Datenflussdiagramme und Vertraulichkeitsanalyse

Sebastian Hahner

Gastvorlesung ∈ Software Security Engineering

Lernziele der Gastvorlesung

- Vertraulichkeit definieren, damit verbundene Herausforderungen nennen
- Einsatzmöglichkeiten von Datenflussdiagrammen und Bestandteile beschreiben und von anderen Diagrammtypen abgrenzen können
- Datenflussdiagramme zeichnen und anhand dieser die Vertraulichkeit eines Software-Systems bewerten können
- Die Technik “Label Propagation“ wiedergeben und an beispielhaften Datenflussdiagrammen anwenden können

⇨ Die Lernziele eignen sich auch gut zur Prüfungsvorbereitung ☺

Definiere: Vertraulichkeit

- ISO 27000: „property that information is not made available or disclosed to unauthorized individuals, entities, or processes" **[1]**
- BSI: „Vertraulichkeit ist der Schutz vor unbefugter Preisgabe von Informationen. Vertrauliche Daten und Informationen dürfen ausschließlich Befugten in der zulässigen Weise zugänglich sein." **[2]**

Beispiele

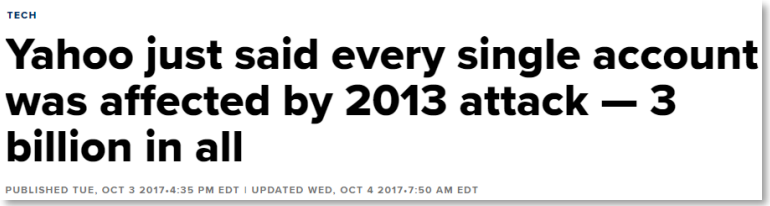
- Zahlungsdaten dürfen nur dem Zahlungsdienstleister offengelegt werden
- Personenbezogene Daten müssen verschlüsselt gespeichert werden
- Konkrete Prüfungsinhalte dürfen nicht kommuniziert werden ☺

[1] ISO, "ISO/IEC 27000:2018(E) Information technology – Security techniques – Information security management systems – Overview and vocabulary", 2018.
[2] BSI, "IT-Grundschutz-Kompendium – Edition 2023", online verfügbar unter: https://www.bsi.bund.de, 2023.

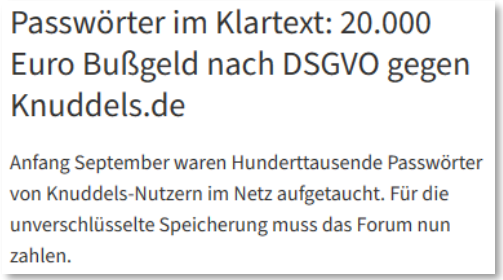
Motiviere: Vertraulichkeit

[3]

[5]



[4]



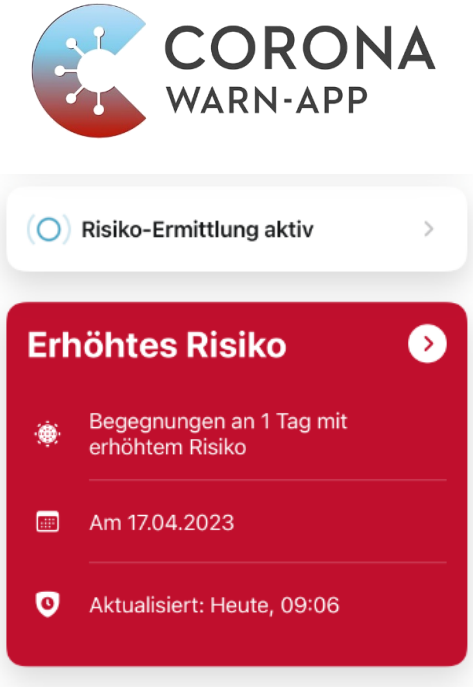
[6]

[7]

[3] ZEIT ONLINE, https://www.zeit.de/digital/datenschutz/2021-04/luca-app-sicherheitsluecken-datenschutz-kritik-corona (abgerufen: 09.11.21)
[4] CNBC, https://www.cnbc.com/2017/10/03/yahoo-every-single-account-3-billion-people-affected-in-2013-attack.html (abgerufen: 09.11.21)
[5] FORTUNE, https://fortune.com/2021/06/30/linkedin-data-theft-700-million-users-personal-information-cybersecurity (abgerufen: 09.11.21)
[6] HEISE, https://www.heise.de/news/Passwoerter-im-Klartext-20-000-Euro-Bussgeld-nach-DSGVO-gegen-Knuddels-de-4229798.html (abgerufen: 23.05.23)
[7] RNZ, https://www.rnz.de/politik/wirtschaft-regional_artikel,-Mannheim-Hackerangriff-auf-ABB-_arid,1112774.html (abgerufen: 23.05.23)

Vertraulichkeitsverletzungen finden

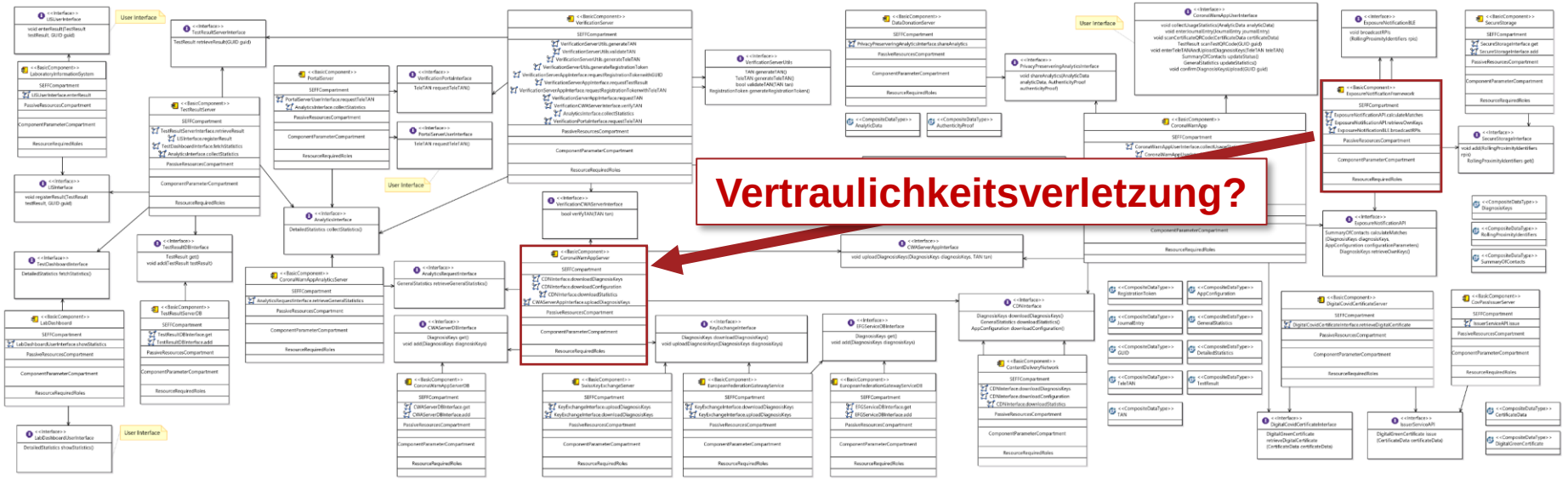
- Vorzeigeprojekt: Corona Warn App **[8]**
  - Entwickelt von SAP und Deutsche Telekom
  - Mehr als 48 Millionen mal heruntergeladen
  - Kosten: Mehr als 220 Millionen Euro **[9]**
  - Open Source, umfangreich von Sicherheitsforschenden geprüft
- Vielfältige sensible Daten, u.a.
  - Daten für Kontaktpersonennachverfolgung
  - Covid-19 Testergebnisse, Impfzertifikate
  - Persönliches Kontakt-Tagebuch



[8]

[8] Robert Koch-Institut (RKI), https://www.coronawarn.app/ (abgerufen 23.05.2023)
[9] SPIEGEL, https://www.zeit.de/gesundheit/2022-12/gesamtkosten-corona-warn-app-gesundheit-millionen (abgerufen: 23.05.2023)

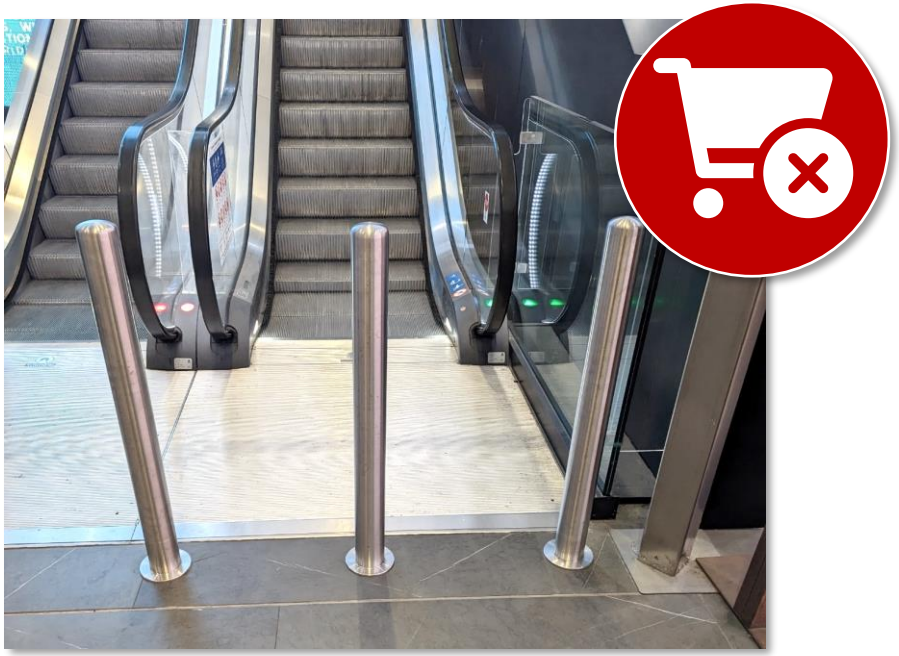
Vertraulichkeitsverletzungen finden

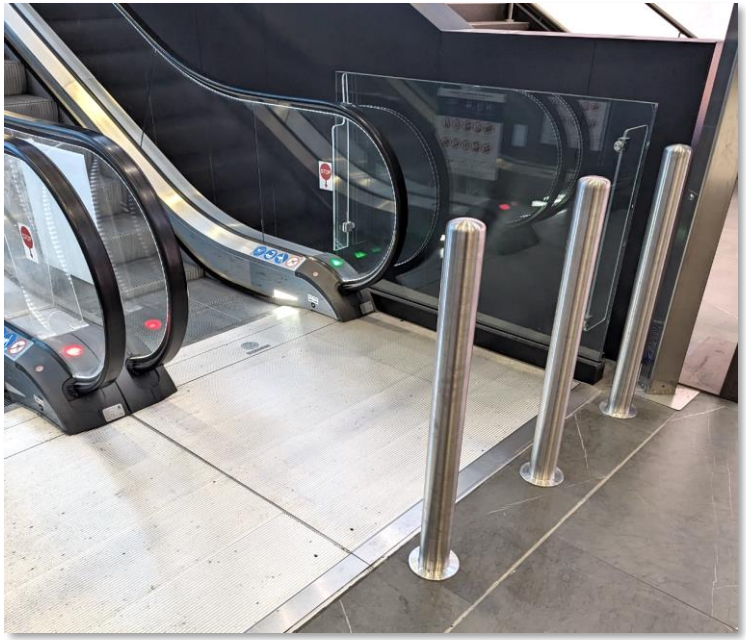


[10]

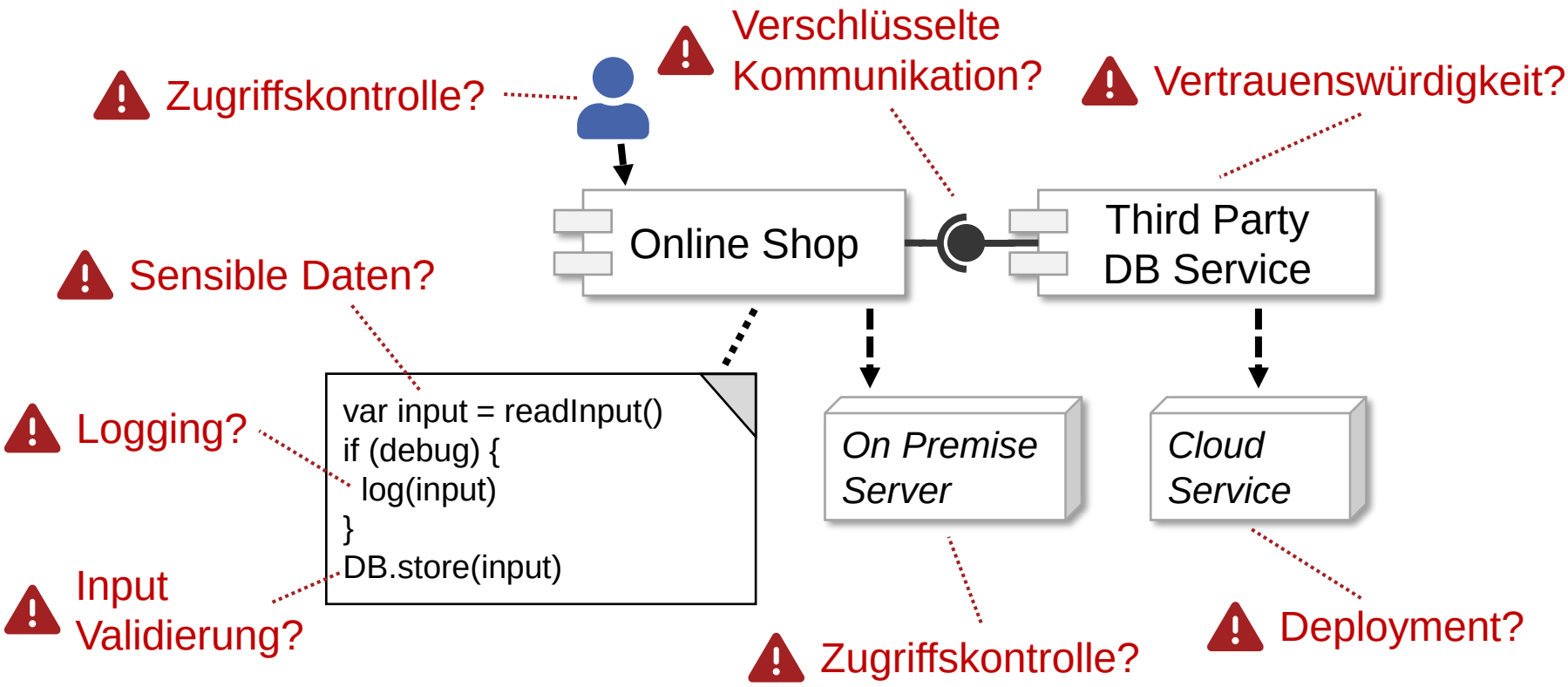
[10] S. Hahner, Corona Warn App Case Study, online verfügbar unter: https://abunai.dev/, 2023.

Vertraulichkeit als Qualitätseigenschaft





Ein letztes motivierendes Beispiel



Aufbau der Gastvorlesung

Erstes Lernziel erledigt ☺

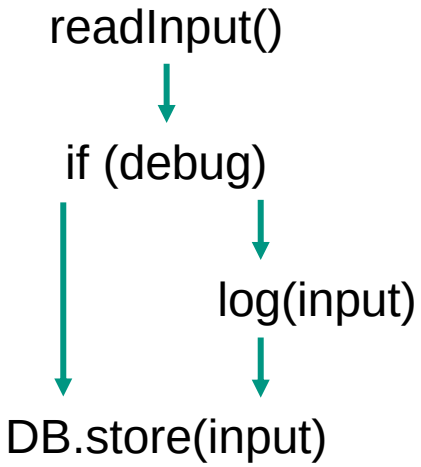
- [x] Vertraulichkeit definieren, damit verbundene Herausforderungen nennen

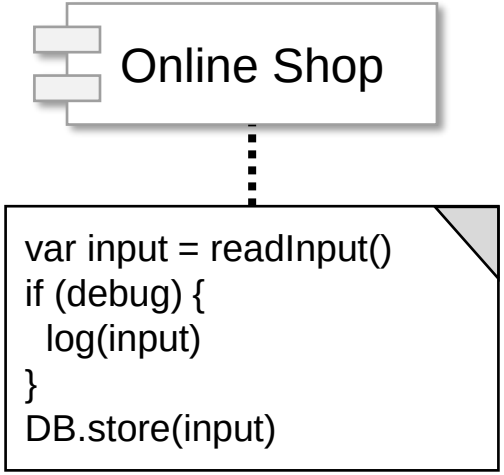
Inhalt

- Modellierung: Datenflussdiagramme und Vertraulichkeitsanforderungen
- Analyse: Label Propagation für effiziente Vertraulichkeitsanalyse
- Praxisbeispiel: Vertraulichkeitsanalyse der Corona Warn App
- Fazit: Ausblick auf aktuelle Forschung und Zusammenfassung

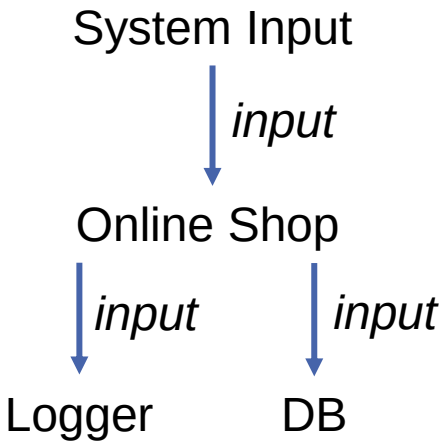
Kontrollfluss vs. Datenfluss, implizit vs. explizit

Kontrollflussgraph (CFG)





Datenflussdiagramm (DFD)

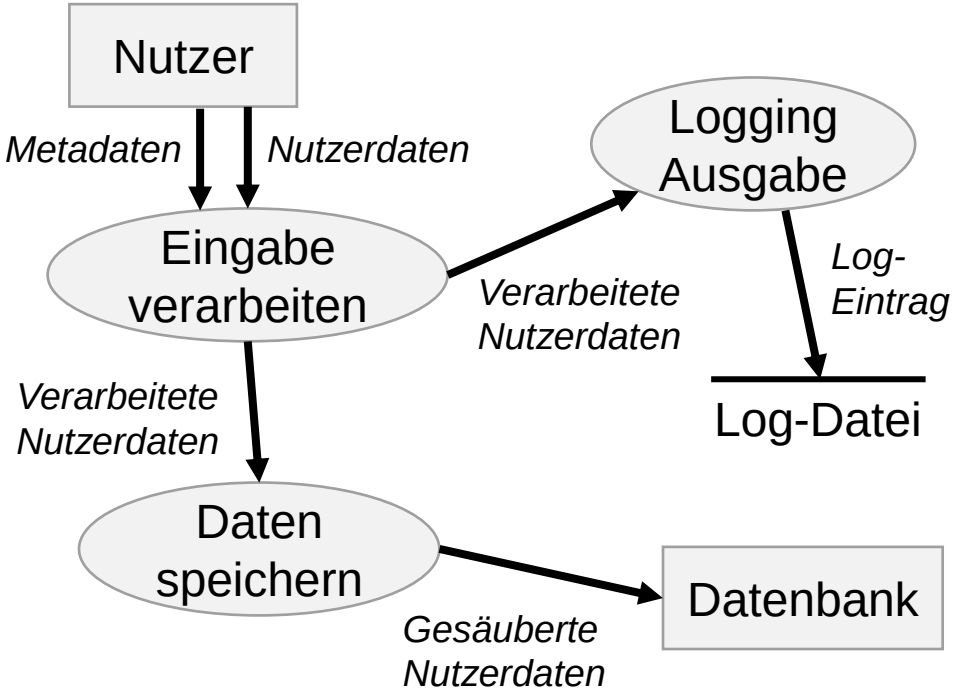


In dieser Vorlesung: Fokus auf explizite Datenflüsse, *nicht* implizite Informationsflüsse!

Syntax von Datenflussdiagrammen

Konkrete Syntax (nach de Marco [11])

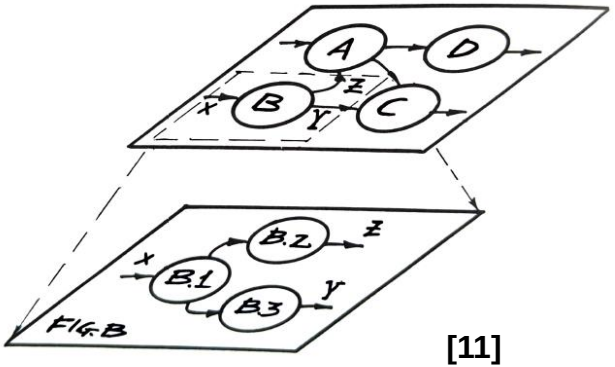
- Datenflüsse für jedes relevante Datum, dargestellt als Pfeile
- Prozesse, dargestellt als Kreise
- Dateien sind (temporäre) Datenspeicher im System, dargestellt als Linien
- Datenquellen und –senken außerhalb des Kontexts des modellierten Systems, dargestellt als Rechtecke



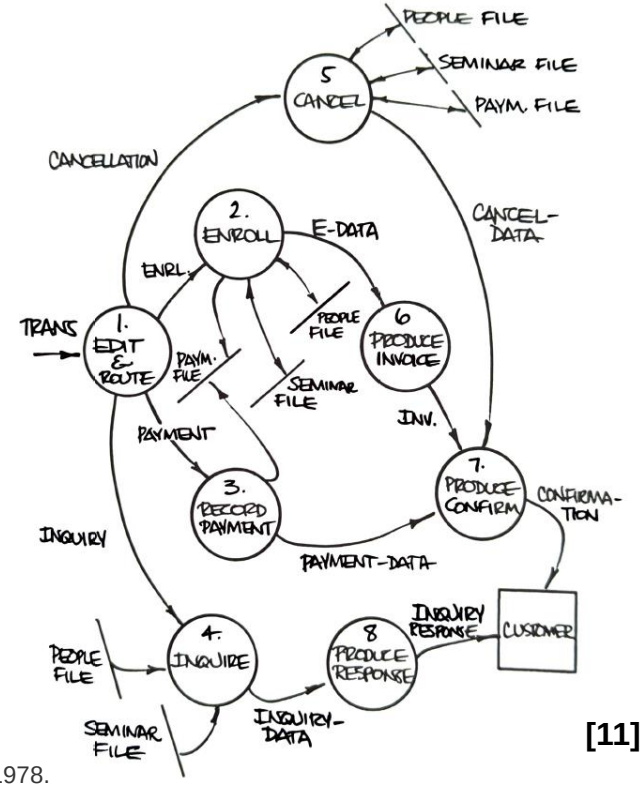
[11] T. de Marco, „Structured Analysis and System Specification“, YOURDON inc., New York, 1978.

Abstraktion in Datenflussdiagrammen

- Datenflussdiagramme können aus mehreren Ebenen bestehen
- Es sollte der für die Analyse passende Abstraktionsgrad verwendet werden



[11]



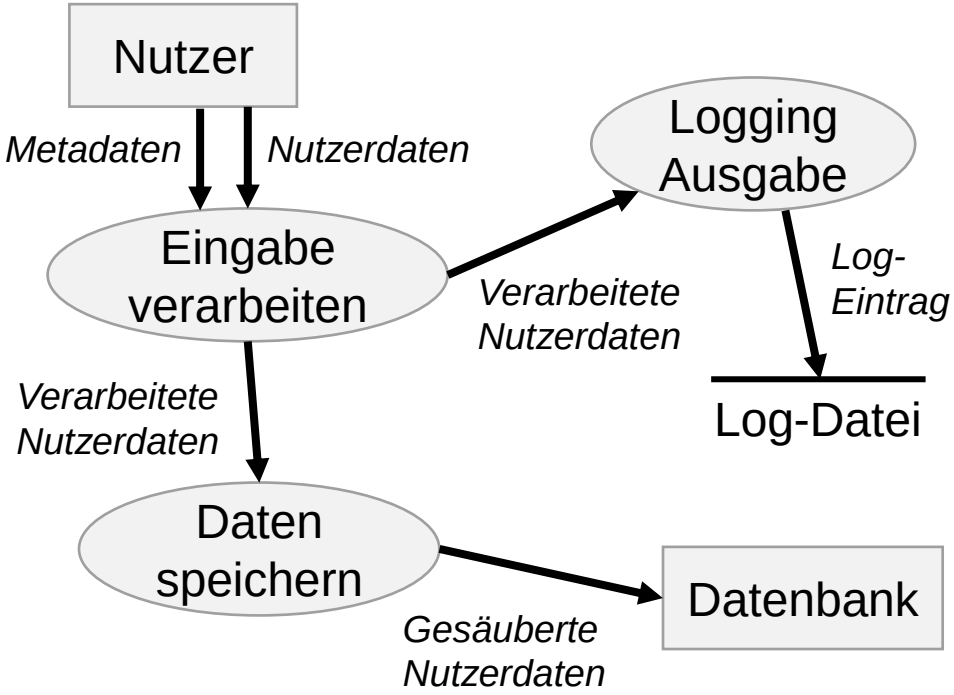
[11]

[11] T. de Marco, „Structured Analysis and System Specification“, YOURDON inc., New York, 1978.

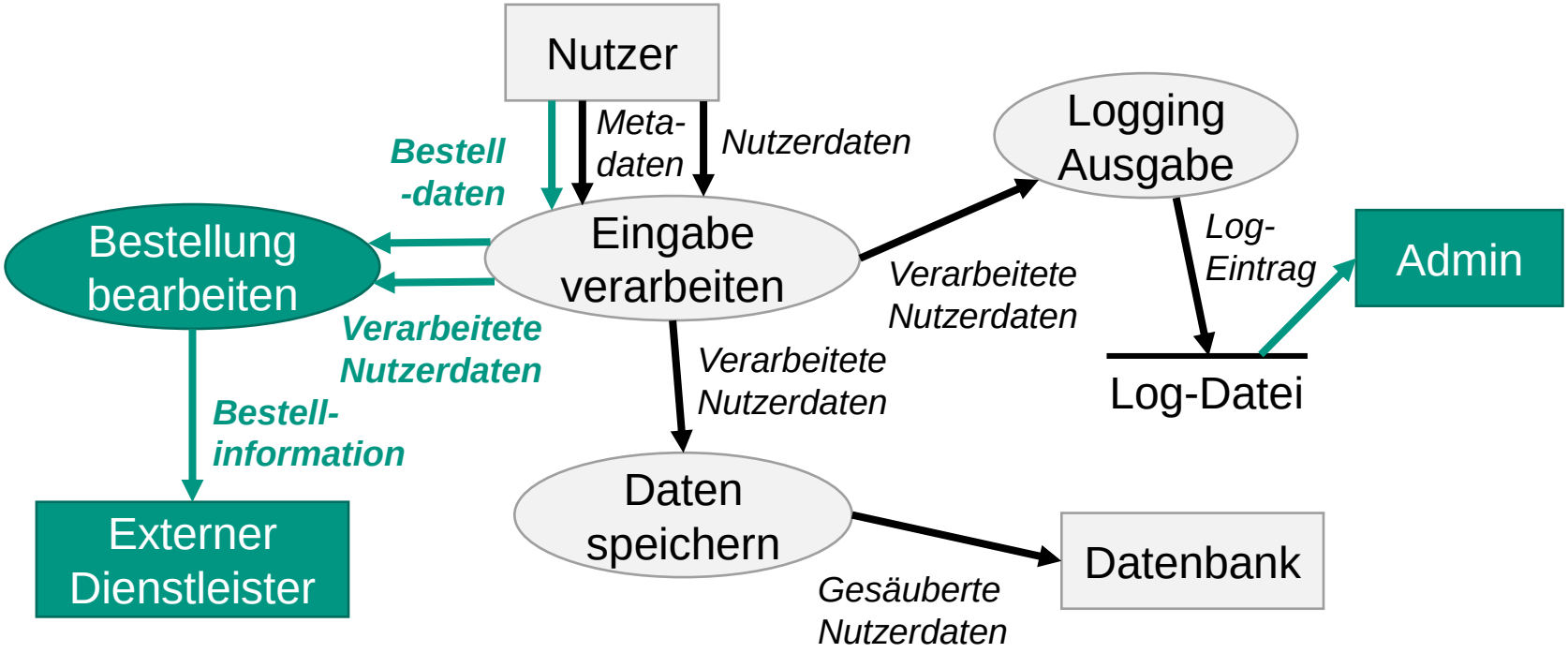
Übung zu Datenflussdiagrammen

- **Übung:** Erweitern Sie das Diagramm
  1. Nutzer geben auch Bestelldaten ein
  2. Verarbeitete Nutzerdaten und Bestelldaten werden zur Logistik für die Bearbeitung der Bestellung gesendet. Anschließend wird die Bestellinformation einem externen Dienstleister weitergegeben
  3. Admins können die Inhalte der Log-Datei auslesen

- Think-Pair-Share: 2 Minuten denken, 3 Minuten diskutieren

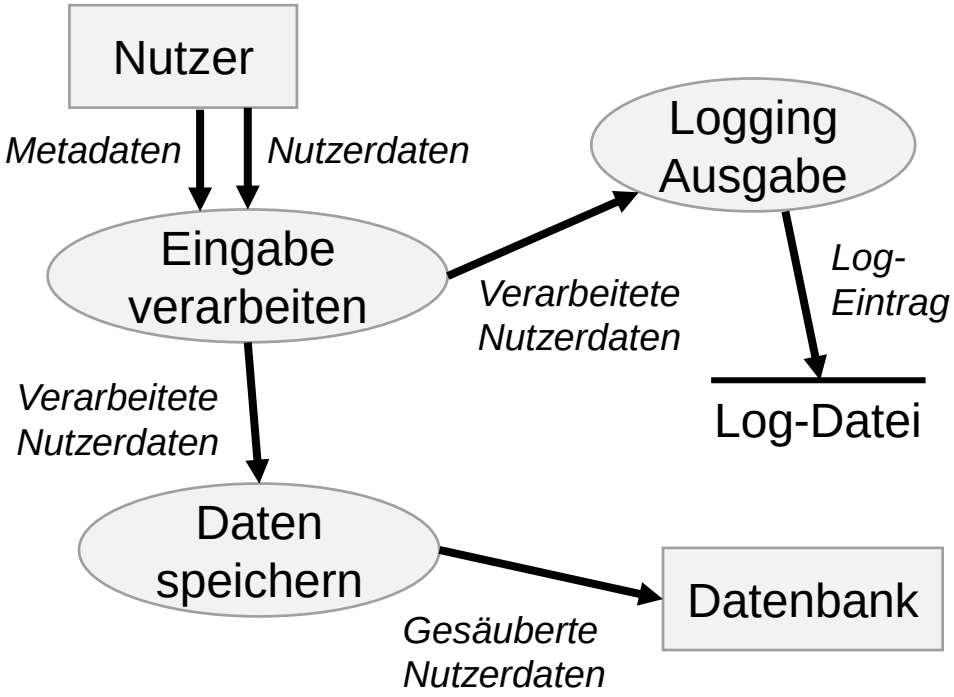


Eine mögliche Lösung



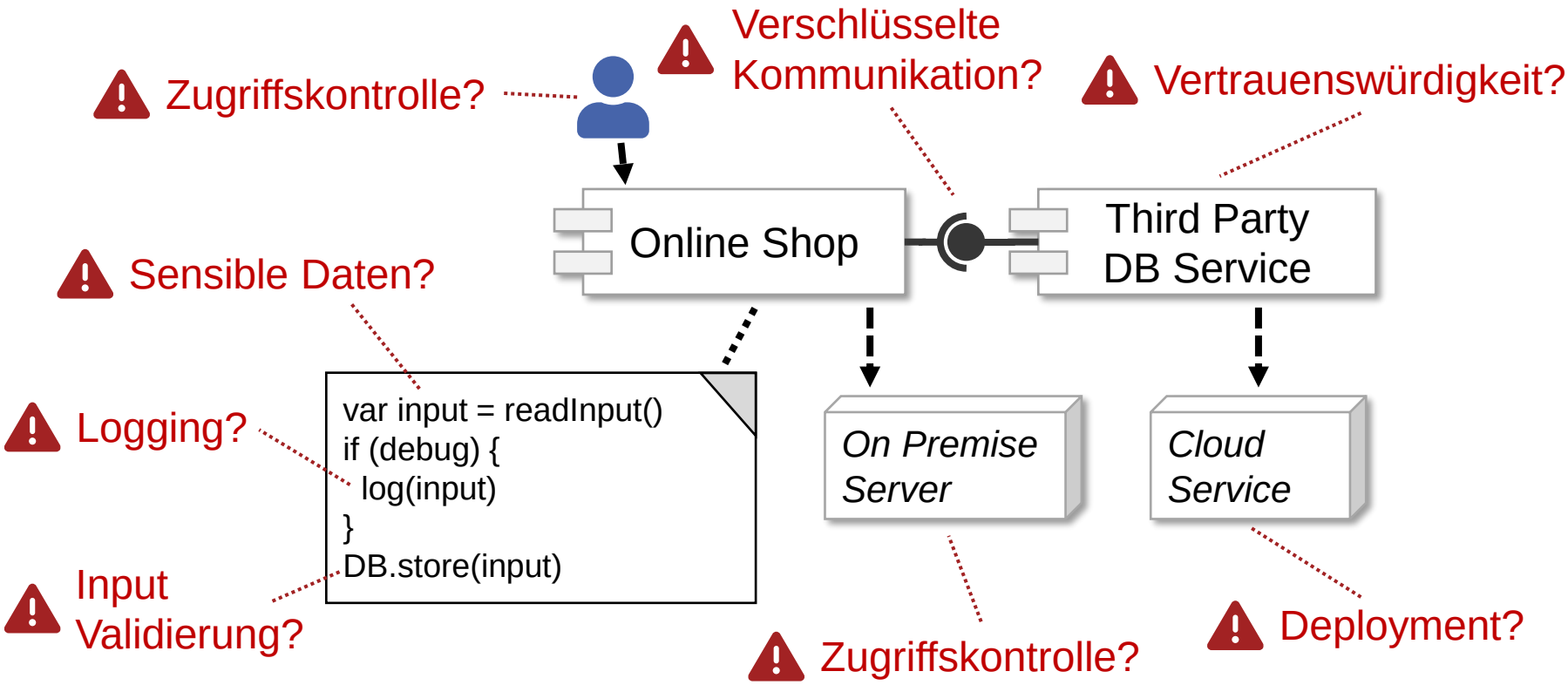
Formale Darstellung

- Schleifenfreiheit ist nicht automatisch gegeben, aber durch Umformung möglich **[12]**
- Dann darstellbar als gerichteter, azyklischer Graph (DAG) G = (V, E), mit Knoten V und Kanten E
- Der Datenfluss ist dann eine strikte partielle Ordnung u ≺ v
  - Irreflexiv, also nie u ≺ u
  - Asymmetrisch, also u ≺ v ⇒ ¬ (v ≺ u)
  - Transitiv, also a ≺ b ∧ b ≺ c ⇒ a ≺ c



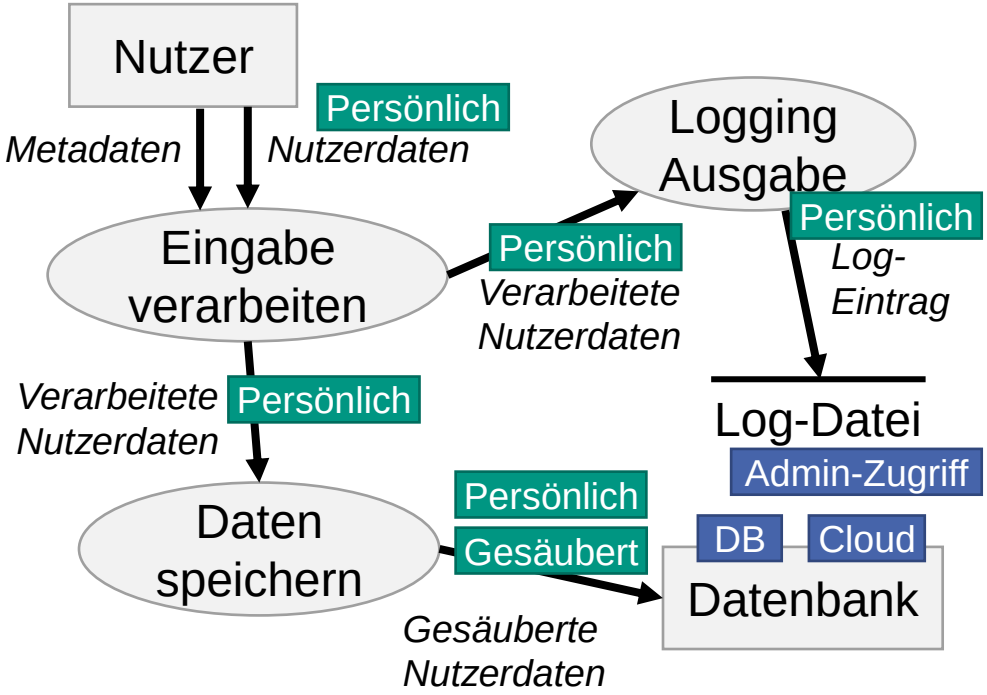
[12] R. Kramer, et al., „The combining DAG: a technique for parallel data flow analysis", In: *IEEE Transactions on Parallel and Distributed Systems*, 1994.

Offenes Problem: Woher kommt die Sicherheit?



Annotation von Vertraulichkeitseigenschaften

- Für die Analyse der Vertraulichkeit sind nicht die fließenden Daten entscheidend, *sondern* deren Eigenschaften [13]
- Annotation von Charakteristiken
  - Datencharakteristiken beschreiben für die Vertraulichkeit relevante Eigenschaften, z.B. Personenbezug oder Verschlüsselung der Daten
  - Knotencharakteristiken beschreiben Eigenschaften der von den Knoten repräsentierten Elemente, z.B. Deployment oder Zugriffskontrolle

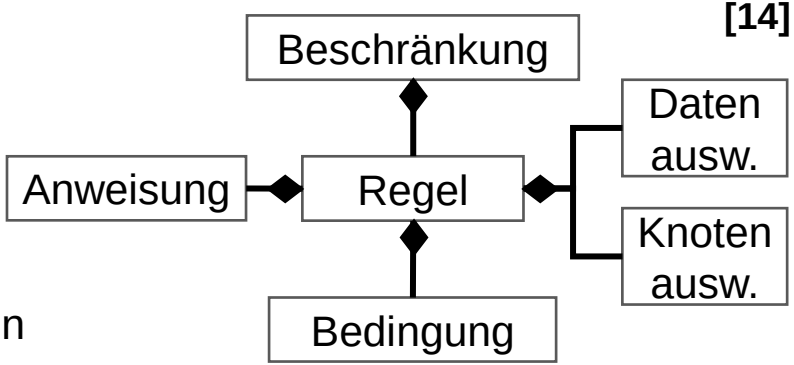


[13] S. Seifermann, et al., “Detecting violations of access control and information flow policies in data flow diagrams”, In: *JSS*, vol. 184, 2022.

Definition von Vertraulichkeitsanforderungen

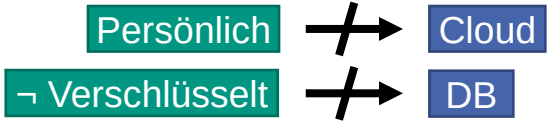
- Für die Formulierung von Vertraulichkeits-anforderungen kommen oft Muster **[14]** oder Schranken **[15]** zum Einsatz
  - Datenflussschranken (engl. *Constraints*) können auch genutzt werden, um Muster auszudrücken und zu analysieren
  - Aussagen über den Fluss von Daten mit ausgewählten Datencharakteristiken zu Knoten mit ausgewählten Knotencharakteristiken

[14]



Beispiele

- Kein Fluss persönlicher Daten in die Cloud
- Kein Fluss unverschlüsselter Daten zur DB

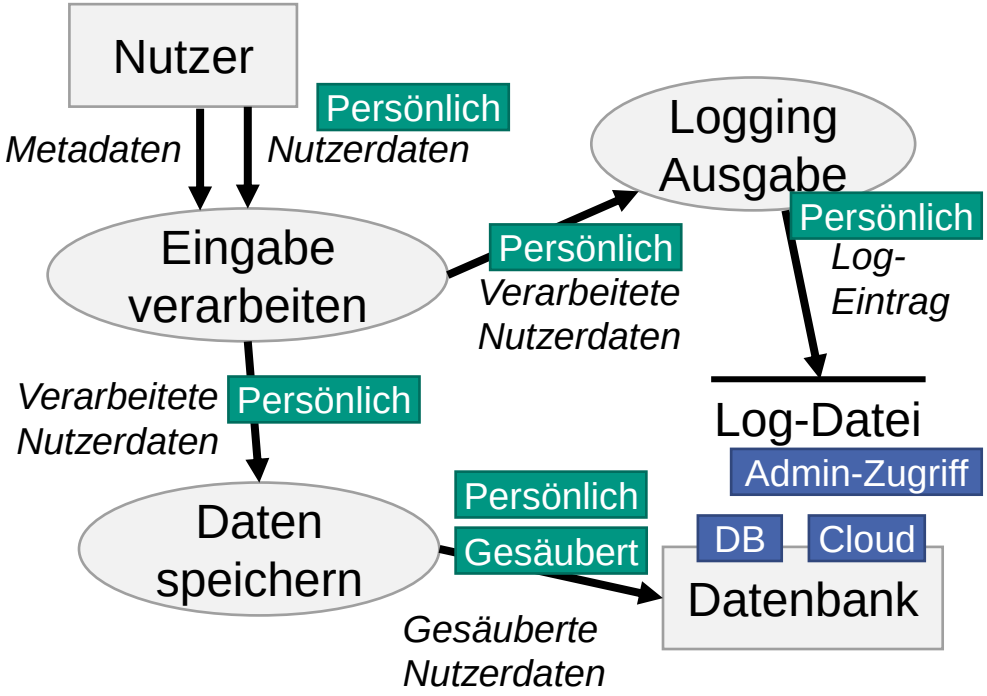


[14] A. Bambhore Tukaram, et al., “Towards a security benchmark for the architectural design of microservice applications”, In: *ARES*, ACM, 2022.
[15] S. Hahner et al., “Modeling Data Flow Constraints for Design-Time Confidentiality Analyses,” presented at *ICSA*, IEEE, 2021.

Charakteristiken und Anforderungen

Frage: Werden die folgenden Anforderungen eingehalten?

1. Persönlich ↛ Admin-Zugriff ✗
2. Persönlich ↛ Cloud ✗
3. ¬ Gesäubert ↛ DB ✓
4. Persönlich ∧ ¬ Verschlüsselt ↛ Cloud ∨ Admin-Zugriff ✗ ✗

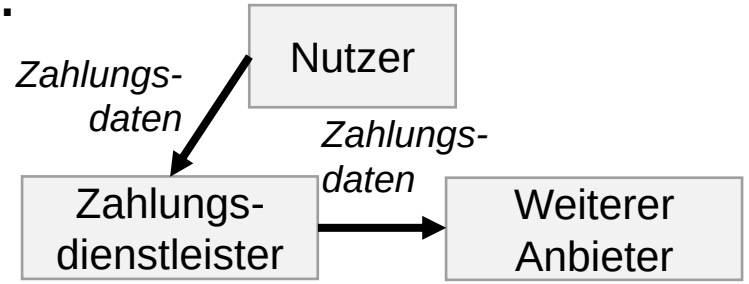


Übung zu Vertraulichkeitsanforderungen

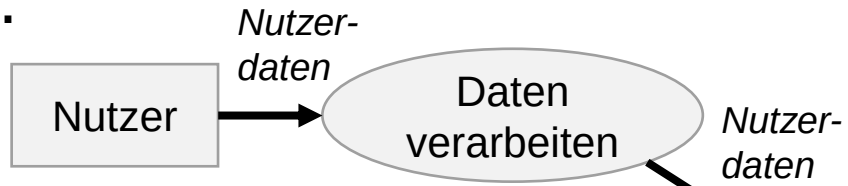
- **Übung:** Annotieren Sie Charakteristiken an die drei Datenflussdiagramme. Formulieren Sie Schranken für die Beispiele von Folie 3:
  1. Zahlungsdaten dürfen nur dem Zahlungsdienstleister offengelegt werden
  2. Daten müssen auf der Datenbank verschlüsselt gespeichert werden
  3. Konkrete Prüfungsinhalte dürfen nicht kommuniziert werden ☺

- Think-Pair-Share: 2 Minuten denken, 3 Minuten diskutieren

1.

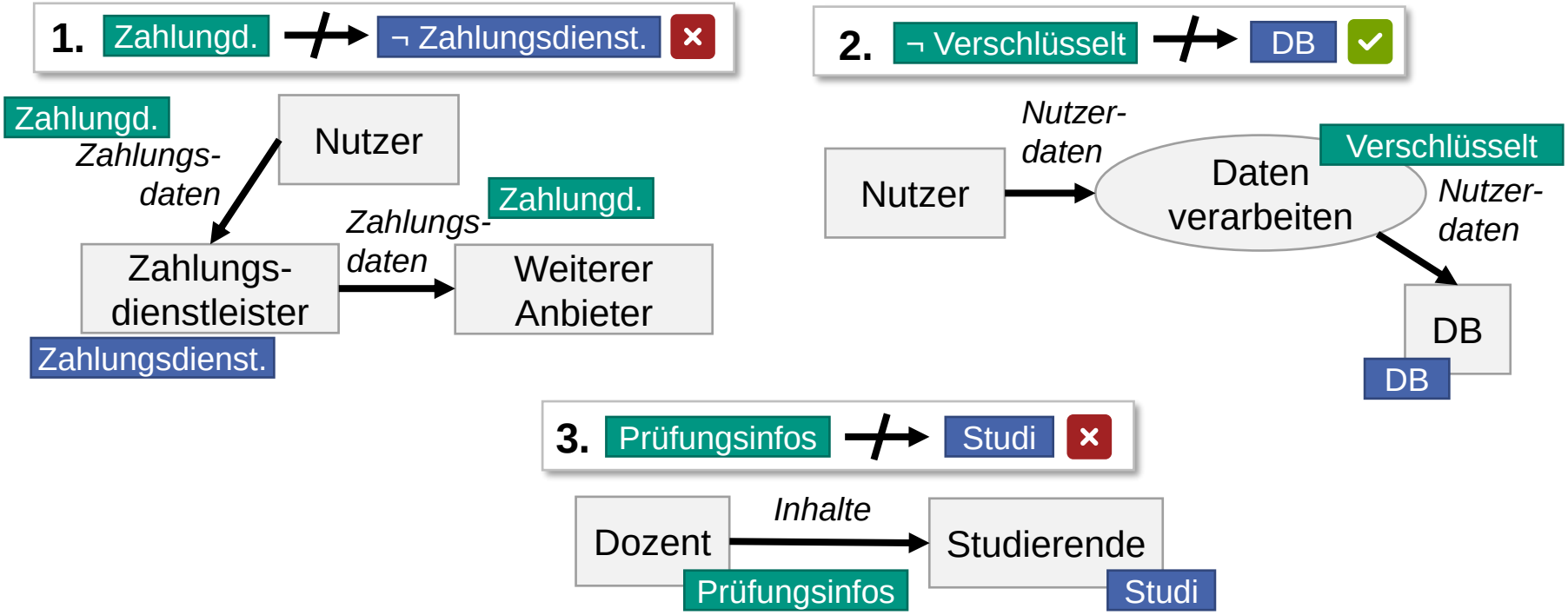


2.



3.

Eine mögliche Lösung

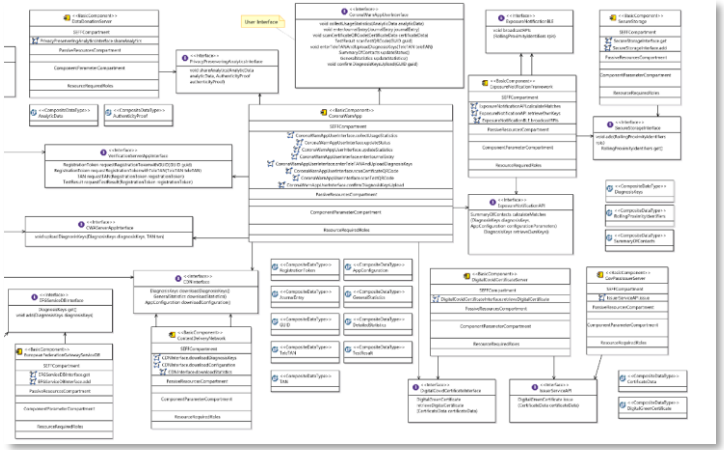


Zwischenfazit

- Datenflussdiagramme bestehen aus benannten Flüssen, Prozessen, Dateien, Datenquellen und –senken
- Datenflussdiagramme eignen sich gut um Vertraulichkeit zu bewerten: *“problems tend to follow the data flow, not the control flow”* **[16]**
- Hierfür annotiert man Charakteristiken und überprüft Datenflussschranken

⇨ 3 / 4 Lernziele erledigt ☺

Offenes Problem: Wie analysiert man das *effizient*?



[16] A. Shostack, Threat Modeling: Designing for Security. John Wiley & Sons, 12, 2014.

Architekturbasierte Vertraulichkeitsanalyse

- Problem: Manuelle Analyse der Charakteristiken auf Einhaltung der Vertraulichkeitsanforderung ist aufwendig und fehleranfällig **[17]**
- Ansatz: Modellierung der Software-Architektur, tool-gestützte Extraktion aller möglichen Datenflüsse, automatische Analyse auf Einhaltung aller Vertraulichkeitsanforderungen

In dieser Vorlesung

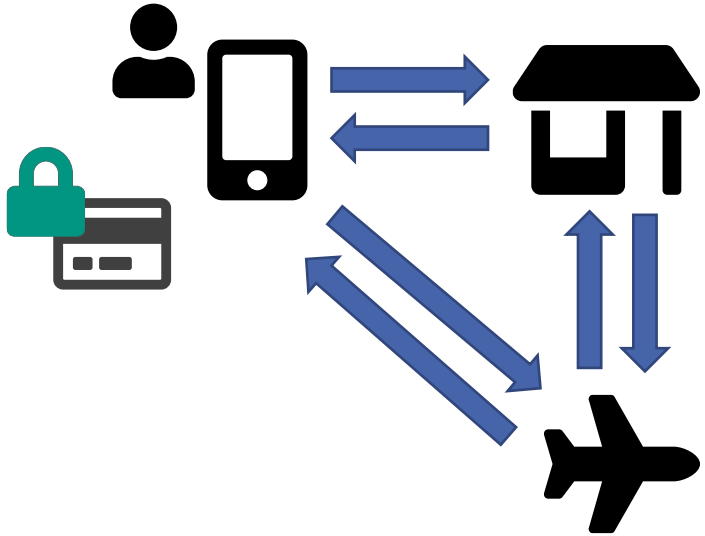
- Kein Fokus auf die Modellierung der Software-Architektur **[18]**
- Stattdessen Fokus auf die Technik zur automatischen Analyse
- Die folgenden Folien stammen aus einem öffentlichen Vortrag **[19]**

[17] S. Seifermann, et al., “Data-Driven Software Architecture for Analyzing Confidentiality”, In: *ICSA*, IEEE, 2019.
[18] S. Seifermann, et al., “Identifying Confidentiality Violations in Architectural Design Using Palladio”, In: *ECSA-C*, 2021.
[19] S. Hahner, “Privacy by (Early) Design”, VKSI Sneak Preview, Invited Talk, 2021. Aufzeichnung: https://www.youtube.com/watch?v=BiE8yaY8rb0&t=3987s

Die Travel-Planner Fallstudie

- Beschreibt den Buchungsvorgang von Flügen über eine Reiseagentur
- Nutzer wählen über eine Smartphone-App einen Flug, buchen diesen unter Eingabe ihrer Kreditkartendaten
- Flugdaten sind öffentlich, Kreditkartendaten sind vertraulich

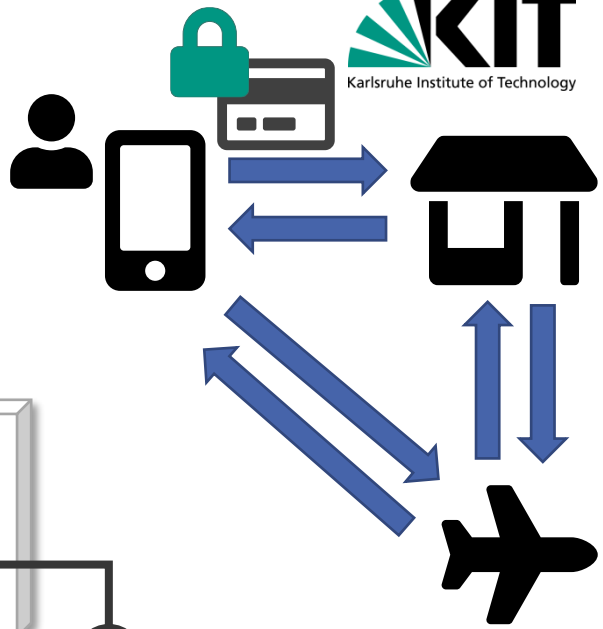
- Im Rahmen von iFlow veröffentlicht **[20]**, vielfach zur Evaluation verwendet **[13,15]**

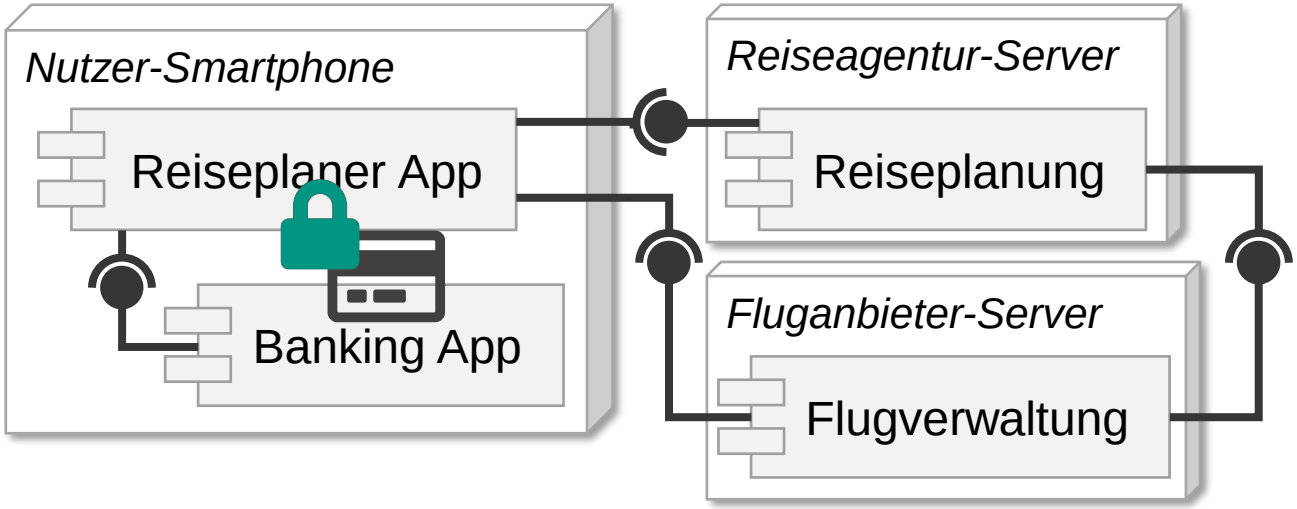


[13] S. Seifermann, et al., “Detecting violations of access control and information flow policies in data flow diagrams”, In: *JSS*, vol. 184, 2022.
[15] S. Hahner, et al., “Modeling Data Flow Constraints for Design-Time Confidentiality Analyses,” presented at *ICSA*, IEEE, 2021.
[20] K. Katkalov, “Ein modellgetriebener Ansatz zur Entwicklung informationsflusssicherer Systeme”, University of Augsburg, 2017.

Vertrauliche Daten im „Travel Planner“

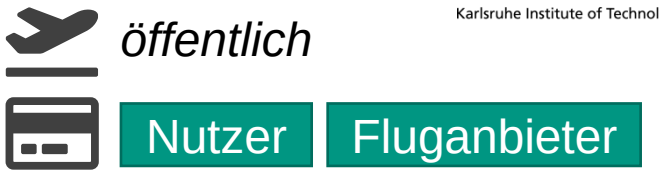
- Datenflussbeschränkungen
  - Kreditkartendaten nach Freigabe zum Fluganbieter
  - Kreditkartendaten niemals zur Reiseagentur

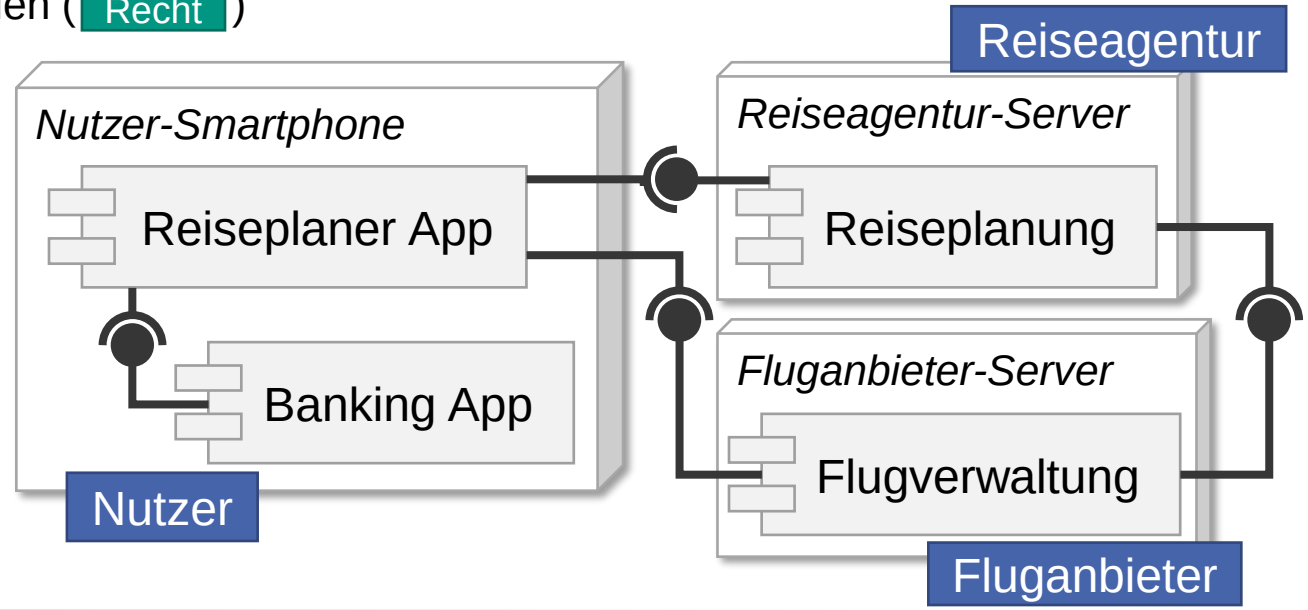




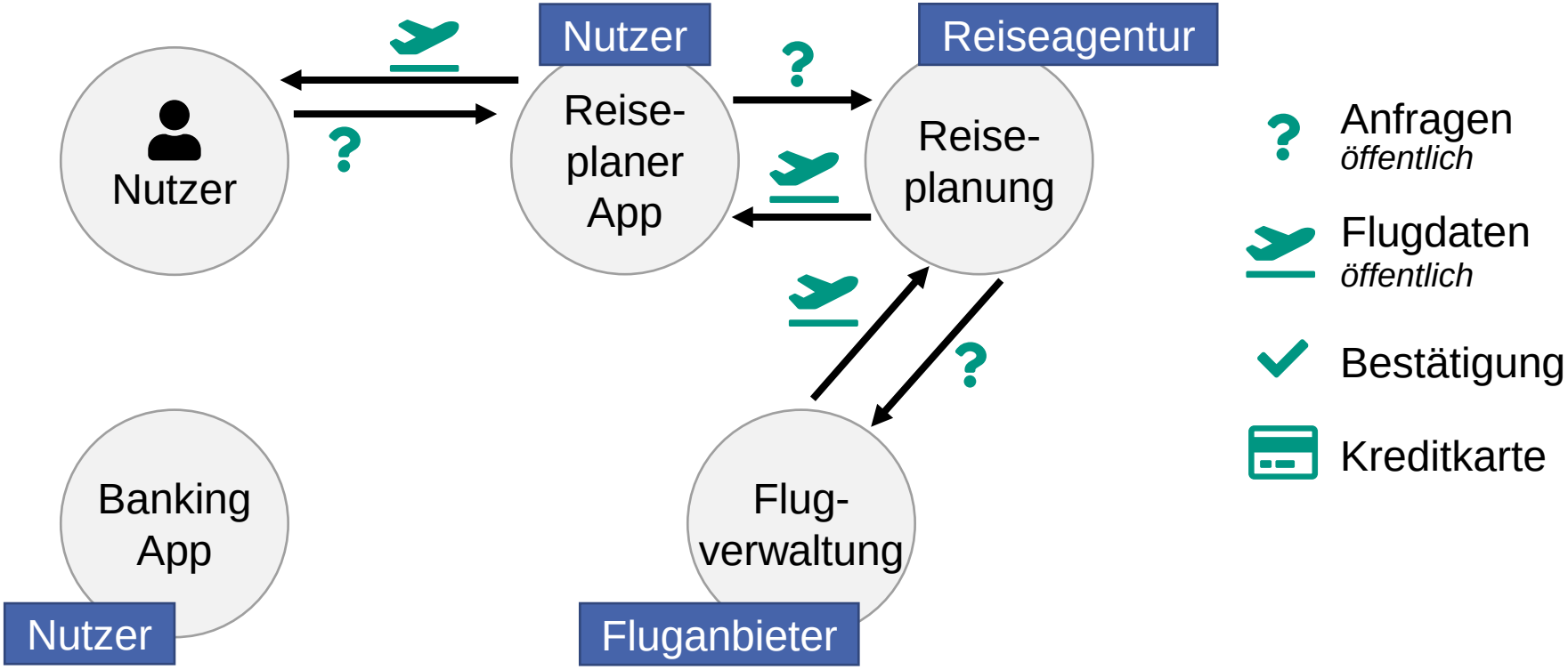
Annotation von Zugriffskontrolle

- Rollenbasierte Zugriffskontrolle
  - Rollen (Rolle)
  - Zugriffsrechte der Rollen (Recht)
- Arten von Daten
  - Flugdaten
  - Kreditkartendaten
- Ableitung von Datenflüssen aus modelliertem Verhalten

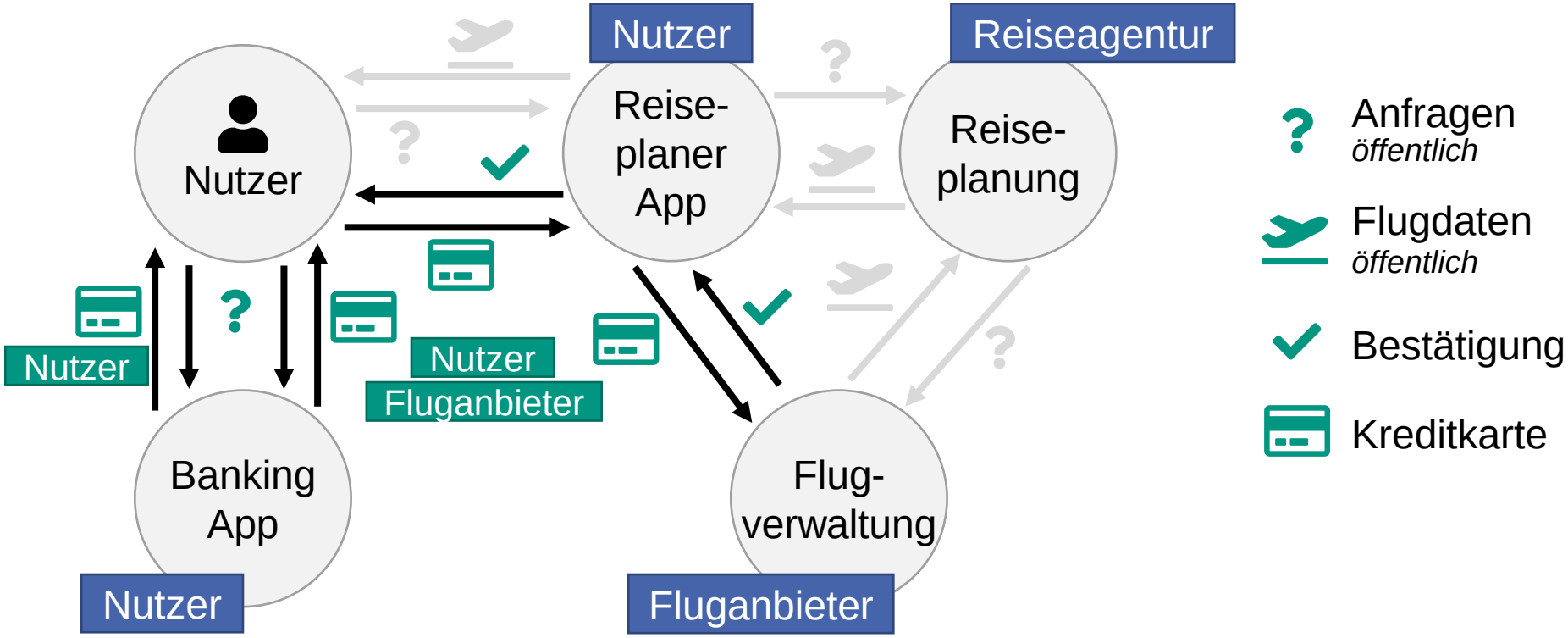




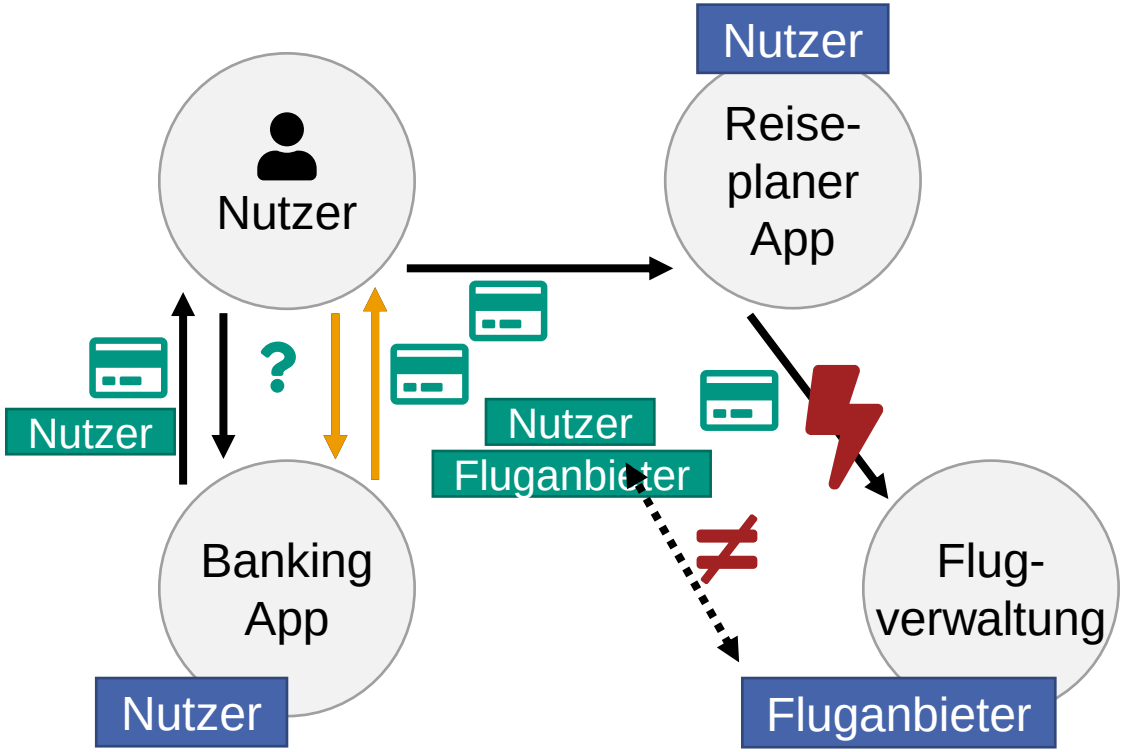
Label Propagation

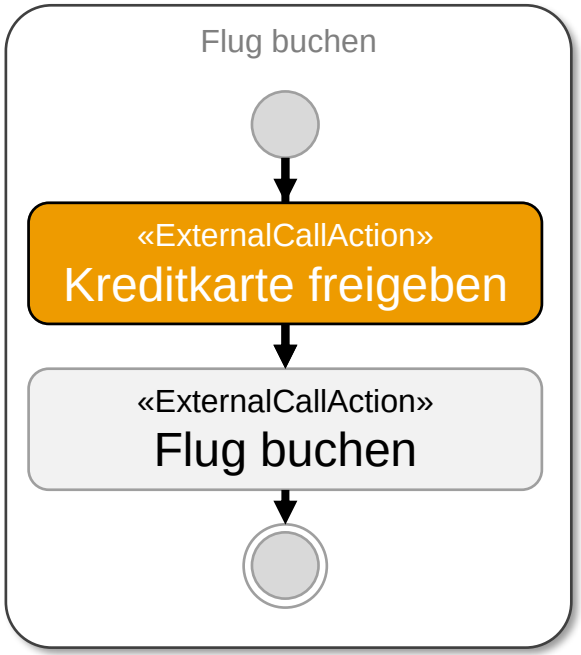


Label Propagation



Ergebnis der Analyse

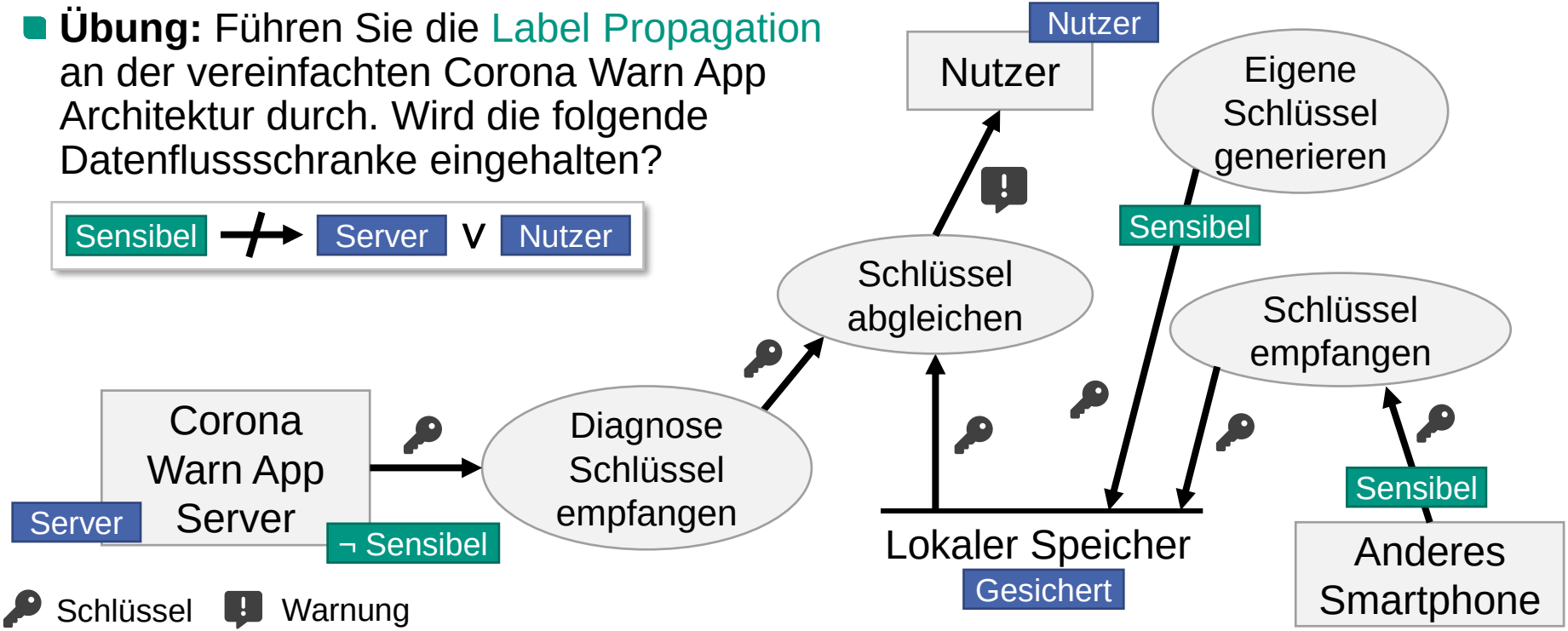




Übung zu Label Propagation

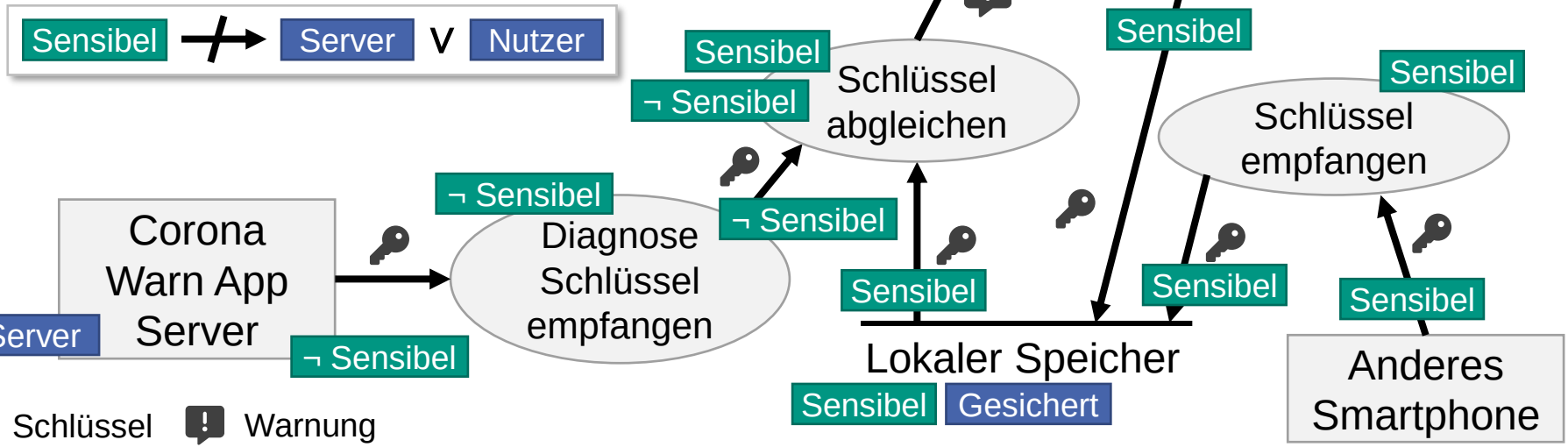
- **Übung:** Führen Sie die Label Propagation an der vereinfachten Corona Warn App Architektur durch. Wird die folgende Datenflussschranke eingehalten?

Sensibel ↛ Server ∨ Nutzer



Lösung zu Label Propagation

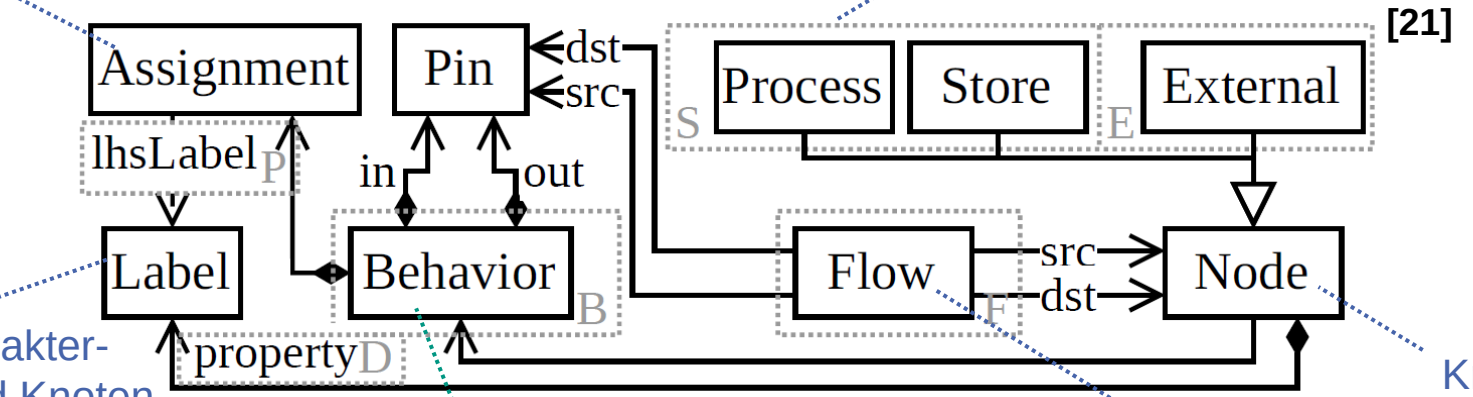
- **Lösung:** Die Schranke wird eingehalten, weil der Nutzer nur über das Ergebnis des Abgleichs informiert wird, nicht über die verglichenen Schlüssel:



Im Detail: Erweiterte Syntax der Diagramme

Annotation oder Berechnung von Charakteristiken

Verschiedene Knotentypen wie in der Syntax von de Marco [21]



Datencharakter-istiken und Knoten-charakteristiken

Verhalten, definiert als aussagenlogische Formel. Legt Ausgabe basierend auf Eingabe fest. Beispiel: ¬ Verschlüsselt → Verschlüsselt

Kante im DFD

Knoten im DFD

[21] S. Seifermann, et al., “A Unified Model to Detect Information Flow and Access Control Violations in Software Architectures”, In: *SECRYPT*, 2021.

Im Detail: Prototypische Umsetzung

Erste Prototypen

- Modellierung mittels Palladio **[22]**
- Extraktion des vollständigen, annotierten Datenflussdiagramms
- Transformation in Fakten und Regeln der logischen Programmiersprache Prolog **[13]**
- Definition der Datenflussschranke als Prolog-Query **[15]**
- Rückübersetzung der Ergebnisse

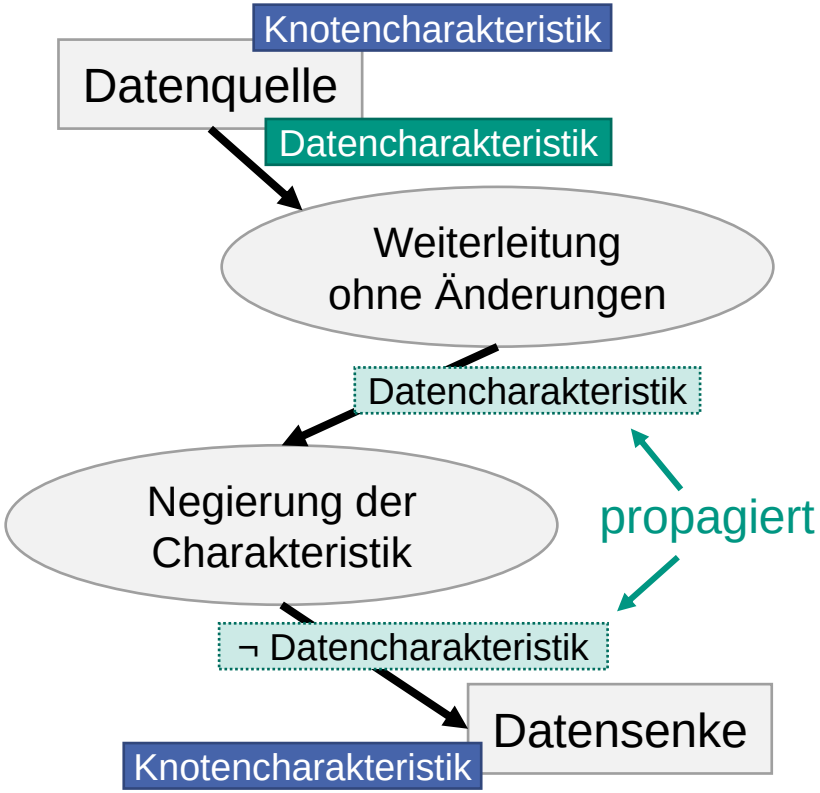
Aktueller Prototyp

- Modellierung mittels Palladio **[22]**
- Extraktion aller möglicher Datenflüsse bzw. Sequenzen
- Propagation der annotierten Charakteristiken entlang aller identifizierten Sequenzen
- Formulierung der Schranke als Prädikat in Java, Überprüfung auf jedem Datenflussknoten

[13] S. Seifermann, et al., “Detecting violations of access control and information flow policies in data flow diagrams”, In: *JSS*, vol. 184, 2022.
[15] S. Hahner, et al., “Modeling Data Flow Constraints for Design-Time Confidentiality Analyses,” presented at *ICSA*, IEEE, 2021.
[22] R. H. Reussner, et al., “Modeling and Simulating Software Architectures: The Palladio Approach”, The MIT Press, 2016.

Zwischenfazit

- Label Propagation ermöglicht die effiziente Analyse der Vertraulichkeit
- Nutzung von Datenflussdiagrammen, annotierten Charakteristiken und definierten Datenflussschranken
- Propagiert Datencharakteristiken entlang aller möglichen Datenflüsse, kommt einer Tiefensuche gleich, mit einer Komplexität von $O(|V| + |E|)$
- Knotencharakteristiken werden **nicht** propagiert und nur zum Überprüfen der Schranken verwendet



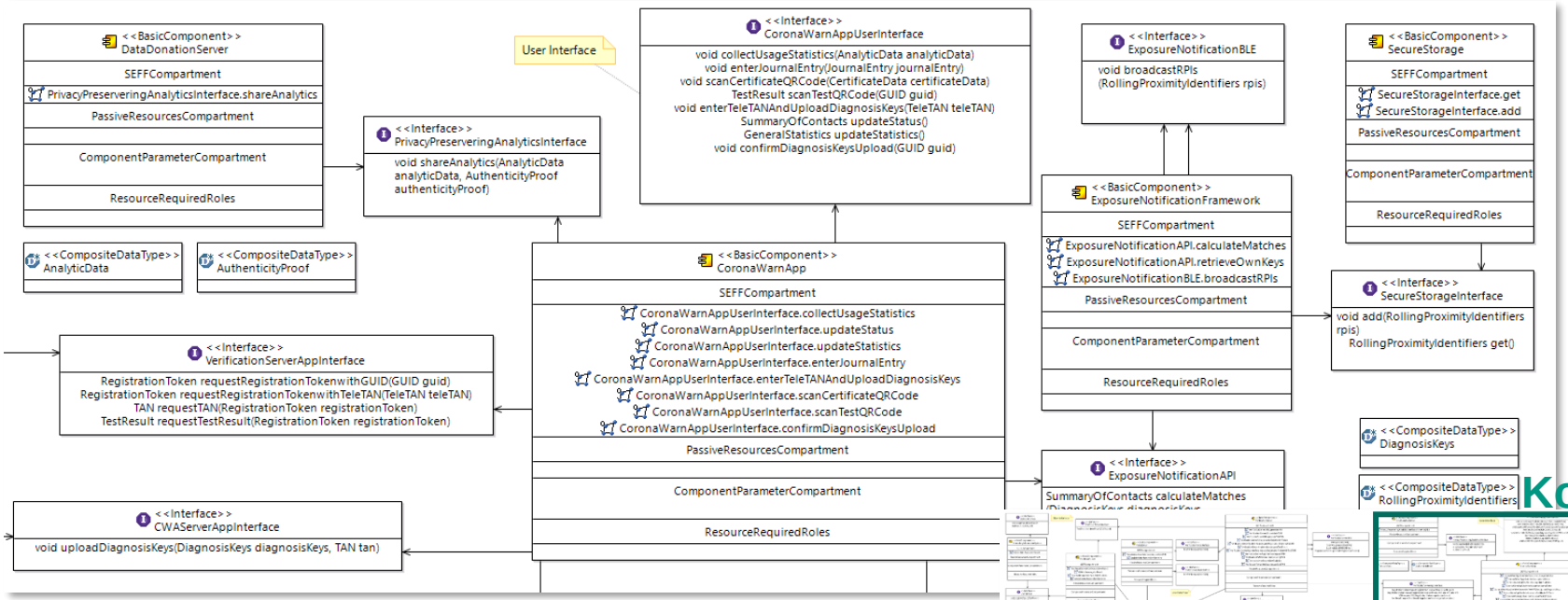
Praxisbeispiel: Corona Warn App

- Corona Warn App Palladio-Modell, erstellt für die Evaluation einer verwandten Analyse **[23]**
- Basierend auf öffentlich verfügbarer Dokumentation modelliert **[24]**
- Besteht aus 19 Komponenten und einem Datenflussdiagramm aus 200 Knoten (500 Elemente insgesamt)
- Alle Artefakte und Prototypen sind online verfügbar **[10]**

- Modellierte Funktionalität, u.a.
  - Schlüsselaustausch via Bluetooth
  - Schlüsselverwaltung, Server-Analytics
  - Testergebnis via QR Code empfangen
  - Testergebnis via Hotline empfangen
  - Warnfunktion bei positivem Testergebnis
  - Impfzertifikate via CovPass abrufen
  - Labor-Schnittstelle für Testergebnisse
  - Internationaler Schlüsselaustausch
  - Privates Kontakt-Tagebuch

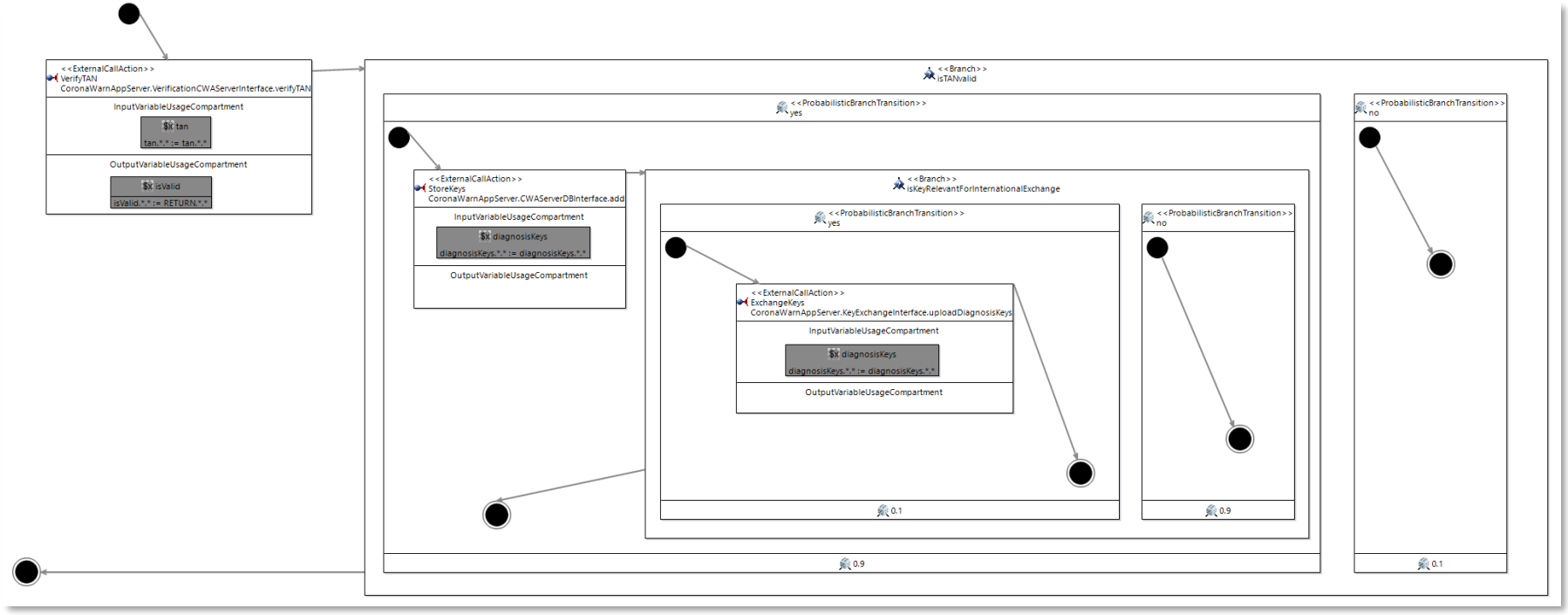
[10] S. Hahner, Corona Warn App Case Study, online verfügbar unter: https://abunai.dev/, 2023.
[23] S. Hahner, et al., “Architecture-based Uncertainty Impact Analysis to ensure Confidentiality”, In: *SEAMS*, IEEE/ACM, 2023. Accepted, to appear.
[24] Robert Koch-Institut (RKI), CWA Dokumentation, https://github.com/corona-warn-app/cwa-documentation (abgerufen 24.05.2023)

Corona Warn App: Komponenten



Corona Warn App: Verhalten

SEFF: CoronaWarnAppServer.uploadDiagnosisKeys



Corona Warn App: Extrahierte Datenflüsse

I	J	K
ID: 4 *(User enters journal entry)*	**ID: 5** *(User scans vaccination certificate)*	**ID: 6** *(User updates status and statistics)*
CallingUserActionSequenceElement / calling (EnterNewJournalEntry, _UbqtQLZ5Ee2xLZp9hElpuA))	CallingUserActionSequenceElement / calling (ScanCertificate, _oJMyYLZ6Ee2xLZp9hElpuA))	CallingUserActionSequenceElement / calling (UpdateStatistics, _b_q_ULZ7Ee2xLZp9hElpuA))
SEFFActionSequenceElement (Beginning enterJournalEntry, _ssANgLK8Ee2Y1pKtbIeM6Q))	SEFFActionSequenceElement (Beginning scanCertificateQRCode, _wfCaQLK8Ee2Y1pKtbIeM6Q))	SEFFActionSequenceElement (Beginning updateStatistics, _reSdMLK8Ee2Y1pKtbIeM6Q))
SEFFActionSequenceElement (StoreEntry, _NPsfsMK3Ee28l_bdB_O6zg))	CallingSEFFActionSequenceElement / calling (RetrieveCertificate, _PAn3gLkDEe25g4oGy4Jxtw))	CallingSEFFActionSequenceElement / calling (DownloadStatistics, _C45C8Lj-Ee25g4oGy4Jxtw))
CallingUserActionSequenceElement / returning (EnterNewJournalEntry, _UbqtQLZ5Ee2xLZp9hElpuA))	SEFFActionSequenceElement (Beginning retrieveDigitalCertificate, _2v6MILKgEe2Y1pKtbIeM6Q))	SEFFActionSequenceElement (Beginning downloadStatistics, _IvysgbLAEe2Y1pKtbIeM6Q))
	CallingSEFFActionSequenceElement / calling (IssueCertificate, _jxGdwLdLEe2lCsB2iGU-8w))	CallingSEFFActionSequenceElement / calling (RetrieveData, _wAzIkLmgEe2dIMSi7oNVYQ))
	SEFFActionSequenceElement (Beginning issue, _4HylELKgEe2Y1pKtbIeM6Q))	SEFFActionSequenceElement (Beginning downloadStatistics, _h4vicbLAEe2Y1pKtbIeM6Q))
	SEFFActionSequenceElement (ReturnDigitalCertificate, _FLa7oLdQEe2lCsB2iGU-8w))	CallingSEFFActionSequenceElement / calling (RetrieveStatistics, _v648ELmmEe2dIMSi7oNVYQ))
	CallingSEFFActionSequenceElement / returning (IssueCertificate, _jxGdwLdLEe2lCsB2iGU-8w))	SEFFActionSequenceElement (Beginning retrieveGeneralStatistics, _bGOwwLNYEe2o46d27a6tVQ))
	SEFFActionSequenceElement (ReturnCertificate, _3tukQLdLEe2lCsB2iGU-8w))	CallingSEFFActionSequenceElement / calling (CollectStatisticsFromEverywhere, _wqx4MLm3Ee2dIMSi7oNVYQ))
	CallingSEFFActionSequenceElement / returning (RetrieveCertificate, _PAn3gLkDEe25g4oGy4Jxtw))	SEFFActionSequenceElement (Beginning collectStatistics, _Qg5JwLNYEe2o46d27a6tVQ))
	CallingUserActionSequenceElement / returning (ScanCertificate, _oJMyYLZ6Ee2xLZp9hElpuA))	SEFFActionSequenceElement (ReturnStats, _AnZ-4Lm2Ee2dIMSi7oNVYQ))
		CallingSEFFActionSequenceElement / returning (CollectStatisticsFromEverywhere, _wqx4MLm3Ee2dIMSi7oNVYQ))
		SEFFActionSequenceElement (CalculateGeneralStatistics, _xtt74Lm3Ee2dIMSi7oNVYQ))
		SEFFActionSequenceElement (ReturnGeneralStatistics, _6VQZcLm3Ee2dIMSi7oNVYQ))
		CallingSEFFActionSequenceElement / returning (RetrieveStatistics, _v648ELmmEe2dIMSi7oNVYQ))
		SEFFActionSequenceElement (ReturnStatistics, _0dVMgLmmEe2dIMSi7oNVYQ))
		CallingSEFFActionSequenceElement / returning (RetrieveData, _wAzIkLmgEe2dIMSi7oNVYQ))
		SEFFActionSequenceElement (ReturnData, _w5jygLmgEe2dIMSi7oNVYQ))
		CallingSEFFActionSequenceElement / returning (DownloadStatistics, _C45C8Lj-Ee25g4oGy4Jxtw))
		SEFFActionSequenceElement (ReturnStatistics, _SPW2oLj-Ee25g4oGy4Jxtw))
		CallingUserActionSequenceElement / returning (UpdateStatistics, _b_q_ULZ7Ee2xLZp9hElpuA))
		CallingUserActionSequenceElement / calling (UpdateStatus, _jd7goLZ7Ee2xLZp9hElpuA))

Kontext:

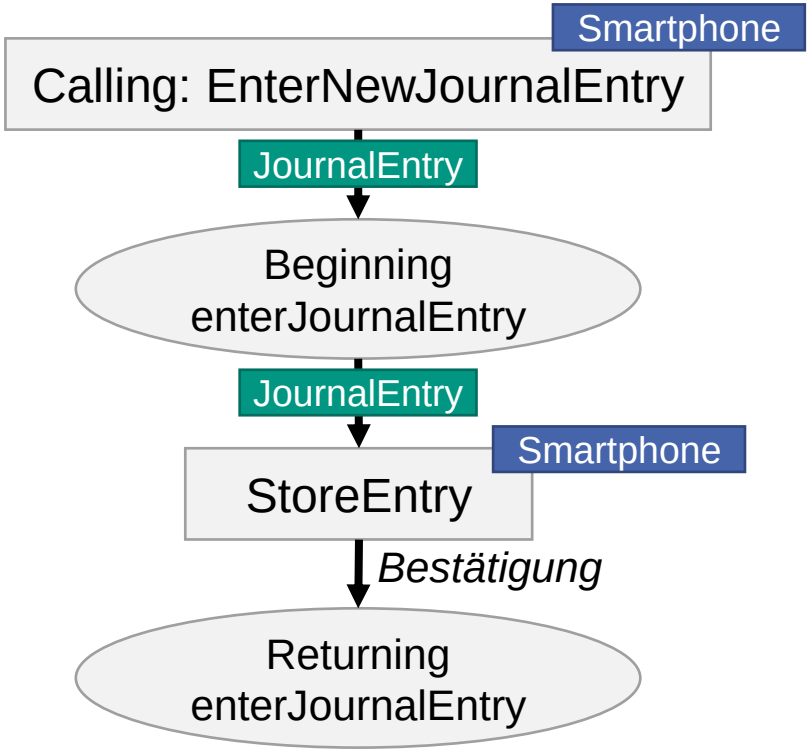
Corona Warn App: Label Propagation

ID: 4 *(User enters journal entry)*
CallingUserActionSequenceElement / calling (EnterNewJournalEntry, _UbqtQLZ5Ee2xLZp9hElpuA))
SEFFActionSequenceElement (Beginning enterJournalEntry, _ssANgLK8Ee2Y1pKtbIeM6Q))
SEFFActionSequenceElement (StoreEntry, _NPsfsMK3Ee28l_bdB_O6zg))
CallingUserActionSequenceElement / returning (EnterNewJournalEntry, _UbqtQLZ5Ee2xLZp9hElpuA))

Datenflussschranke:

JournalEntry ↛ ¬ Smartphone ✓

Kürzester Datenfluss: 4 Knoten
Längster Datenfluss: 78 Knoten ☺



Lernziele der Gastvorlesung

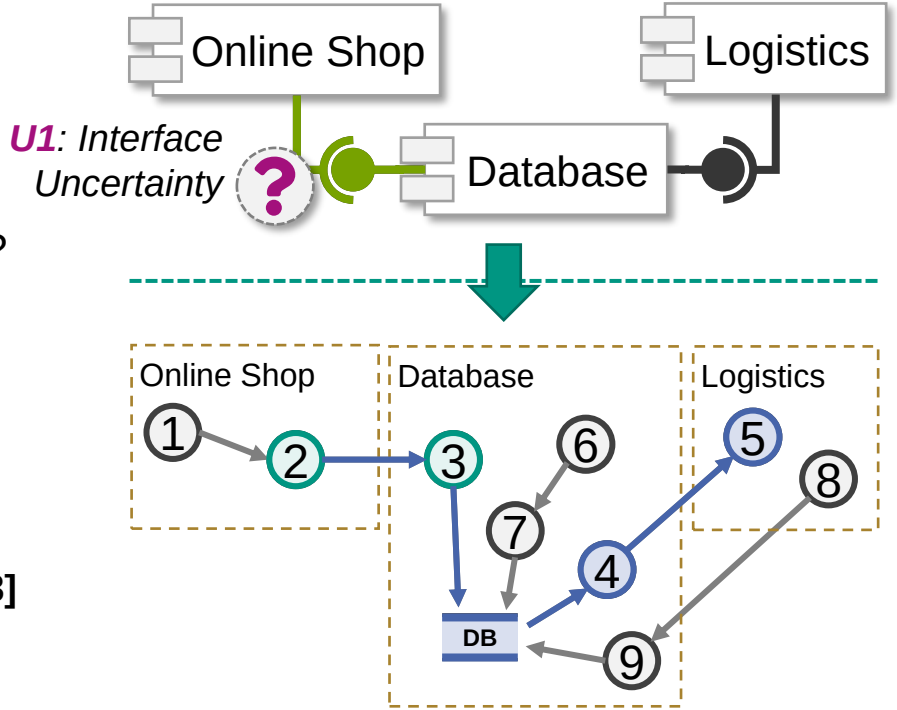
- Vertraulichkeit definieren, damit verbundene Herausforderungen nennen
- Einsatzmöglichkeiten von Datenflussdiagrammen und Bestandteile beschreiben und von anderen Diagrammtypen abgrenzen können
- Datenflussdiagramme zeichnen und anhand dieser die Vertraulichkeit eines Software-Systems bewerten können
- Die Technik “Label Propagation“ wiedergeben und an beispielhaften Datenflussdiagrammen anwenden können

⇨ Die Lernziele eignen sich trotzdem gut zur Prüfungsvorbereitung ☺

Ausblick: Aktuelle Forschung

- Herausforderung: Sowohl im Entwurf als auch dem Betrieb von Software-Systemen tritt Ungewissheit auf
  - Wie werden sich die Nutzer verhalten?
  - Rauscht der Sensor stärker als erwartet?
  - Ist der Anbieter vertrauenswürdig?
- Forschungsfrage: Welche Aussagen lassen sich noch über Vertraulichkeit unter Ungewissheit treffen?
- Ansatz: Propagation der Auswirkungen von Ungewissheit ähnlich zu Labels **[23]**

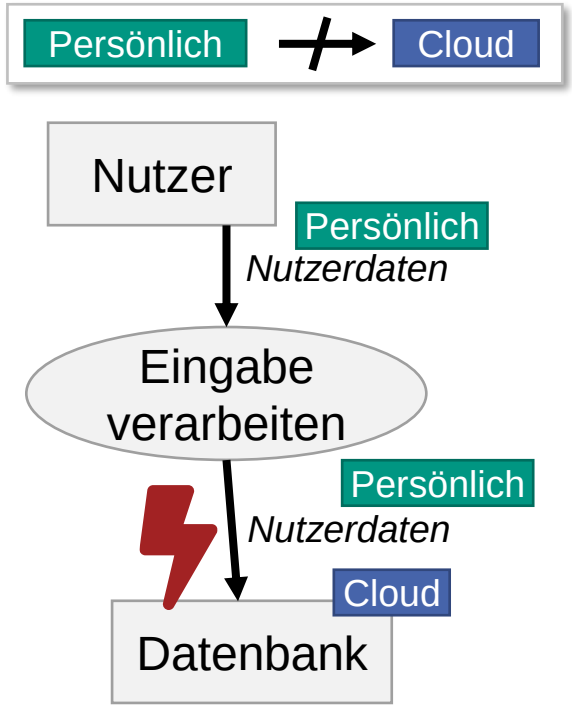
[23]



[23] S. Hahner, et al., “Architecture-based Uncertainty Impact Analysis to ensure Confidentiality”, In: *SEAMS*, IEEE/ACM, 2023. Accepted, to appear.

Zusammenfassung

- Vertraulichkeit behandelt die Preisgabe von Informationen an Unbefugte und kann mit Hilfe von Datenflussdiagrammen untersucht werden
- Datenflussdiagramme bestehen Datenflüssen, Prozessen, Dateien, Datenquellen und –senken
- Vertraulichkeitseigenschaften können via Daten- und Knotencharakteristiken annotiert werden
- Diese Charakteristiken werden auch zur Definition von Datenflussschranken verwendet
- Mit Hilfe der „Label Propagation“-Technik können auch große Datenflussdiagramme automatisch und effizient analysiert werden



Referenzen

[1] ISO, “ISO/IEC 27000:2018(E) Information technology – Security techniques – Information security management systems – Overview and vocabulary”, 2018.
[2] BSI, “IT-Grundschutz-Kompendium – Edition 2023”, online verfügbar unter: https://www.bsi.bund.de, 2023.
[3] ZEIT ONLINE, https://www.zeit.de/digital/datenschutz/2021-04/luca-app-sicherheitsluecken-datenschutz-kritik-corona (abgerufen: 09.11.21)
[4] CNBC, https://www.cnbc.com/2017/10/03/yahoo-every-single-account-3-billion-people-affected-in-2013-attack.html (abgerufen: 09.11.21)
[5] FORTUNE, https://fortune.com/2021/06/30/linkedin-data-theft-700-million-users-personal-information-cybersecurity (abgerufen: 09.11.21)
[6] HEISE, https://www.heise.de/news/Passwoerter-im-Klartext-20-000-Euro-Bussgeld-nach-DSGVO-gegen-Knuddels-de-4229798.html (abgerufen: 23.05.23)
[7] RNZ, https://www.rnz.de/politik/wirtschaft-regional_artikel,-Mannheim-Hackerangriff-auf-ABB-_arid,1112774.html (abgerufen: 23.05.23)
[8] Robert Koch-Institut (RKI), https://www.coronawarn.app/ (abgerufen 23.05.2023)
[9] SPIEGEL, https://www.zeit.de/gesundheit/2022-12/gesamtkosten-corona-warn-app-gesundheit-millionen (abgerufen: 23.05.2023)
[10] S. Hahner, Corona Warn App Case Study, online verfügbar unter: https://abunai.dev/, 2023.
[11] T. de Marco, „Structured Analysis and System Specification“, YOURDON inc., New York, 1978.
[12] R. Kramer, et al., „The combining DAG: a technique for parallel data flow analysis”, In: IEEE Transactions on Parallel and Distributed Systems, 1994.
[13] S. Seifermann, et al., “Detecting violations of access control and information flow policies in data flow diagrams”, In: JSS, vol. 184, 2022.
[14] A. Bambhore Tukaram, et al., “Towards a security benchmark for the architectural design of microservice applications”, In: ARES, ACM, 2022.
[15] S. Hahner et al., “Modeling Data Flow Constraints for Design-Time Confidentiality Analyses,” presented at ICSA, IEEE, 2021.
[16] A. Shostack, Threat Modeling: Designing for Security. John Wiley & Sons, 12, 2014.
[17] S. Seifermann, et al., “Data-Driven Software Architecture for Analyzing Confidentiality”, In: ICSA, IEEE, 2019.
[18] S. Seifermann, et al., “Identifying Confidentiality Violations in Architectural Design Using Palladio”, In: ECSA-C, 2021.
[19] S. Hahner, “Privacy by (Early) Design”, VKSI Sneak Preview, Invited Talk, 2021. Aufzeichung: https://www.youtube.com/watch?v=BiE8yaY8rb0&t=3987s
[20] K. Katkalov, “Ein modellgetriebener Ansatz zur Entwicklung informationsflusssicherer Systeme”, University of Augsburg, 2017.
[21] S. Seifermann, et al., “A Unified Model to Detect Information Flow and Access Control Violations in Software Architectures”, In: SECRYPT, 2021.
[22] R. H. Reussner, et al., “Modeling and Simulating Software Architectures: The Palladio Approach”, The MIT Press, 2016.
[23] S. Hahner, et al., “Architecture-based Uncertainty Impact Analysis to ensure Confidentiality”, In: SEAMS, IEEE/ACM, 2023. Accepted, to appear.
[24] Robert Koch-Institut (RKI), CWA Dokumentation, https://github.com/corona-warn-app/cwa-documentation (abgerufen 24.05.2023)