Konzeption – Entwicklung eines real-time Backends für eine datenintensive Applikation

Seminararbeit

Prüfungsleistung für den

Master of Science

des Studiengangs Data Science

an der Internationalen Hochschule

von

Jasper Bremenkamp

8. Oktober 2024

Kursbezeichnung	Projekt: Data Engineering
Matrikelnummer	92125193
Tutor	Prof. Dr. Max Pumperla

1 Szenario

Die FreshMart AG, eine Kette von Lebensmittelgeschäften, steht vor der Herausforderung, eine **datengesteuerte Entscheidungsfindung** in ihre Betriebsprozesse zu integrieren, um die Effizienz zu steigern, Kosten zu senken und die Kundenzufriedenheit zu verbessern. Die bestehende Infrastruktur, die aus Kassensystemen, Lagerverwaltungssystemen und Kundendatenbanken besteht, wird derzeit noch hauptsächlich zur Erfassung und Speicherung von Daten genutzt, jedoch nicht für fortschrittliche Analysen oder Echtzeit-Entscheidungen.

Das Ziel des Projekts ist es, eine Dateninfrastruktur zu entwickeln, die in der Lage ist, kontinuierlich große Mengen an Echtzeitdaten zu verarbeiten und darauf basierend automatisierte Prozesse zu ermöglichen. Der Schwerpunkt liegt auf der **automatisierten Bestandsverwaltung**, dynamischen Preisgestaltung und personalisierten Kundenangeboten. Die Automatisierung soll sicherstellen, dass Nachbestellungen optimiert werden, indem Verkaufsdaten, Lagerbestände und saisonale Schwankungen berücksichtigt werden.

Die Kassensysteme in den Filialen erfassen kontinuierlich Transaktionsdaten, die durch eine zentrale Datenverarbeitungsarchitektur aggregiert und in Echtzeit analysiert werden sollen. Die wichtigsten Datenquellen umfassen Verkaufsdaten, Lagerbestandsdaten und CRM-Daten, die bereits in den Systemen vorhanden sind. Diese Daten werden genutzt, um eine **automatisierte Nachbestellung** auszulösen, sobald bestimmte Bestandsgrenzen erreicht werden, und um dynamische Preisanpassungen basierend auf der Nachfrage durchzuführen.

Der Kern des Systems ist ein Echtzeit-Reporting, das es dem Management ermöglicht, sowohl auf Filial- als auch auf Unternehmensniveau Einblicke in die aktuellen Lagerbestände, Verkaufszahlen und Nachfrageentwicklungen zu erhalten. Zusätzlich sollen Machine Learning-Algorithmen genutzt werden, um **Verkaufsprognosen** zu erstellen und darauf basierend gezielte Maßnahmen zu ergreifen.

Die neue Infrastruktur wird vollständig in die bestehenden Systeme integriert und schrittweise in allen Filialen implementiert. Während der Einführungsphase wird der Fokus darauf liegen, sicherzustellen, dass das System stabil arbeitet und zuverlässige Entscheidungen in Echtzeit getroffen werden können. Neben den technischen Herausforderungen müssen auch **datenschutzrechtliche Anforderungen** berücksichtigt werden, insbesondere im Hinblick auf die Verarbeitung von Kundendaten. Die Einwilligung zur Nutzung von Daten wird vorausgesetzt und in Übereinstimmung mit den geltenden Datenschutzvorgaben, insbesondere der DSGVO, gehandhabt.

Abschließend verfolgt die FreshMart AG mit dieser neuen Dateninfrastruktur das Ziel, ihre internen Prozesse zu optimieren, indem Entscheidungsfindungen schneller und effizienter gestaltet werden, gleichzeitig die Lagerkosten gesenkt und die Kundenzufriedenheit durch individuellere Angebote erhöht wird.

2 Strategie

Um die beschriebenen Ziele der FreshMart AG zu erreichen und eine effiziente Datenverarbeitungsarchitektur zu entwickeln, wird eine klar definierte und mehrstufige Strategie verfolgt. Diese Strategie legt die Grundprinzipien fest, die das Projekt leiten, und beschreibt den geplanten Entwicklungsprozess von der Konzeptionsphase bis zur vollständigen Implementierung des Systems.

2.1 Modulare und skalierbare Architektur

Ein zentrales Prinzip der Strategie ist der Aufbau einer **modularen und skalierbaren Architektur**, die auf einer Microservice-Architektur basiert. Durch die Verwendung von Kubernetes und Helm als **Infrastructure as Code (IaC)**-Werkzeuge wird die Bereitstellung und Verwaltung der Infrastruktur automatisiert, was die Skalierbarkeit und Wartbarkeit des Systems erheblich verbessert. Kubernetes ermöglicht das Orchestrieren der Microservices, um deren Skalierung und Fehlertoleranz zu gewährleisten.

Für die Containerisierung wird **Podman** anstelle von Docker verwendet, da es einige Vorteile bietet, wie z.B. Rootless Containers, die eine höhere Sicherheit ermöglichen. Podman kann nahtlos in die Kubernetes-Umgebung integriert werden und unterstützt die reibungslose Erstellung, Verwaltung und Verteilung der Microservices.

Die Architektur teilt das System in unabhängige Komponenten auf, wie die Datenaufnahme (data ingestion) und -verarbeitung, die von **Apache Kafka** als Nachrichten-Streaming-Plattform übernommen wird. Diese Modularität erleichtert die Skalierbarkeit des Systems, da einzelne Services nach Bedarf skaliert oder aktualisiert werden können, ohne dass die Stabilität des Gesamtsystems gefährdet wird.

2.2 Schrittweise Einführung und Testing

Die Implementierung erfolgt **schrittweise** in mehreren Phasen, um Risiken zu minimieren und eine kontinuierliche Verbesserung zu gewährleisten:

- **Phase 1: Konzeptionsphase** – In dieser Phase wird eine gründliche Analyse der bestehenden Systeme und der erforderlichen Technologien durchgeführt. Neben der Auswahl von Apache Kafka für die Datenaufnahme und Streaming wird die Infrastruktur mithilfe von Kubernetes und Helm definiert. Git in Verbindung mit der Plattform GitHub wird für die **Versionskontrolle** genutzt, um sicherzustellen, dass alle Änderungen am Code und an der Infrastruktur nachvollziehbar und reproduzierbar sind.
- **Phase 2: Implementierung und Integration** – Die eigentliche Entwicklung der Microservices beginnt, einschließlich der Integration der Datenquellen wie Kassensysteme und Lagerverwaltung. Die ersten Services werden in Containern mithilfe von Podman entwickelt und in Kubernetes-Clustern bereitgestellt. Die Konfiguration und Bereitstellung der Infrastruktur erfolgt automatisiert durch Helm-Charts.
- **Phase 3: Skalierung und Optimierung** – In dieser Phase wird das System auf größere Datenmengen skaliert und in den produktiven Betrieb überführt. Dabei steht die Optimierung der Datenströme und die Feinjustierung der Algorithmen im Vordergrund, um Echtzeit-Entscheidungen und Vorhersagen zu verbessern.

2.3 Automatisierte Bestandsverwaltung

Ein zentraler Aspekt des Systems ist die **automatisierte Bestandsverwaltung**. Das System wird kontinuierlich Verkaufs- und Lagerbestandsdaten über Apache Kafka sammeln und in Echtzeit verarbeiten. Sobald festgelegte Bestandsgrenzen erreicht werden, werden automatische Nachbestellungen ausgelöst. Durch die Modularität der Architektur kann dieser Service unabhängig skaliert und optimiert werden.

Die schrittweise Einführung dieser automatisierten Bestandsverwaltung beginnt in Pilotfilialen, um die Effizienz und Reaktionsfähigkeit auf saisonale und regionale Unterschiede zu testen. Durch den Einsatz von Kubernetes wird eine flexible Skalierung ermöglicht, um die steigenden Datenanforderungen in den verschiedenen Phasen des Rollouts zu bewältigen.

2.4 Datenbasierte Entscheidungsfindung

Die **datengestützte Entscheidungsfindung** wird durch Echtzeit-Datenanalysen ermöglicht, die es erlauben, auf aktuelle Entwicklungen in den Filialen zu reagieren. Mithilfe von Machine Learning-Algorithmen werden **Verkaufsprognosen** erstellt, die dynamische Preisanpassungen und personalisierte Angebote unterstützen. Diese Algorithmen werden kontinuierlich mit den neuesten Verkaufs- und Lagerdaten trainiert, die über Kafka in das System eingespeist werden. Die Fähigkeit, in Echtzeit auf sich ändernde Daten zu reagieren, ist dabei der Schlüssel, um genaue Vorhersagen und fundierte Geschäftsentscheidungen zu ermöglichen.

Die Implementierung der Machine Learning-Algorithmen wird allerdings nicht im Rahmen dieser Arbeit behandelt, um den Rahmen nicht zu sprengen. Im Fokus der Arbeit steht das Data Engineering.

2.5 Integration und Datenquellen

Die nahtlose Integration bestehender Systeme ist ein weiterer strategischer Fokus. FreshMart AG verfügt bereits über Kassensysteme und CRM-Systeme, die in die neue Dateninfrastruktur eingebunden werden sollen. Apache Kafka fungiert hierbei als zentrale Plattform für die Datenaufnahme, die eine konsistente und skalierbare Verarbeitung von Daten aus verschiedenen Quellen ermöglicht.

Diese Datenquellen werden schrittweise in die Infrastruktur integriert, um sicherzustellen, dass das System stabil arbeitet und die Datenströme effizient verarbeitet werden können.

2.6 Datensicherheit und Datenschutz

Ein zentraler Bestandteil der Strategie ist die Berücksichtigung von **Datensicherheit** und **Datenschutz**. Alle Kundendaten werden gemäß den Vorgaben der **Datenschutz-Grundverordnung (DSGVO)** verarbeitet. Die Speicherung und Verarbeitung der Daten erfolgt verschlüsselt, und es werden strenge Zugangskontrollen implementiert, um unbefugten Zugriff zu verhindern.

Um die Einhaltung der Datenschutzanforderungen zu gewährleisten, werden Sicherheitsmechanismen wie **verschlüsselte Datenübertragungen** und **regelmäßige Sicherheitsaudits** in die Infrastruktur integriert. Zudem sorgt das **Data Governance**-Framework dafür, dass der Umgang mit Daten klar geregelt und nachvollziehbar ist.

2.7 Zusammenarbeit und Versionskontrolle

Die enge Zusammenarbeit zwischen den internen IT-Teams der FreshMart AG und externen Experten ist ein entscheidender Erfolgsfaktor für das Projekt. Um die Arbeit effizient zu koordinieren und sicherzustellen, dass alle Änderungen nachvollziehbar sind, wird Git in Verbindung mit **GitHub** als Plattform für die Versionskontrolle verwendet. Dies ermöglicht eine klare Rückverfolgbarkeit der Änderungen und die kollaborative Entwicklung in verteilten Teams.

Regelmäßige Feedback-Schleifen und Tests während des gesamten Entwicklungsprozesses stellen sicher, dass das System kontinuierlich verbessert und an die spezifischen Anforderungen angepasst wird.

2.8 Schlussfolgerung

Die Strategie zur Entwicklung der neuen Dateninfrastruktur für FreshMart AG basiert auf einer modularen, skalierbaren Architektur, die den Einsatz von Kubernetes, Helm und Podman zur Sicherstellung der Skalierbarkeit, Wartbarkeit und Verfügbarkeit umfasst. Durch die Verwendung von Apache Kafka als zentrale Plattform für die Datenaufnahme und -verarbeitung sowie Machine Learning für datenbasierte Vorhersagen werden die internen Abläufe effizienter gestaltet. Gleichzeitig wird höchsten Wert auf Datensicherheit und Datenschutz gelegt, um den gesetzlichen Anforderungen gerecht zu werden.