ETSI NFV SPECFEST

Hands on activities with NFV SOL specifications and the ETSI Forge

Presented by Michele Carignani (ETSI) for ETSI NFV#19 – Denver – Sept 11th 2017

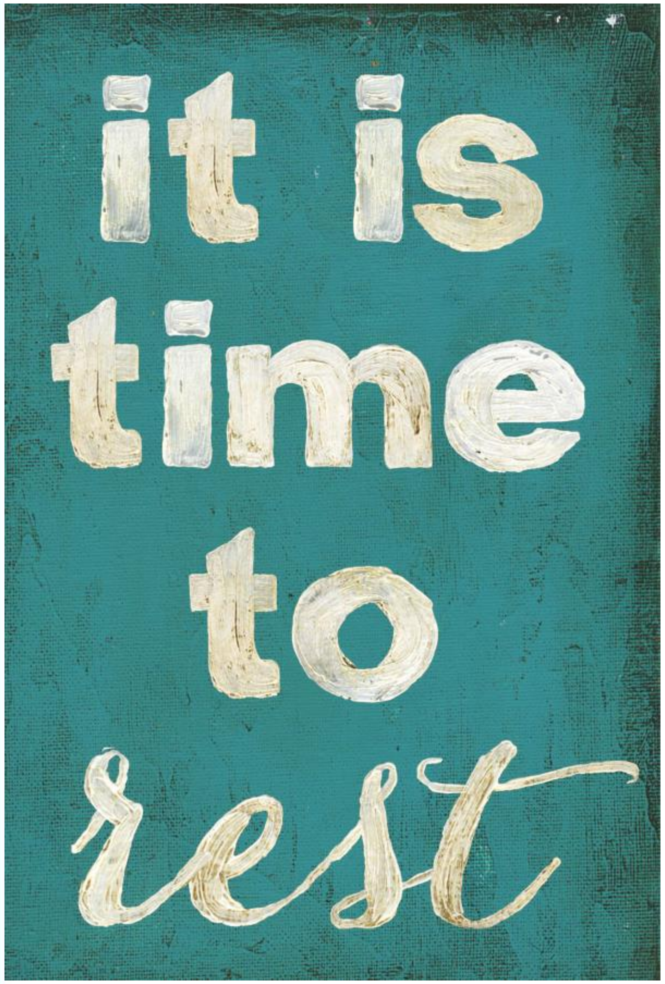
© ETSI 2017. All rights reserved

World Class Standards

NFV STAGE 3 APIS AND THE OPENAPI FILES

© ETSI 2017. All rights reserved

RESTful APIs and description Languages



- RESTful: **resources** accessed with few **uniform operations**

- **Main ingredients**
  - A tree of resources (**paths**)
  - Supported operations (**methods**) for each resource
  - Exchanged **payloads** (parameters and request/response bodies)
  - (Plus authentication, headers, …)

© ETSI 2017. All rights reserved

RESTful APIs are simple to use but…

How to…

Design the API in a **collaborative** (distributed) way?

Document them in a **portable** way?

Avoid **boilerplate** code around them?

Keep documentation and implementations **aligned**?

Manage and support different **versions**?

- How about a **formal language** to define and API to enable
  - Automatic documentation and code **generation**
  - **Version control** on API "blueprints"
- Several initiatives to define an **API description language**
  - WADL, RAML, OpenAPIs (Swagger), API Blueprint, Odata, RSDL, ….

© ETSI 2017. All rights reserved

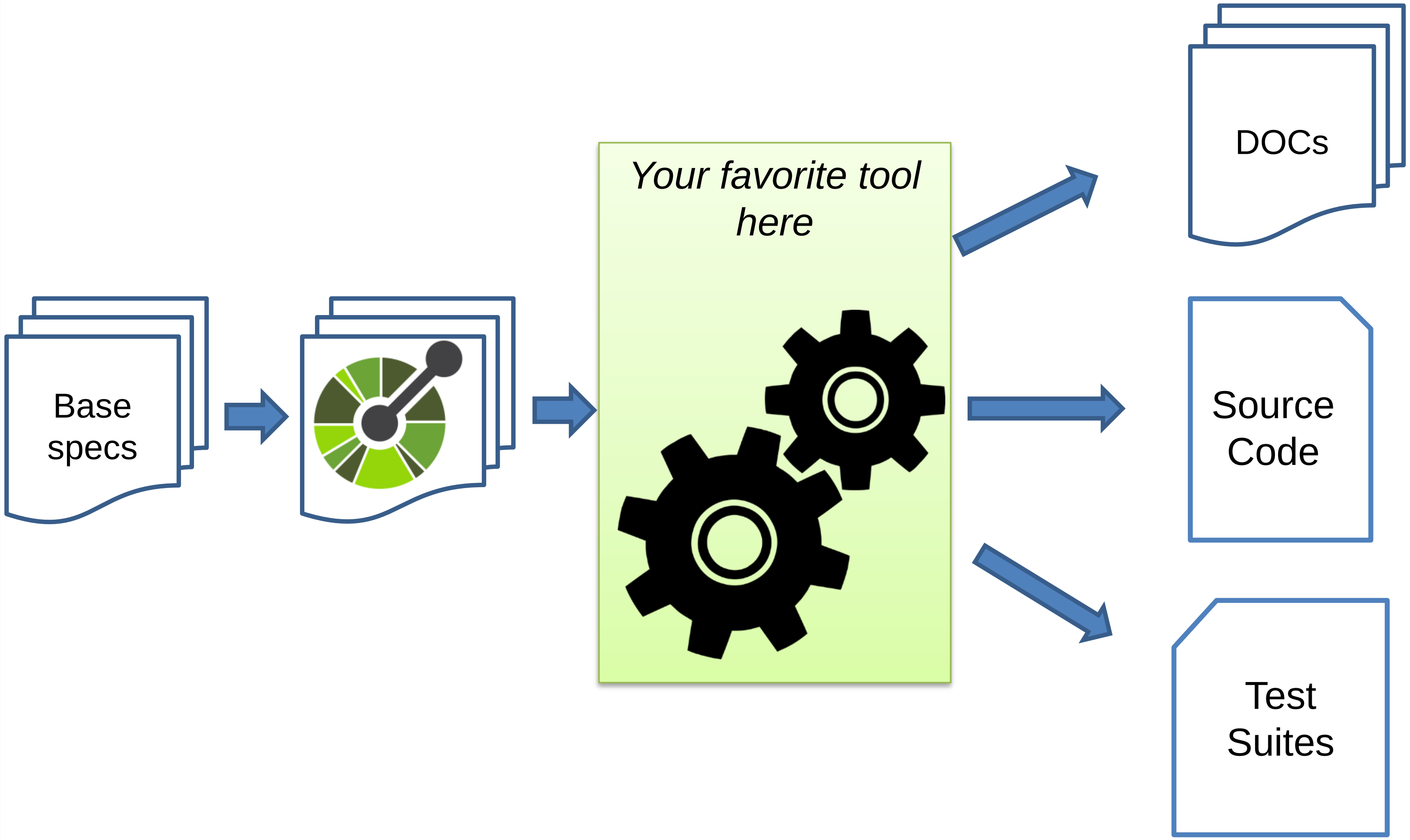
Open API Specification (OAS) Language

www.openapis.org

- A.k.a Swagger (previous name)
- Now an initiative under the Linux Foundation
- Machine readable specification of RESTful APIs
- Syntax
  - Tree based structure
  - JSON based, can be described via YAML
  - Reference for version 2.0
- De facto standard
  - Lively community of users and tooling developers
  - RAML main contributors recently joined OpenAPI initiative (providing a converter tool among the languages)

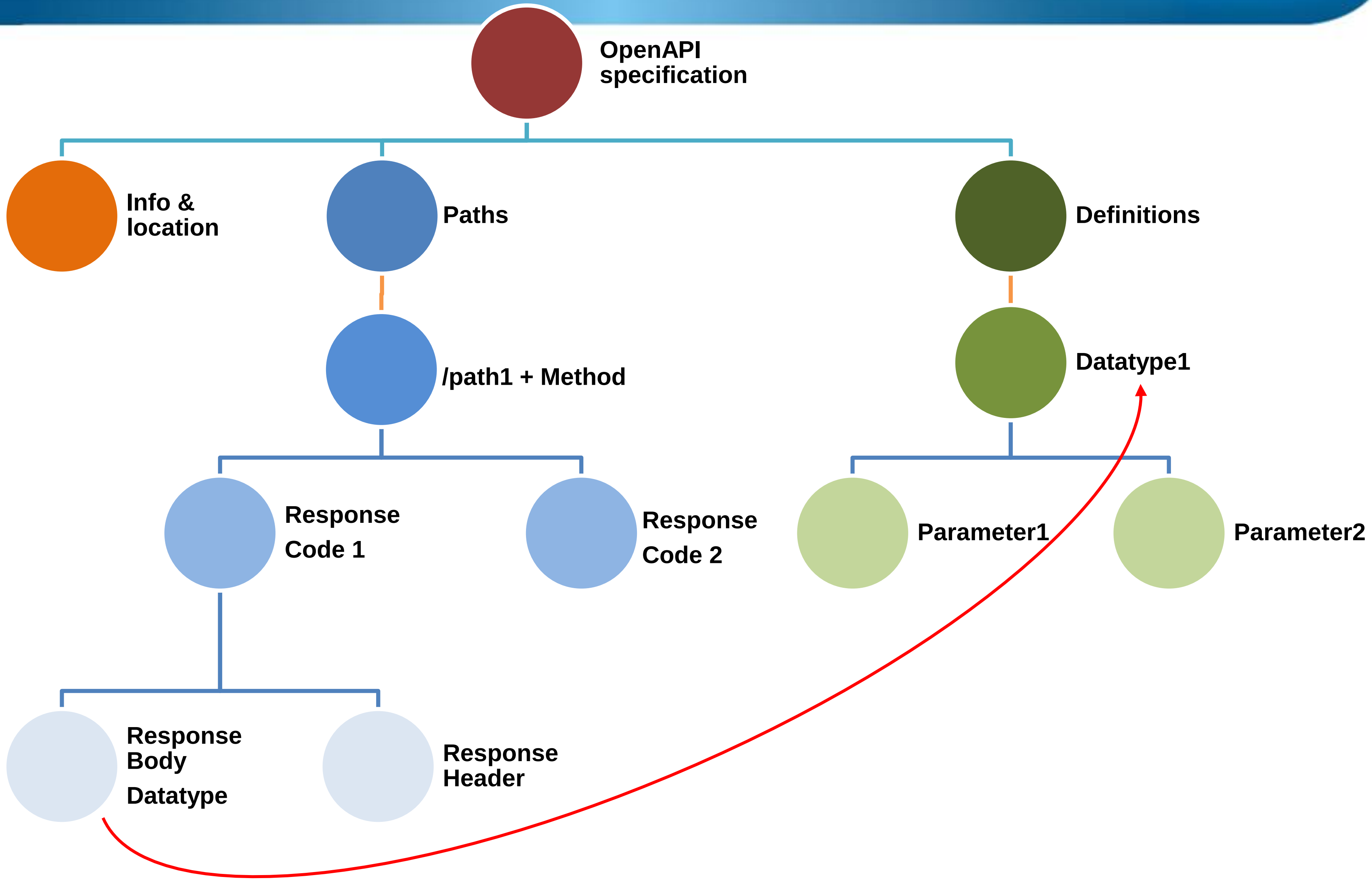
© ETSI 2017. All rights reserved

OpenAPIs on NFV SOL APIs. Why?



© ETSI 2017. All rights reserved

OAS Quick intro: Structure of a definition



© ETSI 2017. All rights reserved

OAS Quick intro: Top level properties

- **Swagger**: Language version (Required)
- **Info**: API Metadata (Required)
- **Host**: API hostname or IP
- **basePath**: API basepath
- **Paths**: Resource tree exposed (Required), each paths contains
  - **Operations**: HTTP methods supported, each contains
    - **Responses:** HTTP status, headers and payloads of each possible response
- **Definitions**: Data types produced and consumed
  - Basic types: *string*, *integer*, *long*, *boolean*, *date*, *dateTime*, *password*…
  - Complex types: *object* (with *properties*) and *arrays* (with *items*)

© ETSI 2017. All rights reserved

YAML Syntax basics

- https://en.wikipedia.org/wiki/YAML
- **Whitespace indentation** gives scope to parts of documents
- **Comments** begin with # and end with the line return/feed
- **Ordered sequences** have each item prefixed with "- "
- **Maps** (lists of key-value pairs) use ":"

```yaml
# YAML example 1
To-buy-list:
  - apples:
      color: red
  - ice-cream:
      flavor: lemon
      num: 5
```

© ETSI 2017. All rights reserved

Vnflcm example: API general information

```
swagger: '2.0'
info:
  title: NFV SOL 002 SpecFest - EXAMPLE
  version: v1
host: forge.etsi.org
basePath: /vnflcm
paths:
  /vnf_instances:
      post:
        description: Creates a new VNF instance res
        parameters:
          - name: createVnfRequest
            in: body
            description: VNF creation parameter
            schema:
              $ref: '#/definitions/CreateVnfRequest
            required: true
        responses:
          '201':
            description: Created successfully.
            schema:
              $ref: '#/definitions/VnfInstance'
definitions:
  VnfInstance:
    description: TBD
    required:
      - id
      - vnfdId
    properties:
      id:
        description: Identifier of the VNF instance
        type: string
      vnfdId:
        description: The VNFD on which the VNF instance is based.
        type: ’#/definitions/Identifier’
```

```
# GENERAL INFORMATION OF THE API
swagger: '2.0'
info:
  title: NFV SOL 002 SpecFest - EXAMPLE
  version: v1
host: forge.etsi.org
basePath: /vnflcm
```

© ETSI 2017. All rights reserved

Vnflcm example: Resources and operations

```
swagger: '2.0'
info:
  title: NFV SOL 002 SpecFest - EXA
  version: v1
host: forge.etsi.org
basePath: /vnflcm
paths:
  /vnf_instances:
      post:
        description: Creat
        parameters:
          - name: createVnfReq
            in: body
            description: VNF creati
            schema:
              $ref: '#/definitions/
            required: true
        responses:
          '201':
            description: Created su
            schema:
              $ref: '#/definitions/
definitions:
  VnfInstance:
    description: TBD
    required:
      - id
      - vnfdId
    properties:
      id:
        description: Identifier of
        type: string
      vnfdId:
        description: The VNFD on which the VNF instance is based.
        type: ’#/definitions/Identifier’
```

```
# AVAILABLE RESOURCES AND OPERATIONS
paths:
  /vnf_instances:
    post:
      description: Creates a new VNF instance resource.
      parameters:
        - name: createVnfRequest
          in: body
          description: VNF creation parameter
          schema:
            $ref: '#/definitions/CreateVnfRequest'
          required: true
      responses:
        '201':
          description: Created successfully.
          schema:
            $ref: '#/definitions/VnfInstance'
```

Vnflcm example: Data definitions

```
swagger: '2.0'
info:
  title: NFV SOL 002 SpecFest - EXAMPLE
  version: v1
host: forge.etsi.org
basePath: /vnflcm
paths:
  /vnf_instances:
      x-etsinfv-description: Represents
      post:
        description: Creates a new VNF i
        parameters:
          - name: createVnfRequest
            in: body
            description: VNF creation pa
            schema:
              $ref: '#/definitions/Creat
            required: true
        responses:
          '201':
            description: Created suc
            schema:
              $ref: '#/definit
definitions:
  VnfInstance:
    description: TBD
    required:
      - id
      - vnfdId
    properties:
      id:
        description: Identifier of the V
        type: string
      vnfdId:
        description: The VNFD on which the VNF instance is based.
        type: ’#/definitions/VndIdentifier’
```

```
# DATATYPES (SCHEMAS) DEFINITIONS
definitions:
  VnfInstance:
    description: TBD
    required:
      - id
      - vnfdId
    properties:
      id:
        description: Identifier of the VNF
          instance
        type: string
      vnfdId:
        description: The VNFD on which the VNF
          instance is based.
        type: ’#/definitions/VndIdentifier’
```

© ETSI 2017. All rights reserved

OpenAPIs, Swagger and the others

THE LINUX FOUNDATION

OpenAPIs Language

swagger.io

SWAGGER

SMARTBEAR

Tools (OSS)

- **Codegen**
- **Editor**
- **UI**
- **… Others**

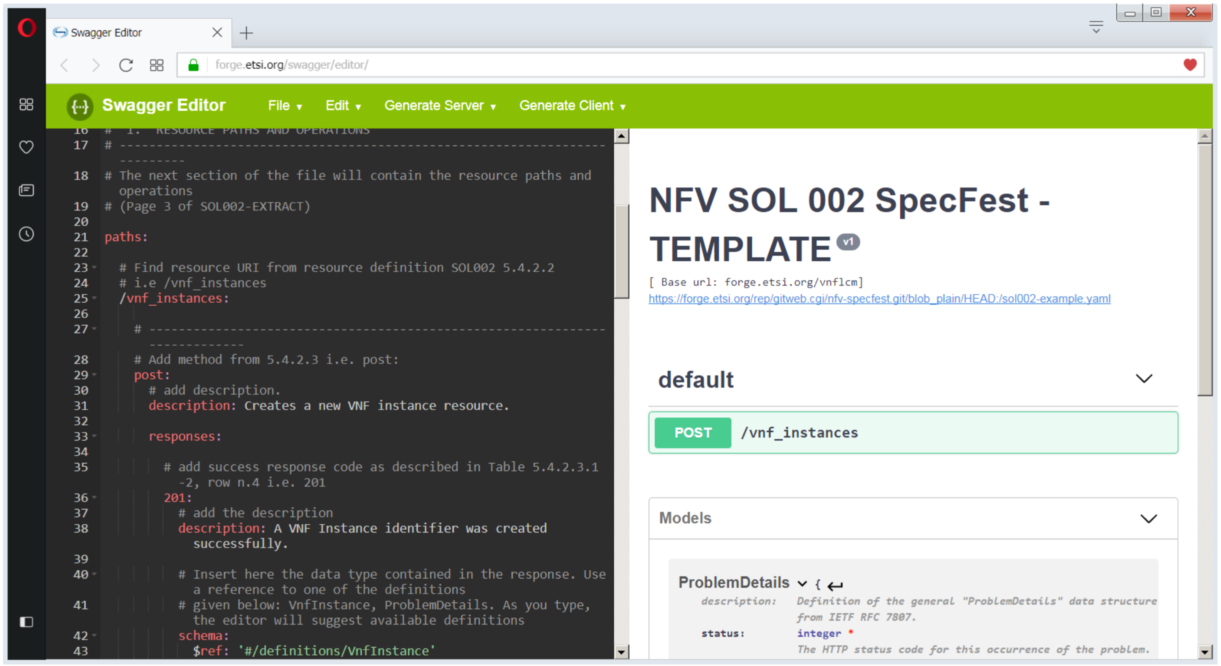
… Several Open Source and commercial tools (non exhaustive list)

Restlet

© ETSI 2017. All rights reserved

Editing online

- Swagger Editor: in browser editor for OAS
- Released under Apache 2.0
- Try it out at forge.etsi.org/swagger/editor
- Will be used in our hands on activity!



```yaml
#   I.   RESOURCE PATHS AND OPERATIONS
# ---------------------------------------------------------------------------
# The next section of the file will contain the resource paths and operations
# (Page 3 of SOL002-EXTRACT)

paths:

  # Find resource URI from resource definition SOL002 5.4.2.2
  # i.e /vnf_instances
  /vnf_instances:

    # ---------------------------------------------------------------------------

    # Add method from 5.4.2.3 i.e. post:
    post:
      # add description.
      description: Creates a new VNF instance resource.

      responses:

        # add success response code as described in Table 5.4.2.3.1-2, row n.4 i.e. 201
        201:
          # add the description
          description: A VNF Instance identifier was created successfully.

          # Insert here the data type contained in the response. Use a reference to one of the definitions
          # given below: VnfInstance, ProblemDetails. As you type, the editor will suggest available definitions
          schema:
            $ref: '#/definitions/VnfInstance'
```

NFV SOL 002 SpecFest - TEMPLATE v1

[Base url: forge.etsi.org/vnflcm]
https://forge.etsi.org/rep/gitweb.cgi/nfv-specfest.git/blob_plain/HEAD:/sol002-example.yaml

default

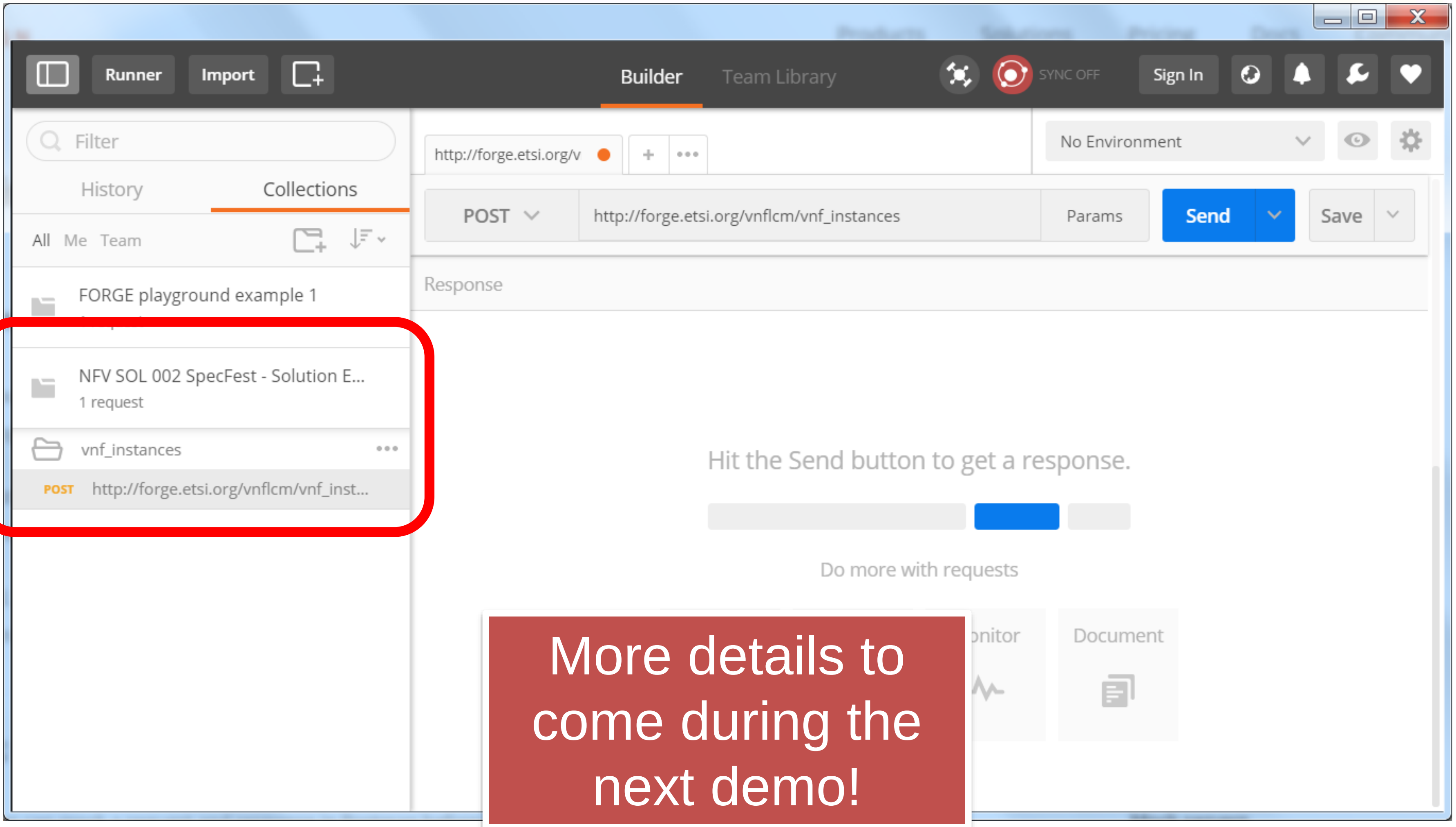
POST /vnf_instances

Models

ProblemDetails {
description: *Definition of the general "ProblemDetails" data structure from IETF RFC 7807.*
status: integer *
The HTTP status code for this occurrence of the problem.

© ETSI 2017. All rights reserved

Test suite generation example



FORGE playground example 1

NFV SOL 002 SpecFest - Solution E...
1 request

vnf_instances

POST http://forge.etsi.org/vnflcm/vnf_inst...

http://forge.etsi.org/v

POST http://forge.etsi.org/vnflcm/vnf_instances

Hit the Send button to get a response.

More details to come during the next demo!

© ETSI 2017. All rights reserved

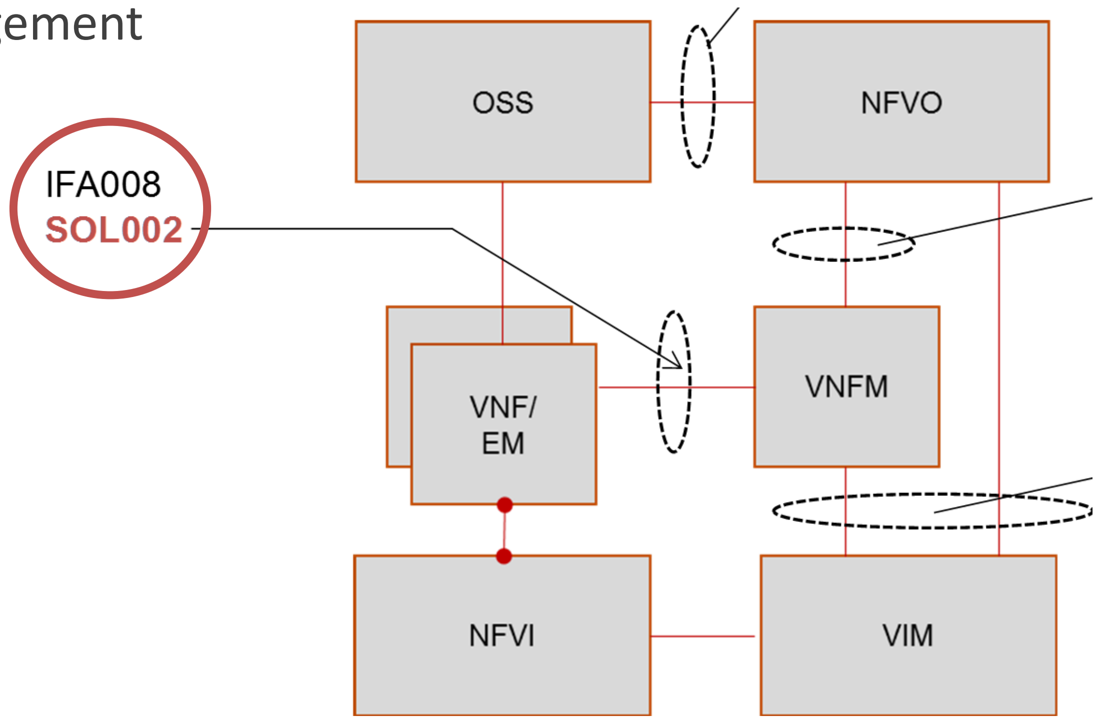
HANDS ON: OAS AT ETSI FORGE

How to create OAS files in a collaborative way at the ETSI FORGE

© ETSI 2017. All rights reserved

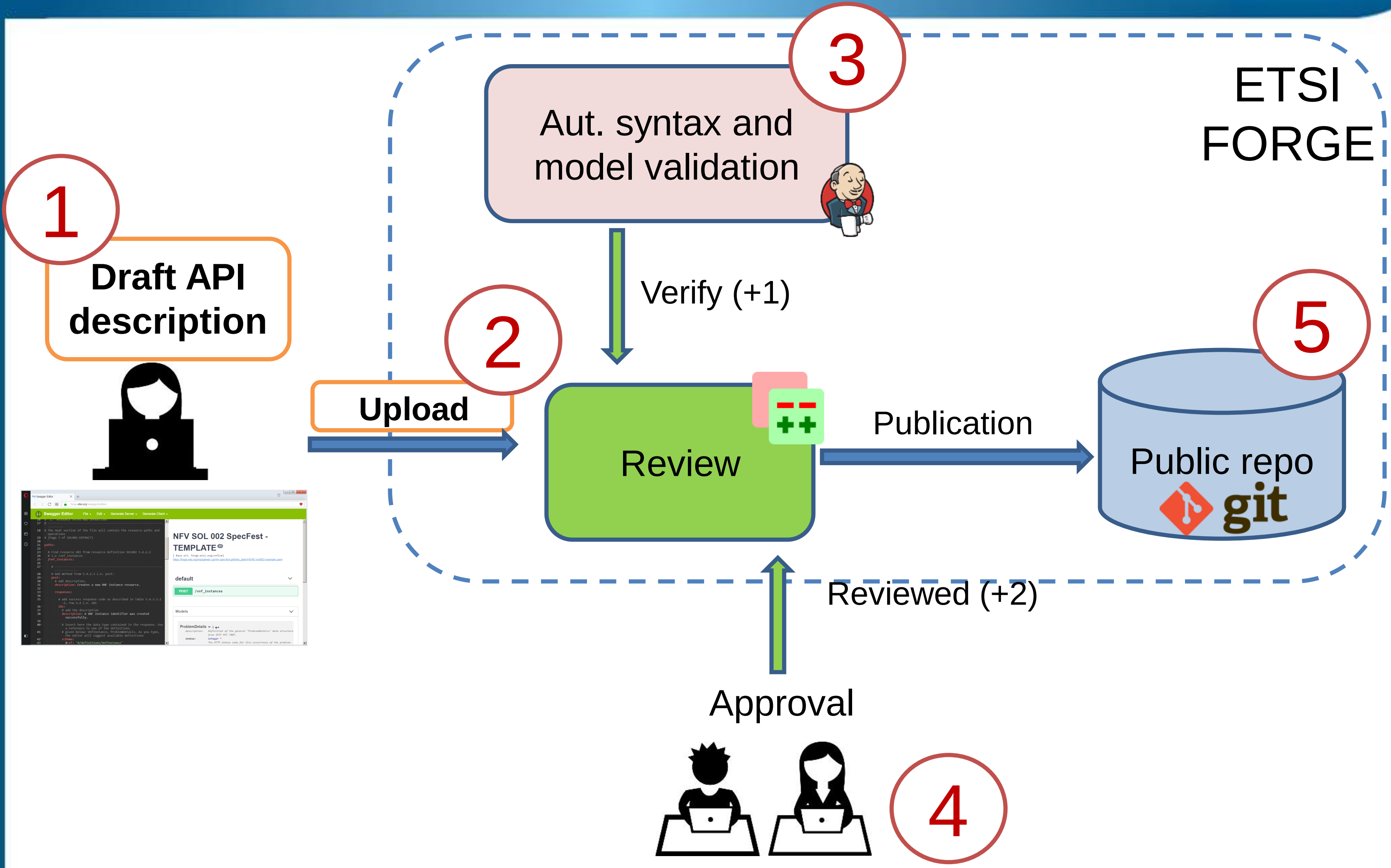
Case study: ETSI GS SOL 002

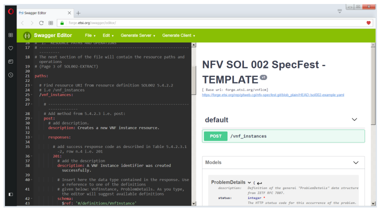
- Describes RESTful protocols specification for the Ve-Vnfm Reference Point
- Contains specification of 5 interfaces
  - **VNF Life Cycle Management**
  - VNF Performance Management
  - VNF Fault Management,
  - VNF Indicator
  - VNF Configuration



© ETSI 2017. All rights reserved

SpecFest workflow





```yaml
# ----------------------------------------------------------------
# The next section of the file will contain the resource paths and
  operations
# (Page 3 of SOL002-EXTRACT)

paths:

  # Find resource URI from resource definition SOL002 5.4.2.2
  # i.e /vnf_instances
  /vnf_instances:

    # ----------------------------------------------------------------

    # Add method from 5.4.2.3 i.e. post:
    post:
      # add description.
      description: Creates a new VNF instance resource.

      responses:

        # add success response code as described in Table 5.4.2.3.1
          -2, row n.4 i.e. 201
        201:
          # add the description
          description: A VNF Instance identifier was created
            successfully.

          # Insert here the data type contained in the response. Use
            a reference to one of the definitions
          # given below: VnfInstance, ProblemDetails. As you type,
            the editor will suggest available definitions
          schema:
            $ref: '#/definitions/VnfInstance'
```

NFV SOL 002 SpecFest - TEMPLATE v1

[Base url: forge.etsi.org/vnflcm]

https://forge.etsi.org/rep/gitweb.cgi/nfv-specfest.git/blob_plain/HEAD:/sol002-example.yaml

default

POST /vnf_instances

Models

ProblemDetails {
description: Definition of the general "ProblemDetails" data structure from IETF RFC 7807.
status: integer *
The HTTP status code for this occurrence of the problem.

© ETSI 2017. All rights reserved

Outlined of the proposed task

1. Read the detailed instructions on the **wiki page**
2. Open **online editor** (link provided in the wiki) and **fill in the required information** taken from the SOL002 extract provided. *Leave the browser window/tab open when you are finished!*
3. **Upload** the produced specification to the ETSI Forge
4. Wait for the automatic validator to execute and **verify** your solution. Fix any errors and submit again if needed.

© ETSI 2017. All rights reserved

Live demo

Navigate to

forge.etsi.org/specfest

© ETSI 2017. All rights reserved

Contact Details:

Michele Carignani, Silvia Almagia

Centre for Testing and Interoperability, ETSI

michele.carignani@etsi.org

silvia.almagia@etsi.org

ANY QUESTIONS?

Thank you!

© ETSI 2017. All rights reserved