# Project ID 36
# Haystack - Dynamic Replication


Haystack - Dynamic Replication

**Submitted By**

**TEAM_45**

**Vilal Al and Shriom Tyagi**

# Strategy Applied to Complete the Project

+ **Defined Objectives:** Established clear goals for implementing dynamic replication in Haystack.
+ **Requirement Analysis:** Identified key metrics like access frequency and storage constraints.
+ **Modular Design:** Adopted a modular architecture for scalability and maintainability.
+ **Dynamic Replication Logic:** Implemented a flexible algorithm to calculate replicas based on file access.
+ **Fault Tolerance:** Integrated mechanisms to improve reliability and system recovery.
+ **Performance Metrics:** Designed metrics to monitor storage, access patterns, and replication impact.
+ **Efficient Storage Utilization:** Balanced storage costs by minimizing unnecessary replicas.
+ **Sample Data & Testing:** Generated realistic data to validate system functionality.
+ **Graphical Representation:** Visualized system performance through graphs for clarity.
+ **Iterative Development:** Followed an iterative process to refine features and resolve issues.
+ **Comprehensive Documentation:** Prepared detailed reports for better understanding and future reference.

# Work Distribution

| S. No. | Task Name | Done By |
|---|---|---|
| 1 | Designed the replication logic and implemented the **calculateReplicas** function. | Vilal Ali |
| 2 | Developed API routes for file upload, metrics, and monitoring replication. | Vilal Ali |
| 3 | Configured the **replicationConfig** and handled system metrics computation. | Vilal Ali |
| 4 | Created **controllers and services** for file management and storage operations. | Shriom Tyagi |
| 5 | Integrated **monitoring** features for replicas and nodes. | Shriom Tyagi |
| 6 | Implemented logging utilities and **error-handling** mechanisms. | Shriom Tyagi |
| 7 | Designed the metrics **Dashboard** and implemented data visualization with graphs. | Vilal Ali |
| 8 | Integrated **API** calls with state management for fetching and displaying data. | Vilal Ali |
| 9 | Developed file **upload and replication** status views with a responsive table. | Shriom Tyagi |
| 10 | Handled user interactions and data binding for seamless functionality. | Shriom Tyagi |
| 11 | Prepared detailed reports for better understanding and future reference. | Vilal Ali |

# Problem Statement

## Replication in Distributed Systems

+ Fixed replication strategies lead to inefficient storage utilization.
+ Popular files may face access bottlenecks.
+ Low-demand files unnecessarily consume resources.

## Key Challenge:

+ How to dynamically optimize replicas based on real-time access patterns?

Where do we start !!!!

# Objectives

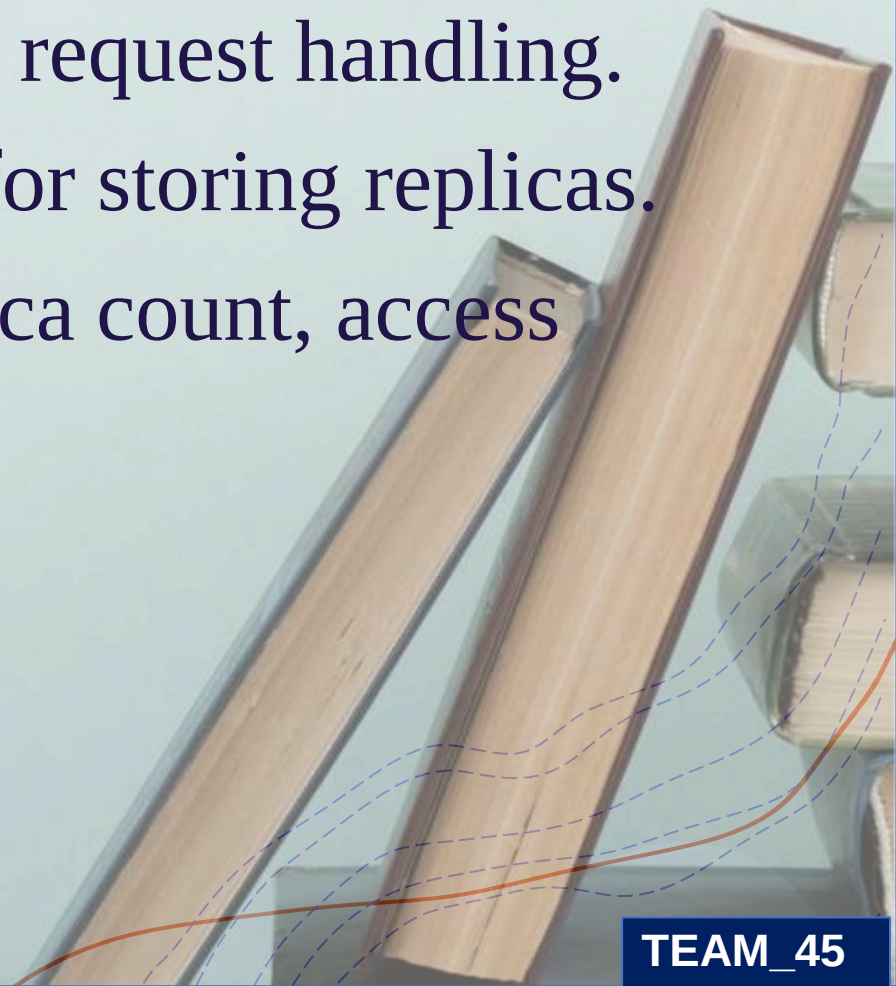
+ Optimize storage utilization by dynamically adjusting replicas.
+ Improve fault tolerance and availability.
+ Scale seamlessly with large datasets and user concurrency.
+ Provide real-time monitoring for users.

# Key Features

- **Dynamic Replication:** Adjust replicas based on access frequency.
- **Fault Tolerance:** Ensure high availability even during node failures.
- **Scalability:** Support growing datasets and concurrent access.
- **User Interface:** Real-time file metadata monitoring and upload.

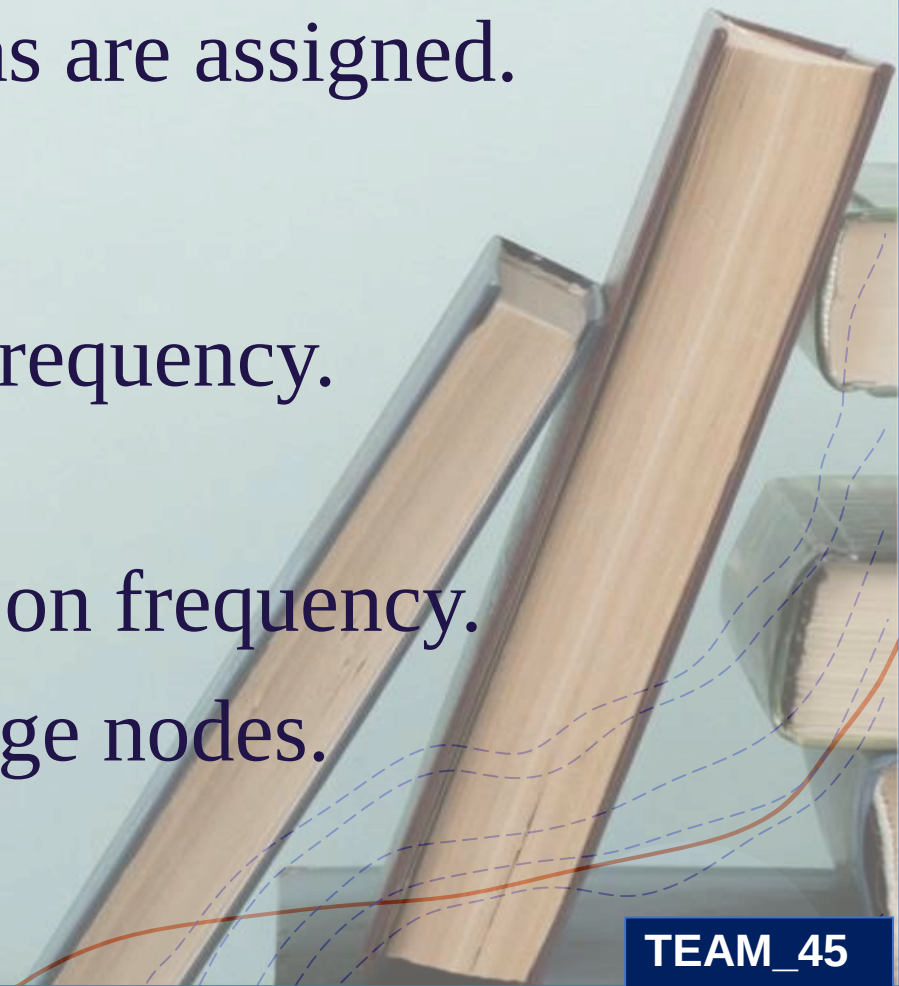
# High-Level Architecture

- **Components:**
  - **Frontend:** React.js and Bootstrap for user interaction.
  - **Backend:** Node.js with Express.js for request handling.
  - **Storage Nodes:** Distributed systems for storing replicas.
  - **Metadata Manager:** Tracks file replica count, access frequency, and locations.



# Workflow

- **File Upload:**
  - Users upload files through the interface.
  - Metadata is created, and initial replicas are assigned.
- **File Access:**
  - Each access updates the file's access frequency.
- **Dynamic Adjustment**:
  - Replica counts are recalculated based on frequency.
  - Replicas are redistributed across storage nodes.

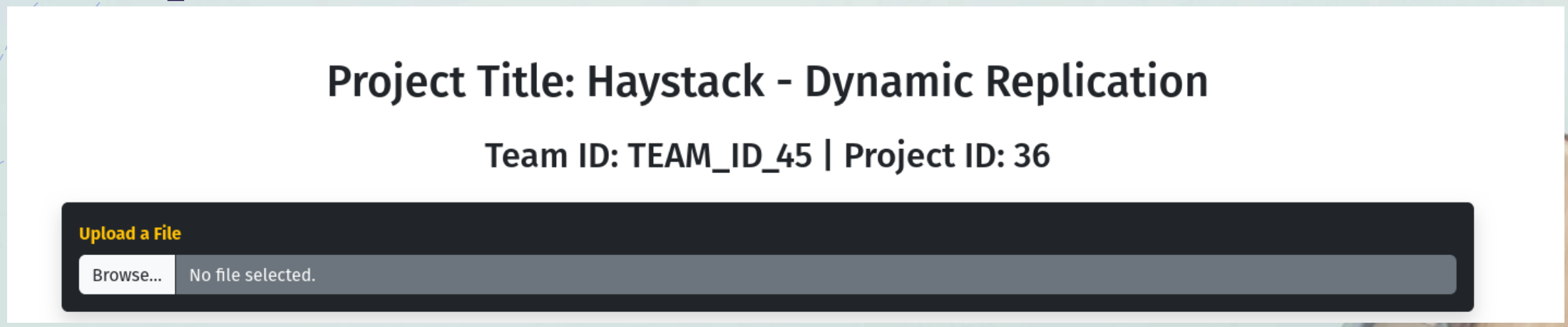


# Implementation Details

- **Frontend:**
  - File upload with Browse File Dialog
  - Metadata visualization.
- **Backend:**
  - Replica adjustment logic based on:
    - Access frequency thresholds.
    - Configurable parameters (replicationconfig.js).
  - APIs for file management.

# Output Screenshots

## File Upload Interface



# Output Screenshots

## Metadata Viewer

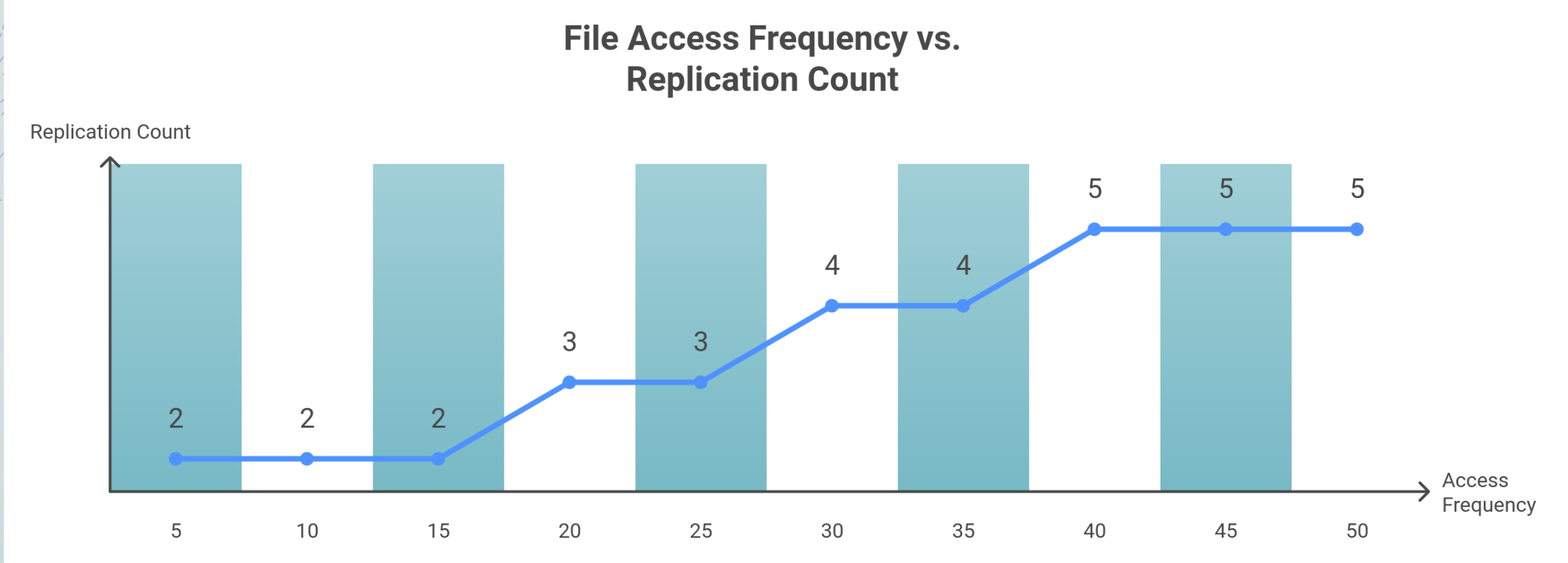
| File Name | Size | Access Frequency | Min Replicas | Max Replicas | Number of Replicas | Replicas | Monitored Nodes |
|---|---|---|---|---|---|---|---|
| 1732816193469.mp4 | 8.17 MB | 5 | 2 | 5 | 2 | replica_1/1732816193469.mp4<br>replica_2/1732816193469.mp4 | Node (Active) - IP: 10.1.42.130 |
| 1732816593124.mp4 | 8.17 MB | 10 | 2 | 5 | 2 | replica_1/1732816593124.mp4<br>replica_2/1732816593124.mp4 | Node (Active) - IP: 10.1.42.130 |
| 1732816613061.mp4 | 8.17 MB | 15 | 2 | 5 | 2 | replica_1/1732816613061.mp4<br>replica_2/1732816613061.mp4 | Node (Active) - IP: 10.1.42.130 |
| 1732816662919.mp4 | 8.17 MB | 20 | 2 | 5 | 3 | replica_1/1732816662919.mp4<br>replica_2/1732816662919.mp4<br>replica_3/1732816662919.mp4 | Node (Active) - IP: 10.1.42.130 |
| 1732816694658.mp4 | 8.17 MB | 25 | 2 | 5 | 3 | replica_1/1732816694658.mp4<br>replica_2/1732816694658.mp4<br>replica_3/1732816694658.mp4 | Node (Active) - IP: 10.1.42.130 |
| 1732816727347.mp4 | 8.17 MB | 30 | 2 | 5 | 4 | replica_1/1732816727347.mp4<br>replica_2/1732816727347.mp4<br>replica_3/1732816727347.mp4<br>replica_4/1732816727347.mp4 | Node (Active) - IP: 10.1.42.130 |
| 1732816774681.mp4 | 8.17 MB | 35 | 2 | 5 | 4 | replica_1/1732816774681.mp4<br>replica_2/1732816774681.mp4<br>replica_3/1732816774681.mp4<br>replica_4/1732816774681.mp4 | Node (Active) - IP: 10.1.42.130 |
| 1732816809102.mp4 | 8.17 MB | 40 | 2 | 5 | 5 | replica_1/1732816809102.mp4<br>replica_2/1732816809102.mp4<br>replica_3/1732816809102.mp4<br>replica_4/1732816809102.mp4<br>replica_5/1732816809102.mp4 | Node (Active) - IP: 10.1.42.130 |
| 1732816837929.mp4 | 8.17 MB | 45 | 2 | 5 | 5 | replica_1/1732816837929.mp4<br>replica_2/1732816837929.mp4<br>replica_3/1732816837929.mp4<br>replica_4/1732816837929.mp4<br>replica_5/1732816837929.mp4 | Node (Active) - IP: 10.1.42.130 |
| 1732816862486.mp4 | 8.17 MB | 50 | 2 | 5 | 5 | replica_1/1732816862486.mp4<br>replica_2/1732816862486.mp4<br>replica_3/1732816862486.mp4<br>replica_4/1732816862486.mp4<br>replica_5/1732816862486.mp4 | Node (Active) - IP: 10.1.42.130 |

# Output Screenshots

## +Backend Console

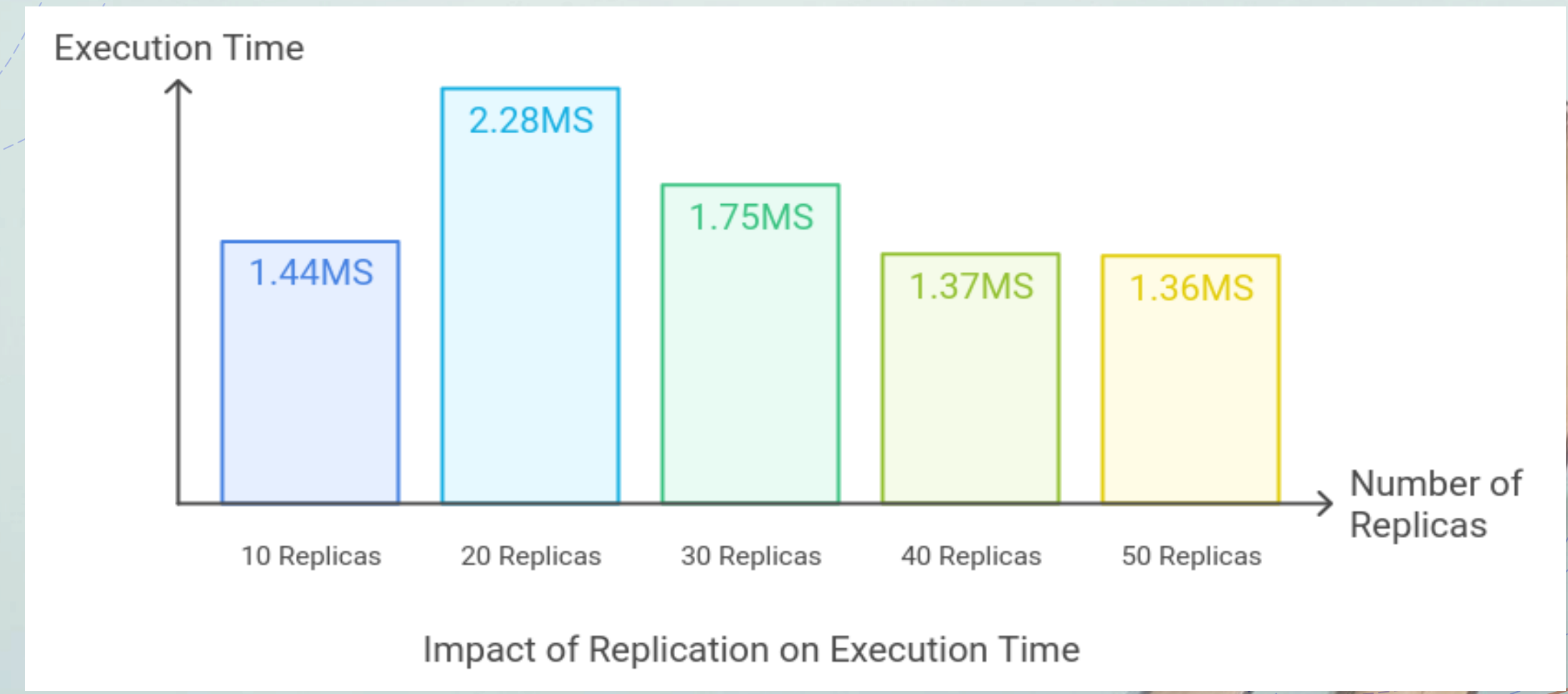
```
❯ node src/server.js
Server running on http://localhost:3001
File upload request received
File successfully uploaded to /home/vilal/MyWork/IIIT-Study/Sem-1_24/CS3.401_Distributed_Systems/Project/36-Project-Team-ID-45/server/uploads/1732818172659.mp4
Creating 10 replicas for file: 1732818172659.mp4
Replica 1 created at /home/vilal/MyWork/IIIT-Study/Sem-1_24/CS3.401_Distributed_Systems/Project/36-Project-Team-ID-45/server/replicas/replica_1/1732818172659.mp4
Replica 2 created at /home/vilal/MyWork/IIIT-Study/Sem-1_24/CS3.401_Distributed_Systems/Project/36-Project-Team-ID-45/server/replicas/replica_2/1732818172659.mp4
Replica 3 created at /home/vilal/MyWork/IIIT-Study/Sem-1_24/CS3.401_Distributed_Systems/Project/36-Project-Team-ID-45/server/replicas/replica_3/1732818172659.mp4
Replica 4 created at /home/vilal/MyWork/IIIT-Study/Sem-1_24/CS3.401_Distributed_Systems/Project/36-Project-Team-ID-45/server/replicas/replica_4/1732818172659.mp4
Replica 5 created at /home/vilal/MyWork/IIIT-Study/Sem-1_24/CS3.401_Distributed_Systems/Project/36-Project-Team-ID-45/server/replicas/replica_5/1732818172659.mp4
Replica 6 created at /home/vilal/MyWork/IIIT-Study/Sem-1_24/CS3.401_Distributed_Systems/Project/36-Project-Team-ID-45/server/replicas/replica_6/1732818172659.mp4
Replica 7 created at /home/vilal/MyWork/IIIT-Study/Sem-1_24/CS3.401_Distributed_Systems/Project/36-Project-Team-ID-45/server/replicas/replica_7/1732818172659.mp4
Replica 8 created at /home/vilal/MyWork/IIIT-Study/Sem-1_24/CS3.401_Distributed_Systems/Project/36-Project-Team-ID-45/server/replicas/replica_8/1732818172659.mp4
Replica 9 created at /home/vilal/MyWork/IIIT-Study/Sem-1_24/CS3.401_Distributed_Systems/Project/36-Project-Team-ID-45/server/replicas/replica_9/1732818172659.mp4
Replica 10 created at /home/vilal/MyWork/IIIT-Study/Sem-1_24/CS3.401_Distributed_Systems/Project/36-Project-Team-ID-45/server/replicas/replica_10/1732818172659.mp4
System Metrics:
---------------
Total Files: 7
Total Replicas: 10
Storage Used: 57.21 MB
Available Storage: 77919.21 MB
CPU Usage: 3%
Execution Time: 1.44 ms
---------------
```

# File Access Frequency vs. Replication Count:



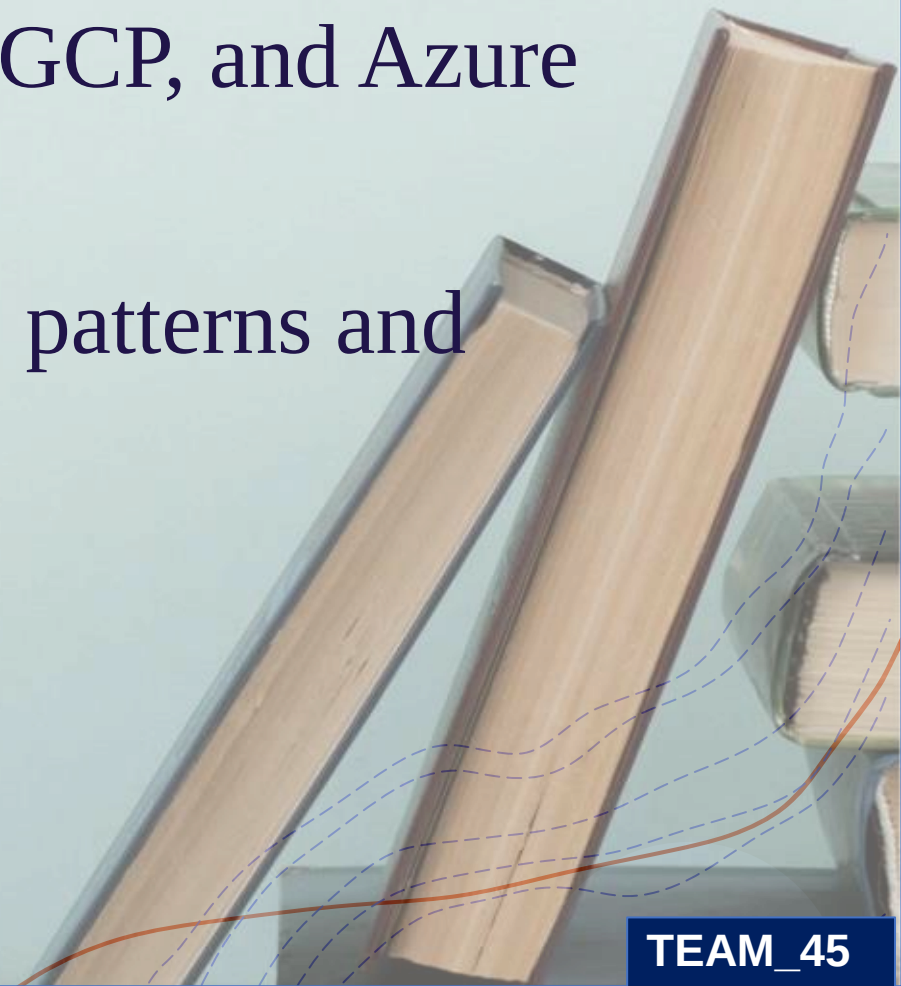
# Comparative Analysis

## + Replication Impact on Execution Time:



# Future Scope

- **gRPC Transition:** Replace REST APIs with gRPC for real-time communication.
- **Cloud Integration:** Support for AWS, GCP, and Azure storage systems.
- **AI-Driven Replication:** Predict access patterns and optimize replicas proactively.



# Conclusion

- **Conclusion:**
  - Haystack - Dynamic Replication optimizes resource utilization while ensuring fault tolerance and scalability.
  - It addresses the inefficiencies of traditional replication strategies.

# Q&A

???

...

TEAM_45

# thank you!