

# **JQASSISTANT – DOUBLE FEATURE**

**JUG CH**

**16. Februar 2021**

**Dirk Mahler & Stephan Pirnbaum**

- ▲ Wer wir sind
  - ▲ Dresdner IT-Beratungsunternehmen, gegründet im Jahre 2008
- ▲ Unsere Schwerpunkte
  - ▲ Architekturberatung und Entwicklung moderner Geschäftsanwendungen
  - ▲ Software-Qualitätsanalysen und -Sicherung
  - ▲ Köpfe hinter jQAssistant: Entwicklung, Trainings, Workshops
- ▲ Unsere Kunden im Bereich Software-Qualitätsanalyse
  - ▲ u. a. ITZBund, Sächsische Aufbaubank-Förderbank, ASML, GlobalFoundries, Thyssenkrupp Steel

# **ANALYSE, DOKUMENTATION & VALIDIERUNG VON SOFTWARESTRUKTUREN**

**JUG CH**

**16. Februar 2021**

**Dirk Mahler**

Exploration

Analyse



# jQAAssistant

<http://jqassistant.org>

Open Source (GPL v3)

Validierung



Reporting

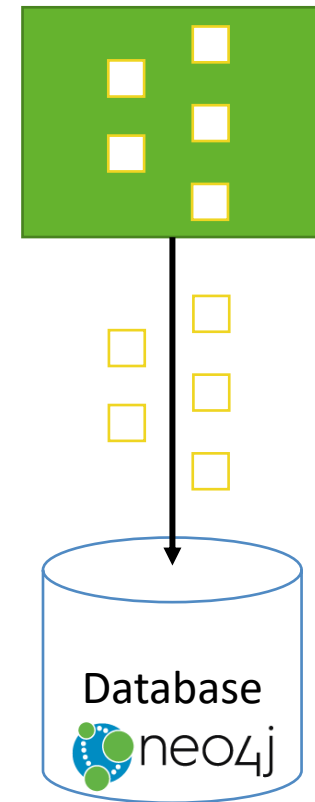


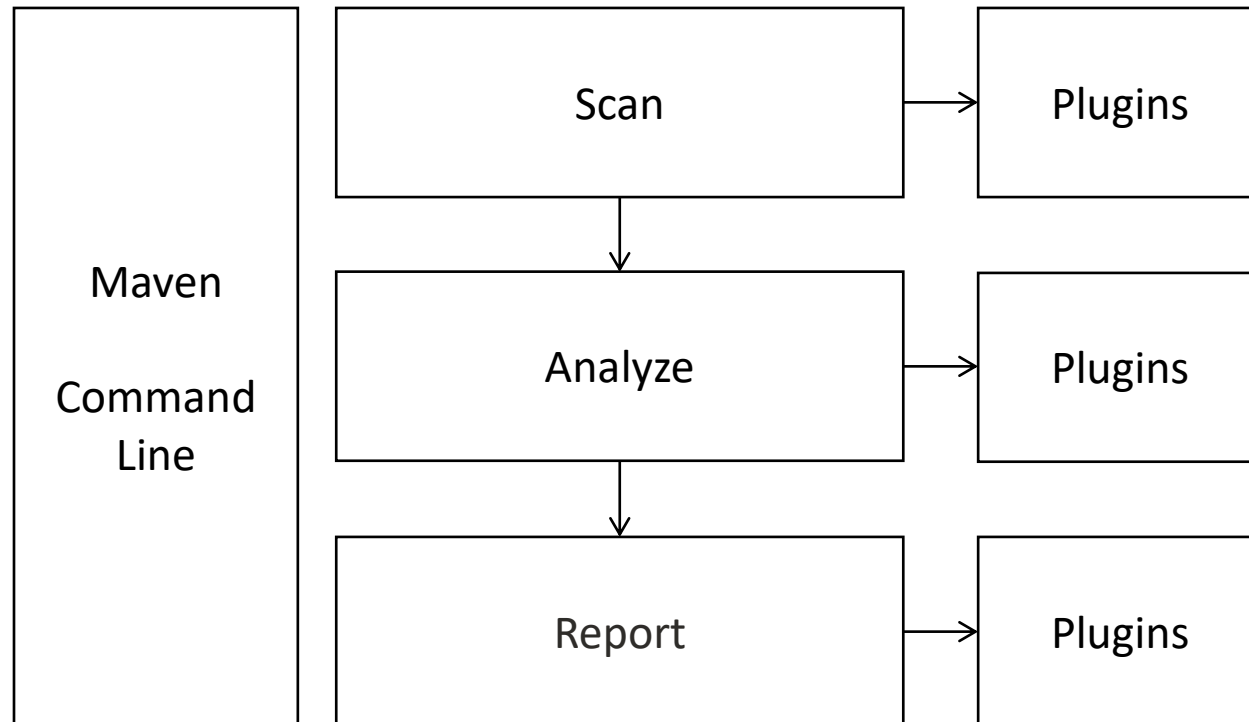
Dokumentation



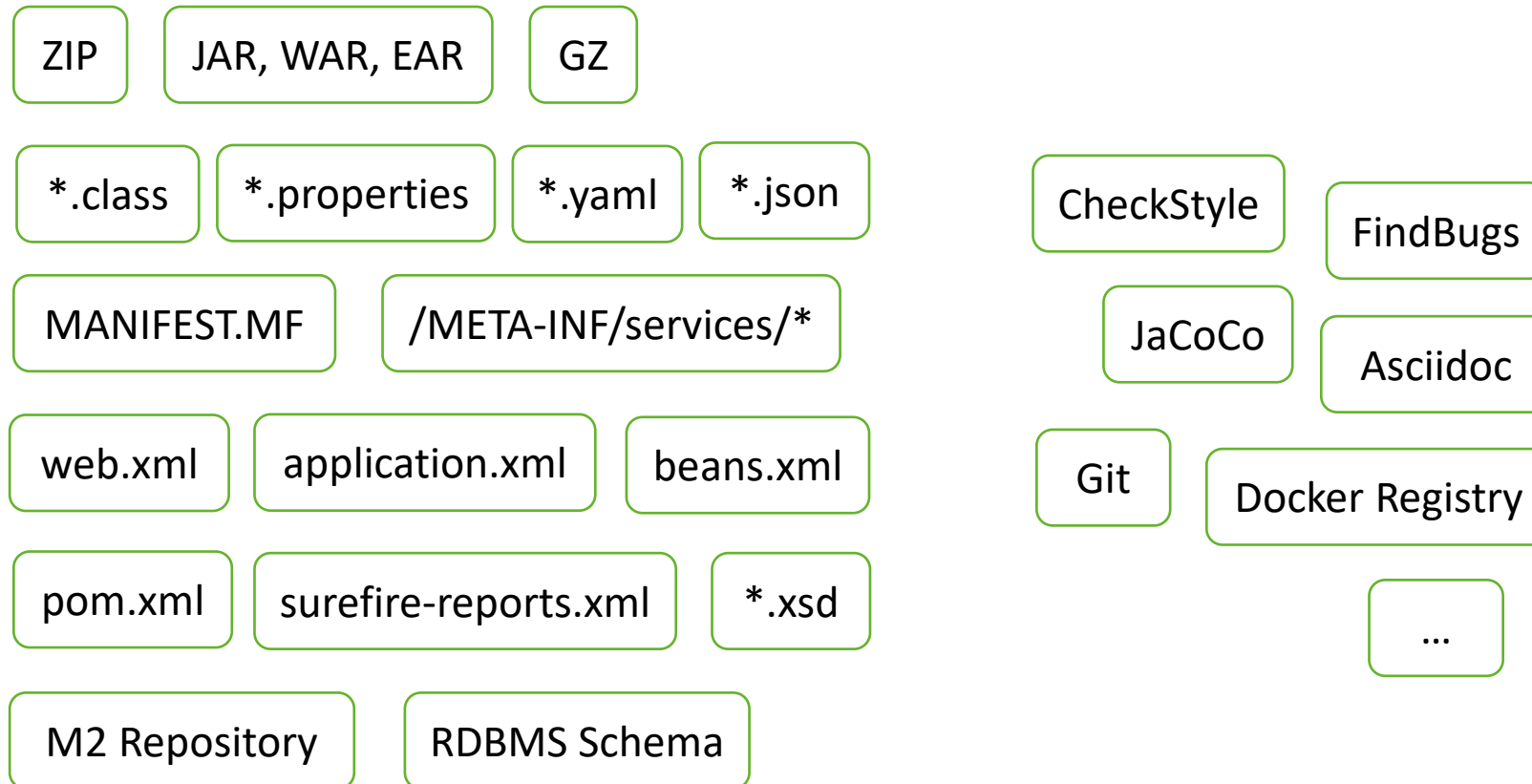
# DIE IDEE

- ^ Einlesen von Softwarestrukturen
- ^ Speichern in einer Datenbank
  - ^ (Automatisierte) Ausführung von Abfragen
    - ^ Exploration
    - ^ Anreicherung von Architektur-/Design-Konzepten
    - ^ Auffinden von Verletzungen
- ^ Erzeugen von Reports
  - ^ Strukturen und Metriken
  - ^ Tabellarisch und Grafisch



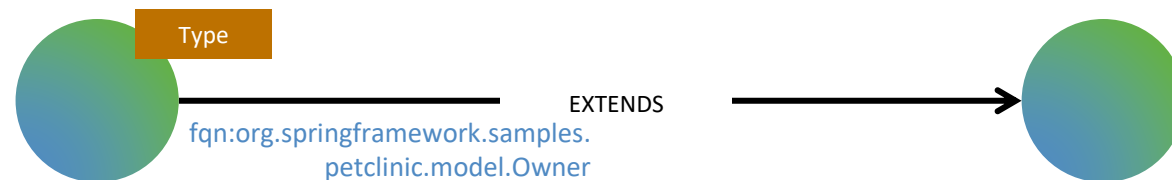


# SCANNER-PLUGINS

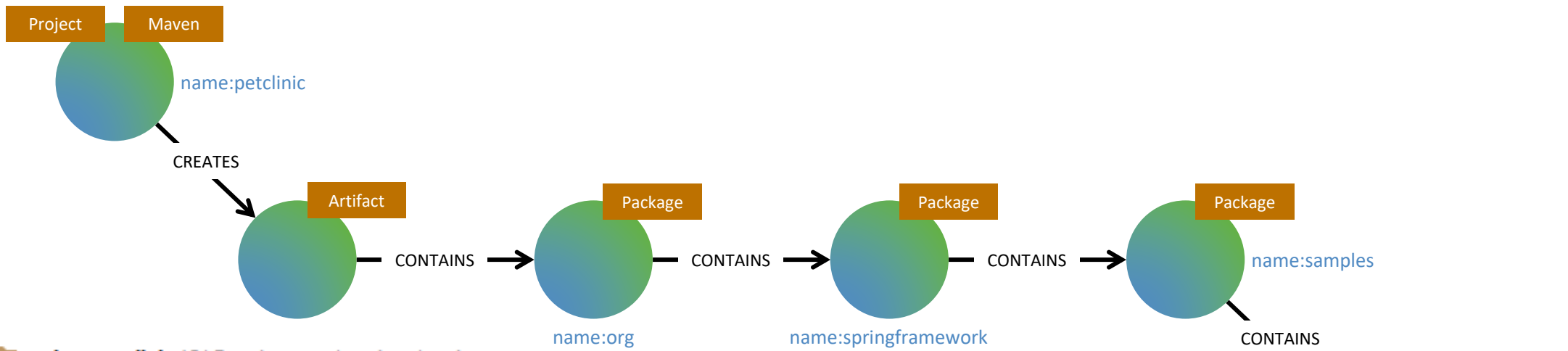


# ^ Softwarestrukturen als Graph

- ^ Nodes
- ^ Labels
- ^ Properties
- ^ Relationships







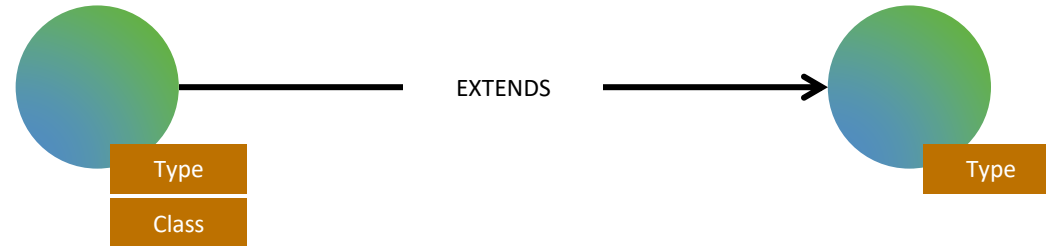
```

spring-petclinic (C:\Development\projects\spring-pet
├── .idea
├── jqassistant
└── src
    ├── main
    │   └── java
    │       ├── org.springframework.samples.petclinic
    │           └── model
    │               ├── BaseEntity
    │               ├── NamedEntity
    │               ├── Owner
    │               ├── package-info.java
    │               ├── Person
    │               ├── Pet
    │               ├── PetType
    │               ├── Specialty
    │               ├── Vet
    │               ├── Vets
    │               └── Visit
  
```



## ^ Exploration mittels Abfragen

^ Welche Java-Klasse leitet von einer anderen ab?



## ^ Graph-Strukturen als ASCII-Art

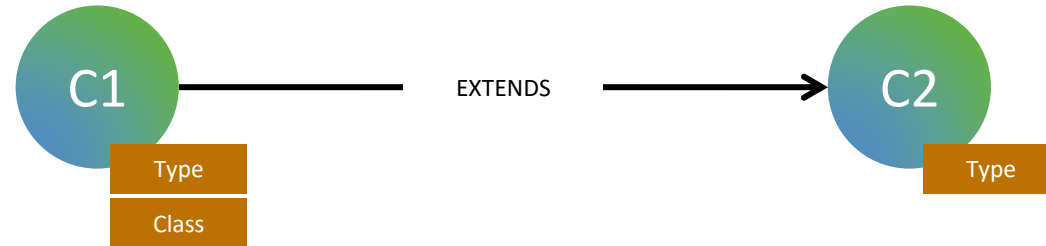
^ () als Knoten

^ -[]-> als gerichtete Beziehungen

( ) - [ ] - > ( )

## ^ Exploration mittels Abfragen

^ Welche Java-Klasse leitet von einer anderen ab?



## ^ Graph-Strukturen als ASCII-Art

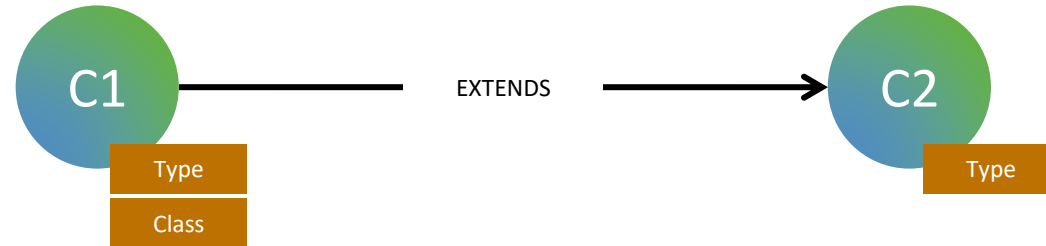
^ () als Knoten

^ -[]-> als gerichtete Beziehungen

(c1) - [] -> (c2)

## ^ Exploration mittels Abfragen

^ Welche Java-Klasse leitet von einer anderen ab?



## ^ Graph-Strukturen als ASCII-Art

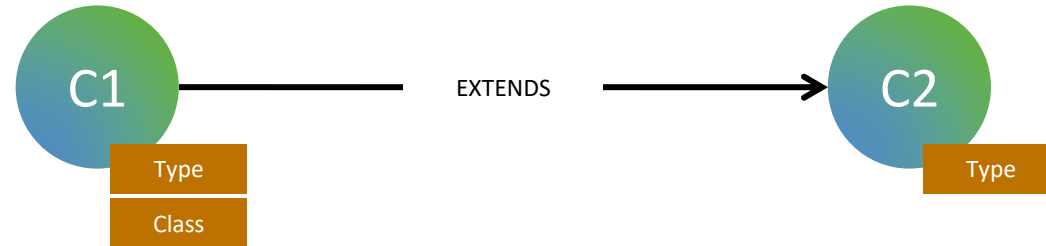
^ () als Knoten

^ -[]-> als gerichtete Beziehungen

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

## ^ Exploration mittels Abfragen

^ Welche Java-Klasse leitet von einer anderen ab?



## ^ Graph-Strukturen als ASCII-Art

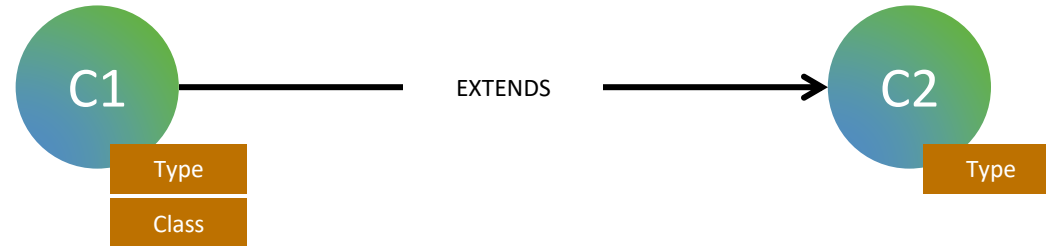
^ () als Knoten

^ -[]-> als gerichtete Beziehungen

`(c1:Class) - [ :EXTENDS ] -> (c2:Type)`

## ^ Exploration mittels Abfragen

^ Welche Java-Klasse leitet von einer anderen ab?



^ „Pattern matching“ → Cypher-Kernprinzip

```
MATCH (c1:Class)-[:EXTENDS]->(c2:Type)
RETURN c1.fqn, c2.fqn
```

# DEMO

## Exploration & Analyse

[www.buschmais.de](http://www.buschmais.de)

BUSCHMAIS



## ^ Exploration & Analyse

### ^ Suche nach (Anti-)Mustern im Code

- ^ „Log & Throw“

### ^ Erhebung von Metriken

- ^ Identifizierung von Hotspots

### ^ Ad-Hoc-Analyse von Abhängigkeiten

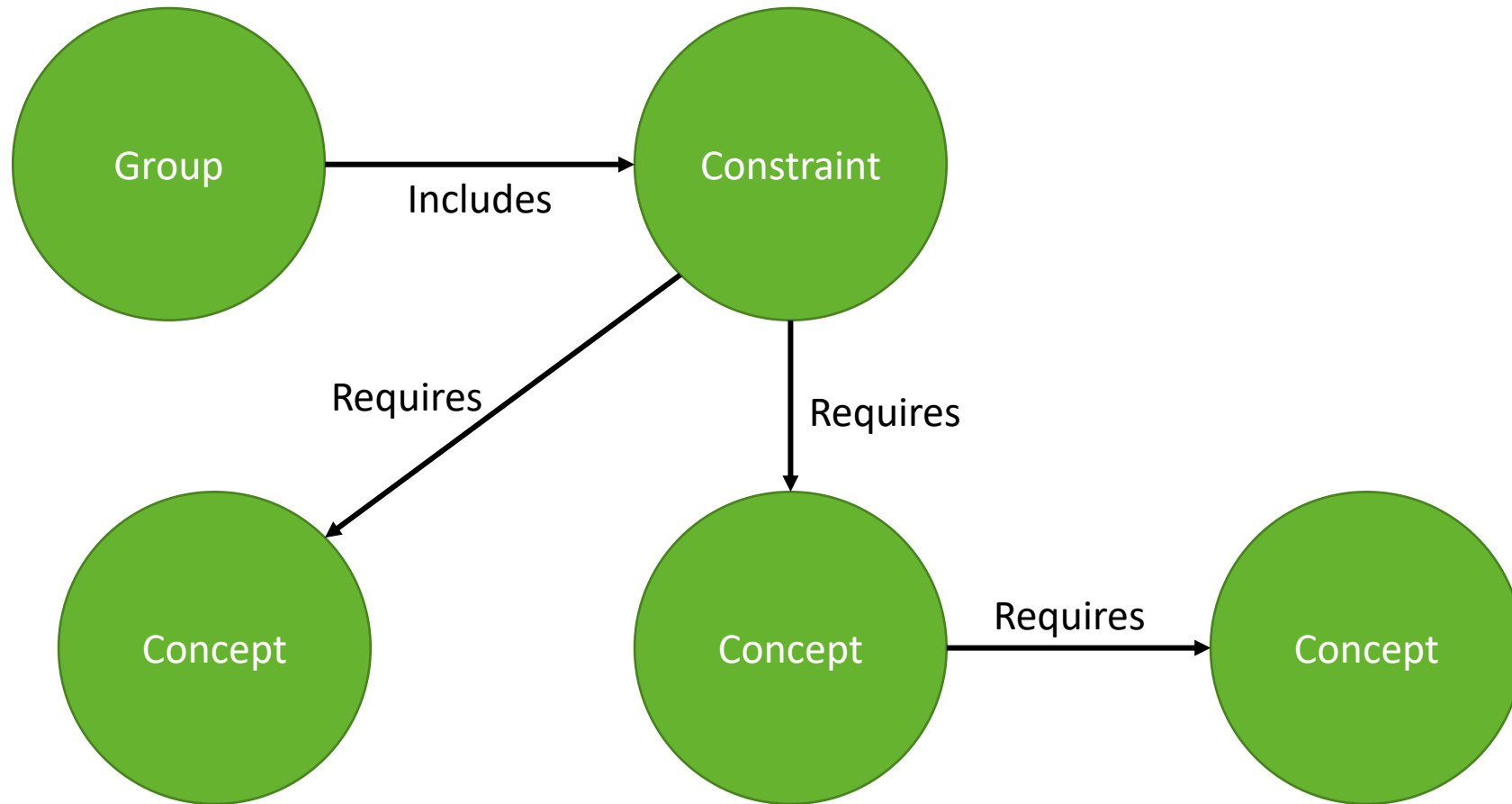
- ^ Fachliche bzw. technische Module

- ^ Verwendung von Frameworks und Bibliotheken

- Planung & Risikoabschätzungen

# ANALYSE

- ^ Ausführung von Regeln
  - ^ Hinterlegt in AsciiDoc- oder XML-Dokumenten
  - ^ Erzeugung von Reports
- ^ Concept
  - ^ Anreicherung von Architektur- und Design-Konzepten
- ^ Constraint
  - ^ Auffinden von Verletzungen
- ^ Group
  - ^ Zusammenfassen von Regeln zur gemeinsamen Ausführung



# ^ Concept

== JPA Entities

```
[[jpa2:Entity]]
```

.Labels all types annotated with @javax.persistence.Entity with Jpa and Entity.

```
[source,cypher,role=concept]
```

```
----
```

```
MATCH
```

```
  (t:Type)-[:ANNOTATED_BY]->()-[:OF_TYPE]->(a:Type)
```

```
WHERE a.fqn = "javax.persistence.Entity"
```

```
  SET t:Jpa:Entity
```

```
RETURN t AS Entity
```

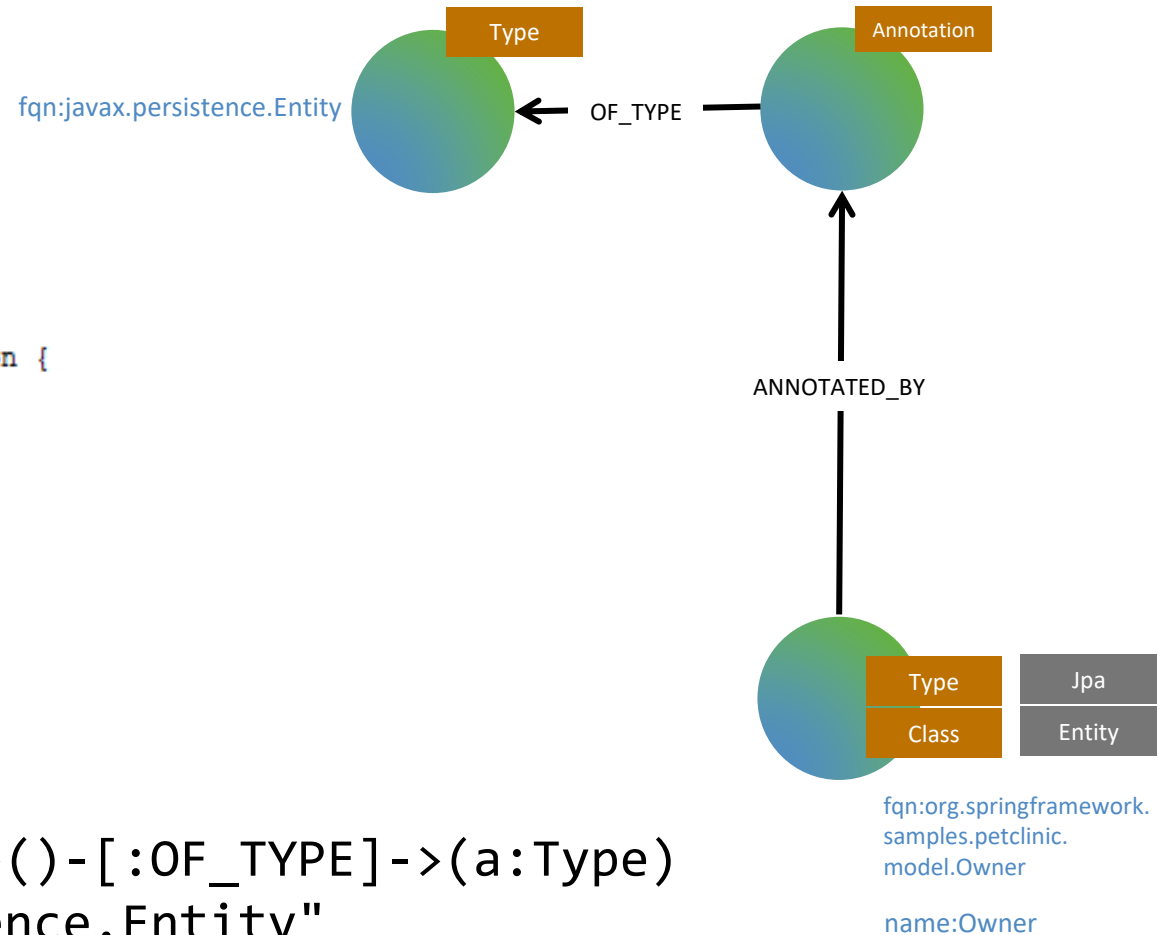
```
----
```

# Λ Concept

```
@Entity
@Table(name = "owners")
public class Owner extends Person {
    @Column(name = "address")
    @NotEmpty
    private String address;
}
```

MATCH

```
(t:Type)-[:ANNOTATED_BY]->()-[:OF_TYPE]->(a:Type)
WHERE a.fqn = "javax.persistence.Entity"
SET t:Jpa:Entity
RETURN t AS Entity
```



# ^Constraint

```
[[model:JpaEntityInModelPackage]]
```

```
.All JPA entities must be located in packages named "model".
```

```
[source,cypher,role=constraint,requiresConcepts="jpa2:Entity"]
```

```
----
```

```
MATCH
```

```
    (package:Package) - [:CONTAINS] -> (entity:Jpa:Entity)
```

```
WHERE
```

```
    package.name <> "model"
```

```
RETURN
```

```
    entity AS EntityInWrongPackage
```

```
----
```

# ^Group

```
[[default]]
```

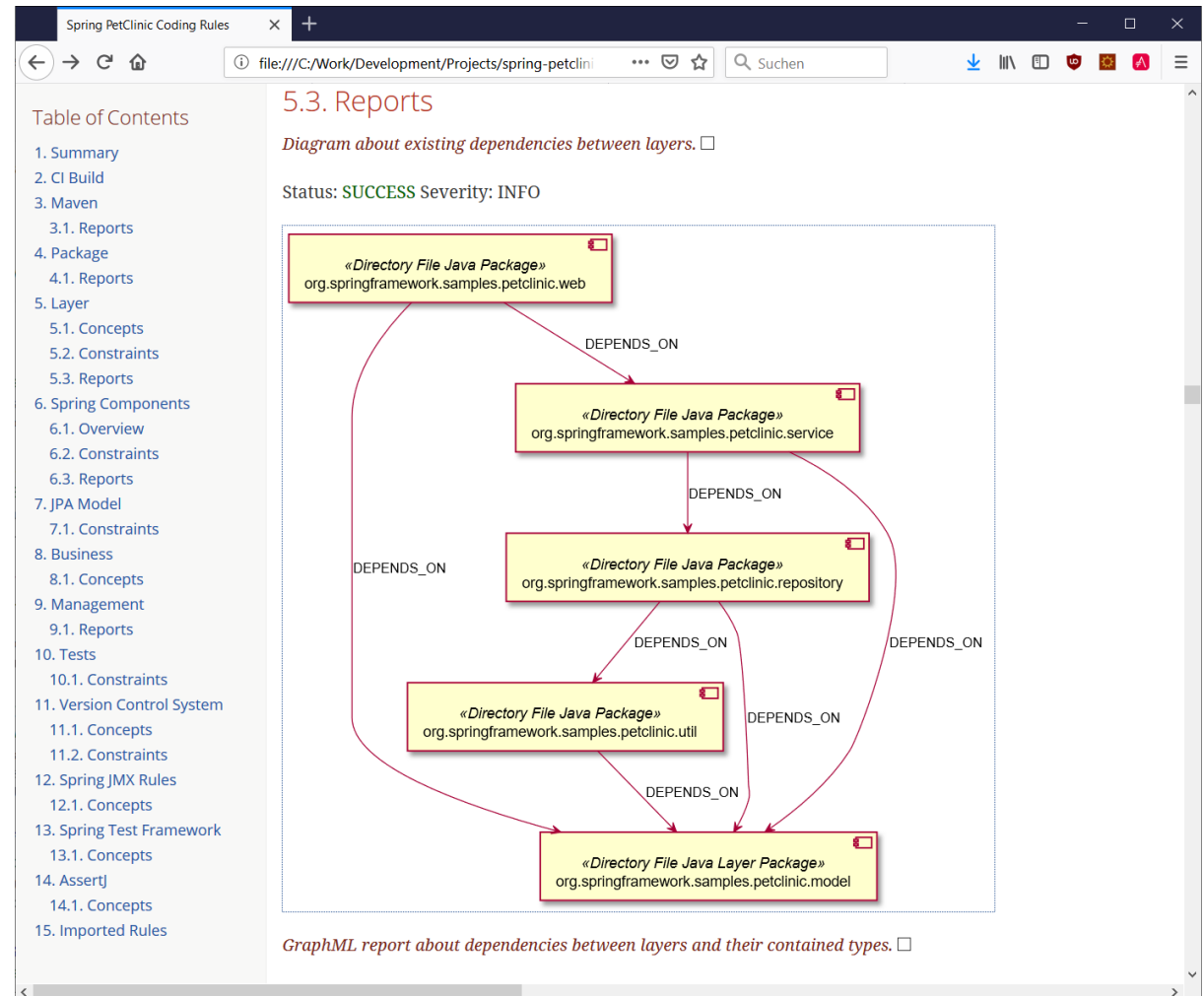
```
[role=group,includesConstraints="model:*"]
```

```
== Naming Rules
```

```
...
```

# REPORTS

- ▲ Ergebnisse von Konzepten & Constraints
  - ▲ Tabellen & Diagramme
  - ▲ CSV, GraphML, ...
- ▲ Regeln in AsciiDoc
  - ▲ Ausgabe als HTML
  - ▲ Eingebettete Reports
- ▲ Individuelle Reports
  - ▲ Dateien, Datenbank, etc.
  - ▲ Als Plugin realisierbar





# REPORTS

^ SonarQube-Plugin

^ Verfügbar via Marketplace

The screenshot displays the SonarQube web interface for a project named 'petclinic'. The main content area shows a list of issues, with the following details visible:

- Issue 1:** File: `src/.../samples/petclinic/SomeEntity.java`. Description: "All JPA entities must be located in packages named 'model'." Severity: Major. Created: 8 minutes ago.
- Issue 2:** File: `src/.../petclinic/repository/jdbc/JdbcOwnerRepositoryImpl.java`. Description: "There must be no dependencies between layers that are not explicitly defined. Layer: repository, LayerDependency: util, TypeDependency: org.springframework.samples.petclinic.util.EntityUtils." Severity: Major. Created: 5 years ago.
- Issue 3:** File: `src/.../petclinic/repository/jdbc/JdbcPetRepositoryImpl.java`. Description: "There must be no dependencies between layers that are not explicitly defined. Layer: repository, LayerDependency: util, TypeDependency: org.springframework.samples.petclinic.util.EntityUtils." Severity: Major. Created: 5 years ago.
- Issue 4:** File: `src/.../petclinic/repository/jdbc/JdbcVetRepositoryImpl.java`. Description: "There must be no dependencies between layers that are not explicitly defined. Layer: repository, LayerDependency: util, TypeDependency: org.springframework.samples.petclinic.util.EntityUtils." Severity: Major. Created: 6 years ago.
- Issue 5:** File: `src/.../samples/petclinic/util/EntityUtils.java`. Description: "There must be no dependencies between layers that are not explicitly defined. Layer: util, LayerDependency: model, TypeDependency: org.springframework.samples.petclinic.model.BaseEntity." Severity: Major. Created: 5 years ago.

The left sidebar shows a filter for 'Rule' with 2 selected rules: 'model.JpaEntityInModelPackage' (1 issue) and 'layer.LayerDependencyViolation' (4 issues). The search bar contains 'jQAssistant'.

# DEMO

## Validierung & Dokumentation

[www.buschmais.de](http://www.buschmais.de)

BUSCHMAIS

# LINKS

^ Projekt-Homepage

^ <https://jqassistant.org>

^ 101 Tutorials

^ <https://101.jqassistant.org>

^ Spring PetClinic

^ <https://github.com/jqassistant-demo/spring-petclinic/>

^ Talk: Fixing Your Microservice Architecture Using Graph Analysis

^ <https://jqassistant.org/fix-your-microservice-architecture-using-graph-analysis/>

# ^ Validierung & Dokumentation

## ^ Typische Regeln

- ^ Management von internen/externen Abhängigkeiten
- ^ Namens- und Strukturregeln
- ^ Design bzw. Verwendung von Schnittstellen (intern & extern)
- ^ Testdesign

## ^ „Lebendige Dokumentation“ → ADRs

## ^ Kontinuierliches Feedback und Reporting für...

- ^ Agile Teams
- ^ Test/QA-Abteilungen (bei Realisierung von Projekten durch Dritte)

## KONTAKT

Dirk Mahler

[dirk.mahler@buschmais.com](mailto:dirk.mahler@buschmais.com)

Twitter: @dirkmahler

@buschmais

## BUSCHMAIS GbR

Leipziger Straße 93

01127 Dresden

Tel. +49 351 3209230

[info@buschmais.com](mailto:info@buschmais.com)

[www.buschmais.de](http://www.buschmais.de)