Team Name : Red Chilly

Deepthi Devaiah D - 65848248
Sachin B Raj - 28568830
Vinu K S - 11043792

Transaction Manager for Policies

In this project we are implementing a transaction manager to handle policies. The input to our module is the policy data from TIPPERS. From our point of view, a policy is equivalent to a CLOB/ BLOB field which consists of policy information along with its validation period. We plan to implement the following schema:

Policy ID(CLOB/BLOB)	Version(Integer)	Validity(Boolean)	From(Timestamp)	To(Timestamp)

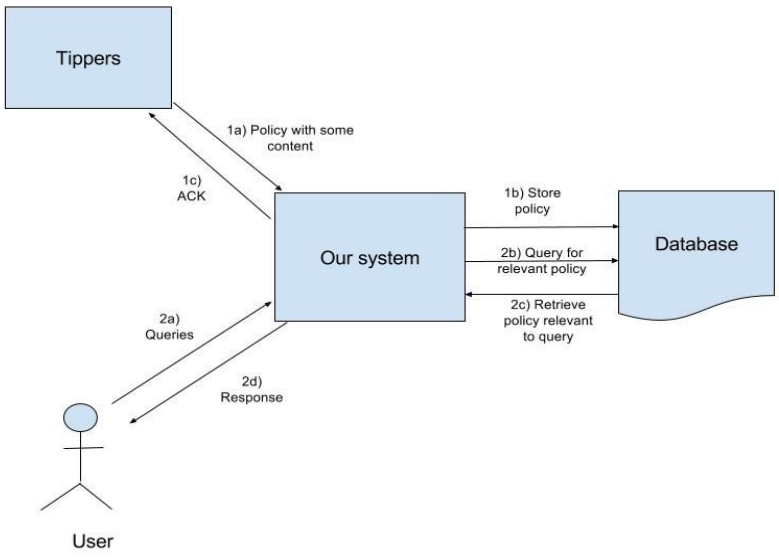
Suppose there is a policy P_0^1[content] where P_0 is the policy ID and 1 is the version and content is the policy data. Several operations are possible at the transaction level of policies.

- User can **create a new policy** called as P_2^0[content] at some timestamp.
- Existing policy can be **updated** by the user. After the policy gets updated it becomes P_2^1[content] where 1 is the version, after getting updated. This will have updated values for policy validity period. *Validity* is a field that depends on the current timestamp and the *to* and *from* columns. If *to* and *from* time is not mentioned, we will mark them as infinity and -infinity respectively.
- **Rollback to older version**: Policies can be reverted to an older version. This will also help if we want to go back in time for some information which we need.

Example: If we want to set the policy to a version on a particular date last month, then we will have to get the highest version of the policy which is valid on that particular day (assuming granularity of hours). We can implement timestamp based protocol to achieve this.

All information about policies is maintained in a database. Also, we need to maintain an in-memory copy of the working set of policies. A batch process can periodically check the validity of all policies and mark them accordingly. If some policies are invalid then we need to remove it from the in-memory copy of the working set.

Architecture Design



TimeLine

Week 4 : Discuss with TIPPERS team, finalise on the requirements and datamodel
Week 5 - 7 : Implementation of our module Week 8 : Testing