P2POCKET

Mandip Thapa (077BCT044)

Manish Kunwar (077BCT045)

TABLE OF CONTENT

INTRODUCTION

BACKGROUND

- Digital storage surge
- Current storage solutions
  - Cloud service
  - Secondary storage device

PROBLEM

- Centralized architecture
- Limitation of centralized architecture
  - Single point of failure
  - Performance bottleneck
- Privacy and security concerns

FINALLY, WHAT'S P2POCKET?

- Decentralized storage sharing platform
- P2P network
- Distributed system
- Optimized utilization of network resources

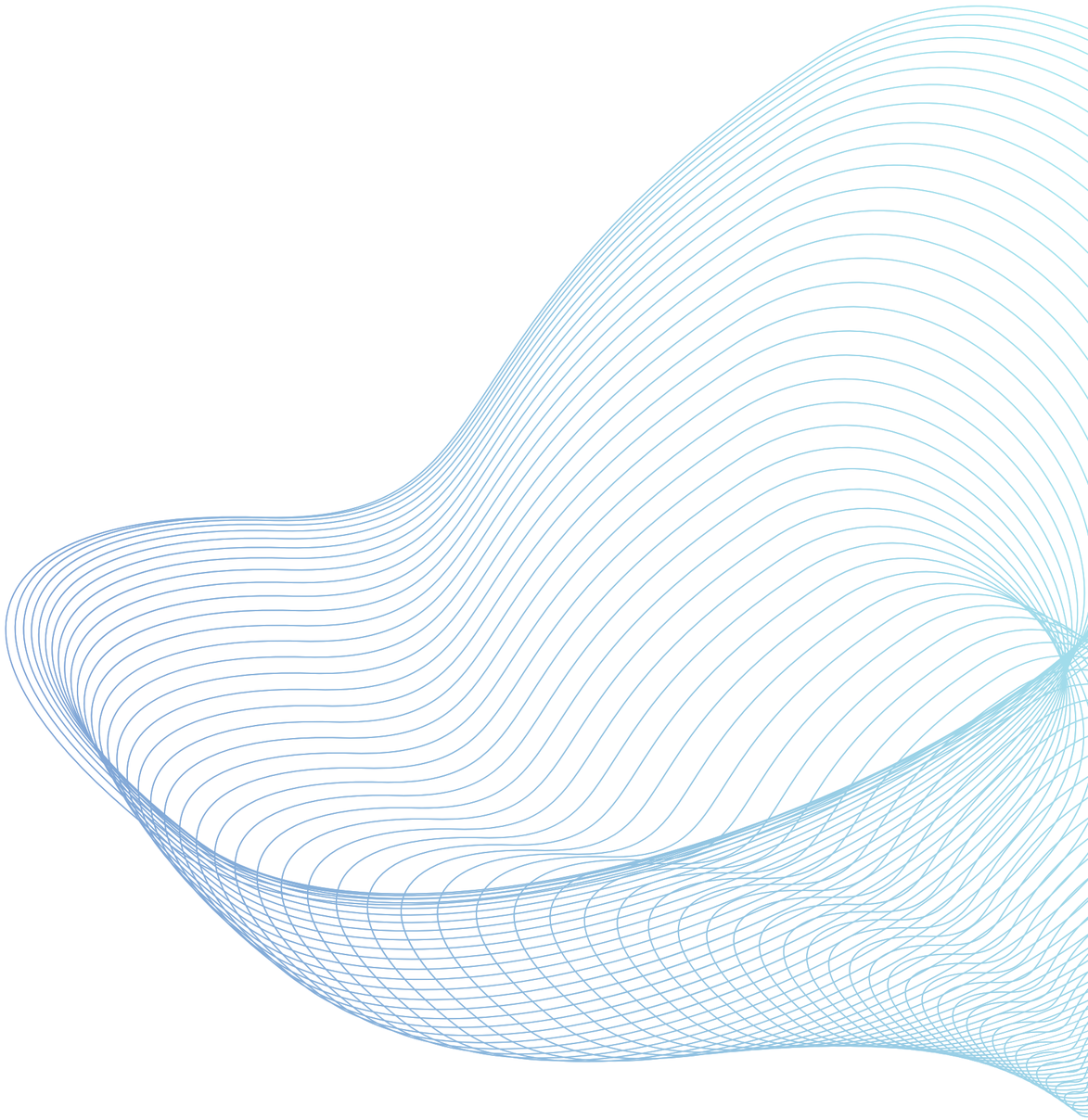
OBJECTIVES

- Develop a P2P system for efficient network storage
- Establish secure framework for storage sharing
- Implement decentralized storage for data integrity, reducing server dependency

SCOPE

- Practical alternative of traditional storage
- Feasibility evaluation
- Contribution to academic research

LITERATURE REVIEW



RELATED WORKS

- BitTorrent
- Napster
- Inter Planetary File System (IPFS)
- Filecoin

RELATED THEORY

- Distributed Hash Table (DHT)
- Kademlia
- Overlay network
- Encryption (AES)
- UDP Hole Punching

METHODOLOGY

FEASIBILITY STUDY

- Economic Feasible
- Time Feasible
- Technical Feasible

REQUIREMENT SPECIFICATION

- Functional requirement
  - Join network
  - Store and retrieve files from network
  - View the data stored in network
- Non functional requirement
  - Scalable
  - Data integrity
  - Optimal P2P connection

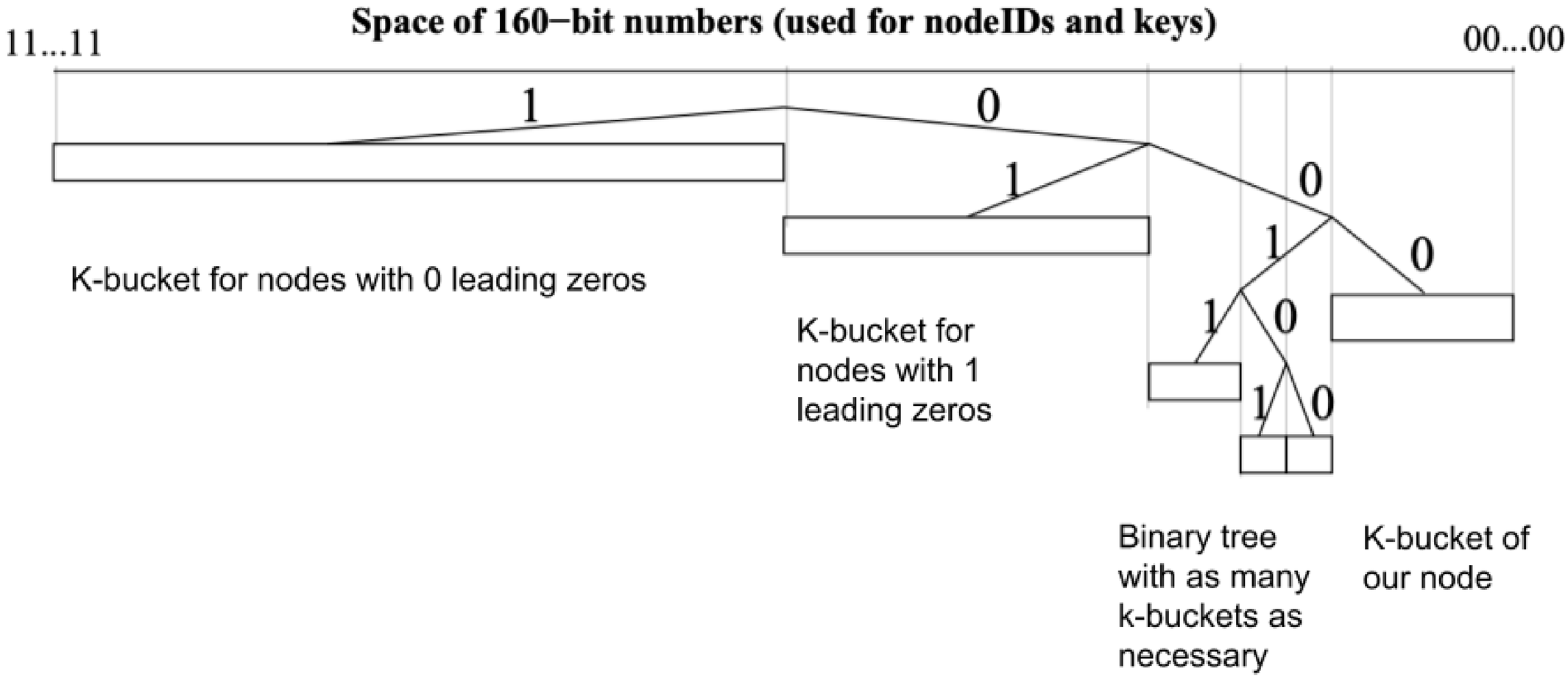
WORKING PRINCIPLES

ROUTING TABLE

- Routing table organizes nodes by distance.
- It uses buckets for different distance ranges.
- Each bucket holds nodes with similar ID prefixes.
- Closer nodes are in lower-index buckets.
- Each search iteration advances one bit towards the target.
- The table enables efficient node lookup and communication.

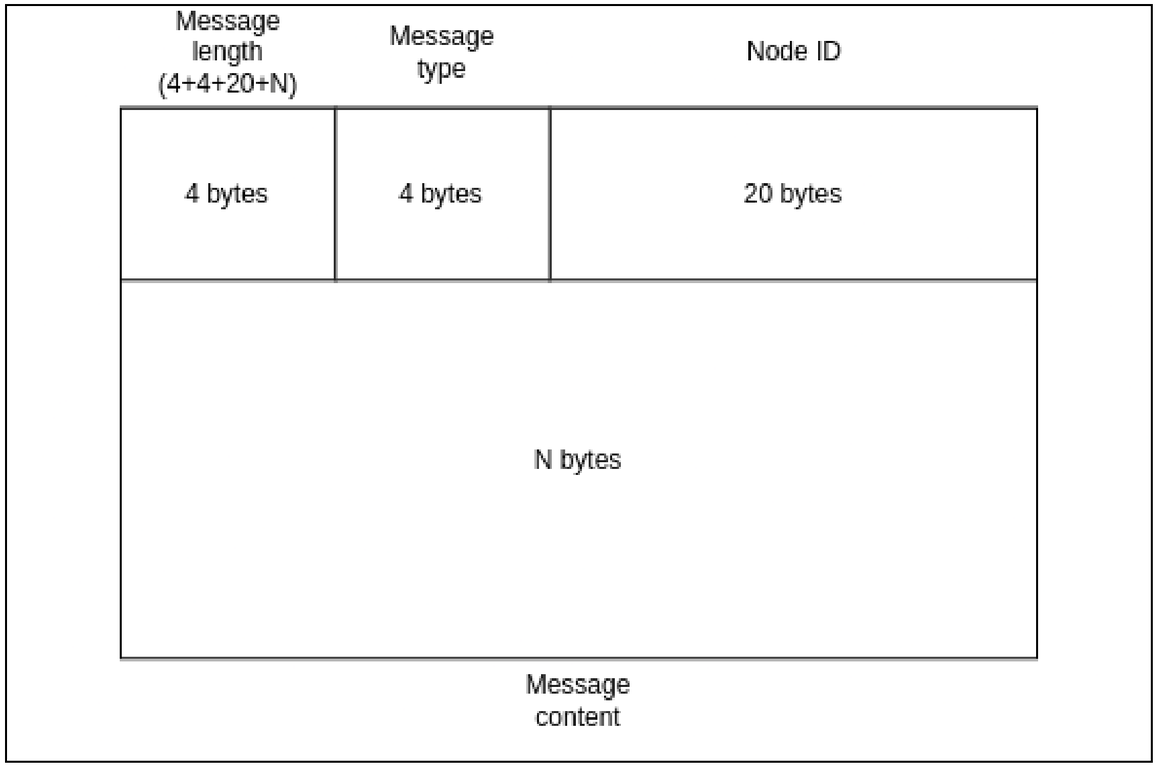
WORKING PRINCIPLES

- VISUAL REPRESENTATION OF ROUTING TABLE



WORKING PRINCIPLES

- NETWORK MESSAGE FORMAT



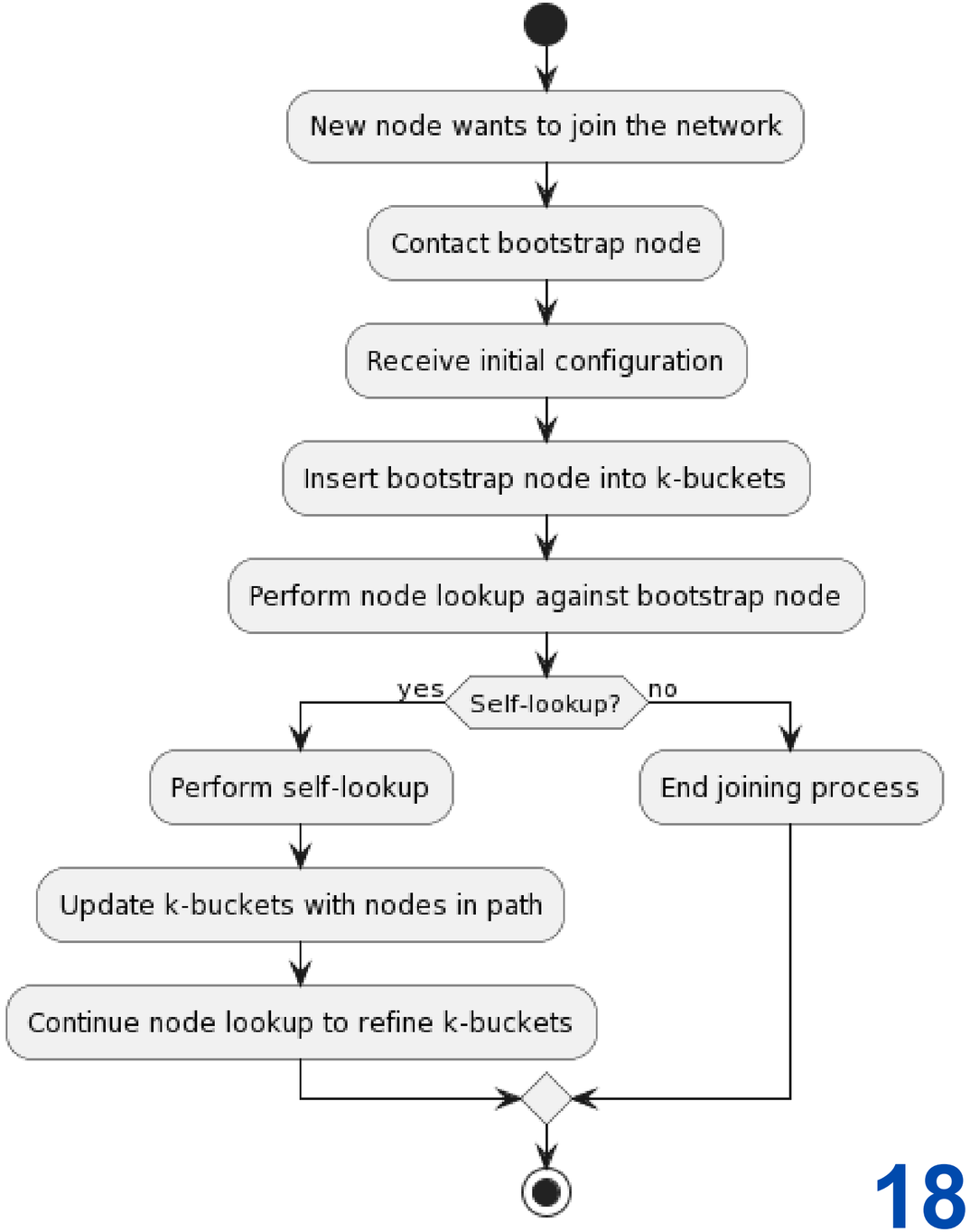
WORKING PRINCIPLES

- MESSAGE TYPE AND CORRESPONDING CONTENT

PING	Simple data packet
STORE	The <key,value> pair
STORE_RESPONSE	Success or Failure
FIND_NODE	ID of node
FIND_NODE_RESPONSE	Specific node info or K-Bucket
FIND_VALUE	Key of value to search
FIND_VALUE_RESPONSE	Value or K-Bucket

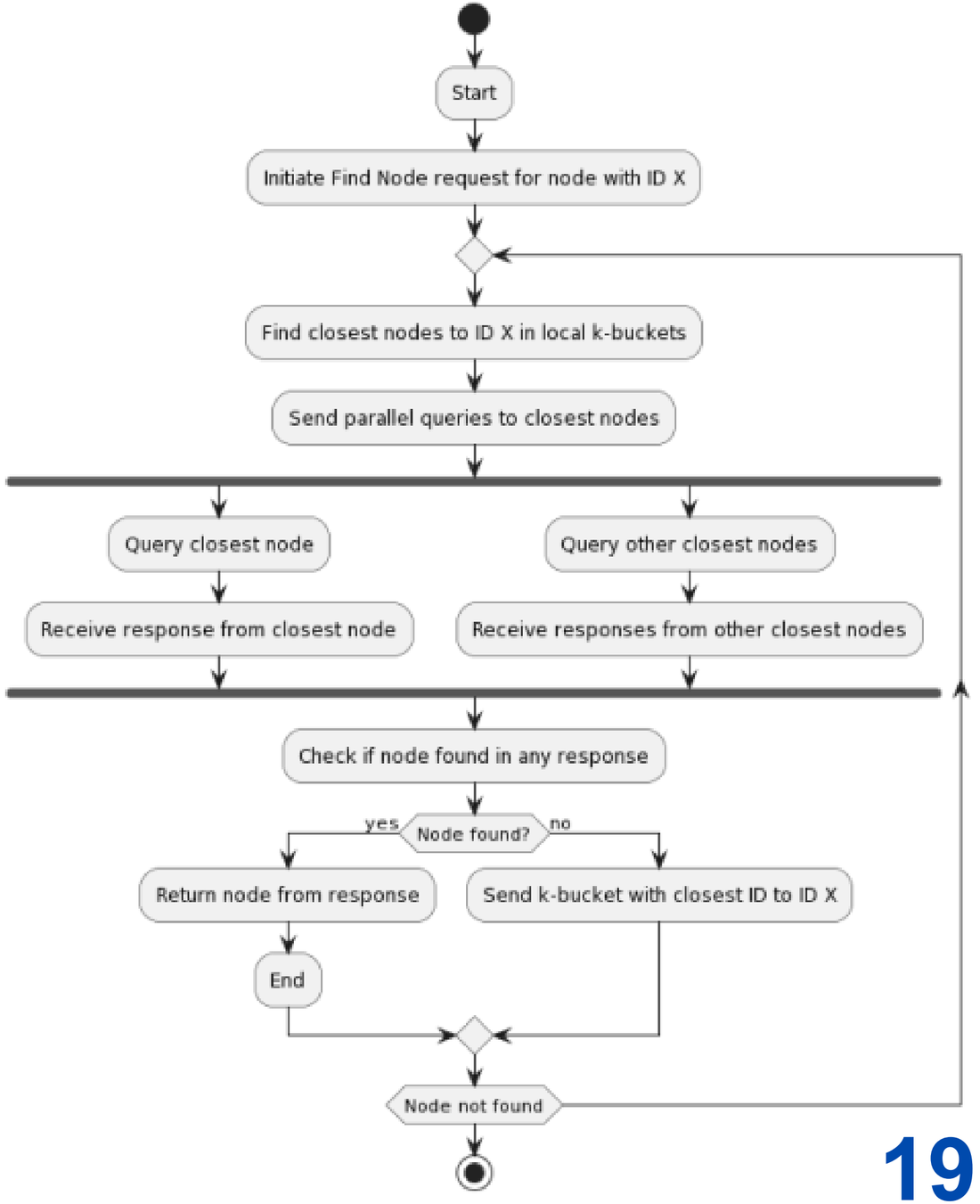
WORKING PRINCIPLES

- JOINING THE NETWORK



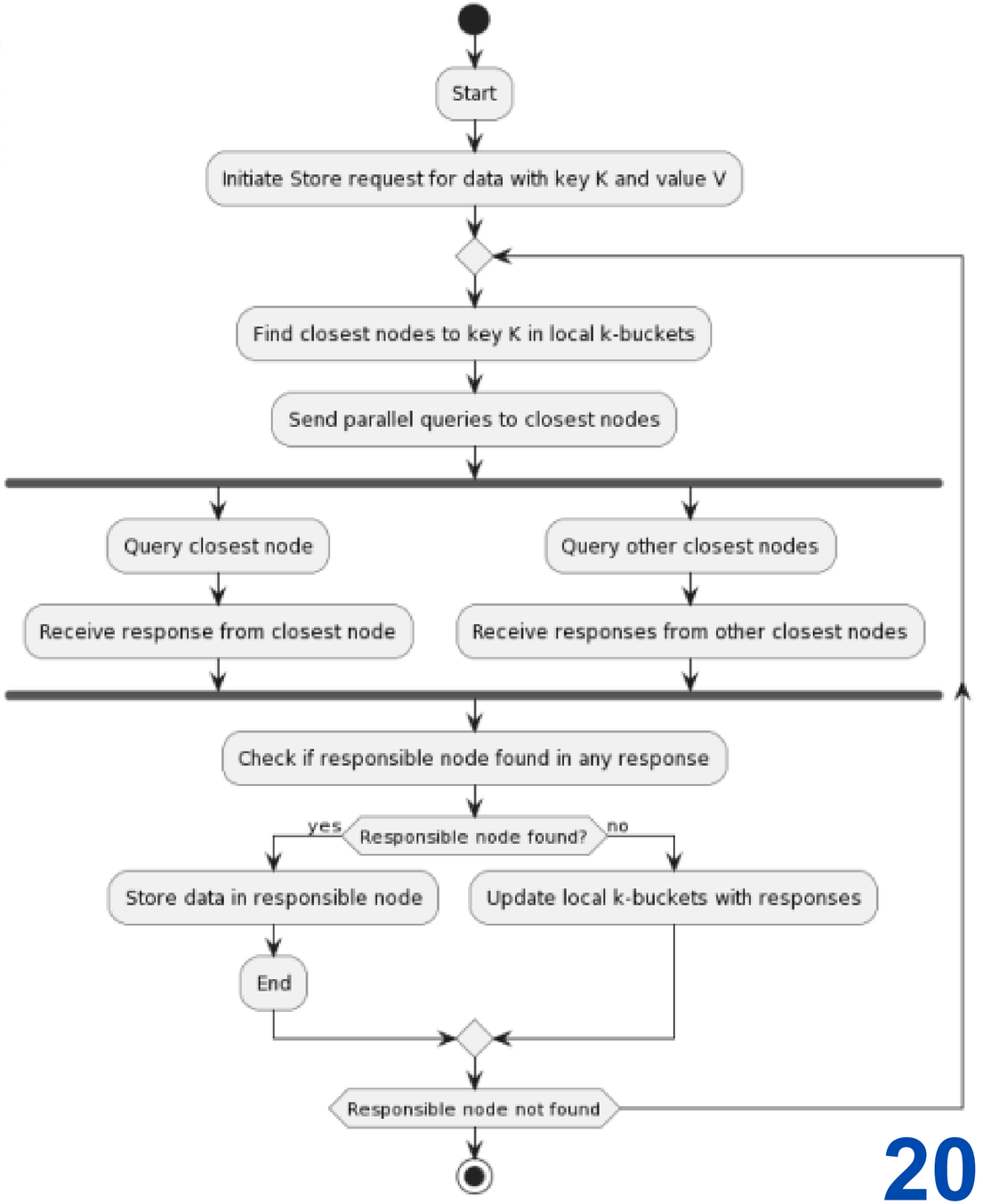
WORKING PRINCIPLES

- LOCATING NODES



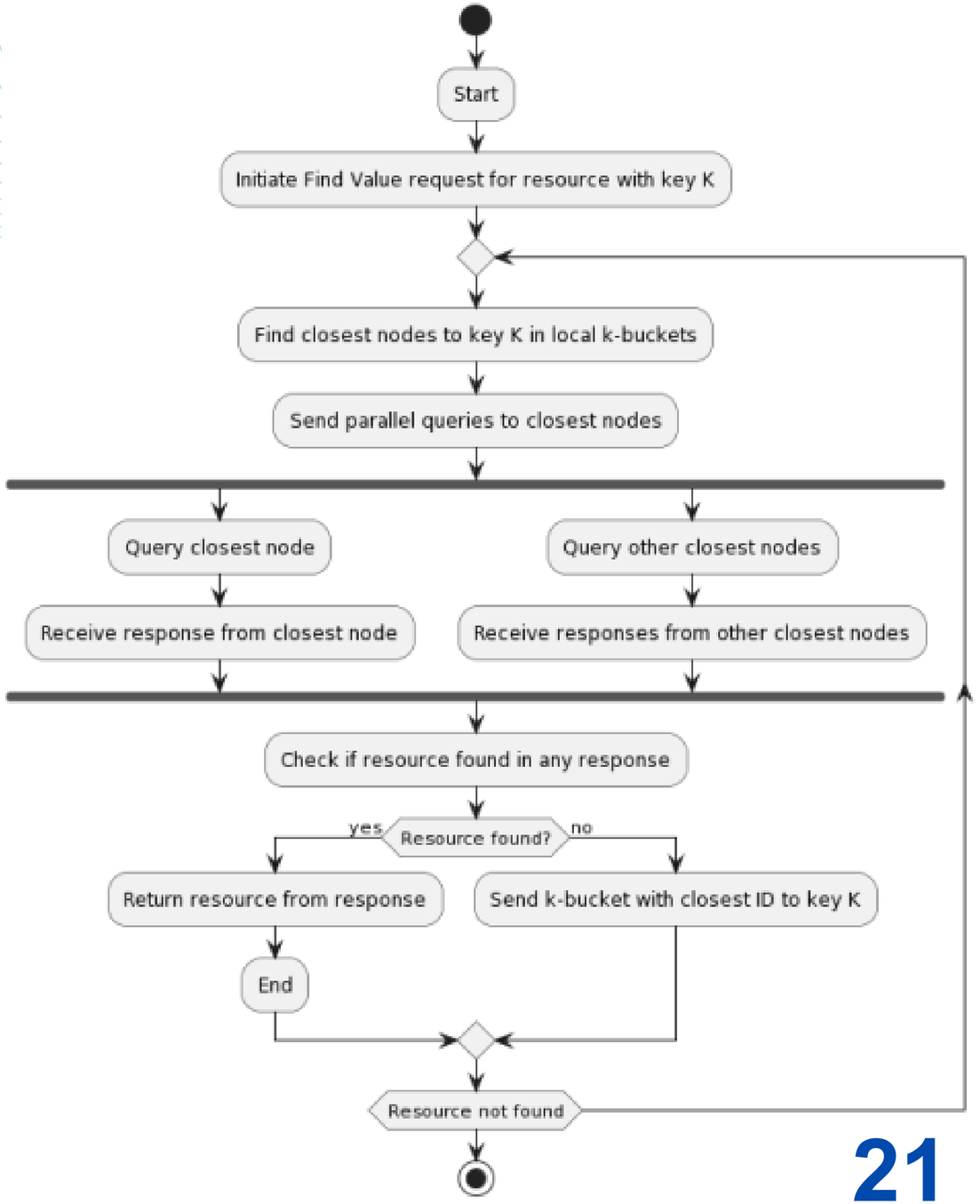
WORKING PRINCIPLES

- STORING IN NETWORK



WORKING PRINCIPLES

- LOCATING RESOURCES



SYSTEM DESIGN

COMPONENTS

- KADEMLIA
  - Kademlia computes distance using XOR between node IDs.
  - Node IDs and keys share identical format and length.
  - Node IDs are large, unique random numbers.
  - Node ID, IP and PORT are stored and maintained in routing table.
  - Each search iteration advances one bit towards the target.
  - Basic Kademlia search has O(log2(n)) complexity.

COMPONENTS

- ASYNCHRONOUS I/O (ASIO)

  - ASIO enables efficient I/O operations in network programming and high-performance computing.
  - Unlike synchronous I/O, ASIO allows concurrent task execution, reducing latency.
  - Programs can continue functioning while I/O operations progress asynchronously.
  - This boosts CPU utilization and sustains application responsiveness during intensive I/O tasks.

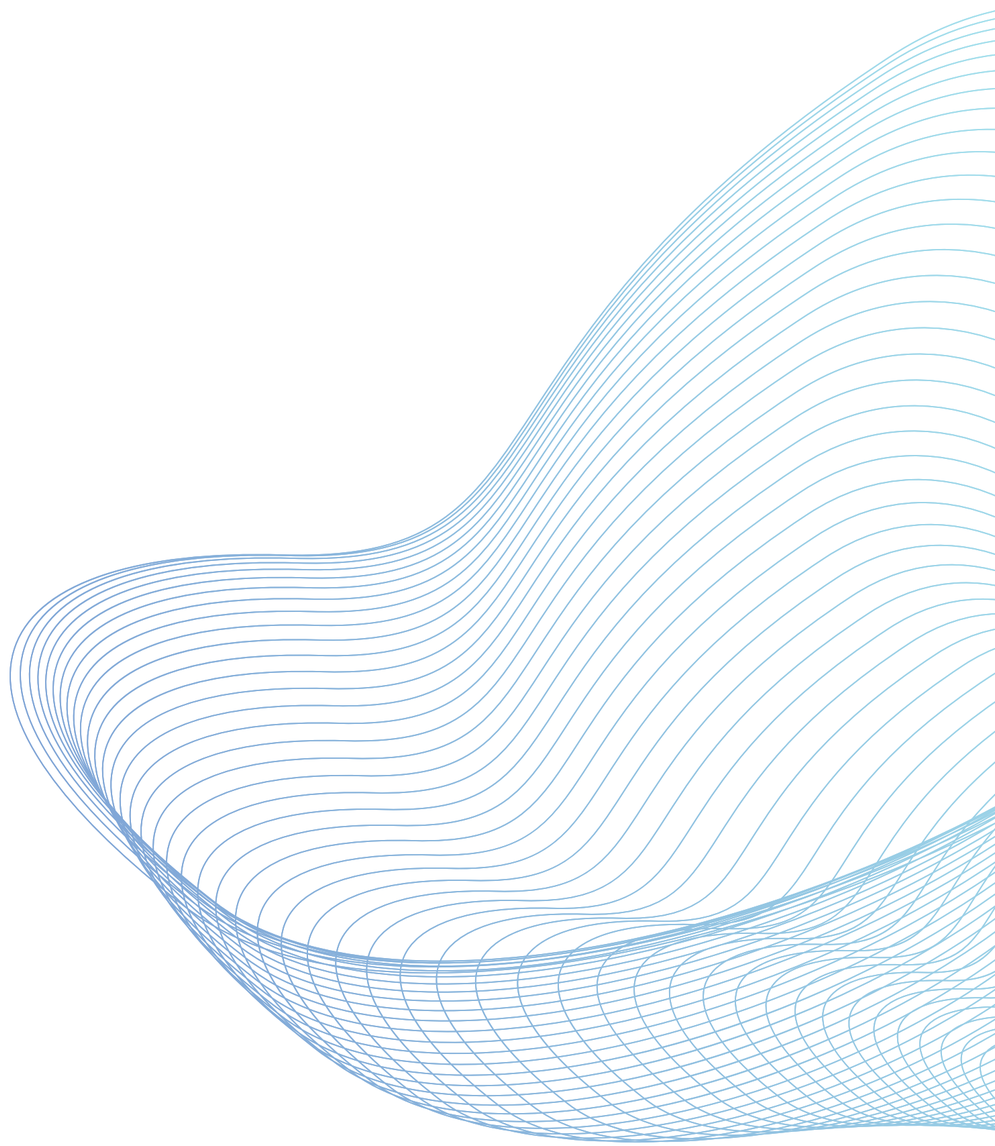
COMPONENTS

- MONITORING

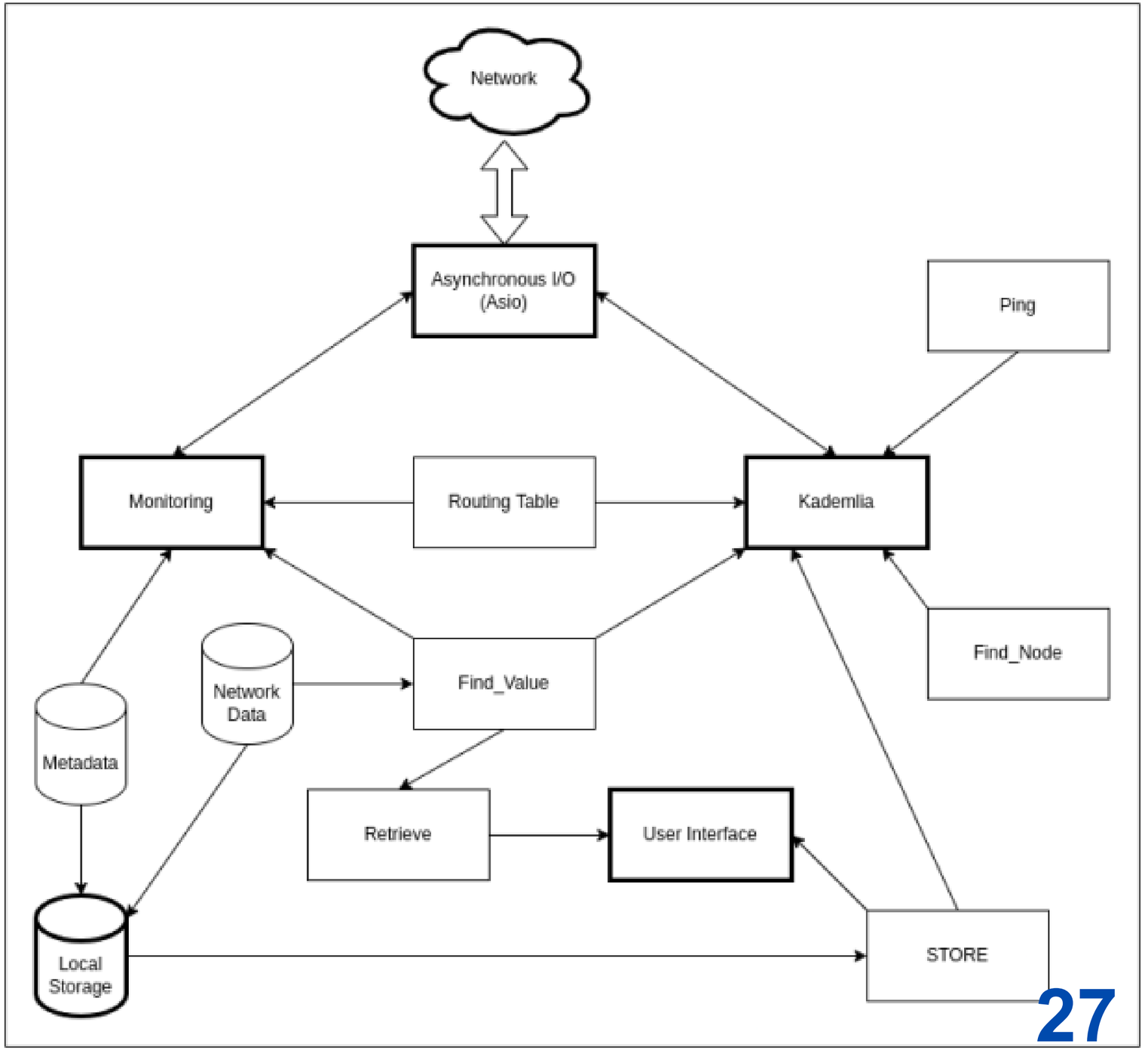
  - "Monitoring" ensures file persistence in the network.
  - It uses metadata to track file locations and status.
  - Nodes holding file fragments are periodically queried.
  - Queries occur at least once every 24 hours to ensure file availability and completeness.

COMPONENTS

- COMMAND LINE INTERFACE
  - Help
  - Store in network
  - Retrieve from network
  - List files/folders stored in network

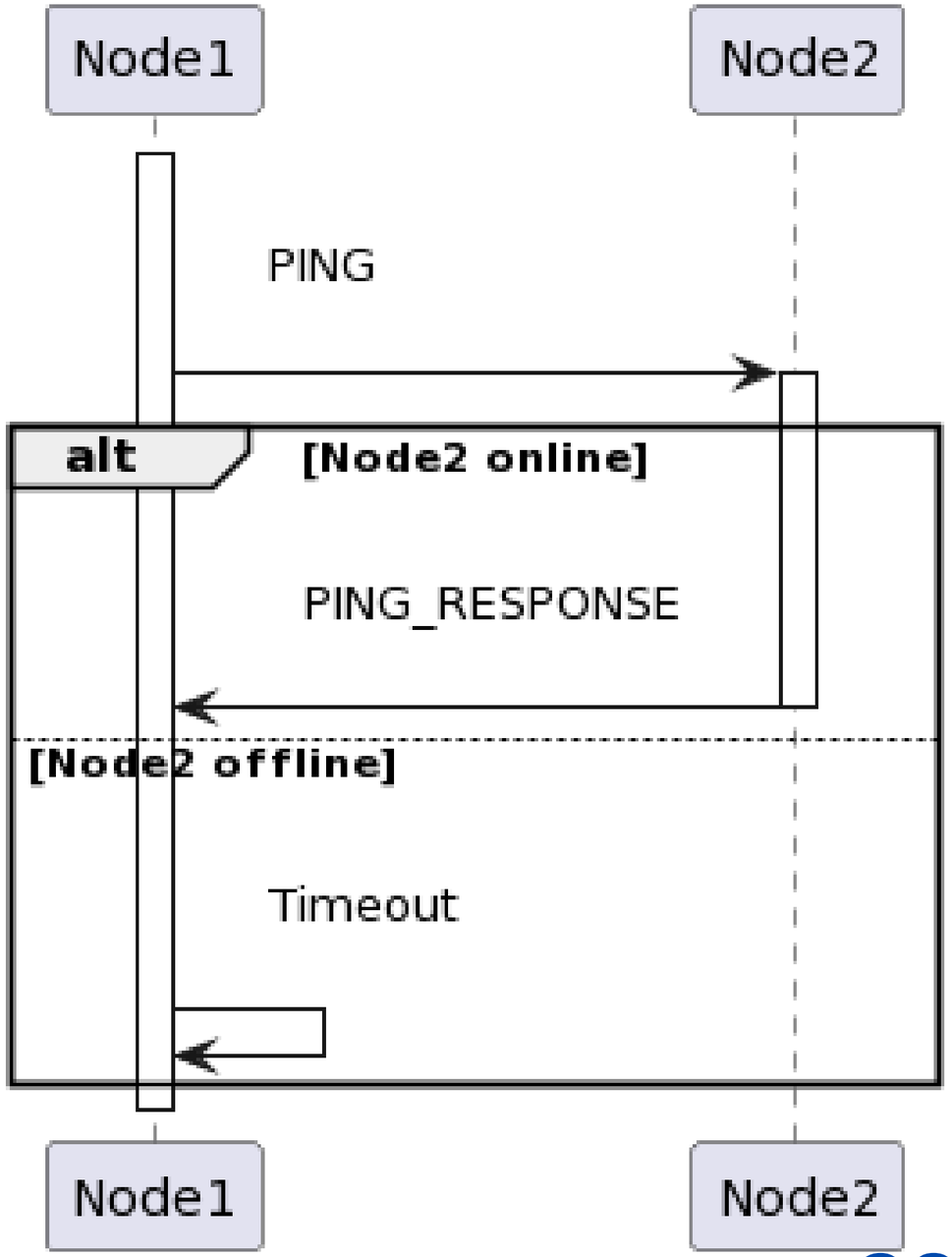


INTERACTION BETWEEN COMPONENTS



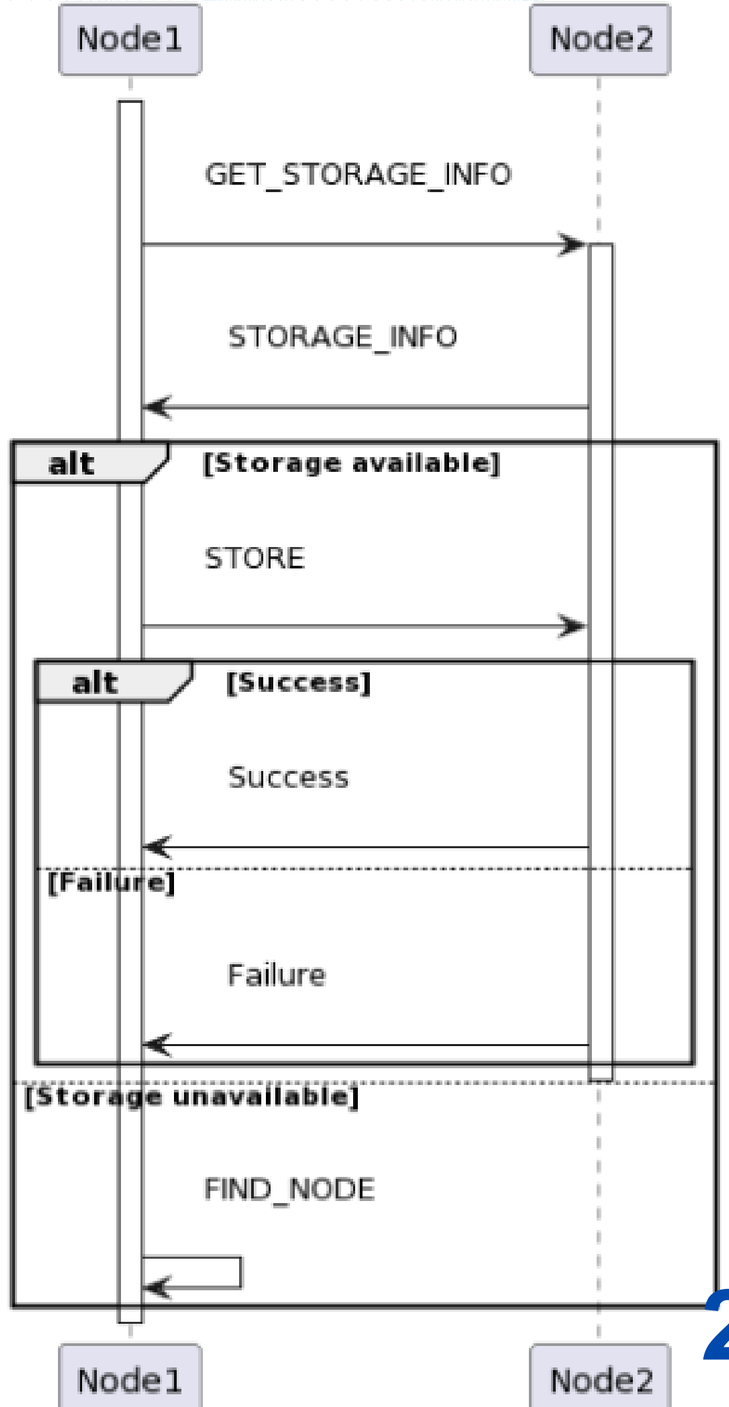
SEQUENCE DIAGRAM

- PING



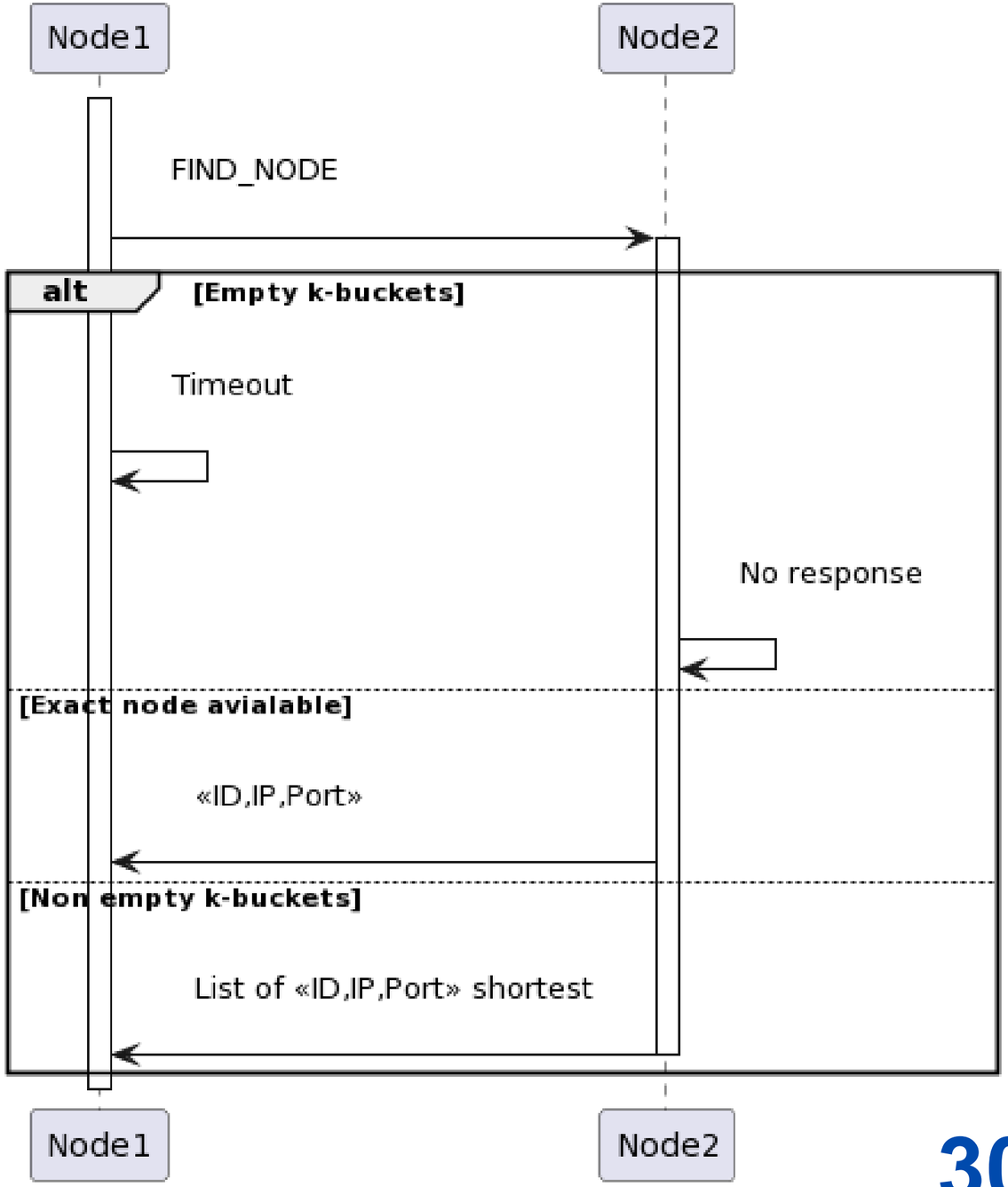
SEQUENCE DIAGRAM

- STORE



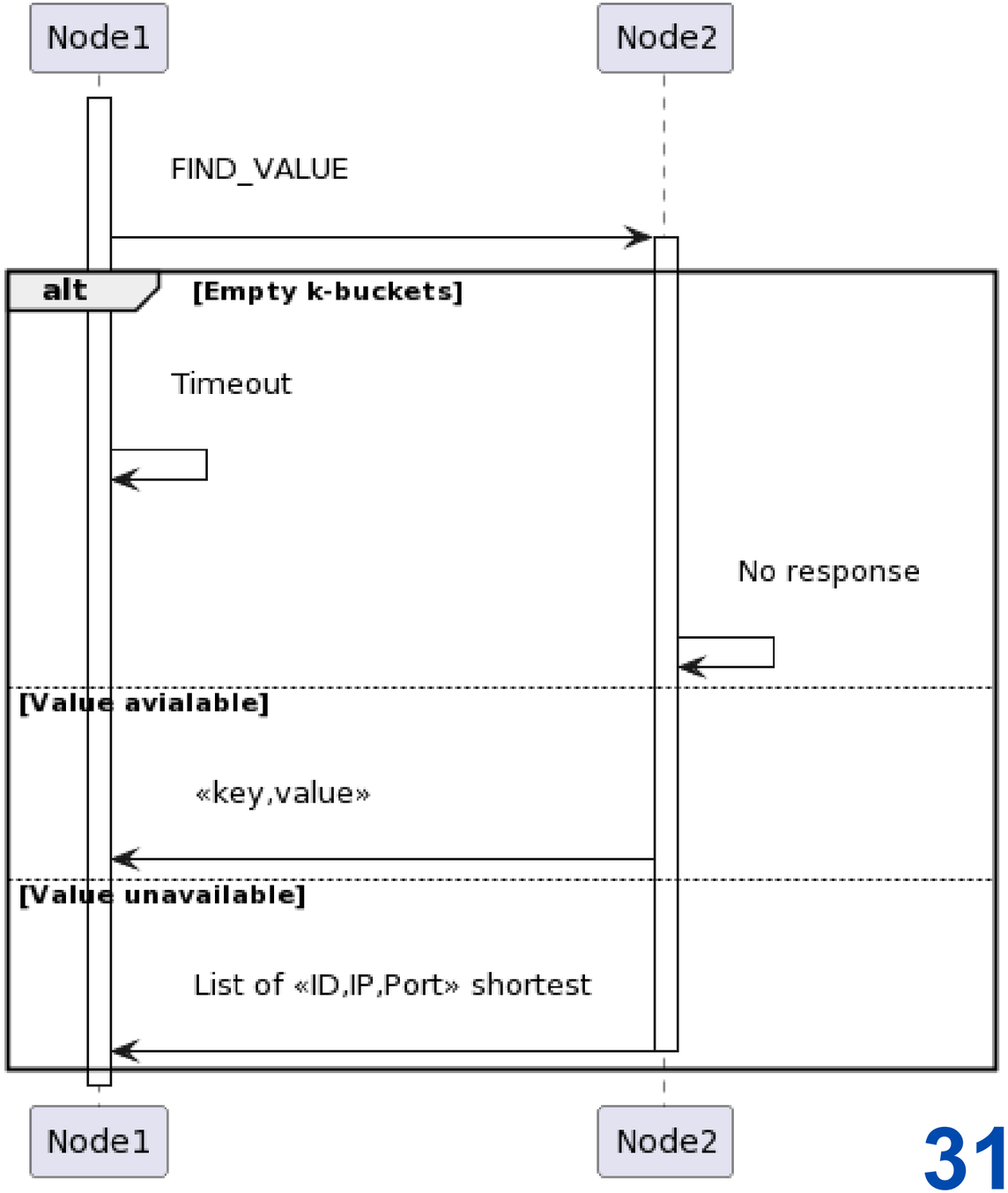
SEQUENCE DIAGRAM

- FIND_NODE



SEQUENCE DIAGRAM

- FIND_VALUE



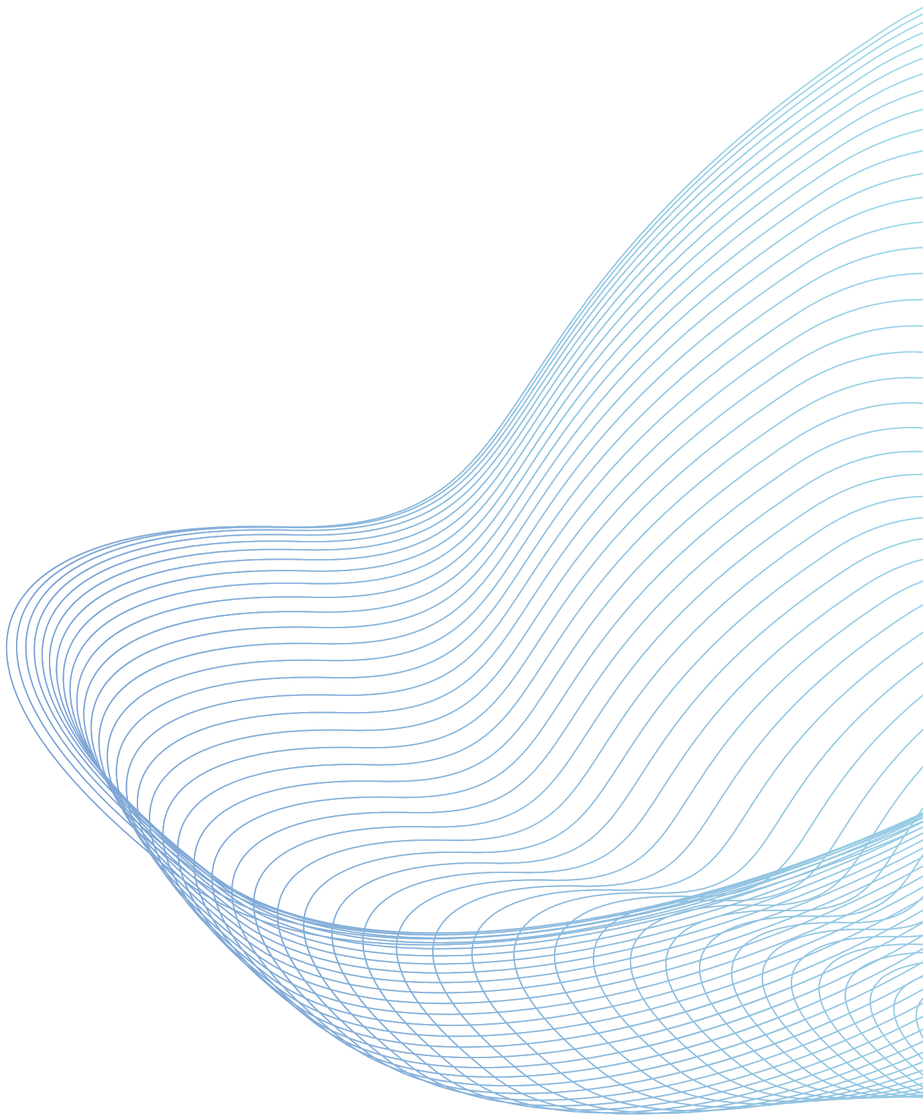
DEMONSTRATION

RESULT AND DISCUSSION

- Store
  - The files are encrypted separated into the smaller chunks of pieces.
  - Unique id is created for every chunk of pieces through hashing.
  - Shortest node is shortlisted for every pieces and stored in network.
- Retrieve
  - Every pieces are collected from network utilizing knowledge from metadata.
  - Pieces are merged and decrypted.

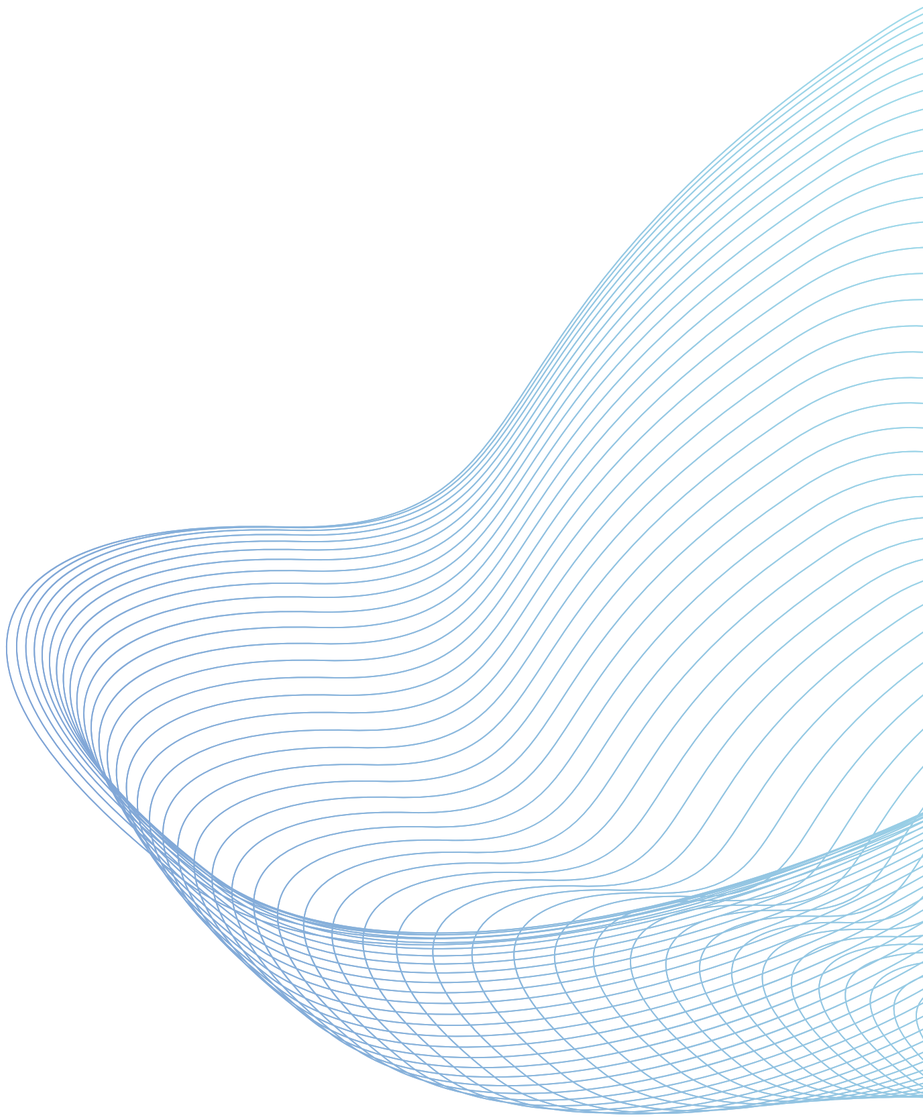
CONCLUSION

- Decentralized P2P storage.
- Maximizes storage utilization.
- Guarantees data integrity and confidentiality.
- Offers reliable storage with scalability.
- It drives innovations in digital storage.



LIMITATIONS AND FUTURE ENHANCEMENTS

- Limitation
  - Hole punching
  - Lack of file status checking
  - Broadcast IP change
- Future Enhancement
  - User interface
  - Handle large files
  - Proper error handling
  - Incentivize participation



REFERENCES

- Bram Cohen. Bittorrent - a new p2p app. 2001
- Petar Maymounkov and David Mazi`eres. Kademlia: A peer-to-peer information system based on the xor metric. 2002.
- Bryan Ford and Pyda Srisuresh. Peer-to-Peer Communication Across Network Address Translators. 2005.

THANK YOU