NOn: Network Function Virtualisation Ontology Towards Semantic Service Implementation

Luis Cuellar Hoyos

MEng. Candidate

Prof. Dr. Christian Esteve Rothenberg

Advisor

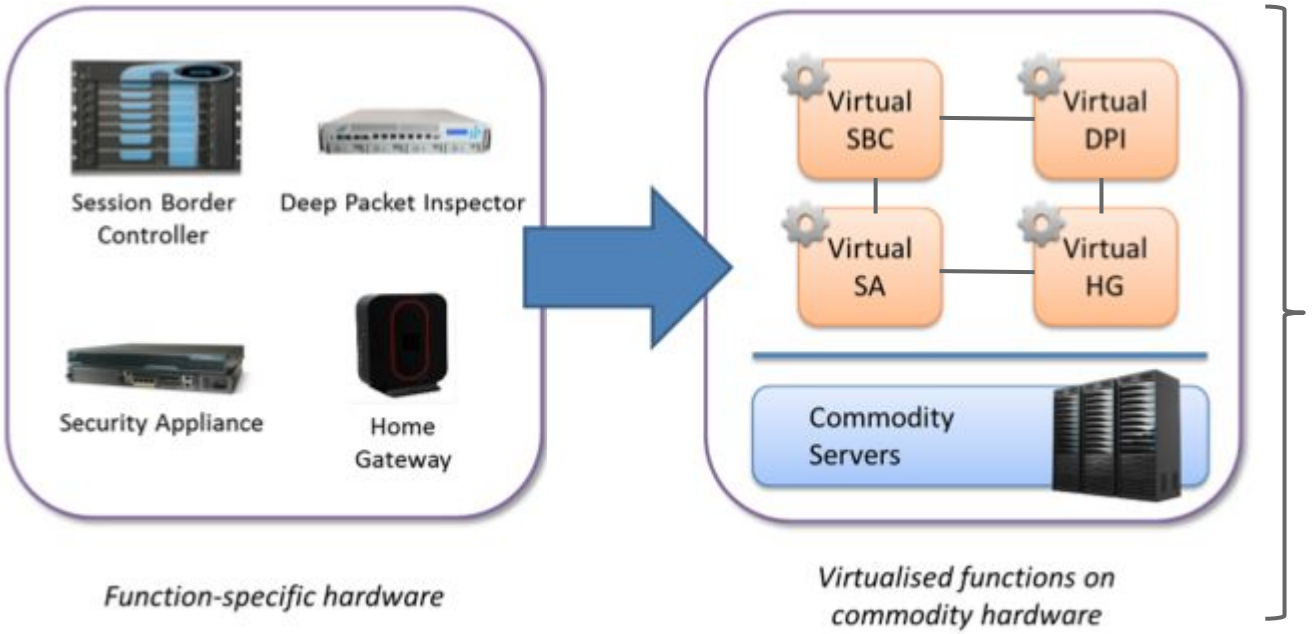
Agenda

- Introduction
- Problem Review
- Goals
- Background
- NFV Ontology (NOn)
  - Implementation & PoC
- Semantic nFV Services (SnS)
  - Implementation & PoC
- Conclusions and Future Work

1.
Introduction
A NFV Overview

Network Function Virtualization

To change the current appliance model of network functions, trying to remove the typical constraints

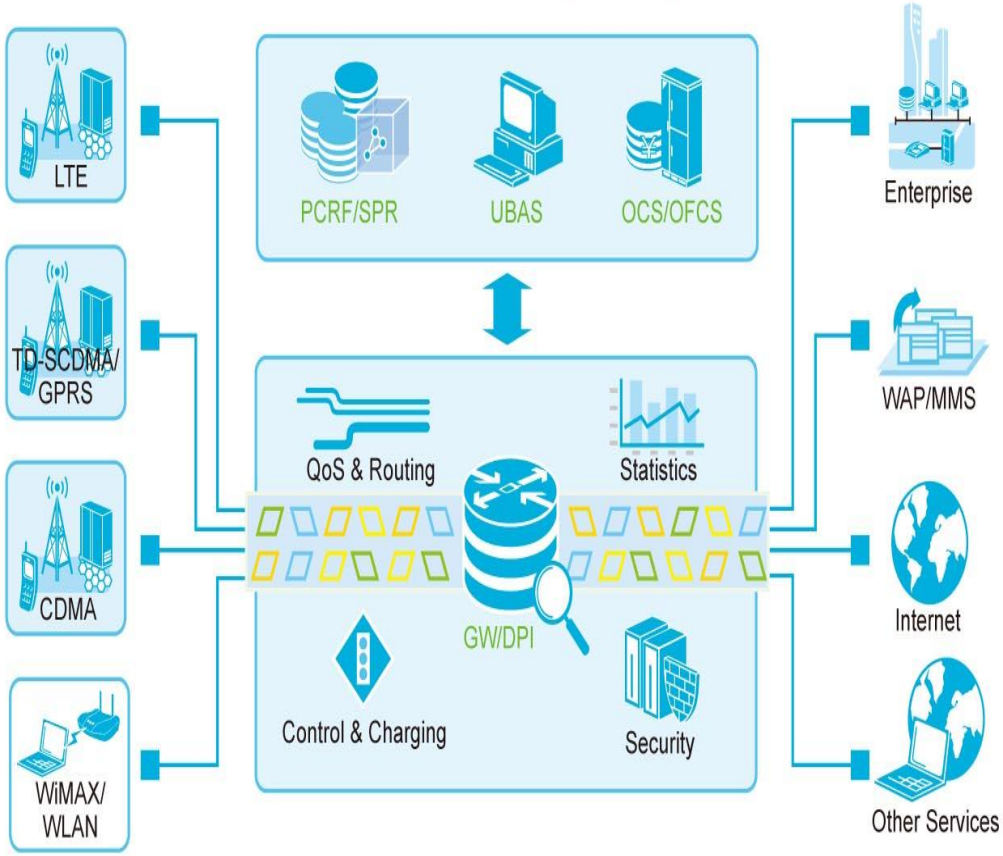


NFV Principal Components

VNF: Software Defined Function Networks

NFVI: Provide the resources necessaries (Physical and virtual) to deploy a VNF

MANO: in charge of orchestrate and manage all the aspects related to the NS and VNF deployment



NFV High Level Architecture

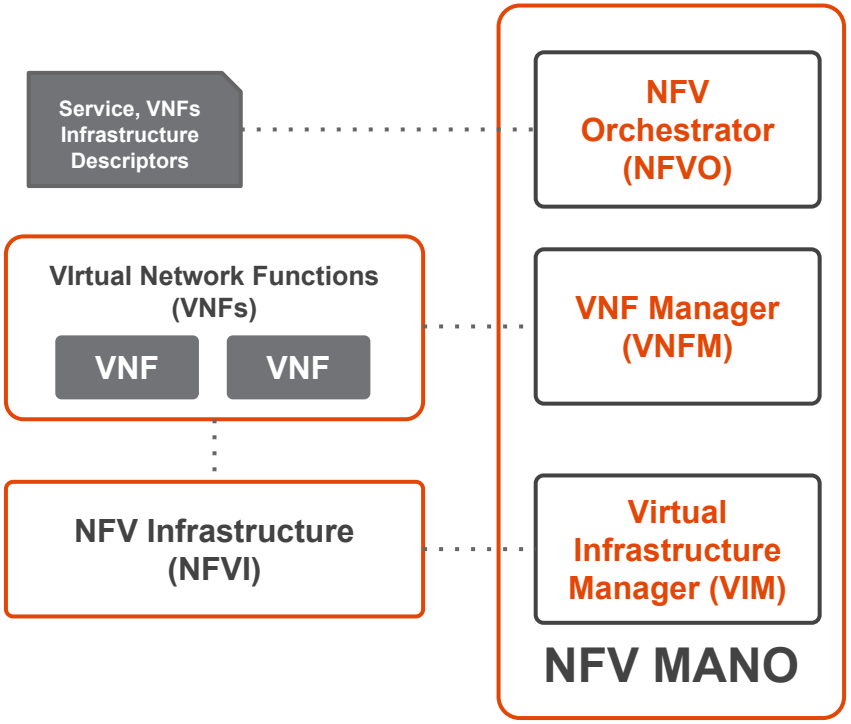
2.

Problem Review

The cost of Interoperability

NFV Implementation - Local Domain

- ➔ Different developers
- ➔ Different Technologies



- ➔ DIfferent Communication Protocols
- ➔ Different terminology

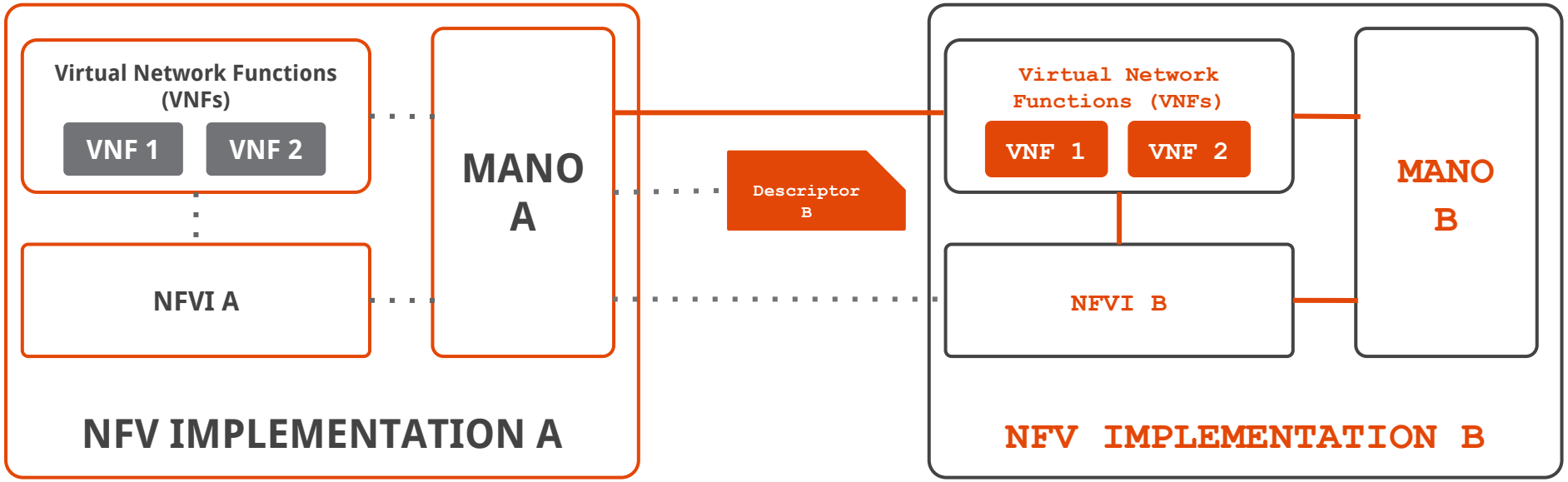
How to gain interoperability?

Integration Using Manual Intervention

Manual Intervention

- ➔ Read descriptions, user manuals and metadata
- ➔ Interpret and understand
- ➔ Use capabilities and functionalities

NFV Implementation - Inter Domain



How to gain interoperability?

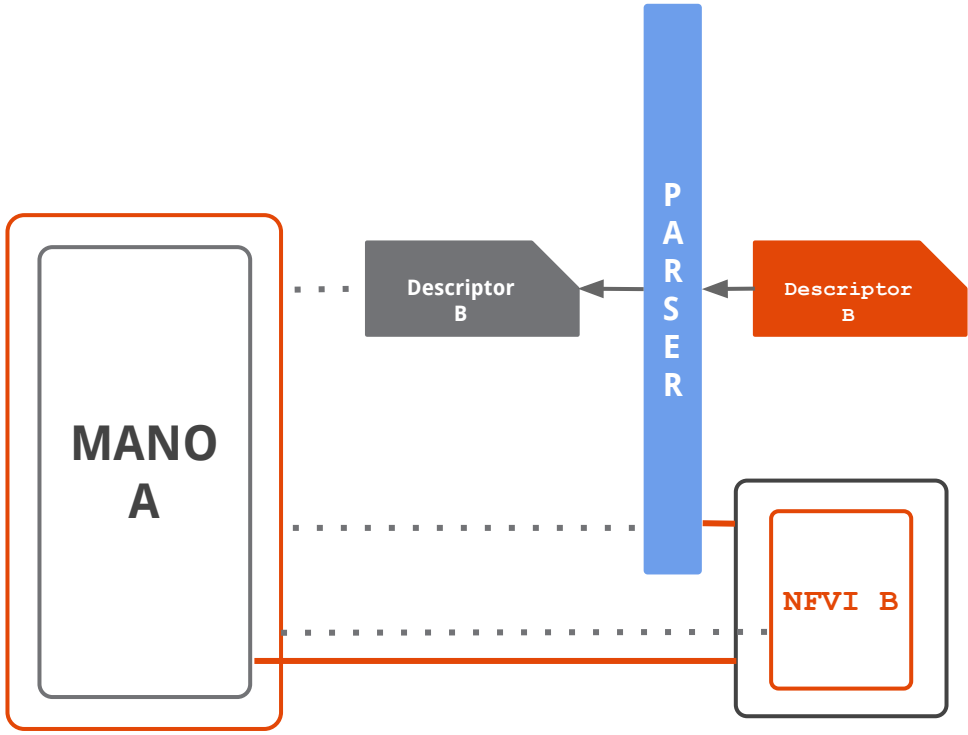
Integration Using Manual Intervention

Manual Intervention

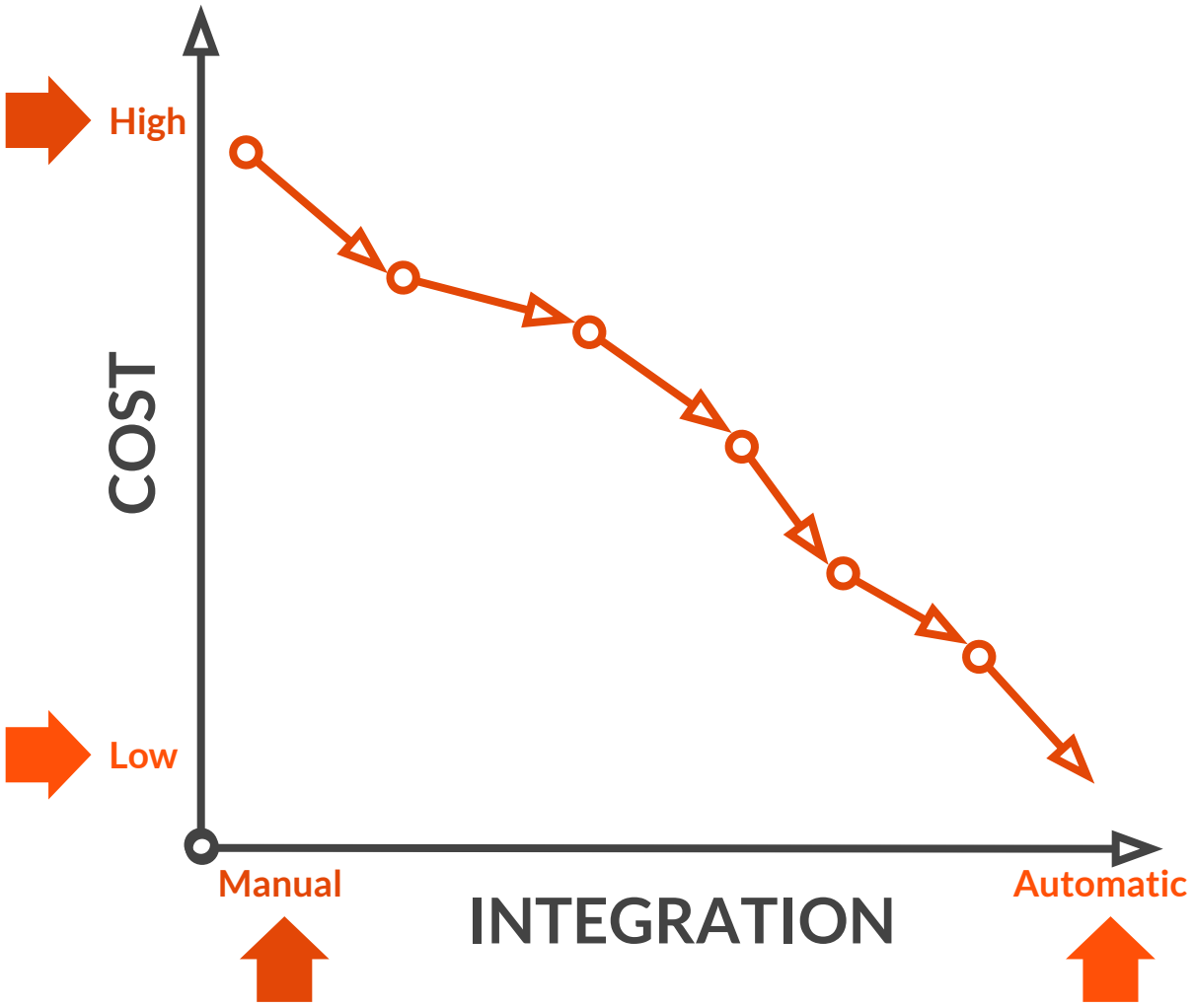
Build new blocks to integrate NFV components
Middlewares

Some Component Integrations

- Parser: Change whole component. Interface remain same
- Parser: Transforms requests, from one side to another. Both interfaces
- Adding new software layers. Both interfaces can be used



Interoperability Cost



Problem Summary

NFV implementations are done without common knowledge

NFV descriptors are done in an arbitrary way

Inherited the problems of WS

Manual intervention to integrate services

NFV Service descriptions call to be done in a implicit way

Higher integration costs

“Which kind of common understanding, semantics and communication mechanism can be used on NFV implementations to reduce manual intervention in order to obtain interoperability with an automatic integration process?”

Research Question

3.
Goals

Our contribution

Specific Goals

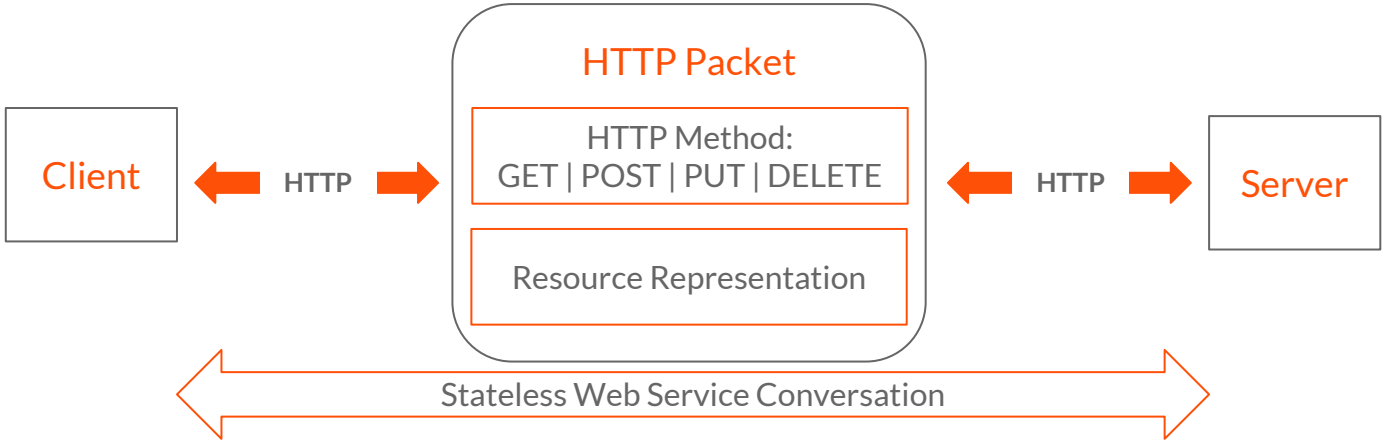
- ➔ Develop a Network Function Virtualisation ontology NOn using as a base the ETSI Virtual Function Network Descriptor (VNFD).
- ➔ Create a semantic representation of the VNFD.
- ➔ Implement NFV interfaces following a semantic service approach.
- ➔ Automate NFV service integration by using an NFV Ontology and a semantic service implementation.
- ➔ Validate the concept of NFV semantic services by proof of concept implementations showcasing automatic service integration

4.
Background

Related technologies

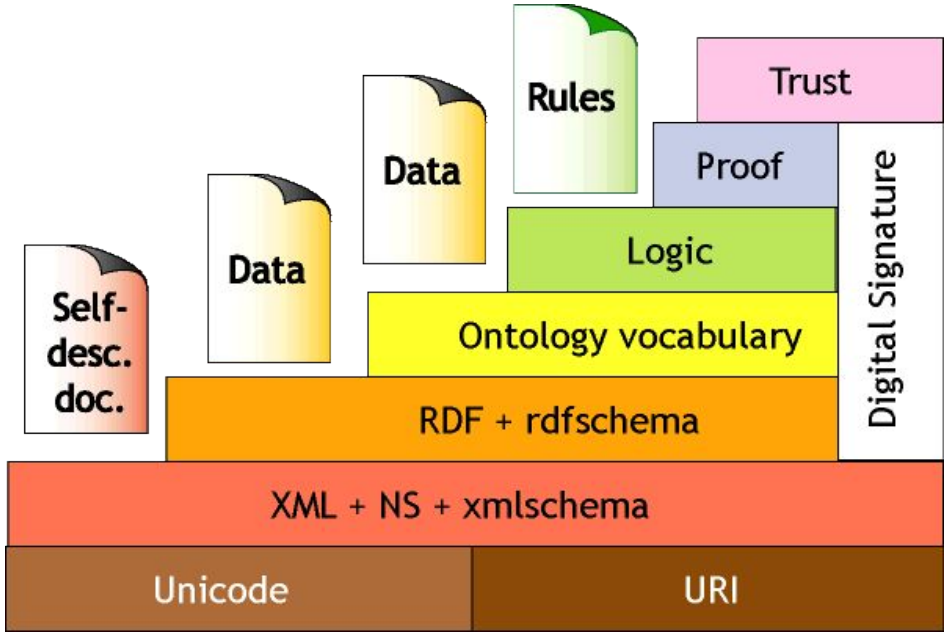
Rest Web Services

- Client-Server
- Stateless Interactions
- Self-descriptive messages
- Uniform Interface
- Named Resources
- Interconnected resource representations
- Layered components

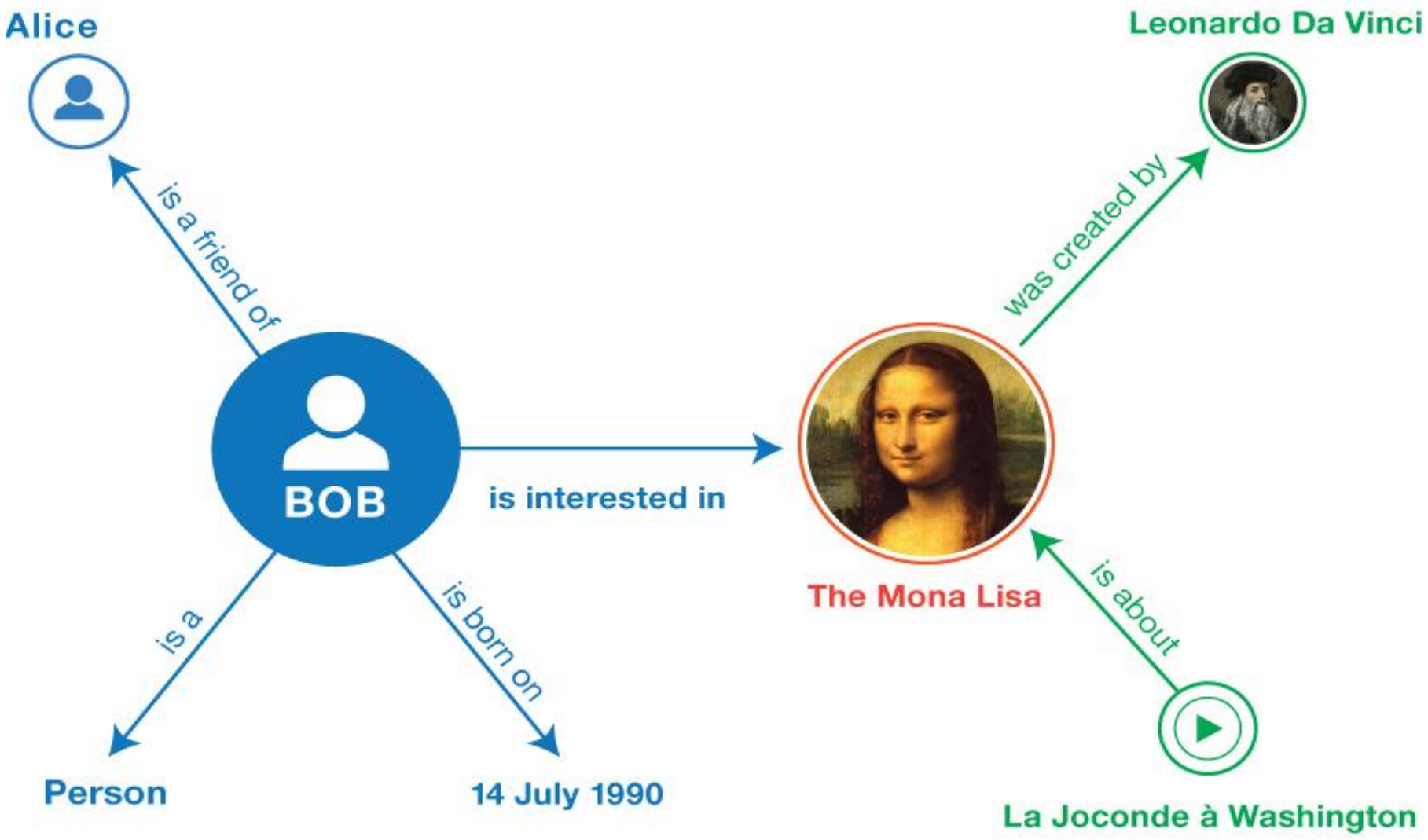


Web Semantic

- ➔ Provide a common syntax for machine-understandable statements
- ➔ Establish common vocabularies as in an ontology
- ➔ Agree on a logical language
- ➔ Use the language to exchange proofs



RDF - Resource Description Framework

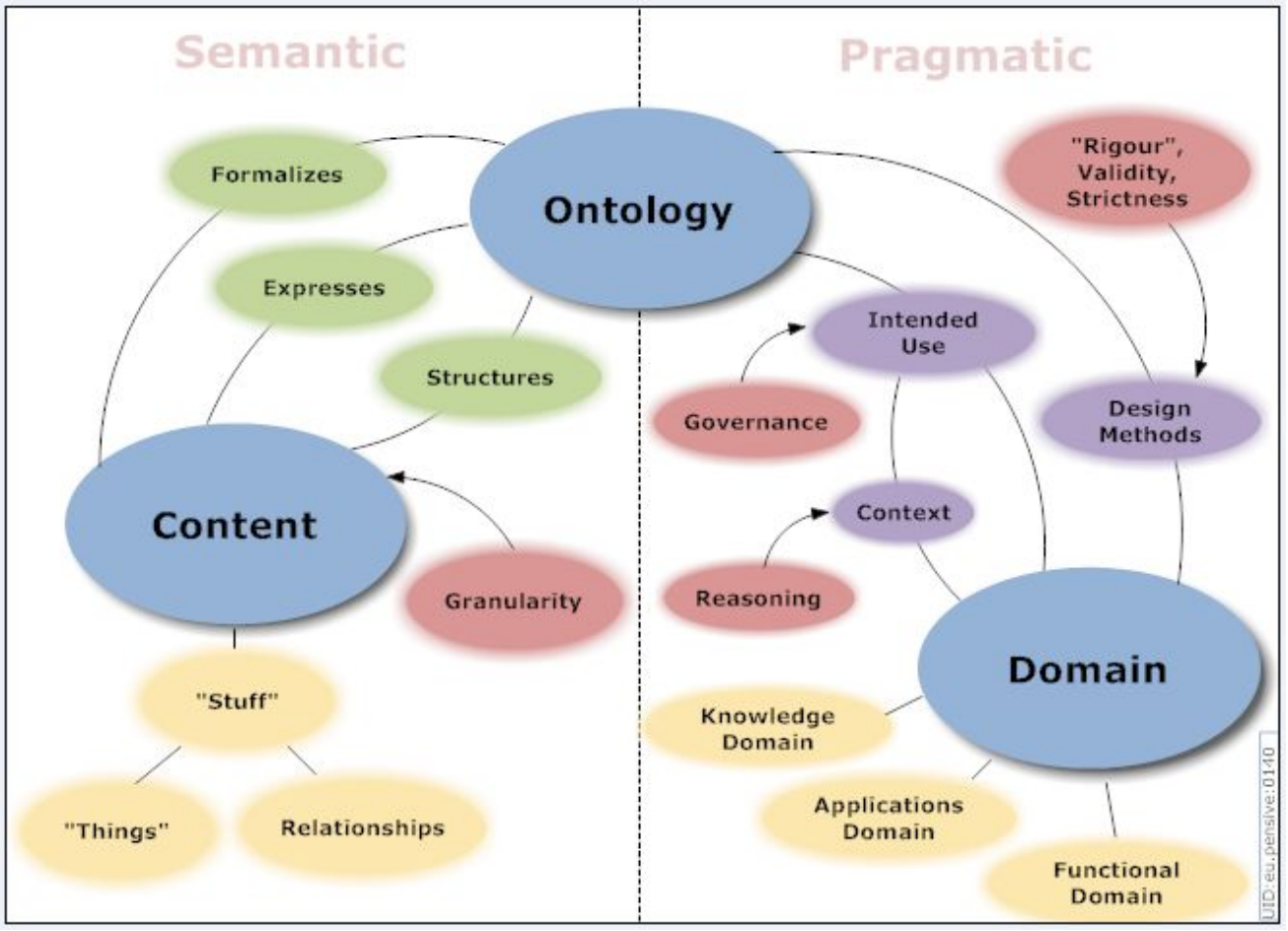


N3 - Notation 3

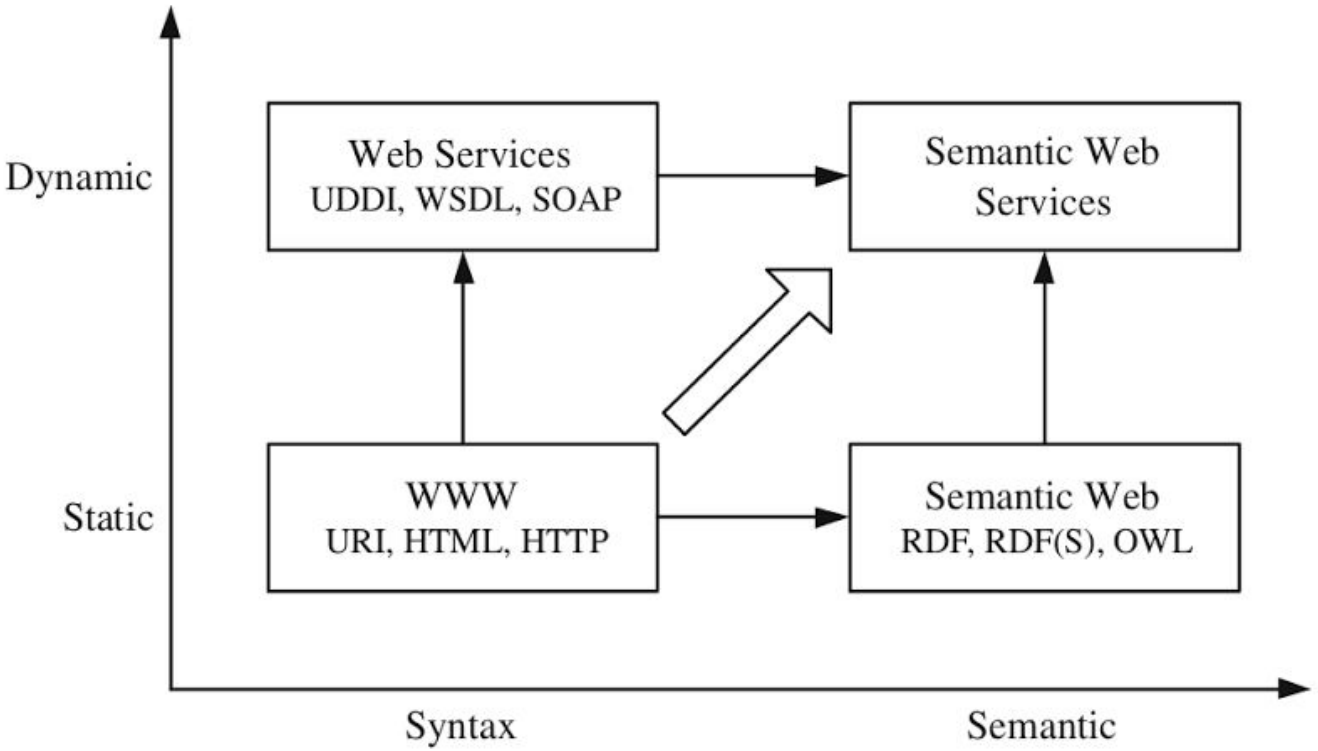
A super set logic language of RDF and extends the data model

```
@prefix ppl:<http://example.org/people#>.
@prefix foaf:<http://xmlns.com/foaf/0.1/>.
{
   ppl:Cindy foaf:knows ppl:John.
}
=>
{
   ppl:John foaf:knows ppl:Cindy .
}.
```

Ontologies


Semantic
Pragmatic
Ontology
Formalizes
Expresses
Structures
Content
Granularity
"Stuff"
"Things"
Relationships
"Rigour", Validity, Strictness
Intended Use
Governance
Design Methods
Context
Reasoning
Domain
Knowledge Domain
Applications Domain
Functional Domain
UID:eu.pensive:0140

Semantic Web Services



Web Semantic

+

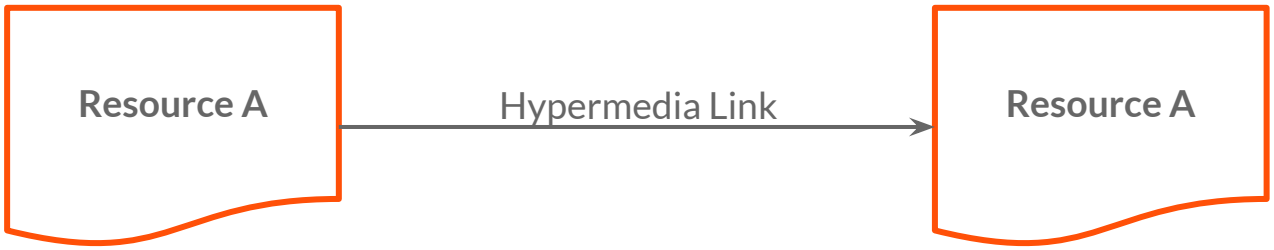
Web Services

=

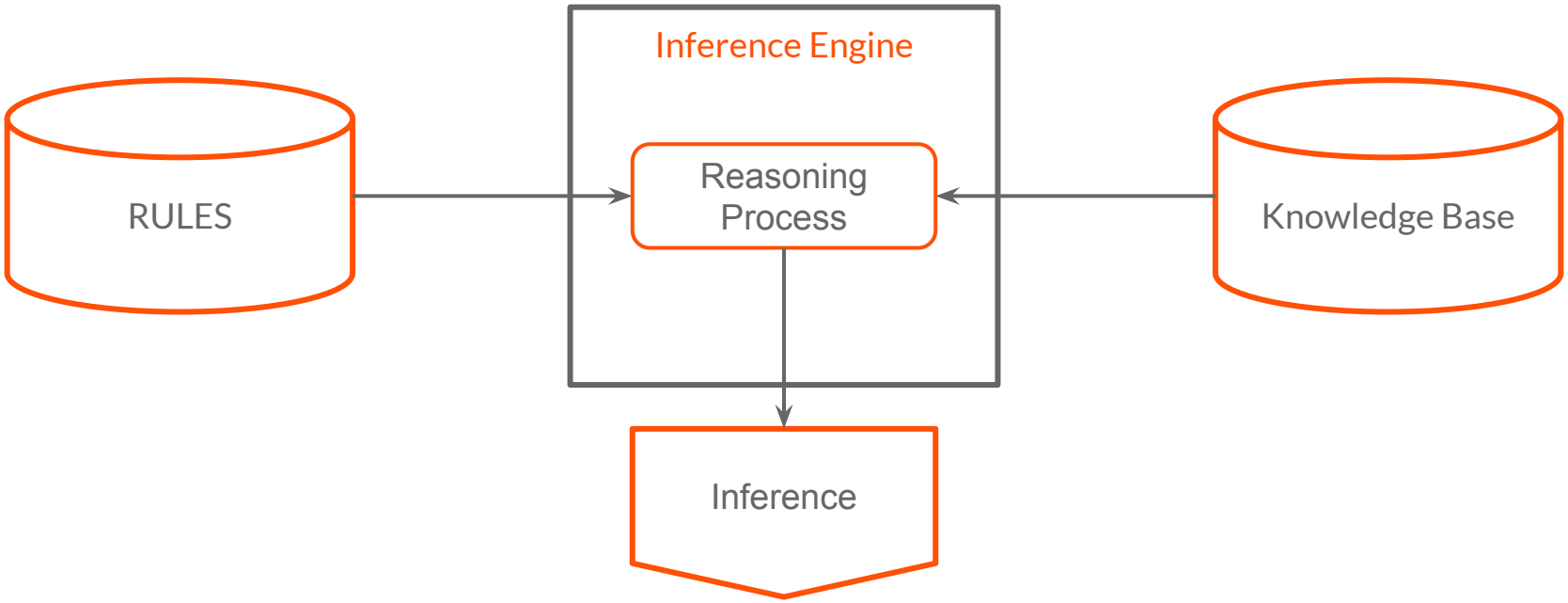
Semantic Web Services

RESTdesc describes hyperlinks

- ➔ REST API have hypermedia links
- ➔ Machines should browse the Web like humans
- ➔ What does it mean to follow a link?
  - Explain to machines in RDF
  - Use if/then construct



Inference Engine



5
NFV Ontology

Open Project:

A common understanding of NFV

NOn Roadmap


Reading
Abstracting
Making Relationships
Designing
Building

5.1
Low Level Elements

Defining Model Elements

Identifier	Type	Cardinality	Description
vendor	Leaf	1	The vendor generating this VNFD.
vdu	Element	1...N	This describes a set of elements related to a particular VDU, see clause 6.3.1.2.
connection point	Element	1...N	This element describes an external interface exposed by this VNF enabling connection with a Virtual Link, see clause 6.3.1.4 (see note).

Element	Type	Cardinality	Slot Type	Descriptor
`vnfd`	Object	1	N/A	VNFD
`vdu`	Object	1...*	N/A	VNFD
`vendor`	Slot	1...*	String	VNFD

Extending Model Elements

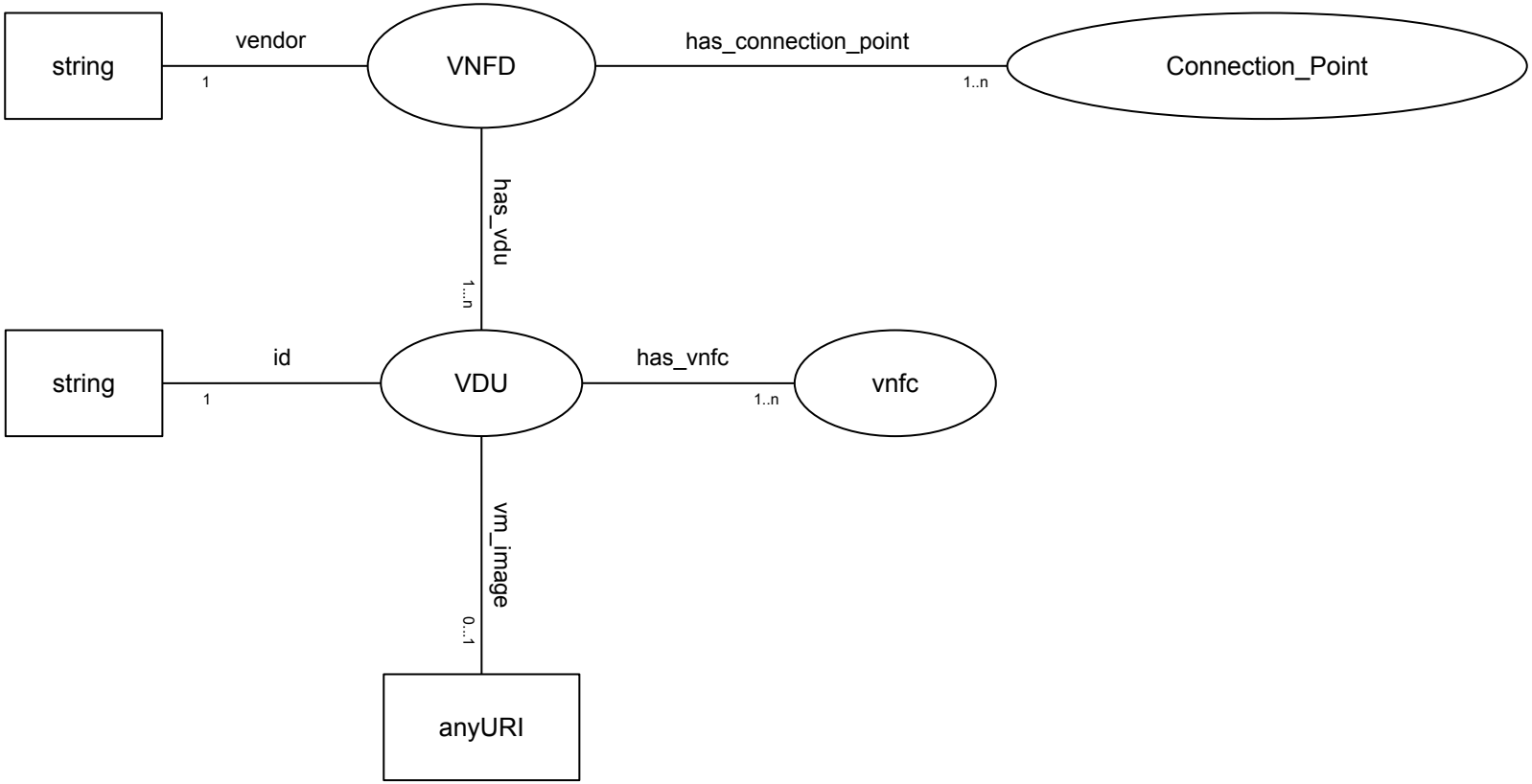
Identifier	Type	Cardinality	Description
id	Leaf	1	A unique identifier of this VDU within the scope of the VNFD, including version functional description and other identification information. This will be used to refer to VDU when defining relationships between them..
vm image	Element	1...N	This provides a reference to a VM image
vnfc	Element	1...N	Defines minimum and maximum number of instances which can be created to support scale out/in..

Element	Type	Cardinality	Slot Type	Descriptor
`vnfd`	Object	1	N/A	VNFD
`vdu`	Object	1...*	N/A	VNFD
`vendor`	Slot	1...*	String	VNFD
`virtual_memory_resource_element`	Slot	1	Integer	VDU
`id`	Slot	1	String	VDU
`vm_image`	Slot	0...1	anyURI	VDU
`vnfc`	Object	1...*	N/A	VDU

5.3
Relationships

Defining Relationships

Descriptor	Property	Cardinality	Value
VNFD	id	1	String
	descriptor_version	1	String
	vnf_version	1	String
	has_vdu	1...*	vdu
	vendor	1	String
	has_deployment_flavor	0...*	deployment_flavor
	has_connection_point	1...*	connection_point
VDU	virtual_memory_resource_element	1	Integer
	scale_in_out	0...1	Integer
	computation_requirement	1	Integer
	id	1	String
	vm_image	0...1	anyURI
	has_vnfc	1...*	vnfc
VNFC	id	1	String
	has_connection_point	1...*	connection_point


string
vendor
1
VNFD
has_connection_point
1..n
Connection_Point
has_vdu
1...n
string
id
1
VDU
has_vnfc
1..n
vnfc
vm_image
0...1
anyURI

5.2
High Level Elements

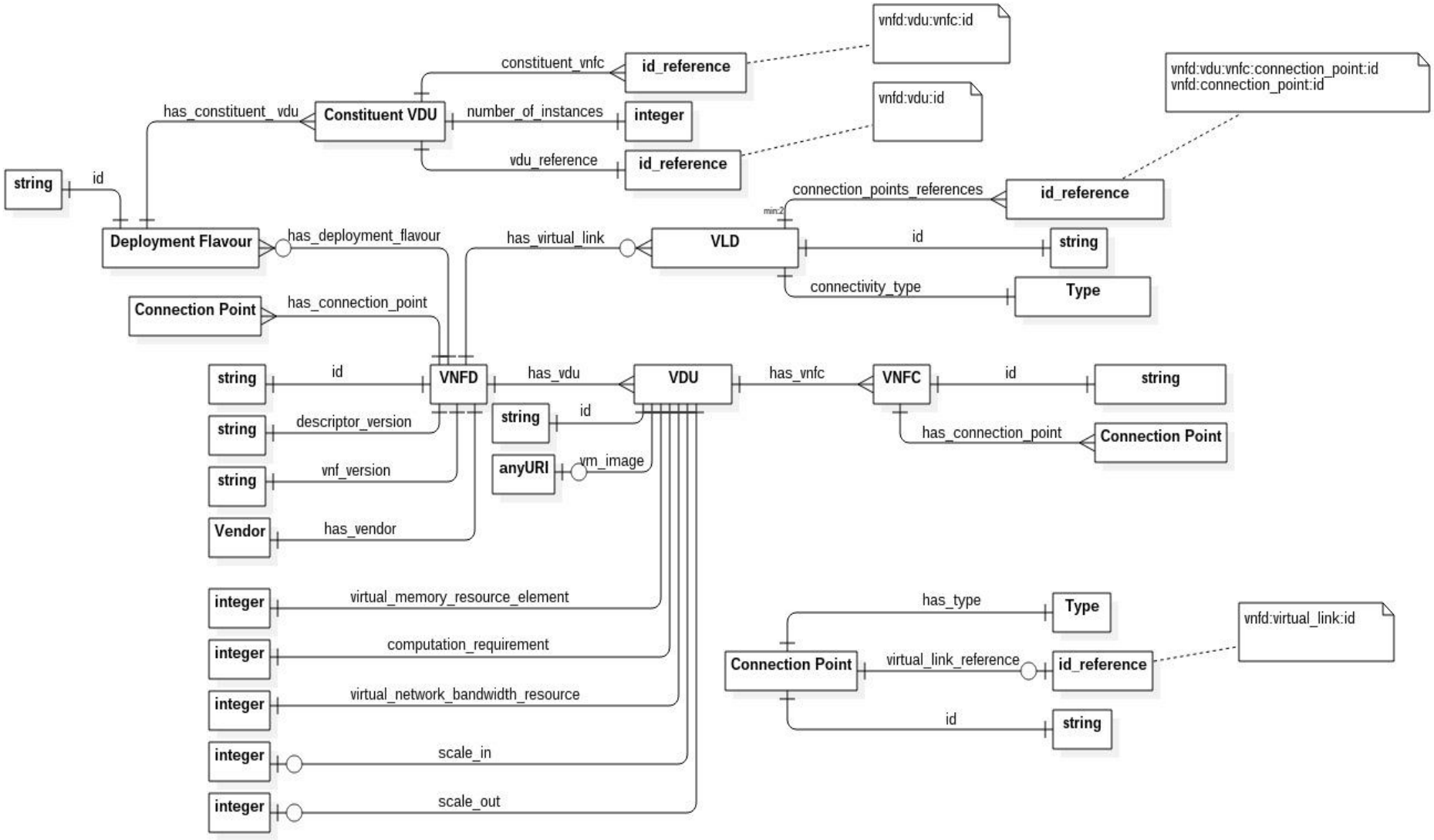
NFV Framework Elements

Root/Functional_Blocks

- Manager and Orchestrator (MANO)
  - NFV Orchestrator Component
  - VNF Manager Component
  - VIM Component
- NFV Infrastructure {INDL}
  - Hardware Components {Resources}
    - Compute Component
      - CPU {Node_Component}
      - Memory {Node_Component}
    - Network Component {SwitchingMatrix}
    - Storage Component {Node_Component}
  - Virtualisation Components
    - Virtual Compute Component {Virtual_Node}
    - Virtual Network Component {Virtual_Node}
    - Virtual Storage Component {Virtual_Node}
  - Hyper-visor
- Descriptors
  - VNFD
  - Virtual Link Descriptor (VLD)
  - VDU
- VNF

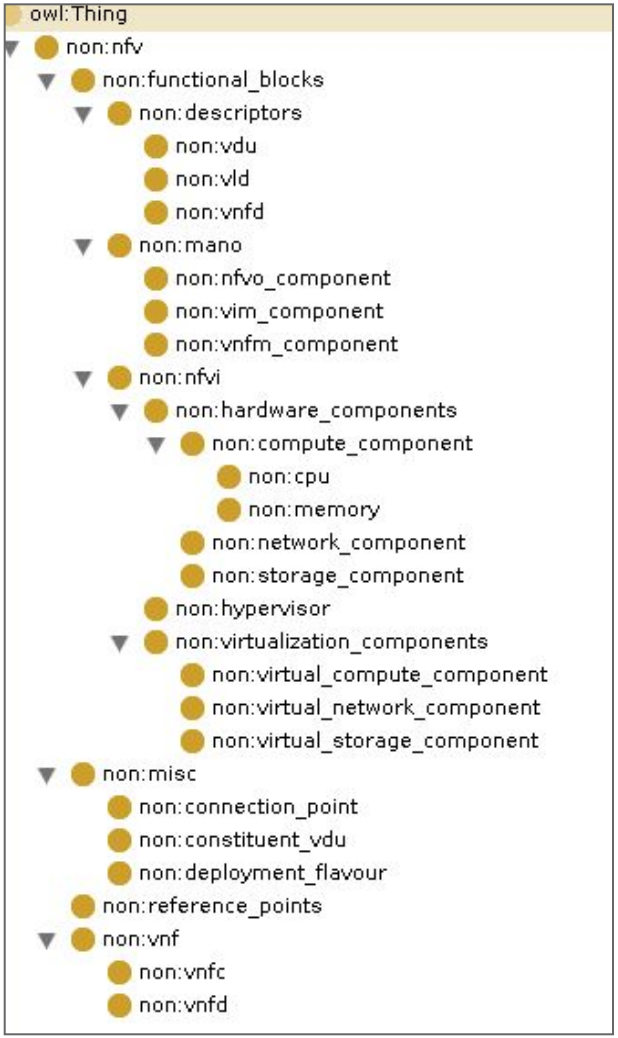
5.4
Model

NFV Ontology



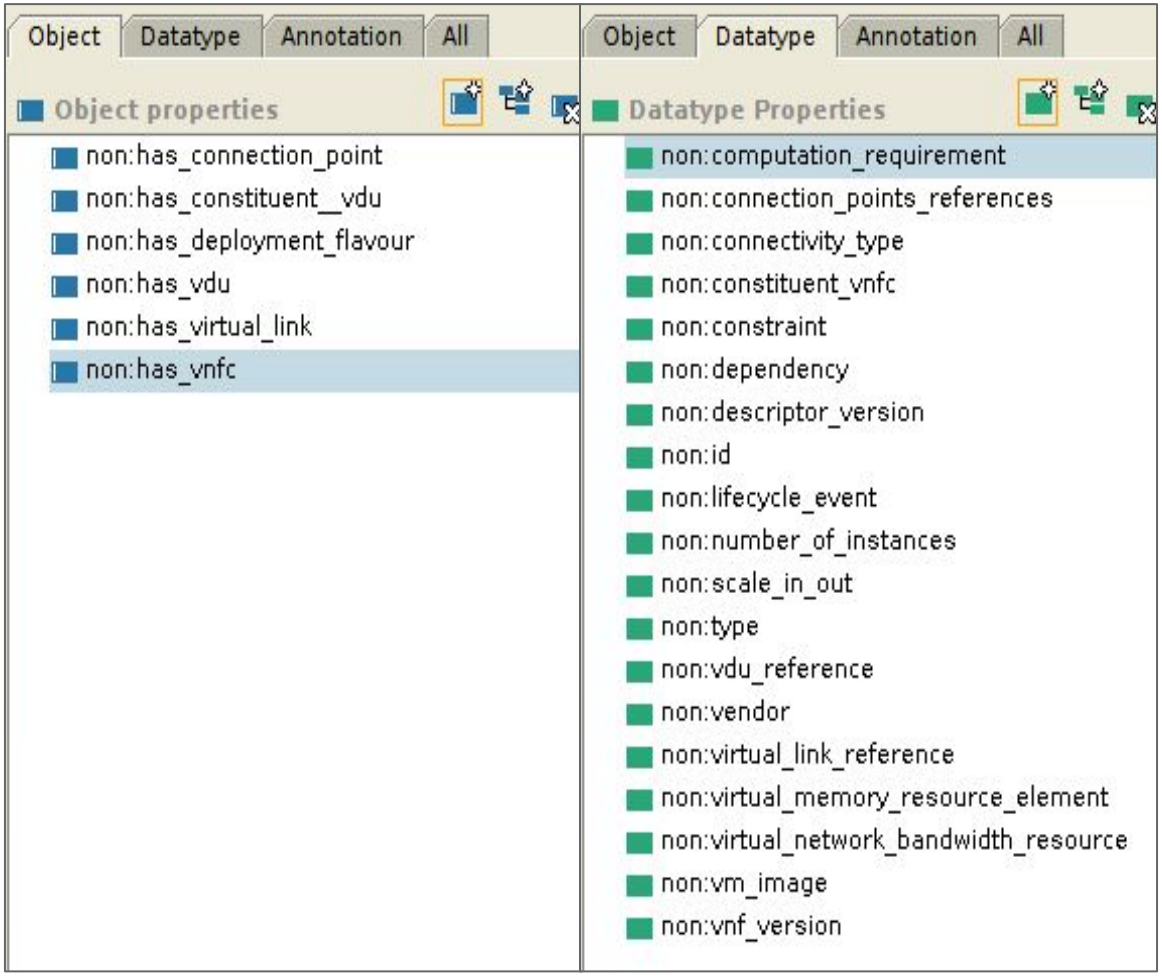
5.4
Implementation

NOn Main Classes



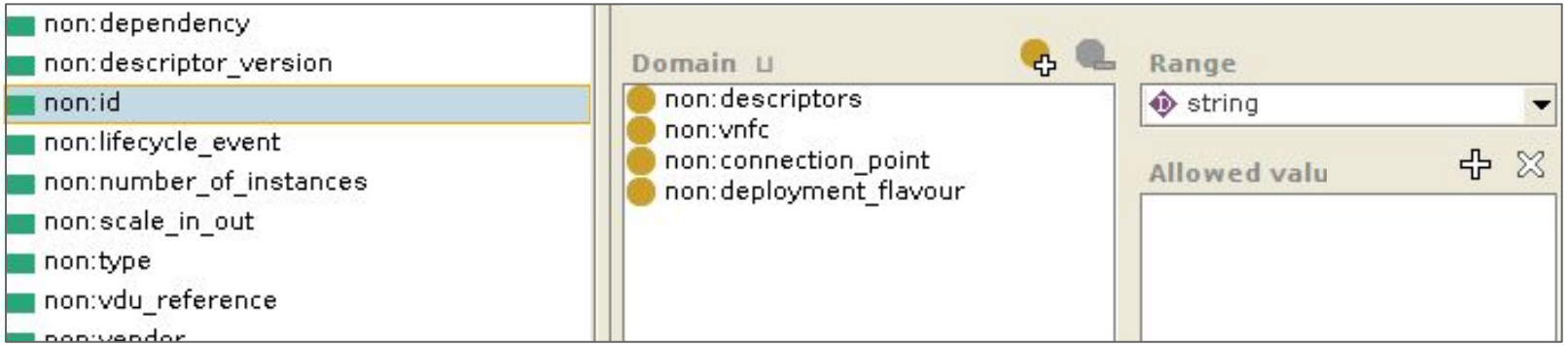

```
<owl:Class rdf:about = "#functional_blocks">
     <rdfs:subClassOf rdf:resource = " #nfv "/>
</owl:Class >
<owl:Class rdf:about = "#mano">
     <rdfs:subClassOf>
          <owl:Class rdf:ID = "functional_blocks"/>
     </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about = "#descriptors">
     <rdfs:subClassOf>
          <owl:Class rdf:about = "#functional_blocks"/>
     </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID = "#vnfd">
     <rdfs:subClassOf >
          <owl:Class rdf:about = "#descriptors"/>
     </rdfs:subClassOf >
</owl:Class >
```

NOn Properties



Property Facets

Range and Domain



Cardinality

Properties Data Model

Object Properties

```
<owl:ObjectProperty rdf:about="#has_vdu">
      <rdfs:range rdf:resource="#vdu"/>
      <rdfs:domain rdf:resource="#vnfd"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#has_deployment_flavour ">
      <rdfs:range rdf:resource="#deployment_flavour "/>
      <rdfs:domain rdf:resource="#vnfd"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#has_vnfc">
      <rdfs:range rdf:resource="#vnfc"/>
      <rdfs:domain rdf:resource="#vdu"/>
</owl:ObjectProperty>
```

Data Properties

```
<owl:DatatypeProperty rdf:about="#vnf_version">
      <rdfs:domain rdf:resource="#vnfd"/>
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#dependency">
      <rdfs:domain rdf:resource="#vnfd"/>
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#virtual_link_reference">
      <rdfs:domain rdf:resource="#connection_point"/>
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

VNFD CLASS

```
<owl:Class rdf:ID="vnfd">
      <rdfs:subClassOf>
            <owl:Class rdf:about="#descriptors"/>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction>
                  <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int ">0</owl:
            minCardinality>
                  <owl:onProperty>
                        <owl:ObjectProperty rdf:ID="has_virtual_link"/>
                  </owl:onProperty>
            </owl:Restriction></rdfs:subClassOf>
            <rdfs:subClassOf>
            <owl:Restriction>
                  <owl:onProperty>
                        <owl:ObjectProperty rdf:ID="has_connection_point"/>
                  </owl:onProperty>
                        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int ">1</owl:
                  minCardinality>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction>
                  <owl:onProperty>
                        <owl:DatatypeProperty rdf:ID="vnf_version"/>
                  </owl:onProperty>
                  <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int ">1</owl:cardinality>
            </owl:Restriction>
      </rdfs:subClassOf>
</owl:Class>
```

5.5
Use Cases

Semantic Description Model

Test Process

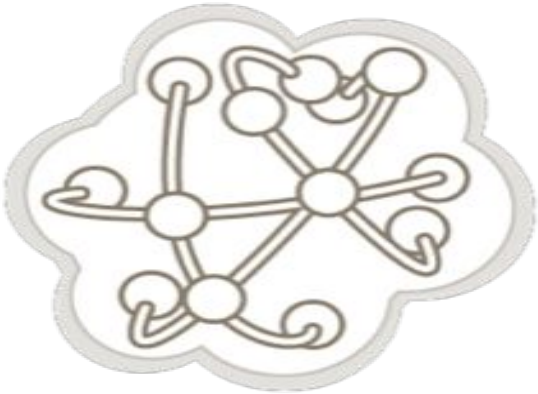
Use VNFD files from current NFV deployments

Parse the elements into a semantic file (Protégé).

Compare original file and the semantic one.

From Syntax to Semantics

NFV
Descriptor



Semantic
NFV
Descriptor

Use Case: OpenBaton

VNFD

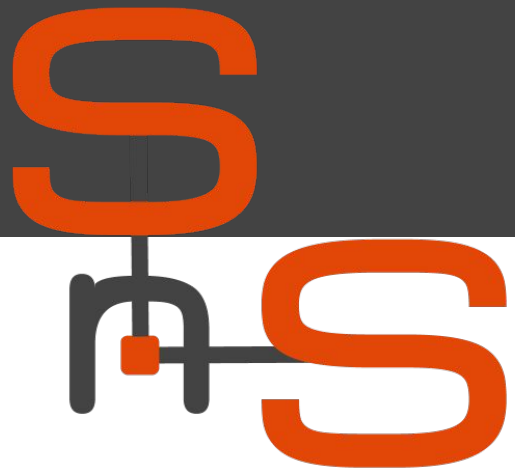
```
{
      "vendor":"fokus",
      "version":"0.2",
      "name":"iperf-server",
      "type":"server",
      "endpoint":"generic",
      "virtual_link":[{"name":"private"}],
      "lifecycle_event":[{"event":"INSTANTIATE"}],
      "deployment_flavour":[{"flavour_key":"m1.small"}],
      "vdu":[{
            "vm_image":["ubuntu-14.04-server-cloudimg-amd64-disk1"],
            "vimInstanceName":"vim-instance",
            "scale_in_out":2}]
}
```

Semantic VNFD

```
###  https://github.com/LCuellarH/NOn/blob/master/datamodel/non.owl#ob-vnfd-1
non:ob-vnfd-1 rdf:type owl:NamedIndividual ,
                        non:vnfd ;
            non:descriptor_version "0.2"^^xsd:string ;
            non:vnf_version "0.2"^^xsd:string ;
            non:lifecycle_event "INSTANTIATE-install.sh-install-srv.sh"^^xsd:string ;
            non:vendor "fokus"^^xsd:string ;
            non:id "iperf-server"^^xsd:string ;
            non:has_vdu non:ob-vdu-1 ;
            non:has_virtual_link non:ob-vld-1 ;
            non:has_deployment_flavour non:os-m1-small .
```

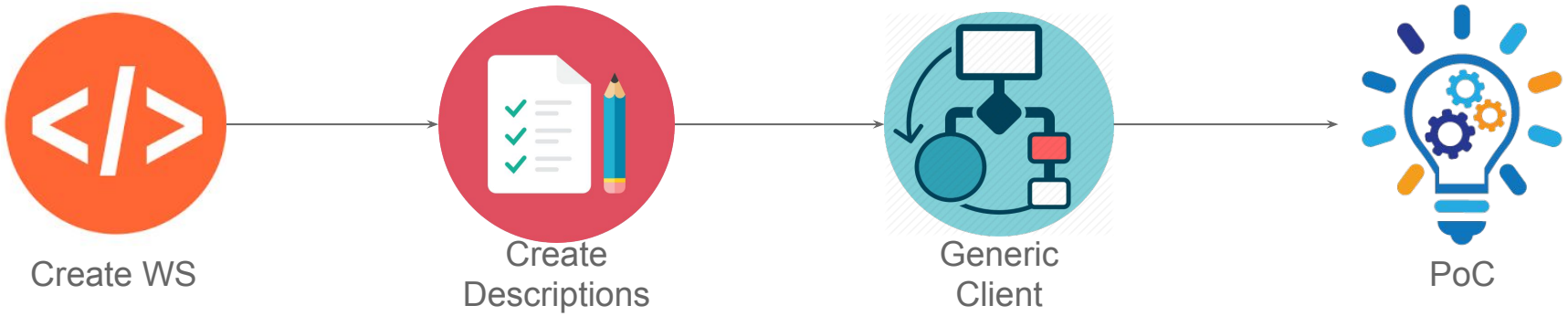
6

Semantic nFV services

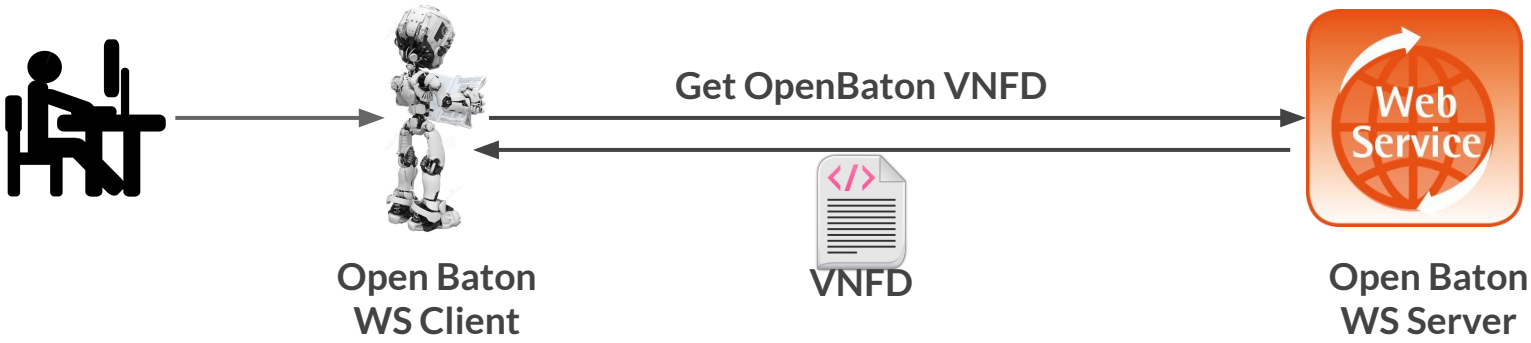


Reasoning NOn

SnS Roadmap



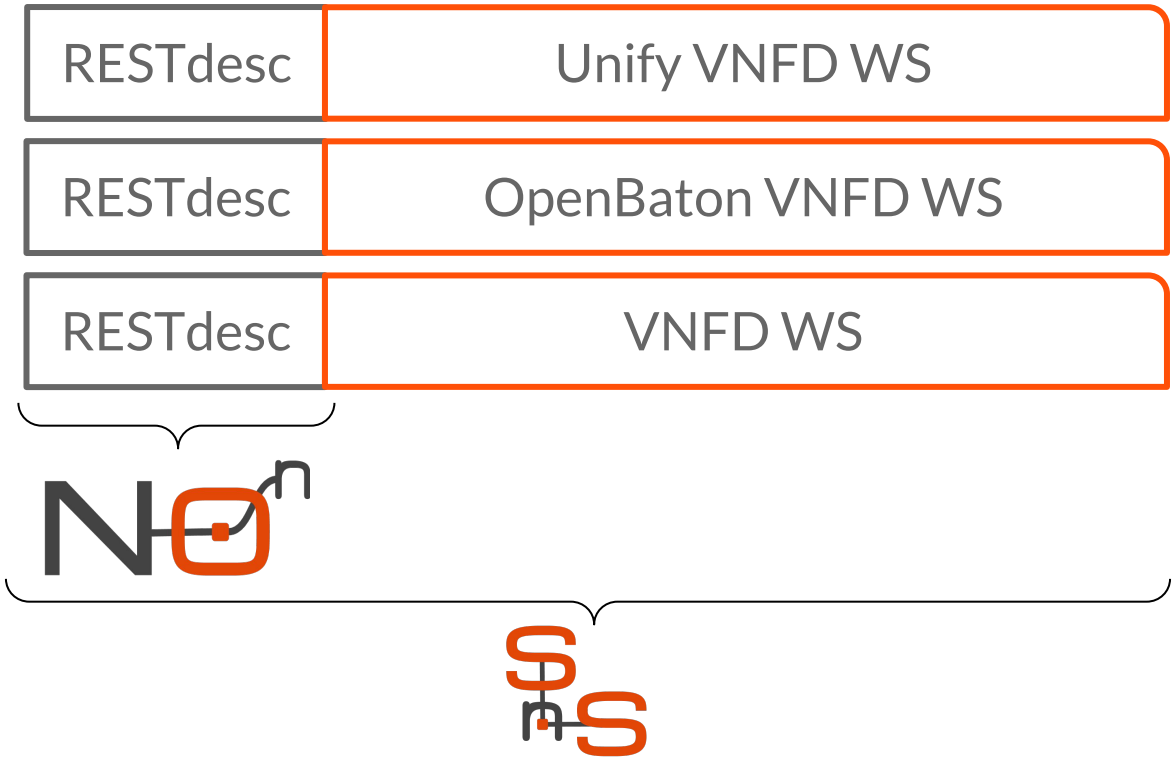
Creating OpenBaton VNFD WebService




```
Params:
        vendor
        name
        vm_image
        virtuallink
        lifecycle
        dev_flavour
        scaleinout
```

```
"vendor":"fokus",
"version":"0.2",
"name":"iperf-server",
"type":"server",
"endpoint":"generic",
"virtual_link":[{"name":"private"}],
"lifecycle_event":[{"event":"INSTANTIATE"}],
"deployment_flavour":[{"flavour_key":"m1.small"}],
"vdu":[{
        "vm_image":["ubuntu-14.04-server-cloudimg-amd64-disk1"],
        "vimInstanceName":"vim-instance",
        "scale_in_out":2}]
```

Adding Semantics NFV Services

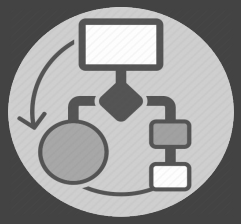


RESTdesc - Semantic Description

```
{
#Pre-conditions

?vnfd a non:vnfd;
       non:id ?vnfd_id;
       non:descriptor_version ?ver_des;
       non:has_vdu ?vdu .
...
?vdu  a non:vdu;
       non:vm_image ?vm_image
...
?vl   a non:vld;
       non:connectivity_type ?vl_type.
}
```

=>

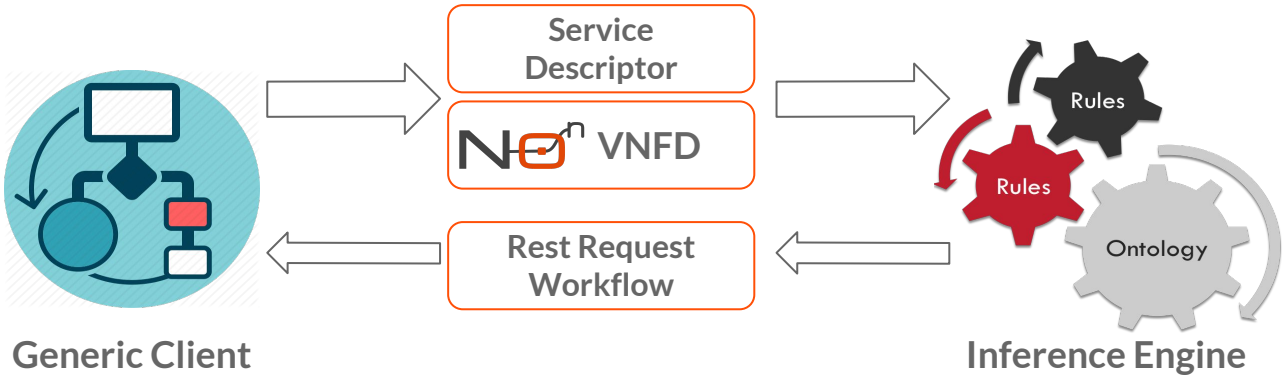
```
{
#Process

_:request http:methodName "GET";
         http:MessageHeader "Content-Type:application/json";
         http:requestURI ("http://localhost:8080/nfv/parser/openbaton/vnf/vnfd?vendor="?
vendor"&version="?ver_des"&vm_image="?vm_image"&virtuallink="?vl_type"");
         Http:resp[http:body json:openbaton_vnfd].

#Post-conditions

?vnf non:has_vnfd ?non_vnfd.
}..
```

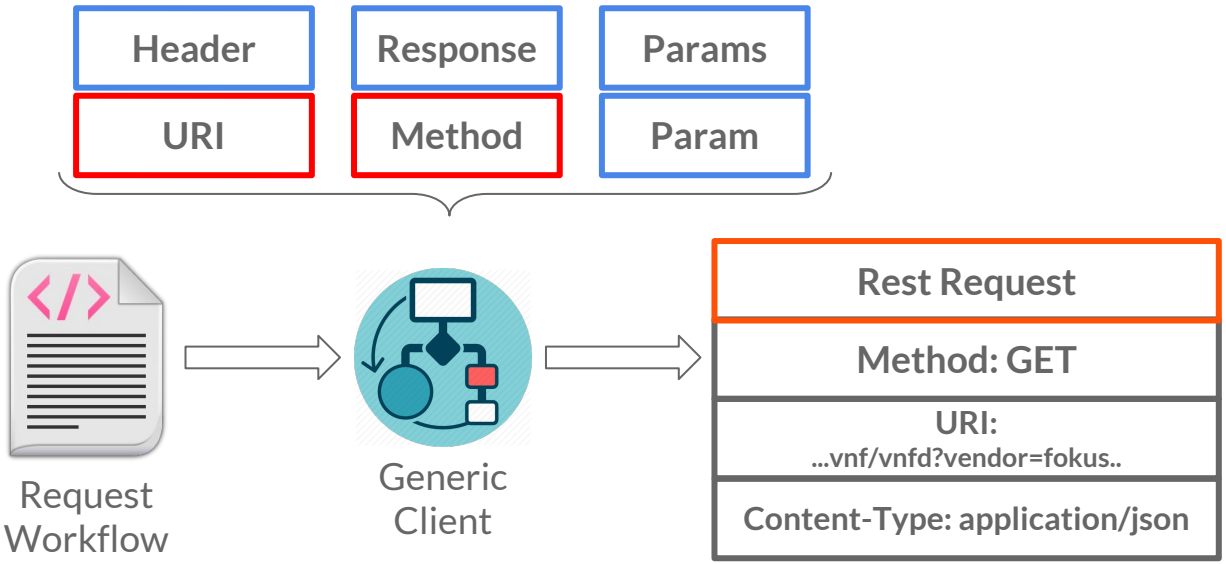


Generic Client

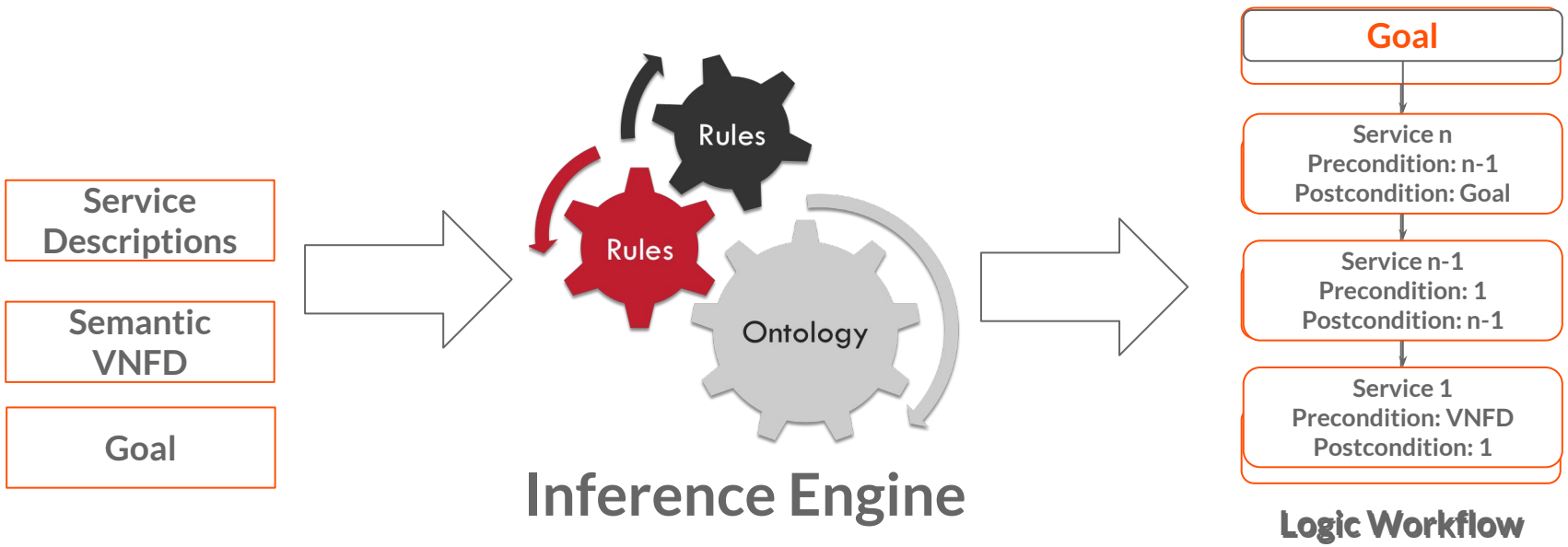
Adapting itself to consume workflows without using a predefined context or background

Consuming Semantic Services

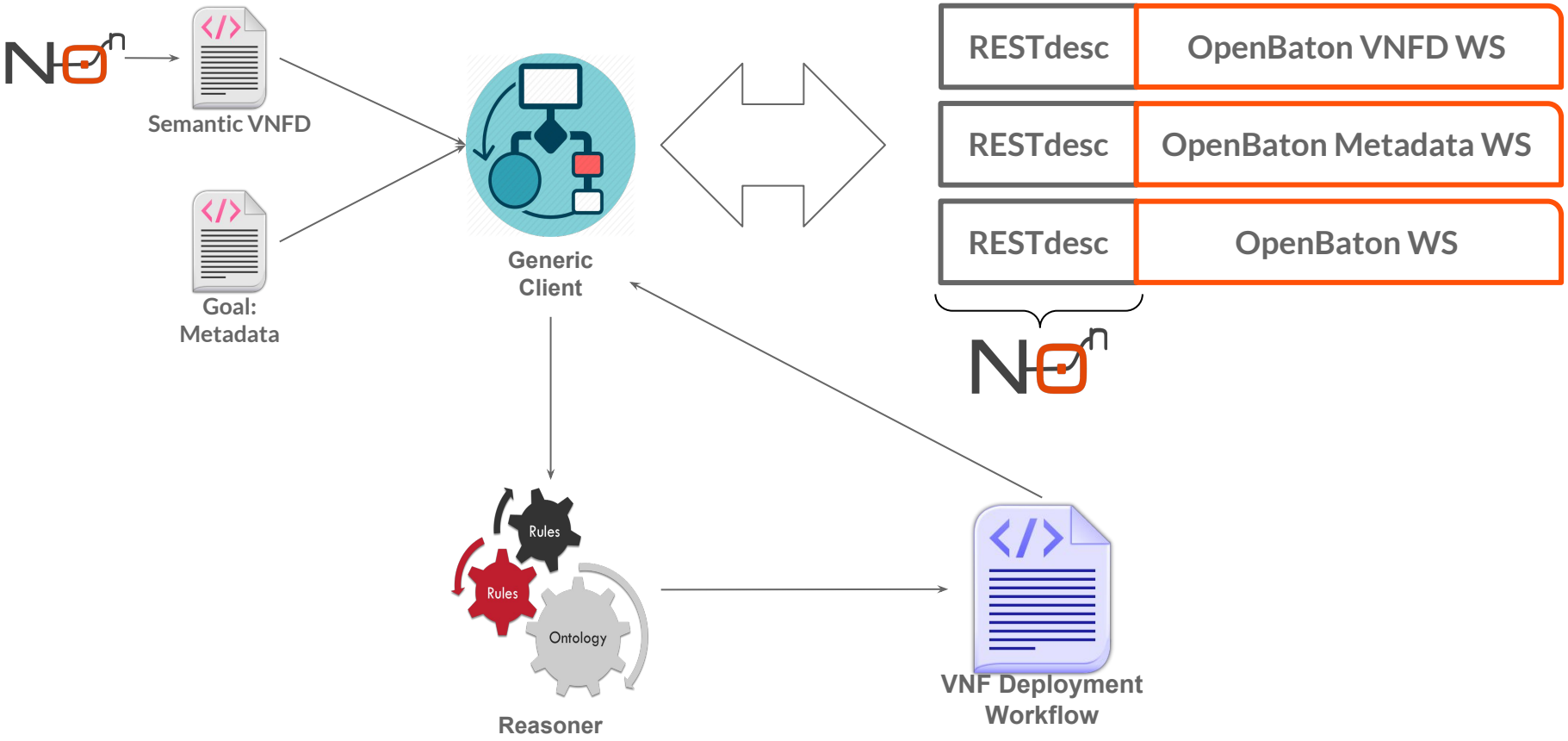




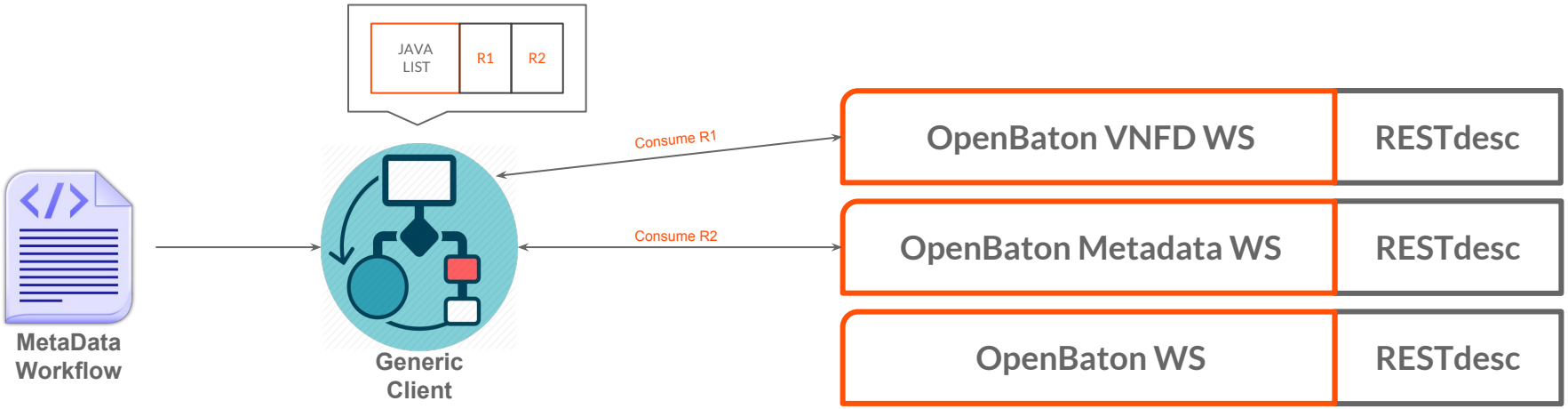
SnS Workflow Inference



Consuming a Goal-Based Workflow



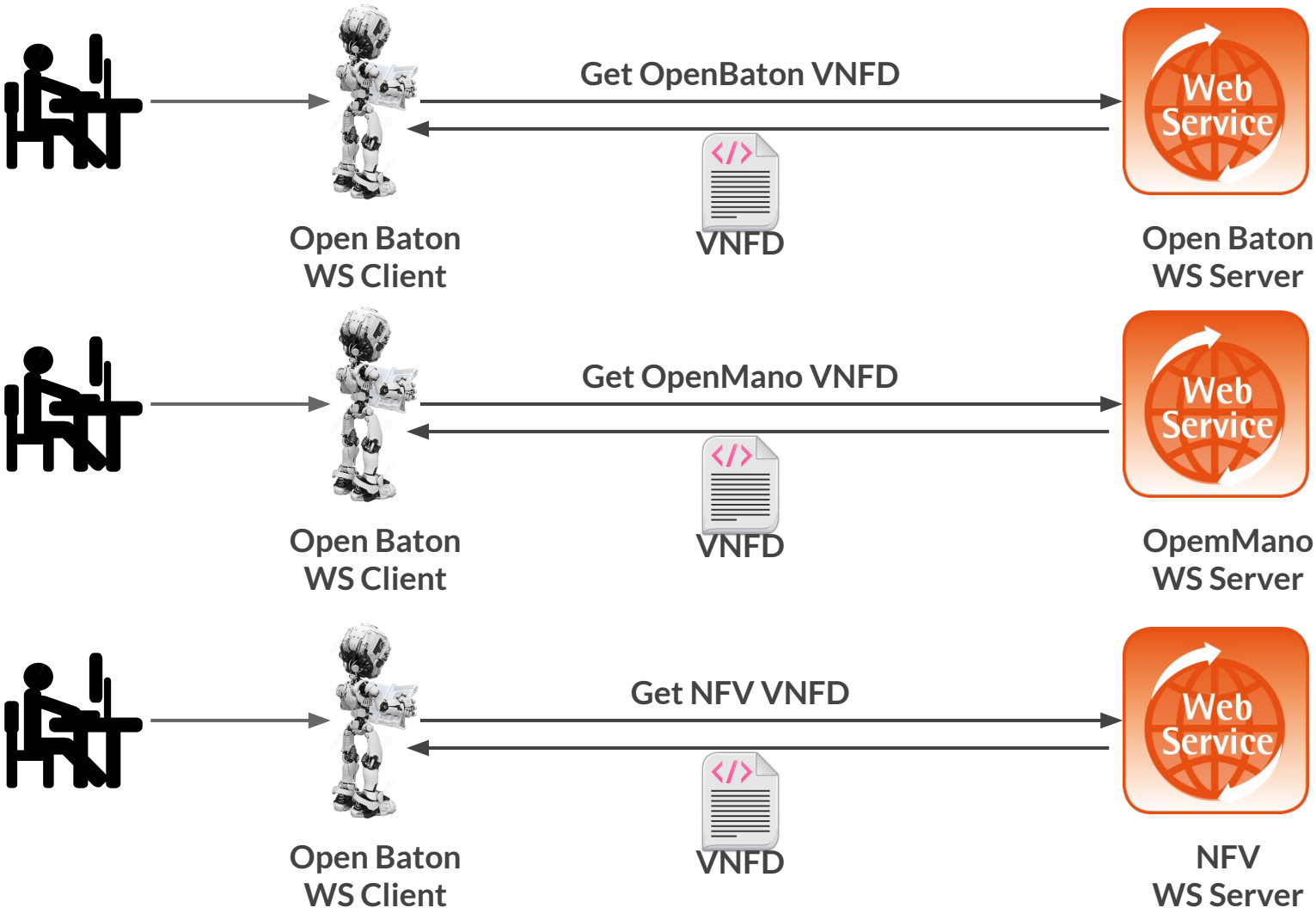
Consuming a Goal-Based Workflow



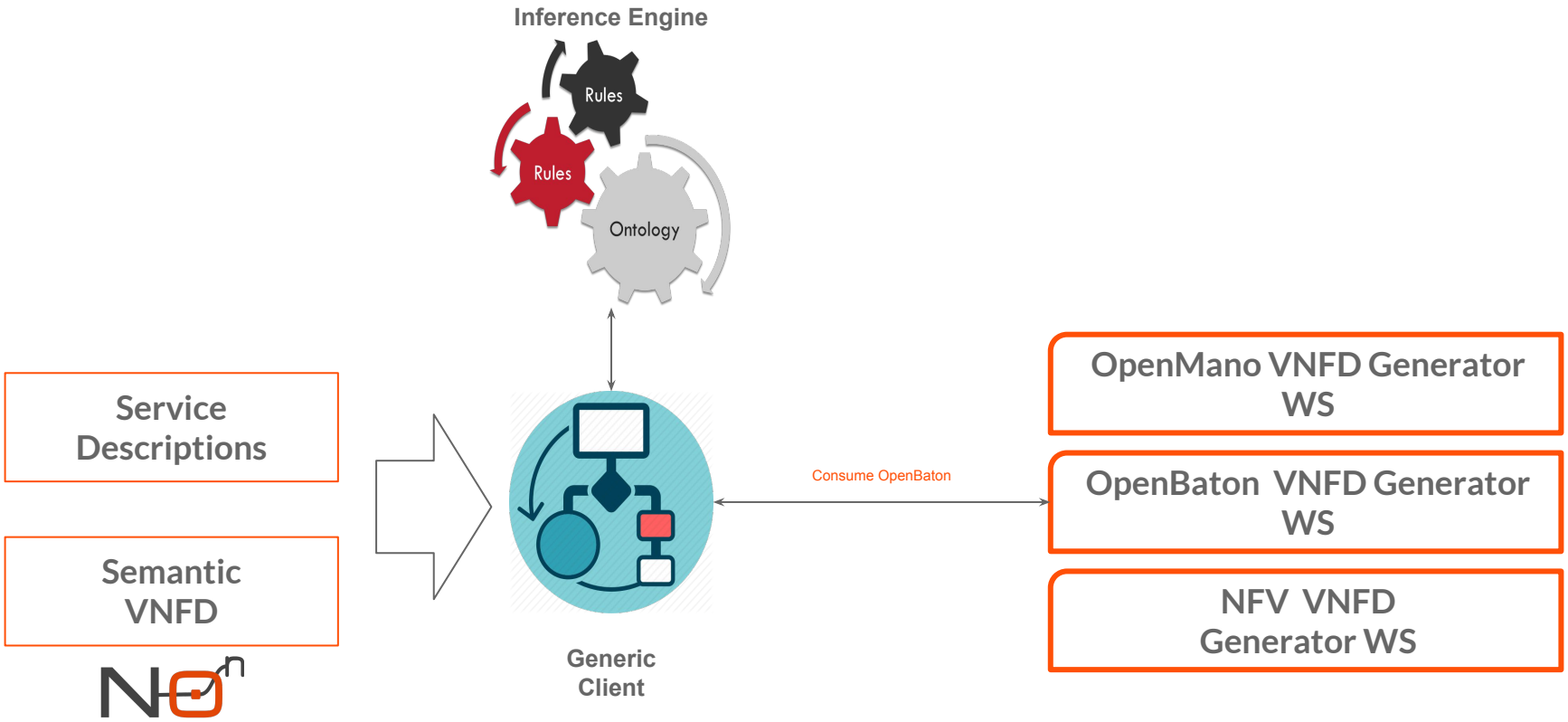
Use Case I

VNFD Benchmarking

Consuming Multiple NFV Services



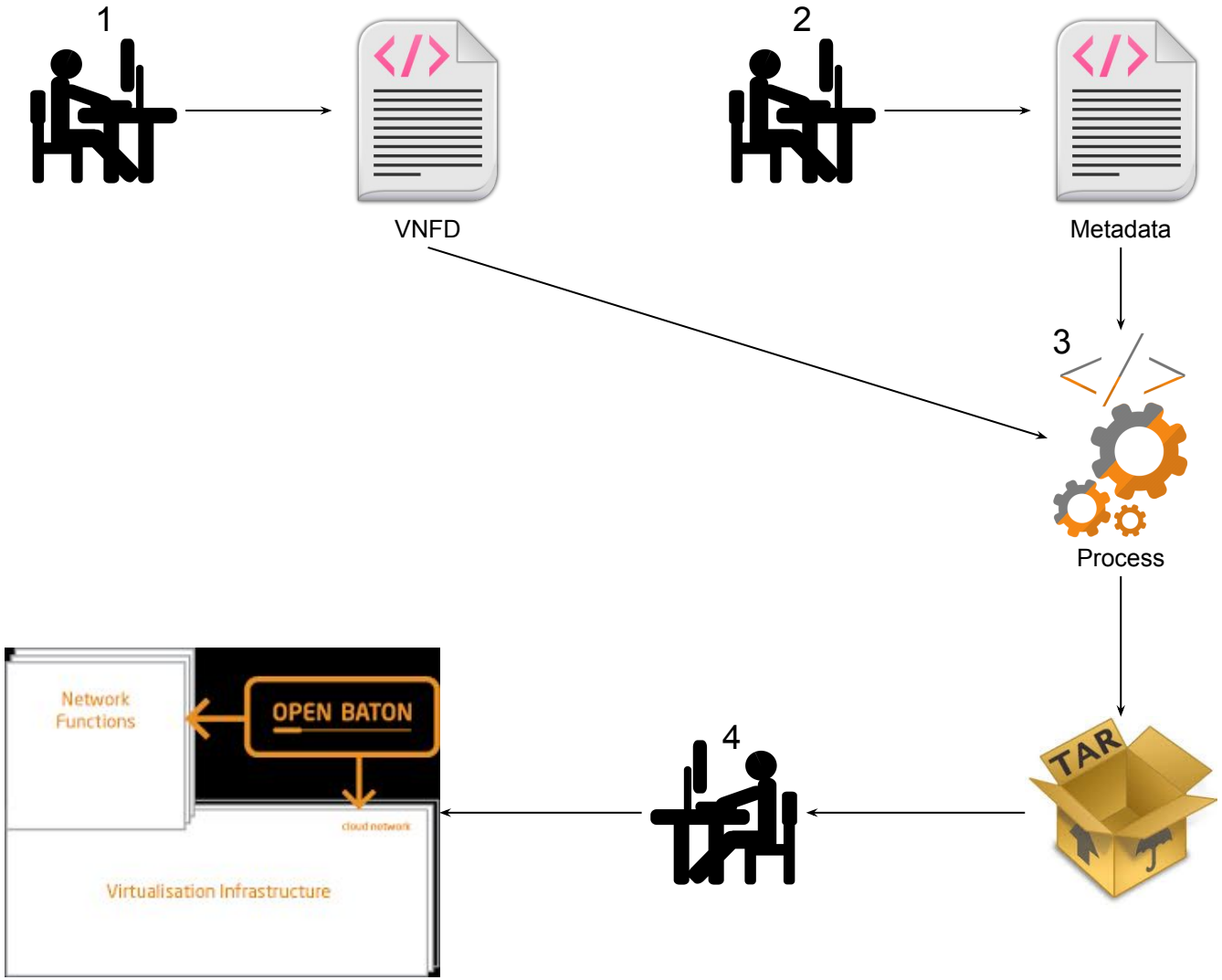
Scenario


Inference Engine
Rules
Rules
Ontology
Service Descriptions
Semantic VNFD
Generic Client
Consume OpenBaton
OpenMano VNFD Generator WS
OpenBaton VNFD Generator WS
NFV VNFD Generator WS

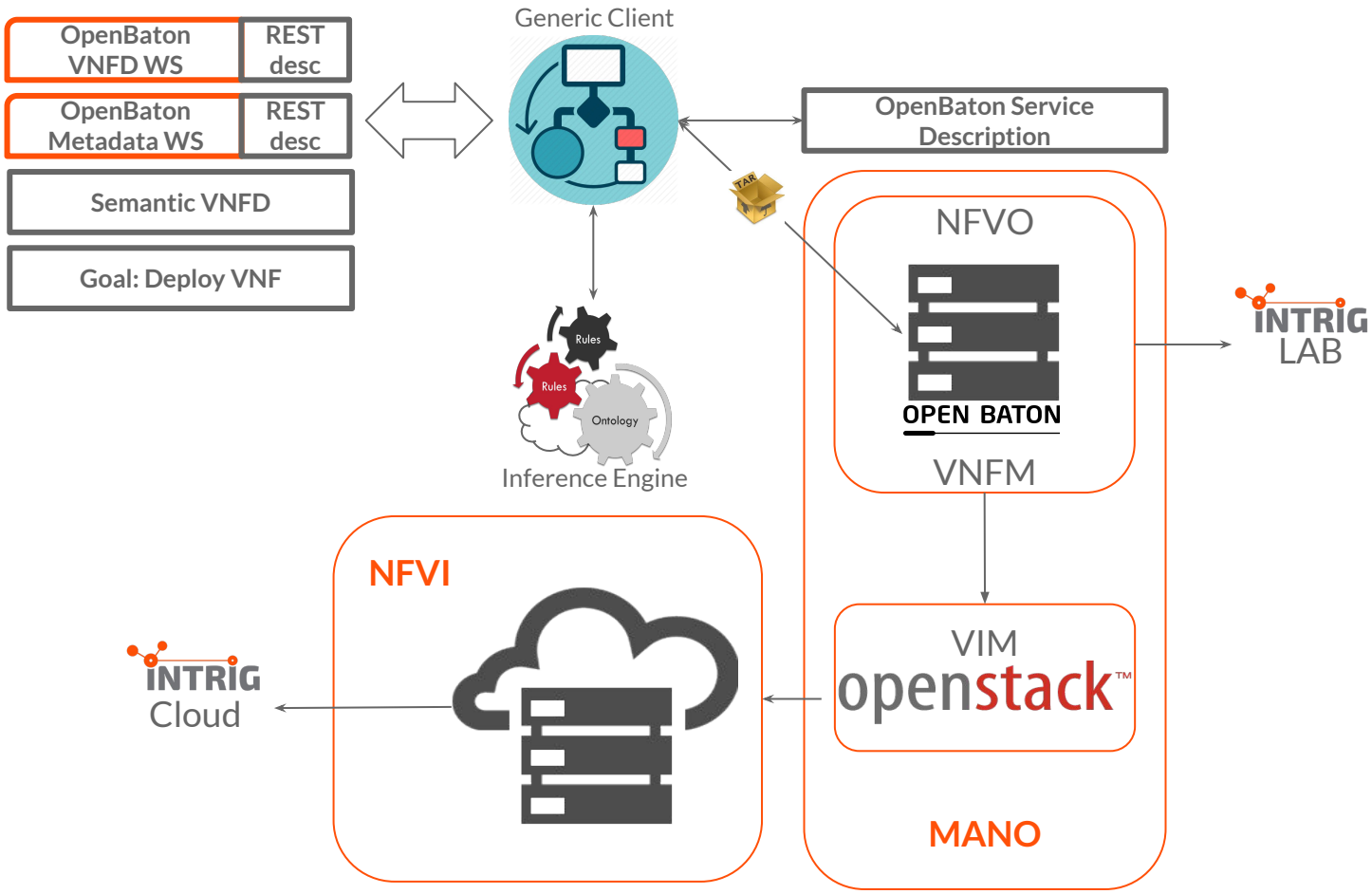
Use Case II

Deploying a VNF Semantic Service

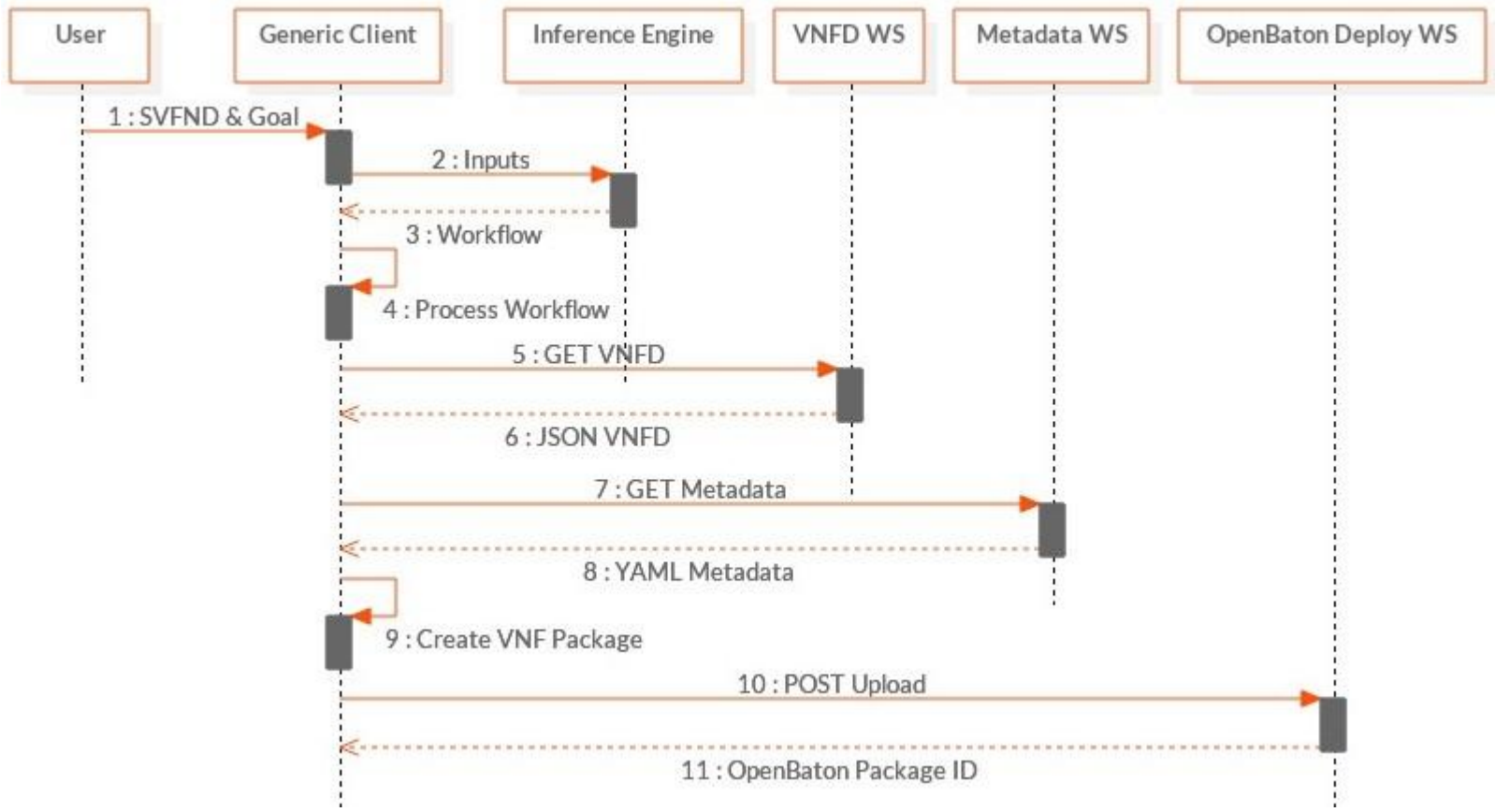
OpenBaton VNF Deployment Process



Scenario


OpenBaton VNFD WS
REST desc
OpenBaton Metadata WS
REST desc
Semantic VNFD
Goal: Deploy VNF
Generic Client
OpenBaton Service Description
TAR
NFVO
OPEN BATON
VNFM
INTRIG LAB
Rules
Rules
Ontology
Inference Engine
NFVI
INTRIG Cloud
VIM
openstack™
MANO

Sequence Diagram



7.
Conclusions and Future Work

Conclusions and Limitations

Interoperability gaps were identified and attempted to be removed by the implementation of a common NFV data model known NOn

Projects based on the information models defined by the ETSI exhibit better chance inter-working in multi-domain scenarios without requiring manual intervention

NOn opened the door to create Semantic nFV Services. The implementation of REST interfaces, explicit service descriptions and the use of inference engines

A Generic Client was capable of self adapting to consume dynamic REST Web Service workflows without the need of humans in the loop

Open source NFV projects are currently not following ETSI specifications in the syntax definition.

Current NFV implementations do not use semantic technologies in their developments, implementation of semantic technologies were scoped just to component interfaces

Conclusions and Limitations

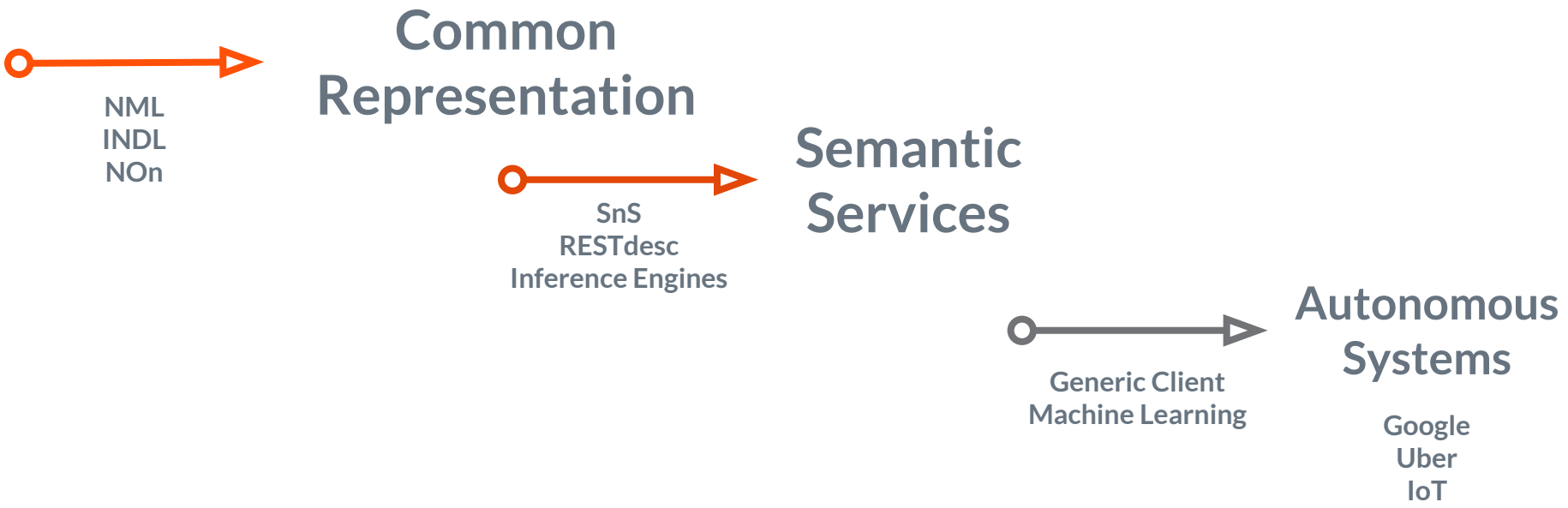
The implementation of NOn and SnS are an initial step towards automatic service integration

The full potential was not explored in the sense of leveraging of semantic technologies to build full components

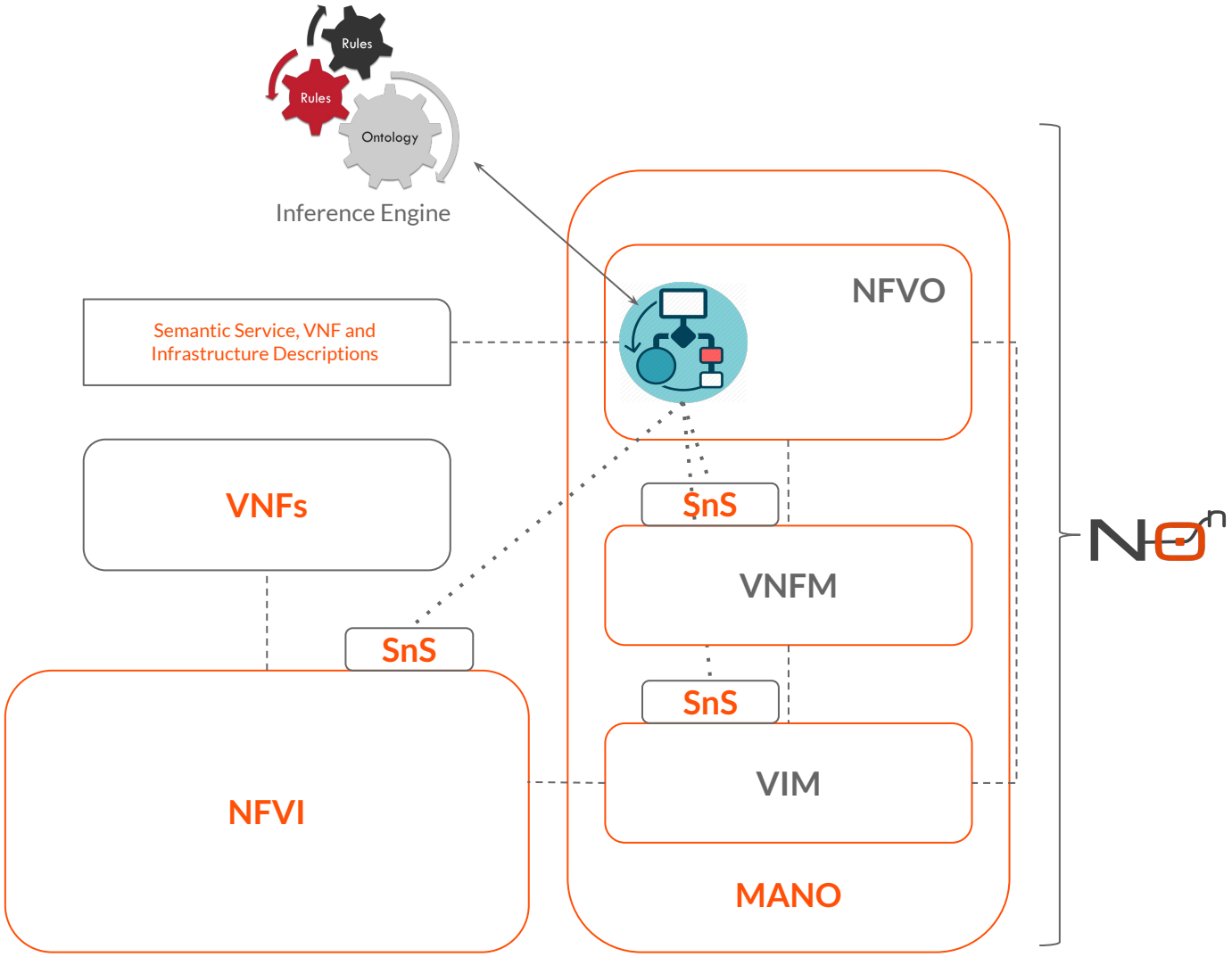
The current state of NOn does not fulfill all the needs of a complete data model capable of describing all possible VNFD files

Contributions and Related Work

- A structured knowledge representation and a common language on the NFV domain, Non.
- A Virtual Network Functions Descriptor with a semantic approach.
- The concept of semantic services for NFV implementations
- A Generic Client capable of read, interpret and consume dynamic workflows.



Future Work

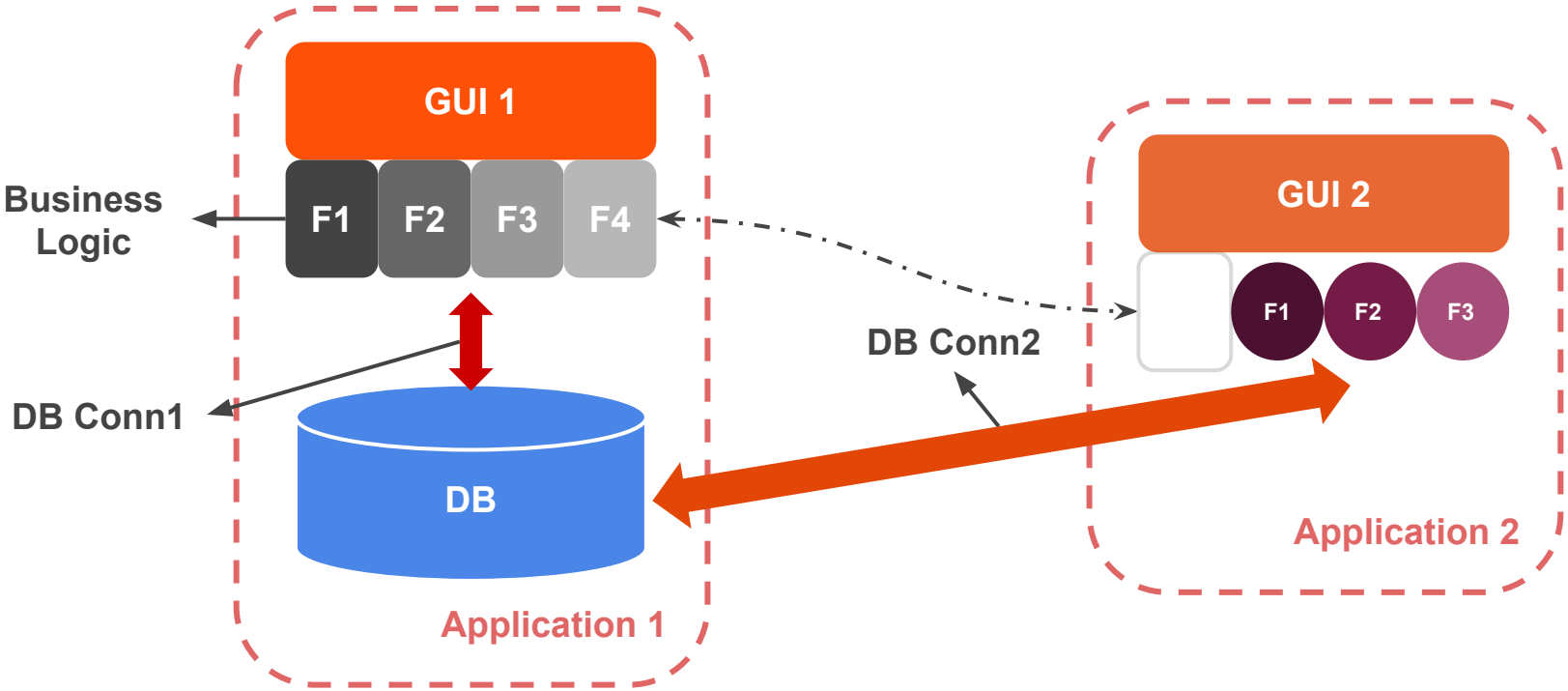

Rules
Rules
Ontology
Inference Engine
Semantic Service, VNF and Infrastructure Descriptions
VNFs
NFVI
SnS
SnS
SnS
NFVO
VNFM
VIM
MANO

Thanks!

Any questions?

You can find me at:
@lcuellarh
luiscuellarh@gmail.com

Legacy Software Systems



Related Work - Description Models

	Semantic Description Defined by	Web Semantic Approach	Web Service Interoperability	Autonomous Interoperability
Network Modeling Language	Share Ontology	Semantic Description	N/A	Can be use
Infrastructure Network Description Language	Share Ontology	Semantic Description	N/A	Can be use
Resource Information Service	Stand Alone Ontology	-	Manual Intervention External Plugin	-

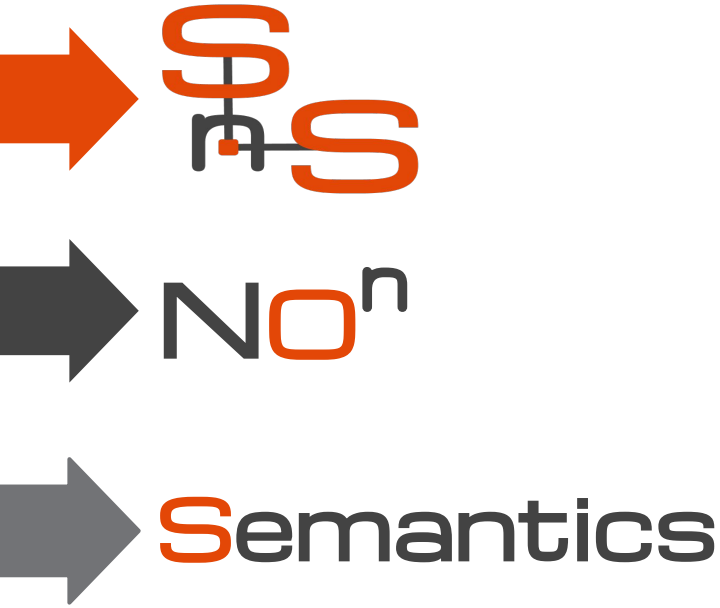
Related Work - Interoperability

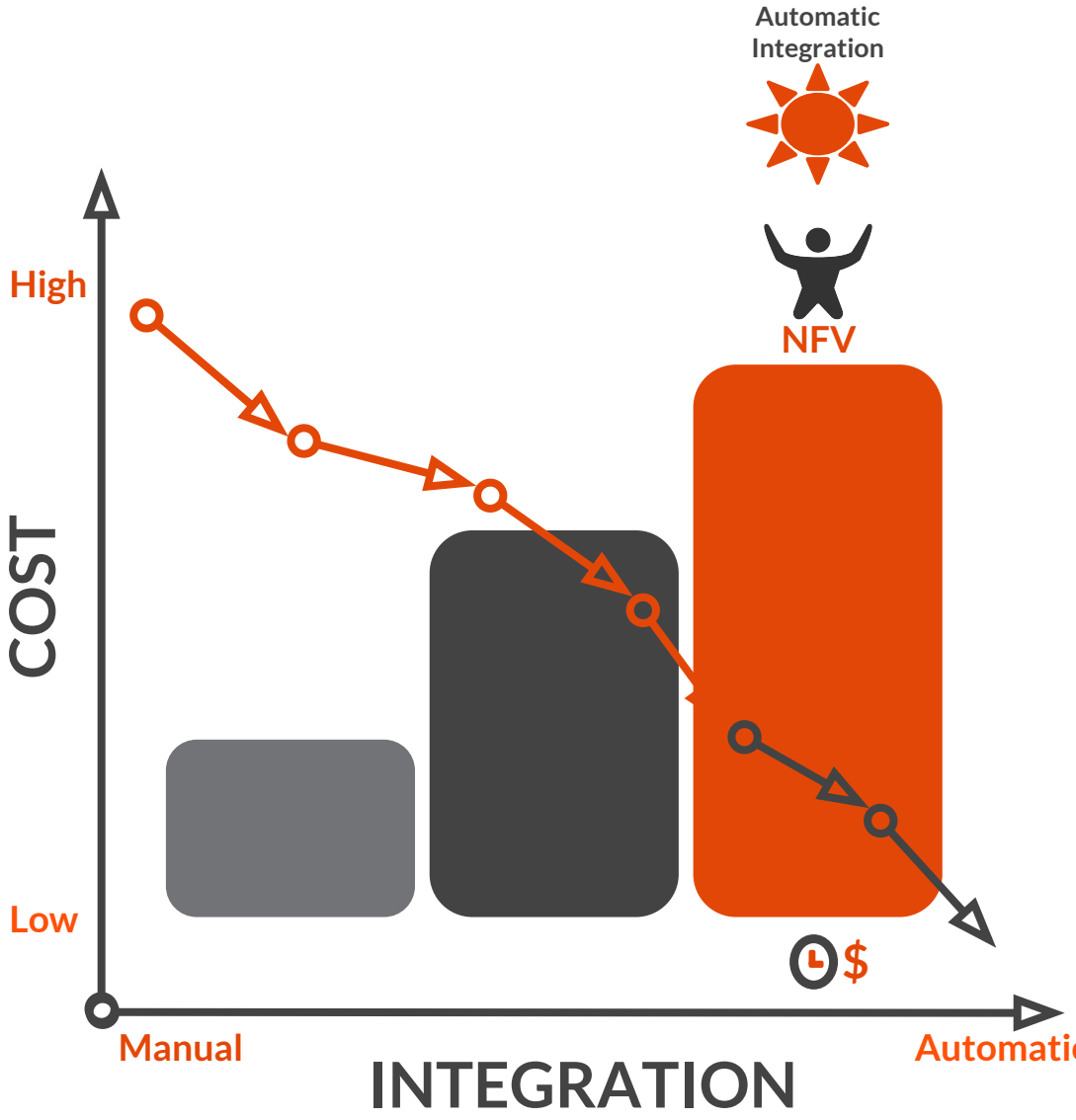
	Semantic Description Defined by	Web Semantic Approach	Web Service Interoperability	Autonomous Interoperability
SDN Rest API	Developers	-	Manual Intervention REST Protocol	-
ML2	Developers Metadata	-	Manual Intervention External Plugin	-
OpenStack	Developers Metadata	-	Manual Intervention External Plugin	-
RESTdesc	Share Ontologies	Semantic Services	REST Protocol	Service Functionality Discovery

Related Work - NFV Implementations

	Semantic Description Defined by	Web Semantic Approach	Web Service Interoperability	Autonomous Interoperability
T-NOVA	Developers Metadata	-	Manual Intervention External Plugin	-

Achieving Our Goal!





NFV High Level Architecture

We Wonder…..

Why if a component developed in a domain A, such NFVI can not just be “Plug and Play” in a domain B and use it by other components, such as MANO?

Why a developed VNF can not have a descriptor that multiple NFV domains can interpret and use?

How different NFV implementations following the same ETSI specifications could have the same capabilities independently of the development technologies by sharing common terminology to define the NFV elements?

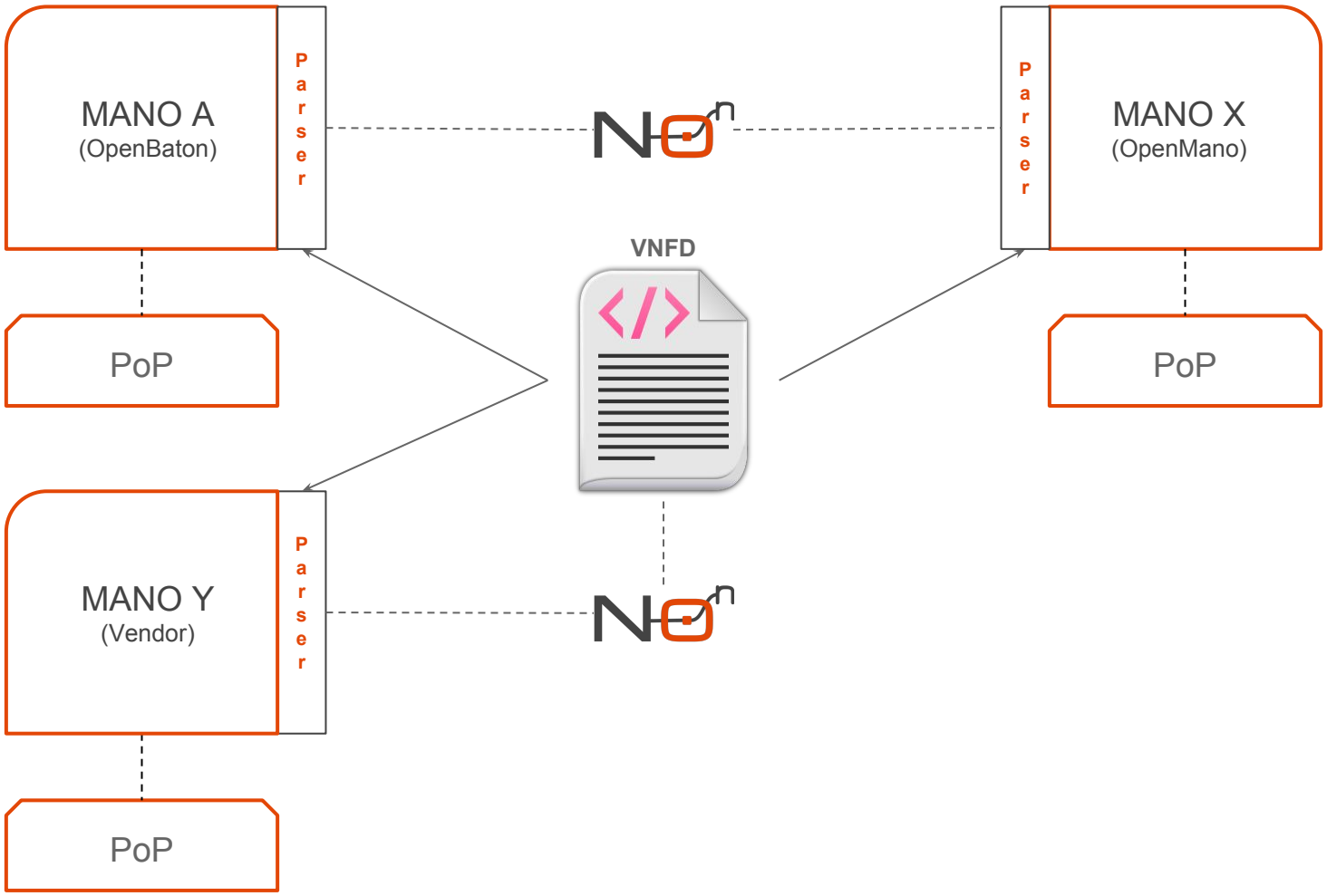
OpenBaton

- All elements belonging to ETSI specifications were able to be mapped, proprietary elements do not
- Some elements can be mapped using other components of the ontology
- Follows ETSI specifications in the syntax
- Easy to map

OpenMano

- Few elements belonging to ETSI specifications were able to be mapped, proprietary elements do not
- Some elements can be mapped using a manual intervention over the semantic file
- Does not follows ETSI specifications in the syntax
- Difficult to Map

Use Case I: First Concept



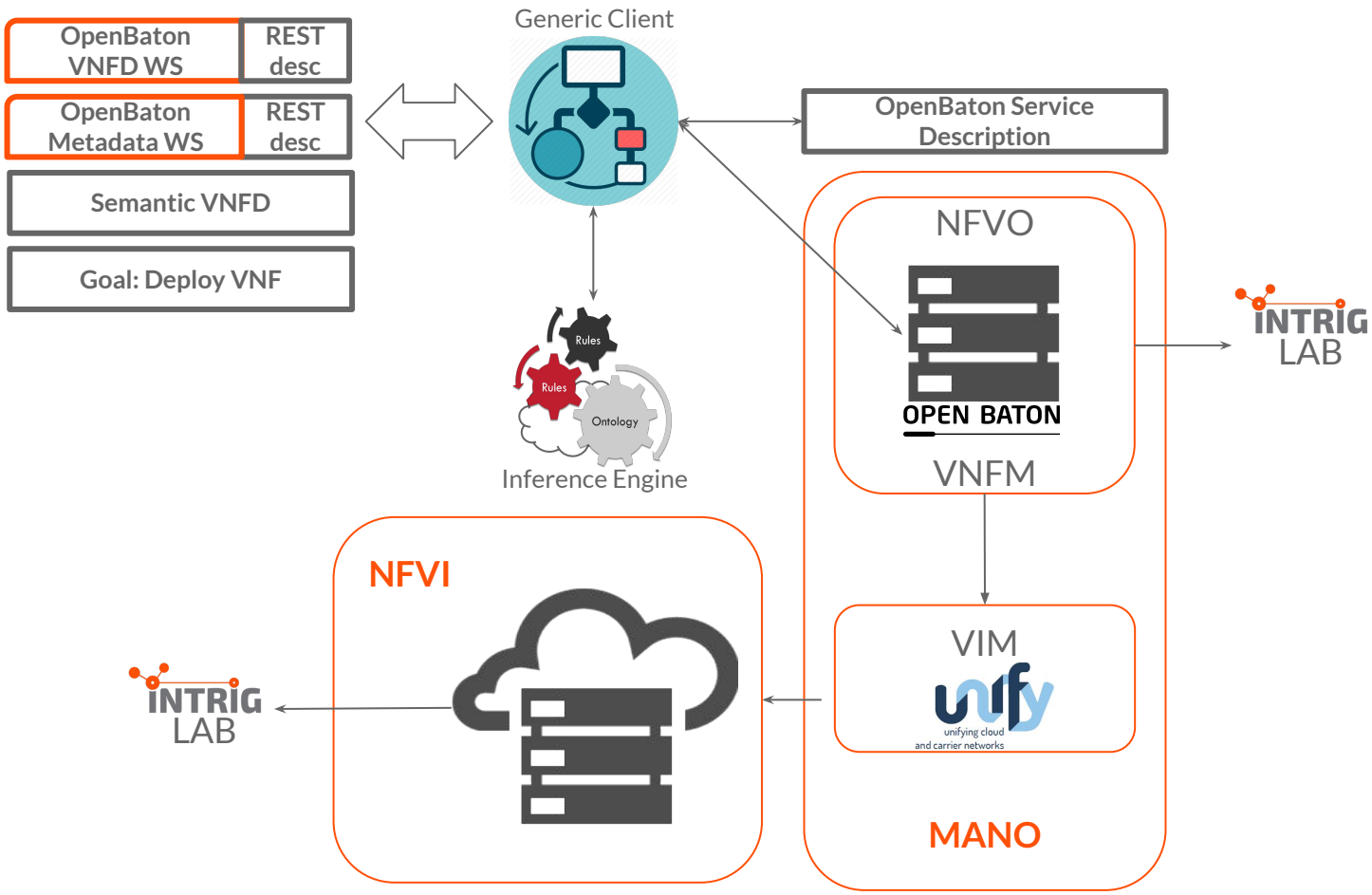
Use Case II: Results

	Goal	Expected Result	Expected vs Obtained Results	Modifications
Scenario I	Consume OpenBaton SnS	OpenBaton SnS Consumed	Service not Consumed: VIM component missed	VIM component was added manually
Scenario II	Consume OpenMano SnS	OpenMano SnS Consumed	Service not Consumed: several components missed	Other components of NOn were added manually, components outside of NOn were deleted
Scenario III	Consume any SnS	SnS Not Consumed	OpenMano service was consumed	VIM component was added manually

Use Case III

OpenBaton - Unify Integration Proposal

Scenario


OpenBaton VNFD WS
REST desc
OpenBaton Metadata WS
REST desc
Semantic VNFD
Goal: Deploy VNF
Generic Client
OpenBaton Service Description
Rules
Rules
Ontology
Inference Engine
NFVO
OPEN BATON
VNFM
INTRIG LAB
NFVI
INTRIG LAB
VIM
unify
unifying cloud and carrier networks
MANO

Minicons Free Vector Icons Pack by Webalys is published under a Creative Commons Attribution license and Free for both personal and commercial use. You can copy, adapt, remix, distribute or transmit it. If you use this set on your presentation remember to **keep the "Credits" slide or provide a mention** of this "Minicons Free Vector Icons Pack" and a link back to this page: http://www.webalys.com/minicons