

INFO SHEET

Communication Entities

- **System-orientierte** und **Problem-orientierte** Sichtweise
- **Prozesse:** intuitive Systemsicht, führt zu der weit verbreiteten Darstellung von verteilten Systemen als Menge von Prozessen, die über IPC-Mechanismen interagieren
 - sehr kleine Systemen ohne ein Prozess-Konzept
- **Objekte:** Problem-orientierte Sicht — in einem verteilten Objekt-basierten System bestehen Anwendungen aus einer Menge verteilter, interagierender Objekte
 - Objekte mit mehreren Threads
- **Komponenten:** Weiterentwicklung des Objekt-Konzepts hin zu Software-Bausteinen
- **Web Services:** ähnlich zu Komponenten, aber explizit auf das WWW als Kommunikationsplattform ausgerichtet

Communication Paradigm

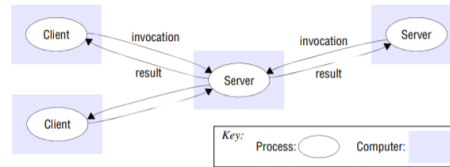
- **Interprocess Communication**
- **Remote Invocation**
 - **Request-Reply Protocols**
 - **Remote Procedure Calls**
 - **Remote Method Invocation**
- **Indirect Communication**
 - **Group Communication**
 - **Publish-Subscriber System**
 - **Message Queues**
 - **Tuple Spaces**
 - **Distributed Shared Memory**

Roles and Responsibilities

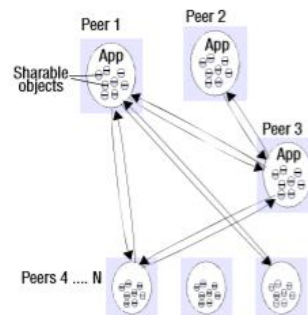
- **Client-Server Architektur**
- Client: Prozess, der auf Daten zugreifen, Ressourcen verwenden oder Vorgänge/Operationen auf einem anderen Computer ausführen möchte
- Server: Prozessverwaltung von Daten und allen anderen gemeinsam genutzten Ressourcen zwischen Servern und Clients, Ermöglichung des Clientzugriffs auf Ressourcen
- Interaktion: anfrage- / Antwortnachrichtenpaare (request/reply message pairs)

Architecture

- Client fordert Dienste auf Server an z.B. will er www.xx.de in IP umwandeln lassen und fordert hierzu den DNS-Dienst auf dem Server an



- **Peer-to-Peer-Architektur**
- Ein P2P-System ist ein dezentrales System mit gleichartigen Anwendungen
- - Prozesse spielen eine ähnliche Rolle bei der kooperativen Interaktion wie bei Peers
- Verwenden Sie Ressourcen auf vielen teilnehmenden Computern, um eine bestimmte Aufgabe in einer Anwendung zu erfüllen
- Eine große Anzahl von Datenobjekten wird gemeinsam genutzt, während ein einzelner Computer nur einen kleinen Teil davon enthält
- Objekte werden aus Gründen der Lastverteilung und Ausfallsicherheit auf mehreren Computern repliziert
- Komplexer als die Client-Server-Architektur



Placement

- **Zuordnung von Objekten oder Diensten zur zugrunde liegenden physischen verteilten Infrastruktur**
- **Mapping of services to multiple servers**
- **Caching**
- **Mobile code**
- **Mobile agents**

Architectural Patterns

- **Layering**
- jede Ebene nutzt die Dienste der darunterliegenden Ebene
- **Tiered architecture**
- Abgestufte Architekturen ergänzen die Überlagerung.
- **Thin clients**
- Verlagerung der Komplexität vom Endgerät des Benutzers auf Dienste im Internet