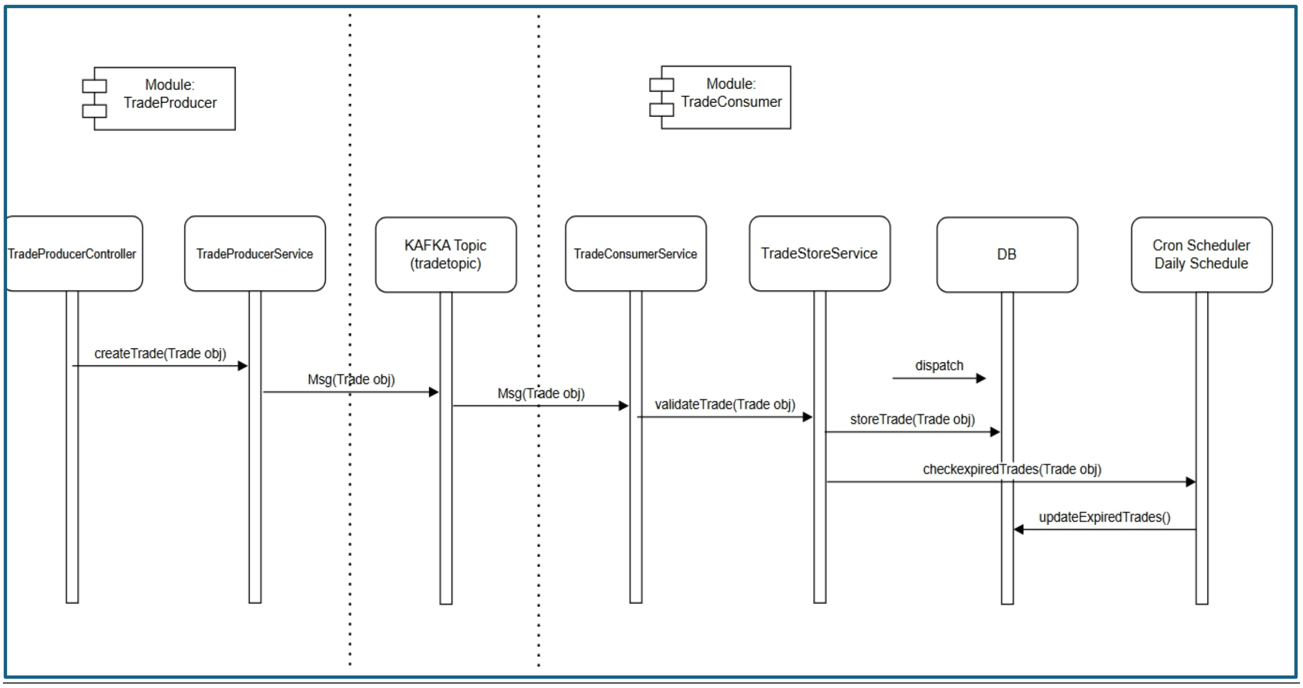
Trade Store Solution-Sanjay Phadke

GitHub: https://github.com/sanjayphadke/tradeprocessor

Technology Stack

1) Spring Boot 3.4.3
2) Java 21
3) H2 DB (IN memory)
4) Kafka Broker deployed locally on Docker Desktop

Architecture and Workflow



Key Application Components:

1. TradeProducerController-REST Controller for publishing Trade message through TradeControlelrService on Kafka Topic-trade Topic
2. TradeProducerService - Service for publishing Trade message on Kafka
3. TradeConsumerService- Subscribes to Kafka Topic and consumes message.
4. TradeStoreService - Service for performing necessary validations on incoming Tade message and store in H2 DB. This also has a scheduler (Cron Job) set to run at midnight daily to scan all Trades in the DB and mark the Trades as Expired for those whose Maturity Date is less than current date.

Key Validations performed on the incoming Trade:

1. If a trade with a lower version is received during transmission, the store will reject it and generate an exception. Trades with the same version will replace the current record.
2. The store must reject any trade that has a maturity date earlier than today's date.
3. When a trade's maturity date is surpassed, the store should automatically mark the trade as expired.

Sample Trade Structure:

```
{ "tradeId": "T1",

"version": 1,

"counterPartyId": "CP-1",

"bookId": "B1",

"maturityDate": "2025-05-20",

"createdDate": "2025-03-15",

"expired": false

}
```