INDIGO IAM development status

Enrico Vianello (INFN-CNAF)

INDIGO IAM Workshop and Technical Hackathon
Feb. 10-12 @ CERN

INDIGO IAM

INFN CNAF
Istituto Nazionale di Fisica Nucleare

Latest release: v1.11.0

Latest release v1.11.0 - release notes (1)

IAM v1.11.0 has been released on Feb. 3rd
Release notes: https://github.com/indigo-iam/iam/releases/tag/v1.11.0

What's changed

- IAM optional groups excluded from VOMS AC (Fix #886)
  - when a group in IAM is labeled as **wlcg.optional-group**, such group will appear neither in the token (not a change) nor in the VOMS AC (this is the change)
  - in practice we decoupled the **wlcg.optional-group** concept from the **voms.role** one → *roles* could become a first class entity in the near future
- **Restricted access to iam:admin.read|write and scim:read|write scopes**
  - only admins can obtain tokens with these scopes, through admin-approved clients
  - any policy defined on these scopes is now effect-less
- Exclude *openid* scope from scope policies
  - we're excluding the "id-token" request logic from any scope filtering logic

Latest release v1.11.0 - release notes (2)

Added

- **Add POST endpoint for registration requests confirmation** (fix #862)
  - Many tools (e.g. Outlook) try to protect their users from phishing URLs performing a HEAD request to the URL contained into the registration request confirmation email. This HEAD was confirming the registration request without any user interaction and, consequently, it was causing a 401 error on the subsequent user's click.
  - The original page linked into the confirmation email now it's a page where user must confirm with another interaction and a subsequent POST to another endpoint which will effectively confirm the registration.
- (Experimental) Implement MFA
  - A 2FA can be enabled for local credentials
- Add explanation message on the user (device) code page (fix #300)
- Add confirmation before rotate client secret (fix/mitigate #864, #634)
- **Redirect to login page when signing AUP** without being authenticated (fix #855)

Latest release v1.11.0 - release notes (3)

Fixed

- **Fix CERN lifecycle handler** (*see next slides*)
- Fix account mapping in VOMS AA + Fix account research API by checking both certificate subject and issuer in VOMS AA
- Client-credentials flow won't create a refresh token (*but it wasn't returned*)
- Fix missing update of matchingPolicy (Fix #851)
- Prevent the issue of broken SAML login flow (Fix #710)
- Fix the list of user's group membership requests see by admins (Fix #866)
- Fix missing creation of an approved site during device code flow
- Combine all scope filtering logic into one ScopeFilter
- Fix authorization on some me endpoints

Latest release v1.11.0 - release notes (4)

https://github.com/indigo-iam/iam/releases/tag/v1.11.0

Note:

- This is **a release with no database migrations** → easier to upgrade/rollback

Changes on lifecycle handlers

Lifecycle Handlers - Main problems

Expired Accounts Handler:

- User is considered as expired at the precise moment set on his “endTime” attribute, considering both Date and Time part

CERN lifecycle handler:

- No grace period → **The handler was suspending users without a grace period** when the user’s participation got from HR database was a 404 not found
- Performance → **Two calls per user** to HR API to retrieve quite the same information
- **Too many logged events** → each run changes timestamp label for every user and an event is trigger even if the label value is the same

Lifecycle Handlers - Solutions

Expired Accounts Handler:

- User is considered as expired and starts the grace period when its *endTime* value is greater than current Date, **considering only the Date part**

CERN lifecycle handler:

- **No more suspensions** → plays with users endTime in order to make them start the grace period at next expired accounts handler iteration
  - User's **endTime value is synchronized with the current experiment participation endDate** value (retrieved from HR API). In case of a 404 from HR API + endTime not expired → user's endTime is set to current Date
- Performance → **One single call** per user to HR API is now done
- Fewer logged events: **no timestamp label** is set and events are raised only if label's value has really changed
- It keeps restoring accounts suspended by Expired Accounts Handler in case a valid experiment participation is found

CERN Lifecycle Handler logic (1)

Case	Changes on user
No “**hr.cern.cern_person_id**" label set	Ignored by this logic.
The "**hr.cern.ignore**" label is set	"**hr.cern.status**" → **IGNORED** "**hr.cern.message**" = "Skipping account as requested by the 'ignore' label"
Runtime exception on retrieving user's VOPerson record from hr db api	"**hr.cern.status**" → **ERROR** "**hr.cern.message**" → "Account not updated: HR DB error"
404 Not Found on retrieving user's VOPerson record from hr db api → *this means user has no valid participation to the experiment*	"**hr.cern.status**" ← **EXPIRED** "**hr.cern.message**" ← "No person with id {cernPersonId} found on HR DB" (*if not yet expired*) **User.endTime** ← **current Date** *Note*: this setting will trigger the grace period as soon as the *ExpiredAccountsHandler* will run the day after

CERN Lifecycle Handler logic (2)

Case	Changes on user
VOPerson record retrieved but **no participation** to the experiment is found into the record obtained → *this case should not happen but it has to be managed*	"**hr.cern.status**" ← **EXPIRED** "**hr.cern.message**" ← "No participation to {experiment} found" User.**GivenName** ← VOPerson.**firstName** User.**FamilyName** ← VOPerson.**Name** User.**email** ← VOPerson.**email** * (*if not yet expired*) **User.endTime** ← **current Date** *Note*: this setting will trigger the grace period as soon as the *ExpiredAccountsHandler* will run the day after ** if label "hr.cern.skip-email-synch" is not present*
VOPerson record retrieved, a **valid participation** to the experiment is found and user is currently active (not suspended)	"**hr.cern.status**" ← **VO_MEMBER** "**hr.cern.message**" ← "Account's membership to the experiment synchronized" User.**GivenName** ← VOPerson.**firstName** User.**FamilyName** ← VOPerson.**Name** User.**email** ← VOPerson.**email** * User.**endTime** ← VOPerson.**endDate** ** ** if label "hr.cern.skip-email-synch" is not present* *** if label "hr.cern.skip-enddate-synch" is not present*

CERN Lifecycle Handler logic (3)

Case	Changes on user
VOPerson record retrieved, a **valid participation** to the experiment is found and user has been **suspended** by the *ExpiredAccountsHandler* (= not manually by an admin)	"**hr.cern.status**" ← **VO_MEMBER** "**hr.cern.message**" ← "Account's membership to the experiment synchronized" User.**GivenName** ← VOPerson.**firstName** User.**FamilyName** ← VOPerson.**Name** User.**email** ← VOPerson.**email** * User.**endTime** ← VOPerson.**endDate** ** User.**active** ← **true** “**lifecycle.status**” label is removed ** if label "hr.cern.skip-email-synch" is not present* *** if label "hr.cern.skip-enddate-synch" is not present*

Then, the CERN lifecycle handler has the ability to restore but not suspend users → Users expiration is managed only by the internal “endTime” handler.

What’s next

Current main development activities (1)

On review / to be rebased

- oidc-agent clients assigned to whom approved it
- Fix #809 JDBC session on same db
- AUP requirement configurable per account
- ROLE_READ to allow seeing account details of all users
- User’s registration "Notes" field as optional
- Spring Boot upgrade to latest 2.7 version
  - Makes MySQL 8 mandatory

Probably on next IAM v1.12.0

Current main development activities (2)

In progress

- Support RFC 8707 to request Access Tokens audience
- Integrate 2FA when login with external identity providers
  - IAM should request 2FA to the remote provider if mandatory and/or understand if it has been already done by the user or not → we're following the guidelines defined within EOSC Beyond project and drafted in the document named Design Document: Advanced Authentication Methods Across EOSC Nodes
- Porting the needed MitreID's classes into IAM login service codebase
  - Main benefits: codebase easier to maintain, easier migration to latest Spring Security libraries, completing the iam-persistence layer allows the development of new components such as → garbage collector, a version of IAM which is not storing access-token on database
- Evaluate if we can start from scratch with Spring Authorization Server implementation or evolve current logic
  - as soon as the porting of the needed MitreID classes is completed the picture will be clearer
- Development of new IAM dashboard → see next presentation

Current main development activities (3)

Longer term

- Scope policy integration with **Open Policy Agent**
  - a proof of concept has been already done
- Support **OpenID Federations**
- Conformance with AARC BluePrint Architecture and (more) guidelines
- Implement Spring Authorization Server framework

Conclusions

- Several adds/fixes within v1.11.0 → easy to upgrade with no db migrations
  - restrictions for admin and scim scopes
  - enabled MFA for local credentials
  - users lifecycle fixes
- 1.12.0 can be already prepared with other features/fixes
  - new role “reader”
  - db migrations expected
- The short and longer term development roadmap is known
  - simplifying the codebase (importing what is needed by MitreID) → benefits for the development of IAM with no Access Tokens on database and for next Spring Authorization Server migration/implementation
  - RFC 8707, MFA on external IDPs, OPA integration, OIDC Federation, etc..