Section: PUBP-6727

Mutual Monitoring in the Cloud
Progress Report 3

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 an automated build pipeline for the architecture specification as a website and downloadable document. 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, in 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 the final release and incremental draft copies of the architecture specification to separate release environments. The latter is helpful to streamline feedback with multiple advisors with different copies of changes in parallel. I will also leverage this tooling and infrastructure to manage 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 convert the website content into a downloadable PDF document.
   - 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 architecture specification webpage into a PDF document and cross-link to it from the webpage to better suit the preferences of several of my advisors.
2. I wrote multiple sections of the architecture specification draft. Due to the time spent troubleshooting, I have several more sections to complete.

3. For authoring the specification and preparing to develop the core of the prototype, I reviewed the sources below in more detail, modifying the outline and specification 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)
4. I began the development of initial framework code and test harness for the primary component of the data service, but did not complete full development of the transparency service code.
5. I continued work on outline for the critical analysis of FedRAMP, but it is not yet complete.

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. I will have shift focus to complete and start milestones previously scheduled for weeks three and four due to the troubleshooting described above.

1. Ramp up development of submission API for cloud service providers and external third-party auditors.
2. Complete the architecture specification in full.
3. Complete outline of FedRAMP critical analysis.
4. Start draft of FedRAMP critical analysis and potentially complete it so I can 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 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 professionals do not formally criticize FedRAMP and other programs that require continuous monitoring "on the record" with their name and affiliation in peer-reviewed journals or in the mainstream media. 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
W3 (May 26 - Jun 1)	Implement data service internals and submission API.	Deferred
W3	First draft of data service architecture specification.	Deferred
W4 (June 2-8)	Implement data service internals and submission API.	In Progress
W3 (May 26 - Jun 1)	First draft of data service architecture specification.	In Progress
W4	Continue outline of FedRAMP ConMon critical analysis.	In Progress
W4	Finalize architecture specification with advisors' reviews.	Deferred
W5 (June 9-15)	Implement data service client to submit to submission API instances.	In Progress
W5	Complete data service internals and submission API.	Pending
W5	Complete outline of FedRAMP ConMon critical analysis.	In Progress
W5	Complete FedRAMP critical analysis document.	Pending
W5	Finalize architecture specification with advisors' reviews.	Pending
W6 (June 16-22)	Complete data service client to submit to submission API instances.	Pending
W6	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

Summary

The project, as detailed in the intial proposal, examines the many challenges to effective multi-party security monitoring of cloud service providers and designing a solution based on two areas of work. The first area of work is an analysis of best-in-class contemporary techniques for multi-party cloud security monitoring, typified by FedRAMP's administration of their Continuous Monitoring Program. The second area of work, informed by the first, is a specification and prototype for a novel architecture for multi-party security monitoring of cloud service providers, addressing challenges and shortcomings identified from the first work area.

To evaluate the solution, I plan to use a multi-disciplinary approach to assess the project's final deliverables, identifying benefits to the proposed solutions; confirm and discover limitations to the solution; and propose future areas of work. I categorize this multi-disciplinary approach into qualitative and quantitative methods, which I describe in more detail below.

Quantitative Methods

Evaluating this project will include the quantitative methods below.

1. Model the range of costs for continuous monitoring process, data access, and submission for FedRAMP's current requirements.

2. Model the range of duration for processes related to continuous monitoring activities for FedRAMP's current requirements.

3. Model and estimate the equivalent processes in the proposed mutual monitoring architecture.

Qualitative Methods

Evaluating this project will include the qualitative methods below.

1. Identify challenges and obstacles to current FedRAMP continuous monitoring processes through resources including. but not limited to:

   - literature review of multi-party security monitoring of cloud service providers, as FedRAMP and other regulatory frameworks implement it;

- sentiment analysis FedRAMP's official forum for its 20x modernization program, in which stakeholders often critique current processes.

2. Use data from 1 to identify features and use cases of the mutual monitoring architecture to address identified challenges with a qualitative analysis of their positive or negative impact.

3. Interview stakeholders with different roles in FedRAMP authorizations and continuous monitoring. Participants will answer questions regarding the relevance and impact of challenges identified and benefits of the mutual monitoring solution's features to their work. The final report will summarize qualitative analysis of their answers to model how a mutual monitoring ecosystem will benefit the persona the stakeholder represents in the ecosystem.

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:** 7dd0d317fdb3e0d8ed99657efb34d1c3efe02fa2
Modified at: 2025-06-09

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.

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 intervals. 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.

Conventions

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.

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 in mitigating threats. 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.

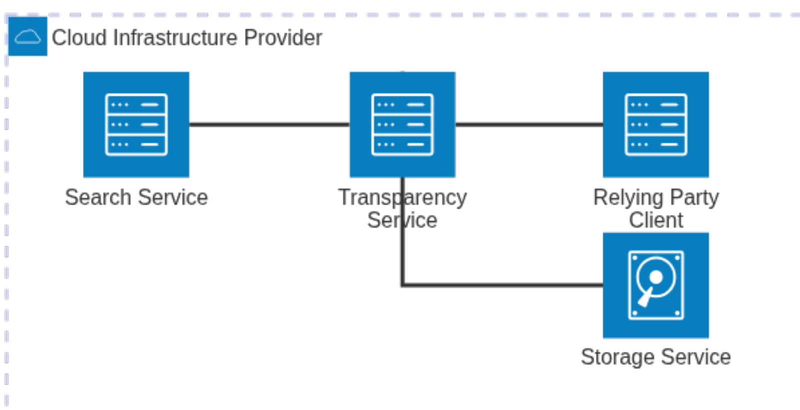
Cloud Infrastructure Customer A customer of a cloud infrastructure uses the cloud infrastructure provider as a dependency to build their own application services or derivative cloud infrastructure, thereby creating its own need for an configuration management system. By virtue of this architecture, the 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 The Transparency Service is the core component of the mutual monitoring architecture. An implementation MUST conform to the normative requirements in the current draft of the IETF SCITT Architecture (Birkholz et al., 2025). These requirements document the minimally viable features, listed below, for a Transparency Service to function for the mutual monitoring use cases documented above.

1. Transparency Services have an Append-Only Log of Signed Statements in order of Registration so one or more instances can maintain their integrity and prevent equivocation or other forms of misuse.
2. Transparency Services have a Registration Policy API with endpoints for any Relying Party to determine signing and Claim requirements before Registration.
3. Transparency Services have a Submissions API with endpoints for an Issuer to complete Registration of a Signed Claim.
4. Transparency Services have an Entry API with endpoints for any Relying Party to retrieve one or more entries previously registered with in the Append-only Log.

For a fully conformant implementation, Transparency Services for Mutual Monitoring MUST implement minimally required API endpoints in the SCITT Reference API specification draft (Birkholz & Geater, 2025).

Append-only Log The foundation of the Transparency Service is the Append-Only Log. The Append-ony Log is a sequence of Signed Statements that completed Registration through the Submission API and are accessible to a Relying Party from the Entry API. The append-only characteristic is integral to providing the integrity of individual Signed Statements, but the sequence itself. To do so, a Transparency Service needs to serialized Signed Statements in the Append-Only Log with a Verifiable Data Structure.

The Verifiable Data Structure, and supporting algorithms, for serializing data MUST use allow only for Append-only updates that do not permit reordering; enforce Non-equivocation for the Append-only Log; and allow Replayability so any Relying Party can consume the Append-only Log's data and check individual Statements or the full sequence (Birkholz et al., 2025). Transparency services instances for mutual monitoring MUST implement the Verifiable Data Structure the IETF COSE Working Group specifies in its draft specification for COSE Receiptis (2025).

Registration Policy API

Submissions API

Entry API

Adjacent Service for Storage

Adjacent Service for Search

Actors and Roles

Flows

Terminology

- Append-only Log: This document uses the normative definition from the IETF SCITT Architecture (Birkholz et al., 2025).
- Equivocation: This document uses the normative definition from the IETF SCITT Architecture (Birkholz et al., 2025).
- Issuer: This document uses the normative definition from the IETF SCITT Architecture (Birkholz et al., 2025).
- Non-equivocation: This document uses the normative definition from the IETF SCITT Architecture (Birkholz et al., 2025).
- Registration 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).
- Replayability: This document uses the normative definition from the IETF SCITT Architecture (Birkholz et al., 2025).
- Signed Statement: This document uses the normative definition from the IETF SCITT Architecture (Birkholz et al., 2025).
- Transparency: This document uses the normative definition from the IETF SCITT Architecture (Birkholz et al., 2025).
- Transparency Service: This document uses the normative definition from the IETF SCITT Architecture (Birkholz et al., 2025).
- Verifiable Data Structure: 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/

Birkholz, H., & Geater, J. (2025). *SCITT reference APIs* (Internet Engineering Task Force SCITT Working Group, Ed.) [Internet-Draft]. Retrieved from https://www.ietf.org/archive/id/draft-ietf-scitt-scrapi-04.html

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/

Steele, O., Birkholz, H., Delignat-Lavaud, A., & Fournet, C. (2025). *COSE receipts* (Internet Engineering Task Force COSE Working Group, Ed.) [Internet-Draft]. Retrieved from https://datatracker.ietf.org/doc/draft-ietf-cose-merkle-tree-proofs/14