



A subjective tour of event processing

Event Processing Symposium

Opher Etzion

March 14, 2006

Where am I coming from ?

I am wearing three different hats today



The hat that is paying my salary:

Chief 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



Outline



The position Questions – some vision



The position Questions – Back to the Present



Terminology, Model, Taxonomy



Conclusion



Position Questions – some vision

The Vision: Event Processing in 2013



- ❑ Event Processing repeats (in 30-something years offset) the success of “Data Management”
- ❑ Part of the “main stream computing”
 - Wide coverage in term of applications that are doing some type of event processing
 - There are broadly accepted standards
 - Event Processing extensions to programming languages
 - Large amount of developers are familiar with the concepts
 - It is widely taught in universities with popular textbooks
 - Well-established Research community
 - Other disciplines focused on extracting events and event patterns (image processing, information retrieval, search engines, data mining).

What do we have to do in order to get there Lessons from Relational Databases

□ Why it succeeded ? ("Business Rules Applied", Barbara Von Halle)

- It has a sound theory behind it.
- Software vendors understood the theory (well... to some extent), and delivered commercial products
- There were some good books that eloquently explained the theory, the benefits, and the practicality, to the IT community
- Practitioners developed methodologies for using it.
- (my addition): SQL as an inter-galactic standard enabled interoperability.
- (my addition): A lot of R&D Investment to advance this area in engineering issues (transactions, query optimization, concurrency control)...




Challenges for the Event Processing community

- ❑ Pass the declining slope in the hype-cycle without crashing – being captured as a mature discipline.
- ❑ Good understanding of performance trade-offs, optimizations of various types, scalability etc..
- ❑ Adopt an agreed upon terminology. **Currently we are confusing the market**
- ❑ Adopt a solid model – “platform and implementation independent” as a basis (similar to semantic data model, but with extended role)
 - Vendors should be able to map from this model to their own implementations in an automatic way (or in time develop “native” implementation – but there is no commercial native implementation to semantic data models).
 - Standard in the model level, but also in event structures, interfaces etc..
- ❑ Adopt standard benchmarks
- ❑ software engineering practices will be crucial in the success.
- ❑ And good textbooks to educate the practitioners.



Position Statement – back to the present

What drives this area ?

- ❑ From the bird eye's view:
 - Nothing is really new
 - We process events for many years
 - (e.g. exception handling in OS).
- 
- ❑ However, recently:
 - Significant amount of events – types, sources, instances
 - Variety of application need to process events
 - Some traditionally stand-alone applications need to be integrated with regular information systems (simulation, real-time)
 - Functionality requires sophistication – e.g. temporal capabilities, spatio-temporal capabilities.
 - ❑ These all contributed to COTS tools
 - It is not cost-effective to develop this functionality for a single application
 - It is recognized as middleware level capability, thus customer preference for COTS.
 - Drive for standards is next step

Types of applications

1. Agile Process Integration

2. Autonomic behavior in business cases

3. Awareness to Business Situations

3. Change Management/ Impact analysis

4. Delivery of information services

5. Management of services and processes

6. Proactive systems



This is an attempt to classify different scenarios

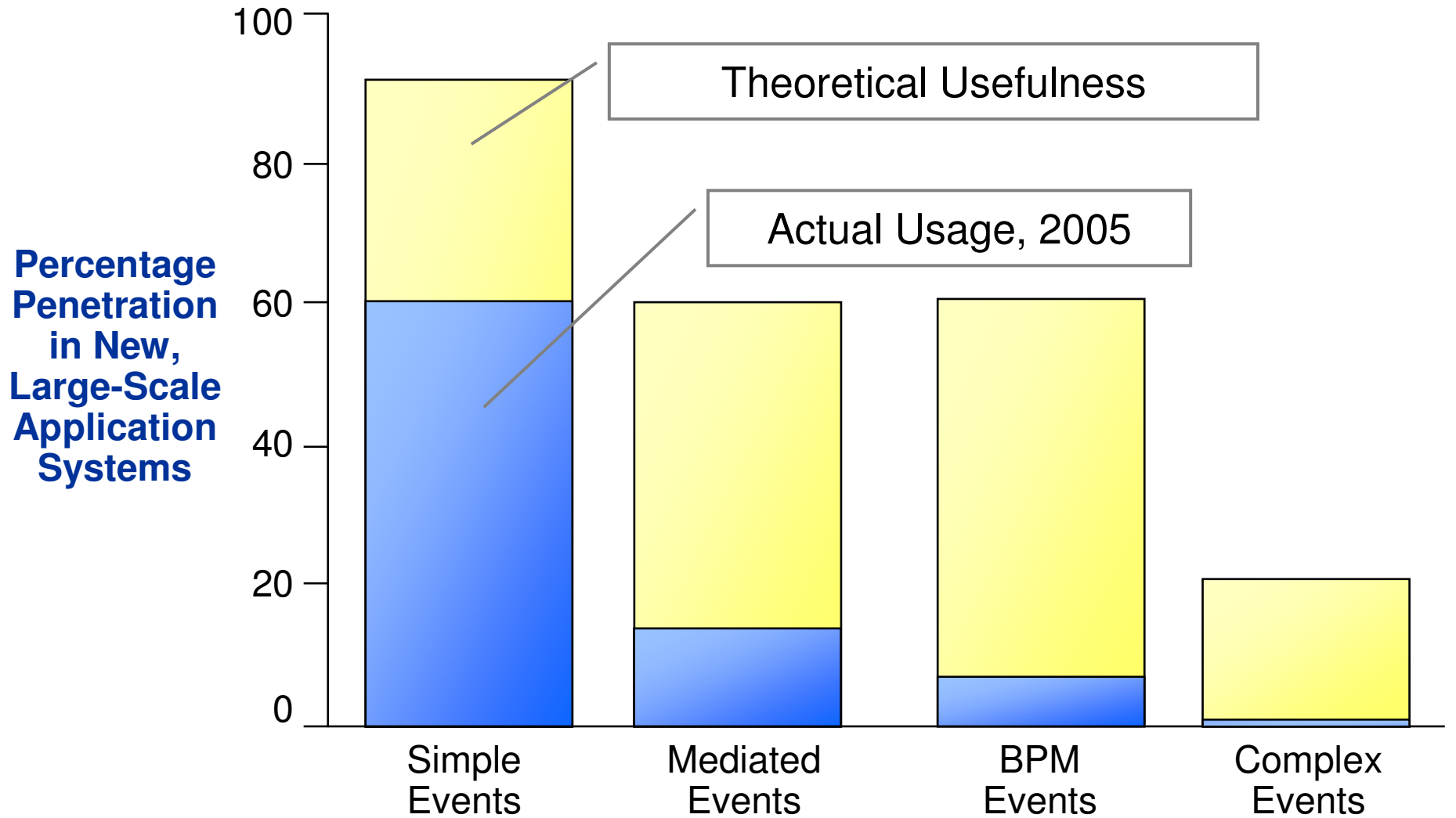
The Language Issue

- ❑ Some communities have been engaged in language wars over the years
- ❑ In the event processing community there are several approaches:
 - SQL oriented (either database approach or data stream approach in which the queries are constant in a time-frame)
 - Rule-based oriented (can be some gentle form of temporal logic)
 - Script oriented languages
 - Code generation vs. interpreted one.
- ❑ Each language may have benefits and shortcomings in various cases (type of user, the functionality needed of a certain application)
- ❑ The challenge (and key to standardization) – turn to a more abstract model of event processing that can be mapped to these languages (equivalent to the: platform independent model in the OMG classification).



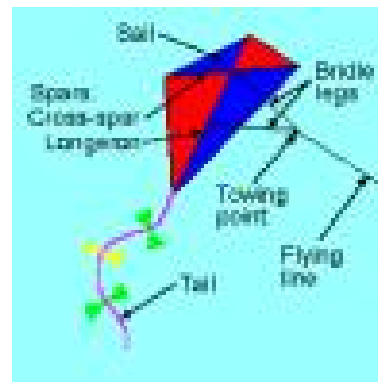
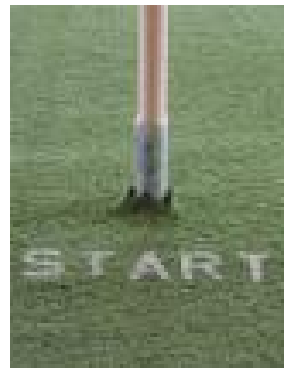
*Event Processing - Taxonomy,
Terminology, Model*

Adoption of EDA Application Types (Roy Schulte, Gartner, December 2005)

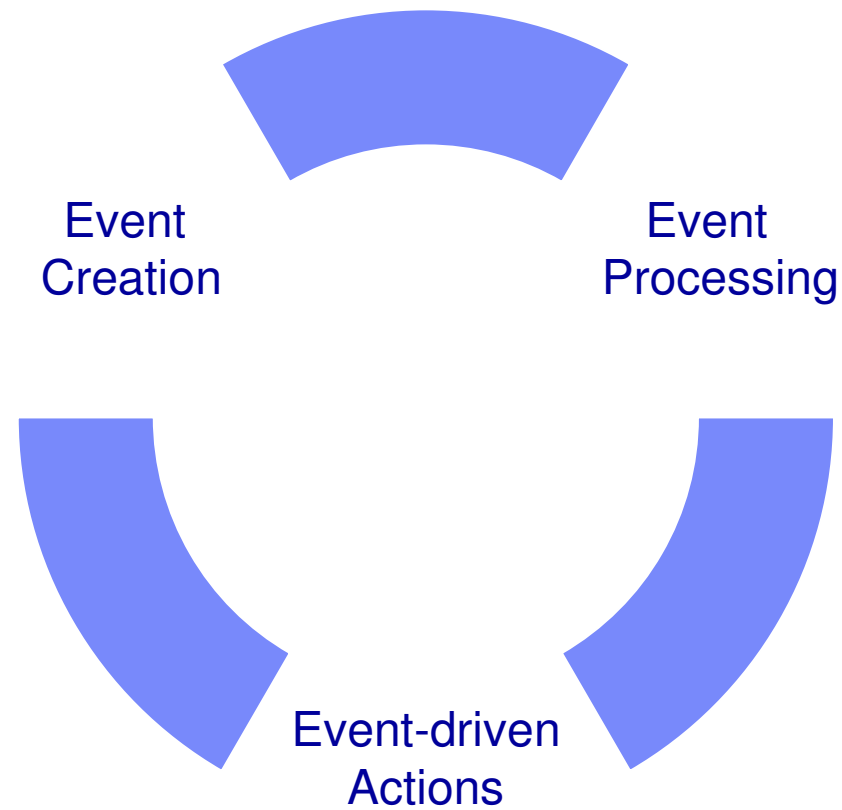


The use of terminology

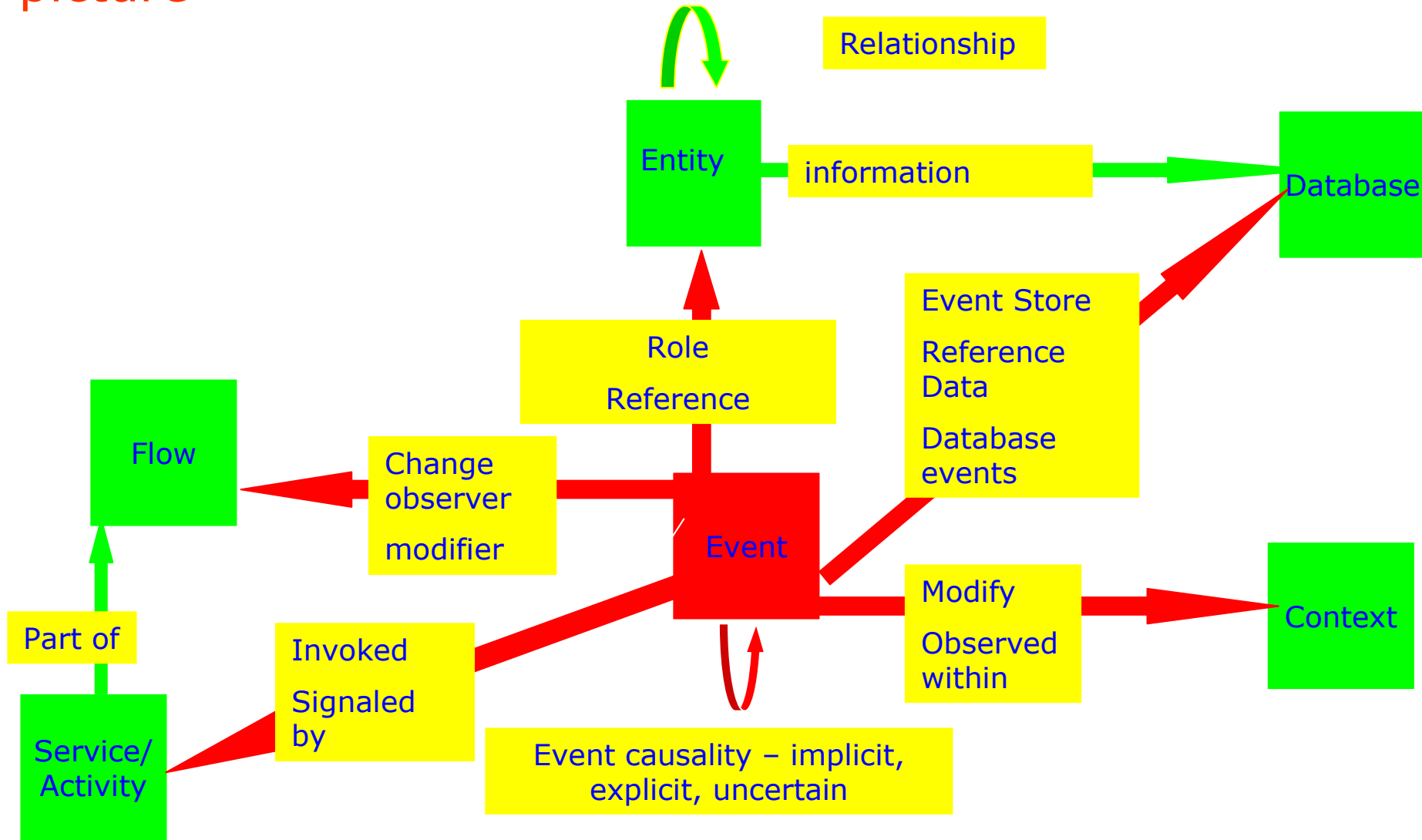
- ❑ There is no agreed upon terminology
- ❑ Thus, I'll make terminology definitions as side-notes in each slide...
- ❑ This may serve as a discussion starting point.



Event Driven Applications



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

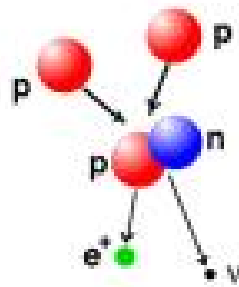


Event Creation

- ❑ Produced or sensed
- ❑ Push, periodic pull, on-demand pull.
- ❑ challenge: Increase the “event scope”
 - Events extracted from video streams
 - Events from RSS feeds
 - Events extracted from textual information

Event-driven Action

- ❑ Notify/Store – for consumer
- ❑ Trigger, Orchestrate – for actor (BPM event processing)



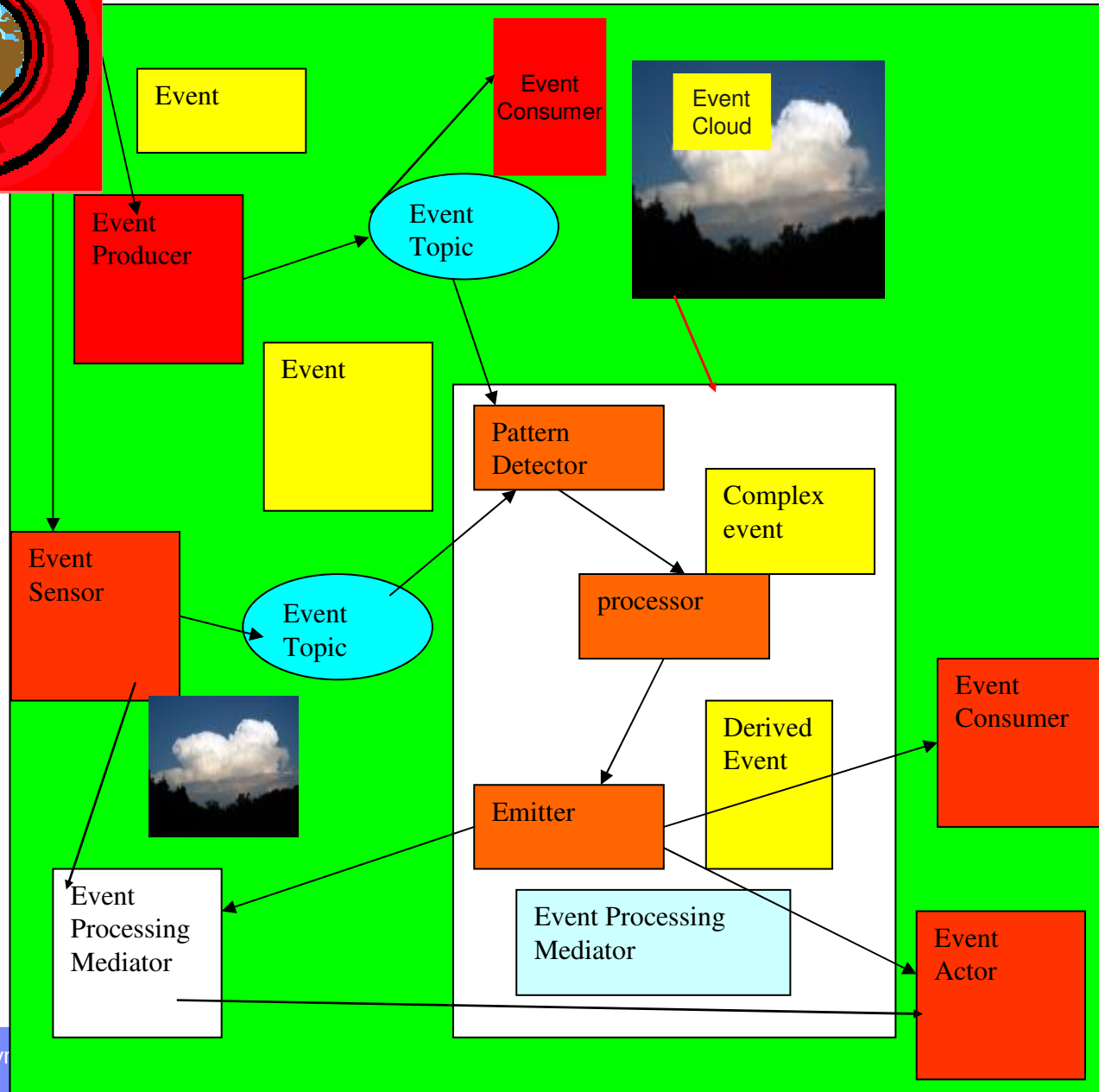
Architecture

Glossary

Context: A set of criteria to partition the space of events according to temporal (e.g. time window), spatial (e.g. space window) and partitioning entity (e.g. platinum customers)

Event Stream: A collection of events that arrive to a single consumer over a single event topic within a single context

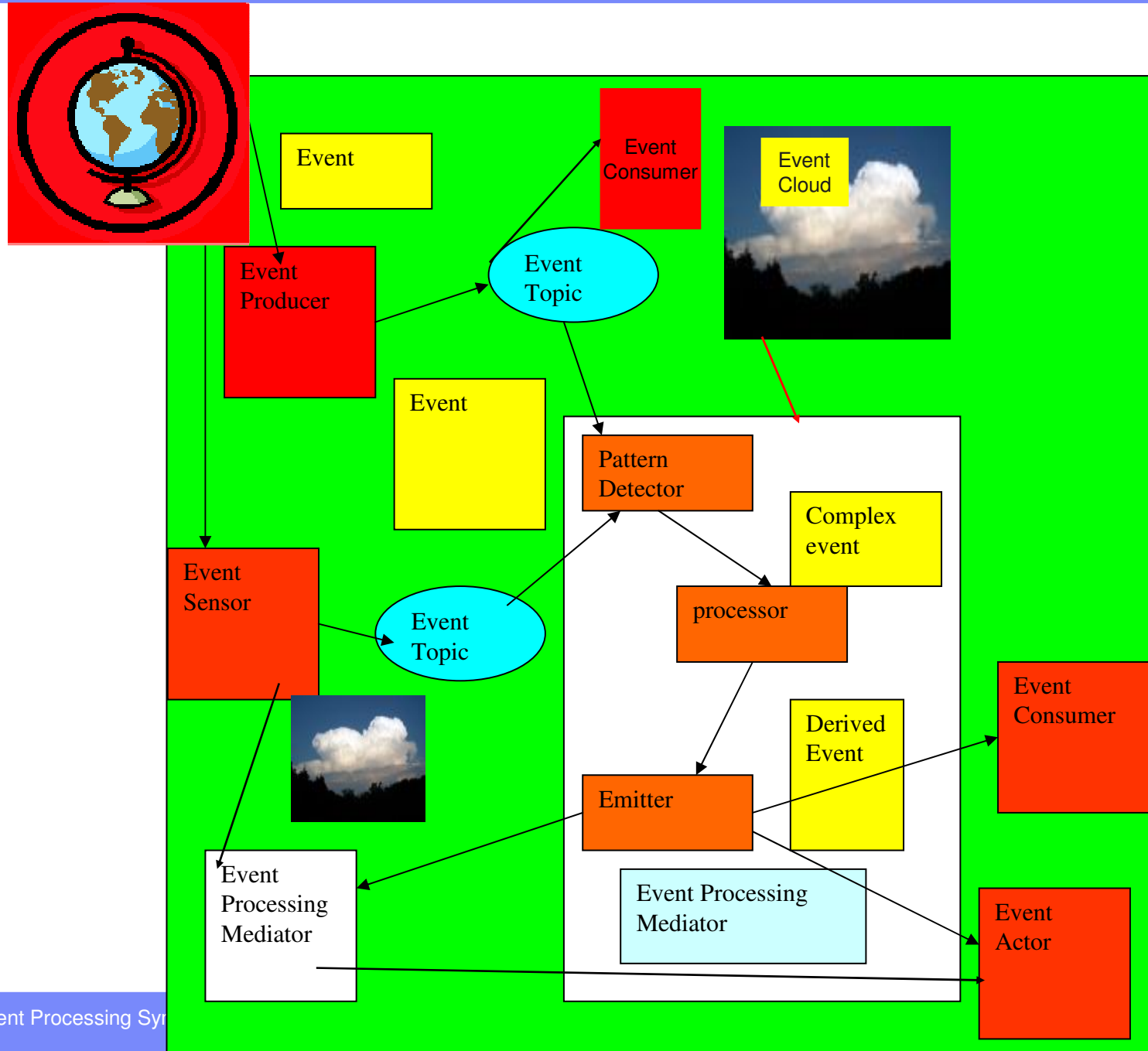
Event cloud: A collection of all event streams that a single consumer receives



Architecture

Glossary

Event Processing Mediator:
 A software artifact that gets an “event cloud” as an input, and produces a collection of events as output.

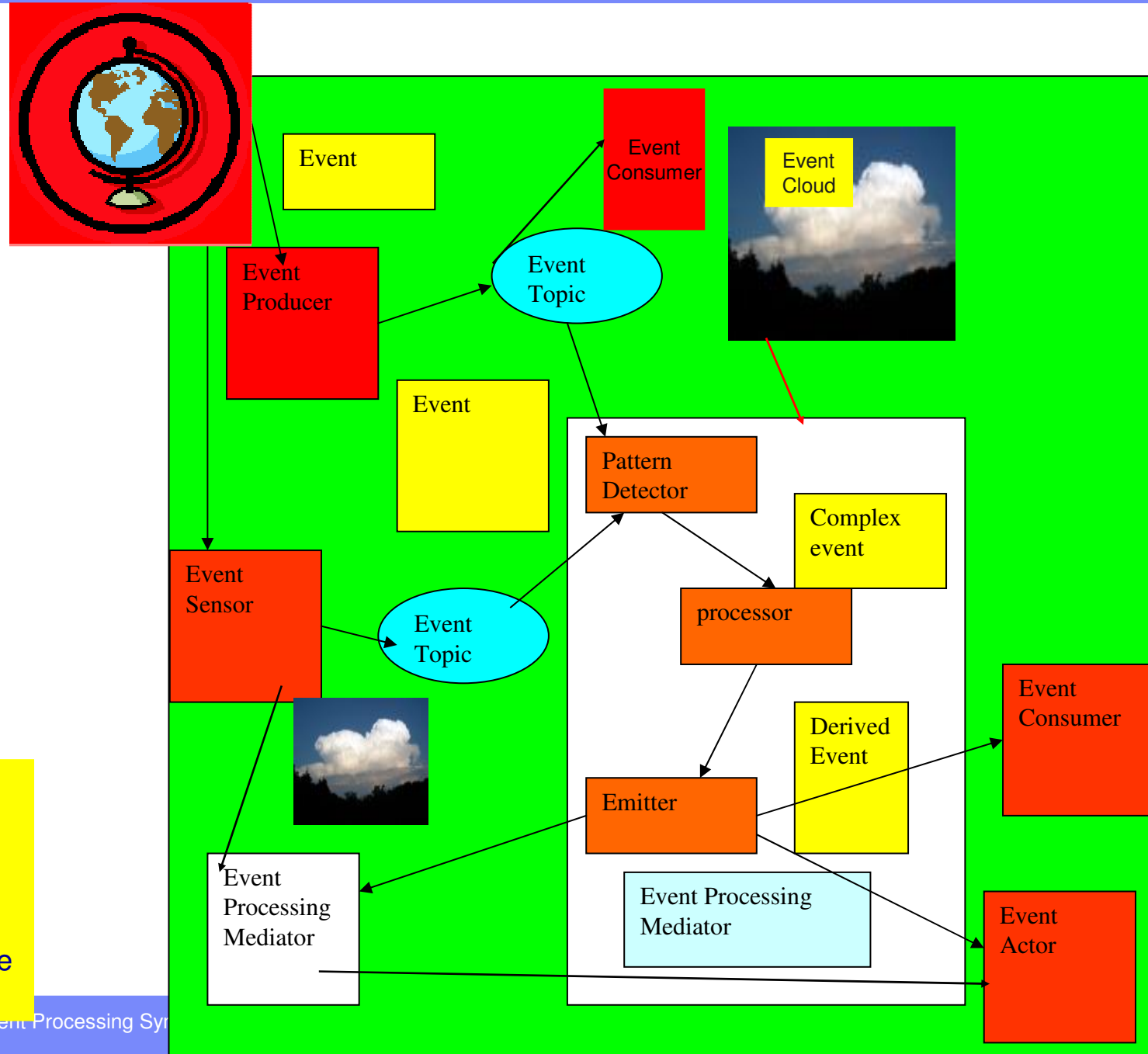


Architecture

Glossary

Pattern Detector: A software artifact that obtains an event cloud as an input, and creates a “complex event”
 –Example: the stock quote is monotonically decreasing within 2 hours. Return the collection of stock quotes.

Complex event: An event that contains reference to all events that participate in the pattern

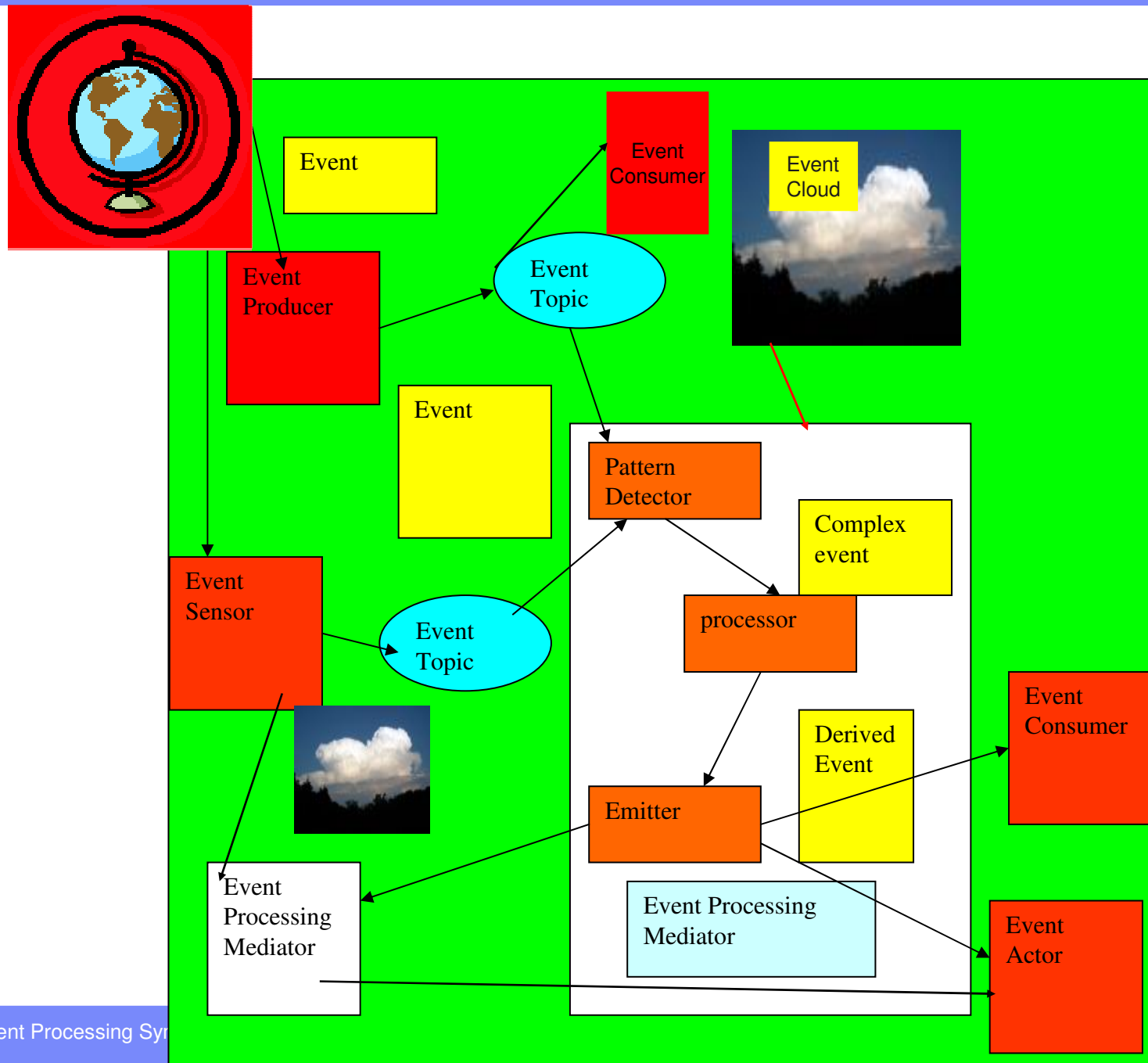


Architecture

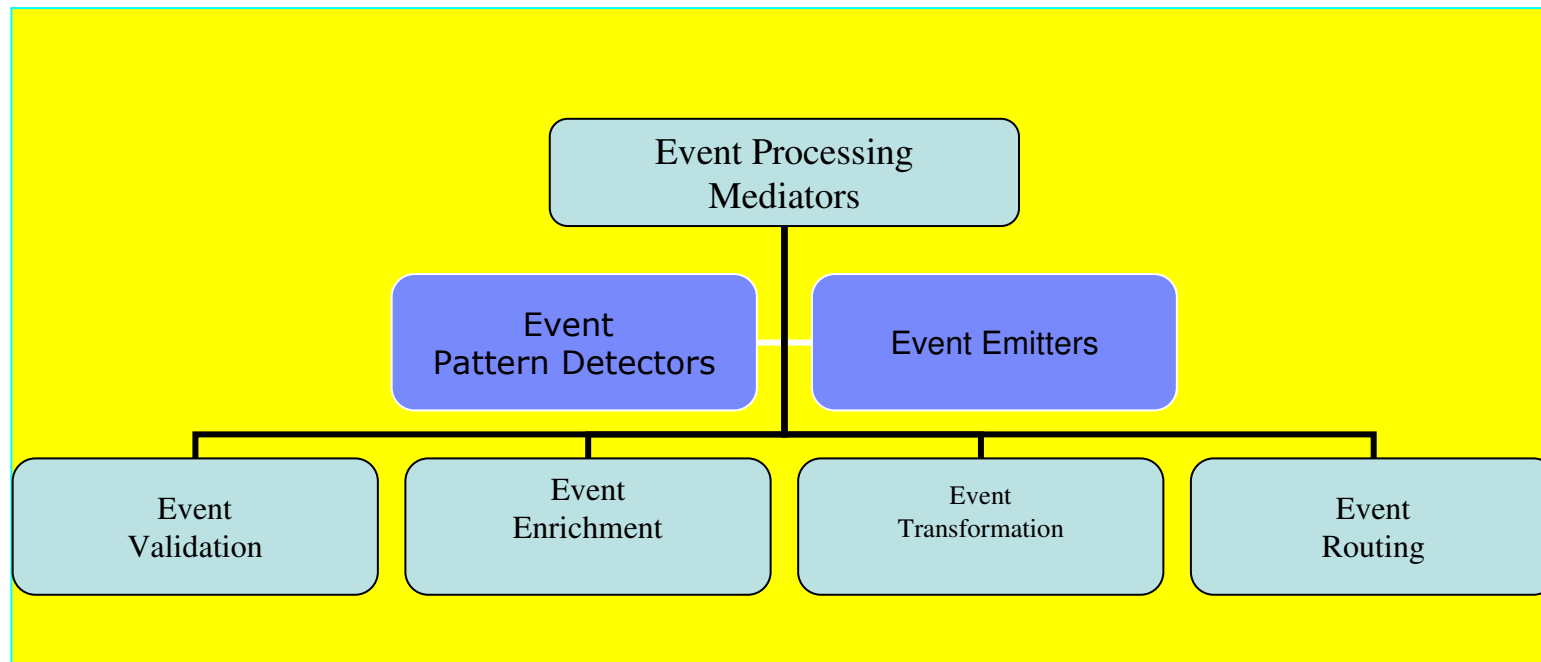
Glossary

Event Processor: A software artifact that obtains a composite event and creates collection of derived events using some function on the input events, such as: Enrichment, Transformation, Aggregation, Split.

Derived Event: An event that is calculated as a function of other events.



Event Processing Mediators



SEP – Simple Event Processing

Detector
Stateless
Basic Statefull
Advanced Statefull
Retrospective

Processor
Route
Enrichment
Validation
Basic Transformation
Advanced Transformation

Emitter
Pub/sub
Itinerary
Intelligent Routing

Activate
Notify
Trigger
Orchestrate

MEP – Mediated Event Processing

Detector
Stateless
Basic Statefull
Advanced Statefull
Retrospective

Processor
Route
Enrichment
Validation
Basic Transformation
Advanced Transformation

Emitter
Pub/sub
Itinerary
Intelligent Routing

Activate
Notify
Trigger
Orchestrate

BEP – BPM Event Processing

Detector
Stateless
Basic Statefull
Advanced Statefull
Retrospective

Processor
Route
Enrichment
Validation
Basic Transformation
Advanced Transformation

Emitter
Pub/sub
Itinerary
Intelligent Routing

Activate
Notify
Trigger
Orchestrate

CEP (Complex Event Processing)

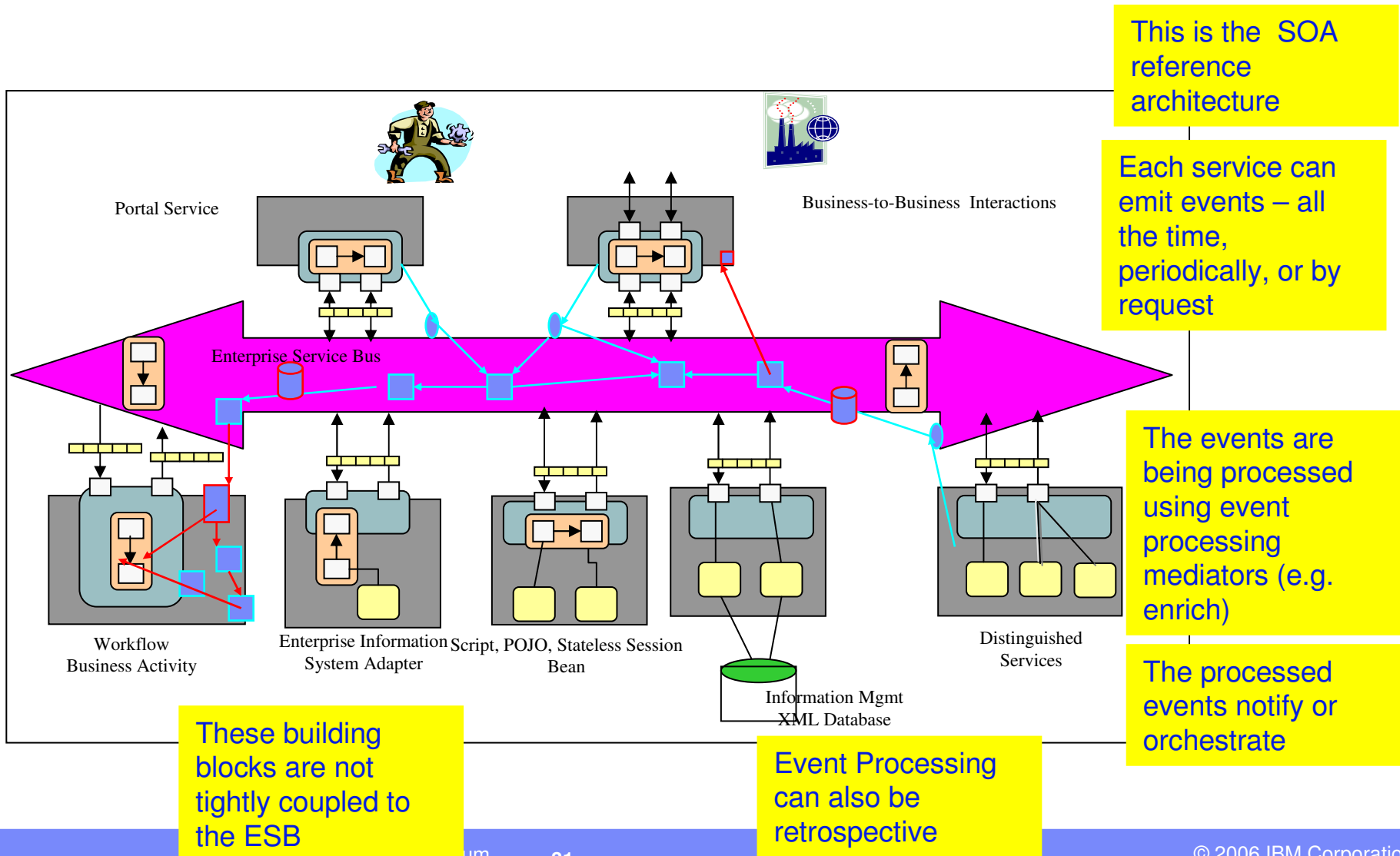
Detector
Stateless
Basic Statefull
Advanced Statefull
Retrospective

Processor
Route
Enrichment
Validation
Basic Transformation
Advanced Transformation

Emitter
Pub/sub
Itinerary
Intelligent Routing

Activate
Notify
Trigger
Orchestrate

EDA as part of SOA





Conclusion

Wearing the catalyst hat:



- ❑ I am personally a true believer in the potential of the event processing area (otherwise I would have done something else !).
- ❑ Getting event processing to the “main stream” of computing will be beneficial for all of us
- ❑ In addition to the fact that all of us (or at least part) will generate {great, reliable, efficient, effective etc}.. Products
 - a community effort is needed to make it happen.