

Yes We Scan Software Analysis Using jQAAssistant

JavaLand 2015

Dirk Mahler

AGENDA

- Black Boxes Called Artifacts
- Software As A Graph
- jQAssistant
- Let's Explore Libraries!

Yes We Scan – Software Analysis Using jQAssistant

Black Boxes Called Artifacts

■ Artifact

- Result of a build/integration processes
- Black Box – What's in it?

acme.war

```
/index.xhtml  
/WEB-INF/lib/model.jar
```

Person.java

```
@Entity  
public class Person {  
    private String name;  
}
```

pom.xml

```
<dependency>  
    <groupId>org.eclipse.persistence</groupId>  
    <artifactId>persistence-api</artifactId>  
    <version>2.1.0</version>  
</dependency>
```

model.jar

```
/com/acme/model/Person.class  
/META-INF/persistence.xml
```

person.sql

```
create table PERSON (id number, name varchar(255))
```

- Let's open it!

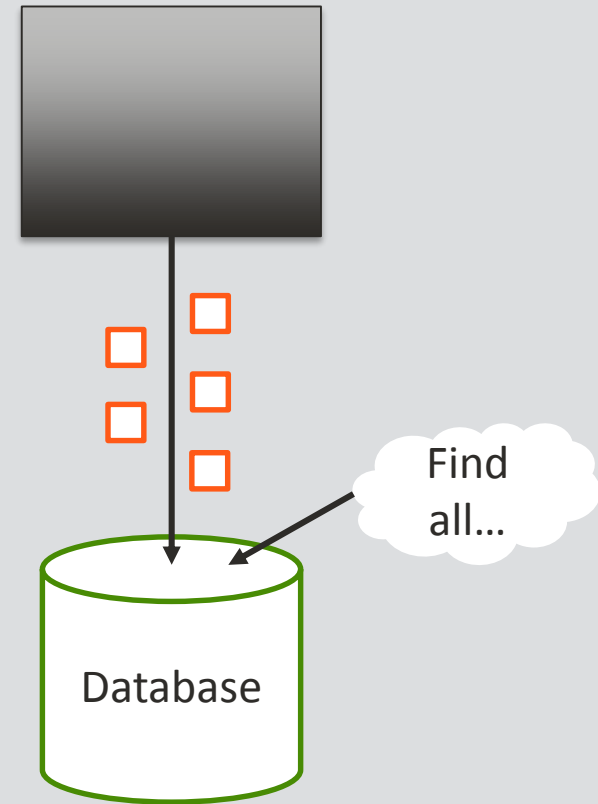
■ The Idea

- Scan software structures
- Store them into a database
- Execute queries

■ Why?

- Holistic view on „real world“
- Exploration
 - effective searches beyond IDE capabilities
- Individual metrics and visualization
 - Complexity, Dependencies
- Validation
 - Coding rules, design & architecture constraints
- Reporting

- And finally... it's fun!



- Which database?
 - Flexible data model
 - schema free
 - extensible
 - good support for relationships
 - Easy-to-use query language
 - expressive
 - Performance 😊



<http://neo4j.org>

**Yes We Scan
Software Analysis Using jQAssistant**

Software As A Graph

- All we need is...

- Nodes
- Labels
- Properties
- Relationships

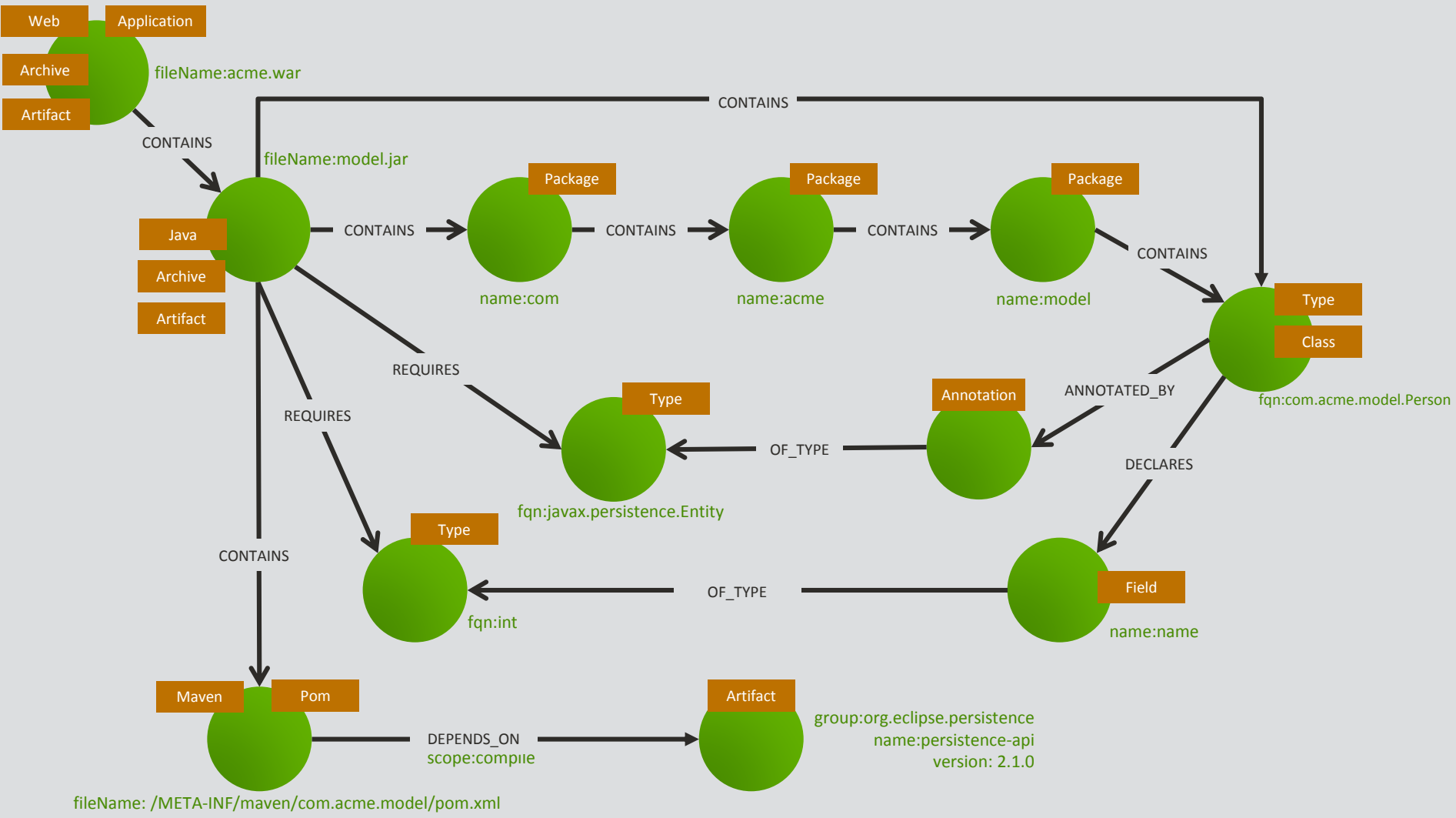


- Modeling is just...

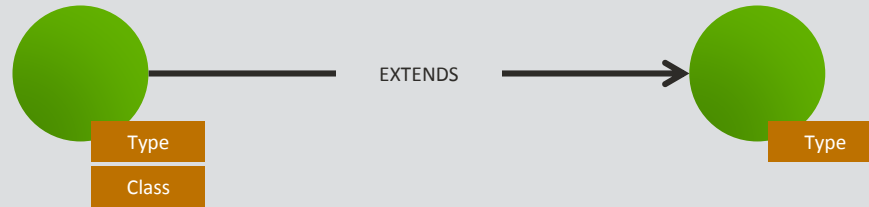
- Taking a pen
- Drawing the structures on a whiteboard (i.e. the database)

- We don't need...

- Foreign keys
- Tables and schemas
- Knowledge in graph theory

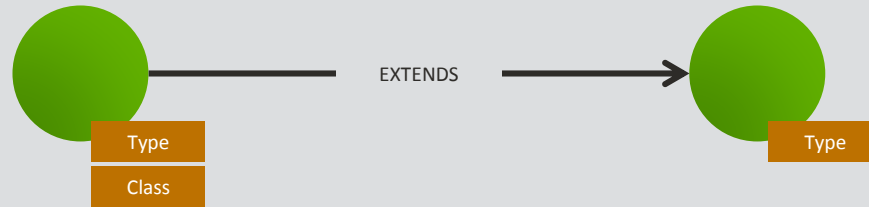


- Explore an application using queries
 - Which class extends from another class?



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

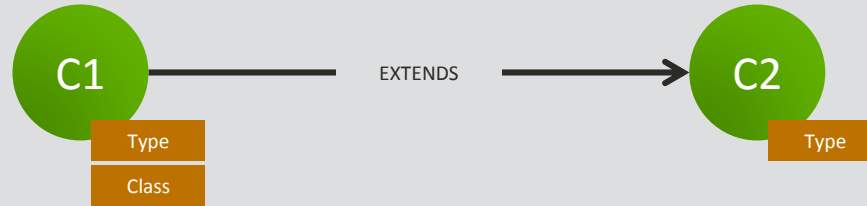
- Explore an application using queries
 - Which class extends from another class?



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

() - [] -> ()

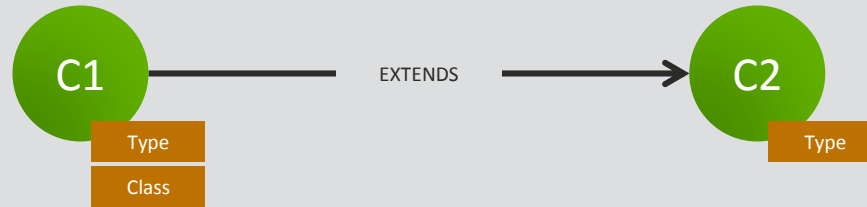
- Explore an application using queries
 - Which class extends from another class?



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

(c1) - [] -> (c2)

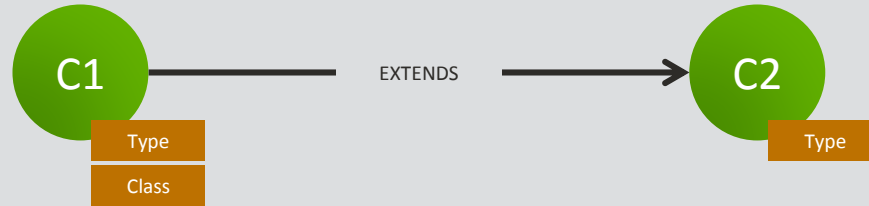
- Explore an application using queries
 - Which class extends from another class?



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

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

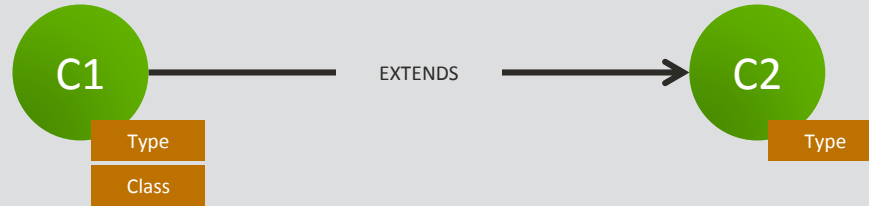
- Explore an application using queries
 - Which class extends from another class?



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

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

- Explore an application using queries
 - Which class extends from another class?



- Pattern matching is the core principle of Cypher

MATCH

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

RETURN

```
c1.fqn, c2.fqn
```

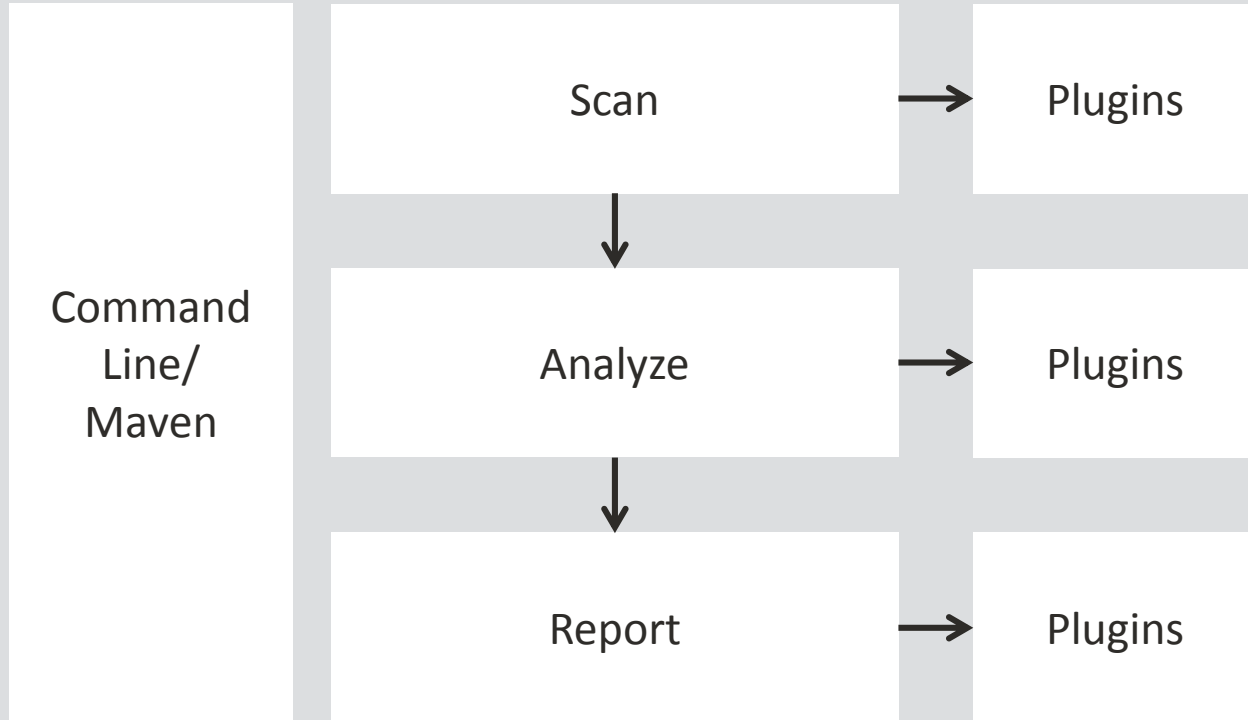
**Yes We Scan
Software Analysis Using jQAssistant**

jQAssistant

jqAssistant

<http://jqAssistant.org>

- Open Source: GPLv3
- Current release: 1.0.0-RC1
 - initiated: 03/2013
 - first stable release: 04/2015
- Neo4j Community Edition embedded
 - no installation necessary



■ Getting Started – Command Line

- `unzip jqassistant.distribution-1.0.0.zip`
- `cd jqassistant.distribution-1.0.0/bin`

- `jqassistant.sh scan -f model.jar`
- `jqassistant.sh scan -f acme.war`
- `jqassistant.sh scan -f acme.ear`
- `jqassistant.sh scan -u http://somewhere.com/acme.ear`
- `jqassistant.sh scan -u`
`maven:repository::http://host/releases`
- `jqassistant.sh scan -u`
`rdbms:schema::jdbc:oracle:thin:user/secret@host:1521:sid`

- `jqassistant.sh server`
→ <http://localhost:7474>

■ Getting Started – Maven Project

```
<build>
  <plugins>
    <plugin>
      <groupId>com.buschmais.jqassistant.scm</groupId>
      <artifactId>jqassistant-maven-plugin</artifactId>
      <version>1.0.0</version>
    </plugin>
  </plugins>
</build>
```

- `mvn install jqassistant:scan`
- `mvn jqassistant:server`

■ Available scanner plugins

ZIP

JAR, WAR, EAR

GZ

*.class

*.properties

MANIFEST.MF

/META-INF/services/*

web.xml

application.xml

beans.xml

pom.xml

surefire-reports.xml

*.xsd

Liquibase

CheckStyle

FindBugs

JaCoCo

Git

*.java

■ Plugin API is public

M2 Repository

RDBMS Schema

**Yes We Scan
Software Analysis Using jQAssistant**

Let's Explore Libraries!

- Examples
 - Artifacts and Maven descriptors
 - API changes between releases

 - Declared deprecations
 - Thrown exceptions
 - Structural problems (metrics)
 - Declared Members
 - Cyclomatic Complexity
 - Fan In/Fan Out of classes and packages
 - Package Cycles

**Yes We Scan
Software Analysis Using jQAssistant**

Live Demo

- What makes the difference?
 - Flexible data model
 - Get information about different aspects of the same software in just one place
 - Extend it using plugins
 - Expressive queries
 - Enrich data with user defined concepts
 - Create specific metrics and reports

Thank You! – Questions?

Mail: info@jqassistant.de

Web: jqassistant.de

Twitter: [@jqassistant](https://twitter.com/jqassistant)