A Distributed Architecture of ONS/DS for the tracking of product in EPCglobal supply chain

August 07, 2010
Ubiquitous Computing Database Lab.
Mahbubur Rahman
<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Environment</td>
</tr>
<tr>
<td>Overview of the Discovery Service</td>
</tr>
<tr>
<td>Define Problem</td>
</tr>
<tr>
<td>Considerable Architecture</td>
</tr>
<tr>
<td>Proposed Architecture</td>
</tr>
<tr>
<td>Conclusion</td>
</tr>
</tbody>
</table>
## Table of Contents

- **Target Environment**
- **Overview of the Discovery Service**
- **Define Problem**
- **Considerable Architecture**
- **Proposed Architecture**
- **Conclusion**
I. Target Environment

- **EPC supply chain**
  - Products move from manufacturer to consumer
  - Each part in the supply chain keep information about the products
Table of Contents

- Target Environment
- Overview of the Discovery Service
- Define Problem
- Considerable Architecture
- Proposed Architecture
- Conclusion
II. Overview of Discovery Service

◆ EPCglobal Supply chain
  ➔ Various kinds of event occurred in a product's life cycle
  ➔ Different EPCIS keeps various event information of a specific product

Core product info
- Description
- Brand name
- Color
- Height
- Weight
- Exp. Date
- Man. Date
- etc

All events are on the same product
II. Overview of Discovery Service

- **Search for Information of the product**
  - Retailer need to know the info of the product

How to find the info of the product?
II. Overview of Discovery Service

- ONS compliant to DNS can find the location

DNS (Domain Name Server)
- Root Domain Name Server
- CO
- JP
- KR
- ACM
- OR
- Yahoo
- Naver
- XXX

ONS (Object Naming Service)
- Company 1 (Samsung)
- Company 2 (LG)
- Company 3 (SONY)
- Product A (LCD)
- Product B (PDP)
- Product C (Laptop)

Core product info
- Description
- Brand name
- Color
- Height
- Weight
- Exp. Date
- Man. Date
- etc

How about events?
II. Overview of Discovery Service

жение Service
- Can find serial level info
- Accumulate info from various EPCIS

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>packaged</td>
<td>2010-08-07:10:59:00KST</td>
<td>kimhae</td>
</tr>
<tr>
<td>received</td>
<td>2010-08-07:08:41:00KST</td>
<td>kimhae</td>
</tr>
<tr>
<td>shipped</td>
<td>2010-08-07:00:12:12KST</td>
<td>kimhae</td>
</tr>
<tr>
<td>received</td>
<td>2010-08-07:01:25:21KST</td>
<td>Busan</td>
</tr>
<tr>
<td>pick</td>
<td>2010-08-07:02:02:00KST</td>
<td>Busan</td>
</tr>
<tr>
<td>received</td>
<td>2010-08-07:01:01:00KST</td>
<td>Seoul</td>
</tr>
</tbody>
</table>
Table of Contents

- Target Environment
- Overview of the Discovery Service
- Define Problem
- Considerable Architecture
- Proposed Architecture
- Conclusion
III. Define Problem

◆ **EPCglobal’s Standard**
  ➔ No standard but only some functional Specifications

- Many important Roles
- No Standard
III. Define Problem

✦ EPCDS Name server

1. Name server function as Local ONS.
2. This server response with the correct EPCDS of the epc.
3. When query, it also response with correct EPCDS address.

➤ This is a kind of centralized
➤ Does not maintain EPCglobal standard
III. Define Problem

- LIT developed
  → Centralized system which has to process events of the connected EPCISs

Information are stored in text files.

- No use of database.
- Avoid concept of root ONS.
- Suffer from huge performance problem
III. Define Problem

✦ Discovery Service
  ➔ Can find serial level info
  ➔ Accumulate info from various EPCIS

<table>
<thead>
<tr>
<th>EPC RECORD</th>
<th>EPCIS Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>8552:4250:320</td>
<td><a href="http://www.epcis1.com/ws">www.epcis1.com/ws</a></td>
</tr>
<tr>
<td>8552:4250:320</td>
<td><a href="http://www.epcis2.com/ws">www.epcis2.com/ws</a></td>
</tr>
<tr>
<td>8552:4250:320</td>
<td><a href="http://www.epcis3.com/ws">www.epcis3.com/ws</a></td>
</tr>
</tbody>
</table>

EPCIS Sony DS
EPCIS LGDS
EPCIS DellDS
Table of Contents

- Target Environment
- Overview of the Discovery Service
- Define Problem
- Considerable Architecture
- Proposed Architecture
- Conclusion
IV. Considerable Architecture

- **DS architecture based on region**
  - Each considerable region should maintain one DS

- **Advantages**
  - Local info are kept in local EPCDS
  - No of EPCDS is limited

- **Disadvantages**
  - Need to search all EPCDS
  - If user increase, processing time increase
IV. Considerable Architecture

- Wal-mart business area
  - 4081 store in USA and 2091 store outside USA
IV. Considerable Architecture

- Wal-mart region based EPCDS
  - Certain no of EPCDS need to be deployed

No of user query at the same time
No of query processing concurrently
IV. Considerable Architecture

- Other retailers (Tesco-Homeplus)
  ➔ More than 250 retailers are running on the world market
IV. Considerable Architecture

- Combined of all retailer
  ➔ Can combine and share where more than one retailer works

- Can handle all with a certain no of EPCDS
- No of query to Every DS will be much higher
- So performance will be degraded
IV. Considerable Architecture

- **DS architecture based on Company**
  - Each company should maintain a DS

**Advantages**
- Tag id and ONS system is proposed based on the company code
- For tracing, only one DS have to be accessed.

**Disadvantages**
- Only problem is that each company should maintain a DS server
IV. Considerable Architecture

❄ Wal-Mart Supplier

→ 60,000+ supplier all over the world

- Each company need to EPCDS
- As supplier increase, EPCDS increase
- User get services using one EPCDS
- Each EPCDS need to process less query concurrently
- ONS importance is much more
IV. Considerable Architecture

◆ Wal-Mart Supplier
  ➔ 60,000+ supplier all over the world

- More than 250 retailer all over the world
- Same company is supplier of more than one retailer
Table of Contents

- Target Environment
- Overview of the Discovery Service
- Define Problem
- Considerable Architecture
- Proposed Architecture
- Conclusion
V. Proposed Architecture

- **DS architecture based on Company**
  - Each company should maintain a DS

**Advantages**
- Compliance to EPCglobal ONS Standard
- No each company need to deploy.
- Global relation can be maintained.

**Disadvantages**
- Need to include the name server.
V. Proposed Architecture

DS architecture based on Company

- Each company should maintain a DS

- EPCDS load decrease as EPCDS distributed
- Centralized Performance Problem resolved

- Including Name Server handle the product which does not have EPCDS
VI. Conclusion

❖ In this work I have tried to
  ➔ Find out the problem of the EPCDS.
  ➔ Describe two distributed architecture of ONS-DS and their pros & Cons.
  ➔ Proposed an architecture combined with Company based & Name server.