The Hard Life of securing a Particle Accelerator

Antonio Nappi, CERN
Sebastian Łopieński, CERN

Speakers

Antonio Nappi, CERN

- In charge of hosting Java applications at CERN
- Kubernetes engineer
- Previously, Sysadmin, OpenStack and Python consultant

Sebastian Łopieński, CERN

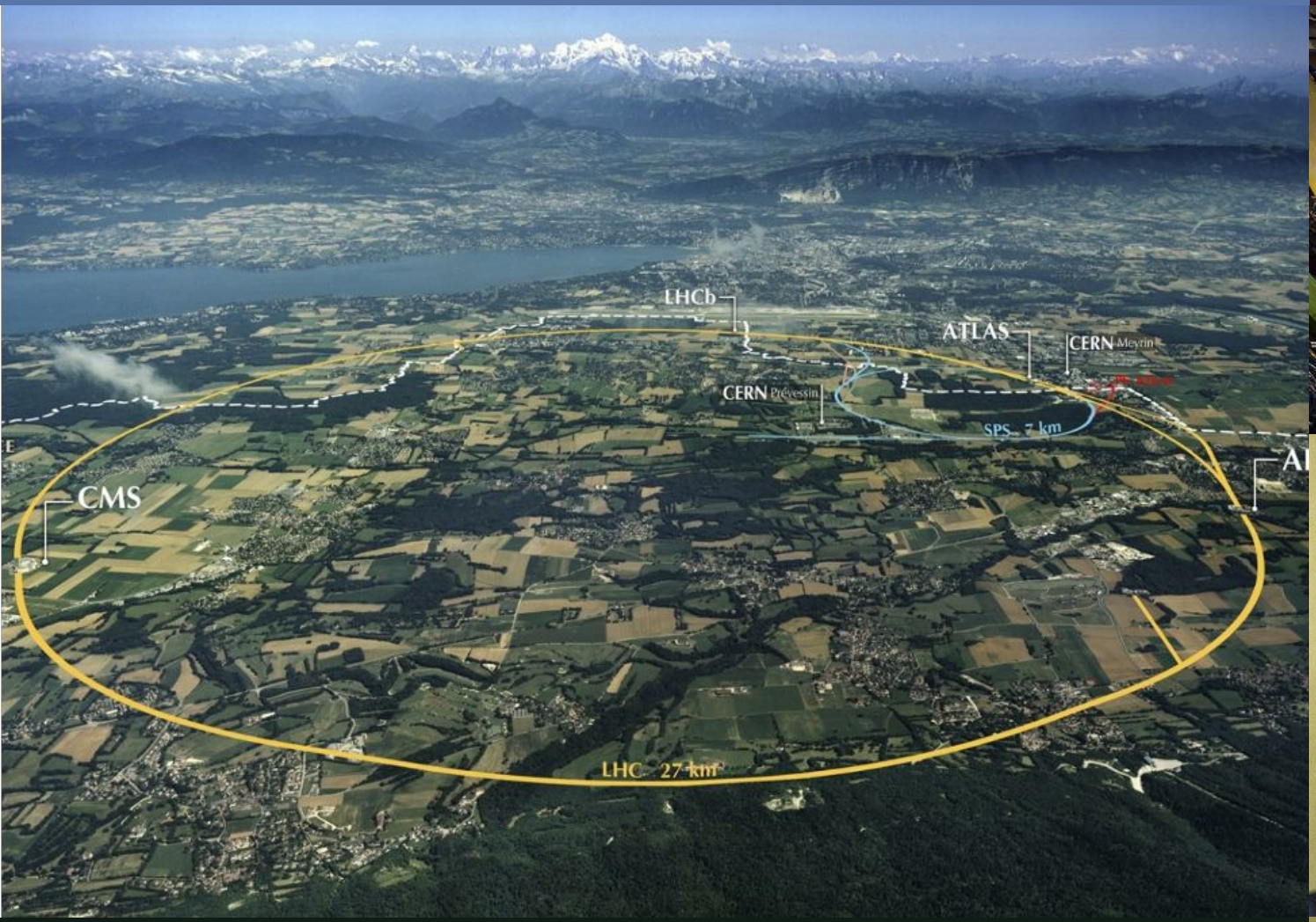
- CERN Single Sign-On service manager
- Previously, 15 years as CERN Deputy Computer Security Officer
- Background: software engineering

CERN - European Laboratory for Particle Physics

CERN - European Laboratory for Particle Physics

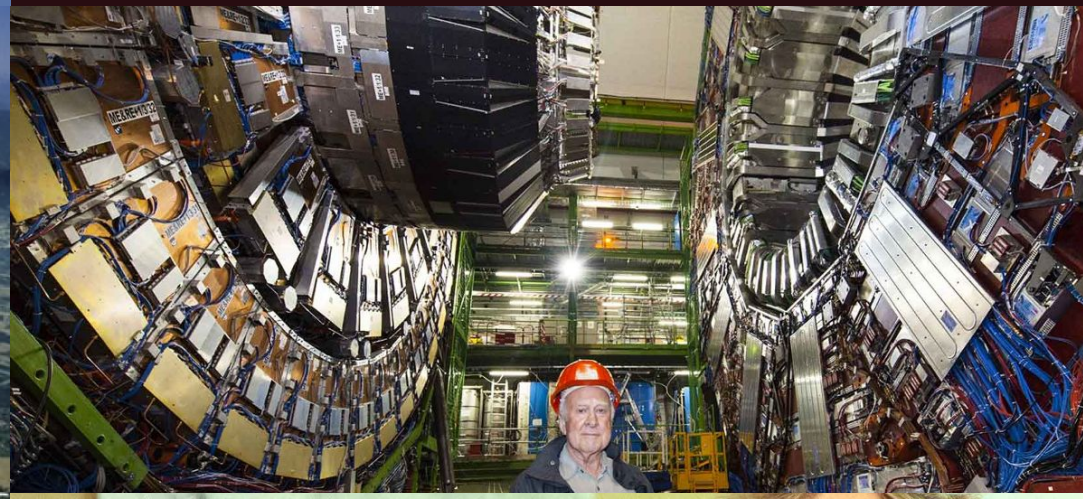
KubeCon CloudNativeCon Europe 2024

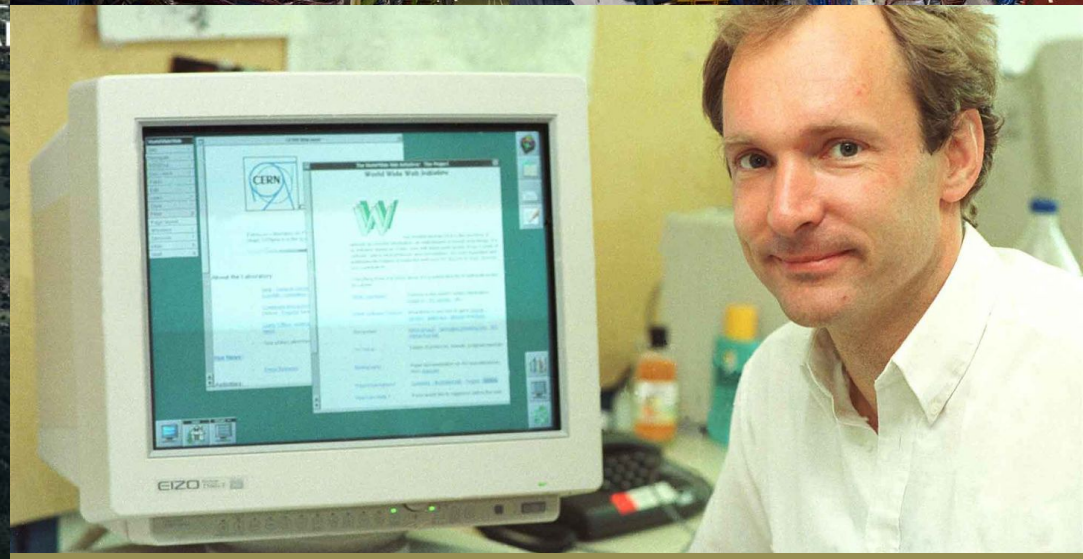
Large Hadron Collider 27km long, 100m underground



over 15'000 scientists, 100 nationalities

2012: Higgs boson discovered





1989: Web invented by Tim Berners-Lee

Outline

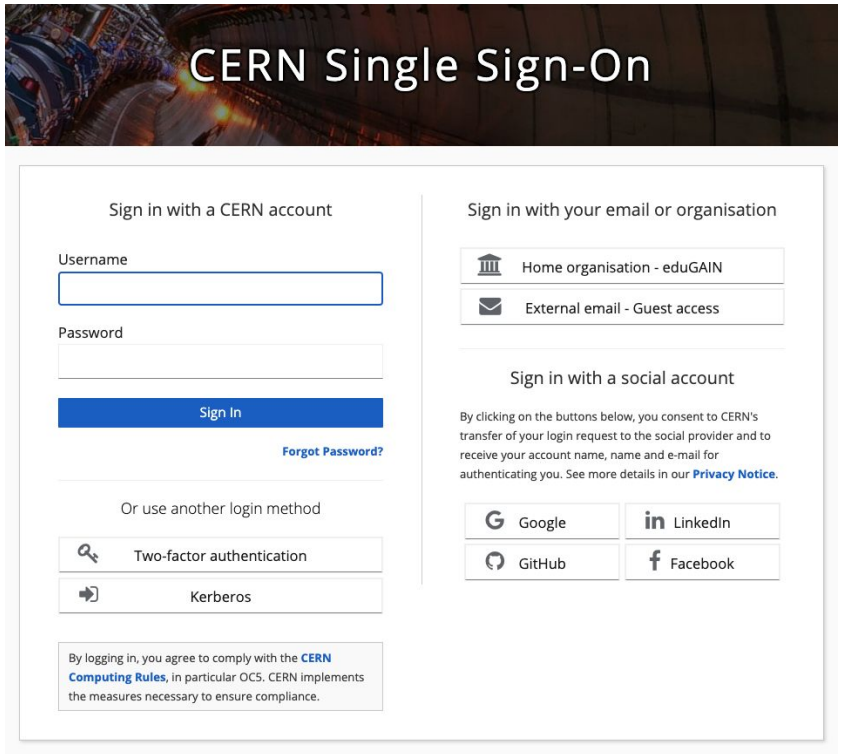
CERN Single Sign-On service

- Service overview
- Using Keycloak
- Integrations and customizations
- Challenges and limitations

Service hosting on Kubernetes

- Moving from VM-based infrastructure
- Why Kubernetes
- Current hosting architecture
- Performance and experiences
- Next steps

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, compromised password detection etc.
- Credentials are not exposed to applications

Cost / efficiency:

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

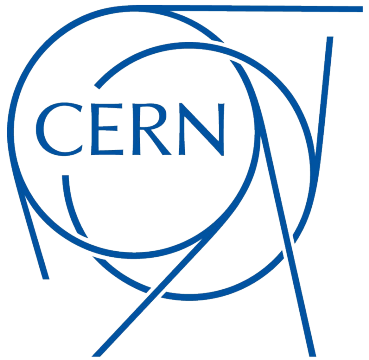
Keycloak

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 is a **CNCF incubation project** since spring 2023

Why on-prem? Why FOSS? Why Keycloak?



+

Started in late 2018 with Keycloak 4

We operate particle accelerators and experiments

- Full control over configuration, release and patching cycle
- Accessible from our internal control systems network

We value openness!

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

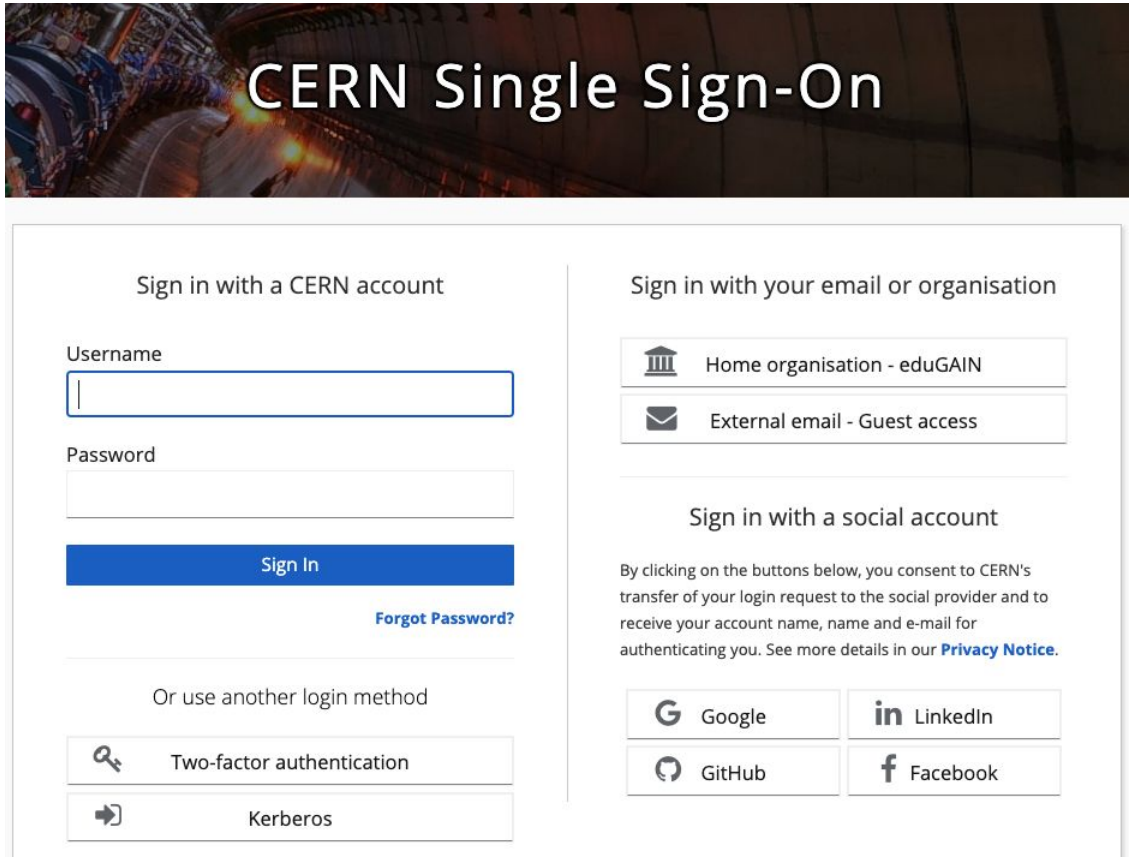
Keycloak fits our needs

- 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)

Keycloak-based SSO service at CERN

200k users (including externals)
10k clients (applications)
10k logins per hour during office hours



2FA authentication (TOTP, WebAuthn)

Kerberos authentication

eduGAIN federated identities

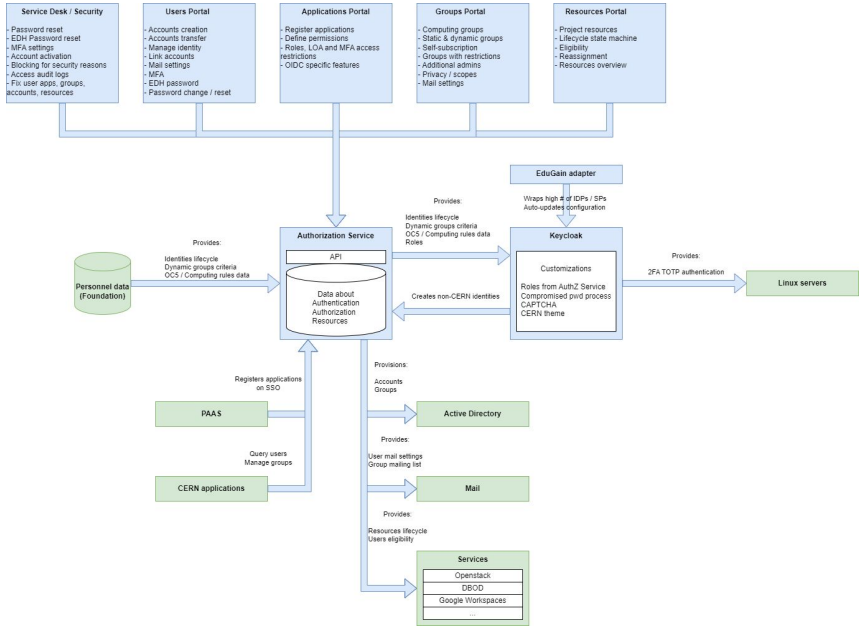
Social logins (Google, Facebook, GitHub, LinkedIn)

Guest accounts

Integration with CERN Authorization Service

CERN Authorization Service

- **separate from Keycloak-based SSO service**, but tightly integrated
- **manages identities and accounts**, **applications** and their authorization (roles, levels of assurance etc.), **groups** (80k)
- provides portals for users, service desk, admins



The decision back in 2018 was to implement this outside of Keycloak.
However, **Keycloak provides support for most of the above**.

Our CERN-specific Keycloak extensions

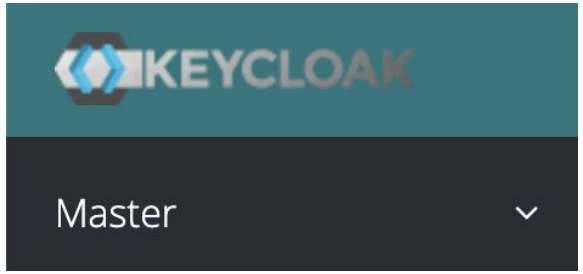
CERN Authorization Service integration *(see the previous slide)*

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

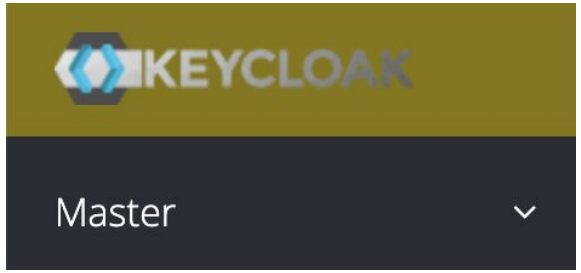
CERN theme

- CERN customisations and look & feel for user-facing login pages
- admin console: different header colors per environment:

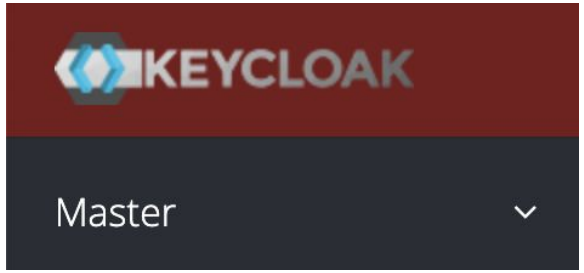
Dev



QA



Production



Our generic Keycloak extensions

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

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 the default Google reCAPTCHA (for privacy and availability reasons)

Running SSO on Keycloak: challenges and limitations

Various (minor) inconsistencies, limitations and bugs, for example:

- editing a Keycloak user blocked in AD/LDAP permanently blocks that user in Keycloak
- logs: no “username” (CODE_TO_TOKEN), username in “userId” (REFRESH_TOKEN) etc.
- admin console provides different features & details, depending on the chosen theme

Major version upgrades occasionally bring (unexpected) breaking changes

- e.g. in Keycloak 20, *“openid”* scope became mandatory in calls to UserInfo endpoint (to make it standard-compliant)

Some features stay in *preview* forever, e.g. OAuth 2.0 Token Exchange support

- 2019-2023: regular questions from users
- January 2024: plans to move it out of preview

UI-managed configuration → no versioning, no change detection

- custom solution: regular Keycloak config backups (sorted JSON exports) pushed to git

Old infrastructure

One proxy VM to serve Keycloak instances

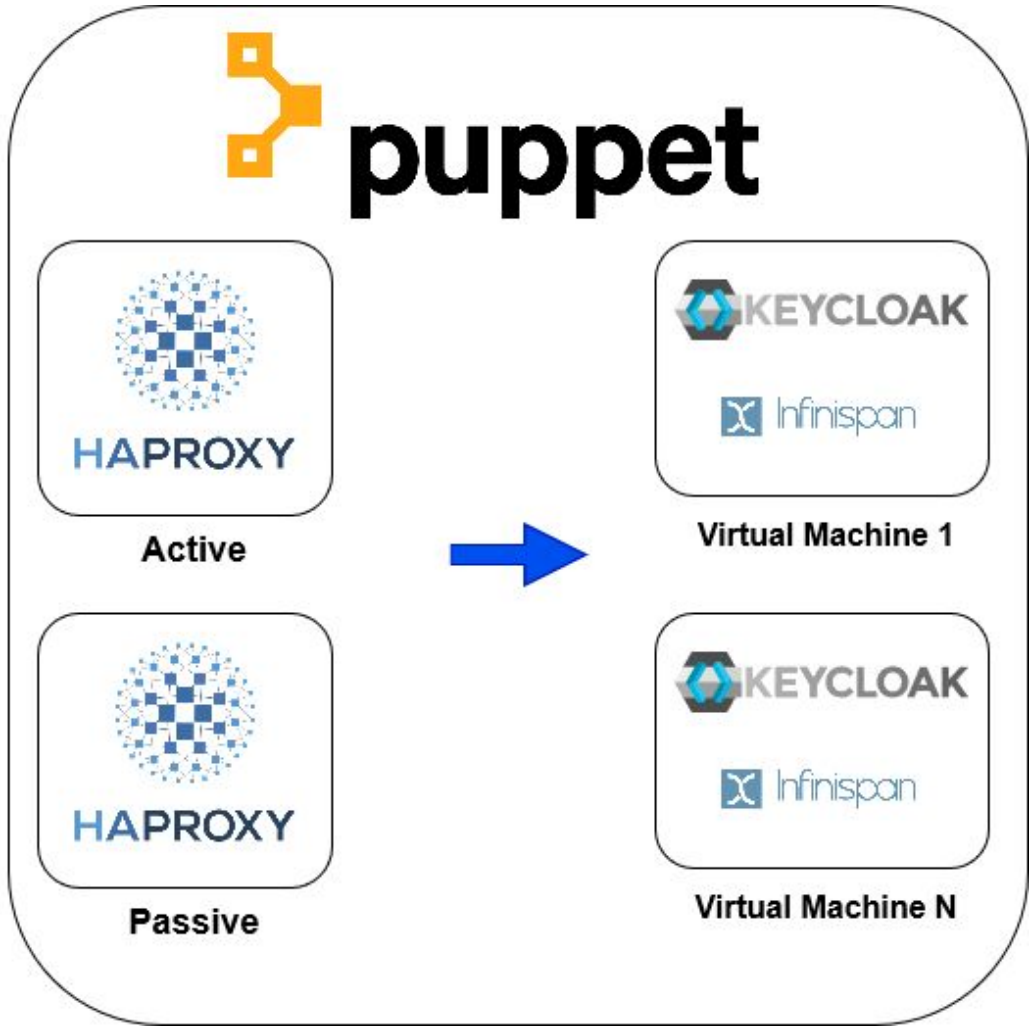
- Switch to passive could take up to 15/20 minutes

Multiple VMs running

- Keycloak and Infinispan sharing same Linux process

Puppet module

- not officially supported by Keycloak



Why Kubernetes?

Keycloak direction is clear

- Jboss replaced by Quarkus (designed for Kubernetes)
  - brings immutability to containers, faster startup, and more predictability
- Kubernetes operator for deployment

Portable

- Facilitate BC/DR

Reproducible and Immutable

- Speeds up operations, reducing team effort

Easier to maintain and deploy in long term

- Vibrant community supporting Kubernetes
- Small community in Puppet world; one main maintainer for the Puppet module

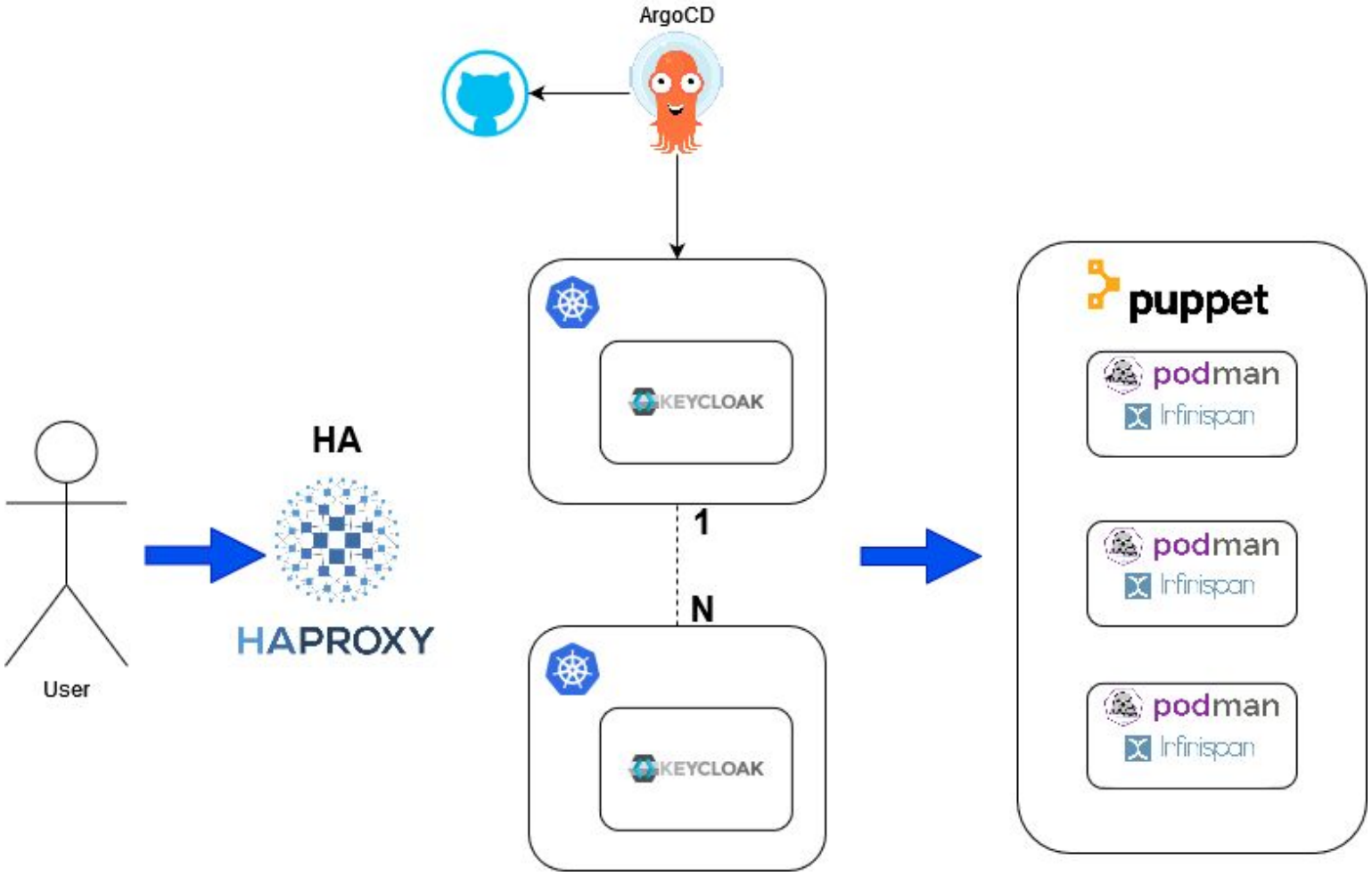
New infrastructure

Git as a source of truth

- Make operations and updates trackable and easy to rollback
- Secrets stored in CERN secret store and dynamically retrieved at deployment time

Split Keycloak and Infinispan

Kubernetes Cattle service model for Keycloak



New infrastructure

Dedicated Infinispan cluster

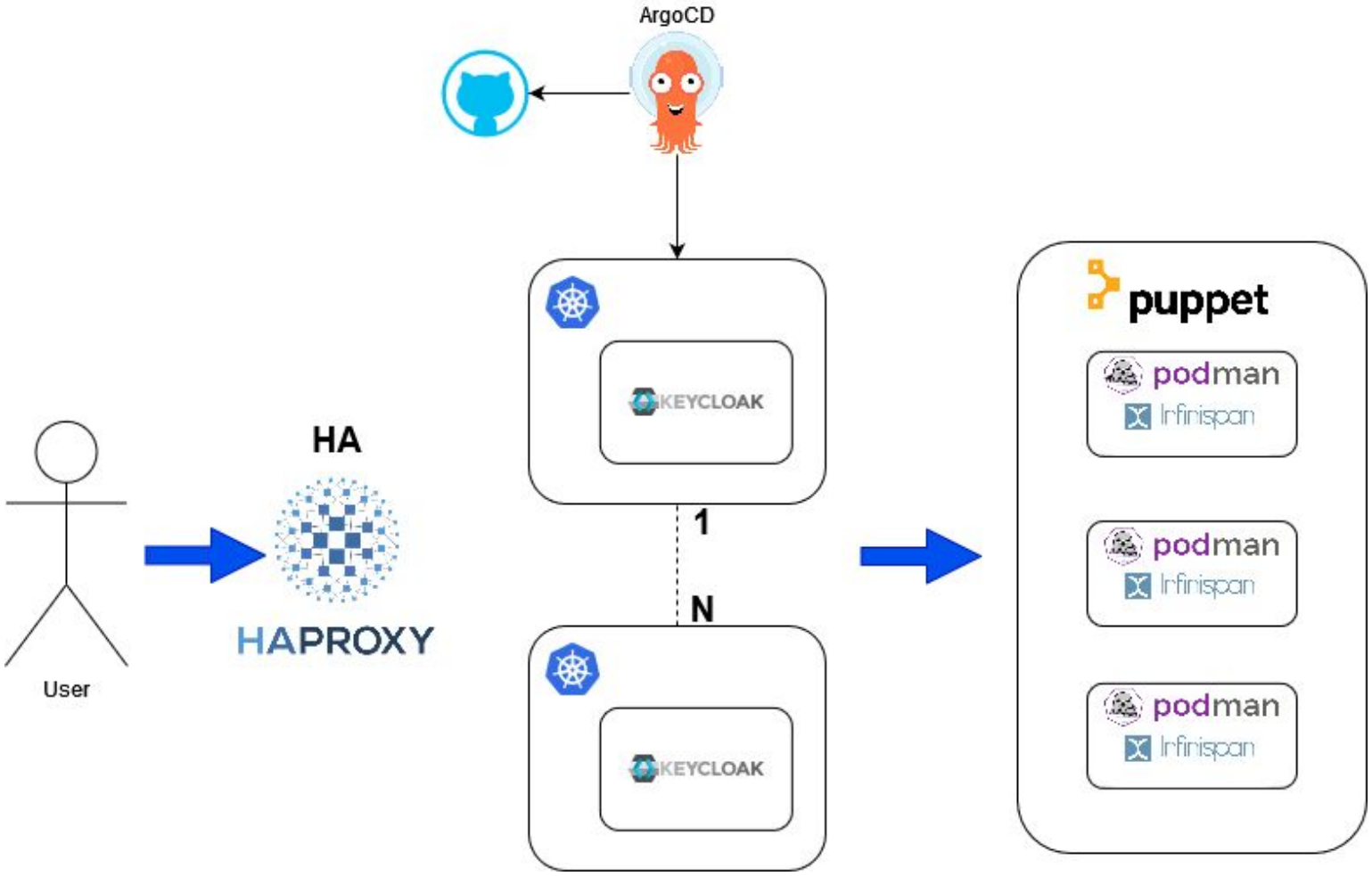
- Build with Podman + Puppet

HA HAproxy cluster

- Automatic failover with no downtime

Monitoring and Logging

- Fluent bit
- Prometheus



Stress tests

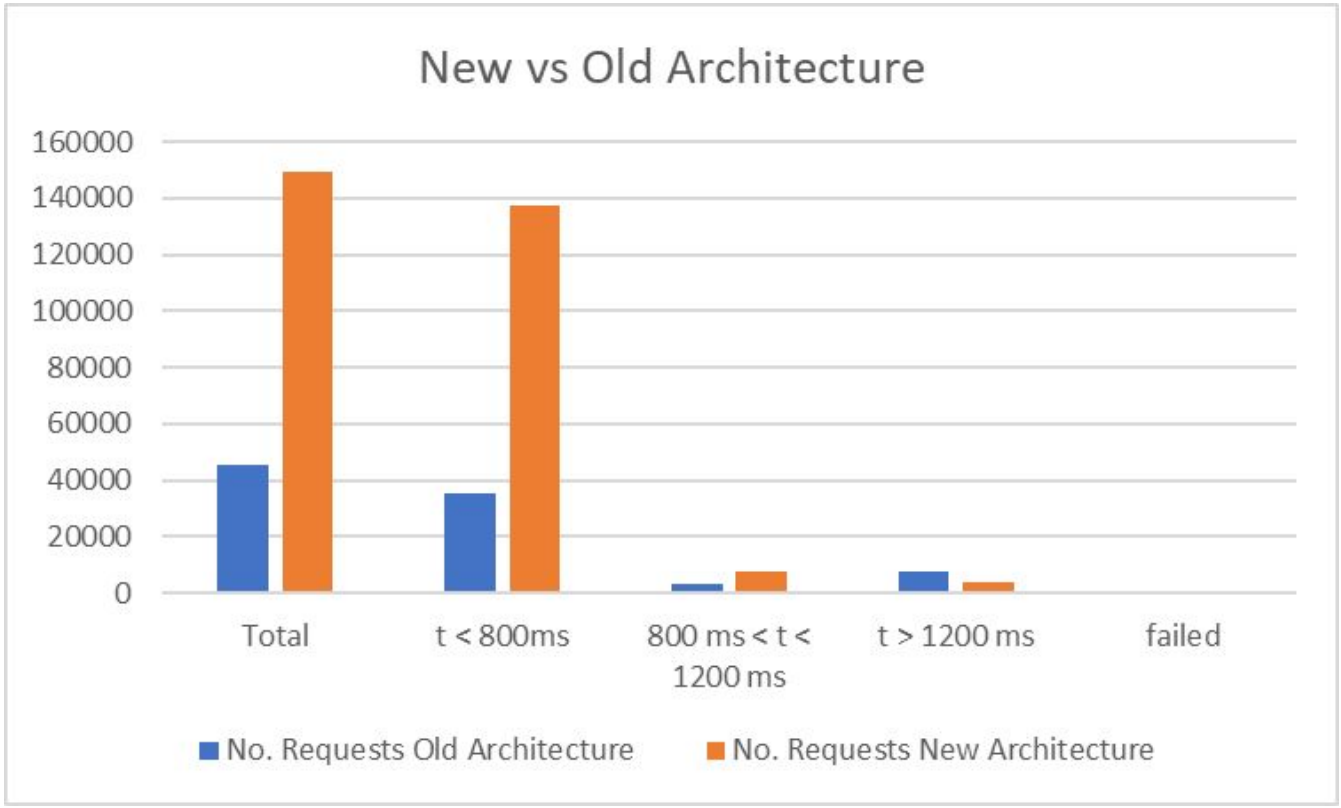
Keycloak 20.0.5

Testing infrastructure

- VMs (3 nodes)
  - 4CPU
- Kubernetes (3 pods in 2 clusters)
  - 4CPU limits

Close workload model

- Number of users executing the same scenario multiple times
- 10 minute simulation
- 50 concurrent users



Split Infinispan and Keycloak

Why

- Components can be scaled, tuned and monitored independently
- Simplify operations
- Keycloak almost (sticky sessions) stateless

How

- Create CM out of XML configuration file
  - Specifying ***remote-server***
- No official documentation(for version 20)

```
volumes:
  - name: cache-ispn
    configMap:
      name: cache-ispn
- volumeMounts:
  - name: cache-ispn
    mountPath: /opt/keycloak/conf/cache-ispn.xml
    subPath: cache-ispn.xml
additionalOptions:
- name: cache-config-file
  value: "cache-ispn.xml"
```

```
<distributed-cache name="sessions" owners="2">
  <expiration lifespan="-1"/>
  <remote-store xmlns="urn:infinispan:config:store:remote:13.0"
                cache="sessions"
                fetch-state="false"
                purge="false"
                preload="false"
                segmented="false"
                shared="true"
                raw-values="true"
                marshaller="org.keycloak.cluster.infinispan.KeycloakHotRodMarshallerFactory"
  <remote-server host="dev-infinispan.cern.ch" port="13335" />
  <security>
  <authentication server-name="infinispan">
  <plain username="username_placeholder" password="password_placeholder"/>
  </authentication>
  <encryption>
  <truststore filename="/etc/keycloak/cerntruststore" password="not_relevant" type="JKS"/>
  </encryption>
  </security>
  </remote-store>
</distributed-cache>
```

6 months of Keycloak in K8s: good things

Operations

- Faster and easier to test new:
  - feature
  - SPIs
  - Keycloak versions
- Keycloak restarts are almost invisible
  - Don’t kill user sessions
- GitOps give us a way to track and revert changes easily

More reliable

- Following all best practices in the CNCF ecosystem
- Redundant architecture

Stability and easier long term maintenance

- Keycloak Puppet module maintainer could disappear any time

6 months of Keycloak in K8s: less good things

CRD with **unsupported** field

Infinispan on VMs

- Multi K8s clusters and stateful workloads are not best friends

Is there any alternative cache to Infinispan?!

```
apiVersion: k8s.keycloak.org/v2alpha1
kind: Keycloak
metadata:
  name: example-kc
spec:
  ...
  unsupported:
    podTemplate:
      metadata:
        labels:
          my-label: "keycloak"
```

6 months of Keycloak in K8s: less good things

CRD with **unsupported** field

Infinispan on VMs

- Multi K8s clusters and stateful workloads are not best friends

Is there any alternative cache to Infinispan?!

Our plans

Define internal Keycloak upgrade policy

- Frequent releases to keep up with

05 Mar Keycloak 24.0.1 released 04 Mar Keycloak 24.0.0 released 22 Feb Keycloak 23.0.7 released

Contribute back to Keycloak

- Slowly starting (https://auth.docs.cern.ch/documents/our-contributions)

Re-assess Keycloak's Autorization Services

- Currently implemented outside of Keycloak

Prepare BC/DR plan

- Test Multi Site Setup

Investigate service mesh for Infinispan deployment on Kubernetes

Last words

We are very happy with Keycloak

- great software with a strong community behind

We are very happy with the move to Kubernetes

- mainstream, supported approach
- much more reliable infrastructure
- easy to test and deploy changes

Thank you for your attention

Our slides on Sched

KubeCon | CloudNativeCon Europe 2024