CERN SSO
service evolution in the last two years

Sebastian Łopieński, Antonio Nappi, Hannah Short, Paul Van Uytvinck
(CERN IT-PW)

ITTF - 31 January 2025

CERN Single Sign-On

Sign in with a CERN account

Username

Password

Sign In

Forgot Password?

Or use another login method

Two-factor authentication

Kerberos

By logging in, you agree to comply with the **CERN Computing Rules**, in particular OC5. CERN implements the measures necessary to ensure compliance.

Sign in with your email or organisation

Home organisation - eduGAIN

External email - Guest access

Sign in with a social account

By clicking on the buttons below, you consent to CERN's transfer of your login request to the social provider and to receive your account name, name and e-mail for authenticating you. See more details in our **Privacy Notice**.

Google

GitHub

Facebook

LinkedIn

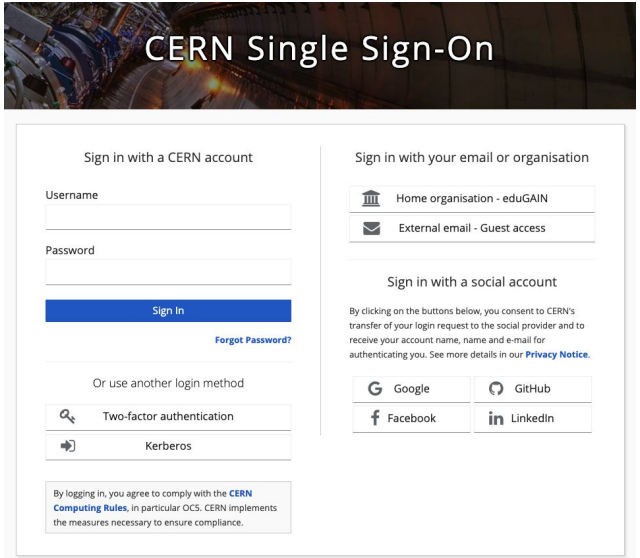
What we will talk about today - and why

- ➔ SSO at CERN
- ➔ Evolution and consolidation of the service
- ➔ The recent upgrade to Keycloak 24
- ➔ Future directions and ideas

CERN Single Sign-On

→ SSO at CERN

Why SSO (Single Sign-On)?



Usability (better user experience)

- One set of credentials to access all of organization's computing resources
- A single login per day

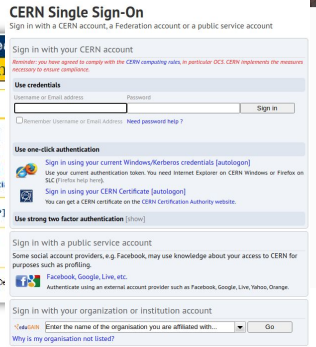
Security

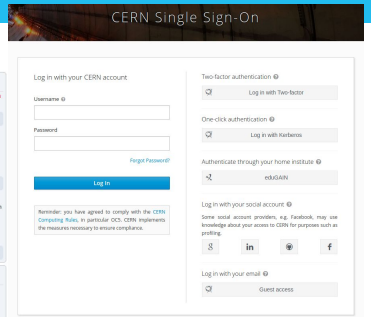
- A central place for enforcing 2FA and password complexity policies, security monitoring and blocking, compromised password detection etc.
- Credentials are not exposed to applications

Cost / efficiency

- No need to implement authentication and authorization in each application separately

The history of SSO at CERN





First SSO introduced (ADFS), presented at HEPiX

MALT project: **Keycloak testing begins** following market survey

Keycloak opened through firewall, available to first CERN adopters

"CERN's new and improved Single Sign-On" presented at CHEP

Major outage during DG talk, **focus on stability & performance**

Moving from Puppet to **Kubernetes hosting**

Old SSO put behind firewall

2007 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024

Over the years: gradual adoption and increased enforcing of **2FA**

Keycloak

Keycloak is an **open-source identity and access management (IAM) solution**

- Provides **single sign-on (SSO)** to organization's applications / resources, with **2FA authentication** (OTP, WebAuthn) and **role-based authorization**
- Allows **user federation** by connecting to LDAP or AD servers (including Kerberos)
- Supports **external Identity Providers (IdP)** and **social logins**
- Uses **standard protocols** such at OAuth 2.0, OpenID Connect (OIDC), and SAML

Keycloak, initially developed by RedHat, is a **CNCF incubation project** since spring 2023

Why on-prem? Why FOSS? Why Keycloak?

+

We operate particle accelerators and experiments

- Full control over configuration, release and patching cycle
- Accessible from the Technical Network

We value openness!

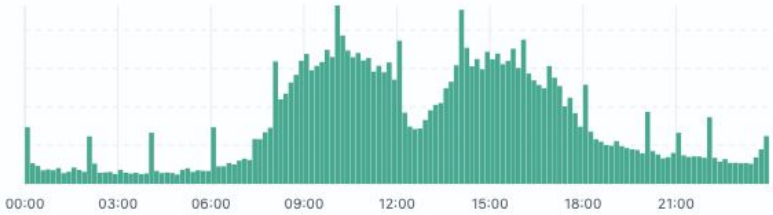
- Open-source is compatible with Open Science / Open Access
- No vendor lock-in, not subject to sanctions or export restrictions

Keycloak fits our needs

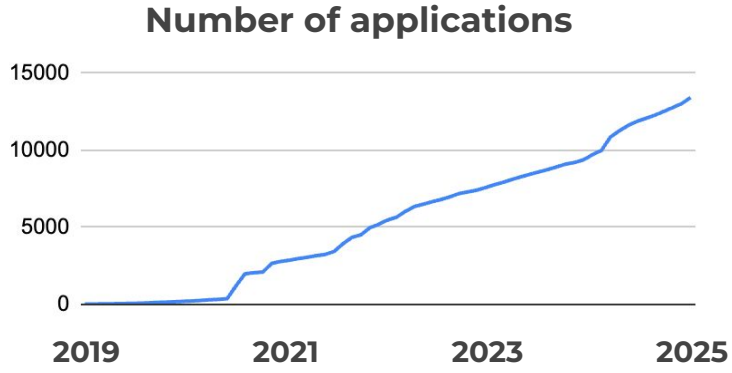
- A lot of big adopters (works at scale)
- A growing usage in academia and research institutes
- Engaged user base, actively developed with frequent releases
- Extensible - can be adapted to our needs

(More at https://auth.docs.cern.ch/documents/why-keycloak)

CERN SSO in a glance



300k users (including externals)
13k clients (applications)
10k logins per hour during office hours



Authentication

- **password authentication** (with Active Directory)
- **Kerberos authentication**
- **2FA authentication** (TOTP, WebAuthn)
- **eduGAIN federated identities**
- **Social logins** (Google, Facebook, GitHub, LinkedIn)
- **Guest accounts** (+ legacy lightweight accounts)

Role-based authorization

- Levels of Assurance (LoA)
- linking accounts of the same identity

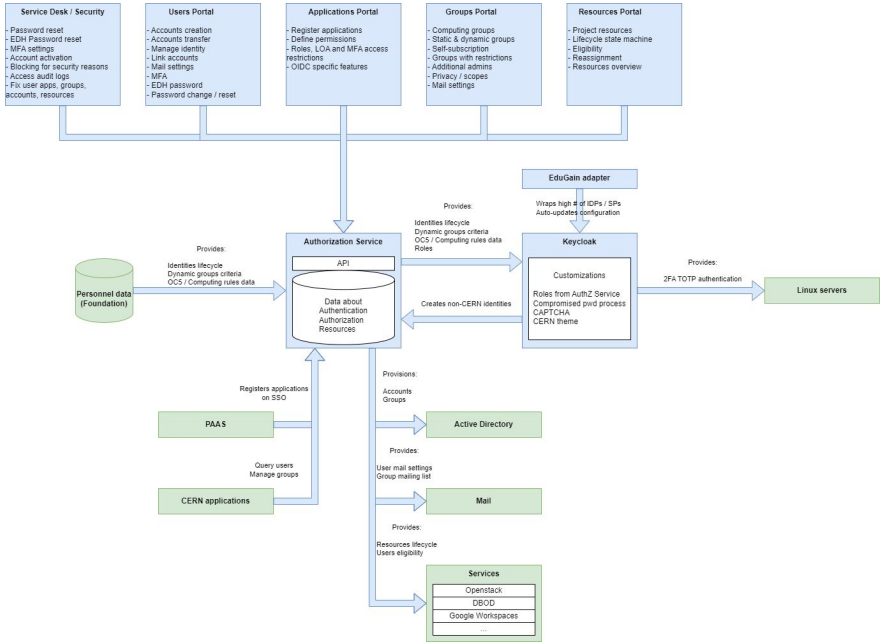
Delivers **SAML assertions** and **OIDC tokens**

Supports **SSH 2FA** (TOTP)

SSO versus **CERN Authorization Service**

CERN Authorization Service:

- **separate from the SSO service** but tightly integrated
- **manages identities and accounts, applications** and their **authorization** (roles, levels of assurance etc.), **groups** (80k)
- **SSO clients are registered** via the Authorization Service API, through the Application Portal where access control can optionally be defined
- Also provides the backend for GMS, Accounts Management and the New Resources Portal



Our CERN-specific Keycloak extensions

CERN Authorization Service integration

- reads and enforces authorization to applications
- creates identities for external accounts on first login

CERN theme

- CERN customisations and look & feel for user-facing login pages, and guest account email template
- *"Message of the day"* (usually, security or service announcements)
- admin console: different header colors per environment

Dev

QA

Production

Our generic Keycloak extensions

`api-access` endpoint (docs)

- extension beyond the OAuth standard to get an OIDC token for a given audience (usually, some API) without doing the full Token Exchange

```
curl --location --request POST
  https://auth.cern.ch/auth/realms/cern/api-access/token
  --data-urlencode 'grant_type=client_credentials'
  --data-urlencode 'client_id=[my-client-id]'
  --data-urlencode 'client_secret=[my-client-secret]'
  --data-urlencode 'audience=[the-target-api]'
```

OTP validation endpoint

- confirms whether a given OTP is currently valid for the given user
- used by a custom PAM module to enforce 2FA on SSH access to sensitive machines

```
> ssh aiadm.cern.ch
(slopiens@aiadm.cern.ch) Your 2nd factor (slopiens):
```

Compromised password detection

- during the login process, SHA1 hash of user's password is checked against a huge list of known compromised passwords (from HIBP and other security sources)

CERN CAPTCHA

- used during guest account registration
- replaces Keycloak's default Google reCAPTCHA (for privacy and availability reasons)

CERN Single Sign-On

→ Evolution and consolidation of the service in the last two years

Timeline

10/2022

DG talk incident

SSO was unavailable and many people could not access the protected webcast to listen to the speech

11/2022

First proposals

The first month after the incident was spent on studying architecture and evaluate possible improvements. These ideas were presented to the Architecture Review Board (ARB)

02/2023

First Proof of Concept

In February a PoC was produced so that SSO team could start the evaluation of new architecture.

09/2023

Production SSO on new architecture and new SSO team

In agreement with ATS and profiting from the first technical stop the SSO was smoothly moved to the new architecture with minimal impact on users

Legacy architecture

One proxy VM to serve Keycloak instances

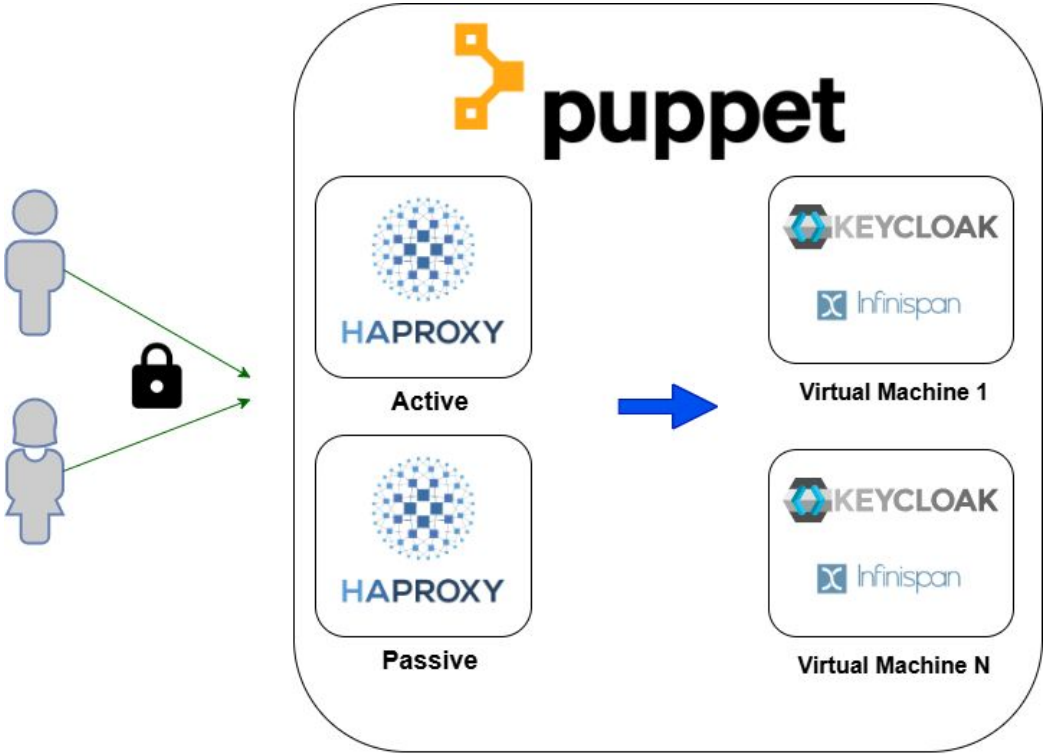
- Switch to passive could take up to 15/20 minutes

Multiple VMs running

- Keycloak and Infinispan sharing same Linux process

Puppet module

- not officially supported by Keycloak



Modernize infrastructure to accelerate team operations

Git source of truth for deployment configuration

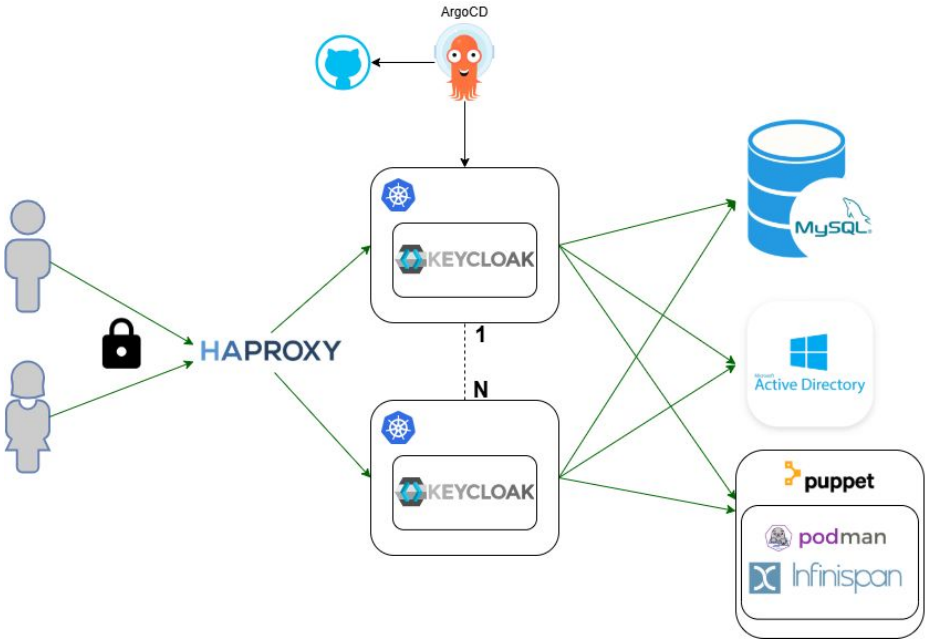
- Operations and updates are traced and easily rolled back
- Unavailability of git repos does **NOT** affect running system
- GitOps components are **NOT** fundamental

Separation of Keycloak and Infinispan

- Components can be scaled, tuned and monitored independently
- Simplify operations
  - Keycloak becomes stateless!
- Keycloak with Kubernetes Cattle service model

HA HAproxy cluster

- HA cluster with automatic failover with no downtime



Modernize infrastructure to accelerate team operations

Monitoring and Logging improvements

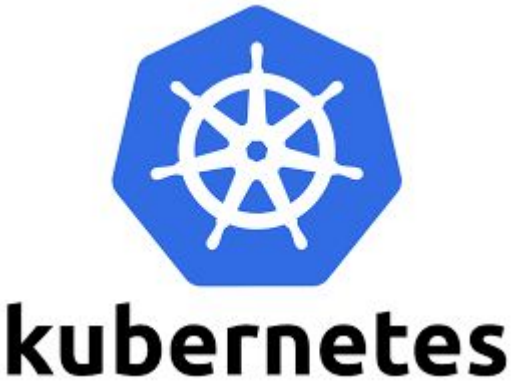
- Implement widely adopted standard tools such as:
  - Fluent Bit
  - Prometheus
- Infinispan metrics
- Log parsing simplified debugging.

Strong dependencies:

- DNS and Network
- Active Directory
- Database

All other dependencies are optional and can be easily bypassed (more info in BC/DR and mini-SSO part)

Why Kubernetes (K8s)



Red Hat direction clear

- Jboss replaced by Quarkus (designed for Kubernetes)
- Red Hat maintains Kubernetes operator for deployment

Portable, reproducible and Immutable

- Speeds up operations, reducing team effort
  - Kubernetes is not intended for improving performance!
- Facilitate BC/DR

Easier to maintain and deploy in long term

- Vibrant community supporting Kubernetes (Keycloak is CNCF incubating project)
- Small community in Puppet world; one main maintainer for the Puppet module

Profiting of experience running production service on K8s since many years (2020)

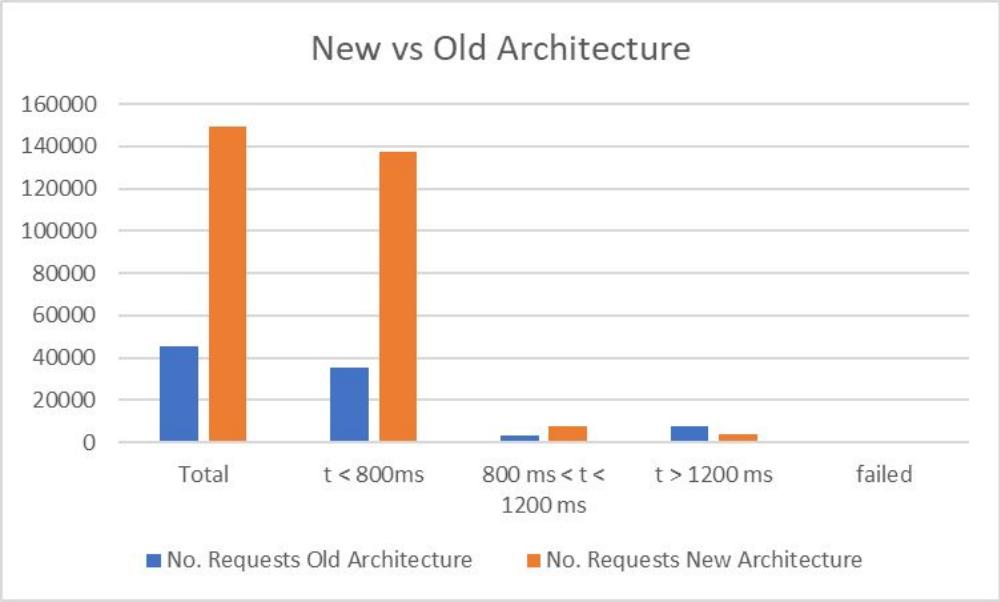
- Extended experience within the IT-PW group (this is a cross section collaboration)

Stress tests and results

Performance improved by **300%**

Reduced infrastructure service incidents by **86%** (7 vs 1 OTGs)

Authentication experts can focus on their own domain



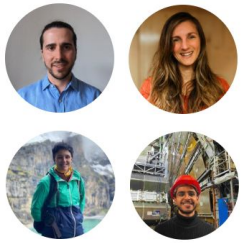
Stress tests and results

Pe

Re
in

Au
th

September 2023 - a very challenging moment!



- 😱 The previous team (Asier, Maria, Adeel, Hannah) **all** left
  - 2 end of contracts, 1 resignation, 1 long-term leave
- ✅ Antonio provides infrastructure (hosting, HAproxy, Infinispan)
- 😬 Sebastian and Paul took over the SSO service
  - basically, a brand new team - learning from scratch
  - (very) limited overlap with the previous team

Service well designed, with strong foundations, and widely used.

At the same time, considerable work remaining to consolidate
and to reach full maturity that everyone desired for a critical service.

Review of the service → **SSO Masterplan**

Some examples in the following slides (that we hope might be useful for other teams!)

CERN SSO service - Masterplan 2023-2024

Author: Sebastian.Lopienski@cern.ch

NB: This document, first published in November 2023, is a live document and will be updated regularly (priority changes, tasks marked as done, new points added etc.).

Priorities (from highest to lowest)

	Priority
1	Very important, urgent (or easy to complete quickly)
2	Very important but less urgent
3	Important but not urgent (or much bigger task)
4	Less important but should be done eventually
5	Nice to have

JIRA

75 epics

636 tasks

Service clean-up and consolidation

- ✅ DONE 3 Review and complete SSO/Keycloak internal docs (link)
- 🛠 IN PROGRESS 2 Handle and monitor key/certificate expiration
- 🛠 IN PROGRESS 3 Review and clean up Keycloak configuration
- ✅ DONE 3 Review SSO deployment configuration
- ✅ DONE 3 Clean up Keycloak databases, database accounts and backups
- ✅ DONE 4 Clean up Keycloak docker images
- ✅ DONE 4 Clean-up obsolete SSO aliases, VMs, Puppet environments etc.
- ✅ DONE 5 Clean up SSO CronJobs

Regular updates

- ✅ DONE 2 Test and document the deployment process (Docker image, SPIs
- ✅ DONE 2 Update `keycloak-cern-theme`
- ✅ DONE 2 Update `keycloak-cern-providers`
- ✅ DONE 3 Update `keycloak-stepup-rest`
- ✅ DONE 3 Update `keycloak-cern-captcha`
- 🛠 IN PROGRESS 4 Update CAPTCHA API
- 4 Update and review `health-checks-spi`
- 4 Review `keycloak-rest-adapter`
- ✅ DONE 4 Update `metrics-spi`
- 🛠 IN PROGRESS 2 Upgrade eduGAIN components

Service monitor

- ✅ DONE 3 Consolidate integration tests
- ✅ DONE 2 Add new integration tests
- ✅ DONE 2 Document "sanity checks"
- ✅ DONE 3 Improve canary (availability) test
- 🛠 IN PROGRESS 1 Monitor and follow-up SSO service abuses
- 🛠 IN PROGRESS 3 Enhance SSO logging, monitoring and alerting
- 🛠 IN PROGRESS 3 Create and gather SSO metrics
- 2 Investigate, fix and monitor Keycloak system errors
- ✅ ONGOING 4 Regular service maintenance and checks
- ✅ ONGOING 4 One-off service tasks

Service security and hardening

- ✅ DONE 1 Review and clean up access lists (admin, developers, ma
- 🛠 IN PROGRESS 1 Review database accounts and their privileges
- ✅ DONE 1 Monitor Keycloak configuration changes
- 🛠 IN PROGRESS 2 Change and protect passwords/secrets
- ✅ DONE 2 Review GitLab repositories and their security settings

BC/DR preparations

- ✅ DONE 1 Plan, test and document cold recovery
- 🛠 IN PROGRESS 1 Prepare breaking-glass accounts
- ✅ DONE 2 Document and test emergency procedures
- 2 Create SSO Business Continuity Plan for PDC
- 🛠 IN PROGRESS 2 Various BC/DR preparations
- 🛠 IN PROGRESS 3 Display a static info page during interventions or

Reviewing all elements of the service

A systematic effort to discover, learn, update, clean-up and document:

- Keycloak configuration
  - environments (dev/QA/prod)
  - realms (cern/mfa/kerberos/etc.)
  - hidden and obsolete settings
  - etc.
- CERN Keycloak extensions (SPIs)
- Docker images and deployment configuration
- Databases
- Certificates and keys
- PaaS CronJobs
- Integration tests
- ... and many more

Note: This work came at the expense of other features and deliverables that had been promised.

We feel the concentration of effort is a worthwhile investment and is already paying off.

Managing and auditing Keycloak configuration changes

❓ The challenge: Losing track of changes

- Most **Keycloak configuration updates** happen via the Admin Console
- But **manual changes leave no history**—creating tracking challenges

📂 Our solution: Automated backups

- Every hour, a script **exports all realm configurations** from Keycloak
- The exported and processed JSON files are committed to a **GitLab repository**
- Every change triggers an **alert** in our Mattermost channel

Keycloak config change BOT 2:00 PM

keycloak-config-backups-script pushed to branch master of authzsvc / config-backups / keycloak-config-backups (Compare changes)

> e8f18fef: Keycloak PROD configuration updated - keycloak-backups

📖**Take-home message**📖
Having full visibility into every change—what, when, and why

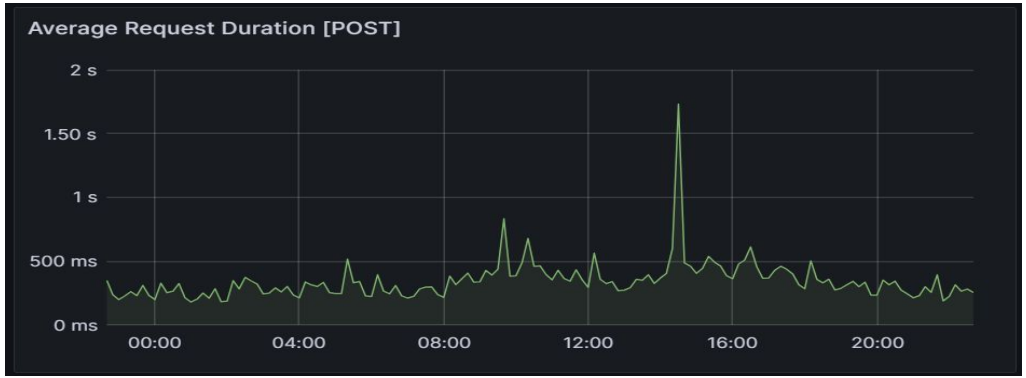
Monitoring and alerting

📊 Grafana: Proactive monitoring

- **Logs-based** insights → **Login trends**, token exchange usage, ...
- **Metrics-based** insights → Node status, **response time**, DB & Infinispan health, ...

⚠️ Alerting setup

- Mattermost channel for real-time notifications
- **Canary test** to monitor SSO status and send **Telegram alerts**
- Automated SSO integration tests on PaaS with Selenium



📖**Take-home message**📖
Monitoring detects issues—but also provides data to optimize performance

Performance Optimization & Stress Testing

🔴 **Example** of performance bottleneck revealed by Grafana

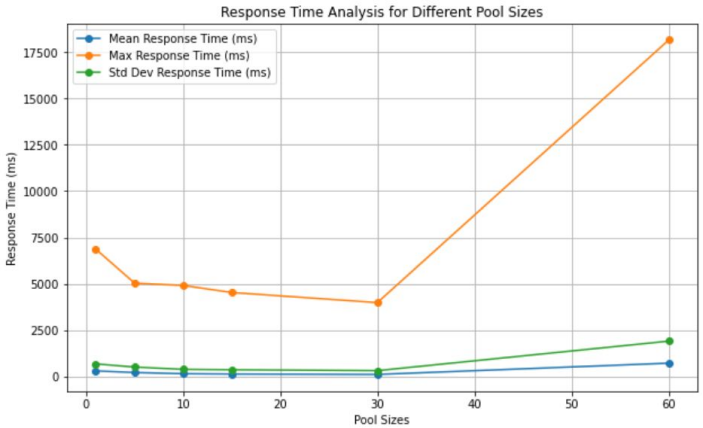
- Custom SPI for Keycloak connects to authorization service DB
- Under **heavy load**, too many connections, and it **slowed down** the database (**OTG0151188**)

✅ **Solution:** Fine tuning of connection pool

- We integrated HikariCP to **control and optimize** database connections

⚡ Stress testing

- In that example, tested various pool sizes to balance performance
- **Always conduct extensive stress tests** for updates!



Security and hardening
(we don't need it, until we need it)

Review, clean-up and document **privileged access**
- admin egroups (access lists)
- service accounts
- database accounts and their privileges

Check **access restrictions**, enforce **2FA**, and increase **traceability** for all infrastructure components, and deployment process
- including database backups, internal tools / systems etc.

Change/renew shared **passwords, tokens, keytab files** etc.

Review **GitLab security settings** (with a dedicated tool)
- project visibility
- members (especially those added individually and not via egroups) and their access
- secrets in GitLab variables, tokens
- security-related settings such as branch protection, push and merge restrictions etc.

BC/DR
(to sleep better at night)

🚨 Establish, document and test **emergency procedures**

- disabling 2FA authentication *(proved useful during Raivo OTP app failure)*
- **scale up/down Keycloak clusters** *(proved useful during Keycloak 24 upgrade)*
- restarting remote **Infinispan cache** container; switching to local Infinispan cache
- **database** backup, cloning and recovery; switching from database proxy to master

🔑 Prepare local **breaking-glass accounts**

- in ArgoCD, OpenSearch, Grafana, VM with internal tools etc.
- to avoid circular dependency on SSO

🥶 Cold-recovery test → ✅ **mini SSO instance**

- can be built anywhere, on a Linux machine, in one hour ⏳
- using a regularly synchronized **offline backup of all SSO service components** (Docker image, Keycloak extensions, configuration, secrets, certificate and docs)
- external dependencies: the Active Directory, a database, and switching `auth.cern.ch`

SSO Masterplan

Status as of today →

```
## Service clean-up and consolidation

Admin docs                          : ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅
Key/certificate expiration          : ✅ ⬜ ✅ ➖ ✅ ⬜ ✅ ⬜
Keycloak config clean-up            : ⬜ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ⬜ ⬜ ⬜ ⬜ ✅ ✅ ✅ ⬜ ⬜ ⬜ ⬜ ⬜ ✅ ⬜ ✅
Keycloak deployment config          : ✅ ✅ ✅ ✅ ✅ ✅ ✅ ⬜ ✅ ✅ ✅ ✅ ✅ ✅ ✅
Keycloak database cleanup           : ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅
Keycloak docker images              : ✅ ✅ ✅ ✅ ✅ ✅ ✅
SSO deployment clean-up             : ✅ ✅ ✅ ✅ ✅ ⬜
CronJobs clean-up                   : ✅ ✅ ❌ ✅

## Regular updates

Update keycloak-cern-theme          : ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅
Update keycloak-service-providers   : ✅ ➖ ➖ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ➖ ✅ ✅ ⬜
Update keycloak-step-up-rest        : ✅ ✅ ✅ ✅ ✅ ✅ ✅
Update keycloak-cern-captcha        : ✅ ✅ ✅ ➖ ✅ ✅
Update CAPTCHA API                  : ✅ ⬜ ⬜ ✅ ✅ ⬜
Update health-checks-spi            : ✅ ✅ ⬜ ✅
Upgrade eduGAIN components          : ➖ ➖ ✅ ⬜ ⬜ ⬜ ✅ ✅

## Service upgrades and major changes

Keycloak 24                         : ✅ ✅ ✅ ✅ ✅ ✅ ✅ ⬜ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ⬜ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ⬜ ✅
MFA realm phase-out                 : ⬜ ⬜ ⬜ ✅ ✅

## Usability and service improvements

2FA usability                       : ❌ ➖ ⬜ ➖ ⬜ ⬜ ⬜ ✅ ✅ ✅ ✅ ❌ ✅ ⬜ ❌ ✅ ✅ ⬜ ⬜ ⬜ ⬜ ✅
Application registration            : ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅
Application management              : ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ⬜ ⬜ ✅ ✅
Improve application portal usability : ⬜ ✅ ⬜ ⬜ ⬜ ⬜ ✅ ✅ ✅ ⬜ ➖ ⬜ ✅ ⬜ ⬜ ⬜ ⬜ ⬜ ✅ ✅ ⬜ ⬜ ⬜ ⬜ ⬜ ✅ ⬜ ⬜ ⬜ ⬜ ⬜ ⬜
Password issue detection            : ⬜ ⬜ ⬜ ✅ ⬜ ✅
Password change/reset               : ⬜ ⬜ ➖ ⬜ ⬜ ⬜
More IdPs                           : ⬜ ✅
Guest accounts improvements         : ✅ ✅ ✅ ✅ ❌ ✅ ➖ ⬜ ✅
Keycloak theme improvements         : ✅ ✅ ⬜ ✅ ✅ ✅ ✅ ✅ ✅ ⬜ ✅ ✅ ➖ ➖ ⬜ ✅ ✅ ✅ ✅ ✅
Keycloak investigations             : ⬜ ⬜ ⬜ ✅ ⬜ ⬜ ⬜ ⬜ ⬜ ⬜ ⬜ ⬜ ⬜ ⬜ ⬜ ✅ ✅ ✅ ✅ ⬜ ⬜ ⬜ ⬜ ⬜ ⬜ ✅ ⬜ ⬜ ⬜
SSO features requests               : ⬜ ⬜ ⬜ ✅

## User tools and support

User docs                           : ✅ ✅ ⬜ ✅ ✅ ✅ ⬜
SSO examples                        : ⬜ ✅ ✅ ✅ ✅ ⬜ ⬜ ✅ ⬜ ✅ ⬜ ✅ ✅ ✅ ⬜ ✅
auth-get-sso-cookie improvements    : ⬜ ➖ ✅ ✅ ⬜ ⬜ ⬜ ✅ ⬜ ⬜ ⬜ ⬜ ⬜ ➖ ⬜
auth-get-sso-cookie and 2FA         : ✅ ✅ ✅ ⬜

## Service monitoring and operations

Consolidate integration tests       : ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ❌ ✅ ✅ ✅ ✅ ✅ ⬜ ✅ ⬜ ✅ ✅ ✅ ✅ ⬜ ⬜
Add more integration tests          : ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅
SSO abuses                          : ⬜ ✅ ⬜ ⬜ ✅ ➖ ⬜
SSO monitoring                      : ⬜ ⬜ ✅ ✅ ⬜ ➖ ✅ ✅ ⬜ ✅ ⬜ ✅ ✅
SSO metrics                         : ➖ ➖ ✅ ✅ ⬜
Keycloak server errors              : ⬜ ⬜ ⬜
SSO tasks                           : ✅

## Service security and hardening

Groups and access clean-up          : ✅ ✅ ✅ ✅ ✅ ✅ ✅
Keycloak config monitoring          : ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅
Secrets                             : ⬜ ✅ ⬜ ⬜ ⬜ ⬜ ✅ ⬜ ⬜

## BC/DR preparations

SSO cold recovery                   : ✅ ✅ ✅ ✅ ✅ ✅ ✅ ✅ ➖
Emergency procedures                : ✅ ✅ ➖ ✅ ✅
SSO BC/DR                           : ⬜ ✅ ⬜ ⬜ ⬜ ⬜ ⬜ ⬜ ✅ ⬜ ⬜
```

SSO Masterplan

Progress over time →

```
## Service clean-up and consolidation

Admin docs                          : ⬜ ✅
Key/certificate expiration          :
Keycloak config clean-up            :
Keycloak deployment config          :
Keycloak database cleanup           :
Keycloak docker images              :
SSO deployment clean-up             : ⬜
CronJobs clean-up                   :

## Regular updates

Update keycloak-cern-theme          :
Update keycloak-service-providers   : ⬜
Update keycloak-step-up-rest        :
Update keycloak-cern-captcha        :
Update CAPTCHA API                  :
Update health-checks-spi            :
Upgrade eduGAIN components          : ⬜

## Service upgrades and major changes

Keycloak 24                         :
MFA realm phase-out                 : ⬜ ⬜

## Usability and service improvements

2FA usability                       : ⬜ ⬜ ⬜ ⬜ ⬜ ⬜ ⬜ ⬜
Application registration            :
Application management              :
Improve application portal usability :
Password issue detection            :
Password change/reset               :
More IdPs                           : ⬜
Guest accounts improvements         : ⬜ ⬜
Keycloak theme improvements         : ⬜ ⬜
Keycloak investigations             : ⬜ ⬜ ⬜
SSO features requests               : ⬜ ⬜ ⬜

## User tools and support

User docs                           :
SSO examples                        : ⬜
auth-get-sso-cookie improvements    : ⬜ ⬜
auth-get-sso-cookie and 2FA         : ⬜ ⬜ ⬜

## Service monitoring and operations

Consolidate integration tests       :
Add more integration tests          :
SSO abuses                          :
SSO monitoring                      : ⬜
SSO metrics                         : ⬜ ⬜
Keycloak server errors              :
SSO tasks                           :

## Service security and hardening

Groups and access clean-up          :
Keycloak config monitoring          :
Secrets                             :

## BC/DR preparations

SSO cold recovery                   :
Emergency procedures                :
SSO BC/DR                           :
```

Uptime statistics

- Market alternatives seem to aim for a max downtime of roughly 20 minutes per month.
- In general, CERN SSO has consistently been equivalent or better over the past 18 months.
- Recent incidents increasingly related to external factors rather than internal; e.g. DBOD (note: suboptimal use also from our side - being investigated), Storage, Networking.

Service Incidents by Year	Self caused	Within IT	External to CERN	Other at CERN	Total
2023	15	2	1	1	19
2024	5	7	1		13

Google Identity: “The Covered Service will provide a Monthly Uptime Percentage to Customer of at least 99.95%”

Recent worldwide SLA performance

To help you plan for moving workloads to Microsoft Entra ID, we publish past SLA performance. These numbers show the level at which Microsoft Entra ID met the requirements in the SLA for Microsoft Entra ID, for all tenants.

The numbers in the table are a global total of Microsoft Entra authentications across all customers and geographies. The number is truncated at three places after the decimal. Numbers aren't rounded up, so actual SLA attainment is higher than indicated.

Expand table

Month	2021	2022	2023	2024
January		99.998%	99.998%	99.999%
February	99.999%	99.999%	99.999%	99.999%
March	99.999%	99.999%	99.999%	99.999%
April	99.999%	99.999%	99.999%	99.999%
May	99.999%	99.999%	99.999%	99.999%
June	99.999%	99.999%	99.999%	99.999%
July	99.999%	99.999%	99.999%	99.999%
August	99.999%	99.999%	99.999%	99.999%
September	99.999%	99.998%	99.999%	99.999%
October	99.999%	99.999%	99.999%	99.998%
November	99.998%	99.999%	99.999%	99.998%
December	99.978%	99.999%	99.999%	99.998%

Entra: achieves roughly 99.998%

CERN Single Sign-On

→ The recent upgrade to Keycloak 24

CERN Single Sign-On upgrade to Keycloak 24 OTG0149673

Type: Planned Intervention
Begin: Wed Jan 15, 2025 17:30
End: Wed Jan 15, 2025 22:00
Impact: Down
Last Updated: Thu Jan 16, 2025 17:45
Locations: Not Specified
SE Single Sign On and Account Management Services
FE Authentication
Services Affected: Single Sign On and Account Management Services

Description:
CERN SSO service (Single Sign-On, at auth.cern.ch) will be upgraded to Keycloak 24 on January, 15th 2025.

The QA environment was upgraded to Keycloak 24 on Monday, November 11th at 4PM. Application owners are strongly encouraged to test their applications against this QA environment as soon as

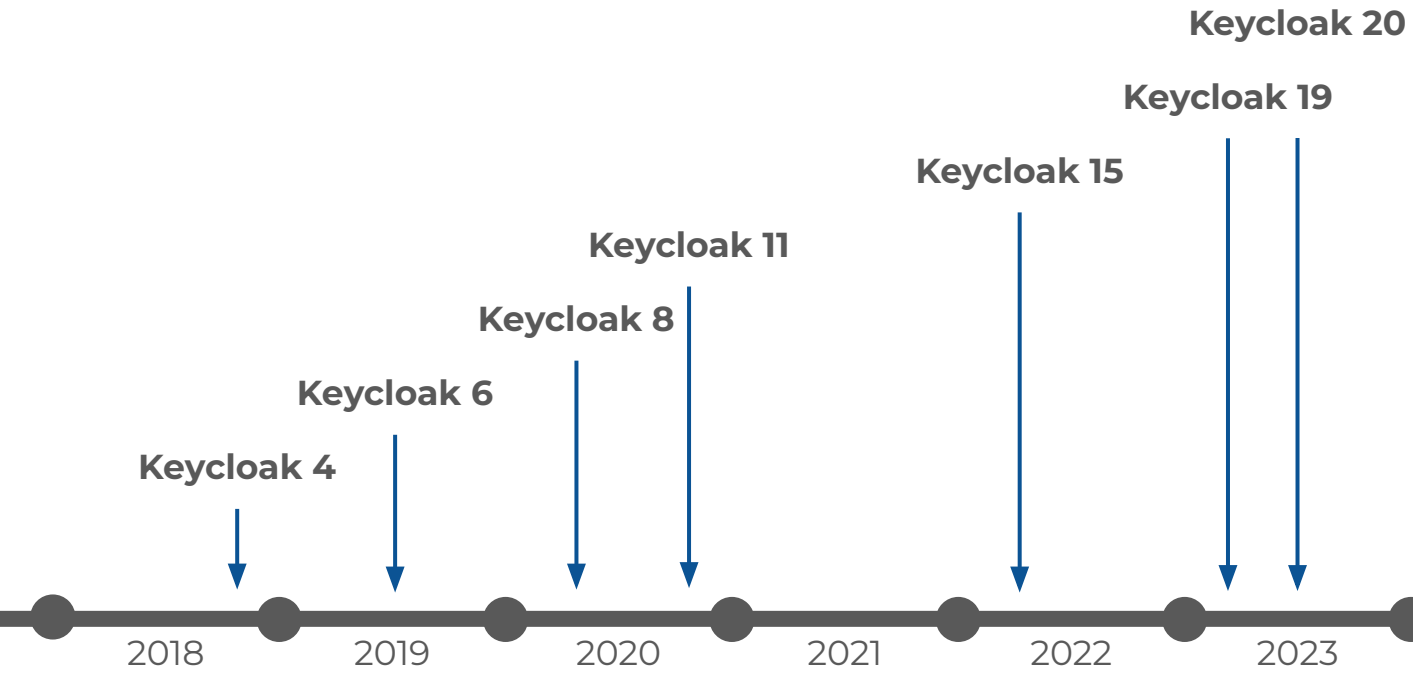
Why upgrading

Users don’t request SSO upgrades

However, we want and need to upgrade regularly, for many reasons:

- Keep up to date with Keycloak releases
- Get minor bug fixes, various new features, usability improvements
- Avoid technical debt (the longer we wait, the harder to upgrade)
- Maintain security (patches to possible security vulnerabilities may not be backported to previous Keycloak versions)

Keycloak versions at CERN - what we learned



Don't use version X.0.0 - wait for a more mature X.0.5+

Communicate ++

Always surprise breaking changes

Many client issues due to custom settings, obsolete libraries, not following standards…

QA testers hugely valuable

Planning and communicating the intervention

- **Timing constraints**:
  - Give application owners enough time for test on QA
  - During LHC and Injectors' Technical Stops
  - Avoid Council weeks and other VIP events
  - Not just before the annual closure
  - Avoid Experiment operators' shift handover
- **Formal steps**:
  - CRMB
  - IMPACT form
- The **initial plan** for Keycloak 23 (March → June 2024) vs. the **final plan** for Keycloak 24 (→ January 2025)
- **Communication**: OTG + documentation, Auth & SSO Mattermost channel, ITUM, SSO mailing list, CNIC, engagement channels, direct contacts etc.

Validation

🔍 Rigorous performance validation

- Our **systematic approach**—document findings and validate major changes in the past with stress tests—proved **essential**
- Analyzed Keycloak pods for performance bottlenecks (garbage collection, threading)

🛠️ Service quality check

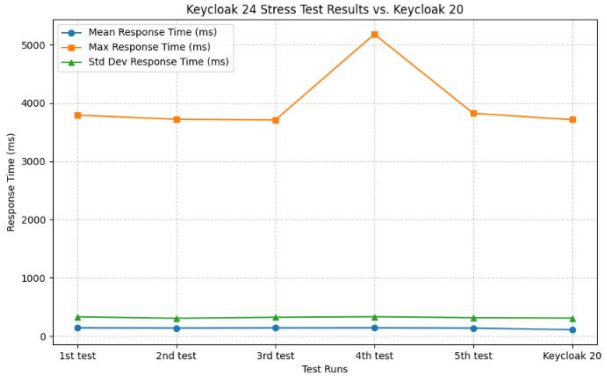
- Conducted sanity checks in every environment
- Coordinated with application owners to **test in QA**

✅ Pre-upgrade simulation – No surprises on release day

- Built a parallel infrastructure one month in advance, **mirroring production**
- Full upgrade **simulation** to identify potential risks

🔄 Prepared for the unexpected

- Developed a **rollback strategy** to ensure fast recovery in case of unexpected issues



📖**Take-home message**📖
Upgrading a critical system like SSO isn't just about deploying a new version—it's about confidence

Intervention checklist

During the upgrade, we followed a detailed intervention checklist, prepared in advance - helps us to focus and reduces stress!

This is our “standard” checklist →

(The Keycloak 24 upgrade checklist was much longer and more detailed!)

Just before

- [ ] Make the OTG public
- [ ] Update on both Mattermost channels:
  “FYI, an SSO intervention will start in a few minutes: OTGxxxxxx”
- [ ] Run integration tests (locally or on openshift)
- [ ] Export Keycloak config
- [ ] Take a DBOD snapshot
- [ ] Dump `keycloak` database on auth-tool-prod

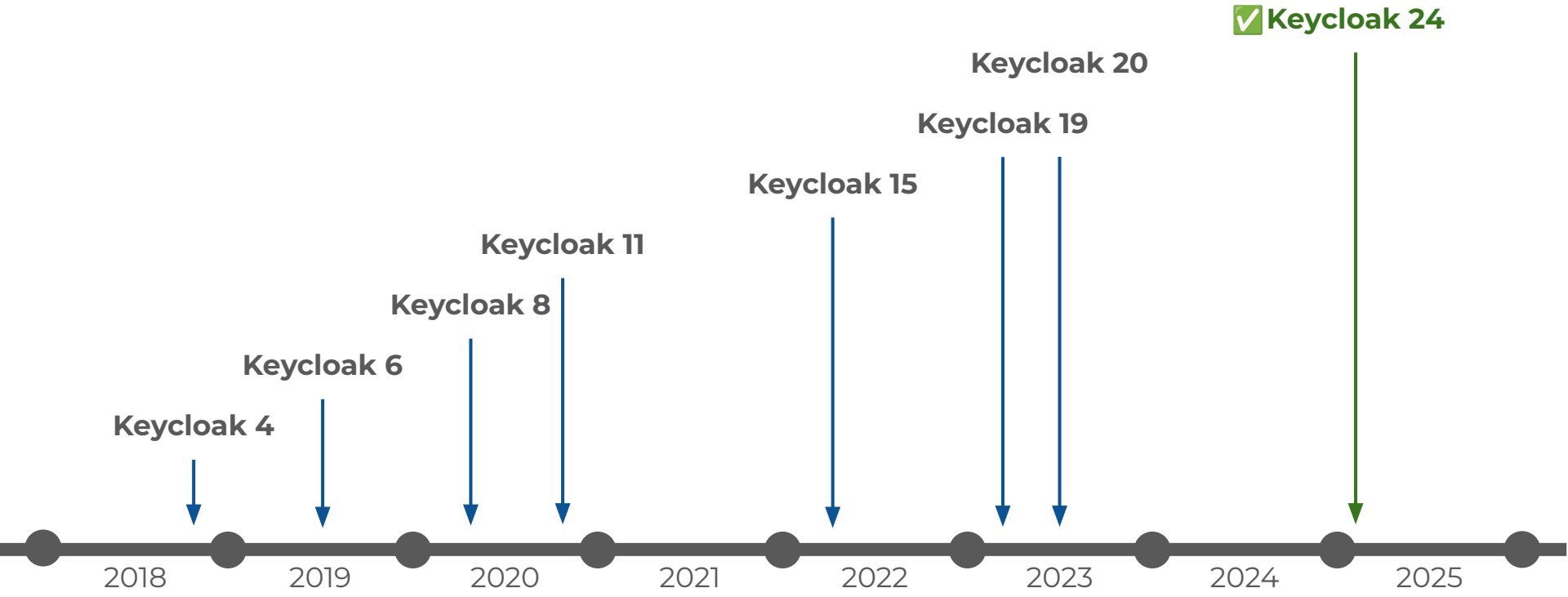
Actions

- [ ] Hard refresh all applications
- [ ] Check “App Diff” in ArgoCD
- [ ] Restart pods one by one, and try to log in using that pod (cluster)
- [ ] Check Keycloak logs in ArgoCD -> application -> Keycloak pod -> details -> logs
- [ ] Check Keycloak logs in OpenSearch, e.g. warnings and errors
- [ ] Check “Error” section in Grafana
- [ ] Implement any other steps of the intervention...

After

- [ ] Export Keycloak config and check changes
- [ ] Run integration tests
- [ ] Make the OTG CERN-only
- [ ] Inform both Mattermost channels that the intervention is finished
- [ ] Remove `sso-intervention` label from the relevant JIRA tickets
- [ ] Move OTG notes to the list of “Past interventions”

Keycloak versions at CERN



Follow-up actions. Lessons learnt from KC24?

🛠️ Post-upgrade challenge example

- After the upgrade, one minor system kept using the old database for a full day
- Lesson learned: Changed old DB access rights to **read-only**—now, unnoticed systems will trigger errors **instead of silently failing**

🔍 QA testing – The key to a smooth rollout

- Many applications don't test against QA SSO, **making upgrades riskier**
- One application owner tested early and discovered an issue with Keycloak 24's new OIDC "iss" (issuer) field
- His test allowed us to **prepare a fix** (SQL script) and **streamline support**
- Because we **advertised this issue**, we handled the few cases of incompatibility quickly post-upgrade

📖**Take-home message**📖
Through proactive testing, QA app owners played a key role in ensuring a seamless and efficient upgrade

CERN Single Sign-On

→ Future directions and ideas

Future directions and ideas

- (Scheduled) Simplify 2FA internal architecture (removal of `mfa` realm)
- (Up next) Plan next upgrades, and establish an upgrade strategy
- Improve Keycloak configuration management (GitOps approach)
- Usability improvements, e.g.
  - multiple 2FA tokens of the same type
  - remember the last-used 2FA method per device
  - advertise existing but rarely used 2FA factors (fingerprint readers, FaceID etc.)
- Other improvements, e.g.
  - add AppleID as an external IdP (required for iOS apps that use CERN SSO)
  - extend compromised & non-compliant passwords detection
- Infrastructure:
  - multi-site setup (PDC setup and maybe public cloud)
  - host Infinispan on Kubernetes

Key messages from the past 2 years

- SSO has seen huge improvements, reflected in performance and reliability
- Right personnel allocation essential - the team (Antonio, Paul, Sebastian) really stepped up in a difficult situation and built on excellent groundwork by past colleague
- Use standard tools and building blocks wherever possible
- Benefit from the expertise of other teams to optimize the service (Tomcat service/Jeedy (Antonio et al.), DBOD)
- Slowing down and investing time in service management best practices seems to be worthwhile, though requires difficult compromises

We will be building on this stable, reliable, reproducible service to bring new features to Service Managers and End Users.

Watch this space.

CERN Single Sign-On

Thank you for your attention!

Backup slides

Challenges with upgrading - past experiences

Major version upgrades occasionally bring (unexpected) breaking changes - some examples:

- **Keycloak 19**:
  - Keycloak added line breaks in SAML response signature
  - Ports stripped from SAML client return URLs
- **Keycloak 20**:
  - "`openid`" scope became mandatory in calls to UserInfo endpoint (to make it standard-compliant)
- **Keycloak 23**:
  - The operator imposes "`--optimized`" option, which requires Keycloak to be fully configured and "built" in the Docker image (not compatible with our deployment model; fortunately this was reverted in Keycloak 24)
- **Keycloak 24**:
  - Keycloak became more strict with the redirect URL
  - Added the issuer parameter to the authentication flow, breaking the clients using an old OIDC/Keycloak authentication facilitator.
  - Some attributes removed from token introspection responses.

Keycloak 24 upgrade - technical preparations

- “Easy” tasks (a few examples):
  - Java 11 -> 17
  - javax -> jakarta
  - SPI deprecations
  - Deprecated crypto
  - Update third-party extensions
  - Infinispan cache 13 -> 14
  - Kubernetes 1.25 -> 1.30
- Less obvious tasks:
  - Investigate all migration changes and release notes
  - Identify and check potentially affected applications
  - Run stress tests; identify and fix possible performance degradations

Past Keycloak upgrade experiences

- Don’t use version X.0.0 - wait for a more mature X.0.5+
- Usually, the upgrades are smooth
  - few breaking changes, reasonably well documented
  - vast majority of applications not impacted
  - users don’t even notice the upgrade
- There are always changes that break some applications
  - usually because of custom settings, using obsolete libraries or weak crypto, not following standards etc.
  - we are documenting known changes, and proactively addressing them with application owners. However there are always some surprises.
- It’s crucial that applications test in QA SSO environment
  - however, few applications use QA SSO
  - we need to follow up critical applications individually