Evolving INDIGO IAM towards the next challenges

Federica Agostini[1], *Luca* Bassi[1,2], *Donald* Chung[4], *Ivan* De Simone[3], *Manoj* Garai[4], *Jacopo* Gasparetto[1], *Francesco* Giacomini[1], *Davide* Marcato[5], *Roberta* Miccoli[1], *Saiteja* Vennapusa[4], *Enrico* Vianello[1], and *Stefano E.* Zotti[1]

[1]INFN-CNAF, Bologna, Italy
[2]GARR, Roma, Italy
[3]Università degli Studi di Bologna, Bologna, Italy
[4]Science and Technology Facilities Council (UKRI-STFC), United Kingdom
[5]INFN-LNL, Legnaro, Italy

Abstract. INDIGO IAM (Identity and Access Management) is a comprehensive service that enables organizations to manage and control access to their resources and systems efficiently, by implementing a standard OAuth Authorization Service and OpenID Connect Provider. It has been chosen as the AAI solution by the WLCG community for the transition from VOMS proxy-based authorization to JSON web tokens.
This contribution describes the recent updates introduced by the latest IAM releases and the current roadmap for its evolution. In the near future, a primary focus is on avoiding to store access tokens in the database, to enhance the performance of both token issuance and token deletion. Another important milestone is the integration of a Multi-Factor Authentication mechanism. Additionally, substantial effort will be dedicated to migrating from outdated frameworks, such as MITREid Connect and AngularJS, to more stable and robust solutions based on Spring Security and React, respectively. As a consequence, a new dashboard is also being developed, aligned with the latest advances in the User Interface design.
This contribution highlights the progress made in the development roadmap described above, not forgetting the general auditing and performance improvements introduced with the latest releases or planned, such as the use of Open Policy Agent to re-implement the internal mechanism of the Scope Policy API.

1 Introduction

INDIGO Identity and Access Management (IAM) [1] plays a key role in distributed computing and offers organizations a solution to manage and secure access to their resources. It implements a standard OAuth Authorization Server [2] and an OpenID Connect (OIDC) Provider [3], supporting authentication and authorization workflows. Since 2017, the Worldwide LHC Computing Grid (WLCG) community has adopted INDIGO IAM as its preferred Authentication and Authorization Infrastructure (AAI) to transition from legacy VOMS proxy-based authorization [4] to modern JSON Web Tokens (JWTs) [5].

This article describes the latest advancements in INDIGO IAM, including its future roadmap, the technical innovations and the performance enhancements, which position the

service as a valuable solution for identity and access management in distributed environments.

2 The INDIGO IAM service

INDIGO IAM is a Spring Boot application [6], which implements the OAuth/OIDC specifications [2, 3], and is based on the MITREid Connect library [7]. INDIGO IAM supports multiple authentication mechanisms, including local authentication, SAML Identity Providers [8], SAML federations (e.g., EduGAIN), OIDC Providers (e.g., Google, GitHub). The service allows users to link various authentication credentials — coming from the above mentioned providers, but also X.509 certificates [9] and SSH keys — to a single account. INDIGO IAM offers a flexible user enrollment flow, with or without the intervention of an IAM administrator. In case of no administrator intervention, users authenticating through trusted Identity Providers can access the service for the first time without the need to submit a registration form. INDIGO IAM can optionally require users to sign a custom Acceptable Use Policy (AUP) at configurable intervals, restricting access to the service until they do so. INDIGO IAM issues JSON Web Tokens (JWT [5]) as credentials, following the OAuth/OIDC flows, supporting delegation and token renewal. A JWT usually includes identity information, attributes and capabilities, and it is consumed by the Relying Parties, where authorization decisions are applied. The adoption of standard flows allows for easy integration with third-party components. Additionally, INDIGO IAM can enable a VOMS (Virtual Organization Membership Service) Attribute Authority micro-service, named VOMS-AA. This micro-service provides backward-compatible support for VOMS [10], facilitating the integration of existing VOMS infrastructures with modern authentication mechanisms. Figure 1 provides a high-level overview of the INDIGO IAM architecture.

The INDIGO IAM service is designed for performance and scalability. Typically operating behind an NGINX reverse proxy, it efficiently handles incoming requests. A MySQL

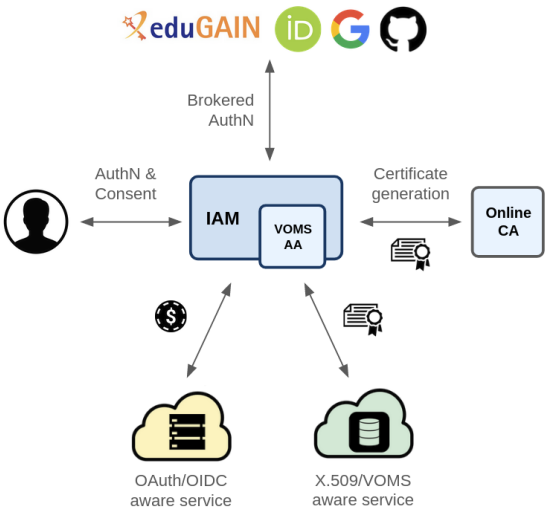

Figure 1: INDIGO IAM high-level architecture, including the VOMS-AA micro-service.

database ensures reliable data management, while a Redis service allows for horizontal scalability, thanks to the ability to store sessions and external caching. Currently, about 30 instances are deployed at INFN-CNAF for internal purposes and scientific collaborations; at least 4 instances are deployed at CERN, one for each LHC experiment, and other instances are used for organization management and testing (e.g., *dteam*); approximately 10 IAM services with custom configurations are deployed at STFC and IN2P3.

3 Recent releases and relevant upgrades

Since CHEP 2023 [4], the INDIGO IAM service has introduced several notable updates:

- **v1.8.2** (May 2023) has added scope-based authorization to the administrative API endpoints so that access is restricted to tokens which present new restricted admin scopes
- **v1.8.3** (December 2023) has hashed the access token value stored in the database to improve the performance, has added an external Redis caching mechanism, which now stores also the scopes allowed for a client, and has published the scopes on the well known endpoint
- **v1.8.4** (March 2024) has added enhancements to the database schema and improved the appearance and the usability of the login page
- **v1.9.0** (June 2024) has allowed the functionality todisable a client, with the consequence of stop issuing or refreshing new tokens, has added AUP management features which has made managing the VO similar to the legacy VOMS-Admin [10], has included the ability to provide more information from SCIM [11] endpoints (authorities, attributes and managed groups) and has added information about the usage of a client (days passed since the last token was issued)
- **v1.10.0** (August 2024) has enhanced AUP management and notifications, has added the automatic group enrollment upon user registration, has introduced a statistical endpoint and a strict quality check for local credential passwords
- **v1.10.1** (September 2024) has fixed minor bugs and updated the AngularJS [12] dependency to the latest version
- **v1.10.2** (October 2024) has fixed CERN lifecycle logic issues.

These updates demonstrate the ongoing efforts of the INDIGO IAM development team to enhance security, user experience and administrative functionalities.

4 Roadmap for future evolution

This section describes the future evolution of the INDIGO IAM service in terms of developments and supported features. These evolutions are planned to reach the targets of enhancing the performance, improving the security layer and supporting a diverse and growing ecosystem of users and services.

4.1 Multi-Factor Authentication

In order to further strengthen security, INDIGO IAM plans to implement Multi-Factor Authentication (MFA) [13] as part of the login flow. MFA adds an additional layer of protection by requiring users to provide multiple forms of verification, which is particularly critical in environments that handle sensitive data or require strict compliance with security standards.

An initial implementation of MFA [1] allows local authentication (i.e. login with username and password) using Time-based One-Time Passwords (TOTPs) [14]. In particular, each authenticated user can enable/disable MFA via a button on their homepage. Users must use an authenticator app (typically on a mobile device) to generate the necessary TOTPs for authentication. IAM administrators have the power to disable the MFA for a user at any time. Of course, MFA secrets are kept encrypted in the database to ensure confidentiality. Figure 2 shows the typical login flow with local authentication and MFA enabled.

Future releases of INDIGO IAM will extend MFA support to both external providers, such as SAML and OIDC, and X.509-based authentication, creating a more comprehensive security framework across different authentication systems. INDIGO IAM will not request a second factor if the external provider has already authenticated the user using MFA.

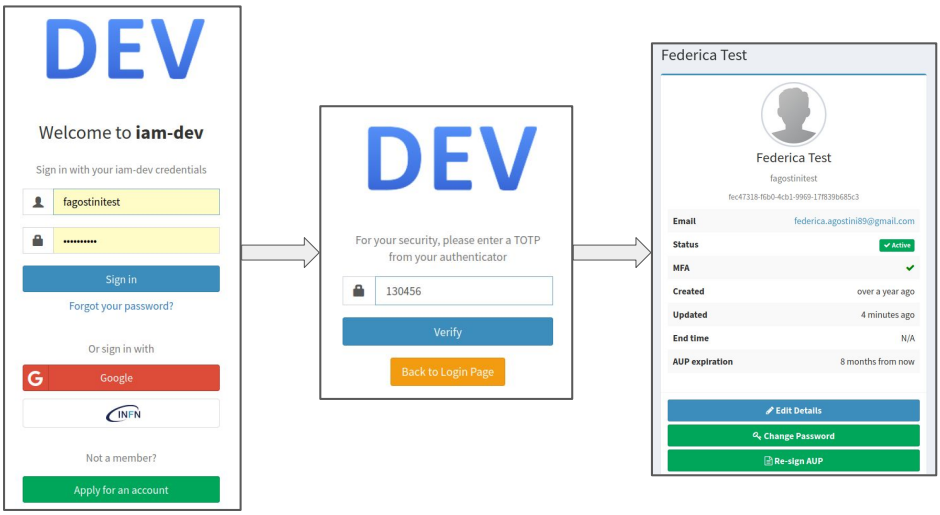

Figure 2: Demonstration of a login flow with MFA enabled, via local authentication (username/password) plus TOTP.

4.2 Transition to modern frameworks

4.2.1 Adoption of Spring Authorization Server

In line with future plans for technology upgrades, INDIGO IAM will be migrated to the Spring Authorization Server (SAS) [15] to manage authentication and authorization tasks. SAS offers a secure, lightweight, and highly customizable way to implement OAuth and OIDC services. This migration is driven by the limitations of the current MITREId Connect library [7], which has been forked into the INDIGO IAM repository and is currently self-maintained, since it has not shown significant updates or support in recent years. By moving to the Spring Authorization Server, INDIGO IAM will benefit from more modern Java/Spring architectures, ensuring long-term support, improved maintainability and better compliance with the latest OIDC and OAuth standards.

[1] At the time of writing, this initial implementation of MFA for local credentials (i.e. login with username and password) has been introduced in the INDIGO IAM v1.11.0 release (January 2025).

4.2.2 A new dashboard

In the context of framework migration, INDIGO IAM is designing a new user dashboard [16] that follows modern principles to improve usability and accessibility. The primary motivation behind this redesign is the deprecation of AngularJS [12]. React [17] has been chosen for its lightweight structure, flexibility and compatibility with a modern development stack based on HTML5, TypeScript and CSS. Moreover, the new architecture decouples the frontend from the OAuth/OIDC implementation, improving modularity and facilitating authentication and authorization. Figure 3 shows a preliminary view of the new INDIGO IAM user dashboard.

From the implementation perspective, this dashboard follows security best practices by managing authentication and authorization directly within the web application. It uses the OAuth Authorization Code flow with PKCE [18] for secure token exchange, with INDIGO IAM acting as both an Authorization Server and a Resource Server.

To further strengthen security, the architecture adopts a Backend-For-Frontend (BFF) pattern, where the BFF component serves as a confidential OAuth client that handles all responsibilities and forwards requests to the Resource Server after adding the appropriate access token. This approach prevents tokens from being exposed to the browser by using a cookie-based session mechanism and ensuring that rendering and computations occur entirely on the backend, exposing only the final HTML content.

The new dashboard is developed using Next.js [19], a modern framework that enables server-side rendering, for improved performance and security. It is designed to be lightweight, easily deployable in a containerized environment, and highly scalable to accommodate different needs.

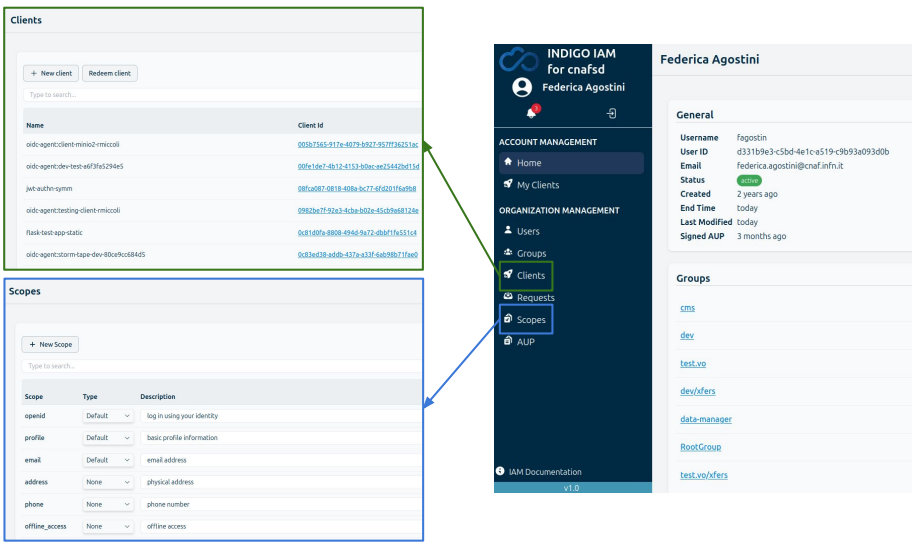

Figure 3: Example of the new React-based INDIGO IAM dashboard, showing the user homepage (right) and a focus on the Clients page (top left) and the System Scopes page (bottom left).

4.3 Avoid storing access tokens in the database

The INDIGO IAM development roadmap includes the need to avoid storing access tokens in the database. This change is expected to significantly improve service performance by simplifying token issuance and deletion processes. By adopting a stateless token model, the system will reduce latency and improve scalability, ensuring that it can handle the growing demands of distributed environments. The need for this change became evident during the 2024 WLCG Data Challenge [20], when the database accumulated millions of access tokens, most of which had already expired and were slowly being deleted, leading to performance degradation.

To implement this feature, access tokens used for IAM API endpoints will be dynamically validated, without persisting them in the database. Validation results may be cached with an eviction time that matches the token expiration date, reducing redundant checks. The only tokens that will still be stored are the revoked ones, preventing unauthorized access to resources.

4.4 Adoption of Open Policy Agent

One of the greatest technical advances in INDIGO IAM is the integration of Open Policy Agent (OPA) [21] as an authorization engine. OPA is based on *Rego*, a high-level declarative language that allows policies to be defined as code. OPA ensures low-latency policy decisions, even when processing a large number of rules. Any service that needs to make a policy decision can query OPA by passing arbitrary structured data (JSON or YAML) as input. The OPA engine then evaluates the incoming request against Rego policies and optionally external static data, returning a decision. An OPA decision is not limited to a simple *allow*/*deny* response but can also include arbitrarily structured data, allowing for finer policy enforcement. Figure 4 shows a schema of the OPA engine.

OPA will be integrated into INDIGO IAM in order to replace the current Scope Policy engine. This embedded service provides a mechanism to control access to token scopes by filtering the requested scopes based on policies applied to users or groups. Through Rego-based rules, a re-implementation of the IAM policy decision logic is already in place. The evolution includes the possibility to apply policies also to OAuth clients, enabling a finer grained control to every OAuth flow, also the ones not directly tied to a specific user. The

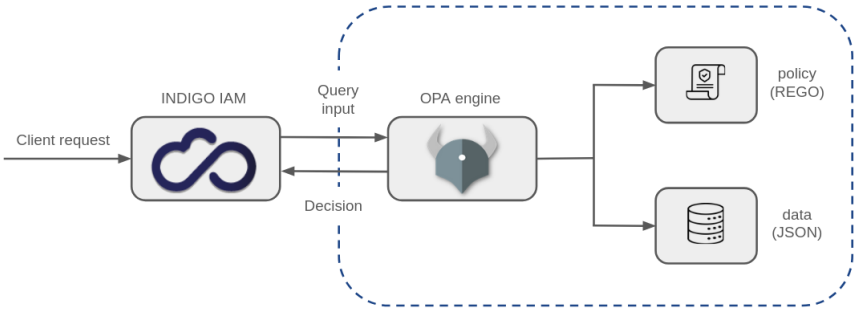

Figure 4: Schema of the OPA engine. A service which needs to make a policy decision can query OPA passing arbitrary structured data as input. The OPA engine evaluates the query input against REGO policies and optional data.

policy definitions remain backward-compatible with existing IAM configurations, ensuring a smooth transition to the new engine.

To test the performance impact of this integration, the OPA command line, offering a subcommand for profiling information, has been used and allowed to further optimize the policy decision logic. The results demonstrated that applying 10000 Scope Policies through OPA took approximately 130 milliseconds, while the traditional IAM implementation times out. These findings highlight the significant efficiency gains achievable through OPA, making it a solid authorization engine for INDIGO IAM and possibly other applications.

4.5 Support for OpenID Federation and AARC BPA guidelines

As part of its continuous evolution, INDIGO IAM is extending its capabilities to support OpenID Federation [22]. An OIDC federation consists in a multilateral federation, which allows dynamic trust establishment between entities through a hierarchical system of trust anchors and signed metadata, with the goal of building scalable and interoperable federations. This approach facilitates client registration and enhances security by automating trust relationships, reducing administrative overhead and ensuring consistent policy enforcement across participants.

Another upcoming development is to ensure compliance with the AARC Blueprint Architecture (BPA) [23] and related guidelines. The AARC BPA defines a set of modular software building blocks designed to enable federated access management for international research collaborations. By implementing AARC best practices and recommendations, INDIGO IAM aims at enhancing the interoperability between infrastructures, facilitating the integration with other identity and access management solutions in the research community.

5 Conclusion

INDIGO IAM is continuously improving identity and access management by addressing current challenges and preparing for future needs, ensuring its usefulness for the scientific and distributed computing communities. Key updates include performance enhancements and the integration of advanced security measures like MFA, making it more adaptable to users' needs. The adoption of modern software frameworks for its core parts and the development of a new dashboard further aim to enhance user experience.

The roadmap of INDIGO IAM reflects a clear vision for the future, focused on innovation, security and performance.

Acknowledgements

The work presented in this paper has been supported by the NextGenerationEU European initiative through the Italian Ministry of University and Research, PNRR Mission 4, Component 2 - ICSC [24]: Investment 1.4, Project code CN00000013 - CUP I53C21000340006; TeRABIT [25]: Investment 3.1, Project code IR0000022 - CUP I53C21000370006. This research was also co-funded by the Italian Complementary National Plan PNC-I.1 "Research initiatives for innovative technologies and pathways in the health and welfare sector" D.D. 931 of 06/06/2022, "DARE [26] - DigitAl lifelong pRevEntion" initiative, code PNC0000002.

References

[1] *INDIGO IAM*, DOI 10.5281/zenodo.14525993, https://indigo-iam.github.io/v/current/

[2] *The OAuth 2.0 Authorization Framework* , DOI 10.17487/RFC6749, URL https://www.rfc-editor.org/rfc/rfc6749, October 2012
[3] *OpenID Connect Core 1.0*, URL https://openid.net/specs/openid-connect-core-1_0.html, Dicember 2023
[4] T. Dack et al., *WLCG Transition from X.509 to Tokens. Status, Plans, and Timeline*, EPJ Web Conf. 295 (2024) 04054, DOI https://doi.org/10.1051/epjconf/202429504054
[5] *JSON Web Tokens*, DOI 10.17487/RFC7519, https://datatracker.ietf.org/doc/html/rfc7519, May 2015
[6] *Spring Boot*, https://spring.io/projects/spring-boot
[7] *MITREid Connect*, https://github.com/mitreid-connect/
[8] *Security Assertion Markup Language (SAML) V2.0*, http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html, March 2008
[9] *Internet X.509 Public Key Infrastructure Certificate*, DOI 10.17487/RFC5280, https://datatracker.ietf.org/doc/html/rfc5280, May 2008
[10] *Virtual Organisation Membership Service (VOMS)*, DOI 10.5281/zenodo.12634651, https://italiangrid.github.io/voms
[11] *System for Cross-domain Identity Management (SCIM)*, DOI 10.17487/RFC7642, https://datatracker.ietf.org/doc/html/rfc7644
[12] *AngularJS*, https://angularjs.org/
[13] *REFEDS MFA Profile*, DOI 10.5281/zenodo.5113295, https://refeds.org/profile/mfa
[14] *Time-Based One-Time Password (TOTP)*, DOI 10.17487/RFC6238, https://datatracker.ietf.org/doc/html/rfc6238
[15] *Spring Authorization Server*, https://spring.io/projects/spring-authorization-server
[16] *INDIGO IAM dashbaord*, https://github.com/indigo-iam/iam-dashboard
[17] *React*, https://react.dev/
[18] *Proof for Key Code Exchange (PKCE)*, DOI 10.17487/RFC7636, https://datatracker.ietf.org/doc/html/rfc7636
[19] *Next.js*, https://nextjs.org/
[20] A. Arora et al., *WLCG/DOMA Data Challenge 2024: Final Report*, DOI 10.5281/zenodo.11402618, June 2024
[21] *Open Policy Agent*, https://www.openpolicyagent.org/
[22] *OpenID Federation 1.0*, https://openid.net/specs/openid-federation-1_0.html
[23] *AARC BPA*, https://aarc-community.org/architecture/
[24] *ICSC*, https://www.supercomputing-icsc.it/en/icsc-home/
[25] *TeRABIT*, https://www.terabit-project.it/it/
[26] *DARE*, https://www.fondazionedare.it/en/