buschmais

Beratung . Technologie . Innovation

JavaLand 2014

Exploration and Verification of Java Applications Using a Graph Database

Dirk Mahler
buschmais GbR

Inhaber
Torsten Busch, Frank Schwarz,
Dirk Mahler und Tobias Israel

dirk.mahler@buschmais.com
http://www.buschmais.de/

Phantasialand Brühl, 05/26/2014

Agenda

buschmais
Beratung . Technologie . Innovation

- Introducing A Graph Database: Neo4j
- Modeling Software Structures As A Graph
- Exploring An Application Using Queries
- Live Demo #1
- Structures, Rules and Erosion
- Validation Of Conventions And Constraints
- jQAssistant
- Live Demo #2

buschmais

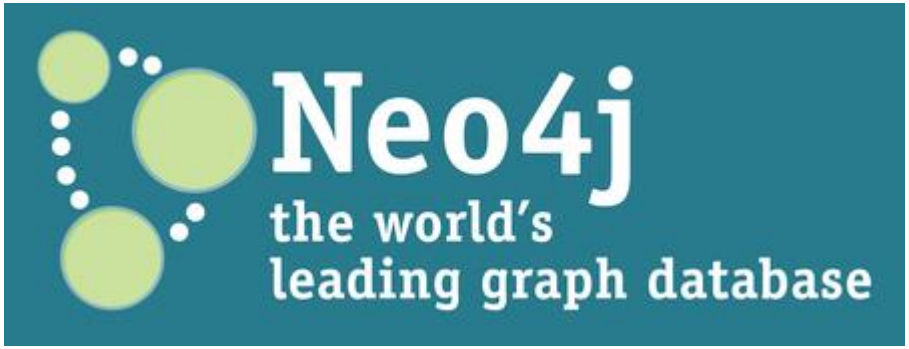
Beratung . Technologie . Innovation

Exploration And Verification Of Java Applications Using A Graph Database

Introducing A Graph Database – Neo4j

Introducing A Graph Database – Neo4j

buschmais
Beratung . Technologie . Innovation



- Some facts
  - http://www.neo4j.org
  - Latest Stable Release: 2.0.1
  - Implemented in Java(!)
  - Runnding embedded with native Java API…
  - ….or as standalone server via REST
  - Several Language Bindings, e.g. Java, JS, Ruby, PHP, .NET, …
  - HA features
  - Query language: Cypher
  - Comprehensive documentation and online tutorials
  - Community (Open source) and commercial licenses available

Exploration And Verification Of Java Applications
Using A Graph Database

Modeling Software Structures As A Graph

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

- Let‘s model a Java class as a graph!

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

- Let‘s model a Java class as a graph!

```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

- All we need is…
  - Nodes
  - Labels
  - Properties
  - Relationships
- Modeling is just…
  - Taking a pen
  - Drawing the structures on a whiteboard (i.e. the database)
- We don‘t need…
  - Foreign keys
  - Tables and schemas
  - Knowledge in graph theory

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```

Modeling Software Structures As A Graph

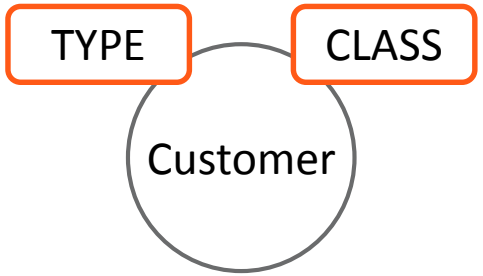
buschmais
Beratung . Technologie . Innovation

```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

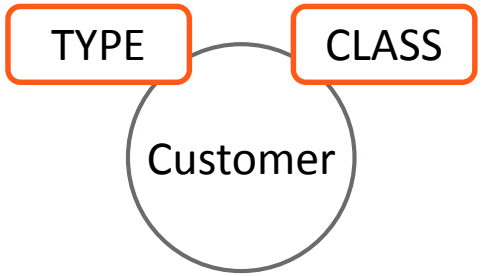
```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



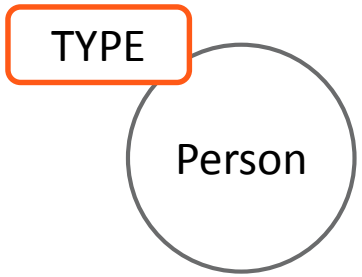
FQN:com.buschmais.model.Customer
VISIBILITY:PUBLIC

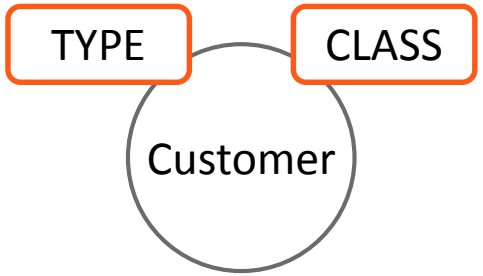
Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```

FQN:com.buschmais.model.Person





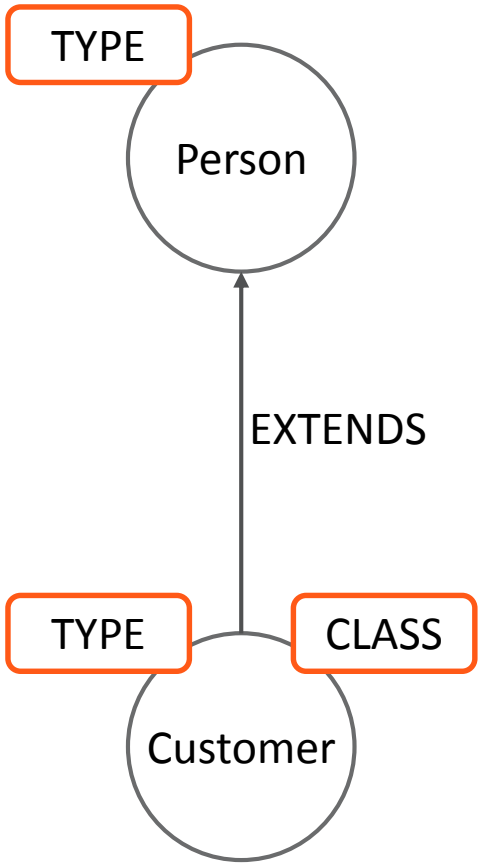
FQN:com.buschmais.model.Customer
VISIBILITY:PUBLIC

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```

FQN:com.buschmais.model.Person

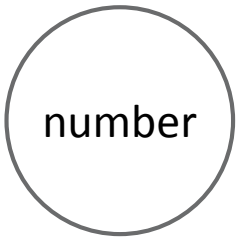


FQN:com.buschmais.model.Customer
VISIBILITY:PUBLIC

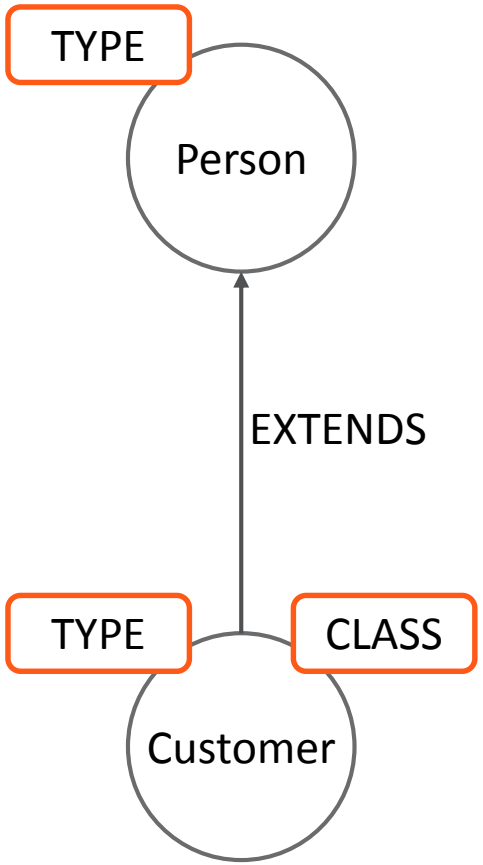
Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



FQN:com.buschmais.model.Person

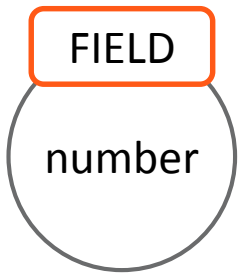


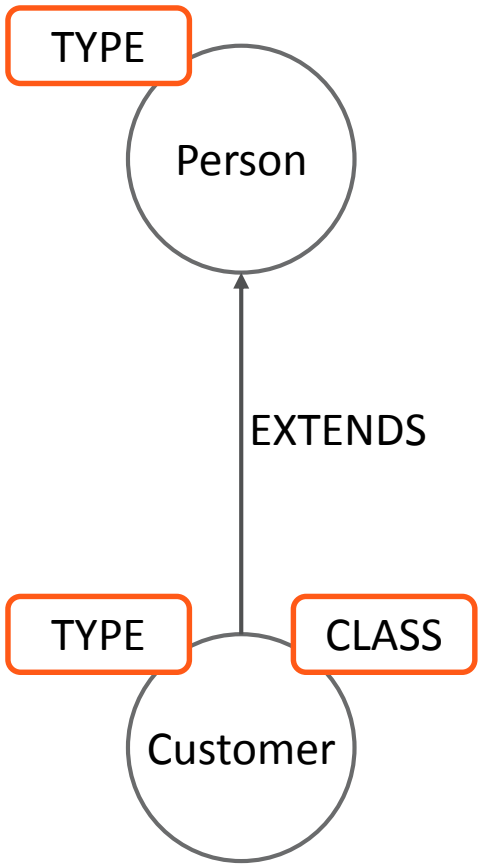
FQN:com.buschmais.model.Customer
VISIBILITY:PUBLIC

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```

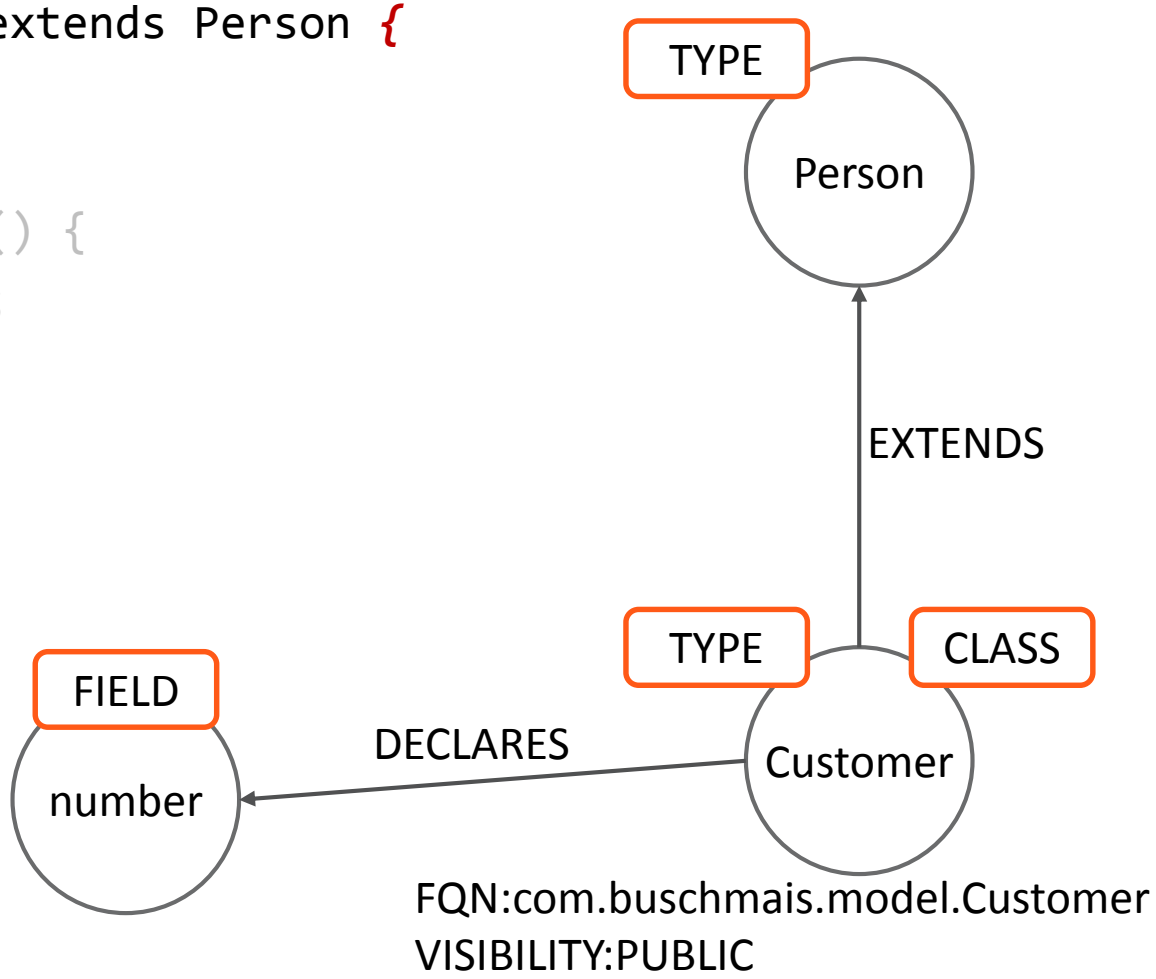




Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

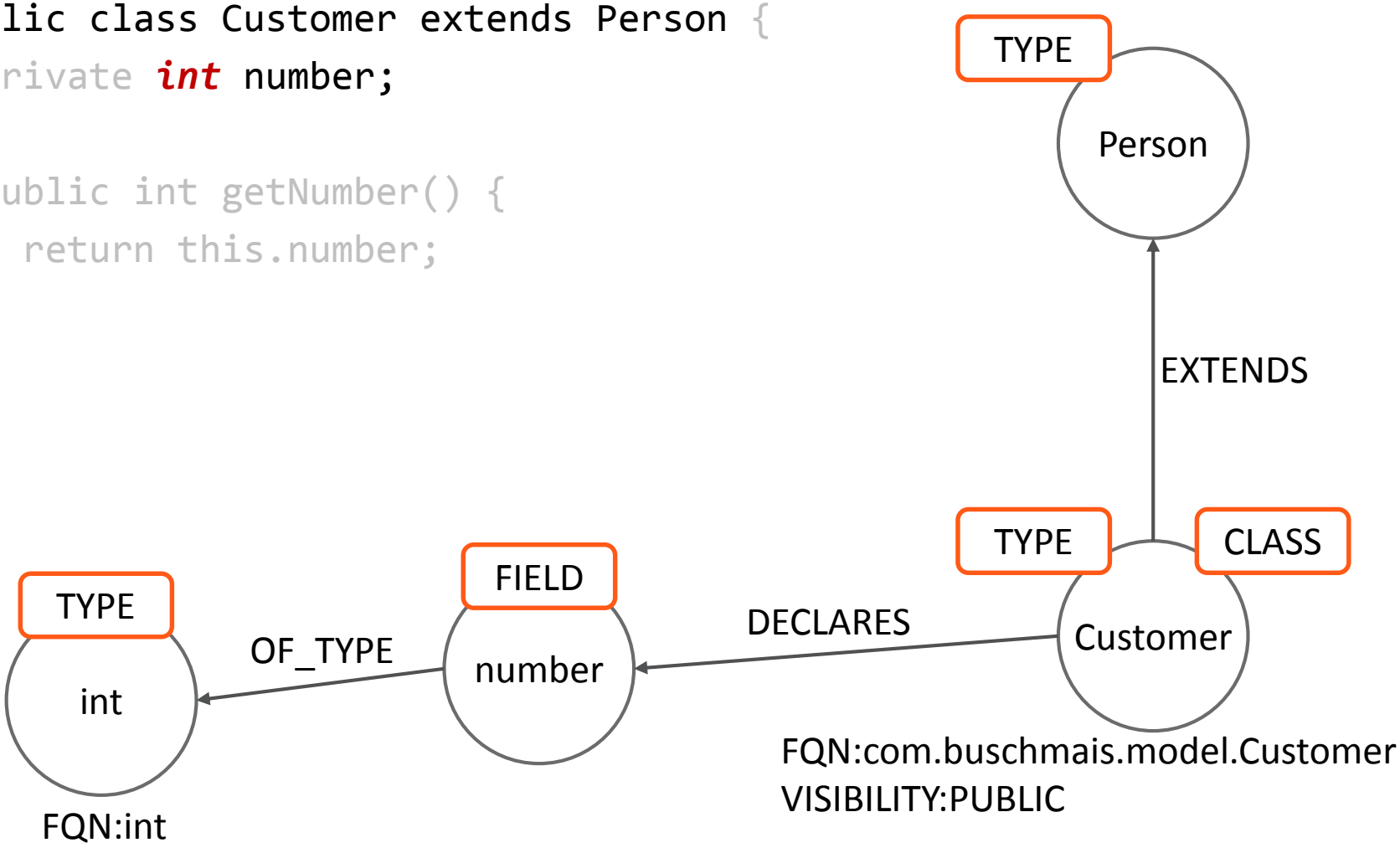
```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

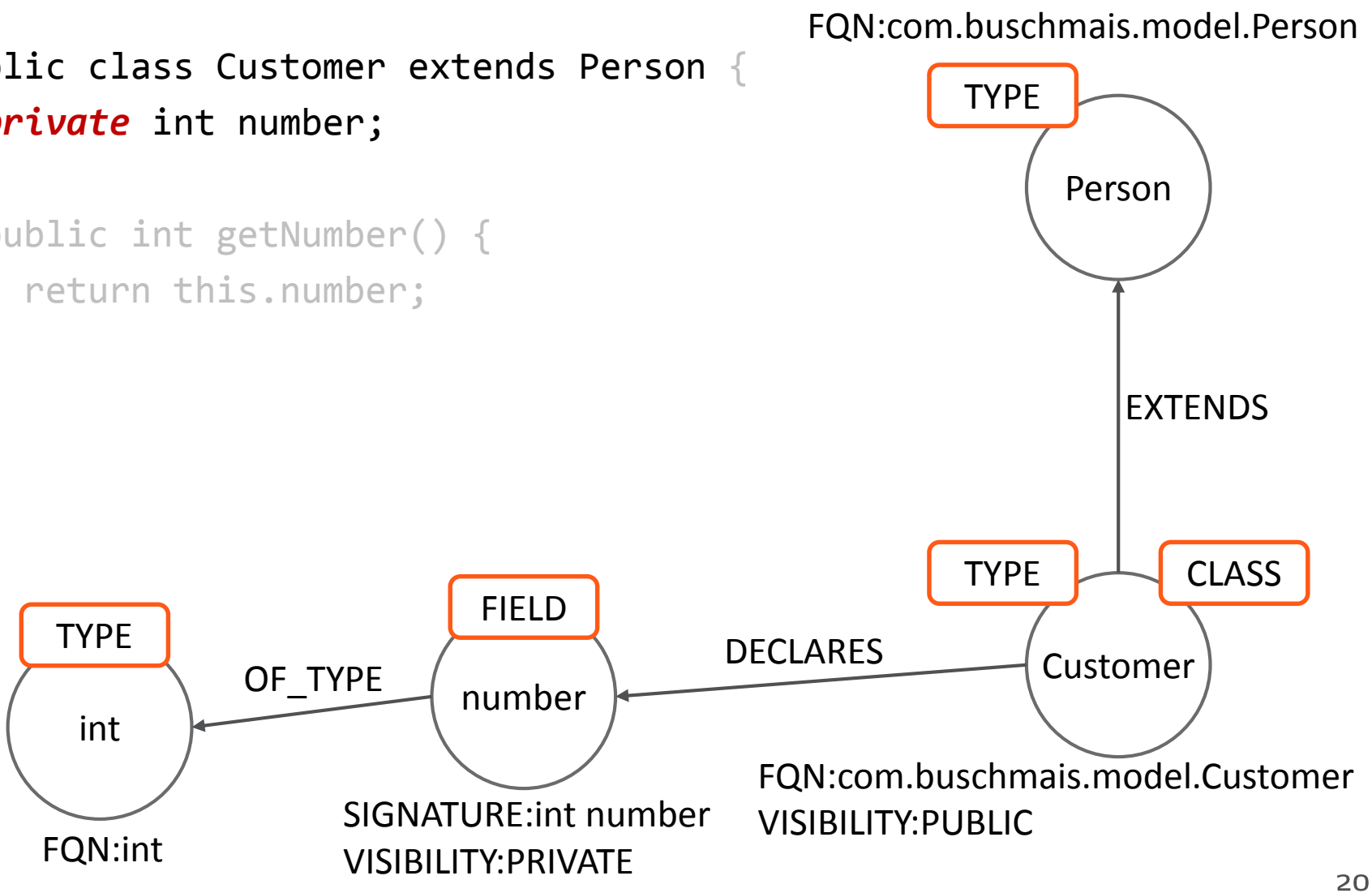
```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

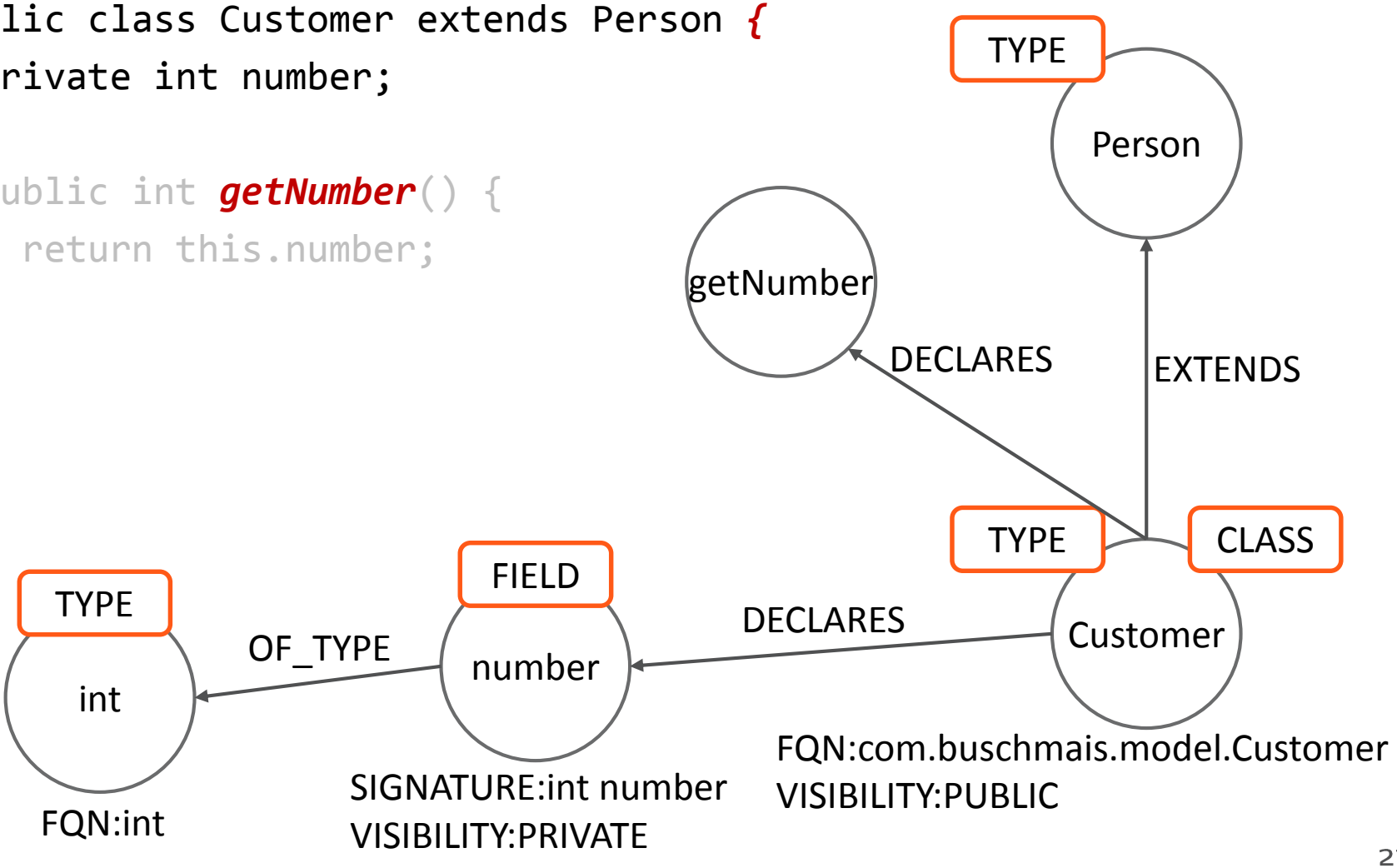
```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

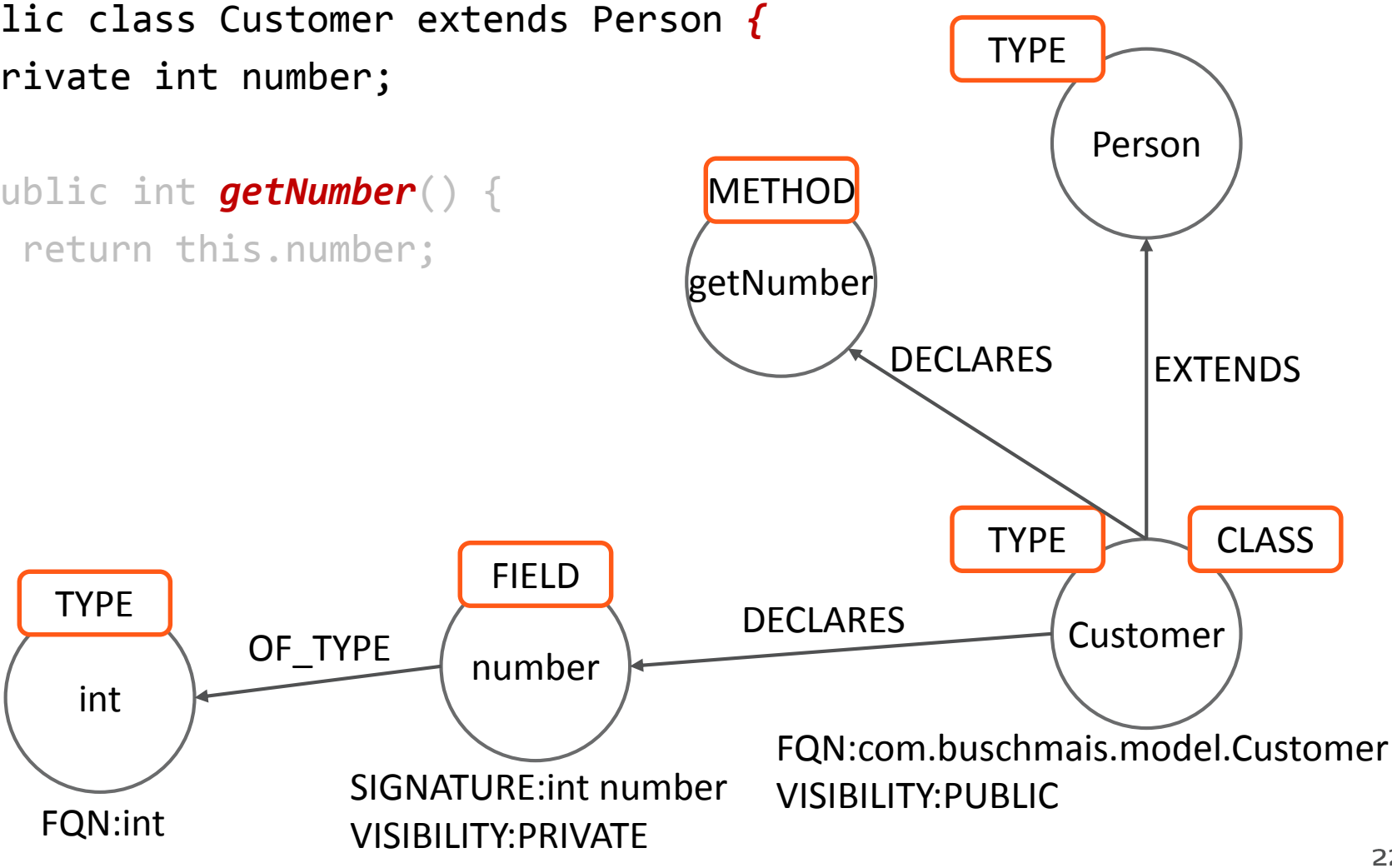
```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

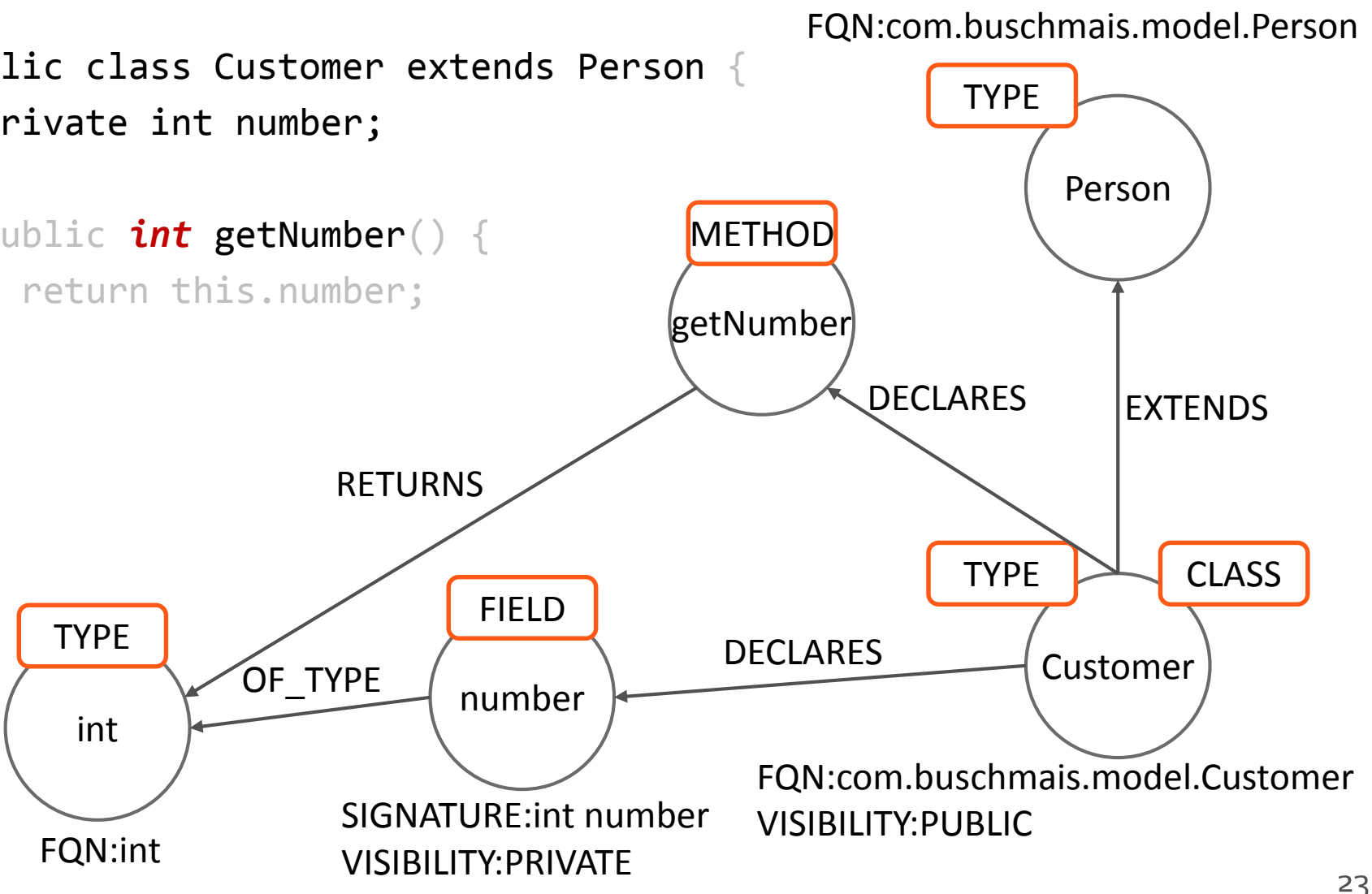
```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

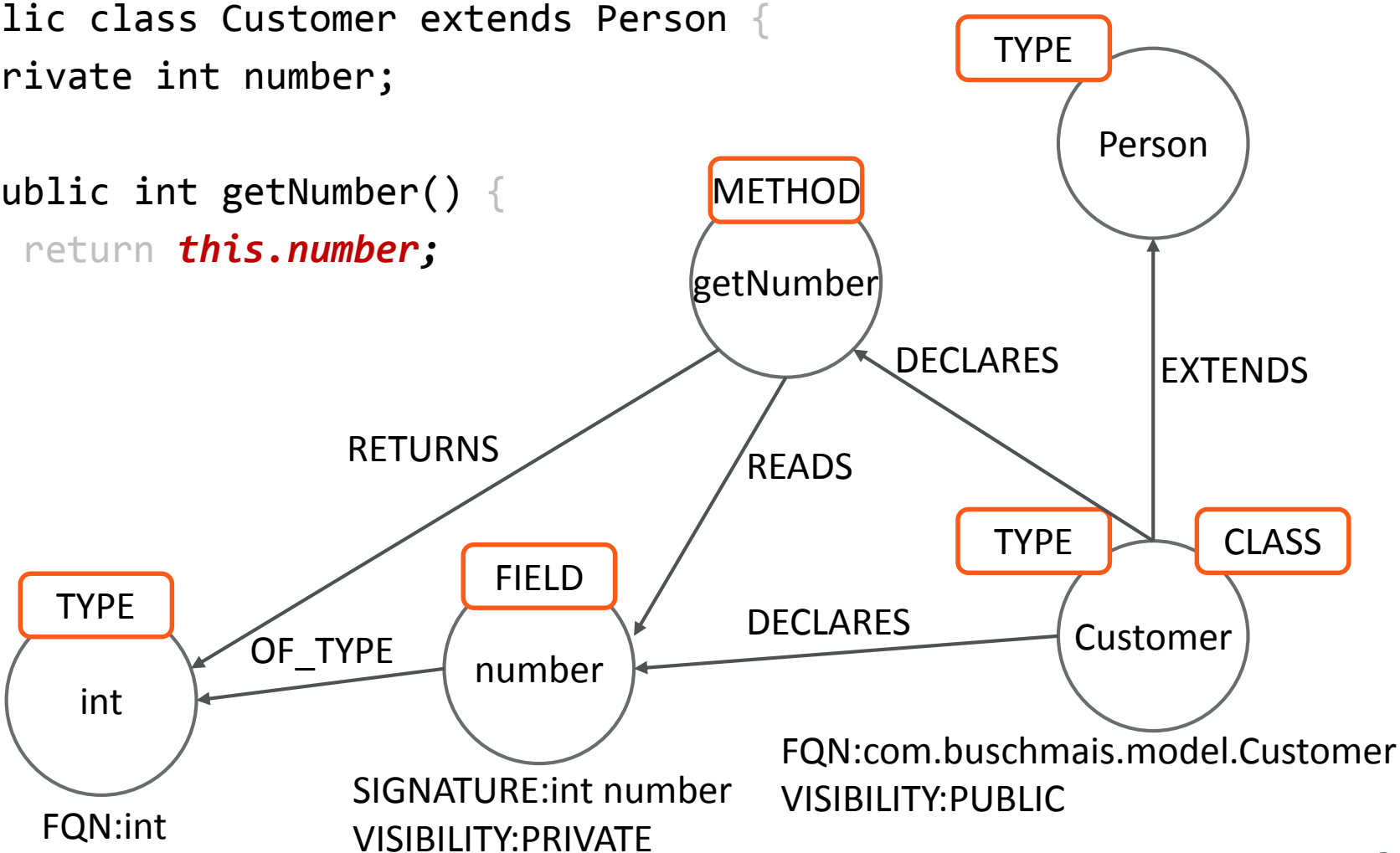
```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

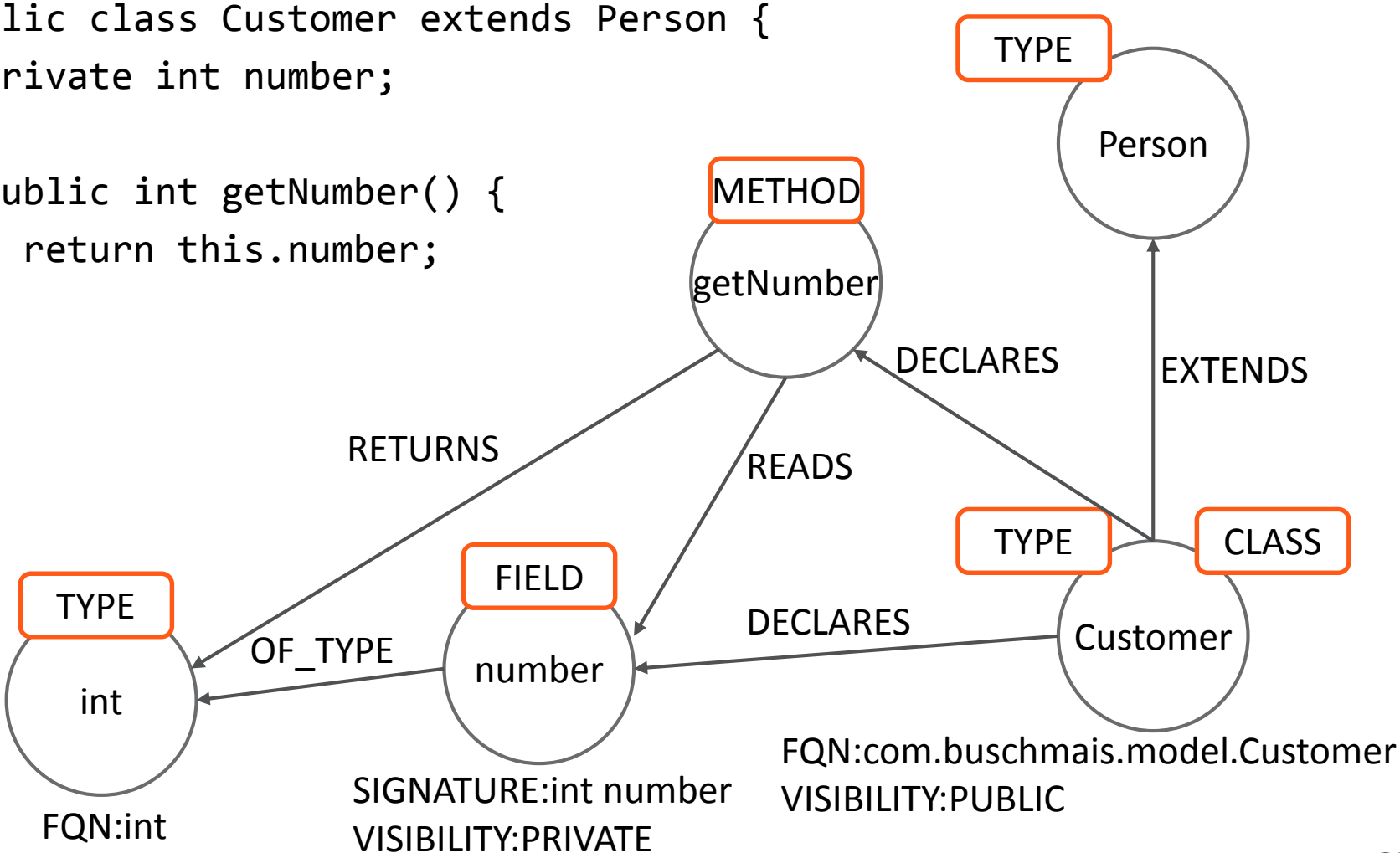
```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



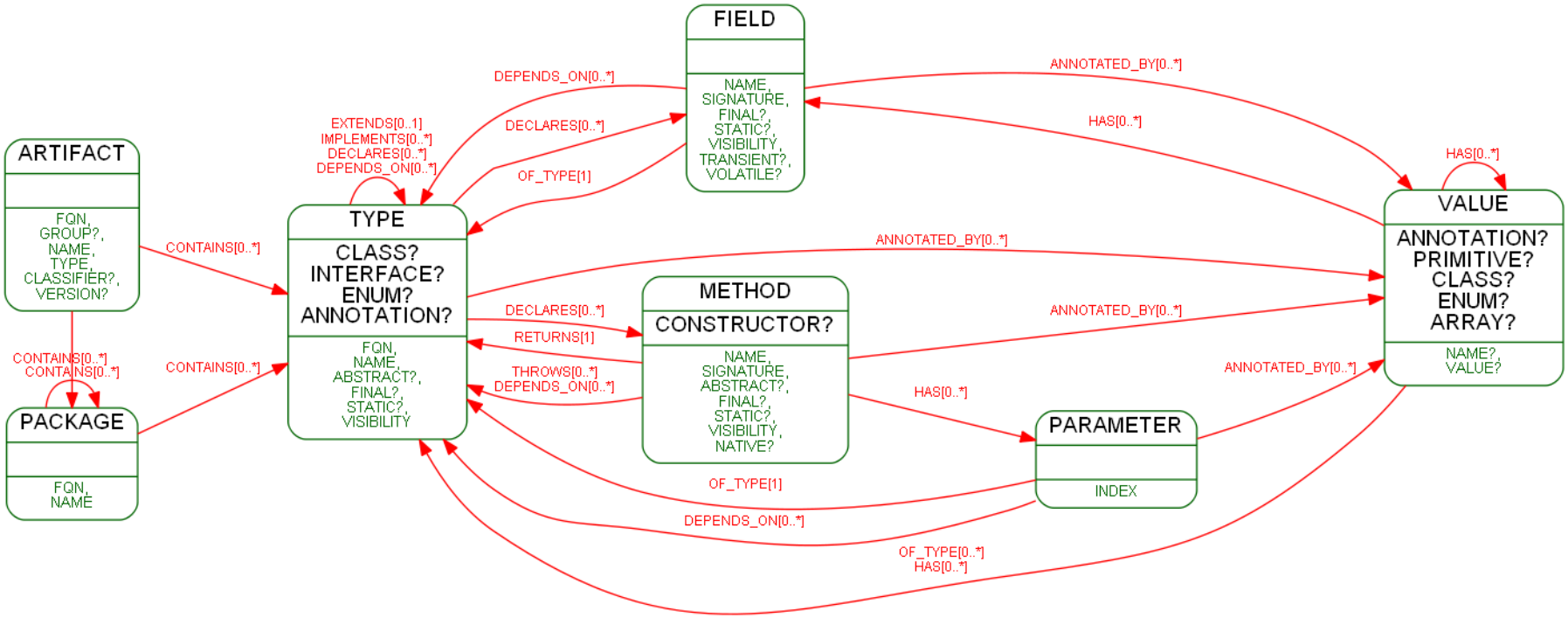
Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

- Elements of the „Java software graph model“
  - Nodes and their labels (but without properties)
    - ARTIFACT
    - PACKAGE
    - TYPE, CLASS, INTERFACE, ANNOTATION, ENUM
    - METHOD, CONSTRUCTOR, PARAMETER
    - FIELD
    - VALUE, CLASS, ANNOTATION, ENUM, PRIMITIVE, ARRAY
  - Relationships
    - CONTAINS, DECLARES
    - EXTENDS, IMPLEMENTS
    - RETURNS, THROWS, INVOKES, HAS
    - ANNOTATED_BY, OF_TYPE

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation



Modeling Software Structures As A Graph

- The model is stored as at has been modeled!

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

- The model is stored as at has been modeled!
- Embedded API of Neo4j:

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

- The model is stored as at has been modeled!
- Embedded API of Neo4j:

```
Node node = graphDatabaseService.createNode();
```

Modeling Software Structures As A Graph

- The model is stored as at has been modeled!
- Embedded API of Neo4j:

```
Node node = graphDatabaseService.createNode();
node.addLabel(MyLabels.TYPE);
```

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

- The model is stored as at has been modeled!
- Embedded API of Neo4j:

```
Node node = graphDatabaseService.createNode();
node.addLabel(MyLabels.TYPE);
node.setProperty("SIGNATURE", "int number")
```

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

- The model is stored as at has been modeled!
- Embedded API of Neo4j:

```
Node node = graphDatabaseService.createNode();
node.addLabel(MyLabels.TYPE);
node.setProperty("SIGNATURE", "int number")
node.createRelationshipTo(otherNode,
MyRelations.OF_TYPE)
```

Modeling Software Structures As A Graph

buschmais
Beratung . Technologie . Innovation

- The model is stored as at has been modeled!
- Embedded API of Neo4j:

```
Node node = graphDatabaseService.createNode();
node.addLabel(MyLabels.TYPE);
node.setProperty("SIGNATURE", "int number")
node.createRelationshipTo(otherNode,
MyRelations.OF_TYPE)
```

- All operations (and even more!) also possible via
  - Cypher
  - REST

Exploration And Verification Of Java Applications

Using A Graph Database

Exploring An Application Using Queries

Exploring An Application Using Queries

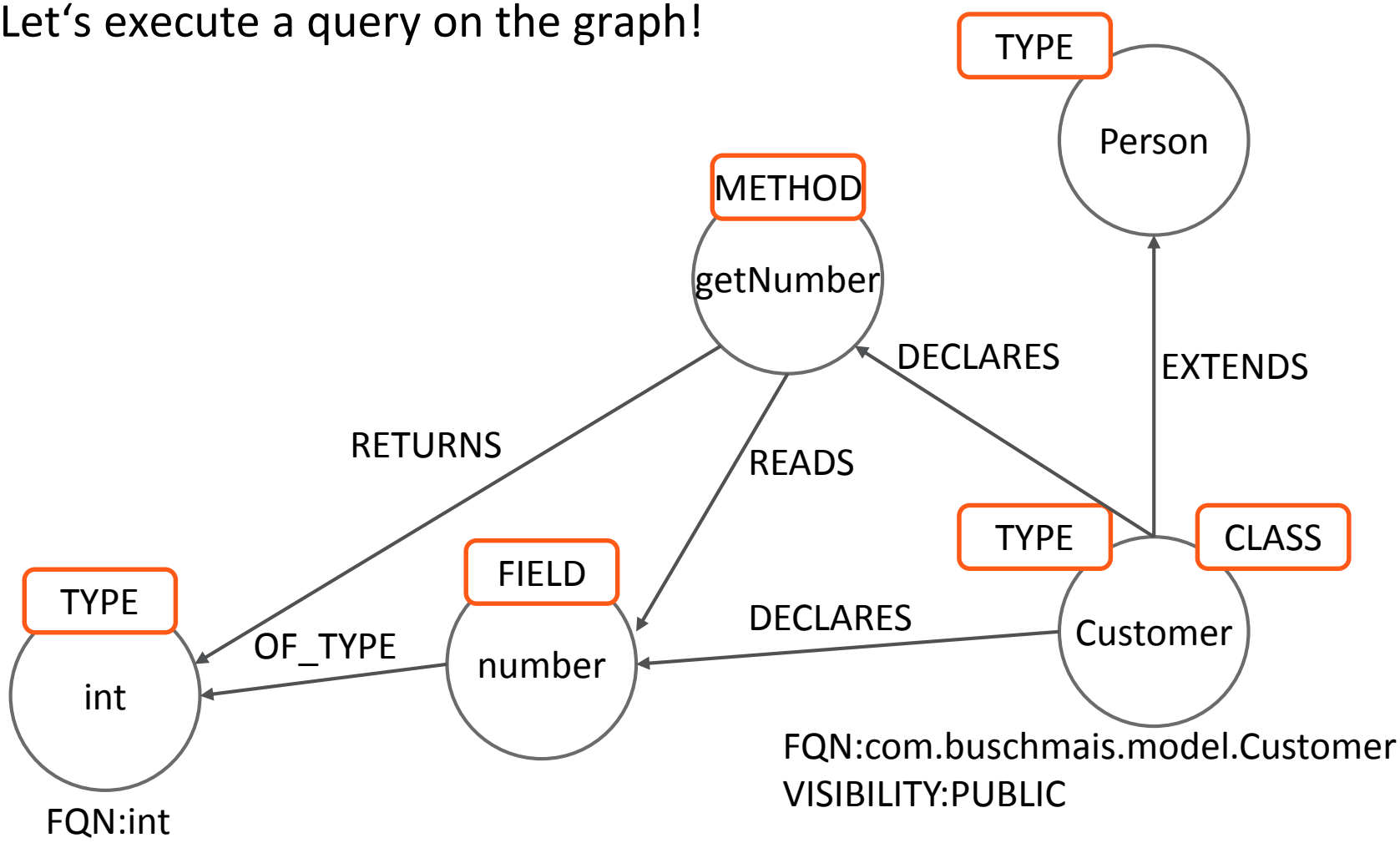
buschmais
Beratung . Technologie . Innovation

- Let‘s execute a query on the graph!

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

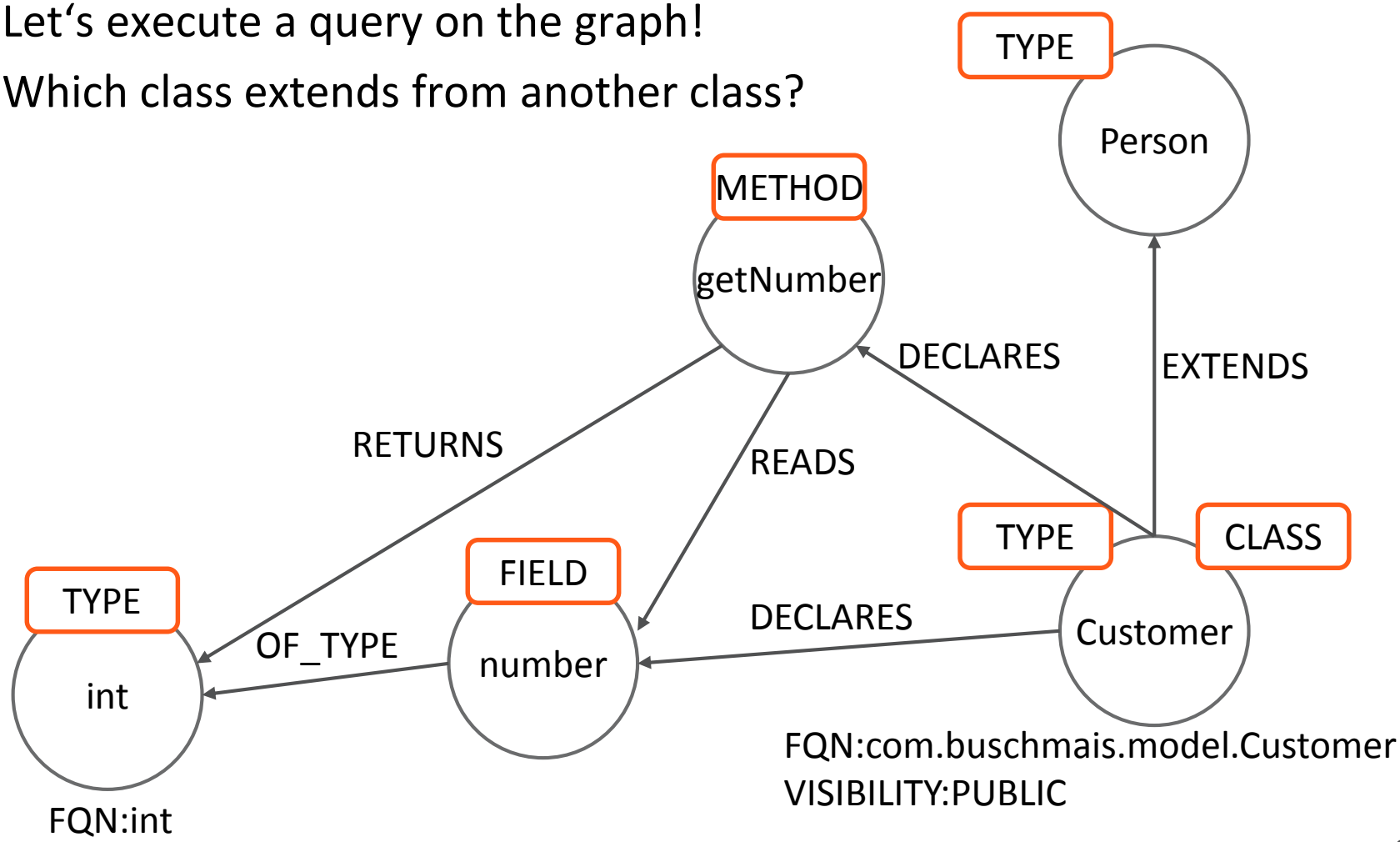
- Let‘s execute a query on the graph!



Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

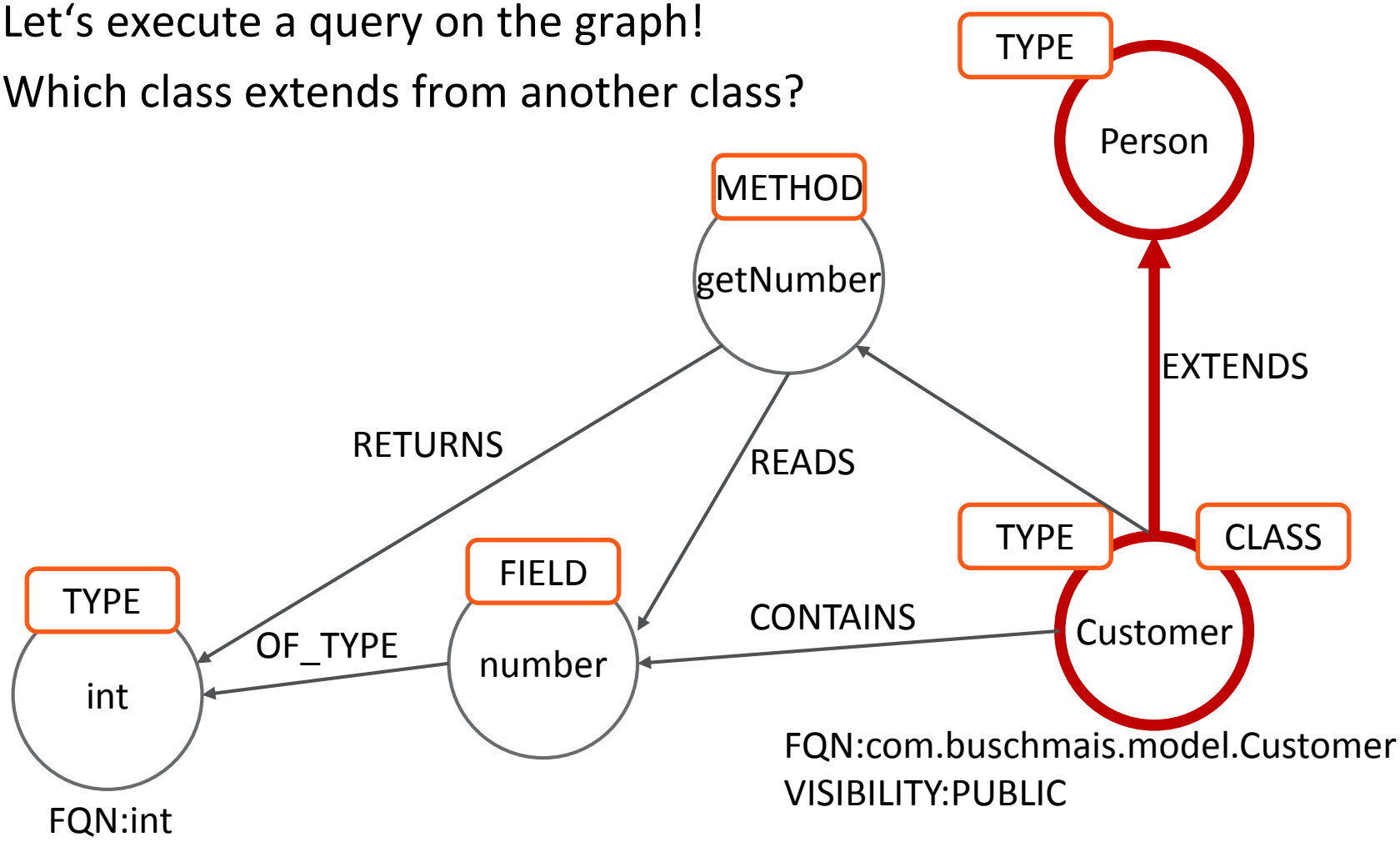
- Let‘s execute a query on the graph!
- Which class extends from another class?



Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

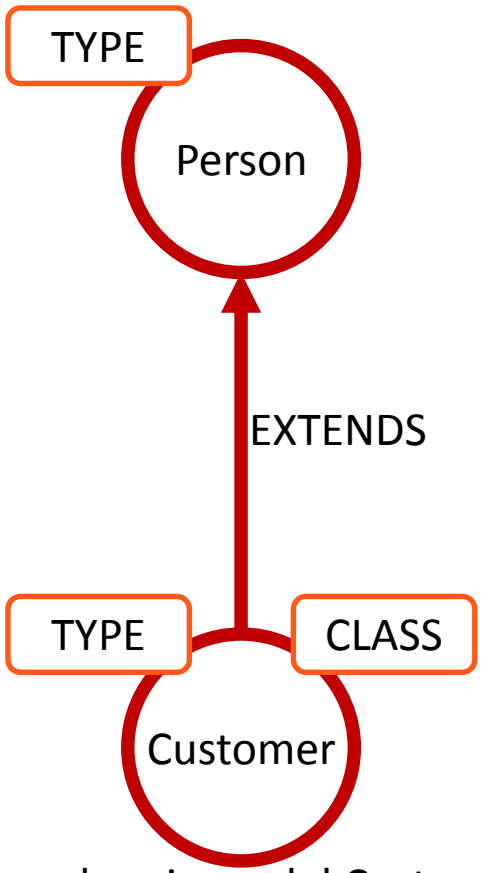
- Let‘s execute a query on the graph!
- Which class extends from another class?



Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Let‘s execute a query on the graph!
- Which class extends from another class?
- How can we express a query on this pattern?



Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Let‘s convert this to ASCII art…

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Let‘s convert this to ASCII art…
  - () as nodes

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Let‘s convert this to ASCII art…
  - () as nodes
  - -[]-> as directed relationships

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Let‘s convert this to ASCII art…
  - () as nodes
  - -[]-> as directed relationships

()-[]->()

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Let‘s convert this to ASCII art…
  - () as nodes
  - -[]-> as directed relationships

(c1)-[]->(c2)

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Let‘s convert this to ASCII art…
  - () as nodes
  - -[]-> as directed relationships

(c1)-[:EXTENDS]->(c2)

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Let‘s convert this to ASCII art…
  - () as nodes
  - -[]-> as directed relationships

(c1:CLASS)-[:EXTENDS]->(c2:TYPE)

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Pattern matching is the core principle of Cypher!

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Pattern matching is the core principle of Cypher!

```
MATCH
  (c1:CLASS)-[:EXTENDS]->(c2:TYPE)
RETURN
  c1.FQN, c2.FQN
```

Exploring An Application Using Queries

- Which classes contain the highest number of methods?

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Which classes contain the highest number of methods?

```
MATCH
  (class:CLASS)-[:DECLARES]->(method:METHOD)
```

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Which classes contain the highest number of methods?

```
MATCH
  (class:CLASS)-[:DECLARES]->(method:METHOD)
RETURN
  class.FQN, count(method) as Methods
```

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Which classes contain the highest number of methods?

```
MATCH
  (class:CLASS)-[:DECLARES]->(method:METHOD)
RETURN
  class.FQN, count(method) as Methods
ORDER BY
  Methods DESC
```

- Which classes contain the highest number of methods?

```
MATCH
  (class:CLASS)-[:DECLARES]->(method:METHOD)
RETURN
  class.FQN, count(method) as Methods
ORDER BY
  Methods DESC
LIMIT 20
```

Exploring An Application Using Queries

- Which class has the deepest inheritance hierarchy?

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Which class has the deepest inheritance hierarchy?

```
MATCH
   h=(class:CLASS)-[:EXTENDS*]->(super:CLASS)
```

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Which class has the deepest inheritance hierarchy?

```
MATCH
  h=(class:CLASS)-[:EXTENDS*]->(super:TYPE)
RETURN
  class.FQN, length(h) as Depth
```

- Which class has the deepest inheritance hierarchy?

```
MATCH
  h=(class:CLASS)-[:EXTENDS*]->(super:TYPE)
RETURN
  class.FQN, length(h) as Depth
ORDER BY
  Depth desc
```

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Which class has the deepest inheritance hierarchy?

```
MATCH
  h=(class:CLASS)-[:EXTENDS*]->(super:TYPE)
RETURN
  class.FQN, length(h) as Depth
ORDER BY
  Depth desc
LIMIT 20
```

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Queries on graph structures allow…

Exploring An Application Using Queries

- Queries on graph structures allow…
  - Calculation of metrics, e.g.
    - Classes per package, fields/methods per class
    - Depth of inheritance hierarchies
    - Fan in/out of artifacts, packages, classes

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Queries on graph structures allow…
  - Calculation of metrics, e.g.
    - Classes per package, fields/methods per class
    - Depth of inheritance hierarchies
    - Fan in/out of artifacts, packages, classes
  - Impact-Analysis, e.g.
    - Which methods/classes/packages/artifacts are potentially affected by changes on an element?

Exploring An Application Using Queries

buschmais
Beratung . Technologie . Innovation

- Queries on graph structures allow…
  - Calculation of metrics, e.g.
    - Classes per package, fields/methods per class
    - Depth of inheritance hierarchies
    - Fan in/out of artifacts, packages, classes
  - Impact-Analysis, e.g.
    - Which methods/classes/packages/artifacts are potentially affected by changes on a class, method or field?
  - Validation of constraints and conventions, e.g.
    - Naming rules
    - Cyclic dependencies (types, packages)
    - Internal and external dependencies
      - Modules
      - Frameworks and libraries

Exploration And Verification Of Java Applications
Using A Graph Database

Live Demo #1
Metrics And Exploration

Exploration And Verification Of Java Applications

Using A Graph Database

Software, Rules and Erosion

- At the beginning of a new project…
  - Draft of the application architecture
  - Definition of conventions and constraints
    - Modules, layers, internal and external dependencies
    - Naming rules
  - Initial setup of the project structure

- At the beginning of a new project…
  - Draft of the application architecture
  - Definition of conventions and constraints
    - Modules, layers, internal and external dependencies
    - Naming rules
  - Initial setup of the project structure

- Goals
  - Breaking down complexity of problems
  - „Accessibility“ for developers
    - Similar structures and approaches for similar problems

Structures, Rules and Erosion

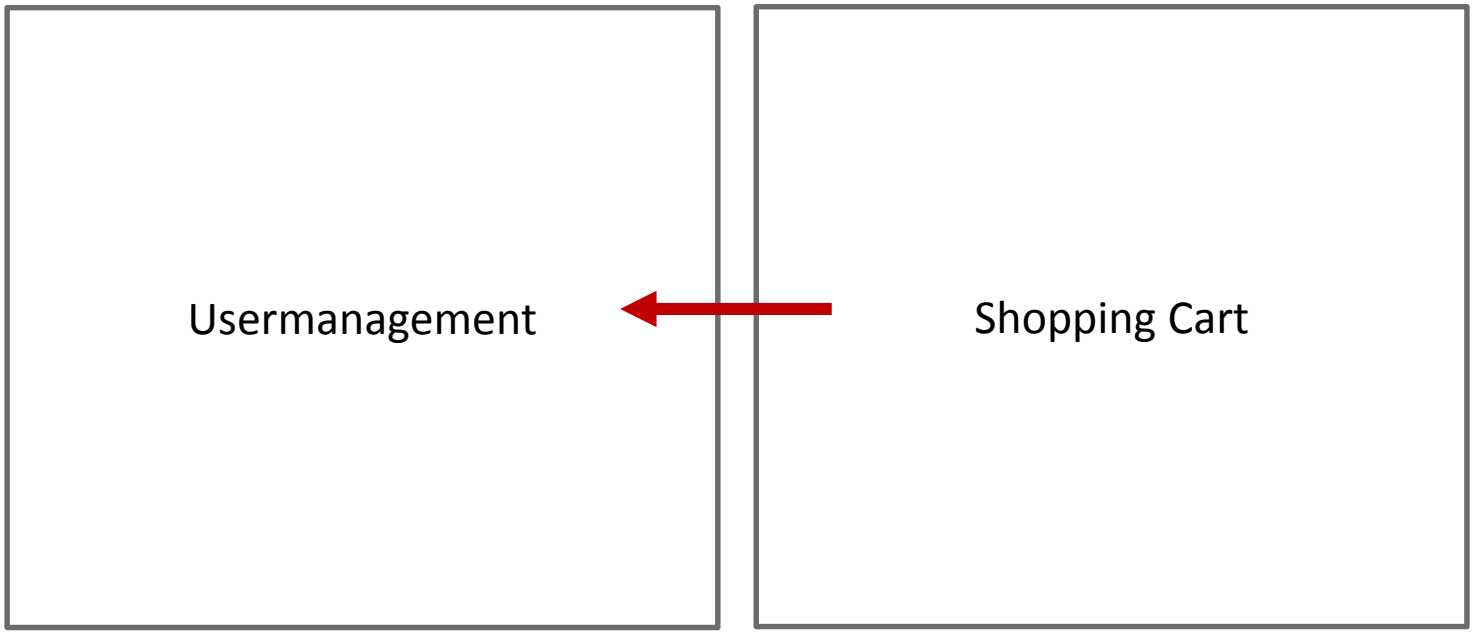
buschmais
Beratung . Technologie . Innovation

- Sketch of an architecture

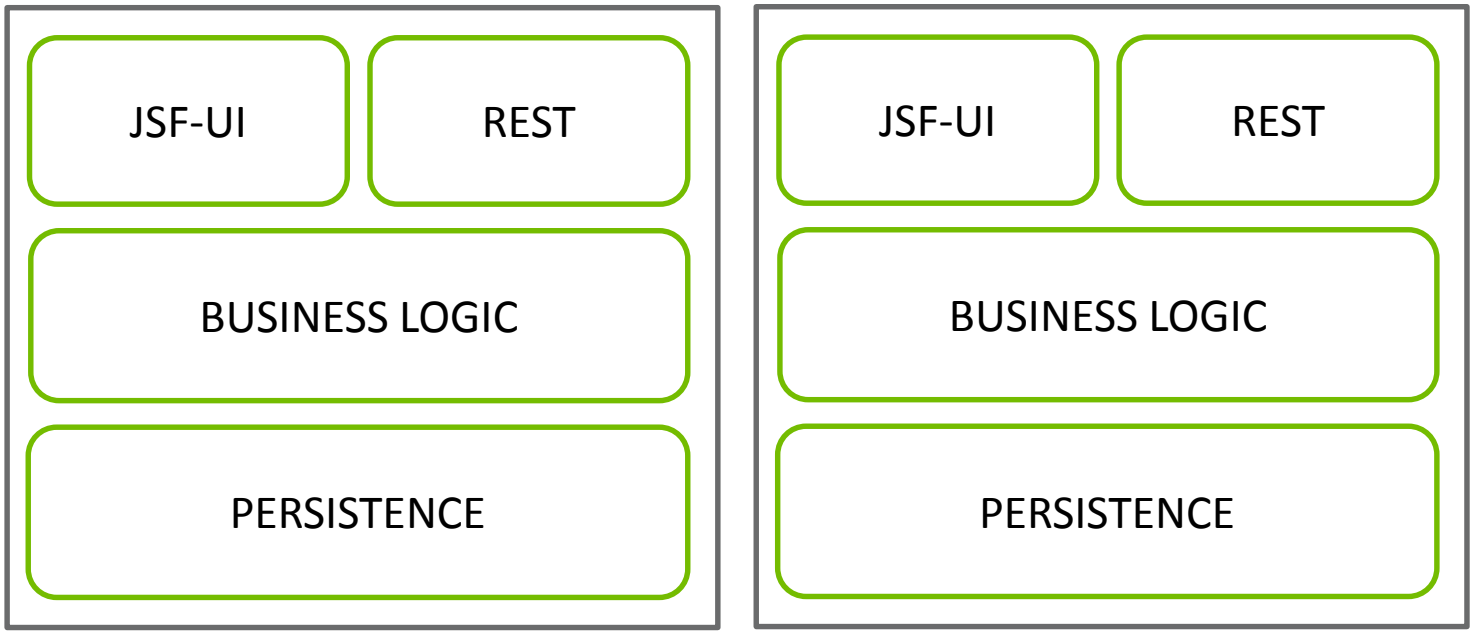
- Sketch of an architecture

- Sketch of an architecture
  - Business modules

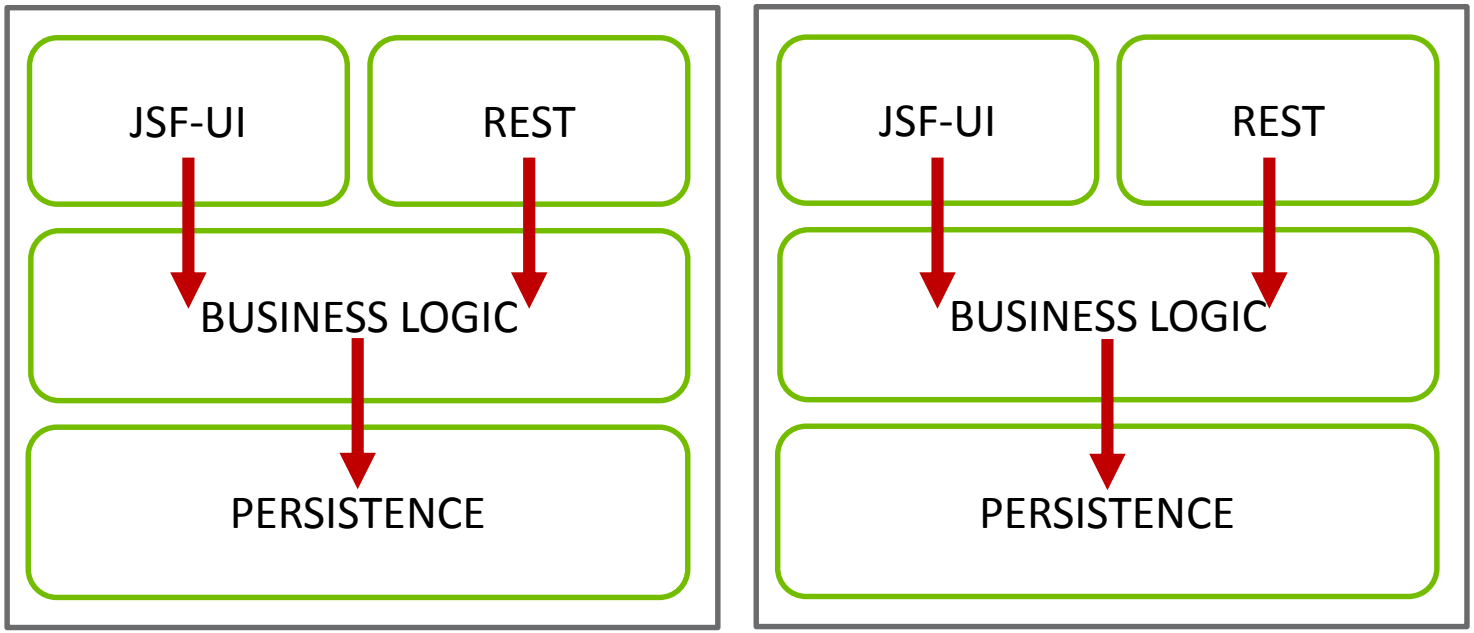
- Sketch of an architecture
  - Defined dependencies between business modules



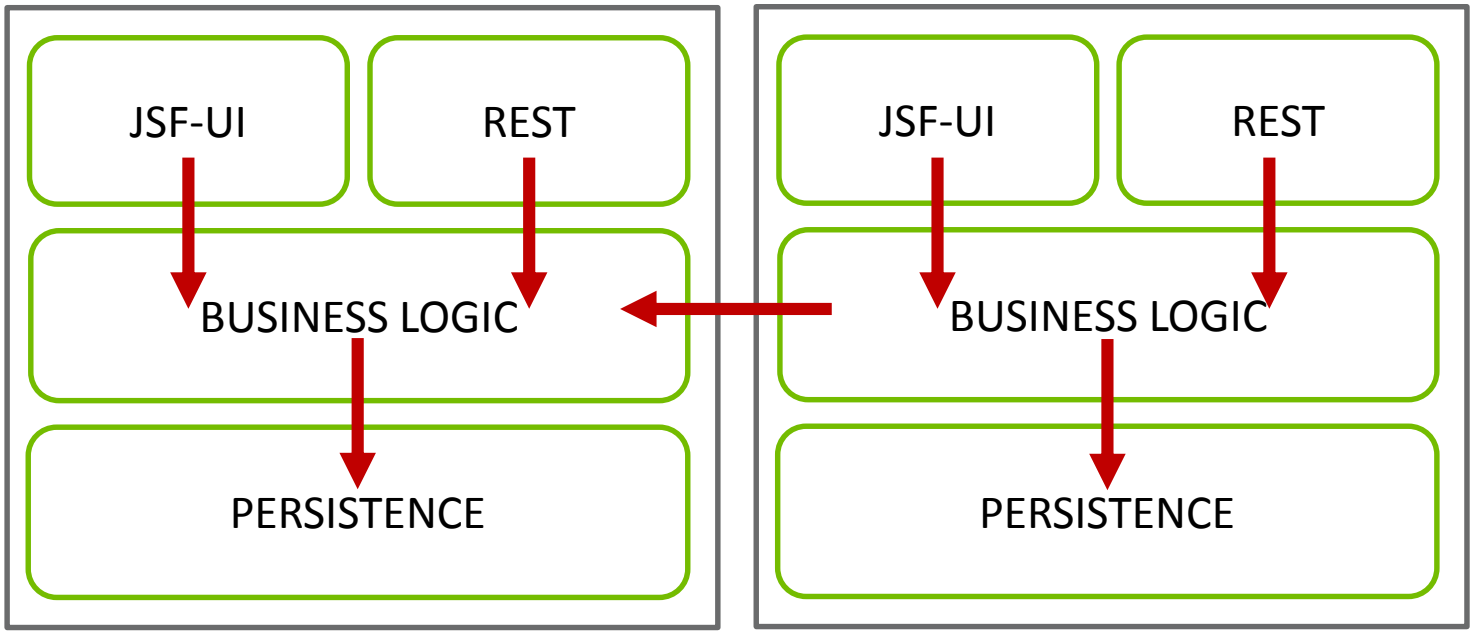
- Sketch of an architecture
  - Technical layering



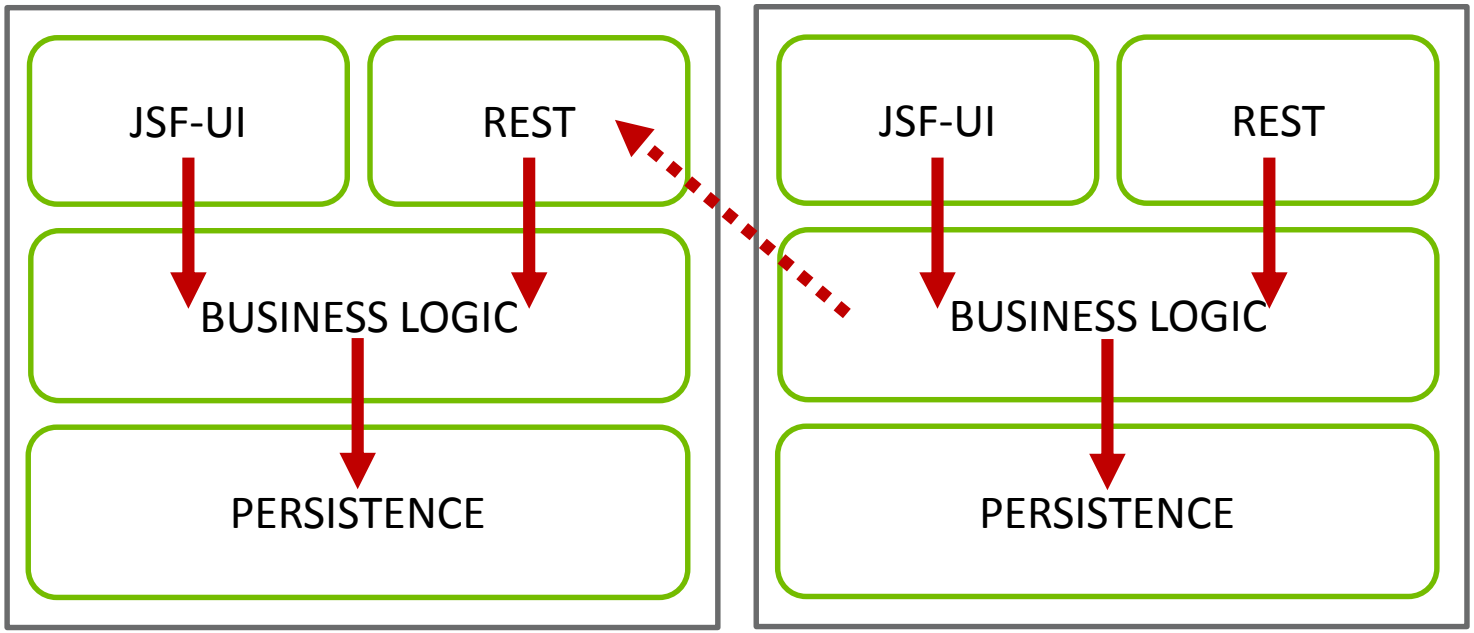
- Sketch of an architecture
  - Defined dependencies between technical layers



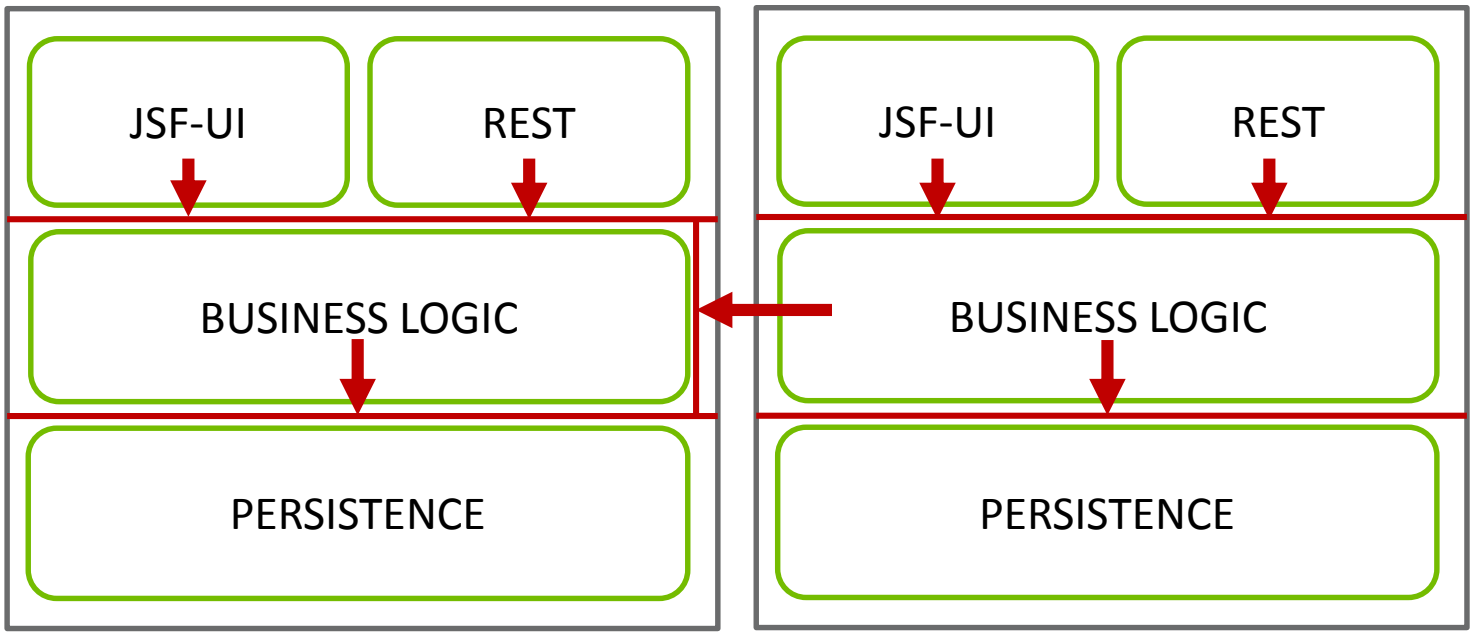
- Sketch of an architecture
  - Defined dependencies of business modules & technical layers



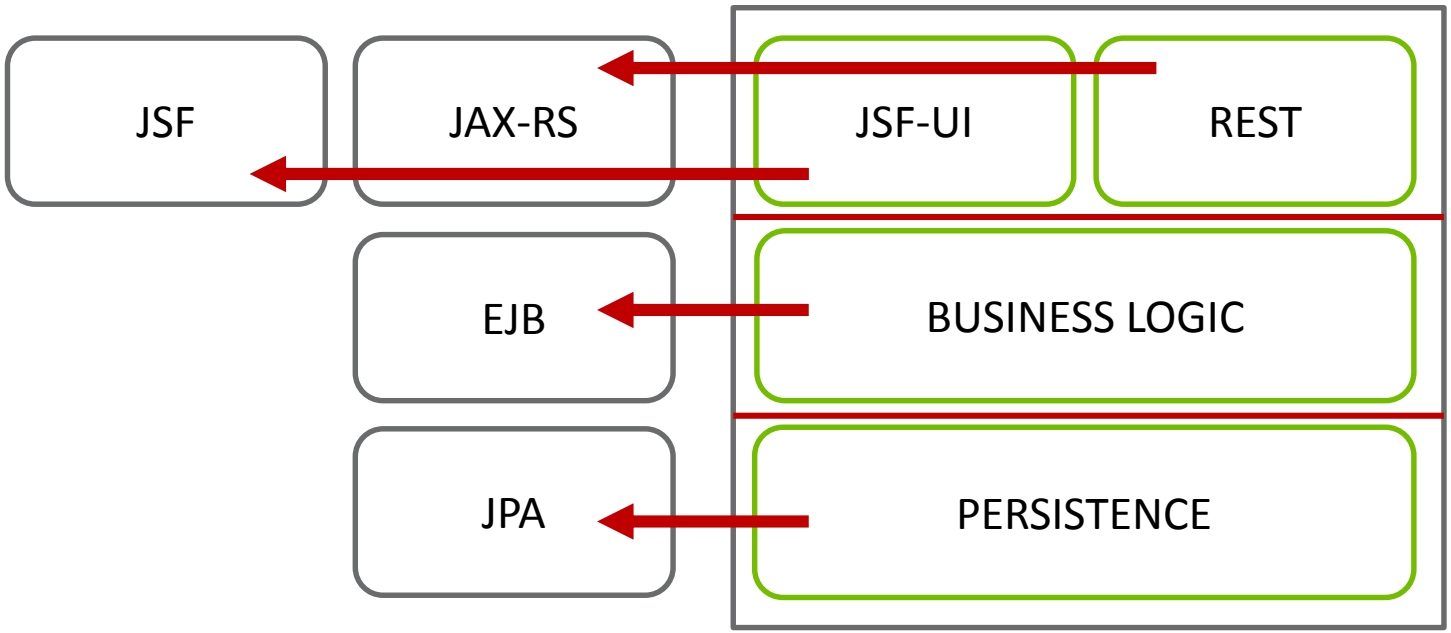
- Sketch of an architecture
  - Defined dependencies of business modules & technical layers



- Sketch of an architecture
  - Decoupling of technical layers (APIs, Interfaces)



- Sketch of an architecture
  - Limitation of the visibility of external dependencies per layer

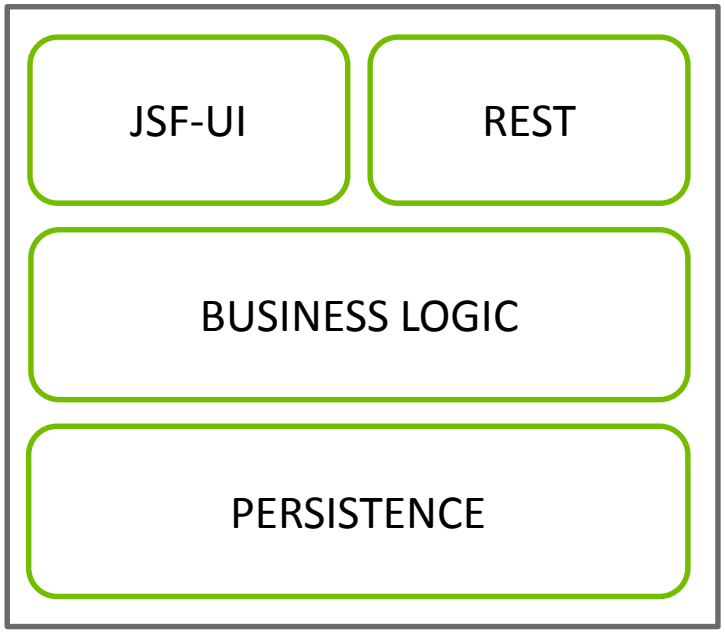


- Translation of architecture rules into the project structure
  - Java language element: Package

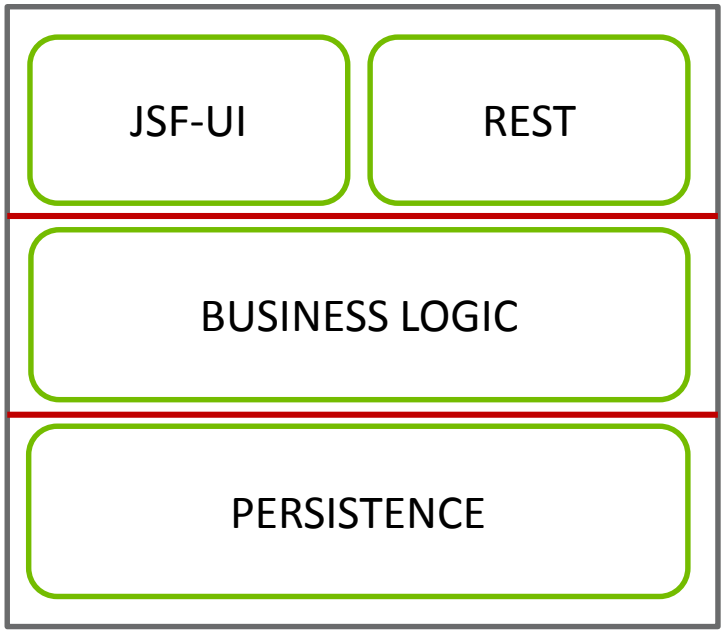
- Translation of architecture rules into the project structure
  - Java language element: Package

  - Definition of business modules on „top level“

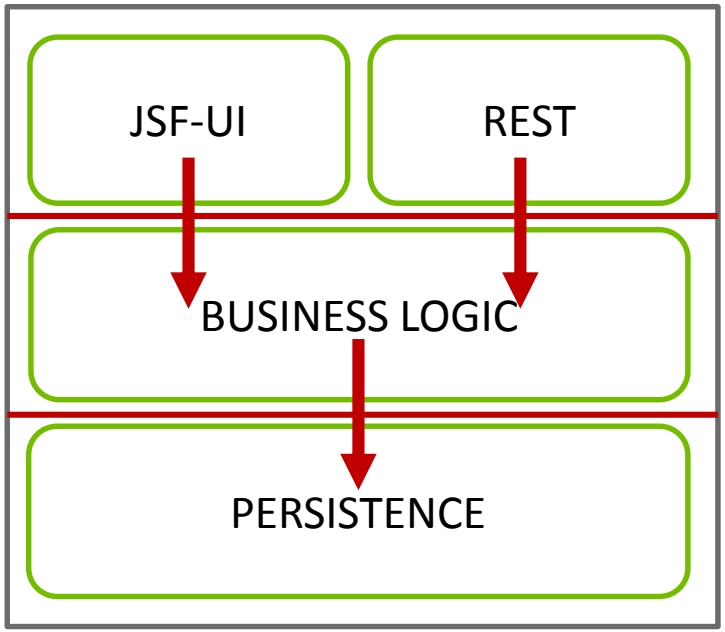
- Translation of architecture rules into the project structure
  - Java language element: Package
  - Technical layers
    - …shop.cart.ui
    - …shop.cart.rest
    - …shop.cart.logic
    - …shop.cart.persistence



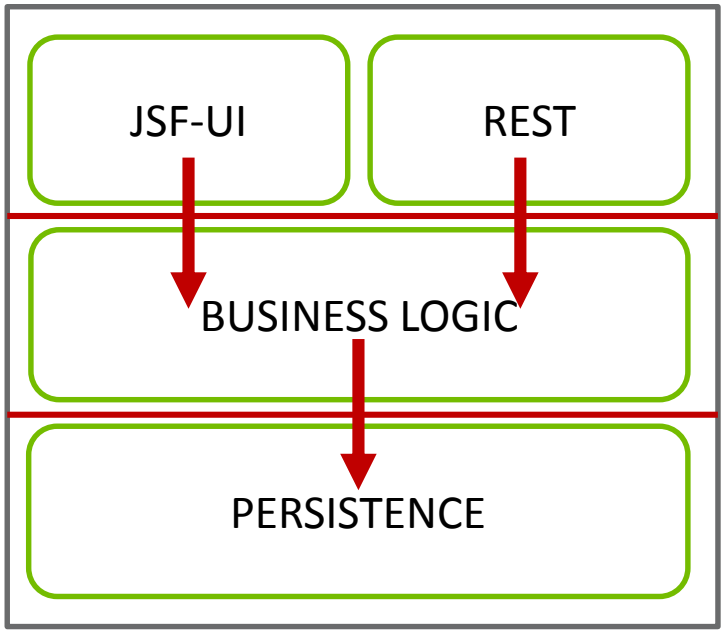
- Translation of architecture rules into the project structure
  - Java language element: Package
  - Technical layers
    - …shop.cart.ui
    - …shop.cart.rest
    - …shop.cart.logic.api
    - …shop.cart.logic.impl
    - …shop.cart.persistence.api
    - …shop.cart.persistence.impl



- Translation of architecture rules into the project structure
  - Definition/restriction of allowed dependencies?
  - Not (yet) supported by Java
  - Solution: using dependency managment of the build system, e.g. Maven



- Translation of architecture rules into the project structure
  - Definition of dependencies using the build system…
    - Lots of small modules
    - Lack of control (changes of build descriptors)
    - Unwanted transitive dependencies
    - No feedback to the developers, i.e. „You can‘t do that because…“



- Conventions, e.g. definition of naming rules, e.g.
  - Package names
    - api, spi, impl
  - Suffixes for JPA elements
    - *Entity, *Key
  - Suffixes for EJBs
    - *Bean, *MDB
  - Location of classes/packages
    - JPA entities in .model packages (persistence layer)
    - JSF controller in .controller packages (UI layer)
- But how does all this work in practice?

- „Erosion“ starts at the first day of development!
  - Constantly increasing number of rule violations
  - Even with a one-man developer team…

Structures, Rules and Erosion

buschmais
Beratung . Technologie . Innovation

Structures, Rules and Erosion

buschmais
Beratung . Technologie . Innovation

- Erosion: some causes
  - Fast growing complexity of applications
  - Increasing amount and complexity of rules
  - Project documentation is never up-to-date
  - Different skill level of developers
  - Different types of developers
  - Time pressure („Hacks“)
  - Broken windows…

Exploration And Verification Of Java Applications

Using A Graph Database

Verification Of Conventions And Constraints

Validation Of Conventions And Constraints

buschmais
Beratung . Technologie . Innovation

- Approach consisting of 3 steps

Validation Of Conventions And Constraints

- Approach consisting of 3 steps
  - 1. Scan
    - Parsing of the application and storing as raw data in a database

Validation Of Conventions And Constraints

- Approach consisting of 3 steps
  - 1. Scan
    - Parsing of the application and storing as raw data in a database
  - 2. Enhancement of raw data by CONCEPT queries
    - Labeling of nodes →Assignment of roles to Java elements
      - Architectural concepts (e.g. modules, layers)
      - Design concepts (e.g. API vs. implementation)
      - Technical concepts (JPA Entities, EJBs, tests)
    - Adding relationships
      - class, package and module dependencies

Validation Of Conventions And Constraints

- Approach consisting of 3 steps
  - 1. Scan
    - Parsing of the application and storing as raw data in a database
  - 2. Enhancement of raw data by CONCEPT queries
    - Labeling of nodes →Assignment of roles to Java elements
      - Architectural concepts (e.g. modules, layers)
      - Design concepts (e.g. API vs. implementation)
      - Technical concepts (JPA Entities, EJBs, tests)
    - Adding relationships
      - class, package and module dependencies
  - 3. Execution of CONSTRAINT queries
    - Queries to detect rule violation
      - API classes which depend on implementation classes
      - Message Driven Beans not having the name suffix MDB

Validation Of Conventions And Constraints

- Concept query: Labeling JPA entities

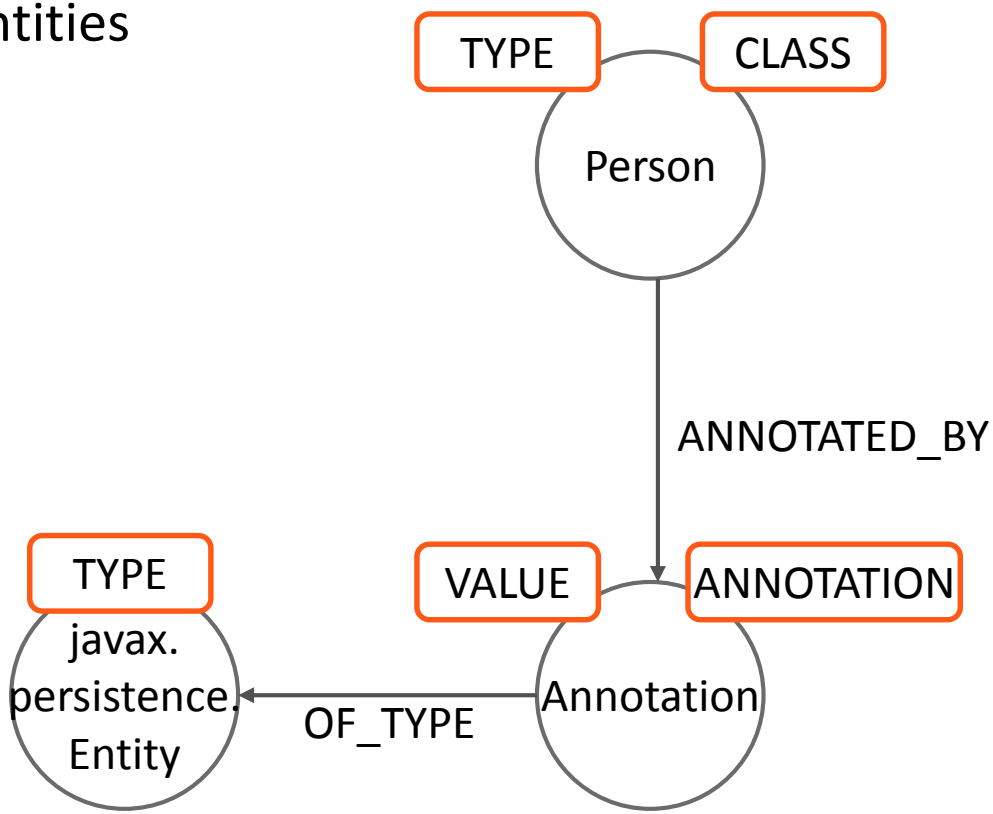
```
@Entity
public class Person { …
```

Validation Of Conventions And Constraints

buschmais
Beratung . Technologie . Innovation

- Concept query: Labeling JPA entities

```
@Entity
public class Person { …
```

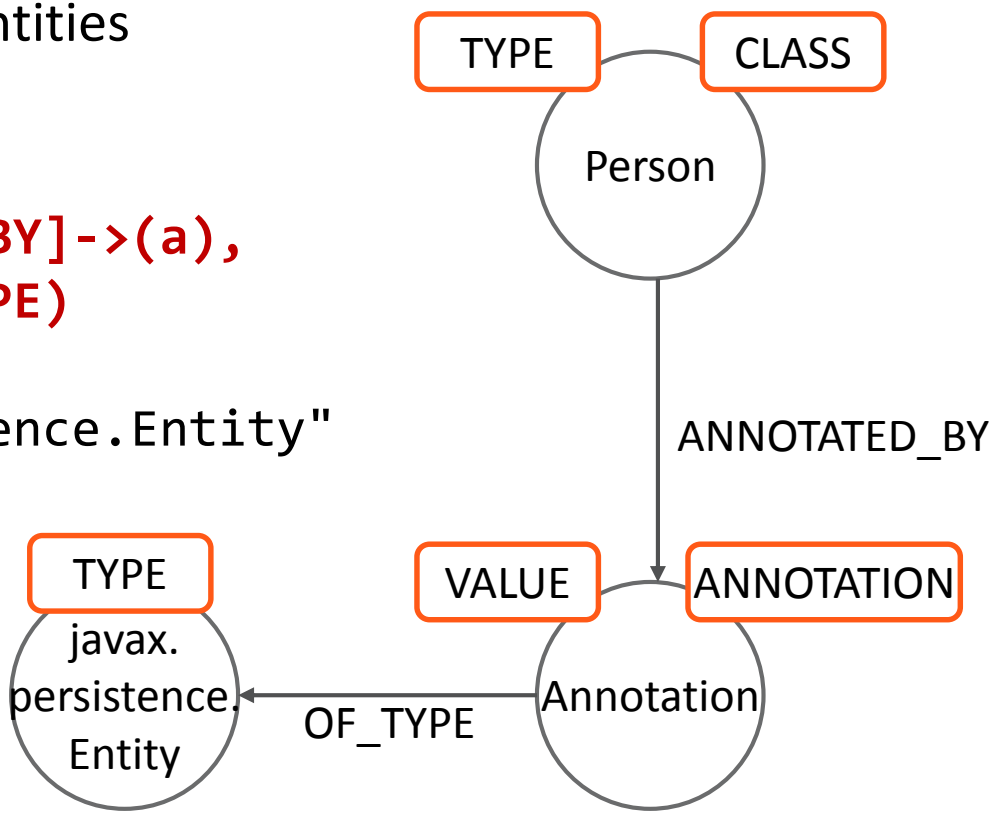


Validation Of Conventions And Constraints

buschmais
Beratung . Technologie . Innovation

- Concept query: Labeling JPA entities

```
MATCH
  (e:CLASS)-[:ANNOTATED_BY]->(a),
  (a)-[:OF_TYPE]->(at:TYPE)
WHERE
  at.FQN ="javax.persistence.Entity"
RETURN
  e.FQN as EntityName
```

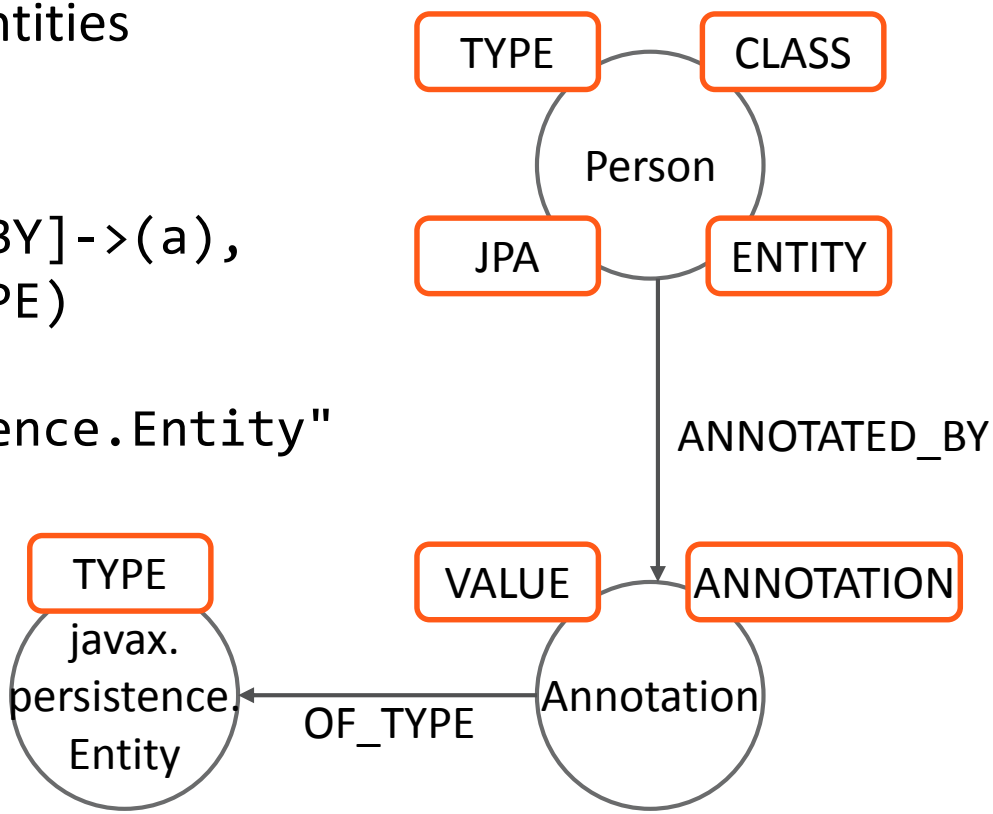


Validation Of Conventions And Constraints

buschmais
Beratung . Technologie . Innovation

- Concept query: Labeling JPA entities

```
MATCH
  (e:CLASS)-[:ANNOTATED_BY]->(a),
  (a)-[:OF_TYPE]->(at:TYPE)
WHERE
  at.FQN ="javax.persistence.Entity"
SET
  e:JPA:ENTITY
RETURN
  e.FQN as EntityName
```

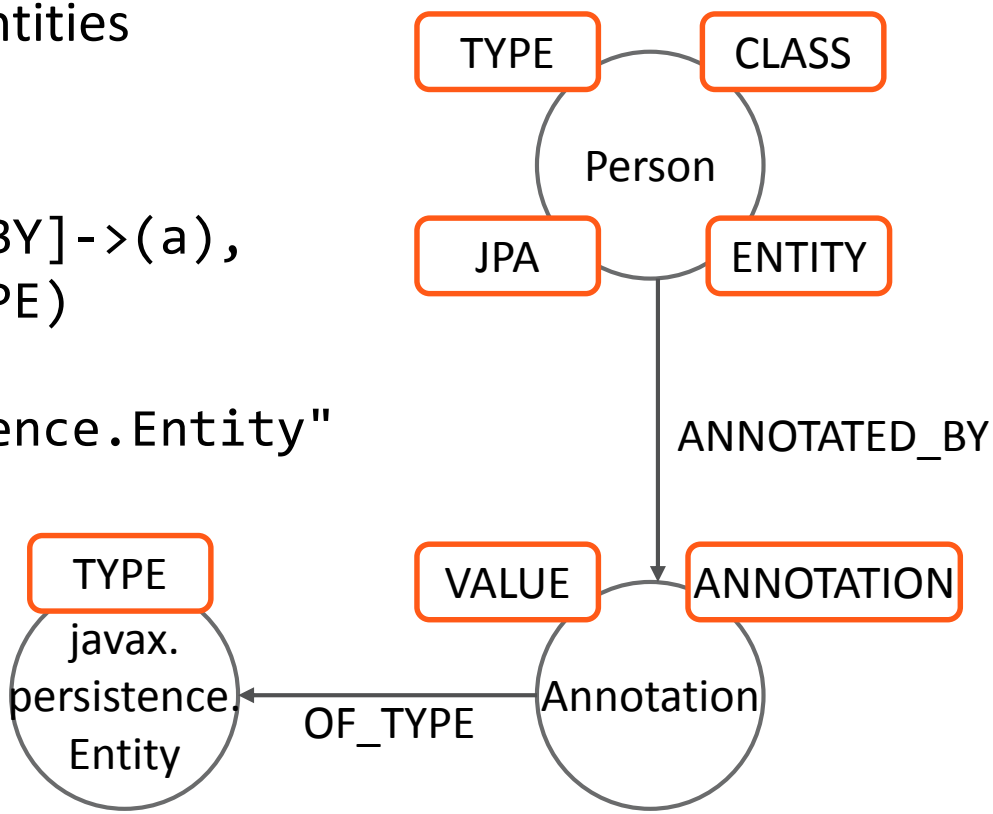


Validation Of Conventions And Constraints

buschmais
Beratung . Technologie . Innovation

- Concept query: Labeling JPA entities

```
MATCH
  (e:CLASS)-[:ANNOTATED_BY]->(a),
  (a)-[:OF_TYPE]->(at:TYPE)
WHERE
  at.FQN ="javax.persistence.Entity"
SET
  e:JPA:ENTITY
RETURN
  e.FQN as EntityName
```



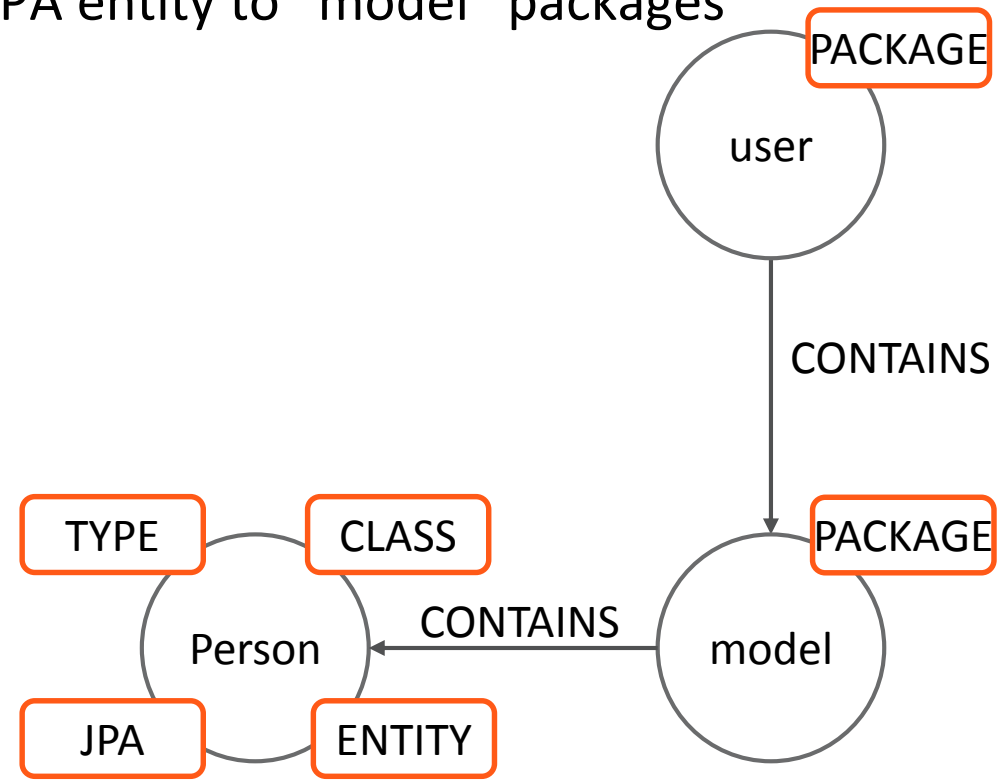
- A concept is **applicable** if its query returns a result

Validation Of Conventions And Constraints

buschmais
Beratung . Technologie . Innovation

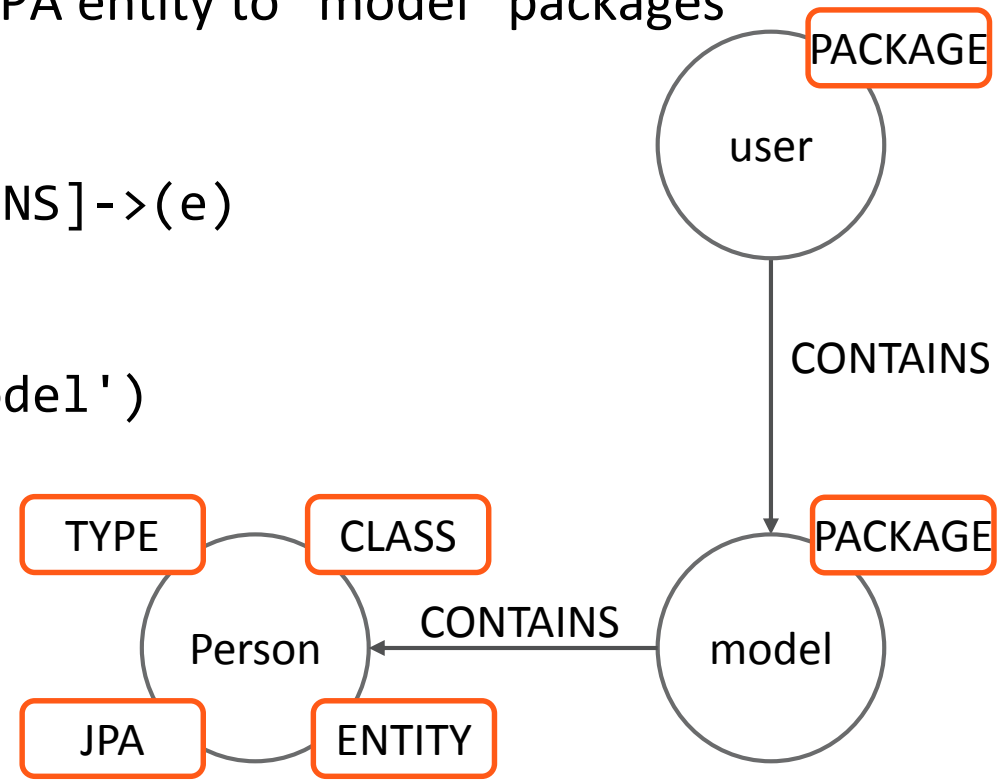
- Constraint query: Restrict JPA entity to “model” packages

- Constraint query: Restrict JPA entity to “model” packages



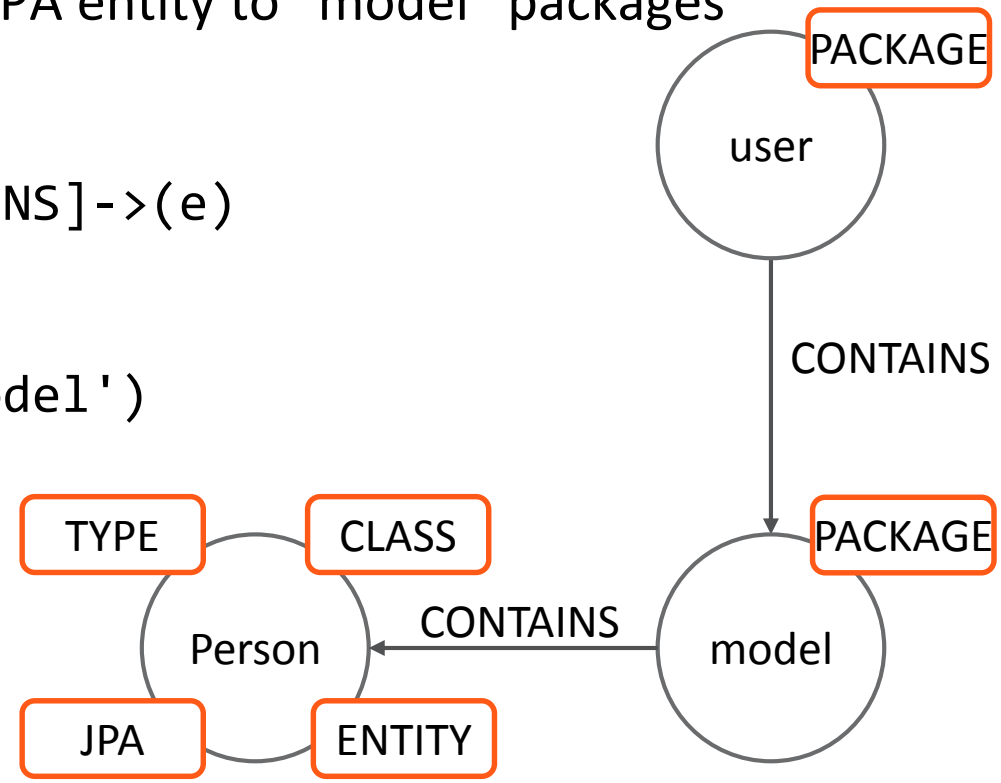
- Constraint query: Restrict JPA entity to “model” packages

```
MATCH
  (p:PACKAGE)-[:CONTAINS]->(e)
WHERE
  e:JPA:ENTITY
  AND NOT(p.NAME = 'model')
RETURN
  e.FQN as EntityName
```



- Constraint query: Restrict JPA entity to “model” packages

```
MATCH
  (p:PACKAGE)-[:CONTAINS]->(e)
WHERE
  e:JPA:ENTITY
  AND NOT(p.NAME = 'model')
RETURN
  e.FQN as EntityName
```

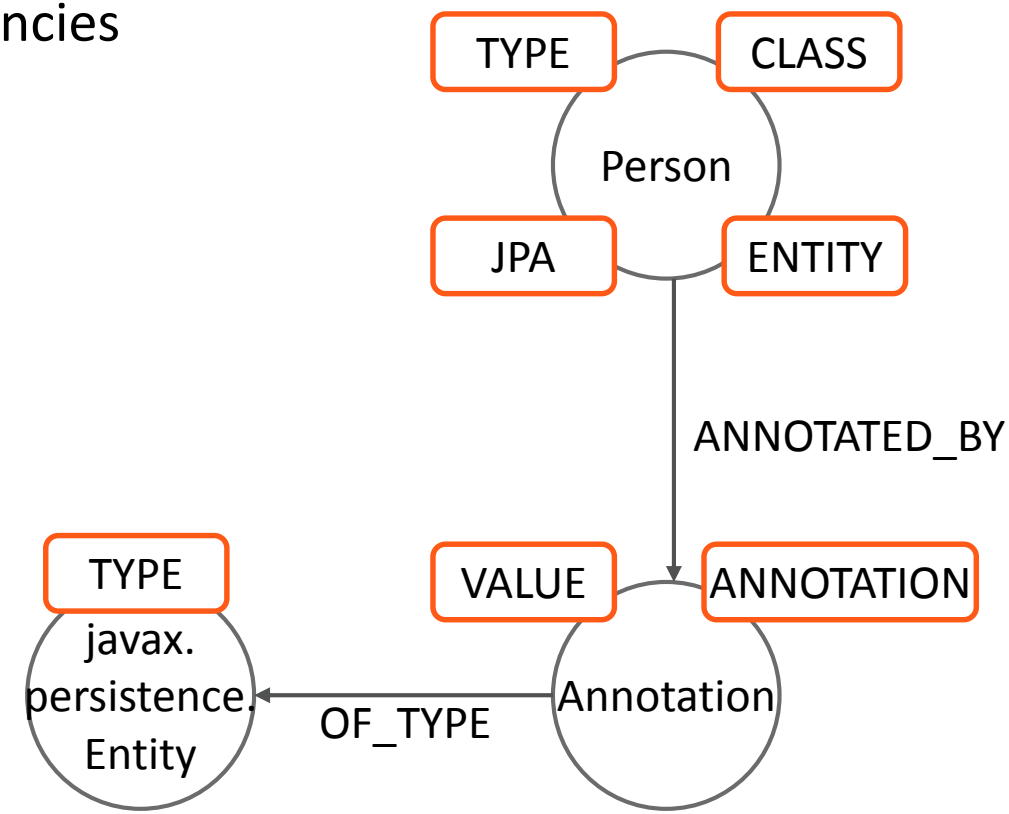


- A constraint is **violated** if its query returns a result

Validation Of Conventions And Constraints

buschmais
Beratung . Technologie . Innovation

- Concept query: Type dependencies

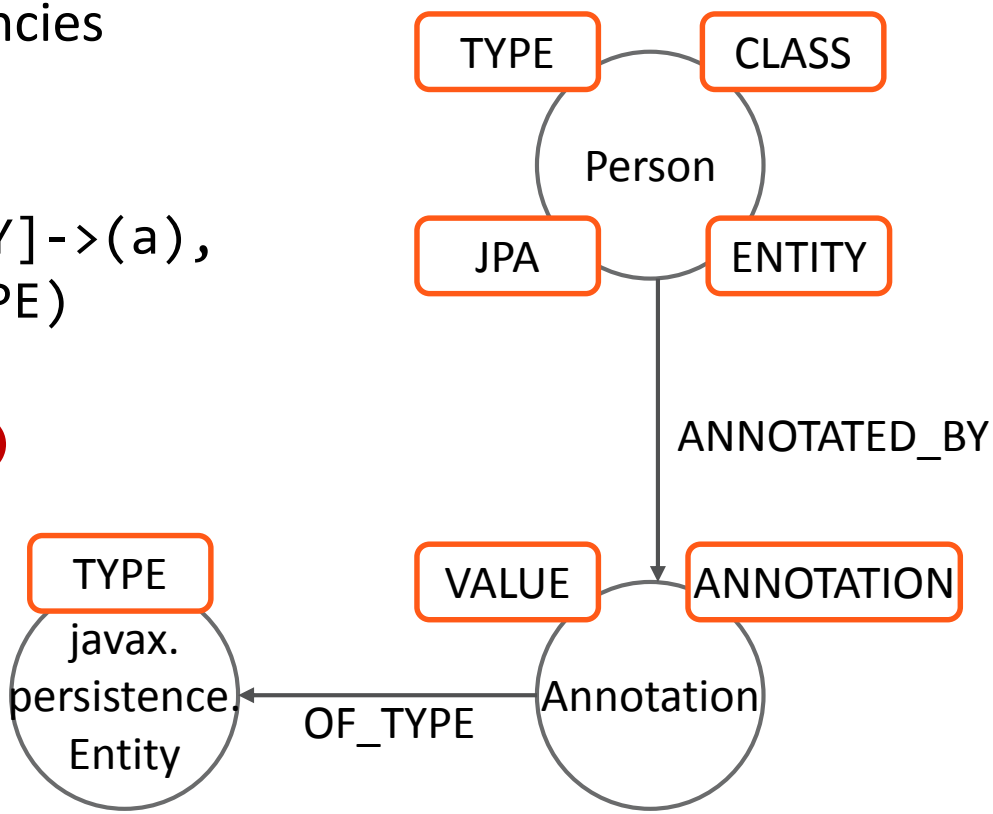


Validation Of Conventions And Constraints

buschmais
Beratung . Technologie . Innovation

- Concept query: Type dependencies

```
MATCH
  (t:TYPE)-[:ANNOTATED_BY]->(a),
  (a)-[:OF_TYPE]->(at:TYPE)
CREATE UNIQUE
  (t)-[:DEPENDS_ON]->(at)
RETURN
  count(t)
  as AnnotatedTypes
```

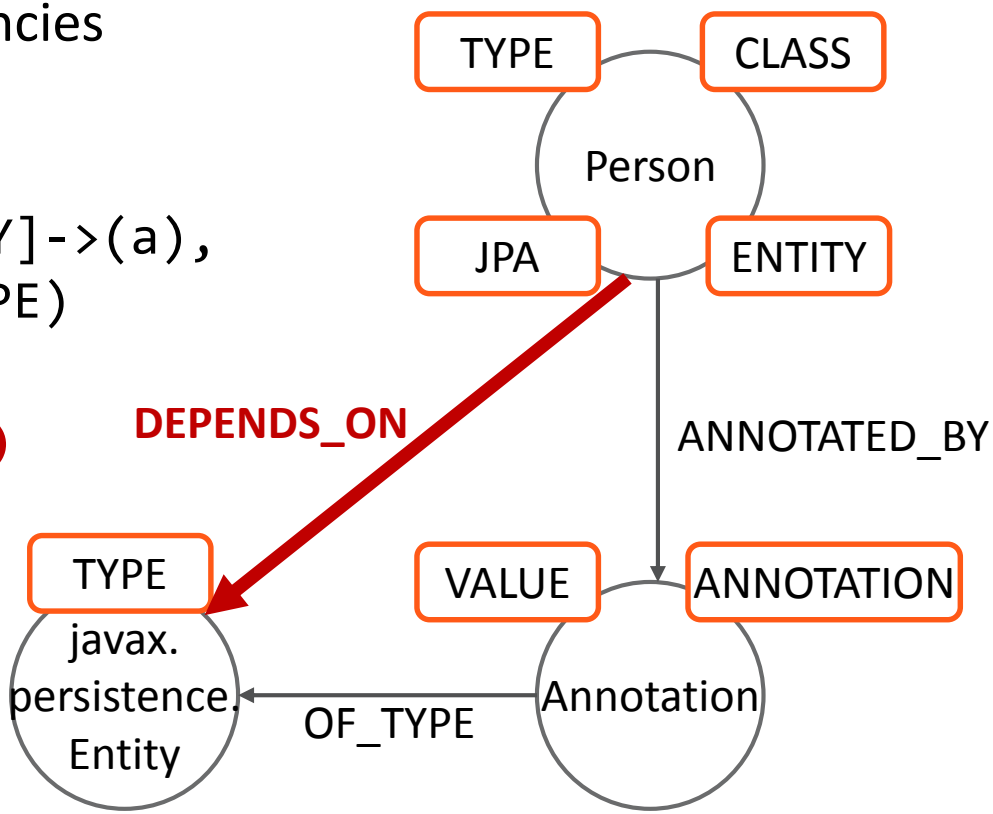


Validation Of Conventions And Constraints

buschmais
Beratung . Technologie . Innovation

- Concept query: Type dependencies

```
MATCH
  (t:TYPE)-[:ANNOTATED_BY]->(a),
  (a)-[:OF_TYPE]->(at:TYPE)
CREATE UNIQUE
  (t)-[:DEPENDS_ON]->(at)
RETURN
  count(t)
  as AnnotatedTypes
```



buschmais

Beratung . Technologie . Innovation

Exploration And Verification Of Java Applications

Using A Graph Database

jQAssistant

- Homepage http://github.com/buschmais/jqassistant
- License: Apache Software License 2.0
- Milestone 1.0.0-M2
- Based on Neo4j (embedded)
- Tool for definition and validation of coding, design and architecture rules.
  - Scan of bytecode, property files, descriptors, etc.
  - Re-usable rules in XML descriptors
  - Cypher based queries
  - Reporting with comprehensive violation messages
- Integration in build process
  - Maven Plugin

- Rule definitions
  - Cypher queries specified in XML files…

- Rule definitions
  - Cypher queries specified in XML files…
  - …in a project directory (project/jqassistant), or…

- Rule definitions
  - Cypher queries specified in XML files…
  - …in a project directory (project/jqassistant), or…
  - …as part of plugins
  - Re-usable rules
    - Technical concepts, e.g. JPA entities, EJBs, test methods, etc.
    - Dependency concepts and constraints
      - class and package dependencies
      - cyclic package constraints
  - Rules can depend on each other
    - e.g. „package dependencies“ requires „type dependencies“
    - jQAssistant resolves correct order and executes only required rules

- Rule definitions: Concept

Rule definitions: Concept

```
<jqa:jqassistant-rules xmlns:jqa="...">

  <concept id="jpa2:Entity">
    <description>Labels all types annotated with
@javax.persistence.Entity with JPA and ENTITY.</description>
    <cypher><![CDATA[
      MATCH
        (t:TYPE)-[:ANNOTATED_BY]->()-[:OF_TYPE]->(a:TYPE)
      WHERE a.FQN="javax.persistence.Entity"
        SET t:JPA:ENTITY
       RETURN t AS jpaEntity
    ]]></cypher>
  </concept>

</jqa:jqassistant-rules>
```

Rule definitions: Concept

```
<jqa:jqassistant-rules xmlns:jqa="...">

  <concept id="jpa2:Entity">
    <description>Labels all types annotated with
@javax.persistence.Entity with JPA and ENTITY.</description>
    <cypher><![CDATA[
      MATCH
        (t:TYPE)-[:ANNOTATED_BY]->()-[:OF_TYPE]->(a:TYPE)
      WHERE a.FQN="javax.persistence.Entity"
        SET t:JPA:ENTITY
       RETURN t AS jpaEntity
    ]]></cypher>
  </concept>

</jqa:jqassistant-rules>
```

Rule definitions: Concept

```
<jqa:jqassistant-rules xmlns:jqa="...">

  <concept id="jpa2:Entity">
    <description>Labels all types annotated with
@javax.persistence.Entity with JPA and ENTITY.</description>
    <cypher><![CDATA[
      MATCH
        (t:TYPE)-[:ANNOTATED_BY]->()-[:OF_TYPE]->(a:TYPE)
      WHERE a.FQN="javax.persistence.Entity"
        SET t:JPA:ENTITY
       RETURN t AS jpaEntity
    ]]></cypher>
  </concept>

</jqa:jqassistant-rules>
```

- Rule definitions: Constraint

Rule definitions: Constraint

```
<jqa:jqassistant-rules xmlns:jqa="...">

  <constraint id="JpaEntitiesInModelPackage">
    <requiresConcept refId="jpa2:Entity"/>
    <description>All JPA entities must be located in the
packages named "model“.</description>
    <cypher><![CDATA[
      MATCH (p:PACKAGE)-[:CONTAINS]->(e)
        WHERE e:JPA AND e:ENTITY AND NOT(p.FQN =~ ".*\.model)
      RETURN
        e AS jpaEntity
    ]]></cypher>

  </constraint>
</jqa:jqassistant-rules>
```

Rule definitions: Constraint

```
<jqa:jqassistant-rules xmlns:jqa="...">

  <constraint id="JpaEntitiesInModelPackage">
    <requiresConcept refId="jpa2:Entity"/>
    <description>All JPA entities must be located in the
packages named "model“.</description>
    <cypher><![CDATA[
      MATCH (p:PACKAGE)-[:CONTAINS]->(e)
        WHERE e:JPA AND e:ENTITY AND NOT(p.FQN =~ ".*\.model)
      RETURN
        e AS jpaEntity
    ]]></cypher>

  </constraint>
</jqa:jqassistant-rules>
```

Rule definitions: Constraint

```
<jqa:jqassistant-rules xmlns:jqa="...">

  <constraint id="JpaEntitiesInModelPackage">
    <requiresConcept refId="jpa2:Entity"/>
    <description>All JPA entities must be located in the
packages named "model“.</description>
    <cypher><![CDATA[
      MATCH (p:PACKAGE)-[:CONTAINS]->(e)
        WHERE e:JPA AND e:ENTITY AND NOT(p.FQN =~ ".*\.model)
      RETURN
        e AS jpaEntity
    ]]></cypher>

  </constraint>
</jqa:jqassistant-rules>
```

Rule definitions: Constraint

```
<jqa:jqassistant-rules xmlns:jqa="...">

  <constraint id="JpaEntitiesInModelPackage">
    <requiresConcept refId="jpa2:Entity"/>
    <description>All JPA entities must be located in the
packages named "model“.</description>
    <cypher><![CDATA[
      MATCH (p:PACKAGE)-[:CONTAINS]->(e)
        WHERE e:JPA AND e:ENTITY AND NOT(p.FQN =~ ".*\.model)
      RETURN
        e AS jpaEntity
    ]]></cypher>

  </constraint>
</jqa:jqassistant-rules>
```

- Rule definitions: Group

Rule definitions: Group

```
<jqa:jqassistant-rules xmlns:jqa="...">

  <group id="default">
    <includeConstraint
      refId="abstractness:ApiMustNotDependOnImplementation"/>
    <includeConstraint
      refId="JpaEntitiesInModelPackage"/>
    <includeConstraint
      refId="EjbLocatedInImplementationPackage"/>
    <includeConstraint refId="TestClassNameHasTestSuffix"/>
    <includeConstraint refId="dependency:TypeCycles"/>
    <includeConstraint refId="dependency:ArtifactCycles"/>
  </group>

</jqa:jqassistant-rules>
```

Rule definitions: Group

```
<jqa:jqassistant-rules xmlns:jqa="...">

  <group id="default">
    <includeConstraint
      refId="abstractness:ApiMustNotDependOnImplementation"/>
    <includeConstraint
      refId="JpaEntitiesInModelPackage"/>
    <includeConstraint
      refId="EjbLocatedInImplementationPackage"/>
    <includeConstraint refId="TestClassNameHasTestSuffix"/>
    <includeConstraint refId="dependency:TypeCycles"/>
    <includeConstraint refId="dependency:ArtifactCycles"/>
  </group>

</jqa:jqassistant-rules>
```

- Maven goals
  - scan
    - Scan the byte code
  - available-rules
    - List all available rules
  - effective-rules
    - List all rules which would be applied using current configuration
  - analyze
    - Execute analysis according to the effective rules
  - report
    - Create a report for maven sites
  - server
    - Run the embedded Neo4j server

- Plugin based and extensible
  - jQAssistant is only a framework
  - Plugins provide scanner and rules
    - Java
      - class and property file scanner
      - dependency concepts and constraints (cycles)
    - JPA2
      - persistence descriptor scanner (persistence.xml)
      - JPA entity concept
    - EJB3
      - concepts for EJB types and interfaces (local, remote)
    - JUnit4
      - Test methods and classes
      - Ignored tests

Exploration And Verification Of Java Applications

Using A Graph Database

Live Demo #2
Concepts and Constraints

Exploration And Verification Of Java Applications
Using A Graph Database

Live Demo #3
Impact Analysis

jQAssistant – Wishlist

- Plugins
  - More scanners
  - e.g. CDI
    - Scanner for beans.xml
    - Concepts for beans, injection points and producer, interceptors, delegates, …
  - Rules, rules, rules
    - Community?

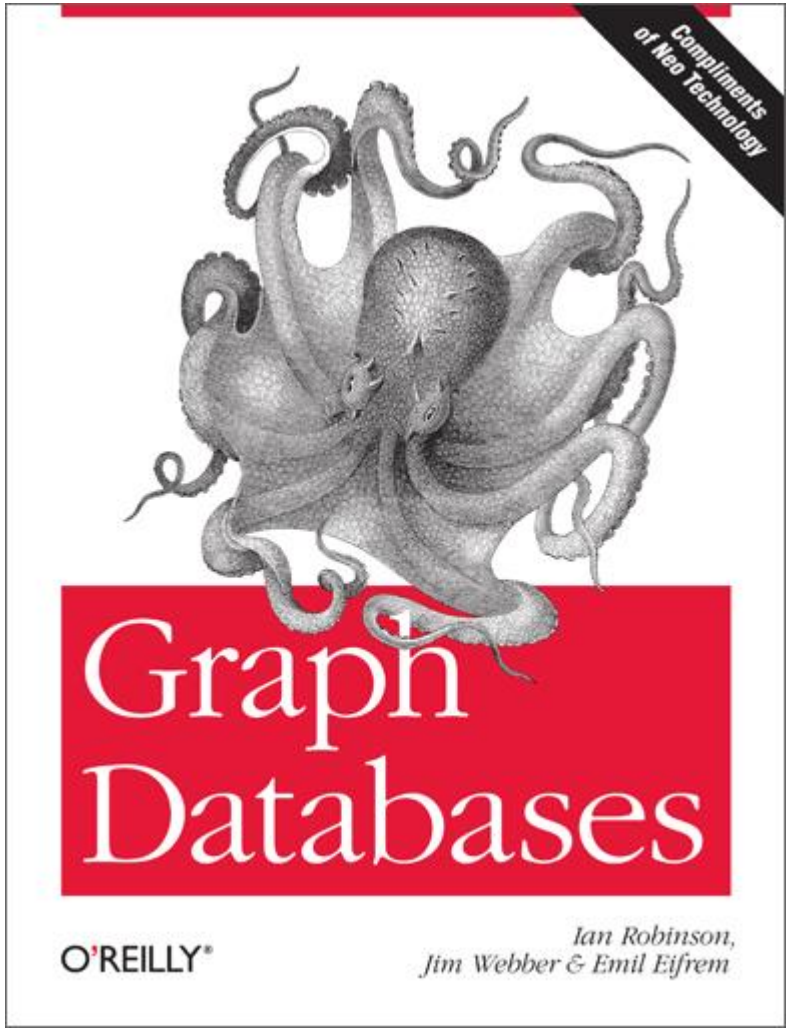
jQAssistant – Wishlist

buschmais
Beratung . Technologie . Innovation

- Visualization
  - Heat maps, e.g. dependencies of packages or modules
- Tool integration
  - Sonar (Work in progress)
  - Gradle
  - Jenkins
  - JUnit: dynamic test suites
  - Eclipse
    - On-The-Fly scan
    - „Manual“ query execution
    - Definable queries, e.g. impact analysis

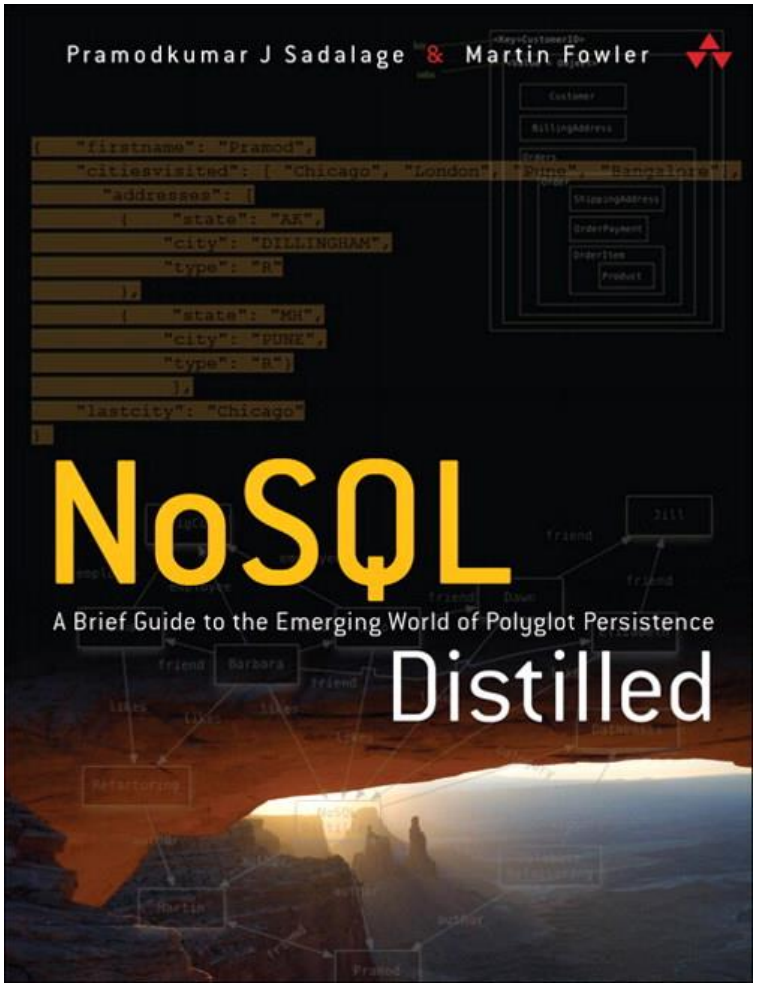
More Information

- Graph Databases
  - Ian Robinson, Jim Webber, Emil Eifrem
  - O'Reilly Media
  - 1. Auflage
  - ISBN: 978-1449356262



More Information

- NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence
  - Pramodkumar J. Sadalage, Martin Fowler
  - Addison-Wesley Longman
  - ISBN: 978-0321826626



More Information

buschmais
Beratung . Technologie . Innovation

- Hypermodelling - Next Level Software Engineering with Data Warehouses

  http://accepted.hypermodelling.com/frey_magdeburg_dissertation_hypermodelling_2013.pdf

- Oliver Gierke: Ooops, where did my architecture go?

  http://www.slideshare.net/olivergierke/whoops-where-did-my-architecture-go-10414858

More Information

buschmais
Beratung . Technologie . Innovation

- Raoul-Gabriel Urma: Expressive and Scalable Source Code Queries with Graph Databases [Paper]
  http://urma.com/pdf/oopsla13.pdf
- Pavlo Baron: Graphlr, a ANTLR storage in Neo4j
  http://github.com/pavlobaron/graphlr
- Michael Hunger: Class-Graph
  http://github.com/jexp/class-graph

buschmais

Beratung . Technologie . Innovation

Vielen Dank! Thank you!

Questions?

buschmais.de

facebook.com/buschmais

twitter.com/buschmais