Section: PUBP-6727

# Mutual Monitoring in the Cloud
# Progress Report 2

Alexander Stein

## Problem Statement

Cloud computing infrastructure is essentially ubiquitous, but adoption is not without challenges. Cloud service providers must cater to customers in regulated sectors, complying with cybersecurity frameworks that create high barriers to entry. One barrier is ongoing evaluation of the provider's cybersecurity posture, often resulting in centralized bureaucracies. FedRAMP oversees a prominent example of such a program, the Continuous Monitoring Program, which is emblematic of these barriers. This program requires hundreds of cloud service providers to contract with one of thirty reputable auditor firms. The providers work with the auditors to send security scans and updated security control documentation for FedRAMP-authorized services monthly to FedRAMP reviewers, in some cases for the largest cloud infrastructures in the world. All three parties collaborate in meetings, emails, and a wiki, forming a unique multi-party bureaucracy that both secures and bottlenecks the government's acquisition of modern cloud services.

Are these bureaucracies an optimal solution, or a last resort that fails to keep pace with cloud technology as it proliferates and evolves? If they are a last resort, is there a better way?

## Solution Statement

I will use this research to design and evaluate an alternative to centralized continuous monitoring, mutual monitoring. The foundation of mutual monitoring will be federated data services, known in other security use cases as transparency services. The positives and negatives of FedRAMP's continuous monitoring model will inform its design. Operating such services can change the economics, and thereby the behavior, of cloud service providers and their customers. A new architecture will incentivize auditors to sell value-add analytics via these federated data services, potentially obsoleting centralized authorities for continuous monitoring like FedRAMP.

## Completed Tasks (Last 2 Weeks)

1. I developed a build workflow and automated publication pipeline for the draft architecture specification. Fixing bugs and troubleshooting took more time than I estimated in my plan.

    - I evaluated the use of software for authoring specifications popular with IETF standards authors.
    - Due to complexity of modifying the aforementioned tooling to remove IETF branding, copyright notices, styling, and features not suitable to my specification, I designed a workflow to author and manage drafts of the specification alongside the same source code repository for the prototype using the open-source pandoc utility.
    - I developed an automation pipeline to execute the publication workflow for the specification remotely in GitHub Actions to make the authoring process more consistent over time.
    - I created and completed troubleshooting on a deployment pipeline for final complete copy and incremental draft copies of the specification. The latter is helpful to streamline feedback with multiple advisors with different copies of changes in parallel. I will integrate this work into managing later deployments of the transparency service prototypes, greatly improving productivity.
    - Due to the custom needs of how I deployed pandoc, I created a customized container image for building the website and a PDF copy.
    - I extended my publication workflow to generate images from architecture diagrams in the Mermaid family of declarative domain-specific languages for diagrams.
    - I extended my publication workflow to automatically convert the PDF copy and cross-link to it on the web version to suit the preferences of several of my advisors.

2. I began writing draft architecture specification. I have not completed it on schedule due to the troubleshooting described above.

3. For authoring the specification and preparing to develop the core of the prototype, I reviewed the sources below and modified the outline accordingly.

    - transparency service specifications (e.g. SCITT; C2SP Static Certificate Transparency API) and industry analysis of their efficacy (for deliverable #3)
    - taxonomies and models for auditing and monitoring (for deliverable #2)

## Tasks for the Next Project Report

In the next two weeks, I will focus on the following goals. I have sorted them in order of priority.

1. Request feedback from advisors for the initial draft of the architecture specification.
2. Start development of submission API for cloud service providers and external third-party auditors.
3. Complete outline of FedRAMP critical analysis.
4. Start draft of FedRAMP critical analysis, request feedback from advisors.

## Questions or issues I'm having

### Evaluation and Measurement

1. For more consistency, objectivity, and productivity with sentiment analysis, I am considering the using hosted (e.g. OpenAI ChatGPT, Anthropic Claude) or preferably self-hosted (e.g. ollama and open-weight models) LLM tools to analyze the content in lieu of developing my own Python code. Is this permissible? If so, how do I correctly document and cite this analysis?
2. Many security operations staff do not formally criticize FedRAMP and other programs that require continuous monitoring "on the record" with their name and affiliation in peer-reviewed journals. However, they frequently write about their frustrations with these programs on social media, primarily by name

on LinkedIn and with pseudonyms on platforms like Reddit, Bluesky, and X/Twitter. Is it permissible to consider these as primary sources for quotations? Is it permissible to use them for sentiment analysis?

3. Some users post statistics on aggregated responses to polls on related topics without attribution or details about the respondents. Is it permissible for me to use these polls in my deliverables and final report? Am I allowed to create such polls myself and use them in the final report?

## Methodology Paragraph Summary

For this project, I will use multiple methods to implement an alternative architecture for monitoring cloud services and modeling its potential impact. To start, I will use a quantitative and qualitative analysis of the current shortcomings and gaps for the current FedRAMP Continuous Monitoring Program. This will be the primary example of centralized continuous monitoring for which I design my mutual monitoring model for comparison. For qualitative analysis, I can perform textual analysis and sentiment analysis. I will leverage academic research, industry analysis, and a new primary source: FedRAMP's web-based forums for the 20x reform initiative and its community working groups. In these forums, stakeholders discuss their praise and criticism of current centralized processes and plans for future ones, often summarizing their pain points highly relevant to designing an alternative process. In addition, I will use publicly available information from FedRAMP and industry analysis to quantify the burden of the current FedRAMP Continuous Monitoring and its manual workflow. As I build a prototype based on my architecture, I will design several use cases to estimate the cost and resource efficiency to compare those costs against the estimated costs for my solution. In addition to these methods, I will use advisors familiar with FedRAMP from different stakeholder perspectives to validate information or analysis where these methods prove lacking and leave gaps.

## Timeline

| Week # | Description of Task | Status |
|---|---|---|
| W1 (May 12-18) | Identify references for key research topics. | Complete |

| | | |
|---|---|---|
| W1 | Identify advisors to review FedRAMP analysis and architecture. | Complete |
| W2 (May 19-25) | Initialize code repository for prototype service. | Complete |
| W2 | Present proposal to advisors and integrate feedback; obtain commitment from advisors. | Complete |
| W2 | Read FedRAMP documentation for continuous monitoring processes. | Complete |
| W2 | Begin outline of FedRAMP ConMon critical analysis. | Complete |
| W3 (May 26 - Jun 1) | Implement data service internals and submission API. | Pending |
| W3 (May 26 - Jun 1) | First draft of data service architecture specification. | Pending |
| W4 (June 2-8) | Implement data service internals and submission API. | Pending |
| W4 | Finalize architecture specification with advisors' reviews. | Pending |
| W5 (June 9-15) | Implement data service client to submit to submission API instances. | Pending |
| W5 | Complete data service internals and submission API. | Pending |
| W5 | Complete FedRAMP critical analysis document. | Pending |
| W6 (June 16-22) | Complete data service client to submit to submission API instances. | Pending |
| W6 (June 16-22) | Implement continuous monitoring quantitative processing module for API. | Pending |
| W6 | Design MVP continuous monitoring use cases and quantitative measurements. | Pending |
| W7 (June 23-29) | Complete continuous monitoring quantitative processing module for API. | Pending |

| | | |
|---|---|---|
| W7 | Implement MVP continuous monitoring use cases in API quantitative processing module. | Pending |
| W8 (June 30 - July 6) | Start prototype deployment to cloud service tenants for testing. | Pending |
| W9 (July 7-13) | Complete prototype deployment to cloud service tenants for testing. | Pending |

# Evaluation

[Include any evaluation plans and/or results by Progress Report 4. This may expand as you finalize the report.]

# Report Outline

[Include an outline of your final report by Progress Report 4. This may expand as you finalize the report.]

# References

Aldribi, A., Traore, I., & Letourneau, G. (2015, August). Cloud slicing a new architecture for cloud security monitoring. In *2015 ieee pacific rim conference on communications, computers and signal processing (pacrim)* (p. 18-22). IEEE. Retrieved 2025-05-24, from http://dx.doi.org/10.1109/PACRIM.2015.7334802 doi: 10.1109/pacrim.2015.7334802

Cambric, S., & Ratemo, M. (2023). *Cloud auditing best practices*. Birmingham, England: Packt Publishing. Retrieved 2025-05-23, from https://learning.oreilly.com/library/view/cloud-auditing-best/9781803243771/

Campitelli, V., Catteddu, D., & Maria, J. D. (2020). *Csa's perspective on cloud risk management.* cloudsecurityalliance.org. Retrieved 2025-05-24, from https://cloudsecurityalliance.org/artifacts/csa-s-perspective-on-cloud-risk-management

Carvallo, P., Cavalli, A. R., Mallouli, W., & Rios, E. (2017). Multi-cloud applications security monitoring. In *Green, pervasive, and cloud computing* (p. 748-758). Springer International Publishing. Retrieved from http://dx.doi.org/10.1007/978-3-319-57186-7_54 doi: 10.1007/978-3-319-57186-7_54

Casola, V., De Benedictis, A., Rak, M., & Villano, U. (2020, May). A novel security-by-design methodology: Modeling and assessing security by slas with a quantitative approach. *Journal of Systems and Software*, *163*, 110537. Retrieved from http://dx.doi.org/10.1016/j.jss.2020.110537 doi: 10.1016/j.jss.2020.110537

FedRAMP. (2025). *Fedramp csp authorization playbook.* Retrieved 2025-05-18, from https://web.archive.org/web/20250413105351/https://www.fedramp.gov/assets/resources/documents/CSP_Authorization_Playbook.pdf

Gartner. (2024). *Gartner forecasts worldwide public cloud end-user spending to total $723 billion in 2025.* Retrieved 2025-05-18, from https://web.archive.org/web/20250415023404/https://www.gartner.com/en/newsroom/press-releases/2024-11-19-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-total-723-billion-dollars-in-2025

Ismail, U. M., Islam, S., & Islam, S. (2016, August). Towards cloud security monitoring: A case study. In *2016 cybersecurity and cyberforensics conference (ccc)* (p. 8-14). IEEE. Retrieved from http://dx.doi.org/10.1109/CCC.2016.8 doi: 10.1109/ccc.2016.8

Kuerbis, B., & Mueller, M. (2023, September). Exploring the role of data enclosure in the digital political economy. *Telecommunications Policy*, *47*(8), 102599. Retrieved from http://dx.doi.org/10.1016/j.telpol.2023.102599 doi: 10.1016/j.telpol.2023.102599

Kunz, I., Schneider, A., & Banse, C. (2022). *A continuous risk assessment methodology for cloud infrastructures.* arXiv. Retrieved from https://arxiv.org/abs/2206.07323 doi: 10.48550/ARXIV.2206.07323

Laurie, B. (2014, August). Certificate transparency: Public, verifiable, append-only logs. *Queue*, *12*(8), 10-19. Retrieved 2025-05-24, from https://dl.acm.org/doi/pdf/10.1145/2668152.2668154 doi: 10.1145/2668152.2668154

Lewis Commission. (2025). *Faster into the cloud: Accelerating federal use of cloud services for security and efficiency.* Center for Strategic and International Studies. Retrieved 2025-05-18, from https://web.archive.org/web/20250116234900/https://csis-website-prod.s3.amazonaws.com/

s3fs-public/2025-01/250115_Lewis_Cloud_Commission.pdf

Majumdar, S., Madi, T., Wang, Y., Tabiban, A., Oqaily, M., Alimohammadifar, A., … Debbabi, M. (2019). *Cloud security auditing*. Cham, Switzerland: Springer Nature. Retrieved 2025-05-23, from https://link.springer.com/book/10.1007/978-3-030-23128-6

Mireles, J. D., Ficke, E., Cho, J.-H., Hurley, P., & Xu, S. (2019, December). Metrics towards measuring cyber agility. *IEEE Transactions on Information Forensics and Security*, *14*(12), 3217-3232. Retrieved 2025-05-25, from http://dx.doi.org/10.1109/TIFS.2019.2912551 doi: 10.1109/tifs.2019.2912551

Rescorla, E. (2023a). *A hard look at certificate transparency: Ct in reality.* Retrieved 2023-05-23, from https://educatedguesswork.org/posts/transparency-part-2/

Rescorla, E. (2023b). *A hard look at certificate transparency, part i: Transparency systems.* Retrieved 2023-05-23, from https://educatedguesswork.org/posts/transparency-part-1/

Rescorla, E. (2024). *Why it's hard to trust software, but you mostly have to anyway.* Retrieved 2025-05-24, from https://educatedguesswork.org/posts/ensuring-software-provenance/

Soveizi, N., & Turkmen, F. (2023). *Secflow: Adaptive security-aware workflow management system in multi-cloud environments.* arXiv. Retrieved from https://arxiv.org/abs/2307.05137 doi: 10.48550/ARXIV.2307.05137

stackArmor. (2024). *How much does fedramp compliance cost?* Retrieved 2025-05-18, from https://web.archive.org/web/20240808151743/https://stackarmor.com/how-much-does-fedramp-compliance-cost/

Torkura, K., Sukmana, M. I., Cheng, F., & Meinel, C. (2021, March). Continuous auditing and threat detection in multi-cloud infrastructure. *Computers & Security*, *102*, 102124. Retrieved from http://dx.doi.org/10.1016/j.cose.2020.102124 doi: 10.1016/j.cose.2020.102124

Weir, G. R. S., & Aßmuth, A. (2024). *Strategies for intrusion monitoring in cloud services.* arXiv. Retrieved from https://arxiv.org/abs/2405.02070 doi: 10.48550/ARXIV.2405.02070

# Appendix

# An Architecture for Mutual Monitoring of Cloud Infrastructures

**Author:** A.J. Stein **Version:** 7e1e41ba402f8d5110a9f4400652cf4bbef8a3c9
**Modified at:** 2025-06-08

*The source code from github.com/aj-stein/conmotion at the linked commit generated this copy of the specification, supporting documentation, and related code. You can click this link to download this specification as a PDF document.*

*NOTE: This specification conforms to IETF's best practice in RFC 2119 to capitalize all letters in key words to indicate requirement levels (Bradner, 1997). This specification also capitalizes certain words or phrases with common meaning when this specification gives them a precise normative definition. See the Terminology section for a complete listing of these terms.*

## Abstract

The transparency of cloud infrastructures is a systemic challenge to industry.

Internal or external stakeholders of a cloud infrastructure may want to publish or verify data about its operational, resiliency, or security properties. However, there are no specifications for common data structures, protocols, or measurement algorithms to transparently demonstrate evidence of those properties at once or over a time interval. This document proposes an architecture that specializes the Transparency Service architecture for providers of cloud infrastructures. The specialization of this architecture will enable them to publish evidence of security properties with verifiable digital signatures. Providers of cloud infrastructures, their consumers, or external auditors may also publish counter-signatures to verify multi-party evaluation and verification of this evidence, known as a mutual monitoring network.

## Introduction

Cloud infrastructures require their providers to design, implement, and document security properties against a threat model and actively monitor these properties for their efficacy. Moreover, cloud infrastructures have essential characteristics that uniquely distinguish them from other deployment models. They have measured services where the provider and consumer control components automatically and precisely through metering capabilities and on-demand self-services for consumers to unilaterally provision components (Mell & Grance, 2011, p. 2).

Despite these essential characteristics and the proliferation of many differentiated, proprietary services for cloud infrastructures, there is no de-facto standard or vendor-agnostic solution to publish digitally-signed data for a cloud service infrastructure, counter-sign the data to acknowledge and verify its contents, and/or enrich a collection of this data with verifiable measurements. Different

providers have monitoring capabilities for security properties of cloud infrastructures, but most are partial, proprietary, confidential, and do not permit scalable multi-party verification of data. Therefore, a Transparency Service architecture is needed for different parties to publish signed data, counter-sign acknowledgements, and publish follow-on measurements for parties to mutually monitor heavily interconnected infrastructures.

This specification specifies an architecture for a Transparency Service to concurrently monitor the security properties of one or more cloud infrastructures by multiple parties, both internal and external to the the infrastructure provider. Previously, experts drafted Transparency Service architectures for monitoring the lifecycle of TLS certificates for encrypted communications on the World Wide Web (Laurie, Messeri, & Stradling, 2021) and another for heterogenous data for software supply chain use cases (Birkholz, Delignat-Lavaud, Fournet, Deshpande, & Lasker, 2025). An industry consortium deployed an emerging de-facto standard, Sigstore and Rekor, for monitoring published open-source software used industry-wide (Sigstore Developers, 2025). Google's Android operating system developers deployed their own to verify the legitimacy of all compiled programs in their operating system releases (Google, 2025). Although they represent similar use cases, the uniqueness of cloud infrastructure requires different design and implementation tradeoffs. Therefore, this specification will inventory use cases; describe the foundation and enhancements to the baseline Transparency Service architecture; the actors in a mutual monitoring network and their roles; specialized components of the architecture; and required protocols for actors to execute their roles with the architecture for given use cases.

## Use Cases

This specification addresses the needs of several use cases for mutual multi-party monitoring of security properties for cloud infrastructures.

### Monitoring System Inventory

Inventory management of systems that comprise components of a cloud infrastructure is a foundational requirement for many security control frameworks that organizations use whether or not they maintain a cloud infrastructure. Examples include control 5.9 in ISO 27001:2022 (2022), control PM-5 in the Special Publication 800-53 Risk Management Framework (NIST, 2020, p. 206), the control CCC-04 in the Cloud Controls Matrix (Cloud Security Alliance, 2024, p. 79), and numerous others. For a cloud infrastructure to satisfy these control requirements, they must maintain an inventory, often incredibly dynamic due to characteristics of cloud computing, for all systems the compromise the components of that infrastructure. Cloud infrastructure providers have different actors, performing different roles, where they must produce, consume, and/or verify data about the inventory of that infrastructure.

**Cloud Infrastructure Provider** A cloud infrastructure provider uses bespoke asset management system(s) predominantly for internal use. The provider's staff can use a Transparency Service as a high-fidelity replica of the asset management system(s) data, tracking changes over time, or as the canonical source of inventory. The provider's staff will integrate inventory management automation to create new entries into the append-only log of the Transparency Service, adding digitally signed records one-by-one for the provisioning and deprovisioning of all systems in the infrastructure. The most recent record embeds a linkage by hash to the previous record in the append-only log. Staff can check the most recent record to now the latest changes or "replay the log" with the fully exported data of the append-only log to understand all changes over time and compose a realistic model of the services monitored.

**Cloud Infrastructure Customer** A customer of a cloud infrastructure uses the cloud infrastructure provider as a dependency to build their own application services or derivative cloud infrastructure, thereby creating its own need for an asset management system and inventory. By virtue of this architecture, the customer's staff must maintain their own inventory, but the assets they manage will be instances of cloud infrastructure systems provided by the upstream cloud infrastructure provider. The customer will use the upstream cloud infrastructure provider's transparency log, consuming digitally signed records and publishing digitally signed receipts to their own transparency log, acknowledging existence of the upstream infrastructure they use to provision an instance in their own infrastructure. This customer will also generate their own records for both internal and external use for their own downstream customers to confirm accurate inventory management.

**Auditor** An auditor, accountable to the cloud infrastructure provider, their customer, or both, must review the efficacy of security control implementations through expert review of artifacts. In the case of inventory management, it is important for the auditor to use these artifacts as evidence. The auditor compares the evidence from the provider to their own artifacts they collect independently, and verify the provider's inventory is accurate and has no anomalies. Auditors can consume the append-only log of the Transparency Service to ascertain contemporary or historical view of the provider's inventory and thereby the efficacy of their inventory management techniques. Auditors can also digitally sign receipts and append them the transparency log to endorse inventory records, so that customers of the cloud infrastructure provider can analyze auditor endorsements in transparency log records to acquire cloud infrastructure or continue using it.

### Monitoring Configuration Management

Configuration management for systems that comprise components of a cloud infrastructure is a foundational requirement for many security control frameworks. Examples include control 8.9 in ISO 27001:2022 (2022), multiple controls in

the Configuration Management (CM) control family for the Special Publication 800-53 Risk Management Framework (NIST, 2020, pp. 96–114), the control CCC-03 in the Cloud Controls Matrix (Cloud Security Alliance, 2024, p. 77), and numerous others. For a cloud infrastructure to satisfy these control requirements, the provider's staff must have known configuration baselines for their inventory, apply them, and possibly prevent provisioning outside of approved processes and create or change assets to not conform to the baselines. Cloud infrastructure providers have different actors, performing different roles, where they must produce, consume, and/or verify data about the configuration management for that infrastructure.

**Cloud Infrastructure Provider** A cloud infrastructure provider uses bespoke configuration management system(s) mostly for internal use. The provider's staff can use a Transparency Service as a high-fidelity replica of the configuration management system(s) data, tracking changes over time, or as the canonical source of inventory. This data will cross-reference which systems link to which configurations with both datasets on the Transparency Service. The provider's staff will integrate inventory management and configuration management automation to create new entries into the append-only log of the Transparency Service, adding digitally signed records one-by-one for the creation, modification, and deletion of configurations for different assets in the cloud infrastructure. The most recent record embeds a linkage by hash to the previous record in the append-only log. Staff can check the most recent record to now the latest changes or "replay the log" with the fully exported data of the append-only log to understand all changes over time and compose a realistic model of the services monitored.

**Cloud Infrastructure Customer** A customer of a cloud infrastructure uses the cloud infrastructure provider as a dependency to build their own application services or derivative cloud infrastructure, thereby creating its own need for an configuration management system. By virtue of this architecture, the customer's staff must maintain their own inventory and configuration management database, even the assets and their configurations are instances of systems the upstream cloud infrastructure provides. The customer will use the upstream cloud infrastructure provider's transparency log, consuming digitally signed records and publishing digitally signed receipts to their own transparency log, acknowledging existence of the upstream infrastructure they use to provision an instance in their own infrastructure. This customer will also generate their own records for both internal and external use for their own downstream customers to confirm accurate inventory management.

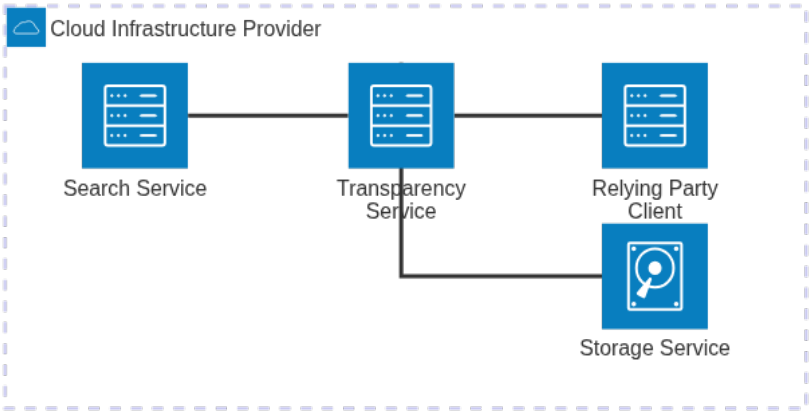
**Auditor** An auditor, accountable to the cloud infrastructure provider, their customer, or both, must review the efficacy of security control implementations through expert review of artifacts. In the case of configuration management, it is important for the auditor to use these artifacts as evidence. The auditor compares

the evidence from the provider to their own artifacts they collect independently, and verify the provider's inventory and related configuration management records are accurate and without anomalies. Auditors can consume the append-only log of the Transparency Service to ascertain contemporary or historical view of the provider's configuration management records and thereby the efficacy of their configuration management techniques. Auditors can also digitally sign receipts and append them the transparency log to endorse inventory records, so that customers of the cloud infrastructure provider can analyze auditor endorsements in transparency log records to newly acquire cloud infrastructure or continue using it.

## Architecture

The mutual monitoring architecture specializes the architecture of a Transparency Service as defined by the IETF SCITT Working Group (Birkholz et al., 2025). This architecture includes a Transparency Service; Adjacent Services, custom services deployed adjacently to the Transparency Service for log search and storage; and Relying Parties, Transparency Service clients that serve specialized use cases for processing the content of each record in the Append-only Log.

Given the above use cases, a cloud infrastructure provider MAY deploy these components with logical relationships like those in the diagram below.




### Components

#### Transparency Service

#### Registration Policy Engine

#### Append-Only Log

### Adjacent Service for Storage

### Adjacent Service for Search

### Actors and Roles

### Flows

## Terminology

- Transparency Service: This document uses the normative definition from the IETF SCITT Architecture (Birkholz et al., 2025).
- Relying Party: This document uses the normative definition from the IETF SCITT Architecture (Birkholz et al., 2025).

## Appendix

### References


Birkholz, H., Delignat-Lavaud, A., Fournet, C., Deshpande, Y., & Lasker, S. (2025). *An architecture for trustworthy and transparent digital supply chains* (Internet Engineering Task Force SCITT Working Group, Ed.) [Internet-Draft]. Retrieved from https://datatracker.ietf.org/doc/draft-ietf-scitt-architecture/12/

Bradner, S. O. (1997). *Key words for use in RFCs to Indicate Requirement Levels*. RFC 2119; RFC Editor. https://doi.org/10.17487/RFC2119

Cloud Security Alliance. (2024). *CCM v4.0 implementation guidelines* (No. 2.0). Cloud Security Alliance. Retrieved from Cloud Security Alliance website: https://cloudsecurityalliance.org/download/artifacts/cloud-controls-matrix-v4

Google. (2025). *Android binary transparency*. Retrieved from https://developers.google.com/android/binary_transparency/overview

International Organization for Standardization. (2022). *Information security, cybersecurity and privacy protection — Information security management systems — Requirements* (Vol. 2022) [Standard]. Geneva, CH.

Laurie, B., Messeri, E., & Stradling, R. (2021). *Certificate transparency version 2.0*. RFC Editor. https://doi.org/10.17487/rfc9162

Mell, P. M., & Grance, T. (2011). *The NIST definition of cloud computing*. National Institute of Standards; Technology. https://doi.org/10.6028/nist.sp.800-145

NIST. (2020). *Security and privacy controls for information systems and organizations* (No. NIST Special Publication (SP) 800-53, Rev. 5). Gaithersburg, MD: National Institute of Standards; Technology. https://doi.org/10.6028/nist.sp.800-53r5

Sigstore Developers. (2025). *Rekor*. Retrieved from https://docs.sigstore.dev/logging/overview/

# Mutual Monitoring in the Cloud Outline

## Problem Statement

Cloud service providers must cater to customers in regulated sectors with high barriers to entry. One barrier is evaluation of the provider's cybersecurity posture, mostly with centralized bureaucracies. Although preemptively limiting cybersecurity risk, such mandatory evaluation often means significant delay and investment before regulated customers can use new or changing building blocks they urgently need for their digital services. Are these bureaucracies an optimal solution or a last resort that failed to keep pace with cloud technology? If the latter, is there a better way?

## Deliverables

### Critical Analysis of FedRAMP

### Brief History and Context

### Current FedRAMP Process

### Initial Authorization

### Continuous Monitoring

### Agency versus JAB Authorizations

### 20x

### Continuous Monitoring in Detail

### Benefits

### Challenges and Problems

### Mutual Monitoring Architecture

### Prototype Code

### Foundational Quantitative Metrics Framework

## Appendix

### Research

### Transparency systems and architecture

- (Rescorla, 2023b)

  - The key word in this system is trust: the endpoints need to trust that the authentication service doesn't falsely attest to a binding for the wrong person (technical term: "misissuing").
  - There are a number of potential approaches for defending against this problem but the one that the community seems to have settled on is what's called a transparency system. The basic concept of such a system is that you retain the idea of a trusted authentication service but add on a layer in which it publishes the bindings it is attesting to so that anyone can check that it's not misissuing.
  - The basic idea behind a transparency system is not to prevent misissuance but to detect it.
  - The obvious thing to to do is for the AS to just publish the list of certificates it has issued on its Web site, but this isn't secure. Consider what happens if the AS gives different answers to different people, like so [...]
  - Moreover, it's not really required that everyone get a full copy of the database: consider the case where we have a fake certificate for example.com.
  - The problem, obviously, is that this kind of flood fill is incredibly inefficient: Let's Encrypt alone has about 300 million valid certificates; at 1K each, this would be a database of 300GB, not something you want to be storing on your phone, let alone having to send to everyone else you come into contact with—ignoring for the moment the question of how you're going to transmit the database around. Clearly, this simple system is not practical.
  - The idea behind a Merkle Tree is to allow a way to efficiently commit to a set of values without actually publishing any of the values.
  - The inclusion proof is comparatively small; using Let's Encrypt as our reference point, it will be about 600-700 bytes.
  - This isn't perfect in that you still have to actually detect misissuance, which isn't always straightforward for the reasons I discussed above, but at least it's not possible to have covert misissuance.
- (Rescorla, 2023a)
  - This design has several advantages. First, it makes life easier for the CAs, who don't have to run logs. This may not seem like a big deal, but it turns out that running a log is a lot of work for reasons we'll get into below, and indeed very few CAs actually run their own logs today. Instead, some entity with a lot of operational resources and experience (i.e., Google), could run a log that supports multiple CAs, hopefully making it easier for the CAs to deploy.
  - Second, having a relatively small number of logs improves the scaling properties of the system somewhat: much of the overhead for the clients comes in the form of getting an authentic copy of the signed root (what CT calls a signed tree head (STH)), and if each CA has its own tree, that means one root for each CA. If there's just a small number of logs then you need a correspondingly smaller number of

roots.

- Finally, the log design makes it possible to publish certificates even for CAs which don't participate because the log can just unilaterally ingest those certificates.
- In order to address this issue, Google introduced a new concept, the signed certificate timestamp (SCT). An SCT is a signed promise that the log will add the certificate to their tree soon, even though they haven't yet. The figure below shows the issuance flow with SCTs.
- The good news is that CT with SCTs is minimally disruptive while also allowing the browser to enforce the use of CT. The bad news is that it has totally different and much weaker security properties from the system we started with. The problem is that the SCT is just a promise that the log will incorporate the certificate into their Merkle tree, rather than a proof that it actually did, so you're reduced to trusting the log not to lie.
- Because the source of the problem is that the client isn't verifying inclusion of the certificate (by checking the inclusion proof) but only that the log says it would include it (by checking the SCT), the obvious fix is to have the client somehow verify that the certificate actually was included. This turns out to be somewhat challenging and there have been a number of attempts, none of which really work.
- In order to prevent this form of tracking, we need some way for the client to retrieve the inclusion proof anonymously. There are a number of possible options here (VPNs or proxies) or Private Information Retrieval. As far as I know, no log deploys any kind of PIR—it would probably be quite expensive—and while proxies or VPNs are technically feasible, they're not free to run. There are similar problems with clients reporting certificates which are not included but should have been. I'm not aware of any major browser which verifies certificate inclusion proofs [Update 2023-12-25] by default (Chrome had some ideas about using DNS,[2] but seems to have abandoned them.[3]), though see below.
- this seemed kind of impractical when CT was originally designed, but in the intervening 10 years, automatic certificate issuance has become far more common (specifically, a protocol called ACME, originally developed for Let's Encrypt), and so it wouldn't be that hard to imagine modifying ACME to send an updated certificate. Importantly, this is something that could be deployed incrementally, because clients have to be able to fall back to SCTs anyway. However, it doesn't seem to be something that's happening.
- Even if we did have some mechanism for verifying the inclusion proof, we still have the problem of getting consensus on the STHs. The original CT design assumed a flood fill technique (what they called "gossip") like I described in part I, but was frustratingly short on specifics [. . . ]
- The problem is that it's expensive futureproofing, both in terms of

protocol complexity and in terms of operational brittleness. A fairly large fraction of the CT RFC is concerned with specifying the Merkle trees, the machinery of Merkle tree proofs, and the like. All of this could just go away if we were to just treat CT as a "countersign + publish" protocol, leaving a dramatically simpler protocol that would be a thin layer on top of HTTP.
  - Worse yet, CT logs turn out to be hugely operationally complex to run correctly. I haven't personally operated one, but the basic problem seems to be tight timing requirements combined with the immutability of the Merkle tree structure.
  - Despite everything I've said above about the limitations of CT verifiability, it's still proven to be exceedingly useful.
  - By contract, CT is yet another patch on top of the WebPKI, but was incrementally deployable. Imperfect though it is, it has gone a long way towards improving the system, both by making undetected misissuance harder and by making simple misbehavior easier to spot and address.
- (Rescorla, 2024)
  - My long-time collaborator Richard Barnes[1] used to say that "in security, trust is a four letter word", and yet the dominant experience of using any software-based system—which is, you know, pretty much anything electronic—is trusting the manufacturer.
  - The release then gets uploaded to the vendor's website, which is probably hosted on some cloud service like Amazon or Netlify. You can also host the binaries on GitHub. In principle, users could just download the binaries directly from your site (or GitHub) but it's common to instead use a content distribution network (CDN) like Cloudflare or Fastly which retrieves a copy of the binary once, caches it, and then gives out copies to each user. CDNs are designed for massive scaling, thus saving both load on your servers and cost.
  - The basic problem with code signing systems is that they rely heavily on user diligence, because the OS only verifies that the code was signed but doesn't know who was supposed to sign it.
  - Of course, if you have to download software over secure transport anyway, this raises the natural question of why bother to sign the code at all? Why not just have all downloads happen over secure transport? One reason is that is that signing allows for third party hosting.
  - If you require software to be signed by some key that chains back to some non-free credential, then this allows you to increase the level of friction to distribute all software, but especially malicious software.
  - Using secure transport to the package repository is standard practice, but it doesn't help against either of these threats because the problem is the data on the package manager itself is compromised. In theory it seems like signatures offer a way out of this: if packages are signed then even if the attacker compromises the repository they won't be

able to replace the package with their own. Unfortunately package signing isn't a complete solution for the same kind of identity reasons as before.
  - Of course, this was all warmup for the real problem we want to solve. Everything up to now was about ensuring that you get the binary that the publisher wanted to send you. This still leaves you trusting the publisher, which you shouldn't, both because it's bad security practice to have to trust people and because there is plenty of evidence of software publisher misbehavior.
  - The process starts with the publisher releasing the source code. As a practical matter, some kind of review of the source code is a necessary but not sufficient precondition to being able to have confidence in a piece of software. Reviewing the binary is not really practical on any kind of scalable level; it is of course possible to reverse engineer binaries, but it's incredibly time consuming even for experts.
  - BT is like Certificate Transparency (CT) but instead of publishing every certificate, the publisher instead publishes a hash of every binary they release.
  - The sigstore project provides generic tooling for binary signing, reproducible builds, and binary transparency and seems to be getting some uptake. However, we're not seeing the kind of large-scale deployment that we have for certificate transparency, and there don't seem to be any generic logs in wide use like there are with CT. Facebook has also deployed a system called Code Verify to provide a form of binary transparency for Facebook Messenger.
  - We're well over 7000 words already, but I do just want to briefly touch on the topic of the Web. The Web has a number of properties that do make the problem somewhat easier [...] However, it also has several important properties that make the problem much harder: [...]
- (Kuerbis & Mueller, 2023)
  - Data enclosure is defined here as the process by which the information about user activity generated by digital operations are withdrawn from an open or shared arrangement with other operators and made more exclusive to the service providers whose operations generate the data. (p. 1)
  - But data enclosures do not just respond to users' and governments' privacy concerns, they may also provide a competitive advantage to a platform by excluding their competitors or other players from access to data generated by their users. Internalizing the costs of protecting privacy can also privatize the economic benefits of data. Any realistic appraisal of the platform economy must look at both sides of this equation. (p. 2)
  - 'Data' is often reified as the resource of the digital economy, but this can be misleading. It implies that the platform economy's value stems from a recorded and stored pile of bits and bytes that, like oil ("data is the new oil"), can be "mined" for its economic value. This view

of data as a static commodity obscures the operational and business reality of the platform economy. Platforms are multisided markets (Evans & Schmalensee, 2016). (p. 2)

- Contrary to the popular critique characterizing users as exploited "unpaid data producers" (Pistor, 2020), a mutual economic benefit underlies the user-platform interaction. Consumers are "paid" by the platforms' provision of free information services and matches. Many of these services are incredibly valuable but now seem to be taken for granted, perhaps because of the platforms' success at making them ubiquitous: search engines, email, informational and entertainment content, document storage, and thousands of connections to products, other individuals and groups. Economists working out the logic of multi-sided markets have repeatedly demonstrated how it makes business sense for one side to subsidize entry or participation costs of another side (Rochet & Tirole, 2006; Parker et al., 2016). It is also misleading to suggest that the users alone are the "producers" of the data, or that data is being "extracted" from them. While their activities do provide content, without the platform infrastructure itself there are no data and no useful applications for the data. Even severe but realistic critics of the digital market economy (Benthall & Goldenfein, 2021) admit that the data is co-produced. (p. 2)
- Data enclosure is a way to create exclusive access to data, and thus raises the issue of property rights in information. (p. 2)
- In a purely abstract sense, digital data seems to resist exclusivity. Digital data is nonrival in consumption (i.e., one person's use of it does not consume or "use up" the resource). Because of the rapid, practically costless way it can be duplicated and transmitted, it is also notoriously difficult to contain. (p. 2)
- In their data enclosure initiatives, platforms and other digital service providers are establishing de facto control over the data by contractual and technological means. Platforms withhold from others operational data generated by their users, instead of sharing it. (p. 3)
- An analysis of data enclosure suggests that in a contested, multi sided market, a digital platform provider competing for users would have two distinct incentives: 1) an incentive to use the privacy and security of their users' data as a product differentiator; 2) an incentive to assert or maintain exclusivity over the data co-generated by its users and its infrastructure to maintain or increase a competitive advantage in the provision of intermediation. (p. 3)
- However, the privacy differentiation incentive might also push them toward limiting potential ways of monetizing the data/users to which they have exclusive access. (p. 3)
- For its part, Google did not come up with the idea for DoH, did not promote it as part of its competitive strategy, nor did it implement it in a way that could have reinforced its advantages. Google could have leveraged the dominance of its Chrome browser to implement

DoH in a way that defaulted DNS resolution to its own DoH resolver service. It refrained from doing so. Google went along with DoH, but did not lead it. But then, it did not need it. (p. 7)

- (Laurie, 2014)
  - I have written extensively on what is wrong with Bitcoin (for example, it is the least green invention ever, and all of its history could be destroyed by a sufficiently powerful adversary if it were truly decentralized, which it is not). Nevertheless, people continue to worship irrationally at the altar of Bitcoin, and this worship extends to DNS and keys—for example, DNSChain (https://github.com/okTurtles/dnschain). Apart from being an extremely costly solution (in terms of wasted energy, in perpetuity), it also introduces new trusted third parties (those who establish the "consensus" in the block chain) and has no mechanism for verification.
- (Aldribi, Traore, & Letourneau, 2015)
  - If a provider is unwilling to disclose information about their security features, we propose allowing clients to deploy their own network level security. Unfortunately, the current cloud model does not allow clients any network level access [2] (see section 3). Therefore, there is a fundamental relationship between the decision making of stakeholders and security transparency problems. (p. 18)
  - Cloud slicing uses hardware partitioning (sometimes named logical partitioning) to allow a client access to the network level of the server. With network level access, clients can implement further protection measures to give assurance to themselves and their customers. (p. 18)
- (Soveizi & Turkmen, 2023)
  - In a multi-tenant environment, tenants themselves have the potential to become threat actors. Malicious tenants can launch various attacks to compromise the assets of other tenants. Their objectives may involve unauthorized access to sensitive information, manipulation of tasks and data, or causing disruptions. (p. 4)
  - Account hijacking and metadata spoofing present significant threats to the sensitive data of users and tenants. These attacks can compromise the CIA of user and tenant responsibilities. Another attack to be aware of is Economic Denial of Sustainability (EDoS), which targets customers' economic resources by fraudulent billing for resource consumption. (p. 6)
  - A framework of a "mimic cloud workflow execution system" is proposed in [17] featuring three strategies: heterogeneity (diversification of physical servers, hypervisors, and operating systems), redundancy (Lagged Decision Mechanism), and dynamics (switching workflow execution environment). This system only covers the execution and monitoring phases of the workflow life cycle and cannot carry out adaptation of the process instances to react to security violations. (p. 6)
  - We assume that the tenants' resources are cleanly isolated from each

other and may be on the same cloud node; we also assume that the middleware is a logically centralized component that can be hosted by a third party. (p. 8)
  - To meet the requirements of a security-aware WfMS, our work focuses on implementing a comprehensive monitoring procedure to detect potential attacks in the considered deployment model. Figure 3 provides a basic overview of this monitoring procedure, illustrating the locations of monitoring modules and their areas of responsibility. This monitoring approach aims to preserve privacy, safeguard sensitive information, and provide the capability to detect all possible attacks. (p. 8)
  - Global Monitoring and Detection: This module is responsible for real-time monitoring of cloud behavior by analyzing the cloud log file and the network traffic data. It includes two main modules: a) Service Model Trainer [. . . ] . b) Service monitoring: [. . . ] c) Tenant's Rule-based Intrusion Detection System (IDS) [. . . ] (p. 11)
  - Unlike previous studies, SecFlow comprehensively addresses security and privacy concerns throughout the entire workflow lifecycle, with particular emphasis on the detection and reaction to violations that are positioned in the adaptation phases of workflows. By considering threats from various parties, SecFlow provides an extensive monitoring functionality of malicious behavior at different levels (e.g., tenant and middleware) and detects abnormal activities. By leveraging the collected monitoring information, many adaptations become possible for safeguarding user privacy and/or tenant confidentiality. The proposed architecture was implemented by extending the jBPM engine and integrating it with the Cloudsim Plus simulation tool. (p. 14-15)
- (Ismail, Islam, & Islam, 2016)
  - Nevertheless, users have over the years gained appreciable understanding of the need to monitor assets and the threats associated with the benefits of cloud computing, and to some extents, methods have been devised by CSPs to effectively provide users with monitoring capabilities so as to increase the adoption of cloud services. (p. 1)
  - The work in this paper seeks to unravel the issue of toolbased (practical) cloud security monitoring by discerning the approaches and tools for enabling users attain security visibility in the cloud. It augments existing literatures in the area of cloud security by using a systematic approach that reflects on real-life requirements to help cloud users address one of the most pressing concerns associated with gaining visibility. (p. 1)
  - The task of security monitoring in the cloud is more complex owing the fact that information of different granularity is aggregated from heterogeneous components dispersed across multiple levels at different time intervals. In our view, challenges to cloud security monitoring include: [. . . ] (p. 2-3)
  - Integrated monitoring - Users of cloud services usually opt to move

some applications to the cloud while other applications are hosted in-house. The challenge in this respect is the ability to integrate monitoring and correlate data from different environments. (p. 3)
  - Positioning of probing agents - This involves identifying the appropriate location within the cloud layers where probing agents could interpret and execute an underlying policy to detect events of interest. (p. 3)
  - Layers of Security Monitoring in Cloud (p. 3)
  - Approaches to Cloud Security Monitoring (p. 3-4)
  - Cloud Monitoring as a Service: Tools in this category are similar to the APIs discussed above, with the distinction that the backend operational processes, source code, and implementations are not made public. (p. 5) <- CMaaS upper bound is still the CSP, not cross-organizational monitoring.
  - A considerable variety of tools offering different supports that are appropriate for security monitoring have long been proposed. An effective monitoring approach ought to provide fine-grained attributes for aggregating variables affecting security condition. (p. 7)
- (Carvallo, Cavalli, Mallouli, & Rios, 2017)
  - Monitoring is a solution that is required to control the correct operation of the whole system running in a multi-cloud environment. According to the taxonomy proposed by [12,13], the term multi-cloud denotes situations were a consumer (human or service) uses multiple independent clouds, unlike Cloud Federations that are achieved when a set of cloud providers voluntarily interconnect their infrastructures to allow sharing of resources among them. A few concrete multi-cloud solutions exist, addressed in research projects like MUSA, OPTIMIS, mOSAIC, MODAClouds, PaaSAge, Cloud4SOA [6,11]. It is out of the scope of this paper to offer a complete survey of such activities. (p. 749)
  - The main goal of MUSA is to support the security-intelligent life-cycle management of distributed applications over heterogeneous cloud resources, through a security framework that includes: (a) security-by-design mechanisms to allow application self-protection at runtime, and (b) methods and tools for the integrated security assurance in both the engineering and operation of multi-cloud applications. MUSA overall concept is depicted in the figure below. (p. 751)
  - The MUSA Security Assurance Platform (MUSA SAP) fits the operation phase of the MUSA framework and it is devoted to continuously monitor and analyze multi-cloud application security with the possibility of activating automatic reactions (based on security enforcement libraries) and sending notifications (alerts and violation information) in case of detecting security issues with the ultimate objective of maintaining confidentiality and privacy of sensitive data and communications. (p. 751)
- To be able to deeply analyze security, the MUSA SAP relies on different

agents to be installed in different VMs or containers where application components are deployed [...] (p. 752-753)
- The SLO Manager is able to check measured attributes we need to assert which objectives are useful in defining an anomalous behavior or a disrespected rule. The latter is already paved since it consists in rules that are continuously checked. (p. 756)

**Continuous assessment and assessment methodologies**

**Quantitative metrics for cloud security**

- (Mireles, Ficke, Cho, Hurley, & Xu, 2019)
  - However, the state-ofthe-art technology does not provide quantitative metrics that can measure how well cyber attackers or defenders are able to adapt or update their resources over time. We call this problem measuring cyber agility. (p. 3217)
  - We propose a systematic set of quantitative metrics to measure cyber attack and defense evolution generations (or simply generations), which are cyber attack and defense updates that can be considered as "building blocks" or "atomic moves" used by cyber attackers and defenders in their operational practice. (p. 3217)
- The aforementioned dynamic view of cybersecurity metrics, which we pursue in the present paper, contrasts with the conventional static view of cybersecurity metrics as follows: the dynamic view reflects a system's evolution over a period of time, while the static view captures measurements of metrics at a certain time point or in an aggregated way. (p. 3217)
- Since the effectiveness of attack and defense generations may not be adequately reflected by a single metric, we consider a suite of metrics that measure evolution generations from multiple perspectives. These metrics may be then aggregated using an appropriate method (e.g., a weighted average). (p. 3219)
- At a high level, we consider two dimensions of evolution: timeliness and effectiveness. Timeliness reflects the time it takes to evolve new generations while effectiveness reflects impacts of these generations. However, timeliness-oriented metrics can use effectiveness as a reference, and effectivenessoriented metrics can use time as a reference. Fig. 2 summarizes these metrics and the structural relationship between them. (p. 3220)
- First, the metrics require the defender to record the network traffic and/or computer execution traces in order to measure At(Dt) in retrospect, where $t < t$ , $t = t$ or $t > t$. This may not always be feasible, especially for high speed networks that generate a large volume of network traffic or complex applications that may incur concurrent executions. Nevertheless, this appears to be the only way to measure the response to new or zero-day attacks. (p. 3229)

**Continuous cloud monitoring**

- (Campitelli, Catteddu, & Maria, 2020)
  - As users and the uses of cloud computing evolve, so must the supporting governance models – this includes the maturity of governance and risk management programs designed to review and evaluate the selection, adoption, and migration of traditional operations to Cloud Service Providers(CSPs). (p. 5)
  - This position paper serves to provide an impartial look at risk by identifying and examining gaps introduced over the last 10 years by the rapid adoption of Cloud Computing. (p. 6)
  - Regulatory compliance is a basic part of doing business. Non-compliance can be an expensive proposition; it diverts the organization's attention from normal operations, attracting scrutiny from regulators that can result in additional fines, repetitive audits, potential legal action, and reputational downturn. (p. 7)
  - The market size of public cloud computing has grown from $58.6Bin 2008 to a projected volume of $266.4B in 2020. (p. 7)
  - Hence, the design of their cloud ecosystem becomes just as important as its operational performance. These changes need to be recognized by the risk management process since it can raise both the category and level of risk. At stake is the entire enterprise's ability to meet its strategic objectives, including the ability to survive in the marketplace. (p. 8)
  - Introduction of new complexities: to create, operate, manage, measure and report upon performance in a cloud-based ecosystem, the organization's risk management system must accommodate a host of new complexities that have dramatically increased the scope and scale of the risk equation (p. 9)
  - This reality of [shared responsibility in] the cloud computing model exacerbates the traditional risk management consideration for third-party IT services. It is complicated by the factors of scale due to the automation of processes and procedures which might create issues of mistake propagation, as well as the risk of hidden interdependencies; visibility, due to the absence of evidence (required logs) to support control process operation, and performance; scope, due to the volume of cloud services and service providers in the supply chain; and continuous monitoring, due to the necessity of real-time, inline information consumption and production and alerts. (p. 10)
  - For instance, CSPs typically prohibit security scanning, penetration testing, and first-party audit by contract. Business continuity and/or disaster recovery testing is often not possible; customer security and privacy policies, including related standards and procedures such as hardening, may conflict with the service provider's efforts; access to key logs and visibility of alerts may be unattainable. (p. 10)
  - Moreover, the key contracting feature of prescriptive SLAs may not be available or measured and reported transparently to the customer if available. (p. 10)

  - While some have described these issues as loss of governance, we perceive them as a driver for a new governance model over cloud service providers. (p. 10)
  - The contractual relationship between the SaaS provider and their cloud usage is not transparent to the customer, creating a problem of information asymmetry. (p. 11)
  - Building on the concept of indirect control discussed in the previous paragraph, such a lack of visibility into the supply chain raises the simple question of how a cloud customer can perform a proper assessment of the security and privacy risks without having access to complete information – a typical issue of accountability. (p. 11)
- (Kunz, Schneider, & Banse, 2022)
  - A central challenge in cloud security certification is to evaluate if a set of measurements is sufficient to prove a pre-defined certification requirement. The goal of our approach is to quickly assess new threats, rather than complying with a pre-defined set of requirements. As such, we do not only decouple expert analyses from the application of their results, but integrate them in periodic re-assessments. (p. 1)
  - Also, configuration monitoring tools and approaches, like GMonE and Clouditor, exist. Yet, they lack the possibility to identify combinatorial CWs, since they just identify vulnerable configurations of single resources. (p. 1)
  - In summary, current literature does not sufficiently integrate regular expert-based threat analyses in the risk assessment process. (p. 2)
  - Before the continuous process starts, the risk assessment activities are prepared. For instance, relevant assets, i.e. cloud resources that should be assessed, are identified (p. 3).
  - ... the company employs several IoT Hubs that accept traffic and store messages in the blob storage. The underlying configurations that enable this threat therefore include public reachability of the IoT Hub, and its access to the blob storage (see also Figure 3). (p. 4)
  - Newly identified CWs may also be shared with collaborators by contributing them to the shared repository. (p. 4)
  - Module 2 covers the continuous application of the previously defined threat profiles to the cloud system. We describe the activities of this module on a high-level here, and present an implementation in Section IV. (p. 5)
  - Risk Calculation: The risk calculation is performed according to the initially defined risk model. In our example we multiply the threat value of an asset's protection goal with the respective impact value—both of which are within the interval of [1, 3]—resulting in a risk value in the interval of [1, 9]. This calculation is done again using Rego policies. (p. 6)
  - The inputs for the risk calculation policies are the identified threats, as well as the threat and impact values defined in Module 1. The output then contains the risk scores for all assets and their protection

goals which are calculated again using a Rego policy. This policy calculates the risk values for all assets and their protection goals according to the threat values that have been assessed before before. (p. 6)
  - Manual analyses: A limitation of our approach is that it does not completely alleviate the necessity for conducting manual risk assessments since the Module 1 activities still need to be performed periodically. (p. 7)
  - Concerning the shared threat profile repositories, cloud users may see a risk in sharing their threat profiles with others, since they reveal information about the architecture of their system. (p. 7)
  - Our approach is furthermore limited by the properties that IaC templates offer, i.e. resource configurations on infrastructure-level. Risk assessments, however, may also include other types of CWs, such as social engineering or application-level threats. Note that such threats can still be integrated into the threat profiles, e.g. by adding a social engineering weakness to every resource of the system. (p. 7)
  - The set up and maintenance of the Module 2 components can introduce an overhead in comparison to traditional security audits and infrastructure monitoring tools. The biggest overhead our approach introduces may be the maintenance of threat profiles which can be necessary for several reasons. (p. 8)
- (Majumdar et al., 2019)
  - This book covers basic to advanced (e.g., retroactive, runtime, and proactive) auditing techniques to serve different stakeholders in the cloud and to build the trust and transparency (p. 1)
  - However, there are currently many challenges in the area of cloud auditing and compliance validation. (p. 2)
  - As a very first step, an organization should define the scope of its auditing. Part of it is to identify the critical and sensitive assets, operations, and the modules in the system that deal with those assets and operations. The following step is to identify threats or nature of threats to be considered for the auditing process. (p. 3)
  - Also, the data collection phase has become more dynamic with the virtualization and multi-tenancy, which results in an increase in the amount of data to be collected. In this book, we also consider security aspects of data collection in addition to the different runtime and continuous data collection techniques of different data types. The trust model ensures that the audit data provided by a tenant is real and fresh. (p. 4)
  - For better understanding and interpretation, different correlation methods are applied on sanitized data tcategorize them. There are several techniques (e.g., call graph [67], information flow graph [11], and reachability graph [117]) to represent the audit data. Heterogeneous data is normalized by different methods, e.g., [26]. Storing this processed audit data is also an important phase especially when

dynamic cloud auditing generates a ginormous amount of data over time. (p. 4)
- Even though auditing has been in practice for years, the unique nature, such as dynamic, elastic, self-service, of clouds brings new challenges in devising security auditing solutions in the cloud environment. (p. 5)
- Runtime Policy Enforcement The traditional auditing mechanism, which is conducted retroactively, is not always useful for a dynamic environment like cloud. In other words, the state of the cloud usually changes frequently, and thus, a retroactive auditing approach cannot stop any irreversible damages to the cloud. (p. 5)
- Retroactive auditing approach (e.g., [9, 27, 28, 56, 63, 65, 112, 116, 118]) in the cloud is a traditional way to verify the compliance of different components of a cloud. Works under this approach in the cloud target a wide range of security properties that cover various cloud layers, such as data, user, and virtual infrastructure. (p. 9)
  * The works in [27, 112] have the same general objective, which is cloud auditing, but they use empirical techniques to perform auditing, whereas we use formal techniques to model and solve the auditing problem. (p. 10)
  * Gouglidis and Mavridis [41] leverage graph theory algorithms to verify a subset of the access control security properties. Gouglidis et al. [42] utilize model checking to verify custom extensions of RBAC with multi-domains [41] against security properties. (p. 10)
- Existing intercept-and-check approaches (e.g., [13, 46, 62, 69, 89, 105, 107]) perform major verification tasks while holding the event instances blocked. Works under this category cover the virtual network, user level, and software-defined network (SDN) layers of a cloud environment as discussed in the following. (p. 10)
  * Designing cloud monitoring services based on security service-level agreements have been discussed in [96]. (p. 11)
  * There are also few works (e.g., TopoGuard [46] and TopoGuard+ [105]) whichadopt the intercept-and-check approach in the software-defined network (SDN) environment. TopoGuard [46] and TopoGuard+ [105] perform the interception and enforcement to prevent topology tempering attacks in SDN. (p. 11)
- The concept of proactive security auditing for clouds is different than the traditional security auditing concept. The first proactive auditing approach for clouds is proposed in [13]. Additionally, the Cloud Security Alliance (CSA) recommends continuous auditing as the highest level of auditing [19], from which latter works (e.g., [66, 67]) are inspired. The current proactive and runtime auditing mechanisms are more of a combination of traditional auditing and incident management. (p. 11)
  * Proactive security analysis has also been explored for software security enforce- ment through monitoring programs' behaviors

and taking specific actions (e.g., warning) in case security policies are violated. Many state-based formal models are proposed for those program monitors over the last two decades. First, Schnei- der [103] modeled program monitors using an infinite-state-automata model to enforce safety properties. (p. 11)
    - Weatherman [13] is aiming at mitigating misconfigurations and enforcing security policies in a virtualized infrastructure. Weatherman has both online and offline approaches. Their online approach intercepts management operations for analysis, and relays them to the management hosts only if Weatherman confirms no security violation caused by those operations. Otherwise, they are rejected with an error signal to the requester. (p. 12)
    - Two major limitations of this proposition are: (1) the model capturing the whole infrastructure causes a scalability issue for the solution and (2) the time consuming operation-checking that should be performed on the emergence of each event makes security enforcement not feasible for large size data centers. (p. 12)
- Fig. 2.1 A taxonomy of cloud security auditing (p. 13) <- important to note missing layers for auditing (i.e. compute) and higher order management capabilities (i.e. inventory management)
- Table 2.1 (p. 14-15)
- There exist few works which support auditing of multiple requests together. For them, we mark the batch auditing feature. (p. 16)
- The active auditing feature is an active-probing-based auditing solution which does not fully rely on the cloud provider for the audit data and instead actively participates in the targeted protocol to verify certain properties. (p. 16)
- Literature review confirms need for out-of-band "auditing solution" to efficiently aggregrate and analyze results -> The key observations of this comparative study are as follows: First, there is no single auditing solution to verify multiple layers of the cloud. Therefore, today's cloud tenants require at least three different solutions to fulfill their auditing need, which might not be very usable for the tenants. Second, even though intercept-and- check approach is designed to prevent security violations, existing works under this category are not practical due to their prohibitive delay. Third, the proactive auditing approach is a promising solution to overcome the limitations of both retroactive and intercept-and-check approach. However, this approach still suffers from several practical issues, such as relying on manual efforts and limiting the expressiveness of security properties. Finally, there exist several features in the wild which significantly can improve the efficiency and accuracy of the auditing solution. However, there is a need of a unified solution with all these features at least to overcome major constraints. In the next chapters, we present solutions to the above- mentioned limitations. (p. 16)

- (Torkura, Sukmana, Cheng, & Meinel, 2021)
  - Most of these attacks are caused by customer misconfigured cloud resources e.g. over-privileged users, publicly exposed databases, and lack of audit logging (MacDonald, 2019). (p. 1)
  - However, the upper layer of cloud infrastructure (control plane), which customers are responsible for securing has increasingly become vulnerable to attacks introduced by misconfigurations and human errors. According to Gartner, 99% of cloud failures will be directly caused by customer errors, until 2025. (p. 1)
  - For example, Cloud compute and cloud networks leverage traditional security control like firewalls and Intrusion Detection Systems (IDS) (Takabi et al., 2010) to enforce detective and preventive security controls. However, these security controls are ineffective for protecting some cloud services including cloud storage and IAM. (p. 2)
  - The Shared Security Responsibility Model (SSRM) is a popular security model employed by public CSPs (AWS, 2020b; Institute, 2019b). It clearly defines security responsibilities for cloud stakeholders. The ability to detect and mitigate the attacks highlighted in the running example is the responsibility of cloud users, however, most cloud users do not clearly understand these responsibilities, or lack the tools to implement commensurate technical and organizational measures (p. 3)
  - However, there is a gap between these high-level benchmarks and commensurate low-level, technical implementation (Majumdar et al., 2019). A typical example of this gap is the lack of mapping of wide adopted security metrics to the CIS benchmarks.
  - While compliance benchmarks are good, they do not equal to security. Therefore, another category of policies called Enterprise Security Policy (ESP) are implemented. ESPs are aligned with enterprise security goals and internal policies including access control regulations for various organizational units e.g. Finance and Human Resources. (p. 4)
  - When cloud resources are already deployed on one or more cloud platforms, a discovery process is employed to enumerate all cloud resources. The discovered resources are subsequently adopted as the expected-state (or added to an already established expected-state). The discovery process leverages cloud APIs to acquire the comprehensive list of deployed cloud resources. During the discovery process (as illustrated in Fig. 3) CSBAuditor queries the cloud infrastructure for important information of the deployed resources based on the already defined object (using IaS paradigms). (p. 5)
  - Due to the rapid changes that occur in the cloud, it is imperative to deploy a control mechanism that leverages automation, hence the need for the SCC. The SCC leverages the following mechanisms ensure dynamic SecCM ... (p. 5)
  - Early detection of malicious changes is crucial as such events be-

come important IoCs, hence the need for continuous monitoring and auditing strategies (CSA, 2020). (p. 5)
  - The RCA is imperative for investigate detected changes or failures for security reasons. This is a critical requirement for security assurance, the knowledge of what went wrong? is essential for employing detective and preventive counter-measures since changes e.g. disablement of logging, might indicate on-going attacks. (p. 6)
  - However, logs are not available in real-time for public cloud systems. Log delivery takes about 20 min on AWS, and over 90 min on Google Cloud Platform (GCP). Therefore it is imperative to evolve techniques that overcome this gap that exposes a window of opportunity for attackers. (p. 6)
  - Section 3.7 should describe alerting and reporting, but only talks about alerting. (p. 8)
  - However, we assume that the CSP may be trusted for the integrity of the audit data (e.g., access policies and IAM policies) collected through API calls. (p. 10)
  - A limitation of our methodologies: reconciler pattern and state transition analysis is the inability to reverse changes against some cloud resources. These resources include databases, object storage. Essentially, advanced efforts are required to achieve these since the actual content cannot represented in the expected-state using either IaC or IaS. (p. 17)
  - CSPM is a relatively new category of security tools and there are very few implementations. Most CSPM are commercial and proprietary therefore it is quite difficult to compare these tools with CSBAuditor for evaluation reasons. Similarly, traditional security tools like firewalls and IDS are much easier to evaluate since their are public data sets available for conducting experiments. This is not the case also for CSPM. These two major factors pose challenges for deeper and comparative evaluation. (p. 19)
- (Weir & Aßmuth, 2024)

**FedRAMP**

- (Lewis Commission, 2025)
  - "Fully automate FedRAMP processes. Eliminate written, qualitative assessments and document review in favor of automated security controls that can be verified through the continual assessment of network security telemetry. Rather than requiring cloud service providers (CSPs) to provide proof that they have met control requirements that cannot be automated and have those artifacts reviewed, CSPs should be allowed to 'attest' that they have met requirements for areas such as personnel training and documentation." (p. 2)
  - "The slow pace of government cloud adoption increases both costs and risks for federal operations and hinders the provision of better services

to citizens. Cloud technologies offer advantages in modernization, efficiency, cybersecurity, and resilience. An agency can access and use cloud computing resources without having to buy, maintain, or modernize hardware—actions that are problematic given cumbersome federal budgeting and certification processes." (p. 3)
  - "Despite the potential gains, law, regulation, policy, and agency culture all remain obstacles to reaping the benefits of federal cloud adoption." (p. 3)
  - "To place FedRAMP on more solid legal footing, in 2022 Congress passed the FedRAMP Authorization Act. The act strongly encouraged automation and continuous monitoring to speed up the accreditation process." (p. 5)
- (FedRAMP, 2025)
- (Gartner, 2024)
- (stackArmor, 2024)


Aldribi, A., Traore, I., & Letourneau, G. (2015). Cloud slicing a new architecture for cloud security monitoring. *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 18–22. IEEE. https://doi.org/10.1109/pacrim.2015.7334802

Campitelli, V., Catteddu, D., & Maria, J. D. (2020). *CSA's perspective on cloud risk management*. cloudsecurityalliance.org. Retrieved from https://cloudsecurityalliance.org/artifacts/csa-s-perspective-on-cloud-risk-management

Carvallo, P., Cavalli, A. R., Mallouli, W., & Rios, E. (2017). Multi-cloud applications security monitoring. In *Green, pervasive, and cloud computing* (pp. 748–758). Springer International Publishing. https://doi.org/10.1007/978-3-319-57186-7_54

FedRAMP. (2025). *FedRAMP CSP authorization playbook*. Retrieved from https://web.archive.org/web/20250413105351/https://www.fedramp.gov/assets/resources/documents/CSP_Authorization_Playbook.pdf

Gartner. (2024). *Gartner forecasts worldwide public cloud end-user spending to total $723 billion in 2025*. Retrieved from https://web.archive.org/web/20250415023404/https://www.gartner.com/en/newsroom/press-releases/2024-11-19-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-total-723-billion-dollars-in-2025

Ismail, U. M., Islam, S., & Islam, S. (2016). Towards cloud security monitoring: A case study. *2016 Cybersecurity and Cyberforensics Conference (CCC)*, 8–14. IEEE. https://doi.org/10.1109/ccc.2016.8

Kuerbis, B., & Mueller, M. (2023). Exploring the role of data enclosure in the digital political economy. *Telecommunications Policy*, *47*(8), 102599. https://doi.org/10.1016/j.telpol.2023.102599

Kunz, I., Schneider, A., & Banse, C. (2022). *A continuous risk assessment methodology for cloud infrastructures*. arXiv. https://doi.org/10.48550/ARXIV.2206.07323

Laurie, B. (2014). Certificate transparency: Public, verifiable, append-only logs. *Queue*, *12*(8), 10–19. https://doi.org/10.1145/2668152.2668154

Lewis Commission. (2025). *Faster into the cloud: Accelerating federal use of cloud services for security and efficiency*. Center for Strategic; International Studies. Retrieved from https://web.archive.org/web/20250116234900/https://csis-website-prod.s3.amazonaws.com/s3fs-public/2025-01/250115_Lewis_Cloud_Commission.pdf

Majumdar, S., Madi, T., Wang, Y., Tabiban, A., Oqaily, M., Alimohammadifar, A., ... Debbabi, M. (2019). *Cloud security auditing*. Cham, Switzerland: Springer Nature. Retrieved from https://link.springer.com/book/10.1007/978-3-030-23128-6

Mireles, J. D., Ficke, E., Cho, J.-H., Hurley, P., & Xu, S. (2019). Metrics towards measuring cyber agility. *IEEE Transactions on Information Forensics and Security*, *14*(12), 3217–3232. https://doi.org/10.1109/tifs.2019.2912551

Rescorla, E. (2023a). *A hard look at certificate transparency: CT in reality*. Retrieved from https://educatedguesswork.org/posts/transparency-part-2/

Rescorla, E. (2023b). *A hard look at certificate transparency, part i: Transparency systems*. Retrieved from https://educatedguesswork.org/posts/transparency-part-1/

Rescorla, E. (2024). *Why it's hard to trust software, but you mostly have to anyway*. Retrieved from https://educatedguesswork.org/posts/ensuring-software-provenance/

Soveizi, N., & Turkmen, F. (2023). *SecFlow: Adaptive security-aware workflow management system in multi-cloud environments*. arXiv. https://doi.org/10.48550/ARXIV.2307.05137

stackArmor. (2024). *How much does FedRAMP compliance cost?* Retrieved from https://web.archive.org/web/20240808151743/https://stackarmor.com/how-much-does-fedramp-compliance-cost/

Torkura, K. A., Sukmana, M. I. H., Cheng, F., & Meinel, C. (2021). Continuous auditing and threat detection in multi-cloud infrastructure. *Computers & Security*, *102*, 102124. https://doi.org/10.1016/j.cose.2020.102124

Weir, G. R. S., & Aßmuth, A. (2024). *Strategies for intrusion monitoring in cloud services*. arXiv. https://doi.org/10.48550/ARXIV.2405.02070