Software Architectures 2018/19 - MEB-POC Project

Group: LinuxFellows

Requirements

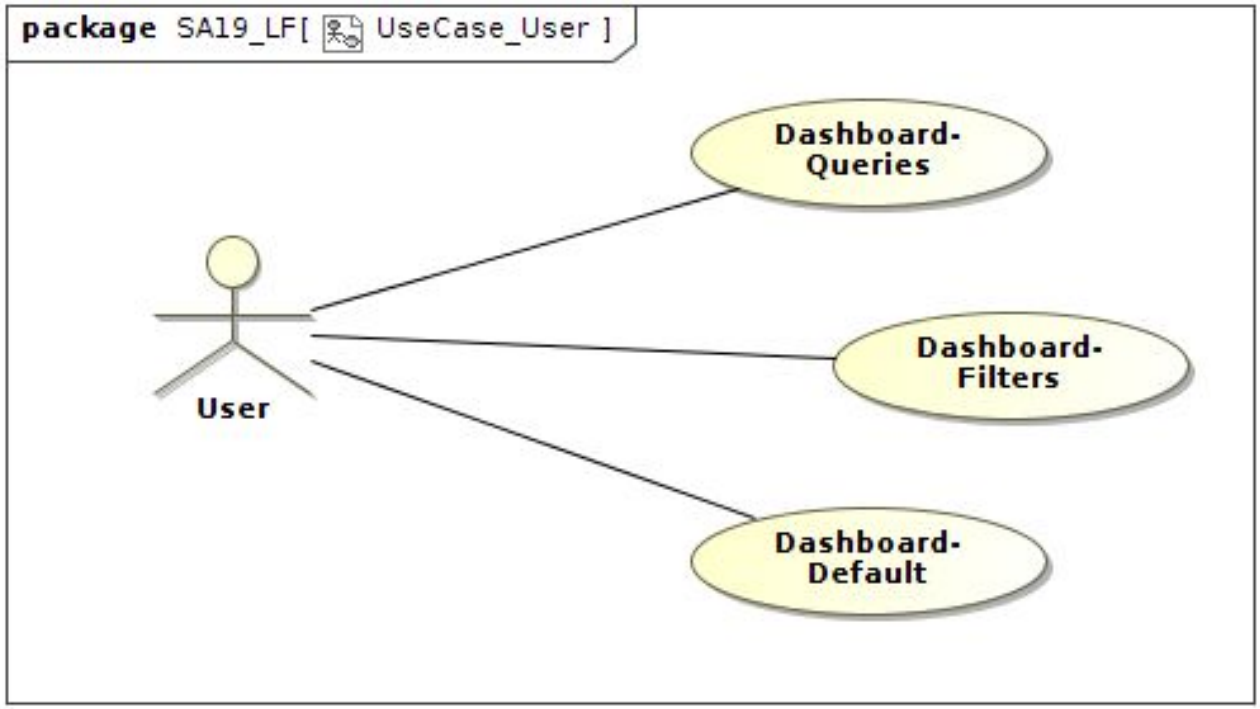
Functional

- ❏ Dashboard-Queries
- ❏ Dashboard-Filters
- ❏ Dashboard-Default
- ❏ Information-Structure
- ❏ Information-Concurrency
- ❏ Information-Processing
- ❏ IO-Input
- ❏ IO-Output

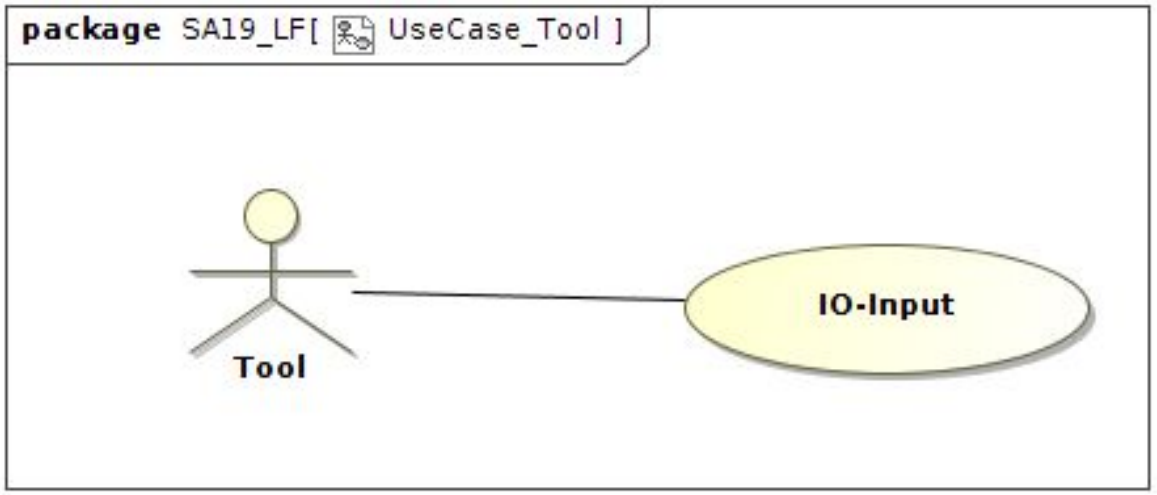
Non Functional

- ❏ Scalability
- ❏ Performance
- ❏ Availability
- ❏ Reliability

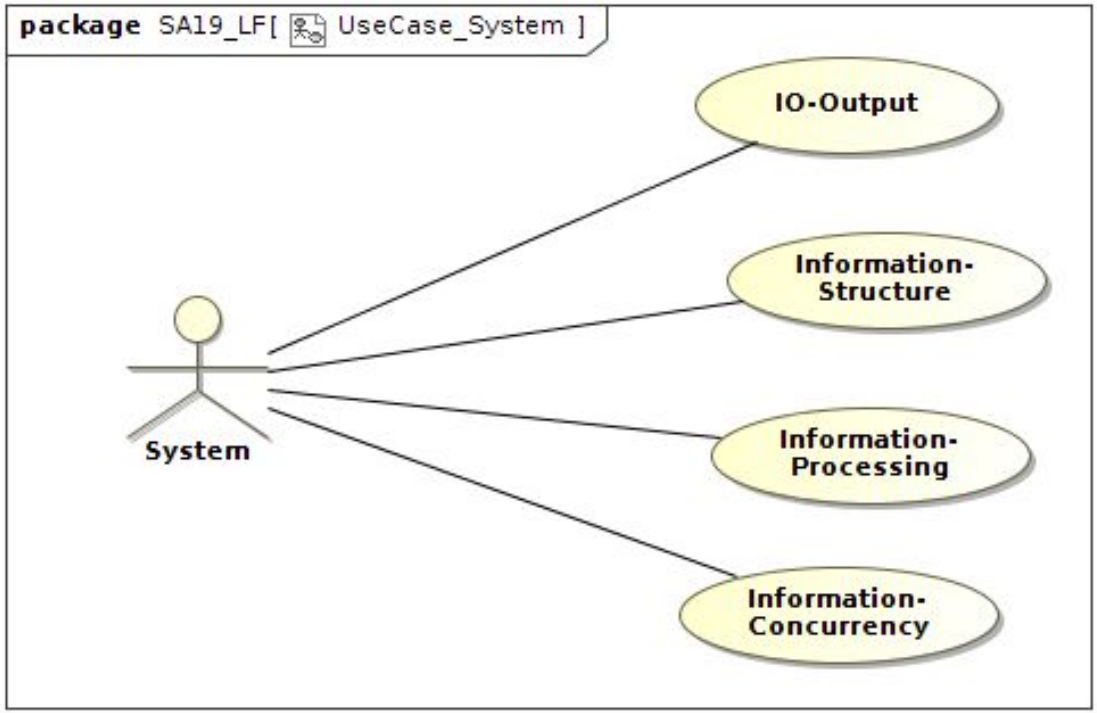
Use case - User



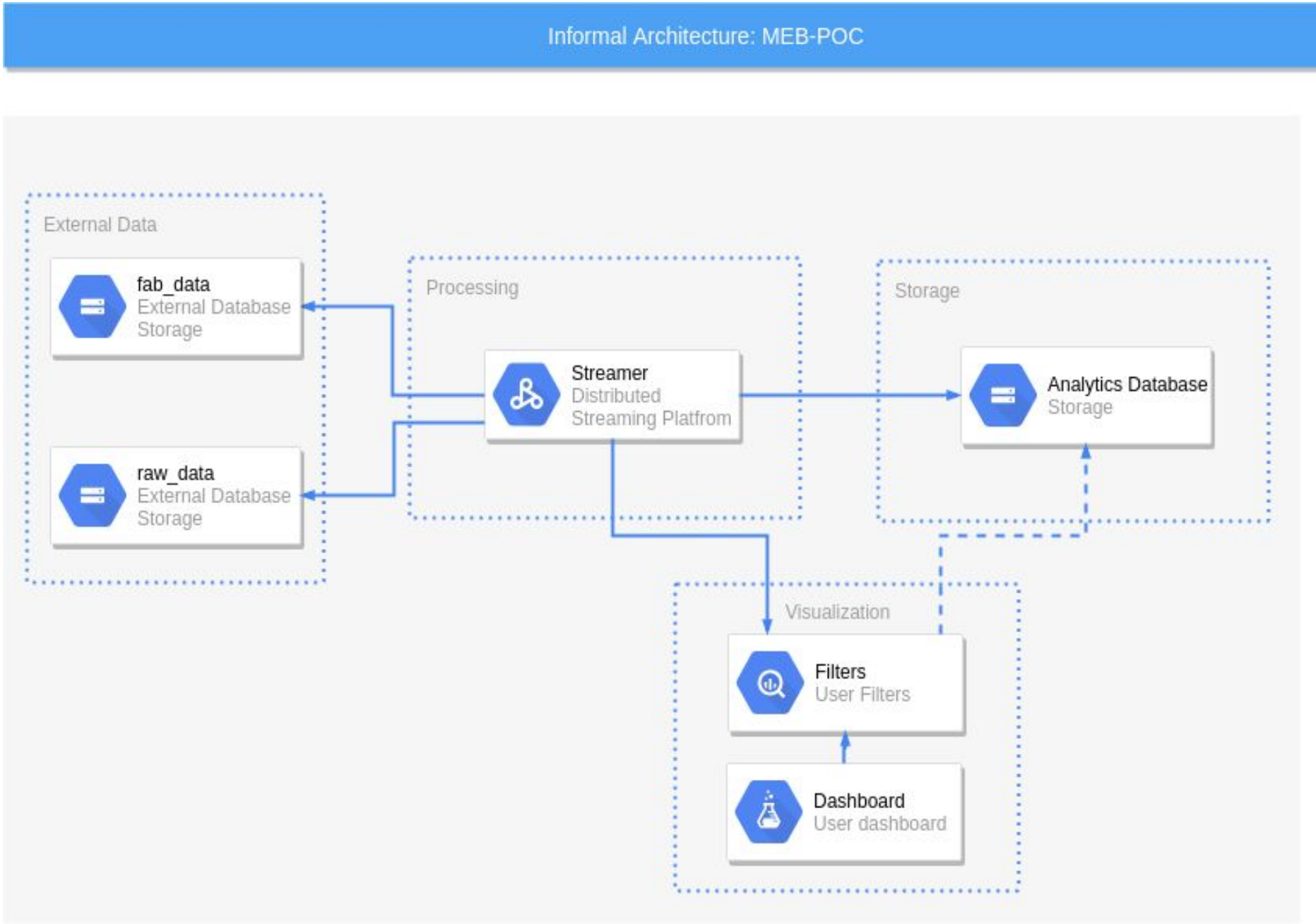
Use case - Tool



Use case - System



Informal description



Design decision #1

Problem: How does the system take the events from the tools?

Decided option: Pull events from the provided database.

Motivation: Number of messages & network traffic, fault-tolerance, loss of data, ease of simulation.

Design decision #2

Problem: Which architectural pattern should the system use?

Decided option: Publish/Subscribe

Motivation: Scalability (horizontal-scalability), de-coupling.

Design decision #3

Problem: Which message broker/streaming platform should the system use?

Decided option: Apache Kafka

Motivation: Real-time, persistence, history, team knowledge about the technology and available documentation, Kafka Connect and Stream API.

Design decision #4

Problem: How does the system handle raw_data?

Decided option: Internal Caching

Motivation: Memory usage, Performance, Scalability.

Design decision #5

Problem: Where are the information stored?

Decided option: Kafka State & External Database

Motivation: Fault-tolerance, Relations between the data, Simplicity of the architecture, Usage of third-parts dashboard systems.

Caps Design decision #1

Problem: How to illustrate the flow sequence of commands and data in the SAML diagram?

Decided option: AlphaNumeric Tagging with Alphabetic Sources.

Motivation: Human Readable, Easily Expandable, Easy to Follow.

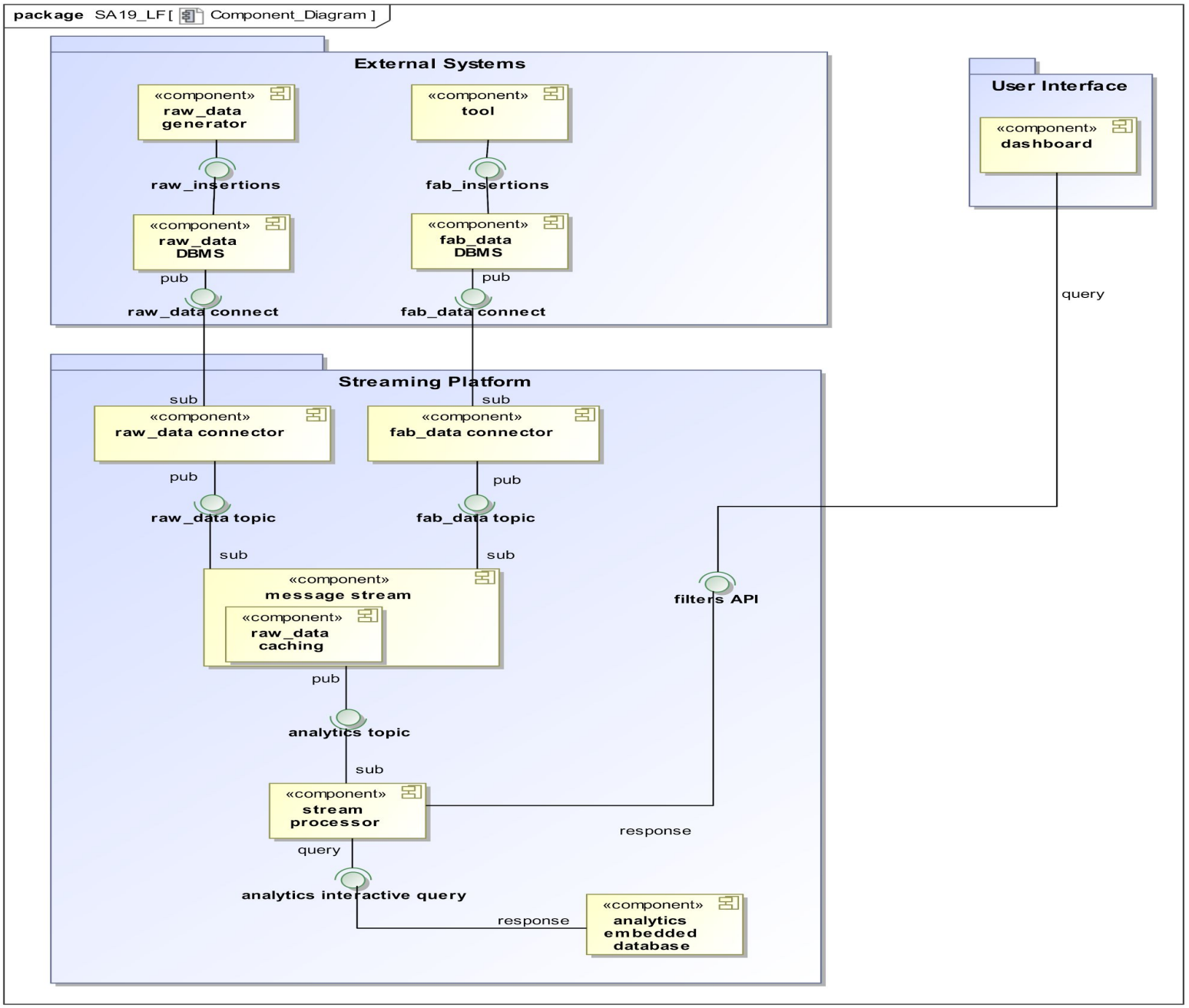
Caps Design decision #2

Problem: How do we represent the tool output in CAPS?

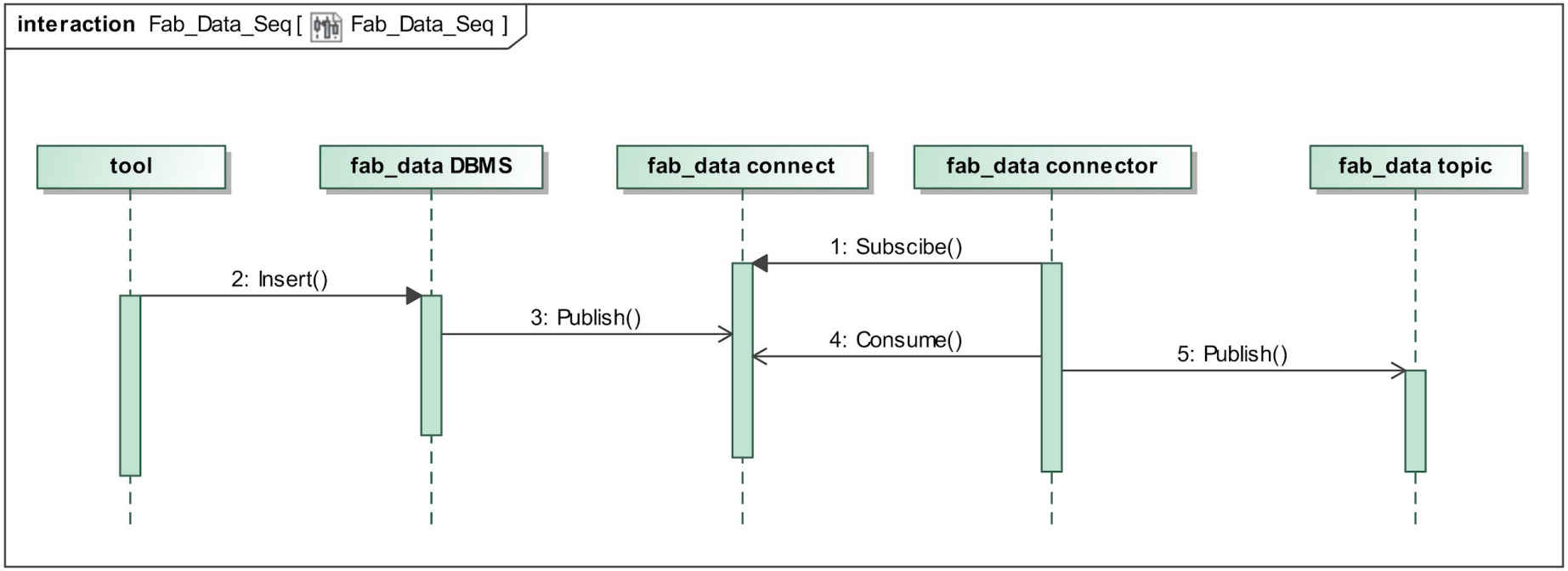
Decided option: XML

Motivation: To comply with the specification assuming the fab_data database can convert XML.

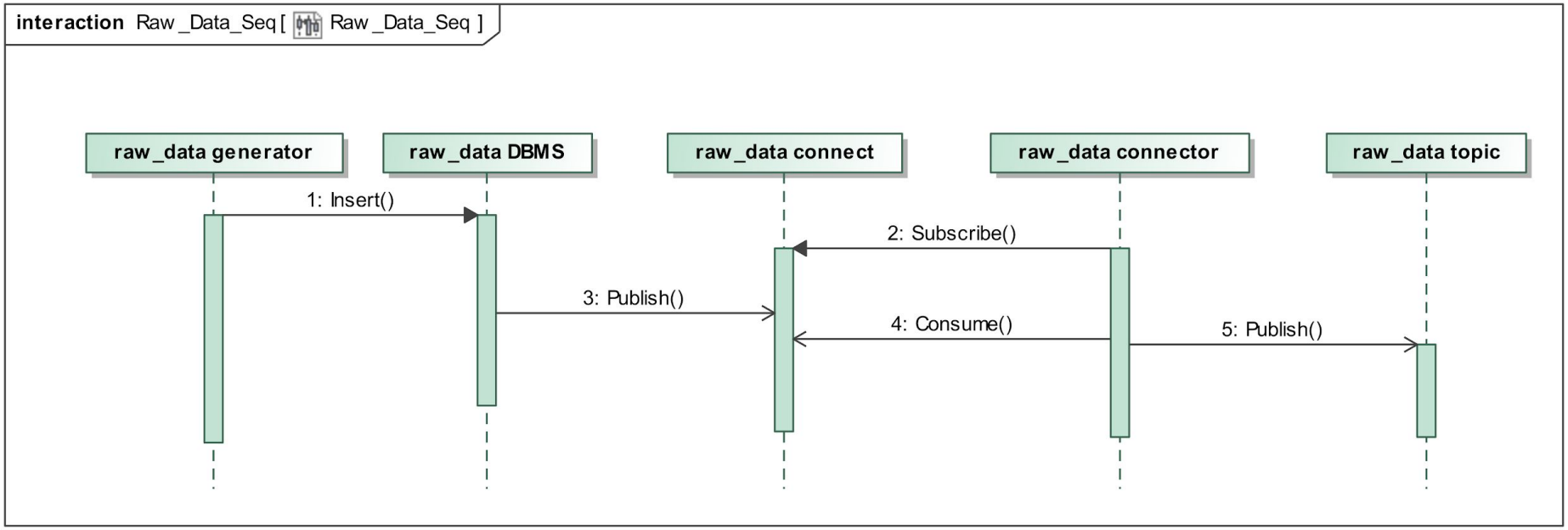
Component Diagram



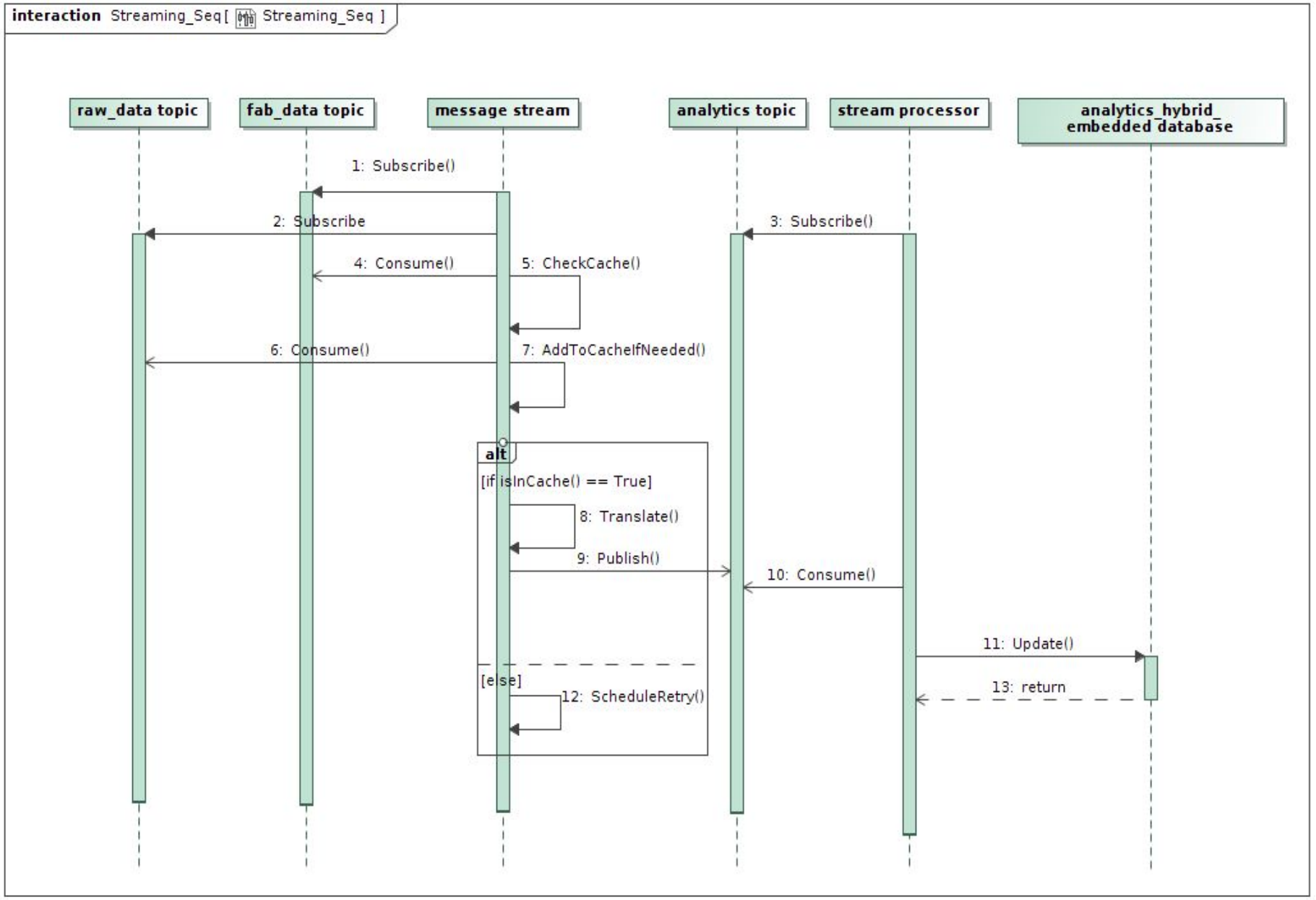
Sequence diagram - Kafka connect



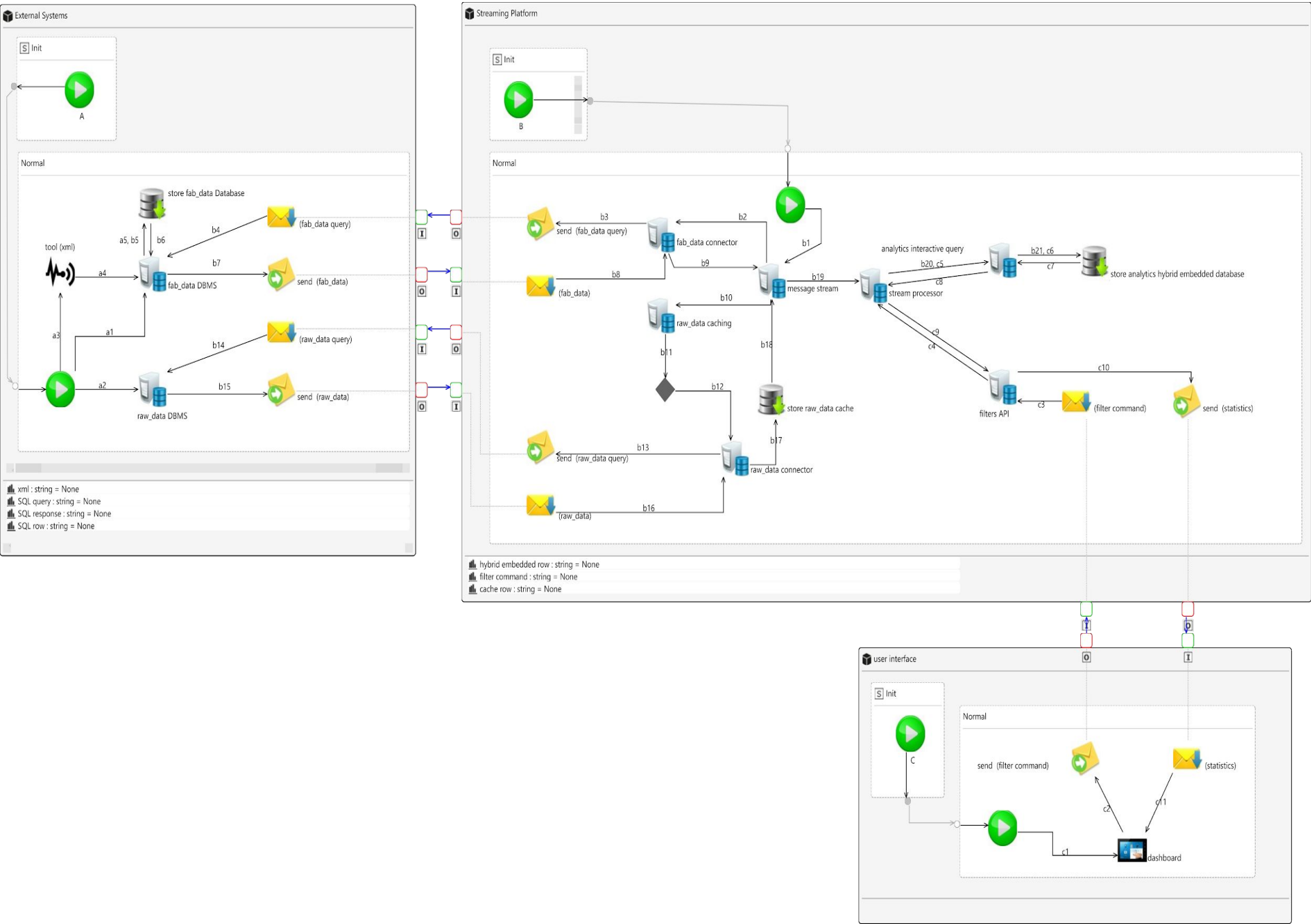
Sequence diagram - Kafka connect



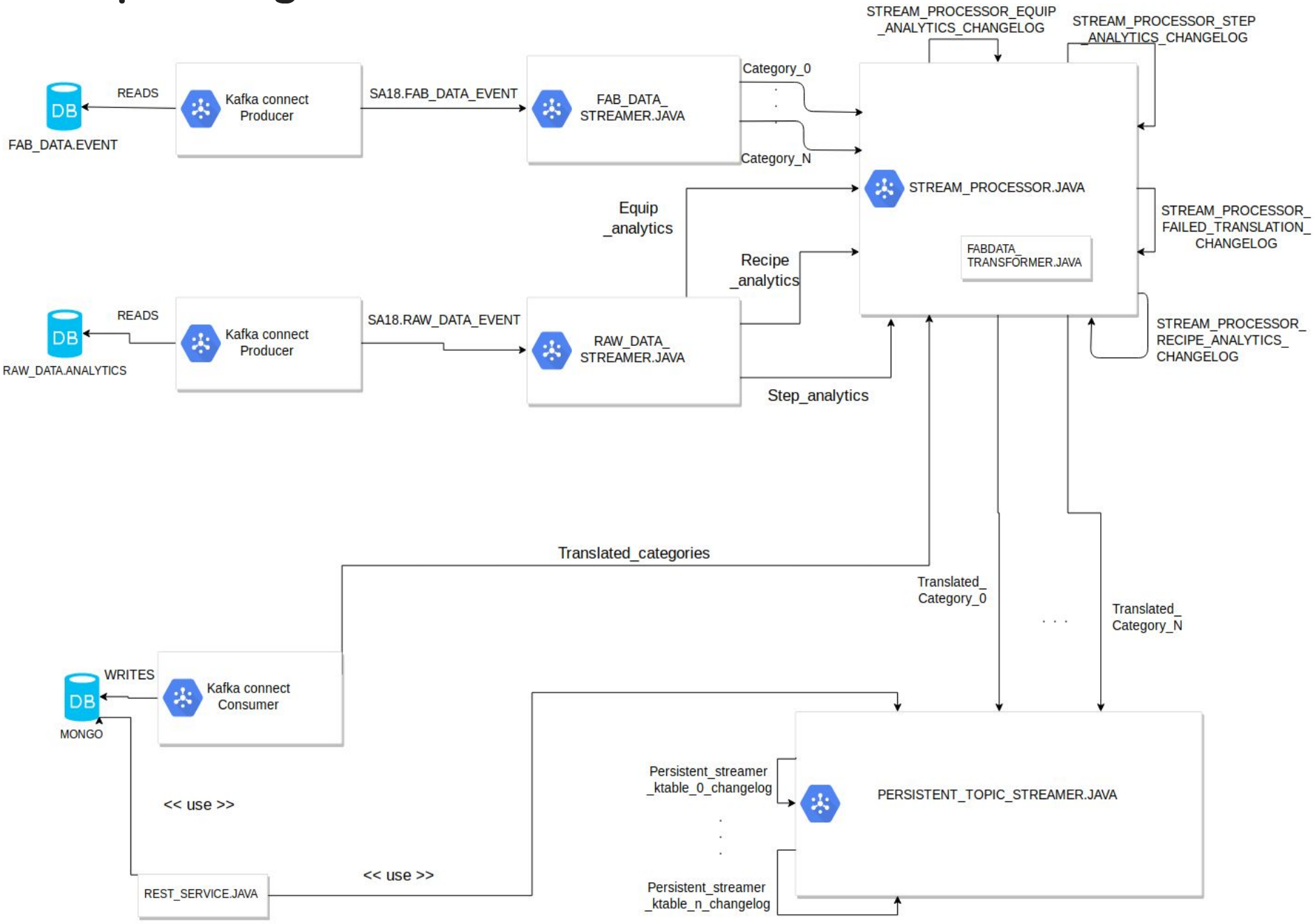
Sequence diagram - General



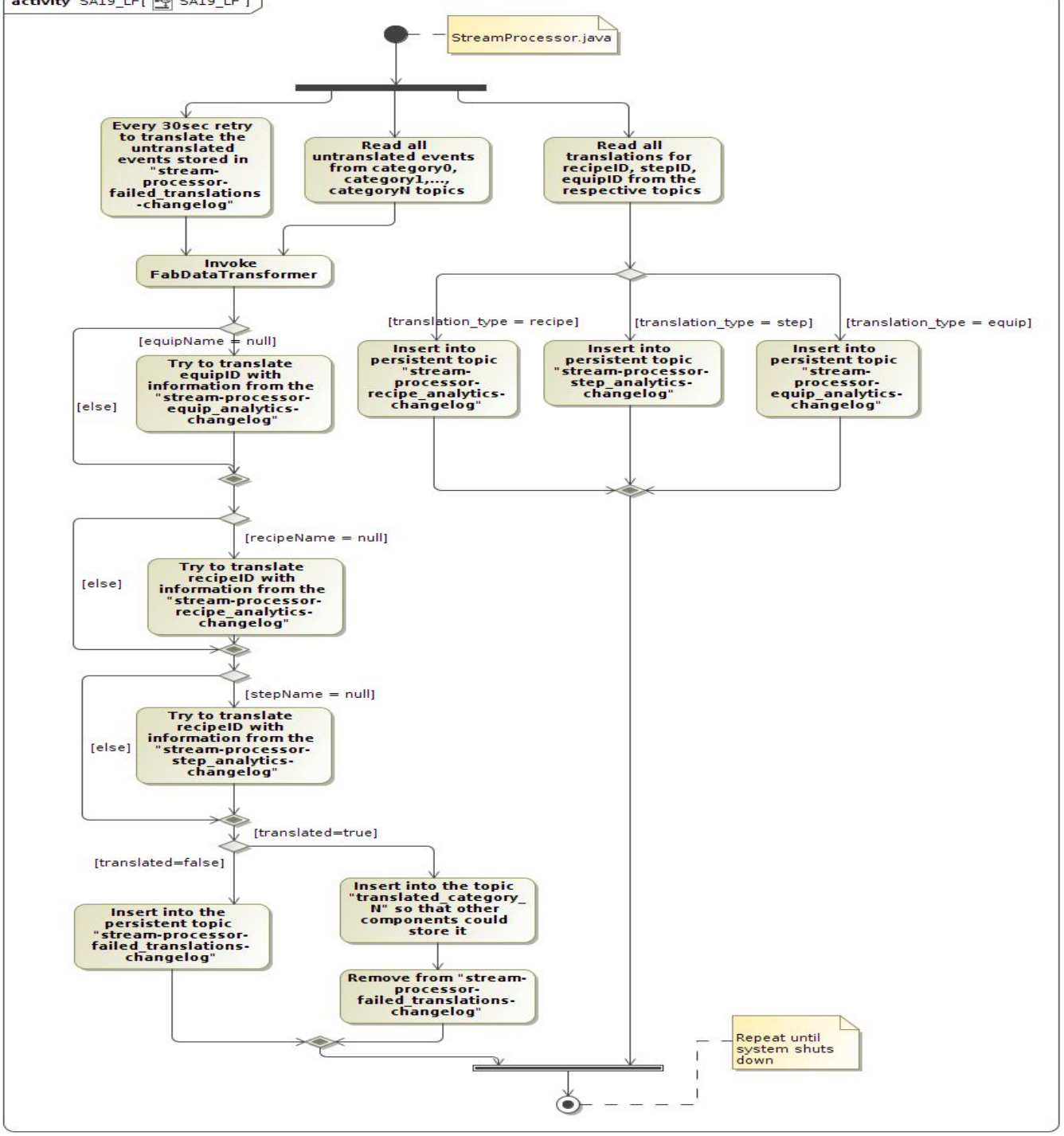
CAPS - SAML


External Systems
Init
A
Normal
store fab_data Database
(fab_data query)
tool (xml)
a5, b5
b6
b4
a4
fab_data DBMS
b7
send (fab_data)
a3
a1
b14
(raw_data query)
a2
b15
send (raw_data)
raw_data DBMS
xml : string = None
SQL query : string = None
SQL response : string = None
SQL row : string = None
Streaming Platform
Init
B
Normal
send (fab_data query)
b3
fab_data connector
b2
b1
b8
b9
(fab_data)
b19
message stream
analytics interactive query
b20, c5
b21, c6
c7
c8
stream processor
store analytics hybrid embedded database
b10
raw_data caching
b11
b18
c9
c4
c10
b12
store raw_data cache
filters API
c3
(filter command)
send (statistics)
b13
b17
send (raw_data query)
raw_data connector
b16
(raw_data)
hybrid embedded row : string = None
filter command : string = None
cache row : string = None
user interface
Init
C
Normal
send (filter command)
(statistics)
c2
c11
c1
dashboard

Topics diagram



Translation Logic



Sample of execution

Play the video…