WHITE PAPER

Machine Identity Management Architecture

Venafi®

Introduction

The speed at which applications and their corresponding architectures are evolving in the modern network is staggering. In just a few short years, applications have transitioned from legacy monolithic and API-wrapped service-based architectures to microservice, serverless, and edge architectures. This decoupling and granularity in the application architectures has brought speed, agility and autonomy to workloads. However, these changes also have resulted in an explosion in the number and type of machine identities that organizations now rely on, as well as new levels of complexity in managing them. This complexity can have extremely negative consequences on security, reliability and speed of development.

In modern architectures, most organizations are grappling with the challenges of a rapidly expanding set of machine identity use cases, as well as the shift from human-centric processes for deploying applications and creating compute capacity (infrastructure). Traditional PKI and InfoSec teams are under more pressure than ever before to provide solutions for a seemingly endless supply of new use cases from the business, while balancing the need for operational security, reliability and compliance. These pressures, along with other elements of digital transformation, are driving the need for a new set of required capabilities for machine identity management, including:

- High-speed issuance
- Scalability and high availability
- Multiple machine identity types
- Integrations and API flexibility
- Autonomy and freedom of choice

The need to deliver these required capabilities to contain the increasing complexity and to overcome the consequences of modernization has led to the emergence of a new approach to reducing the complexity of managing machine identities: the Modern Machine Identity Management Architecture. This modern architecture is built on new required foundational capabilities such as identity issuance, policy enforcement and observability that have evolved alongside the application architectures.

The Early Days of Machine Identity Management

The earliest machine identity management architectures were largely human-driven and were built around manual processes and rudimentary inventories. The initial goal of these preliminary architectures was to stop service outages caused by TLS certificates that had expired unexpectedly. To avoid these certificate outages, some organizations opted to use long-lived certificates, often 3-to-5 years, which prolonged the expiration dates but, in turn, conflicted with security efforts.

As organizations began to apply the aspects of people, processes and technology to machine identity management, outages declined, but the results were still inconsistent. Often, the Public Key Infrastructure (PKI) team was hamstrung with limited visibility, lack of expiration notifications and lack of accountability. In response to these challenges, vendors developed platforms and tools for expiration monitoring and notification, but machine identity ownership was still difficult to track. Capabilities that optimized enrollment, renewal, review and approval processes alleviated the burden of manual tasks but presented a black box to anyone outside of the PKI team. With the addition of discovery, visibility, reporting and self-service, machine identity management solutions were able to mitigate bottlenecks and increase operational efficiency. Even with these technical advances, machine identity management remained largely a manual people-centric process.

Then, as organizational concerns grew around confidentiality, integrity and availability, they began to apply TLS and other PKI-centric technologies to the management of other types of machine identities. At the same time, the number of machine identities that organizations had to manage exploded and, with it, the risk of outages. This acceleration has driven the need for full lifecycle automation, which has greatly increased availability and relegated outages to occur in fringe use cases when organizations use manual processes. Furthermore, capabilities, such as policy definition, enforcement and approvals helped PKI and Infosec teams meet compliance and reporting requirements.

Over time, increased technical capabilities, such as API integration and orchestration, expanded machine identity use cases faster than ever. The growth in the use of automation added pressures to machine identity management platforms—specifically for availability, scalability and speed. The traditional PKI/InfoSec group found itself pressed hard from two directions: to provide an increasingly complex service that enabled teams within the organization to innovate quickly and being able to fulfill the security and compliance requirements of organization.

The Modern Machine Identity Management Architecture

As organizations look to modernize their machine identity management architectures, it is imperative that any new technical capabilities also mirror the shift in processes that organizations employ today. These required changes relate not only to improvements in application development and deployment cycles but may also include changes to how machine identities themselves are used. For example, ephemeral certificates and the increased use of distributed subordinate issuers require a new way to architect these capabilities. And this new architecture needs to be built in a way that allows security teams to retain the control, observability and consistency they need to protect the business. While many of the technical capabilities established within existing machine identity management solutions are still applicable, many new technical capabilities are needed to support the new use cases and technical capabilities required by modern architectures. It is critical that the entire set of technical capabilities are a part of an overall "control plane" architecture that provides the control, consistency and observability across legacy and modern architectures. (Figure 1).

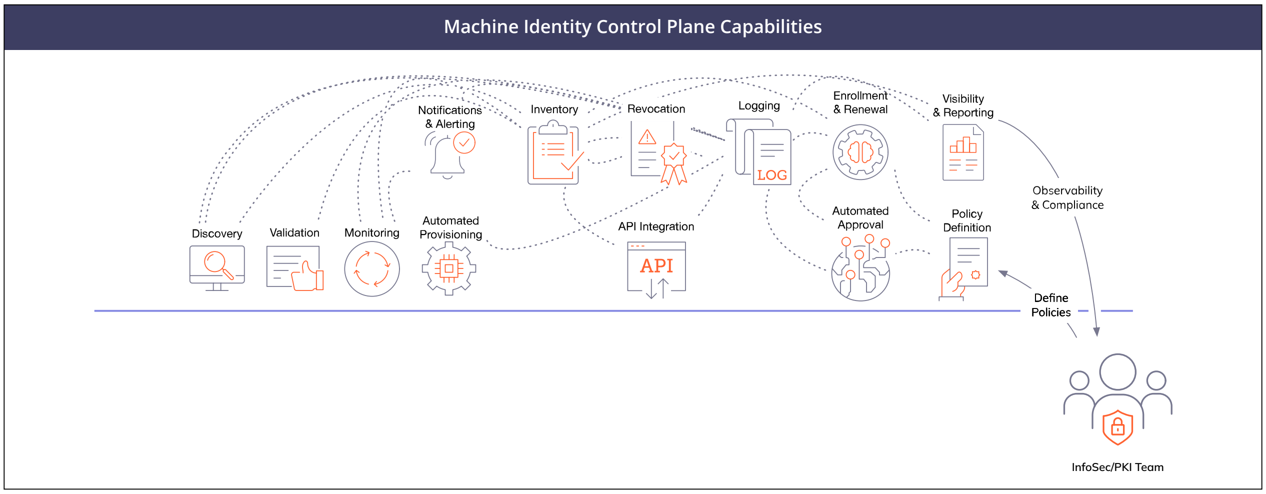

Figure 1: Required capabilities of a classic machine identity management architecture

Requirements of the new software development lifecycle

Software development is one area where we are seeing radical changes in the use of machine identities. With the development and deployment lifecycle shrinking, security considerations are now integrated into development and deployment using a Continuous Integration, Continuous Deployment (CI/CD) lifecycle and orchestration of the virtualized compute infrastructure for the automated application. The machine identity (or certificate) lifecycle should match the development cycle and this lifecycle should start when the application is deployed in any environment, including development and testing.

As software development teams deploy a constant stream of new workloads, they create new pressures on machine identity management in terms of visibility, speed and scalability. These pressures drive the evolution of machine identity management from being very human-centric to application-centric and from long duration to short duration lifespans. This applies even to ephemeral machine identities. A modern machine identity management solution should be architected to be as transparent as possible to the users and teams responsible for deploying an application.

In particular, application teams have specific responsibilities surrounding the ownership of an application, including the business requirements, the operational delivery, support for the application and its consumers, as well as the security of the application and its data. Historically, applications may have been developed by one team, tested by another, deployed by an operations team and finally secured by the InfoSec teams. Developers tend to focus on applications, not machine identities. Conventionally, machine identities have been considered relatively independent of applications. Modern machine identity management architecture changes that. It takes an application-first view. That means the first step in the machine identity lifecycle is not to “request a certificate”, but rather to define the application that it is to be used for.

An application may need one—or even several—machine identities. The application may even have different types of identities, such as a TLS certificate, a client-authentication certificate or potentially an SSH certificate. Regardless of machine identity type, enterprises still need to confirm that each one is issued appropriately and associated with the right people who are responsible for the application. Ensuring the application is properly registered allows for the validation of any machine identity requests and enforcement of applicable policies (Figure 2). A similar process is followed for secrets that may grant the application access to other enterprise resources. Typically, these would be coordinated through an Identity and Access Management (IAM) system and may also interface with the Configuration Management Database (CMDB).

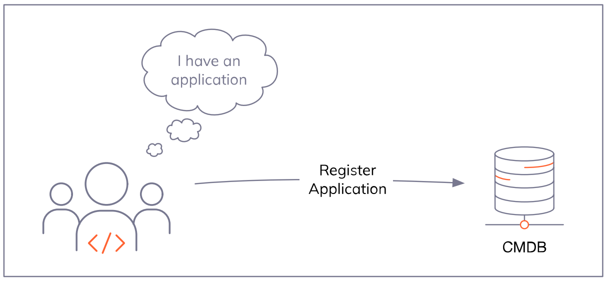

Figure 2: Registering an application to a CMDB

In modern environments, deploying an application requires that organizations define the compute infrastructure and requirements in addition to the application code itself. In this scenario, the application code (and the infrastructure-as-code) is committed to a repository from which a CI/CD toolchain reads. Part of the CI/CD infrastructure will include or interface to an orchestration system that is responsible for the deployment of any compute infrastructure and network resources including the installation or deployment of the application and its dependencies. Ultimately, the orchestration system is where applications and the machines they depend on are created within the organization’s hybrid cloud environments.

Scaling with orchestration

Whether it's an application that's being deployed or even new infrastructure that's being created in support of that application, execution is centered around the orchestration level. Here again, it's impossible for humans to stay in the loop, given the dynamic and elastic requirements for speed and scalability. The orchestration should be invoked in an automated way directly from the IT service management (ITSM) system where the application is registered. Applications and infrastructure are then deployed within the organization through this orchestration layer (Figure 3).

Machine identities need to be part of the core underlying infrastructure that's being deployed, not an afterthought. This proactive stance should be the case whether orchestration is creating cloud, cloud native or data center infrastructure and services. Taking this approach solves many of the typical issues that may have arisen around machine identity management such as availability, security, operational efficiency and compliance.

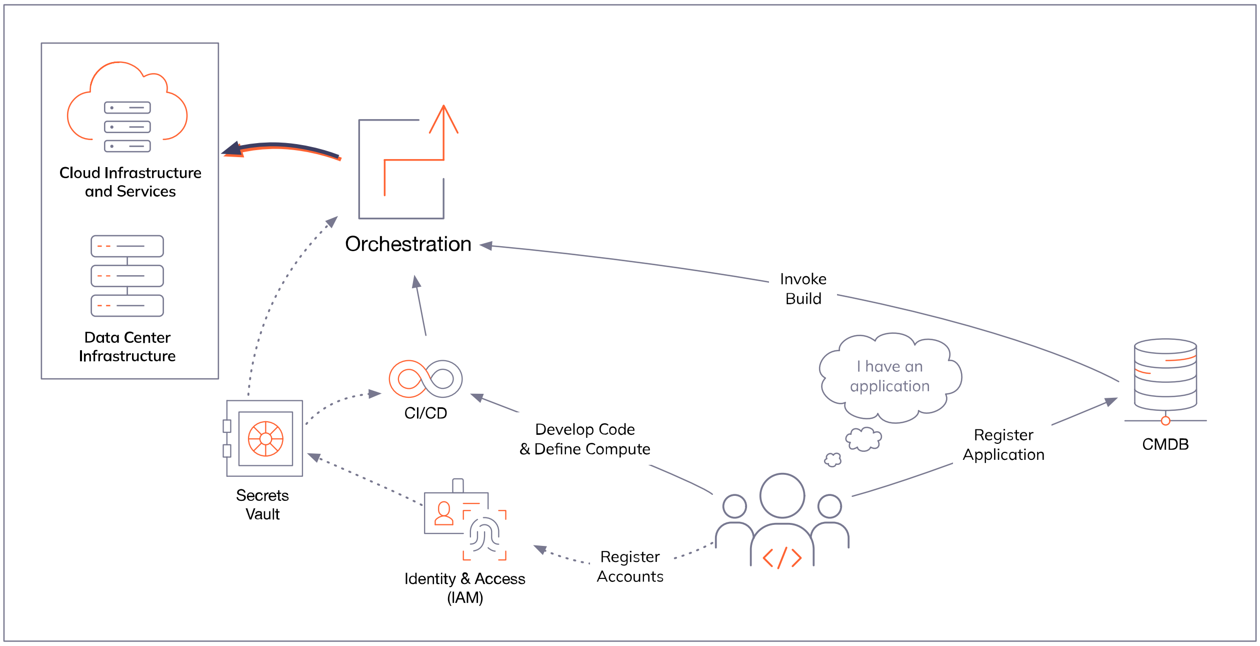

Figure 3: Orchestration invoked through an ITSM or manually

As more workloads become ephemeral, machine identities that represent those workloads have a much shorter duration as well. With orchestration arises the requirement for fast issuance of machine identities to address the autoscaling nature of applications and resources being deployed. Additionally, some organizations require that the issuance of the required machine identities be capable of existing on "edge" applications and devices, or outside typically managed data center or cloud networks. In these cases, machine identities are commonly deployed with the orchestration system or within an application itself (Figure 4). This deployment capability also enables the machine identity issuer to be incorporated in the build/test/run infrastructure, or even on a developer's system. To meet scalability requirements, the machine identity issuers also need to reliably scale with the orchestration. Just like the applications they serve, there may be a need for one machine identity issuer today and 100 issuers next week to satisfy the dynamic nature of orchestrated application environments.

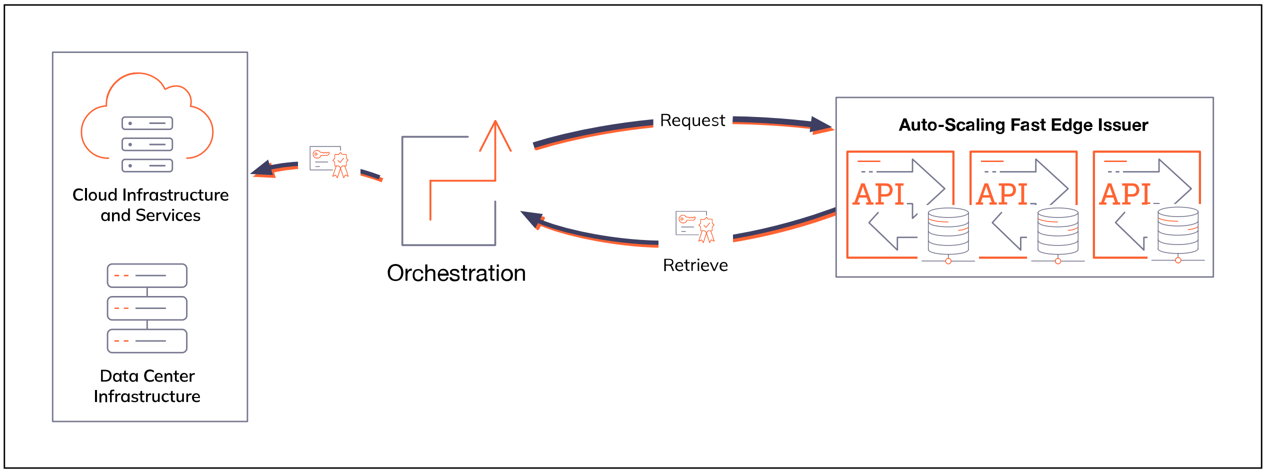
Cloud Infrastructure and Services
Data Center Infrastructure
Orchestration
Request
Retrieve
Auto-Scaling Fast Edge Issuer
API
API
API

Figure 4: Fast issuance addressing the scalability and speed requirements of autoscaling applications

Balancing requirements for control and speed

With the delegated, subordinated and highly distributed issuance of machine identities, the classic requirements of visibility, policy enforcement and consistency gain even more importance. Machine identities must be inventoried for visibility and reporting purposes and validated to make sure they're issued according to policy and operating as designed. To handle these classic requirements, the new issuers require capabilities to "bootstrap" themselves from the machine identity management platform, retrieve and enforce policies established by the PKI and InfoSec teams, but also provide visibility into what has been issued. This allows those machine identity issuers to comply with InfoSec policy. By enforcing policy at the sub-CA level, organizations can extend inventory, control and validation that correct cryptographic standards usage to even short-duration machine identities.

The new requirement for rapidly issued, scalable, short-lived machine identities now automatically becomes an extension of machine identity management and provides organizations with the best of both worlds. Fast and scalable issuance, which addresses the need for availability by operating autonomously when necessary, but also ensures that compliance and visibility remain intact (Figure 5).

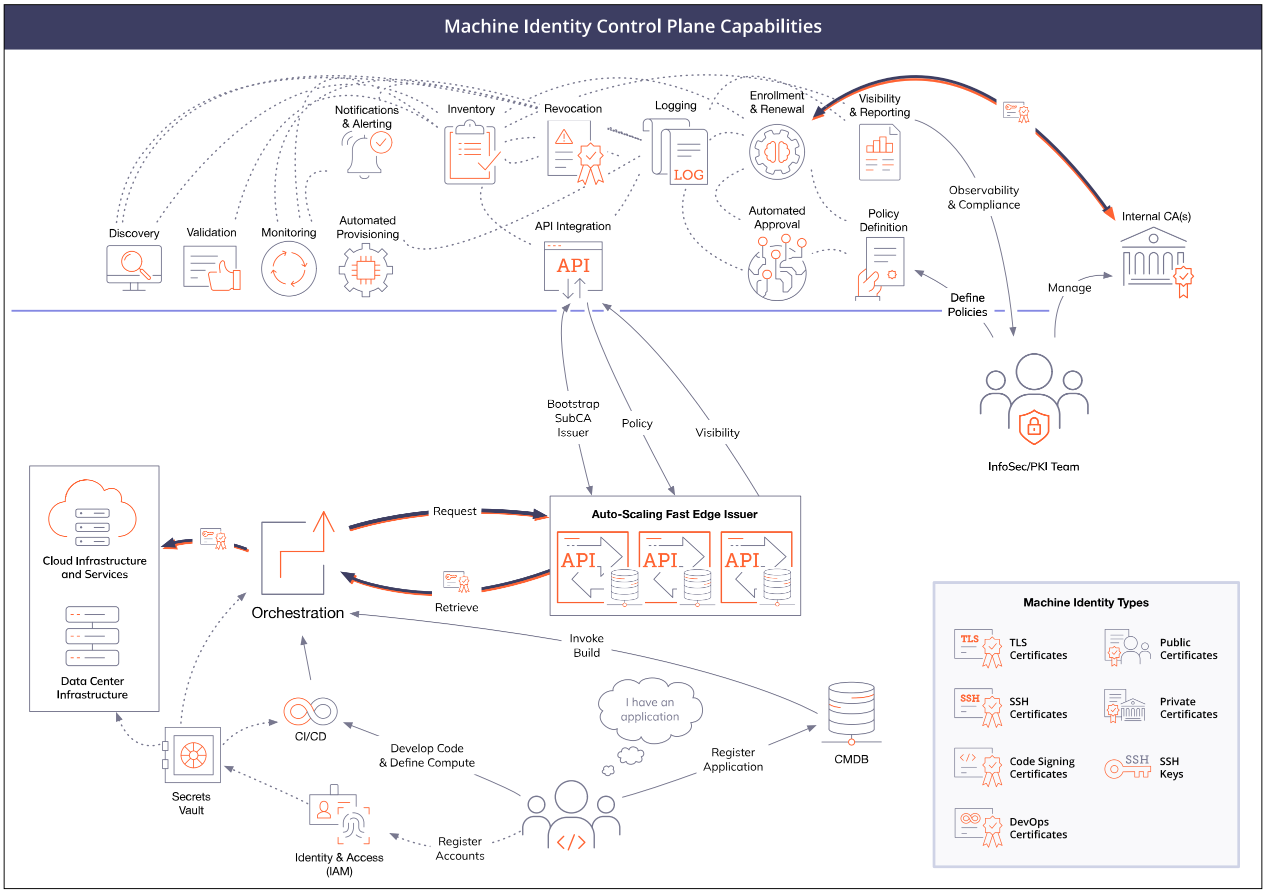

Figure 5: Fast issuance with compliance and visibility combined

Maintaining consistency across modern and traditional environments

This modern architecture should still support other use cases that may require the use of more classic automation approaches like push provisioning. For example, there could be a use case for a 1-year publicly trusted certificate, which is commonly provisioned to a load-balancer. That use case is not going to use the sub-CA issuer since public CAs are not able to be subordinated. Instead, it will leverage the enrollment capabilities of the machine identity management platform to issue from a public CA and the push provisioning capabilities for installation. Although this use case would employ a more classic approach to machine identity automation, the difference would be that the orchestration is now driving the machine identity lifecycle and management. This ensures consistency across the use cases and eliminates the need for manual intervention across the infrastructure.

Similarly, the orchestration may be designed to inject instructions into a workload (script, utility, etc.) to pull and retrieve a machine identity directly into the application or infrastructure that's being built. It is critical that flexibility is designed into a modern architecture, leaving many options for how each use case can be solved.

A complete architecture can be difficult to represent in a single picture, given the additional capabilities and myriad of use cases to be solved (Figure 6). However, it is important to recognize that the capabilities that are key to solving modern use cases leverage well-established capabilities that already exist in the machine identity management solution. This is important because the availability of those capabilities across environments also protects the organization from key misuse and compromise, enforces established policy and provides the reporting and visibility necessary to ensure compliance.

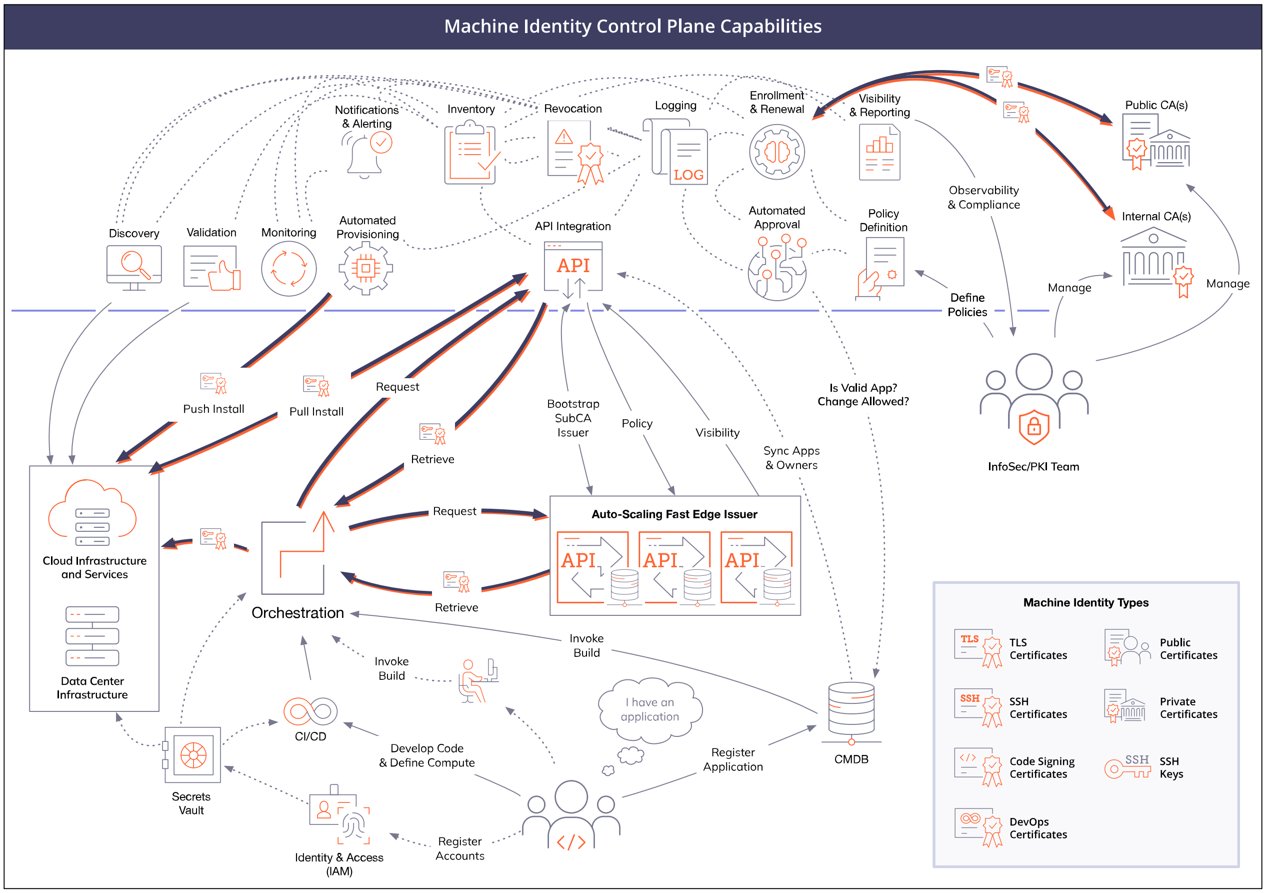

Figure 6: The Blueprint for a Modern Machine Identity Management Architecture

Conclusion

Increased complexity is one of the main consequences of adopting modern application and deployment architectures. This complexity becomes exponential when organizations are enabling trust between an elastic number of machines, many of which are ephemeral. Poorly executed and manual machine identity management can negate the expected gains that modern architecture and deployment architectures seek to achieve. Additionally, applying traditional machine identity management to modern architectures can result in a decrease in overall security posture as well as impacting the reliability of the services themselves.

The only way to handle the complexity of modern application architecture in a way that results in stronger security and resiliency is to use a control plane for machine identity management. A control plane for machine identities provides the observability, control and consistency that organizations require to reduce or eliminate the complexity of managing millions of distributed machine identities, resulting in more successful operational outcomes.

Trusted by

5 OF THE 5 Top U.S. Health Insurers
5 OF THE 5 Top U.S. Airlines
3 OF THE 5 Top U.S. Retailers
3 OF THE 5 Top Accounting/Consulting Firms
4 OF THE 5 Top Payment Card Issuers
4 OF THE 5 Top U.S. Banks
4 OF THE 5 Top U.K. Banks
4 OF THE 5 Top S. African Banks
4 OF THE 5 Top AU Banks

Venafi is the cybersecurity market leader in identity management for machines. From the ground to the cloud, Venafi solutions automate the lifecycle of identities for all types of machines—from physical devices to software applications, APIs and containers. With more than 30 patents, Venafi delivers innovative solutions for the most demanding, security-conscious organizations in the world. **To learn more, visit venafi.com.**