MANUEL TECHNIQUE & RÉFÉRENCE D'ARCHITECTURE

Authentification Pronote

Référence technique

Analyse des flux protocolaires, mécanismes de session et architectures de sécurité

NATURE DU DOCUMENT

Spécification Protocoles & Sécurité

CLASSIFICATION

Référence Publique

DATE & VERSION

Septembre 2026 • Version 2.0

TYPOGRAPHIE

EB Garamond / Carlito

[!] AVERTISSEMENT

Ce document **n'est pas un guide officiel**. Il n'est ni approuvé, ni validé, ni autorisé par le Ministère de l'Éducation Nationale française ni par Docaposte (éditeur de Pronote / Index Éducation). Les informations présentées reposent sur des recherches publiques, des travaux de rétro-ingénierie menés par la communauté open-source et des analyses techniques. Les protocoles décrits ne sont pas officiellement publiés par Index Éducation et peuvent évoluer sans préavis.

Ce document a été produit à l'aide de plusieurs agents basés sur des modèles de langage (LLM) :

- **Mercury 2.5** (agent *explorer*) — exploration du code pour établir les faits techniques.
- **Gemini 3.5 Flash** (agent *web-explorer*) — recherche des méthodes d'authentification externes.
- **Hy3** (agent *planner*) — planification de la structure du document et découpage en sections.
- **Mistral Medium** (agent *tech-writer*) — rédaction du document.
- **GPT-5.6 Luna** (agent *reviewer*) — revue du contenu pour l'exactitude et la cohérence.
- **GLM-5.2** (agent *orchestrator*) — coordination et intégration du travail.
- **Gemini 3.7 Flash** (agent *ui-designer*) — conception de la version LaTeX/PDF.

Table des matières

1 Introduction

Ce manuel documente l'ensemble des méthodes d'authentification connues pour accéder aux données élèves/parents de Pronote (notes, emploi du temps, devoirs, absences, etc.). Il couvre à la fois les mécanismes officiellement supportés et les protocoles issus de l'analyse communautaire.

Le public visé inclut les développeurs, ingénieurs sécurité et intégrateurs système devant maîtriser l'authentification Pronote au niveau protocolaire. Les descriptions utilisent du pseudocode générique, des échanges HTTP et des schémas protocolaires, sans présupposer de langage ou framework spécifique.

> **i Remarque méthodologique**
>
> Les méthodes au-delà de l'export iCal s'appuient sur des recherches publiques et l'analyse de la communauté open source. Ces protocoles, non publiés officiellement par Index Éducation, peuvent évoluer sans préavis.

2 Vue d'ensemble comparative

Le tableau 2.1 offre une synthèse comparative des sept méthodes d'authentification recensées pour Pronote.

Méthode	Périmètre	Identifiants	Expiration	Complexité	Statut
URL iCal sécurisée (`icalsecurise`)	Emploi du temps (+ devoirs si inclus)	Jeton dans l'URL	Longue durée	Très faible	Officiel
Connexion directe	Complet	Identifiant + mot de passe établissement	Session ∼15–30 min	Élevée	Reverse-engineered
SSO ENT (CAS / SAML / Oze)	Complet	Identifiants ENT	Dépend session ENT	Très élevée	Reverse-engineered
SSO EduConnect	Complet	Identifiants nationaux EduConnect	Dépend session EduConnect	Très élevée	Reverse-engineered
CAS	Complet	Identifiants CAS	Ticket de service	Modérée	Officiel
QR Code + jeton mobile	Complet	PIN 4 chiffres → `jetonConnexionApp` + UUID	QR 10 min ; jeton long terme	Modérée	Reverse-engineered
API publique	N/A	N/A	N/A	N/A	Inexistante

TABLE 2.1 : Matrice comparative globale des méthodes d'authentification Pronote.

> **i Définition du périmètre « Complet »**
>
> Le terme *Complet* désigne l'accès aux notes, emploi du temps, devoirs, absences, messagerie et paramètres, tandis que la méthode iCal se limite à l'emploi du temps et éventuellement aux devoirs (cahier de textes).

3 Méthode 1 : URL iCal sécurisée (icalsecurise)

3.1 Principe général

Pronote expose un flux de calendrier en lecture seule conforme à la norme iCalendar (RFC 5545). L'accès est contrôlé par un jeton secret intégré dans l'URL sous forme de paramètre de requête `icalsecurise`. Ce jeton est unique par utilisateur et par établissement. **Le jeton constitue la seule crédentiale** : il doit être traité comme un mot de passe. Toute requête HTTP GET vers cette URL permet de récupérer les données du calendrier, sans nécessiter de cookies, de session ni d'en-têtes d'authentification.

3.2 Obtention du jeton

Le jeton s'obtient manuellement depuis l'interface web de Pronote :

1. Se connecter à Pronote (Espace Parents ou Espace Élève) via n'importe quelle méthode d'authentification.
2. Accéder à la vue « Emploi du temps ».
3. Utiliser la fonction « Export iCal » ou « Exporter ».
4. Pronote génère une URL contenant le paramètre `icalsecurise`.
5. Copier cette URL : elle constitue la crédentiale.

Cette URL doit être stockée de manière sécurisée (ex. : gestionnaire de secrets, variables protégées). **Elle ne doit jamais être versionnée ou partagée en clair.**

3.3 Format de l'URL

L'URL suit la structure suivante :

```
https://{etablissement}.index-education.net/pronote/ical/Edt_{prenom}.ics?icalsecurise
    ={jeton}&version={version}&param={param}
```

Exemple masqué :

```
https://XXXXXXX.index-education.net/pronote/ical/Edt_Alice.ics?icalsecurise••••••••=&
    version=2023&param=...
```

Paramètres de requête :

— `icalsecurise` : jeton secret (crédentiale).
— `version` : version de Pronote.
— `param` : paramètres supplémentaires (optionnels).

3.4 Flux détaillé

Le protocole d'authentification se résume ainsi :

1. **Préparation** : Le client dispose de l'URL iCal sécurisée (obtenue comme décrit ci-dessus).
2. **Requête HTTP** : Le client effectue une requête HTTP GET :

```
GET {ical-url}
Accept: text/calendar, */*;q=0.5
User-Agent: {identifiant-client}
```

   — Délai d'attente : configurable (recommandé : 20 secondes).
3. **Validation de la réponse** :
   — Si le code HTTP n'est pas 2xx → échec d'authentification ou erreur serveur.
   — Si le corps de la réponse ne contient pas `BEGIN:VCALENDAR` → le jeton est probablement expiré ou l'URL est invalide. Le serveur peut retourner une page HTML d'erreur au lieu des données de calendrier.
4. **Traitement** : Si la validation réussit, le corps de la réponse est une donnée iCalendar valide, prête à être analysée.

3.5 Données échangées

— Le client envoie une seule requête HTTP GET vers l'URL iCal sécurisée.
— En-têtes de requête : `Accept: text/calendar, */*;q=0.5` et `User-Agent: <identifiant-client>`.
— Le serveur retourne une charge utile iCalendar (RFC 5545) si le jeton est valide.
— Si le jeton est invalide ou expiré, le serveur retourne une réponse non-calendrier (typiquement du HTML).
— Aucun cookie, jeton de session ou en-tête d'authentification n'est échangé — l'URL **est** la crédentiale.

3.6 Identifiants et jetons

Le seul identifiant utilisé est le jeton `icalsecurise`, intégré directement dans l'URL. Ce jeton est :

— **Unique** par utilisateur et par établissement.

— **Sensible** : il équivaut à un mot de passe et doit être traité comme tel.
— **Transmis en clair** dans la chaîne de requête de l'URL, mais protégé en transit par HTTPS.
— **Autosuffisant** : aucune autre information (nom d'utilisateur, mot de passe, cookie de session ou jeton OAuth) n'est requise. L'URL **est** l'authentification.

3.7 Cycle de vie

Le jeton est **pérenne** : il reste valide jusqu'à sa révocation manuelle par l'utilisateur dans les paramètres Pronote, ou jusqu'à sa régénération par l'établissement (généralement à la rentrée scolaire).

Aucun mécanisme de rafraîchissement automatique ou de rotation n'existe. En cas d'expiration ou de rotation, le serveur retourne une réponse non-iCalendar (souvent une page HTML mentionnant « *Session expirée* »). Le client détecte cette situation en vérifiant l'absence de `BEGIN:VCALENDAR` dans le corps de la réponse.

La récupération nécessite une **réextraction manuelle** d'une nouvelle URL iCal depuis l'interface Pronote.

> **i Bonnes pratiques de cycle de vie**
>
> Tester l'URL avant chaque rentrée et la régénérer proactivement si l'établissement est connu pour renouveler les jetons à cette période.

3.8 Sécurité

> **§ Analyse de sécurité — URL iCal sécurisée**
>
> 1. **Surface d'attaque** : Le jeton est encodé dans l'URL. Il peut être exposé via les logs d'accès du serveur, les logs proxy, les en-têtes `Referer`, l'historique du navigateur ou une interception réseau (atténué par HTTPS).
> 2. **Exposition des identifiants** : En cas de fuite, le jeton accorde un accès en lecture à l'emploi du temps (et aux devoirs, si inclus) jusqu'à sa rotation par l'établissement.
> 3. **Résistance au rejeu** : Faible — absence de *nonce*, de validation temporelle ou de protection contre le *replay*. Toute entité disposant de l'URL peut récupérer les données à tout moment.
> 4. **Rotation** : Manuellement uniquement, via la régénération de l'URL dans Pronote. L'établissement contrôle les réinitialisations côté serveur.
> 5. **Recommandations** : Conserver l'URL comme un secret (jamais en contrôle de version). Utiliser HTTPS (par défaut). Limiter l'accès à l'URL au strict besoin d'en connaître. Renouveler proactivement aux changements d'année scolaire. Masquer l'URL dans les messages d'erreur et les logs pour éviter les fuites accidentelles.

3.9 Limitations

— **Périmètre des données** : limité à l'emploi du temps et, si inclus par l'établissement, aux devoirs (*cahier de textes*).
— **Accès en lecture seule** : aucune modification possible.
— **Exclusions** : notes, absences, messagerie, bulletins ou paramètres sont inaccessibles.
— **Latence** : l'export iCal peut présenter un délai de mise à jour (plusieurs heures) avant de refléter les modifications Pronote.
— **Rafraîchissement** : impossible par programmation — une intervention manuelle est toujours requise.

3.10 Statut

Officiel — L'export iCal est une fonctionnalité supportée par Pronote, éditée par Index Éducation. Le mécanisme de jeton fait partie intégrante du produit, bien que son format interne et sa logique de génération ne soient pas documentés publiquement.

4 Méthode 2 : Connexion directe (identifiant / mot de passe)

4.1 Principe général

La connexion directe utilise un protocole propriétaire de type JSON sur HTTP(S), sécurisé par un chiffrement AES-256-CBC spécifique à la session et un mécanisme de défi-réponse. Il s'agit du protocole natif de Pronote, tel qu'utilisé par son interface web.

4.2 Flux détaillé

Étape 1 — Initialisation de la session

Requête : `GET /pronote/<espace>.html`

Le client effectue une requête GET vers l'espace cible (ex. `/pronote/eleve.html` pour un élève, `/pronote/parent.html` pour un parent). La réponse HTML contient un gestionnaire JavaScript `onload` avec les paramètres de session :

```
Session initialization parameters:
  h     = <session_id>
  a     = <espace_id> (3 = Élève, 7 = Parent)
  sCrA  = <encryption_flag>
  sCoA  = <compression_flag>
```

- `h` : identifiant de session (chaîne ou nombre unique).
- `a` : identifiant de l'espace (`3` pour Élève, `7` pour Parent).
- `sCrA` / `sCoA` : indicateurs de chiffrement et de compression.

Étape 2 — Échange de clés

Requête : `POST /pronote/appelfonction/<espace_id>/<session_id>/<numeroOrdre>`

Le client envoie une charge utile `FonctionParametres` contenant `donneesSec.donnees.Uuid` :

- En HTTPS : un IV AES de 16 octets encodé en base64.
- En HTTP non sécurisé : un IV chiffré avec RSA-1024.
- `numeroOrdre` est un compteur incrémental (début à 1).

— Pour la première requête, `numeroOrdre` est chiffré avec AES-256-CBC, une clé vide MD5 (`d41d8cd98f00b204e9800998ecf8427e`) et un IV nul.
— Pour les requêtes suivantes, l'IV de session (issu de `Uuid`) est utilisé.

Étape 3 — Identification

Requête : `POST ... / Identification`

Le client soumet une charge utile JSON :

```
{
  "nom": "Identification",
  "session": "<session_id>",
  "numeroOrdre": "<compteur_chiffré>",
  "donneesSec": {
    "donnees": {
      "identifiant": "<nom_utilisateur>",
      "genreConnexion": 0,
      "genreEspace": <espace_id>,
      "pourENT": false,
      "enConnexionAuto": false
    }
  }
}
```

Le serveur retourne : `alea` (sel aléatoire), `challenge` (chaîne hexadécimale), `modeCompLog` et `modeCompMdp` (indicateurs de normalisation de casse).

Étape 4 — Résolution du défi

Le client calcule le hachage du mot de passe :

```
mtp = MAJUSCULE(HEX(SHA256(alea + mot_de_passe_utilisateur)))
```

La clé de déchiffrement du défi est dérivée :

```
key_challenge = MD5(nom_utilisateur + mtp)
```

Le client déchiffre `challenge` avec AES-256-CBC, `key_challenge` et l'IV de session. Il supprime ensuite un caractère sur deux dans le texte en clair (ex. `abcdef` → `ace`), puis rechiffre la chaîne modifiée avec les mêmes paramètres AES et l'encode en hexadécimal.

Étape 5 — Authentification

Requête : `POST ... / Authentification`

Le client envoie la réponse au défi via la fonction `Authentification`. Le serveur retourne les métadonnées utilisateur et une chaîne `cle` (entiers séparés par des virgules). Le client déchiffre `cle`, analyse les octets et calcule `MD5(octets)` pour obtenir la clé principale de chiffrement de session pour tous les appels API ultérieurs.

4.3 Données échangées

Identifiant de session, identifiant d'espace, IV AES (base64 ou chiffré RSA), compteur `numeroOrdre`, sel `alea`, chaîne de défi `challenge`, clé de session (`cle`). Toutes les données sensibles sont chiffrées en transit (AES-256-CBC).

4.4 Identifiants et jetons

— **Identifiants** : nom d'utilisateur Pronote et mot de passe attribués par l'établissement.
— **Jetons de session** : identifiant de session (`h`), compteur de séquence (`numeroOrdre`), clé symétrique AES-256 (dérivée de `cle`).

4.5 Cycle de vie

— Les identifiants de session expirent après une courte période d'inactivité (généralement 15 à 30 minutes).
— La clé de session doit être utilisée pour tous les appels API ultérieurs dans la même session.
— Une réauthentification est nécessaire après expiration de la session.

4.6 Sécurité

§ Analyse de sécurité — Connexion directe

1. **Surface d'attaque** : protocole propriétaire avec cryptographie personnalisée. Le point de terminaison de connexion est accessible depuis Internet. Les attaques de l'homme du milieu (MITM) sont atténuées par HTTPS (mais RSA-1024 est utilisé pour l'échange d'IV en HTTP non sécurisé, ce qui est faible selon les normes modernes).
2. **Exposition des identifiants** : le nom d'utilisateur est transmis dans la requête d'identification (chiffré). Le mot de passe n'est jamais transmis en clair : seule la réponse au défi est envoyée.
3. **Résistance au rejeu** : modérée. Le mécanisme de défi-réponse utilise un sel aléatoire (`alea`) par session, rendant difficile le rejeu d'une réponse de défi capturée. Cependant, le compteur `numeroOrdre` doit être géré avec soin pour éviter toute manipulation de séquence.

4. **Rotation** : aucune rotation automatique des jetons. La rotation des mots de passe dépend de la politique de l'établissement. Les clés de session expirent avec la session.
5. **Recommandations** : utiliser systématiquement HTTPS. Implémenter une gestion rigoureuse de `numeroOrdre`. Stocker les identifiants de manière sécurisée. Noter que la cryptographie personnalisée n'est pas équivalente à une authentification TLS standard : s'appuyer sur HTTPS pour la sécurité du transport.

4.7 Limitations

— La plupart des établissements secondaires français liés à un ENT ou à EduConnect bloquent les connexions directes par nom d'utilisateur/mot de passe et imposent le SSO.
— Le protocole propriétaire n'est pas officiellement documenté et peut changer sans préavis.
— La cryptographie personnalisée (AES-256-CBC avec des clés dérivées de MD5) est non standard et n'a pas fait l'objet d'un audit indépendant.

4.8 Statut

Rétro-conçu — Index Éducation ne publie pas le protocole. Toutes les connaissances proviennent de l'analyse communautaire du client web Pronote en JavaScript.

5 Méthode 3 : ENT (Espace Numérique de Travail)

5.1 Principe général

La plupart des collèges et lycées français accèdent à Pronote via un ENT (Espace Numérique de Travail) régional ou départemental. L'ENT agit comme fournisseur d'identité (IdP) : l'utilisateur s'authentifie auprès de l'ENT, qui établit ensuite une session avec Pronote par SSO (*Single Sign-On*). Pronote reçoit des identifiants délégués sans gérer directement la connexion.

5.2 Flux détaillé

1. **Connexion ENT** : L'utilisateur soumet ses identifiants à l'endpoint de connexion spécifique à l'ENT. Chaque ENT utilise son propre mécanisme (formulaire, CAS, SAML, Keycloak, etc.).
2. **Redirection SSO vers Pronote** : L'ENT redirige la session authentifiée vers Pronote via un lien connecteur ou une URL proxy (ex. `/cas/proxySSO/...` ou un lien SAML direct). L'ENT valide la session et redirige vers l'instance Pronote de l'établissement avec des cookies ou assertions SAML valides.
3. **Handshake Pronote** : La réponse HTML initiale de Pronote contient des identifiants temporaires dans l'attribut `onload` du `<body>` :

   ```
   Session initialization parameters:
     e = <temp_login>
     f = <temp_auth_token>
     ...
   ```

   — `e` : chaîne de connexion temporaire (unique par session SSO).
   — `f` : jeton d'authentification temporaire.
4. **Challenge de session** : Le client exécute la requête standard `Identification` de Pronote (comme en Méthode 2) avec `pourENT: true`. Le challenge est résolu via une dérivation simplifiée :

   ```
   key_ENT = MD5(UPPERCASE(HEX(SHA256(f))))
   ```

   Cela remplace la dérivation `MD5(username + mtp)` utilisée en connexion directe. Aucun mot de passe utilisateur n'est nécessaire : le jeton `f` émis par l'ENT sert de justificatif.

5.3 Architectures ENT prises en charge

Le tableau 5.1 détaille les principales familles d'ENT et leurs mécanismes d'interfaçage.

Architecture	Exemples d'académies / départements	Mécanisme SSO
Open ENT NG / Open Digital Education	ent.iledefrance.fr, Paris Classe Numérique, Mon Collège Val d'Oise, L'Éduc de Normandie	SAML / redirection
Kosmos / Skolengo CAS	Mon Bureau Numérique, Mon-ENT-Occitanie, Cybercollèges42	CAS
Oze ENT	Divers collèges et lycées	Keycloak avec endpoints proxy `/v1/ozapps`
WAYF / Shibboleth	e-lyco (Pays de la Loire)	SAML / Shibboleth
Portails personnalisés	Atrium Sud, LaClasse Lyon	Formulaires simples

Table 5.1 : Architectures ENT régionales et protocoles SSO associés.

5.4 Données échangées

Cookies/assertions de session ENT → identifiants temporaires Pronote (`e`, `f`) → session Pronote (via challenge-response avec clé dérivée de l'ENT).

5.5 Identifiants et jetons

- **Identifiants principaux** : nom d'utilisateur et mot de passe ENT (spécifiques à chaque plateforme ENT).
- **Identifiants délégués** : `e` (connexion temporaire) et `f` (jeton) émis par Pronote après la redirection SSO.
- **Jeton de session** : clé AES-256 standard de Pronote (identique à la Méthode 2, dérivée après résolution du challenge).

5.6 Cycle de vie

- Les sessions ENT sont temporaires : leur durée dépend des politiques de chaque plateforme.
- La session Pronote établie via l'ENT suit le même cycle qu'une session en connexion directe (timeout d'inactivité ∼15–30 min).
- Les tâches automatisées récurrentes doivent se réauthentifier régulièrement auprès de l'ENT.

5.7 Sécurité

§ Analyse de sécurité — SSO ENT

1. **Surface d'attaque** : Les chaînes de redirection multiples (ENT → Pronote) augmentent la surface d'attaque. Chaque redirection est une opportunité d'interception de jetons.
2. **Exposition des identifiants** : Les identifiants utilisateur n'atteignent jamais Pronote directement. L'ENT agit comme intermédiaire de confiance. Le jeton `f` est éphémère (valide uniquement pour l'établissement initial de la session).
3. **Résistance au rejeu** : Le mécanisme challenge-response (identique à la Méthode 2) offre une résistance au rejeu pour la session Pronote. Les jetons de redirection SSO sont à usage unique.
4. **Rotation** : Le cycle de vie de la session ENT contrôle la rotation des identifiants. La clé de session Pronote est renouvelée par session.
5. **Recommandations** : Valider les certificats SSL à chaque étape de redirection. Ne pas journaliser les jetons intermédiaires. Les formulaires de connexion ENT évoluent fréquemment, ce qui peut rompre les clients automatisés.

5.8 Limitations

— Les modifications des formulaires web de connexion ENT, des endpoints SAML ou de l'authentification multifacteur (MFA) rompent souvent les clients automatisés sans interface.
— Chaque ENT possède un flux de connexion différent : aucune automatisation universelle n'est possible.
— Les sessions ENT sont temporaires ; les tâches automatisées récurrentes doivent se réauthentifier régulièrement.
— Certains ENT implémentent du MFA ou des CAPTCHA empêchant une automatisation complète.

5.9 Statut

Implémentation SSO standardisée par Index Éducation et les éditeurs d'ENT, mais la consommation du protocole par des clients tiers est **reverse-engineered**.

6 Méthode 4 : EduConnect

6.1 Principe général

EduConnect est le service national d'authentification et de gestion des accès opéré par le Ministère de l'Éducation Nationale et de la Jeunesse (MENJ). Il agit comme fournisseur d'identité (IdP) pour les élèves et parents en France. L'authentification vers Pronote s'effectue selon deux modes, selon l'infrastructure de l'établissement.

6.2 Flux détaillé

6.2.1 Mode A : EduConnect via ENT régional

1. Le client initie une requête vers la page de connexion de l'ENT régional avec le paramètre `selection=EDU_parent_eleve`.
2. L'ENT redirige vers l'endpoint SAML2 d'EduConnect :

   ```
   https://educonnect.education.gouv.fr/idp/profile/SAML2/Unsolicited/SSO
   ```

3. Le client soumet les identifiants à EduConnect :
   - `j_username` : identifiant EduConnect,
   - `j_password` : mot de passe EduConnect,
   - `_eventId_proceed` : chaîne vide.
4. EduConnect retourne un formulaire `SAMLResponse` signé, posté vers le service de consommation d'assertions de l'ENT (ex. `/Shibboleth.sso/SAML2/POST`).
5. L'ENT établit des cookies de session et redirige vers Pronote (le flux ENT → Pronote suit alors la Méthode 3).

6.2.2 Mode B : HubEduConnect direct (SSO Index Éducation Cloud)

Pour les établissements sans ENT régional :

1. Le client accède à la passerelle CAS centralisée d'Index Éducation :

```
https://hubeduconnect.index-education.net/EduConnect/cas/login?service=<
    URL_INSTANCE_PRONOTE>
```

2. La passerelle initie une requête SAML vers `educonnect.education.gouv.fr`.
3. L'utilisateur s'authentifie sur EduConnect (mêmes identifiants que le Mode A).
4. HubEduConnect reçoit l'assertion, la valide via une liste blanche, et redirige vers Pronote avec un ticket de service.
5. Pronote valide le ticket et établit la session (flux côté Pronote identique à la Méthode 2).

6.3 Données échangées

— **Mode A** : Identifiants EduConnect → assertion SAML2 → cookies ENT → handshake SSO Pronote.
— **Mode B** : Identifiants EduConnect → assertion SAML2 → ticket CAS → session Pronote.

6.4 Identifiants et jetons

— **Identifiants principaux** : identifiants nationaux EduConnect (gérés par le MENJ).
— **Jetons intermédiaires** : assertions SAML2 (Modes A/B), tickets de service CAS (Mode B).
— **Jeton final** : clé de session AES-256 Pronote (identique à la Méthode 2).

6.5 Cycle de vie

— Les sessions EduConnect sont temporaires (durée définie par le MENJ).
— Le ticket CAS (Mode B) est à usage unique et consommé lors de la validation par Pronote.
— La session Pronote suit un timeout d'inactivité standard (∼15–30 min).

6.6 Sécurité

§ Analyse de sécurité — EduConnect

1. **Surface d'attaque** : Chaînes de redirections multiples (EduConnect → ENT/Hub → Pronote). Chaque redirection est un point d'interception potentiel. Les assertions SAML sont signées numériquement, réduisant les risques de contrefaçon.
2. **Exposition des identifiants** : Les identifiants EduConnect sont soumis directement à l'IdP EduConnect et ne transitent jamais vers Pronote ou l'ENT. Seule l'assertion SAML signée est transmise.

3. **Résistance au rejeu** : Les assertions SAML incluent des horodatages et sont à usage unique. Les tickets CAS le sont également.
4. **Rotation** : Gérée centralement par le MENJ. Aucun mécanisme local.
5. **Recommandations** : Valider les signatures des assertions SAML. Ne pas mettre en cache les identifiants EduConnect. Les authentifications 2FA par SMS ou via FranceConnect ne sont pas gérables par des clients automatisés.

6.7 Limitations

— Les authentifications 2FA par SMS ou FranceConnect ne sont pas contournables par des clients programmatiques.
— Le flux repose sur des redirections HTTP multiples, fragiles et sensibles à la gestion des cookies.
— Les identifiants nationaux sont hautement sensibles : leur compromission affecte l'ensemble des services éducatifs connectés.

6.8 Statut

Service officiel (MENJ / Index Éducation) — EduConnect est un service gouvernemental officiel. Cependant, l'interaction programmatique par des clients tiers relève du **reverse engineering**.

7 Méthode 5 : CAS (Central Authentication Service)

7.1 Principe général

CAS (*Central Authentication Service*) est le protocole SSO sous-jacent utilisé dans les réseaux scolaires propriétaires et les ENT régionaux pour autoriser l'accès à Pronote. Protocole standardisé (RFC 4520), il est implémenté côté serveur. Pronote délègue l'authentification au serveur CAS, sans gérer directement les identifiants.

7.2 Flux détaillé

1. **Demande de ticket de service** : Pronote redirige l'utilisateur vers le serveur CAS :

   ```
   https://<cas-server>/login?service=https://<pronote-host>/pronote/<espace>.html
   ```

2. **Authentification CAS** : L'utilisateur soumet ses identifiants (ou effectue une authentification fédérée via EduConnect) sur le formulaire CAS.
3. **Génération du ticket** : CAS renvoie une redirection HTTP 302 vers Pronote avec un paramètre `ticket` :

   ```
   https://<pronote-host>/pronote/<espace>.html?ticket=ST-XXXXX-cas
   ```

   Le ticket, préfixé par `ST-` (*Service Ticket*), est à usage unique.
4. **Validation du ticket de service** : Le backend Pronote contacte directement l'URL de validation CAS pour vérifier le ticket et obtenir les attributs utilisateur :

   ```
   https://<cas-server>/serviceValidate?service=https://<pronote-host>/pronote/<
       espace>.html&ticket=ST-XXXXX-cas
   ```

   Le serveur CAS retourne les attributs (ex. : code UAI de l'établissement, identifiant élève). Pronote génère la session active et injecte le contexte d'initialisation dans le HTML client (mécanisme `onload` identique aux Méthodes 2 et 3).

7.3 Données échangées

Identifiants CAS → Validation serveur CAS → Ticket de service (`ST-XXXXX-cas`) → Réponse de validation (attributs utilisateur : UAI, identifiant élève) → Initialisation de la session Pronote.

7.4 Identifiants et jetons

— **Identifiants principaux** : Nom d'utilisateur et mot de passe CAS (ou identifiants fédérés EduConnect).
— **Jeton** : Ticket de service CAS (`ST-`, à usage unique).
— **Jeton de session** : Clé de session AES-256 Pronote (identique à la Méthode 2).

7.5 Cycle de vie

— Le ticket de service CAS est **à usage unique** : il est consommé lors de la validation et ne peut être réutilisé.
— La session Pronote résultante suit un timeout d'inactivité standard (∼15–30 min).
— La durée de vie de la session CAS est régie par la politique de *Ticket-Granting Ticket* (TGT) du serveur CAS.

7.6 Sécurité

§ Analyse de sécurité — Protocole CAS

1. **Surface d'attaque** : Protocole standardisé et documenté. La surface d'attaque concerne principalement le formulaire de connexion CAS et la transmission du ticket (protégée par HTTPS).
2. **Exposition des identifiants** : Les identifiants sont soumis uniquement au serveur CAS — Pronote ne les voit jamais. Seul le ticket de service est transmis à Pronote.
3. **Résistance au rejeu** : Élevée — les tickets de service sont à usage unique et strictement liés à une URL de service spécifique.
4. **Rotation** : La rotation des TGT est gérée par la politique du serveur CAS. Les tickets de service expirent rapidement (généralement en quelques secondes ou minutes).
5. **Recommandations** : Utiliser systématiquement HTTPS. Valider le paramètre `service` pour éviter le vol de tickets via des URL de service malveillantes. Appliquer une gestion rigoureuse du cycle de vie des TGT.

7.7 Limitations

— CAS est un protocole côté serveur : le serveur CAS doit être correctement configuré et accessible.
— L'appel `serviceValidate` s'effectue de serveur à serveur (backend Pronote → serveur CAS), nécessitant une connectivité réseau directe entre eux.
— Toutes les écoles n'utilisent pas CAS : certaines privilégient SAML ou des solutions SSO personnalisées.

7.8 Statut

Officiel — CAS est un protocole standardisé (RFC 4520) officiellement implémenté côté backend Pronote et serveurs CAS.

8 Méthode 6 : QR Code et jeton mobile

8.1 Principe général

Pronote propose un mécanisme d'appairage par QR code pour connecter les appareils mobiles sans saisir les identifiants ENT complexes. L'utilisateur génère un QR code depuis l'interface web, définit un code PIN temporaire à 4 chiffres, puis le scanne avec l'application mobile. Le QR code contient des identifiants chiffrés qui, une fois déchiffrés, permettent un échange de jeton à longue durée de vie. Ce mécanisme contourne entièrement l'authentification ENT/EduConnect.

8.2 Flux détaillé

Phase 1 — Génération (interface web Pronote)

1. Dans l'interface web, l'utilisateur accède à *Paramètres → Accès mobile / Application mobile*.
2. Il définit un code PIN temporaire à 4 chiffres.
3. Pronote affiche un QR code contenant un JSON chiffré :

```
{
  "login": "<AES_HEX_encrypted_username>",
  "jeton": "<AES_HEX_encrypted_token>",
  "url": "https://<host>/pronote/mobile.eleve.html"
}
```

Phase 2 — Déchiffrement

- Algorithme : **AES-256-CBC**.
- IV : 16 octets nuls (`0x00...`).
- Clé : `MD5(PIN_code_string)` (ex. `MD5("1234")`).
- Résultat : `login` et `jeton` en clair.

Phase 3 — Échange d'appairage initial

1. Le client génère un UUID permanent (`uuidAppliMobile`).
2. Il envoie une requête vers l'endpoint mobile :

```
https://<host>/pronote/mobile.<espace>.html?login=true
```

(contourne la redirection ENT).

3. Requête `Identification` avec :
   — `demandeConnexionAppliMobile: true`
   — `demandeConnexionAppliMobileJeton: true`
   — `uuidAppliMobile: "<DEVICE_UUID>"`
   — `identifiant: "<decrypted_login>"`
4. Le défi est résolu avec `<decrypted_jeton>` comme mot de passe (mécanisme identique à la Méthode 2).
5. Le serveur retourne `jetonConnexionAppliMobile` (jeton à longue durée de vie).

Phase 4 — Connexions ultérieures (auto-login)

— `identifiant: <decrypted_login>`
— `uuidAppliMobile: <DEVICE_UUID>`
— `enConnexionAppliMobile: true`
— Mot de passe pour le défi : `jetonConnexionAppliMobile`
— À chaque connexion réussie, Pronote retourne un nouveau `jetonConnexionAppliMobile` à conserver.

8.3 Données échangées

QR code JSON (login + jeton chiffrés + URL) → déchiffrement → `Identification` avec drapeaux mobiles → défi-réponse → `jetonConnexionAppliMobile`.

8.4 Identifiants et jetons

— **Initial** : Code PIN à 4 chiffres (temporaire, utilisé uniquement pour le déchiffrement du QR code).
— **Déchiffrés** : `login` et `jeton` extraits du QR code (usage unique pour l'appairage initial).
— **Persistants** : `uuidAppliMobile` (UUID de l'appareil, permanent) + `jetonConnexionAppliMobile` (jeton à longue durée de vie, rafraîchi à chaque connexion).

8.5 Cycle de vie

— Le QR code est valide **10 minutes** après génération.
— Le `jetonConnexionAppliMobile` reste valide indéfiniment (souvent toute l'année scolaire), sauf :
   — Révoqué par l'utilisateur dans les paramètres Pronote.
   — Invalidé côté serveur.
— Le jeton est rafraîchi à chaque connexion réussie.

8.6 Sécurité

§ Analyse de sécurité — Appairage QR Code Mobile

1. **Surface d'attaque** : Le QR code est affiché à l'écran (risque de *shoulder-surfing*). Le PIN à 4 chiffres offre seulement 10 000 combinaisons. Le `jetonConnexionAppliMobile` est un identifiant à longue durée de vie stocké sur l'appareil.
2. **Exposition des identifiants** : Le QR code contient des identifiants chiffrés. Si intercepté avant déchiffrement, l'attaquant a besoin du PIN. Une fois le `jetonConnexionAppliMobile` obtenu, aucun PIN ni mot de passe n'est requis.
3. **Résistance au rejeu** : Le `jetonConnexionAppliMobile` est un *bearer token* : toute partie en possession du jeton peut s'authentifier. Le `uuidAppliMobile` offre un lien faible avec l'appareil, mais non vérifié cryptographiquement.
4. **Rotation** : Le `jetonConnexionAppliMobile` est rafraîchi à chaque connexion, mais l'ancien reste valide jusqu'à invalidation côté serveur. Aucune expiration automatique.
5. **Recommandations** : Utiliser un PIN robuste. Traiter le `jetonConnexionAppliMobile` comme un identifiant à longue durée de vie (stockage sécurisé). En cas de vol de l'appareil, révoquer l'accès mobile dans Pronote.

8.7 Limitations

— Le QR code expire après **10 minutes** : l'appairage doit être rapide.
— Le PIN à 4 chiffres est faible selon les normes modernes.
— Le `jetonConnexionAppliMobile` n'a pas d'expiration automatique : il persiste jusqu'à révocation manuelle.
— Le `uuidAppliMobile` n'est pas lié cryptographiquement à l'appareil : il peut être copié.

8.8 Statut

Mécanisme **officiellement intégré** à l'application mobile Index Éducation. L'utilisation par des clients tiers repose sur de l'**ingénierie inverse**.

9 Méthode 7 : API officielle et application mobile

9.1 Principe général

Index Éducation ne propose pas d'API publique pour Pronote. L'accès tiers aux données repose sur l'ingénierie inverse des mêmes endpoints JSON-over-HTTP(S) utilisés par les clients web et mobile officiels. L'application mobile officielle s'authentifie via le mécanisme d'appairage par QR Code (Méthode 6) et communique via le même protocole propriétaire que le client web.

9.2 Flux détaillé

L'application mobile s'authentifie via le flux d'appairage par QR Code (Méthode 6) puis communique via le même protocole JSON-over-HTTP(S) que le client web, en utilisant les endpoints :

```
/pronote/mobile.<espace>.html
/pronote/appelfonction/<espace_id>/<session_id>/<numeroOrdre>
```

9.3 Disponibilité d'une API développeur

— **API publique** : Inexistante. Index Éducation ne propose ni API REST ni GraphQL pour les élèves, parents ou développeurs tiers.
— **API institutionnelle / entreprise** : Des services d'intégration propriétaires sont proposés pour les systèmes partenaires (ex. connecteurs UDTS, HYPERPLANNING, ENT officiels). Ceux-ci nécessitent des accords de partenariat signés et des licences serveurs institutionnelles.

9.4 Authentification de l'application mobile officielle

L'application mobile officielle (iOS/Android) se connecte via les mêmes endpoints JSON-over-HTTP(S) que l'interface web mobile :

— Chemins de base : `/pronote/mobile.<espace>.html`

— Dispatcher de fonctions : `/pronote/appelfonction/<espace_id>/<session_id>/<numeroOrdre>`
— Authentification : Utilise le flux d'appairage par QR Code (Méthode 6) et le jeton mobile persistant (`jetonConnexionAppliMobile` + `uuidAppliMobile`).

9.5 Données échangées

Mêmes charges utiles JSON chiffrées en AES-256-CBC que le client web (Méthode 2). La variante mobile (`mobile.<espace>.html`) contourne les redirections SSO ENT/EduConnect.

9.6 Identifiants et jetons

— `jetonConnexionAppliMobile` : Jeton bearer à longue durée de vie (voir Méthode 6).
— `uuidAppliMobile` : UUID de l'appareil.
— Clé de session : Clé AES-256 (même dérivation que la Méthode 2).

9.7 Cycle de vie

Le `jetonConnexionAppliMobile` est rafraîchi à chaque connexion. La durée de vie de la session suit le même délai d'inactivité (~15–30 min) que le client web.

9.8 Sécurité

§ Analyse de sécurité — API & Application Mobile

1. **Surface d'attaque** : Les endpoints mobiles sont accessibles publiquement. Aucune clé API ni enregistrement développeur n'existe — néanmoins, l'accès exige un matériel de session Pronote valide et, pour l'auto-login mobile, le matériel d'authentification correspondant (`login`, `uuidAppliMobile`, `jetonConnexionAppliMobile`) issu du flux d'appairage par QR Code (Méthode 6).
2. **Exposition des identifiants** : Le `jetonConnexionAppliMobile` est un jeton bearer stocké sur l'appareil. S'il est extrait, il accorde un accès complet jusqu'à révocation.
3. **Résistance au rejeu** : Faible — le jeton est basé sur un bearer. Le `uuidAppliMobile` offre un lien faible avec l'appareil, non appliqué cryptographiquement.
4. **Rotation** : Le jeton est rafraîchi à chaque connexion, mais les anciens restent valides. Aucune

expiration automatique.

5. **Recommandations** : Évitez l'ingénierie inverse du protocole pour un usage en production sans comprendre les implications légales. Stockez les jetons mobiles de manière sécurisée. Soyez conscient qu'Index Éducation peut modifier le protocole à tout moment.

9.9 Limitations

— Aucune documentation ou support officiel pour les développeurs tiers.
— Protocole propriétaire susceptible de changer sans préavis.
— Les implémentations basées sur l'ingénierie inverse peuvent cesser de fonctionner après les mises à jour de Pronote.
— Le statut légal de l'ingénierie inverse est incertain dans certaines juridictions.

9.10 Statut

— API publique : **Inexistante**.
— Endpoints mobiles : **Ingénierie inverse** (même protocole que le client web, accessible via appairage QR Code).

A Bibliothèques open source tierces

Les bibliothèques open source répertoriées dans le tableau A.1 implémentent les protocoles d'authentification Pronote décrits dans ce manuel. Ces projets sont maintenus par la communauté et ne sont pas affiliés à Index Éducation. Les fonctionnalités décrites reposent sur les informations publiques disponibles dans leurs dépôts et peuvent avoir évolué depuis la rédaction de ce document.

Bibliothèque	Langage	Dépôt	Méthodes prises en charge
pronotepy	Python	`bain3/pronotepy`	Connexion directe, jeton mobile / QR code, CAS SSO, EduConnect (HubEduConnect direct et via 30+ ENT régionaux : Open ENT NG, Skolengo, Oze, Shibboleth/WAYF).
pronote-api	TypeScript / JS	`Litarvan/pronote-api`	Connexion directe, CAS SSO, redirections ENT, authentification par jeton.
pronote-qrcode-api	JavaScript	`Androz2091/pronote-qrcode-api`	Implémentation de référence pour le déchiffrement des QR codes Pronote et l'échange initial de jeton mobile.
pawnote / Blocksnote	TypeScript / JS	`BlocksHub/Blocksnote`	Clients modernes implémentant le protocole Pronote complet (direct, ENT, QR code, jetons).

Table A.1 : Bibliothèques open source de référence pour l'écosystème Pronote.

> **! Risques liés aux intégrations tierces**
>
> Ces bibliothèques illustrent la faisabilité des méthodes d'authentification présentées. Leur utilisation en environnement de production comporte des risques : les modifications de protocole par Index Éducation peuvent rendre les implémentations obsolètes sans préavis, et le statut juridique de l'ingénierie inverse des protocoles propriétaires varie selon les juridictions.

B Tableau comparatif synthétique

Cette annexe consolide les caractéristiques clés des 7 méthodes d'authentification Pronote en un tableau de référence unique, incluant l'ensemble des dimensions d'analyse de sécurité.

Méthode	Périmètre	Identifiants	Expiration	Complexité	Surface d'attaque	Exposition	Rejeu	Rotation	Recommandation	Statut
URL iCal sécurisée	EDT + devoirs (si inclus)	Jeton URL	Longue durée	Très faible	Jeton dans URL (logs, referers)	Credential longue durée	Faible (pas de nonce)	Manuelle	Stocker en secret, HTTPS, renouveler à la rentrée	Officiel
Connexion directe	Complet	Identifiant + mot de passe	Session ∼15–30 min	Élevée	Endpoint public, crypto custom	Mot de passe non transmis (challenge)	Modérée (alea aléatoire)	Session seulement	HTTPS obligatoire, gérer `numeroOrdre`	Reverse-engineered
SSO ENT	Complet	Identifiants ENT	Session ENT	Très élevée	Redirections multiples	Credentials via ENT (intermédiaire)	Jetons SSO single-use	Session ENT	Valider SSL à chaque redirection	Reverse-engineered
SSO EduConnect	Complet	Identifiants nationaux	Session EduConnect	Très élevée	Redirections EduConnect → ENT/Hub	Credentials MENJ (très sensibles)	SAML single-use + timestamps	MENJ	Ne pas cacher credentials, 2FA bloque automation	Reverse-engineered
CAS	Complet	Identifiants CAS	Ticket single-use	Modérée	Login form CAS	Credentials via CAS uniquement	Ticket single-use	TGT (politique CAS)	Valider paramètre `service`, HTTPS	Officiel
QR Code + jeton mobile	Complet	PIN 4 chiffres → jeton + UUID	QR 10 min ; jeton long terme	Modérée	QR écran, PIN faible, jeton bearer	Jeton long terme stocké sur appareil	Faible (bearer)	À chaque login (ancien reste valide)	PIN fort, stockage sécurisé, révoquer si perte	Reverse-engineered
API publique	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Inexistante

Table B.1 : Synthèse globale et analyse multidimensionnelle de la sécurité des méthodes d'authentification Pronote.

Légende :

- *Complet* : accès aux notes, emploi du temps, devoirs, absences, messagerie et paramètres.
- *Reverse-engineered* : protocole non publié officiellement par Index Éducation, basé sur l'analyse communautaire.
- *Officiel* : mécanisme fourni et pris en charge par Index Éducation.