Innovate 2025

Team Number: T-2496
Team Name: SoloDev

Title	Decentralized File Storage
Problem Theme	Open Innovation
Institute Name	Thapar Institute of Engineering and Technology, Patiala
Team Member(s)	Saksham Mittal
Contact	smittal1_be23@thapar.edu

Background

The demand for distributed storage solutions has surged with the rise of:

- **Web3 & Decentralized Applications (DApps)** – Blockchain projects need reliable, decentralized data storage.
- **Privacy-Conscious Users** – Growing concerns over surveillance and corporate data collection drive interest in decentralized alternatives.
- **Edge Computing & IoT** – The explosion of IoT devices creates vast amounts of data that need distributed, low-latency storage.
- **Resilient Infrastructure for Critical Applications** – Industries like healthcare, finance, and defense need robust data storage that remains functional despite server failures or cyberattacks.

Existing distributed storage solutions like IPFS (Inter Planetary File System) and Filecoin attempt to solve these problems but have significant limitations, including slow retrieval speeds, lack of efficient tracking mechanisms, and complex economic models.

Problem Statement

Data storage has become a fundamental challenge in the digital era. Centralized cloud storage solutions (AWS S3, Google Cloud Storage, Dropbox, etc.) dominate the market, but they come with significant limitations:

- **Single Points of Failure** – If a central cloud provider experiences downtime (e.g., AWS outages), all hosted services can be disrupted.
- **Data Privacy & Security Risks** – Centralized storage providers have control over user data, creating risks of unauthorized access, surveillance, or data breaches.
- **High Costs** – Storing data on cloud providers can become expensive, with costs increasing based on bandwidth and storage needs.
- **Lack of Decentralization** – Users are dependent on a handful of cloud giants, reducing resilience and increasing monopolization.

To address these challenges, we propose a fully decentralized, peer-to-peer (P2P) storage system with self-organizing, leaderless coordination for managing storage nodes efficiently.

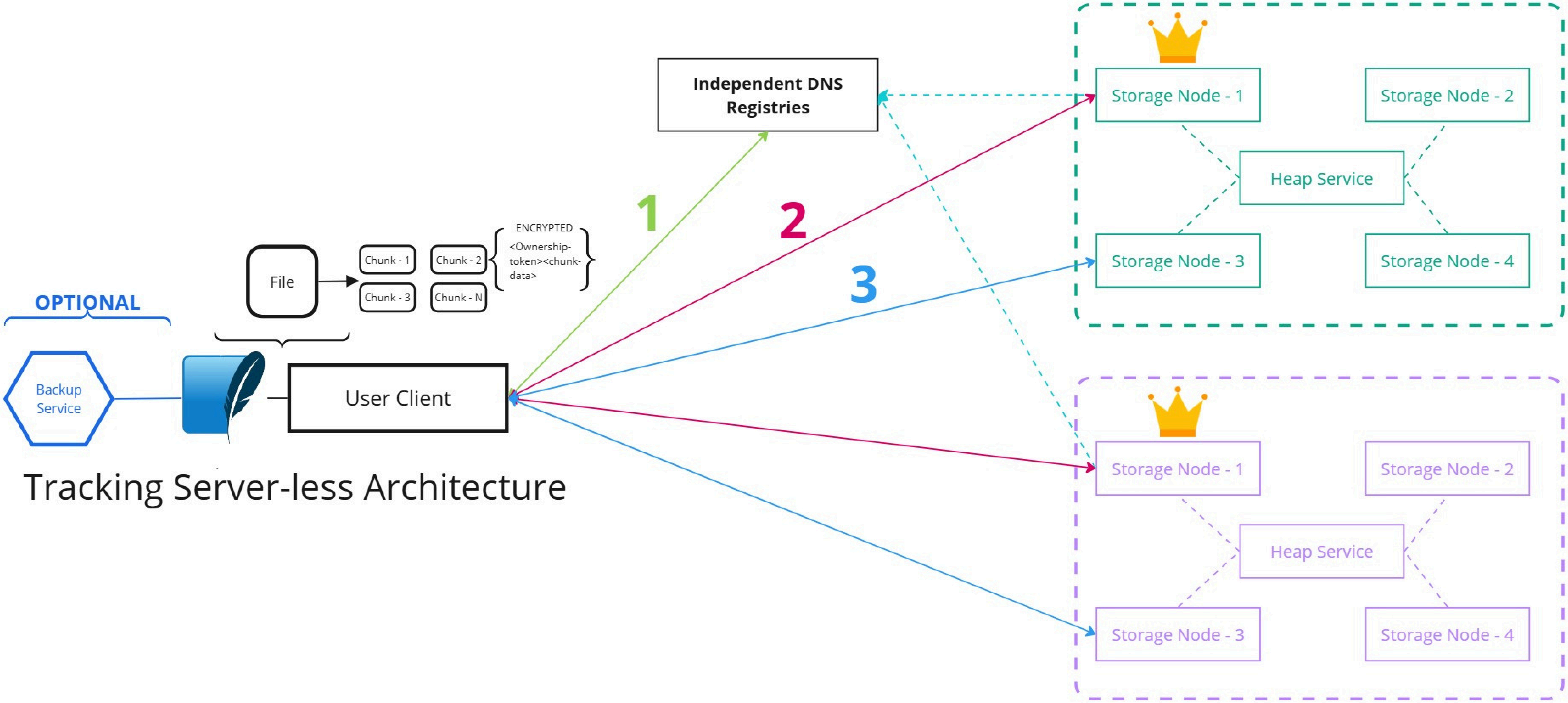
Objectives & Expected Outcomes

The primary goal of this project is to build a framework for a decentralized file storage system which allows for:

- **User control on where their data is stored:** The user should have the power to decide where and with whom will their data be stored with, making this decision easy for the user.
- **Robust data storage:** The data should be stored in such a way that only the user should be able to reconstruct and use it. Not even the storage providers should be able to use it.
- **Leak-proof data storage model:** Most data storage solutions fail when an insider sabotages it. The project aims to build a data storage model such that even in case of leaks, the actual thing leaked is unusable (even in scenarios where encryption has somehow been broken).
- **Eliminate vendor dependence & create an open market for data storage:** Eliminate monopolies for data storage needs and create an open market for the same.

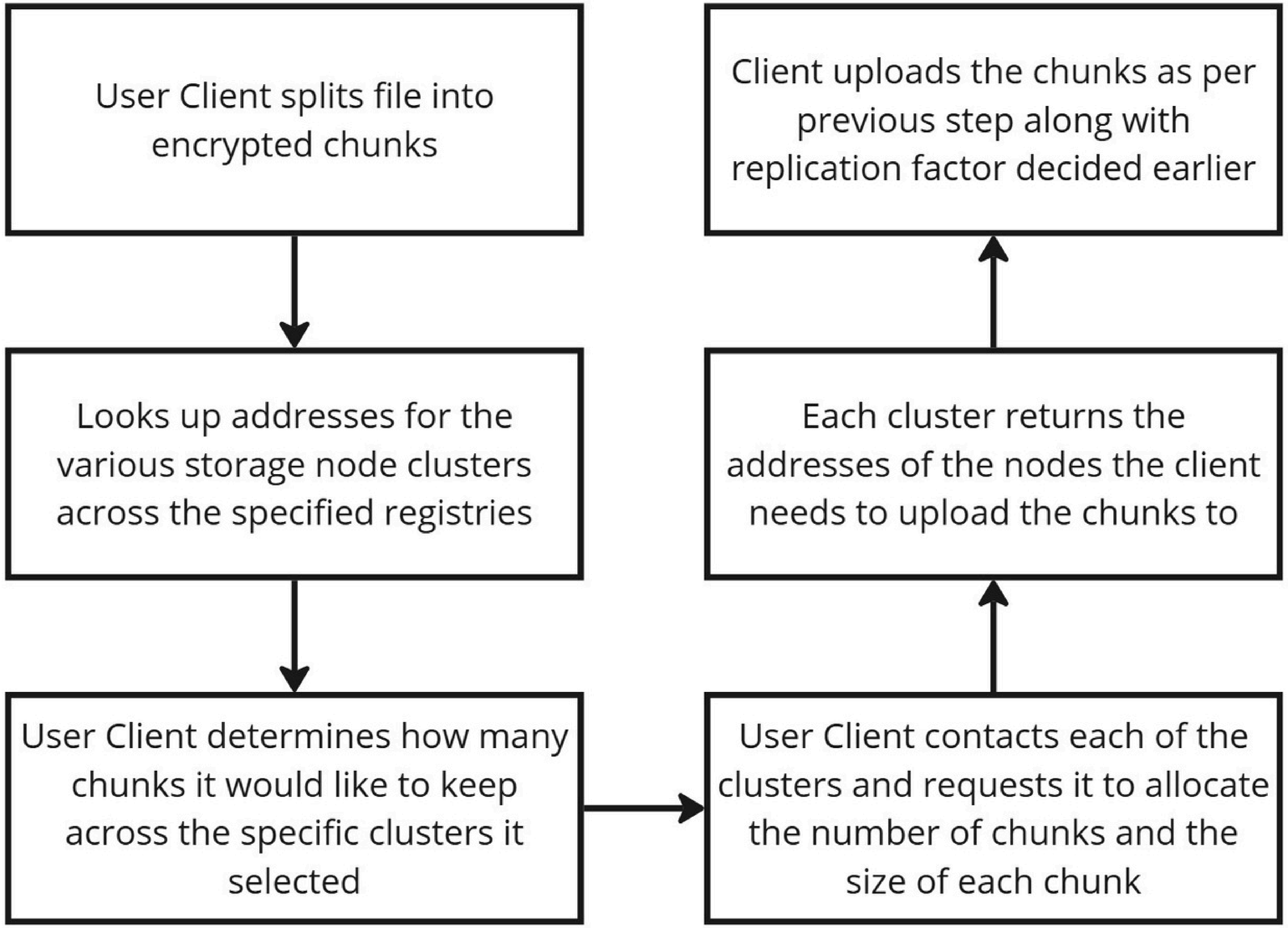
This project is under active development:
https://github.com/B4S1C-Coder/Decentralized-File-Storage

Methodology



Tracking Server-less Architecture

Methodology



User Client splits file into encrypted chunks

Looks up addresses for the various storage node clusters across the specified registries

User Client determines how many chunks it would like to keep across the specific clusters it selected

User Client contacts each of the clusters and requests it to allocate the number of chunks and the size of each chunk

Each cluster returns the addresses of the nodes the client needs to upload the chunks to

Client uploads the chunks as per previous step along with replication factor decided earlier

Proposed Solution

The proposed data storage solution is a self-organizing, decentralized distributed storage network with a strong focus on user control and has the following key features:

(a) Leaderless Coordination with Efficient Resource Allocation

Instead of a centralized tracking server, each storage node can function as a mini-tracker, electing a leader dynamically using a chain-of-command system rather than traditional leader election.

- The highest-ranked active node in the chain becomes the leader.
- If a leader fails, the next in line automatically takes over.
- Heap-based storage management allows efficient allocation of storage resources.

(b) DNS-Based Address Resolution for Seamless Access

Instead of a static discovery mechanism, a DNS registry dynamically tracks leader nodes.

- Clients query DNS to resolve the leader node’s address and request storage allocation.
- If a leader fails, DNS automatically updates with the new leader’s address.

Proposed Solution

(c) Storage Nodes with Self-Reporting Mechanisms

Storage nodes update their status in a distributed Fibonacci Heap structure maintained by a single, lightweight, in-memory heap service.

- Nodes periodically report available storage, online status, and health metrics.
- If a node goes offline, the system removes it from allocation pools without requiring centralized intervention.

(d) Data Sharding & Redundancy for Fault Tolerance

- Each file is split into encrypted chunks and distributed across multiple nodes.
- Redundancy is ensured using erasure coding or replication.
- Nodes can reconstruct lost data from redundant chunks.

(e) Efficient Load Balancing & Scalability

- No bottlenecks since storage nodes handle their own coordination.
- Heap-based allocation ensures nodes with most available space receive more storage.
- System scales horizontally as new nodes join.

Techstack & Budget Estimates

The techstack for this project includes: C/C++, gRPC, Protobuf, Sodium, Angular, SQLite, Infer (for static code analysis & identifying memory leaks).

Estimating the cost for such a system at such an early stage is not possible. However, following is the breakdown for a small scale prototype:

Sr. No.	Category	Estimated One Time Cost	Estimated Monthly Cost
1.	Hardware	$2,350 – $5,200	-
2.	Power & Cooling	–	$25 – $100
3.	Internet & Networking	–	$100 – $200
4.	DNS & Domain	$10 – $50	$5 – $15
5.	Development & Misc.	$1,500 – $5,000	–

Thank You