“Many customers are already using the AWS IoT service to build Smart City services, connected cars, and many other applications. Working with Intel, we’ve developed a Proof of Concept to showcase Smart City applications that help to monitor vehicle emissions.”

Claudiu Pasa
Amazon Web Services EMEA IoT Business Development Lead

**Smart City management for lower emissions**

**Intel® IoT Gateway, AWS IoT Service**

**Internet of Things**

**Industry IoT integration with Intel**

By 2020, more than 20 billion Internet enabled devices will be connected to the cloud and each other in what is commonly called the Internet of Things (IoT)\(^1\). Communication protocols, device SDKs, and secure connectivity are key requirements to using the power of cloud resources and realizing the potential of the IoT. A key objective for most IoT solutions is to enable greater insights to be derived from data provided by connected devices, allow devices to make smarter decisions for users, and create entirely new business models and opportunities.

Unfortunately, industrial devices and other systems are not always designed with interconnectivity and easy data sharing in mind. Lots of useful data is often locked away in company equipment, like HVAC units, vending machines, or a fleet of vehicles. Computing advancements, such as Big Data clusters, give businesses the ability to process that data and gain useful information from it.

However, this data was not initially intended for analytics so remains inaccessible and uninterrogable in many cases. There is a definite need to address legacy system interoperability in order to avoid the significant resource sink of replacing existing infrastructure with Internet enabled components.

The Intel® IoT Gateway (Figure 1) is a key building block, allowing companies to connect legacy industrial devices and next-generation intelligent infrastructures to the IoT. Intel IoT Gateways integrate technologies and protocols for networking, embedded control, enterprise-grade security, and easy manageability on which application specific software can run smoothly. It enables:

- Local decision-making, enabling easy connectivity to legacy systems
- Connectivity up to the cloud and down to sensors and existing controllers embedded in a system
- A hardware root of trust, data encryption, attestation, and software lockdown for security
- Pre-process filtering of selected data for delivery
- Local computing for in-device analytics
IoT in the cloud

Amazon Web Services (AWS) is a leading provider of cloud computing infrastructure and services, and is working with Intel to build a robust IoT proof of concept (PoC). The AWS IoT Service (AWS IoT) is a managed cloud platform that lets connected devices easily and securely interact with cloud applications and other devices. AWS IoT (Figure 2) can support billions of devices, trillions of messages, and can process and route those messages to AWS endpoints and other devices reliably and securely. With AWS IoT, business applications can keep track of and communicate with all company devices all the time, even when they don’t have an Internet connection.

AWS IoT makes it easy to use services like AWS Lambda, Amazon Kinesis, Amazon S3, Amazon Machine Learning, and Amazon DynamoDB in the cloud. These help organisations to build IoT applications that gather, process, analyze, and act on data generated by connected devices, without having to manage any infrastructure, which saves time and costly investment.

“We believe that the power of the IoT lies in its simplicity, and by combining intelligent hardware with the cloud we can help to enhance the lives of people around the world.”

Claudiu Pasa
Amazon Web Services EMEA IoT Business Development Lead

Figure 1: The Intel® IoT Gateways can address various use cases

Figure 2: The AWS IoT Service
Smart City management PoC

Overcrowding, traffic congestion, and air pollution are becoming more critical as populations grow and more vehicles cram onto our roads. This poses a pressing challenge for city leaders, as it affects the quality of life for residents and commuters alike.

In order to demonstrate a complete end-to-end solution to this growing problem, using industry-ready building blocks from both Intel and AWS, the companies developed a PoC to demonstrate how Smart City applications can be architectured in the cloud.

In a real implementation, CO2 sensors would be located around the city and then sensor data would be read by the Intel IoT Gateway and sent to the AWS IoT in real-time. This data can be monitored by the City Manager web application (Figure 3) and would trigger predefined actions based on air quality values.

Such data could be used in many valuable ways. Actions that might be taken as a result of the data that would be collected and analyzed could include setting a congestion charge for entering areas of high pollution to try and reduce traffic, or offering tailored discounts on a park and ride option. The data could also be displayed on electronic road signs located at strategic locations in order to inform drivers, and hopefully reduce emissions even further.

Secure connectivity for cities

The IoT offers the opportunity for governments, city bodies, and local enterprises to collaborate on projects that use smart technology and data analytics to help monitor and manage congestion and other city challenges. AWS, together with Intel, developed an edge to cloud solution for this, which utilizes Intel and AWS components (Figure 4).

Figure 3: City Manager, with sensor locations and values, and Smart Road Signs

Figure 4: End-to-end Smart City management solution from Intel and AWS
This solution is centered around the Intel® IoT Gateway Development Kit DK100 Series, an intelligent device that connects the sensor simulator to the Internet. It collects sensor data and sends the data to the cloud using MQTT. The sensor simulator is a custom hardware unit consisting of eight potentiometers which simulate legacy technology and, when moved, voltages are set. In this case, the values simulate different levels of CO2 in the air.

A range of software runs on the Intel IoT Gateway, including Node.js Agent, which is a MQTT client service that sends sensor data from the gateway to the MQTT broker that is available in the AWS IoT, using MQTT-S. Key benefits come from the device security, smart connectivity, and rich network options built into Intel® technology. McAfee Embedded Control* Monitors and protects data flow by dynamically managing whitelists, which lends an extra layer of security.

Authentication and authorization is performed using AWS IoT-issued certificates. The process of creating and registering a certificate with AWS IoT is called provisioning. Restricted authorization is achieved using an AWS IoT policy (Appendix A) attached to the certificate, which then allows the Intel IoT Gateway to only publish to specified topics.

This solution supports both MQTT and HTTP protocols to provide a secure mechanism so devices and IoT applications can publish and receive messages between each other. For added privacy, it uses the secure MQTT protocol, MQTT-S, to communicate with the Intel IoT Gateway Node.js agent and obtain sensor data safely.

The solution has an in-built Rules Engine that provides message processing and integration with other AWS services and is used to send sensor data to Amazon DynamoDB tables (Appendix A). The table “MQTT log” stores all historical sensor data, including a timestamp, which can then be used for analytics. The other table, “Sensor Store”, is used to initialize sensor values and updated with the last known sensor value generated by an AWS Lambda function.

On the client side, the City Manager is a web service application run in an Amazon EC2 instance that communicates with the AWS IoT Device Gateway’s MQTT broker. A secure, bidirectional data stream is established via a websocket between the web service and the client’s browser. Using an AWS Elastic Load Balancer when the client first logs in, client side scripting relies on Google Maps APIs to identify and display the sensor locations with data from the Amazon DynamoDB table “Sensor Store”. This data is then used to set the initial sensor values via a websocket connector. Real-time sensor data is then published by the broker to the web service via the websocket connector, creating a quick, easy, at-a-glance view of all the sensor data that is securely and smoothly available at any time, on any device.

**Next steps**

This PoC demonstrates how an intelligent system gathers, analyzes, and acts on data from CO2 or other sensors placed around city streets. Connected via an Intel IoT Gateway to AWS IoT, this solution can be used to change driving behaviours and thereby offers the potential to improve traffic flows, reduce pollution, and cut congestion to improve the environment for everyone.

Often, the power of the IoT is in the simplicity of the solution. Intel and AWS are engaging the community to support, contribute to, and scale their system to solve this very real environmental problem. A City Hall or council can look into various options; for example, installing sensors on public transportation vehicles to allow a limited number of devices to cover a wider city area, or subsidizing device distribution to people keen to participate in a home-monitoring program and share the collected data. The key aspect here is the power of open sourcing the data to the wider development community to create innovative solutions and products for public services, from traffic routing to speed management, and consumers, with life style apps and real-time environmental alerts.

For more information visit [www.intel.com/iot](http://www.intel.com/iot), [www.aws.amazon.com/iot](http://www.aws.amazon.com/iot)

Find the solution that’s right for your organization. View success stories from your peers and check out the IT Center, Intel’s resource for the IT Industry.
Appendix A

AWS IoT Certificate Policy

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["iot:Publish"],
      "Resource": [
        "arn:aws:iot:us-east-1:awsaccountn#:topic//device/amazon_loft/sensors/input_0",
        "arn:aws:iot:us-east-1:awsaccountn#:topic//device/amazon_loft/sensors/input_1",
        "arn:aws:iot:us-east-1:awsaccountn#:topic//device/amazon_loft/sensors/input_2",
        "arn:aws:iot:us-east-1:awsaccountn#:topic//device/amazon_loft/sensors/input_3",
        "arn:aws:iot:us-east-1:awsaccountn#:topic//device/amazon_loft/sensors/input_4",
        "arn:aws:iot:us-east-1:awsaccountn#:topic//device/amazon_loft/sensors/input_5",
        "arn:aws:iot:us-east-1:awsaccountn#:topic//device/amazon_loft/sensors/input_6",
        "arn:aws:iot:us-east-1:awsaccountn#:topic//device/amazon_loft/sensors/input_7",
        "arn:aws:iot:us-east-1:awsaccountn#:topic//device/amazon_loft/sensors/$update"
      ],
      "Effect": "Allow",
      "Action": ["iot:Connect"],
      "Resource": ["*"]
    }
  ]
}
```

AWS IoT Rules

Invoke Lambda Function IoT Rule

```json
{
  "sql": "SELECT * FROM '/device/+/sensors/#'",
  "ruleDisabled": false,
  "actions": [
    {"lambda": {}}
  ]
}
```

DynamoDB Table Insert Sensor values Rule

```json
{ "sql": "SELECT * FROM '/device/+/sensors/#'",
  "ruleDisabled": false,
  "actions": ["dynamodbDB": {"hashKeyField": "topic",
                          "roleArn": "arn:aws:iam::awsaccountn#:role/my-iot-role",
                          "tableName": "mqtt_log",
                          "hashKeyValue": "${topic()}",
                          "rangeKeyValue": "${timestamp()}",
                          "rangeKeyField": "timestamp"}],
  "ruleDisabled": false,
  "actions": ["context.fail('unexpected event')"]
}
```

Lambda Function Code

```javascript
const AWS = require('aws-sdk');
vary dynamodbDoc = new AWS.DynamoDB.DocumentClient();
exports.handler = function(event, context) {
  console.log('Received event:', JSON.stringify(event, null, 2));
  if (event.topic) {
    var splittopic = event.topic.split('/');
    if (splittopic.length !== 5) {
      context.fail("Error: unexpected topic length");
      var devicename = splittopic[2];
      var sensorname = splittopic[4];
      var value = {
        value: event.value,
        lat: event.lat,
        lng: event.lng
      }
    var params = {
      TableName: "SensorStore",
      Item: {
        "devicename": devicename,
        "sensorname": sensorname,
        "fulltopic": event.topic,
        "sensorvalue": JSON.stringify(value),
        "lastupdated": Date.now()
      }
    }
    dynamodbDoc.put(params, function(err, data) {
      if (err) {
        context.fail(JSON.stringify(err), null, 2);
      } else {
        context.succeed('Item inserted!');
      }
    });
  } else {
    context.fail('unexpected event')
  }
} //context.fail('Something went wrong');
```


You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

In cases where security vulnerabilities may exist in Intel products, Intel may recommend and provide solutions to address these vulnerabilities.

You should consult your system manufacturer or retailer or learn more at http://www.intel.com

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to www.intel.com/performance

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries. *Other names and brands may be claimed as the property of others.

© 2016, Intel Corporation 334105-001 Please Recycle