Projektarbeit: OpenWeatherMap-Streaming

Datum: 26.05.2025
Autor: Philippe Christen
Weiterbildung: CAS Data Engineering an der FHNW

Der vollständige Quellcode inklusive Docker-Konfiguration und einer Beispieldatei «.env.example» ist im zugehörigen Git-Repository verfügbar. Sensiblen Konfigurationswerte (z. B. API-Schlüssel) sind darin bewusst nicht enthalten und müssen vom Benutzer lokal in einer .env-Datei ergänzt werden. Die Umgebungsvariable für den Zugriff auf die OpenWeatherMap-API ist dabei wie folgt zu definieren: API_KEY="DEIN_API_KEY"

Problemstellung (Ist-Zustand)

Im Kontext moderner IoT- und Smart-City-Anwendungen besteht ein wachsender Bedarf, Wetterdaten in nahezu Echtzeit zu analysieren. Die OpenWeatherMap-API erlaubt das periodische Abfragen von Wetterdaten. Es fehlt jedoch an standardisierten, containerisierten Pipelines zur robusten Verarbeitung und Speicherung dieser Datenströme. Herausforderungen ergeben sich insbesondere durch das asynchrone Eintreffen der Daten (verschiedene Eventzeiten je Stadt), das Handling unvollständiger Datensätze sowie die Notwendigkeit, Datenverluste und Duplikate zu vermeiden. Zudem besteht ein Bedarf, meteorologische Auffälligkeiten automatisiert zu erkennen und diese für spätere Analysen bereitzustellen.

Ein zusätzlicher Aspekt betraf die Limitierung und Pflege der externen Datenquelle: OpenWeatherMap definiert im kostenfreien API-Zugang (Free Access Plan) klare Nutzungsbeschränkungen. Neben einem Monatslimit von 1.000.000 API-Aufrufen dürfen:

- pro Standort maximal ein API-Aufruf alle 10 Minuten erfolgen (entspricht der Aktualisierungsfrequenz des OpenWeather-Modells),
- systemweit maximal 60 API-Aufrufe pro Minute abgesetzt werden (Rate-Limit pro API-Key).

Diese Begrenzungen dienen der Lastverteilung und dem Schutz der Infrastruktur von OpenWeatherMap.
Quellen: OpenWeatherMap Pricing, API Guidelines

Daher wurde in der Umsetzung ein Abrufintervall von 600 Sekunden (10 Minuten) pro Stadt konfiguriert, um regelkonform zu bleiben und einen stabilen Betrieb sicherzustellen.

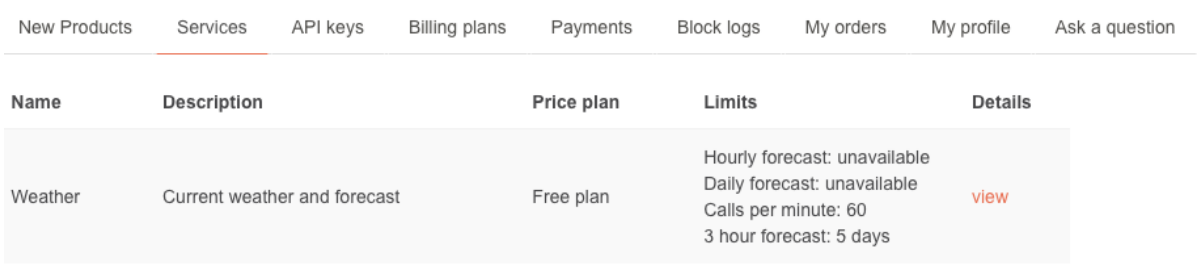


Abb.1: Verwendeter OpenWeatherMap Price-Plan, Quelle: Philippe Christen

Zielarchitektur (Soll-Zustand)

Ziel ist der Aufbau einer containerisierten Streaming-Architektur zur Erfassung, Anreicherung, Speicherung und Visualisierung von Wetterdaten in nahezu Echtzeit. Die Architektur basiert auf einem modularen Aufbau:

- OpenWeatherMap API: Quelle der Sensordaten (Polling)
- Kafka: Messaging Layer zur Entkopplung der Verarbeitung
- Apache Beam: Verarbeitungspipeline mit Filterung, Standardisierung, Anomalieerkennung und Persistenz
- PostgreSQL: Zentrale relationale Datenhaltung für strukturierte Auswertungen
- Grafana: Visualisierung mit Dashboards und Zeitreihenanalysen

Die gesamte Architektur ist mit Docker Compose orchestriert. Zeitstempel (event_time, received_time, processing_time) werden zur Nachvollziehbarkeit aller Verarbeitungsschritte in der Datenbank gespeichert. Duplikate werden durch ein kombiniertes Schlüsselattribut (UNIQUE(city, timestamp)) und ON CONFLICT DO NOTHING verhindert.

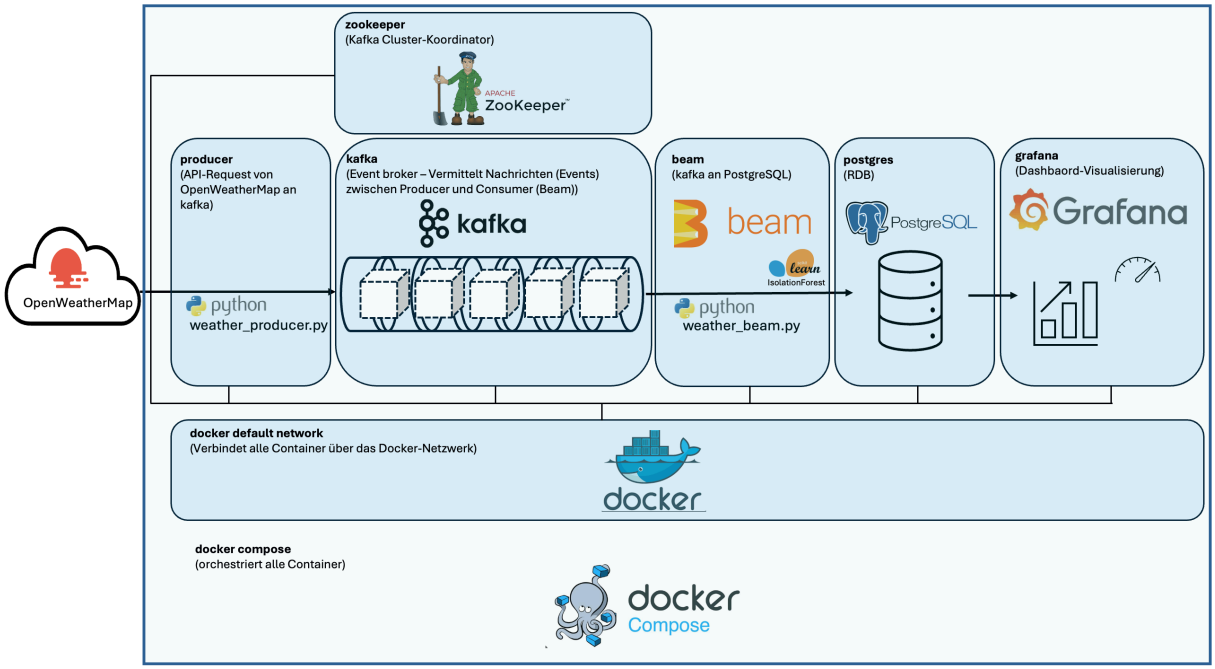


Abb.2: Zielarchitektur, Quelle: Philippe Christen

Technologie-Entscheidungen

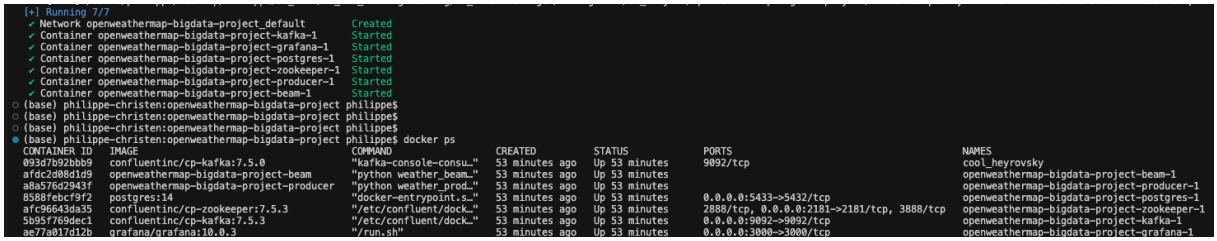
Apache Beam wurde gewählt, um eine erweiterbare, portierbare und testbare Streaming-verarbeitung zu ermöglichen. Beam trennt klar zwischen Pipeline-Definition und Ausführung (Runner), was lokale Tests mit dem DirectRunner und später Cloud-Migrationen (z. B. Google Cloud DataFlow) erleichtert.

Kafka dient als Puffer und Messaging-System. Durch die Verwendung von Topics wird eine lose Kopplung zwischen Producer und Beam-Consumer erreicht. Nachrichten können mehrfach gelesen oder reprocessed werden.

PostgreSQL bietet eine robuste, relationale Speicherung, ist mit Grafana integrierbar und erlaubt komplexe Zeitreihenanalysen. Funktionen wie date_trunc, Aggregationen und Filter unterstützen flexible Auswertungen und Heatmaps.

Grafana wurde als Visualisierungsplattform gewählt, da es eine direkte Anbindung an PostgreSQL erlaubt, interaktive Dashboards unterstützt und für Monitoring- und Analysezwecke im DevOps-Umfeld etabliert ist. Zudem ermöglicht der JSON-Export ein schnelles Reproduzieren der Visualisierungen in anderen Umgebungen.

Python wurde als Sprache aufgrund der Bibliotheken kafka-python, psycopg2, apache-beam, joblib, pandas und dotenv verwendet. Alle Services werden via Docker containerisiert und gemeinsam orchestriert.

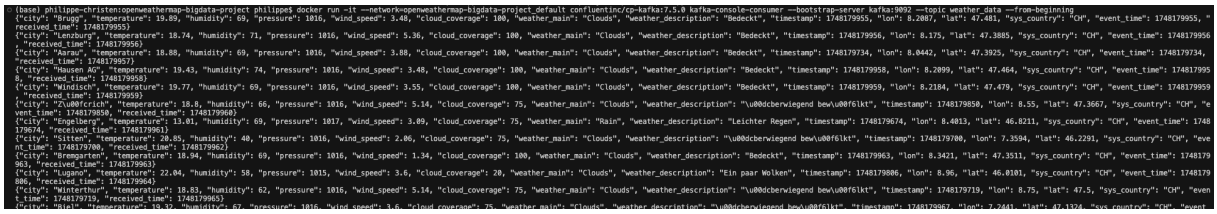


```
[+] Running 7/7
 ✔ Network openweathermap-bigdata-project_default        Created
 ✔ Container openweathermap-bigdata-project-kafka-1      Started
 ✔ Container openweathermap-bigdata-project-grafana-1    Started
 ✔ Container openweathermap-bigdata-project-postgres-1   Started
 ✔ Container openweathermap-bigdata-project-zookeeper-1  Started
 ✔ Container openweathermap-bigdata-project-producer-1   Started
 ✔ Container openweathermap-bigdata-project-beam-1       Started
(base) philippe-christen:openweathermap-bigdata-project philippe$
(base) philippe-christen:openweathermap-bigdata-project philippe$
(base) philippe-christen:openweathermap-bigdata-project philippe$
(base) philippe-christen:openweathermap-bigdata-project philippe$ docker ps
CONTAINER ID   IMAGE                              COMMAND                  CREATED         STATUS         PORTS                                            NAMES
093d7b92bbb9   confluentinc/cp-kafka:7.5.0        "kafka-console-consu…"   53 minutes ago  Up 53 minutes  9092/tcp                                         cool_heyrovsky
afdc2d08d1d9   openweathermap-bigdata-project-beam "python weather_beam…"  53 minutes ago  Up 53 minutes                                                   openweathermap-bigdata-project-beam-1
a8a576d2943f   openweathermap-bigdata-project-producer "python weather_prod…" 53 minutes ago  Up 53 minutes                                                   openweathermap-bigdata-project-producer-1
8588febcf9f2   postgres:14                        "docker-entrypoint.s…"   53 minutes ago  Up 53 minutes  0.0.0.0:5433->5432/tcp                           openweathermap-bigdata-project-postgres-1
afc96643da35   confluentinc/cp-zookeeper:7.5.3    "/etc/confluent/dock…"   53 minutes ago  Up 53 minutes  2888/tcp, 0.0.0.0:2181->2181/tcp, 3888/tcp       openweathermap-bigdata-project-zookeeper-1
5b95f769dec1   confluentinc/cp-kafka:7.5.3        "/etc/confluent/dock…"   53 minutes ago  Up 53 minutes  0.0.0.0:9092->9092/tcp                           openweathermap-bigdata-project-kafka-1
ae77a017d12b   grafana/grafana:10.0.3             "/run.sh"                53 minutes ago  Up 53 minutes  0.0.0.0:3000->3000/tcp                           openweathermap-bigdata-project-grafana-1
```

Abb.3: Container-Umgebung, Quelle: Philippe Christen

```
', 'event_time': 1748179955, 'received_time': 1748179955}
INFO:kafka.conn:<BrokerConnection client_id=kafka-python-producer-2, node_id=1 host=kafka:9092 <connecting> [IPv4 ('172.21.0.2', 9092)]>: connecting to kafka:9092 [('172.21.0.2', 9092) IPv4]
INFO:kafka.conn:<BrokerConnection client_id=kafka-python-producer-2, node_id=1 host=kafka:9092 <connected> [IPv4 ('172.21.0.2', 9092)]>: Connection complete.
INFO:kafka.conn:<BrokerConnection client_id=kafka-python-producer-2, node_id=bootstrap-0 host=kafka:9092 <connected> [IPv4 ('172.21.0.2', 9092)]>: Closing connection.
INFO:__main__:Sende an Kafka: {'city': 'Lenzburg', 'temperature': 18.74, 'humidity': 71, 'pressure': 1016, 'wind_speed': 5.36, 'cloud_coverage': 100, 'weather_main': 'Clouds', 'weather_description': 'Bedeckt', 'timestamp': 1748179956, 'lon': 8.175, 'lat': 47.3885, 'sys_country': 'CH', 'event_time': 1748179956, 'received_time': 1748179956}
INFO:__main__:Sende an Kafka: {'city': 'Aarau', 'temperature': 18.88, 'humidity': 69, 'pressure': 1016, 'wind_speed': 3.88, 'cloud_coverage': 100, 'weather_main': 'Clouds', 'weather_description': 'Bedeckt', 'timestamp': 1748179734, 'lon': 8.0442, 'lat': 47.3925, 'sys_country': 'CH', 'event_time': 1748179734, 'received_time': 1748179957}
INFO:__main__:Sende an Kafka: {'city': 'Hausen AG', 'temperature': 19.43, 'humidity': 74, 'pressure': 1016, 'wind_speed': 3.48, 'cloud_coverage': 100, 'weather_main': 'Clouds', 'weather_description': 'Bedeckt', 'timestamp': 1748179958, 'lon': 8.2099, 'lat': 47.464, 'sys_country': 'CH', 'event_time': 1748179958, 'received_time': 1748179958}
INFO:__main__:Sende an Kafka: {'city': 'Windisch', 'temperature': 19.77, 'humidity': 69, 'pressure': 1016, 'wind_speed': 3.55, 'cloud_coverage': 100, 'weather_main': 'Clouds', 'weather_description': 'Bedeckt', 'timestamp': 1748179959, 'lon': 8.2184, 'lat': 47.479, 'sys_country': 'CH', 'event_time': 1748179959, 'received_time': 1748179959}
INFO:__main__:Sende an Kafka: {'city': 'Zürich', 'temperature': 18.8, 'humidity': 66, 'pressure': 1016, 'wind_speed': 5.14, 'cloud_coverage': 75, 'weather_main': 'Clouds', 'weather_description': 'Überwiegend bewölkt', 'timestamp': 1748179850, 'lon': 8.55, 'lat': 47.3667, 'sys_country': 'CH', 'event_time': 1748179850, 'received_time': 1748179960}
INFO:__main__:Sende an Kafka: {'city': 'Engelberg', 'temperature': 13.01, 'humidity': 69, 'pressure': 1017, 'wind_speed': 3.09, 'cloud_coverage': 75, 'weather_main': 'Rain', 'weather_description': 'Leichter Regen', 'timestamp': 1748179674, 'lon': 8.4013, 'lat': 46.8211, 'sys_country': 'CH', 'event_time': 1748179674, 'received_time': 1748179961}
INFO:__main__:Sende an Kafka: {'city': 'Sitten', 'temperature': 20.85, 'humidity': 40, 'pressure': 1016, 'wind_speed': 2.06, 'cloud_coverage': 75, 'weather_main': 'Clouds', 'weather_description': 'Überwiegend bewölkt', 'timestamp': 1748179700, 'lon': 7.3594, 'lat': 46.2291, 'sys_country': 'CH', 'event_time': 1748179700, 'received_time': 1748179962}
INFO:__main__:Sende an Kafka: {'city': 'Bremgarten', 'temperature': 18.94, 'humidity': 69, 'pressure': 1016, 'wind_speed': 1.34, 'cloud_coverage': 100, 'weather_main': 'Clouds', 'weather_description': 'Bedeckt', 'timestamp': 1748179963, 'lon': 8.3421, 'lat': 47.3511, 'sys_country': 'CH', 'event_time': 1748179963, 'received_time': 1748179963}
INFO:__main__:Sende an Kafka: {'city': 'Lugano', 'temperature': 22.04, 'humidity': 58, 'pressure': 1015, 'wind_speed': 3.6, 'cloud_coverage': 20, 'weather_main': 'Clouds', 'weather_description': 'Ein paar Wolken', 'timestamp': 1748179806, 'lon': 8.96, 'lat': 46.0101, 'sys_country': 'CH', 'event_time': 1748179806, 'received_time': 1748179964}
INFO:__main__:Sende an Kafka: {'city': 'Winterthur', 'temperature': 18.83, 'humidity': 62, 'pressure': 1016, 'wind_speed': 5.14, 'cloud_coverage': 75, 'weather_main': 'Clouds', 'weather_description': 'Überwiegend bewölkt', 'timestamp': 1748179719, 'lon': 8.75, 'lat': 47.5, 'sys_country': 'CH', 'event_time': 1748179719, 'received_time': 1748179965}
INFO:__main__:Sende an Kafka: {'city': 'Biel', 'temperature': 19.32, 'humidity': 67, 'pressure': 1016, 'wind_speed': 3.6, 'cloud_coverage': 75, 'weather_main': 'Clouds', 'weather_description': 'Überwiegend bewölkt', 'timestamp': 1748179967, 'lon': 7.2441, 'lat': 47.1324, 'sys_country': 'CH', 'event_time': 1748179967, 'received_time': 1748179966}
```

Abb.4: Producer-Log, Quelle: Philippe Christen



```
(base) philippe-christen:openweathermap-bigdata-project philippe$ docker run -it --network=openweathermap-bigdata-project_default confluentinc/cp-kafka:7.5.0 kafka-console-consumer --bootstrap-server kafka:9092 --topic weather_data --from-beginning
{"city": "Brugg", "temperature": 19.89, "humidity": 69, "pressure": 1016, "wind_speed": 3.48, "cloud_coverage": 100, "weather_main": "Clouds", "weather_description": "Bedeckt", "timestamp": 1748179955, "lon": 8.2087, "lat": 47.481, "sys_country": "CH", "event_time": 1748179955, "received_time": 1748179955}
{"city": "Lenzburg", "temperature": 18.74, "humidity": 71, "pressure": 1016, "wind_speed": 5.36, "cloud_coverage": 100, "weather_main": "Clouds", "weather_description": "Bedeckt", "timestamp": 1748179956, "lon": 8.175, "lat": 47.3885, "sys_country": "CH", "event_time": 1748179956, "received_time": 1748179956}
{"city": "Aarau", "temperature": 18.88, "humidity": 69, "pressure": 1016, "wind_speed": 3.88, "cloud_coverage": 100, "weather_main": "Clouds", "weather_description": "Bedeckt", "timestamp": 1748179734, "lon": 8.0442, "lat": 47.3925, "sys_country": "CH", "event_time": 1748179734, "received_time": 1748179957}
{"city": "Hausen AG", "temperature": 19.43, "humidity": 74, "pressure": 1016, "wind_speed": 3.48, "cloud_coverage": 100, "weather_main": "Clouds", "weather_description": "Bedeckt", "timestamp": 1748179958, "lon": 8.2099, "lat": 47.464, "sys_country": "CH", "event_time": 1748179958, "received_time": 1748179958}
{"city": "Windisch", "temperature": 19.77, "humidity": 69, "pressure": 1016, "wind_speed": 3.55, "cloud_coverage": 100, "weather_main": "Clouds", "weather_description": "Bedeckt", "timestamp": 1748179959, "lon": 8.2184, "lat": 47.479, "sys_country": "CH", "event_time": 1748179959, "received_time": 1748179959}
{"city": "Zürich", "temperature": 18.8, "humidity": 66, "pressure": 1016, "wind_speed": 5.14, "cloud_coverage": 75, "weather_main": "Clouds", "weather_description": "\u00dcberwiegend bew\u00f6lkt", "timestamp": 1748179850, "lon": 8.55, "lat": 47.3667, "sys_country": "CH", "event_time": 1748179850, "received_time": 1748179960}
{"city": "Engelberg", "temperature": 13.01, "humidity": 69, "pressure": 1017, "wind_speed": 3.09, "cloud_coverage": 75, "weather_main": "Rain", "weather_description": "Leichter Regen", "timestamp": 1748179674, "lon": 8.4013, "lat": 46.8211, "sys_country": "CH", "event_time": 1748179674, "received_time": 1748179961}
{"city": "Sitten", "temperature": 20.85, "humidity": 40, "pressure": 1016, "wind_speed": 2.06, "cloud_coverage": 75, "weather_main": "Clouds", "weather_description": "\u00dcberwiegend bew\u00f6lkt", "timestamp": 1748179700, "lon": 7.3594, "lat": 46.2291, "sys_country": "CH", "event_time": 1748179700, "received_time": 1748179962}
{"city": "Bremgarten", "temperature": 18.94, "humidity": 69, "pressure": 1016, "wind_speed": 1.34, "cloud_coverage": 100, "weather_main": "Clouds", "weather_description": "Bedeckt", "timestamp": 1748179963, "lon": 8.3421, "lat": 47.3511, "sys_country": "CH", "event_time": 1748179963, "received_time": 1748179963}
{"city": "Lugano", "temperature": 22.04, "humidity": 58, "pressure": 1015, "wind_speed": 3.6, "cloud_coverage": 20, "weather_main": "Clouds", "weather_description": "Ein paar Wolken", "timestamp": 1748179806, "lon": 8.96, "lat": 46.0101, "sys_country": "CH", "event_time": 1748179806, "received_time": 1748179964}
{"city": "Winterthur", "temperature": 18.83, "humidity": 62, "pressure": 1016, "wind_speed": 5.14, "cloud_coverage": 75, "weather_main": "Clouds", "weather_description": "\u00dcberwiegend bew\u00f6lkt", "timestamp": 1748179719, "lon": 8.75, "lat": 47.5, "sys_country": "CH", "event_time": 1748179719, "received_time": 1748179965}
```

Abb.5: Kafka Live-Stream, Quelle: Philippe Christen

```
INFO:kafka.consumer.subscription_state:Updating subscribed topics to: ('weather_data',)
INFO:__main__:Kafka Consumer verbunden.
INFO:__main__:Starte Batch-Verarbeitung in 10er-Schritten...
INFO:kafka.consumer.subscription_state:Updated partition assignment: [TopicPartition(topic='weather_data', partition=0)]
INFO:kafka.conn:<BrokerConnection client_id=kafka-python-2.2.10, node_id=1 host=kafka:9092 <connecting> [IPv4 ('172.21.0.2', 9092)]>: connecting to kafka:9092 [('172.21.0.2', 9092) IPv4]
INFO:kafka.conn:<BrokerConnection client_id=kafka-python-2.2.10, node_id=1 host=kafka:9092 <connected> [IPv4 ('172.21.0.2', 9092)]>: Connection complete.
INFO:kafka.conn:<BrokerConnection client_id=kafka-python-2.2.10, node_id=bootstrap-0 host=kafka:9092 <connected> [IPv4 ('172.21.0.2', 9092)]>: Closing connection.
INFO:kafka.consumer.fetcher:Resetting offset for partition TopicPartition(topic='weather_data', partition=0) to offset 0.
INFO:__main__:Empfangen von Kafka: {'city': 'Brugg', 'temperature': 19.89, 'humidity': 69, 'pressure': 1016, 'wind_speed': 3.48, 'cloud_coverage': 100, 'weather_main': 'Clouds', 'weather_description': 'Bedeckt', 'timestamp': 1748179955, 'lon': 8.2087, 'lat': 47.481, 'sys_country': 'CH', 'event_time': 1748179955, 'received_time': 1748179955}
INFO:__main__:Empfangen von Kafka: {'city': 'Lenzburg', 'temperature': 18.74, 'humidity': 71, 'pressure': 1016, 'wind_speed': 5.36, 'cloud_coverage': 100, 'weather_main': 'Clouds', 'weather_description': 'Bedeckt', 'timestamp': 1748179956, 'lon': 8.175, 'lat': 47.3885, 'sys_country': 'CH', 'event_time': 1748179956, 'received_time': 1748179956}
INFO:__main__:Empfangen von Kafka: {'city': 'Aarau', 'temperature': 18.88, 'humidity': 69, 'pressure': 1016, 'wind_speed': 3.88, 'cloud_coverage': 100, 'weather_main': 'Clouds', 'weather_description': 'Bedeckt', 'timestamp': 1748179734, 'lon': 8.0442, 'lat': 47.3925, 'sys_country': 'CH', 'event_time': 1748179734, 'received_time': 1748179957}
INFO:__main__:Empfangen von Kafka: {'city': 'Hausen AG', 'temperature': 19.43, 'humidity': 74, 'pressure': 1016, 'wind_speed': 3.48, 'cloud_coverage': 100, 'weather_main': 'Clouds', 'weather_description': 'Bedeckt', 'timestamp': 1748179958, 'lon': 8.2099, 'lat': 47.464, 'sys_country': 'CH', 'event_time': 1748179958, 'received_time': 1748179958}
INFO:__main__:Empfangen von Kafka: {'city': 'Windisch', 'temperature': 19.77, 'humidity': 69, 'pressure': 1016, 'wind_speed': 3.55, 'cloud_coverage': 100, 'weather_main': 'Clouds', 'weather_description': 'Bedeckt', 'timestamp': 1748179959, 'lon': 8.2184, 'lat': 47.479, 'sys_country': 'CH', 'event_time': 1748179959, 'received_time': 1748179959}
INFO:root:Missing pipeline option (runner). Executing pipeline using the default runner: DirectRunner.
INFO:root:Verarbeite Batch mit 5 Nachrichten via Beam...
INFO:apache_beam.runners.worker.statecache:Creating state cache with size 104857600
INFO:__main__:Empfangen von Kafka: {'city': 'Zürich', 'temperature': 18.8, 'humidity': 66, 'pressure': 1016, 'wind_speed': 5.14, 'cloud_coverage': 75, 'weather_main': 'Clouds', 'weather_description': 'Überwiegend bewölkt', 'timestamp': 1748179850, 'lon': 8.55, 'lat': 47.3667, 'sys_country': 'CH', 'event_time': 1748179850, 'received_time': 1748179960}
INFO:__main__:Empfangen von Kafka: {'city': 'Engelberg', 'temperature': 13.01, 'humidity': 69, 'pressure': 1017, 'wind_speed': 3.09, 'cloud_coverage': 75, 'weather_main': 'Rain', 'weather_description': 'Leichter Regen', 'timestamp': 1748179674, 'lon': 8.4013, 'lat': 46.8211, 'sys_country': 'CH', 'event_time': 1748179674, 'received_time': 1748179961}
INFO:__main__:Empfangen von Kafka: {'city': 'Sitten', 'temperature': 20.85, 'humidity': 40, 'pressure': 1016, 'wind_speed': 2.06, 'cloud_coverage': 75, 'weather_main': 'Clouds', 'weather_description': 'Überwiegend bewölkt', 'timestamp': 1748179700, 'lon': 7.3594, 'lat': 46.2291,
```

Abb.6: Beam-Log, Quelle: Philippe Christen

city	temperature	humidity	pressure	wind_speed	cloud_coverage	weather_main	weather_description	timestamp	lon	lat	sys_country	event_time	received_time	processing_time	anomaly
Yverdon-les-Bains	21.26	68	1017	14.7	93	Clouds	Bedeckt	1748180001	6.6412	46.7785	CH	1748180001	1748180001	1748180002	f
Aigle VD	22.1	63	1016	12.9	98	Clouds	Bedeckt	1748180000	6.9646	46.3181	CH	1748180000	1748180000	1748180002	f
Sarnen	16.67	66	1017	11.1	75	Clouds	Überwiegend bewölkt	1748179999	8.2507	46.8985	CH	1748179999	1748179999	1748180002	f
Herisau	15.21	76	1017	7.7	100	Clouds	Bedeckt	1748179998	9.2792	47.3862	CH	1748179998	1748179998	1748180002	f
Altdorf	15.84	76	1017	3.6	100	Clouds	Bedeckt	1748179997	8.6444	46.8804	CH	1748179997	1748179997	1748180002	f
Appenzell	14.14	76	1017	5.3	100	Clouds	Bedeckt	1748179994	9.41	47.331	CH	1748179994	1748179994	1748179996	f
Delémont	19.4	65	1016	14	100	Rain	Mäßiger Regen	1748179993	7.3445	47.3649	CH	1748179993	1748179993	1748179996	f
Wil SG	17.23	75	1016	14.5	100	Clouds	Bedeckt	1748179988	9.0455	47.4615	CH	1748179988	1748179988	1748179991	f
Burgdorf BE	18.64	76	1016	13	75	Rain	Leichter Regen	1748179984	7.6279	47.059	CH	1748179984	1748179984	1748179986	f
Thun	19.4	62	1016	9.6	99	Rain	Leichter Regen	1748179983	7.6217	46.7512	CH	1748179983	1748179983	1748179986	f
Locarno	23.28	61	1014	6.4	77	Clouds	Überwiegend bewölkt	1748179980	8.7995	46.1709	CH	1748179980	1748179980	1748179980	f
Samedan	11.89	58	1017	24.1	75	Clouds	Überwiegend bewölkt	1748179979	9.8712	46.5342	CH	1748179979	1748179979	1748179980	t
Chur	18.24	76	1017	7	96	Clouds	Bedeckt	1748179978	9.5329	46.8499	CH	1748179978	1748179978	1748179980	f
St. Gallen	15.74	63	1017	11.1	75	Clouds	Überwiegend bewölkt	1748179977	9.3748	47.4239	CH	1748179977	1748179977	1748179980	f
Kanton Neuenburg	21.41	46	1017	16.7	75	Clouds	Überwiegend bewölkt	1748179975	6.8333	46.9167	CH	1748179975	1748179975	1748179975	f
Freiburg	18.94	56	1017	16.7	75	Clouds	Überwiegend bewölkt	1748179974	7.1513	46.8024	CH	1748179974	1748179974	1748179975	f
Biel	19.32	67	1016	13	75	Clouds	Überwiegend bewölkt	1748179967	7.2441	47.1324	CH	1748179967	1748179966	1748179970	f
Bern	18.99	70	1016	14.5	75	Clouds	Überwiegend bewölkt	1748179966	7.4474	46.9481	CH	1748179966	1748179968	1748179970	f
Bremgarten	18.94	69	1016	4.8	100	Clouds	Bedeckt	1748179963	8.3421	47.3511	CH	1748179963	1748179963	1748179965	f
Basel	21.23	49	1016	22.2	0	Clear	Klarer Himmel	1748179959	7.5733	47.5584	CH	1748179959	1748179973	1748179975	f
Windisch	19.77	69	1016	12.8	100	Clouds	Bedeckt	1748179959	8.2184	47.479	CH	1748179959	1748179959	1748179959	f
Hausen AG	19.43	74	1016	12.5	100	Clouds	Bedeckt	1748179958	8.2099	47.464	CH	1748179958	1748179958	1748179959	f
Lenzburg	18.74	71	1016	19.3	100	Clouds	Bedeckt	1748179956	8.175	47.3885	CH	1748179956	1748179956	1748179959	f
Brugg	19.89	69	1016	12.5	100	Clouds	Bedeckt	1748179955	8.2087	47.481	CH	1748179955	1748179955	1748179959	f
Bulle FR	18.04	58	1017	17.7	92	Rain	Leichter Regen	1748179859	7.0567	46.6195	CH	1748179859	1748179991	1748179991	f
Zürich	18.8	66	1016	18.5	75	Clouds	Überwiegend bewölkt	1748179850	8.55	47.3667	CH	1748179850	1748179960	1748179965	f
Olten	18.87	66	1016	1.6	100	Clouds	Bedeckt	1748179847	7.9033	47.35	CH	1748179847	1748179985	1748179986	f

Abb.7: psql-Abfrage, Quelle: Philippe Christen

Abb.8: Grafana Dashboard, Quelle: Philippe Christen

Erweiterung: Anomalieerkennung mit Machine Learning

Zur qualitativen Erweiterung wurde ein auf IsolationForest basierendes ML-Modell integriert, das meteorologische Anomalien (z. B. unplausible Kombinationen von Wind, Temperatur, Luftdruck) erkennt. Das Modell wurde offline trainiert und via joblib serialisiert. In der Beam-Pipeline wird es mit einem standardisierten Feature-Vektor (6 Merkmale: Temperatur, Wind, Luftdruck, Luftfeuchtigkeit, Längengrad, Breitengrad) verwendet. Die Vorverarbeitung erfolgt mittels StandardScaler. Die Features werden in einem pandas.DataFrame organisiert, um eine skalierbare Übergabe an den StandardScaler zu ermöglichen. Der boolesche Prädiktor «anomaly» wird in der Datenbank gespeichert und zur Analyse in Grafana genutzt. Fehlerquellen – wie inkonsistente Typen oder fehlende Spaltennamen – wurden durch explizite Konvertierung der Anomalievariable zu bool und die Übergabe benannter Spalten im DataFrame behoben.

Fazit und Ausblick

Die entwickelte Pipeline zeigt, wie sich mit Apache Beam und Kafka ein modular aufgebautes, robustes Streaming-System realisieren lässt. Durch die Anomalieerkennung wurde die analytische Tiefe der Daten erheblich erhöht. Die Visualisierung in Grafana erlaubt Echtzeit-Monitoring und Auswertung historischer Latenzen (Lag).

Zukünftig denkbare Erweiterungen

- Einführung von Windowing und Triggering in Beam für dynamische Zeitfenster
- Verwendung von Apache Flink als Runner für produktive Latenzanforderungen
- Modell-Drift-Überwachung und Nachtraining des ML-Modells
- Echtzeit-Alerting bei Anomalien via Webhook/Slack/E-Mail
- Die Architektur ist flexibel, lokal lauffähig, nachvollziehbar und auf zukünftige Anforderungen erweiterbar.
- Deployment in Cloud-Umgebungen wie Google Cloud Dataflow oder AWS MSK zur weiteren Skalierung
- Orchestrierung mit Kubernetes, um Hochverfügbarkeit, Auto-Scaling und Ausfallsicherheit zu gewährleisten
- Data Governance & Monitoring: Implementierung eines zentralen Monitorings (z. B. mit Prometheus und Alertmanager), um Logs automatisiert auszuwerten und bei API-Ausfällen oder Fehlern im Streaming-Prozess frühzeitig Benachrichtigungen auszulösen. Dies erhöht die Betriebssicherheit und unterstützt Data Governance durch transparente Überwachung der Pipeline