

14.–17. 09. 2009
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Konkurrenzkampf

NHibernate vs. Microsoft Entity Framework

Thomas Haug
MATHEMA Software GmbH

Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

Eigenschaften NHibernate (1/3)

- Aktuell Version 2.1 GA
- Open Source
 - GNU Lesser General PUBLIC License
 - www.nhforge.org
- Basiert ursprünglich auf Hibernate 2.1 (Java)
- Einsatzgebiet: Applikationen mit OO Domänenmodell und Geschäftslogik in .Net Applikation und persistenz in relationaler Datenbank



NHIBERNATE

Eigenschaften NHibernate (2/3)

- Kapselt ADO.Net und SQL vor Entwickler
- Unterstützt eine Vielzahl von Datenbanken (SQL Server, Oracle, DB2, Sybase, MySql, Postgres, ...)
- Transparente Persistenz für POCO Klassen (Entitäten)
- Unterstützung von Vererbungshierarchien und polymorphen Abfragen
- Transitive Persistenz für Objektgraphen
- Unterstützung für sog. Components

Eigenschaften NHibernate (3/3)

- Unterstützung für ‚Lazy Loading‘ und ‚Eager Fetching‘
- Transaktionales ‚Write-Behind‘
- Unterstützung von dynamischen Updates und Inserts
- Optimistische und Pessimistische Sperr-Mechanismen
- Unterstützung von abgekoppelten Objekten
- Second level Caching
- (SQL) Tracing Logging integraler Bestandteil

Eigenschaften EF (1/3)

- Offizielle Version Entity Framework 1.0
 - Bestandteil .Net 3.5 SP1
- EF 2.0 Beta 1
 - EF 2.0 erfordert .Net 4.0 (Beta)
- Community Technology Preview 1
 - Self-tracking Entities
 - POCO
 - Code Only

Eigenschaften EF (2/3)

- Kapselt ADO.Net und SQL vor Entwickler
- Unterstützt SQL Server „out-of-the-box“
- (Transparente) Persistenz für (POCO) Klassen
- Unterstützung von Vererbungshierarchien und polymorphen Abfragen
- Transitive Persistenz für Objektgraphen
- Unterstützung für ‚Lazy Loading‘ vs ‚Eager Fetching‘

Eigenschaften EF (3/3)

- Transaktionales ‚Write-Behind‘
- Optimistische Locking
- Unterstützung von abgekoppelten Objekten
- SQL Tracing ist Bestandteil, aber nur für Abfragen
 - Abhilfe schafft EFProviderWrapper
 - Caching provider (Velocity)
 - EFTracing Wrapper
 - Microsoft Public License (Ms-PL)

Vergleich Eigenschaften



- Open Source
- Version 2.1 GA
- .Net 2.0
- Unterstützte verschiedene Datenbanken
- Optimistisches und Pessimistisches Locking
- SQL Tracing leicht möglich

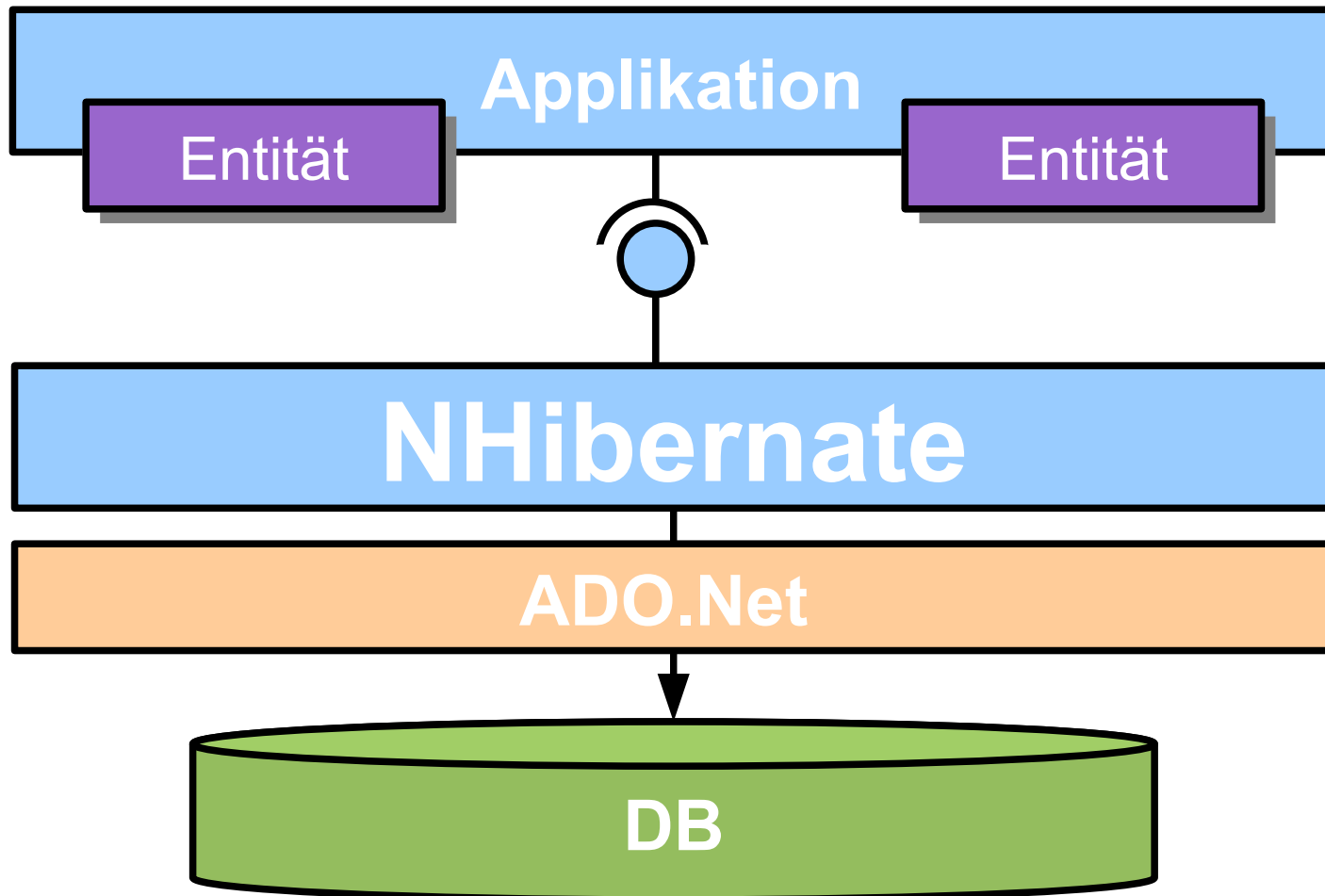
Entity Framework

- Closed Source
- Version 1.0 (3.5 SP1)
- Version 2.0 auf .Net 4.0
- Nur SQL Server, andere über Drittanbieter
- Optimistisches Locking
- SQL Tracing nur eingeschränkt

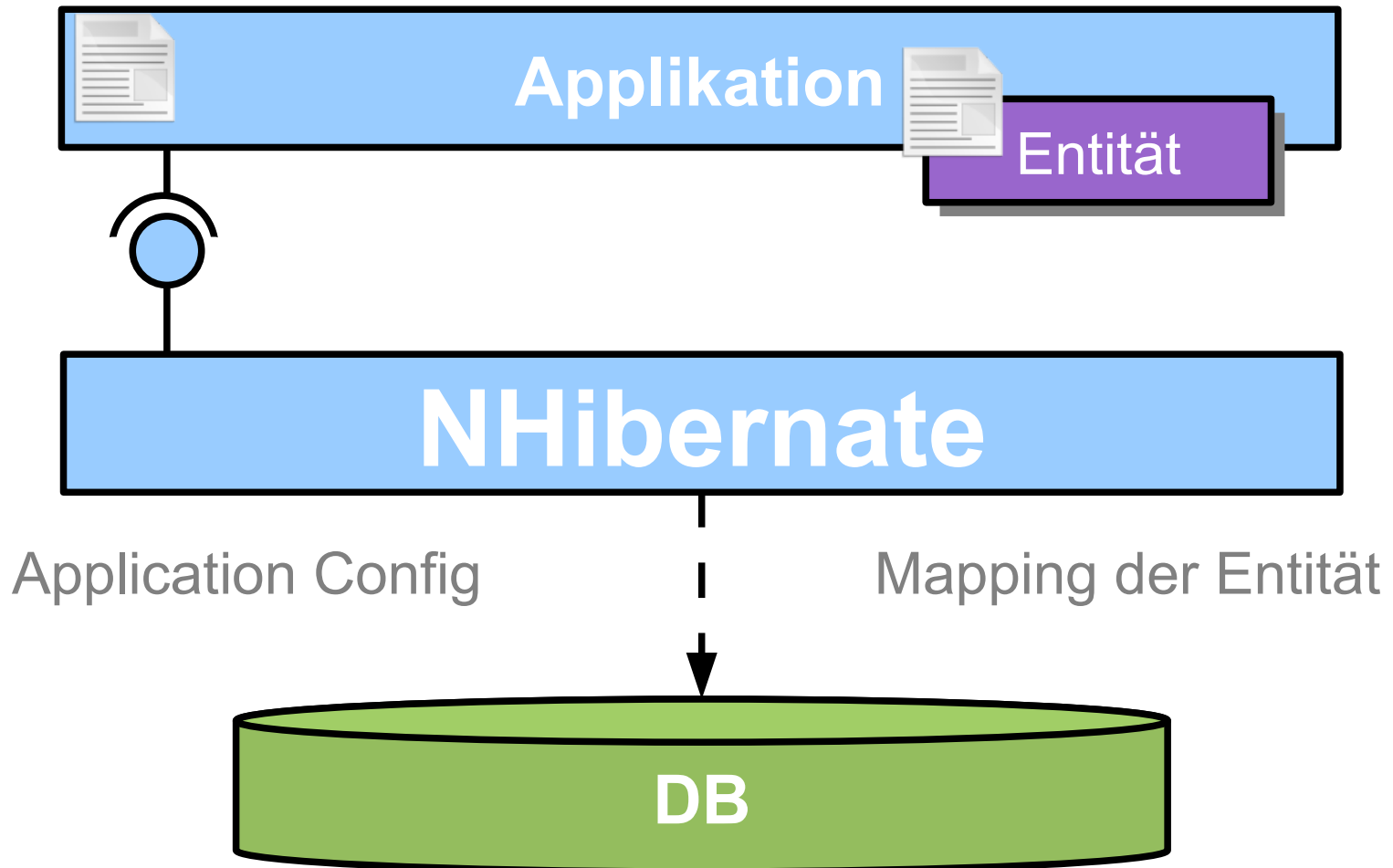
Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

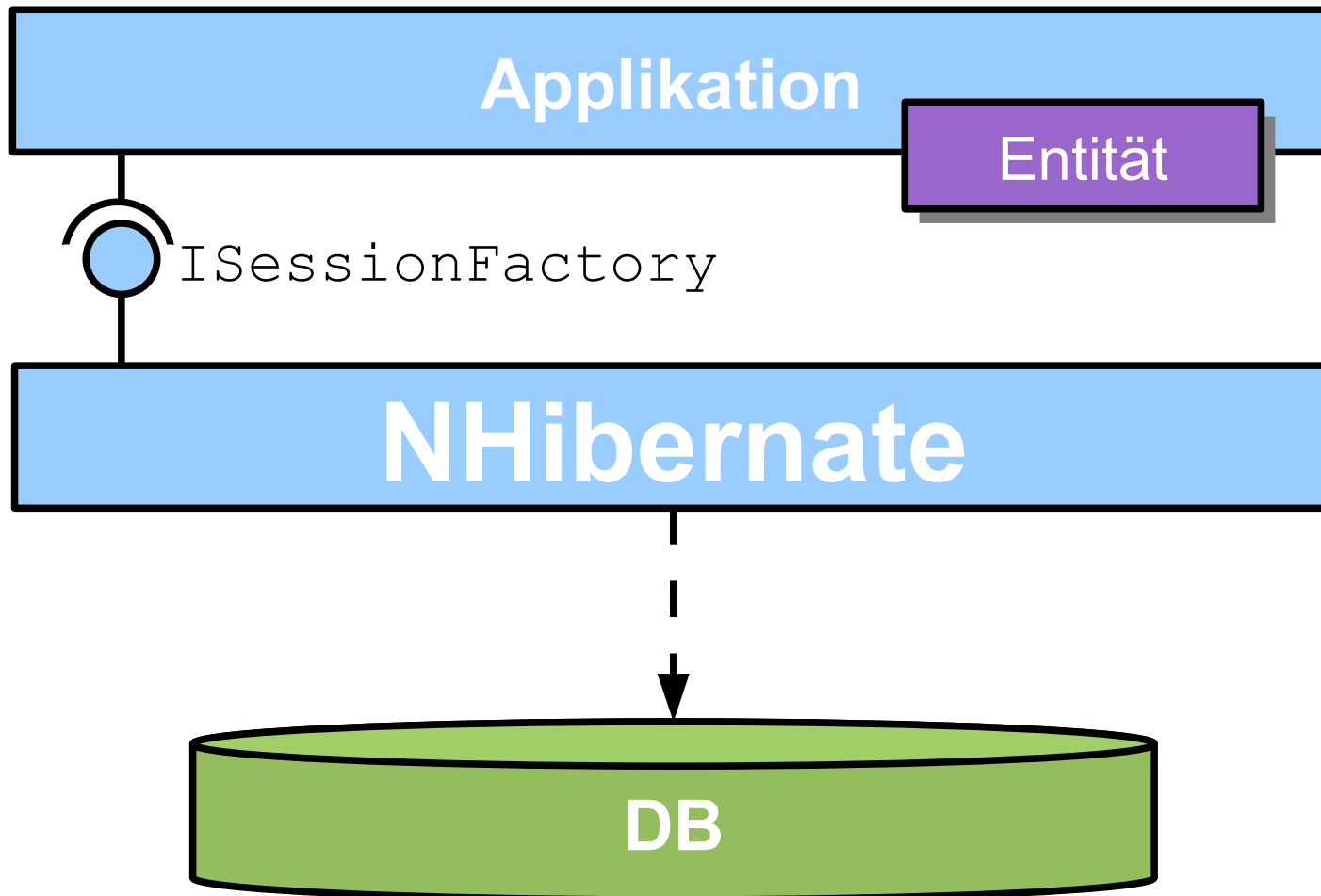
NHibernate Architektur (1/2)



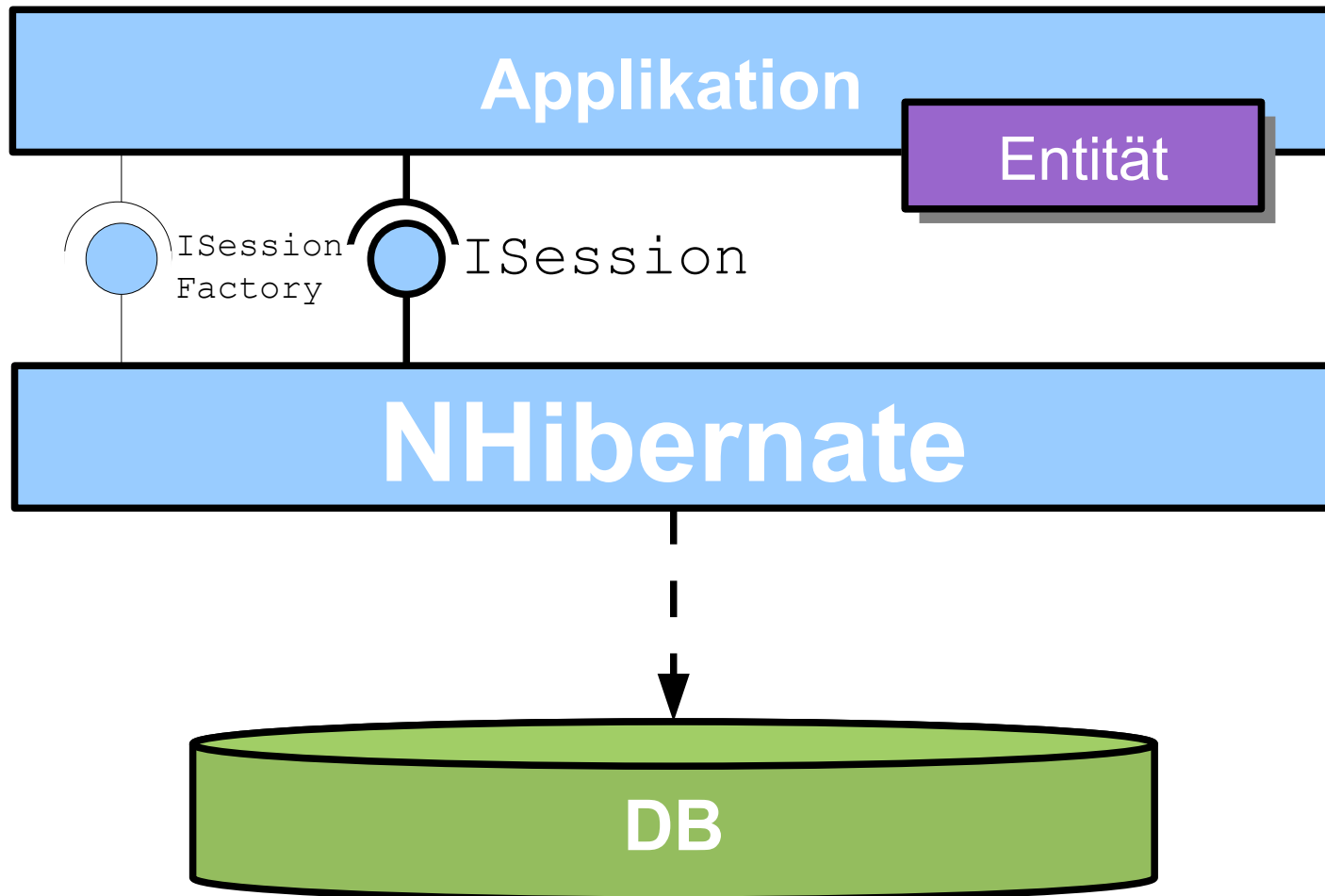
NHibernate Architektur (2/2)



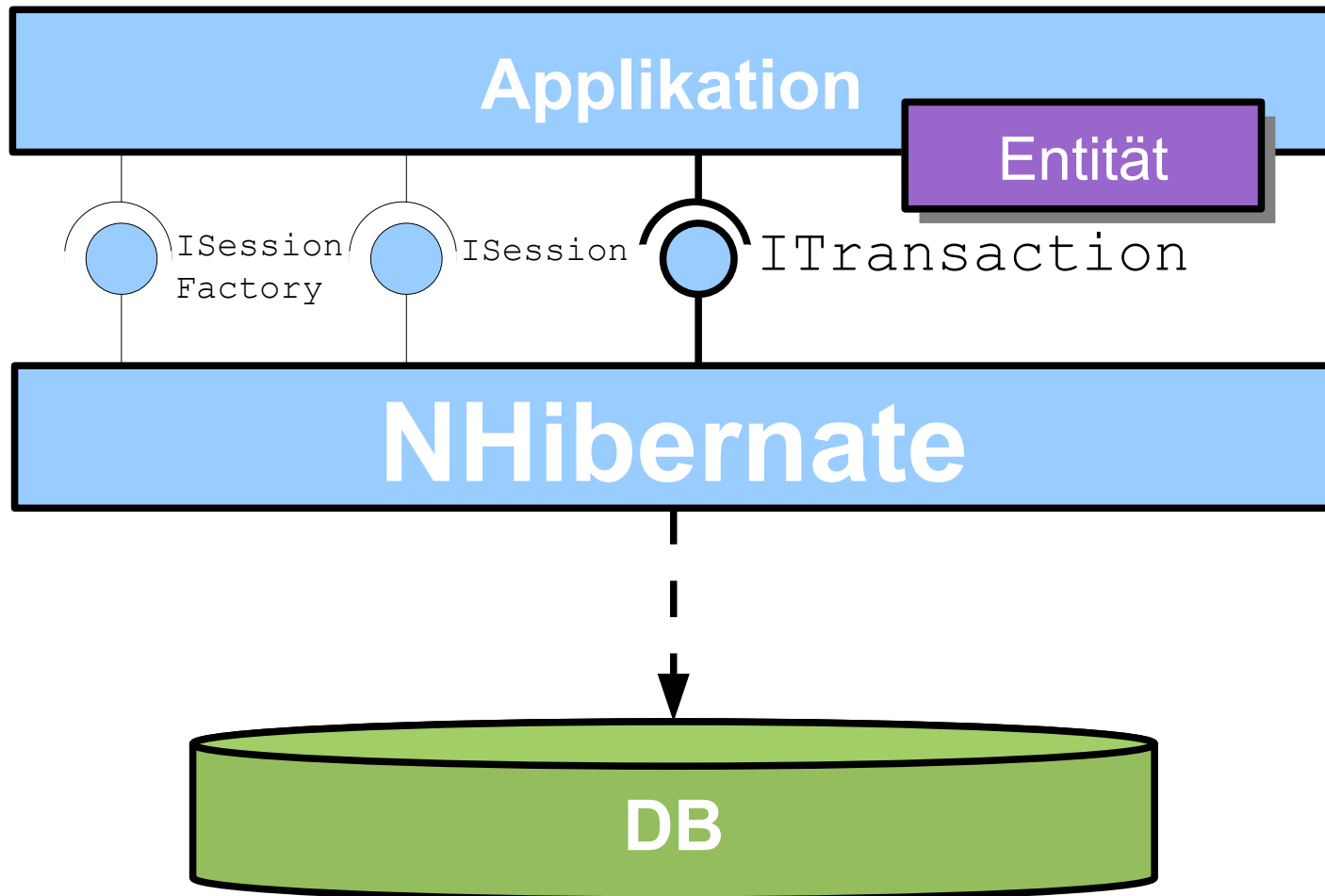
NHibernate Architektur (2/2)



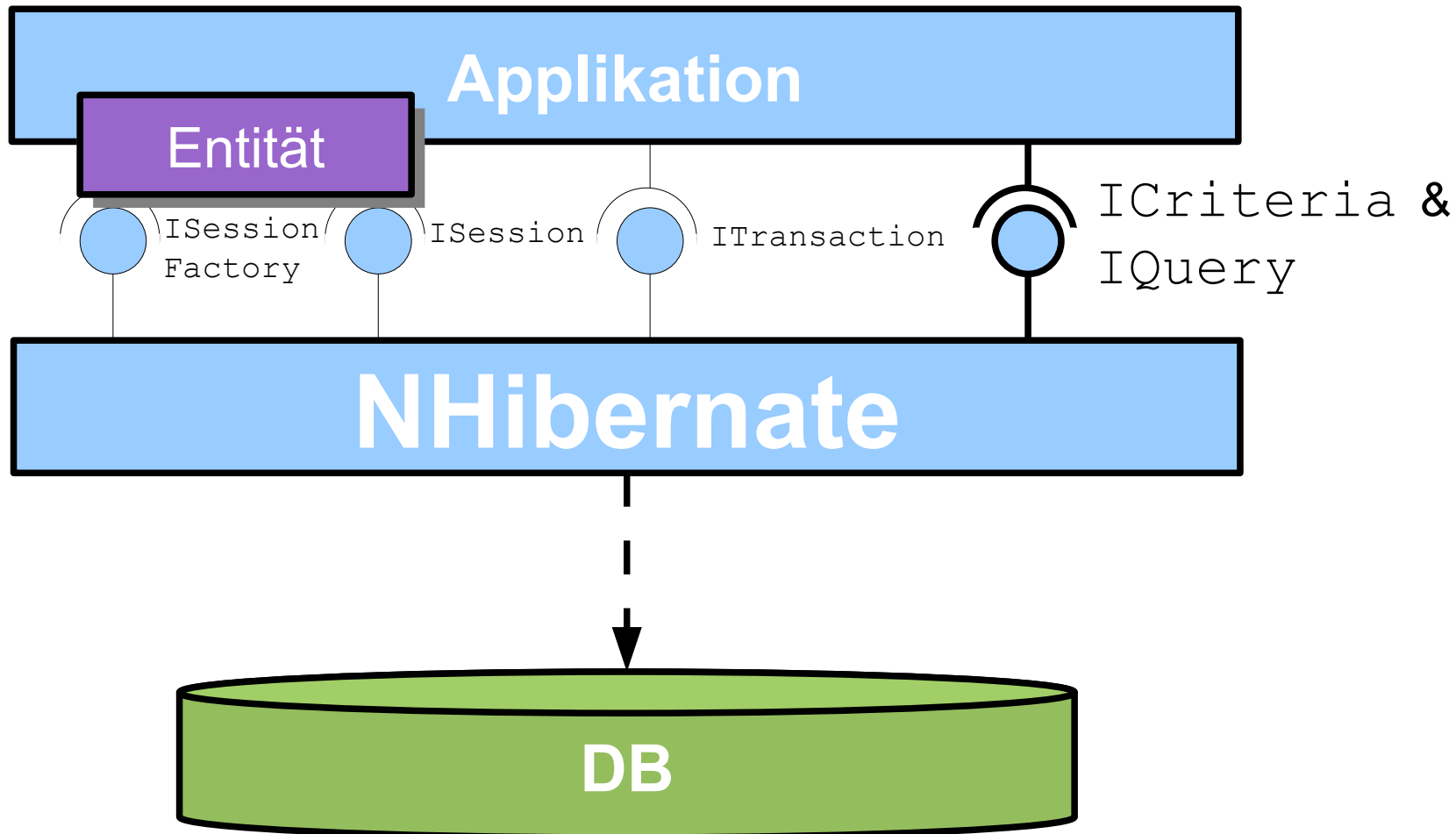
NHibernate Architektur (2/2)



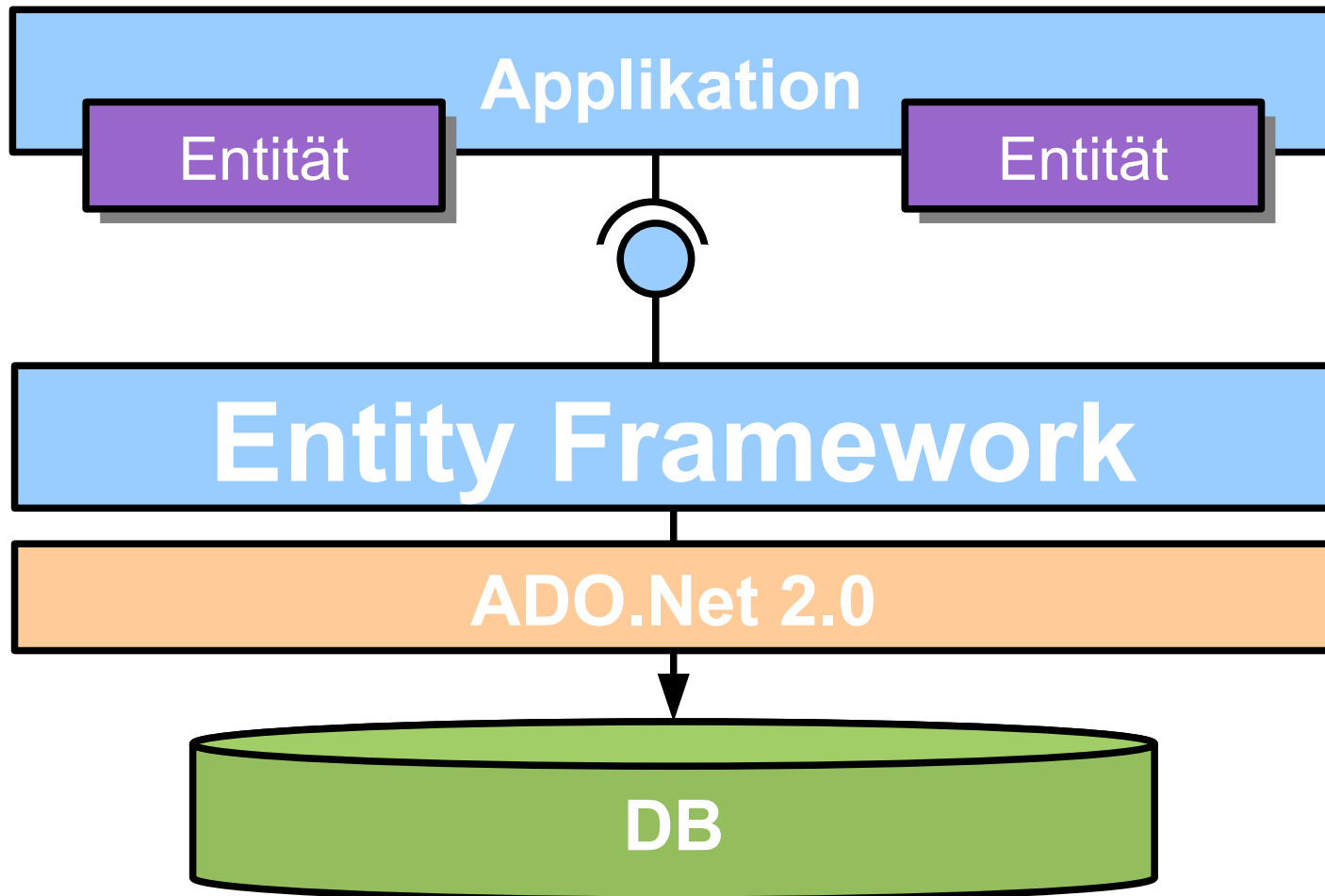
NHibernate Architektur (2/2)



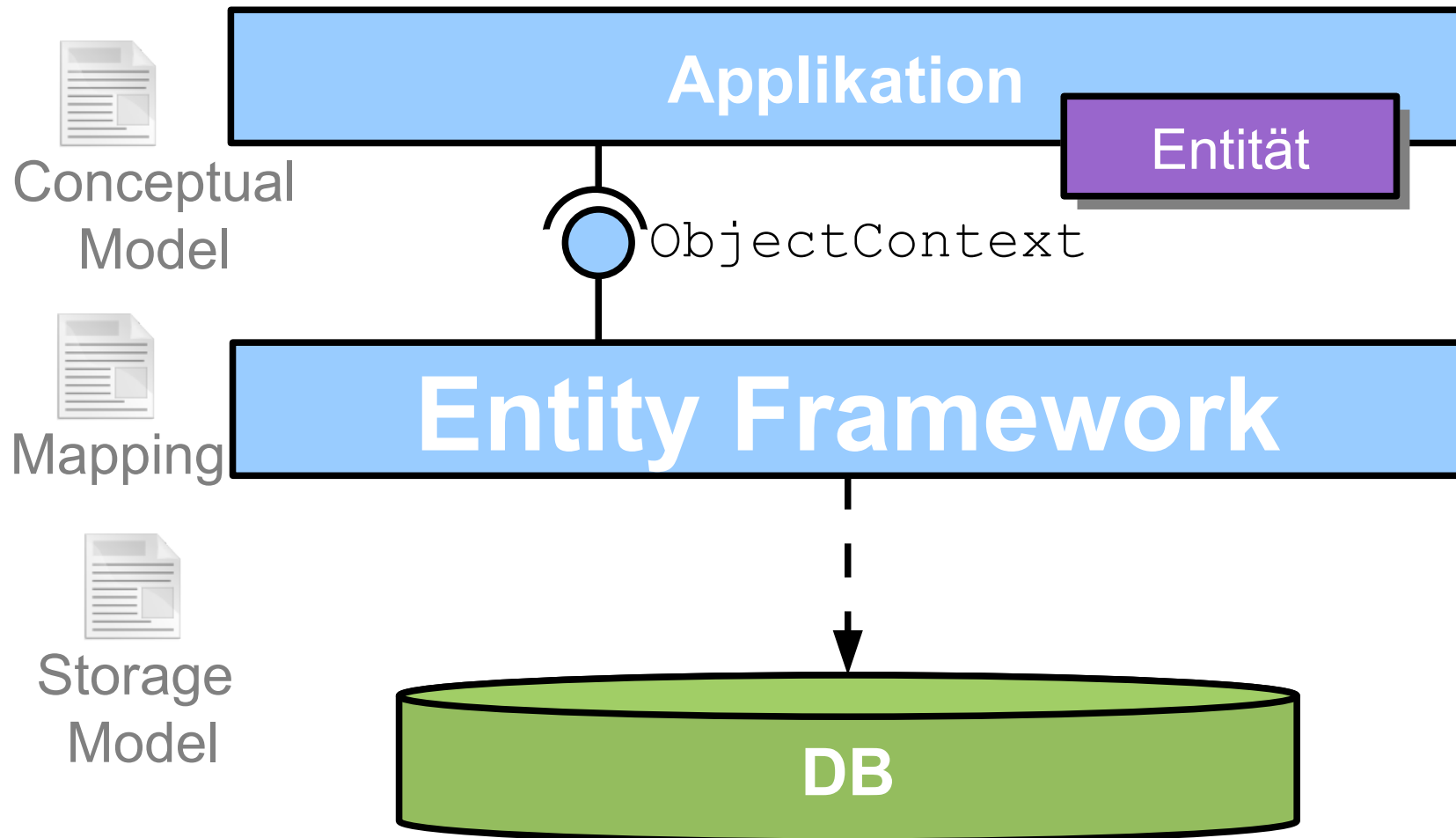
NHibernate Architektur (2/2)



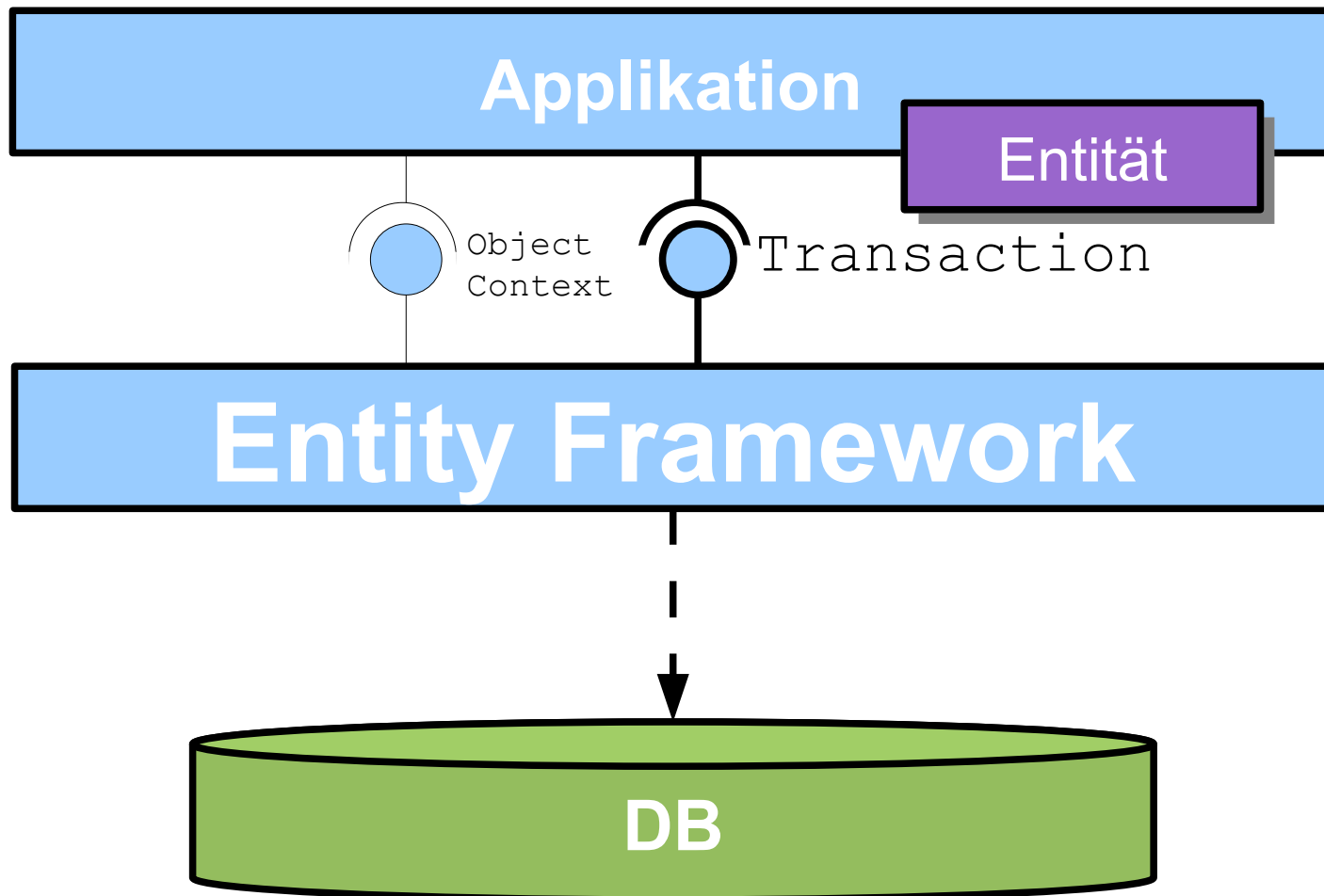
Entity Framework Architektur (1/2)



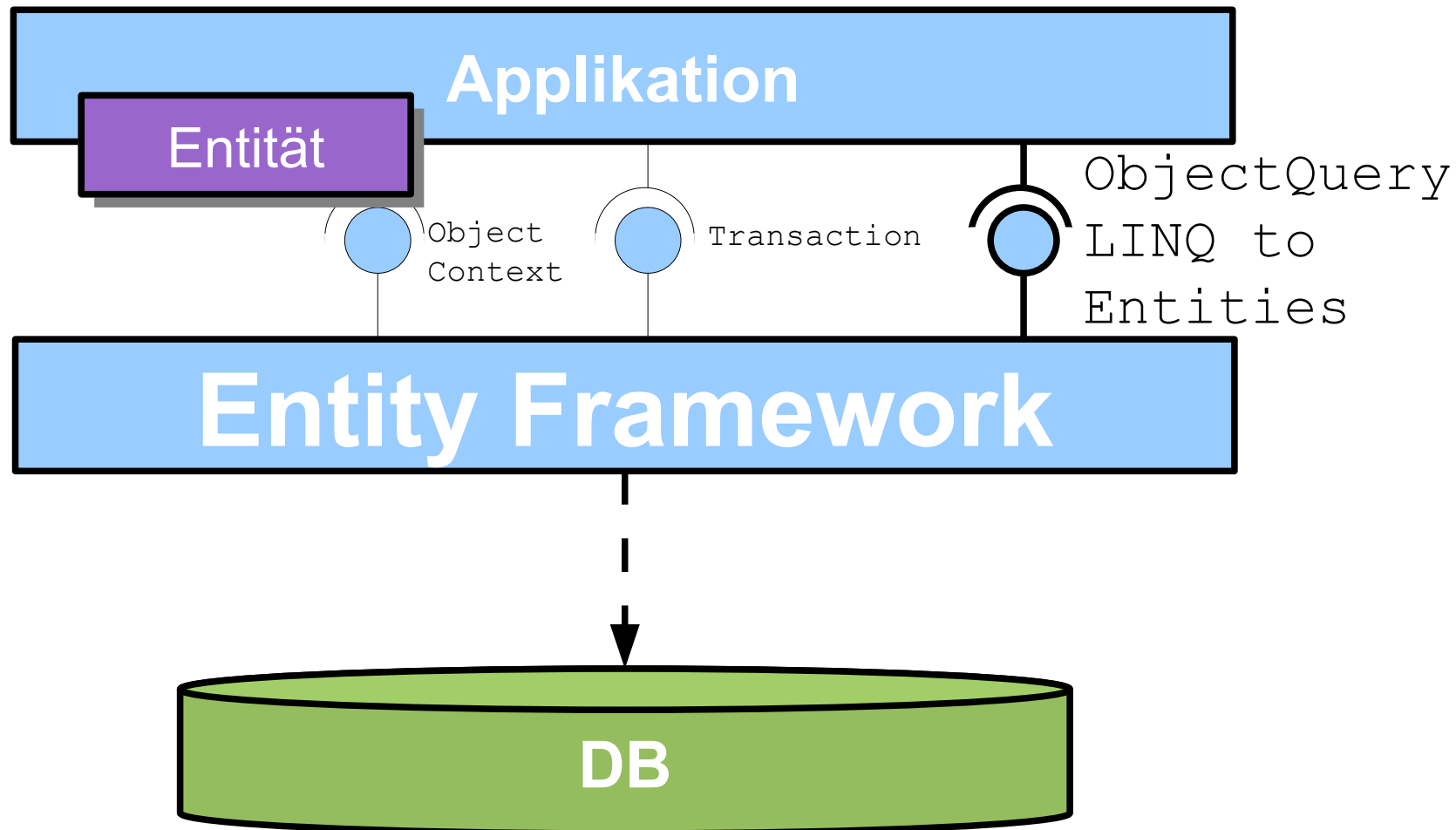
Entity Framework Architektur (2/2)



Entity Framework Architektur (2/2)



Entity Framework Architektur (2/2)



Essentielle Klassen und Schnittstellen



Entity Framework

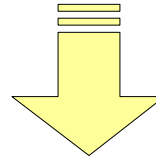
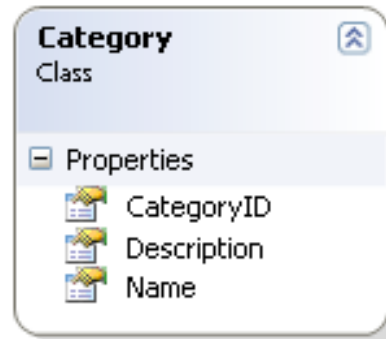
- **Configuration**
- **ISessionFactory**
- **ISession**
- **ITransaction**
- **Abfrage**
 - **ICriteria**
 - **IQuery**
 - **[LINQ to NHibernate]**


- **EDM**
- **}ObjectContext**
- **Transaction**
- **Abfrage**
 - **ObjectQuery**
 - **LINQ to Entities**

Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

Beispiel für weitere Betrachtung



Simple_Categories				
	Column Name	Data Type	Nullable	Identity
	CategoryID	int	No	<input checked="" type="checkbox"/>
	name	nvarchar(50)	No	<input type="checkbox"/>
	description	nvarchar(200)	Yes	<input type="checkbox"/>
				<input type="checkbox"/>

Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

Entitäten beschreiben



Entity Framework

```

<hibernate-mapping
  xmlns="urn:nhibernate-mapping-2.2">
  <class name="Entities.Category,
    NHBeispielMitXML"
    table="Simple_Categories" >
    <id name="CategoryID"
      column="CategoryID"
      type="integer">
      <generator class="native" />
    </id>
    <property name="Name"
      column="name"
      type="String"
      length="50" />
    <!-- weitere Properties -->
  </class>
</hibernate-mapping>
  
```

```

<edmx:Edmx Version="2.0" ...>
  <edmx:ConceptualModels>
    <Schema Namespace="EFvsNHModel" Alias="Self"
      xmlns:store="http://schemas.microsoft.com/ado/2007/12/edm/EntityStoreSchemaGenerator"
      xmlns="http://schemas.microsoft.com/ado/2008/09/edm">
      <EntityContainer Name="EFvsNHEntities">
        <EntitySet Name="Simple_Categories" EntityType="EFvsNHModel.Category" />
      </EntityContainer>
      <EntityType Name="Category">
        <Key>
          <PropertyRef Name="CategoryID" />
        </Key>
        <Property Name="CategoryID" Type="Int32" Nullable="false"
          store:StoreGeneratedPattern="Identity" />
        <Property Name="Name" Type="String" Nullable="false" MaxLength="50" Unicode="true"
          FixedLength="false" />
        <Property Name="Description" Type="String" MaxLength="200" Unicode="true"
          FixedLength="false" />
      </EntityType>
    </Schema>
  </edmx:ConceptualModels>
  
```

Entitäten beschreiben



Entity Framework

```

<hibernate-mapping
  xmlns="urn:nhibernate-mapping-2.2">
  <class name="Entities.Category,
    NHBeispielMitXML"
    table="Simple_Categories" >
    <id name="CategoryID"
      column="CategoryID"
      type="integer">
      <generator class="native" />
    </id>
    <property name="Name"
      column="NAME"
      type="String"
      length="50" />
    <!-- weitere Properties -->
  </class>
</hibernate-mapping>
  
```

```

<edmx:Edmx Version="2.0" ...>
  <edmx:Runtime>
    <edmx:StorageModels>
      <Schema Namespace="EFvsNHModel.Store" Alias="Self"
        Provider="System.Data.SqlClient" ProviderManifestToken="2008"
        >
        <EntityContainer Name="EFvsNHModelStoreContainer">
          <EntitySet Name="Simple_Categories"
            EntityType="EFvsNHModel.Store.Simple_Categories" store:Type="Tables" Schema="dbo" />
        </EntityContainer>
        <EntityType Name="Simple_Categories">
          <Key>
            <PropertyRef Name="CategoryID" />
          </Key>
          <Property Name="CategoryID" Type="int" Nullable="false"
            StoreGeneratedPattern="Identity" />
          <Property Name="name" Type="nvarchar" Nullable="false" MaxLength="50" />
          <Property Name="description" Type="nvarchar" MaxLength="200" />
        </EntityType>
      </Schema>
    </edmx:StorageModels>
  </edmx:Runtime>
</edmx:Edmx>
  
```

Entitäten beschreiben



Entity Framework

```
<hibernate-mapping
  xmlns="urn:nhibernate-mapping-2.2">
  <class name="Entities.Category,
    NHBeispielMitXML"
    table="Simple_Categories" >
    <id name="CategoryID"
      column="CategoryID"
      type="integer">
      <generator class="native" />
    </id>
    <property name="Name"
      column="NAME"
      type="String"
      length="50" />
    <!-- weitere Properties -->
  </class>
</hibernate-mapping>
```

```
<hibernate-mapping
  xmlns="urn:nhibernate-mapping-2.2">
  <edmx:Mappings>
    <Mapping Space="C-S"
      xmlns="http://schemas.microsoft.com/ado/2008/09/mapping/cs">
      <EntityContainerMapping
        StorageEntityContainer="EFvsNHModelStoreContainer"
        CdmEntityContainer="EFvsNHEntities">
        <EntitySetMapping Name="Categories"><EntityTypeMapping
          TypeName="EFvsNHModel.Category"><MappingFragment
            StoreEntitySet="Simple_Categories">
          </MappingFragment></EntityTypeMapping></EntitySetMapping>
          <ScalarProperty Name="CategoryID" ColumnName="CategoryID"
            />
          <ScalarProperty Name="Name" ColumnName="name" />
          <ScalarProperty Name="Description"
            ColumnName="description" />
        </MappingFragment></EntityTypeMapping></EntitySetMapping>
      </EntityContainerMapping>
    </Mapping>
  </edmx:Mappings>
```

Entitäten beschreiben (No XML)



NHIBERNATE

```
[Class(Table="CATEGORY")]
public class Category {
    [Id(Name = "CategoryId",
        Column="ID",
        Access = "nosetter.camelcase",
        UnsavedValue="-1")]
    [Generator(1,Class="native")]
    public virtual int Id {
        get { return id; }
    }

    [Property(Column = "NAME",
        Length=50,
        NotNull=true,
        Index="NAME_IDX")]
    public virtual string Name
        { get;set; }
```

Entity Framework

```
public class Category {
    public virtual int
        CategoryId {get; set;}
    public virtual string
        Name { get;set; }

    //Programm: (EF2.0 Beta1 mit CTP1)
    ContextBuilder<CategoryCtx> builder =
        new ContextBuilder<CategoryCtx>();

    builder.RegisterKey(
        (Category c) =>c.CategoryID);

    // Zukunftsmusik

    var catConfig =
        new EntityConfiguration<Category>();
    catConfig.ForProperty(c => c.ID)
        .Identity();

    builder.Configure(catConfig);
```

Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

CRUD

- **(C)reate**

Erzeugen und Persistieren einer Category Instanz

- **(R)ead**

Lesen eines Category Objekts aus der Datenbank

- **(U)pdate**

Modifizieren des Category Objekts in der Datenbank

- **(D)elele**

Löschen der Category Instanz aus der Datenbank

CRUD



```

SessionFactory sF =
    CreateSessionFactory();

using (ISession session =
    sF.OpenSession()) {
    using (ITransaction tx =
        session.BeginTransaction()) {
        Category myCat = new
            Category("Motorteile");
        myCat.Description = "Alles";
        session.Save(myCat);

        [session.Flush();]
        tx.Commit();
    }
}

```

Entity Framework

```

using (EFvsNHEntities ctx =
    new EfvsNHEntities()) {

    Category myCat = new
        Category();
    myCat.Name = "Motorteile";

    [ctx.AddToCategories(myCat);]

    ctx.Categories.
        AddObject(myCat);

    ctx.SaveChanges();
}

```


cRUD



NHIBERNATE

```

using (ISession session =
    sF.OpenSession()) {
    using (ITransaction tx =
        session.BeginTransaction()) {

        Category gefCat =

        //erste Alternative
        session.Load<Category>(myCat.
            CategoryID);

        //zweite Alternative
        session.Get<Category>(myCat.
            CategoryID);

    }
}
  
```

Entity Framework

```

using (EFvsNHEntities ctx =
    new EfvsNHEntities()) {

    EntityKey key =

    new EntityKey("EFvsNH.Categories"
        , "CategoryID"
        , myCat.CategoryID);

    Category gefCat = (Category)
        ctx.GetObjectByKey(key);

    bool gefunden =
        context.TryGetObjectByKey(key,
            out gefundeCategory);

}
  
```



CRUD



Entity Framework

```
using (ISession s =
    sF.OpenSession()) {
    using (ITransaction tx =
        s.BeginTransaction()) {
        // bereits existierende, aber
        // nicht 'attached' Entität

        myCat.Name = "neuer Name";
        // erste Alternative
        session.Update(myCat);

        // zweite Alternative
        session.SaveOrUpdate(myCat);
    }
}
```

```
using (EFvsNHEntities ctx =
    new EfvsNHEntities()) {

    // bereits existierende, aber
    // nicht 'attached' Entität

    ctx.Attach(myCat);

    myCat.Name = "neuer Name";

    ctx.SaveChanges();
}
```

CRUD



```
using (ISession s =
    sF.OpenSession()) {
    using (ITransaction tx =
        s.BeginTransaction()) {

        //dritte Alternative
        Category myCat =
session.Load<Category>(myCat.
                        CategoryID);

        myCat.Name = "neuer Name";
    }
}
```

Entity Framework

```
using (EFvsNHEntities ctx =
    new EfvsNHEntities()) {

    // bereits existierende, aber
    // nicht 'attached' Entität

ctx.Attach(myCat);

myCat.Name = "neuer Name";

ctx.SaveChanges();
}
```

CRUD



```
using (ISession session =
    sF.OpenSession()) {
    using (ITransaction tx =
        session.BeginTransaction()) {

        session.Delete(myCat) ;

    }
}
```

Entity Framework

```
using (EFvsNHEntities ctx =
    new EfvsnHEntities()) {

    //object in Context laden ...

    ctx.DeleteObject(myCat) ;

    ctx.SaveChanges () ;

}
```

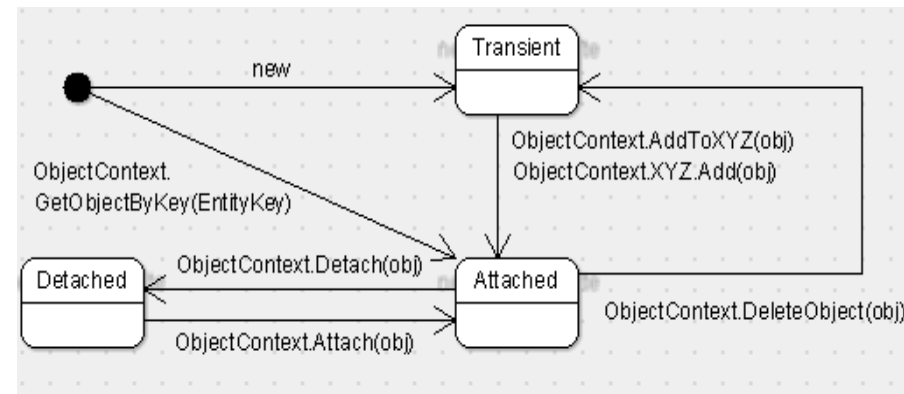
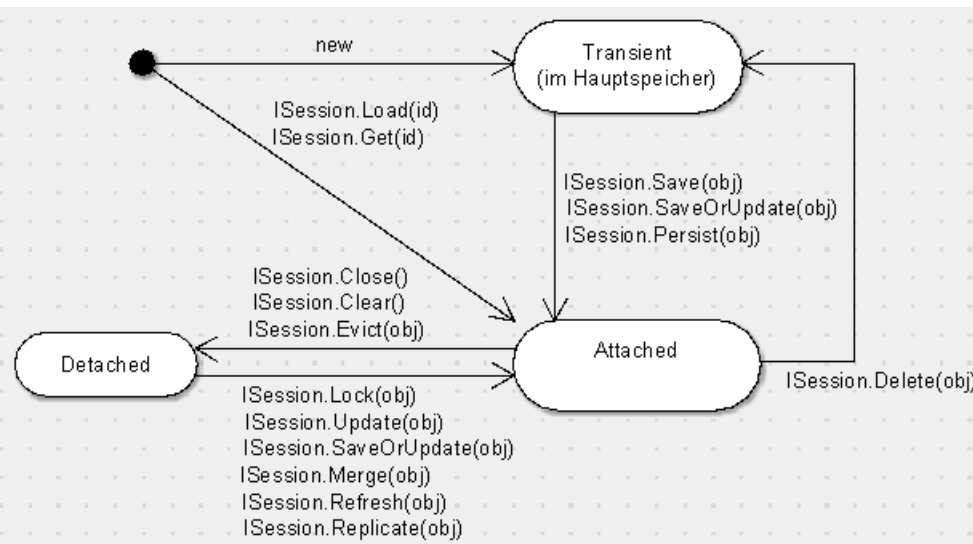
Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

Lifecycle



Entity Framework



Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
 - Eager fetching / lazy Loading
- Fazit

Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

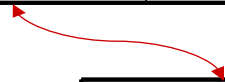
Beziehungskiste

- Assoziations-Multiplizitäten

- 1 : 1 Beziehung

ADDRESSID_	TYPE_	STREET_	CITY_	STATE_	ZIP_
10101	A	Y-STR	BIYEM	CAM	457

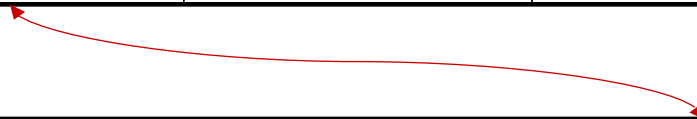
CUSTOMERID_	VERSION_	NAME_
10101	0	Ernie



- 1 : N Beziehung

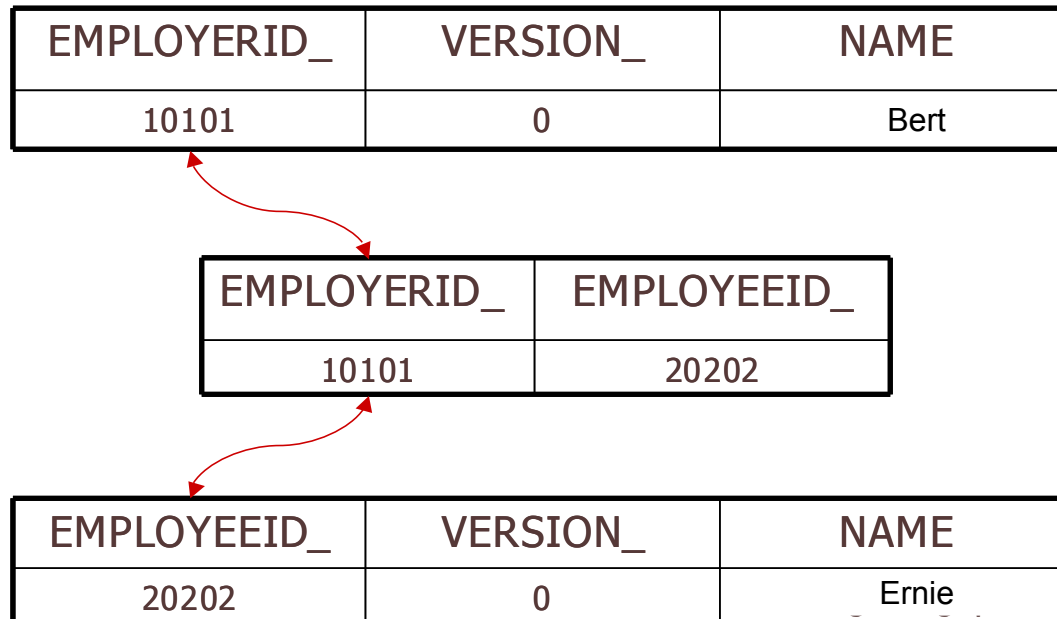
CUSTOMERID_	VERSION_	NAME_
10101	0	Ernie

ACCOUNTID_	TYPE_	CUSTOMERID_
20202	B	10101
20203	S	10101



Beziehungskiste

- Assoziations-Multiplizitäten
 - M : N Beziehung



Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- **Beziehungskiste**
 - Multipizitäten
 - **Persistence-by-Reachability**
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

Beziehungskiste

- Persistence-by-Reachability

- NHibernate unterstützt

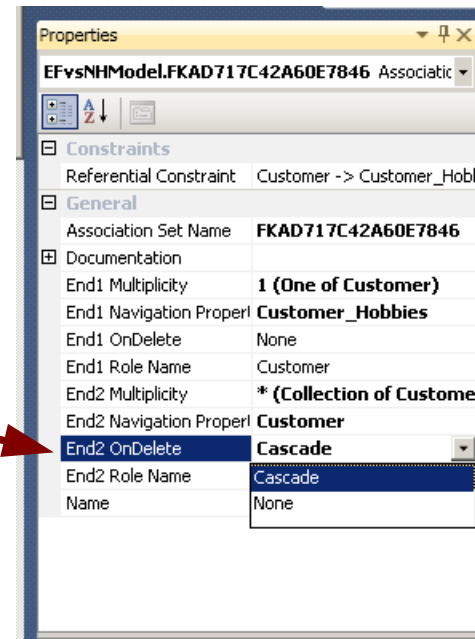
```

cascade= none |
           save-update |
           delete |
           all"
  
```

- Entity framework

```

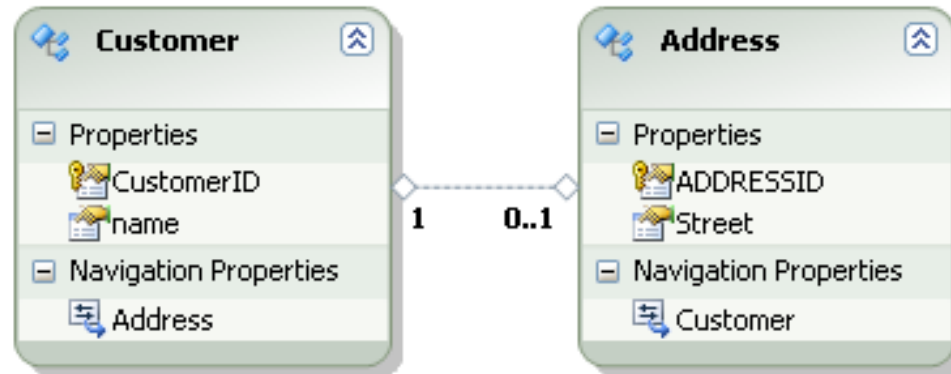
"save"
delete
  
```



Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

Beziehungskiste – Beispiel 1:1



Beziehungskiste – Beispiel 1:1



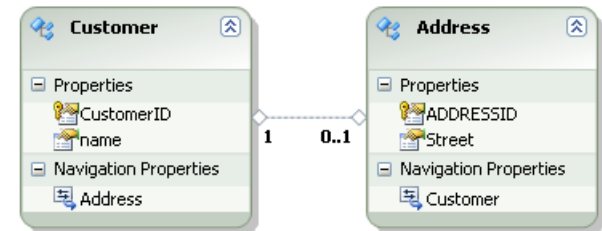
NHIBERNATE

Entity Framework

1 : 1 Beziehung mit Shared Primary Key

```
<class name="Customer" ...>
  <one-to-one name="Address"
    class="Address"
    cascade="save-update"/>
</class>
```

```
<class name="Address" ...>
  <id name="Id"
    column="ADDRESSID">
    <generator class="foreign">
      <param name="property">
        Customer
      </param>
    </generator>
  </id>
  <one-to-one name="Customer"
    class="Customer"
    constrained="true"/>
</class>
```



```
<EntityType Name="Address">
  <Key>
    <PropertyRef Name="ADDRESSID" />
  </Key>
  <Property Name="ADDRESSID" Type="int"
    Nullable="false" />
</EntityType>
<EntityType Name="Customer">
  <Key>
    <PropertyRef Name="CustomerID" />
  </Key>
  <Property Name="CustomerID" Type="int"
    Nullable="false"
    StoreGeneratedPattern="Identity" />
</EntityType>
```

Beziehungskiste – Beispiel 1:1



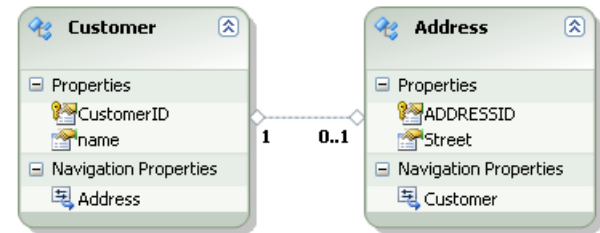
NHIBERNATE

Entity Framework

1 : 1 Beziehung mit Shared Primary Key

```
<class name="Customer" ...>
  <one-to-one name="Address"
    class="Address"
    cascade="save-update"/>
</class>
```

```
<class name="Address" ...>
  <id name="Id"
    column="ADDRESSID">
    <generator class="foreign">
      <param name="property">
        Customer
      </param>
    </generator>
  </id>
  <one-to-one name="Customer"
    class="Customer"
    constrained="true"/>
</class>
```



```
<Association Name="FK8C1490CB15229CAB">
  <End Role="Customer"
    Type="EFvsNHModel.Store.Customer"
    Multiplicity="1" />
  <End Role="Address"
    Type="EFvsNHModel.Store.Address"
    Multiplicity="0..1" />
  <ReferentialConstraint>
    <Principal Role="Customer">
      <PropertyRef Name="CustomerID" />
    </Principal>
    <Dependent Role="Address">
      <PropertyRef Name="ADDRESSID" />
    </Dependent>
  </ReferentialConstraint>
</Association>
```


Beziehungskiste – Beispiel 1:1



NHIBERNATE

Entity Framework

1 : 1 Beziehung mit Shared Primary Key

```
INSERT INTO Customer (name) VALUES (@p0);  
select SCOPE_IDENTITY();@p  
0 = 'Haug'
```

```
INSERT INTO Address (Street, ADDRESSID)  
VALUES (@p0, @p1);@p0 = 'Am  
Gründla 3c', @p1 = 1
```

```
insert [dbo].[Customer]([name])  
values (@0)  
select [CustomerID]  
from [dbo].[Customer]  
where @@ROWCOUNT > 0 and [CustomerID] =  
scope_identity()  
-- @0 (dbtype=String, size=50,  
direction=Input) = "Thomas Haug EF"
```

```
insert [dbo].[Address]([ADDRESSID],  
[Street])  
values (@0, @1)
```

```
-- @0 (dbtype=Int32, size=0,  
direction=Input) = 5  
-- @1 (dbtype=String, size=50,  
direction=Input) = "Am Gründla 3c EF"
```

Agenda

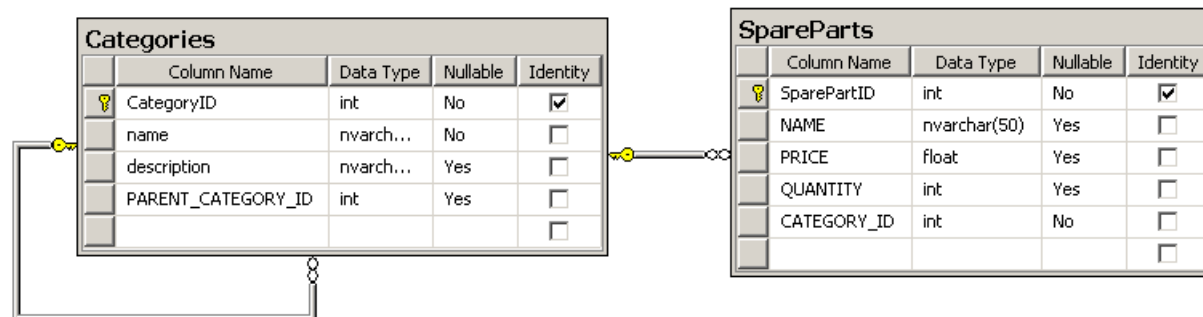
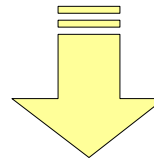
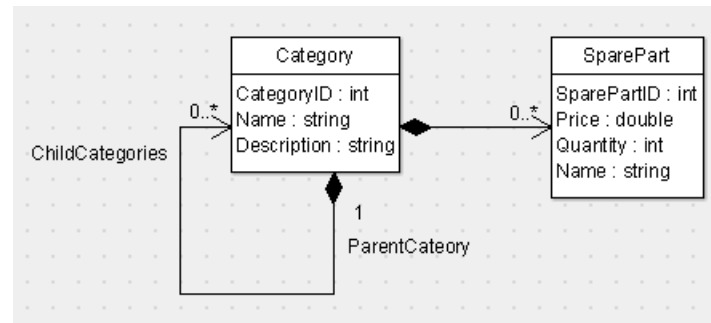
- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

Beziehungskiste – Erweitertes Beispiel

Anforderungen

- Eine Kategorie kann Unterkategorien besitzen
- Eine Kategorie kann Ersatzteile enthalten
- Ein Ersatzteil muss einer Kategorie zugeordnet sein
- Wird eine Kategorie gelöscht, so sollen alle enthaltenen Kategorien und Ersatzteile gelöscht werden

Erweitertes Beispiel - Modelle



Erweitertes Beispiel - Konfiguration



Entity Framework

```

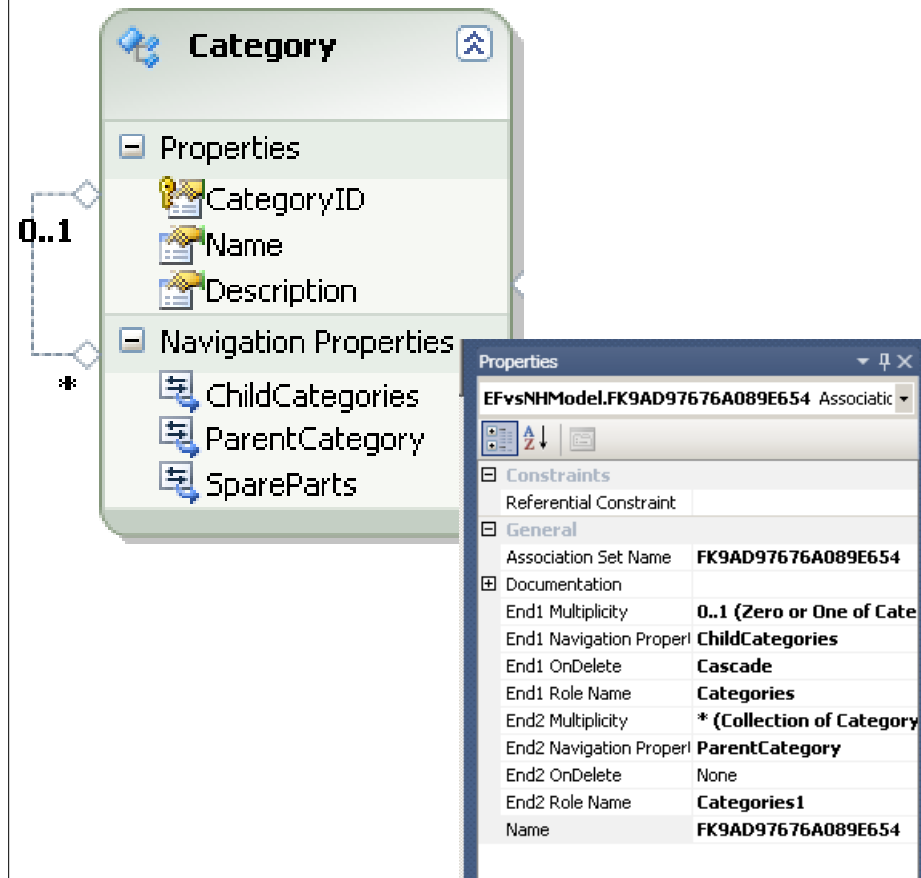
<hibernate-mapping
  xmlns="urn:nhibernate-mapping-2.2">

  <class name="Entities.Category,
    NHBeispielMitXML"
    table="Categories" >

    <set name="ChildCategories"
      table="Categories"
      cascade="all"
      lazy="true"
      inverse="true">

      <key
        column="PARENT_CATEGORY_ID"/>
      <one-to-many
        class="Entities.Category" />
    </set>

    <many-to-one
      name="ParentCategory"
      class="Entities.Category"
      column="PARENT_CATEGORY_ID"
      cascade="none"/>
  </class>
</hibernate-mapping>
  
```



Erweitertes Beispiel - Konfiguration



Entity Framework

```
<hibernate-mapping
  xmlns="urn:nhibernate-mapping-2.2">
  <class name="Entities.Category,
    NHBeispielMitXML"
    table="Categories" >
    <set name="ChildCategories"
      table="Categories"
      cascade="all"
      lazy="true"
      inverse="true">
      <key
        column="PARENT_CATEGORY_ID"/>
      <one-to-many
        class="Entities.Category" />
    </set>
    <many-to-one
      name="ParentCategory"
      class="Entities.Category"
      column="PARENT_CATEGORY_ID"
      cascade="none"/>
  </class>
</hibernate-mapping>
```

```
<edmx:Edmx Version="2.0" ...>
  <edmx:ConceptualModels>
    <AssociationSet Name="FK9....654"
      Association="EFvsNHModel.FK9....654">
      <End Role="ChildCategories"
        EntitySet="Categories" />
      <End Role="ParentCategory"
        EntitySet="Categories" />
    </AssociationSet>
    <Association Name="FK9....654">
      <End Role="ChildCategories"
        Type="EFvsNHModel.Category"
        Multiplicity="0..1" >
      <OnDelete Action="Cascade"/>
    </End>
      <End Role="ParentCategory"
        Type="EFvsNHModel.Category"
        Multiplicity="*" />
    </Association>
  </edmx:ConceptualModels>
```

Erweitertes Beispiel - Konfiguration

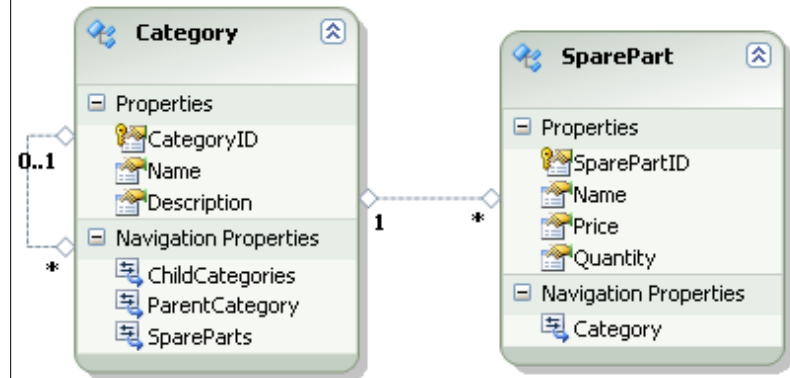


NHIBERNATE

```
<hibernate-mapping
  xmlns="urn:nhibernate-mapping-2.2">
  <class name="Entities.Category" >
    <set name="SpareParts"
      table="SPARE_PART"
      cascade="all"
      inverse="true">
      <key column="CATEGORY_ID" />
      <one-to-many
        class="Entities.SparePart" />
    </set>
```

```
<class name="Entities.SparePart" >
  <many-to-one name="Category"
    class="Entities.Category"
    column="CATEGORY_ID"
    not-null="true" />
```

Entity Framework



Properties	
EFvsNHModel.FKA14325FA4DFEDC56 Associati...	
<input checked="" type="checkbox"/> Constraints Referential Constraint	
<input checked="" type="checkbox"/> General	
Association Set Name	FKA14325FA4DFEDC56
<input checked="" type="checkbox"/> Documentation	
End1 Multiplicity	1 (One of Category)
End1 Navigation Proper	SpareParts
End1 OnDelete	Cascade
End1 Role Name	Categories
End2 Multiplicity	*(Collection of SparePar
End2 Navigation Proper	Category
End2 OnDelete	None
End2 Role Name	SpareParts
Name	FKA14325FA4DFEDC56

Erweitertes Beispiel - Konfiguration



Entity Framework

```
<hibernate-mapping
  xmlns="urn:nhibernate-mapping-2.2">
  <class name="Entities.Category" >
    <set name="SpareParts"
      table="SPARE_PART"
      cascade="all"
      inverse="true">
      <key column="CATEGORY_ID" />
      <one-to-many
        class="Entities.SparePart" />
    </set>

    <class name="Entities.SparePart" >
      <many-to-one name="Category"
        class="Entities.Category"
        column="CATEGORY_ID"
        not-null="true" />
    </class>
  </class>
</hibernate-mapping>
```

```
<edmx:Edmx Version="2.0" ...>
  <edmx:ConceptualModels>
    <AssociationSet Name="FKA...C56"
      Association="EFvsNHModel.FKA...C56">
      <End Role="Categories"
        EntitySet="Categories" />
      <End Role="SpareParts"
        EntitySet="SpareParts" />
    </AssociationSet>

    <Association Name="FKA...C56">
      <End Role="Categories"
        Type="EFvsNHModel.Category"
        Multiplicity="1" >
      <OnDelete Action="Cascade" />
    </End>
    <End Role="SpareParts"
      Type="EFvsNHModel.SparePart"
      Multiplicity="*" >
    </End>
  </edmx:ConceptualModels>
```


Erweitertes Beispiel - Objektgraph speichern



NHIBERNATE

Entity Framework

```
Category rootCategory =
    new Category("Käfer-Teile");

Category motorCat =
    new Category("Motorteile");
motorCat.Description =
    "Alles für den Käfer Motor";

SparePart kipphebel =
    new SparePart("Standard Kipphebel",
        250.0, 5);

motorCat.AddSparePart(kipphebel);

rootCategory.AddChildCategory(motorCat);
```

```
using (ITransaction tx =
    session.BeginTransaction())
{
    session.Save(rootCategory);
    tx.Commit();
}
```

```
using (EFvsNHEntities ctx =
    new EfvsNHEntities())
{
    ctx.Categories.AddObject(rootCategory);
    context.SaveChanges();
}
```

Erweitertes Beispiel – Im Graph navigieren



NHIBERNATE

Entity Framework

Eager fetching

```
<hibernate-mapping
  xmlns="urn:nhibernate-mapping-2.2">
  <class name="Entities.Category" >
    <set name="SpareParts"
      table="SPARE_PART"
      cascade="all"

      lazy = "false"
      fetch = "select" | "join"

      inverse="true">
      <key column="CATEGORY_ID"/>
      <one-to-many
        class="Entities.SparePart" />
    </set>
```

Keine Entsprechung im Entity Framework
(!per Konfiguration!)

Erweitertes Beispiel – Im Graph navigieren



NHIBERNATE

Entity Framework

Eager fetching

```
string queryStr =
    "from Category cat

    left join fetch cat.ChildCategories
    left join fetch cat.SpareParts

    where cat.CategoryID = :id";

IQuery query=
    session.CreateQuery(queryStr)
        .SetInt32("id", id);

Category root =
    query.UniqueResult<Category>();
```

```
Category tempRoot =
    (from cat1 in
        context.Categories
        .Include("ChildCategories")
        .Include("SpareParts")

        where cat1.CategoryID ==
            rootCategory.CategoryID
        select cat1)
    .Single<Category>();
```

Erweitertes Beispiel – Im Graph navigieren



NHIBERNATE

Entity Framework

Lazy Loading

Keine direkte Entsprechung in NHibernate

Explicit Lazy Loading

```
Category tempRoot= (Category)
ctx.GetObjectByKey(rootCategory.EntityKey);
tempRoot.ChildCategories.Load();
Console.WriteLine("Direkte Kinder {0}",
    tempRoot.ChildCategories.Count());
foreach (var category
    in tempRoot.ChildCategories) {
    category.SpareParts.Load();
    Console.WriteLine("Ersatzteile {0}",
        category.SpareParts.Count());
}
```

Erweitertes Beispiel – Im Graph navigieren



NHIBERNATE

Entity Framework

Lazy Loading

Implicit Lazy Loading

```
ctx.ContextOptions
    .DeferredLoadingEnabled = true;
```

Default Verhalten

```
Category tempRoot= (Category)
ctx.GetObjectByKey (rootCategory.EntityKey);

Console.WriteLine("Direkte Kinder {0}",
    tempRoot.ChildCategories.Count());

foreach (var category
    in tempRoot.ChildCategories) {
    Console.WriteLine("Ersatzteile {0}",
        category.SpareParts.Count());
}
```

Agenda

- Eigenschaften
- Architektur-Überblick
- Einfaches Beispiel
 - Konfiguration
 - CRUD Beispiel
 - Lebenszyklus
- Beziehungskiste
 - Multipizitäten
 - Persistence-by-Reachability
 - Beispiel für 1:1 Assoziation
 - Erweitertes Beispiel (1:N) Beziehungen
- Fazit

Fazit

- Entity Framework und NHibernate sind im Kern ähnlich
- Wesentliche Unterschiede
 - Entity Framework ist (noch) stark Visual Studio verankert
 - Datenbank Support
 - CASCADE Funktionalität
 - *Unterstützung von verschiedenen Collection Typen*
- ABER:
 - Viele Aspekte wurden nicht beleuchtet: z. B. Abfragesprachen, Einbetten von Funktionen, Vererbung, Performance ...

14.–17. 09. 2009
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Vielen Dank!

Thomas Haug

MATHEMA Software GmbH