privado

Privacy Code Scanning

What is it, why do you need it, and how to get started

Vaibhav Antil
Co-founder and CEO, Privado

Foreword by
Nishant Bhajaria

A WHITE PAPER BY PRIVADO.AI

FOREWORD

Nishant Bhajaria

Nishant Bhajaria is a leading technical privacy expert and the author of *Data Privacy: a runbook for engineers*.

The phrase "Privacy Enhancing Technologies" (PET) is ubiquitous in the data security and data privacy spaces. Those tools think of privacy as a risk-compliance vector. I have a different acronym for PET: “Productivity Efficiency Transparency.

And the reason I was very excited to be asked to write this foreword is because that is exactly what a privacy code scanning solution offers.

This approach enables, rather than blocks, engineers. Risk, data misuse, and inappropriate access of data often occur at scale due to problems that originate in the code. Engineers often do not realize the downstream consequences of their code, and then they are asked to clean up data privacy issues.

Privacy code scanning works as the invisible but ubiquitous ally, helping flag potential issues even as engineers write code. This capability continually connects logic, code, storage, and transit as part of a larger whole. By doing so, it helps detect risk in real-time and identify patterns over a span of time. Privacy teams will then be able to create efficiency and transparency rather than depend on unreliable, manual assessments.

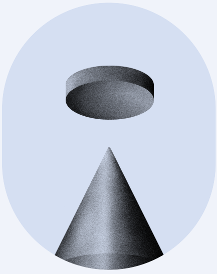
Put simply, your product and engineering teams need a degree of certainty and structure to drive innovation and engagement. Instead of using compliance as a brake, you can now use privacy code scanning as an accelerator.

Table of Contents

Introduction

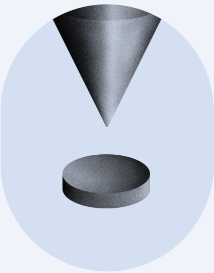
By year-end 2024, Gartner predicts that 75% of the world's population will have its personal data covered under modern privacy regulations. Just since the beginning of 2023, three countries (Switzerland, South Korea, and Saudi Arabia) and five U.S. states have put new privacy regulations into effect, including the pivotal California Privacy Rights Act (CPRA). These regulations pose real concerns around data privacy and data security for businesses.

Companies have poured considerable resources into their people, processes, and technology to keep up with these laws. Yet, many are still grappling with the fundamental issue of "where is my data?", resulting in steep fines from regulatory bodies like the FTC and EU regulators for improper data usage and sensitive data breaches.



Despite ongoing efforts, companies still struggle with privacy issues when they rely on manual assessments or focus on data storage. Unfortunately, these approaches fail to address the root of the problem. Manual assessments don't align with the realities of data processing in engineering, resulting in incomplete, inaccurate, and outdated information. Similarly, data discovery efforts only scratch the surface, offering no control over data collection, usage, or sharing. To truly address these issues, we need a new approach that tackles the fundamental source of the problem: the code itself.

We need a new approach that tackles the fundamental source of the problem: **the code itself.**



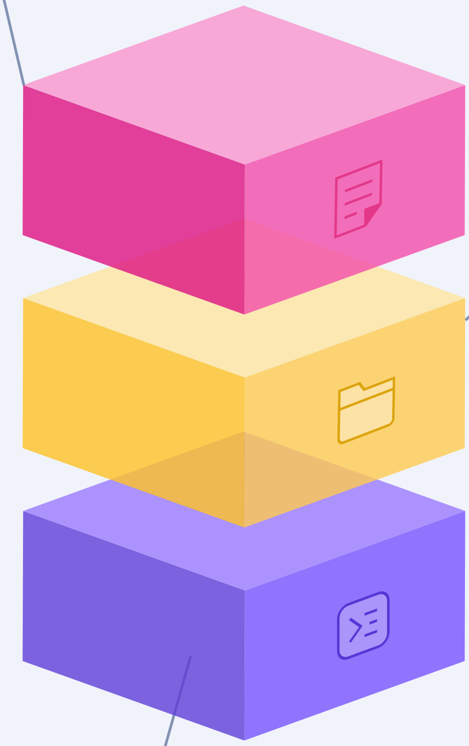
The code is where developers define the data collection, sharing, usage, and storage logic. By implementing privacy code scanning, companies can bridge the gap between privacy and engineering. This innovative solution provides complete visibility into the data lifecycle, including collection, flows, sharing, and storage. It also enables governance of data usage and allows for continuous privacy compliance within the product development lifecycle. In this white paper, we delve into the challenges of operationalizing privacy for engineering and describe our solution to this ongoing problem. Privacy code scanning is the missing piece of the puzzle, and it can serve as a game-changer for global technical privacy programs.

Approaches to Privacy

REACTIVE

Manual assessments

Data maps created from assessment questionnaires are surface-level and don't align with the realities of data processing in engineering, resulting in incomplete, inaccurate, and outdated information.



REACTIVE

Data store level

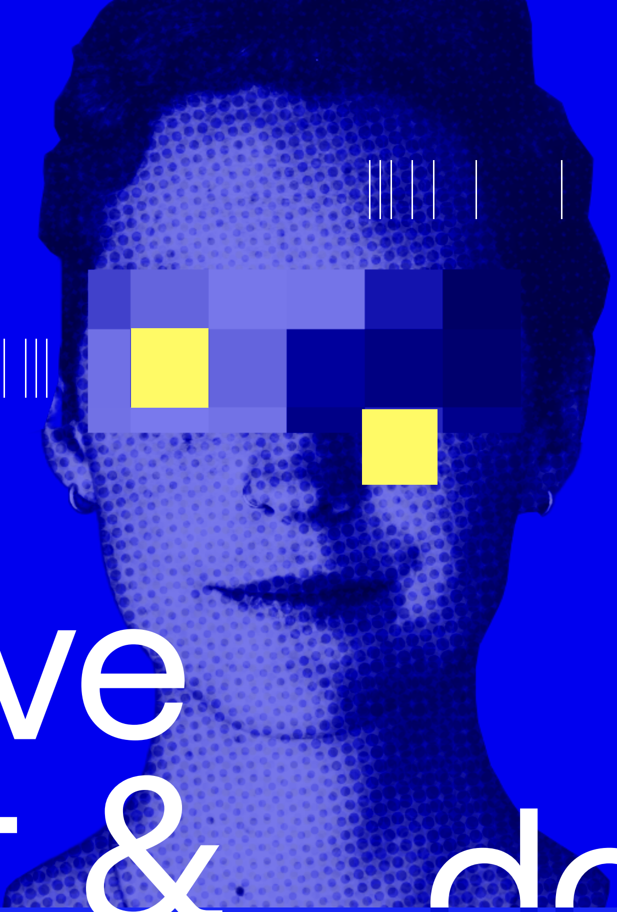
Data discovery efforts only scratch the surface, offering no visibility or governance over data collection, usage, or sharing.

PROACTIVE

Source code level

Enables complete visibility into the data lifecycle, including collection, flows, sharing, and storage. Source code scanning also provides continuous privacy governance by identifying policy violations as they appear in the codebase.

CHAPTER 01



move fast & don’t break things.

Sounds familiar? “Move fast and break things” was Meta’s rallying cry in the 2000s, something that has changed since the 2018 Cambridge Analytica scandal and fines. While this slogan has lost its popularity, moving fast is going nowhere. This creates tension for companies to innovate at speed with trust and privacy baked in from the get-go.

This creates tension for companies attempting to innovate at speed and build with privacy in mind.

Let’s look at the problem →

Let’s look at why it is hard to operationalize privacy for engineering at scale and speed

1 Distributed Data Processing

Modern software isn’t just written anymore; it’s assembled. A typical tech company has several user-facing apps or websites coupled with hundreds of internal services to achieve scale and reliability. For example, Netflix famously has over 1,000 services in production. Each service has its own data processing lifecycle that can use, share, store and leak personal data. Manually mapping data across these apps and services has proven to be a never-ending exercise.

2 Continuous Delivery

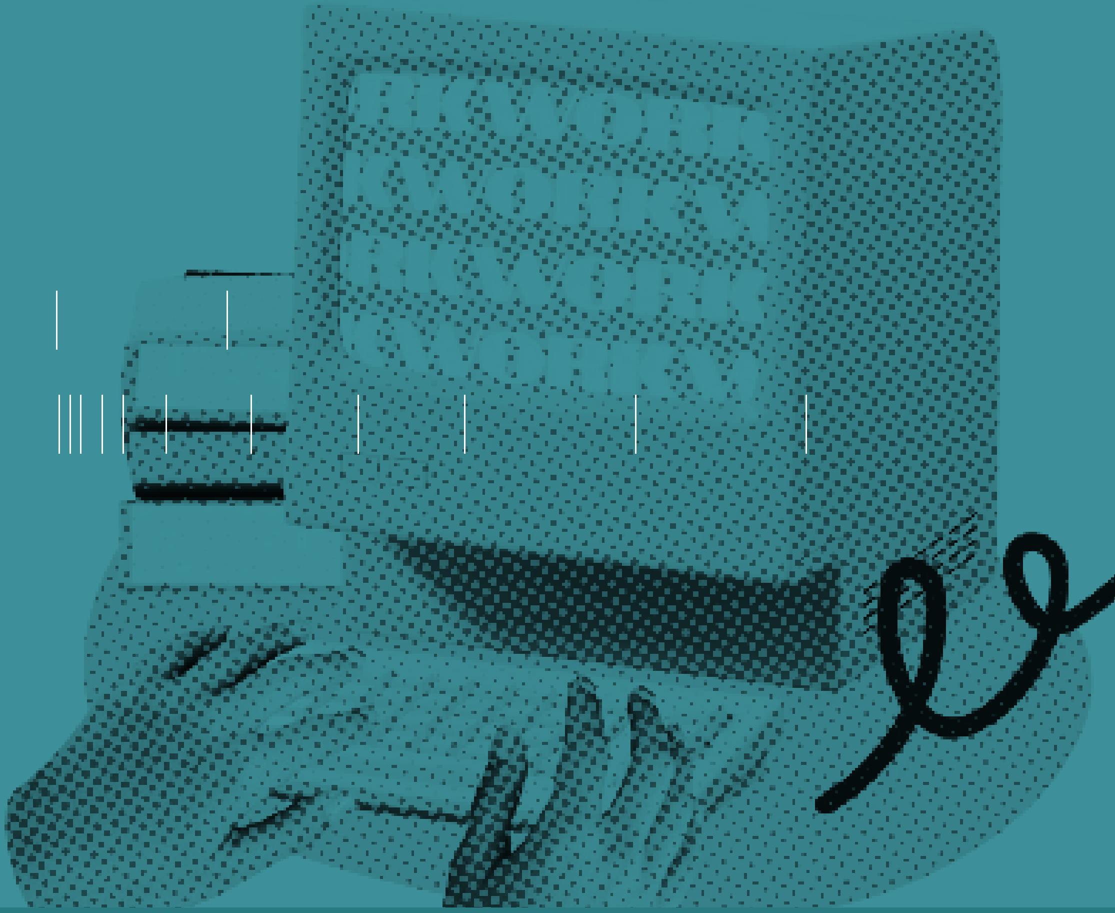
Tech companies today follow Agile product development where product and engineering teams continuously launch features, iterate with users, and drive value at unprecedented speed. The “shipping fast” culture makes it challenging for privacy and security teams to stay on top of data processing and ensure new product changes don't break privacy.

3 The Privacy Engineering Gap

Engineering teams are instructed to innovate and release new products at a fast clip. On the other hand, privacy teams must rigorously evaluate every aspect of each upcoming product and feature to ensure compliance with privacy policies. This can lead to a disconnect between the teams, delay new product releases, and create privacy debt.

> “
>
> We do not have an adequate level of control and explainability over how our systems use data, and thus we can’t confidently make controlled policy changes or external commitments. Yet, this is exactly what regulators expect us to do, increasing our risk of mistakes and misrepresentation.
>
> **Leaked internal document at Meta**

> “
>
> Twitter doesn't understand how much data it collects, why it collects it, and how it's supposed to be used.
>
> **Peiter “Mudge” Zatko –** Former Head of Security at Twitter



how do we operationalize privacy for engineering?

Requirements for a solution

Operationalizing privacy presents what we like to call the **"problem of plenty."**

Organizations have plenty of engineers with access to plenty of data, collectively creating features quickly and adding plenty of privacy debt. This overall privacy debt is the unwanted side effect of the bottom-up distributed culture that drives innovation.

To address this problem, we must make privacy solutions available at scale to engineers before their code creates more privacy issues.

Let's look at the requirements for building such a solution.

Complete Data Flow Visibility

As data processing is distributed, the solution should give complete data visibility across all products and applications. It is important that the solution is fast and automated so that privacy and security teams can focus on ensuring compliance and protecting data. To get this visibility the solution should:

Discover Data Processing Activities

Automatically discover processing activities in engineering across mobile apps, websites, APIs, services, data pipelines, and SDKs.

Autogenerate Data Flow Diagrams

Identify data flows to third parties and internal infrastructure including data stores, log files, inter-service flows, and messaging queues.

Detect Product Privacy Issues

Find privacy issues in the product such as excessive data collection, sensitive data sharing, or data leakages.

Proactive Privacy Control

As engineering teams continuously ship new features, the solution should continuously monitor these changes, detect drift in data use, and prevent product privacy issues from going live. For proactive privacy, the solution should offer:

Continuous Privacy Assurance

Continuously scan code for changes to update data maps, detect changes in data use, and find any privacy issues in the code.

Privacy by Code

Enforce privacy rules and policies in the software development lifecycle for data privacy laws like GDPR and CCPA, frameworks like PCIDSS, and even the company’s internal policies.

Developer-Friendly Privacy Workflows

Finally, to ensure developer adoption of the solution, we need to ensure it is available to developers within the tools they use and that it can guide and educate them as they write code.

The solution should:

Empower Developers

Give developers feedback on privacy issues as they are writing code with guidance on how to fix them.

Offer Just-in-Time Privacy Training

Offer privacy training to developers as they use data.

Integrate with Dev Tools

Integrate with source code management (SCM) tools, CI/CD pipelines, and developer portals.

CHAPTER 03

the current approach to privacy

How current approaches compare to our requirements.

Let’s look at how current approaches to running privacy programs compare to our requirements.

Manual Assessments

In this approach, privacy teams first define business processes or processing activities by interviewing key stakeholders and then conduct privacy reviews like RoPA assessments and PIAs for these processing activities. This process can be done on excel sheets or using privacy management tools. On paper, this approach gives complete visibility of a company's data lifecycle, but the reality is far from that.

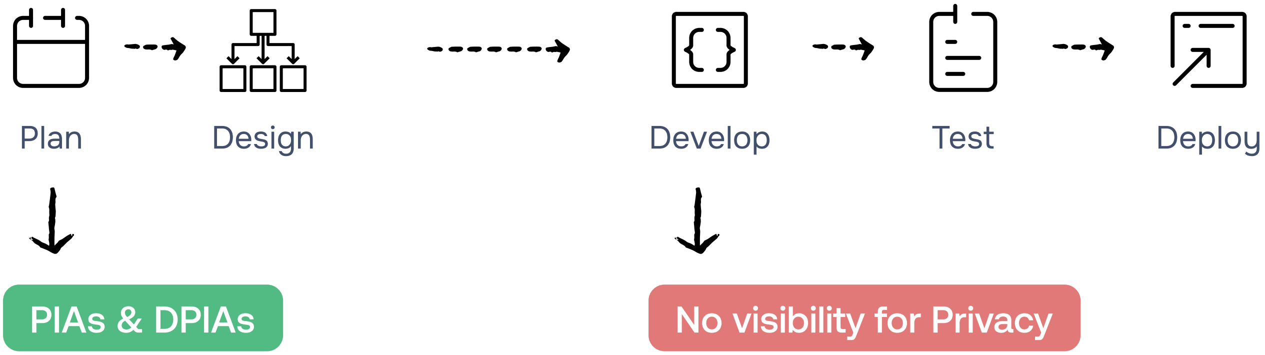
The first issue companies face here is that defining these processing activities is a big task that creates a lot of confusion. Especially when it comes to engineering; data processing happens through applications, services, APIs, and data pipelines & not a business process. In software development, there is no linear process for collecting, storing, and sharing personal data. Personal data is processed through many applications, services, APIs, and data pipelines simultaneously.

The next problem with manual assessments is that privacy teams depend on manual input and interpretations from product teams.

Lastly, since data maps take such a long time using manual assessments, they are typically done only once a year. This means data maps quickly become outdated because engineering teams are typically shipping new code weekly or biweekly.

To comply with the Privacy by Design approach, companies have to do a privacy review for new changes at the design stage. While this is possible for top-down planned features, many features are built bottoms-up that never get to the privacy review stage. Even if design reviews are conducted for all new changes, development can still deviate from the original design, causing privacy gaps and issues to emerge.

Privacy must have a seat at the Development Table



Overall this approach is fast to start, creates the initial documentation but is manual, takes a long time, does not scale, and quickly becomes out of date.

Data Discovery

With the data discovery approach, you scan structured and unstructured data across data stores and some third parties to find personal data. Data discovery tools can scan column names and the actual data, using ML/AI techniques to discover and classify data.

This approach provides visibility into where the data is stored, helps process DSAR requests, and improves data protection. As part of data governance initiatives, data discovery can also help companies improve data quality, privacy, and security.

While data discovery can be fully automated, it only focuses on data storage and therefore, misses how personal data is collected, used, and shared.

Data discovery can feel like playing whack-a-mole, where you are always reacting to personal data popping up in data stores with no control over the source of the problem.

Once discovery is done, privacy teams still struggle to identify which teams use this data and still lack the data flows needed to accurately create RoPAs, conduct PIAs and find privacy issues.

Doing data discovery alone can create a false sense of maturity in privacy programs because you know the data you have in data stores. But in reality, you don't understand how your data is being used, you don't know how it is being shared, and you don't know how it is being collected. These gaps lead to privacy issues such as:

Excessive Data Collection

Collecting precise location data over coarse location data has led to fines by CNIL and EU regulators.

Sensitive Data Sharing

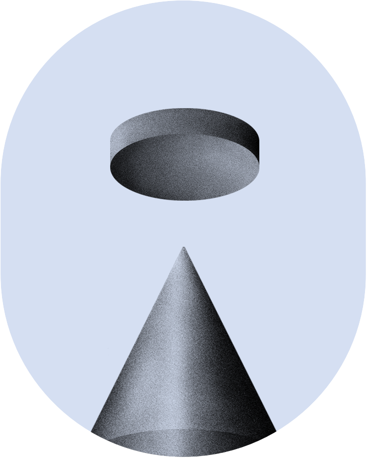
Health and financial data flows to Meta pixels and third parties has led to FTC enforcements and health breach notifications to customers and the media.

Misuse of Personal Data

Failure to limit the use of data for collection purposes led to an FTC fine on Twitter.

Non-Compliance in Data Processing

Personal data being used in automated decision-making with no oversight from privacy teams has led to GDPR fines.



While the data discovery approach is automated and helps companies to know their stored data, it has a big gap in the data lifecycle, especially regarding data flows and data use. These gaps make the data discovery approach reactive since it does not offer any controls and lacks an interface with developers.

Privacy Code Scanning

The privacy code scanning approach starts with the code. This is where business logic for data collection, storage, sharing, use, and processing is written by developers. First, by scanning all code repositories, privacy code scanning discovers all repositories processing personal data, which serve as processing activities for engineering.

Since each of these code repositories has a development team that owns it, privacy teams can easily get more information about data processing and data flows to complete compliance activities and identify how to fix any privacy issues in the code. Finally, by continuously scanning code for changes, privacy code scanning can update data maps automatically, identify any drift in data use, and prevent code changes that violate policies from going live.

When evaluating privacy code scanning solutions, each should have the following capabilities:

Data Discovery & Classification

Discover personal data processed across products and applications built by developers.

Data Flows Discovery

Generate data flows from collection points to destinations such as third parties and internal infrastructure, including data stores, APIs, messaging queues, and log files.

Monitoring & Prevention

Continuously scan code for changes and prevent privacy issues from going live.

Developer Workflows

Provide workflows for privacy teams to interface with developers to fix privacy issues and generate privacy compliance evidence.

	DATA DISCOVERY	PRIVACY CODE SCANNING
Data mapping method	Scan data stores	Scan codebase
Time to implement	6-12 months	< 3 days
Data visibility	Retrospective	Real-time
Data collection, usage, and sharing analysis	🔴	🟢
Automated discovery of new data flows	🔴	🟢
Unstructured data discovery	🟢	🔴
Pre-filled and self-updating assessments	Limited	🟢
Automated validation that data flows are compliant	🔴	🟢
Privacy integrated in software development lifecycle	🔴	🟢
Continuous monitoring for data leaks and third party data sharing	🔴	🟢
Privacy training and remediation embedded in dev workflows	🔴	🟢

🔴 No 🟢 Yes

CHAPTER 04

who can benefit from privacy code scanning?

Code scanning can deliver immediate benefits to organizations with fast-paced product development and modern infrastructures. Multicloud environments, microservices, and bottoms-up product development create the perfect setting for data sprawl and drift that can only be truly wrangled with privacy code scanning. With minimal technical lift, privacy teams can gain the data visibility and control required to maintain privacy policies at scale.

Some enterprises have built their privacy code scanning tools internally.

Meta has built Pysa and Zonoclan, OSS privacy code scanning tools, to detect privacy issues in large code bases. Pysa creates data flows and checks against technical privacy policies to identify privacy issues before they hit production. For example, Instagram uses privacy code scanning to ensure location data is only used to calculate status but is never stored.

However, it is costly and inefficient for every organization to build and maintain tooling for every business need, especially when viable solutions are readily available. Privacy code scanning is a nascent approach, but it has quickly risen to become business critical for organizations where data visibility and privacy automation has scaled and matched the speed of innovation. Here Technologies, a leader in mapping and location services, is one such enterprise that experienced the immediate benefits of privacy code scanning. See their story here.

privado

Privado is a privacy code scanning solution with open source and enterprise-ready offerings that give privacy and security teams unparalleled visibility into data flows, the ability to enforce privacy policies at the speed of product development, and increased collaboration with development teams. With Privado, enterprises can embed privacy checks in their Software Development Lifecycle. **Schedule a demo to learn more.**

TRUSTED BY TEAMS AT

headspace ZEGO snap finance

Docplanner Group trendyol here

The Author

Vaibhav Antil is the CEO and founder of Privado.ai, a privacy tech startup, and holds a CIPM from the IAPP. He previously led product teams at Gaana, the largest music streaming platform in India, working in sync with privacy, legal, and engineering teams, and co-founded BC Jukebox, which was later acquired by Gaana.

With Privado, enterprises can embed privacy checks in their Software Development Lifecycle. Visit **privado.ai** for more details.

✉ hello@privado.ai

in linkedin.com/company/privado-ai/