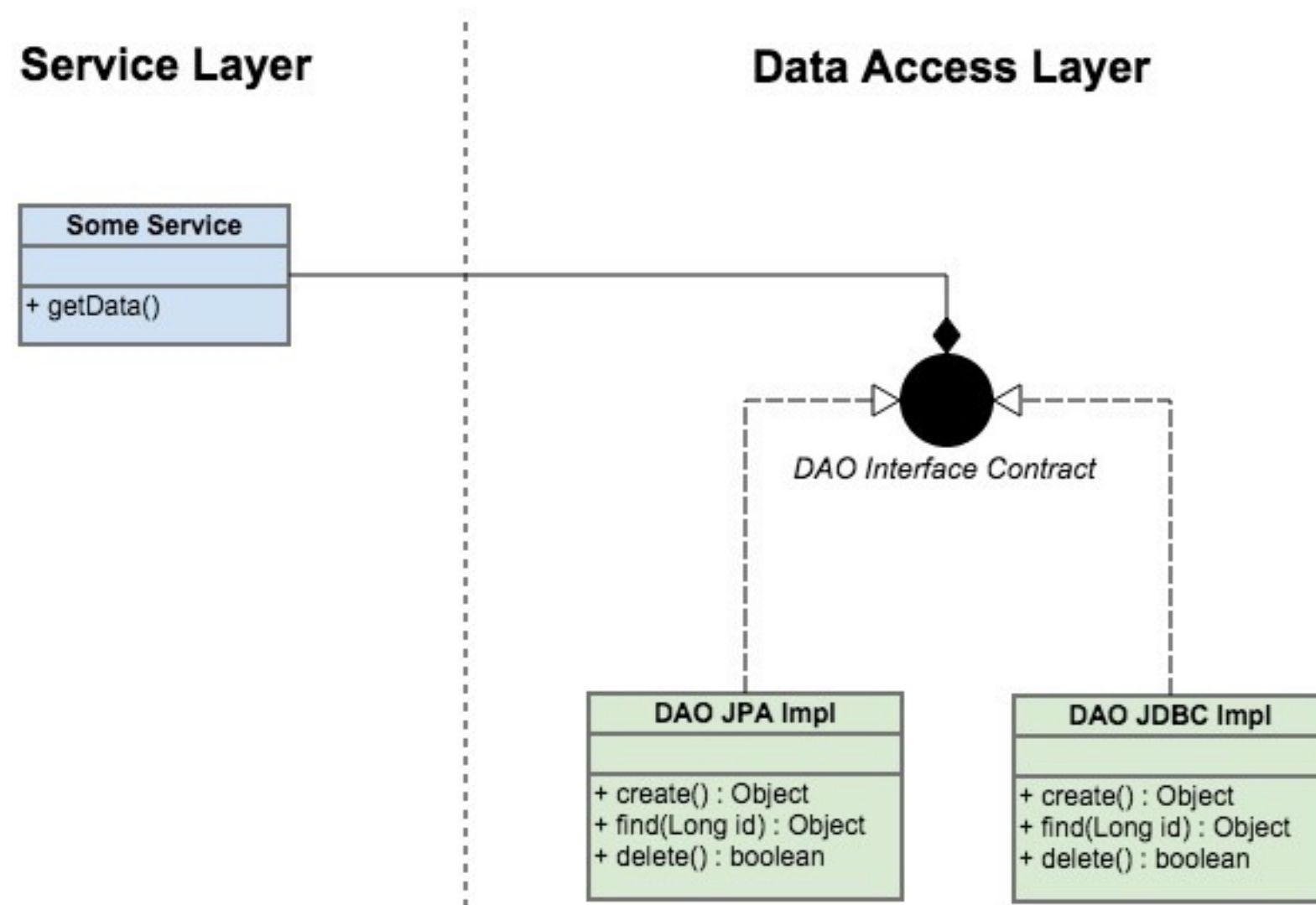


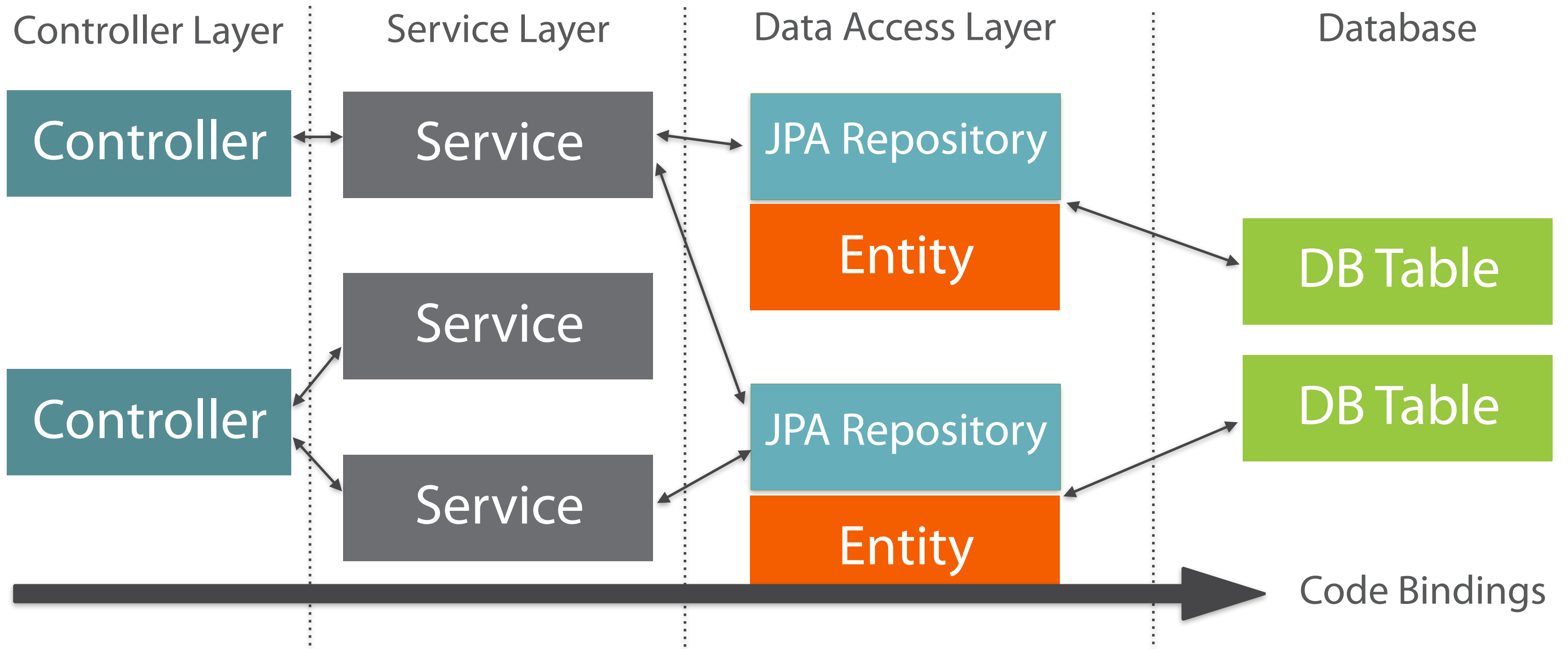
Spring Repositories

- Repositories are data access contracts that client code can depend and bind to



```
@Repository
public class MyRepository{
    public Object create(){...}
    private void jpaCreate(){...}
}
```

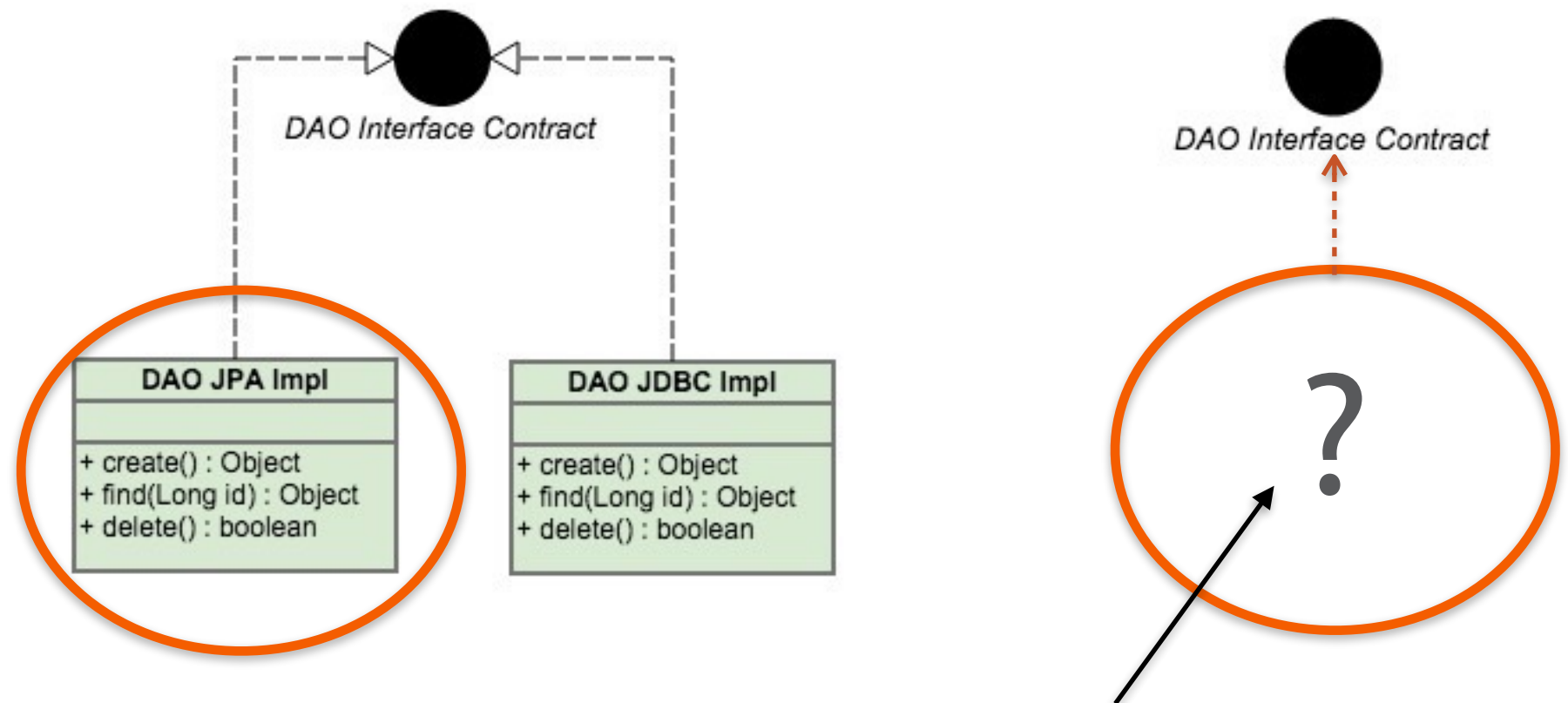
Repository Architectural Overview



Spring Data JPA Repositories

JPA Repository Differences

- Java interface, not class
- Map 1 to 1 with a JPA entity
- Focus on DAO contract



Thanks Spring Data JPA for implementing this for us!

```
public interface MyJpaRepository extends JpaRepository<Entity, Id Type>{  
}
```

JpaRepository Features

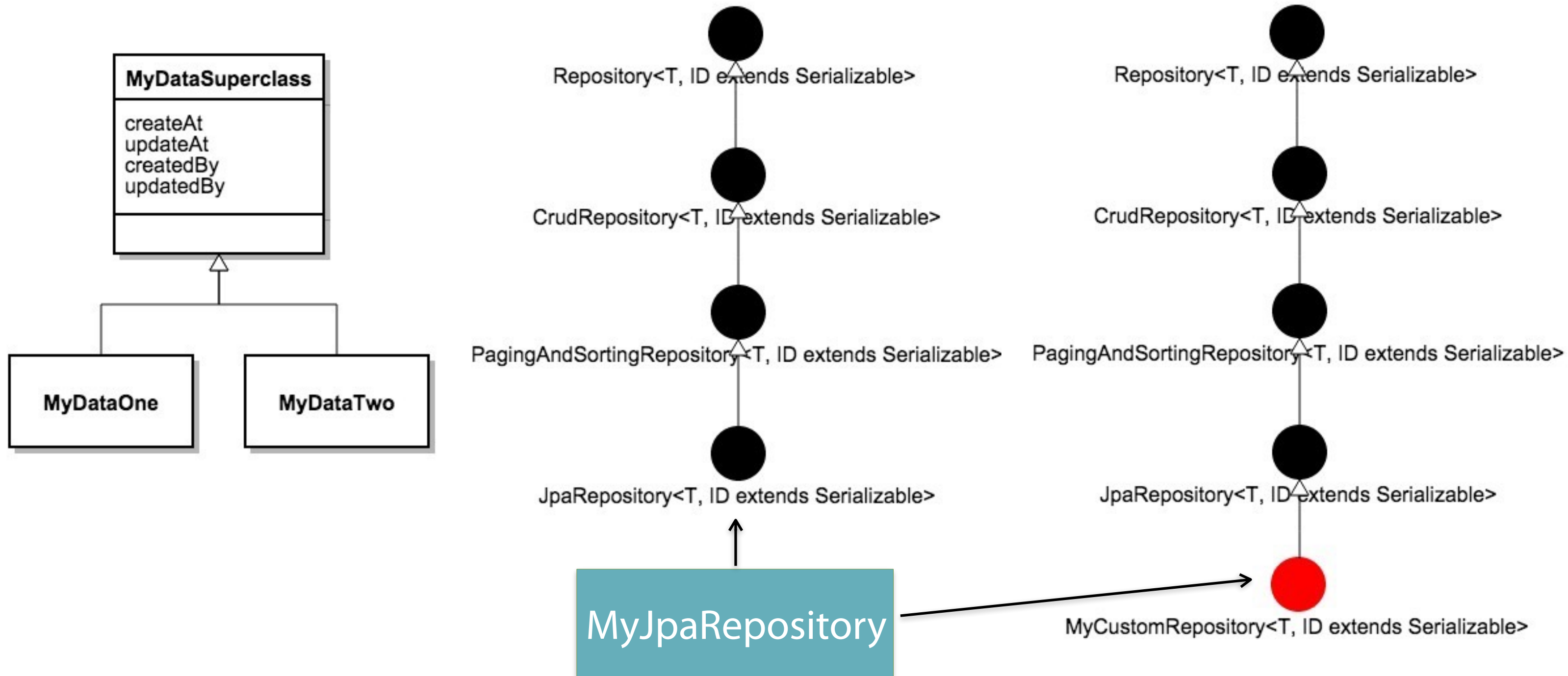
Functionality

- Query DSL
 - CRUD operations
 - Paging and sorting
 - Helpers
 - count()
 - exists(Long id)
 - flush()
 - deleteInBatch(Iterable entities)
-
- ```
graph TD; R["Repository<T, ID extends Serializable>"] --> CR["CrudRepository<T, ID extends Serializable>"]; CR --> PSR["PagingAndSortingRepository<T, ID extends Serializable>"]; PSR --> JPR["JpaRepository<T, ID extends Serializable>"];
```

} Spring Data

} Spring Data JPA

# Tip: Customize JpaRepository Hierarchy



# Tip: Refactoring Existing Repositories

Test suite

Create a new interface with a new name  
MyRepository -> MyJpaRepository

Switch out client  
code to new DAO  
contract

Move repository to  
a proxy, inject new  
JpaRepository

# Summary



Architectural concepts

JpaRepository features

Tips and tricks

DAO contract is important