HTTP2 Server

Components:

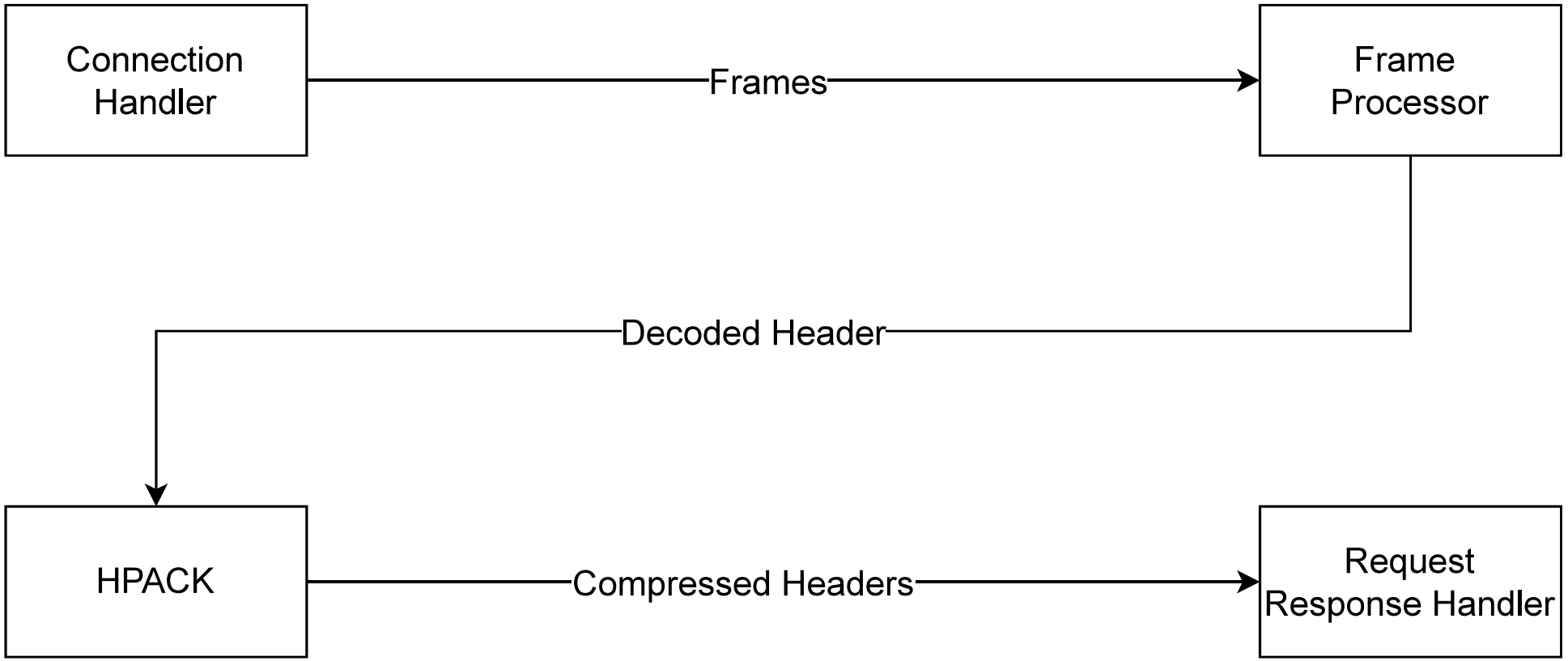
- **Connection Handler:** initiates and manages the connection between the server and the client and managing the TLS handshake.
- **Frame processor:** Handles the frames
- **HPACK Compression/Decompression Module:** Implements header compression and decompression as in RFC 7541
- **Request/Response Handler:** Parses HTTP requests and formulates responses
- **Stream Manager:** Manages individual streams
- **Error Handler:** Detects and handles protocol errors.
- **Performance and Optimization Module:** Implements flow control and prioritization and Optimizes frame sizes and communication.

Interaction Between Components:

1. The **Connection Handler** initiates connections and passes the frames to the **Frame processer**.

2. The **Frame Processor** decodes incoming frames and forwards them to the **HPACK Module.**

3. The **HPACK Module** compresses and decompresses headers and passes it to the **Request/Response Handler.**


Connection Handler
Frames
Frame Processor
Decoded Header
HPACK
Compressed Headers
Request Response Handler

User Authentication Mechanism:

User Authentication here is the TLS handshake that is done by the connection handler component. Using the TLS handshake the client and the server share their certificates to be validated and share the encryption keys.

1. **Server Sends Certificate**:
   - The server sends its certificate (containing its public key) to the client during the handshake.
   - The certificate is issued by a trusted Certificate Authority (CA).
2. **Client Validates Server Certificate**:
   - The client checks if the server certificate is signed by a trusted CA and valid for the domain.
3. **Encryption Key Exchange**:
   - The client and server use asymmetric encryption (public/private keys) to securely exchange a pre-master secret or compute a shared secret.
4. **Session Key Derivation**:
   - Both client and server derive the symmetric session key from the shared secret to encrypt and decrypt data for the session.

Key Functionalities of Your Web Server:

1. **Handle Client Connections**
   - Accept incoming connections from clients
   - Maintain persistent connections (keep-alive) to avoid re-establishing connections for each request.

2. **Protocol Handling (HTTP/2)**
   - Process HTTP/2 frames
   - Compress HTTP headers using HPACK to reduce bandwidth usage.

3. **Security Features**
   - Enforce HTTPS to ensure secure connections by redirecting HTTP to HTTPS.
   - Support TLS handshake for mutual authentication

4. **Error Handling and Custom Error Pages**
   - Return proper HTTP status codes.
   - Provide custom error pages.