

## **Web Engineering**

# **Hosting & Deployment**

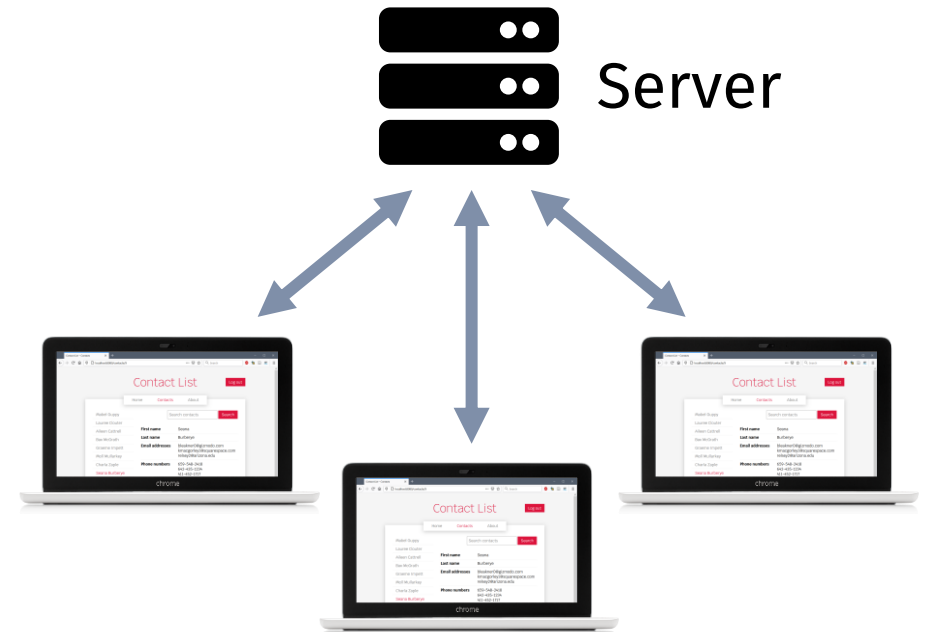
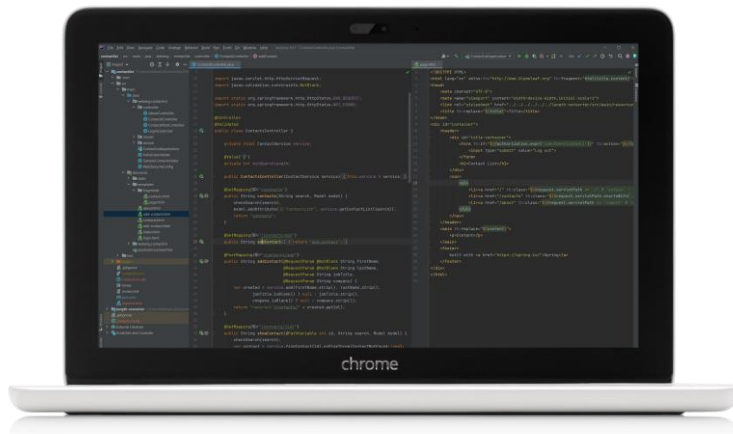
Michael Faes

*(viele Dinge schamlos von Michael Heinrichs & Prof. Dierk König abgekupfert)*

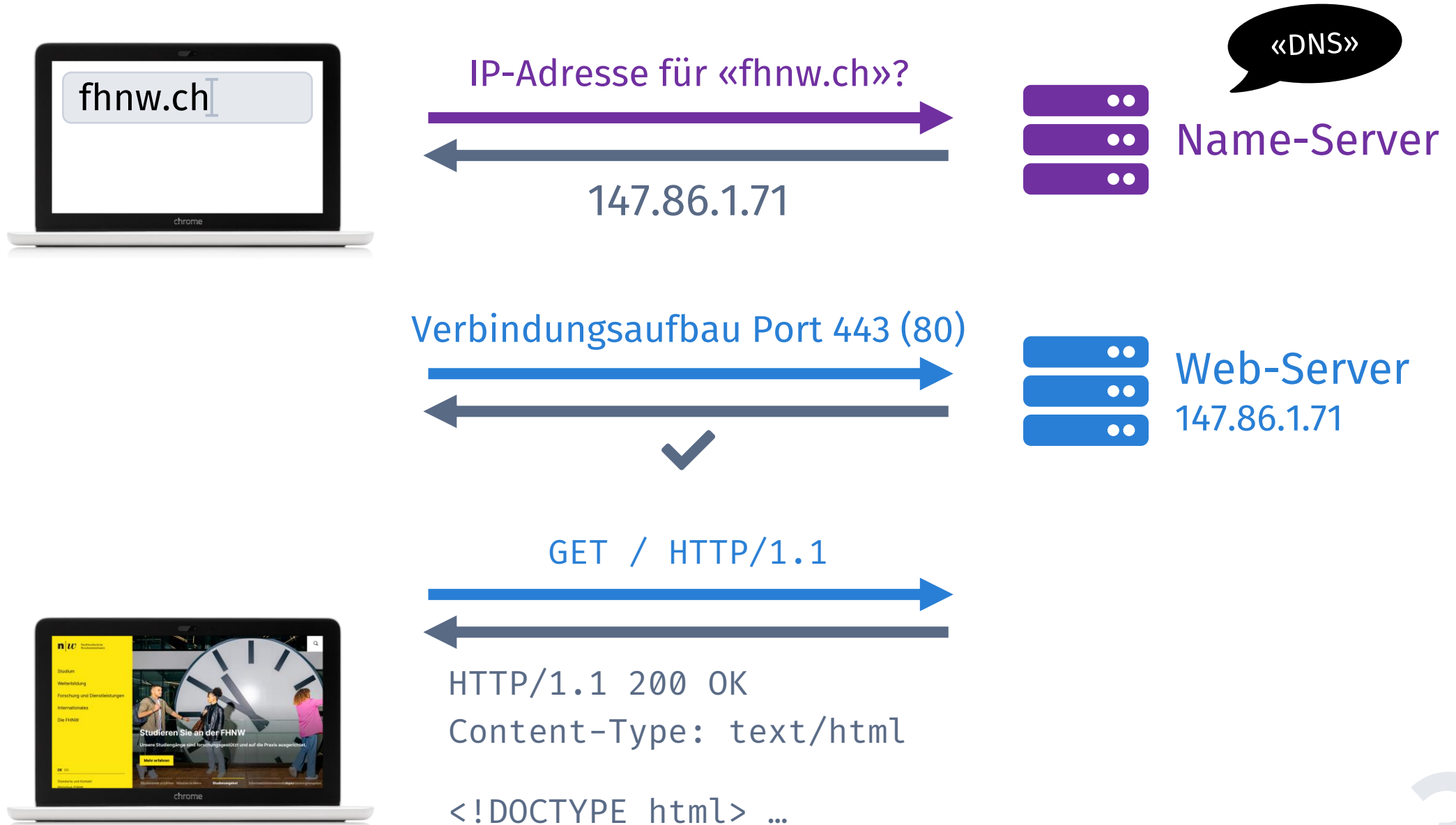
localhost:8080



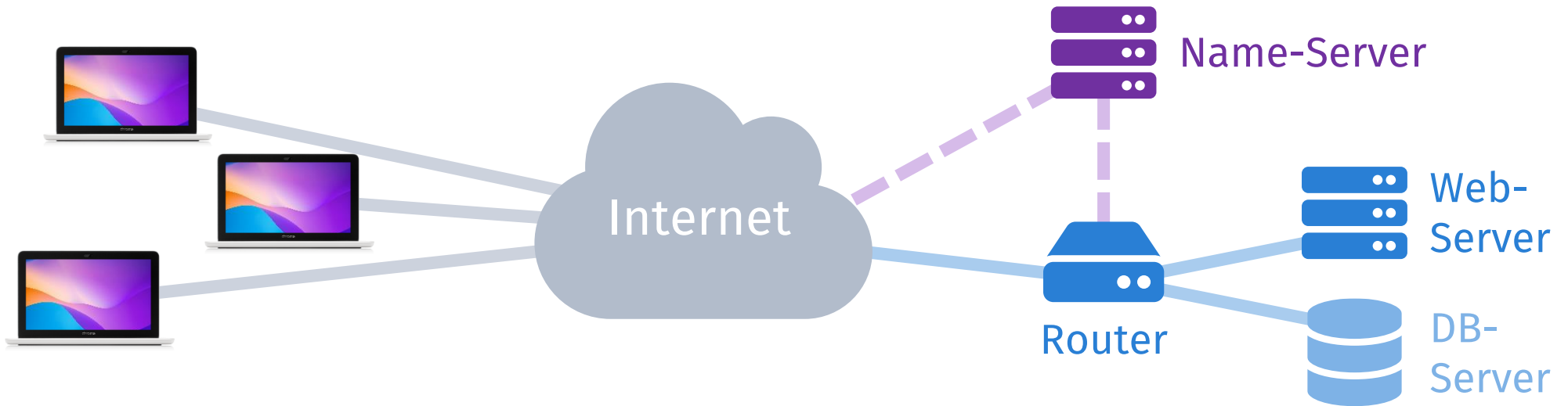
app.example.com



# Zugriff auf Web-App



# Infrastruktur für Web-App

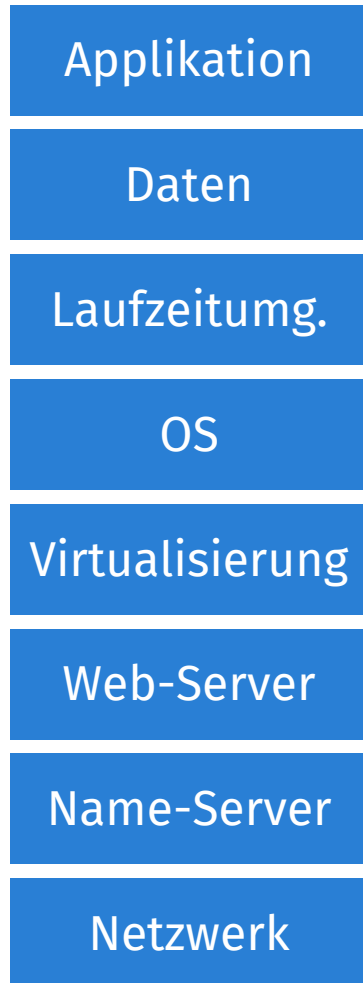


- Verbindung zum Internet: Öffentliche IP
- Name Server mit einem Eintrag (*A Record*): «*app.example.com* → IP»
- Eintrag in Name-Server für *example.com* (*NS Record*):  
«Name-Server für *app.example.com* → IP»
- Computer, auf dem die Web-App läuft (Web-Server)
- (evtl. separater Computer für Datenbank)

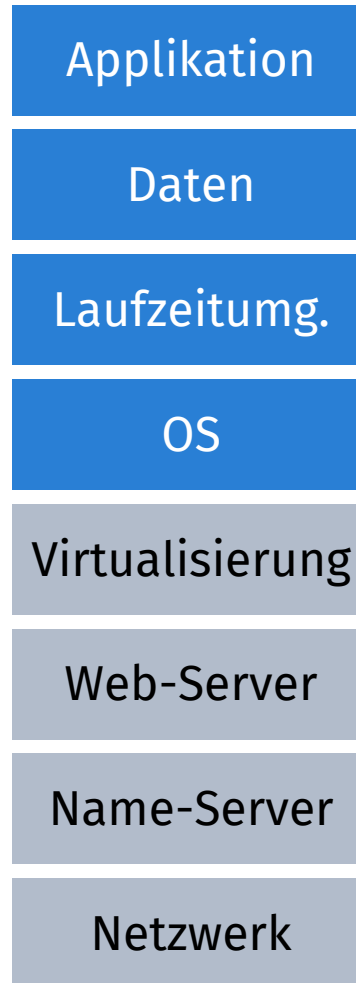
# Professionelles Hosting

Etwas  
dazwischen:  
Container

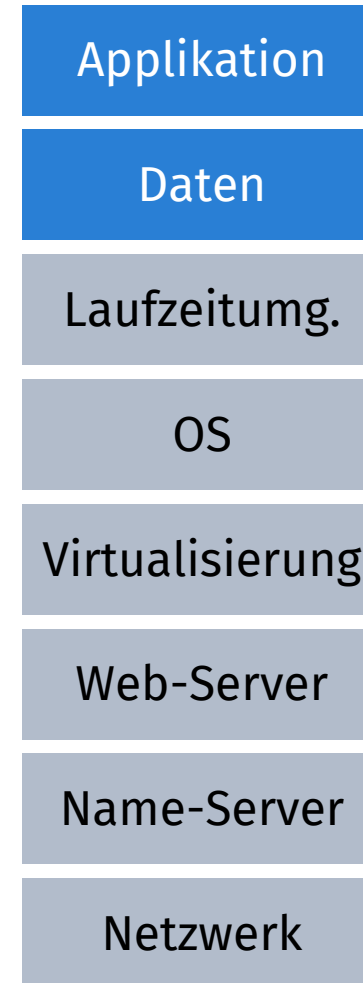
On-premises



IaaS



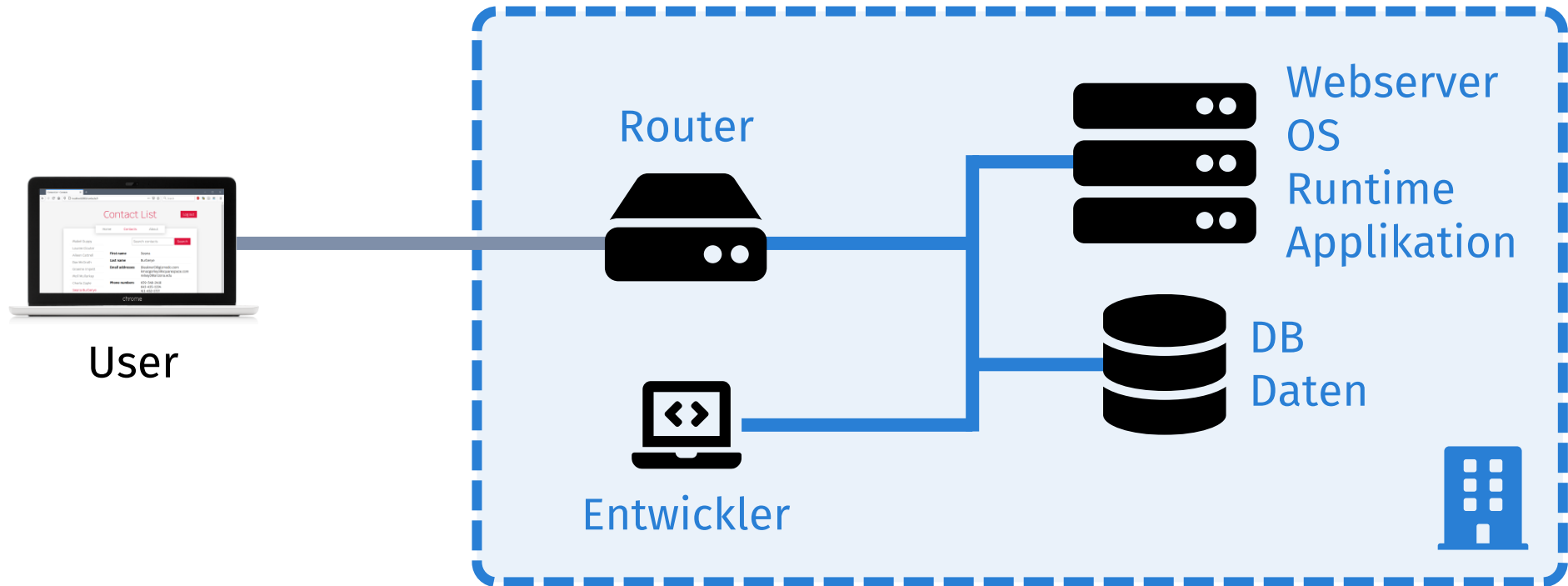
PaaS



Selber  
verwaltet

Provider  
verwaltet

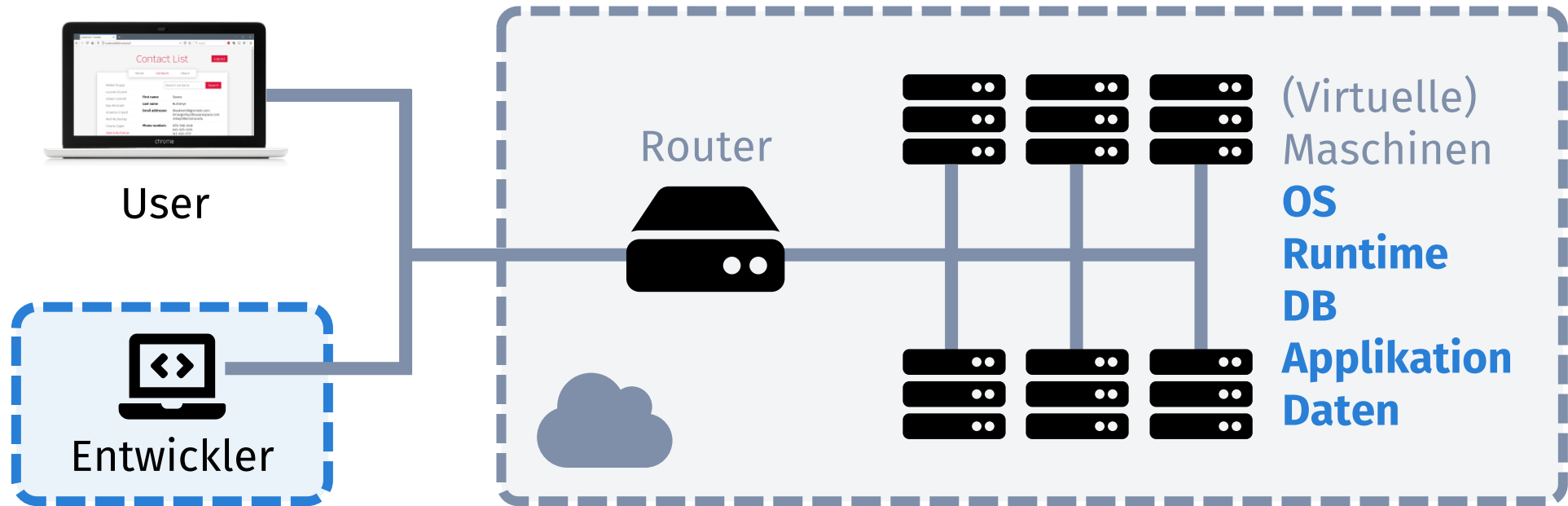
# On-premises Hosting



**+** Komplette Kontrolle  
Niedrige Betriebskosten

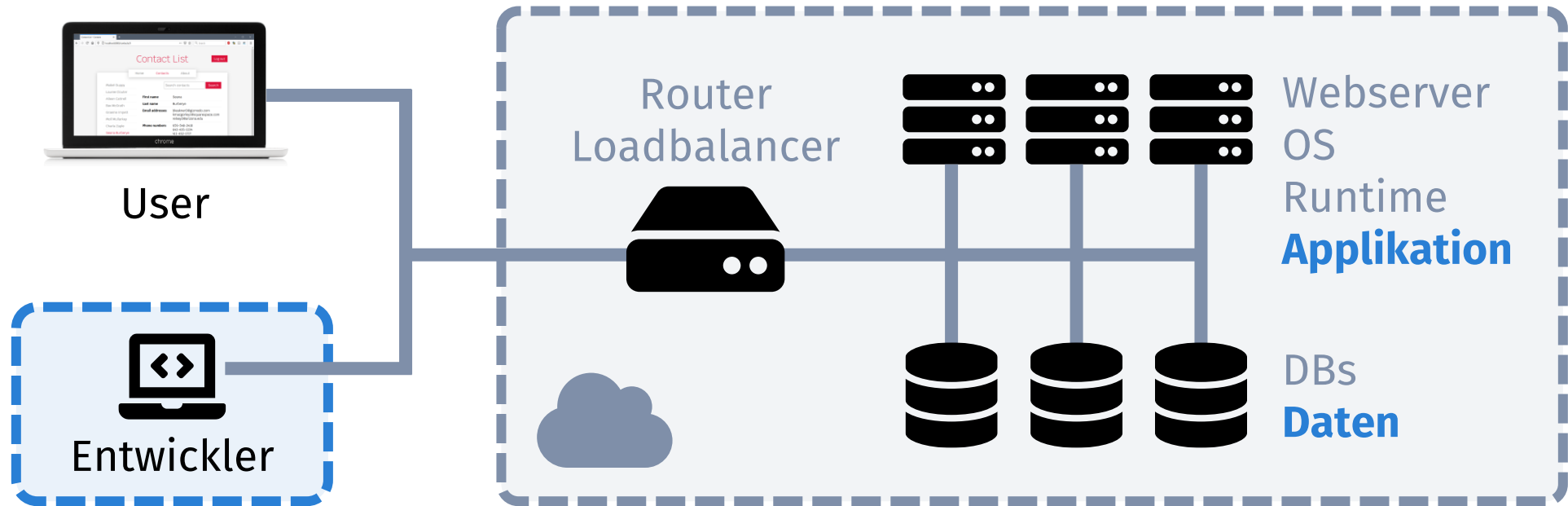
**—** Hohe Investitionskosten  
Lange Time-to-Market  
Hohe Wartungskosten

# IaaS Hosting



- +** Keine Investitionen
- Schnellere Time-to-Market
- Wenig Nutzung → niedrige Kosten
- Weniger Kontrolle
- Viel Nutzung → hohe Kosten

# PaaS Hosting



- + Keine Investitionen
- + Sehr schnelle Time-to-Market
- + Wenig Nutzung → niedrige Kosten
- Sehr wenig Kontrolle
- Viel Nutzung → hohe Kosten



# Packaging von Java-Web-Apps



Traditionell  
**WAR-Datei**

**Enthält:**

Applikations-Code  
Libraries

**Braucht:**

OS, Java-Runtime,  
Webserver

.class-  
Dateien

«Servlet-  
Container»



Typisch Spring Boot  
**Ausführbare JAR-Datei**

**Enthält:**

Applikations-Code  
Libraries  
Webserver

**Braucht:**

OS, Java-Runtime



Modern  
**Container-Image**

**Enthält:**

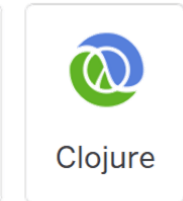
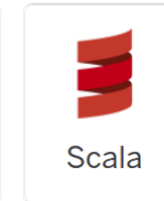
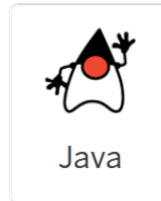
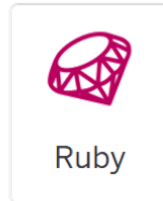
Applikations-Code  
Libraries  
Webserver  
Java-Runtime  
OS-Userspace

**Braucht:** OS-Kernel

# Beispiel für PaaS-Provider: Heroku



OFFICIALLY SUPPORTED LANGUAGES



Unterstützte Java-Packages: WAR, JAR oder Docker-Container

Deployment/Verwaltung via Web-Interface/Git/Kommandozeile

***Student Offer* bietet 13\$ pro Monat für Hosting & DB für 24 Monate**


- Reicht für durchgehenden Betrieb von Web-Server mit 512 MB RAM, 1 CPU (nicht dediziert), plus Postgres- und Redis-DB
- Oder für andere Produkte

# Gratis-Stunden auf Heroku: Vorbereitung

1. (Falls nötig, Account auf GitHub erstellen)
2. Für *GitHub Education* registrieren (braucht hinterlegte @students-Adresse): [github.com/edu](https://github.com/edu)
3. Heroku-Account erstellen: [signup.heroku.com](https://signup.heroku.com)
4. Den *Student Offer* auf Heroku beantragen: [www.heroku.com/github-students](https://www.heroku.com/github-students)

Könnte einige Stunden dauern, bis Student Offer freigeschaltet wird...

# Spring-Boot-App auf Heroku deployen

1. Installiere Heroku-CLI: [devcenter.heroku.com/articles/heroku-cli](https://devcenter.heroku.com/articles/heroku-cli)
2. Über Web-Interface neue App erstellen (<app-name> merken)
3. App als JAR-Datei verpacken: `mvn package (-DskipTests)`
4. In Terminal in Projekt (Bash/PowerShell):  
`heroku login`  
`heroku plugins:install java`  
`heroku deploy:jar target/<jar-name>.jar`  
`--app <app-name> --jdk 21`  
`--options '--server.port=$PORT'`
5. Im Heroku Web-Interface Deployment und Logs prüfen  
(«More» → «View Logs»)
6. App aufrufen: <https://<app-name>.herokuapp.com>

**Fragen?**

