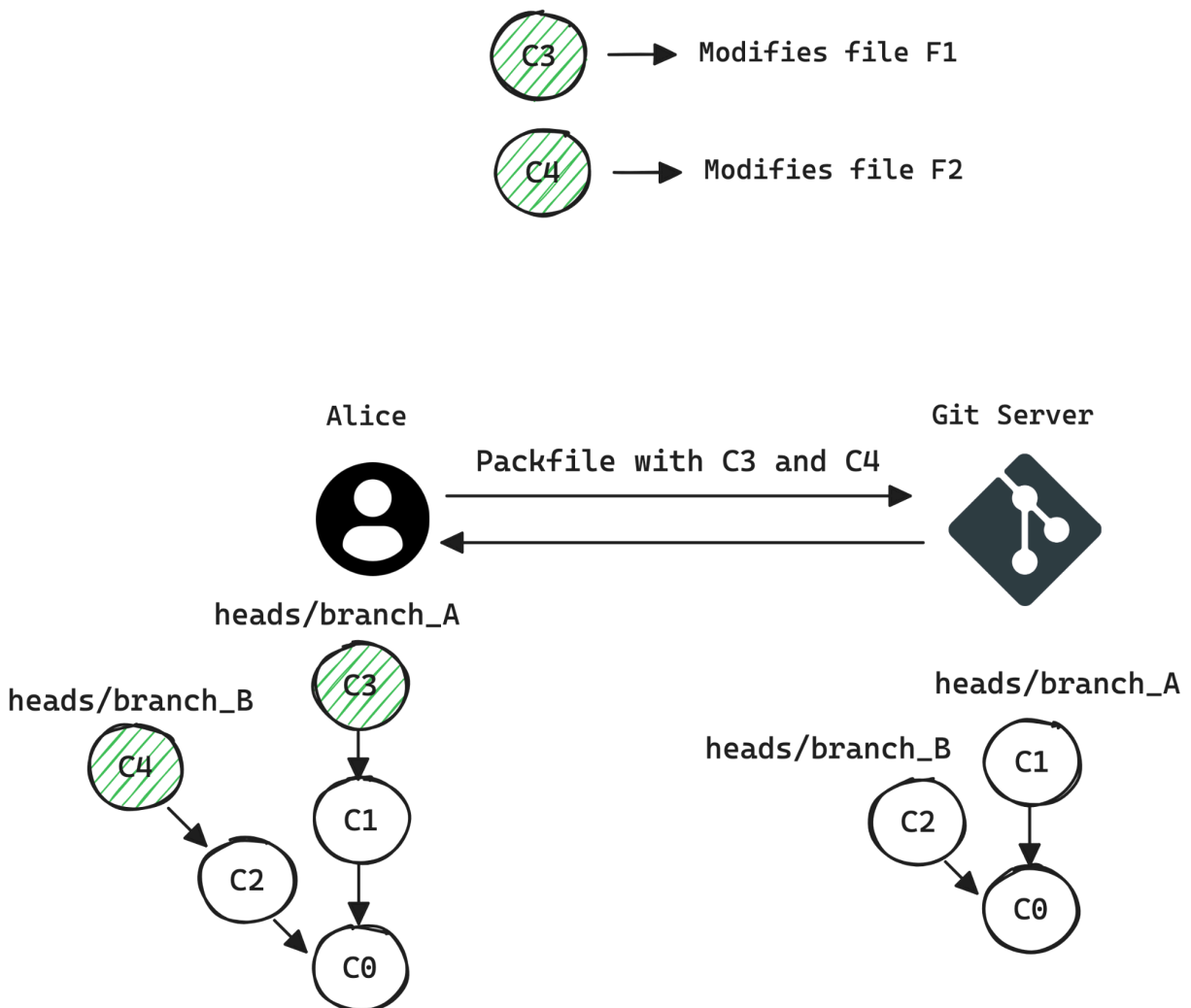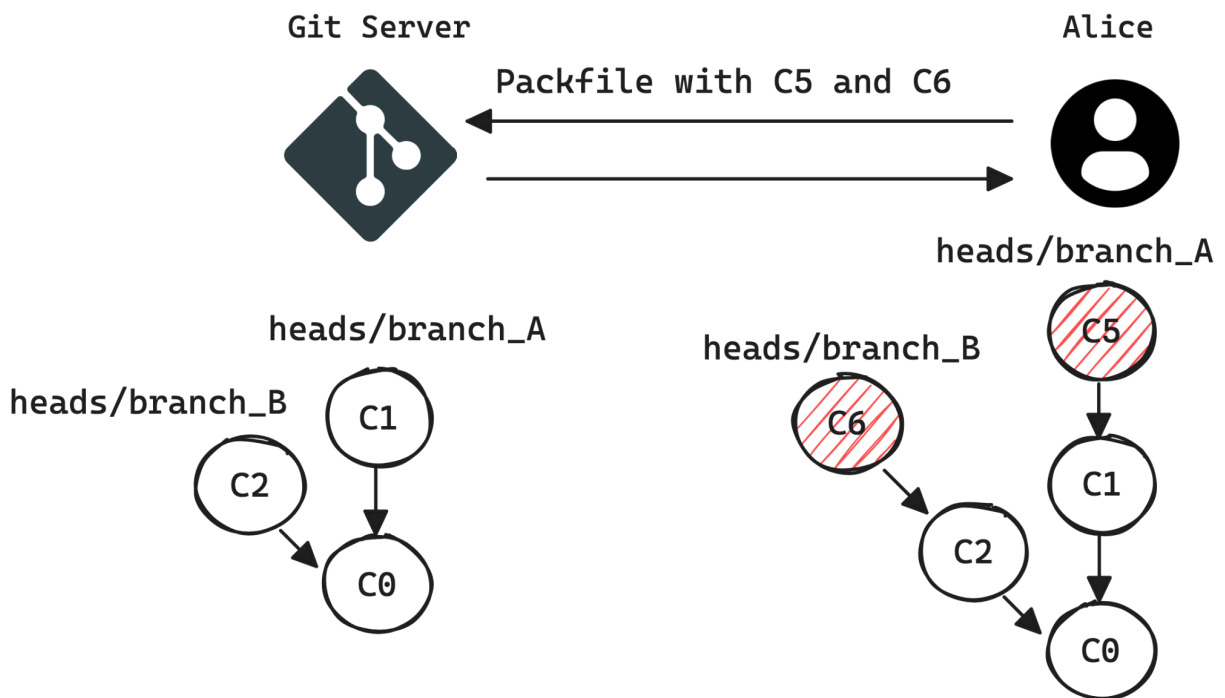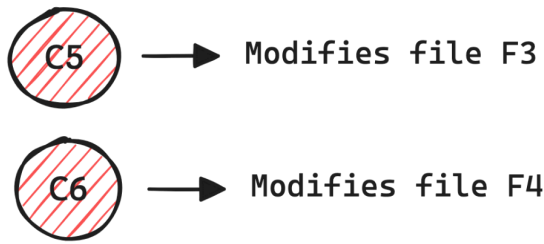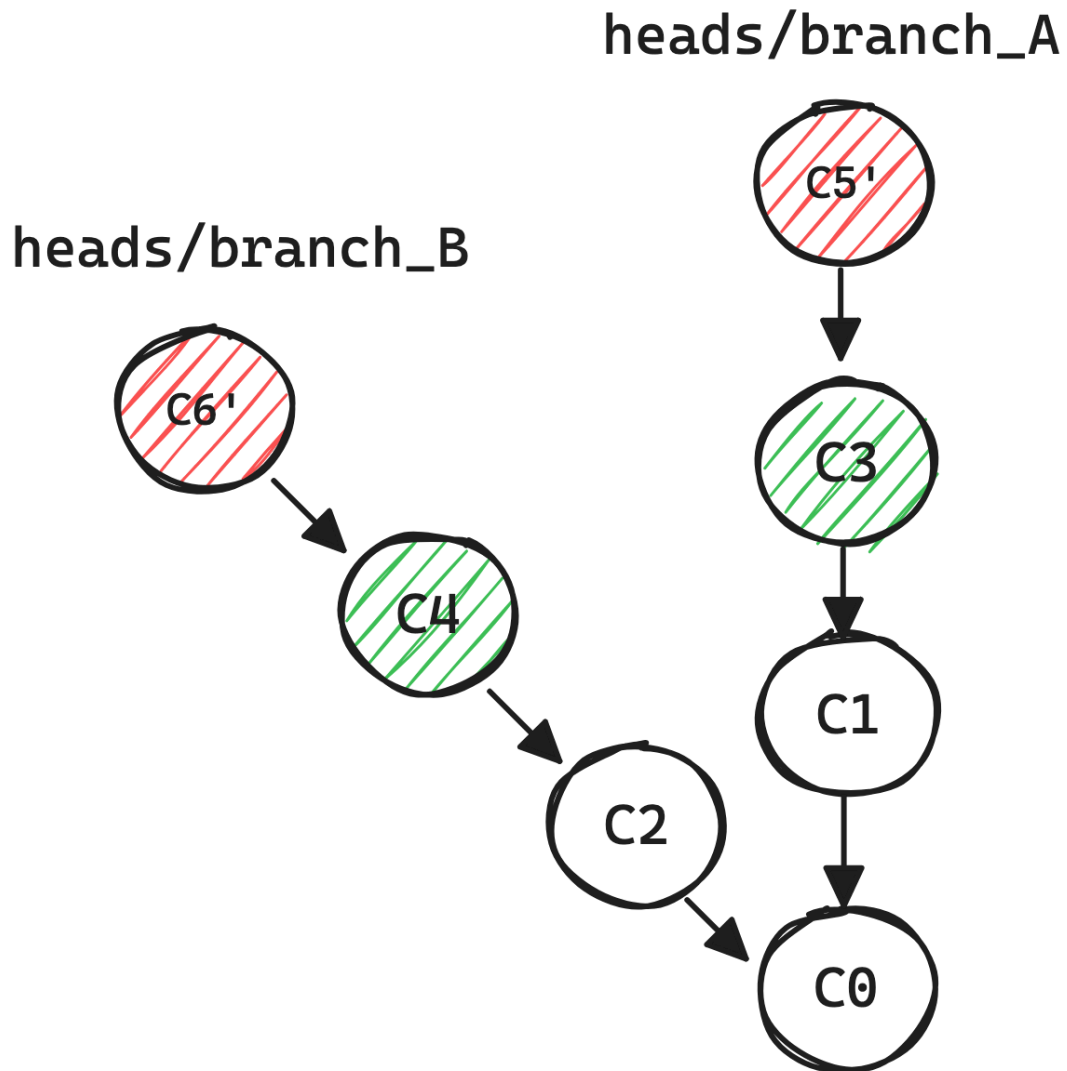# Git Push Protocol v2

Auto Push Rebase

Problem Statement

- Alice and Bob have independent changes on branches branch_A and branch_B and they are trying to push to the server at the same time.
- If they are concurrent in their push, one of the two pushes would get rejected even though the changes were independent of each other

C5 → Modifies file F3

C6 → Modifies file F4

Git Server   Alice

Packfile with C5 and C6

heads/branch_A

heads/branch_A

heads/branch_B
C1

C2

C0

heads/branch_B
C6

C5

C2

C1

C0

Desired Behaviour

heads/branch_A

heads/branch_B

C5'

C6'

C3

C4

C1

C2

C0

Issues

Commit Signing
Merge Commits
Thousands of highly concurrent pushes

Commit Cloud

Problem Statement

- In most organizations, CI and other machinery needs to run on a user's changes that haven't yet been merged/added to master/main.
- This requires some way for the server to distinguish between public commits (merged into actual release branches) and local commits (yet to be merged/added).
- Many code-review systems work around these problems by creating special refs (like refs/for) which represent commits that are currently under review or not-yet-merged.
- However this leads to the development of special language specific to each Code Review system
- If this is an industry wide requirement (i.e. running tests, checks and validations on user's unmerged changes), it would be useful for Git to have a first class representation of it in its core model

Desired Behaviour

- Commit Cloud as a feature could provide a common language that would allow disparate systems to understand each other and repositories to be easily migratable from one set of systems to another

Issues

TODO:

Resumable Pushes

Problem Statement

- Current push is very all-or-none in nature
- If the push is big, this can become difficult.
  - It might take longer, making your reverse proxy or load balancer to just terminate the connection while the server is processing the push
  - If you have a slower connection, you have high chances of failing in between and losing all the progress made till that point

Desired Behaviour

- Server allows for push continuation. Client can pick up right where they left off
- Servers break up the push phase into upload and update so its easy to keep track what is completed and what is pending