# Release Management and Stuff

Bryan Call, Oath
Leif Hedstrom, Apple

# Releases

# LTS releases

- GOAL 1: All versions within a major version are 100% backwards compatible (except possibly for severe security issues). New features should be safe, and non-invasive.

- GOAL 2: Upgrading within a major version is always safe, upgrading from v8.0.0 -> 8.1.0 should be as safe as upgrading from v8.0.0 to v8.0.1.

- GOAL 3: We only maintain one LTS branch for every major version, for ~2 years + 6 months sunset

# Don’t be a git

- Use the git pre-commit hook that we provide!
- Run make clang-format
- Rebase your branches if you are pushing a new or updated PR

# Don’t be a git (cont)

- Continue to rebase and merge?
  - Rebase and merge vs Squash and merge vs Create a merge commit

# Don’t be a git (cont)

- How do we deal with PRs with multiple commits?

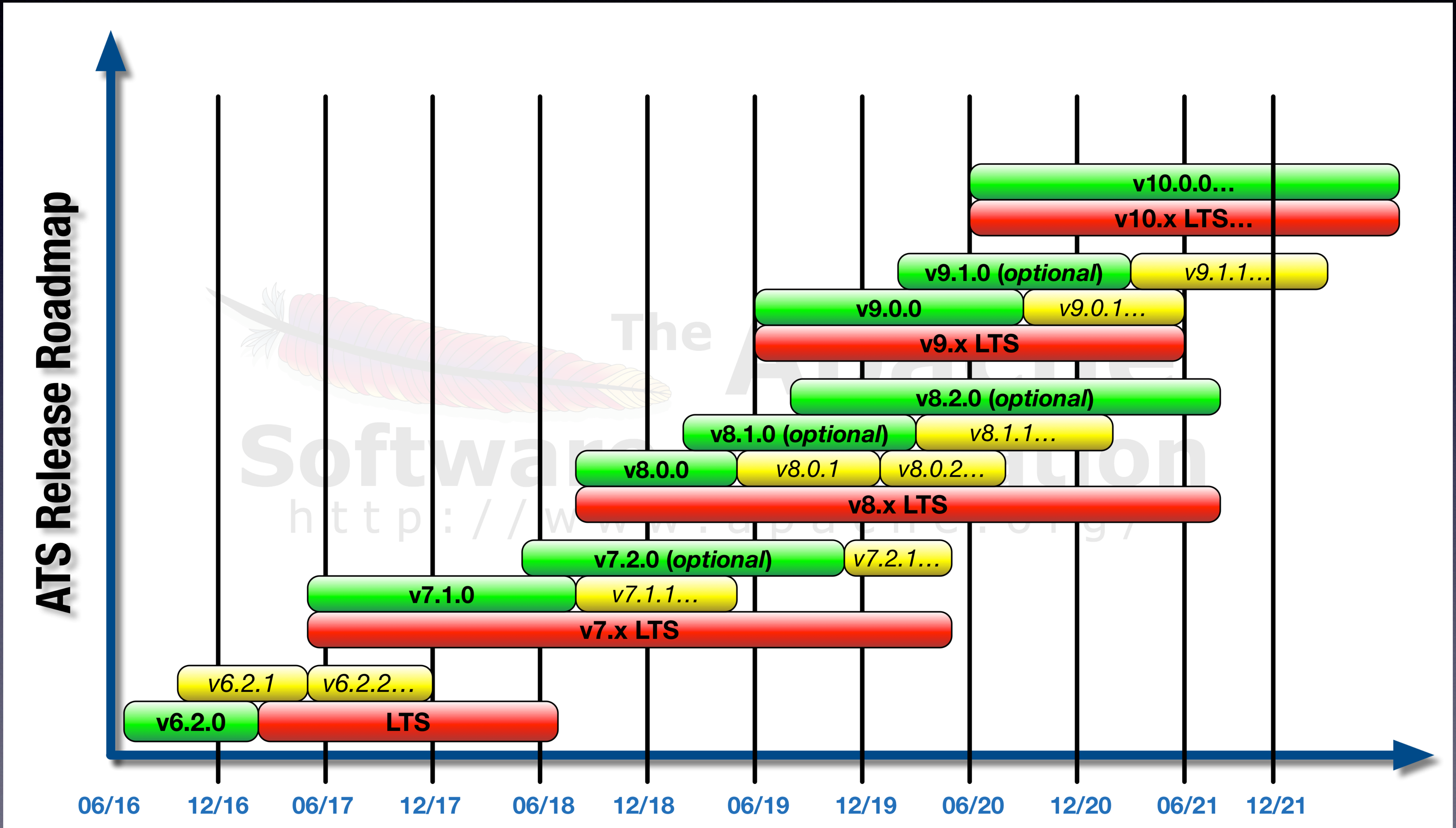
    1. Don’t do it
    2. Merge commits
    3. Squash before merge

- The issue is that it’s difficult for RMs to handle such PRs for back ports

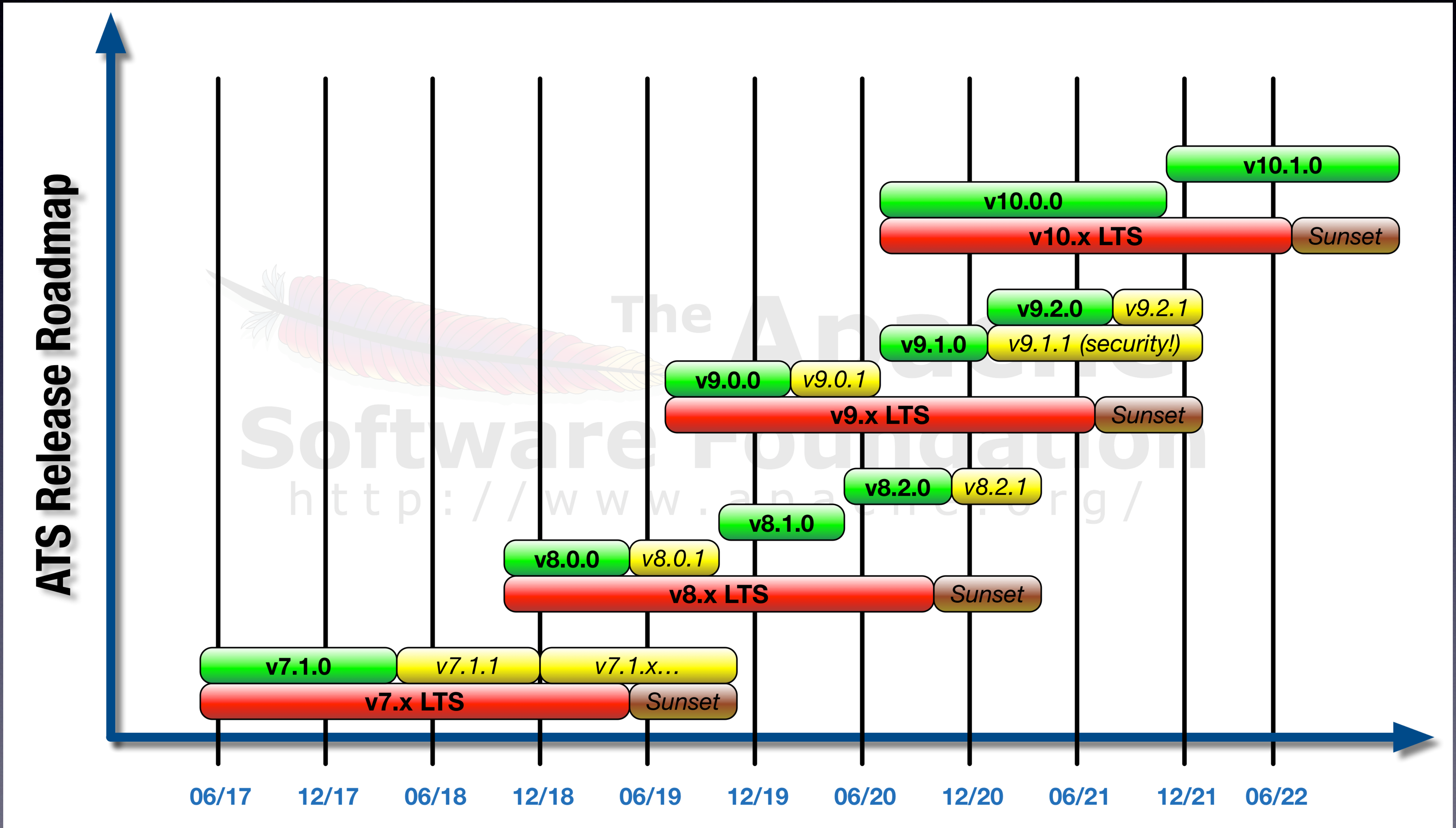
# Don’t be a git (cont)

- Create new PR for 7.1.x
- Mark PR as "Projects 8.x releases"

# Current LTS schedule



# New LTS schedule



# GitHub / Misc

- GitHub PRs are rebased on master when running tests
- Don’t include external bug ids in commit comments and PR titles (e.g. YTSATS) - PRs comments are OK
- docs.trafficserver.apache.org runs master built with ASAN

# Release Process

- Branch tree
- Github PR project gets marked for the release
- Cherry pick to release
- Test
- Make release candidate
- Test more
- Vote - How do we get more people to test and vote?

# 7.1.x Releases

- Who is going to test it?
  - Our largest users a
- How do we get the votes?