CarRentalGAE-session1

Stien Vanderhallen, Joke Denissen

November 2018

1 All-or-nothing semantics

We did not use cross-group transactions, because we have implemented an *Agency* class, which is the root for all the car rental companies and thus all the cars and car types. The transactions to confirm quotes thus only interact with one entity group instead of one per car rental company. We implemented it this way because cross-group transactions can only deal with a maximum of 25 entity groups and we would have had 32 entity groups, one for each car rental company.
We did rely on JPA transactions.
We have only one entity group so it is not easy to make the system distributed. If there would have been less than 25 car rental companies, we would have used cross-group transactions to make the system more distributed.

2 Client profiles

We would still apply the same design decision when dealing with client profiles. The clients are all child entities of the same agency so they can interact easily with all the car rental companies without using cross-group transactions.

3 Transactional semantics

3.1 Transactions only

When relying on transactions only, and thus on cross-group transactions, those transactions guarantee atomicity and thus no concurrent changes can be made while a transaction is executing.

3.2 Application logic

When using application logic to rollback, atomicity is not guaranteed and thus concurrent changes can be made to a particular set of data. In application logic, there would be an iteration over all reservations when a rollback occurs, which could interfere with concurrent iterations. Atomicity could thus not be guaranteed.

3.3 Preferred alternative

For cloud applications, relying on transactions would be preferable because of the concurrency control. Application logic would have a higher responsiveness in comparison to cross group transactions, because those use a two-phase-commit protocol. The use of transactions that are limited size would thus be the ideal compromise for cloud applications.