privado

Privacy Code Scanning

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

Vaibhav Antil
Co-founder and CEO, Privado

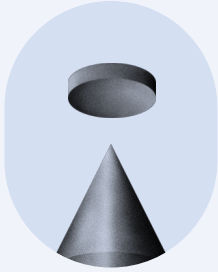
A WHITEPAPER BY PRIVADO.AI

Table of Contents

Introduction

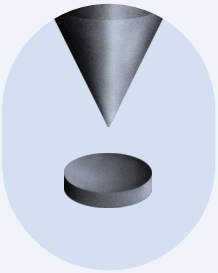
By year-end 2024, Gartner predicts that 75% of the world's population will have its personal data covered under modern privacy regulations. These regulations pose real concerns around data privacy & data security for businesses.

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 implement process-level solutions or focusing on data storage. Unfortunately, these approaches fail to address the root of the problem. Process assessments don't align with the realities of data processing in engineering, resulting in incomplete, inaccurate, and outdated information.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.**



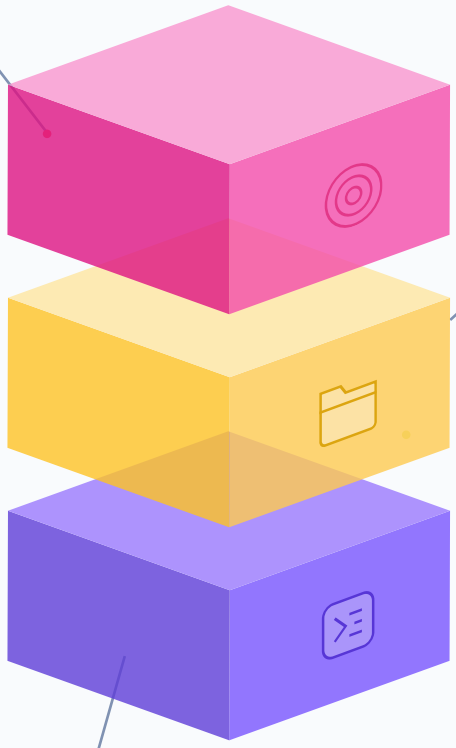
The code is where developers define the data collection, sharing, usage, and storage logic. By implementing "Privacy Code Scanning," companies can bridge the gap between privacy and engineering. This innovative solution provides complete visibility into the data lifecycle, including collection, flows,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

Approaches to Privacy

REACTIVE

Process level

Manual processes dont 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 control over data collection, usage, or sharing.

PROACTIVE

Source code level

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 as code.

CHAPTER 01

move fast & don't break things!

Sounds familiar? Move Fast & Break Things! was Meta's rallying cry in the 2000s, something that has changed lately with Cambridge Analytica Scandal & fines. While this slogan has lost its popularity, moving fast is going nowhere. This creates tension for companies to innovate at speed with trust & privacy baked in from the get-go.

Let's look at the problem

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

1 Distributed Data Processing

Modern software isn’t just written anymore; it’s assembled. A typical tech company has some user-facing apps or websites coupled with hundreds of internal services to achieve scale and reliability, for example, Netflix famously has over 1000 services in production. Each service has its own data processing lifecycle using, sharing, storing and leaking personal data. Manually mapping data across these apps & services is a never-ending exercise.

---

2 Continuously Changing

Tech companies today follow Agile product development where product & engineering teams continuously launch features, iterate with users & drive value at unprecedented speed. ‘Shipping Fast’ culture makes it challenging for privacy & security teams to stay on top of data processing as it’s always changing & to ensure new product changes don't break privacy.

3 The gap between Privacy & Engineering

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 and delay new product releases or worse, 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 And yet, this is exactly what regulators expect us to do, increasing our risk of mistakes and misrepresentation.

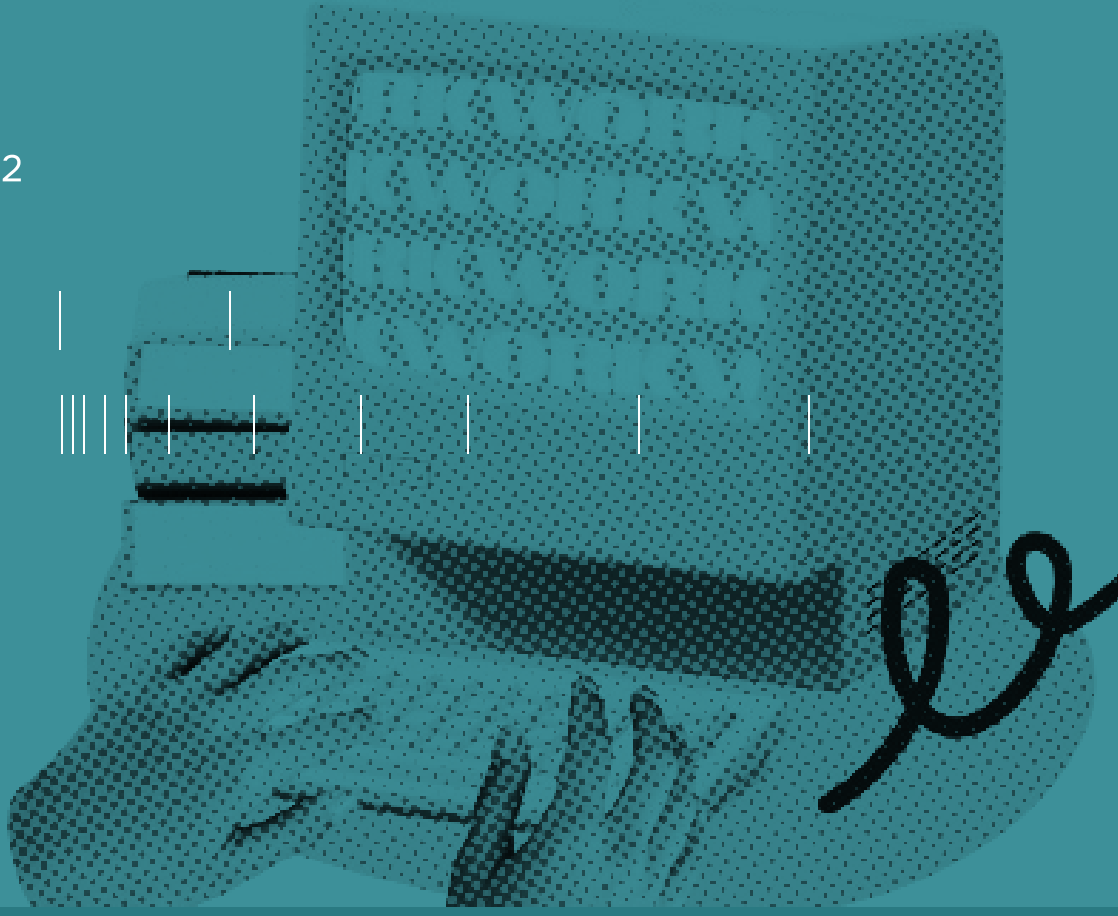
Leaked internal document at Meta

“

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

CHAPTER 02



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 available at scale & velocity to the engineers before their code creates any privacy problems.

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 these products & applications. It is important that the solution is automated and time to get this visibility is low so that privacy & security teams can focus on actual compliance & protecting data. To get this visibility the solution should:

Discover Data Processing

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

Auto-Draw Data Flow Diagrams

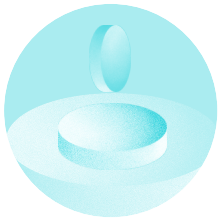
Identifies data flows to third parties & internal infrastructure including data stores, log files, inter-service flows & messaging queues.

Detect Product Privacy Issues

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

Proactive Privacy Control

As engineering teams are continuously shipping new features, the solution should continuously monitor these changes, detect drift in data use & prevent product privacy issues to go live. For proactive privacy, the solution should:



Continuous Privacy Assurance

To continuously scan new code changes to update data maps, detect changes in data use of data & find any privacy issues in the code.

Privacy by Code

Enforce privacy rules & policies in the software development lifecycle for laws like GDPR, CCPA or frameworks like PCI-DSS and even the company's own 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.

Just-in-Time Privacy Training

Offer privacy training to developers as they use data

Dev Tool Integrations

Integrate with SCM tools, CI/CD pipelines, and developer portals.

CHAPTER 01

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 it.

The first issue that companies face here is that defining these processing activities is a big task that creates lots of confusion. Especially when it comes to engineering; data processing happens through applications, services, APIs, and data pipelines & not a business process.

The next problem with this approach is that it is dependent on manual input and interpretations made by the product teams in order for privacy teams to get answers.

Finally, it takes a long time for companies to do this exercise, so it is done once a year, whereas engineering is shipping new code changes weekly; this means your data maps are out of date with each release.

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.

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 Data Discovery, 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 and use ML/AI techniques to discover and classify data.

This approach gives companies visibility into where the data is stored and helps them with DSAR request processing and to better protect data. With data governance initiatives, data discovery can also help companies with data quality and privacy and security.

While this approach is fully automated, one of the big gaps is that it only focuses on data storage and misses collecting, using, and sharing personal data.

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 miss the privacy workflows to 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 like:

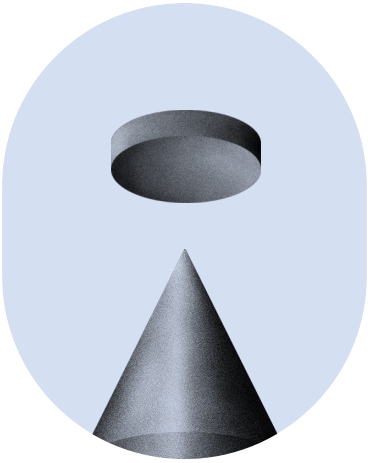
Excessive Data Collection	Collecting precise location data over coarse location data led to fines by CNIL & EU regulators
Sensitive Data Sharing	Health & Financial data flows to Meta pixels and third parties led to FTC enforcements & health breach

Mis-use of personal data

Failure to limit the use of data for collection purposes led to FTC enforcement & fine on Twitter

Non-Compliance in Data Processing

Personal data being used in automated decision-making with no oversight from the privacy teams led to a GDPR fine



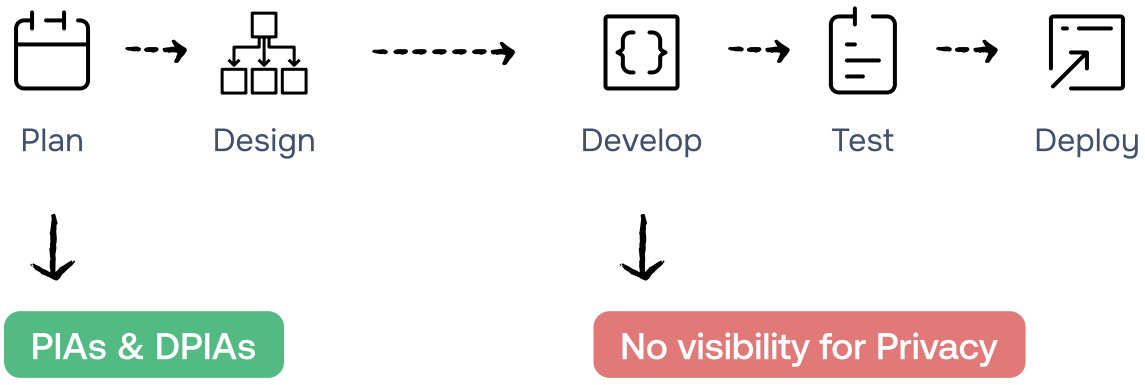
While the Data Discovery approach is automated and helps companies to know their stored data, it has a big gap in the data lifecycle, especially in data flows & data use. This makes the data discovery approach reactive since it does not offer any controls & also lacks an interface with developers.

Privacy Scanning

In the Privacy Scanning approach, you start at 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, a privacy scanner 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; data flows to finish compliance activities & fix any privacy issues in the code. Finally, by continuously scanning new code changes, privacy scanners update data maps automatically, identify any drift in data use, and prevent code changes that violate policies from going live.

Privacy must have a seat at the Development Table



As you evaluate a privacy scanner, it must have the following functionality:

Data Discovery & Classification	Discover personal data processed across products and applications built by developers
Data Flows Discovery	Generate data flow from collection points to data destinations like third parties and internal infrastructure, including data stores, APIs, messaging queues & log files.
Monitoring & Prevention	Continuously scan new code changes & preventing privacy issues to go live.
Developer Workflows	Should have workflows for privacy teams to interface with developers to fix privacy issues & generate privacy evidence.

CRITERION	USE ASSESSMENTS	DATA DISCOVERY	PRIVACY SCANNING
Processing Activity Discovery	●	●	●
Data Flow Diagrams	Manual	●	●
Product Privacy Issues	●	●	●
Unstructured Data Discovery	●	●	●
Monitoring	●	●	●
Privacy by Code	●	●	●
Integration into SDLC	●	●	●
Empower Developers	Maybe	●	●
Just in Time Training	●	●	●
Implementation	Easy	Complex	Easy

● No ● Yes

CHAPTER 04

who can benefit from privacy scanning?

Code scanning can deliver immediate benefits to organizations with fast paced product development and modern infrastructures. Multi-cloud environments, microservices, and bottoms-up product development create the perfect setting for data sprawl and drift that can only be truly wrangled with privacy 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 scanning tools internally.

Meta has built Pysa & Zonoclan, OSS privacy scanners to detect privacy issues in large code bases. Pysa creates data flows & checks against technical privacy policies to identify privacy issues before they hit production. For example, Instagram uses privacy scanners to ensure location data is only used to calculate status but 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 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 scanning. See their story here.

privado

Privado is the privacy code scanner with open source and Enterprise ready solutions 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 Life Cycle.

Schedule a demo to learn more.

TRUSTED BY TEAMS AT

headspace ZEGO snap finance here

THRASIO trendyol Docplanner Group

The Author

Vaibhav Antil is the Founder of Privado.ai, a privacy tech startup. He previously served as Senior Product Manager at Gaana, a commercial music streaming service, and co-founded BC Jukebox, which was later acquired by Gaana. He holds a degree from Indian Institute of Technology, Bombay.

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

hello@privado.ai

twitter.com/privadohq