

# Introduction to IPFS: A Simple File Sharing Application

Maros Bednar

# 1. Problem(s) statement / Research question(s)

- PS1. Centralized File Storage Limitations:** Traditional file storage systems often rely on centralized servers, which can become bottlenecks, leading to limited scalability and potential performance degradation as the load increases.
- PS2. Security and Privacy Concerns:** Centralized data storage can pose privacy issues, especially when handling sensitive information, as these systems may not guarantee adequate privacy measures.
- PS3. Speed of Data Transfer:** With centralized data there is often a problem that servers may be far away from the devices.

## 2. Proposal

- Pr1. Decentralized Storage with IPFS:** I am going to implement a file-sharing application in Python utilizing IPFS's distributed network to eliminate central points of failure.
- Pr2. Content Addressing:** Leverage IPFS's method of identifying files by their unique cryptographic hashes, ensuring data integrity and efficient retrieval.
- Pr3. Peer-to-Peer Network:** Utilize IPFS's P2P architecture to distribute files across multiple nodes, enhancing accessibility and redundancy. Also, it will probably increase data transfer speed, since data will be routed to the closest device with this hash.

### 3. Hypotheses

- H1. Increased Resilience:** The absence of a central server makes the system more robust against censorship and single points of failure.
- H2. Enhanced Security:** By distributing files across a decentralized network using hash-codes, the system reduces the risk of data breaches and unauthorized access.
- H3. Improved Performance:** IPFS's parallel downloading from the closest nodes can lead to faster file retrieval times compared to traditional methods.

Source <https://filebase.com/blog/comparing-ipfs-to-traditional-file-storage-systems/>

## 4. Experiments / Use cases

- E1. Development:** I will create a prototype application in Python that allows users to upload and retrieve files using IPFS, ensuring a user-friendly interface.
- E2. Testing:** Afterwards, performance evaluations to compare file retrieval times and reliability against traditional file-sharing methods will be created.
- E3. User Feedback:** Gather insights from potential users to assess usability, identify potential issues, and compose a steps to refine the application accordingly for the future using.

It will take me probably 60 hours to create the application and compose a deep analysis for this type of experiment.

## 5. Results

- R1. Functional Application:** I will deliver a working file-sharing system that leverages IPFS for decentralized storage and retrieval.
- R2. Performance Metrics:** Documentation to the application will also demonstrate measurable improvements in file access speeds and system resilience compared to centralized counterparts.
- R3. User Satisfaction:** Achieve positive feedback from users regarding the application's efficiency, security, and ease of use.
- R4.** Project will be successful when the application will work and the experiment results will fulfil my hypothesis.

## 6. Conclusions (& Opened questions / Future work)

- C1. Addressing Centralization Issues:** The project showcases how IPFS can effectively mitigate the vulnerabilities inherent in centralized file-sharing systems.
- C2. Future Implications:** This work paves the way for broader adoption of decentralized technologies, promoting a more resilient and open internet infrastructure.