IMPLEMENTATION OF OPAQUE TOKENS FOR IRODS - KEYCLOAK OPENID SOLUTION

WORKSHOP ON CLOUD STORAGE SYNCHRONIZATION AND SHARING SERVICES
COPENHAGEN, 27-29 JAN 2020

RUBÉN JESÚS GARCÍA-HERNÁNDEZ (1), MARTIN GOLASOWSKI (2)

(1) BAdW-LRZ, (2) IT4Innovations
INTRODUCTION

Technologies

• OpenID
  ◦ Open standard and decentralized authentication protocol

• Keycloak
  ◦ Open source Identity and Access Management solution
  ◦ Single-Sign On, Identity Brokering and Social Login, User Federation, Client Adapters
  ◦ Admin Console, Account Management Console, Standard Protocols, Authorization Services

• iRODS
  ◦ The Integrated Rule-Oriented Data System is open source data management software
  ◦ Aimed at deployment in mission critical environments
  ◦ Virtualizes data storage resources
  ◦ Supports microservices, storage systems, authentication, networking, databases, rule engines, and an extensible API
PROBLEM STATEMENT

High-level description

• The standard solutions for iRODS OpenID authentication send tokens using the username field
• This username field has a maximum length of 1024+64 bytes

• Keycloak provides non-opaque JWT tokens with extensive information, with signature. Tokens exceed the length mentioned above (typical: 1200 bytes, up to 65000 bytes)

• The iRODS / Keycloak combination, due to the issue above, produces an iRODS error when the token is sent from client to server: USER_PACKSTRUCT_INPUT_ERR
PROBLEM STATEMENT

Diagram

heliumdatacommons/auth_microservice

Irods-contrib/irods_auth_plugin_openid

iRODS server

irods/irods

communication

USER_PACKSTRUCT_INPUT_ERR

Login URL Token

User interface code

Keycloak

Irods/

python-irodsclient
SOLUTIONS

• Depending on whether the user is available or not:

A) For web-based applications interfacing directly with the user
   ◦ Use parallel execution to perform the query in the background,
   ◦ While the user is led through the authentication
   ◦ Send the data to the user once it is gathered.

B) For back-end applications, the solution above is not applicable.
   ◦ Implement opaque tokens in microservice by accepting a hash of the token.
   ◦ Pre-authorize the token by talking to microservice before submitting to iRODS.
   ◦ Optimization: hash token in iRODS libraries if >1024 bytes.
CONSORTIUM

Large-scale EXecution for Industry & Society

CONTACTS

Rubén Jesús García Hernández (LRZ)
garcia@lrz.de

Martin Golasowski (IT4I)
martin.golasowski@vsb.cz