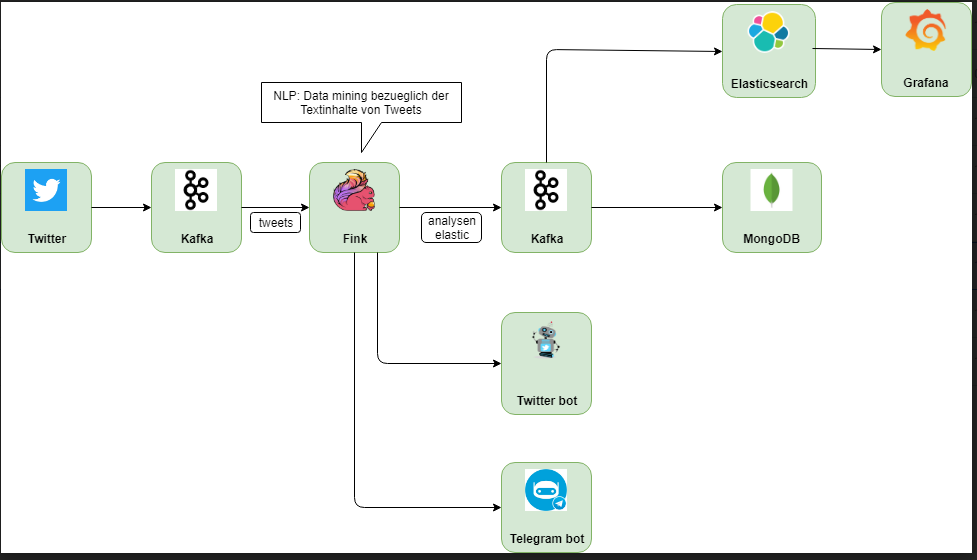
**Moderne Datenbanken – SS 2018 Projekt-Fragebogen**



|  |  |
| --- | --- |
| **Gruppenbezeichnung** | Gruppe 5 |
| **Gruppenmitglieder** | Tim Eichholz  Marvin Wölk  Sebastian Linde |
| **Anzahl der Komponenten** |  |
| 1. Komponente (oblig. DB) | Elasticsearch |
| 2. Komponente (oblig. DB) | Mongo DB |
| 3. Komponente (optional) | Kibana |
| 4. Komponente (optional) | Telegram Bot (Mankis\_bot) |
| 5. Komponente (optional) | Kafka |
| Mehr (optional) |  |
| **Apache Kafka & Flink** |  |
| Kafka verwendet | Ja |
| einfacher Flink-Job (Filtern, Aggr.) |  |
| Flink-Job mit Split/Join/Steuerung | Ja, aufsplitten der Tweets, analysieren und join |
| Flink-Job mit Iterationen, Feedback Batchverarb. |  |
| Nutzung Flink API ML, Gelly, CEP, Table | Aylien NLP API, Apache OpenNLP |
| **Datenstromdesign** |  |
| Einfaches Design mit 2 Kafka-Topics | 3 Topic: Tweets, analysen, elastic |
| Single-Event Pattern Univ. Microarchit. |  |
| Mehrphasige Verarb. Anomalieerkennung |  |
| External Lookup, Entkopplung Pattern |  |
| Lambda-Architektur CEP |  |
| **Dashboard & Schnittstellen** |  |
| Connector (oblig.) | Ja |
| Dashboard (Monitoring aller Datenströme) | Kibana |
| Zus. Dashboard-Funk (Alerts) | Telegram Bot |
| Schnittstellenprog. implementiert | * Telegram * Apache Open NLP * Aylien NLP API |
| Datenquelle implem. (z.B. Sensoren via Arduino-Board) | Twitter streaming API |

Datenstromdesign