Securing Microservices with MicroProfile & Keycloak

In 40 minutes or less!

Fabio Turizo

Speaking Today

Fabio Turizo

Service Manager

Bluesky: @fturizo.bsky.social

LinkedIn: www.linkedin.com/in/fturizo/

Give me Feedback!

Scan the QR code to share comments about the talk!

Payara Services Helps Shape the Future of the Industry

- Contributor Members of the Eclipse Foundation
- Strategic Members of the Jakarta EE Working Group
- Members of the MicroProfile Working Group
- Project Management Committee member of Jakarta EE

Payara Platform Enterprise

Payara Server Enterprise
Robust. Reliable. Supported.

The best application platform for production Jakarta EE apps.

Payara Micro Enterprise
Small. Simple. Serious.

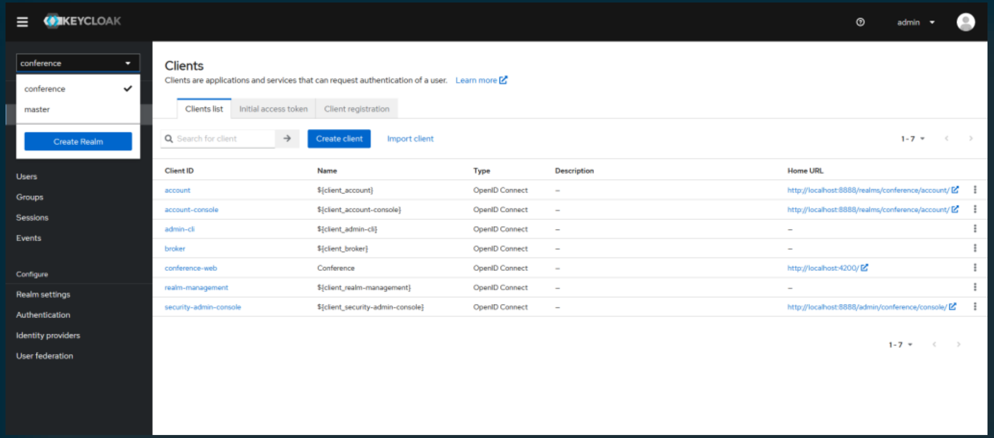
The platform of choice for containerized Jakarta EE microservices deployments.

Security is Hard!

- How do I “correctly”:
  - Safely manage user identities?
  - Apply security patterns without re-inventing the wheel?
  - Secure my micro-services with minimal risks?
  - Reduce performance impact on HA environments?

Why Keycloak?

- Security Patterns on premises
  - Centralized User management
  - OpenID Connect Compatible
  - Multiple realms, IdP integration
- Seamless SSO Integration
- Maintenance is on you!



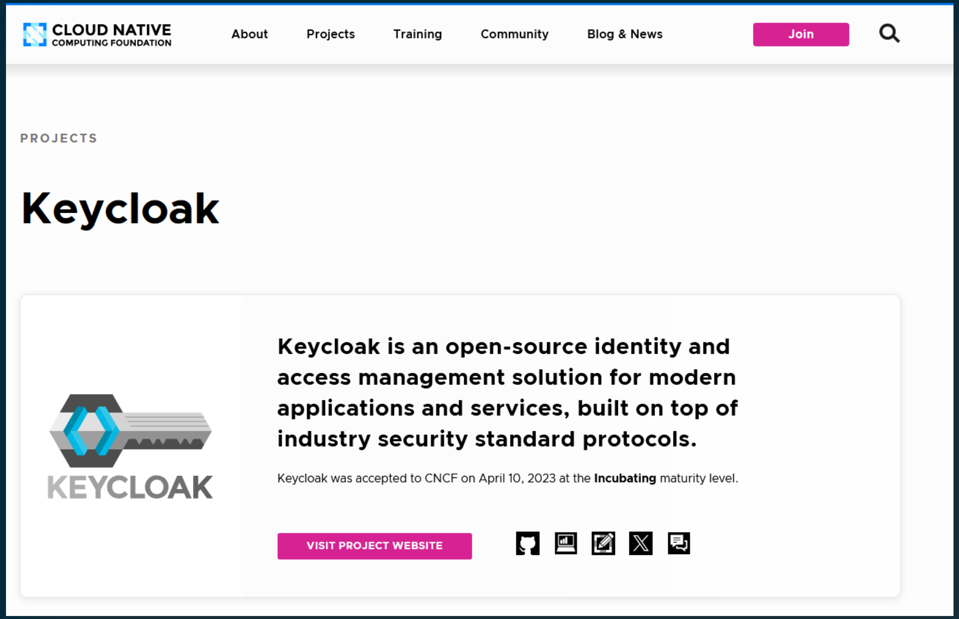
Clients

Clients are applications and services that can request authentication of a user. Learn more

Client ID	Name	Type	Description	Home URL
account	${client_account}	OpenID Connect	–	http://localhost:8888/realms/conference/account/
account-console	${client_account-console}	OpenID Connect	–	http://localhost:8888/realms/conference/account/
admin-cli	${client_admin-cli}	OpenID Connect	–	–
broker	${client_broker}	OpenID Connect	–	–
conference-web	Conference	OpenID Connect	–	http://localhost:4200/
realm-management	${client_realm-management}	OpenID Connect	–	–
security-admin-console	${client_security-admin-console}	OpenID Connect	–	http://localhost:8888/admin/conference/console/

Incubating in the CNCF

- A CNCF project
- Currently in incubating stage
- Oriented towards highly scalable environments
- Fully open-source



PROJECTS

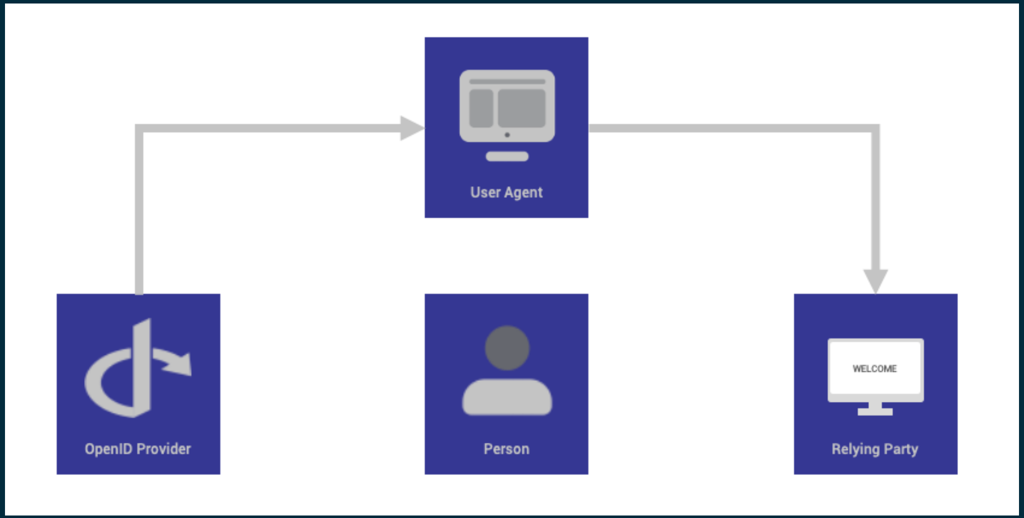
Keycloak

Keycloak is an open-source identity and access management solution for modern applications and services, built on top of industry security standard protocols.

Keycloak was accepted to CNCF on April 10, 2023 at the **Incubating** maturity level.

How OpenID Connect Works

1. End user **navigates to a website or web application** via a browser.
2. End user **clicks sign-in** and types their username and password.
3. The RP (Client) **sends a request** to the OpenID Provider (OP).
4. The OP **authenticates the User** and obtains authorization.
5. The OP **responds with an Identity Token** and usually an **Access Token**.
6. The RP can **send a request** with the Access Token to the User device.
7. The **UserInfo** Endpoint **returns Claims** about the End-User.



Securing Microservices

- Microservices (usually) rely on stateless communication
- Effective security mechanisms should:
  - Validate each request separately
  - **Security context** propagation in one span(*)
- **JWT** to the Rescue!

JSON Web Tokens

- Token-based Authentication and Authorization
- **OpenID Connect** Compatible
- JWT → RFC7519
- Relies on Standard “Claims”

Open ID Connect

- Interoperable Auth protocol
- Ideal for REST/JSON services
- Built on top of OAuth2
- Easy to use message flows

Securing Microservices

- Microservices rely on stateless communication
- Effective security mechanisms should:
  - Validate each request separately
  - **Security context** propagation in one span(*)
- **JWT** to the Rescue!

Eclipse MicroProfile

- Initially conceived for Microservices
- Cloud-Oriented
- Jakarta EE aligned
- Part of the Eclipse Foundation

  http://microprofile.io/

MicroProfile JWT (2.1)

- Role based Access Control (**RBAC**)
- Caller's identity is **introspected**
- Identity validation is **delegated**
- Access to resources is **verified**
- Flexible claim validation

Demo Time

https://github.com/fturizo/MicroServicesKeycloakDemo

Demo: Conference Event Management

- 2 microservice applications:
  - `Speaker`: Manage the speakers that will talk at the event.
  - `Session`: Manage the session talks held at the event.
- SPA web application that uses the previous microservices

Demo: Groups & Roles

- Admin
  - Create new session talks (can-create-sessions)
  - View all session talks (can-see-sessions)
  - Delete session talks (can-delete-sessions)
  - View all registered speakers (can-see-speakers)
  - Add new speakers (can-add-speakers)
  - Accept speakers into the conference (accept-speakers)

Demo: Groups & Roles

- Speaker
  - View registered sessions (can-see-sessions)
  - View all fellow speakers (can-see-speakers)
  - Register themselves as speakers (speaker, can-add-speakers)

Demo: Groups & Roles

- Attendee
  - View all speakers registered (can-see-speakers)
  - View all session talks (can-see-sessions)
  - Attend a session if interested (can-register-to-sessions)

In Summary…

- Keycloak will allow you to:
  - Delegate user management and authentication
  - Simplify authentication workflow in applications
  - Remove security challenges from micro-services
  - Centralize user management, maintenance and security

… And

- Eclipse MicroProfile (JWT) will allow you to:
  - Plugin an existing Keycloak realm via config properties
  - Not worry about token validation algorithms
  - Focus on setting role constraints to existing services.

Give me Feedback (again)!

Scan the QR code to share comments about the talk!

Thank You