Karlsruher Institut für Technologie

Real-time Visualization of Analyzed Industrial Communication Network Traffic

FAKULTÄT FÜR INFORMATIK

The Background

- Industrial Network Security – want to understand the traffic
- Analysis of the traffic
- Real-time visualization to help the user understand
- Incidents can be detected visually

The System

- Network traffic is recorded
- Traffic data is analyzed (dissected)
- Data is fed to a streaming platform (Kafka)
- A visualization tool displays data and analysis results

The System

- Network traffic is recorded
- Traffic data is analyzed (dissected)
- Data is fed to a streaming platform (Kafka)
- A visualization tool displays data and analysis results

Requirements

- Functional Requirements
  - 21 „must“
  - 4 „should“
  - 3 data requirements
- 12 Non-Functional Requirements
- 15 Testcases

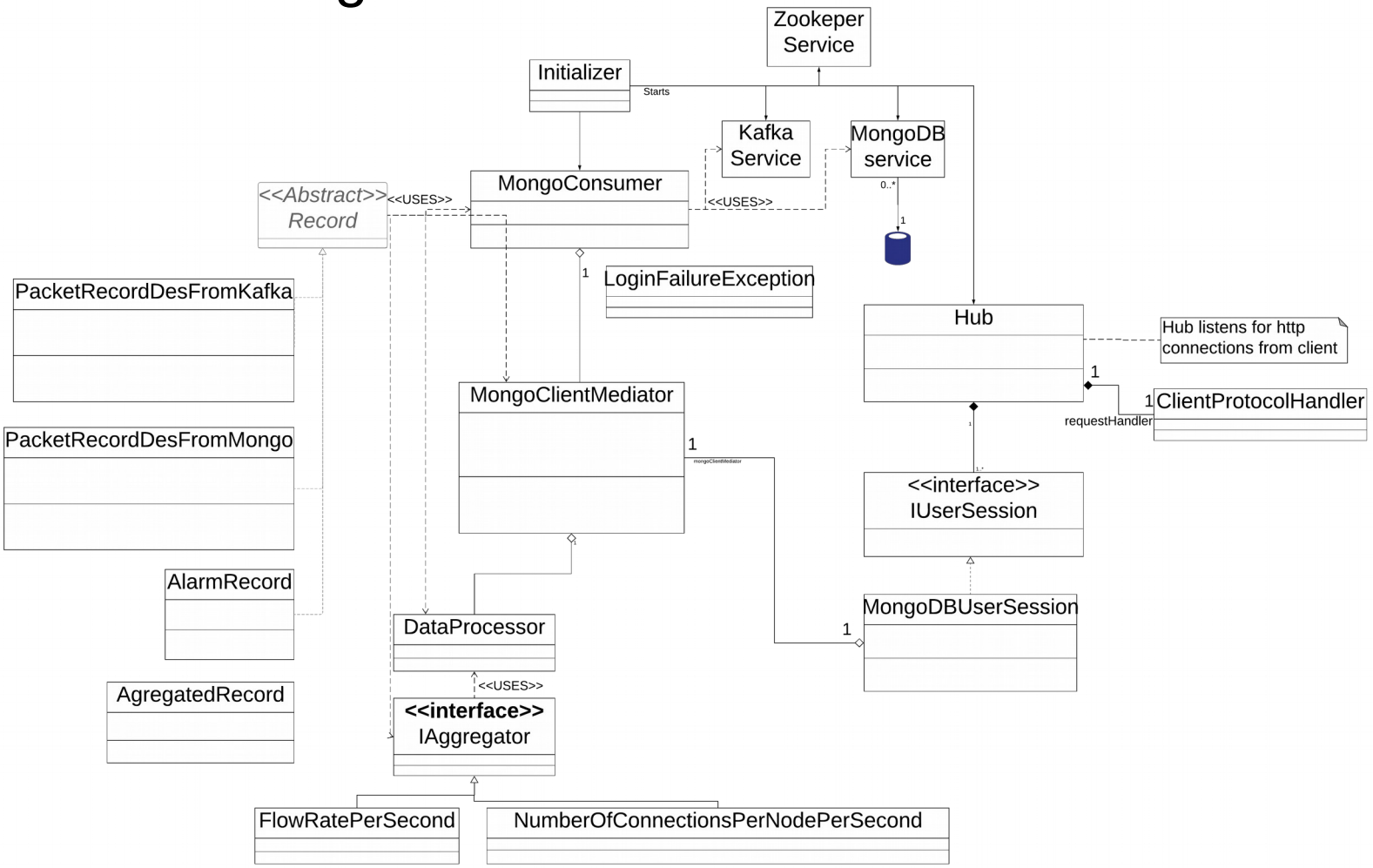
Architecture & Design

- Client-Server Architecture
- Back-End:
  - Mediator pattern
  - Strategy pattern
- Front-End:
  - Model-View-Controller
  - Observer

Back-end components

- Kafka streaming server
- MongoDB for both archived and real-time data
- Network hub realized as servlet running in Apache Tomcat
- A Mediator component for the database accesses
- Written in Java
- Kafka, MongoDB and Tomcat are open source

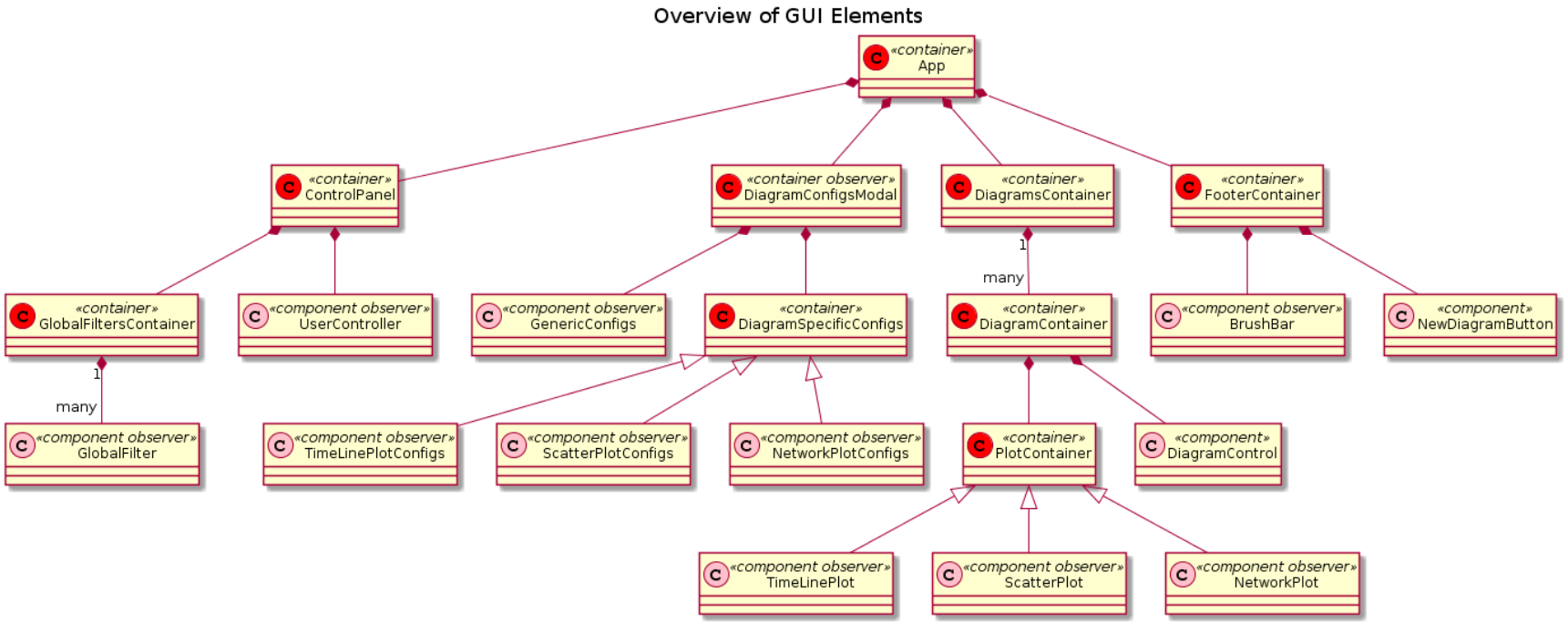
Design


Zookeper Service
Initializer
Starts
Kafka Service
MongoDB service
MongoConsumer
<<Abstract>> Record
<<USES>>
<<USES>>
0..*
1
1
LoginFailureException
PacketRecordDesFromKafka
Hub
Hub listens for http connections from client
1
1
ClientProtocolHandler
requestHandler
MongoClientMediator
PacketRecordDesFromMongo
1
<<interface>> IUserSession
AlarmRecord
DataProcessor
MongoDBUserSession
1
AgregatedRecord
<<USES>>
<<interface>> IAggregator
FlowRatePerSecond
NumberOfConnectionsPerNodePerSecond

Front-End

- Written in Javascript
- React library
- D3 graphics library
- nivo diagram components
- MobX state management
- All of the above are open source

Design


Overview of GUI Elements
«container» App
«container» ControlPanel
«container observer» DiagramConfigsModal
«container» DiagramsContainer
«container» FooterContainer
«container» GlobalFiltersContainer
«component observer» UserController
«component observer» GenericConfigs
«container» DiagramSpecificConfigs
«container» DiagramContainer
«component observer» BrushBar
«component» NewDiagramButton
«component observer» GlobalFilter
«component observer» TimeLinePlotConfigs
«component observer» ScatterPlotConfigs
«component observer» NetworkPlotConfigs
«container» PlotContainer
«component» DiagramControl
«component observer» TimeLinePlot
«component observer» ScatterPlot
«component observer» NetworkPlot
1
many
1
many

Implementation

- Tools for Back-end development:
  - Eclipse
  - Maven
  - Junit
- Tools for Front-end development:
  - Visual Studio Code
  - Netlify (CD)
- Github
- Latex

Challenges

- Only four team members
- Many different technologies:
  - Java, Javascript
  - Kafka
  - MongoDB
  - React framework
  - D3 graphics library
  - Nivo graphics library

Unexpected Difficulties

- Javascript and its libraries are more functional oriented than OO
- Complexity of D3
- Nivo components have inconsistent features
- MongoDB idiosyncracies

Requirements fulfilled

- „Must“ requirements:
  - 16 of 20 implemented
  - 1 dropped
- „Should“ requirements
  - 3 of 4 implemented

Lessons Learned

- Design more thoroughly
  - Especially data structures
- Plan and schedule more strictly
- Waterfall model didn‘t work well

Conclusion

- We produced a working system
- Usable as base for future work and extensions
- Gathered experience with teamwork