Software Provisioning Infrastructure (UMD/CMD)

Mario David (LIP), João Pina (LIP), Carlos Manuel (LIP), Jorge Gomes (LIP), Pablo Orviz (IFCA)

EGI Repositories

The EGI Software Repository provides access for the middleware distributions together with some Community Repositories

Three main categories:

- The Unified Middleware Distribution (**UMD**): redistribution of traditional middleware

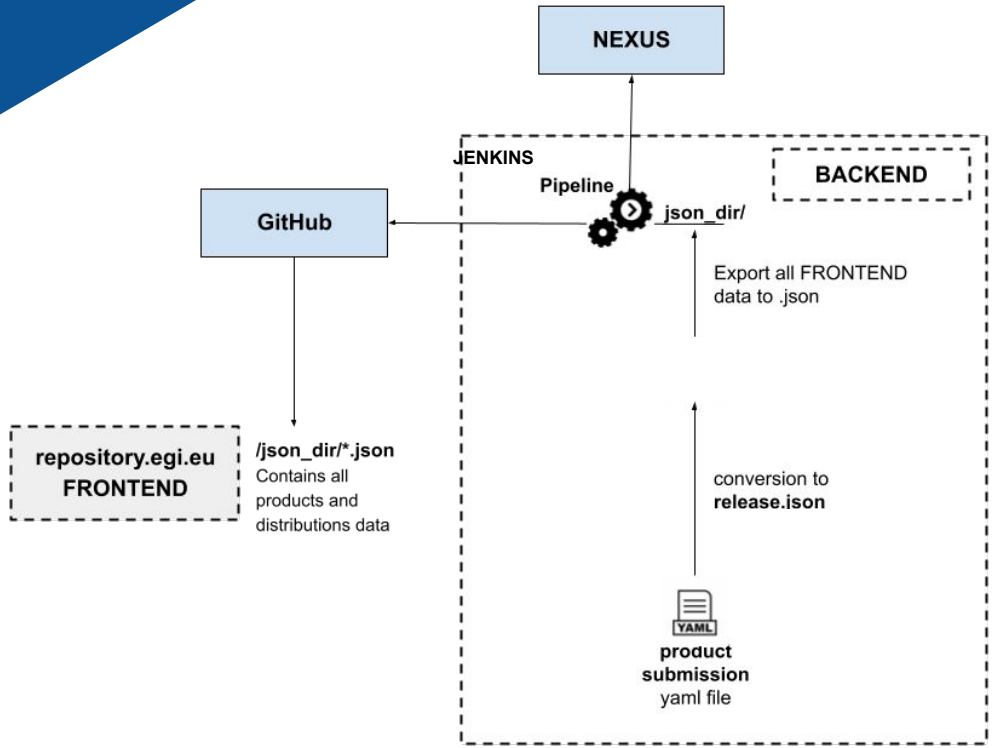
  OS Support
  - Centos 7 : **UMD-4**

  Provider:
  - IBERGRID since 2021

- Cloud Middleware Distribution (**CMD**): specific middleware for OpenStack and OpenNebula integration components

- **IGTF** Distribution of Authority Trust Anchors: packages with the trust anchor information

Current architecture



Future:

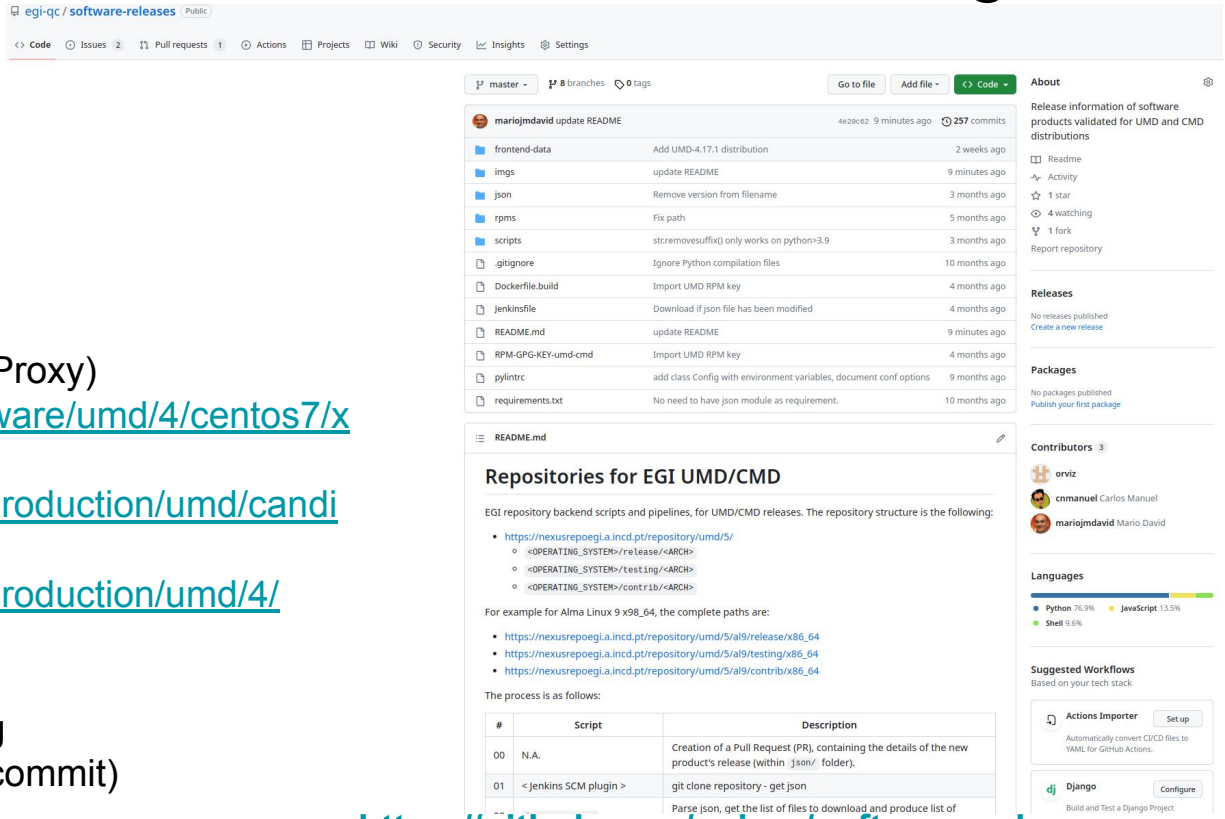
- Integration with SQAaaS from EOSC-Synergy
- Allow it to be used by external users:
  - EGI / EOSC marketplace

Tracking of the workflow all based on git

Frontend: https://repository.egi.eu/

New Functionalities:

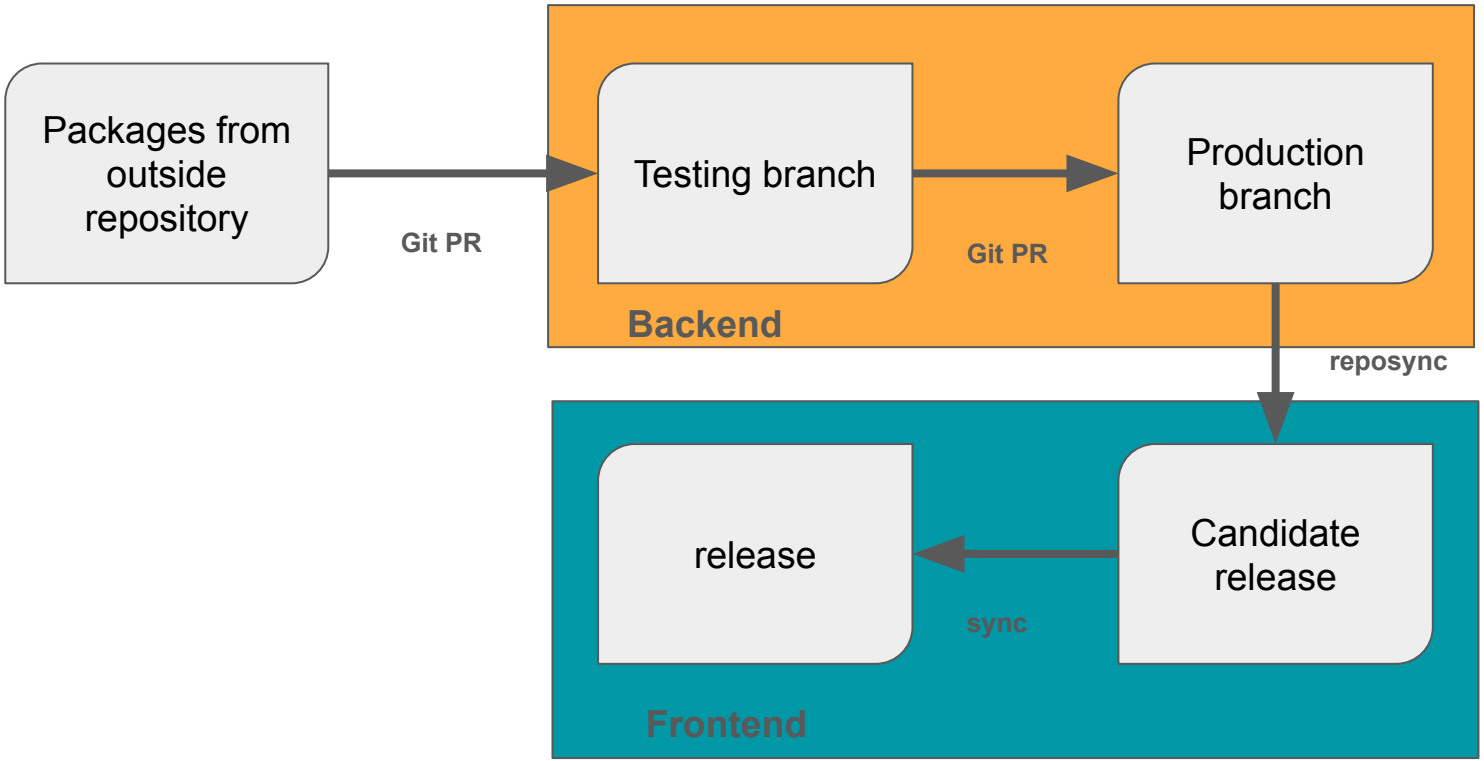
- IPV6 compliant
- New backend behind a proxy (HAProxy)
  - https://repository.egi.eu/software/umd/4/centos7/x86_64/ (prod new)
  - https://repository.egi.eu/sw/production/umd/candidate/4/ (candidate)
  - https://repository.egi.eu/sw/production/umd/4/ (prod)
- High Availability
- Full automatic process until testing
- Testing -> production (manual git commit)



https://github.com/egi-qc/software-releases

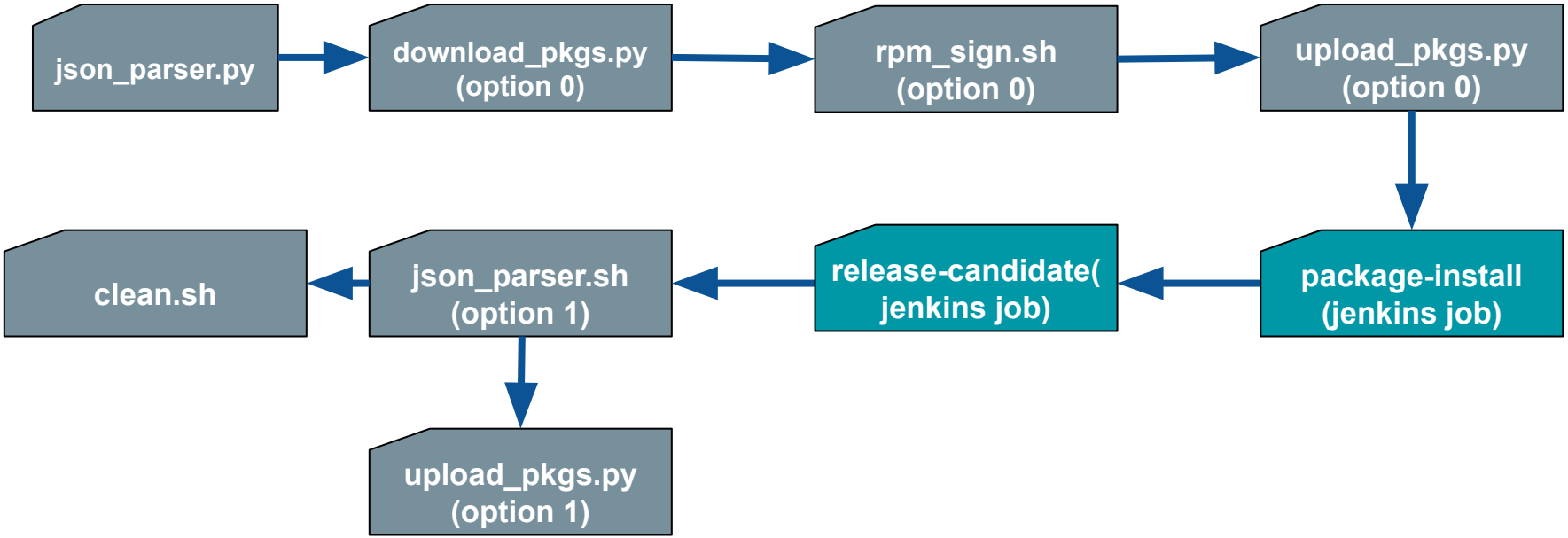
UMD/CMD Workflow

https://github.com/egi-qc/software-releases



UMD/CMD Workflow

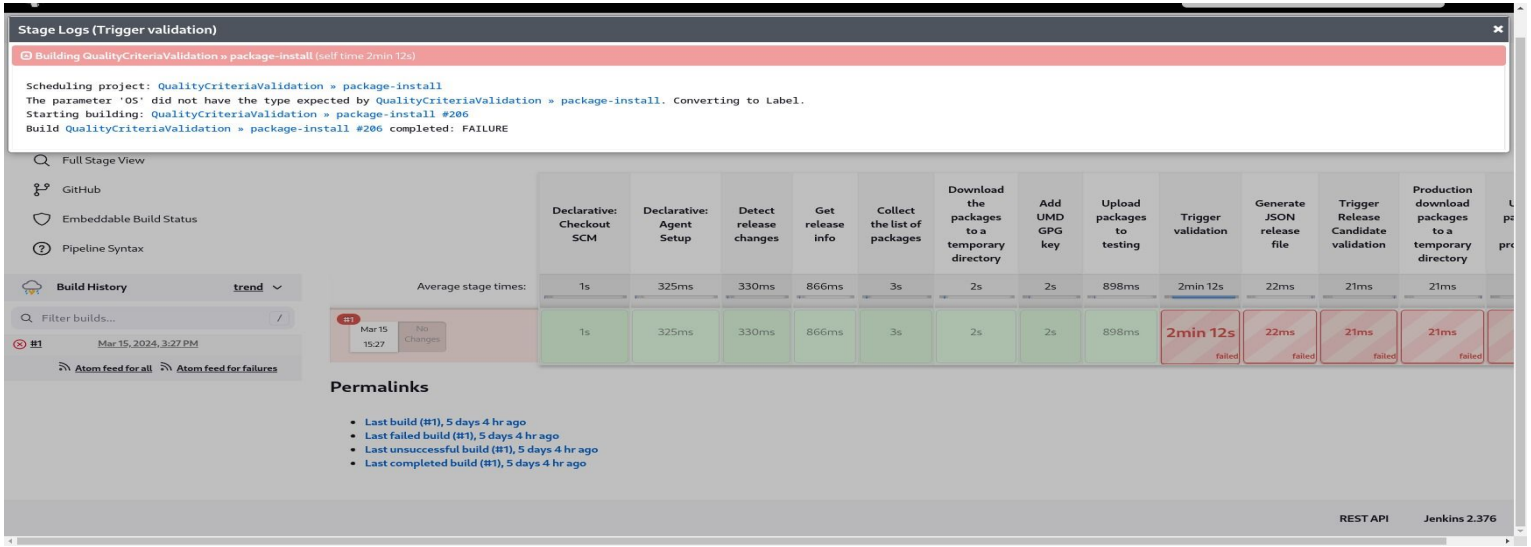
https://github.com/egi-qc/software-releases



Approved the Release (Testing/Production repository)

UMD/CMD new Workflow

https://github.com/egi-qc/software-releases



Packages -> Testing

UMD/CMD new Workflow

https://github.com/egi-qc/software-releases


```
            "use_backend": "auto",
            "validate_certs": true
        }
    },
    "item": "apel-ssm-service-3.4.0-1.el7.noarch",
    "msg": "Error unpacking rpm package apel-ssm-3.4.0-1.el7.noarch\n",
    "rc": 1,
    "results": [
        "Loaded plugins: fastestmirror, ovl, priorities\nLoading mirror speeds from cached hostfile\n * base: ftp.csuc.cat\n * epel: cdn.centos.no\n * extras: ftp.csuc.cat\n * updates: ftp.csuc.cat\n5 packages exc
due to repository priority protections\nResolving Dependencies\n--> Running transaction check\n---> Package apel-ssm-service.noarch 0:3.4.0-1.el7 will be installed\n--> Processing Dependency: apel-ssm for package:
apel-ssm-service-3.4.0-1.el7.noarch\n--> Running transaction check\n---> Package apel-ssm.noarch 0:3.4.0-1.el7 will be installed\n--> Finished Dependency Resolution\n\nDependencies
```


Packages -> Testing

Pipeline
Workflow

UMD/CMD new Workflow

https://github.com/egi-qc/software-releases

	Declarative: Checkout SCM	Declarative: Agent Setup	Detect release changes	Get release info	Collect the list of packages	Download the packages to a temporary directory	Add UMD GPG key	Upload packages to testing	Trigger validation	Generate JSON release file	Trigger Release Candidate validation	Production download packages to a temporary directory	Upload packages to production
Average stage times: (Average full run time: ~8min 1s)	1s	353ms	338ms	1s	1s	0ms	0ms	0ms	0ms	356ms	12min 13s	967ms	1s
#3 Mar 15 16:08 — 1 commit	1s	387ms	337ms	876ms	628ms					356ms	15min 43s	1s	3s
#2 Mar 15 15:51 — 1 commit	1s	328ms	338ms	1s	3s					356ms	8min 42s failed	23ms failed	74ms failed
#1 Mar 15 15:46 — No Changes	1s	344ms	340ms										

Testing -> Production

Yum repositories (new structure)

UMD-5 only

1. `release`: Main repository with all the packages (**= to production**).
2. `testing` : testing repository with rpm not fully validated in production environment. RPM belonging to release declared as production ready will be moved from testing to release.
3. `contrib`: repository for specific communities or special services. (**NEW**)

Dropped repositories: Staged Rollout

UMD production readiness

UMD-4 (first release plan) : Second week of April

UMD-5 (first release plan) : Second half of May

- List of products for al9 (provisory):
  - BDII and site-bdii
  - Gfal2
  - Dcache
  - Globus (evaluation)
  - HTCondor-CE

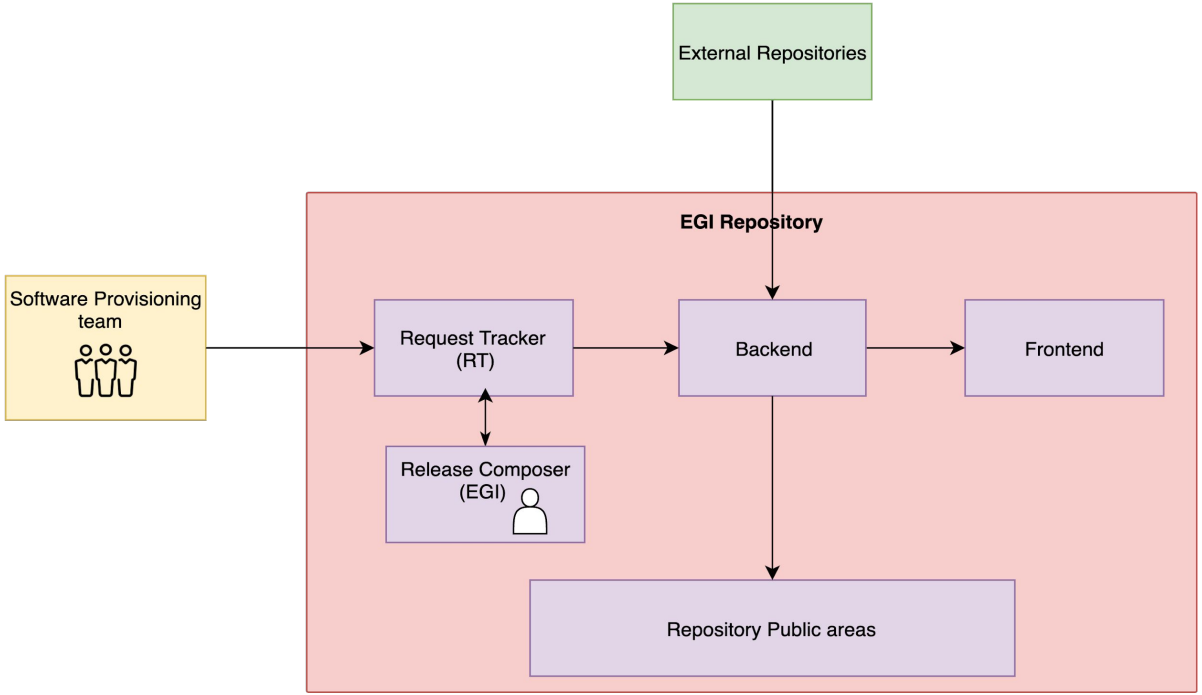
End

Questions?

UMD-4 architecture

Old setup:

- No new functionalities for many years, since it's hard to implement them.
- Backend based on perl scripts with tight relation
  with RT.
- To many manual steps (RT fully manual) leading to a large delay in the releases of packages
- Frontend outdated which could lead to security issues (**already replaced by a simplified version**)



Pipeline

https://github.com/egi-qc/software-releases

The pipeline is as follows:

1. `json_parser.py`: parse json, get the list of files to download and produce list of filenames (packages).
2. `download_pkgs.py` (option 0): download the packages to a temporary directory.
3. `rpm_sign.sh` (option 0): rpm sign each package.
4. `rpm_sign.sh` (option 0): verify signature of each package.
    a. a. verification of the packages
5. `upload_pkgs.py`: upload each package to nexusrepo.
6. `Package-install`: Validate package installations from testing repository and perform functional tests.
7. `release-candidate`: Install all packages in release repo together with the new packages from testing
8. `json_parser.sh`: Produce new json file as asset of the new release
9. `clean.sh`: clean temporary directories.