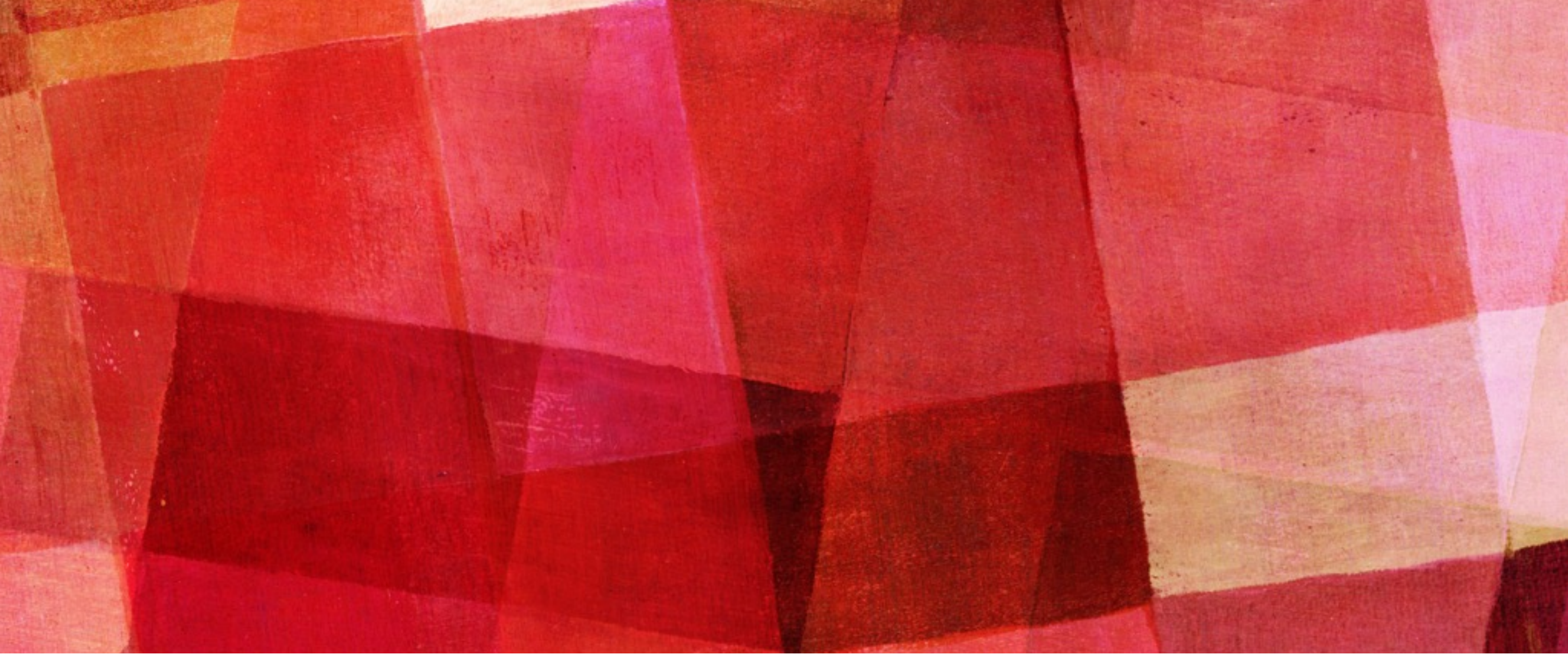
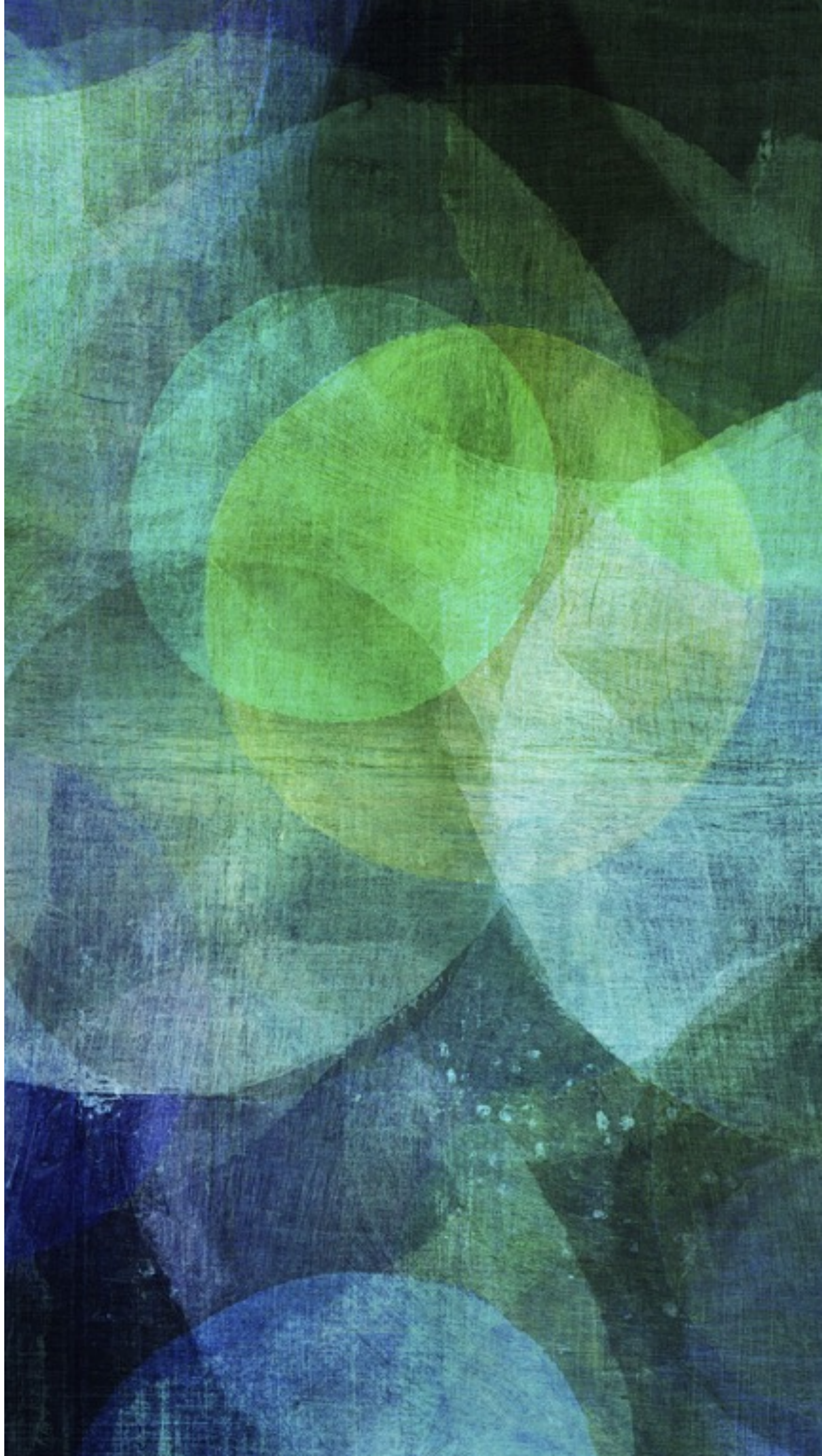


FLASK-SOCKET.IO

Web Sockets Made Simple

WEB SOCKETS!

OK. BUT, WHY?



FIRST THERE WAS THE INTERNET

AND THEN THERE WAS TCP/IP

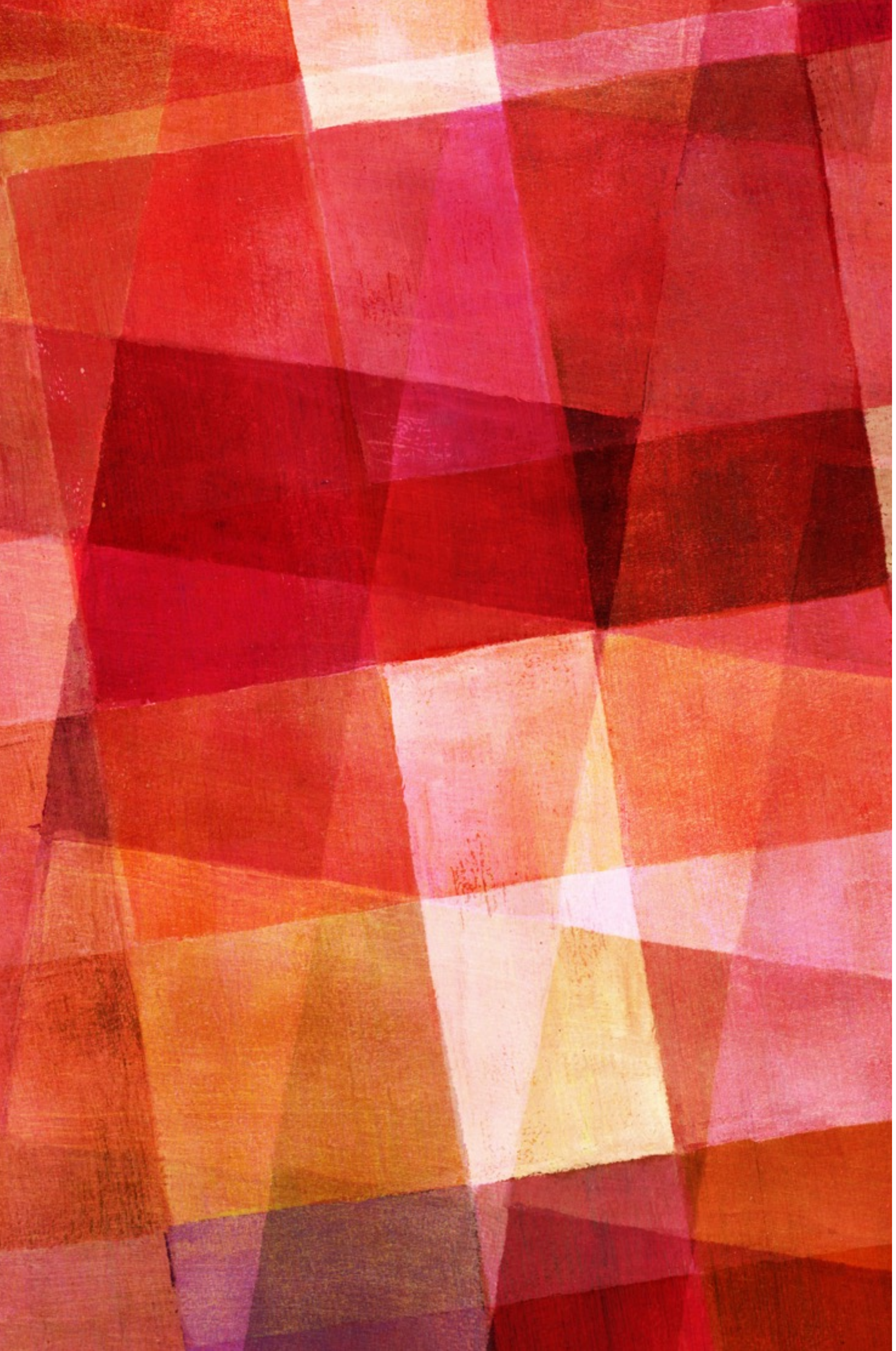
TCP/IP SOCKETS

```python
import socket

sock = socket.create_connection(('24.244.4.54', 80), timeout=30)
try:
    # Send the request
    sock.sendall("Really important stuff")

    # Get the response
    response = sock.recv(1024)  # Bytes
finally:
    sock.close()
```

TCP/IP SOCKETS

- Real time
- Reliable delivery
- No standards for delivery
  - … just bits on the wire

- Requires persistent connection



BUT…

I HAVE A 90 MHZ PROCESSOR AND 16 MB OF RAM

Microsoft
Windows 95
Microsoft Plus!
SAMSUNG

“

From 1973 to 1974, Cerf's networking research group at Stanford worked out details of the idea, resulting in the first TCP specification

- 4 74181 ALUs (~45 MHz)
- 128-512 KB of RAM
- 2.5 MB Single Platter Storage Cartridge

HTTP

HTTP

- Request / Response
  - Disconnect TCP socket after response
- Meta-data
  - Information about the payload
  - Date/time sent
  - Caching
  - etc…
- Lighter on server resources

- The Web!

HTTP Request

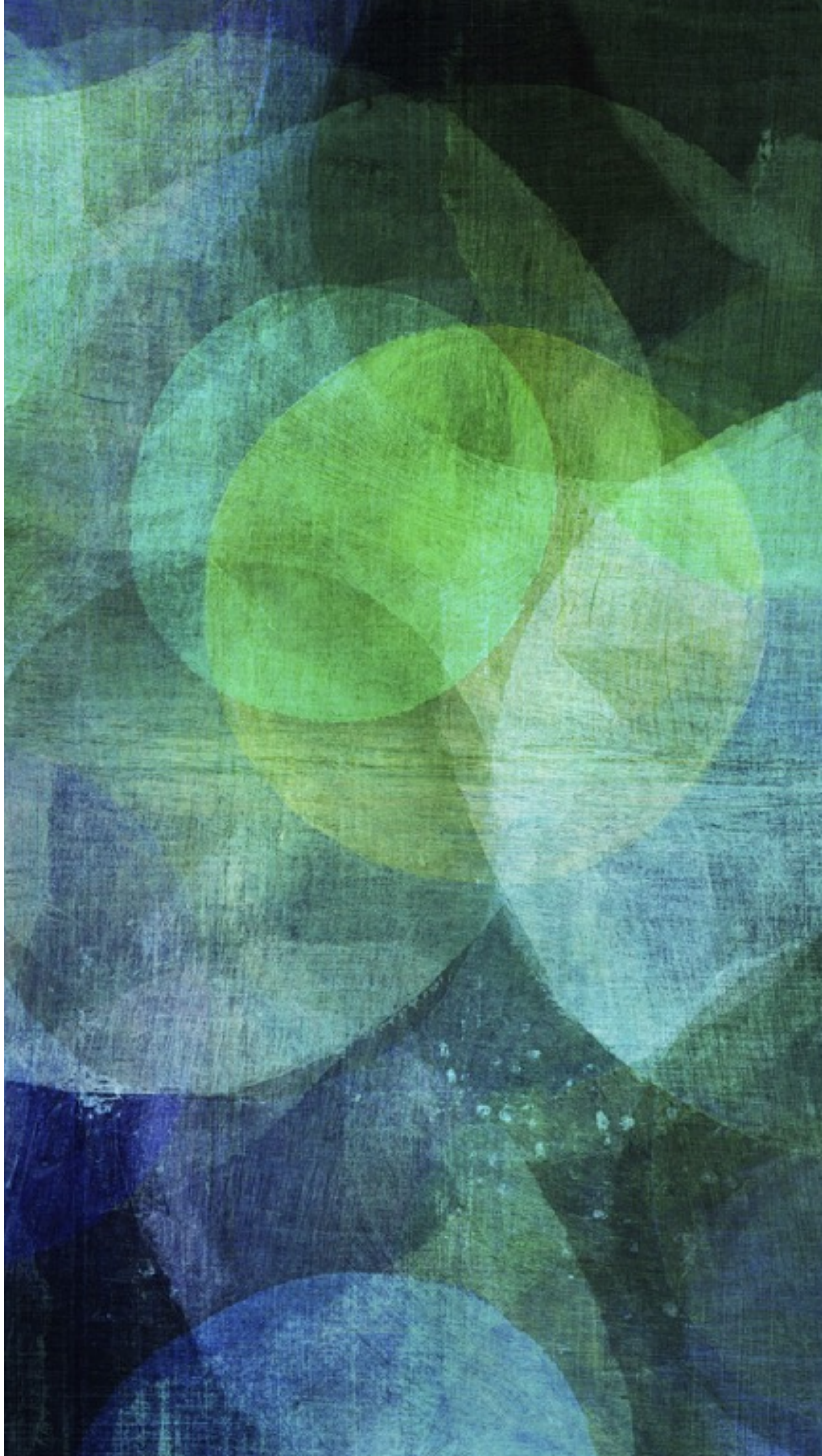
http://www.example.com/example?key=value&something=other

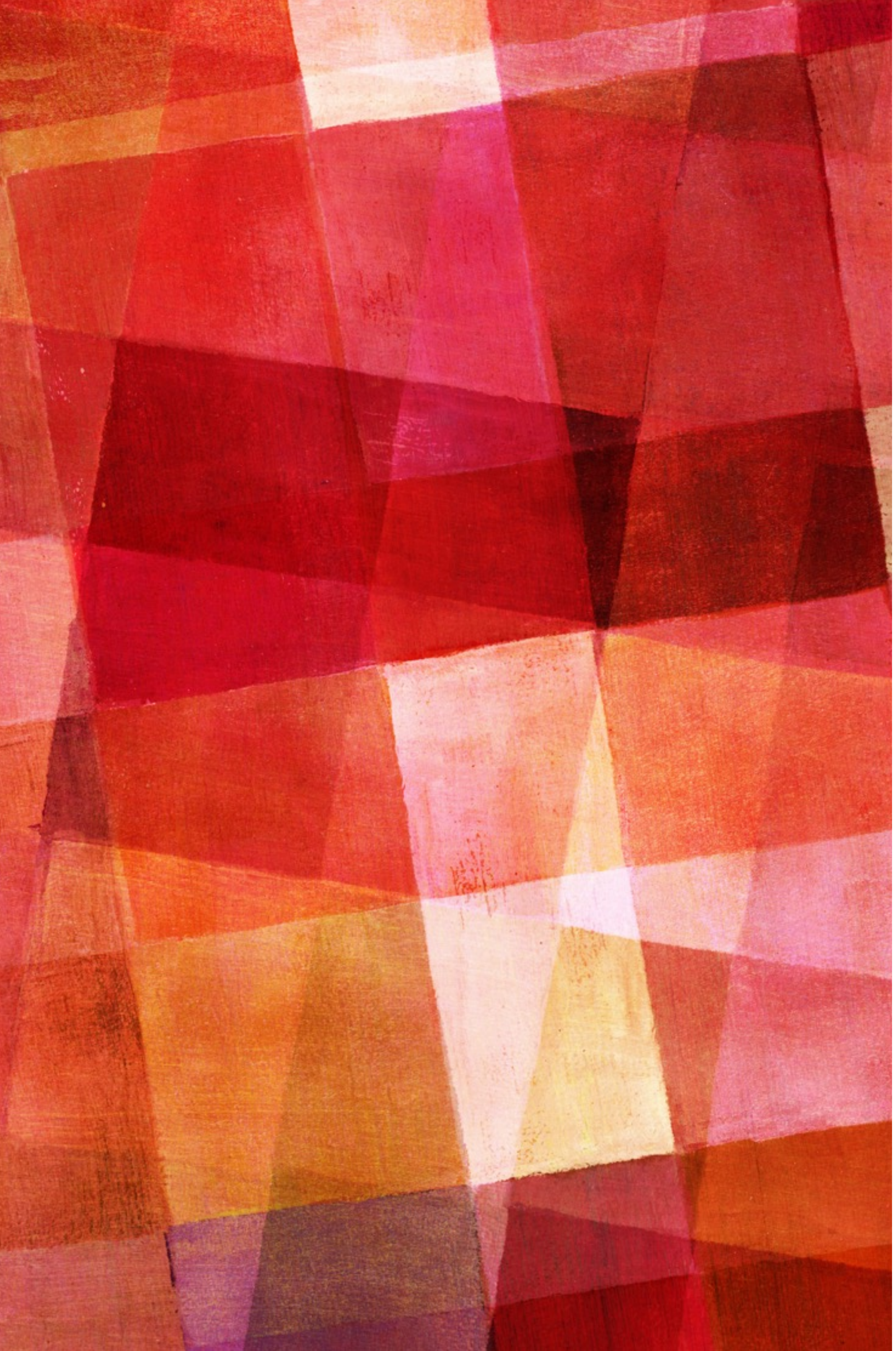
```
POST /example?key=value&something=other HTTP 1.1

Host: www.example.com

Accept: application/json, application/xml

Accept-Language: EN-US

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)

Connection: Keep-Alive


{ "Content": "Adding my content", "Other": [2, 4, 6, 8] }
```

HTTP (W/ REQUESTS)

```python
import requests

response = requests.get('http://www.google.ca')
```

UH,
SO…
SECURITY?





WHAT ABOUT IP ADDRESSES?

BRAVE NEW WORLD (OF NETWORKS)

- Network Address Translation (NAT)
- Closed ports (pretty much web only)
- Security first
- ….

HTTP 1.1

WEB SOCKETS!

WEB SOCKETS

- Upgrade from a standard HTTP request
- Can navigate the modern NAT
- Can be authenticated
- More secure

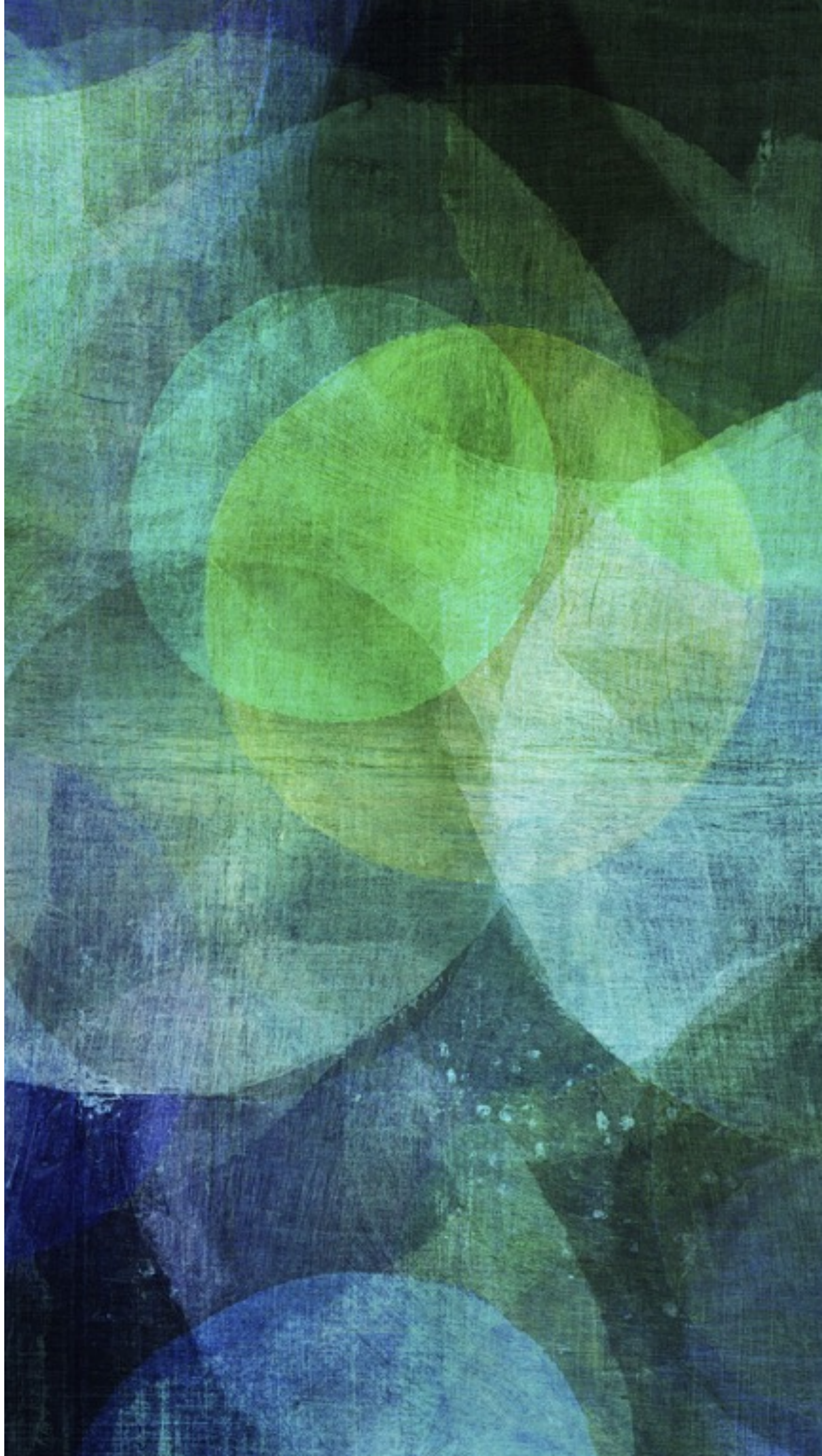
```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
Origin: http://example.com
```

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: HSmrc0sMlYUkAGmm5OPpG2HaGWk=
Sec-WebSocket-Protocol: chat
```

FLASK-SOCKET.IO

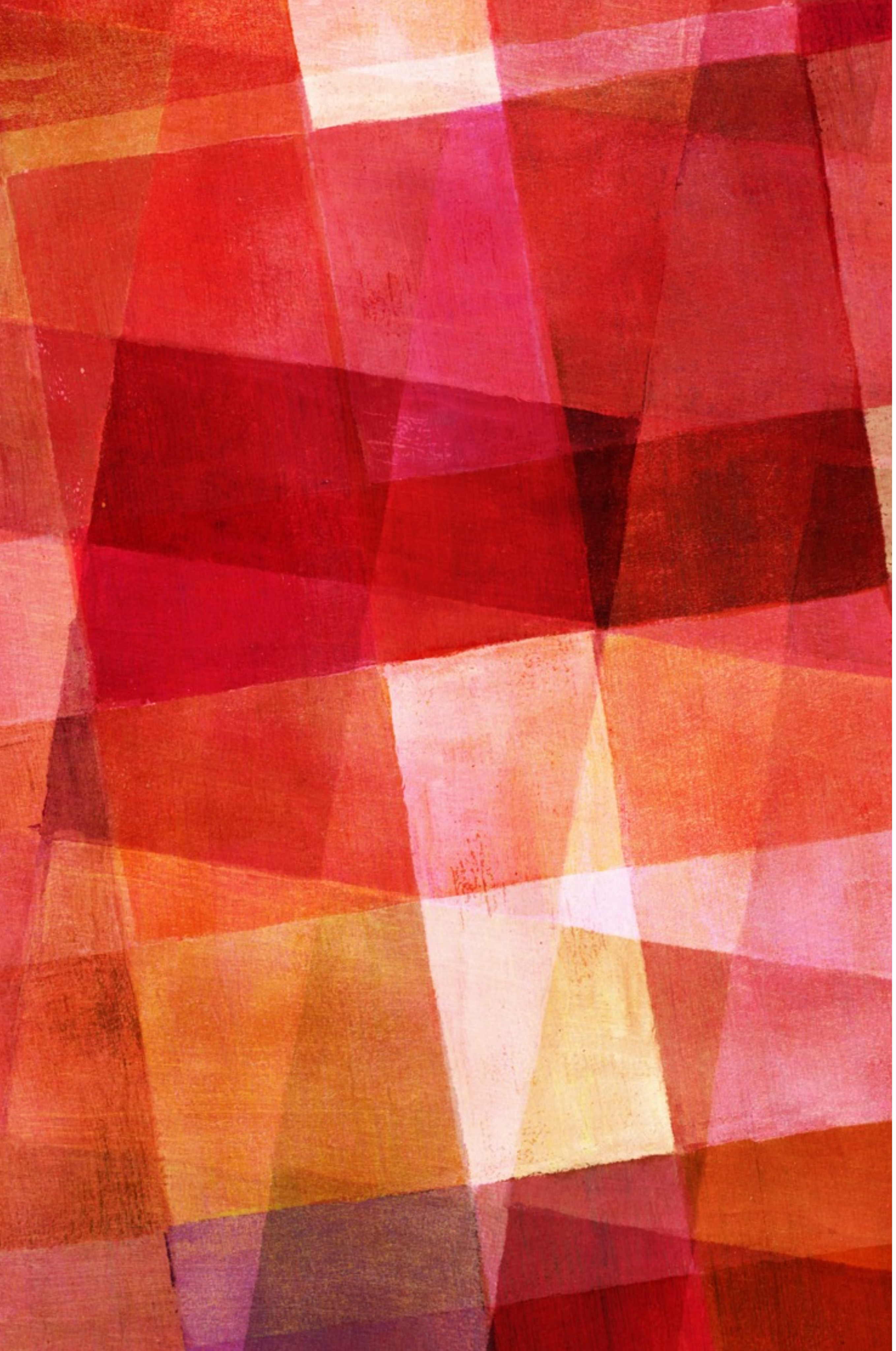
Web Sockets Made Simple

OH, BUT FIRST LETS CHECK OUT FLASK



FLASK

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```



AND SOCKET.IO TOO

SOCKET.IO CLIENT

```
<script src="/socket.io/socket.io.js"></script>
<script>
  var socket = io('http://localhost');
  socket.on('news', function (data) {
    console.log(data);
    socket.emit('my other event', { my: 'data' });
  });
</script>
```

SOCKET.IO

- Channels
- Namespaces
- Rooms (server-side)

FLASK-SOCKET.IO

Web Sockets Made Simple

FLASK-SOCKET.IO

```python
from flask import Flask
from flask_socketio import SocketIO

app = Flask(__name__)
app.config['SECRET_KEY'] = 'secret!'
socketio = SocketIO(app)

@socketio.on('channel')
def handle_message(message):
    print('received message: ' + message)

if __name__ == '__main__':
    socketio.run(app)
```

DEMO