DORM
Core & Extensions

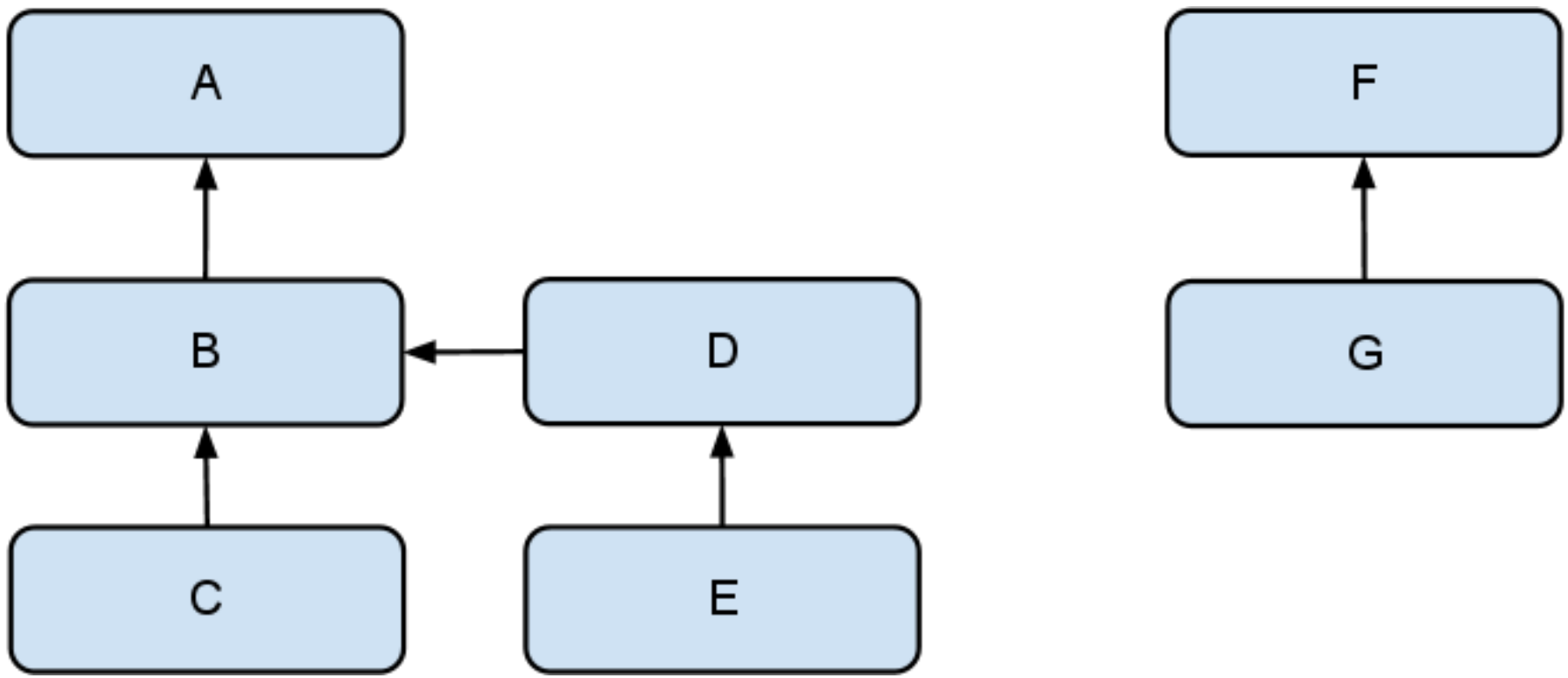
Last update : 01/08/2011

Core

1. the model
2. the processor
3. the workflow

Flexible model

- each dependency contains x metadatas and 1 physical file
- n dependency deep
- generic and extensible metadata



Core model composition

- Dorm Metadata

- Dependency

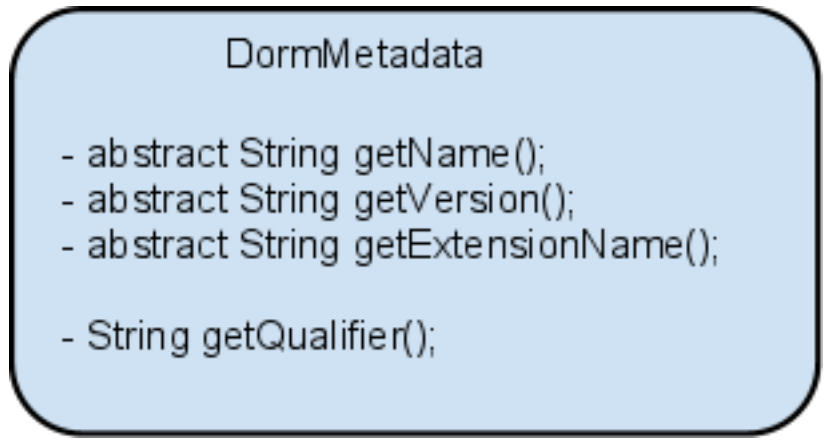
- Dependency Node

Metadata and Dependency are immutables.

Dorm Metadata

The extension point of the model

Each metadata must implement the abstract class **DormMetadata**.



The extension name is the unique name of an extension (for example "maven").

To respect immutability of the core model, all extensions must be immutables.

Dorm Metadata

The extension point of the model

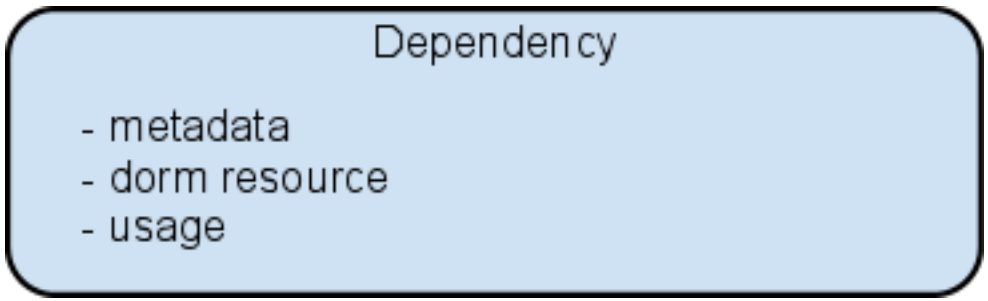
The functional id represents the unique id of a dorm artifact.
The pattern is :

extension name : name : version

The functional id should not be redefined by the plugins.

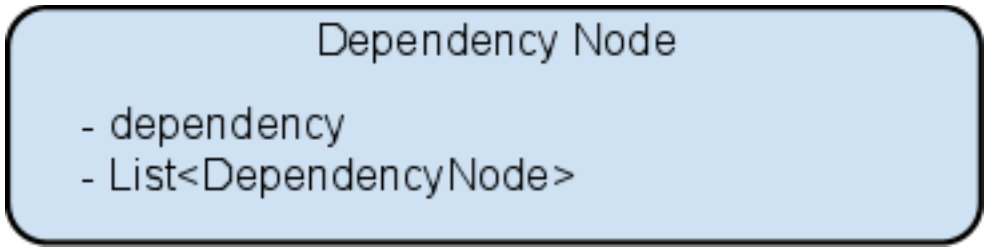
Dependency

The dependency links a metadata with a file.
This link is defined by an usage.



Dependency node

The node dependency represents the dependency in a specific graph. A node can contains n-child.



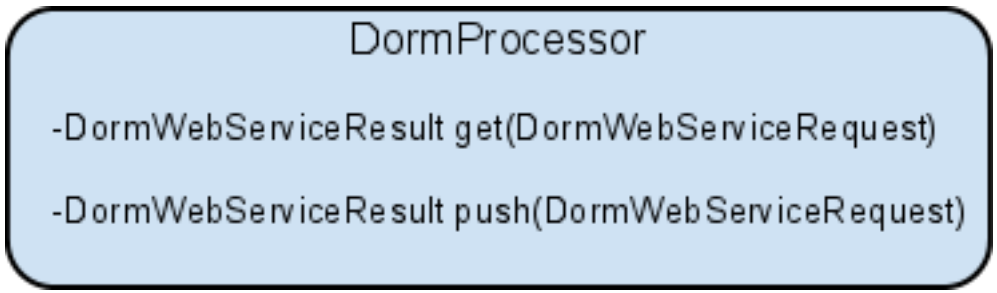
A graph of dependency nodes can be filtered and read through by visitors.

The visitors can use validators to filter a graph.
The visit process use the filter pattern.

Dorm Processor

The processor is the entry point of the dorm application.

It's used by the webservice or any other service to push and retrieve dependencies from dorm.



Dorm Processor

A processor extension is the 2nd extension point which permits to execute a specific logic during operations like push, get, etc.

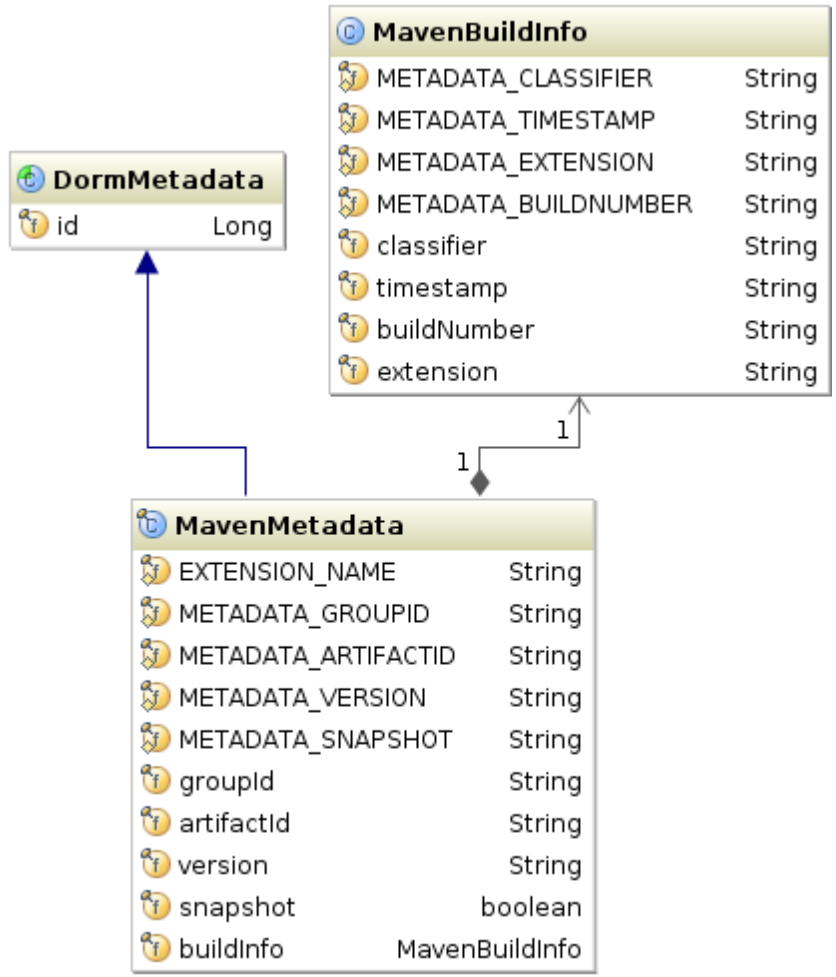
Maven extension

1. the model
2. the processor

Maven Metadata

The model extension point

Add maven metadatas
to the core model.



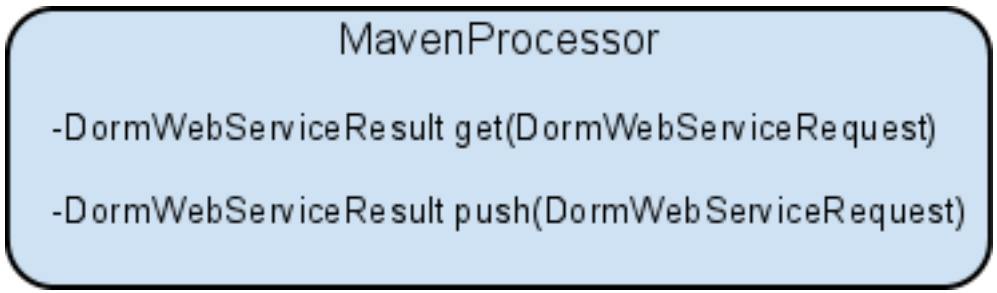
The name is : **groupId:artifactId:timestamp:buildNumber:classifier:extension**

The extension name is **maven**.

Maven processor

Maven processor is the extension point to the dorm workflow.

The main processor deleguates to the maven processor for push and get operations.



Dorm Dao

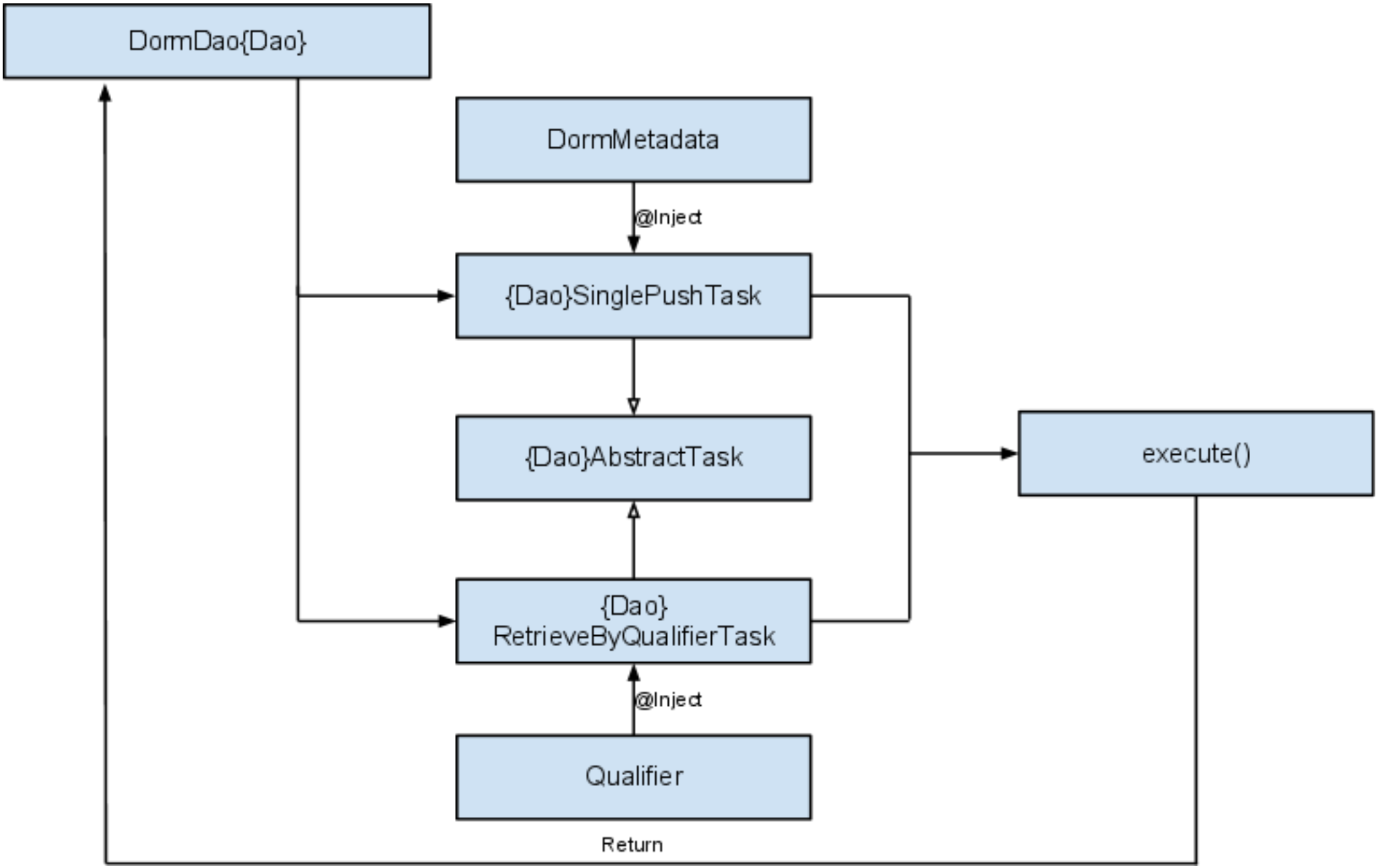
Three Dao are available :

- JDCB Dao.
- Neo4j Dao.
- Nuxeo Dao.

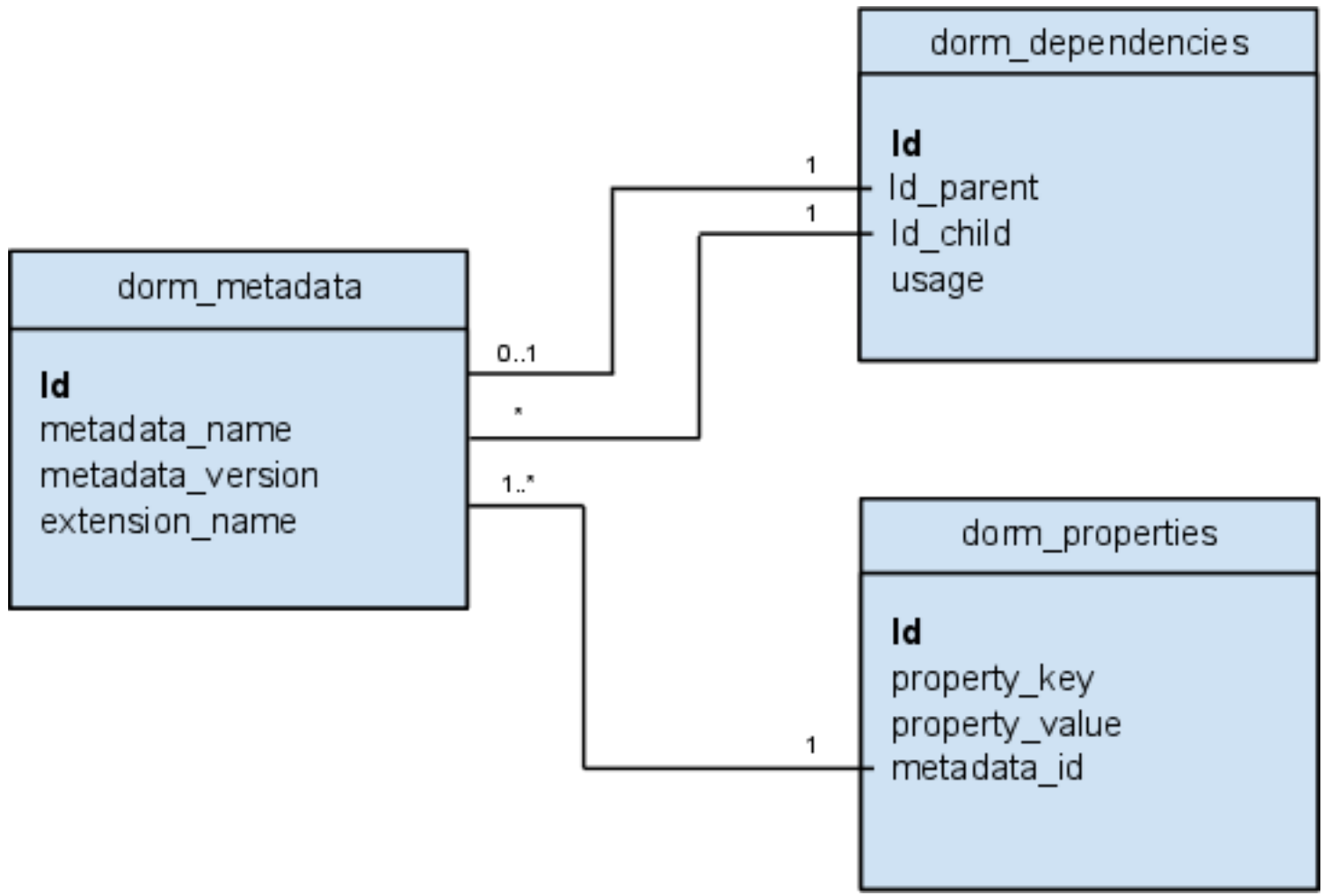
Two methods services are implement :

- saveMetadata witch take a DormMetadata.
- getByQualifier witch take a Qualifier and return a DormMetadata.
- addDependenciesToNode witch take a DependencyNode.

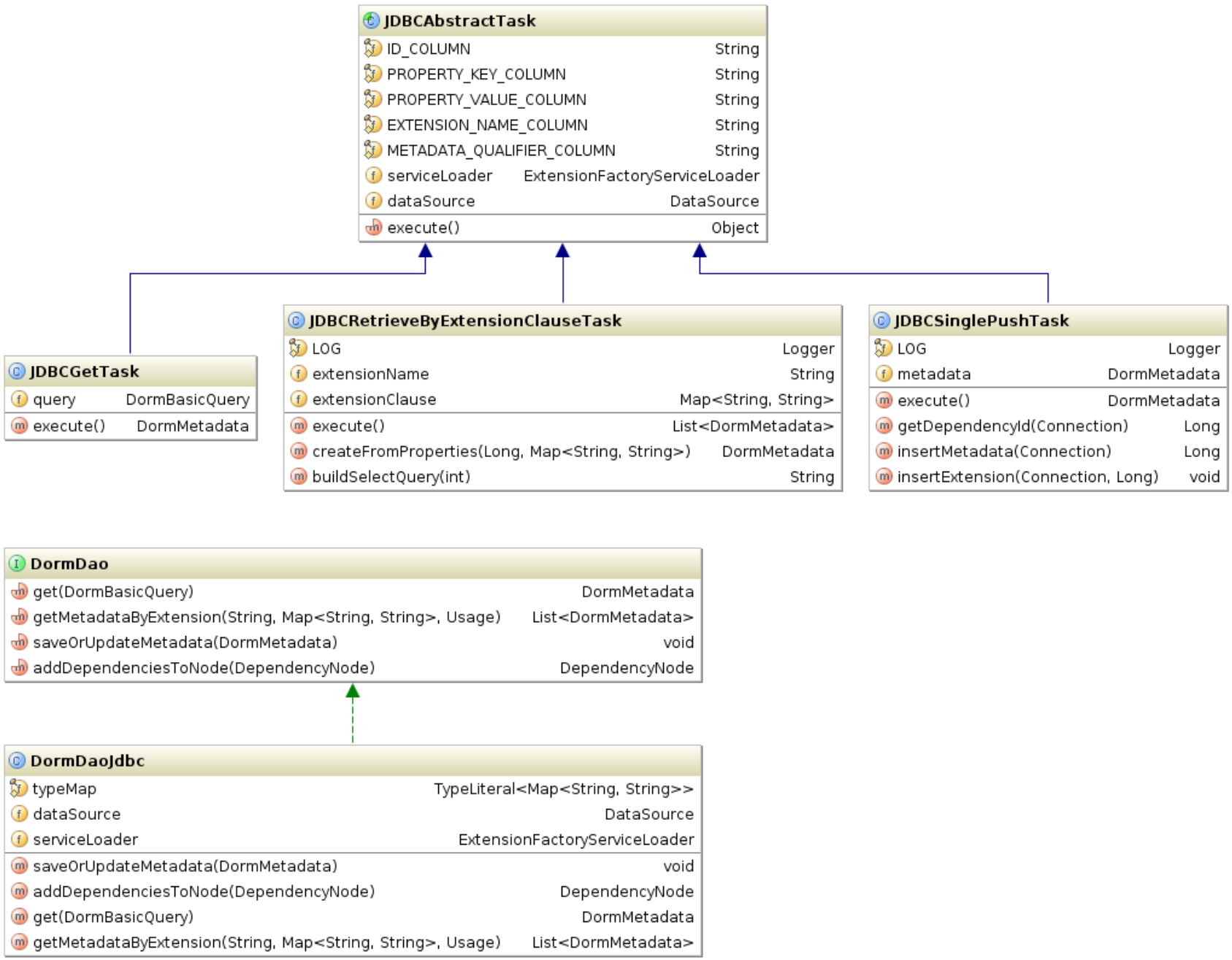
Dorm Dao internal structure



JDBC Dao database shema



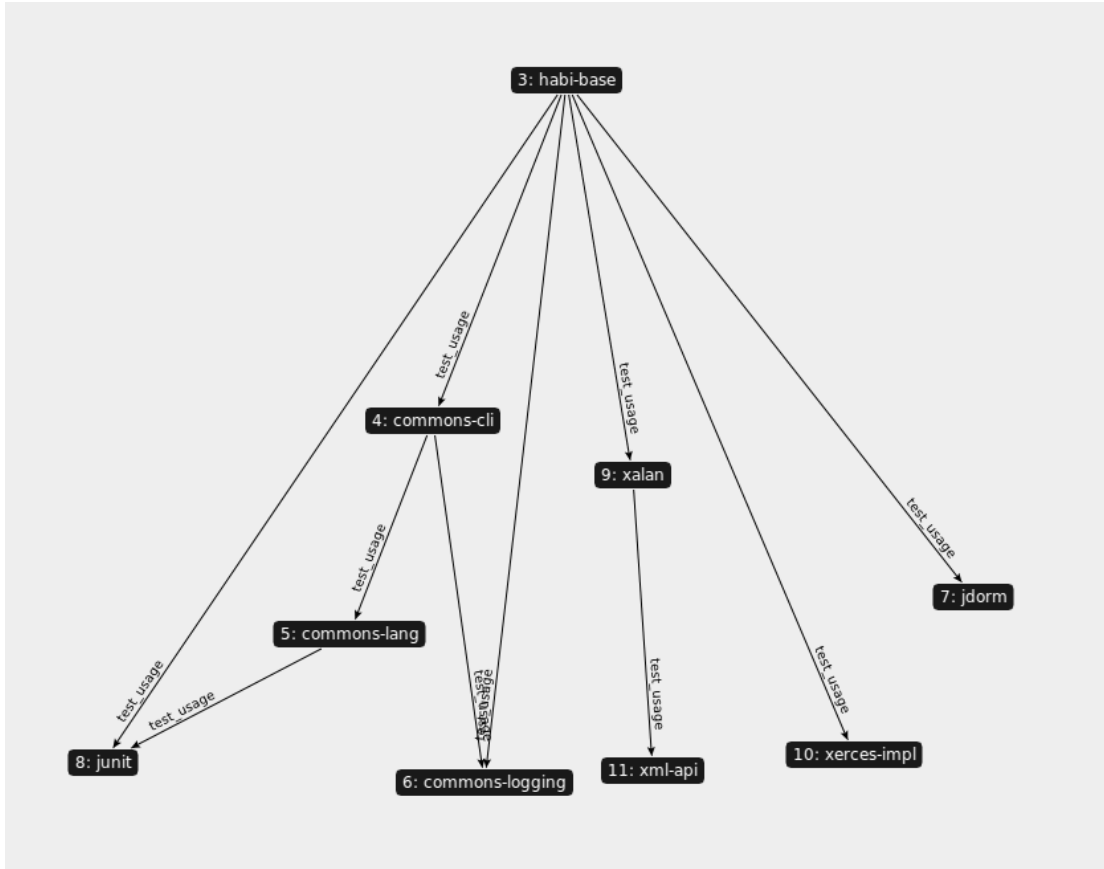
JDBC Dao diagram



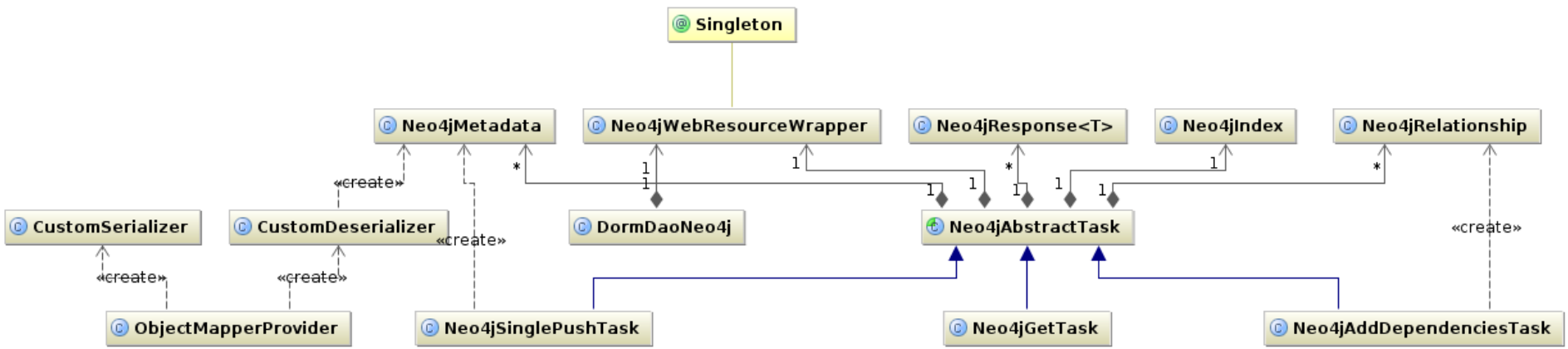
Neo4j Dao

- Graph database
- Call with RESTfull API
- One node per DormMetadata

Graph example:

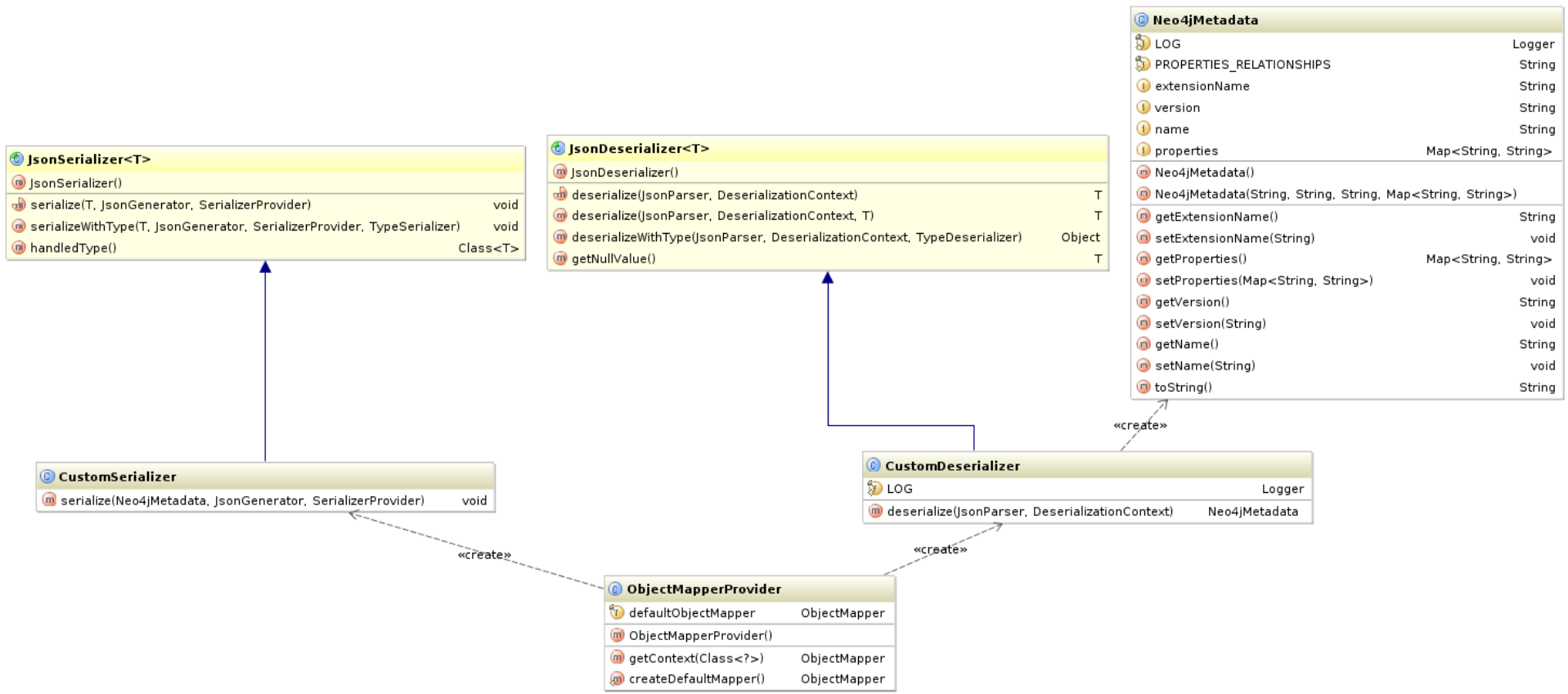


Neo4j Dao diagram



Neo4j Dao Diagram

Jackson (de)serialiser customization



Neo4j Dao Diagram

Dao methods implementation

Nuxeo Dao

- Document Database
- Call with RESTfull web service

Nuxeo Dao diagram