



IBM Software Group – Event Processing technologies

Event Processing – a semantic approach

*Dr. Opher Etzion, Senior Technical Staff Member
Lead Architect, Event Processing Technologies
Application Integration Middleware,
IBM Software Group*

Where am I coming from ?

I am wearing three different hats



The hat that is paying my salary:

Lead Architect of Event Processing Technologies for IBM Software Group



Once a week I am wearing the Professor hat and teach an event processing course



Additional hat – a catalyst in establishing this community



Syntax, Semantics, Pragmatics

Green cat is the five ones



The rain in Spain stays mainly in the plain

The shallow cat splits the sea



One of my favorite examples will be used for most of the talk..

THE LUGGAGE PERSPECTIVE:

Across the 24 largest airlines more than 5.6 million bags went missing in 2006, this is an average of 15.7 bags per 1,000 travelers. 15% of the bags are never found.

BBC News, April 4, 2007

Orchestrate:
If missed the connection, can be re-routed anyway and the luggage will get there with the passenger

Orchestrate:
Bag has reached to the wrong aircraft



Notify:
Bag has been checked but did not reach the ULD within 20 minutes

Notify: Bag has been checked but did not reach the connecting flight

Outline



Event Semantics



Event Processing in the large - Semantics



Event Processing in the small - semantics



Pragmatic Conclusion



Event Semantics

Event Semantics



Event – types, content,



Event – relationships to the rest of the universe

Event Example 1 – raw event

- ❑ What happened ? --- bag has been checked
- ❑ When it happened ? --- June 18, 05:30
- ❑ Who is involved ? --- Passenger = John Galt; Airline = Air Canada, Flight = 1181, Bag (ID = 496351).
- ❑ What else we can say about the event ?, From = Toronto; To = Edmonton; type = direct; bag's weight = 15 KG
- ❑ Who reported it ? Check-in system
- ❑ [where did it happen ?] Counter 6
- ❑ [why did it happen ?] Part of check-in process
- ❑ [what is the certainty it happened ?]

Event Example 2 – derived event

- ❑ What happened ? --- bag has reached the wrong aircraft
- ❑ When it happened ? --- June 18, 06:00 [or during an interval ?]
- ❑ Who is involved ? --- Bag (496351), Flight = AC7910
- ❑ What else we can say about the event ? Intended Flight = AC1181
- ❑ Who reported it ? Sensor on vehicle going to the airplane
- ❑ [where did it happen ?]
- ❑ [why did it happen ?]
- ❑ [what is the certainty it happened ?]

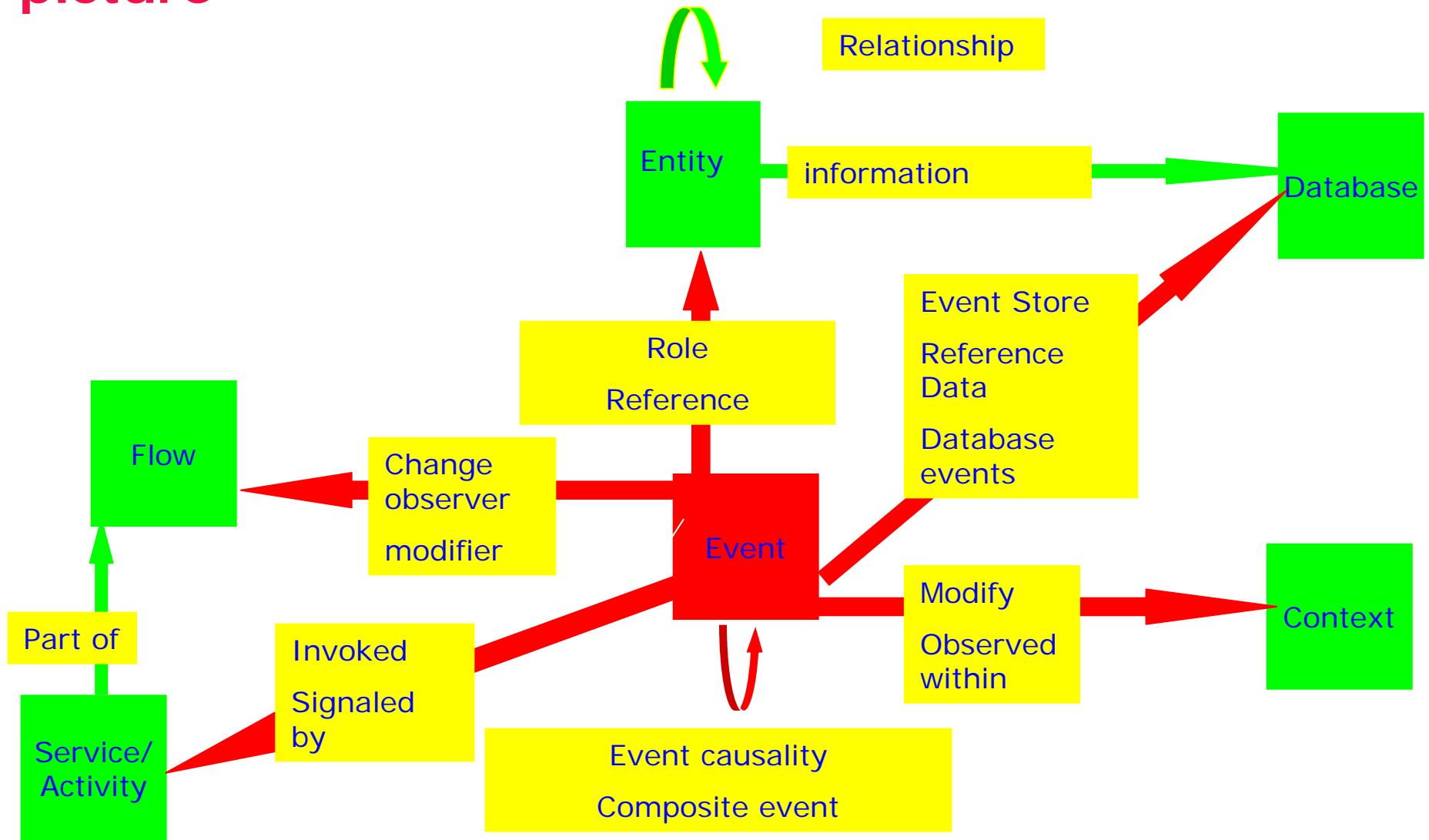
Time Point or Interval

- ❑ Events occur over an interval –
- ❑ In many cases (especially for raw events) event interval can be mapped to time-point and treated like a discrete space
- ❑ However – the semantics of derived event can be:
 - The time-stamp of the last event that occurred
 - The time-interval starting in the earliest participating event, ending in the latest participating events.

Spatial dimension

- ❑ Relative to the time dimension, the space dimension contributed to an event in limited (but growing) set of applications.
- ❑ Space can be: coordinates; Space abstraction that is well-defined: e.g. inside the kitchen...

Event is a semantic term and a part of a larger picture



Event Causality

- ❑ Type I : Explicit
 - The event “passenger notifies about lost luggage” is caused by the event “luggage did not arrive” that can be detected as a derived event.
- ❑ Type II : Induced
 - The event “bag has re-routed” is caused by the event “passenger missed the connection” (but not every time that passenger misses the connections there are bags to re-route)...
- ❑ Type II : Implicit
 - The event “luggage was not found within 72 hours” triggers a “compensation process” which in turn results in the event “transfer money to passenger”. The compensation process by itself is a black box for us (and may be totally manual) - yet, there is a causality relationship between the two events

Event Hierarchy

- Generalization / Specialization/delegation
 - The event “passenger notifies about lost luggage” is a generalization of the event “platinum passenger notifies about lost luggage”. (can be done by context)
 - May be conditional / context-related (“in the night hours, in USA airports”)

Temporal issues in event processing

- ❑ Temporal dimensions:
 - Arrival time
 - Occurrence time
 - (more times: reporting time, transaction time, valid time).
- ❑ Temporal derivation
 - Event occurs over an interval
 - Derive future events
 - Derive past events – retroactive derivation



Event Processing in the large - Semantics

Event Processing Semantics



Event Flow and Event Processing Networks

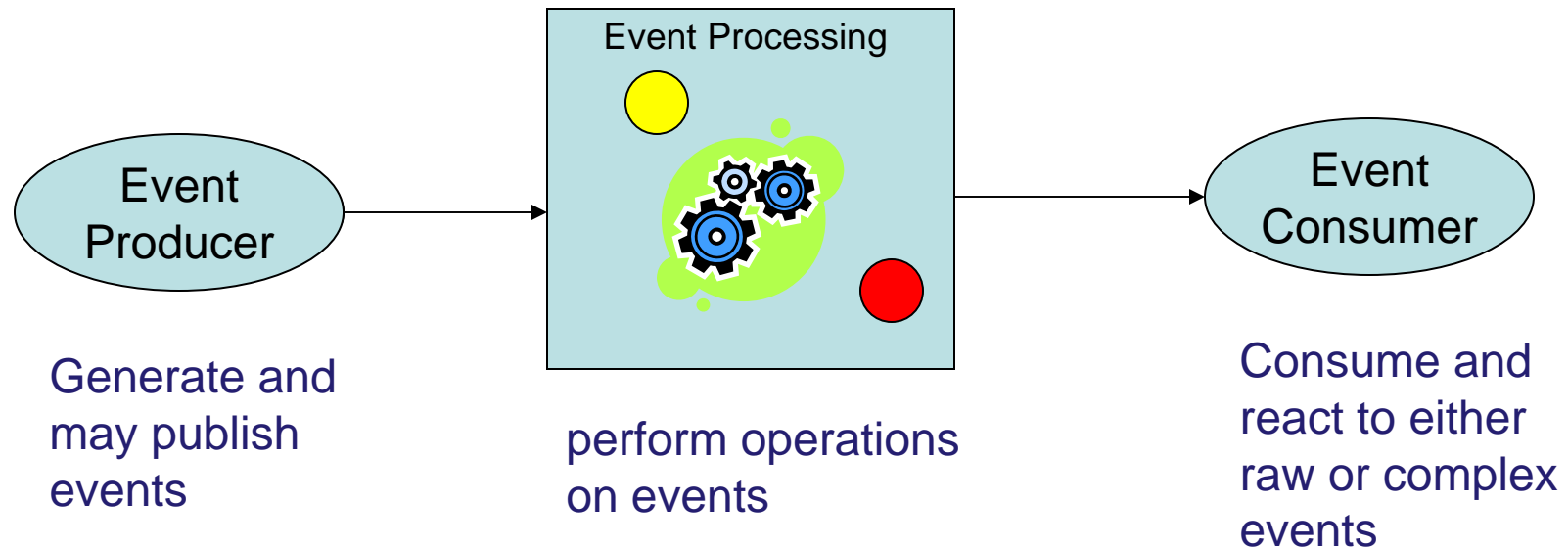


Nodes and Edges in the networks

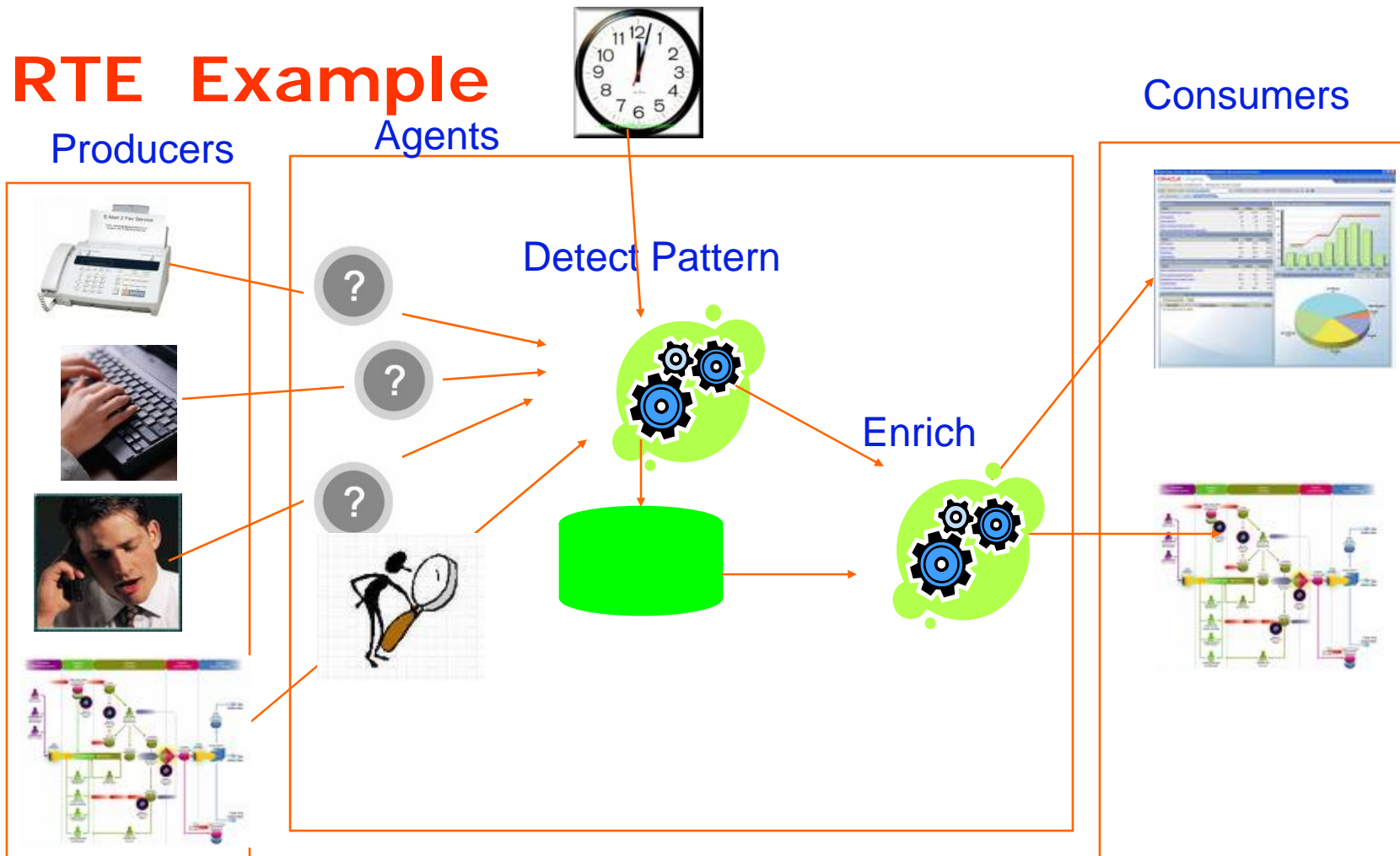


Agents type: simple, mediated and complex event processing

Event Processing – The Big Picture

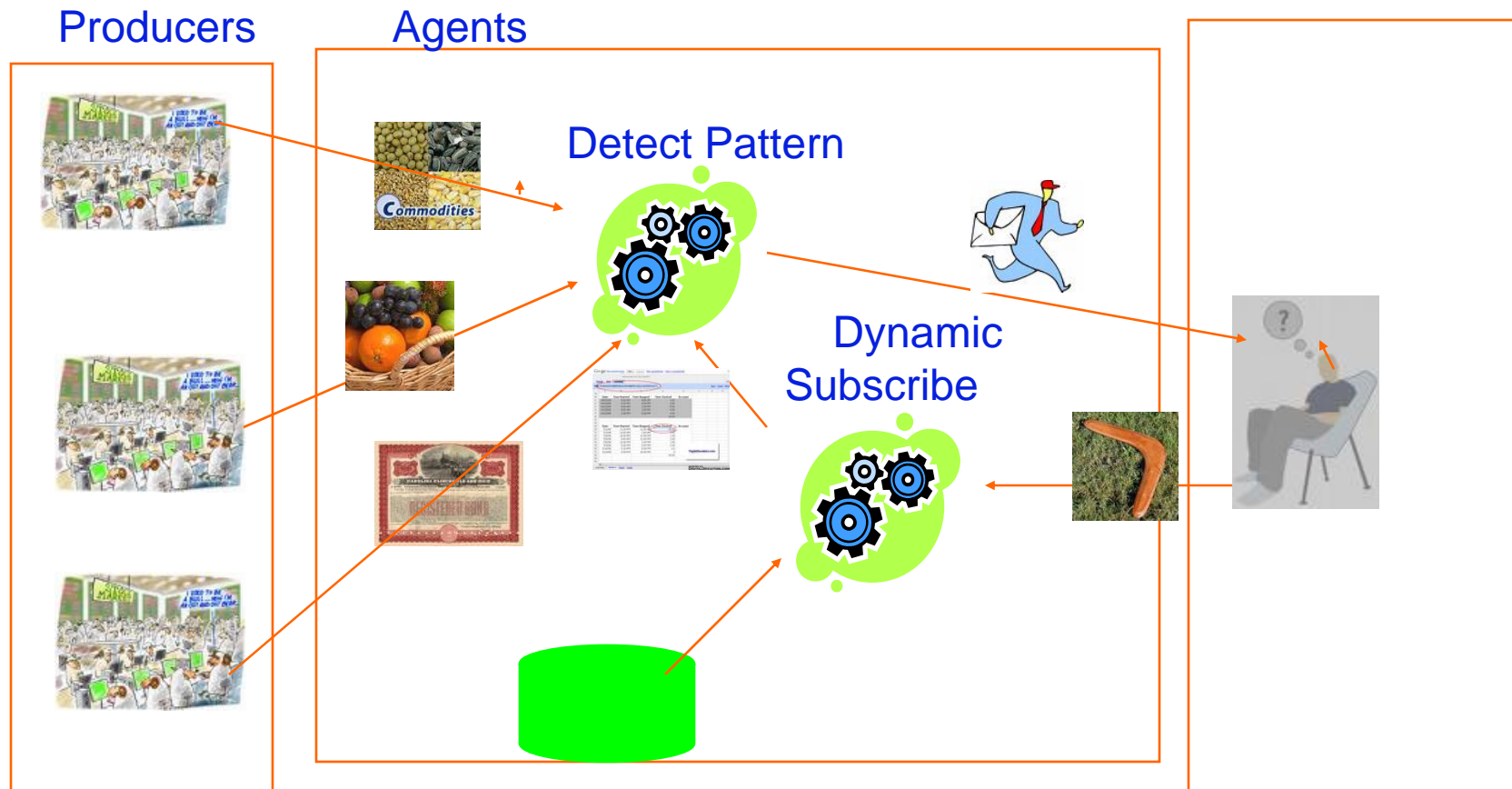


RTE Example



If a customer cancels an order, cancel the order, divert it to another customer, or produce for stock, according to the type of order and current state.

Information Dissemination Example



I am interested to know of all corporate actions in my portfolio

Event Processing Network

- ❑ An implementation-neutral abstraction from the point of view of event flow.

- ❑ Each

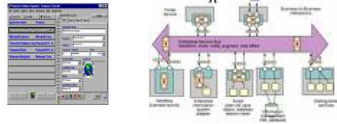


- can be implemented as – Java class, rule, query, hybrid..

Network semantics

- ❑ Nodes:
 - Producer
 - Consumer
 - Agent
 - Channel – push / pull (periodic / on-demand).
- ❑ Edges:
 - (P, CH); (CH, A); (A, CH); (CH, C); (CH, P); (C, CH)
- ❑ Stream = an edge in the graph
- ❑ Cloud = collection of all incoming edges to an agent.

The Players – Event Producers



Applications/services



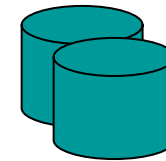
Direct Media



BP state observers



Sensors

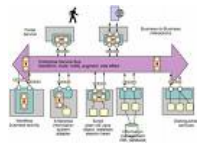


Data Stores

Event Producing Modes:

- **Push** - producer sends an event over a channel by its own initiative
- **Periodic pull** - Channel uses “pulling adapter” to obtain events periodically
- **On-Demand pull** - An agent initiates pull in an ad-hoc fashion.

The Players – Event Consumers



Applications/services



Actuators



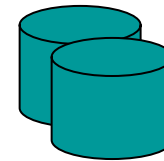
Individuals and Groups



Workflows



Dashboards

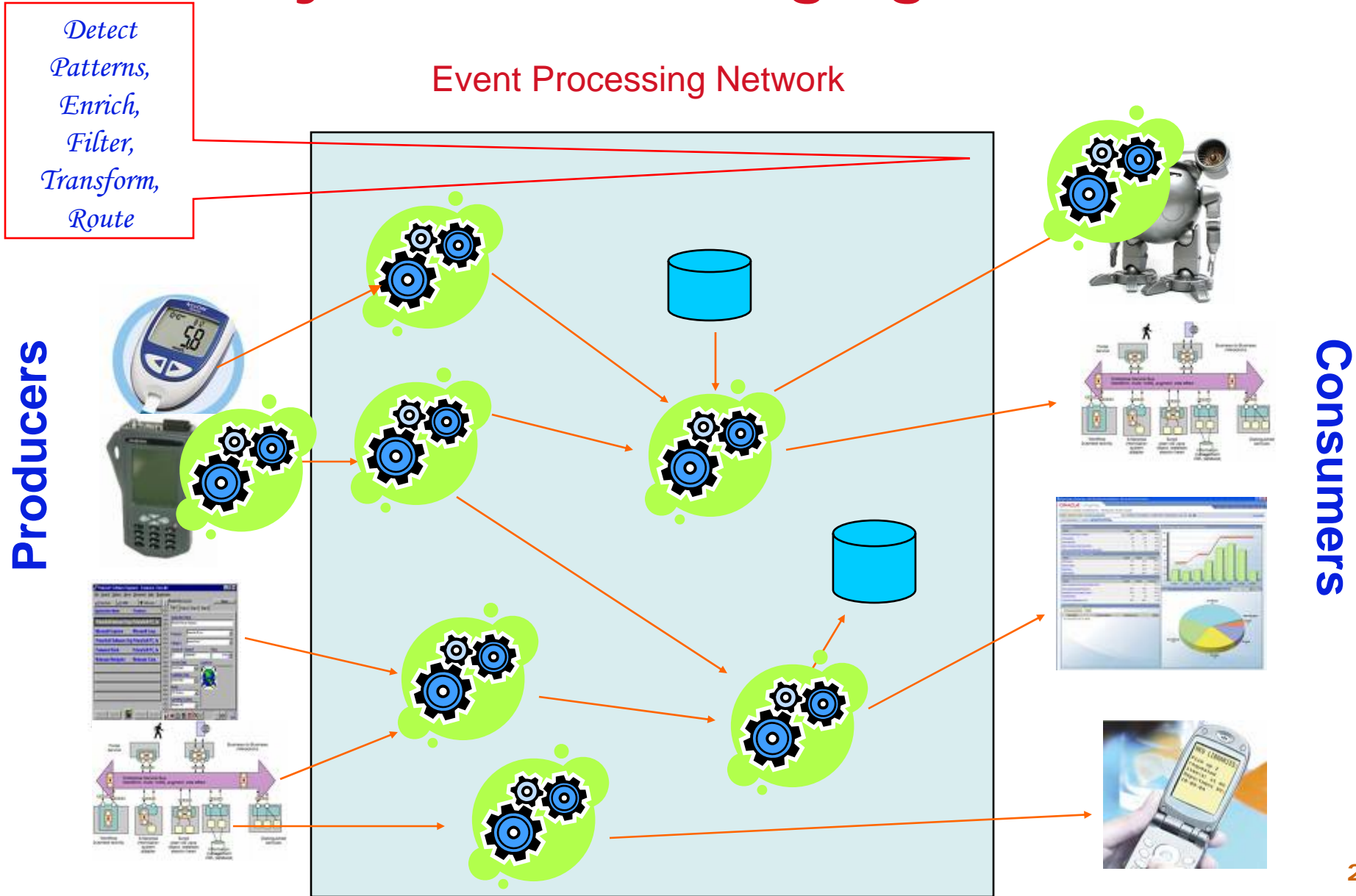


Data Stores

Event Consumption Modes:

- **Notification** – The consumer is notified
- **Orchestration** – The consumer acts

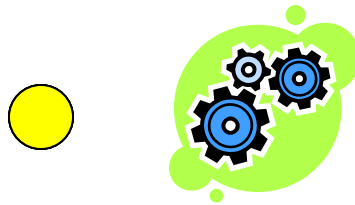
The Players - Processing Agents



Event Processing Overview

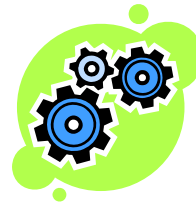
Event Processing Agents - Filtering

Filtering is a stateless agent that filters events. It is the most common agent, and a filtering agent has been required by many applications.

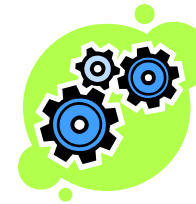


Event Processing Agents - Routing

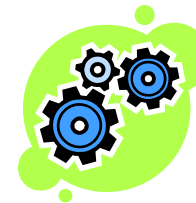
Itinerary-based routing



Subscription-based routing

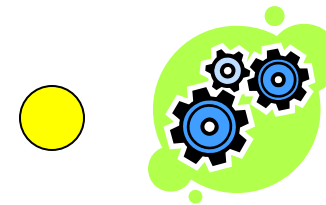


Intelligent routing

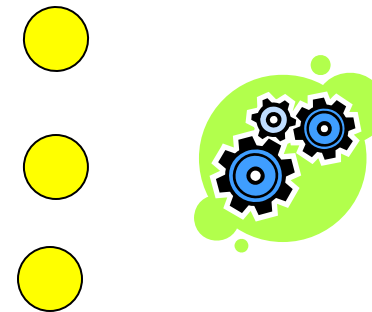


Event Processing Agents - Transformation

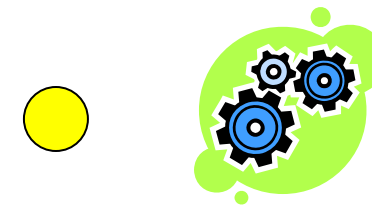
Translator: Both transformation and translation



Aggregator: statistical aggregator or concatenating events, can be stand alone agent or a scalar derivation. May be in network edge.

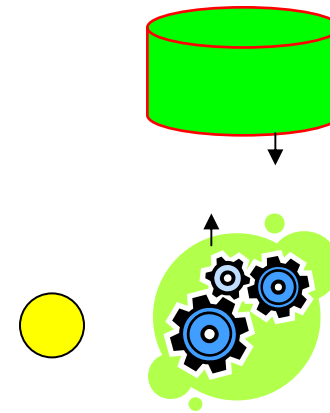


Splitter: Splits events to multiple events



Event Processing Agents - Enrichment

Enricher: Enriches the content of events from reference data in databases, spreadsheets, Email messages, text files etc..



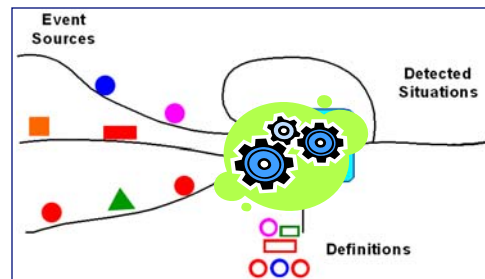
Event Processing Agents - Pattern Detection

Time series pattern detector:
"Set at a time" pattern detector.

Basic pattern detector: fixed set
of most common patterns.

Advanced pattern detector:
various packaging of patterns,
pattern policies.

Output is "complex event"
representing the situation
detected by the specified
pattern



Context as a first class abstraction

- ❑ First class citizen –
 - during working hours
 - For important customers
- ❑ Partition of the stream – each partition may have different rules
 - Partition by time
 - Partition by space
 - Partition by property (one or more)

Transactional semantics of event processing

- ❑ Event processing typically considered as asynchronous – thus, non transactional
- ❑ But --→ in some applications has to have transactional properties
 - A part of a larger transaction
 - Transactional properties of EPN
- ❑ Easy semantics:
 - Each event that enters the EPN starts a sub-transaction
 - If an agent derives event (even if most participants are not part of this transaction) it is considered as a part of this transaction
 - The transaction is atomic (rollback is provided)
 - Questions:
 - Can we semantically say that event is withdrawn ?
 - Is it the simple solution – always the correct one ?



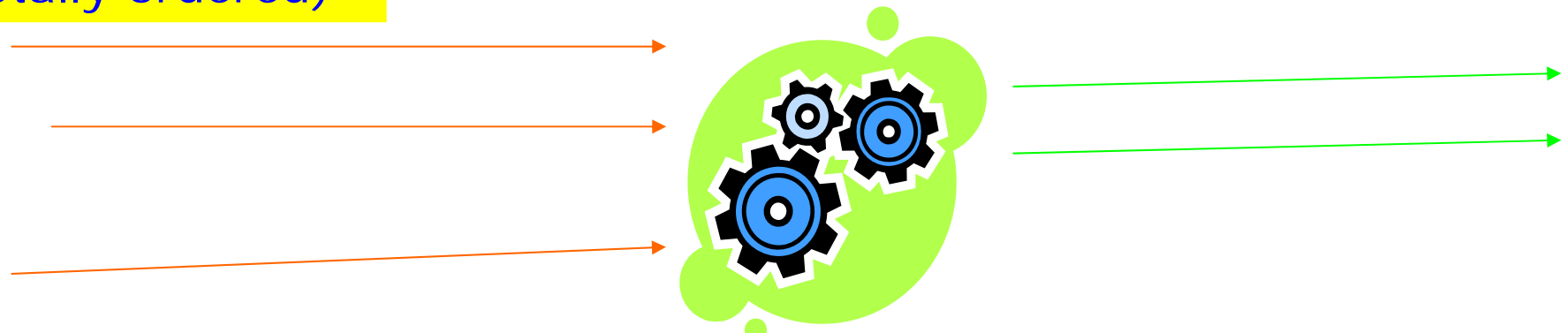
Event Processing in the small - semantics

Agent Semantics

An agent does three things:

- Selects which of the events in the cloud are relevant for it
- Perform the function
- Decide how to route the results

Stream
(sometimes
totally ordered)



Input Cloud
= collection
of stream in
context

Pattern Detection Semantics – pattern type

- ❑ Operation on Event Type + Predicate on content (together)
- ❑ Some examples:
 - Conjunction
 - Disjunction
 - Sequence (where supported)
 - Negation (can be detected when context ends)
 - Counting
 -

Scenario number 1: The heavy trading example

□ Given:

- A stream of events of a single topic, about the activity in the stock market for a certain stock.
- An event is produced every 10 minutes when there is trade in the stock.
- Each event consists of: quote (current stock-quote), volume (an accumulated volume of traded events within these 10 minutes).
- A selection specification: "trigger an automatic trade program if the volume exceeds 300,000 3 times within an hour; pass as an argument the last quote and the sum of the 3 volume values".

Does it provide a complete semantics?

But now look at the actual stream

Event-Id	Time-Stamp	Quote	Volume
E1	9:00	33.23	
E2	9:10	33.04	320,000
E3	9:20	33.11	280,000
E4	9:30	33.01	400,000
E5	9:40	32.90	315,000
E6	9:50	33.04	320,000
E7	10:00	33.20	303,000
E8	10:10	33.33	219,000
E9	10:20	33.11	301,000
E10	10:40	33.00	210,000
E11	10:50	32.78	400,000
E12	11:00	32.70	176,000

- How many times the trade programming is triggered ;
- Which arguments are used in each triggering?

Semantic decisions (some examples)

- ❑ **Decision 1: When is the selection applicable?** *Every day*
- distinctly from trade-start - start new context every half hour, close the context after an hour
- ❑ **Decision 2: Single or multiple time windows?** *Multiple*
(in this case 2 can co-exist).
- ❑ **Decision 3: When a detected event should be reported?** *Immediately.*
- ❑ **Decision 4: Within an interval – one or many result streams?** *Only Once.*
- ❑ **Decision 5. What quantifier is used in case of multiple instances?** *First*
- ❑ **Decision 6: Can a consumed event be re-consumed?**
no
- ❑ **Decision 7: Are new events override previous events?**
no

Scenario number 2: The matching example in electronic bidding system – specification including decisions

□ Given:

- A stream of events that belongs to 2 event-topic: buy-request and sell-request
- The buy-request consists of : item, customer-id, quantity, upper-limit
- The sell-request consists of: item, customer-id, quantity, lower-limit
- If a customer sends another sell or buy request before the previous one was settled, it overrides the previous one.
- The result is a “settlement” event that occurs which matches buy-request with sell-request such as $\text{upper-limit} \geq \text{lower-limit}$. The result event is: item, buying-customer, selling-customer, minimum of quantities, lower-bound, upper-bound.
- The buy-request and sell-request do not participate in any matching again, even if not all desired quantity was settled.
- The matching should be “fair” in the sense of “first bid, first matched”, where the time of a bid is considered as the time of the last override.

Semantic decisions (some examples)

- ❑ **Decision 1: When is the selection applicable?** For each item - distinctly when a bid for this item is open
- ❑ **Decision 2: Single or multiple time windows?** One time window in parallel
- ❑ **Decision 3: When a detected event should be reported?** At the end of the bid period
- ❑ **Decision 4: Within an interval – one or many result streams?** Multiple.
- ❑ **Decision 5. What quantifier is used in case of multiple instances?** First
- ❑ **Decision 6: Can a consumed event be re-consumed?** no
- ❑ **Decision 7: Are new events override previous events?** Yes, for a buy or sell event in the same context - a new event overlaps the old one.



Pragmatics Conclusion

Implementation

- ❑ This semantics can be defined using various types of implementations / APIs.
- ❑ The question of ordered vs. unordered is one characteristic, but there are many of them:
 - Requirements for latency, throughput
 - Need to support – transaction, recoverability
 - Size of state
 -
- ❑ One size fits all will not work – but the partition is not necessarily stream processing / CEP...
- ❑ Hybrid implementations are emerging...

API and standards

- ❑ It seems that the “religious” war is on API type:
 - SQL extensions
 - Scripts
 - Rules:
 - Inference Rules
 - Derivation Rules
 - Visual Languages
- ❑ Using unified semantics we can get to a *standard meta-language*