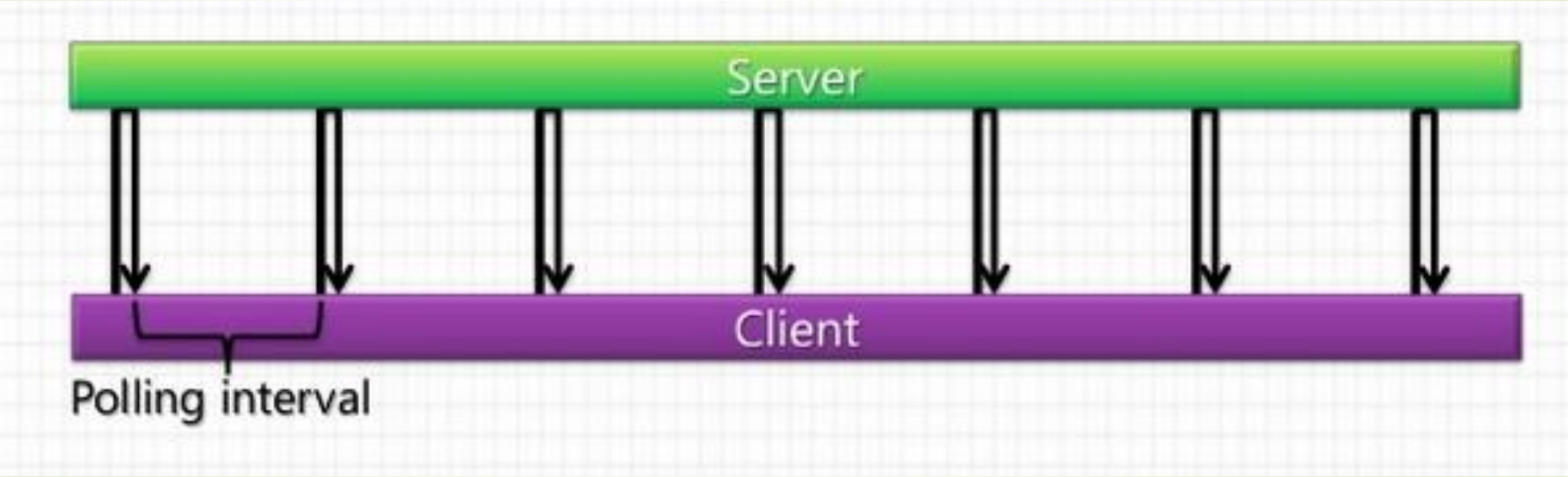
Socket IO

Desarrollo de Aplicaciones Web Avanzado

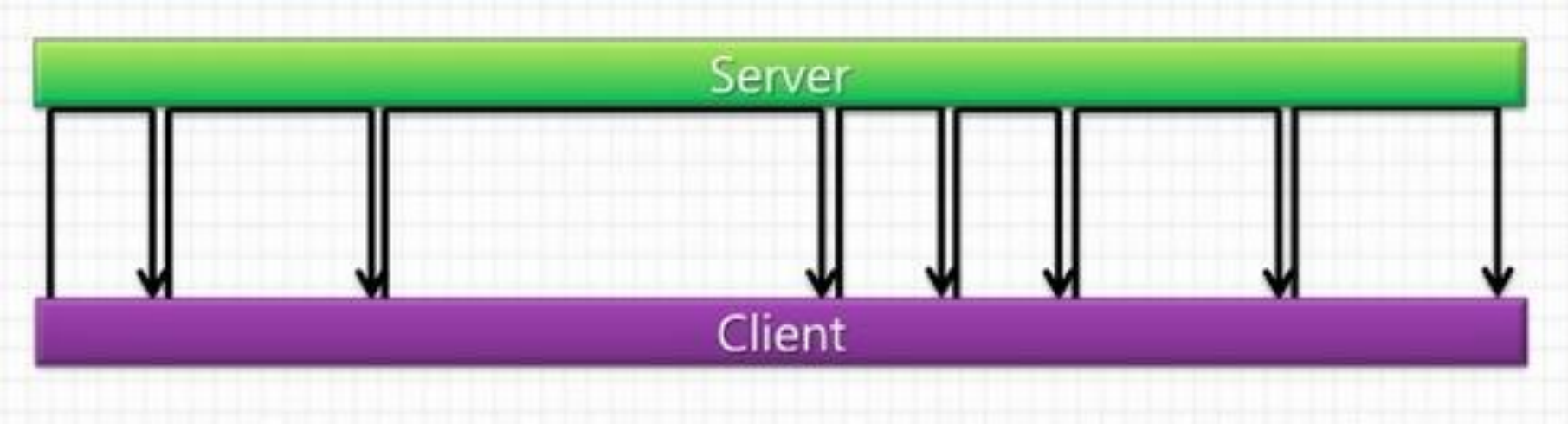
Periodical Polling

- Se realiza una consulta cada cierto tiempo utilizando Ajax
- Demora en comunicaciones debido al intervalo de la consulta
- Se desperdicia ancho de banda y latencia



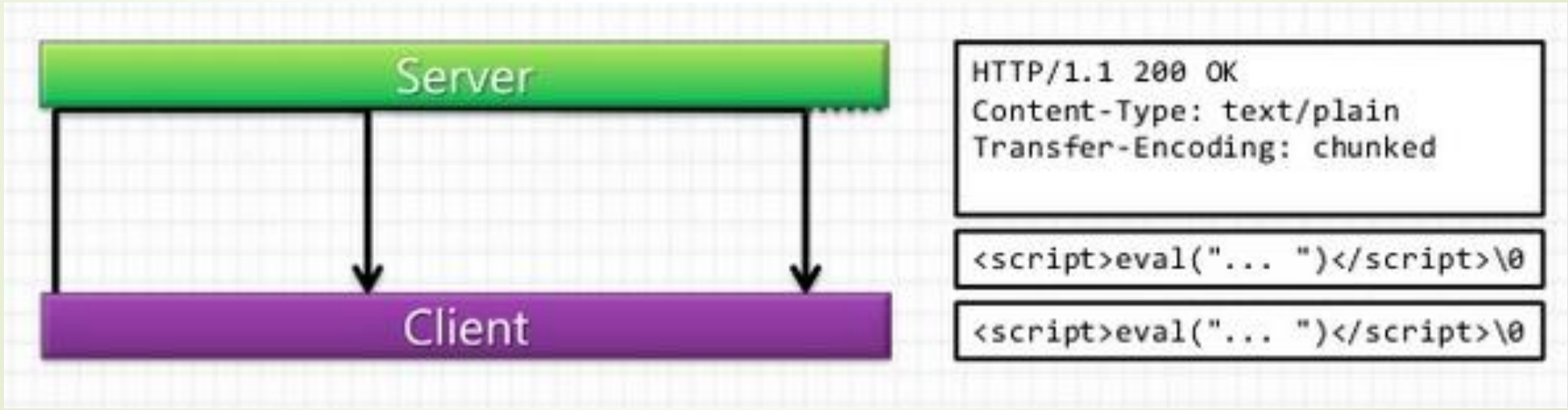
Long Polling

- La consulta no responde hasta que hay información que enviar.
- Se consulta nuevamente solamente después de haber obtenido una respuesta a la consulta anterior.
- Consumo alto de servidor y recursos de conexión



Forever frame

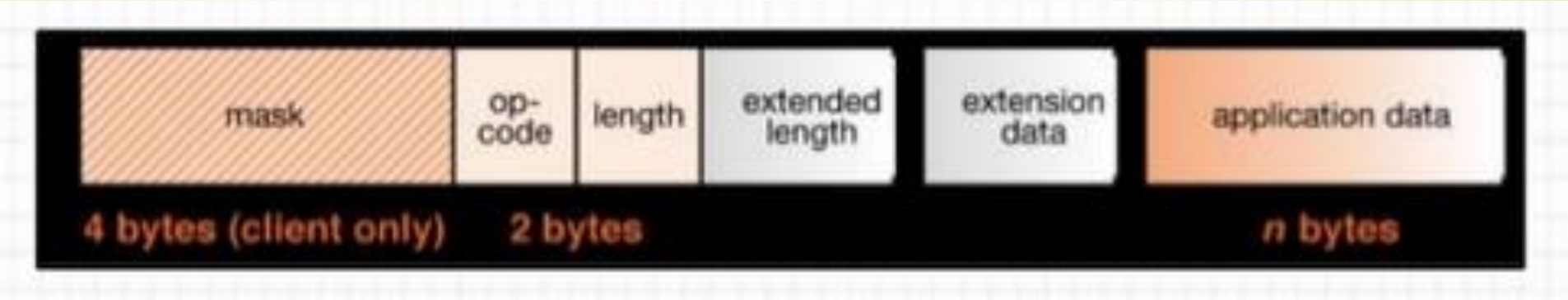
- El servidor le dice al cliente que la respuesta está fragmentada.
- El cliente mantiene la conexión abierta hasta que el servidor la cierra.
- El servidor envía la data cerrando con un \0
- Consume hilos de servidor



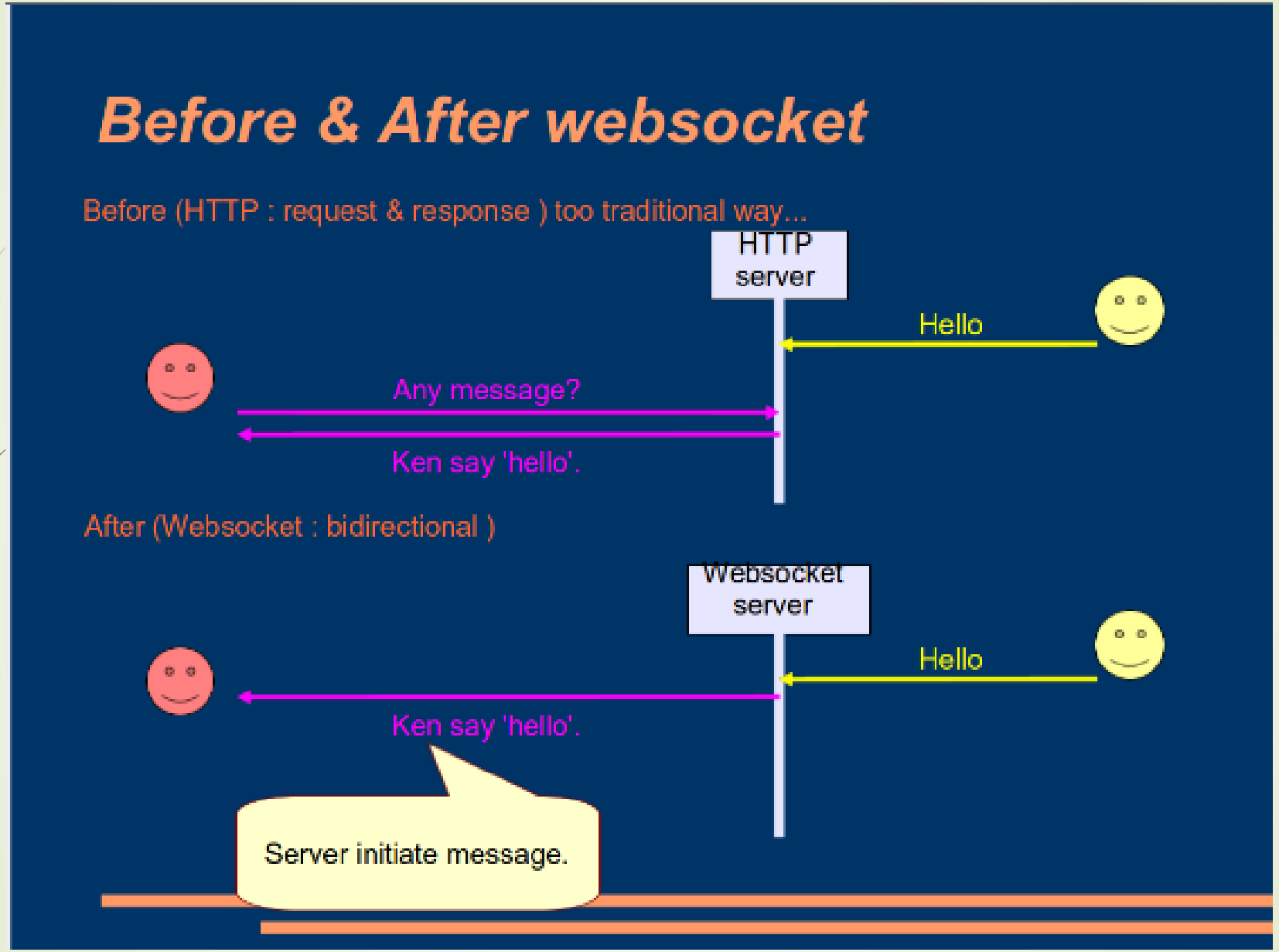
Number of people who actually understands Javascript

¿Qué es web sockets?

- Web Sockets es una tecnología que proporciona un canal de comunicación bidireccional y full-dúplex sobre un único socket TCP
- Para establecer una conexión WebSocket, el cliente envía una solicitud "handshake" y el servidor responde con una respuesta "handshake"



Before & After websocket




PICK ONE
JAVASCRIPT FRAMEWORK
imgflip.com

Socket IO

- Es una librería que nos permite establecer comunicación o conexiones entre uno o varios clientes y el servidor para el desarrollo de aplicaciones de tiempo real o Real Time Apps.
- Funciona igual en cada navegador, dispositivo o plataforma, enfocándose por igual en la confiabilidad de la conexión y su velocidad.

Echo Sample (Server)

```
var http = require("http");
var connect = require("connect");
var socketio = require("socket.io");
var app = connect();

app.use(connect.static("public"));
var server = http.createServer(app);

var io = socketio.listen(server);
io.on("connection", function (socket) {
    socket.on("message", function (data) {
        socket.emit("echo", data);
    });
});

server.listen(8000);
```

Echo Sample (client)

```
<!DOCTYPE html>
<html>
<head>
    <script src="/socket.io/socket.io.js"></script>
</head>
<body>
    <body>
        <script>
            var socket = io.connect("http://localhost");
            socket.emit("message", "Hello!");
            socket.on("echo", function(data) {
            document.write(data);
            });
        </script>
    </body>
</html>
```

Broadcasting

- Para poder enviar un evento a todos los clientes conectados, Socket IO nos otorga la función io.emit:

```
io.emit('some event', { for: 'everyone' });

socket.broadcast.emit('some event');
```

Namespaces

- Socket IO permite otorgar namespaces a tus sockets, lo que esencialmente significa que se le pueden otorgar distintos endpoints o caminos.
- El namespace por defecto es / y es al que un cliente Socket.IO se conecta por defecto y en el que el servidor siempre escucha por defecto.

```
// Server Side
var nsp = io.of('/my-namespace');
nsp.on('connection', function(socket) {
    console.log('someone connected');
});
nsp.emit('hi', 'everyone!');
```

```
// Client Side
var socket = io('/my-namespace');
```

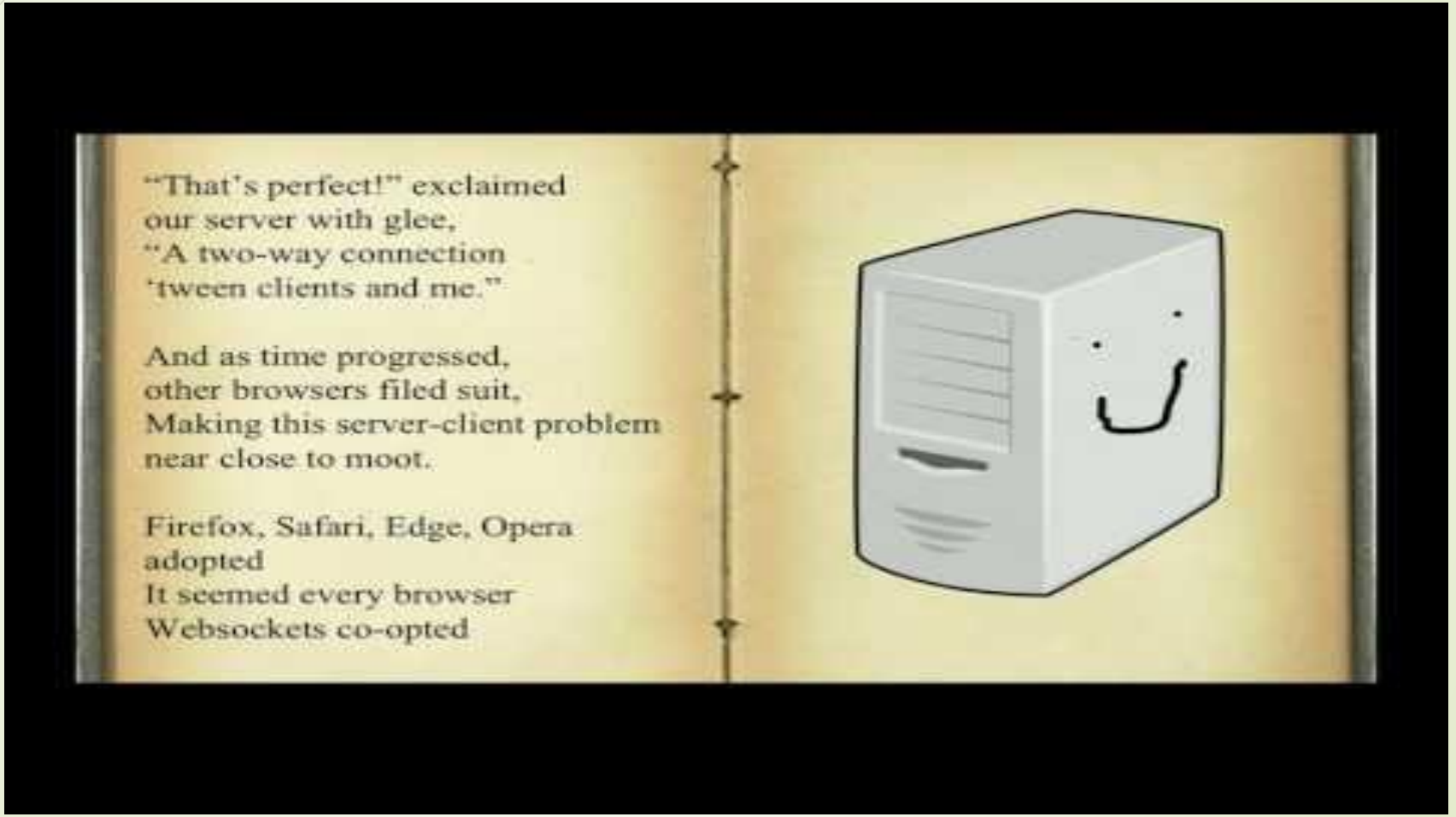
Rooms

- Dentro de cada namespace, puedes definir arbitrariamente canales a los que los sockets pueden unirse o abandonar.

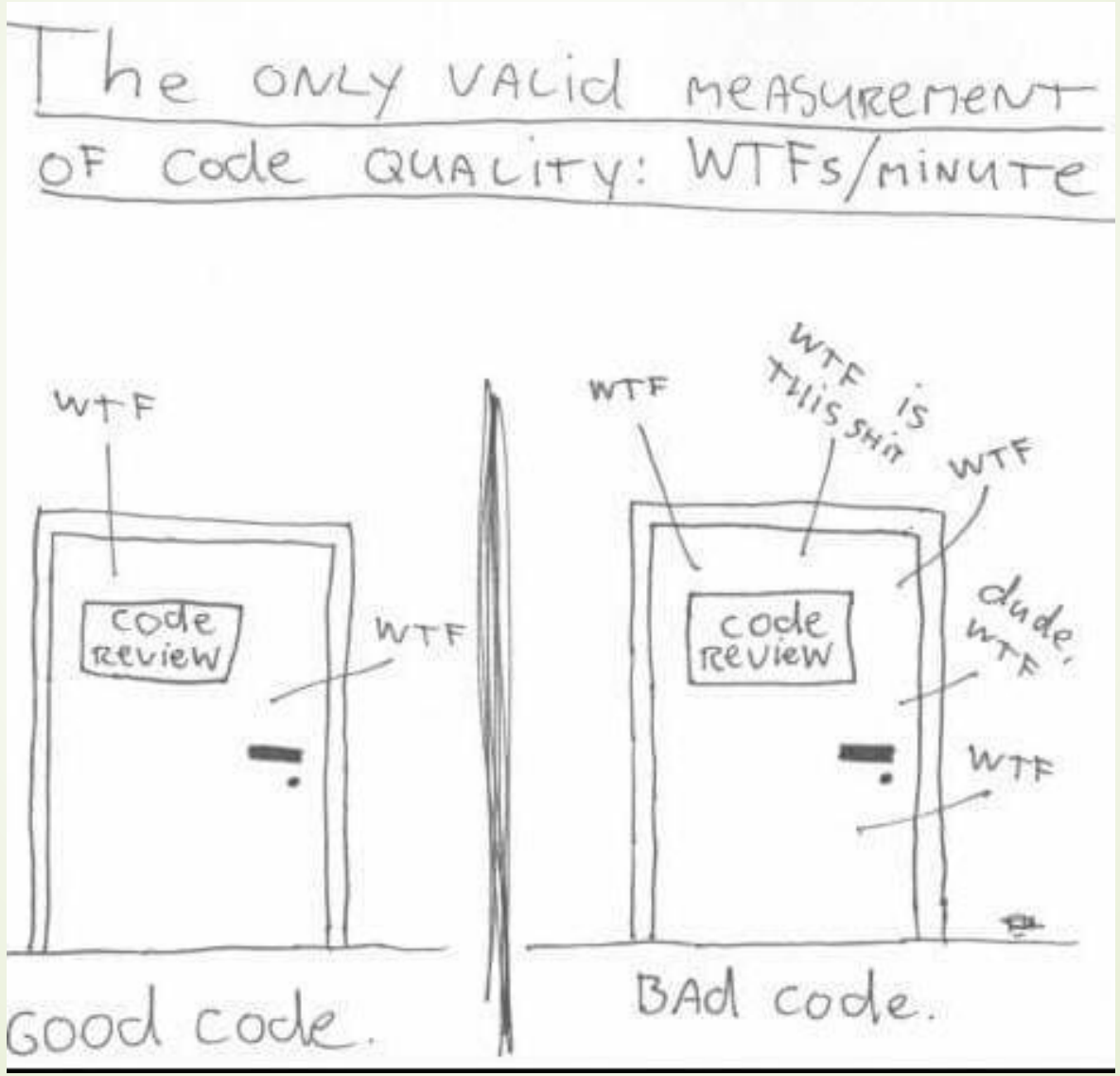
```
io.on('connection', function (socket) {
    socket.join('some room');
});

io.to('some room').emit('some event');
```

Como Sockets IO salvó la navidad



¡Gracias!


The ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE
WTF
code review
WTF
Good code.
WTF
WTF is THIS SHIT
WTF
code review
dude, WTF
WTF
BAD code.