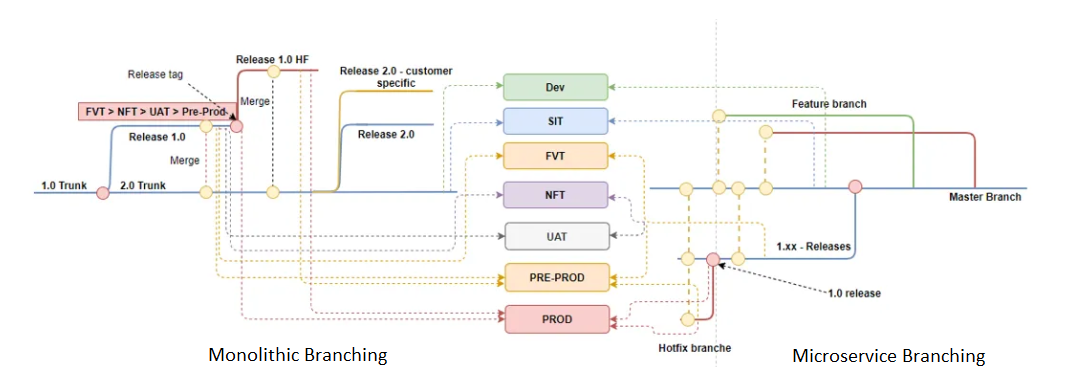
The migration branching strategy consists of the following branches:

* Main Branches
  + Master
  + Trunk
* Support Branches
* Release Candidate branch
* Feature branch
* Hotfix branch



The monolithic release cycles and strategies are not able to change, especially during the initial stage of migration. We must think of a way to extend the microservices branching strategy to support monolithic requirements.

The approach suggested here is to create release branches for the major microservices API releases. A separate release branch will be created for the API releases, as the pre-release branch and various deployments can be done with that branch.

Once testing has been completed, a separate release tag can be created on the release branch for the proper release, and the hotfix branch can be created parallelly.

**Proposed Flow:**

* Combining the benefits of feature branching and trunk-based development
* hybrid branching encourages regular merges to prevent code conflicts and ensure that the main branch remains stable.
* Encourages regular merges to prevent code conflicts and ensure stability.
* Hybrid branching also provides the flexibility to adopt different branching strategies based on the project's needs.
* The hybrid branching strategy provides a balance between the isolation of feature branching and the continuous integration of trunk-based development.
* Balances isolation and continuous integration for a more efficient development process.
* The hotfix branches are created in case of any critical issues to be fixed on the production release being done from the master branch.