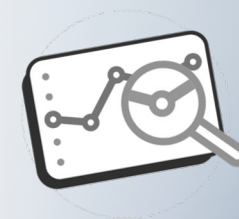




CLOUD-NATIVE

Unit:
Observability

(1) Was ist das?



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 13

Beobachtbare Architekturen



13.1 Konsolidierung von Telemetriedaten

13.2 Instrumentierung von Systemen

- Logging
- Monitoring
- Tracing

13.3 Automatisierte Instrumentierung

- Service Meshs
- Traffic-Management
- Resilienz
- Sicherheit
- Management und Analyse von Verkehrstopologien

13.4 Zusammenfassung

KAPITEL 13

Beobachtbare Architekturen (Instrumentierung von Systemen)



Nane Kratzke

Cloud-native Computing

Software Engineering von Diensten und Applikationen
für die Cloud

284 Seiten. E-Book inside

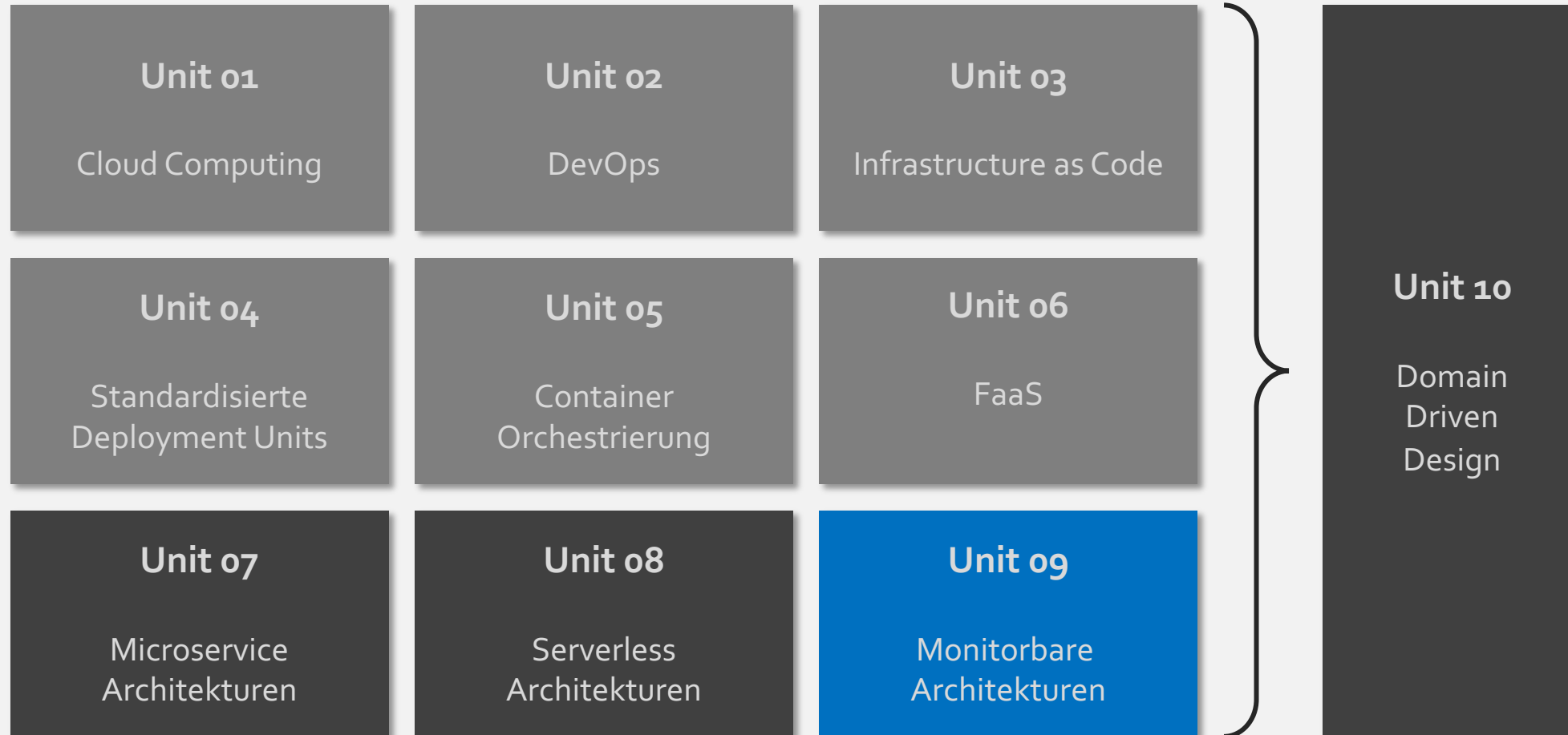
€ 59,99. ISBN 978-3-446-46228-1

Weitere Informationen unter: www.hanser-fachbuch.de

HANSER

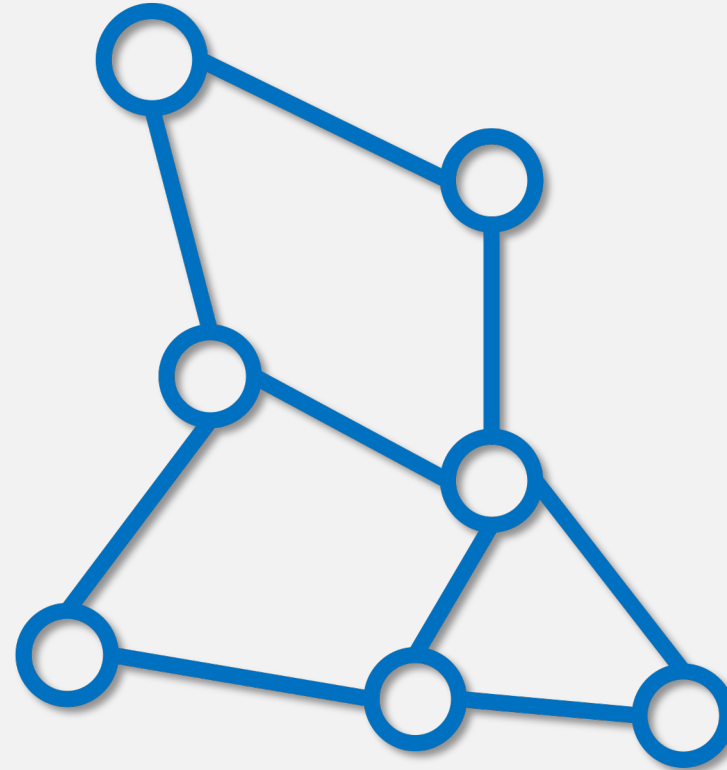
INHALTSVERZEICHNIS

Überblick über Units und Themen dieses Moduls



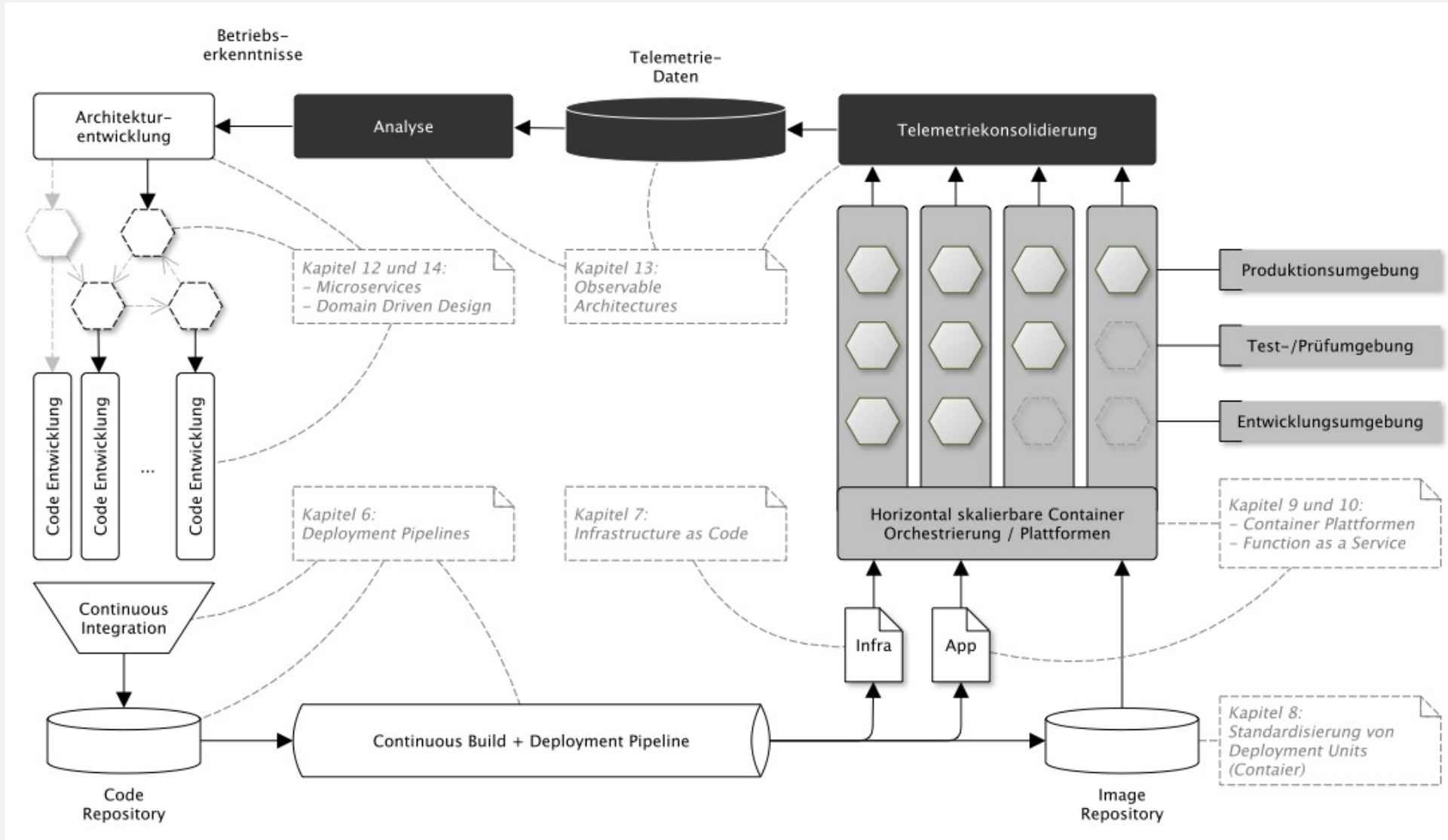
INHALTE

- **Beobachtbarkeit von Systemen**
- **Metriken**
- **Logs**
- **Tracing**



DEVOPS

konforme Architekturen und Infrastrukturen



ÜBERWACHUNG VON SYSTEMEN

System-Monitoring

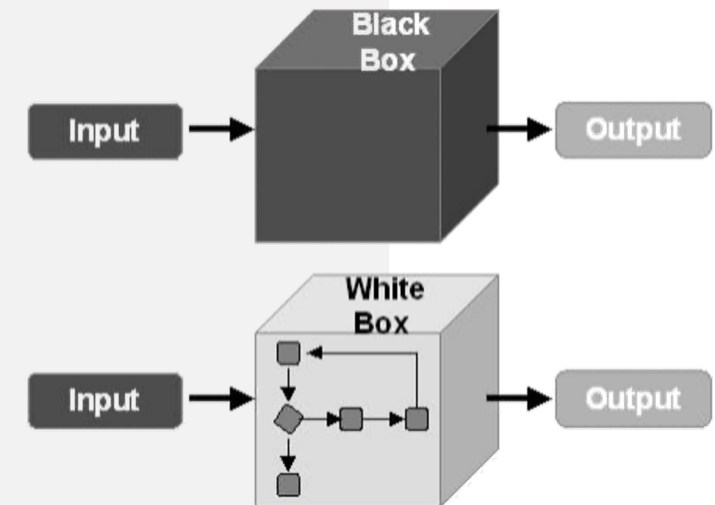
- Unter System-Monitoring versteht man die **kontinuierliche Überwachung** von IT-Systemen, Netzwerken, Anwendungen oder anderen technischen Systemen
- Dient dazu eventuelle **Probleme frühzeitig zu erkennen** und zu lösen
- System-Monitoring kann sowohl manuell als auch automatisiert erfolgen
- Typischerweise werden dabei verschiedene **Telemetriedaten** wie CPU-Auslastung, Speicher- und Festplattenverbrauch, Netzwerkverkehr, Anzahl von Verbindungen oder Anfragen, Systemereignisse, sowie Fehlerraten und mehr kontinuierlich erfasst und ausgewertet
- Diese Metriken können bspw. auf Schwankungen oder Anomalien überprüft werden, um mögliche Probleme zu erkennen
- System-Monitoring trägt dazu bei **Ausfallzeiten zu minimieren**, Probleme frühzeitig zu erkennen und die Systemleistung und -stabilität zu verbessern
- Es kann helfen die **Kapazitätsplanung** und -optimierung zu verbessern



ÜBERWACHUNG VON SYSTEMEN

White-Box- und Black-Box-Monitoring

- **Black-Box Monitoring:**
 - Überwachung eines Systems oder einer Anwendung **ohne Kenntnisse der internen Funktionsweise** oder des Quellcodes
 - System wird als "Black Box" betrachtet, bei dem nur die Ein- und Ausgänge des Systems betrachtet werden
 - Überwachungsprozess erfasst nur die Einflüsse von außen, wie z.B. Netzwerkverkehr, Systemleistung, Fehlermeldungen die **ohne Instrumentierung** erfolgen kann
 - Wird meist für einfach erfassbare Metriken (CPU, Memory, Netzwerk) auf Infrastrukturebene genutzt
- **White-Box Monitoring:**
 - Basiert auf Verständnis der **internen Funktionsweise** des Quellcodes eines Systems
 - Ermöglicht eine detailliertere und vor allem fachliche Überwachung und Analyse von Komponenten und Ereignissen innerhalb des Systems
 - Erfasst interne Metriken und Protokolldaten mittels einer **Instrumentierung**
 - Liefert tiefergehende und fachliche Erkenntnisse als Black-Box-Monitoring



ÜBERWACHUNG VON SERVICE-ARCHITEKTUREN

Logging, Monitoring, Tracing

Die Überwachung von Software erfolgt üblicherweise mittels der Erfassung von drei Arten von Telemetriedaten:

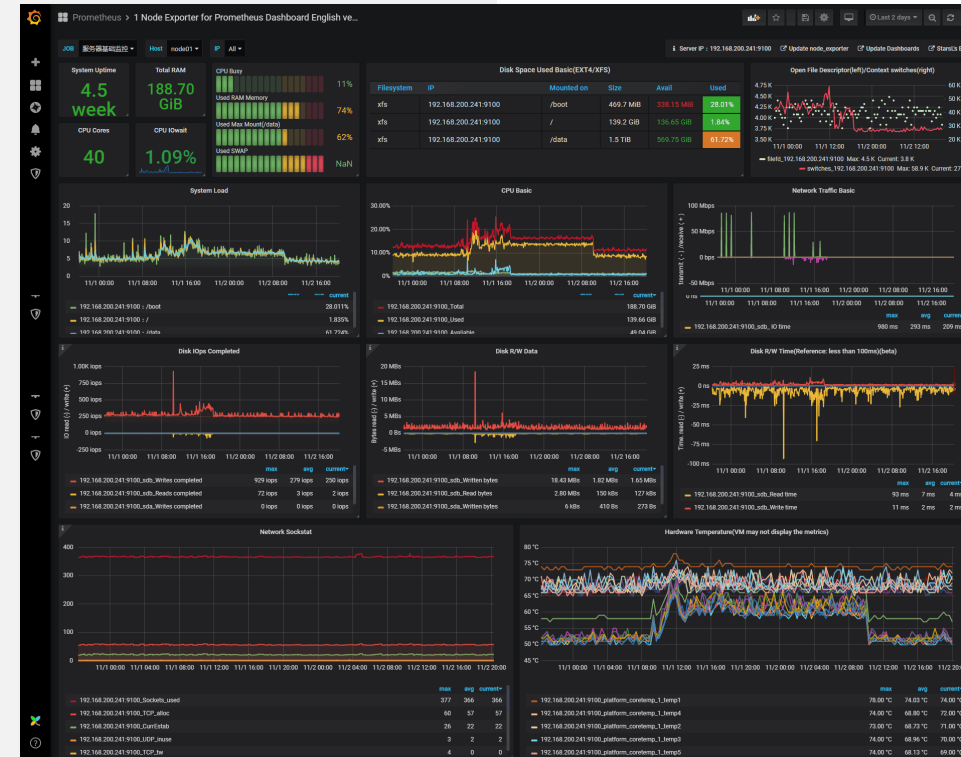
- Metriken im Rahmen eines **Monitorings** liefern **quantitative** Informationen zu Prozessen, die im System ausgeführt werden.
- Mittels **Logging** (Protokollierung) lässt sich **qualitativer** Einblick in anwendungsspezifische Ereignisse gewinnen, die von Prozessen verarbeitet werden.
- **Distributed Tracing** (verteilte Ablaufverfolgung) ermöglicht Einblick in den gesamten Lebenszyklus von Requests entlang von Systemkomponenten (Services), um so bspw. Fehler und Latenzen in der verteilten Verarbeitung zu erkennen.



WHITE-BOX MONITORING

Zur Erfassung quantitativer Zusammenhänge

- Im Cloud-native Umfeld werden Services oft mittels White-Box-Monitoring überwacht
- Hierzu müssen die überwachten Anwendungen instrumentiert werden, um deren interne Metriken mittels **Exportern** selbst bereitzustellen
- Es existieren unterschiedliche Metrik-Exporter Programmibliotheken (eine häufig genutzte Lösung ist hier Prometheus)
- Es lassen sich damit sowohl Service-spezifische Metriken als auch Black-Box Standardmetriken exportieren (bspw. CPU, Festplatte oder Netzwerk)
- Exporter werden periodisch von Monitoring-Lösungen abgefragt
- Oft aggregiert eine zentrale Monitoring-Lösung dann Daten verschiedener Quellen
- Einige Lösungen (wie bspw. Prometheus) verfügen hierzu über eine automatische Service-Discovery, um Ressourcen, die als Datenquellen verwendet werden sollen, automatisch ermitteln zu können
- Diese gesammelten Daten werden in einer Datenbank zur Zeitreihenanalyse gespeichert und können dann wie rechts gezeigt in Dashboards zur Visualisierung von Systemzuständen dargestellt werden



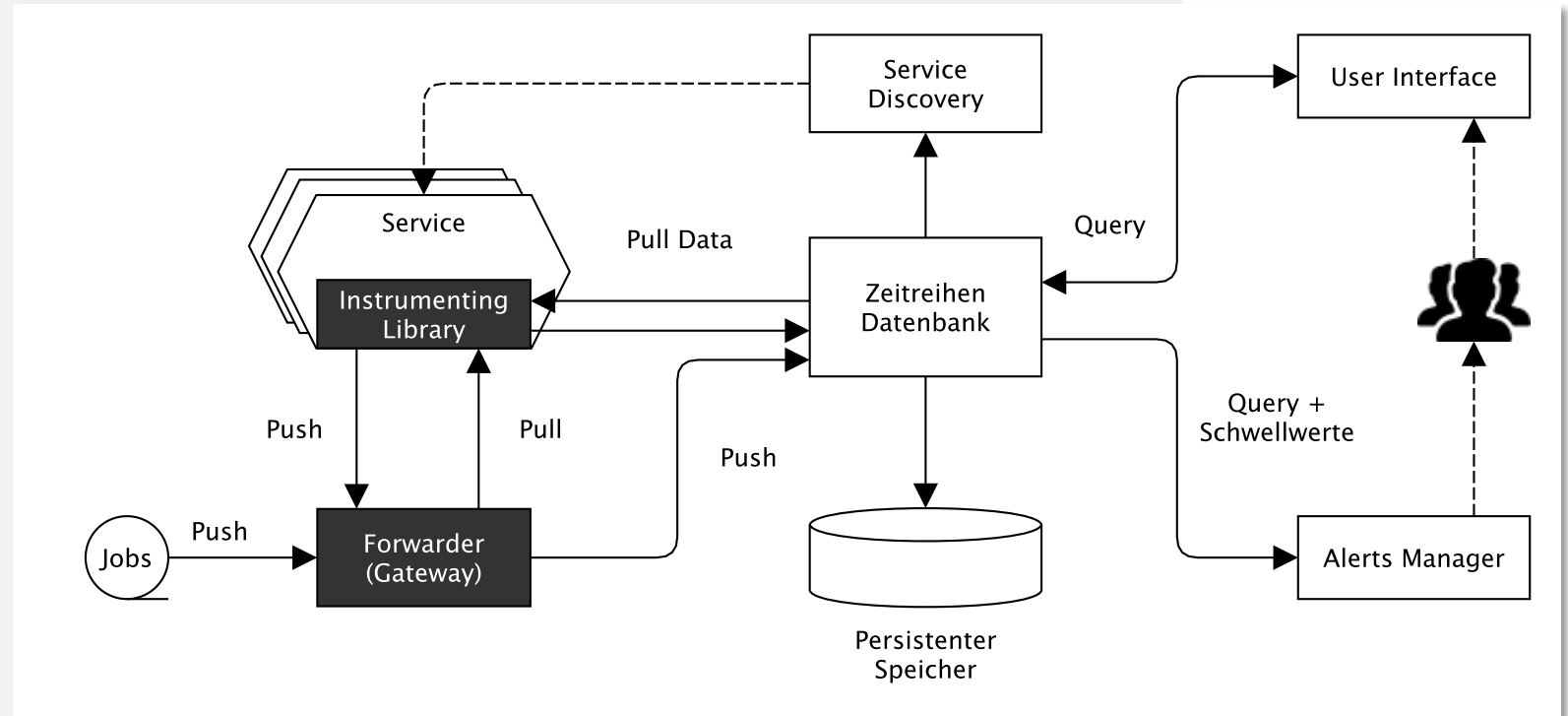
MONITORING

Telemetriedaten-Konsolidierung

Monitoring-Lösungen zeichnen Echtzeitmetriken mittels Zeitreihendatenbanken auf, die die Analyse quantitativer Zusammenhänge und u.a. Echtzeit-Warmmeldungen ermöglichen.

Monitoring-Lösungen bestehen meist aus mehreren Tools:

- Forwarder, die auf überwachten Hosts ausgeführt werden, um lokale Host- und Service-Metriken an einen Zentralspeicher zu exportieren
- Zeitreihendatenbank zur zentralisierten Speicherung von Metriken
- Alertmanager, der bei einer Schwellwertüberschreitung Benachrichtigungen verschicken kann
- Nutzeroberfläche (UI) zum Darstellen von Dashboards
- Querying-System, dass zum Erstellen von Dashboards und Warnungen verwendet wird

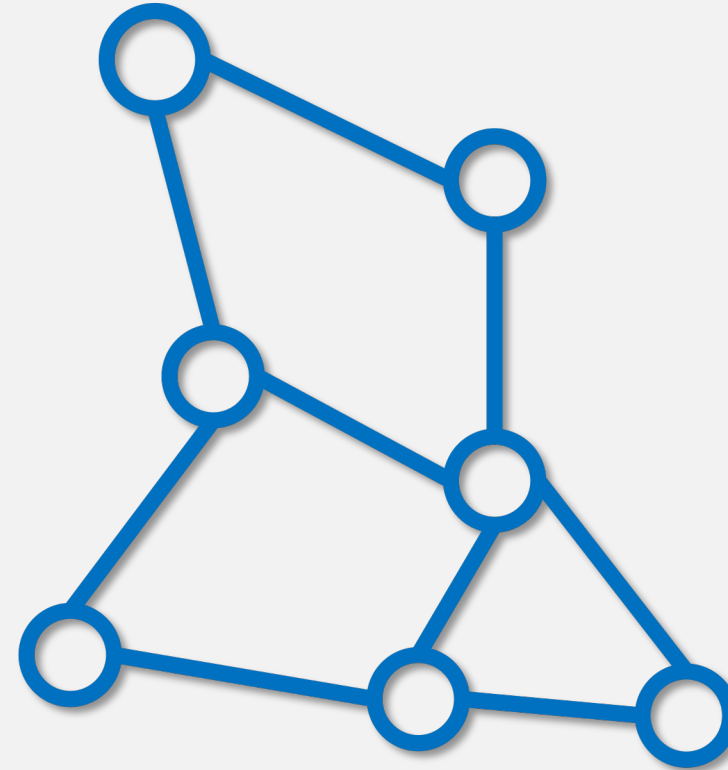


Architektur zur Konsolidierung von Telemetriedaten

Die Architekturen zur Konsolidierung von Logging und Distributed Tracing sehen recht ähnlich aus.

AUSBLICK

- Beobachtbarkeit von Systemen
- **Metriken**
- Logs
- Tracing



KONTAKT

Disclaimer

Nane Kratzke

📞 +49 451 300-5549

✉ nane.kratzke@th-luebeck.de

🔗 kratzke.mylab.th-luebeck.de

