

Gitlab-ci under steroids: chained-ci with ONAP usage illustrations

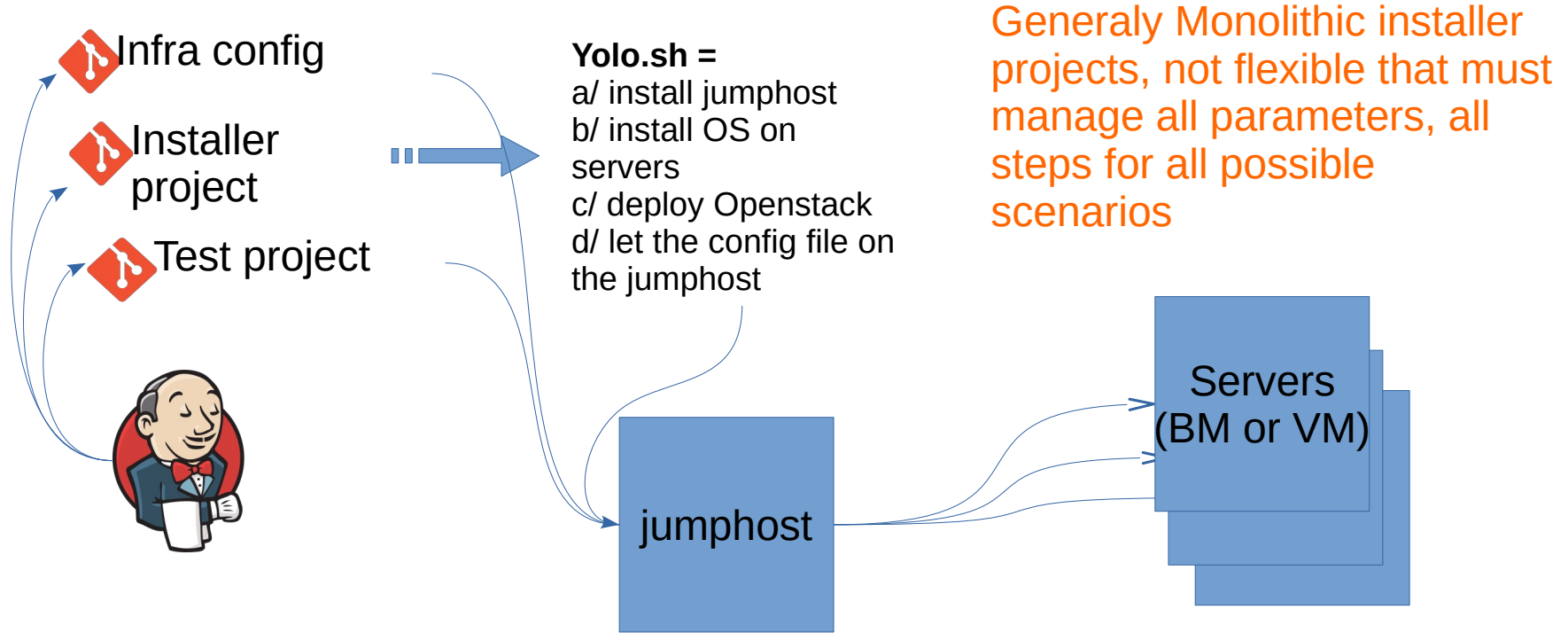
CDF Interoperability SIG Meetings - 2020.03.05



Orange and CI/CD

- The need is to replace many baremetal network functions
 - Deploy more and more VNFs
 - From more simple services, to complex ones, from firewall to a full 5G network
- We must deal with many vendors
- We must deal with existing networks:
 - Orange is an historical company
 - dealing with existing and heterogeneous networks, from circuit switch network to 5G
 - With heterogeneous competencies
 - We must deal with actual competencies, and keep things as much simple as possible

REX on auto deploy of a Telco IaaS by OPNFV



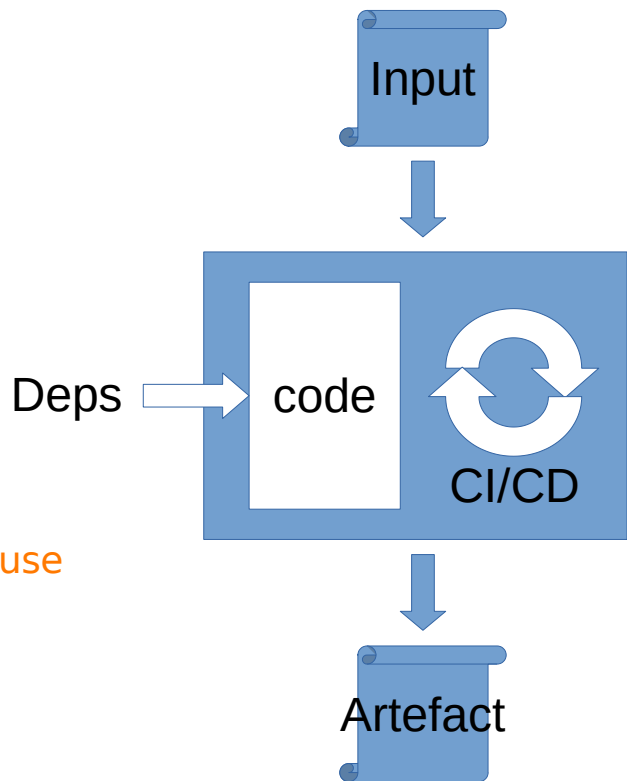
Jenkins is managed by a small group of developpers and the projects do not manage easily their way of doing CI or CD

Our needs

- Creation of hardware ressources can be long, complex, and may be the base of further deployments
- Need to chain severals steps/projects
- Why Gitlab-ci
 - manage CI/CD simply and natively, with artifact handling
 - reference source code manager in Orange
- But Gitlab-CI is strongly mono-project oriented
 - Even if triggers exists... but they do not manage easy artifacts passing from a project to the other.

Back to basis, what is an agnostic project with CI/CD

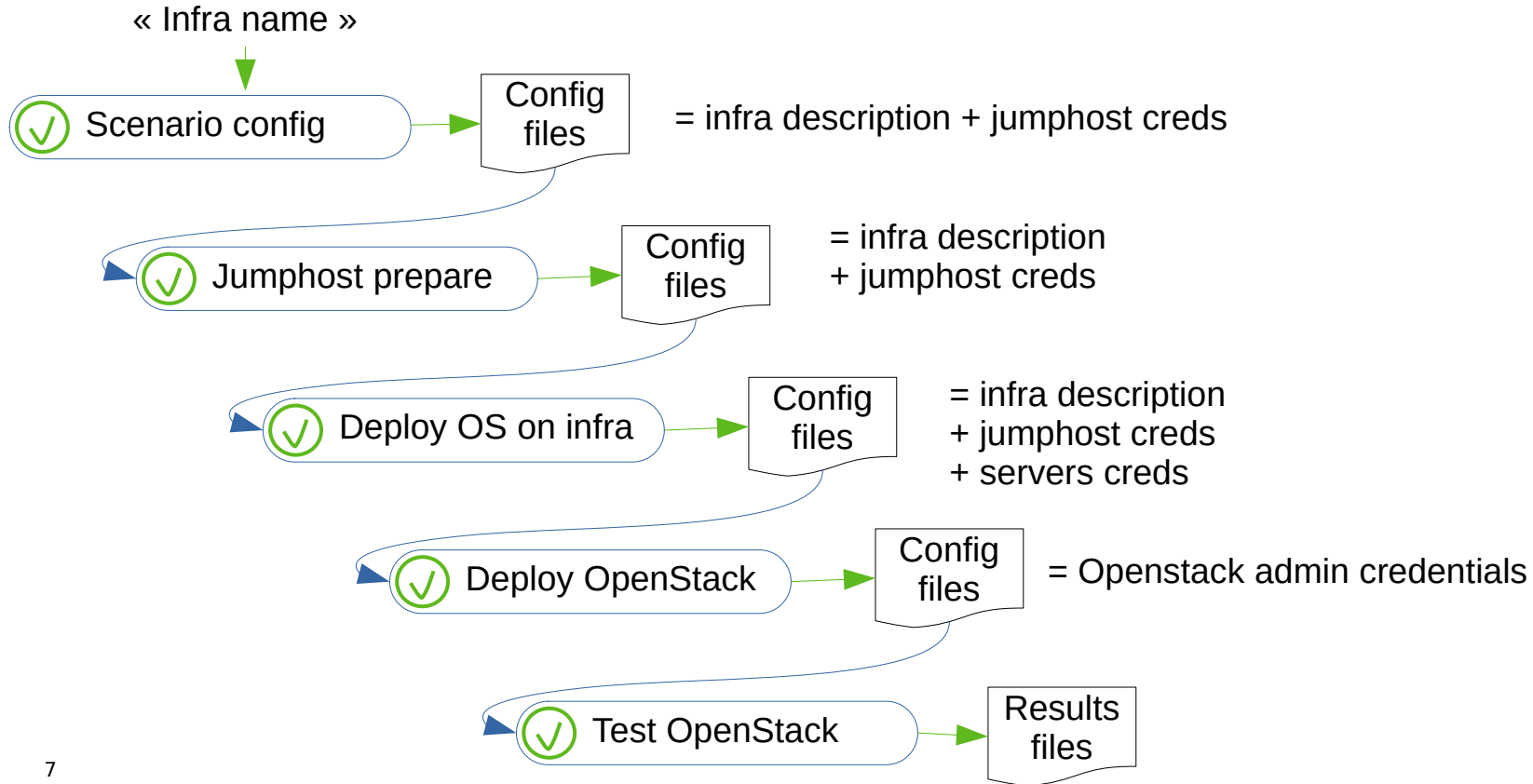
- Some configuration files and parameters
 - To give the environment configuration
 - Resources to use (infra, services XaaS...)
 - Options
 - To give the specificities to access the resources
 - Authentication / Certificates
- Code and dependencies list
- A list of CI/CD jobs specifics to the project
- The CD result as an artefact archive = A set of data to use the deployed resources.
 - login/password/certificat
 - Url
 - Build results



Our choice

- To break any big monolithic project into a set of micro projects
 - With only one function with a fixed perimeter.
 - Each project is responsible of his code and the way it deploys its resources
 - It is easier to manage dev versions on a micro-project than on a monolithic one
 - 1 CI/CD config managed by the project
 - To let developers manage the CI/CD jobs linked to the project
 - To avoid having to refer to a meta project managing all CI/CD jobs
 - To simply replace a project by another doing the same function
- To make the results of a project being the input of the next projects :
 - To gain flexibility in scenarios
 - To avoid any configuration stored in a functional project

In theory

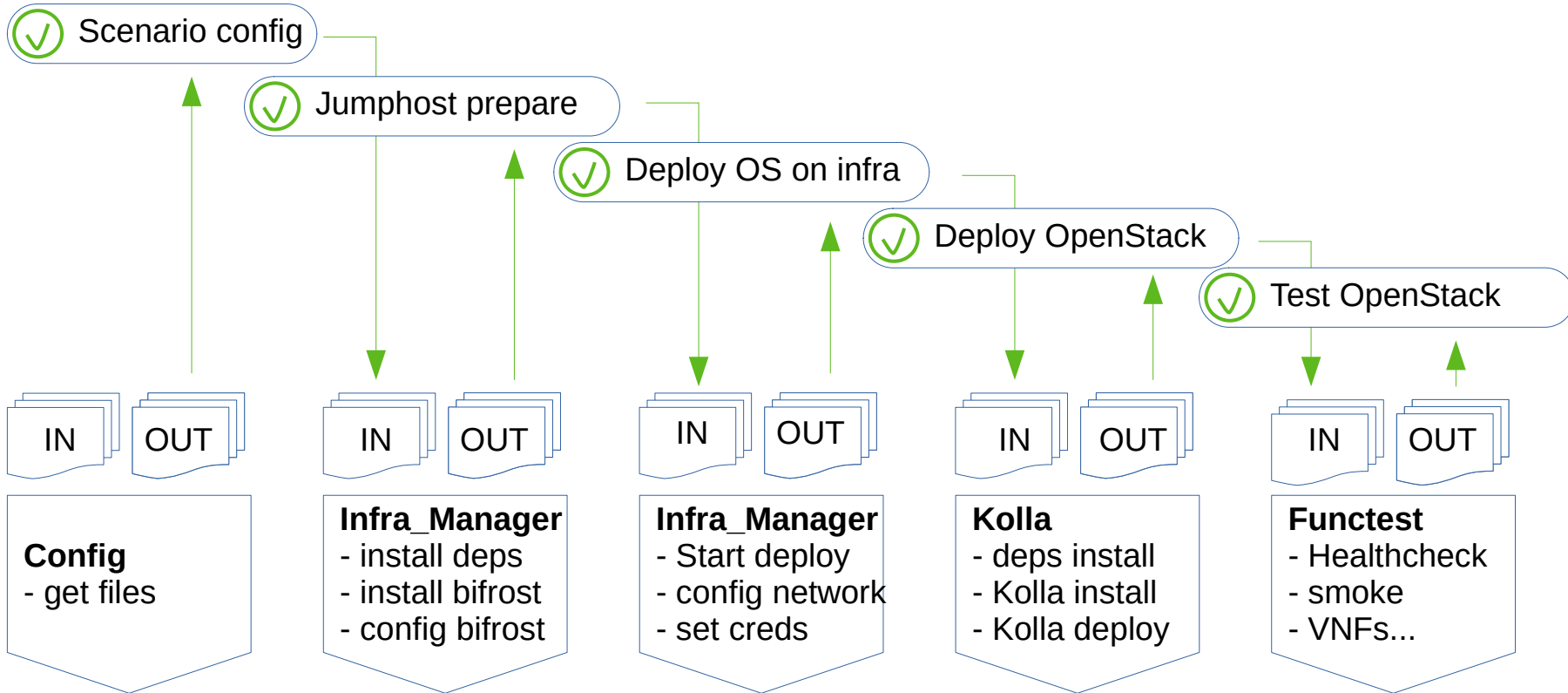


In reality



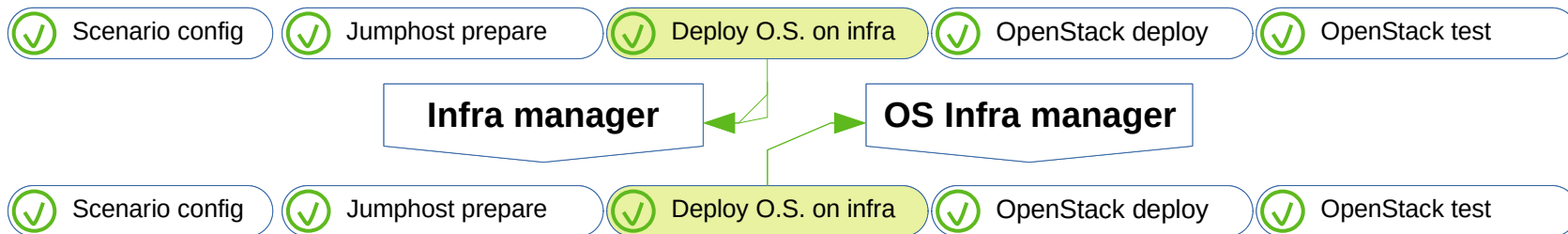
Scenario Manager

Projects CI



Chaining pipelines

- Deploy Openstack on VM or Baremetal



- Kubernetes deployment

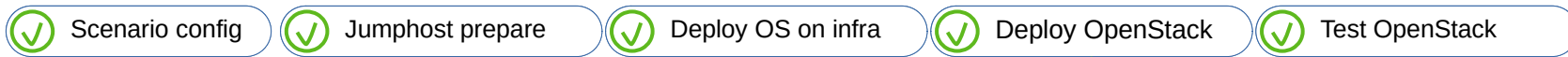


- ONAP Deployment



And chaining pipelines of pipelines

- Openstack deployment on baremetal or VM



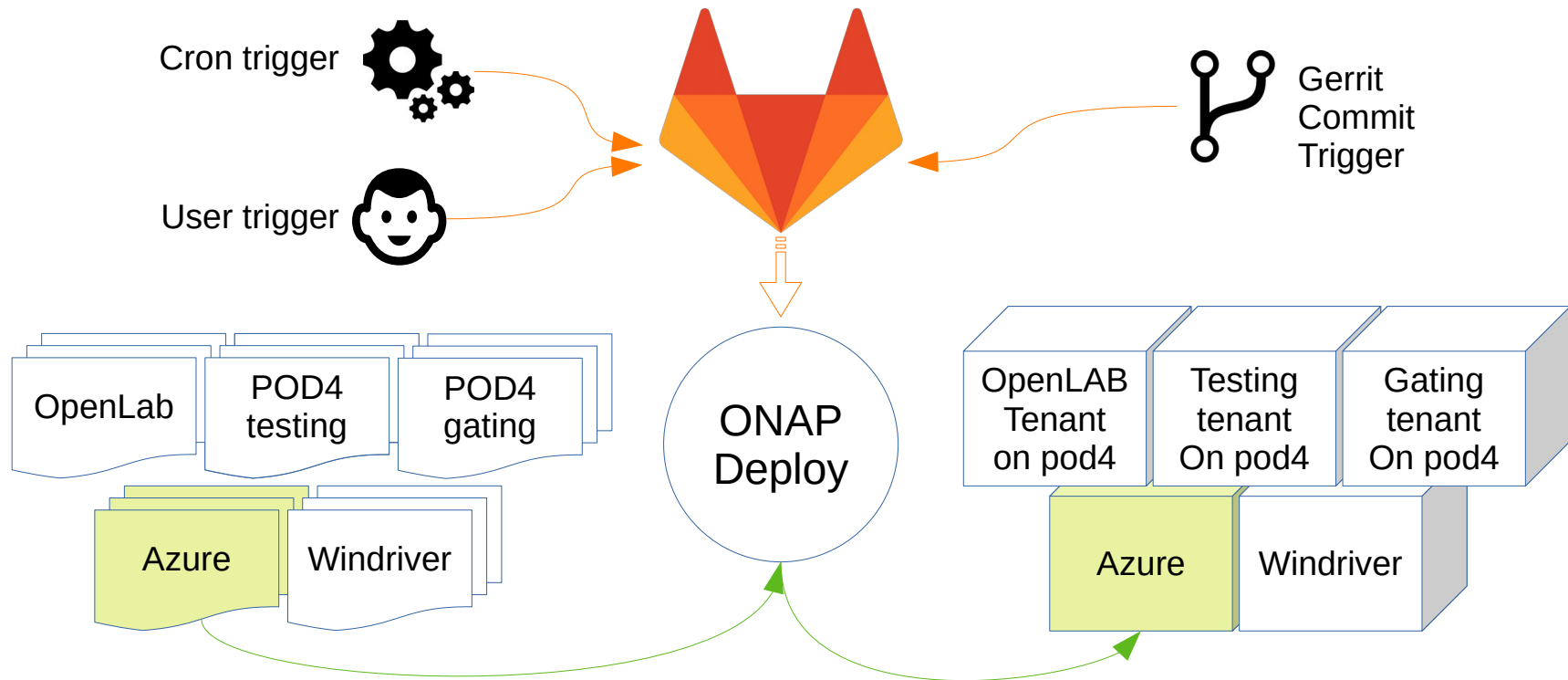
- Kubernetes deployment



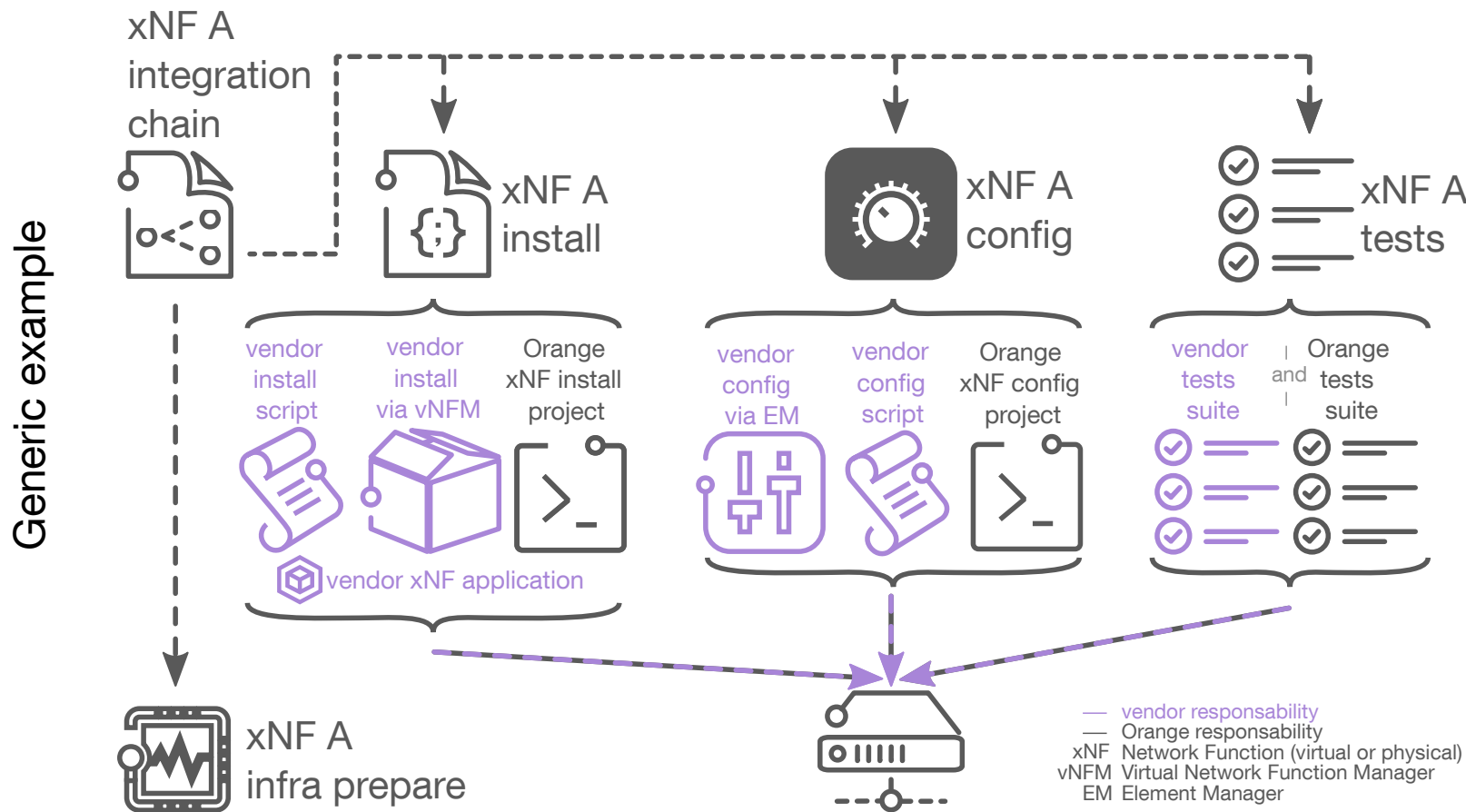
- ONAP Deployment



Deploy on multi targets, and enable gating.



Better integration of external projects



At the end, what is Chained-CI ?

- A scenario manager
- A simple Gitlab project
 - who triggers other gitlab projects
 - Who prepares a set of input data
 - Who fetches the scenario steps results to send them to following projects
- A set of scripts to help projects
 - Easy input data fetching, decrypt, SSH config and key preparation
 - Easy result encryption

Demo



Pros

- Only yaml configuration, no scripts on scenario side
- High modularity : easy function project swap
- No extra servers, only gitlab
 - Possibility to use shared runners
- Harmonize inputs/outputs of projects
- Decentralization of the CI/CD, each project manages its part
- Can trigger projects on multiple gitlabs
- Encryption of results: from a public gitlab, to a public or private cloud, securely

Cons

- Must update existing project to make it compatible (input/output management, parameters data structure...)
- Significant ticket fees:
 - Knowledge of gitlab required
 - Understanding of the use of the tool
- does not prevent the necessary understanding of the access path to the resources (proxy ...)

Possible evolutions :

- Enhance chained-ci
 - Version 2 in python will replace ansible version (testing phase)
 - Better Gitlab environment usage
 - Rebuild the interface with by real web developer :)
- Or just reuse the automated projects with another CD tool

Some links :

- Running chaines :
 - https://gitlab.com/Orange-OpenSource/lfnci_cd/chained-ci
- Core :
 - Chained-ci-roles : https://gitlab.com/Orange-OpenSource/lfnci_cd/chained-ci-roles/
 - Chained-ci-py : https://gitlab.com/Orange-OpenSource/lfnci_cd/chained-ci
- Some Triggered projects :
 - Infra : https://gitlab.com/Orange-OpenSource/lfnci_cd/infra
 - Onap : https://gitlab.com/Orange-OpenSource/lfnci_cd/onap
-

Thanks

For more information :

David Blaisonneau – Orange Labs

Mail : david.blaisonneau AT orange.com

Twitter: DBlaisonneau

