The kingdom of Keycloak

21 November 2022

Suche

Search

Keycloak Blog

Intelligent Keycloak realm planning: Dos and Don’ts

If you want to implement access on the net via Keycloak, there’s no getting around realms. This is because Keycloak architects use this terminology to define instances and plan access rights. Like kingdoms, it’s a good idea to avoid having too many realms – in the real world and in IT.

The use of too many realms results in too much complexity

In Keycloak, each realm is like its own client, which is why we also speak of multi-tenancy or multi-client capability. Data and configurations are stored in it, and they are not visible to other realms.

A realm consists of:

- users,
- user groups, and
- the assigned applications to which access is granted (single sign-on).

Roles supplement the model: they can be assigned either to a single application or an entire realm.

Here is an example that isn't too abstract: The widely used Atlassian software, which provides tools for developers, makes use of different roles for its users. At first, only a default role is of interest, which allows a user to access his or her account after registration. If this login is to run via Keycloak, this simple role must be defined in the Keycloak realm.

An application role is a set of permissions for the associated application. In a group, individual roles can be summed up and, in this case, a group represents a set of users who have these roles.

- One or several Identity Providers (IDPs) can also be included, who can be directly integrated in one or more realms.
- Applications can usually only be integrated in one realm. To integrate an application in several realms, it is often necessary to use an additional instance of that application.

The master realm forms the basis

The addition of several realms should be carefully considered and planned. First, you work with the “master realm”, from which you set and control the other realms. It makes sense to give the realm a meaningful name, such as “Customers”. Numbering is also possible: “Realm 1”, “Realm 2”, etc.

Find out how to configure realms here: Configure realms in Keycloak

For better understanding, here is a diagram to demonstrate:

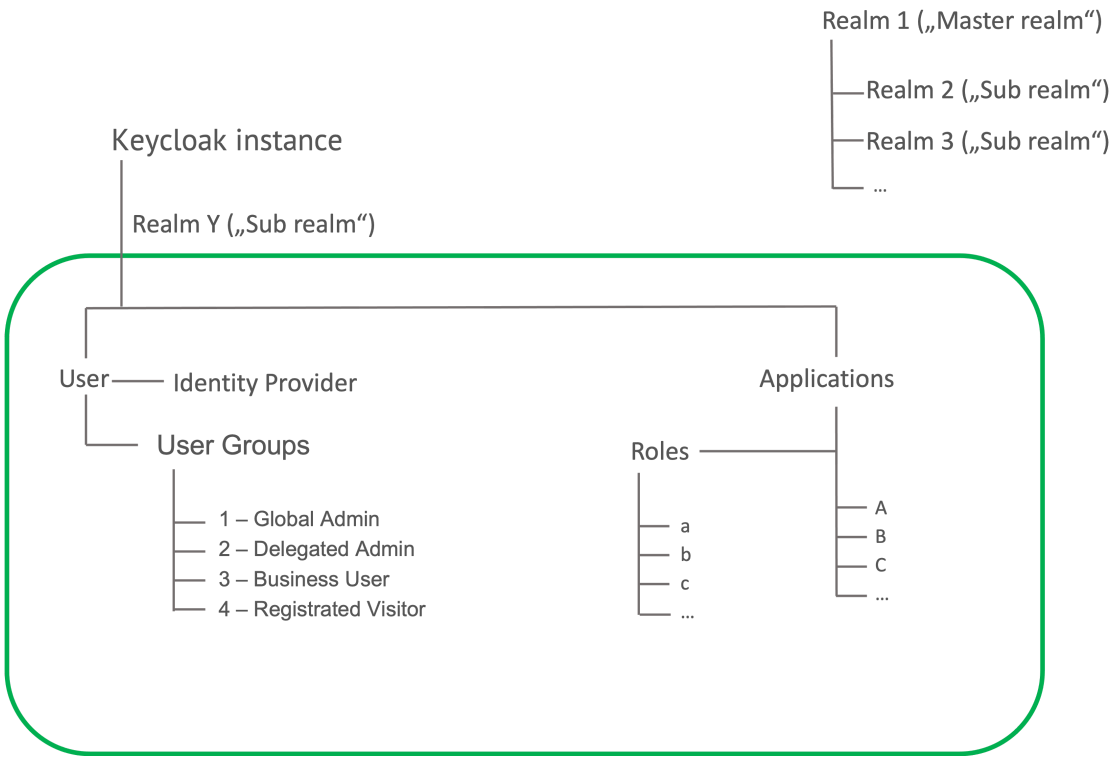

Diagram: overview of a realm

User groupings are an important building block for realms in Keycloak

The need for different user groups is often a reason for using multiple realms in Keycloak. The following reasons for realm divisions are common:

- To separate according to compliance criteria, for example a country-

specific realm division (USA, EU, Asia-Pacific, etc.).

- To separate according to different customers, users or user groups (with or without own end users); for example to isolate data and IDP configurations from each other.
- To separate by different user groups in the sense of groups with different access rights, e.g. role combinations (role-based access control = RBAC).
- To separate according to a set of different applications.

This sounds logical at first, but it creates a certain level of complexity. It becomes inefficient if the separation, especially of customer groups, is too fine-grain.

Why are multiple realms not recommended?

It is not at all advisable to create one realm per client, for example. There would be too many to operate such a system efficiently. Moreover, with the current Keycloak versions, there is a drop in performance when there are more than 300 Keycloak realms in an instance. The new storage model is the only way to eliminate such obstacles, but it will not reach production maturity until the end of 2023 at the earliest. Until then, the

current workaround involves the operation of multiple Keycloak instances.

However, this also means that each instance will have its own security vulnerabilities, and it must be configured and maintained separately. Security patches have to be applied individually for each instance. Memory problems can occur. Due to the growing complexity, the overview can be quickly lost and errors can creep in during operation. Each realm is a separate, self-contained unit, even if you try to counteract that via automation.

The user landscape determines the complexity of the Keycloak realms

Complexity is already there from the beginning. It is determined by the organisational structure to be mapped or the requirements of user groupings. However, there is a smart solution for this mammoth task: the authorisation framework SecuRole® avoids unnecessary realm proliferation.

The required authorisation management function comes from the internal IAM and ensures clarity in the delegation and allocation of roles for specific users. For web applications in the external IAM, it can easily

be implemented with the Keycloak extension mentioned above. In doing so, one can even increase security and compliance in the interaction with identity providers. How?

How does end-to-end security work with Keycloak?

An example with Active Directory (AD) or Azure shows how the Keycloak extension SecuRole® delivers end-to-end security:

If users come from internal sources – for example, sales staff who look after customers – they are transferred from AD groups into existing Keycloak groups. But since AD functions in an infrastructure-related capacity and has nothing to do with IAM processes in this sense, this happens without a digital signature. By incorporating another security layer with the authorisation framework SecuRole® , this can be added without having to abandon AD or Keycloak, or without abandoning the whole concept.

Neither Keycloak nor AD or Azure can guarantee end-to-end security

But, today, end-to-end security is indispensable, especially when IT security requirements increase and you want to switch from a legacy infrastructure to the cloud. If you want to understand this point even better, we recommend the following blog article or the Kuppinger&Cole Analyst Chat on the topic of Active Directories:

Analyst Chat KC #77: Don't Manage Access in Active Directory Groups

Blog article: Active Directory als der Teil der IAM-Stragie – wichtig, aber nicht ausreichend (in German)

This article covered: Planning Keycloak realms right

IAM Blog
Keycloak Blog
Technical information
Connect webportals