Real-World Guide to Implementing IDP-Initiated SSO in Keycloak

Vivek Koodakkara Shanmughan

Independent Researcher, USA

Abstract

IDP-initiated SSO is a key feature in Keycloak that lets organizations enable federated authentication across a portfolio of applications. This document describes the architecture, configuration, and operation of IDP-initiated SAML flows where Keycloak is a service provider that receives unsolicited authentication assertions from an external identity provider. When interoperating with enterprise identity providers such as Microsoft Entra ID and Okta, or legacy SAML identity providers, attention must be paid to metadata compatibility, endpoint configurations, signature validation, and attribute profiles. In production deployments, enterprises must implement security features such as certificate rotation, clock skew, assertion lifetime validation, and single logout to maintain a secure authentication posture. Support for RelayState URL parameters improves post-authentication routing between multiple applications. Wide-ranging logging in Keycloak may assist during the integration process. When configured correctly, IDP-initiated SSO with Keycloak enables a better authentication experience, reducing login prompts and minimizing user cognitive load in traversing a collection of applications from centralized dashboards and portals. This depends on the knowledge of the SAML specification as well as the details of SAML implementation variations, performance tuning and optimization best practices, operational monitoring, and statistics tools. Organizations that have properly configured federated identity systems find they achieve measurable help desk, policy, and productivity benefits through reduced help desk calls, centralized policy enforcement, and consistent access patterns across application ecosystems.

Keywords: Identity Provider-Initiated SSO, Keycloak Configuration, SAML Federation, Enterprise Authentication, Service Provider Implementation

1. Introduction

One of the features of the Keycloak project that is poorly documented or not well understood is Identity Provider (IDP)-Initiated Single Sign-On (SSO). The market of Identity and Access Management is rapidly changing, and it is gaining popularity that federated identity management systems are the infrastructural backbone of a lot of organizations worldwide. According to recent market research, IAM market is expanding because companies are implementing zero-trust architecture and providing a consistent user experience to multi-cloud and on-premises applications they use [1]. A large number of enterprises employ third-party identity providers, like Microsoft Entra ID, to gain access to cloud-based and on-premises applications, conditionally accessing, multi-factor authenticating, and federating identity services [2].

In this situation, however, Keycloak is merely a Service Provider; it receives a SAML response inbound, and a session requires authentication with Keycloak, not presenting a login page. Although Keycloak can be used in this use case, it would demand one to have a good understanding of metadata in the use, that is, actual implementations and Keycloak release differences. The key drawbacks of federation based on SAML can be found in the fact that there are many complicated aspects of interoperability that need to be handled on a per-implementation basis. These are certificate trust chain, endpoint specifications, and attribute mapping schema. Some of the potential areas of interoperability challenge in organizations that apply federated SSO may include metadata and/or protocol requirement synchronization, test coverage in relation to user situations, and network topologies. This paper details such scenarios and other scenarios in addition to operational experiences acquired during production deployments, so system administrators can discover their way to the challenge of implementing and deploying IDP-initiated SSO without losing any of its security or usability.

2. Understanding the IDP-Initiated Flow Architecture

All the flows triggered by IDP use the same initial step, and the SAML Response message is delivered by the external IDP to the SAML endpoint of the Keycloak instance. According to the SAML Technical Overview, this completely contrasts with service provider-initiated flows, in which the identity provider generates and transmits the assertion without receiving an authentication request initially from the service provider. The message format and the semantics of the processing are specified by the SAML to allow the cross-domain single sign-on, and the flow based on the IDP is not the only profile provided by the SAML specification [3]. Keycloak does not send any authentication request as SP-initiated flows do. The authentication process starts at the external identity provider, where the user is authenticate,d and a SAML assertion is generated with the identity and attributes.

Microsoft Entra ID is not an exception to this trend, as Microsoft Entra ID administrators can enable multiple SSO applications by defining SAML-based applications in its enterprise applications. Customers of Microsoft Entra have a choice to apply either built-in support of IDP-initiated or SP-initiated flows, since the Keycloak Identity Provider may be configured to favor either of the patterns. Where Keycloak has been set up to allow IDP-based SSO, Entra ID can produce a properly-formed SAML assertion with user attributes and group memberships, as well as information about the authentication context that can be consumed by receiving service providers to create authenticated sessions [2]. This claim is subsequently forwarded to the SAML consumption endpoint of Keycloak, which must be configured to receive and authenticate unsolicited authentication claims.

The SAML specification defines in detail the syntax and security constraints required or applicable to SAML assertions. Assertions typically include the identity of the subject, conditions that govern the validity of the assertion, and statements such as authentication decisions and attribute values. For integrity and authenticity, assertions are digitally signed. Digital signatures must cover only certain parts of an assertion to prevent selective signing attacks. SAML implementations must conform to additional restrictions on the XML schema definition in order to be interoperable, such as namespace declarations, element order, or encoding of attribute values [3]. Keycloak must at least be configured to trust the assertion, map the attributes from the assertion to the attributes in Keycloak, and route the user to the application's context path as indicated by RelayState or any routing information.

From the point of deployment and security, the difference is that the service provider must accept an unsolicited SAML response, validate it against the deployed and trusted IDP metadata, and establish the local session using only the attributes in the assertion. The authentication context must be evaluated by the relying party to ensure that the level of assurance of the external IDP is in line with its requirements to access its protected resource. Processing must also take into account particular edge cases, such as an assertion that expires in transit, missing attributes, or signature validation failures. Microsoft Entra ID provides wide-ranging logging capabilities and error reporting to assist administrators in the debugging of integration issues and the provision of diagnostic information when the processing of assertions fails or attribute mapping returns unexpected results [2].

Component	Description	Validation Requirements	Common Issues
Subject Identification	Contains NameID format and value identifying the authenticated user	Must match the expected format configured in Keycloak	Format mismatch between IDP and SP
Conditions Element	Defines temporal validity and audience restrictions	NotBefore/NotOnOrAfter must be within clock skew tolerance	Clock synchronization failures
Authentication Statement	Describes the authentication method and context	Must meet minimum authentication level requirements	Insufficient authentication context
Attribute Statement	Carries user attributes for provisioning	Must map to Keycloak user model attributes	Missing or incorrectly named attributes
Digital Signature	Ensures integrity and authenticity	Certificate must match trusted IDP metadata	Certificate expiration or mismatch

Table 1: SAML Assertion Components and Processing Requirements [3, 4]

3. Configuring Keycloak as a Service Provider

In order to make Keycloak accept IDP-initiated SSO, follow the steps presented in the chapter on how to configure identity brokering in the Keycloak Server Administration Guide in the appropriate sequence and with the appropriate values. The initial one involves ensuring that the administrator establishes a domain that outlines the service provider entity configuration. Second, the administrator will then need to define an identity provider that will define how

Keycloak will communicate with the external SAML identity provider. This is because building this identity provider necessarily involves an elementary understanding of how Keycloak works in identity brokering. Identity brokering feature is a Keycloak service that is interoperable with external identity providers and applications of the client. This technology can be used to translate the authentication protocols and user sessions across the federation boundary [5]. The first step involves configuring an Identity Provider record in Keycloak under a domain that needs to be SAML Authenticated. Select Identity Providers, then choose from SAML v2.0.

The second thing to do is to import the metadata of the IDP at this point. These are the signing certificates, entity ID, as well as the service endpoints of the IDP. Metadata, as defined in the SAML standard by the OASIS consortium, refers to the technical agreement between the parties in the federation describing capabilities, endpoints, and security requirements of the parties in the federation. Metadata XML document includes the IDP entity identifier (a globally unique identifier of the IDP), single sign-on and single logout endpoints locations, X. 509 signature validation certificates, and parameters related to protocol specifics of the federation relationship [6]. This metadata is frequently simply loaded either off an XML file released by the external IDP or linked to a famous metadata URL released by the IDP itself. Having received this, Keycloak has an established trust relationship and is now capable of validating received SAML assertions.

For provisioning and session creation, the NameID formats and attribute mapping must match the IDP. According to Keycloak documentation, attribute mappers transform data from the external authentication assertion into Keycloak's internal user model. It also states that the admin can map incoming SAML attributes to user attributes, roles, or group memberships. This mapping configuration can be used to define transformation rules involving hard-coded values, request attribute values, role mapping, or JavaScript custom mappers for more complicated use cases [5]. The NameID format will be the same as that used by the external IDP, and can include email addresses, persistent identifiers, or transient identifiers, depending on the privacy level and technical integration needs.

Applications registered with Microsoft Entra ID expect certain user attributes to be sent in a particular format, and the application administrators will have to ensure that the mapping is done correctly. When registering enterprise applications in the Azure portal, Microsoft Entra ID administrators can configure which user attributes are sent in SAML assertions. The attributes supported largely correspond to the Microsoft Entra ID common attributes, as well as other directory extension attributes. The attribute statement contains multiple attributes, as per the SAML specification. Each attribute has a name and one or more values that the service provider will accept when creating a session [2]. Other integration errors are more common, such as audience restrictions mismatch, Assertion Consumer Service URLs mismatch, or signature algorithms not matching. According to the OASIS SAML specification, audience restrictions can be used to ensure that a directed recipient of an assertion cannot use it with another service provider. An IDP and a service provider must have the same value for their respective ACS URLs. Any mismatch, including slight variations in URLs or the protocol scheme, will cause the IDP to reject the assertion in [6]. But if paid attention to these configuration details while launching the application, this saves a lot of troubleshooting down the line.

Configuration Parameter	Purpose	Typical Values	Impact if Misconfigured
Entity ID	Unique identifier for the external IDP	URI format matching IDP metadata	Assertion validation failure
Single Sign-On Service URL	Endpoint where authentication requests are sent	HTTPS URL from IDP metadata	Connection failures in SP-initiated flows
Single Logout Service URL	Endpoint for coordinated logout requests	HTTPS URL from IDP metadata	Sessions persist after logout
Signing Certificate	Public key for validating IDP signatures	X. 509 certificate from metadata	All authentications rejected
NameID Policy Format	Expected format of subject identifier	Email, persistent, transient, or unspecified	User provisioning failures

Want AuthnRequests Signed	Whether IDP expects signed requests	Boolean value based on IDP requirements	SP-initiated flows rejected by IDP
Validate Signatures	Whether Keycloak validates incoming signatures	Enabled for production security	Security vulnerability if disabled
Attribute Mappers	Rules for transforming IDP attributes	Custom mapping configuration	Incomplete user profiles

Table 2: Keycloak Identity Provider Configuration Parameters [5, 6]

4. Endpoint Configuration and RelayState Management

Another practical problem is determining the correct IDP-Initiated SSO URL for use to access a target domain and client. Setting the correct endpoints requires knowledge about Keycloak's URL patterns, how requests are routed, and how the Keycloak server is architected. As noted in the Keycloak scaling and tuning documentation, this can have a performance impact since different processing paths are used, based on endpoint type. The same documentation notes that a Keycloak administrator needs to understand how Keycloak determines how to trust a request on incoming URLs, and how protocol adapters treat authentications [7]. While administrators expect a stable endpoint URL, the location of the endpoint varies based on the protocol being used by the client and the type of redirect and binding. In Keycloak, the SAML POST binding endpoint is typically based on the domain name and the type of protocol adapter being utilized by the client application.

The bindings in the SAML Protocols lay out the transport mechanisms that carry SAML protocol messages between two communicating SAML entities as described in the SAML Executive Overview published by the OASIS technical committees. The HTTP POST binding is typically employed in cross-domain applications, when using a web browser to transport. The binding represents SAML messages as base64-encoded values in HTML form fields. This enables web browsers to carry SAML authentication assertions between a SAML identity provider and a SAML service provider across the HTTP without having direct connections between the SAML entities. This architecture facilitates the federation of identities using authentication pathways across across multiple organizations and securing the pathways using cryptographic signatures and encryption [8]. In the case of IDP-initiated flows, the endpoint should be set to tolerate unsolicited SAML responses that do not have an authentication request before them issued by Keycloak.

According to the Keycloak administration documentation, service provider endpoints have a domain-level hierarchy in addition to protocol adapters that handle SAML, OpenID Connect, and other protocols. When administrators configure external IDPs to initiate SSO flows, they must specify the exact endpoint URLs to which SAML assertions should be posted to ensure the IDP configuration and the endpoint URLs used by the service provider are compatible. The endpoint receiving the assertion extracts it from the HTTP POST body, validates the signature with the trusted certificates, checks the conditions, checks the audience, and then establishes the session if all validation steps are successful [5]. The endpoint receiving the SAML response via HTTP POST processes it according to the domain configuration and the identity provider configuration that determines how assertions from that identity provider are processed.

In generic circumstances, external IDPs may be configured with a RelayState parameter, or target application value, so that a user will be redirected to the IDP's service with context after a successful authentication. RFC 7522 specifies bearer token handling in OAuth contexts, where other state-maintaining mechanisms are used, for example, through a multi-step authentication process. The SAML specification provides RelayState, which allows the identity provider to send opaque state information to the service provider to be used when redirecting the user after authentication [9]. Keycloak uses this RelayState for determining where Keycloak redirects the user after Keycloak processes the assertion and returns control to the service provider. This is particularly useful when multiple applications share the same Keycloak domain, and the user must redirect to a particular application based on the flow that initiated.

Another example shows how RelayState behaves inside an enterprise environment according to Axway Keycloak configuration documentation. The handling of RelayState depends on the IDP configuration and the client application. When a user triggers authentication from the portal or application catalog, the source application usually includes contextual information about where the user will be redirected upon successful authentication. This contextual information must be carried through the authentication flow and interpreted by Keycloak's session management layer so that a user is redirected to the correct destination rather than a generic domain landing page [10]. The RelayState value contains the client ID or the URL of the application the user is trying to access. Keycloak can redirect the user to the application after establishing a new session. Managing RelayState properly is critical to ensuring a good user experience and preventing confusion and additional steps, without which single sign-on in the enterprise is often overlooked.

Endpoint Type	URL Pattern	Protocol Binding	Primary Use Case
SAML Broker Endpoint	/realms/{realm}/broker/{idp-alias}/endpoint	HTTP-POST or HTTP-Redirect	General SAML broker operations
SAML Protocol Endpoint	/realms/{realm}/protocol/saml	HTTP-POST	IDP-initiated SSO flows
Assertion Consumer Service	/realms/{realm}/broker/{idp-alias}/endpoint	HTTP-POST	Receiving SAML responses
Single Logout Service	/realms/{realm}/broker/{idp-alias}/endpoint/logout	HTTP-POST or HTTP-Redirect	Coordinated logout requests
Metadata Endpoint	/realms/{realm}/broker/{idp-alias}/endpoint/descriptor	HTTP-GET	Publishing SP metadata to IDPs

Table 3: Endpoint Types and URL Structures in Keycloak [7, 8]

5. Production-Grade Security and Troubleshooting Considerations

For production systems, there are additional considerations, such as security and runtime maintenance, that depend on the connectivity capabilities. Problems that are common in enterprise systems include certificate management, time management, and protocol compliance. Signature validation for SAML assertion events is an important security consideration for Keycloak. The Keycloak Server Administration Guide states that signature validation settings should be aligned with the signing policy of external IDPs, and that Keycloak's options for enforcing response, assertion, or both signings may be configured according to security requirements and the capabilities of external IDPs. Administrators may also configure Keycloak to validate signatures using certificates included in IDP metadata, and Keycloak uses cryptographic algorithms to verify the signature before accepting the assertion [5].

The requirements for signing can depend on the capabilities of the individual IDPs. According to the SAML specification maintained by OASIS, signing may be required at the Response level, the Assertion level, or both. Each method has advantages and drawbacks. The second method can be used as its own extreme case for added security. The signature scope controls which parts of the message are cryptographically signed, so that changes to specific parts can be detected. Different IDPs will have different implementations of these aspects of the message signing depending on their security considerations and technical design [6]. For example, a combination of wanting AuthnRequests signed and validating signatures can lead to unexpected rejection if the upstream IDP signs only the assertion element of the response envelope, or vice versa, depending on defaults.

Another practical point is clock skew tolerance, which also partly determines the successful authentications. This topic is included in the Keycloak scaling and tuning guide, which mentions reliable clock synchronization in the context of preferred practices for distributed systems. Assertion validity windows depend upon synchronized time on both the IDP and service provider sides. According to the documentation, a reasonable clock skew tolerance should be configured to account for slight time differences. In production deployment environments, it is common to provide reasonable tolerances for the security requirements and the operational reliability [7]. If the clock of the IDP and the Keycloak server drift apart beyond allowed tolerances, then valid assertions will expire prematurely, or will not yet be valid if the IDP clock is ahead of the service provider clock. The SAML specification also provides for the NotBefore and NotOnOrAfter conditions that specify the time range in which an assertion is valid, using the service provider's clock when the assertion is processed [6].

Correctly setting the maximum assertion lifetime allows security to be optimized while minimizing the potential for failure of authentication. The OASIS SAML Executive Overview states that shorter lifetimes reduce the time the attacker has to replay the assertion, but that too-short lifetimes can cause failures to authenticate over high-latency links or when the systems involved are under load beyond a certain threshold. In practice, the specification recommends keeping assertion lifetime values to a reasonably low value, but long enough to account for regular processing and network delays [8]. Reasonable values are typically determined by production systems by considering network round-trip latency, load balancer processing time, and peak volumes of traffic.

The logout procedure in IDP initiated flows should be given more consideration since the condition of the session state requires to be correctly swept across all the parties. AXway Keycloak documentation gives details on SLO configurations. SLO also enables the use of logout actions on one of the participants of that session in a federation to occur on the rest of the participants of the session. If working correctly, logging out from the external IDP will result in logging out of all connected SPs where the user has an active session, and vice versa for a Keycloak logout, where a request is sent to the originating IDP to correctly end the upstream authentication session [10]. This prevents orphaned sessions after logout, improving the user's experience and security.

Enterprise customers' onboarding and integration debugging requires logging and debug traces at the SAML level. The Keycloak Server Administration Guide describes that a Keycloak admin can enable protocol-level logging to get full SAML messages, including the XML assertions, signature validation process, and the results of attribute processing. This diagnostic information is also useful when integrating with external identity providers that may not strictly adhere to the HTTP spec and can have quirks and irrationalities [5]. Enabling debugging logs can help with issues such as getting a view of the contents of the SAML assertion message, attribute mapping issues when attributes are not sent or are incorrectly formatted, or signature validation issues, such as certificate mismatch or unsupported signature algorithms. The RFC 7522 specification of SAML bearer assertion can be useful for debugging OAuth and SAML integration issues that use SAML assertions as OAuth authentication grants, since it defines the format and processing requirements applicable to bearer assertions used in OAuth [9]. This makes it much easier to solve authentication problems.

Security Aspect	Recommended Configuration	Monitoring Approach	Troubleshooting Steps
Signature Validation	Enable validation, configure trusted certificates	Monitor authentication failure rates	Enable SAML debug logging, verify certificate chain
Clock Skew Tolerance	Configure tolerance between one and three minutes	Track timestamp-related rejections	Verify NTP synchronization across systems
Assertion Lifetime	Set the validity window between one and five minutes	Monitor expired assertion errors	Adjust the lifetime based on network latency
Attribute Mapping	Map all required attributes with fallback values	Check for incomplete user profiles	Inspect actual assertion content in logs
Single Logout	Configure SLO endpoints on both IDP and SP	Track orphaned session occurrences	Verify the logout URL configuration and binding
Certificate Management	Implement rotation before expiration	Set alerts for approaching expiration	Maintain certificate inventory with expiration dates

Table 4: Security Configuration Best Practices and Troubleshooting [9, 10]

Conclusion

Collectively, these advantages send a strong message regarding the importance of IDP-initiated SSO for Keycloak users to meet the requirements of organizations creating secure, yet easy-to-use, authentication experiences for users across dozens of decentralized enterprise applications. The IDP-initiated flow avoids unnecessary screens and provides smooth SSO experiences to meet expectations for frictionless access to enterprise services, especially when users access applications from a corporate portal or centralized application catalog rather than navigating directly to each application. To enable a smooth user experience, the identity federation needs to take into account fine-grained details in the configuration process, such as ensuring synchronized metadata between federated identity providers and Service Providers, specifying endpoints to accommodate protocol-specific characteristics, and attribute mapping between the identity provider and the service provider. Organizations need to be familiar with the technical details specified in the SAML specifications, as well as the practical implementation variations and non-standard behaviors of various identity provider implementations. Signature validation, certificate management, clock skew, and assertion lifetime are security-related parameters of the protocol. They must be set carefully, such that they do not introduce security issues and do not prevent continued operation during normal usage. When deployed in production, logging and event capture enable synchronization and integration issues to be diagnosed and fixed quickly, especially early in the process of onboarding new external identity providers, and diagnosing sporadic authentication failures caused by network conditions, clock drift, or expired certificates. Organizations that successfully implement IDP-initiated SSO with Keycloak have established the foundation of enterprise identity federation. Depending on the implementation, this may be either for partner access programs, customer-facing applications with federated credentials, multi-tenant applications with different user groups authenticating from different federated identity service providers, or combinations of the three. The ROI can be measured in various ways, including but not limited to cost reductions with password-centric help desk requests and associated end-user impact, improved security through consistent identity policy enforcement and authentication standards, and greater organizational agility in onboarding new applications or building federated identity relationships with partners. In production, continuing operational attention should be paid to performance monitoring, capacity planning as users increase, proactive certificate renewal, and validating that authentication flows still exhibit expected reliability and security characteristics while both Keycloak and the external identity provider platforms evolve via version upgrades and configuration changes over time.

References

[1] Dan Verton, "Identity and Access Management (IAM) Market Guide 2025," Information Security Media Group, 2025. [Online]. Available: https://ismg.io/resource/identity-and-access-management-iam-market-guide-2025/

[2] Microsoft Corporation, "Microsoft Entra ID documentation". [Online]. Available: https://learn.microsoft.com/en-us/entra/identity/

[3] Nick Ragouzis et al., "Security Assertion Markup Language (SAML) V2.0 Technical Overview". [Online]. Available: https://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

[4] Vittorio Bertocci, "OAuth2.0 and OpenID Connect: The Professional Guide," Auth0. [Online]. Available: https://auth0.com/resources/ebooks/oauth-openid-connect-professional-guide

[5] Keycloak, "Server Administration Guide," 2025. [Online]. Available: https://www.keycloak.org/docs/latest/server_admin/index.html

[6] Cloudflare, "What is SAML? | How SAML authentication works". [Online]. Available: https://www.cloudflare.com/learning/access-management/what-is-saml/

[7] Keycloak Project, "Scaling". [Online]. Available: https://www.keycloak.org/getting-started/getting-started-scaling-and-tuning

[8] Thomas Wisniewski et al., "SAML V2.0 Executive Overview," OASIS Committee Draft, 2005. [Online]. Available: https://www.eecs.yorku.ca/course_archive/2007-08/W/4213/Projects/sstc-saml-exec-overview-2.0-cd-01-2col.pdf

[9] Brian Campbell et al., "Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants RFC 7522," datatracker, 2015. [Online]. Available: https://datatracker.ietf.org/doc/rfc7522/

[10] Axway Software, "Install and configure Keycloak server". [Online]. Available: https://docs.axway.com/bundle/EBICSClient_11_allOS_en_HTML5/page/install_and_configure_keycloak_server.html