
The State of Policy as Code Report:

# Insights as Organizations Scale Authorization Policies

# Table of Contents

# Policy as code's role in the cloud environment

The conventional approach to authorization is outdated. Homegrown authorization policies are prone to errors and difficult to manage. In turn, they waste valuable time that developers could spend on more meaningful work, like building innovative products or fine-tuning new features. It's no wonder **three-quarters of developers say they waste time on mundane, manual tasks**.

**The cloud native environment drives greater speed, efficiency, and innovation than ever. Why shouldn't authorization function the same way?**

Policy as code empowers developers to define, enforce, and manage unified authorization policies seamlessly across the cloud native stack. However, old habits die hard — and new security processes require time to take hold.

Styra surveyed 285 developers and technical decision-makers across the U.S. in May 2023 to better understand the current landscape of policy as code — and where it's heading. Survey respondents are involved in managing authorization and compliance processes at their organizations.

Our research found that policy as code has become a crucial component of the cloud-native development landscape. Not only is policy as code expanding, it's providing greater benefits for organizations as they streamline, standardize, and scale authorization. However, most organizations still have work to do when it comes to scaling policy as code across their tech stack.

The following report provides greater insight into the current state of policy as code and showcases how organizations are using the technology to optimize development in the cloud environment — and how much further they still have to go.

# Key Findings

| | |
|---|---|
| **86%** | of developers and technical decision makers face some type of challenge with implementing authorization. |
| **55%** | of developers and technical decision makers say they write and enforce policy as code to manage authorization in their day-to-day work. |
| **46%** | of developers and technical professionals who use policy as code rely on Open Policy Agent (OPA) in some form. |
| **97%** | of developers and technical decision makers say policy as code is a vital component of building software efficiently in the cloud. |
| **30%** | of organizations are using policy as code in a significant capacity for most or all of their systems. |
| **83%** | of organizations plan to invest somewhat or significantly more in policy as code. |

Section 1

# Pain Points in the Development Lifecycle

# Authorization challenges are holding developers back

In today's cloud native environment, organizations face myriad challenges when managing authorization policies: 86% of all respondents said they face some type of challenge with implementing authorization. Despite its importance, authorization is often a point of friction and frustration that hinders developers from doing their best work.

**The top challenges organizations face?** Lack of alignment between teams is No. 1 according to more than one-third (34%) of respondents. That's followed by a lack of visibility into authorization (31%), a lack of consistent or centralized policy development (29%), and difficulty meeting security, compliance, or auditability requirements (29%).

**86% of all respondents said they face some type of challenge with implementing authorization.**

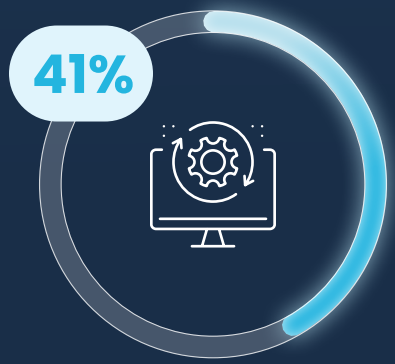
## The biggest challenges to implementing authorization

| | | |
|---|---|---|
| 1 | Lack of alignment between teams | 34% |
| 2 | Lack of visibility into authorization implementation, enforcement, monitoring, and reporting | 31% |
| 3 | Lack of consistent or centralized policy development and management lifecycle | 29% |
| Tie | | |
| 3 | Difficulty meeting security, compliance, or auditability requirements | 29% |
| 4 | Difficulty managing policies at greater scale or complexity | 26% |
| 5 | Slow/hard to modify and deploy existing authorization policies | 22% |
| Tie | | |
| 5 | Difficulty scaling authorization practices across the organization | 22% |
| 6 | Difficulty authoring policies | 21% |
| 7 | Limited tooling to implement, enforce and monitor authorization | 10% |

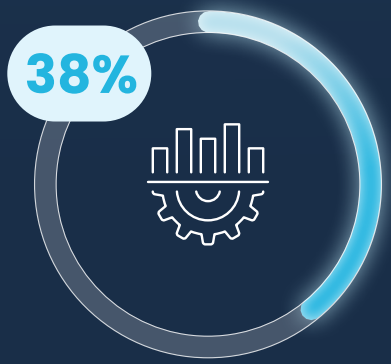
# How do authorization challenges impact different teams?

Developer and platform teams each deal with unique challenges impacting their work. For example, front-end developers cite a lack of alignment between teams as the biggest challenge while back-end developers name difficulties managing policies at greater scale or complexity. For teams that manage cloud infrastructure/DevOps, the top challenge is meeting security, compliance, or auditability requirements.

## Top challenge to implementing authorization cited by teams




**Front-end developers:**
Lack of alignment between teams




**Back-end developers:**
Difficulty managing policies at greater scale or complexity


**Cloud infrastructure/ DevOps team:**
Difficulty meeting security, compliance, or auditability requirements

# Policy as code offers a better approach

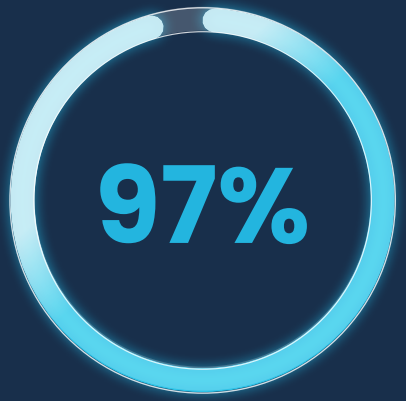
From inconsistent policies to a lack of coordination between teams, managing cloud native authorization is challenging enough as is. The conventional approach to authorization — homegrown policies — often exacerbates these challenges.

**Although the overwhelming majority (95%) of all respondents said homegrown authorization offers a highly customizable solution, in practice two-thirds cite major flaws in efficiency, security, and app performance.**

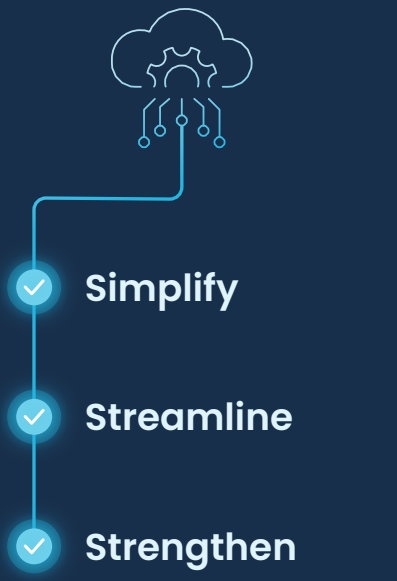
These less-than-enthusiastic reviews are a stark contrast to the overwhelmingly positive views of policy as code.

Policy as code allows organizations to automatically manage authorization policies separately from the software that enforces them, providing uniform access-control standards across infrastructure and applications. These solutions have emerged as a powerful alternative to simplify, streamline, and strengthen authorization in the modern development landscape.

It's striking (but not surprising) to see such strong support for policy as code. More than nine in 10 respondents agree that policy as code streamlines authorization, speeds up time-to-market, bolsters security, and makes life easier for developers.




of all respondents say policy as code is a vital component of building software efficiently in the cloud



## Developers and technical decision makers view on policy as code vs. homegrown authorization

### Policy as code

Agree policy as code is a vital component of preventive security and compliance at scale


Agree policy as code is a valuable approach to simplify and streamline authorization


Agree policy as code is a vital component of speeding up time to market


Agree policy as code makes work easier for developers


### Homegrown authorization

Agree homegrown authorization is inefficient and wastes developers' time


Agree homegrown authorization slows down time-to-market


Agree homegrown authorization leads to degraded app performance


Agree homegrown authorization jeopardizes security

Section 2

# The Rise of Policy as Code

# Policy as code is expanding (and so are its benefits)

As organizations work to improve authorization, it's clear that policy as code has gained traction as an effective method to manage authorization policies at scale and a crucial component of the modern development landscape.

More than half (55%) of developers and technical decision-makers say they write and enforce policy as code to manage authorization in their day-to-day work. It's even surpassing traditional methods. Meanwhile, homegrown authorization solutions — such as manual enforcement, custom systems, and hard-coded policies — are now only used by 41% of survey respondents. The remaining 4% of respondents write and enforce authorization another way.

**More than half (55%) of developers and technical decision makers say they write and enforce policy as code to manage authorization in their day-to-day work.**

## Policy as code is well-known even among respondents who don't use policy as code tools in their daily work:

- Only 3% of respondents have never heard about policy as code.


- 51% of respondents who don't use policy as code in their daily work have still tried out such tools in some capacity.


- Meanwhile, 39% have heard about policy as code and know how it works, and 8% have heard about policy as code but don't know how it works.

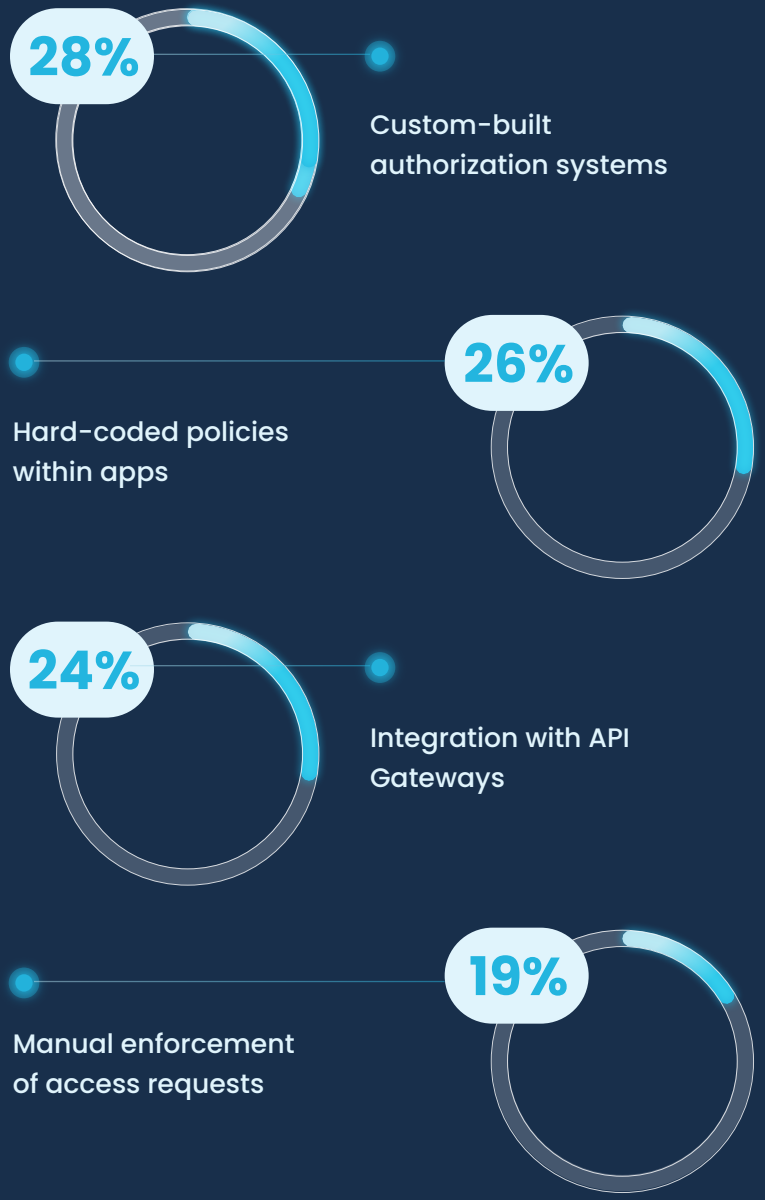
# Which teams are leading on policy as code — and which are lagging?

Teams managing architecture and solution integrations are the furthest ahead in using policy as code, with 64% saying they write and enforce policy as code to manage authorization in their day-to-day work.

And who's behind? Teams that manage identity and access management are lagging on incorporating policy as code in their everyday workflows, with less than half (25%) saying they write and enforce policy as code in their day-to-day work.

**64% of teams managing architecture and solution integrations write and enforce policy as code to manage authorization in their day-to-day work.**

## How do organizations that don't use policy as code manage authorization?



# Policy as code makes life easier for developers

What's driving greater adoption of policy as code? Organizations have turned to policy as code solutions for a variety of reasons. But identity and access management (IAM) (32%), zero-trust architecture (28%), and continuous authentication and access (28%) are ranked as the top business drivers that motivated policy as code adoption.

Those that have adopted policy as code are reaping a variety of benefits. The top benefits include greater consistency of policies (41%), faster implementation (38%), and centralization of the policy development and management lifecycle (36%).

Across the development lifecycle, organizations that are using policy as code tools are gaining real, tangible value — and transforming authorization from a headache to a jumpstart for agile, efficient, and flexible development.

**41% of respondents using policy as code say the the top benefit is greater consistency of policies.**

## Top business drivers that motivate adoption of policy as code

| | | |
|---|---|---|
| 1 | Identity and access management (IAM) | 32% |
| 2 | Zero-trust architecture | 28% |
| Tie | | |
| 2 | Continuous authentication and access | 28% |
| 3 | Attribute-based access control (ABAC)/ Fine-grained access control | 27% |
| 4 | Compliance management and auditing | 23% |
| Tie | | |
| 4 | Cloud migration | 23% |

## Top 10 benefits of using policy as code

1. Greater consistency of policy development and management lifecycle
2. Faster authorization policy implementation
3. Greater centralization of policy development and management lifecycle
4. Faster, more streamlined deployments
5. Better visibility into authorization implementation, enforcement, and monitoring
6. Easier to modify policies
7. Improved compliance
8. Faster detection of security problems
9. Improved ability to scale authorization policies across teams and systems
10. Improved collaboration between teams

# OPA & other open source tools help streamline authorization

Taking advantage of open source tools allows organizations to gain even more value from policy as code. Open source solutions provide the fastest, easiest, and most reliable way to define, enforce, and manage policy as code across the cloud native stack — all with the help of a community of open source contributors, developers, and experts from around the world.

And when it comes to open-source, Open Policy Agent (OPA) reigns supreme. Among the wide range of open source tools available today, OPA has emerged as the most popular option to create and manage policy as code. OPA (pronounced oh-pa) is an open source, general-purpose policy engine that provides a standardized policy language to unify authorization policies across the cloud native stack. Nearly half of respondents (46%) who use policy as code rely on OPA in some form.

There's widespread familiarity with OPA even among organizations that have yet to formally adopt policy as code solutions: 63% of respondents who don't use policy as code are familiar with OPA or plan to use it in the next 12 months, while 48% are familiar with or plan to use OPA Gatekeeper.

## Open source policy as code tools used most frequently, by respondents who write and enforce policy as code

| | | |
|---|---|---|
| 1 | Open Policy Agent (OPA) | 29% |
| 2 | OPA Gatekeeper | 17% |
| 3 | Sentinel | 15% |
| 4 | Kyverno | 10% |
| 5 | Chekov | 6% |
| Tie | | |
| 5 | OPAL | 6% |
| 6 | XACML | 3% |

Section 3

# The Path to Scaling Authorization

# The current landscape of policy as code

How do developers use policy as code to manage authorization today? Let's take a look at three areas: **environments, production requirements, and reliance.**

## Environments

As policy as code becomes widespread, organizations are applying policy as code solutions for both cloud native applications and cloud infrastructure. Most respondents rely on policy as code for cloud native applications (88%), while a sizeable portion use policy as code solutions for cloud infrastructure (67%).

**A deeper dive into these two areas underscores how versatile policy as code tools function:**

- **When it comes to application authorization**, 55% of respondents apply policy as code to secure API gateways, 52% apply solutions to implement role-based access control (RBAC), and 46% to implement attribute-based access controls, among numerous other uses.
- **With infrastructure authorization**, two-thirds of respondents apply policy as code to implement AWS CloudFormation infrastructure resource configuration checks, while 60% implement HashiCorp Terraform infrastructure resource configuration checks. Meanwhile, 45% use policy as code solutions to implement infrastructure compliance monitoring.

**88% of respondents rely on policy as code for cloud native applications while 67% use policy as code solutions for cloud infrastructure.**

## Production requirements

For policy as code to be deployed in production, teams are most focused on continuous integration and deployment, compliance management, and automated testing and validation.

## Top production requirements for policy as code

| | | |
|---|---|---|
| 1 | Continuous integration and deployment | 39% |
| 2 | Compliance management | 37% |
| 3 | Automated testing and validation | 33% |
| 4 | Role-based access control/Attribute-based access control | 32% |
| 5 | Version control | 27% |

## Reliance

Today, 87% of respondents say their organizations use policy as code in production. But most haven't yet scaled policy as code across organizational processes. Less than one-third (30%) of organizations are using policy as code in a significant capacity for most or all of their systems. Meanwhile, 32% of organizations are using policy as code in production but in a limited capacity for non-mission critical systems, while one-fourth (25%) are using policy as code in production for several systems.

**Less than one-third (30%) of organizations are using policy as code in a significant capacity for most or all of their systems.**

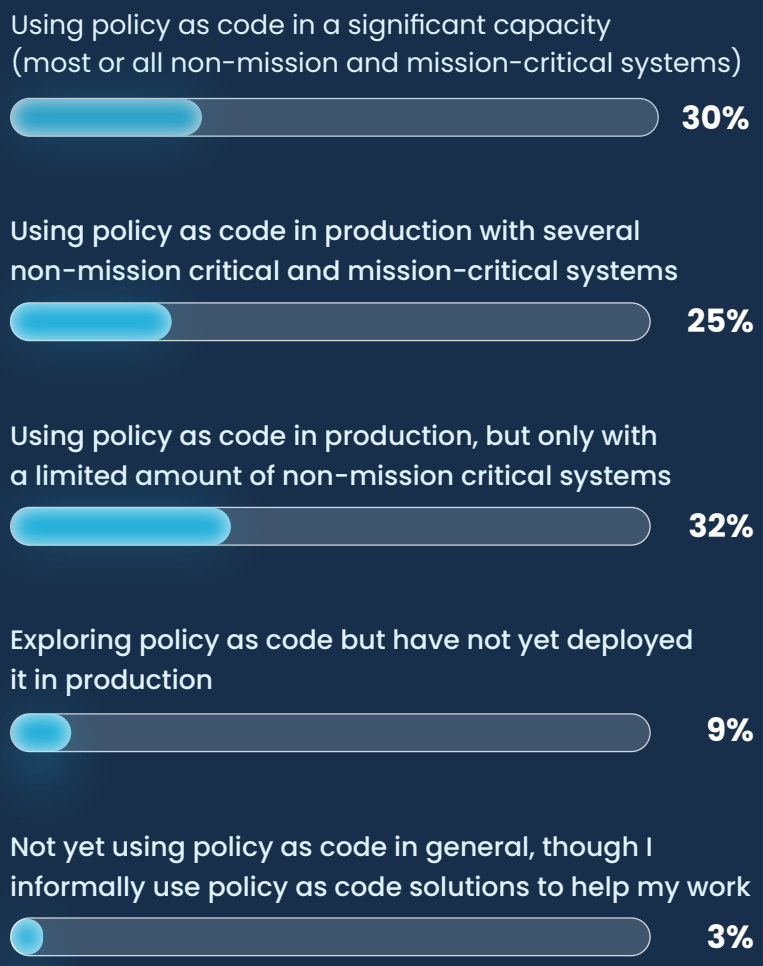
# How does policy as code adoption differ amongst organizations?

**High-revenue organizations show a more sophisticated, mature approach to policy as code compared to lower-grossing companies.**

Organizations that make $500 million or less in revenue are most likely to use policy as code in production, but only with limited non-mission critical systems. On the flip side, organizations making $501 million or more are most likely to use policy as code in a significant capacity, with both non-mission and mission-critical systems.

- **$1 million-$50 million:** 33% are using policy as code in production, but only with a limited amount of non-mission critical systems
- **$51 million-$500 million:** 35% are using policy as code in production, but only with a limited amount of non-mission critical systems
- **$501 million-$1 billion:** 40% are using policy as code in a significant capacity
- **More than $1 billion:** 44% are using policy as code in a significant capacity

## How developers and technical decision makers use policy as code



# Organizations are making progress, but still have a way to go

Organizations have spent years stuck in the early stages of adopting, implementing, and investing in policy as code. But even as many organizations start to show signs of maturation, they still have work to do when it comes to scaling policy as code tools.

That's unsurprising given that policy as code is still a relatively new tool for most organizations. Over half of respondents (51%) have only adopted policy as code in the past two years, including 14% who have used policy as code for less than a year and 37% for 1-2 years. Meanwhile, one-third (33%) for 3-5 years, and 16% have used policy as code tools for more than 5 years.

---

## But the trend is clear:

Although organizations still need to make significant headway when it comes to scaling solutions, they are making good progress in their policy as code journeys. And they have no plans to stop anytime soon: 83% of organizations plan to invest somewhat or significantly more in policy as code, while virtually no organizations plan to invest less.

**83% of organizations plan to invest somewhat or significantly more in policy as code, while virtually no organizations plan to invest less.**

Section 4

# Unlocking Policy as Code's Full Potential

# Organizations face barriers to scaling policy as code

Developers value policy as code. At organizations that use policy as code, and those that don't, nearly all respondents are eager to build on their knowledge or use of policy as code.

Among all respondents, 43% are interested in learning more about policy as code, while 21% are interested in trying out policy as code solutions. Meanwhile, more than one-third (35%) are interested in using policy as code more often or for more use cases.

**In fact, policy as code is viewed as a strategic priority for 94% of organizations.**

But despite recent progress, organizations are also running into challenges — and they're preventing developers from unlocking all the benefits of policy as code. **Nearly nine in 10 (88%) respondents face barriers preventing them from using or scaling policy as code.**

## Biggest barriers preventing the use or expansion of policy as code

| | | |
|---|---|---|
| 1 | Complexity of digital transformation projects | 28% |
| 2 | Organizational resistance to change | 27% |
| 3 | Lack of awareness of policy as code | 26% |
| 4 | Policy as code is too technically complex | 25% |
| Tie | | |
| 4 | Concerns about security and compliance | 25% |
| 5 | Limited resources/tooling to implement policy as code | 22% |
| Tie | | |
| 6 | Vendor or tooling lock preventing change | 22% |
| 7 | Integration challenges with existing systems and processes | 21% |
| 8 | Declarative vs. imperative programming challenges | 19% |
| 9 | Lack of budget | 16% |

# So what's holding organizations back?

The complexity of digital transformation, organizational resistance to change, and lack of awareness about policy as code have emerged as the top challenges, according to more than a quarter of respondents.

Common performance challenges are also getting in the way, especially given that most organizations are in the first few years of implementation. For example, the most common performance challenges/bottlenecks of policy as code include writing efficient policies as code (52%), data fetch latency to/from an authorization service (45%), and policy evaluation at runtime or compile time (38%).

**52% of respondents say the most common performance challenges/ bottlenecks of policy as code include writing efficient policies as code.**

## Top performance challenges to policy as code

| | | |
|---|---|---|
| 1 | Writing efficient policies as code | 27% |
| 2 | Data fetch latency to/from an authorization service | 26% |
| 3 | Policy evaluation at runtime or compile time | 25% |
| Tie | | |
| 3 | Request latency on the path to/from an authorization service | 25% |
| 4 | The size and scale of my authorization data | 25% |

# How do performance challenges differ among organizations?

These problems cut across organizations — and the challenge of writing efficient policy as code only seems to grow for organizations with increased revenue.

## Top answer choice by revenue:

1

**$1 million-$50 million:**
Writing efficient policies as code (50%)

2

**$51 million-$500 million:**
Writing efficient policies as code (55%)

3

**$501 million-$1 billion:**
Writing efficient policies as code (62%)

4

**More than $1 billion:**
Interestingly, employees at organizations that make more than $1 billion in revenue — the highest of all respondents surveyed — cited a lack of budget as the biggest barrier using or expanding the use of policy as code at their organization (38%).

# The imperative for organizations

It's critical that organizations address challenges that hold teams back from working together, strengthening security, and creating consistent, centralized authorization policies at scale.

With rising cyber attacks, policy as code tools are a valuable part of organizations' security configuration: 96% of respondents say policy as code is vital to building, securing, and maintaining cloud infrastructure. Open source tools and policy as code management platforms offer effective solutions to many of the common challenges holding back organizations from scaling policy as code.

By leveraging the right technology tools, support, and third-party expertise, organizations can simplify digital transformation and streamline authorization implementation.

**It's critical that organizations address challenges that hold teams back from working together**


of respondents say policy as code is vital to building, securing, and maintaining cloud infrastructure.

# Accelerating your policy as code journey

Developers can't afford to continue wasting precious time on homegrown policies that confuse teams, muddle visibility, and complicate authorization. In today's cloud native environment, there's no reason they need to.

Policy as code empowers developers to unlock the full potential of the cloud environment, making the modern development life cycle more efficient, effective, and secure. It also makes life easier than ever for the developers who power it.

But despite the value, benefits, and opportunities that policy as code brings, there are common roadblocks that are holding companies back from scaling these tools. Organizations that use open source tools to manage policy as code not only alleviate many of their current challenges, they set themselves up for future growth and long-term success.

Overcoming issues with complexity, resistance to change, and other barriers can feel overwhelming, but organizations don't need to undergo this process alone.

At Styra, we provide the technology tools and expertise for every team to access, use, and scale policy as code tools, enabling developers and platform teams to focus on what they do best: Creating cutting-edge features, better functionalities, and innovative products.

styra

**Contact us**
to learn more about how we can support your organization's policy as code journey.

# Methodology

To better understand the state of policy as code, we surveyed 285 developers and technical decision makers in the U.S. in May 2023 about their understanding and use of policy as code.

Respondents included full-time employees working at a company that uses cloud native tools and technologies in its operations, involved in managing authorization and compliance processes. All respondents were at least 22 years or older and worked at companies with 250 or more employees.

# About Styra

Styra enables enterprises to define, enforce, and monitor policy as code across their cloud native environments. With a combination of open source (Open Policy Agent) and commercial products (Enterprise OPA and Styra Declarative Authorization Service), Styra provides policy authorization enabling security, operations, and compliance to protect applications, as well as the infrastructure they run on. Styra helps developers, DevOps, and security teams mitigate risks, reduce human error, and accelerate application development.

**Learn more at Styra.com**