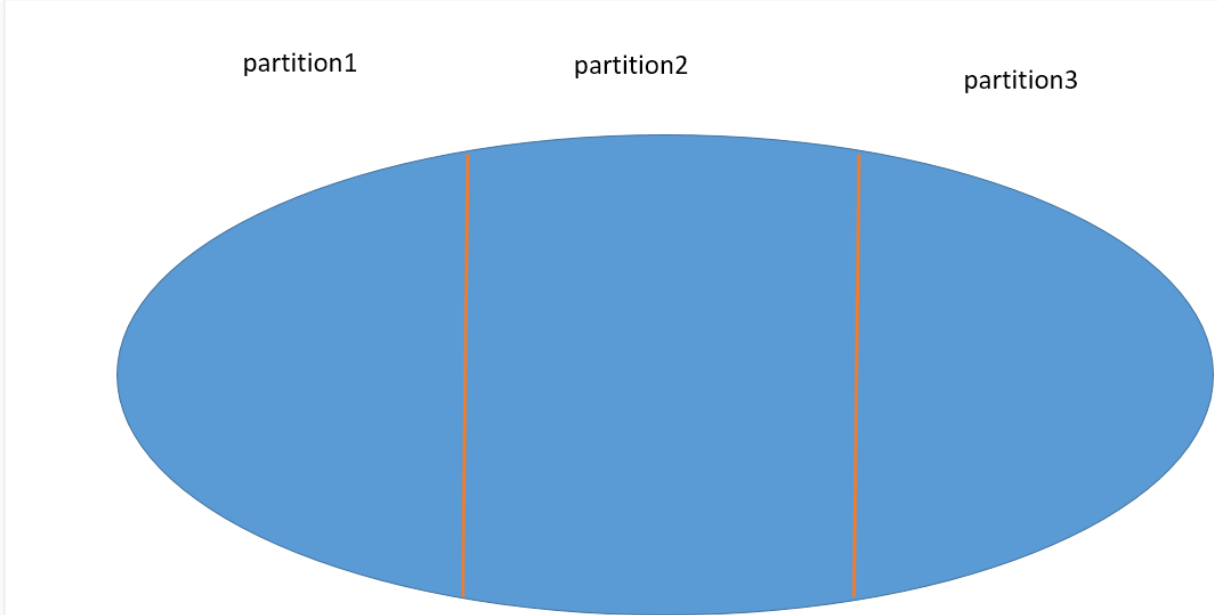


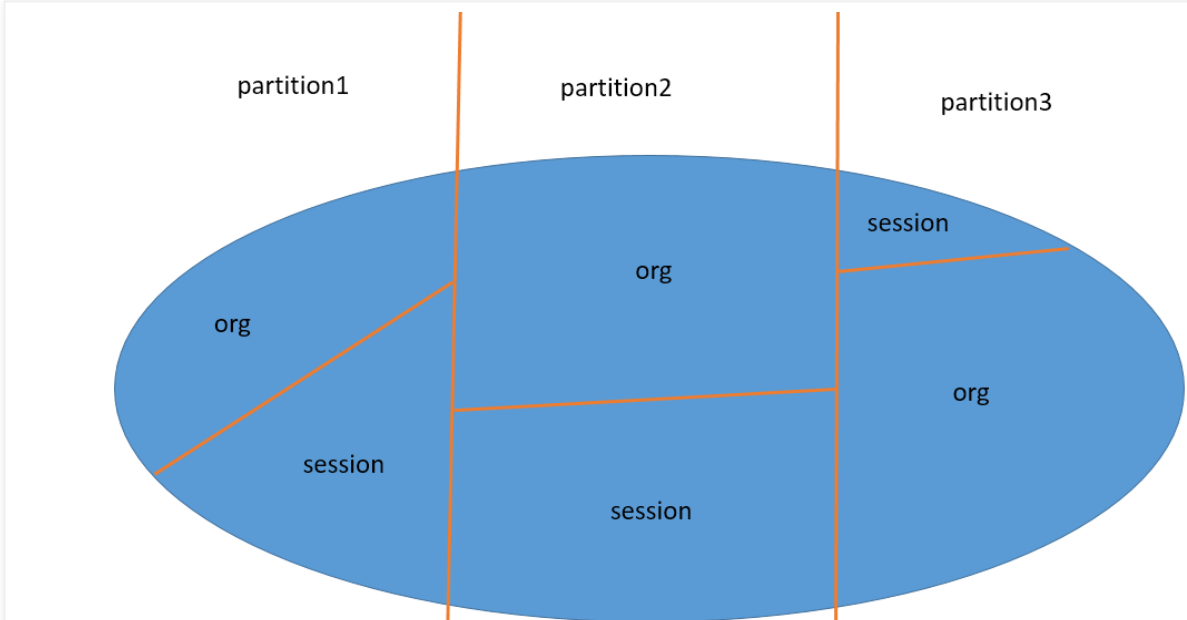
# Salesforce Platform Cache

Nothing new, just short summary on platform cache :

To over simplify, Platform cache is glorified hash map. Platform cache is first divided into different partitions. These are hard partition, cache usage in one partition will not overflow in another partition. Usually different partition is used for different project.



Partitions are further divided into Org cache (available to all users) and Session cache (used for user session cache). These are again hard partition, cache from Org will not overflow to Session. Minimum cache size is 5MB for Org or Session Cache.



## To store key in the Cache

```
1: User u = [ select id, username from user where username = 'chintan_shah@abc.com' ]
2: Cache.Org.put( 'local.MyPartition1.key1', u );
3: System.debug(' key1 is stored ');
4:
5: String name = 'Chintan';
6: Cache.OrgPartition MyPartition1 = Cache.Org.getPartition('local.MyPartition1');
7: MyPartition1.put('key2', name );
8: System.debug(' key2 is stored ');
```

- We can either put key directly using Cache.Org.put or get access to a specific partition using Cache.Org.getPartition
- To work with Session partition, Org would just change to Session in above code.

## To retrieve the key from Cache

```
1: Object u = Cache.Org.get( 'local.MyPartition1.key1');
2: System.debug(' key1 is stored ' + u );
3:
4: Cache.OrgPartition MyPartition1 = Cache.Org.getPartition('local.MyPartition1');
5: System.debug(' key2 is stored ' + MyPartition1.get('key2' ) );
```

## To clean the Cache

- Cache is not guaranteed persistence store, it can be cleaned any time by Salesforce
- Can also be wiped out during code deployment
- Session cache max TTL is 8 hours and Org is 24 hours
- Internally it uses LRU when it hits size limit to clean up old data

Hence, we don't ever have to do the clean up, but if we need to for certain reason, we can use below code. It also has limitation if cache is stored using cache builder.

```
1: for(String key : Cache.Org.getKeys() ) {
2:   Cache.Org.remove(key);
3: }
4:
5: for(String key : Cache.Session.getKeys() ) {
6:   Cache.Session.remove(key);
7: }
```

## Cache Builder

Instead of storing and retrieving cache, it is better to provide loading strategy to Platform cache, so upon cache miss, Salesforce automatically calls the class to load the cache for that key. This reduces the code and handles cache miss much more gracefully.

We have to specify cache loading strategy as class. Below is small class which loads the user information based on username. Idea is username is being the key, we need to load user data if not already in the cache.

```
1: /**
```

```
2:  * Created by chshah on 9/14/2017.
3:  */
4:
5:  public class UserInfoCache implements Cache.CacheBuilder {
6:
7:      public Object doLoad(String usernameKey) {
8:          String username = keyToUserName(usernameKey);
9:          System.debug(' UserInfoCache load usernameKey ' + usernameKey + ' userName ' +
10:             User u = (User)[SELECT Id, firstName, LastName, IsActive, username FROM User
11:             return u;
12:         }
13:
14:         public static String usernameToKey(String username) {
15:             return username.replace('.', 'DOT').replace('@', 'ATRATE');
16:         }
17:
18:         public static String keyToUserName(String key) {
19:             return key.replace('DOT', '.').replace('ATRATE', '@');
20:         }
21:     }
```

```
1: String usernameKey = UserInfoCache.usernameToKey('chintan_shah@abc');
2: User u = (User) Cache.Org.get(UserInfoCache.class, 'local.MyPartition1.' + usernam
3: System.debug( ' u ' + u );
```

- The reason for converting username to usernameKey, is special characters are not allowed in platform cache key.
- In line 2, we provide cache loading strategy, hence if key is not found, it will call our class to load the key.

Consideration

- ISV (managed package) can supply their own cache, hence it will use different namespace than "local"
- Cache put follows same transaction boundary as SOQL updates, so any rollback due to failure will not put data in cache
- Cache TTL limits (8/24 hours), and plus limit on how much data we can store per transaction (usually 1MB)