

Aufgabe 16

	Sequentielle bzw. parallele/verteilte/nebenläufige Verarbeitung	Ein- bzw. Mehrbenutzerfähigkeit	Zuordnung der Schichten zu Client bzw. Server	Notwendigkeit unterschiedlicher Plattformen	Wiederverwendung vorhandener Bausteine
a) Heißluftföhn	sequentiell - vermutlich geringe Komplexität des Programms und geringe Anforderung an Performance - Realisierung z.B. mittels Mikrocontroller ohne Betriebssystem -> ein sequentielles Programm zur Steuerung ausreichend.	einbenutzerfähig - übliche Benutzungsweise eines Heißluftföhns	eine Schicht - lediglich Softwaresteuerung des Heißluftföhns	keine unterschiedlichen Plattformen - folgt aus der Definition "Embedded System" -> Steuerungssoftware wird speziell für das Gerät (bspw. Den enthaltenen Mikrocontroller) geschrieben	Wiederverwendung von Softwarebausteinen ist nebensächlich - neue Softwaresteuerung speziell für das Gerät - relativ geringe Komplexität der Softwaresteuerung ist zu erwarten.
b) Pizza-Konfigurations App	parallel/verteilt/nebenläufig - aus der Beschreibung folgt das Aspekte der App dem Client-Server-Modell folgen und damit bereits verteilt sind und Programmteile z.B während der Synchronisation nebenläufig sein können	mehrbenutzerfähig - App wird zwar von einem Besitzer des jeweiligen Smartphones genutzt, allerdings werden Kundenkonten für die Verwaltung der erstellten Pizzen benötigt -> Mehrere Kunden Nutzen die Datensynchronisation	Client: Präsentations-, Logik- und Teile der Datenhaltungsschicht Server: Datenhaltungsschicht - Präsentation ist die GUI der installierten App - "nur beim synchronisieren ist eine Internetverbindung notwendig" -> die Funktion der App ist nicht als Webservice gedacht. Daher ist die Logik zur Erstellung und Speicherung der selbst-konfigurierten Pizzen als Programmlogik auf dem jeweiligen Smartphone zu verstehen. Dazu zählt aber auch eine gewisse Datenverwaltung/Datenspeicherung der eigenen Pizzen. - Benutzerkonten und sämtliche Pizzen aller User müssen auf dem Server gespeichert werden	ja - alle gängigen Smartphones sollen unterstützt werden -> iOS und Android Versionen müssen unterstützt werden (verschiedene Betriebssysteme und genutzte Programmiersprachen) - Server-seitige Software für entsprechendes Serversystem (bspw. Linux Distribution).	ja - GUI bibliotheken der verschiedenen Plattformen der client-seitigen Software - 5G Kommunikation - Server Software (z.B. Apache) - Datenbank-Software (z.B. SQL)
c) Pizza-Besatzsystem	parallel/verteilt/nebenläufig - Wie in Beispiel b) handelt es sich um ein Client-Server-System. Client ist hier einmal der Automat, der mit dem Datenbankserver kommuniziert, aber auch ein Webclient (Webpage) zur Registrierung der Kunden der wiederum den Datenbankserver mit den Kundendaten versorgt - weiterhin die Speicherung der Kunden ID auf der Karte.	mehrbenutzerfähig - Datenverwaltung von mehreren Benutzerkonten - allgemein nutzbare Webpage zur Registrierung - Automaten nutzbar für registrierte Kunden	- Präsentationsschicht: webpage GUI - Logikschicht: 1) Logik zur Registrierung der Kunden nach eingabe der Kundendaten (z.B. PHP Serverprogramm zur Interaktion mit Datenbank) 2) Logik in Automaten (Steuerung von Kartenleser, Kommunikation mit Serversoftware für Kundendaten) - Datenverwaltung: serverseitige Datenbank, KundenID auf Karte	ja - In der Hand des Herstellers liegen die Plattformen der Server, Automaten und Karten. Für alle müssen verschiedenste Programmteile entwickelt werden - Zur Webpage Realisierung müssen die zu unterstützten Webbrowser spezifiziert werden -> Möglicherweise Portierung für verschieden Webbrowser-Engines (bspw. chromium-based <-> Firefox-Engine)	ja - Datenbank-Software (z.B. SQL) + Verschlüsselungstechnologien - Server Software - Aus der Aufgabenbeschreibung ist zu entnehmen dass die eigentlichen Automaten bereits existieren

	Notwendigkeit und Gestaltung eines Hilfesystems	Verwendung und Auswahl eines DBMS	Realisierung der Benutzerschnittstelle	Rückgriff auf sonstige Dienstleistungen
a) Heißluftföhn	kein Hilfesystem - User wird nicht mit der Steuerungssoftware direkt interagieren	kein DBMS - Steuerung wird keine Daten verwalten	keine Benutzerschnittstelle - User wird nicht mit der Steuerungssoftware direkt interagieren - Steuerung impliziert die Interaktion von User und dem Gerät und ist damit gewissermaßen selbst Schnittstelle zwischen Usereingabe (bspw. Anschaltknopf, Regler und funktionsweise des Geräts)	keine weiteren Dienstleistungen nötig - Dienstleistung als Steuerungssoftware steht eigenständig
b) Pizza-Konfigurations App	nein - Präsentationsschicht sollte den üblichen App-Funktionalitäten entsprechen und daher möglichst intuitiv bedienbar sein - Softwareseitig implementiertes Bedientutorial denkbar	relational - Pizza Objekte lassen sich gut auf die relationale Tabellenstruktur mappen - Keine Notwendigkeit für Vererbungen Assoziationen - relational meist bessere Leistung als objektorientiert (Auswahl: z.B. SQL)	- aus Beschreibung der Aufgabe und vorherigen Überlegungen folgt, dass die App GUI die Schnittstelle für User-Interaktionen darstellt	- evtl Serverbereitstellung und -wartung
c) Pizza-Besatzsystem	ja mögliche Realisierung durch Hypertext der Webpage	relational - feste Form des Datensatzes lässt sich sehr gut als Tabelleninformation darstellen - gute, einfach zu nutzende Verschlüsselungstechnologien sind vorhanden	- Erste Schnittstelle in Form der Webpage GUI - Weiter Schnittstelle: Karte -> Kartenleser als Schnittstell zum Automaten/Server/Bezahllogik	- evtl. Serverbereitstellung und -wartung - evtl. Möglichkeit zur Änderung des Kundenkontos - evtl. Möglichkeit zur Anforderung einer neuen Karte