Des millions d'utilisateurs avec Shibboleth : comment ça se passe (bien) ?

Nicolas Romero
Pôle national de compétences identités et accès – Ministère de l'Éducation Nationale et de la Jeunesse
10 rue Molière
45000 Orléans

Pierre Sagne
Pôle national de compétences identités et accès – Ministère de l'Éducation Nationale et de la Jeunesse
10 rue Molière
45000 Orléans

Vincent Leblanc
Pôle national de compétences identités et accès – Ministère de l'Éducation Nationale et de la Jeunesse
10 rue Molière
45000 Orléans

Résumé

La gestion des identités et des accès dans l'Éducation Nationale fait face à plusieurs défis. D'abord technique : les produits déployés dans la trentaine de centres de production ne répondent plus aux besoins, ou sont en fin de vie. Ensuite organisationnel : les cloisonnements académiques sont remis en questions par les mutualisations et il est nécessaire de gérer les identités et les accès à un niveau régional, voire national. Et enfin conjoncturel : il faut ouvrir les SI à de nombreux publics et partenaires, par exemple pour la dématérialisation des démarches (« Action Publique 2022 »), ou encore la saisie des résultats de contrôles continus par les lycées agricoles ou à l'étranger (réforme du baccalauréat).

Dans ce contexte, l'Éducation Nationale a engagé la ré-urbanisation de ses infrastructures de gestion des identités et des accès. Ainsi, un dispositif national, EduConnect, qui s'appuie sur Shibboleth IdP (Identity Provider, fournisseur d'identité) et Shibboleth SP (Service Provider, fournisseur de service), remplace désormais les dispositifs de gestion des identités et des accès des parents et élèves propres à chaque académie. C'est le point d'entrée unique de tous les parents et les élèves vers tous les services éducatifs, portés par l'Éducation Nationale ou par des tiers (ENT, applications de vie scolaire, …). Il doit donc être hautement disponible puisqu'on cible plusieurs dizaines de millions d'utilisateurs par jour. De même, les infrastructures de gestion des identités et des accès pour les personnels évoluent, avec toujours la nécessité d'homogénéiser le déploiement sur tous les centres de production.

Cette présentation décrit cette démarche et aborde les problématiques rencontrées, notamment sur la haute disponibilité et la montée en charge. L'industrialisation avec Ansible sera également abordée.

Mots-clefs

Shibboleth, SSO, fédération d'identité, Ansible, disponibilité

1 Introduction

Le système d'information de l'Éducation Nationale est construit comme une juxtaposition des systèmes d'information académiques et nationaux.

Chaque académie gère les applications et les populations de son périmètre géographique dans son propre centre de production, mais le Ministère pilote les infrastructures et le développement des applications nationales, qui peuvent être installées en académie ou centralisées dans des centres de production nationaux. En particulier, la mise en œuvre de la gestion des identités et des accès au début des années 2000 s'est appuyée sur une démarche basée sur des infrastructures identiques dans chaque académie, respectant d'une part les standards et l'état de l'art, et implémentant d'autre part un système de fédération d'identité. Ainsi, des guichets d'authentification académiques normalisés permettent aux personnels de l'Éducation Nationale, aux parents et aux élèves d'accéder aux services en ligne proposés au niveau national ou académique.

Aujourd'hui, l'Éducation Nationale a initié un chantier de ré-urbanisation de ses infrastructures de gestion des identités et des accès. En effet, les produits d'infrastructure de gestion des accès et de fédération sont en fin de vie, sans chemin de migration vers la nouvelle offre logicielle proposée par l'éditeur. Par ailleurs, les bouleversements engendrés par la nouvelle organisation territoriale et l'ouverture des systèmes d'information à de nouveaux partenaires remettent en question les cloisonnements académiques et nécessitent de gérer les identités et les accès à un niveau régional, voire national. Le projet EduConnect, qui vise à fournir un compte national unique aux parents et élèves pour les services numériques éducatifs est un produit issu de cette démarche, et s'appuie sur de nouvelles briques techniques. En cible, la population des personnels et des externes s'inscrira également dans cette refonte.

Nous présentons ici les critères de décision qui ont amené à privilégier Shibboleth IdP et SP en remplacement des produits éditeurs précédemment déployés, ainsi que les adaptations nécessaires à notre contexte et les améliorations apportées.

2 Contexte

2.1 Critères de choix

Une centaine d'applications développées au niveau national et des dizaines d'applications locales dans chaque académie s'appuient sur ces services d'authentification, de SSO, de contrôle d'accès et de fédération d'identité via SAML 2.0. Ces éléments sont assurés par des produits « éditeur » qui ont été choisis il y a plus d'une quinzaine d'années. Chaque académie est ainsi fournisseur d'identité et les centres de production nationaux sont fournisseurs de service. De plus, un hub de fédération de type « hub-and-spoke », construit sur les mêmes technologies, simplifie la mise en place des fédérations internes à l'Éducation Nationale et permet l'ouverture vers des services externes. L'exigence primordiale pour le remplacement des

infrastructures de contrôle d'accès et de fédération d'identité a bien sûr été que la nouvelle architecture garantisse une compatibilité complète avec l'existant. Il s'agissait donc de conserver a minima les fonctionnalités :

- de Single-Sign-ON (SSO) ;
- de contrôle d'accès par url applicative au niveau des reverse-proxy ;
- de fédération SAML 2.0 en mode Idp et SP ;
- de transmission des attributs aux applications via les entêtes HTTP ;
- d'authentification substantielle pour certains accès en fonction des urls applicatives ;
- d'industrialisation des configurations.

La transition des briques techniques existantes vers de nouvelles était coûteuse, quels que soient les produits cibles. Ce n'était donc pas un critère de choix pertinent. En revanche, d'autres exigences concernant des points d'amélioration recensés au fil des ans ont guidé les choix d'évolution de l'architecture.

En premier lieu, le fonctionnement adopté à l'origine s'appuyait fortement sur les caractéristiques des produits éditeur utilisés, ce qui a grandement simplifié la démarche initiale d'industrialisation de la gestion des identités et des accès. Mais le revers de la médaille est apparu lorsque l'éditeur n'a plus fait évoluer ses produits de concert, ou pire, que les caractéristiques sont devenues des limitations. Ainsi, les règles d'accès du produit de gestion des habilitations et du SSO, qui étaient amplement suffisantes lorsqu'il s'agissait d'identifier quelques macro-profils, ne permettent pas d'exprimer la complexité des besoins d'aujourd'hui. Par ailleurs, les spécificités d'implémentation du standard SAML 2.0 dans le produit assurant la fédération d'identité ont parfois rendu complexe l'interconnexion avec d'autres entités SAML 2.0. La dépendance à ces produits a également rendu difficile l'ouverture à de nouveaux protocoles, comme OpenIDConnect, dans la mesure où l'éditeur avait fait le choix de ne pas l'implémenter et qu'ils n'étaient pas facilement interopérables ou extensibles. Deux exigences fortes découlèrent de ce constat. La première était de diminuer l'adhérence trop forte aux produits de remplacement choisis, quels qu'ils soient. La deuxième était de concevoir une architecture adaptative, afin de permettre l'utilisation de nouveaux protocoles, par exemple Oauth/OpenidConnect, sans remettre en question la totalité des applications.

Nous souhaitions également aller plus loin dans les possibilités d'automatisation des déploiements. Jusqu'alors, seules les définitions des règles de contrôle d'accès pouvaient s'appliquer par script, la configuration des authentifications et des fédérations nécessitant des interventions manuelles. De plus, la communication des différents éléments de configuration des accès (scripts et documentations) s'effectuait par mail, de façon totalement indépendante des éléments relatifs aux applications proprement dites. Il était ainsi difficile de suivre les installations dans les différentes académies et d'analyser les problèmes remontés. Ces éléments ont fait apparaître la nécessité de pouvoir réaliser le déploiement automatique des autorisations d'accès et le cas échéant des fédérations d'identité associées, de façon coordonnée avec l'installation d'une application.

La perspective de constituer un guichet d'accès unique à un niveau national pour les parents et les élèves imposait évidemment de pouvoir répondre à des exigences de volumétrie, de performances et de haute disponibilité.

Enfin, dans une perspective d'assurer une continuité de service en cas de coupure partielle ou totale d'Internet, il était indispensable que les infrastructures de gestion des identités et des accès puissent continuer à fonctionner, ce qui interdisait des solutions purement orientées cloud.

2.2 Pourquoi Shibboleth ?

Le choix de Shibboleth IdP et SP s'est imposé après avoir étudié d'autres alternatives. Côté éditeurs, les solutions étaient clairement orientées vers le cloud, en imposant parfois le déploiement de toute ou partie de l'infrastructure à l'extérieur, ce qui n'était pas possible à la fois pour des questions de sensibilité des données et de l'introduction d'une dépendance à internet qui n'existe pas aujourd'hui dans le réseau de l'Éducation nationale. De plus, on retrouvait les avantages et les inconvénients de ce type de produits : une intégration poussée dès lors qu'on utilisait la suite provenant du même éditeur, mais des limitations inhérentes à la clientèle principale visée par l'éditeur : pas d'équivalent d'un SP SAML 2.0, ou un IdP très couplé avec un portail orienté accès aux principales solutions SaaS du marché, pour ne citer que deux exemples. À mi-chemin entre produit éditeur et Opensource, Keycloak commençait tout juste à apparaître comme une solution viable, mais la stratégie de RedHat n'était pas encore très claire à son sujet. Enfin, CAS arrivait en version 5, mais il s'agissait d'une réécriture d'une grande partie du logiciel, portée par un seul développeur, et nos doutes sur sa stabilité et sa pérennité furent renforcés par des tests qui montraient notamment des manques dans la prise en charge de SAML, ce qui constituait un problème majeur pour nous. De plus, dans le cadre du remplacement du hub de fédération, CAS et Keycloak ne permettaient pas de transférer correctement les niveaux d'authentification.

Au final, aucune de ces solutions ne remplissait tous les critères.

Le couple Shibboleth IdP/SP avait l'avantage d'être très répandu dans la communauté Enseignement Supérieur et Recherche, et également dans des déploiements locaux en académies. C'était aussi la seule solution permettant de faire du contrôle d'accès de manière compatible avec nos briques existantes, sans avoir à réécrire toutes les applications.

Contrairement aux produits utilisés jusqu'alors, dont le déploiement nécessite au moins quatre types de serveurs par rôle IdP ou SP (serveur frontal, serveur d'application dédié, serveur de fédération, serveur LDAP, sans compter les consoles d'administration), ces briques sont légères (un serveur d'application pour l'IdP, un serveur frontal dédié pour le SP), facilement déployables et configurables de manière industrialisée.

3 Mise en œuvre et adaptations

3.1 Authentification

L’architecture ouverte de Shibboleth IdP offre beaucoup de possibilités en termes d’adaptation aux besoins. Nous avons par exemple mis à profit le mécanisme des intercepteurs pour mettre en place une série de contrôles post-authentification sur l’état du compte de l’utilisateur et lui proposer les actions nécessaires à la poursuite de la connexion, comme la validation de son adresse mail ou la modification de son mot de passe temporaire. Cette ouverture nous a aussi permis de consolider des sources de données différentes (LDAP, base de données, langage de script) pour construire les vecteurs d’identité en fonction des services qui les consomment.

La possibilité d’implémenter de nouvelles méthodes d’authentification nous a également permis d’ajouter la connexion par FranceConnect comme une méthode standard. Nous avons aussi utilisé ce mécanisme pour développer une procédure d’auto-enrôlement.

3.2 Contrôle d’accès

Pour la traduction des politiques d'habilitation de nos applications en règles d’accès, nous nous sommes appuyés sur les fonctionnalités d’autorisation de Shibboleth SP, qui permettent d'exprimer des règles logiques et des expressions régulières basées sur les valeurs des attributs d’identité des utilisateurs et sur les méthodes d'authentification. En revanche, la configuration de ces autorisations dans Shibboleth SP se fait par chemin et par fournisseur de service. La notion d’application, regroupant tous les points d’accès à une même application, n’existe pas. Cela nous a posé problème par rapport à l’existant. En effet, d’une part, le SP est mutualisé pour plusieurs applications. D’autre part, les politiques d'habilitation doivent pouvoir évoluer par application de manière autonome et pas de manière globale. Or, nous n’étions pas en mesure d’ajouter ou de modifier simplement les règles d’accès à une application de manière indépendante des applications gérées par le même fournisseur de service. Par ailleurs, dans Shibboleth SP, les attributs utilisés pour les règles d’accès et ceux pour constituer le vecteur d’identité fourni aux applications sont deux ensembles séparés, ce qui n’est pas la logique des politiques d’habilitation telles que nous les définissons.

Nous avons donc créé un formalisme YAML, qui nous permet de traduire en éléments techniques ces règles d’accès et de s’affranchir de ces limitations. Cette manière de procéder offre également un certain degré d’abstraction par rapport à l’implémentation réelle des règles, limitant ainsi le couplage avec la brique technique qui va les traiter. Chaque application dispose donc d’un fichier YAML indépendant et le mécanisme de déploiement ou de mise à jour s’appuie sur l’ensemble des fichiers pour regrouper dans un seul fichier request_mapper les règles d'accès aux différentes applications correspondantes.

3.3 Industrialisation

3.3.1 Utilisation d’Ansible

Afin de gérer de manière unifiée les primo installations et les mises à jour, nous nous sommes tournés vers Ansible. Ainsi, le même package peut être utilisé pour effectuer des mises à jour, déployer des serveurs supplémentaires en cas d’augmentation de la charge ou encore modifier la configuration de manière globale sur tous les serveurs,

qu’il s’agisse d’un simple tuning, de l’enrichissement du vecteur d’identité ou de l’ajout d’un partenaire au cercle de confiance de la fédération.

Un autre avantage d'Ansible est la possibilité de structurer les playbooks en différents rôles modulaires et réutilisables. Dans le cas du playbook Shibboleth IdP, par exemple, un rôle installe la JVM, un autre a la charge de l’installation de Jetty, un troisième déploie Shibboleth IdP à partir du package officiel et la configuration elle-même est à la charge d’un dernier rôle.

Cette architecture modulaire permet de capitaliser sur les développements existants, facilite la maintenance, les tests et l’adaptation à différents environnements.

3.3.2 Utilisation de Git

La diffusion des packages est un autre point important pour fluidifier le cycle de vie de l'infrastructure. Afin de faciliter la diffusion et le suivi de versions, les playbooks Shibboleth IdP et SP sont versionnés sur un dépôt Git. Les mises à jour de playbooks sont donc intégrées dans le cycle de développement standard des applications, avec les mêmes outils. Ce mode de fonctionnement présente également l’avantage de faciliter la diffusion de mises à jour.

Cependant, GIT n'est pas le plus approprié pour stocker des binaires, comme les archives d'installation de Jetty ou de Shibboleth IdP. Ces archives doivent donc être déposées sur un dépôt spécifique dédié et récupérées au besoin par les playbooks.

3.4 Paramétrage et optimisations

Le choix de Shibboleth devait être confirmé par plusieurs phases de tests techniques afin de valider que la volumétrie finale de plus de 30 millions de comptes ainsi que les pics de connexions pouvant être concentrés sur des périodes de temps assez courtes pourraient être supportés.

Les premiers tests, qui ont été réalisés avec les paramètres par défaut sur un environnement simplifié, nous ont assez rapidement permis d’atteindre le niveau de charge maximal supporté par le produit précédent avec environ 20 connexions par seconde. Ils ont également montré la nécessité de réaliser un tuning sur les différents composants pour pouvoir dépasser ce seuil.

Les phases de tests suivantes ont été exécutées sur une plateforme virtualisée plus proche des standards de production, avec des serveurs Shibboleth IdP / SP équipés de 2vCPU et 4Go de ram, doublés pour la répartition de charge. Un scénario d’authentification simple en mode SP initiated était utilisé pour simuler la charge et une volumétrie se rapprochant de celle attendue en production était présente sur la base de données et l’annuaire ldap.

Alors que la charge appliquée sur les serveurs Shibboleth augmentait progressivement, plusieurs comportements sont apparus :

- Tout d’abord sur les Shibboleth IdP, avec une forte sollicitation des CPU qui limitait la montée en charge. Cette saturation a pu être atténuée en modifiant des paramètres de configuration dans Jetty ainsi qu’en allégeant les pages de login et en déplaçant les ressources statiques sur les apaches.

- Ensuite au niveau des Shibboleth SP, où une limitation du nombre de connexions a été identifiée. Après analyse, il s'avéra que Shibboleth n'était pas directement en cause mais que la configuration d'apache 2.4 par défaut en RHEL6 n'était pas adaptée. Le passage du module MPM en mode worker avec un paramétrage correspondant à la charge attendue nous a permis de lever cette limitation.
- Et finalement la présence croissante d'erreurs de sessions invalides, dont le nombre d'occurrences augmentait directement avec la charge. L'examen des différents composants nous a permis de remonter jusqu'au Load Balancer matériel utilisé jusqu'ici, qui ne semblait pas parvenir à maintenir correctement la persistance des routes utilisées (sticky ip). Ce comportement provoquait des erreurs sur sessions non trouvées quand l'utilisateur était ré-orienté vers un autre IdP / SP que celui où il avait été dirigé à l'origine. Le boîtier a été finalement conservé et utilisé uniquement pour le déchiffrement SSL, la répartition de charge étant alors confiée aux Apaches via le mode proxy balancer.

Les derniers tests réalisés avec ces modifications ont permis d'atteindre une charge stabilisée se rapprochant des 80 connexions par seconde sur cette architecture, ce qui représente plus de 4 fois celle supportée précédemment.

4 Conclusion et perspectives

Les hypothèses que nous avions faites sur la pertinence du couple Shibboleth IdP / SP (compatibilité avec l'existant, extensible, souplesse des règles d'habilitations, tenue à la charge, industrialisable) se sont confirmées. La solution est aujourd'hui déployée en production, avec plus de 200 000 visiteurs différents par jour pour le dispositif national EduConnect, qui deviendra à terme le point d'entrée unique de tous les parents et les élèves vers tous les services éducatifs portés par l'Éducation Nationale ou par des tiers, puis celui des personnels de l'Éducation Nationale. Les fournisseurs de service sont progressivement migrés vers Shibboleth SP et tous les nouveaux besoins de fédération sont adressés avec ces produits.

Il reste évidemment de nombreux axes d'amélioration. Par exemple, l'interrogation directe des annuaires et des bases de données par Shibboleth IdP montre certaines limites à l'usage, comme la difficulté de faire évoluer les requêtes SQL lors de mises à jour de la structure de la base de données. Une évolution envisagée est d'utiliser le nouveau connecteur de shibboleth IdP 3.4 permettant de récupérer les attributs au travers d'API REST, ce qui diminuerait l'adhérence et faciliterait la résolution des problèmes. Par ailleurs, le support d'OpenIdConnect et d'OAuth est aujourd'hui une nécessité. CAS, que nous n'avions pas considéré comme assez mature à l'époque de nos premiers travaux, apparaît aujourd'hui comme une solution qui pourrait être plus adaptée à certains cas d'usages, comme celui d'un hub de fédération multi-protocolaire ou d'un serveur d'autorisation OAUTH.