

SASE: Complex Event Processing Over Streams

Daniel Gyllstrom, Eugene Wu, Hee-Jin
Chae, Yanlei Diao, Gordon Anderson,
and Patrick Stahlberg



Complex Event Processing

- **High-volume event streams**
 - Sensing devices
 - Financial services
 - Network monitors
 - ...

- **Complex event processing (CEP)**
 - ⊕ Filtering
 - ⊕ Correlation
 - ⊕ Aggregation
 - ⊕ Transformation



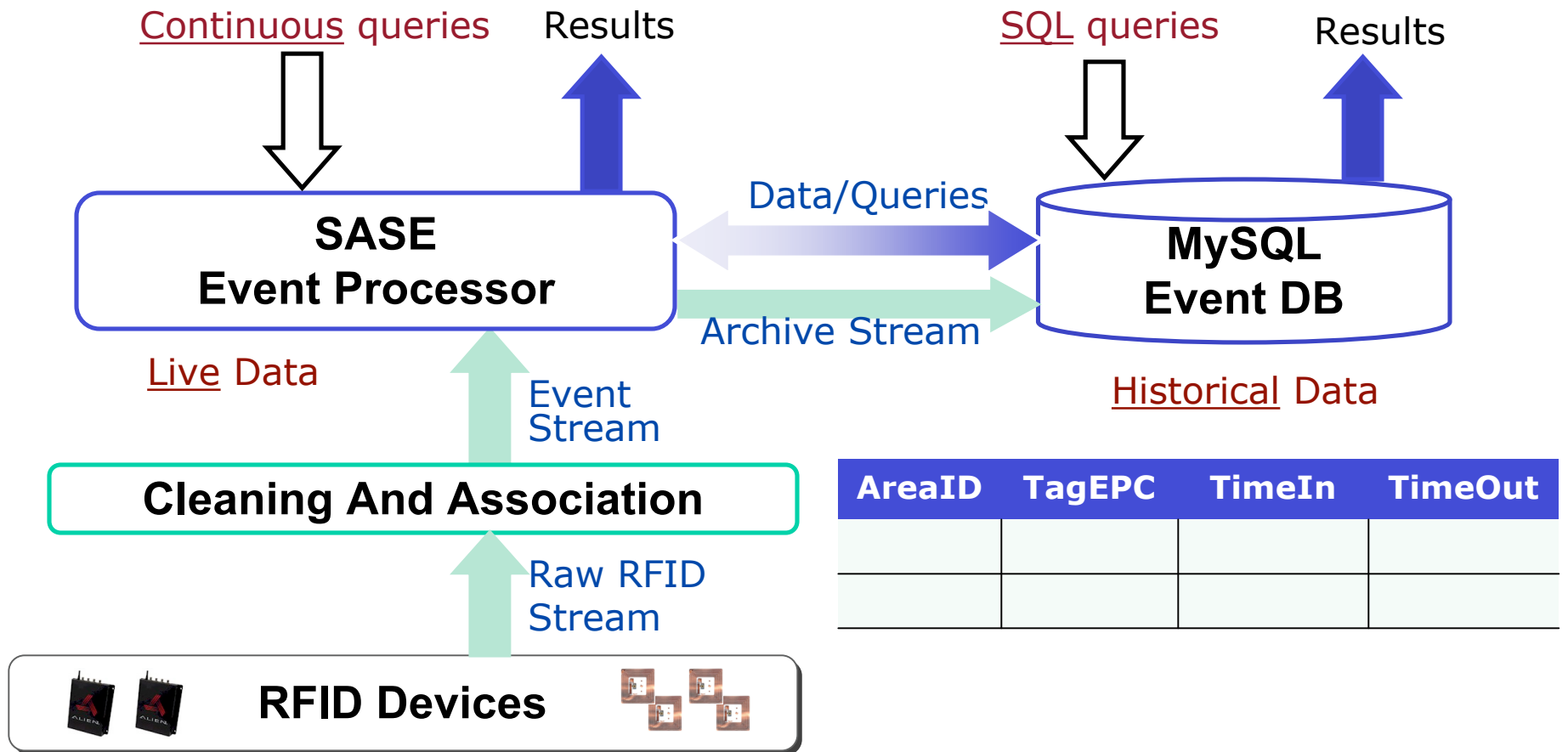
The SASE Event Processor

- SASE: A declarative event language, formal semantics, and an efficient implementation
 - ⊕ Sequencing
 - ⊕ Negation
 - ⊕ Kleene closure
 - ⊕ Parameterized predicates
 - ⊕ Sliding window...
- Related Systems
 - Relational stream systems
 - Active databases
 - Commercial event systems
 - ...

Unwieldy for use

Unable to express

Demo Architecture



Shoplifting

“Item seen at a shelf and then at an exit, but not at any register in between.”

```
EVENT      SEQ(Shelf_Reading x,  
              !(Register_Reading y),  
              Exit_Reading z)  
WHERE      x.TagId = z.TagId AND x.TagId = y.TagID  
           /* Equivlanent to [TagId] */  
WITHIN     12 HOURS  
RETURN     x.TagId, x.ProdName,  
           x.AreaId, z.AreaId,  
           retrieveLocationName(z.AreaId)
```

Shoplifting Screenshot

The screenshot displays the 'RFID Retail Store Demo' application interface. The main window shows a query editor with the following SQL-like query:

```

EVENT      SEQ(Shelf_Reading [x],!(Register_Reading)[y], Exit_Reading[z])
WHERE      [TagId]
WITHIN     12 HOURS
RETURN     x.TagId, x.ProdName, x.Areald, z.Areald, _retrieveLocation(z.Areald)
    
```

Below the query editor, an **ALERT:** section displays the following information:

```

There is a Thief!
-----
Item: LIGHT-GREEN
TagId: 0x76CEA5A50000000061062087054
Location: Retail Exit 1
    
```

At the bottom of the main window, there are four buttons: 'Run Stream Query', 'Run Database Query', 'Reset', and 'Stop Query'.

Two other windows are visible on the right side of the screen:

- Cleaning and Association Layer Output Events:** This window displays a list of events, including:
 - 0x76CEA5a00000000610162087054|1|Shelf_Reading|Light-Green|Gen
 - 0x74CEA5a00000000610162012321|1|Exit_Reading|Light-Green|Genc
 - 0x96CEA5a00000000610162324323|2|Shelf_Reading|Light-Green|En
 - 0x86CEA5a00000000610162055454|3|Counter_Reading|Blue|MyCorp
 - 0x76CEA5a00000000610162087646|1|Shelf_Reading|Yellow|Gencorp
 - 0x76CEA5345345999493099588255|1|Shelf_Reading|Light-Green|MyC
 - 0x76CEA5a00000000610162087051|4|Shelf_Reading|Light-Green|Gen
 - 0x76CEA5a00000000610162087054|1|Shelf_Reading|Light-Green|Enr
 - 0x74CEA5a00000000610162012321|1|Exit_Reading|Light-Green|Enron
 - 0x96CEA5a00000000610162324323|2|Shelf_Reading|Light-Green|Gen
 - 0x86CEA5a00000000610162055454|1|Counter_Reading|Blue|MyCorp
 - 0x76CEA5a00000000610162087676|4|Shelf_Reading|Yellow|Enron
 - 0x76CEA5345345999493099588255|5|Shelf_Reading|Light-Green|UB
 - 0x76CEA5a00000000610162087084|1|Shelf_Reading|Light-Green|MyC
- Stream Processor Query Results:** This window shows the result of the query:


```
Result:
(0x76CEA5A50000000061062087054, LIGHT-GREEN, 1, 4, Retail Exit 1)
```

Misplaced Inventory

“Item seen at Shelf A and then Shelf B, without being purchased or put back to the original shelf afterwards.”

```
EVENT      SEQ(Shelf_Reading x,  
              Shelf_Reading y,  
              !(ANY(Register_Reading, Shelf_Reading) z) )  
  
WHERE      [TagId] AND  
           x.AreaId != y.AreaId AND  
           x.AreaId = z.AreaId  
  
WITHIN     1 minute  
  
RETURN     x.TagId, x.ProdId, x.AreaId, y.AreaId,  
           retrieveHistOfMvmt(x.TagId)
```

Misplaced Inventory Screenshot

The screenshot displays the 'RFID Retail Store Demo' application interface. The main window shows a query editor with the following SQL-like query:

```

EVENT      SEQ(Shelf_Reading [x], Shelf_Reading[y], !(ANY(Register_Reading,Shelf_Reading))[z])
WHERE      [TagId] AND
           x.Areald != y.Areald AND
           x.Areald = z.Areald
WITHIN     1 MINUTE
RETURN     x.TagId, x.ProdName, x.Areald, y.Areald, _retrieveHistOfMvmt(x.TagId)
    
```

Below the query editor, an 'ALERT:' section is visible, indicating a 'Misplaced Inventory!' for a yellow item. The details are as follows:

Item:	Yellow
TagId:	0x53A6A5A500000006101622072340
Movement History:	2 (2007 - 01-09 20:25:13.0) 3 (2007 - 01-09 20:27:22.0) 2 (2007 - 01-09 20:27:57.0) 1 (2007 - 01-09 20:28:57.0)

At the bottom of the main window, there are four buttons: 'Run Stream Query', 'Run Database Query', 'Reset', and 'Stop Query'. To the right, there are two smaller windows: 'Cleaning and Association Layer Output Events' showing a list of event logs, and 'Stream Processor Query Results' showing the output of the query, including the item ID, color, and movement history.

Questions



S
A
S
E