Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

ICSC
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Effortless Identity Management

Setting Up Authentication, Authorization and SSO with Keycloak

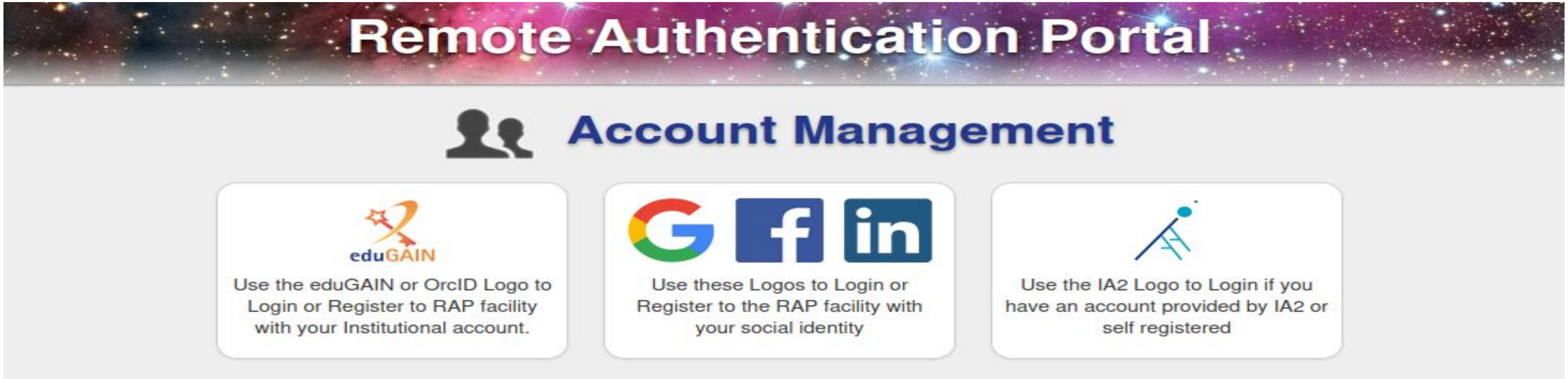
Massimo Costantini

INAF - OATs

Archives and Data Management Systems, Bologna Feb 26-27, 2025

RAP & GMS

- The RAP (Remote Authentication Portal) and GMS (Group Membership Service) were developed by Franco Tinarelli in collaboration with the IA2 group in Trieste several years ago.
- They were written in PHP and Java and are currently the central authentication point for INAF (IA2 and Rosetta), integrating also eduGAIN and social authentication providers.
- They also serve as the entry point for all our portals in Trieste, ensuring a unified authentication across different services.



Remote Authentication Portal

Account Management

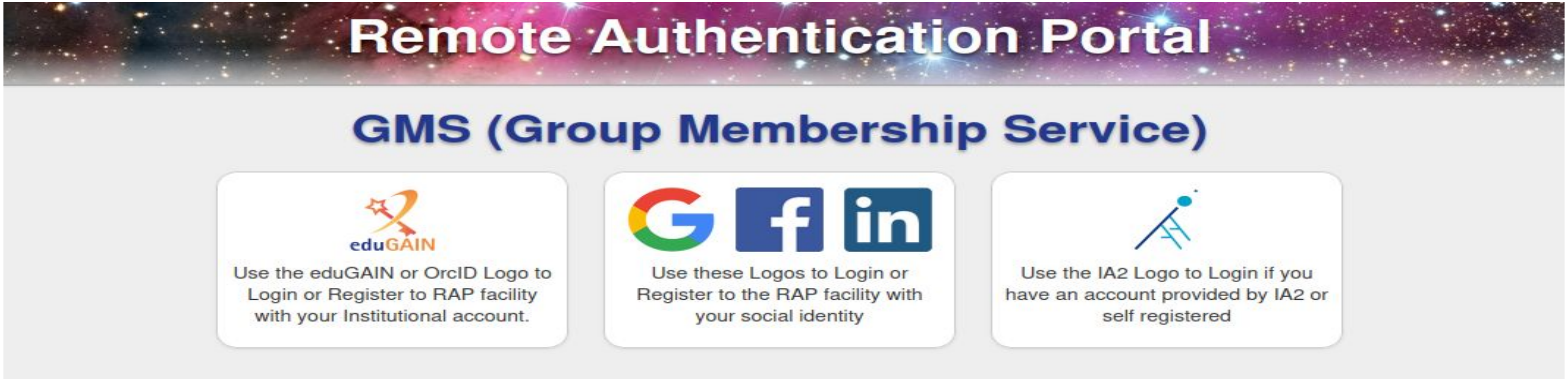
Use the eduGAIN or OrcID Logo to Login or Register to RAP facility with your Institutional account.

Use these Logos to Login or Register to the RAP facility with your social identity

Use the IA2 Logo to Login if you have an account provided by IA2 or self registered

RAP & GMS

- Both products were developed and refactored in-house by some colleagues who left INAF several years ago.
- Over the years, the products have become vulnerable to updates, making it difficult to remain compliant with modern security standards.
- So, what should we do next? Update the existing code? Rewrite everything from scratch? Or use a reliable open-source solution?



Remote Authentication Portal

GMS (Group Membership Service)

Use the eduGAIN or OrcID Logo to Login or Register to RAP facility with your Institutional account.

Use these Logos to Login or Register to the RAP facility with your social identity

Use the IA2 Logo to Login if you have an account provided by IA2 or self registered

INDIGO IAM

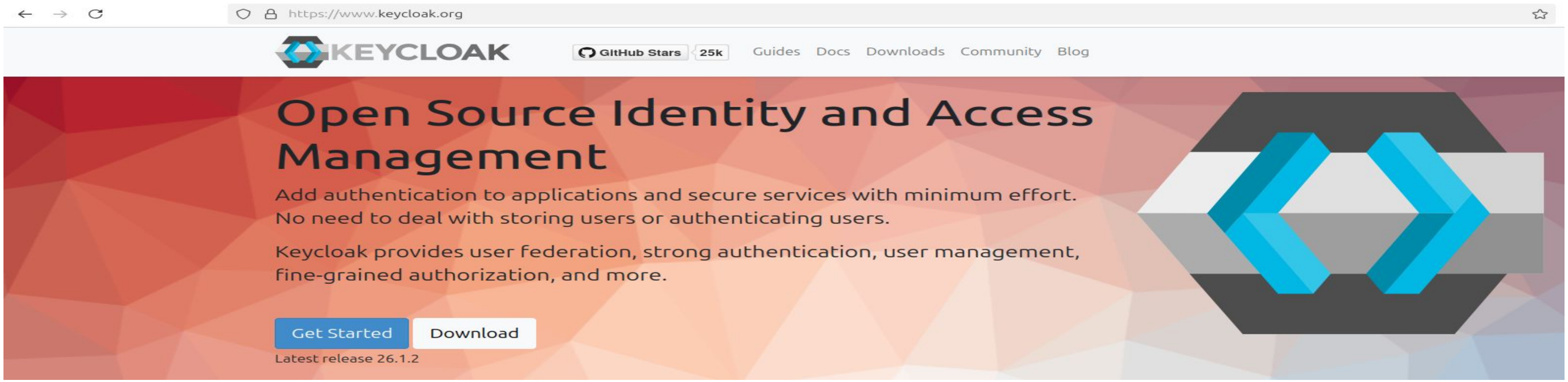
- INDIGO Identity and Access Management (IAM) is an open-source solution designed to manage primarily Authentication and Authorization for scientific and research infrastructures.
- Developed as part of the INDIGO-DataCloud project by INFN, INDIGO IAM is widely used in research environments to enable secure and federated access to distributed computing resources.
- Compatible with OpenID Connect, OAuth2, SAML, X.509, natively supporting EduGAIN (and used, for example, by CERN for Identity and Authorization Management in federated scientific infrastructures).

Welcome to the INDIGO IAM service website!

The open source, self-contained Identity And Access Management (IAM) solution for Scientific computing!

KEYCLOAK

- Keycloak is one of the most widely used open source Identity and Access Management (IAM) solutions.
- Compatible with OpenID Connect, OAuth2 and SAML, enabling integration with EduGAIN (and used, for example, by CERN for centralized Authentication and SSO).
- Keycloak was originally developed by Red Hat, but in April 2023, Red Hat donated Keycloak to the Cloud Native Computing Foundation (CNCF), remaining open source, with its governance now more open to community contributions.



Open Source Identity and Access Management

Add authentication to applications and secure services with minimum effort.
No need to deal with storing users or authenticating users.

Keycloak provides user federation, strong authentication, user management, fine-grained authorization, and more.

Latest release 26.1.2

8 minutes Keycloak video demo...

Server-side vs Client-side

- As you have seen in the video demo, setting up Keycloak on the server-side is very straightforward, allowing easy management of realms, users, groups, and applications.

- But what about the client-side? Here, things get more complicated.

Step-by-Step Guide

- Obtain an Authorization Code: the client redirects the user to Keycloak's login page, the user authenticates with their credentials, Keycloak returns an Authorization Code to the client.
- Exchange the Authorization Code for a Token: the client sends a request to Keycloak with the Authorization Code, the client ID and the redirect URI registered on the server.
- In subsequent requests, the client includes the Access Token and when it expires, it can be refreshed using the Refresh Token, or the user must log in again if the Refresh Token is also expired.
- The Security aspect is even more complex and requires careful management to protect applications and user data.
- But it's not enough: over time, the code must be maintained and updated to address new vulnerabilities and emerging threats.

Security Best Practices in Keycloak Authentication (1)

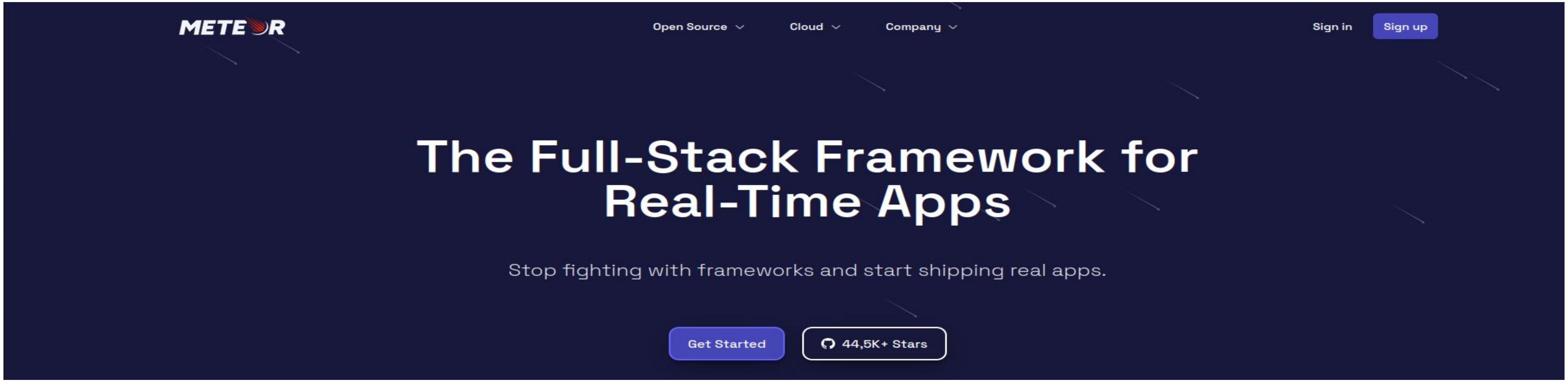
- Use PKCE (Proof Key for Code Exchange) for public clients (prevents “Authorization Code interception” attacks).
- Enable HTTPS for all requests (prevents interception and “man-in-the-middle” attacks).
- Use Short-Lived Access Tokens and Refresh Tokens (reduces the risk of theft, Access Tokens should be short-lived, ~15 minutes).
- Validate Tokens on the backend (always check the Token signature and expiration before accepting it).

Security Best Practices in Keycloak Authentication (2)

- Restrict Token Scope (if you request an SSH connection, you should only be allowed to do that).
- Handle Logout correctly (ensure users are logged out from all connected applications and revoke Refresh Tokens on logout to prevent session hijacking).
- These best practices focus only on Authentication, but Authorization and SSO must also be properly managed to ensure a complete and secure Identity Management system.
- How can we avoid having to think and worry about client-side security issues?

Meteor.js

- JavaScript framework with built-in Login and Accounts packages ready to use with your applications.
- "Never rebuild an Authentication system again" quotes the slogan of Meteor.js.
- Provides Authentication and Security out of the box, so you don't have to worry about managing them manually.

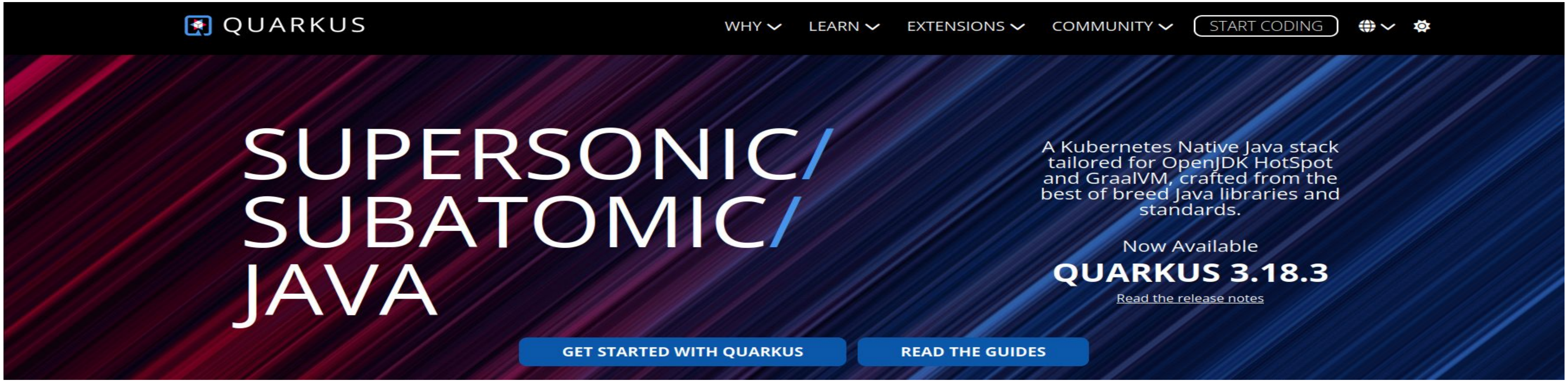


The Full-Stack Framework for Real-Time Apps

Stop fighting with frameworks and start shipping real apps.

Quarkus

- High performance (supersonic) and extremely lightweight (subatomic) Java framework.
- Native integration with Keycloak: since both Quarkus and Keycloak were developed and sponsored by Red Hat, they work together to provide a secure and scalable Identity Management solution.



A Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM, crafted from the best of breed Java libraries and standards.

Now Available
QUARKUS 3.18.3
Read the release notes

What we have seen today

- Keycloak is an open-source Identity and Access Management (IAM) solution.
- Very straightforward on the server side but more complex on the client side.
- Frameworks like Meteor.js and Quarkus provide built-in Authentication, Authorization, and SSO features to simplify integration.

Thanks for your attention!