CLOUD NATIVE COMPUTING

CLOUD-NATIVE

Unit:

Infrastructure as Code

(3) Automatisierte Provisionierung

creative commons

CLOUD NATIVE COMPUTING

Urheberrechtshinweise

Diese Folien werden zum Zwecke einer praktikablen und pragmatischen Nutzbarkeit im Rahmen der **CCo 1.0 Lizenz** bereitgestellt.

Sie dürfen die Inhalte also kopieren, verändern, verbreiten, mit eigenen Inhalten mixen, auch zu kommerziellen Zwecken, und ohne um weitere Erlaubnis bitten zu müssen.

Eine Nennung des Autors ist nicht erforderlich (aber natürlich gern gesehen, wenn problemlos möglich).

Diese Folien sind insb. für die Lehre an Hochschulen konzipiert und machen daher vom **§51 UrhG (Zitate)** Gebrauch.

Die CCo Lizenz überträgt sich nicht auf zitierte Quellen. Hier sind bei der Nutzung natürlich die Bedingungen der entsprechenden Quellen zu beachten.

Die Quellenangaben finden sich auf den entsprechenden Folien.

KAPITEL 7

Infrastructure as Code

7 Infrastructure as Code

7.1 Virtualisierung

- Virtualisierung von Hardware-Infrastruktur
- Virtualisierung von Software-Infrastruktur

7.2 Provisionierung

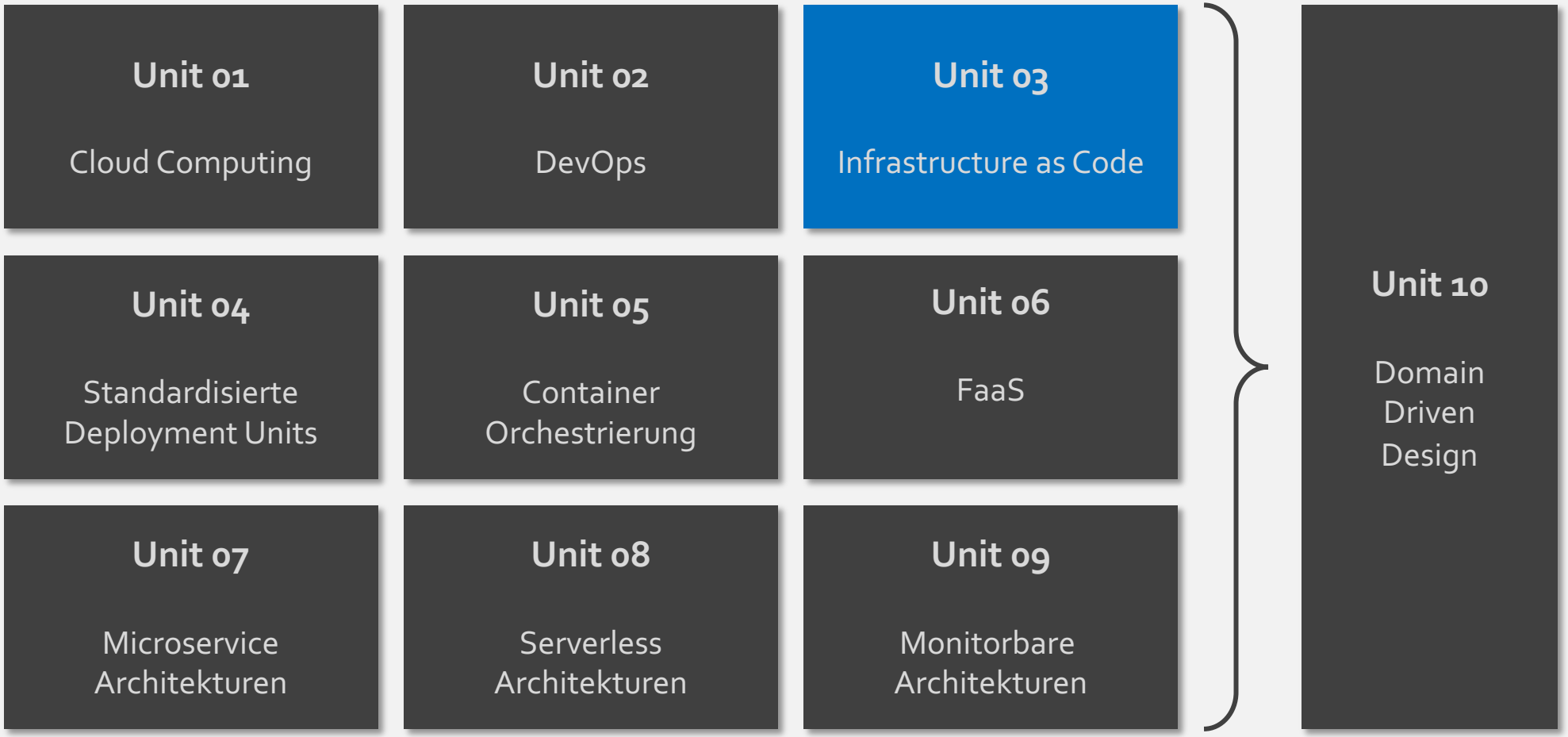
- Immutable Infrastructure
- Infrastructure as Code
- Provisionierung von lokalen Umgebungen
- Provisionierung von Multi-Host Umgebungen

7.3 Zusammenfassung

CLOUD NATIVE COMPUTING

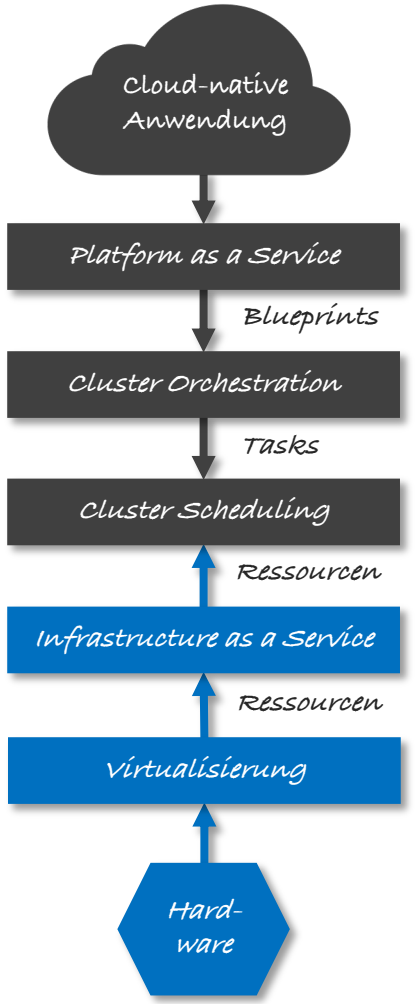
INHALTSVERZEICHNIS

Überblick über Units und Themen dieses Moduls



INHALTE

- Virtualisierung
- Infrastructure as a Service
- Provisionierung in IaaS-basierte Infrastrukturen
  - Historische Entwicklung und
  - Ebenenmodell
  - Immutable Infrastructures
- Infrastructure as Code



PROVISIONIERUNG

Automatisierte Bereitstellung von IT-Ressourcen im Verlaufe der letzten Jahrzehnte

Ohne Virtualisierung (vor 2000):
- Manuelles Installieren von Betriebssystem auf dedizierter Hardware
- Manuelle Installation von Infrastruktur-Software
- Manuelle / teilautomatisierte / automatisierte Installation der Anwendungssoftware per Installer, Skript, proprietäre Lösungen

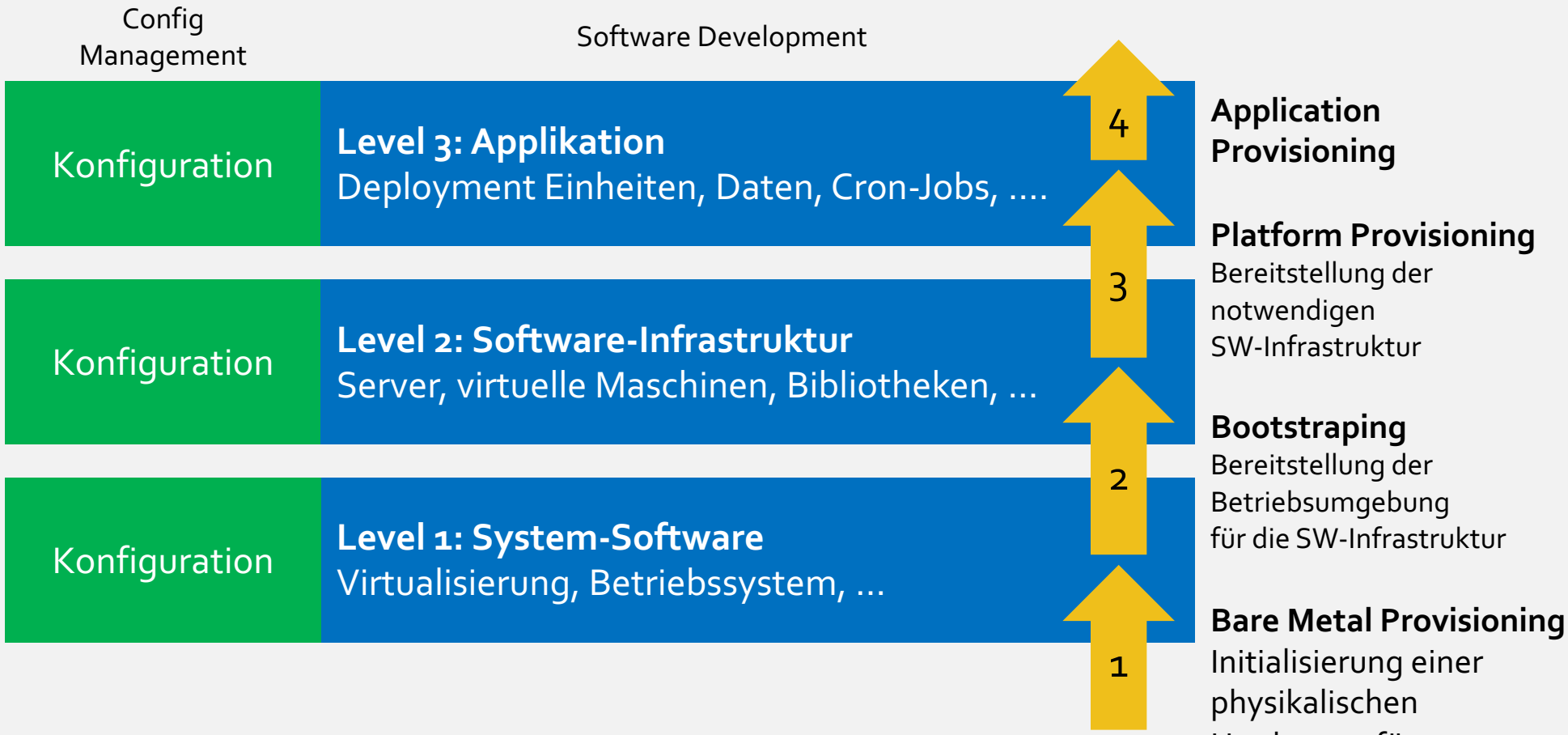
Virtualisierung einzelner Maschinen (2000 – heute)
- Manuelles Installieren von virtuellen Maschinen
- Manuelle Installation von Infrastruktur-Software
- Manuelle / teilautomatisierte / automatisierte Installation der Anwendungssoftware per Installer, Skript, proprietäre Lösungen

Virtualisierung in der Cloud (seit 2010)
- Automatisches Bereitstellen von vorgefertigten virtuellen Maschinen und Containern
- Manuelle Installation der Infrastruktur-Software nur 1x Clone-Master-Image
- Bereitstellung einer definierten Umgebung auf Knopfdruck

Infrastructure as Code (2010 – heute)
- Programmieren der Provisionierung und weiterer Betriebsprozeduren

PROVISIONIERUNG

Ebenenmodell



Application Provisioning

Platform Provisioning
Bereitstellung der notwendigen SW-Infrastruktur

Bootstraping
Bereitstellung der Betriebsumgebung für die SW-Infrastruktur

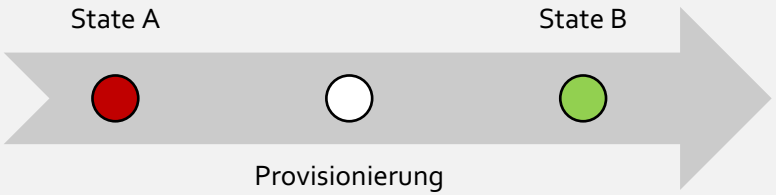
Bare Metal Provisioning
Initialisierung einer physikalischen Hardeware für den Betrieb

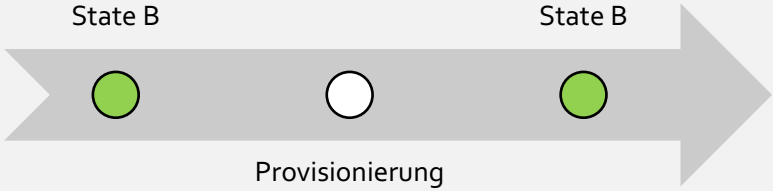
PROVISIONIERUNG

Konzeptionelle Überlegungen

CLOUD NATIVE COMPUTING

Systemzustand := Gesamtheit der Software, Daten und Konfigurationen auf einem System
Provisionierung := Überführung eines System-Ist-Zustands in einen Ziel-Zustand





Provisionierungsmechanismus:

1. Ausgangszustand feststellen
2. Vorbedingungen prüfen
3. Zustandsändernde Aktionen ermitteln
4. Zustandsändernde Aktionen durchführen
5. Nachbedingungen prüfen (ggf. Zustand zurücksetzen)

Idempotenz: Die Fähigkeit eine Aktion durchzuführen und sie dasselbe Ergebnis erzeugt, egal ob sie einmal oder mehrfach ausgeführt wird (Wiederholungen ändern den Zustand nicht).

Konsistenz: Nach Ausführung der Aktionen herrscht ein ein konsistenter Systemzustand. Egal ob einzelne, mehrere oder alle Aktionen gescheitert sind.

IMMUTABLE ARCHITECTURES

Pets vs. Cattle

CLOUD NATIVE COMPUTING

An immutable infrastructure is another infrastructure paradigm in which servers are **never modified** after they're deployed. If something needs to be updated, fixed, or modified in any way, **new servers built from a common image** with the **appropriate changes are provisioned** to replace the old ones. After they're validated, they're put into use and the old ones are decommissioned.

The benefits of an immutable infrastructure include **more consistency and reliability** in your infrastructure and a **simpler, more predictable deployment process**. It mitigates or entirely **prevents** issues that are common in mutable infrastructures, like **configuration drift**.

Server (virtuelle Maschinen) sind Wegwerf-Ware!

https://www.digitalocean.com/community/tutorials/what-is-immutable-infrastructure

IMMUTABLE ARCHITECTURES

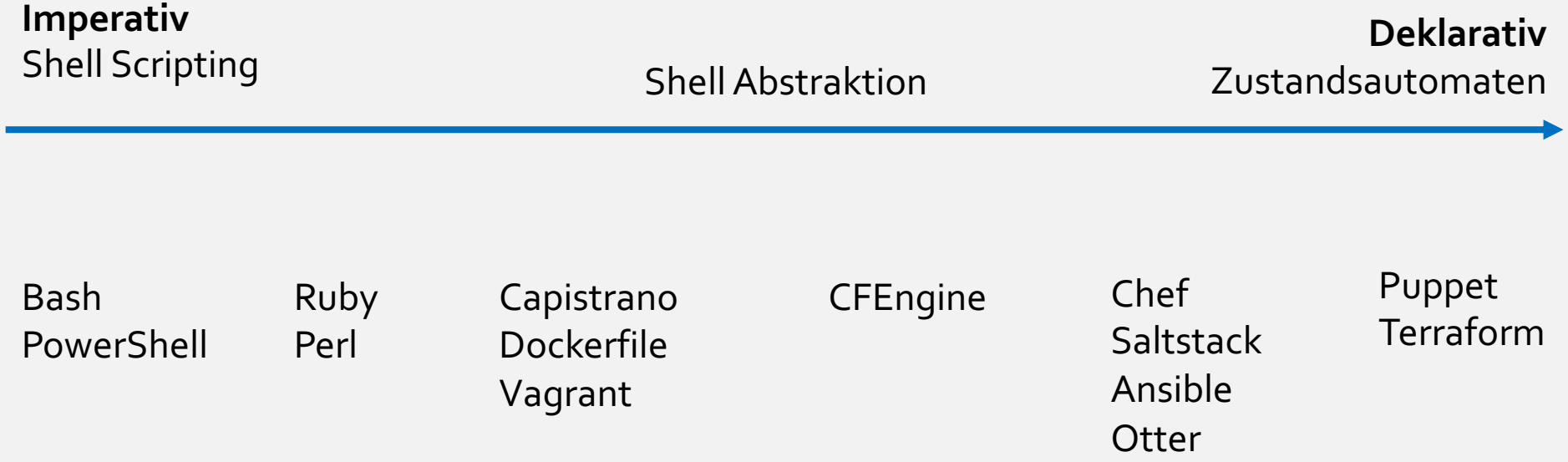
Reduzieren zu berücksichtigende Zustände im Provisionierungsprozess

1. Ausgangszustand feststellen
2. Vorbedingungen prüfen
3. Zustandsändernde Aktionen ermitteln
4. Zustandsändernde Aktionen durchführen
5. Nachbedingungen prüfen (ggf. Zustand zurücksetzen)

1. ~~Ausgangszustand feststellen~~
2. ~~Vorbedingungen prüfen~~
3. ~~Zustandsändernde Aktionen ermitteln~~
4. Zustandsändernde Aktionen durchführen
5. Nachbedingungen prüfen ~~(ggf. Zustand zurücksetzen)~~

WERKZEUGE

Übersicht gängiger Provisionierungswerkzeuge



AUSBLICK

- Virtualisierung
- Infrastructure as a Service
- Provisionierung in IaaS-basierte Infrastrukturen
- Infrastructure as Code
  - Definition, Ansätze und Methoden
  - Überblick gängiger Provisionierungswerkzeuge
  - Single VM-Provisioning mittels Vagrant (Deklaratives Push-basierte IaC)
  - Multi VM-Provisioning mittels Terraform (Deklaratives Push-basiertes IaC)

KONTAKT

Disclaimer

Nane Kratzke

+49 451 300-5549

nane.kratzke@th-luebeck.de

kratzke.mylab.th-luebeck.de