KR18 – 7SHIELD integrated platform

Purpose

Apache Kafka is a publish-subscribe **distributed messaging system** that achieves a high throughput system due to its performance, scalability and fault tolerance properties. The Kafka framework is useful for both streaming and offline processing. In addition, it can exchange data with relational databases, NoSQL databases or other distributed environments.

Therefore, due to its high performance for streaming computing (with many TBs of stored data and over 2 million writes per second), Apache Kafka was chosen to be the **communication bus (message broker) between the modules of the 7SHIELD framework**.

Integration

Apache Kafka is used as a **messaging system** to allow all components of the 7SHIELD framework to communicate with others.

Many systems require data processing as soon as they become available. Kafka transmits data from producers to consumers with very low latency (5 milliseconds, for example). This is useful for real-time data processing on large distributed systems (IoT), where models constantly analyze metric streams from field equipment and trigger alarms immediately upon detecting deviations that could indicate impending failure.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 883284

Partners

ENGINEERING Italy

Engineering group (ENG) supports the specification and integration of the systems of critical infrastructures and makes resources related to integrated cyber-physical situational awareness and threat intelligence.

Besides all the technologies available supporting the integration of different systems, considering the full list of implemented within the 7SHIELD framework and using different program languages and technologies, Kafka broker was chosen for its wide range of supporting languages: C/C++, lojure, C# / .Net, Go Groovy, Java / Spring Boot, Kotlin, Node.js, Python, Ruby, Rust, Scala, Perl, Erlang, PHP, Rust, Storm, Swift.

Modules Communication

The communication between the various modules within the 7SHIELD framework is mostly made through the Apache Kafka broker (as well as through specific methods such as API RESTful Services). The modules using Kafka as a broker are depicted in the **7SHIELD architecture schema**.

All the modules developed in the frame of 7SHIELD have been designed with the consultancy of identified external stakeholders, first responders and following the **requirements** provided by the partners working in the space sector acting as Pilots, who provided the Critical Infrastructures for **testing and demonstration**.

Technology

Apache Kafka (**Apache License 2.0**) is written in Scala, Java and is a Cross-Platform system. Apache Kafka uses Zookeeper, which is a centralized service for maintaining configuration information, naming and providing flexible and robust synchronization within distributed systems.

A Kafka broker can work on a distributed system, and for this reason a Kafka broker allows you to partition N topics into N different machines, in order to overcome the performance limit of the single host machine.

Stakeholders

This messaging tool can be used in **all sectors** for different reasons. In addition to predictive maintenance (IoT), the instant messaging broker is useful for financial organizations to collect and process payments and financial transactions in real time, block fraudulent transactions the instant moment they are detected, or update dashboards with market prices updated per second. It can also be useful for autonomous mobile devices (cars, robots), which require real-time data processing to navigate a physical environment, logistics and supply chain activities, to monitor and update tracking applications, for example to monitor merchant ships for real-time cargo delivery estimates.

Contact

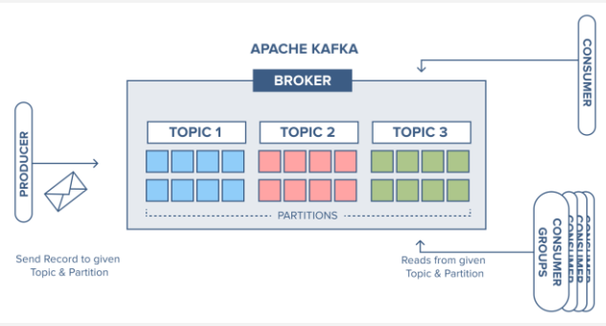
- info@eng.it

Future Improvements

Currently, Apache Kafka uses Apache ZooKeeper cluster to store its metadata such as partition location and topic configuration.

The idea is to break this dependency and bring metadata management within Kafka itself through a **dynamic service that runs within the Kafka Cluster**. It is called Quorum Controller, and uses the Consensus algorithm to agree on a single data value between processes or distributed systems. It includes: performance improvements; support for mutual TLS authentication (SASL_SSL), thus improving the ability to secure communicating environments; better management of the registration hierarchy. Improved LOGs handling to the JSON structure so that they can be analyzed and used more easily by logging toolchains with a hierarchical model.

This is the next generation of Kafka brokers.



www.7shield.eu