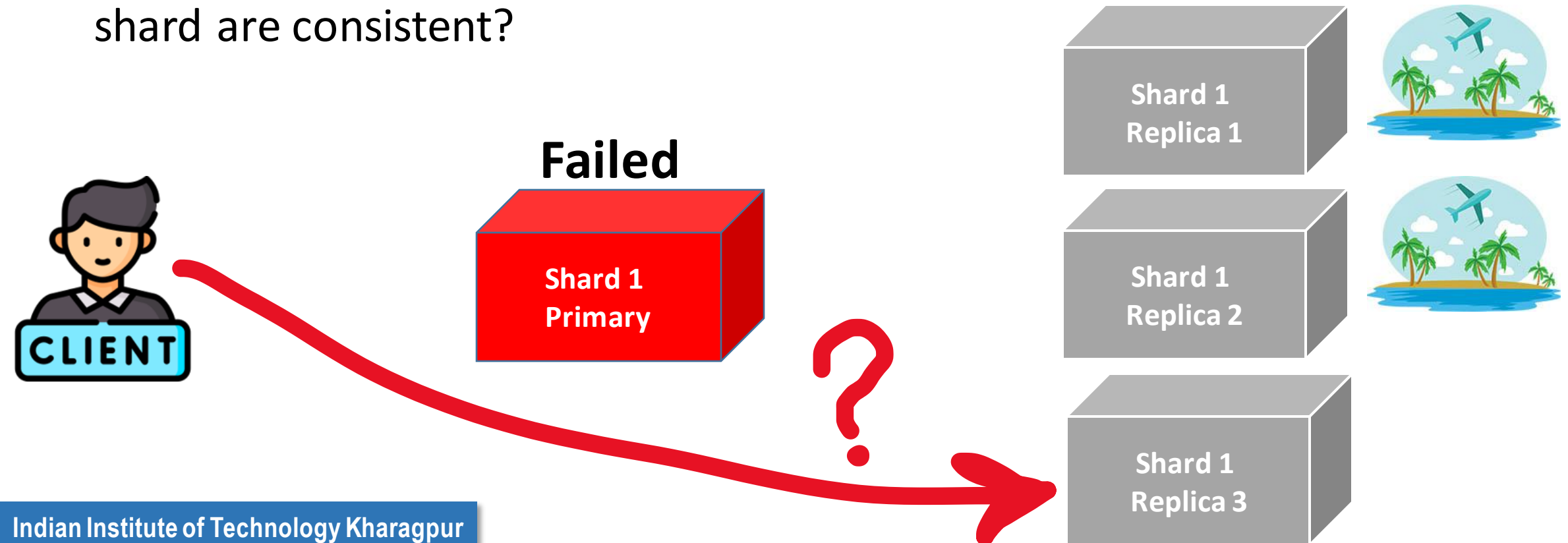
# Facebook Shard Manager

- Maintain multiple replicas for each shard
  - **Why?** Data can be rerouted from another shard when one shard fails
- **Challenge:** During data update, how do you ensure that all the replicas of a shard are consistent?



# Facebook Shard Manager

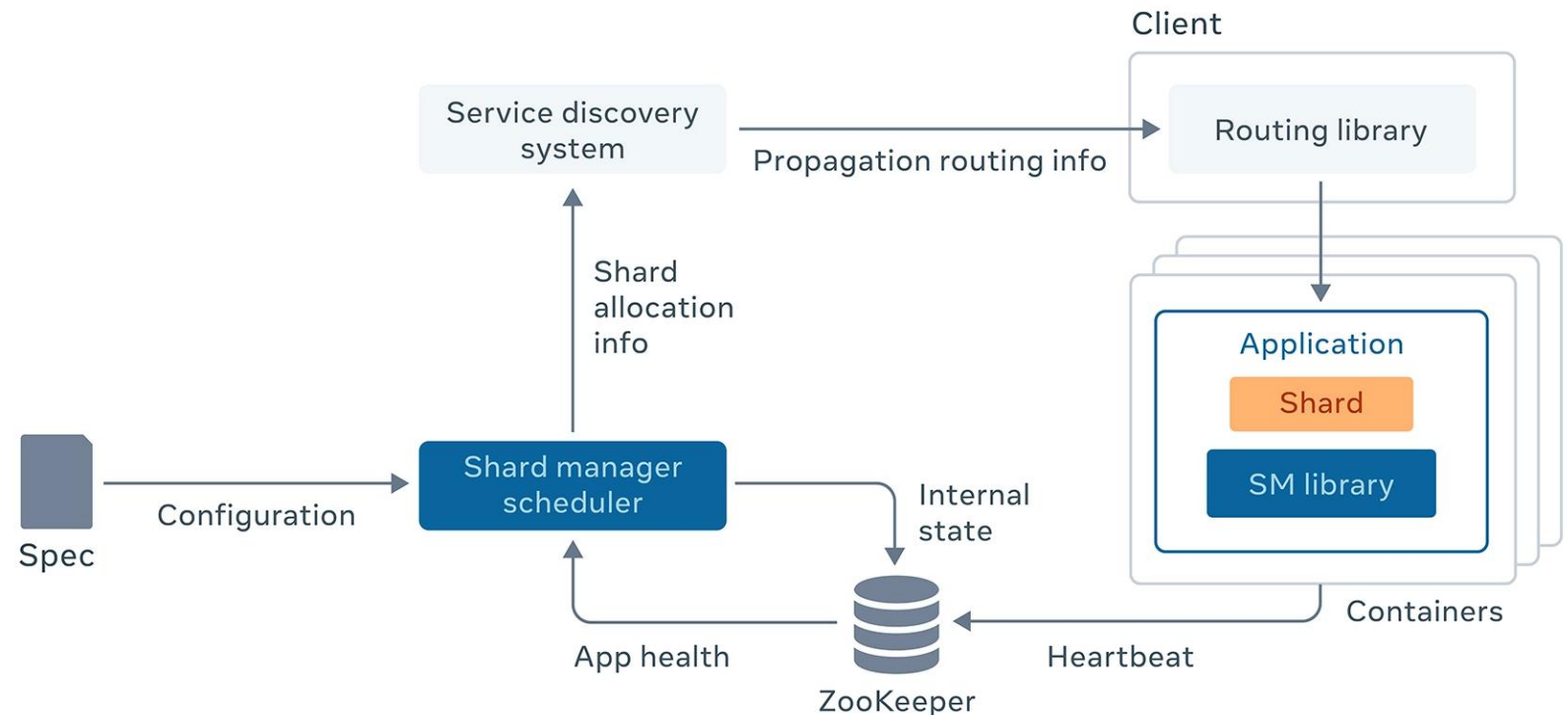
- Maintain multiple replicas for each shard
  - **Why?** Data can be rerouted from another shard when one shard fails
- **Challenge:** During data update, how do you ensure that all the replicas of a shard are consistent?





# Facebook Shard Manager

- The classical problem of **distributed consensus / agreement**
- The shard manager needs to scale up with millions of shards per application
- Further reading: <https://engineering.fb.com/2020/08/24/production-engineering/scaling-services-with-shard-manager/>



# Some Other Distributed Systems from Facebook

- Facebook Ordered Queuing Service (FOQS) -- A distributed priority queue to store and process microservice works and pass them from one microservice to another
  - <https://engineering.fb.com/2021/02/22/production-engineering/foqs-scaling-a-distributed-priority-queue/>
- Async: Distributed asynchronous computing for Facebook applications
  - <https://engineering.fb.com/2020/08/17/production-engineering/async/>
- NTP Service for Facebook
  - <https://engineering.fb.com/2020/03/18/production-engineering/ntp-service/>

# Distributed Computing @ Google

- Pathways: Asynchronous Distributed Data Flow for ML
  - <https://research.google/pubs/pub51473/>
- Debugging incidents in Google's distributed system
  - <https://research.google/pubs/pub49291/>
- Monarch: Google's Planet-Scale In-Memory Time Series Database
  - <https://research.google/pubs/pub50652/>
- Sundial: Fault-tolerant Clock Synchronization for Datacenters
  - <https://research.google/pubs/pub49716/>

## Some Other Resources

- Amazon Builder's library: <https://aws.amazon.com/builders-library/>
- An interesting collection of materials on distributed systems
  - <https://github.com/theanalyst/awesome-distributed-systems>

