Parser List

- **Base Parser**
  - nfv-toscaparser
  - JTosca

- **Wrapper Parser**
  - SDC-TOSCA (base on JTosca)
  - Genericparser(base on nfv-toscaparser)
The TOSCA Parser is an OpenStack project and licensed under Apache 2. It is developed to parse TOSCA Simple Profile in YAML. It reads the TOSCA templates and creates an in-memory graph of TOSCA nodes and their relationship.

- Support: TOSCA 1.1
- Language: Python
- Repo: https://gerrit.opnfv.org/gerrit/#/admin/projects/parser
- How to use: As Lib
JTosca

• ONAP JTOSCA is delivered as helper JAR that can be used by clients that work with TOSCA CSAR files. It parses the CSAR and returns the model object which represents the CSAR contents. Prior to that, it performs validations on the CSAR to check its TOSCA compliance.

• Support: TOSCA 1.1

• Language: Java

• Repo: https://gerrit.onap.org/r/#/admin/projects/sdc/jtosca

• How to use: As Lib
SDC-TOSCA

- ONAP SDC-Tosca is delivered as helper JAR that can be used by clients that work with SDC TOSCA CSAR files. It parses the CSAR and returns the model object which represents the CSAR contents, through designated function calls with SDC flavour.
- It uses the underlying generic JTOSCA parser.
- Language: Java
- Repo: https://gerrit.onap.org/r/#/admin/projects/sdc/sdc-tosca
- How to use: As Lib
Genericparser

- It provides parser service as well as basic package management service by Micro Service. It uses the underlying \texttt{nfv-toscaparser}.
- Language: Python
- Repo: \url{https://gerrit.onap.org/r/#/admin/projects/modeling/genericparser}
- How to use: as Micro Service
- Use case support: CCVPN in Dublin
Generic Parser Architecture

- **VFC**
- **UUI**
- **3rd External System**

**Generic Parser**

- **Package Management Service**
- **Parser Service**

**REST APIs**

- **Upload external package**
- **Fetch package From SDC**
- **Parse TOCSA model**

**Packages**

- **SDC catalog**
- **nfv-toscaparser**

**Onboard packages & parse**

**Parser Service**

- **POST /api/parser/v1/parser**
- **POST /api/parser/v1/parser/manifest**
- **POST /api/parser/v1/parser/upload**
- **POST /api/parser/v1/parser/parse**

**Service package API**

- **POST /api/parser/v1/service_packages**
- **GET /api/parser/v1/service_packages**
- **GET /api/parser/v1/service_packages/{cosnrId}**
- **DELETE /api/parser/v1/service_packages/{cosnrId}**

**NS package API**

- **VNF package API**
- **PNF package API**
Parser Service

- Onboarding package from SDC
  ```
curl -X POST \
http://172.60.3.65:30280/api/parser/v1/service_packages \
-H 'Content-Type: application/json' \
-d '{
   "csarId": "0734e398-a427-49f2-9abe-de8eb02542ad"
}'
```

- Parse package
  ```
curl -X POST \
http://172.60.3.65:30280/api/parser/v1/parser \
-H 'Content-Type: application/json' \
-d '{
   "csarId": "0734e398-a427-49f2-9abe-de8eb02542ad",
   "inputs": "",
   "packageType": "Service"
}'
```

Parser result in JSON format
Package Management Service

• Provide basic pkg management function in Dublin.
  - Fetch package from SDC Catalog
  - Package storage

• Advantage
  - Unified Storage
  - Unified Management
  - Reduce duplication development
Parser in E/F Version

- Parser process optimization

D (Synchronized call)

E/F (Asynchronous call)

- Provide support for VNFSDK
  - VNFSDK has plan to migrate the current python lib usage to parser service
  - Requirement is in discussion

- Call for requirements from other projects
VF-C plans to open its Catalog capabilities, including ETSI Catalog DB and ETSI NS/VNF/PNF pkg management functions and APIs.

Combine the current basic service pkg management function with VF-C Catalog function into one unified Catalog.
Catalog in E/F Version

- SO ETSI Catalog requirements

  The unified catalog can support the ETSI catalog requirements, **suggest SO can use the unified catalog**.

- Catalog requirements from other projects
Thanks