MODERN AUTHENTICATION IN SLING

WITH OPENID CONNECT AND KEYCLOAK

Dmitry Telegin - acutus.pro

Eugen Stan - netdava.com

WHY SHOULD YOU LISTEN

Software industry is growing up.

WHY SHOULD YOU LISTEN (CONTINUED)

You don’t have to build everything yourself. Use pre-existing applications to build your thing.

IN SLING TERMS

- use Sling to handle what is handle best - the content
- use whatever service or application
- integrate the services - **The focus of today's talk**

WHY SHOULD YOU LISTEN (CONTINUED)

Some people don't trust gods either :).

(source: https://baeldung.com)

AUTHENTICATION GOALS

Keep user data secured.

Be simple to use.

Challenges we face: both business and technical.

BUSINESS CHALLENGES

Derived from legislation:

- ISO/IEC 27000 - family of *Information Security Management Systems* standards
- General Data Protection Regulation - more requirements regarding user data management

BUSINESS CHALLENGES

Developer friendly information:

- ENISA Guidelines for SMEs on the security of personal data processing
- https://techblog.bozho.net/gdpr-practical-guide-developers/

WHAT IS ISO/IEC 27000?

- a set of documents that describe security practices
- they are general and target companies
- there is a certification process
- deals with both technical and organisational issues

WHAT IS GDPR?

- EU regulation that entered into force in may 2018
- establishes a set of rules for companies
- focuses on user data protection
- has a high impact on business processes

BUSINESS CHALLENGES

Derived from ISO/IEC 27 000 and GDPR:

- Forget me - functionality
- Allow users to edit their profile - OOTB with Keyclaok
- Consent check boxes - login policy !?
- Re-request consent - login policy !?

BUSINESS CHALLENGES

Derived from ISO/IEC 27 000 and GDPR requirements:

- Implement pseudo-nimisation - usernames and user id's
- Log access to personal data - Keycloak login audit
- Register API consumers - Keycloak has support for UMA

BUSINESS CHALLENGES

- Access control policy
- Access control and authentication
- Logging and monitoring

TECHNICAL CHALLENGES

TECHNICAL CHALLENGES

Multi-factor authentication

Reset credentials

Captcha

LDAP

Single sign-on

Social login

Terms and Conditions

Email verification

Federation

Identity brokering

Brute-force protection

Password policies

SLING

Multi-factor authentication

Reset credentials Captcha LDAP

Single sign-on Social login

Terms and Conditions Email verification

Federation Identity brokering

Brute-force protection

Password policies

ALTERNATIVE:

USE AN EXTERNAL IAM

ALTERNATIVE:

USE AN EXTERNAL IAM

Deploy

ALTERNATIVE:

USE AN EXTERNAL IAM

Deploy → Integrate

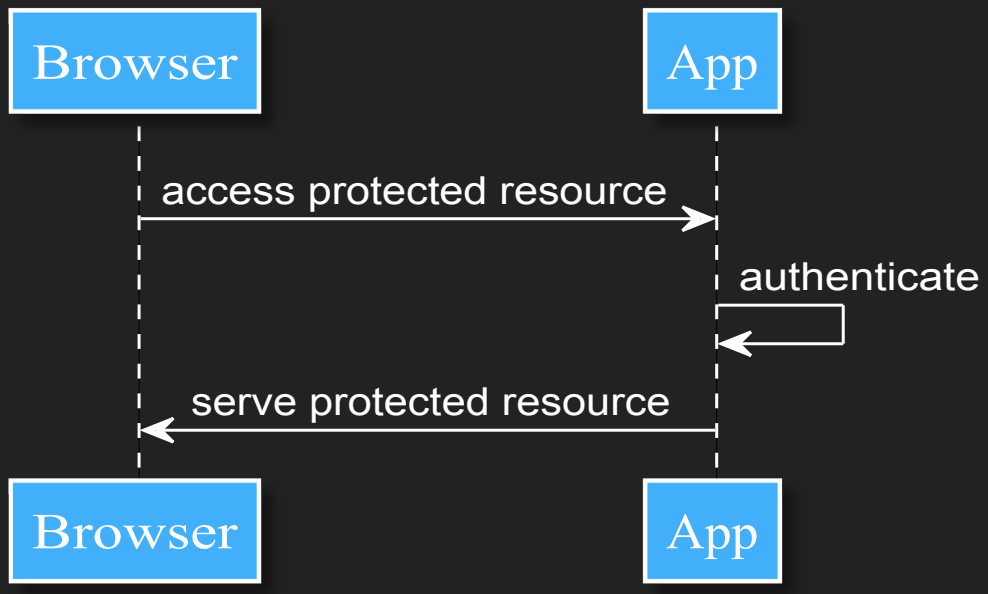
ALTERNATIVE:

USE AN EXTERNAL IAM

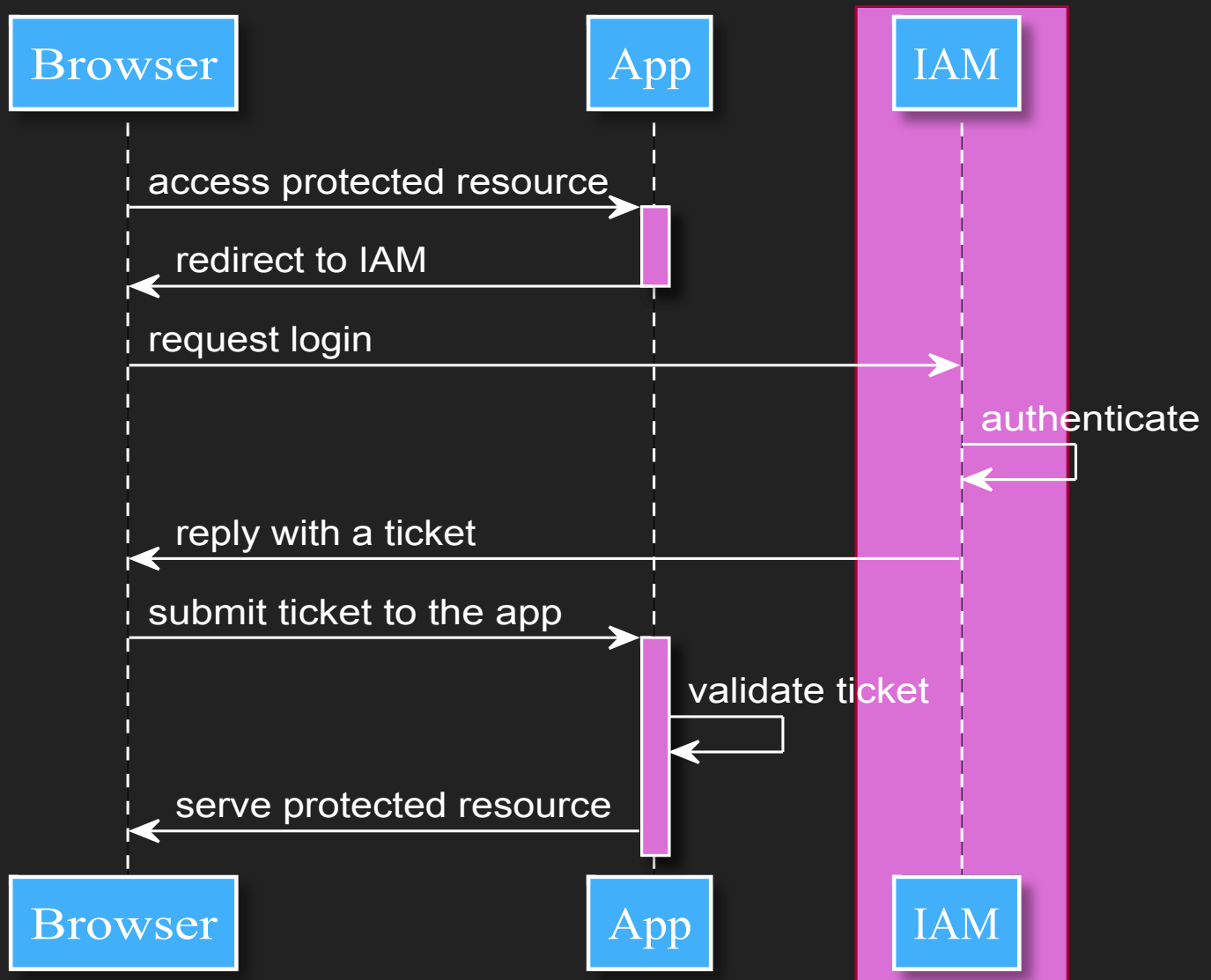
Deploy → Integrate → Delegate

AUTHENTICATION

AUTHENTICATION WITHOUT IAM



AUTHENTICATION WITH IAM: HOW ITH WORKS?



LIVE DEMO

https://adapt-to-demo.netdava.com/

WHAT WAS THAT?

WHAT WAS THAT?

WHAT WAS THAT?

WHAT WAS THAT?

IAM SOLUTIONS

WHY KEYCLOAK?

WHY KEYCLOAK?

Multi-factor authentication

Reset credentials Captcha LDAP

Single sign-on Social login

Terms and Conditions Email verification

Federation Identity brokering

Brute-force protection

Password policies

WHY KEYCLOAK?

Multi-factor authentication

Reset credentials Captcha LDAP

Single sign-on Social login

Terms and Conditions Email verification

Federation Identity brokering

Brute-force protection

Password policies

...and it's free

WHY KEYCLOAK?

Multi-factor authentication

Reset credentials

Captcha

LDAP

Single sign-on

Social login

Terms and Conditions

Email verification

Federation

Identity brokering

Brute-force protection

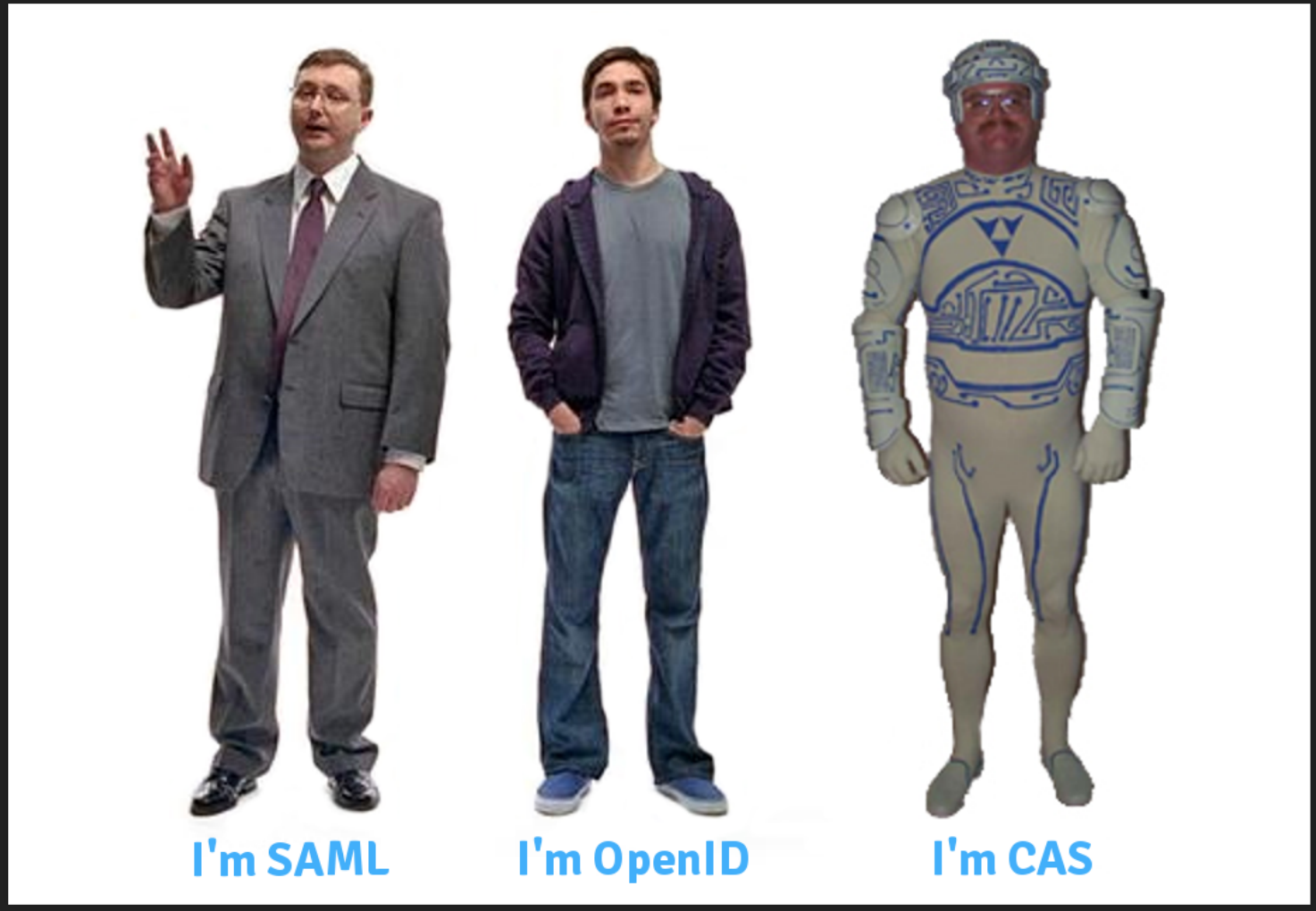
Password policies

...and it's free

redhat.

SSO PROTOCOLS

SSO PROTOCOLS



SSO PROTOCOLS: SAML

- Born in the era of XML WS
- Still very popular in the enterprise
- Mature and feature-rich
- XML-based
- Very verbose
- No backchannel required
- Perfect for monolithic webapps

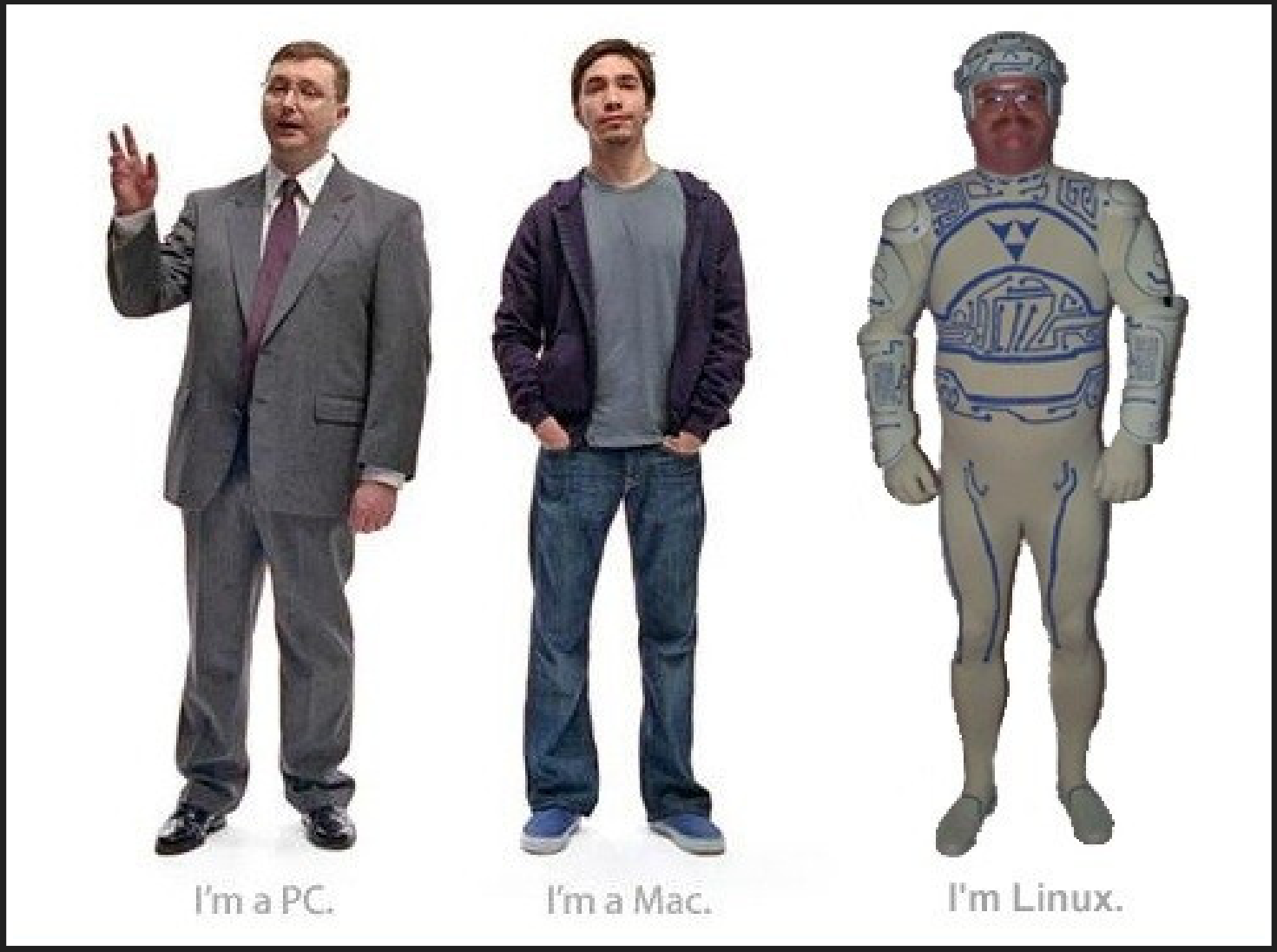
SSO PROTOCOLS: OPENID CONNECT

- First-class web citizen
- JSON-based
- Lightweight
- Optional backchannel
- Perfect for all kinds of apps

SSO PROTOCOLS: CAS

- Implemented by Apereo CAS
- JSON- or XML-based
- Backchannel required
- Suitable for all kinds of apps

CREDITS



I WANT IT NOW, AND I WANT IT NOW

- Keycloak (4.4.0+)
- Keycloak servlet filter adapter
- Sling Keycloak AuthenticationHandler
- (optional) Oak pre-authenticating JAAS module
- (optional) Oak user provisioning

KEYCLOAK SERVLET FILTER

```
@Component(
    immediate = true,
    service = Filter.class,
    property = {
      CONFIG_FILE_PARAM + "=" + "keycloak.json",
      SKIP_PATTERN_PARAM + "=" + "/public/.*",
      HTTP_WHITEBOARD_FILTER_PATTERN + "=" + "/",
      HTTP_WHITEBOARD_CONTEXT_SELECT + "="
    + "(osgi.http.whiteboard.context.name=org.apache.sling)"
    })
public class KeycloakFilter extends KeycloakOIDCFilter { }
```

KEYCLOAK SLING AUTHENTICATIONHANDLER

```
public class KeycloakAuthenticationHandler implements
 org.apache.sling.auth.core.spi.AuthenticationHandler {
        // ....
public AuthenticationInfo extractCredentials(
     HttpServletRequest req, HttpServletResponse res) {
   KeycloakSecurityContext ctx = (KeycloakSecurityContext)
           req.getSession()
.getAttribute("org.keycloak.KeycloakSecurityContext");
   return new AuthenticationInfo("KEYCLOAK",
              "admin", "admin".toCharArray());
 }
// ....
}
```

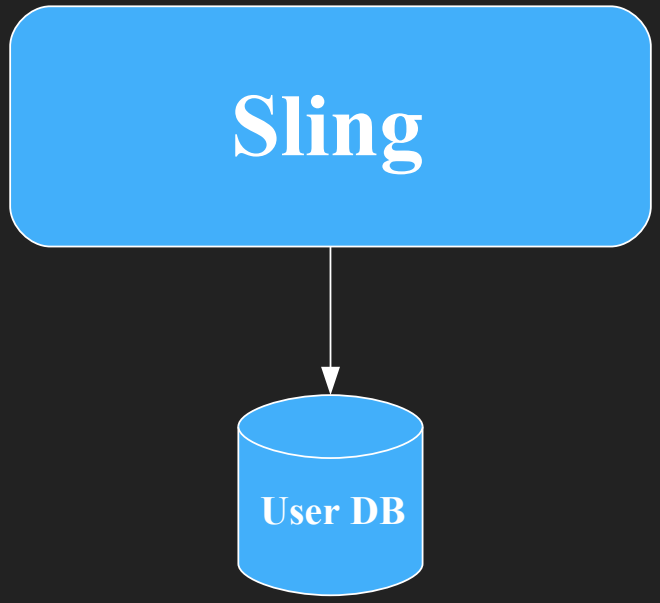
CAN I HACK ON IT?

(Tell about GitHub repos and Docker images; propose improvised hands-on if there is interest)

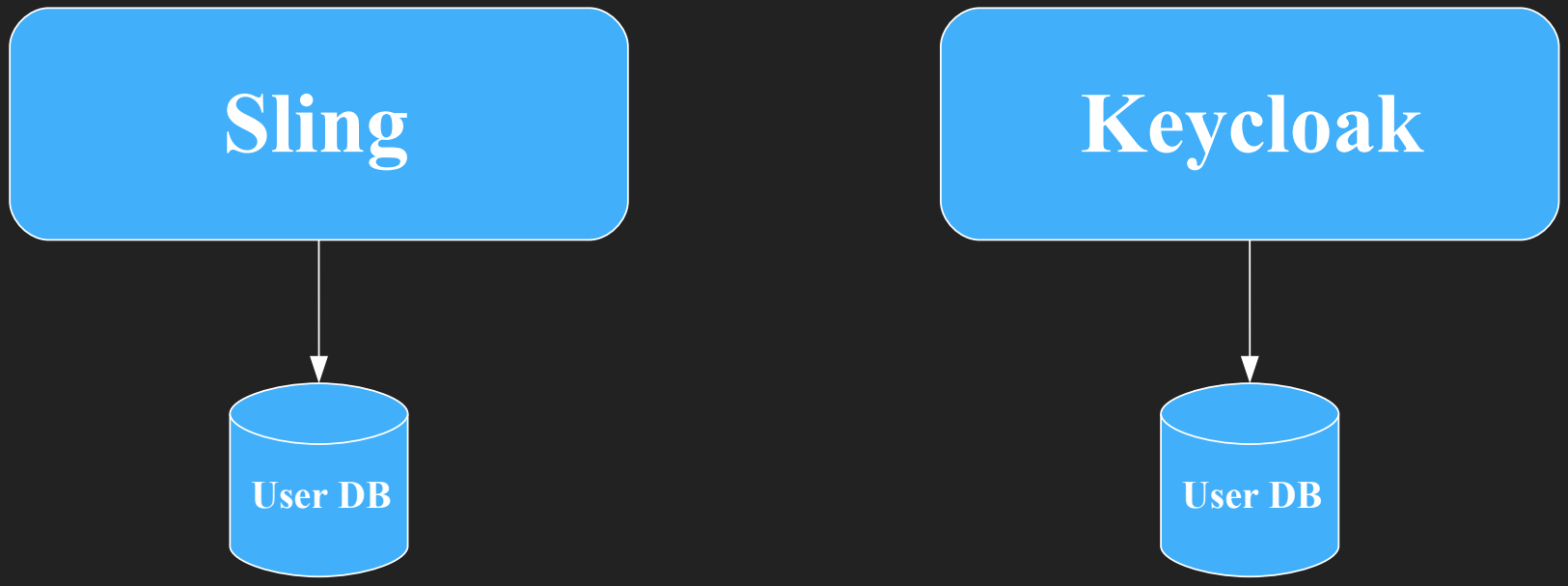
- Demo code: https://github.com/netdava/adapt-to-2018-keycloak-sling-presentation
- https://github.com/netdava/sling-org-apache-sling-karaf-features
- https://github.com/keycloak/keycloak/

IDENTITY PROVISIONING

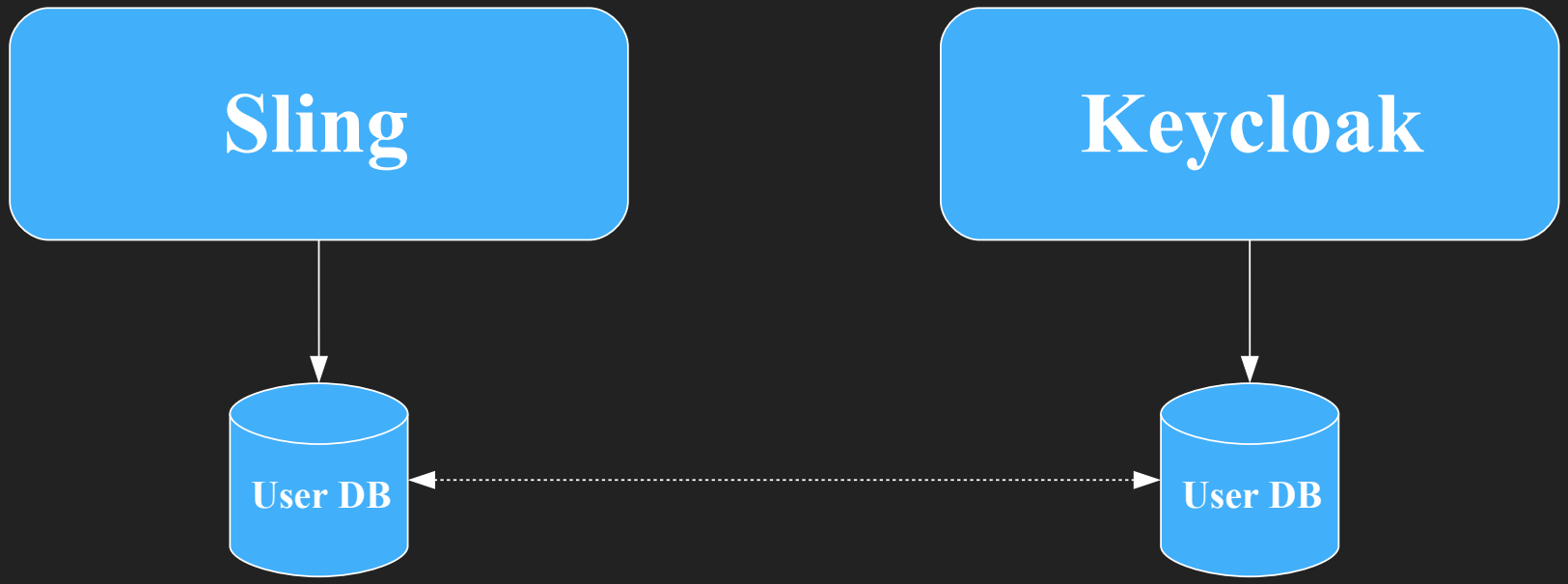
USER PROVISIONING? WHAT IS IT?



USER PROVISIONING? WHAT IS IT?



USER PROVISIONING? WHAT IS IT?



DO I REALLY NEED IT?

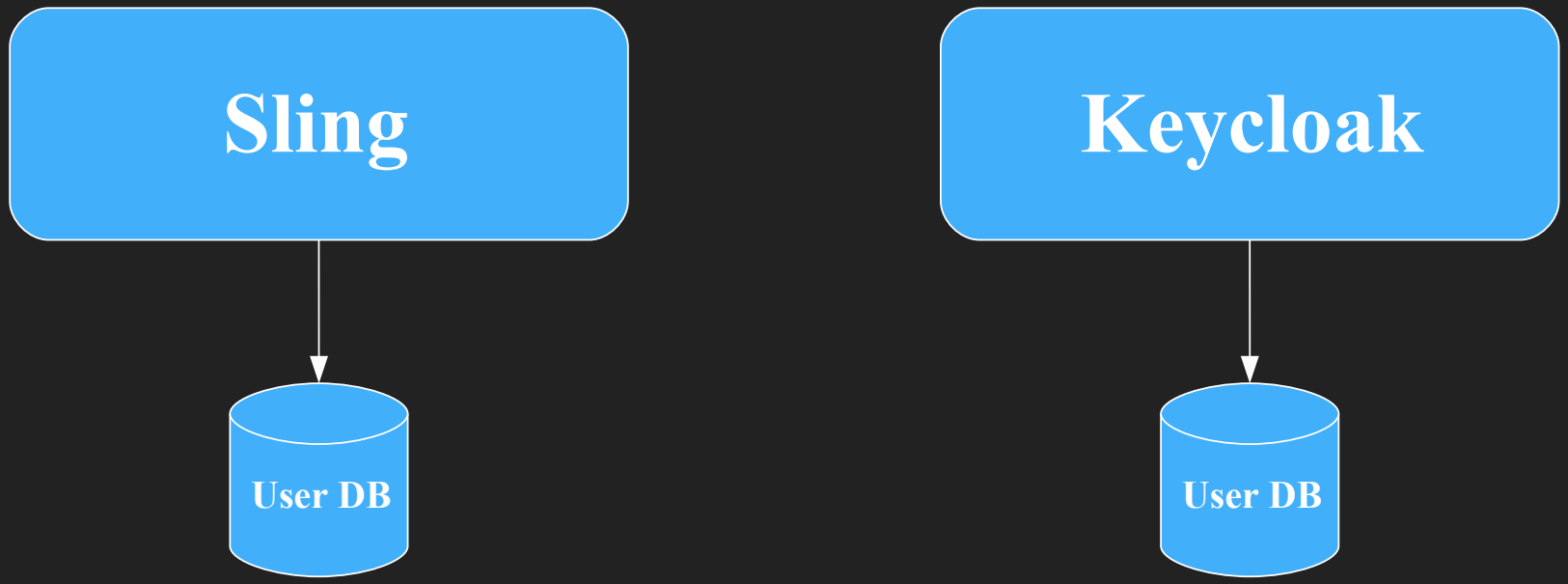
YES:

- You're relying on the users' existence in the underlying JCR;
- If you make heavy use of JCR ACLs.

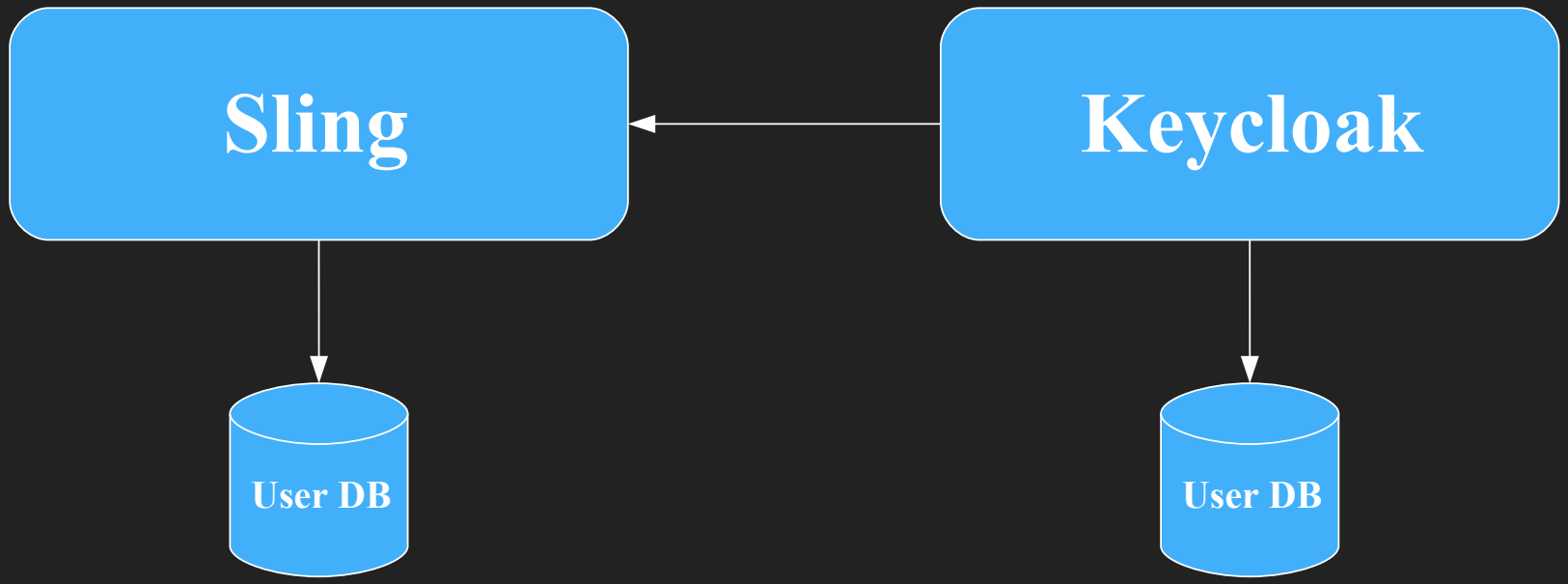
NO:

- You connect the underlying DB using a service account;
- Access control is on the application level
- You're OK to go with a single account for JCR

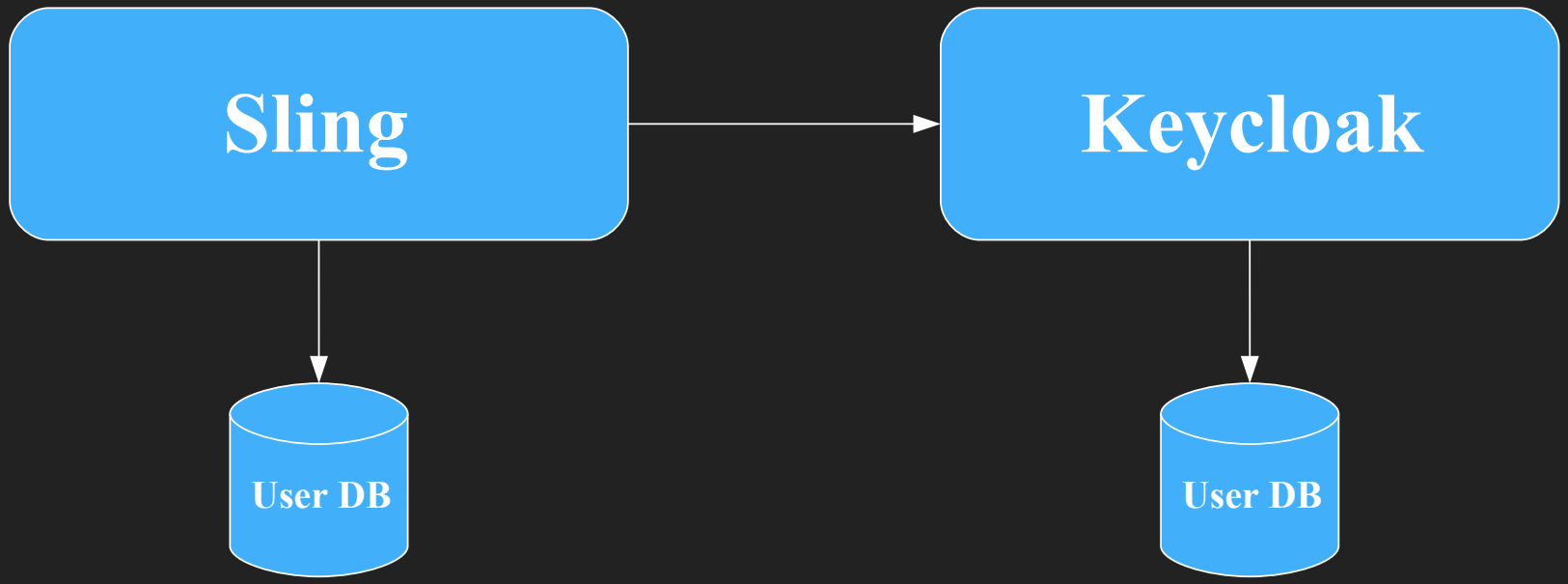
USER PROVISIONING: APPROACHES



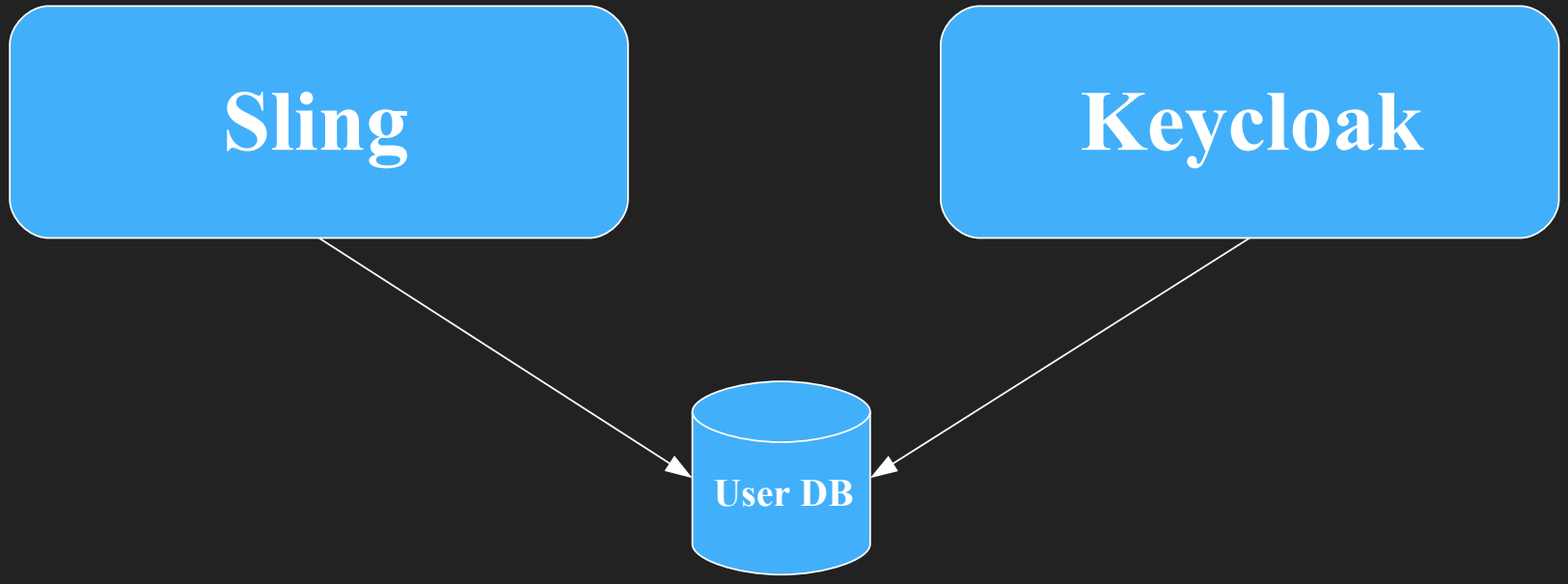
USER PROVISIONING: PUSH



USER PROVISIONING: PULL

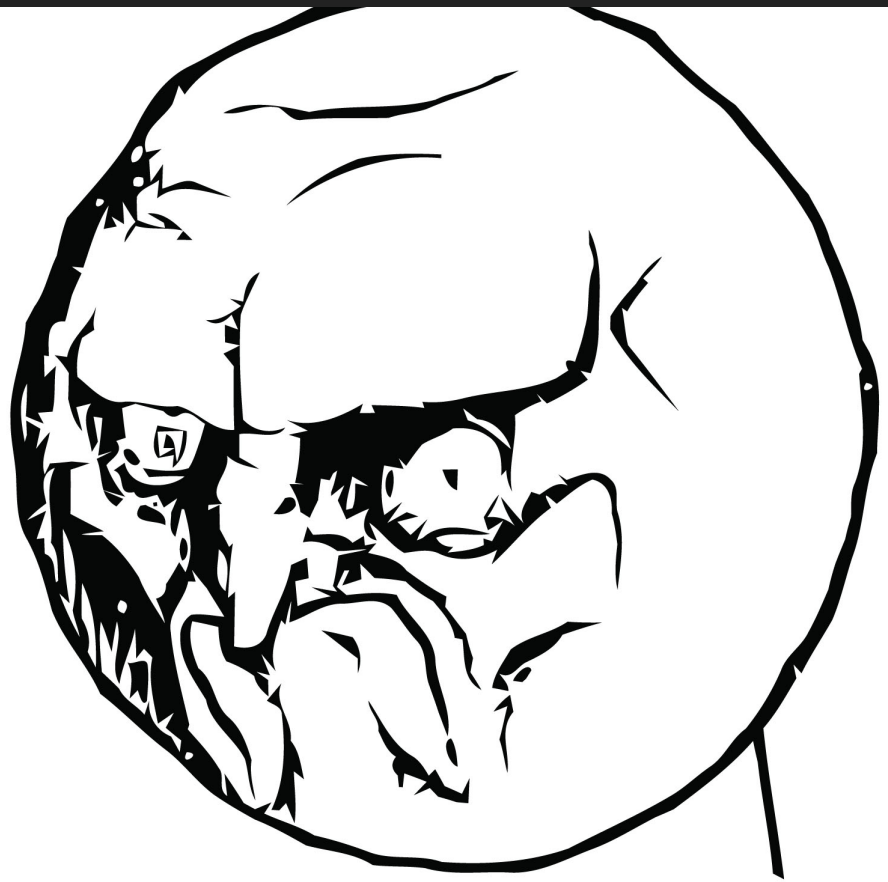


USER PROVISIONING: SHARED STORE



CAN I HAVE PUSH/PULL NOW?

CAN I HAVE PUSH/PULL NOW?



NO.

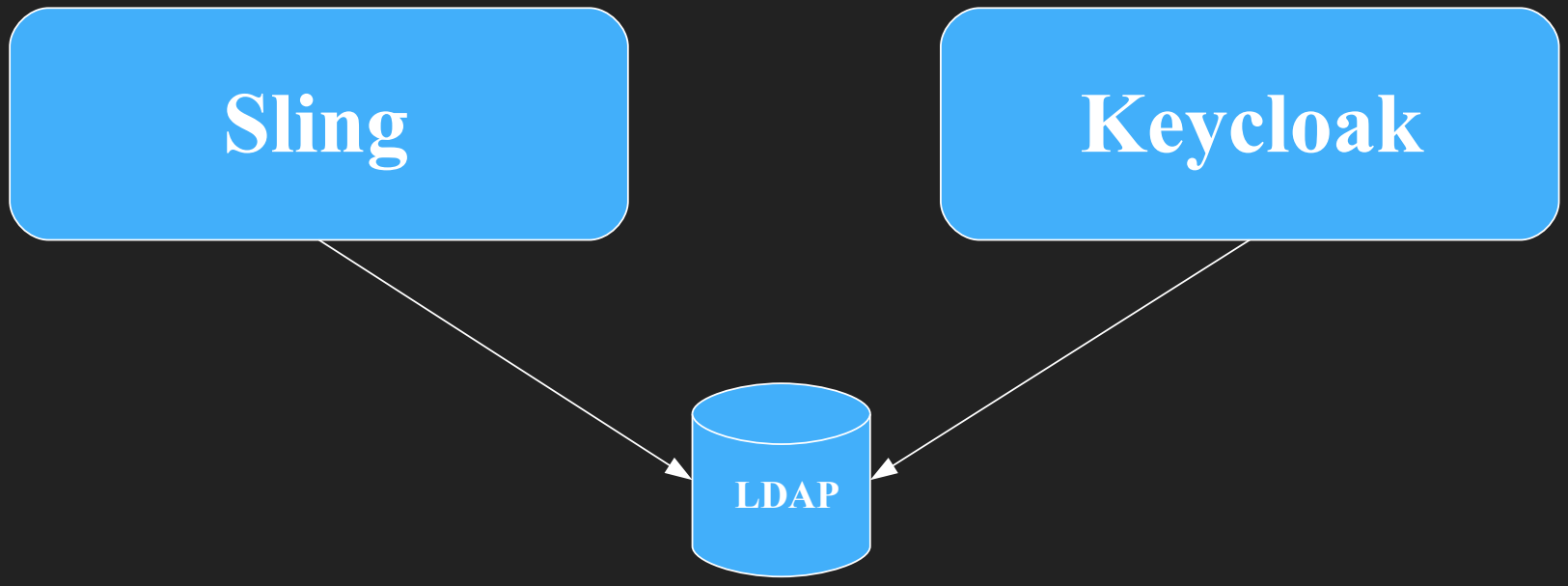
SCIM TO THE RESCUE (FUTURE)

http://www.simplecloud.info

SCIM TO THE RESCUE (FUTURE)

- Open standard for identity data exchange
- Push and pull provisioning in a vendor-neutral way
- JSON-based, lightweight and extensible
- Opensource Java SDK by PingIdentity
- Coming soon to Keycloak

LDAP TO THE RESCUE (PRESENT)



AUTHORIZATION

OPTIONS FOR AUTHORIZATION

- Oak side: JCR ACLs
- OSGi side: OSGi HttpService servlet filter
- Sling side: Resource access security
- Keycloak: keycloak authorization services

ACCESS CONTROL IN OAK (JCR ACL'S)

- Paradigm: declarative
- Character: repository-centric

ACCESS CONTROL IN OAK - BENEFITS

- well-known to Sling developers ?!
- requires no coding
- “live” inside the repository (can be migrated, backed up & restored with the repo)

ACCESS CONTROL IN OAK - DRAWBACKS

- need user provisioning
- have limitations inherent to ACL systems
- need to design content around ACL

ACCESS CONTROL IN SLING

- Paradigm: imperative
- Character: application-centric

ACCESS CONTROL IN SLING - BENEFITS

- can implement any policy you like
- also “live” inside the repo
- no user provisioning required

ACCESS CONTROL IN SLING - DRAWBACKS

- requires coding
- you reinvent the wheel

ACCESS CONTROL WITH KEYCLOAK

- Paradigm: declarative + imperative
- Character: identity-centric

ACCESS CONTROL IN KEYCLOAK - BENEFITS

- an extensive set of built-in policies:
  - RBAC, ABAC, time-based etc.
- create completely bespoke policies (may require coding)
- in all other cases, no coding required
- no user provisioning required

ACCESS CONTROL IN KEYCLOAK - DRAWBACKS

- unfamiliar for most Sling developers
- policies “live” inside Keycloak -> another system to manage

POLICIES IN KEYCLOAK

Allow content readers access to /**secured-content**
(Role based policy)

POLICIES IN KEYCLOAK

Allow access to **/content/for/adapt-to** for users with **@adapt.to** email

(Attribute based access)

POLICIES IN KEYCLOAK

If today is Monday and the wheather in Potsdam is sunny and over 30 degrees then system is closed so employees can have fun.

(Go wild: Time based + custom access policy)

LIVE DEMO

FUTURE DIRECTIONS

- Use OSGi Config Admin Service for multi-tenancy
- Integrate Keycloak authorization and Oak queries
- SCIM support in both Sling and Keycloak

ABOUT US

- it's the first time we worked together
- we met on the Sling mailing list
- we decided to keep the presentation together
- common interests: Sling and Keycloak

ABOUT US - DMITRY TELEGIN

- CTO of Acutus (https://acutus.pro)
- freelance architect & IAM consultant
- Mageia Linux contributor
- agita.to - coming soon

ABOUT US - EUGEN STAN

- passionate about business and technology
- freelance IT architect / consultant
- I like to build systems - especially SaaS
- Open source contributor: Apache Software Foundation, Debian, etc
- Interests: Kubernetes, containers, Java

Q & A

- Demo: https://adapt-to-demo.netdava.com/

THANK YOU !