# THE UBISENSE SMART SPACE PLATFORM

Pete Steggles, Stephan Gschwind[1]

***Abstract.*** *Ubisense has developed a platform for building Smart Space applications. The platform addresses the key requirements for building Smart Spaces: accurate 3D positioning, scalable real-time performance, development and deployment tools. This paper deepens the key requirements and describes how the Ubisense platform components meets them. The demonstration exemplifies the smart space platform by tracking players in a game application.*

## The Ubisense Smart Space Platform

Ubisense has developed a platform for building Smart Space applications. Our platform addresses the key requirements for building Smart Spaces:

- **Accurate 3D positioning** supports applications that can perceive the physical relationships between people and objects in the environment
- **Scalable real-time performance** enables arbitrary numbers of applications, used by arbitrary numbers of people over an arbitrarily-large area
- **Development and deployment tools** make it easy to design, implement, and manage Smart Space applications.

The demonstration shows a Smart Space containing applications that integrate with external software (a simple game that users control by moving around), and devices (a PTZ camera that keeps users in shot while they are playing the game)

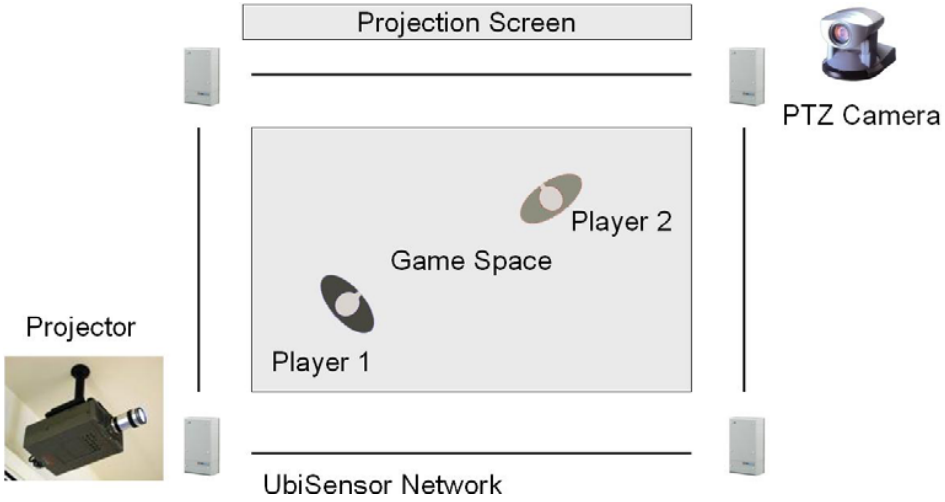


**Fig. 1 Smart space demonstration setup**

[1] Ubisense, St Andrews House, 90 St Andrews Road, Chesterton, CB4 1DL, United Kingdom, http://www.ubisense.net

This paper describes the components of the Ubisense platform and how they relate to the requirements above.

## Accurate 3D Positioning

The Ubisense location system uses ultrawideband (UWB) radio. UWB was chosen because it has a combination of attractive properties for in-building location systems:

- It is a non-line-of-sight technology with a range in free space of a few tens of metres, which makes it practical to cover large indoor areas
- It is easy to filter the signal to minimise the multipath distortions that are the main cause of inaccuracy in RF indoor location systems

This combination of properties means that UWB has the best cost-to-performance ratio of all available indoor location technologies.

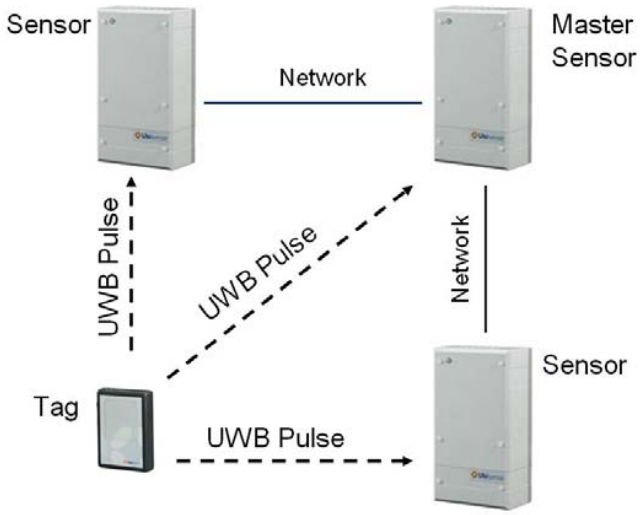


**Fig. 2 Ubisense hardware components**

The Ubisense location system consists of these components:

- A network of Ubisensors, that are fixed in known positions throughout the area to be covered, and networked using standard Ethernet. Each Ubisensor has a conventional RF transceiver, and an phased array of UWB receivers.
- A set of Ubitags, that are carried by people and attached to objects. Each Ubitag has a conventional RF transceiver, and a UWB transmitter.

The Ubisensors are organised into cells, typically composed of four to seven sensors, so that each cell covers a given area. The conventional RF and the UWB channels each have several colours, allowing the cells to be tiled to cover arbitrary areas.

Each cell has one Ubisensor that functions as its master. The master coordinates a TDMA network using the conventional RF channel; each Ubitag in the area covered is allocated an appropriate schedule of slots in which it is active. The Ubitag schedule can be changed on the fly in response to application requirements, so that interesting or fast-moving Ubitags can be scheduled more frequently, and uninteresting Ubitags less frequently, conserving the UWB channel and the Ubitag

power supply. The conventional RF channel also supports bidirectional data communications between each Ubitag and the wider network, and each Ubitag is equipped with a pair of buttons and a bleeper to support control and paging applications.

When a Ubitag is active, it sends out a conventional RF message containing its identity, together with a UWB pulse sequence that is used by the Ubisensors to determine the Ubitag's location.

The Ubisensors use a combination of Time-Difference Of Arrival (TDOA) and Angle Of Arrival (AOA) techniques to determine the location of a transmitting Ubitag. Only two Ubisensor readings are required to generate a 3D position for the tag. An individual timeslot is just over 26ms duration, leading to a maximum update rate per cell of just under 39Hz, though each individual Ubitag has a maximum update rate of 10Hz. In a typical open environment, a location accuracy of about 15cm can be achieved across 95% of readings.

## Scalable Real-Time Performance

The Ubisense Enterprise software system is designed to support multiple, integrated, real-time location-aware applications that work over large areas for large numbers of users. It provides these general services:

- It manages the Ubisensor network by storing the configuration data and code for Ubisensors, controlling Ubitag handover, update rates, and data transmissions, and monitoring and reconfiguring Ubitags and Ubisensors as the system operates.
- It manages the real-time location data by applying an appropriate filtering scheme depending on the object that is being tracked, and distributing the data in a timely manner to arbitrary numbers of applications.
- It provides higher level services such as monitoring of spatial relationships between objects, extensible data modeling to support object properties, and assigning names and visualisations to objects.
- It provides a platform for integration, making it easy to create an abstract view of a device API or of sensor information

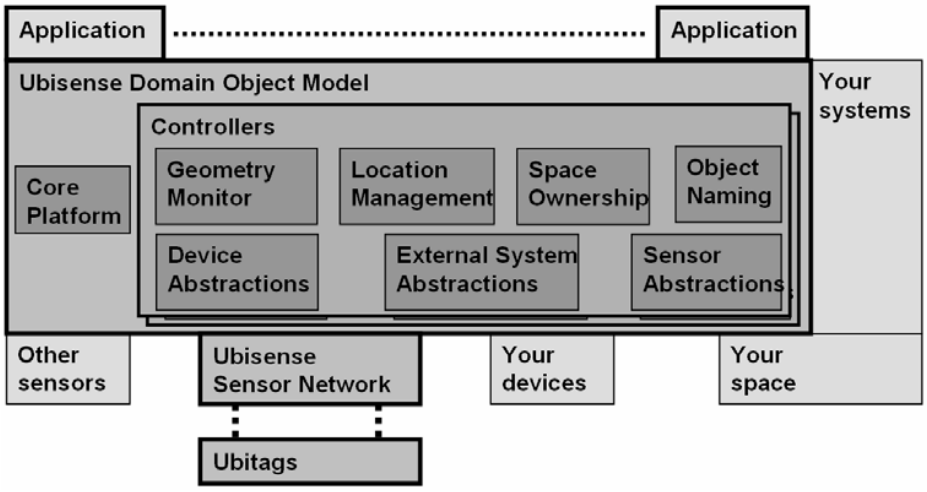


**Fig. 3 Platform architecture and features**

The Ubisense platform is designed to make optimal use of commodity processing and switched Ethernet networking hardware. It is organised into a large number of small services, each of which manages one or more in-memory relational schemas, and a set of clients with client-side caches of the schemas that they wish to use. The caches are kept up-to-date using a reliable multicast protocol, that ensures that clients receive events in a timely manner, and also means that all the data required to support a query is guaranteed to be held in a client's address space.

In order to ensure wide spatial coverage, services are clustered within a cellular hierarchy which ensures that high aggregate update rates can be achieved without requiring high bandwidths at any individual service. The hierarchy has four tiers of cell: individual Ubisensor cells; Location cells, in which sensor information is filtered and presented as abstract location data; Geometry cells, in which spatial relationships between objects are managed; and a Site cell that manages largely constant data. All services are cellular, including application services written by users, so that even application programs will scale to wide areas and large aggregate update rates.

Integration of devices and sensors can be achieved by two methods: for more straightforward tasks, many parts of the platform are accessible via APIs presented in C++ or COM; for more complex tasks where tighter integration is required, it is possible directly to use the Ubisense data modelling language used to build the rest of the Ubisense system.

## Development and Deployment Tools

The Ubisense platform provides a number of visual configuration and development tools, which drastically simplify the generation of end user applications.

- The Building Editor tool allows capturing floor plan data or importing the data from CAD tools.
- Deployment of a Ubisense sensor system is aided by the Cell Editor tool, which allows the user to plan how the cells will be organised in their building, and supports the surveying and calibration process.
- The administration of motion profiles, object types, object instances, representations etc. is supported by the Object Administration tool.
- The Visual Developer tool allows programmers to define, and automatically evaluate, spatial relationships between objects by drawing a geometric representation of the relationship
- The Simulator tool allows developers to draw scripts describing motion and behaviour of people and objects, making it easy to test Smart Space applications.
- Management (installation, backup, upgrade, migration) of the system services is supported by the Service Admin tool. This tool also manages the application-level services developed by the user, making it easy to control services even when they are running in multiple cells on a large cluster of host machines.

## References

[1] http://www.ubisense.net/Product/whitepapers&downloads.html

[2] http://www.uk.research.att.com/location