

# AZURE IOT PLATFORME

# AZURE IOT PLATEFORME

Les architectures des plates-formes IoT basées sur le cloud sont principalement déterminée par les exigences fonctionnelles des applications IoT.

Les principales exigences fonctionnelles sont :

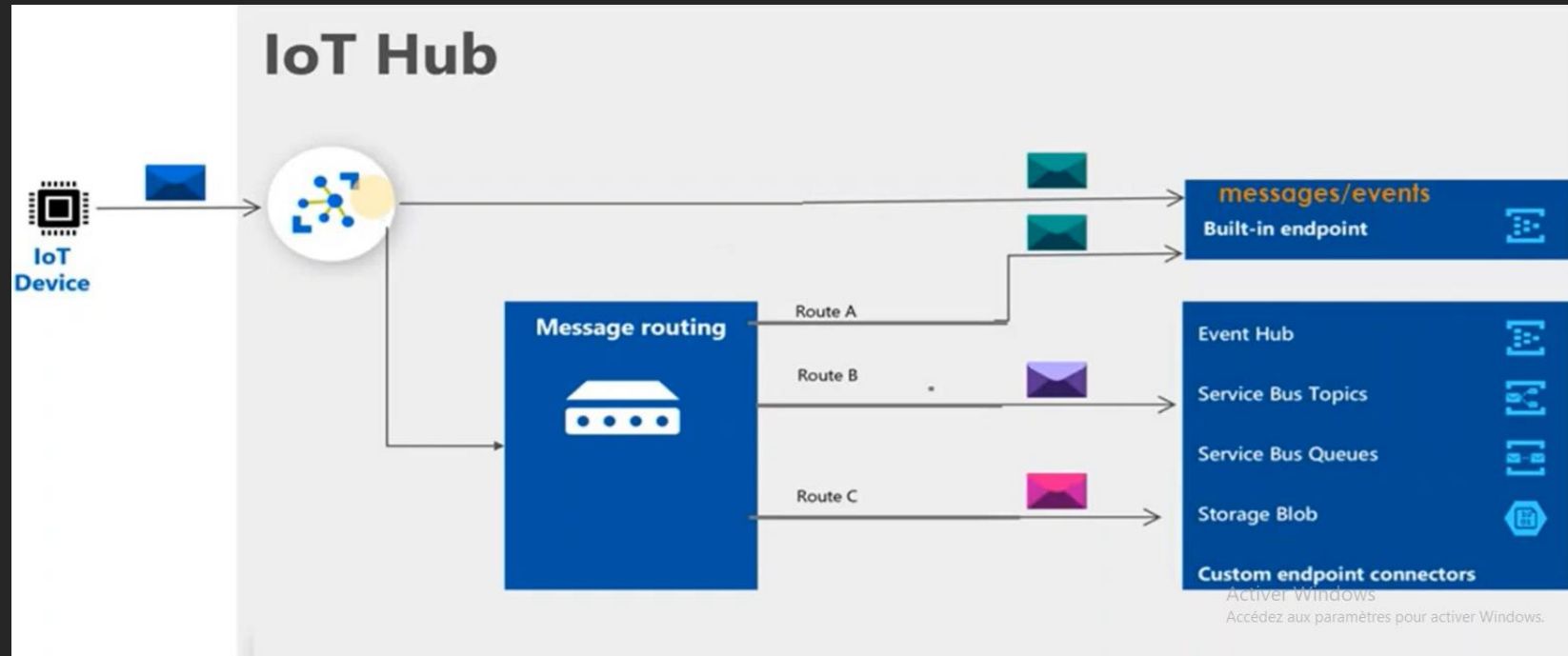
- Enregistrer des objets (devices)
- Collecter, analyser et transmettre des informations collectées à partir de capteurs IoT,
- Stocker les informations collectées,
- Rapporter les résultats de l'analyse aux utilisateurs
- Intégrer les résultats de l'analyse aux systèmes commerciaux .

Les principales exigences non fonctionnelles sont :

- Exigences en sécurité et temps de réponse du système

# AZURE IOT PLATEFORME

- La plateforme Azure IoT fait référence à une suite d'outils logiciels proposés sur la plateforme cloud Azure qui peuvent être utilisés en **combinaison** pour créer des applications IoT.



- IoT SDK, Azure IoT Hub, Stream Analytics et Power BI sont des **applications IoT conçues**,
- Stockage Blob, Cosmos DB et Azure Functions sont destinés aux **applications à usage général** mais fréquemment utilisés dans les applications IoT.

# AZURE IOT PLATEFORME

- La principale fonctionnalité d'une application IoT est de permettre aux objets d'envoyer des messages de télémétrie à une passerelle cloud (cloud Gateway).
- Un protocole de messagerie pour la communication entre les appareils IoT et les passerelles cloud.
- Un client IoT pour les appareils IoT afin qu'ils puissent établir une communication sécurisée avec une passerelle cloud spécifique
- Une application IoT doit également conserver le stockage et assurer le contrôle des appareils

# AZURE IOT PLATEFORME : Protocols de communication

- **MQTT** (Message Queue Telemetry Transport) est un protocole de messagerie client-serveur léger .
  - Faible empreinte mémoire sur l'appareil
  - Des messages compacts qui utilisent moins de bande passante réseau Ainsi
- **AMQP** (Advanced Message Queuing Protocol)
  - Convient aux appareils nécessitant des connexions de longue durée et une communication à haut débit avec Gateway .
  - AMQP permet une topologie d'appareil plus complexe afin que plusieurs appareils puissent se connecter à la passerelle cloud en utilisant la même connexion sécurisée.
- Les appareils IoT peuvent également communiquer directement avec Azure IoT Hub via **HTTP**

# AZURE IOT PLATEFORME : Protocols de communication

MQTT is chosen over HTTP in IoT solutions for several reasons:

- 1.Efficiency:** MQTT is more efficient in terms of bandwidth and network usage compared to HTTP. MQTT uses a publish-subscribe messaging pattern where messages are only sent when there is new data to transmit, reducing the overhead of constant polling or request-response cycles seen in HTTP.
- 2.Real-time Communication:** MQTT is designed for real-time or near-real-time communication. It allows devices to receive updates and messages as soon as they are available, making it suitable for applications where timely data delivery is crucial, such as monitoring systems or industrial automation.
- 3.Reliability in Unstable Networks:** MQTT is designed to handle intermittent or unstable network connections gracefully. It supports features like persistent sessions and Quality of Service (QoS) levels, which ensure that messages are reliably delivered even if the network connection is unreliable or temporarily lost. HTTP, on the other hand, may struggle in such scenarios, especially with its stateless nature and lack of built-in mechanisms for handling network disruptions.
- 4.Scalability:** MQTT is highly scalable and can efficiently handle a large number of devices and clients. Its lightweight nature makes it suitable for IoT deployments with potentially thousands or even millions of devices, as it puts minimal strain on network resources.
- 5.Asynchronous Communication:** MQTT supports asynchronous communication, allowing devices to publish and subscribe to topics independently without waiting for a response from the server. This asynchronous nature can improve the overall responsiveness and performance of IoT systems compared to the synchronous request-response model of HTTP.

# AZURE IOT PLATEFORME : Protocols de communication

# AZURE IOT PLATEFORME : Device client

- **Device Client** Il existe une variété d'appareils IoT avec différents processeurs, systèmes d'exploitation, capacités de mise en réseau et prise en charge d'exécution des langages de programmation. Par exemple,.
  - Pour les appareils qui exécutent un système d'exploitation, les clients IoT peuvent être implémentés à l'aide d'**Azure IoT SDK** à l'aide de divers langages de programmation tels que Java, C, .NET, Python. et JavaScript (Node.js).
  - Pour les appareils basés sur **FPGA** tels que CompactRIO, les clients IoT peuvent être développés à l'aide de langages spécifiques au fournisseur tels que **LabView**.

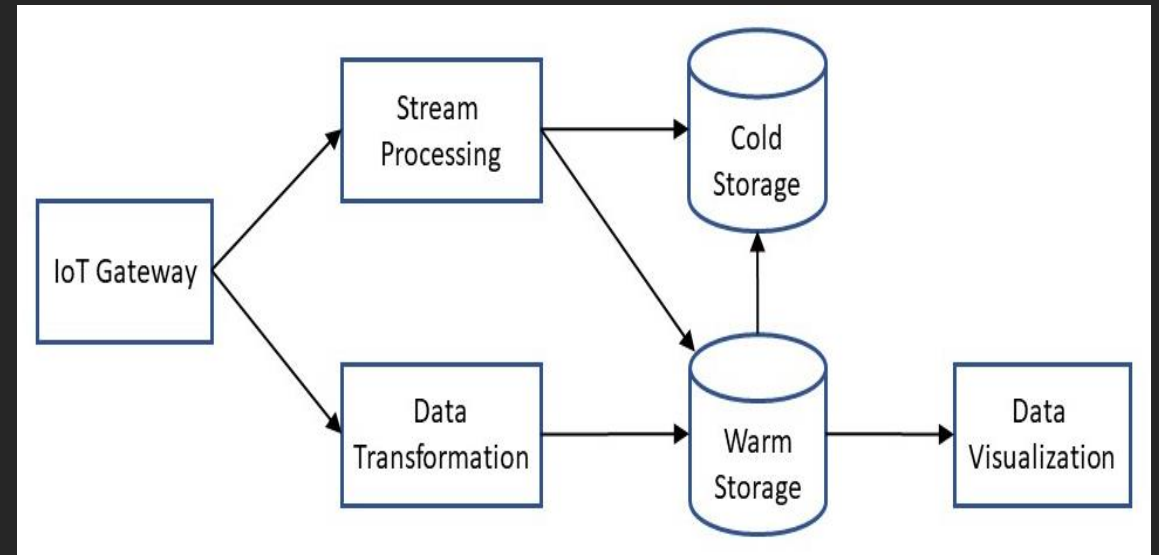


# AZURE IOT PLATEFORME : Cloud gateway

- **Azure IoT Hub** : est plus adapté aux applications qui nécessitent une messagerie bidirectionnelle entre les appareils et les passerelles cloud
  - MQTT, AMQP, et HTTPs sont les protocoles de communications utilisés dans l'IoT hub
- **Azure Event Hubs** : est plus adapté aux applications qui n'ont besoin d'envoyer qu'un grand nombre de messages
  - AMQP en tant que protocole de communication
  - Azure IoT Hub est développé à la base de Event Hub
- Deux type de credentials pour la connexion : 1-IoT Hub 2- Event Hub

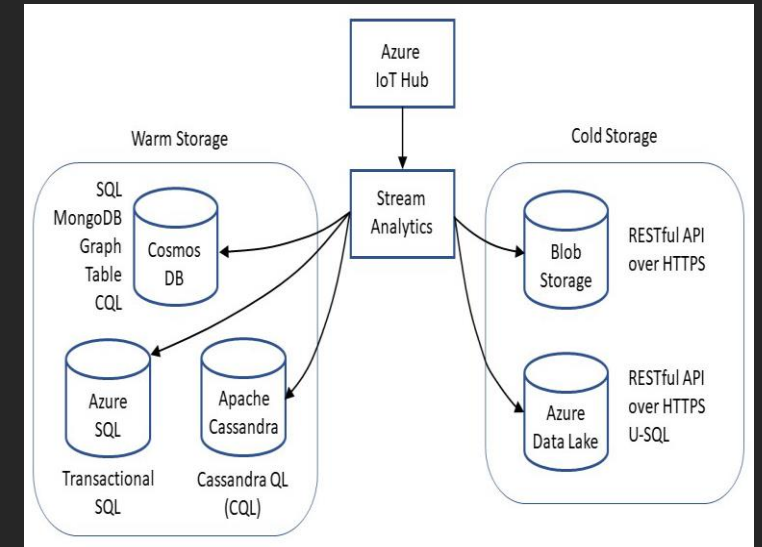
# AZURE IOT PLATEFORME : Storage

- **Warm Storage** : Le stockage à chaud tel que Cosmos DB et Azure SQL Database offre une latence plus faible et une interface de requête flexible, mais à un coût plus élevé
- **Cold Storage** : tel que le stockage Blob et le stockage Data Lake ont une latence plus élevée et une interface plus simple, mais à moindre coût



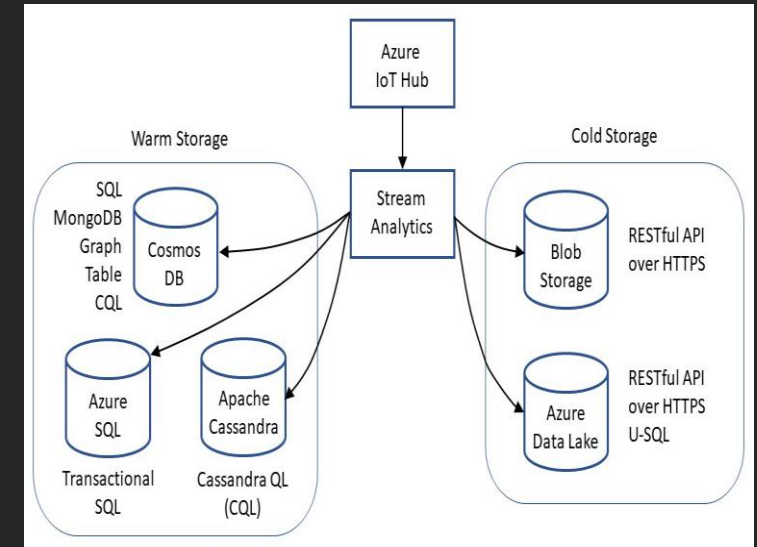
# AZURE IOT PLATEFORME : Storage

- **Worm Storage** : une faible latence est requise.
  - **Cosmos DB** : qui est une base de données NoSQL
    - qui prend en charge cinq méthodes de requête, notamment SQL traditionnel, MongoDB, graph, table et Cassandra Query Language (CQL).
    - Les données dans Cosmos DB peuvent être définies avec une date d'expiration, après laquelle les données sont automatiquement supprimées
  - **SQL Database** : qui fournit un modèle de base de données relationnelle avec prise en charge **transactionnelle**, ce qui peut être nécessaire pour les applications avec de fortes contraintes d'intégrité.
    - Les "applications à **forte contrainte d'intégrité**" se réfèrent à des systèmes où la **fiabilité**, la **cohérence** et la **précision** des données sont d'une importance critique.



# AZURE IOT PLATEFORME : Storage

- **Cold Storage** : un faible coût est requis.
  - le stockage Blob et Azure Data Lake Storage, qui stocke les données sous forme de fichiers ou d'objets et utilise des API RESTful pour l'accès aux données
    - stocke les données sous forme de fichiers ou d'objets
- Pour **un calcul tel que la visualisation de la consommation d'énergie quotidienne** d'un congélateur, le stockage à chaud doit être utilisé car le calcul doit émettre des requêtes qui peuvent renvoyer des résultats précis avec une faible latence.
- Pour des calculs tels **que l'analyse de la consommation électrique moyenne de tous les congélateurs au cours de l'année écoulée**, le stockage à froid peut être utilisé car l'analyse peut être calculée hors ligne avec un ensemble de données agrégé

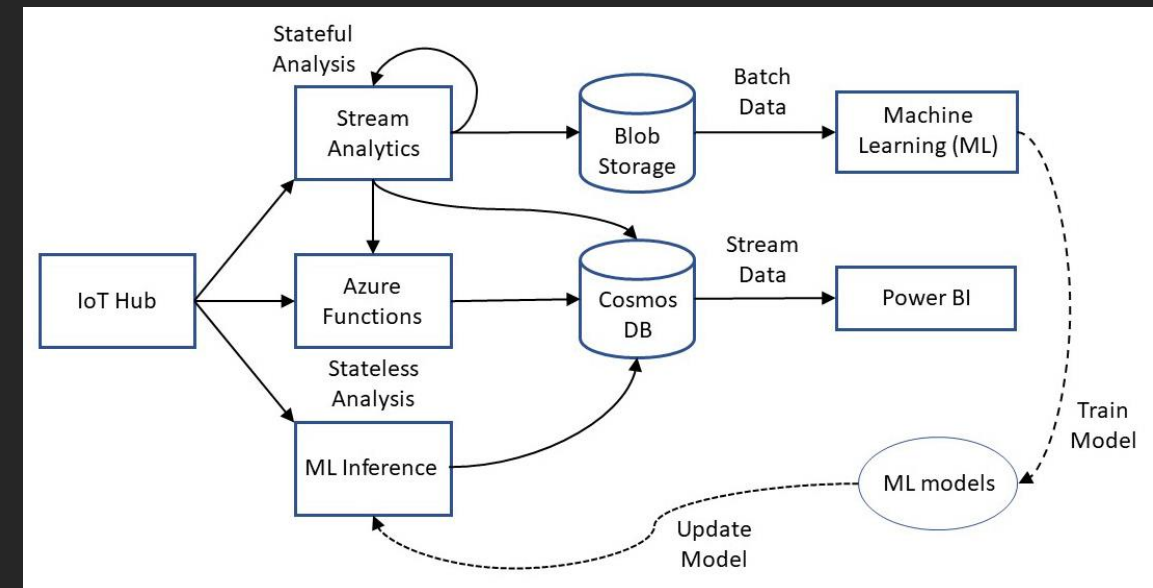


# AZURE IOT PLATEFORME : Analyse des données

L'objectif principal des applications IoT est d'apprendre des informations à partir des données émises par les appareils IoT et d'utiliser ces informations pour prendre des décisions logiques vers des objectifs commerciaux tels que la réduction des coûts d'exploitation des véhicules de la flotte ou la maintenance préventive des équipements industriels. L'analyse des données est l'étape cruciale pour atteindre cet objectif.

Deux types de traitements

- **Traitement en ligne**, le calcul peut également être divisé en sans état ou avec état. Pour le traitement en ligne des données IoT, la latence est une considération importante,
- **Traitement par lot**: la criticité temps n'est pas importantes



# AZURE IOT PLATEFORME : Visualisation

- **Power BI** est un outil de tableau de bord polyvalent pour la visualisation des données et la génération de rapports de données statiques ou à basse fréquence.
  - À l'aide de l'interface graphique de Power BI, les non-programmeurs peuvent créer un tableau de bord avec divers graphiques de données à partir d'une gamme de sources de données.
  - Power BI Desktop et Power BI Service (en ligne).
  - Power BI Desktop a plus de fonctionnalités en termes de visualisation et de génération de rapports.

# AZURE IOT PLATEFORME : Visualisation

- **Applications Web et mobiles Azure** App Service fournit un environnement pour la création d'applications Web, de backend mobile et d'API RESTful.
- Les utilisateurs peuvent créer une interface Web à l'aide d'Azure App Service pour se connecter à des composants IoT tels qu'Azure IoT Hub via WebSocket.
- Azure fournit des hubs de notification , qui permettent aux applications de transmettre des messages aux appareils mobiles de plates-formes telles que iOS et Android.

# Azure vs AWS vs GCP

- Selon Harvey et al., les trois principaux fournisseurs de cloud classés du premier au troisième sont Amazon Web Services (**AWS**), Microsoft **Azure** et Google Cloud Platform (**GCP**).
- La maturité d'AWS, son ingéniosité continue, sa large distribution de centres de données et son vaste ensemble d'outils sont les raisons pour lesquelles les développeurs et les entreprises continuent d'en faire la plate-forme cloud numéro un.
- Selon une enquête auprès des développeurs IoT de la Fondation Eclipse, **34 %** des personnes interrogées utilisent AWS, **23 %** utilisent Azure et **20 %** utilisent GCP.
- À mesure que chaque plate-forme se dispute une plus grande part de marché, les services fournis deviennent de plus en plus similaires. Les similitudes et les différences sont claires car le reste de ce chapitre comparera AWS, Azure et GCP en fonction des services liés à l'IoT.



# Azure vs AWS vs GCP : IoT Related Services

IoT Related Services					
Cloud Platform	Gateway (Bi-Directional)	Gateway (Ingestion Only)	Message Format	Edge Computing	Device SDKs
Amazon AWS	AWS IoT Core	Kinesis Data Firehose, Kinesis Data Streams	MQTT, HTTP	AWS Greengrass	Embedded C, C++, Java, JavaScript, Python, Android, Arduino Yun, iOS
Microsoft Azure	IoT Hub	Event Hubs	MQTT, HTTP, AMQP	IoT Edge	C, Java, Python, Node.js, .NET, iOS
Google Cloud Platform	IoT Core	Cloud Pub/Sub	MQTT, HTTP	Edge TPU	Embedded C

# Azure vs AWS vs GCP : IoT Storage Services

Cloud Storage Services					
Cloud Platform	Object	NoSQL Database	SQL Database	Data Warehouse	Data Lake
Amazon AWS	Amazon Simple Storage Service (S3)	Amazon DynamoDB, Amazon SimpleDB, Amazon EMR (Apache Hbase)	Amazon RDS (Relational Database Service)	Amazon Redshift	AWS Lake Formation, AWS Data Lake Solutions
Microsoft Azure	Blob Storage	CosmosDB, Table, HDInsight (Apache Hbase)	SQL Database, PostgreSQL	SQL Data Warehouse	Data Lake Storage
Google Cloud Platform	Cloud Storage	Cloud BigTable, Cloud Firestore	Cloud SQL, Cloud Spanner	BigQuery	Cloud Storage, BigQuery

# Azure vs AWS vs GCP : Streaming/Analytics/Visualization

Streaming / Analytics / Visualization					
Cloud Platform	Real Time Stream Processing	Analytics	Serverless Compute	Machine Learning Services	Visualization
Amazon AWS	Amazon Kinesis Data Analytics	AWS IoT Analytics, Amazon Athena, Amazon EMR(Apache Framework Tools)	AWS Lambda	SageMaker	Amazon Quicksight
Microsoft Azure	Stream Analytics	Data Lake Analytics, HDInsight (Apache Framework Tools)	Functions	Machine Learning Services	Power BI Desktop, Power BI Service
Google Cloud Platform	Cloud Dataflow	BigQuery, Cloud Dataproc (Apache Framework Tools)	Cloud Functions	Cloud Machine Learning Engine	Cloud Datalab, Data Studio

# Data architecture and considerations Services

