**Design of RPC Service:**

1. **Service Definition (file\_transfer.proto):** The RPC service is defined using Protocol Buffers (protobuf) language. It specifies the FileTransfer service with two RPC methods: SendFile and ReceiveFile. SendFile takes a FileRequest message containing the filename and file content as input and returns a FileResponse indicating success or failure. ReceiveFile takes a FileChunk message containing file content as input and returns an Empty message.
2. **Message Definitions:** The service uses custom message types like FileRequest, FileResponse, and FileChunk to exchange data between client and server. FileRequest contains the filename and content to be sent, FileResponse indicates the success status of the file transfer operation, and FileChunk carries chunks of file content during transmission.

**Organization of System:**

1. **Server-side Organization:**
   * **Server Code (server.cpp):** The server code initializes a gRPC server, registers the FileTransfer service implementation, and listens for incoming requests. It defines a FileTransferServiceImpl class that implements the RPC methods SendFile and ReceiveFile. Each method handles file transfer operations on the server-side, such as writing received files to disk.
2. **Client-side Organization:**
   * **Client Code (client.cpp):** The client code establishes a connection with the server using gRPC. It defines a FileTransferClient class that encapsulates the gRPC stub and provides methods to send and receive files. The client sends the file to the server using SendFile and receives the file from the server using ReceiveFile.

**Implementation of File Transfer:**

1. **Sending File (SendFile RPC):**
   * The client reads the content of the file to be sent and constructs a FileRequest message containing the filename and content.
   * The client calls the SendFile RPC method on the server, passing the FileRequest message.
   * On the server-side, the FileTransferServiceImpl receives the FileRequest, extracts the filename and content, and writes the content to a file on the server's disk.
   * After successful writing, the server sends back a FileResponse indicating the success status to the client.
2. **Receiving File (ReceiveFile RPC):**
   * The server listens for incoming FileChunk messages from the client.
   * The client reads the content of the file to be sent in chunks and constructs FileChunk messages.
   * The client calls the ReceiveFile RPC method on the server, passing FileChunk messages containing file content.
   * On the server-side, the FileTransferServiceImpl receives the FileChunk messages, extracts the content, and appends it to the file being received.
   * After receiving all chunks, the server sends back an Empty response to the client.