Application-aware Security for Cloud-native Applications in a Zero Trust Environment

WHITE PAPER

Aporeto
Cloud-Native Security

Aporeto
Cloud-Native Security

TABLE OF CONTENTS

WHITE PAPER

THE PROBLEM

Cloud-native Applications

Today it is common for cloud-native applications to utilize anywhere from dozens to thousands of servers and be composed of anywhere from a handful to thousands of containers that deliver microservices by spinning up quickly and shutting down when not needed, so as to use the least amount of computing resources across a hybrid, possibly multi-cloud architecture.
DevSecOps personnel struggle with securing modern cloud-native applications using the old network-oriented security model that has traditionally been used for monolithic applications and role-based access control. What is needed is a new security model that assigns trusted identities to each component of the application and users of the application to strictly regulate interactions in what is assumed to be a zero trust environment. Security has to move from being based on a network-oriented model to an application-aware model, and it has to be easy to implement and manage to be truly useful for DevSecOps.

The Network Oriented Security Problem

In the old network-oriented security model, a service given an IP address and uses well known ports to make itself available on the network. Access to the service is secured with firewalls, IPS, WAF, NAC, and VPN gateways that are placed in front of it and are programmed with a set of cumbersome rules and vendor-specific configuration settings to define which ports should be opened or closed and which network traffic originations should be allowed through. Syntax and rule priorities vary across the components comprising the security infrastructure, so it is rare for a single individual to understand or be able to maintain the security settings that have been put into place for the cloud-native application. As application components are

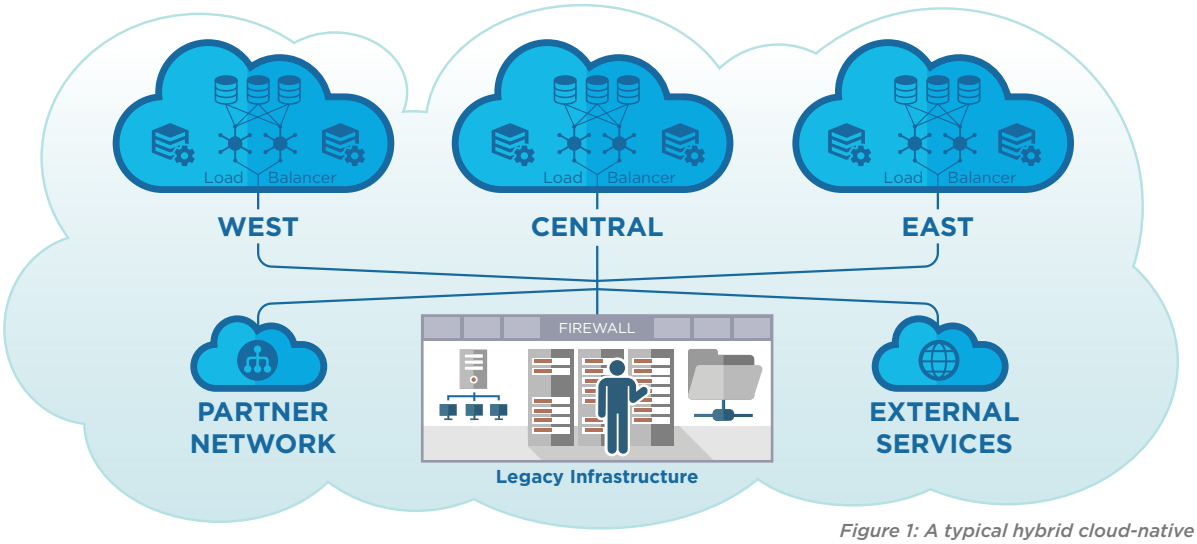

Figure 1: A typical hybrid cloud-native application

THE PROBLEM

spun up and down or moved, the number of security rules that need to be updated across the network topology becomes a management nightmare, bogging down SecOps personnel and constantly surfacing new security vulnerabilities. Another major problem with network-oriented security is that once the IP address and ports for a Web-accessible application are known, any client residing on the Web may attempt to gain access. What is needed is a model whereby the application knows who the client is, in a cryptographically secure way, so access can be granted or denied based on that information.

Security Needs to Be Application-aware

DevSecOps personnel want a security model for cloud-native applications that is more secure as well as being easier to use, understand, and maintain. Ideally it would be integrally tied to the components that comprise the application, not to IP addresses, ports and networking equipment settings. The security model cannot require developers to have to learn a new programming paradigm, because that would slow them down. Ideally the security system could monitor the application component interactions to learn how best to secure them - automating the task of establishing security.

Greater Visibility into the Security Posture

DevSecOps teams want to have better visibility into the security posture of the application. What is needed is a security platform that presents SecOps personnel with a dashboard that clearly shows how the overall application and all its component services are protected end-to-end, based on the application's components themselves and the resources they utilize, not based on firewall rules or networking settings.

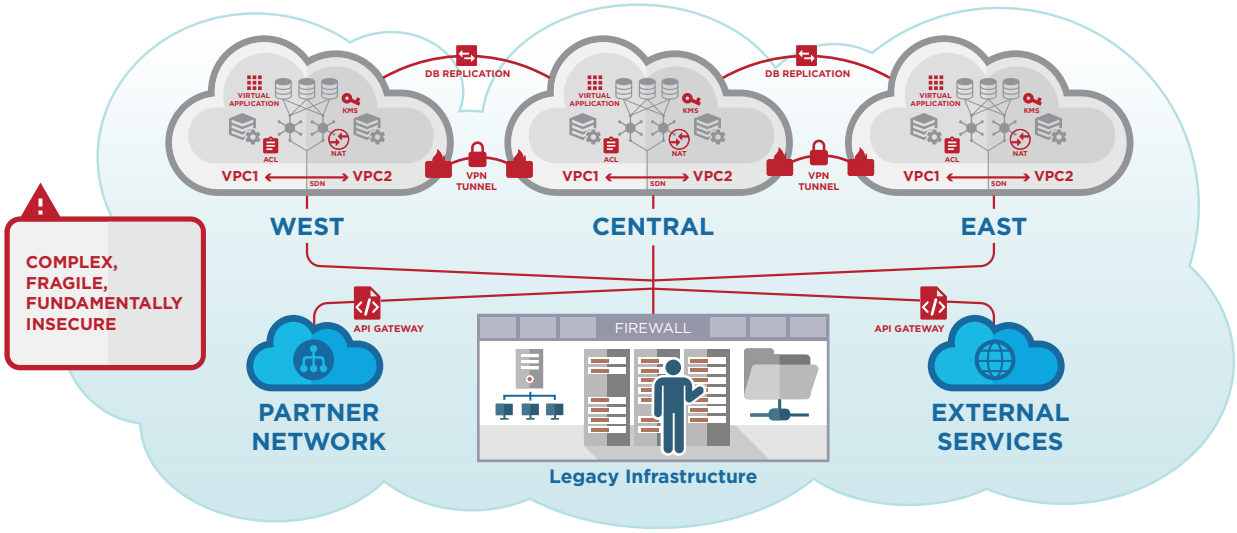

Figure 2: Complex network-oriented security for modern cloud-native applications

SOME BACKGROUND

Start with a Zero Trust Model

Forrester Research, who advise over 3500 organizations worldwide, prepared a landmark analysis on security for modern applications for NIST in which they state that "The traditional [security] mindset does not take into account the current environment", and they put forward a Zero Trust Model as a fundamentally better way to think about application security. The paper can be accessed on the NIST Website (https://goo.gl/f6mw1q). Zero Trust takes into account the possibility of threats coming from internal as well as external sources and protects the organization from both types. This is important because, as the survey results below summarize, most threats actually come from within.

Attribute Based Access Control

In the NIST Cybersecurity Practice Guide SP 1800-3, Attribute Based Access Control, which can be found at https://goo.gl/nZXyFP is pointed out that that there is a fundamental issue that arises when traditional Role Based Access Control (RBAC) is used to gate access to system resources. At the heart of the problem is that each time a user or application component arrives or leaves, an administrator must manually change access rights accordingly, a difficult and inefficient method. Using only one factor, user identity, is not as secure as a model where multiple factors are used to make the decision whether to allow or deny access.

To overcome the problems inherent in RBAC, the National Cybersecurity Center of Excellence has developed a reference design for an Attribute Based Access Control (ABAC) system. In this model, many attributes about the entity, be it a person or a service, that wants to have access to an object along with information about the object itself and relevant environmental information are all taken into account to create granular policies that are used to grant or deny access. Gartner recently predicted that "by 2020, 70% of enterprises will use attribute-based access control...as the dominant mechanism to protect critical assets, up from less than 5% today."

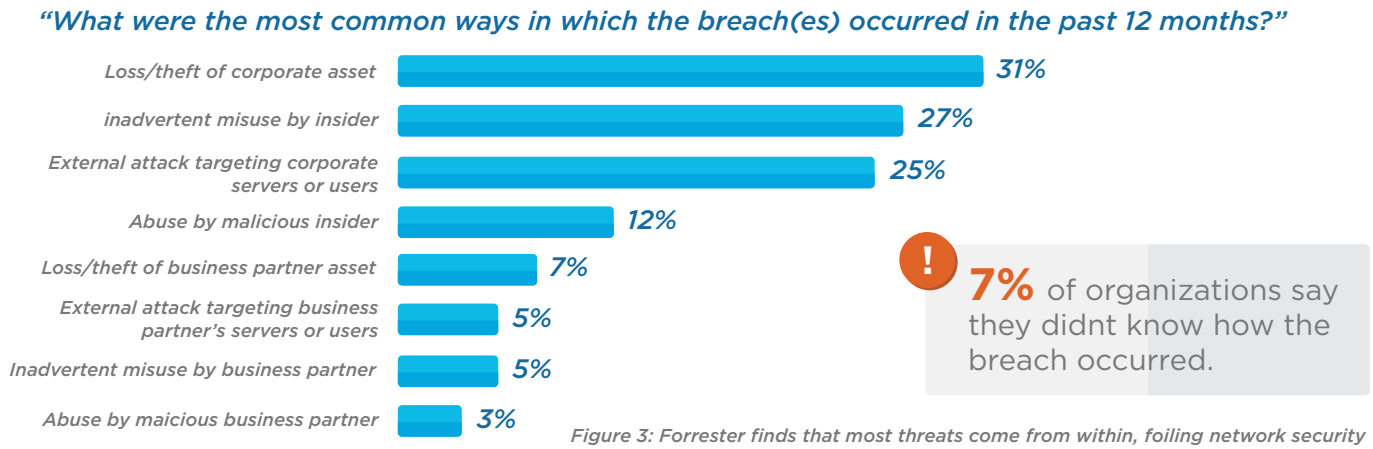

Figure 3: Forrester finds that most threats come from within, foiling network security

THE SOLUTION

Overview of the Aporeto Platform

The Aporeto Platform assumes a Zero Trust Model and uses Attribute Based Access Control (ABAC), as recommended by NIST, to provide a modern platform for enterprises that want to dramatically increase the security of cloud-native applications while significantly lowering their administrative costs, even for very large scale applications.

With Aporeto, each Processing Unit (Linux process or Docker container) is provided with a Trust Profile that is automatically created using the ABAC model, incorporating the id of the process and its host, its location, the user-id it is running under, a historical profile, and any number of other environmental or assigned attributes of the PU.

A Zero Trust environment is assumed, so all PUs must be authenticated and subsequently authorized to perform any interaction with other PUs or resources across the datacenter, cloud, hybrid cloud, or multi-cloud environment. All interactions must be explicitly permitted based on a set of Policies that are enforced by the Aporeto Enforcer that runs on each physical or virtual host.

The Aporeto Platform delivers superior security compared to the old model where security was provided by disparate firewalls, IPS, WAF, NAC, VPN gateways and other encryption products. It operates at the application-aware level, vastly simplifying and strengthening the security of modern applications. It secures cloud-native applications from attacks mounted from within or externally and its

automated operation ensures that security settings are comprehensive, always in force, current, and easy to maintain.

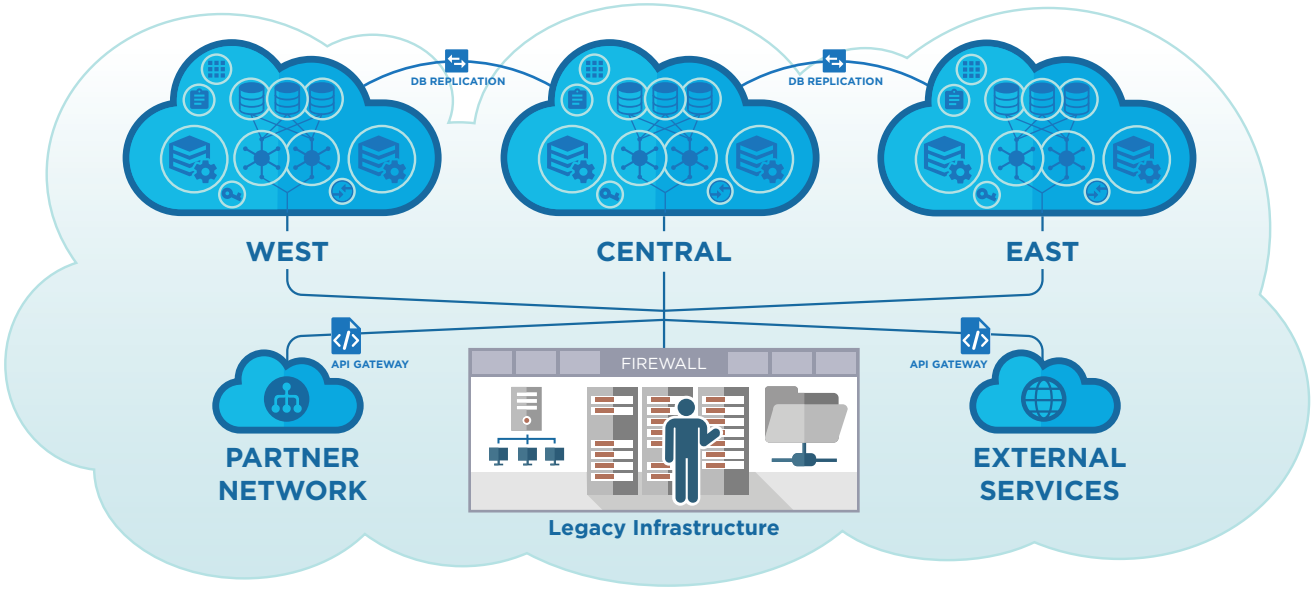

Figure 4: Intuitive application-aware security for modern cloud-native applications

HOW IT WORKS

Overall Operating Model

The Aporeto Platform is the single pane of glass to the Aporeto solution and can be offered as a SaaS product or hosted on premise.
The Platform scales horizontally, supports up to tens of thousands of physical or virtual hosts/nodes and is self-healing for redundancy.
Each participating host runs an Aporeto Enforcerwhich stand in for the TCP/IP stack and permits or denies a Processing Unit (Linux process or Docker container) from performing actions within the system. A Zero Trust Model is used, so a Processing Unit (PU) cannot perform any actions unless it has been explicitly given permission to do so by an Aporeto Network Policy.

Chain of Trust

Each PU must be trusted by the Aporeto Platform. The chain of trust is established by first requiring that each physical or virtual host register and run the Aporeto Enforcer service. Registration is with the Aporeto Identity Access Management (IAM) system, which may be run locally or in the Aporeto SaaS cloud. Aporeto will either store the registration information in its built-in IAM or federate with popular IAMs such as Google cloud, Amazon cloud, or a corporate LDAP-compliant IAM such as Active Directory.
After being registered, Enforcer generates a public-private key pair locally, keeps the private key private, and sends an X.509 digital certificate containing the newly created public key and

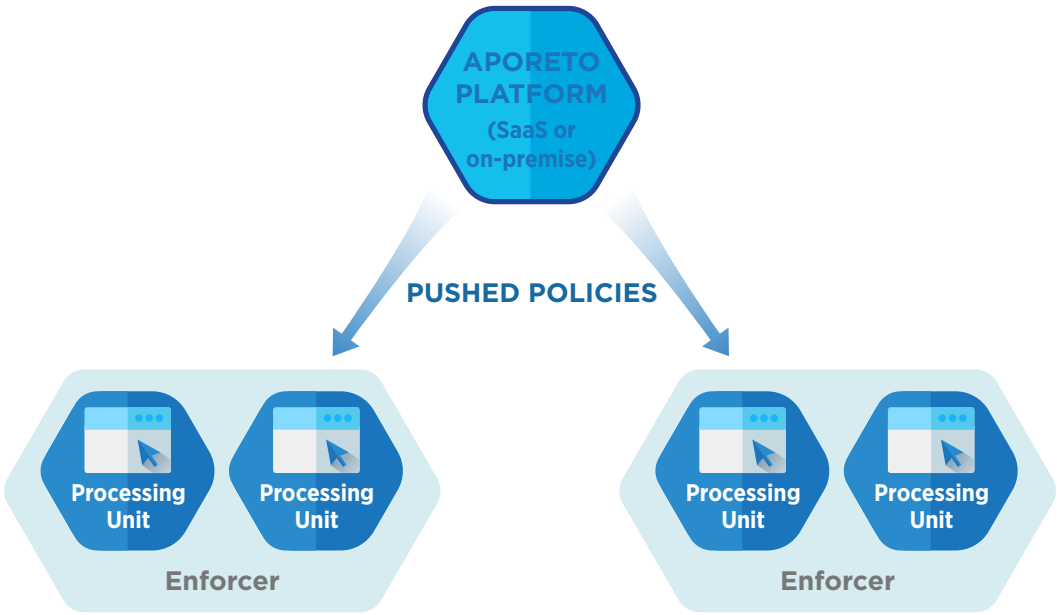

Figure 5: Aporeto Platform protects Processing Units using Authorization Network Policies

HOW IT WORKS

unique information that identifies the host to the Aporeto CA where it is signed and sent back so the host.

Once the Aporeto Enforcer, running on each host, has established a strong, cryptographic a chain of trust with the Aporeto CA, it is ready to use its signed X.509 certificate to sign a Trust Profile for each PU running on that host. The Trust Profile is automatically sent on behalf of the PU when it attempts to perform operations on the Aporeto Platform.

Object Tags

Each Processing Unit (PU) has a collection of tags associated with it that come from 5 different sources: object attributes, attributes defined by the user when starting a Docker container attributes that are auto-defined by the Aporeto Enforcer, attributes from user authentication tokens, and other user-defined attributes. These tags are swept up and included in the Trust Profile that Aporeto automatically generates for each PU.

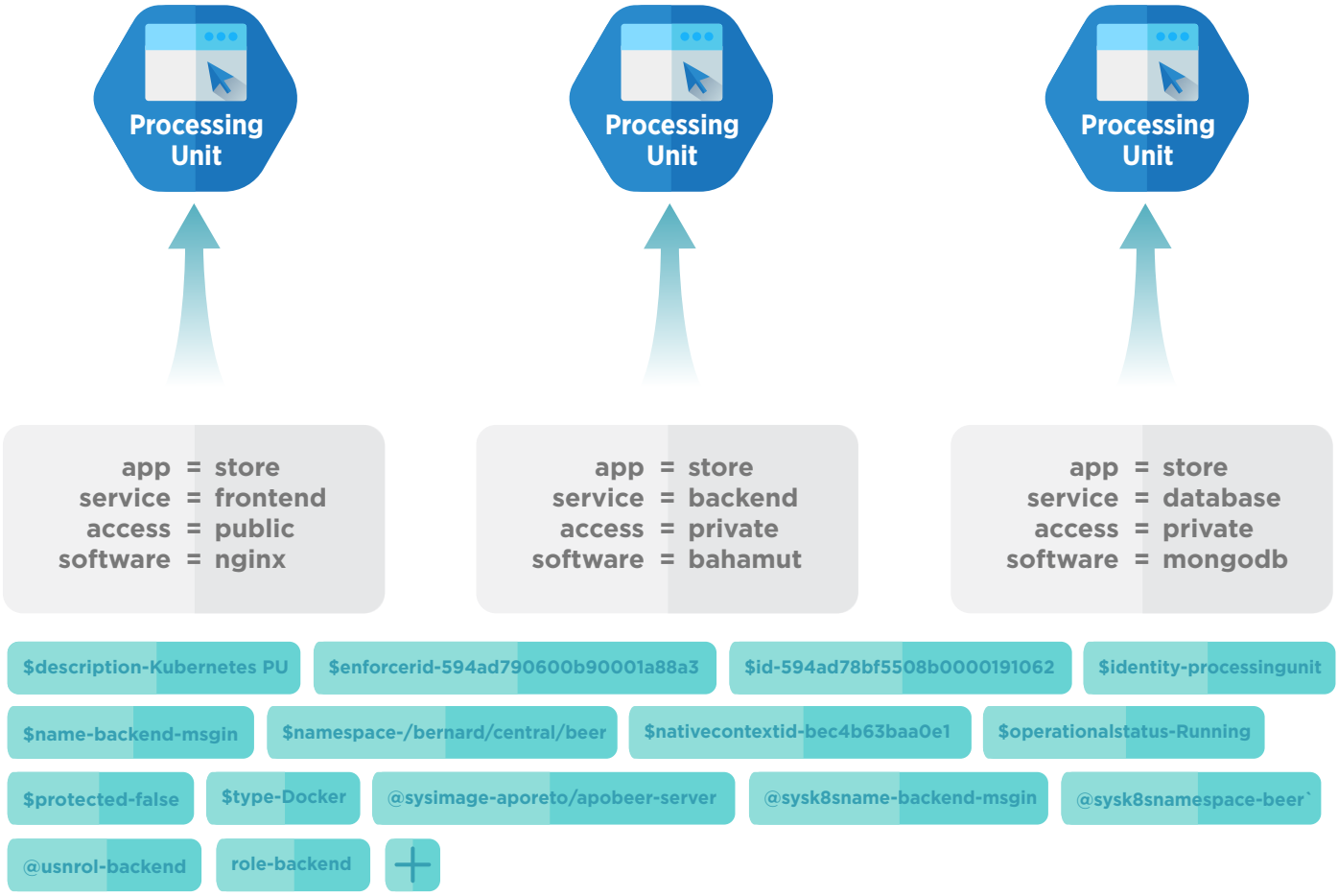

Figure 6: Aporeto uses trusted identity and object tags to create the Trust Profile

HOW IT WORKS

Trust Profile Technology

The Aporeto Platform automatically generates a Trust Profile to be used by each service (PU) in the environment. A Trust Profile includes information such as who (the host-id, process-id, user account that the process is running under), where (network address), what (type of service), metadata (any number of name-value pairs), reputation (from ratings services), and behavior (from past history running in the environment). Trust Profiles are hashed and cryptographically signed by Enforcer on a trusted host to ensure they are genuine and cannot be tampered with. You can think of a Trust Profile as a passport. Every service (PU) that wishes to operate in the environment presents its Trust Profile to the Aporeto Enforcer, where it is used to determine if access should be allowed or denied.

By combining multiple factors into the Trust Profile, using the NIST ABAC model, an unsurpassed security posture can be accomplished. Because the creation of the Trust Profiles is done automatically on-the-fly, security is finer grained than perimeter security around the application, and is constantly kept current without requiring tedious manual efforts.

Network Policies

A Network Policy controls what a PU with a given Trust Profile can do within the system. The syntax to define a new Network Policy is intuitive, leading to better application security, visibility, auditability. They are easily understood because they use a simple subject, verb, object syntax.

Aporeto automatically generates the list of Network Policies that reflect how all PUs in the application have interacted for a period of time. This saves SecOps personnel a lot of time, is comprehensive and is precise. To refine the

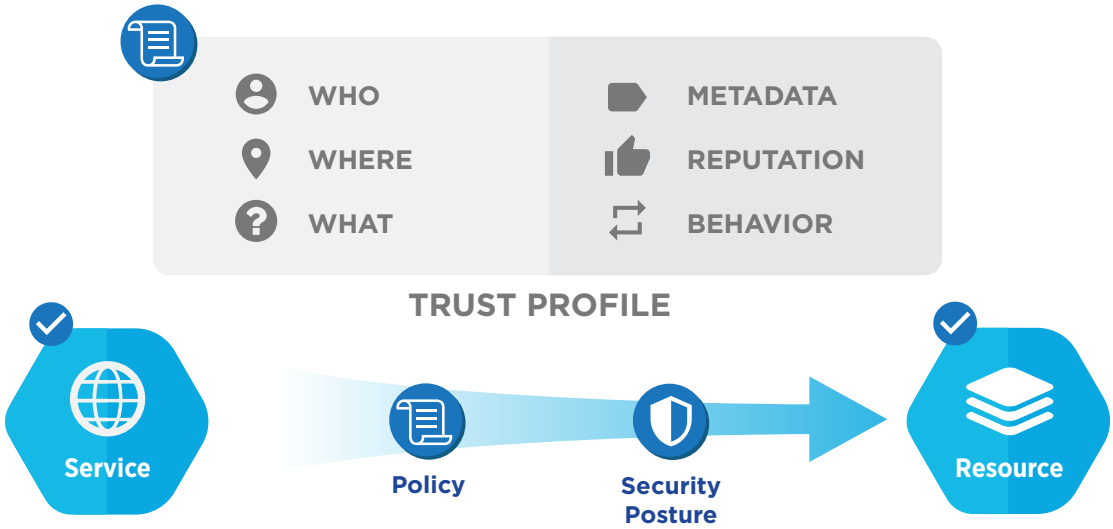

Figure 7:Aporeto uses the Trust Profile to allow or deny access to the environment

HOW IT WORKS

security of the application, Policies can be easily inspected and edited.

Automatic Encryption

Once the connection is established, if the Network Policy requires traffic between PUs be encrypted, the Aporeto Enforcer encrypts all data using the AES256 GCM algorithm. This provides encryption of data in flight with no changes to source code or key management administration required!

Aporeto Enforcer

Naturally, a Processing Unit (Linux process or Docker container) that wishes to interact with another PU does so over a TCP/IP connection. When the Aporeto Platform is installed on a node, it becomes a proxy for the TCP/IP stack in the underlying operating system so that all TCP connection requests go through the Aporeto Enforcer.

Runtime Environment

Aporeto provides comprehensive, automated security for cloud-native applications built with Docker containers, microservices, serverless architectures, as well as traditional Linux or Windows processes running on VMs or bare metal servers. It supports popular orchestration engines including Kubernetes, Red Hat OpenShift, Mesos DC/OS and Docker Swarm as well as popular infrastructure administrative tools including Chef, Puppet, and Ansible, to name a few.
Aporeto can be used in hybrid cloud environments that span on-premises, (private cloud) and public clouds. Applications can span multiple availability zones to ensure uptime and multiple clouds can be utilized simultaneously to provide business agility and optimize costs based on varying pricing policies. In order to support logically isolated virtual networks for billing, regulatory or other purposes, multiple simultaneous VPCs such as Amazon Virtual

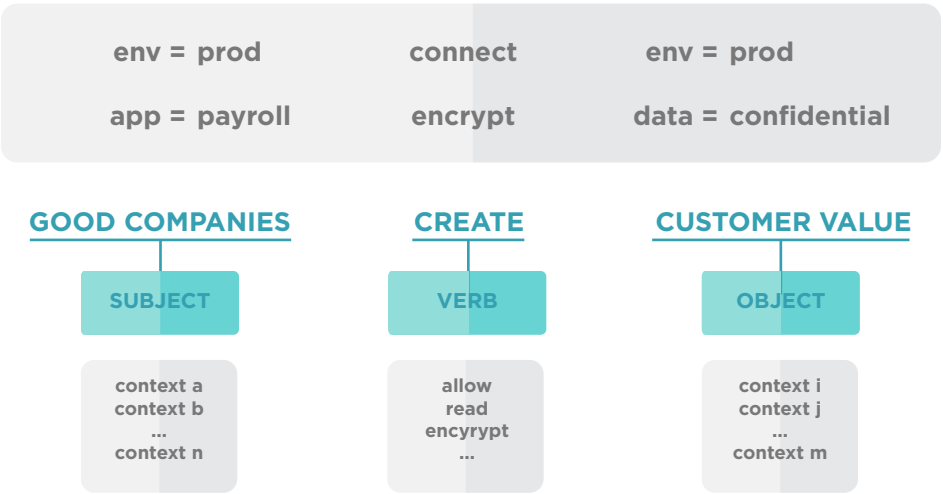

Figure 8: Aporeto uses the Authorization Policies to allow or deny access to the environment

HOW IT WORKS

Private Cloud (VPC) environments are also fully supported.

Aporeto has been designed with strict multitenancy security in mind from the very beginning. This allows a single instance of an application to serve multiple companies, business units or groups with full isolation based on hierarchical namespaces that can be arbitrarily deep and policies that may be propagated from parent to children namespaces to ease the administrative effort. Users are mapped to Roles within namespaces with predetermined Authorization Policies for data, filesystems, and API access control.

Centralized Management

Aporeto runs in a fully distributed fashion across physical and virtual hosts to deliver fast, scalable performance that scales as hosts are added, and to provide high availability while enforcing end-to-end application security. Aporeto is centrally managed, provides comprehensive control and visibility, is easy to implement, and provides verifiable security,

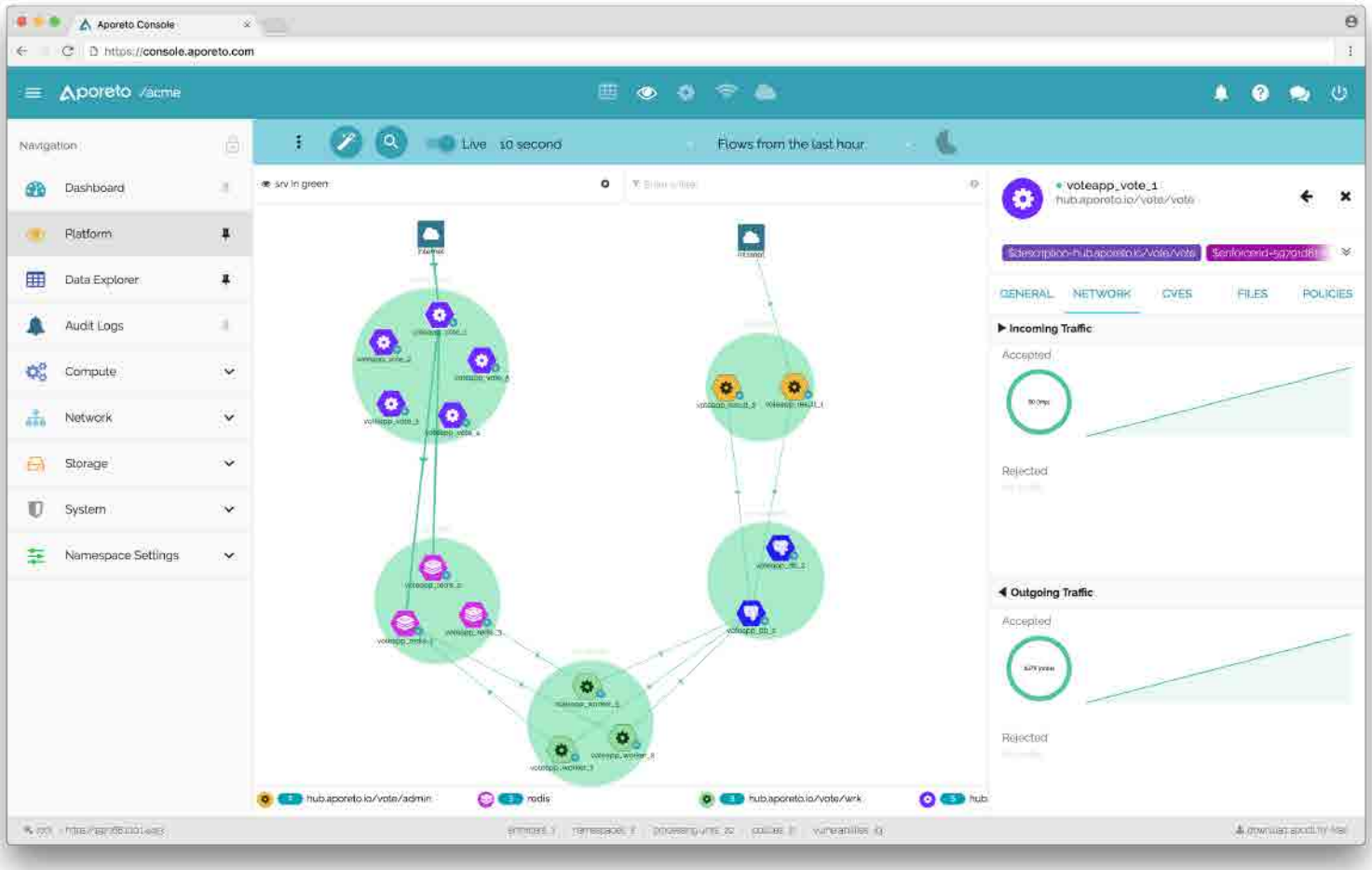

Figure 9: Application-aware security management console

HOW IT WORKS

even for very large scale applications. Operations and Security personnel benefit from improved visibility and control over mission-critical applications that are in production. Management of the Aporeto Platform can be achieved through the Web-based GUI, CLI or REST APIs.

Aporeto Architecture

Aporeto utilizes a lean, clean, extensible architecture that builds upon a foundation of trusted identities operating in a zero trust environment. The Aporeto Enforcer uses modular, pluggable modules to permit or deny access to the network, services, encryption, APIs of the platform itself, file system, and secrets services. The system is extensible, so other 3rd party modules can be added. The analytics and policy management layer ensures that operations are running smoothly, manages policies and provides end-to-end visibility of the security posture of the cloud-native application. Enforcer is provided as an open source project (Trireme at https://goo.gl/TEhDpT) to provide guidance about how to write 3rd party Enforcers to extend the platform.

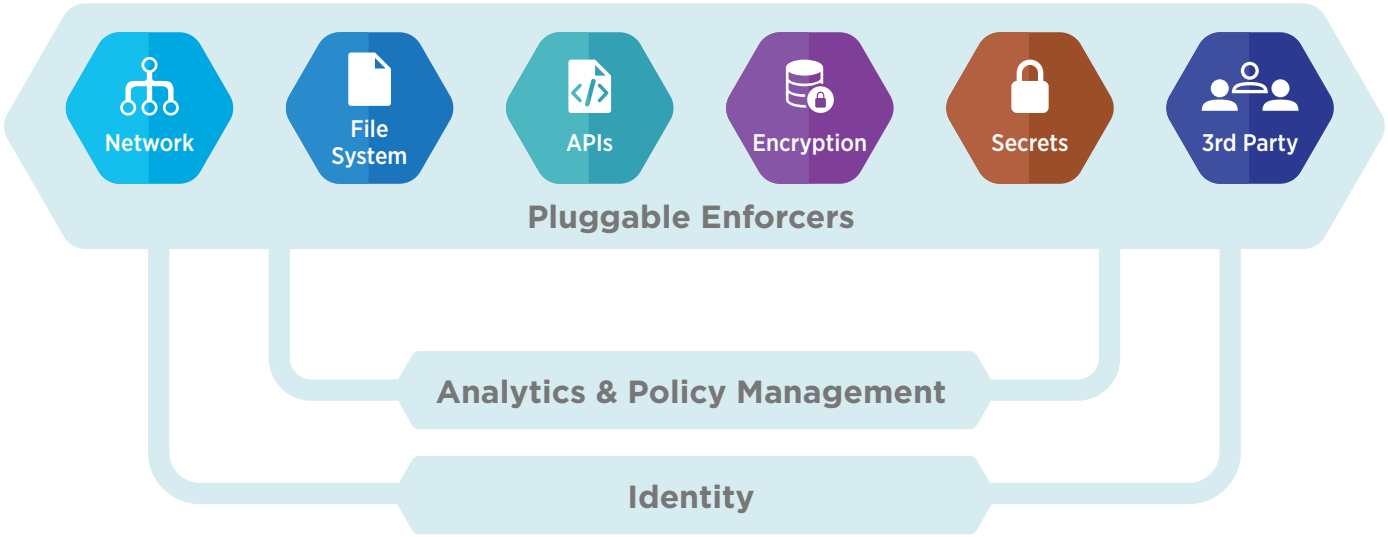

Figure 10: Aporeto provides a pluggable, extensible architecture

HOW IT WORKS

Authentication Protocol

The Aporeto Platform can federate with existing Identity and Access Management (IAM) systems in order to authenticate Enforcers running on hosts, so they can become trusted participants. For Google's IAM, OAuth2 is used. For Amazon Web Services, the AWSIdentityDocument is used. For a corporate IAM, LDAP is used. In all cases, the host provides credentials to the Aporeto Platform and, upon successful authentication, receives a time-based token that is periodically refreshed to ensure tight security.

Communication Protocol

The Aporeto Enforcer on the initiating host will work with the Aporeto Enforcer on the receiving host to establish the TCP connection if the Trust Profile of both PUs are known and the Network Policy permits it. All TCP traffic goes through the Aporeto Enforcers to ensure that only PU interactions that are explicitly permitted by an appropriate Policy get through. As described in RFC 793 (https://goo.gl/CDATMC), two hosts running TCP/IP establish a connection using a three-way handshake protocol. First, the initiating TCP client sends a SYN (synchronize

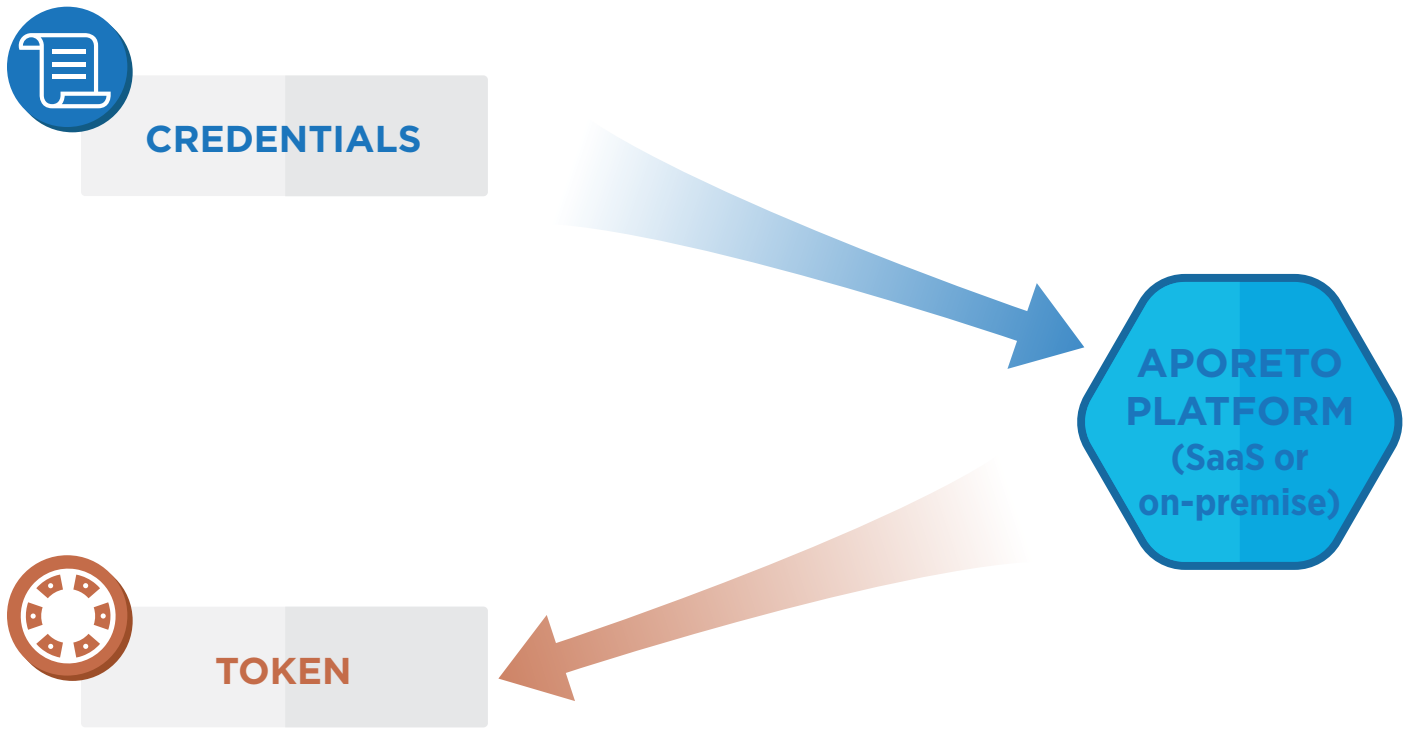

Figure 11: Aporeto authentication protocol returns a JWT token

HOW IT WORKS

sequence numbers so later packets can be kept track of) message to the listening TCP server. Second, the listening TCP server acknowledges this request by sending a SYN-ACK back to the initiating TCP client. Finally, the initiating TCP client responds with an ACK and the connection is established. SYN flooding attacks and effective countermeasures are well known and are described in detail in RFC 4987 (https://goo.gl/DpkpA8).

The Aporeto Enforcers which run on each host use a similar three-way protocol to establish a secure connection and take similar measures to foil SYN flooding attacks. Additional information is attached to the SYN, SYN-ACK and ACK messages as a signed JSON data structure to facilitate mutual authentication of the PUs that wish to participate in an interaction.

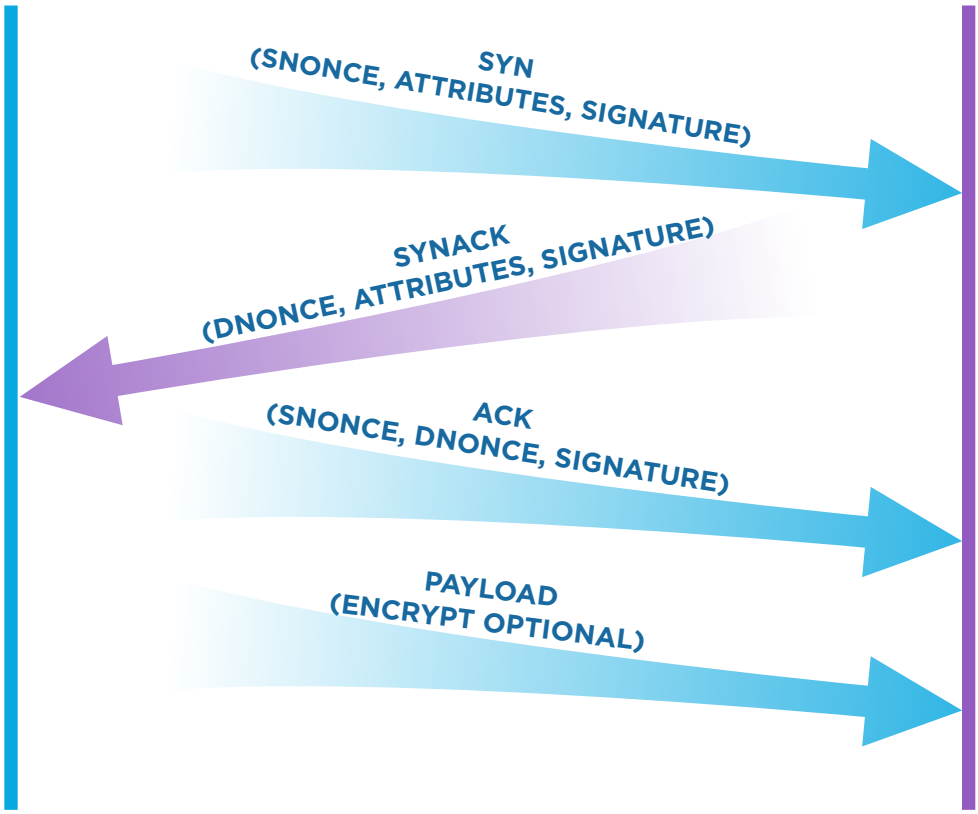

Figure 12: TCP three-way handshake

EXAMPLE

As a concrete example of the Aporeto Platform in use, assume three users or processes want to access a Linux system or a Hadoop cluster. Aporeto will monitor the running application and use advanced analytics on the information collected to understand the intent of the application automatically, then generate the Network Policies that represent a healthy, working application. Once the automatic discovery phase is complete, SecOps personnel can view the Network Policy rules, as well as edit them to fine tune the security of the overall application.

The users of the system are known by more than just their user-id. Additional contextual information is used to create a Trust Profile for each user. Trust is unidirectional, so in this example Asit can access the Linux system, but the Linux system cannot communicate with him unless that Access Policy is defined. Notice that the Linux system can access the Hadoop cluster on behalf of Asit, but Asit cannot access the Hadoop cluster directly. Application Policies make it easy to setup and maintain end-to-end security across all of the components of the cloud-native application.

OUR SECURITY OFFICER DECREES

1. **Hadoop may talk with Hadoop if the user is prod**
2. **MyApp may talk with MyApp if the user is prod**
3. **MyApp may talk with Hadoop if the user is prod**
4. **Any user of group myapp_users may talk to myapp**
5. **Any user of group Hadoop_admins may talk to Hadoop if the user is prod**
6. **Anything not explicitly permitted is forbidden**

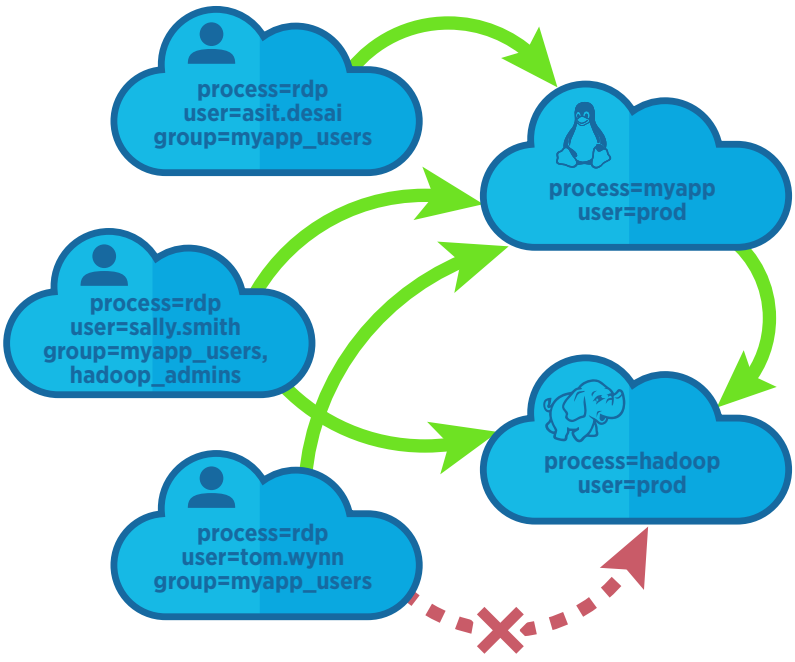

Figure 13: Example of a computing environment that needs to be secured

SUMMARY

The Aporeto Platform eliminates problems inherent in the old security perimeter model of securing applications. Rather than using thousands of static rules and configuration settings across various vendor's firewalls, IPS, WAF, NAC, and VPN gateways, SecOps personnel can now leverage Aporeto. Aporeto does not require any changes to source code.

It watches application component interactions and uses machine learning to automatically generate Network Policies that are enforced by the Aporeto Enforcer which runs on each physical or virtual host as a proxy in front of the TCP/IP stack. The end result is unsurpassed end-to-end security, ease of deployment, visibility and control of the security posture for an application.

Thanks to the lower complexity that comes from using the Aporeto solution, DevSecOps personnel can focus on activities that advance the business, rather than wrestling with the problems inherent in the old, network-oriented application security model and role-based access controls. Once Aporeto is in use, the organization's security posture changes right away from a being unclear and difficult to maintain to being always current, easy to maintain, and verifiable – lowering costs, improving security, mitigating organizational risk, and increasing velocity of innovation of the organization to better serve customers and out maneuver competitors.

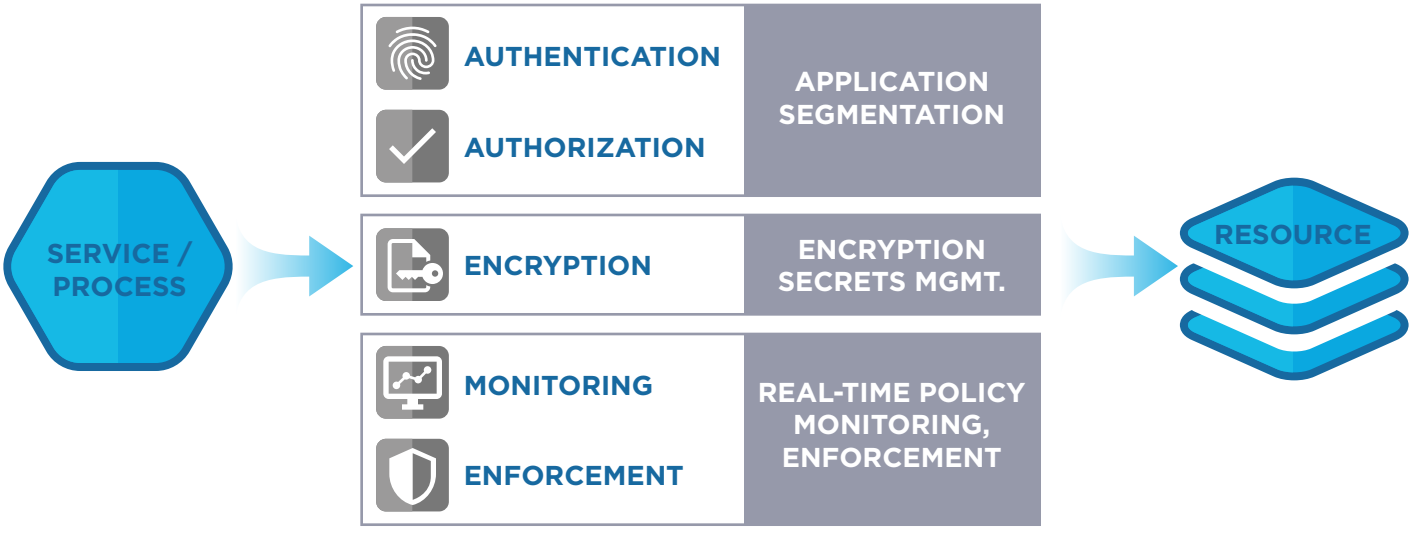

Figure 14: The Aporeto Platform in Action

Aporeto

Cloud-Native Security

About

Aporeto secures cloud-native workloads based on microservices, containers, and serverless architectures as well as legacy workloads built on VMs or baremetal servers. The venture-backed Silicon Valley startup is funded by NVP, Wing VC, and other leading investors.

aporeto.com

info@aporeto.com

@aporeto

(866) 300-0224