Internet-of-Things
Cloud integration

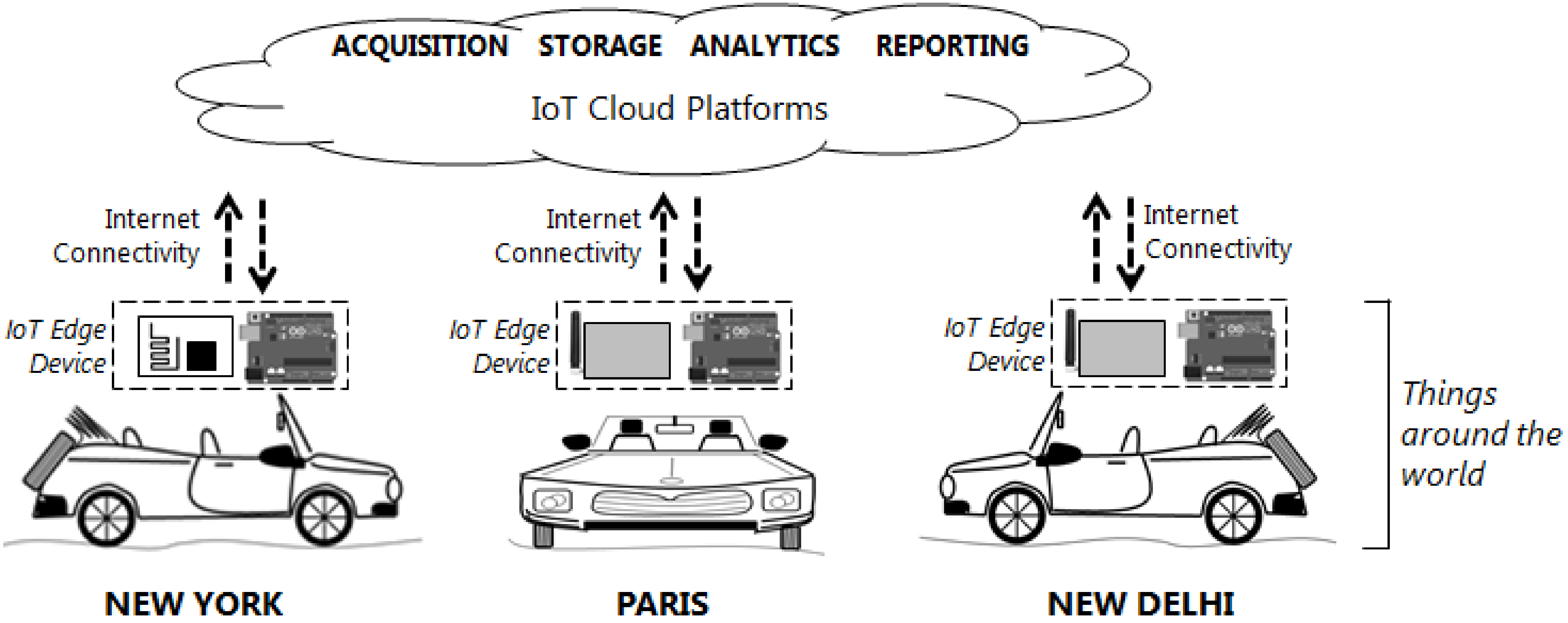
Lasse Berntzen

University of South-Eastern Norway

Internet of Things (IoT) Chapter #12

- IoT concept
- IoT edge devices
- IoT cloud platforms
- Connectivity with ESP8266 Wi-Fi chip
- Smart retail project inspiration

The concept of IoT



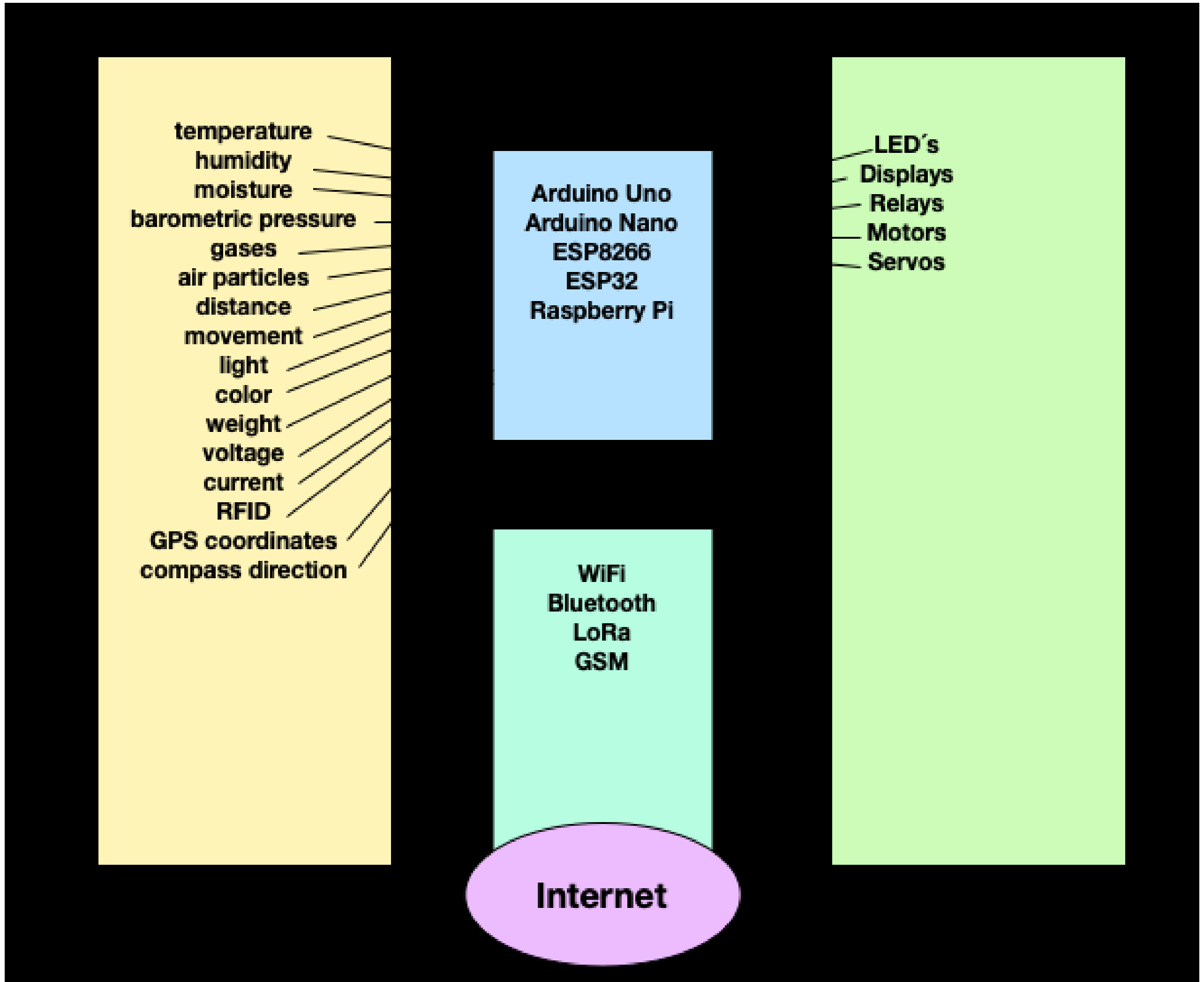
Internet-of-Things - Revisited

Microcontrollers
Processes input to make output

Sensors
Collect input data

Actuators
Makes things happen

Communication
Communicates results to other processing units



IoT edge devices

- **Interfacing with connected machines:** Edge device (ESP8266) is a bridge between the existing machine (Arduino) and the IoT cloud. (ESP8266 can also operate on its own)
- **Reading parameters or signals from connected machines:** Edge device has custom code (for example, Arduino sketches). Bridge device has an interface to connect to the Internet.
- **Connecting to the internet:** The bridge device will connect through some communication channel: WiFi, wired LAN, GSM, Radio link

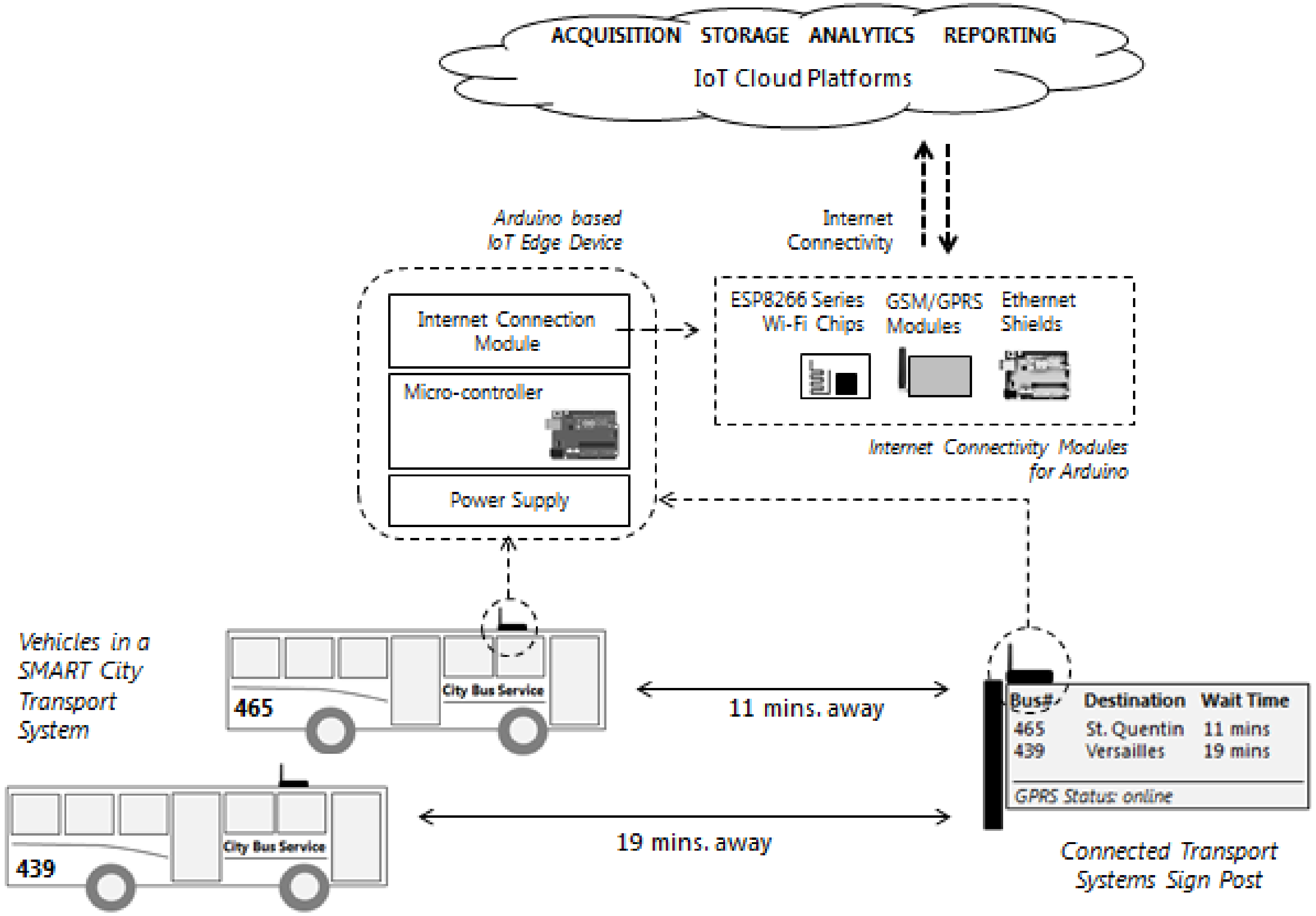
IoT edge devices

- **Transmitting data to an IoT cloud:** Edge device sends data to and receives data from the IoT cloud. The IoT cloud provider often has API or a library to make it easy to connect. Fetching data from an IoT cloud: Similarly, the edge device also runs embedded software that provides functionality to read data from IoT cloud platforms.
- **Hosting local endpoints for receiving requests from an IoT cloud:** A web page, or an app.
  - *Messaging Queue Telemetry Transport (MQTT) protoco*l.

IoT edge devices

- **Sending signals and controlling the connected machines:** The edge devices are also responsible for sending signals and controlling the physical machine that is being connected to the internet:

IoT edge device prototype



Bus#	Destination	Wait Time
465	St. Quentin	11 mins
439	Versailles	19 mins

GPRS Status: online

Two fundamental parts

- A microprocessor or microcontroller unit (which in the case of our study will be the Arduino) that serves as the brain of the edge device. In actual practice, field grade microprocessor units are used for industrializing the actual edge device product.
- An internet connectivity module for facilitating internet connections. Some popular varieties include the GPRS/GSM module, Wi-Fi chips, and Ethernet shields. For learning the basics, we will use the ESP8266 Wi-Fi module with the Arduino platform.

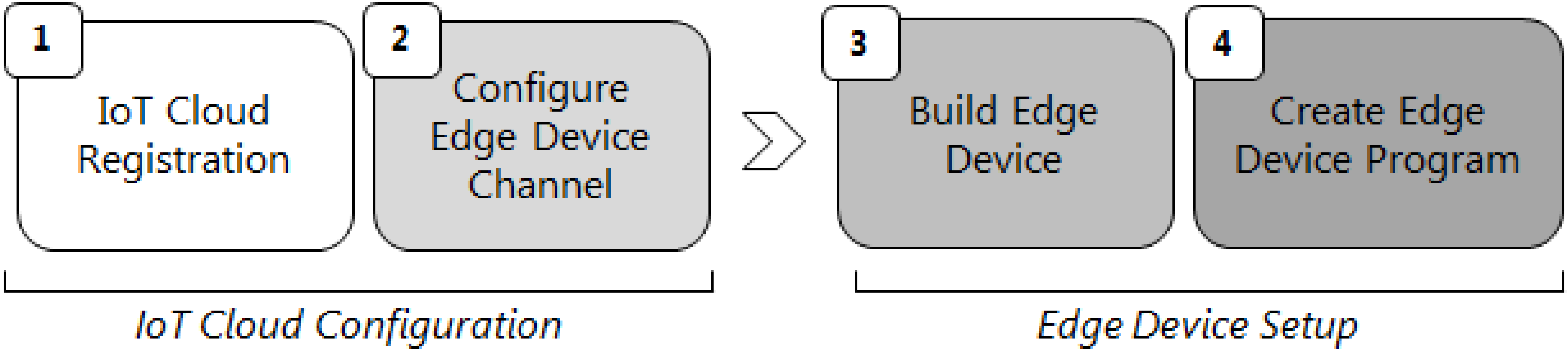
IoT cloud platforms

- Web-based administrative console for managing the connection details of edge devices. Device-specific connection points are also known as channels, while there are also generalized messaging queues to which any device can post data.
- Cloud-based backend databases/stores/lakes for storing the incoming device data.
- Cloud-based analytics services to operate upon the incoming data (also known as stream analytics) and transform the data into more meaningful desirable forms.
- To cap it all, some advanced reporting functionality is also provided by the cloud platforms.

IoT cloud platforms

- Azure IoT Hub/Event Hub from Microsoft
- AWS IoT from Amazon
- Google IoT
- IBM BlueMix IoT
- IBM Watson IoT
- **ThingSpeak IoT**
- Thingworx IoT
- Particle IoT
- Dweet IoT

Steps for IoT prototyping



1 IoT Cloud Registration

2 Configure Edge Device Channel

3 Build Edge Device

4 Create Edge Device Program

IoT Cloud Configuration

Edge Device Setup

Using ThingSpeak

1. The first step is to create an account.
2. Sign-up at https://thingspeak.com
3. After signing up on ThingSpeak, go ahead and log in.
4. Login at https://thingspeak.com

Video: L9_ThingSpeak

ESP8266

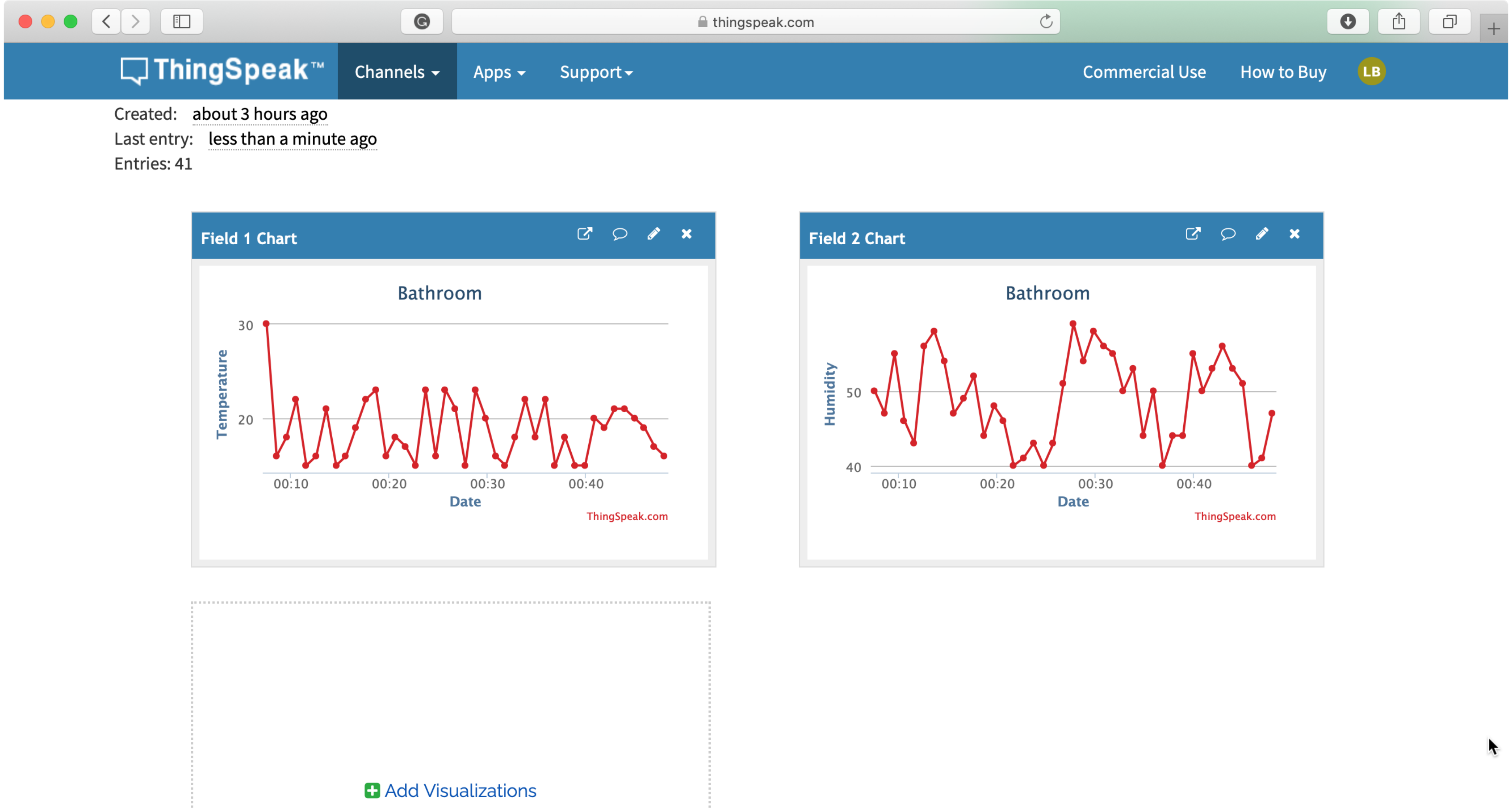
- We will use the example sketch BasicHttpClient
- The BasicHttpsClient does not work, drops the connection
- In this demonstration we will simulate a sensor using the random function, just to get a variation of values

```
delay(55000); // Every minute

t = random(15,25);

delay(5000); // Need the separation of values

h = random(40,60);
```

Sketch

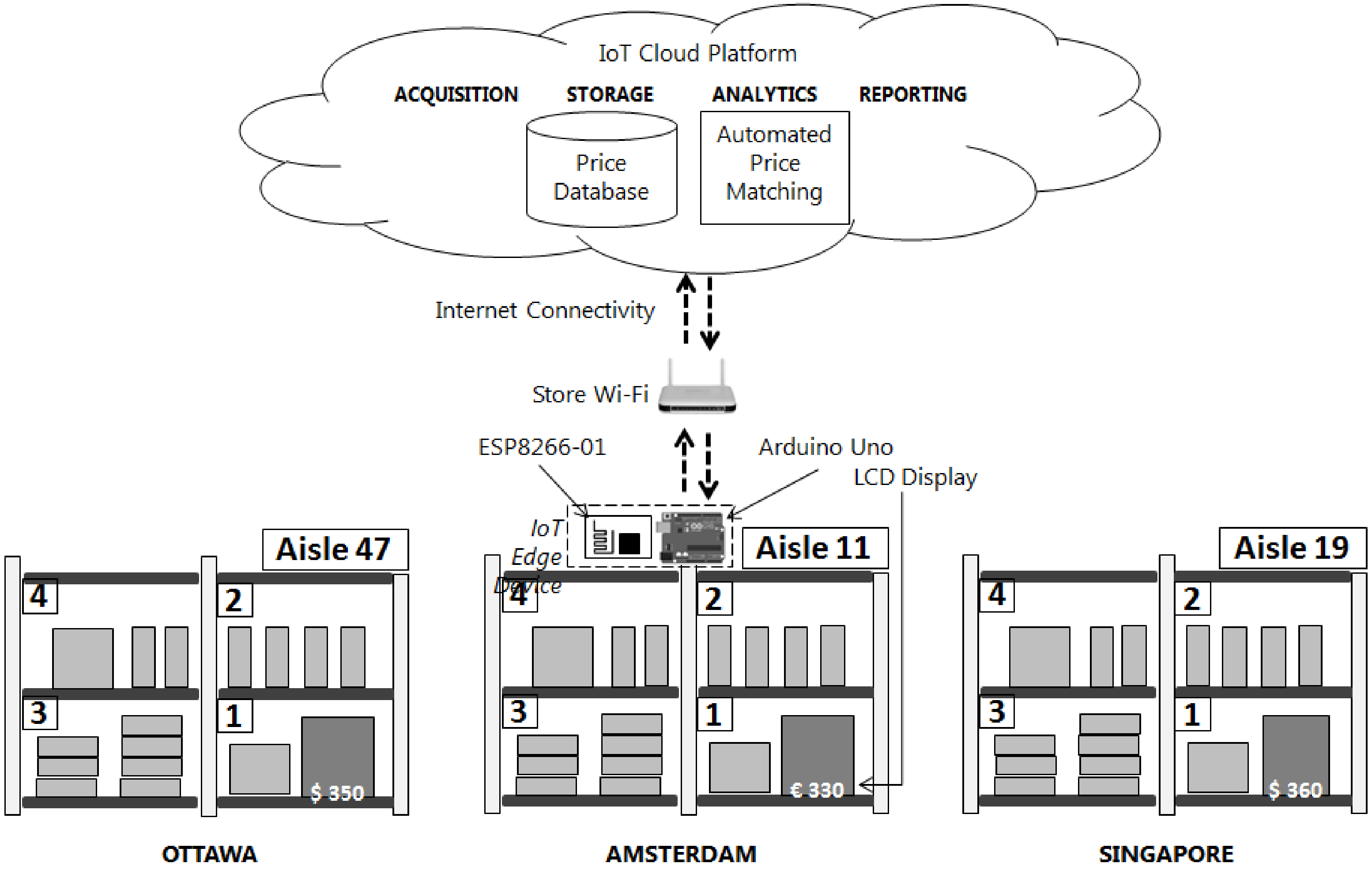
- The complete sketch is available as: L9_ThingSpeak.ino

The Result



Created: about 3 hours ago
Last entry: less than a minute ago
Entries: 41

IoT Cloud Platform



Remember

- Avoid batteries (unless unavoidable) as this will increase operational overhead of tracking and replacing. Instead, provide good quality power adapters (AC -> DC 3.3V, 60 - 170 mA) for powering the ESP modules.
- The ESP series Wi-Fi/GSM/GPRS/RF modules need to be powered from a stable battery power source. If electricity mains are not available, then use a rechargeable battery via Solar Panels.
- Battery lifetime enhancement via device/embedded software design: If possible, the Arduino boards and any Wi-Fi/GSM/GPRS/RF chips should be put to sleep mode to conserve energy.

Remember

- Devise a proper enclosure for the "Thing". Remember the 3D printing technique discussed in Chapter 6, Day 4 - Building a Standalone Device.
- Weather proofing is very important for enclosing remote devices. A suitable weather proof enclosure needs to be procured.

Remember

- This is another major area of concern for outdoor locations and has to be dealt with in a conventional manner for ensuring physical security.
- Device identity/global positioning: In order to identify the location of a remote edge device and validate its identity, you will have to attach its identity either while flashing the embedded code, or use an SD card module with a public/private key to be used while communicating with the cloud.