Karlsruhe Institute of Technology

KASTEL

SDQ Software Design and Quality

Data Flow Analysis

Dr.-Ing. Sebastian Hahner

Guest Lecture ∈ Software Security Engineering

Interactive Lecture

VECTOR

BUNTE NACHT DER DIGITALISIERUNG

Friday, June 19th, 2026 | 15:00 – 21:00

Vector Campus Karlsruhe
(Next Tram Stop: Hirtenweg/Technologiepark)

Free entry!

Scan the QR code to see the program! Registration optional.

HANDS-ON

Experience software innovation for automotive, medical, aerospace, and more.

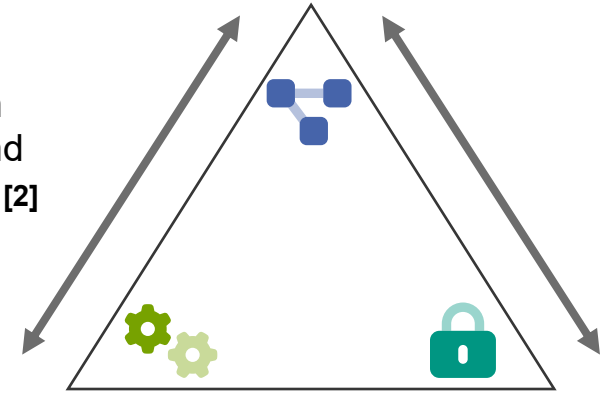
WORKSHOP

Bring your own ECG to life and discover how software, hardware, and testing work together.

TECH-TALKS

Software in safety-critical systems & AI put to the test.

Today's Lecture, Summarized



Software Architecture

"**fundamental concepts or properties of a system in its environment** embodied in its elements, relationships, and in the principles of its design" **[2]**

e.g., payment details, personal data, oral exam questions ☺

"property that **information is not made available** or disclosed to unauthorized individuals, entities, or processes" **[1]**

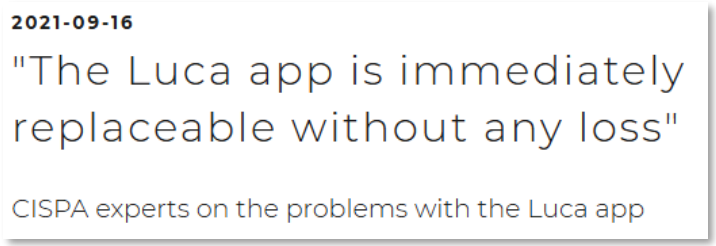
Confidentiality

A system analysis "**from the point of view of the data**" **[3]**

Data Flow Analysis

[1] ISO, "ISO/IEC 27000:2018(E) Information technology – Security techniques – Information security management systems – Overview and vocabulary", 2018.
[2] ISO, "ISO/IEC/IEEE 42010:2022 Software, systems and enterprise - Architecture description", 2022.
[3] T. de Marco, „Structured Analysis and System Specification", YOURDON inc., New York, 1978.

Data Breaches and Confidentiality

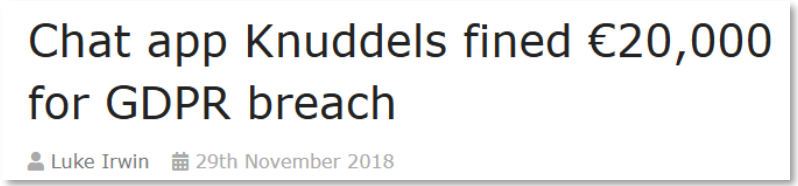


[4]

[5]

[6]

[7]

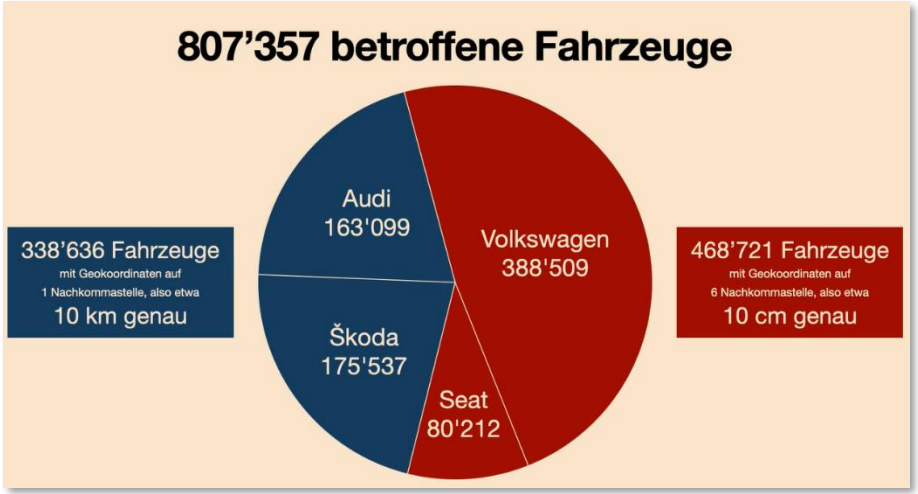


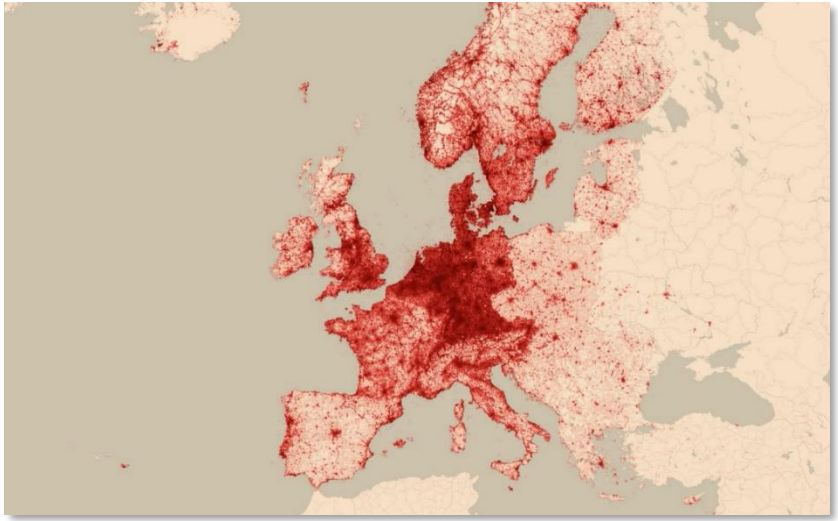
[8]

[4] CISPA, https://cispa.de/en/luca-app, 2021 (last checked: 2025-06-24)
[5] FORTUNE, https://fortune.com/2021/06/30/linkedin-data-theft-700-million-users-personal-information-cybersecurity, 2021 (last checked: 2025-06-24)
[6] Security Affairs, https://securityaffairs.com/163999/data-breach/ticketmaster-confirms-data-breach.html, 2024 (last checked: 2025-06-24)
[7] BBC, https://www.bbc.com/news/articles/c984zenjkz5o, 2024 (last checked: 2025-06-24)
[8] IT Governance, https://www.itgovernance.eu/blog/en/chat-app-knuddels-fined-e20000-for-gdpr-breach, 2018 (last checked: 2025-06-24)

Confidentiality and Software Architecture

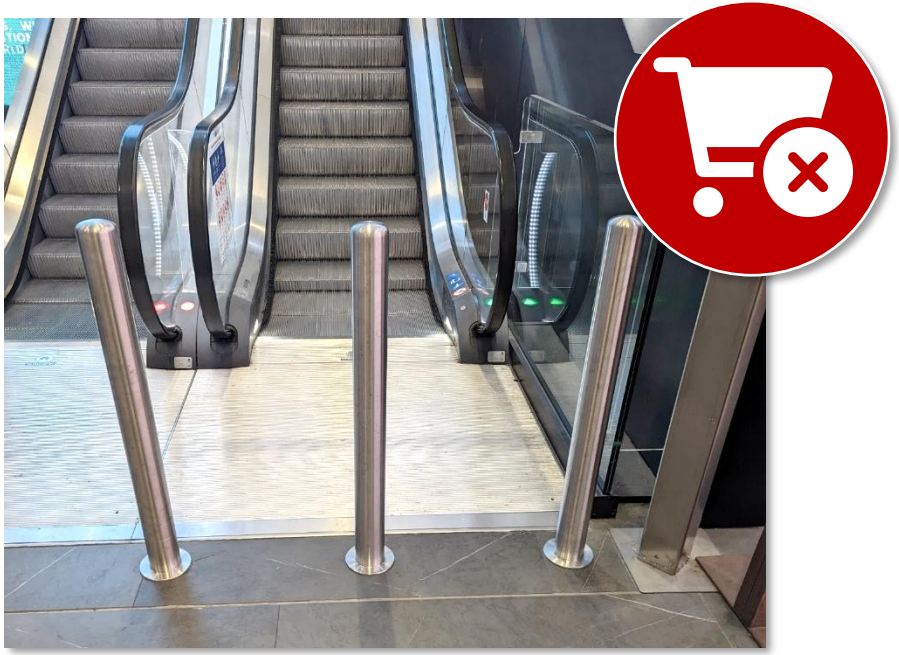
CARIAD Data Breach 2024, ccc: „*Volksdaten von Volkswagen*“ [9, 10]

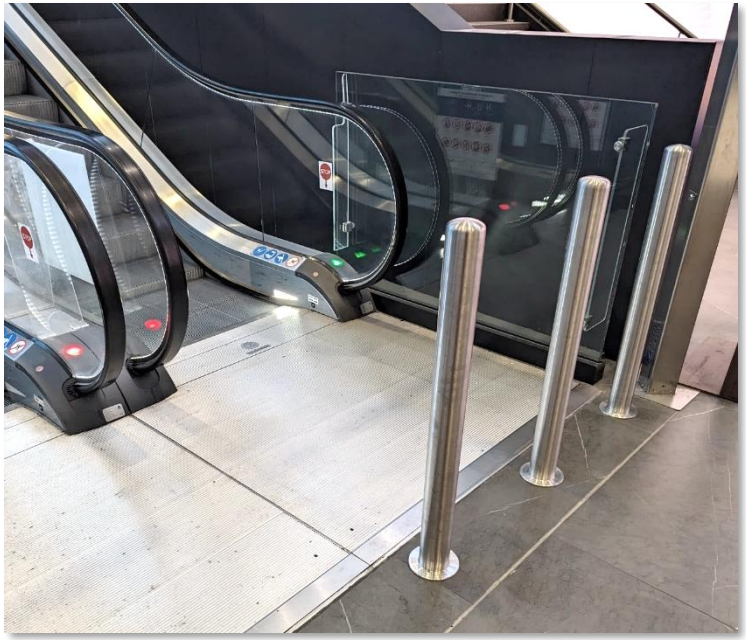




[9] CCC, https://media.ccc.de/v/38c3-wir-wissen-wo-dein-auto-steht-volksdaten-von-volkswagen, 2024 (last checked: 2025-06-24)
[10] Spiegel, https://www.spiegel.de/netzwelt/web/volkswagen-konzern-datenleck-wir-wissen-wo-dein-auto-steht-a-e12d33d0-97bc-493c-96d1-aa5892861027, 2024

Security as a Quality Attribute





Identifying Confidentiality Violations

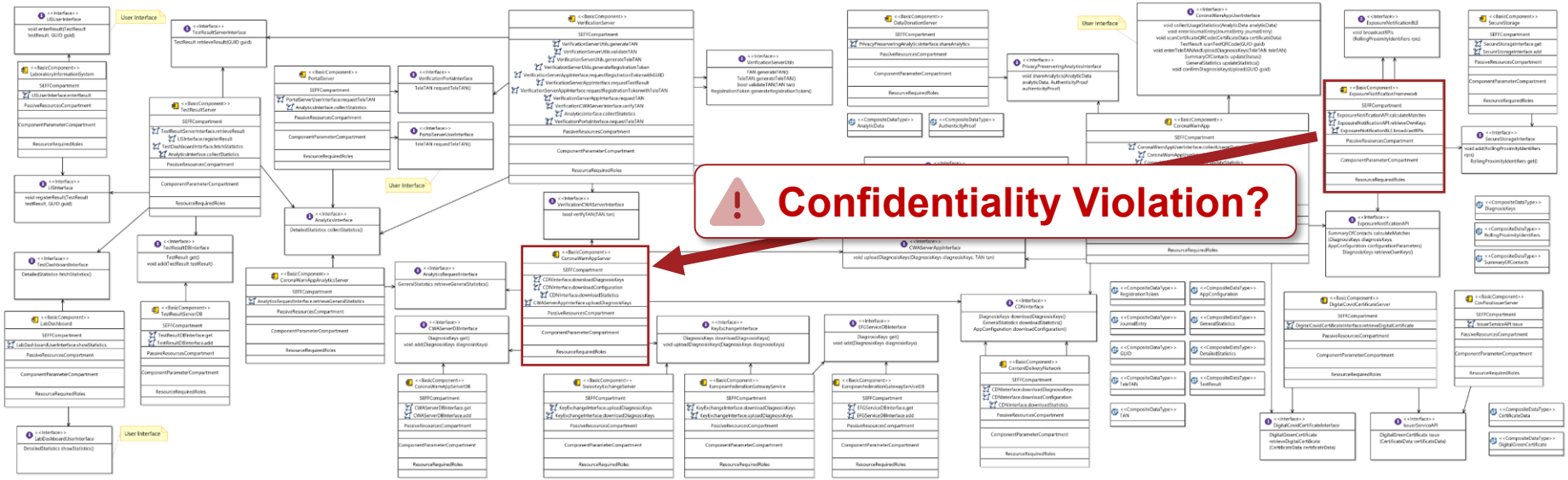
- German Showpiece: **Corona Warn App** [11]
  - Developed by SAP and Deutsche Telekom
  - Downloaded more than 48 Million times
  - Cost more than 220 Million Euro [12]
  - Open Source, comprehensively tested and analyzed by security experts
- Multiple kinds of sensitive data, e.g.
  - Data used for contact tracing
  - Covid-19 test results, vaccination certificates
  - Personal contact person diary

CORONA WARN-APP

[11]

[11] Robert Koch-Institut (RKI), https://github.com/corona-warn-app/, (last checked: 2025-06-24)
[12] SPIEGEL, https://www.zeit.de/gesundheit/2022-12/gesamtkosten-corona-warn-app-gesundheit-millionen (last checked: 2025-06-24)

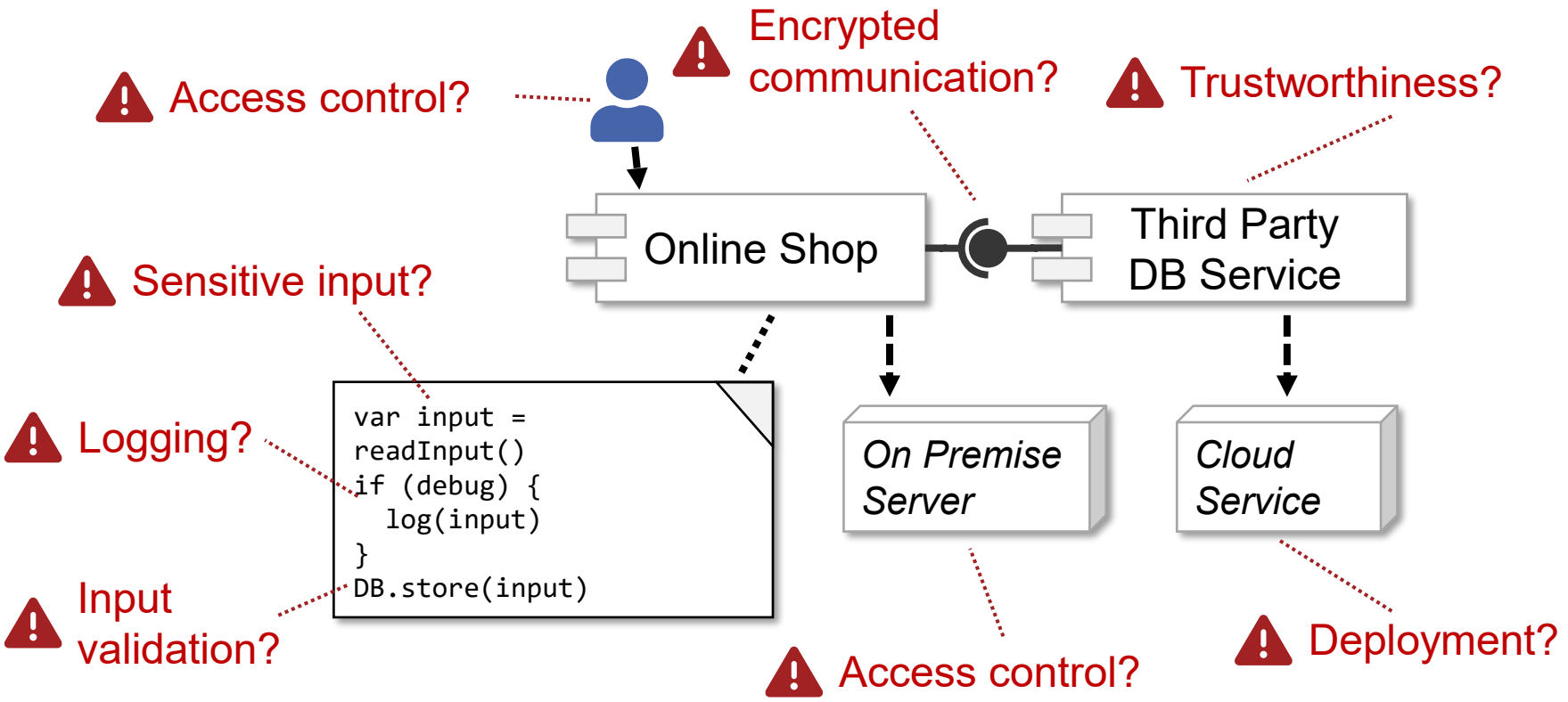
Identifying Confidentiality Violations



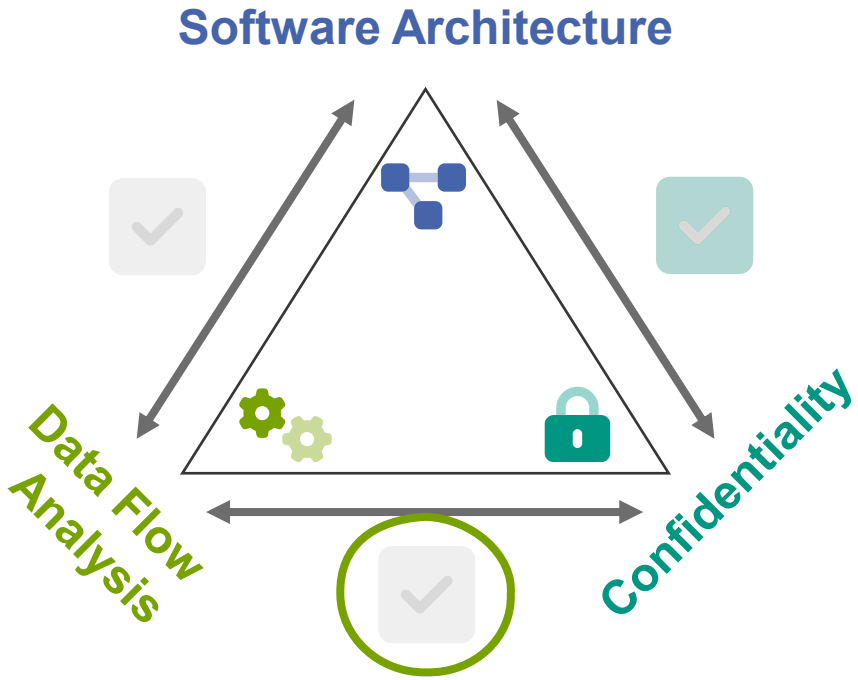
[13]

[13] S. Hahner, Corona Warn App Evaluation Scenario, online available: https://abunai.dev/, 2023, (last checked: 2025-06-24)

Let's Practice: What Could Go Wrong?



Lecture Overview

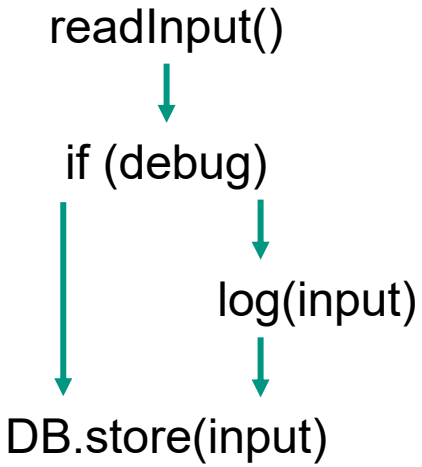


Learning Objectives

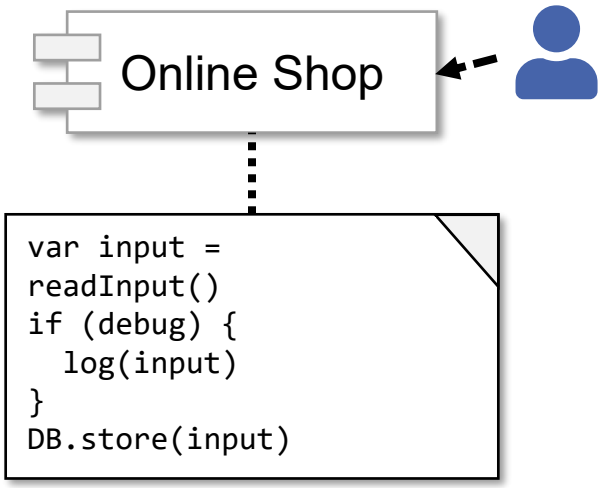
- Be able to **define** confidentiality and **name** related challenges
- Be able to **draw** data flow diagrams and to **use** them to assess a system's confidentiality
- Be able to **explain** how data flow analysis is used in architecture-based confidentiality analysis
- Be able to **explain** and **apply** the analysis method "label propagation"

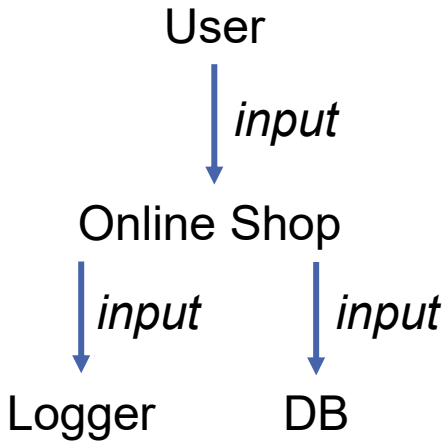
Data Flow vs. Control Flow, Implicit vs. Explicit

Control Flow Graph (CFG)



Data Flow Diagram (DFD)



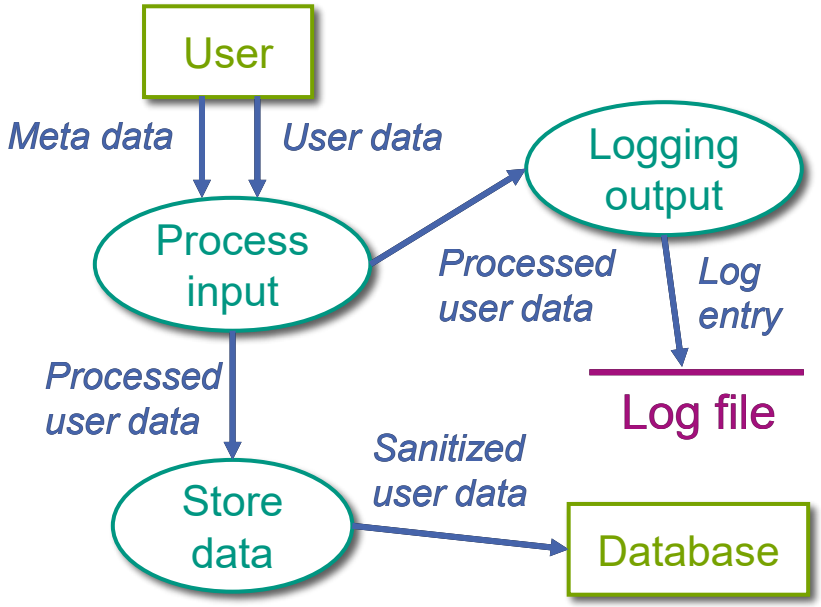


In this lecture: Focus on **explicit** data flows, not **implicit** information flows!

explicit: Data flowing to the database; **implicit**: *if(debug)* affecting the logging behavior

Data Flow Diagram Syntax (de Marco-style [3])

- Data flows represent relevant data types, depicted as named arrows
- Processes represent data processing and transformation, depicted as circles
- Files represent internal (temporary) data storage, depicted as lines
- Data sources and sinks represent users or other entities outside the modeled system's context, depicted as rectangles

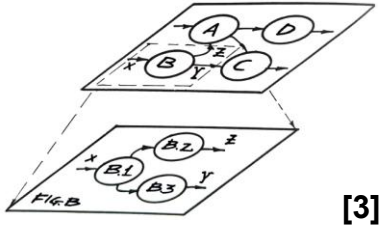


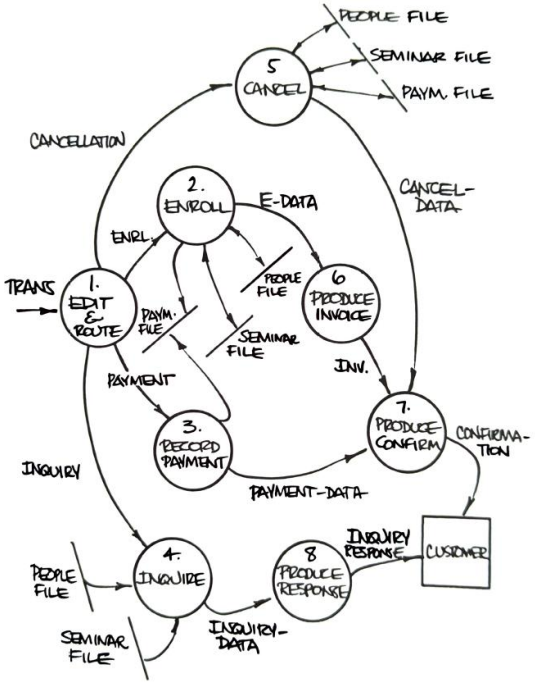
[3] T. de Marco, „Structured Analysis and System Specification", YOURDON inc., New York, 1978.

There is no “Correct”: Communication first!

- DFDs are meant for communication and analysis by engineers [3]
- In a recent empirical study, the analysis correctness significantly increased by 41% using DFDs [14]

- There is no “correct” abstraction level, it should be appropriate for communication and analysis
- DFDs can consist of multiple, cross-referencing layers, if required [3]





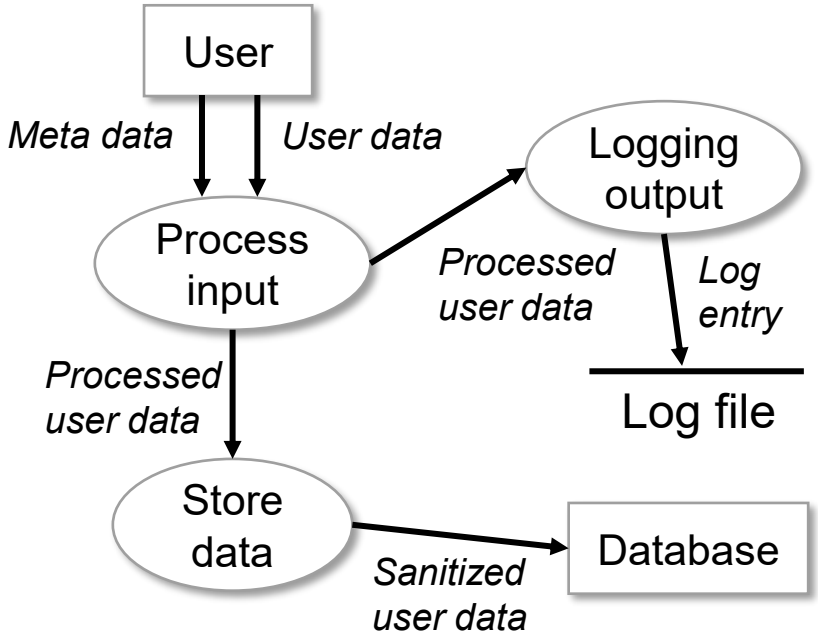
[3]

[3] T. de Marco, „Structured Analysis and System Specification“, YOURDON inc., New York, 1978.
[14] S. Schneider, et al., “How Dataflow Diagrams Impact Software Security Analysis: an Empirical Experiment”, SANER, IEEE, 2024.

Let's Practice: Drawing Data Flow Diagrams

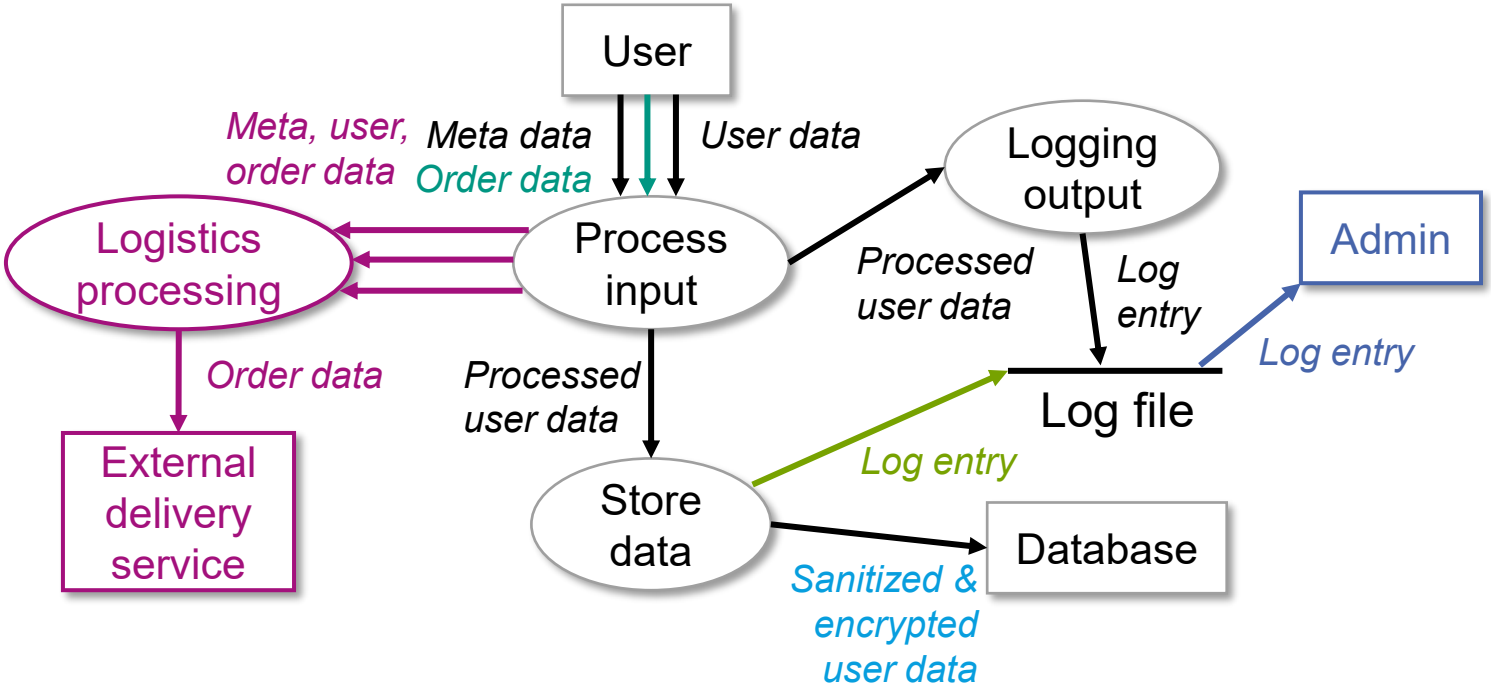
Task: Extend the diagram by…

- Users additionally enter order data
- Admins can read the log file
- The storing of data is also logged
- The system sends all information about the user to the logistics department for further processing. Afterwards, they forward the order data to an external delivery service.
- Data is encrypted before being stored



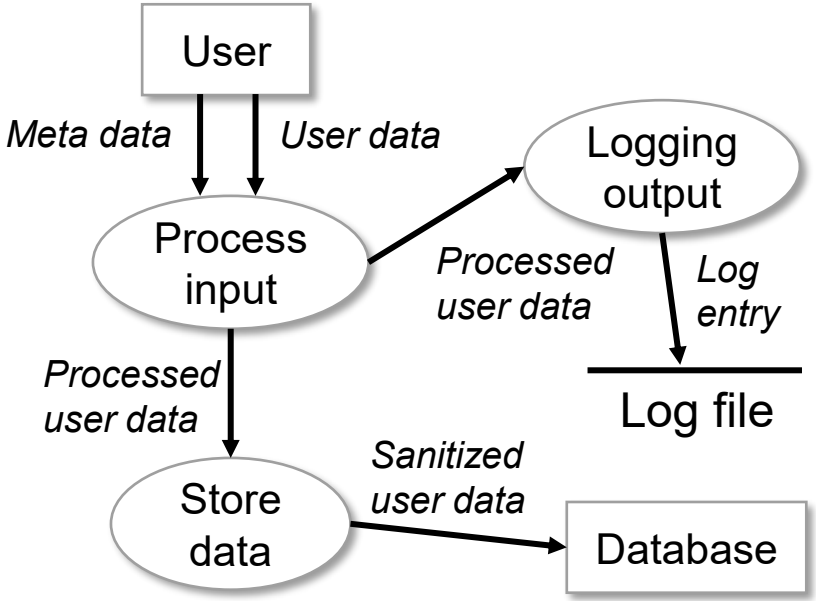
Think for a minute:

Let's Practice: Drawing Data Flow Diagrams



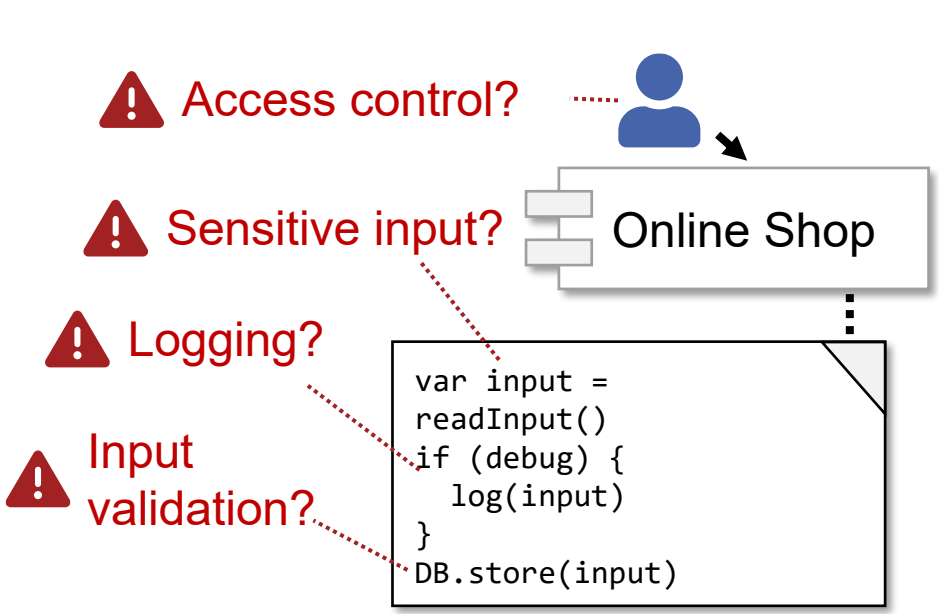
Formalizing Data Flow Diagrams

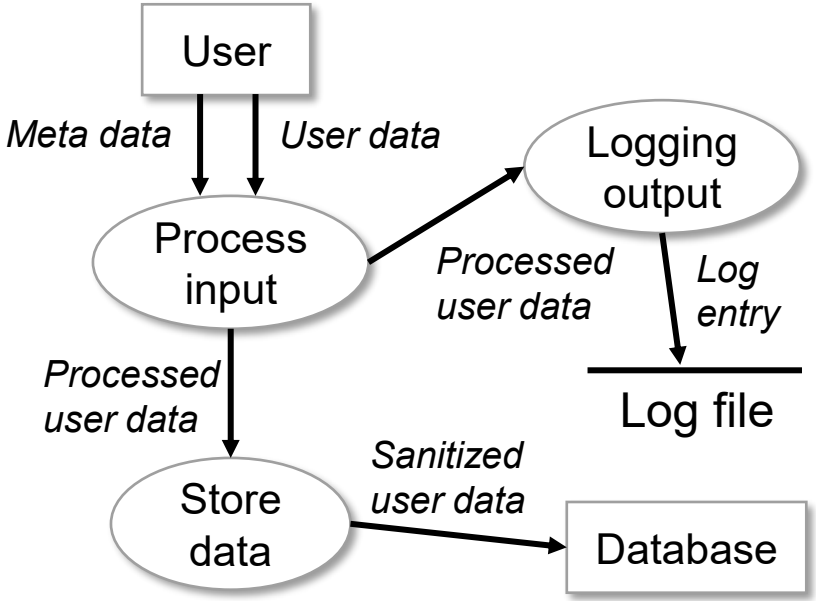
- Cycle-free DFDs can be constructed [15]
- They can be represented as **DAG**, a Directed Acyclic Graph G = (V, E), with vertices V and edges E
  - V = {“User”, “process input”, …}
  - E = {“User” → “process input”, …}
- Data flows can be represented as a strict partial ordering $u \prec v$
  - Irreflexive, which means $\neg (u \prec u)$
  - Asymmetric, so $u \prec v \Rightarrow \neg (v \prec u)$
  - Transitive, so $a \prec b \wedge b \prec c \Rightarrow a \prec c$



[15] B. Arp, et al., “Analyzing Cyclic Data Flow Diagrams Regarding Information Security”, SSP, GI, 2024.

Data Flow Diagrams and Confidentiality





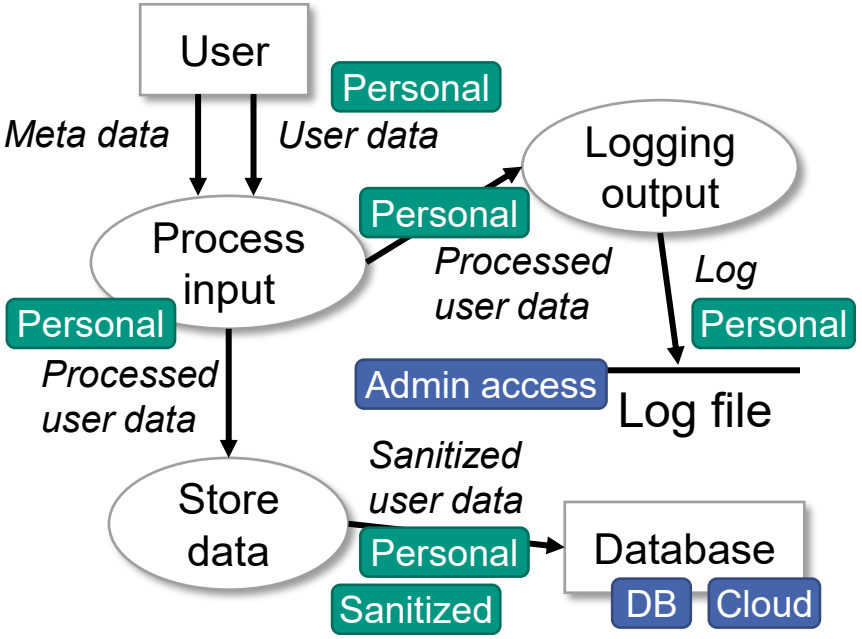
How to represent confidentiality in data flow diagrams?

Annotating Confidentiality Characteristics

When analyzing confidentiality, it is not the flowing data that is crucial, but its *characteristics* [16]

Annotating characteristics

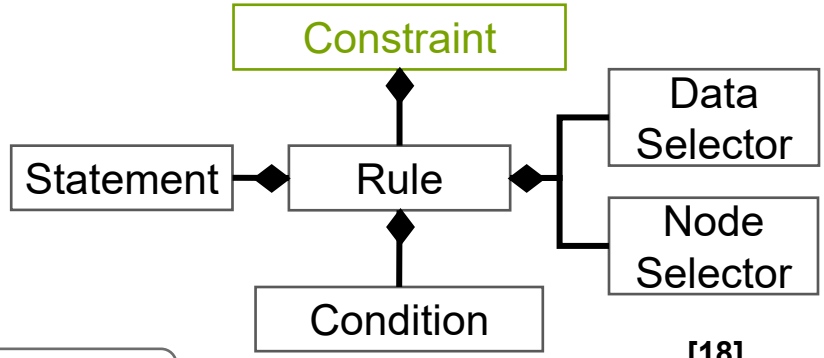
- Data labels describe properties of flowing data, e.g., the sensitivity of the data or its encryption status
- Node labels describe properties of the system elements represented by nodes, e.g., regarding their deployment or access control



[16] S. Seifermann, et al., “Detecting violations of access control and information flow policies in data flow diagrams”, JSS, Elsevier, 2022.

Specifying Confidentiality Requirements

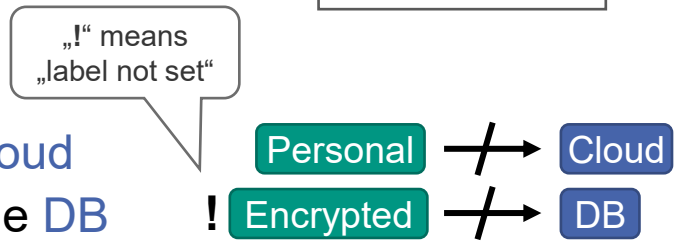
- Confidentiality requirements can be modeled by using patterns [17] or with data flow constraints [18]
- Constraints can express patterns [19]
- Constraints specify which data labels are forbidden to flow to which node labels



[18]

Examples

- Personal data shall *never flow* to the cloud
- Unencrypted data shall *never flow* to the DB

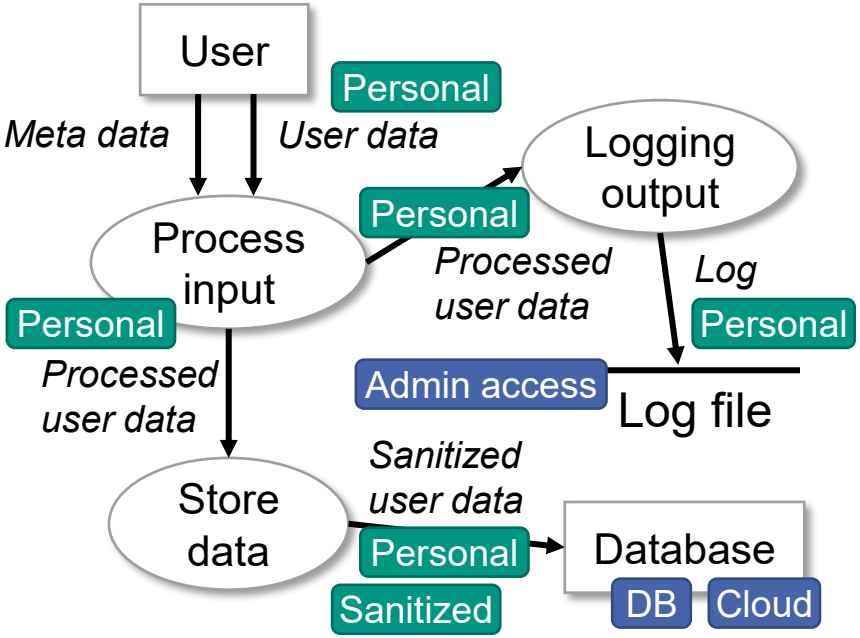


[17] A. Bambhore Tukaram, et al., “Towards a security benchmark for the architectural design of microservice applications”, ARES, ACM, 2022.
[18] S. Hahner et al., “Modeling Data Flow Constraints for Design-Time Confidentiality Analyses”, presented at ICSA, IEEE, 2021.
[19] N. Niehues, et al., “Integrating Security-Enriched Data Flow Diagrams Into Architecture-Based Confidentiality Analysis”, SSP, GI, 2024.

Let's Practice: Data Flow Constraints

Task: Does the system violate the following data flow constraints?

1. Personal ↛ Admin access ✗
2. Personal ↛ Cloud ✗
3. ! Sanitized ↛ DB ✓
4. Personal ∧ ! Encrypted ↛ Admin access ∨ Cloud ✗



Let's Practice: Putting Everything Together

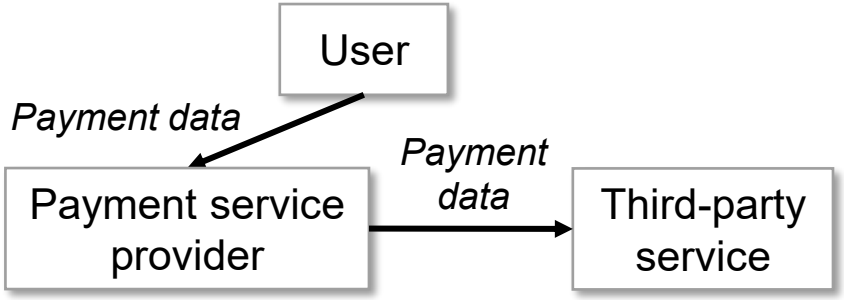
Task: Annotate data labels, node labels, and formulate constraints:

1. Payment data may only be shared with the payment service provider
2. User data must be stored in encrypted form on the database
3. Specific oral exam content may not be communicated to non-lecturers, e.g., students ☺

Think for 2 min:

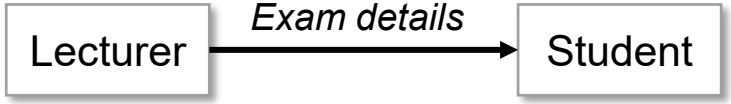
Discuss for 3 min:

1.

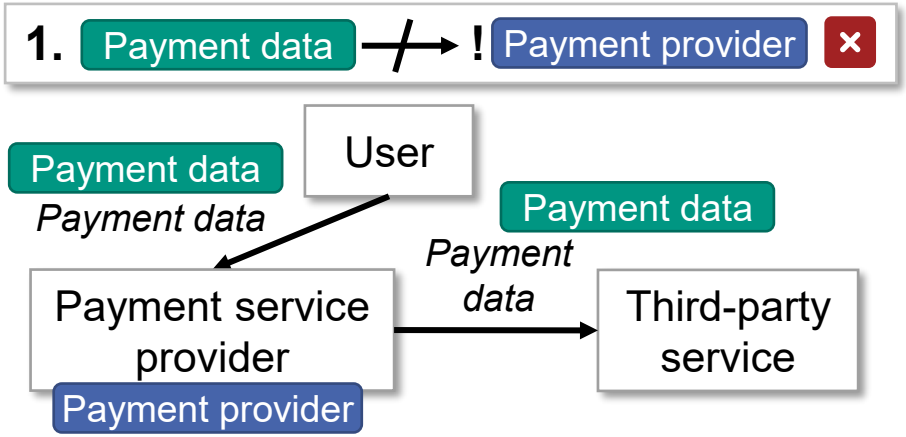


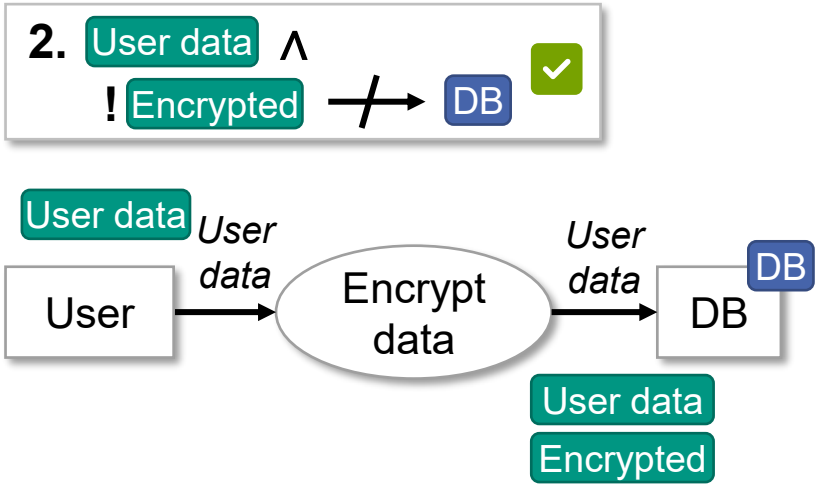
2.

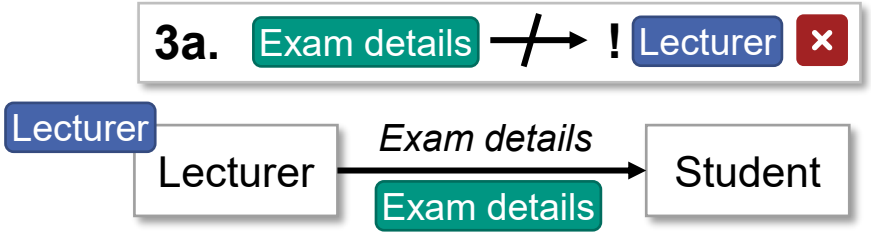
3.

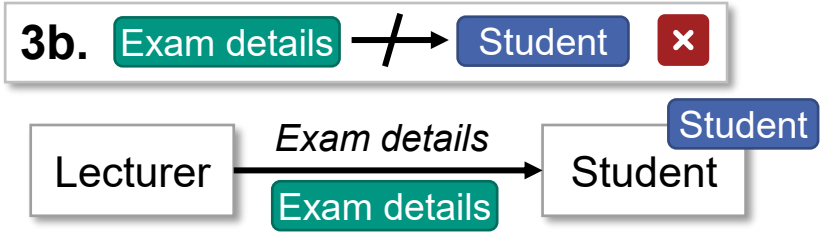


Let's Practice: Putting Everything Together

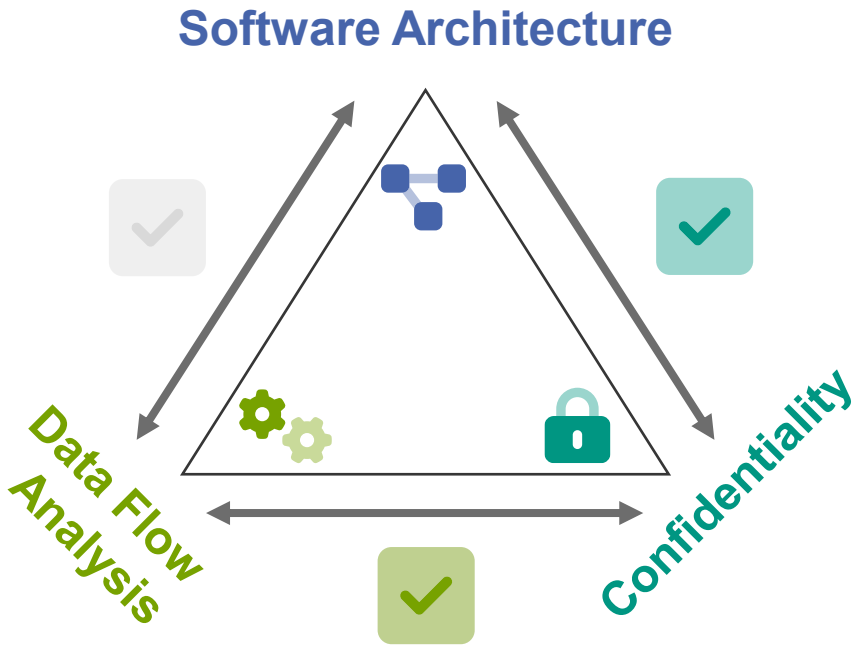








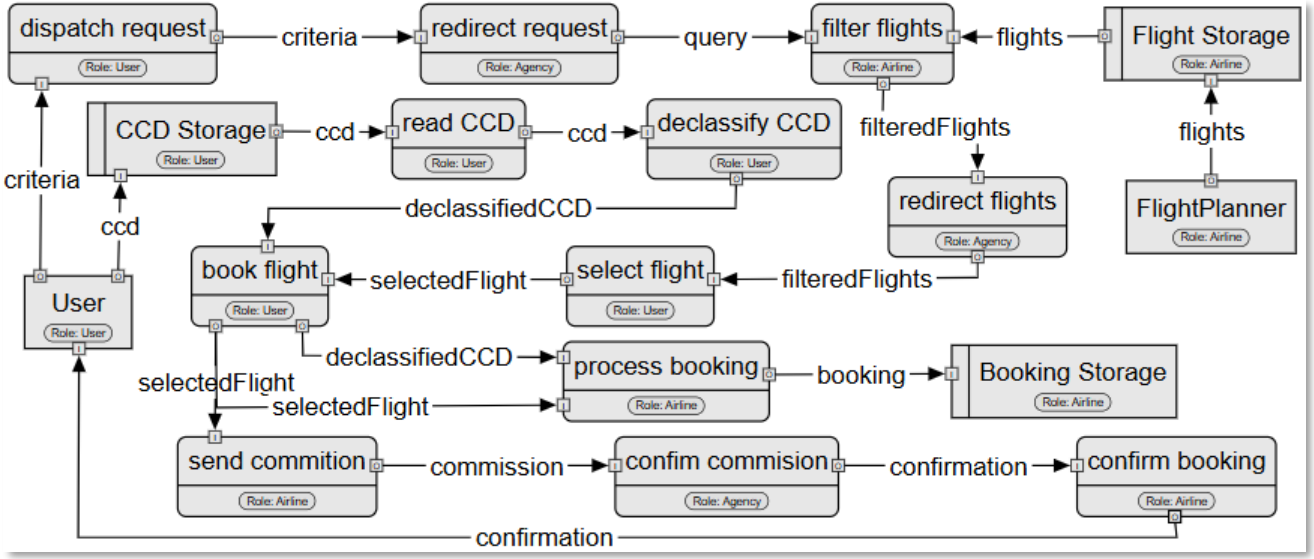
Lecture Overview



Learning Objectives

- Be able to **define** confidentiality and **name** related challenges
- Be able to **draw** data flow diagrams and to **use** them to assess a system's confidentiality
- Be able to **explain** how data flow analysis is used in architecture-based confidentiality analysis
- Be able to **explain** and **apply** the analysis method "label propagation"

Using Data Flow Diagrams in Analysis



How to get to those data flow diagrams?

How to automate the analysis in a scalable way?

Using Data Flow Diagrams in the Analysis

Creating Data Flow Diagrams

- Manual modeling ☺
- Extracting data flows from other modeling languages [16]
- Reverse engineering of data flow diagrams from source code [20]
- Code query-based approaches [21]
- AI-based diagram creation using natural language processing [22]

Data Flow Analysis Requirements [23, 24]

- Support for different input formats
- Tool support for manual modeling
- Precise yet expressive syntax with support for node labels and data labels
- Language for formulating constraints
- Efficient and scalable analysis algorithm
- Display violations for human inspection
- Extensibility for specialized analyses

[16] S. Seifermann, et al., “Detecting violations of access control and information flow policies in data flow diagrams”, JSS, Elsevier, 2022.
[20] S. Schneider and R. Scandariato, “Automatic extraction of security-rich dataflow diagrams for microservice applications written in Java”, JSS, Elsevier, 2023.
[21] GitHub, “CodeQL: Discover vulnerabilities across a codebase with CodeQL”, https://codeql.github.com/ (last checked: 2025-06-27)
[22] G. B. Herwanto, et al., “From User Stories to Data Flow Diagrams for Privacy Awareness: A Research Preview”, REFSQ, Springer, 2022.
[23] L. Sion, et al., “Security Threat Modeling: Are Data Flow Diagrams Enough?”, ICSEW, ACM, 2020.
[24] N. Boltz and S. Hahner, et al., “An Extensible Framework for Architecture-Based Data Flow Analysis for Information Security”, ECSA, 2024.

Focus for the Remainder of Today's Lecture

Creating Data Flow Diagrams

- Manual modeling ☺
- Extracting data flows from other modeling languages [16]
- Reverse engineering of data flow diagrams from source code [20]
- Code query-based approaches [21]
- AI-based diagram creation using natural language processing [22]

Data Flow Analysis Requirements [23, 24]

- Support for different input formats
- Tool support for manual modeling
- Precise yet expressive syntax with support for node labels and data labels
- Language for formulating constraints
- Efficient and scalable analysis algorithm
- Display violations for human inspection
- Extensibility for specialized analyses

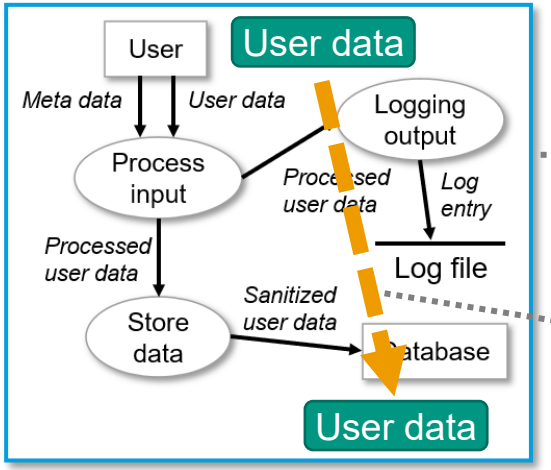
[16] S. Seifermann, et al., "Detecting violations of access control and information flow policies in data flow diagrams", JSS, Elsevier, 2022.
[20] S. Schneider and R. Scandariato, "Automatic extraction of security-rich dataflow diagrams for microservice applications written in Java", JSS, Elsevier, 2023.
[21] GitHub, "CodeQL: Discover vulnerabilities across a codebase with CodeQL", https://codeql.github.com/ (last checked: 2025-06-27)
[22] G. B. Herwanto, et al., "From User Stories to Data Flow Diagrams for Privacy Awareness: A Research Preview", REFSQ, Springer, 2022.
[23] L. Sion, et al., "Security Threat Modeling: Are Data Flow Diagrams Enough?", ICSEW, ACM, 2020.
[24] N. Boltz and S. Hahner, et al., "An Extensible Framework for Architecture-Based Data Flow Analysis for Information Security", ECSA, 2024.

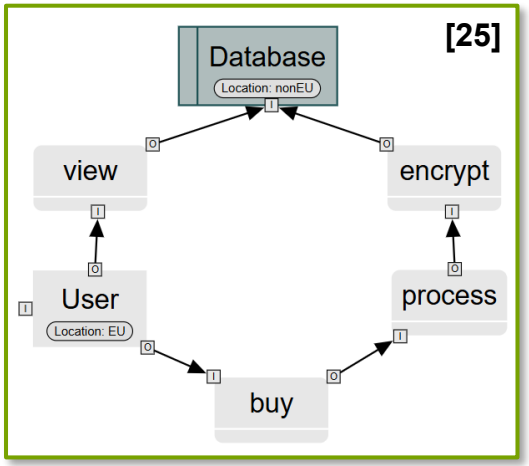
xDECAF – Data Flow Analysis Framework [24, 25]

- State-of-the-art data flow diagram analysis, initially proposed 2016 **[26]**
- Released as open-source framework in 2024 **[24, 25]**
- Under active research and development at KIT
- Fulfills all the requirements on the previous slide ☺

[24] N. Boltz and S. Hahner, et al., “An Extensible Framework for Architecture-Based Data Flow Analysis for Information Security”, ECSA, 2024.
[25] xDECAF, more information: https://dataflowanalysis.org, online editor available under: https://editor.dataflowanalysis.org (last checked: 2025-06-28)
[26] S. Seifermann, “Architectural Data Flow Analysis”, WICSA, IEEE, 2016.

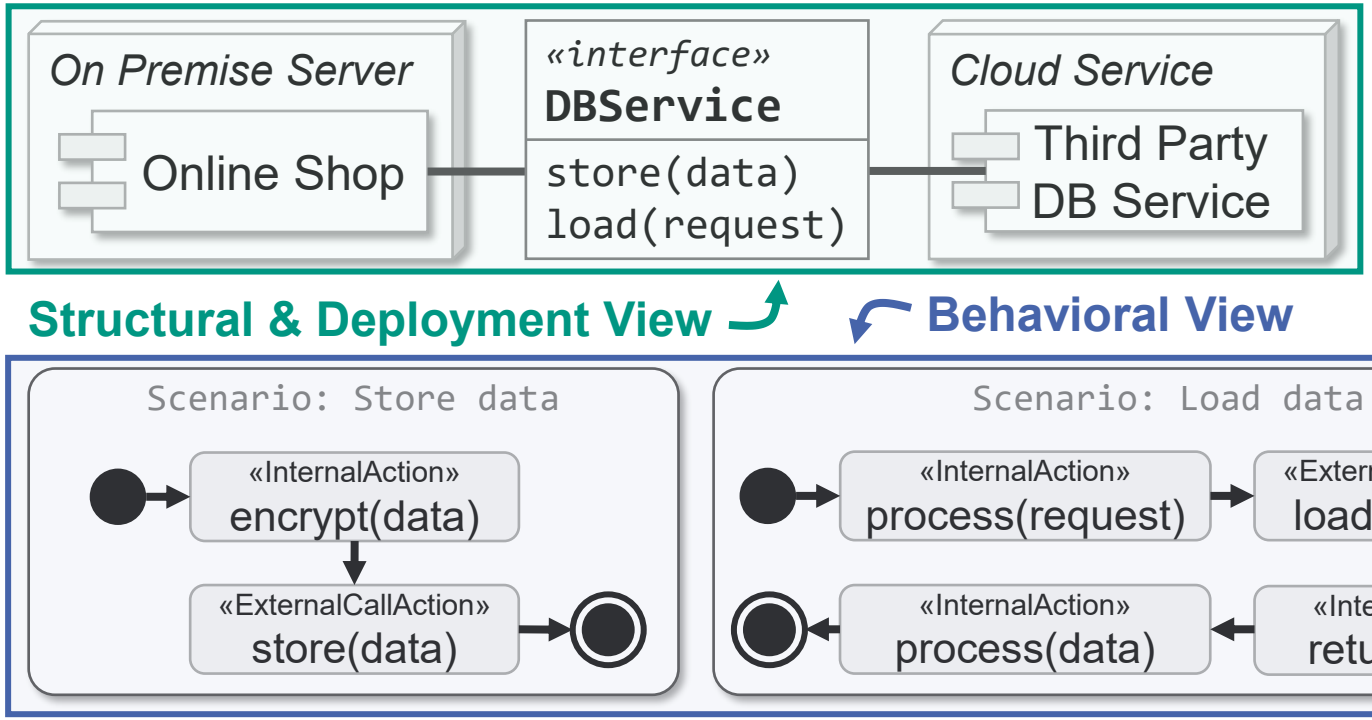
xDECAF – Data Flow Analysis Workflow [24]





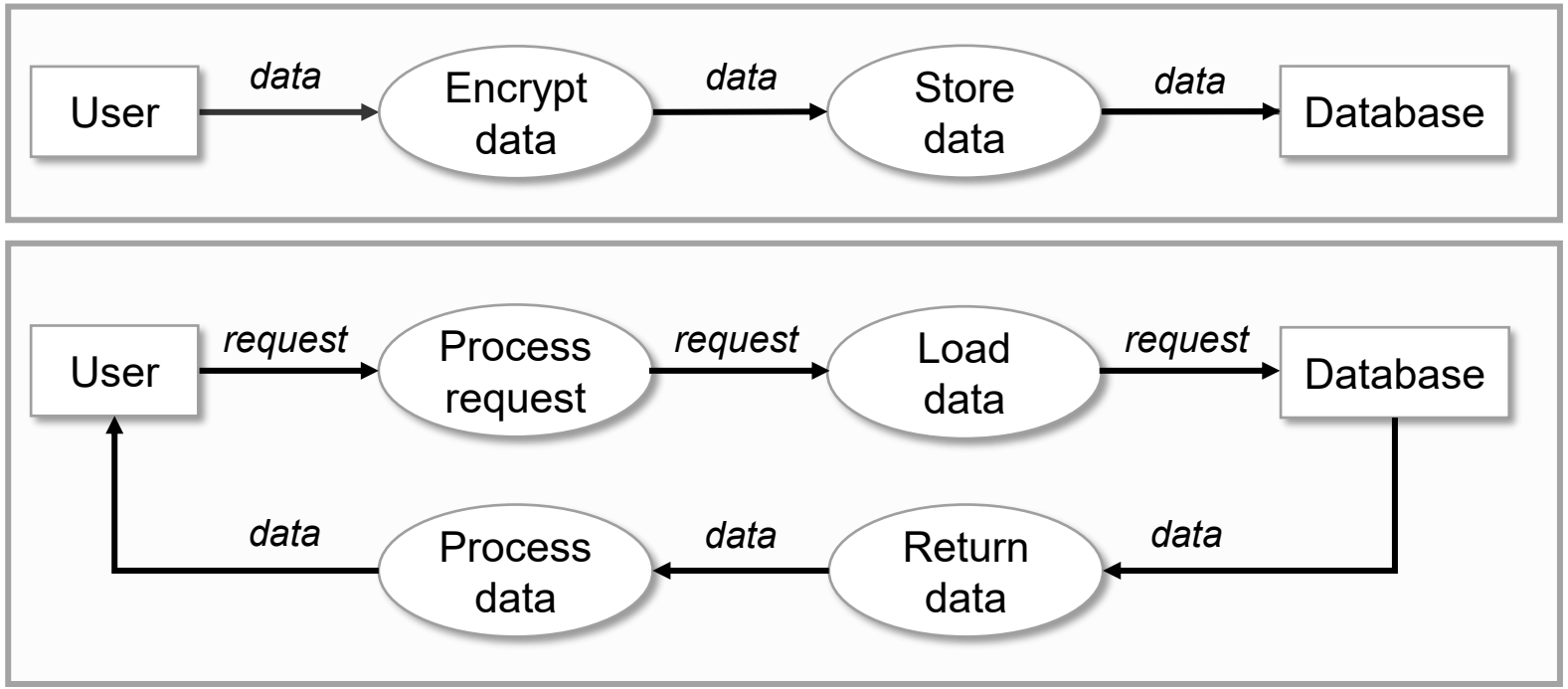
[24] N. Boltz and S. Hahner, et al., “An Extensible Framework for Architecture-Based Data Flow Analysis for Information Security”, ECSA, 2024.
[25] xDECAF, more information: https://dataflowanalysis.org, online editor available under: https://editor.dataflowanalysis.org (last checked: 2025-06-28)

Flow Graph Extraction: Architectural Models [24, 27]



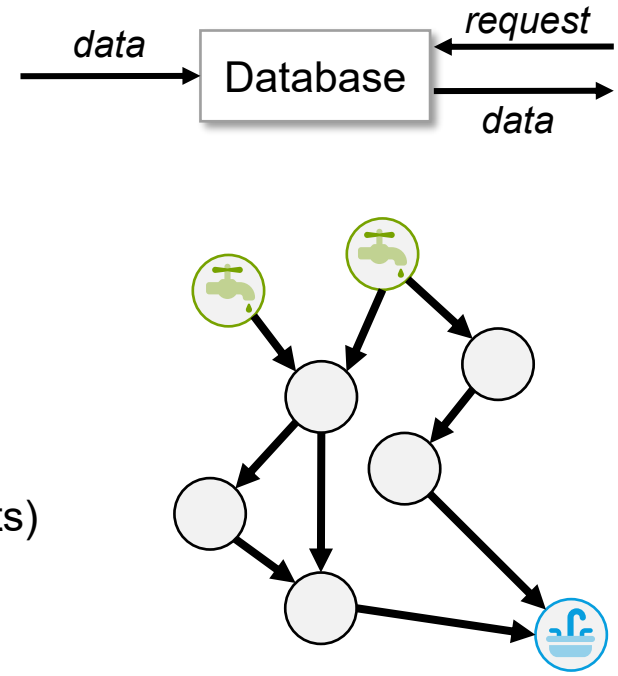
[24] N. Boltz and S. Hahner, et al., “An Extensible Framework for Architecture-Based Data Flow Analysis for Information Security”, ECSA, 2024.
[27] R. H. Reussner et al., “Modeling and Simulating Software Architectures: The Palladio Approach.”, The MIT Press, 2016.

One Data Flow Diagram vs. Multiple Flow Graphs



Transpose Flow Graphs (TFGs) [24]

- **Idea**: Independent data flows can be analyzed independently in parallel
- An outgoing data flow from a node is independent *iff* it is not in any relation with any incoming flows
- **Result**: Transpose Flow Graphs (TFGs) represent partial data flow diagrams with **n** sources but only **1** sink
  - TFGs represent DAGs! (with all the benefits)
  - They are simpler to construct, simpler to reason about, and simpler to analyze **[28]**

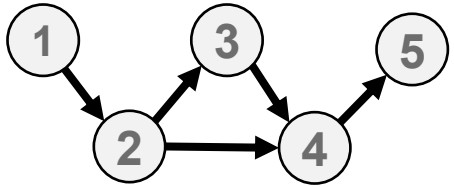


[24] N. Boltz and S. Hahner, et al., “An Extensible Framework for Architecture-Based Data Flow Analysis for Information Security”, ECSA, 2024.
[28] S. Hahner, “Architecture-Based and Uncertainty-Aware Confidentiality Analysis”, Dissertation, Karlsruhe Institute of Technology (KIT), Karlsruhe, 2024.

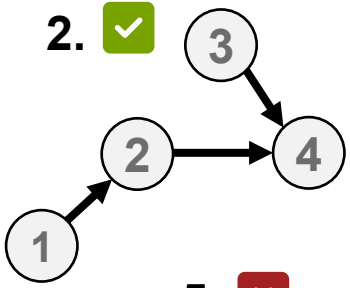
Let's Practice: Transpose Flow Graphs

Task: Does each graph represent a single TFG?

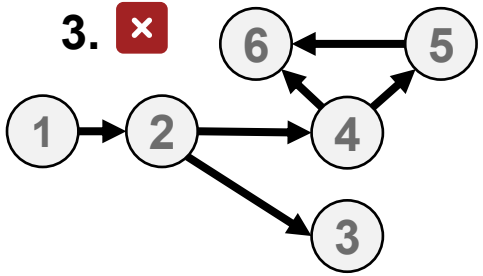
1. ✓



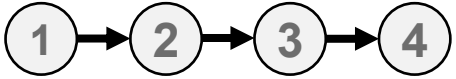
2. ✓



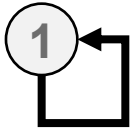
3. ✗



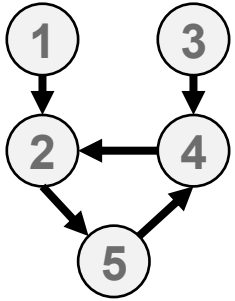
4. ✓



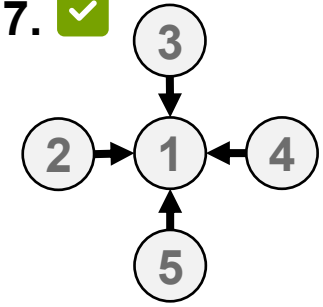
5. ✗



6. ✗

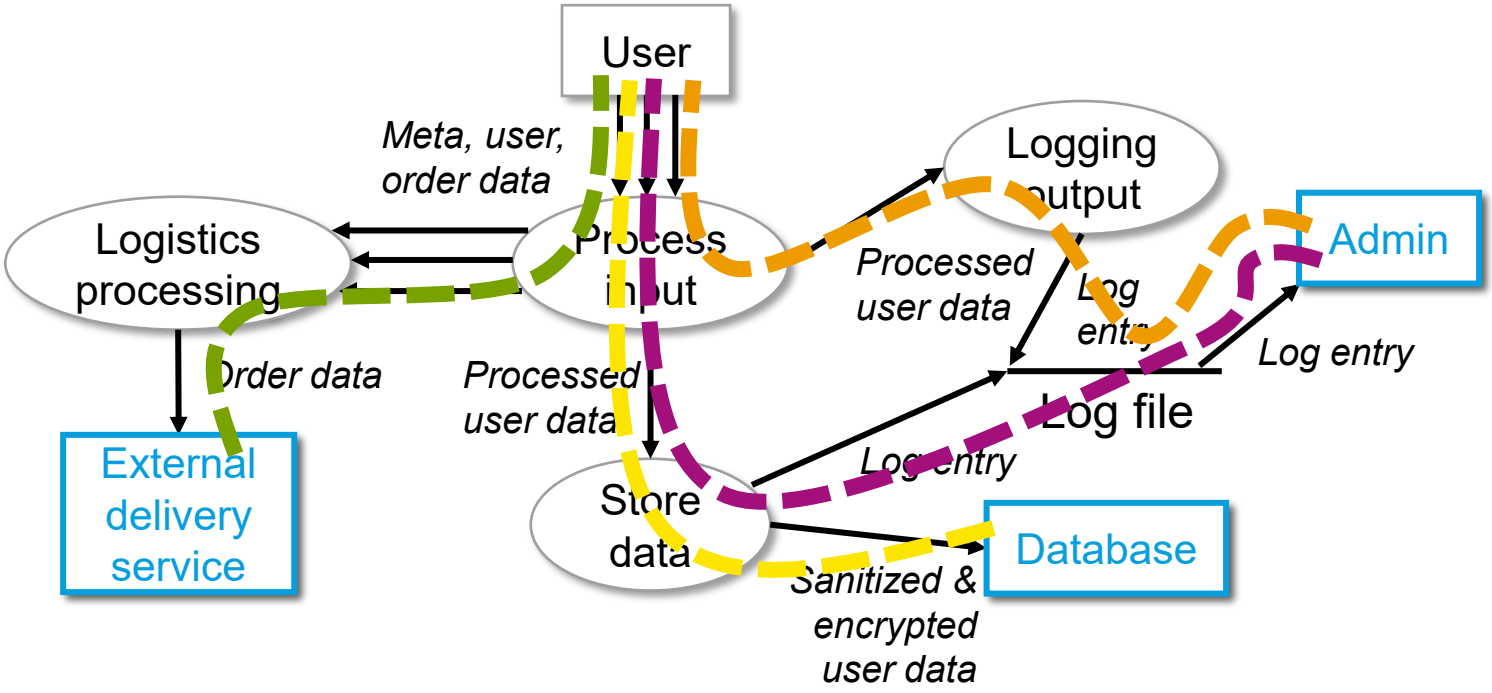


7. ✓

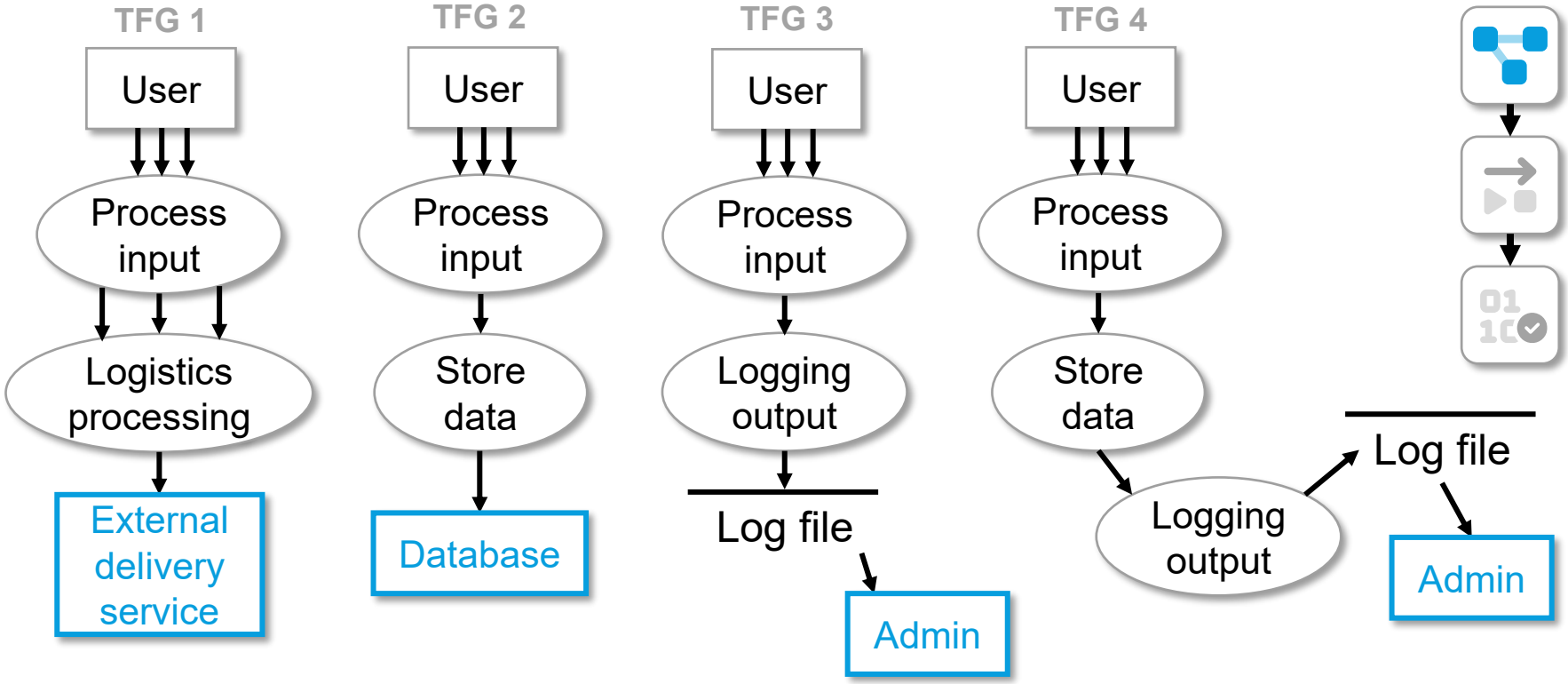


Think for a minute:

Flow Graph Extraction: Data Flow Diagrams [24]

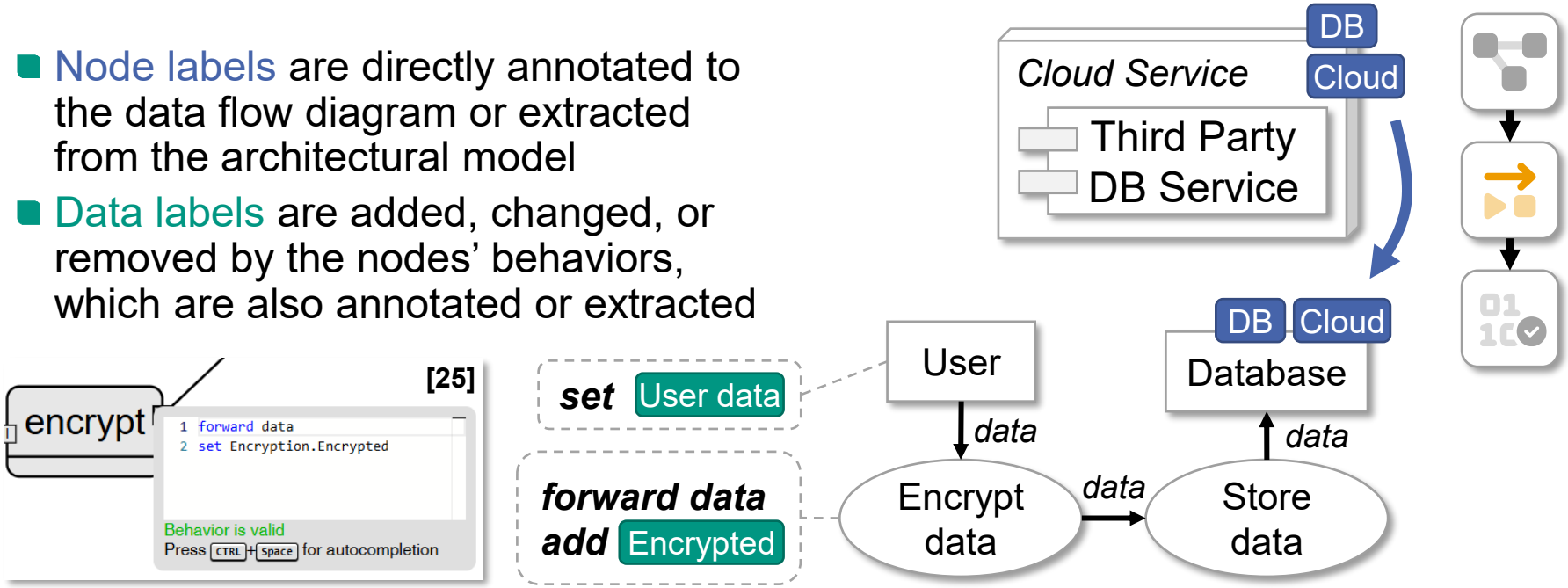


Flow Graph Extraction: Data Flow Diagrams [24]



Label Extraction and Label Propagation [16, 24]

- Node labels are directly annotated to the data flow diagram or extracted from the architectural model
- Data labels are added, changed, or removed by the nodes' behaviors, which are also annotated or extracted

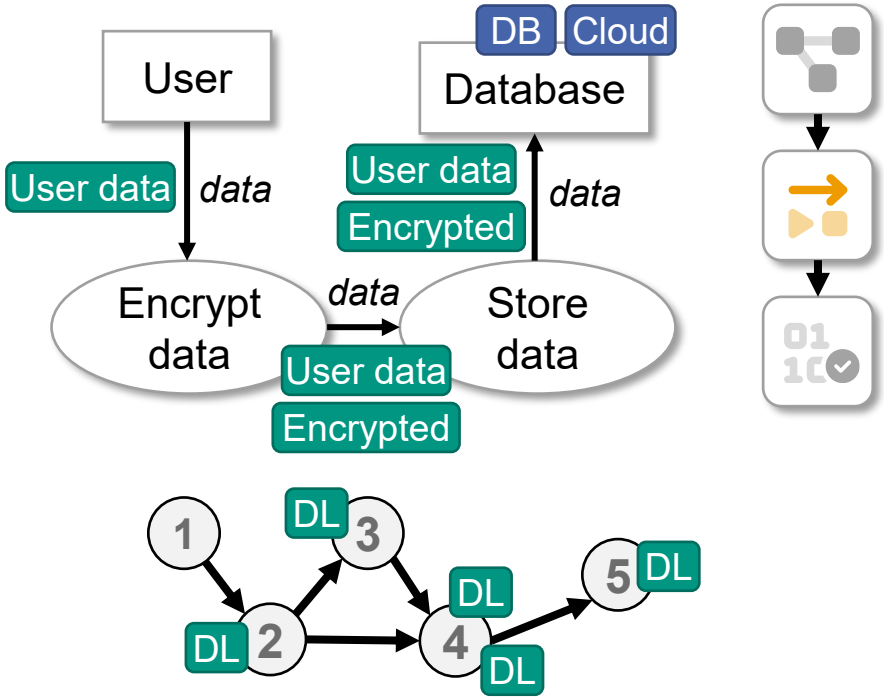


[16] S. Seifermann, et al., "Detecting violations of access control and information flow policies in data flow diagrams", JSS, Elsevier, 2022.
[24] N. Boltz and S. Hahner, et al., "An Extensible Framework for Architecture-Based Data Flow Analysis for Information Security", ECSA, 2024.
[25] xDECAF, more information: https://dataflowanalysis.org, online editor available under: https://editor.dataflowanalysis.org (last checked: 2025-06-28)

Label Propagation: Algorithm [16, 24]

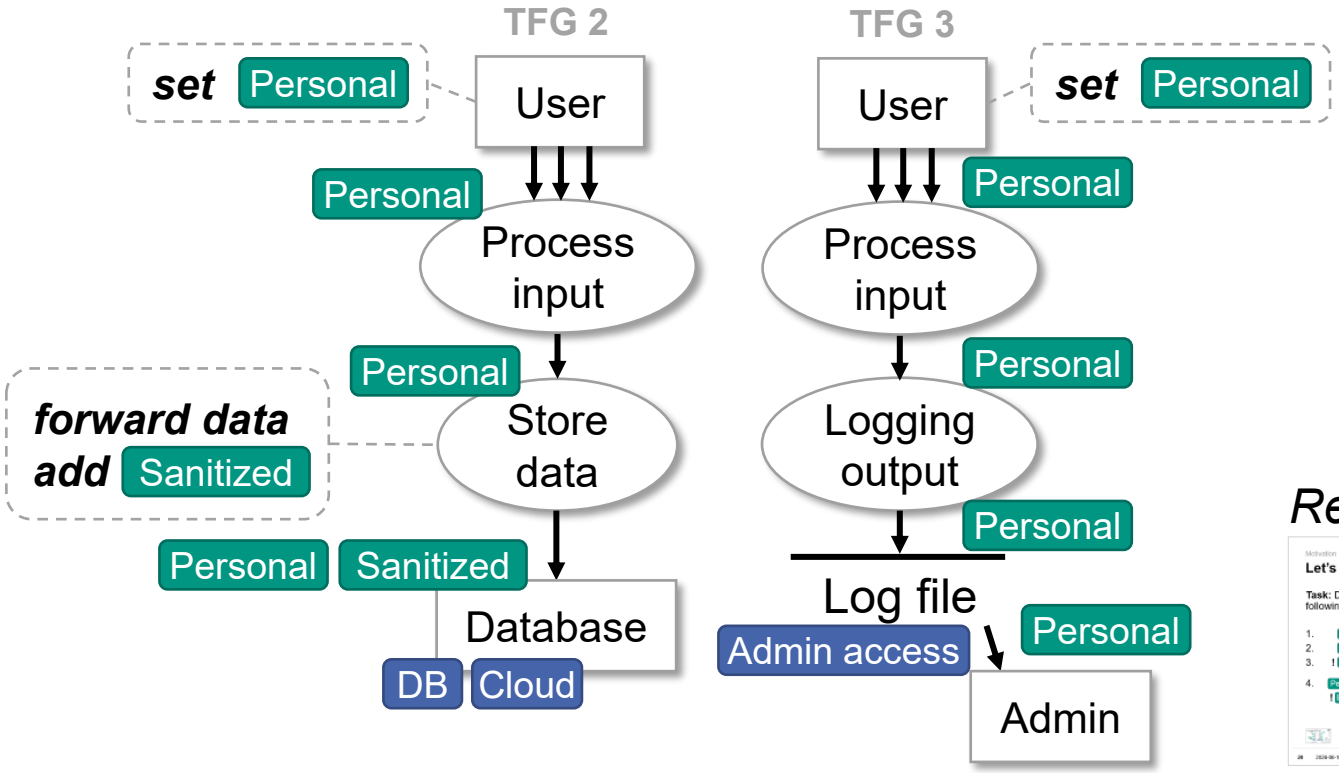
- "Manual labeling is repetitive and […] it is error prone" [16]
- Label propagation requires "a limited set of initial labels that are propagated through the system" [16]

- **Put simply:** For each node, apply the node behavior based on the incoming labels to calculate outgoing labels [24]
- In the following, we assume *forward behavior* if nothing else is stated

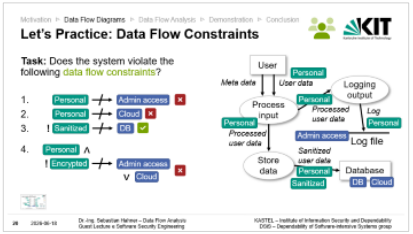


[16] S. Seifermann, et al., "Detecting violations of access control and information flow policies in data flow diagrams", JSS, Elsevier, 2022.
[24] N. Boltz and S. Hahner, et al., "An Extensible Framework for Architecture-Based Data Flow Analysis for Information Security", ECSA, 2024.

Label Propagation: More Examples



Remember?



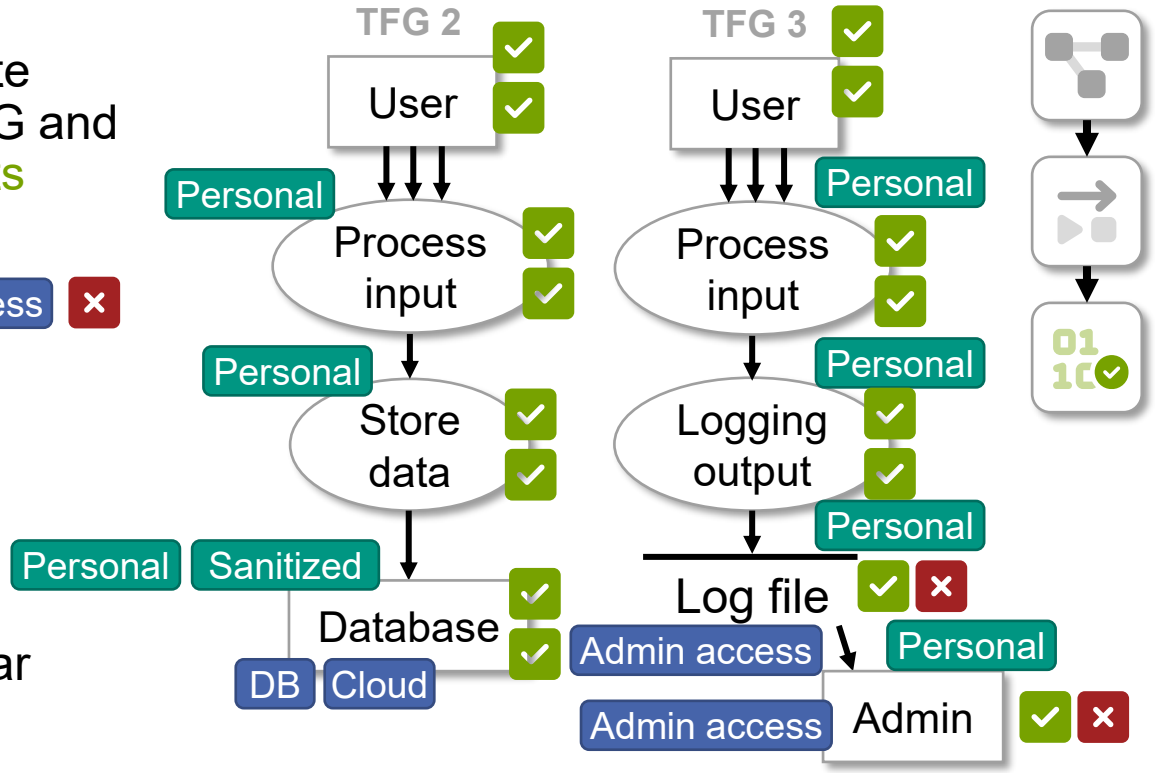
Automated Constraint Checking

After label propagation, iterate over *every* node of *every* TFG and check *all* data flow constraints

1. Personal → Admin access ✗
3. ! Sanitized → DB ✓

Analysis Runtime

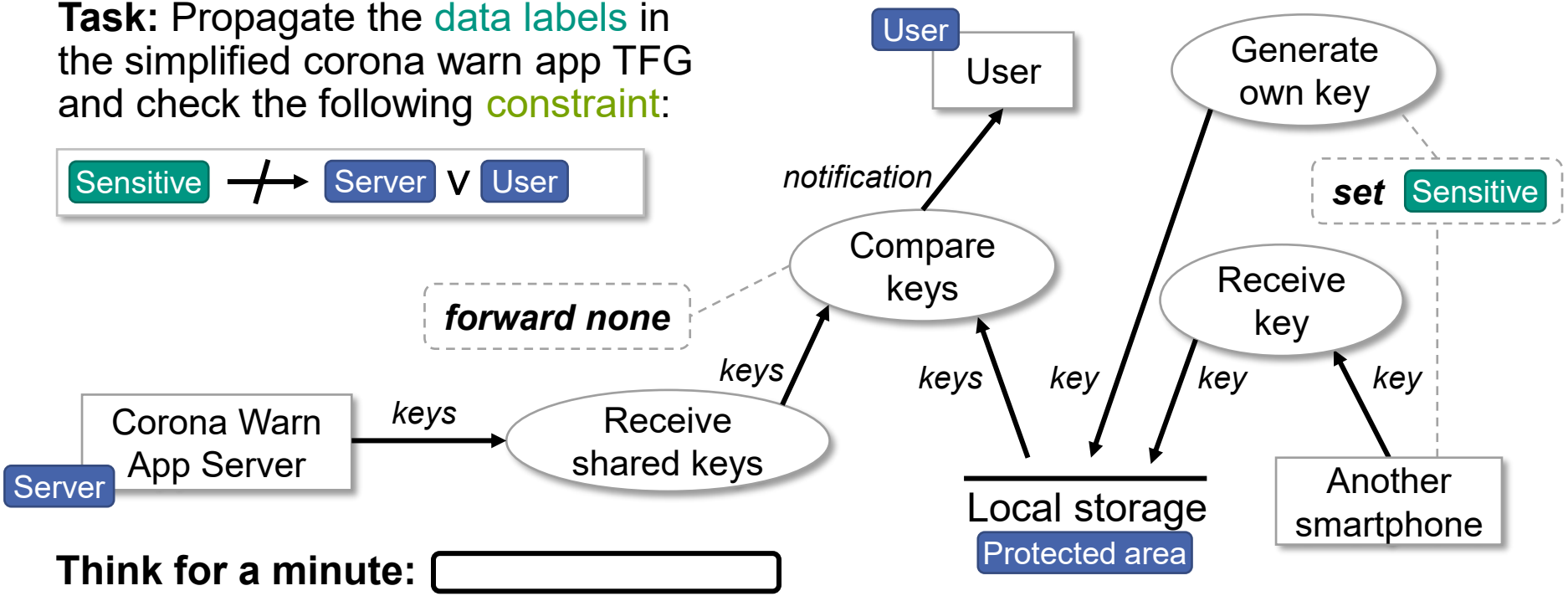
- Label propagation is "like" depth-first search
- Constraint checking is linear
- Thus: **O(|V| + |E|)**



Let's Practice: Data Flow Analysis

Task: Propagate the data labels in the simplified corona warn app TFG and check the following constraint:

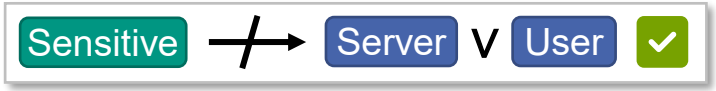
Sensitive ↛ Server ∨ User



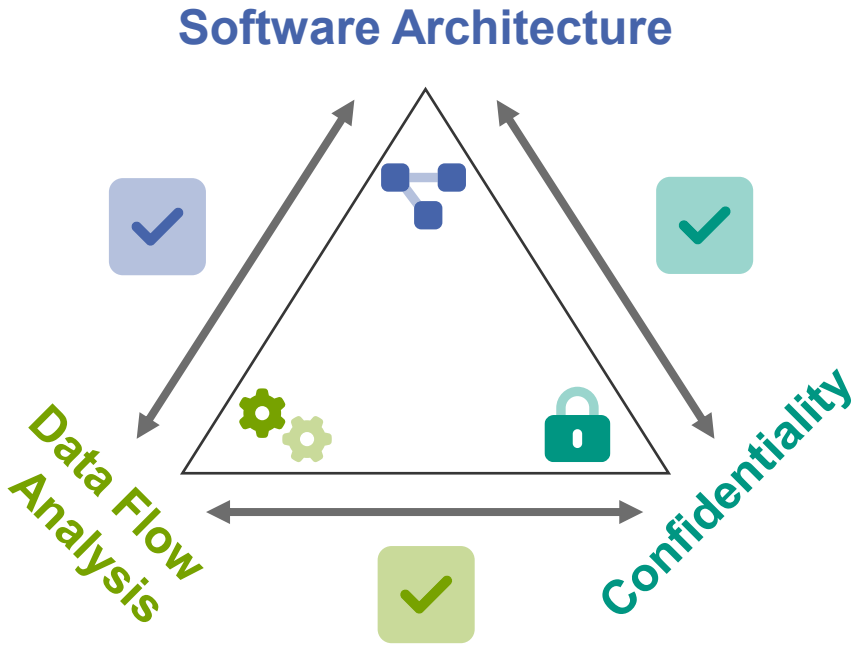
Think for a minute:

Let's Practice: Data Flow Analysis

Task: Propagate the data labels in the simplified corona warn app TFG and check the following constraint:



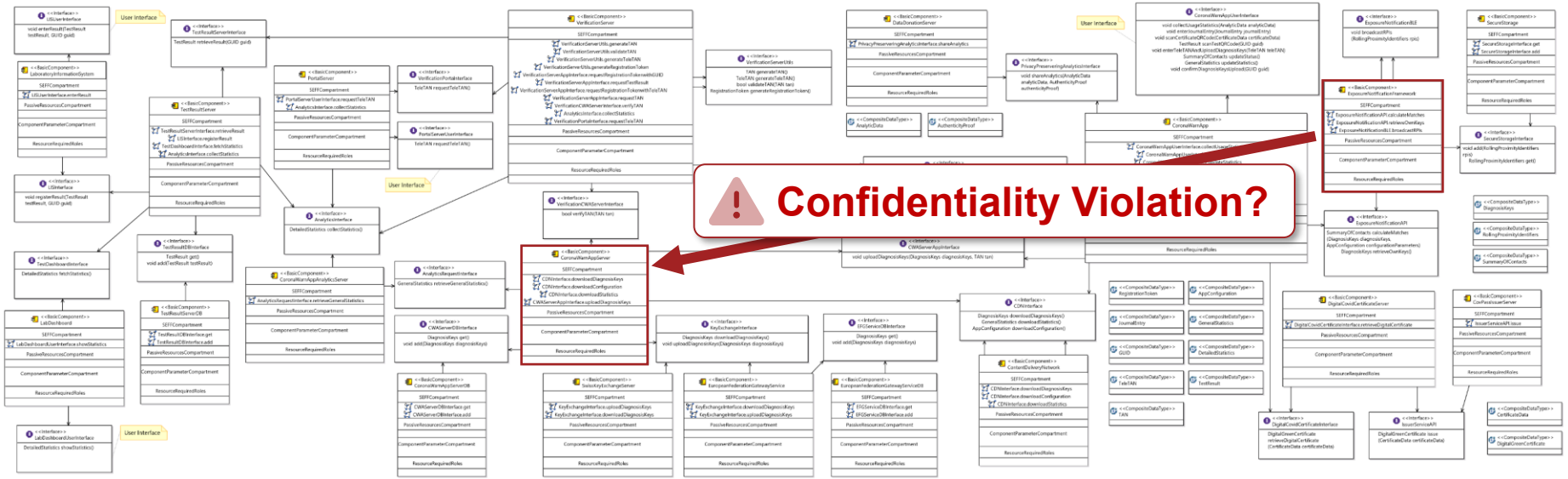
Lecture Overview



Learning Objectives

- Be able to **define** confidentiality and **name** related challenges
- Be able to **draw** data flow diagrams and to **use** them to assess a system's confidentiality
- Be able to **explain** how data flow analysis is used in architecture-based confidentiality analysis
- Be able to **explain** and **apply** the analysis method "label propagation"

Demonstration: Corona Warn App

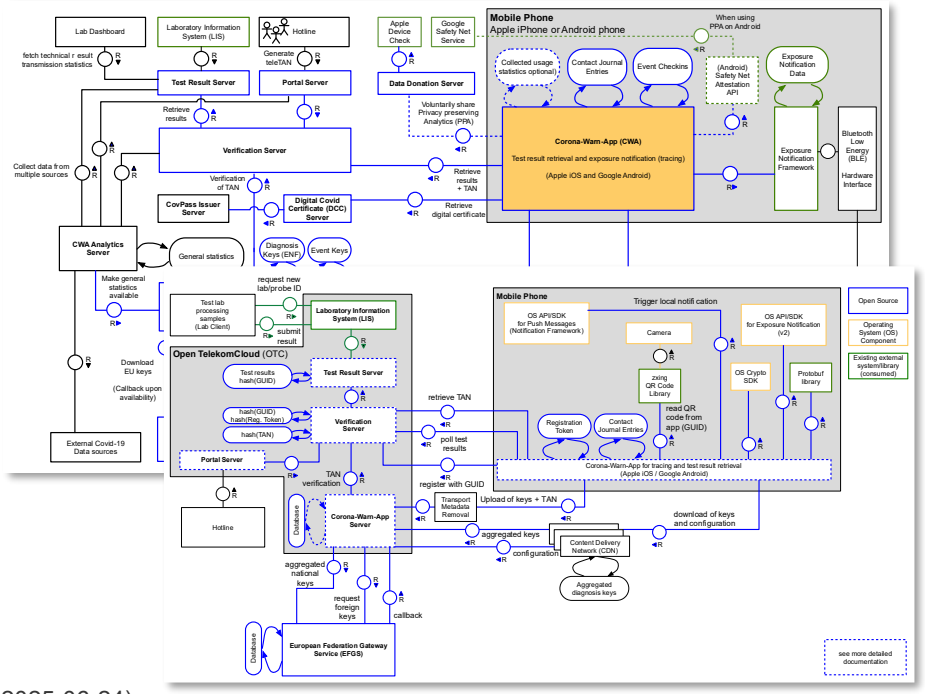


[13]

[13] S. Hahner, Corona Warn App Evaluation Scenario, online available: https://abunai.dev/, 2023, (last checked: 2025-06-24)

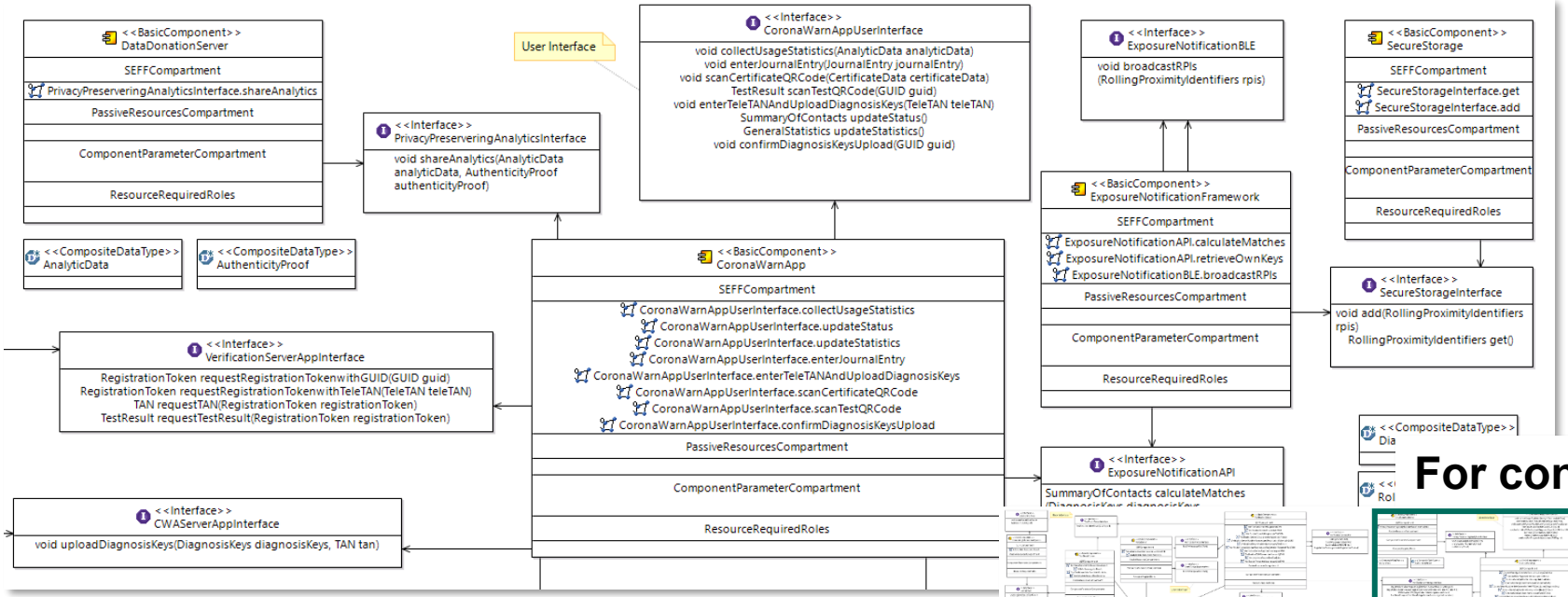
Corona Warn App: Architecture

- Corona Warn App Palladio model [27] using publicly available documentation and models based on SAP TAM [11]
- Modeled features include key handling, covid-19 test results, vaccination certificates, personal contact person diary, international key exchange, …
- Consists of **21** components, resulting in **14** TFGs with a total of **687** vertices
- Used to evaluate a data flow analysis extension to propagate uncertainty [29]



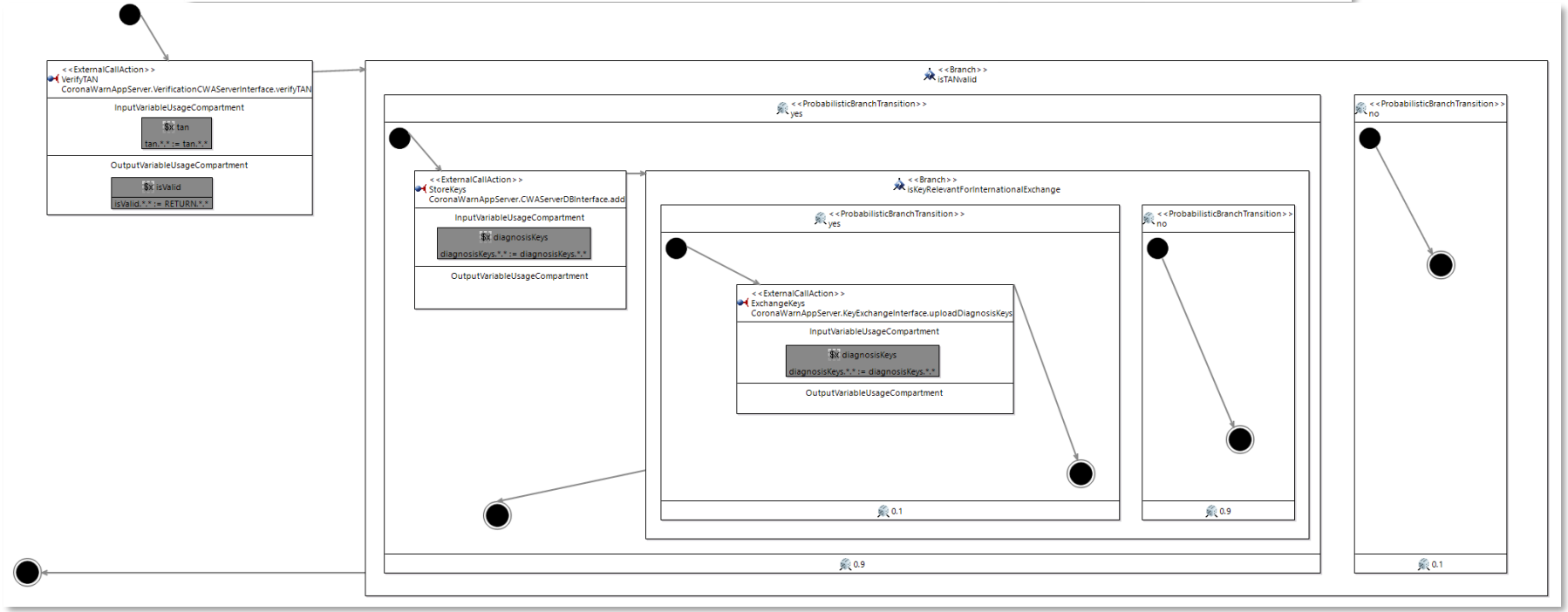
[11] Robert Koch-Institut (RKI), https://github.com/corona-warn-app/, (last checked: 2025-06-24)
[27] R. H. Reussner et al., “Modeling and Simulating Software Architectures: The Palladio Approach.”, The MIT Press, 2016.
[29] S. Hahner, et al., “Architecture-based Uncertainty Impact Analysis to ensure Confidentiality”, SEAMS, IEEE/ACM, 2023.

Corona Warn App: Components



Corona Warn App: High-Level Behavior

SEFF: `CoronaWarnAppServer.uploadDiagnosisKeys`

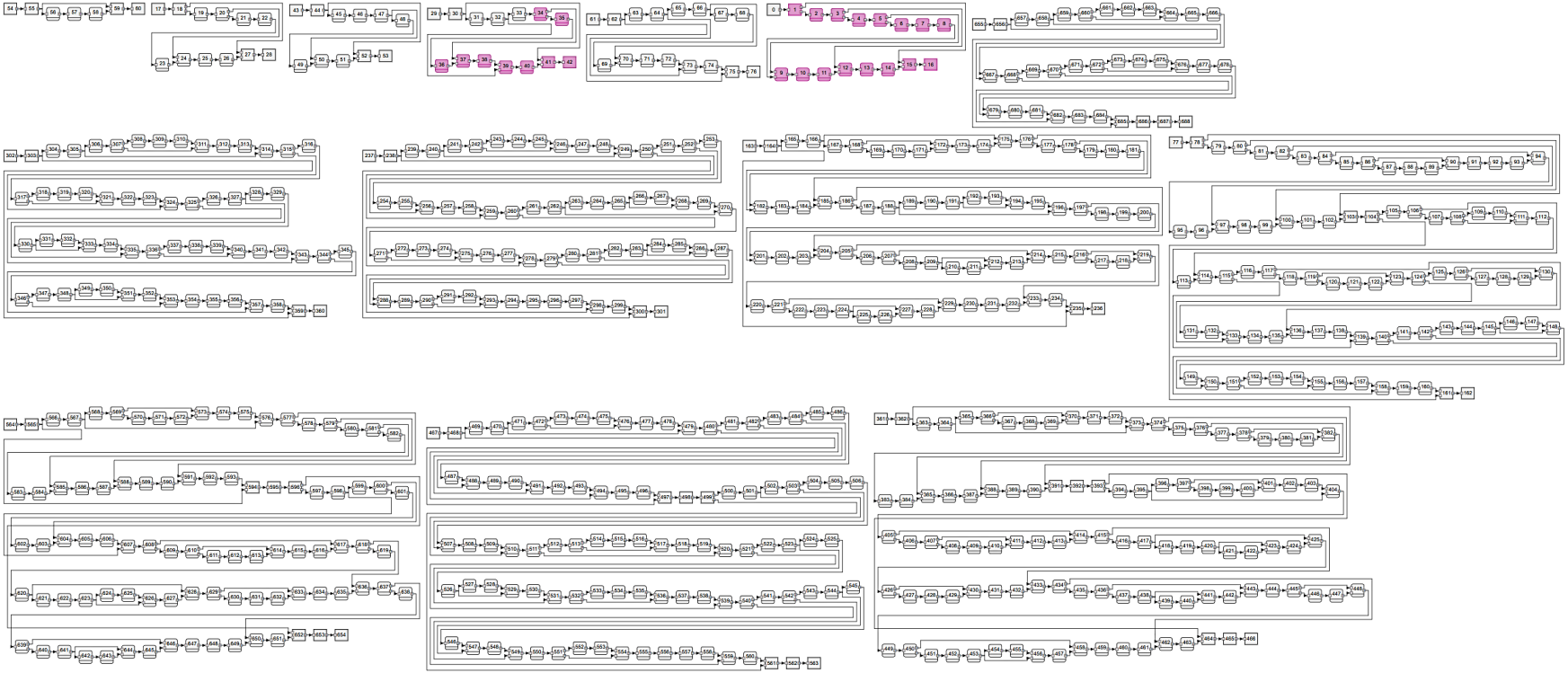


Corona Warn App: Extracted Data Flows

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))
		FActionSequenceElement (ReturnStatistics, _SPW2oLj-Ee25g4oGy4Jxtw))
		lingUserActionSequenceElement / returning (UpdateStatistics, _b_q_ULZ7Ee2xLZp9hElpuA))
		lingUserActionSequenceElement / calling (UpdateStatus, _jd7goLZ7Ee2xLZp9hElpuA))

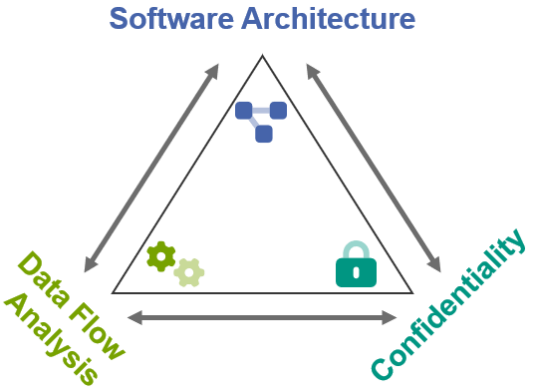
For context:

Corona Warn App: Extracted Flow Graphs



Conclusion

- Use data flow analysis to identify confidentiality violations because *“problems tend to follow the data flow, not the control flow”* **[30]**
- Data flow diagrams, node/data labels, data flow constraints, flow graphs, label propagation



Current Research

- Uncertainty-aware confidentiality analysis **[28]**
- Legal knowledge transfer using data flows **[31]**
- Automated repair of confidentiality violations **[32]**

dataflowanalysis.org **abunai.dev**

[30] A. Shostack, Threat Modeling: Designing for Security. John Wiley & Sons, 2014.
[28] S. Hahner, “Architecture-Based and Uncertainty-Aware Confidentiality Analysis”, Dissertation, Karlsruhe Institute of Technology (KIT), Karlsruhe, 2024.
[31] N. Boltz, et al., "Towards Legal Knowledge Transfer Based on Software Architecture", ECSA, 2025.
[32] N. Niehues, et al., “An Architecture-Based Approach to Mitigate Confidentiality Violations Using Machine Learning”, ICSA, IEEE, 2025.

VECTOR

BUNTE NACHT DER DIGITALISIERUNG

Friday, June 19th, 2026 | 15:00 – 21:00

Vector Campus Karlsruhe
(Next Tram Stop: Hirtenweg/Technologiepark)

Free entry!

Scan the QR code to see the program! Registration optional.

HANDS-ON

Experience software innovation for automotive, medical, aerospace, and more.

WORKSHOP

Bring your own ECG to life and discover how software, hardware, and testing work together.

TECH-TALKS

Software in safety-critical systems & AI put to the test.

References

[1] ISO, “ISO/IEC 27000:2018(E) Information technology – Security techniques – Information security management systems – Overview and vocabulary”, 2018.
[2] ISO, “ISO/IEC/IEEE 42010:2022 Software, systems and enterprise - Architecture description”, 2022.
[3] T. de Marco, „Structured Analysis and System Specification“, YOURDON inc., New York, 1978.
[4] CISPA, https://cispa.de/en/luca-app, 2021 (last checked: 2025-06-24)
[5] FORTUNE, https://fortune.com/2021/06/30/linkedin-data-theft-700-million-users-personal-information-cybersecurity, 2021 (last checked: 2025-06-24)
[6] Security Affairs, https://securityaffairs.com/163999/data-breach/ticketmaster-confirms-data-breach.html, 2024 (last checked: 2025-06-24)
[7] BBC, https://www.bbc.com/news/articles/c984zenjkz5o, 2024 (last checked: 2025-06-24)
[8] IT Governance, https://www.itgovernance.eu/blog/en/chat-app-knuddels-fined-e20000-for-gdpr-breach, 2018 (last checked: 2025-06-24)
[9] CCC, https://media.ccc.de/v/38c3-wir-wissen-wo-dein-auto-steht-volksdaten-von-volkswagen, 2024 (last checked: 2025-06-24)
[10] Spiegel, https://www.spiegel.de/netzwelt/web/volkswagen-konzern-datenleck-wir-wissen-wo-dein-auto-steht-a-e12d33d0-97bc-493c-96d1-aa5892861027, 2024
[11] Robert Koch-Institut (RKI), https://github.com/corona-warn-app/, (last checked: 2025-06-24)
[12] SPIEGEL, https://www.zeit.de/gesundheit/2022-12/gesamtkosten-corona-warn-app-gesundheit-millionen (last checked: 2025-06-24)
[13] S. Hahner, Corona Warn App Evaluation Scenario, online available: https://abunai.dev/, 2023, (last checked: 2025-06-24)
[14] S. Schneider, et al., “How Dataflow Diagrams Impact Software Security Analysis: an Empirical Experiment”, SANER, IEEE, 2024.
[15] B. Arp, et al., “Analyzing Cyclic Data Flow Diagrams Regarding Information Security”, SSP, GI, 2024.
[16] S. Seifermann, et al., “Detecting violations of access control and information flow policies in data flow diagrams”, JSS, Elsevier, 2022.

References

[17] A. Bambhore Tukaram, et al., “Towards a security benchmark for the architectural design of microservice applications”, ARES, ACM, 2022.
[18] S. Hahner et al., “Modeling Data Flow Constraints for Design-Time Confidentiality Analyses”, presented at ICSA, IEEE, 2021.
[19] N. Niehues, et al., “Integrating Security-Enriched Data Flow Diagrams Into Architecture-Based Confidentiality Analysis”, SSP, GI, 2024.
[20] S. Schneider and R. Scandariato, “Automatic extraction of security-rich dataflow diagrams for microservice applications written in Java”, JSS, Elsevier, 2023.
[21] GitHub, “CodeQL: Discover vulnerabilities across a codebase with CodeQL”, https://codeql.github.com/ (last checked: 2025-06-27)
[22] G. B. Herwanto, et al., “From User Stories to Data Flow Diagrams for Privacy Awareness: A Research Preview”, REFSQ, Springer, 2022.
[23] L. Sion, et al., “Security Threat Modeling: Are Data Flow Diagrams Enough?”, ICSEW, ACM, 2020.
[24] N. Boltz and S. Hahner, et al., “An Extensible Framework for Architecture-Based Data Flow Analysis for Information Security”, ECSA, 2024.
[25] xDECAF, more information: https://dataflowanalysis.org, online editor available under: https://editor.dataflowanalysis.org (last checked: 2025-06-28)
[26] S. Seifermann, “Architectural Data Flow Analysis”, WICSA, IEEE, 2016.
[27] R. H. Reussner et al., “Modeling and Simulating Software Architectures: The Palladio Approach.”, The MIT Press, 2016.
[28] S. Hahner, “Architecture-Based and Uncertainty-Aware Confidentiality Analysis”, Dissertation, Karlsruhe Institute of Technology (KIT), Karlsruhe, 2024.
[29] S. Hahner, et al., “Architecture-based Uncertainty Impact Analysis to ensure Confidentiality”, SEAMS, IEEE/ACM, 2023.
[30] A. Shostack, Threat Modeling: Designing for Security. John Wiley & Sons, 2014.
[31] N. Boltz, et al., "Towards Legal Knowledge Transfer Based on Software Architecture", ECSA, 2025.
[32] N. Niehues, et al., “An Architecture-Based Approach to Mitigate Confidentiality Violations Using Machine Learning”, ICSA, IEEE, 2025.