# The current state of gCTS and
# How it could improve your CI-processes for ABAP

Ulrich Auer, Karin Spiegel, SAP
November 2, 2021

PUBLIC

THE BEST RUN SAP

# Disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, this presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or any related document, or to develop or release any functionality mentioned therein.

This presentation, or any related document and SAP's strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP's intentional or gross negligence.

All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

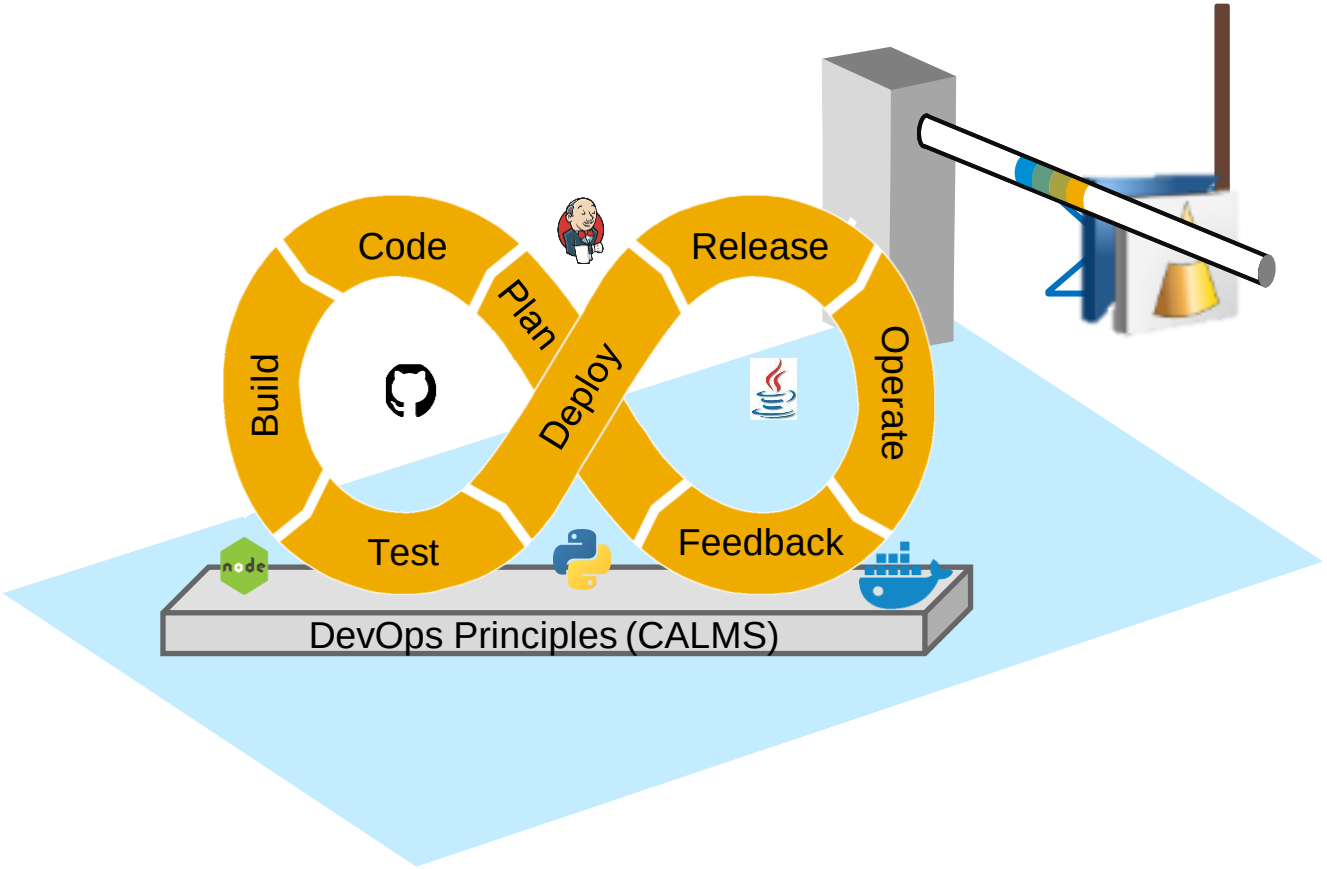
# A short recap

# When and why DevOps with ABAP
## DevOps – get ABAP in

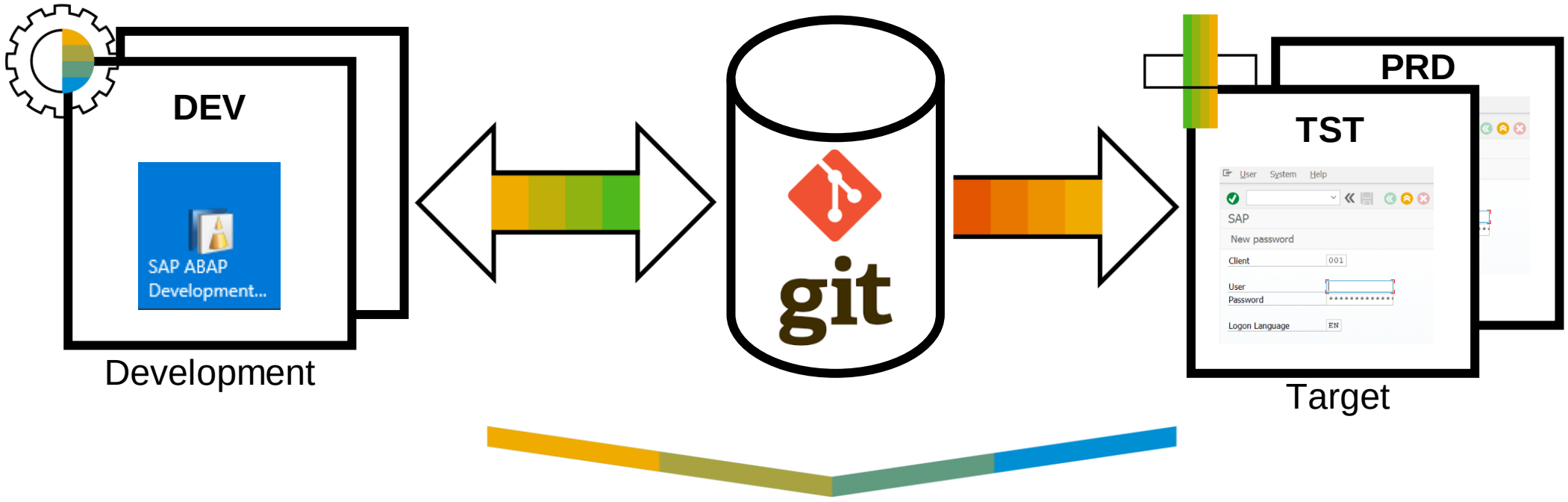
It should be possible to add ABAP to the DevOps world

- To share source code on Git
- To enable automation of development processes via pipelines
- To manage ABAP development similar to what you do in other languages and environments

→ This is what Git-enabled CTS (gCTS) aims at



# Our idea…





* Change and Transport System

# Why today

The gCTS registry to help you manage mainly your customizing via gCTS

The collaborator concept to align access handling to repositories in ABAP and git

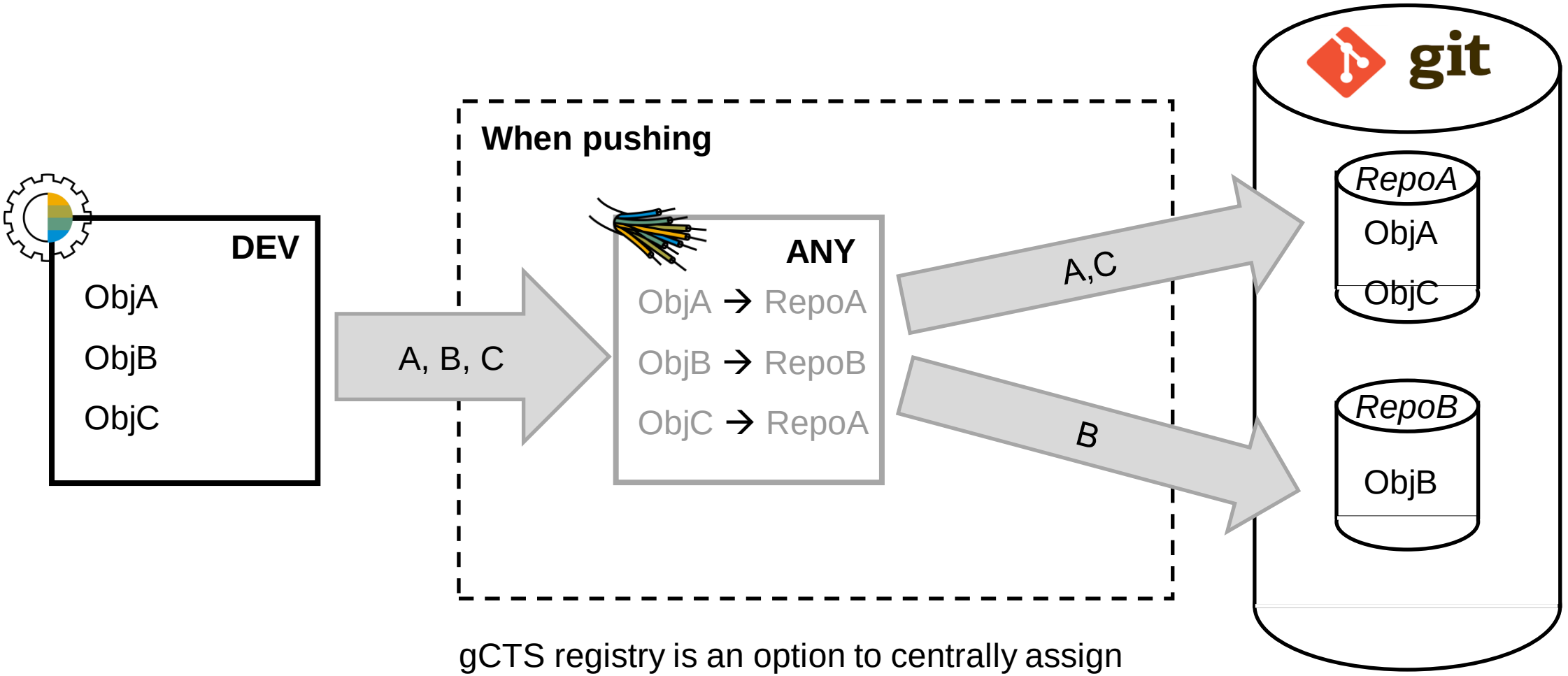
The changes in Project ‘Piper’ to make steps Jenkins-independent, provide more options and remove bugs

Changes to the merge process, conflict resolution and new deploy option to allow more CI like processes

# This presentation is about SAP S/4HANA 2021. Features shown might not be available in other releases or might look different.

# The registry

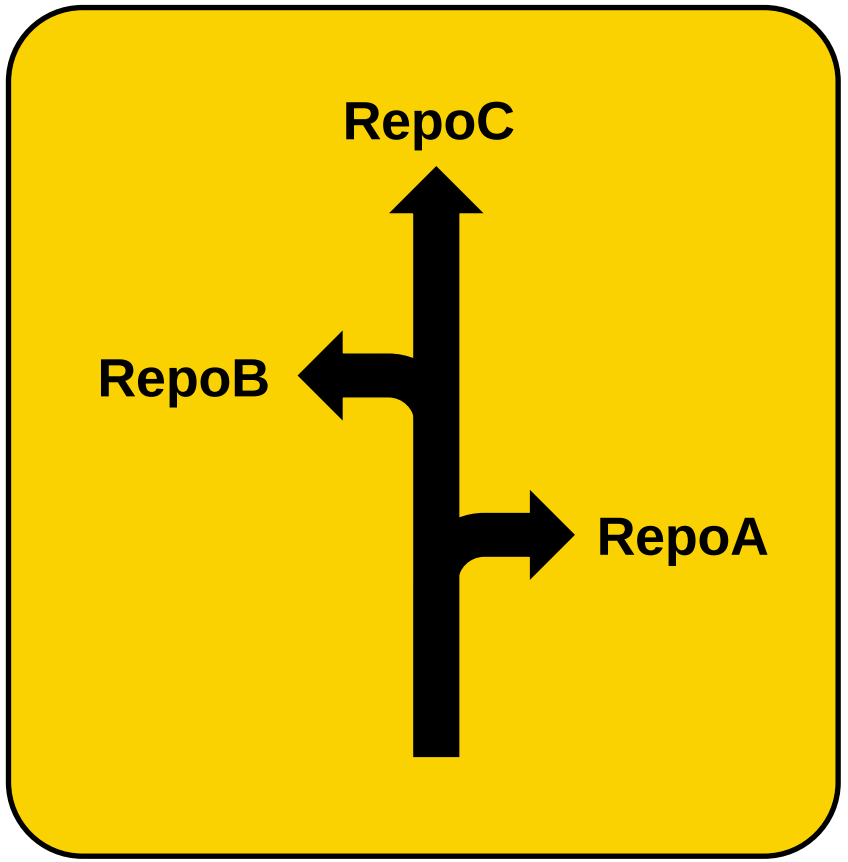
# Introduction to the registry
## The idea of the registry




gCTS registry is an option to centrally assign objects and packages to repositories

# Introduction to the registry
## Facts about the registry

- Registry ‘knows’ to which repository a certain object shall be pushed, especially if several repositories are in place.
- One persistent ABAP system ‘hosts’ the registry
  - Should always be up and stable
  - Must *not* be overwritten (e.g. by system refresh)
- Registration can be done from any connected system
- Registry is recommended when customizing is managed via gCTS
- Requires implementing a BAdI to get integrated in development process



# Introduction to the registry
## Registry for customizing

Customizing entities usually have no object catalog entries (TADIR)

Therefore, we recommend that you use the registry when customizing shall be handled via gCTS

- No need to think about the target for every transport request
- By default, the standard transport layer would be used for customizing if registry is not in use
- Registry makes sure that each customizing entity is stored in exactly one repository

**And how?**

- Start with one customizing repository
- Differentiate in different customizing repositories later – migration tool is planned

# Introduction to the registry
## How many repositories? Best practices

How many repositories?

- If customizing depends on coding / application
  → Use same repository for customizing and coding
- If customizing is client dependent
  → Use one repository with one branch per client
- If customizing is client independent and not part of an application
  → Use one repository for customizing
- If customizing depends on release
  → Use one repository with one branch per release and one branch per client per release

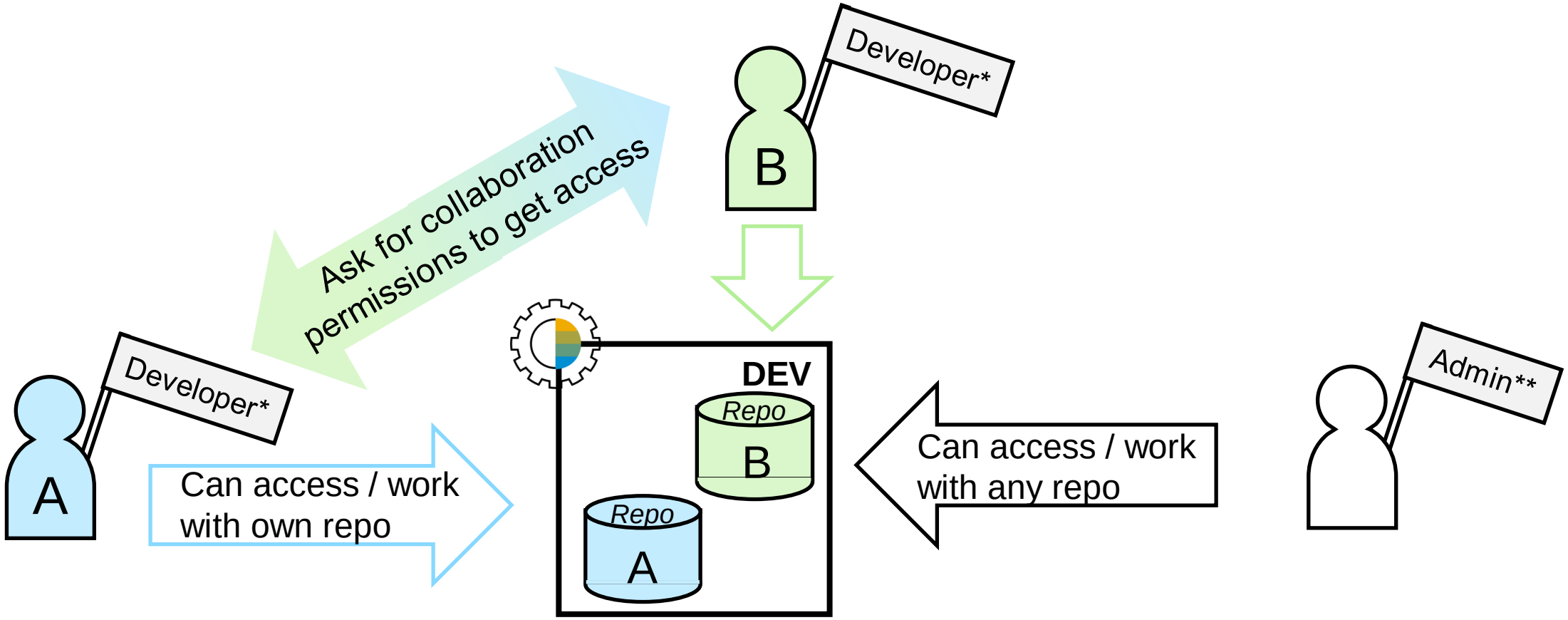
Use an appropriate naming convention for your branches

Customizing is not application data! Application data should not be pushed to git

# The collaborator concept

# Authentication and permissions in gCTS

## Collaborating in gCTS

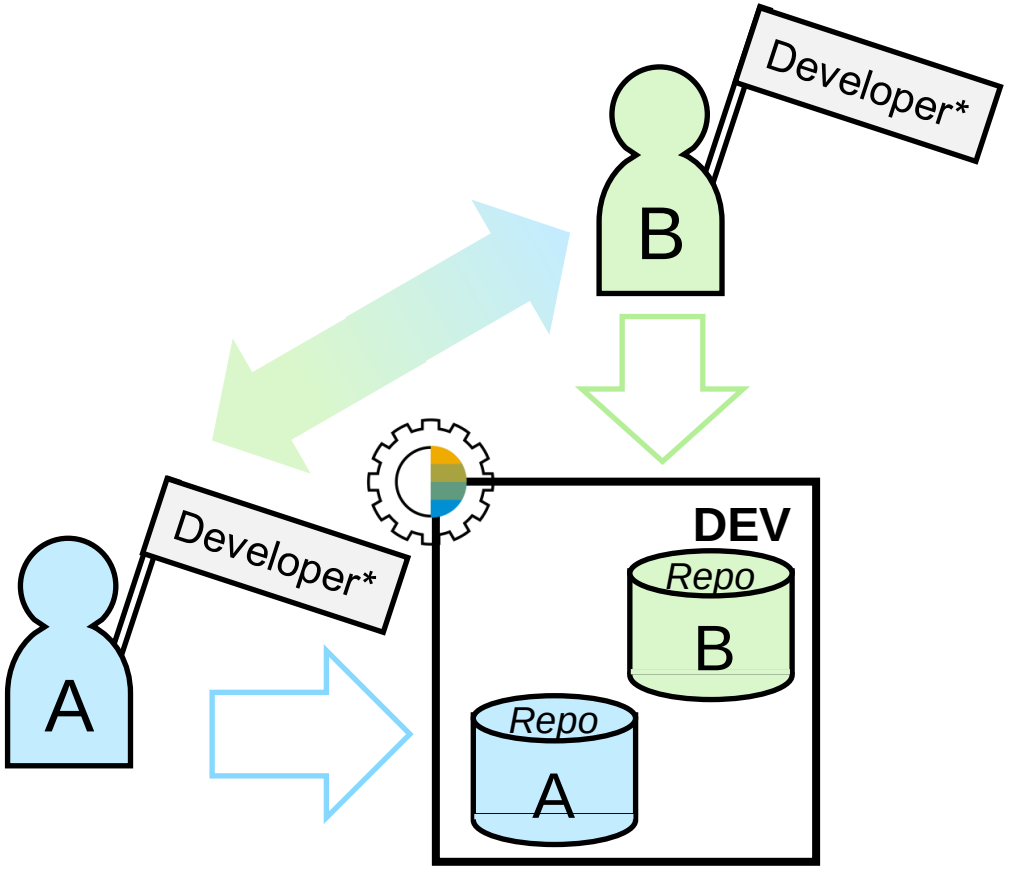



* Role: SAP_BC_GCTS_REPO_DEVELOPER

** Role: SAP_BC_GCTS_REPOSITORY_ADMIN (at least)

# Authentication and permissions in gCTS
## How collaborating works in gCTS

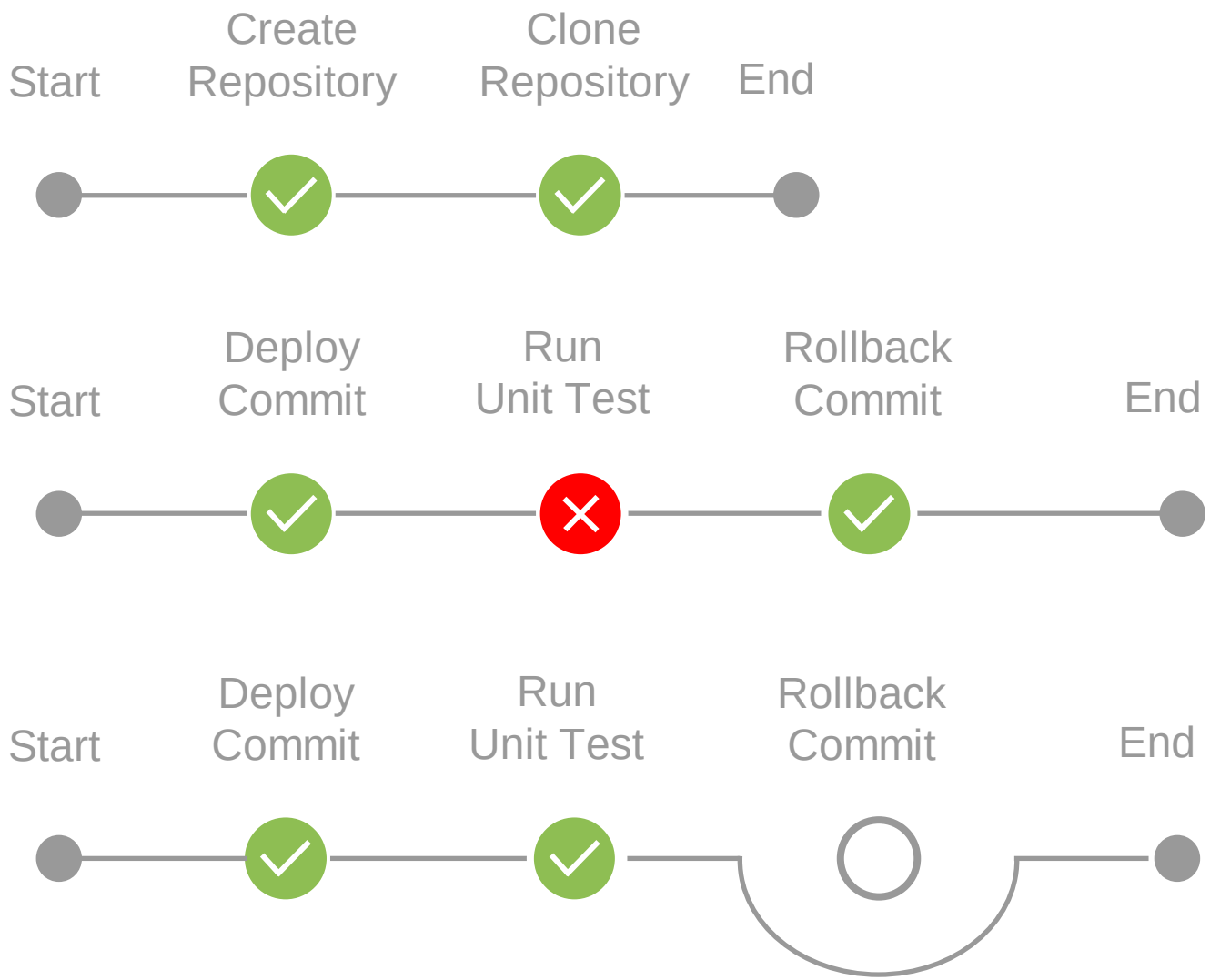
- This is about collaborating on local repositories – Git-providers have similar mechanisms, but there is no synchronization with gCTS
- You can work with teams or assign single collaborators
- Assignments are made per repository
- Teams are created centrally per system



# Demo

# The steps in Project ‘Piper’

# Project ‘Piper’ – available Steps




Library Steps in Project ‚Piper‘ – can be used with SAP S/4HANA 2020 and later

gctsCloneRepository

gctsCreateRepository

gctsDeploy

gctsExecuteABAPUnitTests

gctsRollback

# gCTSDeploy (Revised)

gCTSDeploy-Step

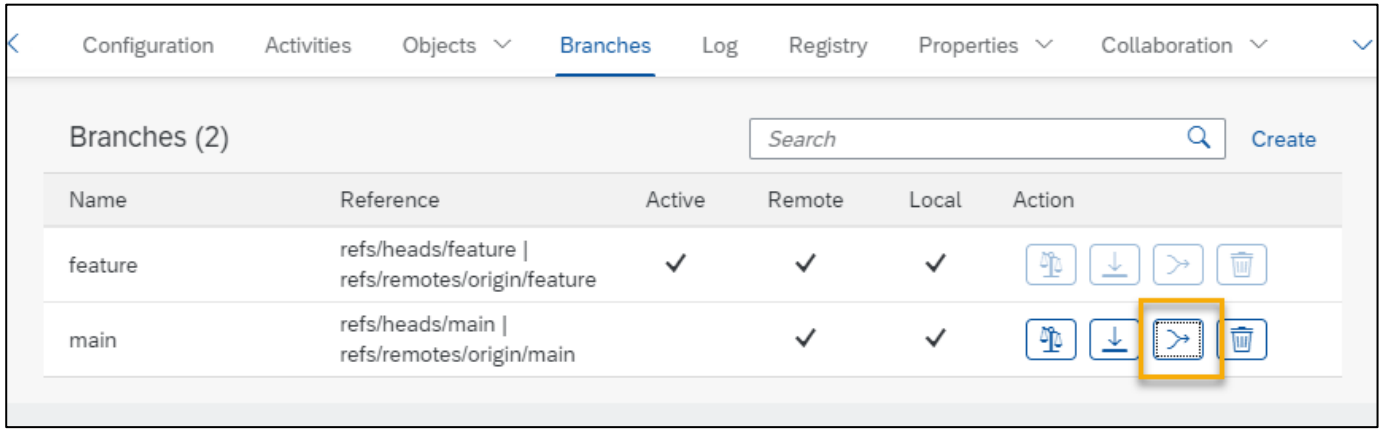
- Extended functionality
  - Creates & clones the repository if it does not exist
    → gCTSCreateRepository & gCTSCloneRepository are not needed any more
  - Can set a certain branch active
  - Can set repository parameters
  - Can set a defined commit as active (or latest)
  - Can execute a rollback
- Can be used from SAP S/4HANA 2020 onwards

# Merge process and conflict resolution

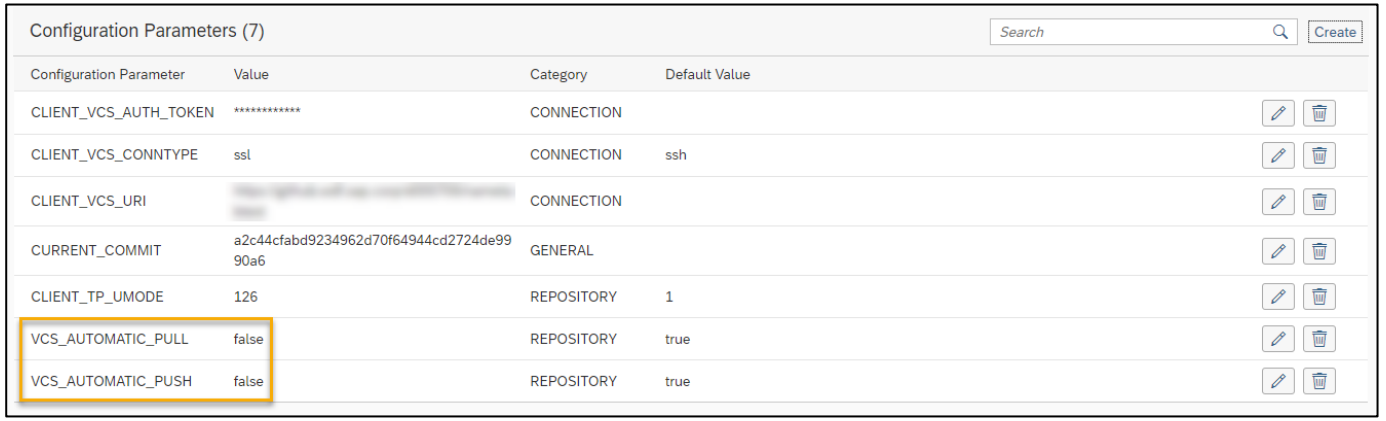
# Merge branches

## Merge branches on the local repository on the Branches tab of the gCTS app

- Merge the selected branch into the active branch



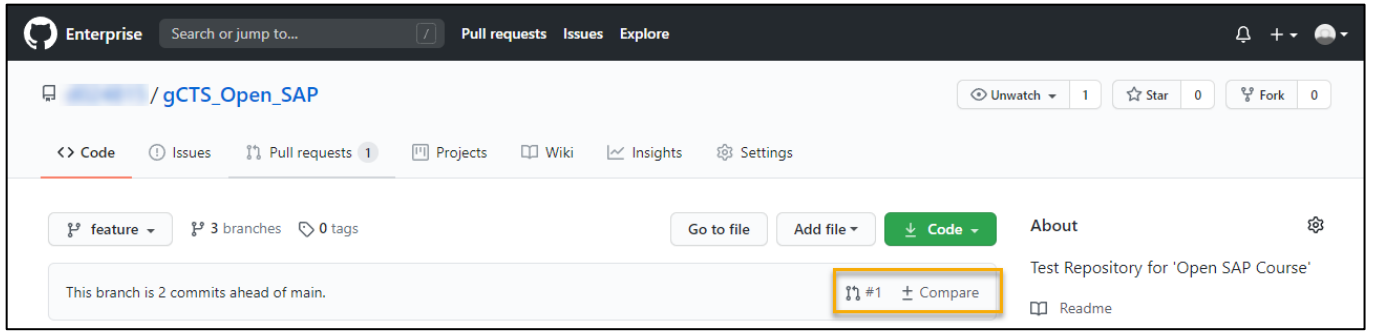

- Parameters 'VCS_AUTOMATIC_PULL' and 'VCS_AUTOMATIC_PUSH' need to be set to false



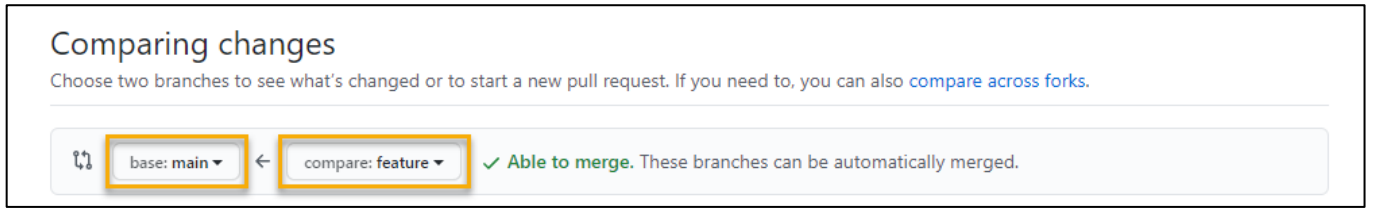
# Merge branches

## Merge branches on the remote repository by using a pull request on GitHub

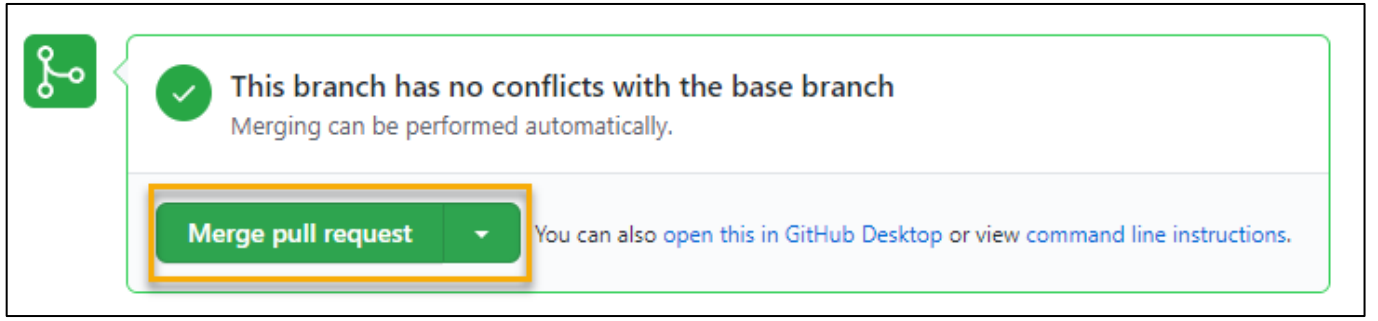
- Compare branches and create a pull request




- Select the branches to be merged into each other

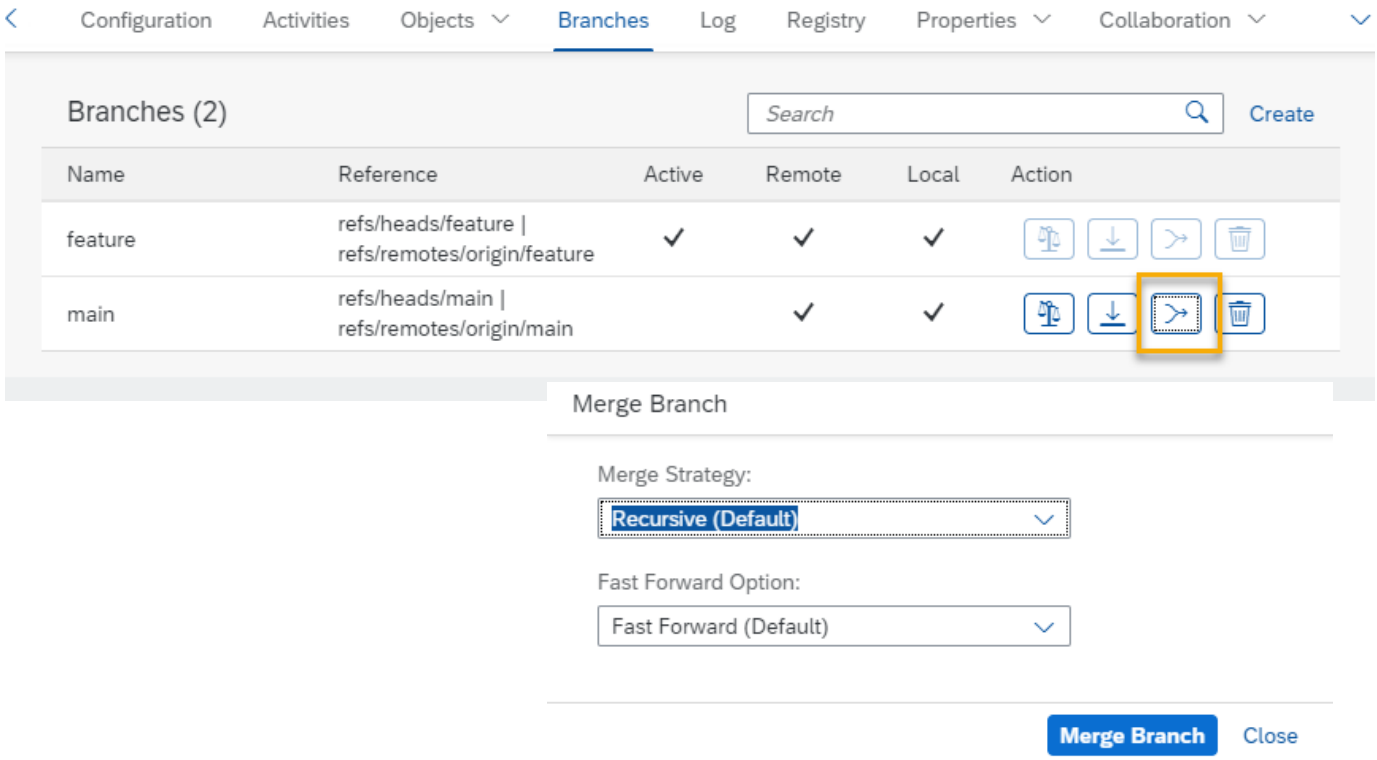



- Merge the pull request



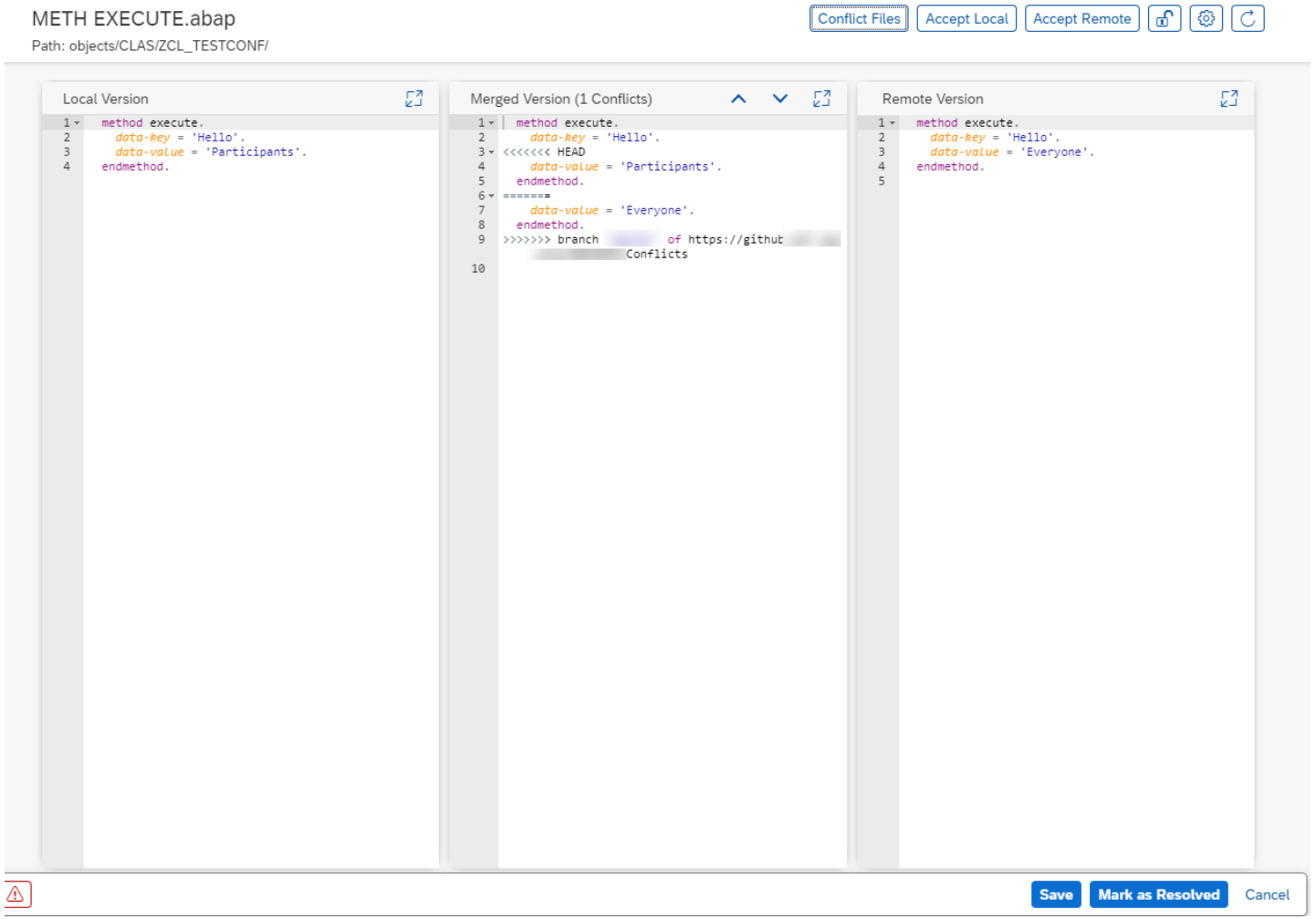
# Merge Process – Tools Involved: Merge on Branches Tab

- Merge selected branch into active
- Choose Merge Strategy and decide about Fast Forward Option – defaults should be a good start
- Parameters VCS_AUTOMATIC_PULL and VCS_AUTOMATIC_PUSH need to be set to false



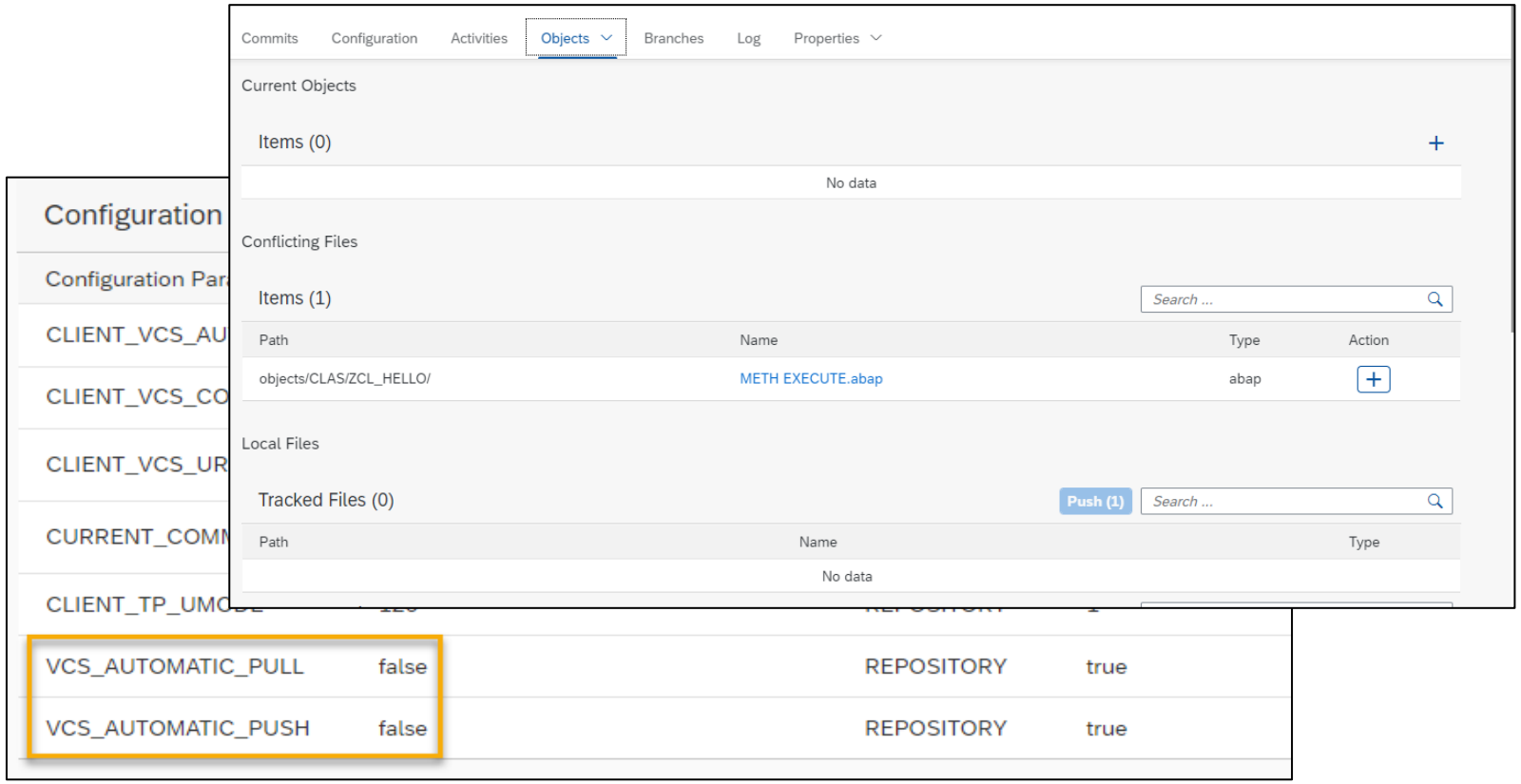
# Merge Process – Tools Involved: Conflict resolution editor

- Integrated in 'Objects' tab of the gCTS app
- Shows local and remote version
- Shows merged version indicating conflicts
- Merged version requires editing
- Uses GitHub-API to load remote version

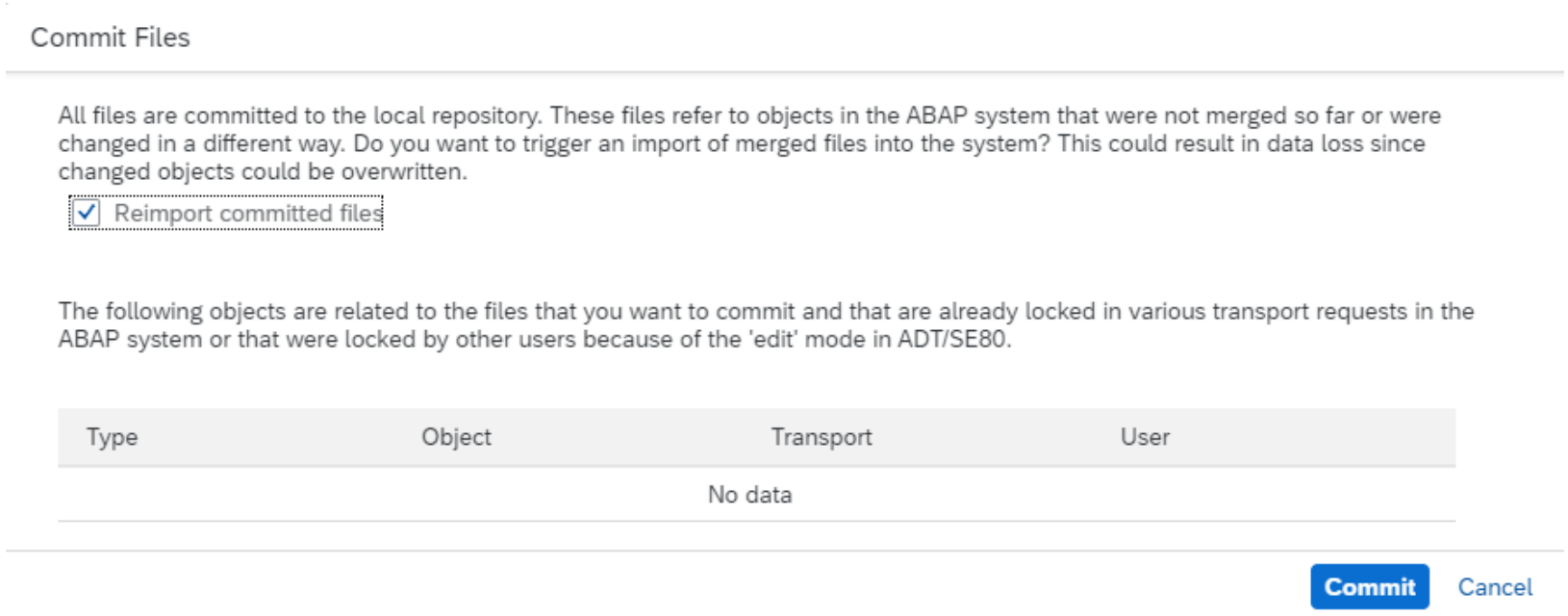


# Conflict resolution in gCTS

- Requires SAP S/4HANA 2020
- Can only happen if you stop automatic pull and push
- Is done on the 'Objects' tab of the gCTS app. From there the 'gCTS conflict resolution editor' is launched



# Conflict resolution in gCTS – enhanced Commit option



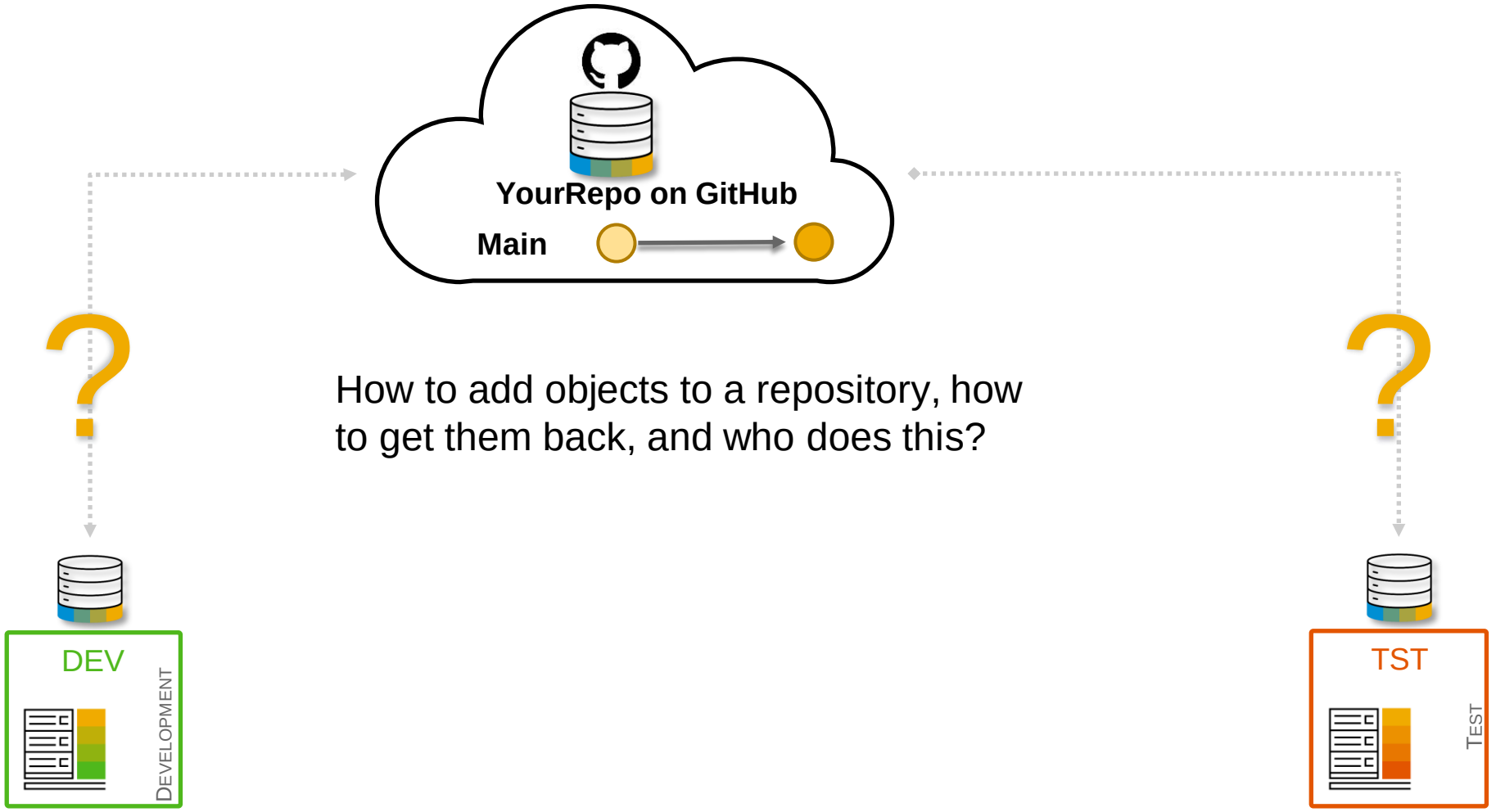

Import & Activate files into development system

- Version with resolved conflicts is available in Dev systems
- Any uncommitted changes done in Dev in the meantime are lost

# Demo

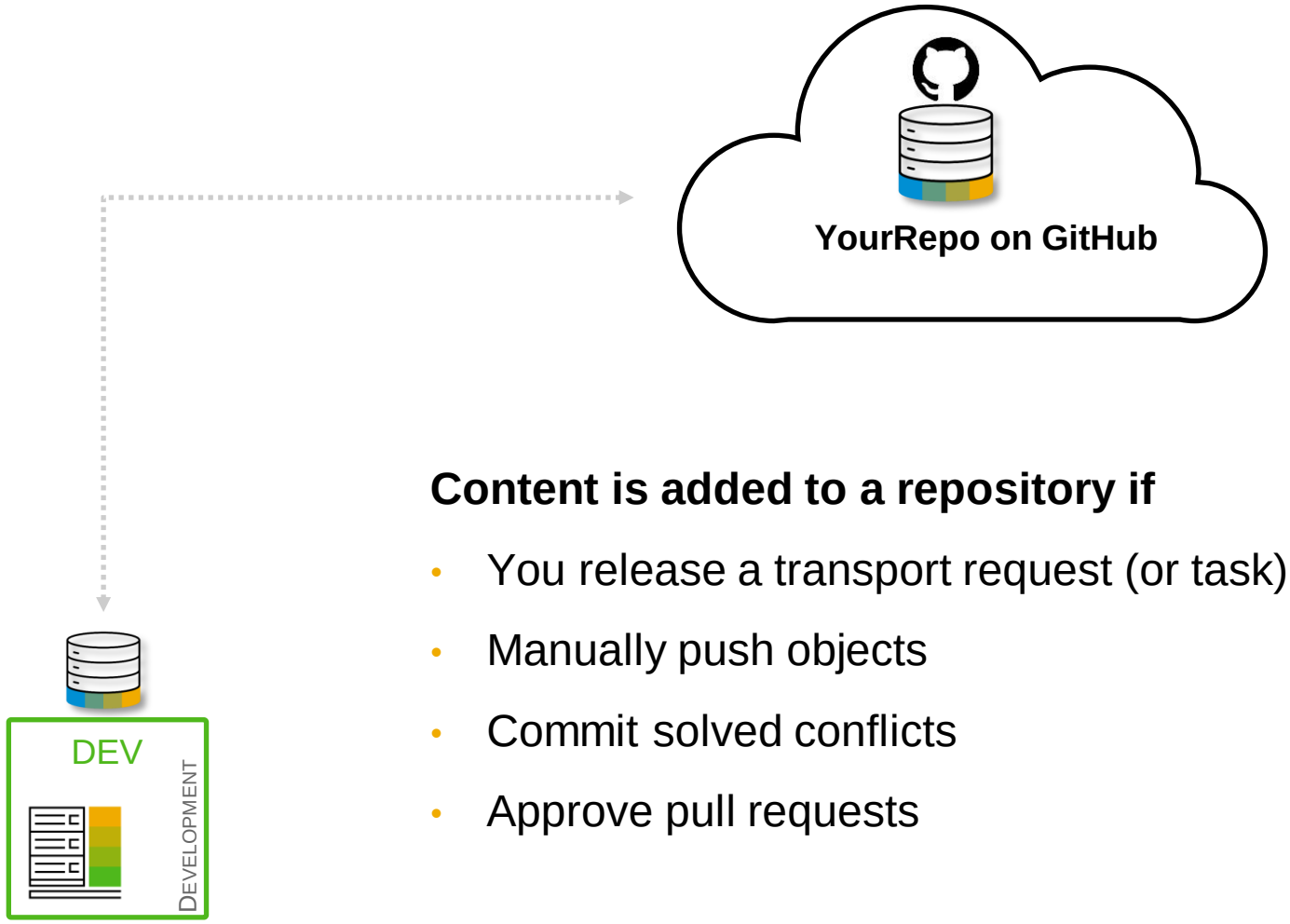
# Deployment Options

# What this is about




How to add objects to a repository, how to get them back, and who does this?

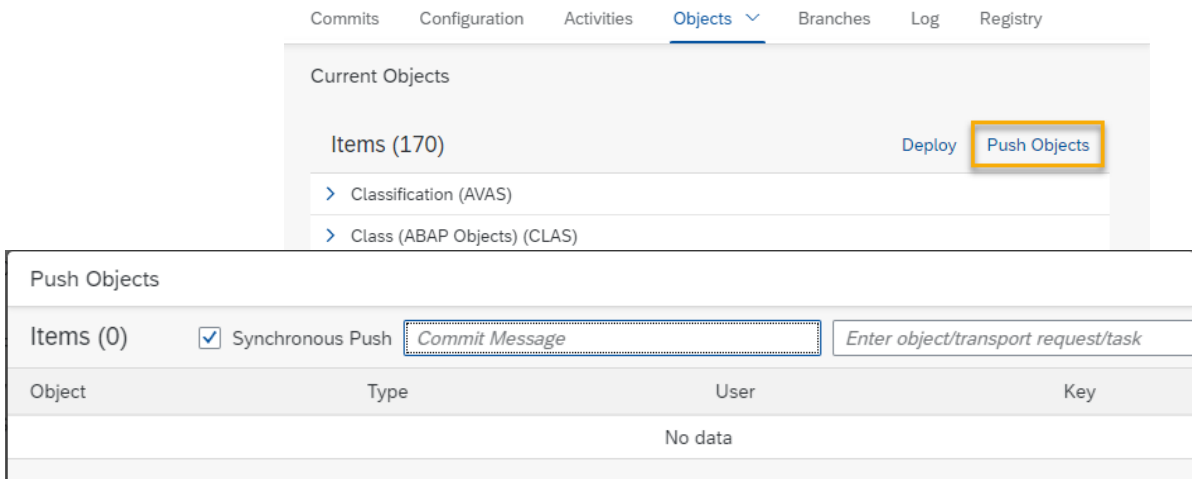
# Fill your repository

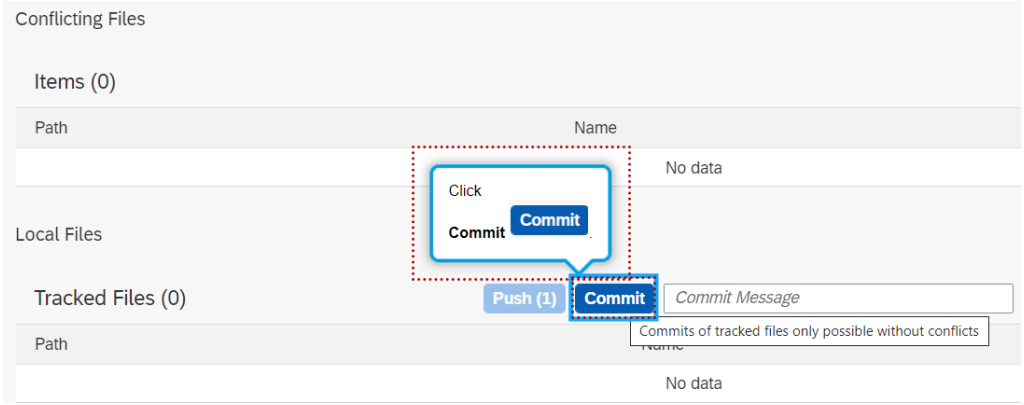



| **Content is added to a repository if** | Require special action to add back into ABAP Runtime |
| --- | --- |
| • You release a transport request (or task) | |
| • Manually push objects | |
| • Commit solved conflicts | X |
| • Approve pull requests | X |

# Initiate pushes in gCTS App

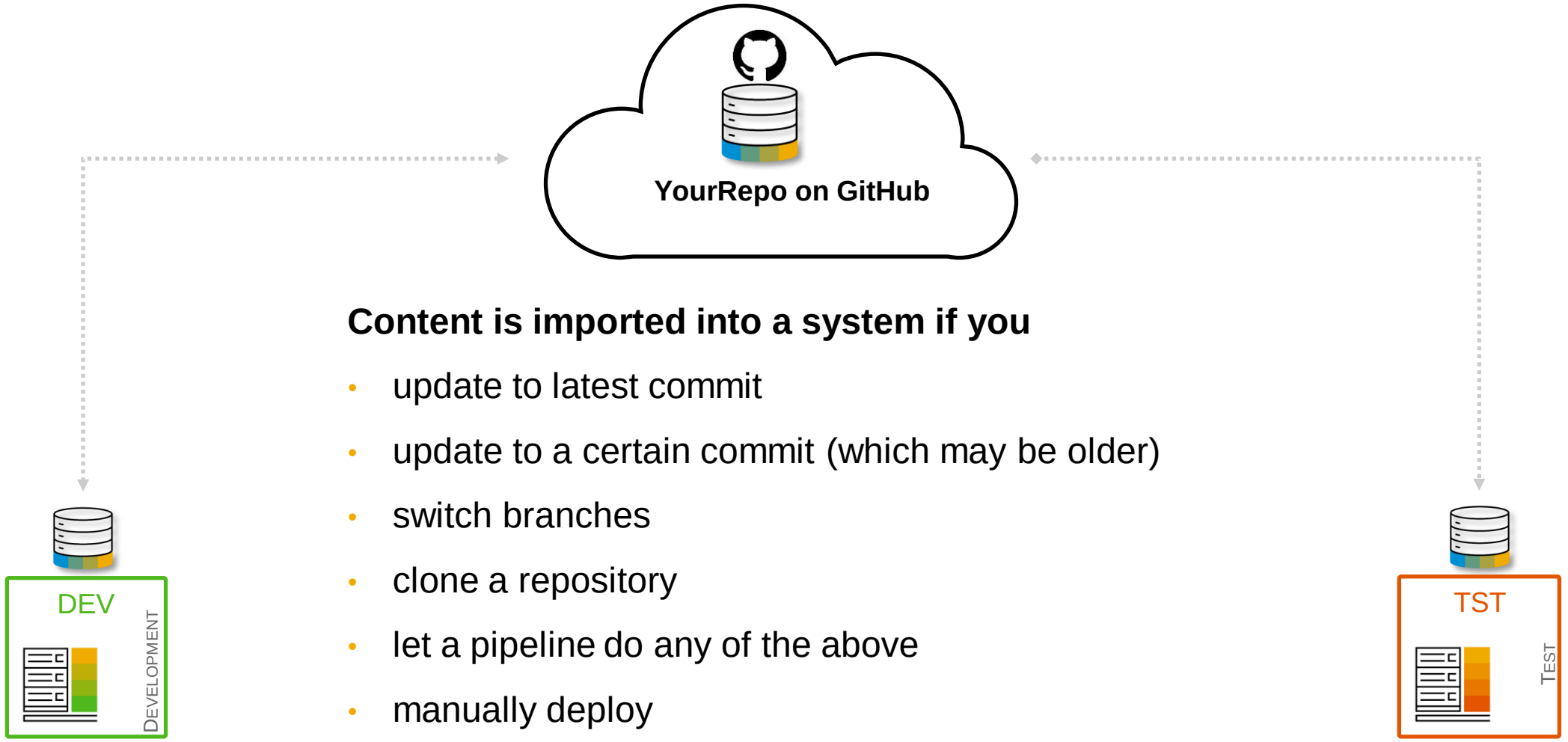
On the objects tab, you can push individual objects, packages or content of transport requests




When you solved a conflict, you need to push and commit the changed objects

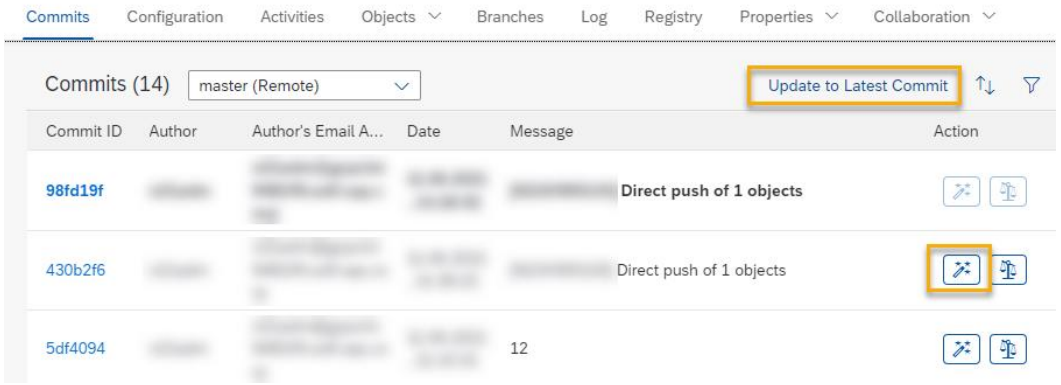
# Pull objects to a target system




**Content is imported into a system if you**

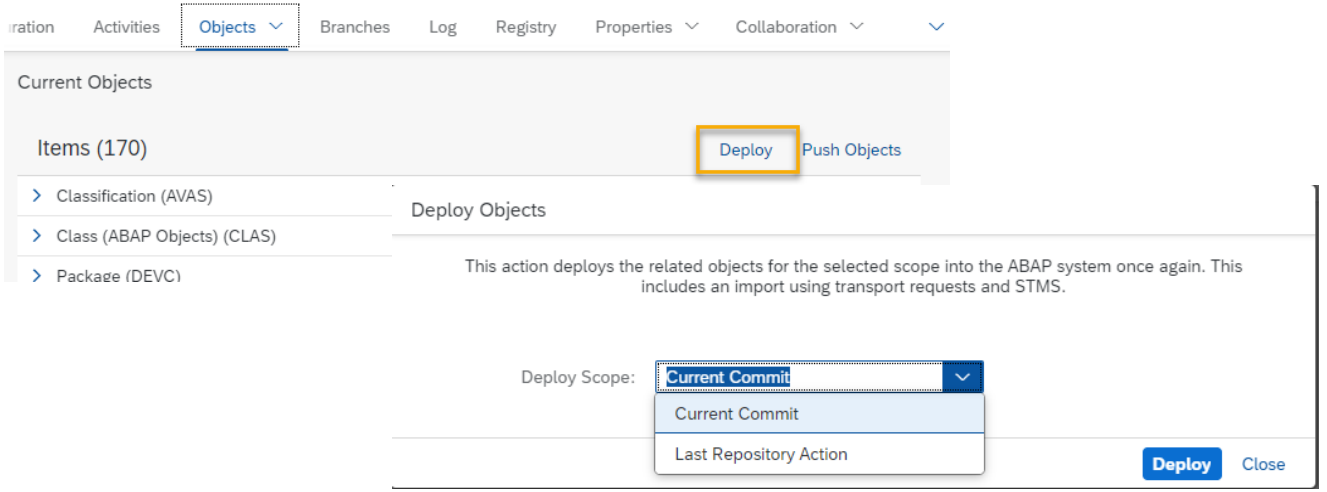
- update to latest commit
- update to a certain commit (which may be older)
- switch branches
- clone a repository
- let a pipeline do any of the above
- manually deploy

# Initiate deployment in gCTS App




On the Commits tab, you can switch between commits or update to latest – this will initiate import of transport requests.

Initiate deployment on objects tab: Can deploy either the current commit or re-do the last repository action Can e.g. be used to import customizing in different clients



# Thank you.

Contact information:

**Karin Spiegel**
Product Management
karin.spiegel@sap.com

Follow us

**www.sap.com/contactsap**



THE BEST RUN SAP®