FenixEdu Connect: An Identity Management System for Academic Organizations

Paulo Ricardo Conde Branco
paulobranco@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2018

Abstract

As security breaches result in ever increasing damages to organizations worldwide the pressure is on to develop additional safeguards against these types of cyberattacks. Identity and Access Management systems sit at the forefront of this protection by providing system administrators with a consistent set of rules and processes for managing the digital identities of their users. However, the specific needs of academic institutions have for long prevented the adoption of the latest state of art technologies and practices at these organizations. This thesis proposes an open-source IAM solution designed to meet the authentication, authorization and identity management challenges of higher education institutions by describing a framework that allows each client institution to tailor the product to its specific requirements. The proposed implementation model is then validated against a possible deployment at Instituto Superior Tcnico, as a replacement for the existent identity management product.

Keywords: Identity, Authentication, Authorization, Identity and Access Management, OAuth, OpenID Connect, Software Development, Integration

1. Introduction

The past few years have seen a dramatic increase in the number of internet-enabled households, with the International Telecommunications Union (ITU) estimating that in 2017 more than half (55.1%) of the world's population was online. Naturally this figure is significantly higher in developed countries, where the latest estimates point to an adoption rate of 81% [6]. Academic institutions have for long taken advantage of the opportunities made available by the global connectivity to expand both their reach and resources. A clear example of this approach is the development of Massive Open Online Courses (MOOCs) consisting of a set of free resources (ranging from classes, notes, exercises and exams) which are made available online.

At the same time, this shift to an online learning model presented academic universities with its own set of security challenges, as organizations were forced to revamp the existing policies and security systems to cope with the additional demands from a realm of users who, while entitled to a set of resources, were not actually part of the universities' internal user-base. Academic institutions are often required to manage accesses for an extremely dynamic user base, often consisting of large thousands of status changes per yer as students start and complete their educational programmes. In addition, roles are often fluid: a student can simultaneously be a part-time teaching assistant or an employee at the university's IT services. This poses a challenge for the traditional role-based access control technologies in place in these organizations.

Initial attempts to manage this challenges mostly consisted of the deployment of Single Sign-On solutions, designed to allow users to authenticate with multiple internal applications using a single set of credentials. The Centralized Authentication Service (CAS)[1] was one of the most common SSO protocols in use by these institutions. The main advantages of these systems were clear: without the need to memorize a wide array of passwords users were free to choose a strong password for their SSO realm, thus ensuring the safety of their resources within the organization while limiting the consequences usually brought on by password reuse.

However, whereas the adoption of SSO systems was arguably a step forward, the issue of managing the users' identities within an organization still remained. Academic institutions often rely on different applications to hold student records, handle employee information, process payments and manage course data. Each of these applications makes up for part of the user's identity, often storing pieces of this information within their own walled gardens,

[1] https://www.apereo.org/projects/cas

with little to no integrations with the remaining applications. The need for an integrated solution that is able to tie into all applications and provide developers with an holistic view of each user's identity should thus be evident.

1.1. The FenixEdu Project

The online-learning paradigm led to the development of the first Learning Management Systems (LMS), applications where educational resources could be made available to the academic community over a network, often being shared online. As these systems grew in popularity, so did the number of features available in each one, with universities often resorting to in-house development teams to integrate other types of services (such as tuition payments or official document requests), with direct consequences to the products complexity and maintainability. Both commercial (Blackboard[2]) and non-profit (Moodle[3]) organizations attempted to develop learning management systems that could be customized to each institutions needs.

The FenixEdu Project, started in 2002 at Instituto Superior Tcnico (IST) aimed to provide a complete and integrated set of open-source software platforms for academic organization management, including a Learning Management System (LMS), a Student Management System (SMS) and a Content Management System (CMS)[5]. FenixEdu's extensible and modular approach to the development of these tools was fundamental to the adoption of its products outside IST, allowing each implementing institution to tailor it to its specific requirements.

However, the decoupled nature of this solution also creates its unique set of challenges as a Single-Sign On solution must be deployed to ensure users can roam between the different applications without being asked to re-authenticate with each one. While support for third-party SSO systems (CAS) is already implemented in FenixEdus suite of products, identity management is still bundled with one of the products (FenixEdu Academic) and only a small set of features is available. These features extend beyond the purpose of FenixEdu Academic and thus, should be replaced by a complete IAM solution that offers a central management point for authentication and user identities across the FenixEdu suite of products.

1.2. Objectives

Lending on the knowledge of the main issues surrounding Identity and Access Management (IAM) systems at academic institutions this body of work aims to lay the foundations for an hybrid solution, capable of offering a similar level of features to the ones currently offered by closed-source cloud-based IAM systems, augmented with additional identity management tools aimed at providing applications and developers with a centralized gateway for user identity data. The provided solution will leverage the latest authentication and authorization standards, ranging from OAuth to OpenID and SAML. In addition, as part of the FenixEdu suite of products, it will be offered in an open-source model with an architectural model focused on the customization and extensibility by each client organization.

2. Background

2.1. Digital Identity & Identity Federation

The concept of identity can be traced back to even before humans learned to speak. Both physical features, behaviour patterns or common gestures were used to identify and characterise a single individual, distinguishing her from her peers. The first languages made this process easier, as both individuals and physical objects could now be named and described using a set of words.

Nowadays, when asked to provide a proof of identity most people resort to personally identifiable documents, such as a driver's license, or a social security card. These documents, often issued by a trusted third party such as a state or a government carry identifiers that are unique for each person, enabling third party entities to identify the owner.

When a customer uses a credit card to pay she is required to enter a PIN (Personal Identification Code) to authorize the purchase. While the card's number is unique worldwide (no two cards have the same number), the user is still required to enter an additional piece of information to ensure the card is being used by its legitimate owner. This constitutes a simple example of credentials, that is, a set of private (and optionally public) data that can be used to assert the authenticity of a given claim (in this case: "the owner is in possession of the card").

In addition to identifiers and credentials, there is often a set of information that, while not unique, is at the core of an entity's digital identity: attributes. Attributes define characteristics associated with a given entity, which can either be based on personal traits, such as fingerprint data and eye color, or temporary, such as an email address or a student number. Attributes, unlike identifiers, are not expected to be used to assert a subject's identity. Rather they make up, along with identifiers and credentials, the foundations for a digital identity.

Digital identities are, similarly to their physical counterparts, dependent of a trust relationship between all the involved parties. *Windley* defines trust as “a firm belief in the veracity, good faith and honesty of another party with respect to a transaction that involves some risk” [7]. Multiple identities for the same user can bear different levels of trust

[2] https://www.blackboard.com/

[3] https://moodle.org/

depending on the set of attributes they contain or the context they are originated from. As an example, whereas both can lay the same claims, a shopkeeper is more likely to accept a driver's license as a proof of date of birth than a library card, since it places a higher degree of trust on the government agency that issued the driver's license than it does on the library. In the same way, before asserting a user's identity, a digital system must be able to trust the provided credentials and, ideally, that they are being held by the correct entity.

As the concept of Digital Identities became widespread, so did the challenges of allowing users access to resources that were not under the control of their home organizations. This often required guest credentials to be set up, which if not carefully managed, could allow users to retain access to resources beyond the intended time. The solution came in the form of identity federation protocols which build upon a pre-established trust relation between multiple organizations to allow users to authenticate in every one with the same credentials used in their home institution.

2.2. Public Key Cryptography & Digital Signatures

The idea of disguising a message to prevent it from being read in transit can be traced back to 1900 BC [1]. Only individuals who where in possession of the key (or the set of steps performed to hide the message) were able to reverse the process. This is known as symmetric cryptography and, while subject to significant advances in algorithm strength and performance it is still ruled by the same underlying principles.

Despite the relatively performance of this encryption algorithms they suffer from the challenge of how to distribute the decryption keys in a secure way, since most transfer media can, in theory, be compromised in transit. The solution came in the form of public key cryptography (also known as asymmetric) which is based around the concept of key pairs (known as the public and private keys) which are related to each other in a way that allows one (the public part) to be discovered from the other but not the reverse.

As an example, assuming that both Bob and Alice have previously generated their key pairs and made their public keys available to the other one (either directly or published in an online exchange) if Bob wanted to send Alice a secret message he would encrypt it with Alice's public key (which is publicly available). Since the keys form a related pair, the message can now only be decrypted with Alice's private key which is, hopefully, kept private by Alice.

While the above scenario illustrates the advantage of public key cryptography, by removing the need for the secure distribution of the encryption key there is yet another notable use of asymmetric cryptography: digital signatures. If Bob wanted Alice to be sure that a certain message came from him (and thus was not tampered in-transit) he could encrypt it with his own private key. A message encrypted with a private key can be decrypted by anyone with the corresponding public key, so this would not ensure the secrecy of its contents. However, while it could be read by any user, only someone in possession of Bob's private key could alter its contents and re-sign it. By keeping his private key secure Bob is able to send signed messages who can be verified by anyone in possession of his public key but can't be changed. Digital signatures will play a major role in Connect's authentication tokens.

2.3. OAuth

As the number of online services increased so did the need for simple and efficient data exchange and integration amongst them. The rise of web APIs as a privileged interface between systems paved the way for many of the web services that are now commonplace. However, it also came with its own set of challenges as users who wanted to access a service that consumed resources from third party websites were often asked to provide their authentication credentials for these websites. These credentials were not restricted in scope and were stored in databases to prevent users from having to provide them every time they accessed the service. The solution to this security risk came in the form of OAuth, an open authorization protocol which allowed users to grant third party entities access to a subset of their resources without having to disclose their credentials to these entities [4]. OAuth is currently in its second version, OAuth 2.0 and while it has undergone significant changes since its initial release the same key principles still apply. Figure 1 provides an overview of the most common authorization flow [2].

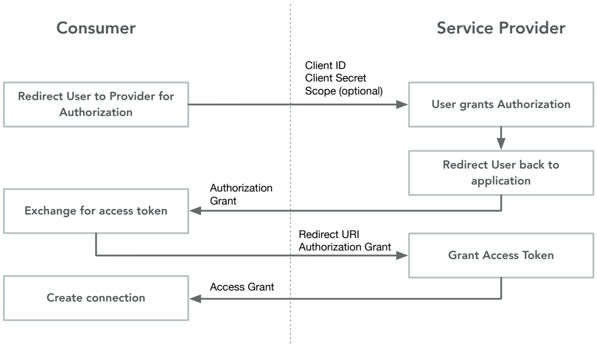

Figure 1: OAuth 2 Authorization Grant flow.

2.4. JWT

JSON Web Token (JWT) is an open standard that defines a compact and self-contained way for securely transfering information between parties as a JSON object [3]. The standard defines an object comprised of three individual section with the first one coomonly refered to as the header, which provides informatino on the cryptographic operations to apply to the token's contents. JWT's can either be signed or encrypted, supporting both symmetric and asymmetric cryptography algorithms for both choices. The second section, also known as the body or claims set, is comprised of a set of key/value pairs that make up the set of assertions that the sending party wishes to expose. Some of the key names are reserved for specific purposes, but there is no limitation on the number or type of key/value pairs that can be contained in this section. The last section includes additional data related to the token's signature or encryption. The final token is assembled by concatenating the three sections with a single period (.) after the first two sections have been encoded with base64 to ensure URL safe transfers. Being based on the aforementioned principles of public key cryptography JWTs provide developers with the flexibility to locally verify the token's authenticity without having to contact a remote service, as long as they are in possession of the issuer's public key.

3. Implementation

3.1. System Requirements

After the initial analysis covering the main issues that surround IAM systems at academic institutions a set of system requirements were identified. The following list provides a summarized overview:

- Authentication: Support for multiple authentication providers, Multi-Factor Authentication and passwordless authentication.
- Identity Management: Connect should be able to aggregate user profile information from multiple systems and provide developers (and other applications) with a unified view of this data. It should have basic account management features and the ability to invite users with 0-day provisioning.
- Access Delegation: Connect should act as an OAuth authorization server for the organization enabling developers to request access to all the public APIs. It should support OpenID Connect authentication over OAuth.
- External Integrations: It should be interoperable with existing LDAP directories. It should have the ability to act as a SAML Identity Provider.
- Security: Connect should provide users with a detailed log of all security-related activities that pertain to their account. Users should be able to remotely terminate Connect sessions. Administrators should have access to comprehensive logs. Audit logs should be made available for forensic purposes.
- Technological: It should be implemented as a Java web application, based on the foundation provided by the Spring[4] framework.

3.2. Architecture

Connect's high level architecture follows a classic three-tier approach, whereby applications are divided in presentation, business and persistence layers. One of its key design requirements was for the backend and frontend to be completely isolated layers, with communication between the two occurring exclusively through RESTful API calls. While it would be possible to develop the two modules as part of this body of work, it was decided to drop the frontend module completely, since each client organization will likely require a custom frontend (that matches its internal design guidelines) to be developed on top of Connect's core product. This decision is aligned with the overall vision of the FenixEdu team for this solution. Connect is to be made available to each institution as a purely RESTful component, with each institution being responsible for developing an appropriate frontend that exposes the required features in a way that is compliant with its own design system. Additionally, the decouplement between these two layers allows them to be individually updated.

The reliance on Spring's set of frameworks is leveraged by FenixEdu Connect to provide each client institution with a significant number of extensibility points, where organizations can easily inject their own logic and/or extend the built-in services. This is accomplished by the development of contribution modules, which when compiled alongside Connect Core are automatically recognized and merged into the Core component's logic. Figure 2 provides an overview of the high level architecture of a Connect solution, where the client institution developed three contribution modules with internal business logic (in brown) and obtained two authentication modules from the open source community (in green) forming the complete FenixEdu Connect solution.

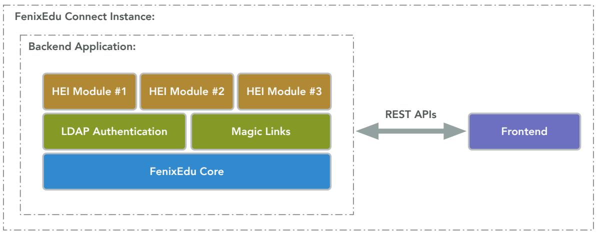

Figure 2: An example of a Connect deployment.

[4] https://spring.io

3.3. Implementation Details

3.3.1 Authentication

As previously discussed in section 2, credentials are one of the corner stones of an IAM system allowing users to lay claims over an existing digital identity. Despite the many forms that credentials can take, they are usually grouped into three types: (1) knowledge factors, which require users to provide information that only they know, (2) ownership factors, where users are asked to provide something that only they own, such as a smart-card and (3) inherence factors, where users provide information based on personal traits, such as their fingerprints.

While the specific forms of credentials range in the level of security they provide the trend has moved on to rely on a combination of two or more factors to authenticate a principal. This is known as Two-Factor Authentication (2FA) (or Multi-Factor Authentication (MFA) in a broader sense, not limited to the use of only two factors).

Connect's Authentication infrastructure defines two different security levels: primary authentication and secondary authentication.

Primary authentication is required for all users to access any restricted feature of FenixEdu Connect and is usually performed with a common pair of user/password credentials. Connect leverages the foundations provided by Spring Security to enable organizations to deploy multiple primary authentication strategies. The `PrimaryAuthenticationProvider` interface defines the abstract behavior required to be implemented by every primary authentication provider. The provided solution ships with two concrete implementations: an internal provider, which uses Connect's database to authenticate users and an LDAP authentication provider which uses an underlying LDAP directory to perform user authentication. Subclasses of `PrimaryAuthenticationProvider` are automatically registered in a central registry, implemented in the `DelegatingAuthenticationProvider` class. This entity ties into Spring Security's authentication flow and delegates authentication attempts to the providers.

Secondary authentication is optional by default, with the exception of some resources which warrant an additional security verification before access can be granted. From an architectural standpoint, it follows the same approach as the primary authentication subsystem. Secondary authentication factors implement the `AuthenticationFactorProvider` class, which defines the common behaviour that should be supported by every factor (create, activate, start verification, complete verification and delete). These classes are automatically registered in the `MFAManagementService` which acts as the central gateway for secondary authentication, managing the factors associated with each user and their verification process. Connect ships with support for Time-Based One-Time Passwords (TOTPs), SMS verification and Universal 2-Factor (U2F), although client institutions can easily create their own authentication factors simply by implementing the required interfaces in a contribution module.

Another pattern of authentication that has been gaining traction is passwordless authentication. As evidenced by its name, it is based on the premise of allowing users to authenticate without using a password. This is usually accomplished through magic links. When a user wants to authenticate with a given service, it provides it with her (previously registered) email address. The service generates a unique link, that is associated with the requestor, persisted in the database and emailed to the user. The user is instructed to visit her inbox, clicks the link (which redirects her to the service) and, if the link is still valid, is authenticated. Connect implements passwordless authentication through magic links in a separate module, `connect-magic-links`, allowing organizations to opt-in/out to use this type of authentication if they so choose.

3.3.2 User Management

One of the key featuers of every IAM system is its ability to manage users, user accounts and their associated lifecycles. Connect offers a complete REST API for user account management, allowing system administrators to create new users, modify the associated information of each one, manage authentication factors and perform account lifecycle transitions. Connect defines a set of six possible account states, which are highlighted in figure 3 along with the associated transitions between them.

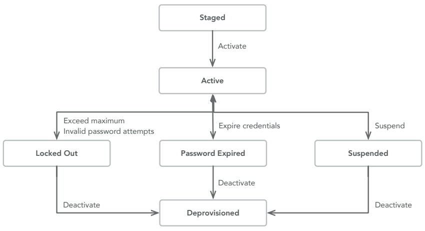

Figure 3: User account lifecycle in Connect.

Connect provides authorized personal with the ability to invite external users to join the organizations, specifying the full set of privileges to be available to the invited user. This is known as 0-day provisioning, and allows users to access all their resources as soon as the invite is accepted, without having to wait for manual approval processes.

Regarding password resets, Connect provides three built-in solutions: SMS, Email and WebRTC.

The SMS and email reset strategies consist of sending a short lived token to the user, which is then used to authorize the password reset. However, during the initial problem analysis it was found that a significant number of students at IST were unable to reset their passwords since their registered phone numbers were no longer in service, which forced them to visit the university's IT services to reset their password. This prompted the creation of an additional built-in solution, which leverages the latest WebRTC protocols to establish an interactive session between the user and a member of the IT service's support staff. After a successful identification the support member can register an additional phone number or email address to be used for password recovery. Password reset strategies implement the `PasswordResetMethod` interface and are automatically registered with a central service, the `CredentialsManagementService`, which manages credential updates and reset requests. Organizations can easily deploy their own reset methods by simply implementing this interface.

While it could be argued that IAM systems should maintain all the users' profile information in a self-contained fashion that is rarely possible. Organizations rely on a number of different applications which typically hold information in semi-isolated silos, with limited integration and interoperability with the remaining products. Connect's architecture of identity management is based on the premise of it acting as a central gateway for user identity interacting with all applications to push and pull data from their internal repositories while providing developers with a single, unified view of the user's data. To accomplish this, Connect defines an interface, `ProfileProvider`, which abstracts the communication between a Connect instance and an application responsible for a set of profile attributes. A profile provider should be able to return a set of attributes (identified by unique keys) for any given user. Optionally, it may support persisting new values for these attributes. Since the users' profile data can be spread out over multiple applications, Connect defines a concrete implementation of this class, the `DelegatingProfileProvider (DPP)`, which implements support for multiple sub-profile-providers (implementations of the `DelegateProfileProvider` interface) to interact with Connect. Whenever an application requires access to a set of profile attributes it can request them through Connect's REST APIs. Internally, the DPP service will demultiplex the attribute set over its registered profile providers, requesting from each one the attributes it is authoritative over. The responses are then merged back and sent to the requestor.

The profile provider subsystem provides a pluggable architecture for organizations to expose profile attributes from any application. Academic institutions can easily deploy a module to interface with an Human Resources and Payroll (HRP) system, such as SAP, to expose the user's employee number in Connect's profile information. This deviates from the traditional approach where systems would be directly integrated with the HRP system, usually resorting to its public APIs. If the system was ever deprecated, all the systems that relied on the employee number information would have to be refactored.

3.3.3 OAuth Authorization Server

In addition to offering Identity Management features Connect's design philosophy rests on it simultaneously acting as a central point for the registration and authorization of OAuth applications. To achieve this, Connect relies on the foundations offered by the Spring Security framework to implement an OAuth Authorization Server.

Connect offers a complete REST API for managing all the resources associated with the OAuth features including, but not limited to, applications, scopes, grant types and authorizations.

Regarding the domain model, the `OAuthApplication` class is used to persist the relevant information for the registered OAuth applications including basic metadata, credentials, authorized grant types and required scopes. A distinction is made between required and optional scopes, allowing developers to fine-tune the requested permissions to the essential resources while still allowing users to opt-in for providing additional information. The `OAuthScope` class represents an OAuth Scope with an option to restrict it to applications created by users with an administration role. This allows for the use of OAuth for internal applications, as regular users will not see administrator-only scopes and thus, will be unable to create OAuthApplications which require these scopes.

In a typical OAuth flow, where access tokens are generated as opaque strings, applications are forced to query the authorization server to validate the authenticity of a given token, as well as the set of scopes which have been granted access by the user. Following the core design principle of allowing for the offline validation of the issued tokens Connect's OAuth access tokens are bundled as JWTs. Applications can have instant access to a set of claims that allows them to check the granted tokens without the need to contact FenixEdu Connect.

In addition to the typical OAuth features, this solution includes support for OpenID Connect (OIDC), an authentication protocol built on top of the foundations provided by OAuth 2.0. OIDC's shares the majority of its request flow with OAuth: the user attempts to access a protected resource and is redirected to the OAuth authorization server for authentication and scope approval. At this point, an additional scope, `openid`, is requested. If access is granted, the OAuth Authorization server issues, along with the typical OAuth access token a digitally signed `ID Token`, which contains information regarding the user which can be used by the application to authenticate her.

3.3.4 Security and Monitoring

As the core component for the users' account security FenixEdu Connect must be able to provide users with an historical overview of the events that may have had a security impact on their accounts. These are typically associated with changes in the user's profile data or credentials. An attacker who was able to obtain temporary access to a user's account could, for instance, register a secondary authentication factor that was in her possession without the user ever becoming aware. Beyond the regular logging and auditing information, which is only available for system operators, Connect keeps a special record of these actions, which is made available to the user through the `api/v1/users/{user}/securityEvents` API endpoint. Users are able to query the endpoint to get a complete list of security events pertaining to their account, or limit the scope to a single type of event or date.

While the most security-conscious users will likely opt to use two-factor authentication whenever it is available, the often voluntary nature of this security feature tends to result in less than ideal adoption rates. With the goal of increasing security and reducing the chances of an account becoming compromised FenixEdu Connect actively monitors authentication attempts against suspicious behaviours. Through a direct integration with the `DelegatingAuthenticationProvider` Connect's `SecurityManager` service validates each authentication event against a set of checks:

- **Repeated attempts with invalid credentials:** Connect limits the number of invalid authentication attempts to a configurable value, after which the offending IP address is blocked from making further requests.
- **Significant location changes between login events:** IP location is used to obtain an approximate distance between the current authentication event and the last. If this distance is incompatible with the time required to travel it at a preset speed the account is suspended and the user is notified.

The use of JWTs as the transport media for authentication assertions requires a careful approach to the issue of key management. While the JWT standard supports both symmetric and asymmetric signature protocols, using a shared secret would require an additional protocol to be developed to ensure the secure key distribution to client applications, whenever a new one wanted to use the Connect infrastructure. As such, it was decided to limit Connect's support to asymmetric keys. At any given point, Connect maintains a set of valid signing keys which are made available to a trusted set of applications. One of those keys is considered the active key and is used to sign the issued tokens but any token signed with a key from the valid keys set should still be accepted by the resource servers. Administrators can manage signing keys, but they are always created internally from a secure random generator. To assist in the horizontal scalability of a FenixEdu Connect system, the keys are persisted in its database, encrypted with AES-CBC with 128-bit keys which can be individual for each generated key. The `KeyEncryptionService` encompasses all the logic required to create, validate and securely fetch the keys used to sign the issued JWTs.

Application logs are an invaluable tool for monitoring the real-time state of any software component as well as a major contribution for troubleshooting issues or retracing user behaviour. onnect takes advantage of Spring's built in integration with SLF4J[5] which provides a common facade for logs while allowing developers to select from a wide range of concrete logging implementations. This allows Connect's log output strategies to vary from simple console output to sophisticated log analysis tools, such as Greylog[6] or Logstash[7] with only minor changes to the configuration files and included dependencies. Most of the relevant actions are logged. While the format of these logs is configurable by each institution (through configuration properties) Connect implements some additional logic to supply relevant information to the logging agents including the authenticated principal and the ID of the token used to assert that identity. This information is automatically added as part of every log line and can be parsed by automated tools to, for instance, reconstruct the set of steps taken by a given user over a time period.

However, the high stakes nature of an IAM application require additional safeguards that allow for forensic reconstructions of the system's state over time. This just isn't possible with simple logs. To accomplish this, FenixEdu Connect takes advantage of a mature audit log framework, *Envers*,

[5] https://www.slf4j.org/

[6] https://www.graylog.org/

[7] https://www.elastic.co/products/logstash

that keeps track of every change made to Connect's database (similarly to a versioning control system such as git). This information is stored alongside with the authenticated principle and timestamp, allowing administrators to conduct a forensic retrospective of the application's data over time, one change at a time.

4. Evaluation

While methods based on the quantitative analysis of key metrics (such as the total execution time or resource requirements of a given task) are often preferred for the evaluation of software applications there were a set of constraints that limited the effectiveness of these methods in the evaluation of the implemented solution. FenixEdu Connect is based on the premise of expanding the realm of features that are typically available in the IAM systems of higher education institutions. As such, it is not expected of this solution to provide a noticeable improvement in the execution time of common tasks, such as authentication or access decisions, preventing any comparison with the existing systems' performance from providing a relevant result.

The most significant validation results would only be achieved for a deployment of this solution in a production environment, where it could be subject to the common load and usage patterns. Only then would some of Connect's main advantages, such as the reduction in the number of support tickets achieved from the multiple self-served password reset strategies have a measurable (and thus comparable) impact. Naturally, the security implications of deploying an untested IAM system to production prevented this from happening in a compatible timeframe.

The evaluation of this solution then rests on a case study centred around Instituto Superior Técnico's IAM challenges and how a deployment of FenixEdu Connect would contribute to mitigate them.

4.1. Authentication

While the use of 2-Factor Authentication (2FA) has become commonplace on most enterprise organizations it is still not part of Tcnico Lisboa's centralized authentication. Support for 2FA must be implemented by each individual application, forcing users to register their 2FA providers on multiple services. This is a clear violation of the Separation of Concerns principle. When operating in an environment protected by a SSO system applications should not have to perform custom authentication logic and should, instead, rely on the existing infrastructure to validate the requestor's identity and provide any required personal information.

FenixEdu Connect allows users to register a number of 2FA factors ranging from Time-based One-Time Passwords, SMS phone numbers or U2F tokens in a centralized application that are made available on every subsequent authentication attempt.

4.2. Account Management

The ID project (`https://id.tecnico.ulisboa.pt`) aimed to provide both end users and developers with a common gateway for user authentication augmenting the existing CAS service with additional account management tools including a self-served password recovery system. There are currently two reset strategies: (1) an SMS code sent to the user's registered mobile phone or (2) the use of the Portuguese Citizen Card for identity verification, which suffer from a set of challenges already mentioned in section 3.

FenixEdu Connect extends the existing password recovery strategies by offering users the ability to reset their credentials by email, using an alternative address that may have been previously registered by the user. When the registered contact information cannot be used for password recovery (either because it is missing or no longer available) FenixEdu Connect provides an alternative approach consisting of a video call with a member of the IT support staff, implemented on top of the WebRTC protocol suite. During this call the support staff member will validate the user's identity by a predetermined protocol (such as requiring the user to display her ID card and comparing its photo with the user). If validation is successful, the member of the support staff will be able to add an additional contact point (an email address or phone number) to be used by the user for password recovery.

4.3. Identity Provider

At IST user identities live in semi-isolated silos maintained by each application, with attributes commonly duplicated in multiple applications, in some instances with conflicting values. Students who enroll in a degree fill out their personal information in FenixEdu Academic, which exports a small subset of it to the core LDAP directory. Other applications may query the LDAP directory for these attributes, but they are unable to access the information that was left in FenixEdu Academic.

FenixEdu Connect's would allow for all applications at IST to share a global view of the user's profile data through the use of small add-on modules, known as profile providers, responsible for establishing an interface between the Connect service and the internal applications.

Connect provides an abstraction layer over where the profile data is located. Applications are not made aware of where a specific profile attribute is stored. Rather they request it to be read or written through a Connect instance. This flexibility allows for data to be moved between authoritative sources, without any disruption to the existing applications.

4.4. OAuth Use Case

As part of the architectural shift that IST has been undergoing there is an ever increasing number of applications exposing REST APIs for interoperability with external systems. Some of these APIs are secured with OAuth. Developers who want to take advantage of multiple APIs are forced to register their applications in each service. As a consequence, users are also forced to manually authorize access to each one. While this problem is mitigated by the presence of an SSO realm, where users are able to authenticate with most services using the same credentials the issue of having to grant individual authorizations for each service is deterimental to the adoption of these resources by developers at IST.

The deployment of FenixEdu Connect would severely reduce the complexity of this use case by allowing services to register their scopes in a centralized location, responsible for managing the OAuth access credentials to all services. Users would only have to undergo the authorization process a single time, with the generated credentials being usefull for all the services secured by FenixEdu Connect.

4.5. Section Management Use Case

Under the current SSO system in place at Instituto Superior Tcnico it is impossible to list the current active sessions for each user or to remotely terminate a single session (or all of them). A user whose account credentials have been stolen may be unaware of this fact until unusual changes are made to the account. There is no security warnings for significant location changes between subsequent authentications or any limit on the amount of invalid password attempts that can be performed on a specific account.

FenixEdu Connect offers complete session management allowing users to, for instance, remotely terminate any of their active sessions from any device. Users would be able to remotely end a session that was accidentally left open in a public computer from their device or end all active sessions simultaneously. In addition, the implemented solution ships with some basic security measures that detect brute-force attacks and block the originating IP address after a number of incorrect authentication attempts. There is also protection against significant location changes in subsequent login attempts. All of these events are logged and are made available to the end user who can then evaluate if further security measures (such as a password reset) are necessary.

4.6. External Integrations Use Case

Beyond the mentioned authentication mechanisms IST actively takes part in a number of identity federations, mainly to interface with external vendors (such as Microsoft's Office365) or scientific publication repositories. These federations are based on the SAML 2 protocol. The existing SAML IdP lives in the ID Project and, while effective, is severely limited in the attributes it can provide to the client applications, as its only attribute source is the university's LDAP directory.

FenixEdu Connect's SAML IdP leverages the extensible profile provider infrastructure to allow SAML client applications to request access to any profile attribute that is managed by Connect, regardless of the application responsible for maintaining it.

4.7. Comparison with commercial solutions

Both Okta[8] and Auth0[9] offer commercial solutions with a similar feature set. Their pricing structure is based on a per-user subscription model with prices ranging from 2 to 10$/month/user. While the key protocols and technologies are supported, the closed-source nature of this solutions prevent them from being molded to fit the organizations's specific requirements. Developers lack the tools to extend the existing solutions and any updates must come from upstream, limiting the organization's ability to protect itself from new (still unpatched) threats.

4.8. Rollout Plan

The initial rollout phase of FenixEdu Connect at Instituto Superior Tcnico is designed to validate the performance profile of the solution under real load conditions. This step is expected to provide valuable insights on possible optimizations while also limiting the extent of the damages brought on by any unidentified security vulnerabilities that may be disclosed during this trial period. It consists of migrating a small number of non-vital applications to Connect while still allowing the ID system to maintain operational control.

Once the required performance optimizations were identified and carried out the remaining applications can be switched over to FenixEdu Connect. The development of a Bennu Authentication Provider module would instantly allow all Bennu-based applications (such as the FenixEdu suite of products and the DOT applications) to use Connect, without any further modifications to their codebases. SAML integrations, including Office365, OpenStack or the Portuguese Citizen Card would also be instantly supported by FenixEdu Connect, since it exposes a SAML 2.0 IdP. Applications

[8] http://okta.com/

[9] https://auth0.com/

which could not be migrated to Connect would still use the ID system for a predetermined amount of time.

The third deployment phase would consist of the identification of the legacy applications that cannot be directly ported to FenixEdu Connect and the development of auxiliary systems to overcome this challenge. Solutions may include the development of additional authentication modules, such as support for Kerberos[10] or NTLM[11].

Finally, the last phase would allow Connect to act as the authentication provider for external systems, that are rarely included in IAM solutions, such as the development of a contribution module to allow it to act as a RADIUS[12] server (for authenticating clients in IST's eduroam wireless network) or the integration of FenixEdu Connect with the campi's building security systems.

5. Conclusions

This thesis set out to design and implement the foundations for a customizable Identity and Access Management system that was able to meet the requirements of a wide array of academic institutions. While the initial requirement analysis was heavily inspired by the specific challenges faced at Instituto Superior Tcnico, its architecture was designed around the premise of ease of extensibility, limiting the opinionated decisions to no more than a few sensitive defaults whenever it was justified. Both small to medium HEIs should be able to customize the final solution to fit their specific requirements.

The motivation behind this product was never to develop an entire new authentication technology or protocol. Rather, it focused on solving a real-world need of academic institutions: to reliably authenticate its users across a variety of channels and ensure all applications have secure access to their identity. Its outcome is thus, two-fold: from a theoretical standpoint, it provided the community with a framework for the design of extensible IAM solutions, capable of supporting the custom business practices in place at each institution. From a practical standpoint, it provided a working solution for how such systems could be implemented.

The final solution still requires a frontend layer to be implemented on top of the provided Connect Core. This work is already underway at IST's IT department with the first trials of the complete Connect solution scheduled to begin in early 2019.

As it is expected of a solution that is responsible for protecting a wide range of personal information, FenixEdu Connect must still undergo a thorough security validation by IST's IT department before it can be used in a production environment. Future developments should focus on fostering the adoption of the system by third party developers, by creating the necessary documentation and SDKs that allows FenixEdu Connect to effortlessly integrated with the next generation of applications at IST. At the same time, it is expected of developers to extend the current capabilities of this solution, by leveraging the driving force of the open-source community to develop contribution modules that allow Connect to be used with an even greater number of applications and authentication protocols.

References

[1] Cypher Research Laboratories. A brief history of cryptography, 2006. http://www.cypher.com.au/crypto_history.htm, Last accessed on 2018-09-04.

[2] D. Hardt. The OAuth 2.0 Authorization Framework. RFC 6749, October 2012.

[3] M. Jones, J. Bradley, and N. Sakimura. JSON Web Token (JWT). RFC 7519, IETF, May 2015.

[4] S.-T. Sun and K. Beznosov. The devil is in the (implementation) details: An empirical analysis of oauth sso systems. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 378–390, New York, NY, USA, 2012. ACM.

[5] I. S. Tcnico. The fenixedu project: an open-source academic information platform, March 2001.

[6] I. T. Union. Ict facts and figures 2017, 2017.

[7] P. Windley. *Digital Identity*. O'Reilly Media, Inc., 2005.

[10] https://web.mit.edu/kerberos/

[11] https://msdn.microsoft.com/en-us/library/cc236699.aspx

[12] https://freeradius.org/