



---

# RAPPORT D'ÉVALUATION DES FORMALISMES DE DONNÉES

---

## Projet SMILE, Chantier SEN1 – phase POC



Tableau de suivi des versions

Version	1			
Auteurs	Gautier			
Date	23/04/2019			
Commentaires	Version initiale			

Licence du document : EUPL v1.2, voir : <https://joinup.ec.europa.eu/collection/eupl/eupl-text-11-12>

Diffusion : illimitée

Contact : pro \_@\_ consometers.org

# TABLE DES MATIÈRES

Introduction.....	3
Définitions.....	3
Démarche de sélection.....	4
Critères d'évaluation.....	4
Critères d'universalité.....	4
Critères fonctionnels.....	5
Critères de propriété.....	5
Critères isolé : efficacité.....	5
Pondération et matrice de choix.....	6
Formalismes évalués.....	6
Résultat d'évaluation.....	7
Choix final.....	7
Choix d'un formalisme décrit en XML : SENML.....	8
Conclusion.....	9
ANNEXE 1 : Références des formalismes évalués.....	10
ANNEXE 2 : Matrice d'évaluation finale.....	11

## INTRODUCTION

Ce document présente la première partie du travail réalisé dans le cadre d'une preuve de concept (proof of concept - POC) pour le projet SEN1 – « Sensibilisation aux consommations d'énergie ». Il s'agit de la démarche suivie pour retenir, parmi l'ensemble des possibilités normatives disponibles, la norme qui conviendra le mieux pour l'échange de données entre les serveurs fédérés au sein de notre POC.

Pour réaliser les objectifs de SEN1, et en accord avec le Demandeur, nous avons souhaité mettre en place une **fédération technique**, qui permet un échange de données entre serveurs pairs (appelés **nœuds**, au sens topologique, dans le reste du document), sans lien de subordination entre eux. Une fois les problématiques de protocole de communication réglées, nous avons souhaité normaliser le formalisme de données échangées. Cela dans le but que tous les nœuds qui échangeront des données parlent la « même langue », soit avec le même formalisme de données.

Conformément à la charte « PrestaLibre », nous avons effectué ce travail de manière ouverte et documentée, en lien avec le collectif « Consometers ». Les documents sont disponibles sur le site <https://www.consometers.org> et diffusés selon la licence EUPL v1.2 permettant une libre réutilisation.

## DÉFINITIONS

Le(s) Prestataire(s) est(sont) une(des) Société de Services en Informatique. A ce titre, les Demandeurs mettent à disposition du Commanditaire le résultat d'un travail. Ce travail est cadré par un planning prévisionnel, le détail des tâches, son chiffrage est justifié par des devis et des factures. Ce travail est réalisé par les Prestataires, appuyés par le collectif bénévole des Consometers

Demandeur : ALOEN et l'ALEC du Pays de rennes

Commanditaire : Région Bretagne

Collectif des Consometers : Collectif bénévole qui promeut l'utilisation des logiciels libres autour des besoins d'affichage et de maîtrise de la consommation d'énergie, et l'acculturation sur les logiciels libres, voir : <https://cloud.consometers.org/index.php/s/3665sZsz5xz8FS3>

Documentation : Manuels techniques et d'information afférents aux résultats de Prestations et mis à disposition du Demandeur par les Prestataires.

Internet : réseau de réseaux permettant l'échange d'informations à partir d'un protocole dénommé IP. Les données sont acheminées à travers des réseaux de natures différentes qui sont capables de transmettre les messages selon cette norme technique. Chaque élément de ce réseau appartient à des organismes privés ou publics qui les exploitent en coopération sans nécessairement impliquer une obligation bilatérale de qualité.

## DÉMARCHE DE SÉLECTION

Après la définition des critères et la constitution d'une matrice de choix pondérées, 8 normes ont été évaluées sur la base de leur documentation disponible publiquement. Ces 8 normes ont été trouvées par recherche Internet et lecture de documents spécialisés.

## CRITÈRES D'ÉVALUATION

L'objectif de cette tâche était de définir un ensemble de critères pouvant être évalués objectivement, avec par conséquent une définition et un score (de 0 à 5) explicites. La pondération a été déterminée conjointement par deux profils techniques : un administrateur système et un développeur.

Les critères d'évaluation des protocoles ont été organisés en trois catégories : les critères représentant l'universalité de la norme, les critères fonctionnels et les critères de propriété intellectuelle. Enfin, un critère isolé représentant l'efficience a été retenu, bien que son évaluation soit moins aisément objective.

### Critères d'universalité

Ces critères représentent l'adoption d'une norme, la rendant de fait universelle. Deux critères ont été retenus.

Le critère « librairies existantes » représente la disponibilité de librairies de code logiciel dans les trois langages de programmation les plus utilisés, selon les informations disponibles ici : <https://www.developpez.com/actu/185087/Quels-sont-les-langages-de-programmation-les-plus-utilises-par-les-developpeurs-Une-analyse-des-evenements-publics-sur-GitHub/>). Les trois langages sont JavaScript, Python et Java. Si une norme ne dispose pas d'au moins une librairie dans un de ces trois langages, elle est éliminée.

Le critère « présence WEB / utilisation large » représente le nombre de pages web publiques présentant le nom de la norme recherchée. Le résultat nous donne une idée de l'adoption de la norme.

## Critères fonctionnels

Ces critères évaluent la pertinence de la solution au niveau technique. Quatre critères ont été retenus.

Le critère « types de données gérés » représente le nombre de types de structures de données (manière d'organiser les informations) représentables par la norme.

Le critère « épreuve du temps » représente le fait que la norme soit extensible et/ou générique. Une norme fixe ne sera pas à l'épreuve du temps car on ne pourra pas en étendre les usages. Ce critère est donc éliminatoire.

Le critère « typage des données » représente le fait que la manière de coder les informations avec des 1 et des 0 (langage binaire, tables de codage, représentation d'une donnée au sens informatique) est d'une part extensible et d'autre part fortement normée. Pour des échanges entre systèmes divers, nous avons besoin de typer fortement les données (ex : manière de représenter une date). Cela évitera les problèmes d'interprétation des données échangées. Par ailleurs nous souhaitons également pouvoir étendre les types de données représentables (ce qui rejoint le critère « épreuve du temps »).

Le critère « simplicité de mise en œuvre » représente la simplicité d'utiliser la norme. Une norme avec une certaine complexité provoquera des réticences à l'adoption.

## Critères de propriété

Ces deux critères représentent les droits d'auteur liés aux normes et à leur utilisation ainsi que la paternité.

Le critère « type d'organisme éditeur » représente la structure porteuse de la norme. Plus la structure porteuse est connue et reconnue à l'international, meilleure est l'évaluation.

Le critère « licence de la normalisation » représente la licence sous laquelle est distribuée la norme de formalisation des données. A noter qu'une norme payante (à l'utilisation ou à l'obtention) n'est pas viable dans notre projet. Ce critère est donc éliminatoire.

## Critères isolé : efficience

Le critère « efficience » représente le seul critère noté subjectivement. Il aurait fallu passer plusieurs jours de travail pour définir des jeux de tests et comparer réellement les formalismes entre eux en utilisant plusieurs bibliothèques de codes. Nous avons choisi de nous baser sur l'expérience afin de juger si un formalisme est efficace au sens de la taille des données à transmettre (influe sur la bande passante réseau) et de la complexité algorithmique d'encodage/décodage (influe sur l'utilisation de processeur et de mémoire informatique). Nous nous sommes basés sur la notion d'*overhead* (information d'encodage autour de la donnée utile) afin de noter les formalismes de données.

## PONDÉRATION ET MATRICE DE CHOIX

L'ensemble des critères a été pondéré conjointement entre un profil d'ingénieur système et un profil d'ingénieur en développement informatique. Certains de ces critères ont été jugés éliminatoires car ne correspondant pas aux besoins du projet :

- universalité – librairies existantes
- fonctionnel – épreuve du temps
- propriété – licence de la normalisation

Les normes présentant une note de 0 sur ces critères ont donc été éliminées, sans besoin de réaliser leur évaluation sur l'ensemble des critères.

## FORMALISMES ÉVALUÉS

Une phase de recherche sur différents moteurs de recherche et dans divers corpus, nous ont permis d'identifier les formalismes pertinents à ce jour. Ces formalismes doivent être bien définis, avoir des librairies d'implémentation disponibles sous licence libre et être extensibles.

Dans les formalismes évalués, il y a :

- des formalismes directs et cadrés, liés à une utilisation particulière (ex : warp10, haystack)
- des méta-formalismes qui donnent une méthode pour décrire et analyser un formalisme que nous définissons nous-même (ex : JSON, XML)
- un méta-méta formalisme, qui donne une méthode pour décrire et analyser un formalisme selon plusieurs méta-formalismes (ex : ASN-1 qui peut donner une description XML ou JSON par exemple)

Le moteur de recherche utilisé est duckduckgo.com. Les termes de recherche ont été les suivants :

- iot data encoding
- sensor data encoding transmit
- iot data format
- data formalism
- data exchange standard format
- data interchange standard format
- open data standard france

Les deux pages web nous ayant donné le plus de candidats sont :

- [https://en.wikipedia.org/wiki/List\\_of\\_open\\_formats#Various](https://en.wikipedia.org/wiki/List_of_open_formats#Various)
- [https://en.wikipedia.org/wiki/Comparison\\_of\\_data\\_serialization\\_formats](https://en.wikipedia.org/wiki/Comparison_of_data_serialization_formats)

Les formalismes identifiés et évalués sont au nombre de 9. Leurs références (sites web officiels, code source public et documentation) sont présentées en ANNEXE 1 : Références des formalismes évalués.

## RÉSULTAT D'ÉVALUATION

Chaque protocole a été évalué par notre profil administrateur système. Puis les 4 meilleurs profils (JSON, XML, ASN-1, haystack) ont été revus avec notre profil développeur. L'ensemble des résultats est disponible en ANNEXE 2 : Matrice d'évaluation finale. La note maximale est de 340.

Les trois lauréats de cette phase sont : XML (288 points) , ASN-1 (286 points) et haystack (282 points).

A noter que JSON, un formalisme en vogue, n'est pas dans le top 3 car il présente certains défauts :

- *overhead* (informations à transmettre en plus de l'information utile : espaces d'indentation, caractères de séparation, encodage inefficace au sens de la taille, ...) vis à vis d'un encodage XML EXI (voir : <https://www.w3.org/TR/exi/>).
- typage de données (manière prédéfinie d'encoder différents types de valeurs, exemple : nombre entiers ou non entiers) insuffisamment défini (ex : pas d'encodage défini pour les dates)

## CHOIX FINAL

Après réflexion, il est apparu que :

- ASN-1 est un méta-méta formalisme. Bien qu'intéressant à avoir, il n'est pas nécessaire et s'éloigne trop du besoin. Si un méta-formalisme est trouvé, sa description en ASN-1 est réalisable sauf complexité exceptionnelle.
- Haystack est très orienté internet des objets et données industrielles. Bien qu'il corresponde au besoin actuel, il pourra présenter des limites ultérieurement.

**XML**, arrivé en tête du classement, est finalement retenu. C'est un méta-formalisme. Il faut donc trouver un formalisme décrit en XML.

## CHOIX D'UN FORMALISME DÉCRIT EN XML : SENML

La matrice de choix nous a ainsi orienté vers le méta-formalisme XML. En XML nous pouvons décrire nous-même notre formalisme dans un cadre normalisé. Nous avons souhaité éviter de créer notre propre formalisme en XML pour des raisons de moyens et d'optimisation (nous préférons éviter de réinventer la roue). Lors de nos recherches nous avons remarqué le formalisme Sensor Measurement Lists (SenML). SENML est défini au sein de l'Internet Engineering Task Force (IETF : organisation internationale responsable de définir des normes techniques pour l'architecture et le bon fonctionnement d'Internet). Après plusieurs itérations depuis juillet 2010, SENML a été ratifié en version finale en août 2018 sous la RFC (Request For Comments, méthode de publication de l'IETF) numéro 8428, disponible ici : <https://www.rfc-editor.org/rfc/pdf/rfc8428.txt.pdf>

Ce formalisme, décrit selon le méta-formalisme XML, nous semble adapté à nos besoins. Il est orienté internet des objets. Il peut aussi bien servir à transmettre des lots de données que des données unitaires simple issues directement de capteurs. SENML a été co-rédigé par des industriels (CISCO, ARM, Erricsson), l'université Technology Center Informatics and Information Technology de Brême et quelques personnes à titre personnel. Le premier brouillon de spécification (RFC) a été mis à disposition le 5 avril 2016. La version finale de cette spécification date d'août 2018. Enfin SENML est suffisamment générique pour être à l'épreuve du temps.

## SPÉCIFICATIONS PARTICULIÈRES À SENML DANS LE CADRE DE SEN1

Afin de faire circuler les données dans le réseau fédéré, il est nécessaire de définir une manière de nommer les données qui circulent. Le protocole SENML définit la notion de *Name*. Nous souhaitons spécifier une manière d'écrire ce *Name* afin qu'il soit unique pour chaque valeur remontée par un capteur lié à la fédération. L'intérêt est de pouvoir transmettre les données au sein de la fédération de manière fluide et sans risque de confusion. Nous proposons le format suivant pour la valeur de *Name* :

```
<nom de l'application>@<nom du serveur XMPP 1 de l'organisme A>/<nom ou ID de  
l'utilisateur ou du capteur>@<nom du serveur XMPP 2 de l'organisme B>/<identification de  
la donnée qui peut avoir plusieurs sous-niveaux séparés par des />
```

Nous imposons donc d'avoir dans les données transmises deux informations relatives à la donnée :

- <nom de l'application>@<nom du serveur XMPP 1 de l'organisme A> : permet de savoir quelle application de quel serveur a injecté la donnée dans la fédération, et permet donc de donner le contexte de qui est responsable de la transmission initiale
- <nom ou ID de l'utilisateur ou du capteur>@<nom du serveur XMPP 2 de l'organisme B> : Dans le cas d'une donnée transférée au sein de la fédération, cela permet de savoir quel est l'utilisateur concerné par la donnée (sous forme de nom lisible ou d'identifiant numérique à



des fins d'anonymisation) et sur quel serveur la donnée a été initialement reçue (passerelle ou point d'entrée à la fédération). Dans le cas d'une donnée issue directement d'un capteur, on utilise de préférence un identifiant matériel unique (ex : adresse MAC de la carte réseau) ou le nom de l'utilisateur à défaut.

- A noter : l'organisme A et B peuvent être le même. Le serveur 1 et le serveur 2 peuvent être le même.

Par ailleurs nous donnons de la liberté concernant le **chemin logique de la valeur transférée**. Cela permet aux développeurs d'utiliser plusieurs niveaux logiques d'organisation. Cela permet également aux applications réceptrices de la donnée d'adapter comme elles le souhaitent leur affichage. Par exemples, la donnée Température d'un capteur d'un utilisateur peut être représentée sous différents formes, au libre choix du développeur :

- /Temp
- /Salon/Temp
- /Maison\_principale/Salon/Temp

Cependant, le chemin d'une valeur ne doit pas être par exemple : /MaisonPrincipale\_Salon\_Temp car elle ne permet pas de reconstruire les niveaux logiques d'organisation.

#### Quelques exemples complets :

Pour un capteur :

```
bmhs@sen1.bmhs.com/capteur32048234932748239@sensors.bmhs.com/maison1/pi2/dht22_1/Temp1
```

Pour un proxy de données :

```
bmhs@sen1.bmhs.com/user32490384239473947@bmhs.com/maison1/pi2/dht22_1/Temp1
```

Pour un proxy PRIDE :

```
sen1-proxies-pride@breizh-sen2.eu/<data owner  
ID>@<warp10_endpoint>/14803617839618/pride.enedis.cdc
```

Ce format nous permet aussi bien de transférer des données au sein de la fédération que d'injecter directement des données depuis des capteurs. Cela permet donc de simplifier l'architecture générale d'un service lié à de la donnée issue d'objets connectés.

La fédération au sens technique, utilisant le protocole XMPP permet le transfert d'informations directement entre les nœuds XMPP. Il ne doit pas avoir de routage d'informations à priori (chaînage à travers des nœuds de la fédération). Cependant, si ce cas se présentait, on souhaiterait avoir la traçabilité du routage d'informations. Nous pourrions utiliser une valeur dédiée au sein des informations transmises selon le formalisme SENML. Mais cela n'est pas directement lié aux informations. Nous préférons mettre cette information de routage dans les méta-données gérées à un autre niveau (là où nous traiterons le consentement, la facturation, etc), ou alors au niveau du protocole XMPP directement.

## CONCLUSION

Malgré la popularité du méta-formalisme JSON, c'est XML que nous avons retenu. Le critère fort de ce choix est la volonté d'avoir des typages de données forts, afin de ne pas laisser de doute quant à l'encodage des données et ainsi diminuer les risques d'erreurs de décodage entre les services de la fédération. Nous avons sélectionné **le formalisme SENML**, décrit en méta-formalisme XML et EXI. A noter que SENML décrit complètement les types de données (ex : une date est codée selon la méthode epoch Unix) et est décrit également avec les méta-formalismes suivants : JSON, CBOR. Nous pourrions donc utiliser JSON si nécessaire.

## ANNEXE 1 : RÉFÉRENCES DES FORMALISMES ÉVALUÉS

### 1 JSON

<https://json.org/>

<https://en.wikipedia.org/wiki/JSON>

### 2 XML

<https://www.w3.org/XML/Core/>

<https://en.wikipedia.org/wiki/XML>

### 3 UBJSON

<http://ubjson.org/>

### 4 REBOL

<https://curlie.org/Computers/Programming/Languages/REBOL>

<https://en.wikipedia.org/wiki/Rebol>

### 5 haystack

<https://project-haystack.org/doc/Zinc#overview>

### 6 warp10

<https://www.warp10.io>

### 7 SDXF

<https://tools.ietf.org/html/rfc3072>

<https://en.wikipedia.org/wiki/SDXF>

### 8 ASN-1

<https://www.itu.int/en/ITU-T/asn1/Pages/introduction.aspx>

[https://en.wikipedia.org/wiki/Abstract\\_Syntax\\_Notation\\_One](https://en.wikipedia.org/wiki/Abstract_Syntax_Notation_One)

### 9 Prometheus

[https://prometheus.io/docs/instrumenting/exposition\\_formats/](https://prometheus.io/docs/instrumenting/exposition_formats/)

## ANNEXE 2 : MATRICE D'ÉVALUATION FINALE

					JSON		XML		UBJSON		REBOL		haystack		warp10		SDXF		ASN-1		Prometheus	
Critère	Sous-critère	Définition	Coef.	Elim.	Notes	Rés	Notes	Rés	Notes	Rés	Notes	Rés	Notes	Rés	Notes	Rés	Notes	Rés	Notes	Rés	Notes	Rés
universalité	librairies existantes	Librairies de code libre existant dans JavaScript, Python et Java ( <a href="https://www.developpez.com/actu/185087/">https://www.developpez.com/actu/185087/</a> ) ? les 3 : 5 / 2 : 2 / 1 : 1 / 0 : 0	10	1	5	50	5	50	5	50	0	0	5	50	5	50	0	0	5	50		
universalité	présence WEB / utilisation large	Recherche sur <a href="http://www.google.fr">www.google.fr</a> . > 100 millions : 5 / > 50 millions : 4 / > 1 millions : 3 / > 500 milles : 2 / sinon 1	8		5	40	5	40	1	8			3	24	1	8			3	24		
efficience	overhead	Overhead approximatif : Bas : 5 / moyen : 3 / grand : 1	6		1	6	1	6	1	6			3	18	3	18			5	30		
fonctionnel	types de données gérés	Chaînes/nombres : 1 / + tableaux : 3 / + dictionnaires : 4 / + structure d'objet : 5	10		5	50	5	50	5	50			4	40	1	10			5	50		
fonctionnel	épreuve du temps	Est ce que le formalisme est extensible/générique : oui = 5 / non = 0	10	1	5	50	5	50	5	50			5	50	5	50			5	50	0	0
fonctionnel	typage des données	Est ce que les données sont fortement typées/normées ou non ? Fortement typées et extensibles : 5 / fortement typées et fixe : 3 / faiblement typées : 1	5		1	5	5	25	3	15			3	15	3	15			3	15		
fonctionnel	simplicité de mise en œuvre	Est il facile de mettre en œuvre le formalisme ? Facile : 5 / moyen : 3 / difficile : 1	7		3	21	3	21	3	21			5	35	5	35			3	21		
propriété	type d'organisme éditeur	1-specs publiques 3-rédaction commune 5-passé par un organisme international	5		5	25	5	25	1	5			3	15	1	5			5	25		
propriété	licence de la normalisation	Licence du document de spécification : libre : 5 / limité : 3 / payant : 0	7	1	3	21	3	21	5	35			5	35	3	21			3	21		
		Résultat				268		288		240		0		282		212		0		286		0