hybris Developer Training Part I - Core Platform

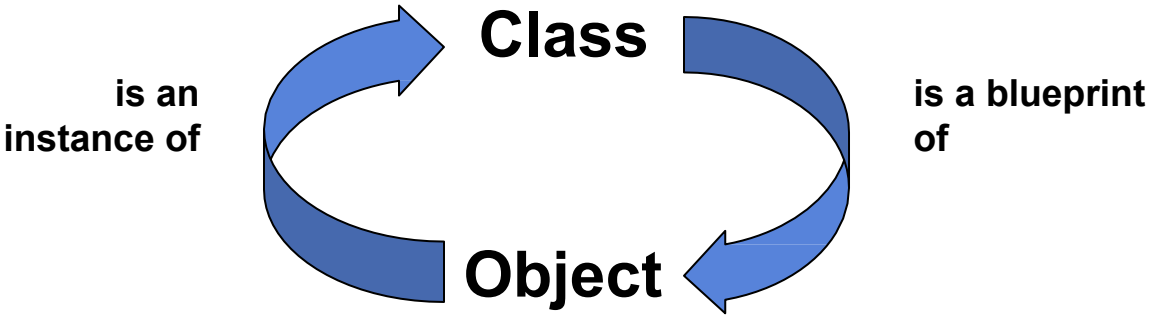
Data Modeling

Introduction to the Type System

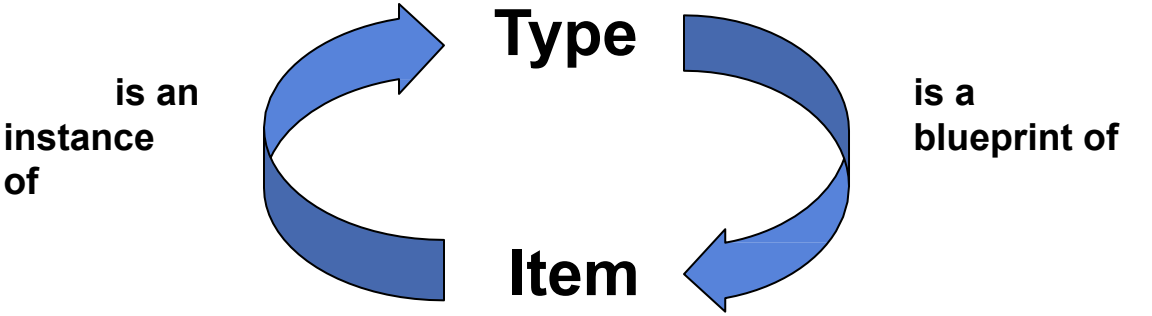
2. Relations

Java vs hybris

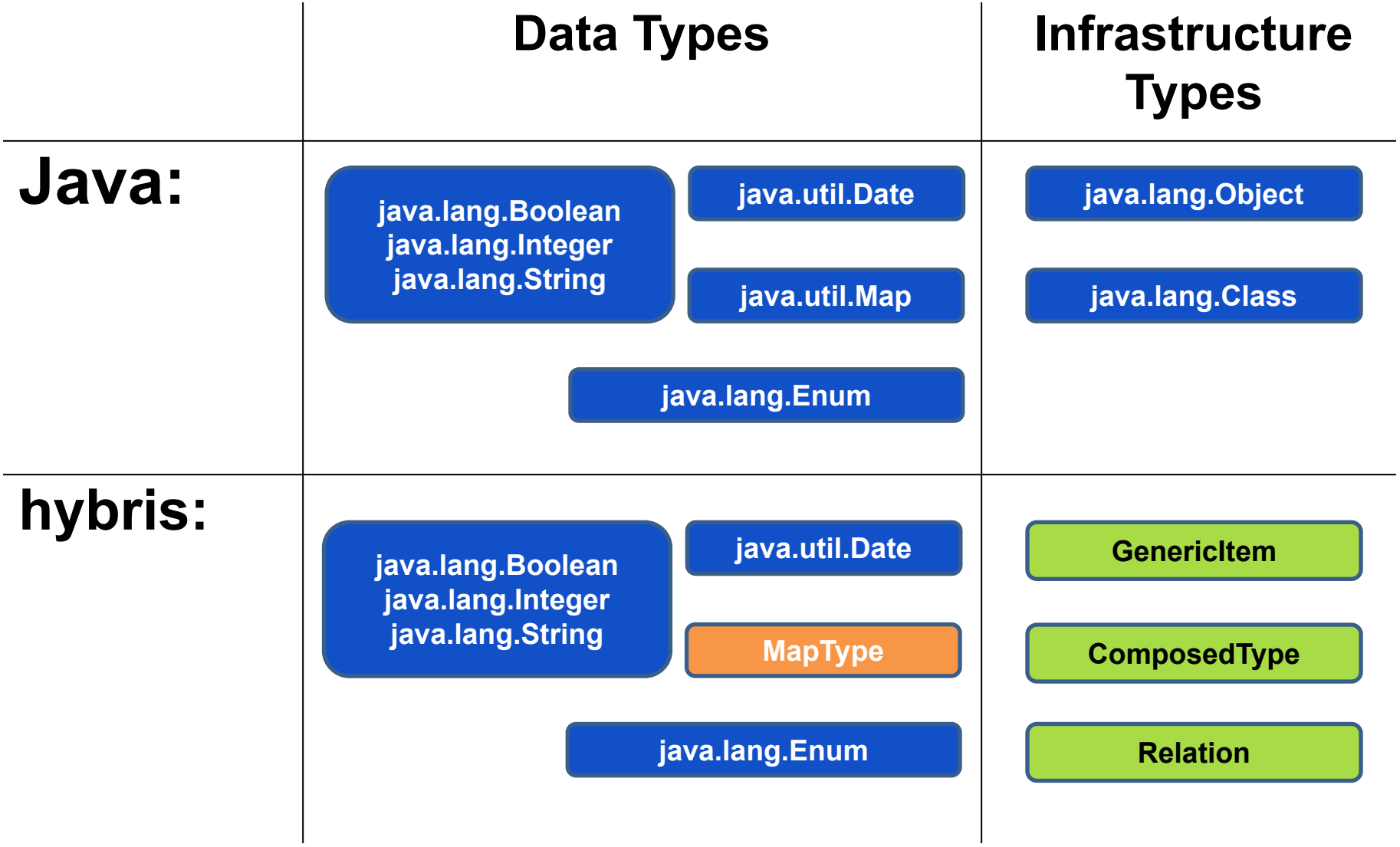
Java:



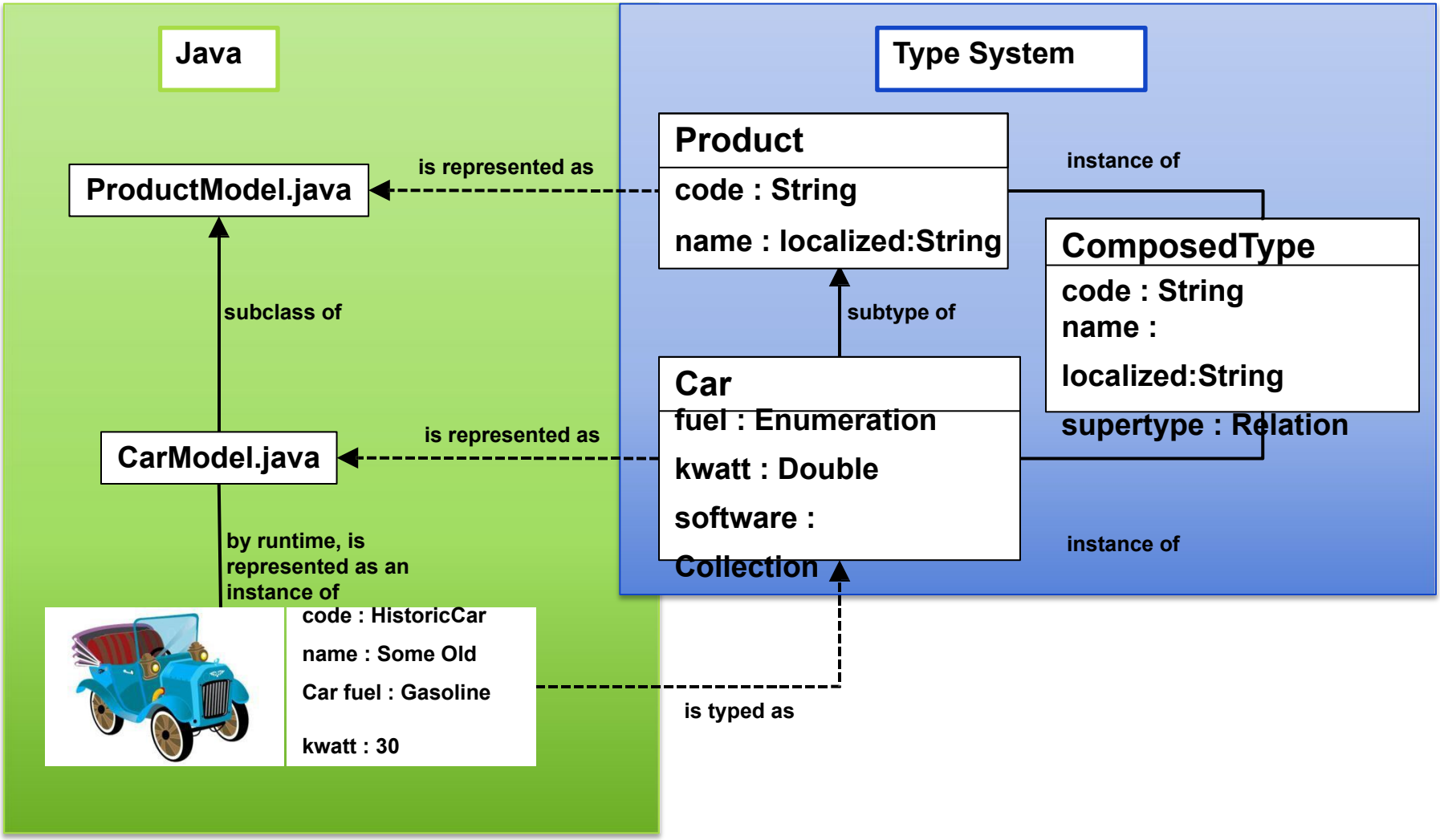
hybris:



Best of Both Worlds



Java Classes vs hybris Types



Attributes of a (Composed) Type

- An attribute in the hybris Multichannel Suite
  - Can be a reference to
    - a basic Java type, called an "atomic type"
      (for example, the **code** attribute is of type **java.lang.String**)
    - a composed type
      (for example, the **picture** attribute is of type **Media**)
  - Can have a localized name and description ("Image" / "Bild")
  - Can have a default value

Attributes of a (Composed) Type

PK	code	name	…	mechanic
8796256894977	vehicle01	Ferrari F40	…	8796128083972

product

Atomic type property – value stored directly in db. For instance, property of type String is stored as VARCHAR

Composed type property – reference to another item. The db column stores a PrimaryKey (PK) value.

code	vehicle01
name	Ferrari F40
mechanic	Nikola Tesla

PK

8796128083972

name	…
Nikola Tesla	…

users

Extending the Data Model

- Create new types:
  - Define a type by extending already existing types, such as:

```
<itemtype code="Car" extends="Product">
```

  - Define “completely new types”, such as:

```
<itemtype code="Car">
```

    (implicitly extends from **GenericItem**)

- Extend existing types:
  - Add attribute definitions to existing types (attribute injection), such as:

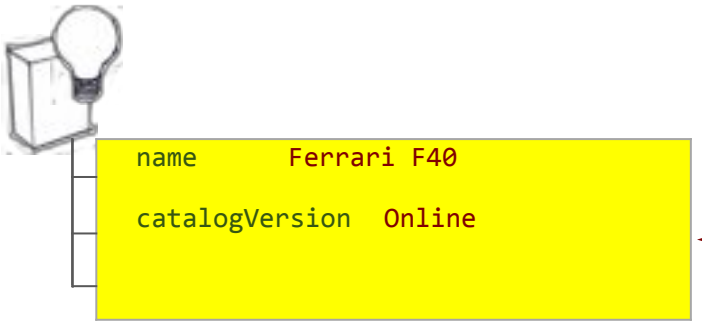
```
<itemtype code="Product" ...>
        ...
        <attribute qualifier="MyAttribute">
</itemtype>
```

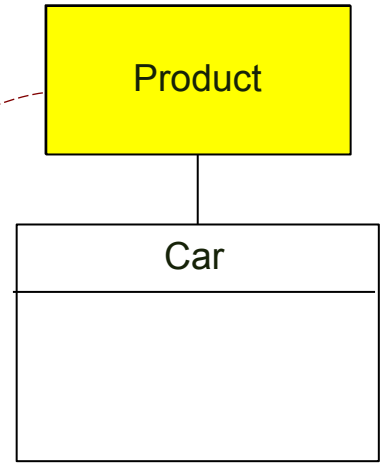
  - Redefine inherited attribute definitions from super type

```
<attribute qualifier="code" redeclare="true">
```

    If you change the attribute's java type, the new type must extend the original type

hybris Type System • Extending

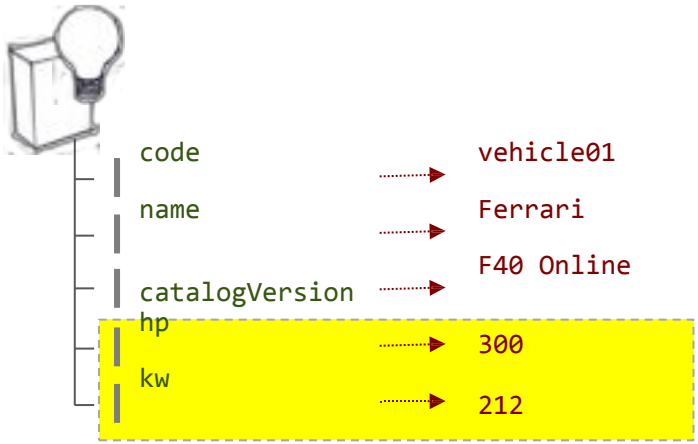


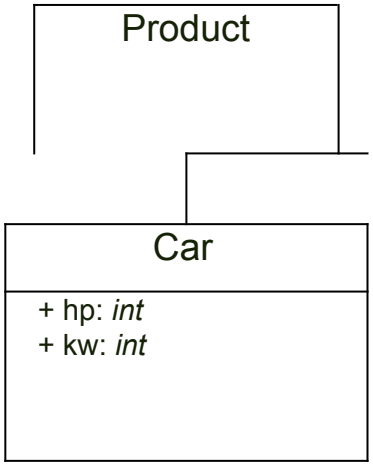



```
<items>

  <itemtypes>
     <itemtype code="Car" extends="Product" autocreate="true"
               generate="true"
               jaloclass="de.hybris.platform.lecture.jalo.Car">
        ...
```

hybris Type System • Attributes



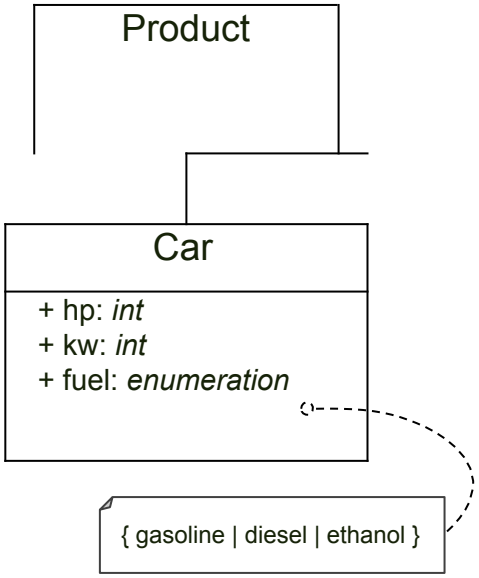



```
<items>

  <itemtypes>
     <itemtype code="Car" extends="Product" autocreate="true" generate="true">
        <attributes>
           <attribute qualifier="hp" type="java.lang.Integer">
              <description>Horse Power</description>
              <persistence type="property" />
           </attribute>
           <attribute qualifier="kw" type="java.lang.Integer">
              <description>Kilowatt</description>
              <persistence type="dynamic"
                           attributeHandler="kwPowerAttributeHandler" />
              <modifiers write="false" />
           </attribute>
```

hybris Type System • Enumerated types

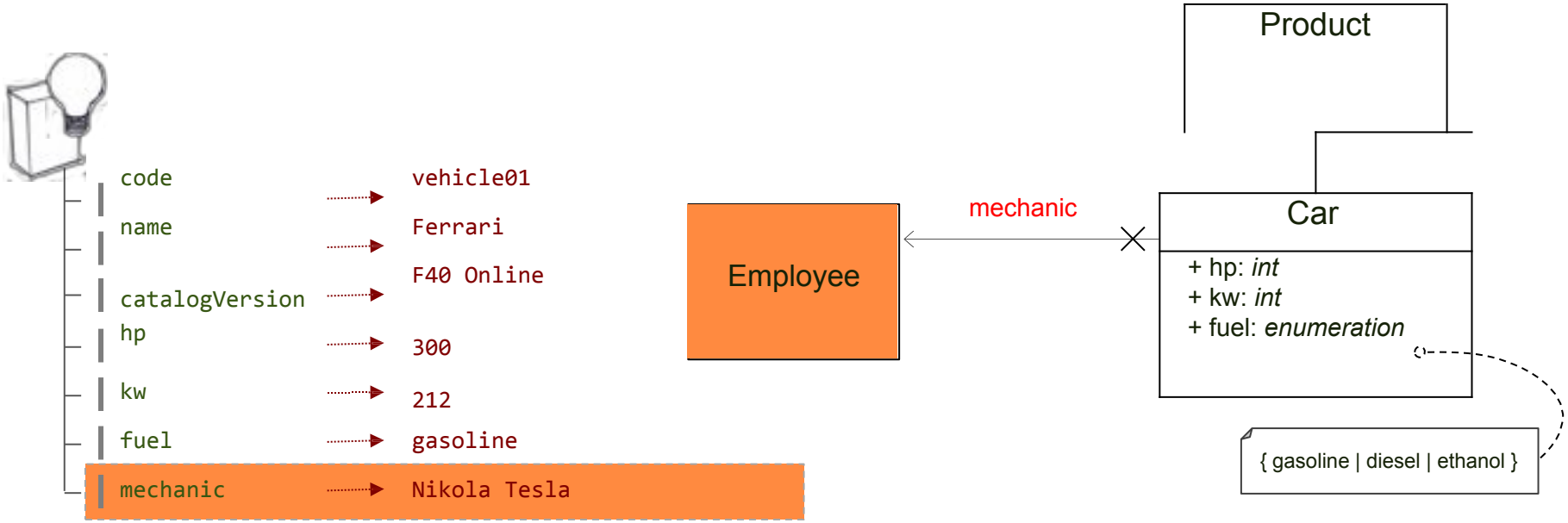
code	vehicle01
name	Ferrari
catalogVersion	F40 Online
hp	300
kw	212




```
<items>
  <itemtypes>
     <itemtype code="Car" extends="Product" autocreate="true" generate="true">
        <attributes>
           <attribute qualifier="hp" type="java.lang.Integer">
              <description>Horse Power</description>
              <persistence type="property" />
           </attribute>
           <attribute qualifier="kw" type="java.lang.Integer">
              <description>Kilowatt</description>
              <persistence type="dynamic"
                           attributeHandler="kwPowerAttributeHandler" />
              <modifiers write="false" />
           </attribute>
           <attribute qualifier="fuel" type="FuelEnumeration">
              <description>Fuel for this car</description>
              <persistence type="property"></persistence>
           </attribute>
```

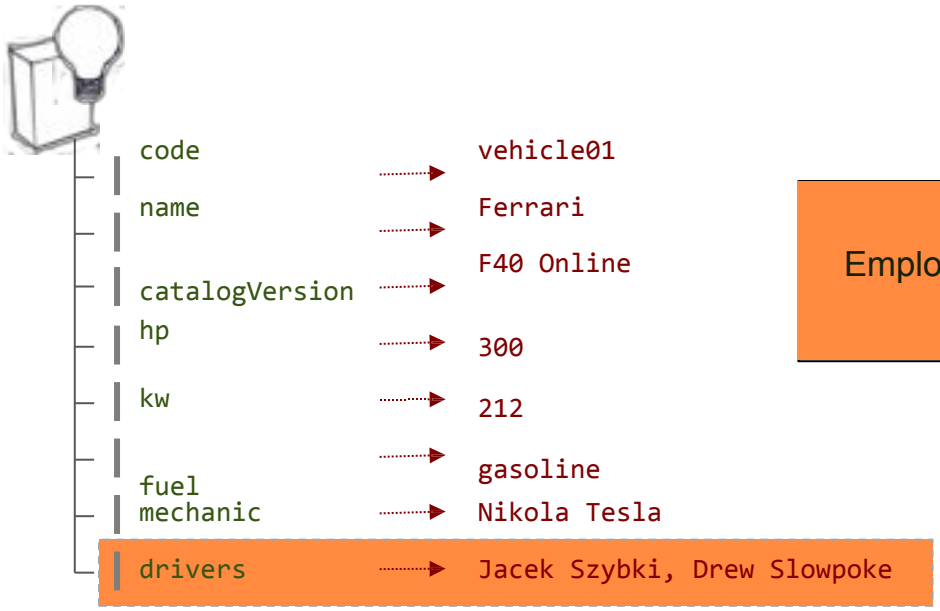
```
<enumtypes>
  <enumtype code="FuelEnumeration"
            generate="true"
            autocreate="true
            "
            dynamic="true">
     <value code="diesel"></value>
     <value code="gasoline"></value>
     <value code="ethanol"></value>
</enumtype>
```

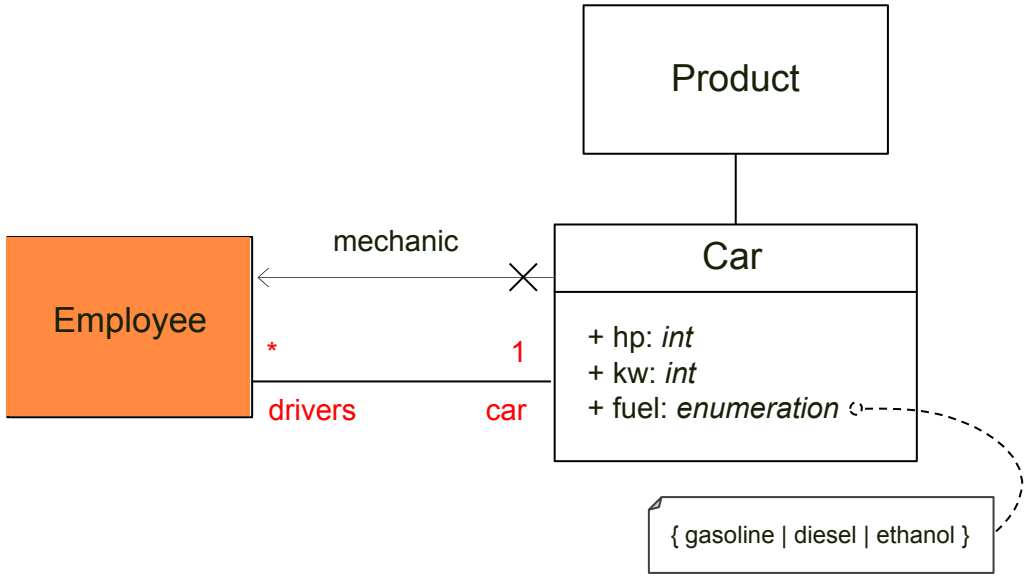
hybris Type System • Composed Type reference




```
<items
  <itemtypes>
     <itemtype code="Car" extends="Product" autocreate="true" generate="true">
        <attributes>
           ...
           <attribute qualifier="mechanic" type="Employee">
              <modifiers read="true" write="true" search="true" />
              <persistence type="property" />
           </attribute>
```

hybris Type System • Relations






```
<items>
   ...
   <relations>
      <relation generate="true" localized="false" code="Car2EmployeeRelation" autocreate="true">
         <sourceElement qualifier="car" type="Car" cardinality="one"></sourceElement>
         <targetElement qualifier="drivers" type="Employee" cardinality="many"></targetElement>
      </relation>
   </relations>

  <itemtypes>
     ...
```

hybris Type System • Table structure

PK	code	catVersion	…	hp	fuel	mechanic
8796256894977	vehicle01	Online	…	300	8796103934043	8796128083972

product

code	vehicle01
name	Ferrari
catalogVersion	F40 Online
hp	300
kw	212
fuel	gasoline
mechani	Nikola Tesla
c	Jacek Szybki, Drew Slowpoke
drivers	

car2Employees

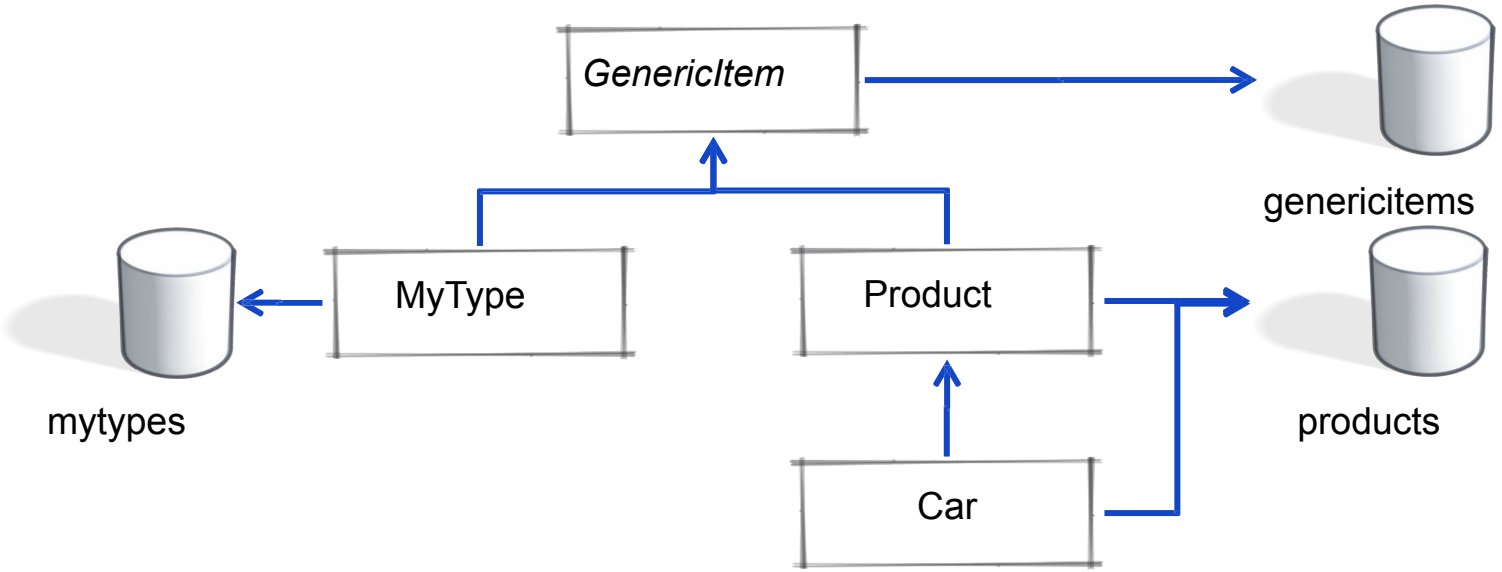
Car	Employee
8796256894977	8796191391748
8796256894977	8796191358980

users

PK	uniqueid	name	passwd	…
8796128083972	mechanic01	Nikola Tesla	Wkxo&k3*jl!jL	…
8796191391748	driver16	Jacek Szybki	yP$jHYYnz(kQA	…
8796191358980	driver16	Drew Slowpoke	_#kJUPPcau&!i	…

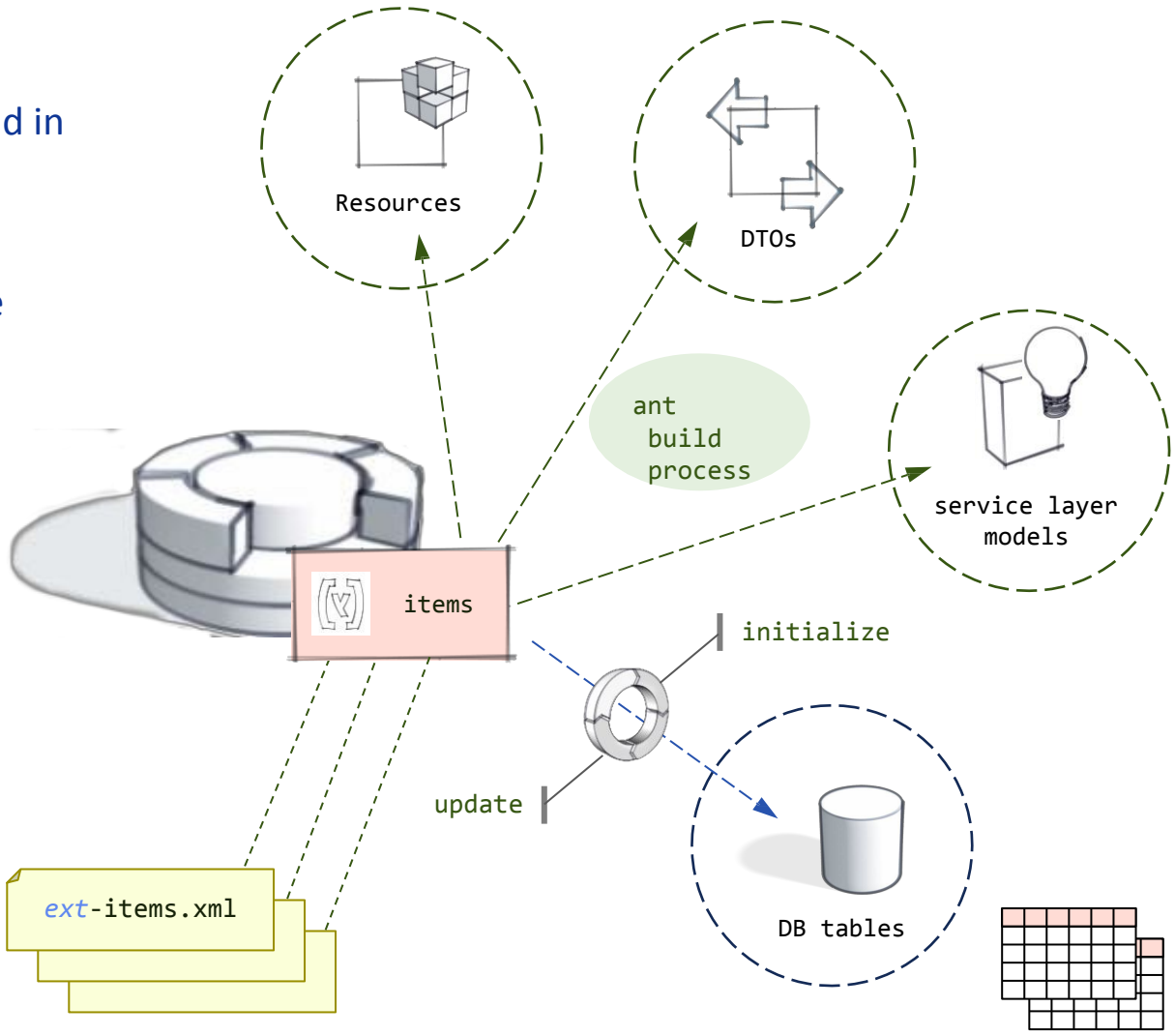
How Items are Stored in the Database

- By default, all items for subtypes are stored in the same database tables as their supertypes
- Technically, the process of specifying into which database table the items of a type are stored is called *deployment*
- Each Item type can have its own db table
  - specify deployment only for the first layer of `GenericItem` subtypes



hybris Type System • Automatic Generation

1. hybris item definitions are found in each extension's *extensionName*-items.xml
2. The ant process assembles type definitions and generates Models, DTOs, and Resources.
3. hybris also creates the required tables.
4. Invoking initialize or update creates the required table.



Quiz Questions

1. What is the equivalent of a Java object in hybris?

2. What is the difference between a composed type and an atomic type in hybris?

3. What file is used to configure types in hybris?

4. In hybris, can you create new types, or only extend existing types?

5. Do you have to define a model-class in Java after defining it in xml?

6. Explain the notion of *deployment* in hybris.

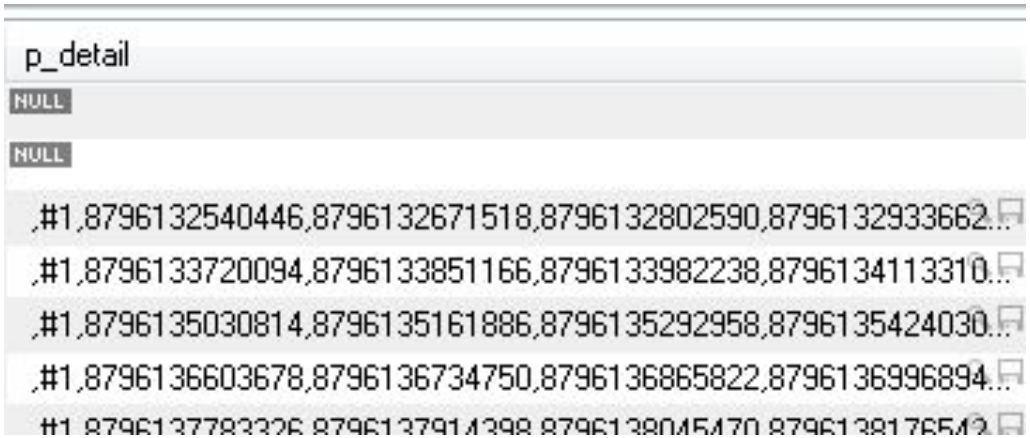
1. Introduction to the Type System

Relations

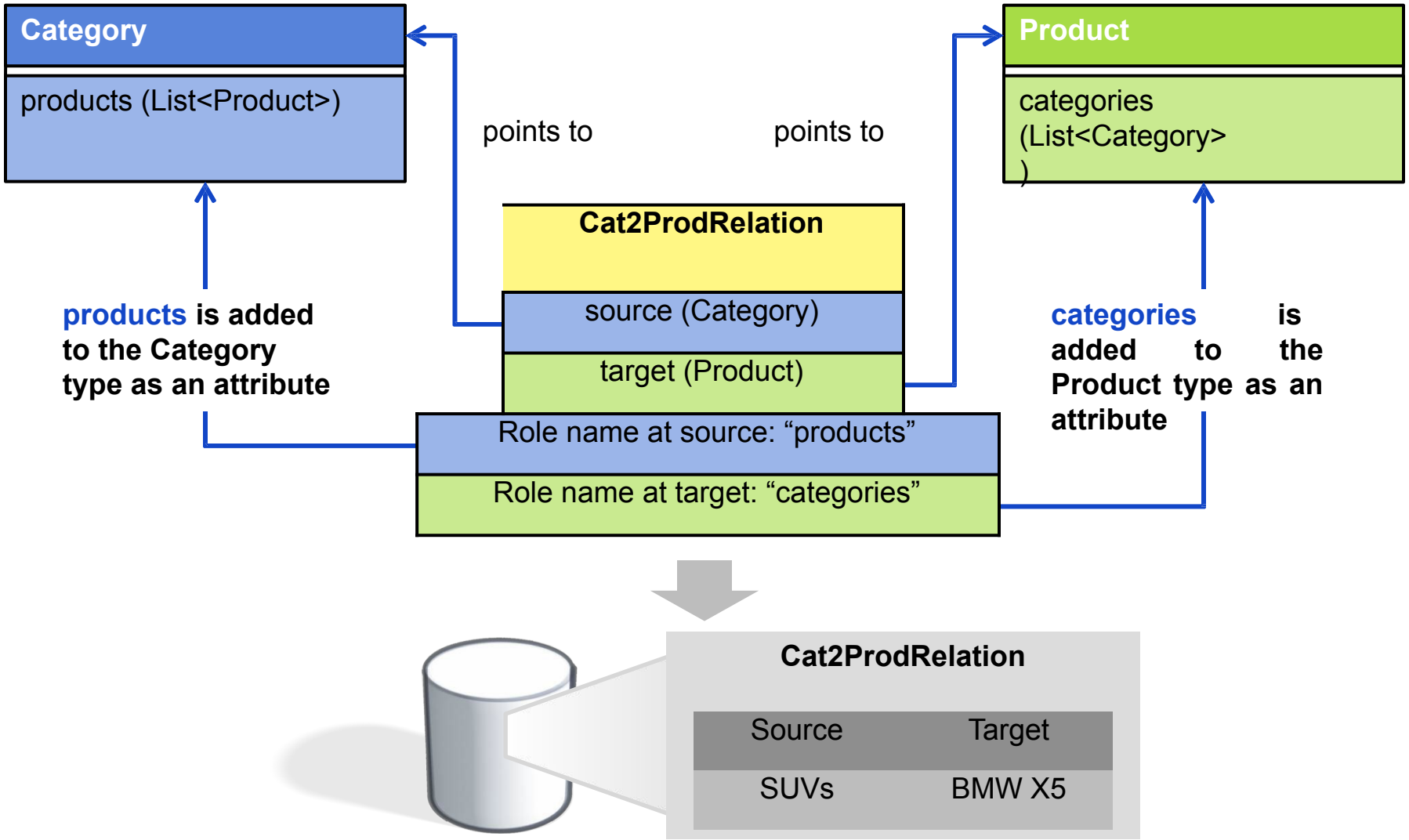
Modeling Collections in hybris

Basically, there are two technically different ways of modeling collections in hybris:

- CollectionTypes
- RelationTypes
  - Think of CollectionTypes in hybris as a backpack mounted onto a type
  - By runtime, CollectionTypes are resolved into a Collection of a kind of item, such as a List of MediaModels
  - On the database level, CollectionTypes are a comma-separated list of PKs



Relations in hybris – An Overview



What's so Important About Relations?

Enhanced flexibility:

- Create links between all kinds of types
- Create type-safe n-to-m relations:
  - Only link such elements of the source / target type declared at the relation

What's so Important About Relations?

If in doubt: Use relations, do not use `CollectionTypes` because:

- Opposite side is not "aware" of the `CollectionType`
- `CollectionTypes` are stored in a database field as a comma-separated list of references (PKs)
- Can cause overflow, resulting in truncation and therefore loss of data
- More difficult to search and lower performance

Implicitly, relations are collections at source and target type, but …

- Values for relations are stored in a separate database table
- Each value is stored in a separate table row

Quiz Questions

1. Name the two ways to model a collection in hybris

2. How are collection types stored in the database?

3. Specify some advantages and disadvantages of collection types

4. What is a relation in hybris?

References

- /wiki.hybris.com/display/release5/Type+System+Documentatio
- m wiki.hybris.com/display/release5/items.xml
- wiki.hybris.com/display/release5/
  Specifying+a+Deployment+for+hybris+Platform+Types