UNIVERSITY OF CAMBRIDGE

Supply Chain sensor support by integrating the OGC Sensor Web Enablement and the EPC Network architectures

Tomás Sánchez López and Mark Harrison
University of Cambridge

Auto-ID Lab White Paper *SWNET-028*

March 2010

Abstract: *The existing standards dealing with Auto-ID data for supply chain scenarios do not integrate sensor information from the products that are being monitored. In particular, although a number of efforts have been undertaken in the last few years to incorporate sensor data inside the EPCglobal Network, its has proven to be a slow process due to both the complexity of the task and the standardization processes themselves. On the other hand, the Open Geospatial Consortium (OGC) has been developing a complete set of Internet-based sensor standards under the name of the Sensor Web Enablement (SWE). This set of standards is built on a distributed architecture whose main components perform similar functionalities to those found in the EPC Network. Based on this observation, this paper discusses a different approach to the integration of sensor information into the EPCglobal Network which consists of building an application level layer to connect the EPC Network components with those of the OGC SWE. We show how this approach can provide sensor+auto-ID-enabled supply chains with little or no changes to the existing standards and that this can support some advanced functionalities such as ambient sensors or multiple sensors per product.*

This research was supported in part by the EU IST Sixth Framework (FP6) project BRIDGE (Building RFID Solutions for the Global Environment) under contract number 033546

1. Introduction

Radio Frequency Identification (RFID) technology is gaining momentum as a key player in asset tracking systems, and it has become a central spot of attention for improving the logistics processes in medium and global supply chains. The advantages of RFID systems for asset tracking and management, as compared to traditional identification methods such as the bar code, are many [RFID]. The evolution of standards has also influenced the popularization of RFID in cross-enterprise supply chains, allowing various players to deploy RFID hardware and services with the certainty that the logistics operation will function. Among the variety of RFID standards that have appeared in the last few years, the GS1/EPCglobal [EPCnetwork] are the most promising, delivering a number of services on top of the RFID infrastructures and providing features such as global access to tracking data, discovery of data sources, data filtering and others.

RFID technology in general, and the EPCglobal standards in particular, have naturally evolved towards completeness of services based on item identification. However, new applications of RFID require tagged items to report more than their identification numbers, and cases for the periodic monitoring of product and object condition are constantly being built in research projects and commercial companies [BRIDGE] [SupplyChain] [EPCsensor]. There is thus an obvious trend on the industry to move towards an integration of RFID and sensors systems, but the complexity of incorporating sensor information within the current RFID standards has hampered an standard-based integration of sensor into supply chain logistics.

The key RFID standardization bodies are currently working on incorporating provisions for integrating sensor data into their standardization document. Example of these standards are the ISO/IEC 24753-2, 18000-6 and the 18000-7, and the IEEE 1451.7. The GS1/EPCglobal is currently sponsoring research on this area, but is not actively working in modifying its current standards, probably aiming at improving current RFID functionality or developing new functionality seen as more critical.

Sensors and sensor data are, of course, not of exclusive interest to RFID related applications. Sensors are used in many other applications such as environment monitoring, industrial monitoring, vehicle health management, etc, which need not be related with RFID. Some standardization bodies have developed important standards on this regard over the years, some times in parallel (and independently) with their efforts on RFID. IEEE, for example, started over 15 years ago setting the foundations of the IEEE 1451 set of standards [1451] There are also other sensor standardization bodies that never had a well-defined involvement with RFID. The most clear example is the Open Geospatial Consortium (OGC), whose Sensor Web Enablement (SWE) [OGCSWE] is of special interest due to its completeness and adoption. RFID-independent sensor standards can be of great value to the RFID community, since they can be adapted to work together with existing RFID standards. IEEE, in a privileged position due to the broad range of its standards, has already considered an extension to the IEEE 1451 family to include RFID, and is already collaborating with other standardization bodies to integrate existing RFID and sensor standards.

This document discusses the addition of sensor support to the Supply Chain data gathering services by means of integrating the EPC Network and the OGC Sensor Web Enablement architectures. The objective of this task is to show an alternative integration strategy in which the existing RFID/EPC Network standards are not extended to support new functionalities (I.e sensor data), but they are linked at an application software layer with other well established standards that implement the required functionality. The EPC Network and the OGC Sensor Web Enablement architectures are chosen due to their leading position in supporting two different technologies: Networked RFID and Web-based sensor systems. The documents starts by reviewing the main features of these two architectures and identifying potential synergies. In the second part of the document, a fresh meat traceability case study is used to show step-by-step how the two architectures could be linked and which challenges this strategy would face. Finally, the document proposes a few architectural additions in order to support a flexible set of sensor dispositions, namely ambient sensors, sensors in reusable assets and multiple sensors per product.

2. EPC Network – OGC SWE comparison

2.1.1. Background

2.1.2. OGC Sensor Web Enablement (SWE)

OGC SWE [OGCSWE] is a set of standards defined on top of general geospatial standards by the Open Geospatial Consortium. The aim of the SWE is to define architectures and models for defining, discovering, configuring and retrieving sensors and sensor data, in the framework of distributed Web systems.

This set of standards provides models to describe sensor data (SensorML [SensorML] and TML [TML]). The standards also describe how to pack this data into higher-level meaningful information (Observations and Measurements [O&M]). The role of the Sensor Observation Service (SOS) [SOS] is to receive observation queries from the clients and respond according to the sensors and sensor systems that are under its management. SOS also gives clients access to the information about the sensors themselves and their capabilities (metadata described in SensorML or TML). Due to the complexity that an observation query might involve, a planning service (Sensor Planning Service – SPS [SPS]) is also defined, through which clients can request query feasibility prior to querying for the data itself. A Sensor Alert Service (SAS) [SAS] provides ways of alerting clients about particular sensor conditions, either by synchronous or asynchronous means (the latter using the Web Notification Service – WNS [WNS]). Finally, a generic catalogue (repository) service (CS) is defined by OGC that can be used within the SWE used in its CS-W extension for discovering data [CS-W].

2.1.3. EPC Network

The EPC network architectural framework [EPCnetwork] is a set of standards for defining, discovering, recording and retrieving unique IDs [EPCs] and related information. The EPC Network standards are developed by GS1/EPCglobal and target item-level tagging to drive automatic identification of products in the logistics processes that take place in the product's supply chain. The EPC Network currently focuses on observations of uniquely identified objects and the associations between objects, locations, business transactions and business context throughout supply chain processes.

Clients of the EPC Network are able to access event information obtained from RFID systems by querying the EPC Information Services [EPCIS] interfaces. RFID tag reads are filtered, enriched with business context and stored in on-line repositories, as well as being pushed to clients that have subscribed to queries that match the events. Application Level Events [ALE] provides a standard interface for clients to specify filtering criteria. ALE v1.1 also provides methods for reading and writing to tags. The Reader Management standard and forthcoming Discovery, Configuration and Initialisation standard allow for configuration and monitoring of readers. A Management application can be used to monitor the health of readers and reader networks. Two systems can be used to obtain the addresses of relevant repositories: The Object Name Service [ONS] returns addresses of authoritative information for a particular EPC class; typically it returns the address of the manufacturer's EPC Information Service [EPCIS] repository. Discovery Services provide authenticated authorized clients with addresses of information resources provided by multiple organisations that claim to hold information for an individual EPC and allow multiple organizations to register such assertions and create protected links to their information resources – i.e. other sources of information can be found in addition to information provided by the manufacturer of the product.

2.2. Similarities

	EPC Network	OGC SWE
Repository of observations & data	EPCIS	SOS
Discovery Services	Discovery Services	CS-W Catalogue
Filtering	ALE, Reader Protocol	SOS (SAS & SPS)
Single point-of-entry queries	ONS, EPCIS query interfaces	SOS (SAS & SPS)
Alerting service	EPCIS/ALE standing queries	SAS & WNS

Table 1: Similarities between EPC Network and OGC SWE and involved standards

2.3. Differences

	EPC Network	OGC SWE
Data Sources	No registration is needed, and data is queried via pre-defined attributes	Registration needed prior data access, providing attributes for later query
	Separation between readers and tags	Only one data source type
Identification numbers	Globally unique IDs (EPCs)	No global IDs. IDs are discovered through sensor data queries, which return the IDs of the resources matching the data.
Queries	ALE and EPCIS may report to interfaces or user-specific URIs (standing queries)[1]	WNS is used inside SAS, supporting multiple data-delivery types including asynchronous messaging.
Alerts	Alerts originate in the data repositories.	Alerts are independent from data repositories (SAS)
Planning	No planning element[2]	SPS as planning service

Table 2: Differences **between EPC Network and OGC SWE**

2.4. Identified synergies and recommendations

The following list identifies a number of synergies found during the initial comparison between the EPC Network and the OGC SWE. Some recommendations as to how to proceed for an integration of the architectures as also included where appropriate:

- *Sensor data models:* SensorML appears as a mature tool for describing sensor metadata and sensor data model. TML might provide additional concepts, although this needs to be further investigated depending on the targetted applications.
- *Query and filtering of data:* SOS/SPS are also mature methodologies for querying and filtering sensor data and metadata. Adopting these methodologies would imply the adoption of the Observation and Measurements (O&M) standards and to require that sensor nodes / Base Stations discover / register to the local SOS service. At this moment, it appears inappropriate to use EPCIS 1.0 to store sensor events and metadata because the current EPCIS 1.0 query language is not sufficiently expressive or flexible.
- *Format of Identification numbers:* The sensor IDs used by SOS don't have a defined format. It would be possible to enforce the use of URIs (e.g EPC) as the sensorIDs, which are transferred at registration time. From the point of view of SOS, it is not important if a registration is done by a sensor, a sensor node, a Base Station (BS) or another system. In this sense, a BS could register the sensor nodes or the BS could be registered as a sensor system providing all the sensor capabilities of their sensor nodes. In any case, the SOS could be queried either by ID (URI or EPC) or sensor data / metadata.

1 EPCIS can return a '*QueryTooComplex*' or '*QueryTooLarge*' exception, which is probably the nearest equivalent

2 EPCIS supports a general purpose notification URI to allow results of standing query subscriptions to be delivered via any mechanism indicated by that URI.

- *Approach to data integration:* The orchestration of separate repositories could provide a rapid and flexible approach for the data integration. In this context, the EPCIS would be used for supply chain related events while SOS would be use for sensor data. If retrival of object ID's from objects matching a certain sensor data criteria was necessary, SOS would be queried first, and the EPCs obtained in this way could be used to query EPCIS for relevant supply chain transactions (i.e. which objects were in contact with a particular EPC). SOS addresses could be added to ONS or Discovery Services as additional service types, alongside EPCIS.
- *Discovery services:* An integrated discovery service could support queries that involve both EPCs/transactions and sensor information (e.g. obtain addresses of nodes which participated in a particular transaction and which support certain sensor capabilities). Although both the OGC SWE and the EPC Network include their own discovery services (CS-W and EPC Network Discovery respectively), it could be easier to extend the EPC Network Discovery Services to support sensor metadata as defined in OGC SWE. EPC Network Discovery requirements include some very specific requirements about protecting confidentiality of information that could otherwise reveal volumes and flows of goods, so it would probably be the more complex problem to solve. On the other hand, an interface to ease the process of querying multiple repositories and services could be developed.
- *Metadata considerations:* Metadata is information about the sensor (currently encoded in SensorML in the OGC SWE) that is not the data it produces. Dynamic metadata is metadata that can be configured in the sensors, such as reported ranges, reporting format, etc. SOS has no direct way of updating dynamic metadata. An additional optional operation called *UpdateSensor* could be designed in order to update sensor dynamic data, or extend the current *RegisterSensor* to allow sensor metadata updates.

3. Case Study: Fresh Meat Traceability

The EPC Network provides many benefits on complex supply networks where the products undertake several stages of processing and where the supply chain involves a big number of partners potentially spread over several countries. The EPC Network is able to cope with many challenges arising from such complex scenarios, including:

- automatic identification of items
- tracking and tracing opportunities for vendors and consumers
- restrictred access to information
- complete visibility of the entire logistics operations
- management of agreegations and decomposition of products.

The fresh meat supply chain is an example of the aforementioned complexity, suffering of many of the challenges mentioned perviously. This includes a wide spread supplier network from farms to end consumers, the limited knowledge among supply chain partners, the complex decomposition of meat (cow, half-cow, steak, etc) and a strong regulatory framework requiring that "the traceability of food [...] and food-producing animals [...] shall be stablished at all stages of production, processing and distribution" [Regulation178]. Figure 1 provides an example of a fresh meat supply network, aiming to show its potential complexity.

Despite the many benefits that the EPC Network could have over the fresh meat supply chain, it presents other business and regulatory challenges derived from its perishable

nature. Meat stays fresh for a limited time and needs permanent refrigeration at a constant level. Meat might also suffer from contamination such as too many or not permitted antibiotics, swine fever, foot and mouth disease, etc, and is prone to decay if storage and transportation quality is not carefully monitored. In order to provide a complete cold chain visibility, as well as detect the presence of unfresh or unsafe meat along the supply chain, it is necessary to incorporate sensing mechanisms for real-time alerting and historical analysis. However, as mentioned earlier in previous sections, there is no existing standards-based solution that addresses both the complexity of supply chain logistics and the monitoring of product's status. Under these premises, the rest of the this document analyses the integration of two independent architectures, namely the EPC Network and the OGC SWE, to meet the challenges that this fresh meat case study brings.

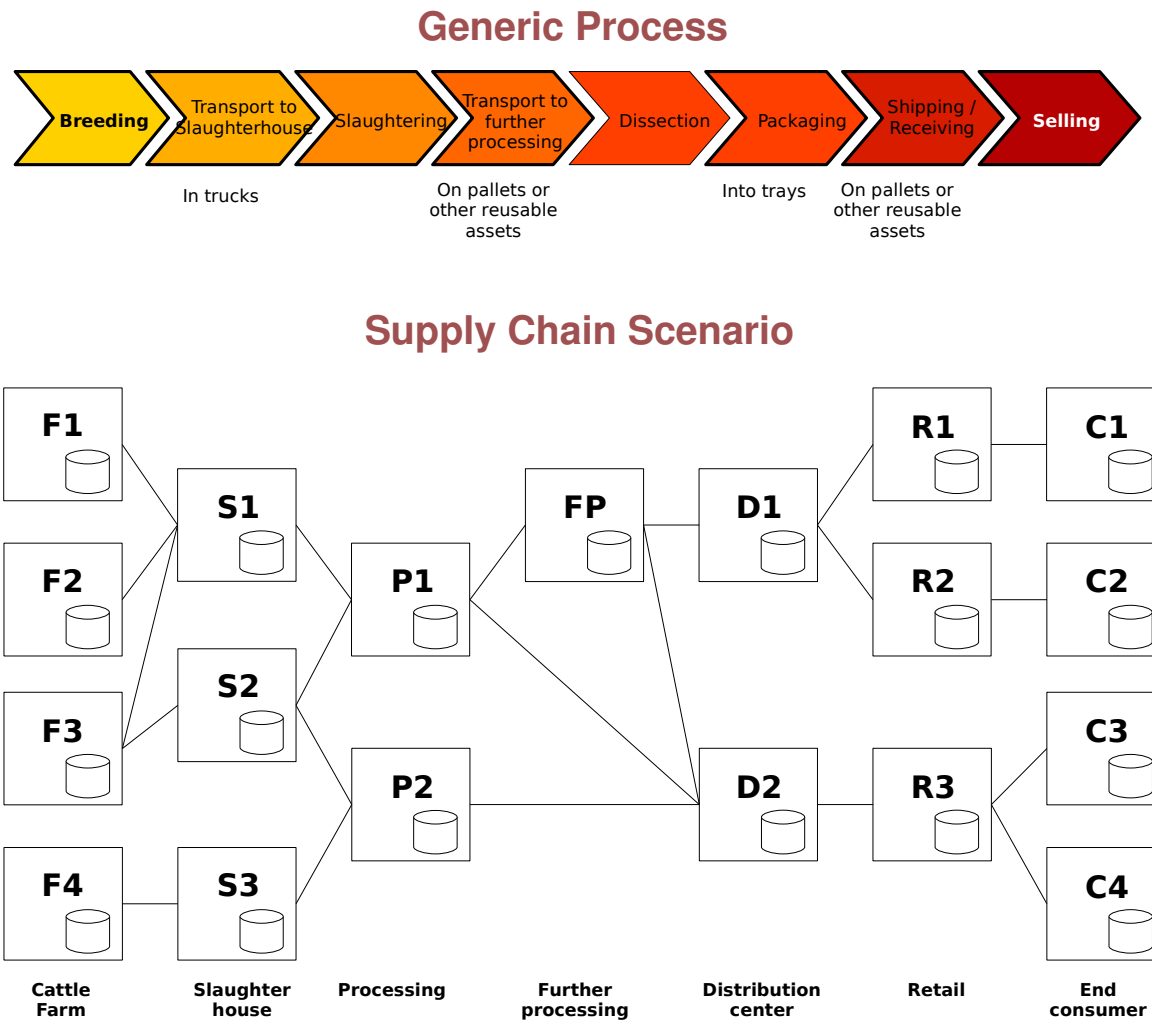

Figure 1: Fresh meat use case

4. Functionality study

As the functionality of our particular scenario, we will require that:

1. Sensor and ID information is captured and stored in both OGC SWE and EPC network respectively, with appropriate cross-referencing mechanisms
2. This information can be retrieved for historical data analysis
3. An alert is sent to the client if the temperature of the meat, at any stage of the supply chain, exceeds a certain threshold. Note that the time accuracy of this depends on how often the data is read. Recall that we assumed periodic updates at the introduction.

4.1. Assumptions

The objective of this scenario is to judge if the proposed merging of the OGC SWE framework and the EPC Network would provide sufficient functionality to enable real-life condition monitoring and tracking. This scenario assumes that:

- For each item to be monitored, RFID data is read and stored in a local EPC network instance (EPCIS), and sensor data is read and stored in a SOS instance. This means that, as a principle, we assume that the sensor readings can be unequivocally and automatically matched with a specific item[3]. We further assume that the data capture happens in the following way:
  - For RFID only, at least once at each supply chain step (preferably upon arrival) and once every time aggregation and decomposition occurs. Aggregation and decomposition is recorded and stored in the EPCIS of the entity where the event is generated.
  - For sensor information, at least once at the arrival to each supply chain step, and then periodically according to any real-time needs for the condition to be monitored.
- Either a catalogue service exists at each supply chain step that points to the SOS and that can be accessed by the sensors/sensor gateway, or a generic gateway at each step is able to read the sensor data without sensor reconfiguration and is configured to contact the SOS of that step (the last being very similar to what RFID readers do).
- Once the address of the SOS is known, the system in place (be it individual sensors or a certain gateway) registers every sensor source. This happens prior to the first data capture at each step.
- Discovery services would be functional and either there are no security and privacy issues involved in the data sharing or they are addressed by the implementation of the discovery services.

As a general statement, these assumptions mean that for each relevant item in the supply chain, mechanisms exist that allow its ID and sensor data to be read and stored in EPC Network and OGC SWE instances respectively, with the desired periodicity and granularity, and that discovery and catalogue services exist that allow the access to that data from a given (authorised) client.

3 Other option, as for example the use of ambient sensors, will be discussed at the end of the discussion.

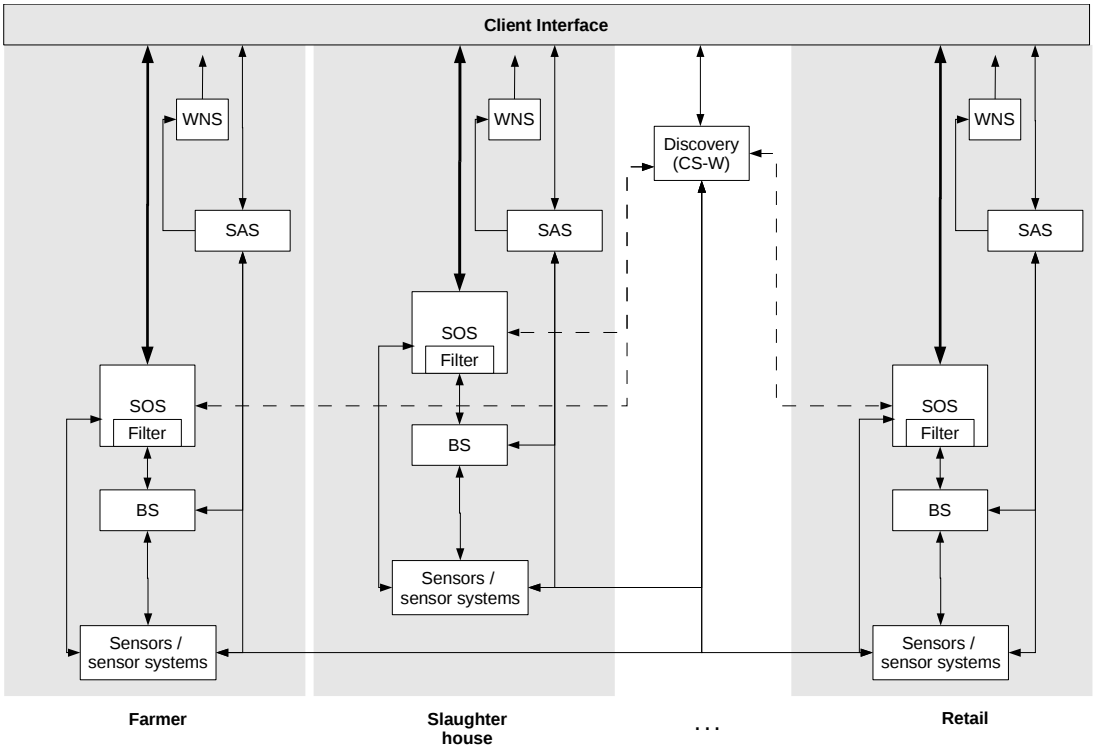

Figure 2: Possible OGC SWE installation for the Supply Chain

4.2. Data Capture

Regarding the first functionality point, it is achieved by meeting the previously listed assumptions: at each stage of the supply chain, ID and sensor data is independently captured by each architecture.

Regarding the RFID architecture, this document does not discuss in detail how the information flows from tags to Information Services since this is a widely documented elsewhere [EPCnetwork]. Typically, the EPC Network architecture for a supply chain is replicated at that each step (e.g Farm, Slaughter house, Distribution centre, etc) such that each has its own implementations of Readers, ALE and EPCIS. Via interaction with the ONS and Discovery Services, supply chain clients can gather all the information distributed in the databases across the whole supply chain (or, at least, the information they have permission to access).

In a similar way, an OGC SWE architecture for a supply chain would be distributed in the same way. This is not a design limitation, but rather an implementation decision based on the notion that the organizations that govern each supply chain step want to have control over the data that is produced in their step of the supply chain. This implementation decision is analogous to that of the EPC Network.

For simplicity, we will ignore the OGC SWE components that are not fundamental for obtaining the functionality that we require. This only includes the SPS, since an alerting

service (SAS + WNS) is very desirable for a scenario involving sensor data. However, note that we could proceed without the alerting service and limit our knowledge to an analysis of the historical data when the meat arrives at the retailer.

Figure 2 depicts a possible installation of the OGC SWE for our Supply Chain scenario. Note that the only non-replicated shared component is the catalogue service, which clients (sensor devices, Base Stations and other clients) use to discover SOS instances in a similar way to how Discovery Services are used in the EPC Network to discover EPCIS instances. Regarding the replication of components by each organisation, it is worth considering sensor values and how the are used to represent the condition of a given item (meat, in our case). It may be reasonable to think that for a sensor value that is associated with meat, the criteria used to raise alarms would be the same across its supply chain, post-slaughter. For example, if it is decided that above +5 °C, a beef product is in danger of being spoiled, this threshold value would apply regardless of which step of the supply chain the meat is at. It is also true, however, that alarm recipients would probably vary, so a single service with a single subscription will not suffice. The design of SAS allows for sensors to advertise their capabilities and clients to define alerts regarding those capabilities. It could be possible, thus, to provide a central alert service to which a sensor belonging to a meat item would advertise and to which any client could subscribe independently. This could solve the component replication issue in that it is not necessary for the same sensor to advertise in every supply chain step. However, it would not solve the problem of multiple subscriptions with the same criteria, since there is no mechanism to support the discovery of existing subscriptions. In any case, providing a centralized SAS service would create a similar problem as that of creating a centralized SOS: In the SAS design, sensors publish their data to the service, which effectively becomes a hub through which all the sensor readings, IDs and timestamps pass. This would surely become a confidentiality concern for the various organizations that send the data. We can thus conclude that the replication of SAS+WNS in all the supply chain steps is probably the best solution.

For a sensor to be able to publish data in an SOS repository, it must first register with the repository. The registration could be done by the sensor itself (in which case the sensor/sensor node needs to be able to encode XML messages) or could be done by the BS on behalf of the network. The registration will only be used while the item remains within that organization[4]. Note that the address of the SOS is not known a-priori by the sensor (each SOS from each organization is in a different address). The sensor would need thus to discover the SOS using the Discovery Service. Once that happens and a sensor is registered, data can be published and can be queried or discovered by clients. In case the registration is done by a BS on behalf of the network sensors, the BS could be pre-installed with the address of the SOS. Other combinations are also possible. For example, the BS (active tag reader) could know the address of the SOS and let the sensor nodes learn it when they first communicate. In this case, the sensors would still register to the SOS by themselves but no discovery mechanism would be necessary.

We have described so far how both the ID data and the RFID data are stored in distributed databases (EPCIS and SOS) along the supply chain. We now need a mechanism that will match both data streams and that can be queried seamlessly. The SOS returns a *RegisterSensor* response upon registering a sensor. This response includes an

4 Interestingly, it seems that there is no operation to cancel a registration. This means that once the data of a sensor has been inserted after registration, it can't be deleted.

AssignedSensorID, which by default will be randomly chosen by the SOS. We would like, however, to know this ID beforehand or, more specifically, be able to inject a known ID instead of letting the SOS choose one for us. The reason for this is that without a known ID it would be impossible to retrieve information about a specific sensor, a step that is required in order to match EPCnetwork IDs and OGC SWE sensor data. To be able to control the assignation of the sensorID, we could include this information as part of the SensorDescription when registering the sensor, and program the SOS so it will read, assign and return this ID as a response of the registration. This identifier is described as "of type anyURI" by the SOS specification. Section 5.1. will discuss which type of ID should be injected into the SOS registration.

4.3. Data Retrieval

The methodology that we are proposing here combines two independent architectures . For this reason, a client should either invoke independent procedures to retrieve data from them, or we should provide an orchestration component that takes unified client requests and coordinates the connection with both architectures to provide a unified answer. In any case, the intention of this section is to prove that meaningful condition information unequivocally linked to "legacy" EPC information can be obtained by using the approach presented here.

For similar reasons as explained in the previous section, we will not explain in detail the intricacies of EPC Network data retrieval. We will assume, however, that the goals of accessing additional sensor data from the OGC SWE architecture are the following:

- To be able to obtain condition data for meat (or other perishable goods) associated with a specific EPC and for a particular time period and/or location.
- To be able to issue alarms under particular conditions (e.g. relating to food safety)
- Once an alarm is raised, to be able to analyse the trace of the offending meat product in order to determine partners potentially responsible for unsafe handling conditions.
- To be able to query historical condition data.

Sensor data is sent to the SOS encoded in Observations as specified by the O&M OGC SWE specification. The O&M specification is very complete and flexible, and includes among other things time stamps and multiple ways of defining locations. In the EPC Network, time-stamped records are captured at the EPCIS layer and include two type of locations, the ReadPointID and the BusinessLocationID. A match between both time-stamps might be trivial if understood correctly. Regarding the locations, it is evident that O&M allows much more complex specifications (e.g geometric areas). It is also evident, however, that it would be possible to use similar location IDs in both architectures in order to ease the integration of both sensor and ID data streams. To this extent, we may thus want to extend the Observations sent by the sensor nodes with custom fields called *ReadPointID* and *BusinessLocationID.* This extension would be most suitable for fixed sensors (I.e ambient sensors) as presented in section 5.1.1.

The SOS supports two types of queries that are of interest to our proposal. The first query is called *getObservation* and allows a client to request the retrieval of any type of data according to the O&M structure. This includes location and time. The second query is called *getObservationById* and allows the retrieval of data directly based on sensor IDs. As explained earlier, by being aware of which sensorIDs represent which product (i.e by

controlling their assignation), it would be straightforward to retrieve both ID and condition data based on the product instance identifier (EPC).

Finally, alarms naturally provide the ID of the offending sensor when they are triggered. They also inherently provide the time that the alarm was produced. Additionally, further sensor details can be accessed by querying the SOS with the same sensor ID. It would also be possible to extract the EPC and timestamp of a meat product from an alarm and use that information to query the Discovery Service and relevant EPCIS (or the SOS) to obtain a trace of what happened prior to the alarm being triggered.

4.4. Alerts

The Sensor Alerting Service (SAS) of the OGC SWE specification supports the subscription and triggering of alerts based on sensor data. SAS works independently from SOS, so sensors wishing to participate in alerts need also to subscribe to SAS even if they already subscribed to SOS. One of the advantages of this approach is that the sensor description doesn't need to be the same (e.g. it is possible to register a single alarm for a set of sensors). Unfortunately, for this same reason the SensorID returned from SAS upon registration is unrelated to the SensorID returned by SOS. Let's assume that our interest is to receive alarms from specific items. In this case, we would like be able to inject the SensorID and so relate it to the product, the same way we registered the sensor with the SOS. For this purpose, the sensor would have to send the ID to the SAS in its advertisement (Advertise operation), so SAS can use the injected ID as the SensorID instead of generating one ID by itself. Unlike the O&M, the SAS specification, which has not been updated since 2006, does not provide a wildcard field that can contain extra information, although the description of SensorID specifies a “Unique ID for every registered sensor, usually set by SAS”. In any case, we could easy extend the XML encoding of an Advertise request to include an extra ID field, and program the logic of the SAS so this ID would be used as the SensorID in the Advertise response. From that moment, all the operations that refer to any alert coming from that sensor will include the injected ID which is related to the relevant product.

Sensors register their capabilities, while subscribers register their interest in specific conditions of those capabilities. SAS offers an interface for clients to discover and subscribe to alerts of registered sensors. Once registered sensors have been discovered, SAS provides a language to allow clients to express filters that define to which sensor data they want to subscribe. In this way, only the sensor data that passes through the filter will be forwarded to the clients. This sensor data is encoded together with the SensorID and timestamp.

SAS uses XMPP to allow clients to subscribe to a particular stream of sensor data. Subsequent alerts are delivered either by the XMPP protocol itself, or by other means using the WNS OGC specification (supporting e-mail, HTTP, SMS, Fax and others). Once the alerts have been received, clients can extract the information and request further information from other services (e.g. SOS or EPCIS Network) if necessary.

5. Architecture considerations

5.1. Sensor Disposition and ID assignation

As underlined in the introduction, the assumption for the previous argumentation was that any sensor data can be matched unequivocally with a particular item, specifically via the EPC of the product. Technically, this could be achieved either by using a single sensor tag in each product or by combining one sensor tag and one passive tag. In the former case, the Base Station and Reader used for capturing data would be the same for both the EPC Network and the OGC SWE, and in the latter case they would be different. Since the handling of sensor data at any level of the EPC Network is not yet standardized, it is likely that an actual implementation would have to use standard passive RFID tags together with some proprietary sensor tags (e.g. wireless sensor nodes). In this case, it would be necessary to be able to map the ID of the sensors to the ID of the products they are representing. There might be several ways of assuring this mapping, and they might depend on the disposition of the sensors in relation to the product that needs to be monitored. However, it is paramount to devise a system that would be flexible enough to adapt to all the possible disposition scenarios. This section discusses the most relevant of these scenarios and tries to design an cross-reference mechanism that satisfies all of them.

5.1.1. Ambient sensors

We now consider the use of “ambient” sensors and item-level passive tags. In this approach, the sensors are located in the environment that surrounds the actual meat items, and a middleware layer is used to extrapolate the ambient sensor readings to each item. In this case, the matching of sensor data and identification data is not one-to-one, but one-to-many, as one sensor reading would be assigned to all the items that were located, at the same time, in the area of influence[5] of the sensor. Technically, the difficulty of the implementation of this approach resides in how to decide which sensor reading is assigned to which item. It could be argued that the most straightforward way would be to do this by matching locations, either by developing a simple *locationID* comparison (e.g. Reader 1 is located in Room 2, so all the sensor values from Room 2 are assigned to any item read by Reader 1) or by comparing geographical locations (e.g. Reader 1 is located in coordinates (x,y,z), and the area of influence of Sensor 2 is determined by the polygon W. If (x,y,z) is contained in polygon W, then readings from Reader 1 will be assigned to Sensor 2). Furthermore, we should make an approximation of how long an item has remained within a given area of influence of a sensor. Doing so would involve taking into account that a passive RFID

5 We must not confuse Sensor Range and the area of influence of a sensor. A sensor range extends exactly to the area where the sensor can physically capture a particular phenomena (or, at least, the area that has been determined that the sensor can capture the phenomena lie within a particular confidence value). A sensor area of influence is the area represented by a particular sensor, as determined a priori. A sensor area of influence could be equal or smaller than the sensor range. Typically, sensor ranges overlap between adjacent sensors, but sensor areas of influence do not.

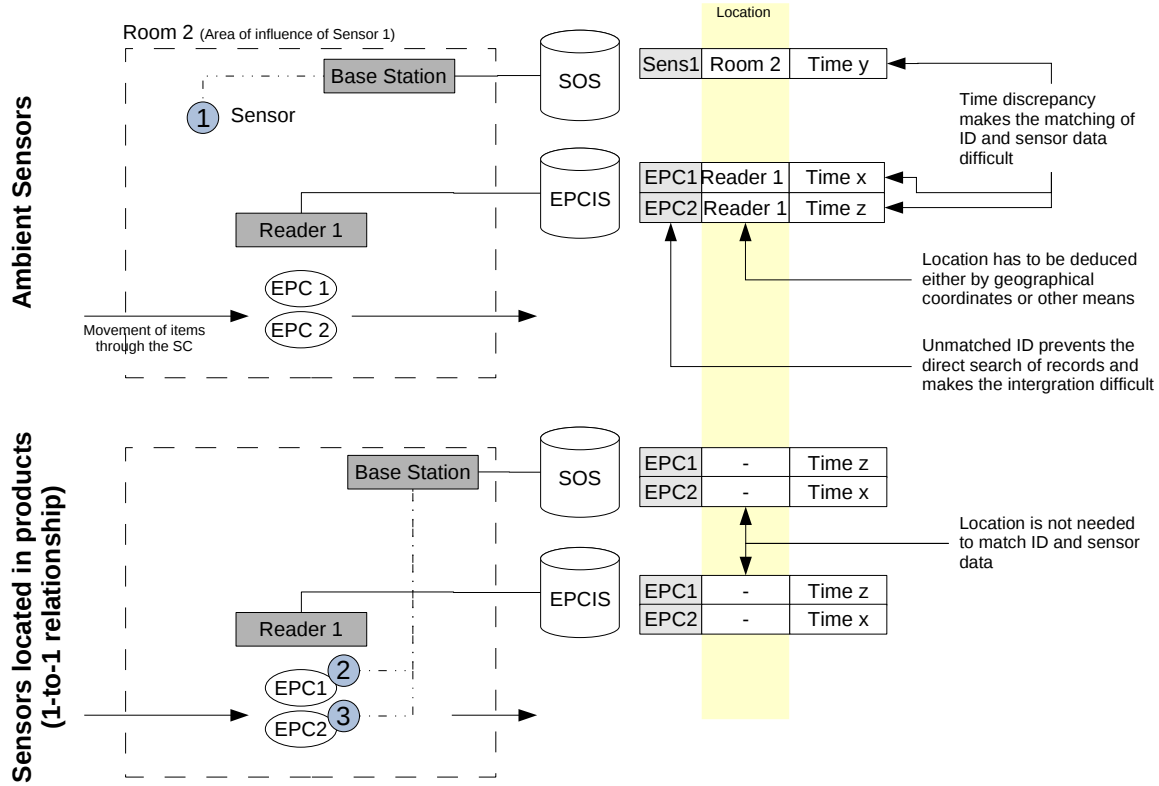

Figure 3: Ambient sensors vs. sensors in product.

reading point is discrete, and depending on the travelling speed of the item it could have remained in a certain area for a longer or shorter time.

The implementation of this latest option bears a certain difficulty and needs the development of a software layer with access to both the EPC Network architecture and the OGC SWE. This software layer could be a more complex version of the orchestration component that was mentioned earlier. For each piece of integrated information requested by the client, a comparison of location should take place from records of the EPCIS and SOS. For example, should we want to know the historical condition data of a certain item, the procedure would have to:

1. Search the EPC Network for each occurrence of the item's EPC that has a different location. For convenience, order the retrieved records by time-stamp. The Event Gathering Layer presented in task 3.2 of this Work Package already supports this functionality.
2. For each different location, search the SOS for sensors allocated to an area covering that location.
3. Select only the sensor values within the same time period as the observed EPC. Note that here an approximation of time should take place. For example, we might conclude that a reader is located in the centre of the area of influence of a sensor, and that according to its estimated travelling speed, the tagged object has remained in that area ±10 seconds from the time-stamp registered by the EPC reader.

Running this type of algorithm for every client request is certainly not efficient in terms of resource utilization. We could consider, thus, to build a secondary SOS database that continuously matches RFID and sensor data location and time. Clients would only have to query this database to obtain the sensor information, in a similar way to what was explained in the first approach.

As mentioned in section 4.3., it is nevertheless possible to make a simple comparison of locations using, for example, *bizLocationIDs* defined in the EPCIS. In this way, any sensor reading taken at the same time and at the same location as product tag readings would be considered to be a match for that product. There might still be inaccuracies unless a particular location can register products on their way in and way out (e.g portals), since otherwise we can not be certain when the product entered the room and was therefore inside the area of influence of a certain sensor. As explained earlier, this problem can become more acute if the area of influence of a particular sensor/sensor system cannot be approximated to a physically defined area where the way in/out cannot be recorded systematically.

Figure 3 shows the difference between the implementation of an ambient-sensors approach versus an approach in which an individual sensor is located in each product and its ID is made equal to the ID of the product's RFID tag. Considering sensor IDs and product ID to be equal is a rather simplistic approach, but enough for the purpose of understanding the implications on the use of ambient sensors or where there is only one sensor per product instance. Other options regarding the number of sensor per product and unequal IDs are discussed in the following sections.

5.1.2. Sensors in reusable assets

Rather than attaching sensors to each individual product, it also possible to attach them in the reusable assets that transport those products along the supply chain. Examples of these reusable assets are trays, pallets or even containers. This option may be desirable due to the fact that it reduces cost in the potentially costly sensor devices, both reducing their numbers and making them reusable. We now discuss how this scenario would affect the proposed architecture.

A shared sensor device in a reusable asset carrying several meat products implies that there can not be a one-to-one correspondence between sensor ID and product ID. The question that we should answer is therefore, given a product's EPC, how to find out the ID of the sensor of the reusable asset in which it was stored or transported. Of course, if the reusable asset changes in any way (e.g aggregation of pallets, change to a new reusable asset) we should also be able to find this out.

Probably the first possibility that comes to mind is to use some kind of aggregation-disaggregation tracking service that records when a product has been put in a reusable asset as well as when is has been removed. The EPC Network already enables this by means of aggregation events within the EPCIS event data model, and the Event Gathering Layer developed in BRIDGE WP3 provides a mechanism for automatically following such changes of aggregation. By using this functionality, we could ultimately know which reusable asset was transporting which product at any point in the supply chain. The problem with this approach is that, for the EPC Network to be able to capture the reusable asset's ID data (and thus record the aggregation events), the reusable asset must have a tag that can be read by

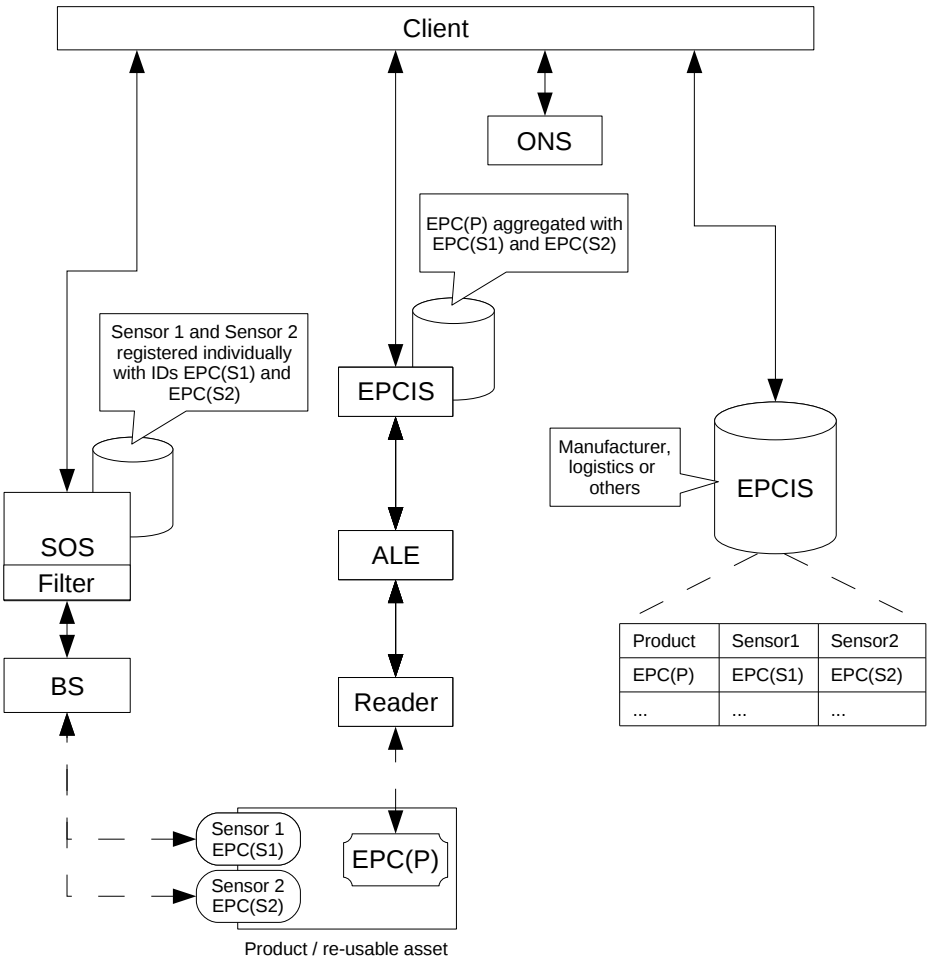

Figure 4: Players with several sensor devices per product or reusable asset

the EPC Network readers. Now, to achieve this, we could either use special tags that can be read by both the OGC SWE gateways and the RFID readers (i.e EPC compatible sensor tag), or we could have a separate device for the sensor data (e.g sensor node) and RFID data (e.g passive RFID tag). In the case of the former, there is no such well-established device. In the latter case, we find the additional problem of how to match the ID or EPC of the passive tag with the sensor device ID so we can associate the sensor reading with the reusable asset and hence with the products transported with it. For example, imagine that the EPC Network has recorded all the aggregation events along the supply chain of trays with the meat products. Therefore we know which product has been in which tray and when. The OGC SWE's SOS has collected sensor data from all the trays, and has assigned to the sensors in those trays some ID, either automatically or by some kind of strategy as explained earlier in section 4. We need to be able to match the sensor ID with the tray ID in order to infer the condition of those meat products. In this respect, we find the following possibilities:

1. The company in charge of managing the reusable assets makes sure that the sensor device in the tray knows the EPC of the tag in the tray. A mechanism is set so this EPC is assigned as the ID of the tray sensor.

2. The company in charge of managing the reusable assets assigns any ID to the sensor device in the tray, but makes sure that there is a networked and automatic way of matching both IDs. An example could be an EPCIS server discoverable via ONS containing this correspondence between the EPC for each tray provided by the company and the IDs of the sensors that it contains. Ideally the sensor IDs should be globally unique in this situation.
3. The matching is done using location and time of the readings. The problem with this approach is that there must be an unequivocal match of time and place for both the data captured by the OGC SWE gateway and the RFID reader. This assumption might be reasonable for reading one tray at a time, making sure that both RF antennas only capture data from that tray. However, if multiple trays are read at the same time, it is not possible to distinguish which sensor data came from where in an accurate way.

5.1.3. Multiple sensors

When a sensor is registered with the OGC SWE, it does not need to do so as an individual sensor, but it might register a sensor system (e.g sensor node) which can provide several sensor readings. This registry will have a single sensorID, no matter from how many physical sensor devices the information comes from. This is analogous to the concept of a 'logical reader' at the ALE layer of the EPC Network or a 'readPoint' at the EPCIS layer of the EPC Network.

- **Reusable assets:** Option number 2 in 5.1.2. could provide more flexibility if several sensor devices were to be installed in the same reusable asset (e.g. large containers or pallets). In this case, it is still advisable to use a single reusable asset ID, although several passive tags with the same ID might be distributed around the asset to improve the reading success. However, it might be problematic to assign the same ID to more than one sensor device. Firstly, multiple registrations with the same ID are not supported, and a mechanism should be devised in order to allow only one registration to the SOS. Secondly, information arriving to the SOS with the same sensor ID would be treated as coming from the same device, and stored as such. This could result in problems such as different locations, clock synchronization and others. Furthermore, any top-down communication to the sensor devices would be rendered unusable unless the same message should be transmitted to all the sensor devices. Obviously, in any case the sensor devices must be identical so the registration information is the same. A registration as a single sensor system could also be possible and would allow the sensor devices to be different, but a mechanism at the base station should be put in place to register the system and merge the messages of the various sensor devices in a single report. Option 2 would allow association of any number of sensor device IDs with a single reusable asset ID. Option 2 thus appears as the more flexible and feasible of the alternatives.
- **Products:** We might want to attach several sensor devices to a single product in a similar way to how we attach several sensor devices to a single reusable asset. The discussion and conclusions of this are also similar to what was discussed in section 5.1.2.. The only difference is that the EPCIS containing the matching table would

have to be managed by either the manufacturer who tags the products, or some third party in charge of installing and managing the sensor devices.

Figure4 depicts the players that would be necessary using option number 2 and several sensors per product or reusable asset.

5.1.4. Structure of the matching repository for multiple sensors

Figure 4 shows how clients can match the EPCs of the products/reusable assets and sensor devices by querying a repository held by the manufacturer/logistics company (i.e. matching repository). Although that repository could be of any kind, as long as it answers the question of which sensor devices belong to which reusable asset or product, figure 4 suggests the use of an EPCIS repository. The main reason for this suggestion is that something similar to EPCIS aggregation events could be also utilized to record the association of products/reusable asset IDs with the IDs of the sensor devices attached to them. In this way, standard and existing methods can be used as well for the matching repository and the interfaces that offer its services to clients and repository holders. Furthermore, because ONS supports multiple service types (and Discovery Services are likely to do so), EPCIS can be easily referenced by existing EPC Network services and thus be integrated inside the EPC Network architecture which already exists in the proposed integrated implementation.

An interesting discussion can be held around the semantics of the existing EPCIS aggregation events and the purpose of the repository. EPCIS aggregation events have been initially designed to track physical 'containment' relationships along the supply chain. Although the relationship between e.g a pallet and the sensors located on it is also a physical relationship, the aggregation is more likely to happen only once at the beginning of the supply chain, and in the case of reusable assets or location, probably remain like that over a period of time throughout many supply chain cycles of many products carried by the reusable asset. Of course, using EPCIS aggregation events would allow one to change this relationship at any point (e.g. a sensor is broken in transport and replaced in some middle-point), which is nevertheless beneficial in terms of flexibility. However, there might be problems of interpretation by application software when all the items inside a reusable asset have been removed (e.g. it receives an EPCIS aggregation event with action field set to 'DELETE' and childEPCs field set to null), since they might think that also the sensors associated with the reusable asset have been removed, even when it is not the case.

Given the difference in semantics, it might be appropriate to define a new sub-type of EPCIS event that is structurally similar to an aggregationEvent but that would encompass our intended meaning; the semantics of this new sub-type of EPCIS event would not be the same as the aggregation event issued when items are added or removed from the reusable asset, so no confusion about removal of sensors could occur. For the sake of clarity, we could call this new event *association* event instead of *aggregation* event. For fixed sensors, the *parentID* could also be a bizLocation (rather than the event simply occurring at a bizLocation), and this case would represent an association between one or more sensors and a location of the type discussed in section 5.1.1. “Ambient Sensors”. For sensors attached to reusable assets, the *parentID* could be the EPC or ID of the reusable asset, while the *childEPCs* field could contain a list of the IDs of one or more sensors associated with that asset.

Other fields such as bizStep, disposition, or bizTransactionList would probably not be used in this new type of aggregation type, although they are already optional fields in the EPCIS specification.

6. Conclusion

We have seen how an independent installation of the EPC Network and the OGC SWE allows a client to retrieve supply chain data based both on ID and sensor conditions in an integrated way. The only additional development needed for this approach to work is an orchestration engine able to access both architectures and interpret the retrieved data based both in location and time. Very limited or no additional modification of current standards are needed. A cattle meat supply chain was used as an example, providing all the expected functionality of a condition-based supply chain, including the generation of alerts in real-time and the track and trace of meat products based on ID and sensor data. We have also seen the complexity of various implementation approaches to the system, and discussed how sensors can be distributed either in the environment surrounding the monitoring products, in reusable assets where the products are transported or attached to the products themselves. We concluded that the most flexible approach involves an additional networked repository where identities of sensors and products can be cross-referenced, since this allows the use of unlimited sensor devices per product. We note that in principle these associations between sensor IDs and EPCs of physical objects could even be recorded within EPCIS repositories, as an associationEvent, a proposed new subtype of EPCIS event, similar to an aggregationEvent, but with subtly different semantics.

References:

[EPCnetwork] Ref. EPCglobal Network Architecture Framework Document, http://www.epcglobalinc.org/standards/architecture

[ALE] Application Level Events
http://www.epcglobalinc.org/standards/ale

[EPCIS] EPC Information Services
http://www.epcglobalinc.org/standards/epcis

[ONS] Object Naming Service
http://www.epcglobalinc.org/standards/ons

[EPCs] Electronic Product Codes – defined in EPCglobal Tag Data Standard
http://www.epcglobalinc.org/standards/tds

[OGCSWE] OGC® Sensor Web Enablement: Overview And High Level Architecture, OGC White paper. Version 3, Date: 2007-12-28

[SOS] Sensor Observation Service. Date: 2007-10-26, Version: 1.0

[SAS] OGC® Sensor Alert Service Implementation Specification. Date: 2007-05-14, Version: 0.9.0 (Candidate OpenGIS® interface standard)

[WNS] Draft OpenGIS® Web Notification Service Implementation Specification. Date: : 2006-11-18

[CS-W] OpenGIS® Catalogue Services Specification, Implementation Specification, Date: 2007-02-23

[TML] OpenGIS® Transducer Markup Language (TML) Implementation Specification. Date: 2007-07-02, version 1.0.0

[O&M] Observations and Measurements – Part 1 - Observation schema. Date: 2007-12-08, version 1.0

[SensorML] OpenGIS® Sensor Model Language (SensorML) Implementation Specification. Date: 2007-07-17, version 1.0

[SPS] OpenGIS® Sensor Planning Service Implementation Specification. Date: 2007-08-02, Version: 1.0

[RFID] Ron Weinstein, “RFID: Atechnical Overview and Its Application to the Enterprise”, IT Professional, Volume 7, Issue 3, May-June 2005 Page(s):27 - 33

[BRIDGE] The BRIDGE project, www.bridge-project.eu, accessed 11/01/2010

[SupplyChain] Beth Bacheldor , \emph{RFID, Sensor Technologies Can Build Smarter Supply Chains, IBM Says}, RFID Journal, March 2009, http://www.rfidjournal.com/article/view/4653, accessed 11/01/2010

[EPCsensor] J. Sung, T. Sanchez Lopez, and D. Kim, \emph{The EPC Sensor Network for RFID and WSN Integration Infrastructure} PerComW 2007, IEEE Computer Society, 2007, 618-621

[1451] Eugene Y. Song and Kang Lee, \emph{Understanding IEEE 1451—Networked SmartTransducer Interface Standard}, IEEE Instrumentation \& Measurement Magazine, April 2008

[Regulation178] Regulation (ec) no 178/2002 of the european parliament and of the council, 28 January 2002