Wednesday, November 27

INDIGO IAM v1.10.3 release status

What's Changed

- Add confirmation before rotate client secret by @SteDev2 in #875
- Fix account mapping in VOMS AA by @rmiccoli in #872
- Add POST endpoint for registration requests confirmation by @enricovianello in #881
- Fix CERN lifecycle handler by @enricovianello in #871
- Grant admin scopes to admin-approved clients only by @rmiccoli in 6bbaccd
- Client-credentials flow won't create a refresh token by @rmiccoli in #22 (MitreID)
- Redirect to login page when signing AUP by @federicaagostini in 5acde91
- Fix missing update of matchingPolicy by @garaimanoj in f15ef57

INDIGO IAM v1.11.0 scheduled at the end of the year

- Will not include the new dashboard yet but may release instructions on how to run it side to side with the current INDIGO IAM

Move out of MitreID: work in progress, still a lot to do, probably not before mid-2025

- MitreID is a problem with recent version of Shibboleth
- Requires implementing the persistence layer into INDIGO IAM
- Will require the new dashboard as the current one is based on JSP pages in MitreID

MFA: should be available in v1.11.0

- First only for local users, then extended to external IdPs and X509
- External account: will rely on IdP doing it if a given claim is present, else will do it
- Need to give a way for the user to contact the admin to reset the MFA

Discussion points

MFA

How can a user contact the admins when they have to reset the MFA? We're going to add a link at the TOTP verification page that will include a textbox for user comment and a submit button. This button will submit the request to the admins to reset the MFA.

Disabled user logins via X509

We found that when the user (with a certificate linked to their account) gets disabled, the "Login with X509 certificate" disappears. It is ok, but we need to change the error message as explained in the issue *Login screen with X.509 for suspended account*

- fix in progress

Rely on an external Attribute Authority

The idea is to retrieve user's information (affiliation, entitlements, attributes, etc.) from an external source (such as the HR database for CERN). Currently we have an handler inside the IAM codebase that queries the external server through an API exposed by another component which queries directly the HR database. This approach doesn't scale well if it's included into the "iam-login-service" component: if we think about a high availability deployment where more than one instance is running in parallel, there's no need for all of them to update users' info. Decoupling this logic is already possible by implementing a new module/component based on iam-persistence module. The idea is that info like user's personal data, affiliation, groups and attributes is synchronized with the external source. The component will expose an endpoint queried by the IAM instances. We could surely define which kind of response we're expecting from the remote source provider in order to support solutions/endpoints directly provided by the external source.

Another approach suggested by Francesco is to send all the AUDIT events to a message queue that can be read by another external component that can synchronize other IAM instances by using SCIM or other APIs.

From the IAM developers side, the first step can be documenting the whole list of AUDIT events. We can later combine these information with the ones that are relevant for the SKA experiment.

Geographically distributed High availability Setup - Findings and Challenges

- State storage
    - STFC: Galera as drop in replacement for MySQL/MariaDB
        - Tried a distributed setup with London/Glasgow/RAL
        - No effect on performance for read workload
        - Galera cluster is stable, resistant to loss of data etc.
        - Proposed to use SQL for session/cache since IRIS/SKA don't have high performance workload right now, redis gets compromised quite easily, difficult to setup redis cluster.
- DNS load balancing recovery time is <10s usually
- Plan forward
    - Kubernetes setup
    - Passing entitlement to other IAM? (Jens)

Issue discussed Github

- Metadata Refresh/Memory leak #880
- #710 SAML flow breaking if external IdP has incorrect cookie setup
- #884: systematic error at first attempt to refresh eduGAIN SAML metadata

Improvements to the account request validation workflow/dashboard

Story (Michel): in projects without a well-defined membership managed by an external service (membership database), account validation may require some project-specific process to assess the request validity and for a national or international project may involve tens of users. It is not suitable to make all of them INDIGO IAM administrators as they have no clue on what it means concretely and are exposed to mistakes. See https://github.com/indigo-iam/iam/issues/874.

Actions:

- Add a “validator role” giving a normal user the right to validate account requests without having the administrator role
    - Basically a normal user with the right to see the request dashboard and to validate the requests
    - An administrator remains implicitly a validator: no possibility to have an administrator who is not also a validator
- Add the possibility to enter validator notes, distinct from user notes. With a large team of validators, this will improve the sharing of the work, letting other validators know what the request processing status is if already handled by one of them. The easiest is a free-form note, possibly multiple ones as a log journal.
- Preserve user note and validator note(s) after registration
- Ability to communicate with the user through the notes to help him or interact with him (rather than by out-of-band/unlogged emails)
- Add a button to resend confirmation request email: when a user doesn’t validate his/her address, he/she often complains the email was not received…

For all the development in the dashboard to address the points above, agreement that it should be done only in the context of the new dashboard (in particular as it is possible to run the new dashboard side-by-side with the current one).

SQL Data too long error for table authentication_holder_request_parameter

We discussed a bit the issue: https://github.com/indigo-iam/iam/issues/873

The outcome is that this problem depends on the MitreID dependency which is storing the Authentication user's info into the database. We should fix this as soon as we move away from it (we expect to migrate to the latest Spring Security libraries during 2025, hopefully before June).

Incorrect fail match for InResponseto SAML field with redis HTTP session store

Fixed with this PR: https://github.com/indigo-iam/iam/pull/885

Performance Testing

The results of the load tests that are done at CERN don't match the performance on production at CERN. This may be related to the scope policies. In the development instance, there is no configured scope policy, unlike the production instances. (ATLAS has ~90, CMS has ~30). Each token request loops over all of the scope policies to check if it matches the current request. On most of the scope policies, it also needs to do path-matching which may increase the response time.
ACTIONS:

- Run load tests on an instance that has scope policies asking for a scope that has a path
- Developers are already looking into using an external policy agent (i.e. Open Policy Agent) and how it can be implemented with INDIGO IAM.

Thursday, November 28

(WLCG) Workflows that currently rely on long lived tokens

- … to be discussed on Thursday morning …
- recent suggestion in GUT mailing list (may be something like ersatz clients)
    - use short lived access token to talk with service
    - service should use token exchange to get refresh token (long lived)
    - use refresh token to get short lived access token for individual operations
- we already implemented something similar for FTS
    - using client credential flow (not optimal from security point of view) instead of token exchange
    - it'll be more tricky to do something similar for CE
        - design changes in current implementation and not completely clear
        - To be checked: HTCondor has a credential manager that stores a refresh token and refreshes the access token when it expires but will lead to a very high rate of access token requests as it is not connected to the operations that require the tokens (lots of unnecessary renewals)
- requirements coming from this design
    - performance
        - ATLAS submits ~ 1M jobs / day, ~ 2M FTS transfers / day
        - these both use-cases needs "long-lived tokens" => refresh tokens
        - with 30 days refresh token lifetime => token issuer needs to keep (1+2)*30 ~ 90M refresh tokens in database
        - refresh token -> many access tokens => x * 3M access tokens per day (I expect in average x << 10)
    - token exchange policies
        - could be as simple as: take scopes from access token and provide refresh token with same scopes
- *Does this sound like completely crazy requirements on token issuer?*
    - We can definitely design our software to be more lightweight…
        - our current design already use much lower granularity
        - even from security point of view creating refresh token per job / per transfers doesn't bring any benefits
        - Main point: we would like to understand better IAM limits => performance testing as integrated part of development cycle
    - Need to take into account not only INDIGO IAM but the other issuers: a performance test is scheduled at CERN for Keycloak using CERN SSO and the same test suite as for INDIGO IAM.
- Francesco: not convinced doing a token exchange for every request, in particular any transfer, really makes sense. Probably no token issuer can survive the resulting load
    - Need to be creative and think about other approaches using scopes to reduce the impact of a stolen token.
    - Avoid tokens based on the actual path as it will increase the number of access tokens. May have a scope for FTS/Rucio (the vast majority of

transfers) where the path is the VO root at the storage endpoint and can be used for all transfers to this site.

- Enrico: need for a common reference test platform and token usage model to evaluate performances and run tests
    - Need to validate if the difference between the results of tests by IAM team @CERN and those reported by the FTS team are really related to the policy engine (due to the high number of scopes used in FTS). Work on using OPA is progressing well, hopefully in a few months it will be possible to assess this.
    - Requires some prioritization with other work related to performances like moving access tokens out of the DB. Michel: maybe the priority should be on what could be delivered first and it seems to be OPA integration… Francesco: need to close MFA current work first so that we can deliver a first "experimental" version of MFA support that can be tested by advanced users.

CERN HR DB synchronization

- 12h synchronization causing problems
    - complications while resolving issues (e.g. with suspended accounts)
- don't just synchronize givenName, sn, email, status but also create new accounts
    - is this easy to do? probably limited usage by one or two LHC experiments…
    - does IAM provides generic interface for account synchronization or this is just CERN specific hack

~~"Weird" WLCG directions: refresh token protection token and multiple token issuers~~

OAuth resource indicators

- https://github.com/indigo-iam/iam/issues/381
- services (e.g. Rucio) would like to rely on (RFC) standards instead of IAM specific behavior

CERN

- Discussion around IAM at last WLCG OTF: main concern is performance, in particular from FTS

MFA

- Discussed the way we can allow administrators to rotate the secret used to encrypt/decrypt secrets. This update can be done in a way similar to the way used by database migrations (because it is in the end). The idea is that during the spring boot application bootstrap phase we can launch the query that updates the secrets, before letting the service start. One important thing is that we cannot have more than one instance that starts, otherwise we can have conflicts. A transaction is surely necessary. The agreed strategy is providing through the environment variables the current secret and the newest one. Surely this feature won't come with the first IAM release with MFA.

- Completed *Reset MFA for user* PR and merged into the main branch. Now we need to increase its coverage by adding more tests.

- Hide the “Enable/Disable MFA” buttons when *mfa* profile is not selected - work in progress!

Debugging problem Login screen with X.509 for suspended account #846
in progress