Advanced Java

Trainer: Nilesh Ghule

Primary Keys

Auto-generated PKs

```
@GeneratedValue(strategy= ~~)
@Id
```

① IDENTITY → auto 'inc column or identity column in db

② SEQUENCE → db seq or table with one row & one column to keep last generated id.

③ TABLE → dedicated table to keep track of generated ids of all tables.

④ AUTO → as supported by current db & dialect.

* default → user given values (assigned).

Composite Pri key

```
@Embeddable
class StdRoll implements Serializable {
    int std;
    int roll;
    ~~
}
```

Students

std	roll	name	marks

C.P.K.

```
@Entity
@Table(name="students")
class Student {
    @EmbeddedId
    StdRoll id;
    String name;
    double marks;
    ~~
}
```

```
interface StudDao
    extends JpaRepository
        <Student, StdRoll> {
    ~~
    ~~
}
```

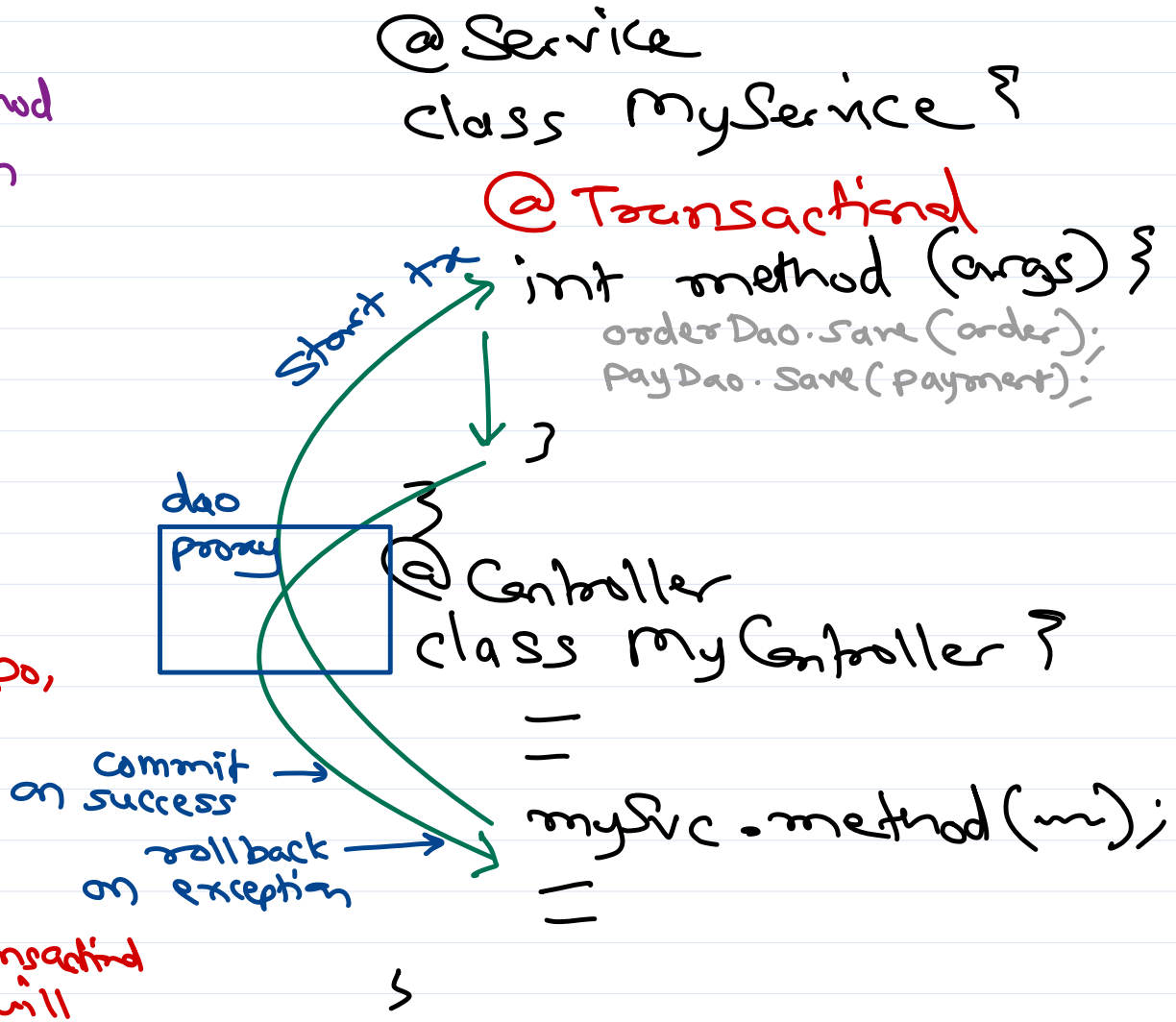
Spring Transaction Management

@Transactional annotation
- method-level i.e. tx for current method
- class-level i.e. tx for all methods in that class.

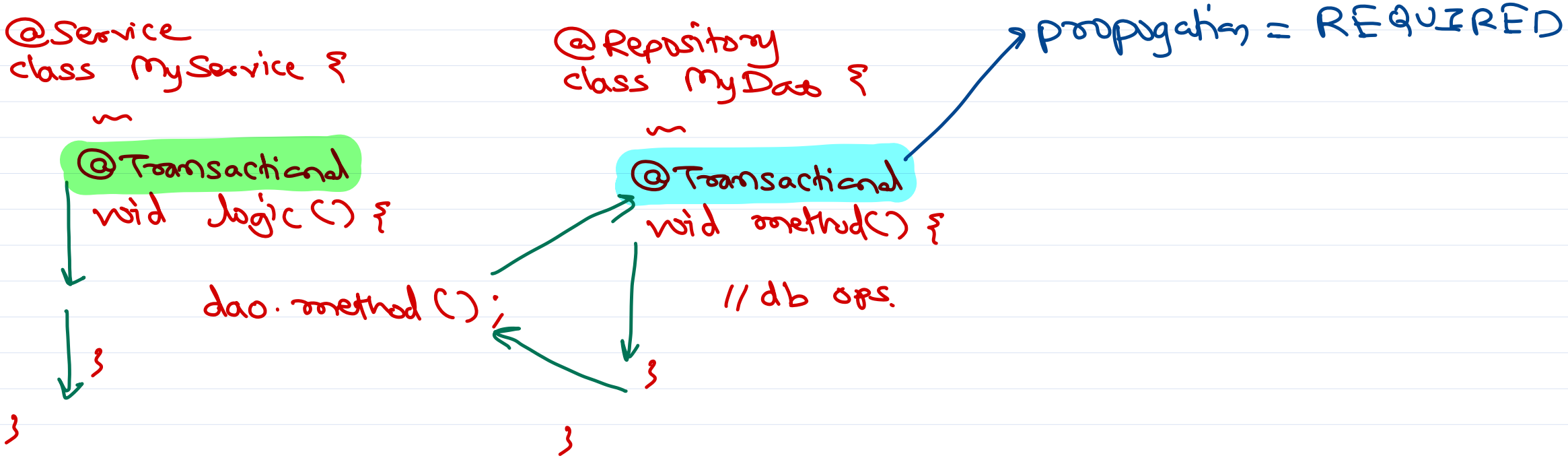
→ @Transactional 'internally use TxManager to start new tx at the start of method & commit/rollback tx at the end of method.

@Transactional is present on SimpleJpaRepo, so each Dao/Repo op create new tx & commit/rollback at the end (of op).

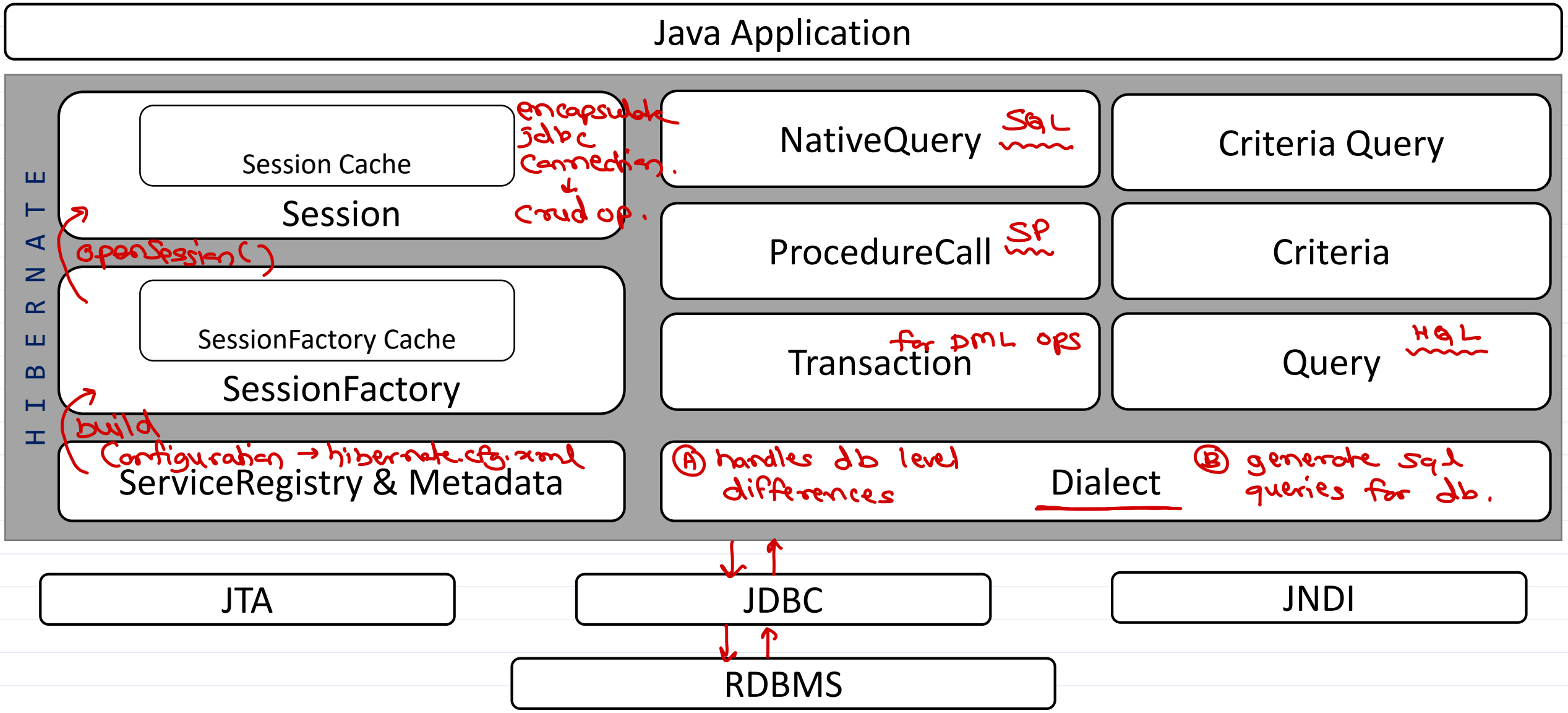
Usually one Business logic op (in service) needs multiple dao ops (from same or diff daos). It is recommended to put @Transactional on service layer, so that either all dao ops will be completed or all will be discarded (if any one fails)



Transaction propagation



Hibernate Architecture



Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>