Proposed Initial Nomenclature for the Internet of Things

By Jeff Stollman

The Internet of Things (IoT) presents such a broad panoply of devices and capabilities that we need to establish some consistent nomenclature if we ever hope to have fruitful discussions on managing the various aspects of identity, privacy, and security necessary to facilitate its success.

This paper is a starting point to define some basic concepts that we can use. It focuses on three areas:

1. device type (Ingo: maybe better to call it instance – because e.g. a processor application is not a device)
2. device ownership (+ identity relationships) because device might have an owner/user/admin
3. device accessibility

These three areas are by no means exhaustive of the topic. The intention is to use this starter set and continuously supplement it as more concepts demand definition.

# 1. Device Type



Figure 1: Devices that comoprise the Internet of Things fall into three basic categories, plus various combinations. (The processor picture looks like an old phone case…I think we need a better one )

Device types fall into the following categories:

1. Sensor (Data collector)
2. Data Processor / Processor Application
3. Actuator
4. Gateway
5. Combination

## 1.1 Sensor (data collector)

A sensor is any device which detects and reports physical phenomena. Not all measuring devices are sensors. In order to qualify as a sensor the device must also report its findings. Reporting could take the form of allowing interrogation of results stored in memory, the broadcasting of results, or open/closing a switch to allow current to flow or cease flowing. Examples include the following:

* a thermometer the measures temperature
* a proximity switch which determines when something is within a fixed range of the sensor
* a strain gauge which measures the torque on the deck of a bridge
* a NOx detector
* an automobile speed detector.

## 1.2 Data Processor / Processor Application

A Data Processor transforms input data (typically from Sensors), applies policies (e.g., access management), and either reports the results or uses them directly to manage an actuator. A data processor could be a device located close to the sensor or it could be an application hosted on a dedicated server or somewhere in the cloud.

For example: A Data Processor may receive input from a Sensor measuring the angle of the sun in order to calculate the optimal position of a solar array. It might then send instructions to an Actuator to turn the array towards the sun.

## 1.3 Actuator

An Actuator is a device that causes a physical action when prompted by a Data Processor, or directly by a human. Actions could include opening/closing a circuit, starting/stopping a motor, sounding an alarm, or detonating (;-)an explosion. Actuators can do more than merely start or stop an action. They could also cause a vehicle to change speed. Typically, however, Actuators are digital on/off devices and variable control is provided by incorporating a Data Processor to manage the amount of time the Actuator is on or off.

## 1.4 Gateway

A Gateway is a device that is typically connected with several Sensors and/or Actuators especially when the Processor Application is located in the cloud. So there is just one remote connection needed between the Processor Application and the Gateway. Therefore the Gateway might have special capabilities for Transmission and protocol adaption.

## 1.5 Combination

Many devices will be combinations of these types.

For example, a "smart" thermostat may receive input from a built-in temperature Sensor. The input will be transmitted to a Data Processor which determines if the current temperature is within the range desired by the owner of the office, apartment, home, etc. If the temperature is outside the acceptable range, an Actuator will turn on the heating or cooling system. If the heating is managed by a smart application located on a server of facility management company the information goes over the network. A local Gateway connects all Sensors and Actuators and manages the communication with the remote server.

The GPS in your cell phone or automobile consists of Sensors that detect the time and location of signals transmitted from various GPS satellites and a Data Processor that then calculates a position from these data. The GPS satellites, themselves, consist of Sensors to detect their position, Data Processors which determine what data to be transmitted, and Actuators that transmit the data dictated by the Data Processor.

Combination devices likely will need to have each of their separate device-type systems treated separately for policy purposes -- even if all the componentry resides on a single chip. This is because different policies may apply to each component

# 2. Device Ownership / Relationships

For the time being, we imagine that device ownership falls under the following rubricks:

(To me it doesn’t matter whether the owner is an individual/or a company. Ownership implies certain responsibility. The owner is responsible for costs/ latest virus checks etc.and the owner has the right to set policies etc.

Example: A university runs a public weather station where cars/companies/everyone can ask for current local temperature. The head of the university is entitled to set the rules for using the sensors because he’s the owner

Then there is an administrator that is entitled to manage the device (it could be the same person like the owner). In our university example it is an employee of the university. He does all the necessary maintenance, virus checks etc.

The “normal user” has just the right to see the current temperature:

 That’s why I propose new categories)

1. Owner (individual/company/community/authority etc.)
2. Administrator / Super user
3. User

There might be a

1. Individual ownership
2. Private, Non-public ownership
3. Public ownership

## 2.1 Individual ownership

Under Individual ownership, there is no ambiguity over who owns and controls the device, be it a Sensor, a Transmitter, a Data Processor, or an Actuator. The device belongs to a particular human --either to Joe or to Sally. No one else has a claim of ownership of the device. While Joe or Sally may delegate control of the device, the device remains their property and responsibility. It may be viewed similarly to owning a computer or an automobile (in jurisdictions where the owner insures a vehicle, regardless of who the driver is).

## 2.2 Private, Non-public ownership

In Private, Ownership the device is owned and managed by a defined set of humans. This could be a family, a club, a corporation, or a government entity. In Private, Non-public ownership, those able to manage the device are an identifiable group. This would include a smart thermostat in a family home which is owned jointly by husband and wife, or a set of weather sensors owned by a club or corporation.

## 2.3 Public ownership

In Public ownership, a device has been placed in the public domain and no longer has an owner. **[Example needed.]**

[NOTE: Device ownership is a sticky topic. In general, device ownership may be more of a continuum than a set of discreet classifications. In these definitions, we conflate ownership with control -- acknowledging that control can be delegated by the owner without relinquishing ownership. it may turn out that these two topics (ownership and control) need to be kept separate, or that management control is the critical factor, not ownership. This may depend on how liability is assessed. As with auto insurance, some jurisdictions insure the driver, others the auto. If only the driver is insured, then liability could fall on someone delegated to manage a device. If the auto (device) is insured, then it does not matter who manages the device, liability falls on the device owner.]

# 3. Device Accessibility

Device accessibility describes the ability of entities to identify a device and/or utilize the device over the network on which the device resides. This protocol may also provide information regarding the device's capabilities. Device accessibility falls into the following categories:

1. Discoverable
2. Not Discoverable

## 3.1 Discoverable

A Discoverable device uses a published protocol and discovery mechanisms to identify itself (e.g., Internet Protocol (IP) and the Domain Name Service (DNS) for computers on the internet or Integrated Services Digital Network (ISDN) for the international telephone network) to facilitate easy access.

It should be noted that just because a device is Discoverable (i.e., uses a published protocol) does not imply that the device is available to anyone for their use. Various access control policies may restrict the ability to communicate with the device. Alternatively, a discoverable device could be placed on a private network in which the network, itself, is not accessible to the general public, such as a Local Area Network (LAN) dedicated only to manufacturing equipment within a factory.

## 3.2 Not-Discoverable

A Not-Discoverable device uses a non-published protocol for network communication or it is simply not connected to a public network. As such, it is not discoverable. Mostly groups of devices are just connect among themselves. Various security devices (e.g., motion detectors, alarm systems) may operate on their own private network using a non-published protocol in order to resist being compromised by adversaries.