EJB Session & Entity Bean

Unit-III

Session EJB

- A session bean instance is:
  - A non-persistent object
  - Implements some business logic
  - Runs on the server
  - Live as long as the client need them

Constraints on Session Bean

- EJB cannot use threads – but container can run multiple instances
- Cannot directly access transaction manager – container is responsible for managing transactions
- Cannot use JDBC commit and rollback – container issues commit and rollback
- Not allowed to change security identity at runtime
- Cannot have static variables – it must be static final

Session bean constraints

- It is irrevocably disappear from the server
  - If timeout value expires
  - If the server crashes, shutdown, or restarted
- Session beans are non-reentrant – another call to the same object from same transaction context throws remote exception

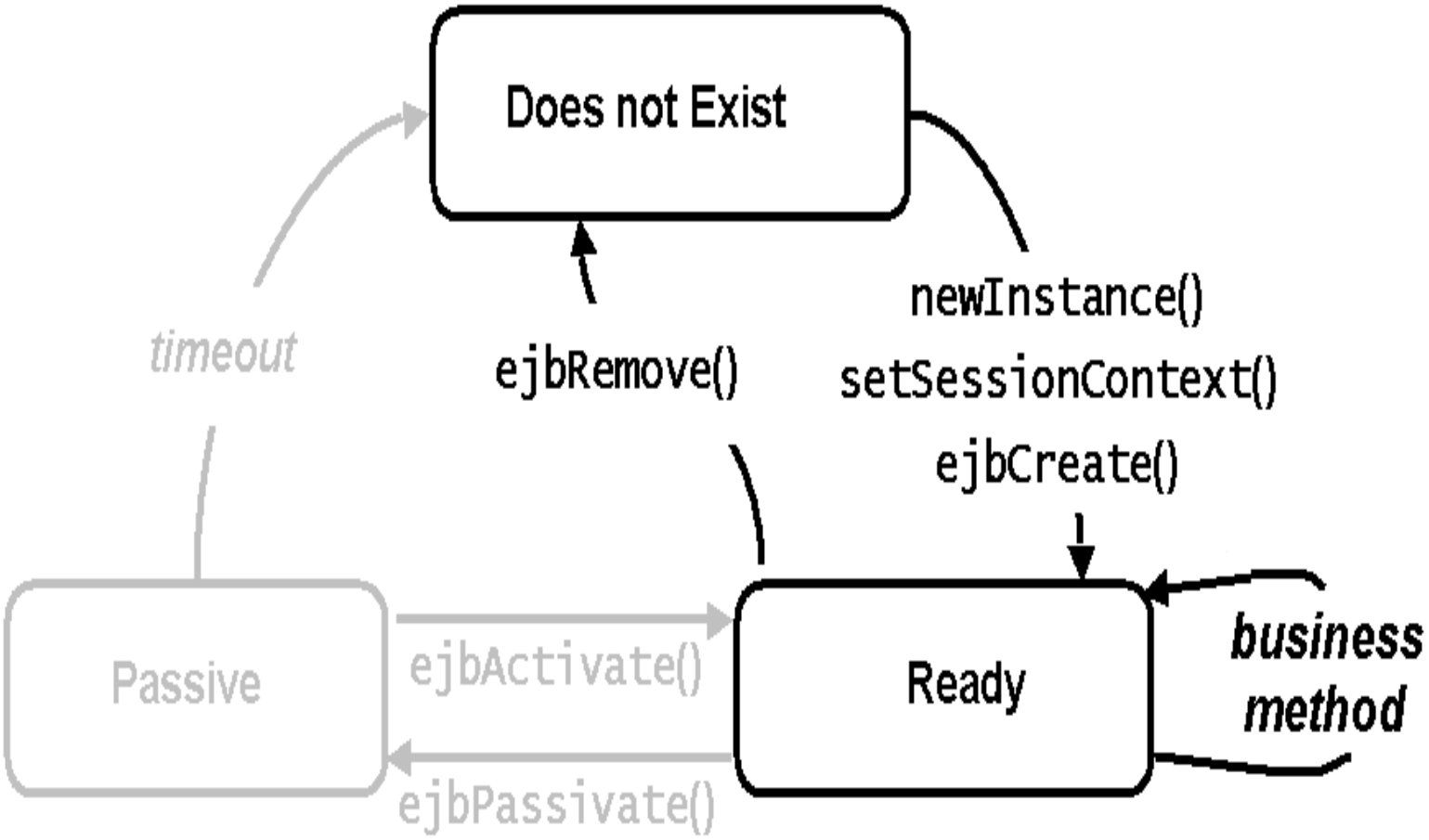
Components of a Session EJB

- The remote interface
  - Extends EJBObject (All extended types are from javax.ejb)
- The home interface -Extends EJBHome
- The bean class itself, called *X*Bean
  - Extends SessionBean
  - Should implement java.io.Serializable (for stateful beans)
  - Implements business methods from the component interface
  - Implements *lifecycle* methods (ejb*XXX*)

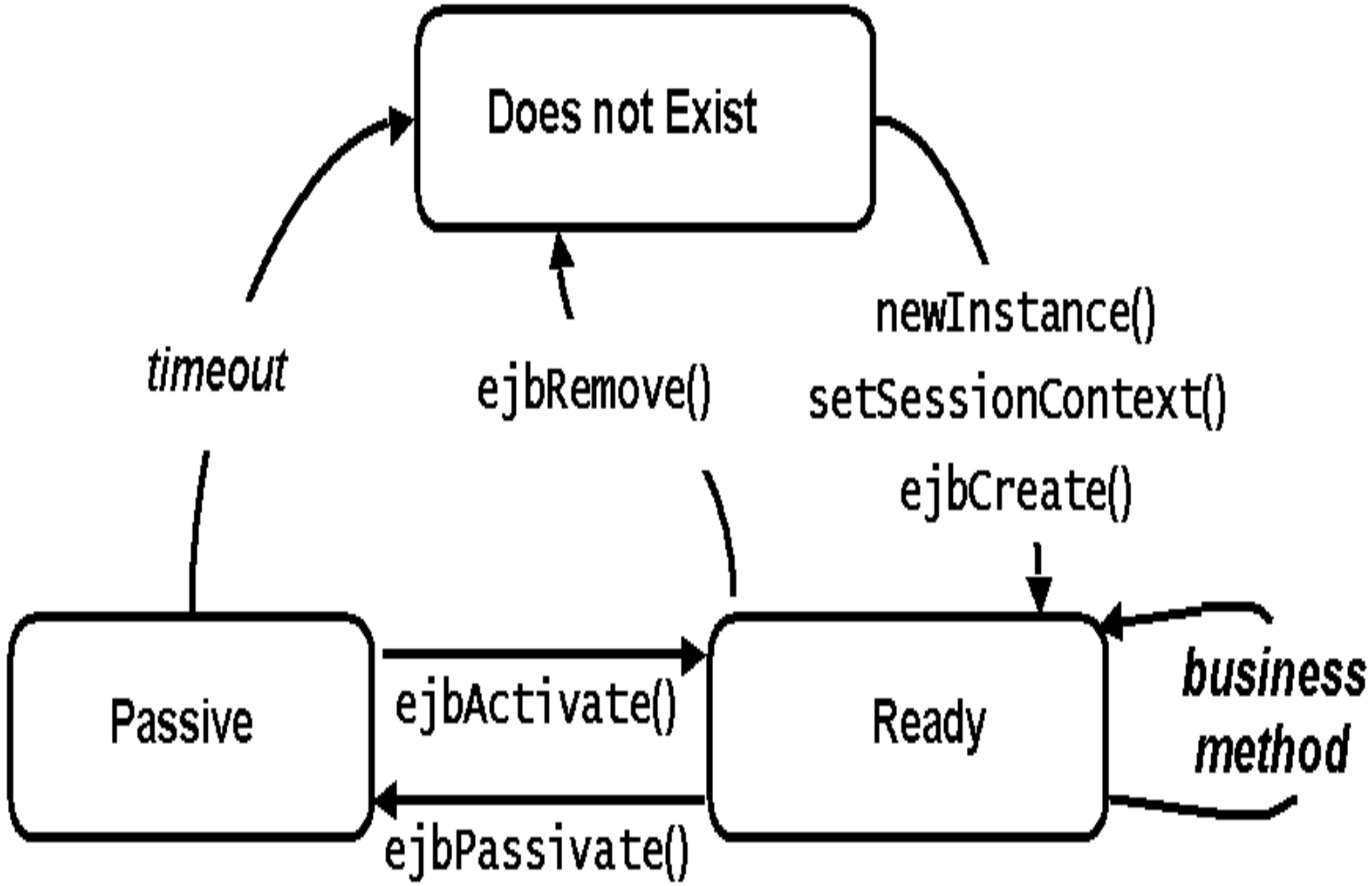
Stateless Session bean life cycle

- Nonexistence
- Method-ready state

- Session EJBs have no state, so activation and passivation are meaningless
  - The container simply destroys instances when low on memory, and creates new ones as needed
  - Still, ejbActivate and ejbPassivate must be implemented in *X*Bean (as empty methods)



Stateful session bean



Stateful session bean

- The state of an object consists of the values of its instance variables.
- In a stateful session bean, the instance variables represent the state of a unique client-bean session.
- Because the client interacts ("talks") with its bean, this state is often called the conversational state.

Stateful session bean

Stateful session beans are appropriate if any of the following conditions are true:

- The bean's state represents the interaction between the bean and a specific client.
- The bean needs to hold information about the client across method invocations.
- The bean mediates between the client and the other components of the application, presenting a simplified view to the client.

Entity Bean

- Can be used concurrently by several clients
  - Container manages database integrity by
    - Queuing client requests
    - Creating an instance of the bean for each client, and synchronizing
- Entity beans persist across multiple sessions and multiple users – long lived
  - No limit on the life time
  - Can be removed by explicit remove() or delete from database

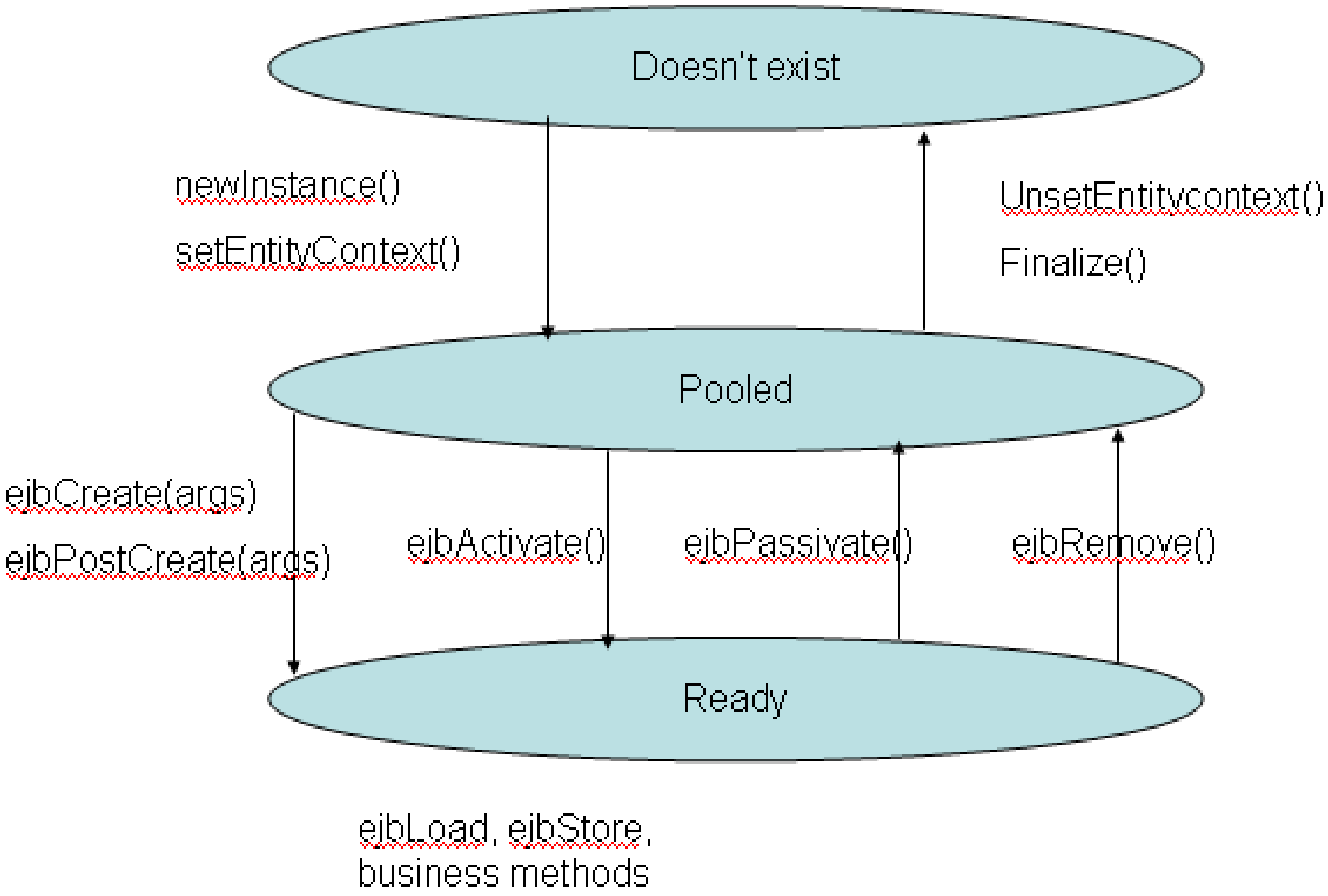
- Will survive a server crash or restart
  - No timeout period
- Directly represent data in a database,
  - Variables map directly to the columns in the table
  - Referenced by primary key
- Bean-Managed Persistence
  - Contains code that updates the database
- Container –Managed Persistence

- Primary Keys
  - Primary key is represented by a Java class

```
Public class Customerpk implements Serializable
{
    public int custid;
}
```

  - Carries the value from the client to server

Entity Bean Life cycle



Components of an Entity EJBs

- Component interface
- Home interface
- The bean class
  - Implements javax.ejb.EntityBean

- And a Primary Key class
  - Can be an existing class
    - String, Integer, Date, etc.
  - Can be a new class (e.g., for composite keys)
    - Must be serializable
    - Naming convention: *X*Key