aembit

E-BOOK

A Deep-Dive Into Secure Workload Access and the Aembit Workload IAM Platform

Break free from the endless cycle of chasing down credentials, scanning code for secrets, and rotating keys manually.

Contents

Part 1

Introduction

Remember when passwords were considered the pinnacle of security, despite commonly being scribbled on sticky notes and stuck to computer monitors in cubicles across the world? Many IT admins of the 2000s are likely still enduring sleepless nights.

Thankfully, we have course-corrected. From password managers, multifactor authentication, and single sign-on to full-blown identity and access management (IAM) and Zero Trust for human identities, today's organizations are in a much more advantageous position to safeguard user identities.

Yet here we are, primed for a deja vu scenario: With the rise of cloud, microservices, and a slew of modern DevOps practices, workload identities are skyrocketing. Meanwhile, most enterprises have merely dipped their toes (as you'll learn below) in deploying dependable secure workload access.

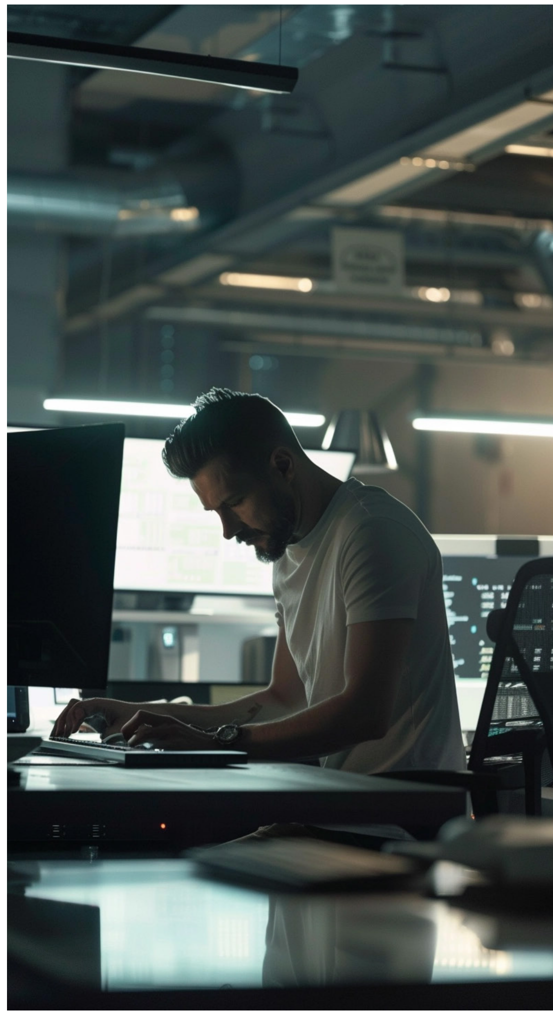
Fortunately, you're not starting from scratch. The strategies and tools that have evolved in the user IAM space, including conditional access, real-time dynamically issued credentials, and the implementation of rigorous access policies, offer a blueprint on which to build. These approaches, along with the real potential integrations that can deliver behavior analytics and anomaly detection, provide a solid foundation for extending security measures into the world of workloads.

This paper will guide you through leveraging these principles via workload IAM to manage and secure application-to-service access effectively,

ensuring your organization's resilience and agility in the face of workload access threats.

Specifically, we'll show you how to use the Aembit Workload IAM Platform to:

- **Establish secure communication channels** between your workloads, helping to ensure data integrity and prevent leakage incidents and unauthorized access.
- **Automate access management processes** to rotate credentials, streamline workflows, and improve operational efficiency.
- **Implement granular access controls** to grant the right level of access to each workload, minimizing risk and optimizing resource utilization.
- **Integrate Zero Trust architecture** principles to ensure continuous verification and minimal trust assumptions across all network transactions, **alongside the implementation of Secret Zero** techniques to securely manage the initial secrets required for system authentication.
- **Demonstrate mature secure workload access** with broader organizational goals, such as digital transformation, risk management, and internal and external compliance.



Upsurge of Workload Identities

For the better part of the past two decades, identities within the corporate world have largely been viewed through the lens of the user. And why wouldn't that be the case? Once we cleared the days of nuisance attacks whose goals were nothing more than disruption, cybercrime got more professional, calculated, and financially motivated – and the No. 1 adversary objective became human identities, which proved easy to fool and enable access and allow malicious hackers to not only infiltrate a target, but elevate privileges once inside so they can inflict maximum harm.

While identity-based attacks are still very prominent – and have largely focused on the individual – a gradual shift toward targeting non-human identities is occurring, and this is only expected to become more pronounced. That's because workload identities are running laps around human identities, outnumbering them by as many as 45:1.

The push toward distributed microservices, SaaS services, and automation within the modern enterprise means more systems and services need to interact without human intervention. This necessity demands more credentials for workload-to-workload communication, expanding the potential attack surface.

Modern IT architectures, including cloud, microservices, and containerized environments, are inherently complex. Each service or microservice requires credentials to communicate with other services.

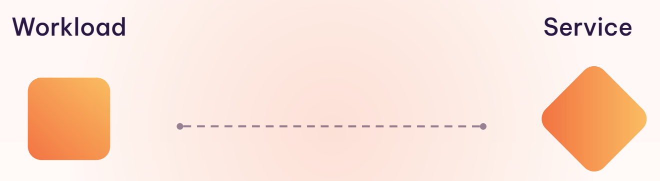
This transformation is not without its consequences, setting in motion a series of cascading effects:

→ **The velocity and scale at which new workloads are deployed are escalating.**

→ **Consequently, the frequency of workload-targeted attacks is on the rise.**

→ **This surge exposes a critical gap: the lack or makeshift nature of tools to effectively counteract these threats.**

Such developments raise pivotal questions about an organization's capacity to safeguard its operations, fulfill customer expectations, and comply with regulations, requirements, and standards. So, how are our workload identities growing at such mind-boggling rates?



Defining Workloads and Service Accounts

Workloads are, pardon the partial pun, your workhorses. They serve as the digital engines that power your business. Client workloads are software applications that access services provided by server workloads. Examples may include custom applications, daemons, ETL processors, one-off or batch jobs, and deployment scripts. Typically, client workloads run as background processes and need to access services on their own behalf, without user intervention. Meanwhile, server workloads are software applications that serve requests from client workloads. Examples may include custom web APIs, third-party SaaS APIs, API gateways, databases, and data warehouses.

Workloads, as automated entities, rely on service accounts for authentication and interaction with other resources. These service accounts, essential for the seamless operation of workloads, can inadvertently introduce vulnerabilities if not properly managed. Due to their extensive privileges and potential for shared access among multiple workloads, service accounts become critical points of concern within the workload ecosystem, emphasizing the need for robust access controls and identity management.

What Makes Workload Identities Targets

Workloads are unique in that they are non-human actors, often automated, which engage with both internal and external systems. Unique traits include:

- **No Human Intuition -** Applications and services lack human judgment and can't identify 'suspicious' activities on their own, making them vulnerable if not properly secured.
- **Speed and Volume -** Automated processes can request access or perform actions at a speed no human could match. While this is generally good for efficiency, it can rapidly escalate the impacts of a security incident.
- **Complex Interdependencies -** Workloads often involve intricate relationships between microservices, APIs, and other applications, across cloud environments and third-parties. This creates a web of data flow and access permissions that can be challenging to secure comprehensively.

Additionally, distinguishing between workload identity and access is crucial here. Workload identity acts as the digital equivalent of a passport, confirming the entity's identity within the system. Access, on the other hand, delineates its actions and resource access, akin to the permissions granted by a visa. This separation enhances security by providing clearer permissions, bolstering identity verification, and mitigating lateral movement within the network.

Building on these points, two weaknesses come to light:

- **Highly Privileged Access -** Workloads often possess access rights with extensive privileges by default. This elevated access level, intended to facilitate seamless interactions between applications and services, can become a significant liability if exploited by malicious actors.
- **Shared Credential Exposure -** The security paradigm for workload-to-workload communication frequently involves shared credentials. However, these credentials, ideally exclusive to specific workloads, are often accessible to multiple individuals within an organization, including developers. This shared access multiplies the risk of unauthorized use or exposure, further complicating the task of safeguarding sensitive operations.

5+ Real-World Examples of Credential Exposure

For DevOps engineers and security operations professionals, credential exposure is a constant concern. Compromised credentials can grant unauthorized access to critical systems and data, potentially leading to devastating consequences. This list dives into the most high-profile, real-world security incidents involving exposed workload credentials. Each case study succinctly analyzes the vulnerability that was exploited and the mitigation efforts undertaken by the affected organization. By understanding these recent attacks, you can gain valuable insights to strengthen your own credential security posture.

Cloudflare Authentication Token Theft (February 2024)

The content delivery company encountered a presumed nation-state attack enabled by stolen authentication details – namely one service token and three service account credentials – from a previous incident impacting Okta. These keys unlocked access to Cloudflare's Atlassian project collaboration software, administrative rights to its Atlassian Jira instance via a Smartsheet application, its Bitbucket source code management, and a restricted AWS environment.

Indicator	Indicator Type	SHA256	Description
193.142.58[.]126	IPv4	N/A	Primary threat actor Infrastructure, owned by M247 Europe SRL (Bucharest, Romania)
198.244.174[.]214	IPv4	N/A	Sliver C2 server, owned by OVH SAS (London, England)
idowall[.]com	Domain	N/A	Infrastructure serving Sliver payload
jvm-agent	Filename	bdd1a085d651082ad-56f7b03e5186d14d6822bb7794715a-b8cce9d850a3caaf	Sliver payload

IOCs from the attack. Source: https://blog.cloudflare.com/thanksgiving-2023-security-incident

The attack was likely waged to collect intelligence for a future planned attack. Cloudflare admitted these credentials hadn't been rotated due to an oversight, believing they were unused. The company responded by rotating over 5,000 production credentials, all test and staging systems were segmented, forensic examinations were performed on thousands of systems, and all company systems were rebooted.

Microsoft OAuth Application Abuse (January 2024)

The software giant said it countered a sophisticated nation-state attack by the Midnight Blizzard group, leveraging malicious OAuth applications to exploit user accounts. This Russian state-sponsored group utilized password spray tactics to breach accounts without multifactor authentication (MFA), revealing a targeted approach towards user identities. Significantly, the attack also illuminated the exploitation of service accounts, a strategy that indirectly compromised workload identities by manipulating these highly privileged entities within Microsoft's ecosystem. In response, Microsoft conducted an audit of identity privileges, tightened conditional access controls, and enhanced anomaly detection to identify and mitigate malicious OAuth applications.

aembit

Sumo Logic AWS Credential Compromise (November 2023)

The prominent data analytics firm experienced a security breach when an attacker infiltrated its AWS account using stolen credentials. Although the exact method of how these credentials were obtained by the attacker has not been disclosed, it typically involves tactics like phishing, exploitation of weak or reused passwords, or perhaps previous data leaks. Despite the breach, the company asserts that its systems and customer data, which remains encrypted, were not affected. As a precaution, Sumo Logic promptly secured the compromised infrastructure, rotated potentially exposed credentials, and advised customers to reset their API keys, collector credentials, and any other shared credentials.

CircleCI Application Key Theft (December 2022)

The continuous integration, continuous delivery and orchestration platform maker faced a sophisticated supply chain attack when intruders compromised an employee's laptop, circumventing two-factor authentication to access company systems. Utilizing malware that eluded anti-virus detection, the attackers stole session cookies to impersonate the employee, escalating their access and extracting data from CircleCI's databases, including tokens and keys belonging to customer organizations. Despite the data being encrypted, the attackers also obtained encryption keys from a running process. In response, CircleCI enhanced its security measures, including mandatory secret rotations, additional authentication barriers, and plans for automatic OAuth token rotations, to bolster defenses against future breaches.

Uber Hardcoded Credentials (September 2022)

An 18-year-old attacker managed to infiltrate Uber's internal systems by first evading multifactor authentication, leading them to discover a PowerShell script on a network share. This script contained hardcoded credentials for an admin account within Uber's Thycotic privileged access management (PAM) platform, a system designed to safeguard and manage sensitive access credentials for both human and machine identities. With these credentials, the attacker gained access to a broad spectrum of Uber's critical internal services and systems, including its Amazon Web Services console, VMware vSphere/ESXi virtual machines, and the Google Workspace admin dashboard.

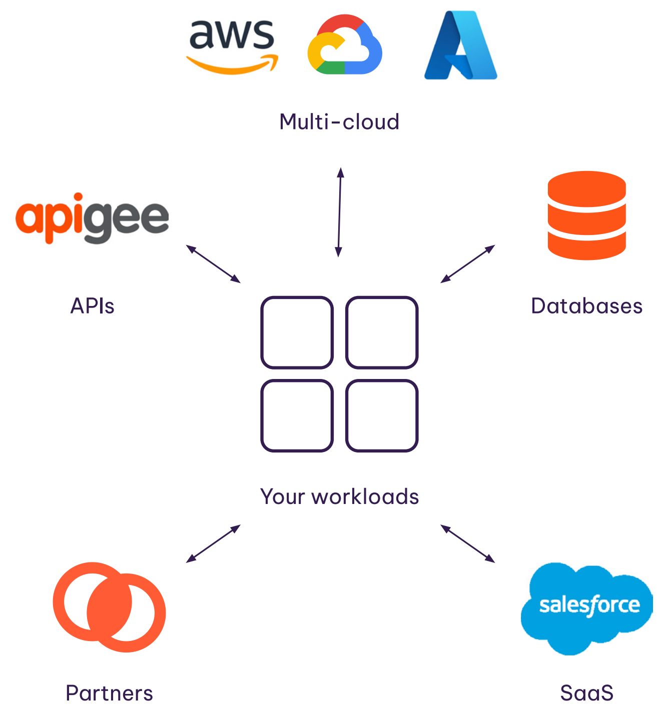
GitHub Hardcoded Credentials (Ongoing)

While not a direct breach of GitHub, the widely used platform for software development has encountered several incidents where public repositories inadvertently exposed sensitive data. This has resulted in unauthorized access to numerous companies' systems. GitHub encourages the open sharing of code through public repositories, promoting collaboration and knowledge exchange. However, this openness can lead to the accidental inclusion of sensitive information such as access keys, passwords, and API tokens in the shared codebase, contributing to the widespread issue of hardcoded secrets sprawl.

Enterprise Context and Today's Approaches for Securing Workload Access

Modern applications are highly distributed, with applications or microservices leveraging many different resources to source data and delegate operations.

For example, you might be using services from multiple clouds, databases that are distributed in multiple locations, APIs from your own applications or other services, SaaS applications (like Salesforce or Snowflake), or even partner applications.



Against this backdrop, you can see an increasing attack surface with every new connection your application makes. Operational complexity is increasing for your teams to manage it all – conducting a regular audit or performing credential rotation, for example, can become a major project.

And secrets, the method used to connect applications today, are sprawled across applications, whether they are hard-coded or managed on a team-by-team basis.

We see three methods of managing access today:

- **Cloud IAM:** This works well for controlling access to native services at a single cloud provider, but breaks down once you start crossing boundaries into other clouds, SaaS services, or even your own on-prem software.
- **Vaults:** They are good for storing secrets, but using them implicitly assumes that you understand the identity of the client requesting access. Further, they typically require effort and maintenance from your dev team to integrate into your applications.
- **DIY:** This is actually what we see most teams doing today. Do-it-yourself approaches generally include a set of manual configurations, best practices, and 'allow' lists that don't scale well and can easily fall apart during an incident.

Much like user access management has moved to user identity and access management (IAM), enterprises would like to move to workload IAM.

The challenges for enterprises typically are seen in two ways:

1. Providing an identity layer that works across environments is particularly cumbersome. If you are operating in one environment (for example, wholly in AWS or using identities within a confined Kubernetes cluster), there are native options that might allow you to achieve your goals. If, however, you are trying to cross boundaries (i.e. AWS to GCP, your cloud to SaaS, your on-prem environment to a partner's cloud), access management tools break down.
2. Enterprises would prefer to have a workload IAM platform that works for their range of environments. Even if you cleanly separate operations in your GCP and on-premises environments, for example, you still need separate ways of managing identity and access. This is operationally burdensome for your security and DevOps teams, leading to potential data protection gaps, as well as tool fatigue and reduced responsiveness.

Targeted Benefits of Workload IAM

At a broad level, the workload IAM architecture looks somewhat similar to your user IAM strategy: a third-party, independent broker that can validate identities and evaluate access policies.

Naturally, with workload IAM the workflows need to be different to work with what machines understand and can do to validate themselves. We'll get into that in more detail.

Workload IAM sits in the area between security, DevOps, and developers, and as a result provides benefits to each group:

Security

- Enforce identity-based access policies centrally.
- Perform identity-based logging for faster incident response, audit, compliance.

DevOps

- Limit service account exposure and eliminate secret sprawl.
- Automate credential rotation.

Developers

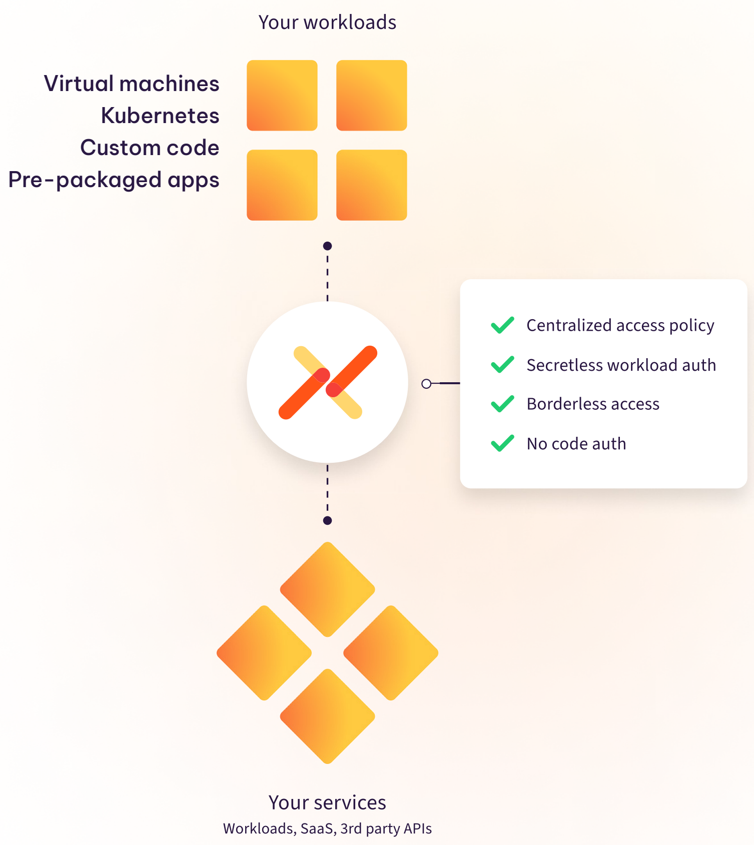
- Avoid the need to code auth.
- Maintain consistent auth implementation everywhere.
- Move to dynamic credentials with no app changes.

Aembit: A Unified Approach to Workload Identity and Access Management

Aembit has built the industry's first Workload IAM Platform to safely allow your applications to connect to any services, applications, or APIs they need. Based on identity instead of secrets, we give you a simple, secure, and central way to provide policy-based access, instead of needing to manag e low-level secrets.

Essentially, you can think of us as Okta, but for workloads. Instead of user-to-workload access, Aembit manages workload-to-workload access.

With Aembit, organizations can implement identity-first security for their workloads in the most complex of situations: Where applications and services need to reach across boundaries and establish trust through identity. Via Aembit, enterprises help their organizations in three key ways:



1. Stop Workload Attacks

Defend against common workload communication threats, including credential exposure, unauthorized access, lack of key rotation, service account sprawl, and weak or misconfigured authentication, to ensure robust security in your distributed application infrastructure.

RSAC Innovation Sandbox 2024 FINALIST

Gartner COOL VENDOR 2022

2. Build Great Products, Instead of DIY Authorization

Aembit saves hundreds to thousands of developer hours by automating the heavy lifting of workload IAM, while ensuring a consistent auth implementation across your entire environment.

3. Automate Workload Access

Save your DevSecOps teams 50% or more of the time they currently spend on workload-to-workload security. Automate and optimize activities, such as credential rotation, compliance, and audits. Aembit gives you a single place to operate, control, and analyze workload-to-workload access.

10:1

Workload identities to human identities.

Ratio that doubled in two years from 2021 to 2023.

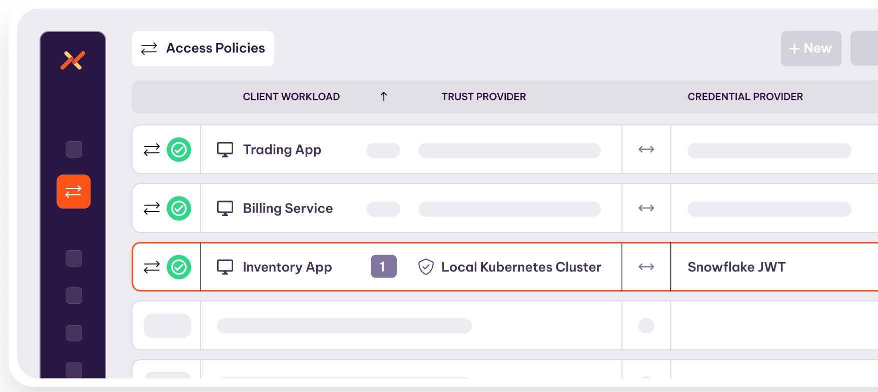
Source: Microsoft

Aembit Architecture and Approach

We will begin with an overview of the architecture at a high level. Following this, we will examine a detailed flowchart that illustrates the interaction between services during a request. Together, these elements will provide a comprehensive understanding of how Aembit functions within your infrastructure.

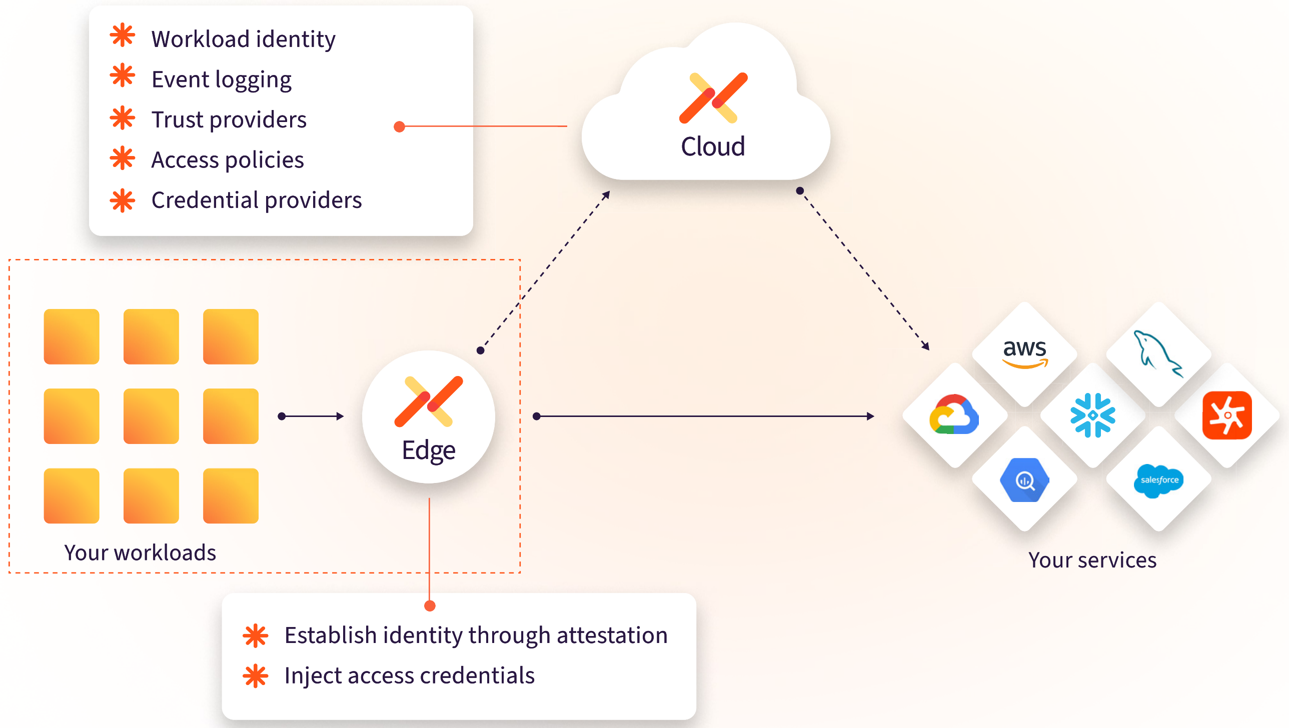
Aembit is designed to be your workload identity provider (IdP). We broker between workload identity and service credentials, using policies to control access. We do this across trust boundaries, giving you a consistent implementation everywhere you operate.

Essentially, we federate with both the workload and the service environments to make this happen. This works across the array of environments and services your business operates in today, with flexibility as you adapt and change.



CLIENT WORKLOAD	TRUST PROVIDER	CREDENTIAL PROVIDER
Trading App		
Billing Service		
Inventory App	Local Kubernetes Cluster	Snowflake JWT

> Workload IAM sits in the area between Security, DevOps, and developers, and as a result provides benefits to each group.



Aembit Architecture

- ✓ No-code auth (proxy design)
- ✓ 2-sided federation with short-lived credentials
- ✓ Centralized auth, policy-based access, and auditing
- ✓ Control-plane (not data plane) architecture

> “Aembit Edge makes no-code auth possible.

Imagine your workloads on the left – custom code that you write, or packaged applications that you run. They can be in the cloud or on-prem, running in Kubernetes, serverless, or in VMs for example. On the right are services you access – APIs, SaaS applications, apps run by your business partners, or additional cloud services.

Up top, you see Aembit Cloud. Aembit Cloud is our SaaS service that provides a centralized approach to workload identity management, workload access policies, visibility, and logging. It is operated by Aembit as a SaaS platform. We built this with a control-plane architecture in mind: Aembit manages access, but doesn’t see your sensitive application data.

On the left, Aembit Cloud federates with your workload environment through trust providers, which allow us to cryptographically validate the workload’s identity.

We do this by leveraging the Aembit Edge. Aembit Edge is a transparent proxy that you deploy alongside each of your applications for which you’d like to manage workload identity. In Kubernetes, Aembit Edge is deployed as a sidecar, on a VM as an agent. Aembit Edge both works to validate the identity of the workload and inject credentials on behalf of it.

Aembit Edge is what makes no-code auth possible. Instead of requiring your developers to modify code in their application, Aembit Edge intercepts authorization requests and injects credentials into validated requests. That means we’ll work well for greenfield but also existing apps. We just slot in and can manage credentials without app changes.

The other big benefit here is that you reduce key sprawl. Workloads don’t even need to store keys as Aembit Edge takes care of this dynamically for the app.

On the service side, Aembit Cloud federates with services through credential providers, which allow us to be trusted by the services you rely on to issue credentials that the client can use to access the service.

Finally, Aembit Cloud provides a structured logging approach that allows you to see and analyze access based on identity. This can be viewed in our console or sent to your SIEM for further correlation or alerting.

This architecture gives you a single workload IAM platform across your environment today – and in the future as you add clouds, services, or workloads.

Detailed Flow of a Workload Access Request

Let's walk through an example of a workload making a request.

1. **Client workload makes request to service.**

   Custom or packaged software that you operate makes a request to a SaaS service, API, database, or another workload. It makes the request as if Aembit doesn't exist – the workload doesn't need to know anything about Aembit.

2. **Aembit Edge intercepts client request.**

   The Aembit Edge is a transparent forward proxy, ensuring that you can continue best practices and will work for the range of your applications.

   As a reminder, Aembit Edge is deployed as a sidecar or agent and injects service credentials without workload code changes. It performs TLS decryption for HTTPS and is a multi-protocol proxy, so it will work for much more than HTTP-based apps.

3. **Aembit Edge retrieves service account token for secretless auth.**

   Aembit Edge extracts signed metadata directly from the workload's environment, serving as a robust attestation to the workload's identity. This approach eliminates the requirement for an identity secret by substituting it with secure attestation.

4. **Aembit Edge requests access credential on behalf of client.**

   Aembit Edge sends the request and the attestation metadata to Aembit Cloud.

5. **Aembit Cloud authenticates client using attestation.**

   Using a trust provider that was configured to trust the workload's environment, Aembit Cloud cryptographically validates the workload's identity.

6. **Aembit Cloud checks access policy.**

   Policies are defined in the cloud by your team to manage access between workloads. As described in the next section, this is also where conditional access policies are evaluated. It's here that Aembit also checks conditional access, which is covered in detail after this flow discussion.

“ This reduces work related to credential rotation up to 95%.

7. **Aembit Cloud requests access credential from Credential Provider.**

   A credential provider may be the service itself (i.e., we communicate directly with a SaaS service to provide a credential) or it may be a trusted provider who provides a credential on behalf of the application (e.g., vault, or an IAM system).

   Depending on the application and use case, Aembit can act as the credential provider as well, further simplifying the environment.

   Aembit uses credential providers to generate credentials (short or long-lived) for many different services.

8. **Aembit Cloud responds with policy and access credential.**

   Aembit passes the shortest-lived credentials the service supports back to Aembit Edge.

   This is particularly valuable because you can move the security posture of the communication to a shorter-lived token without asking developers to make any changes to apps.

   If you are using Aembit with a pre-packaged application, you may not have the option to change the secret type within the application. Aembit manages this transition without changes to the application.

9. **Aembit Edge injects credentials into client request and forwards it to the service.**

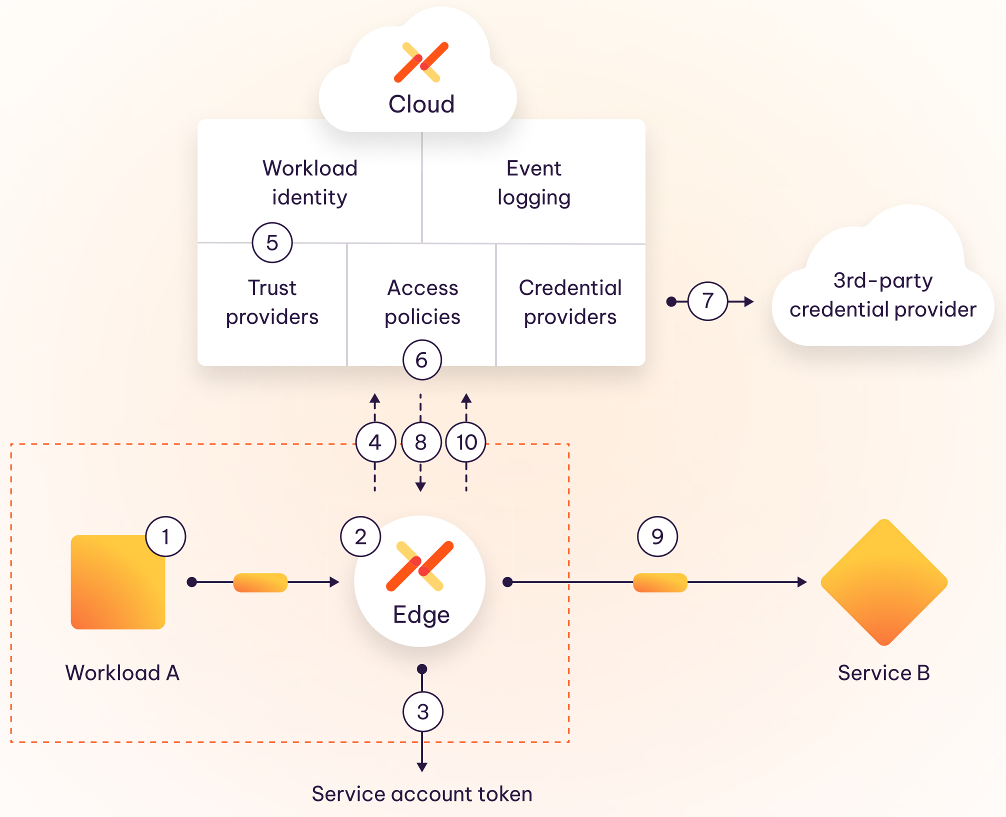
   The workload itself never sees or stores the credential. This significantly reduces key sprawl, as workloads no longer need to know about the credential. Additionally, credential rotation is dramatically simplified. While any service that depends on long-lived credentials still needs to be manually rotated, no downstream activity must happen in workloads requesting access.

   This reduces work related to credential rotation up to 95%.

   After this, communications continue between workload and service as they normally would.

10. **Aembit Edge sends access event log to Aembit Cloud.**

    The access request and access metadata are logged for analytics, auditing, and compliance. You can easily send these into a data warehouse or SIEM.



“ As we began crafting a solution to this problem, we considered User IAM as a model for how we'd like to operate: we wanted to have workload identities that could be verified upon access, with policies, and dynamically issued credentials delivered in real-time... As a result of our research, we began working with Aembit. They provide a Workload IAM system that allows us to move from managing credentials to managing access.

Cameron Tekiyeh
Seior Manager, Snowflake

Extending Aembit to Provide Zero Trust for Workloads

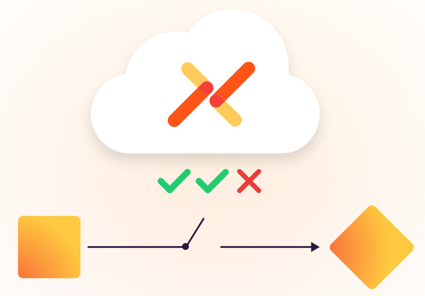
Enterprises are increasingly looking to implement a Zero Trust architecture for workload-to-workload access, much like they have already moved or are moving to Zero Trust Network Access (ZTNA) for user access to applications.

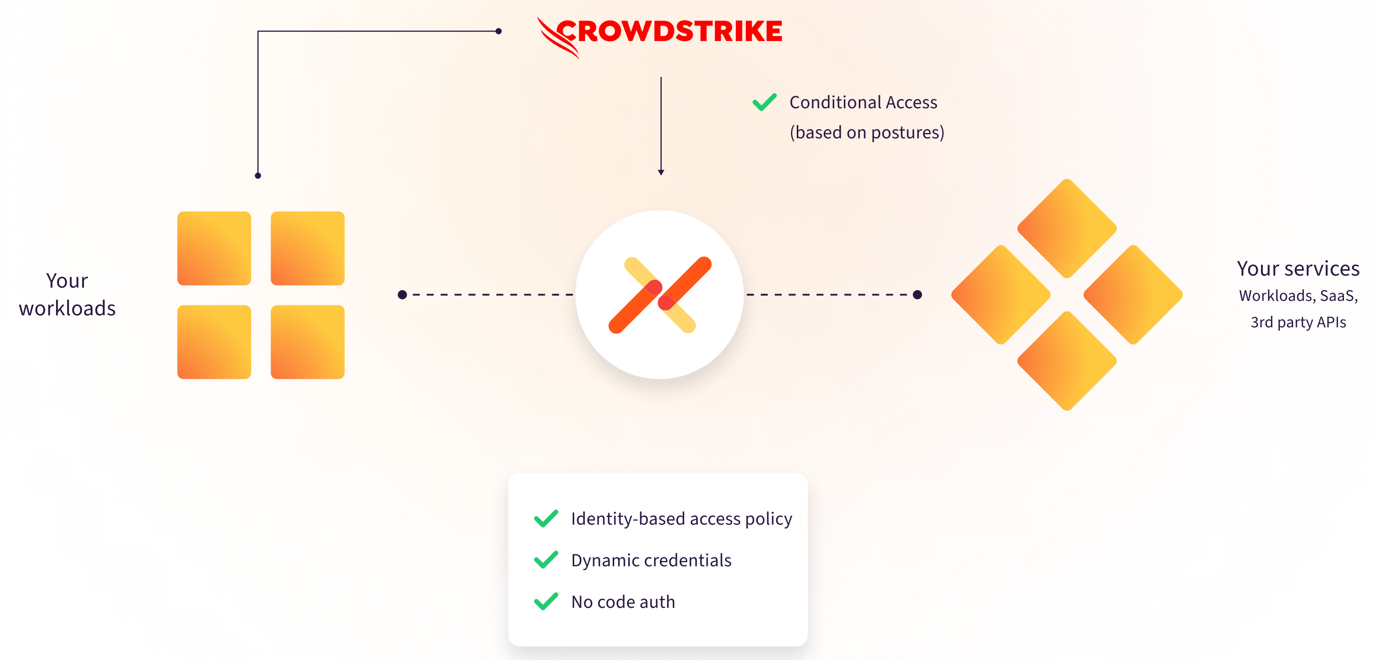
Aembit provides the most foundational elements for Zero Trust for workloads: identity + policy-based access to services. In addition, Aembit enables conditional access based on workload posture.

Not only can you use Aembit to validate the identity of an application, you can also use contextual information to determine the current state of the machine and provide conditional access based on a range of conditions.

These conditions are easily defined within an access policy.

The diagram below shows how Aembit implements Zero Trust with CrowdStrike, as an example. We do similar work with Wiz, as well as characteristics like time of day and geography.




CROWDSTRIKE
Conditional Access (based on postures)
Your workloads
Your services
Workloads, SaaS, 3rd party APIs
Identity-based access policy
Dynamic credentials
No code auth

aembit

Most Common Use Cases

Enterprises generally are taking a step-wise approach to implementing workload IAM, focusing on particular environments where they believe the need for the most amount of control immediately – and then expanding to include more resources to create a unified management environment.

We most commonly see these drivers:

Secure Data Lake and Database Access

Policy-driven access based on workload identity, consistently applied across legacy and modern environments. Applies across multiple clouds.

Workload IAM for Heterogeneous Environments

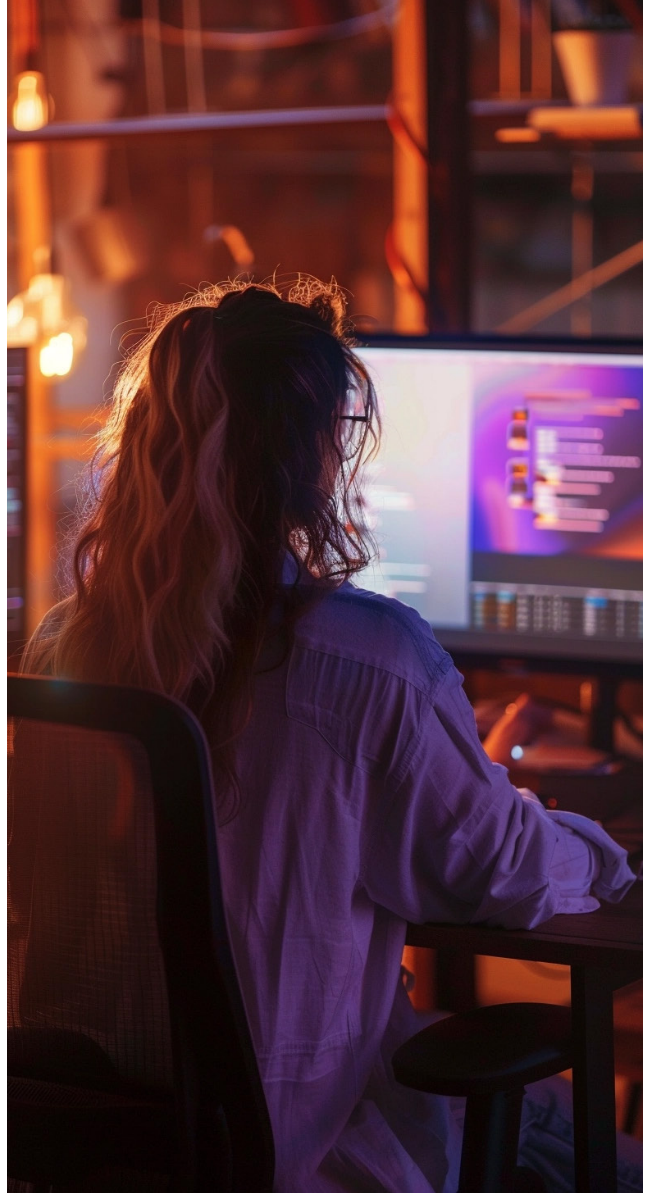
Connect legacy and modern applications across boundaries, with a consistent identity and access management model.

CI/CD Environments

Secure the dozens of connections to services required by CI/CD pipelines to automate your build process.

Zero Trust for Workloads

Allow for conditional access by checking workloads for posture conditions before access. (i.e., Crowdstrike installed with checks passing)



Typical Deployment, Packaging, and Benefits

Aembit is designed as a product that you could easily self-deploy (sign up on our website and begin yourself).

You or your colleagues need to be able to deploy Aembit Edge within your infrastructure or lab and then set up policies in Aembit via UI or API. Typically we see customers are able to deploy initial scenarios within a day.

As a way to begin rollout, we also encourage customers to focus on a particular application or set of applications that have sensitive data.

Typical examples include:

- Sensitive data such as a key database or data warehouse.
- A particular SaaS service you'd like to begin to control, such as cloud large language models.
- Strategic infrastructure, such as HashiCorp Vault or a message bus.

Our model is designed to allow you fast, cost efficient deployment:

1. Make it frictionless to start. Our free tier offers up to 10 workloads at no cost, with production-class performance.
2. The price is based on workloads so that costs are visible and predictable without complex per-request charges.

   Based on this, we are targeting a three-month return on investment for paying customers. We know it will vary depending on your environment and use cases.

We see our customers deriving benefits in three areas:

1. More robust security.
2. Simpler operations, which include less manual labor and reduced toolset.
3. Reduced operational load on your dev team. That means your teams can focus more on shipping your revenue-generating product instead of worrying about your security infrastructure.

Summary

Workload IAM is a powerful way to secure your application-to-application communications, especially where applications need to communicate across trust domains to accomplish their tasks.

Leveraging the Aembit Workload IAM Platform can simultaneously help you prevent credential loss, accelerate your development cycles, and eliminate the manual burdens on DevSecOps teams, including credential rotation, credential tracking, audits, and compliance.

We encourage you to start today with our forever-free tier, enabling you to secure up to 10 workloads at any scale.

Manage Access, Not Secrets.

Aembit is the Workload Identity Platform that lets every business safely build its next generation of applications by inherently trusting how it connects to partners, customers, and foundational services. Aembit provides seamless and secure access from your workloads to the services they depend on, like APIs, databases, and cloud resources, while simplifying application development, delivery, compliance, and audit. For more information visit **aembit.io**

aembit

aembit