RELEASE AUTOMATION WITH LABVIEW AND GITLAB CI/CD

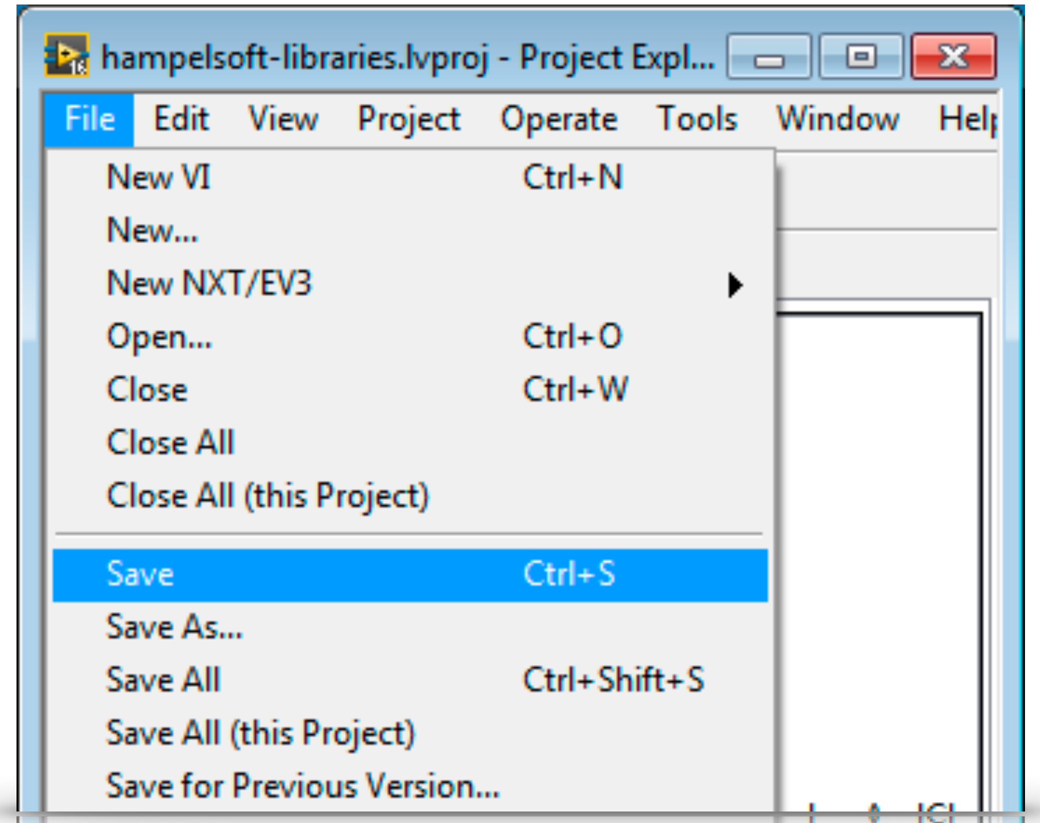
Joerg Hampel
HAMPEL SOFTWARE ENGINEERING

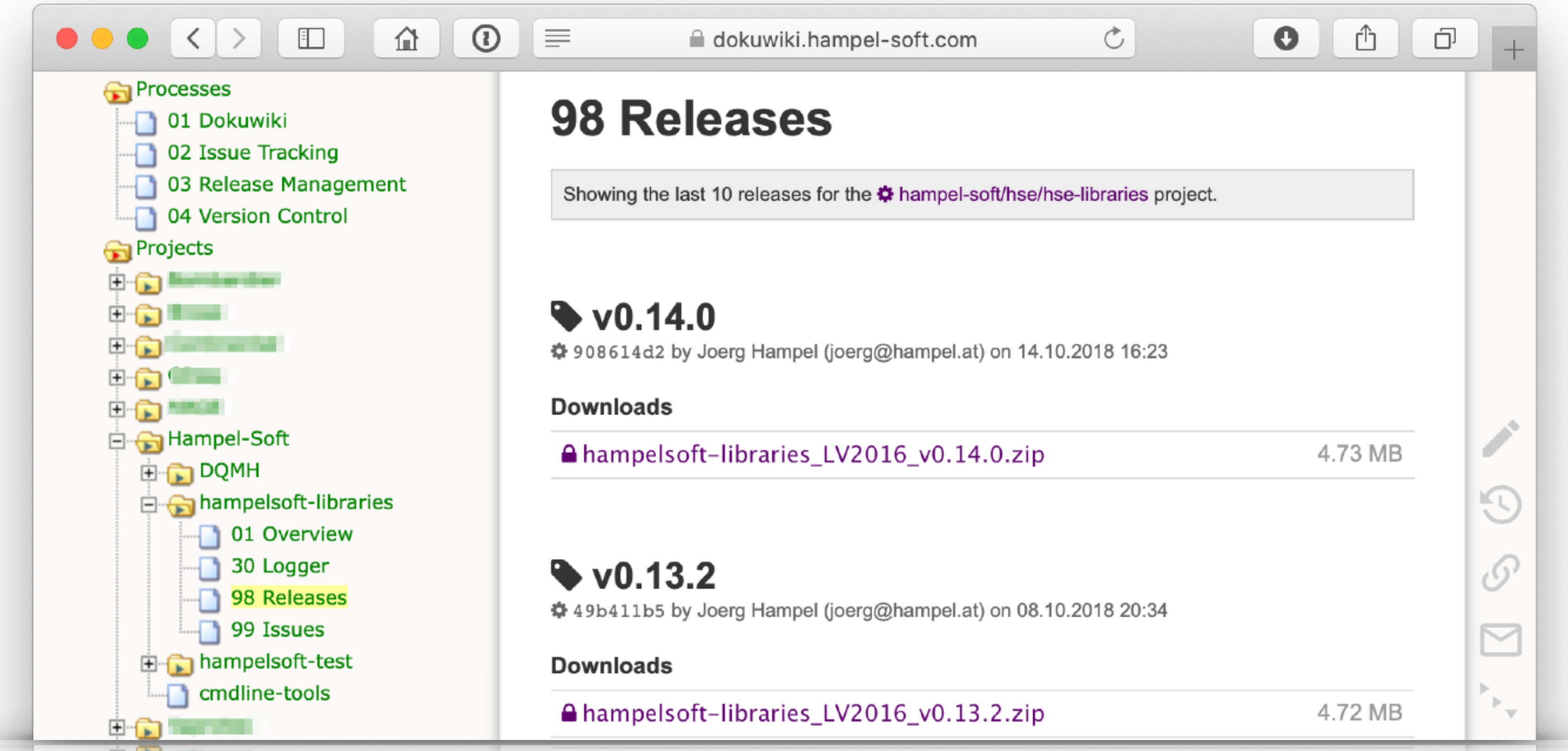
2019-04-07

WE’RE HIRING!

https://www.hampel-soft.com/careers/

„How many clicks does it take you from saving your VI to getting your built application to your customer?“





98 Releases

Showing the last 10 releases for the ⚙ hampel-soft/hse/hse-libraries project.

v0.14.0

⚙ 908614d2 by Joerg Hampel (joerg@hampel.at) on 14.10.2018 16:23

Downloads

🔒 hampelsoft-libraries_LV2016_v0.14.0.zip	4.73 MB

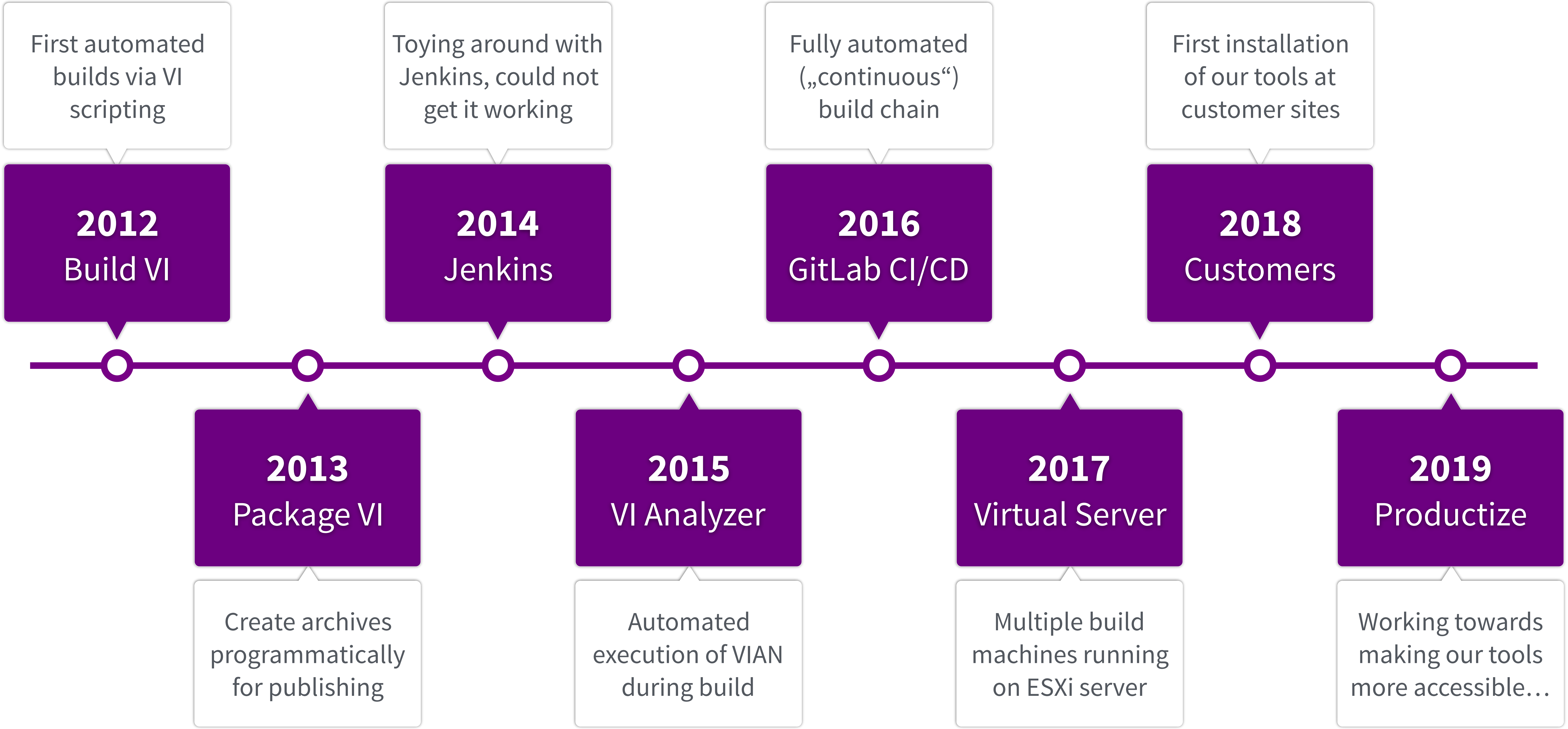
v0.13.2

⚙ 49b411b5 by Joerg Hampel (joerg@hampel.at) on 08.10.2018 20:34

Downloads

🔒 hampelsoft-libraries_LV2016_v0.13.2.zip	4.72 MB

THE HISTORY OF OUR TOOLS



2012
Build VI

First automated builds via VI scripting

2013
Package VI

Create archives programmatically for publishing

2014
Jenkins

Toying around with Jenkins, could not get it working

2015
VI Analyzer

Automated execution of VIAN during build

2016
GitLab CI/CD

Fully automated („continuous“) build chain

2017
Virtual Server

Multiple build machines running on ESXi server

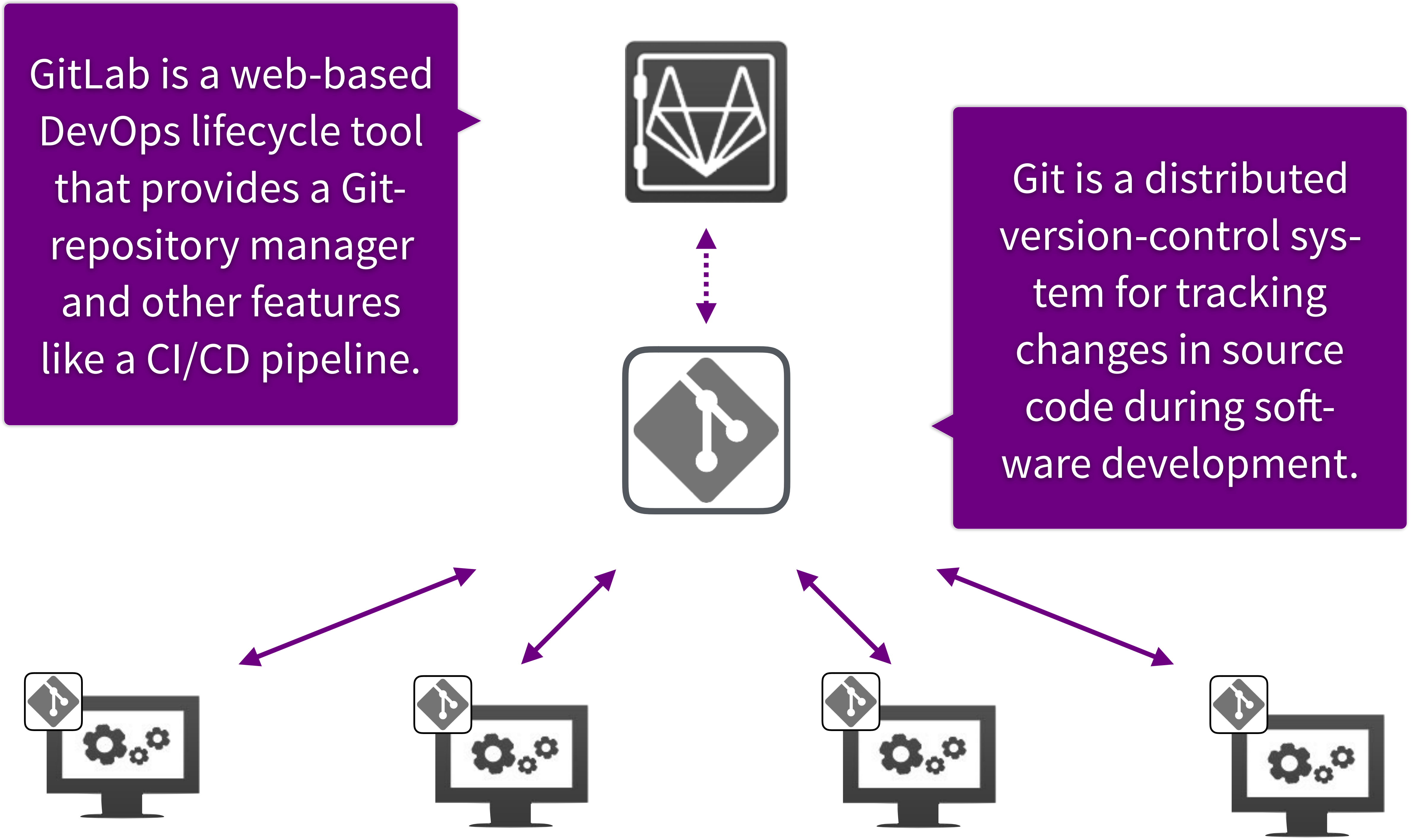
2018
Customers

First installation of our tools at customer sites

2019
Productize

Working towards making our tools more accessible…

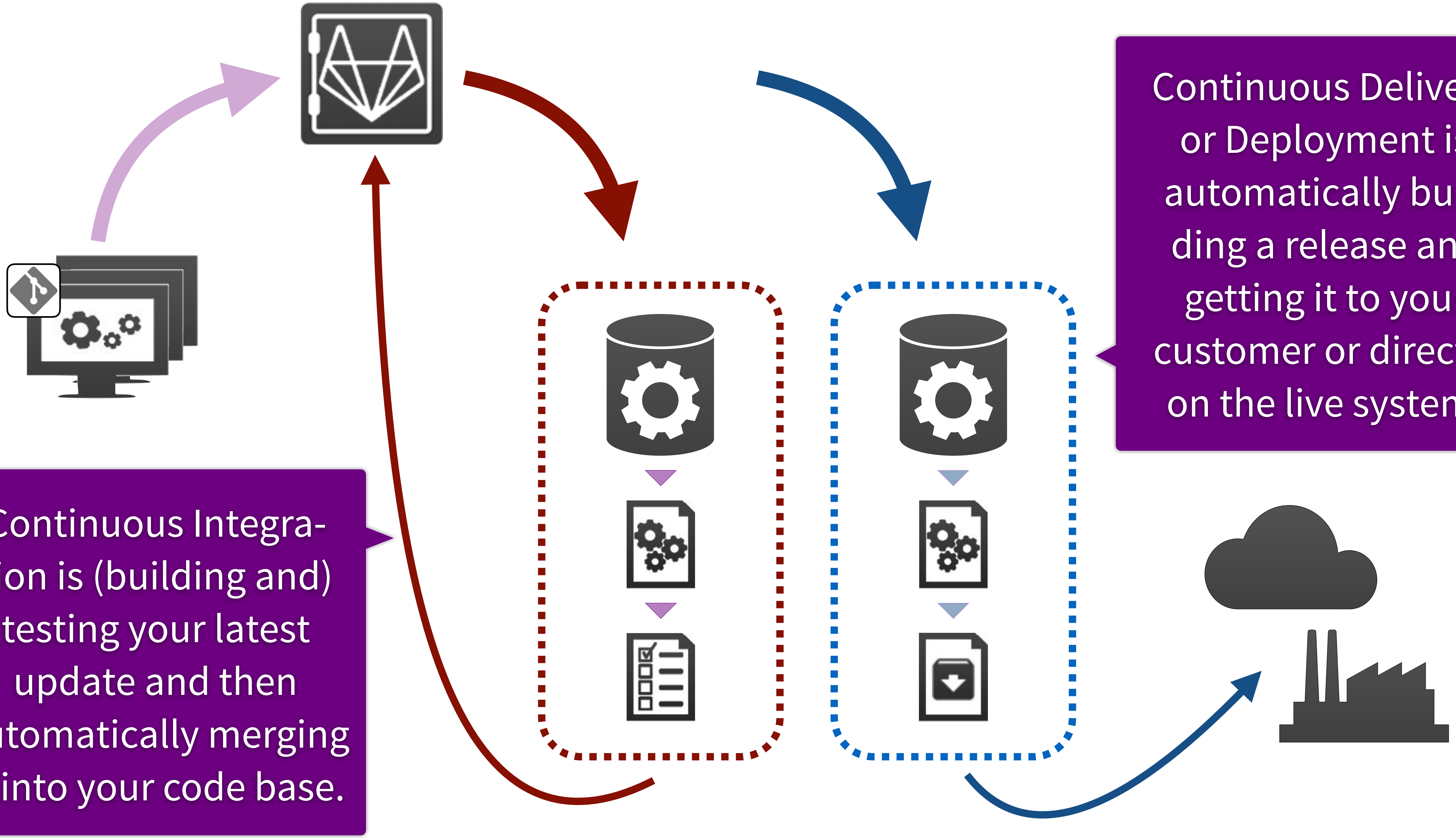
GIT & GITLAB



GitLab is a web-based DevOps lifecycle tool that provides a Git-repository manager and other features like a CI/CD pipeline.

Git is a distributed version-control system for tracking changes in source code during software development.

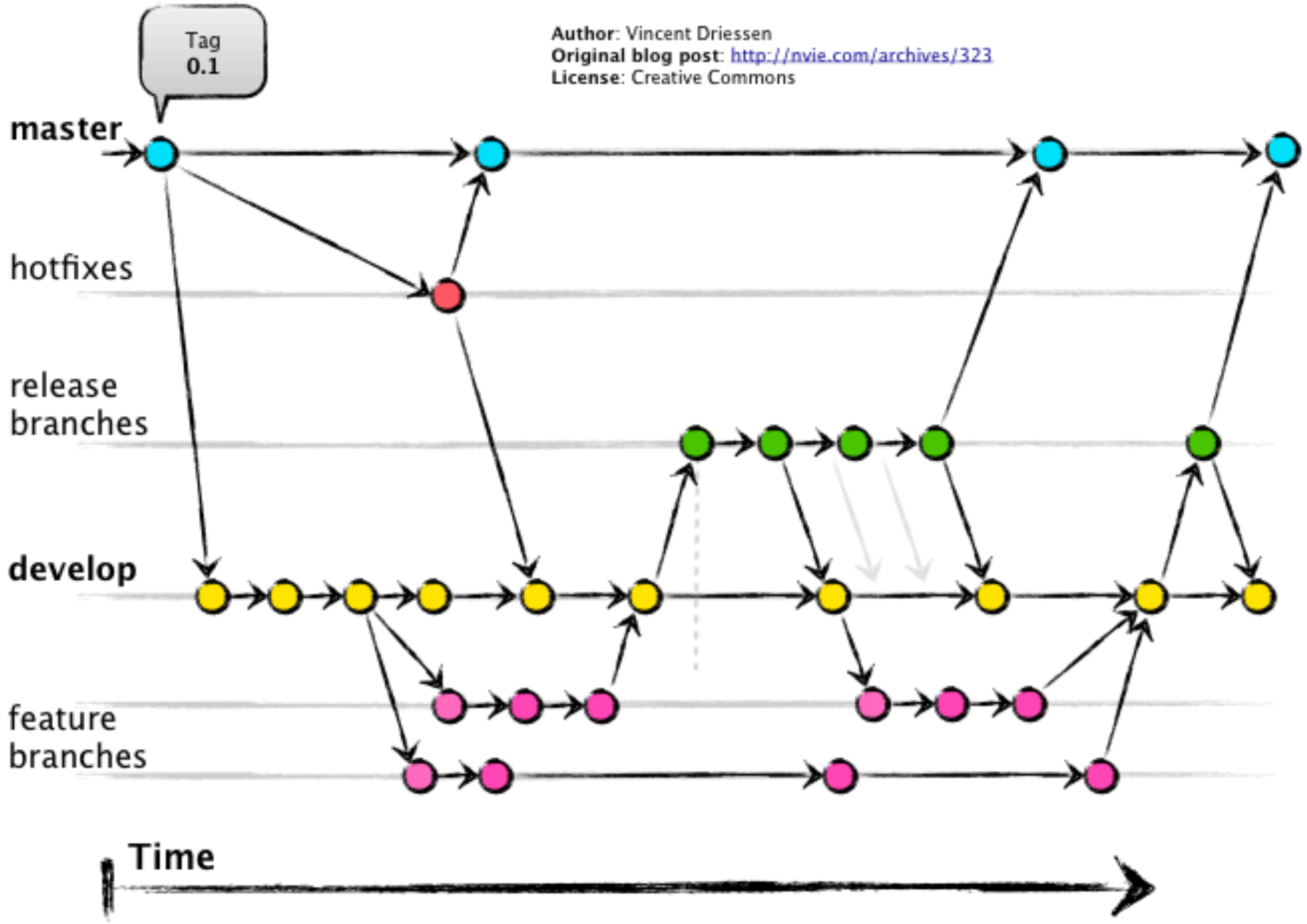
CI/CD = CONTINUOUS INTEGRATION & DELIVERY/DEPLOYMENT



Continuous Delivery or Deployment is automatically building a release and getting it to your customer or directly on the live system.

Continuous Integration is (building and) testing your latest update and then automatically merging it into your code base.

GITFLOW



Author: Vincent Driessen
Original blog post: http://nvie.com/archives/323
License: Creative Commons

GitFlow is a branching model for Git, created by Vincent Driessen.

It forms an elegant mental model that is easy to comprehend and allows team members to develop a shared understanding of the branching and releasing processes.

THE PARTS

Repository

Source code is hosted in git repositories at gitlab.com. Each project got its own repository. Information can be queried via API.

https://about.gitlab.com/gitlab-ci/

Build Mech.

Build specifications are executed with the Application Builder, either manually on the development PC or programmatically on another machine (the „build server“).

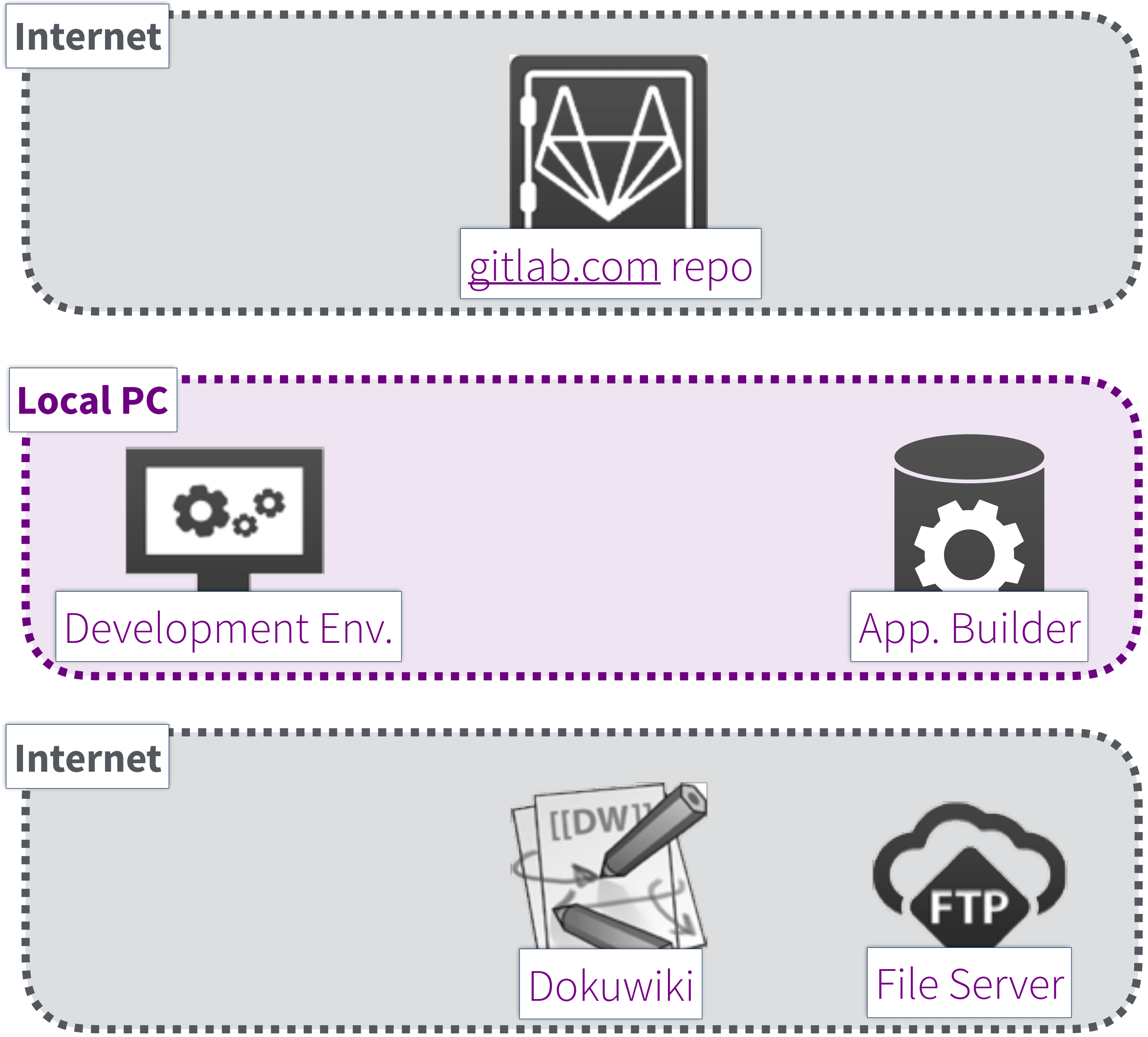
Web Platform

Dokuwiki is used for distribution. Its simple-to-use web interface and built-in authentication make it easy to set up namespaces for project downloads.

https://www.dokuwiki.com/

THE MANUAL WAY

INFRASTRUCTURE

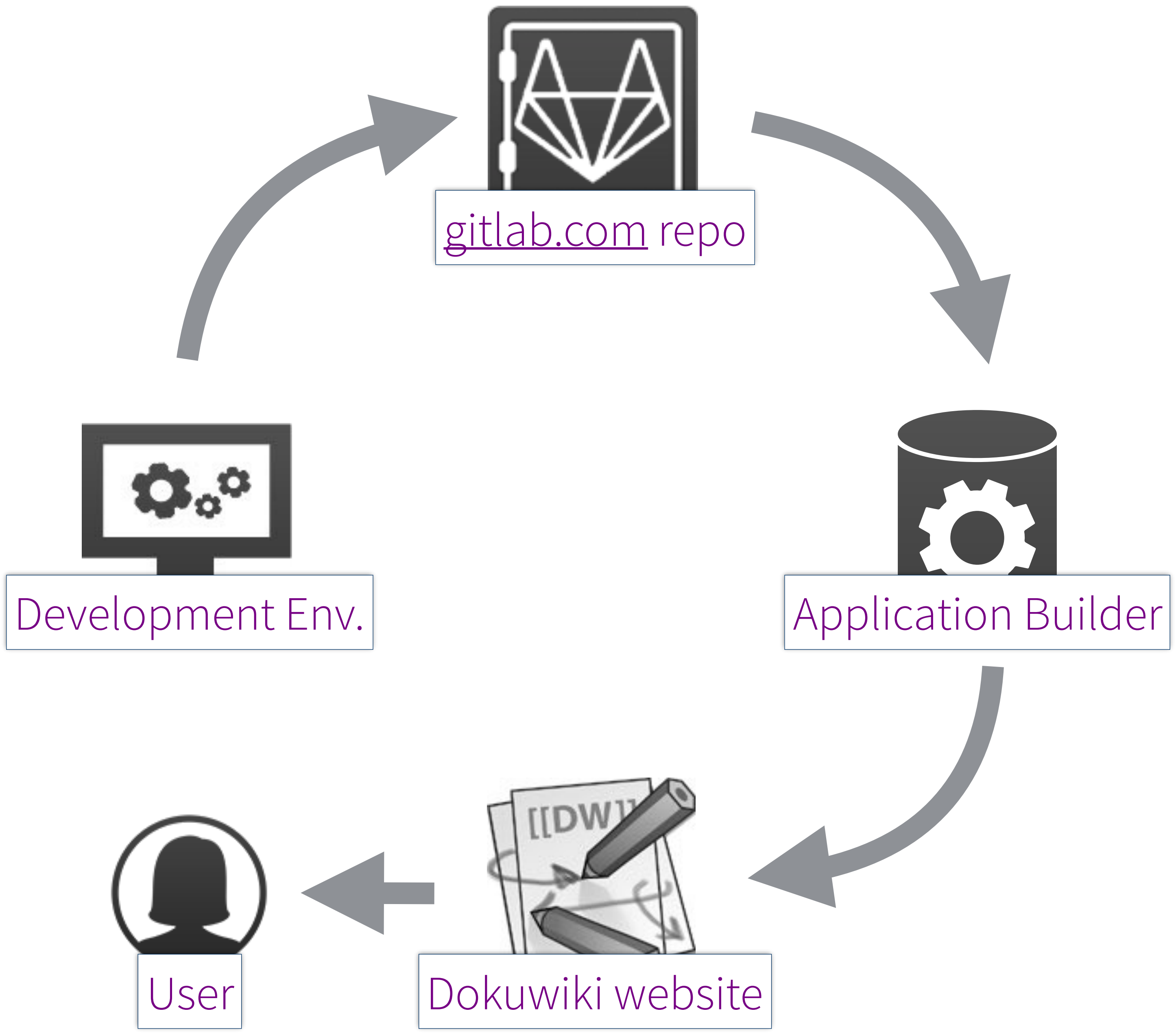


Git repositories are hosted at gitlab.com

Development Env. and App. Builder running locally (PC)

Dokuwiki and files are hosted at a web provider

A LOT OF CLICKS...



gitlab.com repo

Application Builder

Dokuwiki website

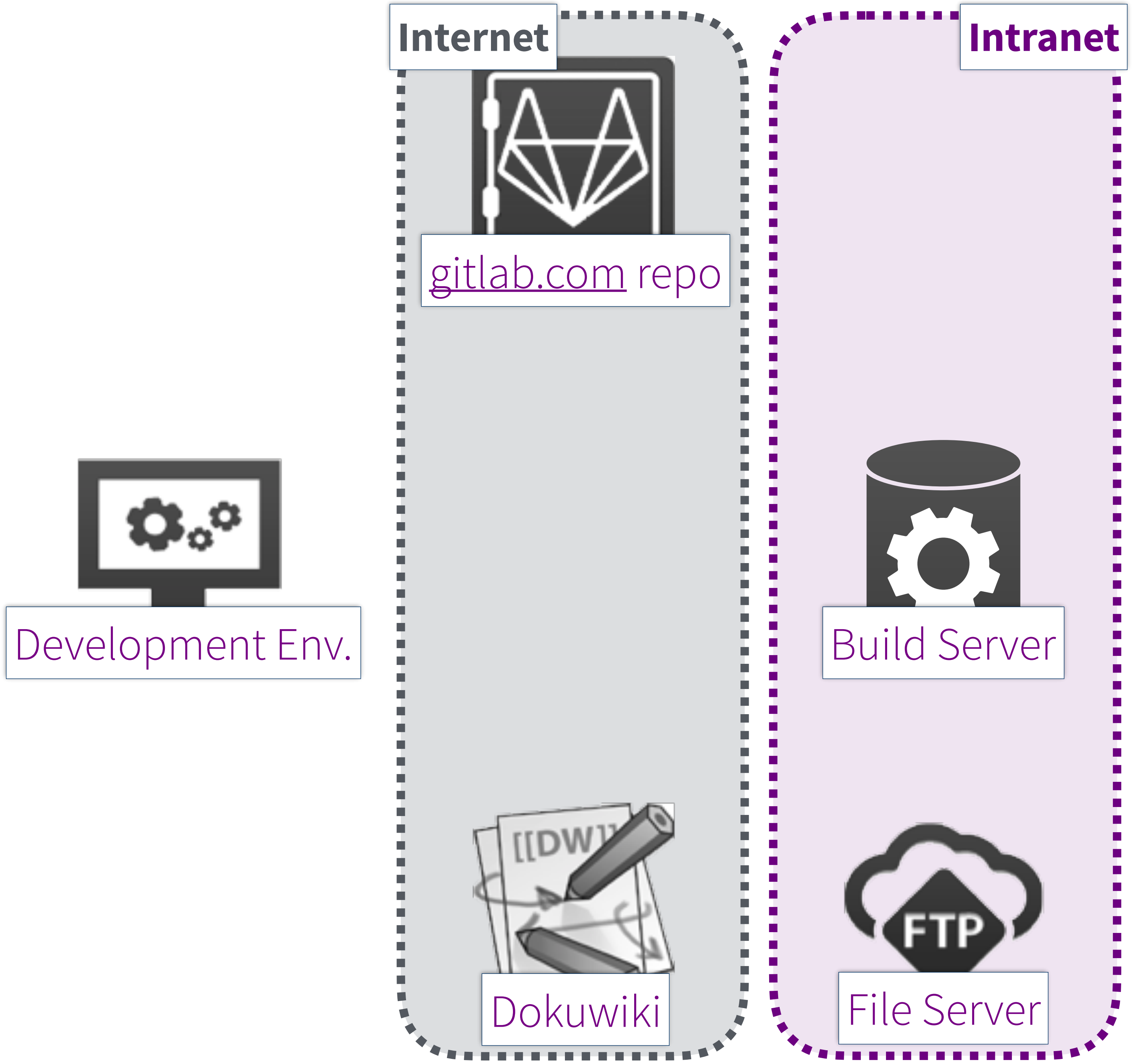
User

Development Env.

- Click „Save“
- Commit/Push
- Run VI Analyzer
- Execute build spec
- Package results
- Upload results
- Create weblink

AUTOMATION OF PROCESSES

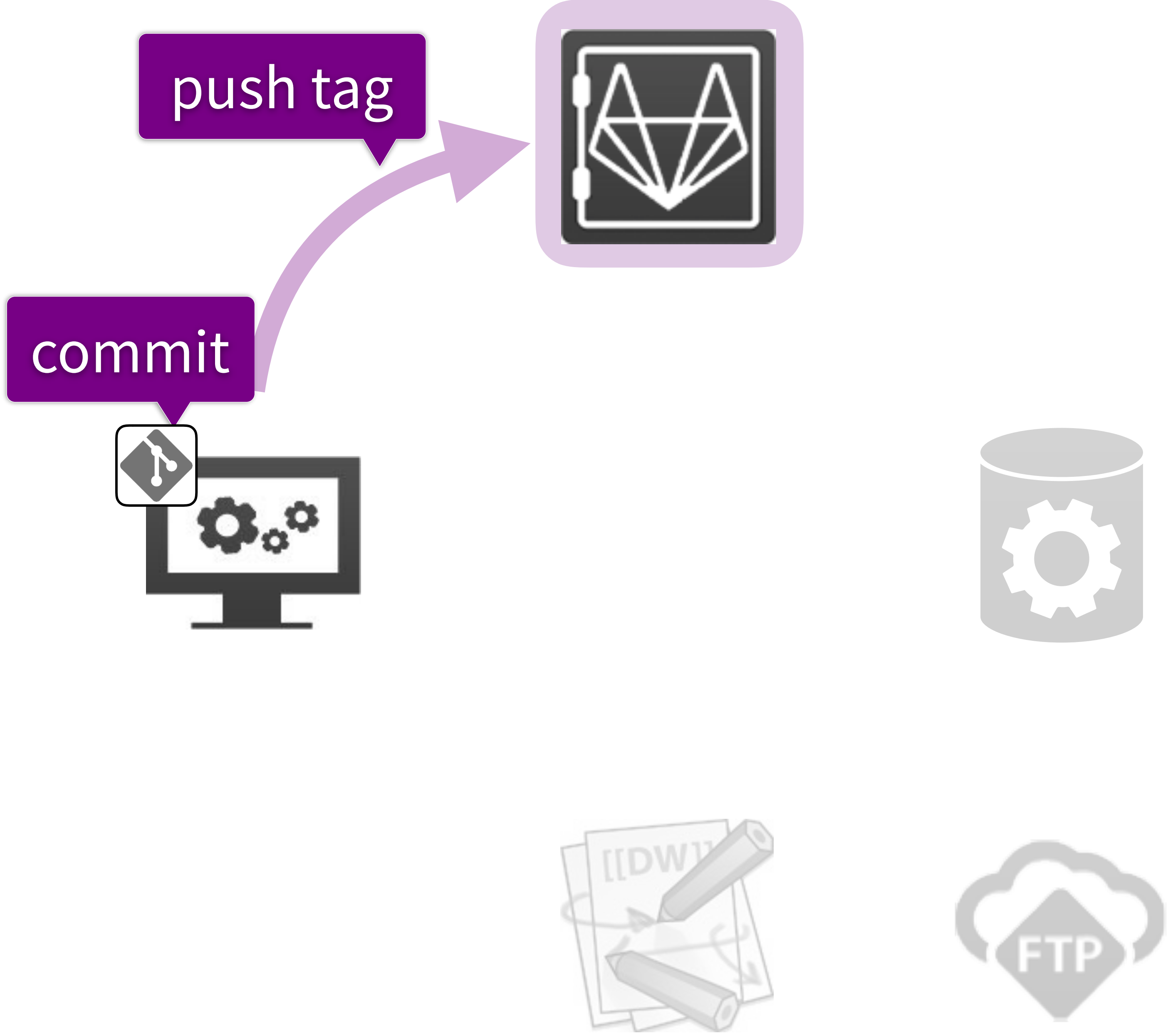
INFRASTRUCTURE



The build server can be any PC running LabVIEW. The file server can be any device reachable via HTTP or FTP.

In this example, the build server is a virtual machine running on a vSphere server and the file server is a NAS.

SEND CHANGES TO SOURCE CODE CONTROL



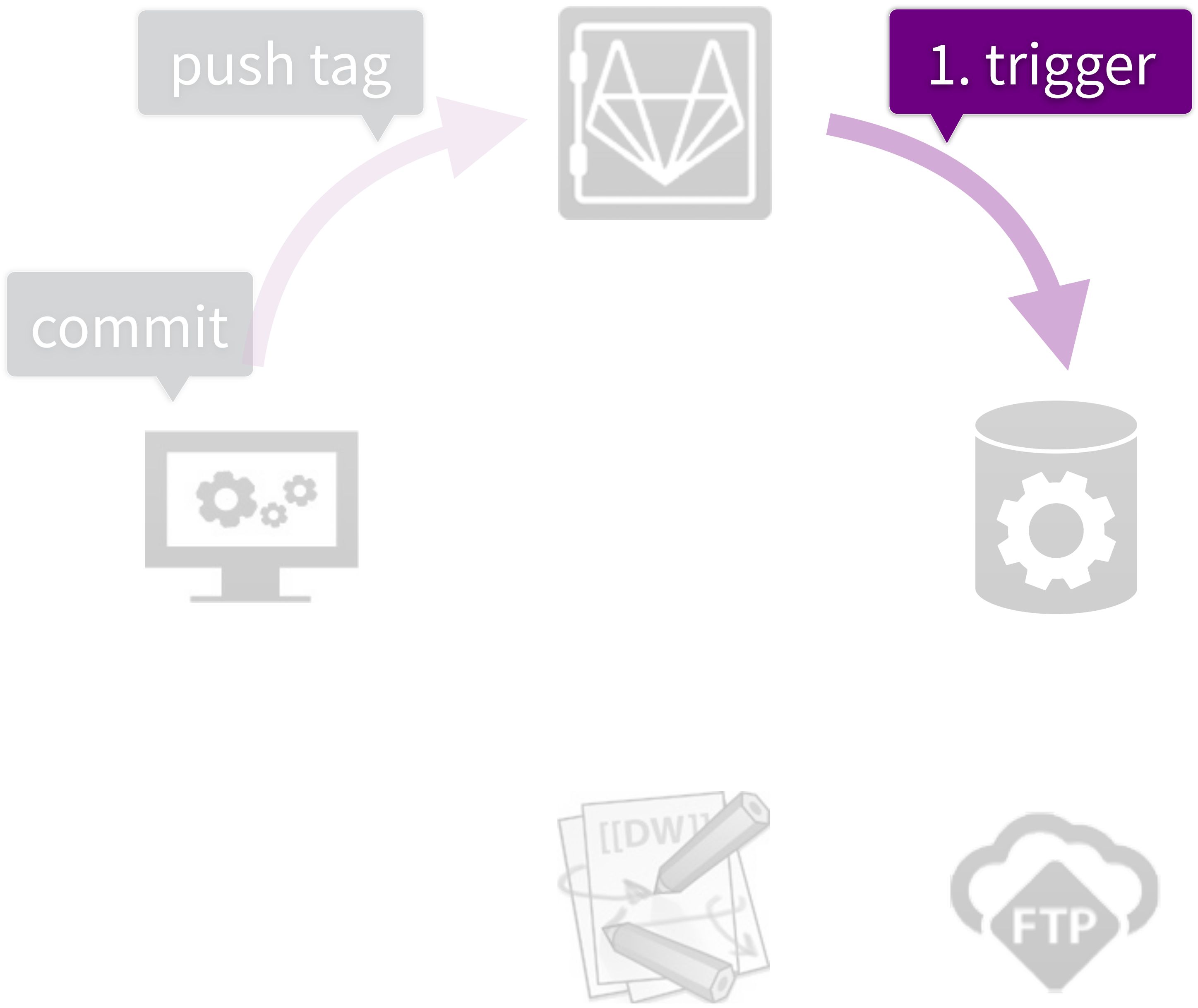
Whenever a developer saves and pushes a tag, the release mechanism is triggered.

The gitlab-ci.yml config file is part of the repo and defines which actions to take on which event („when to call which scripts“).

Video: Source Code Control with Git & Gitflow

https://youtu.be/ue0NeYNEYf0

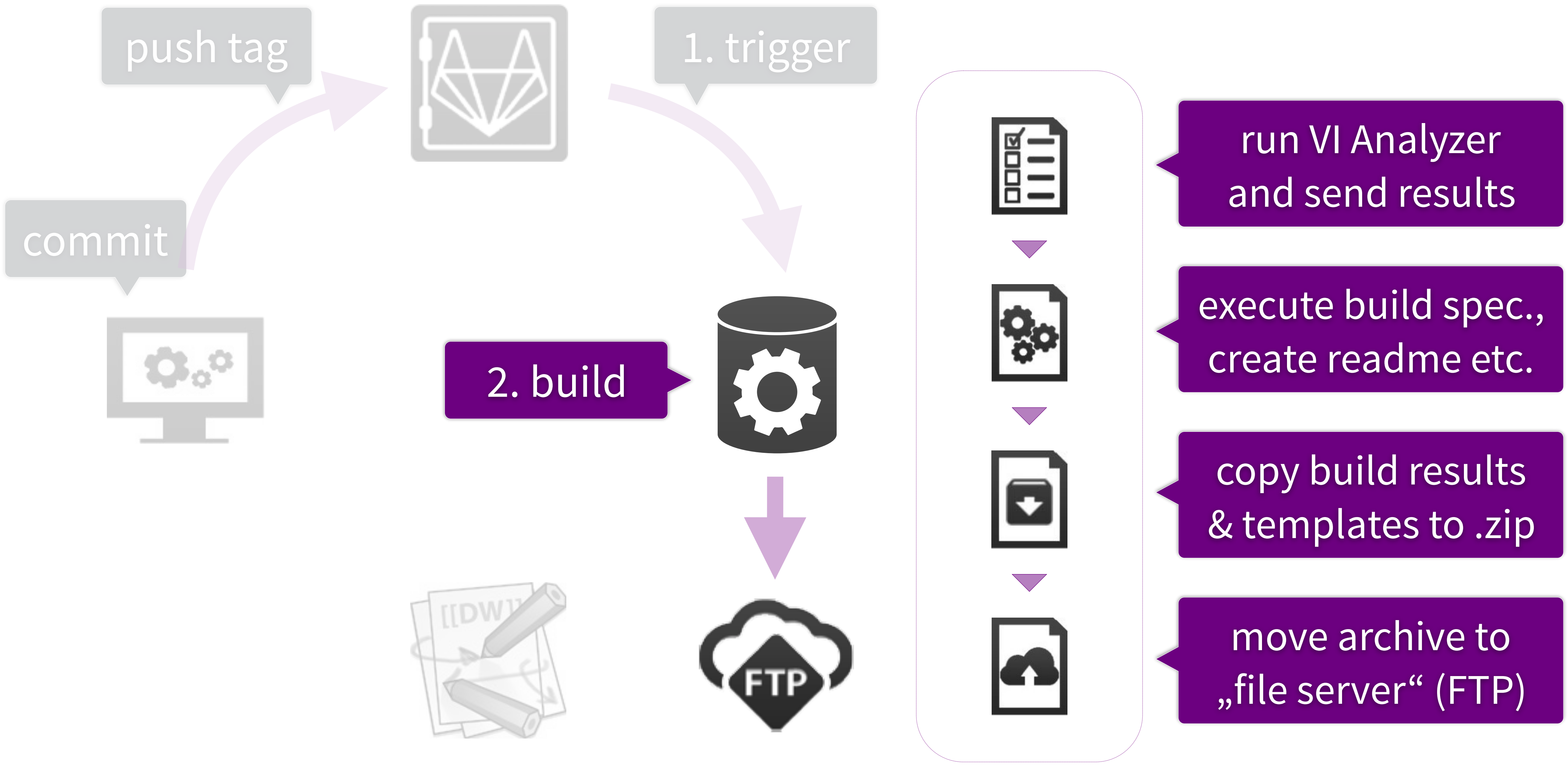
SOURCE CODE CONTROL TRIGGERS BUILD SERVER



Gitlab's built-in CI allows to trigger „runners“ based on your git actions (eg pushing a tag).

The runner is installed on the build server and executes defined actions (scripts).

BUILD SERVER EXECUTES LABVIEW SCRIPTS



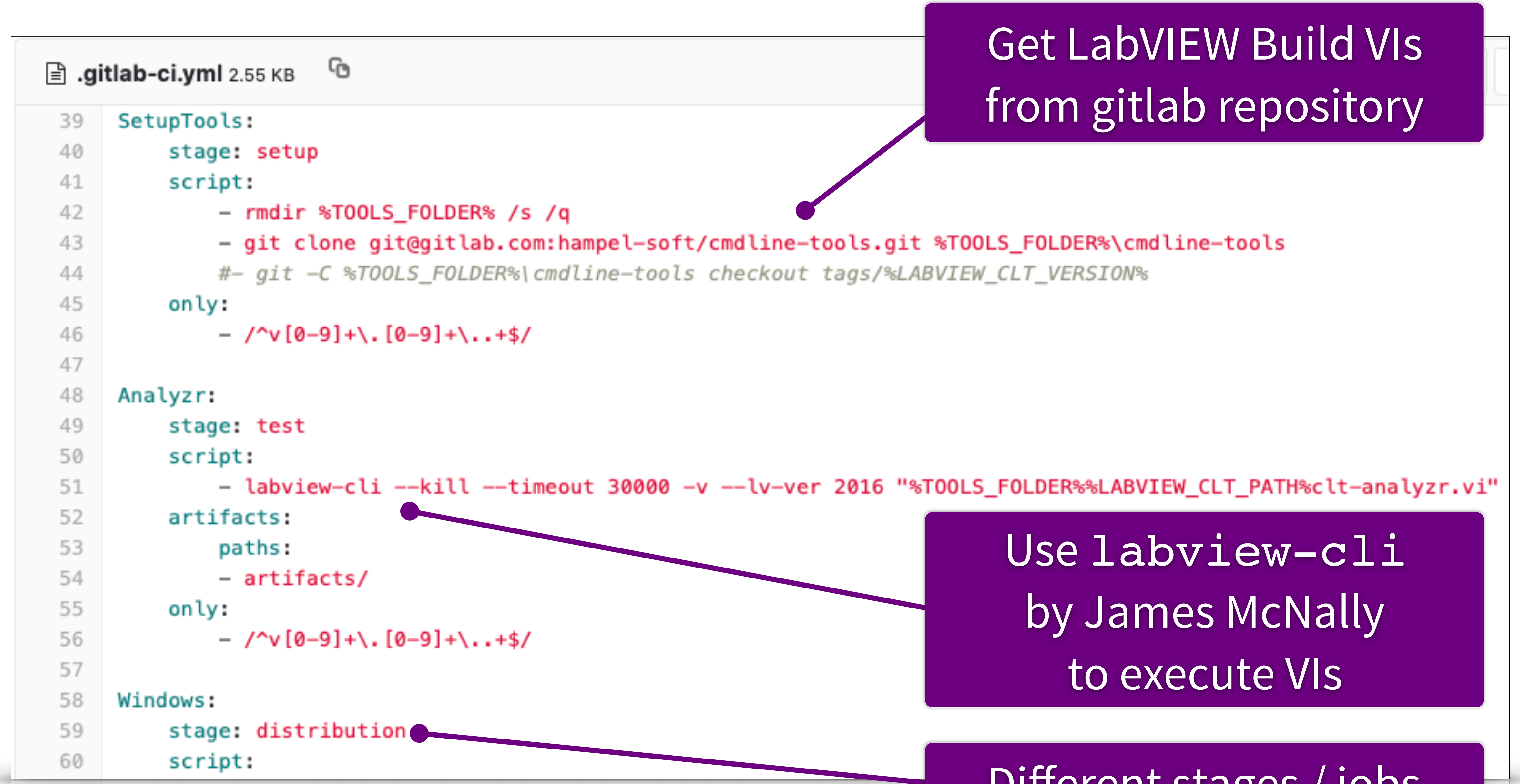
run VI Analyzer and send results

execute build spec., create readme etc.

copy build results & templates to .zip

move archive to „file server“ (FTP)

GITLAB CI CONFIGURATION FILE



.gitlab-ci.yml 2.55 KB

```
SetupTools:
    stage: setup
    script:
        - rmdir %TOOLS_FOLDER% /s /q
        - git clone git@gitlab.com:hampel-soft/cmdline-tools.git %TOOLS_FOLDER%\cmdline-tools
        #- git -C %TOOLS_FOLDER%\cmdline-tools checkout tags/%LABVIEW_CLT_VERSION%
    only:
        - /^v[0-9]+\.[0-9]+\..+$/

Analyzr:
    stage: test
    script:
        - labview-cli --kill --timeout 30000 -v --lv-ver 2016 "%TOOLS_FOLDER%%LABVIEW_CLT_PATH%clt-analyzr.vi"
    artifacts:
        paths:
        - artifacts/
    only:
        - /^v[0-9]+\.[0-9]+\..+$/

Windows:
    stage: distribution
    script:
```

Get LabVIEW Build VIs from gitlab repository

Use `labview-cli` by James McNally to execute VIs

Different stages / jobs allow for flexibility

Video: James McNally about LabVIEW-CLI

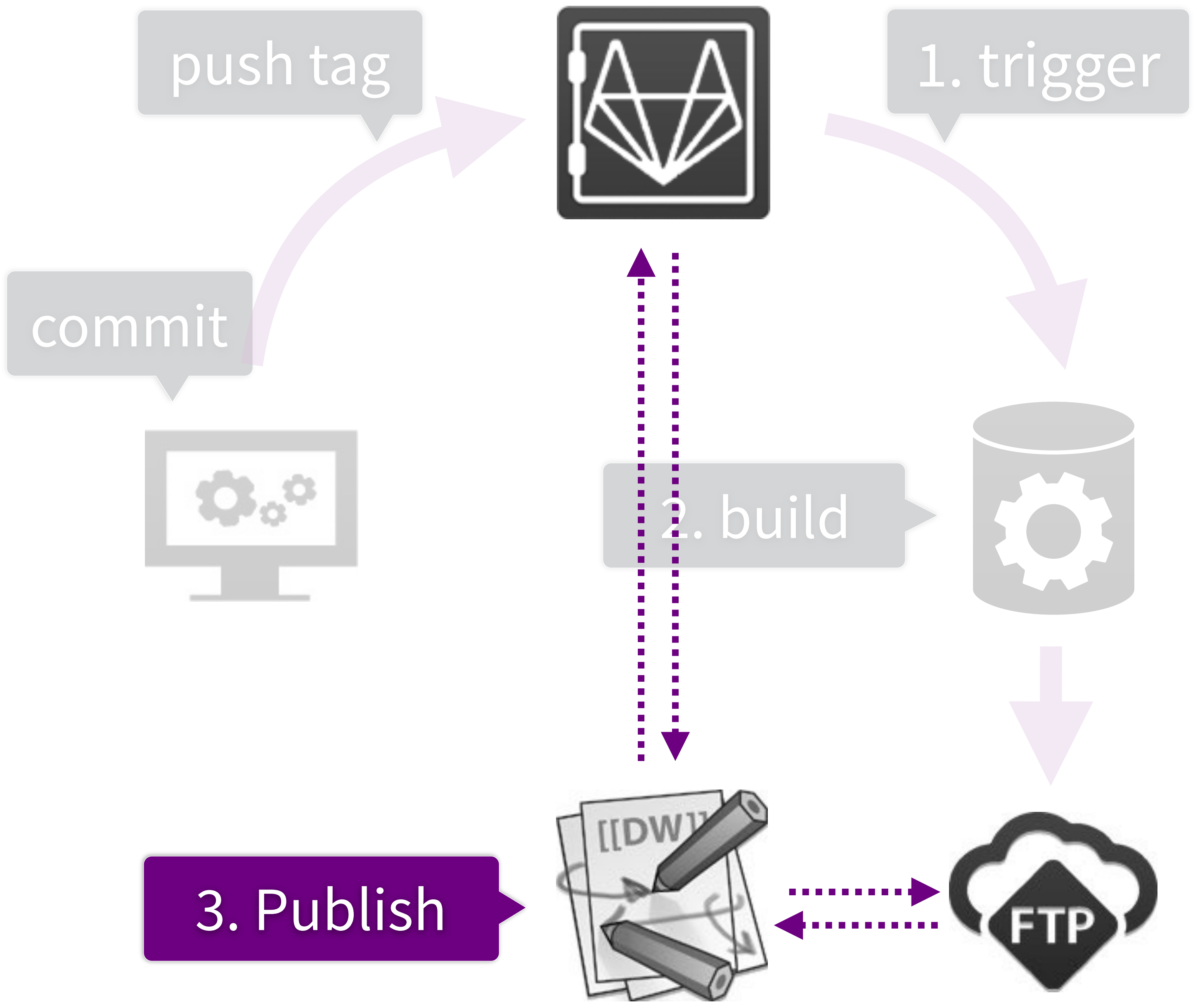
https://youtu.be/WV2GNcfgHMo

Video: GitLab CI Web Interface and Runner Output

https://youtu.be/1MGP5X-LmYE

Video: GitLab Runner triggering scripts on build server
https://youtu.be/FTYpQNPgwAQ

BUILT FILES ARE MOVED TO NETWORKED LOCATION

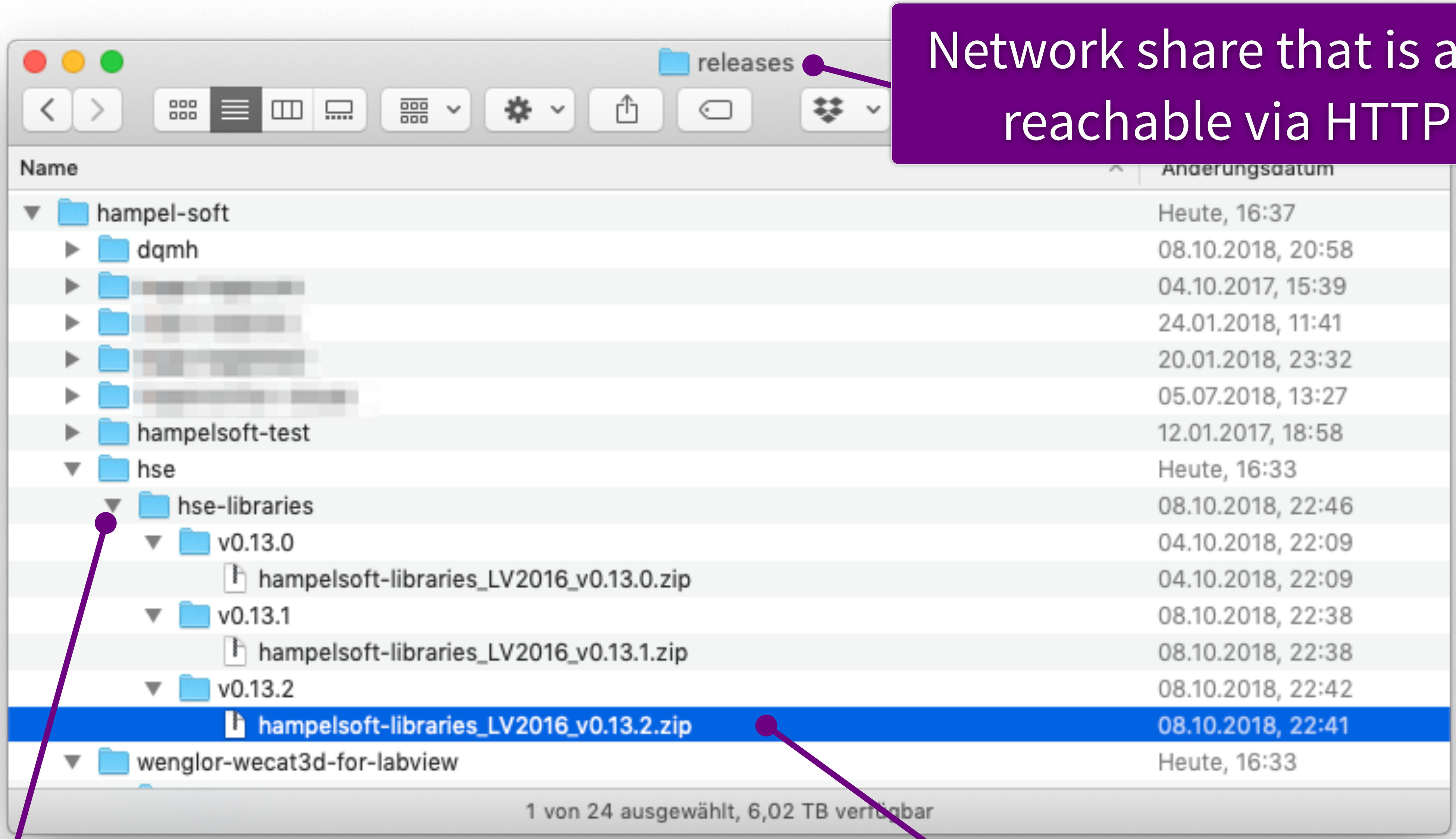


Dokuwiki generates the download links for each release automatically:

A plugin queries the given gitlab repo for tags, and the file server for files related to each tag (relation defined by file structure).

gitlab.com/hampel-soft/hse/dokuwiki-gitlabapi/

FILE SERVER STRUCTURE



releases

Name	Änderungsdatum
▼ hampel-soft	Heute, 16:37
▶ dqmh	08.10.2018, 20:58
▶ [redacted]	04.10.2017, 15:39
▶ [redacted]	24.01.2018, 11:41
▶ [redacted]	20.01.2018, 23:32
▶ [redacted]	05.07.2018, 13:27
▶ hampelsoft-test	12.01.2017, 18:58
▼ hse	Heute, 16:33
▼ hse-libraries	08.10.2018, 22:46
▼ v0.13.0	04.10.2018, 22:09
hampelsoft-libraries_LV2016_v0.13.0.zip	04.10.2018, 22:09
▼ v0.13.1	08.10.2018, 22:38
hampelsoft-libraries_LV2016_v0.13.1.zip	08.10.2018, 22:38
▼ v0.13.2	08.10.2018, 22:42
hampelsoft-libraries_LV2016_v0.13.2.zip	08.10.2018, 22:41
▼ wenglor-wecat3d-for-labview	Heute, 16:33

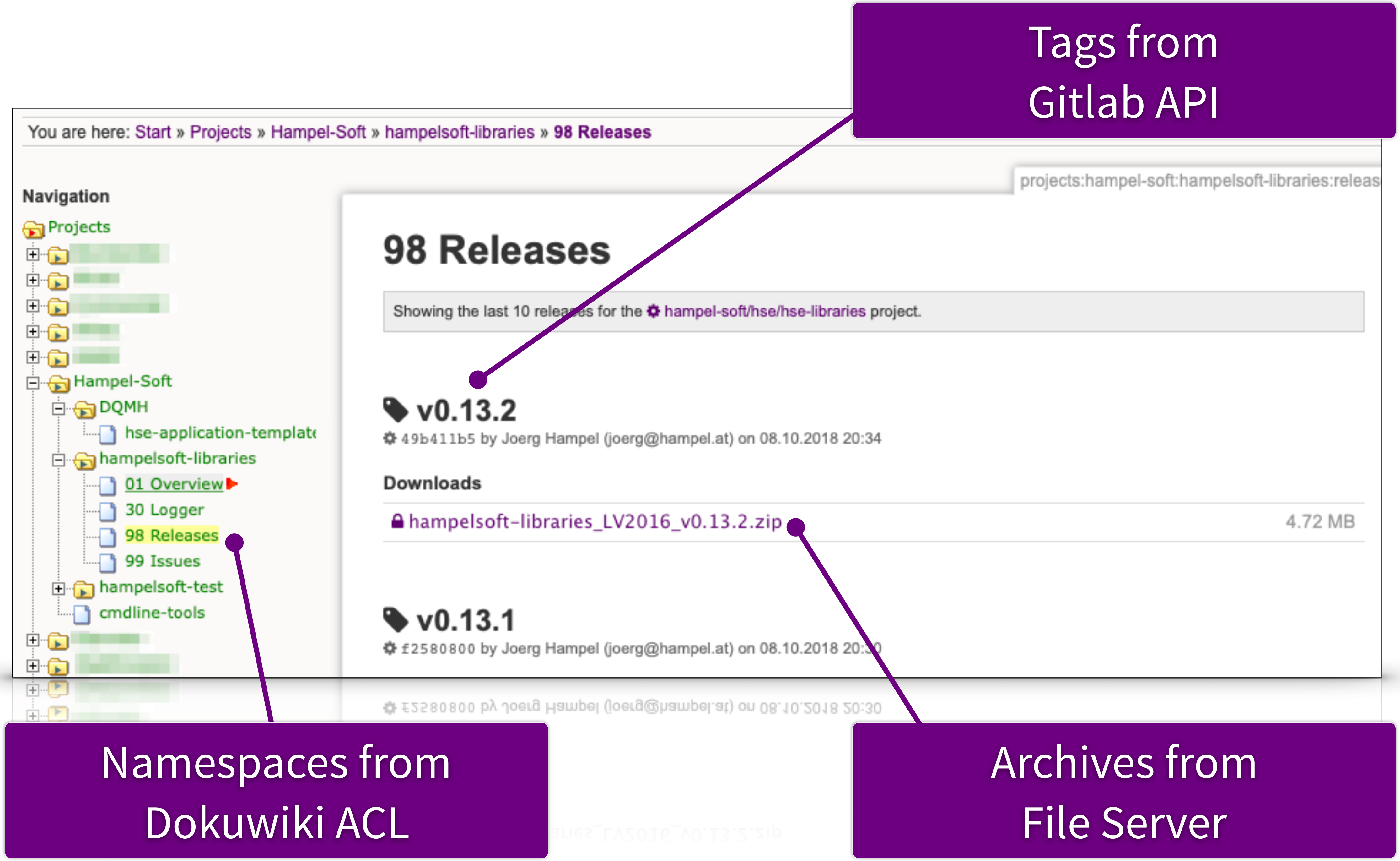
1 von 24 ausgewählt, 6,02 TB verfügbar

Network share that is also reachable via HTTP

File structure mimics namespace/repo/tags

Archives generated by the build server

DOKUWIKI SCREENSHOT



You are here: Start » Projects » Hampel-Soft » hampelsoft-libraries » **98 Releases**

projects:hampel-soft:hampelsoft-libraries:releas

Navigation

- Projects
  - Hampel-Soft
    - DQMH
      - hse-application-template
    - hampelsoft-libraries
      - 01 Overview
      - 30 Logger
      - 98 Releases
      - 99 Issues
    - hampelsoft-test
    - cmdline-tools

98 Releases

Showing the last 10 releases for the hampel-soft/hse/hse-libraries project.

v0.13.2

49b411b5 by Joerg Hampel (joerg@hampel.at) on 08.10.2018 20:34

Downloads

hampelsoft-libraries_LV2016_v0.13.2.zip	4.72 MB

v0.13.1

f2580800 by Joerg Hampel (joerg@hampel.at) on 08.10.2018 20:30

Tags from Gitlab API

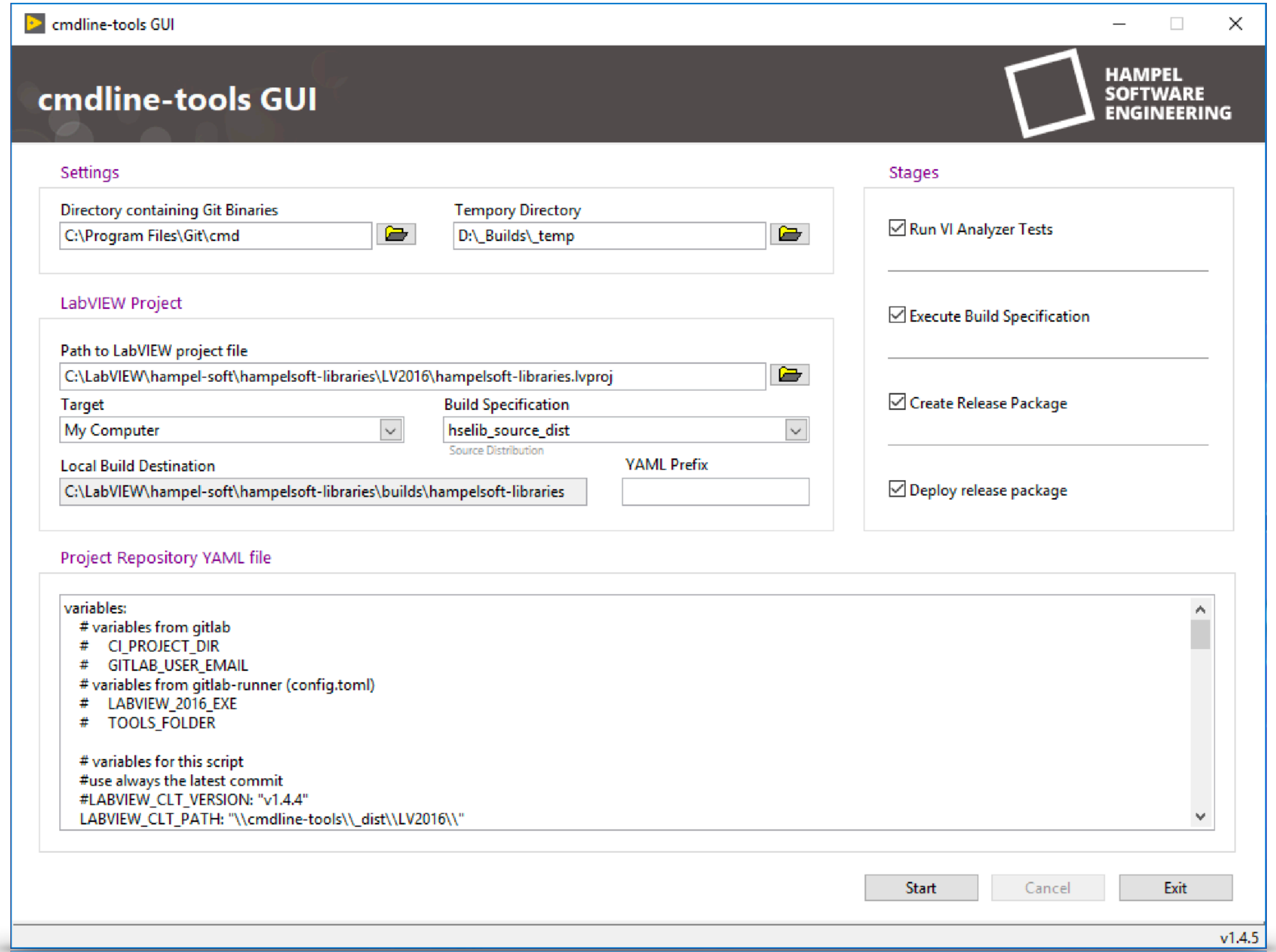
Namespaces from Dokuwiki ACL

Archives from File Server

Video: Release Management on Dokuwiki
https://youtu.be/POB25NRlxD4

AUTOMATION, MANUALLY…

STAND-ALONE APPLICATION WITH GUI



Settings

Directory containing Git Binaries: C:\Program Files\Git\cmd

Tempory Directory: D:_Builds_temp

LabVIEW Project

Path to LabVIEW project file: C:\LabVIEW\hampel-soft\hampelsoft-libraries\LV2016\hampelsoft-libraries.lvproj

Target: My Computer

Build Specification: hselib_source_dist (Source Distribution)

Local Build Destination: C:\LabVIEW\hampel-soft\hampelsoft-libraries\builds\hampelsoft-libraries

YAML Prefix:

Stages

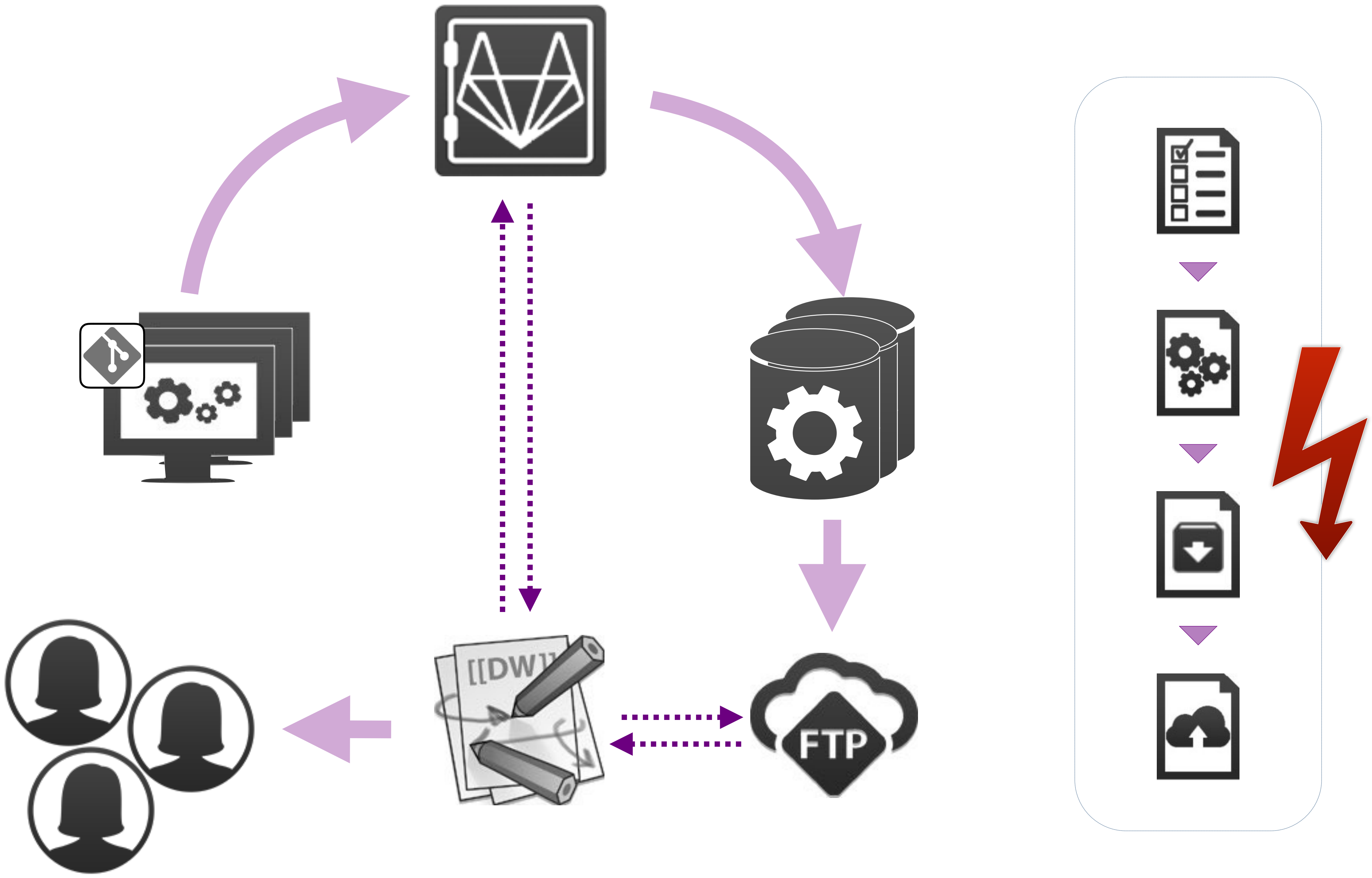
- ☑ Run VI Analyzer Tests
- ☑ Execute Build Specification
- ☑ Create Release Package
- ☑ Deploy release package

Project Repository YAML file

```
variables:
   # variables from gitlab
   #     CI_PROJECT_DIR
   #     GITLAB_USER_EMAIL
   # variables from gitlab-runner (config.toml)
   #     LABVIEW_2016_EXE
   #     TOOLS_FOLDER

   # variables for this script
   #use always the latest commit
   #LABVIEW_CLT_VERSION: "v1.4.4"
   LABVIEW_CLT_PATH: "\\cmdline-tools\\_dist\\LV2016\\"
```

WHERE WE STAND IN TERMS OF CI/CD



Our own build chain is currently lacking the ability to automatically test source code (i.e. unit testing).

As the nature of our customer projects asks for regression and integration testing, this does not hurt us as much as you would expected.

Testing is the next big step on our road map.

DON’T BELIEVE US, TRY FOR YOURSELF!

HSE	https://gitlab.com/hampel-soft/
CI/CD	https://about.gitlab.com/features/gitlab-ci-cd/
WIRESMITH Technology	https://www.wiresmithtech.com
GitHub	https://github.com/JamesMc86/LabVIEW-CLI/
DokuWiki	https://www.dokuwiki.org/dokuwiki
vmware vSphere	https://www.vmware.com/de/products/vsphere-hypervisor.html

IT'S OK TO HAVE FUN!

See you at GDevCon!

GDEVCON #2

Birmingham UK
21st-22nd August 2019