# Regulators and Litigators are Investigating Data Flows Through SDKs An Overview and Practical Steps to Reduce Risk

By Daniel Goldberg and Rick Borden

*A slightly modified version of this article was published on Law360.*

Data collection and transmission through tracking technologies are subjects of significant legal scrutiny, and software development kits, or SDKs, in particular, have become a focus of regulatory action and litigation. In 2023, for example, the Federal Trade Commission has brought multiple enforcement actions relating to SDK use, including against fertility app Premom and health care company GoodRx Holdings. The California Attorney General's Office has issued warning letters and met privately with companies to discuss their SDK use. And plaintiffs' attorneys have brought class actions alleging SDKs violated wiretap laws and the Video Privacy Protection Act — some of which have settled for millions of dollars.

This article explains SDK technology in plain English, outlines the legal exposure around SDKs and concludes with practical steps companies can take to reduce risk.

**What is an SDK?**

An SDK is a set of platform-specific building tools provided by a software company that includes components like debuggers, compilers and libraries to create code that runs on a specific platform, operating system or programming language. App developers, publishers and other companies use SDKs to integrate their apps with the SDK provider's services. To use an SDK, a company signs a license agreement and embeds code offered by the SDK provider in their app environment.

SDKs offer a variety of functionalities. An SDK may help a company evaluate data within their app, such as for purposes of improving user engagement or debugging or addressing errors. An SDK may allow a company to offer advanced features to users, such as the ability to log in to the app using their social media login. And an SDK may be used for monetization purposes, including content personalization and targeted advertising. Some SDK providers offer a single SDK that can be used for all these purposes.

**What are the concerns around SDKs?**

To function, SDKs require access to data within the app environment. This access creates data privacy and security concerns, such as the following.

- Lack of Transparency to Users

Users may not know the SDK exists or that a third party is receiving data about them.

- Lack of Control for Users

Users may not have the ability to control what type of data is collected by the SDK or how that data is used.

- Lack of Authorization from Users

Users may not expect a third party to receive certain types of data about them and not provide authorization for such transmission.

- Lack of Transparency to the Company

The company may not know what types of data the SDK is collecting and transmitting to the SDK provider, or how that data is used.

- Lack of Control for the Company

The company may not be able to control what type of data is collected by the SDK and transmitted to the SDK provider, or how that data is used.

- Excessive Collection or Use

The SDK may receive access to data beyond that required to provide the SDK functionality, or use data for reasons unrelated to the functionality.

- Security Risks

The SDK may pose risk to the security of the app, such as where the SDK is not properly configured, or contains vulnerabilities that may be exploited by threat actors.

SDKs also pose unique risk due to the complexity of the technology. Unlike website cookies, SDKs:

- Cannot be removed or deleted by the user;
- Are built into the code and automatically deploy; and
- Are not easily auditable.

**What are the claims around SDKs?**
As a result of these concerns, SDKs have become a major focus of regulatory action and litigation claims. Several themes emerge from these actions and claims.

- Responsibility for Third Parties

Companies are responsible for the collection and disclosure of data through SDKs within their apps. Unauthorized data flows, whether due to an inadvertent misconfiguration or conduct by an SDK provider, can lead to company liability.

- Combining High-Risk Activities

SDK risk increases where either the company or the SDK provider engages in a high-risk activity, such as collecting sensitive data — e.g., data about health, children or precise geolocation — building profiles, or engaging in targeted advertising and sales. Using an SDK for two or more high-risk activities may make the risk insurmountable.

- Sophisticated Enforcement

Regulators and litigators are using sophisticated cyber techniques, such as data packet inspection, to assess data flows between apps and SDK providers to identify compliance issues and remediation.

- Great Expectations

Regulators and litigants expect companies to rapidly answer questions about their SDKs, explain the mechanisms they use to comply with laws when utilizing SDKs, and identify the specific data that is being transmitted to SDK providers.

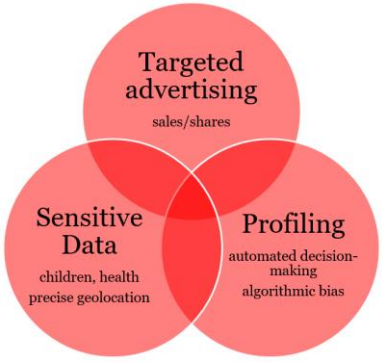


**Image of high-risk activities where overlapping circles illustrate potentially insurmountable risk*

**What steps can companies take?**
Companies should consider the following steps to reduce risk associated with SDKs.

- Establish an SDK Governance Framework

As part of their software development lifecycle, companies should establish a formal SDK governance framework, including policies, processes, procedures and practices related to mapping, measuring and managing of SDK risk. Prior to introducing any SDK into an app environment, companies should conduct a formal evaluation of the SDK. When introducing the SDK, companies should consider using an architectural framework to control and audit data flows. Companies should maintain an up-to-date list of all SDKs within the app environment, and conduct periodic reviews and training. In the event an SDK is no longer needed, companies should promptly remove it.

- Evaluate Each SDK

Companies should conduct a formal review of each SDK, and take into consideration the following.

    - *Code Functions and Configurations*

As noted above, to use an SDK, companies must embed code offered by the SDK provider. Companies should review the code and document where it lives within their app environment. In addition, companies should review all documentation from the SDK provider. Many SDKs offer code functions and configuration options that restrict data transfers or notify downstream recipients to restrict data use in accordance with specific privacy laws — such as the Children's Online Privacy Protection Act, the California Privacy Rights Act and the General Data Protection Regulation. Where possible, companies should use the appropriate functions and options. Some SDKs do not offer any options, which may create compliance hurdles for companies.

    - *Platform Configurations*

Many SDK providers offer platforms (and dashboards) where SDKs can be further configured, and at times, the use of data controlled. But these options may be poorly documented, hidden, and/or turned on by default. A misconfiguration, or conflict with the code, could lead to unexpected data flows. Companies should carefully review their SDK provider platform options.

- *Intermediary SDKs*

Some SDKs serve as an intermediary for other third parties. For example, an SDK primarily used for analytics may also integrate with third party ad networks — which may be activated through a setting in the SDK provider's dashboard — functioning as a single connection to multiple third parties. For these intermediary SDKs, companies should be extra careful to ensure the code and configurations are correct, since an error with the main SDK could flow down to all connected third parties.

- *Data Flows*

While code and configurations are supposed to affect data flows, companies should also assess the actual data being sent from their app environment to an SDK to help confirm that the data flows line up with their expectations. Such assessment may be difficult to perform since SDK providers often do not clearly disclose their data practices. Due to the complexity of the app environment and uniqueness of each SDK, companies may choose to perform a man-in-the-middle attack to determine what data is transferred between the SDK and the app environment. Companies may be surprised that code and configurations often have limited impact on the amount of data sent. Rather, these may trigger the sending of a signal notifying the SDK provider to restrict its data use.

- *Governing Contracts*

Companies should review contracts governing SDK use, to ensure such contracts align with their expectations. SDK contracts are often ambiguous or overly broad around data use, and disproportionately allocate risk to companies. Further, these contracts often include prenegotiated online terms and cross-references to numerous other documents. Contracts that do not appropriately restrict data use or provide for adequate security may affect the ability of companies to lawfully use the covered SDKs.

- *Reputation of SDK Providers and Services*

Companies should consider the reputation of the SDK provider and services they offer. For example, SDKs provided by large advertising networks are inherently risky because of the array of services they offer, the ambiguity of their contracts, and regulatory concern that data they collect could be repurposed. Further, services that require more intrusive data collection, such as session replay services, are inherently risky as they are often the basis of class actions alleging violation of wiretap laws or the Video Privacy Protection Act.

- *Privacy Manifests*

Starting in iOS 17, companies will have better control over the SDKs in their iPhone app environments. SDK providers will be able to include a privacy manifest file within their SDKs through Xcode. This privacy manifest will outline the data practices for the SDK. For example, the privacy manifest will describe the types of data collected through the SDK and the purposes of the collection. In accordance with Apple's recommendations, companies should always request SDK privacy manifests.

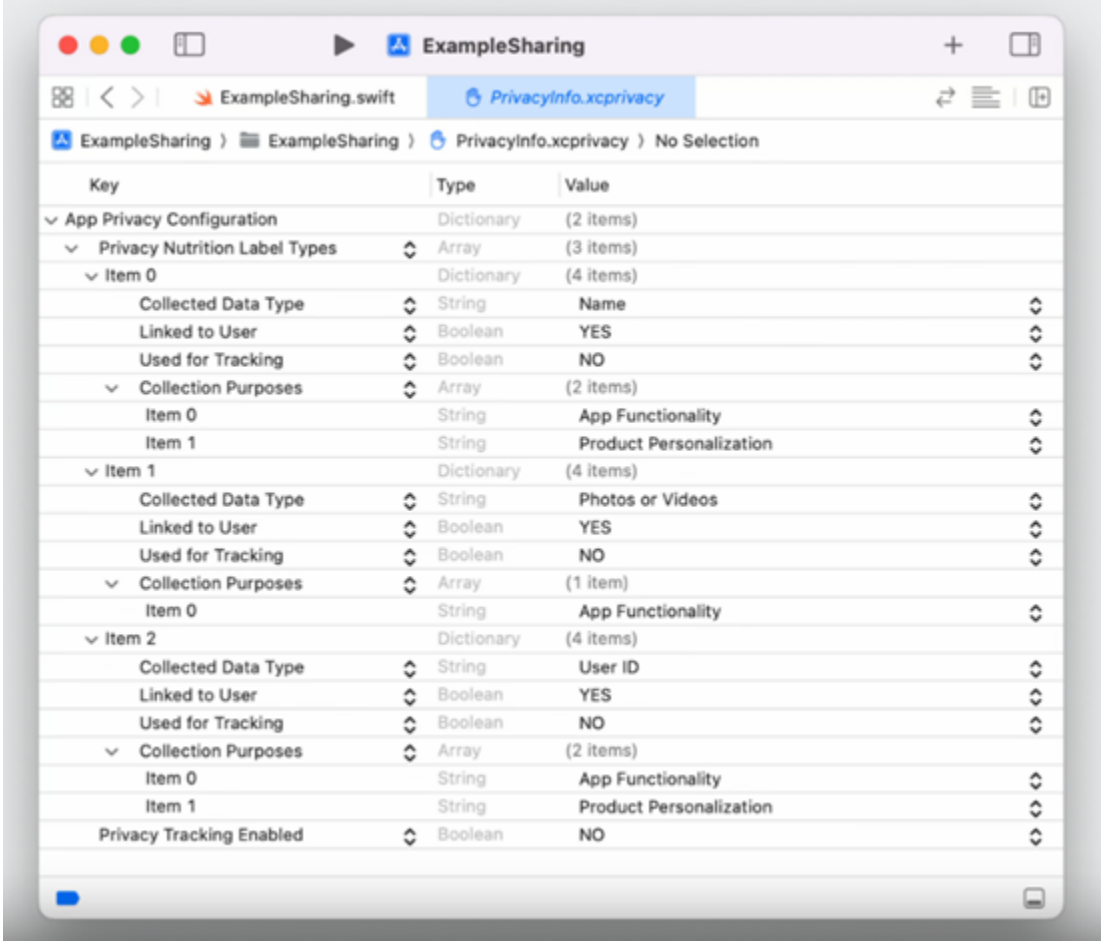


**Image of iOS 17 privacy manifest example*

- Evaluate Company Services

Companies should conduct a review of their own services, including the following.

- *Transparency*

Companies should review their own disclosures regarding SDK use. Under the law, SDK use must be clearly and conspicuously disclosed to users. Overly broad disclosures or disclosures hidden within a privacy policy may violate the law.

- *Control*

Companies should review the controls they offer to users regarding SDK use. Companies may be required by law, or choose, to offer user controls over SDK data flows. Where a company offers user controls, the company must ensure the controls work as promised. For example, if an app interface allows users to select an option to opt in to targeted ads, the SDK data flows should change based on the selection of the user.

- *Security*

Companies should make sure they have appropriate security in the app environment. Companies may be required by law to assess the risks to the security of the data that they process.

- *High-Risk Activities*

As noted above, most regulatory action and litigation claims have involved companies engaging in at least one high-risk activity. High-risk activities often require opt-in consent, and may be entirely prohibited, especially where there are multiple high-risk activities involved. Companies engaging in high-risk activities should carefully review the legality of their activities, conduct impact assessments where applicable, and restrict data flows and use of certain SDK functionalities as necessary.

- *Developers and Other Third Parties*

Companies often outsource app development to third party developers. Remember that companies are responsible for the actions of their developers. If a developer misconfigures an SDK, the company could be held liable. Companies should make sure they have an open line of communication with developers, that developers understand their obligations with respect to SDKs, and that developers document the coding, configuration, and testing of SDKs. To the extent companies work with any other third parties, they should also review their relationship with such third parties in connection with the SDKs.

- Address and Document Deficiencies

Once companies have completed the evaluation process, they must address any deficiencies identified, including misconfigurations. If anything is not working as expected, companies should contact the SDK provider and/or suspend their use of the SDK until resolved. Companies should also conduct an impact assessment, using the findings to identify and weigh the benefits of using the SDK against the potential risks along with mitigating safeguards. This entire process should be clearly documented for potential regulatory review.

**Conclusion**

The above steps should help companies reduce risks associated with SDK use, and create the processes and documentation necessary to respond to regulators or potential claimants.

{SDK risk increases where either the company or the SDK provider engages in a high-risk activity, such as collecting sensitive data — e.g., data about health, children or precise geolocation — building profiles, or engaging in targeted advertising and sales. Using an SDK for two or more high-risk activities may make the risk insurmountable.