Resource Management in the Internet of Things: Clustering, Synchronisation and Software Agents

Introduction

- A number of techniques aimed at addressing the challenges arising from the increasing number of connected objects, such as limited computation and energy, unreliable wireless channels and the impossibility of ubiquitous network access, repetitive and mundane user interactions, given the complexity of the architecture.

Three major interconnected topics

- First, the grouping of objects into clusters in order to overcome scalability, energy efficiency and robustness issues,
- Secondly, the use of software agents to represent and manage objects and users, moving part of the complexity to the architecture and providing a bridge between the users and the things, and,
- Thirdly, techniques for bidirectional synchronisation of object knowledge in order to support operations and provide resilience in situations having only intermittent or unreliable network connectivity

WSN vs MANET

- The main objective of a MANET is network reliability and the accessibility of nodes. This is realised by building meshed networks without central authorities.
- The clustering approaches of WSN are more hierarchical, using CHs as decentralised authorities for realising mostly star or tree topologies.
- WSNs vary in their objectives: there are existing approaches aiming to fault tolerance, load-balancing, energy consumption, increased connectivity and reduced packet delay. While MANETs are generally built to handle objects in dynamic environments, WSN are traditionally used to cluster more or less static nodes.

Some of the properties of Adhoc and WSN

We would first like to compare those ad-hoc wireless clustering protocols that consider both mobility and energy-efficiency. The properties that we would like to compare are :

- Type
- Controlled variable CH period
- CH election according to node conditions
- Synchronisation
- Global cluster information
- Multi-hop routing
- CH election complexity

Comparison of Adhoc and WSN

Protocol	Type	Variable CH Period	Conditioned election	Synchronisation	Global information	Multi-hop	Complexity
DMAC (Basagni 1999)	MANET	No	Weight	No	Yes	No	O(n)
WCA (Chatterjee et al. 2002)	MANET	No	Weight	Yes	Yes	No	O(d+m+1) *
LIDAR (Gavalas et al. 2006)	MANET	Mobility	Energy+	No	Yes	No	O(n)
ANDA (Chiasserini et al. 2004)	MANET	No	Energy	Yes	Yes	No	O(nxc) **
Wu et al. 2001	MANET	No	Energy+	No	Yes	Yes	O(v+N[x]) ***
Liu and Lin 2005	WSN	No	Energy	No	Yes	No	O(n)
Onodera & Miyazaki 2008	WSN	No	Energy	No	No	Yes	O(n)
MoCoSo (Sanchez Lopez et al 2008)	WSN	Yes	Energy	No	No	Yes	O(y) ****

Comparison of WSN clustering protocols

Protocol	Variable CH Period	Conditioned election	Synchro-nisation	Global in-formation	Multi-hop	Complex-ity
LEACH (Heinzelman et al. 2002)	No	None	Yes	Yes	Yes	O(n)
(Liang & Yu 2005)	No	Energy	Yes	Yes	Yes	O(n)
EECS (Ye et al. 2005)	No	Prob + Energy	Yes	Yes	Yes	O(n)
EDAC (Wang et al. 2004)	Energy	Energy	Yes	Yes	Yes	O(n)
HEED (Younis & Fahmy 2004)	No	Energy	No	Yes	No	Nit x O(n) *
GESC (Dimokas et al. 2007)	No	Significance	No	Yes	No	O(n x u) + O(n) **
MoCoSo (Sanchez Lopez et al. 2008)	Energy	Energy	No	No	Yes	O(y) ***

Software Agents

Agent Based Systems are an evolving software paradigm that strives to create software that can possess human characteristics, such as autonomy, adaptability, sociality, judiciousness, mobility and reactivity.

- **Intelligent agents** are software programs that continuously perform three functions: perception of dynamic conditions in the environment; reasoning to interpret perceptions, solve problems, draw inferences, and determine actions.
- **Autonomous agents** are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so, realise a set of goals or tasks for which they are designed.

Software Agents

A software entity to be named an agent, it should maintain the following properties:

- Autonomy
- A description of the current state of its environment
- Reactivity (reflex based agent) and/or proactivity (goal/utility based agent)

Agents might also “learn” to improve their behaviour using feedback from its performance, evolve or self-replicate depending on the needs of a particular application.

Software Agents

Although there have been no attempts for agent integration within the Internet of Things, to date, software agents can greatly enhance the functionality of the core Internet of Things architecture in two ways:

- Firstly, **user centric agents** can enable the automation of user queries and alert users to any changes in specific items or trails. Users, if they wish so, can offload monitoring duties to a user agent and customise alerts to be sent to them.
- Secondly, **product centric agents** can enable the concept of intelligent autonomous products (i.e., things) to be integrated with the Internet of Things, and help bring the intelligent product concept alive, by enhancing services that the Internet of Things can offer to its users.

Software Agents

An **Intelligent Product** is the coupling of a product and an information based representation that

- possesses a unique identification,
- is capable of communicating effectively with its environment,
- can retain or store data about itself,
- deploys a language to display its features and requirements, and
- is capable of participating in, or making decisions relevant to, its own destiny

Data Synchronisation

- The data distributed in the Internet of Things can be stored in the objects themselves or in heterogeneous online repositories, and might exist in connected and/or (partially) disconnected environments. In order to maintain a coherent cross-infrastructure view of the object information, the synchronisation of data across the architecture components is necessary.

- A **distributed database** as a logically integrated collection of shared data, which is physically distributed across the nodes of a computer. In the case of the Internet of Things, these data will be additionally distributed across autonomous and heterogeneous objects, adding even more complexity to the system.

Data Synchronisation

The following requirements for distribute databases are:

- Data Handling,
- Query Optimisation,
- Concurrency Control,
- Recovery,
- Integrity and Security.

Two additional requirements are added by Öszu (1999):

- Transaction Management,
- Replication Protocols.

Data Synchronisation

- These requirements need to be met in order to support efficient, secure and consistent data synchronisation in distributed databases, and can be set as key requirements for data synchronisation in the Internet of Things as well.

- New approaches apply mobile agents without a specific master node. Such agents support distributed transactions and security tasks.

Data Synchronisation

Assis Silva and Krause (1997) describe the agent-based concept as:

- Very suitable for supporting transactions processing in massively distributed environments,
- Very suitable for supporting activities in dynamically changing environments,
- Providing an adequate support for mobile devices,
- Fulfilling coordination requirements of different types of application.

Regarding the data itself, its synchronisation involves different types of information in order to ensure data consistency:

- Object data: information describing an object,
- Security data: information supporting access control to an objects information,
- Event data: information about an objects history.

- Knowledge about the structure, syntax and semantic of object information is required to filter data that has to be synchronised.

- Due to the distributed locations of object information in the Internet of Things, the network availability between all information resources is of a special interest.

- The Internet of Things is envisaged to contain distributed heterogeneous databases, applications and services, which might be always connected, partially connected or even permanently disconnected.

Assumptions and Definitions

- The Internet of Things advocates the extension of the Internet infrastructure that we know today towards the inclusion of objects or things as information producers. For the sake of clarity, we could define these objects as manufactured items, whose information and state is relevant to some service or application that is connected to this Internet of Things and, therefore, to the users that make use of those services.

Assumptions and Definitions

We will assume that the participation level of any Internet of Things object is based at least in the following capabilities:

- A unique identity
- Ability to sense and store their condition. Condition is the status of an object obtained by interpreting the output of sensor transducers associated with it
- Ability to make their information (be it identification, condition or other attributes) available to external entities
- Ability to communicate with other objects
- Ability to take decisions about thereselves and their interactions with other objects

Assumptions and Definitions

- We will therefore assume that the communication capabilities of the devices that represent the objects are realised over the air using radio signals.

- In the context of wireless networking, an independent computing agent is generally called a "**node**". We will call the devices that represent the Internet of Things objects "nodes".

- We assume that local networks of objects can communicate with the Internet of Things infrastructure transparently, either directly with the support of IPs, or via gateways that can translate legacy protocols to the ones used on the Internet. "**infrastructure gateways**".

Assumptions and Definitions

Assumptions and Definitions

- We will also assume that objects can create networks with other objects.
- We will also refer to the clustering capabilities of these networks of objects, where clustering is a particular mechanism for organising the objects into networks.
- We will focus in a single clustered network, and will use the terms "cluster" and "network" interchangeably.

Clustering Principles in an IoT Architecture

- In order to monitor their condition, embedded devices with wireless communication capabilities could be attached to them, becoming a part of the object object, the same way a barcode sticker is part of the vast majority of today's products.

- The main differences include the lack of a standardized unique identification scheme, the assumption of static deployments, the assumption of centralised base stations and the inflexible topologies that WSN are usually constructed upon.

Clustering Principles in an IoT Architecture

- Common WSN features include multi-hop communication, cooperative applications and events triggered inside the network. These are characteristics of active (as opposed to passive) networking. A clustering design for the Internet of Things requires the use of active networking to create collaborative, multi-hop and always-dynamic interactions among objects, which are equipped with wireless embedded devices.
- Active networking also creates the possibility of extending the information of an object by using other nearby object's information to enrich its status. To maximise the potential of these attributes, it would be beneficial to design a data structure model which organises the information of all the objects.

Clustering Principles in an IoT Architecture

- Every action in which the device is involved, such as sensing or using the wireless transceiver, consumes part of its energy. For this reason, the protocols that manage node communication and networking must be carefully considered, since these devices are expected to function for months or years with the same battery charge.

- This election may consider the particular static capabilities of each node (e.g. longer radio range, computing power), as well as its current dynamic status.

Clustering Principles in an IoT Architecture

- One of the most important dynamic attributes of a node is its current residual energy. The election of the best candidate is paired with the decision of how long it will remain as the new CH.
- In the same way, the election itself is based on dynamic and static information about a node and the time that a node will have. The role of CH can be static (i.e. always the same time) or dynamic (i.e. a different time for each CH election), depending on the attributes of this specific node.

Clustering Principles in an IoT Architecture

- **Election Mechanism**

- Clustering is a general strategy with many dimensions, and the protocols that manage the clustering mechanisms have to be tailored to the specific challenges and needs of the Internet of Things architecture.

The Role of Context

- Information can arise from two different sources: out of the physical environment of the object (e.g. other objects, infrastructure gateways, environmental parameters) and/or by connecting spatially separated resources (e.g. central databases) across the Internet of Things.

- Due to the possibility of disconnected environments, in which object networks may temporally loose connection to the Internet of Things infrastructure, the lack of knowledge about the objects environment and its surrounding situation as well as the systemic objective of robustness, a central clustering authority is impractical.

The Role of Context

- Clustering decisions require the objects' direct involvement and depend on the objects' own information, especially the objects' context and their capability of context awareness.

- Contextual clustering offers an access to self-categorised object groups over the Internet of Things; it is driven by a contextual rather than a process perspective (e.g., process oriented package flows).

CH Election

- The main objective of clustering is to extend the object network lifetime by electing a representative network member which collects all the communication within the network and forwards it to the outside.
- CHs are elected according to their residual energy.
- To address this issue, infrastructure gateways could send advertisement packets to announce their presence. Only CHs that receive an advertisement would participate in the CH election process.

CH Election

- T_{CH} could be calculated as a function of the node's residual energy:

  T_{CH} = C x Residual Energy, where C is a constant

A random delay, also function of the node's residual energy could be introduced to avoid collisions in the wireless channel when a big number of nodes are in the same cluster.

CH Election

A CH election procedure would start when any of the following situations occur:

- TCH expires
- The current CH cannot communicate with any infrastructure gateway expires
- A CH, whom did not participate in the previous election and has more residual energy than the current CH, receives an advertisement packet from a gateway
- A CH cannot communicate with the current CH before TCH
- A new object is added to the CH's network.

CH Election

- To avoid this problem, the CH election procedure could be started by any node which runs for more than a certain amount of time without being able to communicate with its CH. When the connection with the information infrastructure is re-established, networks with a CH selected in this way would start a regular CH election procedure again.

Cluster Membership

- In order for objects and networks to find other objects and networks, one or several nodes could send periodic discovery broadcast packets. Nodes receiving these packets would process the packet information and decide to become part of the same network or cluster by sending back a response packet. The results would be communicated to all the network members and a new CH election process would begin. This process could involve several objects and networks at the same time.

Cluster Membership

Association request reception

```
Input: List of association requests
Output: Association response, update to local network

if (list of requests NOT null)
        HAL = list of requests from networks with highest number of nodes;
        if (I am in HAL)
            #Nodes: #Nodes + #Nodes in the list of association requests;
            Initiate CH election
            Send association response to the list of association requests;
            Update my network;
        end
end
```

Cluster Membership

```
Association response reception
-----------------------------------
Input: Association response
Output: Update to local network

According to the association response...
        change my network ID;
        change my address;
        update #Nodes;

Update my network;
```

The objective of object clustering is not only to manage the energy resources of the network, but also to form collaborative groups of objects that share a common situation or purpose. For this reason, we could propose an association procedure which considers aspects of the object's nature in order to filter and classify potential object interactions before they occur.

Cluster Membership

This association procedure would therefore have two phases:

- The first phase, in which association requests would be organised and filtered, and
- a second phase, in which the final association procedure would take place, and which would include the update of the cluster attributes, such as the election of a new CH or the re-factoring of routing addresses.

Cluster Membership

Association requests could include information not only about static attributes of the objects (e.g. ID, address, network), but also dynamic and contextual attributes. We could divide these attributes into two groups, regarding the need for their presence in order for the analysis of the requests to proceed.

- **Mandatory attributes** would need to be held by both, the request sender and the receiver,
- **Optional attributes** would not be a pre-condition for the association procedure to continue, but rather they would add up to the decision on membership of the requesting object.

Cluster Membership

- A **disassociation procedure** would need to be undertaken if an object or group of objects leave the network. This process would need to update all the attributes presented before, such as the number of nodes per network or the addresses of the nodes. The algorithms would also need to include provisions for re-merging a network whose routing structure was broken, the selection of a new network identifier if the previous one is not representative enough after the disassociation, the election of a new CH if the previous CH left the network, etc.

Addressing and Routing

- Dynamic address allocation is a common problem for wireless ad-hoc networks where mobile nodes constantly join and leave the network. Unlike wired networks, which lack of strong power and infrastructure constraints, mobile networks must optimise their operation and keep the connectivity even when unexpected topology changes occur. Wireless Sensor Networks pose additional challenges due to their especially scarce resources.

Addressing and Routing

We devise the following requirements for an Internet of Things addressing scheme:

- Addresses must be unique inside a network
- Addresses must be reused when objects leave
- Addressing must be dynamic. Address assignation should be fully distributed
- Addressing must be scalable
- Support for network merge and split should be provided
- The protocol overhead must be minimised

Addressing and Routing

To meet these requirements, the following list summarises the ideal properties that an addressing scheme for the Internet of Things devices should have:

- Hierarchical
- Distributed and unique
- Scalable
- Low overhead
- Extensible

Software Agents for Object Representation

The communication scenarios that might typically occur in a product lifecycle:

- **Product to product communication**
  - Products that request service, asking other products for batch orders
  - Problem co-diagnosis, where products of the same firm consult each other for undiagnosed failure modes
- **Product to supplier communication**
  - Products asking service provision (including recycling, maintenance, scrappage, and logistics) from suppliers with a given service request, time, and price
  - Manufacturer communicating product upgrades or recalls to products
  - Products communicating performance data to manufacturer

Software Agents for Object Representation

- **Product to user communication**
  - Product performance and actions
  - Product location and state
  - Upgrades, promotions and additional services

Data Synchronisation

Types of Network Architectures

Data synchronisation in the Internet of Things depends on the availability of connectivity within the network architecture in which the objects are moving.

There are three types of architectures that have to be examined:

- Connected architectures (Internet),
- Partitioned architectures (Intranet, Extranet),
- Disconnected architectures (local resources).

Data Synchronisation

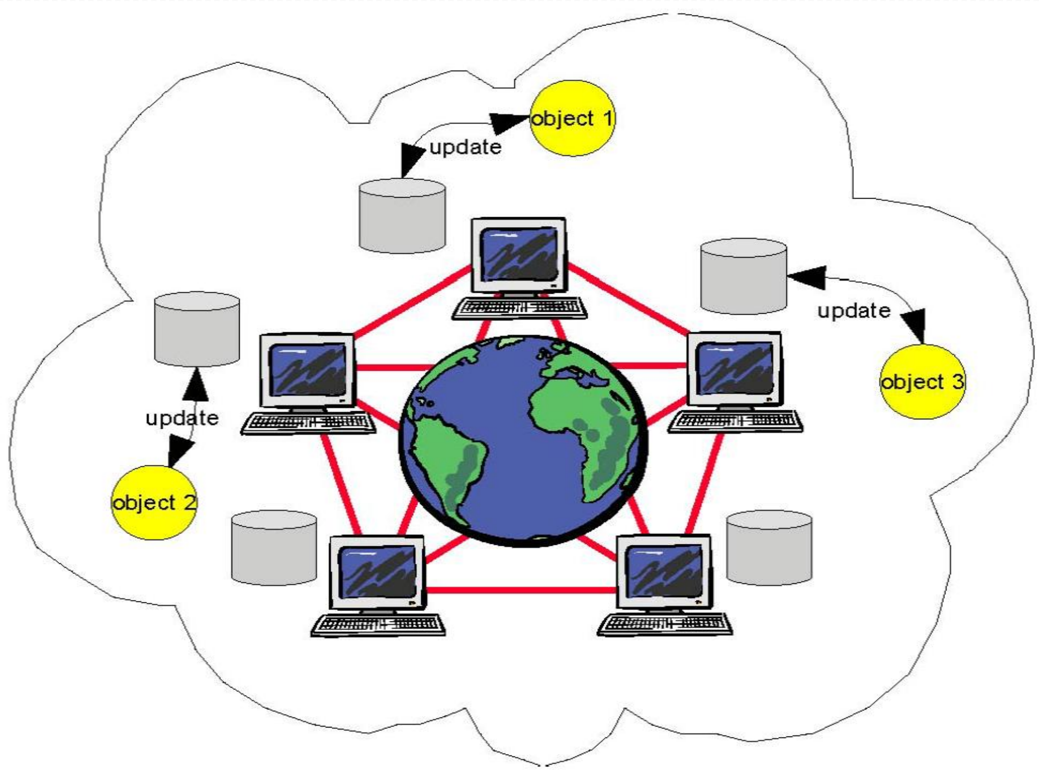
Connected Architectures

The Internet is a globally distributed network of computers carrying many services and information resources. It is assumed that resources are always connected to share and update information. Objects moving in a connected architecture do not need a special synchronisation method, because object information can be updated at any place inside the network in real-time.

Example: EPC Global

Data Synchronisation

Connected Architectures



Data Synchronisation

Partitioned Architectures

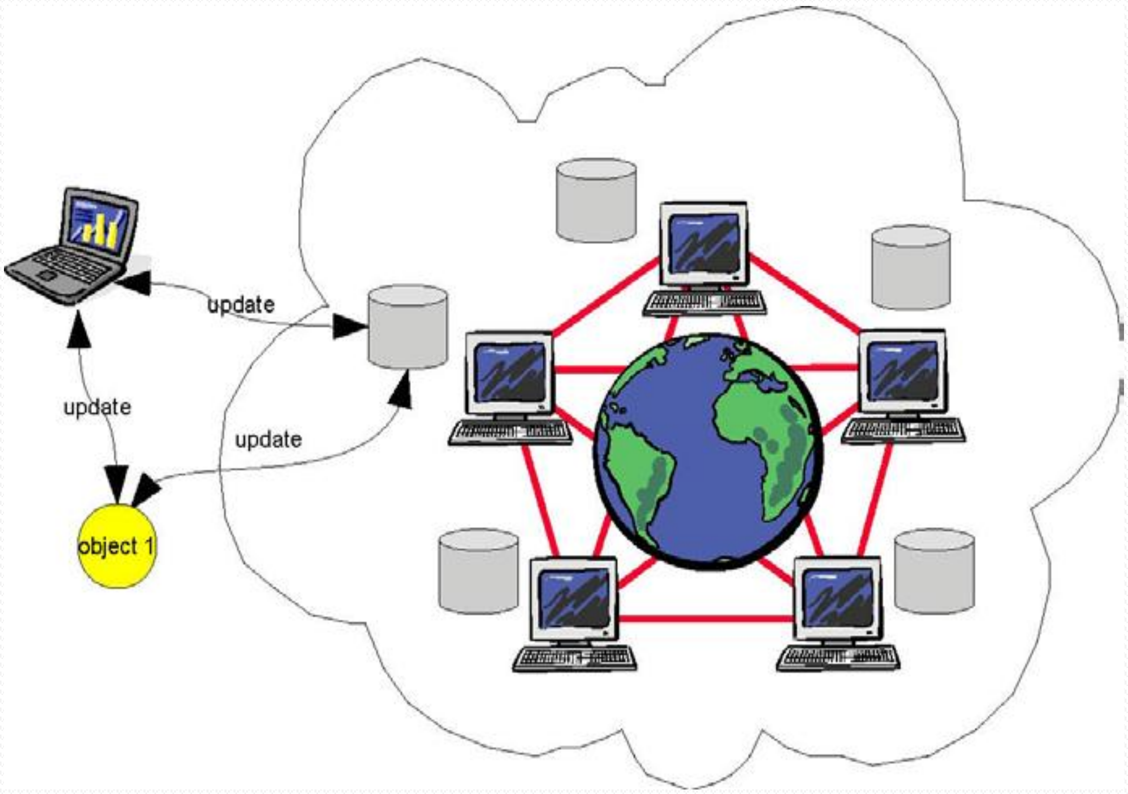
Intranet and Extranet may be seen as types of partitioned networks. They base on the same technology as the Internet, but they are only reachable inside a closed area (Intranet) or with a special authentication (Extranet).

Beside partitioned networks, there are partially disconnected devices, such as mobile devices, which are disconnected while updating object information (e.g., for maintenance).

Data will be updated at the mobile device first and has to be synchronised to related network information resources when the mobile device will be connected again.

Data Synchronisation

Partitioned Architectures



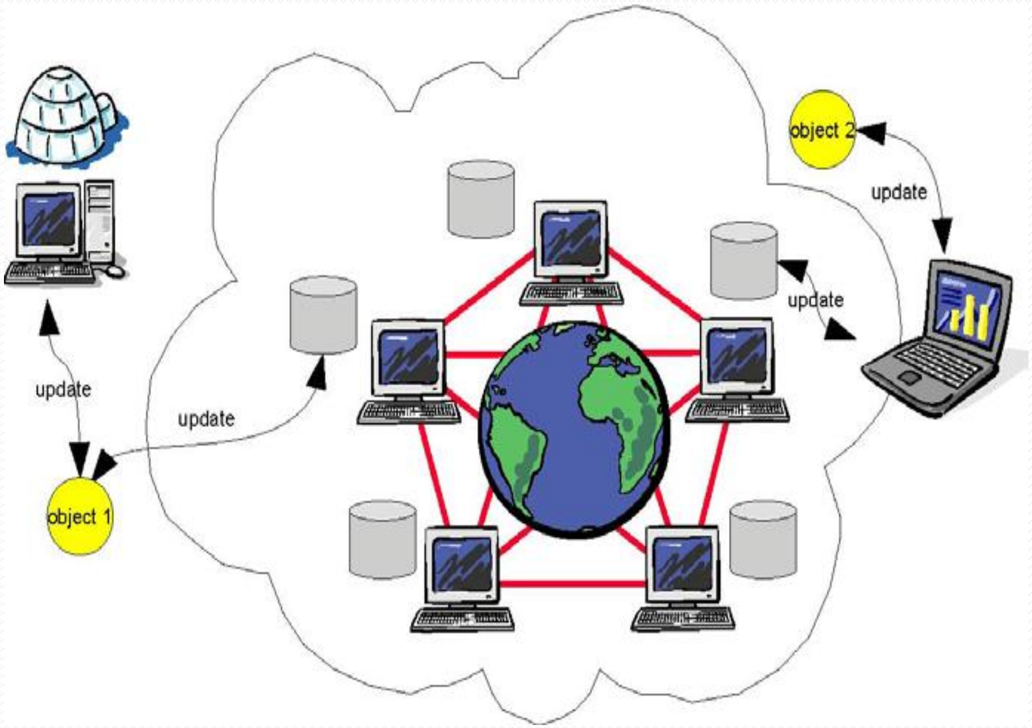
Data Synchronisation

Disconnected Architectures

- A permanent disconnected architecture has to distinguish between objects and other resources without network availability.
- Objects may be disconnected, because they are unmovable or only moving in environments without network availability.
- Mobile devices could be used to exchange information between disconnected objects and other network resources.
- In the case of disconnected applications or repositories, information could be exchanged via mobile objects.

Data Synchronisation

- **Disconnected Architectures**



Interfaces

- The Internet of Things has to handle the information exchange in heterogeneous distributed networks with characteristics of all architectures described above. Therefore, it is important to identify all the interfaces for information exchange and to define synchronisation mechanisms to assure data consistency and security

Architecture	Types of information exchange	Time of synchronisation
Internet	Real-time between object and connected resources	Real-time
Partitioned networks	By objects	Next connection
Partially disconnected	By objects and mobile devices	Next connection
Permanently disconnected (object)	By mobile devices	Next connection
Permanently disconnected (local resources)	By objects	Next connection

Thank You