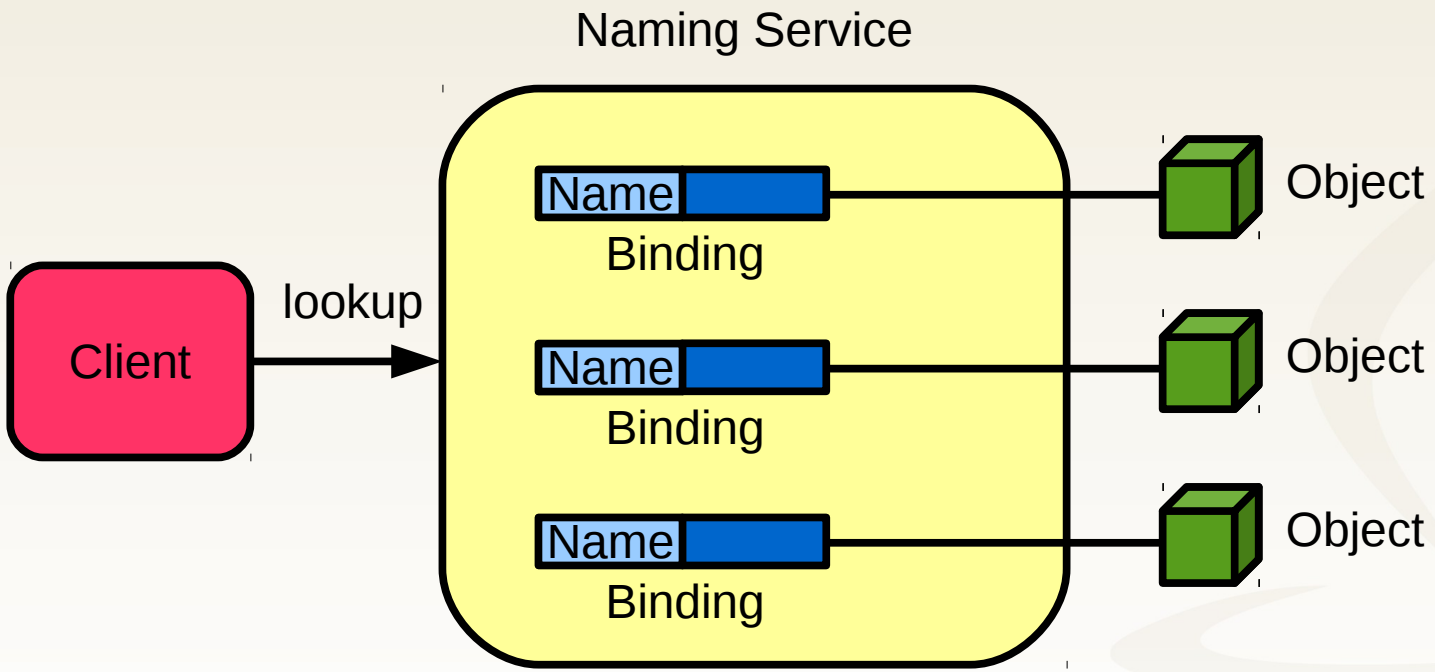
Java Naming And Directory Interface (JNDI)

Naming Service Nedir?

Nesneleri isimlerle ilişkilendirmek ve bu isimler vasıtası ile erişmek bilgisayar bilimindeki temel işlemlerden birisidir;



File sistemdeki dosyaya filename atanması, IP'ye domain name atanması naming sistemlere örnektir

Her naming service'in kendine göre bir isimlendirme syntax'ı vardır

Bazı naming sistemler nesnenin kendisini tutumak yerine referansını tutarlar

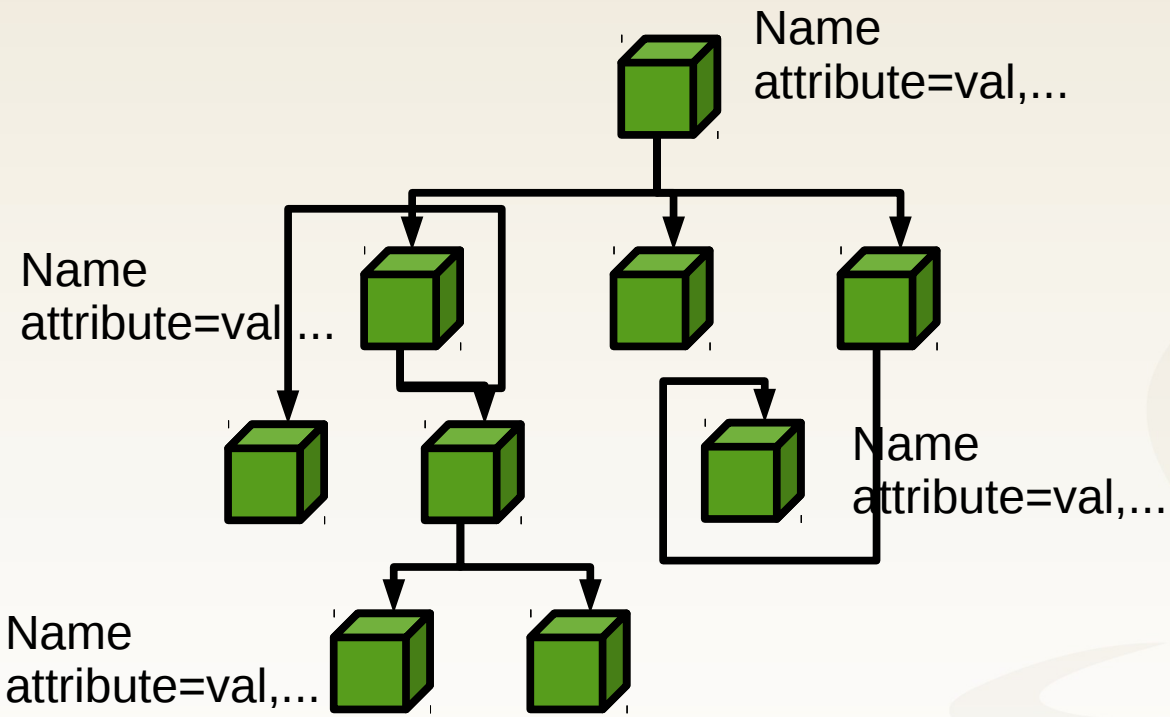
Context & Namespace Kavramları

- **Context**
  - Bir grup **name-object ikilisi** bir context oluşturur
  - Context'ler arasında da **parent-child ilişki**si olabilir
  - Buna subcontext adı verilir
- **Namespace**
  - Bir naming sistemde **tanımlı bütün isimler** namespace'i oluşturur

Directory Service Nedir?

- Naming service'in **extend** edilmesidir
- İsim – nesne eşleştirmesinin yanında nesnelere bir takım **attribute**'lar da atanabilir
- Her bir attribute bir **identifier**'a sahiptir, **bir veya daha fazla da değer** alabilir
- Bir grup directory object'in bir araya getirilmesinden **directory** (dizin) oluşur
- **Directory service** bir dizin nesnesinin attribute'ları ile ilgili **bir takım operasyonlar** yapmayı sağlar

Directory Service Hiyerarşisi



Dizinler çoğunlukla hiyerarşik biçimde oluşturulurlar

Directory nesneleri üzerinde isim ve attribute bilgilerini kullanarak arama yapmak mümkündür

Bu bilgileri içeren sorgular da tanımlanabilir, bunlara “search filter” adı verilir

Örnek Naming ve Directory Servisleri

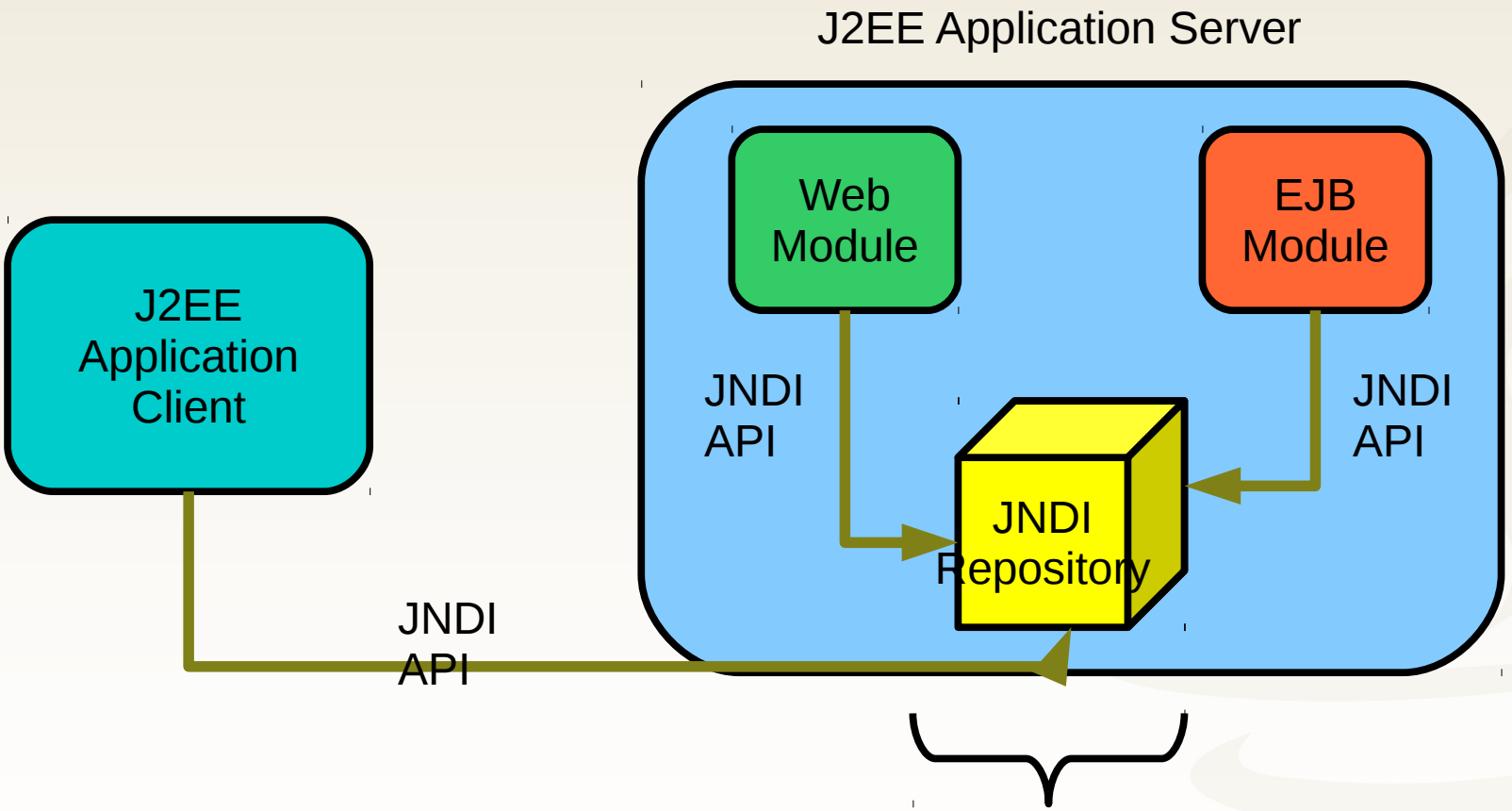
JAVA Eğitimleri

- **UNIX/Windows File System**: Dosya sistemindeki dosyalara ve dizinlere isimleri ile erişmeyi sağlar
- **DNS**: Internet üzerindeki sistemlerin IP'lerine domain isimleri ile erişmeyi sağlar
- **NIS ve NIS+**: Bir ağdaki bilgisayar sistemlerini bir araya getirir
- **LDAP**: Directory Access Protocol'ün basitleştirilmiş halidir, Bir organizasyon hiyerarşisini oluşturmayı ve yönetmeyi sağlar

JNDI API

- Naming ve directory servisleri için **ortak bir arayüz** sunar
- Bir arayüz olduğu için de **arka taraftaki naming ve directory servis implementasyonu** hakkında kullanmak için bilgi sahibi olmanız gerekir
- Örneğin JNDI üzerinden LDAP ile çalışırken **LDAP hakkında da bilgi sahibi** olmanız gerekir

JavaEE ve JNDI



Application Server değişik modüller tarafından paylaşılacak
Ortak data'yı saklamak amacı ile JNDI repository implemantasyonu sağlar

JavaEE ve JNDI

- Bu **repository** içerisinde JDBC DataSource, Session EJB, MDB nesneleri tutulabilir
- Farklı modüller bu nesnelere **JNDI API** vasıtası ile lookup yaparak erişebilirler
- JNDI ağacının oluşturulması, nesnelerin JNDI repository içerisinde konması genellikle **Application Server tarafından** halledilir
- Uygulama geliştiricilerin bu konularla ilgili **manuel işlem** yapmalarına **gerek yok**tur

java:comp/env Nedir?

- J2EE spesifikasyonu tarafından reserve edilmiş **JNDI Context** ismidir
- **Deployment descriptor** dosyalarında tanımlı resource veya ejb referansları **buraya bind** edilir
- Örneğin JMS destination, EJB Home nesnelerine referanslar buraya bind edilir
- **Salt okunur** bir context'dir
- Bind edilen **tanımlar değiştirilemez**

JavaEE JNDI Context Hiyerarşisi

- java:
  - comp
    - env
      - jdbc
      - jms
      - ejb
    - UserTransaction

Container tarafından yönetilen Servlet, EJB Gibi instance'ların environment context'i Rolündedir

Bu nesnelerin her birinin kendine ait bir env Context'i container tarafından oluşturulur ve yönetilir

Belirtilen sub context'lerde DataSource, EJB Home ve JMS Queue, Topic Nesnelerine referanslar bind edilir

Bu referansların bu subcontext'lerde Bind edilmesi zorunlu değildir, ancak Anlaşılırlığı artıran bir unsurdur

JNDI Bind ve Lookup Örneği

```
Hashtable<String, String> props = new Hashtable<String, String>();
props.put("java.naming.factory.initial",
"org.jnp.interfaces.NamingContextFactory");
props.put("java.naming.provider.url", "jnp://localhost:1099");

InitialContext context = new InitialContext(props);
```

Server'daki JNDI'a erişmek için gerekli parametreler Hashtable ile key-value pair şeklinde Verilir. Diğer yol bu propertyleri classpath altında jndi.properties isimli dosyaya key=value Şeklinde tanımlamaktır. Bu durumda InitialContext default constructor ile yaratılır

```
context.bind("test", "foo");
```

→ Belirtilen isimde nesneyi bind eder

```
String str = (String)context.lookup("test");
System.out.println(str);
```

↓ Belirtilen isimdeki nesneyi döner

```
context.close();
```

İletişim

www.harezmi.com.tr

www.java-egitimleri.com

info@harezmi.com.tr

info@java-egitimleri.com

@HarezmiBilisim

@JavaEgitimleri