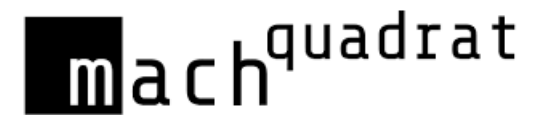
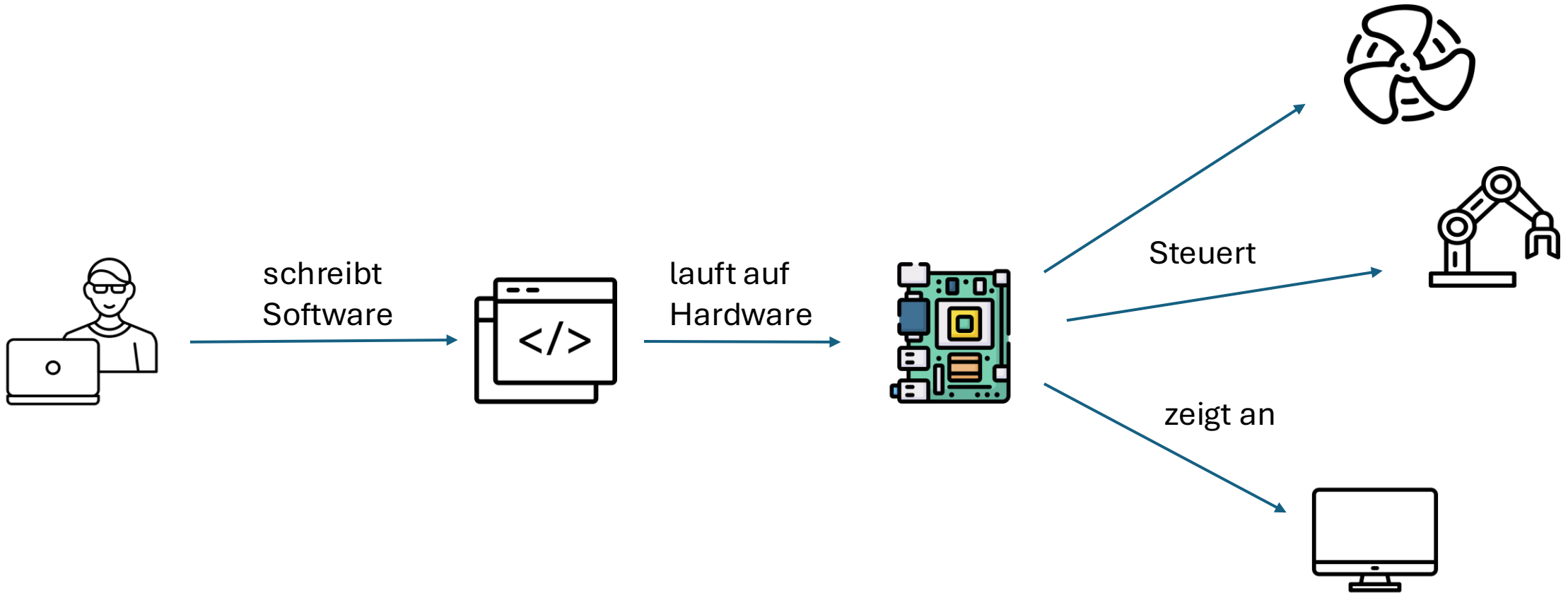




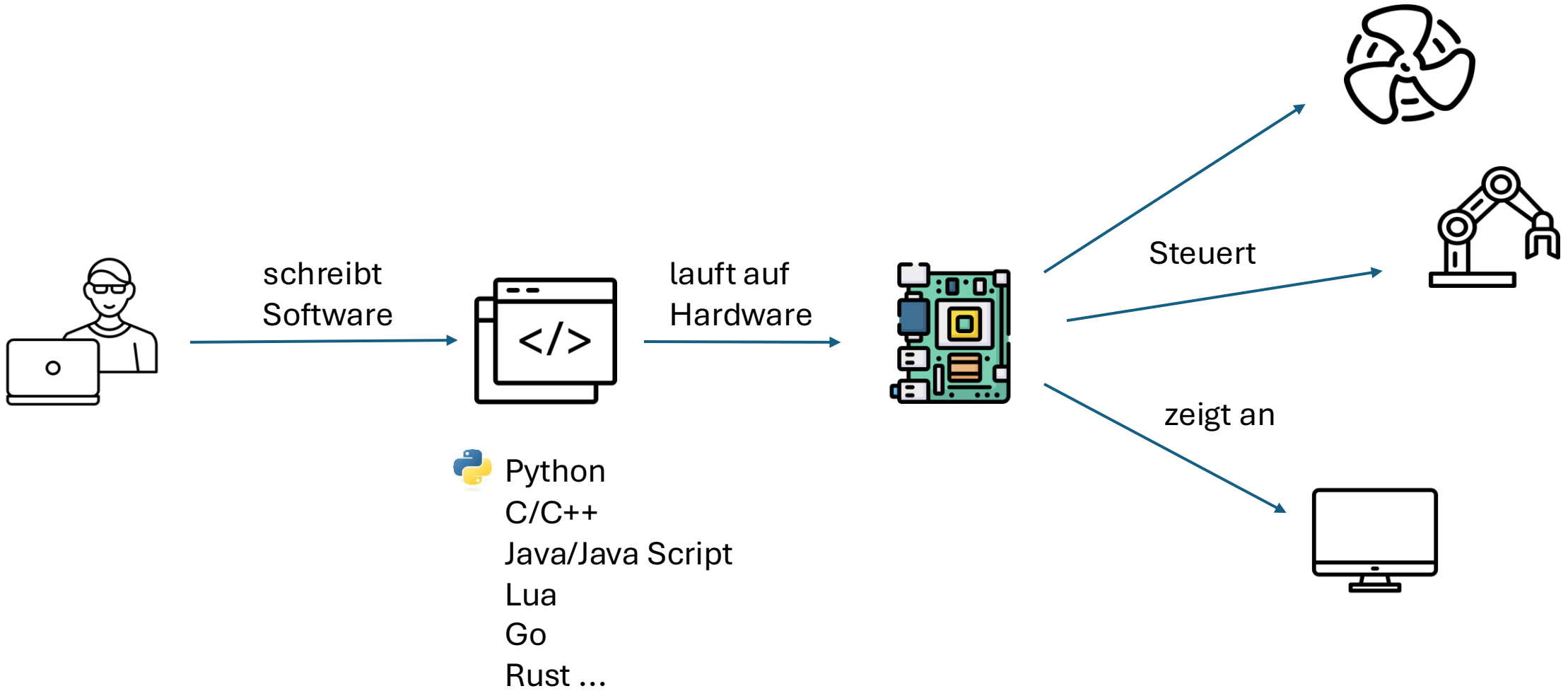
# Python Grundkurs

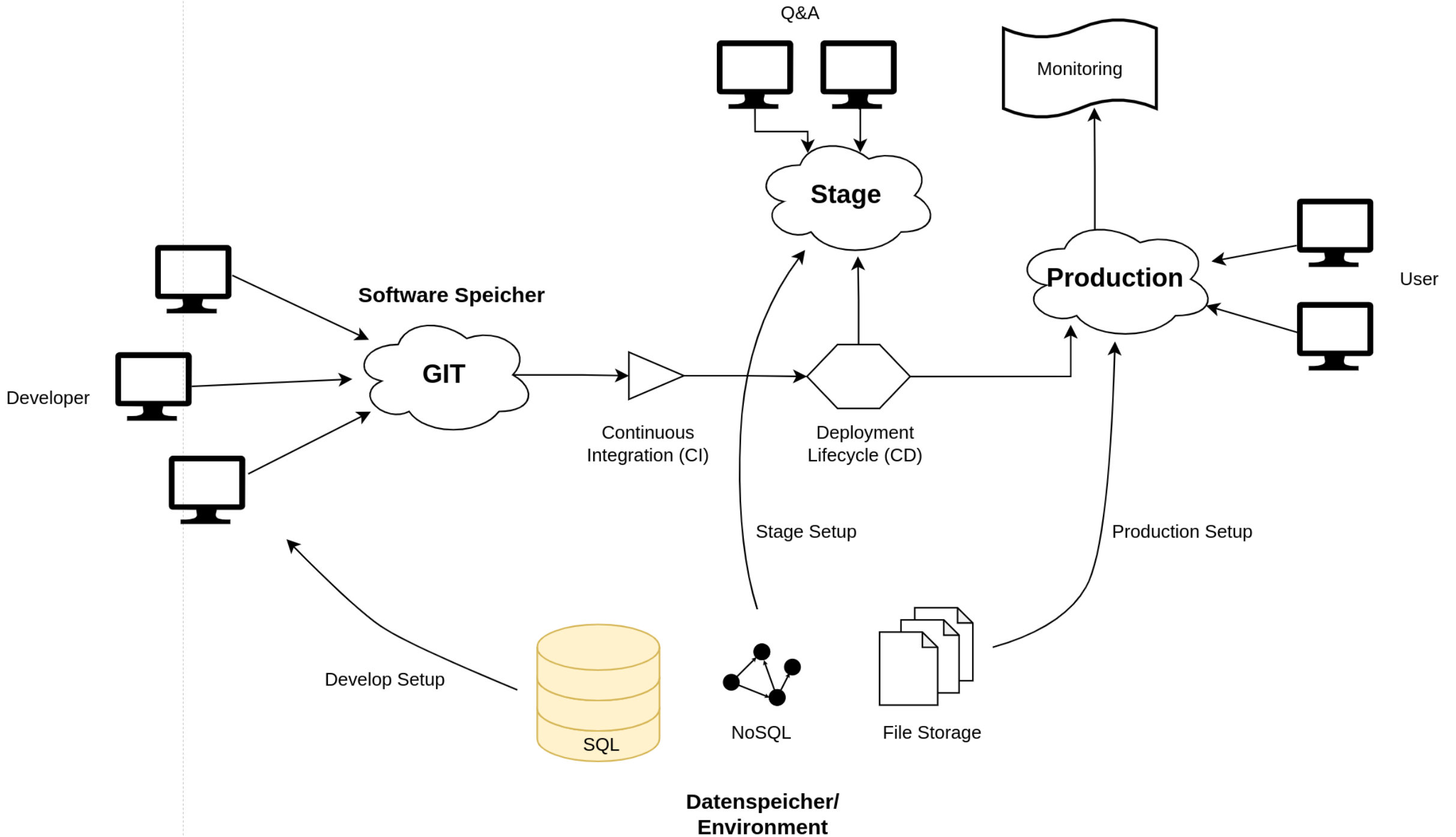


# Software Entwicklung



# Software Entwicklung





# Programmiersprachen

Set von Befehlen um Abläufe zu definieren

## Befehle:

schreibe den Wert 5 auf die Speicherstelle 0, (  $a = 5$  )

Speicher

|   |   |
|---|---|
| 0 | 5 |
| 1 |   |
| 2 |   |
| 3 |   |
| 4 |   |

# Programmiersprachen

Set von Befehlen um Abläufe zu definieren

## Befehle:

schreibe den Wert 5 auf die Speicherstelle 0, ( `a = 5` )

Speicher die Benutzereingabe auf die Speicherstelle 1, ( `b = input()` )

Speicher

|   |   |
|---|---|
| 0 | 5 |
| 1 | 7 |
| 2 |   |
| 3 |   |
| 4 |   |

# Programmiersprachen

Set von Befehlen um Abläufe zu definieren

## Befehle:

schreibe den Wert 5 auf die Speicherstelle 0, (  $a = 5$  )

Speicher die Benutzereingabe auf die Speicherstelle 1, (  $b = \text{input}()$  )

Addiere Speicherstelle 0, Speicherstelle 1 und schreibe es auf Speicherstelle 2, (  $c = a + b$  )

Speicher

|   |    |
|---|----|
| 0 | 5  |
| 1 | 7  |
| 2 | 12 |
| 3 |    |
| 4 |    |

# Ausführungsarten

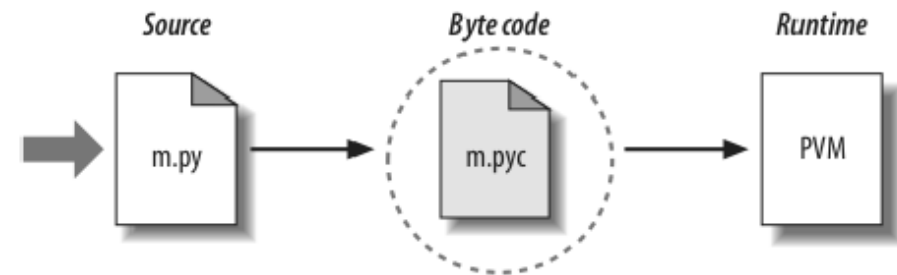
- Direkte Execution auf der Hardware

C/C++, Rust, Go, ....

```
;- main:
0x0042a950    push    r15
0x0042a952    push    r14
0x0042a954    mov     r14, rsi
0x0042a957    push    r13
0x0042a959    push    r12
0x0042a95b    mov     esi, 1
0x0042a960    push    rbp
0x0042a961    push    rbx
```

- Execution über Virtualisierung

Python, Java, C#, ....



Komplexität versteckt in der Runtime

- Multi Architektur



# Live Coding

Übungen und Slides: [github.com/kernware/Presentations](https://github.com/kernware/Presentations)

- Jupyter Notebook

Login: 10.13.37.64:9002 (Passwort: password)