Training on Web Scraping Prices for CPI

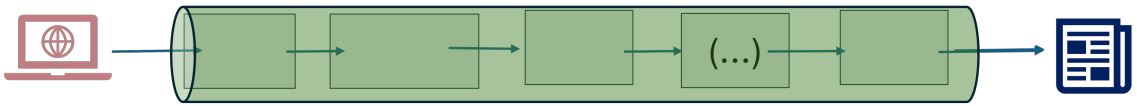
How to RAP

Christophe Bontemps & Serge Goussev

What does a RAP look like?

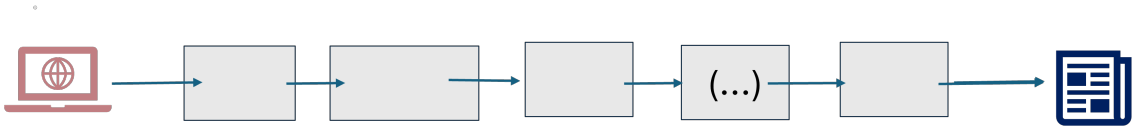
Idealy, Input (website) and output (report) are linked

What does a RAP look like?



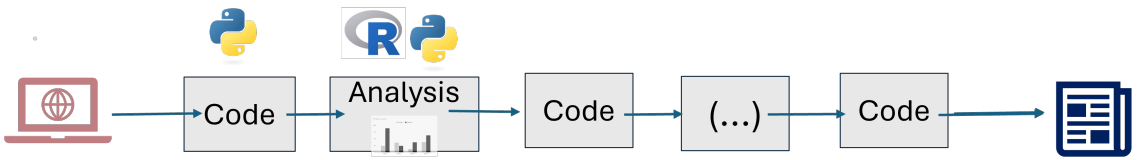
Many steps are needed to create a report

What does a RAP look like?



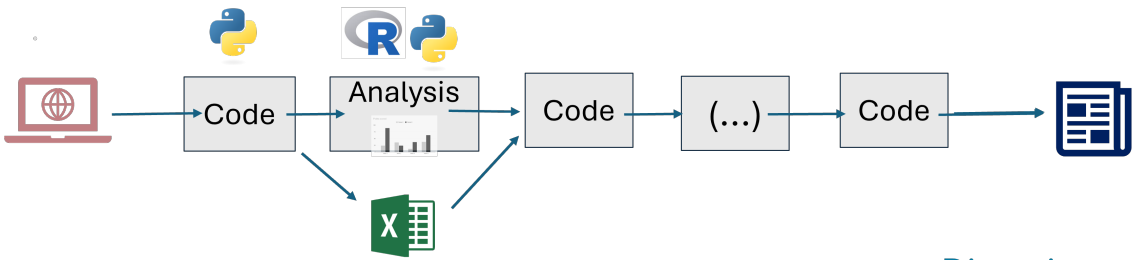
All steps should be linked in a structured process

What does a RAP look like?



And only through code (Python, R, *Stata*), no copy/paste

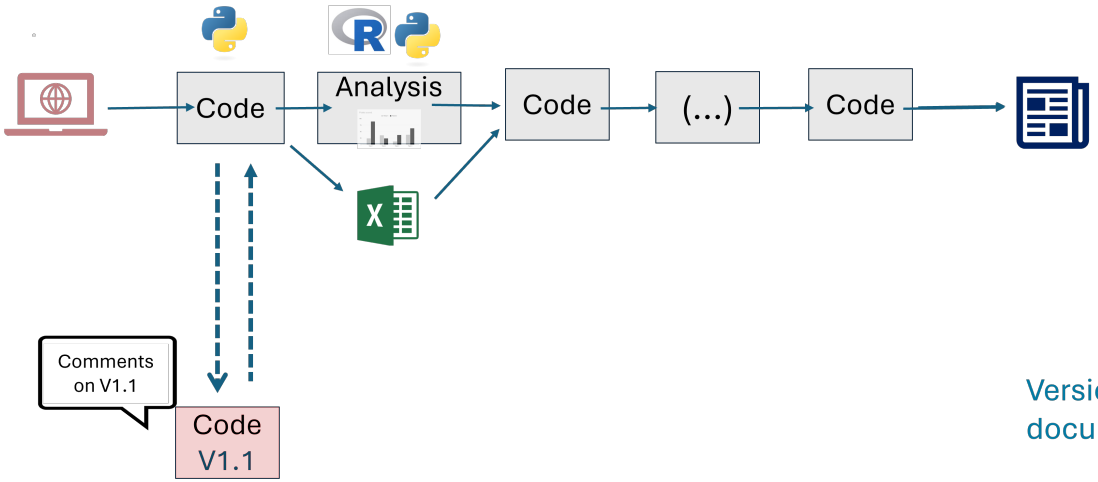
What does a RAP look like?



Direct input-output of data

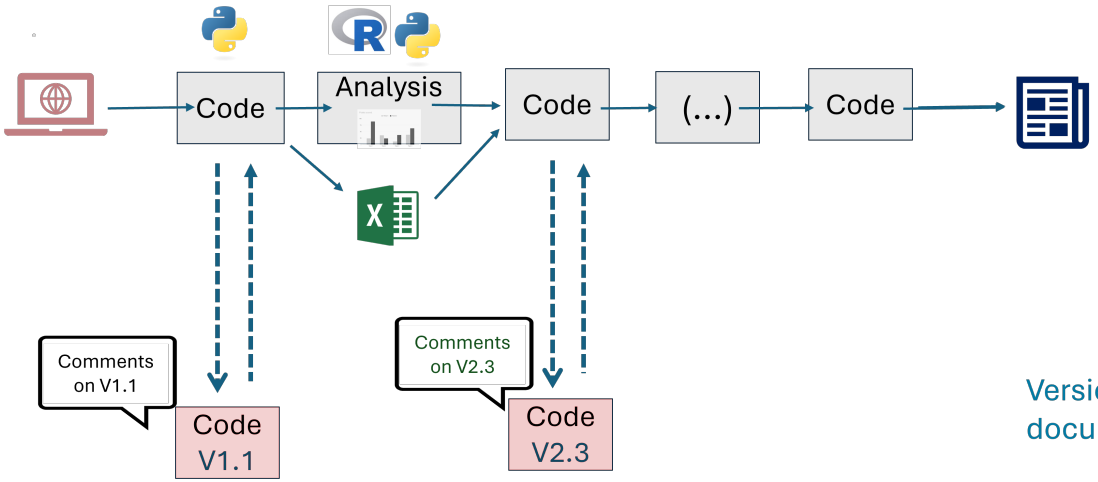
There may be side-products, but with explicit output-input links

What does a RAP look like?



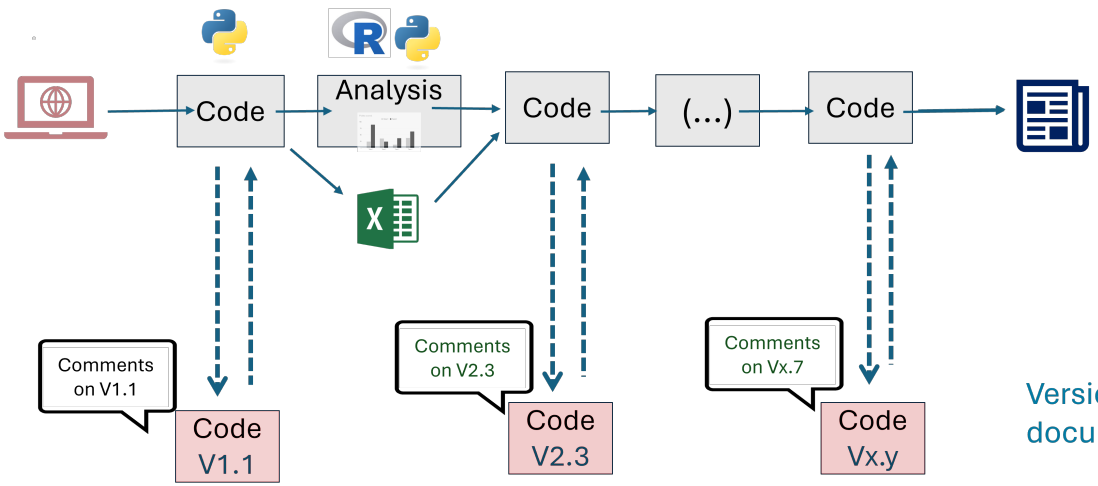
If needed, code can be updated (new versions)

What does a RAP look like?



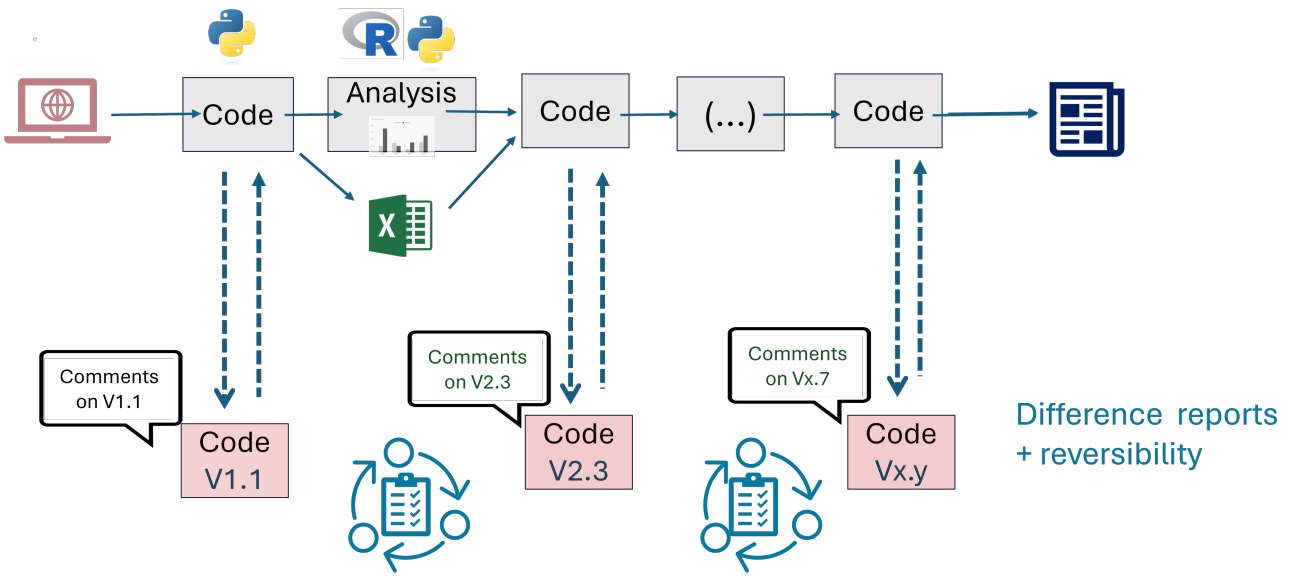
And comments added for each change

What does a RAP look like?

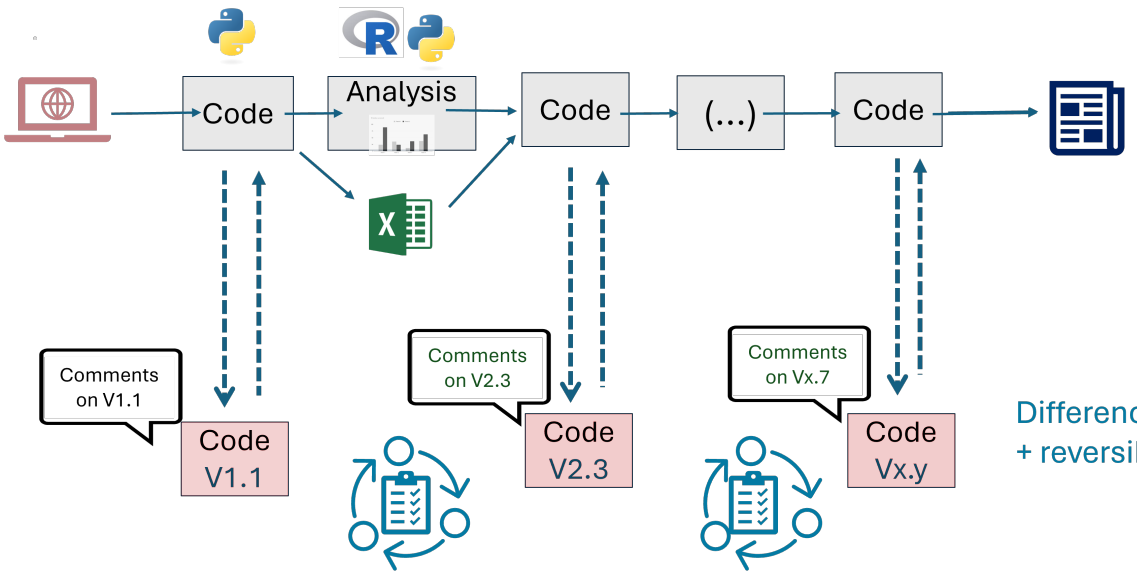


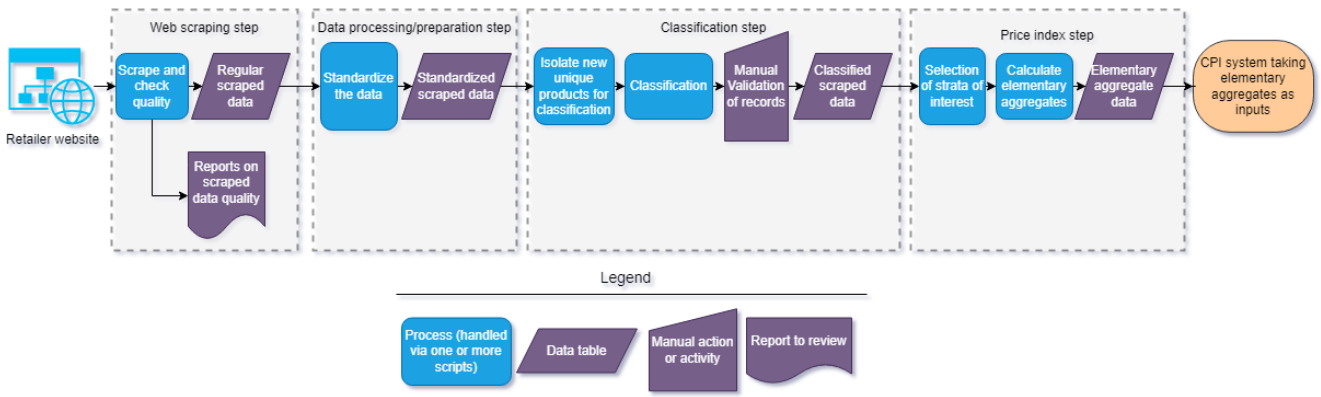
Documentation on the process builds up with code changes

What does a RAP look like?

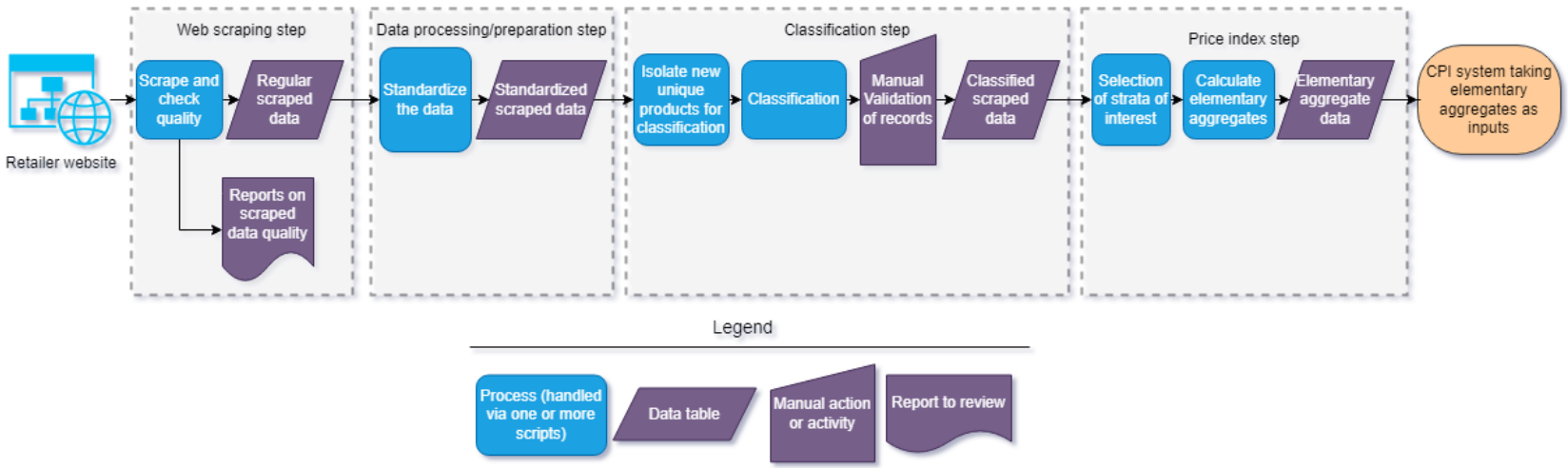


What does a RAP look like?





What does a RAP look like?



RAP principles 1: Automation - Modular coding

RAP PRINCIPLES 1: AUTOMATION - MODULAR CODING

- Automation (*as much as you can*)

RAP principles 1: Automation - Modular coding

- Automation (*as much as you can*)

↪ Where would you start?

RAP principles 1: Automation - Modular coding

- Automation (*as much as you can*)

↪ Where would you start?

Tools: Draw.io, Figma, ···

RAP PRINCIPLES 1: AUTOMATION - MODULAR CODING

- Automation (*as much as you can*)

↪ Where would you start?

Tools: Draw.io, Figma, ···

- Reusable (modular) code

RAP principles 1: Automation - Modular coding

- Automation (*as much as you can*)

↪ Where would you start?

Tools: Draw.io, Figma, ···

- Reusable (modular) code

↪ What does this means for your project?

RAP PRINCIPLES 1: AUTOMATION - MODULAR CODING

- Automation (*as much as you can*)

↪ Where would you start?

Tools: Draw.io, Figma, ···

- Reusable (modular) code

↪ What does this means for your project?

↪ Example

RAP principles 1: Automation - Modular coding

- Automation (*as much as you can*)

↪ Where would you start?

Tools: Draw.io, Figma, ···

- Reusable (modular) code

↪ What does this means for your project?

↪ Example

- Exercise part 1 (15 minutes)

RAP principles 1: Automation - Modular coding

- Automation (*as much as you can*)

↪ Where would you start?

Tools: Draw.io, Figma, ···

- Reusable (modular) code

↪ What does this means for your project?

↪ Example

- Exercise part 1 (15 minutes)
  - *Look at a notebook that you developed for this course - and see how to separate out what you did into functions.*

RAP PRINCIPLES 1: AUTOMATION - MODULAR CODING

- Automation (*as much as you can*)

↪ Where would you start?

Tools: Draw.io, Figma, ···

- Reusable (modular) code

↪ What does this means for your project?

↪ Example

- Exercise part 1 (15 minutes)
  - *Look at a notebook that you developed for this course - and see how to separate out what you did into functions.*
  - *Create functions, use these functions in a main document*

RAP PRINCIPLES 2:
TRANSPARENCY - OPEN SOURCE - VERSION CONTROL

RAP PRINCIPLES 2:
TRANSPARENCY - OPEN SOURCE - VERSION CONTROL

- Transparency

RAP PRINCIPLES 2:
TRANSPARENCY - OPEN SOURCE - VERSION CONTROL

- Transparency

↪ What are the benefits of working in a transparent way?

RAP PRINCIPLES 2: TRANSPARENCY - OPEN SOURCE - VERSION CONTROL

- Transparency
  - ↪ What are the benefits of working in a transparent way?
- Use open source tools

RAP PRINCIPLES 2: TRANSPARENCY - OPEN SOURCE - VERSION CONTROL

- Transparency

↪ What are the benefits of working in a transparent way?

- Use open source tools

↪ Compare Open source *vs* proprietary software

RAP PRINCIPLES 2:
TRANSPARENCY - OPEN SOURCE - VERSION CONTROL

- Transparency

↪ What are the benefits of working in a transparent way?

- Use open source tools

↪ Compare Open source *vs* proprietary software

Tools: Python, R, Git, GitHub(GitLab)

RAP PRINCIPLES 2: TRANSPARENCY - OPEN SOURCE - VERSION CONTROL

- Transparency

↪ What are the benefits of working in a transparent way?

- Use open source tools

↪ Compare Open source *vs* proprietary software

Tools: Python, R, Git, GitHub(GitLab)

- Version control

RAP PRINCIPLES 2: TRANSPARENCY - OPEN SOURCE - VERSION CONTROL

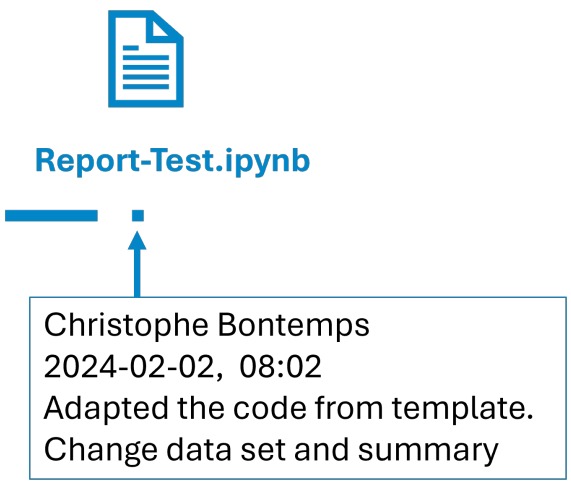
- Transparency
  - ↪ What are the benefits of working in a transparent way?
- Use open source tools
  - ↪ Compare Open source *vs* proprietary software

Tools: Python, R, Git, GitHub(GitLab)

- Version control
- Exercise part 2 (Afternoon session)

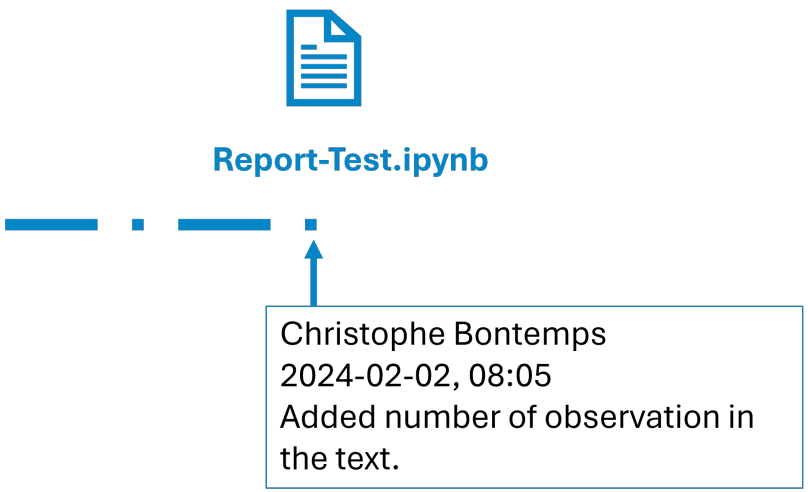
Version Control Main commands

commit for every change!



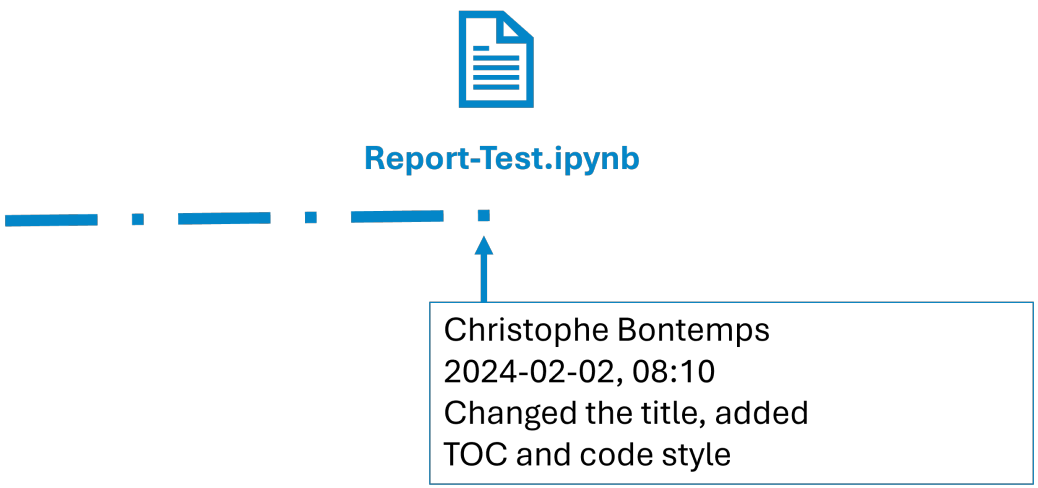
Version Control Main commands

commit for every change!



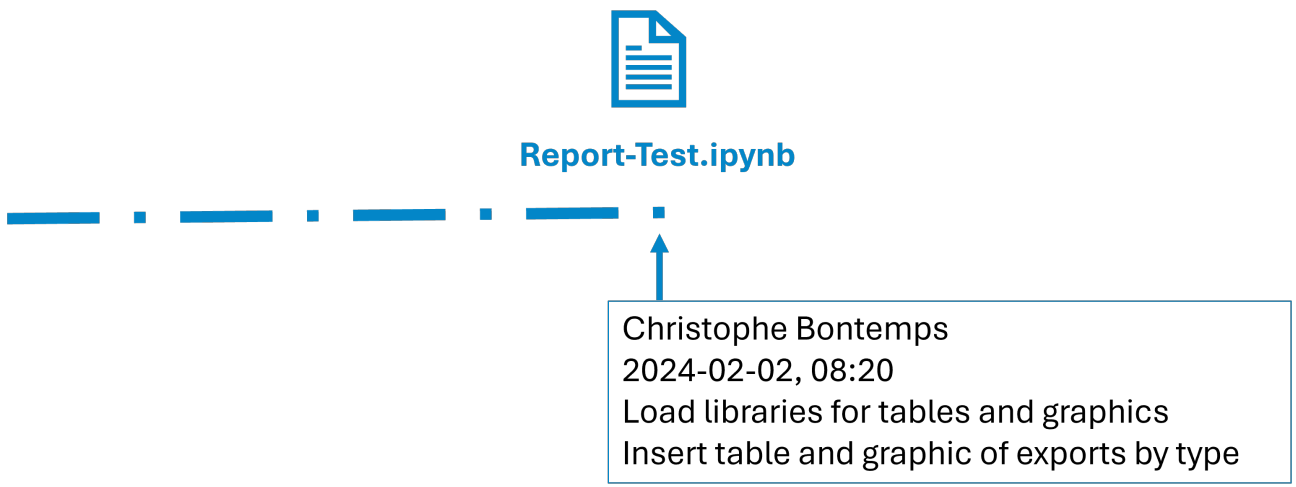
Version Control Main commands

commit for every change!



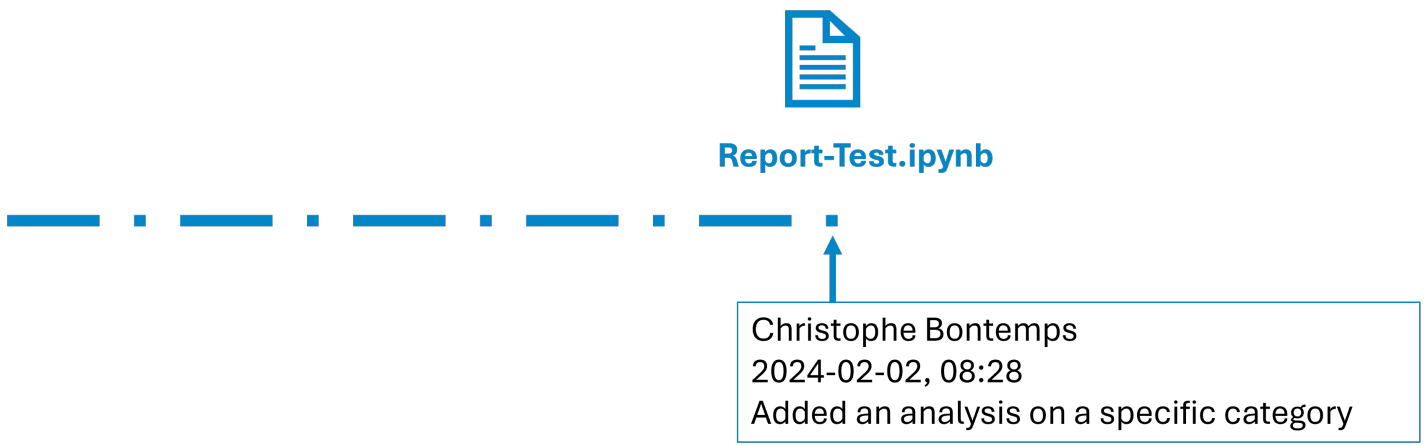
VERSION CONTROL MAIN COMMANDS

commit for every change!



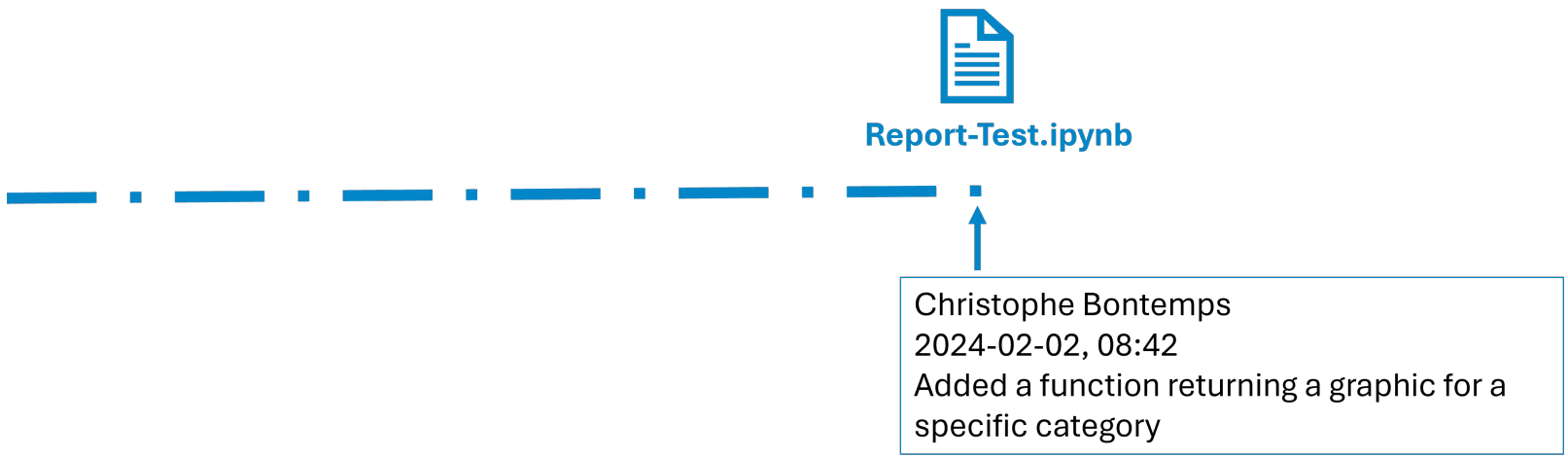
VERSION CONTROL MAIN COMMANDS

commit for every change!



Version Control Main commands

commit for every change!



VERSION CONTROL MAIN COMMANDS

commit for every change!

THE HISTORY OF THE FILE IS RECORDED!

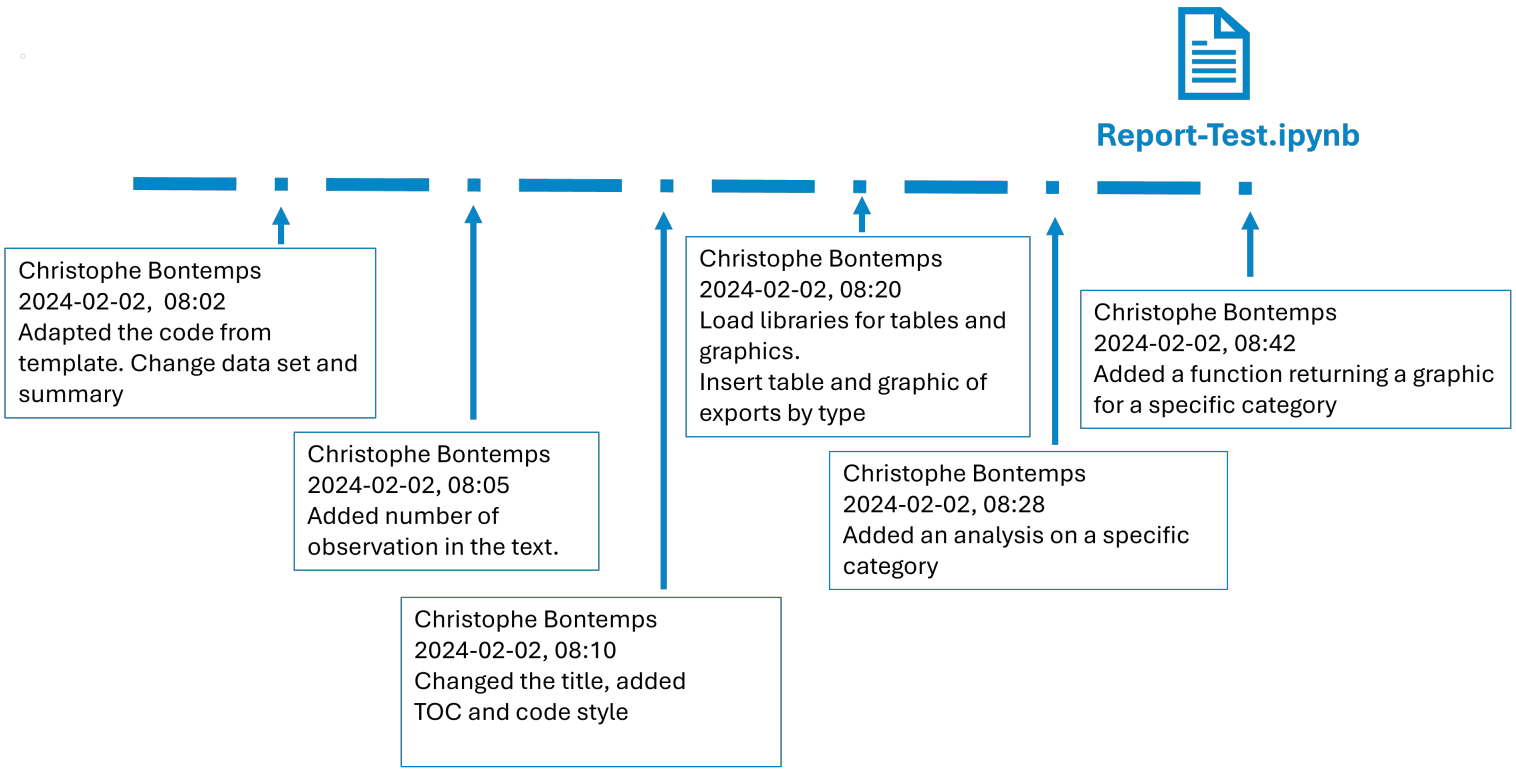
Each version is documented (with *commits*)

THE HISTORY OF THE FILE IS RECORDED!

Each version is documented (with *commits*)

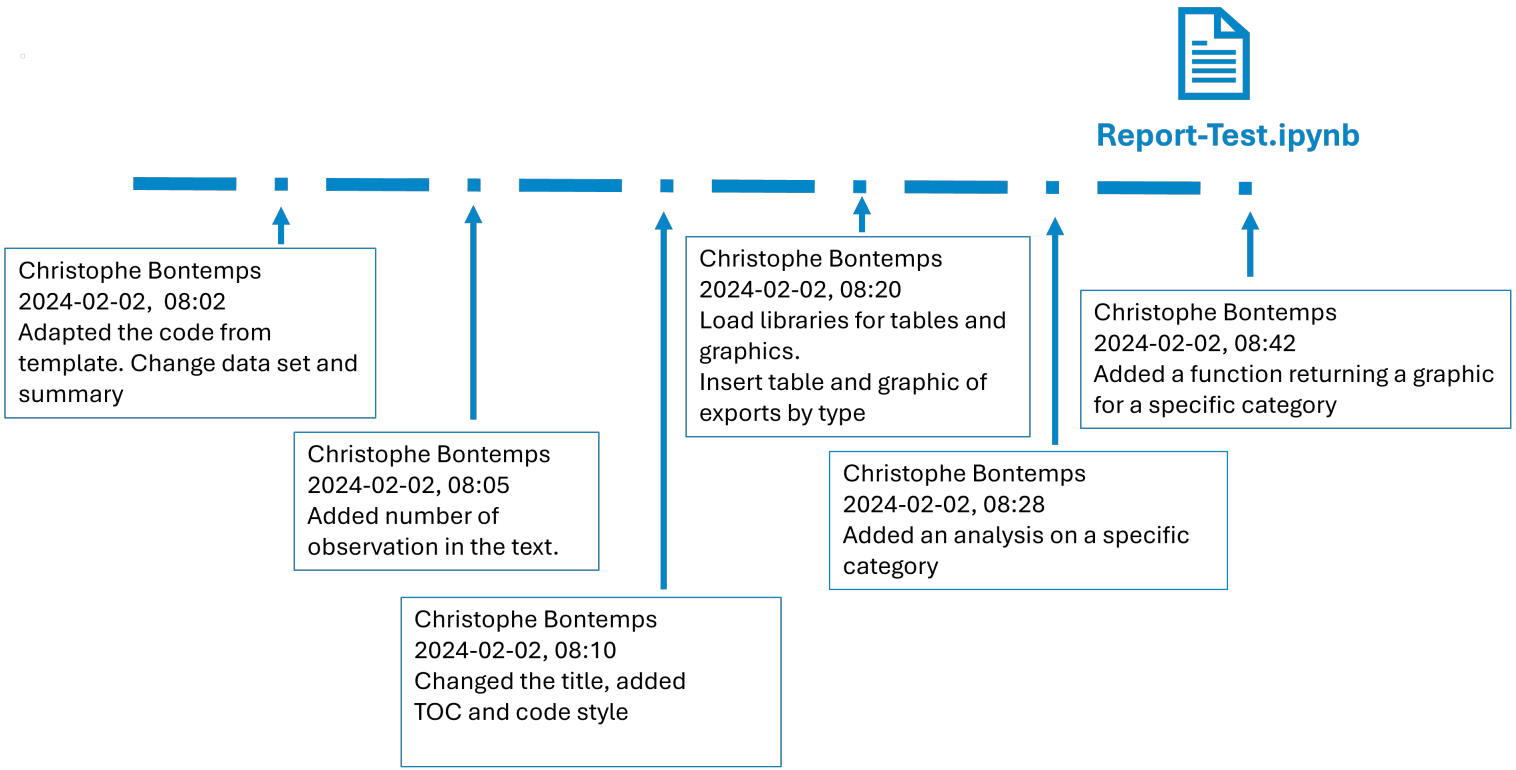
THE HISTORY OF THE FILE IS RECORDED!

Each version embeds the full history!



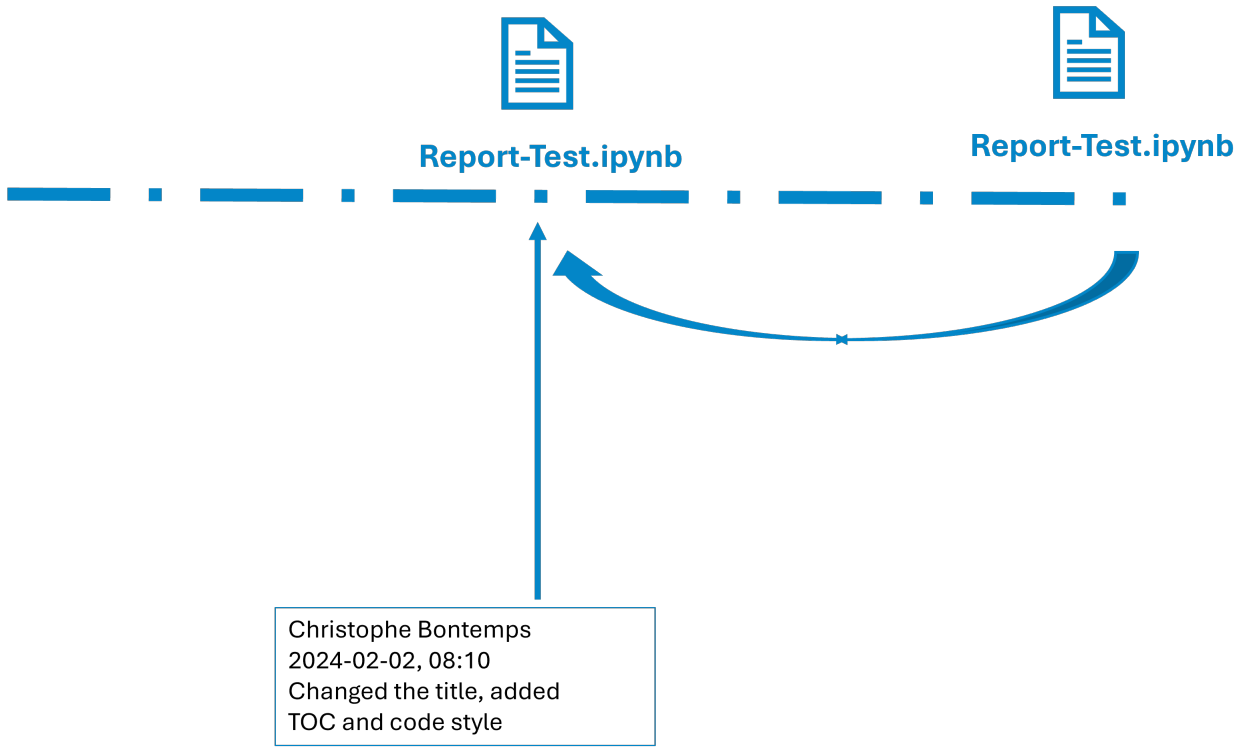
THE HISTORY OF THE FILE IS RECORDED!

Each version embeds the full history!



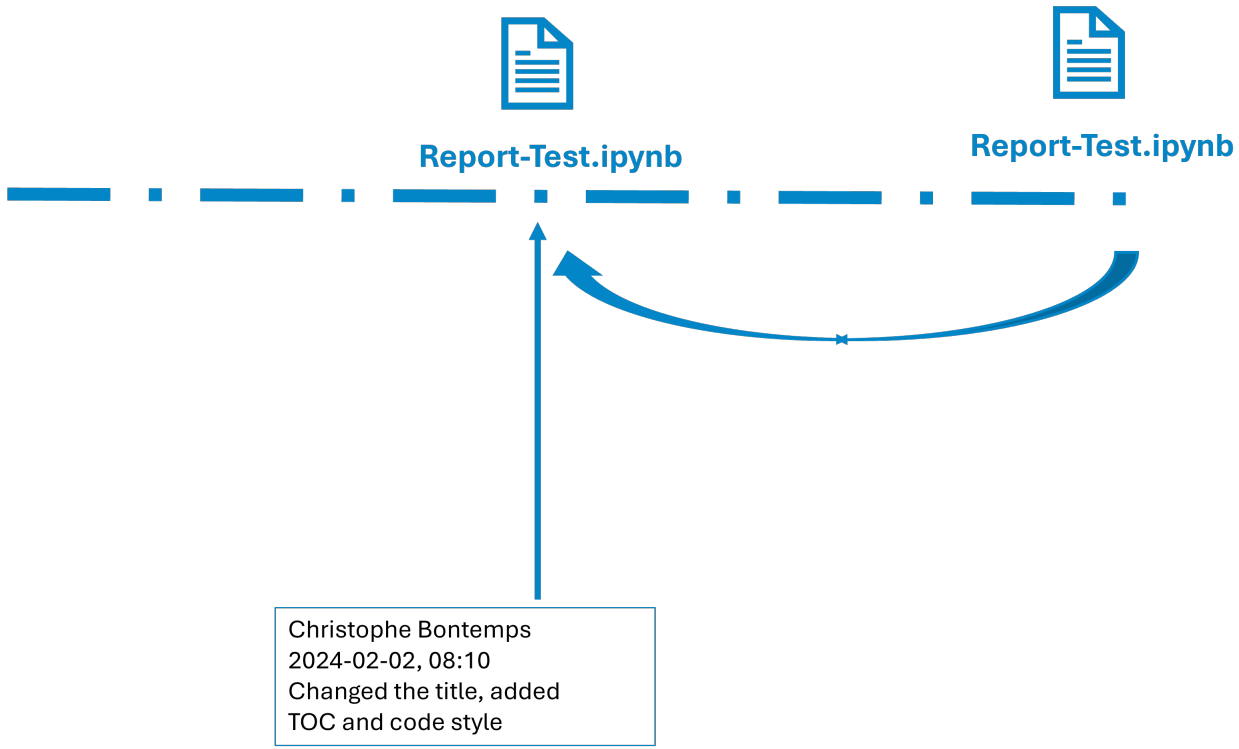
GOING BACK AND "UNDO" IS POSSIBLE

It is possible to review previous version...



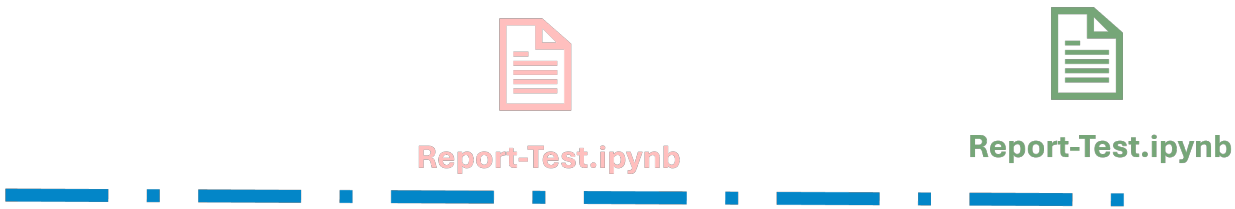
GOING BACK AND "UNDO" IS POSSIBLE

It is possible to review previous version...

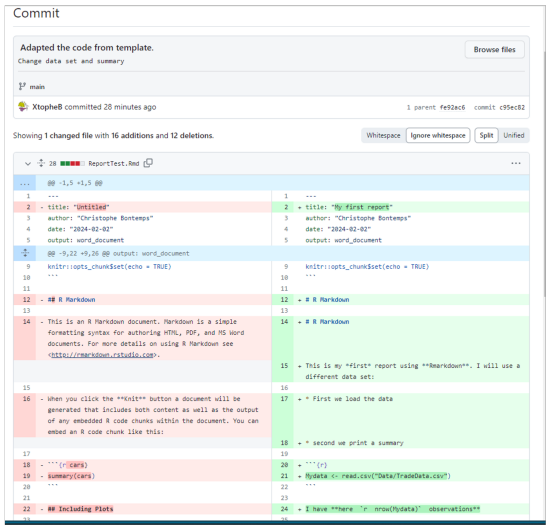


GOING BACK AND "UNDO" IS POSSIBLE

...to compare the changes...

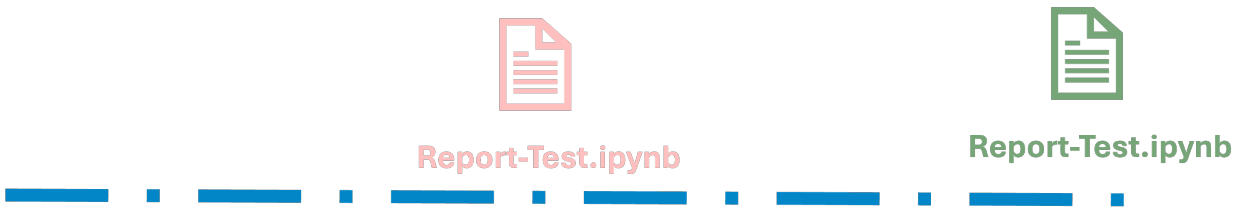


Christophe Bontemps
2024-02-02, 08:10
Changed the title, added
TOC and code style

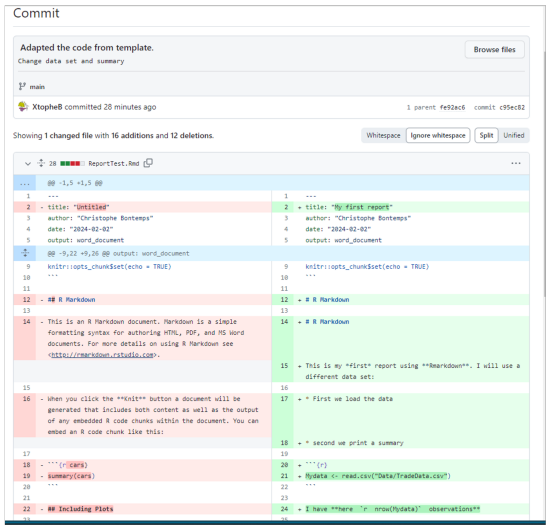


GOING BACK AND "UNDO" IS POSSIBLE

...to compare the changes...

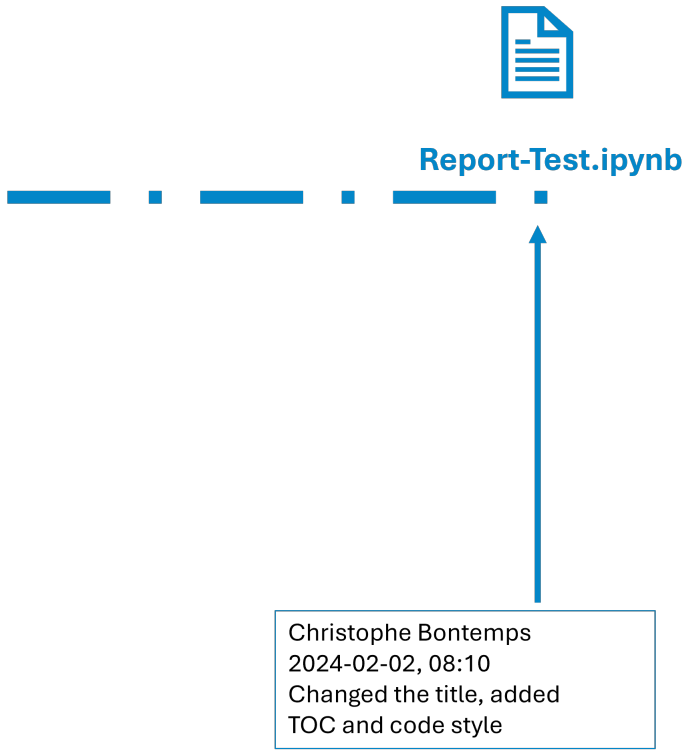


Christophe Bontemps
2024-02-02, 08:10
Changed the title, added
TOC and code style



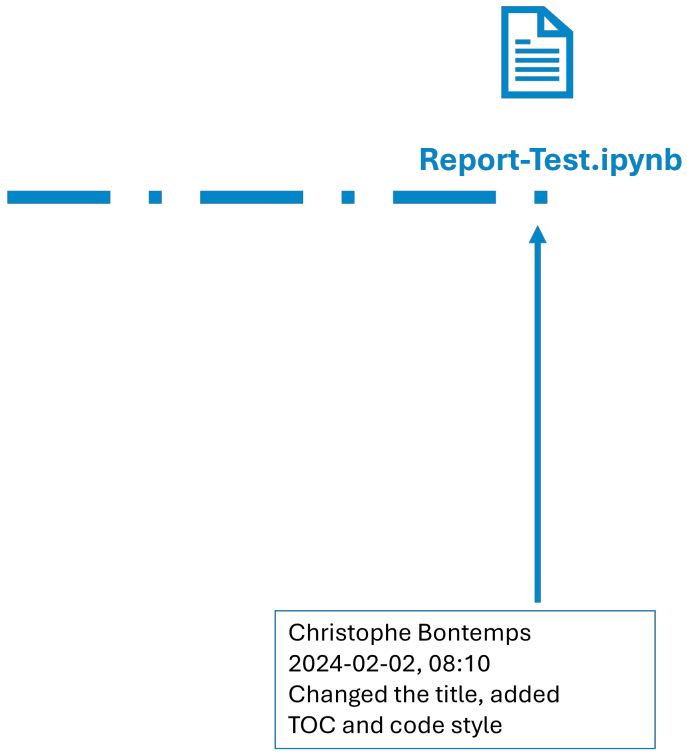
GOING BACK AND "UNDO" IS POSSIBLE

... and to revert to a previous version...



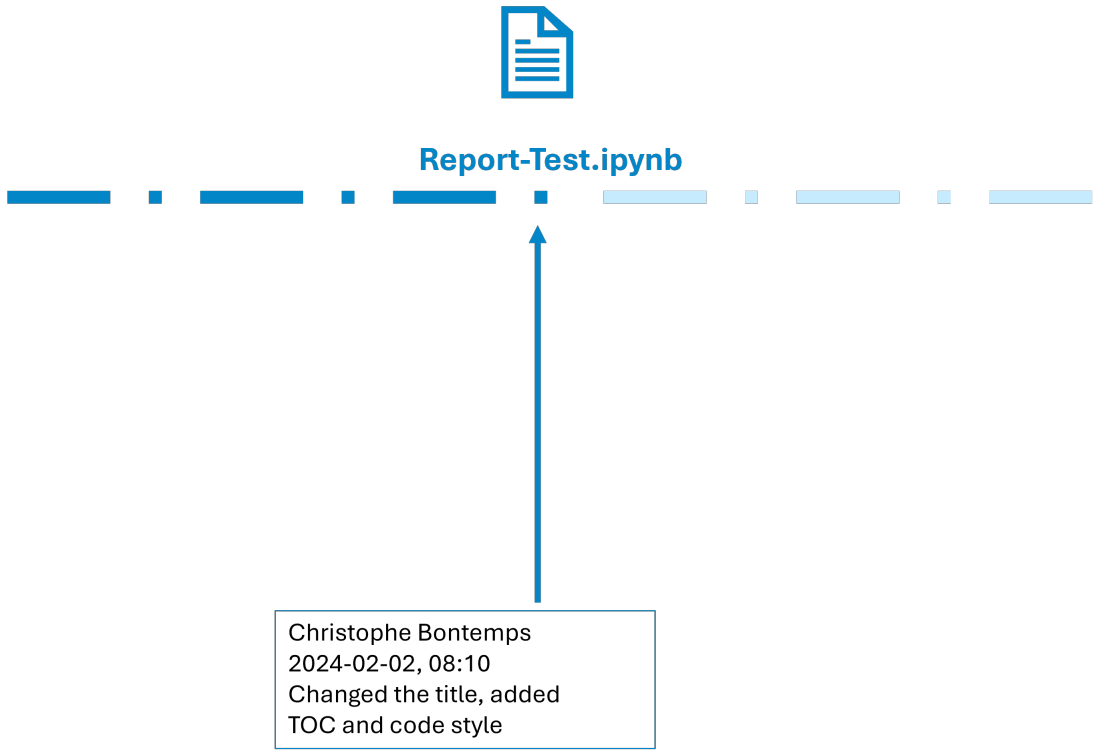
GOING BACK AND "UNDO" IS POSSIBLE

... and to revert to a previous version...



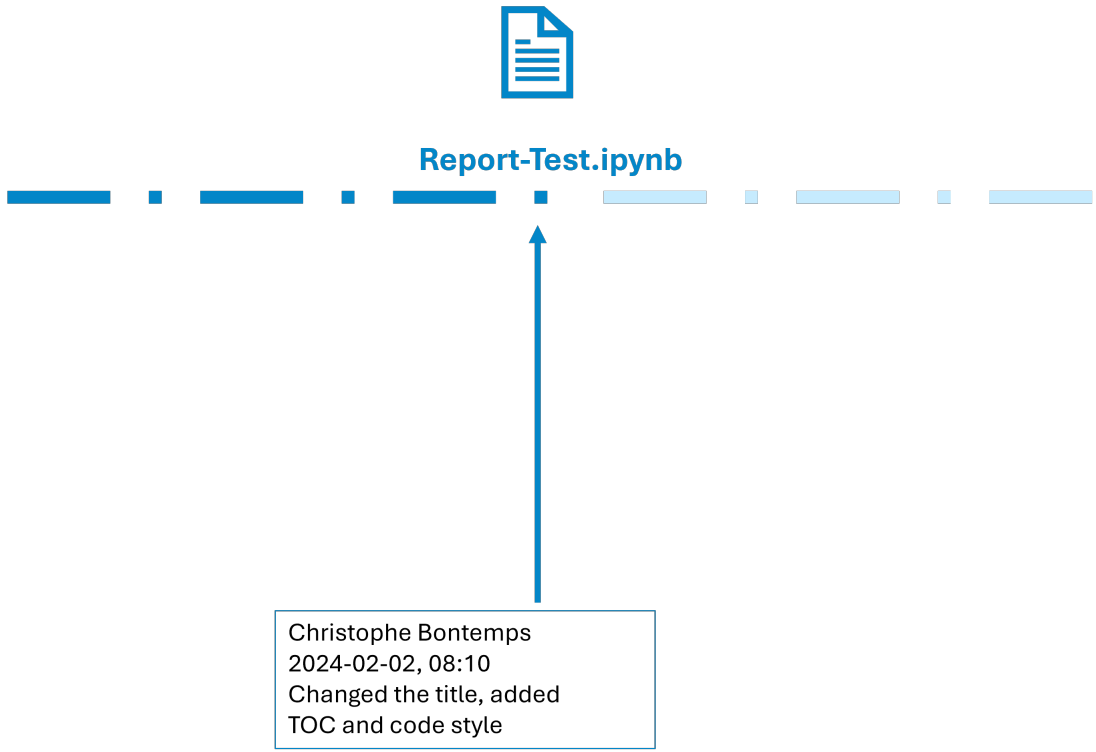
GOING BACK AND "UNDO" IS POSSIBLE

... or *undo* as if nothing happened



GOING BACK AND "UNDO" IS POSSIBLE

... or *undo* as if nothing happened



GIT FOR COLLABORATING

Alice and **Bob** work on the same file

GIT FOR COLLABORATING

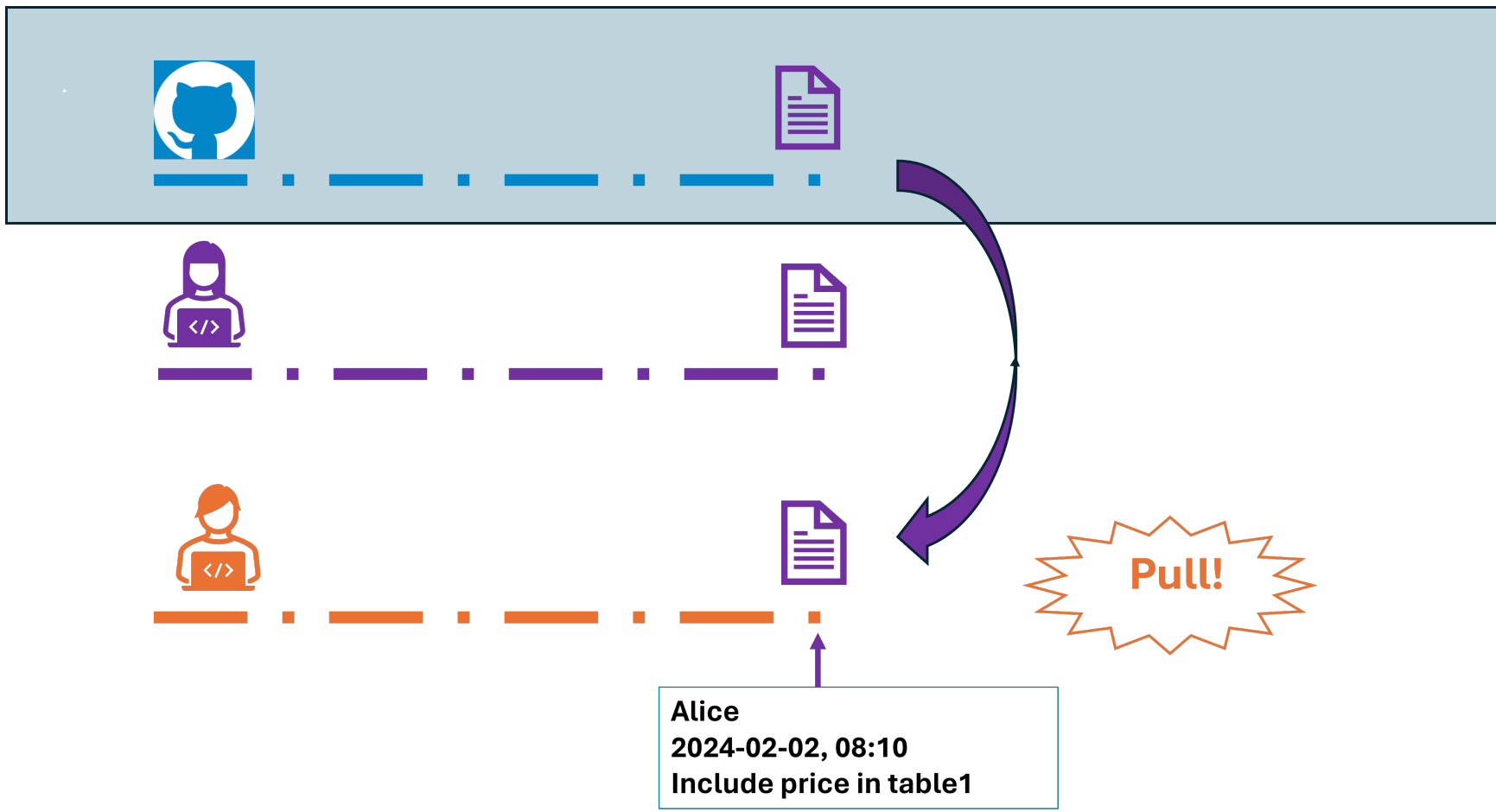
Alice and **Bob** work on the same file

GIT FOR COLLABORATING

Alice and **Bob** work on the same file

GIT FOR COLLABORATING

Alice and **Bob** work on the same file



Important notions

Spaces

IMPORTANT NOTIONS

Spaces

- **Local working directory:**

Important notions

Spaces

- **Local working directory:**

↪ Folder with your file(s)

IMPORTANT NOTIONS

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

IMPORTANT NOTIONS

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

Important notions

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

- **Remote repository:**

IMPORTANT NOTIONS

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

- **Remote repository:**

↪ Folder on GitHub platform

IMPORTANT NOTIONS

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

- **Remote repository:**

↪ Folder on GitHub platform

Actions

IMPORTANT NOTIONS

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

- **Remote repository:**

↪ Folder on GitHub platform

Actions

- **Commit:**

IMPORTANT NOTIONS

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

- **Remote repository:**

↪ Folder on GitHub platform

Actions

- **Commit:**

↪ A snapshot of file changes

Important notions

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

- **Remote repository:**

↪ Folder on GitHub platform

Actions

- **Commit:**

↪ A snapshot of file changes

- **Commit message:**

Important notions

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

- **Remote repository:**

↪ Folder on GitHub platform

Actions

- **Commit:**

↪ A snapshot of file changes

- **Commit message:**

↪ A concise description of the changes made

Important notions

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

- **Remote repository:**

↪ Folder on GitHub platform

Actions

- **Commit:**

↪ A snapshot of file changes

- **Commit message:**

↪ A concise description of the changes made

- **Push:**

Important notions

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

- **Remote repository:**

↪ Folder on GitHub platform

Actions

- **Commit:**

↪ A snapshot of file changes

- **Commit message:**

↪ A concise description of the changes made

- **Push:**

↪ Action of sending modified file(s) to GitHub

Important notions

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

- **Remote repository:**

↪ Folder on GitHub platform

Actions

- **Commit:**

↪ A snapshot of file changes

- **Commit message:**

↪ A concise description of the changes made

- **Push:**

↪ Action of sending modified file(s) to GitHub

- **Pull:**

IMPORTANT NOTIONS

Spaces

- **Local working directory:**

↪ Folder with your file(s)

- **Staging area:**

↪ Local "space" where modified file(s) are stored before commit

- **Remote repository:**

↪ Folder on GitHub platform

Actions

- **Commit:**

↪ A snapshot of file changes

- **Commit message:**

↪ A concise description of the changes made

- **Push:**

↪ Action of sending modified file(s) to GitHub

- **Pull:**

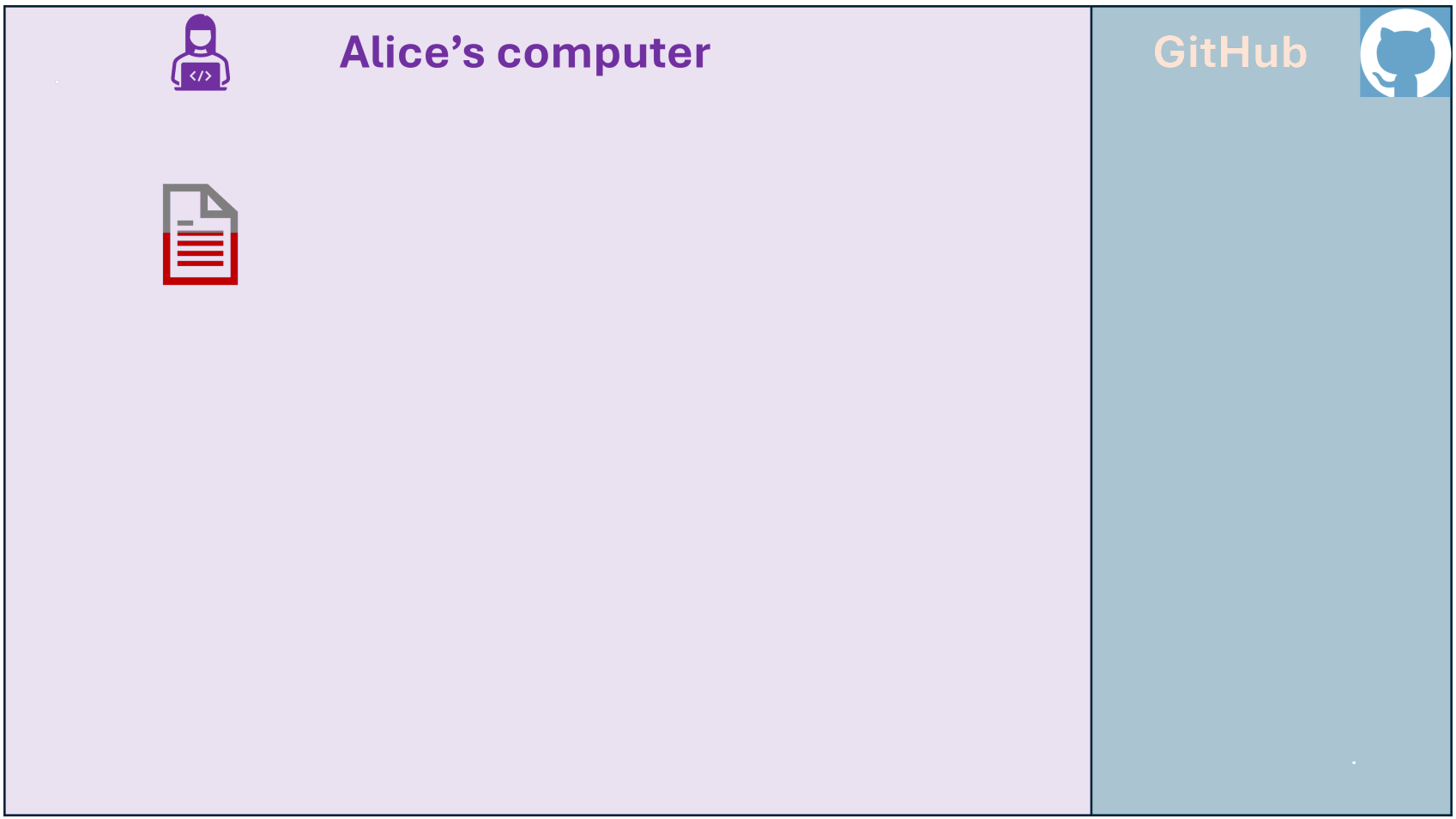
↪ Action of retrieving modified file(s) to local working directory

Git for collaborating: Details

Alice and **Bob** work on the same file (Details)

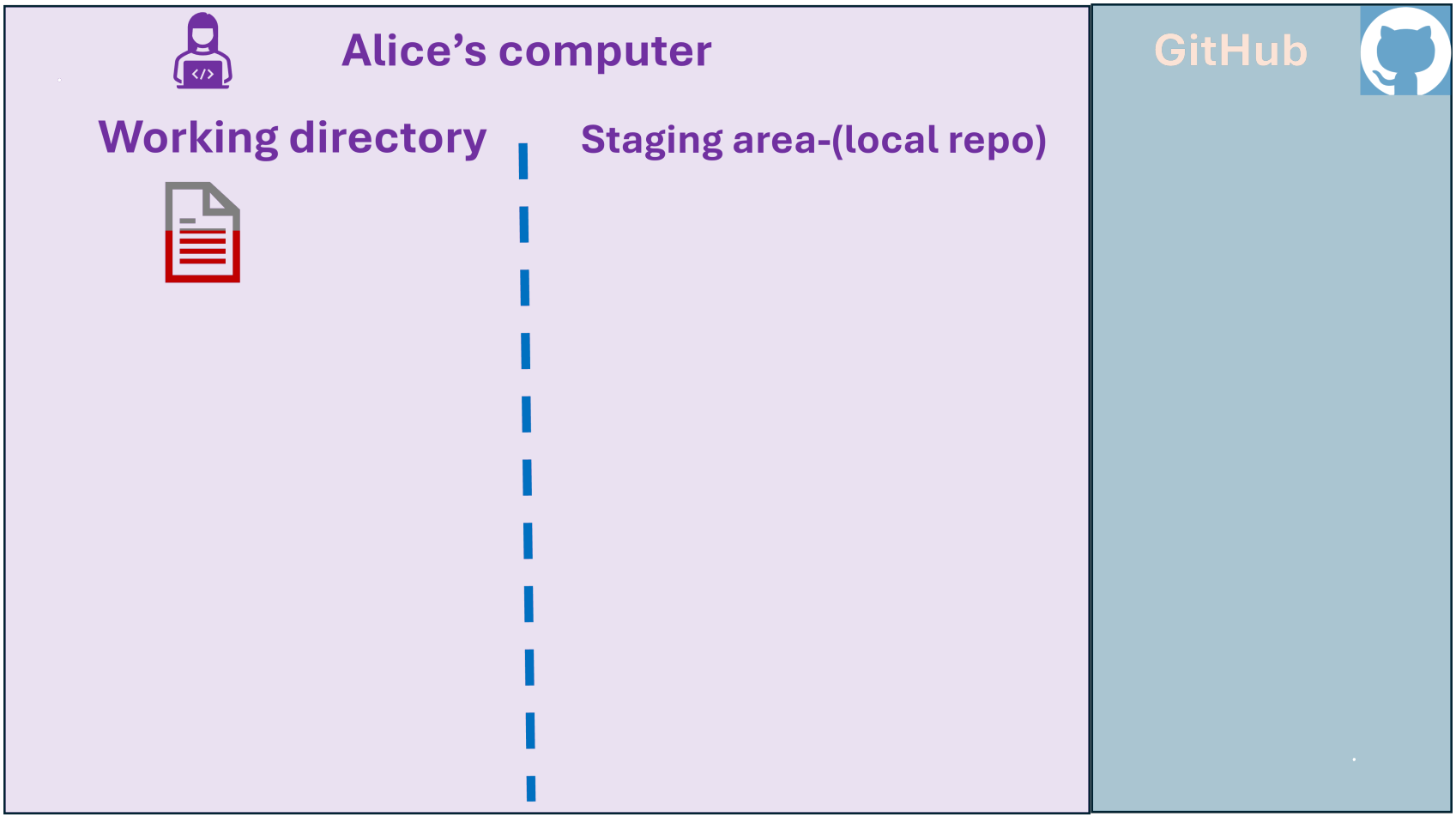
GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



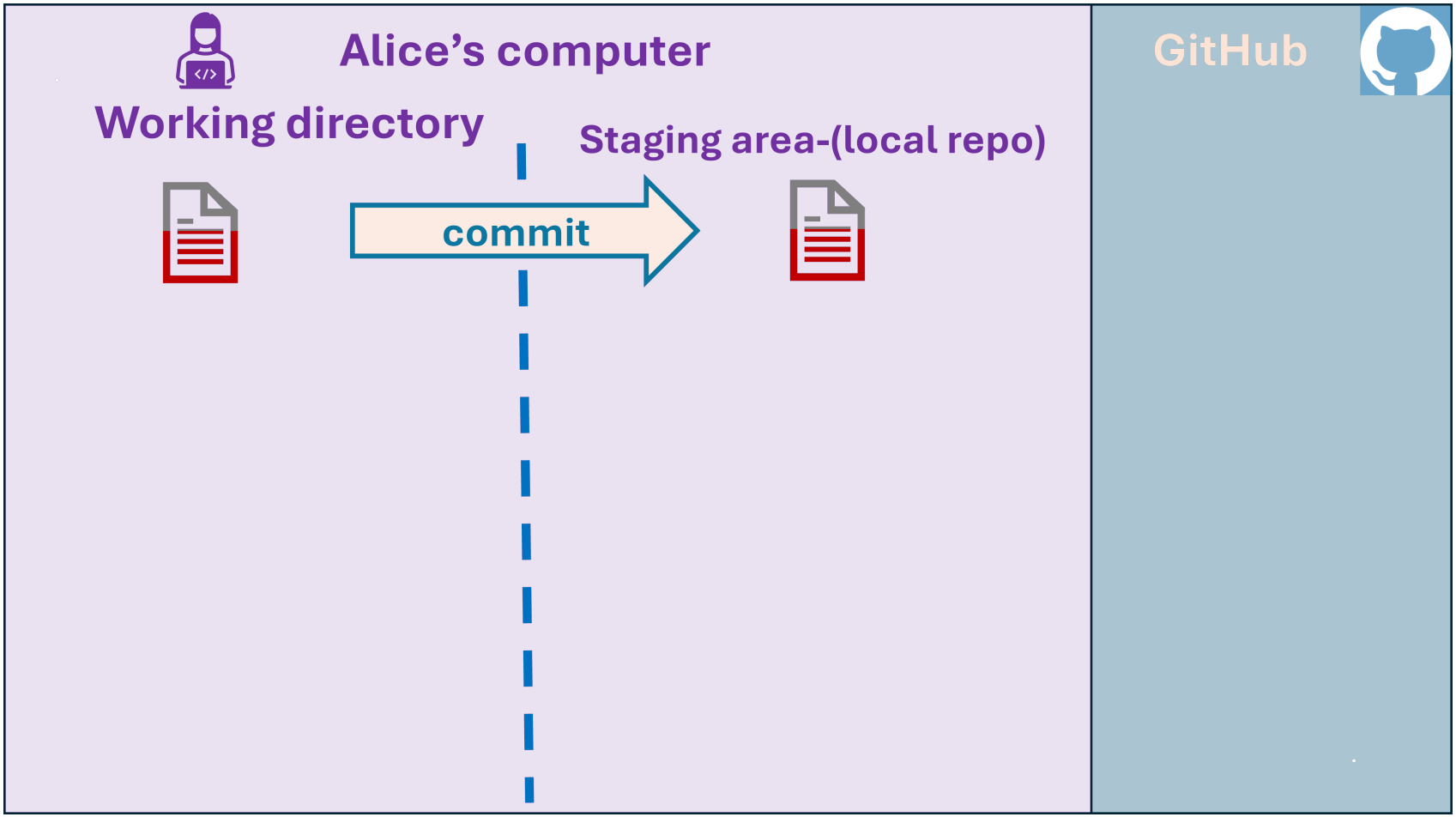
GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



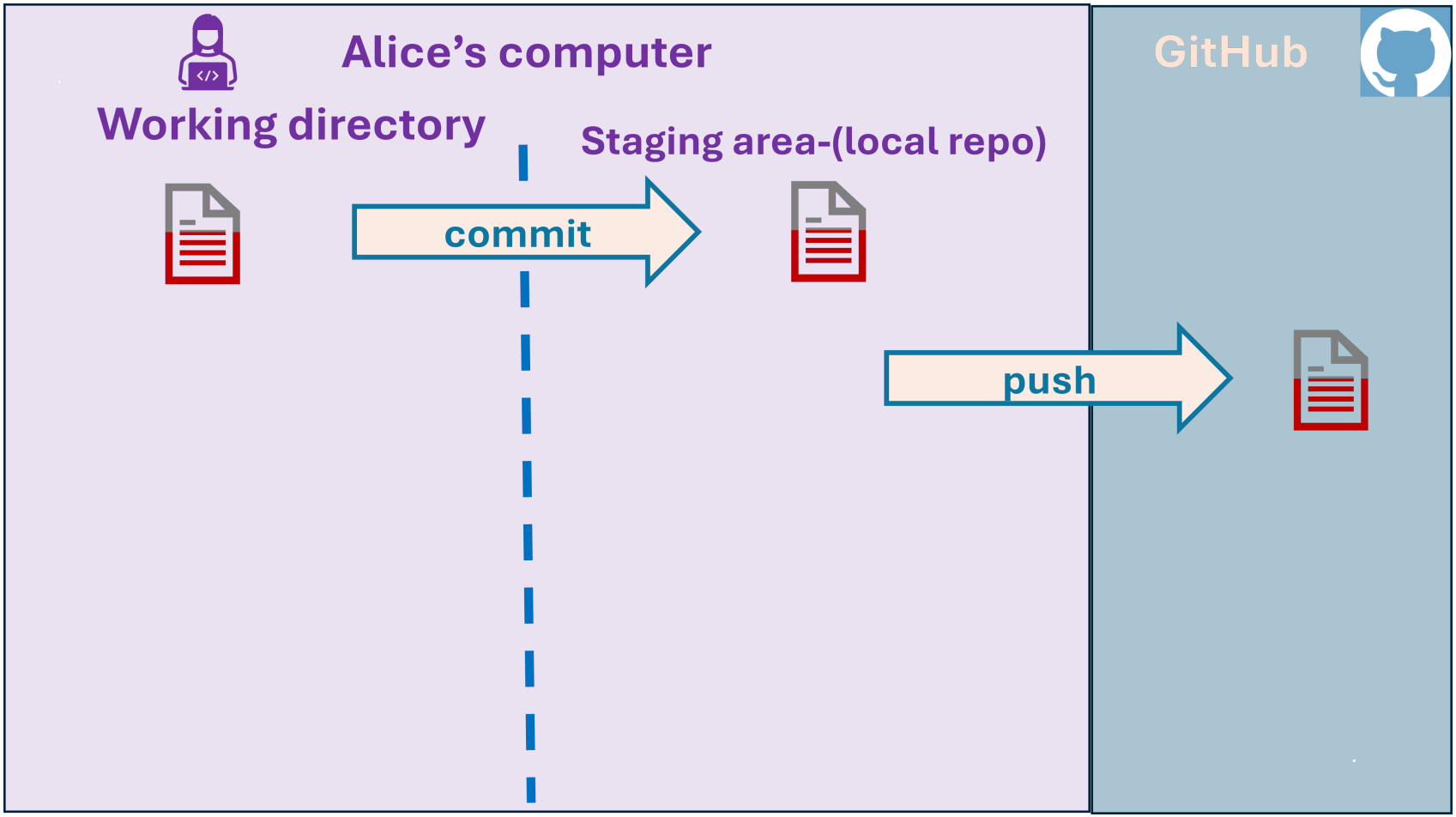
GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



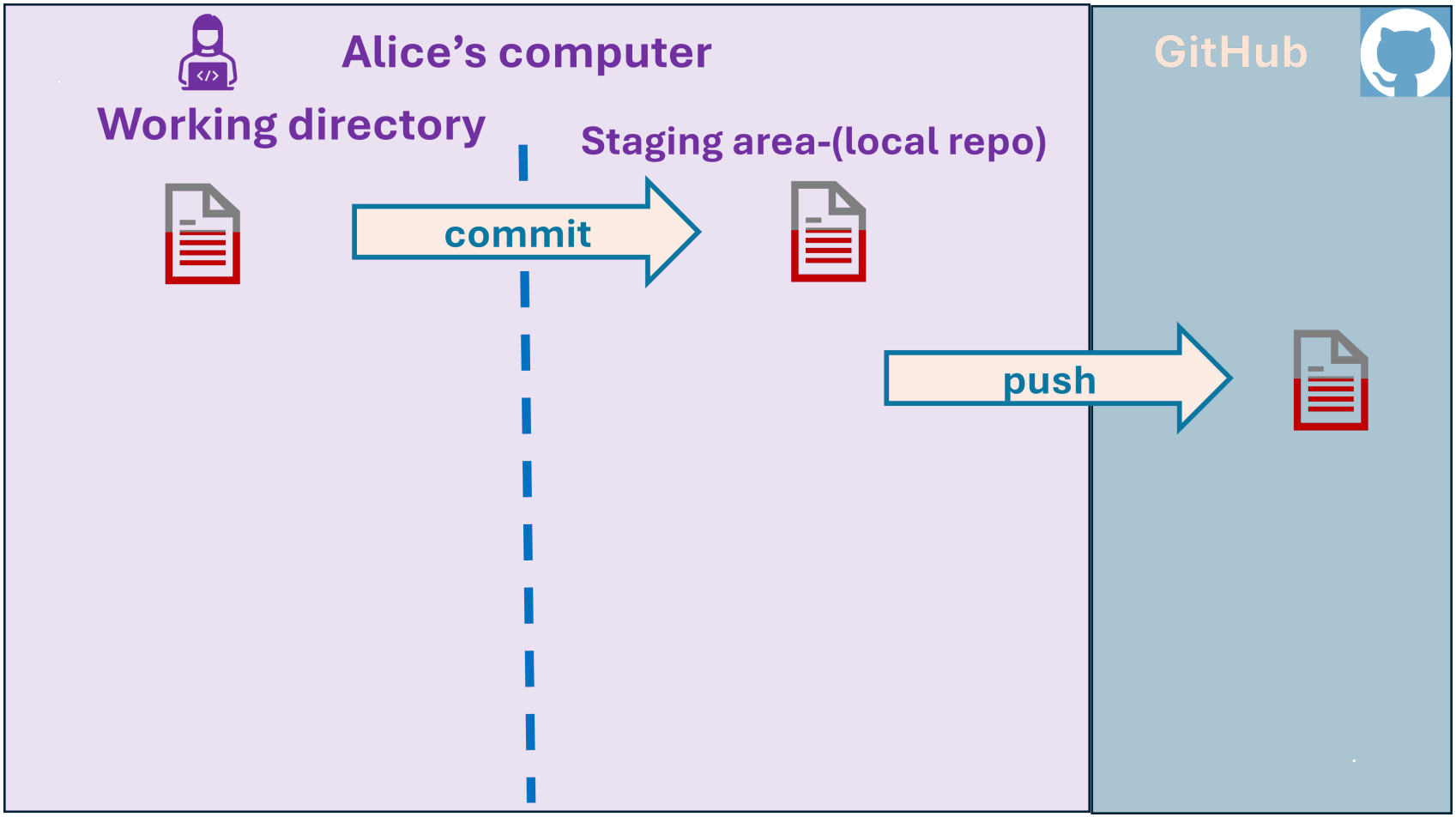
GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



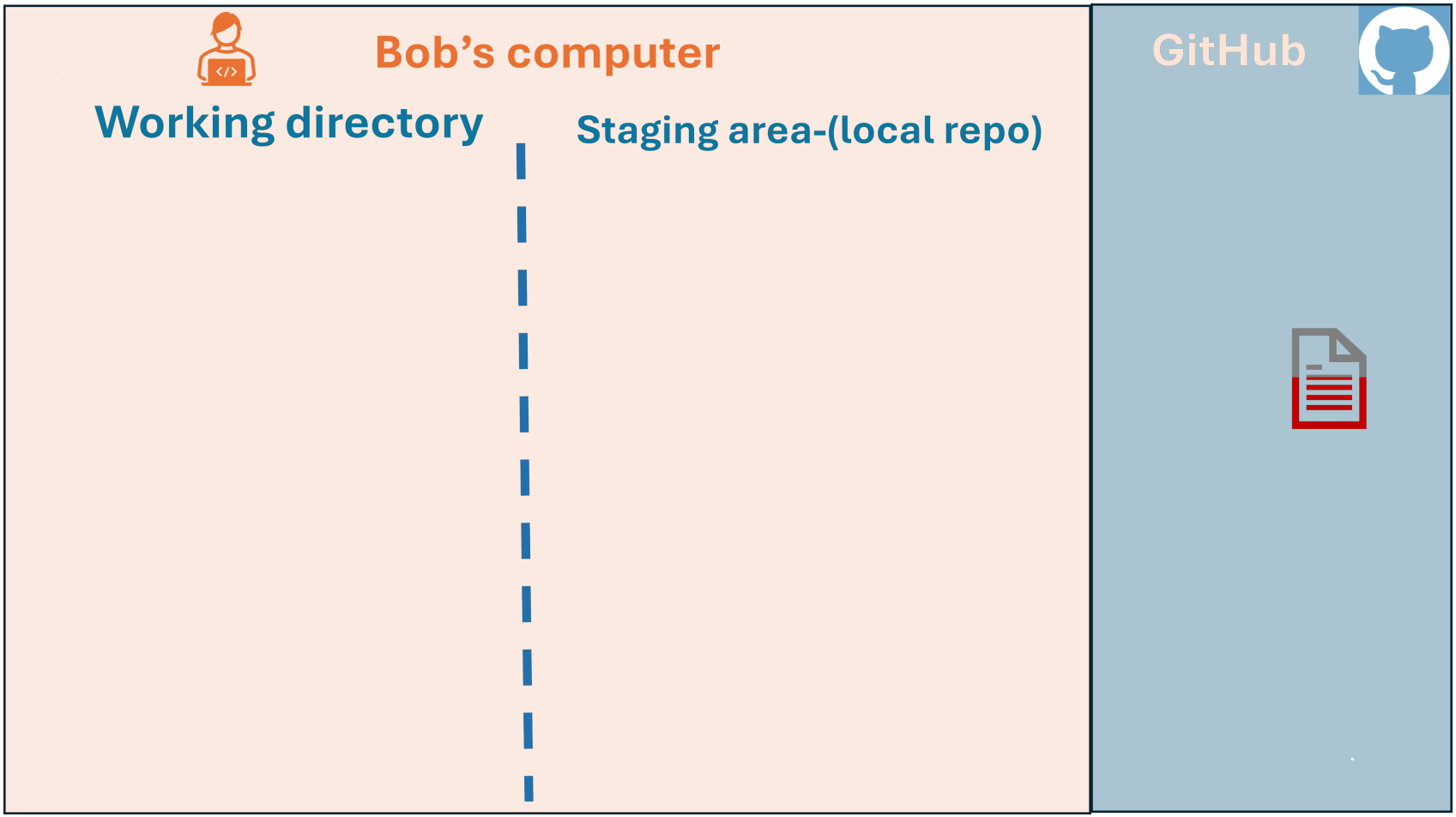
GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



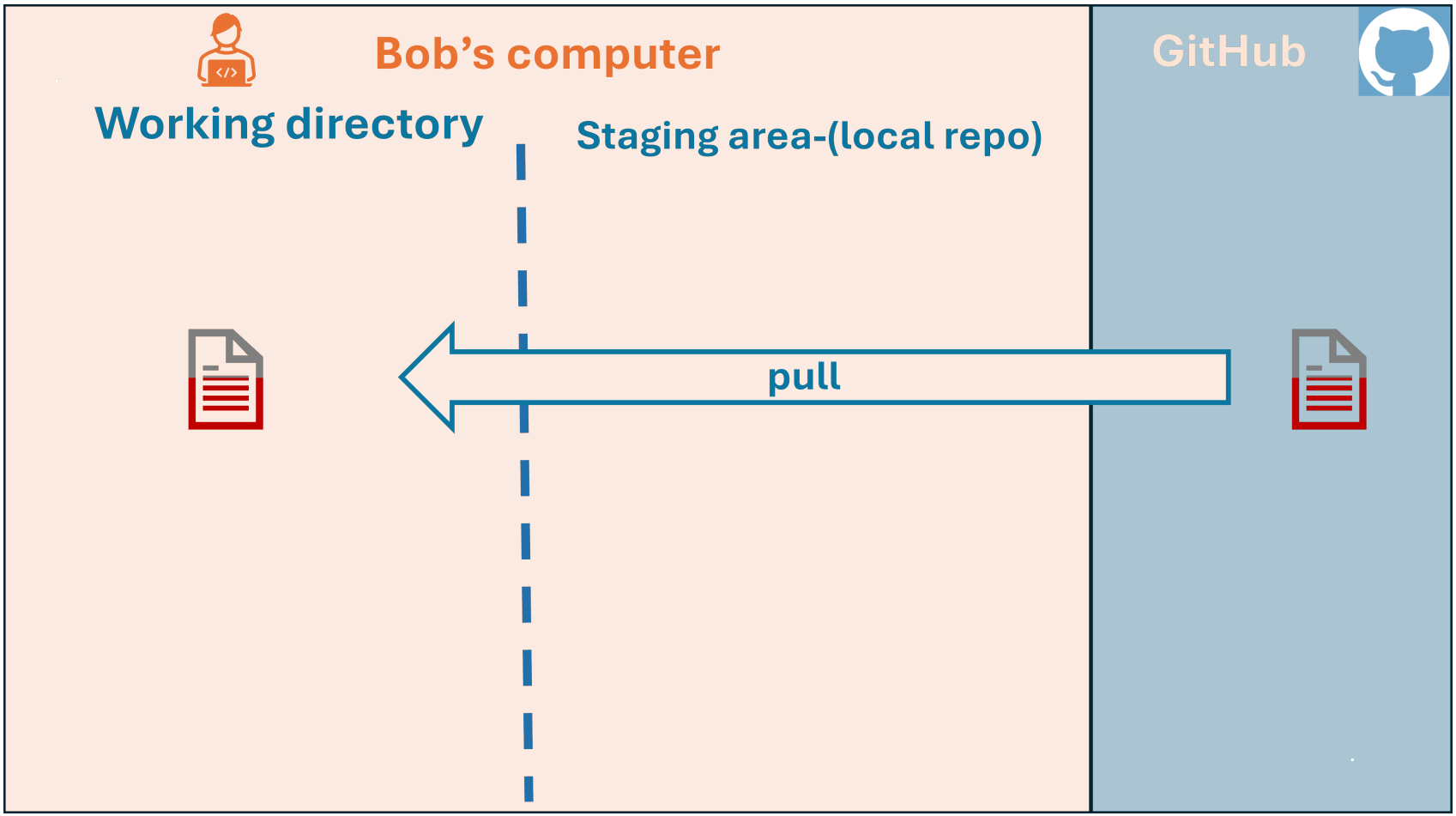
GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



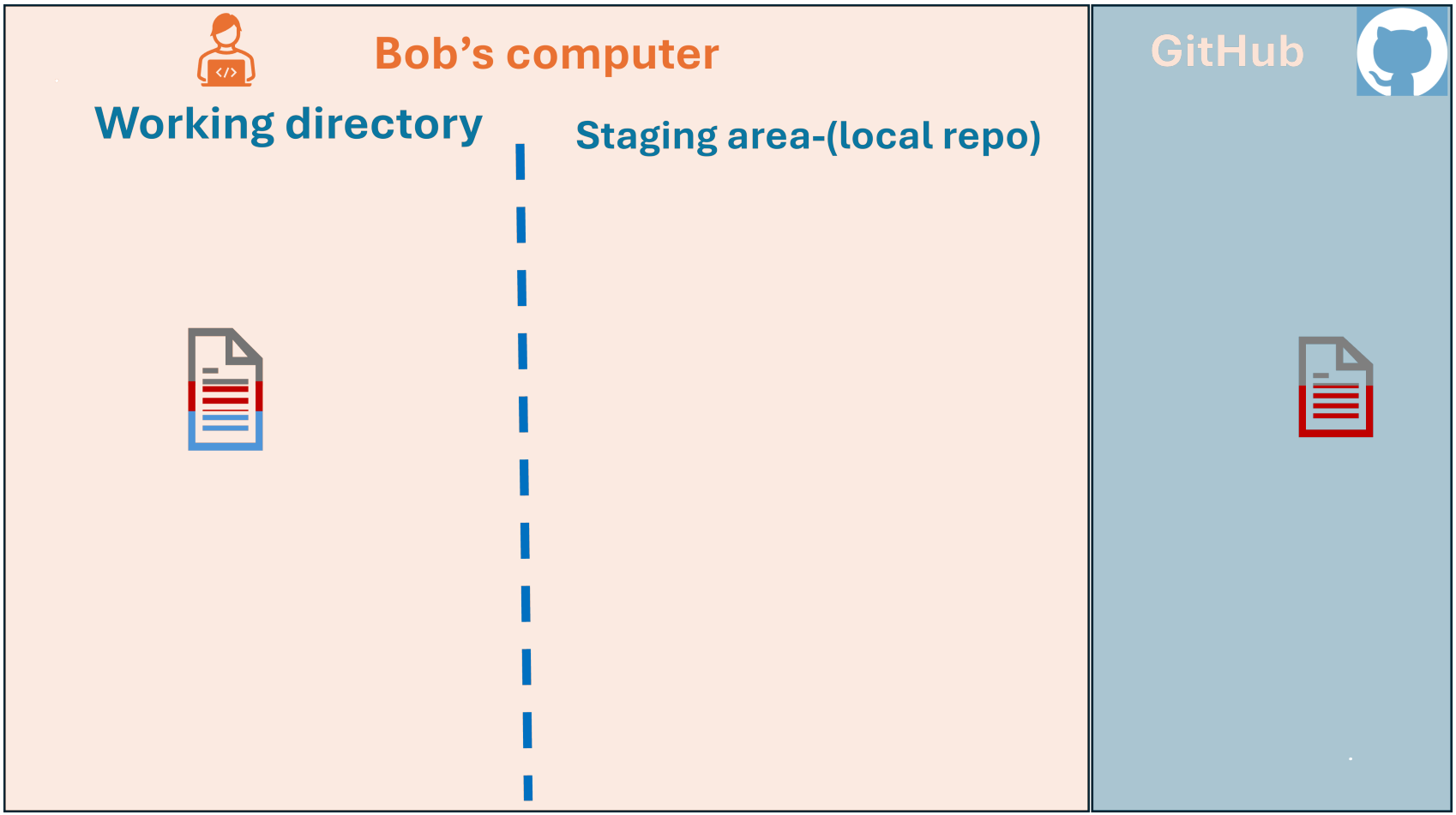
GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



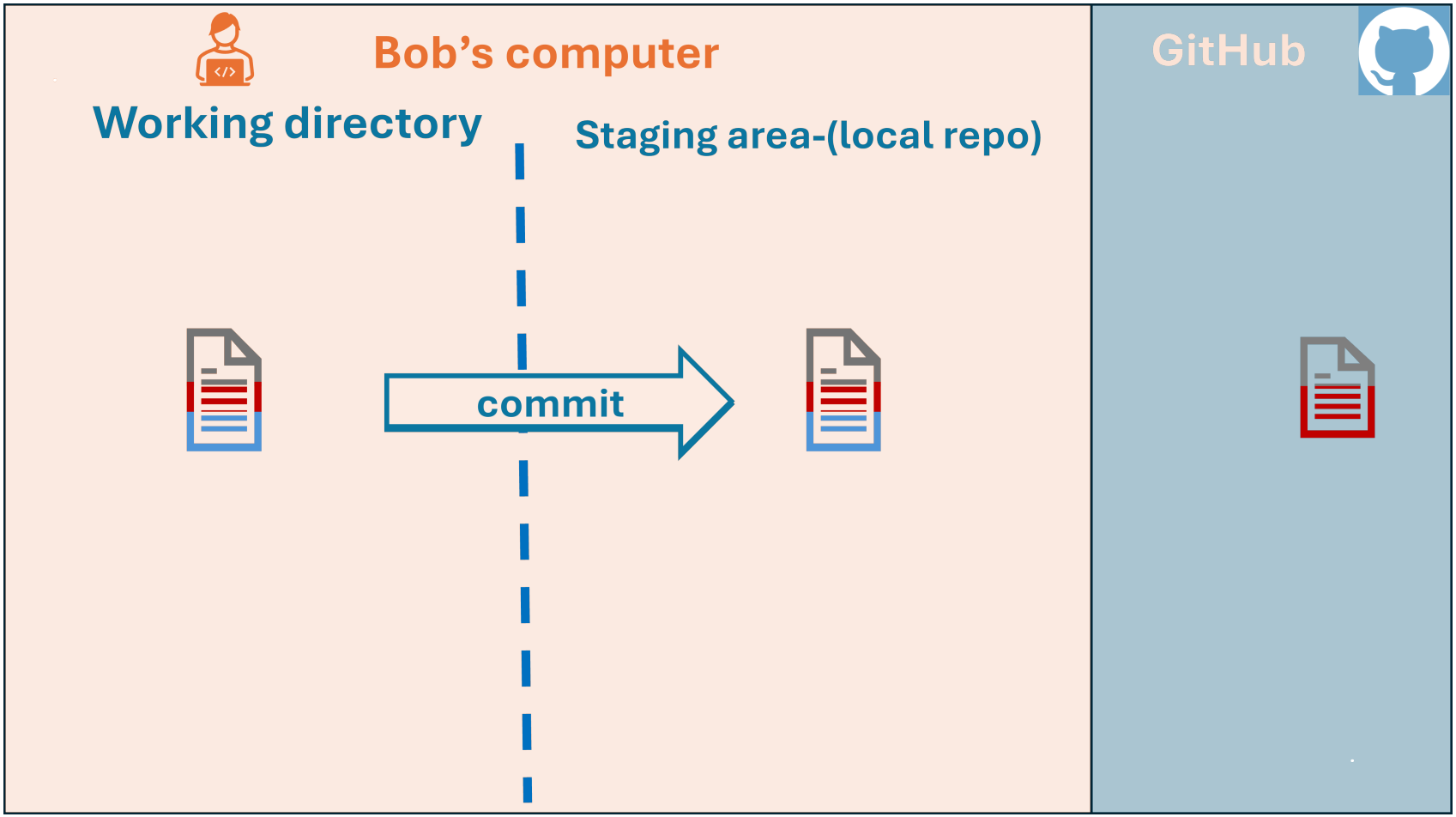
GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



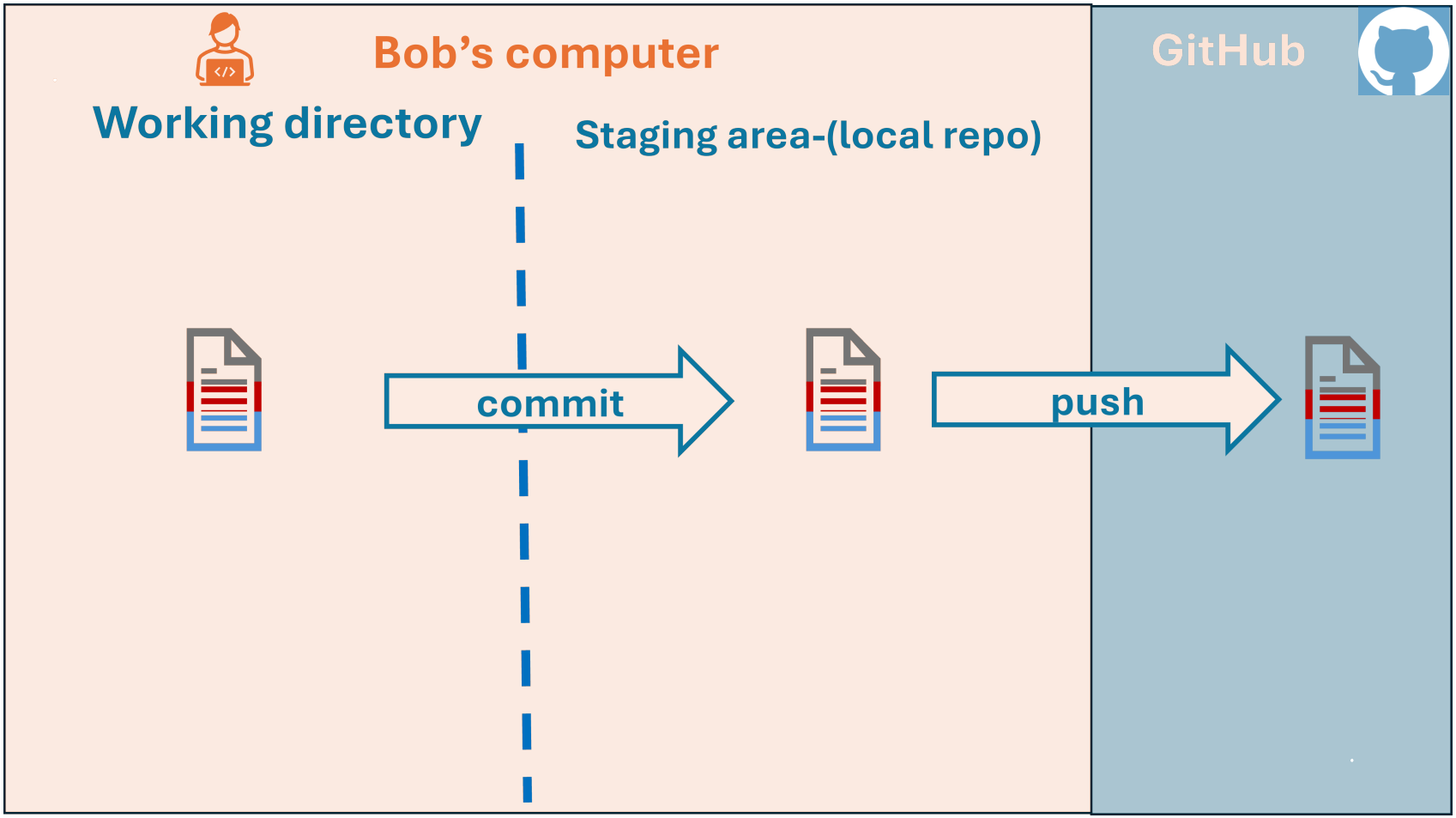
GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



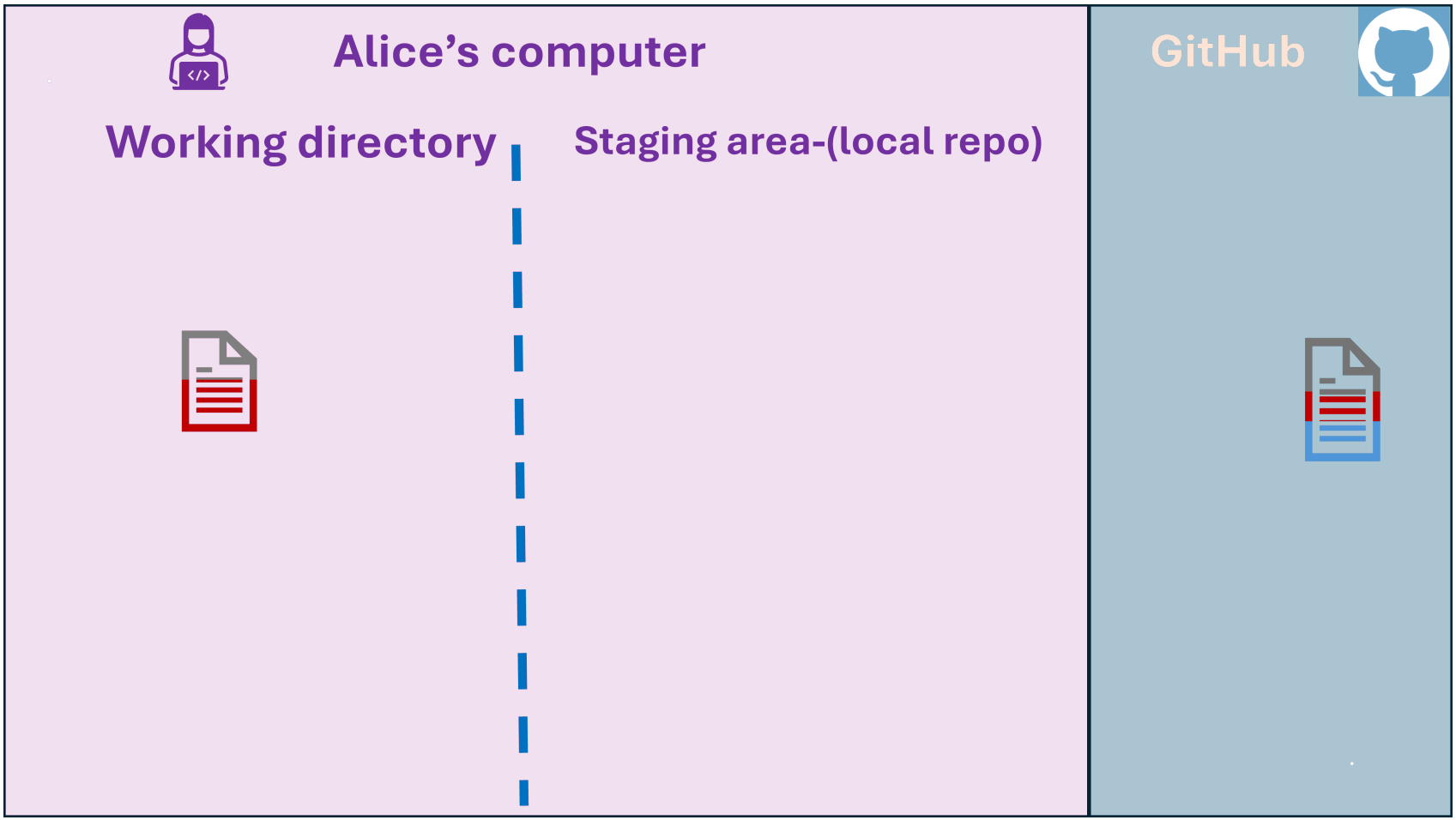
GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



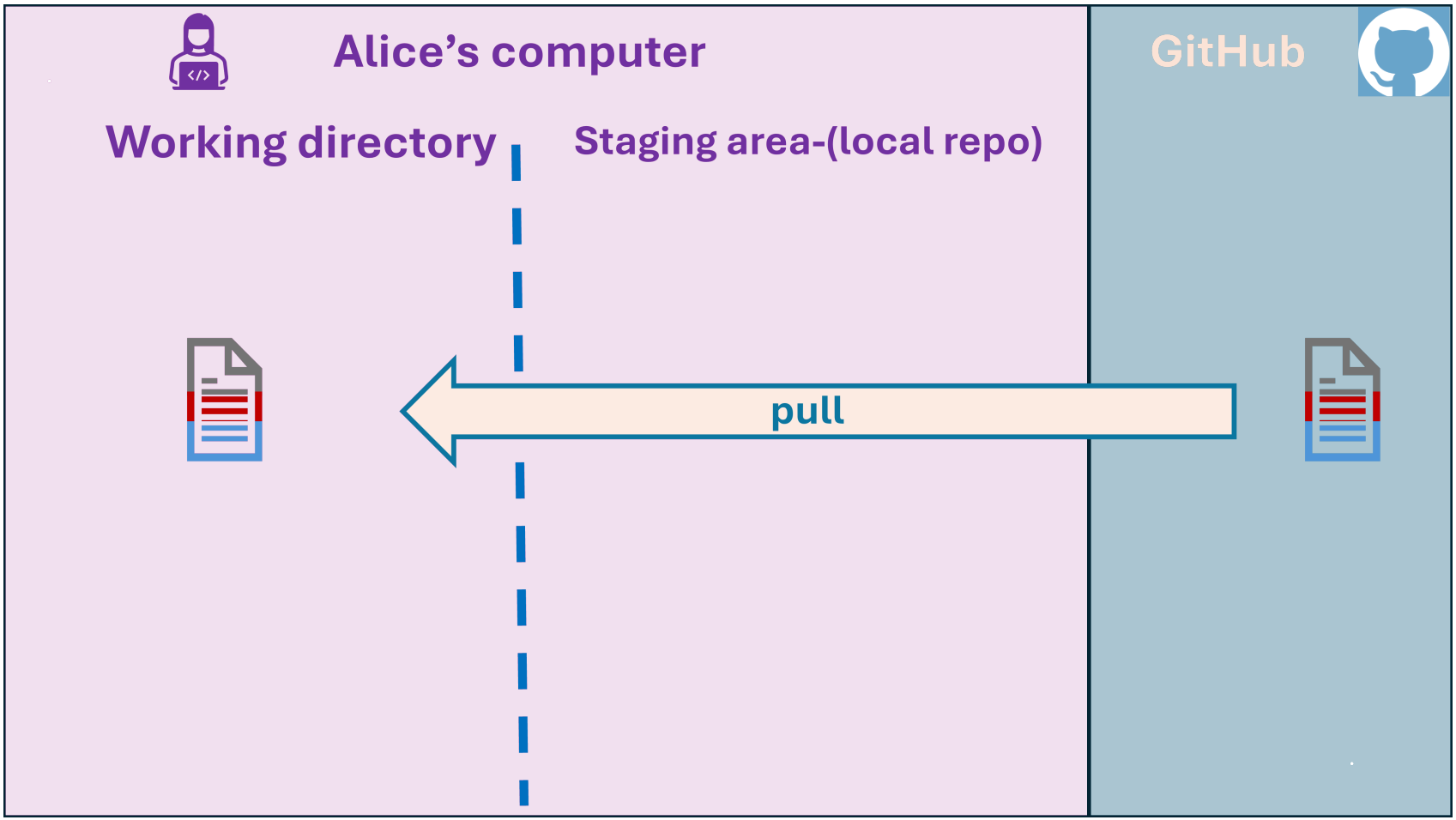
GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



GIT FOR COLLABORATING: DETAILS

Alice and **Bob** work on the same file (Details)



Remarks and issues

- A commit can include changes from multiple files simultaneously

Remarks and issues

- A commit can include changes from multiple files simultaneously
- One can do several commits before pushing (to GitHub)

Remarks and issues

- A commit can include changes from multiple files simultaneously
- One can do several commits before pushing (to GitHub)
- Commit messages can be edited (`amend`)

Remarks and issues

- A commit can include changes from multiple files simultaneously
- One can do several commits before pushing (to GitHub)
- Commit messages can be edited (`amend`)
- Every commit has an identifier (`hash` or `SHA`)

Remarks and issues

- A commit can include changes from multiple files simultaneously
- One can do several commits before pushing (to GitHub)
- Commit messages can be edited (`amend`)
- Every commit has an identifier (`hash` or `SHA`)
- Git manage complex situations

Remarks and issues

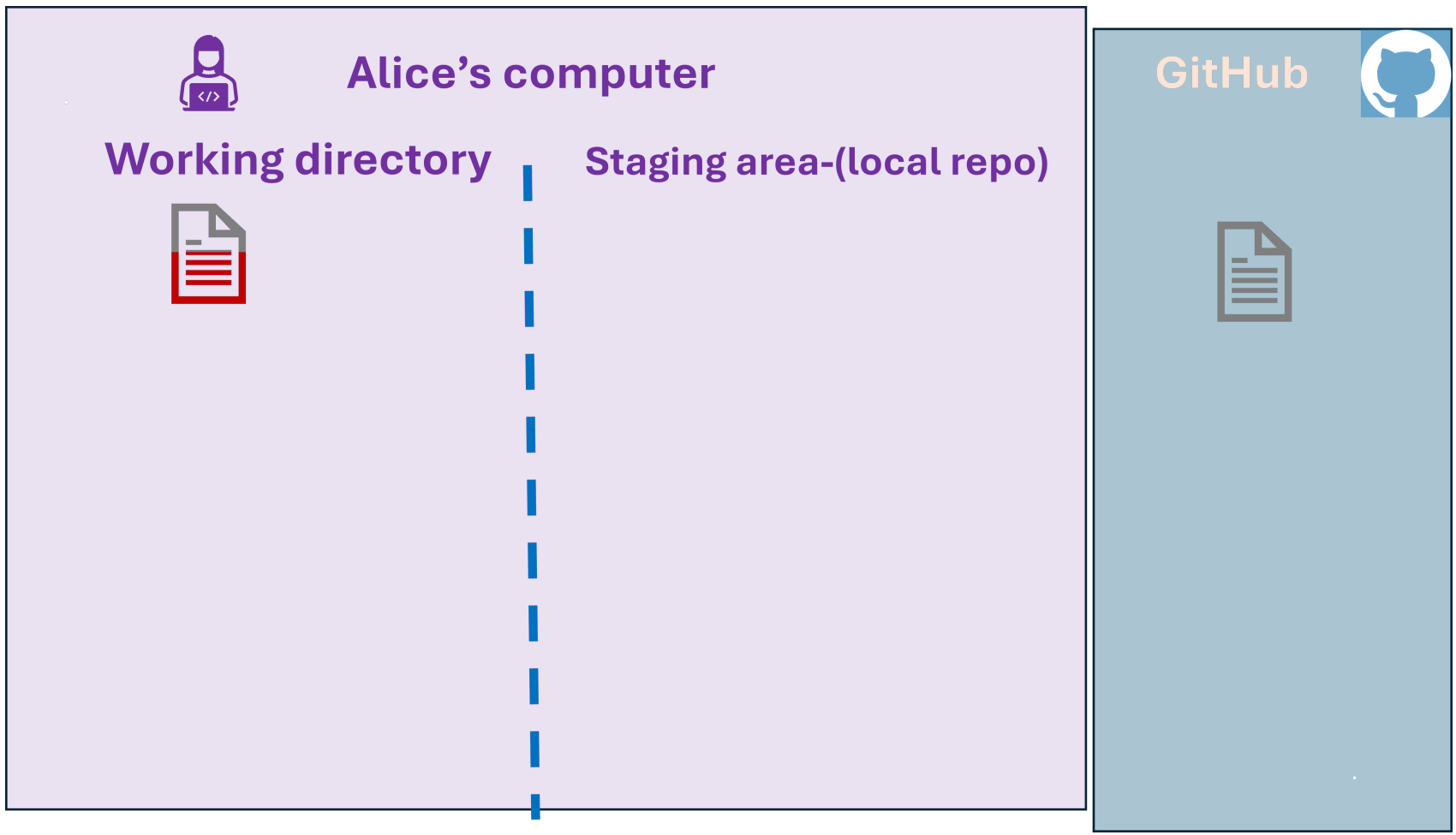
- A commit can include changes from multiple files simultaneously
- One can do several commits before pushing (to GitHub)
- Commit messages can be edited (`amend`)
- Every commit has an identifier (`hash` or `SHA`)
- Git manage complex situations
- Many actions available directly in Visual Studio

GIT FOR COLLABORATING: COMPLEX SITUATIONS

Alice and **Bob** work on the same file **at the same time**

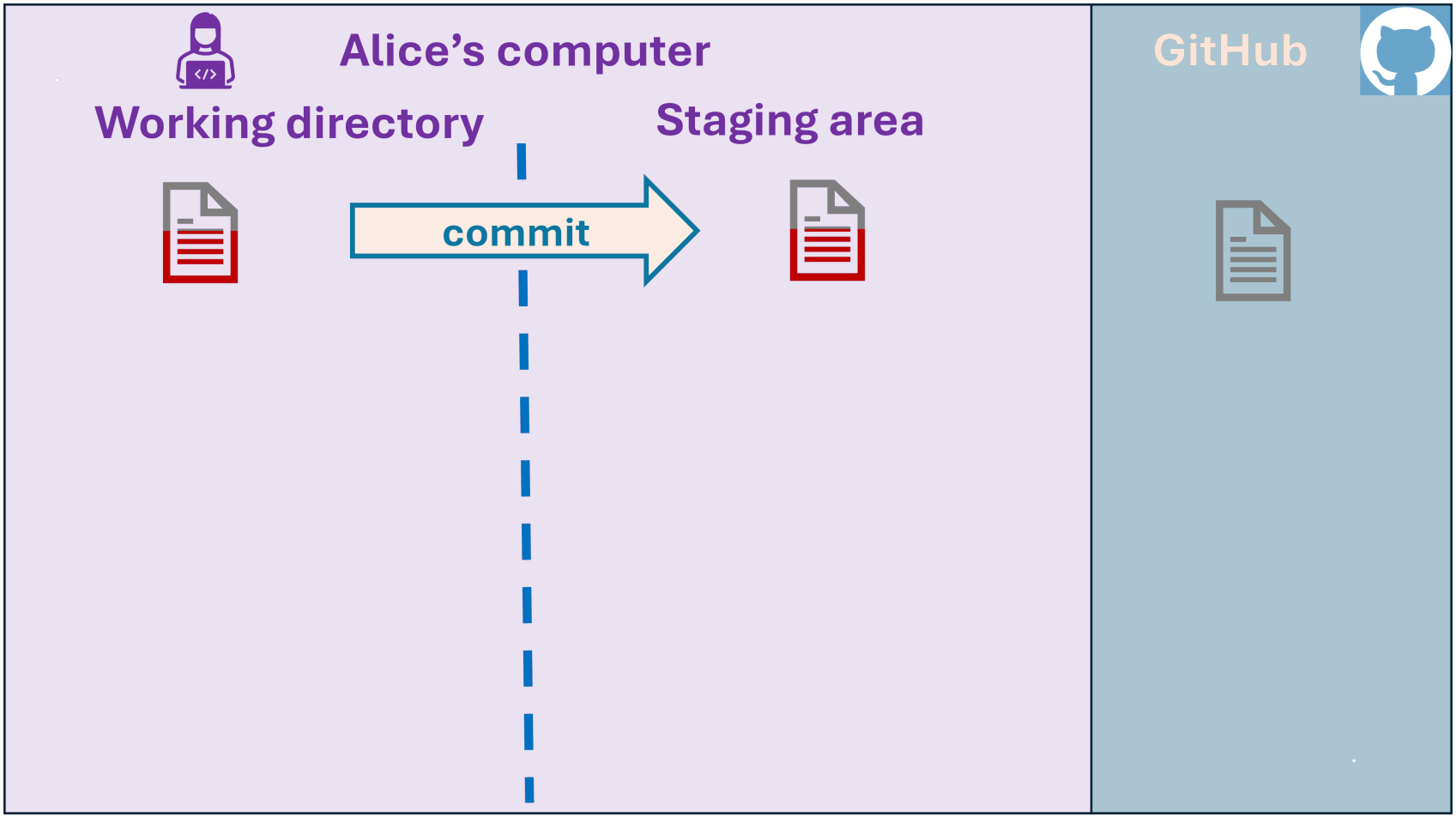
GIT FOR COLLABORATING: COMPLEX SITUATIONS

Alice and **Bob** work on the same file **at the same time**



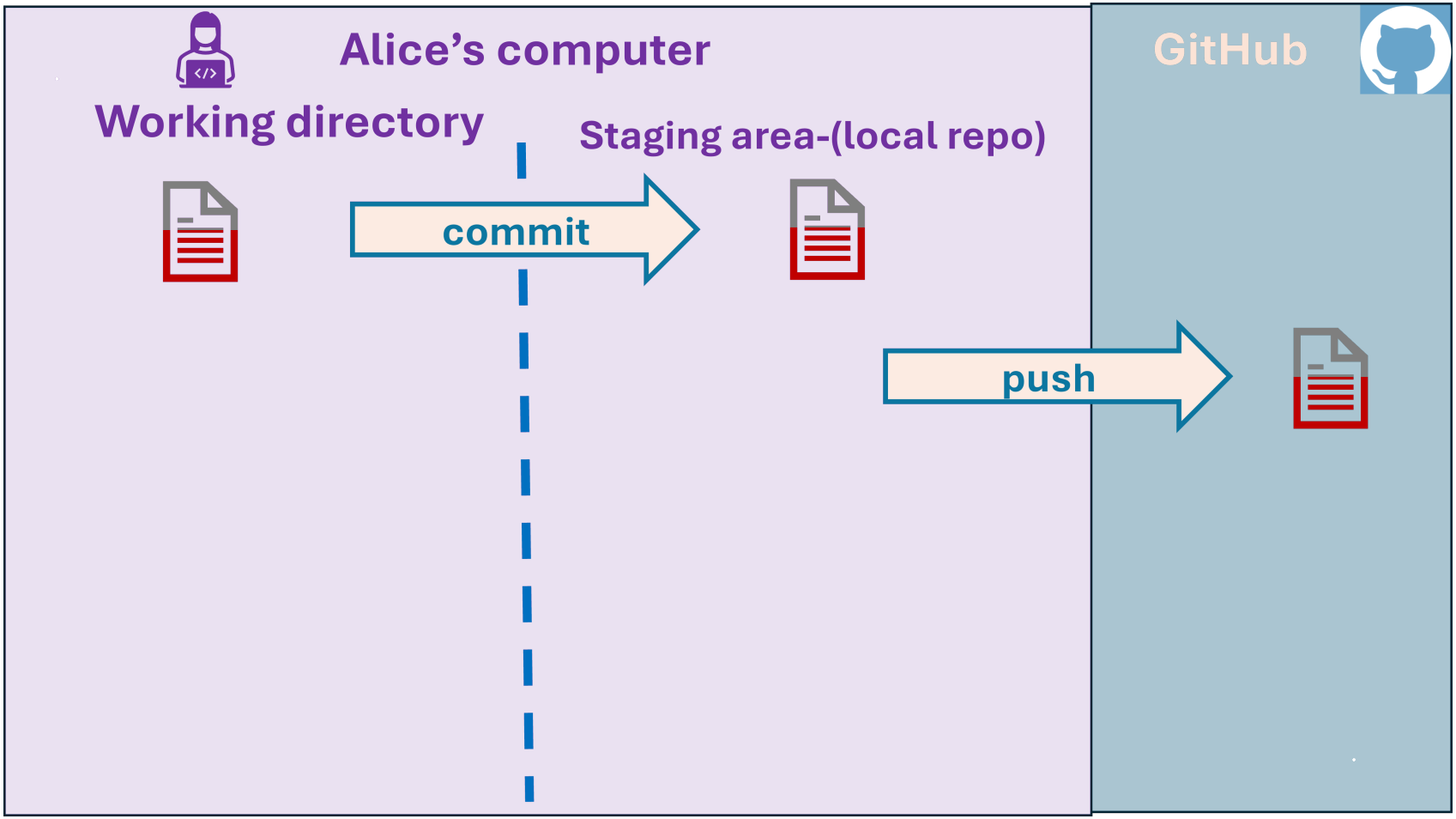
GIT FOR COLLABORATING: COMPLEX SITUATIONS

Alice and **Bob** work on the same file **at the same time**



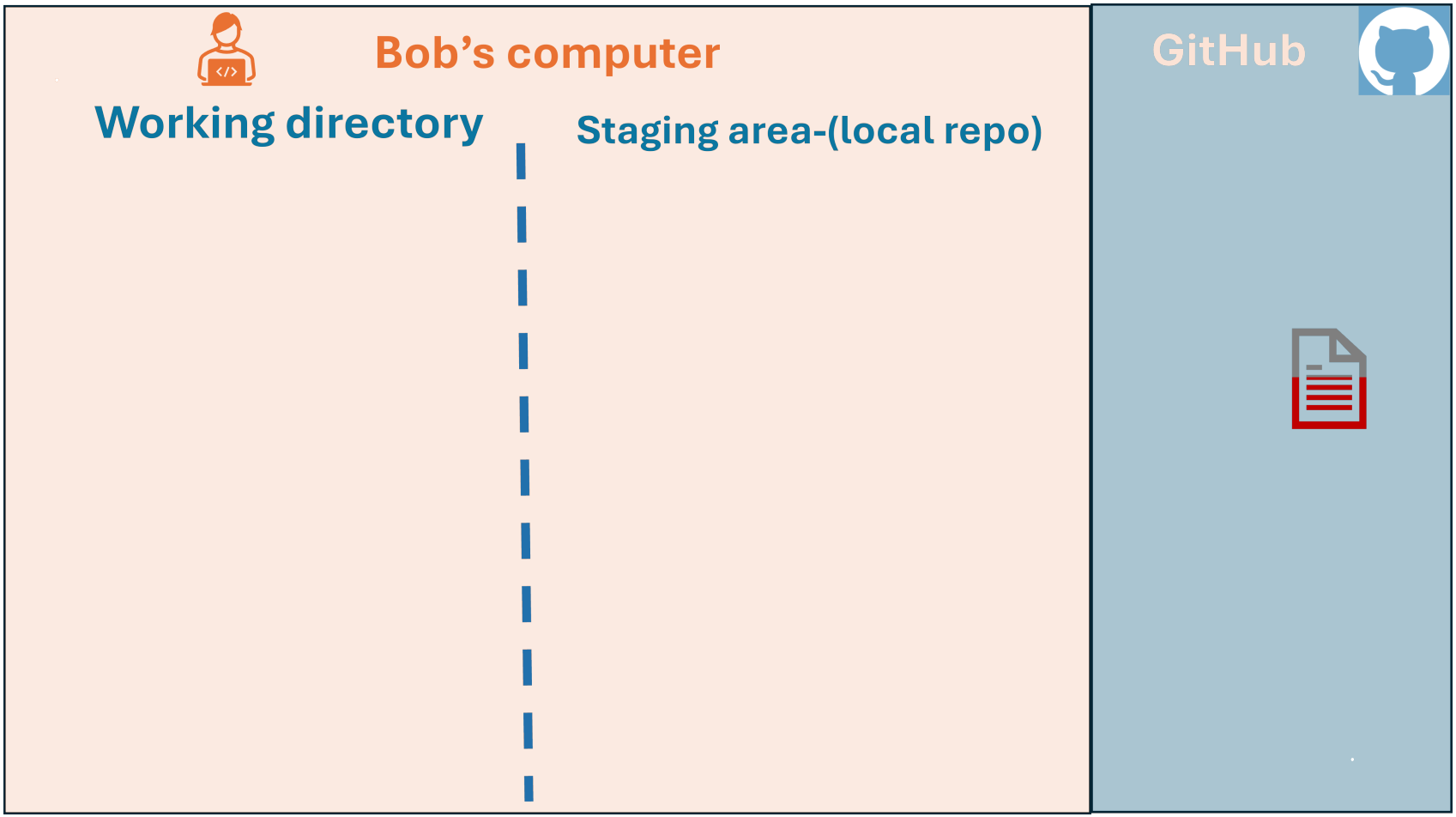
GIT FOR COLLABORATING: COMPLEX SITUATIONS

Alice and **Bob** work on the same file **at the same time**



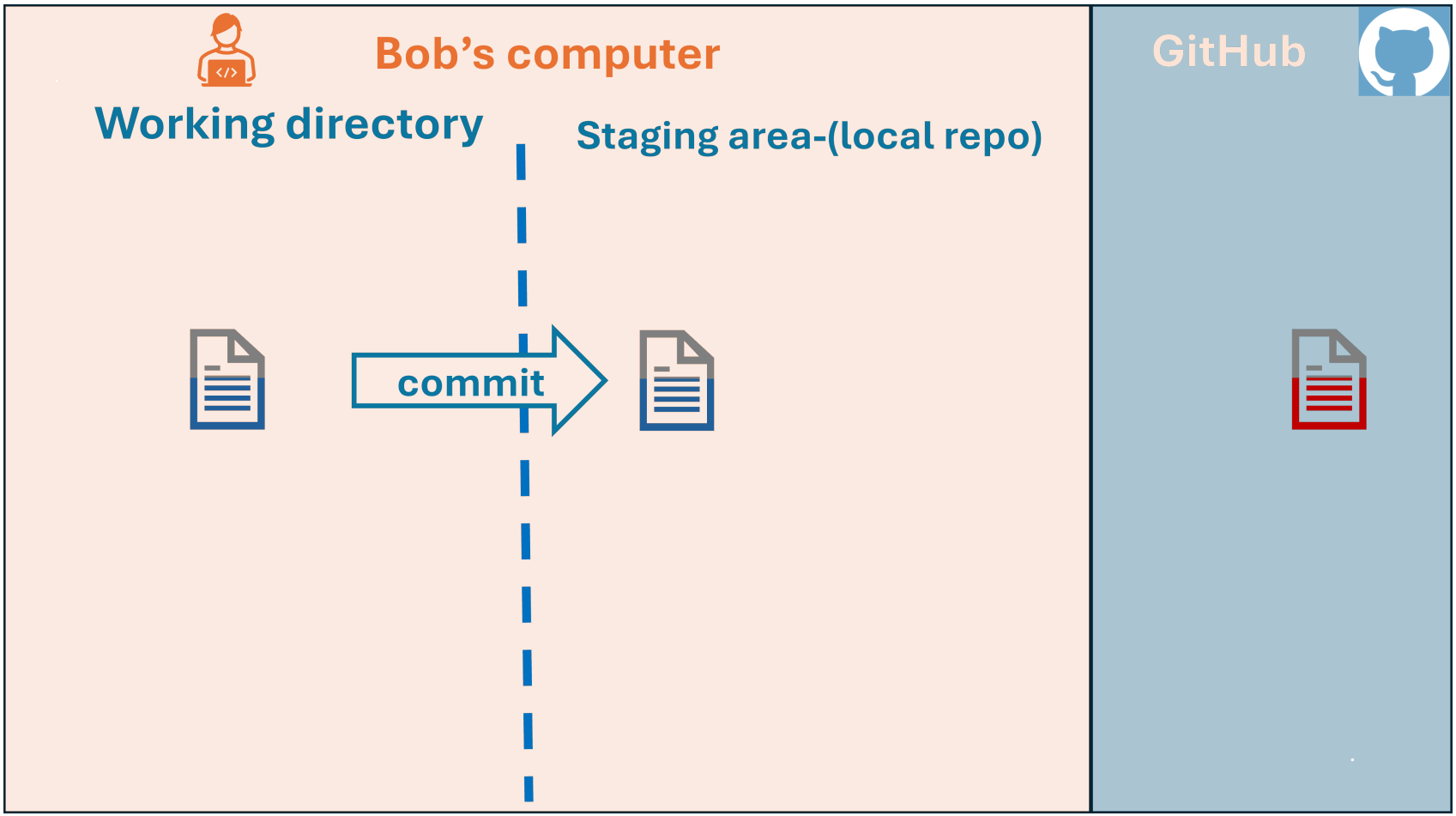
GIT FOR COLLABORATING: COMPLEX SITUATIONS

Alice and **Bob** work on the same file **at the same time**



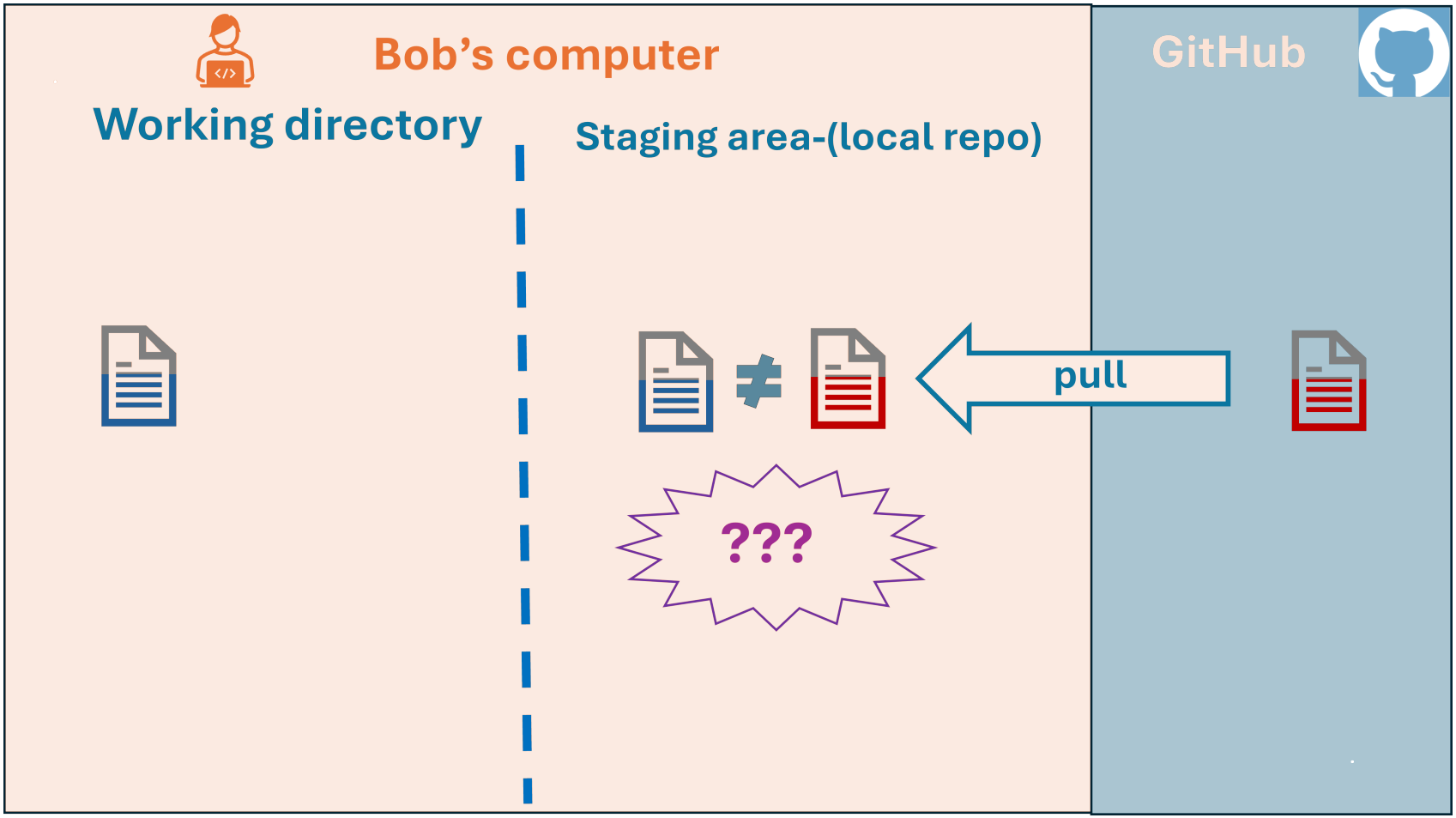
GIT FOR COLLABORATING: COMPLEX SITUATIONS

Alice and **Bob** work on the same file **at the same time**



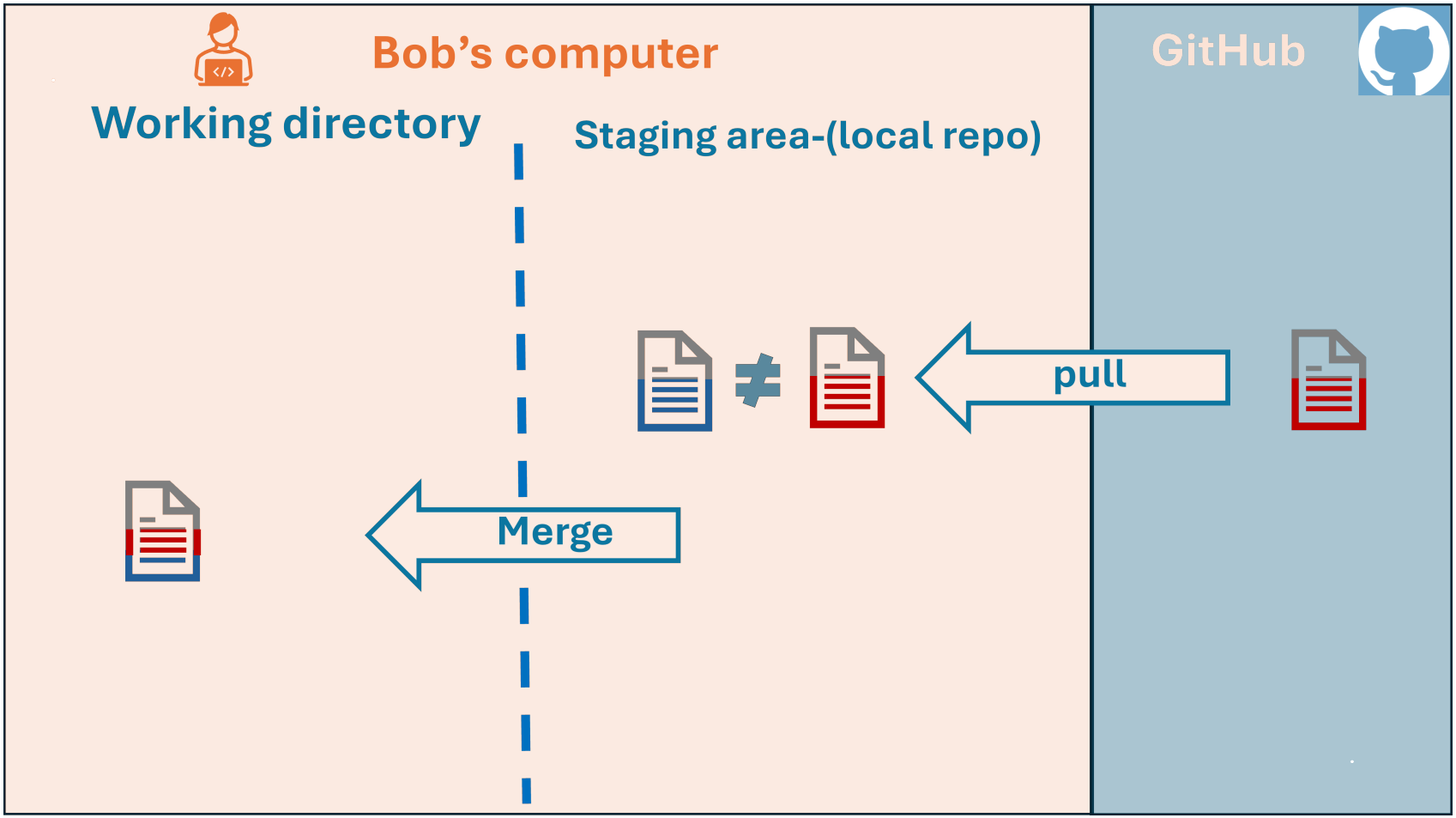
GIT FOR COLLABORATING: COMPLEX SITUATIONS

Alice and **Bob** work on the same file **at the same time**



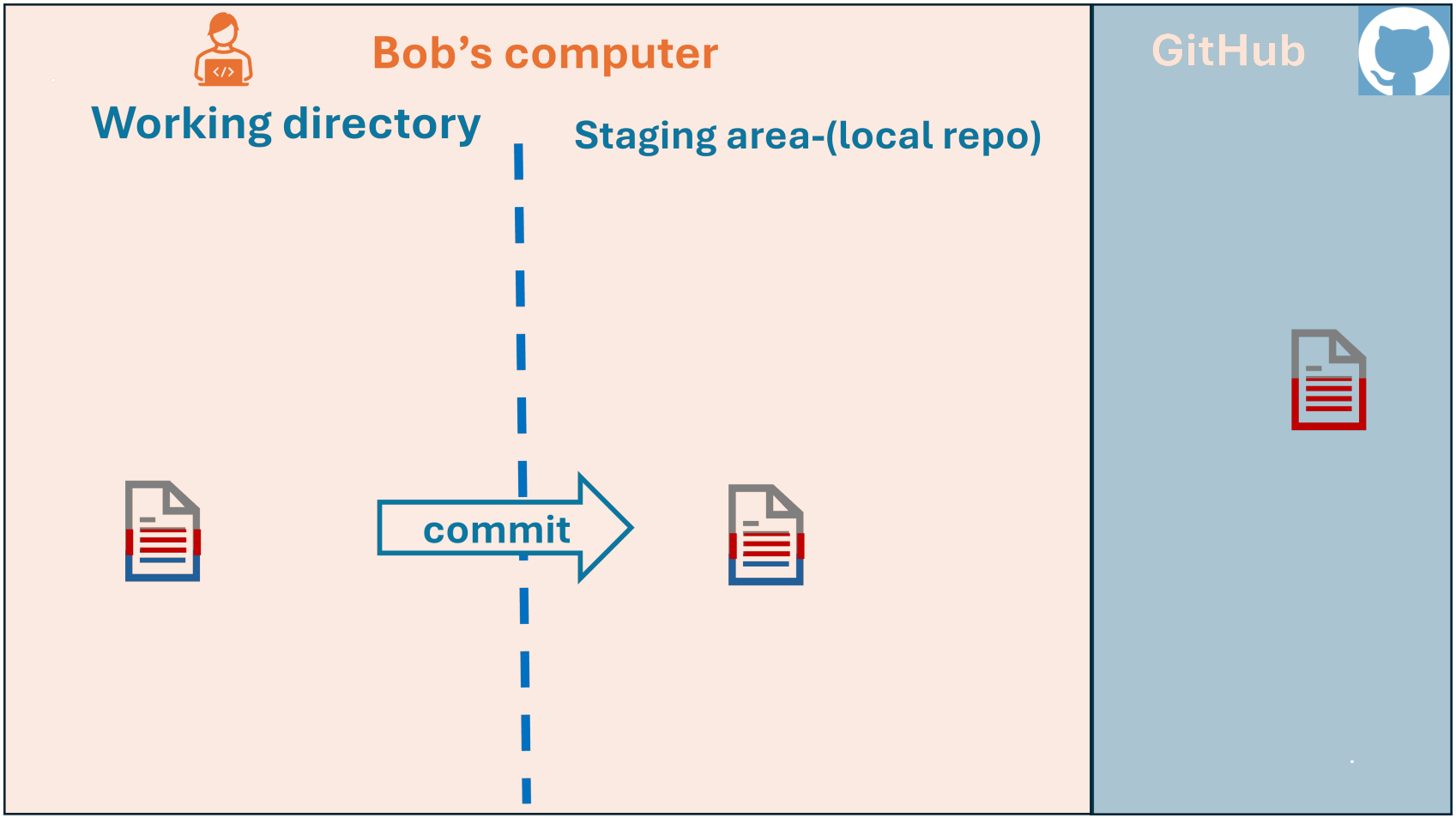
GIT FOR COLLABORATING: COMPLEX SITUATIONS

Alice and **Bob** work on the same file **at the same time**



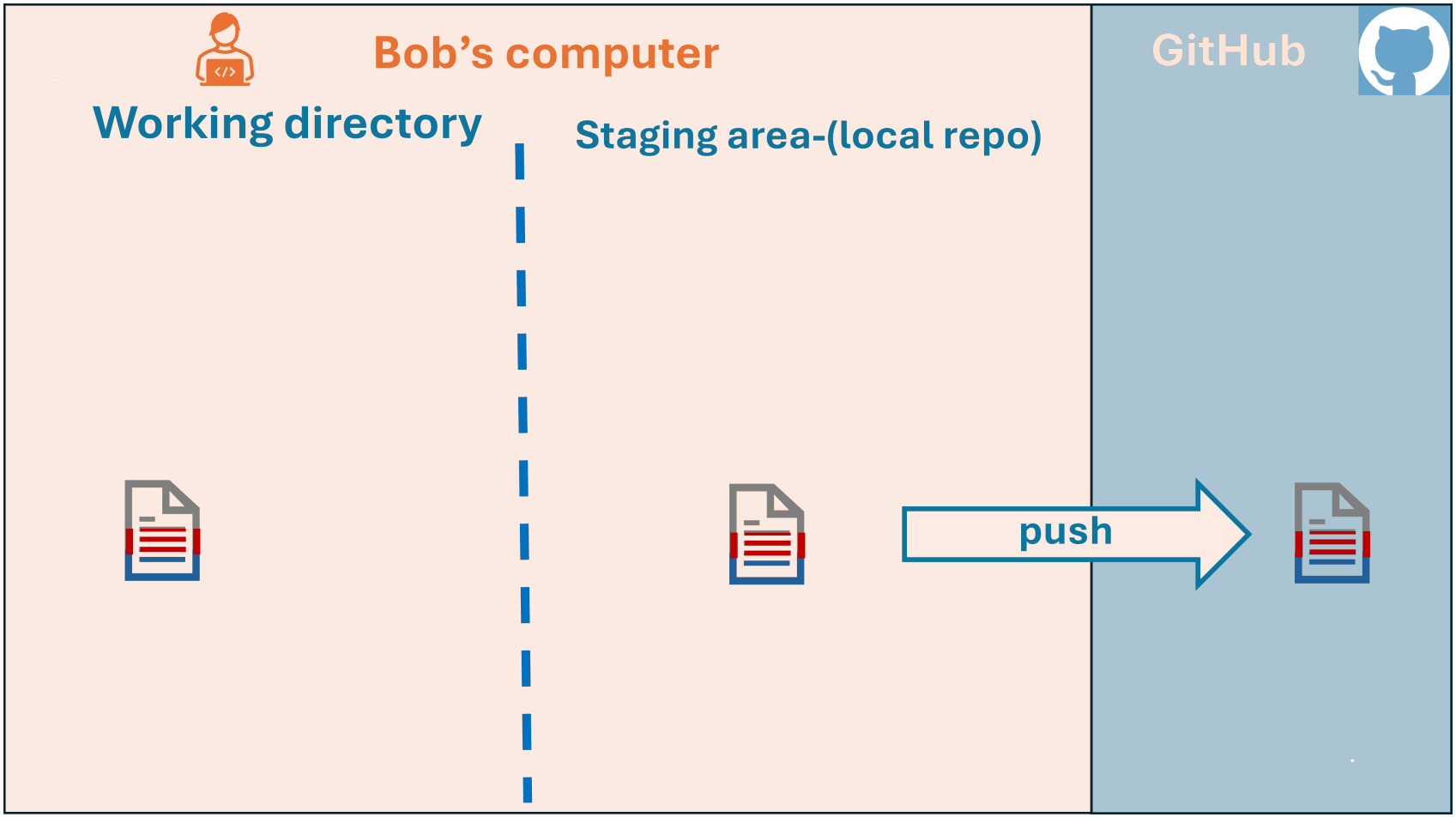
GIT FOR COLLABORATING: COMPLEX SITUATIONS

Alice and **Bob** work on the same file **at the same time**



GIT FOR COLLABORATING: COMPLEX SITUATIONS

Alice and **Bob** work on the same file **at the same time**



Tea Break


UNITED NATIONS
Statistical Institute for
Asia and the Pacific

RAP principles 3 : Good (coding) Practices

RAP principles 3 : Good (coding) Practices

- Good coding practices

RAP principles 3 : Good (coding) Practices

- Good coding practices

↪ Write for humans, not for machines

RAP principles 3 : Good (coding) Practices

- Good coding practices
- ↪ Write for humans, not for machines
- Testing

RAP principles 3 : Good (coding) Practices

- Good coding practices
  - ↪ Write for humans, not for machines
- Testing
- Peer-review

RAP principles 3 : Good (coding) Practices

- Good coding practices
- ↪ Write for humans, not for machines
- Testing
- Peer-review
- Exercise part 3 (15 minutes) Coding, documenting, testing.