



**Université Abdelhamid Mehri – Constantine 2**  
**Faculté des Nouvelles Technologies de l'Information et de la Communication**  
**Département d'Informatique Fondamentale et ses Applications**

1

# Artificial Intelligence of Things (AIoT)

**2<sup>EME</sup> ANNÉE MASTER**  
**SDIA**  
**S1**

**DR ILHAM KITOUNI**

**24-25**

# Chapitre 3- Architecture et Conception<sup>2</sup> des systèmes AIoT

[AIoT-syllabus-ang24-25.docx](#)

## References

- González García, C., Núñez Valdéz, E. R., García Díaz, V., Pelayo García-Bustelo, B.C., & Cueva Lovelle, J. M. (2019). A review of artificial intelligence in the internet of things. International Journal Of Interactive Multimedia And Artificial Intelligence, 5.
- Ghosh, A., Chakraborty, D., & Law, A. (2018). Artificial intelligence in Internet of things. CAAI Transactions on Intelligence Technology, 3(4), 208-218. Andrew S. Tanenbaum, Maarten van Steen. Distributed Systems: Principles and Paradigms. 3rd edition. Pearson Education, 2017.

# 1. Introduction à l'Architecture et à la Conception des Systèmes AIoT

3

## • **Architecture AIoT :**

- Structure globale comprenant les composants matériels (capteurs, passerelles, actuateurs) et logiciels (algorithmes d'IA, plateformes IoT).
- Organisation des composants pour assurer un flux de données optimal et des décisions intelligentes.

## • **Conception AIoT :**

- Planification et création des éléments matériels et logiciels.
- Prise en compte de la performance, sécurité, scalabilité, et expérience utilisateur.

## 2. Importance de la Conception dans les Projets AIoT

4

- **Un système bien conçu :**

- **Scalable** : Capable de gérer un grand nombre de dispositifs et de volumes massifs de données.
- **Flexible** : Adaptable aux différents besoins opérationnels.
- **Sécurisé** : Intégration de mesures de sécurité dès la conception.
- **Performant** : Réactivité essentielle pour les applications critiques en temps réel (ex : véhicules autonomes).
- **Exemple** : Gestion de capteurs dans les **villes intelligentes** pour surveiller l'éclairage public, le trafic, etc.

### 3. Différences entre l'IoT Classique et l'AIoT

5

- **IoT Classique :**

- **Collecte des données** uniquement.
- **Traitement des données au cloud**, souvent supervisé par des humains.

- **AIoT :**

- **Analyse et prise de décision** en temps réel avec des algorithmes d'IA.
- **Traitement en périphérie (Edge AI)** pour des réponses instantanées (ex : véhicules autonomes).

## 4. Interaction et Maintenance dans l'AIoT vs IoT Classique

6

### • **Interaction Homme-Machine :**

- **IoT Classique** : Supervision humaine nécessaire pour l'analyse.
- **AIoT** : Prise de décision autonome (ex : ajustement automatique des machines dans une usine intelligente).

### • **Maintenance :**

- **IoT Classique** : Maintenance réactive (surveillance, réaction aux pannes).
- **AIoT** : Maintenance prédictive avec IA (prédiction des pannes avant qu'elles ne surviennent).



## 5. Exemples Concrets d'AIoT

7

### **1.Santé Connectée :**

- 1.IoT classique : Surveillance des signes vitaux.
- 2.AIoT : Analyse en temps réel et alertes automatiques aux médecins.

### **2.Agriculture Intelligente :**

- 1.IoT classique : Mesure de l'humidité du sol.
- 2.AIoT : Analyse des données pour l'irrigation optimale selon l'IA.

### **3.Industrie 4.0 :**

- 1.IoT classique : Surveillance des machines.
- 2.AIoT : Maintenance prédictive avec apprentissage automatique pour prévenir les pannes.

## 6. Composants principaux de l'Architecture AIoT

8

### 1- Capteurs et Actuateurs

- **Capteurs** : Dispositifs qui collectent des données sur des variables physiques (température, humidité, pression).
  - **Exemple** : Ferme intelligente – Les capteurs mesurent l'humidité du sol, la température et la luminosité pour optimiser l'irrigation via une analyse IA.
- **Actuateurs** : Dispositifs qui réagissent aux données des capteurs en effectuant des actions physiques (ouvrir une vanne, ajuster un thermostat).
  - **Exemple** : Dans un bâtiment intelligent, un actuateur ajuste la climatisation en fonction des données de température collectées.



## 6. Composants principaux de l'Architecture AIoT

### 2 - Passerelles IoT (IoT Gateways)<sup>9</sup>

#### • Fonctions principales :

- **Agrégation de données** : Collecte et envoi des données des capteurs vers un serveur ou système Edge.
- **Sécurité** : Chiffrement des données et authentification des dispositifs.
- **Protocoles de communication** : Traduction entre différents protocoles (MQTT, HTTP, CoAP).
- **Exemple** : Dans une usine intelligente, une passerelle IoT collecte les données de capteurs sur la température et la consommation d'énergie, puis les transmet à un système de calcul pour prévenir les pannes.

## 6. Composants principaux de l'Architecture AIoT

10

### 3 - Edge Computing et Cloud Computing

#### • Edge Computing :

- Traitement des données à la **périphérie** (près des capteurs) pour des décisions en temps réel.
- **Avantages** : Réduction de la latence et optimisation de la bande passante.
- **Exemple** : Un véhicule autonome prend des décisions instantanées (éviter un obstacle) grâce au traitement local des données des capteurs.

#### • Cloud Computing :

- Analyse de grandes quantités de données dans le **cloud**, idéal pour l'entraînement des modèles d'IA.
- **Exemple** : Les vidéos de surveillance sont analysées dans le cloud pour détecter des comportements suspects.

## 6. Composants principaux de l'Architecture AIoT

11

### 4 - Communication et Réseaux

#### • **Protocoles couramment utilisés :**

- **MQTT** : Protocole léger pour la communication entre capteurs et systèmes centraux.
- **CoAP** : Protocole adapté aux dispositifs à faibles ressources.
- **LoRaWAN** : Protocole à longue portée pour des dispositifs IoT sur de vastes zones.

• **Exemple** : Dans une ville intelligente, le réseau LoRaWAN connecte des capteurs pour surveiller la pollution de l'air et gérer l'éclairage public.

## 6. Composants principaux de l'Architecture AIoT

12

### 5 - Plateformes AIoT

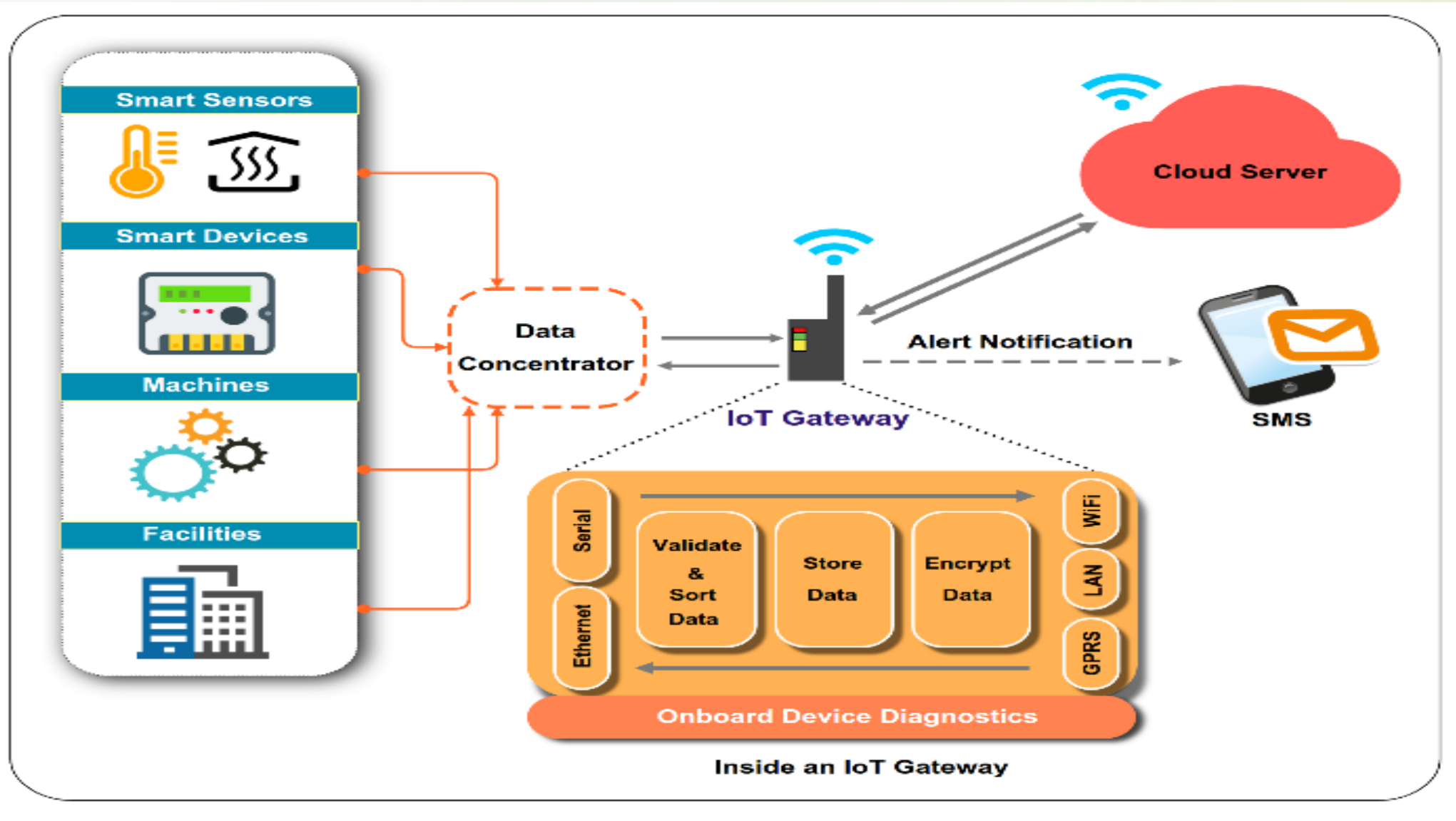
#### • Fonctions des plateformes AIoT :

- **Gestion des dispositifs** : Enregistrement et gestion des capteurs.
- **Analyse des données** : Utilisation d'algorithmes d'IA pour traiter les données en temps réel.
- **Automatisation** : Activation automatique des dispositifs ou envoi d'alertes selon les modèles d'IA.

#### • Exemples de plateformes AIoT :

- **AWS IoT, Google Cloud IoT, Azure IoT** : Ces plateformes gèrent les dispositifs et intègrent l'IA pour l'automatisation des actions et la prise de décision en temps réel.

## 6. Composants principaux de l'Architecture AIIoT





## 7. Conception des systèmes AIoT

### 1- Introduction à la Conception des Systèmes AIoT

- **Objectif** : Combiner la collecte de données IoT et l'intégration de l'IA.

- **Principales étapes** :

- Analyse des besoins.
- Choix des composants matériels et logiciels.
- Conception des flux de données.
- Plan de sécurité et de confidentialité.
- Conception d'interface utilisateur (UI/UX).
- Tests et validation.

- **But** : Optimiser les performances, la sécurité, et la flexibilité.



## 7. Conception des systèmes AIoT

15

### 2 - Analyse des Besoins

- **Identification des besoins métier :**

- Réduction des coûts, amélioration de la productivité.
- **Exemple :** Utilisation des capteurs IoT pour surveiller l'état des machines dans une usine et prévoir les pannes.

- **Exigences fonctionnelles et non fonctionnelles :**

- Fonctionnelles : collecte et analyse en temps réel.
- Non fonctionnelles : scalabilité, sécurité, fiabilité.

- **Exemple :** Réseau IoT dans une ville intelligente, doit être scalable et sécurisé. Exemple ....

## 7. Conception des systèmes AIoT

### 3 - Choix des Composants Matériels et Logiciels

- **Sélection des capteurs et actuateurs :**

- Capteurs adaptés aux objectifs (température, humidité, vibrations).
- **Exemple :** Capteurs mesurant l'humidité du sol dans une serre intelligente.

- **Plateformes et frameworks AI/IoT :**

- Choix de plateformes IoT (AWS IoT, Google Cloud IoT) et frameworks IA (TensorFlow, PyTorch).
- **Exemple :** Google Cloud IoT pour les données de circulation, TensorFlow pour la prédiction du trafic.

## 7. Conception des systèmes AIoT

17

### 4 - Conception des Flux de Données et Sécurité

- **Gestion des flux de données :**

- Collecte locale (Edge Computing) vs traitement dans le cloud.
- **Exemple :** Chaîne de production utilisant Edge pour des réponses rapides et cloud pour les analyses à long terme.

- **Plan de sécurité et confidentialité :**

- Sécurité by design : chiffrement des données, authentification.
- **Exemple :** Capteurs en hôpital intelligent, protection des données sensibles.

## 7. Conception des systèmes AIoT

18

### 5 : Conception d'Interfaces et Tests

- **Conception des interfaces utilisateurs (UI/UX) :**
  - Visualisation intuitive des données IoT et IA via des tableaux de bord interactifs.
  - **Exemple :** Tableau de bord pour une ferme intelligente affichant les données d'humidité en temps réel.
- **Tests et validation :**
  - Tests de performances des capteurs, modèles IA, et sécurité.
  - **Exemple :** Tests de sécurité contre des attaques DDoS dans une infrastructure critique.

## 8. Modèles de Conception pour les Systèmes AIoT

19

### 1- Introduction aux Modèles de Conception pour les Systèmes AIoT

- modèles d'architecture pour concevoir des systèmes AIoT:
  - Modèle centralisé vs distribué.
  - Conception basée sur des microservices.
  - Conception orientée événement.
- **Facteurs clés** : Latence, modularité, évolutivité, réactivité en temps réel.

## 8. Modèles de Conception pour les Systèmes AIoT

20

### 2 - Modèle Centralisé vs Modèle Distribué (Edge Computing)

#### • **Modèle centralisé :**

- **Fonctionnement :** Les capteurs collectent des données envoyées au **cloud** pour traitement et analyse.
- **Avantages :** Puissance de calcul élevée et centralisation des données.
- **Inconvénients :** Latence et dépendance à la connectivité.
- **Exemple :** Smart city – Centralisation des données sur la gestion de l'énergie ou du trafic.



## 8. Modèles de Conception pour les Systèmes AIoT

21

### 2 - Modèle Centralisé vs Modèle Distribué (Edge Computing)

- **Modèle distribué (Edge Computing) :**

- **Fonctionnement :** Traitement des données à la périphérie (Edge) pour des décisions locales rapides.
- **Avantages :** Faible latence, réduction de la bande passante.
- **Inconvénients :** Limites de calcul local et gestion plus complexe.

- **Exemple :** Usine intelligente – Les capteurs surveillent les machines et les anomalies sont traitées localement.

## 8. Modèles de Conception pour les Systèmes AIoT

### 3 - Conception Basée sur des Microservices <sup>22</sup>

#### • Fonctionnement :

- Le système est divisé en **microservices** indépendants (collecte de données, IA, stockage).
- **Communication via API** entre les services.

#### • Avantages :

- **Évolutivité** : Les services peuvent être mis à l'échelle individuellement.
- **Modularité** : Maintenance facilitée, mise à jour et remplacement des composants sans impacter l'ensemble.
- **Flexibilité** : Utilisation de différents langages/technologies pour chaque service.

#### • Inconvénients :

- **Complexité de gestion** des interconnexions et API.
- **Latence accrue** si les microservices sont distribués géographiquement.

• **Exemple** : Gestion des transports publics – Un microservice pour les données de positionnement, un autre pour l'analyse, un autre pour l'interface utilisateur.

# 8. Modèles de Conception pour les Systèmes AIoT

## 4 - Conception Orientée Événement (Event-Driven Architecture)

23

### •Fonctionnement :

- Surveillance d'événements spécifiques par les capteurs IoT.
- Réaction immédiate aux événements via l'activation d'un actuateur ou l'envoi d'une alerte.
- Utilisation de systèmes de messagerie comme **MQTT** pour une communication rapide.

### •Avantages :

- **Réactivité en temps réel** : Idéal pour les systèmes nécessitant des réponses immédiates.
- **Automatisation complète** : Moins d'intervention humaine requise.

### •Inconvénients :

- **Complexité de gestion** des flux d'événements à grande échelle.
- **Dépendance aux systèmes de messagerie** pour la gestion des événements.

### •Exemples :

- **Smart city** : Détection de la pollution par des capteurs déclenchant des actions automatiques.
- **Industrie 4.0** : Capteurs détectant des vibrations anormales, déclenchant l'arrêt automatique des machines.

## 8. Modèles de Conception pour les Systèmes AIoT

24

### 5 - Comparaison des Modèles de Conception

Critère	Modèle Centralisé	Modèle Distribué (Edge)	Microservices	Conception Orientée Événement
Latence	Élevée (temps d'envoi au cloud)	Faible (décisions locales)	Variable	Très faible (réaction immédiate)
Scalabilité	Moyenne	Difficile à gérer	Très élevée	Complexe à grande échelle
Modularité	Faible	Complexe	Très élevée	Moyenne
Réactivité	Modérée	Très élevée	Variable	Très élevée

## 8. Modèles de Conception pour les Systèmes AIoT

25

### Choisir le Bon Modèle

- **Modèle centralisé** pour des traitements intensifs nécessitant une puissance de calcul élevée.
- **Modèle distribué** pour des applications critiques nécessitant une faible latence.
- **Microservices** pour des systèmes modulaires et évolutifs.
- **Conception orientée événement** pour des systèmes réactifs en temps réel avec automatisation.
- **Facteurs à considérer** : Latence, complexité, évolutivité, et réactivité.