mitrais continuous commitment

MITRAIS WHITE PAPER
THE INTERNET OF THINGS
AN EVALUATION OF VENDOR SOLUTIONS

VER.2.6

Table of Contents

Table of Figures

1. Overview

According to the National Intelligence Council's definition, the **Internet of Things** (IoT) refers to the general idea of things, especially everyday objects that are readable, recognizable, locatable, addressable, and/or controllable via the Internet—whether via RFID, wireless LAN, wide-area network, or other means.

Everyday objects in this context include:

- the electronic devices we encounter daily,
- products of higher technological development such as vehicles and equipment,
- things that we do not ordinarily think of as electronic at all, such as food, clothing, shelter, materials, parts, and subassemblies, commodities and luxury items, landmarks, boundaries, and monuments;
- and all the miscellany of commerce and culture.

The number of vendors offering platforms to enable IoT is growing in both closed and open source. Each vendor has its own strategy for implementing the IoT concept.

This document updates the previous Mitrais' evaluation of eight selected IoT development vendors from our Quarter 3 2016 newsletter, highlighting the advantages and, more importantly the limitations of their solutions to assist Mitrais customers in choosing the best IoT platform for their needs.

2. Evaluated Platforms

	PLATFORM	VERSION
1	Azure IoT	"April 2016"
2	ThingWorx	"April 2016"
3	Kaa	"April 2016"
4	AWS IoT	"June 2016"
5	IBM IoT Foundation	"June 2016"
6	Thingspeak	"June 2016"
7	Oracle IoT	"June 2016"
8	GE Predix	"Feb 2016"
9	Cumulocity IoT	"March 2018"

Table 1 – Evaluated Platforms

3. Other Platforms

Mitrais decided to only assess 9 of the leading IoT platforms. There are many vendors in this space, and customers may wish to assess some of the other IoT platforms which were not evaluated from the list below. Mitrais regularly reviews and updates its findings to cater for the changing market, conditions and may develop White Papers on these other platforms in the future.

- Bosch IoT Suite
- Ericsson Device Connection Platform
- Evrythng
- ParStream
- PlatONE
- Xively

4. Evaluation Criteria

CRITERIA	MITRAIS WEIGHTING
Compatibility	1
Connectivity	1
Application Development	1
Big Data Analysis	0.5
On-Premise & Cloud	1
Scalability	1
Security	1
Availability	1
B2C or B2B	0.5
Integration with other technologies (.Net, Java and Web)	1

Table 2 – Evaluation Criteria

Criteria used:

1. Compatibility
   The IoT Platform must effectively integrate users, objects and other enterprise applications on one platform.
2. Connectivity
   The platform must be compatible with heterogeneous devices through the support for various communication standards & protocols (Modbus, RS-232/485, MQTT, CoAP, REST, TR-069, SNMP etc.).
3. Application Development
   The IoT platform should feature a rich set of developer APIs, enabling seamless integration with other enterprise applications and minimization of vendor dependency.
4. Big Data Analysis
   The true business value of IoT is in data processing. A powerful data analytics tool is critical for a more agile and predictive environment.
5. On-Premise & Cloud
   The platform should provide for hosting of data both on-Premise and in the cloud,

based on specific requirements.

6. Scalability
   The platform should support larger scale IoT deployments connecting millions of devices without sacrificing performance.
7. Security
   The platform must support an end to end security mechanism, data auditing and authorization control.
8. Availability
   IoT applications demand service uptime and stable operations. The platform must recognize and handle adverse conditions such as device downtime and connectivity issues.
9. B2C or B2B
   It is vital that IoT applications can be designed as either B2C or B2B, since these modes demand different levels of visualization, performance and scalability.
10. Integration with other technologies (.Net, Java and Web)
   IoT platforms should have the capability to integrate with other technologies to extend the functionality of the IoT applications.

5. Platform Evaluation

PLATFORM	EVALUATION METHOD	1	2	3	4	5	6	7	8	9	10	TOTAL SCORE
ThingWorx	Product Overview	5	5	4	5	5	5	5	4	5	5	**43**
Azure IoT	Product Overview	5	5	5	5	3	4	4	5	5	5	**41**
IBM IoT Foundation	Product Overview	4	4	4	5	3	3	5	3	5	5	**36**
AWS IoT	Product Overview	4	4	5	5	4	3	5	4	5	5	**39**
Kaa	Product Overview	3	4	3	3	5	3	3	3	5	3	**31**
Oracle IoT	Product Overview	4	3	4	4	3	3	3	3	5	3	**30.5**
ThingSpeak	Product Overview	3	3	3	2	2	2	2	2	5	2	**22.5**
GE Predix	Product Overview	3	4	2	5	2	3	5	4	4	3	**30.5**
Cumulocity IoT	Product Overview	4	4	4	5	5	5	5	4	5	5	**41**

Table 3 – Platform Evaluation

6. Platform Recommendation

6.1. Compatibility

AzureIoT supports compatibility with several device types. For the details please see this link. AzureIoT can be integrated with the current Azure environment (user, network, etc.), enabling easier development of an IoT solution using AzureIoT suite.

ThingWorx uses the ThingWorx composer to model the integration of users, objects (devices, etc.) and other relevant elements. The solution is then deployed and connected to the actual devices.

Kaa offers a simpler process for integration of users and devices. It provides an Admin UI so a user can modify the SDK and end points. Kaa supports a few device platforms. The detailed list can be seen through this link. There is no feature to integrate users and apps. Kaa integrates between end points and SDKs.

AWS IoT supports several devices which have been listed as AWS partners - details can be found in this link. AWS has prepared kits for those device types. Several SDKs are used during the development of an IoT application:

- The Device SDK (C, JavaScript, and Arduino Yún) runs on the device
- The AWS SDKs give you access to AWS from your web or mobile app

The AWS console can be used to develop the application on the AWS side then connect the device to the AWS console.

IBM IoT Foundation supports several devices which are listed as its partners. Below are some examples of IBM IoT partners:

- Texas Instruments Store
- ARM IBM-mbed IoT starter kit dev board
- Intel Galileo technology
- Raspberry Pi
- Intel Edison technology
- Arrow Dragon board 410 c
- Avnet Agate IoT gateway
- Avnet MicroZed

IBM also supports device and application development. The list of supported devices and application development elements can be found in this link.

Oracle IoT provides software libraries for development on the device level. Detailed in this link, a device needs to be registered and activated in Oracle IoT Cloud Service. Users use the Oracle Management console to develop the application on top of Oracle IoT.

GE Predix architectures and services (see this link page 6) are mainly intended for Industrial IoT which allows industrial machines (such as Turbines, Aircraft and Locomotives) to communicate with the cloud. However, it is also supports human IoT (phone, tablet, wearable).

A software layer component called Predix Machine can be configured to connect various devices (gateways, industrial controllers and sensors) with the Predix Cloud.

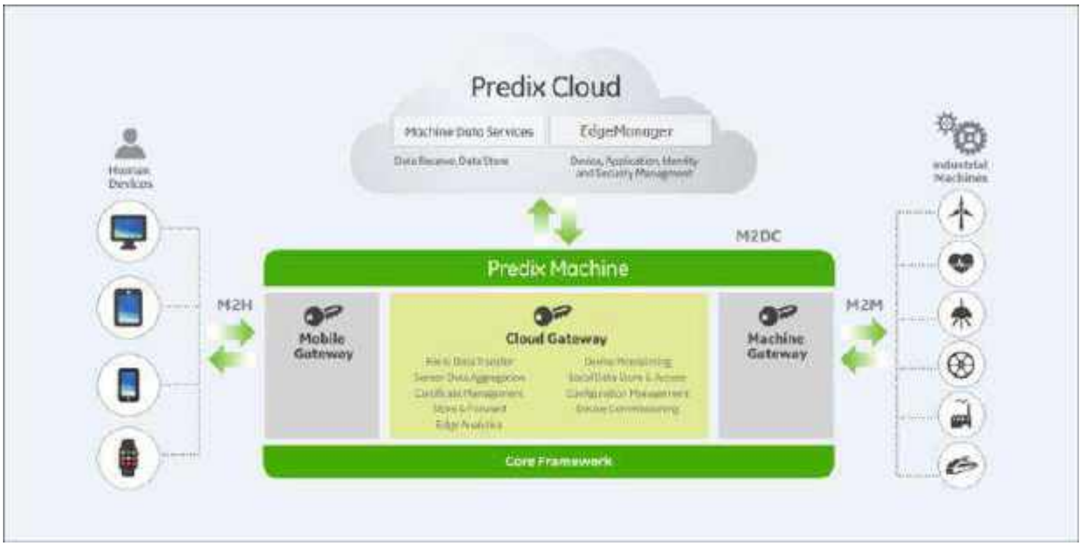
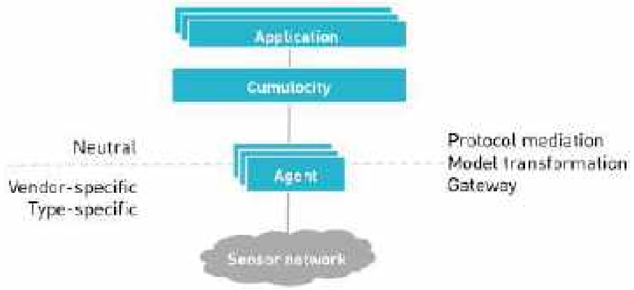


Figure 1 – Predix Machine High-Level Functional Architecture

Cumulocity IoT. The basic life cycle for integrating devices into Cumulocity is discussed in [Interfacing devices](). To interface IoT devices and other IoT-related data sources with Cumulocity, a software driver called an *agent* is required. An *agent* undertakes three functions for a specific vendor and device type:

- It translates the device-specific interface protocol into a single reference protocol.
- It translates the specific domain model of the device into a reference domain model.
- It enables secure remote communication in various network architectures.

.



Related topics can be found in the following sections:

- Cumulocity's domain model, for understanding the data structures exchanged between agents and the Cumulocity core.
- Device integration, for understanding in detail how to develop agent software.
- Reference guide, for a detailed specification of the interfaces between agents and the Cumulocity core.

6.2. Connectivity

All evaluated platforms offer connectivity into many different device and communication types. Until IoT standards evolve, it will continue to be important to choose a tool that supports the device types that your organisation uses. The following tables highlight the complexities, as each vendor specializes in different protocols, programming languages and hardware support.

AZUREIOT	THINGWORX	KAA
Operating System: - Windows IoT Core - Windows Desktop - Windows Server - Linux Variants	**Operating System:** - Windows - Linux Variants	**Operating System:** - Android - iOS - Linux - Snappy Ubuntu - QNX neutrinos - Windows
Hardware: Several IoT Devices	**Hardware:** - Intel Galileo - Intel Edison - Raspberry Pi - PC EMS - Arduino - Beaglebone	**Hardware:** - Intel Edison - Beaglebone - Raspberry Pi - Econais - LeafLabs - Texas Instruments CC 3200 - ESP 8266 - UDOO - ARKTIK
Protocols: - HTTPS - AMQP - MQTT - AMQP over sockets	**Protocols:** - MQTT - CoAP - Odata - OPC - Modbus - Zigbee/Zwave	**Protocols:** - HTTPS - TCP
Programming: - Java Libraries - C# Libraries - NodeJS Libraries	**Programming:** - Java - C - .NET (Available in Marketplace)	**Programming:** - Java - Ansi C and C++ - Objective C

Table 4 – Connectivity Comparison between AzureIoT, Thingworx and Kaa

AWS IOT	IBM IOT FOUNDATION	ORACLE IOT
Operating System: - Linux - Real Time OS	**Operating System:** - Windows - Linux Variants	**Operating System:** - Windows - Linux Variants
Hardware: - Raspberry PI	**Hardware:** - Raspberry PI	**Hardware:** - Raspberry PI
Protocols: - MQTT - Web Socket - HTTP	**Protocols:** - MQTT - HTTP	**Programming:** - Java - POSIX C - Java Script
Programming: - Java Script - C - SQL - Linux - Real Time OS	**Programming:** - C# - Node.js - Python - Embedded C - mBedded C++ - Java - Windows	

Table 5 – Connectivity Comparison between AWS IoT, IBM IoT Foundation, and Oracle IoT

THINGSPEAK	GE PREDIX	Cumulocity
Operating System: - Linux Variants	**Operating System:** - Windows - Unix-like OS such as Linux and MacOSX	**Operating System:** - Raspberry Pi, - the Kontron M2MSSDK, - Mac OS X, - Linux and - Windows
Hardware: - Arduino - Net Arduino	**Hardware:** - Industrial Machines: turbine, locomotive, jet engine - Human Devices: computer, tablet, phone See Predix Machine High-level functional architecture	**Hardware:** https://www.cumulocity.com/dev-center/#devices
Programming: - C	**Protocols**: - For Machine to Machine connection: OPC-UA, Modbus, and MQTT - For Cloud Gateway: Https, WebSockects	**Protocols:** -. Cumulocity uses a simple and secure reference protocol based on REST (i.e., HTTPS) and JSON, which can be used for a wide variety of programming environments down to small embedded systems. To support real-time scenarios, the protocol is designed around a "push" model, i.e., data is sent as soon as it is available. -. MQTT
	Programming (ref this link): For client connectivity: - Java For Analytic Service purposes (ref this link page 23): - Java - Matlab - Python - Ruby on Rails For microservice purpose (ref this link) - Front-end: HTML/CSS/JS on Node.js - Back-end: Java SprintBoot REST	**Programming:** - Java - C++ - Node.Js - Javascript

Table 6 – Connectivity Comparison between ThingSpeak, GE Predix and Cumulocity IoT

6.3. Application Development

Azure IoT supports application development using the [AzureIoT Suite](). Development is integrated with Visual Studio 2015, and it is necessary to install the Azure SDK to develop a solution. The connection to the IoT assets is established using the Azure [IoT hub]() which is an easy and secure.

ThingWorx provides a modelling tool called ThingWorx Composer to enable users to develop an IoT solution. A user can model the 'Thing', drawing on many useful features. After the modelling, the solution is connected to the [devices]().

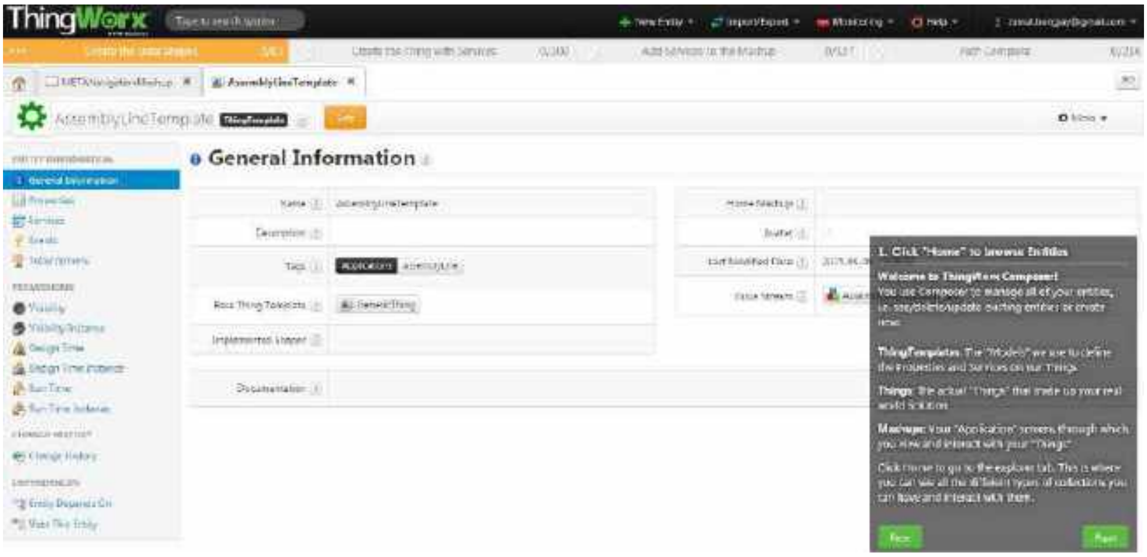


Figure 2 – Screen shot of ThingWorx composer

Kaa provides several features in its Kaa Server which can be found in this [link](). To connect with the devices, Kaa provides several end points including the following SDKs.

- [Java End Point SDK]()
- [C++ End Point SDK]()
- [C End Point SDK]()
- [Objective C End Point SDK]()

AWS IoT provides an SDK to help easily and quickly connect hardware devices or mobile applications. The AWS IoT Device SDK enables devices to connect, authenticate, and exchange messages with AWS IoT using the MQTT, HTTP, or WebSockets protocols. The Device SDK supports C, JavaScript, and Arduino, and includes the client libraries, the developer guide, and the porting guide for manufacturers. Open Source alternatives or write-your-own [SDKs]() are available. The architecture of AWS IoT is shown below:

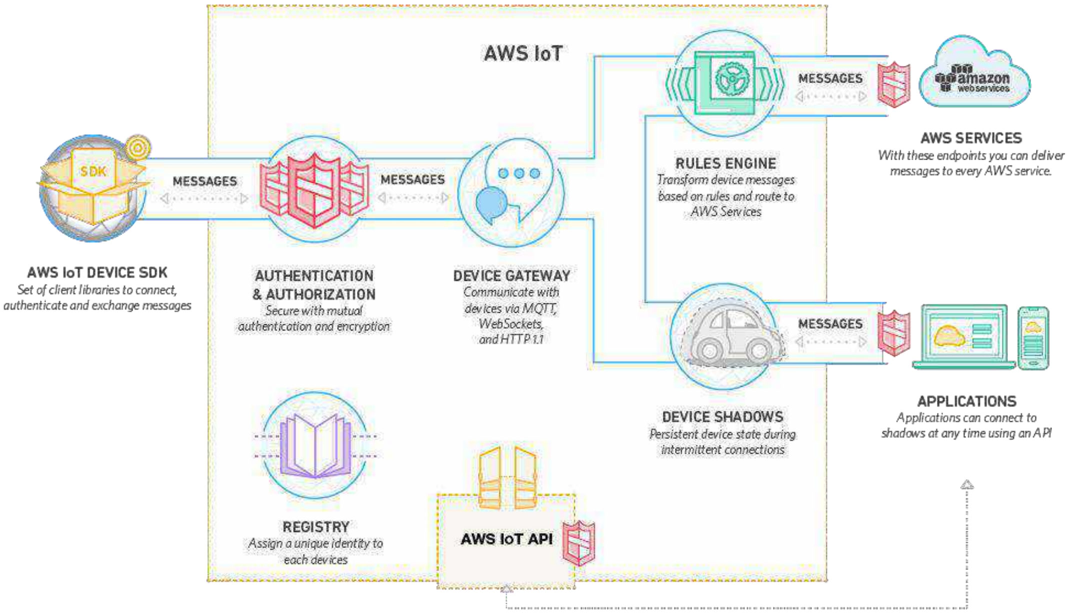


Figure 3 - AWS IoT Architecture

IBM's IoT Foundation supports the development for application and device development. Details can be found in this [link](). The architecture of IBM's IoT Foundation is shown below:

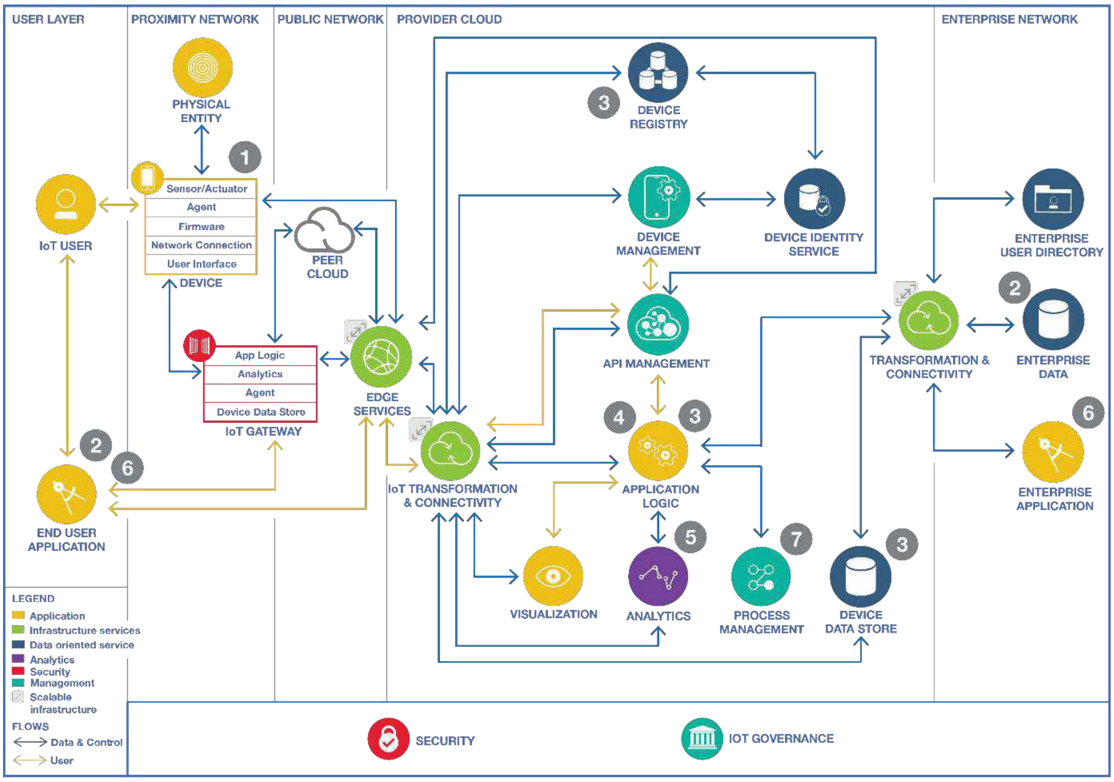


Figure 4 - IBM IoT Foundation Architecture

ThingSpeak only supports application development on the client side using arduino and netduino devices. To connect the device to the ThingSpeak server, it is necessary to create [ThingSpeak Apps]().

Oracle IoT supports development on either the device or the cloud. The Oracle IoT Cloud Service Client Software Libraries are embedded libraries that give a simple and thin-client alternative to the Oracle IoT Cloud Service Gateway. They comprise:

- Device Libraries - These libraries run on the client devices and are designed to make it easy to expose the device's functionality to the Oracle IoT Cloud Service. They make it simple to connect devices to the Oracle IoT Cloud Service through security lifecycle management and bidirectional messaging features.
- Enterprise Libraries - These libraries make it easy to remotely inspect, monitor, and control a device.

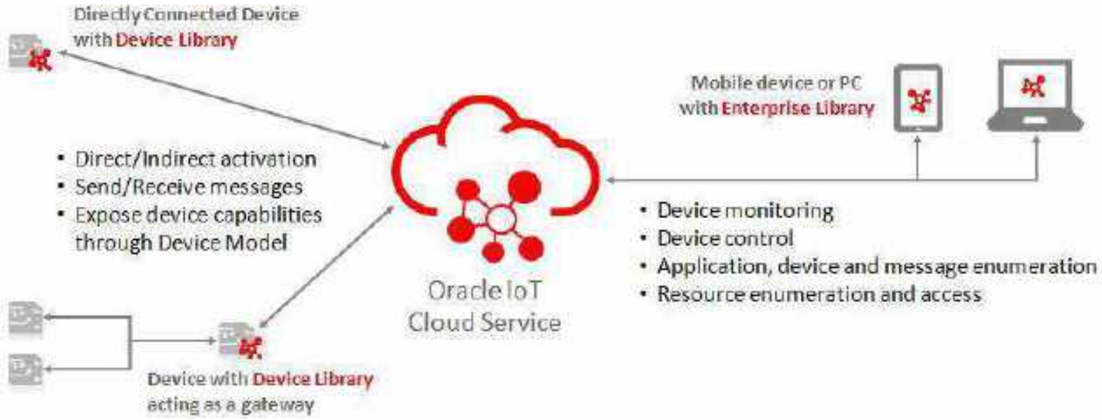
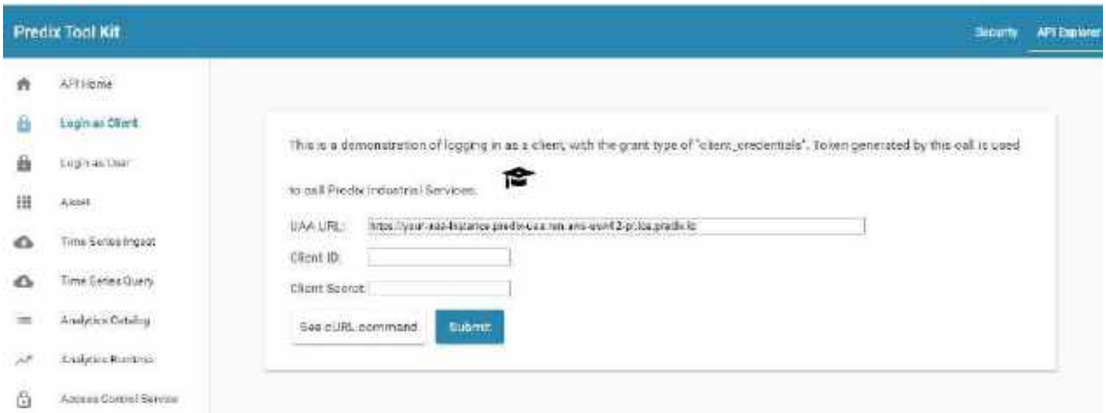


Figure 5 - Oracle IoT Foundation Architecture

GE Predix provides the Predix UI to develop powerful, user-friendly Industrial web applications running on top of Predix services and data.

It also provides Java SDK to communicate with the Time Series service (the cloud API to push and query data). A toolkit (a built-in web application) is available to access the Predix Cloud (as illustrated below)



Cloud Foundry and BizOps Service environment are also available to make building, testing, deploying and scaling applications faster and easier.

Cumulocity is designed to accommodate arbitrary vertical IoT applications in addition to its generic functionality. Cumulocity applications can have two forms:

- Web-based user interface applications ("web applications"). The Cumulocity user interface itself is built around a framework based on AngularJS and Bootstrap, the modern HTML5 web application frameworks. It is designed in a modular fashion around a set of plugins that enables developers to create their own configurations of the Cumulocity user interfaces. For more information on developing plugins, refer to the Web SDK for Plugins in the Web Developer's Guide.

- Server-side business logic through microservices ("microservices"). The Cumulocity microservice is based on docker, hosted by Cumulocity and exposing a REST API. When developing a Cumulocity microservice, a developer is not restricted to any programming language. However, a microservice must serve as a HTTP server working on port 80 and must be encapsulated in a docker image. A docker image is an executable package that includes everything needed to run an application. For more information on docker refer to the Docker documentation).

6.4. Big Data Analysis

Azure IoT supports Big Data analysis in the Azure Environment by enabling the use of the Stream Analytics feature in Azure.

Figure 6 - Example of a Stream Analytic Scenario using Event Hub

ThingWorx supports Big Data analytics using ThingWorx Analytics. This is separated from the core ThingWorx application. The following is an example of analytics using this tool to build an app to provide automated predictive intelligence.

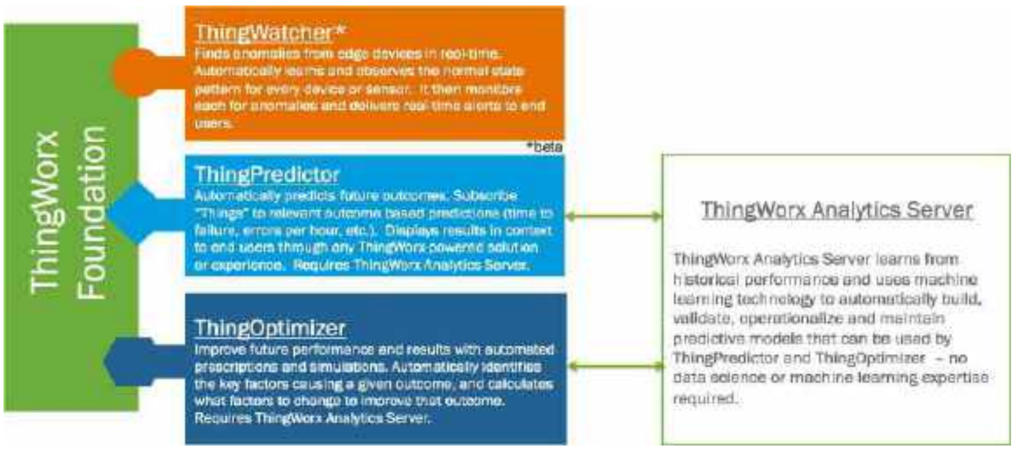


Figure 7 – ThingWorx Analytics Architecture

Kaa does not support analytics on top of its platform, requiring a user to develop their own analytics solution.

AWS IoT uses the AWS Service that handles Big Data Analysis such as Elastic MapReduce(EMR), Amazon Redshift and Amazon Kinesis. It is possible to use AWS IoT to provide Messages to be delivered to other AWS Service.

IBM IoT Foundation supports Big Data Analysis via the platform's connection to the IBM Data and Analytics feature. The detailed architectural diagram is shown below:

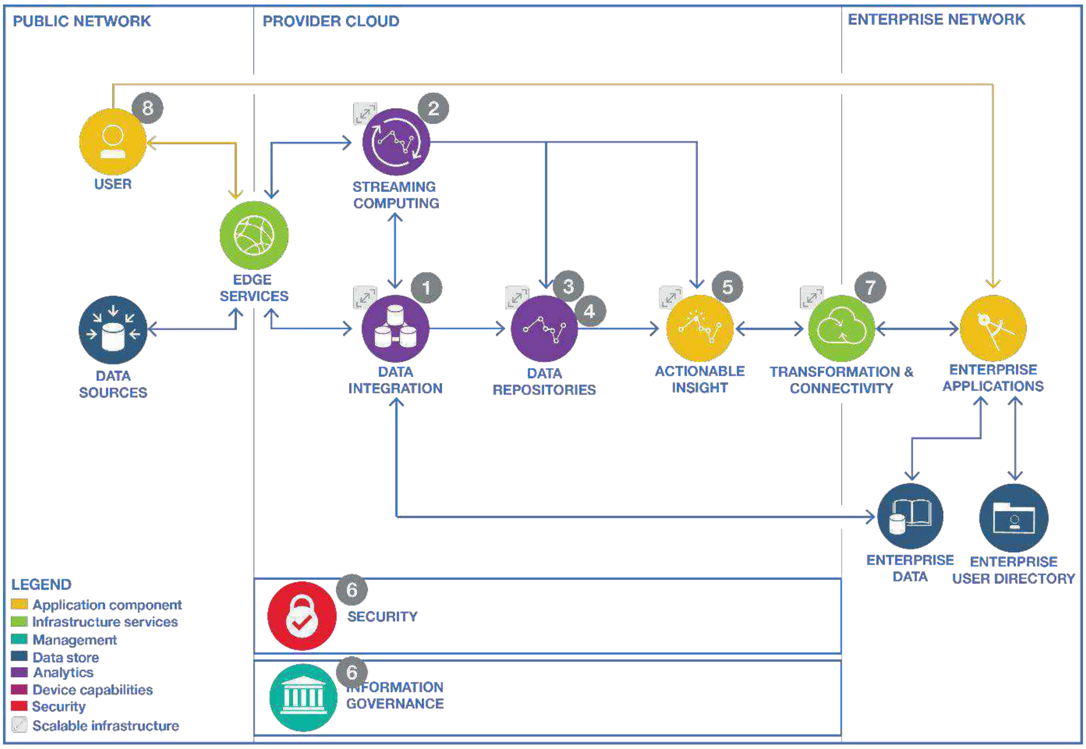


Figure 8 - IBM IoT Analytics Architecture

ThingSpeak lacks any features directly related to Big Data Analysis, but provides integration with MATLAB. This feature assists in developing Big Data Analysis features on top of ThingSpeak.

Oracle IoT uses another Oracle service, Oracle's Big Data Appliance, to implement Big Data Analysis. In addition, there are also other services that support Big Data Analysis listed in this link.

GE Predix provides various models for developing Analytic applications and workflows such as:

- Trend Anomaly Detection: detects trend anomalies in data based on robust statistics
- Image-Based Diagnosis: provides identification of anomalous images out of a set of provided images
- Numerical Data Preprocessing: data cleaning, transformations, standardizations, normalizations, data binning, feature extractions, handling outliers and/or null values, imputing missing data, and more
- Time Series Clustering: a domain-independent, accurate, and scalable algorithm for time-series clustering, with a distance measure that is invariant to scaling and shifting

It also supports Big Data Analytics (data science and advanced cloud analytics).

Cumulocity real-time event processing, allows you to add your own logic to your IoT solution. This includes data analytics logic but it is not limited to it. To define new analytics, you will use the Cumulocity Event Language. The language allows analyzing incoming data. It is using a powerful pattern and time window-based query language. You can create, update and delete your data in real-time. The basics for understanding how the Cumulocity Event Language works and how you can create your own analytics or other server-side business logic and automation can be found in this link.

Typical real-time analytics use cases include:

- Remote control: Turn a device off if its temperature rises over 40 degrees.
- Validation: Discard negative meter readings or meter readings that are lower than the previous.
- Derived data: Calculate the volume of sales transactions per vending machine per day.
- Aggregation: Sum up the sales of vending machines for a customer per day.
- Notifications: Send me an email if there's a power outage in one of my machines.
- Compression: Store location updates of all cars only once every five minutes (but still send real-time data for the car that I am looking at to the user interface).

6.5. On-Premise and Cloud

AzureIoT supports only in-Cloud solutions. Support for the on-Premise solution is promised with Windows Server 2016 but has not yet been released.

ThingWorx supports in-Cloud and on-Premise solutions. For on-Premise options, ThingWorx Composer could be deployed in a local network.

Kaa, like ThingWorx, can be deployed in a local network or in the cloud.

Cumulocity IoT available in the cloud, on-premises, at the edge or in any hybrid combination.

AWS IoT, Oracle IoT, IBM IoT Foundation, **ThingSpeak** and **GE Predix** only provide in-Cloud solutions, without offering an on-Premise service.

6.6. Scalability

AzureIoT, using Azure IoT hub, can support millions of simultaneously connected devices, but at a price. Each IoT Hub unit allows a restricted number of daily messages.

In order to properly scale a solution, it is recommended that users consider the required peak throughput for the following operations:

- Device-to-cloud messages
- Cloud-to-device messages
- Identity registry operations

With **ThingWorx**, there is no price for scaling. While users can scale their 'Thing' using the ThingWorx composer, they should nevertheless consider the architecture of ThingWorx elements.

Kaa, according to their FAQ, is designed to be horizontally scalable. Performance tests have indicated that there is a linear dependency between the system performance and the number of nodes, so as nodes are added performance would increase proportionally.

AWS IoT does not provide any specific features related to scaling, however they calculate the price based on the number of messages and regions. The detailed pricing can be found in this link.

IBM IoT Foundation doesn't have any specific features to allow scaling of devices. However, the software architect\IoT architect needs to consider how the shared subscription feature can be used to scale the device as per requirement. Please find the detail sample in this link.

ThingSpeak doesn't provide any specific features to scale devices. The solution's architecture needs to be considered to achieve this.

Oracle IoT provides different prices for a different number of devices. For the details please see this link.

GE Predix doesn't explicitly limit the connected devices number, however the cost will be vary based on the number of services and amount of usage.

Cumulocity IoT provides a distributed architecture, designed to scale and simplify development.

6.7. Security

AzureIoT claims to provide the most secure IoT platform to keep infrastructure secure, having instituted the Security Development Lifecycle (SDL) to ensure a mandatory Microsoft-wide development process for security.

AzureIoT provides security in three major areas:

- Device Security
- Connection Security
- Cloud Security

The details can be found in this link.

ThingWorx has several security features listed below:

- Authentication and authorization
- Security Logging Sub System
- Encrypted Storage of All Sensitive Data
- Protection against Common Vulnerabilities
- Backdoor Protection
- Support for Transport Layer Security
- Multi-level authentication
- Secure tunnelling for File Transfer and Application
- Provided Connectivity Security:
    - Standard PKI Infrastructure
    - TLS 1.x support
    - Support both client and server certificate validation
    - Password protected PEM key file storage
    - 128 bit AES encryption or higher
    - FIPS validated chippers supported
    - Encrypted configuration file entries

Kaa provides security on top of its default transport communication. It is secured using hybrid encryption using RSA and AES. The size of an RSA Key is 2048 bits and the AES is 256 bits.

Using **AWS IoT,** each connected device needs a credential to access the message broker or the Thing Shadows service. All traffic to and from AWS IoT must be encrypted over Transport Layer Security (TLS). Device credentials must be kept safe in order to send data securely to the message broker. After data reaches the message broker, AWS cloud security mechanisms protect data as it moves between AWS IoT and other devices or AWS services. The detailed AWS IoT Security Diagram is shown below:

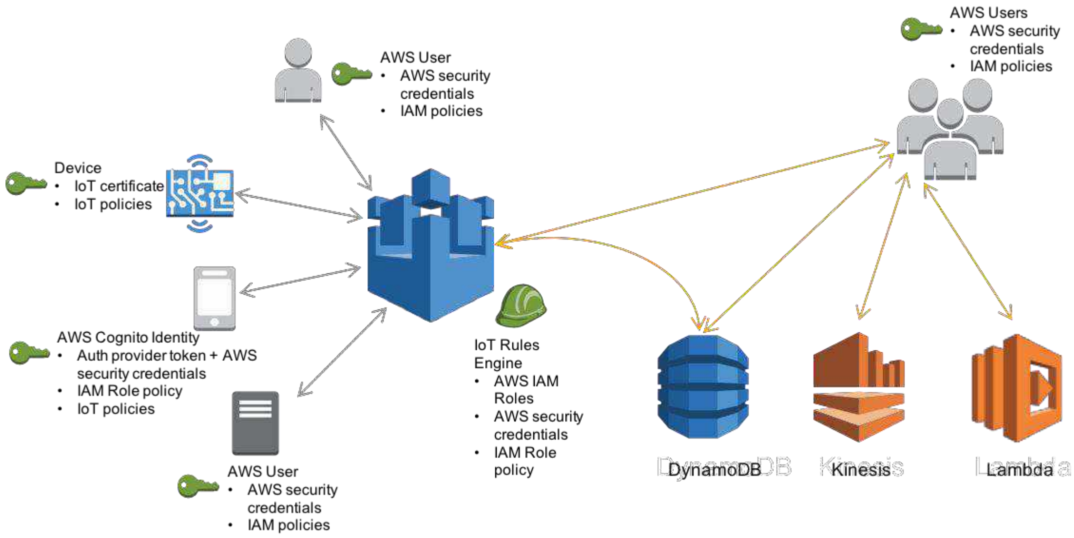


Figure 9 - IBM IoT Analytics Architecture

IBM IoT Foundation implements security on 3 main tiers. IoT solution components that run in each tier need to incorporate specific security measures to protect against various vulnerabilities. Details of the tiers are:

- **Devices/Gateways tier:** Protects against "fake" servers that send malicious commands, or against a hacker that tries to listen to private sensor data being sent from the devices.
- **Network/Transport tier:** Protects against a "fake" device that sends false measurements that might corrupt the data that is being persisted in the application.
- **Applications tier:** Protects against the invalid use of data, or the manipulation of analytical processes that are running in the application tier.

The diagram below shows the three tiers of a typical IoT application that uses IBM Watson IoT Platform in the network/transport tier and the IBM Bluemix cloud platform in the application tier.

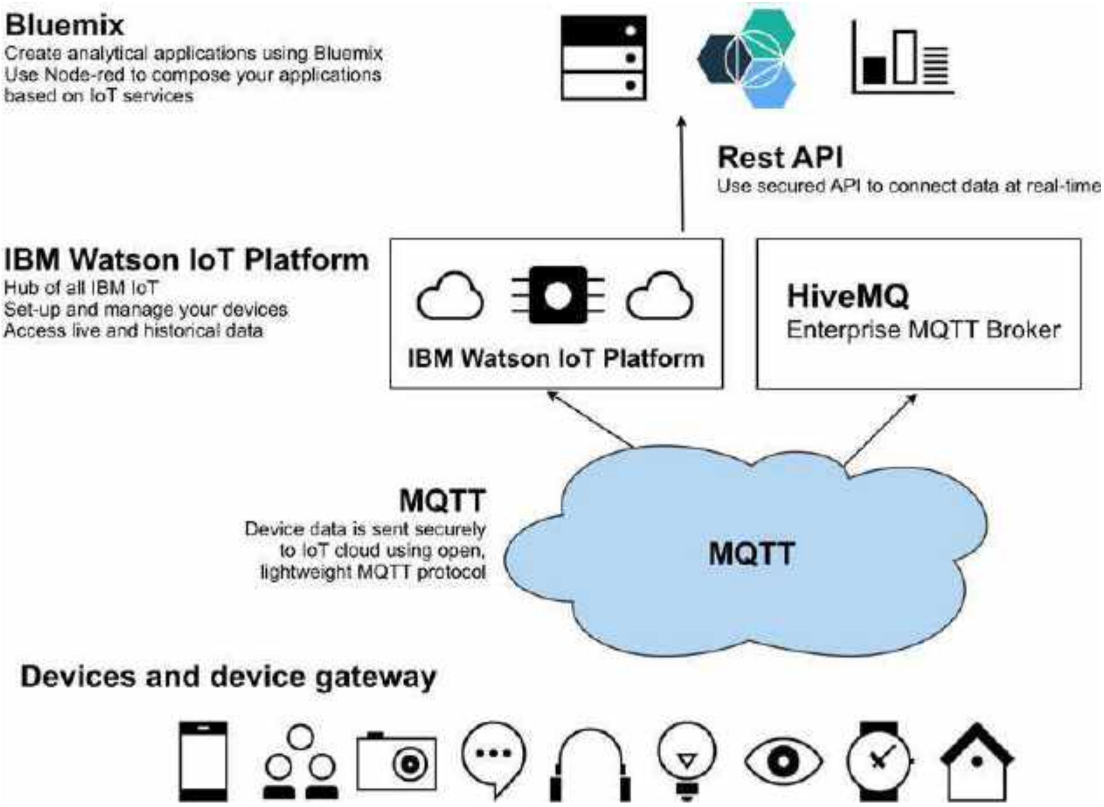


Figure 9 – IBM IoT Security Design Diagram

ThingSpeak doesn't have specific security features within their product, so the developer will need to implement the security features on their own in their devices.

Oracle IoT provides Oracle IoT Service Client Libraries which are available in source and binaries to enable secure, reliable connectivity of devices with the Oracle IoT Cloud Service. Client Libraries will become available for a variety of platforms including C/C++, Java, Android, JavaScript, iOS.

GE Predix provides security, authentication and governance services for endpoint devices through the Predix Machine. It supports certificate management to provide SSL-based connections to the Predix Cloud (ref this link page 8).

Predix also features Application Security Services, i.e. Services to meet end-to-end security requirements, including those related to authentication and authorization (ref this link page 14).

Cumulocity security concepts and aspects are structured into physical security, network security, application security and access control. This shows how Cumulocity helps in managing the security of your IoT solution.

More information can be found in the Cumolocity security-related sections, the REST Implementation reference and the User API. Permissions required for individual API calls are documented in the respective reference guide sections for the APIs.

6.8. Availability

AzureIoT has several features to support the availability of devices. These features can be found in this link.

As **ThingWorx** supports on-Premise and in-Cloud deployment, the user has the freedom to maintain the availability of the solution using ThingWorx. ThingWorx also provides several features to support availability, such as a recommended supported application backup strategy.

Kaa doesn't have any specific features to meet the availability criteria. Since Kaa can be deployed in-Cloud and on-Premise, the user must implement their own solution to achieve high scalability.

AWS IoT has provided an integration of AWS IoT Rules into Amazon Elasticsearch Service and Amazon CloudWatch to support the availability of the devices. The Amazon Elasticsearch Service is used to route device-generated data directly to Amazon Elasticsearch domains, enabling analysis of the data, the performance of full-text or parametric searches on the data, and visualisation using Kibana.

Amazon CloudWatch enables the viewing of graphical device metrics and the setting of alarms.

IBM's IoT Foundation provides an integration to existing IBM features so other features such as IBM Data Back Up and Recovery can support the availability of the devices.

ThingSpeak doesn't have any specific features related to availability and is only supported on Cloud. Users will need to develop features to support availability on their own. This could include a back-up strategy, as well as device monitoring and logging.

Oracle IoT doesn't have any specific features to support availability, however, the Oracle IoT cloud service has features to maintain the devices and device settings through its console. Below are the features:

- Managing Installed Device Software
- Managing Alert Messages
- Managing Device Data Message
- Managing Oracle IoT Cloud Service Settings

For the detail please visit this link.

GE Predix provides these features to promise availability (ref this link page 5):
Client-perspective:

- **Predix Connectivity** is for scenarios where a direct Internet connection is not readily available. It provides fast, secure cloud connectivity from Predix Machine to the Predix Cloud.
- **Predix Machine** component can monitor data collected from sensors and, using physics-based analytics, detect potential error conditions based on the asset model.

Server-perspective:

- **Predix Cloud** is a global secure cloud infrastructure that is optimized for industrial workloads and for meeting regulatory needs.
- **Predix Services:** Predix provides industrial services that developers can use to build, test and run Industrial Internet applications. It also provides a catalog services marketplace where developers can publish their own services as well as consume services from third parties

Cumulocity will provide 99.9% availability. The SLA for availability does not include the planned downtime for software upgrades and other maintenance tasks. For downtime (planned and unplanned), the Cumulocity team will provide an announcement in the "Announcements" section of the help center and also provide the expected time when the system will be back. Ref: https://www.cumulocity.com/sla/

6.9. B2C or B2B

Kaa, ThingWorx, AzureIoT, AWS IoT, IBM IoT Foundation, Oracle IoT, ThingSpeak, GE Predix and Cumulocity IoT all support both B2B (Business to Business) and B2C (Business to Consumer) modes.

6.10. Integration with other technology (.NET, Java and Web)

Azure IoT provides integration to .NET, Java and web using IoT Hub. For web technology, Azure IoT uses the nodejs framework.

ThingWorx also supports integration to .NET (using .NET Edgeserver available in the marketplace), Java (using Java Edgeserver available in the marketplace) and web (HTML support).

Kaa only supports Java integration (Java end point SDK), C++ (C++ end point SDK), C (C end point SDK) and Objective C (Objective C end point SDK). Kaa does not support .NET or Web technology integration.

AWS has SDKs for both .NET and Java, but neither SDK supports **AWS IoT**. Only a few SDKs support the AWS IoT including Embedded C, JavaScript and Arduino SDKs.

IBM IoT Foundation supports several technologies which can be used to develop application and device components. Amongst them are:

- Python
- Node.js
- Java
- C#
- Embedded C
- mBed C++

ThingSpeak only supports C Language for its device development and for the server-side and provides no SDKs.

Oracle IoT only supports Java as its programming language.

GE Predix mainly supports Java (Time Series SDK in only available in Java). Analytic Catalog also accepts scripts written in Java, Matlab or Python languages.

Cumulocity supports agent development on three different levels:

- There are a number of full-featured open source agents and drivers in Cumulocity's bitbucket.org and mbed.org repositories. More information can be found in here.
- Client libraries for major runtime environments such as C/C++, JavaME/SE and Lua, again as open source in bitbucket.org.
- Technology-neutral REST APIs for other runtime environments.

7. Conclusion

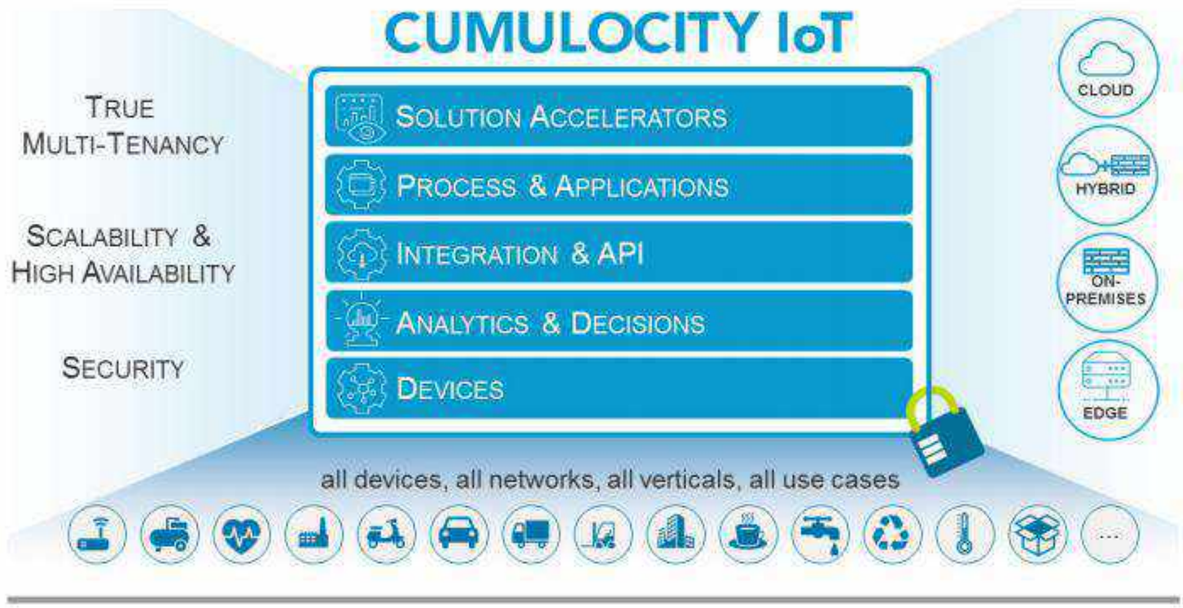
All options above have their pros and cons. Using our criteria, **ThingWorx** achieves the best score. It provides a solution which can be deployed in a Local Network or in the Cloud.

Azure IoT, AWS IoT, IBM IoT and **Oracle IoT** are also good options, but currently only support Cloud solutions. They do, however, offer many features and provide integration to other platforms. With **Azure IoT**, Microsoft is aiming to make its product available cross-platform, so this will ultimately be a benefit to customers.

Kaa, being open source, provides freedom and flexibility for customers to implement a range of solutions. **ThingSpeak** offers only cloud service and receives the lowest score. However, like **Kaa,** it provides freedom and flexibility for customers to implement a range of solutions. However, **ThingSpeak** and **Kaa** may be less productive because both require the building of features which are standard offerings with other IoT platforms.

If you are running an industrial ecosystem targeting Industrial IoT to control turbines, jet engines or locomotives, **GE Predix** is the most suitable. It can communicate with such devices, control them and provide analytical services.

Cumulocity IoT can be good option. It supports both cloud and on-premise solutionS, and offers a lot of features as shown here:



Cumulocity IoT Platform Capabilities

Copyright

Disclaimer

Any and all information in this document has been compiled and provided for information purposes only. The information provided herein may include information compiled from a variety of third parties. Mitrais will not be liable for any loss, damage, cost or expense incurred in relation to or arising by reason of any person relying on the information in this document or any link to any website provided herein, whether or not caused by negligence on Mitrais' part. While Mitrais endeavours to provide the information up to date and correct, Mitrais make no representations or warranties of any kind, express or implied, about the accuracy, reliability, completeness or currency of the information or its usefulness in achieving any purpose. Readers of this document are responsible for assessing its relevance and verifying the accuracy of the content. In any event, the information provided herein should not be construed as providing advice whether legal or otherwise.