Introducing TUS

resumable file uploads protocol

HKOSCON 2019
Sunday Ku

Sunday Ku

Software engineer in HK01

Speaker

- HKOSCON 2018
- Show me the code

Agenda

- What and Why TUS
- How does TUS work
- How to implement TUS

What is TUS?

A File upload protocol

With Rich features

Resumable upload

Parallel upload

…

Using HTTP(s) only

Supported by ALL platforms with HTTP support

Open Source!!!

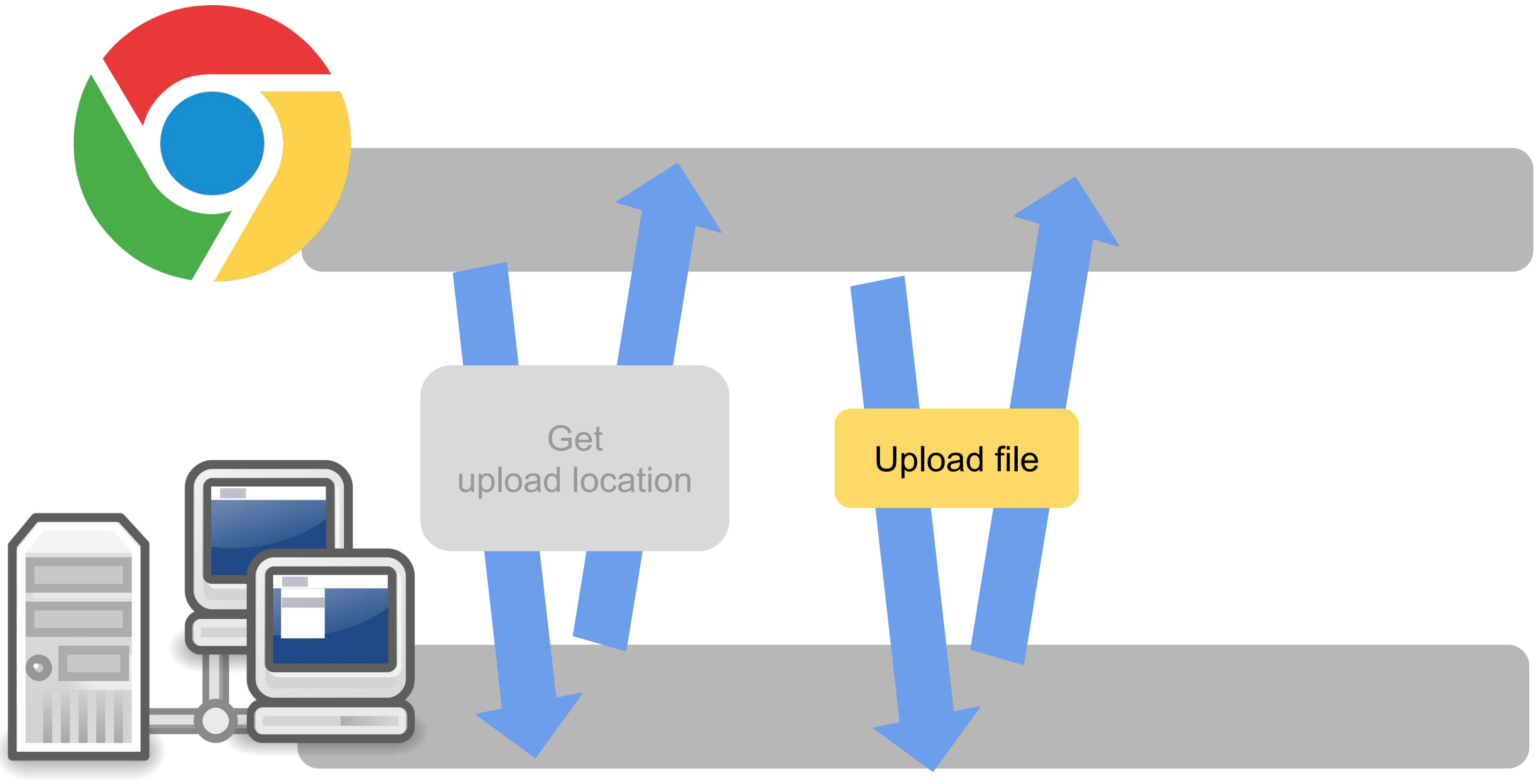
Library available in many languages

Demo

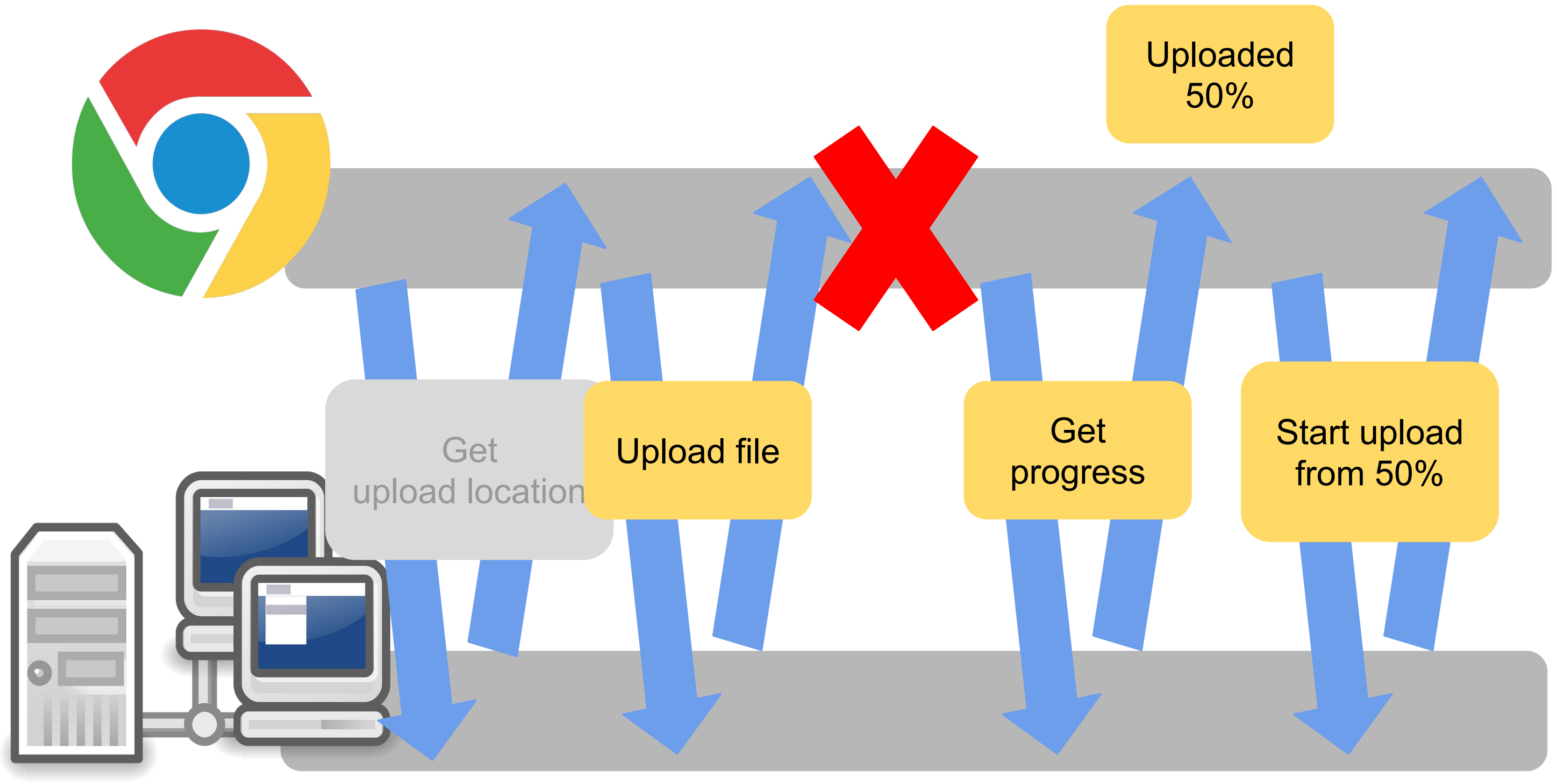
on browser with
Node.js backend

How does it work?

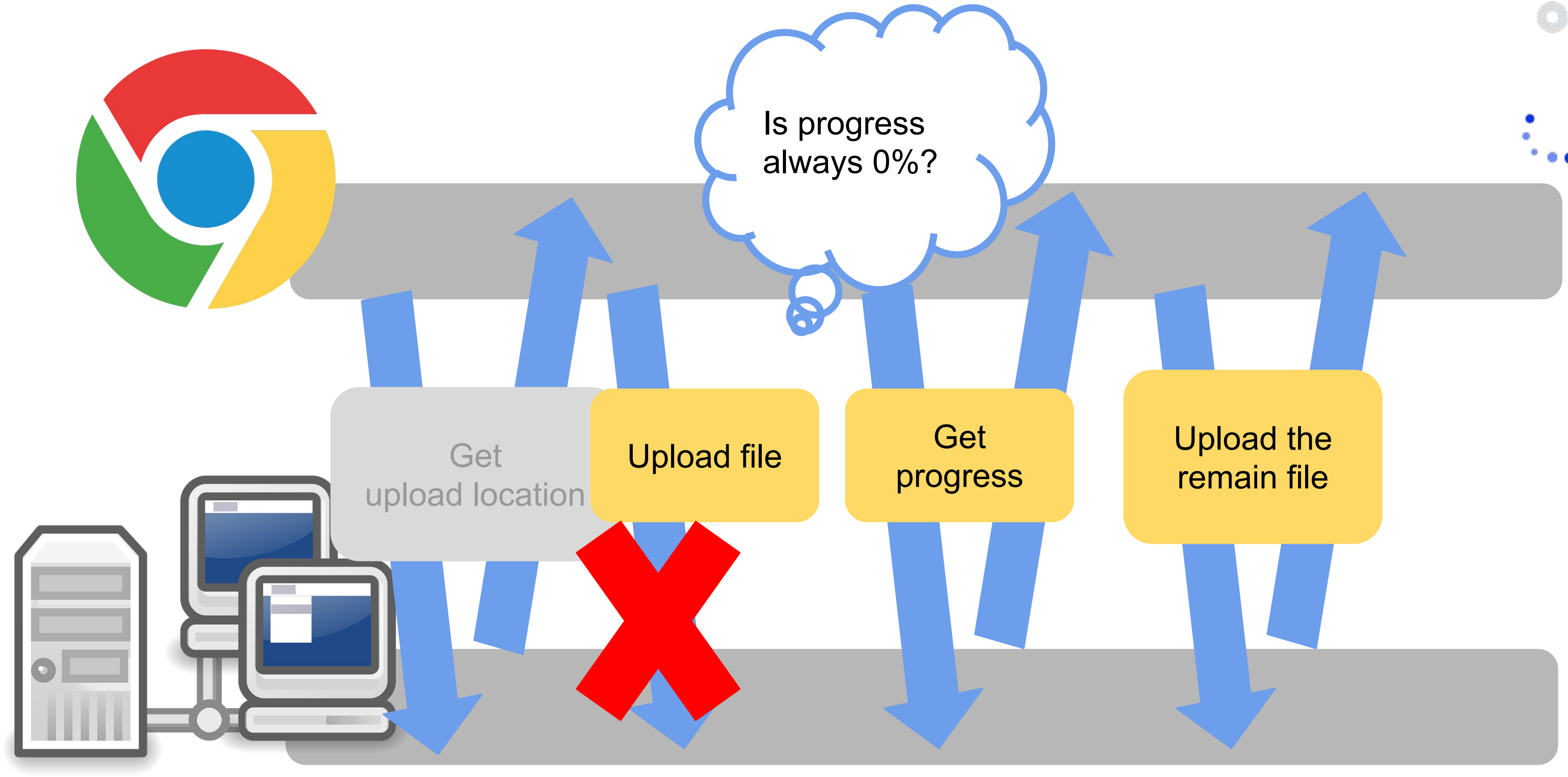
Core protocol - Happy flow



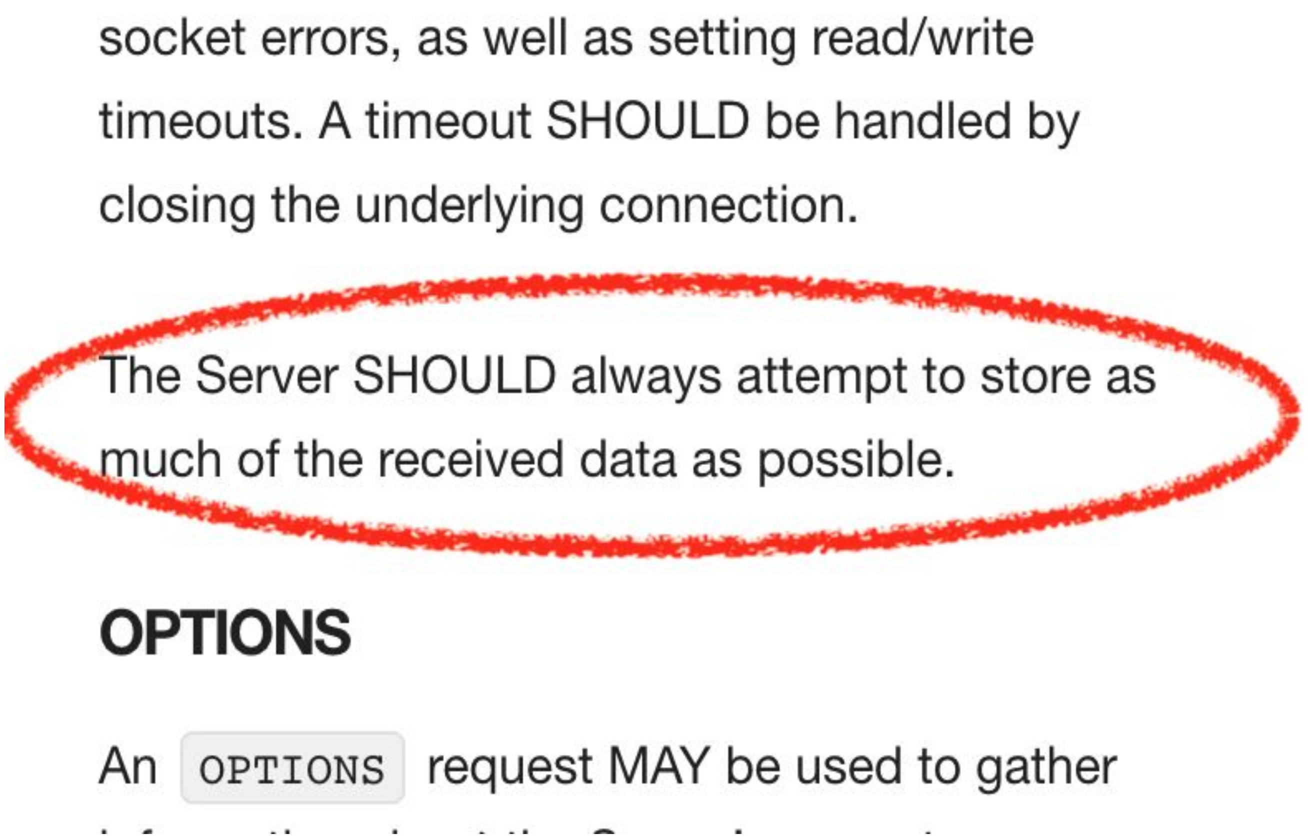
Core protocol - Query upload progress



Core protocol - Resume upload



According to TUS’s specification



socket errors, as well as setting read/write timeouts. A timeout SHOULD be handled by closing the underlying connection.

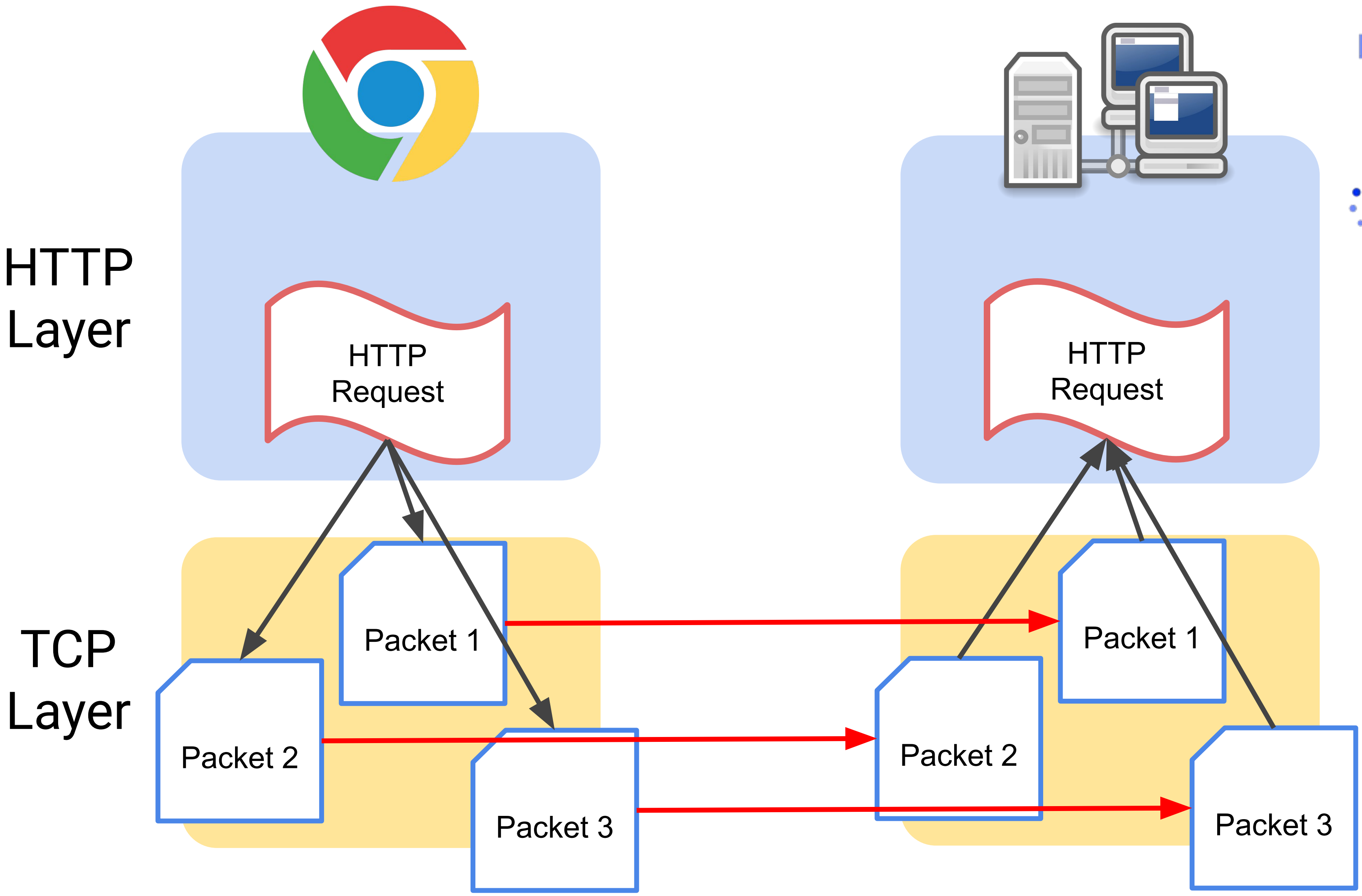
The Server SHOULD always attempt to store as much of the received data as possible.

OPTIONS

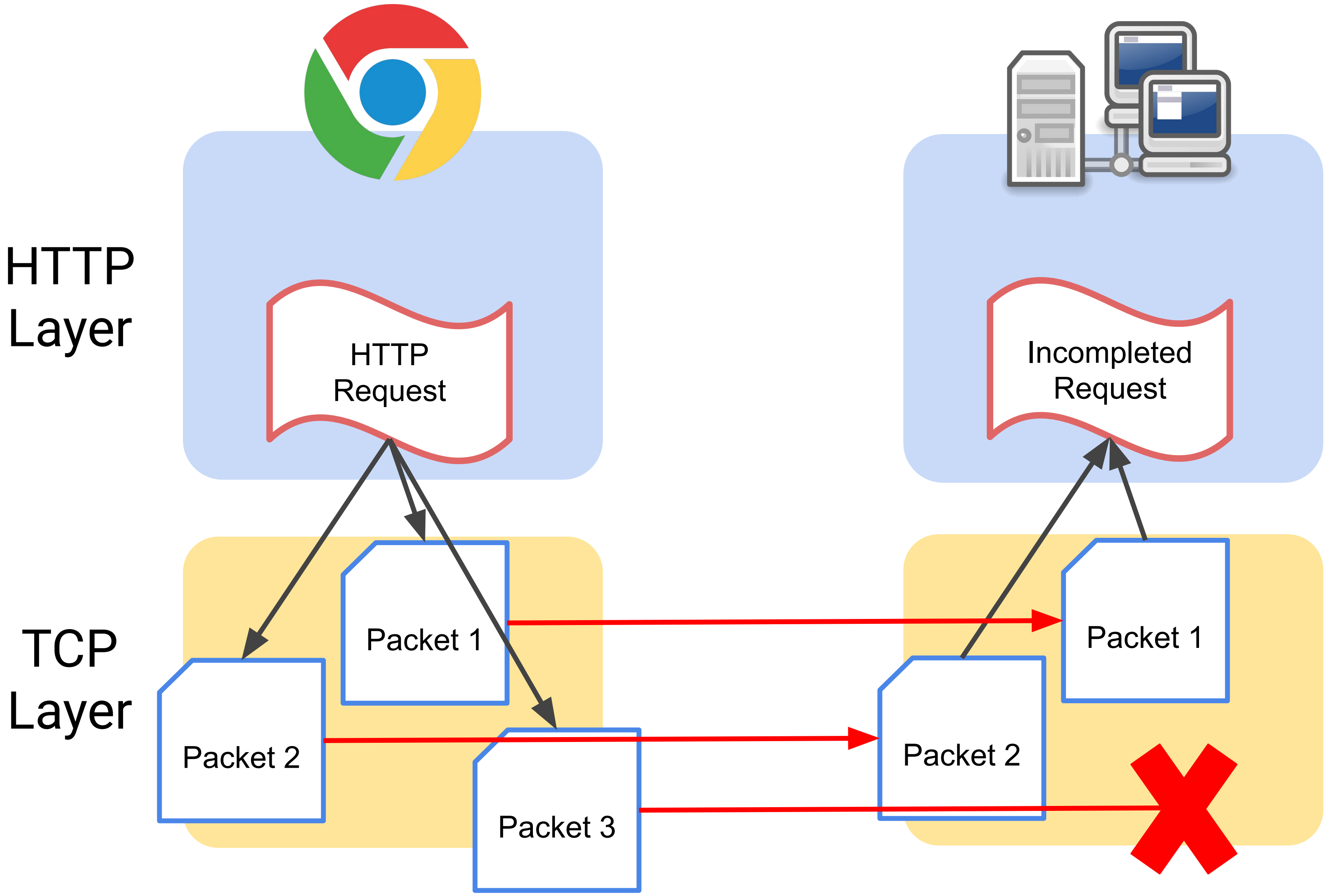
An `OPTIONS` request MAY be used to gather

Let’s dig deep into HTTP!

HTTP is on top of TCP



Server side receive incomplete HTTP request



Node.js implementation demo

Node.js demo code

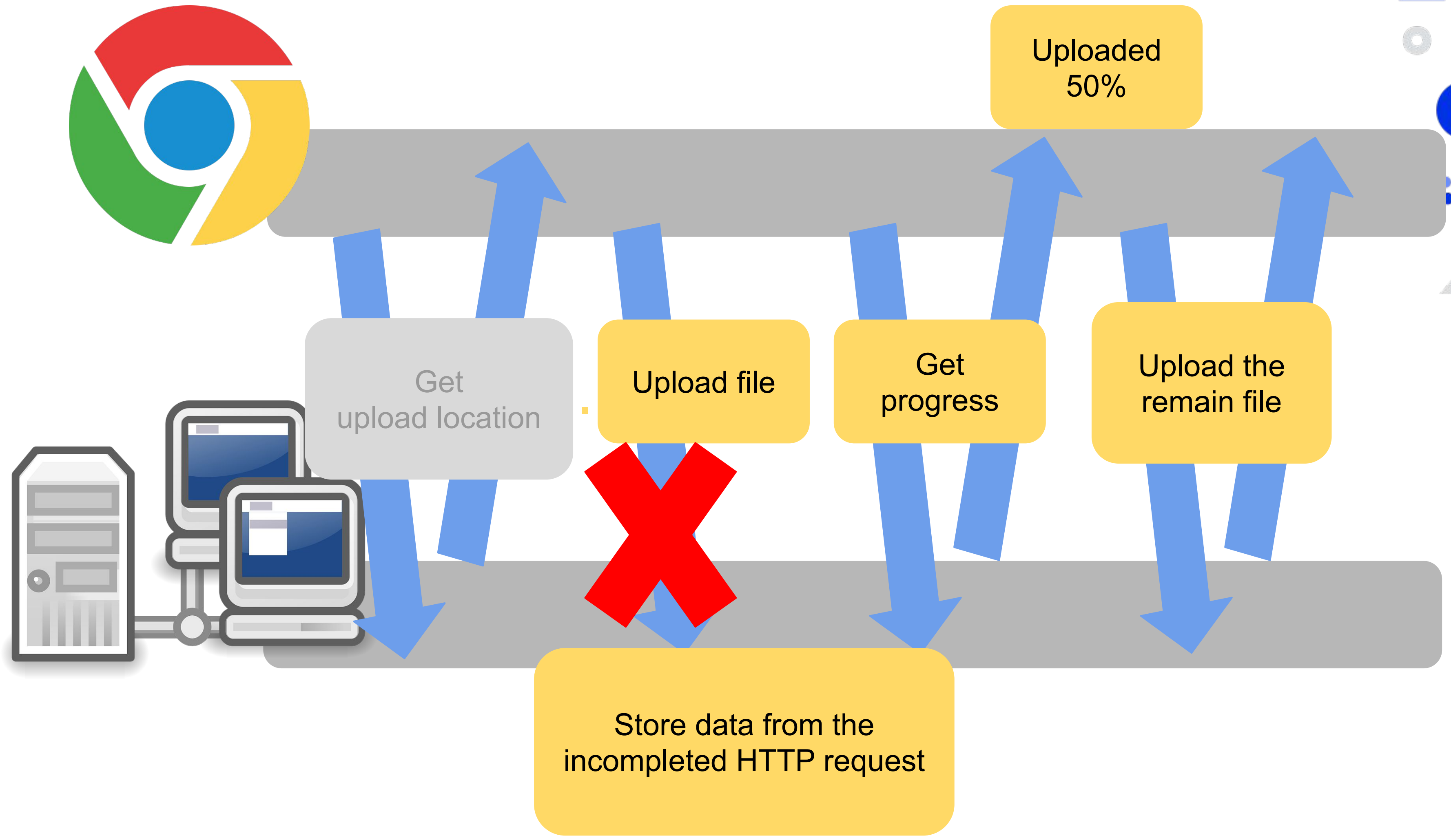
```
request
  .on('data', (chunk) => {
    body = appendToBody(chunk);
    printChunk(chunk);
  })
  .on('end', () => {
    printBody(body);
    sendResponse();
  });
```

TUS implementation pseudo code

```
request
  .on('data', (chunk) => {
    appendChunkToFile(chunk)
  })
  .on('end', () => {
    sendResponse();
  });
```

Summary

Core protocol - Complete flow



Q&A

TUS protocol extentions
(Skipped)

Expiration API
Checksum API
Termination API
Concatenation API