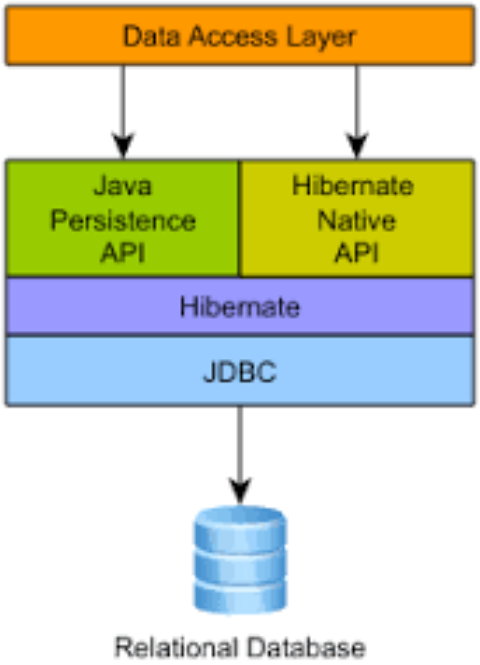
Java Frameworkleri

Hibernate

Hibernate, Java veri tiplerinin SQL veri tiplerine dönüşümünü gerçekleştirir. Veri sorgulama ve veri çekme işlemlerini de kullanıcı için sağlar.
Hibernate uygulamaların geliştirilmesinde büyük bir kolaylık sağlar.

Veritabanı tabloları ile tablolardaki bilgileri tutan sınıfları eşleştirir ve sınıflar üzerinden ilişkilendirerek **Create** - **Read** - **Update** - **Delete** işlemlerini uygulayabilmemizi sağlar. Hibernate üzerinden yapılan sorgulara HQL denir.

Veritabanına sorgu değil, sorgunun içinde olduğu session gider ve sessionın içinde birden fazla sorgu olabilir.



```
create table item (
    id bigint not null,
     price decimal(19,2) not null,
     primary key (id)
)
```

```
@Entity
public class Item {

    @Id
    @GeneratedValue
    private Long id;

    @Column(nullable = false)
    private BigDecimal price;
}
```

JSF (Java Server Faces)

JSF, arayüz sağlayıcıdır. Componentler sayesinde kullanıcı arayüzü oluşturabiliyoruz. Ancak componentler frontend teknolojileri gibi istemci tarafında çalışmadığından, büyük çaplı projelerde maliyet fazla olacaktır. Bu sebeple daha çok kurumsal ve zaman kısıtı olan projelerde kullanılır.

```
package com.mkyong.common;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.SessionScoped;
import java.io.Serializable;

@ManagedBean
@SessionScoped
public class HelloBean implements Serializable {

        private static final long serialVersionUID = 1L;

        private String name;

        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
}
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html">

<head>
        <title>JSF 2.0 Say Hello</title>
</head>
    <body>
        <p>Hey There! My hobby is #{hobbiesBean.hobby}</p>
    </body>
</html>
```

Spring

Spring platformunda yer alan Spring Boot, Spring Data, Spring MVC, Spring Batch, Spring Security gibi projelerin temelinde Spring framework yer alır.

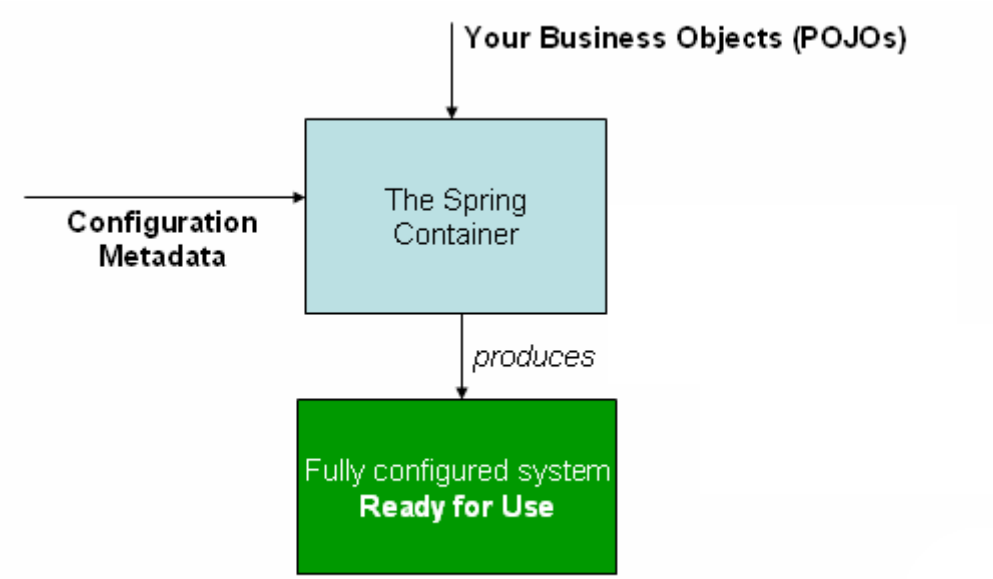
Dependency Injection, Spring'in en önemli parçasıdır ve Inversion of Control'ün bir uygulama metotudur.

Inversion of control ise bir tasarım prensibidir. *Ioc* ile Uygulama içerisindeki obje instance'larının yönetimi sağlanarak, bağımlılıklarını en aza indirgemek amaçlanır. Inversion of control, farklı implementasyonlar arasında geçişi kolaylaştırır, bağımlılıklar en aza indirilir ve modülerlik artar.

Spring bean, IoC Container tarafından yönetilen, uygulamanın yapıtaşlarını oluşturan şeylere Bean denir. Örneğin, Controller sınıfında, başka bir snıfta (Service, Repository,vs) oluşan bir Bean' i kullanmak istediğimizde, bunu Constructor Injection ile yapabiliriz.

```java
@Controller
public class SomeController {
    private final SomeService service;

    @Autowired
    public SomeController(SomeService service) {
        this.service = service;
    }
}
```



Kaynaklar

1) https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/beans.html
2) https://opendart.com/Default.aspx
3) https://neslihanesra.github.io/blog/jsf/2018/04/29/jsf-nedir/
4) https://gokhana.medium.com/
5) https://tugrulbayrak.medium.com/
6) https://www.baeldung.com/
7) https://www.javatpoint.com/
8) https://www.geeksforgeeks.org/