

Verteilung & Kommunikation

Begriffe

Begriffe	Beschreibung
Verteiltes System	System, in dem sich Hardware- und Softwarekomponenten auf vernetzten Computern befinden und miteinander über den Austausch von Nachrichten kommunizieren.
Verteilte Anwendung	Nutzt ein verteiltes System als Kommunikationsinfrastruktur für ihre verteilten Komponenten.
Middleware	Vermittlungssoftware bezeichnet in der Informatik anwendungsneutrale Programme, die so zwischen Anwendungen vermitteln, dass die Komplexität dieser Applikationen und ihre Infrastruktur verborgen werden.

Middleware-Kategorien

- Kommunikationsorientiert (Teil des VSK-Projekts)
- Nachrichtenorientiert
- Anwendungsorientiert (versch. Anwendung auf versch. Plattformen zusammenbringen)

Kommunikationsmodelle

Synchrone Kommunikation:

Beide Partner müssen da sein.

Die Kommunikationspartner (Prozesse) sind beim Senden oder beim Empfangen von Daten immer synchronisiert, also warten (blockieren), bis die Kommunikation abgeschlossen ist.

Asynchrone Kommunikation:

Ich werfe Brief in Briefkasten.

Das Senden und Empfangen von Daten ist zeitlich versetzt und findet ohne Blockieren des Prozesses statt (z.B. durch Warten auf die Antwort des Empfängers). Der Empfänger muss nicht einmal aktiv sein.

Transparenz

Transparenz = Unsichtbarkeit

- **Ortstransparenz:** Ort von Dienst/Ressourcen dem Benutzer nicht bekannt.
- **Zugriffstransparenz:** Art des Zugriffs auf Ressource immer gleich.
- **Nebenläufigkeitstransparenz:** Zugriff mehrerer Benutzer auf Dienste/Ressourcen gleichzeitig möglich.
- **Fehler- und Ausfalltransparenz:** Anwendung erfährt nichts von Fehler.
- **Sprachtransparenz:** Kommunikation zwischen Komponenten unabhängig von Programmiersprache.
- **Replikationstransparenz:**
Aus Performancegründen kann es mehrere Kopien derselben Ressource geben. Das System sorgt für die transparente Replikation der darin vorgenommenen Änderungen.

Architekturmodelle

Client-Server: langlebender Server-, kurzlebende Client-Prozesse.

Peer-To-Peer: Gleichberechtigte Prozesse laufen lokal und tauschen nur bei Bedarf untereinander Informationen aus. Es wird kein zentraler Prozess benötigt.

Fat-Client:

Eigentliche Verarbeitung vor Ort. Stellt (meistens) grafische Benutzeroberfläche zur Verfügung.

Thin-Client:

Stark auf Hilfe anderer Computer angewiesen.

■ Zu einem Fat-Client paart sich meistens auch ein Thin-Client (vice-versa)

2-Tier:

Präsentation, Anwendungslogik und Datenhaltung auf zwei Anwendungen verteilt.

3-Tier:

Anwendungslogik erhält eigenen Tier.

n-Tier:

Verteilung der Anwendungslogik und Datenhaltung auf mehrere Tiers