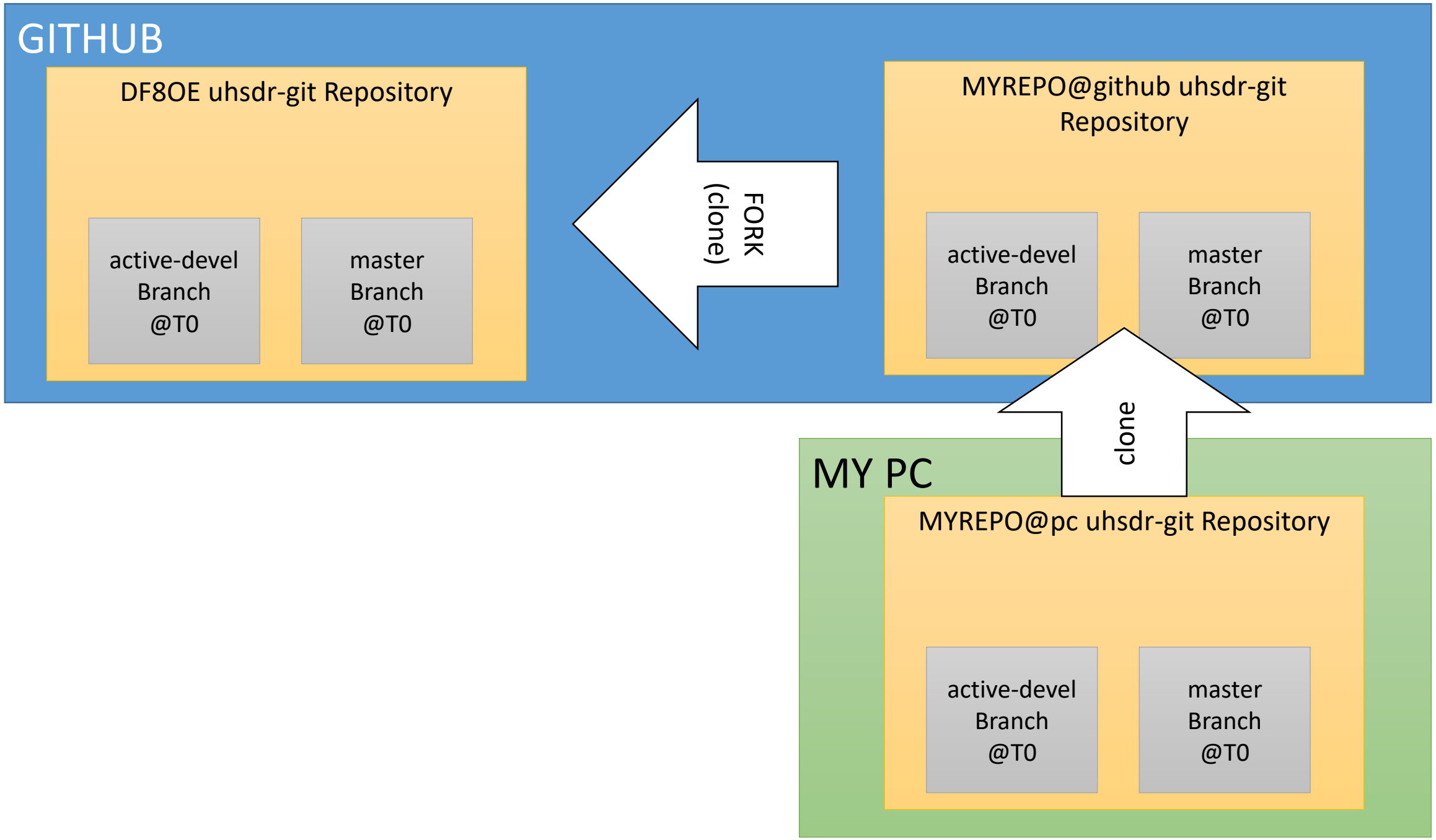
3 (!) Repositories make up a proper github workflow

1 Repository you want to contribute to at github (e.g. DF8OE)

2 Repositories you need for your work (1 at github, forked, 1 at your pc to make the changes.)

Your repository at github is used to provide data to others. You will not work directly in this repository (mostly you will just push commits to it)

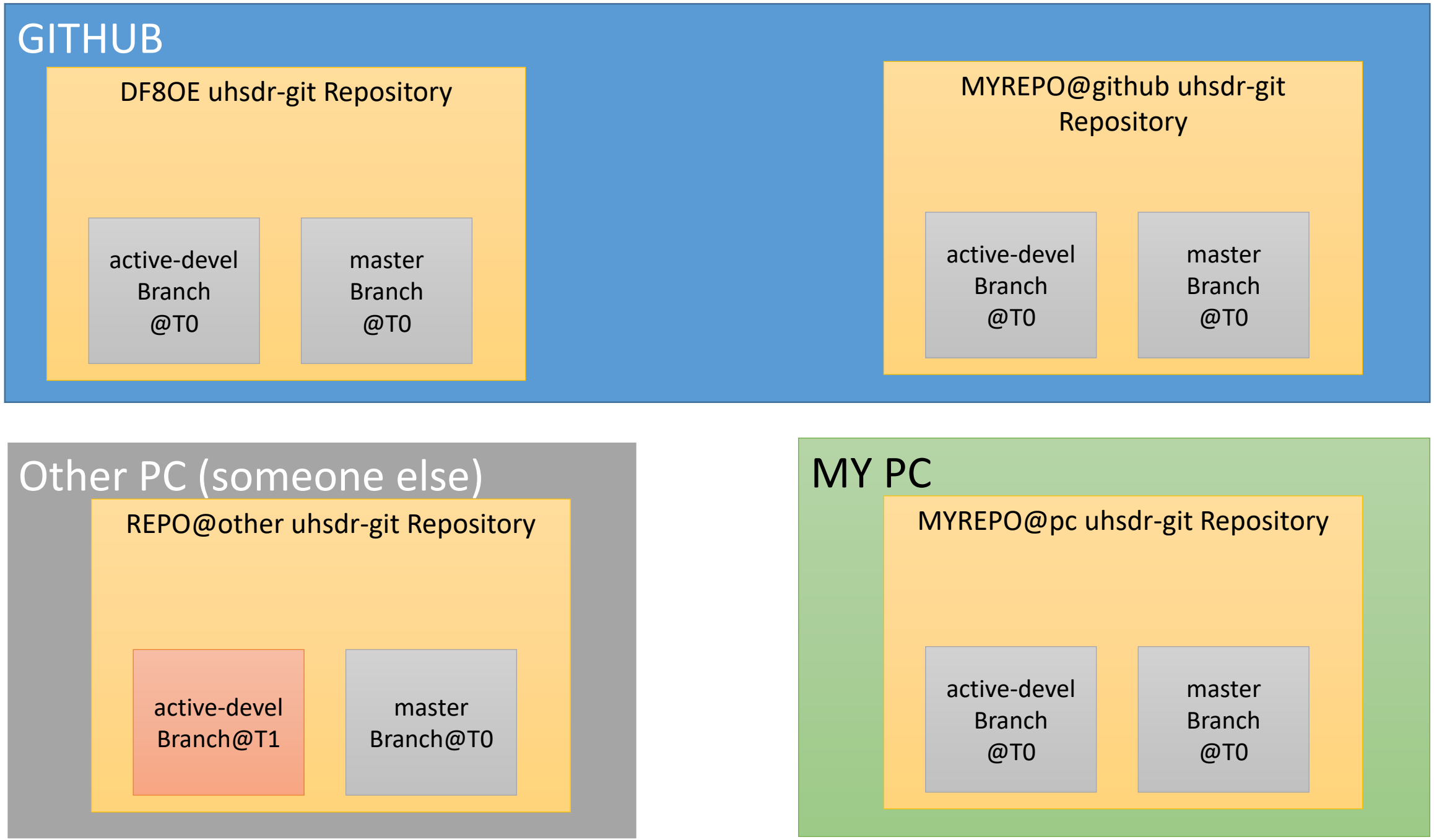
Let us go through it.

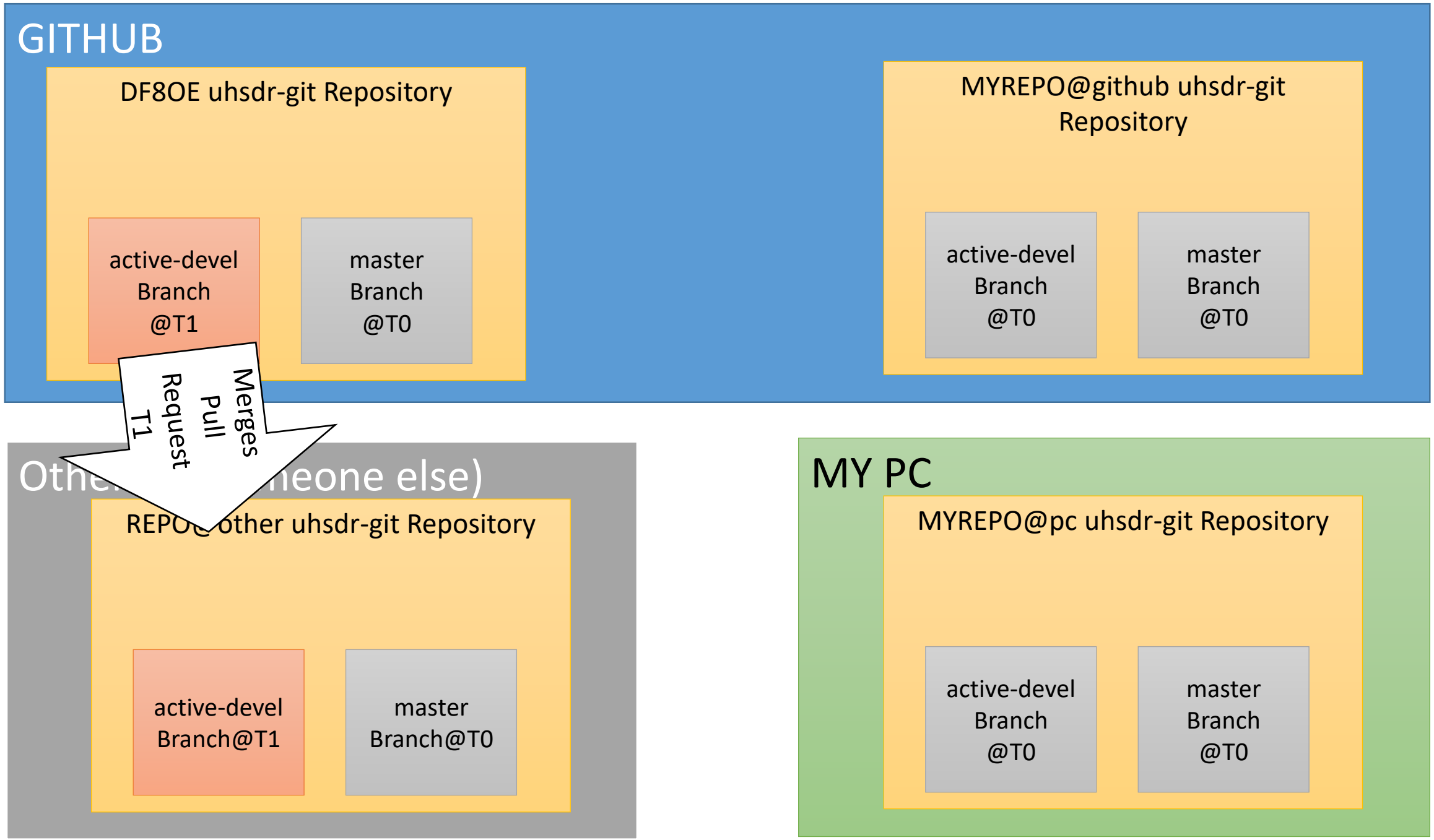


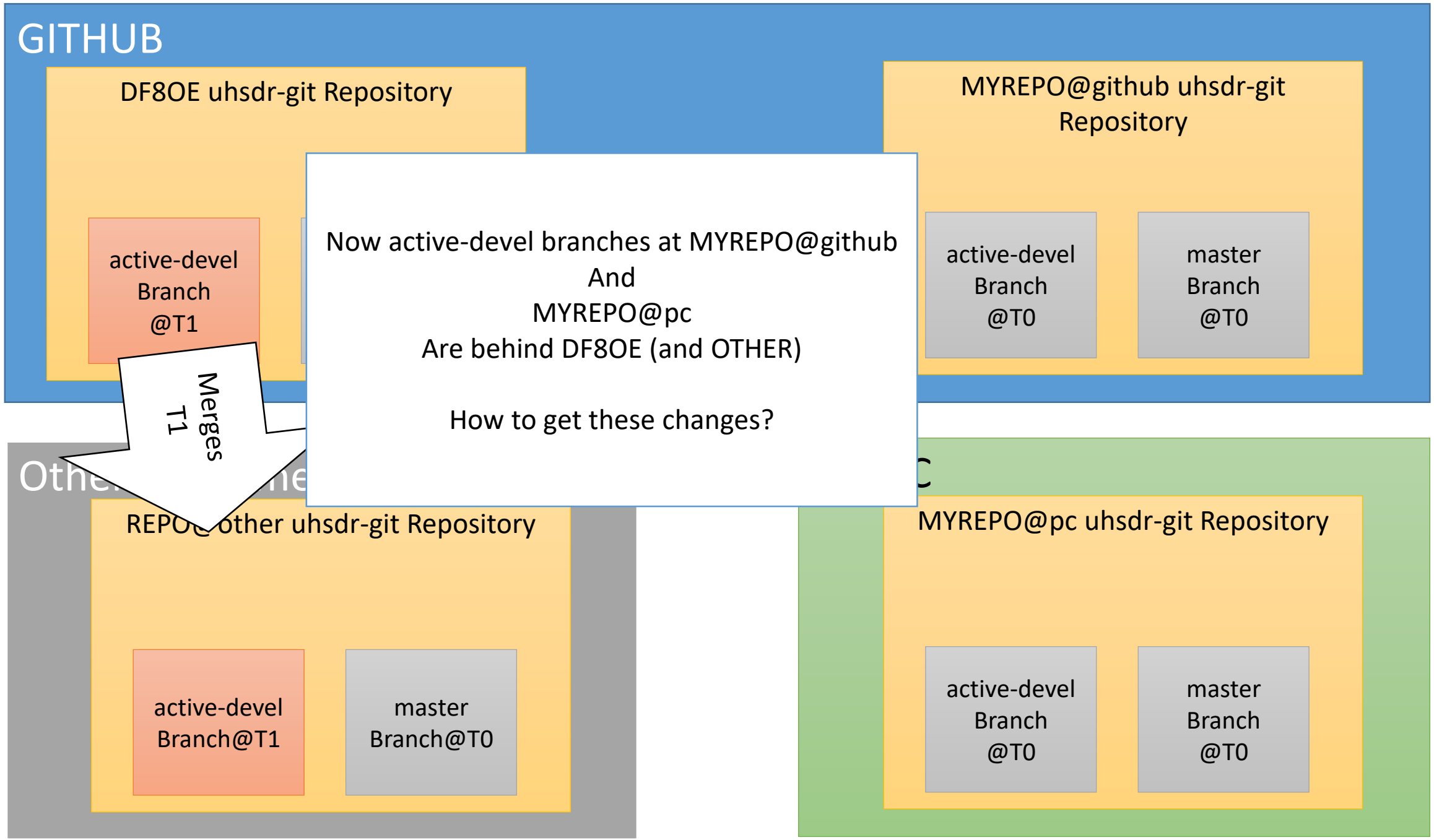
How a change is propagated into my (!) github Repository

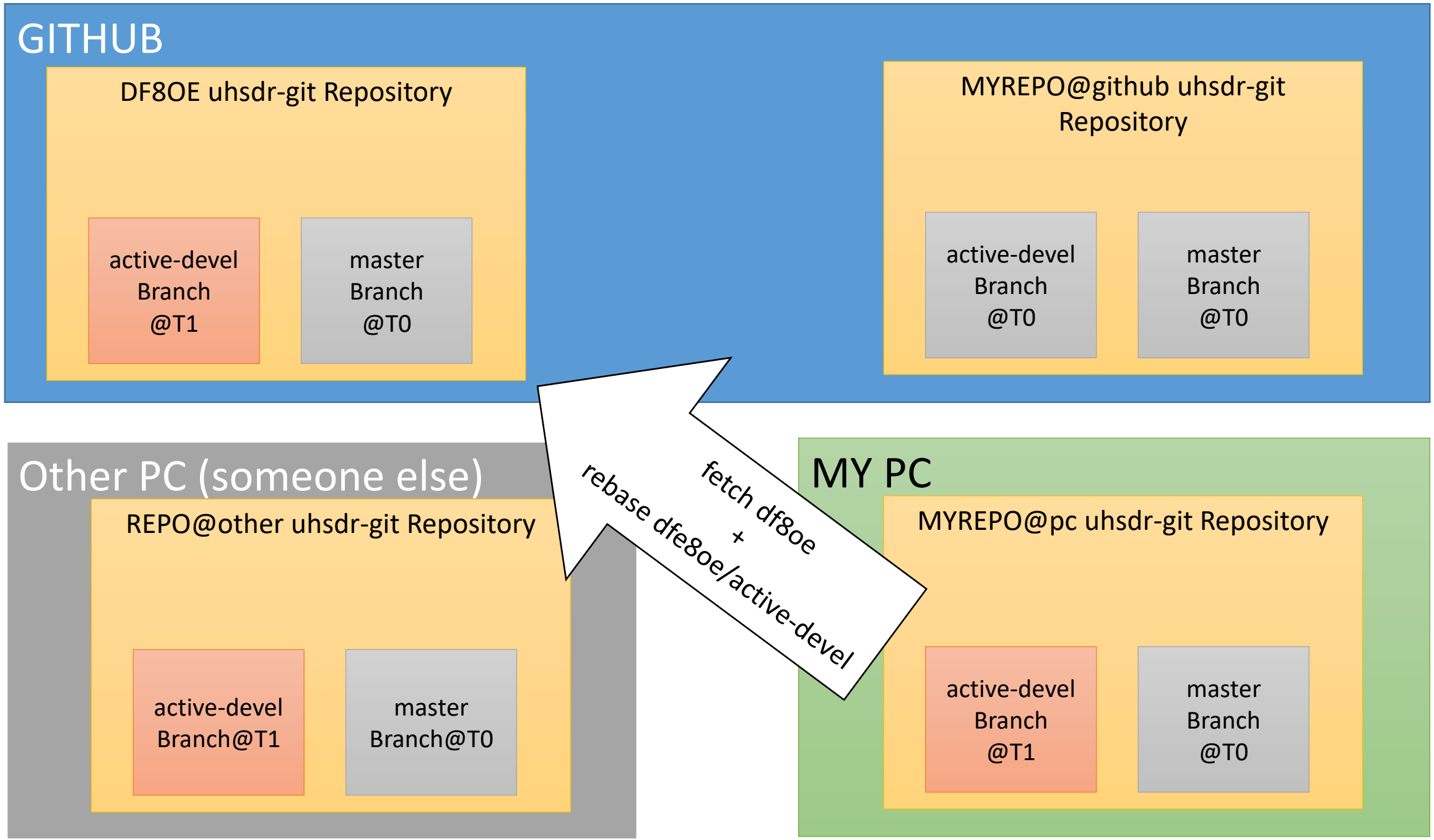
Now the most simple workflow: Someone else creates a commit and this goes into the DF8OE repository active-devel branch. You cannot directly update your github repository, but you can do this in 2 steps.

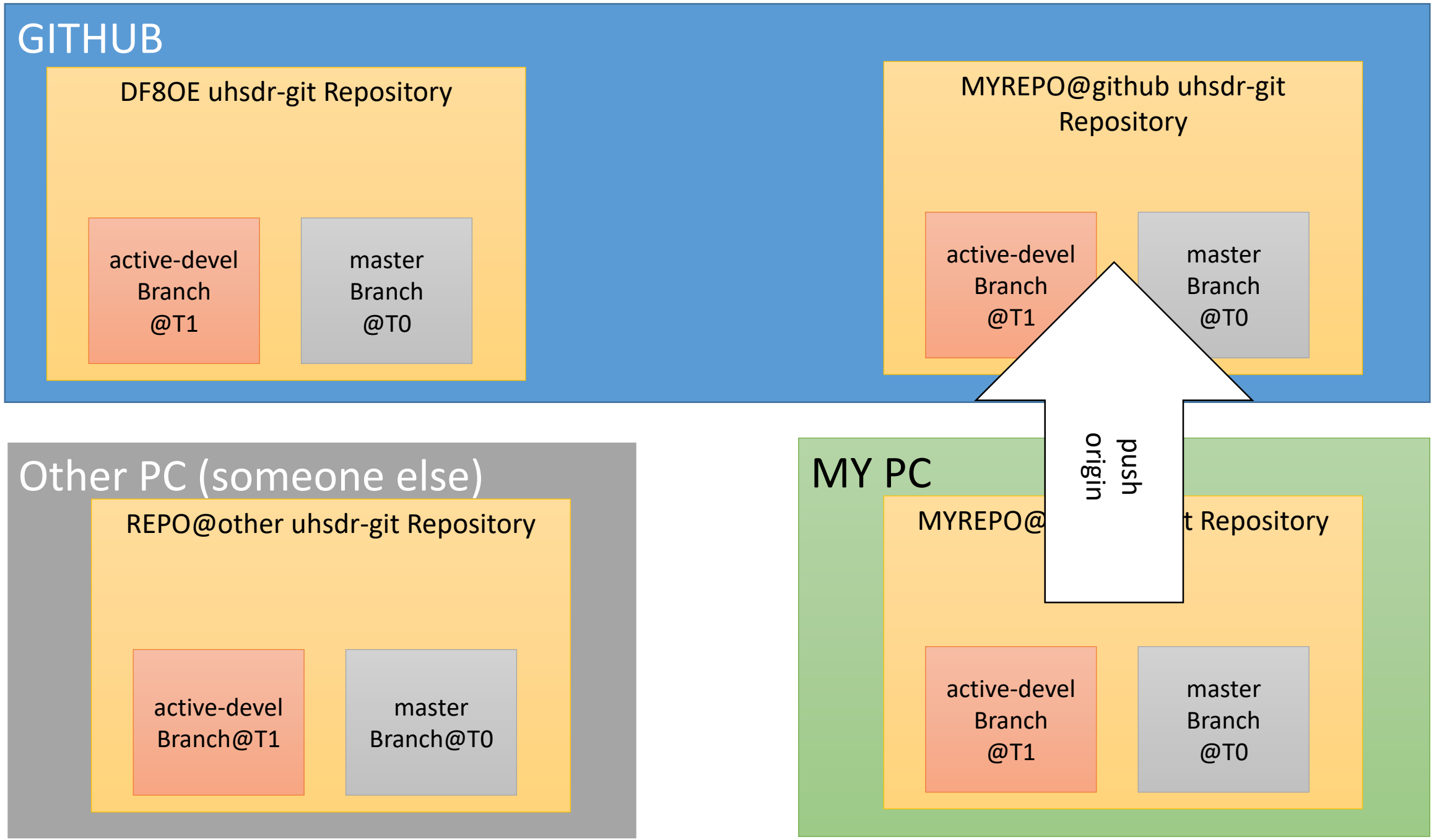
T0 is the original state of the branches. T1, T2 etc. are commits done later







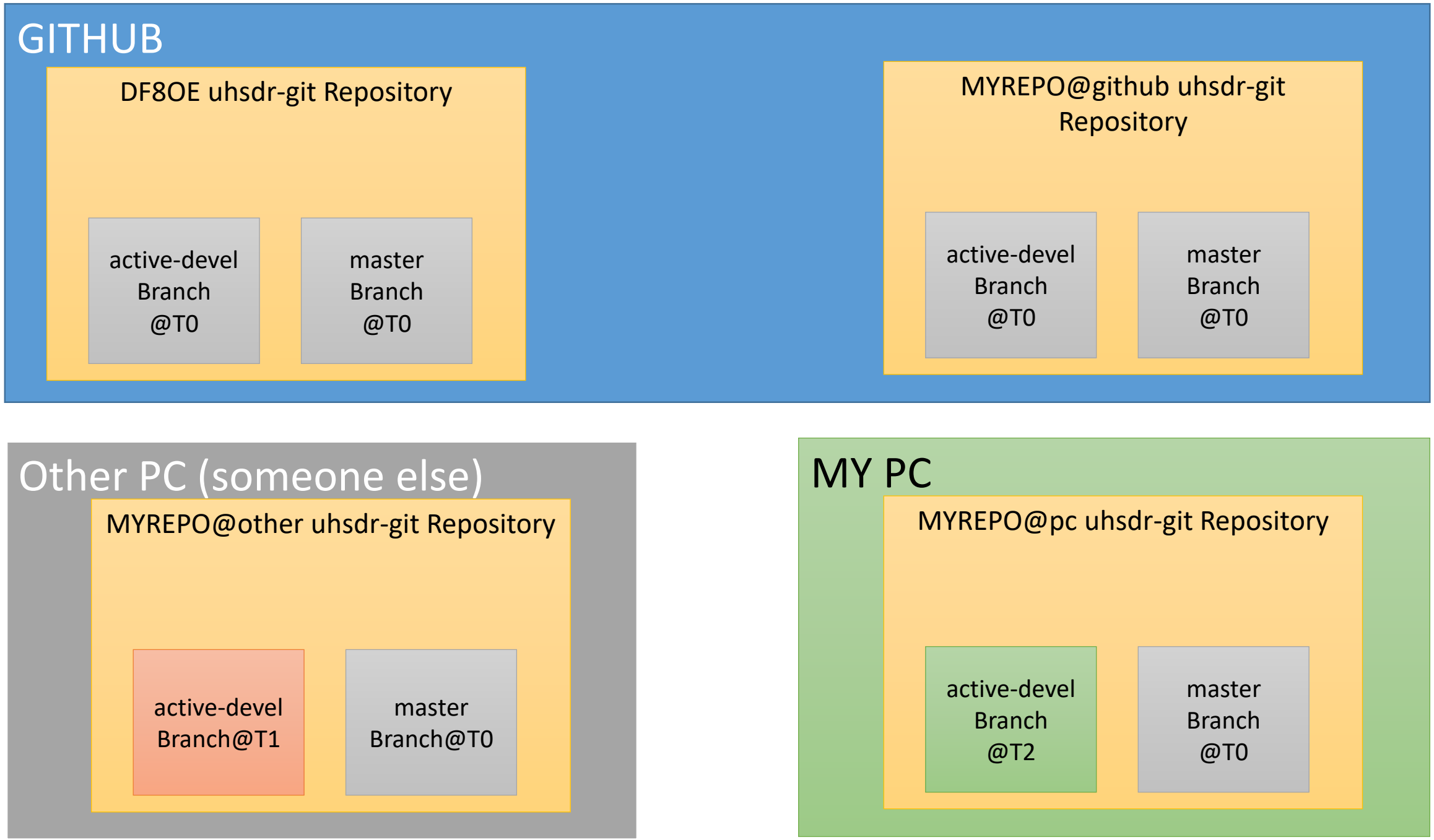


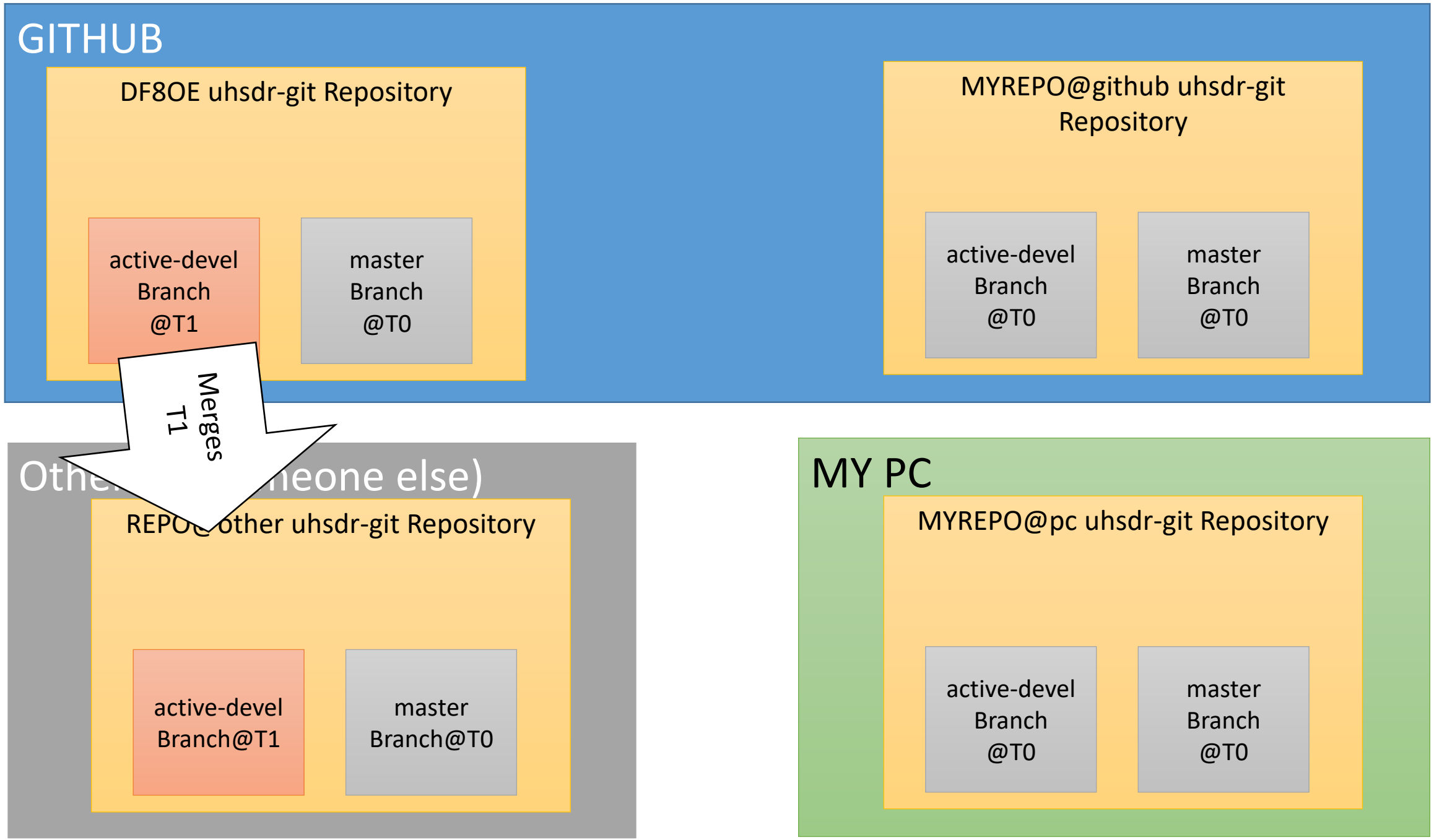


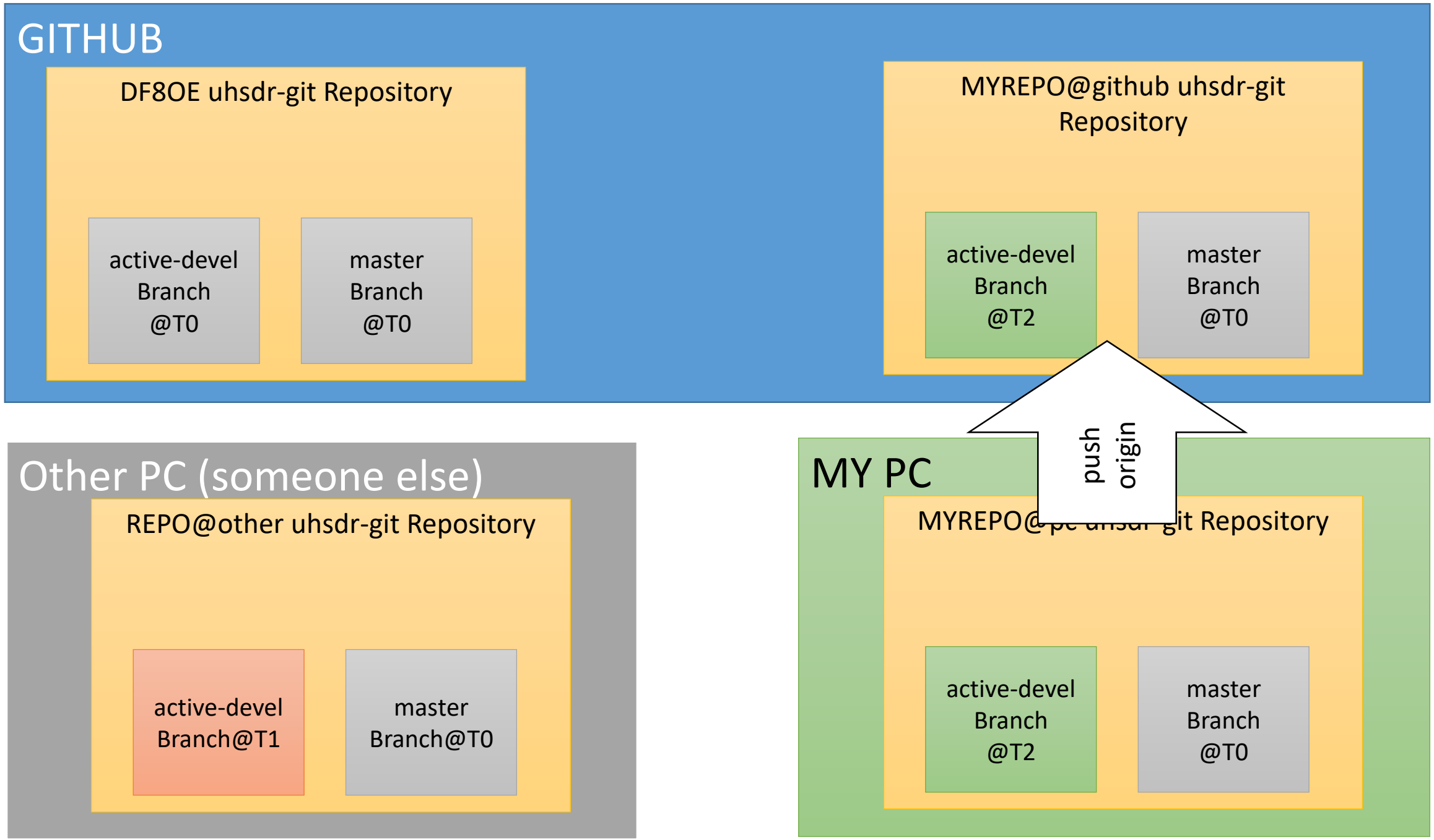
How to make changes?

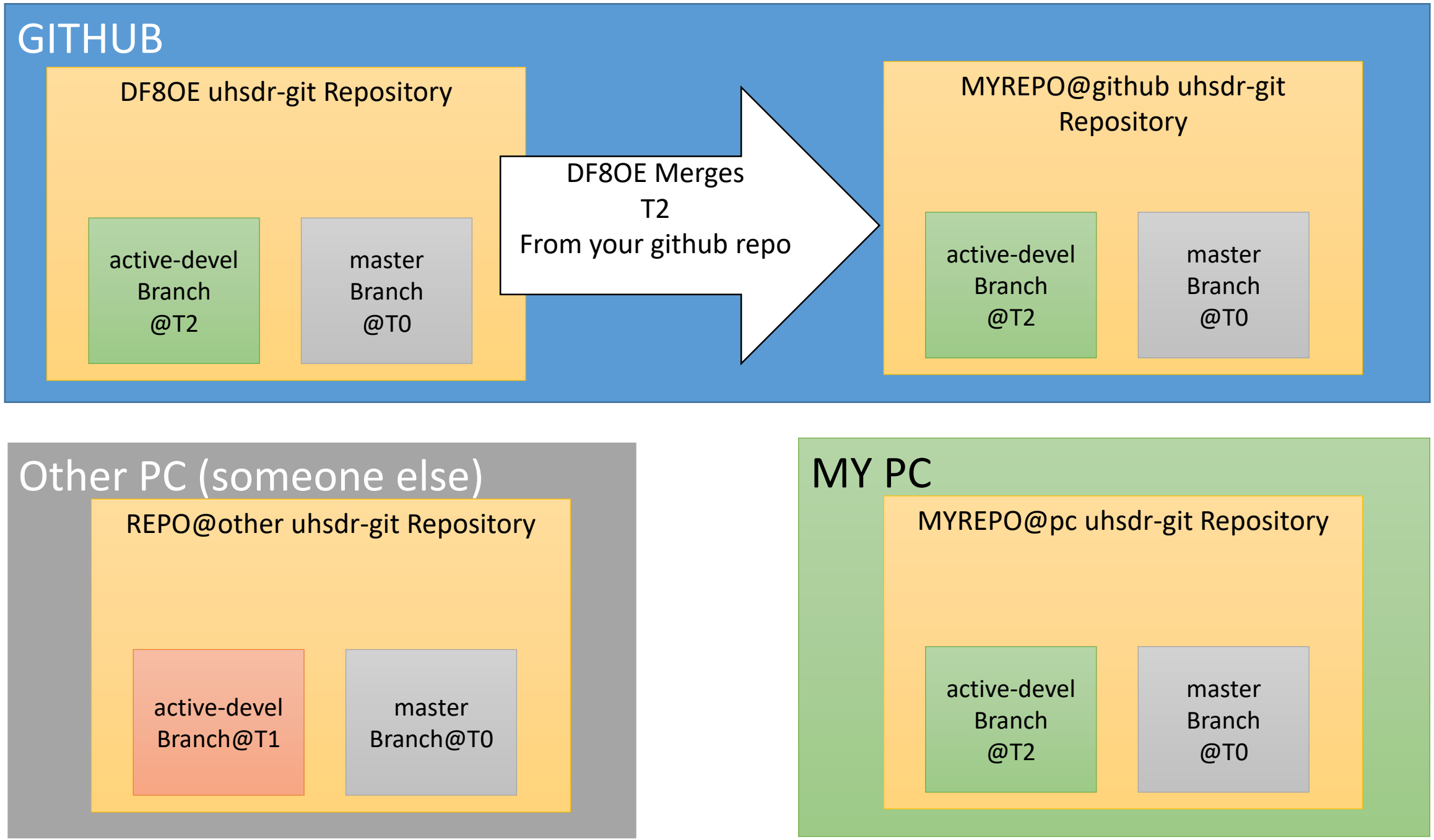
Before starting to make changes, make sure you have updated your working branch to the newest commit of df8oe/active-devel. This will ease later integration and you'll have all the latest fixes and functions.

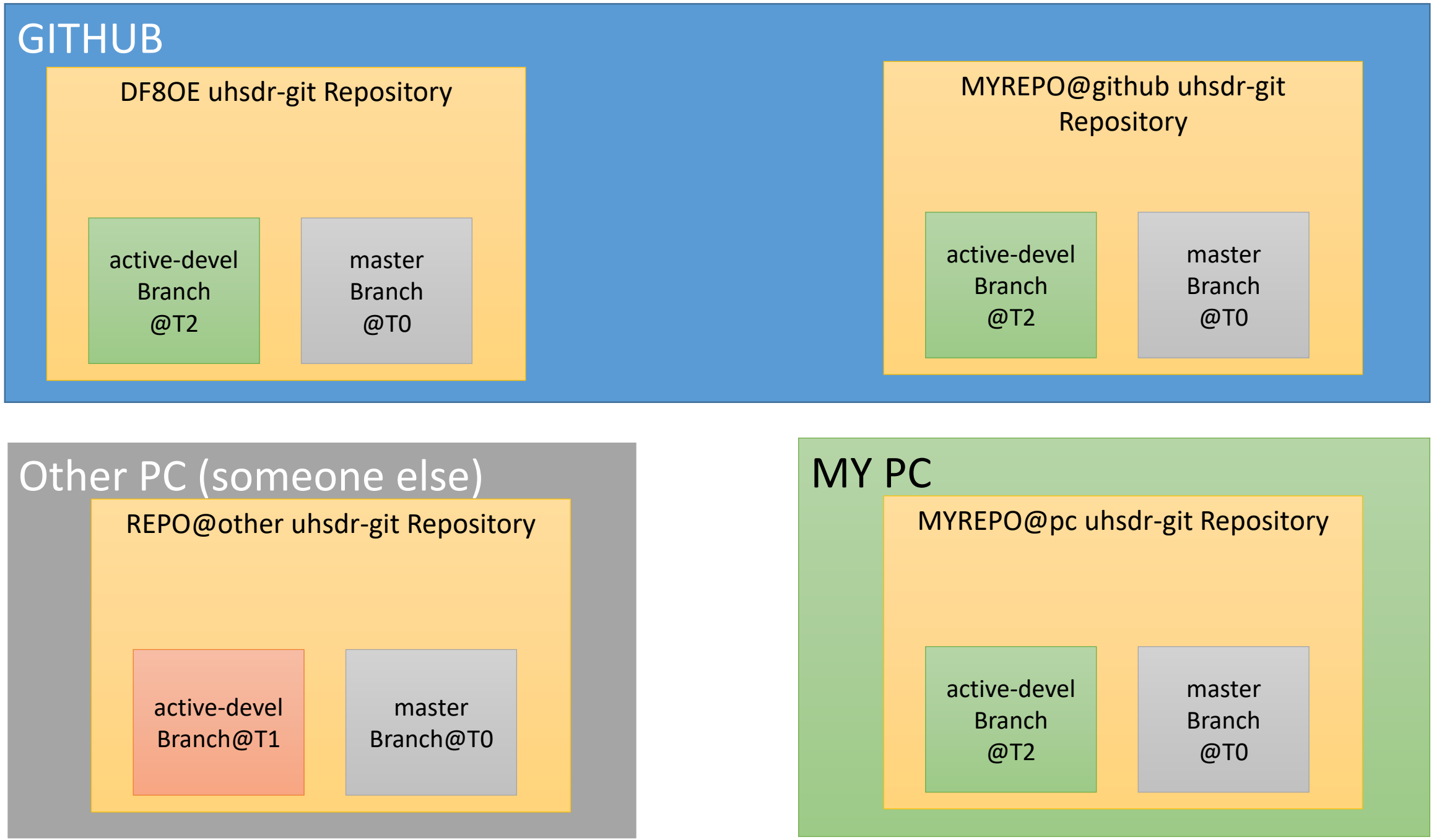
We assume now, that you did a successful rebase (which is a no-brainer, since you did not change anything yet). Your change's name is T2











How to make changes while others have also made changes before my change was finished?

In this case rebase will help us to bring the external changes into your repository on the PC. If there are conflicts, because the other person changed in areas you touch as well, you'll have to resolve these conflicts. For now, assume the changes were made in different locations. T1 is the "other" change, T2 is still yours.