

iStarJSON: A Lightweight Data-Format for i* Models



Oscar Franco-Bedoya
ohernan@essi.upc.edu



David Ameller
dameller@essi.upc.edu



Dolors Costal
dolors@essi.upc.edu



Lidia López
llopez@essi.upc.edu

iStar 2016 Beijing, China



Software and Service Engineering Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Agenda

- 1 Motivation
- 2 iStarJSON Language
- 3 iStarJSON Services
- 4 IStarJSON PoC
- 5 Conclusions & Future Work

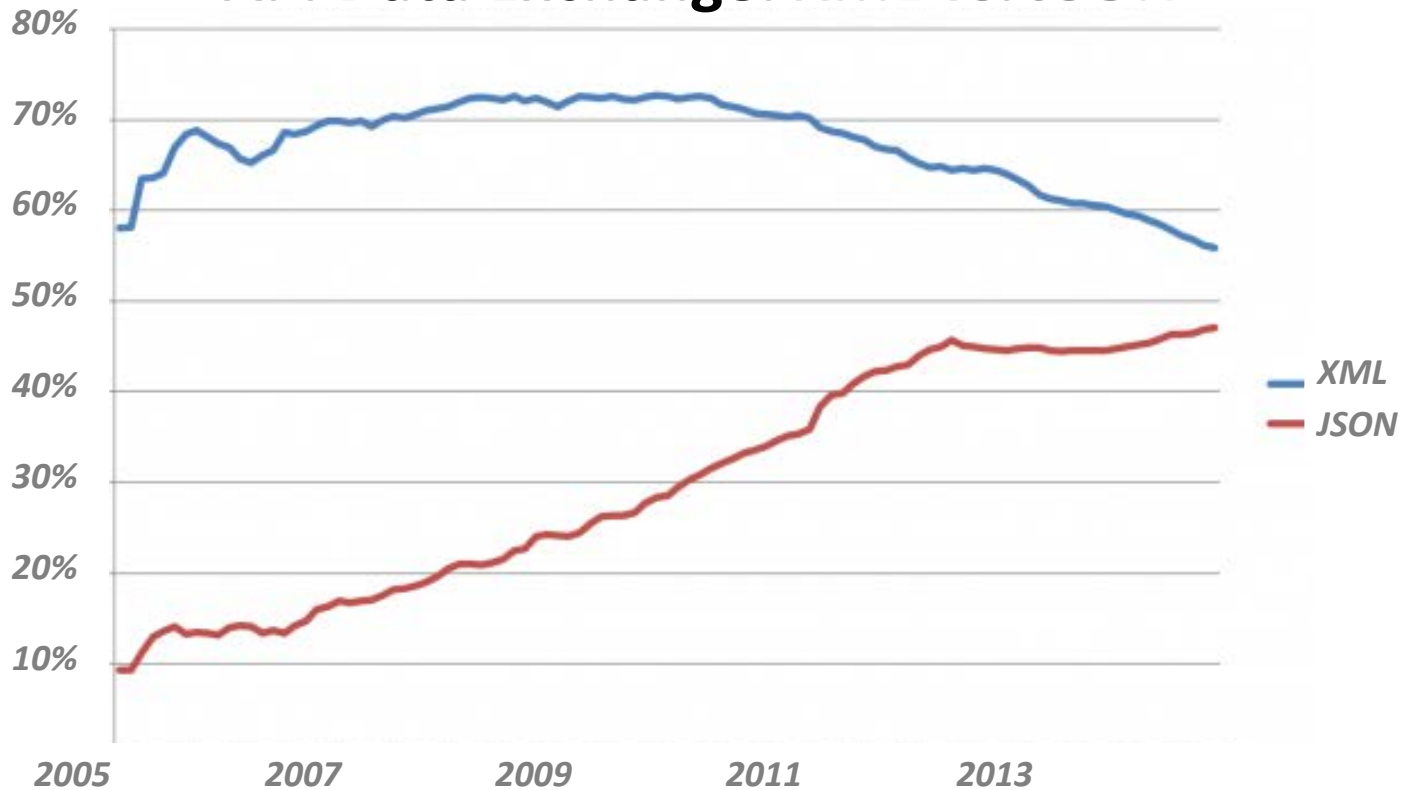
Motivation

iStarJSON has been carried out in the context of a general framework for **Open Source Software ECOsystems** (OSSECOs) quality analysis and modelling process.

*i** is used to represent OSSECOs, and the framework uses **model analysis for OSSECO quality analysis** (e.g., identifying main goals, members relationships, strategic dependencies)

JSON trend is growing

API Data Exchange: XML vs. JSON

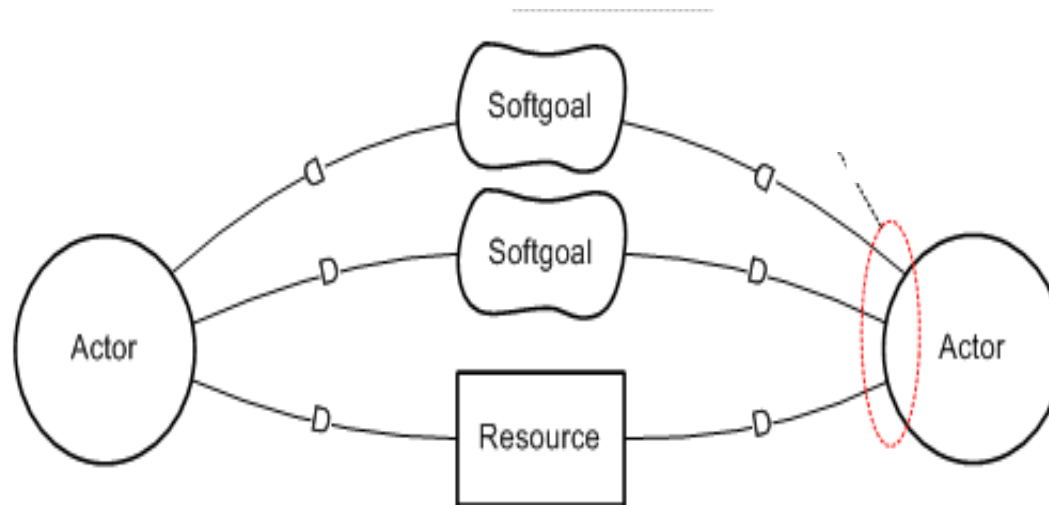


Source: Programmable Web Site
Over more than 16.000 APIs



ISTARJSON LANGUAGE

An iStar model is a directed graph



Source: iStar Wiki

$G = (V; E)$

$V = \{actor; agent; role; position\} \cup \{goal; task; resource; softgoal; belief\}$

$Atype = \{association; dependency; means_end; decomposition; contribution\}$

Dependencies are represented by three nodes (depender, dependee and dependum) and two links, one from the depender to the dependum and other from the dependum to the dependee.

iStarJSON

from **iStarML hierarchical** structure to **iStarJSON graph structure**

Graph structure **does not affect to the *i**** syntax or semantics

Graph structure allow **using existing tools and algorithms** for graph manipulation (e.g., graph layouts, node centrality, shortest path)

iStarJSON (partial) Schema

```
{
  "$schema": "http://testoneosseco.azurewebsites.net/json/schemas/istar/schema#",
  "properties": {
    "diagram": {"type": "string", "name": "diagram"},
    "modelType": {"type": "string", "enum": ["rationale", "dependence"]}
  },
  "nodes": {
    "items": {"properties": {"id": {"name": "id"}, "name": {"name": "name"},
      "elementType": {"enum": ["actor", "goal", "task", "resource", "softgoal",
        "belief"], "name": "elementType"}, "boundary": {"name": "boundary"}}}}
  },
  "edges": {
    "items": {"properties": {"source": {"name": "source"}, "target": {"name":
      "target"}, "linkType": {"enum": ["association", "dependency",
        "means_end", "decomposition", "contribution"]}}}}
  }
}
```

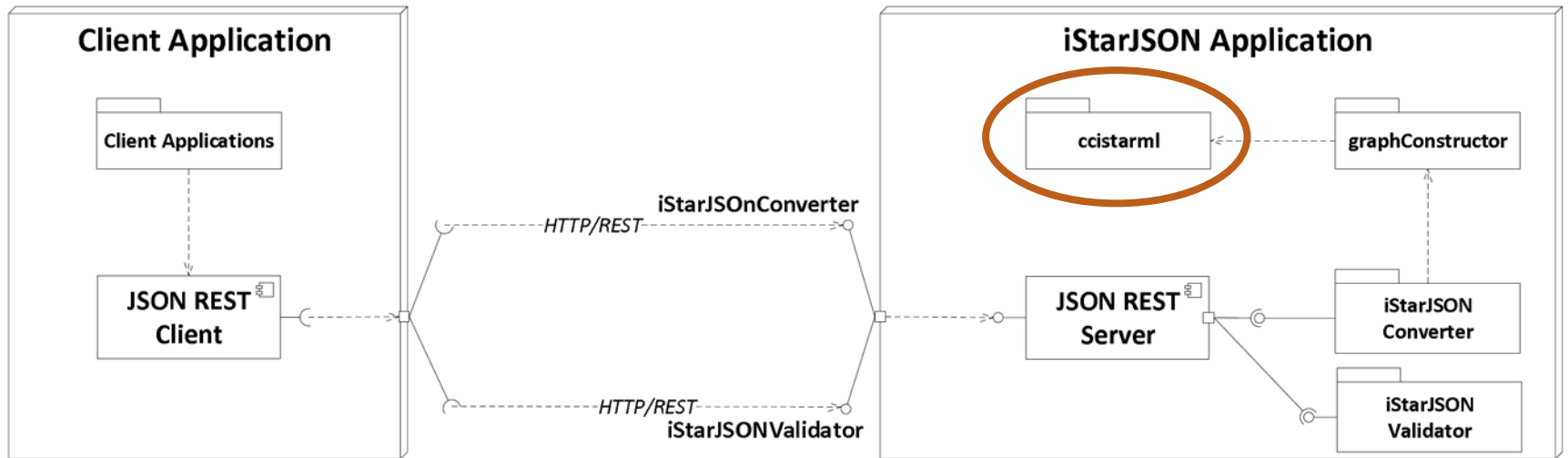

iStarJSON (partial) Example

```
"nodes": [  
  {  
    "boundary": "none",  
    "elementType":  
"softgoal",  
    "name": "Variety",  
    "id": "01"  
  },  
  {  
    "boundary": "none",  
    "elementType":  
"actor",  
    "name": "Customer",  
    "id": "02"  
  },  
  {  
    "boundary":  
"Customer",  
    "elementType":  
"goal",  
    "name": "Have  
software",  
    "id": "03"  
  },  
  .....  
]  
  
"edges": [  
  {  
    "linktype":  
"dependency",  
    "source": "02",  
    "target": "01",  
    "linksubtype": ""  
  },  
  {  
    "linktype":  
"dependency",  
    "source": "01",  
    "target": "09",  
    "linksubtype": ""  
  },  
  {  
    "linktype":  
"decomposition",  
    "source": "04",  
    "target": "03",  
    "linksubtype": "or"  
  },  
  .....  
]
```



ISTARJSON OSS PROJECT

iStarJSON Project Architecture



iStarJSON project in Github



github.com/UPC-gessi-oscar-franco/iStarJson

mail UGDSI ☆ Advanced ☆ 19 Museos Literarios ☆ Home - HeartMath ☆ Más información Javadoc Guidelines - Importado de

iStarJSON

iStarJSON: A Lightweight Data-Format for i* Models

Installation

TODO: Describe the installation process

Usage

iStar to Json file converter

```
curl -v -H "Content-Type: application/xml" -X POST --data @Test2.istarm1 http://testoneosseco.azurewebsites.net/iStar
```

Json Validator

<https://github.com/UPC-gessi-oscar-franco/iStarJson>



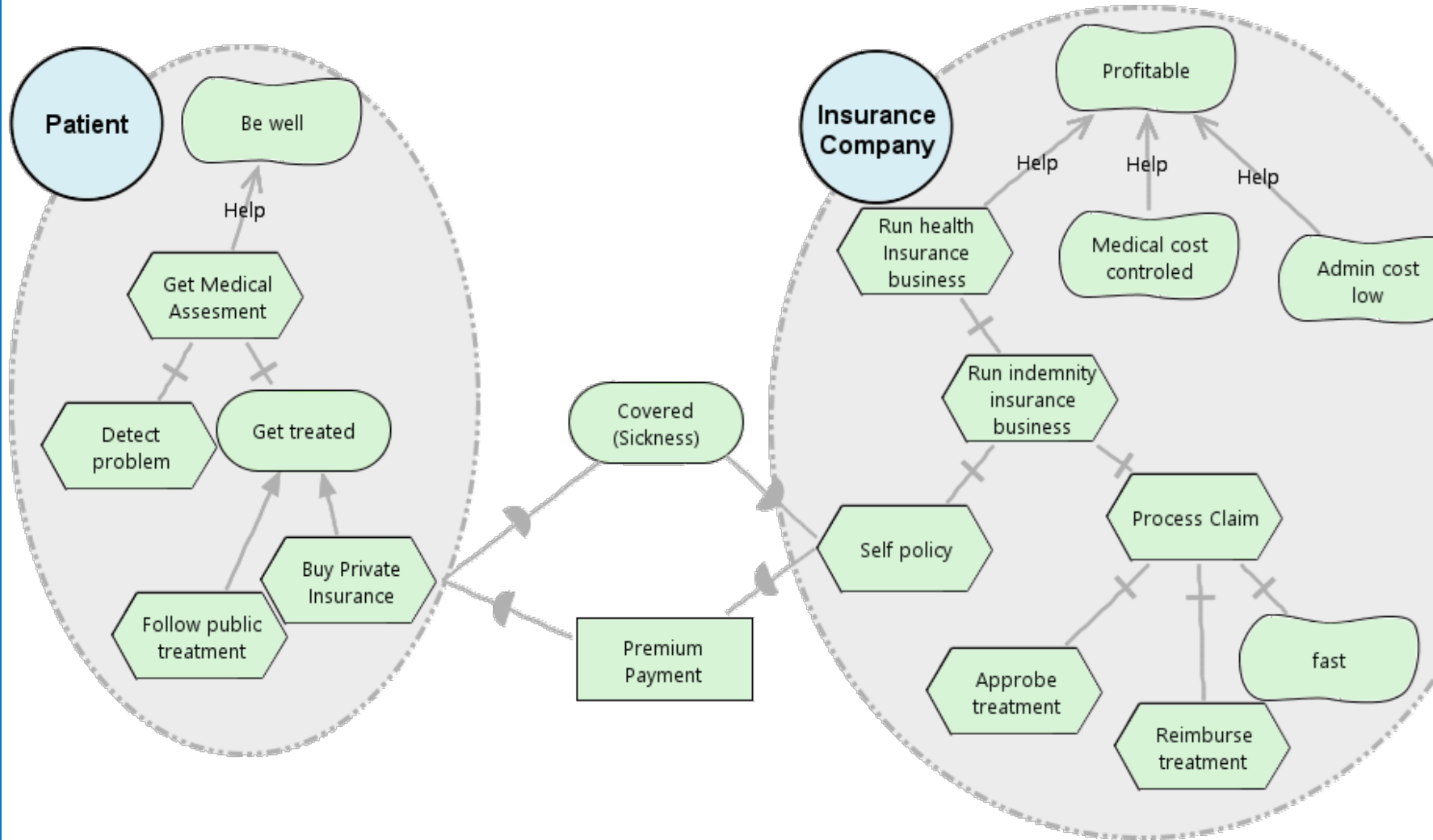


ISTARJSON PoC

iStarJSON REST Web Services

- 1 iStarML to iStarJSON converter
- 2 iStarJSON Validator
- 3 iStarJSON Graph Generator

i* Model for Testing



iStarML to iStarJSON Converter

```
<?xml version="1.0" encoding="UTF-8"?>
<istarml version="1.0">
  <diagram name="Test2.ood">
    <graphic content="SVG"/>
    <ielement type="resource" id="01" name="Premium Payment">
      <graphic content="basic" xpos="0" ypos="0" width="120" height="55" unit="pt"
        bgcolor="FFFFFF"
        fontfamily="Segoe UI"
        fontcolor="000000"
        fontsize="12"/>
      <dependency>
        <depender iref="02" aref="03">
          <graphic content="SVG"/>
        </depender>
        <dependee iref="04" aref="05">
          <graphic content="SVG"/>
        </dependee>
      </dependency>
    </ielement>
    <ielement type="goal" id="_KwQGIPDhEeW7du-QYVGhw" name="Covered (Sickness)">
      <graphic content="basic" xpos="0" ypos="0" width="120" height="55" unit="pt"
        bgcolor="FFFFFF"
        fontfamily="Segoe UI"
        fontcolor="000000"
        fontsize="12"/>
      <dependency>
        <depender iref="04" aref="05">
          <graphic content="SVG"/>
        </dependee>
      </dependency>
    </ielement>
  </diagram>
</istarml>
```

```
curl -v -H "Content-Type: application/xml" -X POST --data @Test2.istarml
http://localhost:8080/iStarJSONServiceREST/istar/istarToJson > Test2.istarjson
```


iStarJSON Validator

```
{
  "diagram": "Test2.ood",
  "nodes": [
    {
      "boundary": "boundary",
      "elementType": "resource",
      "name": "Premium Payment",
      "id": "01"
    },
    {
      "boundary": "boundary",
      "elementType": "goal",
      "name": "Covered (Sickness)",
      "id": "_KwQGIPDhEeWC7du-QYVGhw"
    },
    {
      "boundary": "boundary",
      "elementType": "actor",
      "name": "Patient",
      "id": "05"
    },
    {
      "boundary": "Patient",
      "elementType": "goal",

```

```

    "target": "17",
    "linksubtype": "and"
  },
  {
    "linktype": "decomposition",
    "source": "16",
    "target": "18",
    "linksubtype": "and"
  },
  {
    "linktype": "decomposition",
    "source": "16",
    "target": "19",
    "linksubtype": "and"
  },
  {
    "linktype": "contribution",
    "source": "18",
    "target": "17",
    "linksubtype": "some"
  }
],
"modelType": "rationale"
}

```

```
curl -v -H "Content-Type: application/xml" -X POST --data @Test2.json
http://localhost:8080/iStarJSONServiceREST/istar/istarJSONvalidator > valid.txt
```

```
curl -v -H "Content-Type: application/xml" -X POST --data @Test2Bad.json
http://localhost:8080/iStarJSONServiceREST/istar/istarJSONvalidator > invalid.txt
```

iStarJSON Graph Generator

```
{
  "diagram": "Test2.ood",
  "nodes": [
    {
      "boundary": "boundary",
      "elementType": "resource",
      "name": "Premium Payment",
      "id": "01"
    },
    {
      "boundary": "boundary",
      "elementType": "goal",
      "name": "Covered (Sickness)",
      "id": "_KwQGIPDhEeWC7du-QYVGhw"
    },
    {
      "boundary": "boundary",
      "elementType": "actor",
      "name": "Patient",
      "id": "05"
    },
    {
      "boundary": "Patient",
      "elementType": "goal",

```

```
    "target": "17",
    "linksubtype": "and"
  },
  {
    "linktype": "decomposition",
    "source": "16",
    "target": "18",
    "linksubtype": "and"
  },
  {
    "linktype": "decomposition",
    "source": "16",
    "target": "19",
    "linksubtype": "and"
  },
  {
    "linktype": "contribution",
    "source": "18",
    "target": "17",
    "linksubtype": "some"
  }
],
"modelType": "rationale"
}
```

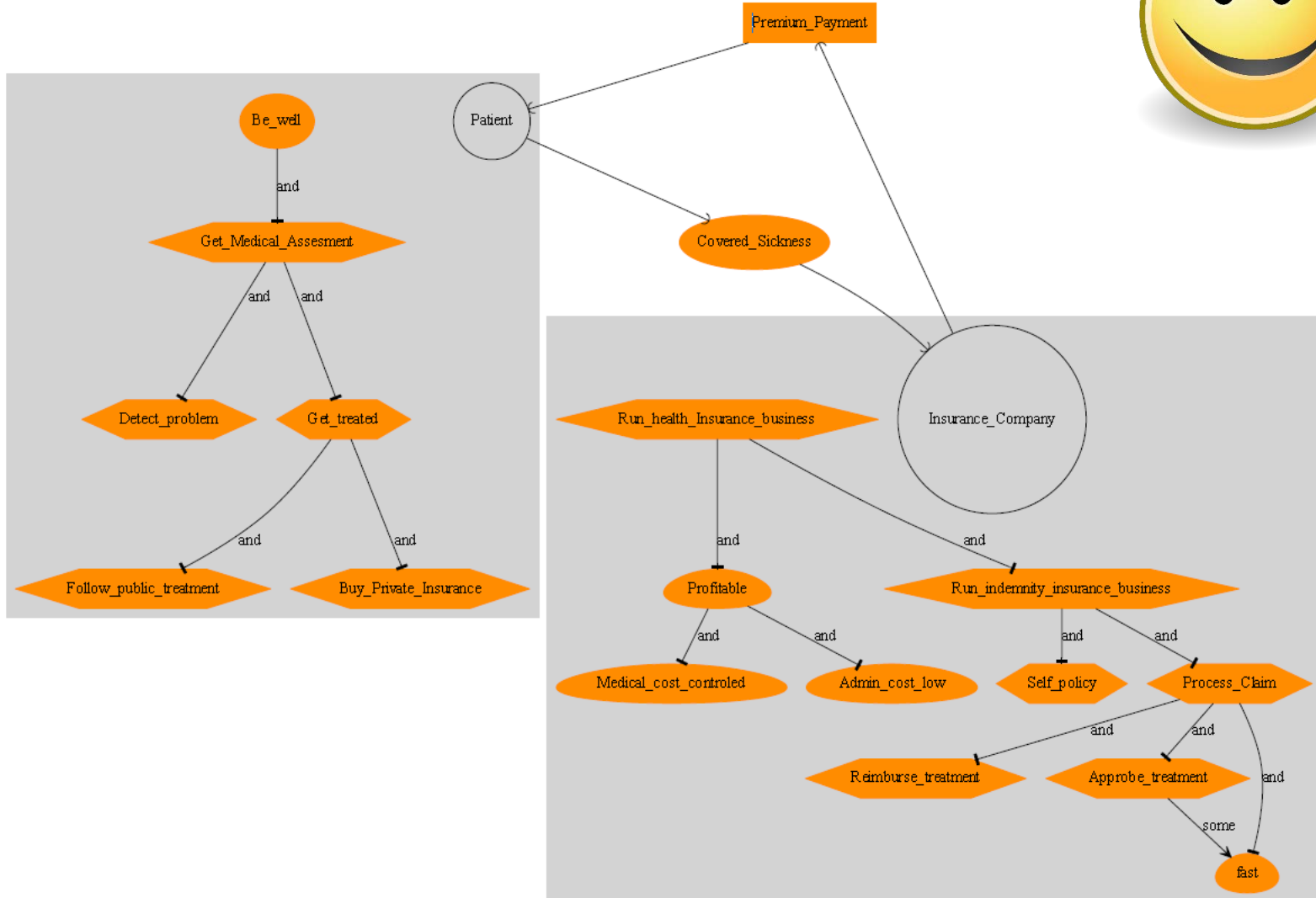


```
curl -v -H "Content-Type: application/xml" -X POST --data @Test2.json
"http://localhost:8080/iStarJSONServiceREST/istar/istarJSONconverter/param?lay
aout=dot&type=pdf" >grap.pdf
```

Graph Visualization

Generated from an iStarML file

Just in case...



iStarJSON on the cloud (AZURE)

- Converter

```
curl -v -H "Content-Type: application/xml" -X POST --data  
@Test2.istarml  
http://testoneosseco.azurewebsites.net/iStarJSONServiceREST  
/istar/istarToJSON
```

- Validator

```
curl -v -H "Content-Type: application/xml" -X POST --data  
@Test2.json  
http://testoneosseco.azurewebsites.net/iStarJSONServiceREST  
/istar/istarJSONvalidator
```

- Graph Generator: Coming soon!

CONCLUSIONS & FUTURE WORK

Conclusions

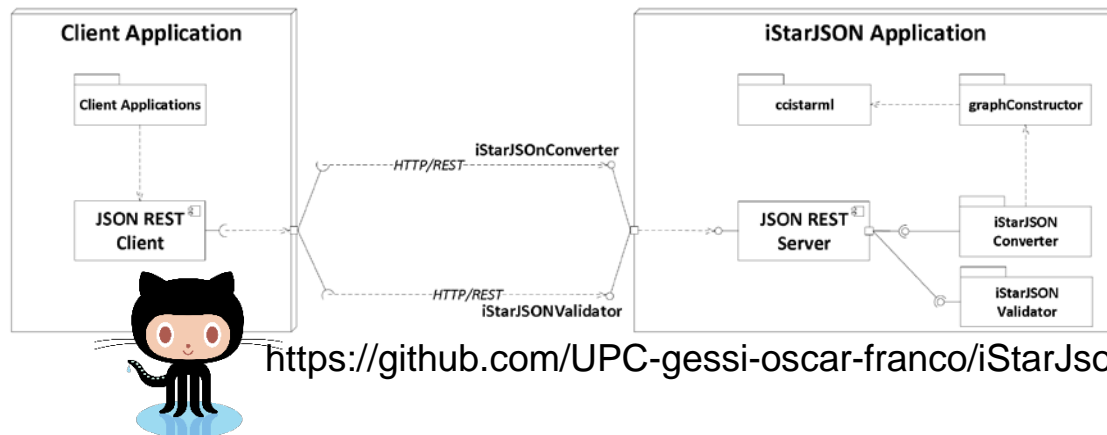
1 Definition of JSON language

```
{
  "$schema": "http://testoneossec.azurewebsites.net/json/schemas/istar/schema#",
  "properties": {
    "diagram": { "type": "string", "name": "diagram" },
    "modelType": { "type": "string", "enum": ["rationale", "dependence"] },
  },
  "nodes": {
    "items": { "properties": { "id": { "name": "id" }, "name": { "name": "name" },
      "elementType": { "enum": ["actor", "goal", "task", "resource", "softgoal", "belief"], "name": "elementType" }, "boundary": { "name": "boundary" } } } },
  "edges": {
    "items": { "properties": { "source": { "name": "source" }, "target": { "name": "target" },
      "linkType": { "enum": ["association", "dependency", "means_end", "decomposition", "contribution"] } } } } }
}
```

3 Provision of 3 REST Web Services

iStarML to iStarJSON converter
iStarJSON Validator
iStarJSON Graph Generator

2 Release iStarJSON as OSS project



<https://github.com/UPC-gessi-oscar-franco/iStarJson>



Future work

Adding more services for...

- ① *i** Model Analysis
- ② Social Network Analysis
- ③ Software Ecosystems



Thank you!

