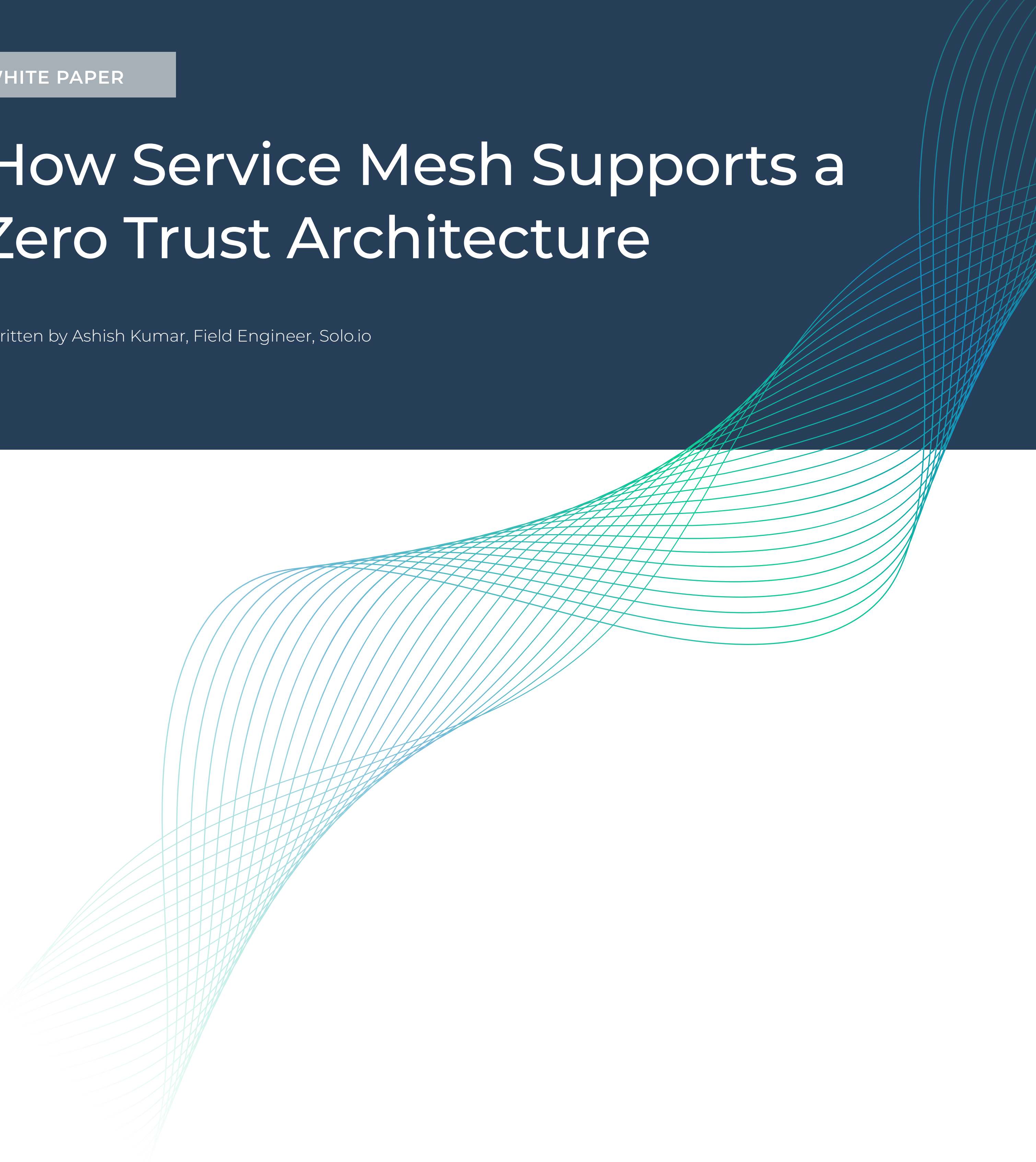
WHITE PAPER

How Service Mesh Supports a Zero Trust Architecture

Written by Ashish Kumar, Field Engineer, Solo.io



solo.io

Contents

INTRODUCTION

As zero trust architecture becomes adopted by more and more organizations, adopting service mesh to implement and enforce zero trust will become increasingly popular and necessary.

Service mesh provides a more thorough way to address all of the zero trust principles. All elements of zero trust can be implemented by a service mesh, with the exception of continuous cybersecurity operations.

The zero trust maturity model includes five distinct pillars – with incremental advancements toward optimization.

UNDERSTANDING ZERO TRUST

The zero trust maturity model includes five distinct pillars – with incremental advancements toward optimization. The pillars are:

- **Identity**
- **Device**
- **Network/environment**
- **Application workload**
- **Data**

Each pillar also includes general details regarding governance, visibility and analytics, as well as automation and orchestration. It can progress at its own pace and should maintain its own maturity model.

Service mesh provides capabilities for implementing optimal identity, network/environment, application workload, and data pillars, according to the CISA maturity model and NIST framework.

The MIT zero trust architecture framework defines zero trust components based on the "Jobs to be done" framework, with three key components:

- Identity verification
- Access control
- Resource protection

As well as two overarching governance components:

- Policy and orchestration
- Monitoring and analytics

There are also some additional components to further improve the maturity of zero trust architecture:

- Microsegmentation
- Least privilege

ADDING THE SUPPORT OF SERVICE MESH

Here's how service mesh can support some of these zero trust pillars and components.

Access Control

In a service mesh, a combination of network policy, authorization, and root trust policy provide mechanisms for access control.

Resource Protection

Gateways and sidecars in a service mesh are designed and configured to protect the attached resources. Gateways provide resource protection at a cluster level as well for resources that are outside the Kubernetes cluster. Sidecars provide granular protection at an individual resource level.

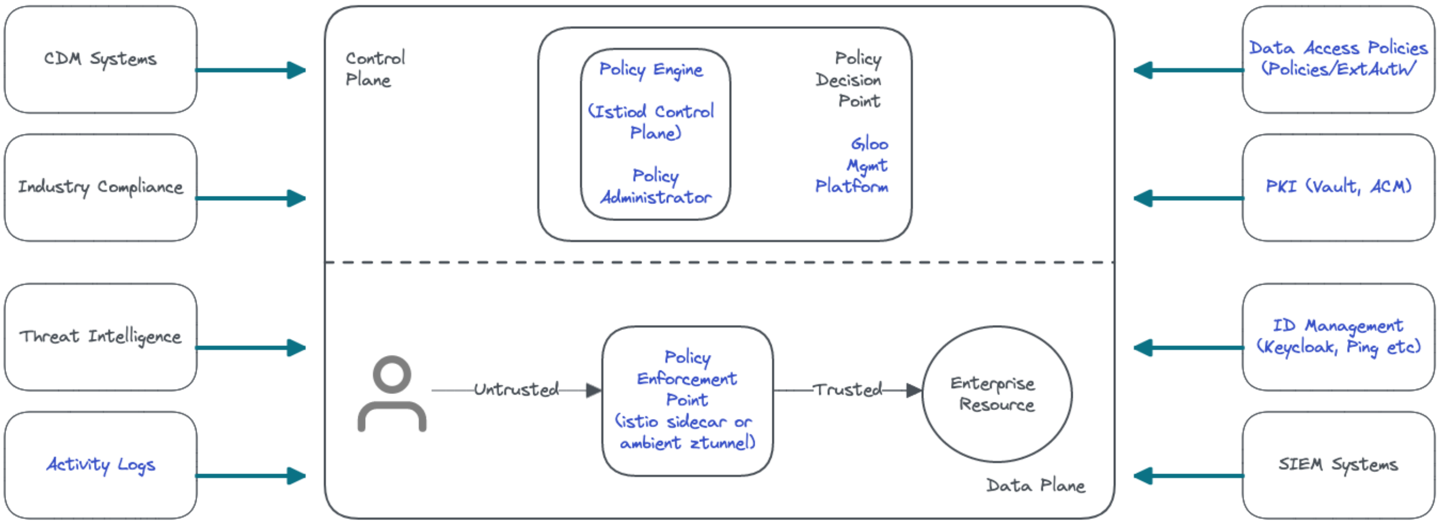
Monitoring and Analytics

All policy enforcement points (gateways and sidecars) in a service mesh are usually configured for pushing metrics and logs. Keeping zero trust principles in mind, only a single management tool like Gloo Platform can configure mesh resources for this. Combined with data masking, this forms the basis of zero trust governance in a service mesh.

Least Privilege

A single configuration in a service mesh can implement "deny all" for network communication between workloads. Authorization, network, and security policies can then be used to selectively open communication based on identity verification.

The examples above are based on the MIT zero trust framework. The picture below shows an overlap of service mesh capabilities with zero trust components based on the NIST framework.



Logical Components of Zero Trust

As zero trust implementations mature, more advanced zero trust capabilities will be implemented.

In this model, a service mesh provides all core components of zero trust architecture, as well integration with other capabilities to implement zero trust maturity.

As zero trust implementations mature, more advanced zero trust capabilities will be implemented. One such concept is "attribute-based access control" or ABAC. ABAC is an authorization framework that dynamically computes decisions for access control based on attributes (properties) of users (invoking clients), application objects (resources), and the environment and policies expressed in terms of attributes.

For example, looking at attributes like location of mobile phone or the speed at which a mobile phone is moving, to decide whether to throw an enhanced security challenge back to the client in addition to verification of identity.

The NIST paper "Attribute Based Access Control for Microservices-based Applications Using a Service Mesh" explains more on how to implement ABAC using Istio.

THE LIMITATIONS OF SERVICE MESH FOR ZERO TRUST

Implementation of zero trust architecture will require long-term commitment from organizations, and will need to be done based on some maturity model.

Service mesh provides a minimum viable set of capabilities for initial implementations of zero trust. In addition to service mesh, a zero trust architecture implementation will require:

- Continuous monitoring and diagnostic systems that can operate on data provided by the mesh and can interact with a policy engine to dynamically configure policy
- Industry compliance based on geography or industry verticals like HIPAA and PCI
- Threat intelligence with external data feeds and machine learning models that can provide information to detect both known and unknown threat vectors, like a machine learning model that can detect change in attributes for a mature of implementation of service mesh
- Activity log integration, since a service mesh can generate a significant amount of logs
- SIEM integration – in a mature implementation, SIEM systems will dynamically configure service mesh policies based on event analysis

SERVICE MESH VERSUS API GATEWAY FOR ZERO TRUST

An API gateway provides capabilities to implement an optimal identity pillar, while service mesh allows for implementation of multiple zero trust pillars. This table compares capabilities provided by both service mesh and API gateways for implementation of zero trust components:

Zero Trust Component	Current State	Future State	Service Mesh	API Gateway
Identity Verification	Corporate SSO	Cloud-based identity	Service mesh verifies workload, application identity e.g SPIFFE identity in istio	Well-defined mechanisms for user identity verification e.g. ext_auth in Envoy
Access Control	Continuous all-to-all connectivity	Brokered one-to-one connectivity i.e. zero trust network architecture	Sidecar proxies can be configured as policy enforcement points for role based access control with default configuration to deny all traffic and providing least privilege	API gateways provides role based access control for end user identity. Also useful for providing access control to legacy monolith applications that cannot be part of service mesh
Resource Protection	Encryption at REST, Redundant apps	JIT authentication, DRM, Data provenance, Resilient apps	Service mesh can enforce mtls for encryption, provide intelligent routing for increasing application resiliency	API gateways cannot provide encryption on their own but are suitable protecting legacy resource types
Policy & Orchestration	Static bespoke, siloed configs	Dynamic decision making	Service meshes have sidecars as policy enforcement points in the data plane and a separate control plane as policy engine and data point. Control plane can be integrated with external systems for dynamic policy configuration	API gateways can also be integrated with external systems to enforce dynamic decision making
Monitoring & Analysis	Edge and endpoint monitoring	Integrated system-wide monitoring and analytics	Responsibility offload model of service mesh with or without a sidecar ensures an integrated system-wide monitoring and analytics by configuring policy enforcement points (sidecars) to provide all relevant data, including golden signals, to external monitoring, logging systems for analysis	API gateways can only provide monitoring and analysis for subset or transitions in an enterprise setup
Continuous Cybersecurity Operations	Compliance focused	Periodic adversarial testing	Regular testing/updates of zero trust network/environment needs to be triggered from outside the mesh	Regular testing/updates of API gateway will be triggered from outside the mesh or gateway

HOW SOLO.IO CAN HELP

Solo's Gloo Platform is based on Istio, and provides all its capabilities for an initial and robust implementation of zero trust architecture:

- **Identity verification** – Gloo Mesh uses SPIFFE for workload attestation and identity and OpenID Connect-based JSON web token verification for non-machine identity.
- **Psychological acceptability** – A cyber security principle that refers to ensuring ease of use to avoid non-compliance among approved users. Gloo Platform, through its management plane, provides a well-defined and user-friendly mechanism to configure, deploy, and operate a service mesh.
- **Microsegmentation** – Gloo Platform's workspace feature provides advanced micro segmentation capability to configure granular access control for the segment. In a kubernetes setup, workspaces span across multiple clusters to provide a manageable access control across microsegments. You can read more about Gloo Mesh workspaces here.
- **Combined gateway and mesh** – Gloo Platform's management plane provides lifecycle management for both Gloo Gateway and Gloo Mesh. This enables use of both API gateway and service mesh for implementing a zero trust architecture, which can extend beyond Kubernetes clusters, into legacy virtual machine-based applications, as well serverless capabilities like lambda.
- **Centralized policy decision point** – Gloo Management plane can manage multiple Istio control planes to create a multi-cluster mesh with identity and certificate distribution and rotation across the entire mesh. It can also be integrated with zero trust components like public key infrastructure, threat intelligence systems, etc. to provide a central policy engine with distributed policy enforcement.

All security concepts mentioned here are also applicable with ambient mesh, with an added advantage of a further reduced blast radius in ambient.

In addition to world class products, Solo's team includes people who've been involved with the Istio project since inception, are part of technical oversight and steering committees, and have been very regularly contributing to Istio projects. Solo was heavily involved in development and release of Ambient Mesh, the sidecarless data plane mode for Istio.

Our field teams are actively involved in multiple zero trust implementation projects across the globe, creating a world of knowledge for solving your zero trust implementations.

Learn more by visiting our Gloo Platform product page today.

For More Information

REQUEST A DEMO AT SOLO.IO/DEMO

contact@solo.io

www.solo.io

About Solo.io

Solo.io, the leading application networking company, delivers a service mesh and API platform for Kubernetes, zero trust, and microservices. The three components of the Gloo Platform – Gloo Gateway, Gloo Mesh and Gloo Network – enable enterprise companies to rapidly adopt microservice applications as part of their cloud journey and digital transformation. Solo delivers open source solutions, and is a community leader in building the technologies of the future.