

# Einführung in Softwaretechnologien

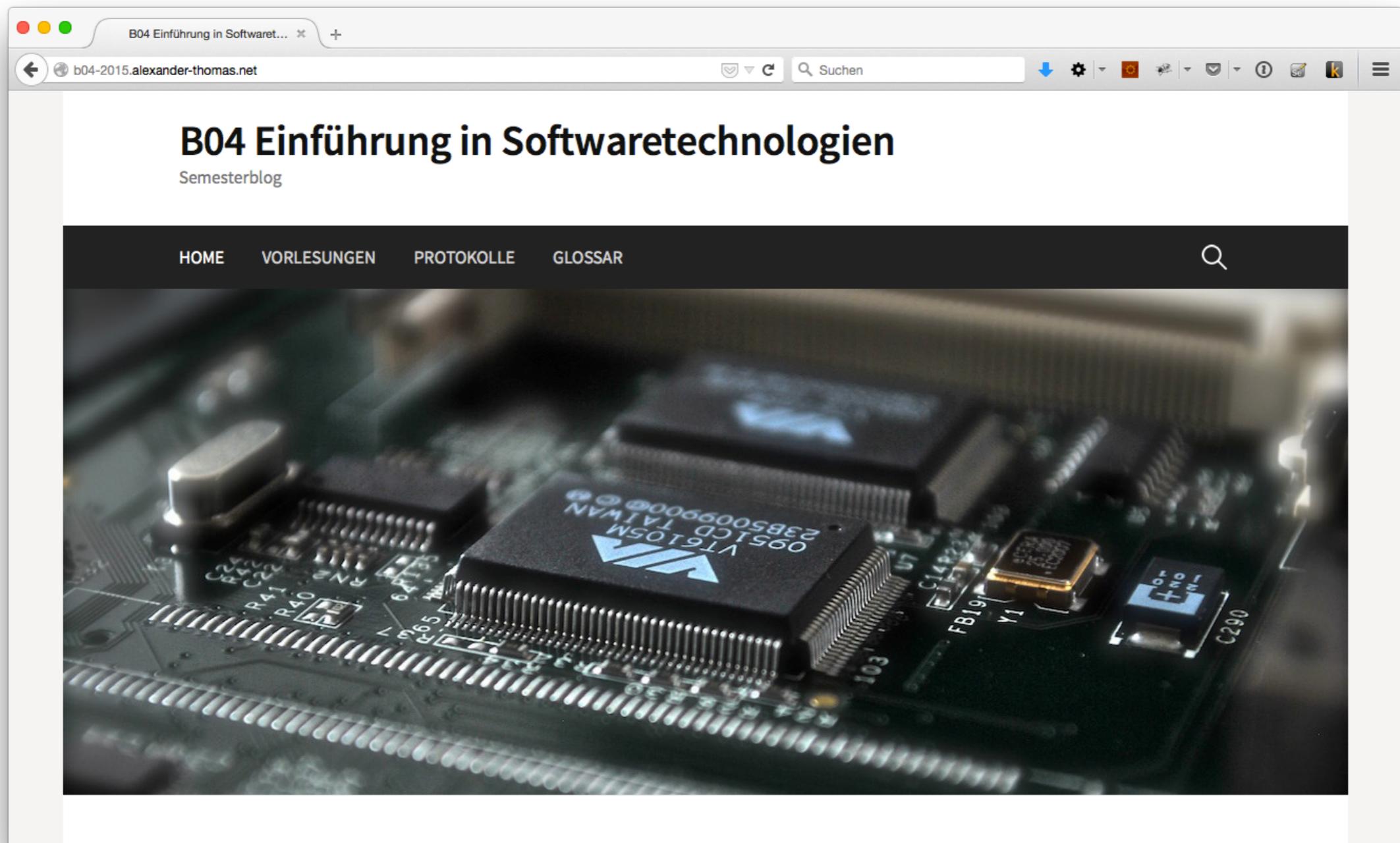
1. Vorlesung: Organisatorisches und Grundlagen

Alexander Thomas  
[me@alexander-thomas.net](mailto:me@alexander-thomas.net)

# Alexander Thomas

- Software Entwickler, Team Leiter  
freiberuflicher Berater, ...
- Kontakt:  
[me@alexander-thomas.net](mailto:me@alexander-thomas.net)





<https://b04-2015.alexander-thomas.net>

# Das Semesterblog

The screenshot shows a WordPress-based website for a course titled "B04 Einführung in Softwaretechnologien". The header includes navigation links for HOME, VORLESUNGEN, ÜBUNGEN, PROTOKOLLE, GLOSSAR, IMPRESSUM, and a search bar. A sidebar on the right contains sections for LETZTE BEITRÄGE, LETZTE KOMMENTARE, and ARCHIVE.

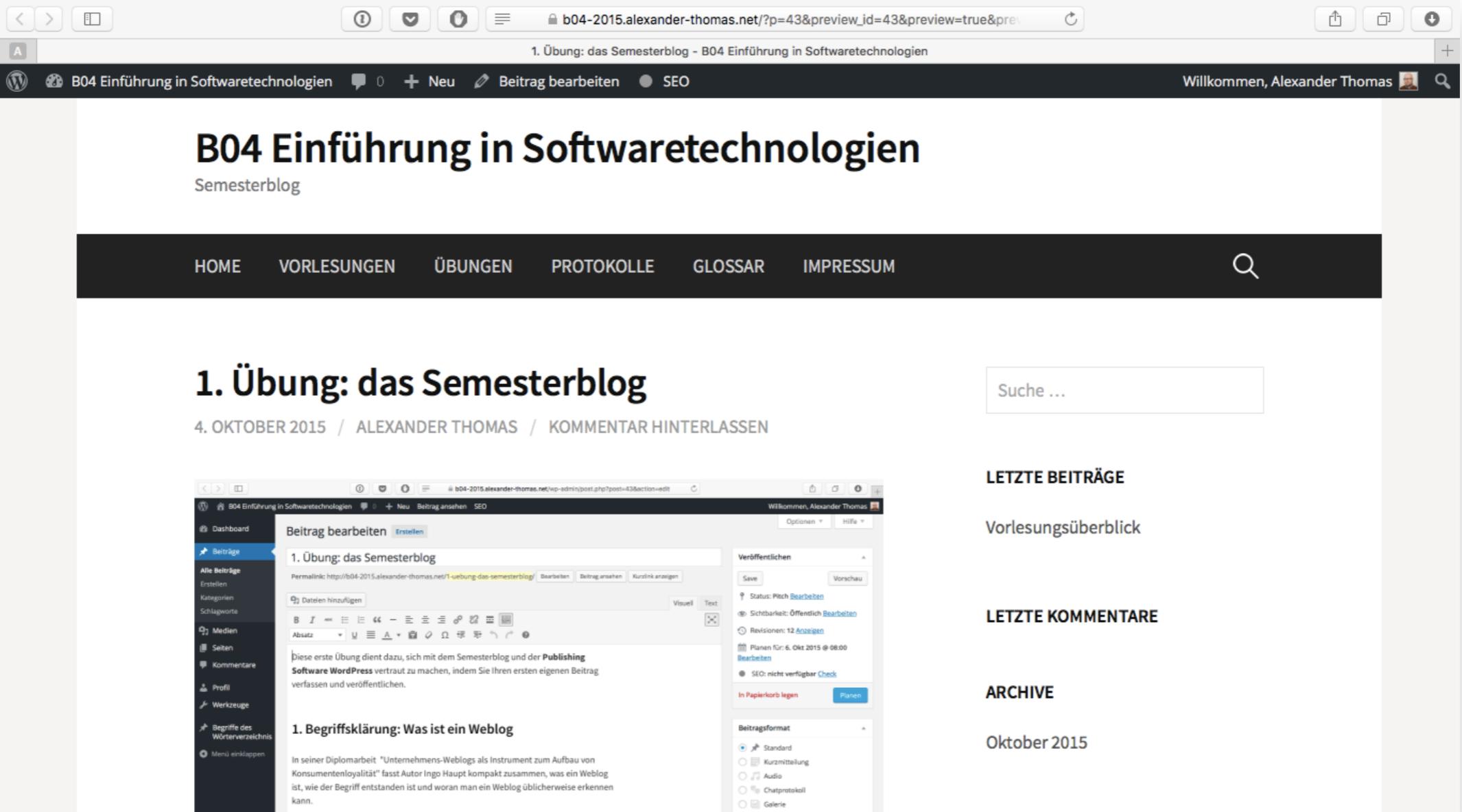
**Vorlesungsüberblick**  
1. OKTOBER 2015 / ALEXANDER THOMAS

The central content features a complex flowchart illustrating the relationships between various software technology concepts:

- Grundbegriffe** leads to **Frontend-technologien**.
- Frontend-technologien** leads to **Server-technologien**.
- Server-technologien** leads to **Netzwerke**.
- Netzwerke** leads to **Technische Konzeption & Modellierung von Softwareanwendungen**.
- Technische Konzeption & Modellierung von Softwareanwendungen** leads to **User Experience & User Interface Design**.
- User Experience & User Interface Design** leads to **Content & CMS**.
- Content & CMS** leads to **Analytics / SEO**.
- Analytics / SEO** leads back to **Grundbegriffe**.
- Frontend-technologien** also connects to **Mobile Apps**, **JS (Javascript)**, and **Datenbanken**.
- Server-technologien** connects to **Schnittstellen**, **Client-Server Systeme**, and **Datenbanken**.
- Netzwerke** connects to **Netzwerktopologien** and **Netzwerkkomponenten**.
- Technische Konzeption & Modellierung von Softwareanwendungen** connects to **UML**.

## Vorlesungsmaterialien

# Das Semesterblog



The screenshot shows a WordPress dashboard with the following details:

- Header:** The title "1. Übung: das Semesterblog - B04 Einführung in Softwaretechnologien" is displayed.
- Toolbar:** Includes standard browser controls (back, forward, search) and a tab for "Beitrag bearbeiten".
- Header Bar:** Shows the site name "B04 Einführung in Softwaretechnologien", a message "Neu", and links for "Beitrag bearbeiten" and "SEO".
- Header Area:** Displays the title "B04 Einführung in Softwaretechnologien" and the subtitle "Semesterblog".
- Navigation Bar:** A dark bar with links to "HOME", "VORLESUNGEN", "ÜBUNGEN", "PROTOKOLLE", "GLOSSAR", and "IMPRESSUM".
- Search Bar:** A search icon on the right side of the navigation bar.
- Content Area:** The main content is the post "1. Übung: das Semesterblog" dated "4. OKTOBER 2015 / ALEXANDER THOMAS / KOMMENTAR HINTERLASSEN".
- Post Editor:** The "Beitrag bearbeiten" (Edit Post) screen is open, showing the post content and various publishing options like status, visibility, and SEO settings.
- Right Sidebar:** Features sections for "LETZTE BEITRÄGE", "Vorlesungsüberblick", "LETZTE KOMMENTARE", and "ARCHIVE".
- Footer:** A "Suche ..." (Search) input field at the bottom right.

## Übungen

# Beitrag im Semesterblog

## Ergebnisprotokoll einer Vorlesung

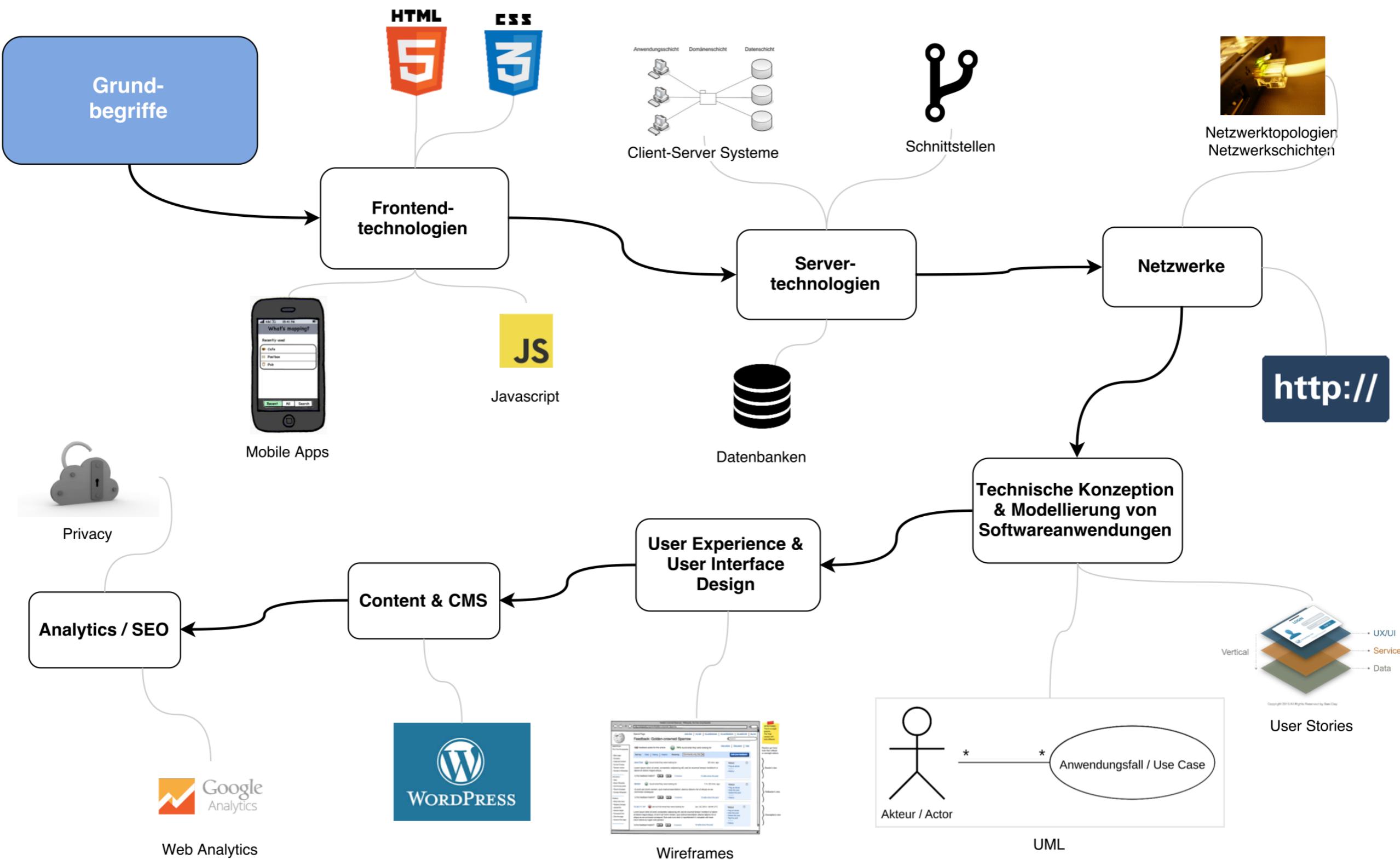
- Zur jeder **Vorlesung** erstellt ein Team von bis zu 3 Studierenden ein **Ergebnisprotokoll**
- Das Protokoll wird als **Artikel im Semesterblog** verfasst und dort veröffentlicht
- Ein Protokoll muss die wesentlichen Ergebnisse einer Vorlesung zusammenfassen und auf geeignete Quellen zur Vertiefung der behandelten Themen verweisen

# Beitrag im Semesterblog

## Begriffsdefinitionen im Glossar

- Das Semesterblog enthält ein Glossar, in dem wichtige Begriffe aus der Vorlesung in Form von kurzen Artikeln und Definitionen beschrieben werden können
- Alternativ zum Protokoll können Sie hierzu gerne ein paar (nicht aus dem Internet kopierte) Artikel beisteuern
- Kommen Sie dazu am besten auf mich zu, um geeignete Begriffe zu besprechen

# Themenüberblick



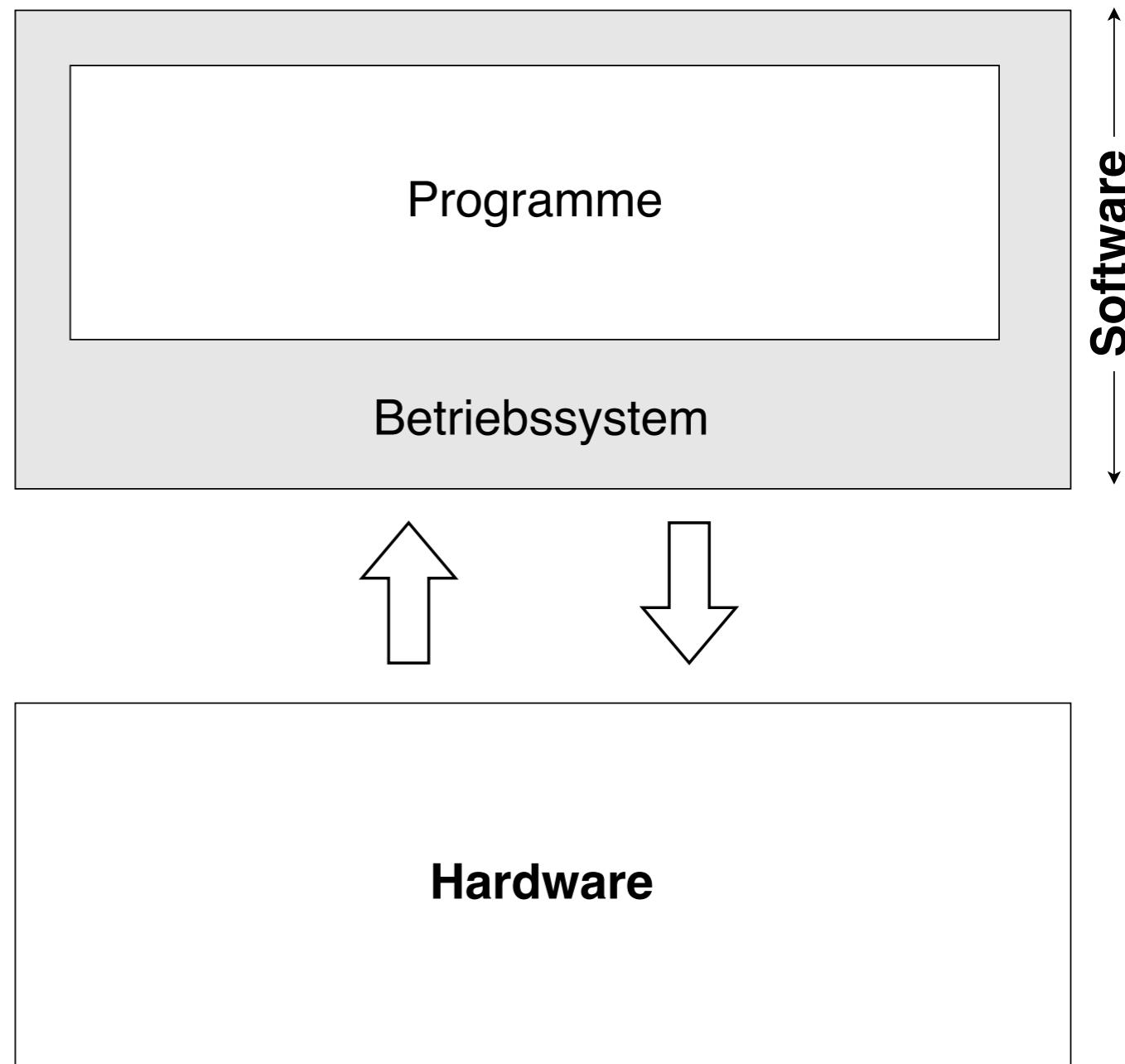
# Grundbegriffe



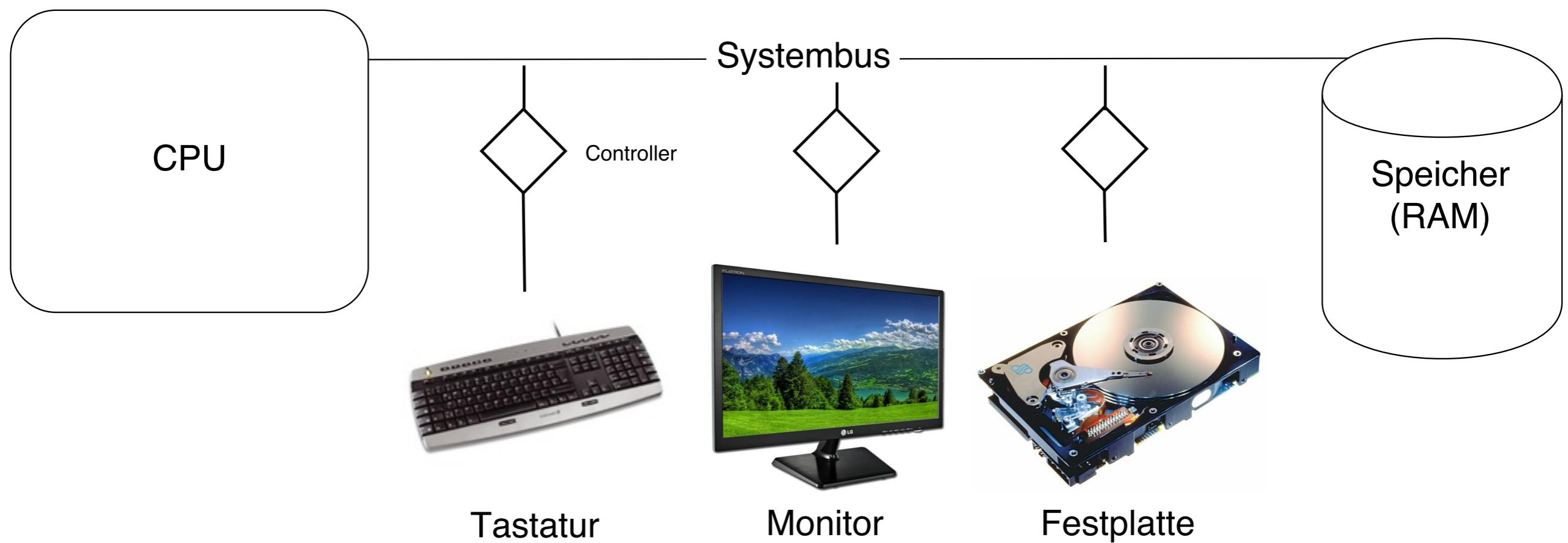
Was ist (alles) ein Computer?

# Computer

- Ein **Computer** ist eine Maschine, die dazu konstruiert wurde, Abfolgen von Anweisungen durchzuführen
- **Software** bezeichnet die Programme, in denen diese Abfolgen von Anweisungen hinterlegt sind.
- **Hardware** ist die Bezeichnung für die greifbaren Teile, aus denen Computer bestehen



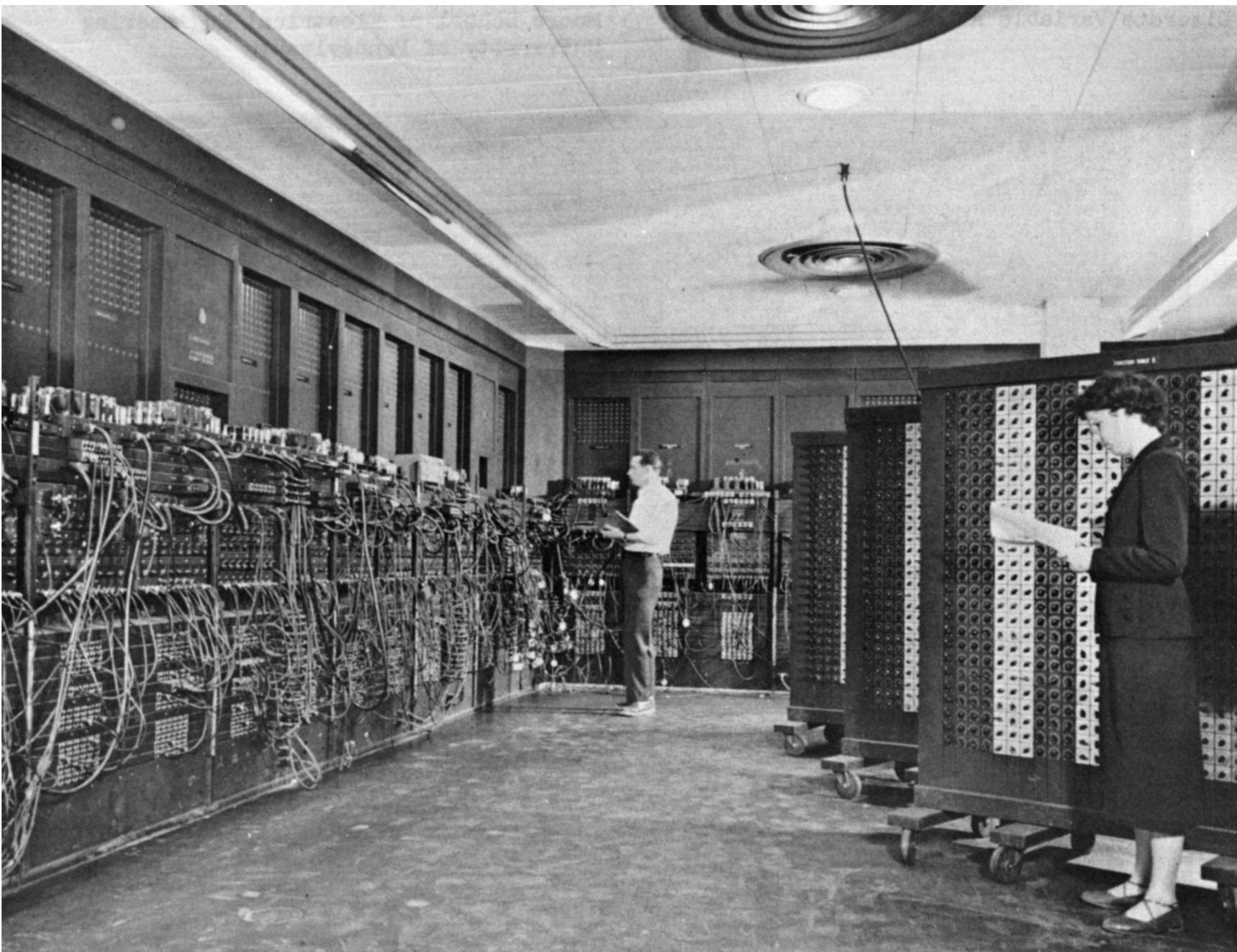
# Hardware – Genereller Aufbau



# Hardware – CPU

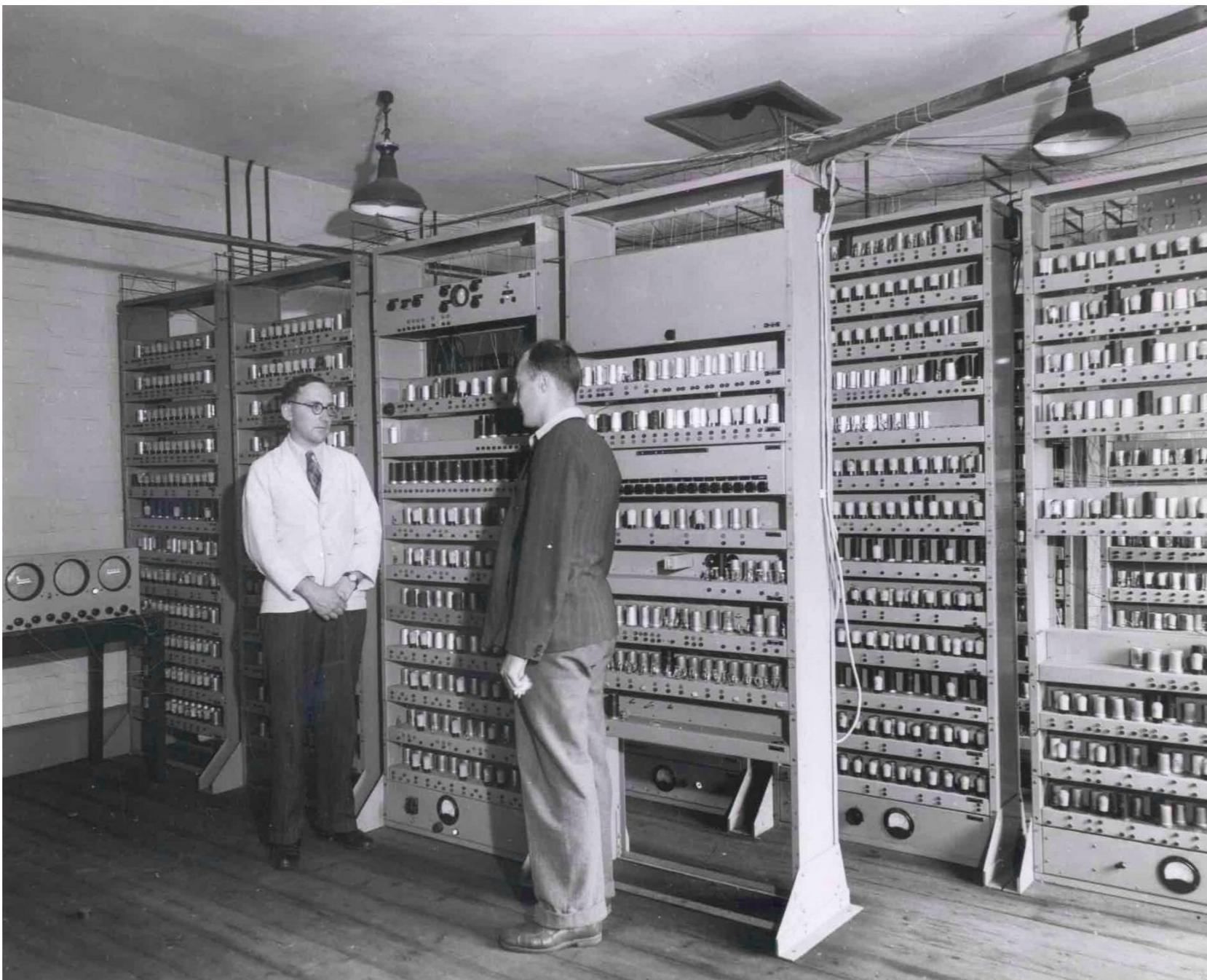
- Die **Central Processing Unit (CPU)** ist das Herzstück eines jeden Rechners und ist für die **Steuerung aller Abläufe** und die **Verarbeitung der Daten** zuständig.
- Das Funktionsprinzip wurde von den Mathematikern **John Von Neumann**, **J. Presper Eckert** und **John Mauchly** erdacht und in dem US-Militärcomputer EDSAC im Jahr 1949 erstmals realisiert.

# Hardware – CPU



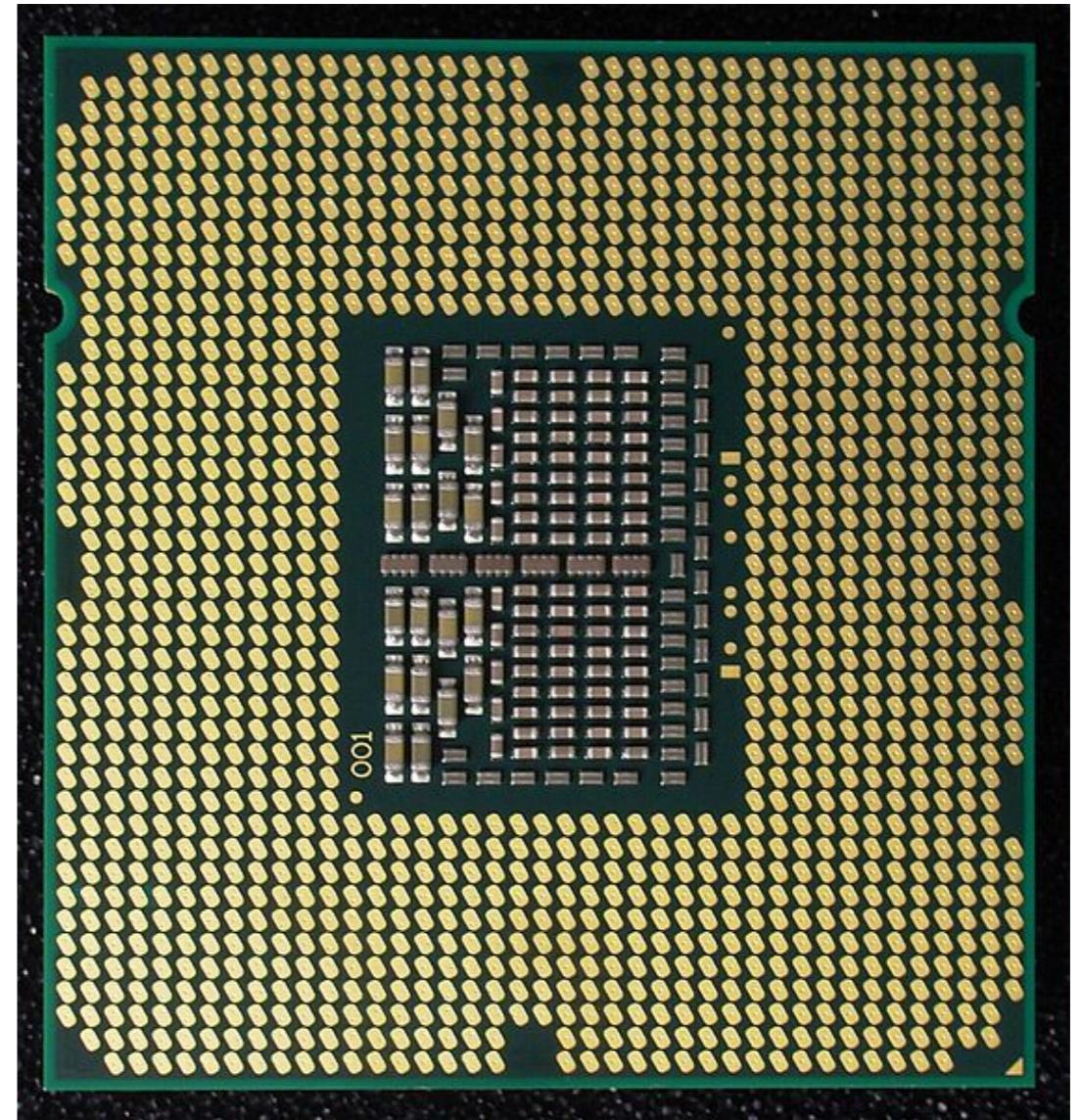
Eniac, 1946

# Hardware – CPU



EDSAC, 1949

# Hardware – CPU



# Hardware – CPU

## CPU (Central Processing Unit)

- Rechenwerk (Arithmetic Logic Unit - ALU)
- Steuerwerk (Control Unit - CU)
- Speicher (Memory)
- Ein-/Ausgabeeinheit (I/O-Unit)



# Hardware – CPU

- **Ansteuerung der Peripheriegeräte-Controller**  
Eingabedaten und Programmanweisungen werden entgegengenommen und die Ausgabe der berechneten Ergebnisse über den Systembus an die Controller der Peripheriegeräte (Display, Lautsprecher, Drucker ...) übertragen
- **Speicherzugriff**  
Schreiben und Lesen von Daten in den Hauptspeicher. Entweder zur Zwischenlagerung bis zur Abarbeitung oder zur Sicherung von Ergebnissen
- **Verarbeitung der Daten**  
Die Daten werden gemäß der Programmanweisungen durch *arithmetische Operationen* (Addition, Subtraktion, Multiplikation ...) und *logische Operationen* (AND, OR, XOR, ...) miteinander verknüpft

# Software

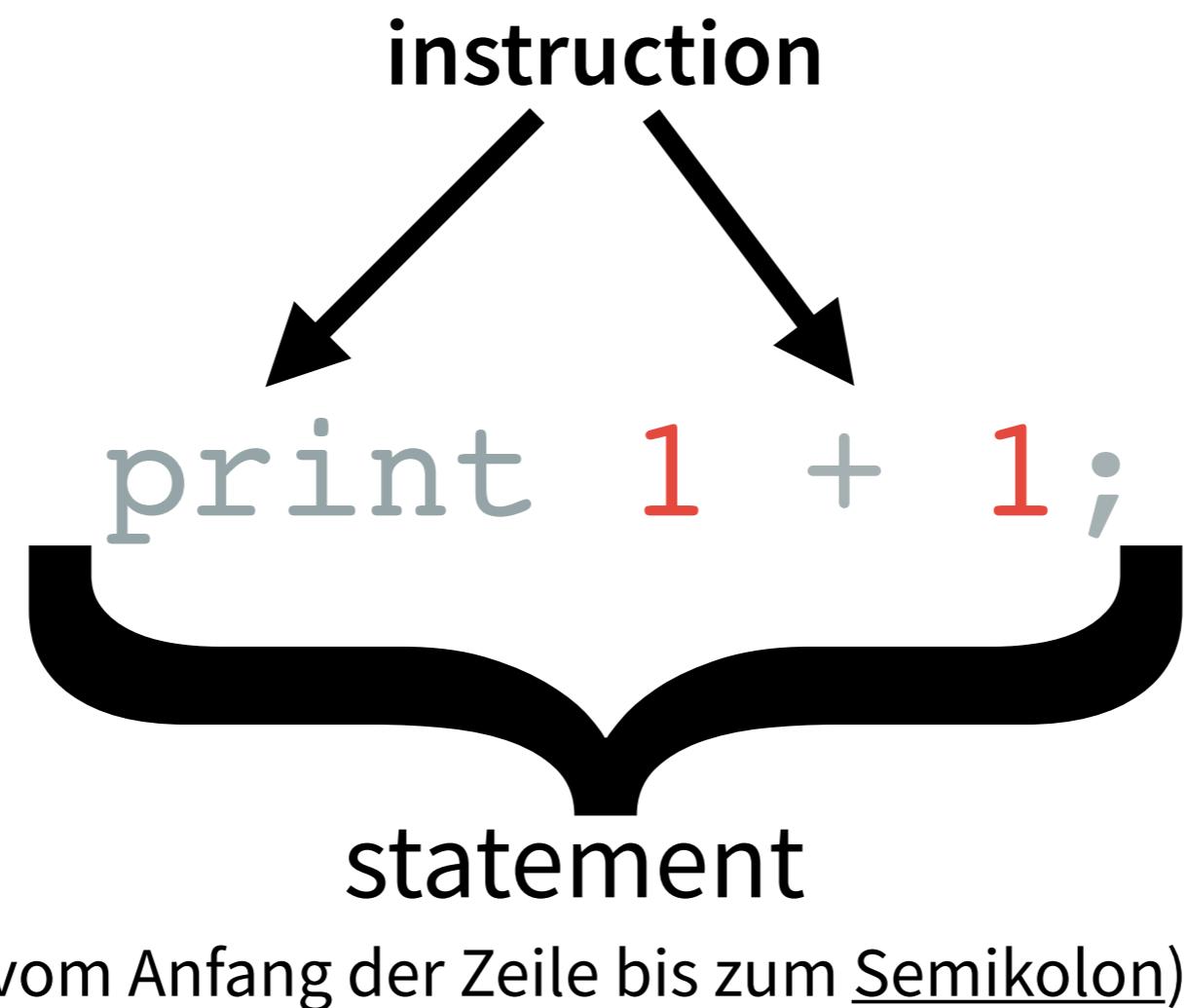
```
print 1 + 1;
```

# Software

- ISO/IEC 2382-1:1993 definiert „computer program“:

*„A syntactic unit that conforms to the rules of a particular programming language and that is composed of declarations and statements or instructions needed to solve a certain function, task, or problem.“*

# Software



# Software – Programmiersprachen

- Ruby:

```
puts "Hallo Welt!"
```

- Javascript:

```
console.log("Hallo Welt!");
```

- Java:

```
class Hallo {  
    public static void main( String[] args ) {  
        System.out.println("Hallo Welt!");  
    }  
}
```

# Software

- Initial waren Software und Hardware eine Einheit. Erst 1958 prägte der Statistiker John W. Tukey den Begriff Software.
- Eine allgemein gültige Definition gibt es bis heute nicht, da die Begriffe **Software**, **Programm** und **Daten** nicht trennscharf sind
- Die aktuelle ISO/IEC-Norm 27465 enthält für Software folgende Definitionen(!):
  - Software ist ein Programm oder eine **Menge von Programmen**, die dazu dienen, einen **Computer zu betreiben**.
  - Software sind **Programme** sowie die zugehörige **Dokumentation**.
  - Software sind **Programme** und ggf. die zugehörige **Dokumentation** und **weitere Daten**, die zum Betrieb eines Computers notwendig sind.

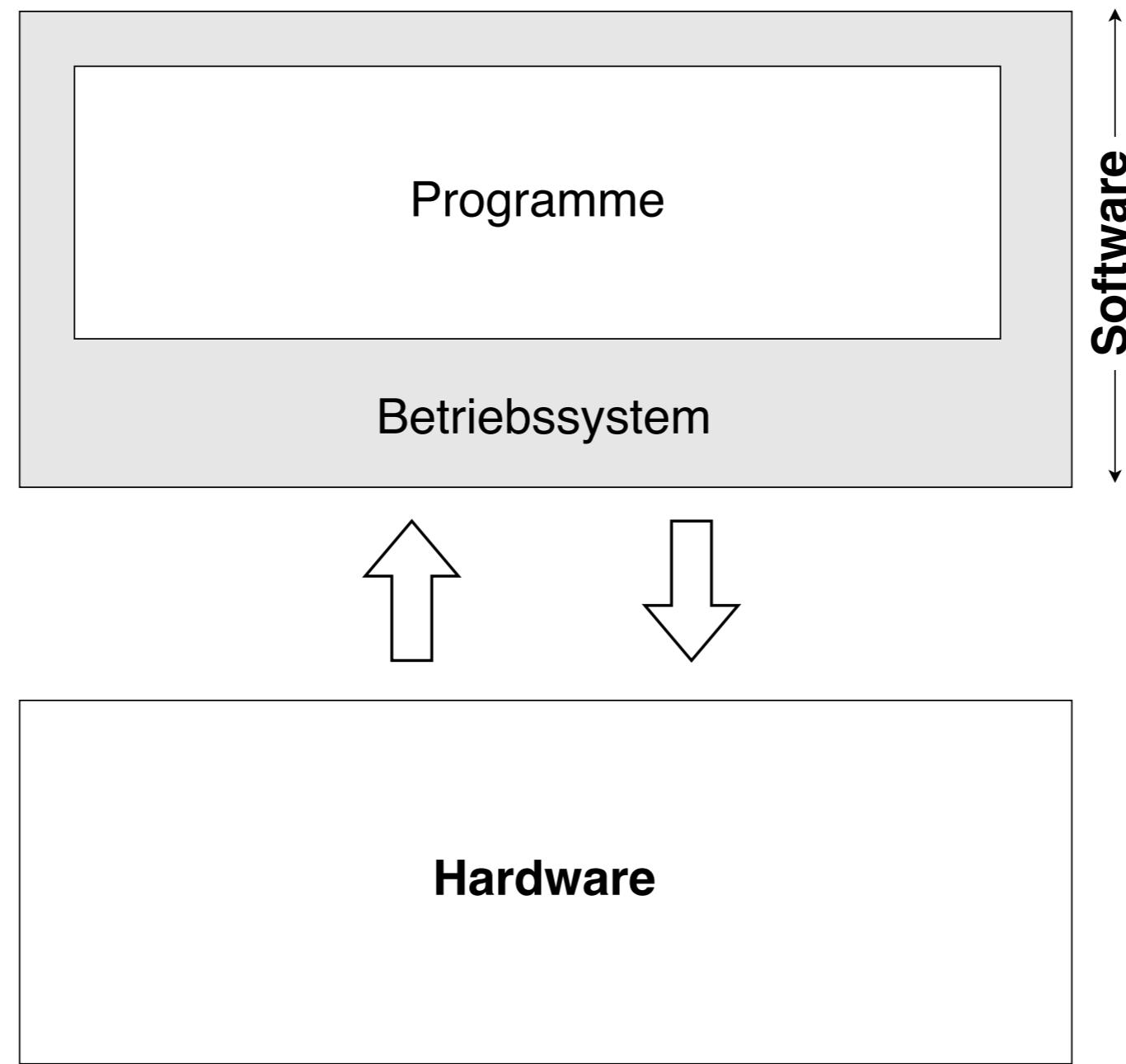
# Software – Programme

- Programme liegen als eine Folge von **binären Maschinenbefehlen** im Speicher. Nur diese Befehle können vom Prozessor verarbeitet werden.

10110000 01100001

- Vor der Ausführung eines Programms bedarf es daher eines **Übersetzungsprozesses** (Kompilierung oder Interpretation)
- Programmiersprachen werden auch als **Hochsprachen** bezeichnet, weil sie die eigentlichen Maschinenbefehle für den Menschen abstrahieren

# Software – Betriebssysteme



# Software – Betriebssysteme

- Ein **Betriebssystem** ist ein Stück Software, dass die **Kommunikation zwischen Hardware und Programmen** koordiniert und es so erlaubt Programme zur Ausführung zu bringen
- Es **verwaltet die Systemressourcen** eines Computers (Arbeitsspeicher, Festplatten, Ein- und Ausgabegeräte etc). D.h. es bestimmt welche **Programme** wie viel **Zugriff** auf Ressourcen wie den Speicher erhalten und wie lange
- Der zentrale Bestandteil ist der Betriebssystemkern (**Kernel**), der Programmen **hardware-abstrahierte Schnittstellen** für den Zugriff auf die Hardware bereitstellt

# Software – Betriebssystemkern

## Multitasking

- Ausführung **mehrerer unabhängiger Programme** auf einem Computer. Durch **Time-Sharing** wird der Eindruck erzeugt, dass die Prozesse gleichzeitig ablaufen, während sie in Wirklichkeit nacheinander abgearbeitet werden.
- Der Kernel enthält einen **Scheduler**, der bestimmt, wie viel lange jeder Prozess ausgeführt werden darf und in welcher Reihenfolge die Programme **Ausführungskontrolle** erhalten. Nur in diesem Zeitfenster erhält ein Programm Zugriff auf die CPU und den Speicher. Diese Kontrollübergabe wird als **Context Switch** bezeichnet.

# Software – Betriebssystemkern

## Memory management

- **Verwaltung des Systemspeichers** für Programme. Dies stellt sicher, dass ein Programm nicht auf Speicherbereiche zugreift, die bereits vergeben sind.
- Der Kernel bestimmt, **welchen Speicher** jedes Programm zu einem bestimmten Zeitpunkt zu verwenden darf. Auf diese Weise kann das Betriebssystem die gleichen Speicherplätze für mehrere Aufgaben verwenden.

### Swapping

Daten, auf die weniger häufig zugegriffen wird, können vom Betriebssystem vorübergehend **aus dem RAM ausgelagert** und auf der Festplatte gespeichert werden. Diese Technik wird als **Swapping** bezeichnet. Ein Speicherbereich kann so von mehreren Programmen benutzt werden, da sein Inhalt je nach Bedarf ausgetauscht wird.

# Software – Betriebssystemkern

## Festplattenzugriff und Dateisysteme

- Die eigentlich vorhandene **Blockstruktur** und die Unterschiede verschiedener Speichersysteme (Festplatten, DVD-Laufwerke etc) werden vom Betriebssystem durch **Dateisysteme** abstrahiert.
- Auf diese Weise können Programme Daten in Form von **Dateien** speichern, die in einem **Verzeichnisbaum** hierarchisch angeordnet werden können.

# Software – Betriebssystemkern

## Gerätetreiber

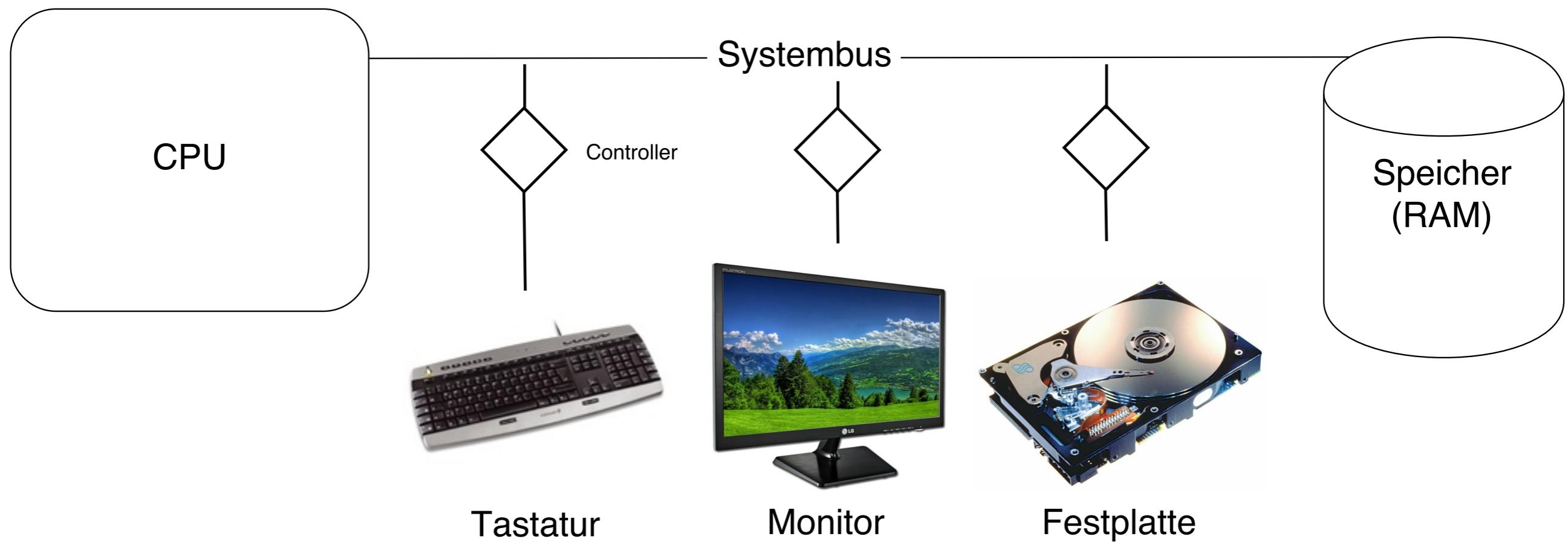
- Ein Gerätetreiber ist eine spezielle Art von Software, die eine **Interaktion zwischen Computer und einem externen Gerät** ermöglichen soll.
- Üblicherweise stellt ein solcher Treiber dem Betriebssystem **Schnittstellen** für die Kommunikation mit dem Gerät bereit, das an den Computer angeschlossen ist. Diese Schnittstelle erlaubt es Befehle und Daten an das Gerät zu senden bzw. zu empfangen.
- Gerätetreiber sind sowohl **hardwareabhängig**, als auch **betriebssystemspezifisch**. Das gilt sowohl für die Architektur, als auch für die Version des Betriebssystems.

# Software – Betriebssystemkern

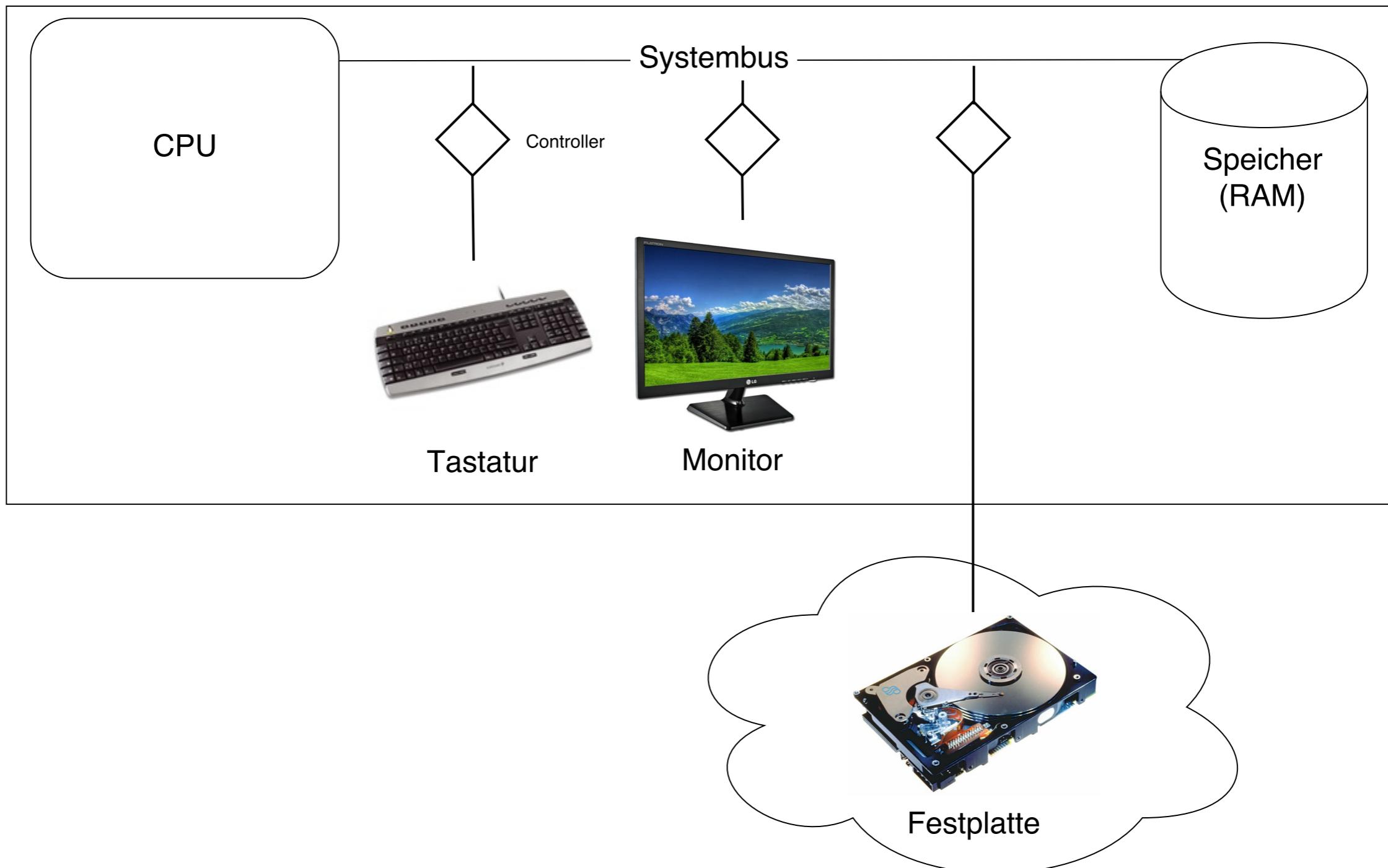
## Networking

- Betriebssysteme unterstützen eine Vielzahl von **Netzwerkprotokollen**, - hardware und Anwendungen. Auf diese Weise können Computer mit **unterschiedlichen Betriebssystemen** in einem **gemeinsamen Netzwerk** miteinander kommunizieren und Ressourcen teilen.
- Netzwerke erlauben es also einem Betriebssystem **auf entfernte Ressourcen** wie Scanner oder Festplatten **zuzugreifen** und diese nutzen, als wären sie direkt mit dem lokalen Computer verbunden.
- Dies beinhaltet alles vom Austausch einfacher Nachrichten über die Nutzung gemeinsamer Datenspeicher bis zur gemeinsamen Nutzung entfernter Grafik- oder Soundkarten.

# Software – Betriebssystemkern



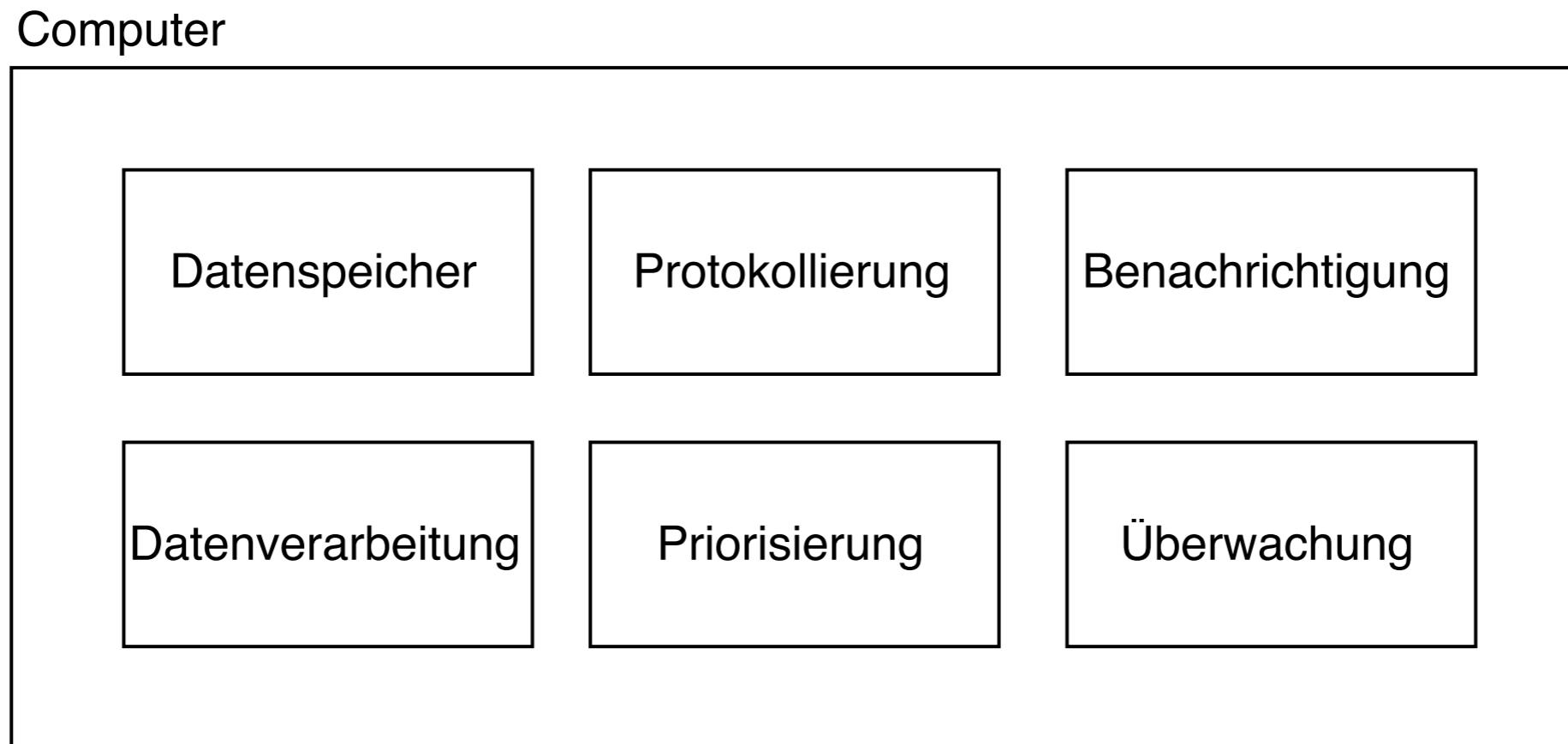
# Software – Betriebssystemkern



ISO/IEC-Norm 27465:

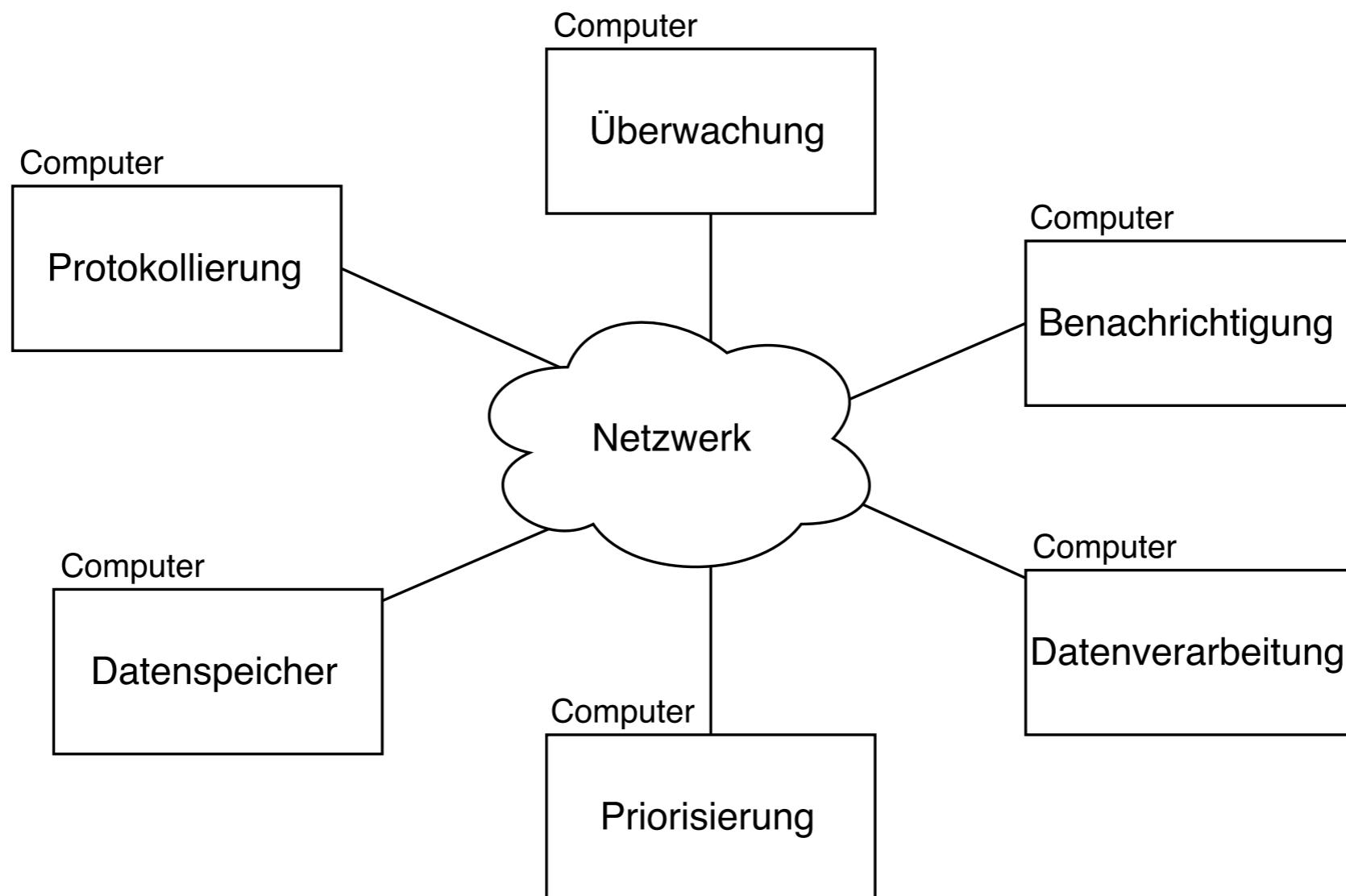
*„Software ist ein Programm oder  
eine Menge von Programmen.“*

# Client-Server-Architektur



Alle Programme laufen auf einem Computer

# Client-Server-Architektur

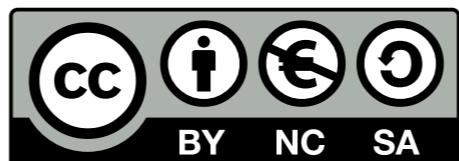


Computer stellen Programme als Dienst im Netzwerk zur Verfügung

# Client-Server-Architektur

- Eine Client / Server-Architektur ermöglicht es einem Programm, das auf einem Computer läuft, sich als so genannter **Client** über ein **Netzwerk** zu einem **entfernten Computer** zu verbinden, dem **Server**.
- Server bieten verschiedene **Dienste** (Services) für andere Netzwerkcomputer und Benutzer.
- Services werden über **Ports** und die **Netzwerkadresse** des Servers angesprochen
- Ein Programm auf dem Server, ein **Daemon**, nimmt die entfernten Anfragen entgegen und koordiniert die Kommunikation mit dem Betriebssystem-Kernel

# Vielen Dank.



Dieses Werk ist lizenziert unter einer Creative Commons 4.0 International Lizenz mit folgenden Eigenschaften:

- Namensnennung
- Nicht-kommerzielle Nutzung
- Weitergabe unter gleichen Bedingungen.

# Quellen

- <http://www.ch-r.de/et/nue-atmmk-mikroprozessoren.pdf>
- <https://srv.rz.uni-bayreuth.de/lehre/fortran90/vorlesung/V01/V01.html>
- <http://www.geeks.com/techtips/2005/techtips-NOV22-05.htm>