

Tabelle1

Allgemein Bei allen Themen ist eine geeignete Anzahl von Datensätzen und Clients zu definieren, um eine Aussage über Skalierung und Performance treffen zu können! UI sollen Funktional sein und gegenüber dem Nutzer einen definierten Transparenzgrad erfüllen. An das Design werden keine Anforderungen gestellt.						
Nr	Titel	Beschreibung	Client-Anteil:	Cluster-Anteil	Anforderungen	Bewertungs-schwerpunkte
1	Entwicklung eines verteilten Chatsystems	Es soll ein Chat-Programm erstellt werden, damit die Studenten Textnachrichten austauschen können.	Benutzeroberfläche und Logik für die Anbindung an das verteilte Backend. Strategien zum Senden von Nachrichten an nicht vorhandene Teilnehmer.	Skalierbarer Chat-Service zur Nachrichtenvermittlung.	20+ Teilnehmer, maximale spürbare Verzögerung beim Absenden 1s, Empfang: keine Anforderung	Analyse von Latenzzeiten, Durchsatz und Skalierbarkeit.
2	Aufbau eines verteilten Dateisystems	Um Daten nicht zu verlieren und Teilen zu können soll eine Anwendung erstellt werden, die es ermöglicht die essentiellen Aktionen (Schreiben, Lesen, Löschen) auf Dateien durchzuführen.	Anwendung zur Dateiverwaltung mit einfacher Oberfläche / Übersicht der vorhandenen Dateien	Definition und Logik für Speicherung, Replikation und Konsistenz der Dateien.	100+ Dateien, maximale spürbare Verzögerung beim Speichern 1s.	Effizienz der Datenreplikation und Umgang mit Netzwerkpartitionen.
3	Entwicklung eines verteilten Spiel-Servers	Es soll ein verteiltes Spiel entwickelt werden, bei dem die Spieler als „Kugel“ ein Rennen gegeneinander fahren. Die Rennstrecken soll dabei zufällige Hindernisse enthalten. Bei Berührung muss der Spieler eine definierte Zeit stehen bleiben.	Spiel-Client für Multiplayer-Spiele. UI, die das Spielfeld darstellt.	Server-Logik zur Synchronisation von Spielzuständen.	10+ Spieler, „ruckelfreie“ Spieldarstellung	Skalierbarkeit und Fairness der Spielerinteraktionen.
4	Aufbau eines verteilten Datenanalyse-Systems	Dem Cluster werden Daten zur Analyse (Primzahlen) gegeben. Das Cluster berechnet, ob es sich um eine Primzahl handelt oder nicht.	Benutzeroberfläche für Datenanfragen.	System zur verteilten Datenverarbeitung.	1000+ Anfragen pro Stunde, maximale spürbare Verzögerung beim Anfragen und Rückmelden einer Antwort 1s.	Verarbeitungszeiten und Skalierbarkeit mit wachsenden Datenmengen.
5	Realisierung eines verteilten Blockchain-basierten Systems	Um Transaktionen zu verwalten soll ein Blockchainansatz gewählt werden. Ein SHA256 Ansatz reicht.	Wallet-Anwendung für Transaktionen.	Blockchain-Netzwerk zur Transaktionsverteilung und -verifikation.	100+ Nutzer, Transaktion soll innerhalb von 500ms gespeichert werden. Bestätigung darf bis zu 10s dauern.	Sicherheit, Transparenz und Skalierbarkeit.
6	Implementierung eines verteilten Ereignisbenachrichtigungssystems	Bei Katastrophen sollen alle Nutzer mindestens 1 mal möglichst schnell über die eine Katastrophe informiert werden.	Anwendung für Ereignisregistrierung und -benachrichtigungen.	Infrastruktur zur Ereignisverarbeitung und -verteilung.	maximale Verzögerung bei Bekanntgabe einer Katastrophe von 10s.	Komplette Informationsverteilung
7	Entwicklung eines verteilten Systems zur Bilderkennung	Es sollen Bilder auf Fahrzeuge überprüft werden. Zudem soll die Farbe des Fahrzeugs kategorisiert werden.	Schnittstelle zum Hochladen von Bildern.	Verteiltes System zur effizienten Bilderkennung.	100+ gleichzeitige Anfragen.	Antwortzeiten und Genauigkeit der Bilderkennung.
8	Aufbau eines verteilten Wettervorhersagesystems (3 Studenten)	Simulierte Wetterstationen schicken Wetterdaten an das Cluster. Dieses verarbeitet es und bestimmt das Wetter von morgen als Durchschnittliches Wetter von heute im Umkreis von 50km.	Anwendung zur Abfrage von Wettervorhersagen / Simulation der Wetterstationen	System zur Wetterdatenanalyse und -vorhersage.	1000+ Wetterstationen, 100+ Nutzer, maximale Antwortzeit 2s.	Aktualität und Genauigkeit der Wettervorhersagen.
9	Verteiltes Lernsystem mit adaptiver Schwierigkeitsanpassung	Ein Lernsystem sammelt die Ergebnisse aller Nutzer und bestimmt daraus die nächsten Fragen aufgrund des durchschnittlichen Schwierigkeitsgrades.	Benutzeroberfläche für die Interaktion mit Lernmaterial.	Algorithmen zur Anpassung des Lernpfades.	1000+ Nutzer; 1000+ Fragen; maximale Verzögerung 2s.	Adaptivität und Benutzerengagement.
10	Verteiltes Reservierungssystem für öffentliche Einrichtungen	Die Hallenbelegung aller Hallen einer mittleren Großstadt soll verwaltet werden. Dabei soll ein Nutzer, der häufig eine Halle mietet erkannt werden, und Vorrang haben.	Anwendung zur Anzeige der Verfügbarkeit und Durchführung von Reservierungen.	Verwaltung von Reservierungen und Verfügbarkeiten.	100+ Hallen, 1000+ parallele Nutzer	Konfliktvermeidung und Fairness.
11	Verteiltes Überwachungssystem für Smart-City-Infrastrukturen	Die Smart-City soll die Wegplanung der Fahrzeuge optimieren.	Fahrzeugsimulation mit Senden von Positionsdaten und Darstellung der von der Smart-City geplanten Wegführung.	Sammlung und Verarbeitung von Fahrzeugdaten. Berechnung des optimalen Weges. Sobald Fahrzeuge kollidieren ist von einem Stopp an der Stelle auszugehen.	1000+ Fahrzeuge, maximal 2s Verzögerung auf Streckenplanungsrequest.	Effizienz der Streckenanalyse und Trendanalyse.
12	Verteilte Plattform für Echtzeit-Kollaboration	Bei einer Echtzeitkollaborationsplattform ist es essenziell, dass Video und Audio verteilt werden.	Tool für das Anzeigen von Video/Audio Daten.	Backend-Logik für Echtzeitsynchronisation unter Berücksichtigung des definierten QOS.	maximale Anzahl von „Rucklern“: 1 pro Stunde; Maximale Verzögerung 1s bei 10 Nutzern.	Latenz der Datensynchronisation und Benutzererfahrung.
13	Verteiltes System zur Auswertung von Feedback für Seiten und Produkte (3 Studenten)	Basierend auf einem Webshop sollen Produktbewertungen nach Produktkategorien analysiert werden. Dabei sollen Stichworte der Produkte berücksichtigt werden.	Benutzerschnittstelle für Suchanfragen und Anzeige von Ergebnissen / Datencrawler (Client Anteil)	Datencrawler (Clusteranteil). Datenpipelines und Algorithmen für Analysen.	100+ Nutzer, maximale Verzögerung bei Anfragen von 2s.	Präzision der Analyse und Skalierbarkeit des Systems.

Tabelle1

14	Überwachungskamera-Videos	Überwachungskamera-Videos sollen durch mehrere Analyse-System in Echtzeit auf unterschiedliche Objekte analysiert werden.	Upload/Anbindung von Video-Streams und Analyse-Pattern (Begriff)	Verteilung der Analyse-Aufgaben an entsprechendes Pattern-System	10+ parallele Streams; 5 unterschiedliche Pattern.	Verteilung der Analyse; Skalierbarkeit bei Streams
15	Music around the world	An unterschiedlichen Teilen der Welt soll es Musikern möglich sein, zusammen zu Musizieren.	Synchroner Audio Input/Output	Verteilen und Synchronisation der Daten +	5+ Musiker, maximale Verzögerung 200ms	minimale Latenz
16	Passworthilfe	Verlorene Passwörter sollen aus Passwort-Hashes (SHA1) mittels Brute-Force bestimmt werden.	Upload eines Passwort-Hashes und Darstellung des Passworts	Verteiltes Berechnen des Passworts.	maximale Verzögerung 1s bei 100+ parallelen Requests.	Antwortzeiten und Korrektheit.
17	Schlüsseldienst	Die Primzahlen eines RSA-Schlüssels (public key) sollen bestimmt werden (Zahlenpool soll gegeben sein).	Generierung eines RSA-Keys (64bit) durch 2 Primzahlen (aus Zahlenpool)	verteilt Berechnen des Private Keys mit gegebenen Zahlenpool.	100+ parallelen Requests;	Korrektheit; kein Requestverlust.