Software Architecture & Midterm Review

Recitation 5 - September 30th

Software Architecture

The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both.

–Software Architecture in Practice

Architectural Drivers

Design decisions are made to meet the requirements of drivers:

- Functional Requirements
- Quality Requirements
  - Performance
  - Availability
  - Scalability
  - ...
- Technical and Business Constraints

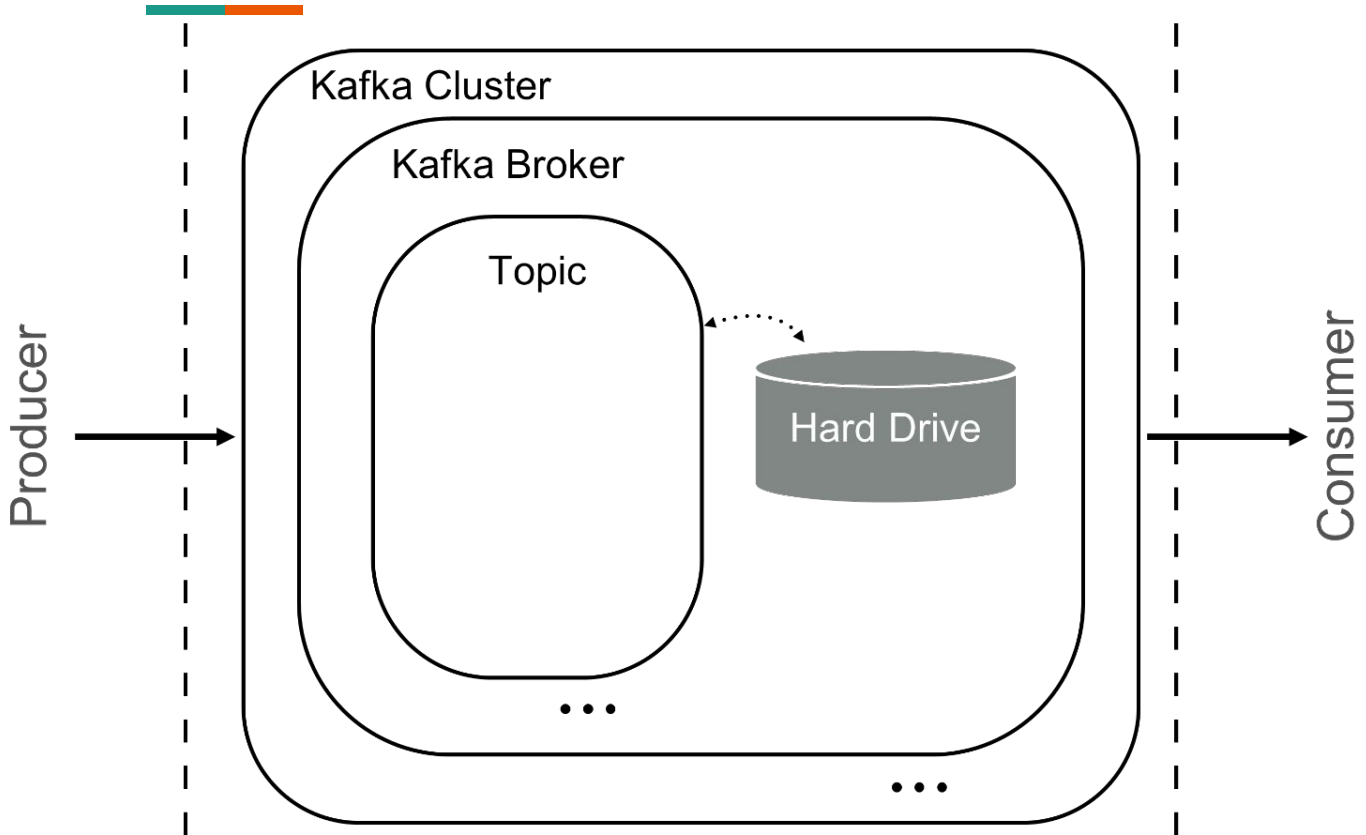
Quality Attributes -> Design Decisions

Driver	Requirement	Decisions
Availability	System should have x% availability	Have redundant copies of service behind load balancer. (How does this meet req.)?
Scalability	System should be able scale with doubling of number of requests but still have same latency	Have redundant copies of service behind load balancer. (How does this meet req.)?
Performance	System should have latency of under x seconds	Add caching for frequent requests

Architectural Diagrams

- Visual representation of the architecture; useful for documentation/communication
  - Notations
  - Informal, UML, etc.
- We do not require a particular notation
- We do require
  - **Clear & consistent meanings for symbols with a legend**
  - Accompanying text if the diagrams alone are not sufficient.

Example: Apache Kafka



Legend

System Boundary

Software Component

Sub-component

Repetition

Data Store

Message

Read/Write

Signs of Bad Documentation

- All lines look the same (arrows mean nothing or could mean many things).
- Inconsistent use of notation
- No key/legend
- Too little or too much detail
- Implementation details mixed with architectural abstractions
- Missing relationships between elements
- For accompanying textual documentation
    - Incomplete prose/rationale
    - **No discussion of alternatives**

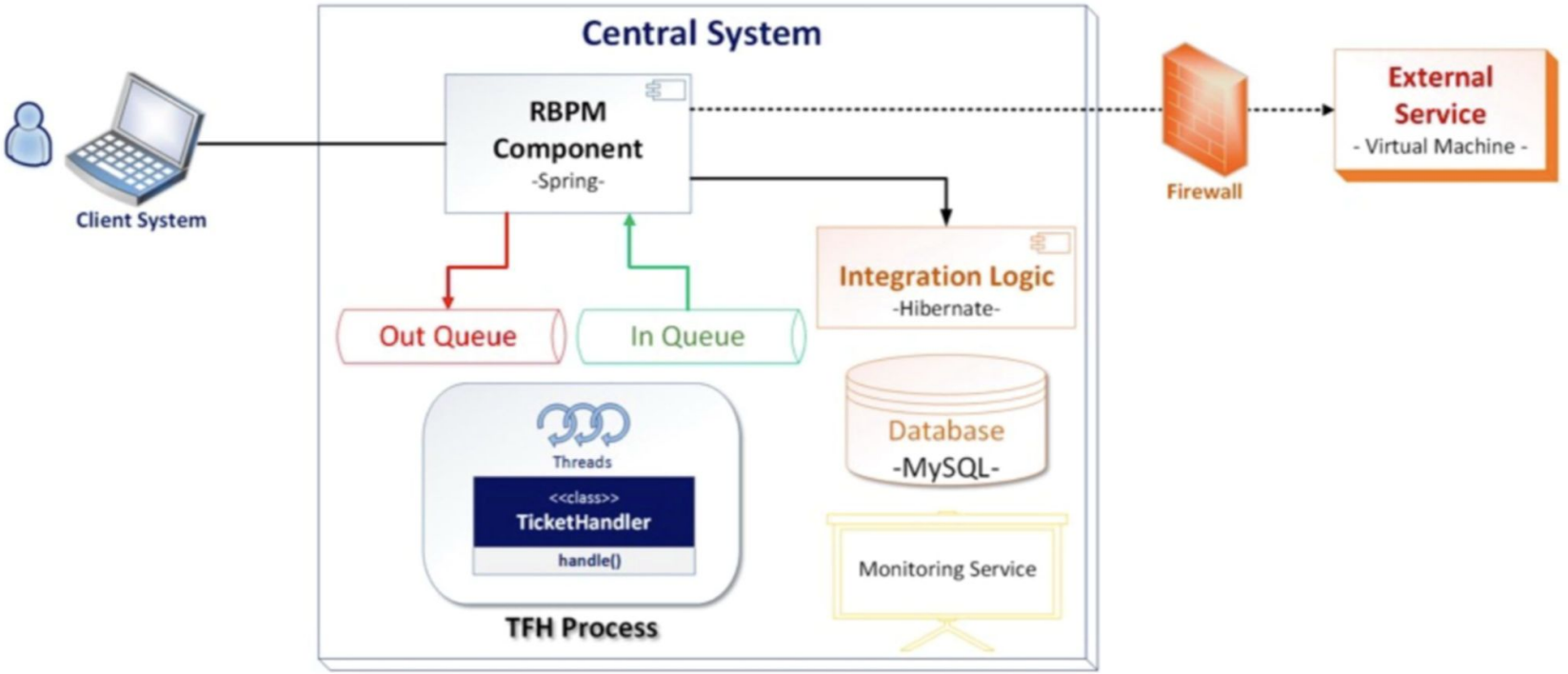
Signs of Bad Documentation

Next, we'll look at some examples.

For each example

- What can you understand from this diagram?
- What qualities are important?
- What is done well?
- What could be improved?

Example 1


Central System
RBPM
Component
-Spring-
Client System
Out Queue
In Queue
Integration Logic
-Hibernate-
Threads
<<class>>
TicketHandler
handle()
TFH Process
Database
-MySQL-
Monitoring Service
Firewall
External
Service
- Virtual Machine -

Example 2

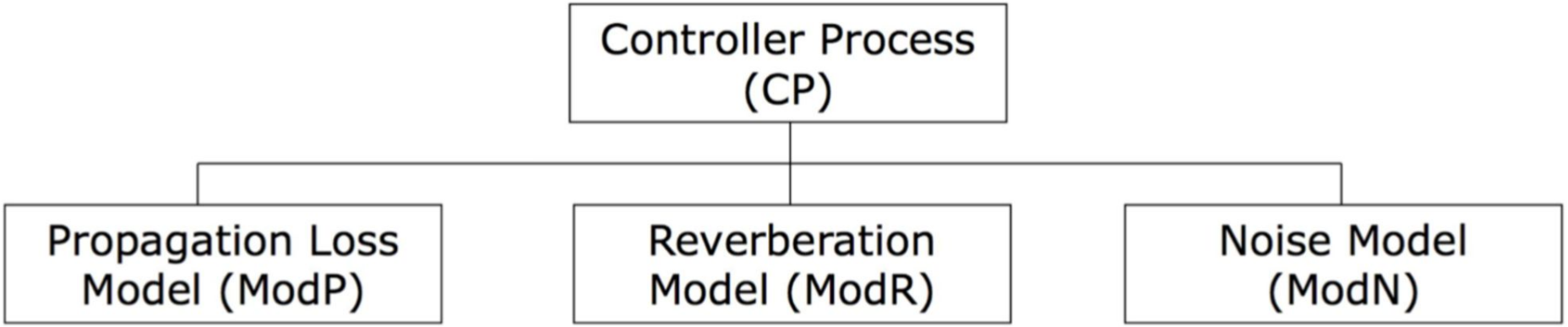


Figure X: Overall Software System Structure

Activity: Doorbell Camera

You work for a company that makes doorbell cameras. The doorbell camera market is very crowded, and you want to add a feature that differentiates your product from the competition. Management has decided the best way to do this is to add customization to real-time doorbell activity alerts. The feature will use computer vision to identify people approaching the door. It must:

- mention if the person is in your contacts (with a similar photo)
- mention if the person is someone it's seen before
- mention if a the person approaching is carrying a package
- mention if the person is unknown and not carrying a package

Most of your doorbells operate on a battery, though some high-end models have a power supply. Your doorbells connect to a smart home “hub” device (e.g. Google Home or Amazon Echo) over WiFi, which in turn is connected to a cloud-based platform that handles routing notifications to homeowners via an app.

Activity: Doorbell Camera

Your job is to design the architecture for this system.

1. Discuss ideas and ***tradeoffs*** between design choices.
2. Choose one design and draw an ***architecture diagram***.
3. ***Justify*** why you chose this design.

Midterm Review

https://github.com/ckaestne/seai/tree/F2022/exams