



LIVRE BLANC

Comprendre le **Building Operating System**

Table des matières

I.	Le Building Operating System	2
A.	Pourquoi utiliser un BOS ?	2
B.	Principe et rôle du BOS	2
C.	La fonction de Middleware	3
D.	Gestion des données	3
E.	Différents types de BOS	5
F.	En résumé	6
II.	Notions fondamentales	8
A.	Généralités	8
1.	Types d'architectures	8
2.	Architecture distribuée	9
3.	Architecture centralisée	10
4.	Couche logicielle unique	10
5.	Réseau fédérateur IP	11
6.	Services tiers	11
B.	Intégration des données	12
1.	Capacités de communication : Protocoles terrain et IoT	12
2.	Qualification des données	12
3.	Types de données	12
4.	Hierarchiser les ordres de commande	13
C.	Structuration des données	13
1.	Modélisation des données	13
2.	Types de modèles de données	15
3.	Capacité d'import des modèles de données (gestion du BIM)	15
4.	Gouvernance de la donnée	16
5.	Traitement local des données	17
D.	Mise à disposition des données en lecture et écriture	17
1.	Connecteurs	18
2.	API Rest native	18
3.	Types de données transmises	18
4.	Sécurité	19
E.	Services locaux	19
1.	Supervision technique	19
2.	Gestion énergétique	20

I. Le Building Operating System

A. Pourquoi utiliser un BOS ?

Depuis quelques années, le secteur du bâtiment évolue sous l'impulsion de la transformation numérique. Cette digitalisation a un impact majeur sur la manière dont les bâtiments sont utilisés.

D'un côté, les usagers recherchent davantage d'interactions avec leur environnement quotidien, poussés par une nouvelle génération adepte de services instantanés : gestion applicative du confort des bureaux, réservation de salles, signalement de problèmes, géolocalisation, chatbot, etc. De l'autre, les bâtiments, bien qu'équipés de nombreux systèmes et capteurs produisant des données, peinent à satisfaire ces nouveaux besoins.

De même, la digitalisation des bâtiments change les méthodes d'exploitation et ouvre de nouvelles voies de création de valeur. Les données du bâtiment sont sous-exploitées car souvent dissociées et peu cohérentes. Or ces données peuvent, en étant mieux exploitées, générer beaucoup de valeur ajoutée : détection de dysfonctionnements, optimisation énergétique, optimisation de l'espace, registre d'entretien, maintenance prédictive, etc.

Les bâtiments qui ne sont pas capables de suivre cette évolution verront leur valeur se déprécier face à des bâtiments dotés de systèmes permettant le déploiement de services de plus en plus larges. La maîtrise des Data d'un bâtiment est devenue primordiale pour éviter son obsolescence. Le BOS assure le trait d'union nécessaire entre un bâtiment et ses données d'un côté, et les services de l'autre.

B. Principe et rôle du BOS

L'objectif du Building Operating System (BOS) est de rationaliser et mutualiser les flux d'informations entre les équipements/capteurs du terrain d'une part et les applications servicielles d'autre part. Les données opérationnelles sont ainsi gérées au sein du BOS qui alimente en données l'ensemble des applications.

Avec son approche Micro-services*, le BOS permet à tout fournisseur de Services de déployer des applications afin de répondre aux besoins des clients (Propriétaires, Usagers, Exploitants, etc.) sans avoir à modifier l'infrastructure de gestion des données locales. Son architecture orientée API / connecteurs Web Services et sa capacité à gérer la complexité du terrain permettent de dissocier les services de l'environnement local.

Les services peuvent ainsi être créés, étendus et déployés indépendamment les uns des autres et consommer les données relatives au bâtiment suivant les autorisations attribuées. Le BOS constitue donc de facto une architecture R2S au sens de la Smart Building Alliance (SBA).

**(L'architecture microservice est une méthode de développement d'applications logicielles en tant que suite de services modulables et indépendants, dans lesquels chaque service exécute un processus unique et communique à travers un mécanisme léger et bien défini pour atteindre un objectif commercial.)*

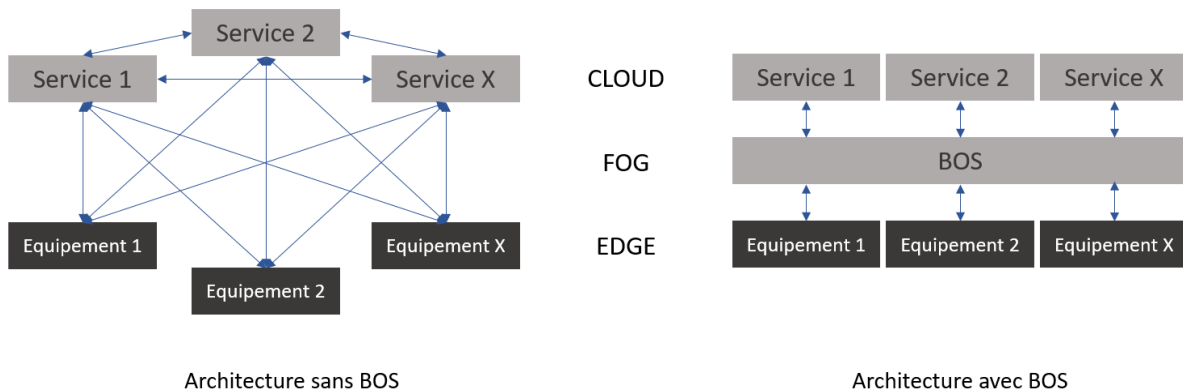
C. La fonction de Middleware

Le BOS assure la fonctionnalité de middleware (de type ESB – Enterprise Service Bus). C’est-à-dire qu’il gère le processus d’intégration des données qui consiste à combiner des données provenant de différentes sources dans une interface unifiée pour rendre les données directement exploitables aux Services qui en ont besoin. Cette intégration des données est réalisée en 3 phases :

1. Acquisition
2. Consolidation
3. Traitement

Le BOS permet donc le regroupement de données hétérogènes en un gisement unifié, facilement accessible par des applications tierces (Services). Il peut acquérir des données localement (Équipements communicants, Gateway, etc.) ou adresser des serveurs de données distants (Météo, IoT, etc.).

Plutôt que de faire office de passerelle entre des systèmes disjoints, le BOS permet de s’adresser directement aux réseaux de ces systèmes. Il doit disposer d’une connectivité directe au plus près des sources, sans empilement de passerelles, ni couches logicielles à programmer. Ainsi, les échanges de données et évolutions futures seront plus simples et moins coûteux à réaliser.



D. Gestion des données

Le BOS est une plateforme de gestion des données qui doit fournir plusieurs outils pour administrer les données acquises selon trois étapes principales :

- Intégration des données comme décrit au paragraphe précédent
- Modélisation et structuration des données
- Mise à disposition et gestion des interactions

A noter que la notion de “donnée” ne se limite pas à des valeurs temps réel lues, mais recouvre également des formats plus élaborés (commandes, historiques, événements, programmes horaires, etc.) qui sont également traités par le BOS. Les données acquises par le BOS au niveau du bâtiment ne se limitent pas forcément à des données locales mais peuvent provenir de systèmes tiers si besoin.

Exemple : une prévision météo ou une facture de consommation relative au bâtiment.

Etape 1 : Intégration de données (acquisition et traitement)

En plus de l'acquisition de données hétérogènes provenant de sources variées, cette première étape consiste aussi à nettoyer et convertir les données. Ces transformations permettent de s'assurer d'un format homogène (conversion d'unité, de temps, etc.) et d'une qualité de donnée optimale (gestion des valeurs incohérentes, gestion d'absence de valeurs, etc.).

Elle permet aussi de convertir les formats de données suivant les besoins. L'accès aux données brutes reste possible.

Exemple : une donnée acquise doit pouvoir être enregistrée sur le temps et déclencher des alertes selon un dépassement de seuil ou une incohérence. Ainsi, pour une donnée acquise en temps réel, deux nouvelles données de types différents sont générées : une nouvelle entrée dans un historique et une alerte en cas de dépassement ou d'incohérence.

Etape 2 : Modélisation et structuration

Pour permettre aux applications de services d'accéder aux données dont elles ont besoin, il est nécessaire de rendre cet accès le plus simple possible en faisant abstraction de la complexité des systèmes.

Pour cela, les données sont associées à plusieurs modèles descriptifs : géographique, métier, nature, etc. qui ont du sens pour l'usage de ces données. Les interactions entre les différents éléments (équipements, capteurs, actionneurs...) sont aussi décrites au travers de relations mutuelles pour décrire l'existant et les hiérarchies entre ces éléments.

Pour cela, le BOS utilise une base de données orientée graphe qui facilite la description de systèmes complexes et la recherche d'informations. Les données sont accessibles via des nœuds individuels qui représentent chacun une partie du système.

Exemple : un nœud peut être un preneur, un équipement, un étage, etc.

Chaque nœud peut posséder des relations avec d'autres nœuds afin de définir leurs interactions. L'ensemble de ces nœuds forme une sémantique (un ensemble de termes utilisés).

Exemple : des relations sont créées entre les nœuds "Groupes Froids" et "Ventilo-convecteurs" pour matérialiser quels ventilo-convecteurs sont alimentés par quel groupe froid. De même, d'autres relations peuvent être utilisées pour définir les zones dont le confort est assuré par tel ou tel ventilo-convecteur ou quel preneur est rattaché à quelle(s) zone(s).

Les noms des relations, des preneurs et des zones font partie de la sémantique. Cette description en graphe permet ainsi de créer facilement des requêtes du type "quelles zones sont alimentées par le groupe froid n°1".

Etape 3 : Mise à disposition et interactions

Les données doivent pouvoir être fournies à des applications indépendantes en utilisant des modes de communication IT selon plusieurs moyens et formats : protocoles Web Services, API Rest, Connecteurs, etc. C'est le principe même de l'architecture R2S : Ready2Services.

A noter que le BOS doit fournir un environnement de développement ouvert pour créer de nouveaux connecteurs ou enrichir l'API simplement. Cela permet de répondre à de nouveaux besoins car les échanges nécessaires peuvent ne pas être connus à la livraison du bâtiment ou évoluer au fur et à mesure de l'intégration de nouveaux services dans la vie du bâtiment.

En plus de sa capacité d'échanges de données, le BOS doit disposer d'outils de Gestion d'autorisations afin de contrôler l'accès aux données pour chaque application.

Exemple : l'administrateur du BOS pourra autoriser le prestataire de services énergétiques à récupérer les données relatives au comptage et aux conditions environnementales, mais lui interdire l'accès à d'autres données (noms des preneurs, Informations sur la maintenance, etc.). De même, l'administrateur du BOS pourra autoriser l'accès aux données de présence dans les zones à une application de réservation de salles et lui interdire l'accès à d'autres données.

E. Différents types de BOS

Le BOS est un concept informatique large et ses fonctions précises peuvent différer selon le point de vue de l'éditeur de la solution.

Local versus Cloud

Les solutions existantes proposent différents types d'architectures :

- Entièrement basée sur le cloud
- Entièrement locale
- Hybrides (cloud + passerelle locale).

Pour assurer une pérennité du BOS face à l'évolution technologique et l'obsolescence rapide des systèmes Cloud, le BOS choisi doit faire partie intégrante du bâtiment et être installé localement. Il est considéré comme le cœur du système d'information du bâtiment et à ce titre, le BOS est un véritable asset du bâtiment.

Une base de données cloud peut lui être associée uniquement si celle-ci n'entraîne pas de dépendance et que le BOS peut continuer à opérer sans cette base.

A noter que cette distinction "local VS cloud" peut aussi s'entendre comme une distinction "faisant partie de l'infrastructure VS externe".

Exemple : dans le cas d'une gestion de nombreux petits sites, le BOS pourrait être intégré à un serveur unique faisant partie de l'infrastructure informatique du client.

Mais les gains en coûts de cette centralisation se feront au détriment de l'évolutivité du BOS (cette disposition permet de gérer les spécificités par bâtiments sans avoir à traiter tous les cas à un niveau centralisé) et de la valorisation des actifs en cas de revente (si le BOS ne fait pas partie du bâtiment même, on perd l'avantage de la valeur ajoutée associée).

Pour résumer cette notion, il est primordial que les éventuelles fonctions du BOS logées dans le cloud puissent être installées sur un environnement maîtrisé par le client.

Plateforme Applicative versus Micro-Services

A l'instar des plateformes et OS proposés par les GAFA sur lesquels des applications sont développées, il doit être possible d'enrichir le BOS au travers d'un SDK (Software Development Kit). Cela permet d'enrichir des visualisations, d'apporter de nouveaux modèles de données ou de créer de nouveaux connecteurs. En devenant une plateforme applicative, le BOS permet au client final de ne pas être contraint dans un système fermé qui ne pourra pas évoluer. Il peut enrichir le BOS tout en étant indépendant de l'éditeur du BOS.

Les applications de service pourraient être développées au sein du BOS (comme sur un OS de téléphone). Cependant, il faut garder en tête les effets contraignants :

- Cela impose aux prestataires de Services de créer une application pour chaque BOS entraînant des coûts conséquents alors que bien souvent ces prestataires disposent déjà de solutions et applications métiers existantes et souvent plus larges.
- Cela rend, de fait, les prestataires de services directement dépendants du BOS et de ses évolutions au bon vouloir de l'éditeur. Dans le cas d'un OS pour la téléphonie, cela n'est pas "stratégique" car on peut facilement changer de téléphone. En revanche, dans le cadre du cycle de vie d'un bâtiment, c'est une contrainte bien plus lourde.

Il est plus judicieux d'adopter une architecture orientée micro-services permettant une plus grande flexibilité et réduisant la dépendance des prestataires de Services au BOS.

Sur ce principe, le BOS fournit les données du bâtiment aux applications extérieures au travers d'API ou de connecteurs. Cela nécessite en revanche pour le BOS de disposer d'une grande richesse de connecteurs IT "sur étagère" afin de pouvoir connecter facilement et de manière industrielle de nouveaux services ou plateformes sans développement spécifique.

Dans tous les cas, le BOS doit être un système sur lequel le développement est ouvert et doit fournir la possibilité d'être enrichi au travers de son SDK avec un langage de développement classique (JAVA, Javascript, C#, .NET, etc.).

Services locaux versus Services externes

Il n'est pas interdit que le BOS assure certains services localement. Toutefois, le BOS doit garantir le principe de portabilité de ces services, c'est-à-dire qu'il doit garantir l'accès des données à une application externe fournissant un service équivalent sans restriction.

Exemple : le BOS peut inclure des rapports énergétiques mais il devra nécessairement garantir la possibilité de transmettre les données associées à une autre application de tableaux de bord.

F. En résumé

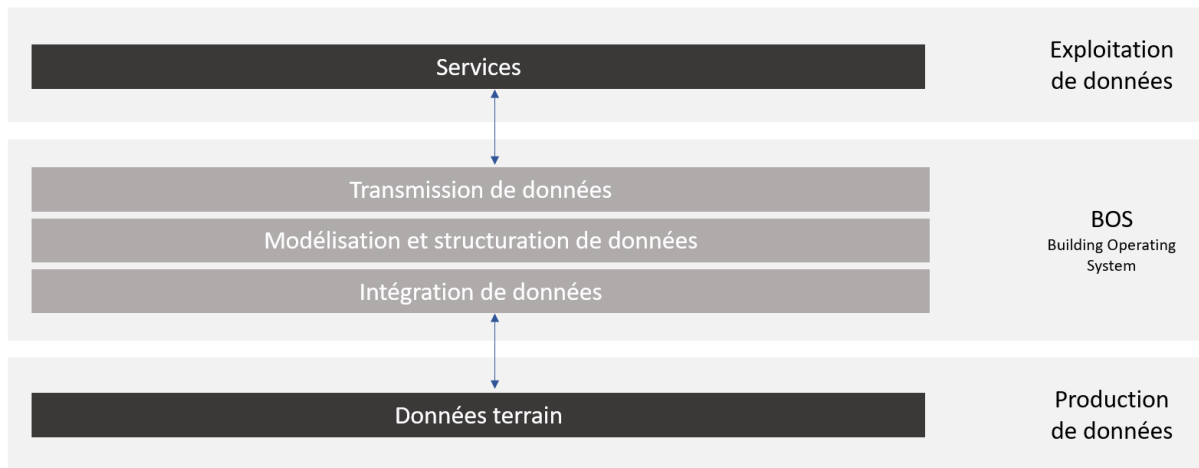
Le BOS permet de gérer les données de tous types au niveau bâtiment avec 3 fonctions principales :

- Intégration des données (protocoles terrain & IoT et traitements associés)
- Modélisation de ces données suivant un modèle sémantique afin d'en comprendre simplement le sens
- Mise à disposition de ces données pour des services tiers suivant les bonnes pratiques de sécurité informatique et en utilisant des technologies IT

En complément, le BOS peut, si nécessaire, héberger des services localement pour les besoins du bâtiment.

Exemple : IHM ou tableaux de bords

Le BOS fait partie intégrante de l'actif immobilier. C'est le lien essentiel entre le monde physique et le monde digital. Le BOS est indépendant, il s'occupe des échanges de données, il gère les modèles et les référentiels de données. Il sert des applicatifs intégrés à l'Asset telle la supervision technique ou des applicatifs souscrits en mode service (BIM, gestion de confort, gestion de maintenance et exploitation, etc.).



Positionnement du BOS

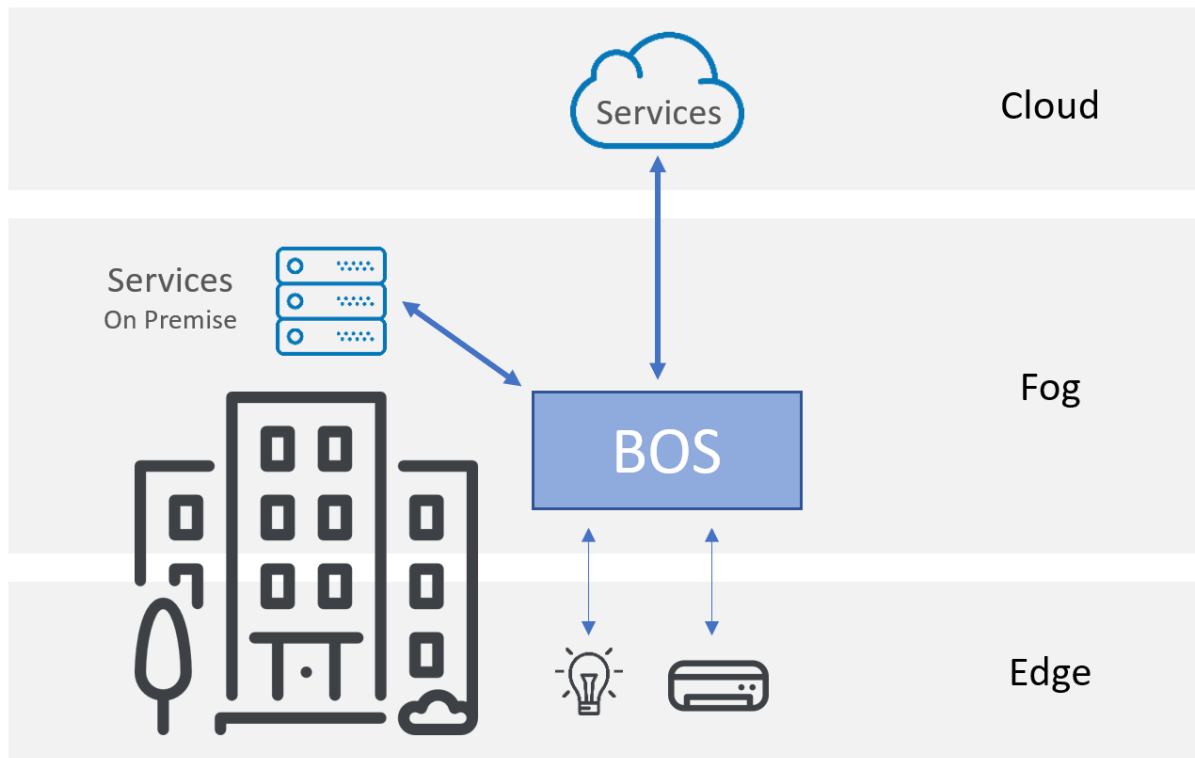
Enfin, afin de conserver les capacités d'ouverture et d'évolutivité du système sans remettre en cause l'architecture générale, le BOS doit proposer une approche de type Micro-services et être extensible : le développement de nouvelles fonctions (IHM, driver terrain, connecteur à des services tiers, etc.) doit être libre (càd, sans avoir l'obligation de faire intervenir l'éditeur du BOS).

II. Notions fondamentales

A. Généralités

1. Types d'architectures

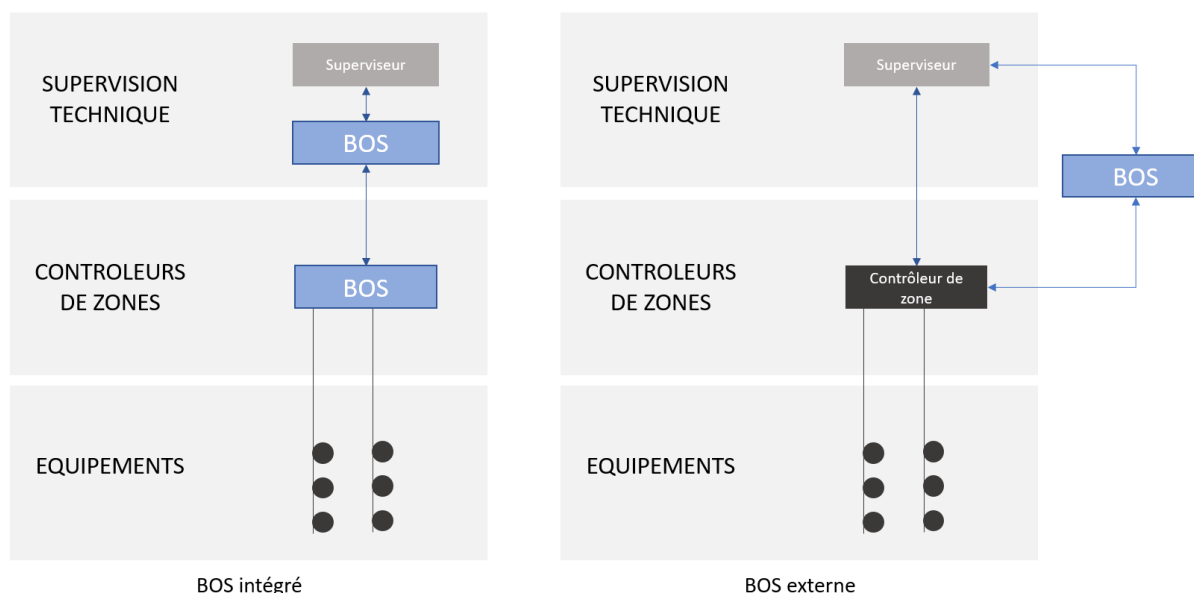
Afin d'assurer un fonctionnement et une sécurité optimale, le BOS doit être mis en place à un niveau intermédiaire entre les Applications de services et les équipements ou capteurs de terrain. Cette couche intermédiaire, appelée FOG, assure la centralisation des données, des fonctions de sécurité et des traitements locaux, sans remettre en cause l'architecture et la capacité d'évolution du bâtiment.



Positionnement du BOS

Le BOS peut être déployé selon deux types d'architectures :

- Un principe d'Intelligence distribuée dans le cadre d'un projet neuf ou en rénovation lourde de son système. Le BOS est fusionné au système de Gestion technique : BOS intégré. Cette architecture permet de réduire considérablement les coûts de déploiement et l'infrastructure matérielle globale.
- Un principe plus centralisé dans le cas où le BOS est greffé à un bâtiment existant disposant déjà d'équipements techniques avec lesquels il communiquera. Le BOS est isolé du système de Gestion technique : BOS externe.



2. Architecture distribuée

Concentrateurs autonomes

Le BOS doit être constitué d'un réseau de concentrateurs autonomes et locaux capables de réaliser toutes les fonctions du BOS sur leurs zones allouées. Ces concentrateurs doivent pouvoir échanger des données les uns avec les autres de façon transparente.

Cette architecture distribuée permet de :

- Limiter les conséquences d'une panne dans le système (pas de SPOF*)

*Single Point Of Failure, point critique du système dont le non-fonctionnement peut entraîner le dysfonctionnement de l'ensemble du système.

- Distribuer la charge de traitement et limiter les flux de données
- Scinder le système très simplement

Exemple : en cas de revente d'un bâtiment à la découpe ou d'occupation par des sociétés différentes, le système pourra être cloisonné entre les occupants pour fournir une indépendance totale sans être contraint à de lourdes modifications

Ces concentrateurs autonomes doivent pouvoir gérer tous types de données (protocoles bâtiment et IoT), les rétrocéder directement à des systèmes tiers, assurer l'archivage temporaire (au minimum un mois) et réaliser des calculs/traitements sur les données « sources ».

Les concentrateurs autonomes peuvent ainsi remplacer les contrôleurs de zones traditionnels des architectures GTB. En effet, les concentrateurs autonomes peuvent assurer les mêmes fonctions que les contrôleurs de zones, mais disposent en plus des fonctions propres au BOS (structuration des données, connecteurs, etc.). En s'évitant la cohabitation d'un système GTB et d'un BOS, les coûts matériels sont largement réduits ainsi que les coûts liés à l'intégration de la GTB dans le BOS.

Concentrateur général

Le but du concentrateur général est de fournir un support centralisé aux concentrateurs autonomes locaux pour le bâtiment ou un ensemble de bâtiments. Il permet d'archiver des quantités de données beaucoup plus importantes (plusieurs années, voire décennies) provenant des concentrateurs autonomes ou de présenter toutes les alarmes survenues en un seul endroit.

Ce concentrateur permet également de réaliser des fonctions de consolidation des données éparses (Synthèses, Ranking, KPI généraux, etc.) et d'administration du BOS (gestion des métadonnées, gestion des utilisateurs et des mots de passe centralisée, sauvegarde automatique des concentrateurs autonomes, etc.).

L'accès à ces fonctions doit se faire sans logiciels particuliers, ni postes clients lourds installés sur l'infrastructure du client. Il est alors nécessaire de proposer une interface multi-utilisateurs utilisant un navigateur Web classique et les dernières technologies Web (technologies Web HTML5 ou ses évolutions) accessibles selon les autorisations données avec des droits d'accès.

Si certains services doivent utiliser une base de données indépendante (comme une GMAO), les connecteurs pour ces bases de données doivent être disponibles au niveau du concentrateur général ou des concentrateurs autonomes. Le concentrateur général doit, de plus, être capable de servir les données des concentrateurs autonomes.

Exemple : pour des raisons de sécurité informatique, l'administrateur du BOS peut décider de ne donner accès qu'au concentrateur général qui peut être utilisé pour atteindre les données des concentrateurs autonomes à la manière d'un proxy.

3. Architecture centralisée

Dans certains cas (bâtiment en rénovation), le BOS sera associé à un système de Gestion Technique existant. Dans cette situation, le BOS est déployé selon une architecture centralisée. Ainsi, les fonctions ci-dessus (concentrateur autonome et concentrateur général) sont assurées par une seule entité, le concentrateur central.

Celui-ci communique directement avec les éléments du système de Gestion Technique existant (via le ou les superviseurs, des concentrateurs de zones ou des équipements) ainsi qu'avec les autres systèmes ou équipements en place. Le concentrateur central se connecte à ces éléments via une liaison IP pour gérer leurs données. De même, il assure les fonctions d'administration et de centralisation qu'assure le concentrateur général dans ce même type d'architecture distribuée.

4. Couche logicielle unique

Afin de simplifier l'évolution du système et d'en maîtriser les coûts, toute installation de passerelle ou empilement de couches logicielles nécessitant être programmées pour échanger des données est proscrit. Le BOS doit assurer la fonction de middleware afin de réaliser l'intégration des données (regroupement de données issues de différentes sources en une interface unifiée).

Dans le cas d'une architecture distribuée, le BOS doit être constitué d'une couche logicielle unique répartie sur différents concentrateurs suivant l'architecture distribuée décrite ci-dessus pour toutes ses fonctions de base.

Il ne doit y avoir qu'un seul intermédiaire entre les services et les équipements terrain : le BOS.

Dans le cas d'une architecture centralisée le BOS est simplement greffé au système existant. Il est nécessaire de définir les interactions entre la GTB et le BOS pour limiter au maximum les opérations à réaliser dans l'un des systèmes après des modifications dans l'autre.

Exemple : la modification du zoning dans la GTB entraîne la mise à jour du modèle géographique dans le BOS. De même, l'ajout de nouveaux compteurs/capteurs dans la GTB implique leur création dans le BOS.

5. Réseau fédérateur IP

L'ensemble des concentrateurs autonomes doit être accessible via un réseau fédérateur IP. Ils sont ainsi en mesure de communiquer leurs données directement ou d'échanger des données les uns avec les autres de manière transparente.

Les échanges de données au niveau fédérateur doivent impérativement être sécurisés et utiliser des protocoles de cryptage de type SSL/TLS. Si des équipements n'utilisent pas de communications sécurisées, leurs communications doivent être complètement cloisonnées.

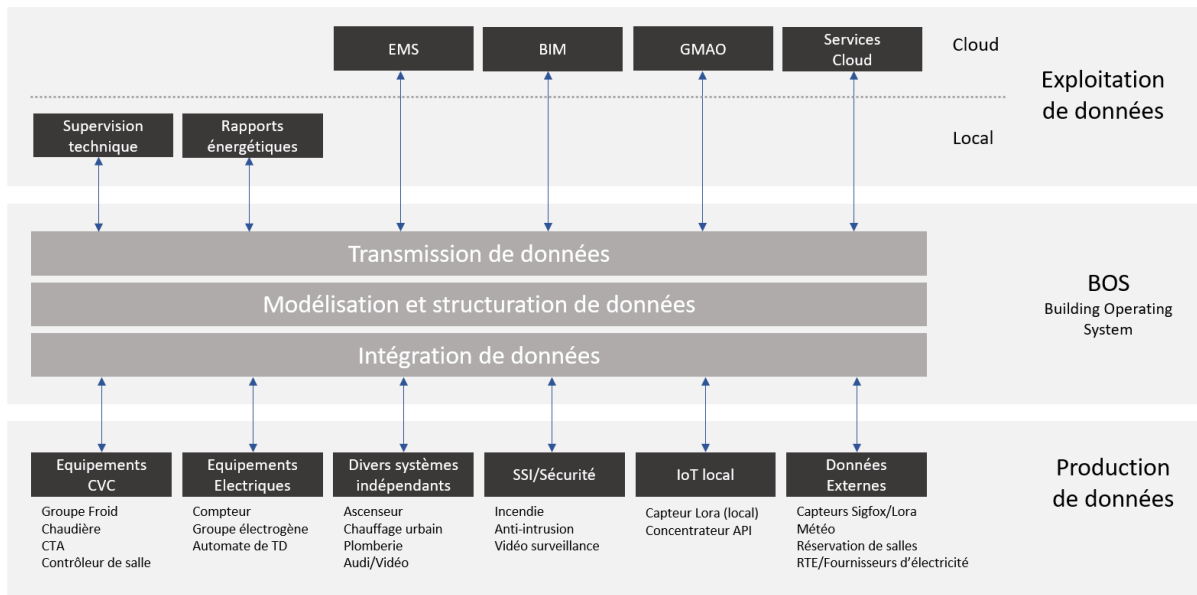
Exemple : un automate Modbus IP ne pourra pas être accessible directement depuis le réseau fédérateur (utilisation de VLAN sécurisés ou de cartes réseau IP isolées du réseau fédérateur).

6. Services tiers

Tout service tiers mis en œuvre est considéré comme indépendant du BOS, de telle sorte que ce service pourra être interchangé avec un autre simplement, sans entraîner de surcoût sur le BOS installé.

Exemple : le bâtiment peut être livré à l'origine sans service particulier. Par la suite, un preneur pourrait décider d'utiliser une application collaborative commune à tout son parc. Le BOS ne doit donc pas préjuger du type de communication nécessaire aux services et fournir une large palette de modes de communication.

L'utilisation d'une base de données externe peut être envisagée, mais le prestataire fournisseur du BOS doit garantir la capacité du système local à renvoyer les données structurées à d'autres bases de données externes standards du marché (SQL, MongoDB, Elasticsearch, etc.) ou à des plateformes standards du marché (Microsoft IoT Hub, Google IoT Core, etc.) à partir du BOS installé localement.



Le parcours optimal de la donnée

B. Intégration des données

1. Capacités de communication : Protocoles terrain et IoT

Les concentrateurs autonomes constituant le BOS doivent être multi-protocoles afin de communiquer avec tous les équipements du marché utilisant un protocole ouvert. Un concentrateur local doit pouvoir communiquer avec plusieurs protocoles simultanément sans forcément avoir à ajouter un nouveau concentrateur.

Ainsi, la solution choisie doit pouvoir s'interfacer à minima avec les protocoles terrains suivants :

- BACnet (Client ou Serveur),
- Modbus (Maître et/ou esclave),
- LonWorks,
- KNX,
- Mbus,
- SNMP
- OPC UA
- MQTT
- Sigfox
- LORA

De même, les concentrateurs autonomes doivent être capables de communiquer avec tous les équipements IoT utilisant une API Restful, JSON ou XML. La liste des connecteurs spécifiques disponibles devra être ajoutée en annexe de la proposition technique et financière.

Ces capacités de communication doivent être nativement disponibles, c'est-à-dire sans surcoût pour une utilisation ultérieure (notamment pour prévoir l'ajout/remplacement d'équipements dans le bâtiment).

Dans le cas d'une architecture BOS centralisée, les capacités du concentrateur central doivent être identiques à celles décrites ci-dessus.

2. Qualification des données

Le BOS a pour objectif de transmettre des données acquises sur le terrain à des applications servicielles (locales ou dans le cloud). Ainsi, afin de donner une meilleure vision sur ces données, le BOS devra fournir nativement la capacité à qualifier ces données. C'est-à-dire à associer aux données acquises la notion d'état de ces données (présence de dérogation, perte de communication, défaut, alarme, etc.). Ces états peuvent être utilisés localement ou transmis comme toute autre donnée.

3. Types de données

Le BOS doit faire l'acquisition des valeurs temps réels, mais également des historiques, des alarmes et des programmes horaires. Ainsi, le BOS pourra gérer :

- Des données temps réel (tous protocoles terrain ou IoT)
- Des historiques (Mbus, BACnet, IoT)
- Des alarmes (SNMP, BACnet, IoT)
- Des programmations horaires (BACnet, IoT)

Si le mode de communication ne propose pas tous ces types de données (par exemple : seules les valeurs temps réel sont disponibles), le BOS doit permettre de les créer à partir des données disponibles via des interfaces utilisateurs dédiées.

4. Hiérarchiser les ordres de commande

L'intégration des données comprend :

- La lecture des informations
- La capacité à envoyer des ordres (commandes).

En conséquence, les applications de services externes doivent pouvoir agir sur le système (selon leurs droits). Dans cette situation, le BOS doit définir une politique de priorisation des ordres qu'il reçoit afin de déterminer quel service externe ou automatisme local est prioritaire.

Exemple : on peut définir qu'une télécommande virtuelle sur smartphone est prioritaire en journée, mais que les programmes horaires de la GTB redeviennent prioritaires en période d'inoccupation.

Le BOS doit fournir les outils pour gérer ce type de situation en permettant plusieurs niveaux hiérarchiques ou de priorisation sur les commandes des équipements qui lui sont connectés. Le BOS doit également fournir les retours d'état des commandes aux applications afin que chacune puisse s'adapter en fonction de la commande réellement appliquée.

C. Structuration des données

La première étape vers la valorisation des données est la contextualisation de ces données afin de les rendre directement compréhensibles pour les applications tierces. Pour ce faire, le BOS utilise un modèle sémantique : système permettant de décrire les données acquises indépendamment de leurs origines.

Ce modèle sémantique doit être appliqué aux données temps réel, aux historiques, aux alarmes et aux points virtuels (données calculées à partir des données brutes ou paramètres utilisés pour interagir avec les systèmes tiers ou terrain).

Exemple : Consigne ou moyenne de températures.

1. Modélisation des données

L'objectif de la modélisation des données est de fournir une méthodologie uniforme et normalisée afin de décrire la signification des données.

Les Entités

La modélisation est basée sur la description des ENTITÉS du système. Ces entités peuvent correspondre à un équipement, une zone, une mesure, un preneur, etc.

ENTITES	Nom d'affichage
Capteur 12	Capteur de présence
Compteur 1	Compteur éclairage

Entités

Les Tags

La description d'une entité est basée sur l'ajout de métadonnées, aussi appelées TAGS, à chaque ENTITÉ.

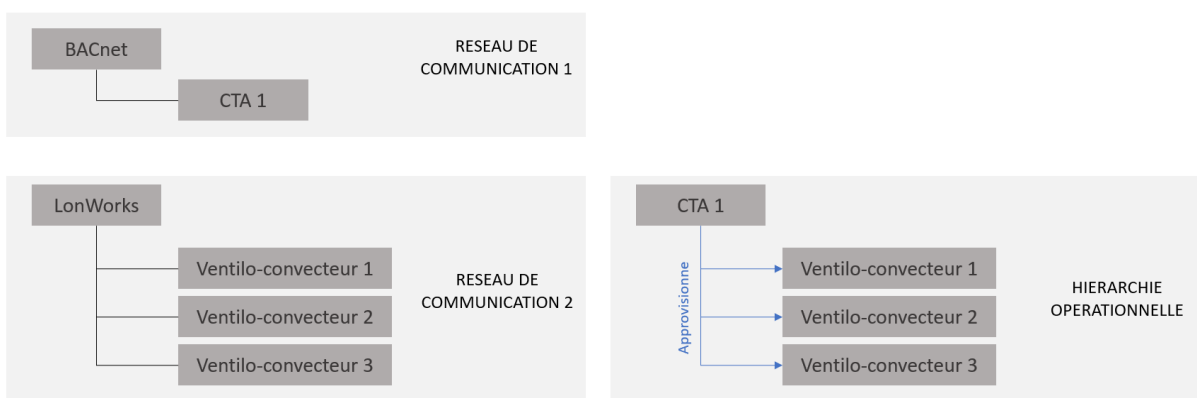
ENTITES	Nom d'affichage	TAGS	Valeur du tag
Capteur 12	Capteur de présence	Type :	Détection présence
		Zone :	Bureau 12
		Preneur :	Société ACME
Compteur 1	Compteur éclairage	Type :	Electricité
		Usage :	Eclairage
		Zone :	Etage 1
		Position :	TD 12
		Preneur :	Société ACME

Entités & Tags

Les relations

Les interactions entre des entités sont décrites par des RELATIONS. Cela permet de compléter l'architecture d'acquisition basée sur les réseaux de communication par une représentation fonctionnelle basée sur ces relations pour obtenir des hiérarchies opérationnelles.

Exemple : dans le schéma suivant, les ventilo-convecteurs et la CTA sont des entités qui peuvent être caractérisées par des tags. Les flèches bleues représentent des relations qui peuvent être typées (celles-ci pourraient s'appeler "approvisionne").



Architecture d'acquisition

Représentation fonctionnelle

Le mécanisme de modélisation des données du BOS doit être compatible avec des modèles sémantiques standards tels que Brick et Haystack. En effet ces modèles ont déjà été standardisés et adoptés par la communauté.

Exemple : plusieurs systèmes de dashboards et d'analytics utilisent Haystack comme base principale pour les règles. Autre exemple : Brick est utilisé par des applicatifs de ChatBot.

Le modèle sémantique doit être extensible pour s'adapter aux bâtiments et à leurs usages facilement. Ainsi, il permet d'enrichir les modèles fournis à l'origine par de nouvelles descriptions correspondant aux besoins des clients/usagers ou de ses prestataires de services.

2. Types de modèles de données

Le BOS doit permettre la construction de différents modèles de données, appelés Aspects, afin de décrire l'environnement dans lequel s'inscrivent les données gérées suivant différents points de vue fonctionnels. Un Aspect peut être assimilé à un point de vue.

Le BOS permet la création de nouveaux Aspects : ainsi, certains modèles de données propres à un type de bâtiment ou à un type de client pourront être créés.

Les modèles génériques suivants doivent être disponibles nativement :

- Aspects Géographique - par exemple :
 - *Bâtiment*
 - *Etage*
 - *Bureau*
 - *Salle de réunion*
 - *Couloir*
 - *Zone*
 - *Locaux techniques*
- Aspect Nature - par exemple :
 - *Mesure*
 - *Air*
 - *Température*
 - *Pression*
 - *Hygrométrie*
- Aspect Système - par exemple :
 - *Réseau*
 - *Equipements*
 - *Points*
 - *Alarmes*
 - *Historiques*
- Aspect Métiers/lots GTB
- Aspect Preneurs

Les structures de ces aspects (définition de l'arborescence des entités liées à un aspect) devront être fournies en annexe par la société en charge de la prestation. Les tags et les relations utilisés par chaque aspect devront pouvoir être modifiés et enrichis librement. Nativement, le BOS doit fournir et utiliser le dictionnaire de tags et relations Haystack.

3. Capacité d'import des modèles de données (gestion du BIM)

Les modèles de données pourront être créés de manière autonome dans le BOS (via l'interface graphique). Ils pourront également être créés automatiquement à partir d'une source externe.

A minima, les connecteurs suivants doivent être disponibles pour manipuler les modèles de données :

- Import de fichier Excel ou .csv
- API Web

Notamment, l'aspect géographique doit pouvoir être importé par un connecteur BIM afin de disposer de la même sémantique que celle définie dans la maquette BIM. De même, une nomenclature liée à une GMAO doit pouvoir être importée comme un nouvel aspect.

4. Gouvernance de la donnée

La gouvernance des données définit les règles de gestion des données et des métadonnées, garantit le respect de l'utilisation de ces règles et leur bonne mise en application. Elle permet également d'assurer leurs évolutions dans le temps en fournissant des outils dédiés.

Contextualisation

Il s'agit d'outils pour contextualiser les données, tels que la :

- Gestion des affectations : cartographie des données réelles ou virtuelles associées aux modèles de données (notion d'Aspect)
Exemple : Affecter un point booléen fourni par une API IoT aux Entités "Capteur occupation", "Salle de réunion 102", "CVC", "Société ACME".
- Gestion de valeurs complémentaires : pour caractériser de manière plus précise une donnée
Exemple : Création de listes de choix pour définir les usages RT qui seront utilisés dans les rapports de consommations et qui doivent être définis pour chaque compteur du site.
- Gestion des relations entre les entités pour pouvoir définir une interaction entre deux entités
Exemple : Lors de l'affectation d'un équipement à l'entité "Ventilo-Convecteur", il est demandé de renseigner par quelle CTA, cet équipement est alimenté.

Ces contextualisations doivent pouvoir être gérées dynamiquement dans le temps par le système (Création, Mise à jour, Nettoyage). De même, le BOS doit fournir des capacités pour renseigner ces informations de manière efficace (import Excel, API, etc.)

Accès à l'information

Un moteur de recherche intégré au BOS doit permettre d'exécuter des requêtes basées sur la sémantique des données dans un but de récupération d'informations, d'application de commandes ou de re-programmation du système. Ainsi, le BOS doit fournir des fonctions construites sur le modèle sémantique :

- Appel de synthèses
Exemple : Synthèse des alarmes par étage ou par niveau de criticité
- Appel de commandes générales
Exemple : Extinction de tous les éclairages bureaux ou de tous les équipements de confort
- Utilisation d'arbres de navigation dynamiques (hiérarchies)
Exemple : Arborescences des productions et de la distribution de chaleur ou électrique
- Re-programmation automatique
Exemple : Changement d'un preneur attribué à une zone qui va déclencher des modifications dans les droits d'accès, les rapports ou l'interface graphique de façon automatique

Ces fonctions sont indépendantes des données présentes dans le système car leur définition est générale et doit être basée uniquement sur la sémantique utilisée. Cela permet de créer des calculs dynamiques et standards sans se soucier du format d'origine des données.

Exemple : la définition d'un calcul "somme des consommations par usages et par étages" est indépendante du nombre de compteurs installés mais nécessite bien de décrire quel compteur correspond à quel usage ou à quel étage dans le modèle sémantique. En cas d'ajout de compteur, il suffira d'indiquer à quels étages ou quels usages, ils sont dédiés pour que le calcul se mette à jour automatiquement.

5. Traitement local des données

Le BOS doit permettre d'appliquer des traitements sur les données acquises. Il fournit donc la possibilité de créer des fonctions logiques et des calculs mathématiques via des bibliothèques fournies nativement afin de créer des KPI, des ratios, des agrégations, des synthèses, etc. et de réaliser de la consolidation de données.

Une fois créés, les résultats de ces traitements doivent pouvoir être gérés exactement comme les points de communication bruts et bénéficier des mêmes traitements (réutilisation pour d'autres logiques, alarmes, historiques, etc.).

La capacité à réaliser des synthèses ou d'autres traitements est primordiale pour réduire le trafic des informations entre le(s) bâtiment(s) et les services externalisés. Cette capacité permet également de délocaliser et de répartir la puissance nécessaire à tous les calculs, réduisant d'autant le coût de fonctionnement des systèmes tiers.

Exemple : le fait de calculer une consommation électrique journalière par compteur permet l'envoi d'un seul message par jour et par compteur sur une plateforme tierce et de diviser par 50 ou 100 le nombre de messages transmis (le coût d'une plateforme tierce peut être évalué en fonction du nombre de messages transmis par jour).

D'autres fonctions doivent être disponibles afin d'assurer un monitoring efficace sur les données acquises :

- Alarmes (gestion de groupes, présence/acquittement, gestion des inhibitions et délais, gestion des avalanches d'alarmes et de leur escalade)
- Historiques (échantillonnage par changement de valeur ou intervalle, groupage des historiques)
- Compteurs d'événements
- Calcul de temps de fonctionnement
- Intégration de valeurs

D. Mise à disposition des données en lecture et écriture

D'un point de vue "échanges de données", le rôle du BOS est :

- De transmettre les données gérées ainsi que leurs métadonnées à tous types d'applications en charge de les traiter ou de les analyser pour en tirer de la valeur
- Et d'appliquer des changements via le BOS suite à des envois d'ordres ou de commandes depuis les services tiers.

Cette fonction constitue l'interface avec des services externes que le propriétaire ou l'occupant souhaitent déployer dans le bâtiment.

La capacité du BOS à s'adapter aux applications servicielles en utilisant leurs propres moyens de communication est l'une des principales clés du succès. Il est donc nécessaire de pouvoir profiter d'un maximum de types de connecteurs intégrés au BOS.

La mise à disposition de ces données doit suivre le modèle sémantique mis en place. Les données envoyées vers une base de données tierce doivent comprendre les différentes métadonnées associées.

1. Connecteurs

Le BOS doit pouvoir gérer plusieurs connecteurs simultanément. L'ajout de nouveaux connecteurs doit être ouvert, c'est-à-dire qu'ils peuvent être développés facilement dans le BOS.

Le BOS doit permettre de transmettre les données et les métadonnées à des applications servicielles via :

- Envoi de fichiers (CSV, TXT, JSON, XML, HTML) par mails et sFTP
- Bases de données c (MS-SQL, MySQL, Elasticsearch, Oracle, MongoDB, Cassandra etc.)
- Protocoles IT/Web Services (oBIX, RESTful, MQTT, Kafka, OPC UA, API Haystack)
- Plateforme IoT (Microsoft Azure, Google IoT Core...)

Suivant les possibilités des plateformes ou bases de données tierces, le BOS doit aussi fournir la possibilité d'appliquer des ordres de commandes transmis depuis ces systèmes tiers. Il ne s'agit pas uniquement de remonter de la donnée.

Exemple : Une application tierce peut créer un nouveau document dans la base de données MongoDB.

Le BOS doit être capable de l'interpréter et de le transformer en commande à appliquer localement.

Autre exemple : L'envoi d'un message dans Azure IoT Hub doit être également appliqué dans le BOS.

2. API Rest native

Le BOS doit disposer également d'une API Rest native offrant davantage de possibilités à une application tierce pour interagir avec le BOS. A minima l'API doit couvrir les cas suivants :

- Enrichissement des modèles de données (ajout de bâtiments, d'étages, de preneurs, etc.)
- Lecture des différentes arborescences de données pour être capable de comprendre le système depuis l'extérieur selon les différents points de vue
- Création de nouvelles sources de données locales (ajout d'équipements IoT, ajout d'historisation etc.)
- Lecture et écriture de données temps réel
- Lecture des données historisées avec agrégation intégrée (le pré-traitement est directement réalisé par le BOS)
- Lecture et acquittement des alarmes

L'API doit gérer les permissions associées.

Exemple : Il est possible d'interroger le système avec une requête du type API Service Web HTTP Get pour récupérer toutes les entités d'un type donné dans le modèle (étages, équipements, preneurs, etc.) avec leurs valeurs associées :

<https://monBOS.io/getSummary?definition=compteurs&data=usage;value>

Cette requête répondra avec un tableau contenant autant de lignes que de compteurs disponibles sur le site, chaque ligne comprenant l'identifiant du compteur, sa valeur et l'usage mesuré.

3. Types de données transmises

Le BOS doit pouvoir transmettre différents types de données

- Les données temps réel
- Les historiques
- Les alarmes
- Les événements des programmes horaires
- Les temps de fonctionnement.

Suivant les besoins et les architectures, le BOS permettra également de mettre à disposition ou de recevoir les données calculées ou consolidées (ratios, synthèses, KPI, etc.)

4. Sécurité

L'échange de données implique la mise en place et le respect d'une politique de sécurité stricte, en lien avec la DSI du client. Nativement, le BOS doit respecter les bonnes pratiques de sécurité informatiques. Notamment :

- Toutes les connexions du système doivent utiliser des moyens de communication cryptés type SSL/TLS (en utilisant des certificats de sécurité signés par une autorité tierce),
- Personnalisation des ports IP utilisés,
- Gestion des droits d'accès au plus proche du besoin (un utilisateur doit se voir attribuer le strict minimum des droits nécessaires à son activité),
- Comptes nominatifs avec gestion des rôles (les comptes génériques type "opérateur", "admin" sont proscrits),
- Modification obligatoire des comptes par défaut,
- Politique de gestion des mots de passes avec définition de la dureté, renouvellement automatique et interdiction de réutiliser les mêmes mots de passes consécutivement,
- Protection contre les attaques en force brute,
- Encryption des données sensibles et des mécanismes de login,
- Utilisation de login durcis (SCRAM SHA 256 par exemple, les login BASIC sont proscrits),
- Gestion d'un espace partagé pour les connexions Web et accès protégé au reste des serveurs
- Utilisation de technologies LDAP et SAML pour la gestion centralisée des mots de passe et utilisation d'IdentityProvider pour le Single Sign On (SSO)

De nombreux sites interdisent toute connexion entrante depuis l'extérieur. Dans ce cadre, un service hébergé dans le Cloud pourrait ne pas pouvoir se connecter à l'API du BOS suite à des restrictions réseaux. Pour pallier ce problème, il sera possible d'utiliser un connecteur en mode "connecté permanent" permettant un échange des données dans les deux sens (lecture et écriture) mais nécessitant uniquement une connexion sortante.

E. Services locaux

Les données opérationnelles du(des) bâtiment(s) sont stockées au sein du BOS et alimentent en données l'ensemble des applications servicielles.

Les deux services suivants doivent être livrés de base dans le cadre d'une rénovation lourde ou d'une installation neuve qui a pour but de remplacer la Gestion Technique du Bâtiment par un BOS couvrant cette fonction.

1. Supervision technique

Le service local de supervision technique communique nativement avec le BOS mis en place.

Cette supervision technique est destinée à fournir les outils nécessaires à l'exploitation du bâtiment :

- Synoptiques animées
- Consoles d'alarmes
- Outils de paramétrages
- Accès aux historiques

L'interface homme-machine doit être accessible depuis un navigateur Web classique, sans coût de licence par poste utilisateur.

Les outils et fonctionnalités détaillés de la supervision technique font l'objet d'un Cahier des Charges fonctionnel dédié.

Quelques principes de base devront être inclus dans ce service :

- Droits d'accès sécurisés
- Reporting (courbes, rapports)
- Outils de suivi du commissioning
- Navigations dynamiques entre les interfaces graphiques (géographie, fonctions, système, métiers, etc.)
- Rapports d'exploitation (inhibitions, forçages, rapports sur les alarmes)
- Synoptiques animés, utilisation de synthèses pour guider la navigation
- Possibilité de modifier tout paramètre fonctionnel depuis l'interface de supervision directement (seuils d'alarmes, adresses mails pour envoi rapport, adresses des équipements repris etc.)

Les fonctions nécessitant des recherches de données ou des croisements de données doivent être basées sur les modèles sémantiques du BOS.

Exemple : Rapport de consommation par usage et par preneur (les « usages » et les « preneurs » correspondant à des définitions des modèles.

2. Gestion énergétique

Conformément aux recommandations de la SBA, à la livraison du système, un service de gestion énergétique communique nativement avec le BOS.

Cette application doit fournir des tableaux de bord de synthèse des consommations du(des) bâtiment(s) ainsi que le diagnostic de performance énergétique (rapport DPE).

La solution choisie doit également fournir un ensemble de règles d'alarmes pour identifier d'éventuelles dérives dans les consommations Électrique, de Gaz et d'Eau.

Les outils et fonctionnalités détaillés de la supervision technique font l'objet d'un Cahier des Charges fonctionnel dédié.

Auteur

VayanData est une société spécialisée dans le Smart Building qui propose des solutions pour simplifier le déploiement de services dans les bâtiments. Son équipe technique dispose d'une longue expérience dans la mise en œuvre de systèmes de gestion technique et dans les réseaux de communication du bâtiment. VayanData propose une vision ouverte du Building Operating System afin de donner aux acteurs du marché les clés d'analyse indispensables pour comprendre le fonctionnement et les enjeux du BOS.