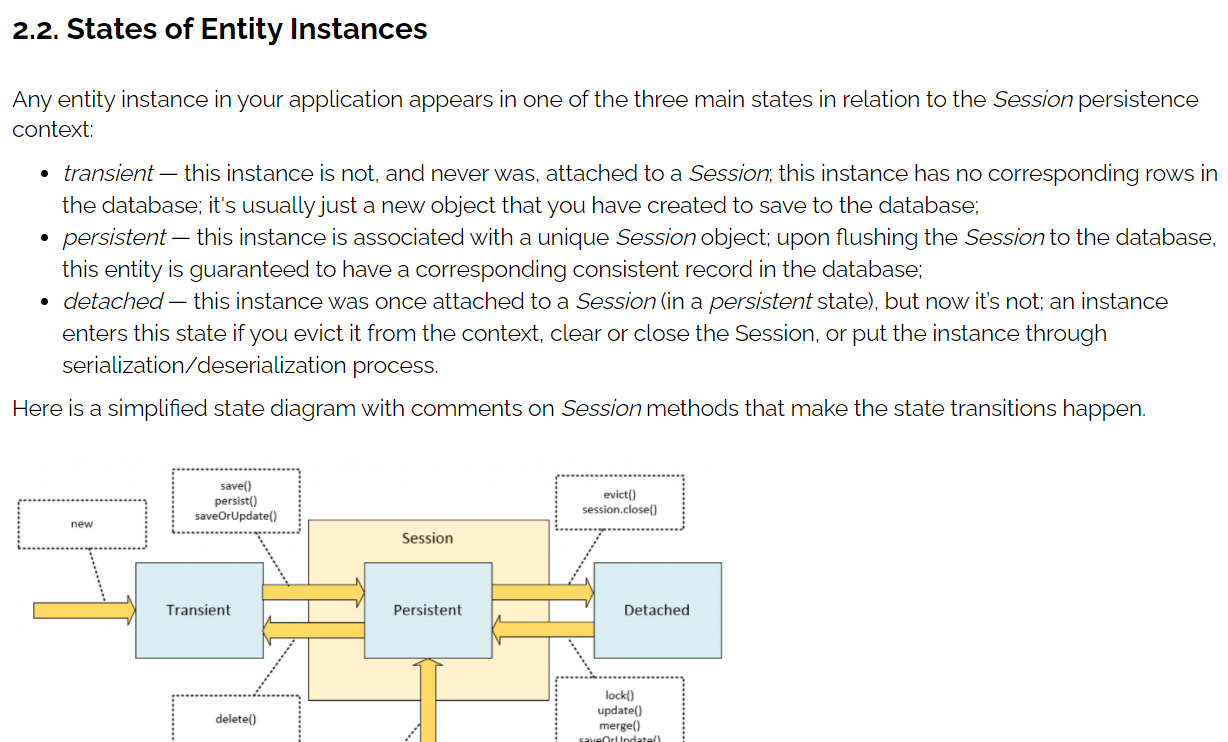
# **Entity Manager**



**Save vs Persist**

Its purpose is basically the same as persist, but it has different implementation details. The documentation for this method strictly states that it persists the instance, “first assigning a generated identifier”. The method is guaranteed to return the Serializable value of this identifier.

**SaveOrUpdate vs Merge**

* void saveOrUpdate(object)->object must be attached to a hibernate session (including all sub objects within the object), and once save/update is done, the object reflects the updated changes (e.g. primary key if saving a new object)
* Object merge(object)-> object does not have to be attached to a hibernate session. Once save/update is done, the object DOES NOT reflect the change. The returned object reflects the changes, and it is attached to hibernate session.
* MERGE method offers greater flexibility when it comes to saving data objects, since you need not worry about attaching object to Session.
* You can create an object, set Id, and other properties on your own, and save it without worrying about the Hibernate Session exception.
* If you’re using saveOrUpdate, the object saved MUST be attached to session. I believe many people have experienced this issue.
* This means you can create a copy of object from Service layer and just pass the object to your DAO. Hibernate takes care of MERGING the data to appropriate hibernate session attached object and saves the data.
* The only downside of using MERGE is that the object passed does not reflect the changed information. So, if you need to use the updated object, you must get it from the returned object, not the parameter object.

**JTA**