SaaS Architecture

CPEN322

Assignment 0

- Submit your Github username
- Make your Group
- See Canvas

Patterns in Software Development

Software Design Pattern: a general architectural solution to a family of similar problems

Software design concentrates on the system's implementation, often delving into considerable detail.

Architecture in Software Development

Software Architecture Pattern: how subsystems are connected together to meet the application's functional and non-functional requirements.
AKA: Architectural **Style**

Software architecture shows the system's structure and hides the implementation details, focusing on how the system components interact with one another.

Local file

Alice's PC

File

How can Alice share this file?

Alice's PC

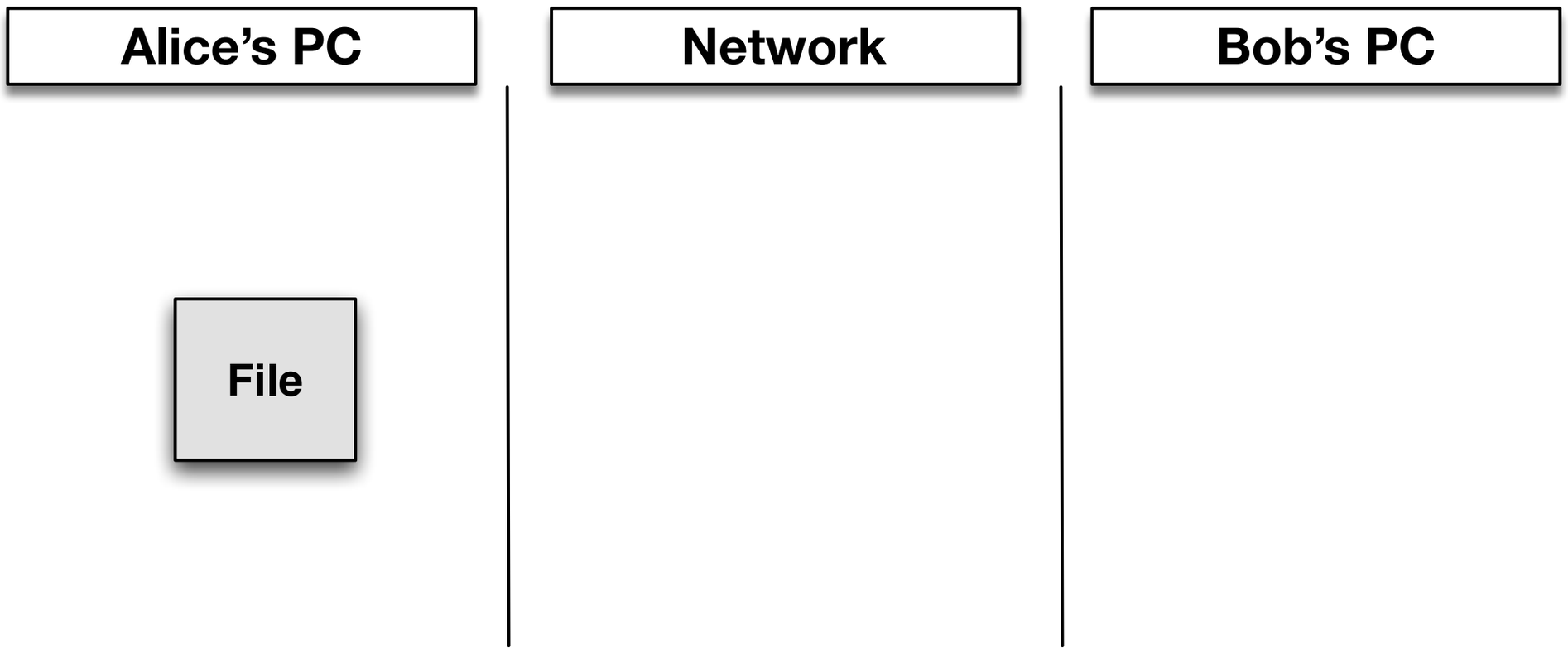
Bob's PC

File

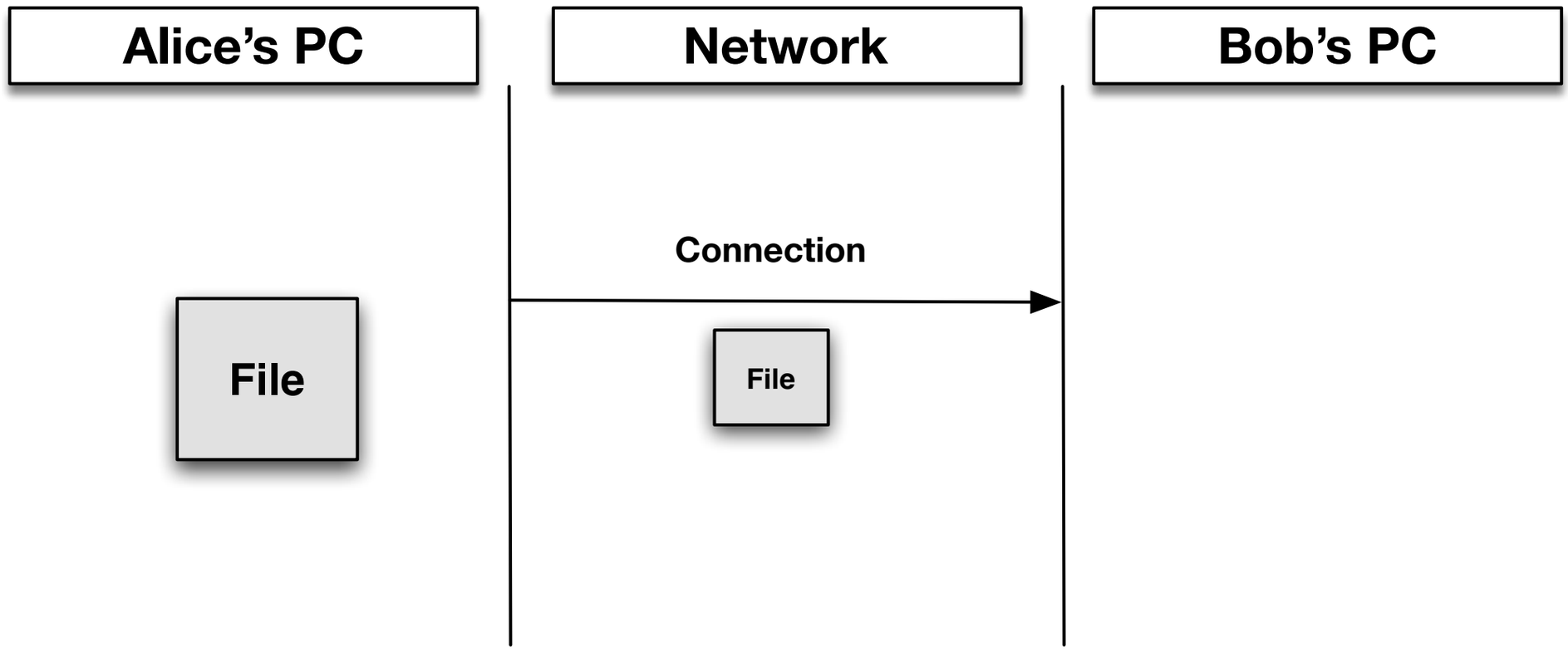
How can Alice share this file?

- What architectural style can be used?
- How?

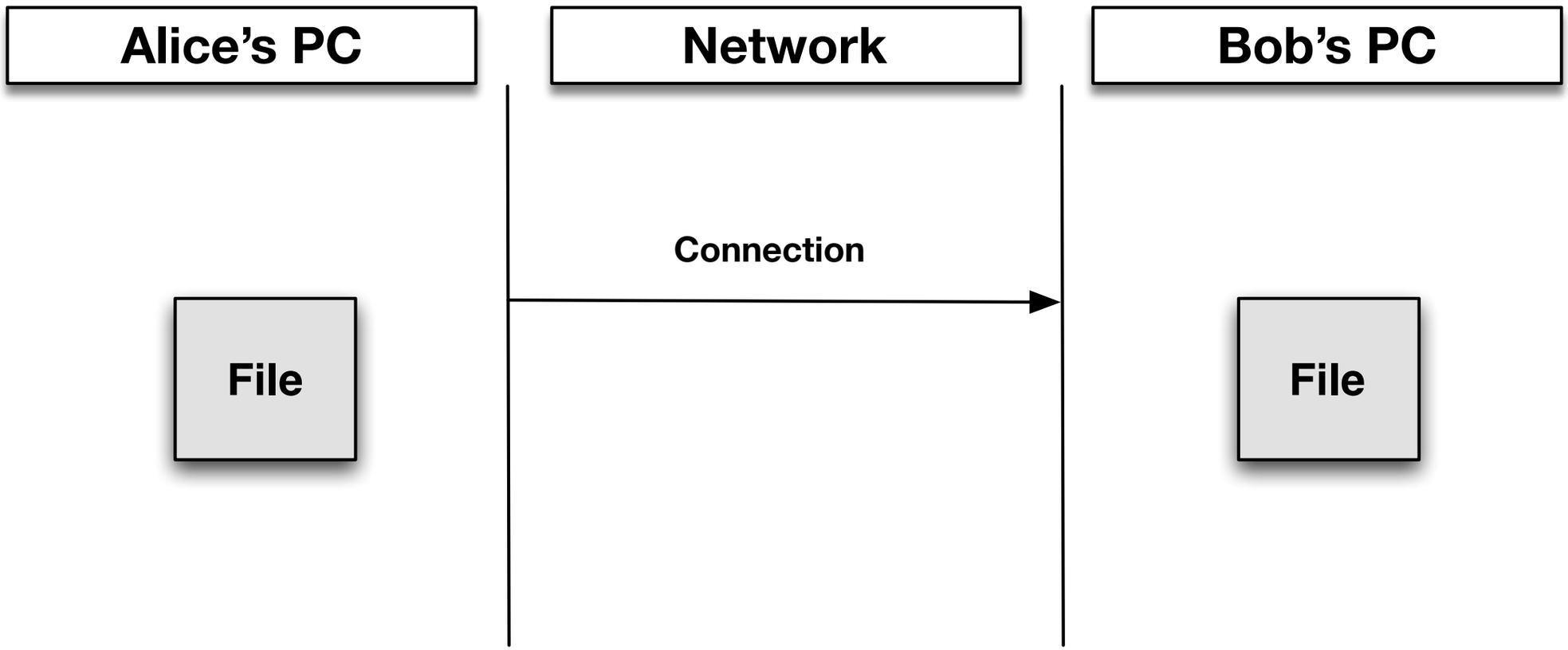
Over a network



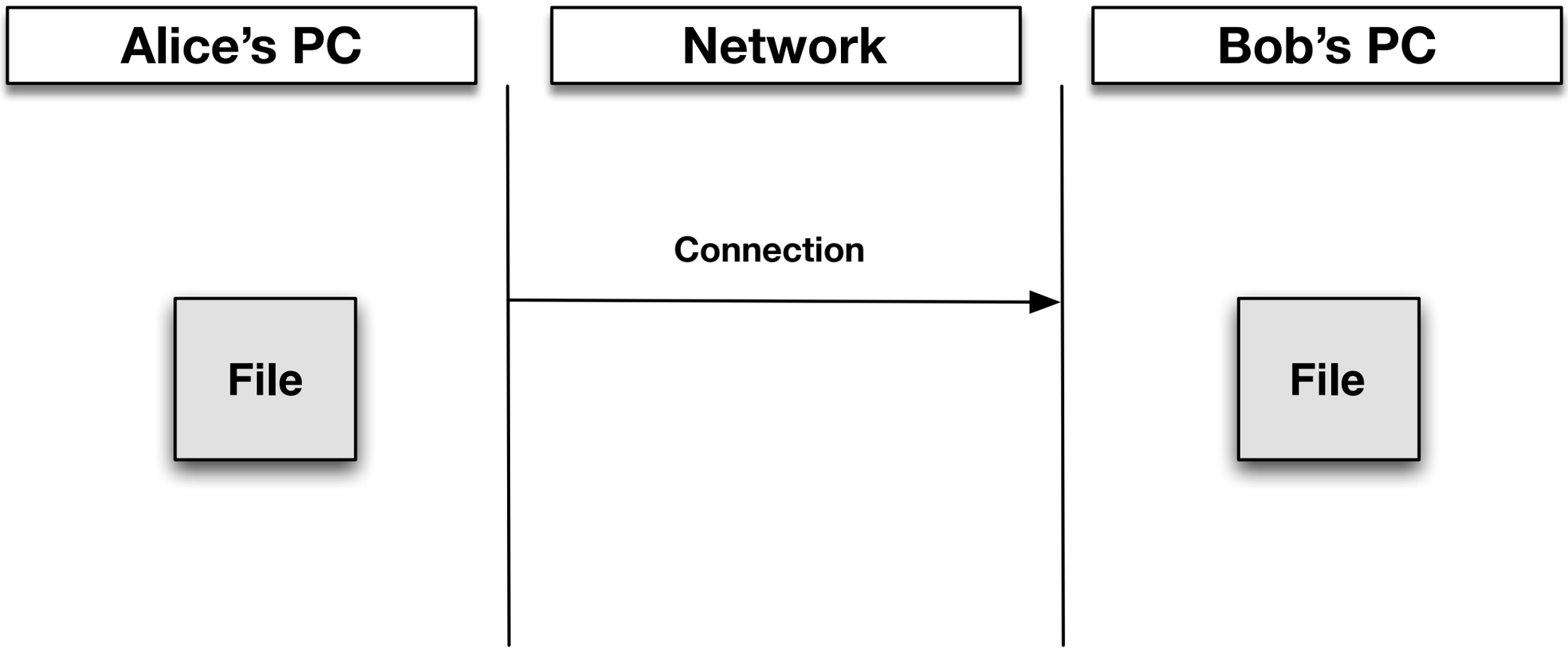
Over a network connection



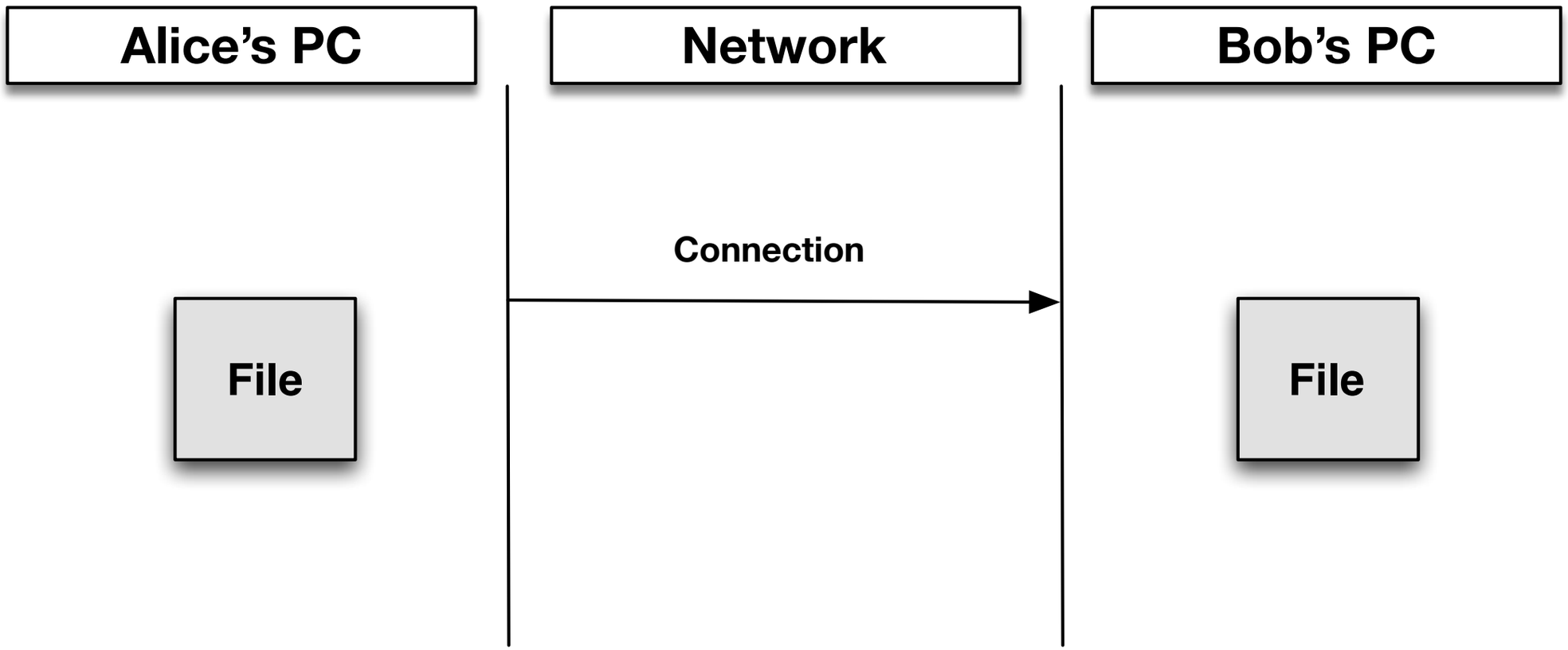
Over a network connection



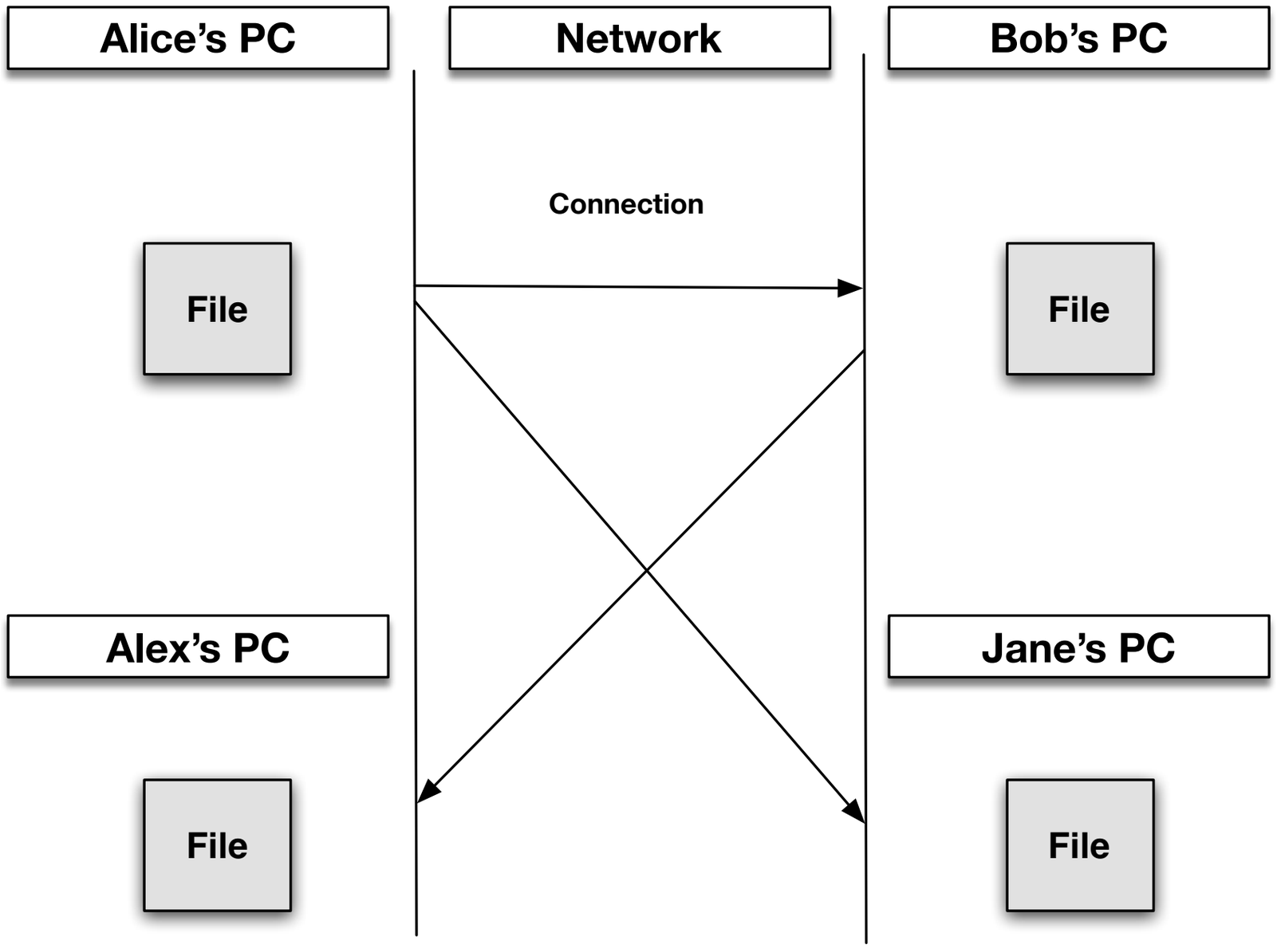
What is this architectural style called?



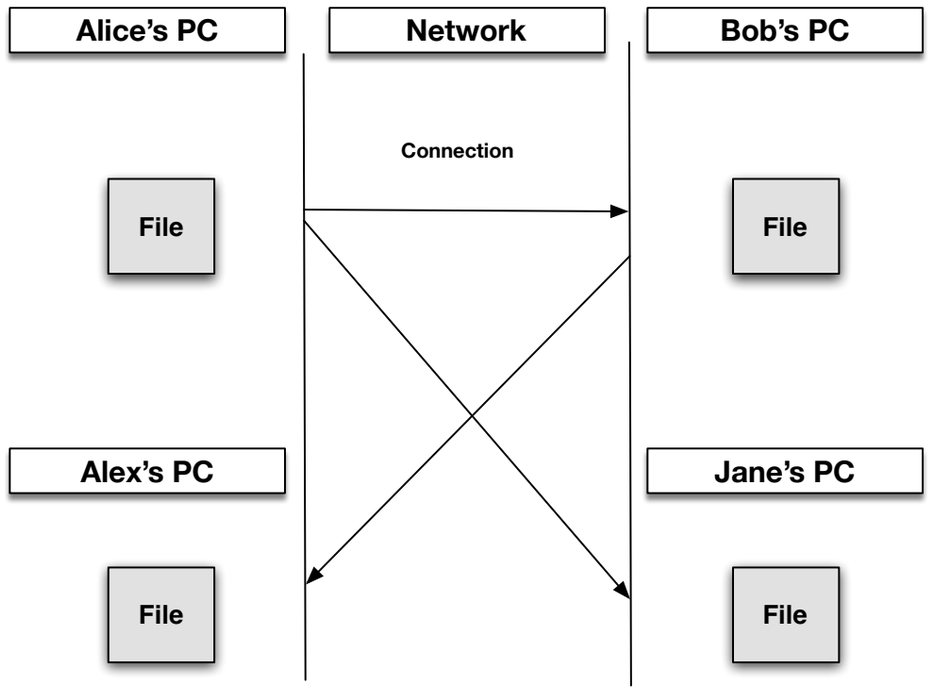
Peer-to-peer



Peer-to-peer


Alice's PC
Network
Bob's PC
Connection
File
File
Alex's PC
Jane's PC
File
File

Pros and cons of peer-to-peer?



Pros of peer-to-peer?

Pros:

1.Decentralization: every node is both a client and a server, which can enhance reliability. If one node fails, the network can still function.

2.Scalability: scale up with the addition of new nodes, as each new node adds additional resources.

3.Cost-Effectiveness: Since resources are shared among peers, there’s no need for expensive central servers.

4.Resource Sharing: Efficient for sharing resources like files directly between nodes.

Cons of peer-to-peer

Cons:

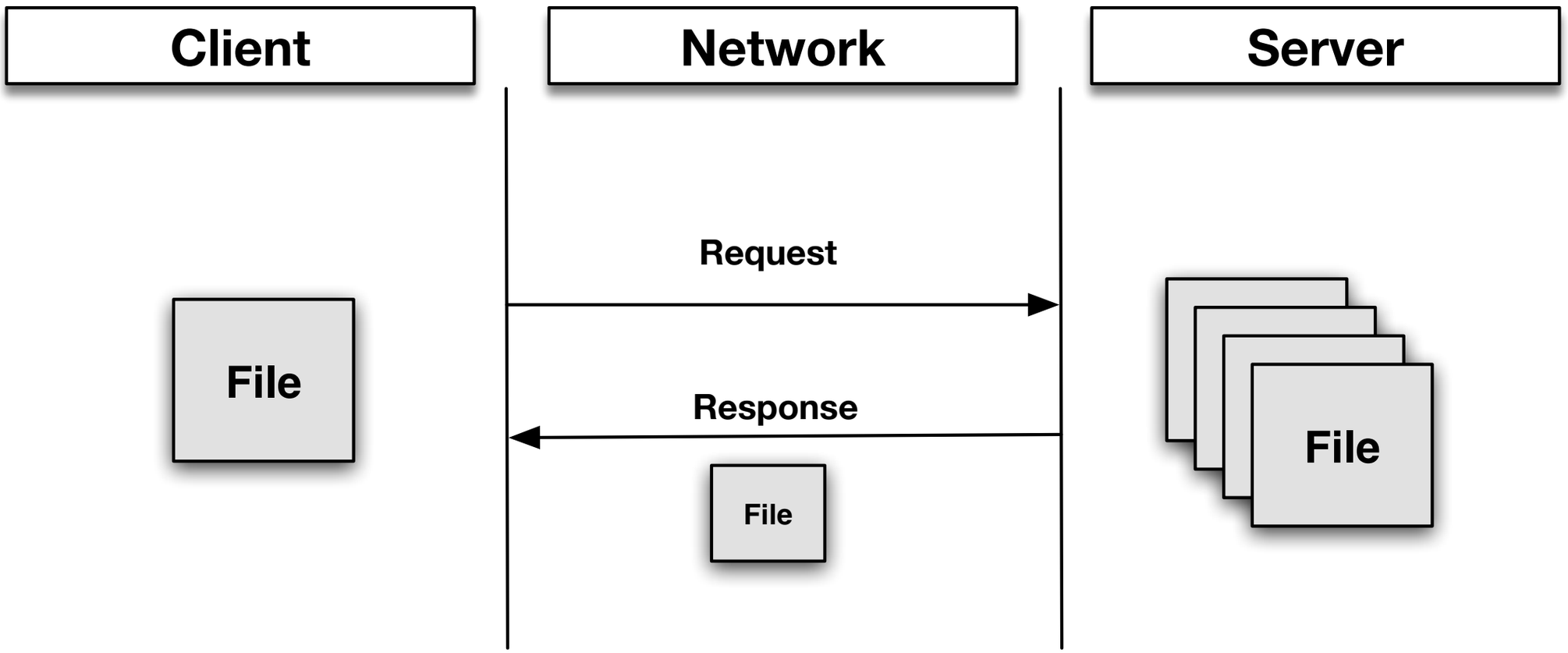
1.Security Risks: vulnerable to security threats since each node is potentially exposed to other nodes.

2.Data Integrity and Quality: There is no centralized authority to ensure the integrity and quality of the data shared.

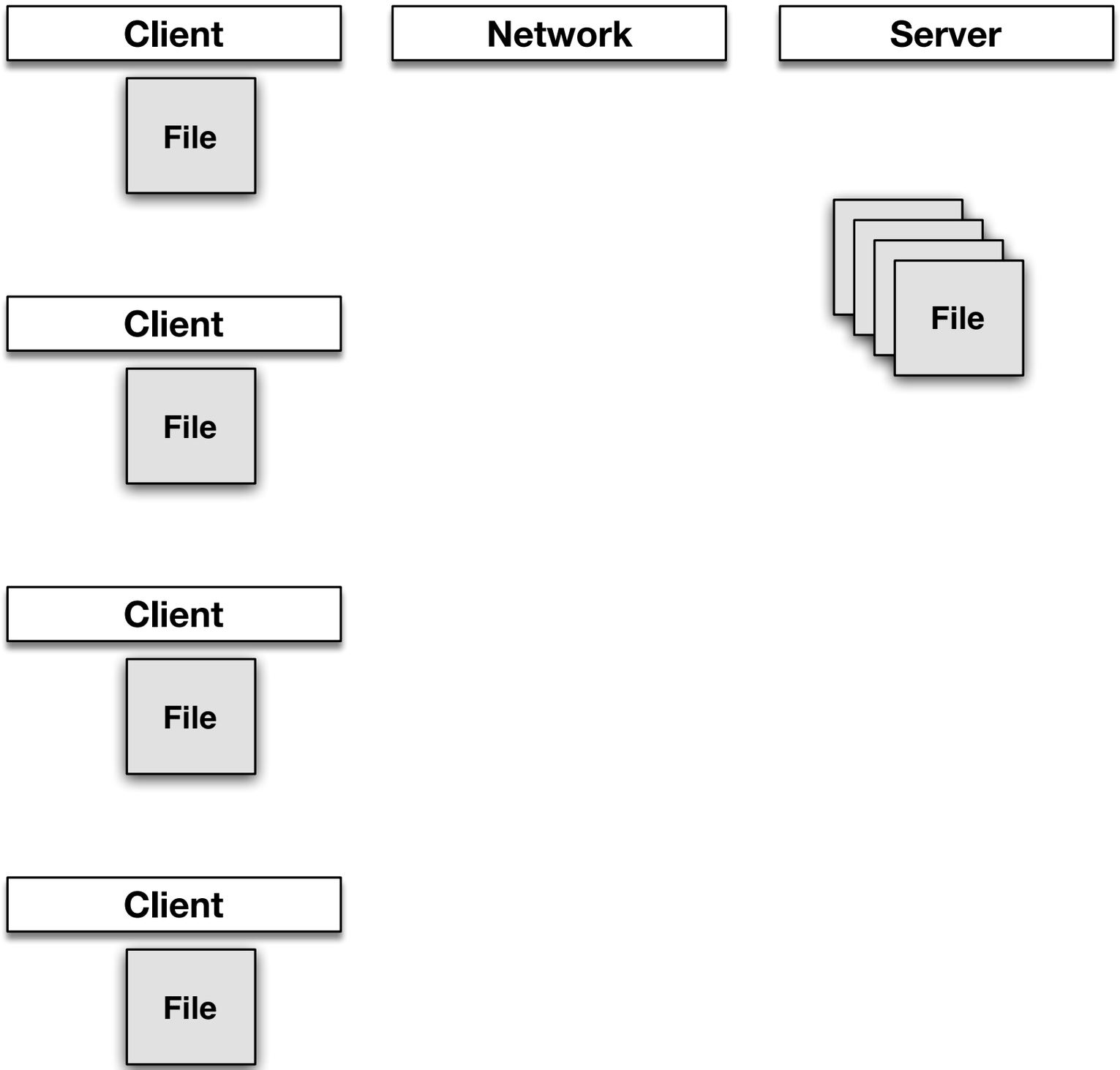
3.Inconsistent Performance: Performance can be unpredictable due to the reliance on individual capabilities and availability of peer nodes.

4.Complex Management: Managing and configuring a P2P network can be more complex due to its decentralized nature.

CLIENT SERVER



Client Server



Pros and Cons of Client Server style?

Pros and Cons of Client Server style?

Cons:

1. Single Point of Failure: The server is a critical point; if it fails, the entire network service can be disrupted.

2. Cost: Requires investment in powerful server hardware and maintenance.

3. Network Bottlenecks: Heavy reliance on the server can lead to bottlenecks, especially with a large number of clients.

Pros and Cons of Client Server style?

Pros:

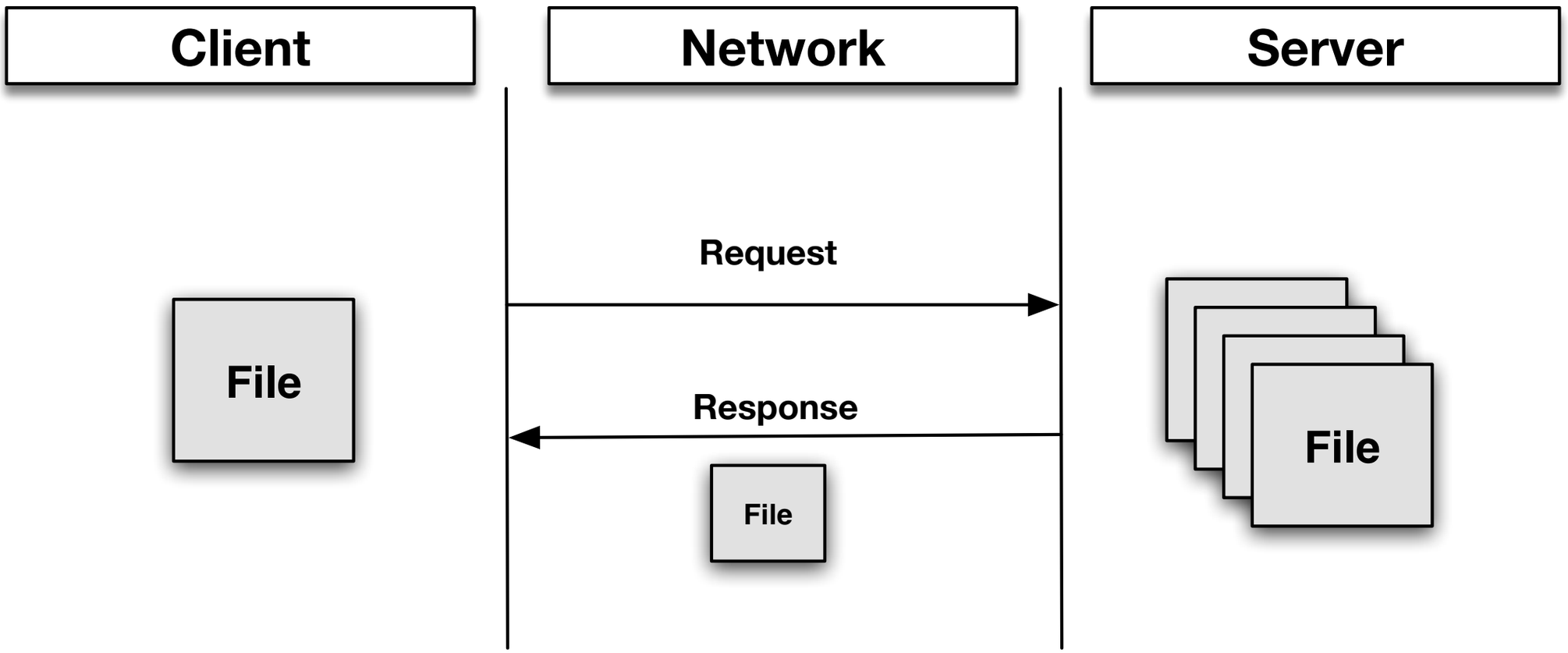
1. **Centralized Control:** Centralized servers allow for better control over data and resources, which can enhance **security and data integrity**.

2. **Performance and Reliability:** Servers in a client-server model are often more powerful and reliable than individual nodes in a P2P network.

3. **Easier Management:** Centralized architecture can be easier to manage and maintain, especially in terms of updates and security.

4. **Consistent Quality of Service:** Clients can expect a more consistent level of service due to the dedicated resources of the server.

What is the network?



Network Evolution

- 1960s: First network communication, the rise of the **ARPANET** and packet-switched networks at UCLA and DARPA.

- 1970s: Development of **TCP/IP** protocol stack at Stanford, Univ. College London, BBN Tech.

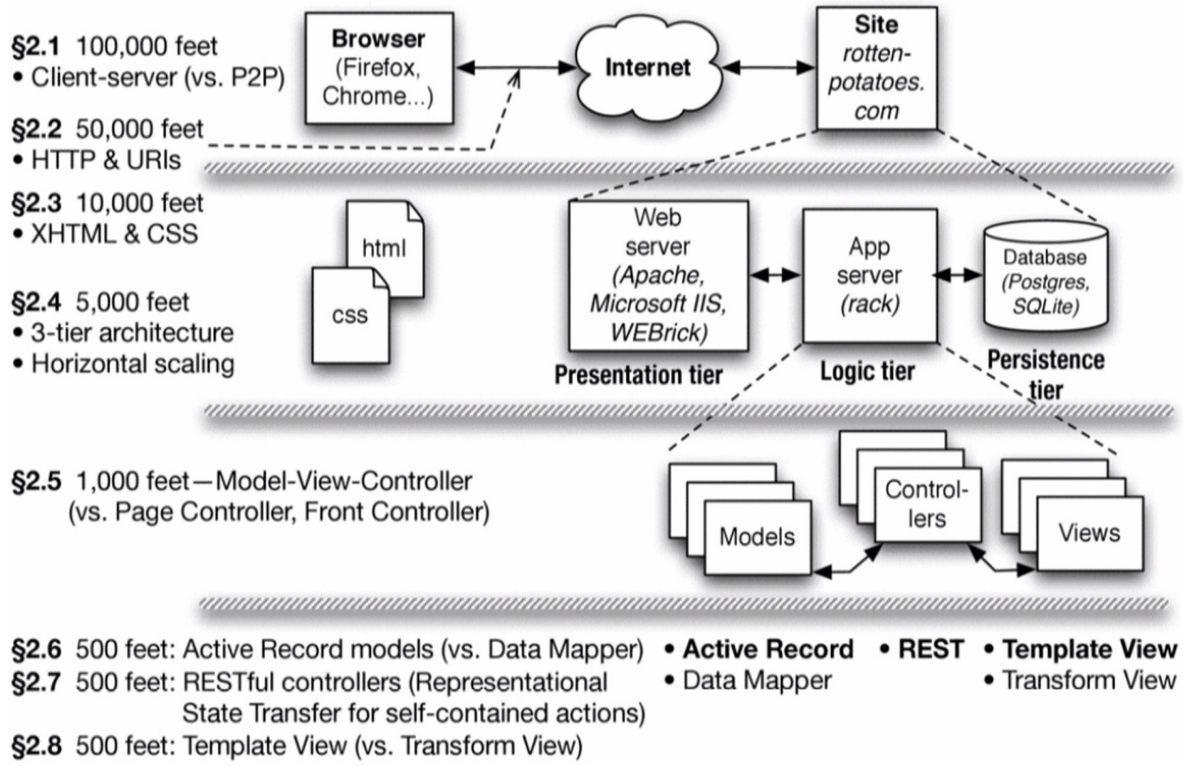
- Internet in the 1980's:

  - TCP/IP becomes standard protocol on ARPANET

  - Tim Berners Lee invents **HTTP** (and HTML)

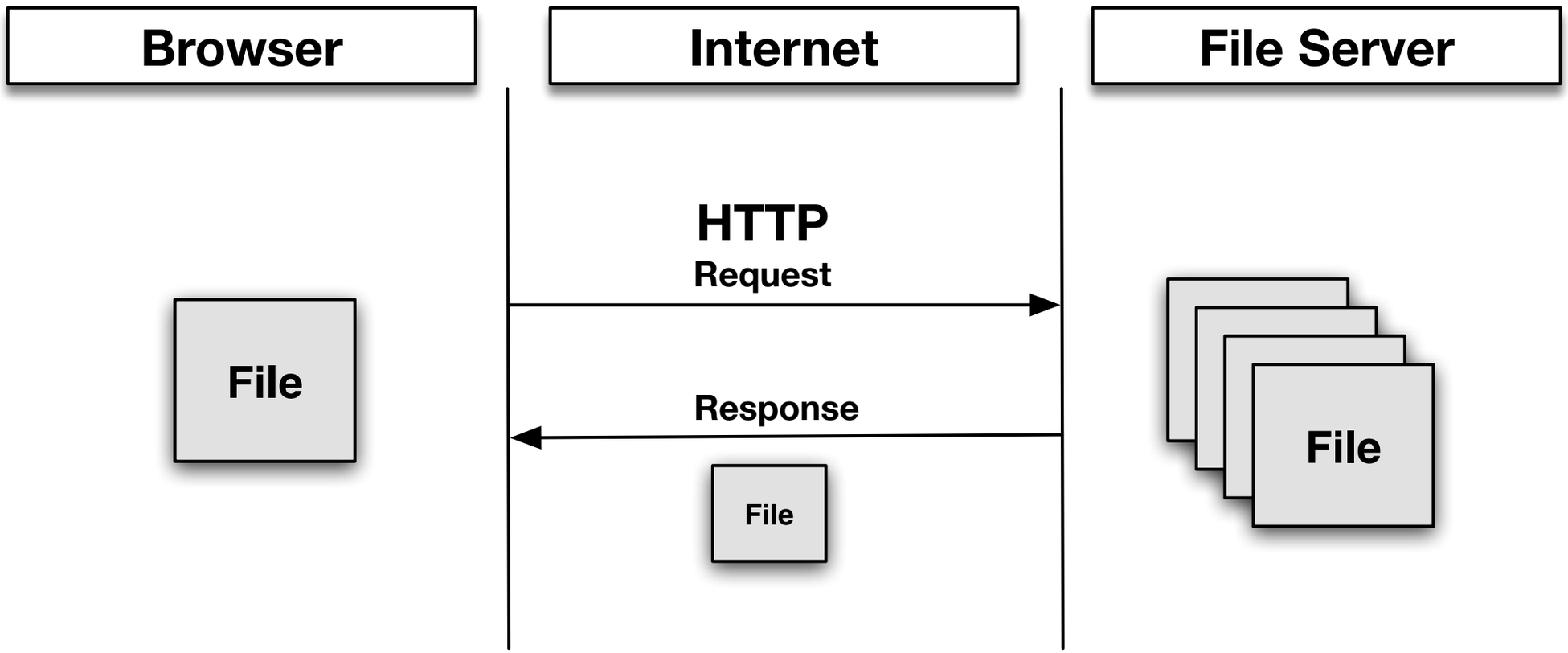
SaaS Infrastructure

- Communication in SaaS is done through the Internet.
- TCP/IP (Transmission Control Protocol/Internet Protocol): communication protocols to interconnect network devices on the internet.
- TCP/IP establishes how information can be transferred from one node to another, by communicating ordered sequences of bytes.
- Network Protocols over TCP/IP:
  - HTTP: Hypertext Transfer Protocol
  - SFTP: SSH File Transfer Protocol
  - FTP: File Transfer Protocol



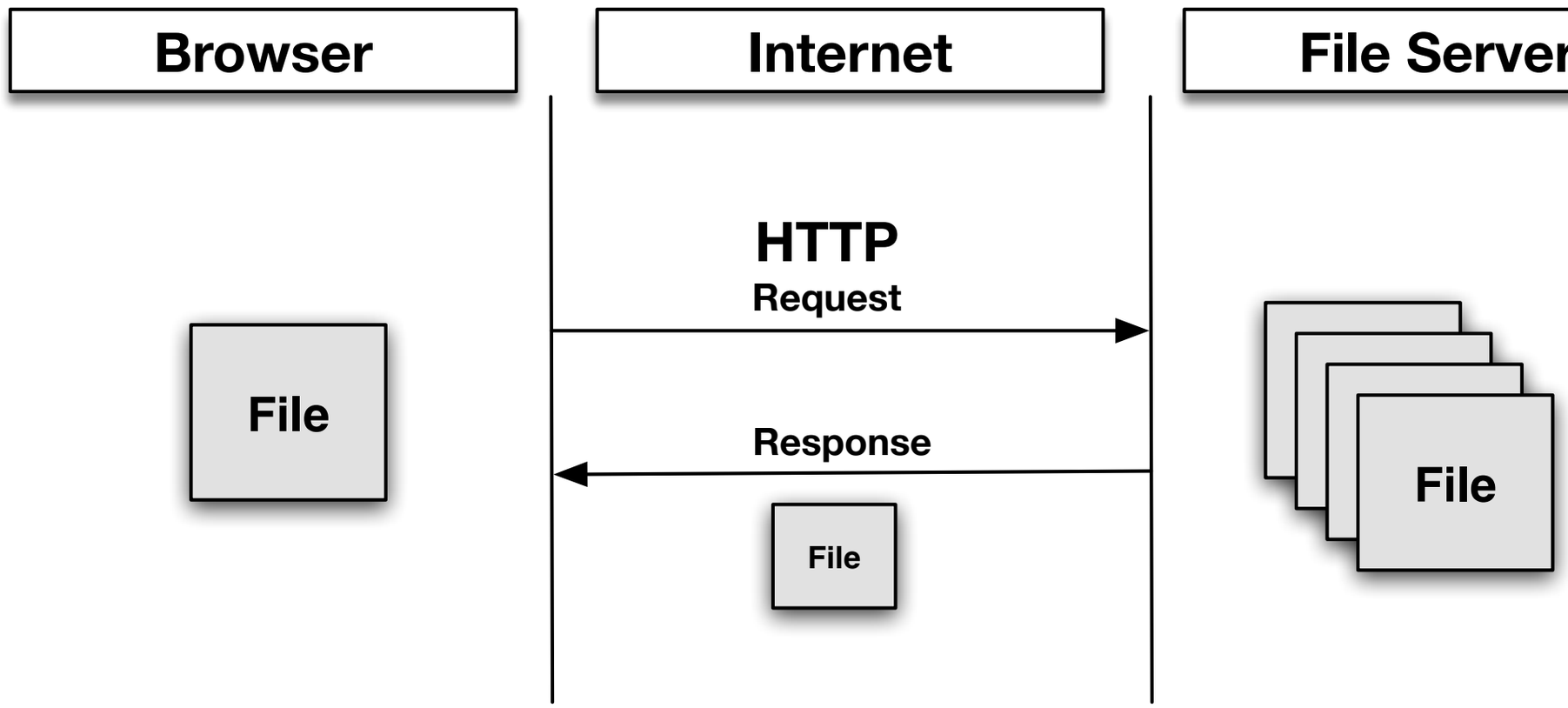
Saas Web Applications use HTTP to communicate which uses TCP/IP

HTTP over Internet



HTTP: Hypertext Transfer Protocol

What is HTTP request?

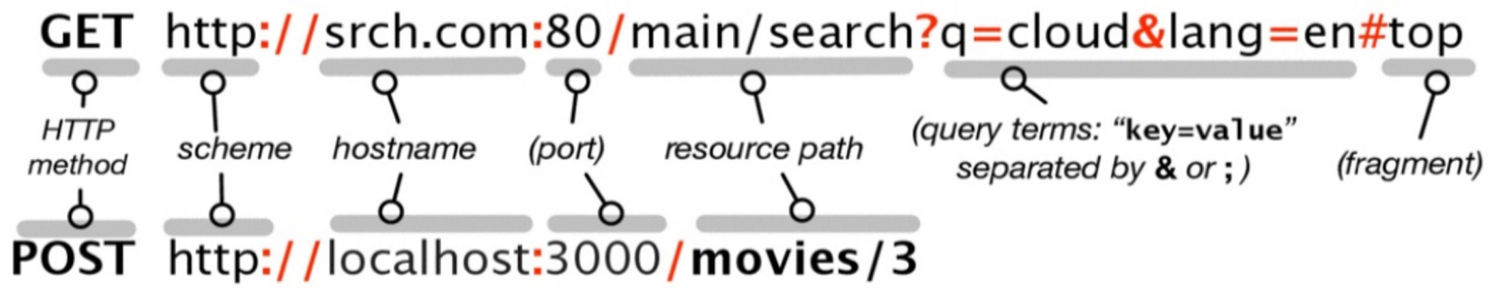


HTTP: Hypertext Transfer Protocol

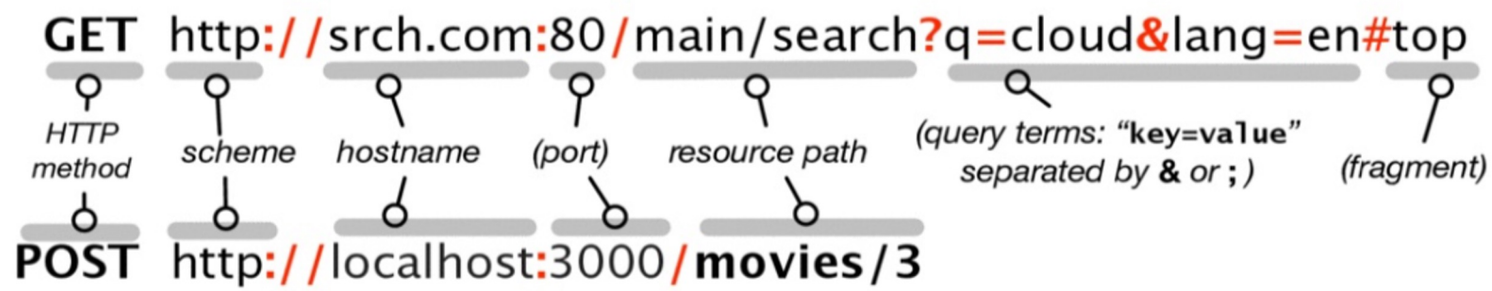
TCP/IP → HTTP

- Allows communication of arbitrary character strings between a pair of network entities.
- In TCP/IP network each computer has an **IP Address**. Browsers automatically contact DNS and get the IP address.
- Referring to **localhost** points to the local computer app is running on.
- Multiple agents on a network can be running at the same IP address, so TCP/IP uses **port** numbers to distinguish different network agents at the same IP address.
- All protocols based on TCP/IP like HTTP must specify the host and port number while opening a connection. Once one agent opens a connection to another agent using the hostname and port number as specified in TCP/IP, the communication in HTTP is then initiated.
- An HTTP server process must be listening for connections on that host and port number.

HTTP Request



HTTP Request



- HTTP request = HTTP method (**GET**,**POST**, **PUT**, **DELETE**) + **URL**
- **URL** (Uniform Resource Locator) is the string typed in address bars that shows:
  - Name of the communication scheme to retrieve info: HTTP
  - Hostname with optional port number: "cs169.com" or "localhost:3000"
  - Resource on that host to retrieve: "movies/15/reviews"
- URL points to a resource available on the internet.
- HTTP is a stateless protocol:
  - independent requests every time
- HTTP is a request-reply protocol
- URI (Uniform Resource Identifier): Umbrella term, contains URL and URN (Unified Resource Name)

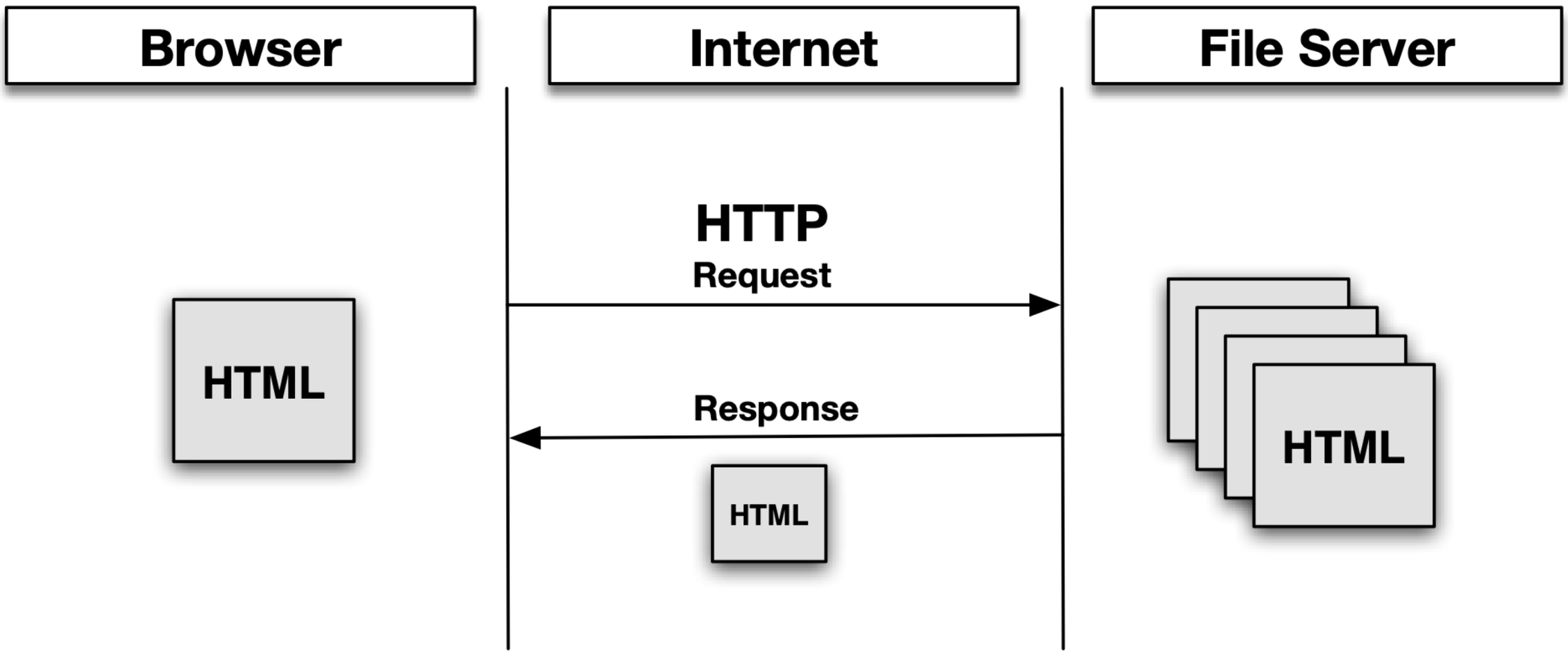
URL: Schemes

- http: is the most common scheme; it means use the HTTP protocol https: is similar to http: except that it uses SSL encryption
- file: means read a file from the local disk
- websocket: means create a TCP connection
- mailto: means open an email program composing a message

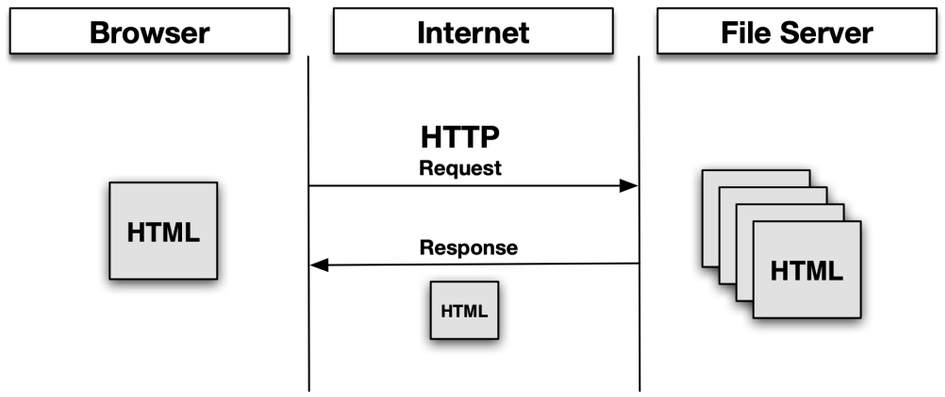
There are many (~350) other schemes: https://www.iana.org/assignments/uri-schemes/

- Example: mongodb: points to a MongoDB database

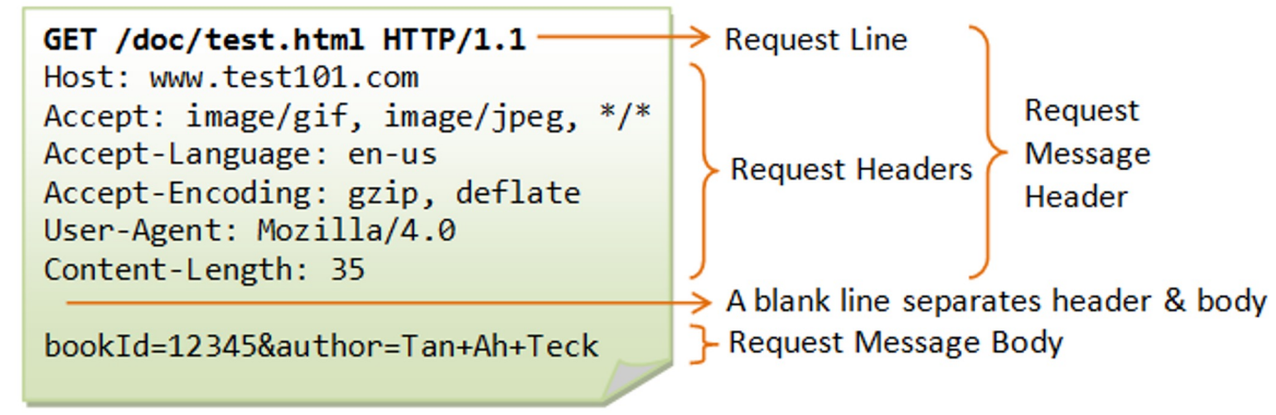
Static HTML Web Pages

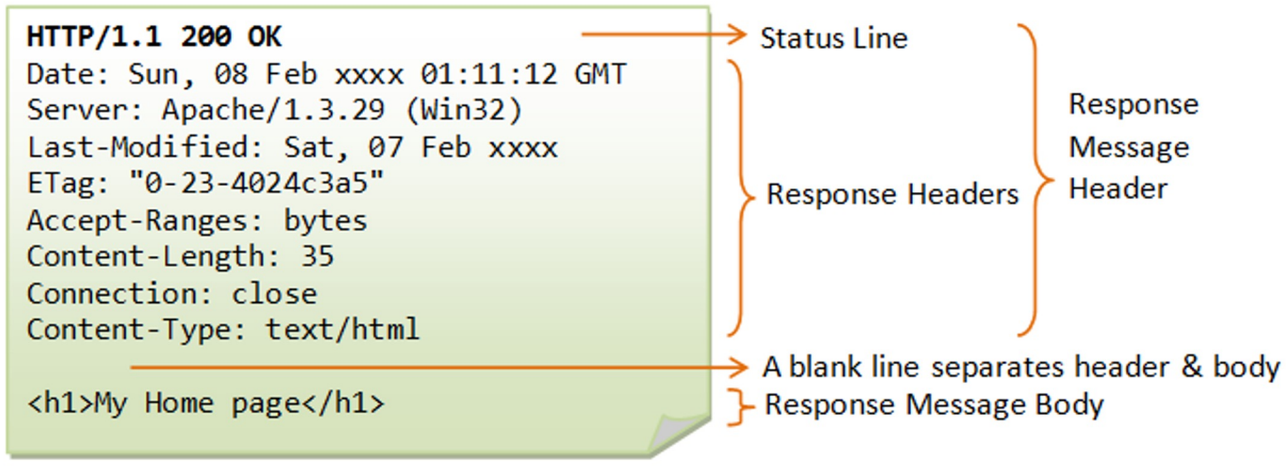


HTTP Communication



1. HTTP Server is listening to localhost:3000
2. Web Client requests Rotten Potatoes home page from a Web Server
   a. Web Client's browser creates HTTP request using GET method and URI: localhost:3000 to contact web server listening to same computer, same port.
   b. Web Server receives the HTTP request
3. HTTP Server obtains Rotten Potatoes page and sends this content back to Client
   a. Server returns content encoded in HTML using HTTP
   b. Client receives the content as a HTTP reply
4. Web Client displays content according to directives
5. Web Client closes its HTTP connection
6. Web Server is still listening for clients.

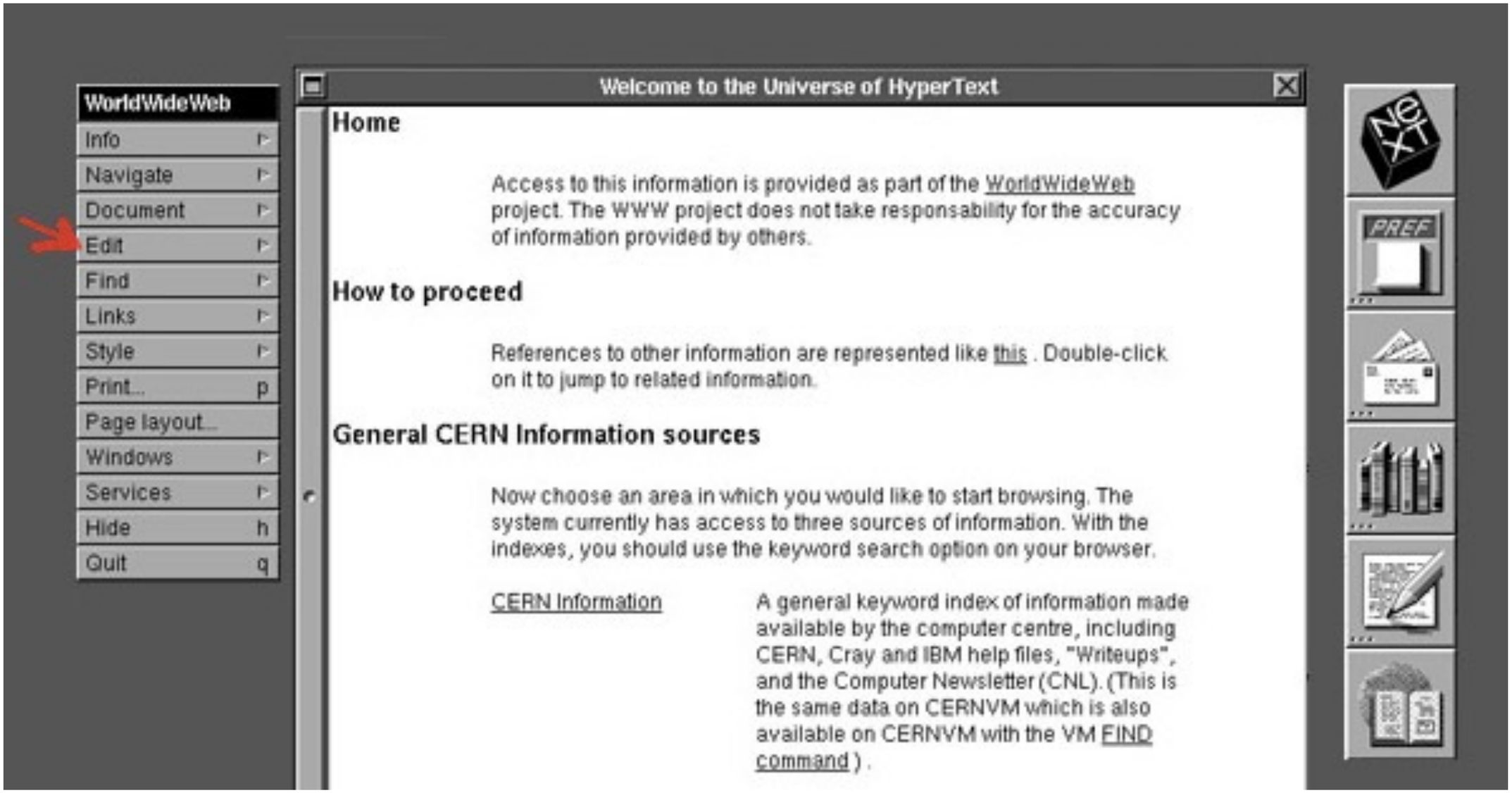




1985-1990: HTML

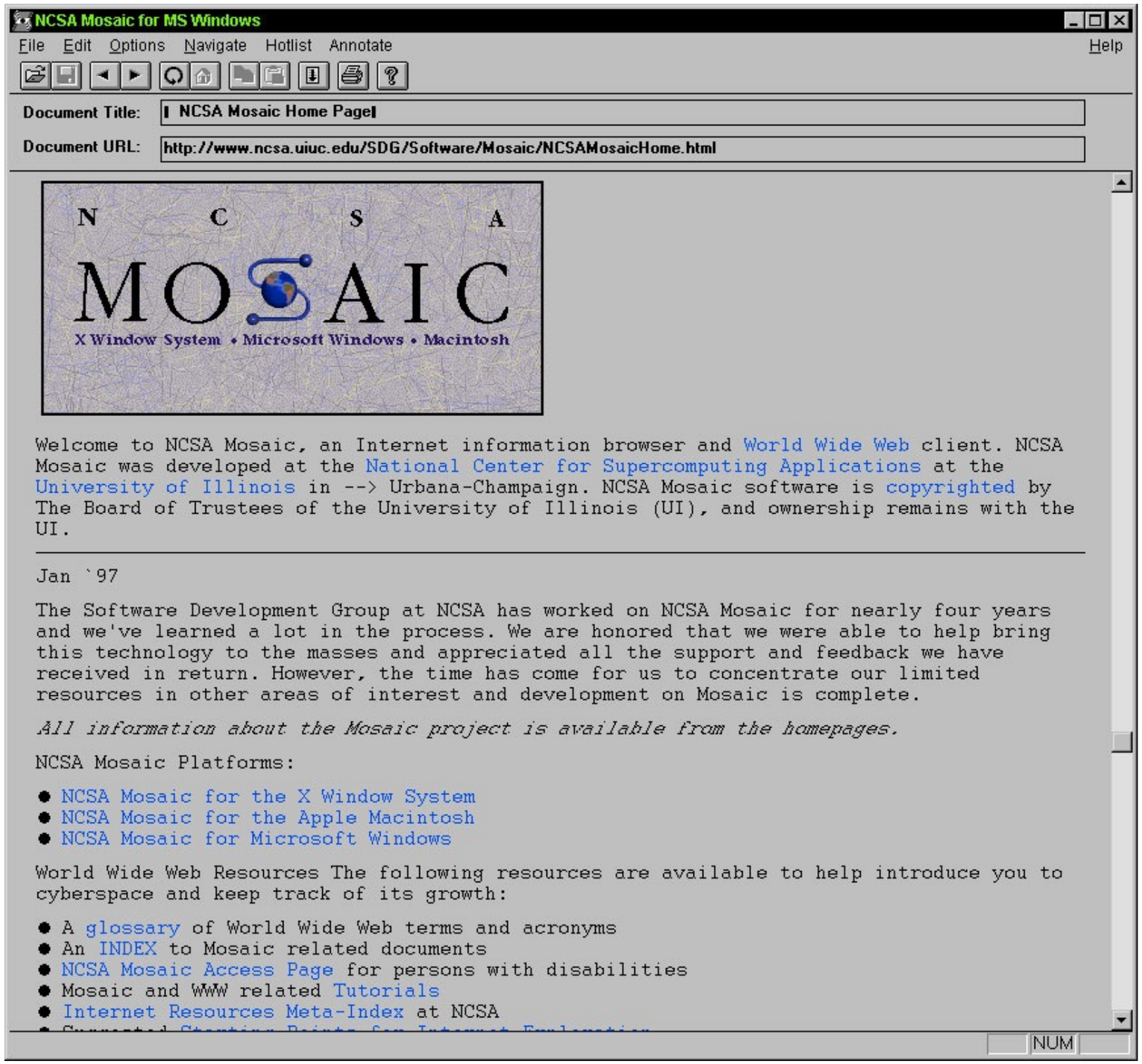
- HTML: Way to embed hyperlinks within Documents to navigate to new documents
  - Invented by Tim Berners Lee at CERN for physicists to share documents in the late 80's
- Extension of the SGML (Standard Generalized Markup Language) in the mid 1980's
- Formalized as the HTTP protocol by Lee (1992)
- Early web browsers were text-based e.g., Lynx

Tim Berners Lee's Nexus Browser



Source: Digital-archeology.org

Mosaic Web Browser: Screenshot

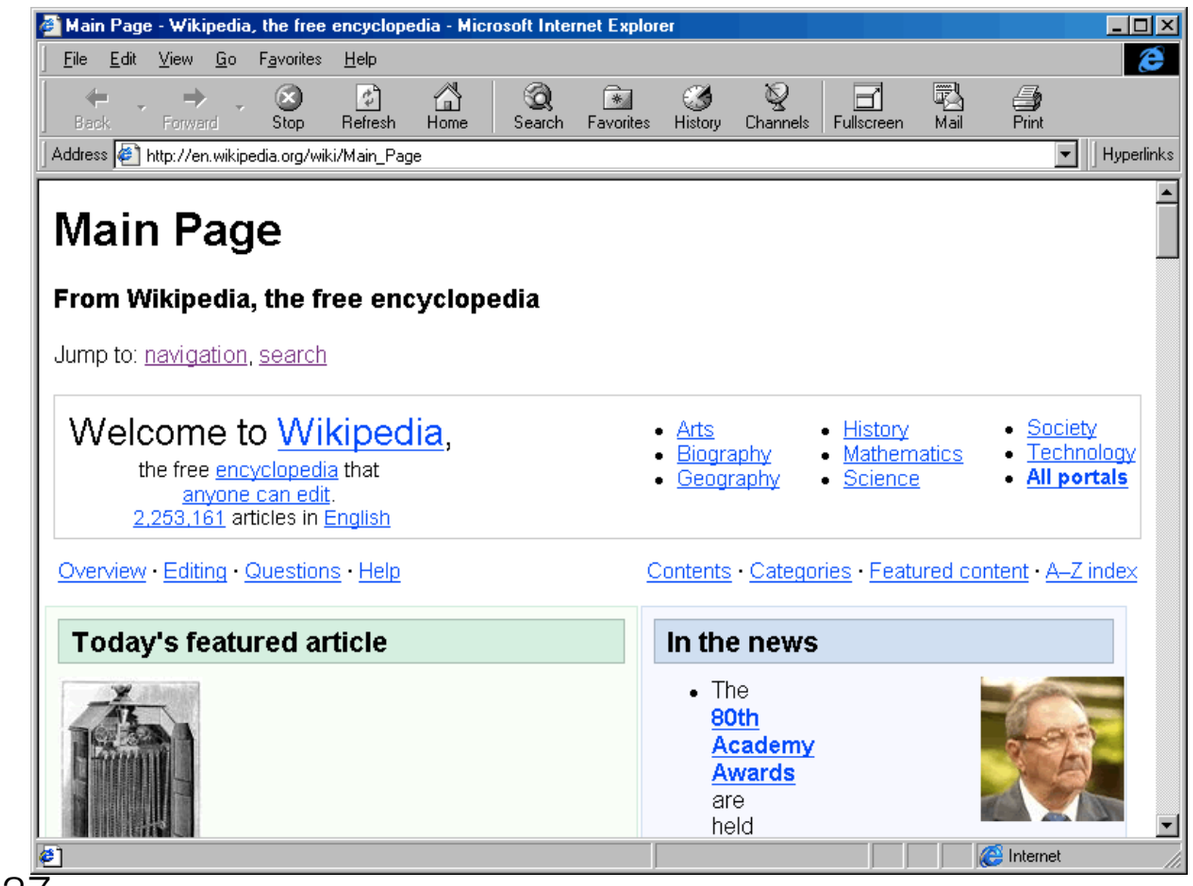


Source:
www.nsf.org

1995-2000: Browser Wars

- Netscape Navigator Versus Internet Explorer (IE)
  - Initially, Netscape had more features and was more robust
  - IE had the marketing power of Microsoft and was bundled with Windows as a free download!
- JavaScript 1.0 was developed and integrated with Netscape by Brendan Eich (more on this later)
- In 1997, Microsoft released IE4 which had feature parity with Netscape and integrated with Windows
  - Netscape was never the same again. They were sold to AOL and essentially lost the browser wars by the early 2000s

Netscape Navigator 4 Vs. IE 4



2000-2005: Browser Wars Part 2

- But, while Netscape was finished, they released their code to the **Mozilla** foundation

  - Team of dedicated volunteers that rebuilt Navigator from scratch and ironed out its quirks

  - First release in 2004. Rapid releases in 2005,2006

  - First browser that was World Wide Web (WWW) **standards-compatible**

  - By 2006, Mozilla (Firefox) was back in the game.

HTML Links

- Browser maintains a notion of current location (i.e. URL)

- Links: content in a page which, when clicked on, causes the browser to go to URL

- Links are implemented with the <a> tag in HTML:

2024 News

Different Types of Links

- **Full URL**: 2029 News
- **Absolute** URL: <a href="/stock/quote.html">
  - same as http://www.xyz.com/stock/quote.html
- **Relative** URL: <a href="2008/March.html">
  - same as http://www.xyz.com/news/2008/March.html
- Define an **anchor** point (a position that can be referenced with # notation):
- <a **name**="sec3">
- Go to a different place in the same page: <a href="**#**sec3">

Uses of URLs

- Loading a page: type the URL into your browser
- Load an image:

```
<img src="..." />
```

- Load a stylesheet:

```
<link rel="stylesheet" type="text/css" href="...">
```

- Embedded a page:

```
<iframe src="http://www.google.com">
```

Activity

Assume we have the following files on the server at: potato.com
- news/index.html
- news/styleA.css
- news/top3.html

Given an HTML document loaded from the URL:

```
http://potato.com/news/
```

That contains the element in its `head` section

```
<link rel="stylesheet" type="text/css" href="/styleA.css?ref=v1" />
```

Explain what the browser shows and why?