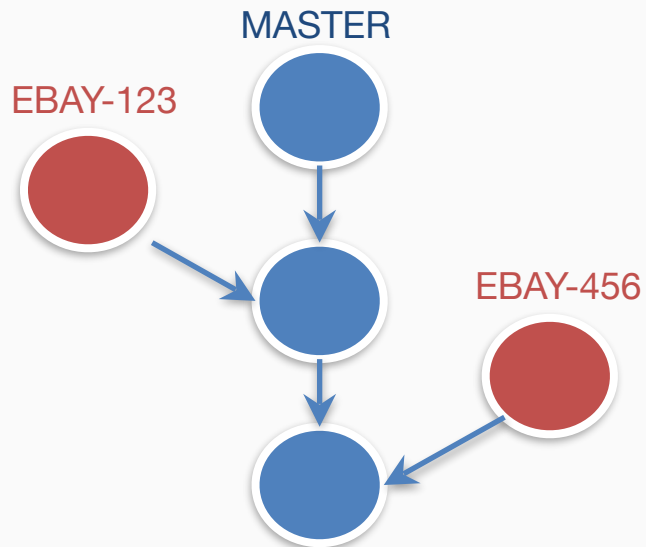




Edge API Lifecycle and Tools

Branching Strategy

Master Branch



Convention to call default branch master

Most tooling accommodate this concept

Branch from and merge to this

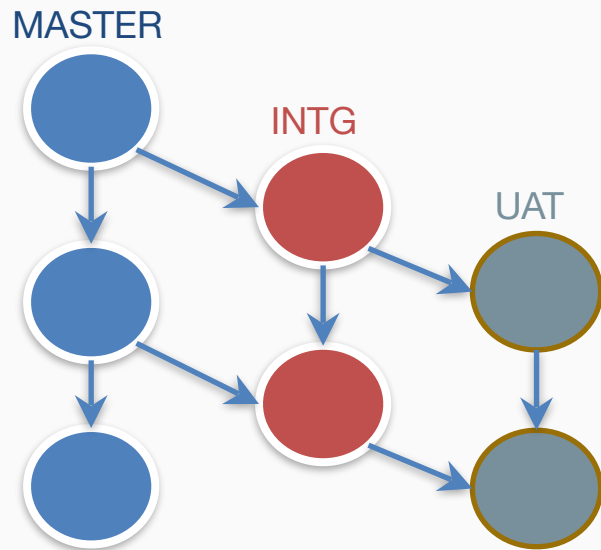
Feature branches

- Short lived – a day

- Named after JIRA issue

- Ensures issue tracking

Environment Branches



When deploying to other environment, just merge to downstream branch

Commits flow downstream – ensure every commit follows SCM

Use 'git diff' to see changes

See release history using 'git log'

Hotfixes

Scenario 1 – production branch is close to master

Hotfix on feature branch – tracked with JIRA

Merge to master – test

Cherry-pick to other environments, do work locally

Similar to Linux kernel development

Linux -> Subsystem (netdev) -> Device

Hotfixes

Scenario 2 – hotfix on staging

Hotfix on staging branch – deployed to staging via CI

Merge to production branch – deployed to prod via CI

Cherry-pick back to other environments, do work locally

Merge Requests

Online place to discuss and review code

Share your in-progress feature, ask for a review

If you work on a branch for more than a few hours

Do MR without assigning to anyone. Mention people in comment.

Code is not ready for merge, but feedback welcome

Anybody can comment or push changes

Ask for a merge

Assign MR to somebody

Commit Often with Right Message

Commit message to reflect intention, not contents of the commit

Contents are already visible

More important than what is why

Stay away from: change, improve, refactor, fix

BAD:

```
git commit -m 'add user.rb'
```

GOOD:

```
git commit -m 'create user model to  
store user session information'
```

Merge Master Into Feature

WHY

- Leverage code pushed to master

- Solve merge conflicts

HOW

- Cherry-pick individual commit

- Merge master to feature

WHEN

- Start of the day – CI strategy

- Well-defined points – synchronization point strategy (e.g., Linux) when the state of the code is better known

- Don't merge at random points – don't litter the history (Git rebase is your friend)

Feature Merge into Master

Testing the feature branch on its own may not be enough. It is not the merged result

Merge your feature branch into master and test locally. Solve conflicts locally.

CI to incorporate merge requests

short-lived feature branches help a lot to minimize conflicts

THANK YOU