

# Technologie OPC UA

## Présentation de la technologie 1<sup>er</sup> jour



### Michel Condemine

- OPC Foundation France (Technique)
- Directeur de **4CE Industry**
- Leader du projet OpenOpcUa
- MichelC@4CE-Industry.com





## Généralités sur OPC UA

### OPC UA 1.0.5

- 24 Parties
- 80 Companion Specifications
- +7000 pages



### Distinguer

- spécification
- implémentation



### OPC UA n'est pas qu'un protocole de communication.

- ☐ C'est une technologie
- ☐ C'est plusieurs protocoles :  
TCP, UDP, HTTP, MQTT, AMQP, WEBSOCKET
- ☐ C'est plusieurs encodages  
Binaire, XML, JSON
- ☐ C'est un métamodèle  
Une langue pour exprimer des modèles



## La fondation OPC

### Son rôle

1. Concevoir et maintenir des spécifications d'interopérabilité
2. Certifier les produits OPC UA
3. Collaborer
4. Promouvoir la technologie OPC





# La Fondation OPC

## Pourquoi ?

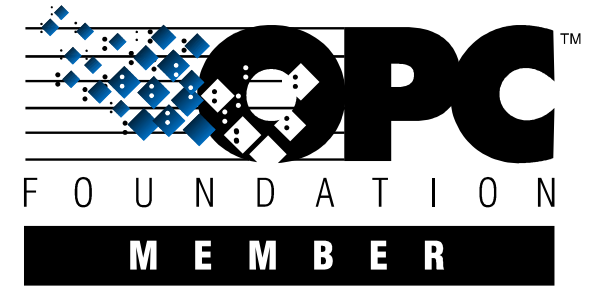
- ✓ Pour concevoir des spécifications pour l'interopérabilité des systèmes.
- ✓ Pour promouvoir la technologie OPC.

## Où ?

La fondation OPC est une organisation mondiale dont le siège est basé aux USA.

OPC Foundation  
16101 N. 82nd Street, Suite 3B  
Scottsdale, AZ 85260-1830 - USA  
Phone: 480-483-6644  
Fax: 480-483-7202

- ✓ [www.opcfoundation.org](http://www.opcfoundation.org) [USA]



Transition



**“Interoperability Standard for Industrial Automation”**

**“The Industrial Interoperability Standard”**

Quatre niveaux de membres :

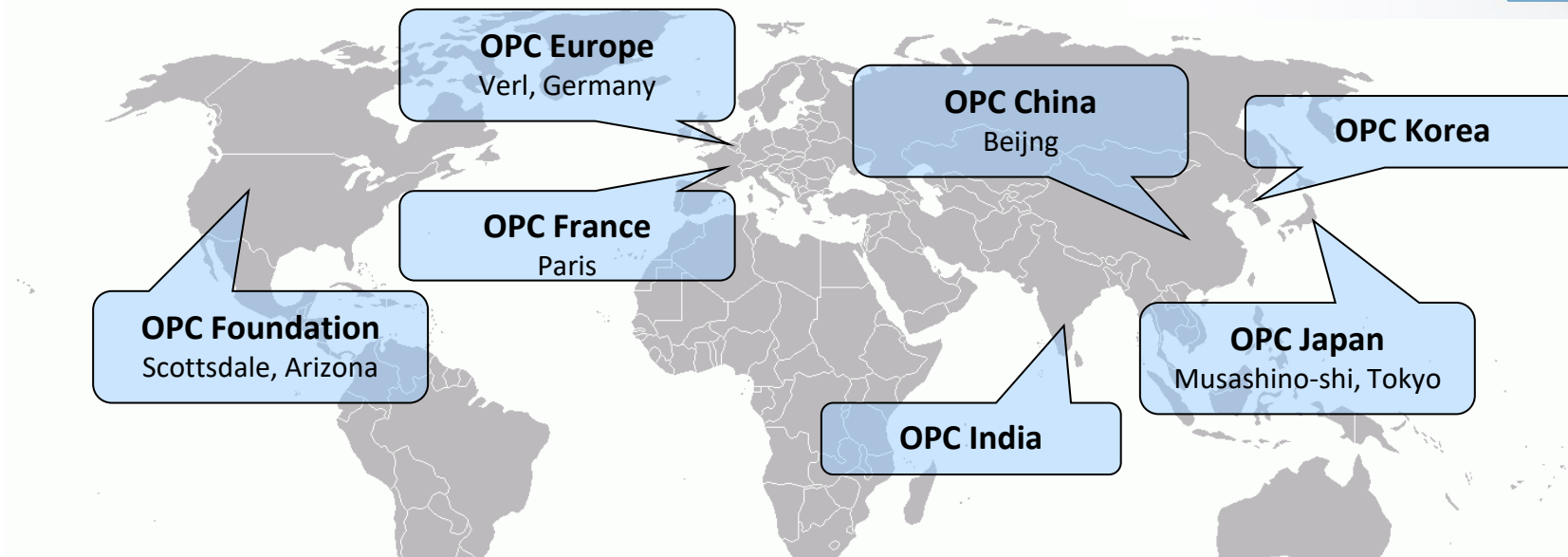
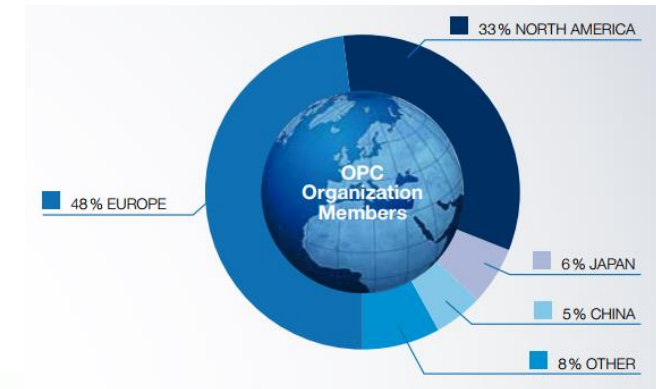
- Corporate member
- End-User
- Non-Voting member
- UA Logo Member

Membership Benefits	Corporate Member	End-User Member	Non-Voting Member	UA Logo Member
IP policy protection	✓	✓	✓	
Access to OPC UA specifications	✓	✓	✓	✓
Access to OPC Classic specifications	✓			
Evaluation of OPC Foundation source code	✓	✓	✓	✓
Use of OPC Foundation source code in commercial products	✓			
Distribution of OPC Foundation source Code	✓			
Opportunities to join working groups, online discussion groups, and collaborations for new technology initiatives	✓	✓	✓	
Access to new draft specifications through participation in	✓	✓	✓	

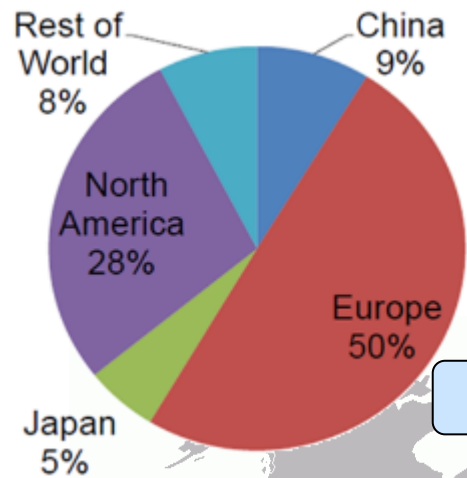
Plus de détails :

<https://opcfoundation.org/membership/benefits/>

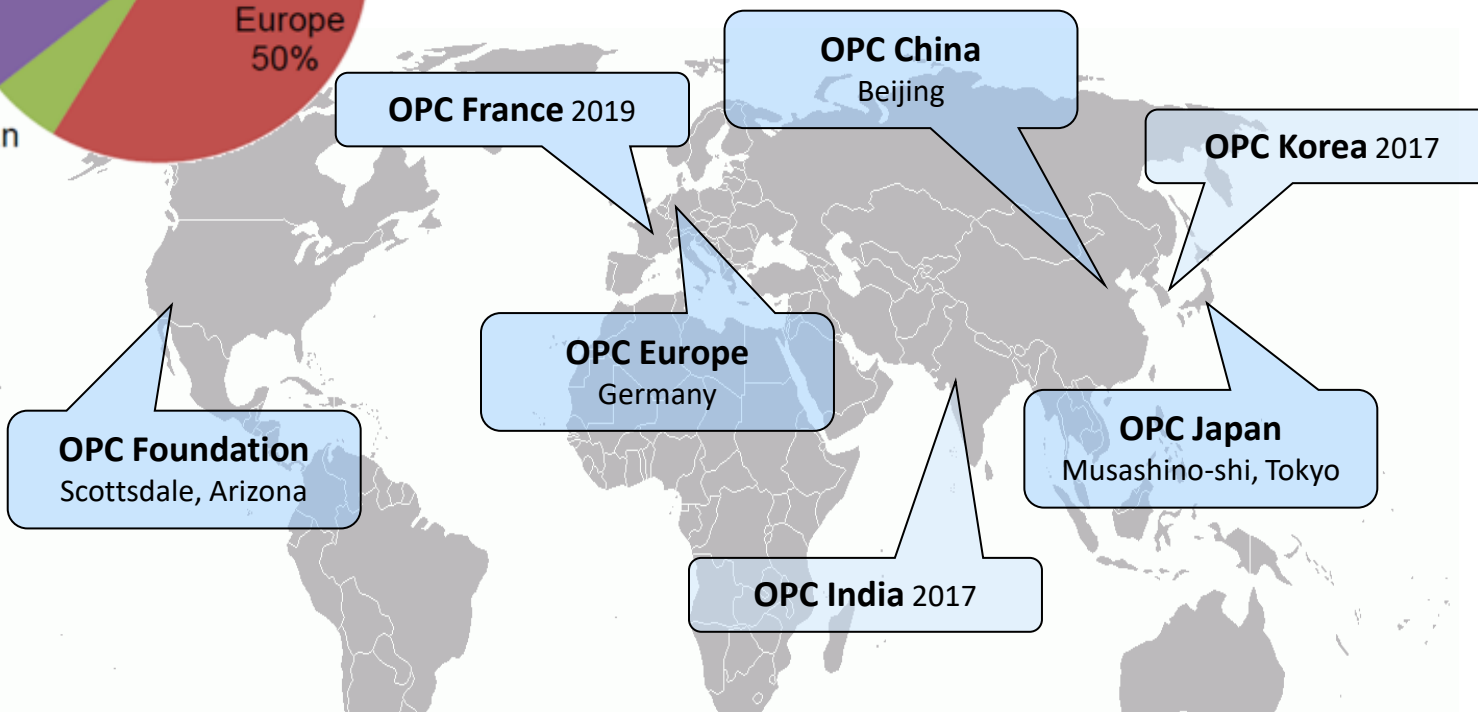
- ▶ +800 membres dans le monde
- ▶ +360 membres en Europe
- ▶ 40 membres en France



## Répartition des membres



<https://opcfoundation.org/france/>



## Où utilise-t-on OPC ?

- ☐ Systèmes automatisés
- ☐ Bâtiments intelligents
- ☐ Systèmes embarqués
- ☐ Gestion de l'énergie (Smart Grid)
- ☐ L'industrie manufacturière
- ☐ Cloud-based (Pub/Sub)
- ☐ M2M (TSN)
- ☐ Et bien d'autres ....



## OPC Fondation France Gouvernance

### Animateur



**Laurent SIEGFRIED**  
Délégué Smart Up Industrie  
GIMELEC

Email : [lsiegfried@gimelec.fr](mailto:lsiegfried@gimelec.fr)  
M : 06 83 83 99 03

Animateur du collectif

### Président



**Patrick LAMBOLEY**  
Innovation & Technology Industrial  
Automation Business  
Schneider Electric

Email : [patrick.lamboleyle@se.com](mailto:patrick.lamboleyle@se.com)  
M : +33 6 88 05 17 70

### Marketing



**Julien KOUUBI**  
Marketing and Communication  
Manager  
B&R Automation

Email : [julien.koubbi@br-automation.com](mailto:julien.koubbi@br-automation.com)  
M : +33 6 42 21 73 38

Pilotage des actions promotionnelles  
et événements

### Expertise technique



**Michel CONDEMINE**  
Directeur  
4CE Industry

Email : [michelc@4ce-industry.com](mailto:michelc@4ce-industry.com)  
M : +33 6 09 46 26 28

Pilotage cellule expertise

### Relations Institutionnelles



**Benjamin FRUGIER**  
Directeur - Développement des  
entreprises et Projets  
Fédération des Industries Mécaniques

Email : [bfrugier@fimeca.org](mailto:bfrugier@fimeca.org)  
T : +33 1 47 17 60 20

Pilotage relations institutionnelles

### Utilisateur-Prescripteur

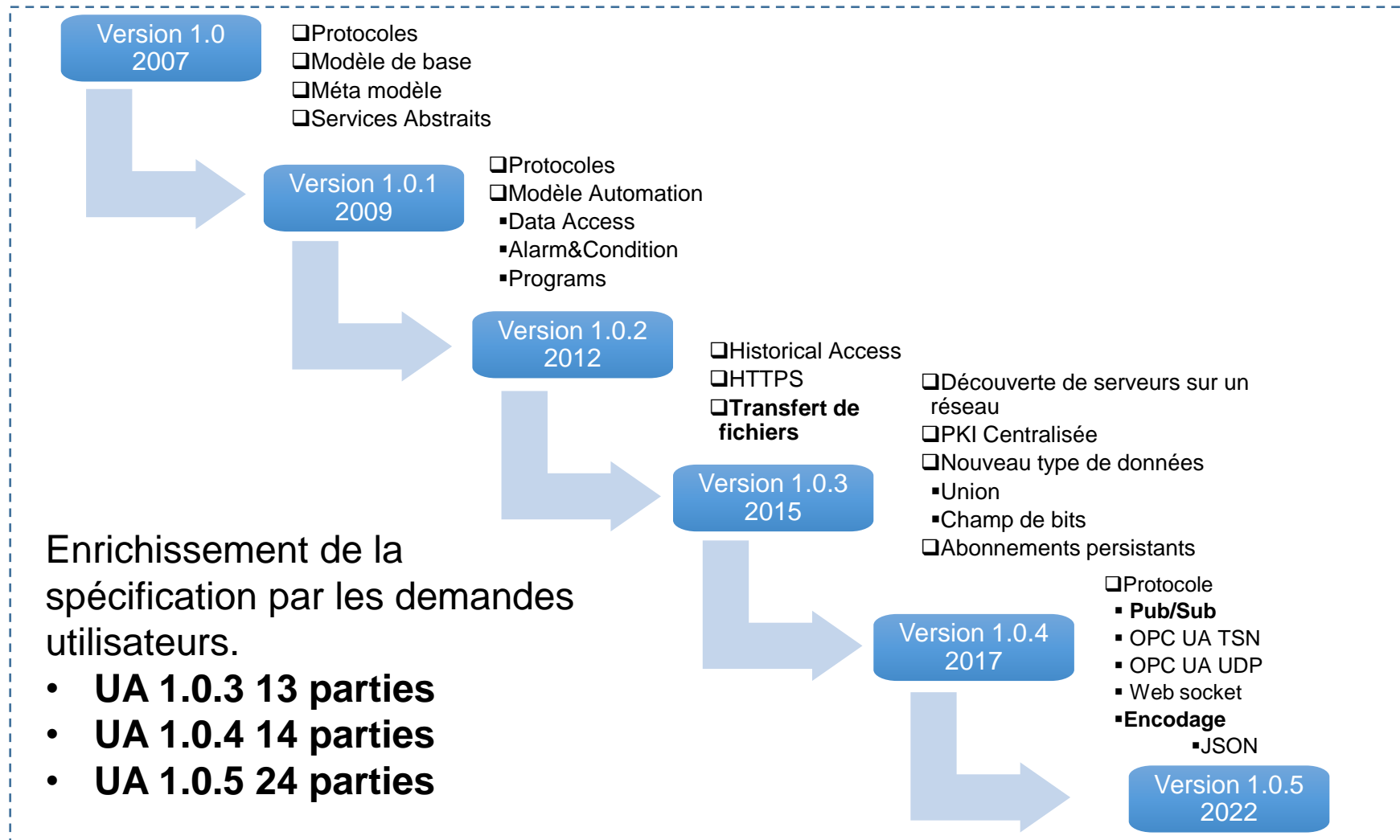


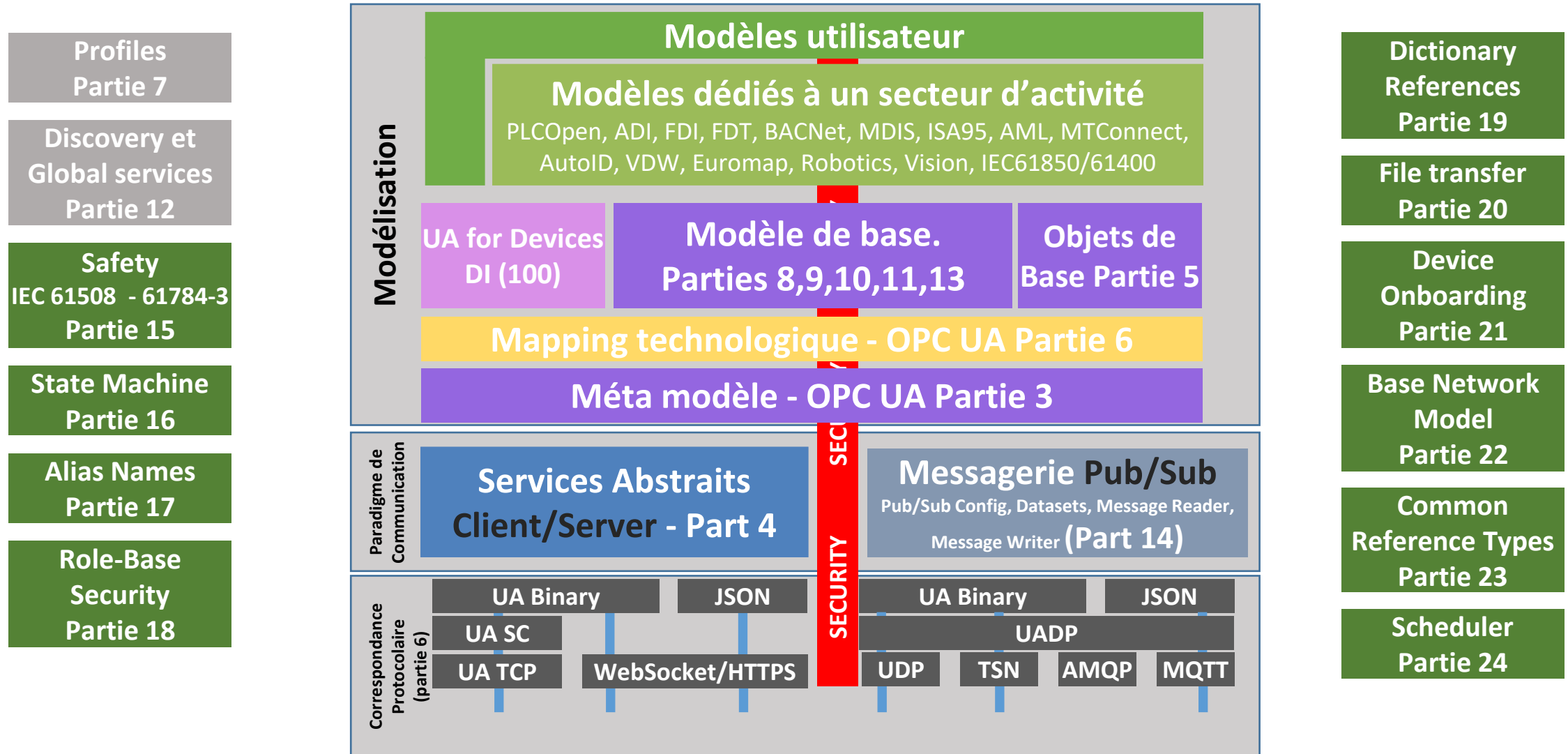
**Mariatou SARIA-BIARE**  
Corporate Filling & Packing Manager  
L'OREAL

Email : [Mariatou.SAGNA-BIARE@loreal.com](mailto:Mariatou.SAGNA-BIARE@loreal.com)  
T : +33 1 58 31 72 63

Pilotage expressions de la demande







### Sécuriser

- Chiffrer la donnée
- Identifier les clients



**OPC UA™**

### Adaptable

- Multiplateformes
- Topologies variées
- Multi-domaines
- ...



### Modéliser la donnée

- Identifier l'information
- Structurer l'information selon les exigences métiers
- Respecter les sémantiques métiers

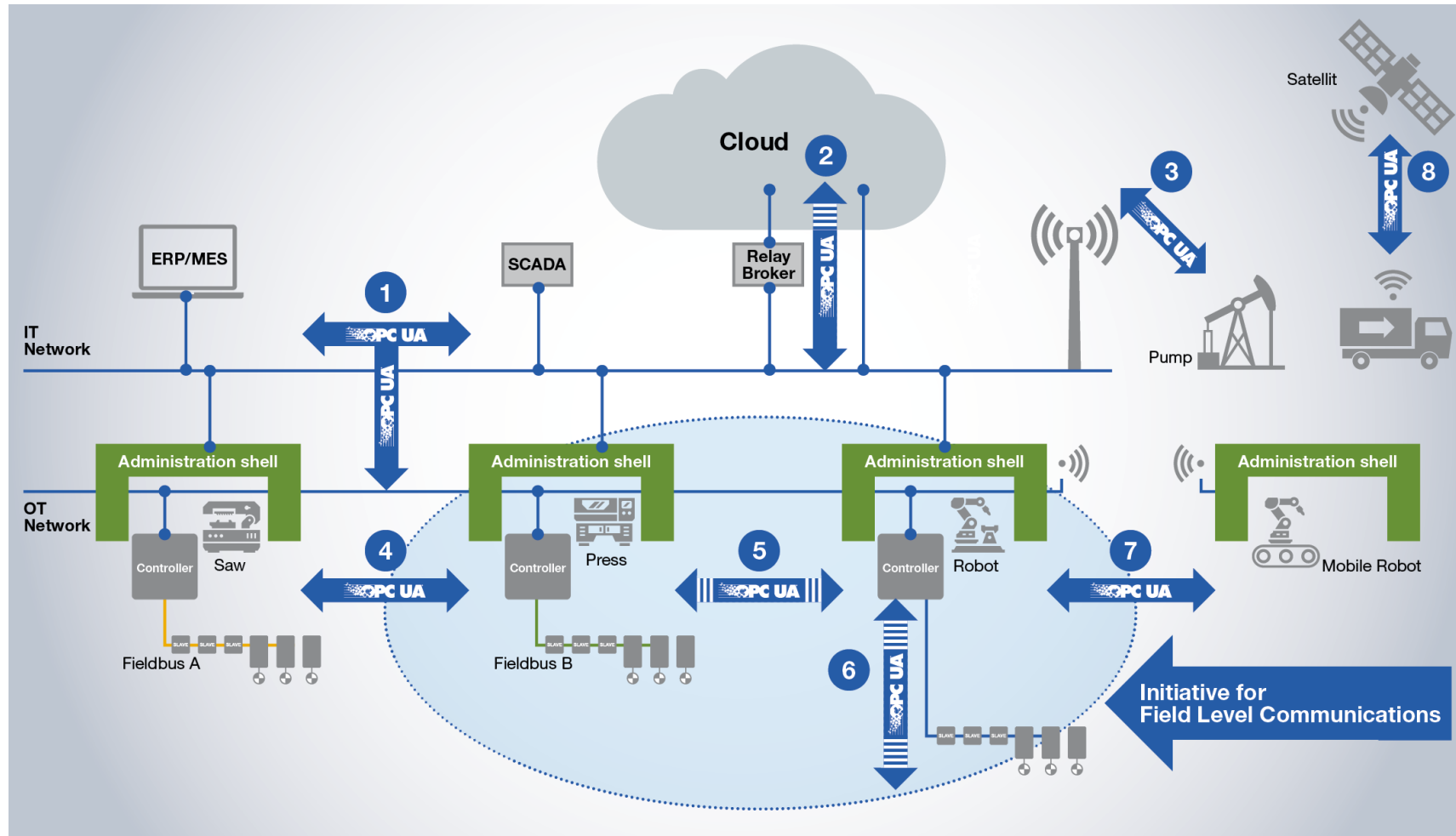




## Le contexte

### Producteur/collecteur d'informations

- Producteur : PLC, ROBOT, RTU, etc.
- Consommateur : SCADA, MES, ERP, etc.

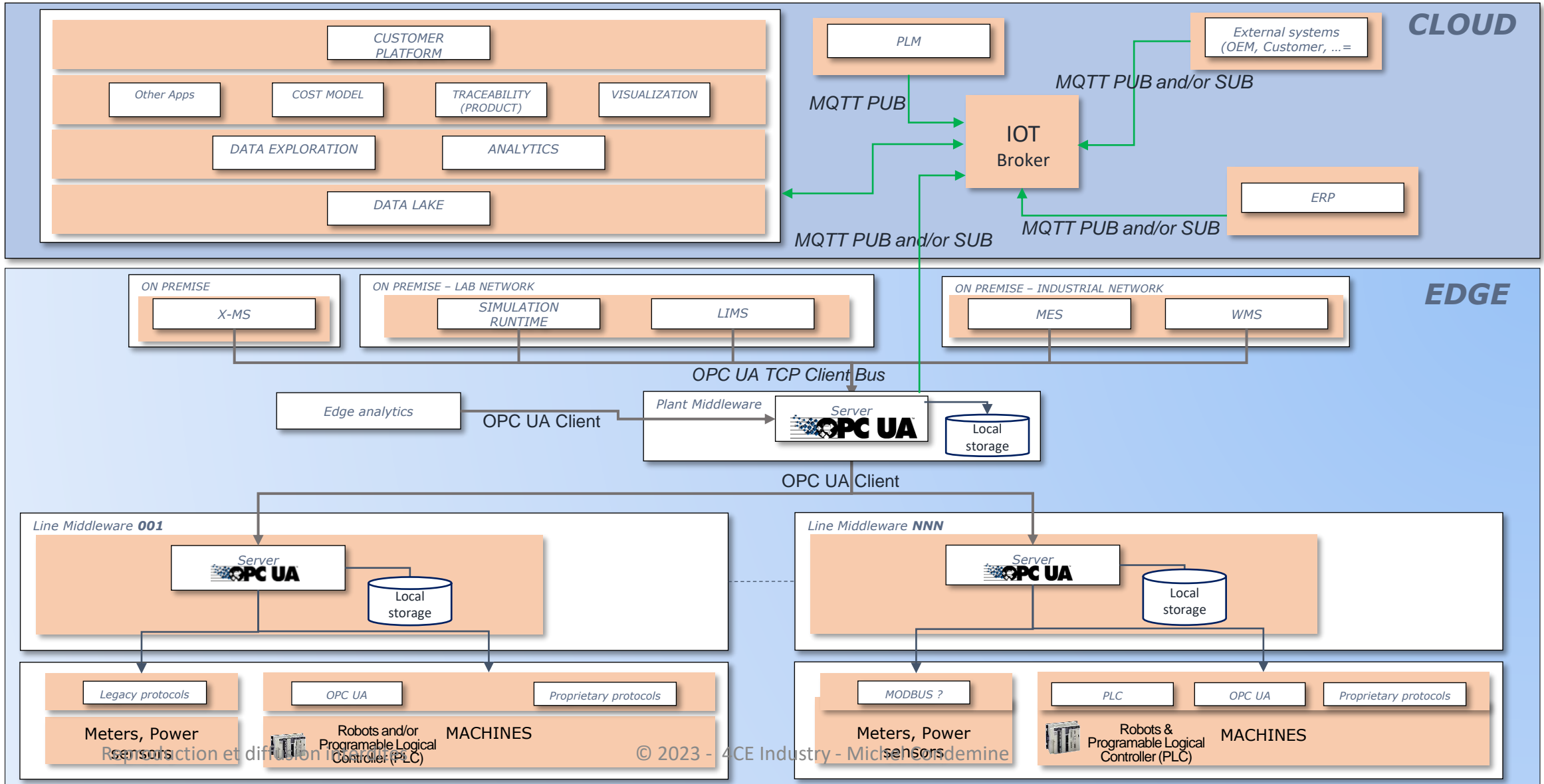


- 1 IT / OT Communication
- 2 Cloud Integration
- 3 Secure Remote Access
- 4 Local OT Communication
- 5 Controller to Controller
- 6 Controller to Field Device
- 7 Wireless Integration (5G)
- 8 Future Ready



# OPC UA (FIA)

## Full Integration Architecture

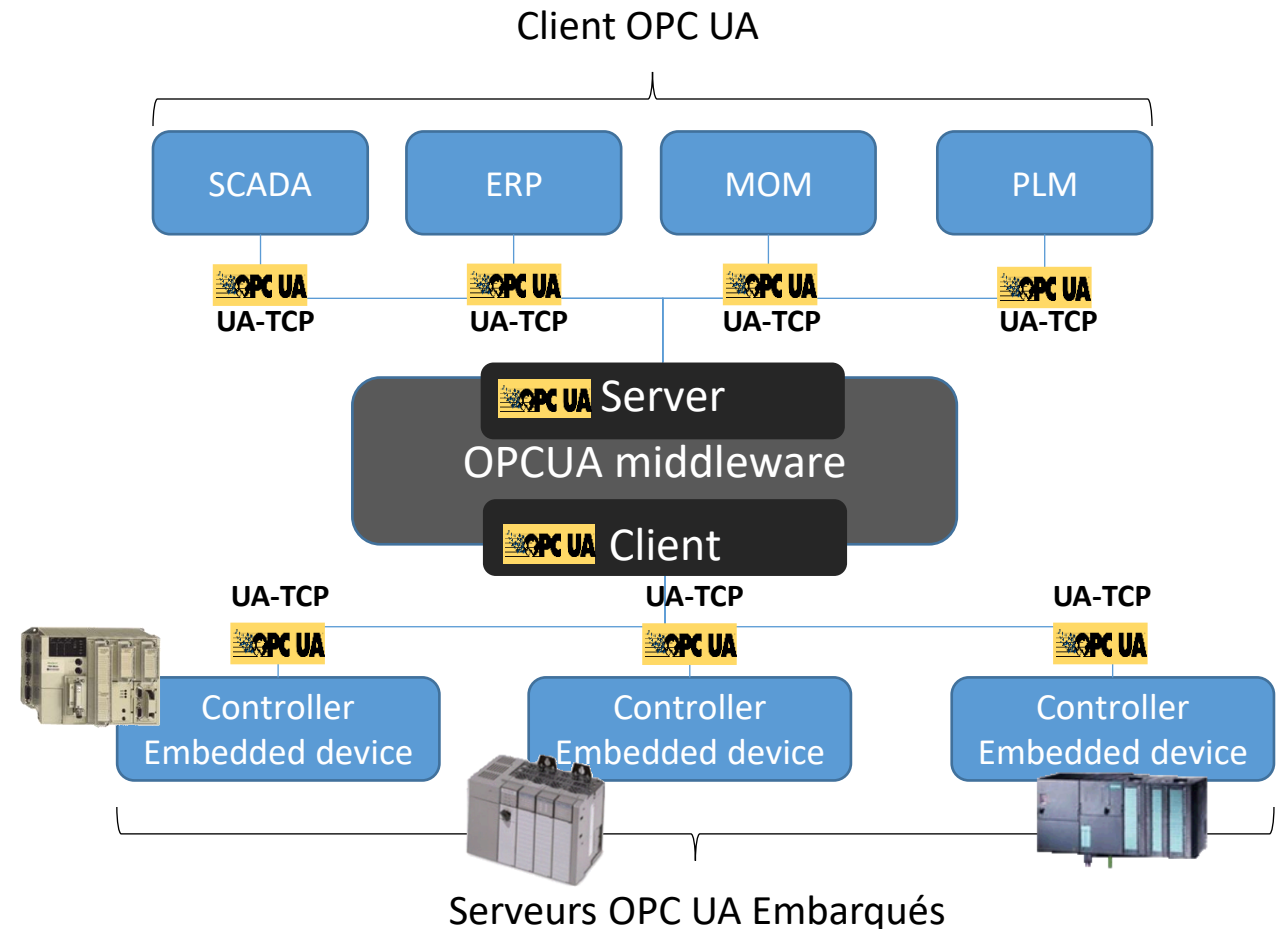




Echange d'informations entre des **équipements de terrain**, PLCs, CN, Robot, etc et des **applications IT**, SCADA, MOM, ERP, en passant généralement par un middleware OPC UA

Il s'agit d'échanges Top/Down, bidirectionnels

- non-déterministes
- Horodatés à la source par le système embarqué
- Sécurisés

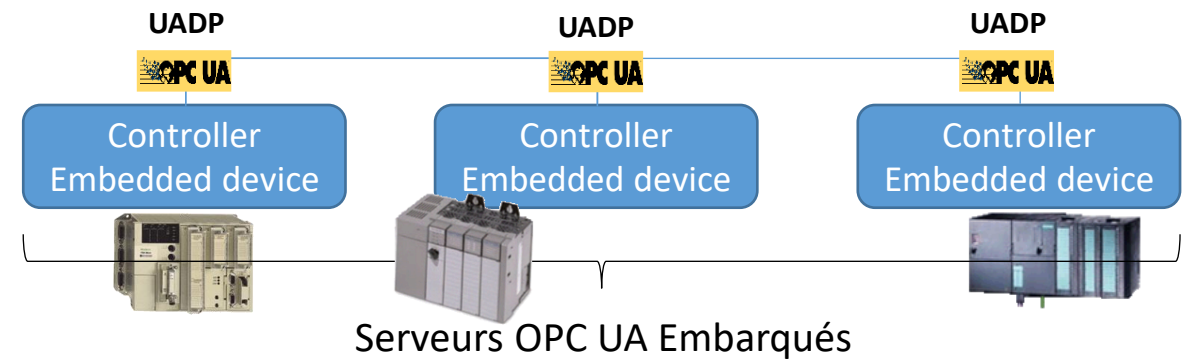




Echange d'informations entre des **équipements de terrain**,  
PLCs, CN, Robot, etc.

Il s'agit d'échanges M2M, bidirectionnels

- non-déterministes
- Horodatés à la source par le système embarqué
- Sécurisés



Echange d'informations entre des équipements de terrain, PLCs, CN, Robot, etc et des applications IT distantes, SCADA, MOM, ERP, en passant généralement par un middleware OPC UA et en traversant un web broker.

OPC UA propose :

### 3 solutions pour le transport

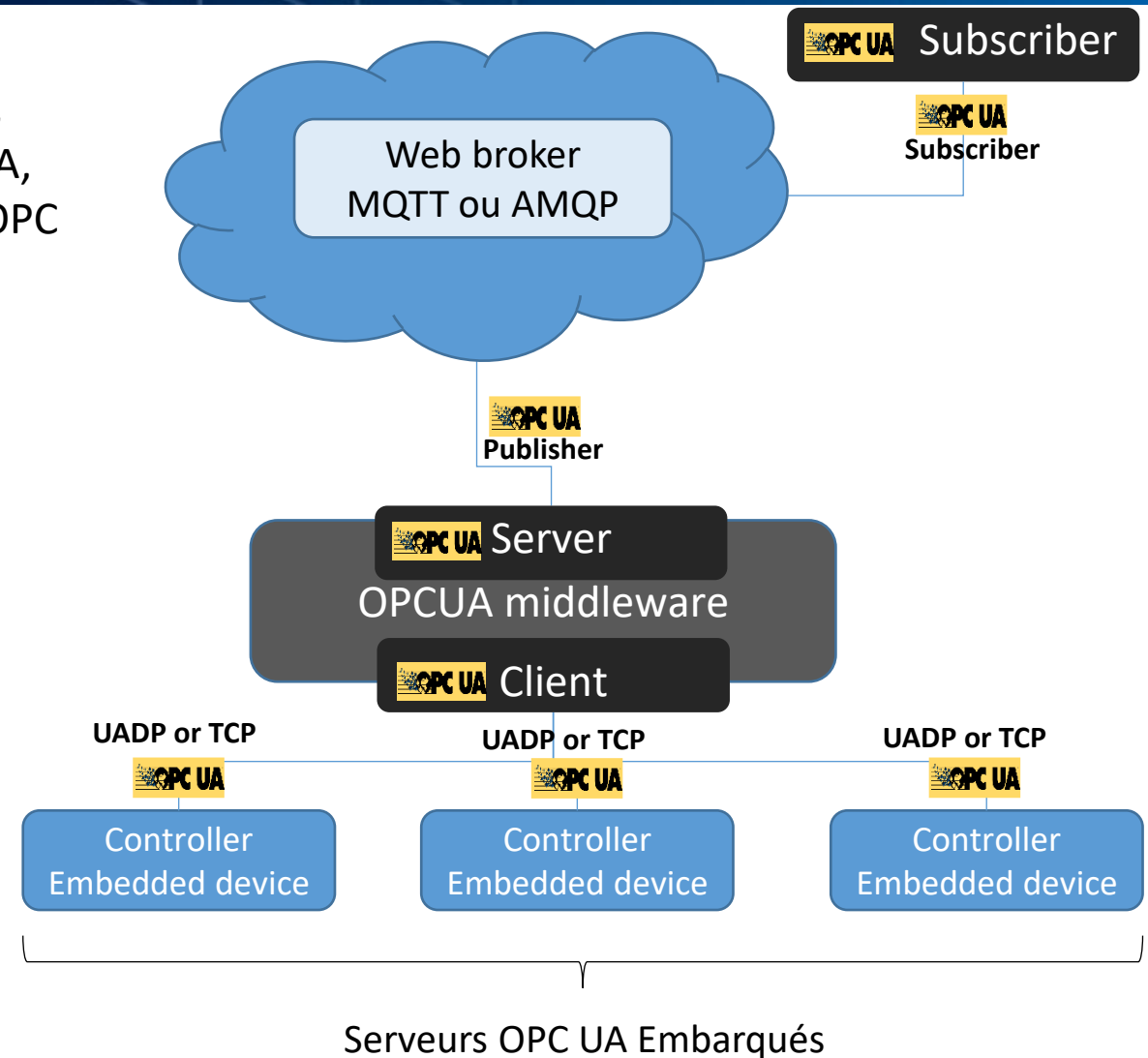
- MQTT
- AMQP
- WebSocket

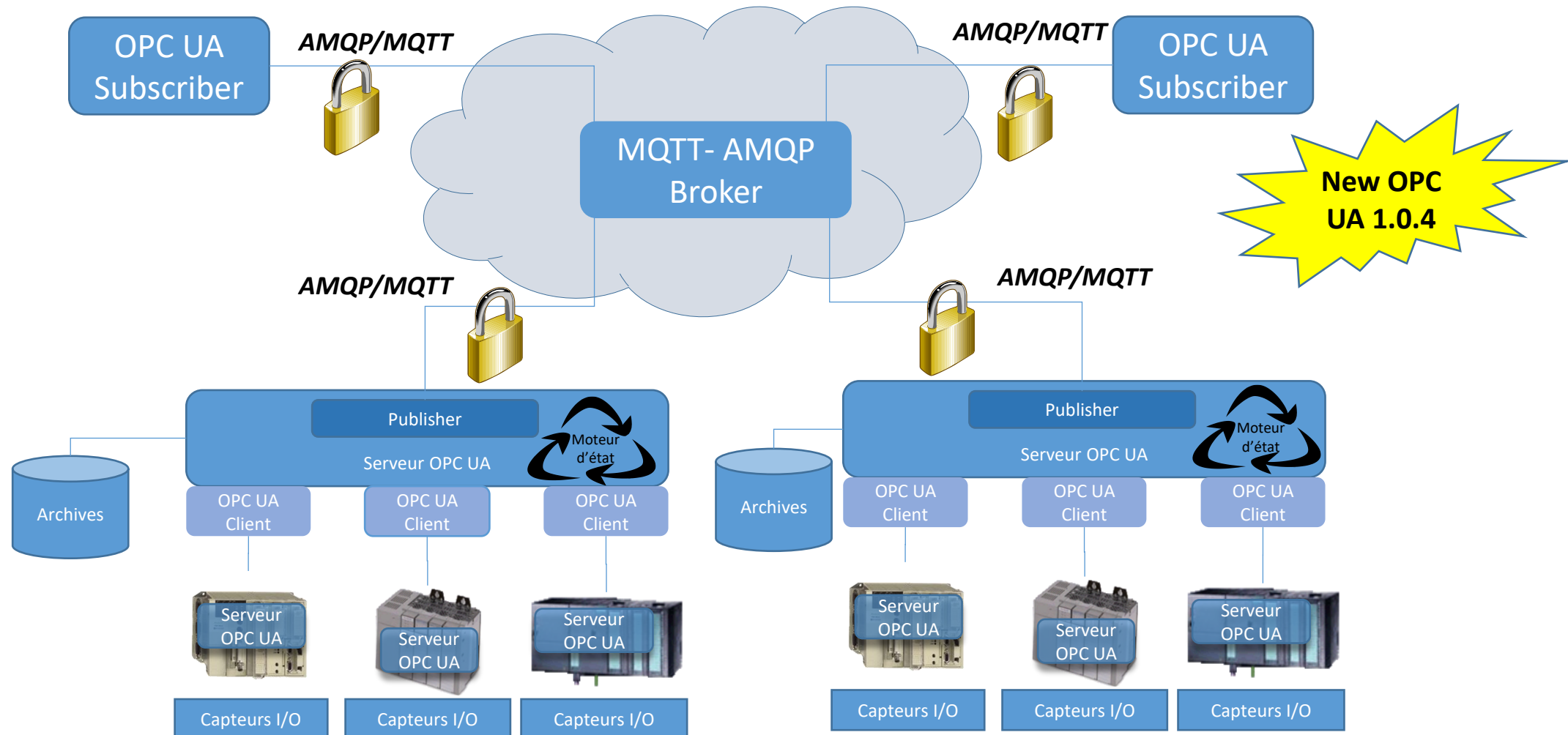
### 3 type d'encodages

- XML
- JSON
- Binaire

Il s'agit d'échanges Bottom/up bidirectionnels

- non-déterministes
- Horodatés à la source par le système embarqué
- Sécurisés



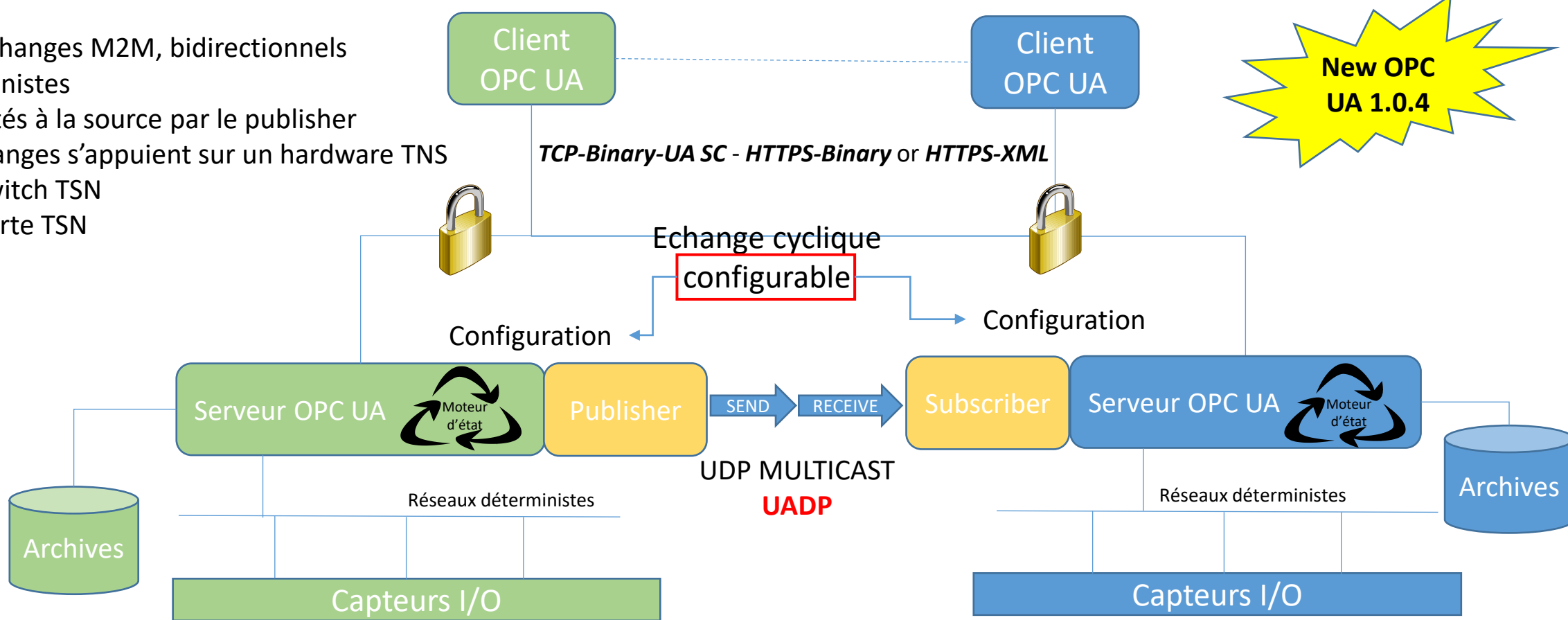


# OPC UA

## case d'usage : Communication M2 déterministe

Il s'agit d'échanges M2M, bidirectionnels

- Déterministes
- Horodatés à la source par le publisher
- Les échanges s'appuient sur un hardware TSN
  - Switch TSN
  - Carte TSN



On peut écrire des applications OPC UA :

- ☐ Avec tous les langages qui permettent de gérer les socket TCP/UDP
- ☐ Sur tous les OS et noyaux RT qui supportent ces langages
- ☐ Les ressources demandées par un serveur OPC UA dépendent de son **profil**
  - **50Ko de ROM pour les petits profiles**
  - **100Ko de RAM.**

**La RAM dépend du nombre, de secure Chanel, de session, de souscription, etc.**



Pour développer une application OPC UA vous devrez chiffrer les messages.  
Il vous faudra développer ou utiliser un API pour le faire. (*OpenSSL*)



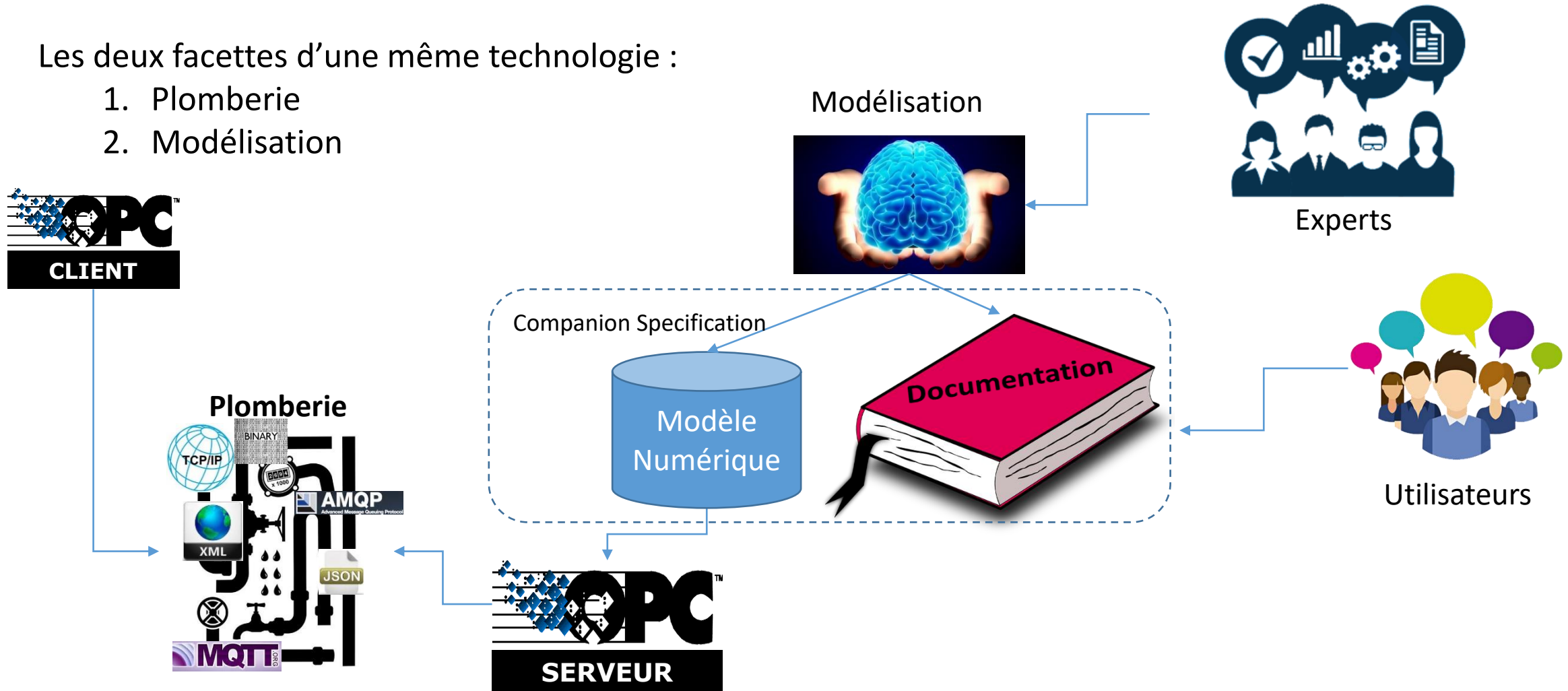


# Serveur OPC UA

## La notion de modèle informationnel

Les deux facettes d'une même technologie :

1. Plomberie
2. Modélisation





**Les nodes** sont caractérisées par au maximum 24 attributs :

avec  $x = n^\circ$  du namespace

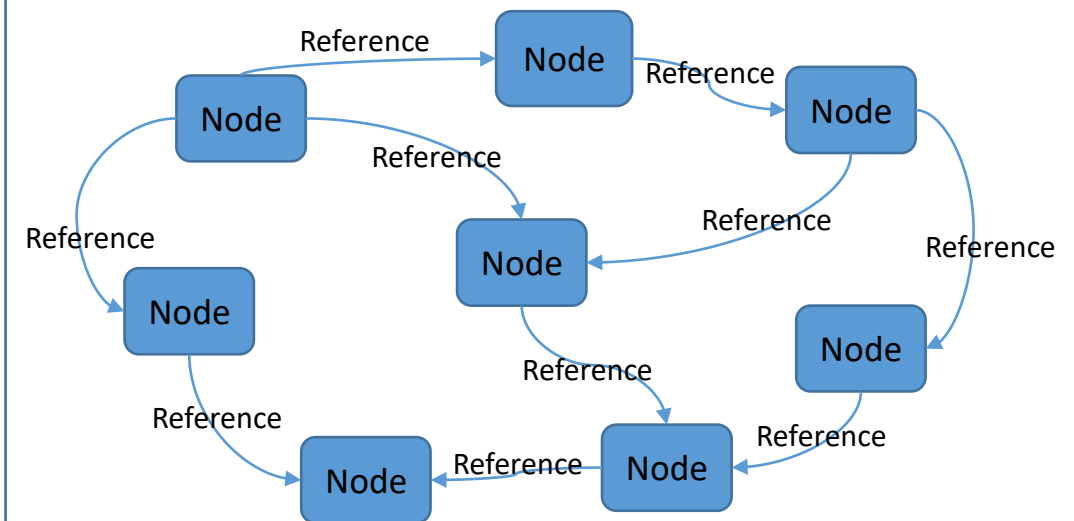
## ❑ NodeClass : Enum

❏ **DisplayName** : Chaîne localisée

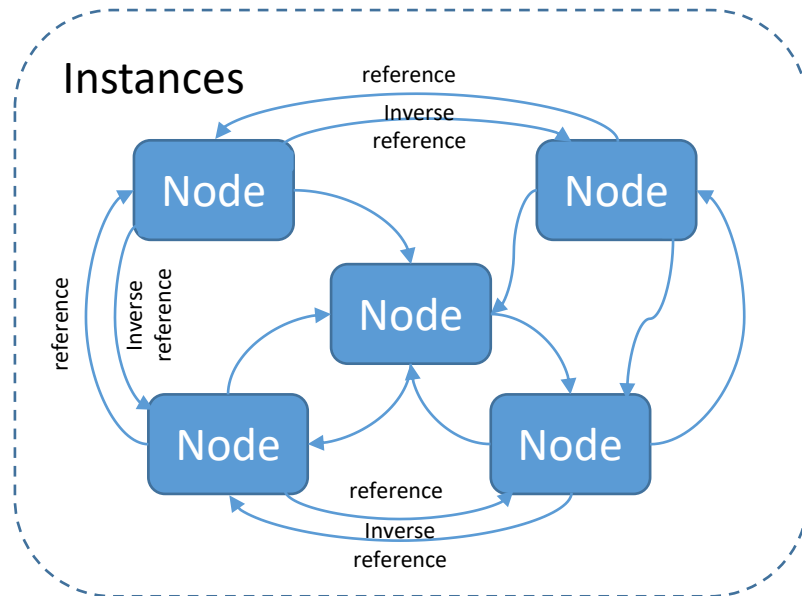
- ❏

**BrowseName** : String

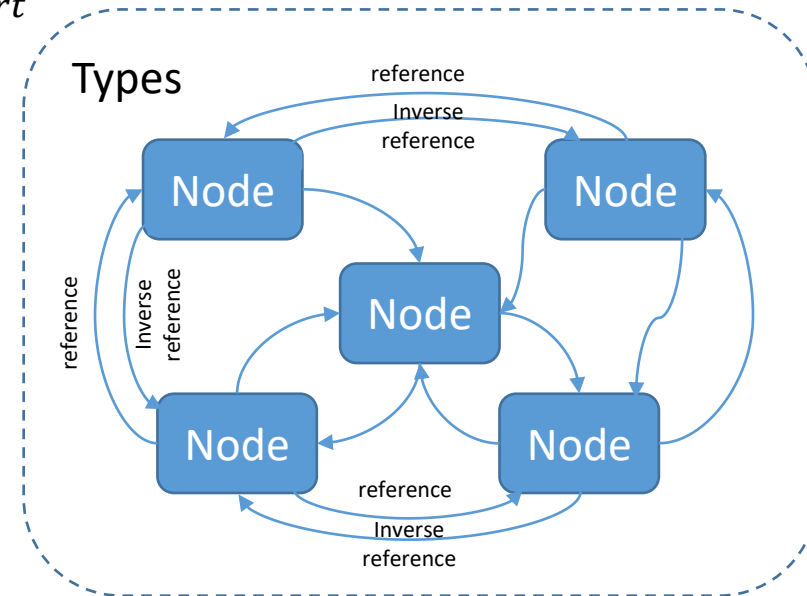
- N'est pas utilisé pour l'affichage
- Il ne change pas au sein d'une branche (path).
- Des services permettent d'utiliser les BrowseName



Espace d'adressage d'un serveur OPC UA =  $\sum_{k=start}^{stop} \text{Types} + \text{Instances}$



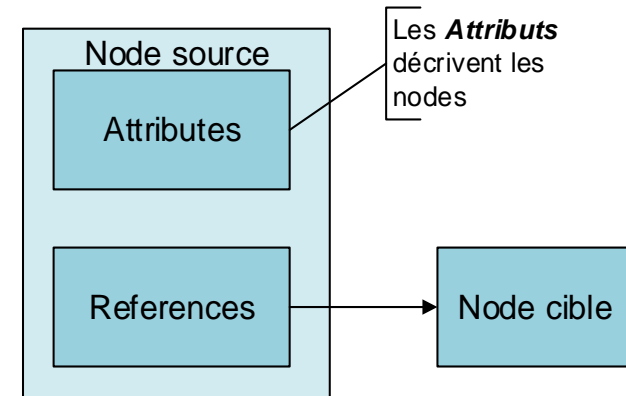
- ☐ Object
- ☐ Variable
- ☐ Methode
- ☐ Reference
- ☐ View



- ☐ ObjectType
- ☐ VariableType
- ☐ ReferenceType
- ☐ DataType

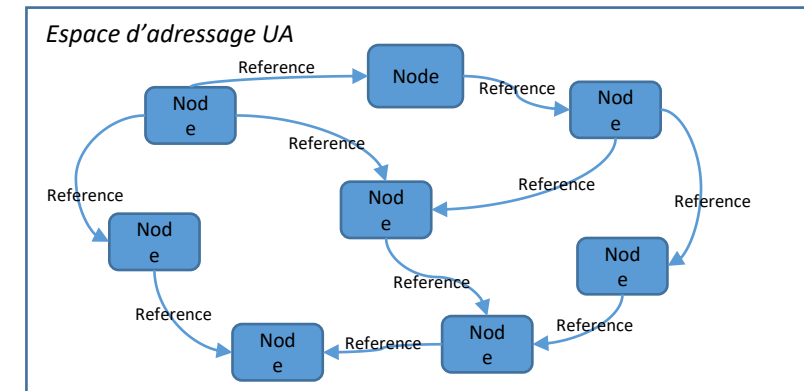
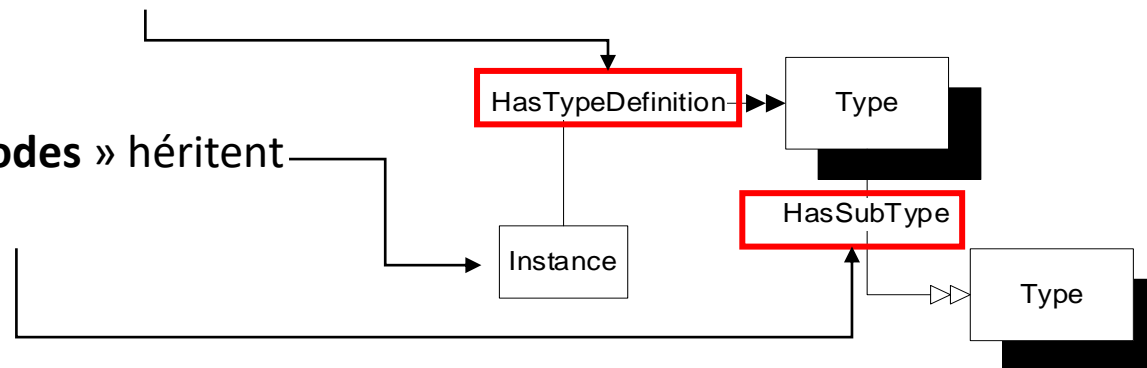
- ❑ La modélisation OPC UA s'appuie sur les concepts orientés objet.
  - Héritage (*simple*)
- ❑ Tous les **Nodes** appartiennent à une classe définie dans le méta-modèle.
- ❑ Les Nodes sont reliés par des références.
- ❑ Les références font partie de l'espace d'adressage
- ❑ Les définitions de type font partie de l'espace d'adressage
- ❑ Les objets de l'espace d'adressage sont mis à disposition des clients

### ❑ Les Nodes



### ❑ Tous les Nodes UA sont typés

❑ Tous les « **Nodes** » héritent d'un ancêtre.



- Les **UAObject** sont des nodes permettant d'organiser l'Espace d'adressage.
- Ils en assurent la description sémantique.

UAObject

- Les **UAVariable** sont des nodes permettant de transporter les valeurs (**OpcData**)

UAVariable

OpcData

OpcData

- Valeur
- TimestampSource
- TimestampServer
- StatusCode

Valeur :

- Type simple
- Tableau ou matrice de type simple
- Enumeration
- Structure
- Tableau ou matrice de structure

## OPC UA PROPOSE UN MODÈLE ORIENTÉ OBJET

- Dans une architecture OPC UA il y a toujours un modèle de données,
- Les instances, objects, variables, views, sont toujours des instances d'un type,
- Même si on ne définit pas de nouveaux types on utilisera des types existant pour créer ses instances,
- Créer de nouveaux types permettra de prendre en compte les spécificités métier.
- Les companions specifications définissent toujours de nouveaux types.

En conception Orienté Objet. un objet représente un concept, une idée ou toute entité du monde physique,

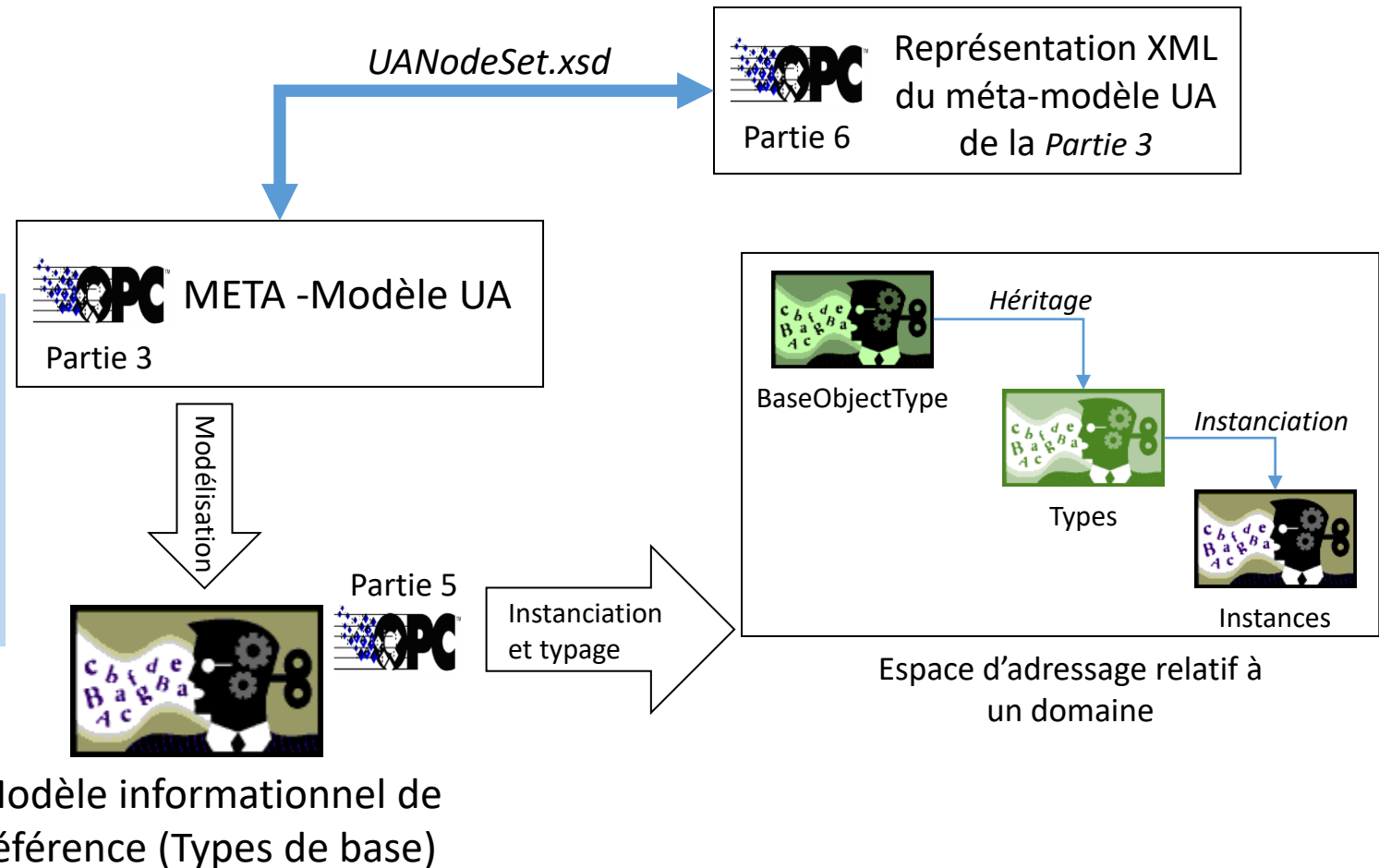
Un objet possède une structure interne et un comportement, et il sait interagir avec ses pairs.

- ❑ OPC UA utilise un **méta-modèle**, comprenez un modèle pour créer des modèles.
- ❑ Il s'agit d'un modèle orienté objet.

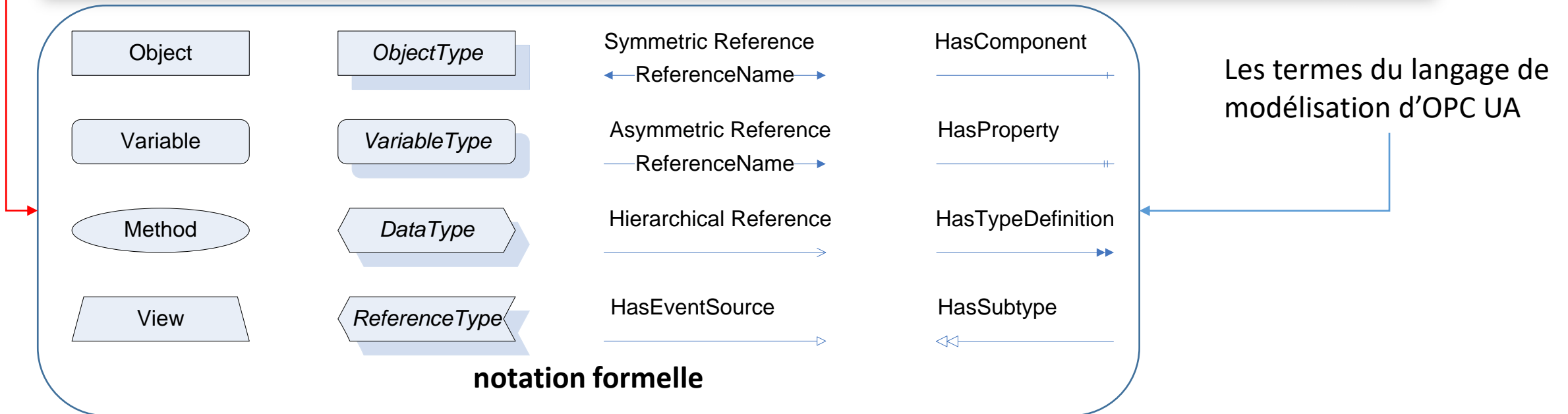
Modéliser consiste à créer des types. Les types sont des nodes. Il existe des :

- ReferenceType
- ObjectType
- VariableType
- DataType

Tous ces types sont définis dans le méta-modèle

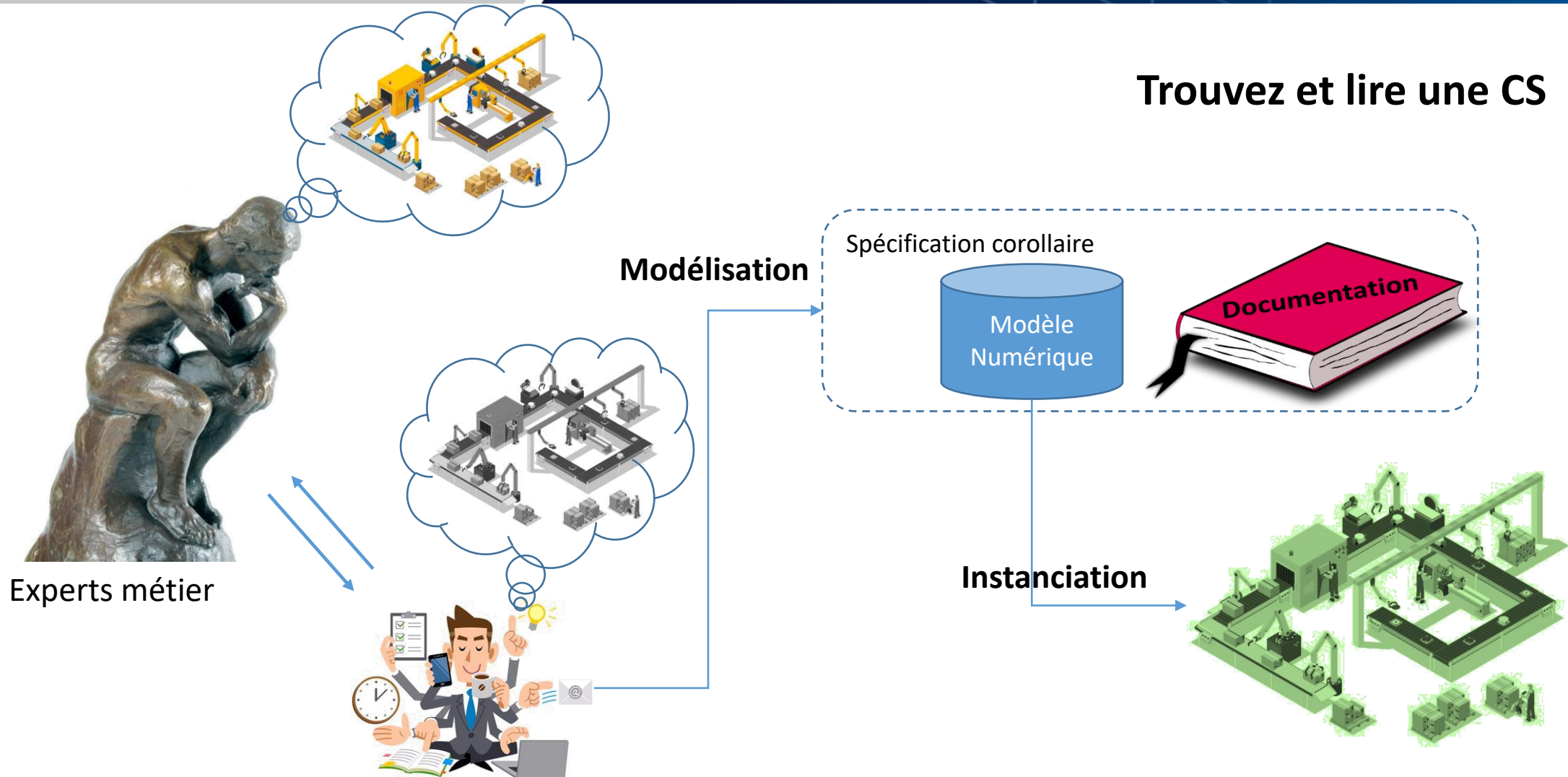


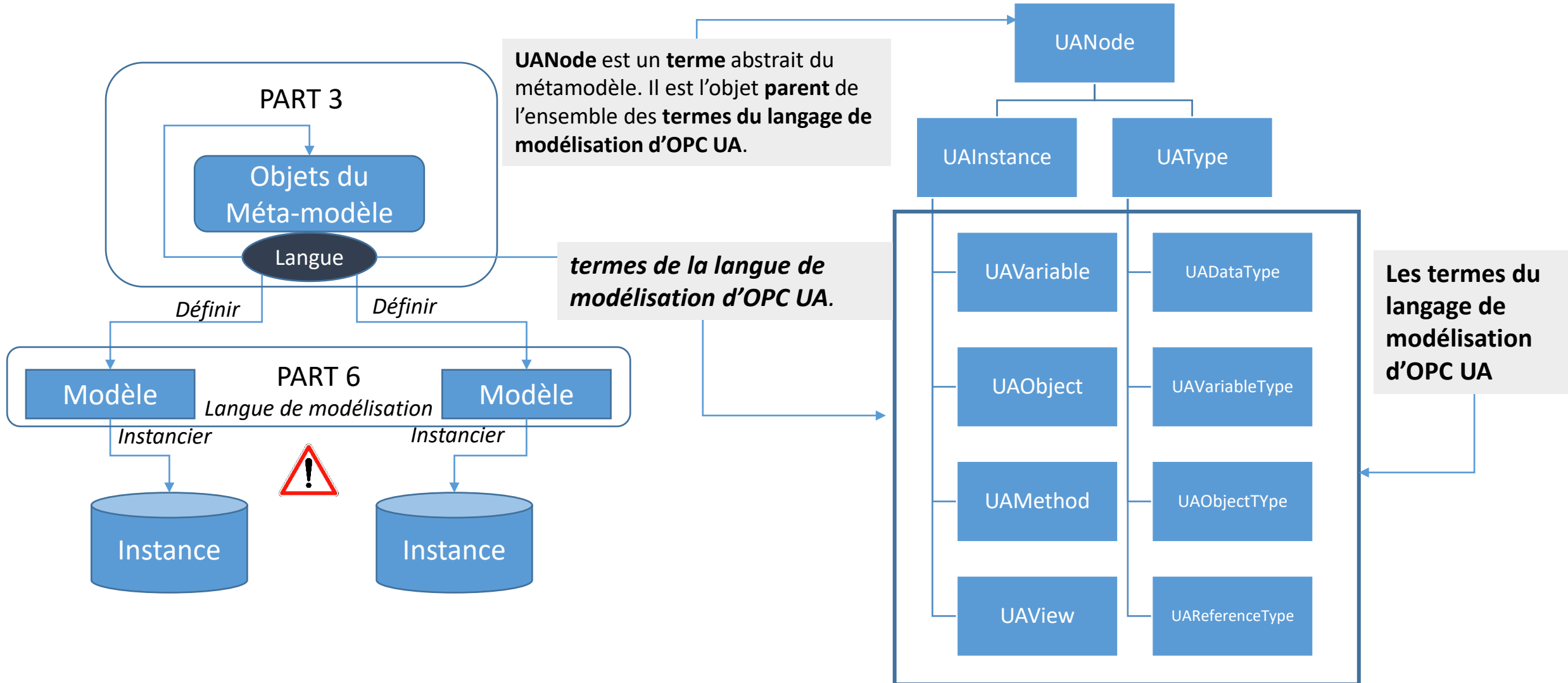
- ☐ L'ensemble des objets exposés par un serveur OPC UA sont regroupés dans un **espace d'adressage UA**.
- ☐ Cet espace d'adressage UA est composé d'**instances** et de **types**.
- ☐ Ces instances et ces types sont construits à partir d'un méta modèle qui est décrit dans la **partie 3**.
- ☐ Tous les objets OPC UA (*instances et types*) sont appelés «**nodes**».
- ☐ Une notation formelle est utilisée pour représenter les nodes OPC UA
- ☐ Cette notation est définie dans la **partie 3** de la spécification
- ☐ La spécification UA et les spécifications complémentaires utilisent cette notation





**Trouvez et lire une CS**







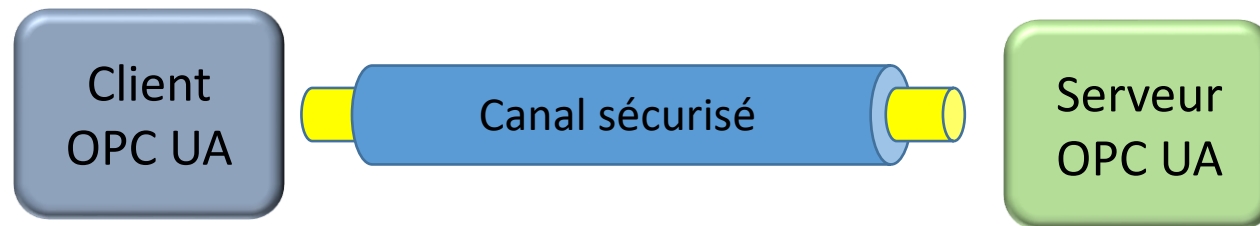
# OPC UA

## La plomberie

Les relations entre applications OPC UA sont toujours client/serveur.  
En MQTT le broker est serveur.  
Subscriber et Publisher sont clients.

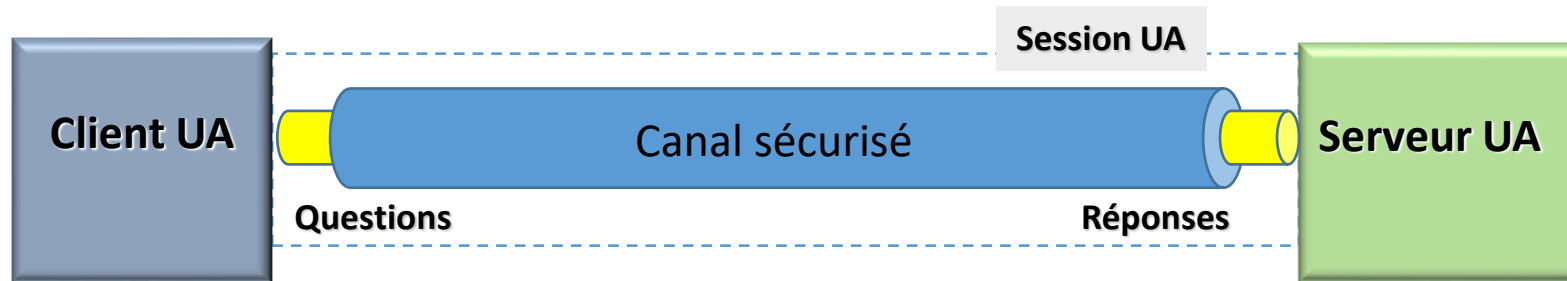
### Le canal sécurisé

- Un canal sécurisé est la première brique de la relation entre applications OPC UA,
- Il est souvent masqué mais toujours présent
- Il est essentiel de le garder à l'esprit pour comprendre d'éventuels problèmes.



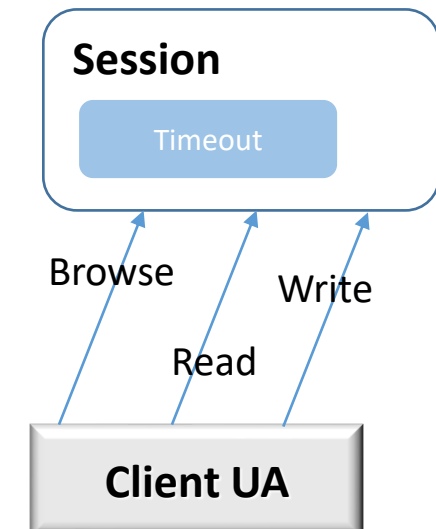
Le canal sécurisé est ouvert à la demande du client et maintenu par la stack du serveur.

### La session



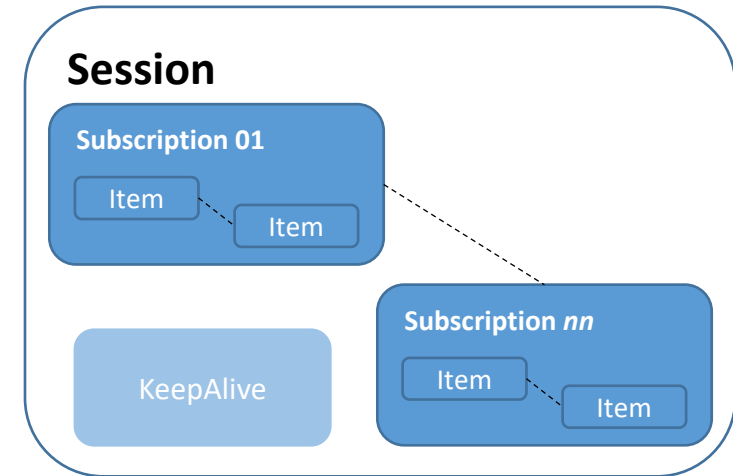
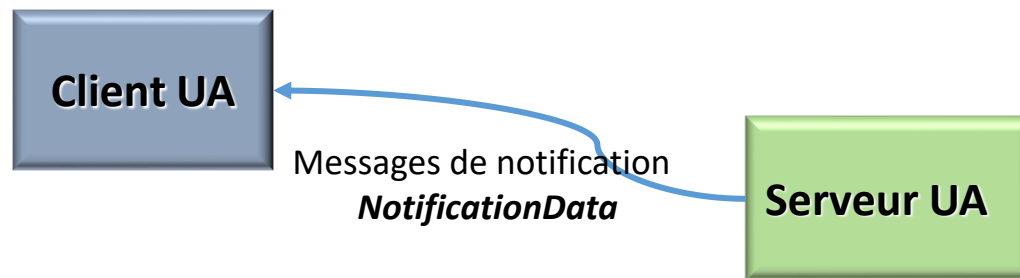
- ☐ Un client peut créer plusieurs sessions.
- ☐ Un timeout est assigné par le client à chaque session.
- ☐ Une fois que la session est créée le client peut :
  - Parcourir l'espace d'adressage du serveur (Browse)
  - Lire les attributs des nœuds (Read)
  - Ecrire les attributs des nœuds (Write)
  - S'abonner à des attributs

On doit toujours privilégier l'abonnement pour accéder aux données

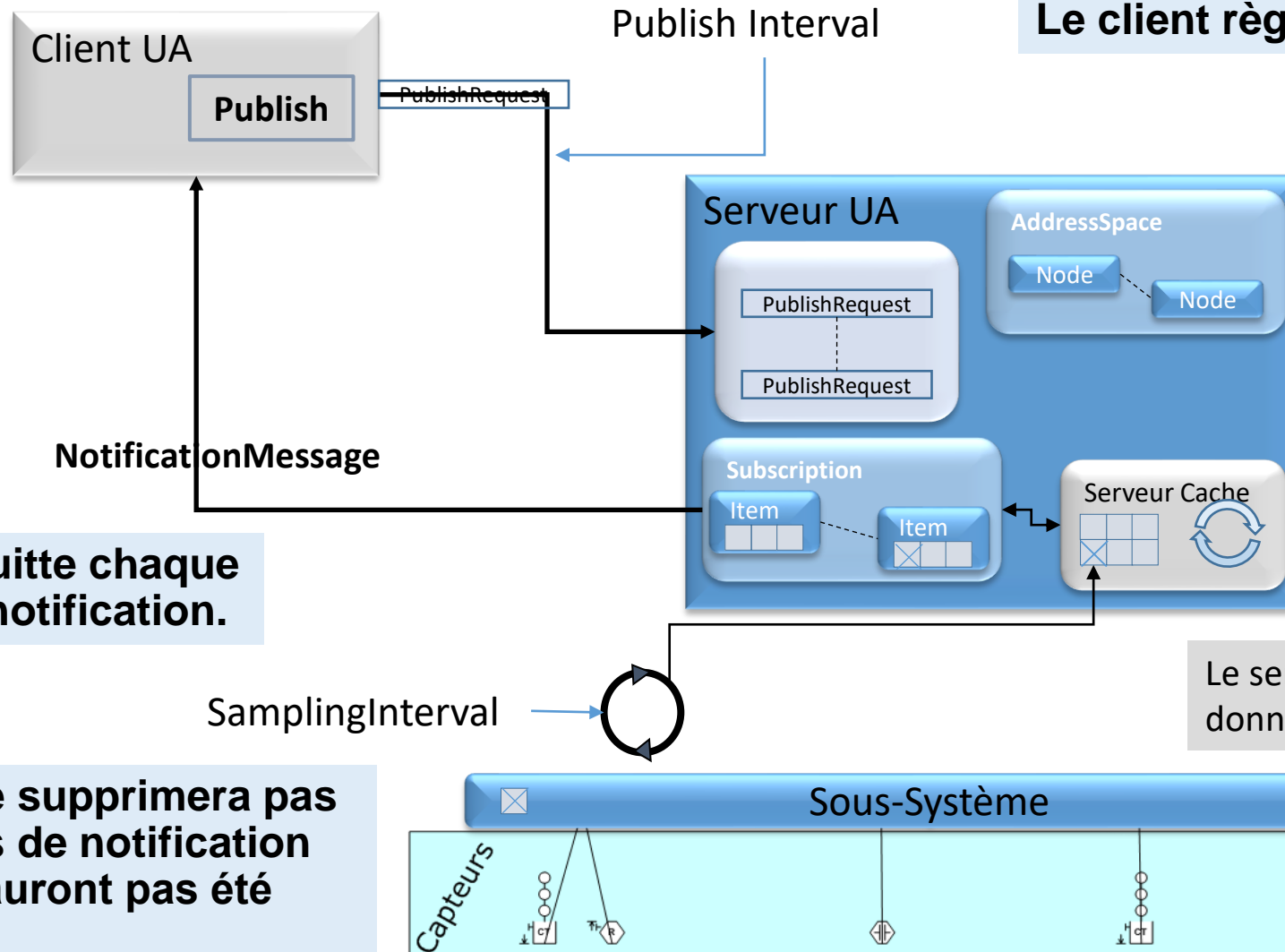


### La souscription

- ❑ Les sessions contiennent un ensemble de **subscription**.
- ❑ Les abonnements contiennent des items **MonitoredItems**.



- ❑ Les MonitoredItems sont transmis au client dans des messages de notification (**NotificationData**).
- ❑ Les messages de notification sont constitués par les **valeurs** et les **alarmes**.
  - EventNotificationList
  - DataChangeNotification
- ❑ Les abonnements sont indépendants de la ligne de vie de la session.



**Le client règle pour chaque item surveillé:**

Mode de surveillance

- Désactivé
- Echantillonné
- Compte rendu

☐ Intervalle d'échantillonnage du sous-système

☐ Filtre(ex. Deadband)

☐ Taille du buffer (**QueueSize**)

☐ Modalités de son remplissage

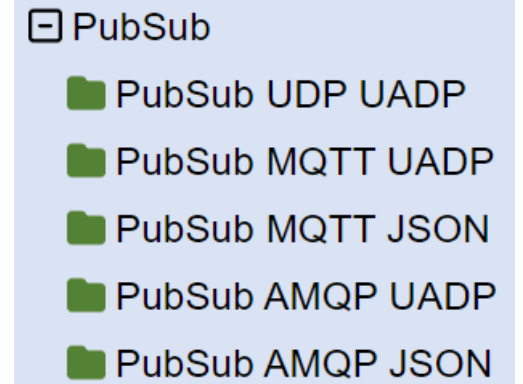
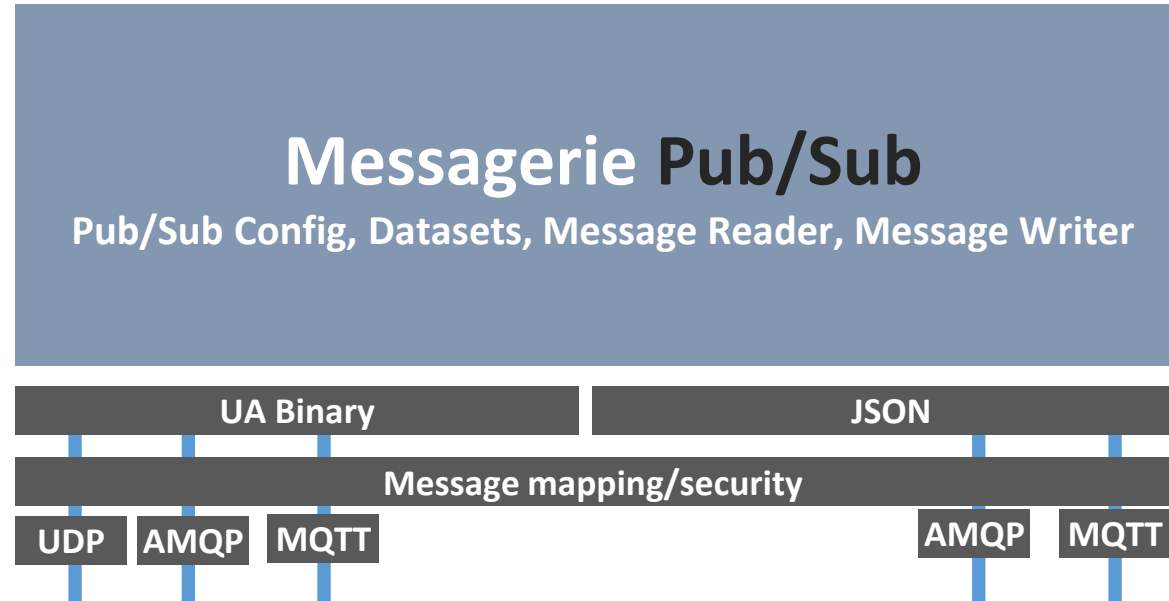
**Le client acquitte chaque message de notification.**

SamplingInterval

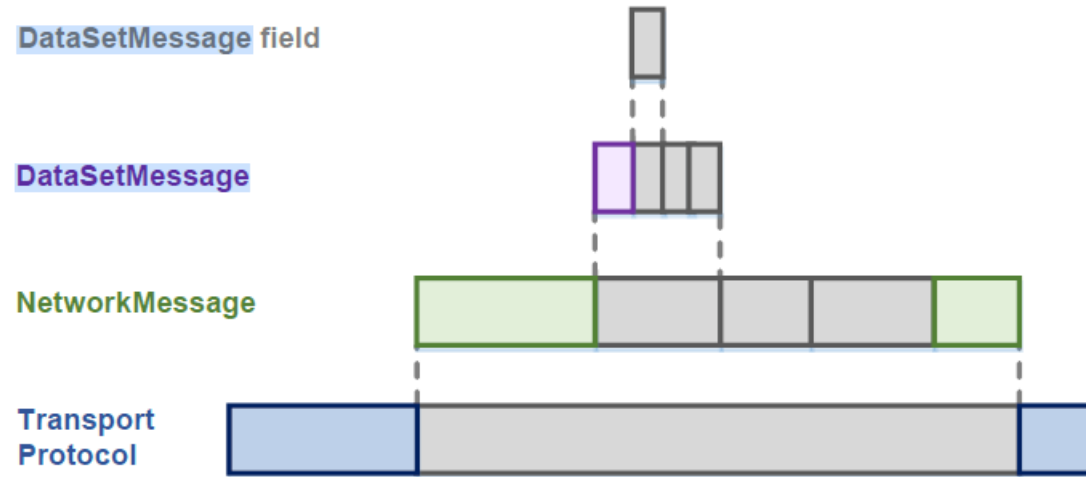
Le serveur met à jour les données de sa cache.

**Le serveur ne supprimera pas les messages de notification tant qu'ils n'auront pas été acquittés.**





- MQTT V3.1.1 et/ou V5
- AMQP 1.0
- En version 1.04 la partie 14 contient 164 pages
- En version 1.05 la partie 14 contient 247 pages
- ***La nouvelle version contient quelques nouvelles fonctionnalités et des clarifications***



- **NetworkMessage**  
Il s'agit du message déposé dans le "Message Oriented Middleware" par le publisher et reçu par le subscriber.
- **DataSetMessages** sont contenus dans les **NetworkMessages**.
- **DataSetMessage**  
Il s'agit de la donnée mise à disposition par le publisher (Charge utile). Un ensemble de mesures issues de capteurs
- **DataSetMessage field** correspond à une donnée au sein d'un DataSetMessage, mesure issue d'un capteur.  
Il existe plusieurs représentations, **DataValue**, **Variant**, **RawData**.

Une application OPC UA est un serveur ou un client OPC UA.

Des SDK sont disponibles dans de nombreux langages pour réaliser ces applications

- C++
- C
- JAVA
- NodeJs
- Go
- C#
- Python



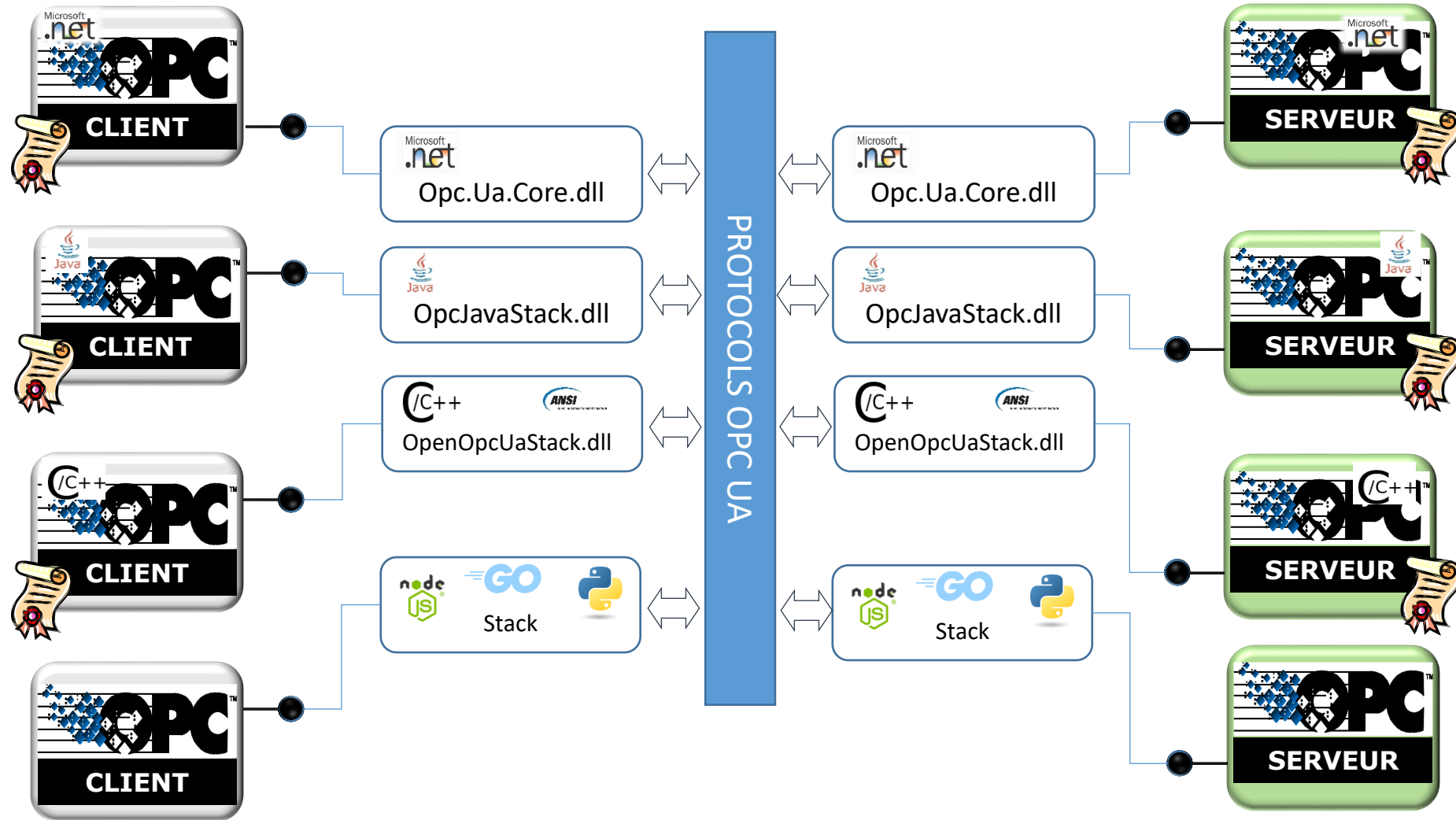
**Tous les SDK n'offre pas :**

- le même niveau fonctionnel,
- Les mêmes performances,
- Le même coefficient mise à l'échelle.



**Distinguer**

- Spécification,
- implémentation



## Applications client OPC UA

Des SDK sont disponibles dans de nombreux langages :



### Distinguer

- Spécification,
- implémentation

Pour chacun d'entre eux il faut considérer plusieurs caractéristiques :

Caractéristique	Coefficient
La couverture fonctionnelle de la spécification	
La flexibilité/disponibilité	
La pérennité	
La facilité d'utilisation	
performances	
L'empreinte	

**Atelier** développement d'applications client OPC UA.  
remplir le tableau pour chaque type de langage

Caractéristique	.Net Coefficient	C/C++ Coefficient	JAVA Coefficient	Go Coefficient	Python Coefficient	NodeJS Coefficient
La couverture fonctionnelle de la spécification						
La flexibilité/disponibilité						
La pérennité						
La facilité d'utilisation						
performances						
L'empreinte						



Client side. Il faut aussi considérer le développement Low/No code.

## Applications serveur OPC UA

Des SDK sont disponibles dans de nombreux langages :



### Distinguer

- Spécification,
- implémentation

Caractéristique	Coefficient
La couverture fonctionnelle de la spécification	
La flexibilité/disponibilité	
La pérennité	
La facilité d'utilisation	
performances	
L'empreinte	





# OPC UA

## Développer des applications

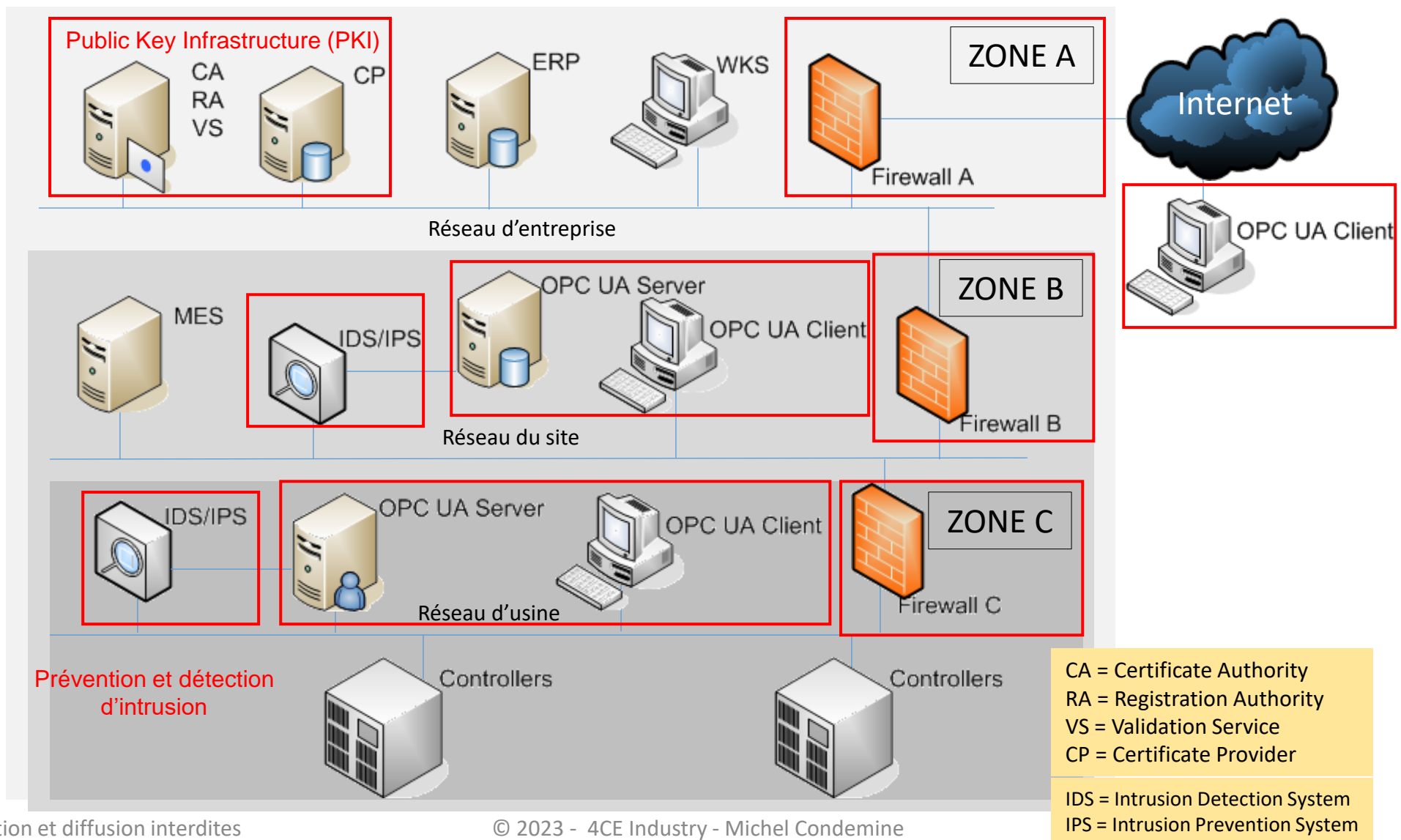
**Atelier** développement d'applications serveur OPC UA.  
remplir le tableau pour chaque type de langage

Caractéristique	.Net Coefficient	C/C++ Coefficient	JAVA Coefficient	Go Coefficient	Python Coefficient	NodeJS Coefficient
La couverture fonctionnelle de la spécification						
La flexibilité/disponibilité						
La pérennité						
La facilité d'utilisation						
performances						
L'empreinte						



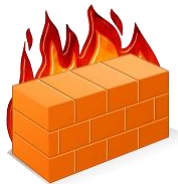
# OPC UA

## La sécurité



## Les objectifs

- ☐ Fiabilité
  - ☐ Communications fiables
  - ☐ Architectures redondantes et hautement disponibilité
- ☐ Sécurité
  - ☐ Protection du canal de communications
    - Canal sécurisé par des certificats X509
  - ☐ Identification des utilisateurs
- ☐ Flexibilité
  - ☐ Permettre la communication de données industrielles sur Internet à travers des firewalls aussi bien que sur les réseaux d'entreprise.

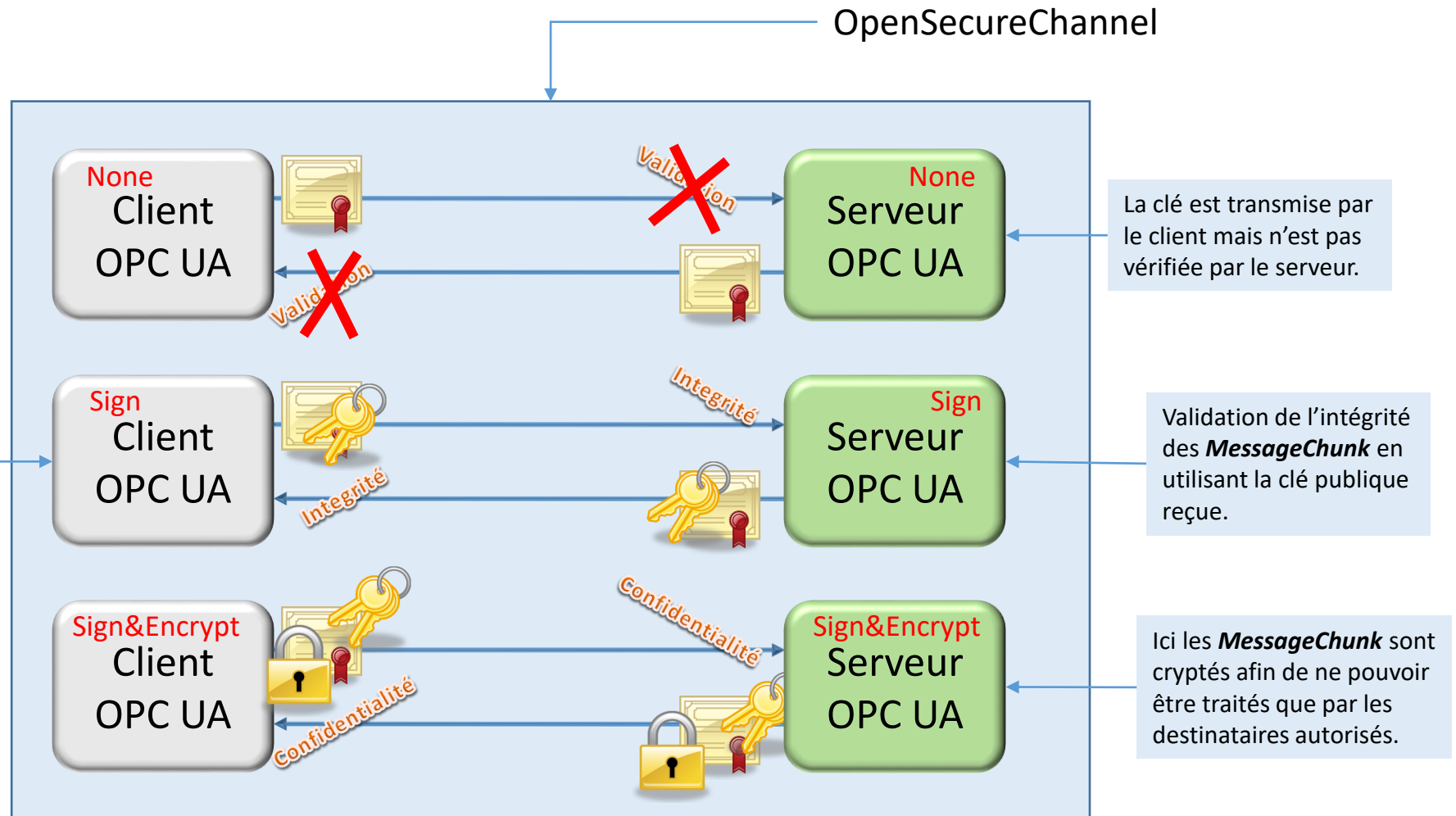


What is  
**Public Key  
Infrastructure ?**

Il existe 3 **modes de sécurité** :

- « Sans sécurité »
- Par signature
- Par signature et chiffrement

Signature de chaque **MessageChunk** avec la clé de l'émetteur.



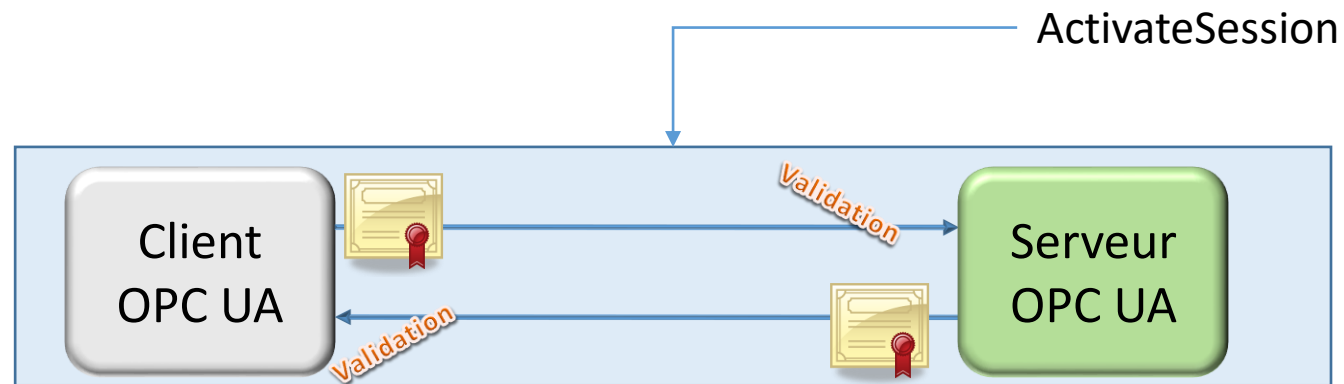
### Politiques de sécurité

- None (*Test uniquement*)
- Basic128RS15 (*Obsolète*)
- Basic256 (*Obsolète*)
- Basic256Sha256
- Aes128-Sha256-RsaOaep
- Aes256-Sha256-RsaPss



Il existe 4 User Identity Tokens :

- Anonymous
- UserName/Password
- X509 Identity
- Issued Identity (WS-Security)



Il y a toujours une relation entre le mode de sécurité, none, sign ou sign&encrypt, sélectionné lors de la l'ouverture du canal sécurisé et l'activation de la session.

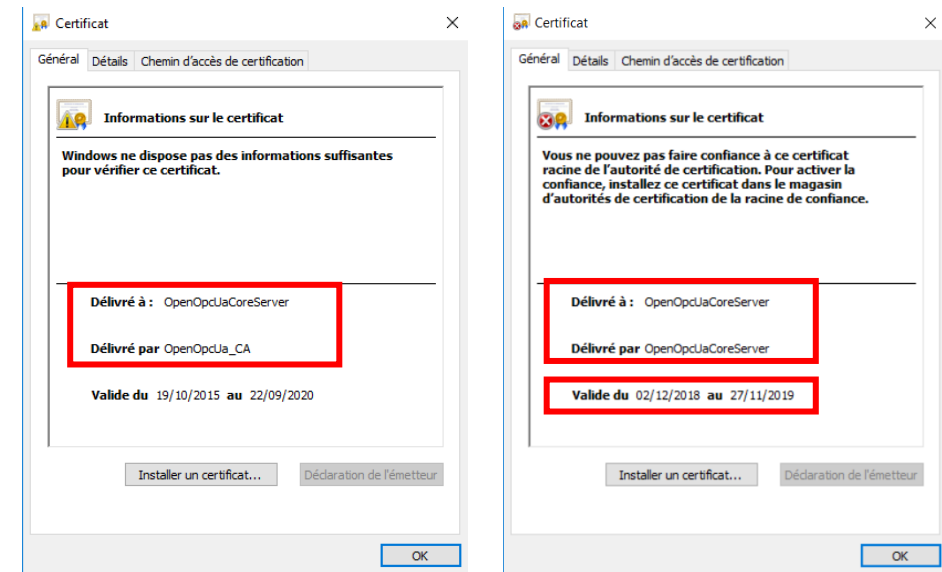
User Identity Tokens	None	Sign	Sign&Encrypt
Anonymous	Aucun chiffrement	Aucun chiffrement	Aucun chiffrement
UserName/Password	Aucun chiffrement	chiffrement de mot du passe	Chiffrement de mot du passe
X509 Identity	Aucun chiffrement	chiffrement du certificat	Chiffrement du certificat
Issued Identity (WS-Security)	Aucun chiffrement	Cryptage en fonction du type de Token WS-Security utilisé	Chiffrement en fonction du type de Token WS-Security utilisé





- ☐ L'utilisateur final choisit sa politique de sécurité et le fournisseur devra s'y conformer
- ☐ Un certificat expirera.
  - Leur durée de vie est de 6 mois à 5 ans.
  - 1 an est une valeur raisonnable.
- ☐ Un certificat peut être auto-signé ou délivré par une autorité tierce.
- ☐ Un certificat est associé à un CRL (**RFC 5280**)
  - La CRL expire à une fréquence élevée (7 jours)
  - Le point de distribution de la CRL (CRLDP) est intégré au certificat
  - La prise en compte de cette CRL est en dehors de la spécification OPC UA

## ATTENTION A LA GESTION DES CERTIFICATS



- Plusieurs solutions permettent de gérer les certificats dans le contexte du déploiement d'une solution OPC UA.
- L'intégrateur ou le fournisseur de solution est souvent soumis à des contraintes externes

Trois scénarii se présente régulièrement :

1. Gestion au travers de l'infrastructure d'entreprise (ie: Active Directory)
2. Gestion au travers d'un GDS
3. Gestion « Manuelle »

La sécurité dans OPC UA PubSub concerne :

- La sécurité du protocole de transport
- La sécurité et la confidentialité des messages

**La sécurité du protocole de transport** dépendra des possibilités offertes par ce protocole. Par exemple en MQTT TLS 1.2 et/ou authentication pourront être utilisés.

Comme en OPC UA TCP **la sécurité des messages** offre trois niveaux :

- Pas de sécurité
- Signature des messages sans chiffrement
- Signature et chiffrement des messages.

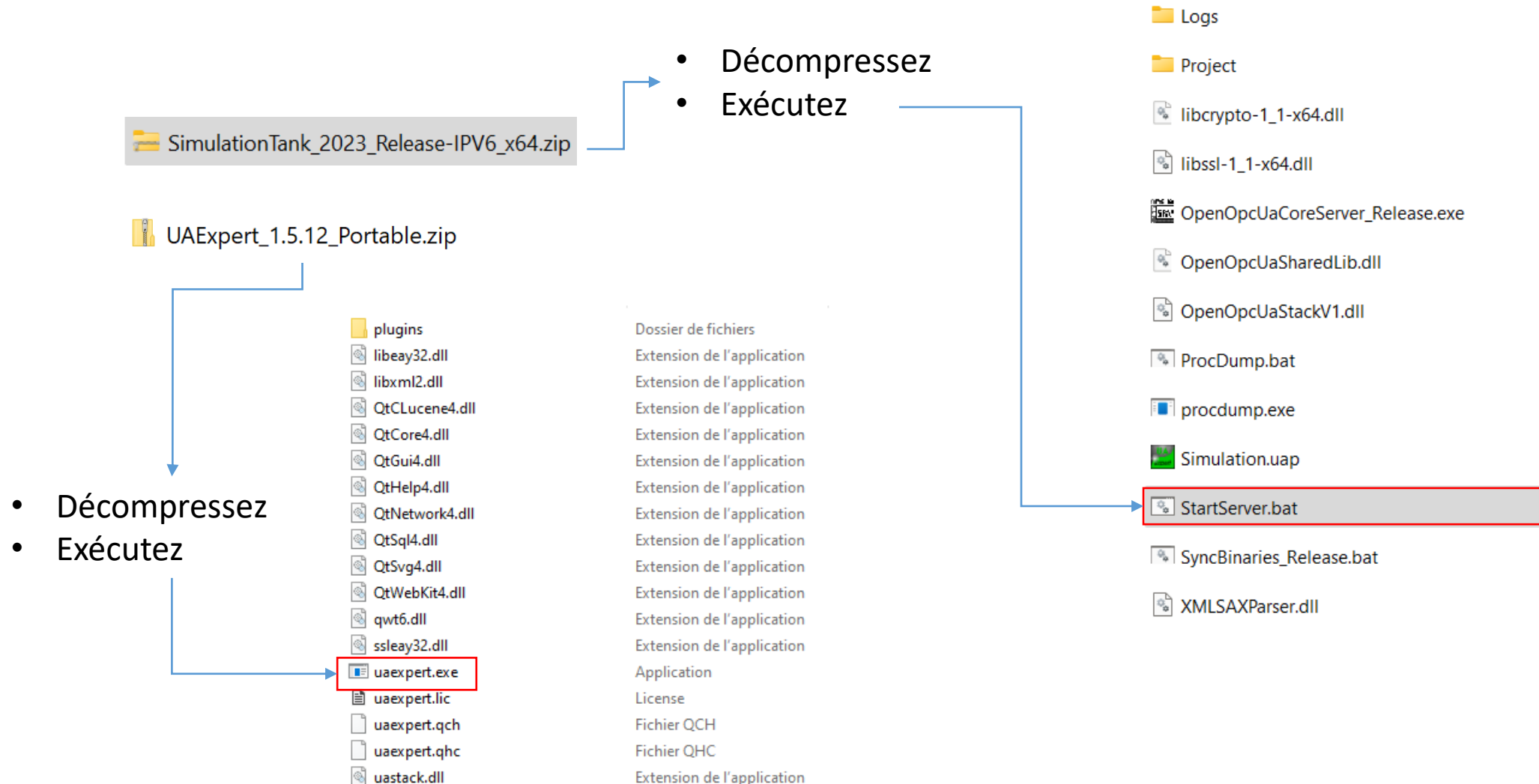


# Atelier



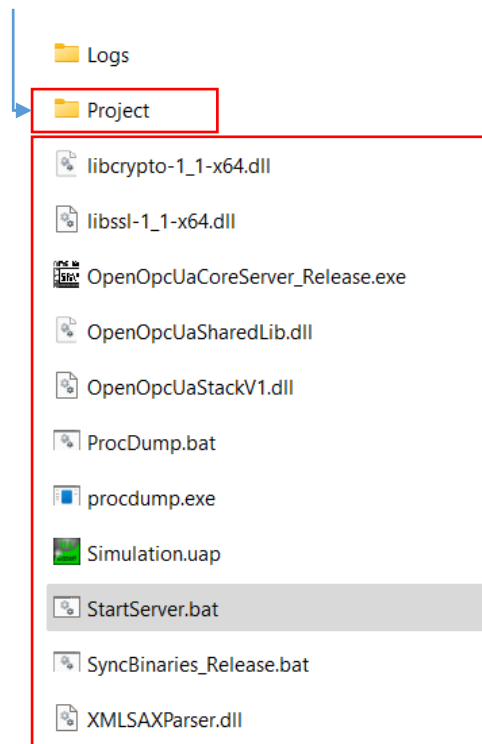
## Atelier 1

Etablir une connexion sécurisée



Le serveur en invite de commande.  
Vous pouvez aussi le lancer en tant  
que service

## Répertoire du projet



Fichiers binaires  
OpenOpcUa (OOUA)

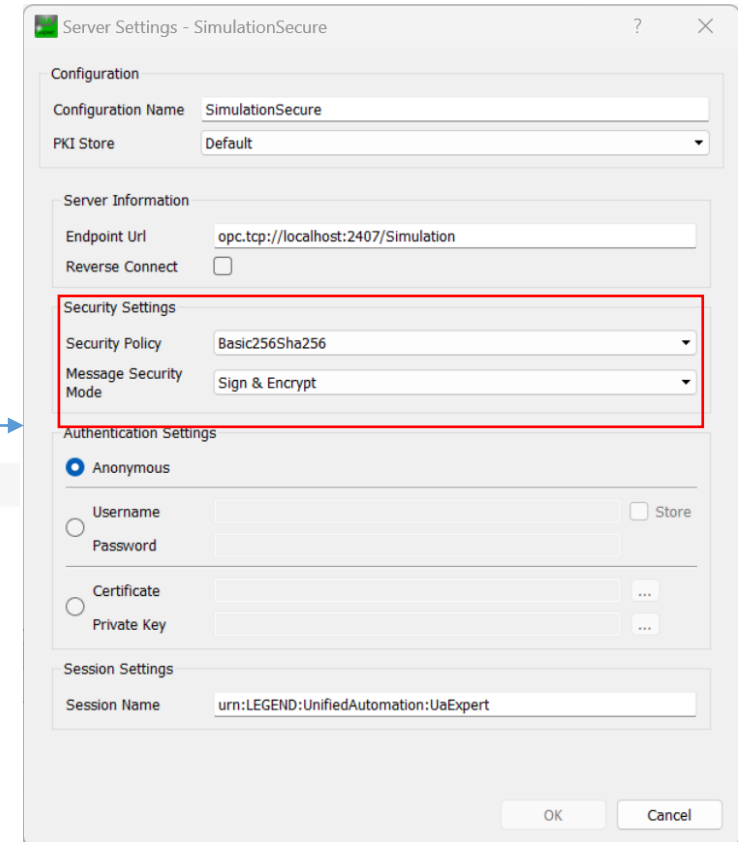
```
C:\WINDOWS\system32\cmd. x + v

C:\Samples\Atelier1>OpenOpcUaCoreServer_Release.exe .\Project\ConfigOpenOpcUa.oooprj
OpenOpcUa:58648> OpcUa_ProxyStub_Initialize: Network Module...
OpenOpcUa:58648> OpcUa_ProxyStub_Initialize: Network Module done!
OpenOpcUa:58648> OpenOpcUaCoreServer 1.1.0.8
OpenOpcUa:58648> Loading .\Project\ConfigOpenOpcUa.oooprj
OpenOpcUa:ALWAYS> .\Project\OpenOpcUaCoreServer.lua was loaded properly
OpenOpcUa:ALWAYS> Server is listening on IPV6 at :
                    opc.tcp://MYTHE:2407/Simulation.
OpenOpcUa:ALWAYS> Press Q or q to exit.
```

Cette demo anime la valeur de nodes disponibles  
dans un espace d'adressage OPC UA

1. lancez le serveur
2. Configurez une connexion sécurisée
3. Tentez une connexion

```
C:\WINDOWS\system32\cmd
C:\Samples\Atelier1>OpenOpcUaCoreServer.Release.exe \Project\ConfigOpenOpcUa.ouaprcj
OpenOpcUa:58648> OpcUa_ProxyStub_Initialize: Network Module...
OpenOpcUa:58648> OpcUa_ProxyStub_Initialize: Network Module done!
OpenOpcUa:58648> OpenOpcUaCoreServer 1.1.0.8
OpenOpcUa:58648> Loading \Project\ConfigOpenOpcUa.ouaprcj
OpenOpcUa:ALWAYS> \Project\OpenOpcUaCoreServer.lua was loaded properly
OpenOpcUa:ALWAYS> Server is listening on IPV6 at :
opc.tcp://MYTHE:2407/Simulation.
OpenOpcUa:ALWAYS> Press Q or q to exit.
```



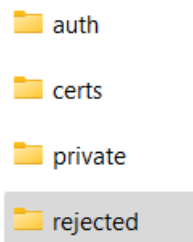
The dialog box 'Server Settings - SimulationSecure' contains the following sections:

- Configuration:** Configuration Name: SimulationSecure, PKI Store: Default.
- Server Information:** Endpoint Url: opc.tcp://localhost:2407/Simulation, Reverse Connect: ☐.
- Security Settings (highlighted with a red box):** Security Policy: Basic256Sha256, Message Security Mode: Sign & Encrypt.
- Authentication Settings:** Anonymous: ☒, Username:  (Store: ☐) Password: , Certificate: ☐ Private Key: ☐ (both with browse buttons).
- Session Settings:** Session Name: urn:LEGEND:UnifiedAutomation:UaExpert.

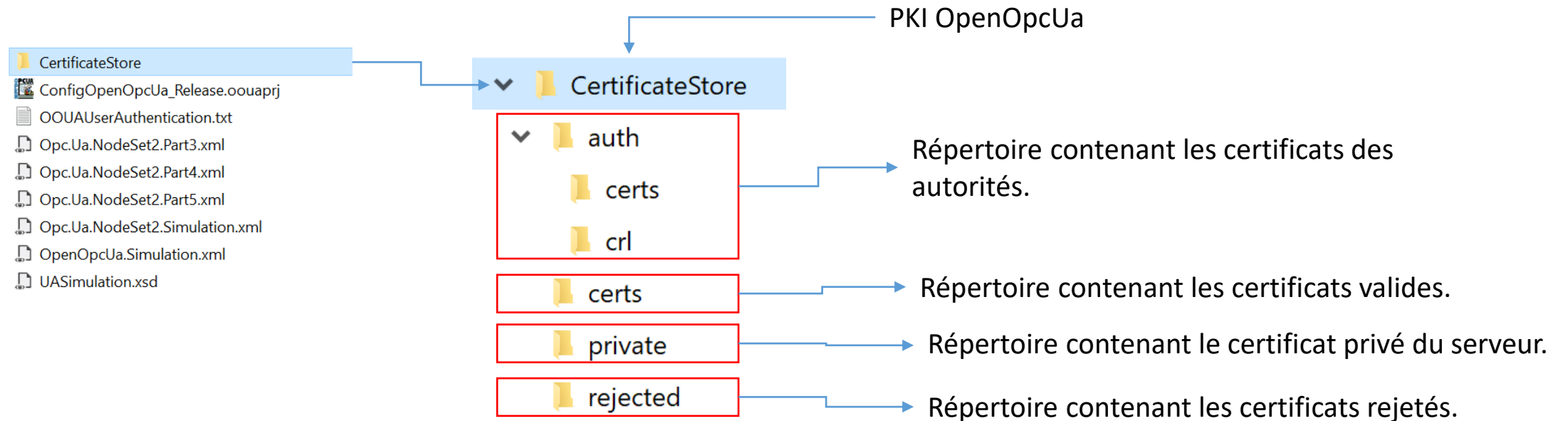
Buttons: OK, Cancel.

SimulationSecure Error 'BadSecurityChecksFailed' was returned during OpenSecureChannel

4. Administrez la PKI du serveur







Pour autoriser la connexion :

- Accédez au répertoire rejected
- Déplacez le fichier du client vers le répertoire certs.

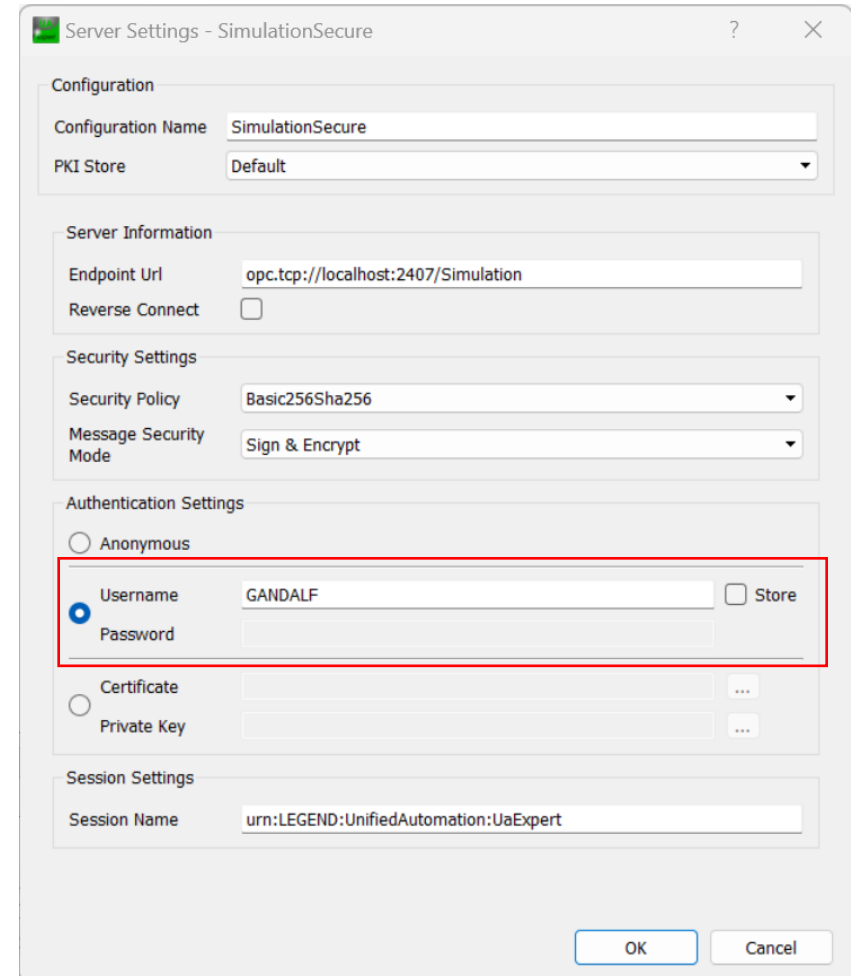
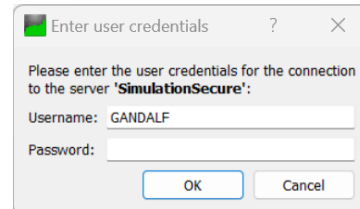
**Bonus** : lancez wireshark et observez :

- une connexion chiffrée
- Un connexion non chiffrée

### Sécurité des utilisateurs

```
<Binding Protocol="TCP" Port="2407" Encoding="Binary">
  <Security Policy="None" Mode="None"/>
  <Security Policy="Basic128Rsa15" Mode="Sign"/>
  <Security Policy="Basic128Rsa15" Mode="SignEncrypt"/>
  <Security Policy="Basic256" Mode="Sign"/>
  <Security Policy="Basic256" Mode="SignEncrypt"/>
  <Security Policy="Basic256Sha256" Mode="Sign"/>
  <Security Policy="Basic256Sha256" Mode="SignEncrypt"/>
  <UserToken Type="Anonymous"/>
  <UserToken Type="UserName"/>
  <UserToken Type="X509"/>
</Binding>
```

1. Lancez le serveur
2. Ajoutez à la connexion sécurisée l'identification des utilisateurs
3. Tentez une connexion





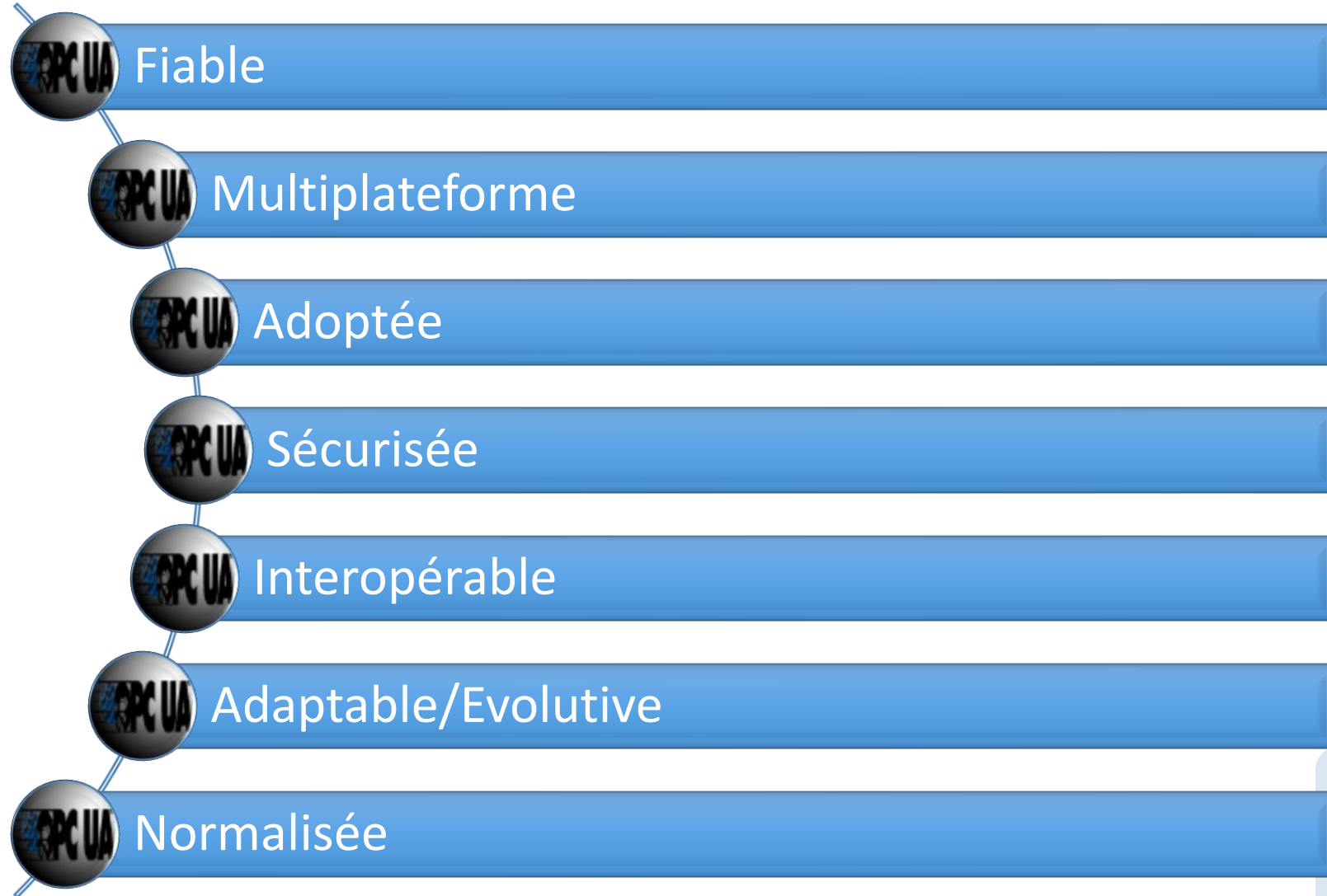
# Atelier 1

## Connexion sécurisée

### REFLEXION SUR LA GESTION DE LA SECURITE

- Sécurité de canal de communication
- Sécurité des utilisateurs
- Politique de sécurité
- Gestion de la PKI

# Les 7 caractéristiques essentielles d'OPC UA



# Questions ?



**Michel Condemine**

- OPC Foundation France (Technique)
- Directeur de **4CE Industry**
- Leader du projet OpenOpcUa
- MichelC@4CE-Industry.com



3<sup>ème</sup> édition prévue  
pour décembre 2023