Data Transformation & Analysis Challenge

Kontext:

Unser Kunde, ein Medienhaus, ist interessiert an aktuellen Trends auf Wikipedia und historischen Veränderungen von Themen. Dafür sollen perspektivisch alle vorgenommenen Änderungen der Wikipedia gespeichert, verarbeitet und Ergebnisse in einem Dashboard dargestellt werden.

Geplant ist ein System, welches die Änderungs-Events mit Hilfe von RabbitMQ bereitstellt und in eine passende Datenbank schreibt, um dort mit einer passenden Query Engine Analysen auszuführen.

Aufgabe:

Als Prototyp soll eine RabbtiqMQ Instanz aufgesetzt werden, um mögliche Szenarien zu testen und zu demonstrieren. Im Rahmen des Prototyps sollen folgende Komponenten programmiert werden:

- Ein Producer, der die Beispieldaten einliest und in zufälligem Abstand zwischen 0-1 Sekunde emittiert. Verwende einen für die Aufgabe sinnvollen Exchange.
- in RabbitMQ Consumer, der diese Daten von einer Queue liest, die folgenden Aggregationen vornimmt und die Ergebnisse abspeichert:
    - Globale Anzahl Edits pro Minute
    - Anzahl Edits der deutschen Wikipedia pro Minute

Die Umsetzung der Komponenten kann in einer Sprache deiner Wahl erfolgen.
Die Abgabe sollte die folgenden Artefakte umfassen:

- Den lauffähigen Prototypen in einem Git Repository
- Ein passendes docker-compose.yml File
- Eine Readme-Datei mit den notwendigen Setup- und Ausführungsschritten
- Der Code sollte eine sinnvolle Testabdeckung implementieren

Neben dem lauffähigen Code sind für uns folgende Aspekte interessant:

- Was wäre eine mögliche Datenbank zur Speicherung der Daten? (Vor-/ Nachteile)
- Welches Datenmodell wäre deiner Meinung nach sinnvoll zur Ablage der Events?
- Welches Exchange Modell wären sinnvoll? Beschreibe Vor-/Nachteile des von dir gewählten Exchanges in Hinblick auf Skalierbarkeit und Fehlertoleranz.

Der zeitliche Umfang für die Umsetzung sollte nicht wesentlich mehr als 6 Stunden betragen.

Viel Erfolg!