Communication Styles

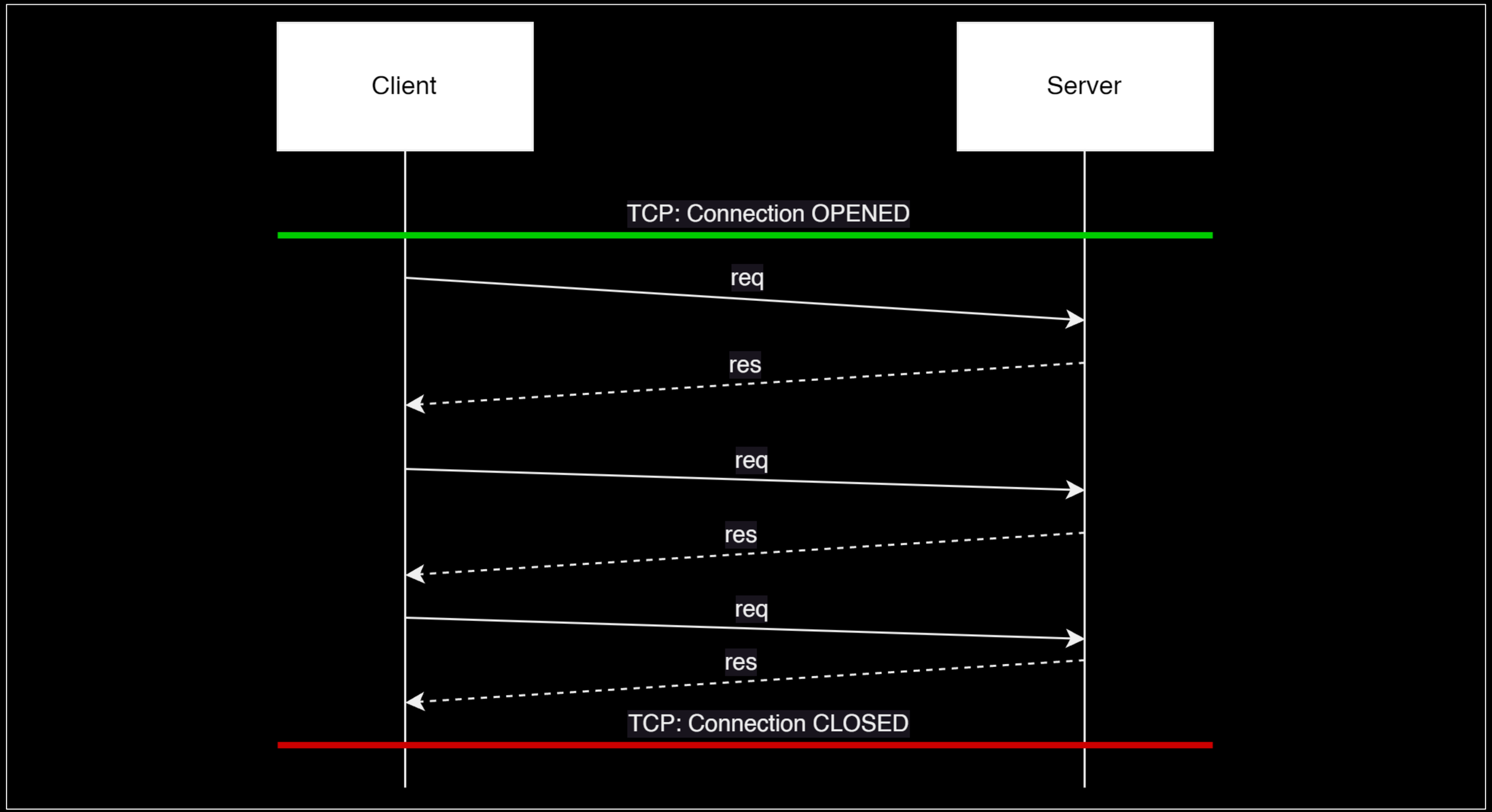
Systems Analysis & Design

Learning Objectives

By the end of this session, you will have acquired the following information:

- Request/Response
- Push
- Short Polling
- Long Polling
- Server-Sent Events

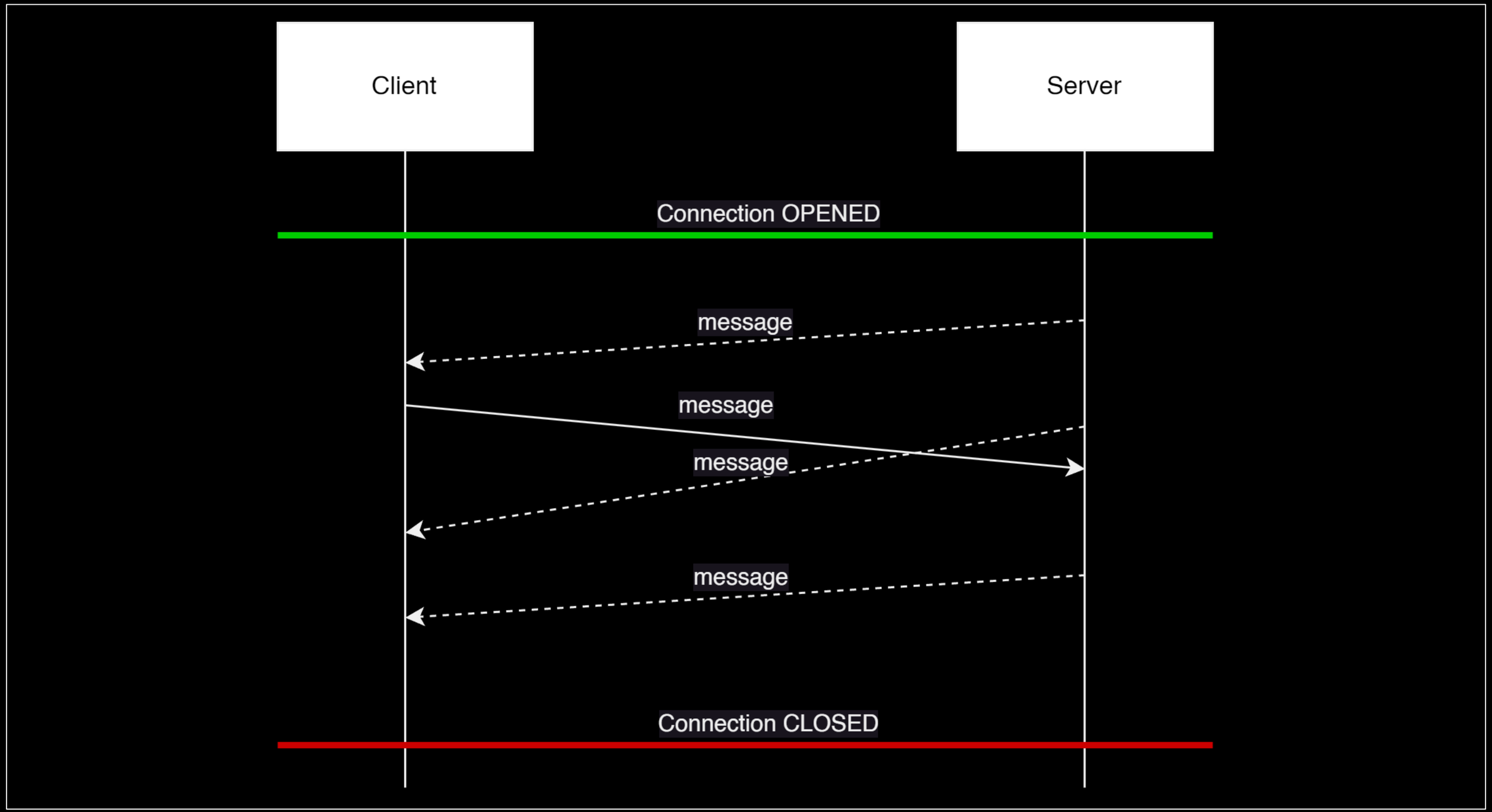
Request/Response


Client
Server
TCP: Connection OPENED
req
res
req
res
req
res
TCP: Connection CLOSED

- The request/response model is used in APIs: REST, GraphQL, gRPC.
- For example, if you want to upload an image with a request/response model, you can either send a large request with the image or chunk the image and send a request per chunk.

```python
@app.post("/upload/")
async def upload_file(file: UploadFile):
    with open("your_image.png", "wb") as buffer:
        chunk_size = 1024
        while content := await file.read(chunk_size):
            buffer.write(content)
    return {"filename": file.filename}
```

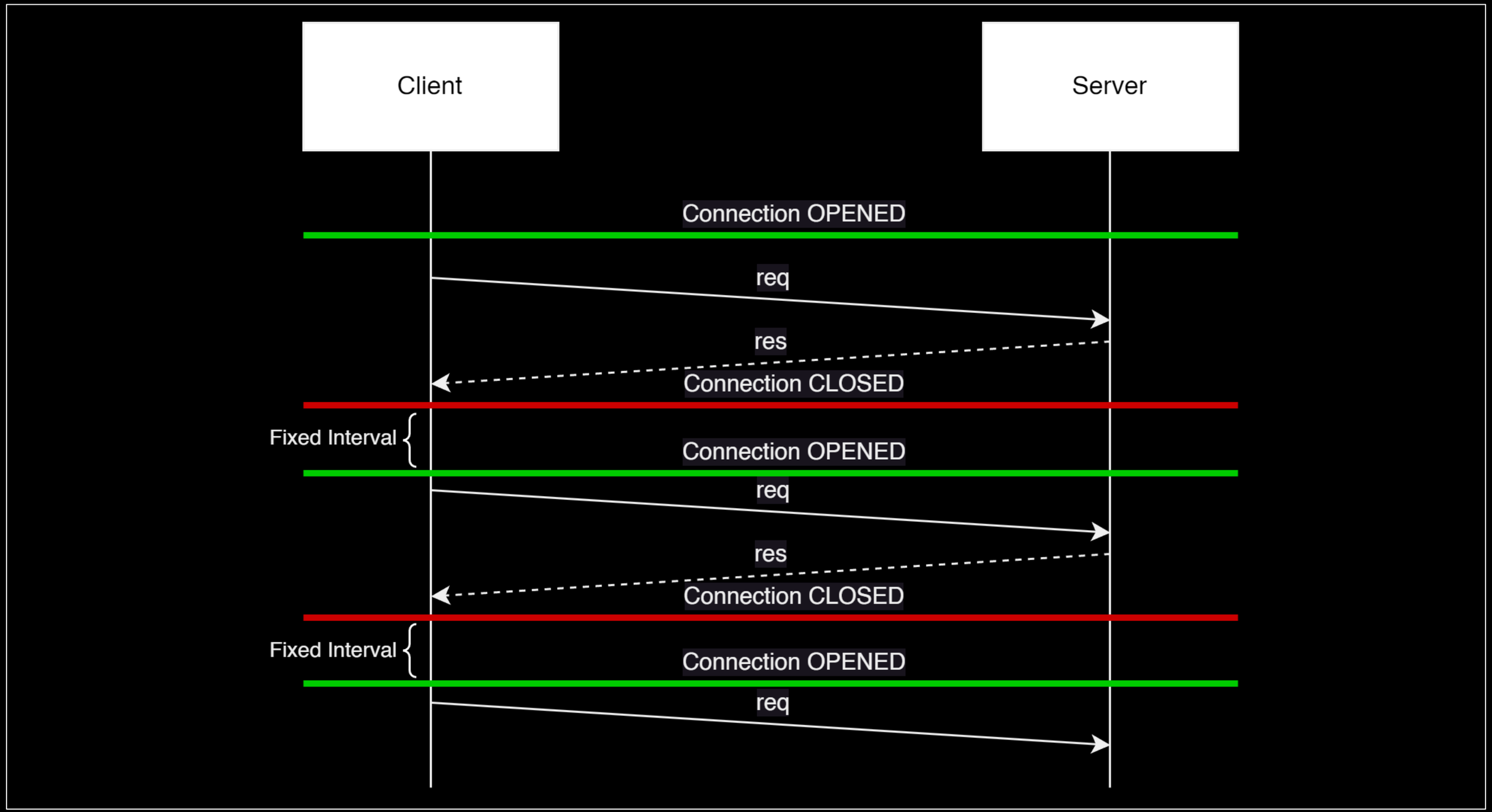
Push


Client
Server
Connection OPENED
message
message
message
message
Connection CLOSED

- The client connects to a server. The server sends data to the client. The client does not have to request anything.
- RabbitMQ and WebSocket use the push model.
- It is bidirectional.

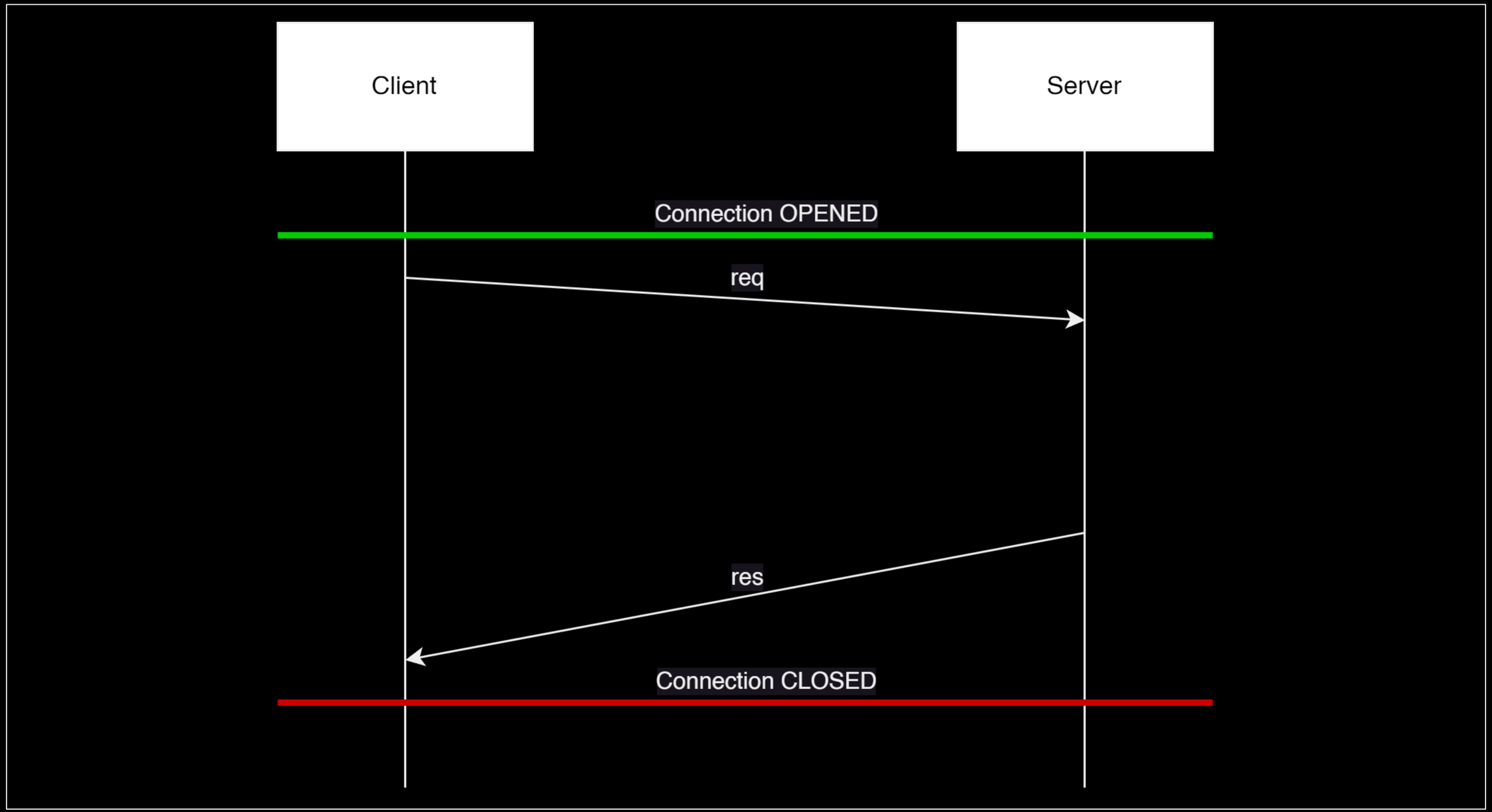
```
@Scheduled(fixedRate = 15000)
private void sendNotifications() {
  var json = """
      {
        "title": "Server says hello!",
        "body": "It is now: %s"
      }
      """;
  subscriptions.forEach(subscription -> {
    sendNotification(subscription, String.format(json, LocalTime.now()));
  });
}
```

Short Polling


Client
Server
Connection OPENED
req
res
Connection CLOSED
Fixed Interval
Connection OPENED
req
res
Connection CLOSED
Fixed Interval
Connection OPENED
req

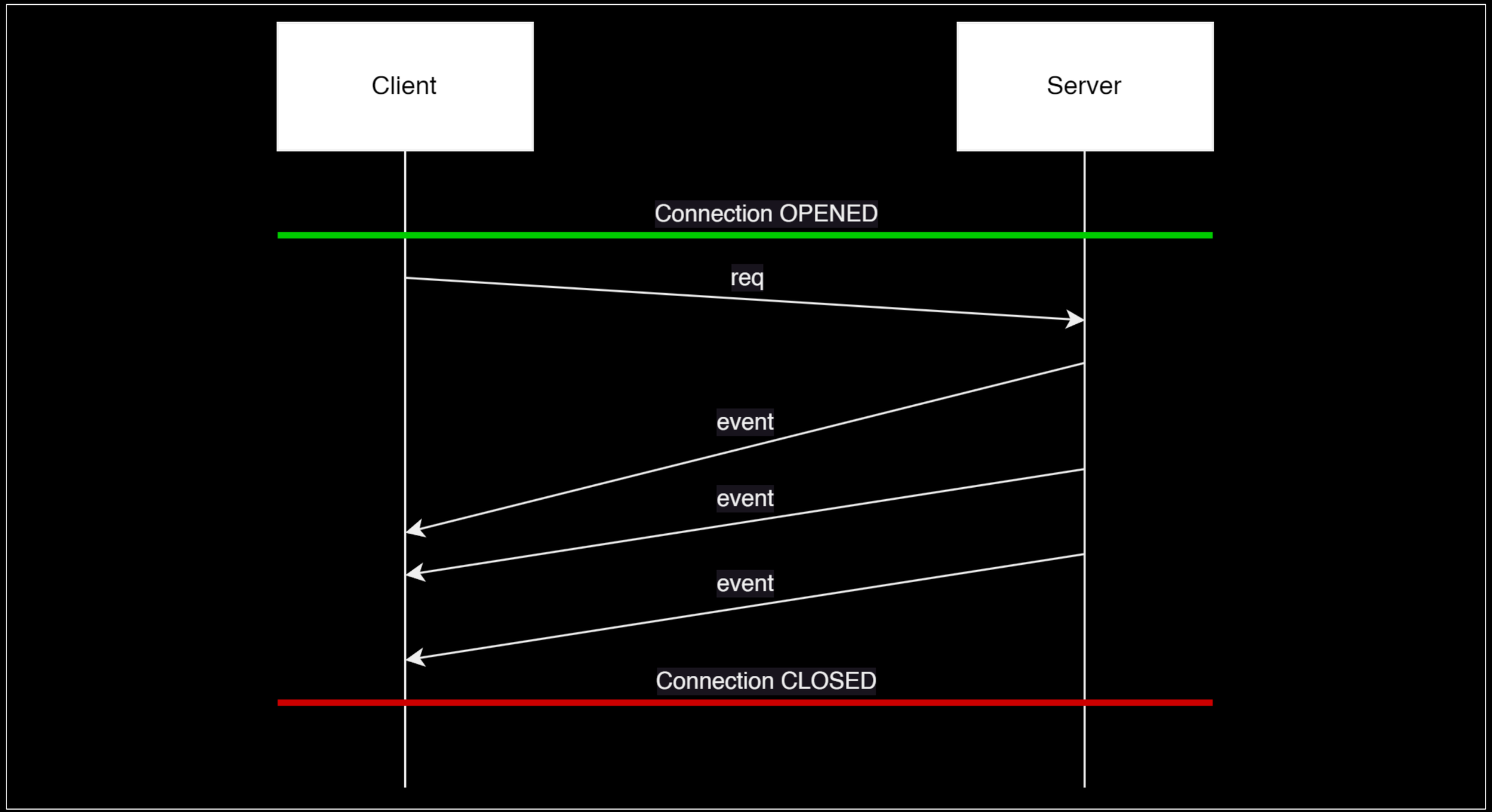
- Short polling is a technique used to periodically check for new data from a source.
- The client sends a request to the server, and the server sends a response to the client.
- This pattern continues at fixed intervals.
- The server will return the data if the response is ready; otherwise, it will immediately return an empty payload.
- Amazon SQS can utilize the short polling mechanism.

Long Polling


Client
Server
Connection OPENED
req
res
Connection CLOSED

- The client sends a request to the server.
- If the server has the payload, it will respond; otherwise, the connection will remain open until the server has the response.
- The connection will stay open for a period of time if it does not have the payload.
- Kafka uses the long polling model.

Server-Sent Events


Client
Server
Connection OPENED
req
event
event
event
Connection CLOSED

- A response has a start and end.
- The client sends a request, and the server sends logical events as part of the response.
- The server does not write the end of the response until it has sent all the events.
- The client parses the stream’s data, looking for events.
- This process works with the request/response model of HTTP.
- ChatGPT uses the server-sent events.

```
// Function to send response
const send = (res, counter = 0) => {
    res.write(`data: message ---- [${counter}]\n\n`);
    setTimeout(() => send(res, counter + 1), 2000);
}
// Handling GET request
const handleStreamRequest = (req, res) => {
    // Setting header
    res.setHeader("Content-Type", "text/event-stream");
    // Calling send function
    send(res);
};
```

Further Resources

- Design a Chat System