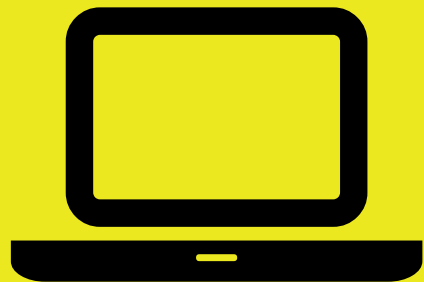


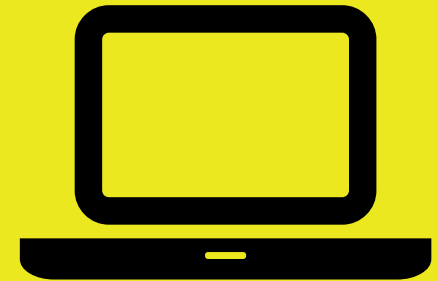
VERTIEFENDE PROGRAMMLEHRE II

CODING IM TEAM

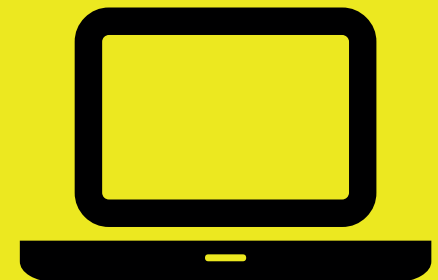
Fritz



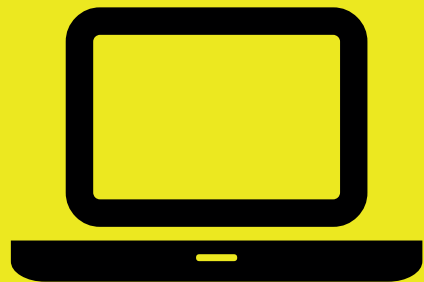
Peter



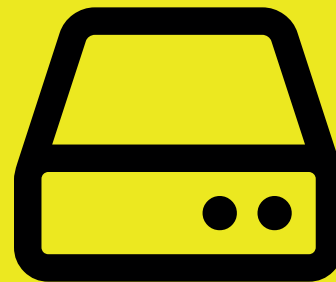
Steffi



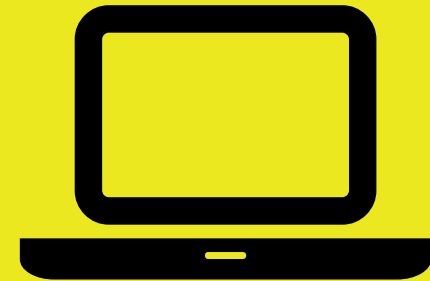
Fritz



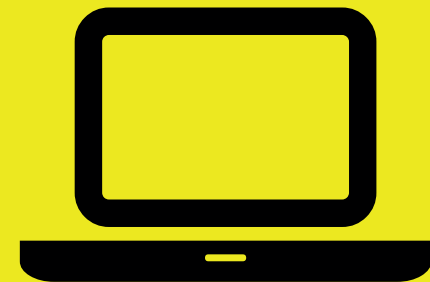
SFTP

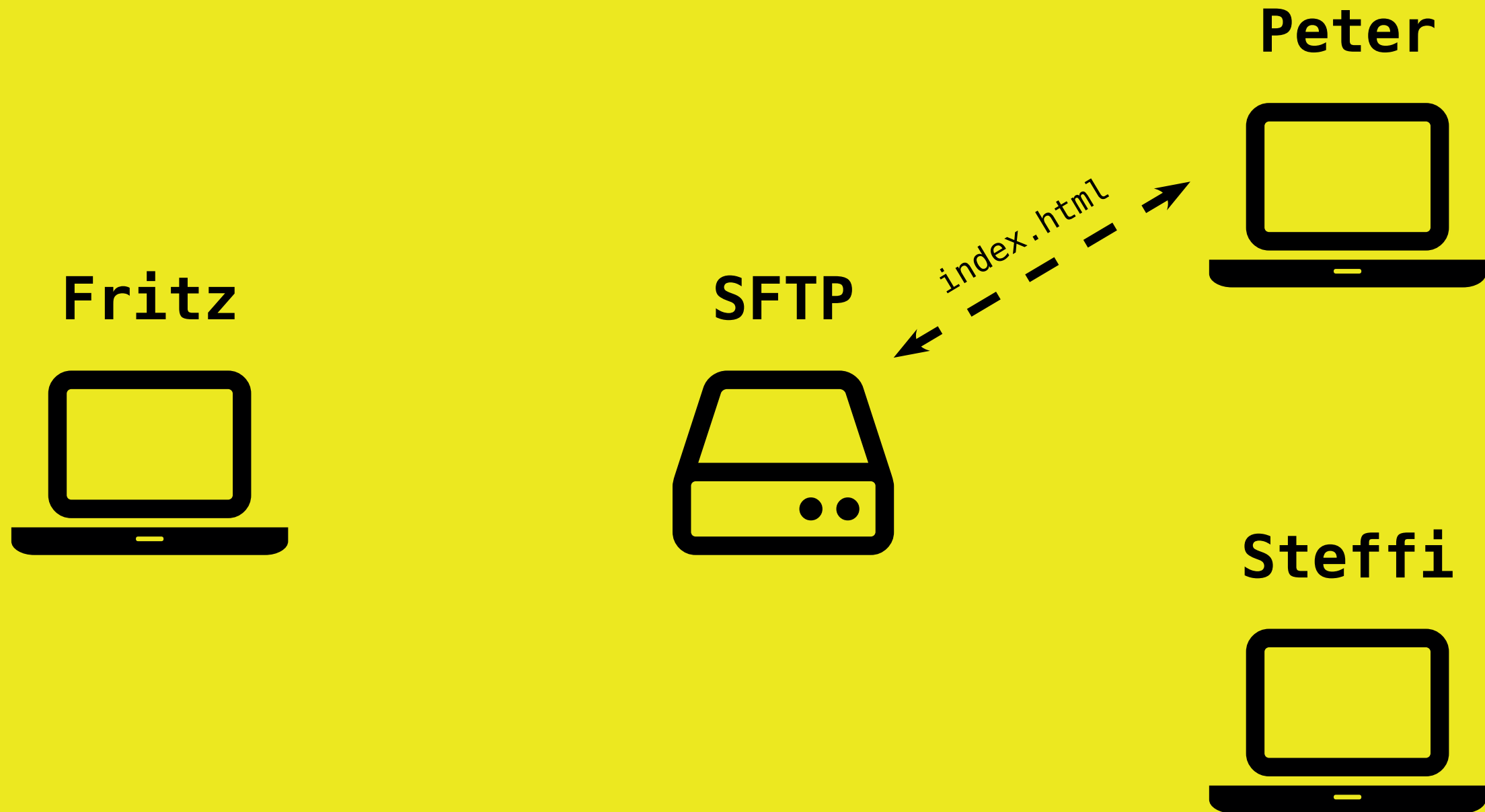


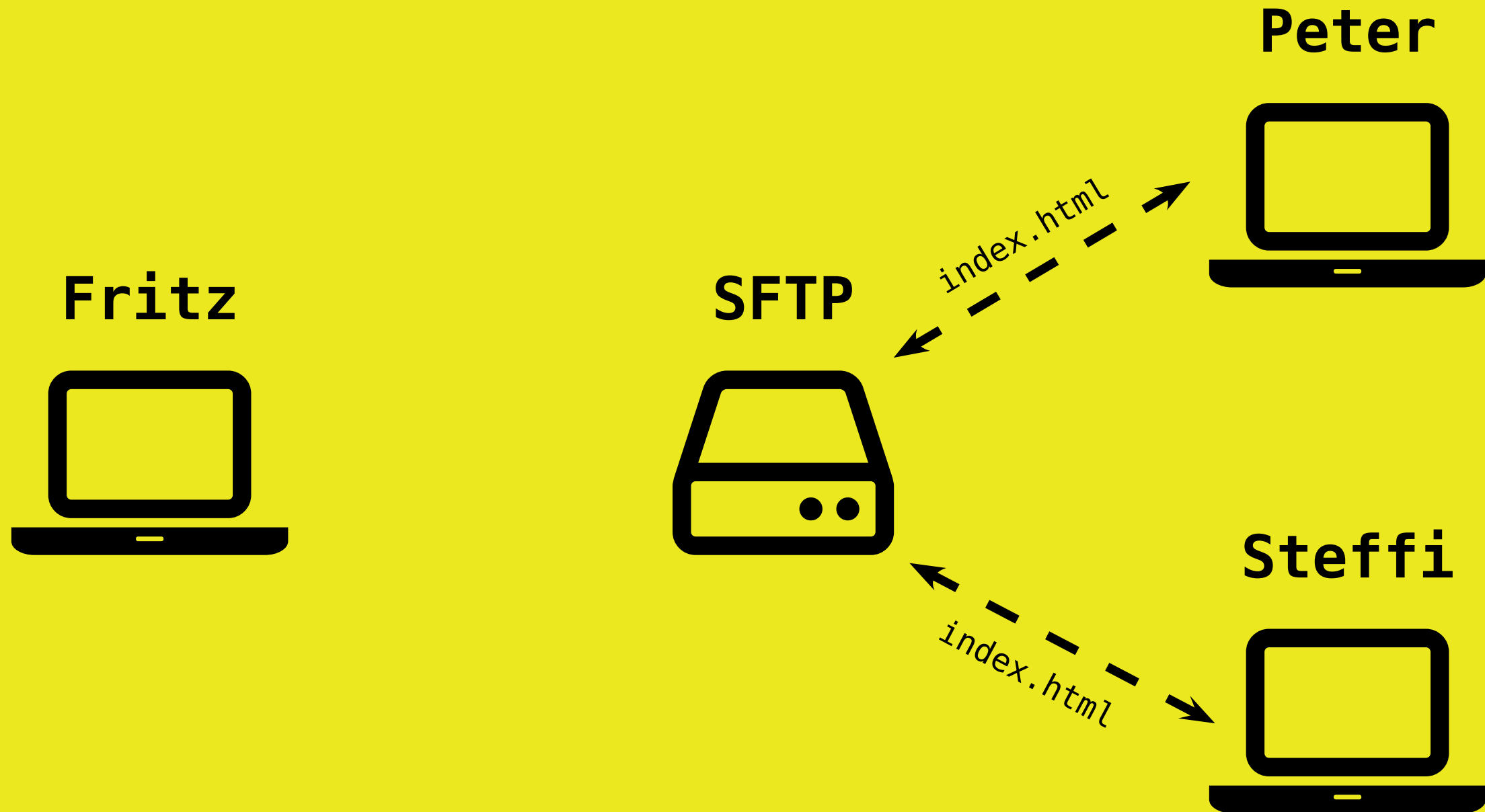
Peter

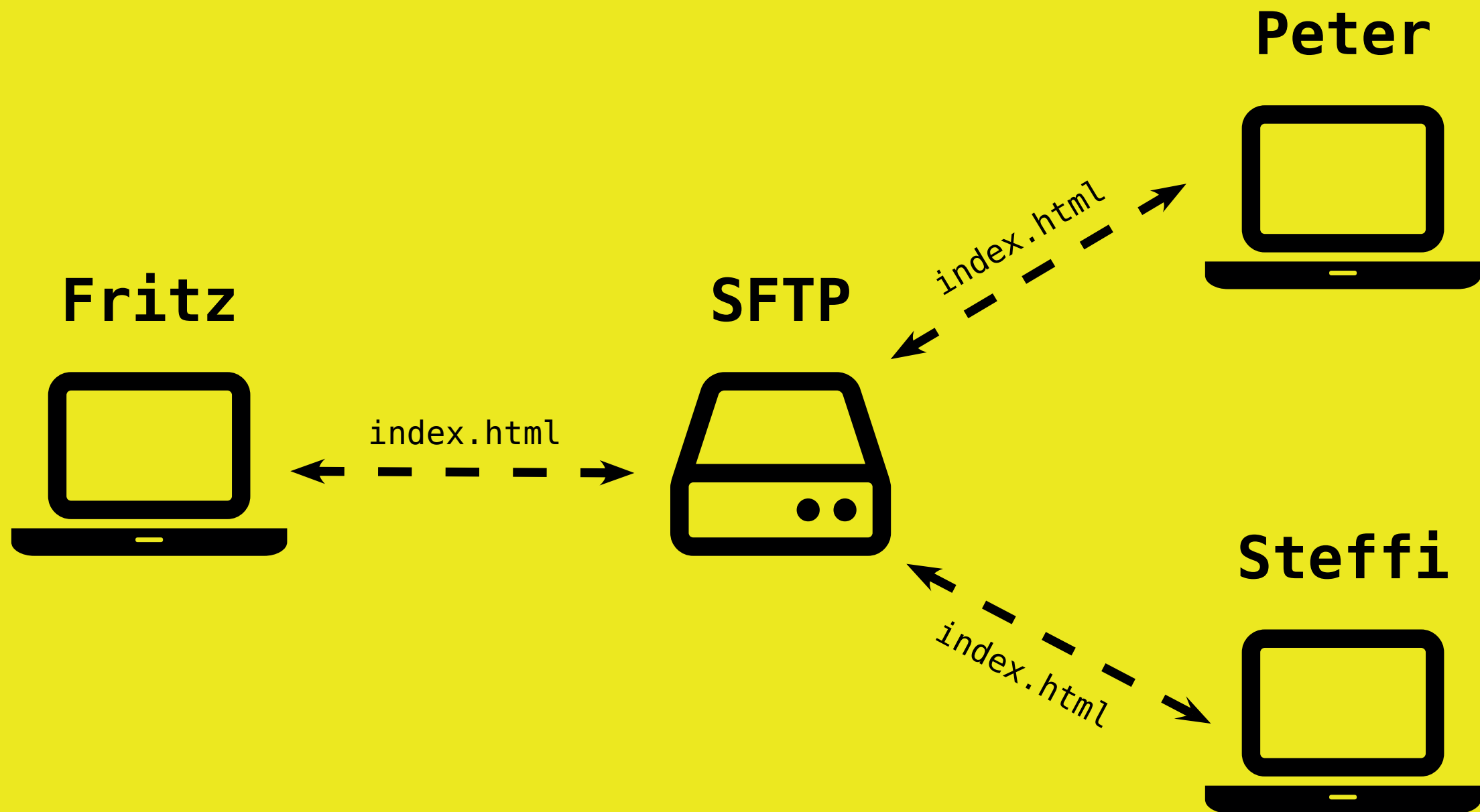


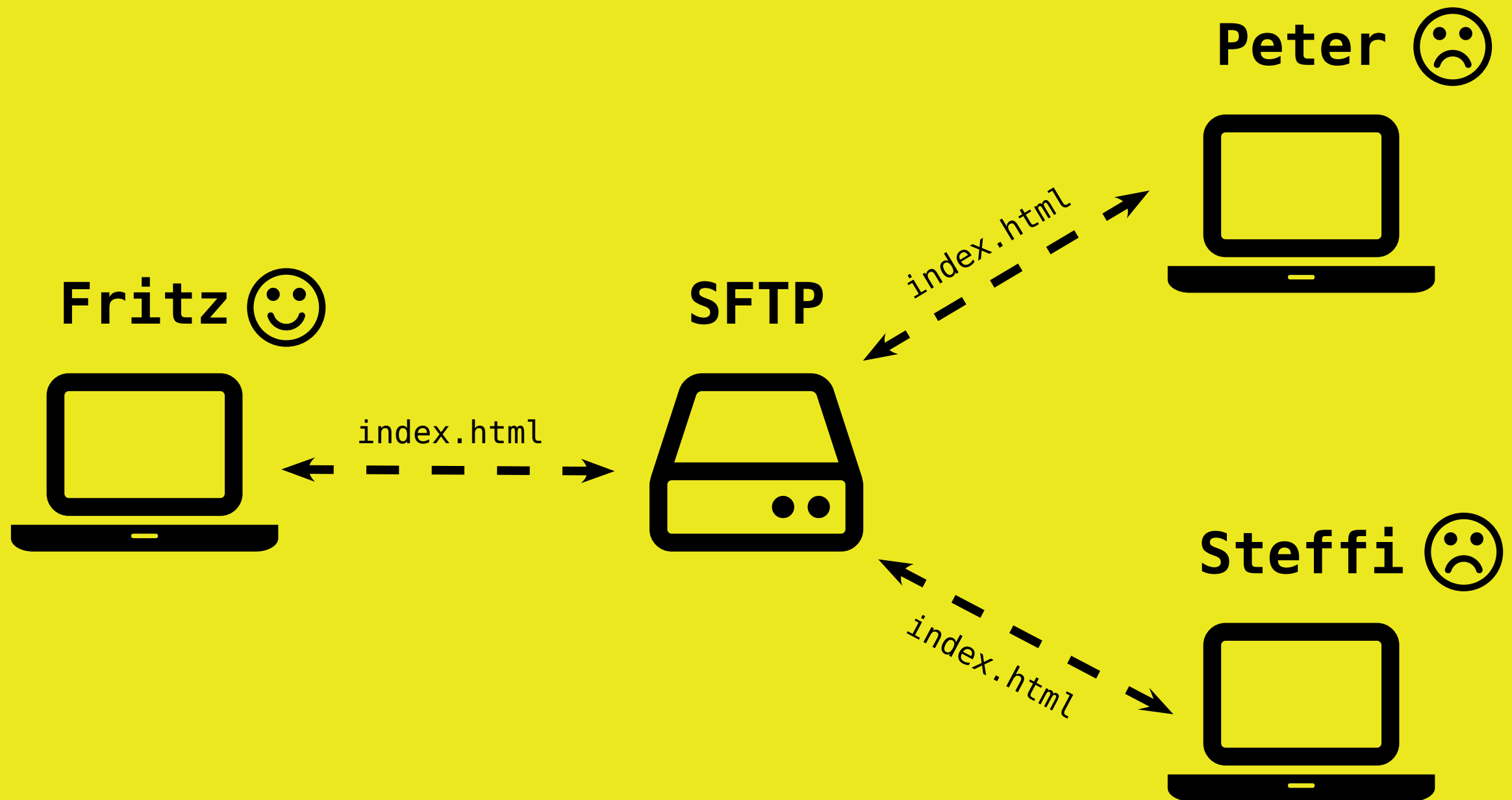
Steffi



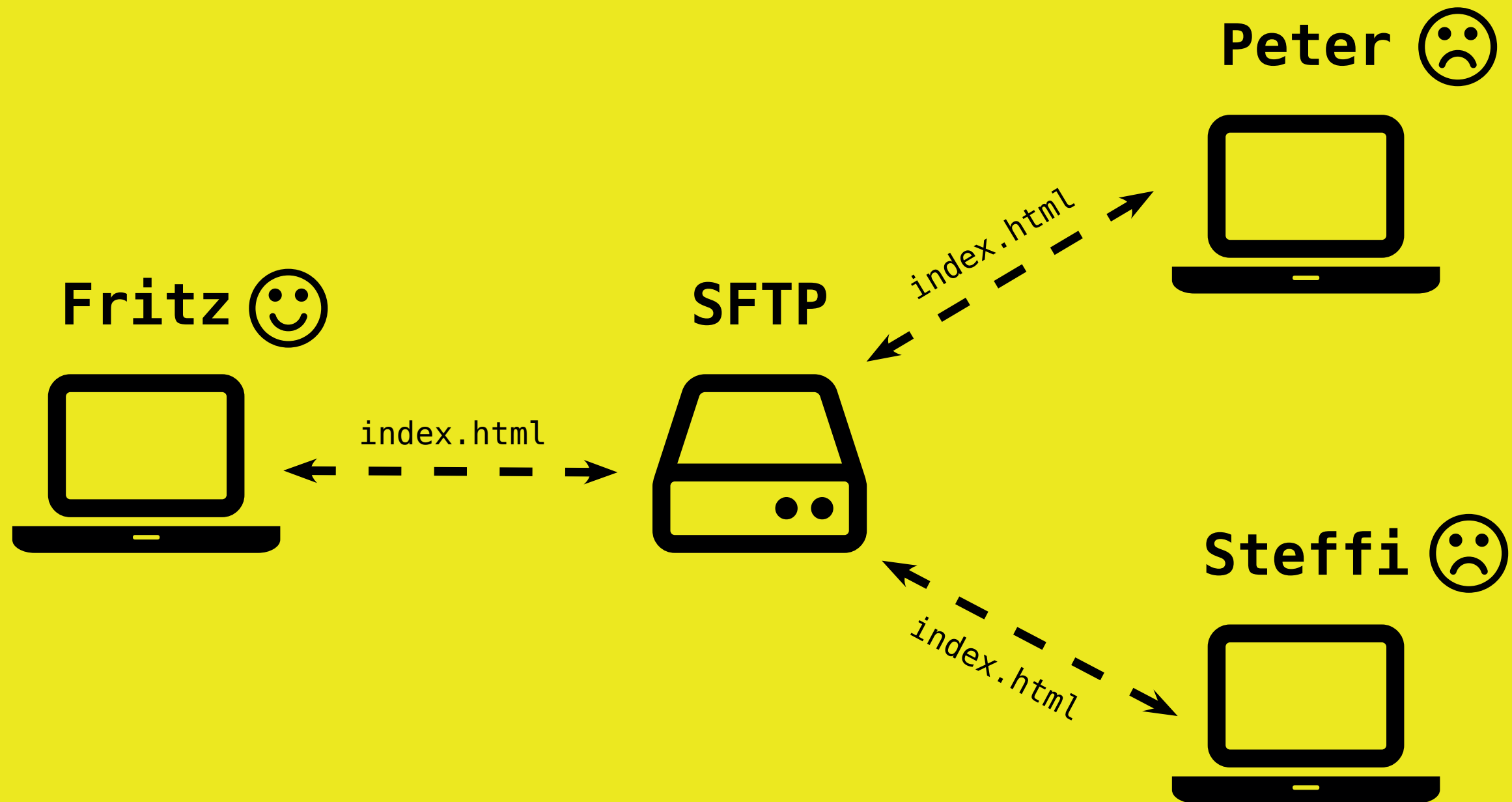








DER LETZTE GEWINNT



VERSIONSVERWALTUNG

VERSIONSVERWALTUNG

- > VCS (engl. Version Control System)
- > Protokollierung, Wiederherstellung, Archivierung
- > Änderungen mit Verfasser und Zeitstempel (Datum, Uhrzeit) im Archiv (**Repository**) gespeichert
- > Master Entwicklungszweig (**Branch**)
- > Branches: Abzweigungen vom Master (parallele Entwicklung von Features)
- > Populäres Beispiel WIKIPEDIA
- > Verschiedene Arten: Lokal, Zentral, Dezentral

LOKAL

- > älteste Art der Versionierung
- > passiert nur auf eigenem PC
- > meist nur eine Datei (bspw. technische Zeichnungen)

LOKAL

- > älteste Art der Versionierung
- > passiert nur auf eigenem PC
- > meist nur eine Datei (bspw. technische Zeichnungen)

ZENTRAL

- > Client-Server-Architektur
- > Repository (kurz: Repo) auf Server
- > Benutzerverwaltung mit Rechteverteilung
- > gesamte **History** nur in Repo auf Server

DEZENTRALE VERSIONSVERWALTUNG MIT GIT

DEZENTRAL

- > zentrales Repo ist kein muss
- > jeder Benutzer hat eigenes Repo
- > jedes Repo kann mit jedem abgeglichen werden
- > jedes Repo hat eigene History
- > Änderungen können lokal verfolgt werden

DEZENTRAL

- > zentrales Repo ist kein muss
- > jeder Benutzer hat eigenes Repo
- > jedes Repo kann mit jedem abgeglichen werden
- > jedes Repo hat eigene History
- > Änderungen können lokal verfolgt werden

GIT

- > engl. Blödmann
- > freie Software zur Versionsverwaltung
- > entwickelt von Linus Torvalds

DEZENTRAL

- > zentrales Repo ist kein muss
- > jeder Benutzer hat eigenes Repo
- > jedes Repo kann mit jedem abgeglichen werden
- > jedes Repo hat eigene History
- > Änderungen können lokal verfolgt werden

GIT

- > engl. Blödmann
- > freie Software zur Versionsverwaltung
- > entwickelt von Linus Torvalds

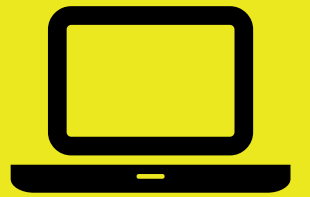
“I’m an egotistical bastard, and I name all my projects after myself. First ‘Linux’, now ‘Git’.”

– Linus Torvalds

Fritz



Steffi



Fritz



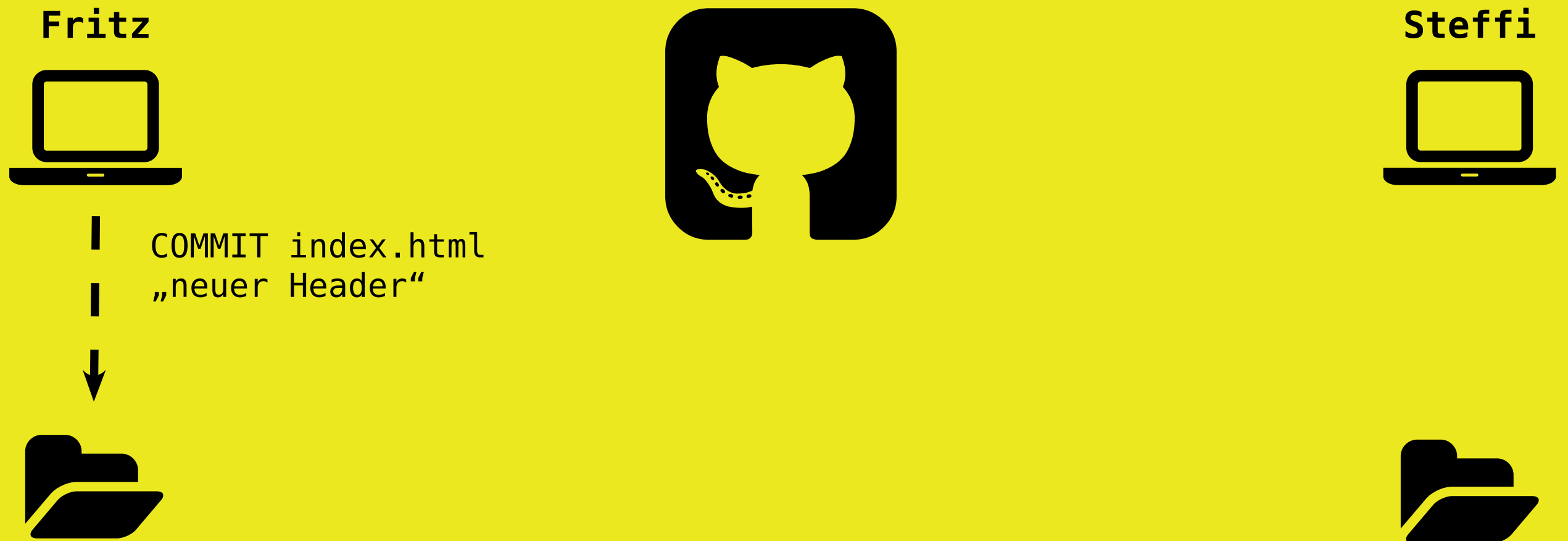
| COMMIT index.html
| „neuer Header“



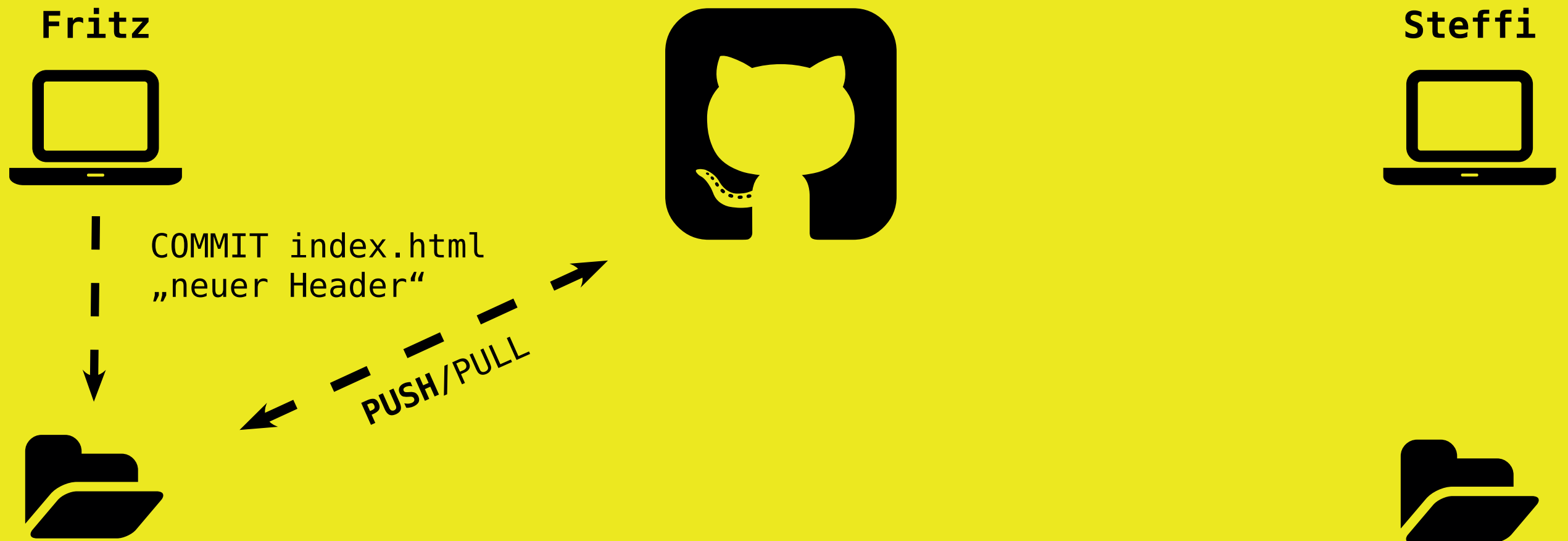
Steffi



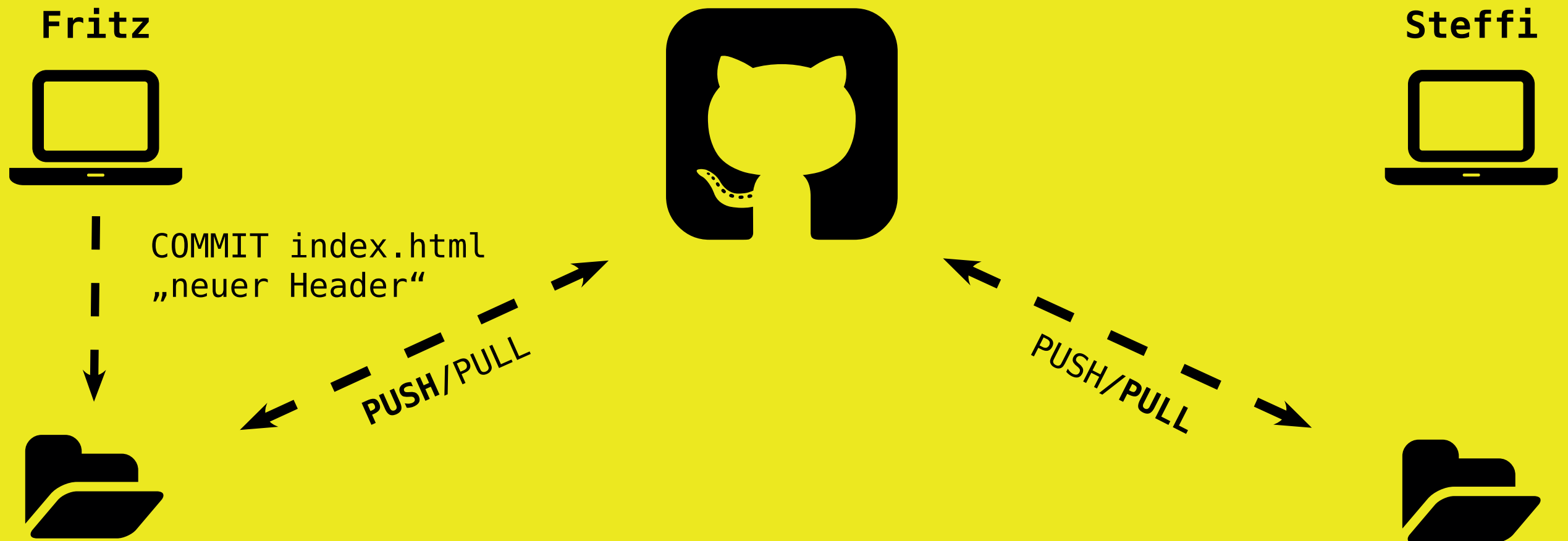
GitHub Repository



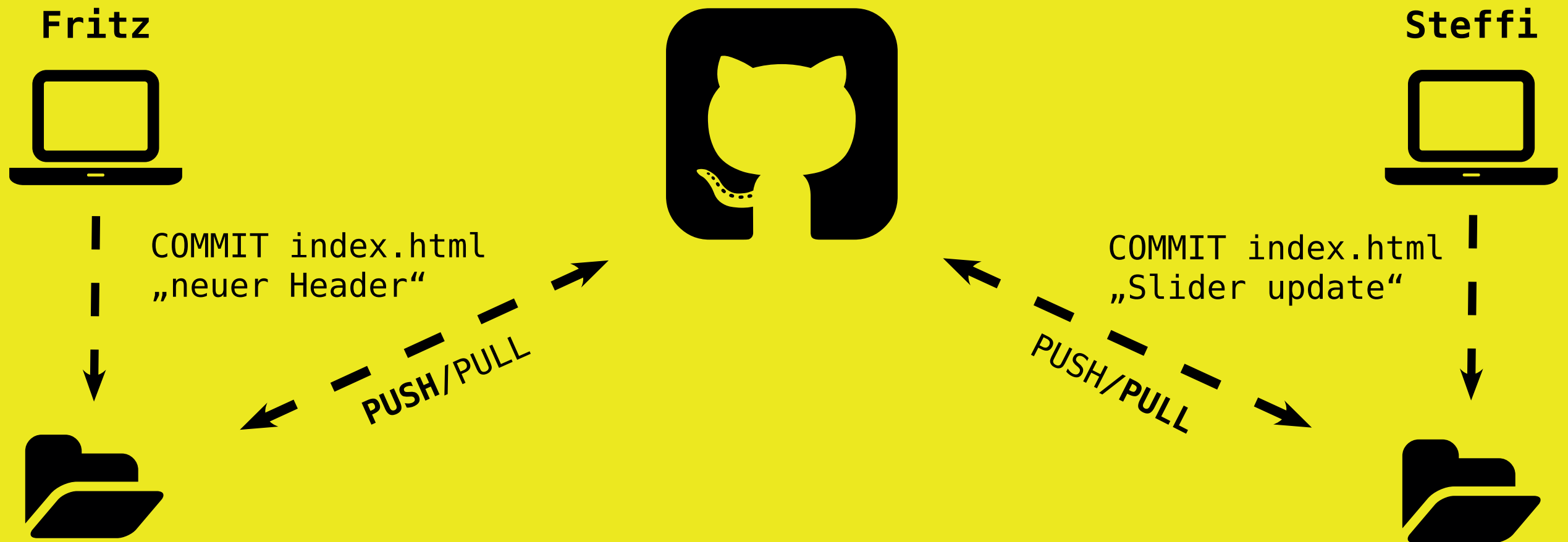
GitHub Repository



GitHub Repository

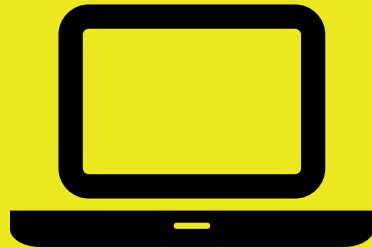


GitHub Repository



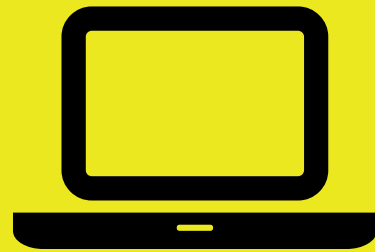
- 1. GitHub Account anlegen - <https://github.com/>**
- 2. SourceTree installieren - <https://www.sourcetreeapp.com/>**
- 3. Git Repository anlegen**
- 4. Clonen in SourceTree**
- 5. README.md bearbeiten**
- 6. Stagen, Comitten, Pushen**

GIT WORKFLOW



**Dateien
bearbeiten**

GIT WORKFLOW

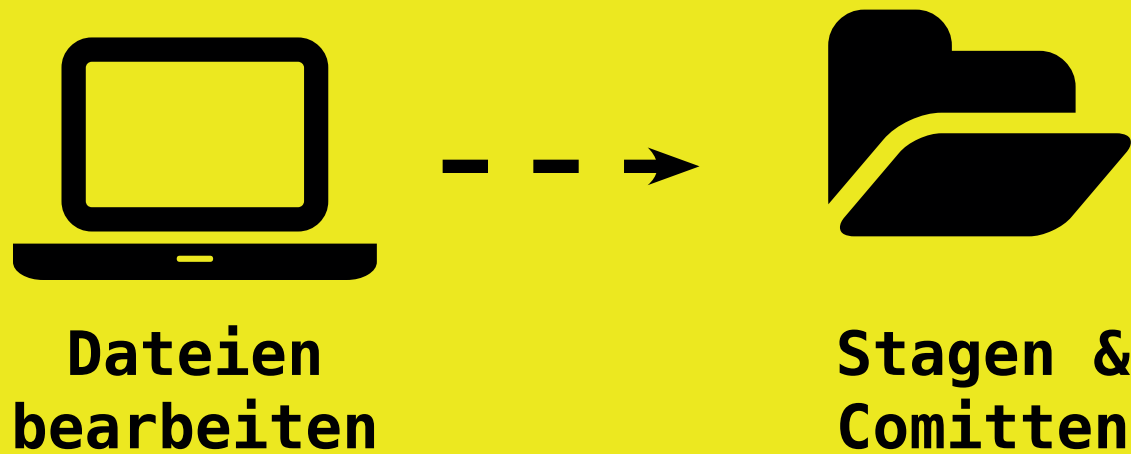


Dateien
bearbeiten



Stagen &
Comitten

GIT WORKFLOW



> Commit Message:
Was wurde verändert?

GIT WORKFLOW



> Commit Message:
Was wurde verändert?

PROJEKT: GIT DOKUMENTATION/ABLAUF

> 3 Gruppe mit je einem Repository

> ohne CMS

> Erkläre folgende Begriffe:

VCS, Repository, Git, GitHub, SourceTree, Branch,
Commit, Staging, Push, Pull, Merge

> Erkläre GitHub Repository Anbindung an SourceTree

> Stelle einen einfachen Git Workflow dar
(Grafik, Diagramm, etc.)

AUFGABE BIS NÄCHSTE WOCH

- > App Landing Page erstellen (ohne CMS)
- > Thema: freie Wahl
- > Erstellung & Synchronisierung eines Git Repos bei GitHub