esup portail

15 OCT. 2018 | Université PARIS Descartes

ESUP-DAYS 26

CAS 5.2 – Retour d'expérience Sorbonne Université

LUDOVIC AUXEPAULES

SORBONNE UNIVERSITÉ

Contexte général

Création de Sorbonne Université au 1er Janvier 2018

- **55300 étudiants** et **6 700 enseignants-chercheurs et chercheurs** et **4 900 personnels administratifs et techniques** issus de l'UPMC (« Paris 6 ») et Paris-Sorbonne (« Paris 4 »)
- Organisation autour de 3 facultés : Sciences & Ingénierie, Médecine et Lettres
- Certains services centraux restent communs au niveau universitaire (e.g. DSI)

« SI » avant le 1er Janvier 2018 dans les deux universités

- Plusieurs serveurs d'annuaire « centralisés » LDAP (1 UPMC + 2 PS)
  - Fonctionnement, structures, identifiants, mails, attributs de personnes gérés différemment
- Plusieurs serveurs CAS (1 UPMC + 1 PS, tout les deux en CAS v3.4.x)
- Plusieurs IdP dans la Fédération d'identités Renater (1 UPMC + 2 PS en v2.4.x)

Contexte général

➔ Une première étape : mise en place d'un annuaire des personnes dit « chapeau » pour Sorbonne Université

- **Alimentation chaque nuit à partir de l'annuaire de l'UPMC et des annuaires de Paris-Sorbonne**, avec un jeu minimal d'attributs supann
- Les nouveaux éléments générés dans cet annuaire (e.g. alias de mail) ne peuvent pas être changés

➔ Gestion de l'authentification pour les premiers applicatifs et ensuite intégration « au fil de l'eau »

- Mise en place d'un **nouveau serveur CAS** basé sur cet annuaire « chapeau »
- **Intégration d'un IdP** à la fédération d'identités Renater (pas encore en place)

Des nouveaux identifiants

Identifiant de connexion (*supannAliasLogin*)

- **Login préexistant** de chaque usager. Identifiant connu de l'usager final

Nouvel identifiant interne unique (*uid*)

- Utilisation des règles de Paris-Sorbonne (**une lettre suivie de 7 chiffres**)
- Identifiant connu des applicatifs

Nouvel ePPN (*eduPersonPrincipalName*) pour les fédérations

- **uid@sorbonne-universite.fr**

Pour la rétrocompatibilité et l'accompagnement au changement, le ***supannAliasLogin*** ou **l'*uid*** peuvent être fournis comme identifiant par CAS en fonction du service configuré

Méthode suivie pour le nouveau CAS dédié à Sorbonne Université

Utilisation **de l'overlay Maven d'Apereo** pour le serveur CAS

- https://github.com/apereo/cas-overlay-template/tree/5.2

« Suivi » du **guide de « The New School »** de David Curry

- https://dacurry-tns.github.io/deploying-apereo-cas/introduction_overview.html
- https://dacurry-tns.github.io/deploying-apereo-cas/pdf/deploying-apereo-cas.pdf

Consultation de la documentation CAS

- https://apereo.github.io/cas/5.2.x/index.html

Consultation des blogs d'Apereo sur CAS pour certains éléments de configuration

- https://apereo.github.io/

Consultation des messages sur la liste de diffusion CAS d'Apereo

- cas-user@apereo.org

Prise en compte des retours de la communauté Esup lors des précédents **GT Esup-Authentification** (Paris 2, Paris 1, AMU…)

Fonctionnalités mises en place

Support et utilisation des protocoles
CASv2, CASv3, SAML1.1 et SAML2

Stockage des tickets (Tickets Registry) avec **Memcached**

Authentification et attributs LDAP

Gestion des services avec le **Json Services Registry**

Fonctionnement en mode IdP (1<-->1)

Tests d'intégration dans les fédérations d'Identités

Throttling Authentication Attempts

Fonctionnalités mises en place

Authentification forte (mfa avec U2F)

Authentification Surrogate (« se connecter en tant que »)

Personnalisation de l'interface utilisateur de CAS aux « couleurs de Sorbonne Université »

TraceMe AGIMUS (pour les logs améliorés seulement)

Dashboard CAS configuré « a minima »

Mise en production de CAS à Sorbonne Université

- **5.2.3 en mars 2018**
- 5.2.6 en juillet 2018
- 5.2.7 en septembre 2018
- 5.2.8 en octobre 2018

Fichiers modifiés ou ajoutés dans l'overlay Maven de CAS server

pom.xml
etc/cas/config/cas.properties
etc/cas/config/log4j2.xml

Fichiers d'internationalisation de CAS
src/main/resources/messages.properties
src/main/resources/messages_fr.properties

Configuration en mode IdP
etc/cas/config/saml/

etc/cas/config/saml/certs/
etc/cas/config/saml/metada-backups
etc/cas/config/saml/sps

Services déclarés en Json
etc/cas/config/services/

Fichier utilisé pour le MFA U2F
etc/cas/config/u2fdevices.json

Scripts de déploiements
etc/cas/scripts/cassrv-install.sh
etc/cas/scripts/cassrv-install.sh

Personnalisation du thème
src/main/resources/favicon.ico
 # SASS et CSS
src/main/resources/static/
src/main/resources/su.properties
 # Vues HTML
src/main/resources/templates/

Administrateurs autorisés à se connecter au CAS Dashboard
etc/cas/config/adminusers.properties

Utilisateurs autorisés à se connecter au CAS Management (on le vide si on ne l'utilise)
src/main/resources/user-details.properties

Json Services Registry

Exemple Limesurvey en protocole CASv3

/etc/cas/config/services/limesurveySU-5.json

```
{  "@class" : "org.apereo.cas.services.RegexRegisteredService",
  "serviceId" : "https:\/\/(XXXX|YYYY)([.]ent)?([.]upmc|[.]sorbonne-universite)[.]fr(\\z|\/.*?)$",
  "name" : "Limesurvey SU",
  "id" : 5,
  "description" : "Authentification du service de formulaires (enquêtes, sondages, rendez-vous, satisfaction..) de Sorbonne Université basé sur Limesurvey",
  "proxyPolicy" : {
    "@class" : "org.apereo.cas.services.RefuseRegisteredServiceProxyPolicy"
  },
  "attributeReleasePolicy" : {
    "@class" : "org.apereo.cas.services.ReturnAllowedAttributeReleasePolicy",
    "allowedAttributes" : [ "java.util.ArrayList", ["cn", "displayName", "givenName", "mail", "sn", "uid", "supannAliasLogin" ] ]
  },
  "usernameAttributeProvider" : {
    "@class" : "org.apereo.cas.services.PrincipalAttributeRegisteredServiceUsernameProvider",
    "usernameAttribute" : "supannAliasLogin",
    "canonicalizationMode" : "LOWER"
  },
  "evaluationOrder" : 5
}
```

Attention au . dans les regex.
Préférer
[.] ou \\. à .
et
.*? à .*

Refus du mode Proxy

Liste des attributs retournés par CAS à l'authentification

Ancienne application de l'UPMC
→ Le **supannAliasLogin** correspond ici à l'ancien uid présent dans les annuaires de P6 et P4 : il est retourné ici à l'applicatif pour que les comptes des usagers soient les mêmes que précédemment

Json Services Registry

Exemple EzProxy en protocole CASv3

/etc/cas/config/services/AccesDistantSU-7.json


```
{"@class" : "org.apereo.cas.services.RegexRegisteredService",
  "serviceId" : "^https:\/\/([a-zA-Z0-9.-_]*?[.])?XXXXX(-test)?[.]sorbonne-universite[.]fr(\\z|\/.*?)$",
  "name" : "Acces distant Sorbonne Universite",
  "id" : 7,
  "description" : "Accès authentifiés aux revues, ressources documentaires de Sorbonne Université (via EZproxy)",
  "proxyPolicy" : {
    "@class" : "org.apereo.cas.services.RefuseRegisteredServiceProxyPolicy"
  },
  "attributeReleasePolicy" : {
    "@class" : "org.apereo.cas.services.ReturnAllowedAttributeReleasePolicy",
    "allowedAttributes" : [ "java.util.ArrayList", ["cn", "displayName", "givenName", "mail", "sn", "uid", "supannAliasLogin",
"eduPersonAffiliation", "eduPersonPrimaryAffiliation", "supannEntiteAffectation", "supannRefId" ] ]
  },
  "usernameAttributeProvider" : {
    "@class" : "org.apereo.cas.services.PrincipalAttributeRegisteredServiceUsernameProvider",
    "usernameAttribute" : "uid",
    "canonicalizationMode" : "LOWER"
  },
  "evaluationOrder" : 7
}
```


Nouvelle application de Sorbonne Université
→ Utilisation du « nouvel attribut uid » comme identifiant interne à l'application

Json Services Registry

Exemple de test avec uPortal et le mode Proxy activé

/etc/cas/config/services/ENTSU-1.json

```
{  "@class" : "org.apereo.cas.services.RegexRegisteredService",
  "serviceId" : "^https://XXXX[.]sorbonne-universite[.]fr/uPortal(\\z|/.*?)$",
  "name" : "ENT Sorbonne Université",
  "id" : 1,
  "description" : "Authentification de l'ENT Sorbonne Université",
  "proxyPolicy" : {
    "@class" : "org.apereo.cas.services.RegexMatchingRegisteredServiceProxyPolicy",
    "pattern" : "^https://XXXX.sorbonne-universite.fr/uPortal/CasProxyServlet"
  },
  "attributeReleasePolicy" : {
    "@class" : "org.apereo.cas.services.ReturnAllowedAttributeReleasePolicy",
    "authorizedToReleaseProxyGrantingTicket" : true
  },
  "usernameAttributeProvider" : {
    "@class" : "org.apereo.cas.services.PrincipalAttributeRegisteredServiceUsernameProvider",
    "usernameAttribute" : "uid",
    "canonicalizationMode" : "LOWER"
  },
  "evaluationOrder" : 1
}
```

Autorisation du mode Proxy

Json Services Registry

Exemple SAML2 avec CAS en mode IdP avec un SP (1<-->1)

Dans /etc/cas/config/services/TestPanopto-10004.json

```
{  "@class" : "org.apereo.cas.support.saml.services.SamlRegisteredService",
  "serviceId" : "https://XXXXX.panopto.eu/",
  "name" : "Test Panopto", "id" : 10004,  "description" : "Test d'authentification Panopto en SAML2 (mode IdP de CAS)",
  "proxyPolicy" : { "@class" : "org.apereo.cas.services.RefuseRegisteredServiceProxyPolicy"  },
  "metadataLocation" : "/etc/cas/config/saml/sps/PanoptoSPMetadata_UPMC.xml",
  "metadataMaxValidity" : 0,
  "metadataSignatureLocation" : "/etc/cas/config/saml/sps/PanoptoCloudSAML2016.crt",
  "metadataExpirationDuration" : "PT60M",
  "signAssertions" : true,
  "signResponses" : true,
  "encryptAssertions" : true,
  "signingCredentialType" : "X509",
  "attributeReleasePolicy" : {
    "@class" : "org.apereo.cas.services.ReturnAllowedAttributeReleasePolicy",
    "allowedAttributes" : [ "java.util.ArrayList", ["givenName", "mail", "sn", "eduPersonPrincipalName"] ]  },
  "attributeFriendlyNames": {
    "@class": "java.util.HashMap",
    "urn:oid:1.3.6.1.4.1.5923.1.1.1.6": "eduPersonPrincipalName", "urn:oid:2.5.4.42": "givenName",
    "urn:oid:2.5.4.4": "sn", "urn:oid:0.9.2342.19200300.100.1.3": "mail", },
  "evaluationOrder" : 10004 }
```

Correspond exactement l'Entity-ID du SP

Non fonctionnel en CAS 5.2
Utilisable seulement à partir de CAS 5.3

Fido U2F - « 2nd Facteur Universel »

Standard d'authentification libre visant à renforcer et à simplifier l'authentification double facteurs en utilisant des périphériques USB ou NFC

Développé par Google, Yubico et NXP, le standard est géré par la FIDO Alliance

Support d'U2F

- Nativement par Google Chrome depuis la version 38 et Opera depuis la version 40
- Mozilla Firefox depuis la version 57 en activant une option dans *about:config*
  Mettre ***security.webauth.u2f à true***
- Mozilla Firefox ESR v52 avec une extension (*U2F Support Add-on*)
- Pas de support pour l'instant sur Edge et aucun support dans Internet Explorer

Plusieurs étapes

- **Le premier facteur d'authentification** (e.g. identifiant + mot de passe)
- **L'enregistrement du périphérique** compatible U2F (*registration*) : une seule fois
- **L'authentification du périphérique** compatible U2F (*authentication*) : à chaque authentification

Fido U2F – Clés compatibles testées

« Keydo » de Neowave

- Entreprise française participant à « France Cyber Sécurité »
- Nécessité de « débrancher/rebrancher » à chaque authentification

ePass FIDO / ePass FIDO –NFC de Feitan

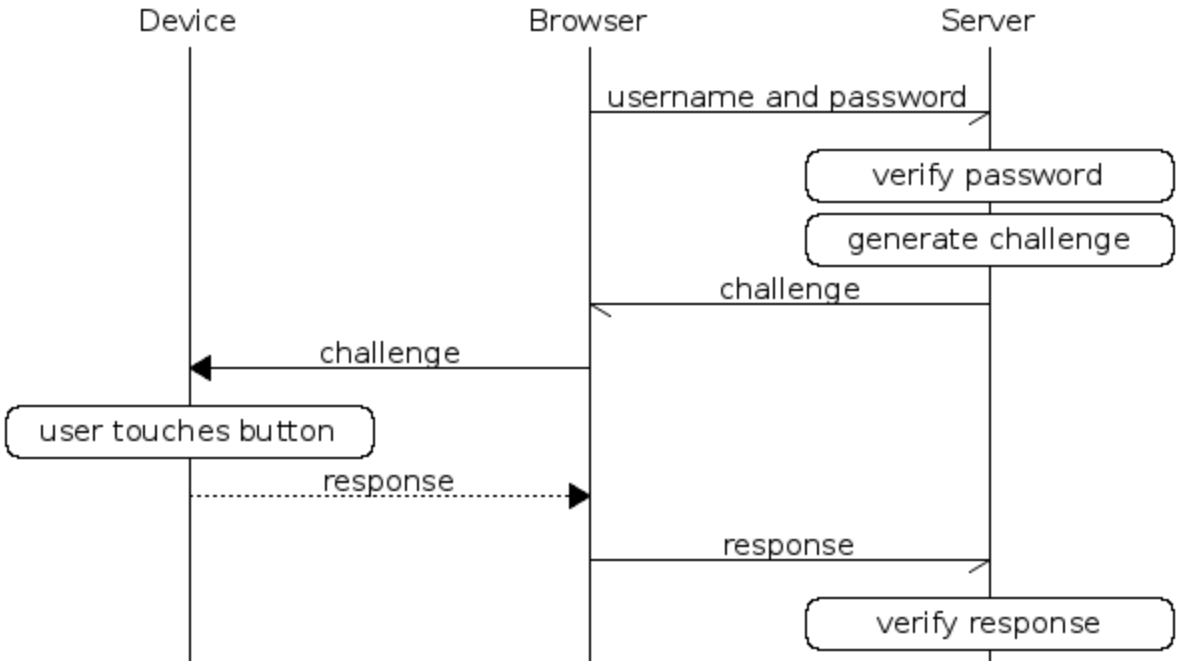
- Capteur de présence sur la clé qu'il faut toucher à chaque authentification lorsqu'elle est branchée en USB
- Gestion d'autres protocoles en plus d'U2F

« Yubikey » de Yubico

- Leader du marché et inventeur du protocole FIDO U2F
- Capteur de présence sur la clé qu'il faut toucher à chaque authentification lorsqu'elle est branchée en USB
- Gestion d'autres protocoles en plus d'U2F pour les modèles plus évolués

Fido U2F – principes

Utilisation de la cryptographie asymétrique (clé publique / clé privée) entre un serveur et un périphérique compatible U2F



Fido U2F – principes

Autre avantages

- Clé privée jamais rendue publique et signature se faisant via des méthodes sécurisées, certifiées
- Utilisation de paires de clés spécifiques à l'application
- Protection contre le phishing et le "Man in the Middle"
- Certification des périphériques
- Connexion en un seul geste
- Un seul périphérique pour plusieurs applications

Documentations

- Universal 2nd Factor : https://www.slideshare.net/FIDOAlliance/fido-u2f-10-specs-overview-and-insights
- Specifications Overview : https://fidoalliance.org/specifications/overview/
- FIDO U2F Cheat Sheet de Neowave : http://www.neowave.fr/pdfs/FIDO-U2F-CHEAT-SHEET.pdf
- Détails du protocole : https://developers.yubico.com/U2F/Protocol_details/Overview.html

Fido U2F mfa avec CAS

Dans cas.properties

U2F - Fido Universal Authentication

cas.authn.mfa.u2f.expireRegistrations=30

cas.authn.mfa.u2f.expireRegistrationsTimeUnit=SECONDS

cas.authn.mfa.u2f.expireDevices=600

cas.authn.mfa.u2f.expireDevicesTimeUnit=DAYS

FIDO U2F JSON

cas.authn.mfa.u2f.json.location=file:/etc/cas/config/u2fdevices.json

FIDO U2F Cleaner

cas.authn.mfa.u2f.cleaner.schedule.enabled=true

Documentation

https://apereo.github.io/cas/5.2.x/installation/FIDO-U2F-Authentication.html

Fido U2F mfa avec CAS

Exemple de service utilisant Apache mod_auth_cas

/etc/cas/config/services/Portail1ApacheSecuredByCASandU2f-201700831132700.json

```
{  "@class" : "org.apereo.cas.services.RegexRegisteredService",
  "serviceId" : "^https://portail1[.]ent[.]upmc[.]fr/secured-u2f(\\z|/.*?)$",
  "name" : "Portail1 : Apache Secured By CAS and U2f",
  "id" : 201700831132700,
  "description" : "CAS development Apache mod_auth_cas server with username/password and U2f MFA protection",
  "attributeReleasePolicy" : {
    "@class" : "org.apereo.cas.services.ReturnAllAttributeReleasePolicy"
  },
  "multifactorPolicy" : {
    "@class" : "org.apereo.cas.services.DefaultRegisteredServiceMultifactorPolicy",
    "multifactorAuthenticationProviders" : [ "java.util.LinkedHashSet", [ "mfa-u2f" ] ]
  },
  "usernameAttributeProvider" : {
    "@class" : "org.apereo.cas.services.PrincipalAttributeRegisteredServiceUsernameProvider",
    "usernameAttribute" : "uid",
    "canonicalizationMode" : "LOWER"
  },
  "evaluationOrder" : 1200
}
```

Fido U2F MFA avec CAS

Corrections de certains bugs qui rendaient non fonctionnel l'usage en production du MFA avec U2F dans CAS

- Erreur dans la gestion de l'expiration des périphériques U2F
  - https://github.com/apereo/cas/pull/3230
- Ecrasement de toutes les clés existantes à l'ajout d'une nouvelle
  - https://github.com/apereo/cas/pull/3286 https://github.com/apereo/cas/pull/3287

Quelques limitations au niveau de l'utilisation d'U2F avec CAS

- Les pages d'enregistrement et d'authentification ont un délai d'expiration mais restent affichées
- Les textes liés à U2F ne peuvent pas être traduits sans modifier directement les pages html en CAS 5.2
  - L'internationalisation est gérée correctement pour U2F en CAS 5.3.x
- Les usagers ne peuvent pas gérer leurs périphériques U2F associés à leur compte
- Les administrateurs n'ont pas d'interface de gestion des périphériques U2F des usagers

Authentification Surrogate

S’authentifier au nom d’un autre utilisateur

Dans pom.xml

```
<dependencies>
....
    <dependency>
        <groupId>org.apereo.cas</groupId>
        <artifactId>cas-server-support-surrogate-webflow</artifactId>
        <version>${cas.version}</version>
     </dependency>
...
</dependencies>
```

Documentation

https://apereo.github.io/cas/5.2.x/installation/Surrogate-Authentication.html

https://apereo.github.io/2018/05/07/cas-impersonation-authn/

Attention la configuration des comptes autorisés en Json (JSON Surrogate Accounts) ne semble pas fonctionner

La configuration avec *Static Surrogate Accounts* depuis le fichier cas.properties fonctionne

Authentification Surrogate

S'authentifier au nom d'un autre utilisateur

Dans cas.properties

Surrogate Authentication (Authenticate on behalf of another user)

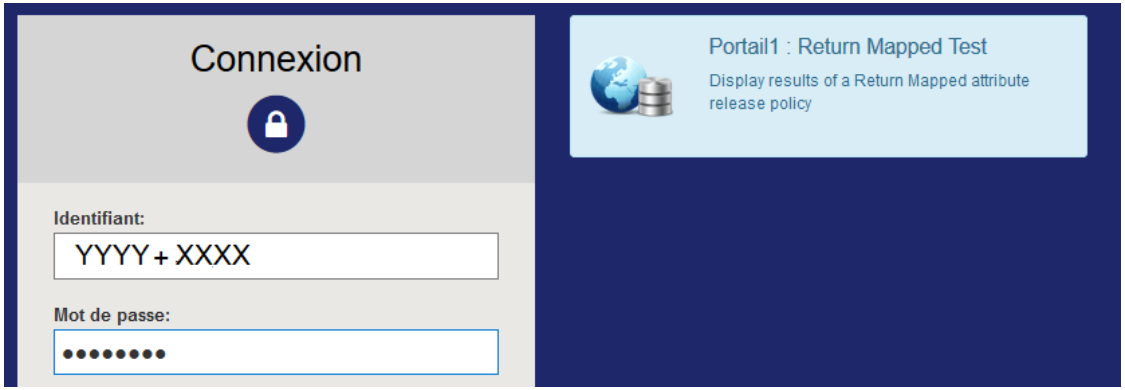
 # JSON Surrogate Accounts

#cas.authn.surrogate.json.location=file:/etc/cas/config/surrogates.json

 # Static Surrogate Accounts

cas.authn.surrogate.simple.surrogates.XXXX=YYYY,ZZZZ

XXXX pourra se connecter au nom de YYYY ou ZZZZ
en tapant YYYY+XXXX dans la mire de connexion de CAS



Des attributs supplémentaires sont fournis par CAS au client pour indiquer qui (XXXX) s'est connecté en tant que YYYY :

surrogateEnabled = true

surrogatePrincipal = XXXX

En plus du REMOTE_USER (correspondant à l'uid ou le supannAliasLogin) qui est YYYY

Modifications du thème de CAS

Modification directe des couleurs dans les fichiers Sass du projet CAS

webapp/resources/static/sass/authnEvents.scss
webapp/resources/static/sass/components/_buttons.scss
webapp/resources/static/sass/components/_cas.scss
webapp/resources/static/sass/components/_dashboard-view.scss
webapp/resources/static/sass/components/_login.scss
webapp/resources/static/sass/components/_statistics-view.scss
webapp/resources/static/sass/components/_tooltips.scss
webapp/resources/static/sass/components/_variables.scss
webapp/resources/static/sass/loggingDashboard.scss
webapp/resources/static/sass/partials/_admin-panels.scss
webapp/resources/static/sass/partials/_base.scss
webapp/resources/static/sass/partials/_fonts.scss
webapp/resources/static/sass/ssosessions.scss
webapp/resources/static/sass/trustedDevices.scss

Pour réaliser des modifications à chaud sans relancer Tomcat (à tester)

- **spring.thymeleaf.cache=false**

Autre méthode « plus propre » et complète

- https://apereo.github.io/2018/06/10/cas-userinterface-customizations/

Modifications du thème de CAS

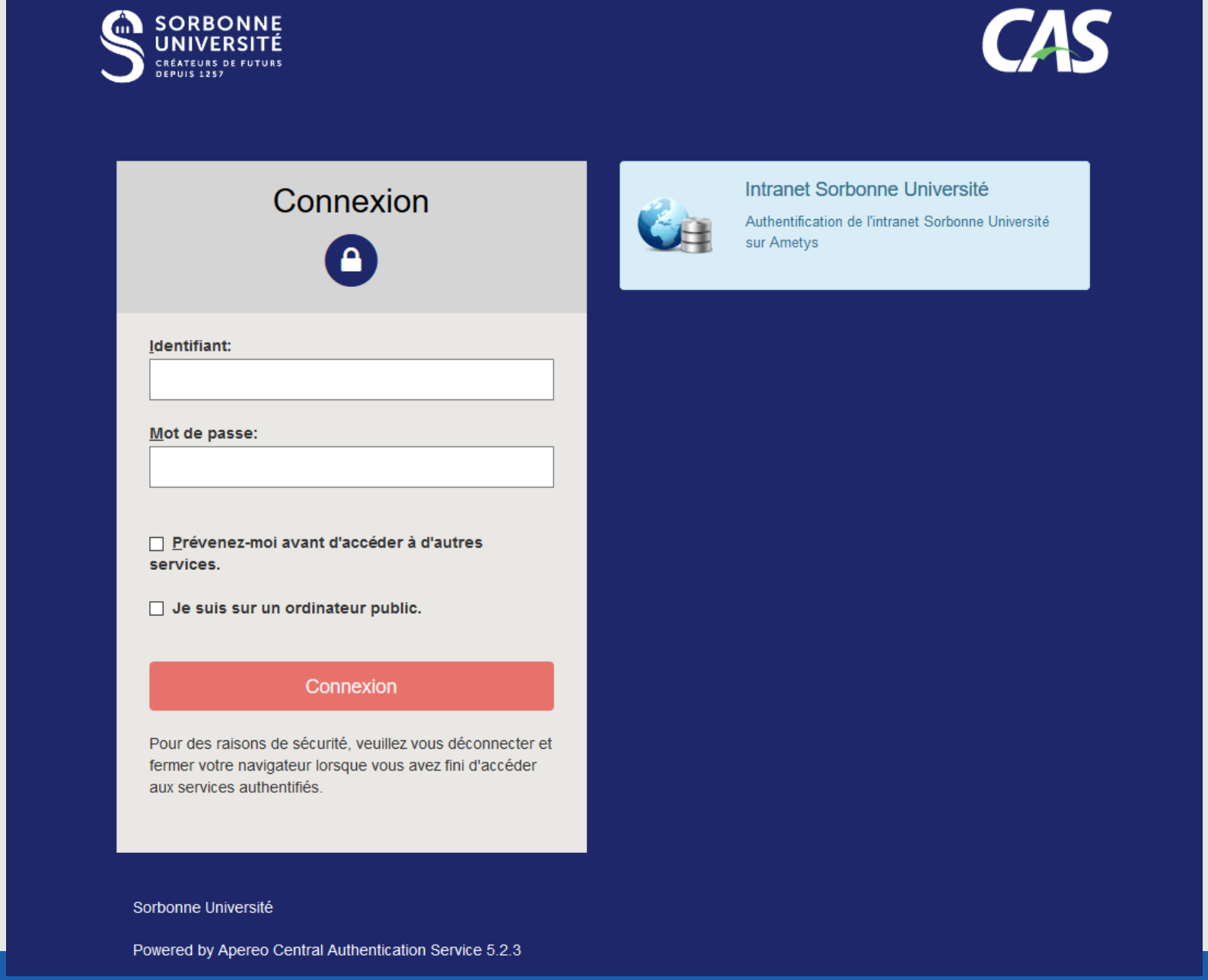
Fichiers et répertoires dans l'overlay

Personnalisation du thème
src/main/resources/favicon.ico

 # SCSS et CSS
src/main/resources/static/
src/main/resources/static/favicon.ico
src/main/resources/static/css
src/main/resources/static/scss
src/main/resources/static/themes/apereo
src/main/resources/static/themes/su

 # Fichiers de configuration du thème SU
src/main/resources/su.properties
standard.custom.css.file=/themes/su/css/cas.css
cas.javascript.file=/themes/su/js/cas.js
admin.custom.css.file=/themes/su/css/admin.css

 # Vues HTML
src/main/resources/templates/
src/main/resources/templates/casGenericSuccessView.html
src/main/resources/templates/casLoginView.html
src/main/resources/templates/fragments
src/main/resources/templates/fragments/bottom.html
src/main/resources/templates/fragments/footer.html
src/main/resources/templates/fragments/head.html
src/main/resources/templates/fragments/loginform.html
src/main/resources/templates/fragments/logo.html
src/main/resources/templates/layout.html



APERÇU DE LA MIRE DE CONNEXION DU NOUVEAU CAS DE SORBONNE UNIVERSITÉ

Déploiement et tests

Exemple de liste de commandes

Redéploiement complet de CAS dans le tomcat et relance du serveur

Assemblage des différentes modules du serveur CAS à partir des sources si on les a modifiés

cd /opt/git/cas

./gradlew build install --parallel -x test -x javadoc -x check -DskipNpmCache=true -DskipNpmLint=true -DskipNodeModulesCleanUp=true

Assemblage d'Agimus

 # Il faut penser à modifier la version du serveur CAS utilisé dans les fichiers pom.xml

cd /opt/git/cas-server-support-agimus-cookie/

mvn package install

cd /opt/git/cas-server-support-agimus-logs/

mvn package install

Assemblage et déploiement du serveur CAS

cd /opt/git/cas-overlay-template/

mvn clean package

 # Création du zip du serveur CAS

sh /opt/cas/scripts/cassrv-tarball.sh

 # Installation du serveur CAS à partir du zip et lancement du serveur Tomcat

sh /opt/cas/scripts/cassrv-install.sh

service cas status

sleep 50

tail -f /var/log/cas/cas.log

Travaux et pistes à explorer

Amélioration Idp et Intégration à la fédération d'identités Renater

Intégration de l'authentification *Risk-based*

- https://apereo.github.io/cas/development/installation/Configuring-RiskBased-Authentication.html

Gestion des protocoles Auth2, OpenId-Connect…

Stockage et réplication des sessions et des tickets avec Redis

Mise en place du « CAS Management »

Haute disponibilité et maintenance facilitée sur plusieurs serveurs CAS

Passage à CAS v5.3 : **bugs résolus, nouvelles fonctionnalités…**

Utilisation d'**Ansible** pour faciliter le déploiement et l'installation

Utilisation de **Vault** pour gérer les secrets (clés, mots de passe…)

Projet Apereo CAS

Politique de mise à jour et maintenance de CAS

Release/Branch	Version	SPM Entry *Période de patchs de sécurité*	EOL
4.1.x	4.1.10 (11/2016)	02/2017	02/2018
4.2.x	4.2.7 (11/2016)	02/2017	02/2018
5.0.x	5.0.10 (10/2017)	10/2017	10/2018
5.1.x	5.1.9 (04/2018)	05/2018	05/2019
5.2.x	5.2.8 (10/2018)	11/2018	11/2019
5.3.x	5.3.4 (10/2018)	06/2019	06/2020

- https://apereo.github.io/cas/developer/Maintenance-Policy.html
- https://github.com/apereo/cas/milestones

Projet Apereo CAS

Sortie de CAS v5.3 (depuis fin juin)

80 nouvelles fonctionnalités ; le ***CAS Management*** et le ***Command-line Shell*** ont été améliorés

- RC4 05/2018 : https://apereo.github.io/2018/05/25/530rc4-release/

Arrivée de CAS v6 dés 2019 ?

- ➡ *Java 11*, *Spring v5*, *Spring Boot v2*, *Servlet 4*…
- RC1 08/2018 : https://apereo.github.io/2018/08/03/600rc1-release/
- RC2 09/2018 : https://apereo.github.io/2018/09/14/600rc2-release/
- RC3 annoncée pour octobre : https://apereo.github.io/2018/10/19/600rc3-release/

esup portail

15 OCT. 2018 | Université PARIS Descartes

MERCI