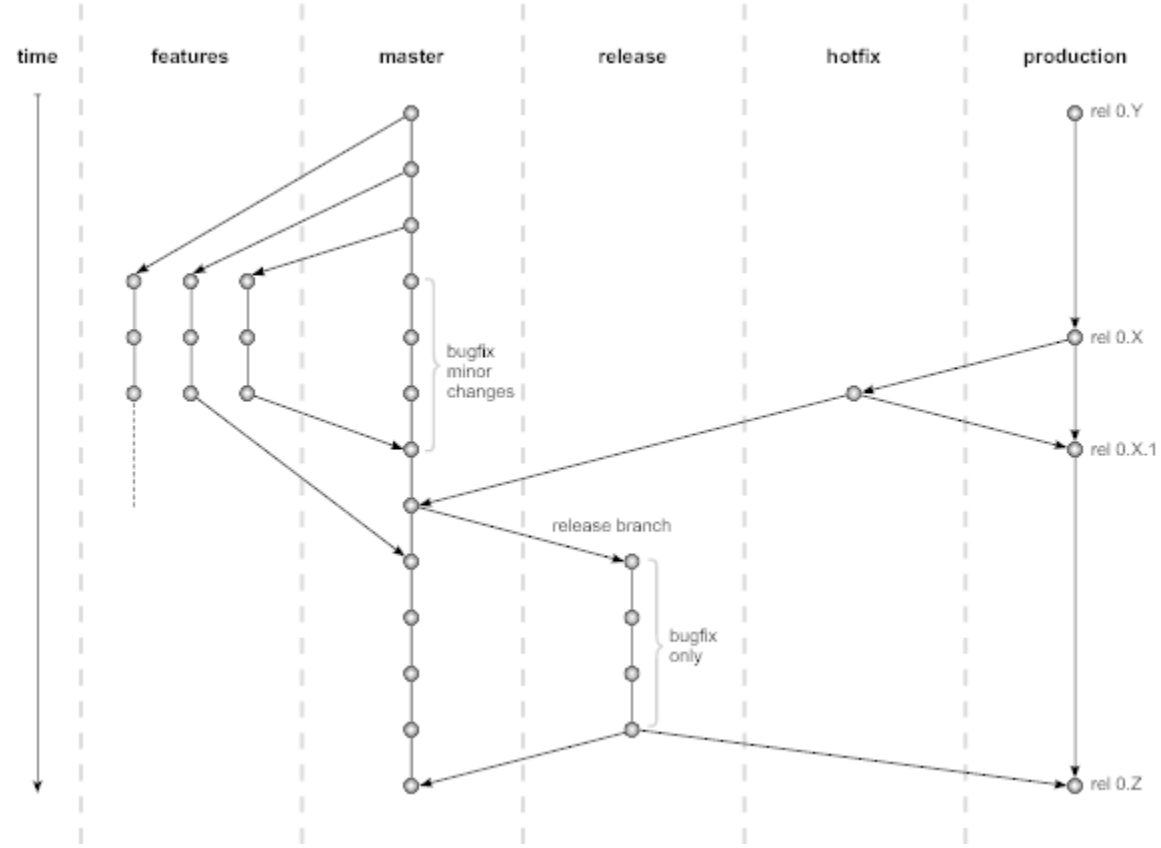
Source Code Management

For KIELER, a central authoritative Git repository is hosted on the KIELER project management server. The typical workflow of a developer is to first commit his source code changes into his local repository before pushing them to the authoritative repository on the server. The KIELER source code management process aims at always having a master branch which can be considered release-ready. This is done by shifting the main KIELER development away from the master branch to feature branches. The following classes of branches are defined:

- The master branch
- Release branches
- Feature branches



How to use branches

As we now know which kinds of branches there are in KIELER, we will now learn how to use them

The master branch

The master branch is the authoritative branch for KIELER development. On the master branch only bugfixes and minor features are pushed directly. The master branch must always be in a release-ready state. This means that all features that are merged into the master branch should be stable and complete. Feature branches are forked by the developers from the master branch. Also, if a release is planned, a release branch is forked from the master branch prior to the release.

Release branches

Prior to each release a release branch is forked from the master branch. No development of features takes place there, only bugfixes are pushed to the release branch. After a release is made, its branch is kept around and a tag of the release is made.

Feature branches

With respect to development, feature branches are the most important type of branches. All development takes place on feature branches. Only when a feature is considered complete and ready for release the corresponding feature branch is merged back into the master branch and deleted afterwards. Names of feature branches always start with the login of its developer followed by a slash and a descriptive feature name, as such: `cds/label_placement`