18th November 2025 – DOAG 2025 Conference + Exhibition

Optimize the CI/CD process with SQLcl and GitLab pipelines

Automation of CI/CD for APEX applications
using SQLcl Projects and GitLab pipelines

code of change

Hyand

About me

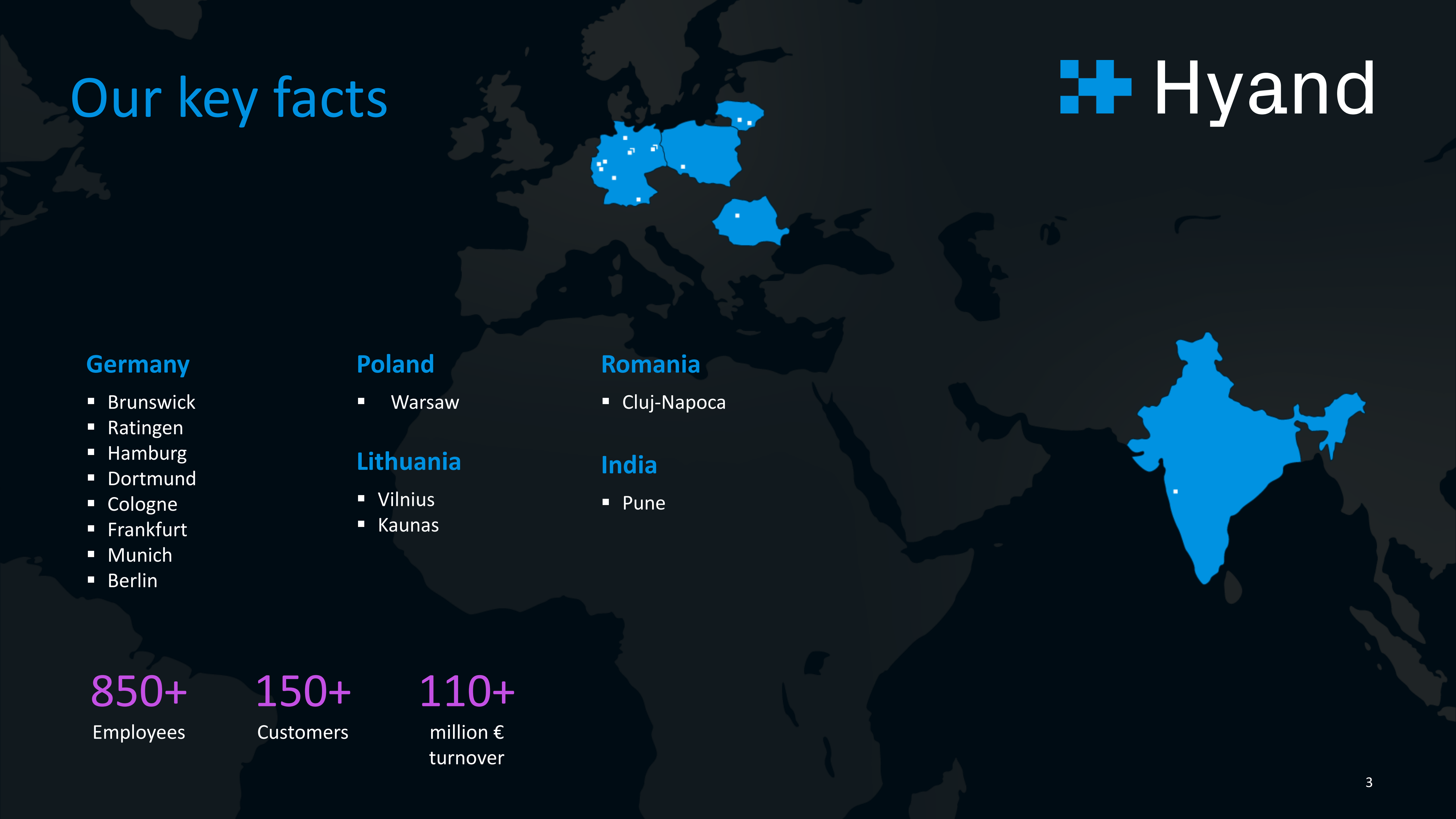
Maurice Wilhelm

Oracle APEX Consultant @Hyand

8+ years of experience with Oracle technologies

APEX enthusiast since the beginning

Focus on DevOps, CI/CD and automation



Hyand

Our key facts

Germany

- Brunswick
- Ratingen
- Hamburg
- Dortmund
- Cologne
- Frankfurt
- Munich
- Berlin

Poland

- Warsaw

Lithuania

- Vilnius
- Kaunas

Romania

- Cluj-Napoca

India

- Pune

850+
Employees

150+
Customers

110+
million €
turnover

Agenda

- Motivation
- GitLab Runner and repositories
- CI/CD variables
- Database proxy user
- Controlling the GitLab API using the APEX app
- Opportunities and challenges
- Migrating existing SQLcl LB projects
- Upgrading an SQLcl project

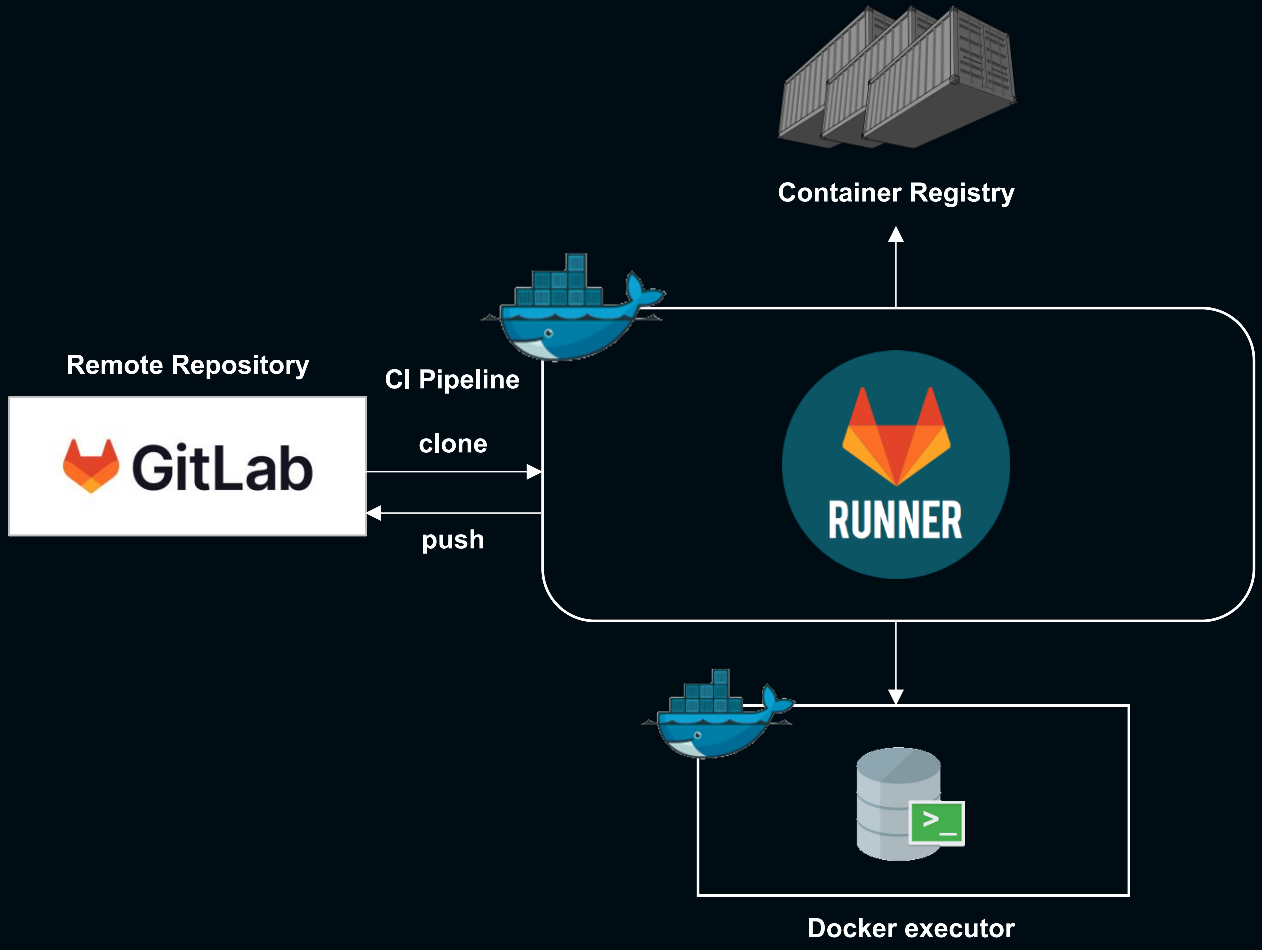
SQLcl Projects

Simplify versioning - minimize errors:
Optimize the CI/CD process with SQLcl and GitLab pipelines

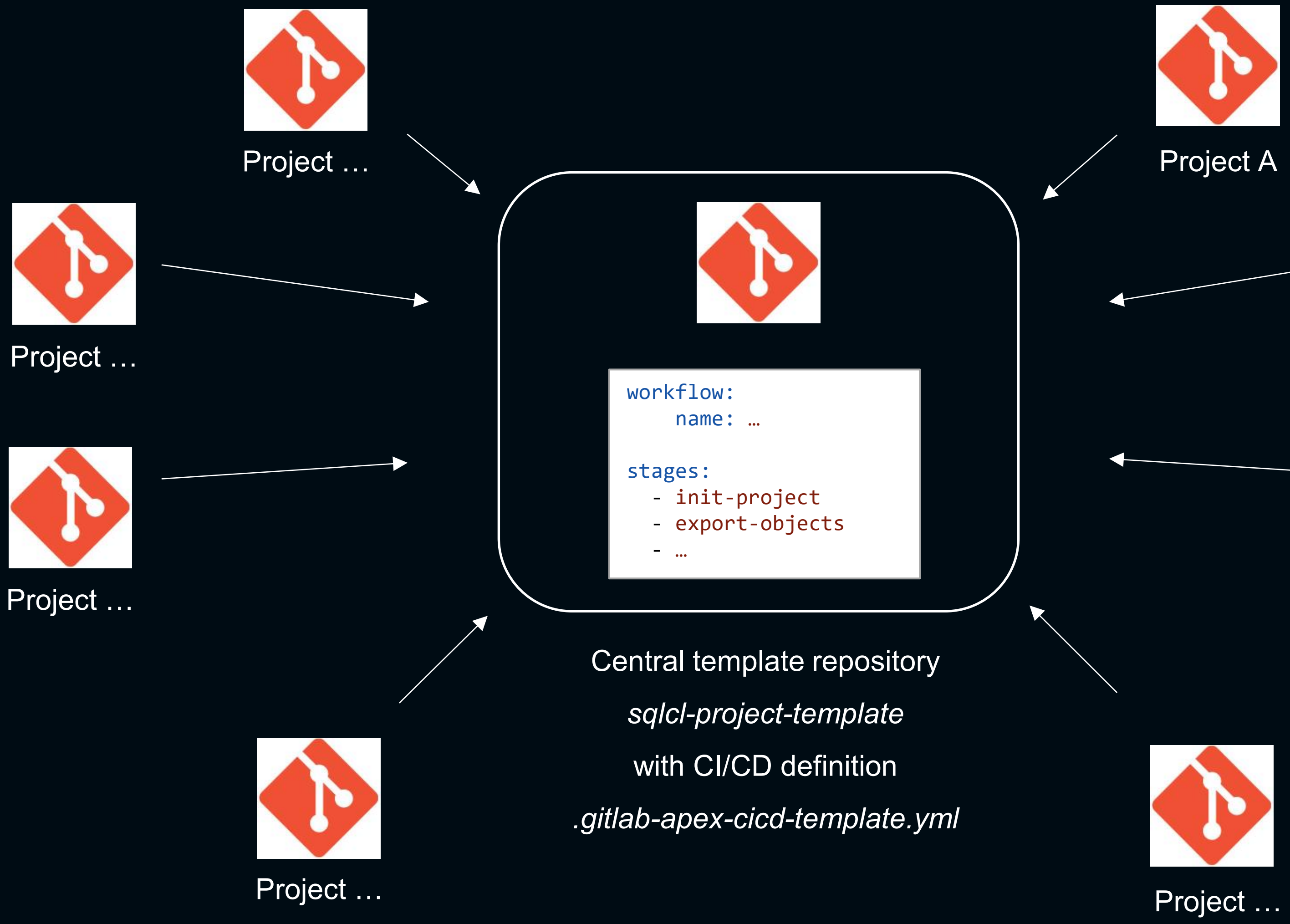
- SQLcl v24.3: New “project” command
- Export of database objects, generation of incremental changesets (Liquibase v4.25)
- Changesets: Changes in “src” are compared with “defaultBranch”, similar to “git diff”
- **Speeding up the development process through automation**
- **Minimizing errors in versioning and deployment**

CI/CD Bash script is executed using GitLab Runner

- 2 repositories
  - sqlcl-project-cicd
  - sqlcl-project-template
- GitLab pipeline
  - GitLab Runner (Docker executor)
  - Custom Docker image based on SQLcl image v25.3.0
  - Bash script cicd.sh



One template for *all* - centrally define GitLab jobs, gain structure



```
workflow:
    name: …

stages:
  - init-project
  - export-objects
  - …
```

Central template repository

sqlcl-project-template

with CI/CD definition

.gitlab-apex-cicd-template.yml

… but customizable per GitLab project

```
include:
  - project: {…}/sqlcl-project-template
    file: .gitlab-apex-cicd-template.yml
    ref: master

  - local: .gitlab-ci-custom-main.yml
    rules:
      - if: '$CI_COMMIT_BRANCH == "master'
        when: always
  - …
```

We use “include:” to access central pipeline definitions

- CI/CD template outsourced to “Template repository”
- Per GitLab project: only a few CI/CD variables and a “.gitlab-ci.yml” required
- Database connection:
  - Deployment user and EZCONNECT details definable at group level
  - Alternatively, defined per GitLab project
- Pull/push to repository via SSH
  - Private key: global CI/CD variable (group)
  - Public key: reusable as “deploy key” in multiple GitLab projects

```
include:
    project: {…}/sqlcl-project-template
    file: .gitlab-apex-cicd-template.yml
    ref: master
```

Pipeline variables are defined at the group and project level

Group/Project	Key	Type	Flag(s)	Beispiel/Syntax
Group	VAR_GLOBAL_DB_DEV_EZCONNECT_ADDRESS	Variable	Visible	db.host.tld:port/service_name
	VAR_GLOBAL_DB_DEV_DEPLOYMENT_USER	Variable	Visible	my_proxy_user
	VAR_GLOBAL_DB_DEV_DEPLOYMENT_USER_PW	Variable	Masked	my_proxy_pw
	VAR_GLOBAL_SSH_PRIVATE_KEY_GITLAB **(*)**	File	Visible	-----BEGIN RSA PRIVATE KEY----- xyz -----END RSA PRIVATE KEY-----
Project	VAR_REPO_DB_DEV_PROJECT_USER	Variable	Visible	my_dev_schema_name

…

* The public key of *VAR_GLOBAL_SSH_PRIVATE_KEY_GITLAB* must be stored/activated as a deploy key with write permissions in the GitLab project.

A proxy user can be used for database access

- Proxy user for deployments
- Available since Oracle 10g Release 2
- Minimal rights required
- Only one password needs to be stored

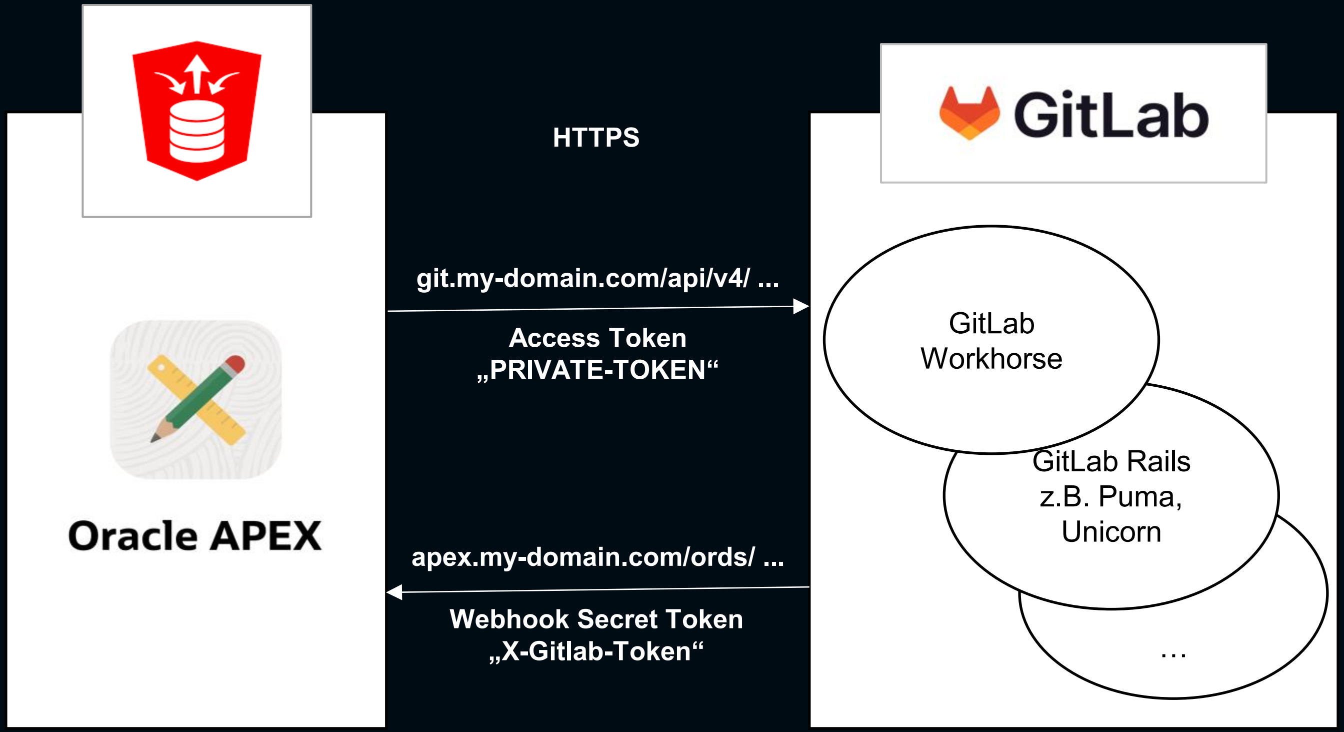
```
grant create session to {my_proxy_user};

alter user {my_dev_user} grant connect through
{my_proxy_user};

connect {my_proxy_user}[{my_dev_user}]@...
```

Pipelines can be controlled and job logs tracked using an APEX application

- APEX application for creating projects
- Start pipelines, view logs
- More convenient UI/UX
- Communication: Access tokens and webhook secret tokens


HTTPS
Oracle APEX
GitLab
git.my-domain.com/api/v4/ ...
Access Token „PRIVATE-TOKEN“
apex.my-domain.com/ords/ ...
Webhook Secret Token „X-Gitlab-Token“
GitLab Workhorse
GitLab Rails z.B. Puma, Unicorn
…

SQLcl Projects enables automated CI/CD workflow

Advantages

- Objects can be exported dynamically (DBMS_METADATA.GET_DDL)
- Automatic generation of Liquibase changesets
- Reduces manual effort for CI/CD
- Universally applicable

Disadvantages

- Filtering: only “where” clauses, no transformations
- When used locally: developers must be familiar with the “project” command (+LB)
- Liquibase deployments: “Move forward”
- Schema references
- Export of APEX metadata only as “complete app”

Existing SQLcl Liquibase projects can be migrated

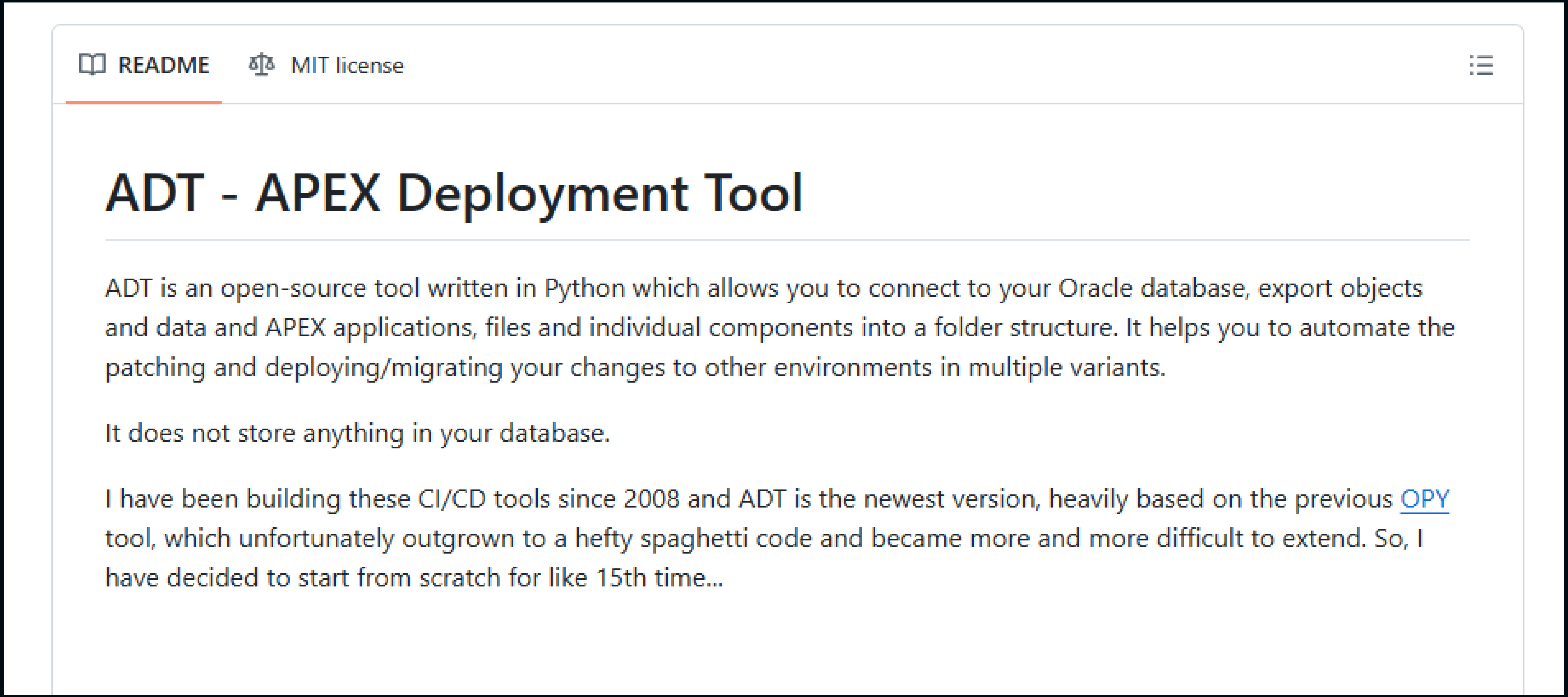
- Oracle: Provide help
- Method 1: Switch between releases
- Method 2: Migrate all changesets:

1. Create branch(es) and execute project init, export, stage
2. Create release via command
3. Adjust dist/install.sql: Change “liquibase update” to “liquibase changelog-sync” (or execute manually)
4. Remove old Liquibase objects in target database (optional)
5. Execute deployment via command

An SQLcl upgrade is easy to perform

1. Create a new branch
2. Change sqlcl->version in .dbtools/project.config.json
3. Execute “project export” – **without** staging
4. Merge branch back into “development”

There are alternatives from the Oracle community to SQLcl Projects



README MIT license

ADT - APEX Deployment Tool

ADT is an open-source tool written in Python which allows you to connect to your Oracle database, export objects and data and APEX applications, files and individual components into a folder structure. It helps you to automate the patching and deploying/migrating your changes to other environments in multiple variants.

It does not store anything in your database.

I have been building these CI/CD tools since 2008 and ADT is the newest version, heavily based on the previous OPY tool, which unfortunately outgrown to a hefty spaghetti code and became more and more difficult to extend. So, I have decided to start from scratch for like 15th time...

GitHub - jkvetina/ADT: APEX Deployment Tool @Jan Květina

Live Demo

Image sources

https://practicalbytes.de/managed-gitlab/ (last accessed on 15th November 2025)

https://oneclick-cloud.com/de/blog/trends/docker/ (last accessed on 15th November 2025)

https://gitlab.com/gitlab-community/gitlab-org/gitlab-runner (last accessed on 15th November 2025)

https://www.oracle.com/database/technologies/appdev/sqlcl/sqlcl-faq.html (last accessed on 15th November 2025)

https://pixabay.com/de/vectors/container-versand-fracht-logistik-147973 (last accessed on 15th November 2025)

https://git-scm.com/community/logos (last accessed on 15th November 2025)

https://www.oappsnet.com/oracle-forms-migration-to-apex (last accessed on 15th November 2025)

https://www.oracle.com/de/database/technologies/appdev/rest.html (last accessed on 15th November 2025)

https://github.com/jkvetina/ADT (last accessed on 15th November 2025)

Thank you very much for your attention!

Say Hy_

+49 (0) 2102 30 961-0
maurice.wilhelm@hyand.com

https://www.linkedin.com/in/maurice-wilhelm-511570272
https://blog.maurice-wilhelm.de