



Integrating an EA Smell Catalog into a Knowledge-Graph-Based Detection Tool

Armond Alexanian
Software Engineering & Internet Computing
e0728633@student.mail.tuwien.ac.at



Problem statement

We have

- EA smells catalog (Salentin & Hacks 2020)
- Prototype for detecting EA smells (Salentin & Hacks 2020)
- Knowledge Graphs to Detect EA Smells (Smajevic, Hacks & Bork 2021)

So far no detection and automation provided for all EA smells

Main idea

- Provide cypher queries for detecting all EA Smells
- Extend and modify EA smell attributes from catalog
 - Cypher query
 - Query options (threshold)
 - Example image for each EA smell
- Integrate queries into the CM2KG Platform
- A comprehensive smells detection automation inside the platform

EA smells catalog

Enterprise Architecture Smells
63/63

ⓘ This website serves as a knowledge base for Enterprise Architecture Smells.

Category

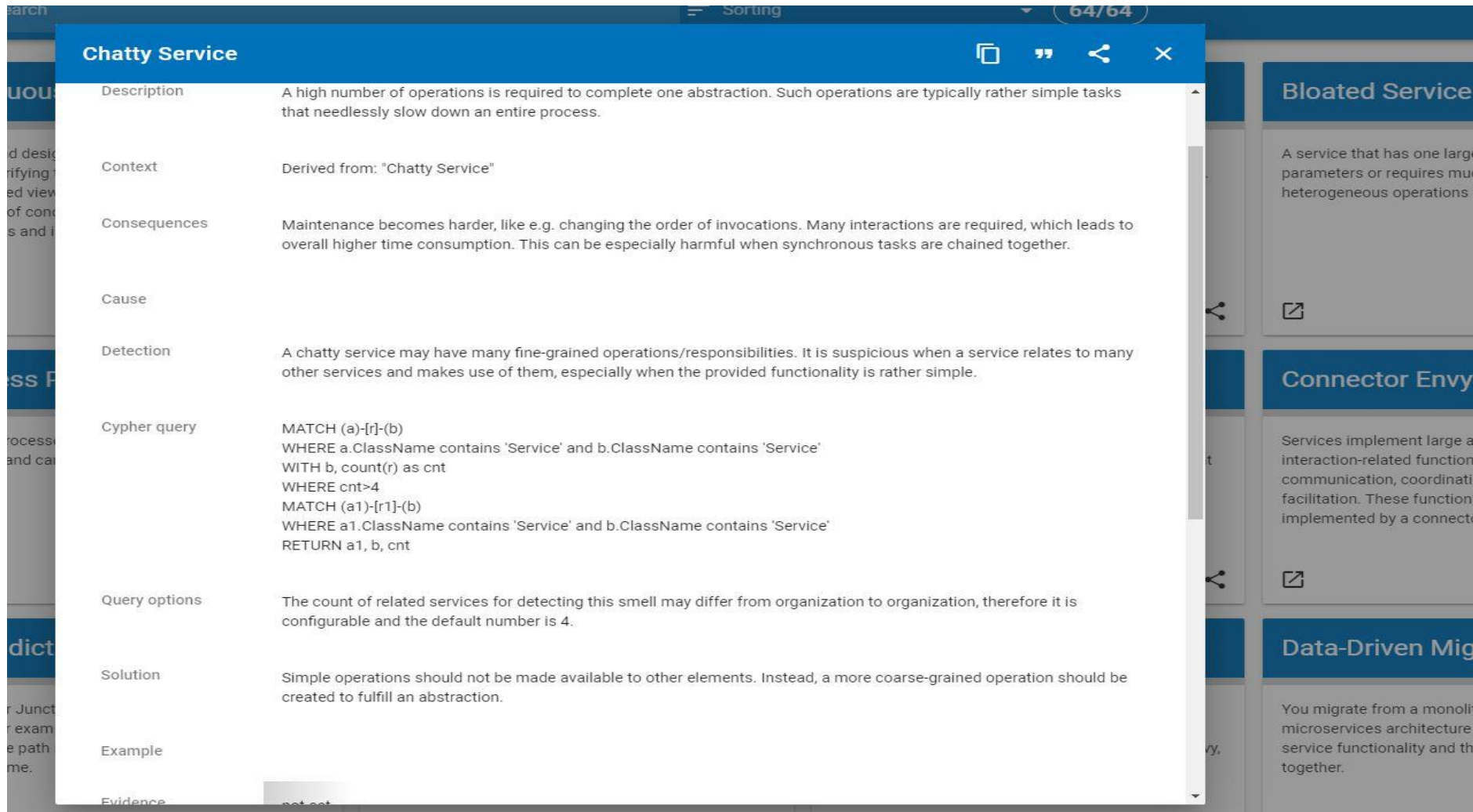
- technology
- business
- application
- Understandability Problem
- Semantic Error
- Rule-related defects
- Data-flow related defects
- Control-flow Problems

Tags

- All
- Process
- The Bloaters

<h3 style="background-color: #0072bc; color: white; padding: 5px;">Ambiguous Viewpoint</h3> <p>Analysis and design models are often presented without clarifying the viewpoint represented by the model. Mixed viewpoints don't allow the fundamental separation of concerns, confusing blueprints of abstractions and implementation details.</p> <div style="text-align: right; font-size: 0.8em;"> </div>	<h3 style="background-color: #0072bc; color: white; padding: 5px;">Architecture by Implication</h3> <p>This antipattern is characterized by the lack of architecture specifications for a system under development. Usually, the architects responsible for the project have experience with previous system construction, and therefore assume that documentation is unnecessary.</p> <div style="text-align: right; font-size: 0.8em;"> </div>	<h3 style="background-color: #0072bc; color: white; padding: 5px;">Big Bang</h3> <p>A strategy often preferred by large vendors where an entire Enterprise Architecture Model is built "at once".</p> <div style="text-align: right; font-size: 0.8em;"> </div>	<h3 style="background-color: #0072bc; color: white; padding: 5px;">Bloated Service</h3> <p>A service that has one large interface with many parameters or requires much data and performs mostly heterogeneous operations with low cohesion.</p> <div style="text-align: right; font-size: 0.8em;"> </div>
<h3 style="background-color: #0072bc; color: white; padding: 5px;">Business Process Forever</h3> <p>Business processes have been strictly defined and are now static and cannot be easily changed.</p> <div style="text-align: right; font-size: 0.8em;"> </div>	<h3 style="background-color: #0072bc; color: white; padding: 5px;">Chatty Service</h3> <p>A high number of operations is required to complete one abstraction. Such operations are typically rather simple tasks that needlessly slow down an entire process.</p> <div style="text-align: right; font-size: 0.8em;"> </div>	<h3 style="background-color: #0072bc; color: white; padding: 5px;">Combinatorial Explosion</h3> <p>A subtle form of Duplication, this smell exists when numerous elements do the same thing using different combinations of data or behavior.</p> <div style="text-align: right; font-size: 0.8em;"> </div>	<h3 style="background-color: #0072bc; color: white; padding: 5px;">Connector Envy</h3> <p>Services implement large amounts of low-level interaction-related functionality, e.g. for communication, coordination, conversation, or facilitation. These functionalities should be implemented by a connector instead.</p> <div style="text-align: right; font-size: 0.8em;"> </div>

EA smells detail view extension



The screenshot displays a software interface with a modal window titled "Chatty Service". The modal window contains the following details:

- Description:** A high number of operations is required to complete one abstraction. Such operations are typically rather simple tasks that needlessly slow down an entire process.
- Context:** Derived from: "Chatty Service"
- Consequences:** Maintenance becomes harder, like e.g. changing the order of invocations. Many interactions are required, which leads to overall higher time consumption. This can be especially harmful when synchronous tasks are chained together.
- Cause:**
- Detection:** A chatty service may have many fine-grained operations/responsibilities. It is suspicious when a service relates to many other services and makes use of them, especially when the provided functionality is rather simple.
- Cypher query:**

```

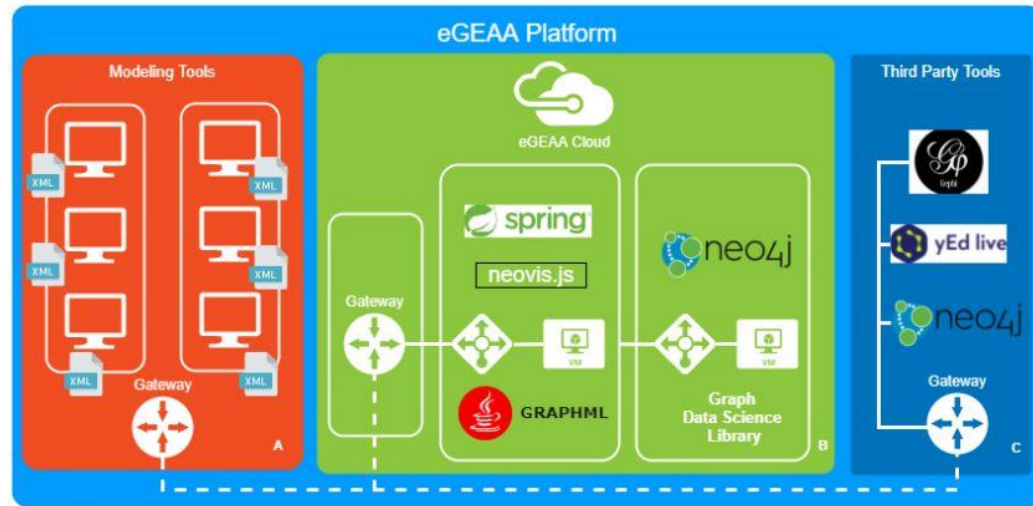
MATCH (a)-[r]-(b)
WHERE a.ClassName contains 'Service' and b.ClassName contains 'Service'
WITH b, count(r) as cnt
WHERE cnt>4
MATCH (a1)-[r1]-(b)
WHERE a1.ClassName contains 'Service' and b.ClassName contains 'Service'
RETURN a1, b, cnt

```
- Query options:** The count of related services for detecting this smell may differ from organization to organization, therefore it is configurable and the default number is 4.
- Solution:** Simple operations should not be made available to other elements. Instead, a more coarse-grained operation should be created to fulfill an abstraction.
- Example:**
- Evidence:**

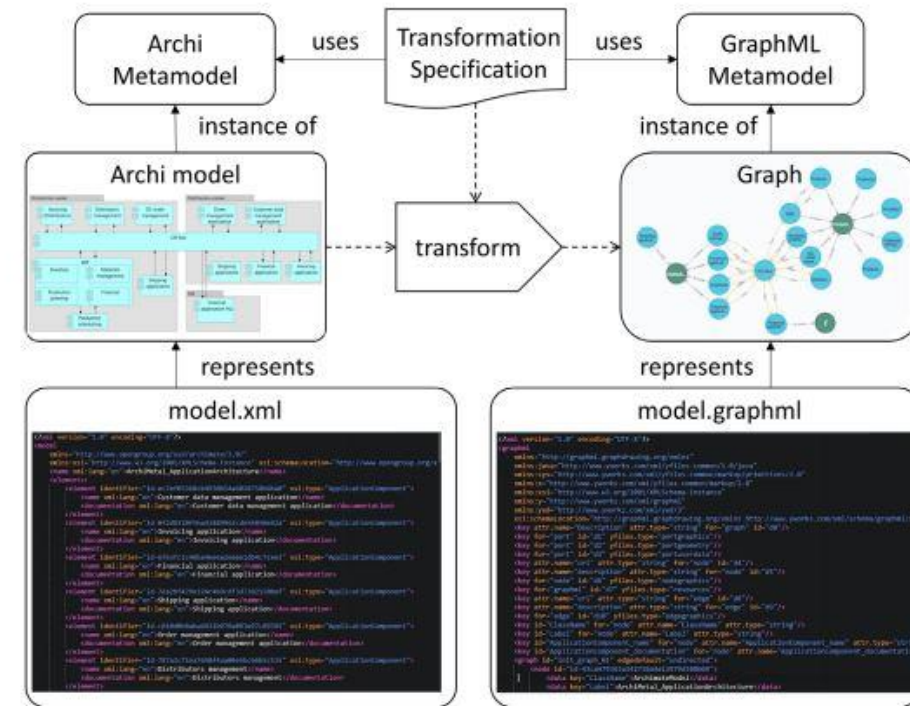
In the background, a list of other EA smells is visible, including "Bloated Service", "Connector Envy", and "Data-Driven Migration".

CM2KG platform

Architecture



Transformation





CM2KG platform

Knowledge Graph

CM2KG Conceptual Model to Knowledge Graph

Transform your conceptual model (Archi, ADOxx, Papyrus ...) to GraphML and gain insights about your model by apply graph based-analysis.

Step 1

Upload your model.

Your uploaded content.

```
<?xml version="1.0" encoding="UTF-8"?>
<model xmlns="http://www.opengroup.org/xsd/archimate/3.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
<name xml:lang="en">Archisurance</name>
<documentation xml:lang="en">An example of a fictional Insurance company.</documentation>
<elements>
<element identifier="id-1544" xsi:type="BusinessInterface">
<name xml:lang="en">mail</name>
</element>
<element identifier="id-1540" xsi:type="BusinessInterface">
<name xml:lang="en">phone</name>
</element>
<element identifier="id-1542" xsi:type="BusinessInterface">
<name xml:lang="en">GIM</name>
</element>
<element identifier="id-1538" xsi:type="BusinessInterface">
<name xml:lang="en">e-mail</name>
</element>
<element identifier="id-1536" xsi:type="BusinessInterface">
<name xml:lang="en">phone</name>
</element>
<element identifier="id-528" xsi:type="BusinessRole">
<name xml:lang="en">Customer's Bank</name>
</element>
<element identifier="id-521" xsi:type="BusinessRole">
<name xml:lang="en">Customer</name>
</element>
<element identifier="id-507" xsi:type="BusinessRole">
```

Step 2

Your model is converted..

Step 3

Take your GraphML model.

Step 4

Gain insight about your model.

Your transformed content. [Initialize Neo4j](#) [Graph Show](#) [GraphML xml](#) [Download RDF OWL](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:java="http://www.yworks.com/xml/yfiles-common/1.0/java"
xmlns:sys="http://www.yworks.com/xml/yfiles-common/markup/primitives/2.0"
xmlns:x="http://www.yworks.com/xml/yfiles-common/markup/2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:y="http://www.yworks.com/xml/graphml"
xmlns:yed="http://www.yworks.com/xml/yed/3"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns http://www.yworks.com/xml/schema/graphml/1.1/ygraphml.xsd">
<key attr.name="Description" attr.type="string" for="graph" id="d0"/>
<key for="port" id="d1" yfiles.type="portgraphics"/>
<key for="port" id="d2" yfiles.type="portgeometry"/>
<key for="port" id="d3" yfiles.type="portuserdata"/>
<key attr.name="url" attr.type="string" for="node" id="d4"/>
<key attr.name="description" attr.type="string" for="node" id="d5"/>
<key for="node" id="d6" yfiles.type="nodegraphics"/>
<key for="graphml" id="d7" yfiles.type="resources"/>
<key attr.name="url" attr.type="string" for="edge" id="d8"/>
<key attr.name="description" attr.type="string" for="edge" id="d9"/>
<key for="edge" id="d10" yfiles.type="edgegraphics"/>
<key id="ClassName" for="node" attr.name="ClassName" attr.type="string"/>
<key id="Label" for="node" attr.name="Label" attr.type="string"/>
<key id="name" for="node" attr.name="name" attr.type="string"/>
<key id="documentation" for="node" attr.name="documentation" attr.type="string"/>
<key id="ReferenceName" for="edge" attr.name="ReferenceName" attr.type="string"/>
<graph id="init_graph_01" edgedefault="directed">
<node id="id-838">
```

<https://github.com/borkdominik/CM2KG>

CM2KG platform

Visualization

Step 1

Upload your model.

Step 2

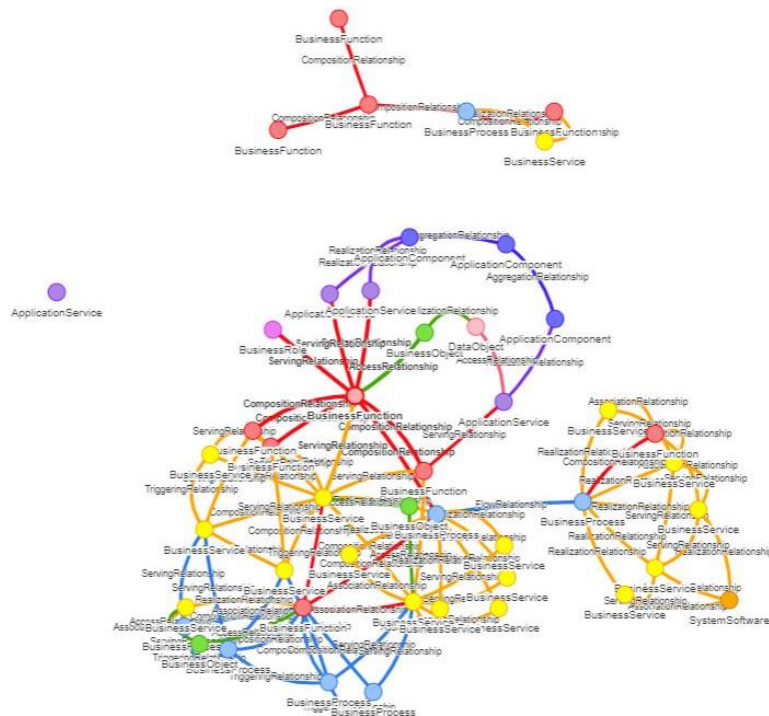
Your model is converted..

Step 3

Take your GraphML model.

Step 4

Gain insight about your model.



Caption

Size

Community

Font size

Font color

Cypher query:

```
MATCH (n)-[r]->(m), (a)
where not (a)--()
RETURN n,r,m,a
```

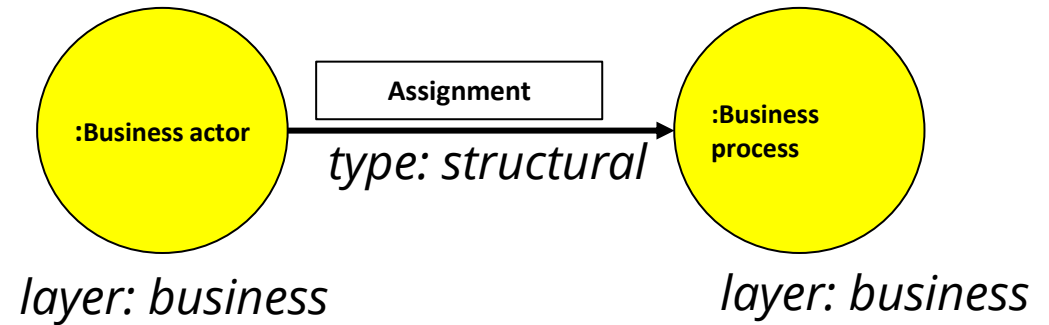

Neo4j database

Node (vertex)

- Label
- Properties

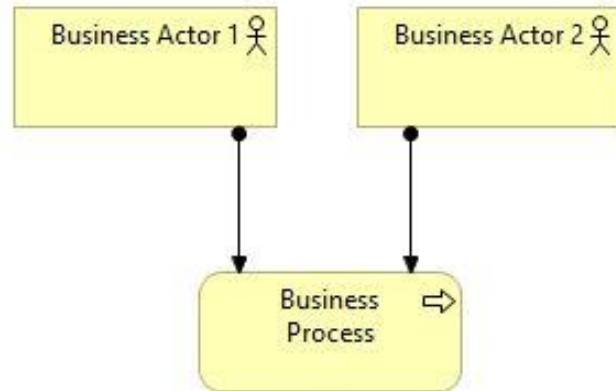
Relationship (edge)

- Direction
- Type
- Properties



Multifaceted abstraction

ArchiMate



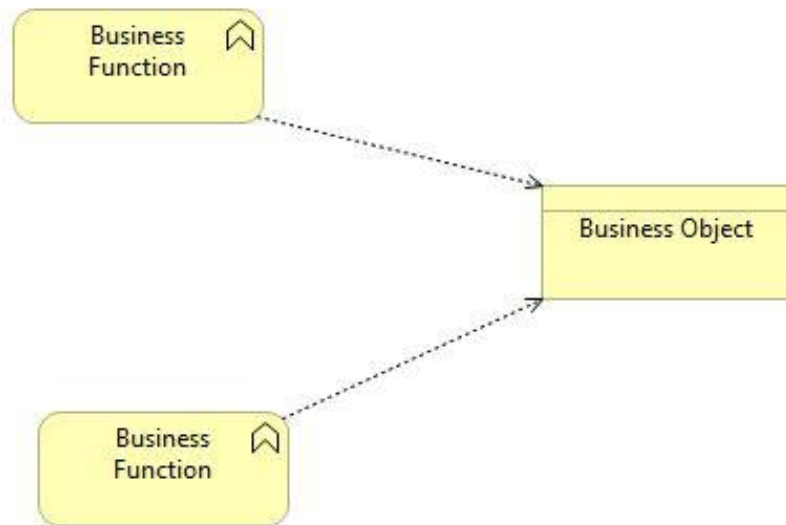
Cypher query

```

MATCH (m)-[r:AssignmentRelationship]-(n)
WITH n, COUNT(r) as rCount
MATCH (n)-[r1:AssignmentRelationship]-(m)
WHERE rCount>1
RETURN m,n
  
```

Deficient encapsulation

ArchiMate



Cypher query

```

MATCH (n)

WHERE n.ClassName = 'DataObject' OR
n.ClassName='BusinessObject'

UNWIND keys(n) AS nkeys

WITH nkeys, n

WHERE ANY (regex IN ['confident', 'classified',
'sensible'] WHERE n[nkeys] CONTAINS regex)

WITH n

MATCH (m)-[r:AccessRelationship]-(n)

WITH n ,COUNT (r) AS accessCount

MATCH (n)--(m)

WHERE accessCount > 1

RETURN n ,m
  
```



Current status

- Cypher queries for 20 EA smells
- Extended EA smells catalog

- Next steps:
 - Provide cypher queries for all EA smells (totally 63 smells)
 - Include all necessary details in the catalog
 - Integrate queries into the CM2KG platform



Questions

- Is detection serving its purpose in industry?
- Expectations about threshold (in catalog, application options)?
- Suggestions for evaluation



Thank you!