

**TP 2.2**

- Netzwerkkommunikation.PDF
  - Sterntopologie
    - Vorteile
      - Man kann, anders als bspw. bei der Ringtopologie einen Host austauschen ohne die anderen Hosts bei der Arbeit zu stören
      - Wenn eine Verbindung zu einem Host ausfällt stört dies die anderen nicht
    - Nachteile
      - Wenn der Hub in der Mitte ausfällt funktioniert gar nichts mehr
  - Access Point
    - Ist ein Gerät der über ein Kabel mit einem Router verbunden ist, und für Geräte ohne LAN-Anschluss ein „WLAN“ erstellt damit sie auch in das Netz kommen
  - Kabel
    - UTP(Unshielded Twisted-Pair) sind billiger und einfacher in der Anwendung wobei die Konkurrenten STP(Shieleded TP) und FTP(Foil TP) jedoch performanter und schwerer in der Installation sind.
  - Netzwerkgeräte
    - Kabelgebunden
      - **Netzwerkkarte:** Gerät zum Verbinden eines Gerätes mit einem lokalen Netzwerk
      - **Repeater:** Gerät zum Erhöhen der Reichweite eines WIFI Signals
      - **Hub:** Gerät welches als Knoten in einem Netzwerk mit Broadcasts arbeitet
      - **Bridge:** Verbindet zwei Netzwerk Segmente(welche das gleiche Protokoll verwenden) miteinander
      - **Switch:**
    - Internet
      - **Router:** Verbindet zwei oder mehrere verschiedene Netzwerke miteinander, schickt Pakete + Verbindung ins Internet
      - **Gateway:** Ist ein Knoten der als Eingang zu einem anderen Netz dient, für verschiedene Protokolle
  - 3 Arten von Hubs
    - **Passiv:** Senden einfach nur die Pakete weiter
    - **Aktiv:** Benötigen Strom um die ankommenden Signale zu verstärken und „aufzuräumen“
    - **Intelligent:** Sind den aktiven sehr ähnlich, haben zusätzlich einen Prozessor welcher Diagnose Informationen filtert, zum „Troubleshooten“
  - Was braucht ein Router:
    - **2x Netzwerkkarte:** Zum Verbinden der Netze
    - **OS:** Gibt Aufgaben an CPU
    - **CPU:** Führt die Aufgaben des OS aus
    - **RAM:** Flüchtiger Speicher für etwa Routingtabellen
    - **Flash:** Bspw. zum Speichern des OS
    - **NVRAM:** Nicht flüchtiger Speicher zum Speichern der Startkonfiguration
    - **BUS:** Wird verwendet zur Kommunikation zwischen CPU und Schnittstellen
    - **ROM:** Speichert die Diagnoseprogramme des Routers
    - **Schnittstellen:** Sind die Verbindungen auf welchen Pakete die Hauptplatine des Routers betreten und verlassen

- Client/ Server
  - Server
    - Bietet Dienste an, Clients können sich über das Netzwerk mit dem Server verbinden, er schickt die Antwort zurück oder einen Fehler
  - Client
    - Schickt die Anfrage an den Server, er bleibt inaktiv bis er die Antwort bekommt und wird aktiv wenn er eine Anfrage schickt
  - Programmierung
    - Man spricht hierbei von einer direktionalen Verbindung, sowohl der Client als auch Server können lesen und schreiben
    - Dafür muss sich der Client zunächst mit dem Server verbinden, dies macht er über den „ServerSocket“
    - Nachher schicken die beiden Komponenten über die Sockets ihre Nachrichten, diese werden in einem sogenannten Buffer zwischengespeichert.
  - Schritte bei der Kommunikation
    1. Server wartet auf Verbindung
    2. Client fragt beim Server an
    3. Server nimmt die Anfrage an und führt den angeforderten Dienst aus
    4. Client wartet auf Antwort
    5. Server schickt die Nachricht und schließt die Verbindung
    6. Client erhält die Antwort und schließt die Verbindung ebenfalls
    7. Server wartet auf Verbindung
  - Zusatzinformationen
    - Die Verbindung kann auch über andere Methoden die nicht teil des TCP/ IP Stacks sind stattfinden
    - Können „Verbindungslos“ oder „verbunden“ sein
    - Auf einzelne Nachrichten basieren oder kontinuierlich
    - Zuverlässig oder nicht zuverlässig sein
  - Begriffsbeschreibungen
    - **Socket**: Ist ein Verbindungspunkt, definiert Adresse, Port und Protokoll
    - **Bind**: Bindet einen Port an ein bestimmtes Programm
    - **Listen**: Wartet auf eine Verbindung von einem Client
    - **Accept**: Wartet auf eine „Connect Request“ und nimmt den Client an
    - **Connect**: Verbindet sich mit dem Server
    - **Send**: Schickt die Daten an Client oder Server
    - **Receive**: Bekommt die Daten von Client oder Server
    - **Close**: Client und Server schließen die Verbindung
  - Java.net
    - TCP → ServerSocket und Socket
    - UDP → DatagramPacket und DatagramSocket
    - http → URL und URLConnection
  - TCP
    - Verbindungsorientiert → Baut eine Verbindung auf
    - Zuverlässig → Garantiert Zustellung der Daten
    - Full-Duplex → Man kann über gleichen Kanal senden und empfangen
    - Point-To-Point → Ein Sender ein Empfänger
    - Stellt sicher dass die Pakete ankommen
    - Stellt sicher dass die Pakete in der richtigen Reihenfolge ankommen

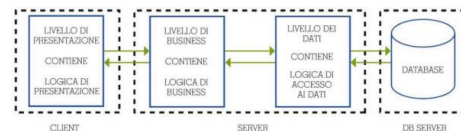
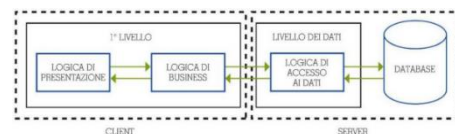
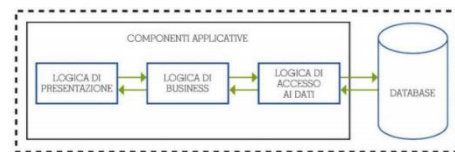
- UDP
    - Connectionless → Stellt keine Verbindung her
    - Pakete können verloren gehen
    - Pakete können in der falschen Reihenfolge ankommen
    - Verwendet multiplexing → Werden zusammengefasst und gleichzeitig gesendet
    - Verbindung mit dem Server muss nicht dauerhaft gehalten werden
    - Es können auch kleiner Pakete geschickt werden
    -
  - HTTP
    - Arbeitet über das World Wide Web
    - Anfrage besteht aus dem Protokoll bspw. http
    - Der zweite Teil sagt welche Ressource er sich holen soll, Dokument, Bild
    - Und den Ort wo er sich dies holt
- 

- Serverstandorte
  - Server Tower:
    - Für kleine bis mittlere Unternehmen
    - Funktioniert selbständig/ „standalone“
    - Verwendung:
      - Virtualisierung
      - Management der Daten und Webseite
    - Nachteile: Groß und Laut
  - Server Rack: Kompakter, besser Skalierbar,
  - Server blade: Am kleinsten, Gut für Virtualisierung und Speicherung
  - Data Center
    - Große Einrichtungen zum Speichern von Daten
    - Alle Daten wie Facebook Konnte, Emails usw. werden hier gespeichert
    - Viel Firmen haben ihre eigenes Data Center
    - Gruppe von Servern: Server-Farm
    - Vorteil: Zentralisierung der Sicherheit, Instandhaltung und Management
  - Vorteile eines **Firmeninternen** Data-Center:
    - Physisch direkt
    - Isoliert vom „Gefährlichen“ Internet
  - Nachteile eines **Firmeninternen** Data-Center:
    - Benötigt Platz
    - Man muss die Hardware usw. kaufen
    - Muss diese immer wieder upgraden
  - Server auslagern:
    - Diese Anlagen sind oft abgeschottet
    - Man kann von der Erfahrung der Angestellten dort profitieren
    - Keine Instandhaltung und Anschaffungskosten
  - Server-Farm:
    - Eigenschaften dieser:
      - Physische Sicherheit
      - Redundanz
      - Garantiert niedrige Temperatur
      - Sichere Internetverbindung
      - Sicherheit durch Firewalls und andere Sicherheitsvorkehrungen
      - Keine Probleme mit der Internetverbindung

- Was bietet eine Server-Farm an
  - Hosting:
    - Man kann dort bspw. seine Webseite hosten lassen
    - Gratis oder zahlen:
      - Einschränkung in Speicher, Performance, Funktionen usw.
  - Serverhousing:
    - Hier stellt der Kunde die Hardware bereit, diese wird lediglich im Rechenzentrum aufgestellt und instandgehalten
    - Vorteile:
      - Platzsparen
      - Muss nicht mieten
      - Kann auch selber administrieren
    - Nachteile:
      - Weit weg – nicht direkt im Haus
      - Startkosten beim Kauf
      - Upgrades müssen selber bezahlt werden
  - Dedizierte Server
    - Server der nur für eine Sache konzipiert ist und deshalb am besten läuft und die Ressourcen gezielt einsetzen kann
    - Vorteile:
      - Alles übernehmen Profis
      - Keine hohen Startkosten
    - Nachteile:
      - Monatliche Kosten
      - Weit weg – nicht direkt im Haus
  - Virtuelle Server
    - Keine eigener Server der nur eine Aufgabe übernimmt, sondern ein Teil des Servers übernimmt eine Aufgabe, ein anderer Teil eine andere Aufgabe
    - Vorteile:
      - Niedrige Kosten
      - Hohe Sicherheit
      - Leistungsstarke Software
    - Nachteile:
      - Mehrere Personen auf dem Server
        - Könnten Probleme machen
      - Hardware, CPU; RAM usw. muss geteilt werden
- Server-Virtualisierung
  - Da heutzutage die Server nicht mehr so schnell ausgelastet sind, jedoch für die meisten Sachen ein eigener Server verwendet werden muss, wurde die Server-Virtualisierung erfunden.
  - Der Client meint dabei er hat einen eigenen Server da er alleine auf dem OS ist, ist er aber nicht da es mehrere OS gibt
  - Es gibt dabei eine sogenannte Abstraktionsschicht welche dies ermöglicht
  - Grob gesagt laufen auf einer Hardware mehrere OS, und auf jedem OS die Anwendung für welche der Client den Server mietet, dies spart dem Anbieter als auch dem Client Geld, Performance und Speicher
  - Vorteile:
    - Startkosten werden gesenkt
    - Niedrigerer Energieverbrauch
    - Dynamische Ressourcenverteilung
    - Einfacheres Testen und Troubleshooten

- Nachteile:
  - Backup Probleme mit den verschiedenen OSs
- Hyper-V
  - Presentation Virtualisierung
    - Virtualisierung der Darstellung
  - Application Virtualisierung
    - Ist eine Softwaretechnologie, die die Anwendung vom ausführenden Betriebssystem isoliert. Auch wenn die Anwendung noch auf dem OS installiert ist, wird sie nicht mehr dort direkt ausgeführt. Die Anwendung wird auf einer virtuellen Umgebung ausgeführt.
  - Desktop Virtualisierung
    - Die Desktop-Virtualisierung ist ein Verfahren, um mehreren Benutzern gleichzeitig und unabhängig voneinander die Ausführung von Anwendungsprogrammen auf einem entfernten Computer zu erlauben
  - Storage Virtualisierung
    - Unter Speicher- oder Storage-Virtualisierung versteht man das Zusammenlegen physikalischen Speichers aus mehreren Netzwerkspeichersystemen; aus Sicht des Administrators existiert dann ein einziger, von einer Zentralkonsole aus gesteuerter Speicher.
  - Network Virtualisierung
    - Die Netzwerkvirtualisierung ist eine Methode, um die verfügbaren Ressourcen eines Netzwerks zu kombinieren, indem man die verfügbare Bandbreite in mehrere unabhängige Kanäle aufteilt, von denen jeder einem bestimmten Server oder einem Gerät in Echtzeit zugewiesen werden kann. Jeder Kanal ist unabhängig abgesichert. Alle Teilnehmer haben von einem Computer aus gemeinsamen Zugriff auf alle Ressourcen des Netzwerks.
- OS Virtualisierung
  - Dabei läuft eine Software auf einem PC und diesem OS welche auf diesem OS ein anderes OS simuliert
  - Bspw. VirtualBox oder VMWare
  - Man benötigt dazu eine ISO – Ein Speicherabbild einer CD oder DVD in welcher das OS gespeichert ist
  - Die Performance hängt dabei von den zugewiesenen Ressourcen(RAM usw.) ab
- Anwendungsvirtualisierung
  - Vorteile:
    - Wird verwendet wenn man nicht ein gesamtes OS Virtualisieren möchte
    - Mehrere Instanzen des gleichen Programms können installiert werden
    - Wird nicht direkt auf OS installiert
    - Deshalb kann man es jederzeit aktivieren und deaktivieren
    - Vorteile:
      - Zentrale Steuerung
      - Sicherung der Programme
      - Sicherung der zusammenhängen Daten
      - Einfache Upgrades
      - Zugriffssteuerung und Überwachung

- o4\_Webanwendungen:
  - Schichtenmodell → layer = tier
    - Unabhängige Schichten:
      - Präsentationsschicht
        - Stellt Daten dar und nimmt Input von Nutzer an
      - Geschäftslogik
        - Hier befindet sich die Logik entweder an den Benutzer nach oben weiter zu geben und diese auch von unten zu verarbeiten
      - Datenzugriffsschicht
        - Hier befinden sich nun effektiv die Daten welche an den Nutzer geschickt werden
  - Trennung in Schichten
    - Ohne Trennung der Schichten
      - Keine Trennung der Schichten alles in einem Programm
    - Client/ Server
      - Verarbeitung und Darstellung getrennt von der Speicherung
    - 3-tier-Architektur
      - Die Darstellung der Daten wird von der Logik abgetrennt
  - Multi-tier-Anwendungen
    - **Networking:** Software die sich um die Netzwerkverbindung kümmert
    - **Security:** Alles was mit Sicherheit der Daten und System zu tun hat
    - **Data-Storage:** Die Verwaltung der Datenspeicherung
    - **User Interface:** Die Oberfläche mit welcher der Benutzer arbeitet
    - Vorteile:
      - Können auf einem Rechner laufen aber auch aufgeteilt werden
      - System ist flexibel und leicht zu verwalten, austauschbar
      - Die Aufgaben können besser verteilt werden
  - 3-tier-Architektur für Webseiten
    - **Präsentationsschicht – Front End:** Stellt der Browser statisch oder dynamisch dar
    - **Geschäftslogik – Business Logik:** Dynamischer Inhalt wird von einem Application Server generiert, verarbeitet und ans Front End geschickt
    - **Datenzugriffsschicht – Back End:** Enthält Daten und Verwaltung der Daten



- SOA – Service-Oriented Architecture
  - Benötigt immer komplexere Programme mit meist Anbindung an Netzwerk und hohen Anforderungen an Sicherheit
  - Wenn nun Clients eine Funktionalität oder ein Service über öffentliche Schnittstellen abfragen, spricht man von Service-orientierter Architektur (SOA).
  - Komponenten:
    - **Service:** Enthält die Businesslogik und stellt die Schnittstelle zur Verfügung
    - **Registry:** Veröffentlicht die Schnittstelle damit Clients diese finden
    - **Client:** Finden den Service über Registry und greifen über Interface zu
  - Vorteil ist diese Anwendungen sind **loosely-coupled**:
    - Unabhängig von Speicherort, Programmiersprache und Hardware
- Application Server
  - Damit Web Applications ausgeführt werden können benötigt man einen Application Server – Business tier
  - Eigenschaften:
    - **Explizit:** Kapselung, Zugang zu Datenbanken
    - **Implizit:** Skalierbar, Monitor, Sicherheit, Logging
  - Beispiele:
    - **Java Application Server:** Enthält alles aus Java EE
    - **Microsoft .NET Framework:** Enthält Aufgaben des Application Server
- MVC
  - Probleme beim entwickeln von Software:
    - **Aufbau und Test:** Wie erstellt und welche Technologien
    - **Wiederverwendbarkeit:** Können Sachen wiederverwendet werden
    - **Skalierbarkeit:** Kommt die Anwendung mit vielen Anfragen zurecht
    - **Sicherheit:** Schützen vor Angriffen
    - **Verschiedene Views auf Daten:** Verwendung von Zugangsbeschränkungen und Authentifizierungsmaßnahmen
  - Design Pattern
    - Template immer wieder verwendet kann → Anleitung kein Produkt
      - Problemstellung
      - Kontext
      - Beste Lösung
      - Fragen die bei der Verwendung aufkommen können
    - Vorteile: erprobt, Wiederverwendbar, liefern Lösungen, Erfahrung
  - Probleme bei Projektierung
    - Ändern des Aussehen(auf verschiedenen Geräten/ Views/ Eingabe → Tastatur Touchscreen) ohne Änderung Logik
  - Lösungen
    - Trennung Logik u. Darstellung, Versch. Views, Versch. Clients
  - Komponenten MVC
    - Komponenten
      - **Model:** Stellt Daten und Aktivitäten dar
      - **View:** Darstellung des Zustands
      - **Controller:** Ändert Model, gibt Daten von Model an View
    - Macht graphische Schnittstellen skalierbar ohne Logik zu verändern
    - Anwendung bei Webentwicklung
      - Model Sprache – HTML in der die Seite geschrieben ist
      - View Stylesheets – Wie wird die Seite dargestellt
      - Controller Browser – Stellt Daten aus HTML nach CSS dar

- MVVM
  - Variante von MVC
    - Model:
      - Datenzugriffsschicht für die Inhalte, die vom Benutzer angezeigt und manipuliert werden können. Es beinhaltet die gesamte Geschäftslogik
    - View:
      - Alles, was durch die GUI dargestellt wird. Die View bindet sich an Eigenschaften des
      - ViewModel, um Inhalte darzustellen und zu manipulieren
    - ViewModel:
      - Beinhaltet die UI-Logik (Model der View) und ist Bindeglied zwischen View und Model. Es ruft Methoden oder Dienste auf und stellt der View Eigenschaften und Befehle zur Verfügung. Das ViewModel darf keine Kenntnis der View besitzen.

