



AHEAD

# Securing Platform Engineering & Developer Enablement

# Executive Summary

The following whitepaper presents a comprehensive framework for secure platform engineering and developer enablement, offering a unified approach to address evolving challenges in the digital landscape. Promoting the seamless integration of security measures into the development lifecycle, this paper aims to demonstrate how enhanced collaboration between security and development teams can foster a proactive security culture.

Below, we'll explore key concepts such as threat modeling, secure coding practices, and continuous monitoring in an effort to empower organizations to build resilient and secure platforms. Emphasizing a holistic perspective, our approach mitigates vulnerabilities and accelerates software development so that developers may innovate confidently in a secure environment.

## Introduction

Platform engineering has emerged as a critical foundation for organizations as they strive to enhance agility, scalability, innovation, and consistency across developer teams. As the demand for rapid software delivery intensifies, the evolution of platform engineering within the enterprise becomes increasingly necessary.

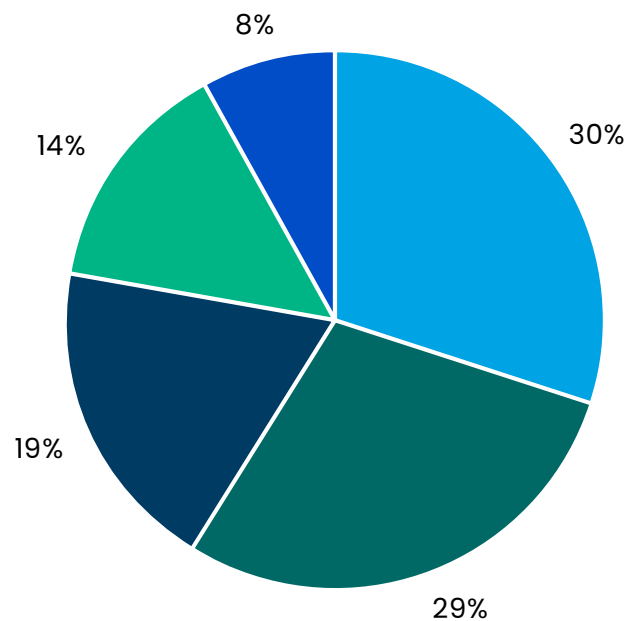
[Gartner](#) recognized platform engineering as one of the key strategic technology trends for 2024.

The paradigm shift toward cloud-native architectures has propelled the prominence of platform engineering, as articulated in the Puppet '[2023 State of DevOps: Platform Engineering](#)' report, where they found that adoption reached 51% in 2023.

[Internal developer platforms](#) (IDPs) are central to this evolution, catalyzing streamlined development workflows and collaboration. Further, a report from [OrangeMatter](#) highlights that more than 58% of participants noted accelerated delivery times, 60% experienced enhanced system reliability, and 59% observed efficiency gains.

### Whose needs are best served by Platform Engineering?

- The whole company
- Developer teams
- The entire department
- Infrastructure teams
- Individual developers



[Source](#)

However, as these platforms scale, challenges related to security vulnerabilities become more pronounced. The alarming rise in cyber threats—with a staggering [104% increase](#) in reported incidents in 2023—emphasizes the critical need for a unified and secure approach to platform engineering.

## Unifying Platform Engineering & Security

With regard to platform engineering, security is the keeper of organizational resources, guaranteeing the availability, integrity, and confidentiality of vital systems. A proactive security approach thus becomes essential for reducing vulnerabilities and protecting against possible breaches in an era of ubiquitous cyber threats.

Organizations can strengthen their infrastructure against changing threats (and retain stakeholder trust) by incorporating security at the heart of platform engineering. In doing so, project leaders and developers need to give preference to security from the very beginning. This early prioritization of security allows developers to innovate confidently, knowing that their work is as safe as possible from potential attacks.

Furthermore, a strong security framework ensures scalability without compromising vulnerability preparedness as systems develop to accommodate the needs of expanding user bases. It encourages steady expansion, enabling businesses to increase their digital presence while maintaining the strongest possible security posture.

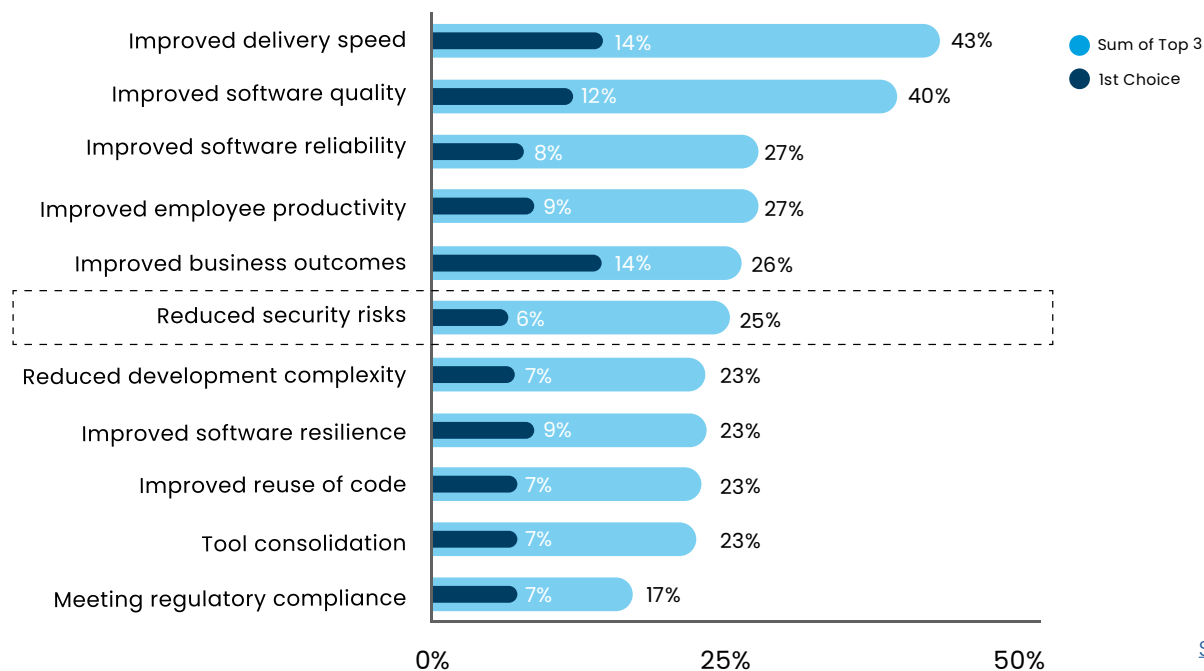
Some of the ways that platform engineers can address security as a core component of growth and adoption include:

- [Adopting a zero trust approach](#) that assumes breach until proven otherwise.
- Implementing [DevSecOps principles](#) that integrate security into every stage of the development process.
- Leveraging cloud-native technologies that provide built-in security features such as encryption at rest/transfer/in transit, identity management, access control, etc.
- Employing a code signing policy to verify the authenticity and integrity of programs, files, software, etc.
- Including a Software Bill of Materials (SBOM) to provide a detailed inventory of software building blocks and dependencies.
- Following industry standards or frameworks such as [NIST](#), [ISO](#), [OWASP](#), [CIS Center for Internet Security](#), etc.
- Educating themselves and others about the latest trends, threats, best practices, tools, etc.

[Gartner's 2022 Software Engineering Leaders Role survey](#) reveals that just 25% of respondents identified “reducing security risks” among their top three goals for platform engineering, with a mere 6% ranking it as their highest priority. It is therefore evident that utilizing platform engineering to extend application security practices throughout the organization needs to be more valued and noticed.

### *Only 6% of respondents cite reduced security risks as the top goal for Platform Engineering*

Percentage of Respondents



[Source](#)

# Infrastructure as Code in the Secure Software Lifecycle

[Infrastructure as Code \(IaC\)](#) is an essential element of platform engineering that allows organizations to manage and automate computing infrastructure through code.

A DuploCloud survey of [40+ enterprises](#) that have made substantial investments in platform engineering showed that the prevalent approach is for DevOps teams to build DevOps platforms with IaC as a core underlying technology.

This approach enhances automation, scalability, and reliability by allowing organizations to track, version control, and automate changes in their infrastructure. The adoption of IaC brings about increased efficiency, diminished human error, and a more agile response to market demands.

The significance of IaC in platform engineering manifests in several key aspects:

## Reusability & Shareability

IaC facilitates the reuse of configurations across multiple environments, allowing organizations to progressively enhance, improve, or build upon existing code.

## Reliability

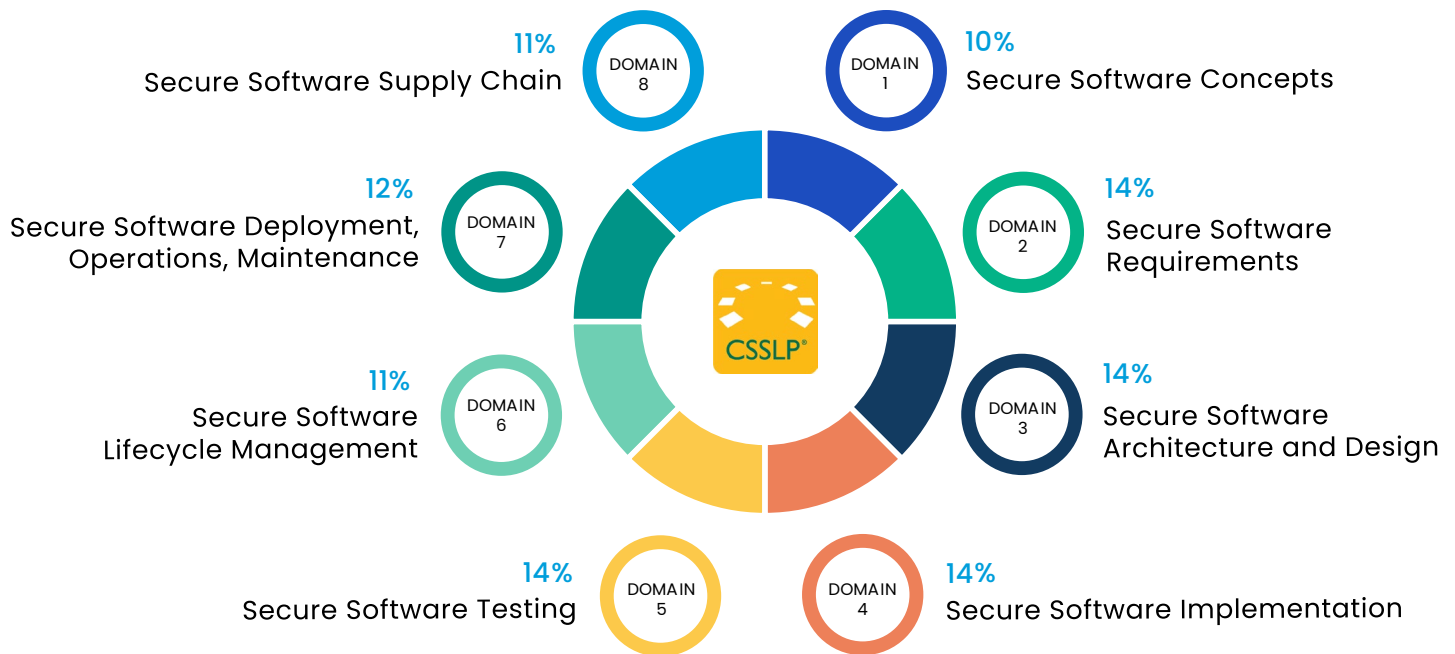
By using automated tests, code reviews, and static code analysis, businesses can use IaC to preserve the integrity of their infrastructure. This guarantees a dependable, safe basis for creating and implementing services and applications.

## Automation & Speed

IaC is essential in automating deployment and provisioning procedures, drastically reducing the time needed to launch new services and apps.

# Integrating CSSLP Principles into IaC Processes

To ensure security over the entire development-to-deployment spectrum, it is imperative that IaC processes incorporate the standards of the [Certified Secure Software Lifecycle Professional \(CSSLP\)](#). The CSSLP validates a person's ability to include security measures at every stage of the software development lifecycle.



[Source](#)

This integration of [CSSLP](#) principles into IaC processes solidifies security as an inherent element within an organization's platform engineering practices. By adhering to CSSLP guidelines throughout each stage of the Software Development Life Cycle (SDLC), developers can proactively address security concerns early in the development process.

For instance:

- Developers can use threat modeling techniques to find potential vulnerabilities and create associated security measures throughout the requirement gathering and analysis process.
- Developers can prevent typical security weaknesses like [injection attacks](#) and [buffer overflows](#) by implementing secure coding techniques throughout the secure design and architecture phase.
- In the secure coding practices phase, developers can utilize static analysis tools to identify and rectify code errors or vulnerabilities before deployment.
- Developers can use penetration testing tools or manual testing methods to validate functionality and security during the comprehensive testing and quality assurance phase.
- In the continuous monitoring and updates phase, developers can employ monitoring tools or alert systems to detect and respond promptly to security incidents or anomalies.

Ensuring security throughout the development-to-deployment continuum remains essential in platform engineering. As emphasized by [DevOps.com](#), integrating built-in security and compliance controls within IaC is imperative to establish security as a fundamental element of the platform.

By integrating CSSLP principles into Infrastructure as Code (IaC) processes, developers bolster the security of their applications throughout the entire development-to-deployment lifecycle. This mitigates the risk of data breaches or service disruptions and fosters enhanced customer trust and satisfaction.



# The Role of Security Automation in Platform Engineering

In the context of platform engineering, security automation is critical for streamlining management, enhancing control, and minimizing the likelihood of security vulnerabilities. As outlined in a report by [PlatformEngr](#), automation facilitates the rapid scanning of logs, monitoring of network activities, and swift identification of suspicious behavior, surpassing human capabilities. This not only eradicates human error, but diminishes the risk of security breaches.

An integral facet of security automation in platform engineering is the augmentation of the developer experience through automated security measures.

According to insights from [The New Stack](#), adopting self-service infrastructure as code (IaC) significantly enhances developer experience via automated security measures.

The provision of self-service IaC empowers developers to construct secure and compliant applications without compromising on speed or agility.

Below are some strategies for enhancing the developer experience through security automation:

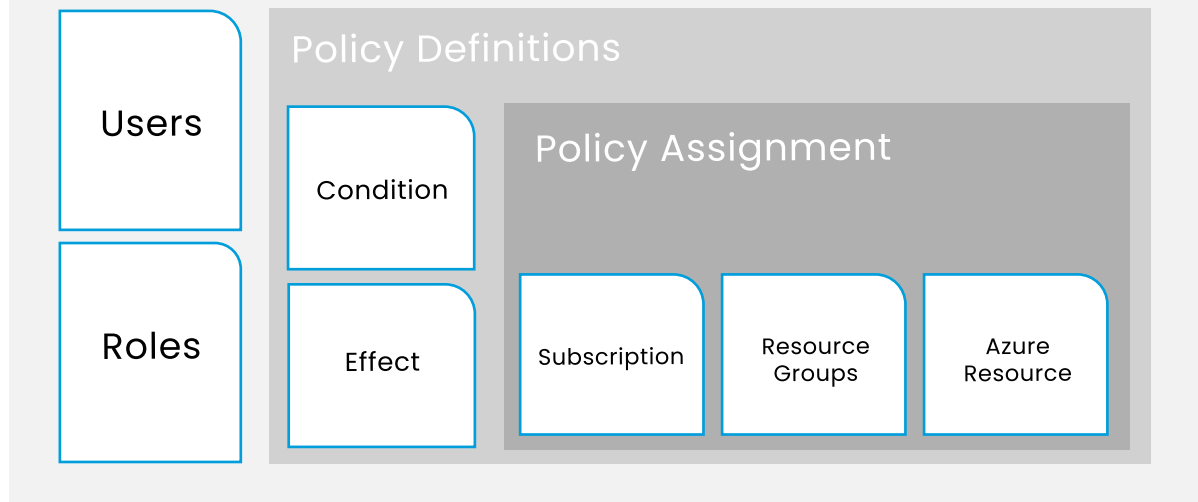
## 01 User Account Provisioning and De-provisioning

User account management is fundamental to robust security practices. Efficient and secure on- and off-boarding can be guaranteed by automating [user account provisioning and de-provisioning](#). In terms of new hires, automated provisioning will help to reduce manual onboarding errors, ensure timely access to resources, and streamline the establishment of user accounts. On the other end of the spectrum, implementing automated processes for de-provisioning accounts when employees leave the organization—such as revoking access and de-provisioning associated resources promptly upon termination—will drastically minimize security risks.

## 02 Role-Based Access Control (RBAC)

Granular control over user permissions is provided by [RBAC policies](#), with automation guaranteeing efficiency and consistency in access management. To ensure that users are automatically granted the proper access based on their roles and responsibilities, implement RBAC policies and automate the assignment of roles to users.

# Role Based Access Control (RBAC)

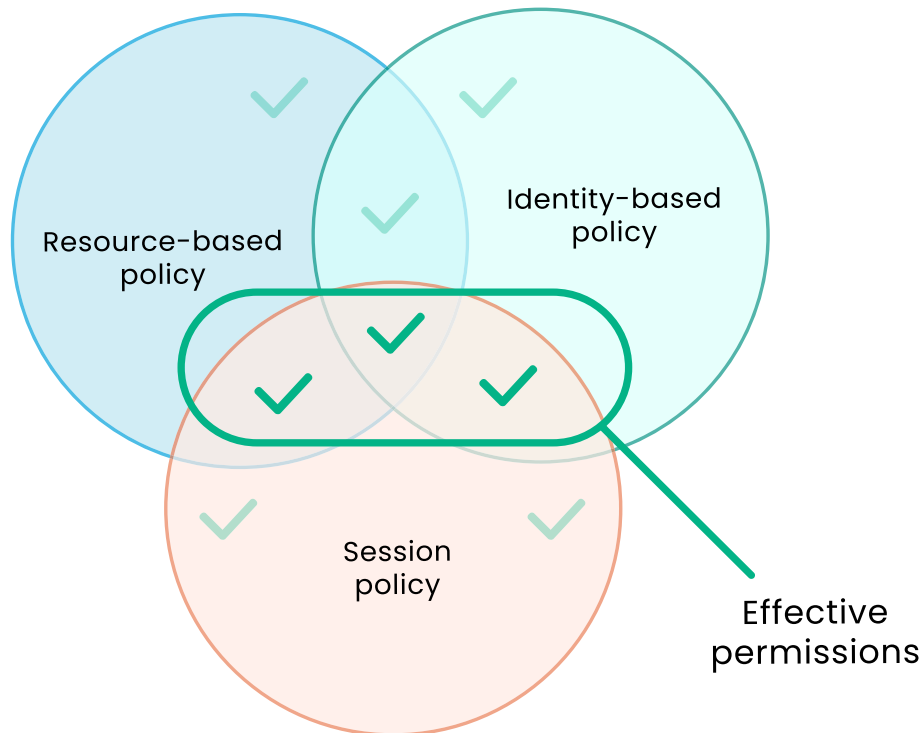


[Source](#)

In addition, utilizing [Identity and Access Management \(IAM\)](#) roles and automating their assignment to users using AWS IAM policies ensures that access aligns with predefined roles and responsibilities. This strategic approach optimizes access control, reducing the risk of unauthorized activities.

## 03 Permission Auditing and Monitoring

Continuous monitoring of permissions and access rights is essential for preemptive threat detection. Thus, automation streamlines the auditing process and enhances security. Implementing IAM roles and automating their assignment using [AWS IAM policies](#), such as resource-based, identity-based, and session policies, will assist in enforcing precise permissions that are in alignment with users' roles and responsibilities. Further, setting up alerts or triggers for suspicious or unauthorized access attempts allows for proactive responses to potential security threats.



[Source](#)

## 04 IAM Policy Generation and Enforcement

IAM policies define resource access rules, and automating their generation and enforcement ensures strict adherence to security standards. For resources and services, automating the creation of IAM policies will guarantee correctness and consistency in access control and reduce the risk of unauthorized access.

## 05 Policy Violation Remediation

Automate the detection of policy violations, such as excessive permissions, and initiate corrective actions swiftly. Remediating violations by automatically adjusting permissions or notifying administrators will ensure a proactive response to potential security risks. Enabling admin control bypass and timely alerting mechanisms further enhances security posture.

# Business Outcomes of Secure Platform Engineering

Implementing secure platform engineering has a substantial positive impact on business outcomes. It allows organizations to enhance efficiency, accelerate delivery timelines, and elevate workflow and process standards. Concentrating on focused and attainable initiatives to establish a robust foundation can also yield tangible benefits across people, the environment, and profitability.

Several key business outcomes associated with secure platform engineering include:

## Enhanced Efficiency

Consolidates various tools and technologies, simplifies management and control processes, and alleviates cognitive burdens on development teams.

## Accelerated Delivery Time

Enables swift identification and response to security threats with centralized monitoring and effective alerting systems.

## Improved Workflow and Process Standards

Fosters shared responsibility via alignment with DevOps principles, encouraging collaboration between development and operations teams and leveraging automation.

## Enhanced Developer Productivity

Simplifies DevOps adoption across the entire enterprise to scale, reducing technical debt and increasing developer productivity.

## Enhanced Security

Reduces vulnerabilities and boosts productivity in the software development lifecycle by ensuring that security is an essential component of the development process.



# The Future Landscape: Security-Driven Platform Engineering

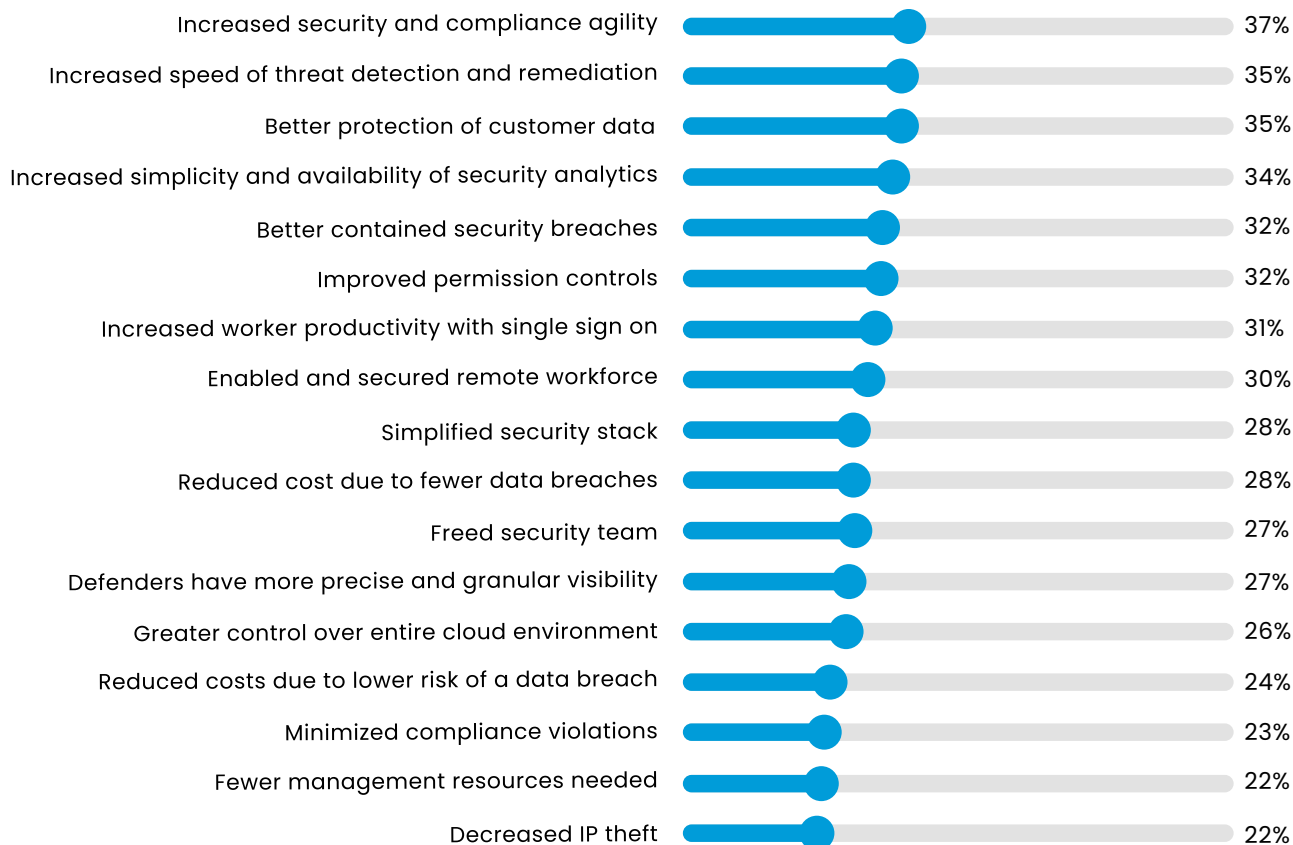
As the digital landscape continues to evolve, the future of platform engineering is tightly tied to security. Security-driven platform engineering emerges as the catalyst for ensuring resilience, compliance, and innovation.

Let's explore some trends shaping the future:

## 01 Zero Trust

[Zero trust](#) is a security framework assuming no inherent trust in any entity – internal or external. It mandates verifying identity and permissions and encrypting data for every user, device, and request. This strategy seeks to eliminate illegal access, stop data breaches, and prevent would-be attackers from moving laterally. The zero trust framework is frequently adopted by organizations because it improves security and compliance agility.

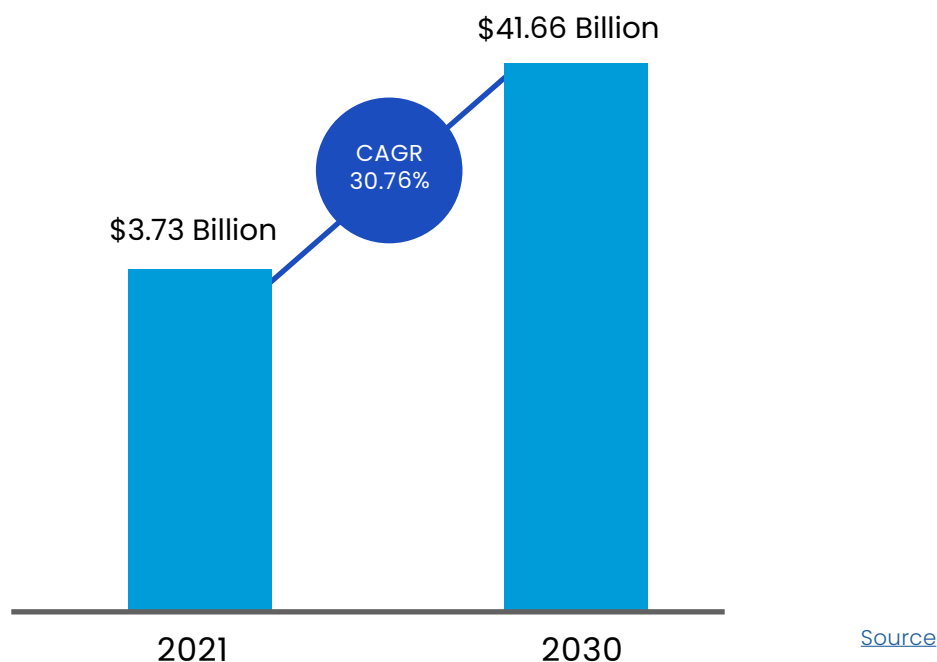
### Zero Trust Benefits



## 02 DevSecOps Integration

Security becomes a shared responsibility throughout the development lifecycle through its integration into the DevOps pipeline. Teams working on development, security, and operations are encouraged to collaborate through [DevSecOps](#). In fact, by 2030, the [DevSecOps market](#) is projected to reach a valuation of USD 41.66 billion, with a Compound Annual Growth Rate of over 30%.

### GLOBAL DevSecOps MARKET | 2022-2030



## 03 AI & Machine Learning in Threat Detection and Classification

In secure platform engineering, a groundbreaking trend involves incorporating AI and machine learning (ML) algorithms into threat detection. This integration empowers the analysis of extensive datasets, identifying patterns and anomalies to improve proactive security measures substantially.

Systems that use ML algorithms for AI-driven threat detection analyze big datasets in real-time, looking for trends and abnormalities that can point to security vulnerabilities. Threat detection powered by AI lowers [false positives by 40%](#), increasing the effectiveness of security procedures.

# Wrapping Up

Given that this unified approach can not only fortify organizations against evolving cyber threats, but simultaneously foster a culture of innovation, the fusion of security and platform engineering should be treated as absolutely imperative. By nurturing this symbiotic relationship between security and development, organizations can increase the pace of business and protect their assets from the threat of disruption.

Secure platform engineering is not just a methodology, but the *foundation* for resilient, efficient, and future-ready digital ecosystems.

## Contributing Author:

Stephen D. King, Chief Architect, Digital Solutions

# AHEAD

Combining cloud-native capabilities in software and data engineering with an unparalleled track record of modernizing infrastructure, we're uniquely positioned to help accelerate the promise of digital transformation.

## National Hubs

### CHICAGO

401 Michigan Ave.  
#3400  
Chicago, IL 60611

### NEW YORK

500 Fifth Avenue  
Suite 1500  
New York, NY 10110

### ATLANTA

1117 Perimeter Center  
W406  
Atlanta, GA 30338

### SAN FRANCISCO

2000 Crow Canyon Place  
Suite 250  
San Ramon, CA 94583