

# Advanced Systems Lab - Design

Lukas Elmer, Matthias Ganz

September 26, 2013

# Table of content

Design Choices

Overview

Messaging System

Database

Client

Request

Response

# Design Choices

- ▶ Every client has a private queue
- ▶ Private messages can only be sent to private queues
- ▶ Every queue is handled by a specific host, this allows caching
- ▶ A message sent to multiple queues is equal to sending multiple messages with the same content to specific queues
- ▶ Use Reactor or Leader/Follower design pattern
- ▶ Use Java conventions and guidelines, like naming :-)

# System Overview

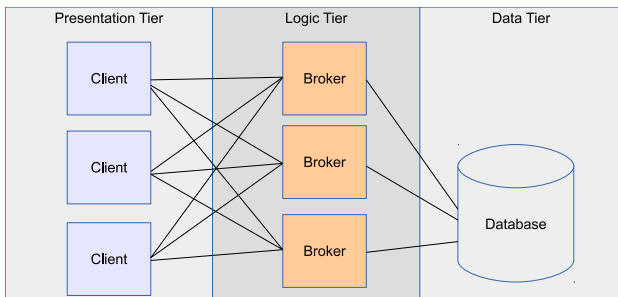


Figure : System Overview

# System Overview

- ▶ The messaging system utilizes a single database instance.
- ▶ On the logic tier multiple broker instances may be running.
- ▶ Each broker is responsible for certain queues.
- ▶ Clients may ask any broker about who handles requests for a specific queue.
- ▶ A client may connect to any number of brokers depending on which target queue it wants to send messages.

# Broker Threading

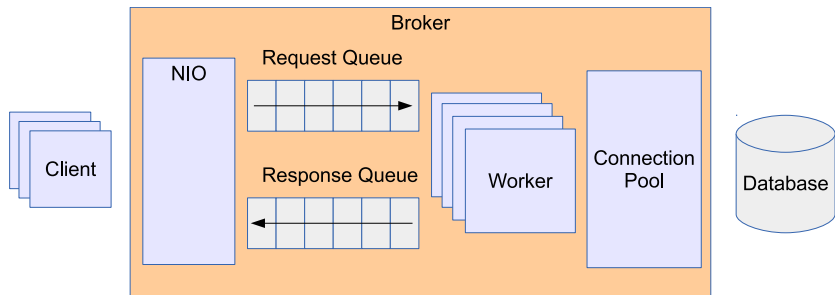


Figure : Broker Threading

# Database Schema

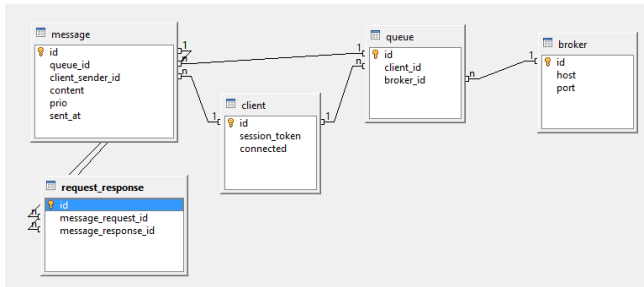


Figure : Database Schema

# Client to Broker Communication

- ▶ Header: Length of the Java object in bytes.
- ▶ Body: Serialized Java object (Request, Response).
- ▶ Serialize POJO's and send it over the network
- ▶ Connection: keep alive, connection pool
- ▶ Security: no authentication, no encryption



# Client

- ▶ Clients may have different configurations depending on the test case currently performed.
  - ▶ Only send messages
  - ▶ Only read messages
  - ▶ Only do request/response
- ▶ Management interface in HTTP
  - ▶ To start/stop the current action
  - ▶ To collect statistics
- ▶ Use any Browser as Management Interface

# Request

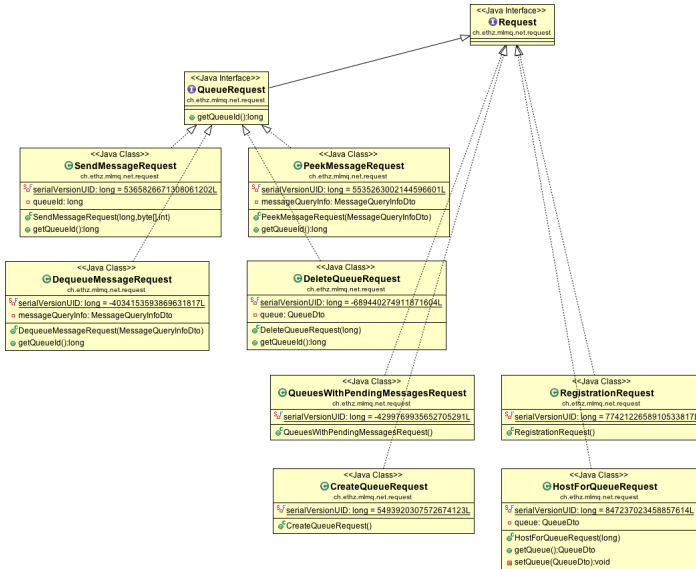


Figure : Request Class Diagram

# Response

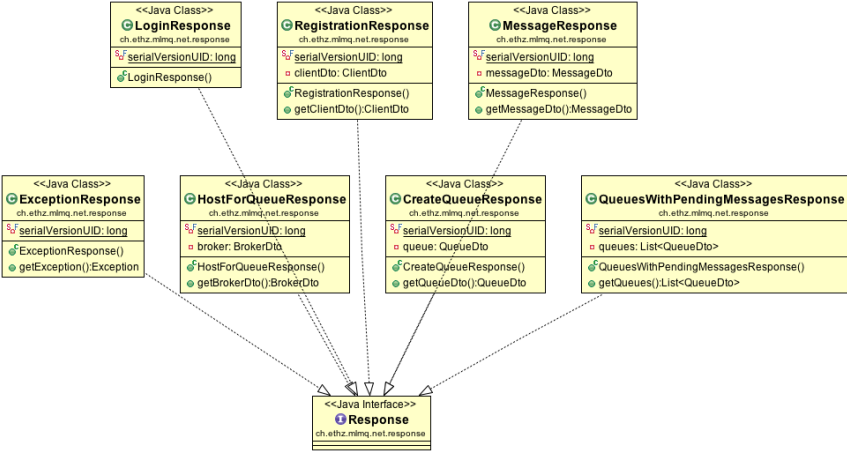


Figure : Response Class Diagram