



Assomni'Hack

VEILLE TECHNOLOGIQUE

Préparée pour le projet Balabox

SOUS-PROJET
Identification

NOVEMBRE 2022



Plan

- Objectifs
- Veille réglementaire
 - Quelle est son utilité ?
 - Données biométriques
 - Stockage du mot de passe
 - Tentative d'accès
 - Limiter la divulgation d'information sur les comptes existants
 - Collecte des données
 - La gestion des utilisateurs
 - Mot de passe administrateur
 - Conclusion
- Technologies récentes
 - Authentification intelligente
 - Authentification forte
 - Conclusion
- Applications similaires
 - Chamilo
 - Présentation rapide de la plateforme
 - Technologie et présentation du système d'identification
 - Application de la technologie et du système d'identification pour Balabox
 - Moodle
 - Présentation rapide de la plateforme
 - Technologie et présentation du système d'identification
 - Application de la technologie et du système d'identification pour Balabox

Plan

- MaDoc
 - Présentation rapide de la plateforme
 - Technologie et présentation du système d'identification
 - Application de la technologie et du système d'identification pour Balabox
- Canvas LMS
 - Présentation rapide de la plateforme
 - Technologie et présentation du système d'identification
 - Application de la technologie et du système d'identification pour Balabox
- Conclusion
- Les protocoles d'identification
 - Introduction
 - CAS
 - Oauth2
 - Services d'annuaire - Authentification HTTPS
 - CSV
 - BD
 - LDAP
- Technologie existante : la MoodleBox
- Conclusion finale

Objectifs

Cette veille technologique aura pour but de constituer une base d'informations clé pour identifier et comprendre les technologies et les applications similaires à notre projet : Balabox. Pour rappel, La Box est une solution nomade de création, de partage et de diffusion de contenus numériques à destination de l'enseignement basée sur la Moodle Box, fonctionnant avec la RaspBerry Pi 4b. Dans ce document, nous allons donc collecter des informations stratégiques en réalisant des recherches plus ou moins approfondies afin de décider du fonctionnement de l'application web Balabox, notamment de son système d'identification. Nous nous concentrerons donc, tout au long de cette veille, sur la possibilité du fonctionnement du système d'identification. Cette veille a été réalisée par Poulain Thomas, Besily Michaël ainsi que Offredi Eve-Anne.

Veille réglementaire

Quelle est son utilité ?

Dans le système d'identification, nous travaillerons énormément avec des données à caractère personnel. Ces données sont protégées par la RGPD ainsi que la Loi Informatique et Libertés. Tout d'abord, que sont les données à caractère personnel : cela correspond à toute information se rapportant à une personne physique identifiée ou identifiable selon la RGPD. Cela couvre donc le nom, prénom, pseudonyme, date de naissance, adresse IP, identifiant de connexion informatique ou de cookie... Ces précédents exemples relèvent donc du système d'identification que nous développerons.

Données biométriques

Les données biométriques pouvant être utilisées aux fins d'identifier une personne de manière unique lors de l'identification sont considérées par le RGPD comme des données sensibles. Ce dernier interdit donc de recueillir ou de les utiliser sauf si la personne concernée a donné son consentement dans une démarche active, explicite et de préférence écrite, qui doit être libre, spécifique et informée.

Veille réglementaire

Si nous utilisons les données biométriques comme méthode d'identification dans le projet Balabox, nous devons donc mettre en place, lors de la première connexion, un formulaire de consentement visant à récupérer l'accord de l'utilisateur. Si ce dernier accepte que Balabox recueille ses données biométriques, il aura alors l'occasion de se connecter avec par la suite. Si ce n'est pas le cas, ce type de connexion ne sera pas activé.

Stockage du mot de passe

Le mot de passe de l'utilisateur ne doit jamais être stocké en clair. Il est recommandé qu'il soit transformé au moyen d'une fonction cryptographique non-réversible et sûr, intégrant l'utilisation d'un sel ou d'une clé. Le sel ou la clé doit être généré au moyen d'un générateur de nombres pseudo-aléatoires cryptographiquement sûr (c'est-à-dire basé sur un algorithme public réputé fort dont la mise en œuvre logicielle est exempte de vulnérabilité connue), et ne peut pas être stocké dans le même espace de stockage que l'élément de vérification du mot de passe. Nous devons donc stocker les mots de passe sous forme de hachage. Il existe aujourd'hui des fonctions spécialisées qui permettent de répondre à ce besoin, comme Scrypt, Argon2, Yescrypt, Balloon ou Bcrypt : des moyens que nous pourrions utiliser, si besoin, dans le projet Balabox.

Veille réglementaire

Si certaines fonctionnalités ne servant qu'à améliorer l'expérience utilisateur sont implémentées, l'utilisateur doit pouvoir choisir d'utiliser ou non cette dernière.

La gestion des utilisateurs

Nous utiliserons des identifiants uniques et propres à chaque individu ou matériel quand la connexion se réfère à l'appareil et est anonyme. Les personnes n'étant pas administrateurs n'auront pas accès aux fonctionnalités de ces derniers.

Mots de passe administrateur

Les mots de passe administrateur devront être spécifiques. Leur forme devrait donc être différente de celle des comptes utilisateurs classiques pour les élèves. Si cela est possible dans le projet, l'authentification forte pourra être priorisée. Il est recommandé d'y associer une politique de mot de passe fort (10 à 20 caractères ou multifacteur) et de limiter à son plus strict nécessaire le nombre de personnes ayant connaissance de celui-ci. Ici, les administrateurs, à la demande de la cliente, seront les professeurs du collège.

Veille réglementaire

Tentative d'accès

Le nombre de tentatives d'accès devra être limité à au plus de 3. Cela permet de mieux sécuriser les authentications.

Limiter la divulgation d'information sur les comptes existants

Les messages d'erreurs d'authentification devront être généralisés comme par exemple avec le message suivant : "l'identifiant ou le mot de passe est inconnu".

Collecte des données

Concernant la collecte des données, nous devons nous baser sur le strict minimum. En effet, il est d'usage d'éviter la collecte de données non-nécessaires ainsi que de traiter et stocker les données de façon à réduire leur précision comme par exemple le fait de ne stocker que l'année de naissance d'une date de naissance complète si nous n'avons besoin que de cette partie. Les données collectées ont une durée de conservation en fonction de la finalité du traitement et des obligations légales ou réglementaires relative à leur conservation et ne doivent jamais être utilisées à d'autres fins. Si certaines fonctionnalités ne servant qu'à améliorer l'expérience utilisateur sont implémentées, l'utilisateur doit pouvoir choisir d'utiliser ou non cette dernière. Dans le projet Balabox, la collecte des données sera moindre. En effet, nous n'avons réellement besoin que des noms et prénoms des élèves.

Veille réglementaire

Conclusion

Dans cette partie, nous venons de voir différentes règles du RGPD concernant la sécurité et la collecte des données. Cependant, rappelons que le réseau de la Balabox ne donne pas d'accès à Internet. Il est impossible de s'y connecter à distance, le seul problème de sécurité se posant nous concernant, est la communication entre les téléphones dans une seule et même valise. Quelle est la limite entre violation de la loi et éthique ? Est-il acceptable qu'un élève puisse récupérer les fichiers audios et vidéos de son camarades qui lui a un autre téléphone ? Ce sont de nombreuses questions que nous, étudiants, nous nous posons. Nous avons décidé de ne pas autoriser cela. Pour se faire, la gestion des mots de passe sera sécurisée de par des mots de passe différents pour chaque élève, donné par le professeur. De plus, partant du principe que les noms et prénoms des élèves sont des données à caractère personnel, ces dernières devront être protégées. Certaines règles de le RGPD citées ci-dessus seront alors appliquées. Les mots de passe seront générés par un super administrateur après l'importation d'un fichier CSV pour créer une classe. Ce dernier les communiquera aux enseignants ayant la Balabox. Ces derniers distribueront un papier avec un mot de passe unique à chaque élève. Si l'élève oublie son mot de passe, l'enseignant peut le réinitialiser et le donner à l'élève. Ce dernier le modifie directement, lors de la première connexion, cela rend donc le mot de passe provisoire.

Technologies récentes

L'authentification intelligente

Avec l'augmentation de la cybercriminalité et de la fraude, de nouveaux moyens d'authentification sont recherchés notamment par les entreprises ayant besoin d'une sécurité optimale comme les banques. Pour répondre à ce besoin urgent, un nouveau moyen d'authentification se met en place : l'authentification intelligente. Cette dernière vise à repenser le protocole d'authentification traditionnel, muni de son identifiant et de son mot de passe. Ce nouveau moyen d'identification offre plus d'options de connexion, comme la biométrie ou les notifications push tout en exploitant les données techniques telles que la localisation de l'utilisateur, l'adresse IP ainsi que le système d'exploitation utilisé et les schémas comportementaux de connexion. Cela a pour but de personnaliser et de sécuriser davantage le processus d'identification. Ce service est piloté par une intelligence artificielle regardant tous ces critères et rapportant à l'utilisateur une connexion louche si un facteur sort de ses habitudes.

Technologies récentes

L'authentification forte

Afin de contrer cette cybercriminalité, un système d'authentification forte se met en place. Restant une simple option sur certaines applications, les certificats électroniques ou authentification à double facteur offrent plus de sécurité qu'un simple mot de passe.

Commençons par les certificats électroniques. Son fonctionnement est simple : identifier une personne physique ou morale en ayant une véritable identité électronique. Les certificats de signature électronique sont délivrés directement par le tiers de confiance de l'application qui s'engage à son tour sur l'identité du porteur du certificat électronique. En effet, il effectue de nombreuses vérifications sur l'existence des personnes morales (validation des extractions kbis, des fichiers INSEE, des gestionnaires, de la propriété des domaines). Il vérifie également l'identité du titulaire du certificat ou peut même vérifier son identité en face-à-face.

Concernant maintenant l'authentification à double facteur, elle permet au compte d'être authentifié uniquement si deux pièces d'identité distinctes sont fournies, ce qui contribue à garantir l'authenticité des personnes derrière le compte. Généralement, un mot de passe à usage unique doit être saisi en plus du mot de passe habituel de l'utilisateur. S'il est incorrect, l'authentification échouera même si le mot de passe saisi correspond au mot de passe associé au compte.

Technologies récentes

L'authentification forte

Voici un exemple du processus d'authentification :

- L'utilisateur ouvre l'application ou le site Web du service ou du système auquel il souhaite accéder. Ils sont ensuite invités à se connecter avec leurs identifiants.
- L'utilisateur saisit ses identifiants de connexion, généralement son nom d'utilisateur et son mot de passe. L'application ou le site Web confirme les détails et confirme que les détails d'authentification initiale corrects sont entrés.
- Si l'application ou le site Web n'utilise pas d'identifiants de connexion par mot de passe, il génère une clé de sécurité pour l'utilisateur. La clé sera prise en charge par l'outil d'authentification et le serveur validera la requête initiale.
- L'utilisateur est ensuite invité à soumettre un deuxième facteur d'authentification. Il s'agit généralement du facteur propriétaire, c'est-à-dire uniquement ce qu'il devrait avoir. Par exemple, l'application ou le site Web enverra un code unique à l'appareil mobile de l'utilisateur.
- L'utilisateur entre le code dans l'application ou le site Web, et si le code est approuvé, il est authentifié et accède au système.

Technologies récentes

Conclusion

Le projet Balabox repose sur un principe : une simplicité d'utilisation. Ne voulant pas perdre de temps lors des cours avec des personnes ayant des difficultés avec la technologie, le système d'identification se doit d'être simple et rapide. Malgré un grand apport en sécurité avec les solutions citées ci-dessus, un système d'identification relevant des dernières technologies à la pointe n'est pas nécessaire dans ce cas. Voyons alors, d'autres moyens d'identifications beaucoup plus accessibles et rapides dans les parties suivantes.

Applications similaires

Chamilo : Présentation du rapide du site

Chamilo est un logiciel open source de gestion de l'apprentissage et du contenu d'apprentissage, dont l'objectif est d'améliorer l'accès global à l'éducation et au savoir. Chacun est libre de télécharger et d'utiliser Chamilo, à condition d'accepter les termes de sa licence. Chamilo a deux objectifs principaux : aider l'enseignant à mieux répondre aux besoins de ses élèves et pour rendre faciles la création et l'édition de supports d'apprentissage numériques de haute qualité que l'enseignant peut non seulement créer, mais être inspiré pour développer et améliorer en permanence le contenu de ses cours. Concernant l'identification sur Chamilo, cette dernière est simple, elle se fait avec un nom d'utilisateur ainsi qu'un mot de passe comme sur l'image ci-dessous.



The screenshot shows the Chamilo login interface. At the top left is the Chamilo logo with the tagline 'E-Learning & Collaboration Software'. Below it is a blue navigation bar with 'Page d'accueil'. The main content area has a login form on the left with a language dropdown set to 'Français', input fields for 'Nom d'utilisateur' and 'Mot de passe', and a 'S'identifier' button. To the right of the form, a welcome message reads 'Bienvenue sur la plateforme pédagogique de l'UCO.' followed by instructions to connect using the provided menu and to click on the 'Mes cours' tab. At the bottom left, a 'Général' sidebar contains links for 'Bien démarrer avec Chamilo - ENSEIGNANTS' and 'CIDEF Etudiants'. At the bottom right, there is a logo for 'ALTISSIA LANGUAGE EMPOWERS PEOPLE' and a promotional message for 'UCO Languages : Formez-vous en langues avec Altissia !' with contact information.

Chamilo
E-Learning & Collaboration Software

Page d'accueil

Français

Nom d'utilisateur

Mot de passe

S'identifier

Bienvenue sur la plateforme pédagogique de l'UCO.

Connectez-vous avec vos identifiants dans le menu ci-contre.

Cliquez sur l'onglet **Mes cours** pour accéder aux espaces de cours auxquels vous êtes inscrit.

Général

Bien démarrer avec Chamilo - ENSEIGNANTS

CIDEF Etudiants

ALTISSIA
LANGUAGE EMPOWERS PEOPLE

UCO Languages : Formez-vous en langues avec Altissia !

Cliquez sur le logo et connectez-vous avec vos identifiants UCO (*n°etudiant@etud.uco.fr* et votre mot de passe habituel).

A vous de progresser à votre rythme.

Applications similaires

Chamilo : Technologie et présentation du système d'identification

Pour la connexion, les développeurs ont décidé d'utiliser PHP. Un objet session est récupéré en faisant une requête getSession() lors de la connexion de l'utilisateur. Afin d'avoir la bonne session associée à l'utilisateur connecté, il récupère l'id unique de ce dernier et associe donc la session.

Pour garder une trace de qui est identifié par quoi, Chamilo suit généralement la source d'authentification d'un utilisateur via le champ auth_source dans la table des utilisateurs. Les utilisateurs identifiés via LDAP utiliseront "ldap" (si synchronisation automatique) ou "extldap" (si premier enregistrement lors de la première connexion).

Sur Raspberry Pi, il est tout à fait possible d'installer php pour gérer la connexion de la même manière que Chamilo. Nous pourrions utiliser cette méthode de connexion pour le projet Balabox. Cependant, nous ne pourrions pas gérer la partie connexion anonyme en prenant exemple sur Chamilo. En effet, ce site web ne propose pas ce mode d'authentification, même si php permet de le faire (voir Moodle).

Applications similaires

Chamilo : Application de la technologie et du système d'identification pour Balabox

Chamilo est pleinement compatible avec le Raspberry Pi notamment sa gestion d'identification et propose même l'installation complète depuis un guide officiel : https://support.chamilo.org/projects/chamilo-18/wiki/Simplified_guide

De plus, la gestion d'identification de Chamilo a été faite en PHP et permet de comprendre assez facilement ce qu'on souhaite nous inspirer pour réussir à créer un système d'identification pour Balabox.

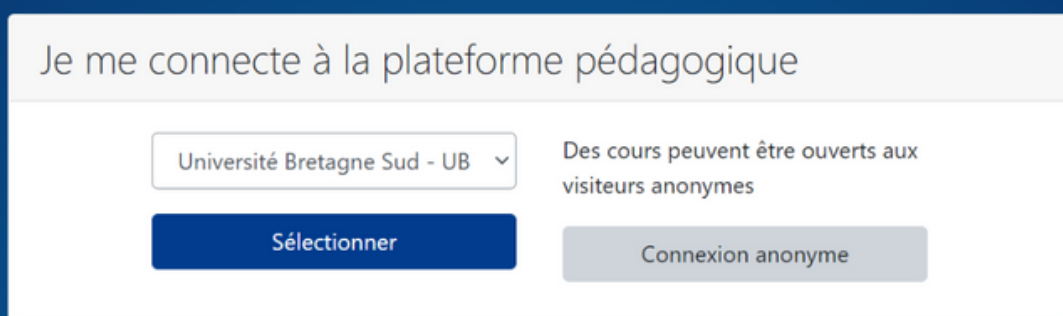
Git : <https://github.com/chamilo/chamilo-lms>

Documentation : <https://docs.chamilo.org/>

Applications similaires

Moodle : Présentation du rapide du site

Moodle est une plateforme d'apprentissage en ligne, elle permet la gestion et la mise en ligne de cours à disposition des élèves. Moodle est libre de téléchargement et doit être installé sur un serveur web, tel que Apache ou nginx, avec une base de données. Pour se connecter et accéder aux ressources, il faut se connecter à un compte (login et mot de passe), mais il est tout à fait possible de se connecter sans compte, l'utilisateur sera donc en mode anonyme.

The image shows a login interface for Moodle. It has a dark blue header bar. Below it is a white box with a light gray border. At the top of this box, the text "Je me connecte à la plateforme pédagogique" is displayed in a dark gray font. Below this text, there is a dropdown menu showing "Université Bretagne Sud - UB" with a downward arrow. To the right of the dropdown, the text "Des cours peuvent être ouverts aux visiteurs anonymes" is displayed in a smaller, gray font. Below the dropdown menu is a dark blue button with the text "Sélectionner" in white. To the right of this button is a light gray button with the text "Connexion anonyme" in dark gray.

Applications similaires

Moodle : Technologie et présentation du système d'identification

Le système de connexion de Moodle en PHP est assez simple. En effet, quand l'utilisateur arrive sur le site, `loginpage_hook()` intercepte la connexion pour rediriger l'utilisateur sur une page de connexion qui possède une vérification des critères alphanumérique pour l'identification. Ensuite, après avoir entré son login et le mot de passe, `authentication_user_login()` cherche dans la base de données le compte de l'utilisateur (ligne 12 ci-dessous). Après cela, il renvoie un objet `$user` qui possède les informations de l'utilisateur ou `false` si l'utilisateur n'a pas été trouvé, si cette objet est valide, alors la connexion a réussi, sinon, l'utilisateur est invité à remplir de nouveaux son mot de passe. Ensuite les informations sont enregistrées dans la variable global `$SESSION`

```
1. function authenticate_user_login($username, $password,
   $ignorelockout=false, &$amp;$failurereason=null, $logintoken=false) {
2.     global $CFG, $DB, $PAGE;
3.     require_once("$CFG->libdir/authlib.php");
4.
5.     if ($user = get_complete_user_data('username', $username,
   $CFG->mnet_localhost_id)) {
6.         // we have found the user
7.
8.     } else if (!empty($CFG->authloginviaemail)) {
9.         if ($email = clean_param($username, PARAM_EMAIL)) {
10.            $select = "mnethostid = :mnethostid AND LOWER(email) =
   LOWER(:email) AND deleted = 0";
11.            $params = array('mnethostid' => $CFG->mnet_localhost_id,
   'email' => $email);
12.            $users = $DB->get_records_select('user', $select, $params,
   'id', 'id', 0, 2);
13.            if (count($users) === 1) {
14.                // Use email for login only if unique.
15.                $user = reset($users);
16.                $user = get_complete_user_data('id', $user->id);
17.                $username = $user->username;
18.            }
19.            unset($users);
20.        }
21.    }
22.    ...
```

Applications similaires

Moodle : Technologie et présentation du système d'identification

Moodle utilise aussi des plugins d'authentification tiers (par exemple, CAS, SAML, Oauth2, OpenID ect). En effet, cela permet de rendre la gestion de comptes Moodle plus simple. Pour plus de précisions, la partie protocole d'identification sera là pour vous aider.

Moodle : Application de la technologie et du système d'identification pour Balabox

Pour le projet Balabox, la connexion en PHP pourrait être une bonne solution, car elle permet de gérer facilement les différents cas de connexions (anonymes et par compte). De plus, PHP possédant déjà les outils pour gérer une connexion, aucune librairie ni plugin sera à installer ce qui permettra de gagner en mémoire.

De plus, il est très accessible d'installer Moodle sur le Raspberry PI avec les différents projets créés par différents développeurs.

La synchronisation de Moodle et de la Balabox

Pour relier Moodle au login Balabox, il est possible d'utiliser le protocole Oauth2, car cela permettra d'avoir un seul compte pour les deux applications Web. De plus, les collègues n'utilisent pas de protocole CAS, mais cela reste facultatif, car ce n'est pas priorité de la cliente.

Documentation : https://docs.moodle.org/dev/Main_Page

Applications similaires

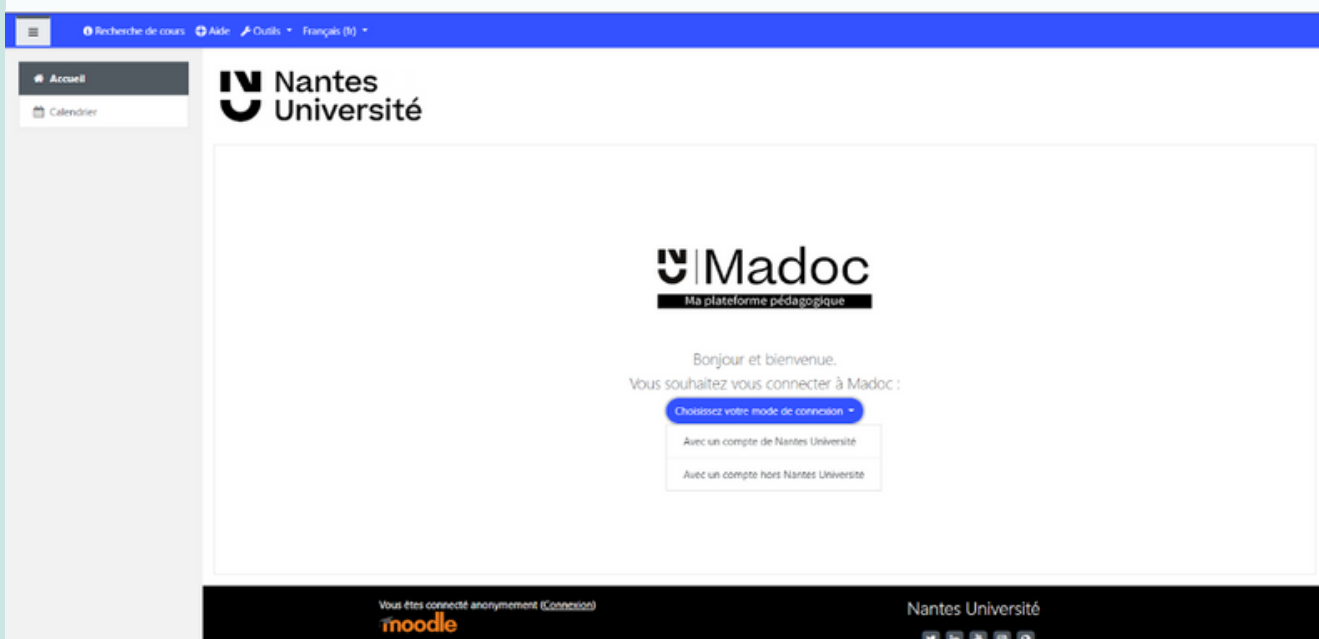
MaDoc : Présentation du rapide du site

La plateforme pédagogique Madoc permet à tout enseignant de mettre à disposition des étudiants ses contenus de cours et documents multimédias en ligne. Elle est utilisée par l'Université de Nantes. Cependant selon l'avis de plusieurs enseignants du département Informatique de l'IUT de Nantes à une accessibilité assez compliquée.

Concernant notre projet Identification, il faut savoir que Madoc a le même système d'identification que Moodle.

Nous allons dans le cas de Madoc nous basé sur l'implémentation de celui-ci à Nantes Université

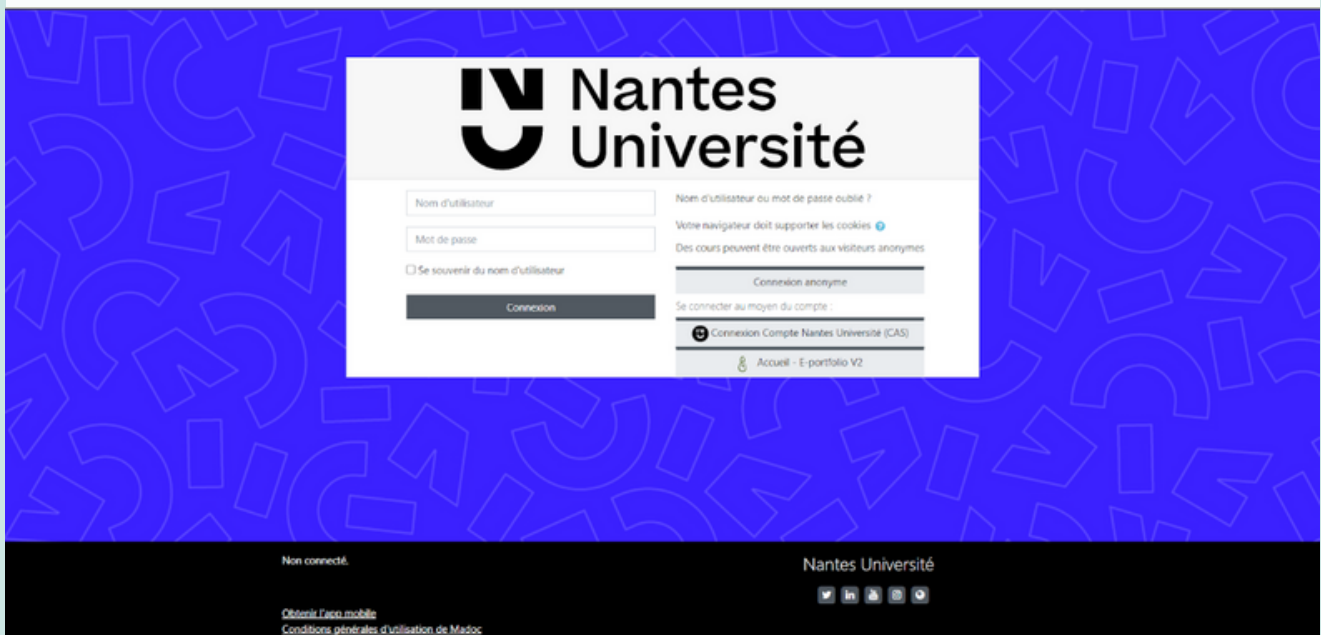
Voici la dernière version de Madoc de Nantes Université (Version : 21 Février 2022) :



Applications similaires

MaDoc : Technologie et présentation du système d'identification

Elle propose une connexion “hors Nantes Université”, c’est-à-dire une connexion depuis un compte Moodle, une connexion anonyme :



Il existe également deux sites externes, dont Nantes Université en utilisant le système d'authentification CAS(Centrale Authentification Service) tout comme Moodle, car il est implémenté de manière native.

Le Central Authentication Service (CAS) est un protocole d'authentification de connexion unique. Celle-ci est très utile dans les environnements où de nombreuses applications réseaux différents partagent un ensemble commun d'utilisateurs dans une même base de données. Celle-ci est utilisée notamment dans les universités françaises telles que Nantes, Bordeaux, l'Université Bretagne Sud.

Applications similaires

MaDoc : Application de la technologie et du système d'identification pour Balabox

Étant donné que la seule différence réside sur le front-end, Madoc pourra nous être utile pour faire des choix visuels qui sont différents de Moodle, mais qui ont été lancé à grande échelle et des améliorations qui ont été faites pour améliorer l'accessibilité de celui-ci pour les élèves. De plus, celle-ci est pleinement compatible avec le Raspberry Pi et utilise un langage de programmation (PHP) que nous connaissons déjà.

Git : <https://github.com/digirati-co-uk/madoc-platform>

Documentation : <https://docs.madoc.io/>

Applications similaires

Canvas LMS : Présentation du rapide du site

Canvas LMS est un système de gestion de l'apprentissage (LMS) basé sur le cloud qui cible les établissements d'enseignement et facilite l'apprentissage en ligne. Contrairement à Moodle et Madoc, celle-ci n'est pas très connue en France, notamment dans les établissements scolaires malgré que ce site web propose une version Open Source disponible sur GitHub.

Tout comme Moodle et Chamilo, Canvas propose différentes manières de se connecter.

Elle permet tout d'abord une connexion sécurisée en HTTPS à partir d'une adresse mail + mot de passe. Ou bien via une connexion externe comme Apple, Twitter, LinkedIn. Elle ne propose cependant pas la connexion via le protocole CAS très utile dans le cas d'une connexion globale sur les différents sites existant d'un établissement.

The image shows the login interface of the Canvas LMS. At the top left is the Canvas logo. To its right are links: 'Besoin d'un compte Canvas ? Cliquez ici, c'est gratuit !' and 'Parcourir les cours'. Below these are two input fields: 'Email' and 'Mot de passe'. Under the password field is a checkbox for 'Rester connecté' and a link for 'Mot de passe oublié ?'. A 'Se connecter' button is positioned to the right of the password field. Below the login fields is a row of social media icons: Apple, Facebook, GitHub, and Google. Below these are icons for LinkedIn, Instagram, and Twitter. At the bottom, there are links for 'Aider', 'Politique de confidentialité', 'Règlement d'utilisation acceptable', 'Facebook', and 'Twitter'. The footer features the 'INSTRUCTURE' logo.

Applications similaires

Canvas LMS : Technologie et présentation du système d'identification

Si on regarde de plus près leur système d'identification, nous pouvons voir 2 types d'identification : OAuth2 et une simple connexion sécurisée composée d'une adresse mail et d'un mot de passe. Ces systèmes seront développés dans la partie protocole d'identification dans les pages suivantes.

Canvas LMS : Application de la technologie et du système d'identification pour Balabox

En appliquant, les instructions d'installation disponible sur [GitHub](https://github.com/instructure/canvas-lms) à l'adresse suivante : <https://github.com/instructure/canvas-lms/wiki/Quick-Start> Celle-ci permet au Raspberry Pi de pouvoir proposer Canvas qui est compatible avec Debian, malgré cette compatibilité, il nous est défavorable de nous inspirer étant donné que celle-ci a été faite en Ruby, un langage que nous n'avons jamais vu.

Git : <https://github.com/instructure/canvas-lms>

Documentation : <https://github.com/instructure/canvas-lms/wiki>

Applications similaires

Conclusion

Nom du site web	Langage de programmation	Système de connexion	Temps d'installation au Raspberry PI 4	TOTAL
Chamilo	PHP	Assez complet	Long	5
Moodle / MaDoc	PHP	Complet	Rapide	8
Canvas LMS	Ruby	Très complet	Assez rapide	6

Système de points

3 points

2 points

1 point

À partir du tableau ci-dessus, il serait judicieux de nous baser sur Moodle lors de la création de notre projet Identification pour la SAÉ. En effet, il propose à la fois un langage de programmation connu par les étudiants de ce projet. De plus, PHP est assez

confortable pour l'implémentation dans le Raspberry Pi 4 proposant une documentation très détaillée ainsi qu'une installation très rapide (à l'aide script). Malgré le fait que Chamilo soit dernier, son code est très accessible visuellement et techniquement parlant pourra nous être utile dans le cadre de la création d'un utilisateur. Concernant le front-end, l'utilisation de javascript serait préférable de par nos connaissances acquises à l'IUT. De plus, l'utilisation d'un framework (React, Angular, Vue...) est envisageable mais cette réflexion se fera par la suite avec les autres groupes du projet Balabox.

Les protocoles d'identification

Introduction

Dans cette partie, nous allons maintenant voir les différents protocoles d'utilisation existants. Après une analyse des protocoles, nous concluons en se demandant s'il est possible de l'appliquer au projet Balabox et déciderons du meilleur pour ce projet. Voici, ci-dessous, un aperçu rapide de différents protocoles d'identification :

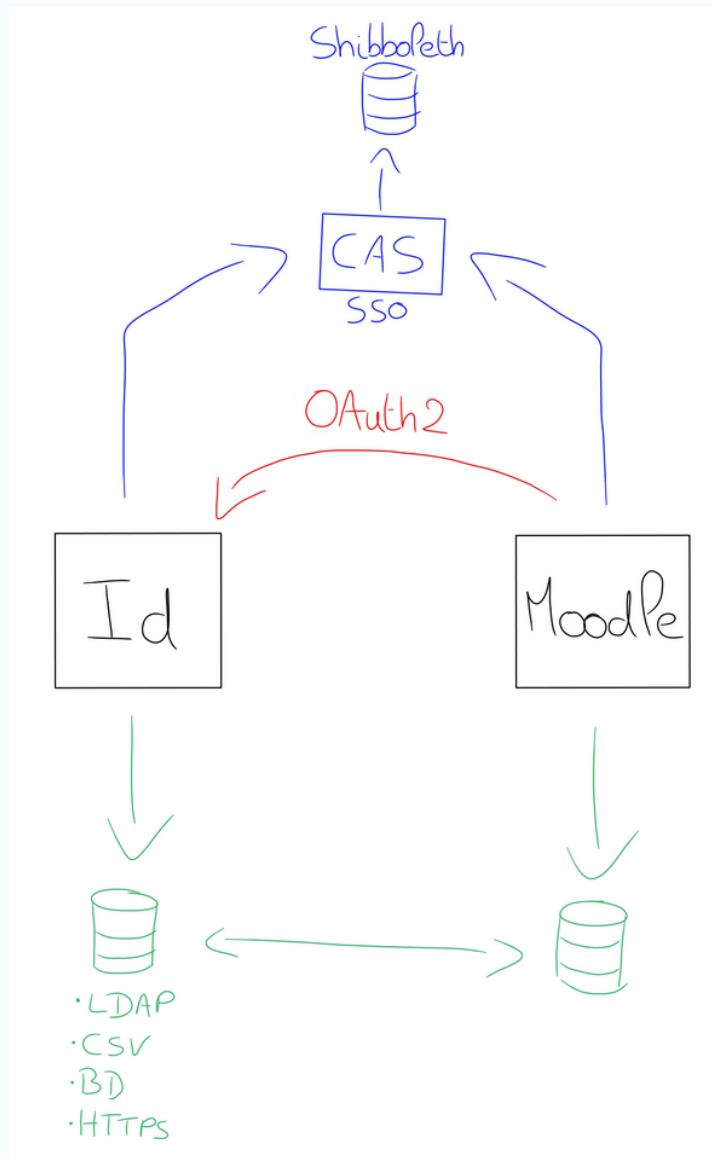


Figure 1 - Vue rapide de différents moyens d'identification

Les protocoles d'identification

Introduction

Sur ce schéma, Moodle représente la partie Moodlebox de la valise Balabox existant déjà. La partie nommée id représente notre projet : l'identification. Chaque méthode de protocole d'identification est dans une couleur différente afin de bien les différencier entre elles.

CAS

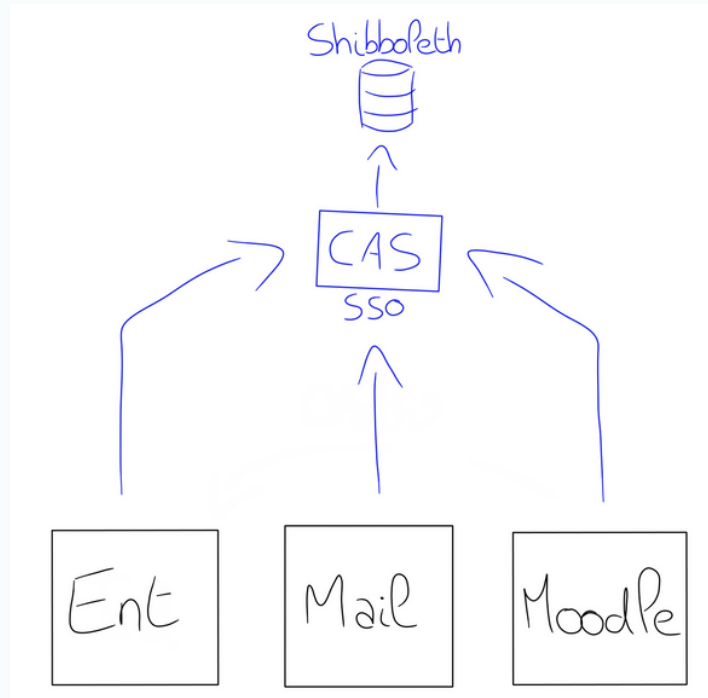
Le système CAS permet de centraliser l'authentification, celle-ci est très utile dans les environnements où de nombreuses applications réseaux différents partagent un ensemble commun d'utilisateurs dans une même base de données. Elle est utilisée notamment dans les universités françaises telles que Nantes, Bordeaux, l'Université Bretagne Sud.

L'avantage de ce protocole est qu'elle n'a jamais besoin de gérer les mots de passe, mais aussi que les utilisateurs peuvent basculer de manière transparente vers un autre site après la première authentification sans avoir à saisir de nouveau leurs identifiants uniquement si celle-ci possède le même protocole relié aux mêmes serveurs.

L'Université de Bretagne Sud, par exemple, utilise CAS, ce qui lui permet aux utilisateurs de l'iut d'avoir le même compte pour leur boîte mail, l'environnement numérique de travail et Moodle.

Les protocoles d'identification

CAS



Voici un exemple de code pour créer cela en Ruby basé sur Canvas LMS :

```
1. require "casclient"
2.
3. class Login::CasController < ApplicationController
4.   include Login::Shared
5.
6.   protect_from_forgery except: :destroy, with: :exception
7.
8.   before_action :forbid_on_files_domain
9.   before_action :run_login_hooks, :fix_ms_office_redirects, only: :new
10.
11.   delegate :client, to: :aac
12.
13.
14.   def create
15.     logger.info "Attempting CAS login with ticket #{params[:ticket]} in
        account #{@domain_root_account.id}"
16.     # only record further information if we're the first incoming ticket
        to fill out debugging info
17.     debugging = aac.debug_set(:ticket_received, params[:ticket],
        overwrite: false) if aac.debugging?
18.
```

Les protocoles d'identification

CAS

```
19.   st = CASClient::ServiceTicket.new(params[:ticket], cas_login_url)
20.   begin
21.     default_timeout = Setting.get("cas_timelimit", 5.seconds.to_s).to_f
22.
23.     timeout_options = { raise_on_timeout: true,
24. fallback_timeout_length: default_timeout }
25.
26.     Canvas.timeout_protection("cas:#{aac.global_id}", timeout_options)
27.   do
28.     client.validate_service_ticket(st)
29.   end
30.   rescue => e
31.     logger.warn "Failed to validate CAS ticket: #{e.inspect}"
32.     aac.debug_set(:validate_service_ticket, t("Failed to validate CAS
33. ticket: %{error}", error: e)) if debugging
34.     flash[:delegated_message] = t("There was a problem logging in at
35. %{institution}",
36.                                   institution:
37. @domain_root_account.display_name)
38.     return redirect_to login_url
39.   end
40.
41.   if st.is_valid?
42.     aac.debug_set(:validate_service_ticket, t("Validated ticket for
43. %{username}", username: st.user)) if debugging
44.     reset_session_for_login
45.
46.     pseudonym =
47. @domain_root_account.pseudonyms.for_auth_configuration(st.user, aac)
48.     if pseudonym
49.       aac.apply_federated_attributes(pseudonym, st.extra_attributes)
50.     elsif aac.jit_provisioning?
51.       pseudonym = aac.provision_user(st.user, st.extra_attributes)
52.     end
53.
54.     if pseudonym && (user = pseudonym.login_assertions_for_user)
55.       # Successful login and we have a user
56.
57.       @domain_root_account.pseudonyms.scoping do
58.         PseudonymSession.create!(pseudonym, false)
59.       end
60.       session[:cas_session] = params[:ticket]
61.       session[:login_aac] = aac.id
62.
63.       pseudonym.infer_auth_provider(aac)
64.       successful_login(user, pseudonym)
65.     else
66.       unknown_user_url = @domain_root_account.unknown_user_url.presence
67.       || login_url
```

Les protocoles d'identification

CAS

```
60.     logger.warn "Received CAS login for unknown user: #{st.user},
    redirecting to: #{unknown_user_url}."
61.     flash[:delegated_message] = t "Canvas doesn't have an account for
    user: #{user}", user: st.user
62.     redirect_to unknown_user_url
63.   end
64.   else
65.     if debugging
66.       if st.failure_code || st.failure_message
67.         aac.debug_set(:validate_service_ticket, t("CAS server rejected
    ticket: #{message} (#{code})", message: st.failure_message, code:
    st.failure_code))
68.       else
69.         aac.debug_set(:validate_service_ticket, t("CAS server rejected
    ticket."))
70.       end
71.     end
72.     logger.warn "Failed CAS login attempt. (#{st.failure_code}:
    #{st.failure_message})"
73.     flash[:delegated_message] = t("There was a problem logging in at
    #{institution}",
74.                                   institution:
    @domain_root_account.display_name)
75.     redirect_to login_url
76.   end
77. end
```

Source : https://github.com/instructure/canvas-lms/blob/master/app/controllers/login/cas_controller.rb

Les protocoles d'identification

Oauth2

Nous allons prendre des bouts de code de Canvas LMS pour mieux comprendre comment fonctionne OAuth2, si vous souhaitez voir toutes les méthodes de ce fichier, je vous suggère de regarder la source en question.

```
42. require "oauth"
43.
44. module Twitter
45.   class Connection
46.     def self.from_request_token(request_token, request_secret,
      oauth_verifier)
47.       access_token = OAuth::RequestToken.new(
48.         twitter_consumer,
49.         request_token,
50.         request_secret
51.       ).get_access_token(oauth_verifier: oauth_verifier)
52.       Twitter::Connection.new(access_token)
53.     end
54.
55.     def self.from_service_token(service_token, service_secret)
56.       access_token = OAuth::AccessToken.new(
57.         twitter_consumer,
58.         service_token,
59.         service_secret
60.       )
61.       Twitter::Connection.new(access_token)
62.     end
63.
64.     attr_reader :access_token
65.
66.     def self.twitter_consumer(key = nil, secret = nil)
67.       require "oauth"
68.       require "oauth/consumer"
69.       twitter_config = Twitter::Connection.config
70.       key ||= twitter_config["api_key"]
71.       secret ||= twitter_config["secret_key"]
72.       OAuth::Consumer.new(key, secret, {
73.         site: "https://api.twitter.com",
74.         request_token_path: "/oauth/request_token",
75.         access_token_path: "/oauth/access_token",
76.         authorize_path: "/oauth/authorize",
77.         signature_method: "HMAC-SHA1"
78.       })
79.     end
80.     private_class_method :twitter_consumer
81.   end
82. end
```

Les protocoles d'identification

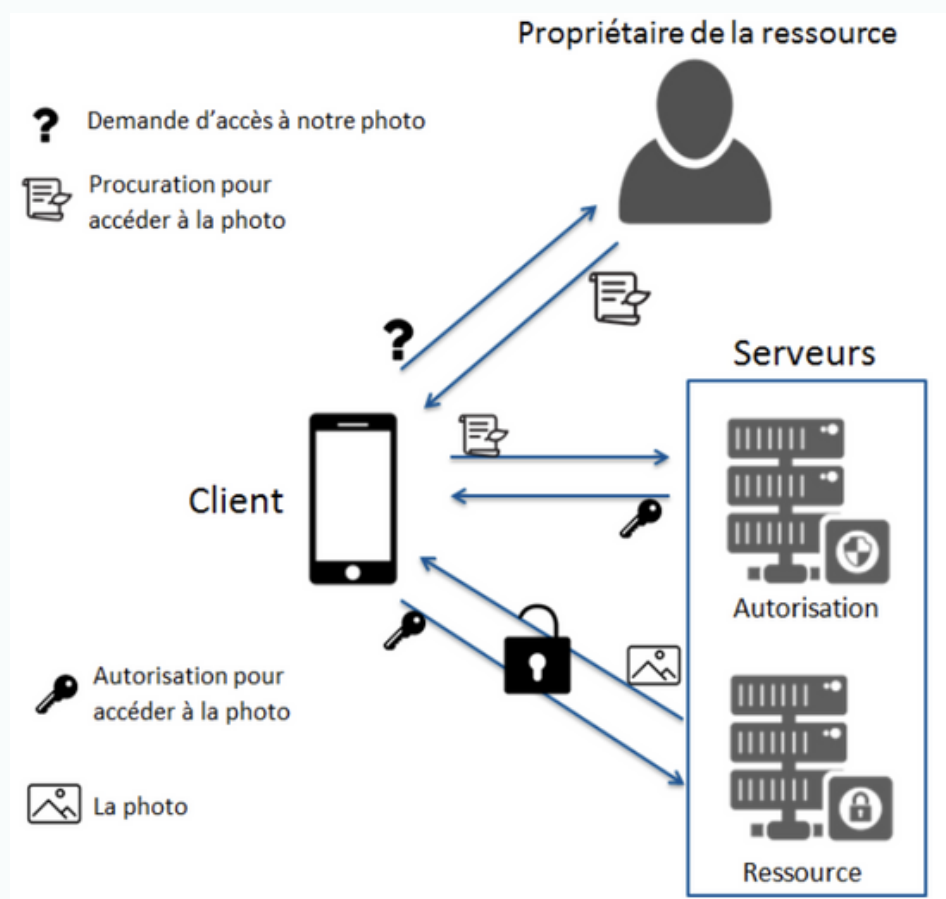
Oauth2

Pour résumer ce code, celle-ci permet à Canvas-lms (client) d'accéder à une ressource protégée de Twitter(resource owner). Cette autorisation se traduit par la délivrance d'un token d'accès (et éventuellement d'un token de rafraîchissement) qui permet au client de dialoguer avec le serveur hébergeant les ressources protégées (serveur de ressource).

Ce "token" est représenté par une chaîne de caractères unique permettant d'identifier le client et les différentes informations utiles dans le cadre du processus d'autorisation.

Le processus de création et connexion se déroule comme cela :

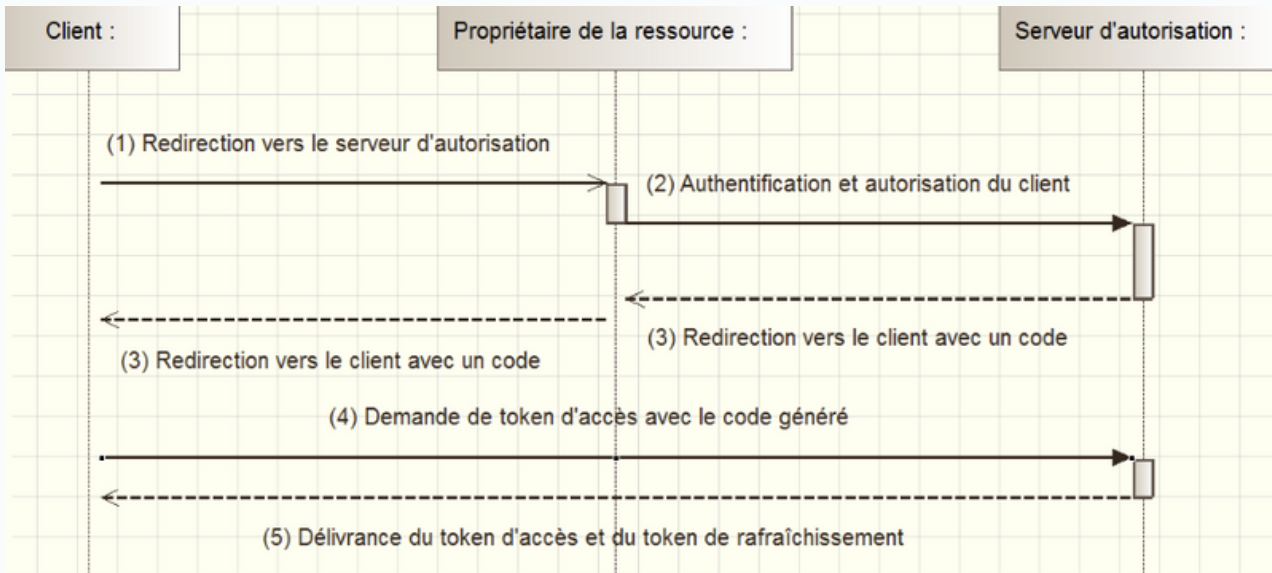
Dans un plan schématisé :



Les protocoles d'identification

Oauth2

Dans un plan détaillé :



- Le client redirige le propriétaire de la ressource vers le serveur d'autorisation. Le client doit inclure son identifiant dans la requête de redirection et le niveau d'accès qu'il souhaite obtenir.
- Le propriétaire de la ressource s'authentifie auprès du serveur d'autorisation et approuve ou non la requête du client.
- Si la requête est autorisée, le serveur d'autorisation redirige à nouveau le propriétaire de la ressource en utilisant l'URL de redirection défini par le client. Cet URL est renseigné à l'enregistrement du client et peut aussi être choisi dans la requête à l'étape 1. La requête de redirection contient un code d'autorisation dans l'URL.

Les protocoles d'identification

Oauth2

- Avec le code d'autorisation ainsi obtenu, le client demande un token d'accès en prenant le soin de s'authentifier à son tour auprès du serveur d'autorisation.
- Une fois le client authentifié, le serveur d'autorisation valide le code d'autorisation et s'assure que l'URL de redirection est identique à celle utilisée dans la troisième étape. Si toutes ces contraintes sont respectées, le serveur d'autorisation renvoie au client un token d'accès et éventuellement un token de rafraîchissement.

Pour assurer ces différentes interactions, le serveur d'autorisation doit mettre à disposition des clients deux URL. Un URL d'autorisation (Authorization endpoint) qui sera utilisée dans l'étape 1 et permettra d'obtenir un code d'autorisation et un URL de génération de tokens (Tokens endpoint) permettant d'obtenir les différents tokens.

Les protocoles d'identification

Services d'annuaires - Authentification HTTP(S) : CSV

Considérons un fichier données Utilisateur.csv provenant d'un site externe compatible avec Balabox dans le cadre d'un transfert de données :

nom;prenom;age;genre;mail;mdp;ville

LEBOURHIS;Hugo;19;homme;lebourhis.hugo@kaz.fr;ergegrzegz+\$;Quimperle

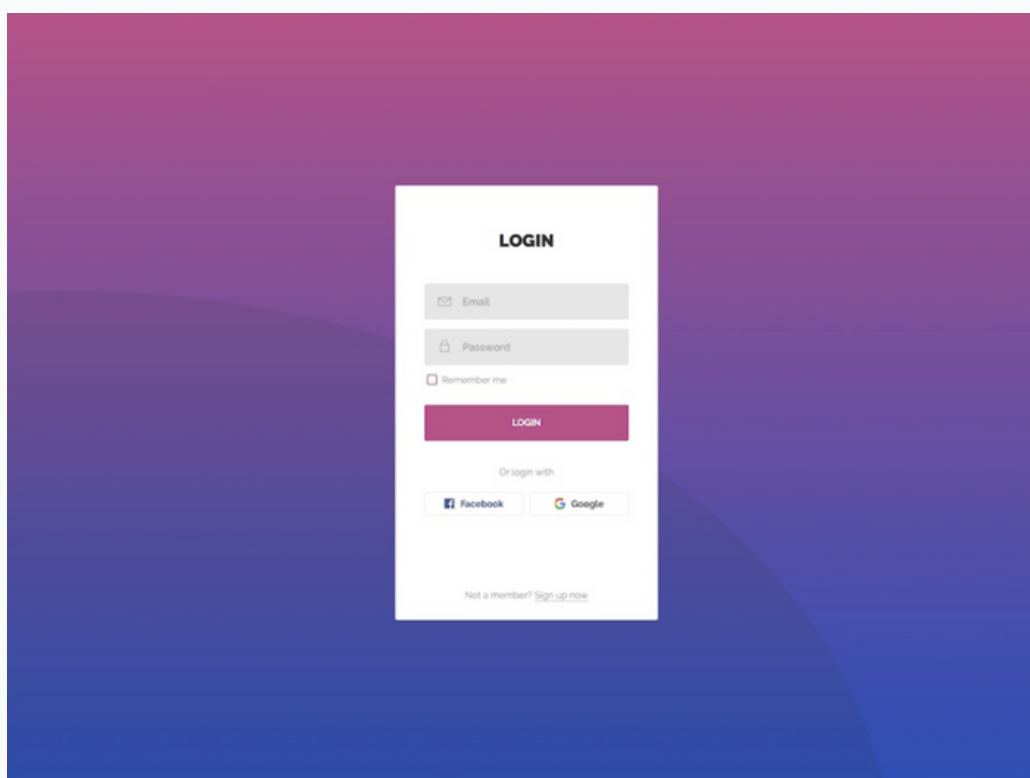
LECLOAEC;Emma;17;femme;lecloaec.emm@kaz.fr;ethehzehe!!;Lorient

DELLA;Djena;26;femme;djenapro.contact@kaz.fr;djenadellalabestdu44++!!;Nantes

RISEANA;Amelia;23;femme;rise.dragon@kaz.fr;ebhebebe!\$::;Rennes

POULA;Thomas;19;homme;poula.thomas2@kaz.fr;brnhnernjte\$++;La Rochelle

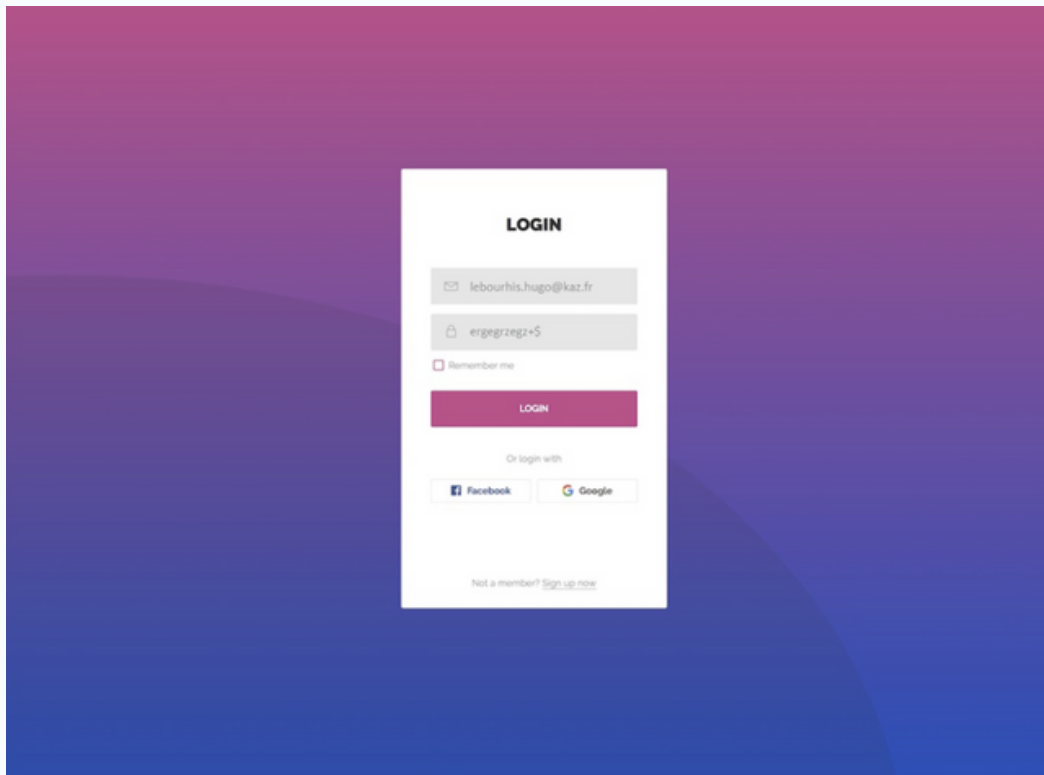
Considérons désormais une page d'accueil :



Les protocoles d'identification

Services d'annuaires - Authentification HTTP(S) : CSV

Entrons les identifiants de M. Le Bourhis :

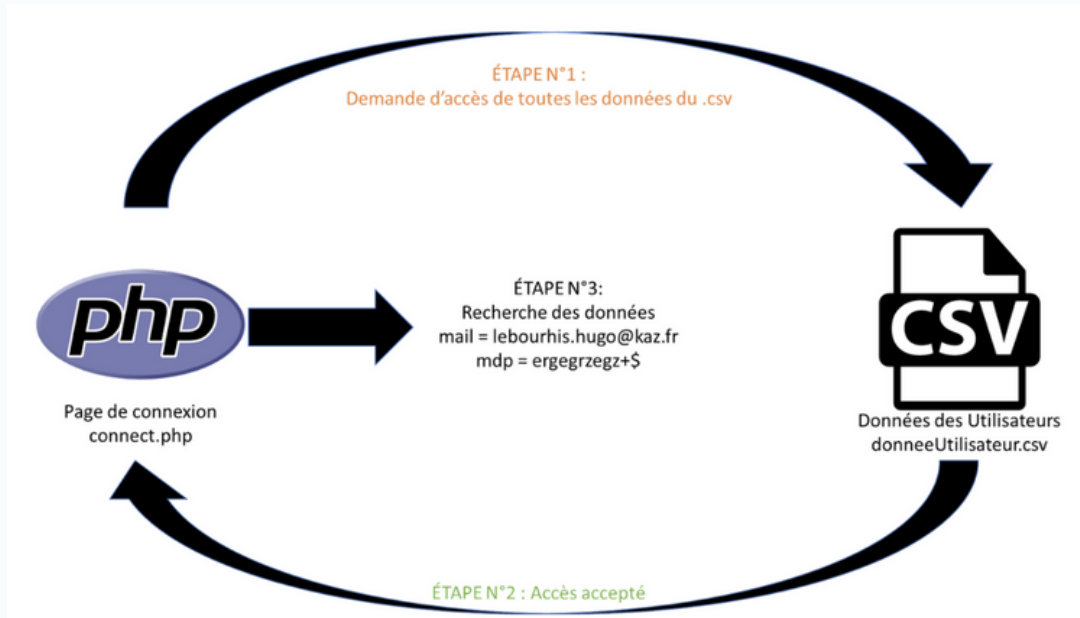


Considérons désormais un fichier connect.php qui fera le nécessaire pour se connecter à partir des informations ci-dessus.

Maintenant des informations vont transiter entre le fichier .csv et le fichier connect.php :

Les protocoles d'identification

Services d'annuaires - Authentification HTTP(S) : CSV



Le fichier .csv fonctionnera comme si c'était une base de données sauf que celle-ci ne sera pas aussi protégée qu'une authentification vers une base de données que nous allons voir dès maintenant.

Si les données entrées sont trouvées depuis le fichier .csv, le site va se connecter à partir des différents données existant, c'est à dire :

nom = LE BOURHIS

prenom = Hugo

age= 19

genre = homme

mail = lebourhis.hugo@kaz.fr

mdp = ergegrzegz+\$

ville = Quimperle

Les protocoles d'identification

Services d'annuaires - Authentification HTTP(S) : BD

Dans le cas de la création et connexion d'un utilisateur en utilisant uniquement la base de données, nous allons nous baser sur SportTrack, un projet qui a été en R3.01 - Développement Web à l'IUT de Vannes. Celle-ci fonctionne en 2 temps pour se connecter :

1ère étape : après avoir inséré un mail et mot de passe, le site web va chercher à partir de la fonction findUser() initialisé dans la classe UserDAO.

2ème étape : si celle-ci existe dans la BD elle renvoie toutes les informations sous forme de liste qui concerne cet utilisateur sinon s'il n'existe pas, il renvoie une erreur.

```
1. <?php
2. require(CONTROLLERS_DIR.'/Controller.php');
3. require(__ROOT__.'/model/UserDAO.php');
4. require(__ROOT__.'/model/User.php');
5.
6.
7. class ConnectUserController extends Controller{
8.
9.     public function get($request){
10.         $this->render('user_connect_form',[]);
11.     }
12.
13.     public function post($request){
14.         $user =
15.         UserDAO::getInstance()->findUser($request['mail'],$request['password']);
16.         if($user != null){
17.             $status = session_status();
18.             if($status == PHP_SESSION_NONE){
19.                 session_start();
20.             } else if ($status == PHP_SESSION_ACTIVE){
```

Les protocoles d'identification

Services d'annuaires - Authentification HTTP(S) : BD

```
21.         session_destroy();
22.         $_SESSION['user'] = $user;
23.         session_start();
24.         $this->render('user_connect_valid', [$user[0] ->
    getName(), $user[0] -> getfName()]);
25.
26.     }
27.     $_SESSION['idUser'] = $user[0] -> getIdUser();
28.     $_SESSION['surname'] = $user[0] -> getName();
29.     $_SESSION['name'] = $user[0] -> getfName();
30.     $_SESSION['date'] = $user[0] -> getBirthDate();
31.     $_SESSION['gender'] = $user[0] -> getGender();
32.     $_SESSION['size'] = $user[0] -> getSize();
33.     $_SESSION['weight'] = $user[0] -> getWeight();
34.     $_SESSION['mail'] = $request['mail'];
35.     $_SESSION['password'] = $request['password'];
36. } else{
37.     $this->render('user_connect_valid', [null, null]);
38. }
39.
40. }
41. }
42. ?>
43.
```

Sources :

1. <https://github.com/Michael16b/SportTrack/blob/master/model/UserDAO.php>
2. <https://github.com/Michael16b/SportTrack/blob/master/controllers/connect.php>

Les protocoles d'identification

Services d'annuaires - Authentification HTTP(S) : LDAP

Définition du modèle LDAP:

L'authentification LDAP suit le modèle client/serveur, où le client est généralement une application ou un système compatible LDAP qui demande des informations à une base de données associée. Et le serveur est, bien sûr, le serveur LDAP.

Le serveur est une base de données dont le schéma est flexible. Elle peut donc stocker des informations relatives aux utilisateurs (nom, prénoms, mot de passe etc...). Par conséquent, le cas d'utilisation courant de LDAP est le stockage d'identités d'utilisateur.

Comment cela fonctionne ?

Pour résumer, un client envoie à un serveur LDAP une demande d'informations stockées dans une base de données LDAP, accompagnée des informations d'identification de l'utilisateur. Si les informations envoyées sont bonnes, alors le serveur retourne les informations demandées, dans le cas contraire, l'accès aux données est refusé. En fonction du niveau de privilège de l'utilisateur indiqué dans la base de données, la demande aura accès à certaines données.

Les protocoles d'identification

Services d'annuaires - Authentification HTTP(S) : LDAP

Que faut-il pour mettre en œuvre LDAP ?

Bien qu'efficace, la réalisation et la mise en œuvre de l'infrastructure LDAP pour répondre aux besoins de la gestion des identités d'une organisation moderne peut être considérable.

La solution LDAP est de plus en plus utilisée, car elle est très adaptée à une solution cloud. Mais pour un projet tel que Balabox, cela serait une perte de temps.

Les protocoles d'identification

Conclusion

Protocole	Facilité à mettre en place	Documentation & protocole utilisé de nombreuses fois par des projets	Utilité pour la Balabox	Sécurité	Total (/20)
CAS	D'une extrême difficulté	Excellent, utilisé par les plus gros projets(Moodle,Madoc, Canvas),	Inutile car Balabox ne possède pas plusieurs sites reliés	Excellente sécurité	12 points
OAuth2	Compiqué mais possible	Excellent, protocole très répandu dans le monde	Intéressant mais Balabox n'a besoin de connexion extérieur	Excellente sécurité	17 points
CSV	Très simple à mettre en place mais possible problème de compatibilité avec le code si le fichier .csv n'est pas EXACTEMENT comme ce que veut le code	Très peu de documentation mais une simple réflexion est utile pour comprendre son système	Intéressant	Aucune sécurité	12 points
BD	Simple à mettre en place	Excellent	Très intéressant	Sécurité qui se base uniquement sur le protocole d'identification de la base de données	18 points

Les protocoles d'identification

Conclusion

Protocole	Facilité à mettre en place	Documentation & protocole utilisé de nombreuses fois par des projets	Utilité pour la Balabox	Sécurité	Total (/20)
LDAP	Compliqué et long	Protocole qui est de plus en plus utilisé grâce au cloud. Possède un module et une documentation complète sur PHP. Mais peu d'exemple, car très personnel au niveau de la mise en place et beaucoup de no code	Peu intéressant	Très sécurisé	11 points

Système de points	Traduction
5 points	Excellent
4 points	Bien
3 points	Moyen
2 points	Mauvais
1 point	A éviter

Après un tableau récapitulatif, il s'avère que le protocole de BD est la meilleure option, concernant notre projet. Elle a déjà été utilisée sur notre projet de développement web en 2ème année de BUT Informatique et pourrait nous faciliter la tâche, de plus la documentation sur internet est très bonne avec une bonne sécurité pour éviter que des personnes mentionnées essayent de se connecter sur la Balabox.

Techonologie existante : la MoodleBox

Le RaspBerry Pi contient déjà la MoodleBox, une version combinant un point d'accès sans fils avec un serveur Moodle complet. Pour cette veille technologique, nous nous demandons alors comment nous pouvons utiliser cette MoodleBox pour notre projet. Est-il possible de lier sa base de données à la nôtre ? La documentation de MoodleBox disponible sur internet (voir lien) ne nous donne pas cette réponse. En effet, nous avons accès à la base de données avec interface administrateur. Cependant, nous voulons communiquer avec cette dernière en ligne de code, ce qui permettrait d'automatiser le processus sans intervention humaine directement sur l'interface. Ne connaissant pas la réponse à notre question, nous avons alors lancé une nouvelle discussion sur le forum du site afin d'avoir une aide. Sur le code présent sur github, l'architecture de la base de données de la MoodleBox n'est pas trouvable (voir lien). Nous savons donc seulement que c'est une base de données en mysql utilisant MariaDB, technologies que nous avons vu l'année dernière lors d'un projet de fin d'année.

Git :

<https://github.com/moodlebox/moodlebox/blob/main/roles/database/files/etc/mysql/mariadb.conf.d/50-server.cnf>

Documentation : <https://moodlebox.net/fr/help/>

Conclusion finale

Après nos différentes réflexions avec les protocoles d'identification et les différentes plateformes en concurrence avec notre projet, nous avons pu déterminer ce qui nous rapprochait au mieux des attentes de notre cliente pour l'identification, c'est-à-dire le langage PHP pour le back-end dans le cas du protocole d'une authentification par service annuaire et plus précisément via une authentification par base de données. Et celle-ci nous a permis d'optimiser notre temps de création de ce projet.