Ereignisgesteuerte Architekturen und Echtzeitdatenintegration

Roland Stühmer, Dominik Riemer

INFORMATIONSINTEGRATION UND WEBPORTALE

MOTIVATION

fast data is a trend!

“continuous access and processing of events and data in real-time”

Source: ams-ix.net

real-time data everywhere

Source: publicpolicy.telefonica.com

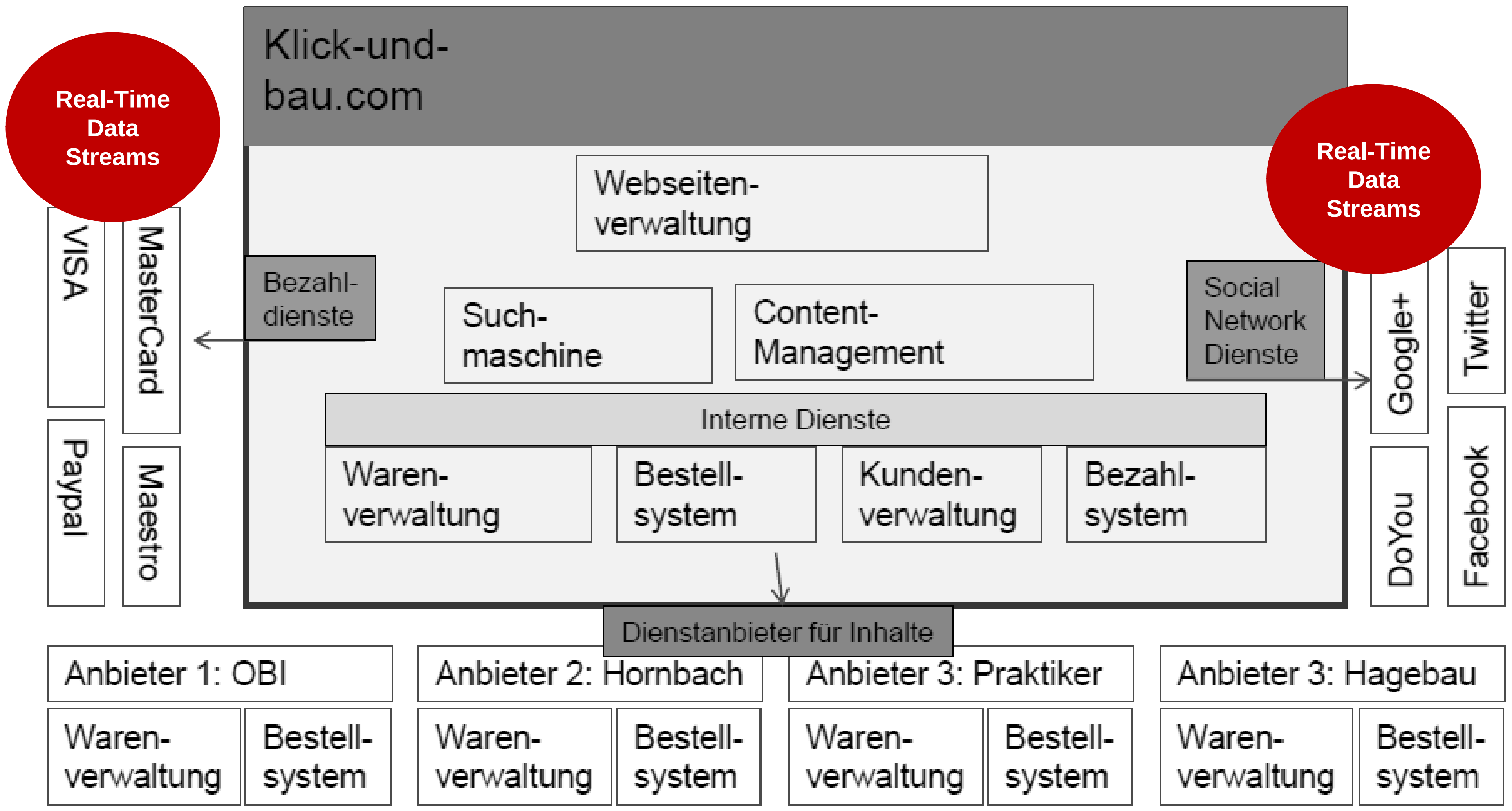
immediate insights

real-time awareness

Source: commons.wikimedia.org

ad-hoc reactions...

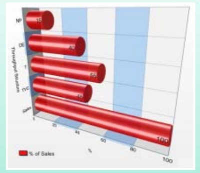
Event-Driven Architecture



Today's Topics

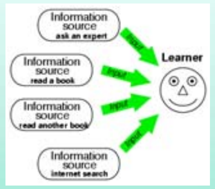
- Real-time (Business Real-Time)

- High Throughput of Events



- Complex Relations between Events
  (Event Operators)

- Combining different event sources
  (loose Coupling)


Information source ask an expert
Information source read a book
Information source read another book
Information source internet search
Learner

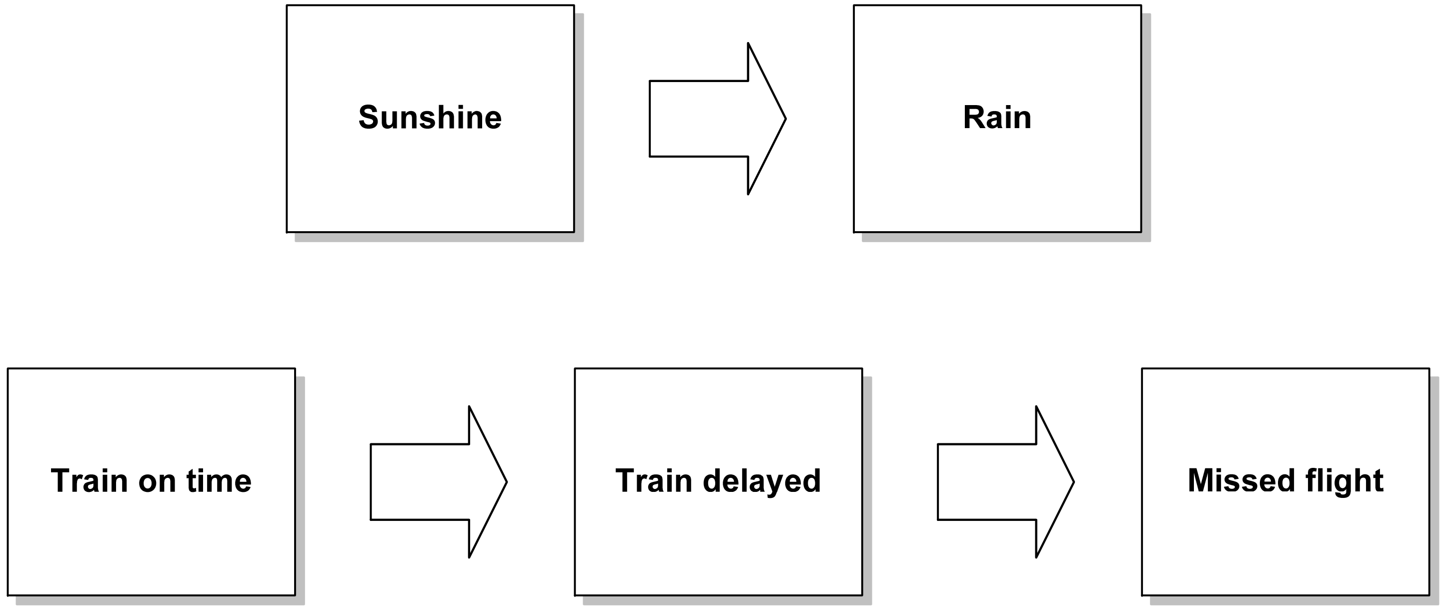
EVENTS

What is an Event?

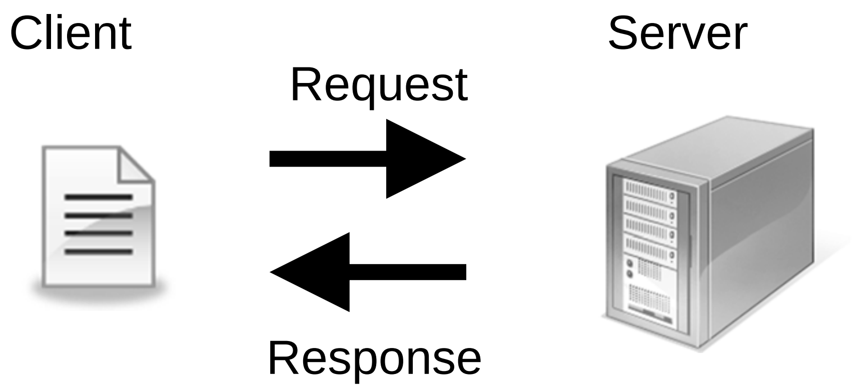
- An Event is an **occurrence** within a particular system or domain; the word event has double meaning: the real-world occurrence as well as its computerized representation
- Represents changes / occurrences
- Data with a timestamp
- Information container
- Triggers actions

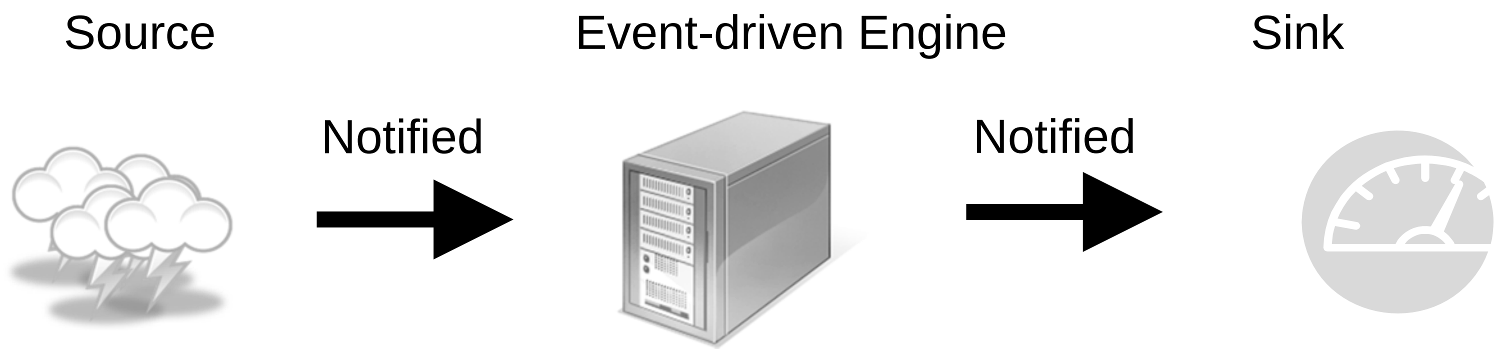
Etzion, Niblett (2011)

An Event Is a Significant State Change!



Time- & Request-driven vs. Event-driven



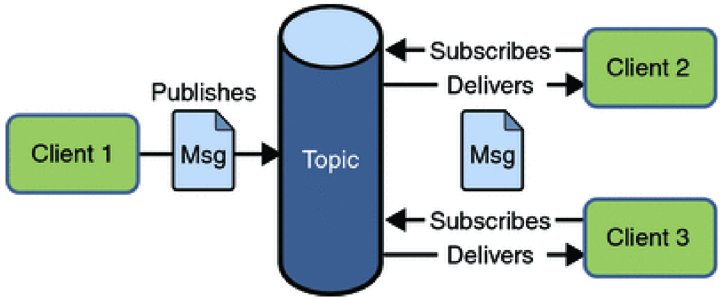


EVENT-DRIVEN ARCHITECTURE

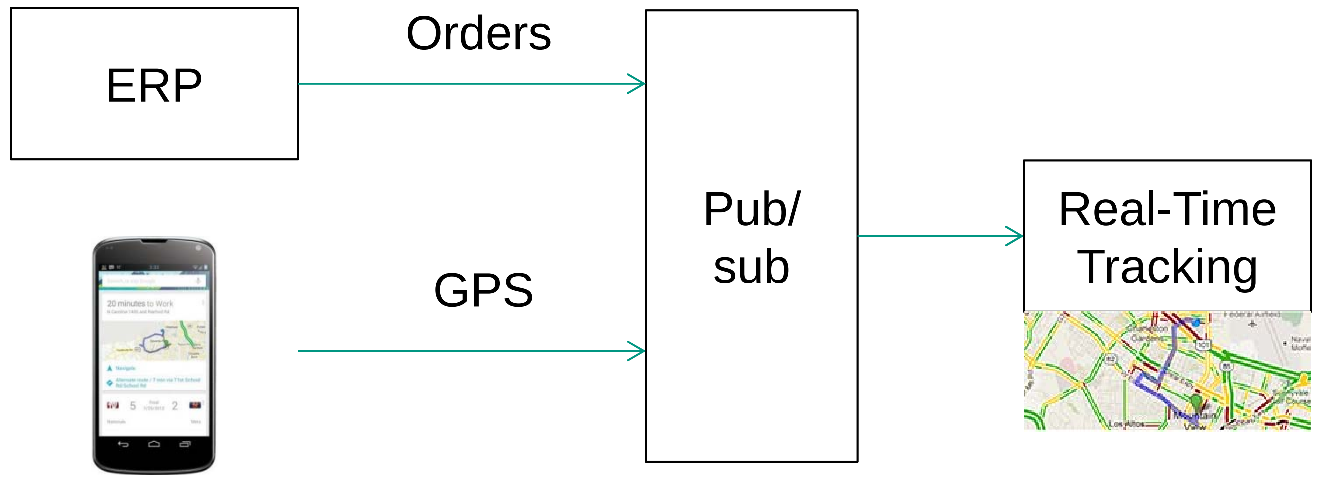
Publish-/Subscribe Middleware

- Message Broker (Middleware)
  - Component to decouple sources and sinks
- Multiple Consumers
  - Sources address messages to a topic (→ bulletin board)
  - Arbitrary number of subscribers can listen to topic
  - One-to-many (1:n) communication
- Timing
  - Consumers receive the message after it is published,
  - after they subscribed to the topic, and
  - only while they are active.

→ **Pub/sub is basic design pattern for Event-driven Architectures (EDA)**



EDA Application Scenario



EDA Application Characteristics

- Sense and Respond
  - Timely response when reality deviates from expectation
- Asynchrony
  - Timing of events are not controlled
  - communication is usually unidirectional
- Global situational awareness
  - Awareness by correlating multiple sources of data
  - from outside the enterprise with enterprise data

EDA Principles

- Report current events as they happen
- Push notifications
- Producer decides when to send
- Respond immediately
- One-way communication; fire-and-forget
- Free of commands; not prescriptive
- Decoupling of sender and receiver

Chandy, Schulte (2009)

SOA Principles

- Modular, distributed, discoverable, reusable components are collaborators
  - Order process (*client*) calls address verification *service* which returns whether the address is valid
- Time of interaction determined by client
- Service protocols and schemas are well defined, often with transactional semantics
- Units obtain global situational awareness by invoking multiple services

Chandy, Schulte (2009)

Comparison of SOA and EDA

Attribute	SOA	EDA
Design focus	Services and service interfaces	Events and event message
Processing model	Client invokes functionality	Source sends message to middleware, sink receives message
Modularity	Clients/ services	Event source/ event sink
Communication pattern	Request/ response	Publish/ subscribe
Transaction control	Client	Independent
Relation between components	Client knows task, name, address; service has to be available; only service knows implementation	Source and sink do not know each other; source does not know if a sink exists
Dependencies between components	Interfaces, versions, SLA	Event schema, version
Degree of coupling	Loose coupling	Extremely loose coupling

Bruns, Dunkel (2010)

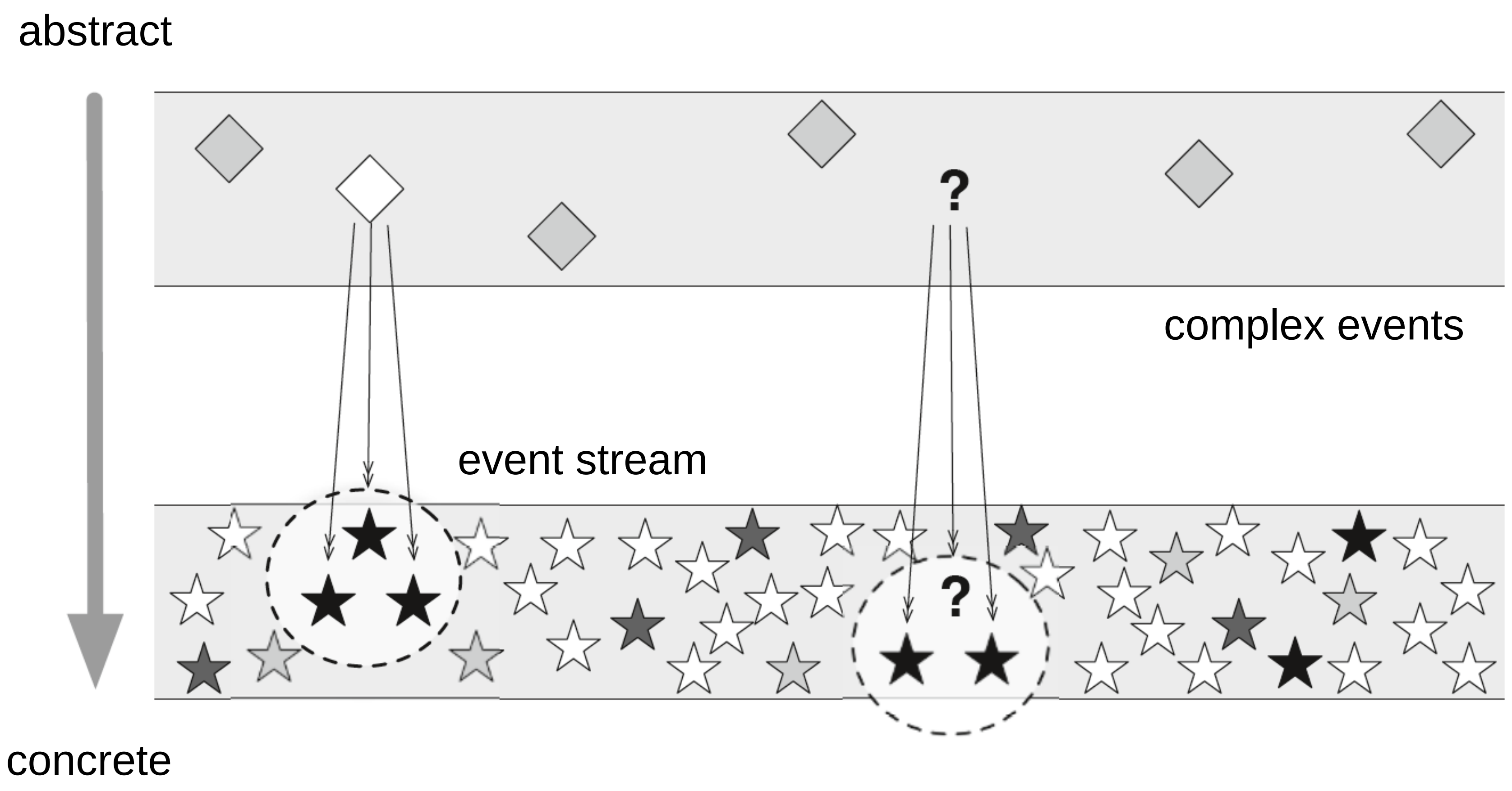
EVENT PROCESSING

When to Use Event Processing

- The application…
  - …is naturally centered on events
  - …needs to identify and react to certain situations
  - …extends an existing application in a flexible, non-invasive manner
  - …can separate intermediary event processing logic
  - …needs to analyse large amounts of data which can be organized in streams
  - …has potential scalability and fault tolerance benefits when using event processing.

Etzion, Nibblett (2011)

From Singular Events to Complex Events



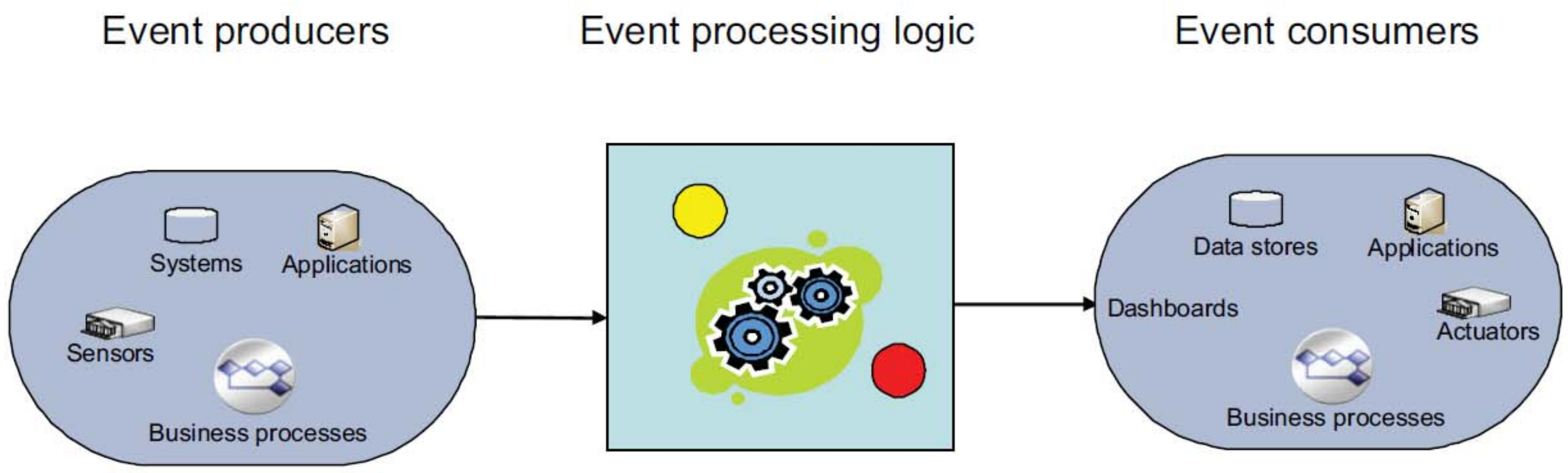
Bruns, Dunkel (2010)

Working With Events

- Event Processing…
  - …is computing that performs operations on events. Common event processing operations include reading, creating, transforming, and deleting events.
  - …is a set of techniques and tools to help us understand and control event-driven information systems.
  - …is an enabling technology that supports on the fly, (business-) real-time processing of huge event streams
  - …is about a timely (or ahead of time) recognition of situations of interest and corresponding reaction

Etzion, Niblett (2011)

The Structure of Event Processing Applications



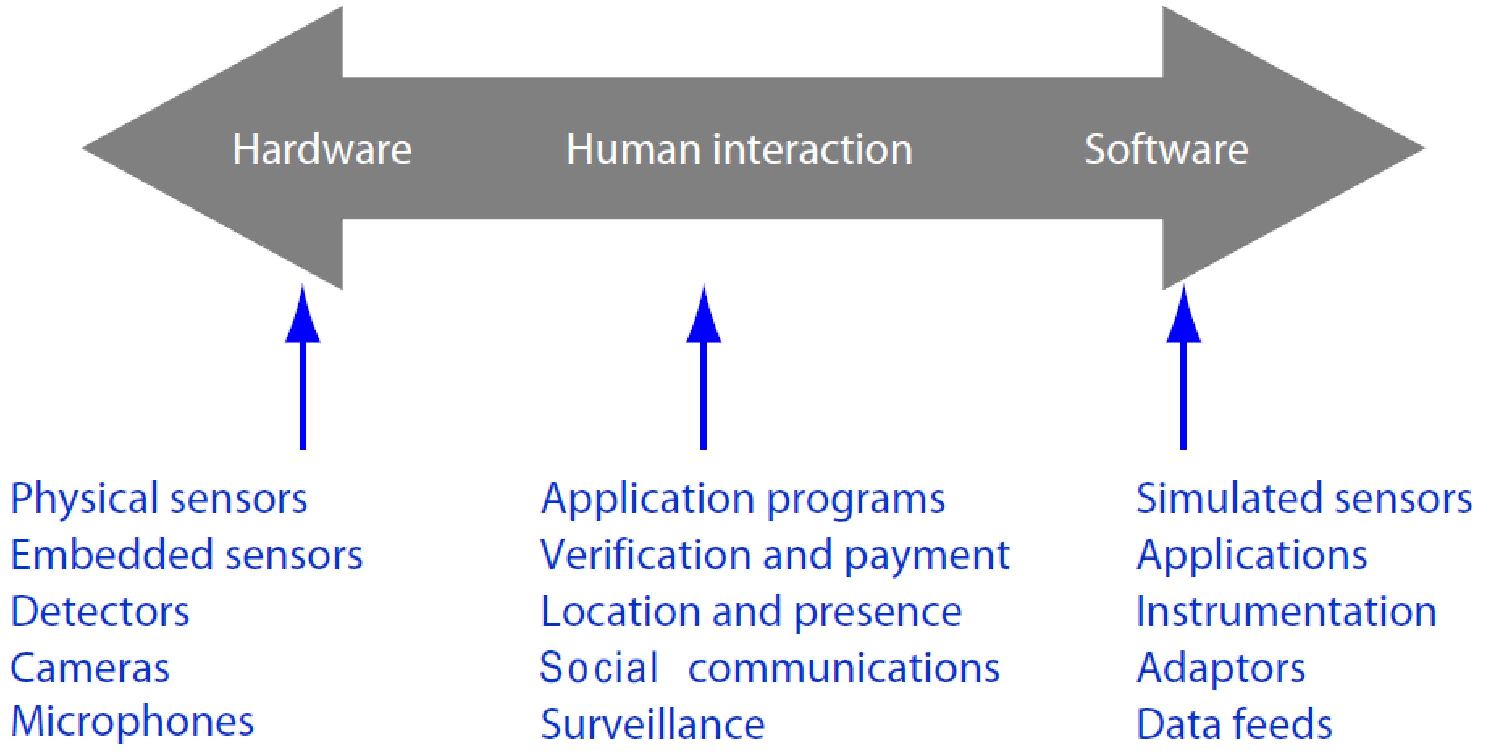
Etzion, Nibblett (2011)

Event Producer

- (Any) Execution Environment
  - …which produces events

- Event Publisher
  - (Pre-)Filter events
  - Format events
  - Publish events

Examples for Event Producers

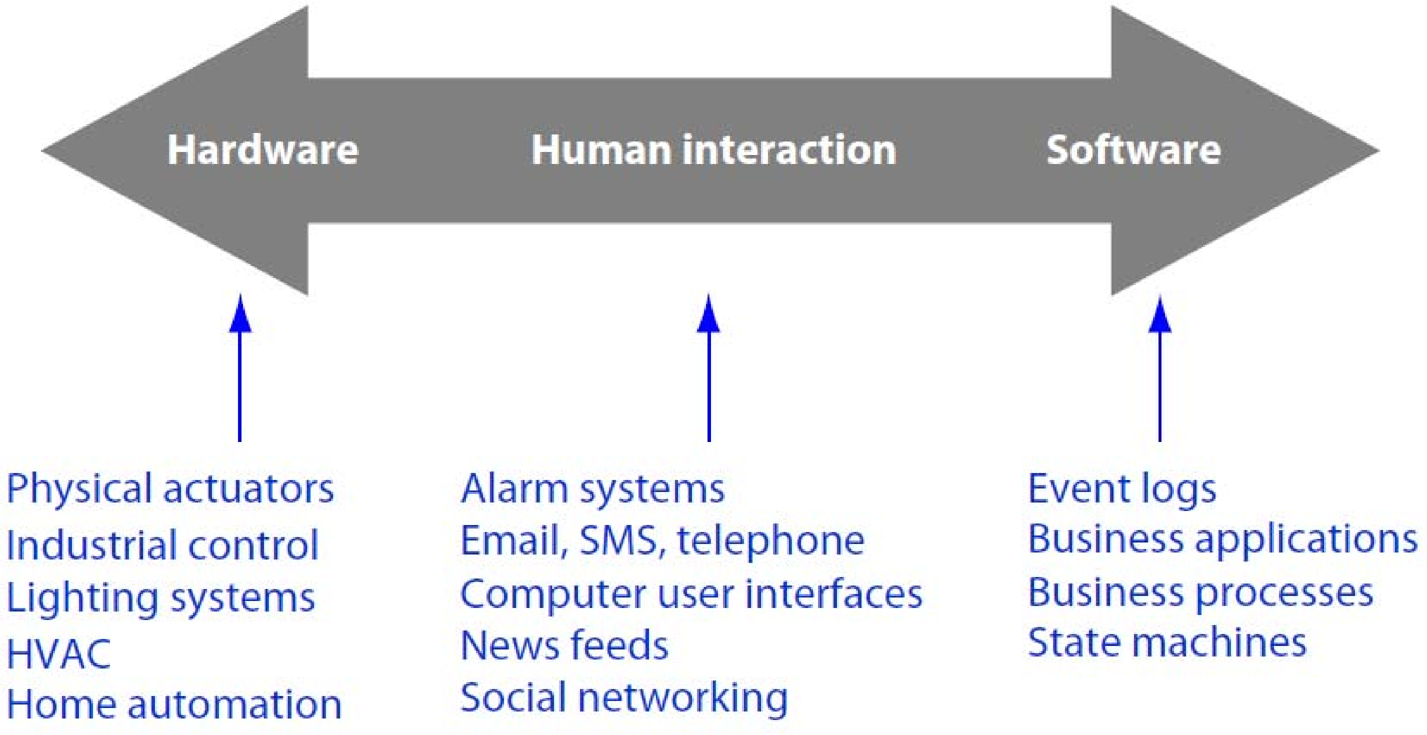


Etzion, Niblett (2011)

Event Consumer

- Input Adapter
  - Receives events and transforms them into internal format
- (Any) Execution Environment
  - Visualizes or acts upon an event
- Typical Consumers
  - Real-time monitoring dashboards
  - Messaging infrastructure (SMS, e-mail, IM, …)
  - Business software

Examples for Event Consumers



Etzion, Niblett (2011)

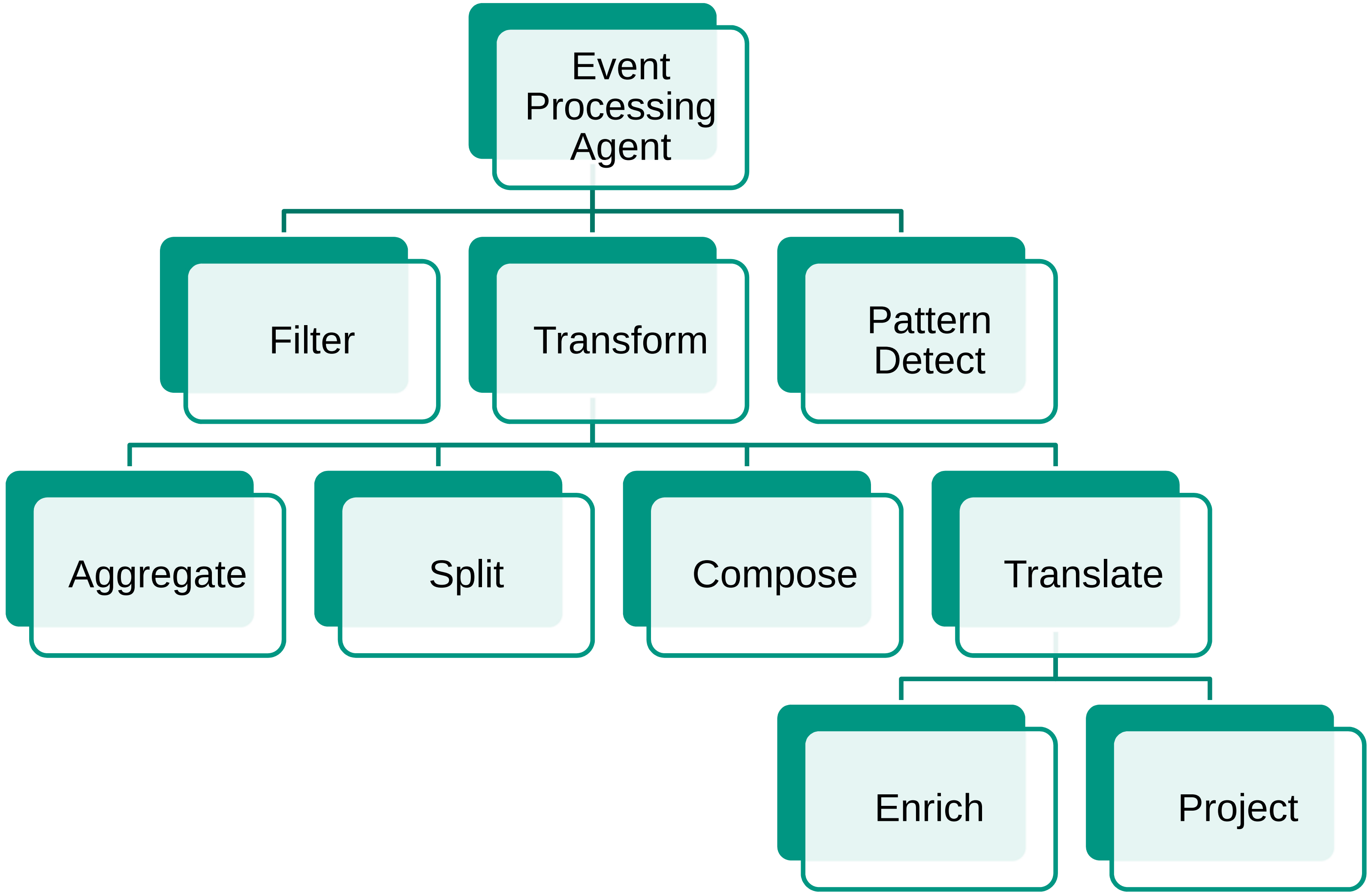
Event Processing Agent

- Input Adapter
  - transforms events into an internal format and puts events into input event stream

- Event Processing Network (EPN)
  - is composed of Event Processing Agents (EPA)
  - EPA monitor events streams to detect and act on events
  - EPA filter, match, and derive (translate, aggregate, split,…)

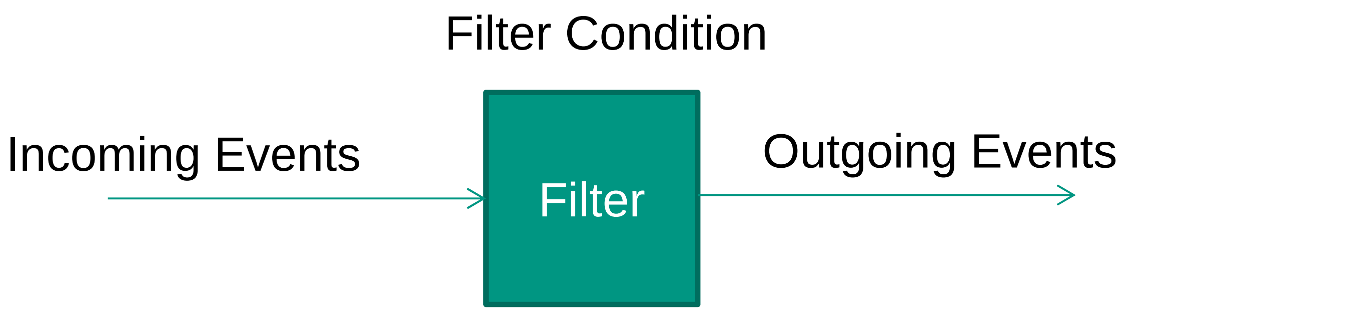
- Output Adapter
  - translates events into metrics, messages or function calls

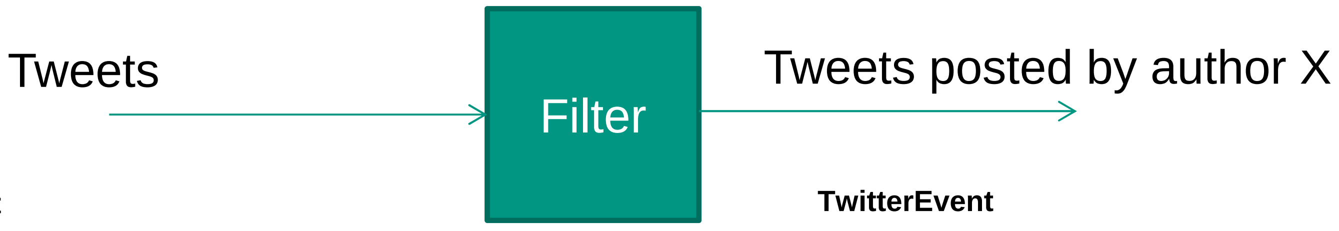
Event Processing Agent Operations



Etzion, Niblett (2011)

Filter EPA





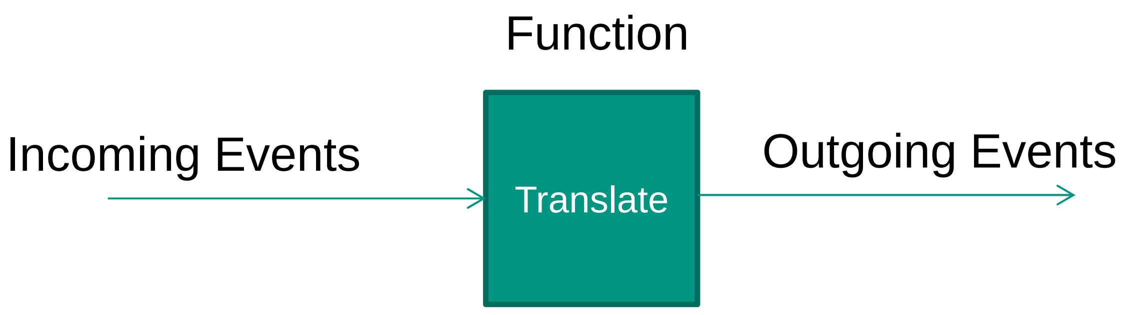
TwitterEvent

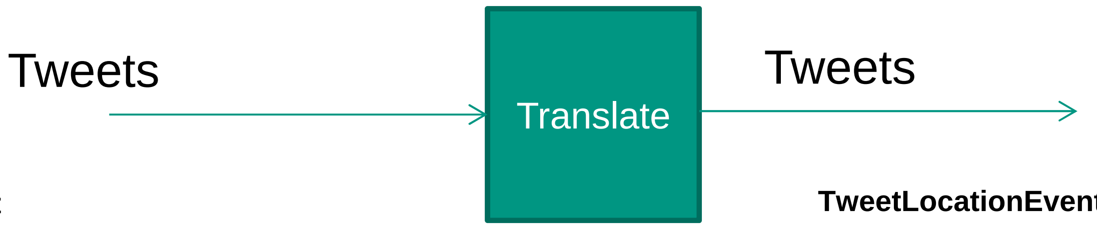
timestamp
author
latitude
longitude
text

TwitterEvent

timestamp
author
latitude
longitude
text

Translate EPA





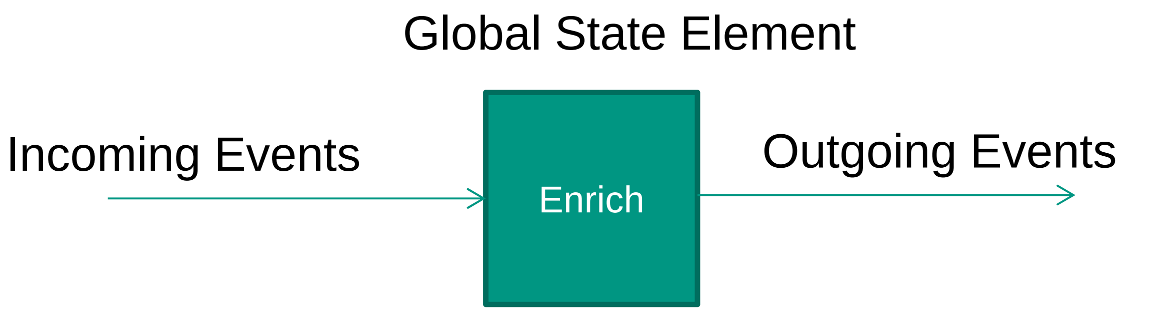
TwitterEvent

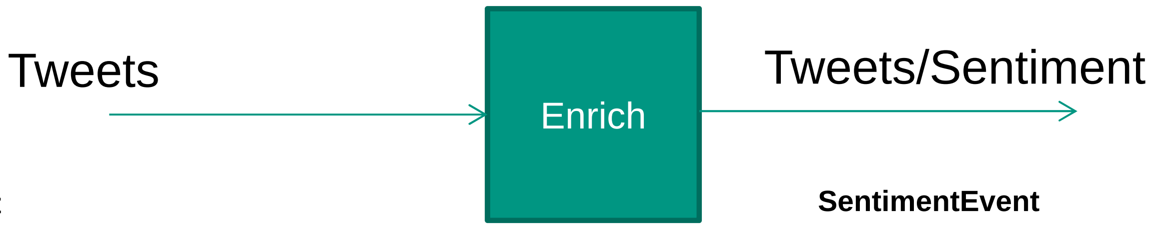
timestamp
author
latitude
longitude
text

TweetLocationEvent

timestamp
author
city
text

Enrich EPA





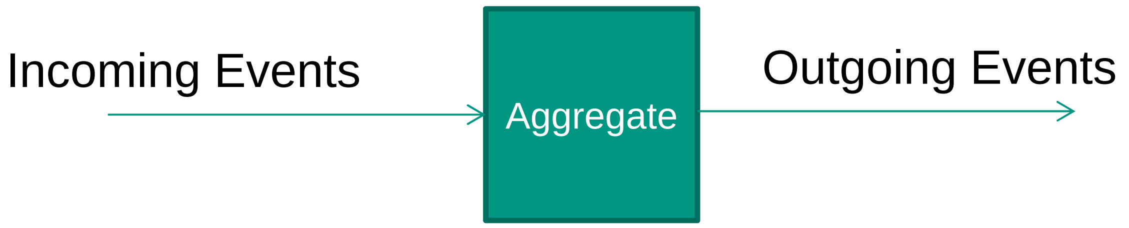
TwitterEvent

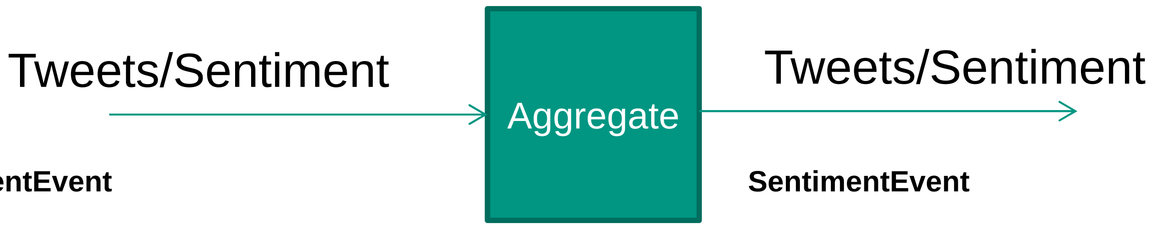
timestamp
author
latitude
longitude
text

SentimentEvent

timestamp
author
latitude
longitude
text
sentiment

Aggregate EPA





SentimentEvent

timestamp
author
latitude
longitude
text
sentiment

SentimentEvent

timestamp
avgSentiment

Pattern Detect EPA

- Definition
  - „A *Pattern detect* EPA is an EPA that performs a pattern matching function on one or more input streams. It emits one or more derived events if it detects an occurrence of the specified pattern in the input events."
- Event Pattern
  - "An *event pattern* is a template specifying one or more combinations of events. Given any collection of events, you may be able to find one or more subsets of those events that match a particular pattern."

Pattern Detect EPA: Basic Patterns

- Pattern **all**
  - **Matching Set:** one event for each type in the participant set
  - **Example:** FlightBooked AND HotelBooked AND CarReserved
- Pattern **any**
  - **Matching Set:** one matching event
  - **Example:** LotteryWin OR HouseSold OR LoanAdvanced
- Pattern **absence**
  - **Matching Set:** empty event type
  - **Example:** FlightBooked AND NOT HotelBooked

Pattern Detect EPA: Temporal Patterns

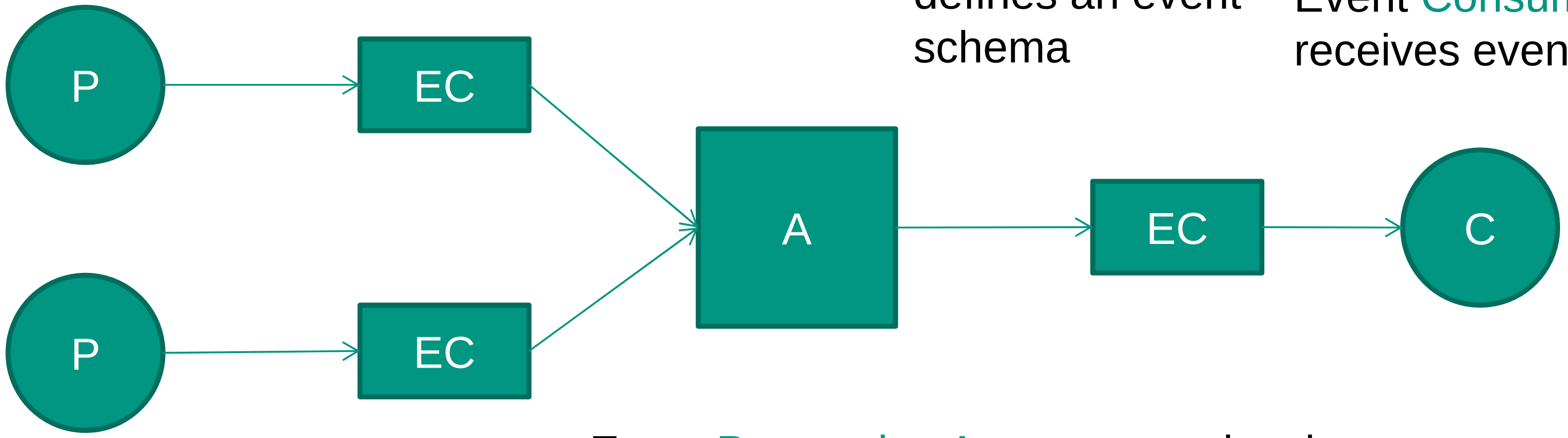
- Pattern **sequence**
  - **Matching Set:** one event for each type in the participant set
  - **Example:** a:CCTransaction(u=1) FOLLOWED BY b:CCTransaction(u=1) WHERE dist(a.loc,b.loc) > 100 WITHIN 5min
- Pattern **increasing**
  - **Matching Set:** entire participant event set
  - **Example:** e1:HeartRate FOLLOWED BY e2:HeartRate WHERE e2.v > e1.v
- Pattern **decreasing**
  - **Matching Set:** entire participant event set
  - **Example:** e1:HeartRate FOLLOWED BY e2:HeartRate WHERE e1.v > e2.v

Event Processing Network

Event Channel – routes events

Event Type – defines an event schema

Event Consumer – receives events

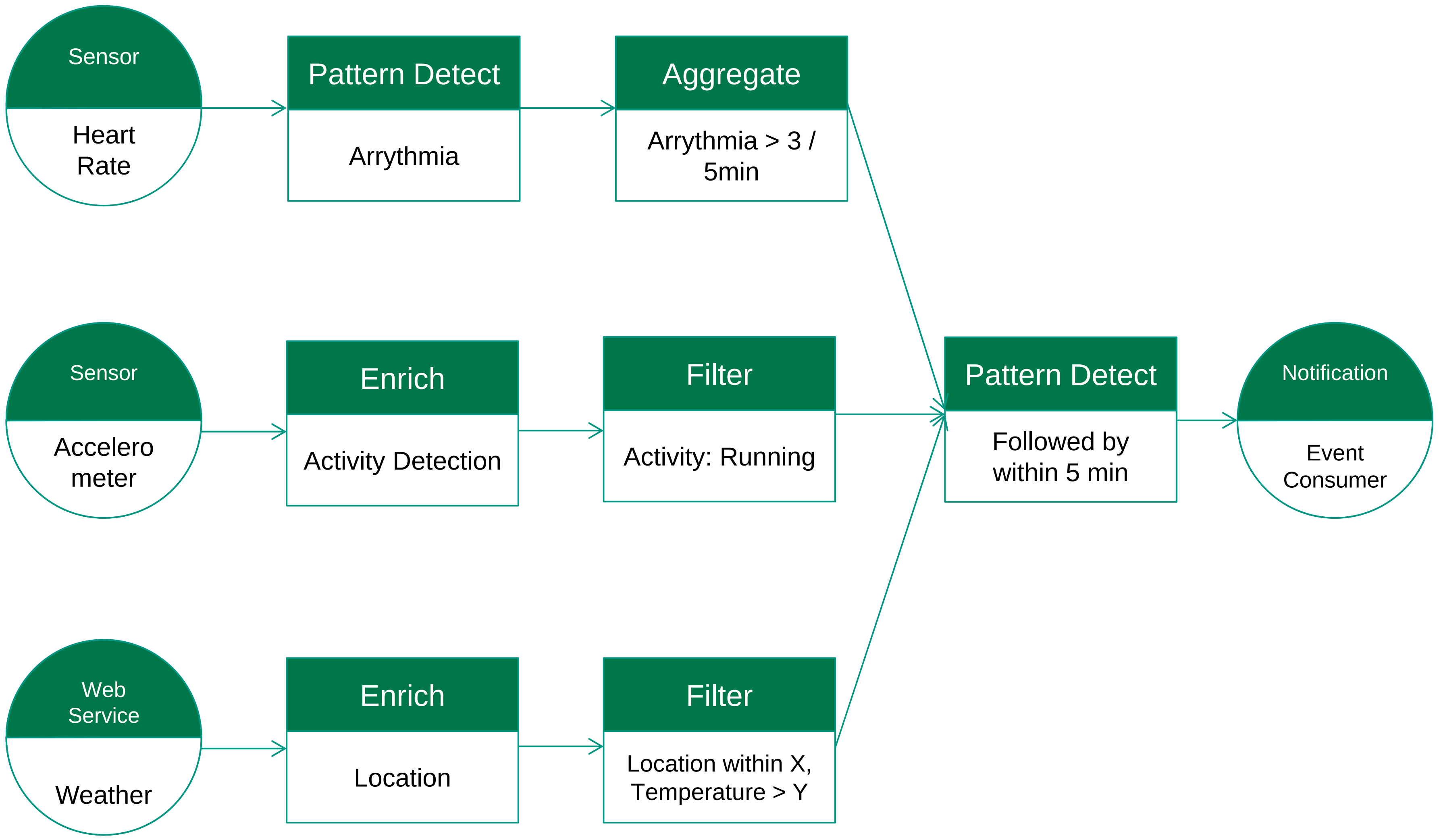


Event Processing Agent – contains the application’s event processing logic

Event Producer – sends events

Etzion, Nibblett (2011)

Example: Event Processing Network

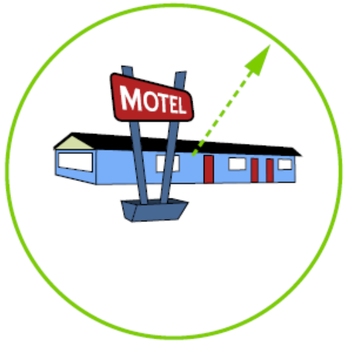


Event Context: Spatial Context

Fixed Location

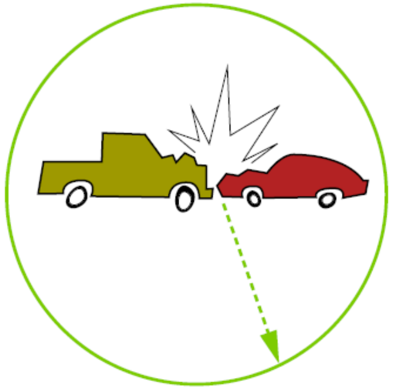
Within the house

Entity Distance Location



Within 2 km from the motel

Event Distance Location



Within 10 km from the accident

Etzion, Niblett (2011)

Event Processing Engine: Esper

- Patterns are represented by a pattern language
  - Rule-based
  - SQL-based

- Patterns are evaluated:
  - continuously (until they are unregistered from the engine)
  - on potentially unbounded event streams

- Esper
  - Open Source Event Stream Processing Engine
  - Versions: Java, .NET
  - Event Representation: POJO, Map, XML
  - Pattern Representation: Esper Event Pattern Language (EPL), *SQL-based*
  - http://esper.codehaus.org

Recap SQL: Basics

- **SELECT**
```
    [ALL | DISTINCT] (column{,column}) | *
FROM
    table [alias]{,table [alias]}
[WHERE constraint]
```

- **WHERE** constraints
  - BETWEEN
  - LIKE (_ or %)
  - NULL
  - IN
  - EXISTS, ALL, ANY (Subquery)

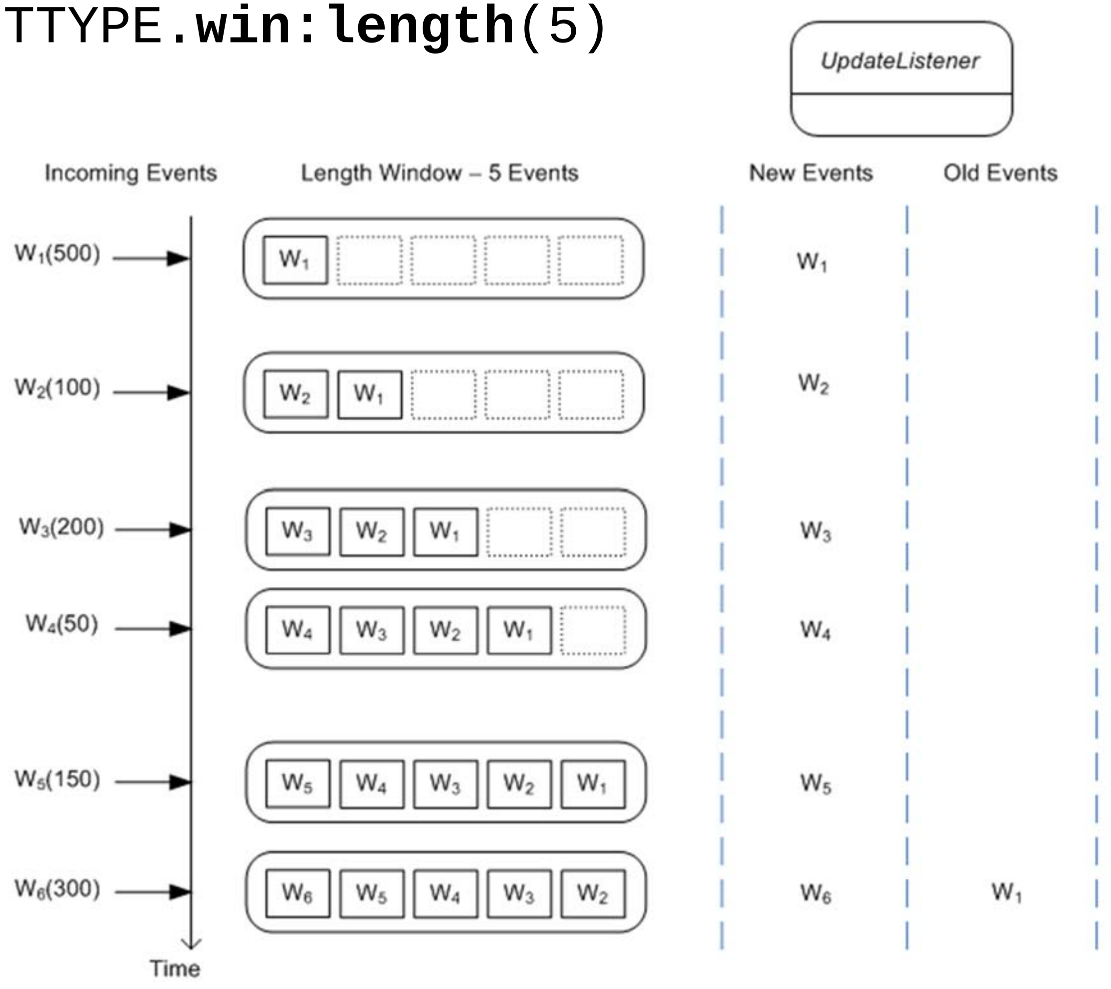
Esper: EPL Syntax

- `[annotations]`
- `[expression_declarations]`
- `[context context_name]`
- `[insert into insert_into_def]`
- **`select select_list`**
- **`from stream_def [as name] [, stream_def [as name]] [,...]`**
- **`[where search_conditions]`**
- `[group by grouping_expression_list]`
- `[having grouping_search_conditions]`
- `[output output_specification]`
- `[order by order_by_expression_list]`
- `[limit num_rows]`

Insert and Remove

- `select * from EVENTTYPE.`**`win:length`**`(5)`

- Every time an event of the type EVENTTYPE occurs, it is picked up; also the last four events are picked up as well

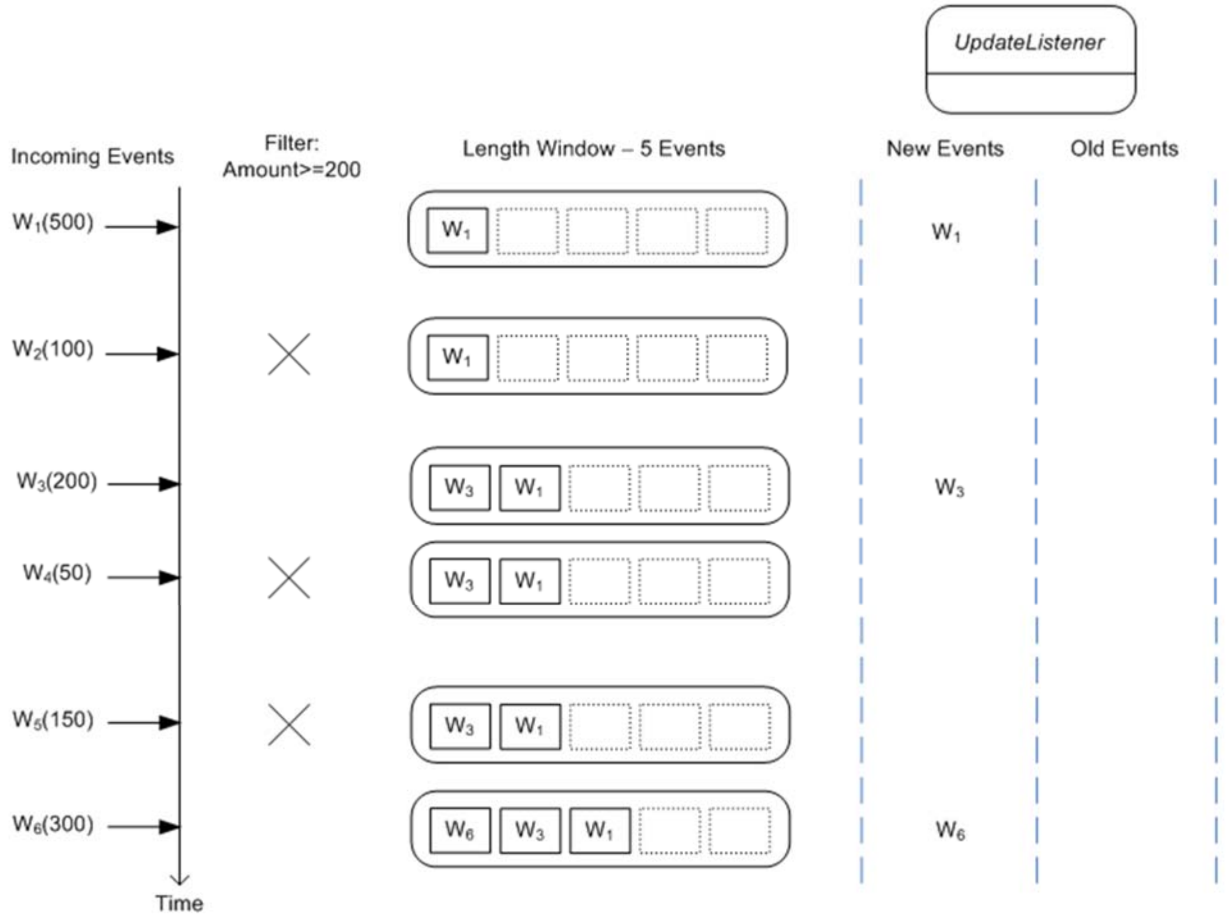


http://esper.codehaus.org/esper-4.9.0/doc/reference/en-US/html/processingmodel.html

Filters and Where Clauses

- `select * from EVENTTYPE(amount>=200).win:length(5)`

- Every time an event of the type EVENTTYPE occurs, which amount is 200 or more, it is picked up; also the last four events are picked up as well

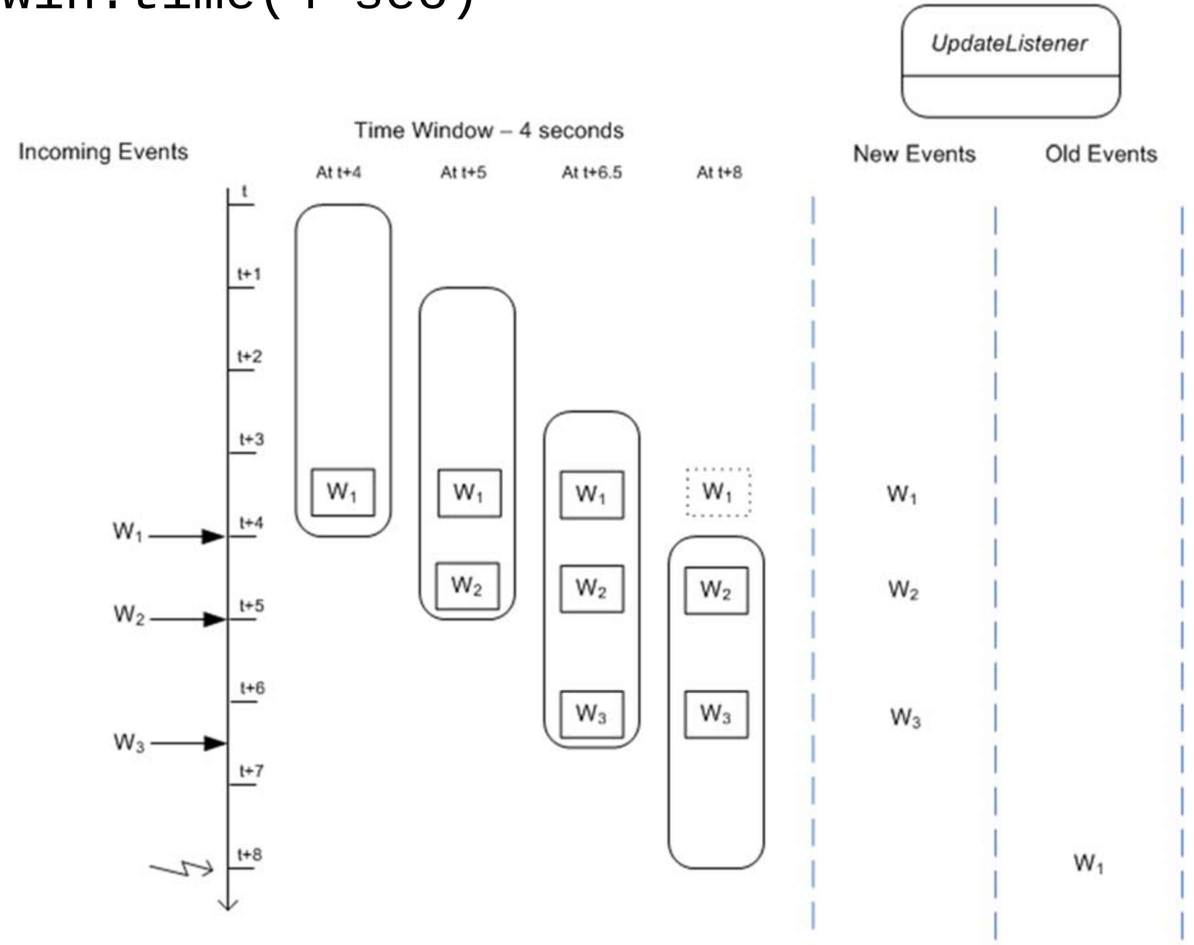


http://esper.codehaus.org/esper-4.9.0/doc/reference/en-US/html/processingmodel.html

Time Window

- `select * from ET.win:time(4 sec)`

- Every time an event of the type EVENTTYPE occurs, the events of the last four seconds are picked up as well



http://esper.codehaus.org/esper-4.9.0/doc/reference/en-US/html/processingmodel.html

Aggregation and Grouping

- `select * from ET.win:time_batch(1 sec)`
  - Un-aggregated and un-grouped

-
  ```
  select sum(amount)
  from ET.win:time_batch(1 sec)
  ```
  - Fully aggregated and un-grouped

-
  ```
  select account, sum(amount)
  from ET.win:time_batch(1 sec)
  ```
  - Aggregated and un-grouped

-
  ```
  select account, sum(amount)
  from ET.win:time_batch(1 sec)
  group by account
  ```
  - Fully aggregated and grouped

http://esper.codehaus.org/esper-4.9.0/doc/reference/en-US/html/processingmodel.html

Event Types

- “*An event type is a specification for a set of event objects that have the same semantic intent and same structure; every event object is considered to be an instance of an event type.*”

Etzion, Niblett (2011)

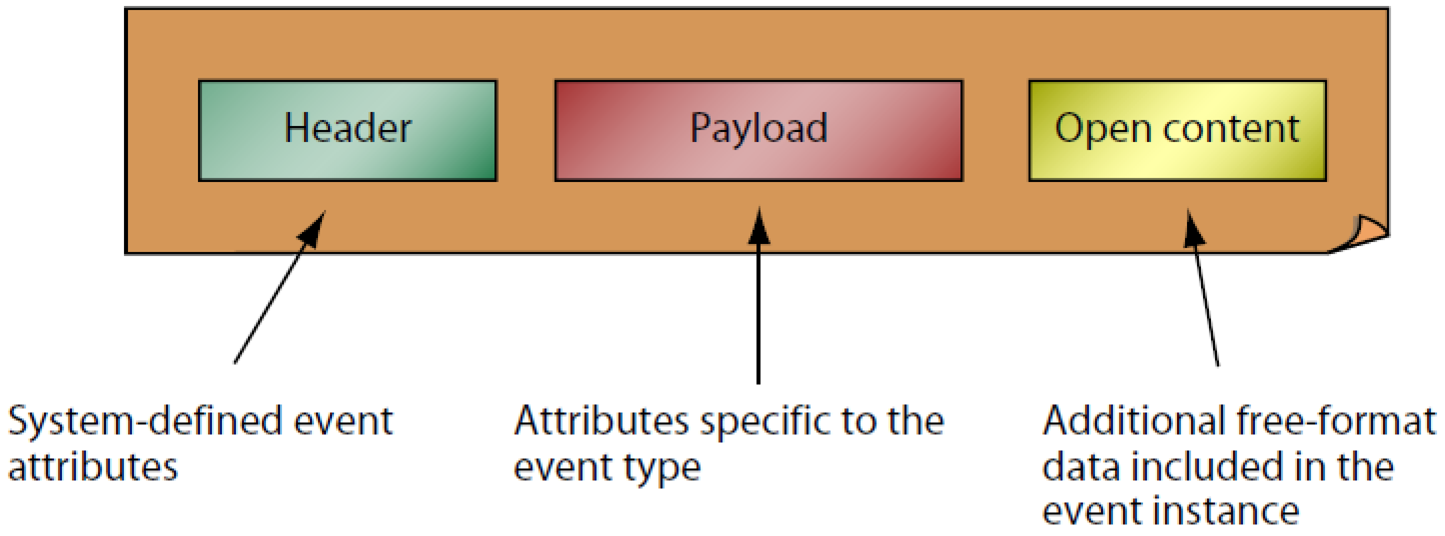
Event Attributes

- An event attribute is a component of the structure of an event. Each attribute has a name and a data type

- Lack of information is a problem
- Surplus information is a burden

Etzion, Niblett (2011)

Event Structure


Header
Payload
Open content
System-defined event attributes
Attributes specific to the event type
Additional free-format data included in the event instance

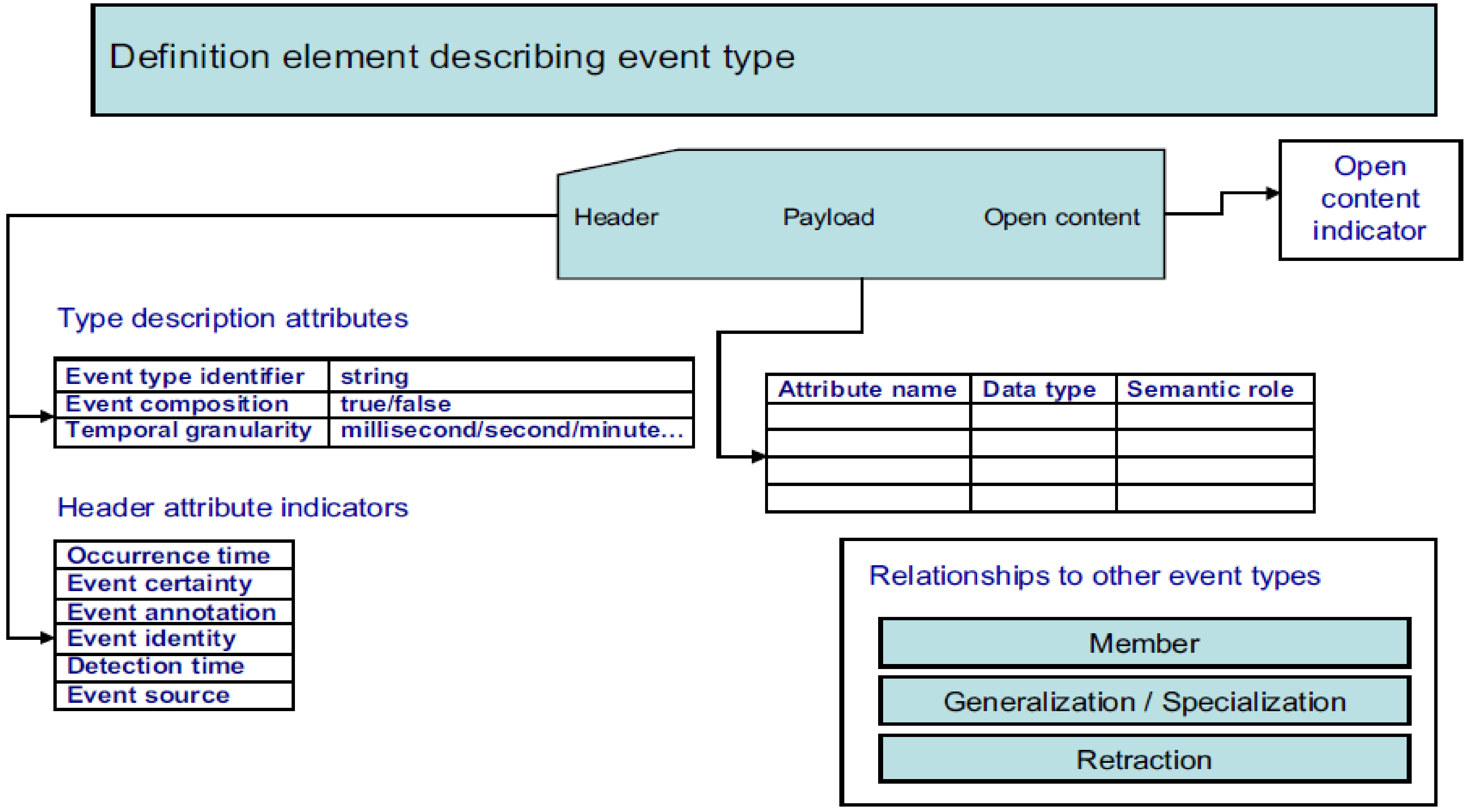
Etzion, Niblett (2011)

Event Type Relations

- Member
  - Instances of this event type can be included in instances of the composite event
- Membership
  - Instances of this event type have the followings members.
- Generalization
- Specialization
- Retraction
  - A property of an event type referencing a second event type, it indicates that the second type is a logical reversal of the event type that references it

Etzion, Niblett (2011)

Event Type Definition


Definition element describing event type
Header
Payload
Open content
Open content indicator
Type description attributes
Header attribute indicators
Relationships to other event types
Member
Generalization / Specialization
Retraction

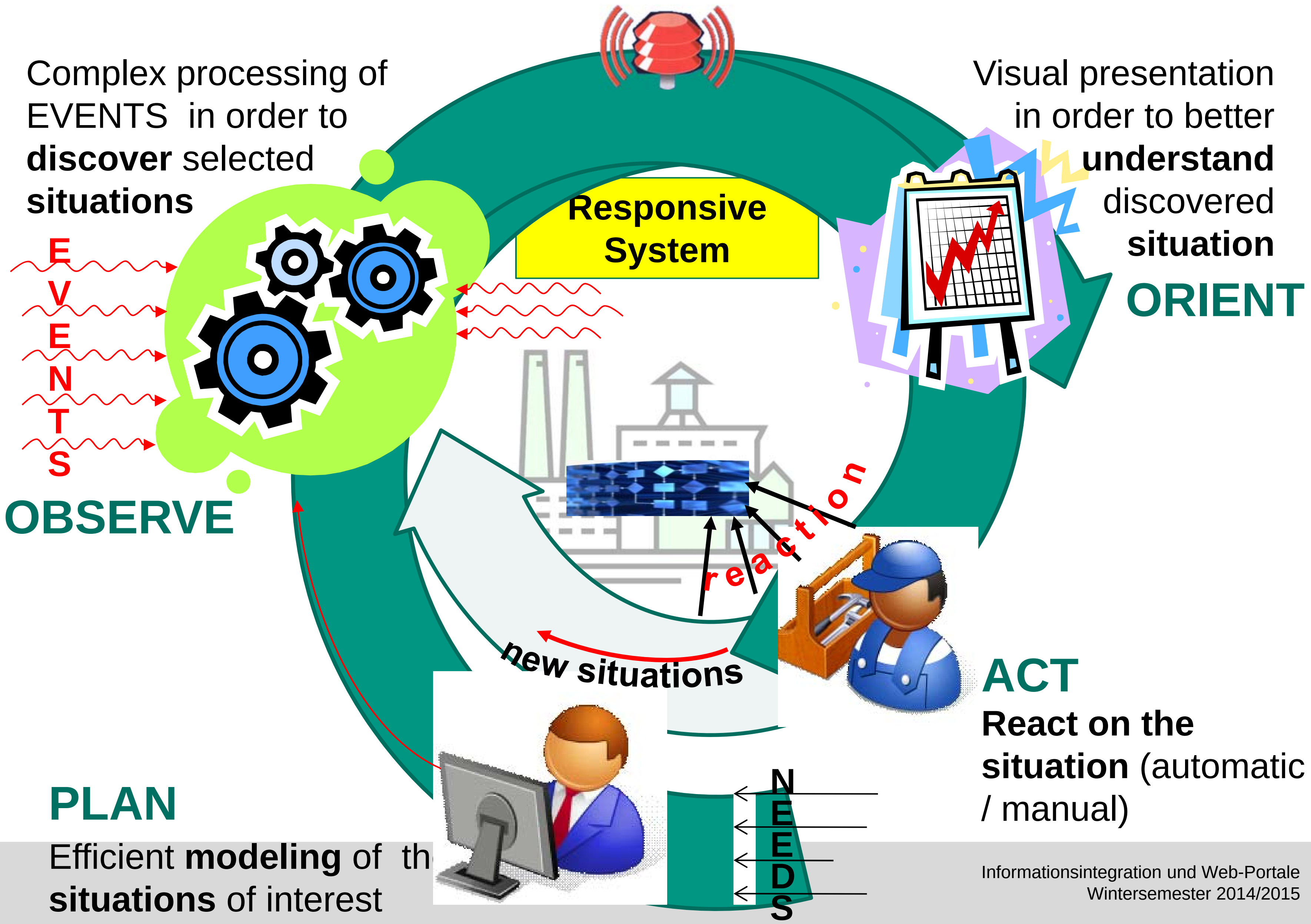
Event type identifier	string
Event composition	true/false
Temporal granularity	millisecond/second/minute…

Occurrence time
Event certainty
Event annotation
Event identity
Detection time
Event source

Attribute name	Data type	Semantic role

Etzion, Niblett (2011)

Research Topic: Process view on responsiveness


Complex processing of EVENTS in order to **discover** selected **situations**
E V E N T S
OBSERVE
Responsive System
Visual presentation in order to better **understand** discovered **situation**
ORIENT
reaction
new situations
ACT
React on the situation (automatic / manual)
PLAN
Efficient **modeling** of th
situations of interest
NEEDS

Event Processing Potential

The role of the real-time push of information has become crucial for many application areas:

- eHealth (e.g. real-time patient monitoring)
- Energy (e.g. real-time energy consumption monitoring)
- Transportation/Logistics (e.g. real-time traffic monitoring)

to name but a few

2. Push of Information is very data-intensive, e.g.

- Twitter is generating up to 15,000 tweets/sec for a topic (avg: 5,700 tweets/sec, numbers for 2013)

3. This overload will be ever “worse” e.g. Real-time Web:

- growing number of resources on the Web move away from traditional request/response communication
- real-time Web technologies:
    - Facebook Graph API supports real-time updates as JSON
    - Google supports push-notifications through PubSubHubbub
    - HTML5 WebSockets can push data to browsers

Why Event Processing?

- Real-time has become one of the crucial characteristics of modern applications and is completely changing the game in the data processing
  - Data is on the move
    - Find results immediately or never
      - *one should be informed as soon as her flight has a delay*
  - Information searches for the relevant consumers
    - instead of searching for information, it should find us
      - *one should be automatically informed as soon as her flight has a delay*
- Google search vs. Twitter followers

Since when does Real-time exist

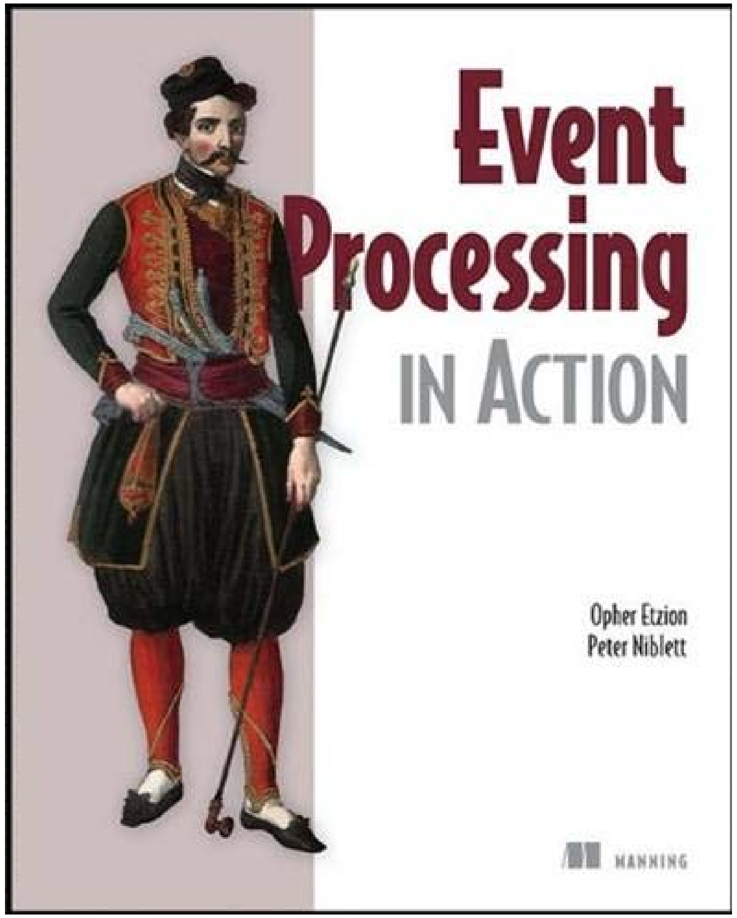
- Real-time is essential for everything we are doing, but we are not aware that it will be possible to:
  - Inform me immediately if my luggage is not onboard and we are about to start (and not after landing)
  - Inform me immediately when two my friends are sitting in the café close to that I am currently sitting
    - Combining different events in the relevant context
  - Inform me immediately after it becomes very likely that there will be jam on my road (but it is not yet)
    - Even predicting the future events

What time is Real-time

- Twitter world record, 29. Aug. 2011
  - Beyoncé's pregnancy announcement during the MTV VMA show resulted in ***8,868 tweets per second***.
  - The previous record was during the final of FIFA Women's World Cup, between Japan and the United States. That resulted in **7,196 tweets** per second
  - In terms of past record events, Bin Laden's death drew a significant peak in Tweets Per Second with 5,106 TPS. Super Bowl 2011 saw 4,064 TPS, and the previous all-time high was New Years Eve 2010 in Japan, which hit 6,939 TPS at its peak.
- Financial market
  - Nanoseconds trading
- eHealth: Remote patient monitoring
  - One semantic signal in 5 sec
- Energy: Smart meters
  - One reading in 15 min
- Real-time in this sense means *business* real-time or *near* real-time

Recommended Reading:

- Opher Etzion and Peter Niblett. **Event Processing in Action**. Manning Publications, 2010



References

- **Etzion, Niblett (2011)**
  Opher Etzion and Peter Niblett. *Event Processing in Action.* Manning Publications, 2010
- **Bruns, Dunkel (2010)**
  Ralf Bruns and Jürgen Dunkel. *Event-Driven Architecture.* Springer-Verlag, 2010
- **Chandy, Schulte (2009)**
  K. Mani Chandy and W. Roy Schulte. *Event Processing: Designing IT Systems for Agile Companies.* McGraw-Hill Osborne Media, 2009

Mögliche Prüfungsfragen

- Nennen Sie Unterschiede zwischen SoA und EDA!
- Erklären Sie den Publish/Subscribe-Nachrichtenaustausch.
- Nennen Sie je eine beispielhafte Anwendung für eine SoA und eine EDA!
- Was sind wichtige Attribute eines Event-Objekts?
- Was sind Bausteine eines Event Processing Networks (EPN)?
- Welche Event Processing Agents (EPA) sind zustandsfrei, welche sind zustandsbehaftet?
- Was unterscheidet traditionelle *Datenbankabfragen* (e.g. SQL-Queries) von *Event Patterns* (e.g. EPL-Queries)?
- Was unterscheidet ein *Time Window* von einem *Length Window*? Was sind *Batch Windows*?

Attribution of Slides

- Some slides are used in this lecture with thanks to:
  - Christian Janiesch, KIT
  - Opher Etzion, IBM
  - K. Mani Chandy, Caltech