



# `git clone` deine Entwicklungsumgebung

Deklarative und reproduzierbare lokale Entwicklungsumgebungen mit Codchi

*"Codchi is a tool that manages your project's development environment in a reproducible and easy-to-use way. **Setting up a development environment should be as easy as a `git clone`!**"*

---

Johannes Hüttinger

 [codchi.dev](https://codchi.dev)

 [aformatik – Training & Consulting](https://aformatik.de)  
[www.aformatik.de](https://www.aformatik.de)

# Wozu Entwicklungsumgebungen?

# Wozu Entwicklungsumgebungen?

## 1. Tools zur Entwicklung eines Projektes bereitstellen

- Wartung: Erstellung, Updates, Rollbacks
- Automatisierung, Reproduzierbarkeit

# Wozu Entwicklungsumgebungen?

## 1. Tools zur Entwicklung eines Projektes bereitstellen

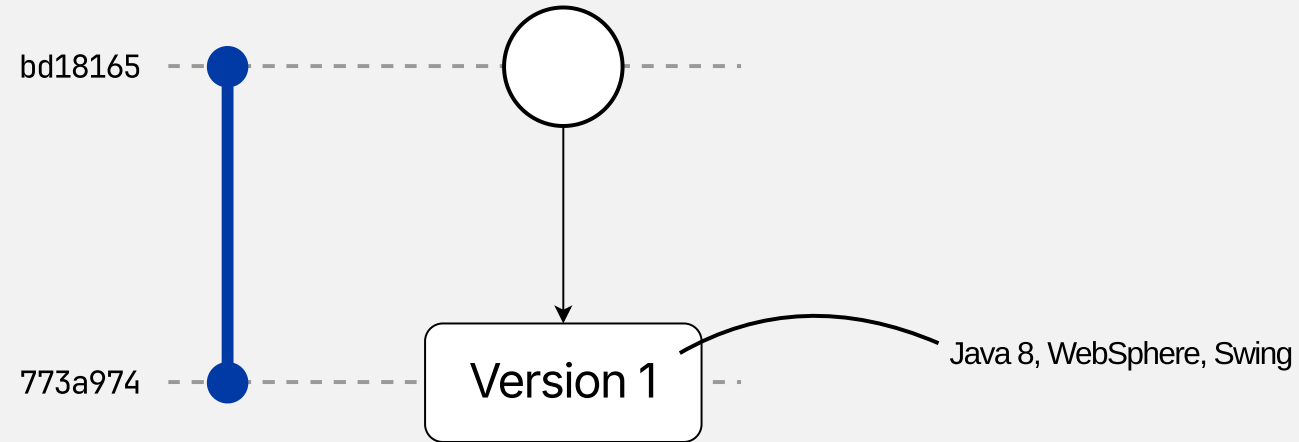
- Wartung: Erstellung, Updates, Rollbacks
- Automatisierung, Reproduzierbarkeit

## 2. Developer Experience - Entwickler entwickeln lassen

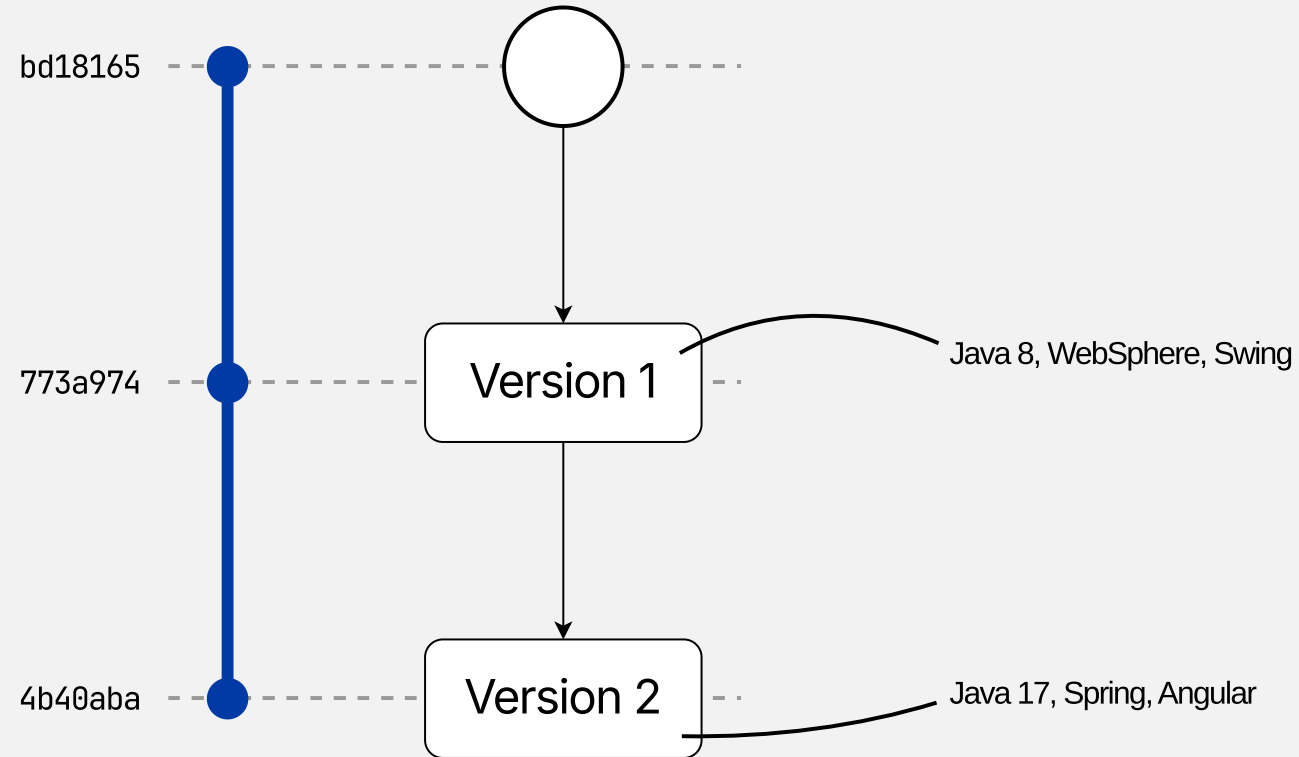
- Integration
- Performance
- Individualisierung

# Die Entwicklung einer Entwicklungsumgebung

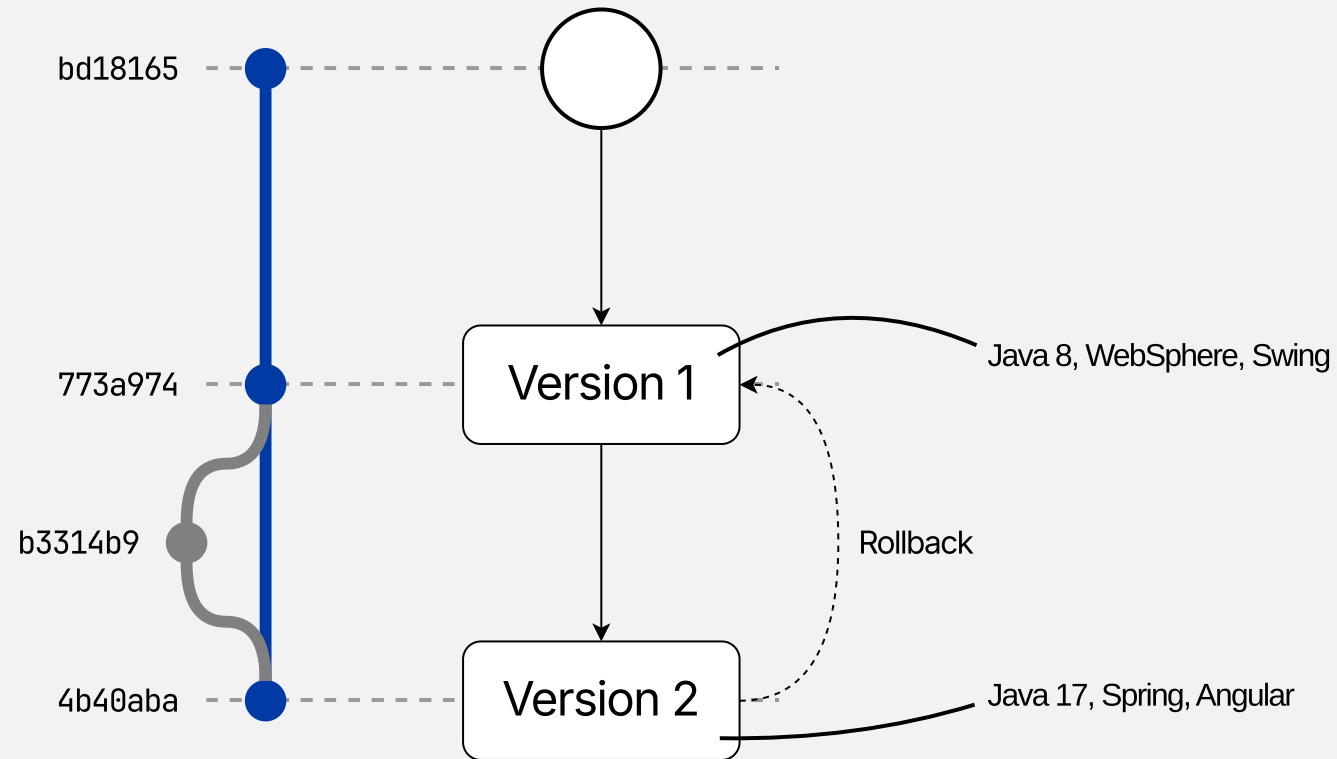
# Die Entwicklung einer Entwicklungsumgebung



# Die Entwicklung einer Entwicklungsumgebung

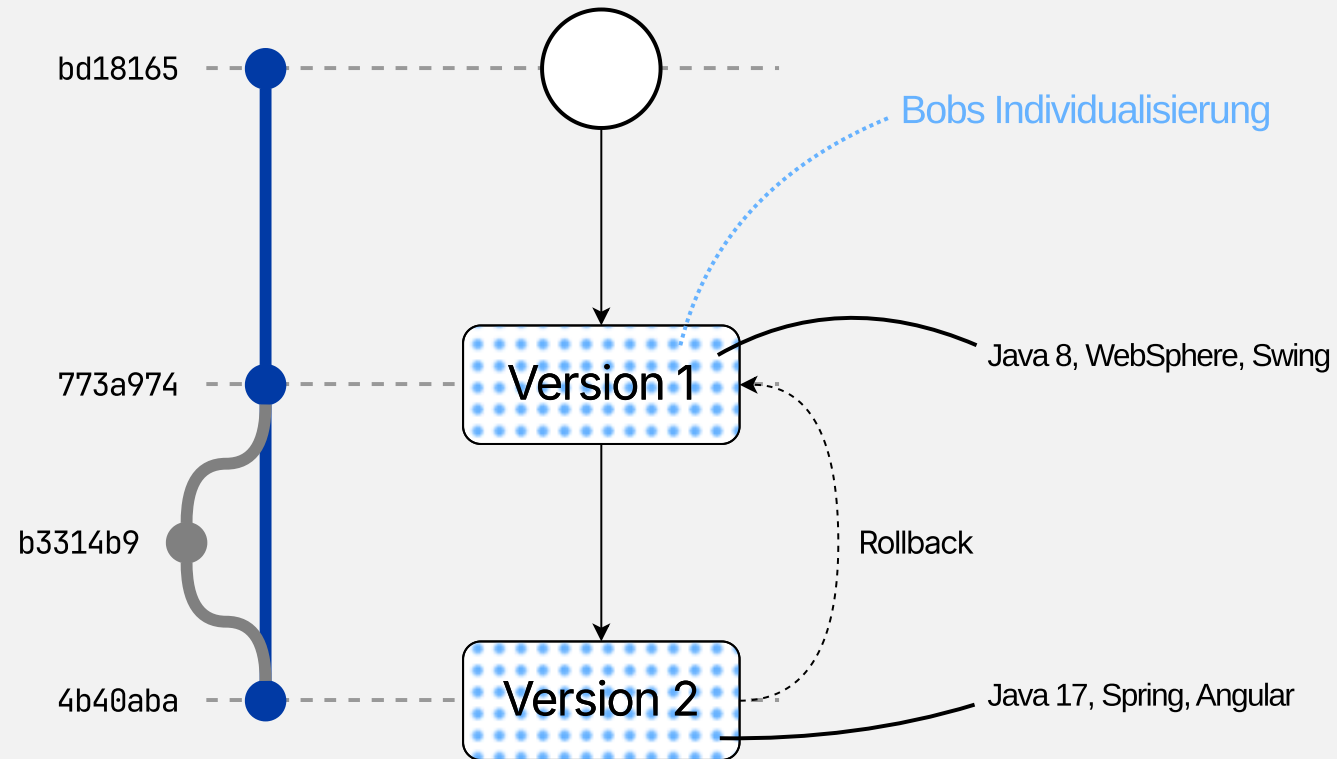


# Die Entwicklung einer Entwicklungsumgebung



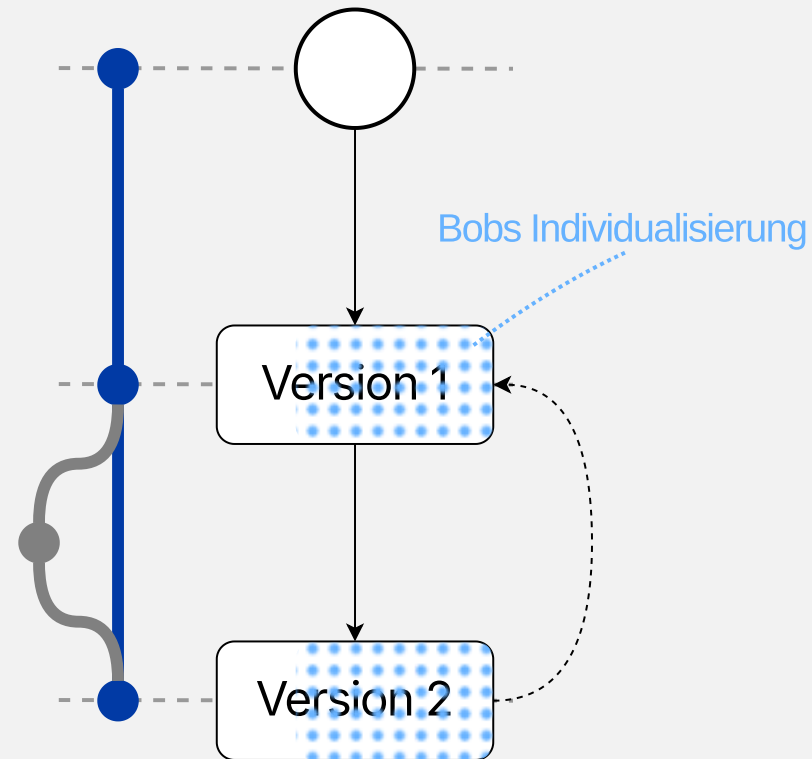


# Die Entwicklung einer Entwicklungsumgebung



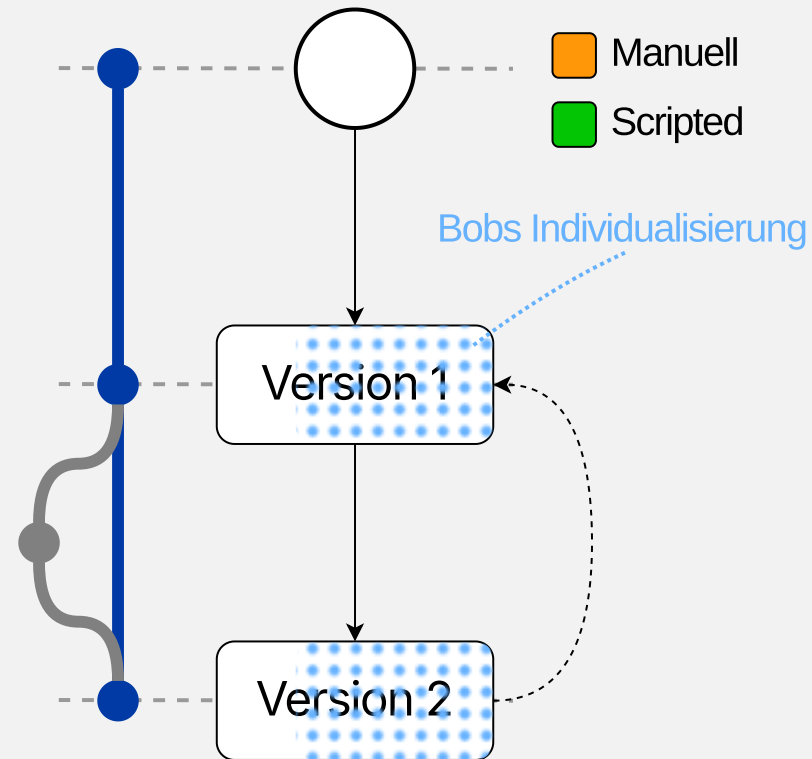
# Entwicklungsumgebungen in der Praxis

- Manuell / Bare Metal
- Virtuelle Maschinen
- VS Code + Docker  
(Devcontainers / Codespaces)
- Vagrant + Ansible
- ...



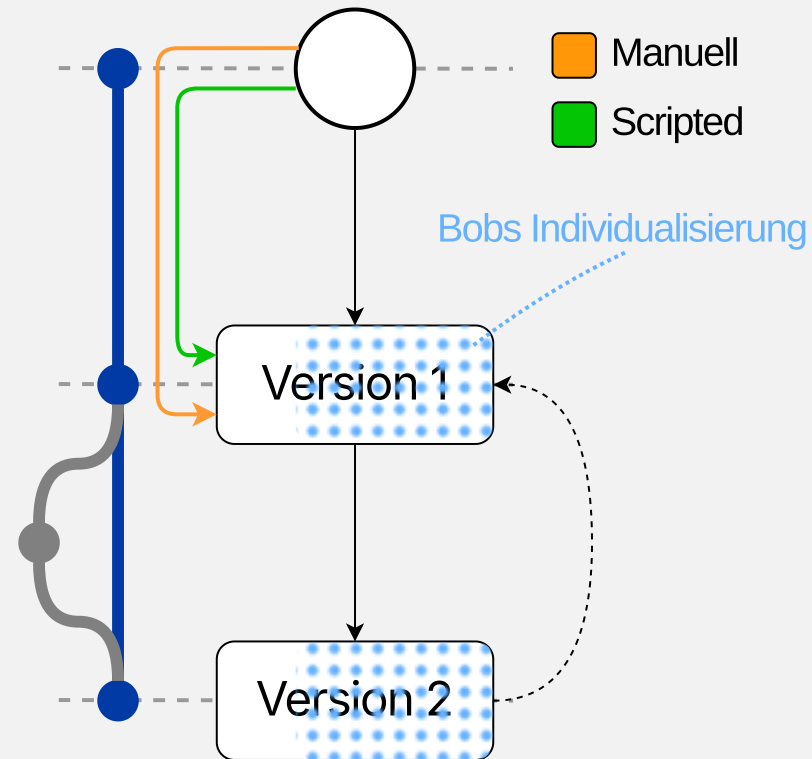
# Entwicklungsumgebungen in der Praxis

- Manuell / Bare Metal
- Virtuelle Maschinen
- VS Code + Docker  
(Devcontainers / Codespaces)
- Vagrant + Ansible
- ...



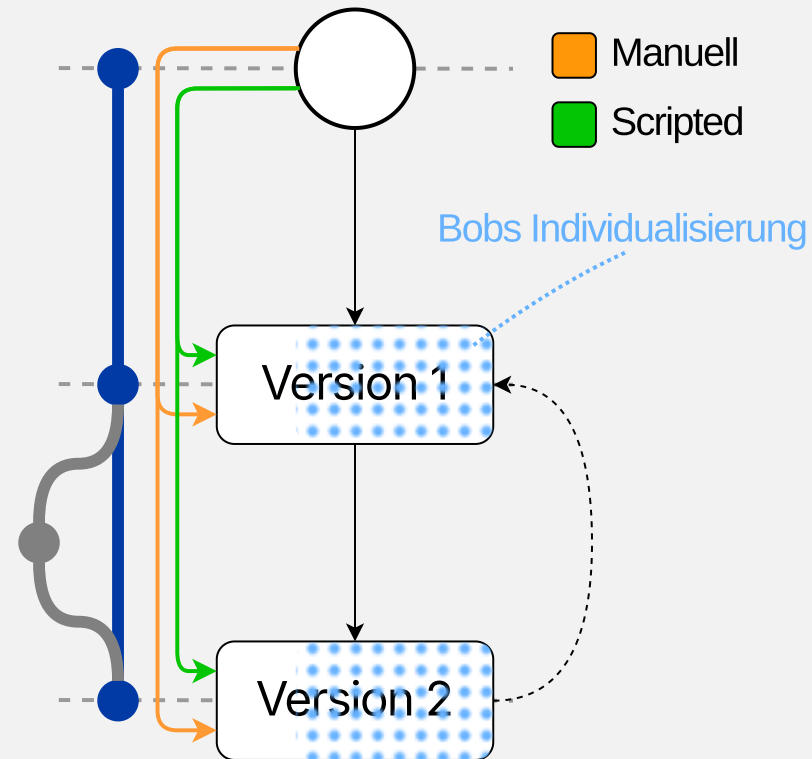
# Entwicklungsumgebungen in der Praxis

- Manuell / Bare Metal
- Virtuelle Maschinen
- VS Code + Docker  
(Devcontainers / Codespaces)
- Vagrant + Ansible
- ...



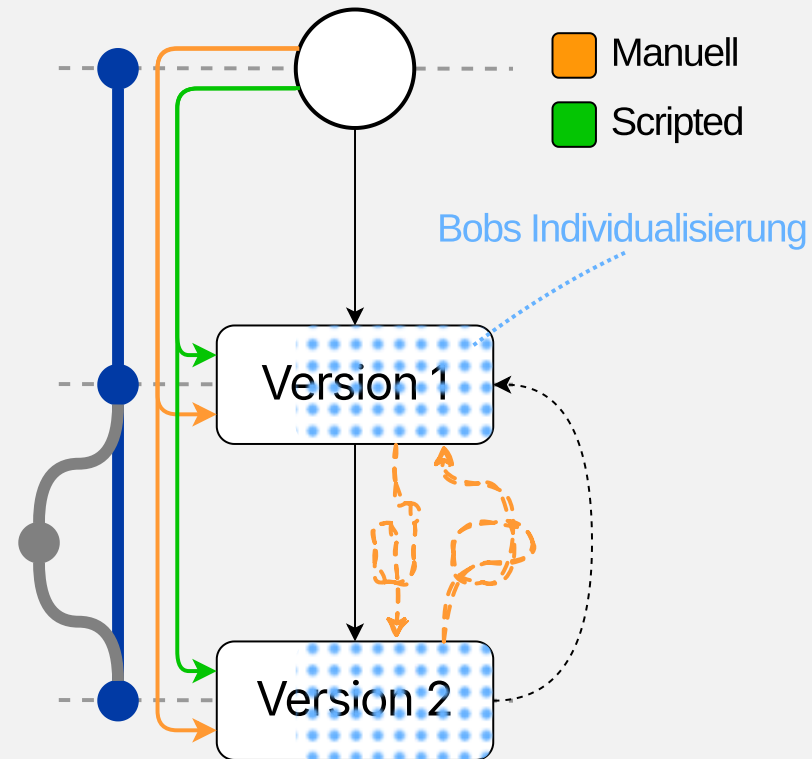
# Entwicklungsumgebungen in der Praxis

- Manuell / Bare Metal
- Virtuelle Maschinen
- VS Code + Docker  
(Devcontainers / Codespaces)
- Vagrant + Ansible
- ...



# Entwicklungsumgebungen in der Praxis

- Manuell / Bare Metal
- Virtuelle Maschinen
- VS Code + Docker  
(Devcontainers / Codespaces)
- Vagrant + Ansible
- ...



# Imperativ vs. Funktional

# Imperativ vs. Funktional

```
static int x;  
  
int add(int a, int b) {  
    ...  
    Date.now()  
    ...  
    new Random()  
    ...  
    x++  
    ...  
}
```



# Imperativ vs. Funktional

```
static int x;  
  
int add(int a, int b) {  
    ...  
    Date.now()  
    ...  
    new Random()  
    ...  
    x++  
    ...  
}
```

- **Mutable** by default
- Uneingeschränkte Seiteneffekte

# Imperativ vs. Funktional

```
static int x;  
  
int add(int a, int b) {  
    ...  
    Date.now()  
    ...  
    new Random()  
    ...  
    x++  
    ...  
}
```

```
plus :: Int → Int → Int  
plus = ...
```

- **Mutable** by default
- Uneingeschränkte Seiteneffekte

# Imperativ vs. Funktional

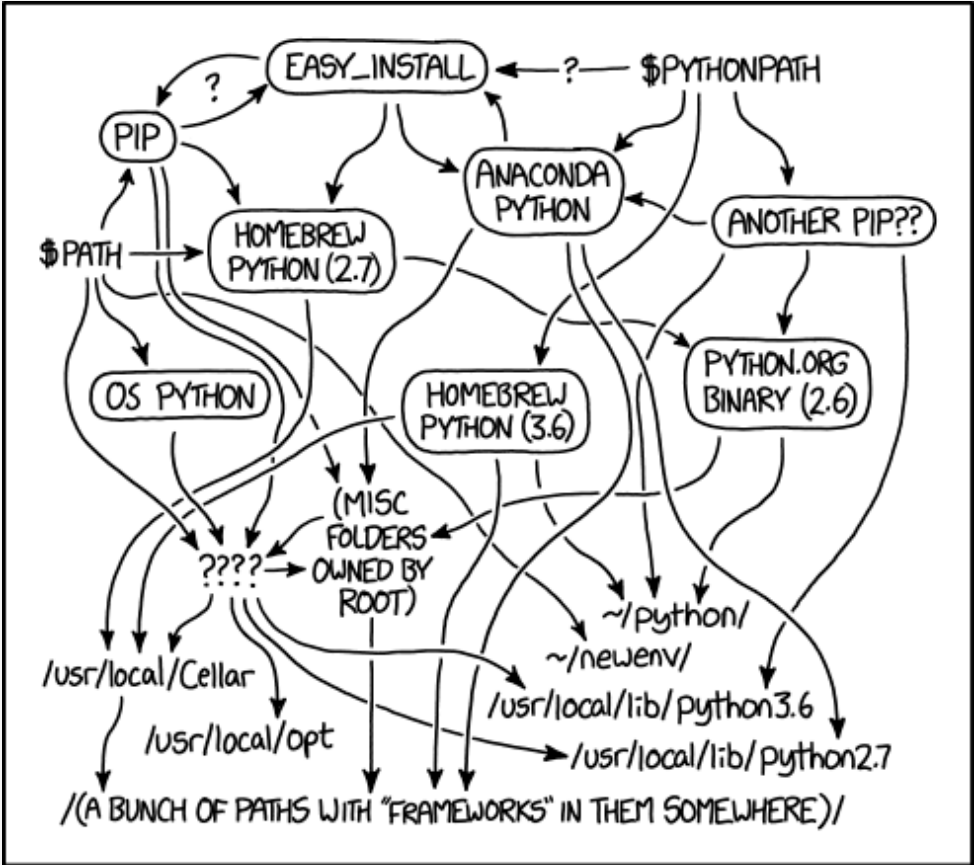
```
static int x;  
  
int add(int a, int b) {  
    ...  
    Date.now()  
    ...  
    new Random()  
    ...  
    x++  
    ...  
}
```

- **Mutable** by default
- Uneingeschränkte Seiteneffekte

```
plus :: Int → Int → Int  
plus = ...
```

- **Immutable** by default
- Referentielle Transparenz

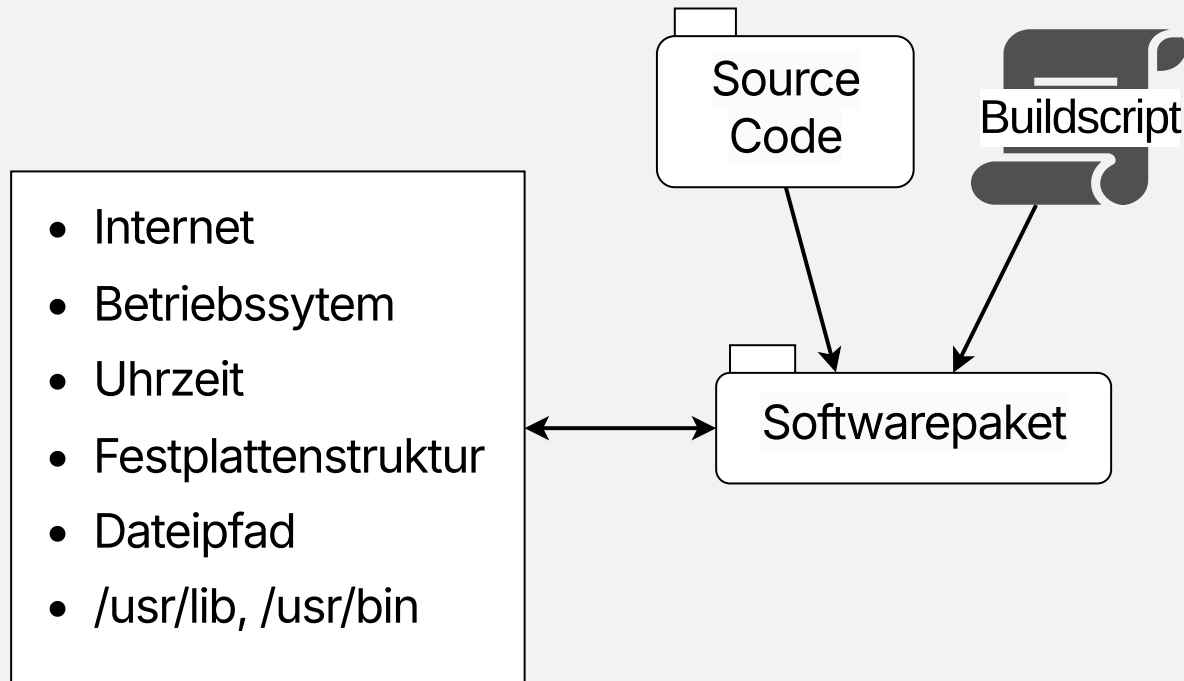
# Imperatives Paketmanagement...



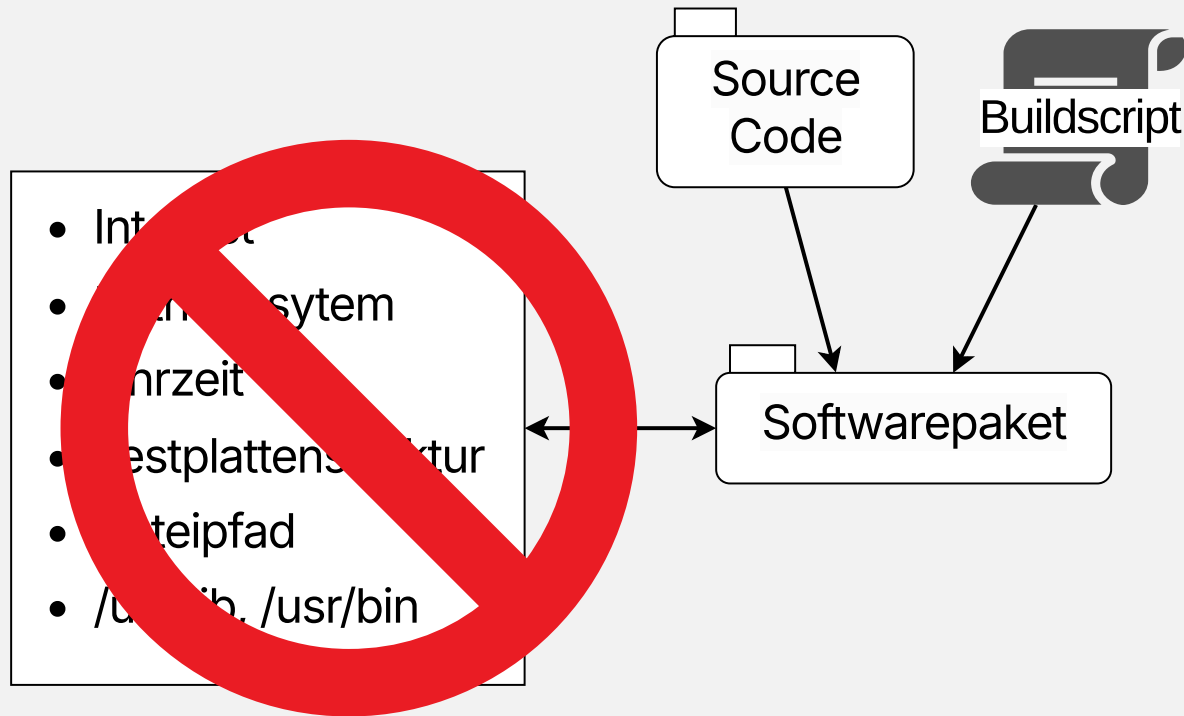
MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED  
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

- Destruktive Inplace-Updates
- Uneingeschränkte Seiteneffekte
- 👉 Zustandsabhängigkeit

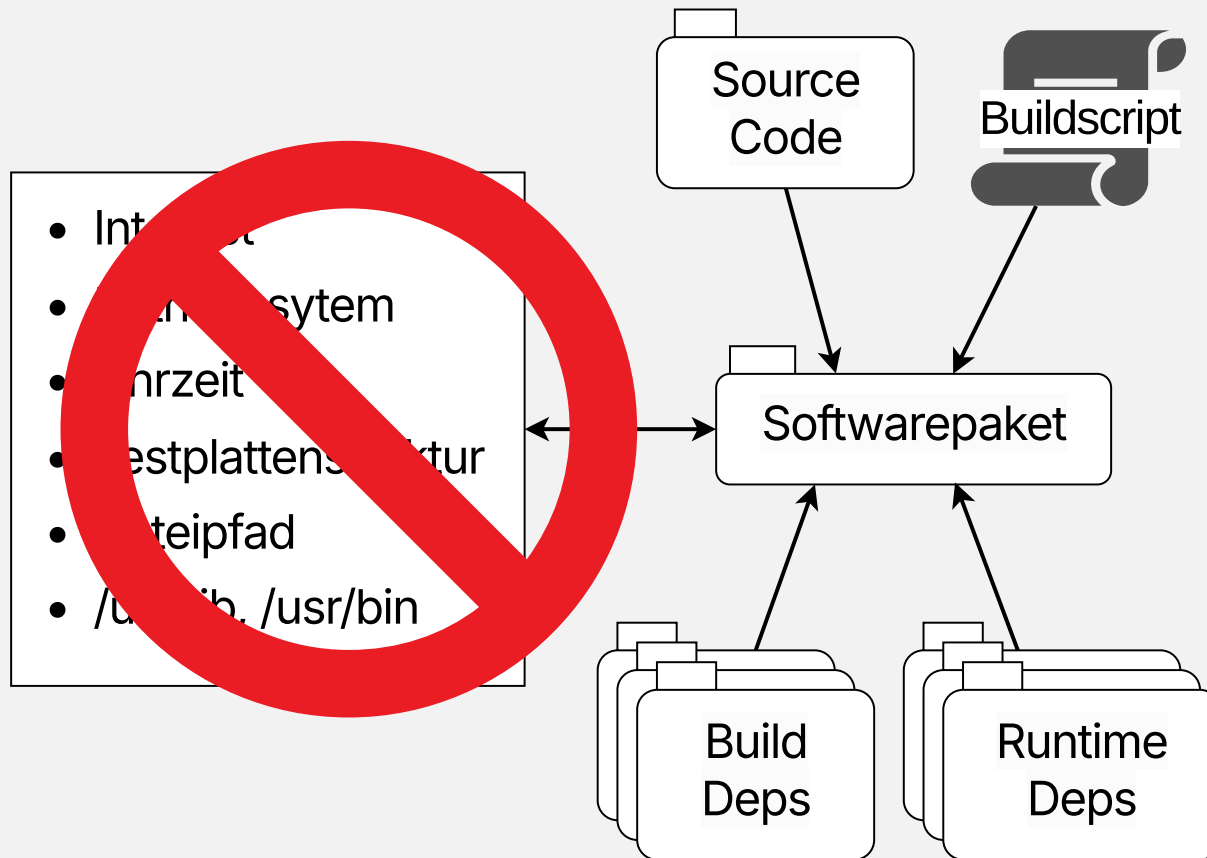
# Funktionales Paketmanagement à la Nix



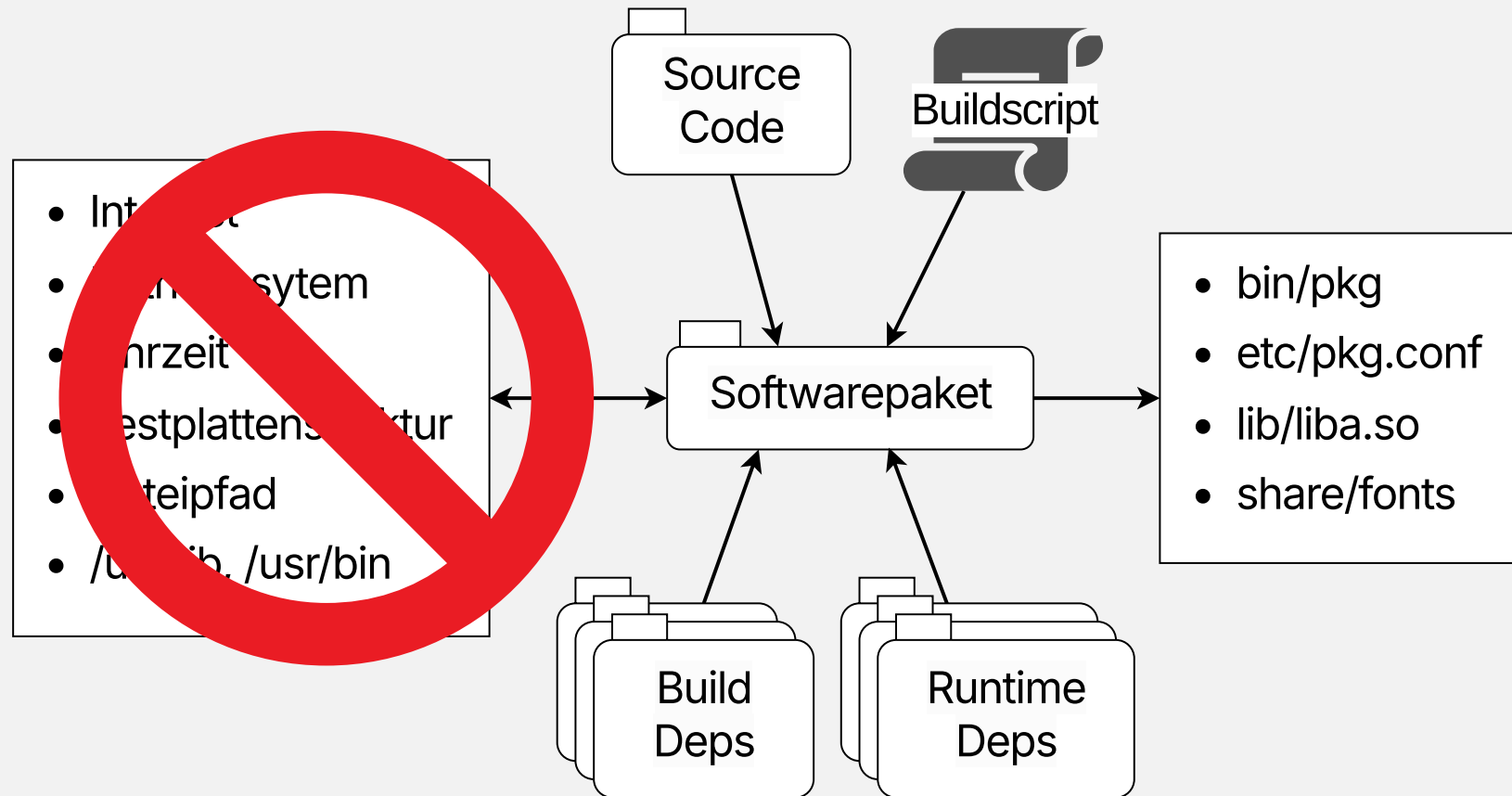
# Funktionales Paketmanagement à la Nix



# Funktionales Paketmanagement à la Nix

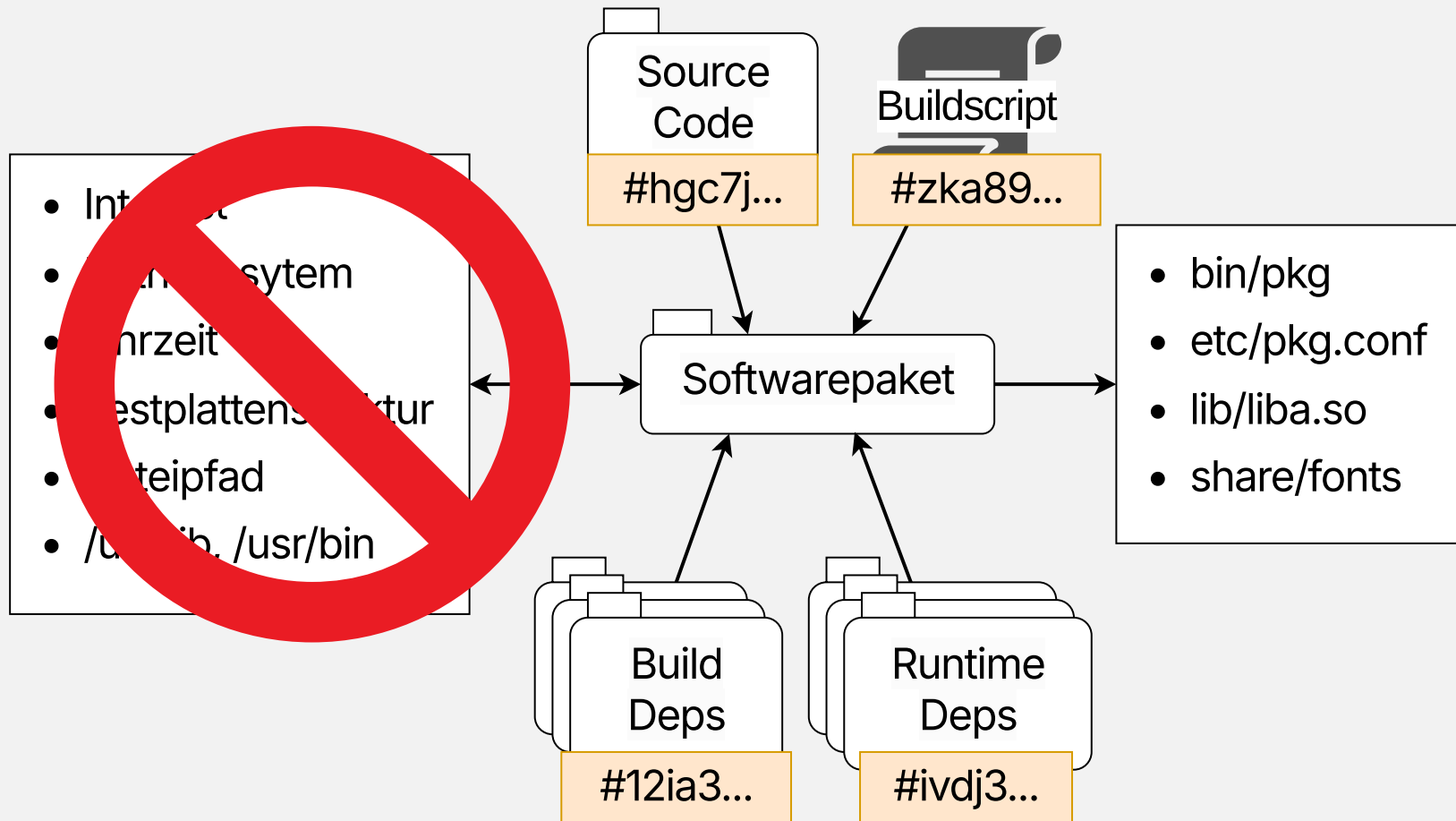


# Funktionales Paketmanagement à la Nix

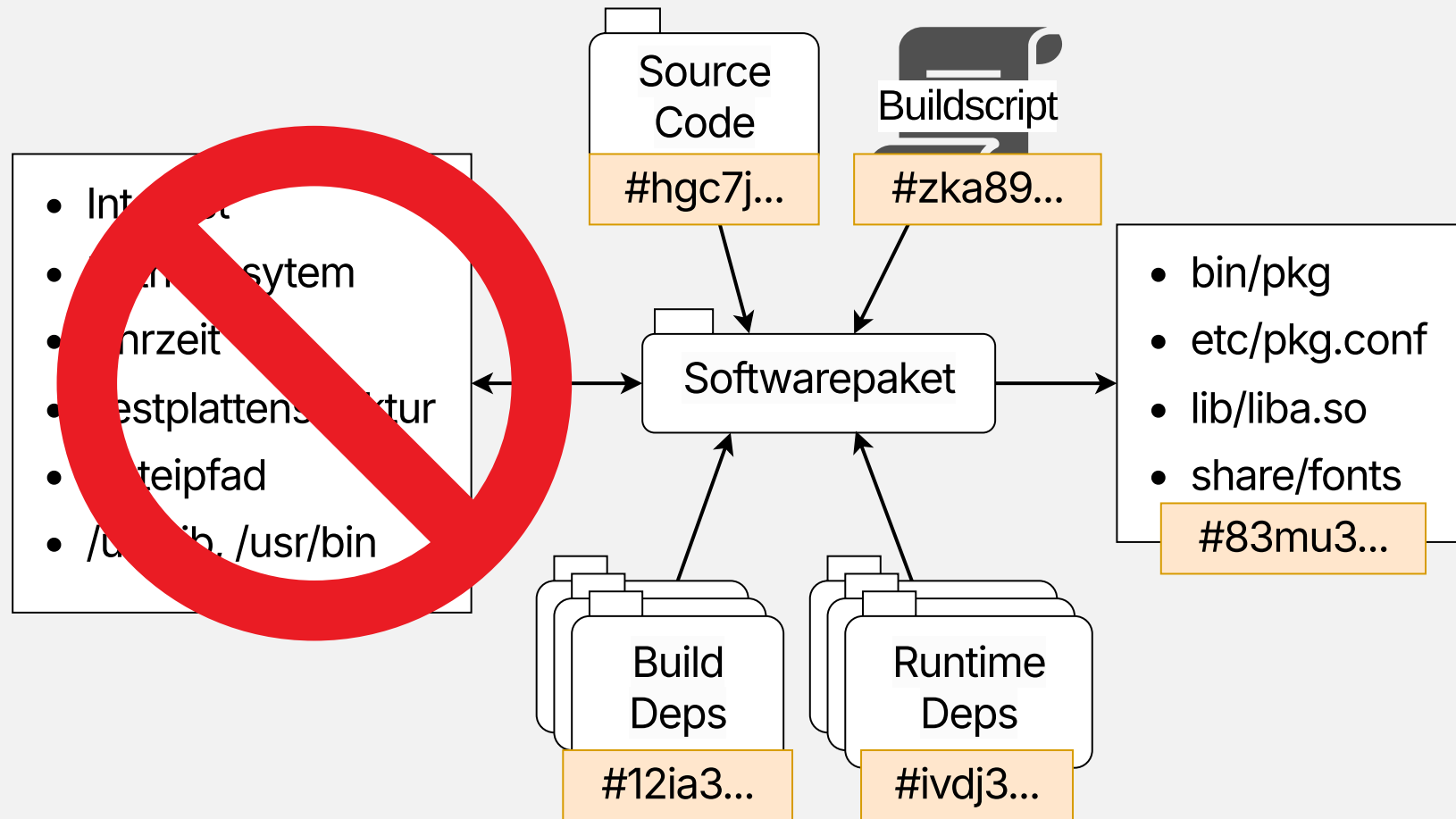




# Funktionales Paketmanagement à la Nix



# Funktionales Paketmanagement à la Nix



```
{ pkgs, ... }: {  
  environment.systemPackages = [ pkgs.firefox ];  
  services.postgres.enable = true;  
}
```



```
{  
  fileSystems."/\" = { device = "/dev/sda1"; fsType = "ext4"; };  
  boot.loader.grub.enable = true;  
  virtualisation.vmware.guest.enable = true;  
}
```



- Bare Metal
- VMs
- WSL, LXD
- Docker (NixOS & Nix)
- Linux & MacOS (Nix, Home Manager)

# Codchi – Code Machines

[codchi.dev](https://codchi.dev) [github.com/aformatik/codchi](https://github.com/aformatik/codchi)

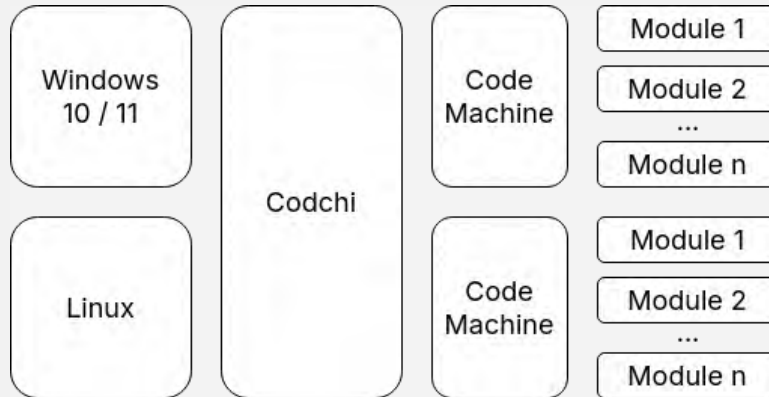
```
{ pkgs, ... }: {  
  environment.systemPackages = [ pkgs.firefox ];  
  services.postgres.enable = true;  
}
```



## Codchi



- Windows (WSL)
- Linux (LXD)
- MacOS (TBD)



- Isoliert (Systemcontainer)
- Host-Integration: Native UI, Network, File Sharing, Clipboard, Sound, GPU, ...
- Alternative Runtime zur VM

# Codchi in Action

 [codchi.dev](https://codchi.dev)

 [github.com/htngr/jfs-25-codchi](https://github.com/htngr/jfs-25-codchi)

 [NixOS Search \(Packages/Options\)](#)

```
codchi clone jfs25 https://github.com/htngr/jfs-25-codchi
```

```
codchi status
```

```
codchi module list jfs25
```

```
codchi exec jfs25
```

```
codchi rebuild jfs25
```

```
codchi delete jfs25
```



# **Codchi - Fazit**

- ✓ Automatisierte und Reproduzierbare Installation, Updates und Rollbacks



# **Codchi - Fazit**

- ✓ Automatisierte und Reproduzierbare Installation, Updates und Rollbacks
- ✓ Entwicklungsumgebung entwickelt sich mit Projekt

# **Codchi - Fazit**

- ✓ Automatisierte und Reproduzierbare Installation, Updates und Rollbacks
- ✓ Entwicklungsumgebung entwickelt sich mit Projekt
- ✓ Deklarative Systemkonfiguration statt manuelle Konfiguration / Scripting

# **Codchi - Fazit**

- ✓ Automatisierte und Reproduzierbare Installation, Updates und Rollbacks
- ✓ Entwicklungsumgebung entwickelt sich mit Projekt
- ✓ Deklarative Systemkonfiguration statt manuelle Konfiguration / Scripting
- ✓ Top Developer-Experience: `codchi clone`, `codchi rebuild`

# **Codchi - Fazit**

- ✓ Automatisierte und Reproduzierbare Installation, Updates und Rollbacks
- ✓ Entwicklungsumgebung entwickelt sich mit Projekt
- ✓ Deklarative Systemkonfiguration statt manuelle Konfiguration / Scripting
- ✓ Top Developer-Experience: `codchi clone`, `codchi rebuild`
- ✓ Individualisierbare Entwicklungsumgebungen

# **Codchi - Fazit**

- ✓ Automatisierte und Reproduzierbare Installation, Updates und Rollbacks
- ✓ Entwicklungsumgebung entwickelt sich mit Projekt
- ✓ Deklarative Systemkonfiguration statt manuelle Konfiguration / Scripting
- ✓ Top Developer-Experience: `codchi clone`, `codchi rebuild`
- ✓ Individualisierbare Entwicklungsumgebungen
- ✓ Kein Vendor-Lockin

# Codchi - Fazit

- ✓ Automatisierte und Reproduzierbare Installation, Updates und Rollbacks
  - ✓ Entwicklungsumgebung entwickelt sich mit Projekt
  - ✓ Deklarative Systemkonfiguration statt manuelle Konfiguration / Scripting
  - ✓ Top Developer-Experience: `codchi clone`, `codchi rebuild`
  - ✓ Individualisierbare Entwicklungsumgebungen
  - ✓ Kein Vendor-Lockin
- Noch Beta-Phase



# Codchi - Fazit

- ✓ Automatisierte und Reproduzierbare Installation, Updates und Rollbacks
  - ✓ Entwicklungsumgebung entwickelt sich mit Projekt
  - ✓ Deklarative Systemkonfiguration statt manuelle Konfiguration / Scripting
  - ✓ Top Developer-Experience: `codchi clone`, `codchi rebuild`
  - ✓ Individualisierbare Entwicklungsumgebungen
  - ✓ Kein Vendor-Lockin
- 
- Noch Beta-Phase
  - Anderes Konzept von Systemmanagement



# Codchi - Fazit

- ✓ Automatisierte und Reproduzierbare Installation, Updates und Rollbacks
- ✓ Entwicklungsumgebung entwickelt sich mit Projekt
- ✓ Deklarative Systemkonfiguration statt manuelle Konfiguration / Scripting
- ✓ Top Developer-Experience: `codchi clone`, `codchi rebuild`
- ✓ Individualisierbare Entwicklungsumgebungen
- ✓ Kein Vendor-Lockin
  
- Noch Beta-Phase
- Anderes Konzept von Systemmanagement
- Steile Lernkurve





Zu Codchi-Workshop anmelden: [codchi@aformatik.de](mailto:codchi@aformatik.de)





# `git clone` deine Entwicklungsumgebung

## Deklarative und reproduzierbare lokale Entwicklungsumgebungen mit Codchi

*"Codchi is a tool that manages your project's development environment in a reproducible and easy-to-use way. Setting up a development environment should be as easy as a git clone!"*

---

Johannes Hüttinger

 [codchi.dev](https://codchi.dev)

 [aformatik – Training & Consulting](https://aformatik.de)  
[www.aformatik.de](https://www.aformatik.de)