



# Programmierungsmethodik 1

## Programmiertechnik

Versionsverwaltung, Live Demo  
Raspberry Pi

- Collections-Framework
- Verkettete Liste und Array-Liste
- Iteratoren
- Vergleichen
- Menge
- Map
- Collections-Operationen

Ausblick für heute

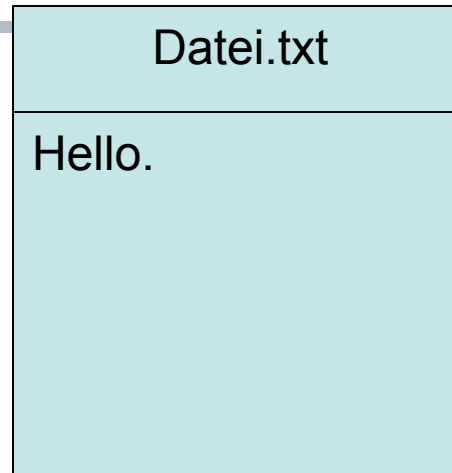
- Mehrere Personen arbeiten am gleichen Quellcode. Es wird gewünscht, dass
  - Änderungen nachverfolgt werden können
  - Änderungen zurückgenommen werden können
  - Konflikte (parallele Änderung des gleichen Quellcode-Fragments) aufgelöst werden können
  - Dritte einfachen Zugang zum Quellcode haben

# Agenda



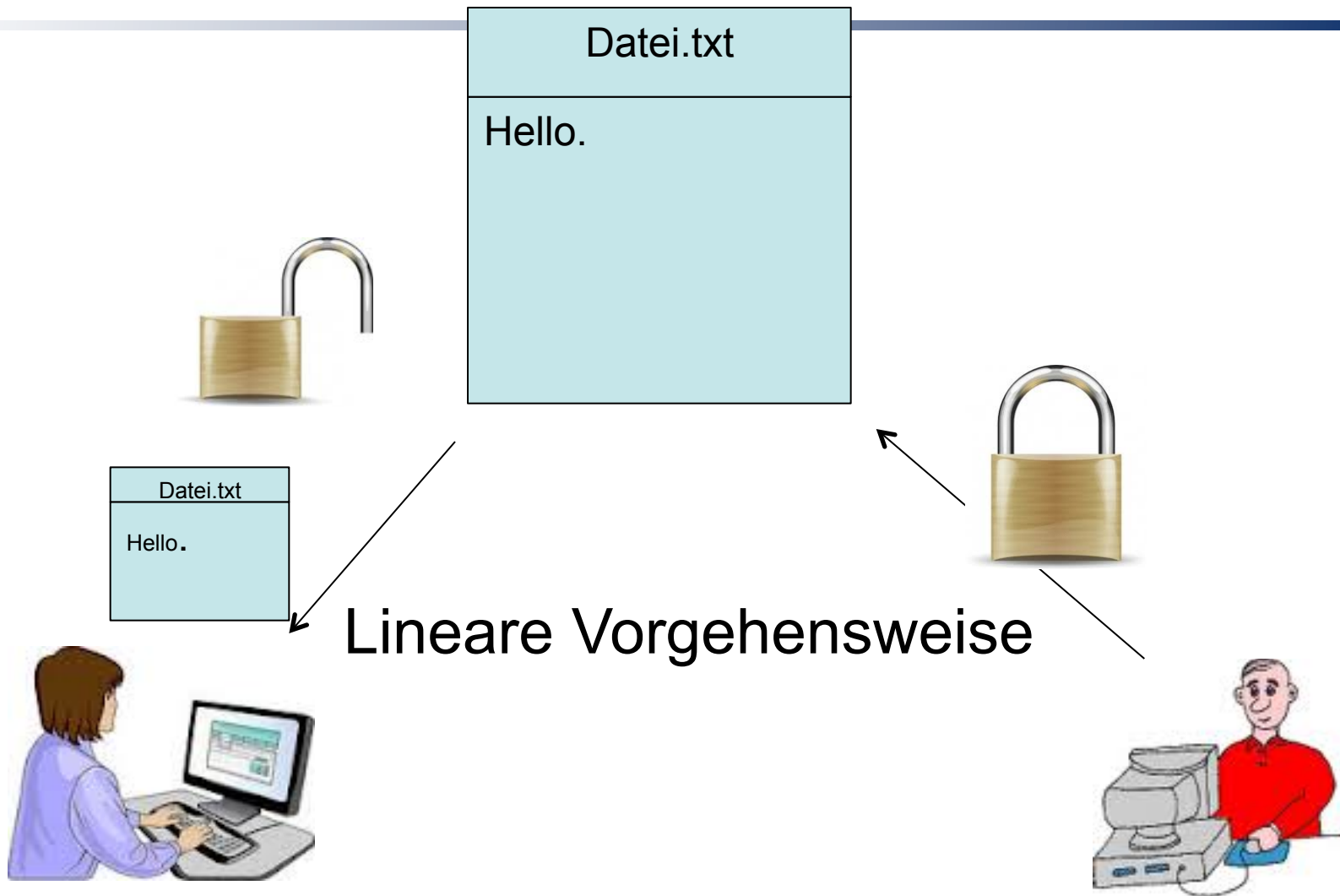
- Versionsverwaltung
- Git
- Live-Demo: Raspberry Pi

Versionsverwaltung

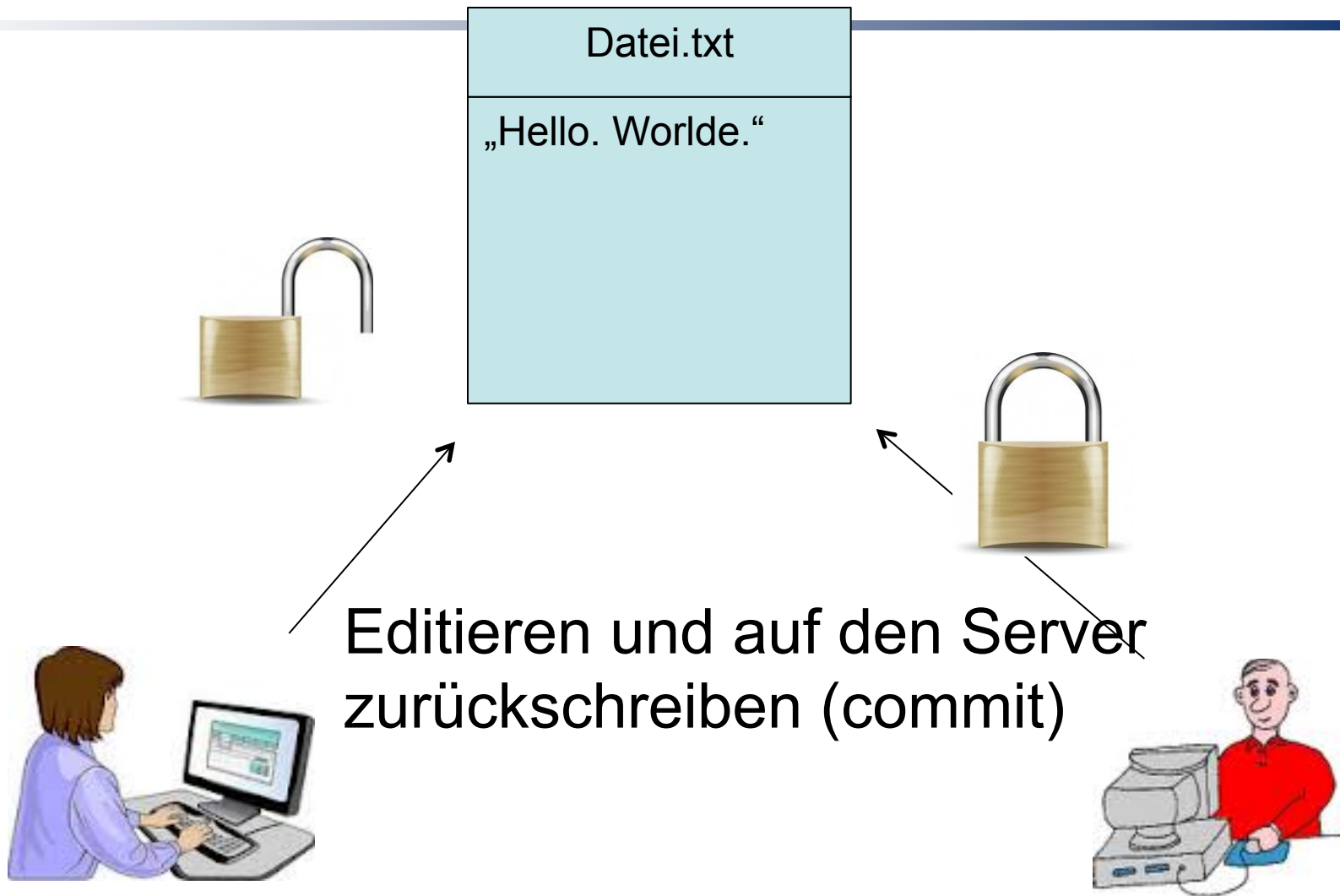


Beide wollen eine  
Datei bearbeiten



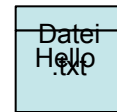


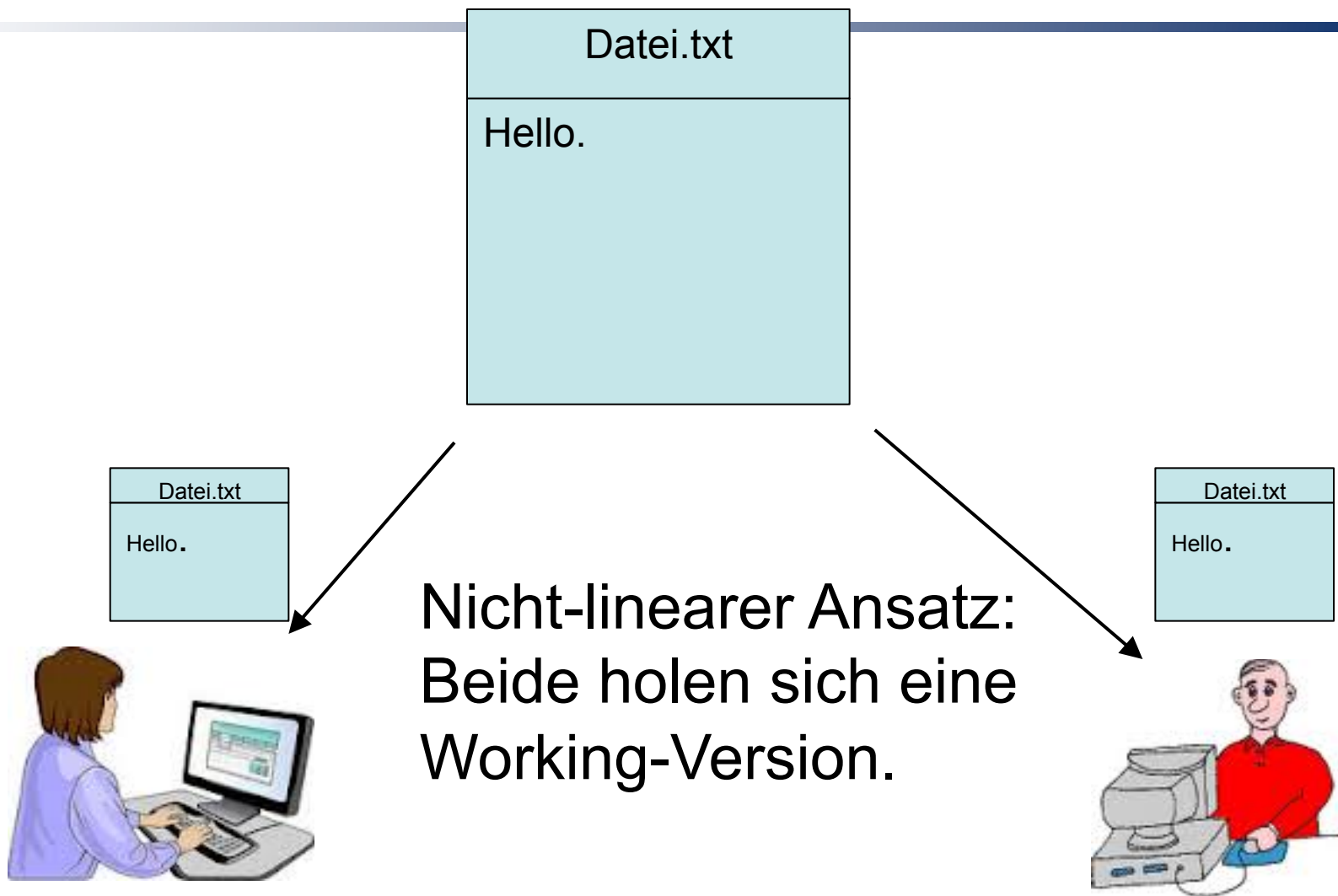






Lokale Kopie holen,  
editieren, committen.





Datei.txt
Hello.

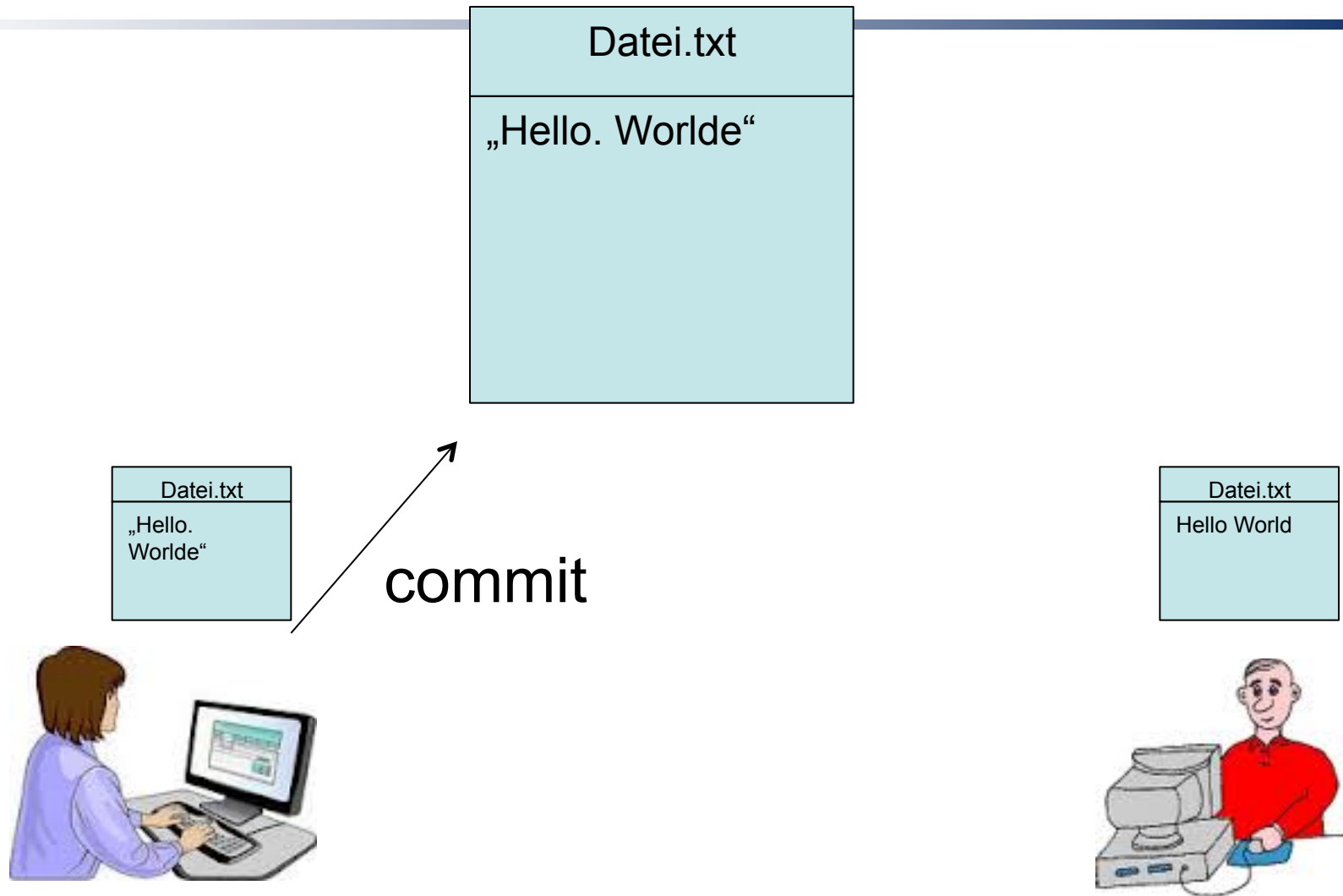
Datei.txt
„Hello. Worlde“

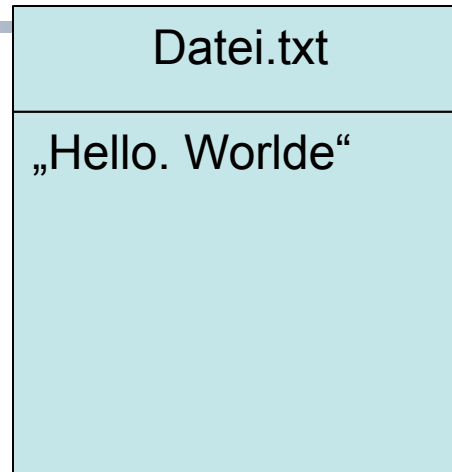


Beide editieren  
gleichzeitig.

Datei.txt
Hello World

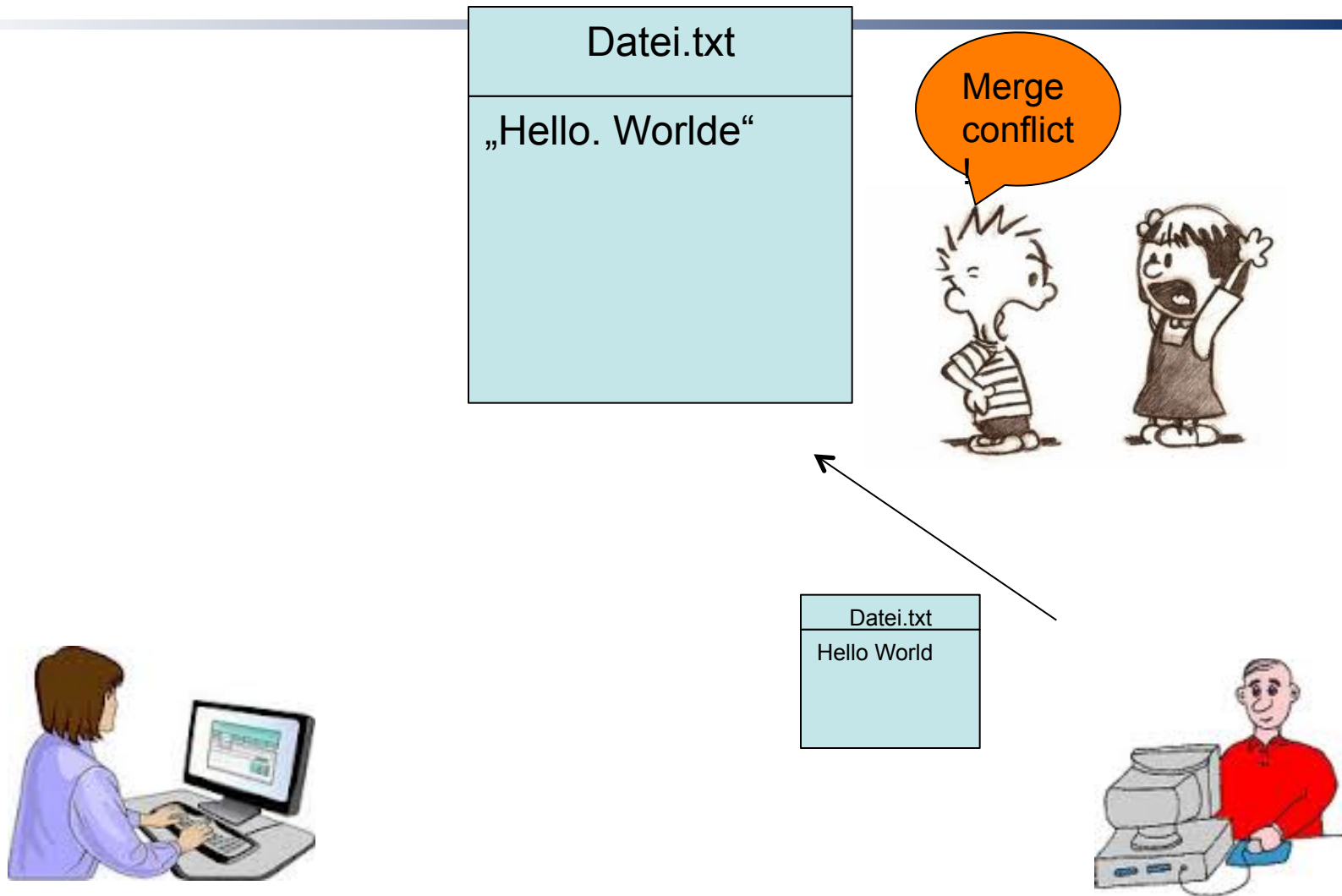




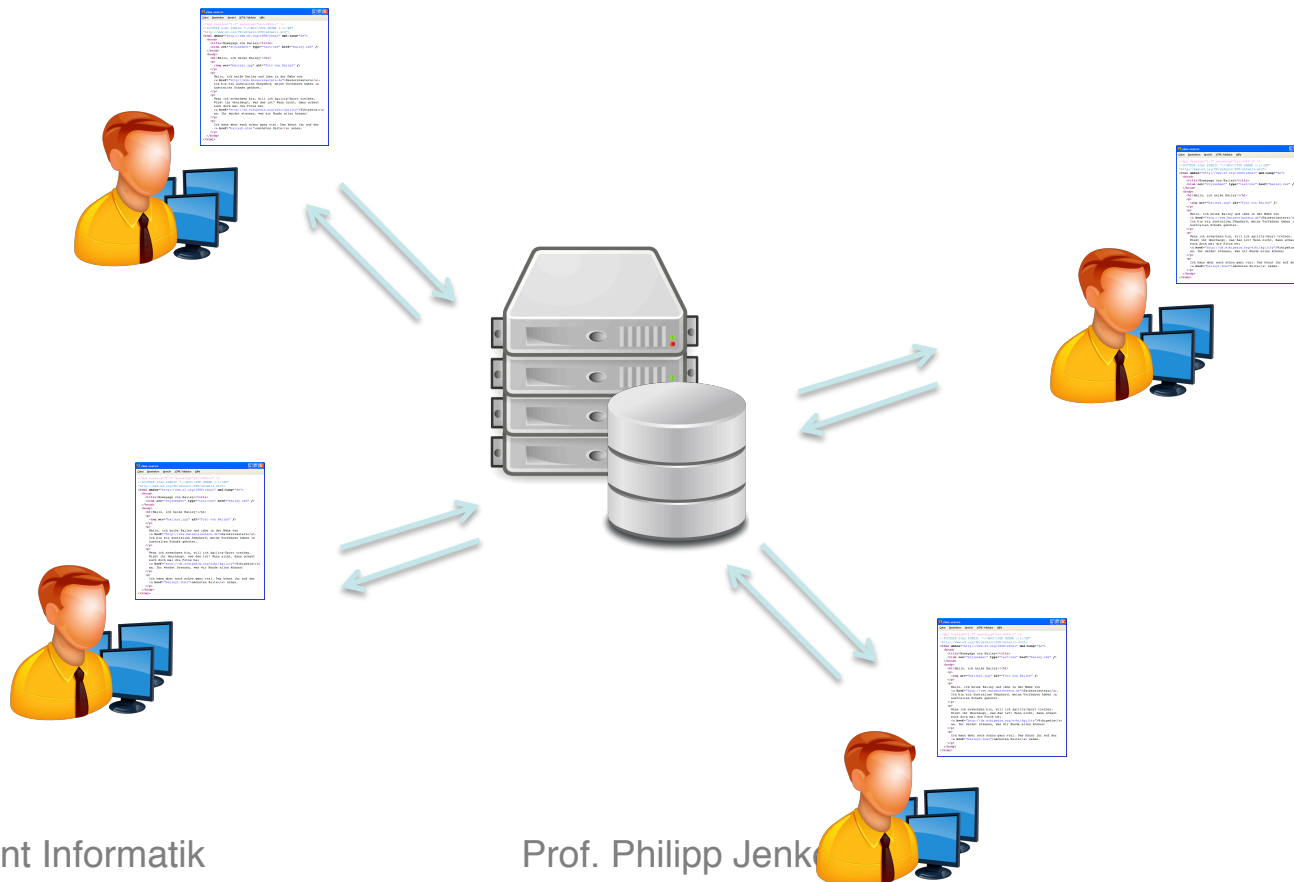


commit



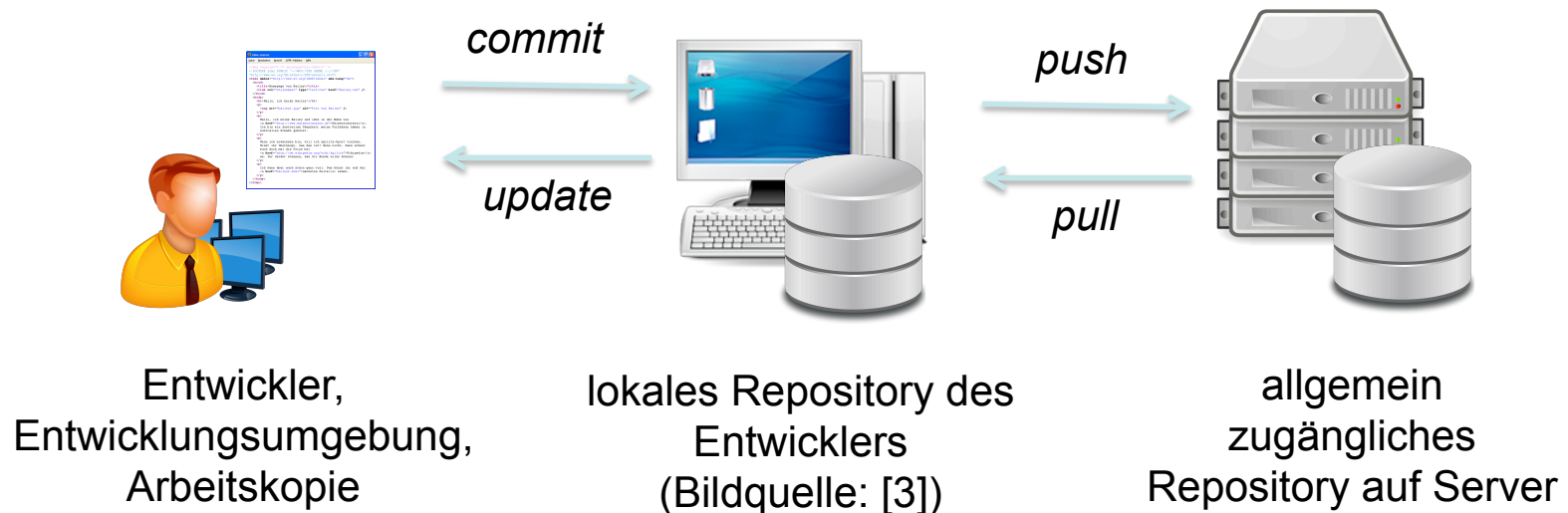


- Zentraler (Server-)Ansatz
- Server synchronisiert die lokalen Kopien aller Entwickler





- mittlerweile üblich
  - dezentrale Versionsverwaltung
  - z.B. Mercurial, GIT



Git

```
$ git config --global user.name "Your Name"  
$ git config --global user.email  
"your_email@whatever.com"
```

Name und E-Mail-Adresse angeben

```
$ git init
```

Lokales Git-Repository erstellen

working

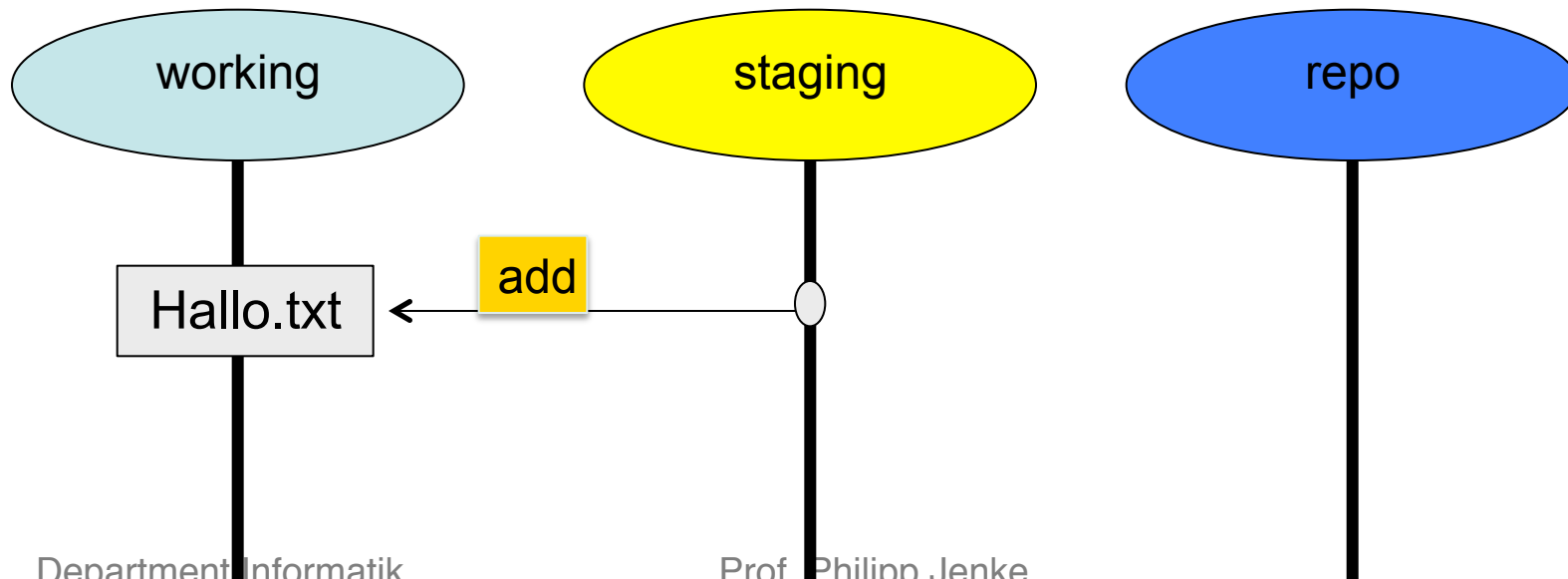
Hallo.txt

staging

repo

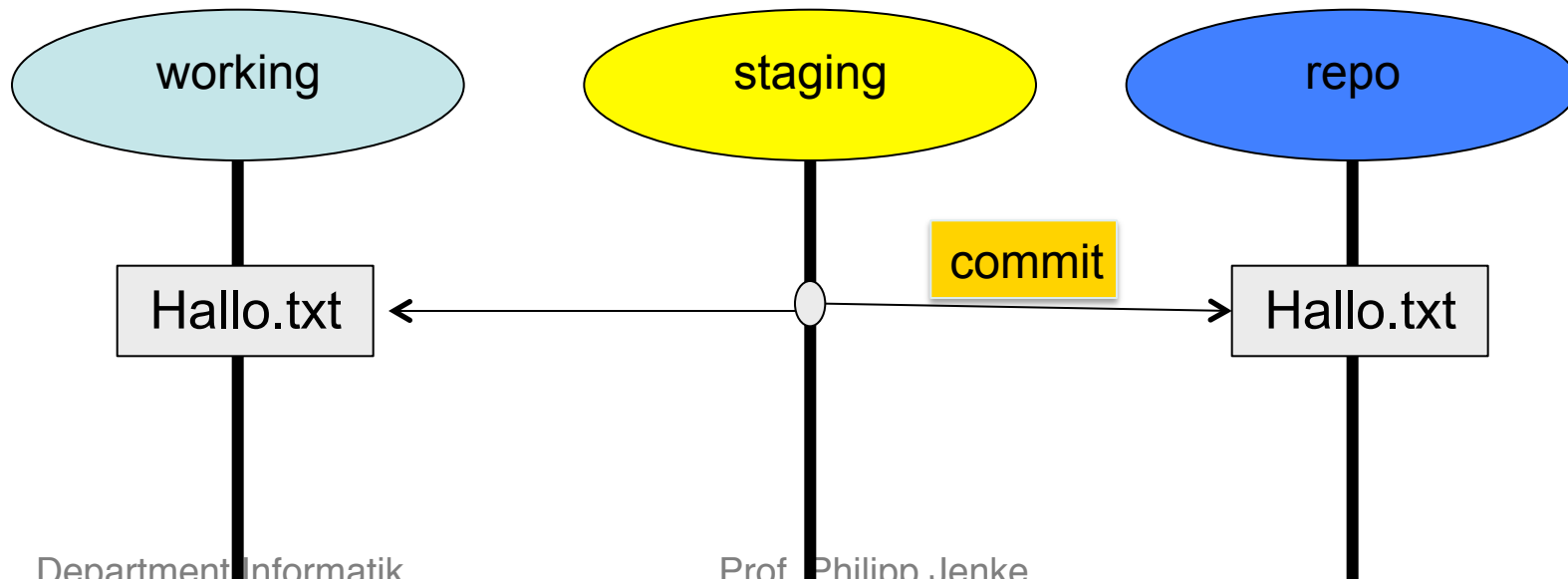
```
$ git add Hallo.txt
```

Zum Index hinzufügen (stagen)



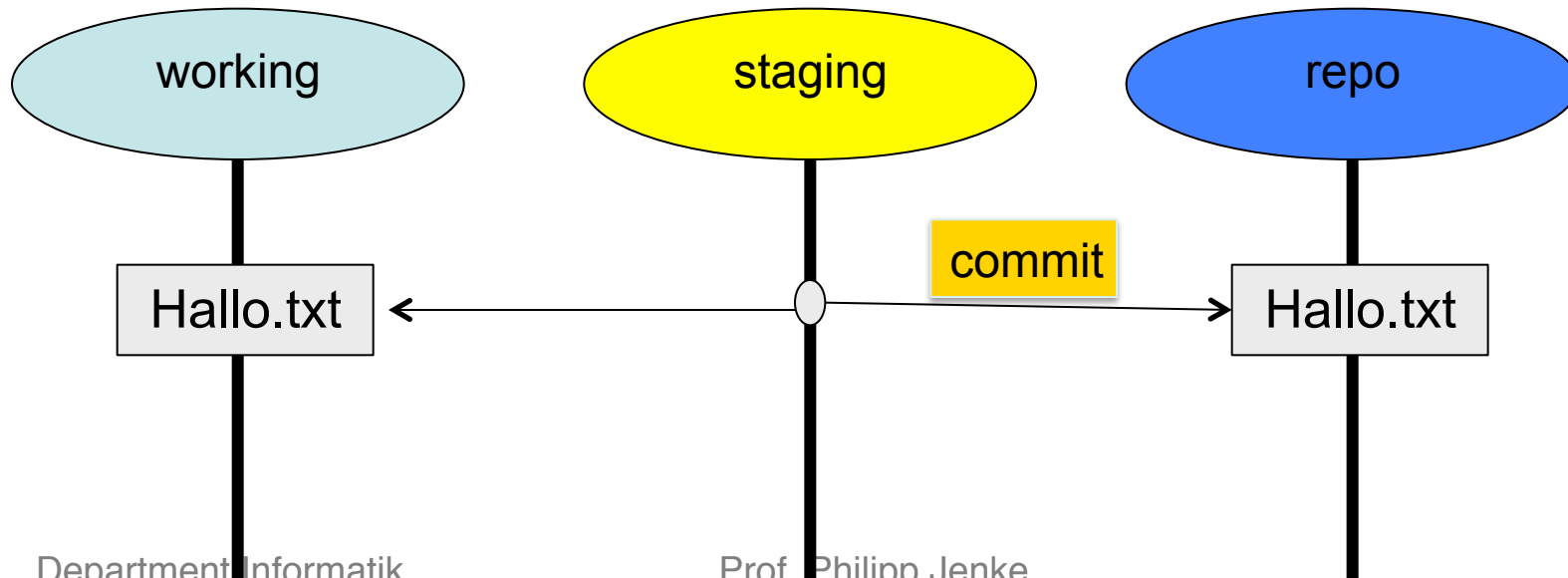
```
$ git commit -m "first Hallo"
```

Auf das lokale Repo schreiben



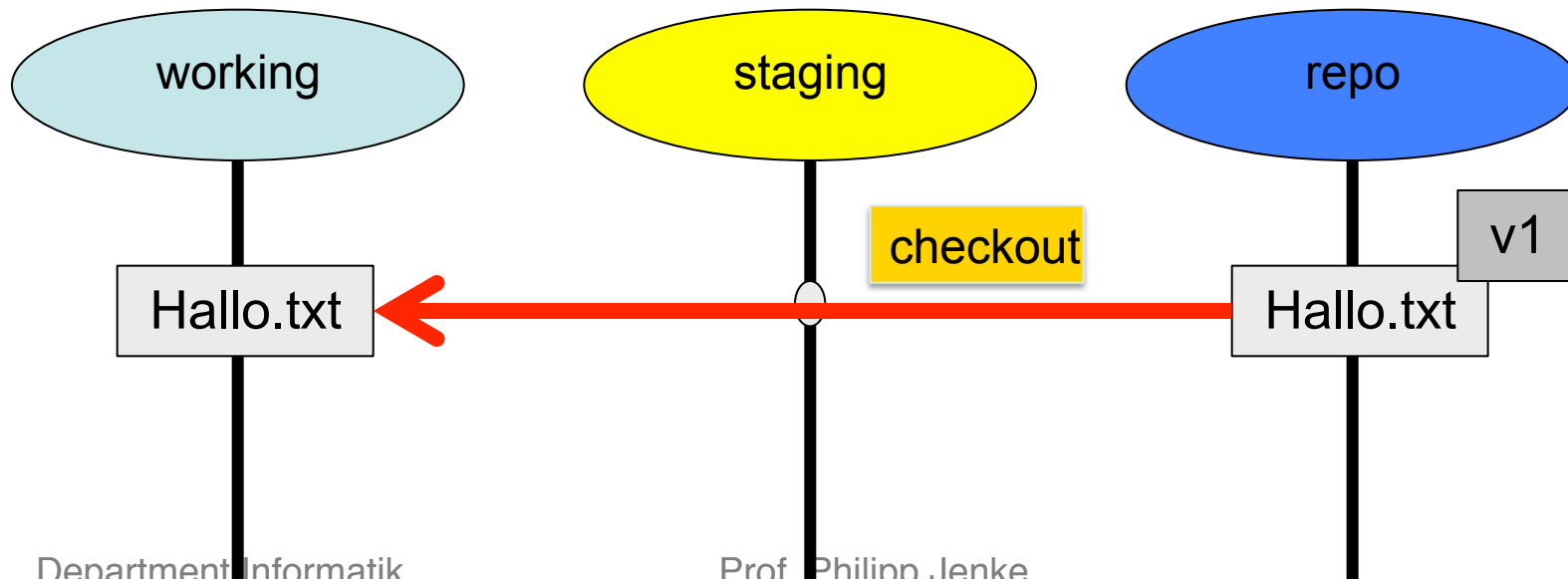
```
$ git status  
# on branch master  
Nothing to commit
```

Status des Staging-Bereichs angeben



```
$ edit Hallo.txt  
$ git checkout master .
```

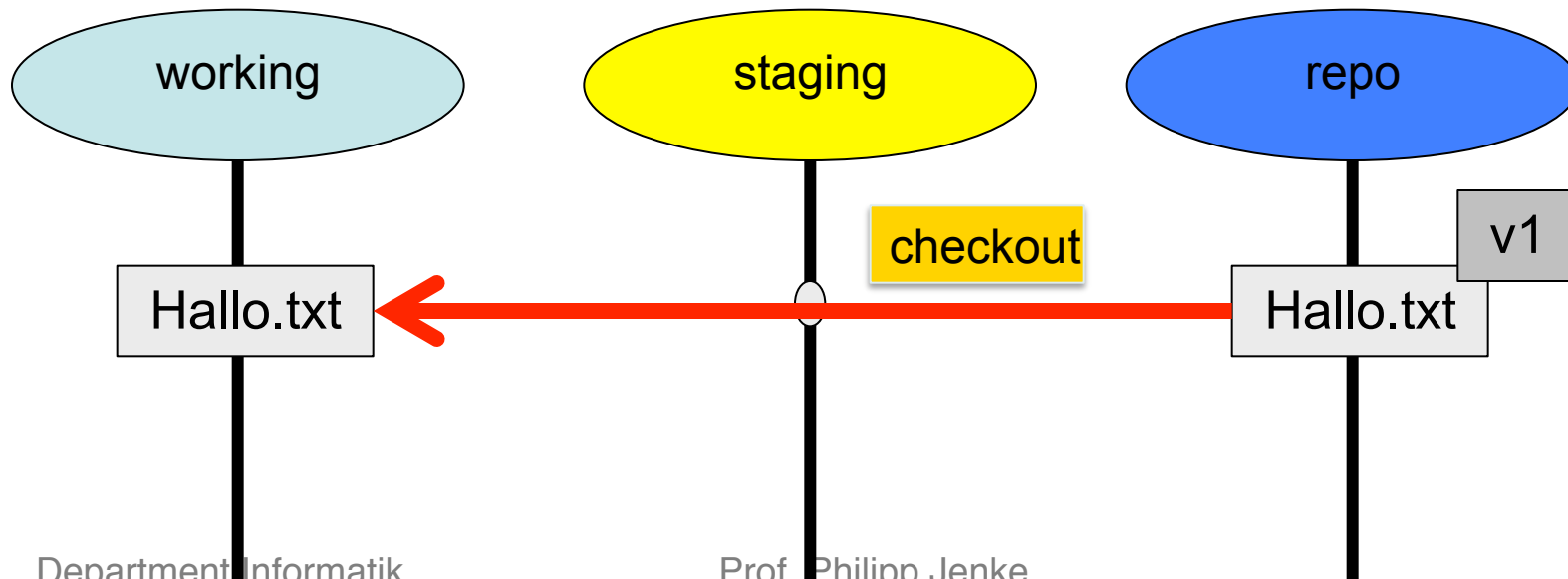
Hallo.txt verändern und Veränderung rückgängig machen





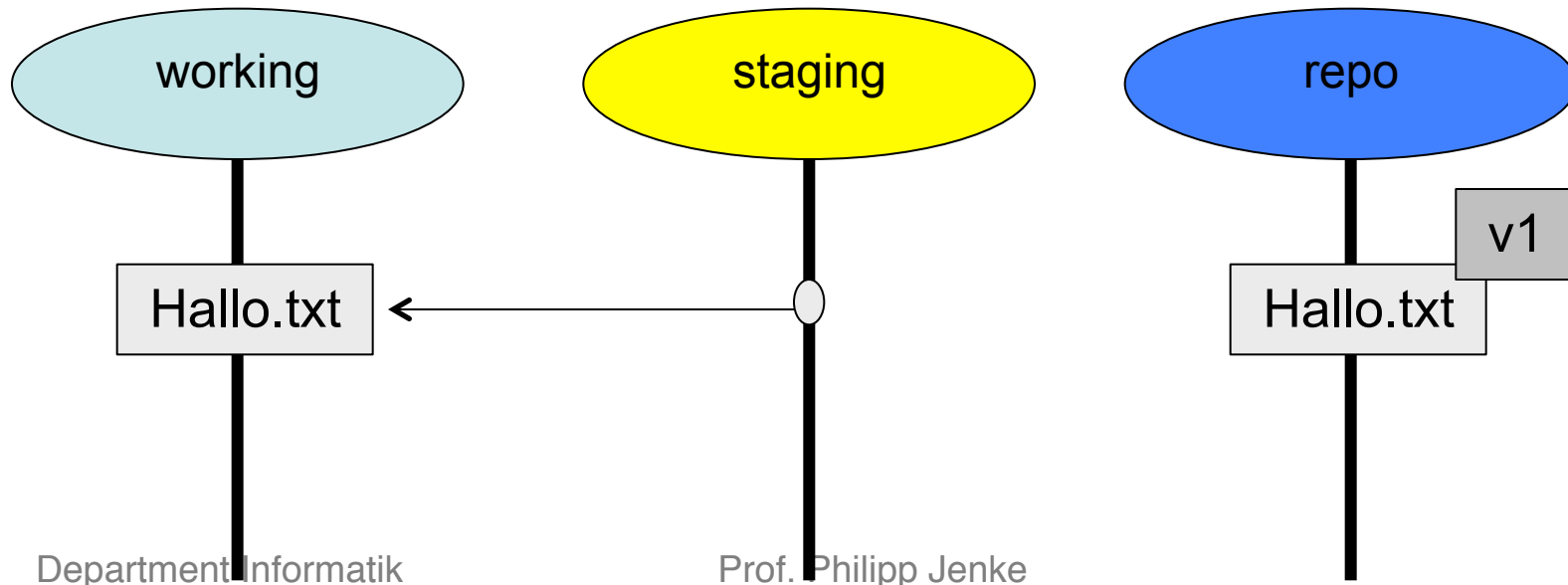
```
$ edit Hallo.txt  
$ git checkout master  
$ git checkout "Hallo.txt"
```

Hallo.txt verändern und Veränderung rückgängig machen



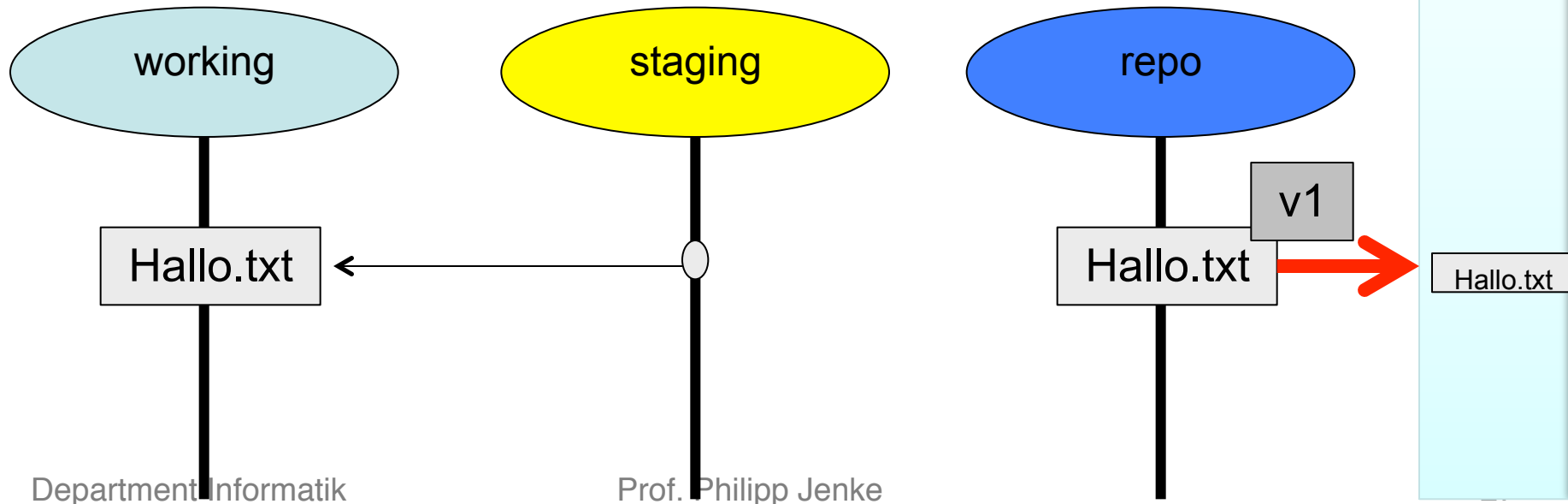
```
$ git remote add origin https://github.com
```

Entferntes Repo angeben



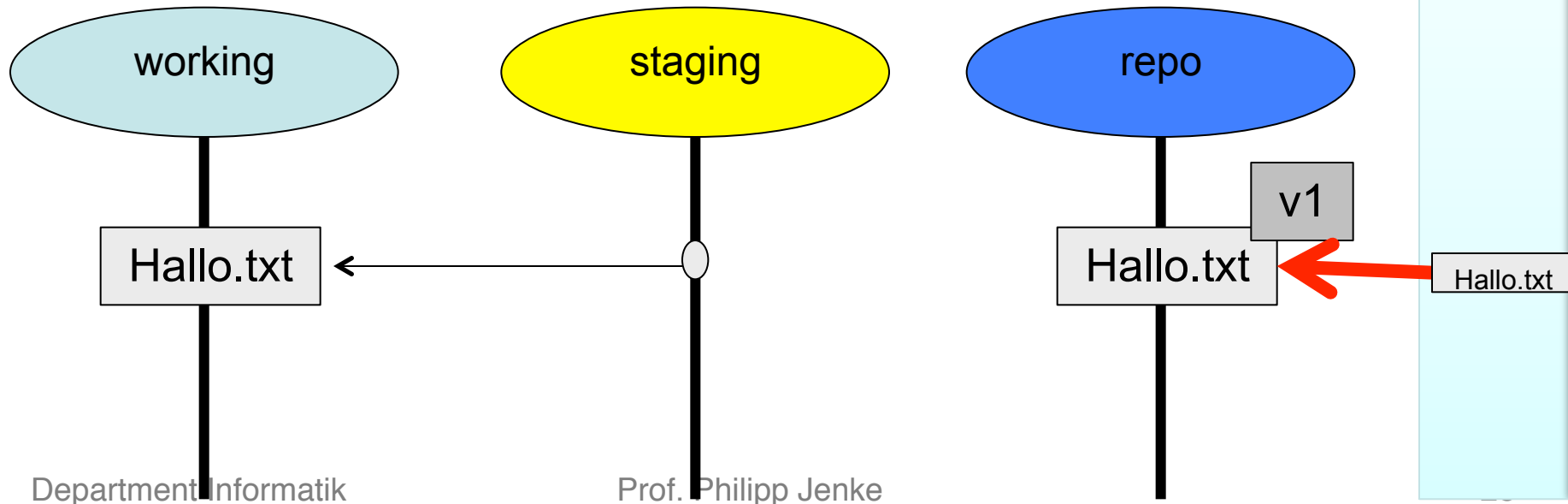
```
$ git push -u origin master
```

Aufs entfernte Repo schreiben



```
$ git pull origin master
```

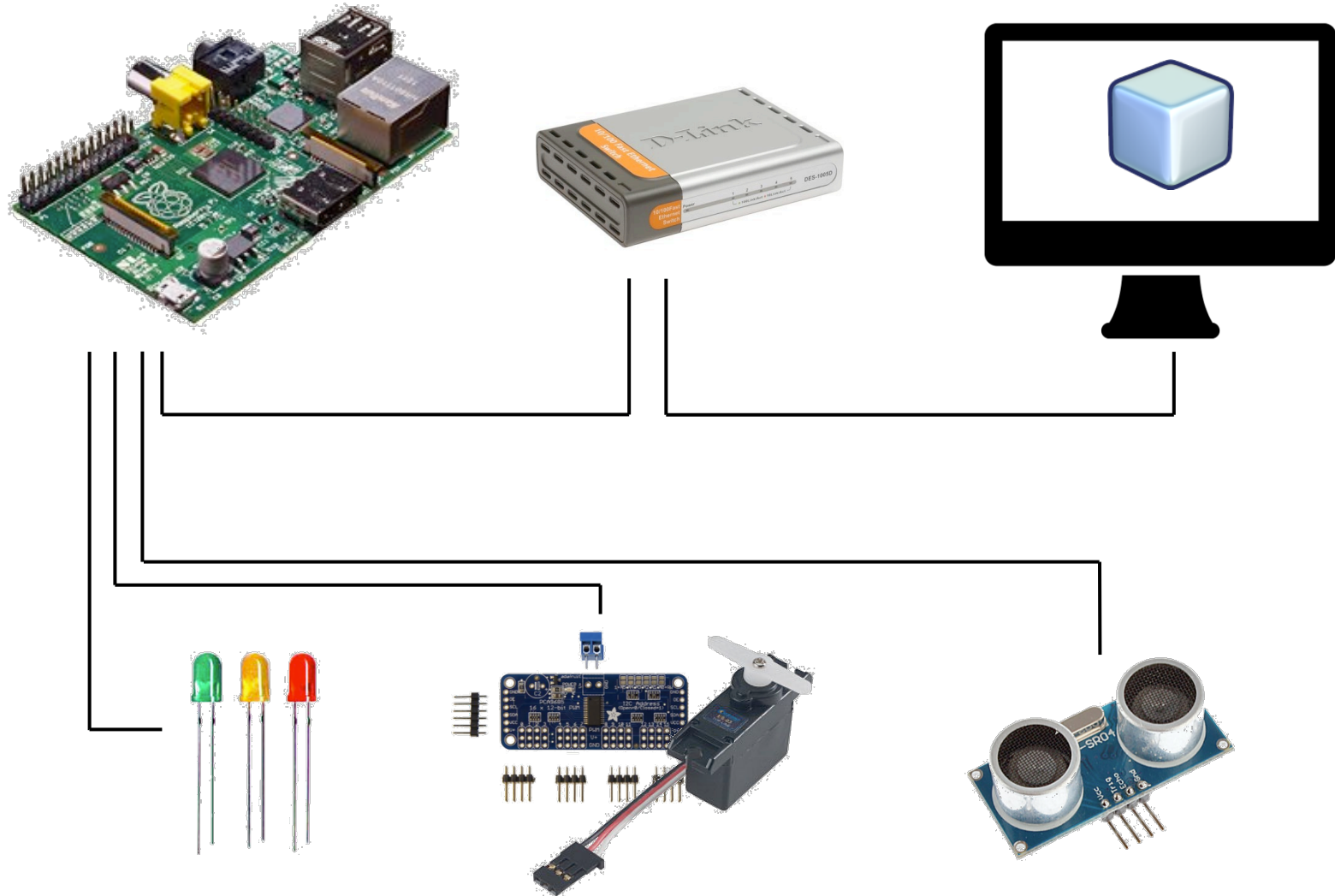
Vom entfernten Repo holen



- Eigenes Netzlaufwerk
- Bitbucket (<https://bitbucket.org/>): Kostenlos für private Nutzung
- Github (<https://github.com/>): Kostenlos bei öffentlichen Repositories
- Sourceforge (<http://sourceforge.net/>): Open Source Projekte
- Angebot der HAW Informatik

Live Demo: Raspberry Pi

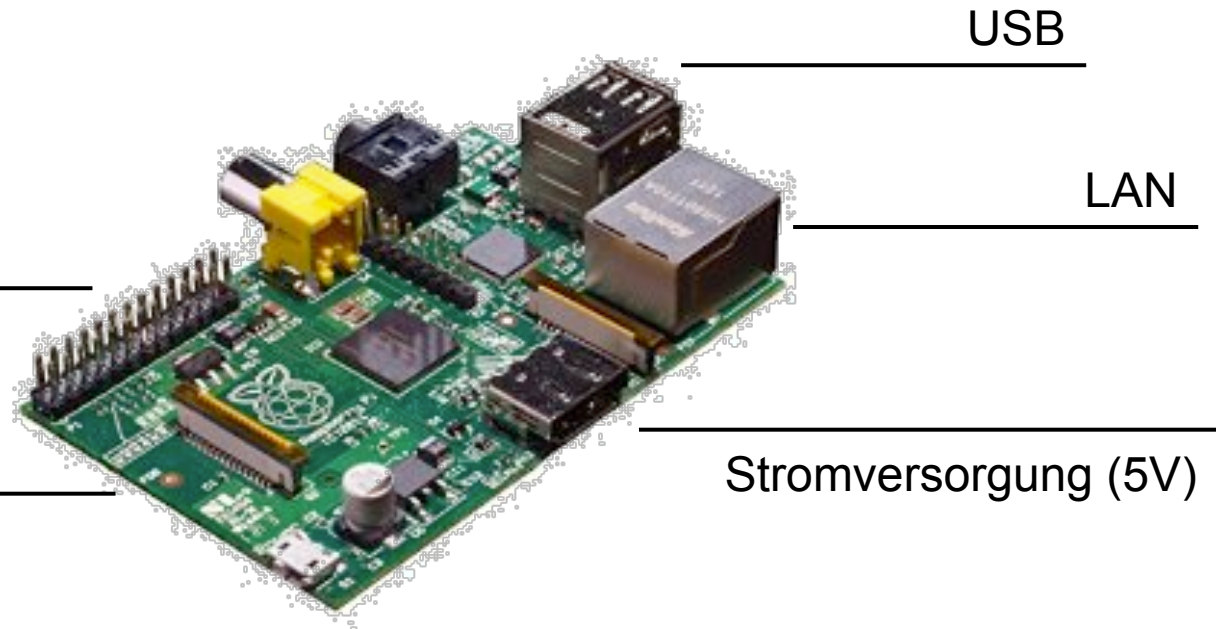
# Aufbau



# Raspberry Pi



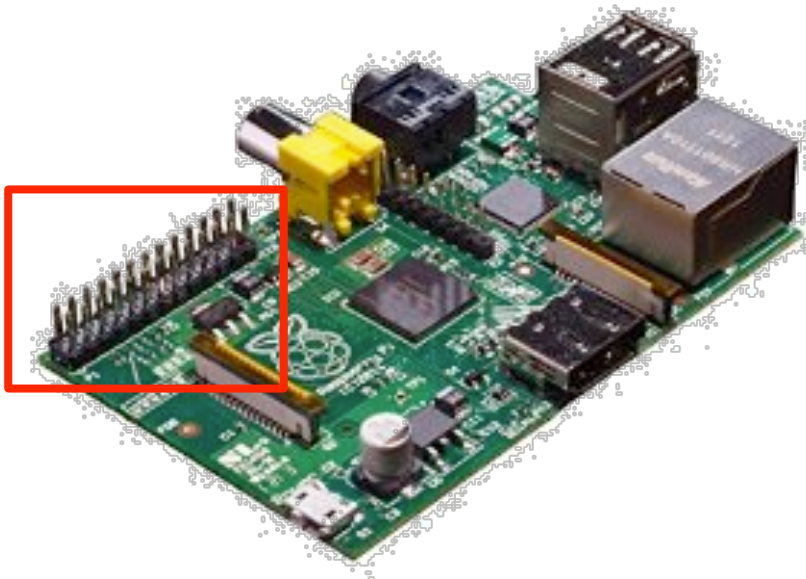
HAW Hamburg  
Fakultät TI  
Technik und Informatik





# GPIO (General Purpose I/O)

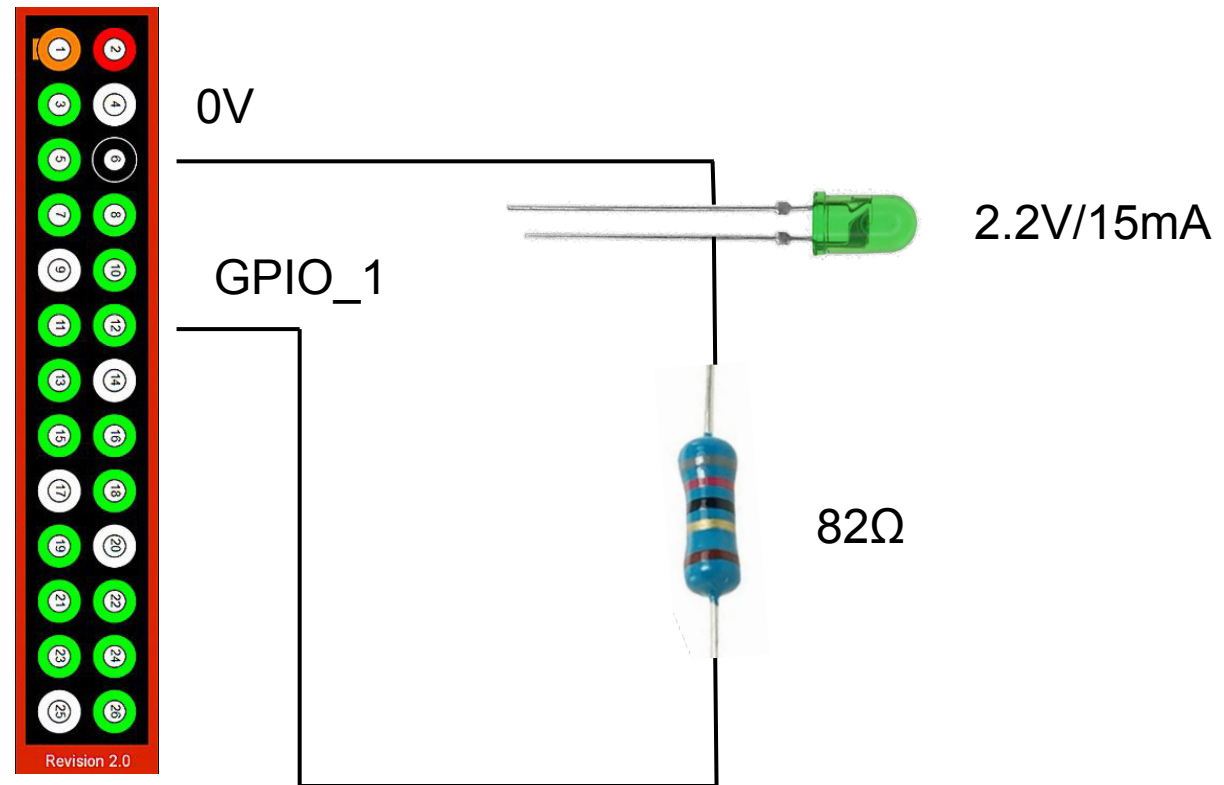
- Bibliothek zur Ansteuerung
  - wiringPi (<http://wiringpi.com/>)



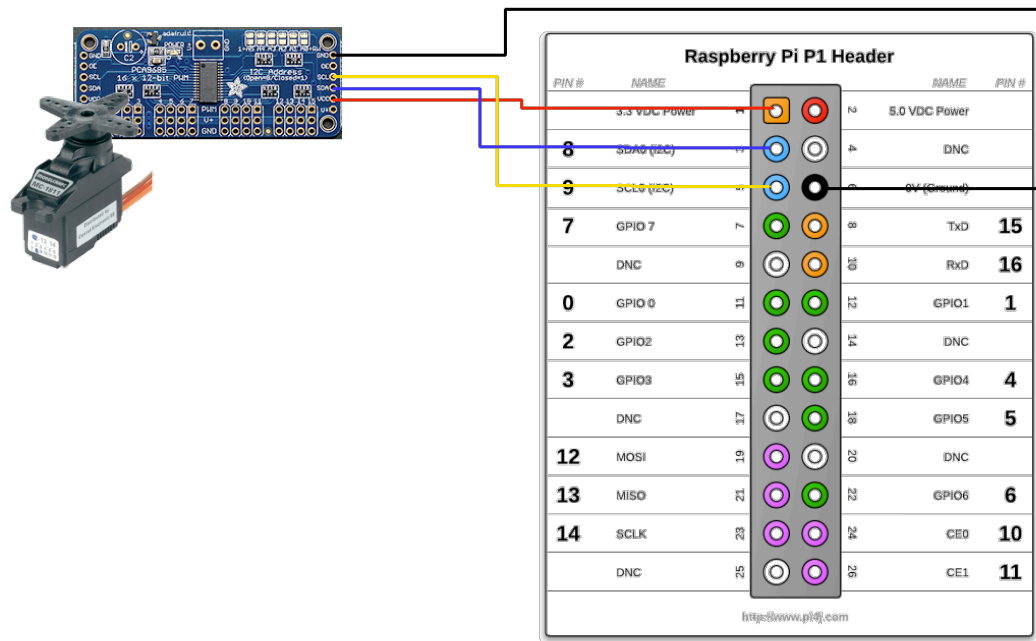
P1: The Main GPIO connector						
WiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	WiringPi Pin
		3.3v	1 2	5v		
8	Rv1:0 - Rv2:2	SDA	3 4	5v		
9	Rv1:1 - Rv2:3	SCL	5 6	0v		
7	4	GPIO7	7 8	TxD	14	15
		0v	9 10	RxD	15	16
0	17	GPIO0	11 12	GPIO1	18	1
2	Rv1:21 - Rv2:27	GPIO2	13 14	0v		
3	22	GPIO3	15 16	GPIO4	23	4
		3.3v	17 18	GPIO5	24	5
12	10	MOSI	19 20	0v		
13	9	MISO	21 22	GPIO6	25	6
14	11	SCLK	23 24	CE0	8	10
		0v	25 26	CE1	7	11
WiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	WiringPi Pin

- Verwendung in Java
  - Wrapper: Pi4J (<http://pi4j.com/>)

# Projekt 1: LED



# Projekt 2: Servo



# Projekt 3: Ultraschallsensor

