

## **Flex Ready® API – CEMS / Agrégateur**

*Ce document définit les API utilisées pour interfacier les CEMS (Customer Energy Management Systems) avec les opérateurs de flexibilité. Dans tous le document, ces opérateurs sont appelés "Service Provider", conformément à la terminologie 62746-4. Quand le bâtiment dispose d'un BACS (Building Automation and Control System), notamment quand la Directive EPBD l'exige, le CEMS est une fonction du BACS. Dans ce cas, on peut utiliser indifféremment le terme CEMS ou BACS.*

*Ces API sont définies en fonction de niveaux de fonctionnalité identifiés pour la fonction BEMS.*

*Chaque niveau de fonctionnalité correspond à un engagement contractuel.*

*Ce document est une première spécification de l'interface CEMS-Agrégateur. Elle intègre les éléments de la norme IEC 62746-4 qui sont pertinents sur le cas d'usage du bâtiment. Cette spécification pourra évoluer pour enrichir les services de flexibilité pris en compte et pour suivre les évolutions de la norme. Dans la mesure du possible, une compatibilité ascendante sera assurée.*

### **1. Ce qu'est l'API**

Une API (Application Programming Interface) est l'interface que propose un serveur pour donner accès ou recevoir des informations d'un logiciel client externe.

Une API est donc composée :

- a : d'un logiciel sur le serveur (type script php) qui va exécuter une requête et retourner une informations
- b : d'une définition du format de la requête envoyée (par exemple sous forme de POST ou GET https),
- c : de la façon dont le client appelle cette fonctionnalité : c'est la partie visible de l'API,
- d : d'une implémentation de cet appel de fonction (par exemple sous forme de librairie fournie).

Par exemple, une API pour demander une opération de flexibilité par le Service Provider vers un CEMS selon un profil donné :

**c: func askFlex(requestID: UUID, cems: CEMSID, asset: AssetID, flexProduct: FlexProductType, puissance: PowerTimeSeries) → Ack**

- Le Service Provider envoie une référence de requête, l'ID du CEMS, l'asset concerné, le produit de flexibilité et le profil de puissance.
- En retour, le CEMS répond en acceptant ou refusant la demande.

a : le CEMS analyse la demande pour savoir s'il peut l'accepter.

b et d : c'est l'implémentation de la requête, qui pourra être disponible dans des librairies ("stack FlexReady").

### **2. Rappel : Paramètres caractéristiques du CEMS**

Deux paramètres principaux sont pris en compte :

- la capacité du CEMS d'établir une prévision de ses consommations futures (horizon H + 2 à J + 2), potentiellement par usage,
- le choix du gestionnaire du bâtiment de s'engager sur un niveau de flexibilité.

#### **Trois niveaux de CEMS considérés**

Niveau 1 : CEMS sans capacité de prévision – pas d'engagement contractuel de flexibilité

Niveau 2 : CEMS sans capacité de prévision, avec engagement contractuel de flexibilité

Niveau 3 : CEMS avec capacité de prévision avec engagement contractuel de flexibilité – hors spec V1.

Les API sont cumulatives (niveau 2 : le niveau 1 plus le spécifique du niveau 2).  
Les API sont sécurisées, selon les exigences de cyber sécurité définies dans un document séparé.

La norme IEC 62746-4 définit dans l'annexe A la séquence d'échanges entre les acteurs pour le Use Case : Incentive-based building energy management. L'API Flex Ready® se base sur cette séquence (jointe en fin de document) pour les échanges qu'elle traite.

### 3. Data model

Le Data model utilisé est basé sur la norme IEC 62746-4.

D'une façon générale, le CEMS dispose d'un identifiant mRID (CEMSID), créé lors de l'enregistrement Flex Ready®. Cet identifiant est unique (selon le standard UUID) conformément à la IEC norme 62746-4 §5.1 Master resource identifiers.

Le CEMS contrôle des 'Assets', qui peuvent correspondre à un périmètre fonctionnel (Chauffage, IRVE...) ou à des zones du bâtiment, voire le bâtiment entier. Un nom d'Asset « WholeBuilding » est réservé pour le cas où le seul Asset pris en compte est le bâtiment entier. Dans ce cas, aucun autre AssetID n'est défini.

Les identifiants **AssetID** des Assets sont définis par le gestionnaire du Bâtiment. Ils sont communiqués au Service Provider dans le contrat. Il est recommandé d'utiliser un UUID pour les AssetID.

**Note** : Si l'Asset est un générateur, cet AssetID est un UUID, enregistré par le DSO. Différents Assets peuvent être gérés par différents Service Providers : l'API est ouverte à cette possibilité.

Une heure est exprimée en heure GMT. Les dates sont au format ISO 8601 :  
Ainsi, pour les jours de la semaine : 1 : lundi, 2 : mardi, ... 7 : dimanche.

**FloatQuantity** est une valeur décimale accompagnée d'une unité (ex : W, kW).

Une PowerTimeSeries de puissance est une séquence [puissance : **FloatQuantity**, date : Date, début : Heure, fin : Heure] associée à une valeur décimale de résolution en secondes. Dans la première version de l'API, cette résolution est fixée à 15' (900 s).

Une requête de flexibilité correspond à un FlexProductType. Dans cette version de l'API, 2 types sont utilisés, selon l'horizon de temps de la requête :

- Hour ahead market (HAM)
- Day ahead market (DAM)

*Certaines API retournent un code :*

- 200 OK : Succès
- 202 Accepted : Demande acceptée avec succès
- 400 Bad Request : Erreur de format dans la requête
- 422 Unprocessable Entity : Erreur de validation métier

*A ce stade, les API n'autorisent pas de données supplémentaires ; une analyse de l'impact cyber sécurité sera faite pour définir dans quelles conditions des variables supplémentaires pourraient être introduites par l'utilisateur.*

### 4. API Niveau 1

Au niveau 1, le CEMS n'a pas la capacité d'estimer sa consommation à venir. Le bâtiment ne peut donc s'engager au résultat contractuellement avec le Service Provider, mais simplement à faire ses best efforts.

- Potentiel des assets du bâtiment : le Service Provider interroge le serveur du bâtiment ; l'API fournit une information statique au Service Provider, sur le potentiel théorique des assets qui peuvent entrer dans la flexibilité (qui peut être évaluée selon les méthodes GOFlex).

**func getCEMSAAssets(cems: CEMSID) → [MEPID: String, [(asset: AssetID, flexProduct: FlexProductType, potentiel: [PotentielFlex])]]**

MEPID : Market Evaluation Product ID est le numéro de PdL.

AssetID est un String UUID.

Le type d'asset (une première liste est définie dans la norme IEC 62746-4 Table 25) sera introduit dans une future version de la spec.

Si l'asset est le bâtiment complet, UUID est celui du CEMS.

Protocole : Timeout : 60s. En cas de non-réponse, un message d'erreur est généré et enregistré. Si le CEMSID est erroné, le CEMS retourne une liste vide.

L'API fournit un tableau donnant pour chaque asset un potentiel de flex, en indiquant pour quel type de service de flexibilité.

Potentiel est un tableau de PotentielFlex (plusieurs éléments si on distingue en fonction de la saisonnalité par exemple). Dans la version 1, une seule valeur de potentiel, à l'instant de la requête. Le niveau de confiance n'est pas exploité en V1.

PotentielFlex est une structure de données contenant : la puissance, le délai de mobilisation (par défaut : 1 h), la durée max, le nombre d'activations dans la journée (par défaut 1, seule valeur acceptée en V1), les périodes horaires, optionnellement un niveau de confiance sur la mise en œuvre (exclu en V1). Le potentiel est celui valable à date.

PotentielFlex :

```

yearPeriod      : startDay, endDay
activationPeriods: (months : [1..12], weekDay : [1..7], hours: [Hour]) inside the yearPeriod
notification    : Int    Une durée exprimée en minutes
maxDuration     : Int    Une durée exprimée en minutes, depuis le début d'une période d'activation
maxActivationsPerDay: Int
confidence level: 1..5      1 minimum – 5 confiance totale - optionnel

```

**Par exemple :**

yearPeriod : ( 2025-03-25, 2025-06-30)

ActivationPeriods : (months : [3, 4, 6], weekDay : [6,7], hours: [10, 17]) : la flex peut être activée (sur cette période de l'année, tous les mois sauf Mai, les samedis et dimanche, à 10h GMT et 17h GMT, pour une durée maxDuration

Les différentes yearPeriod doivent être cohérentes. En cas d'incohérence (par ex : un instant donné apparaissant dans 2 PotentielFlex), le résultat n'est pas garanti (en général, le premier qui correspond à la période sera pris en compte).

Un assetID est prédéfini pour le bâtiment complet. Si cet asset existe, la liste des assets en réponse ne doit contenir que seul asset.

- Envoi de requête de flexibilité : Le Service Provider envoie une demande au CEMS (et génère un identifiant pour la requête) pour une période donnée, par asset (l'Asset peut être le bâtiment complet) ; mais sans avoir de garantie de respect. Pas de réponse attendue excepté un accusé de réception (ou notification d'erreur dans les données) qui permet au Service Provider de relancer (1 fois en V1) sa requête en cas de non-réponse ou d'erreur.

```
func askFlex(requestID: UUID, cems: CEMSID, asset: AssetID, flexProduct : FlexProductType, puissance: PowerTimeSeries) → Ack.
```

*Ack indique la bonne réception et le statut reçu / Refus (Yes / No) le CEMS ne s'engage pas (car niveau 1) : **recuSansEngagement**.*

requestID est un UUID, à conserver par le Service Provider et le CEMS, en tant qu'élément de preuve éventuel.

FlexProductType est l'un des types définis dans la norme 62746-4.

Les types utilisés dans cette version de l'API : RPD (Ramp Down : flexibilité à la baisse), optionnellement RPU (Ramp Up : flexibilité à la hausse), si l'asset est générateur.

Protocole : Timeout : 5s. Le CEMS n'a pas d'évaluation à faire, il indique seulement que la demande a été reçue, ou refusée, afin de permettre au Service Provider d'intégrer cette requête dans ses calculs de potentiel de flex.

Le CEMS a pu vérifier que la demande est cohérente avec son potentiel de flex. L'absence de réponse vaut refus.

- Activation de la requête. Le Service Provider informe le CEMS que la requête est retenue dans son plan d'activation. L'activation est envoyée en tenant compte du délai de mobilisation. Le CEMS n'a pas d'engagement d'exécution (niveau1). Réponse attendue : YES/NO. Le requestID (UUID généré par l'émetteur) permet de garder une référence de la requête.

```
func activateFlexRequest(requestID: UUID, cems: CEMSID, asset: AssetID, flexProduct: FlexProductType, puissance: PowerTimeSeries) → Ack
```

Protocole : Timeout : 5s. Le CEMS n'a pas d'évaluation à faire, il indique seulement que la demande est acceptée ou refusée (YES/NO) et si acceptée, qu'il fera ses best efforts pour la satisfaire (l'incentive étant la rémunération prévue dans le contrat). En cas de non-réponse, le Service Provider exclue cette requête de son plan d'activation.

- Annulation requête : absent en V1

```
func cancelFlexRequest(requestID: UUID) → Ack
```

**Note** : le Service Provider peut annuler la requête en renvoyant, en respectant le délai de mobilisation, la requête (avec le même requestID) avec un **puissance = 0**.

- Reporting du réalisé : Après la période d'activation, le Service Provider demande au CEMS la courbe de puissance appelée (à un pas de temps indiqué dans la **PowerTimeSeries – fixe** de 15' = 900s dans cette version de l'API-) sur la période de la requête de flexibilité, pour l'Asset considéré. Le CEMS répond par un tableau donnant, au pas d'échantillonnage retenu (15'), la puissance appelée par Asset (note : CEMS : CEMSID, asset : AssetID rappelés par sécurité, mais normalement redondant avec requestID. La définition du calcul du réalisé est à préciser dans le contrat. En V1 : valeur moyenne de la puissance sur la période.

```
func realiseFlexRequest (requestID: UUID, cems: CEMSID, asset: AssetID) → [(requestID: UUID, reported : PowerTimeSeries)]
```

Protocole : Timeout : 15' (900 s). Le CEMS doit collecter des données (éventuellement lecture du compteur), ce qui justifie le timeout important. En cas de non-réponse, le Service Provider répète au maximum 1 fois sa requête puis génère un message d'erreur pour log. Si le Service Provider détecte une puissance > potentiel de flex de l'asset : générer erreur dans le log.

## 5. API Niveau 2

Au niveau 2, le CEMS n'a pas la capacité d'estimer sa consommation ni donc sa flexibilité à venir. Le bâtiment peut déléguer cette prévision au Service Provider pour s'engager contractuellement avec lui.

Au niveau 2, une proposition de flexibilité peut être accompagnée d'une proposition de prix. Le CEMS acceptera la proposition ou la rejettera en tenant éventuellement compte de cette proposition.

- Potentiel des assets du bâtiment : idem niveau 1.
- Consommation instantanée par asset (moins d'une heure d'ancienneté) des assets qui peuvent entrer dans la flexibilité, à l'heure de la requête, **pour chaque PdL (MEPID) concerné.**

```
func getCEMSAssetsConsos(cems: CEMSID) → [pdl: MEPID, [(asset: AssetID, date: Date, heure: Heure, conso: FloatQuantity)]]
```

Protocole : Timeout : 15' (900 s). Le CEMS doit collecter des données (éventuellement interrogation des assets), ce qui justifie le timeout important. En cas de non-réponse, le Service Provider répète au maximum 1 fois sa requête puis génère un message d'erreur pour log.

- Consommation historique des assets.

Le CEMS a archivé ses consommations sur une durée d'historique à définir (dans le contrat, minimum 1 mois), en tant que valeur moyenne (ou pMax ultérieurement) sur un pas de temps (1 heure) défini dans le contrat. Le Service Provider peut récupérer cet historique (par asset), ce qui lui donne une information très importante pour construire sa prévision :

```
func getCEMSAssetsHisto(cems: CEMSID, asset: AssetID) → [(date: Date, heure: Heure, conso: FloatQuantity)]
```

Protocole : Timeout : 60s. En cas de non-réponse, un message d'erreur est généré et loggé. Si le CEMSID ou l'AssetID est erroné, le CEMS retourne une liste vide.

Un historique de 1 mois correspond à 720 valeurs par asset.

- Le Service Provider envoie une proposition au CEMS avec offre de prix, qui l'accepte ou la refuse ou demande de renégocier dans sa réponse.

```
func askFlexWithPrice(requestID: UUID, cems: CEMSID, asset: AssetID, flexProduct: FlexProductType, puissance: PowerTimeSeries, askingPrice: PriceTimeSeries) → Reponse
```

*La réponse est explicite (acceptée, refusé, à modifier).*

**PriceTimeSeries** : TimeSeries, avec valeur en €. En V1, 1 seul point. Son time window est le même que pour la puissance.

**Note** : Intègre les paramètres time window, amount, price (BidPriceSchedule) de la norme IEC 62746-4.

Protocole : Timeout : 15'. Le CEMS doit évaluer sa capacité à exécuter la demande de flex, ce qui justifie la durée du timeout. Si le CEMS ne sait pas évaluer, il doit refuser la demande pour cet asset. Si un identifiant est incorrect, le CEMS répond par un code erreur (identifiant inconnu).

En cas de non-réponse, la requête peut être répétée 1 fois.

Si le CEMS refuse, la requête n'est pas réémise.

Dans la version 1 du protocole, la modification ne peut porter que sur la puissance. Si le prix ne convient pas, réponse de refus.

**Note:** 62746-4 définit ProductBid as « a container which has two associations: the top being the MarketProduct (Energy, Reserve, Regulation, etc.) and the bottom association being the BidPriceSchedule ».

The BidPriceSchedule class captures data on three axes signifying:

- 1) The time window, e.g. from 6:00 to 7:00
- 2) The amount of service, e.g. 1 MWh
- 3) The asking price, e.g. 20 €/MWh

Dans le cas de l'API Flex, le MarketProduct est une demande de flexibilité (Regulation up or down : RPU, RPD).

Cette data askingPrice est **optionnelle**, dépendant de la nature du contrat (la valeur peut être définie par défaut). Le CEMS doit savoir, s'il doit ou non, recevoir cette information.

Note : extrait de la norme (Table A.3)

**Table A.3 – Information exchanged in price notification and energy consumption plan notification**

Name of information	Description of information exchanged	Requirements to information data
Energy Price for Building Group	Time series of price for energy for Building Group	Time series of price for energy consumption, (price/Wh) per unit time period (e.g. 30 min, 15 min, etc.)
Energy Price for Building	Time series of price for energy for Building	Time series of price for energy consumption, (price/Wh) per unit time period (e.g. 30 min, 15 min, etc.)
Intended Energy consumption	Time series of energy consumption and in the future (e.g. day ahead, week ahead, etc.) which is set by Building Group Manager as the control target under the given incentive / price.	Time series of Energy (Wh) per unit time period (e.g. 30 min, 15 min, etc.)
Energy consumption plan for Building	Time series of energy in the future (e.g. day ahead, week ahead, etc.) for the building under the given incentive / price.	Time series of Energy (Wh), Power (W) and emission quantity of CO2 (gram) per unit time period (e.g. 30 min, 15 min, etc.)
Energy consumption plan for Building Group	Time series of energy consumption in the future (e.g. day ahead, week ahead, etc.) for the Building Group, which is made from Energy consumption plans of all Buildings within the Building Group.	Time series of Energy (Wh) per unit time period (e.g. 30 min, 15 min, etc.)

- Le Service Provider envoie la confirmation de la requête (selon délai de mobilisation), identifiée par son requestID, au CEMS si celui-ci a accepté la demande initiale, pour activation.

**func activateFlexRequest(requestID: UUID, cems: CEMSID, asset: AssetID, flexProduct : FlexProductType, puissance: PowerTimeSeries, askingPrice: PriceTimeSeries) → Ack**

Protocole : Timeout : 5s. Pour accuser réception.

Réponse attendue : YES/NO. Pas de détail sur la raison du refus (la norme est en cours de définition sur ce point).

L'absence d'envoi de **activateFlexRequest** dans le délai d'activation (pour askFlexWithPrice) entraîne l'annulation de la requête initiale.

- Annulation de la confirmation de requête : absent en V1

**func cancelConfirmation(requestID: String) → Ack**

## 6. API Niveau 3

Au niveau 3, le CEMS aura la capacité d'estimer sa consommation et donc sa flexibilité à venir pour faire des propositions au Service Provider et s'engage contractuellement avec le Service Provider.

Ce niveau 3 sera défini dans une phase ultérieure.

## 7. Validation des API

Pour obtenir le label Flex Ready, avec niveau, le fournisseur du CEMS devra avoir vérifié qu'il respecte les interfaces définies.

Dans un premier temps, ceci sera fait par auto-déclaration, après exécution des scénarios de test sur l'émulateur.

Pour cela, Flex Ready met à disposition des acteurs :

- un émulateur de serveur de Service Provider (opérateur de flexibilité),
- un émulateur de CEMS avec les différents niveaux,
- la définition de la séquence de test à effectuer (incluant des cas de données erronées).

L'émulateur fournit un rapport de test en fin de séquence, à joindre à l'auto-déclaration.

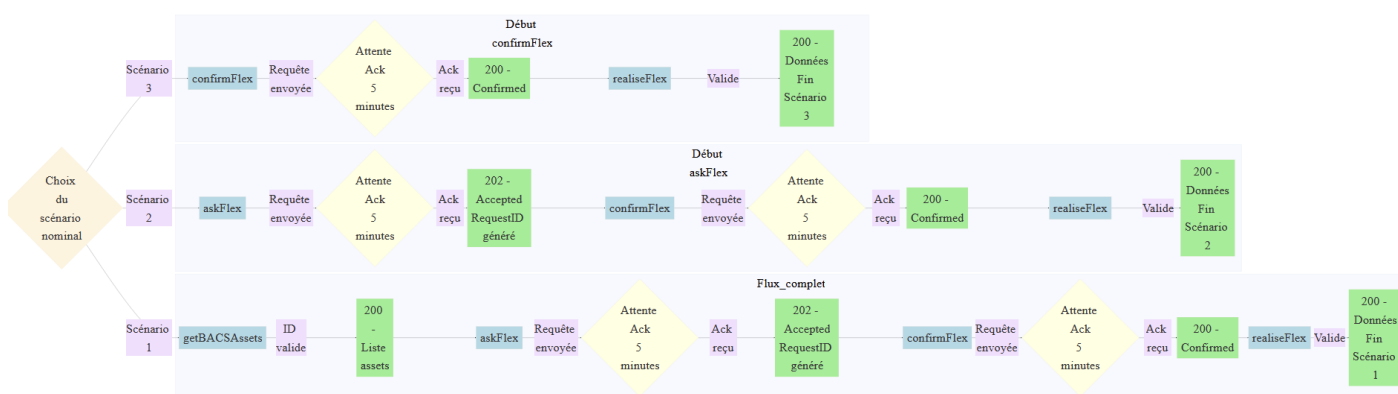
## 8. Scenarios de test

L'échange des Identifiants (eg., CEMSID) est prédéfini au moment du contrat et ne fait pas partie de l'API. Ils sont entrés « en dur » dans l'émulateur.

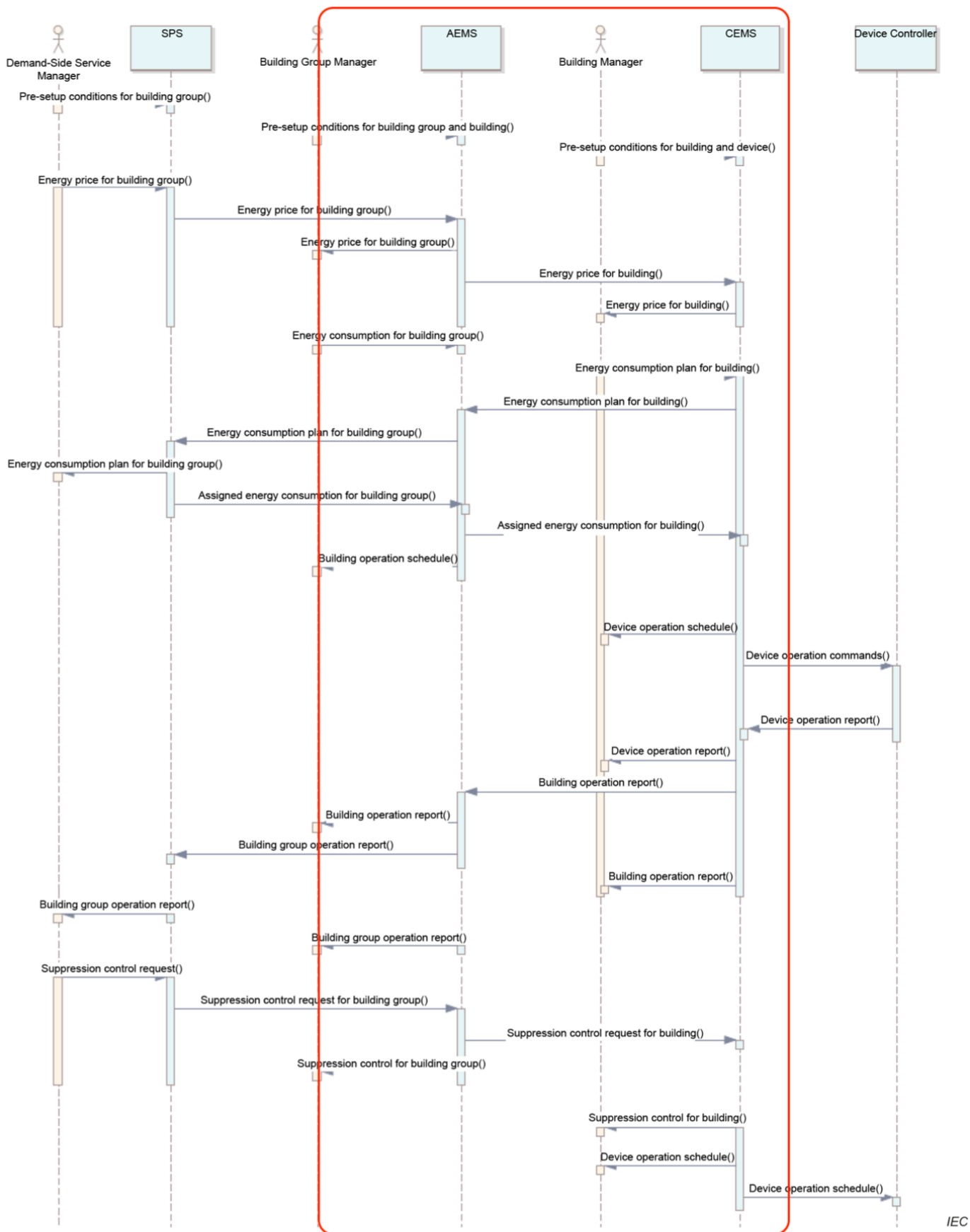
Pour chaque API :

- Echange d'authentification, selon le protocole retenu (Oauth 2., à confirmer).
- Si l'identifiant (e.g., CEMSID) n'est pas reconnu, le CEMS ignore la requête et répond « Erreur ». Après une répétition (5) d'identifiant non valide, le CEMS alerte d'une possible tentative d'intrusion ou de déni de service.
- Envoi d'une requête valide
- Réponse : une fois en acceptation, une fois en refus
  - Test de requêtes non valides (sur les dates, sur les AssetID, sur le volume...).

Les scenarios de test se présentent selon les diagrammes ci-dessous :



**Annexe 1 - Process général d'échanges (issu IEC 62746-4). Le champ de l'API Flex Ready est cadré en rouge.**



**Figure A.2 – The whole view of this use case**

## Annexe 2. Définition des acronymes

<b>Acronym</b>	<b>Phrase</b>
<b>AEMS</b>	Aggregator Energy Management System
<b>CEMS</b>	Customer Energy Management System
<b>DAM</b>	Day-Ahead Market
<b>DER</b>	Distributed Energy Resource
<b>EPS</b>	Electrical Power System
<b>PV</b>	Photovoltaic
<b>RTM</b>	Real-Time Market
<b>SPS</b>	Service Procurement System
<b>UML</b>	Unified Modeling Language
<b>VPP</b>	Virtual Power Plant
<b>XML</b>	Extensible Markup Language
<b>XSD</b>	XML Schema Definition