**1. High-Level Design (HLD) Document: P2P File Sharing Application**

**Overview**

This P2P File Sharing Application enables peers to share and download files in a decentralized manner. It uses a central tracker for peer discovery, supports chunked file transfers, verifies data integrity using checksums, and downloads chunks in parallel.

**Components**

**1. Tracker Server**

* **Responsibility**: Maintains a directory of peers and files.
* **Technology**: Python socket server.
* **Functions**:
  + Registering files from peers.
  + Providing peer lists to requesting clients.

**2. Peer Client**

* **Responsibility**: Serves files to other peers, downloads files in parallel, verifies data integrity.
* **Technology**: Python socket client + threading + concurrent.futures.
* **Functions**:
  + Registering files with tracker.
  + Starting server to send file chunks.
  + Downloading file chunks from other peers.
  + Merging downloaded chunks into a single file.
  + Validating chunk data using SHA-256 checksum.

**Data Flow**

1. **Peer A** starts its server and registers "file.txt" with the tracker.
2. **Peer B** contacts the tracker to download "file.txt".
3. Tracker returns a list of peers who have the file.
4. Peer B uses threads to download different chunks from available peers.
5. Each chunk is validated with a SHA-256 checksum.
6. Once all chunks are downloaded and verified, they are merged into the complete file.

**Design Decisions**

* **Chunking**: Improves parallelism and efficiency.
* **SHA-256 Checksums**: Ensures data integrity.
* **Threading**: Enables concurrent handling of peer requests and parallel downloads.
* **Minimal Dependencies**: Uses only standard Python libraries.

**Future Improvements**

* NAT Traversal support (e.g., STUN for NAT hole punching).
* Persistent database in the tracker.
* Retry logic for failed chunk transfers.
* GUI for user-friendly operation.