# Context Awareness

## Introduction:

Context is any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects.

Context can be defined as any information that can be used to characterize the situation of an entity. An entity is a person, place, piece of software, software service or object that is considered relevant to the interaction between a user and an application, including the user and applications.

Context-awareness can be defined as the ability of a system to provide relevant information or services to users using context information where relevance depends on the user’s task

Context-aware systems are concerned with the acquisition of context (e.g. using sensors to perceive a situation), the abstraction and understanding of context (e.g. matching a perceived sensory stimulus to a context), and application behavior based on the recognized context (e.g. triggering actions based on context). As the user's activity and location are crucial for many applications, context awareness has been focused more deeply in the research fields of location awareness and activity recognition.

## Context Awareness Overview:

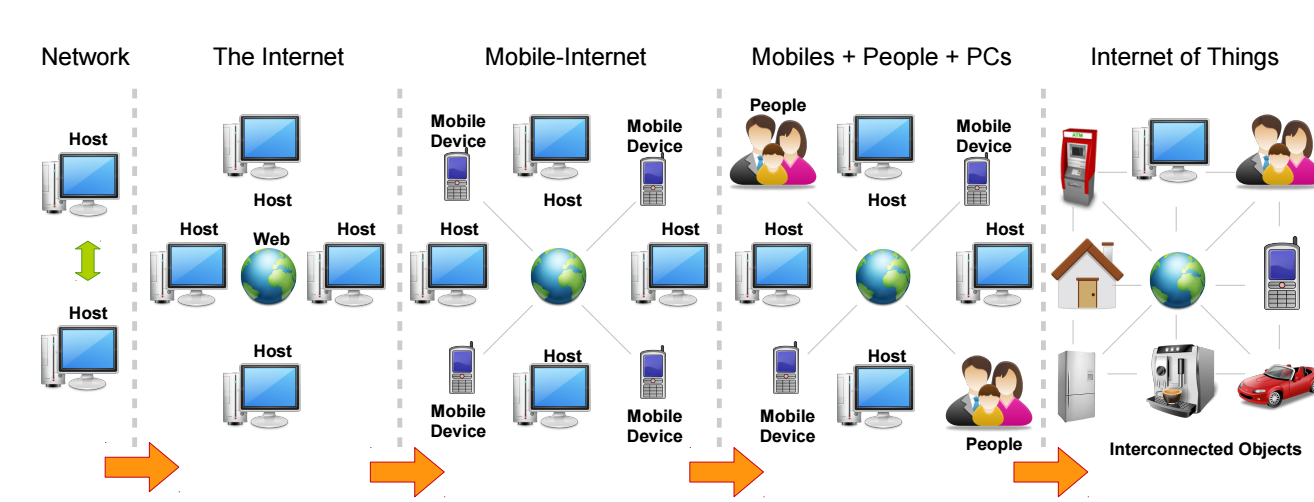
Context awareness plays an important role in the software architectures to enable services customization according to the current situation with minimal human intervention. Acquiring, analyzing, and interpreting relevant context information regarding the user will be a key ingredient to create a whole new range of smart entertainment and business applications that are more supportive to the user. Although context-aware systems have been in the research epicenter for more than a decade the ability to convey and select the most appropriate information to achieve nonintrusive behavior on multiuser-converged service platforms in mobile and heterogeneous environments remains a significant management challenge. Interoperability at the scale of the Internet of Things should go beyond syntactical interfaces and requires the sharing of common semantics across all software architectures. It also demands a seamless integration of existing computational artifacts (hardware and software) and communication infrastructures. The Internet of Things Only then can context information be successfully shared between highly adaptive services across heterogeneous devices on large-scale networks that consider this information relevant for their purposes.

The notion of context is widely understood in the pervasive and ubiquitous computing domain as relevant information referring to the situation and circumstances in which a computational artifact is embedded. As such, context awareness is the ability to detect and respond to contextual changes. The goal of context-aware computing is to gather and utilize information to positively affect the provisioning of services that are considered appropriate for a particular person or device. Therefore, context information can only be considered useful if it can be interpreted. As context is a rather vague concept, we first mention how context has been defined by leading experts in the field before continuing to describe how context can be modeled, acquired, and used to achieve autonomous and nonintrusive behavior in a service-oriented architecture.

## Context Awareness and IoT:

What It Means for IoT Analytics? The IoT’s real value will be realized only when we can derive physical contexts from the sensor data gathered. For sensors deployed on physical objects, this means answering questions such as what happened, where, when, and why. For sensors deployed on humans, the context means answering questions such as who is doing what, in where, or who is feeling what and why. Thus, context-awareness in the IoT ultimately boils down to using analytics to detect a physical event, cause, identity, location, activity, physiology, and psychology from sensed data.

## Evolution Of Context Aware Technologies:



In the early phase of computer networking when computers were connected to each other in point-to-point fashion, context-aware functionalities were not widely used. Providing help to users based on the context (of the application currently open) was one of the fundamental context-aware interactions provided in early computer applications and operating systems. Another popular use of context is context-aware menus that help users to perform tasks tailored to each situation in a given application.

When the Internet came into being, location information started to become critical context information. Location information (retrieved through IP addresses) was used by services offered over the Internet in order to provide location aware customization to users.

Once the mobile devices (phones and tablets) became a popular and integral part of everyday life, context information collected from sensors built-in to the devices (e.g. accelerometer, gravity, gyroscope, GPS, linear accelerometer, and rotation vector, orientation, geomagnetic field, and proximity, and light, pressure, humidity and temperature) were used to provide context-aware functionality. For example, built-in sensors are used to determine user activities, environmental monitoring, health and well-being, location and so on.

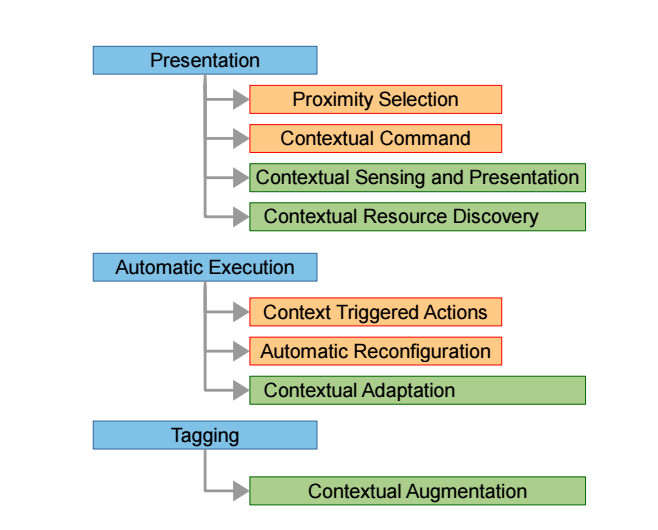
Over the last few years social networking has become popular and widely used. Context information gathered through social networking services (e.g. Facebook, Myspace, Twitter, and Foursquare) has been fused with the other context information retrieved through mobile devices to build novel context-aware applications such as activity predictions, recommendations, and personal assistance.

For example, a mobile application may offer context-aware functionalities by fusing location information retrieved from mobile phones and recent ‘likes’ retrieved from social media sites to recommend nearby restaurants that a user might like. In the next phase, ‘things’ were connected to the Internet by creating the IoT paradigm.

## A Definition of Context

Schilit and Theimer first introduced the term context awareness, also called sentient, in 1994. Later, Ryan defined it in both cases; the focus was on computer applications and systems. As stated by Abowd those definitions are too specific and cannot be used to identify whether a given system is a context-aware system or not.

After analyzing and comparing the two previous efforts conducted by Schilit and Pascoe, Abowd identified three features that a context aware application can support: presentation, execution, and tagging. Even though, the IoT vision was not known at the time these features are identified, they are highly applicable to the IoT paradigm as well. We elaborate these features from an IoT perspective.

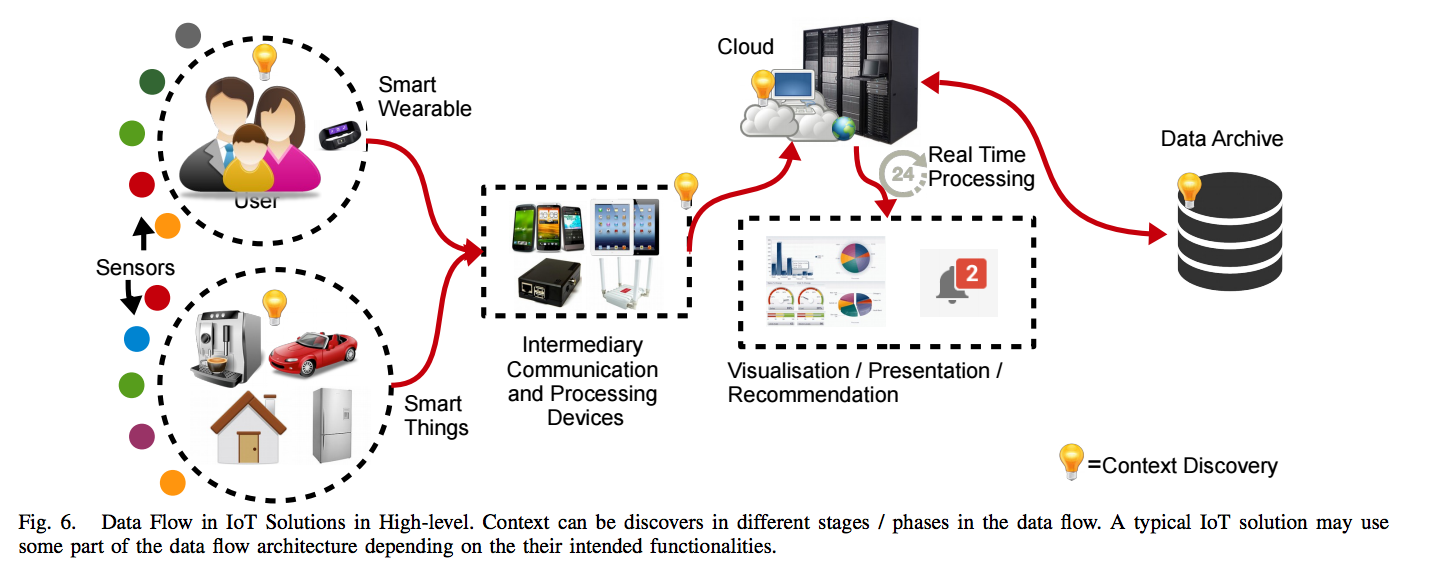


**Presentation:** Context can be used to decide what information and services need to be presented to the user. Let us consider a smart environment scenario. When a user enters a supermarket and takes their smart phone out, what they want to see is their shopping list. Context aware mobile applications need to connect to kitchen appliances such as a smart refrigerator in the home to retrieve the shopping list and present it to the user. This provides the idea of presenting information based on context such as location, time, etc. By definition, IoT promises to provide any service anytime, anyplace, with anything and anyone, ideally using any path/network.

**Execution:** Automatic execution of services is also a critical feature in the IoT paradigm. Let us consider a smart home environment. When a user starts driving home from their office, the IoT application employed in the house should switch on the air condition system and switch on the coffee machine to be ready to use by the time the user steps into their house. These actions need to be taken automatically based on the context. Machine-to machine communication is a significant part of the IoT.

**Tagging:** In the IoT paradigm, there will be a large number of sensors attached to everyday objects. These objects will produce large volumes of sensor data that has to be collected, analyzed, fused and interpreted. Sensor data produced by a single sensor will not provide the necessary information that can be used to fully understand the situation. Therefore, sensor data collected through multiple sensors needs to be fused together. In order to accomplish the sensor data fusion task, context needs to be collected. Context needs to be tagged together with the sensor data to be processed and understood later. Context annotation plays a significant role in context aware computing research. The tagging operation also identified as annotation.

## Evaluation Framework



The Figure visualizes how data is being collected transferred, processed, context discovered and annotated in typical IoT solutions. It is important to note that not all solutions may use the exact same data flow. Each solution may use part of the architecture in their solution.

## Strategies Used By IoT Products to offer Context Aware Functionality:

**Context Aware Tagging Section**

Context-aware tagging, which is also called context augmentation and annotation represent the idea of sensing the environment and collecting primary context information. We also believe that secondary context generation is also a part of context-aware tagging feature. Primary context is any information retrieved without using existing context and without performing any kind of sensor data fusion operations.

**Context Selection and Presentation Section:**

There are number of commonly used strategies, by most of the IoT solutions in the marketplace, to present context to the users. Most of the IoT products use some kind of visualization techniques to present context information the users. We call this visual presentation.

## Context Awareness Sensors:

## RFID:

RFID is an emerging technology for embedding sensing capabilities in everyday objects and is gaining momentum as a popular means for automatic identification and tracking in supply chain management. Each tag contains a unique ID number that can be read by an RFID reader. Active RFID tags have their own power supply to transmit their signal, while passive ones use the electrical current induced in the antenna during reception of the incoming radio frequency signal emitted by the RFID reader. The presence of an internal power supply helps to extend the range of operation and the amount of information that can be transmitted. The majority of passive tags typically have anywhere from 64 or 96 bits to 1 KB of nonvolatile EEPROM memory, while active tags have battery-backed memories as high as 128 KB and more. The ability to store and remotely recognize tags at a high pace (in the order of hundreds per second) make RFID a promising technology for identification and locating purposes in context-aware and pervasive computing.

RFID tags are mainly used for asset tracking and in inventory systems at libraries and shopping malls where they replace the older barcode technology. However, RFID technology is also being applied to the tagging of humans to identify them as well as locate their whereabouts. RFID enabled E-passports are issued by many countries, while implantable RFID chips are used to track patients in a hospital and access their medical records. As such, it is clear that RFID provides added value to the domain of context-aware computing and to the Internet of Things paradigm in general for sensing identity and location.

## IR Tags:

IR tag readers and IR active tags that identify indoor locations and that recognize the user’s circumstances, such as sitting in a meeting room, sitting in the driver’s seat of a car, etc., microphone sensors that recognize the user’s voice, and GPS receivers that provide outdoor location information.

## Internet Connected Sensors:

The Industrial Internet of Things (IIoT) is using contextual awareness to open new frontiers for improving processes. Industries such as chemical production are installing Internet-connected sensors to bring greater granularity to monitoring. These sensors feed data to process control systems (and sometimes to the cloud) after which the data is analyzed and signals are sent to actuators that adjust and optimize processes, for example by modifying ingredient mixtures, temperatures, or pressures.

## Position Sensors:

Position sensors in IoT applications provide access to location.  A rotary-position sensor, for example, translates an angular mechanical position to an electrical signal and is used in applications where it is necessary to control a variable output such as frequency and speed

## Requirements for Representing and Exchanging Context

For humans, it is natural to communicate and interpret context. The goal of using open context specification languages is to simplify the capturing, transmission, and interchangeability of this context between systems. Different modeling approaches for context have been investigated in the past. A model consisting of key-value pairs is the simplest approach to represent context, but lacks the ability to structure information and only supports exact matching. Markup scheme models typically introduce a fixed structure and allow expressing more complex relations, such as associations. Ontologies are well known in the knowledge representation community to model concepts and the relationships that hold among them and their semantic interpretation is universally accepted.

## 

Following is a review of some of the main requirements for an open context specification to be used for exchange between computational artifacts in the Internet of Things:

**Comprehensive domain coverage and terminology:**

An open context specification language should make available a terminology that provides appropriate coverage and a comprehensive representation of a domain in order to model most of the concepts and terms needed for describing entities in a particular domain. Concepts may be modeled with multiple synonymous representations and may have hierarchical relationships to other concepts.

**Semantic no ambiguity and expressiveness:**

Information semantics involve the uniform interpretation of a concept. This requires a strict support for no ambiguity in the specification language to ensure that each concept in the terminology has only one meaning. Semantic expressiveness is the ability to easily enhance the knowledge domain using the semantic primitives of the specification language. More advanced semantic specification languages for context provide support to model specialization and inheritance relationships, aggregation, dependencies and constraints, etc.

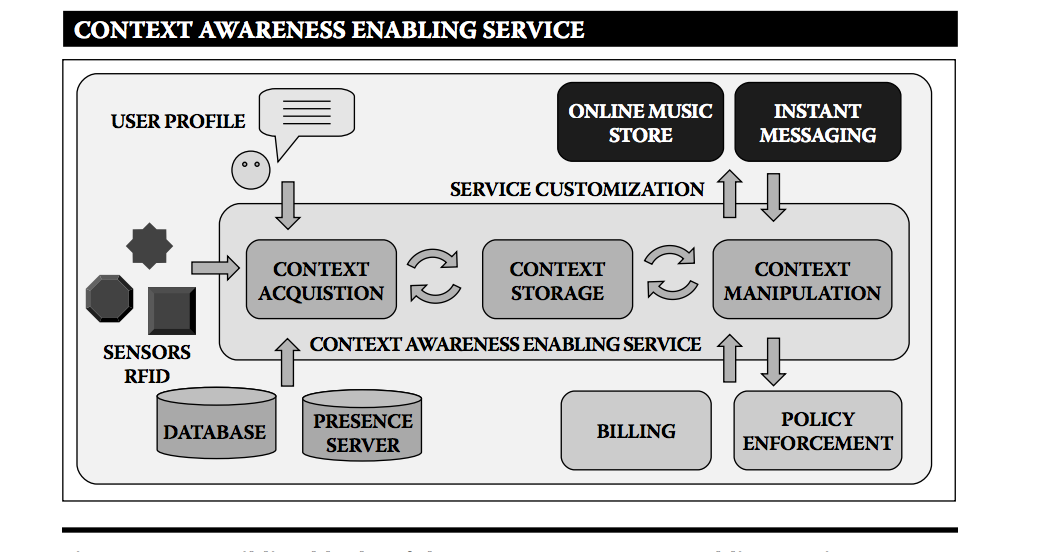
**Processing complexity:**

Some languages require quite complex processing steps for both the data acquisition and processing (e.g., reasoning and inference) parts. For example, context models using Resource Description Framework (RDF) [4] or OWL [20] may refer to other remotely available context models to detail the semantics and relationships of concepts being reused. This, of course, is a good thing as it improves knowledge sharing, interoperability, and universal interpretation. However, it also increases the complexity to process context models. As the Internet of Things targets computational artifacts of different shapes and sizes, an appropriate balance should be found between processing complexity and language expressiveness.

**Specification language interoperability:**

There is no one single specification that will solve all issues with respect to context provisioning in an open fashion. For some purposes, the use of a particular domain-specific language may already be so pervasive that this in itself would be one of the reasons not to implement the same concepts in another more generic specification language. Therefore, the ease of reusing and integrating different dedicated specification languages is also an important concern.

## Context Storing And Distribution:



After having consolidated context information from various acquisition systems (sensors, people, devices, etc.) and remote input channels (services, databases, etc.), the collected information may be selected for distribution to remote subscribers (people, services, devices). When information is gathered from or needs to be delivered to a remote entity, choices need to be made on where context will be stored. The obvious choice would be close to the area where it is considered relevant, and perhaps be duplicated if necessary. If context is stored locally on the device that sensed the information, it is readily available and round trips on the network to retrieve the information are avoided. However, if the information is also useful for other parties, then it makes sense to store the information remotely on a repository that is more easily accessible. The following concerns are taken into account whether information should be stored locally or remotely:

**Scope of relevance:** Certain information can be relevant only for computational artifacts that interact in particular with local services.

**Information sensitivity:** Tracking information retrieved from RFID tags could be of interest to multiple parties on the network and be made available on a remote server for further analysis.

**Information correctness and period of relevance:** Static and profiled information is likely to remain valid for longer periods. Volatile information on the other hand, such as the current system load, may no longer be accurate or relevant when it is finally remotely stored in a repository.

**Caching for information reachability:** If some sensor information can be obtained quite easily, then storing the information may not be worthwhile. However, as some appliances might have no direct connection to context sensors that provide useful information, a repository with standard SOAP or SIP interfaces can collect and make the information accessible to all interested parties.

**Exploiting information history:** In some cases, an appliance might want to exploit the historic values of a certain context attribute in order to derive new information.

## Context Awareness and Developing Analytical Applications

In the diverse IoT world, we can no longer afford to build vertical applications from scratch—a platform-based development approach promoting software re-use via APIs and services is a must. Going further, one of the fundamental game-changers for successful IoT adoption will be the democratization of knowledge derived from IoT data. To make this happen, we need a crowd sourced application development and consumption ecosystem. Developing analytical applications for the IoT is a complex process that needs diverse knowledge of domains, sensors, algorithms, programming, and deployment infrastructure.

A viable way to address such complexity is through a **model-driven development (MDD) framework**. MDD is an approach that aims to model knowledge across different stakeholders (such as sensor providers, algorithm providers, domain experts, and infrastructure providers) by allowing separation of concerns for each stakeholder. It can assist an IoT application developer in easily creating an application based on data and goal descriptions. Domain-agnostic semantic data interoperation will also play a big role in such a framework.

The return on investment for major IoT deployments isn’t sufficiently motivating at this stage for most businesses. This is a primary reason why IoT applications have rarely moved beyond pilot deployments. There are deployment and cost issues when it comes to installing multitudes of sensing hardware across physical spaces. To make it more cost effective, enterprises are looking into **crowdsourcing sensing data** from mobile phones where appropriate. Mobile phones are already pervasive, and they come with rich sensors such as cameras, microphones, accelerometers, gyroscopes, magnetometers, GPS, and altimeters. Participatory and opportunistic sensing using mobile phones will play a key role in IoT deployment. A plethora of applications are possible using mobile phone sensors, prominent among them being road condition monitoring, driving behavior analysis, traffic monitoring, wellness/health monitoring, and so on. In the future, robots and unmanned aerial vehicles carrying sensors will also contribute to this pervasive, affordable sensing paradigm

Using the intelligence of a vast interconnected organism, however, is nothing new: the venerable Oxford English Dictionary may in fact be the earliest example of crowdsourcing. From avoiding traffic jams, to analyzing pedestrian flow patterns, to finding the best public toilet in town, crowdsourcing apps are showing that many smartphones make for light work.

Today, crowdsourcing is used in investing, in creative work and in funding start up projects. The wide availability of smartphones now makes it easier than ever to devote them to data gathering, with or without actual human intervention.

Crowdsourcing apps have been quick to gather pace in the U.S. and Europe -- where dealing in a marketplace with strangers is commonplace -- but slower in Asia where connections are still important.

## Conclusion:

The Internet of Things is all about convergence and integration of the latest advancements in the research areas of software and hardware with industrial technologies invented many decades ago. Research carried out in the context-aware computing domain aims to create a pleasant user experience at the workplace, in public areas as well as in the home environment by simplifying human interactions with everyday services and making them less intrusive. We have discussed how RFID technology can contribute to context awareness as a way to identify and locate everyday objects as well as humans, for that matter.