**1. Selection of MPI Implementation:**

For our file transfer system using MPI, we chose Open MPI as our MPI implementation. We selected Open MPI due to its widespread adoption, robustness, and support for various platforms and architectures. Open MPI provides a rich set of features for message passing between processes, making it suitable for building parallel and distributed applications like our file transfer system.

**2. Design of MPI Service:**

In our MPI service design:

* The server process receives files from the client process using MPI\_Recv.
* The client process sends files to the server process using MPI\_Send.
* The server process writes the received files to disk.

**3. Organization of the System:**

In our system organization:

* The client and server processes run independently on different nodes or cores.
* Each process handles its respective tasks of sending or receiving files.
* The client initiates the file transfer process and communicates with the server using MPI.

**4. Implementation of File Transfer:**

In implementing the file transfer system using MPI:

* The client process reads the file to be sent and sends it to the server process using MPI\_Send.
* The server process receives the file data from the client using MPI\_Recv and writes it to disk.
* Both processes use MPI functions for communication and synchronization, ensuring reliable and efficient file transfer.