

# Embracing CI/CD workflows for building ETL pipelines

how we will gather and monitor multi-source spatially-interpolated meteorological parameters in near-real time

Elena Maines

11.11.23

# Who



ELENA MAINES  
Junior Researcher



PIERO CAMPALANI  
Software Engineer



ALICE CRESPI  
Senior Researcher

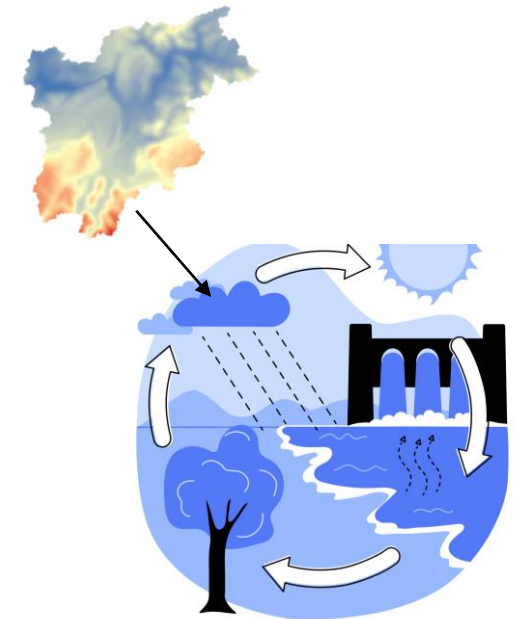
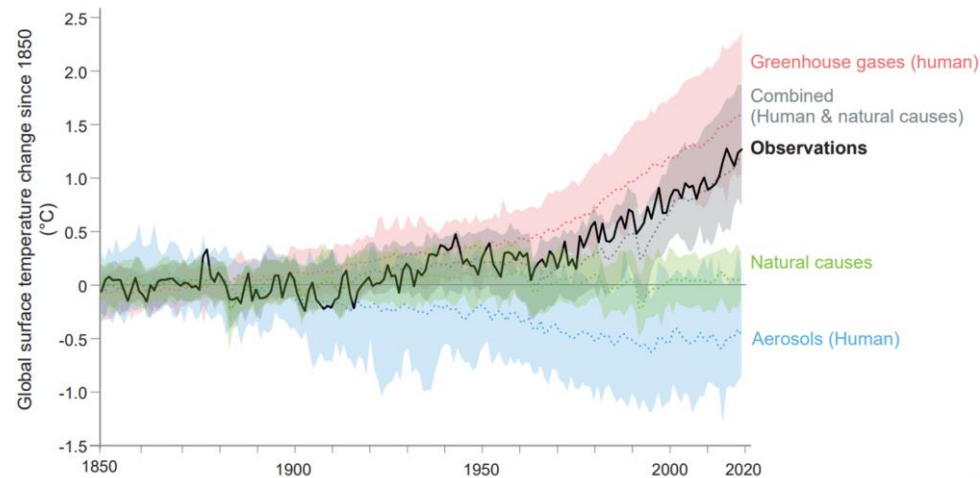
# Pipeline: motivation

Having reliable and up-to-date long-term meteorological data is crucial for

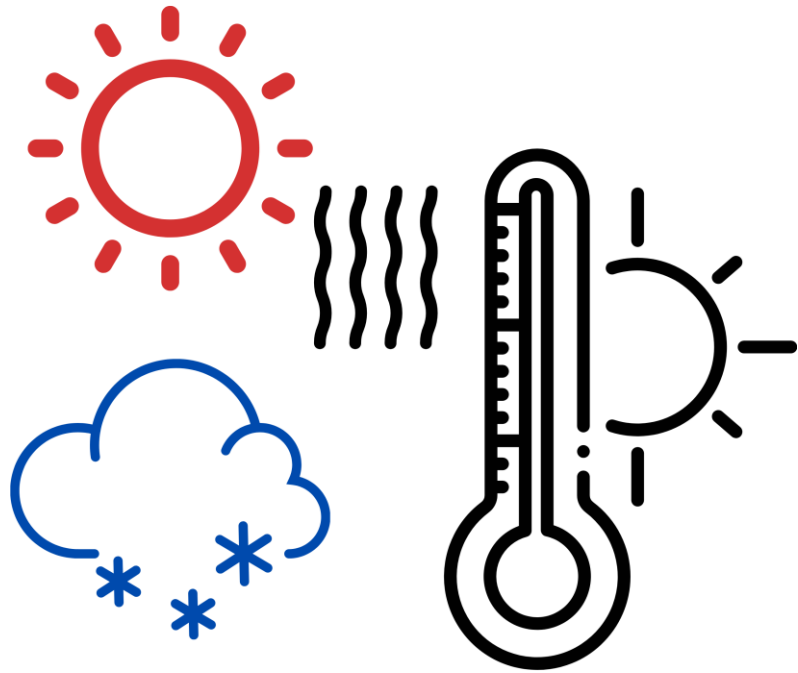
Monitoring current meteorological conditions at regional level

Monitoring climate variability at local level

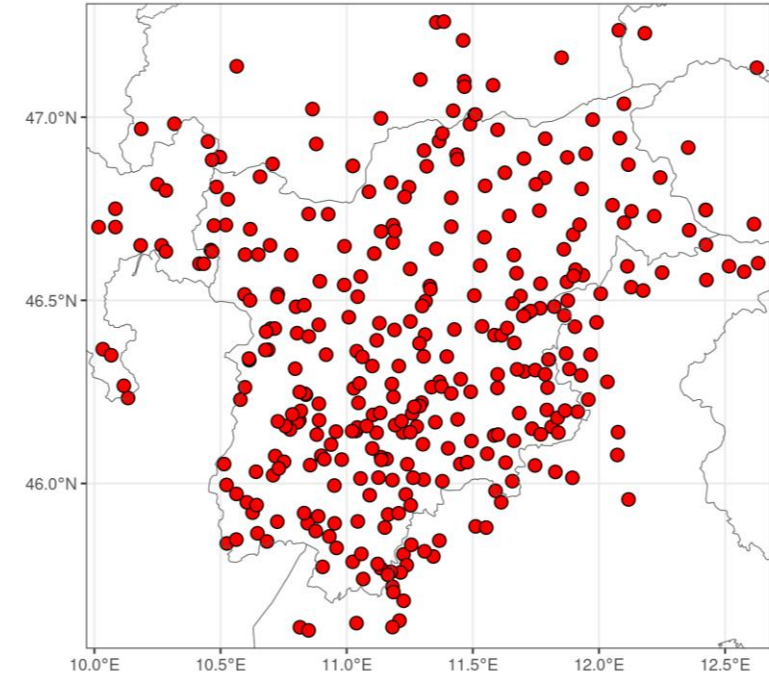
Providing spatially distributed meteorological inputs for impact models (e.g., for hydrological simulations)



# Meteorological data and data preparation

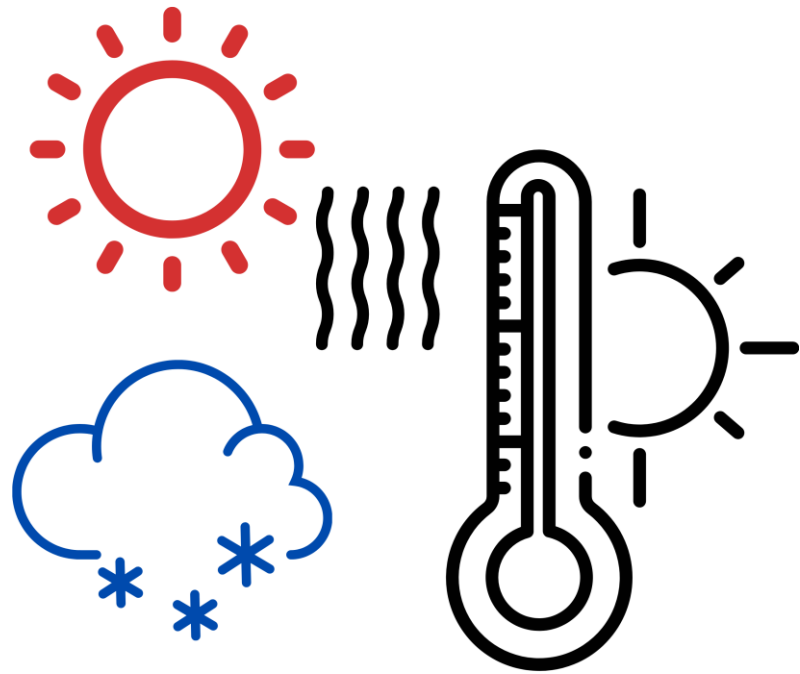


Time series of key weather variables:  
e.g., precipitation, min/max  
temperatures.

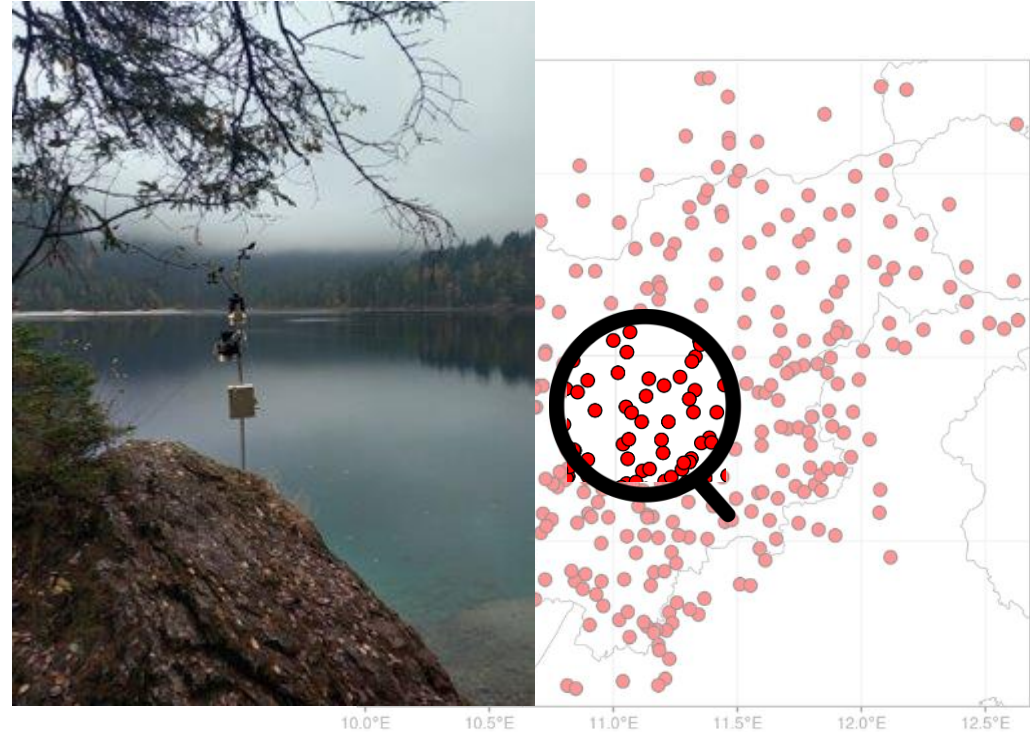


These data are recorded at daily  
and sub-daily resolution by  
weather stations

# Meteorological data and data preparation

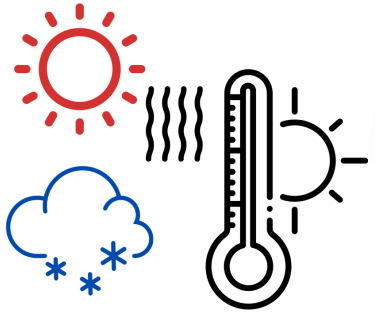


Time series of key weather variables:  
e.g., precipitation, min/max  
temperatures.

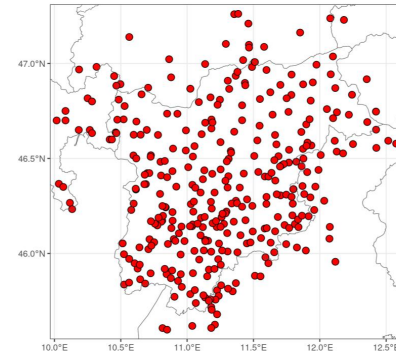


These data are recorded at daily  
and sub-daily resolution by  
weather stations

# Meteorological data and data preparation

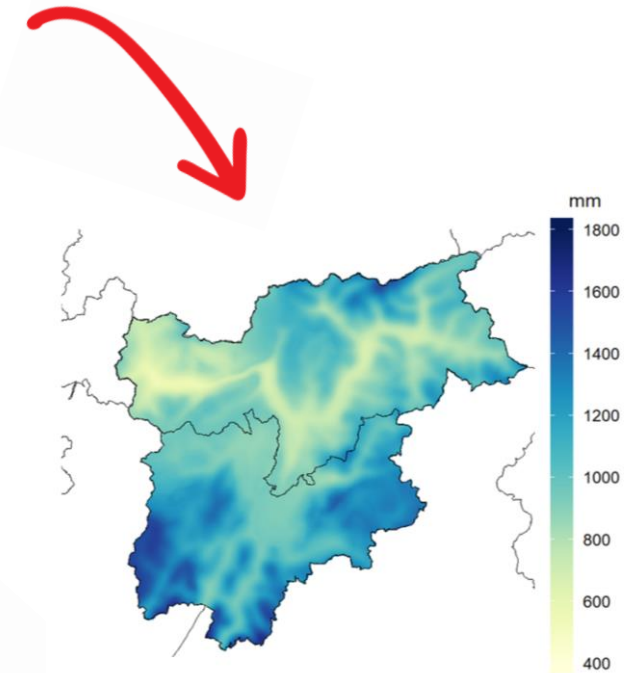


Comprehensive and updated time-series dataset: a robust compilation of at-site meteorological conditions



Irregular spatial coverage of weather stations

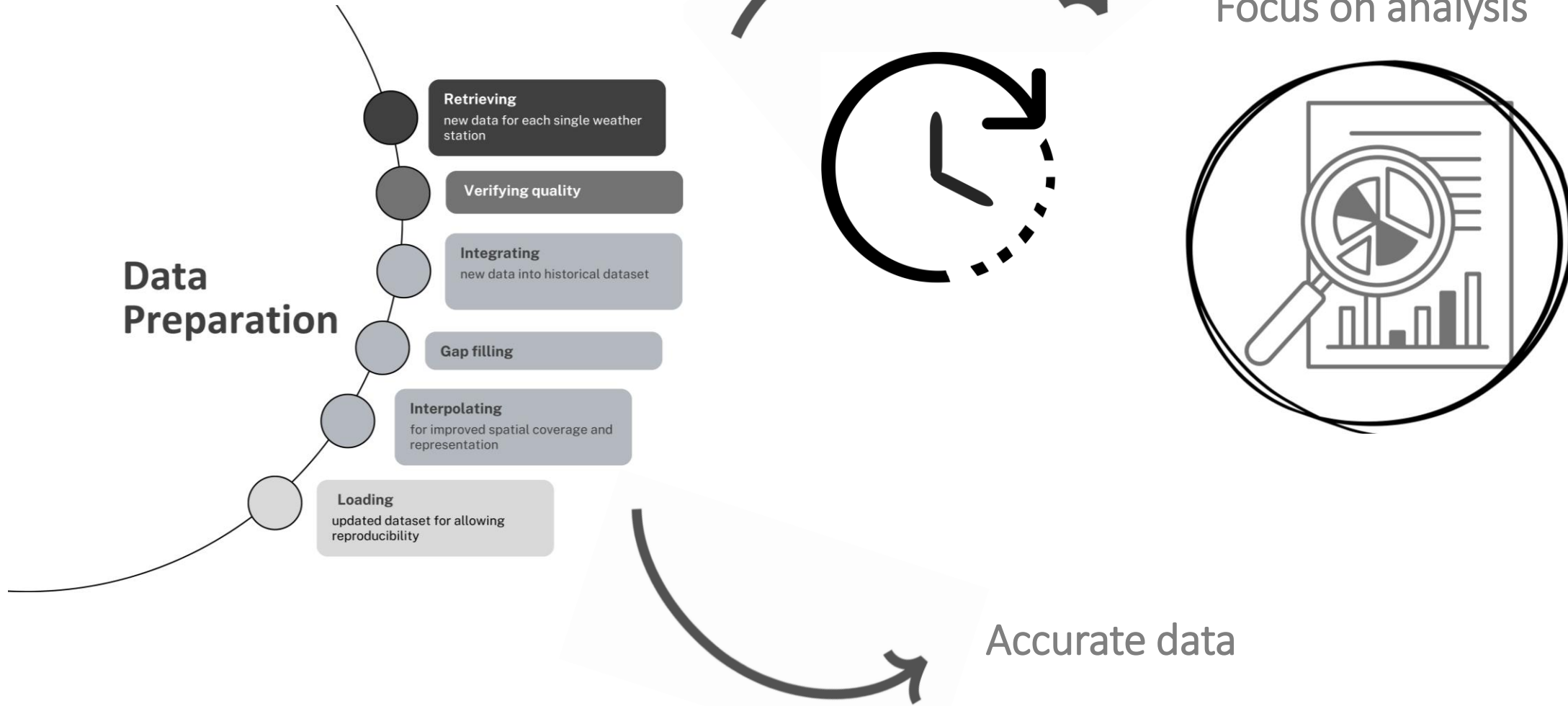
Interpolation for obtaining a gridded dataset



PostgreSQL

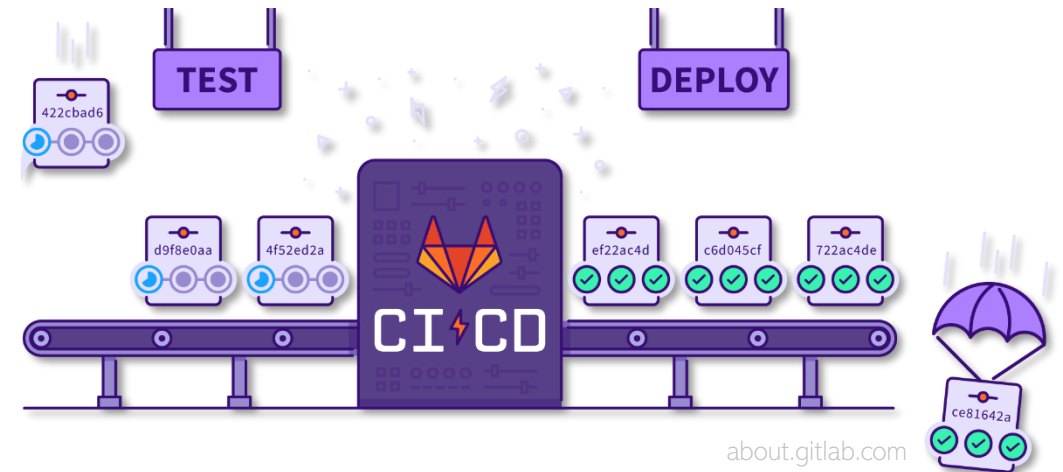


# Pipeline: motivation



# Pipeline: implementation

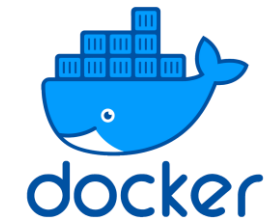
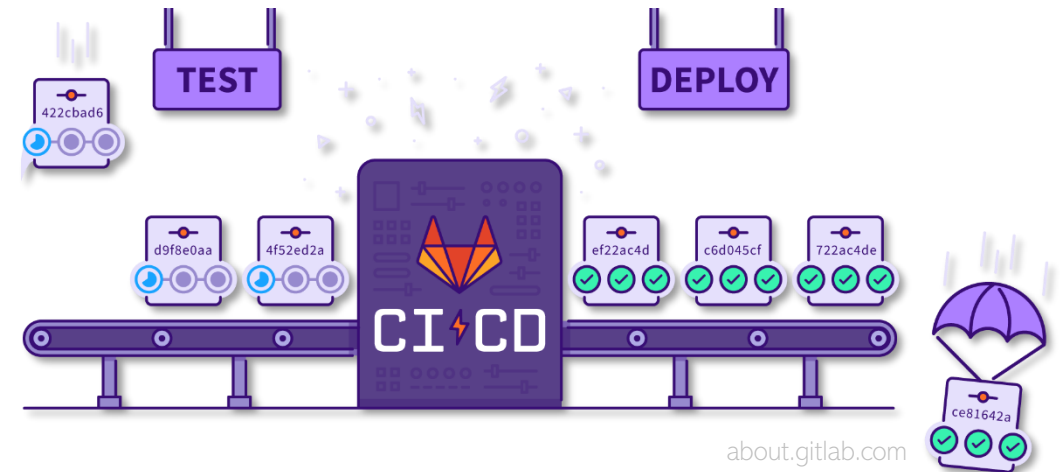
➤ GitLab CI/CD pipeline





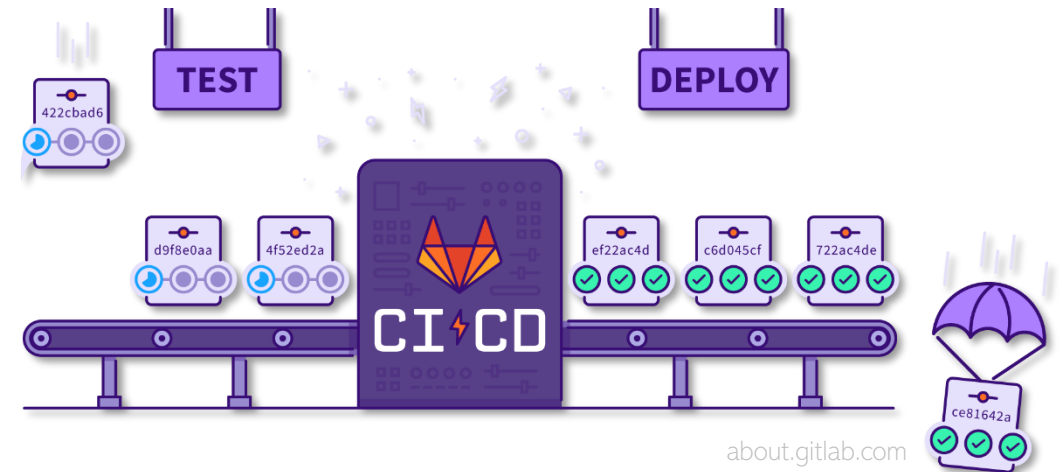
# Pipeline: implementation

- GitLab CI/CD pipeline
- Dockerized jobs runners



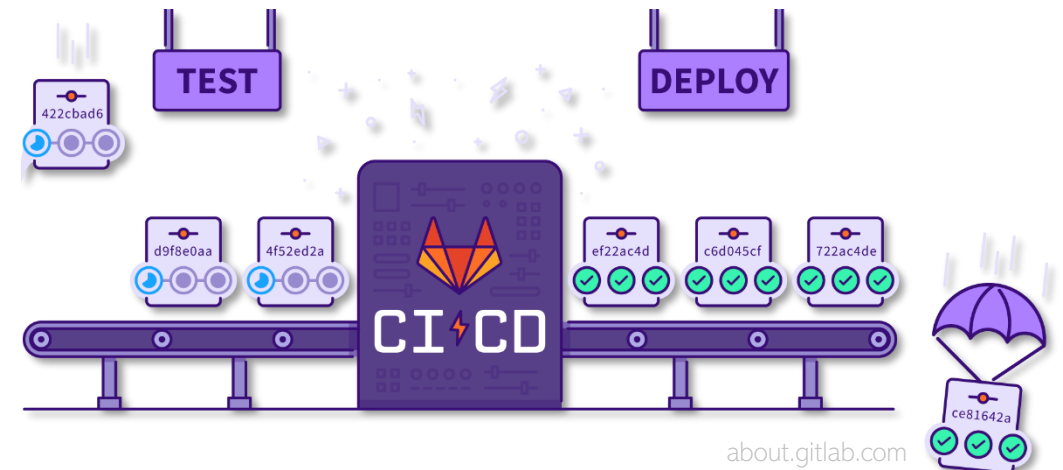
# Pipeline: implementation

- GitLab CI/CD pipeline
- Dockerized jobs runners
- R modules + geo-spatial libraries



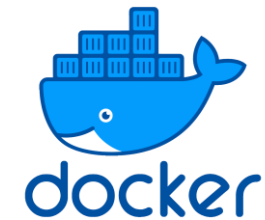
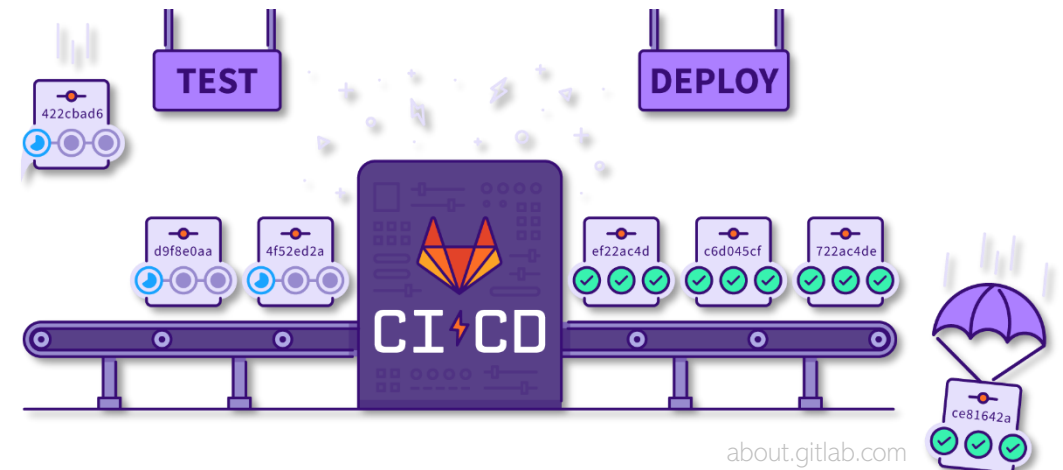
# Pipeline: implementation

- GitLab CI/CD pipeline
- Dockerized jobs runners
- R modules + geo-spatial libraries
- testthat + shell-based unit test suites



# Pipeline: implementation

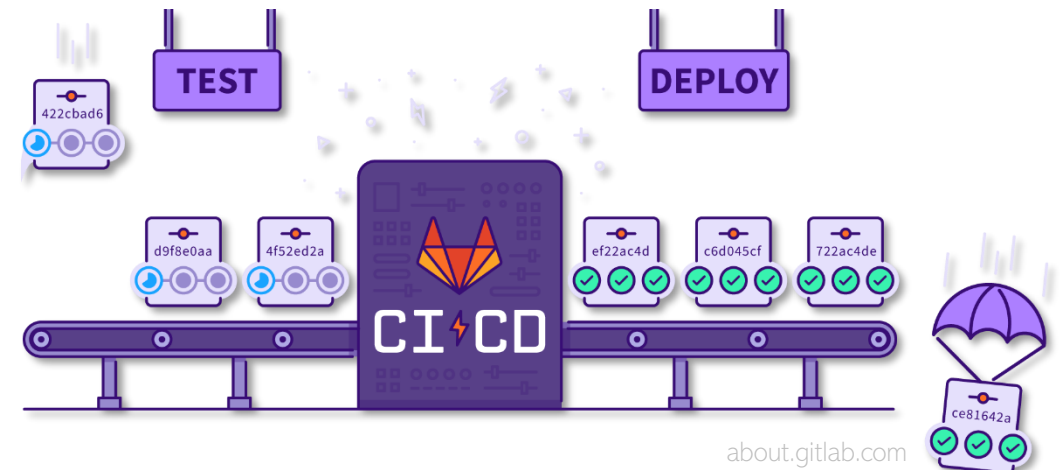
- GitLab CI/CD pipeline
- Dockerized jobs runners
- R modules + geo-spatial libraries
- testthat + shell-based unit test suites
- renv for reproducible environment



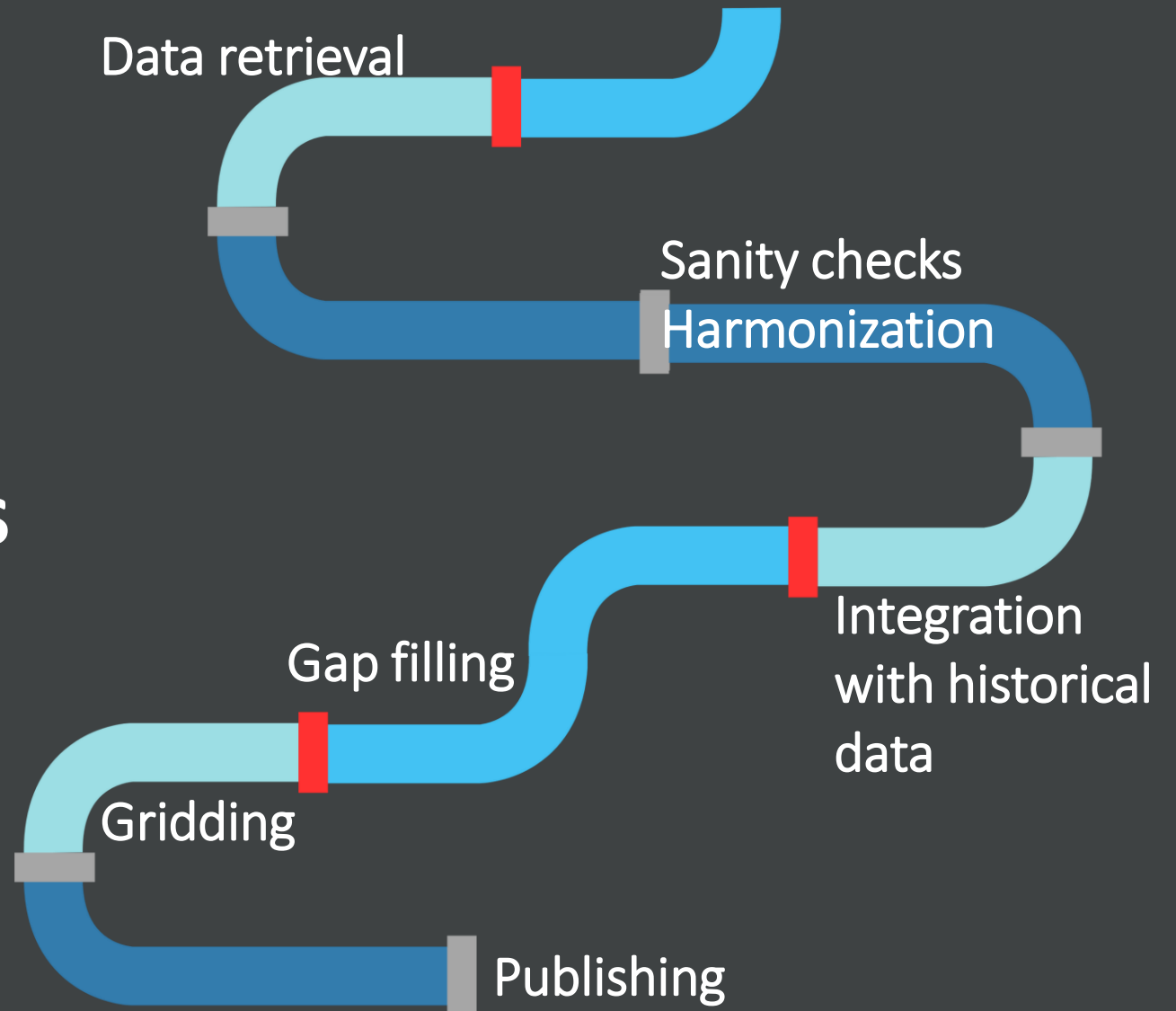
test  
that

# Pipeline: implementation

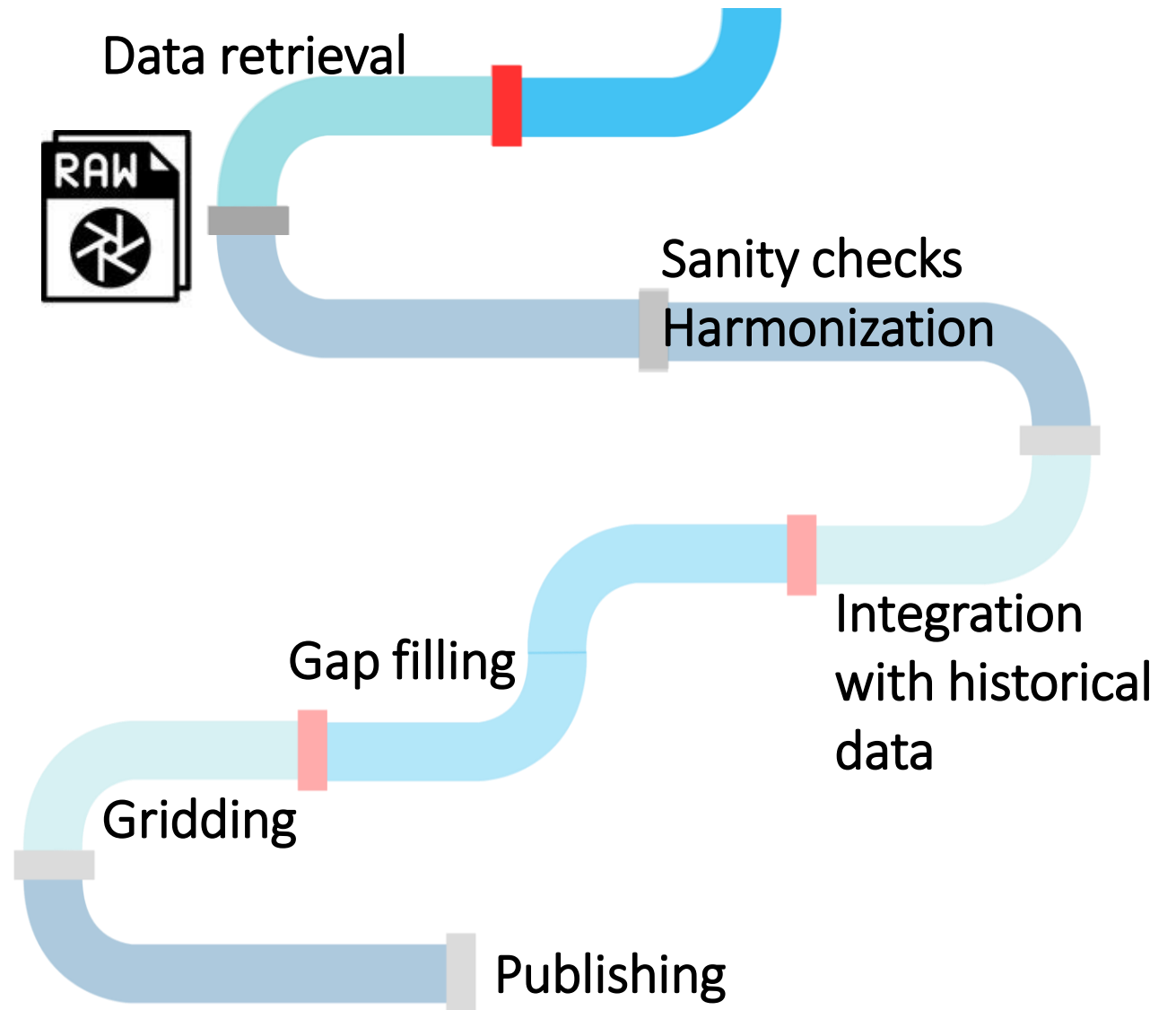
- GitLab CI/CD pipeline
- Dockerized jobs runners
- R modules + geo-spatial libraries
- testthat + shell-based unit test suites
- renv for reproducible environment
- bash scripts and wrappers



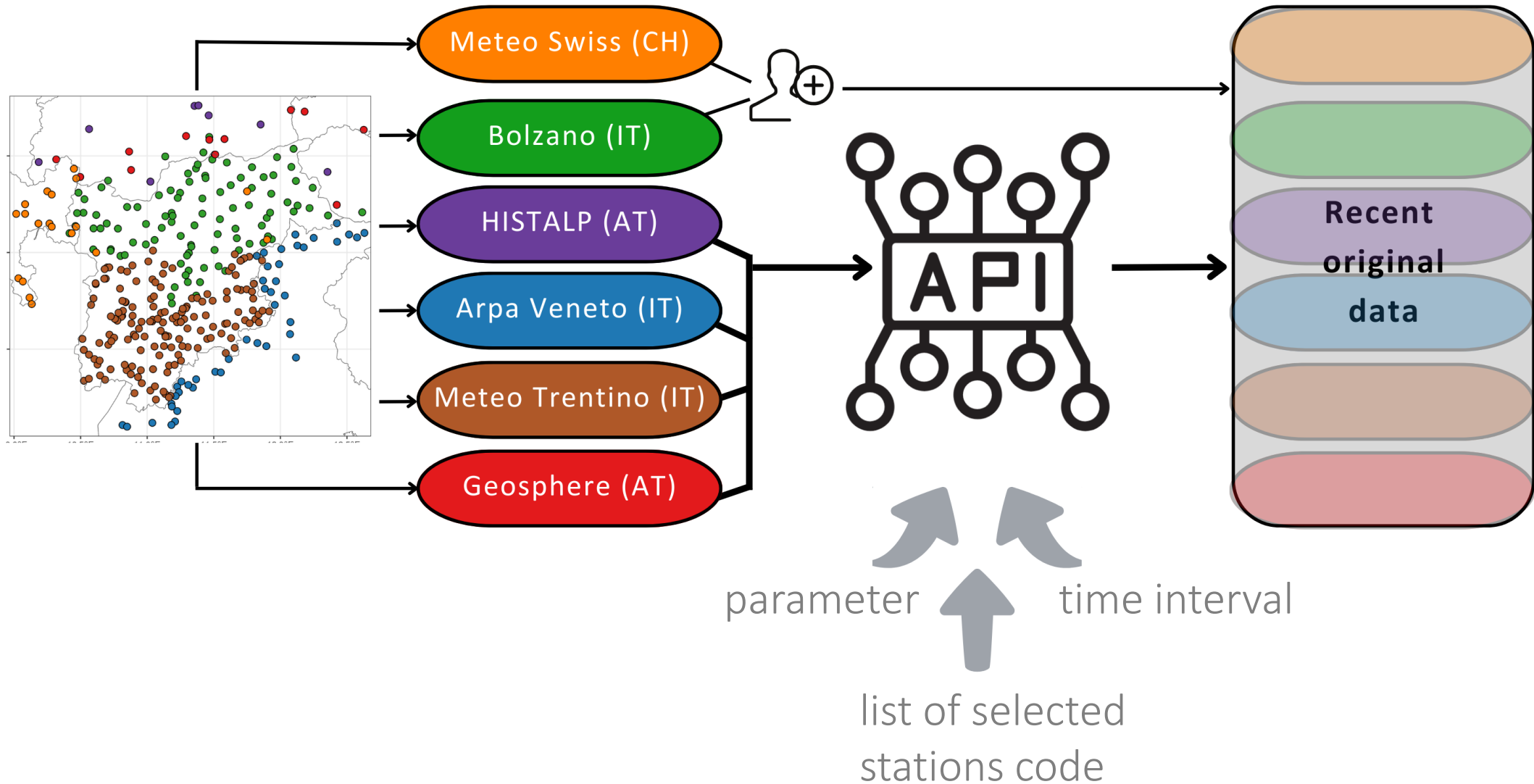
# ETL pipeline for updating meteorological datasets in near real-time



# ETL pipeline for updating meteorological datasets in near real-time

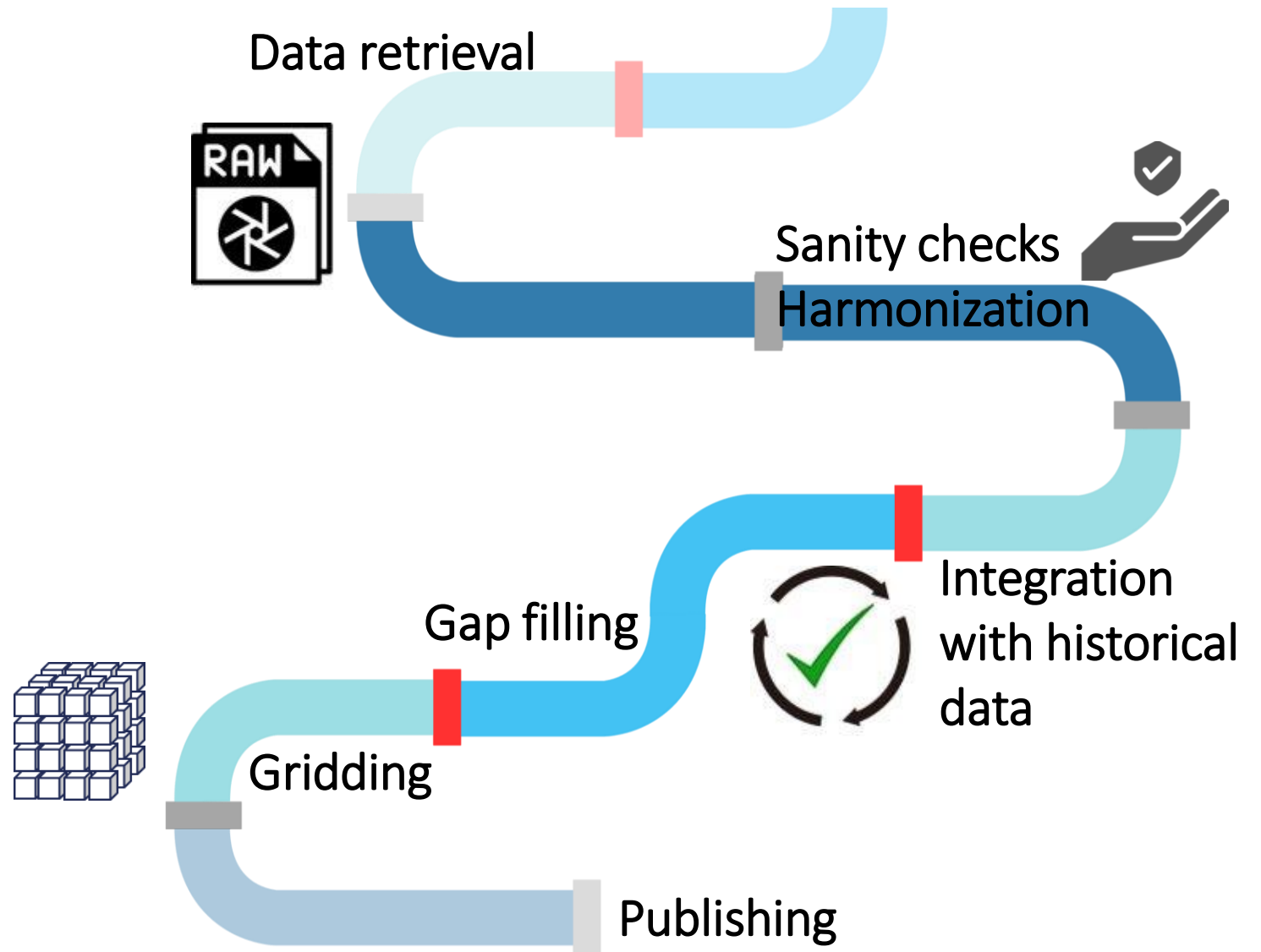


# 01 : Data extraction

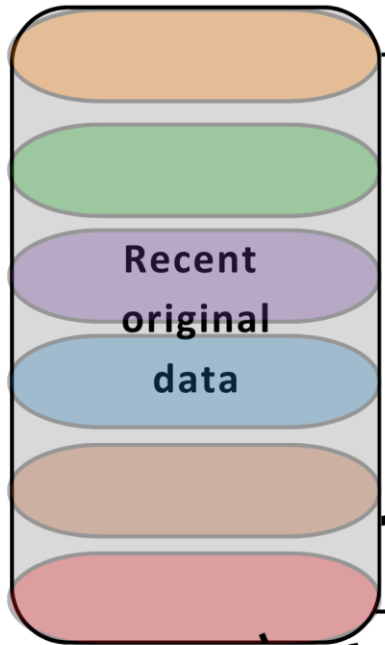
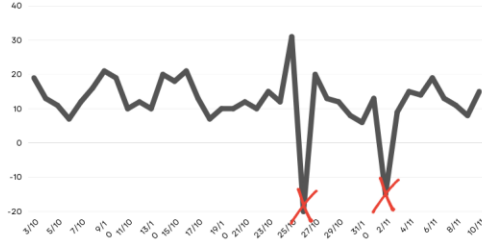




# ETL pipeline for updating meteorological datasets in near real-time



# 02 : Sanity checks and harmonization



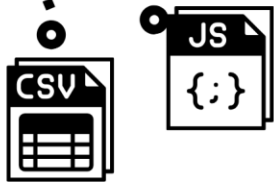
REMOVE INCONSISTENCIES

CLEANING USING QUALITY FLAGS



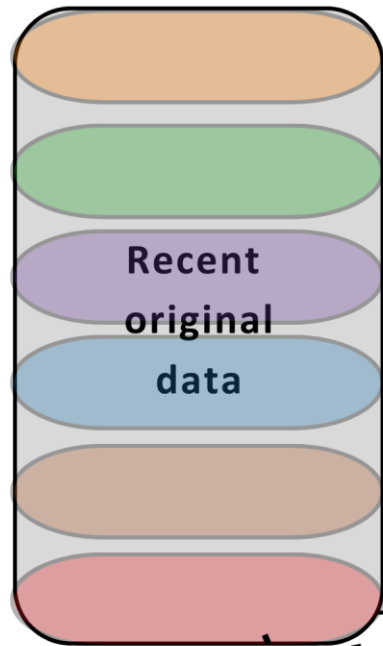
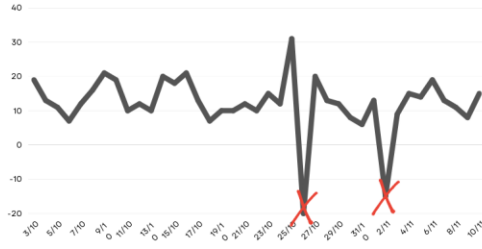
Date	Value	Quality flag
3/10/2023	<del>13</del>	bad!
4/10/2023	<del>13</del>	nope!
5/10/2023	11	ok!

HARMONIZATION



common data structure

# 02 : Sanity checks and harmonization



REMOVE INCONSISTENCIES

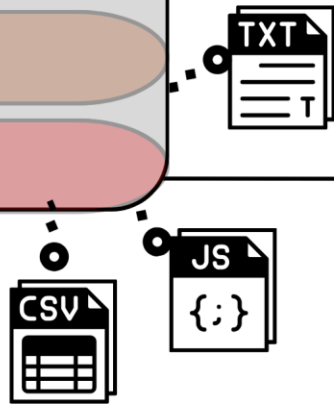
CLEANING USING QUALITY FLAGS

Date	Value	Quality flag
3/10/2023	13	bad!
4/10/2023	11	nope!
5/10/2023	11	ok!



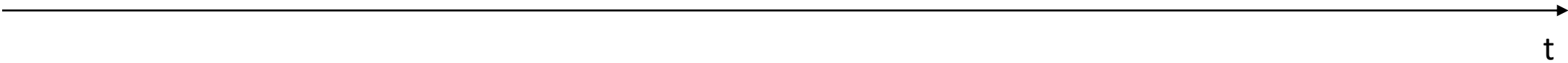
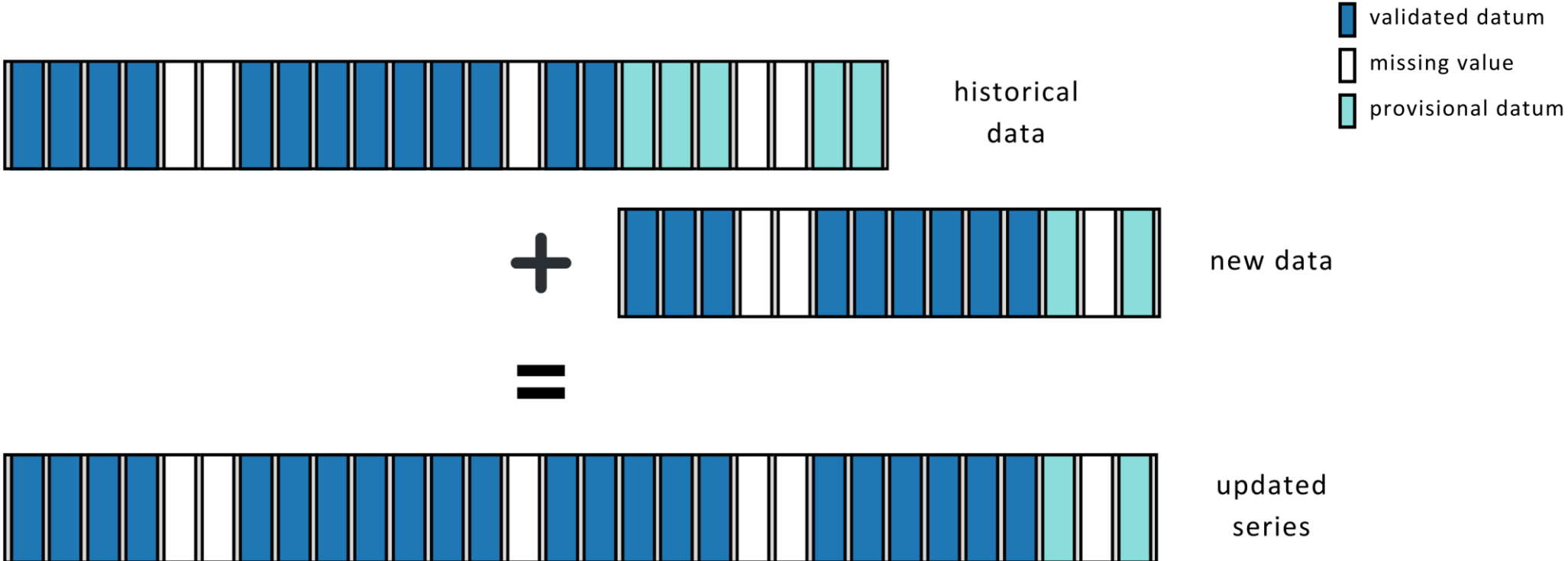
Daily mean temperature calculated as  $\text{mean}(\text{min}, \text{max})$

HARMONIZATION

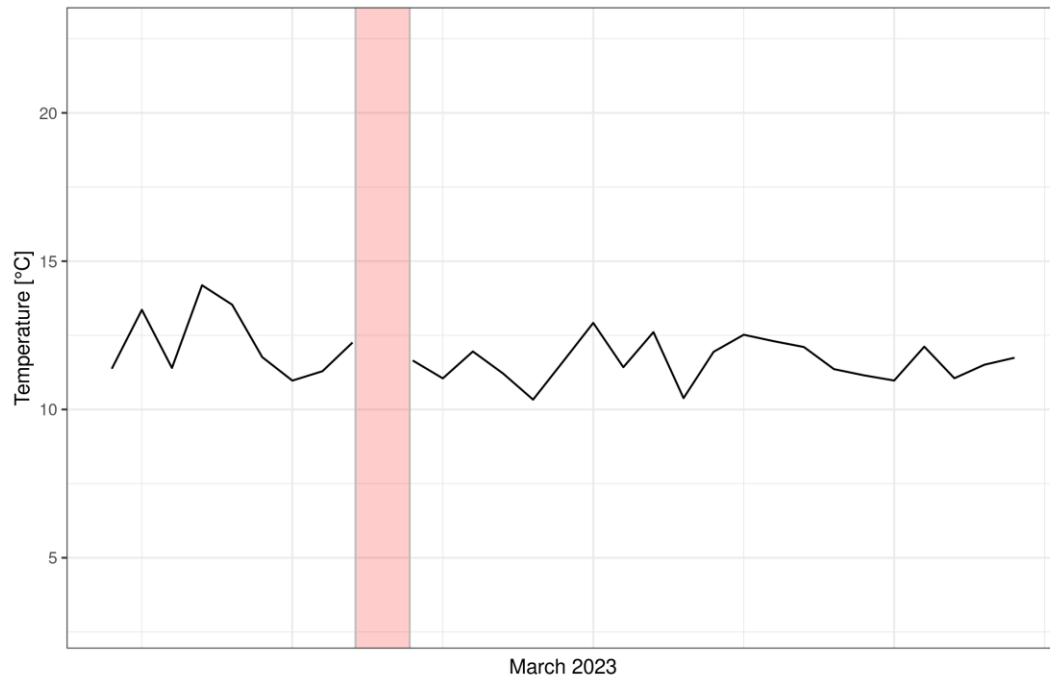


common data structure

# 03 : Integration with historical data

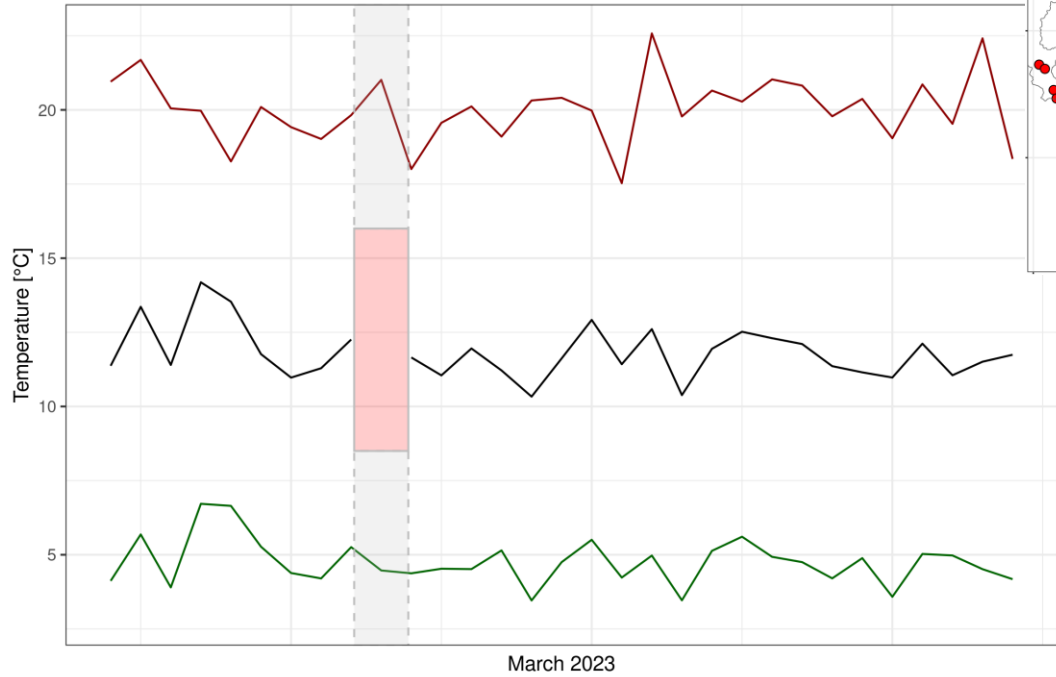
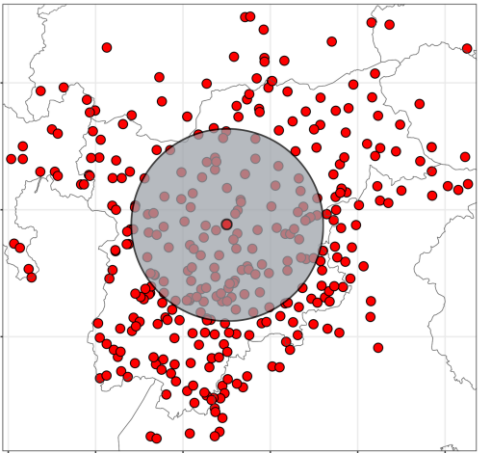


# 04 : Gap filling



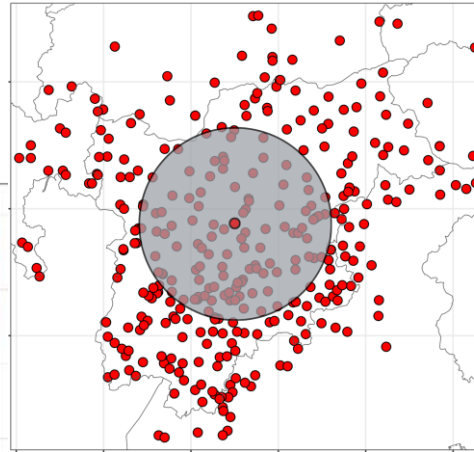
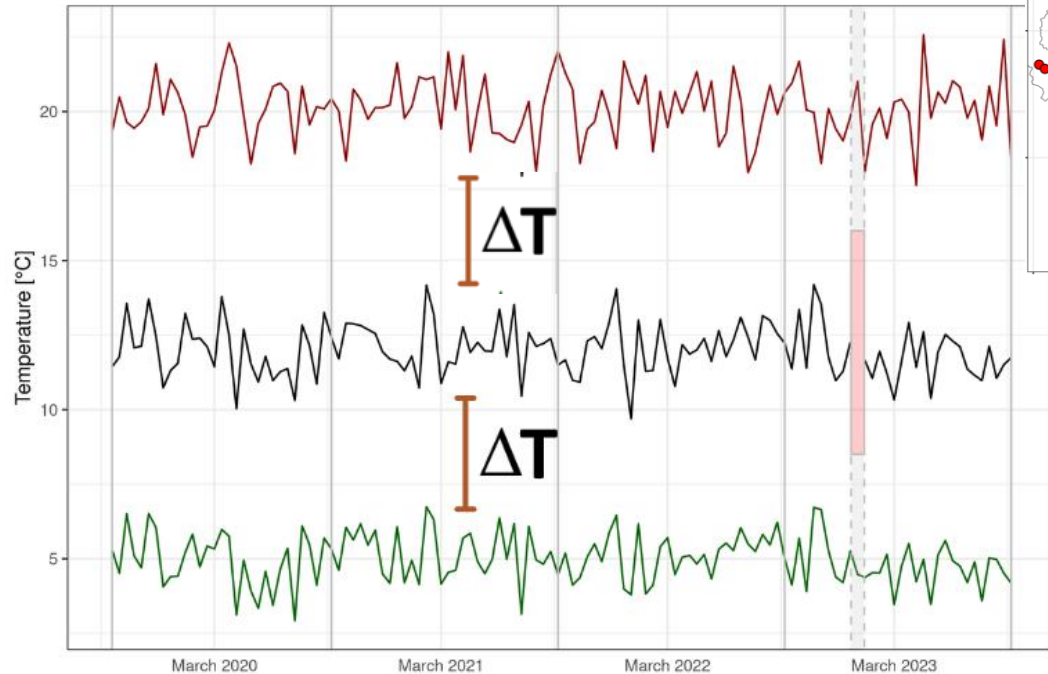
Only isolated gaps

# 04 : Gap filling



Only isolated gaps  
Interpolation based on  
surrounding  
stations

# 04 : Gap filling

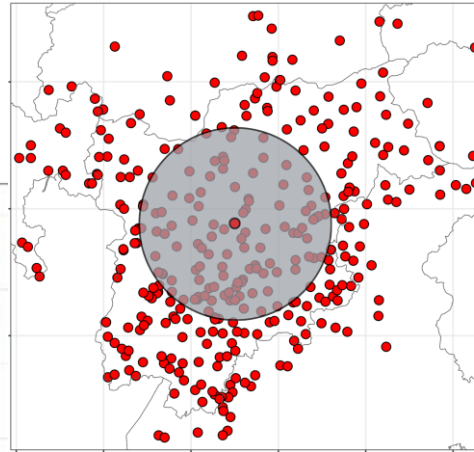
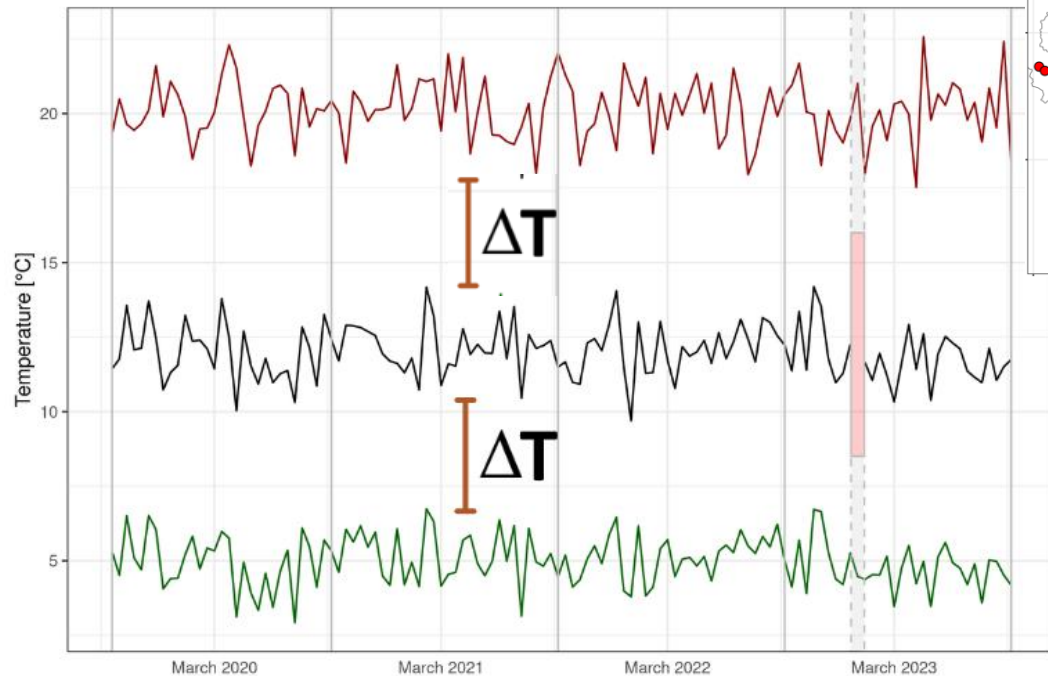


Only isolated gaps

Interpolation based on surrounding stations

Considering common historical data

# 04 : Gap filling



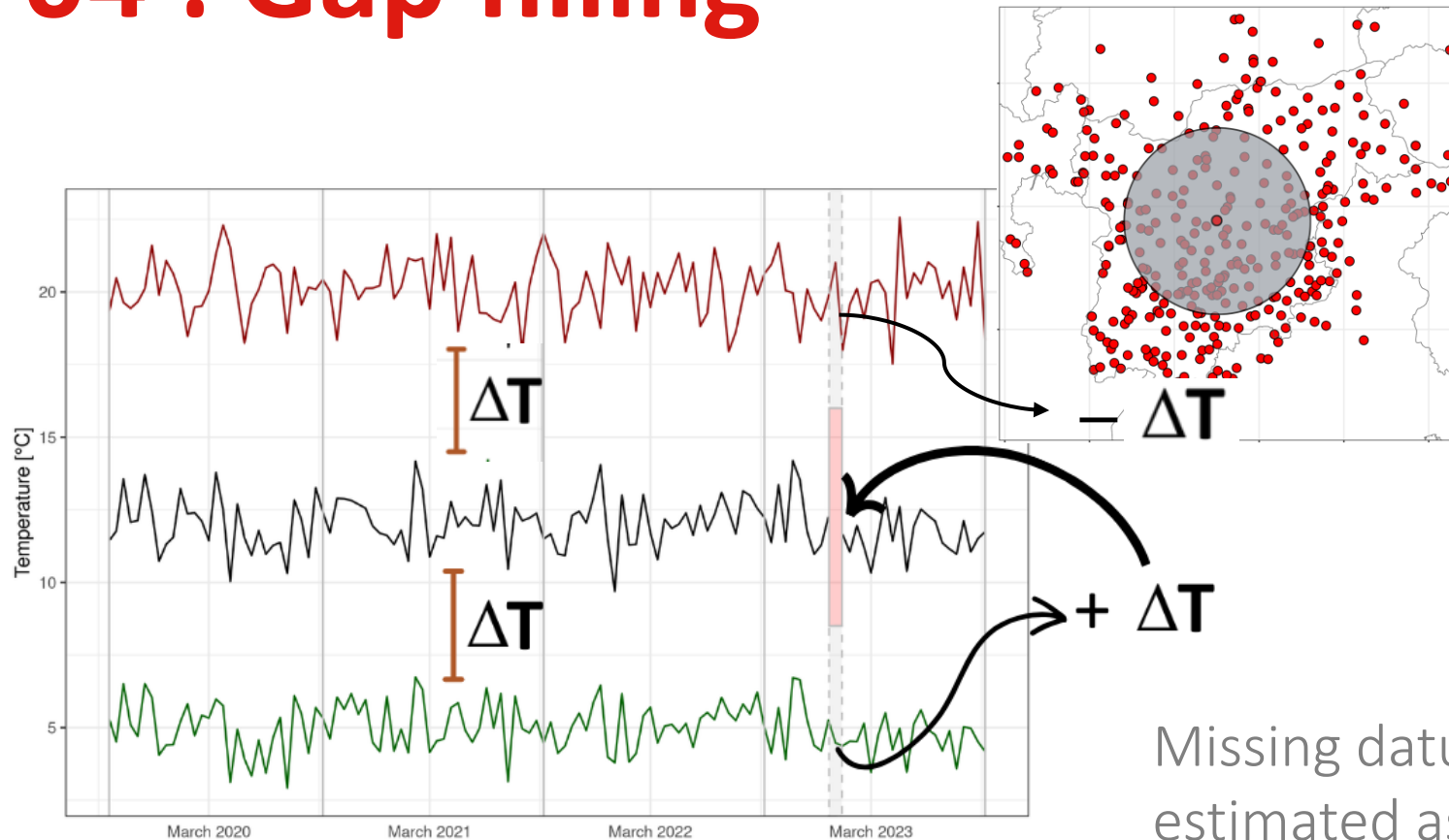
Only isolated gaps

Interpolation based on surrounding **correlated** stations

Considering common historical data



# 04 : Gap filling



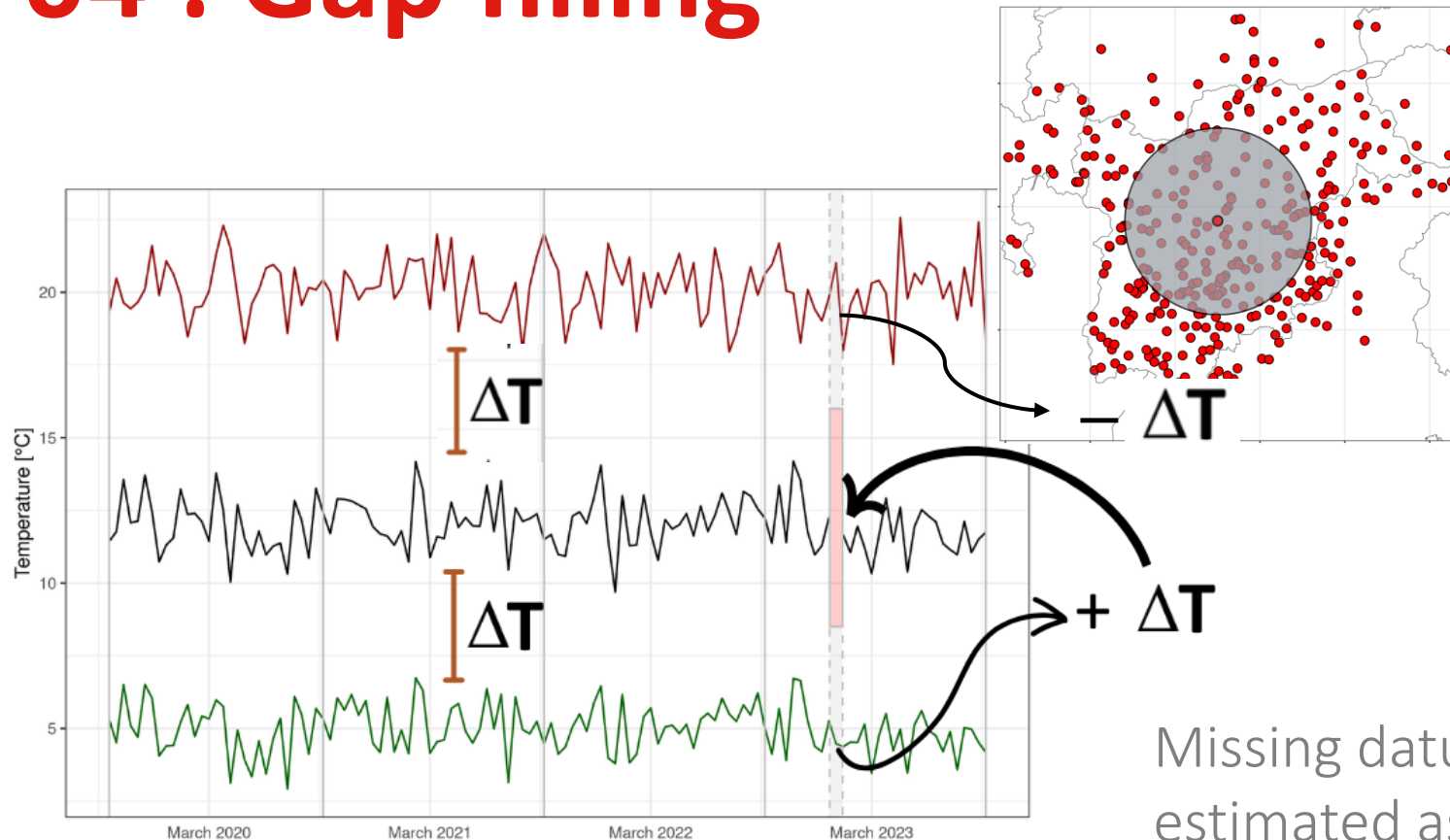
Only isolated gaps

Interpolation based on surrounding **correlated** stations

Considering common historical data

Missing datum estimated as the weighted average of rescaled values

# 04 : Gap filling



Simulated values are **checked for consistency and mean temperature missing values** are determined from reconstructed values of min and max temperatures

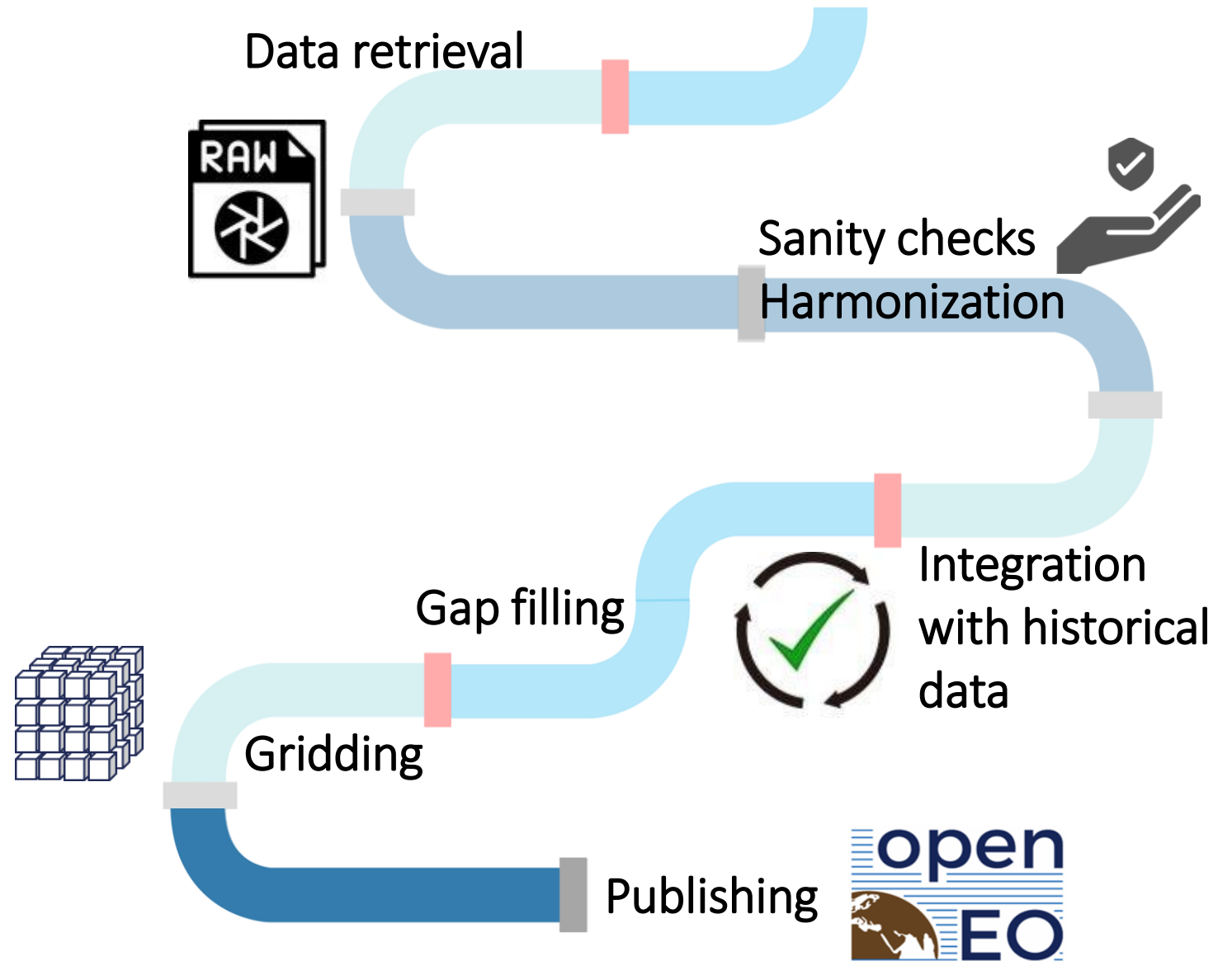
Only isolated gaps

Interpolation based on surrounding **correlated** stations

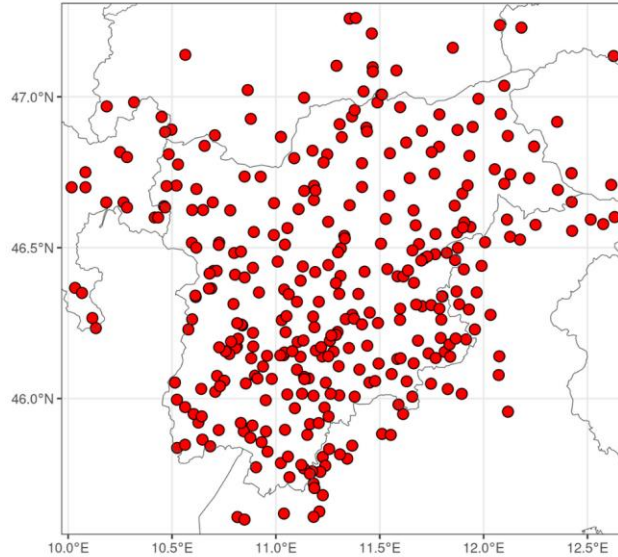
Considering common historical data

Missing datum estimated as the weighted average of rescaled values

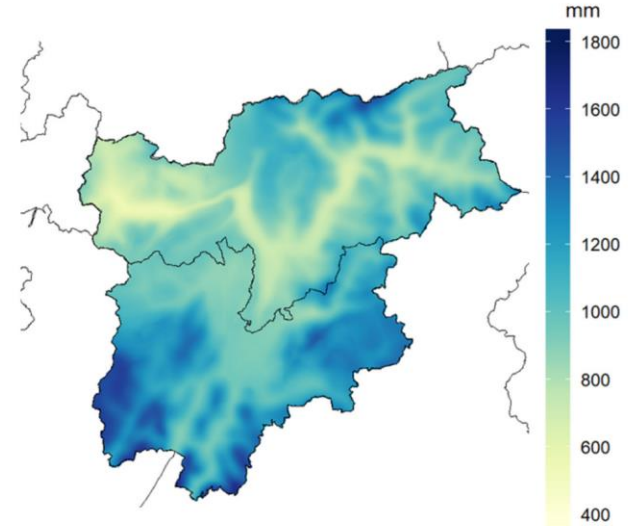
# ETL pipeline for updating meteorological datasets in near real-time



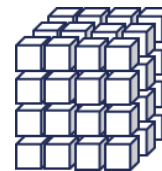
# 05 : Interpolation/gridding



daily at-site time series



daily grids



250m x 250m

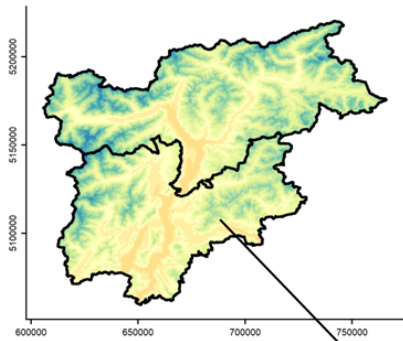
NetCDF

# 05 : Interpolation/gridding: temperature

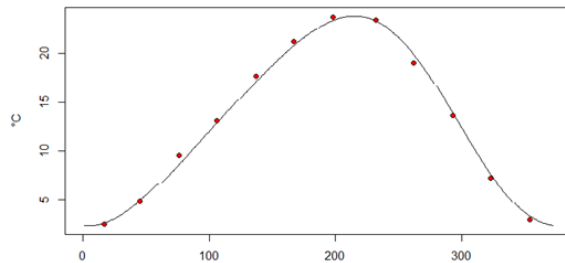
## *Local weighted linear regression*

of temperature normals vs  
elevation and weights depending  
on topographic features

$$\bar{t}_m(x, y) = a_m(x, y) + b_m(x, y) \cdot h(x, y)$$



*Gridded 1981-2010  
March climatologies of  
Tmax*

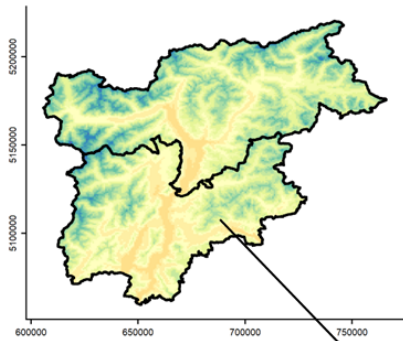


# 05 : Interpolation/gridding: temperature

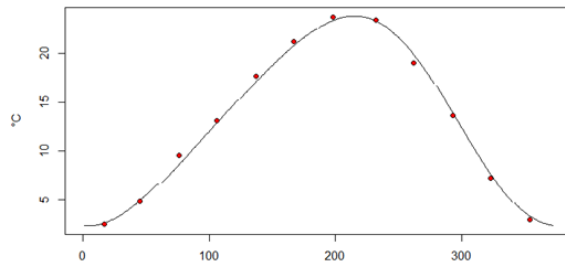
## *Local weighted linear regression*

of temperature normals vs  
elevation and weights depending  
on topographic features

$$\bar{t}_m(x, y) = a_m(x, y) + b_m(x, y) \cdot h(x, y)$$



*Gridded 1981-2010  
March climatologies of  
Tmax*



# 05 : Interpolation/gridding: temperature

**Local weighted linear regression** of temperature normals vs elevation and weights depending on topographic features

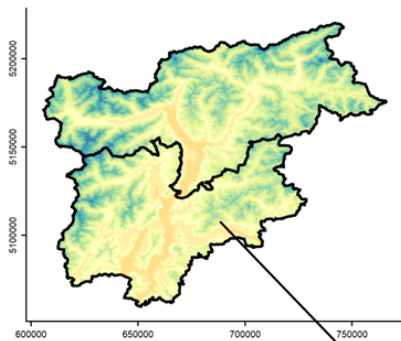
$$\bar{t}_m(x, y) = a_m(x, y) + b_m(x, y) \cdot h(x, y)$$



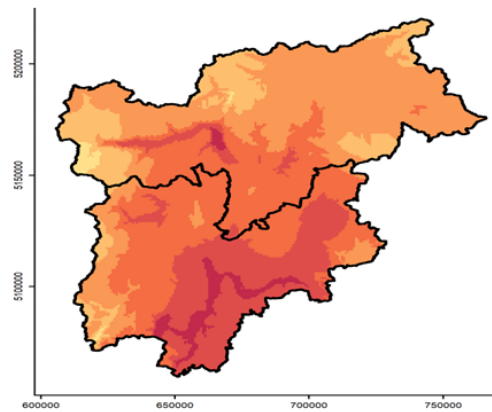
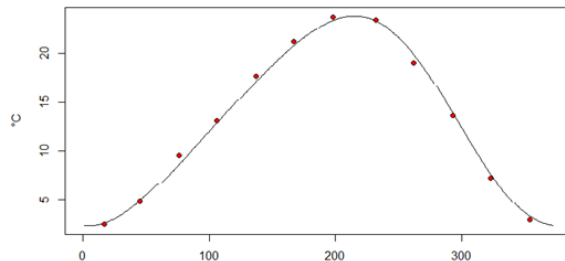
**Daily normals** based on a trigonometric fit from monthly normals

**Local weighted average** of station daily anomalies based on distance and elevation difference

$$a_d(x, y) = \frac{\sum_j w_j(x, y) \cdot a_{d,j}}{\sum_j w_j(x, y)}$$



Gridded 1981-2010  
March climatologies of  
Tmax



Gridded anomalies on 5<sup>th</sup> March 2001

# 05 : Interpolation/gridding: temperature

**Local weighted linear regression** of temperature normals vs elevation and weights depending on topographic features

$$\bar{t}_m(x, y) = a_m(x, y) + b_m(x, y) \cdot h(x, y)$$



**Daily normals** based on a trigonometric fit from monthly normals

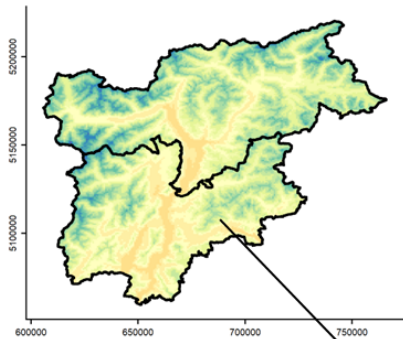
**Local weighted average** of station daily anomalies based on distance and elevation difference

$$a_d(x, y) = \frac{\sum_j w_j(x, y) \cdot a_{d,j}}{\sum_j w_j(x, y)}$$

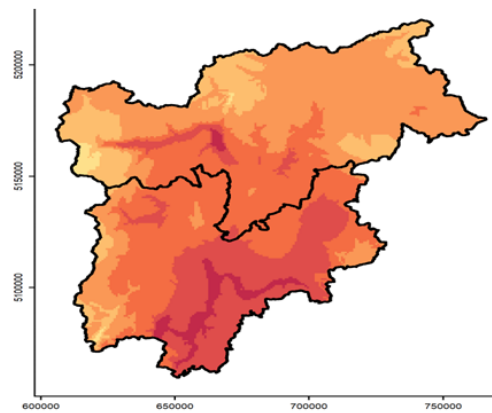
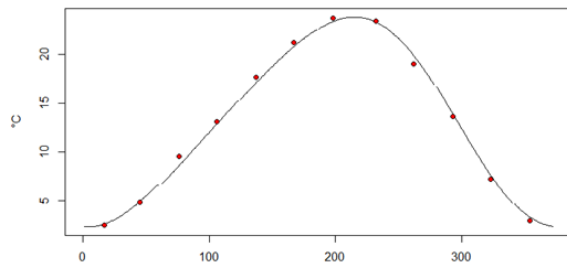


**Superimposition** (sum) of gridded anomalies and climatologies

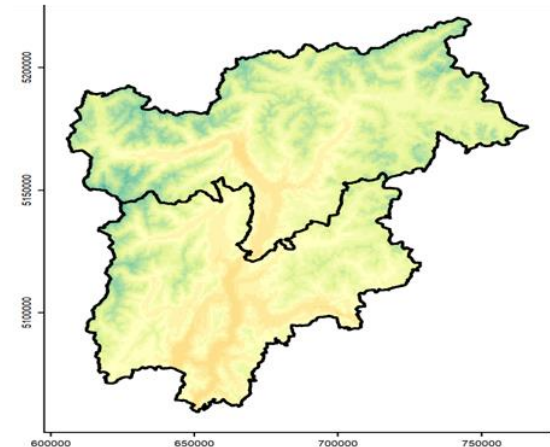
$$t_d(x, y) = a_d(x, y) + \bar{t}_d(x, y)$$



Gridded 1981-2010  
March climatologies of  
Tmax



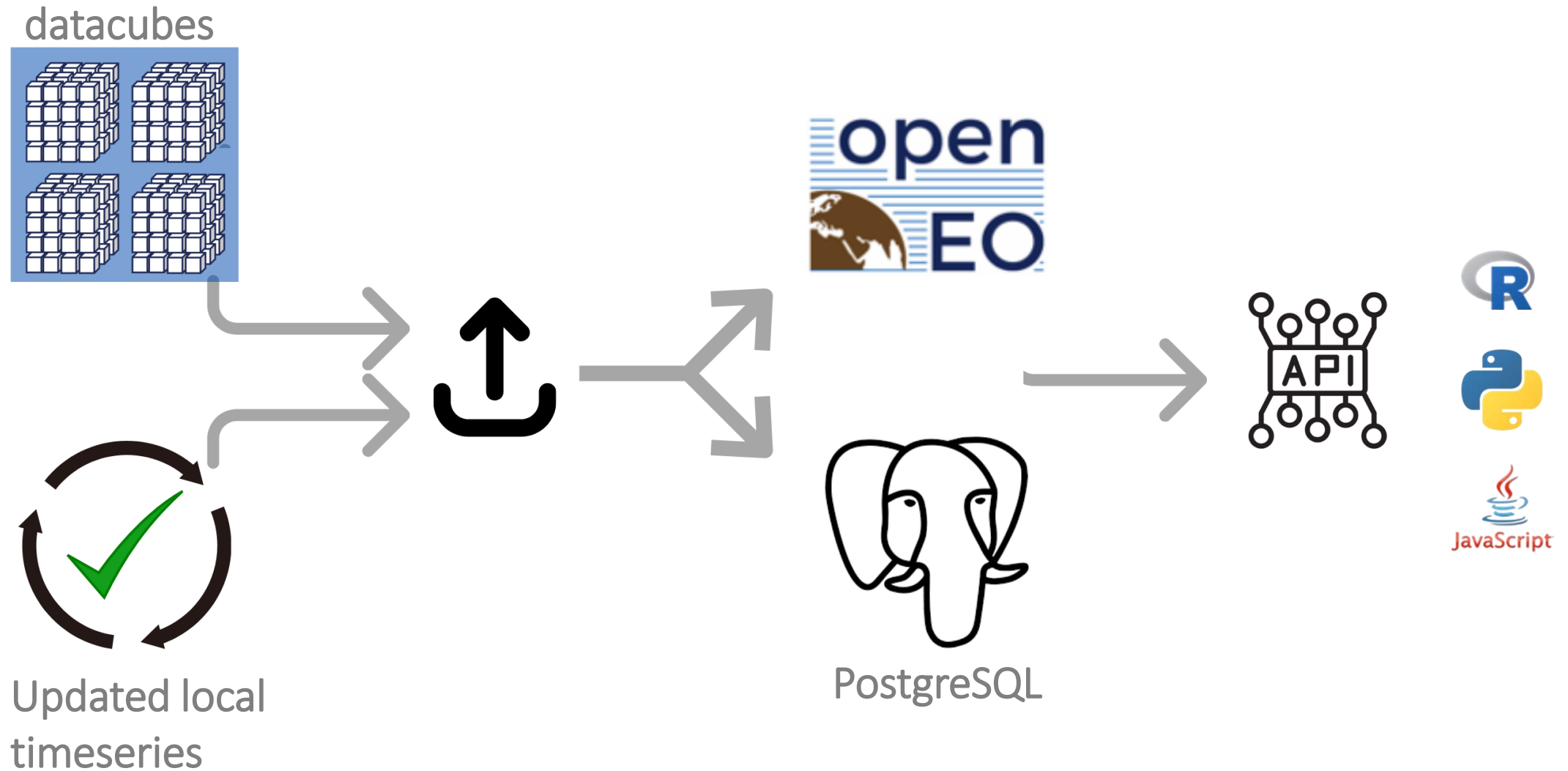
Gridded anomalies on 5<sup>th</sup> March 2001



Gridded Tmax on 5<sup>th</sup> July 2001

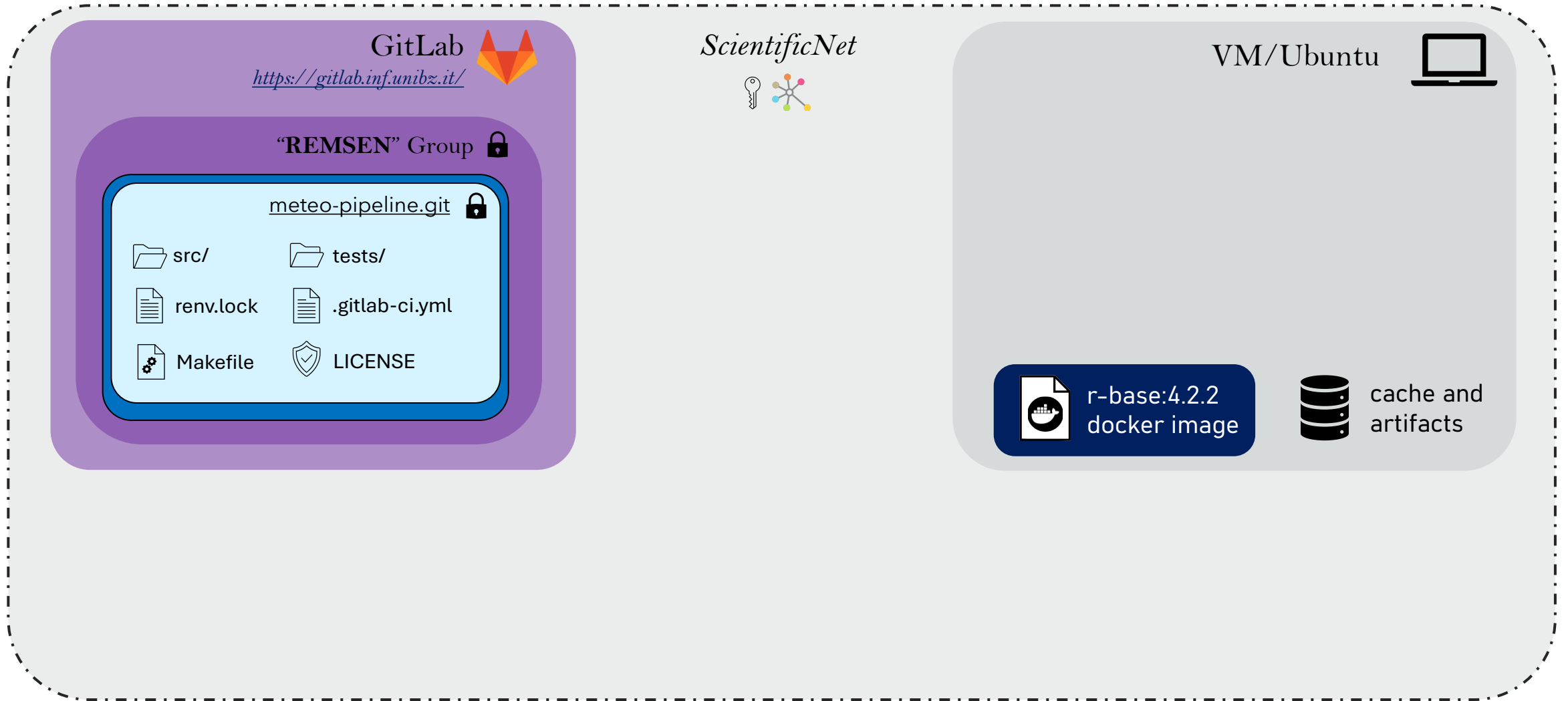


# 06 : Publishing

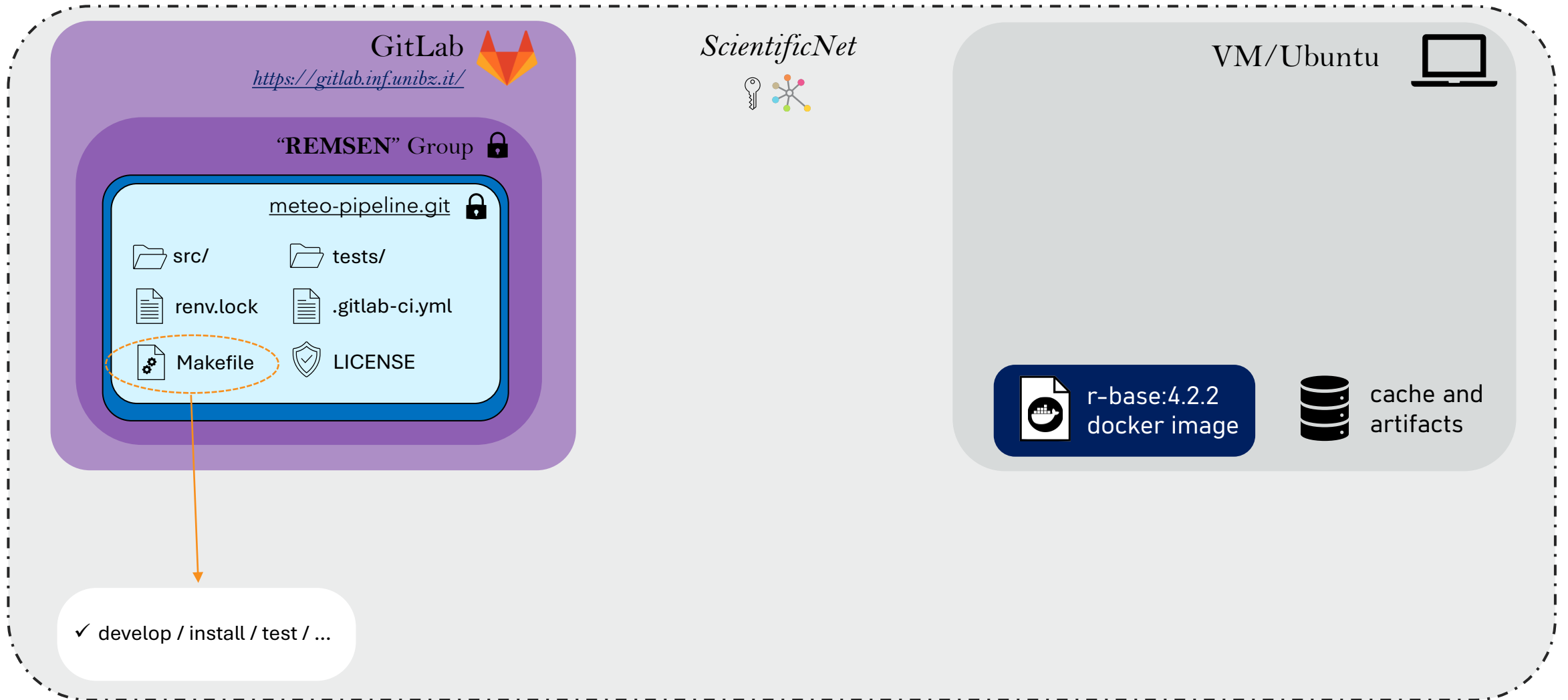


# Automating the pipeline

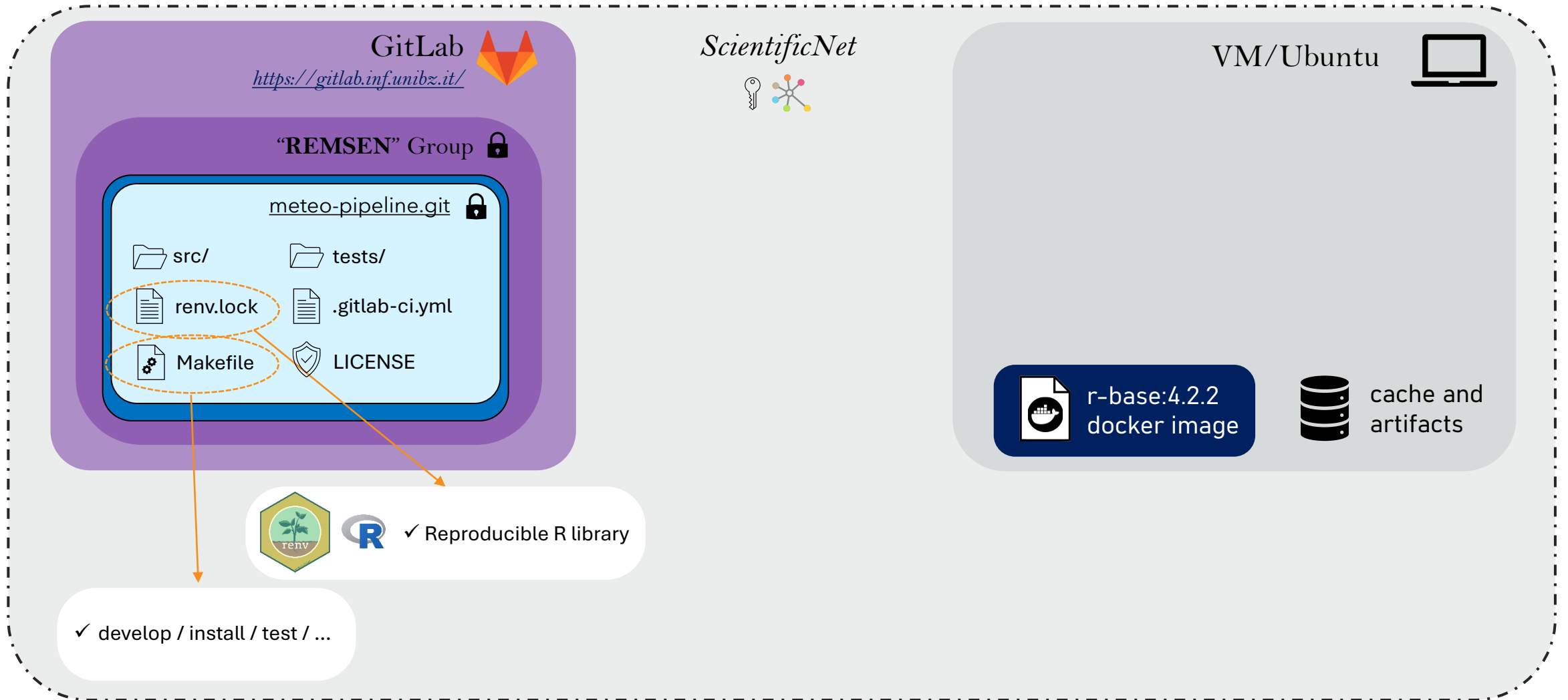
# Automating the pipeline



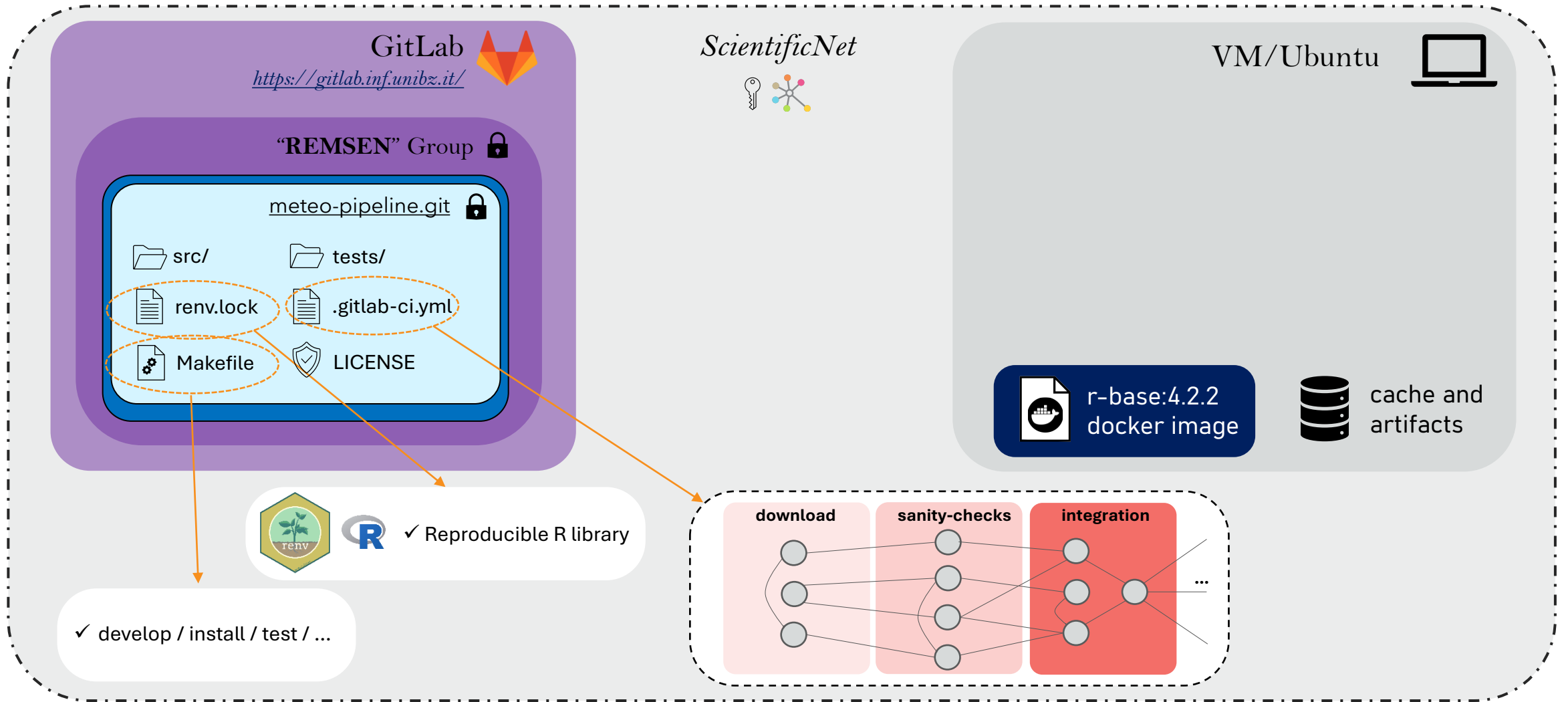
# Automating the pipeline



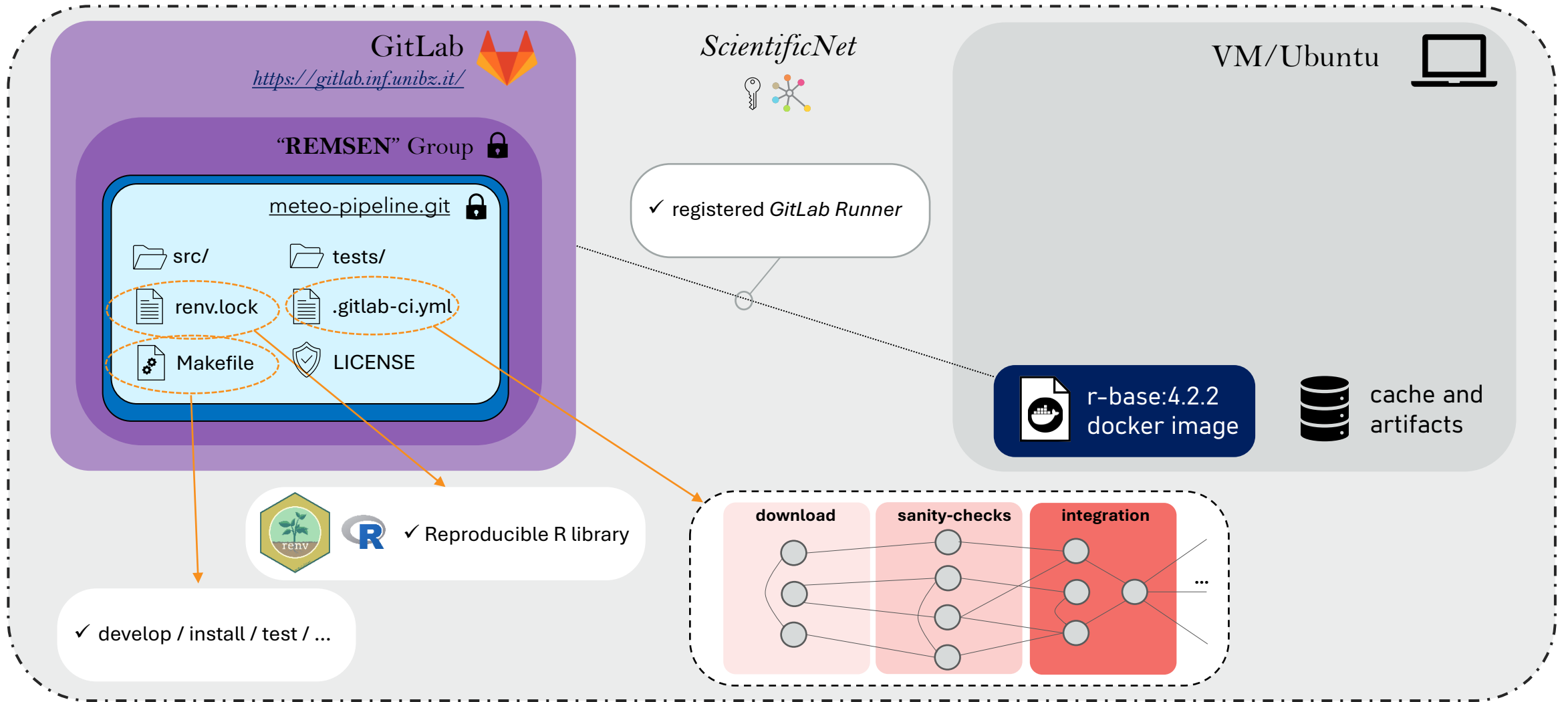
# Automating the pipeline



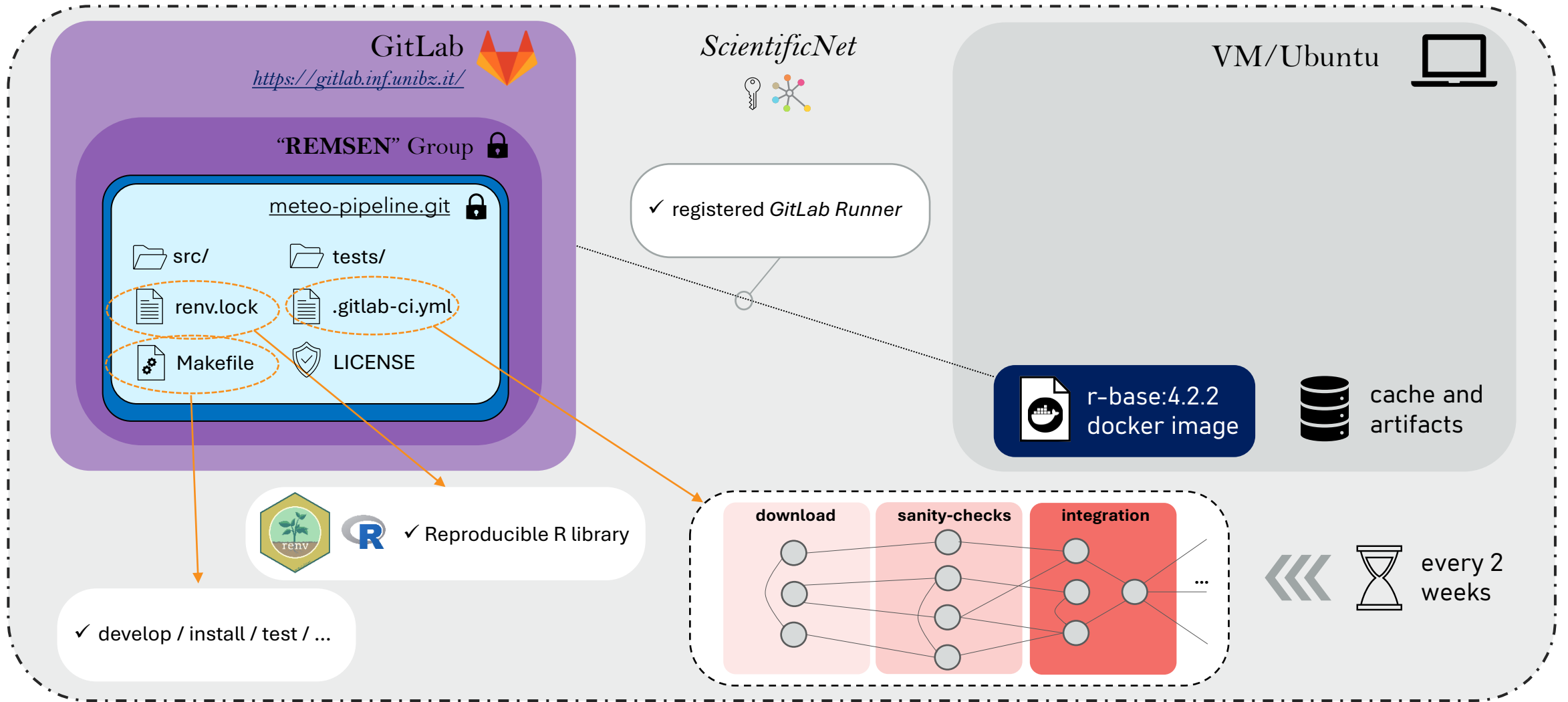
# Automating the pipeline



# Automating the pipeline

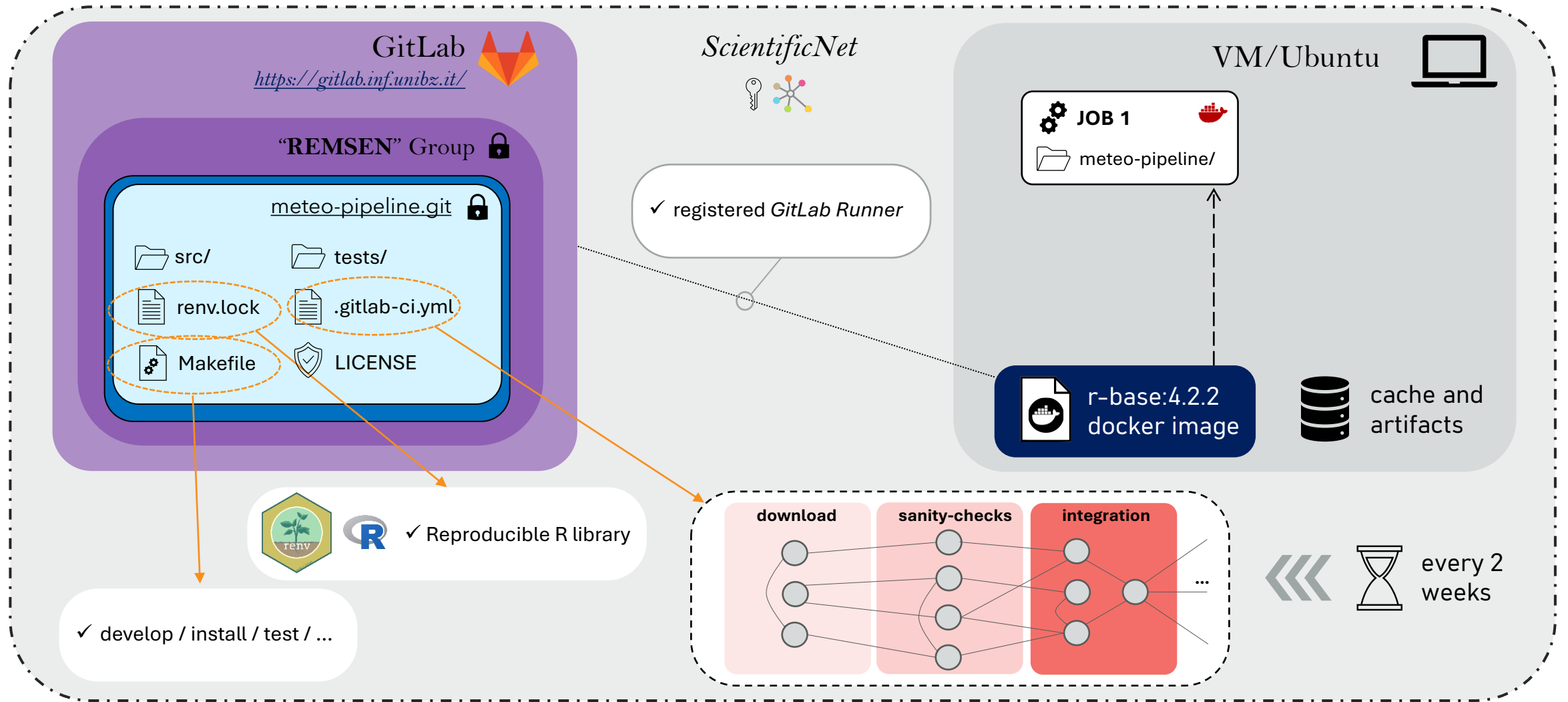


# Automating the pipeline

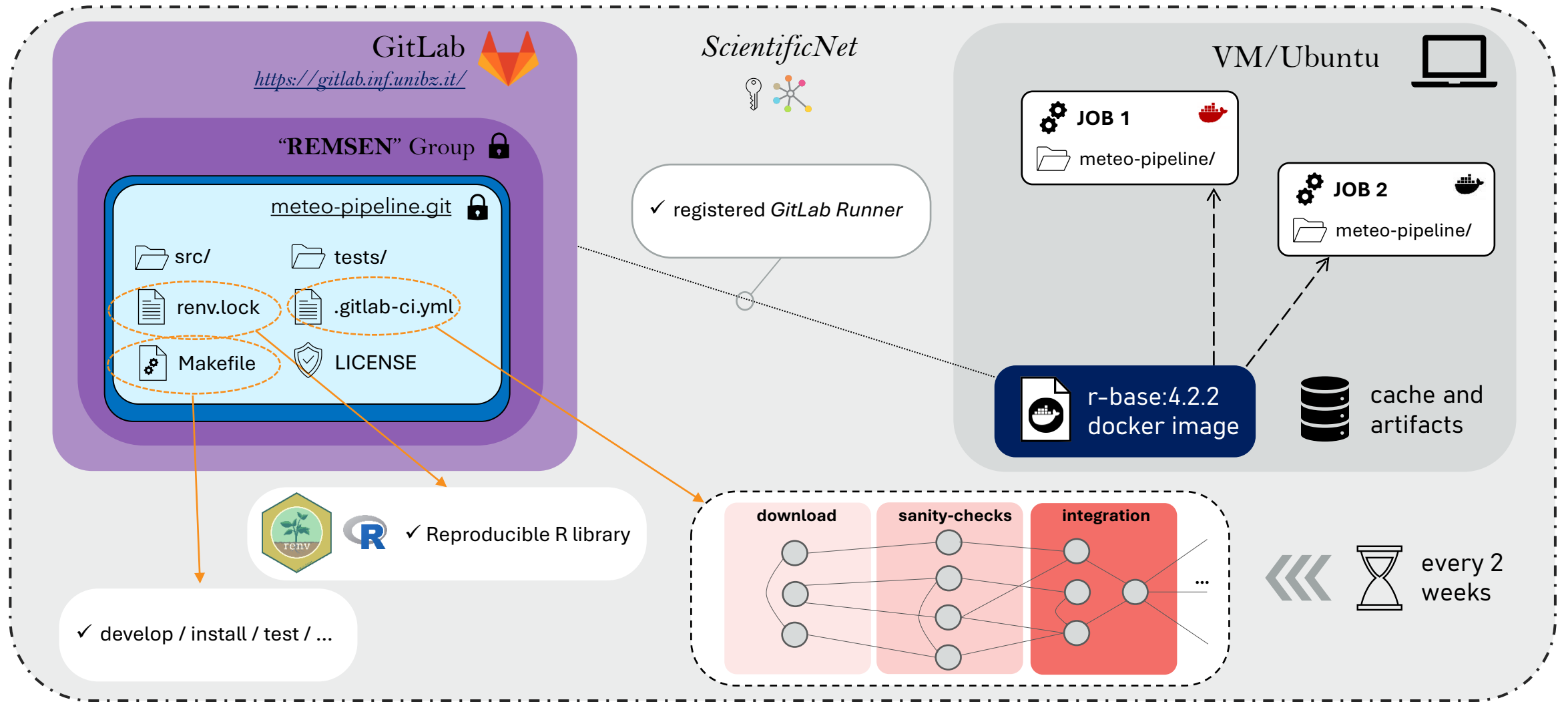




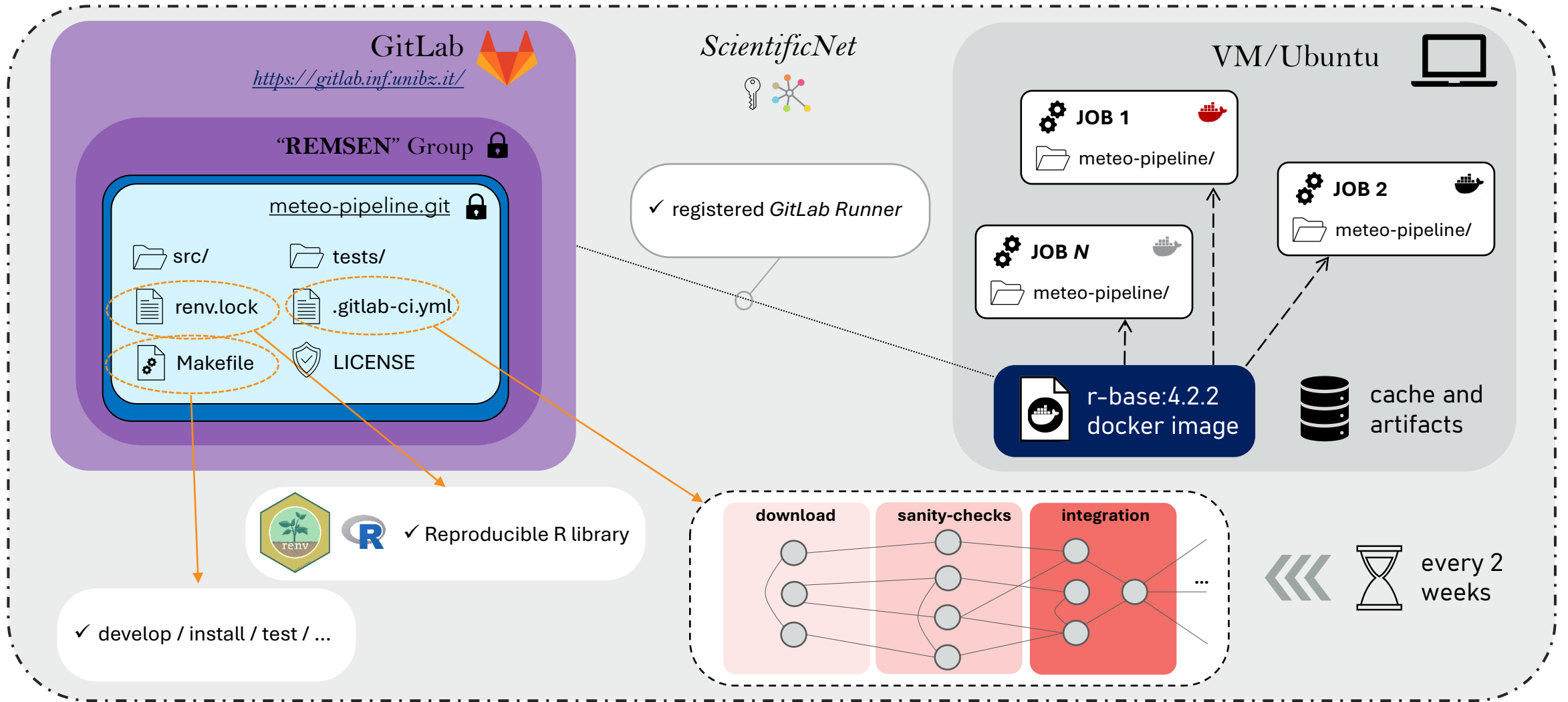
# Automating the pipeline



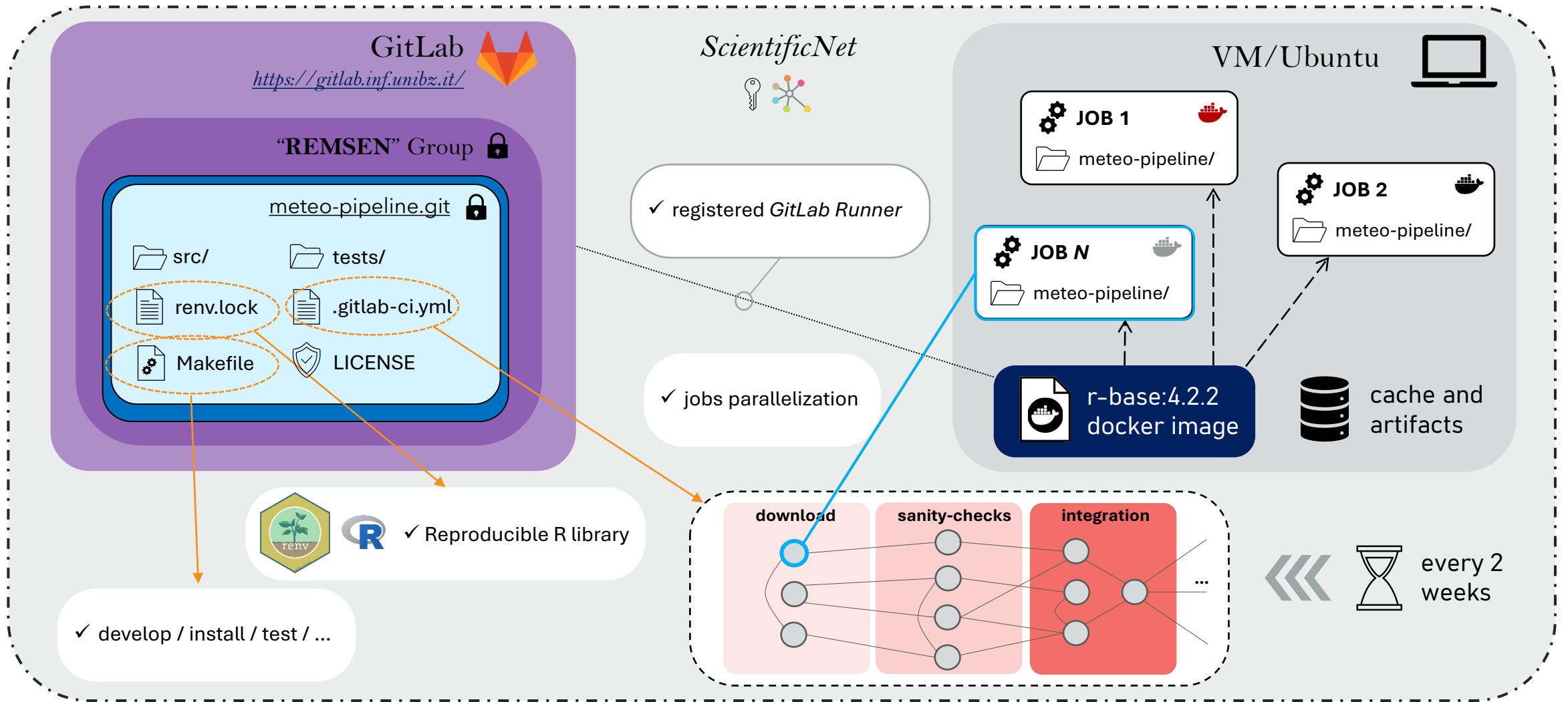
# Automating the pipeline



# Automating the pipeline















# Automating the pipeline



# GitLab's out of the box: Monitoring & Troubleshooting

# GitLab's out of the box: Monitoring & Troubleshooting

- ✓ UI-based pipeline status monitoring

Status	Pipeline	Created by	Stages
 Passed 🕒 01:07:14 🕒 1 hour ago	CI/CD: Fix missing closing quotes. #38523  main   latest		
 Failed 🕒 00:07:15 🕒 2 hours ago	CI/CD: Place echos to simulate successful... #38522  main  		

# GitLab's out of the box: Monitoring & Troubleshooting

- ✓ UI-based pipeline status monitoring
- ✓ Jobs logs visualization

The screenshot displays the GitLab CI/CD interface. At the top, a pipeline overview shows a 'Passed' status for the pipeline 'CI/CD: Fix missing closing quotes.' with a duration of 01:07:14. The pipeline is created by '6beebeac' and is currently on the 'main' branch. The pipeline consists of several stages, with the current stage highlighted in orange. Below the pipeline overview, the job logs for 'prec:netcdf-export-job' are shown. The logs indicate that the job started 1 hour ago by 'Campalani Piero' and passed. The job log shows the following steps:

- Running with gitlab-runner 16.5.0 (853330f9)
- on Meteo-pipeline runner (R 4.2.2) H2j\_kMVn, system ID: s\_3c5c6aa67698
- Preparing the "docker" executor (00:10)
- Using Docker executor with image r-base:4.2.2 ...
- Pulling docker image r-base:4.2.2 ...
- Using docker image sha256:3de1ef2039fbd2fcbaf609da402d6b2a94fee892755cc7bb074fa26b8738ce0b for r-base:4.2.2 with digest r-base:sha256:ad49725f24f2abf3f3cb8010abfd00b74d424bc47c4e3841f10e805143e5a6fa ...
- Preparing environment (00:05)
- Running on runner-h2jkmvn-project-5565-concurrent-0 via eosao49...
- Getting source from Git repository (00:05)
- Fetching changes with git depth set to 20...
- Reinitialized existing Git repository in /builds/REMSEN/cdr/climax/meteo-data-pipeline/.git/
- Checking out 6beebeac as detached HEAD (ref is main)...
- Removing install/
- Skipping Git submodules setup
- Restoring cache (00:06)
- Checking cache for main-protected...
- No URL provided, cache will not be downloaded from shared cache server. Instead a local version of cache will be extracted.

On the right side of the job logs, there is a summary of the job's performance and related information:

- Duration: 1 minute 42 seconds
- Finished: 1 hour ago
- Queued: 1 minute 41 seconds
- Timeout: 1h (from project)
- Runner: #348 (H2j\_kMVn) Meteo-pipeline runner
- Tags: meteo-pipeline r-4.2.2
- Commit: 6beebeac
- CI/CD: Fix missing closing quotes.
- Pipeline #38523 Passed for main
- gridding
- Related jobs: prec:gridding-job, prec:netcdf-export-job, tmax:gridding-job, tmax:netcdf-export-job, tmean:gridding-job

# GitLab's out of the box: Monitoring & Troubleshooting

- ✓ UI-based pipeline status monitoring
- ✓ Jobs logs visualization
- ✓ Timing information

The screenshot displays the GitLab CI/CD interface. At the top, a pipeline summary shows a 'Passed' status for the pipeline 'CI/CD: Fix missing closing quotes.' with ID #38523, created by user 6beebeac. The pipeline consists of several stages, with the current stage highlighted in orange. Below this, the job 'prec:netcdf-export-job' is shown, also 'Passed', started 1 hour ago by user Campalani Piero. The job log is visible, showing steps such as 'Preparing the "docker" executor', 'Using Docker executor with image r-base:4.2.2 ...', 'Pulling docker image r-base:4.2.2 ...', 'Preparing environment', 'Getting source from Git repository', 'Fetching changes with git depth set to 20...', 'Reinitialized existing Git repository in /builds/REMSEN/cdr/climax/meteo-data-pipeline/.git/', 'Checking out 6beebeac as detached HEAD (ref is main)...', 'Removing install/', 'Skipping Git submodules setup', and 'Restoring cache'. A metadata box on the right provides details for the job: Duration: 1 minute 42 seconds, Finished: 1 hour ago, Queued: 1 minute 41 seconds, Timeout: 1h (from project), Runner: #348 (H2j\_kMVnn) Meteo-pipeline runner, and Tags: meteo-pipeline, r-4.2.2. Below the job log, the pipeline status is shown as 'Passed' for the main branch, and a dropdown menu is set to 'gridding'. A list of related jobs is also visible, including 'prec:gridding-job', 'prec:netcdf-export-job' (highlighted with an arrow), 'tmax:gridding-job', 'tmax:netcdf-export-job', and 'tmean:gridding-job'.



# GitLab's out of the box: Monitoring & Troubleshooting

- ✓ UI-based pipeline status monitoring
- ✓ Jobs logs visualization
- ✓ Timing information
- ✓ Job's manual re-runs

The screenshot displays the GitLab CI/CD interface. At the top, a pipeline summary shows a 'Passed' status for the pipeline 'CI/CD: Fix missing closing quotes.' with ID #38523. Below this, a job log for 'prec:netcdf-export-job' is shown, detailing the execution steps from preparing the Docker executor to restoring the cache. On the right, a sidebar provides job details, including duration (1 minute 42 seconds), finish time (1 hour ago), and a 'Re-run' button (circular arrow icon) which is highlighted with an orange circle. An orange arrow points from this button to the pipeline's stage status, where the corresponding stage is also highlighted with an orange circle. The interface includes a search bar for the job log and a dropdown menu for the pipeline.

Status	Pipeline	Created by	Stages
Passed	CI/CD: Fix missing closing quotes. #38523	6beebeac	✓ ✓ ✓ ✓ ✓ ✓ ✓

```
prec:netcdf-export-job
Passed Started 1 hour ago by Campalani Piero

Search job log

1 Running with gitlab-runner 16.5.0 (853330f9)
2 on Meteo-pipeline runner (R 4.2.2) H2j_kMVn, system ID: s_3c5c6aa67698
3 Preparing the "docker" executor 00:10
4 Using Docker executor with image r-base:4.2.2 ...
5 Pulling docker image r-base:4.2.2 ...
6 Using docker image sha256:3de1ef2039fbd2fcbaf609da402d6b2a94fee892755cc7bb074fa2
6b8738ce0b for r-base:4.2.2 with digest r-base:sha256:ad49725f24f2abf3f3cb8010abf
d00b74d424bc47c4e3841f10e805143e5a6fa ...
7 Preparing environment 00:05
8 Running on runner-h2jkmvn-project-5565-concurrent-0 via eosao49...
9 Getting source from Git repository 00:05
10 Fetching changes with git depth set to 20...
11 Reinitialized existing Git repository in /builds/REMSEN/cdr/climax/meteo-data-pi
peLine/.git/
12 Checking out 6beebeac as detached HEAD (ref is main)...
13 Removing install/
14 Skipping Git submodules setup
15 Restoring cache 00:06
16 Checking cache for main-protected...
17 No URL provided, cache will not be downloaded from shared cache server. Instead
a local version of cache will be extracted.
```

Duration: 1 minute 42 seconds  
Finished: 1 hour ago  
Queued: 1 minute 41 seconds  
Timeout: 1h (from project)  
Runner: #348 (H2j\_kMVnn) Meteo-pipeline runner  
Tags: meteo-pipeline r-4.2.2

Commit 6beebeac  
CI/CD: Fix missing closing quotes.

Pipeline #38523 Passed for main  
gridding

Related jobs

- ✓ prec:gridding-job
- ✓ prec:netcdf-export-job
- ✓ tmax:gridding-job
- ✓ tmax:netcdf-export-job
- ✓ tmean:gridding-job

# GitLab's out of the box: Monitoring & Troubleshooting

- ✓ UI-based pipeline status monitoring
- ✓ Jobs logs visualization
- ✓ Timing information
- ✓ Job's manual re-runs
- ✓ API also available

The screenshot displays the GitLab CI/CD interface. At the top, a pipeline summary shows a 'Passed' status for the pipeline 'CI/CD: Fix missing closing quotes.' with a duration of 01:07:14. Below this, a job log for 'prec:netcdf-export-job' is shown, detailing the execution steps from preparing the Docker executor to restoring the cache. On the right, a sidebar provides job details, including duration (1 minute 42 seconds), finish time (1 hour ago), and a 'Re-run' button (circular arrow icon) which is highlighted with an orange circle. An orange arrow points from this button to the pipeline status bar at the top, where the specific job's status is also highlighted with an orange circle. The sidebar also shows the commit hash '6beebeac' and the pipeline ID '#38523'.

Status	Pipeline	Created by	Stages
Passed	CI/CD: Fix missing closing quotes. #38523	6beebeac	✓ ✓ ✓ ✓ ✓ ✓ ✓

```
prec:netcdf-export-job
Passed Started 1 hour ago by Campalani Piero

Search job log

1 Running with gitlab-runner 16.5.0 (853330f9)
2 on Meteo-pipeline runner (R 4.2.2) H2j_kMVn, system ID: s_3c5c6aa67698
3 Preparing the "docker" executor 00:10
4 Using Docker executor with image r-base:4.2.2 ...
5 Pulling docker image r-base:4.2.2 ...
6 Using docker image sha256:3de1ef2039fbd2fcbaf609da402d6b2a94fee892755cc7bb074fa2
6b8738ce0b for r-base:4.2.2 with digest r-base:sha256:ad49725f24f2abf3f3cb8010abf
d00b74d424bc47c4e3841f10e805143e5a6fa ...
7 Preparing environment 00:05
8 Running on runner-h2jkmvn-project-5565-concurrent-0 via eosao49...
9 Getting source from Git repository 00:05
10 Fetching changes with git depth set to 20...
11 Reinitialized existing Git repository in /builds/REMSEN/cdr/climax/meteo-data-pi
peLine/.git/
12 Checking out 6beebeac as detached HEAD (ref is main)...
13 Removing install/
14 Skipping Git submodules setup
15 Restoring cache 00:06
16 Checking cache for main-protected...
17 No URL provided, cache will not be downloaded from shared cache server. Instead
a local version of cache will be extracted.
```

Duration: 1 minute 42 seconds  
Finished: 1 hour ago  
Queued: 1 minute 41 seconds  
Timeout: 1h (from project)  
Runner: #348 (H2j\_kMVn) Meteo-pipeline runner  
Tags: meteo-pipeline r-4.2.2

Commit 6beebeac  
CI/CD: Fix missing closing quotes.

Pipeline #38523 Passed for main  
gridding

Related jobs

- ✓ prec:gridding-job
- ✓ prec:netcdf-export-job
- ✓ tmax:gridding-job
- ✓ tmax:netcdf-export-job
- ✓ tmean:gridding-job

# GitLab's out of the box: The Pipeline Editor

The screenshot displays the GitLab Pipeline Editor interface. The top navigation bar shows the project path: REMSEN > Climax > Meteo data pipeline > Pipeline Editor. The breadcrumb trail includes a home icon, a dropdown menu set to 'main', and a search bar with the text 'Search or go to...'. The main content area shows a successful pipeline run: 'Pipeline #38523 Passed for 6beebeac: CI/CD: Fix missing closing ...' with a 'View pipeline' button. Below this, a message states 'Pipeline syntax is correct. Learn more'. The 'Edit' section has three tabs: 'Visualize' (highlighted with a blue border), 'Validate', and 'Full configuration'. The pipeline visualization is divided into four stages: 'integration', 'gap-filling', 'gridding', and 'apply'. Each stage contains several jobs represented by rounded rectangles. The 'integration' stage has four jobs: 'prec:integration-job', 'tmin:integration-job', 'tmax:integration-job', and 'tmean:integration-job'. The 'gap-filling' stage has eight jobs: 'prec:gap-filling-job', 'tmin:gap-filling-job', 'tmax:gap-filling-job', 'tmean:gap-filling-job', 'prec:flags-update-j...', 'tmin:flags-update-j...', 'tmax:flags-update-j...', and 'tmean:flags-update...'. The 'gridding' stage has eight jobs: 'prec:gridding-job', 'prec:netcdf-export-...', 'tmean:gridding-job', 'tmean:netcdf-expor...', 'tmax:gridding-job', 'tmax:netcdf-export...', 'tmin:gridding-job', and 'tmin:netcdf-export-...'. The 'apply' stage has four jobs: 'cdb-export-job', 'prec:overwrite-cdb...', 'tmin:overwrite-cdb...', 'tmax:overwrite-cdb...', and 'tmean:overwrite-cd...'. The 'Pipeline editor' option in the left sidebar is highlighted with an orange box.

# GitLab's out of the box: The Pipeline Editor

The screenshot displays the GitLab Pipeline Editor interface for a project named 'REMSSEN'. The breadcrumb navigation shows the path: 'REMSSEN > Climax > Meteo data pipeline > Pipeline Editor'. The interface includes a top navigation bar with a search bar and a sidebar on the left with a 'Project' menu. The main content area shows the pipeline status as 'Pipeline #38523 Passed for 6beebeac: CI/CD: Fix missing closing ...' with a 'View pipeline' button. Below this, a message states 'Pipeline syntax is correct. Learn more'. The 'Edit' section has tabs for 'Visualize', 'Validate', and 'Full configuration', with 'Visualize' selected. The pipeline visualization is organized into four stages: 'integration', 'gap-filling', 'gridding', and 'apply'. Each stage contains several jobs represented by rounded rectangles. The 'gap-filling' stage has a job 'prec:flags-update-j...' highlighted with an orange border and a dashed orange box labeled 'job' with an arrow pointing to it. The 'integration' stage jobs are 'prec:integration-job', 'tmin:integration-job', 'tmax:integration-job', and 'tmean:integration-job'. The 'gap-filling' stage jobs are 'prec:gap-filling-job', 'tmin:gap-filling-job', 'tmax:gap-filling-job', 'tmean:gap-filling-job', 'prec:flags-update-j...', 'tmin:flags-update-j...', 'tmax:flags-update-j...', and 'tmean:flags-update-j...'. The 'gridding' stage jobs are 'prec:gridding-job', 'prec:netcdf-export-...', 'tmean:gridding-job', 'tmean:netcdf-expor...', 'tmax:gridding-job', 'tmax:netcdf-export...', 'tmin:gridding-job', and 'tmin:netcdf-export-...'. The 'apply' stage jobs are 'cdb-export-job', 'prec:overwrite-cdb...', 'tmin:overwrite-cdb...', 'tmax:overwrite-cdb...', and 'tmean:overwrite-cd...'. The 'Visualize' tab is highlighted with a blue border.

# GitLab's out of the box: The Pipeline Editor

REMSSEN > Climax > Meteo data pipeline > Pipeline Editor

main

Pipeline #38523 Passed for 6beebeac: CI/CD: Fix missing closing ...

✓ Pipeline syntax is correct. [Learn more](#)

Edit **Visualize** Validate Full configuration

**integration**

- prec:integration-job
- tmin:integration-job
- tmax:integration-job
- tmean:integration-job

**dependency**

- prec:gap-filling-job
- tmin:gap-filling-job
- tmax:gap-filling-job
- tmean:gap-filling-job
- prec:flags-update-j...**
- tmin:flags-update-j...
- tmax:flags-update-j...
- tmean:flags-update...

**gridding**

- prec:gridding-job
- prec:netcdf-export-...
- tmean:gridding-job
- tmean:netcdf-expor...
- tmax:gridding-job
- tmax:netcdf-export...
- tmin:gridding-job
- tmin:netcdf-export...

**apply**

- cdb-export-job
- prec:overwrite-cdb...
- tmin:overwrite-cdb...
- tmax:overwrite-cdb...
- tmean:overwrite-cd...

Project

- M Meteo data pipeline
- Pinned
- Issues 0
- Merge requests 0
- Manage
- Plan
- Code
- Build
- Pipelines
- Jobs
- Pipeline editor**
- Pipeline schedules
- Artifacts

# GitLab's out of the box: The Pipeline Editor

The screenshot displays the GitLab Pipeline Editor interface for a pipeline named "Meteo data pipeline". The breadcrumb navigation shows the path: REMSEN > Climax > Meteo data pipeline > Pipeline Editor. The interface includes a sidebar on the left with navigation options like "Project", "Pinned", "Issues", "Merge requests", "Manage", "Plan", "Code", "Build", "Pipelines", "Jobs", "Pipeline editor", "Pipeline schedules", and "Artifacts". The main area shows the pipeline status as "Pipeline #38523 Passed for 6beebeac: CI/CD: Fix missing closing ..." and a message "Pipeline syntax is correct. Learn more". Below this, there are tabs for "Edit", "Visualize", "Validate", and "Full configuration". The "Visualize" tab is active, showing a pipeline graph with four stages: "integration", "dependency", "gridding", and "apply". Each stage contains several jobs. The "integration" stage has jobs: prec:integration-job, tmin:integration-job, tmax:integration-job, and tmean:integration-job. The "dependency" stage has jobs: prec:gap-filling-job, tmin:gap-filling-job, tmax:gap-filling-job, tmean:gap-filling-job, prec:flags-update-j..., tmin:flags-update-j..., tmax:flags-update-j..., and tmean:flags-update-... The "gridding" stage has jobs: prec:gridding-job, prec:netcdf-export-..., tmean:gridding-job, tmean:netcdf-expor..., tmax:gridding-job, tmax:netcdf-export..., tmin:gridding-job, and tmin:netcdf-export-... The "apply" stage has jobs: cdb-export-job, prec:overwrite-cdb..., tmin:overwrite-cdb..., tmax:overwrite-cdb..., and tmean:overwrite-cd... Hand-drawn orange annotations highlight the "dependency" stage as a "dependency", the "prec:flags-update-j..." job as a "job", and the "apply" stage as a "stage".

# GitLab's out of the box: The Pipeline Editor

The screenshot displays the GitLab Pipeline Editor interface for a pipeline named "Climax" in the "Meteo data pipeline" project. The interface is divided into several sections:

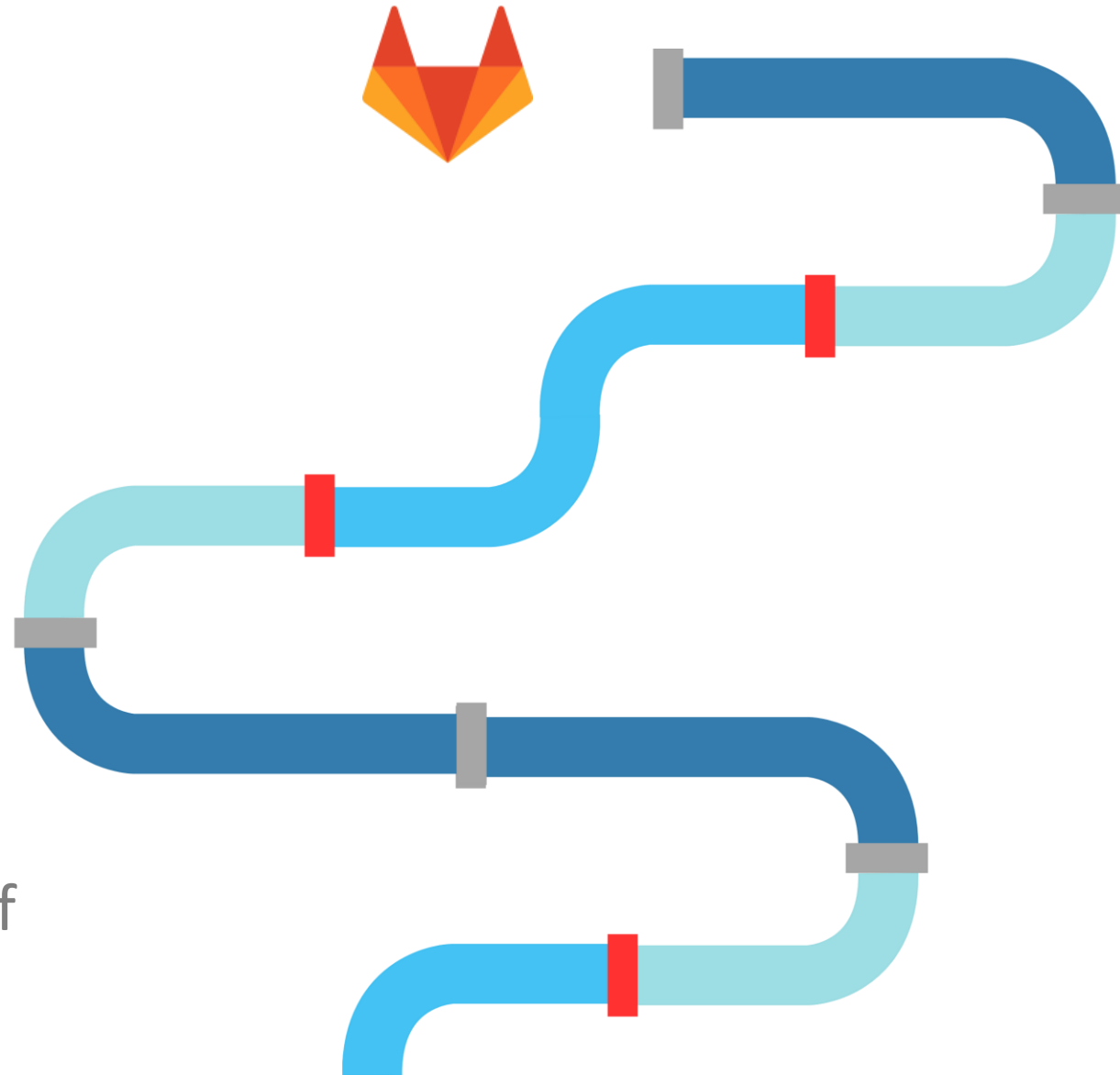
- Header:** Shows the project path "REMSSEN > Climax > Meteo data pipeline > Pipeline Editor" and the current branch "main".
- Status Bar:** Indicates "Pipeline #38523 Passed for 6beebeac: CI/CD: Fix missing closing ..." with a series of green checkmarks and a "View pipeline" button.
- Syntax Check:** A message states "Pipeline syntax is correct. [Learn more](#)", with an orange arrow pointing to it from the annotation "real-time syntax validation".
- Navigation:** Includes "Edit", "Visualize" (highlighted with a blue box), "Validate", and "Full configuration" tabs.
- Job List:** A grid of job buttons is shown, organized into stages:
  - integration stage:** Contains jobs like "prec:integration-job", "tmin:integration-job", "tmax:integration-job", and "tmean:integration-job".
  - dependency stage:** A dashed orange box labeled "dependency" encompasses a set of jobs including "prec:gap-filling-job", "tmin:gap-filling-job", "tmax:gap-filling-job", "tmean:gap-filling-job", "prec:flags-update-j...", "tmin:flags-update-j...", "tmax:flags-update-j...", and "tmean:flags-update...". An orange arrow points from this box to the "prec:flags-update-j..." job, which is also highlighted with an orange circle and labeled "job".
  - gridding stage:** Contains jobs like "prec:gridding-job", "prec:netcdf-export-...", "tmean:gridding-job", "tmean:netcdf-expor...", "tmax:gridding-job", "tmax:netcdf-export...", "tmin:gridding-job", and "tmin:netcdf-export-...".
  - apply stage:** A dashed orange box labeled "stage" encompasses a set of jobs including "cdb-export-job", "prec:overwrite-cdb...", "tmin:overwrite-cdb...", "tmax:overwrite-cdb...", and "tmean:overwrite-cd...".

# What's next



# What's next

- Finalized implementation of the GitLab pipeline
- Automating the connectors with online catalogues
- Set up of webhooks for reporting to MS Teams channel(s)
- Implementation of a monitoring dashboard with GIS-based visualization of the data store and of near-real time updated fields



# Thank you!



Contact us:

✉ [alice.crespi@eurac.edu](mailto:alice.crespi@eurac.edu)

✉ [elena.maines@eurac.edu](mailto:elena.maines@eurac.edu)

✉ [piero.campalani@eurac.edu](mailto:piero.campalani@eurac.edu)