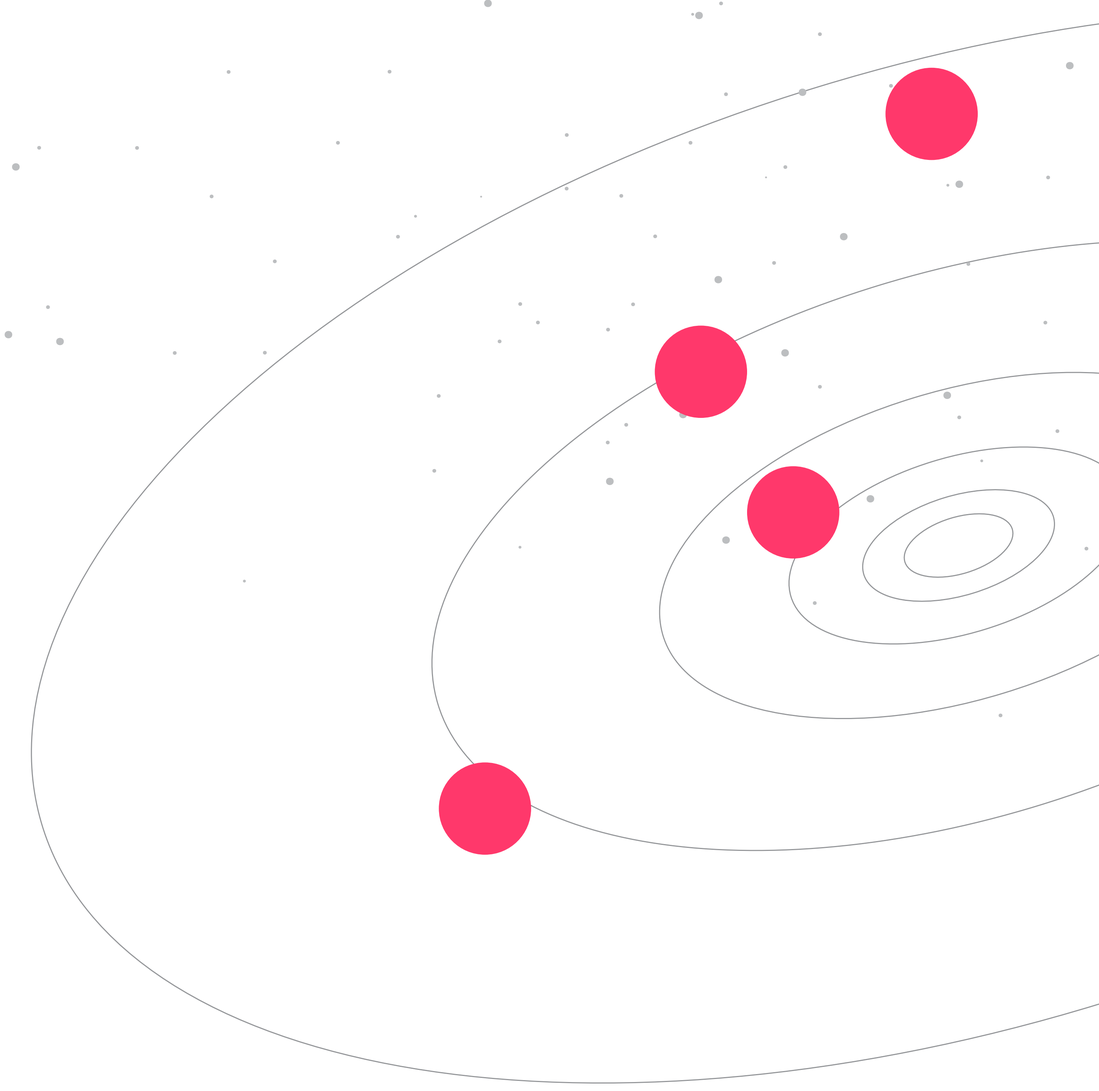
LaunchDarkly

Security overview for government.

Contents

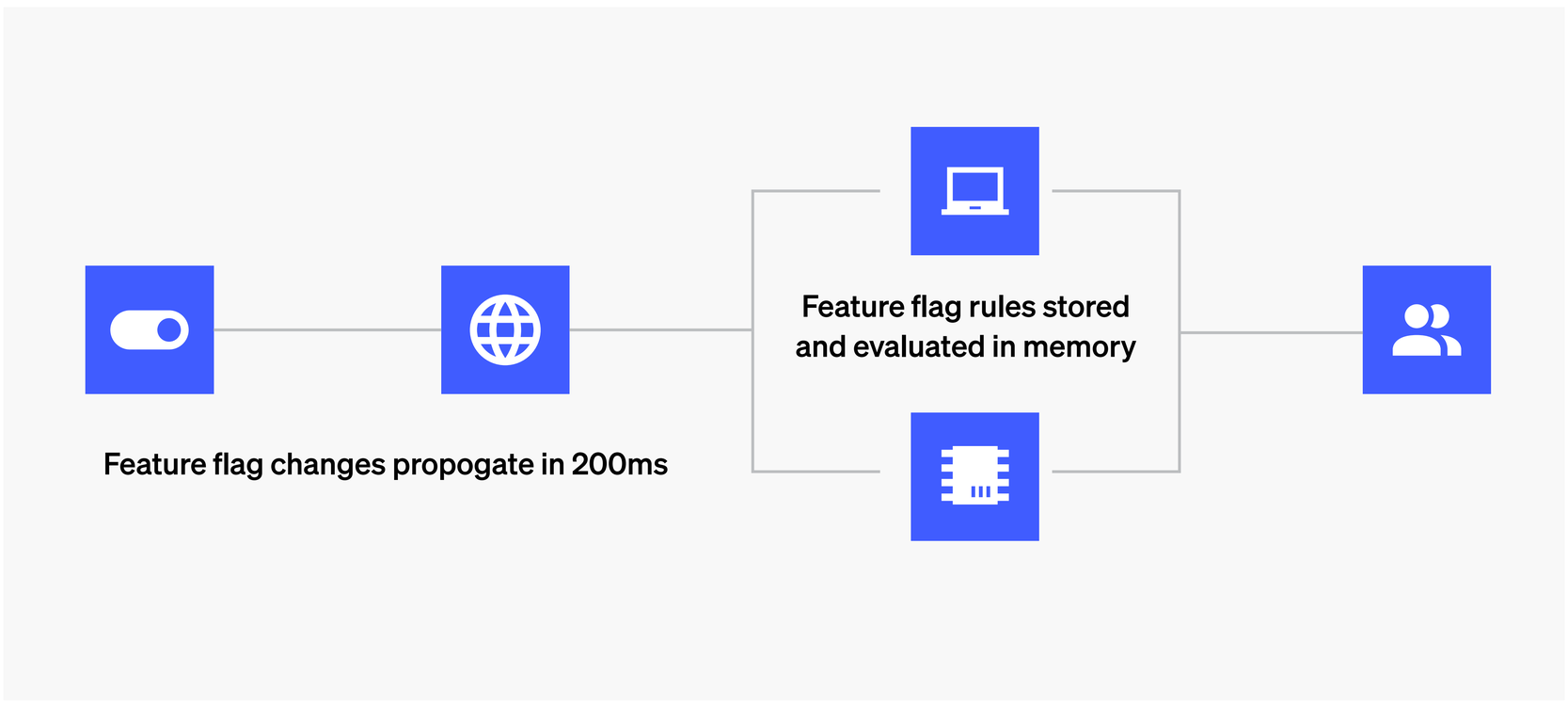


LaunchDarkly security for government agencies

LaunchDarkly created and leads the feature management space. We empower software teams to operate at an elite level in the areas of DevSecOps and CI/CD while lowering risk and costs. This paper outlines how data flows between your application and LaunchDarkly, best practices for secure operation, and how to minimize user PII from LaunchDarkly's feature management platform. LaunchDarkly fully supports U.S. government organizations charged with erring on the side of caution when protecting user data.

LaunchDarkly offers a commercial SaaS platform and a FedRAMP Moderate platform called LaunchDarkly Federal. This document describes the functionality for both environments unless otherwise specified.

Data flow



Organizations embed LaunchDarkly SDKs into their applications to enable feature evaluations at runtime. During initialization, the LaunchDarkly SDK fetches all feature flags and targeting rules from LaunchDarkly and stores them locally in memory. This means that your application can enable a feature immediately without making any external polling calls back to LaunchDarkly (except when desirable, such as with mobile applications).

Flag status or rulesets can be changed in the LaunchDarkly UI or via a REST API. When a flag is updated, the new data about that specific change is streamed to each SDK for in-memory storage using a one-way, server-sent connection. Generally, these updates propagate to all SDKs in 200ms or less, ensuring your end-users always have the correct, up-to-date experience.

In this document, we'll discuss how your application can still use PII, PHI, and other information required to enable or disable a given feature—but without needing to send that sensitive data

to LaunchDarkly. You can even restrict the ability to make flag changes based on LaunchDarkly Projects, Environments, and specific flags with granular RBAC controls.

Encryption in transit

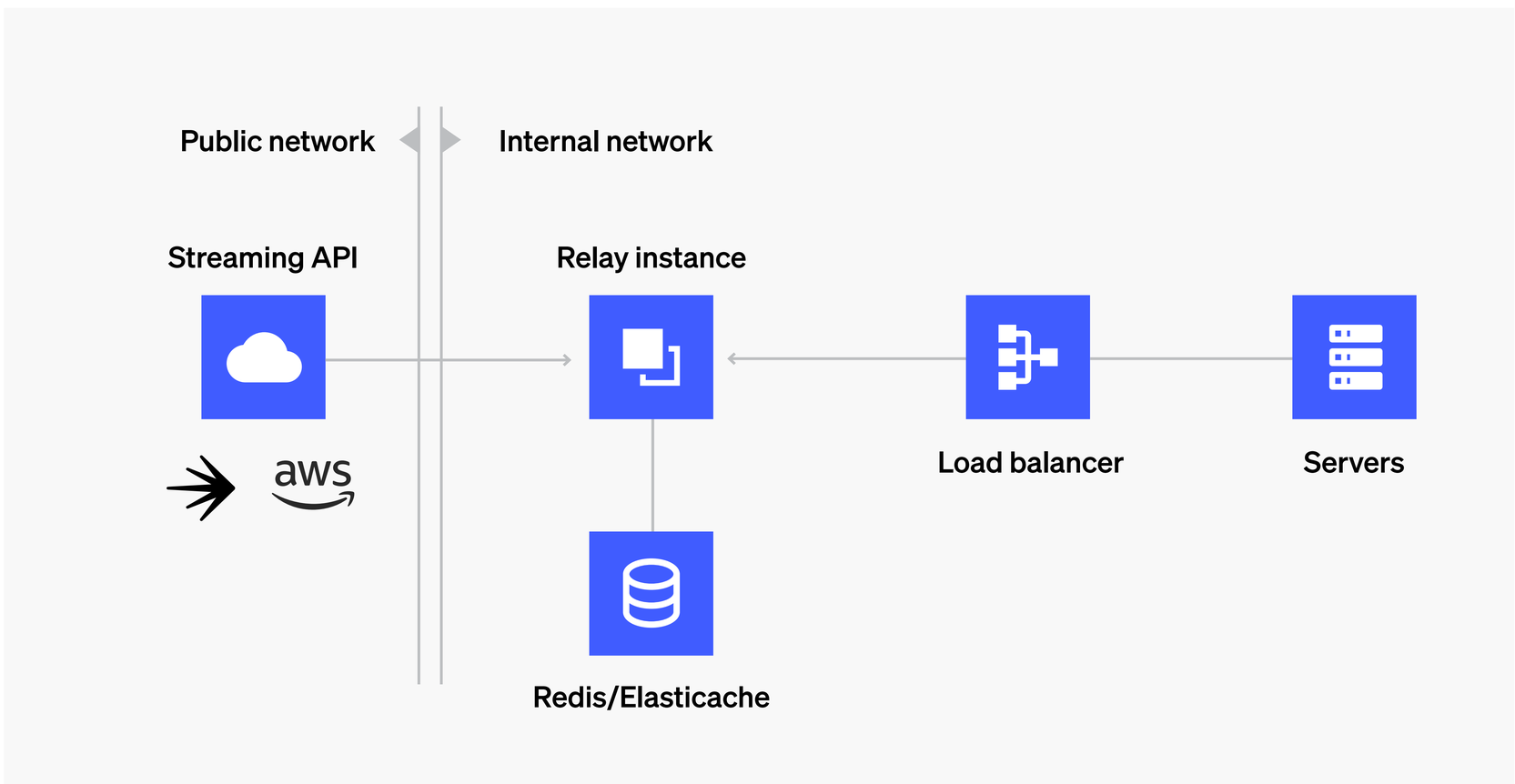
The LaunchDarkly FedRAMP platform encrypts data in transit using TLS 1.2 for all public connections.

Encryption at rest

LaunchDarkly encrypts all data, including any user data shared with LaunchDarkly, in its internal databases using AES-256 encryption.

Relay Proxy Enterprise

When organizations need to minimize the number of direct connections to LaunchDarkly, our Relay Proxy should be considered. With the Relay Proxy, servers can connect directly to hosts in your own datacenter, instead of connecting directly to LaunchDarkly's streaming API.



The LaunchDarkly Relay Proxy is an open source service written in Go, supported by LaunchDarkly, and available in a GitHub repository. It can run anywhere Go can run in binary form. The Relay Proxy is also provided as a Docker container and available in DockerHub.

The Relay Proxy is geared towards server-side SDKs but also provides mobile and client-side evaluation endpoints. This means that you can initialize a client-side SDK directly against the Relay instead of connecting it directly to LaunchDarkly. The benefit of this approach is that you reduce the number of direct connections to LaunchDarkly, thus making LaunchDarkly easier to secure and monitor. In addition, the SDKs function the same as they would in a standard environment, allowing you to still take advantage of our streaming API and real-time flag updates.

Offline mode

Enabling offline mode on the Relay Proxy Enterprise lets you run the Relay Proxy without ever connecting it directly to LaunchDarkly. Instead of retrieving flag and segment values from LaunchDarkly's servers, the Relay Proxy gets them from files located on your localhost or filesystem. This allows you to run your application in a highly secure system, such as a FedRAMP-compliant system or on a disconnected network, while taking advantage of CI/CD and LaunchDarkly's feature management solution. By using offline mode, you can secure the Relay Proxy and your application away from any external touchpoints.

Handling sensitive data

What is user data?

User data is any information about your application's end-users that could be used for feature evaluation. This is different from information about your team members, who are authorized users of your LaunchDarkly account and who are sometimes end-users for targeted feature releases.

User data can include Personally Identifiable Information (PII) and Personal Health Data (PHI) including names, email addresses, and other unique identifiers. User data can be business-critical information and can present significant risk if exposed to unauthorized parties. Government organizations should err on the side of caution when dealing with PII and PHI as there are few use cases where end-user data is needed. When end-user data is critical for feature evaluation in an application, it can still be used without storing it inside of LaunchDarkly itself.

How LaunchDarkly receives user data

Developers configure a LaunchDarkly SDK to collect and transmit attributes about end-users to LaunchDarkly for the purpose of feature targeting. When an application evaluates a feature flag, it takes into account whatever is in the user object (request context). The user object includes various key-value pairs that contain information about your users along with a user identifier key. We recommend keeping this data to a minimum and relying primarily on user roles along with anonymous users.

Considering data requirements

Every organization has different types of data collected and used for various purposes. Consider whether data is advantageous for your organization to collect and if any data exposes you to unwanted risk. For data that presents a risk, understand your requirements for handling that data. If user data is passed to LaunchDarkly, consider defaulting to a more restrictive set of data transmitted and stored to comply with relevant government restrictions. Depending on the requirements of the project and your organization, you may want to limit or completely restrict the user data you send to LaunchDarkly.

How the LaunchDarkly SDK affects user data

LaunchDarkly's SDKs have different constraints that affect user data. Specifically, client-side SDKs differ from server-side SDKs in the following ways:

- For security reasons, client-side SDKs cannot download and store an entire ruleset. Client-side SDKs typically run on customers' own devices, so they are vulnerable to having users investigate SDK content by unpacking the SDK on a mobile device or inspecting its behavior in a browser. Instead of storing potentially sensitive data, the client-side SDKs confirm and update flag rules by communicating with LaunchDarkly servers through streaming connections or with REST API requests.
- By default, client-side SDKs aren't authenticated. Because of this, one user could use another user's account to evaluate flags not meant for them. To authenticate user data, you can enable

the SDK's secure mode, which requires you to pass a server-generated hash along with your user data. To learn more, read Secure mode.

- Client-side SDKs send user data in the URL as a GET query parameter. If you are concerned about that data being stored in logs or by intermediary proxies, you can use the useReport setting to use the HTTP REPORT verb. This sends the user data in the request body rather than in the header.
- You must enable each flag that you want client-side SDKs to be able to access using the "Make this flag available to client-side SDKs" setting.

To learn more, read Client-side and server-side SDKs.

Features to minimize user data

Private User Attributes

You can use LaunchDarkly's Private Attribute settings to restrict the user data your service sends to LaunchDarkly while still using that data for flag targeting. You can make all attributes private, choose specific attributes to make private, or make attributes private for specific users.

Using the Private Attributes feature, LaunchDarkly server-side SDKs are able to ensure that no data ever leaves your server. Since all feature flag evaluation occurs locally in memory, there is no need to send user data to LaunchDarkly in order to take full advantage of our platform.

On the client side, this poses more of a challenge because, by default, feature flag evaluation occurs at endpoints hosted by LaunchDarkly.

When the LaunchDarkly Javascript SDK initializes in the default state, it sends a base64 encoded user object to LaunchDarkly to perform an evaluation.

If you set allAttributesPrivate, no user-specific data will be sent back to LaunchDarkly in the event stream, and no user-specific data will ever be stored in LaunchDarkly. However, because the evaluation occurs against the LaunchDarkly servers, the base64 encoded user object appears in any request logs since it is a part of the URL.

Note that when using the Private Attributes feature on either the server-side or client-side SDK, the user ID will always be sent to LaunchDarkly. For this reason, we recommend the user ID be a GUID, hash, or some other non-identifiable piece of data.

Greater client-side security

LaunchDarkly is designed to work in two modes: server-side and client-side. The LaunchDarkly server-side SDK initializes all of the values necessary to serve any user of your application, including all the business rules that you specify and the IDs of any users you intend to target.

Client contexts, like browsers and mobile phones, are inherently vulnerable to users tampering with client-side data. Users on these platforms can inspect your source code or de-compile

your application to reveal details about how they work. In theory, anyone can peer in and observe the rules and data that you have configured. That's why LaunchDarkly makes use of specific browser and mobile SDKs that behave differently in these client contexts.

LaunchDarkly client SDKs:

- Allow you to choose which rules should be sent to a client context. Generally, only a subset of your flags is needed.
- Ensure that they initialize only with the ruleset for a single user, so no information about other users is available.

Internal team use

LaunchDarkly has built-in authentication and authorization and supports multi-factor authentication (MFA). It can also be integrated with an IdP for Single Sign-On using SAML. It is typical for government agencies to restrict access by Project, Environment, and flags themselves using LaunchDarkly's granular RBAC settings. To learn more, read Custom roles. Government agencies that need to monitor employee access also take advantage of our built-in auditing, "last seen" team member reporting, full project history, and Slack and Splunk integrations. To simplify granting RBAC roles to individuals, LaunchDarkly supports the concept of teams. Teams are groups of your organization's members. Administrators can:

- Easily give new members a set of custom roles by adding them to an existing team.

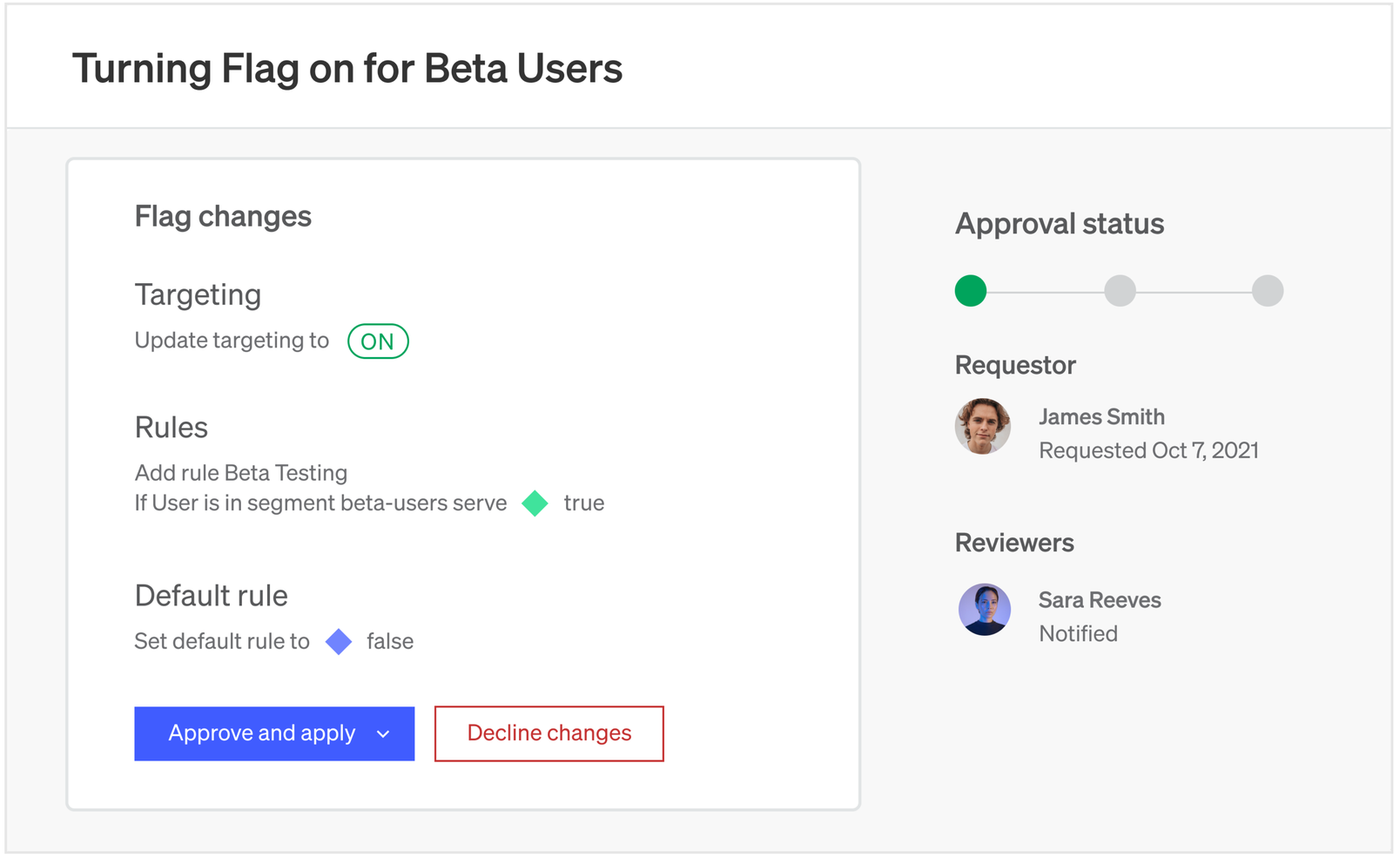
- Control environment permissions at the group level rather than individually assigning an environment's access privileges to members.
- Map permissions in LaunchDarkly to your organizational structure. For example, you can give mobile flag permissions to the mobile team and desktop flag permissions to the desktop team, or give all organization members access to the staging environment, but only people on a particular team permissions to control flags on production.

To learn more, read Teams.

Approval workflow

LaunchDarkly supports complex workflows to better map how your teams build, ship, and control software. One workflow that pertains to security is the ability for users to request or require approvals before making flag changes.

For example, you may want to get your manager to review and sign off on the changes you are about to make in production.



In addition to requesting approvals to gain confidence, admins can also require approvals for certain environments.

LaunchDarkly also supports teams that may already have change management practices in place or use third-party tools to manage changes to their production environment for compliance purposes. Our approval workflows integrate with tools such as ServiceNow and Jira to allow users to manage requests from within them.

Privacy and security standards

LaunchDarkly has undergone SOC 2 Type II and ISO 27001 certifications. We conduct bi-annual penetration testing. Reports on these are available with a signed NDA.

ISO 27001

LaunchDarkly has received certification that our Information Security Management System (ISMS) follows the ISO 27001 standard. See our blog post: Launched: ISO/IEC 27001:2013 (e) Certification. Further information can be provided with a signed NDA.

SOC 2 Type II

SOC 2 is an auditing procedure against Trust Services Criteria. LaunchDarkly has passed a SOC 2 Type II audit - see our blog post Launched: SOC 2 Type II Certified. Further information can be provided with a signed NDA.

Penetration testing

LaunchDarkly uses Cobalt.io to perform penetration tests against LaunchDarkly twice a year. The most recent penetration tests report can be provided with a signed NDA.

FedRAMP

LaunchDarkly's Federal instance is FedRAMP authorized at the moderate impact level. More information on our FedRAMP authorization can be found in the FedRAMP marketplace.

Zero trust

LaunchDarkly supports zero trust initiatives including allowing organizations to configure access with SSO using SAML 2.0 and SCIM.

HIPAA

LaunchDarkly provides comprehensive privacy and security protections that enable our customers to use our products in compliance with HIPAA.

About LaunchDarkly

Founded

2014

Headquartered

Oakland, CA

Mission

Software powers the world, we empower all teams to deliver and control their software.

About us

Our vision is to eliminate risk for developers and operations teams from the software development cycle. As the government transitions to a world built on software, there is

an increasing requirement to move quickly—but that often comes with the need to maintain control. LaunchDarkly is the feature management platform that enables development and operations teams to control the whole feature lifecycle, from concept to launch to value.

Equipping government organizations with the ability to move at the speed of deployments allows an entire set of programs to learn rapidly, deliver value faster, and produce more value. Developers can build and management can launch all with less risk, fewer failures, and faster recovery times.

Financials

Funding round

Series D

Total funding

$330 million

Our scale

Number of customers

4,000+

Flag evaluations

20 trillion+ flag evaluations per day

Supplementary reading

[LaunchDarkly in federal environments]

This documentation page explains how the instance of LaunchDarkly that is available on domains controlled by the U.S. government differs from the instance available to the general public.

[Value of LaunchDarkly architecture]

This topic covers the benefits of the LaunchDarkly architecture in the areas of speed, consistency, security, redundancy, and scale.

[Built-in attributes]

This topic explains what user attributes are, how they impact what you see in LaunchDarkly, how to configure them, and how LaunchDarkly uses them to calculate and display flag settings for users.

[LaunchDarkly documentation]

Our complete set of documentation.

[Minimizing LaunchDarkly's access to user data]

Our best practices for restricting access to user data.

APPENDIX

LaunchDarkly security policy

Introduction

LaunchDarkly's core mission is to provide a world-class service for our customers so they can realize their core mission more quickly and with less risk. Central to our core mission is a strong compliance posture, such that our customers and the data they pass into our services are protected by policies, practices, and technologies that are at least as stringent if not more stringent than their own services. To serve that end, the Office of the CTO, along with key stakeholders across the various business arms of LaunchDarkly, created the following policies with sponsorship from John Kodumal, the CTO of LaunchDarkly.

- Ongoing Compliance and Improvement Programs.
- LaunchDarkly is a SOC 2 Type II and ISO 27001 certified company and is GDPR and Privacy Shield compliant. We support HIPAA compliance and offer a FedRAMP-authorized instance of our platform.
- Human Resources Security.
- LaunchDarkly employees, as a condition of employment, undergo background checks and participate in a confidentiality agreement that includes binding them to the corporate policies and procedures of LaunchDarkly.
- Onboarding and offboarding procedures are documented, formalized, and audited on a regular basis.

Security training and orientation

All employees are enrolled in mandatory operational and information security training, driven by the Office of the CTO, within their first week of employment. Clear policies and procedures are established regarding but not limited to:

- Password and credential security.
- Data management.
- Intended usage of LaunchDarkly systems.
- Identity verification, phishing, and outside threats.
- Supplemental training is mandatory for those in special roles.

Physical security

Our core infrastructure is hosted by Amazon Web Services, spread across multiple availability zones. The physical and environmental security is provided by AWS. More details can be found here, but some of the highlights include:

AWS provides physical data center access only to approved employees. All employees who need data center access must first apply for access and provide a valid business justification. These requests are granted based on the principle of least privilege, where requests must specify to which layer of the data center the individual needs access, and are time-bound. Requests are reviewed and approved by authorized personnel, and access

is revoked after the requested time expires. Once granted admittance, individuals are restricted to areas specified in their permissions.

Physical access to AWS data centers is logged, monitored, and retained. AWS correlates information gained from logical and physical monitoring systems to enhance security on an as-needed basis.

Authorized staff must pass two-factor authentication a minimum of two times to access data center floors. All visitors and contractors are required to present identification and are signed in and continually escorted by authorized staff. AWS only provides data center access and information to employees and contractors who have a legitimate business need for such privileges. When an employee no longer has a business need for these privileges, his or her access is immediately revoked, even if they continue to be an employee of Amazon or Amazon Web Services. All physical access to data centers by AWS employees is logged and audited routinely.

With regard to LaunchDarkly offices, we are a paperless office, save for documents that are required by law to maintain in physical form. Those records are kept in locked and secured locations. All staff and non-staff personnel under LaunchDarkly employment undergo background checks as a condition of employment, and all visitors to the site are recorded and are escorted only.

Endpoint security

To protect employee workstations from unauthorized access, the following policies are in place:

- Full disk encryption.
- Workstations must be password protected when left unattended for any period of time.
- Workstations must be locked to an immovable object when left unattended.
- Password policies ensure string passwords and prevent re-use.
- Mobile Device Management systems are in place to audit and ensure continuing compliance.

The LaunchDarkly Platform

Availability

In order to ensure the availability of the LaunchDarkly service, it is hosted in multiple datacenters (availability zones, in the AWS parlance). All production services are also architected to be tolerant of one or more nodes going down without any substantial interruption in service. This is accomplished through:

- Globally-distributed CDNs.
- Redundant health-checking load balancers.
- Replicated data stores.
- Multiple application server nodes for each service.
- Other AWS regions serve as disaster recovery (DR) sites.

Transmission security

We have implemented the following industry-standard measures to prevent customer or personal data from being intercepted or altered by unauthorized parties as it is transmitted to or from the LaunchDarkly system:

- All traffic routes between customer sites and our CDNs, our CDNs and the LaunchDarkly origin, and customer sites and the LaunchDarkly Origin are encrypted using up-to-date versions of TLS, with strong encryption algorithms and keys.
- Firewalls are used to restrict access to only the required ports and protocols, abiding by the principle of least privilege.

Internal application/network security measures

We have architected our application to keep different types of data (with different access patterns) segregated, and to ensure that access to this data follows the principle of least privilege.
To this end:

- Firewalls restrict access to internal services, abiding by the principle of least privilege.
- Customer account data and business rules are encrypted at rest.
- Systems holding users' PII have strict access controls managed by IAM roles.
- Management tools are not accessible over the Internet interface, and can only be accessed by authorized LaunchDarkly employees over VPN.

- Access keys used to authenticate to services are kept out of source control at all times.
- Network routes between systems are monitored actively.

Access control

We have also taken several steps to ensure that only authorized parties can access all parts of the system. In our customer-facing application, this is accomplished by:

- TLS encryption is used from the client site (or client browser) through all routes to the LaunchDarkly system.
- User login/passwords are stored as salted hashes, never as plain text.
- We offer a powerful custom role system allowing customers to define roles restricting access to only certain parts of the LaunchDarkly application for their users.
- We can also integrate with any identity provider that speaks SAML 2.0 and SCIM.

Internally, we also take the following steps to secure access to management systems and infrastructure:

- Management UIs are all only accessible to the internal private network.
- Access to the private network is only possible via VPN, utilizing an industry-standard Public Key Infrastructure (PKI), with up-to-date encryption algorithms and cipher suites.
- All VPN and SSH access is logged.

- Access to management tools and infrastructure follows the principle of least privilege.
- Access is revoked or modified immediately upon the employee's need for access changing (including departure from the company and changes in roles).
- Password policies ensure string passwords and prevent re-use.
- MFA is used whenever available.

Third-party providers

Other providers that we rely on include Fastly and PubNub for content delivery networks. Their security policies meet the same standards outlined in the rest of this document, and we have the appropriate data processing agreements in place for each of our subprocessors. For more details:

- Fastly (CDN used for the management web app and REST API for our commercial instance only): https://docs.fastly.com/guides/compliance/security-program.
- PubNub (Used to power the dev console for our commercial instance only; data sent through here is encrypted end-to-end): https://www.pubnub.com/products/security-overview/.

A full list of subprocessors can be found on our website.

A list of all third-party services is maintained: Classifications of all data ingress and egress to third-party providers, and relevant breach/availability/solvency risk matrices are maintained by the Office of the CTO and updated on a semi-annual basis.

LaunchDarkly

Fundamentally change how you deliver software.

LaunchDarkly for U.S. government agencies