

# **Einführung in das Web**

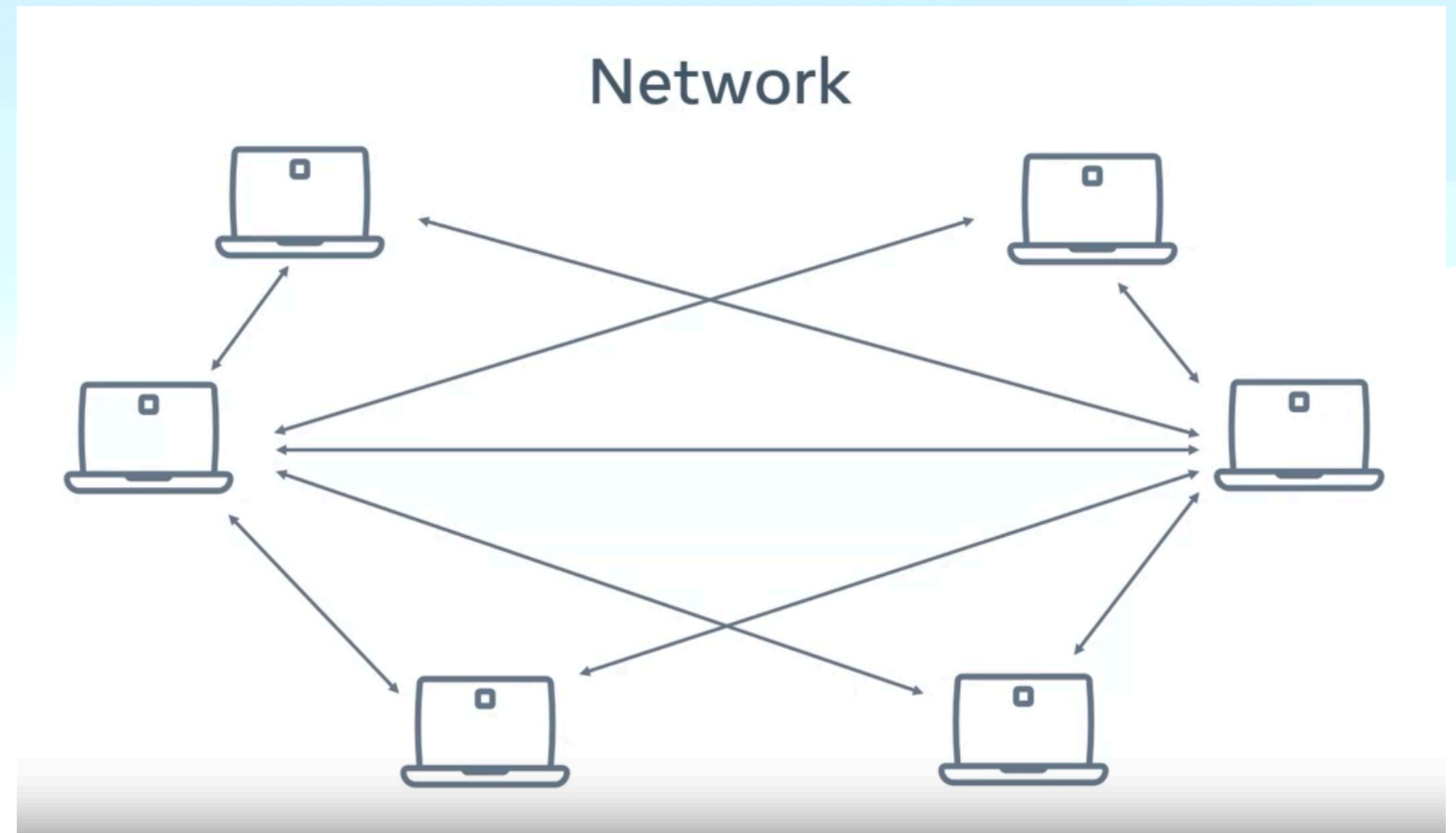
**Netzwerke und Protokolle**

# Netzwerk

Das Internet wird durch Geräte ermöglicht, die sich über kabel- oder kabellose Verbindungen miteinander kommunizieren.

So entsteht ein Netzwerk, an das mehrere Geräte angeschlossen werden können.

Es muss jedoch jedes Gerät mit jedem verbunden werden.



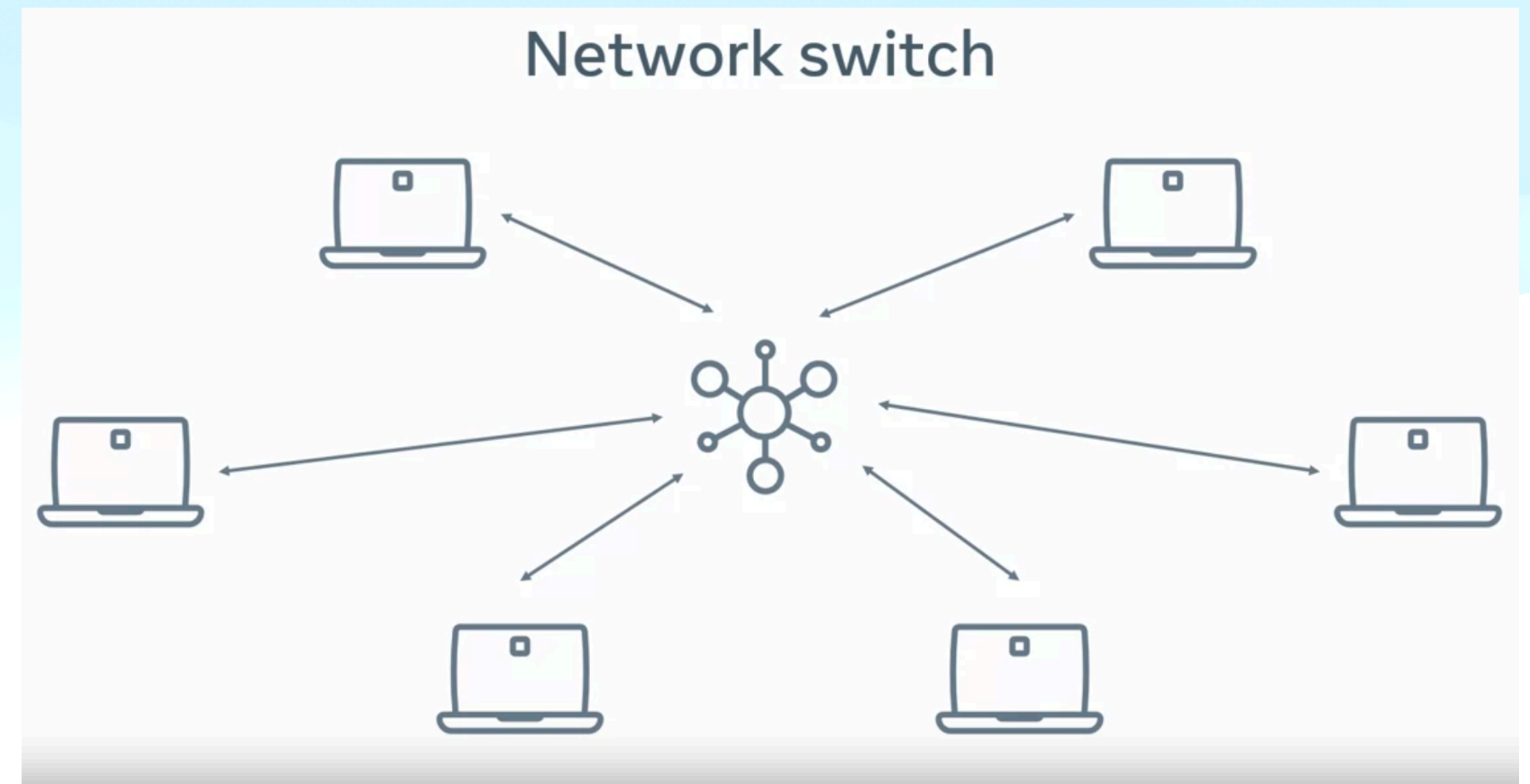


# Netzwerk-Switch

Um eine effektive Kommunikation zwischen Geräten zu ermöglichen, wird ein Netzwerk-Switch verwendet.

Er verbindet mehrere Geräte und ermöglicht es ihnen, miteinander zu kommunizieren.

Netzwerk-Switches können auch mit anderen Netzwerk-Switches verbunden werden, wodurch ein zusammenhängendes Netzwerk entsteht.

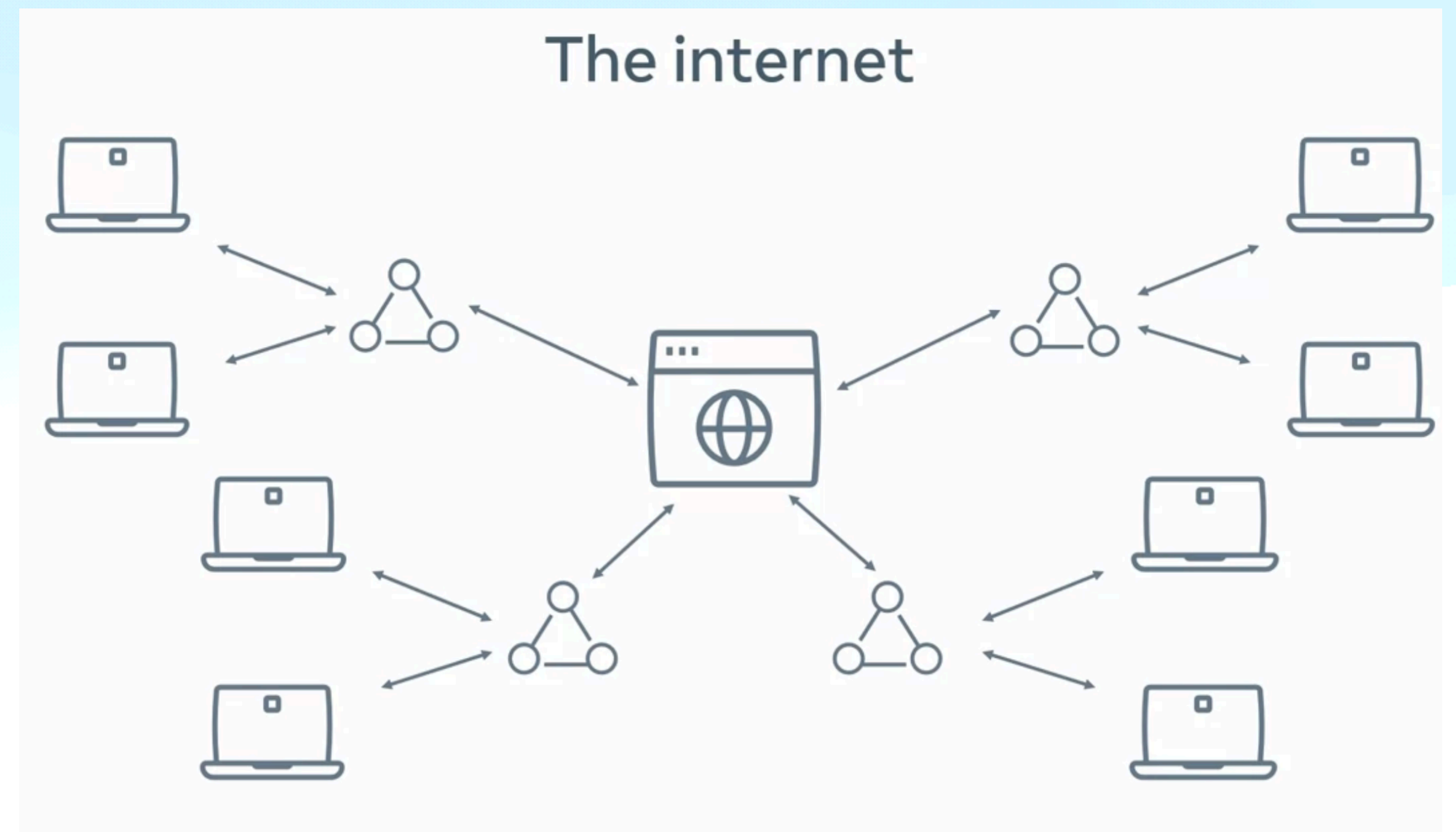


# Das Internet

Das durch Netzwerk-Switches gebildete Verbundnetz wird als Internet bezeichnet.

Es verbindet die ganze Welt und ermöglicht den Zugang zu Websites und Diensten, die von Computern, sogenannten Servern, bereitgestellt werden.

Das World Wide Web (kurz „Web“) ist ein System, das es uns ermöglicht, Informationen über das Internet zu finden und anzuzeigen. Es basiert auf einer Client-Server-Architektur, bei der ein Client (meist ein Webbrowser wie Chrome oder Firefox) mit einem Server kommuniziert, um Webseiten und andere Inhalte anzufordern.





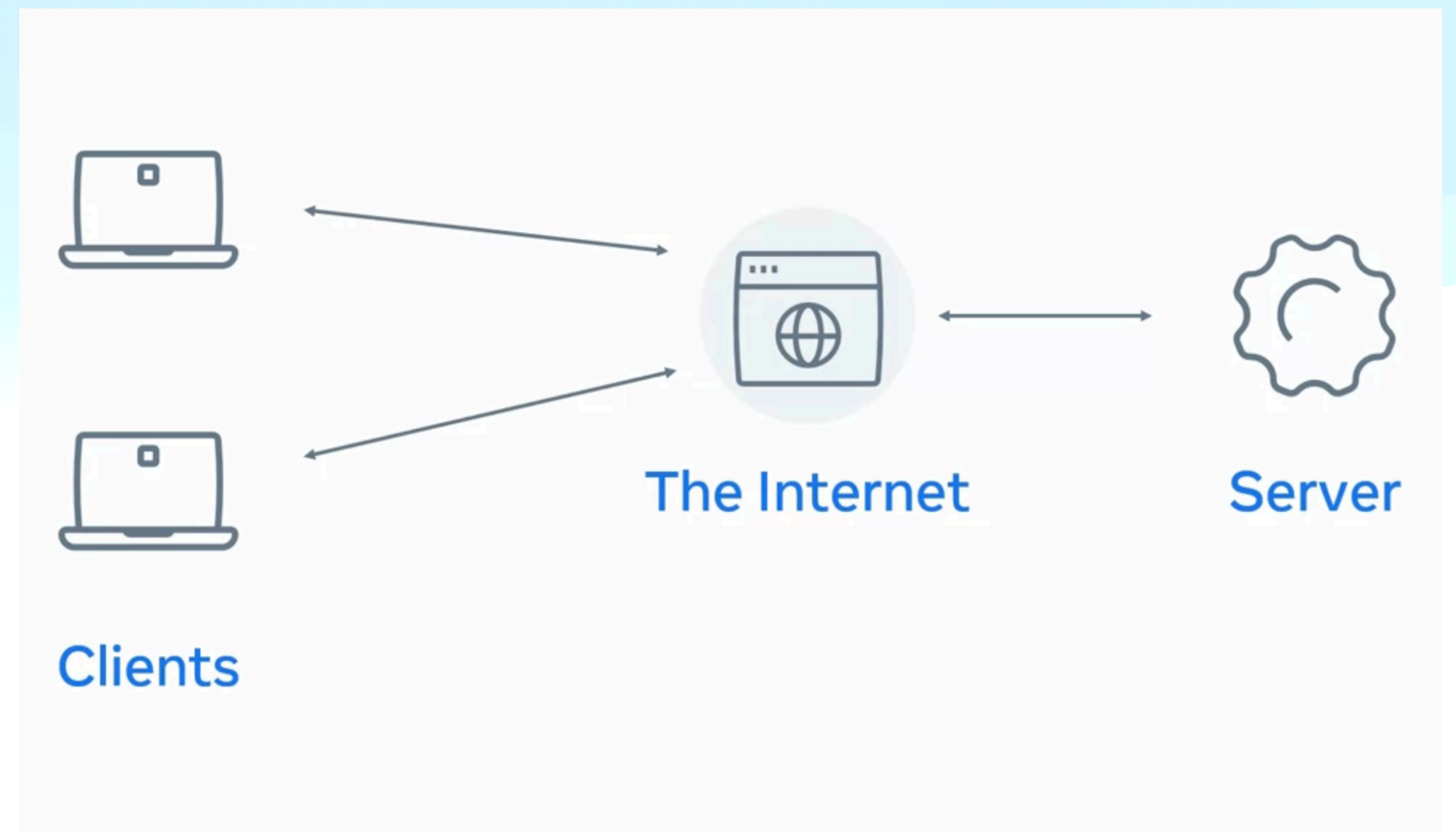
# Client-Server-Architektur

Die Client-Server-Architektur beschreibt, wie zwei Computer (Client und Server) miteinander kommunizieren, um Informationen auszutauschen.

- **Client:** Das Gerät oder die Software, die eine Anfrage sendet (z.B. Laptop mit dem Browser).
- **Server:** Ein Computer oder Programm, das Anfragen von Clients empfängt und die angeforderten Daten (z.B. HTML-Dateien) zurücksendet.

## Ablauf einer Kommunikation:

1. Der Client (z.B. Browser) sendet eine Anfrage an den Server, um eine Webseite anzufordern. Dies wird als **HTTP-Request** bezeichnet.
2. Der Server empfängt die Anfrage, verarbeitet sie und sendet die angeforderten Informationen (z.B. die Webseite) als **HTTP-Response** zurück.
3. Der Client erhält die Antwort und zeigt die Webseite im Browser an.



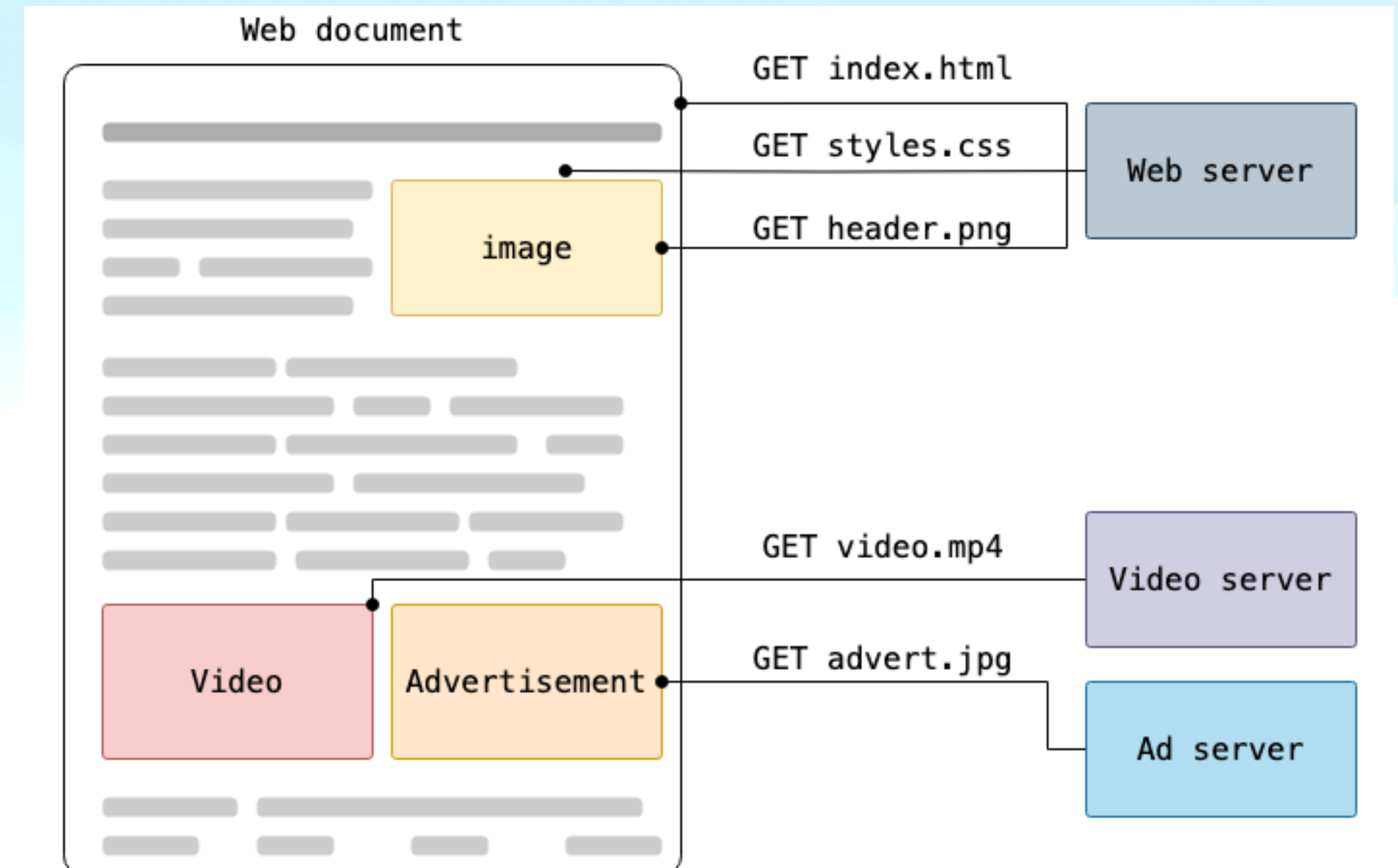
# HTTP

## Hypertext Transfer Protocol

### Overview of HTTP auf man web docs

HTTP ist das Protokoll, das die Kommunikation zwischen Client und Server ermöglicht.

Es legt fest, wie Anfragen (**Requests**) gesendet und Antworten (**Responses**) empfangen werden.





# HTTP-Request

Jeder HTTP-Request beginnt mit der Anfragezeile.

Diese besteht aus:

- der HTTP-Methode,
- der angeforderten Ressource und
- der HTTP-Protokollversion.

In diesem Beispiel ist GET die HTTP-Methode,  
/index.html ist die angeforderte Ressource und  
HTTP 1.1 ist das verwendete Protokoll.

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
Accept-Language: en-US,en;q=0.5
```

# HTTP-Request-Methoden

## GET

- Fordert Daten vom Server an. (z.B. beim Laden einer Webseite oder beim Abrufen von Daten über eine API).
- **Eigenschaften:**
  - Daten werden in der URL gesendet (z.B. als Query-Parameter).
  - Sicher: GET-Anfragen sollten keine Veränderungen auf dem Server vornehmen.
  - Antwort enthält die angeforderten Daten, z.B. eine HTML-Seite oder JSON.

## POST

- Sendet Daten an den Server, um eine Aktion auszuführen (z.B. Formulardaten absenden).
- **Eigenschaften:**
  - Die gesendeten Daten werden im **Body** der Anfrage übertragen, nicht in der URL.
  - Verwendet, wenn auf dem Server etwas verändert werden soll (z.B. das Erstellen eines neuen Eintrags in einer Datenbank).
  - Antwort enthält häufig eine Bestätigung, dass die Daten erfolgreich empfangen wurden.

## PUT

- Aktualisiert oder ersetzt eine vorhandene Ressource auf dem Server (z.B. bei der Aktualisierung von Daten in einer Datenbank über APIs).
- **Eigenschaften:**
  - Überträgt Daten im **Body** der Anfrage.
  - Im Gegensatz zu POST wird PUT verwendet, um eine spezifische Ressource vollständig zu aktualisieren oder zu überschreiben.
  - Idempotent: Mehrere identische PUT-Anfragen haben den gleichen Effekt wie eine einzige.

## DELETE

- Löscht eine Ressource auf dem Server (z.B. entfernen von Daten in der Datenbank).
- **Eigenschaften:**
  - Löscht eine spezifische Ressource, die über die URL identifiziert wird.
  - Idempotent: Mehrere identische DELETE-Anfragen führen zum gleichen Ergebnis (die Ressource ist nach der ersten Anfrage gelöscht).



# Request Headers

Nach der Anfragezeile folgen die HTTP-Header mit einem Zeilenumbruch.

Es gibt verschiedene Möglichkeiten, einen HTTP-Header in die HTTP-Anfrage aufzunehmen.

Ein Header ist ein Name, bei dem die Groß- und Kleinschreibung nicht beachtet wird, gefolgt von einem `:` und dann von einem Wert.

- Host spezifiziert den Host des Servers und gibt an, wo aus die Ressource angefordert wird.
- User-Agent-Header informiert den Webserver über die Anwendung, die die Anfrage stellt. Er enthält oft das Betriebssystem (Windows, Mac, Linux), die Version und den Hersteller der Anwendung.

```
Host: example.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:50.0)
Gecko/20100101 Firefox/50.0
Accept: */*
Accept-Language: en
Content-type: text/json
```

- Accept-Header teilt dem Webserver mit, welche Art von Inhalt der Client als Antwort akzeptiert.
- Accept-Language-Header gibt die Sprache und optional das Gebietsschema an, das der Client bevorzugt.
- Content-Type-Header gibt die Art des Inhalts an, der im Body der Anfrage übertragen wird.

# Request Body

HTTP-Anfragen können optional einen Anfragekörper enthalten.

Ein Request Body wird häufig bei der Verwendung der HTTP-Methoden POST und PUT zur Übermittlung von Daten verwendet.

```
POST /users HTTP/1.1
Host: example.com

{
  "key1": "value1",
  "key2": "value2",
  "array1": ["value3", "value4"]
}
```

```
PUT /users/1 HTTP/1.1
Host: example.com
Content-type: text/json

{"key1": "value1"}
```



# HTTP Response

Wenn der Webserver die Bearbeitung der HTTP-Anfrage abgeschlossen hat, sendet er eine HTTP-Antwort zurück.

Die erste Zeile der Antwort ist die Statuszeile. Diese Zeile zeigt dem Client an, ob die Anfrage erfolgreich war oder ob ein Fehler aufgetreten ist.

HTTP/1.1 200 OK

Die Zeile beginnt mit der Version des HTTP-Protokolls, gefolgt von dem Statuscode und einer Begründung. Die Reason Phrase ist eine textuelle Darstellung des Statuscodes.

```
HTTP/1.1 200 OK
Date: Mon, 10 Sep 2024 12:00:00 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Type: text/html; charset=UTF-8
Content-Length: 1024
```

```
<!DOCTYPE html>
<html>
<head>
  <title>Example Page</title>
</head>
<body>
  <h1>Hello, World!</h1>
  <p>Welcome to the example page.</p>
</body>
</html>
```

# HTTP Status Codes

Die erste Ziffer eines HTTP-Statuscodes gibt die Kategorie der Antwort an:

Information, Erfolgreich, Umleitung, Client-Fehler oder Server-Fehler.

Die üblichen Statuscodes-Kategorien sind:

## 1xx: Informative Antworten

- **100 Continue:** Der Server hat den Anfang der Anfrage empfangen und der Client kann fortfahren.

## 2xx: Erfolgreiche Anfragen

- **200 OK:** Die Anfrage war erfolgreich, und die angeforderten Daten werden zurückgegeben.
- **201 Created:** Eine Ressource wurde erfolgreich erstellt (meist bei POST-Anfragen).

## 3xx: Umleitungen

- **301 Moved Permanently:** Die angeforderte Ressource wurde dauerhaft verschoben. Die neue URL wird in der Antwort angegeben.
- **302 Found:** Die Ressource ist temporär an einer anderen Stelle verfügbar.

## 4xx: Client-Fehler

- **400 Bad Request:** Die Anfrage war fehlerhaft (z.B. falsche Syntax).
- **401 Unauthorized:** Der Client ist nicht authentifiziert.
- **403 Forbidden:** Der Zugriff auf die Ressource ist verboten.
- **404 Not Found:** Die angeforderte Ressource konnte nicht gefunden werden.

## 5xx: Server-Fehler

- **500 Internal Server Error:** Es gab einen Fehler auf dem Server.
- **503 Service Unavailable:** Der Server ist vorübergehend nicht verfügbar (z.B. wegen Wartungsarbeiten).



# Response Headers

Nach der Statuszeile folgen optionale HTTP-Antwort-Header, gefolgt von einem Zeilenumbruch.

Ähnlich wie bei den Anfrage-Headern gibt es viele mögliche HTTP-Header, die in die HTTP-Antwort aufgenommen werden können.

- Date-Header gibt das Datum und die Uhrzeit an, zu der die HTTP-Antwort erzeugt wurde.
- Server-Header beschreibt die Webserver-Software, die zur Erzeugung der Antwort verwendet wurde.
- Content-Length-Header beschreibt die Länge der Antwort.
- Content-Type-Header beschreibt den Medientyp der zurückgegebenen Ressource (z. B. HTML-Dokument, Bild, Video).

```
Date: Fri, 11 Feb 2022 15:00:00 GMT+2
Server: Apache/2.2.14 (Linux)
Content-Length: 84
Content-Type: text/html
```

# Response Body

Nach den HTTP-Response-Headern folgt der HTTP-Response-Body.

Dies ist der Hauptinhalt der HTTP-Antwort.

Er kann Bilder, Videos, HTML-Dokumente und andere Medientypen enthalten.

```
HTTP/1.1 200 OK
Date: Fri, 11 Feb 2022 15:00:00 GMT+2
Server: Apache/2.2.14 (Linux)
Content-Length: 84
Content-Type: text/html
```

```
<html>
  <head><title>Test</title></head>
  <body>Test HTML page.</body>
</html>
```



