Internet Of Things

A WHITE PAPER SERIES

Enable IoT Solutions using Azure

Atos | Syntel

TABLE OF CONTENTS

Executive Summary

Gartner has predicted that in few years, billions of devices will be sending information to systems and will be connected globally[1]. Almost all organizations will need to transform their businesses and technology to be able to leverage these **connected things**.

With Cloud becoming mainstream within organizations, it is being used as an excellent lever for digital transformation initiatives. Communication and information processing to and from devices, more commonly referred to as **Internet of Things (IoT)**, is one such digital transformation initiative.

However, transformations are disruptive in nature, and IoT also has its own share of challenges. First, we need to decide on the devices and their information architecture and how they can be used to transform the business. Second, we need to figure out a solution that will enable information storage and processing in an efficient and agile manner, without compromising on the enterprise security standards. Finally, we also need to consider the non-functional requirements as we are talking about billions of devices for any given solution and performance. Scalability and availability are the key areas here.

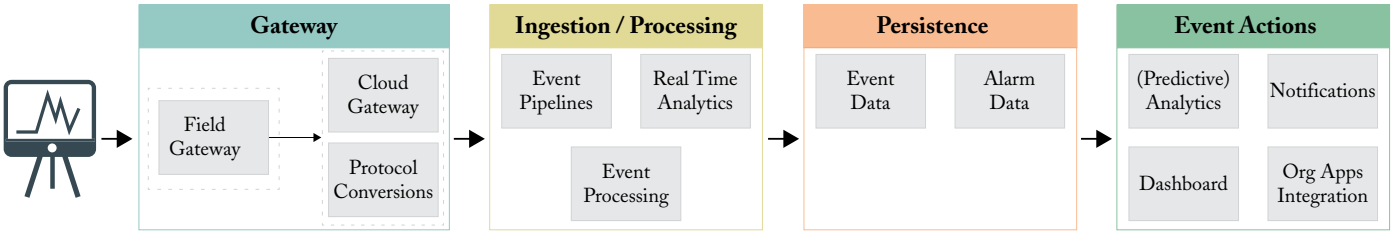
From our experience we have observed an increased focus within organizations on adopting Cloud for enabling their IoT solutions. Microsoft Azure is also one of the leaders in providing services and solutions that enable IoT.

This paper addresses the second and the third challenges by explaining how to go about creating an IoT solution and how to enable the same using the Microsoft Azure platform.

1. http://www.gartner.com/newsroom/id/3143718

2. Internet of Things

IoT refers to a new form of software and service related to device communications. Here devices send across real time data which is analyzed in real time and necessary actions taken based on the analysis. These actions can be anything from email and mobile notifications, to triggering actions within other applications and back onto the devices. Given below is a high level architecture of IoT:



High Level Architecture for IoT

The architecture starts with your devices that will be sending across signals containing data. These signals can be in any form, and generally use binary based proprietary protocols that vary from device to device.

• GATEWAY

Direct communication from the device to your applications that process data, can be a little messy. First, one will need to be able to scale to individual devices. Second, security and authentication can become a problem, since your applications will generally use a more heavy protocol for authentication, whereas the device will have a one-time programmable firmware that cannot have things such as passwords and certificates. Hence, we recommend a Field Gateway, which will be placed near the device and contain ways to communicate with the devices. It can also work as an aggregator (aggregate and send as a single unit) in cases where there are multiple types of devices sending out different sets of information. Additionally, this can also contain a protocol transformation, which will convert the binary-based protocol into a more standard protocol to be read by the other components of the system. Finally, this can also contain any decryption/decoding that is required before the information can get processed by the system.

This is usually implemented in two places, with one being near the device, called as the Field Gateway, and the other being near the event ingestion engine, called as Cloud Gateway. The field gateway should be able to communicate in proprietary protocol with the sensors. Additionally, it might have a predefined authentication handshake embedded in its firmware with the sensors to ensure device security.

The Cloud Gateway will have a second level of authentication with the field gateway, which will be more open and can also rely on a software based scheme. Also, it is recommended to use encryption between the field gateway and the cloud gateway to ensure message security. Finally, since the cloud gateway communicates with several field gateways, one will have to plan for scale and hence it is recommended to make the cloud gateway a stateless service. This will mean that one will need to ensure guaranteed message delivery to the next component, the event ingestion engine. Though this is like any other gateway, the reason why we call it a cloud gateway is due to its ability to be able to scale extremely fast in order to meet the event load from the field gateways, but not persisting the event. A device registry is also an important part of the Cloud Gateway, which authenticates messages from the right devices. The device registry can also double up as a reference registry in cases where the messages are supposed to be short and bandwidth is a concern.

• EVENT INGESTION

Event Ingestion is required with the cloud gateway to ensure message delivery and temporary message persistence, apart from message processing. Using the same scalability parameters as the Cloud Gateway, this will now need to ingest as many messages as have been passed by the field gateways. Considering the scale of messages, a message filtering logic during ingestion is highly recommended, as that will ease the pressure on the engine as well as reduce the processing time of the messages. A partitioned queue is a very good solution for this scenario, as it lends excellently to the design.

Apart from ingestion and partitioning, additional processing will also be required such as decryption, in case of message encryption. Additional processing consists of the decoding of messages with reference data coming out of another data source. Finally, some real time analytics in terms of time window based aggregation ensures that data is being looked at instantly and actions taken accordingly. All these activities should be done through Azure Stream Analytics, Azure Storm Service, and Microsoft Event Processing Host, hosted on custom Azure worker roles.

• EVENT PERSISTENCE

Though the event ingestion engine provides some amount of message persistence, there is a need to have a permanent storage that can save the data contained in the events. Hence, after decoding, one needs to save the data to a persistent store. Though the design of the store will depend on data, some guiding principles are as under:

- Do consider the size of data as it is going to be huge considering event information. A NOSQL/big data solution works best. Possible options on Azure are Azure Table Storage and/or Azure HDInsight service.
- Considering events will be generated from many devices, contemplate batch based transactions to ensure optimum and fast performance. Hence, having a separate worker process that fetches data from the temporary queue store and saves it to the permanent store is a good idea.

• EVENT ACTIONS

The fundamental reason behind ingesting information from the devices is to ensure we perform an action out of it. This will mean that we will need to continuously analyze incoming data and use it for subsequent communications or actions. This is done in many possible ways:

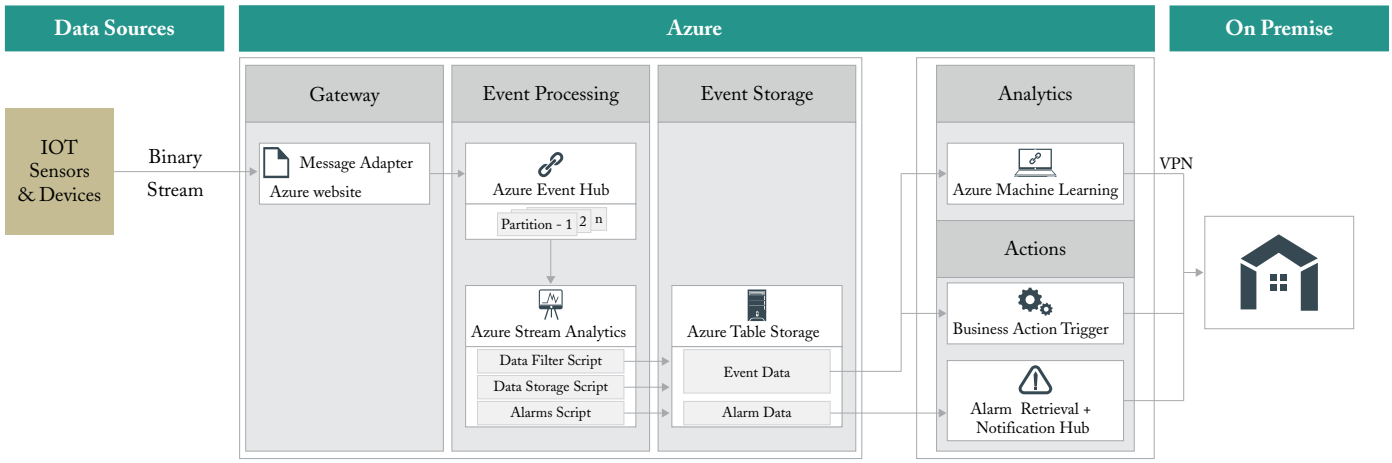
- Perform predictive analytics on historically collected data to predict future events. This could be done to identify outliners or disruptions in trends. This will require an analytic solution with some kind of statistical computation based on classification or neural based algorithms.
- Use data from real time analytics and event processing to alert or notify people and systems of a possible concern. This will require integration with third party services for email and mobile notification services. A separate component which examines persistent data to perform this function is recommended for scalability and availability purpose.
- Send processed information from the event message received to a third party application, hosted either on premise or on cloud. This can be implemented in a push model, where a component can process the information and then send it to the third party system. Alternatively, the third party system can directly examine this data to get the information. Decision will depend on how easy or difficult it is to change the third party applications, as well as where the processing logic for this needs to reside in.
- Communicate back to the device any actions you would want it to perform. This generally is possible in intelligent systems. This will require the setting up of a reverse queuing mechanism as well as the protocol conversion which was implemented in the Gateway. More often, the same gateway components can be used for two way communication, apart from a separate queue along the ingestion engine.

There are multiple ways to design and implement all these solutions on Azure. Azure offers IaaS as well as PaaS services. Using Azure IaaS, one can bring any of the components and host them on Azure Virtual Machines (VMs) to do the work as expected. However, that would lead to additional effort of managing the VMs apart from the application.

We recommend using Azure PaaS service as an alternative. With Azure PaaS, your VM management is completely owned by Microsoft, and you only need to ensure the scalability and availability of the applications.

3. Atos Syntel's IoT Solution on Azure Platform

Microsoft has been in the forefront of digital transformation for customers using its Azure Cloud platform. After some attempts on the IoT, it recently launched its Azure IoT Suite: a set of components that can tie together to provide a robust, scalable, and available platform for IoT based solutions. Though the suite has most of the components, Atos Syntel realized that there are some parts of the high level architecture which are not provided by Microsoft Azure as services. After doing implementations across our customers, we have built a reusable architecture for enabling IoT solutions on Azure. Given below is the IoT architecture for Azure:



Azure Reference Architecture for IOT

Detailed description of the components in each section of the architecture is given below:

- ### IoT DEVICES AND FIELD GATEWAY

These components are specific to vendors and MS Azure does not have anything in the IoT Suite. These tend to be micro devices with very limited capability of hardware and software. Most of the time these devices would be passing the collected data to a Field Gateway situated nearby through Bluetooth or wired connectivity. The Field Gateway has data reconciliation and Wi-Fi capability to share this data to the Cloud Gateway for further analysis, processing, and action. Field Gateway also has a data reconciliation program to consolidate information from a group of devices located nearby.

- ### CLOUD GATEWAY – AZURE WEB ROLES

One of the recommended ways to implement a Cloud Gateway is to use Azure Web Roles. In case of Azure, the need for Cloud Gateway also comes from the fact that Azure Event Hub supports HTTP and Advanced Message Queuing Protocol (AMQP) protocols for message ingestion. Devices and Field Gateways supporting binary protocols such as MQTT will need to convert data from one to another for communication. Thus, apart from authentication and encryption, here the Cloud Gateway also has a third functionality – protocol conversion. Cloud Gateway also needs to register and authenticate devices ingesting data to the IoT platform.

Atos Syntel has developed a component called ProCon to receive binary data and convert it to AMQP and also push it to the Azure Event Hub. ProCon is developed using .NET C# language and runs on the .NET framework. ProCon can be installed on Azure Web Roles and Azure App Service web apps, and can be scaled easily according to the incoming traffic using Azure Automation. ProCon is deployed as a web application and can be scaled up to millions of ingestion requests per second. ProCon has a device registration mechanism that persists the device details in Azure DB or Azure table storage. ProCon also has data ingestion device authentication that allows only the registered devices to push data and avoid any unwanted device and data intervention in the customer's IoT platform.

• AZURE EVENT HUB

Atos Syntel has analyzed the Azure event hub thoroughly and has suggested it to be used as a message intake component in its IoT reference architecture. Event hub is a highly scalable publish-subscribe event ingestor that can intake millions of events per second using AMQP communication protocol. Event hub has a limit of 32 partitions and 20 throughput units where each throughput unit has a capacity of 1mb per second. Maximum message size allowed through the Azure event hub is 256 kb or 1000 messages per throughput unit. IoT devices are mainly sensors which produce data in bits and pieces forming a message less than 1kb size most of the times. Event hub can be scaled further by adding 20 throughput units with each request resulting into millions of message per second capacity. Event hub also provides a message retention period of seven days and egress messages capacity of 2mb per throughput unit. Event hub also supports partition keys which can be used to filter and categorize data for customers and trigger actions for customer specific scenarios. Azure event hub exposes an end point to its clients to push messages into it, where consumer groups can be defined inside the event hub to isolate customer specific data.

• AZURE STREAM ANALYTICS

Atos Syntel recommends Azure Stream analytics for real time analysis and event processing. Once the messages are ingested into the IoT platform through event hub, these messages should be processed to convert them into useful events and real time actions. Real time actions would be concluding based upon the data ingested in the subsets of events grouped with time. Events grouped with time are known as windows which could be of three types - tumbling, hopping, and sliding. Given below are the algorithms used to process the data under different windows modes:

- Tumbling windows are a series of fixed sized, non-overlapping, and contiguous time intervals
- Hopping windows are modeled as overlapping windows defined by the window time and its overlapping with the previous one
- Sliding windows logically consider all possible windows of a given time length when the content of the windows actually changes

Azure Stream analytics has a strong processing power to process data in real time and predict abnormal data patterns. It also helps systems to take corrective actions based on the event data. Azure Stream analytics persists abnormal data patterns in terms of events which are retrieved by customer specific custom components to trigger corrective actions. Stream analytics support Event hub and reference data as input sources and Azure blob, Azure table, and Azure SQL DB as output sources. Stream analytics computing capacity depends upon the streaming units selected for a given instance of stream analytics. Streaming units have real time message processing capacity of 1 mb per second and the number of streaming units should be selected based upon the throughput units of the source Event hub instance. Streaming unit's configuration, taking the number of throughput units of the Event hub into consideration, would help in zeroing out any latency in processing the messages and generating alarms and events for customers. Data processed through Stream analytics can be persisted into Azure table storage or another instance of Event hub for further processing and actions.

• AZURE TABLE STORAGE

In our IoT reference architecture we have recommended Azure storage table for IoT data persistence. IoT scenarios produce huge data that need to be stored for later analysis in Azure platforms. Azure table storage service is a cost effective solution to store terabytes of structured data generated by sensors and devices pushed through Event hub and Stream analytics. Azure table storage is a NoSQL data store which accepts authenticated calls from inside and outside the Azure cloud. Azure table storage can be easily integrated with Azure Machine Learning (ML) for analytical processing and predictions. Azure table storage can be connected to HDInsight using map reduce connectors like Hive storage handlers for data processing. A record in the Azure table storage is known as an entity which is uniquely identified with the combination of a row key and partition key. Azure table storage also supports batch processing for its CRUD operations where a single batch operation can include around 100 entities at a time.

• AZURE MACHINE LEARNING

In order to leverage the true potential of the data generated through an IoT solution, it is extremely important to start analyzing data for identifying trends and anomalies. This requires predictive analysis and is often restricted to certain off-the-shelf tools. With Azure, a lot of such predictive analytics capability is offered through an as-a-service manner using Azure ML. With Azure ML, we can quickly build data models and train them using a variety of analytical algorithm to predict outcomes. Azure ML also supports R language and hence we can also migrate existing R based solutions to Azure. With Azure ML, it is easier to crunch huge data, as it is extremely scalable, in order to process any size of data. Azure ML is cost effective, since its pricing is based on data processing. With ML, one can quickly create, test, operationalize, and manage predictive models, and there is no need to buy any hardware nor manually manage virtual machines.

• ACTIONS

With Azure, we can define the following types of actions:

- Sending notifications to devices and applications
- Sending messages to other applications for performing an action
- Sending messages back to the device to perform certain functions

For sending notifications, Azure provides notification hubs. With notification hubs, you can seamlessly send mobile notifications to devices spanned across iOS, Android, and Windows phones. We can also broadcast messages to multiple devices, selectively using tags in notification hubs, or to all registered devices.

For sending messages to other applications, Atos Syntel recommends the use of Azure Service Bus for asynchronous communication across applications. In case of multiple applications, we can use Service Bus Topics as well. For single communication, Service Bus Queues work fine.
Finally, for sending messages back to the device, we will need to route the messages back to the Cloud Gateway, where the messages will be converted back to the protocol understood by the device/Field Gateway. Also, the messages will be encrypted and sent to the device, where they will be processed as per predefined firmware instructions.

4. Conclusion

We see tremendous potential in Internet of Things (IoT). However, there are considerations for building a connected system, and a reference architecture is highly recommended. Microsoft Azure does provide the building blocks to build such a reference architecture easily and quickly.

about Us:

Atos Syntel is a leading global provider of integrated information technology and knowledge process services. Atos Syntel helps global enterprises evolve the core by leveraging automation, scaled agile and cloud platforms to build efficient application development and management, testing and infrastructure solutions. Our digital services enable companies to engage customers, discover new insights through analytics, and create a more connected enterprise through the internet of things. Our "Customer for Life" philosophy builds collaborative partnerships and creates long-term client value by investing in IP, solutions and industry-focused delivery teams with deep domain knowledge.

To learn more, visit us at www.atos-syntel.net

ISO 9001 2000 SEI CMM LEVEL 5

For more information, visit us at **www.atos-syntel.net**

Atos | Syntel