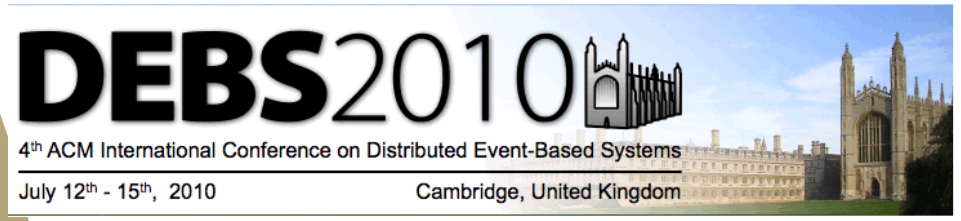


Tutorial v1.1

epts

event processing technical society

Content by: members of the
EPTS Reference Architecture Working Group

Event Processing Architectures

leading to an **EPTS Reference Architecture**

Courtesy of: Adrian Paschke (Freie Universitaet Berlin)
Paul Vincent (TIBCO Software)
Catherine Moxey (IBM)
Alex Alves (Oracle)
Themis Palpanas (University of Trento)

Tutorial

- Event Processing is an increasingly important area in the field of IT
- Event Processing Architectures have evolved to handle the needs of low-latency / high-throughput event processing
- Event Processing Architecture diagrams are used to describe the functions and component layouts of event processing systems
- Various providers and suppliers use their own architectural descriptions, and EPTS has collated and refined these into a candidate “reference” architecture

Agenda

- Introduction to architectures, architecture methodologies, and event processing
- Member architectures and salient features
- Skeleton reference architecture from EPTS Reference Architecture Working Group
- Summary and future work of the EPTS Reference Architecture Working Group

epts
event processing technical society

DEBS2010
4th ACM International Conference on Distributed Event-Based Systems

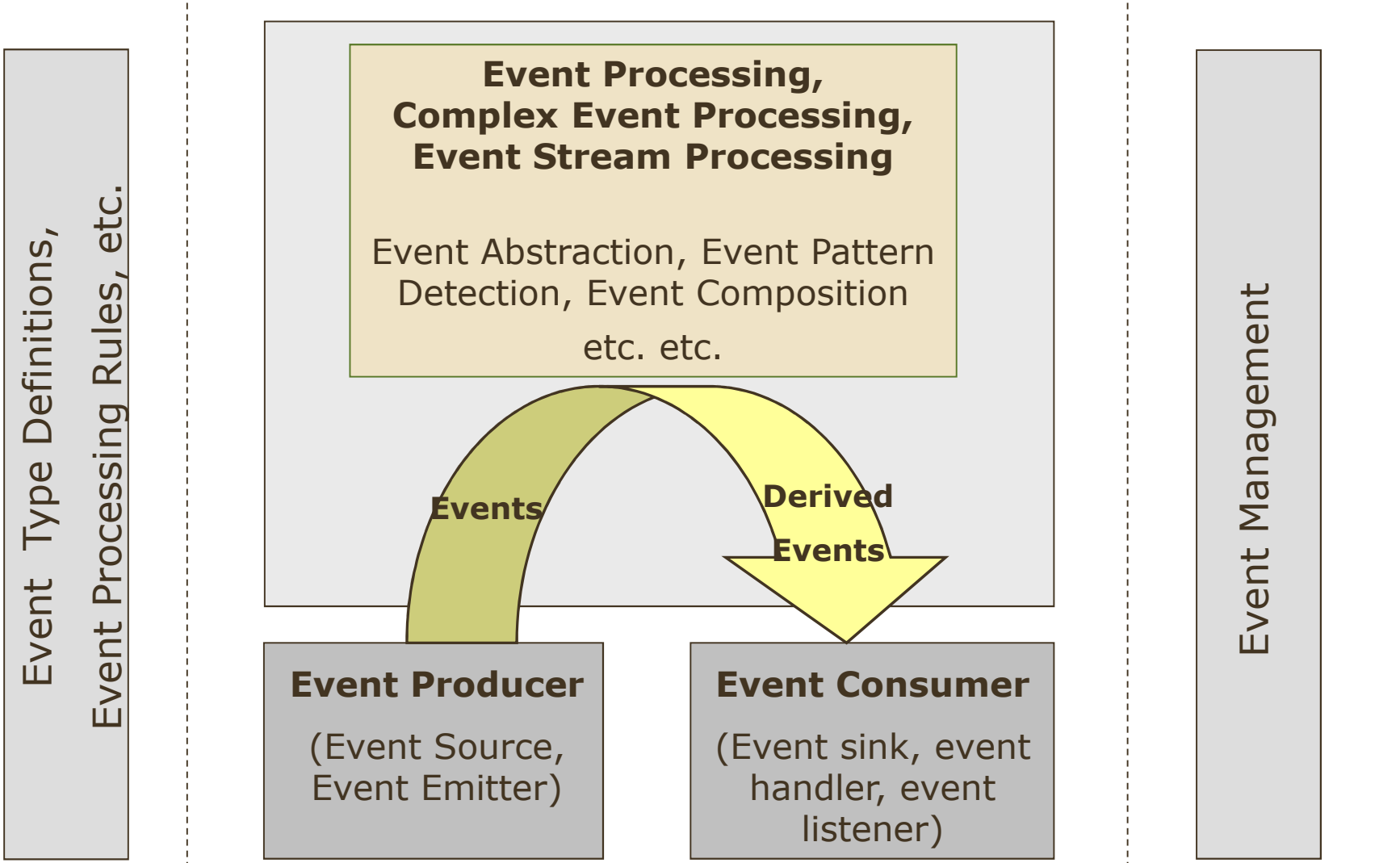
Agenda

- **Introduction to architectures, architecture methodologies, and event processing**
- **Member architectures and salient features**
- **Skeleton reference architecture from EPTS Reference Architecture Working Group**
- **Summary and future work of the EPTS Reference Architecture Working Group**

Introduction to Event Processing

- **Event-centric view of IT**
- **Events are**
  - Sent and Received
  - Aggregated, Transformed into Data, Deleted
  - Processed in queries, rules etc
  - Cause actions like processes, service invocations, etc

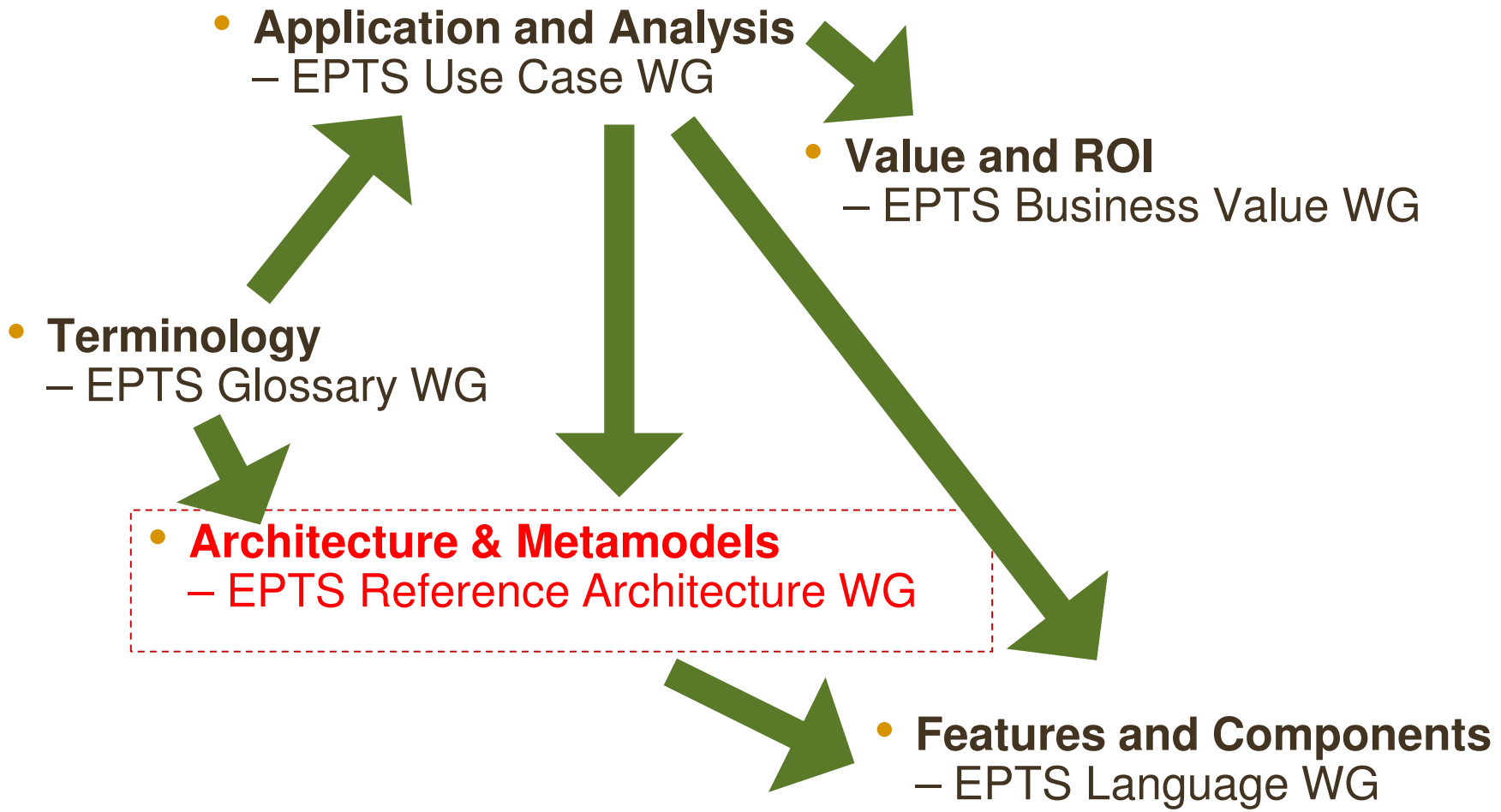
Introduction to Event Processing



About the EPTS Reference Architecture Working Group

- **Started March, 2009**
  - 18 members, co-chairs are Adrian Paschke (RuleML) and Paul Vincent (TIBCO)
  - July 09 added responsibilities from Metamodel Working Group
- **Scope**
  - Define **architecture patterns** that are compatible with EPTS members' Event Processing solutions and products.
  - Define **terminology and components** regarding Event Processing in accordance with EPTS
  - Identify and utilize **best practices and methods** for Technical Architecture descriptions and interchange
  - **Liaise with relevant standards bodies** for EP metamodels and reference architectures
- **Current work is focused on**
  - Discovery of **existing Event Processing Architectures**
    - collected RAs from e.g. IBM, Oracle, Tibco, Streambase, Aleri, Microsoft…
  - Definition of **Terminology and Methodology** for comparing and describing Event Processing Reference Architectures

Relationship with the other EPTS Groups / Areas



Reference Architecture and Reference Model

- **Reference Architecture**

  A reference architecture **models the abstract architectural elements** in the domain independent of the technologies, protocols, and products that are used to implement the domain.

- **Reference Model**

  A reference model **describes the important concepts and relationships** in the domain focusing on what distinguishes the elements of the domain.

epts
event processing technical society

DEBS2010
4th ACM International Conference on Distributed Event-Based Systems

Motivation and Benefits

- **Motivation**
  - Event Processing is evolving from many existing technologies and creating or adapting different tools
  - Potential adopters (stakeholders) may have problems understanding and adequately defining EP-based architectures and solutions.
- **Benefits**
  - ***a Reference Architecture*** aids efficient Event Processing solution development, by predefining *customizable abstract frames of reference for specific stakeholder concerns and application domains*.
    - aids in reusability of successful EP architectures for frequently occurring EP design problems
    - enables easier comparison of proposed EP solutions
  - Underlying **Reference Model** defines / explains the *terminology and components in Event Processing architectures*

ANSI/IEEE Std 1471 :: ISO/IEC 42010 Methodology

- ***Recommended Practice for Architectural Description of Software-intensive Systems***
  - Now an **ISO/IEC 42010:2007 standard**
  - Includes 6 elements
    1. **Architectural description**
    2. System **stakeholders** and their concerns
    3. One or more architectural **views**
    4. **Viewpoints**
    5. A **record of** all known **inconsistencies** among the architectural description's required constituents
    6. A **rationale for selection** of the architecture

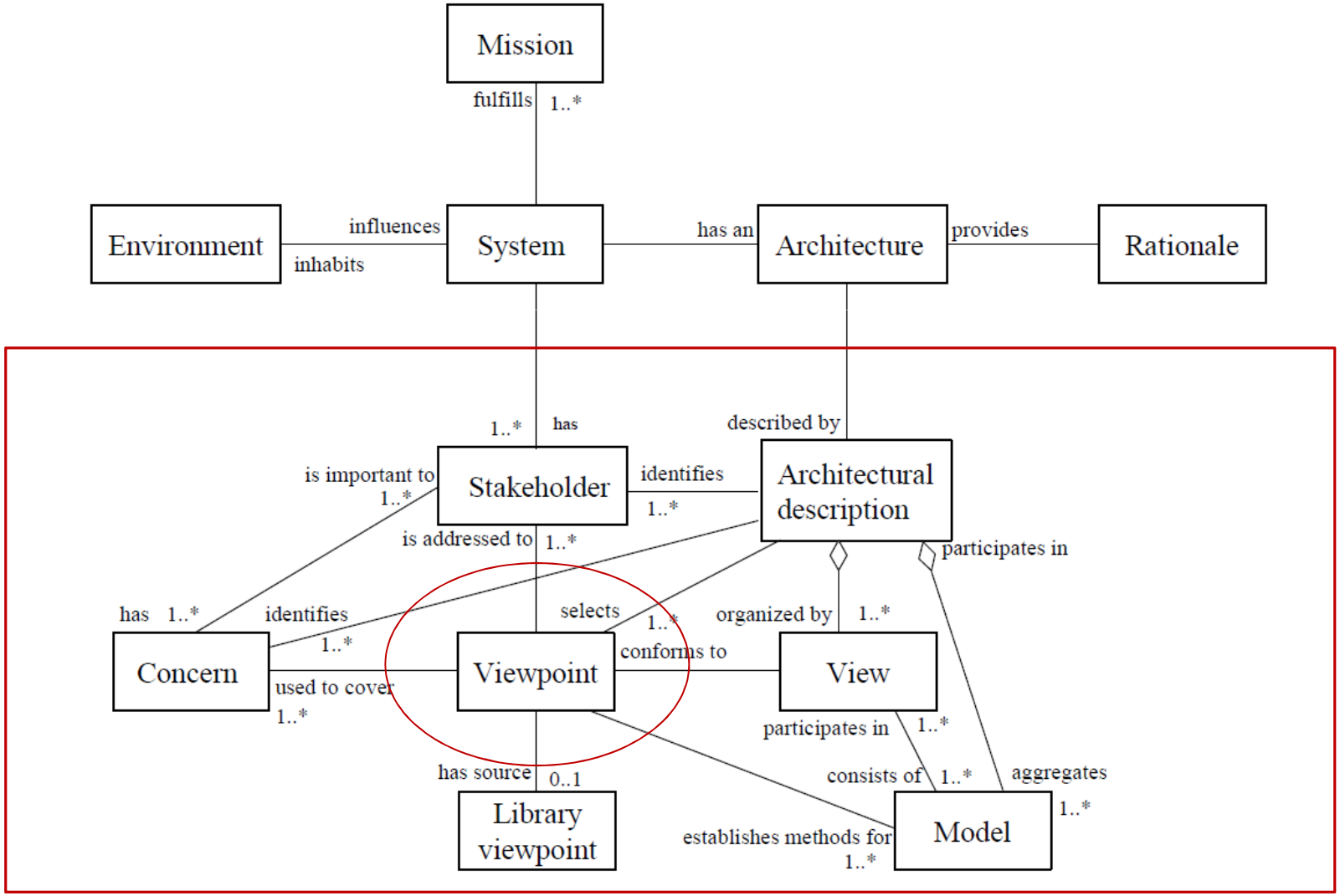
ISO/IEC 42010:2007 Terminology (1)

- **Architecture**
  - **The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding ist design and evolution.**
- **Architectural Description**
  - **A collection of products that document the architecture.**
- **System**
  - **A collection of components organized to accomplish a specific function or set of functions.**
- **System Stakeholder**
  - **A system stakeholder is an individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system**

ISO/IEC 42010:2007 Terminology (2)

- **View**
  - **A representation of the whole system from the perspective of a related set of concerns.**
- **Viewpoint**
  - **A specification of the conventions for constructing and using a view - a pattern or template which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.**
- **Model**
  - A view may consist of one or more ***models and a model may participate in one or more views***.
  - Each model is defined according to the **methods established in the corresponding viewpoint definition**.

Conceptual model of architectural description from IEEE Std 1471- 2000 (ISO/IEC 42010:2007)



Declaring a Viewpoint

- ***Each viewpoint is specified by:***
  - Viewpoint name
  - The stakeholders addressed by the viewpoint
  - The stakeholder concerns to be addressed by the viewpoint
  - The viewpoint language, modeling techniques, or analytical methods used
  - The source, if any, of the viewpoint (e.g., author, literature citation)
- ***A viewpoint may also include:***
  - Any consistency or completeness checks associated with the underlying method to be applied to models within the view
  - Any evaluation or analysis techniques to be applied to models within the view
  - Any heuristics, patterns, or other guidelines which aid in the synthesis of an associated view or its models

Architecture Views - Examples

- **Domain Architecture**

  domain models represent domain requirement and logic

  - e.g. process representation, use case analysis, free text

- **Application Architecture**

  structural, logic architecture

  – **Functional Layering**, e.g. functional multi-tier structure

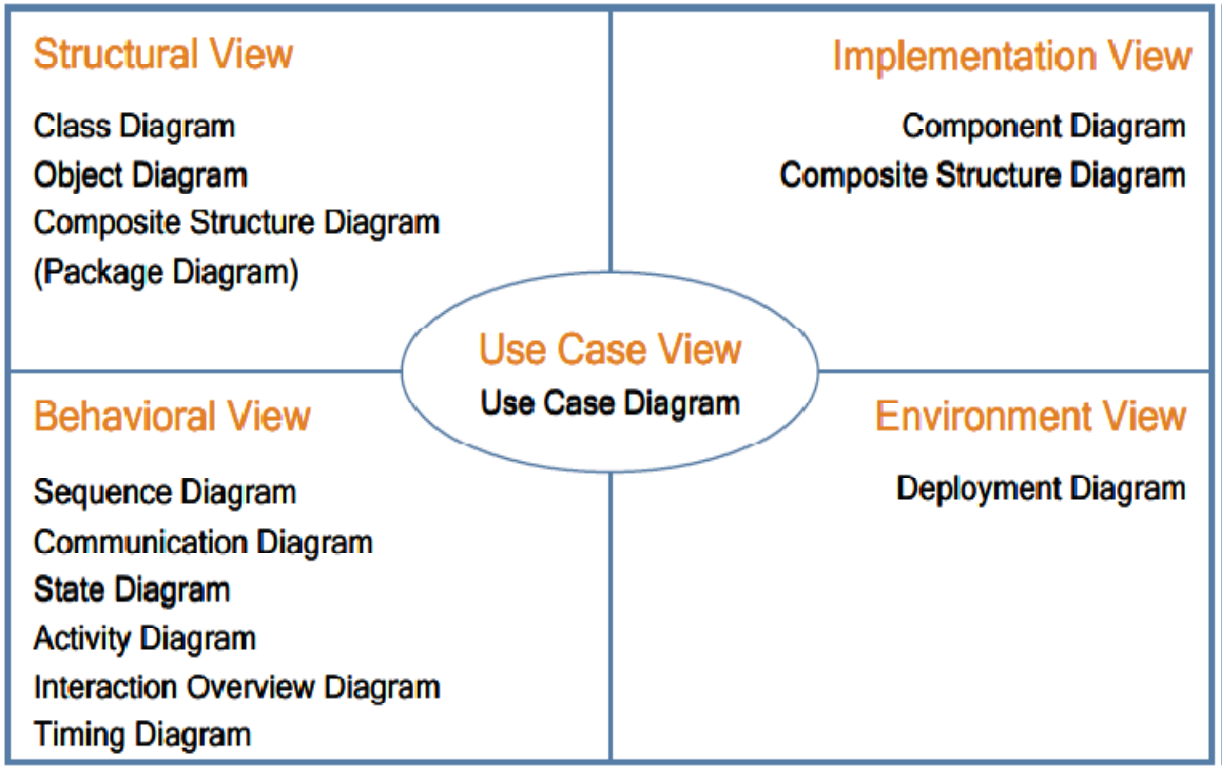
  – **Functional Decomposition**, e.g. Component Architecture, UML (class/activity/sequence)

- **System Architecture**

  – Concrete implementation of an architecture in a target platform

Example UML Architectural Views and Diagrams

UML defines 13 diagram models that describe 4+1 architectural views



4+1 architectural views model by Philippe Kruchten, IBM

Event Processing Reference Architecture Viewpoints

Viewpoint Element	**Viewpoint**		
	Engineering EP Architecture	*Managing EP Architecture*	*Business with EP Architecture*
Concepts	How to implement?	How to apply?	How to utilize / sell / own?
Stakeholders	Architects / Engineers	Project Manager	Decision Maker, Customer, Provider
Concerns	Effective construction and deployment	Operational Management	Strategic and tactical management
Techniques / Languages	Modeling, Engineering	IT (service/appl) management, project management	Monitoring, Enterprise Decision Management, Governance

epts
event processing technical society

DEBS2010
4th ACM International Conference on Distributed Event-Based Systems

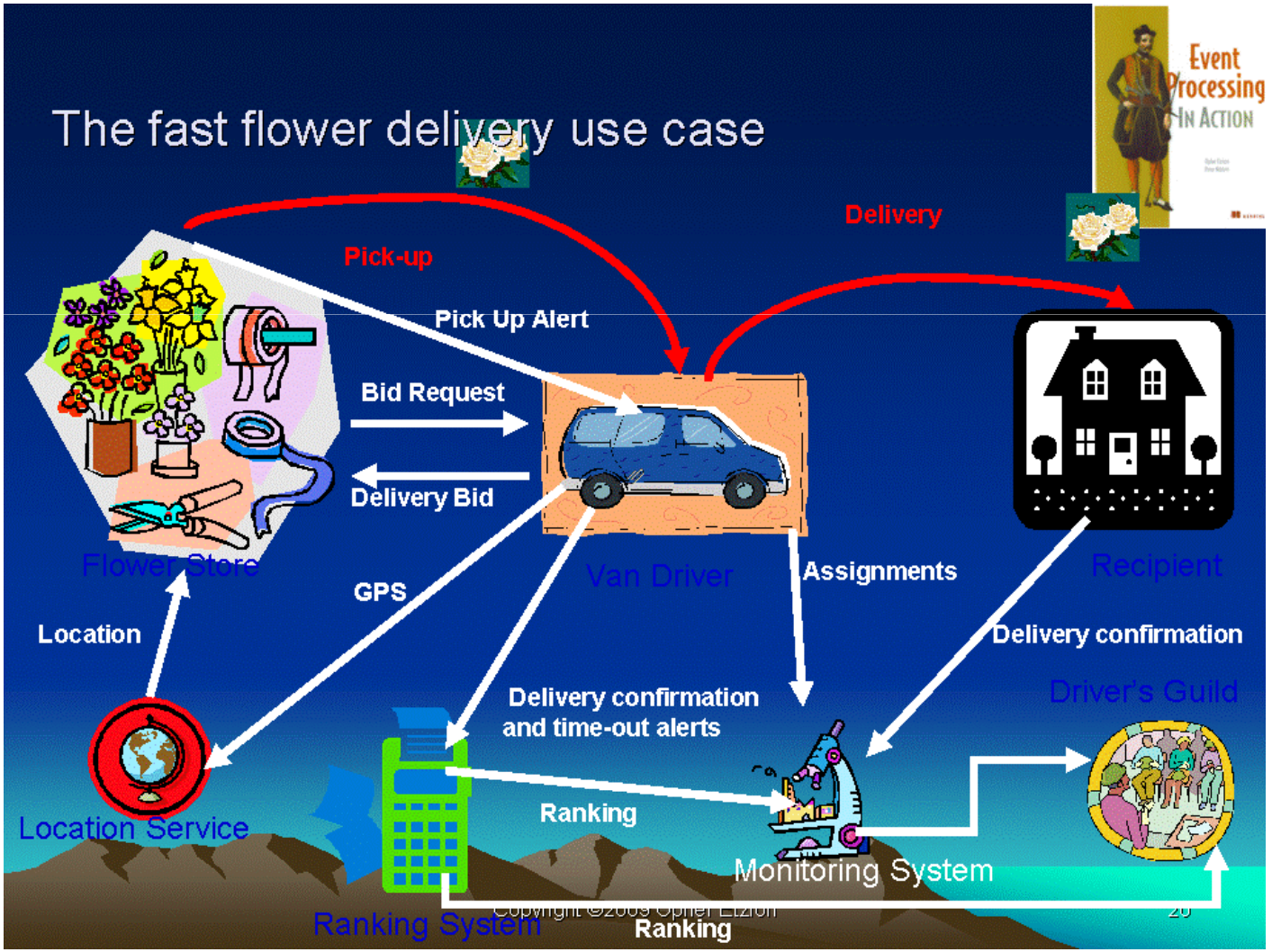
Agenda

- Introduction to architectures, architecture methodologies, and event processing
- **Member architectures and salient features**
- Skeleton reference architecture from EPTS Reference Architecture Working Group
- Summary and future work of the EPTS Reference Architecture Working Group

Overview – Sample Member Architectures

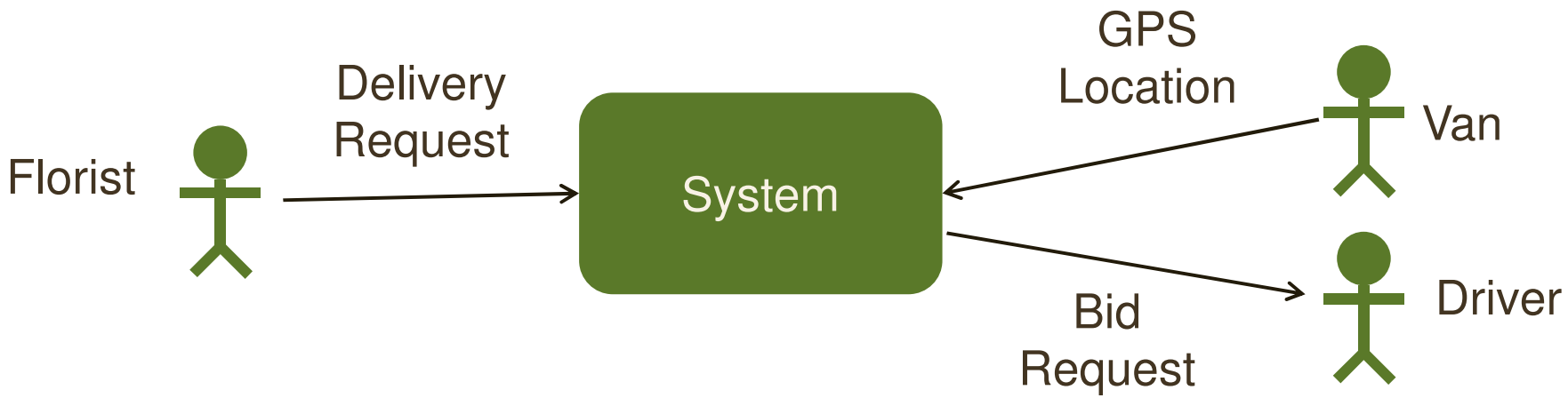
- **University of Trento - Themis Palpanas**
- **TIBCO – Paul Vincent**
- **Oracle - Alex Alves**
- **IBM - Catherine Moxey**

Example Use Case



Courtesy of Opher Etzion & Peter Niblett, IBM

Use Case Detail: FFD delivery and bid requests



High level Description

Stakeholders vs Events (high level)

Id	Precondition	Event	Stakeholder sending	Stakeholder receiving	Other actions
1.		Delivery Request	Florist	System	System also enriches event with "minimum ranking" based on florist
2.		GPS Location	Van	System	System maps GPS data to "city region" for van
3.	Delivery Request matched to Drivers based on ranking, nearby region	Bid Request	System	Driver	

Overall Preconditions

None described.

Postconditions

Bid sent out OR Bid not sent out (an exception).

FDD Use Case: Stakeholders, Events, and Details

Detailed Description

Stakeholders vs Events (detail)

Step	Actor Actions	Requirement for System Response
1.	Florist places Delivery Request	Accepts Delivery Request
2.	System accepts Delivery Request	Enriches Delivery Request, creates Bid List from Driver List, filters Bid List, creates Bid Request
3.	Van sends LocationEvent	Maps to Van Region Location, updates DriverList

Extensions

branches from main flow for special conditions

- NoBidderFound

Step	Actor Actions	System Response
2b.	System accepts Delivery Request	send Rejection to Florist

Variations

TBA

Frequency

Estimates per event are:

Event	Stakeholder sending	frequency
Delivery Request	Florist	# deliveries per day (max and average) per florist: 30 and 10; #florists: 20
GPS Location	Van	# update events per GPS system per hour: 60; # vans: 20
Bid Request	System	# bid requests ~= # deliveries per day

Overview – Sample Member Architectures

- **University of Trento - Themis Palpanas**
- **TIBCO – Paul Vincent**
- **Oracle - Alex Alves**
- **IBM - Catherine Moxey**

epts
event processing technical society

DEBS2010
4th ACM International Conference on Distributed Event-Based Systems

Entity Name System Service for CEP Entity Identification

- what is the problem?
  - different pieces of data may refer to the same real world entity
    - entity: person, organization, location, conference, piece of hardware, etc
  - references to entities are often in an event's payload

Entity Identification

- what is the problem?
  - different pieces of data may refer to the same real world entity
    - entity: person, organization, location, conference, piece of hardware, etc
  - references to entities are often in an event's payload
- why is this problem important?
  - incorrect identification of entities may lead to
    - erroneous decisions in subsequent processing steps
    - reduced overall performance quality

Entity Identification

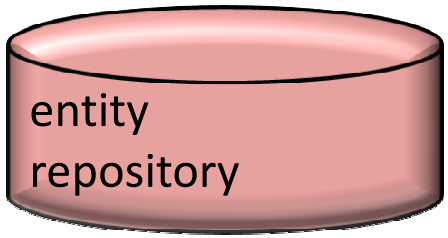
- what is the problem?
  - different pieces of data may refer to the same real world entity
    - entity: person, organization, location, conference, piece of hardware, etc
  - references to entities are often in an event's payload
- why is this problem important?
  - incorrect identification of entities may lead to
    - erroneous decisions in subsequent processing steps
    - reduced overall performance quality
- why is this problem relevant?
  - events originate from several different sources
  - sources may refer to same real-world entity in different ways
  - if entity identification problem not tackled, event filtering/correlation/etc. will produce wrong results

Our solution

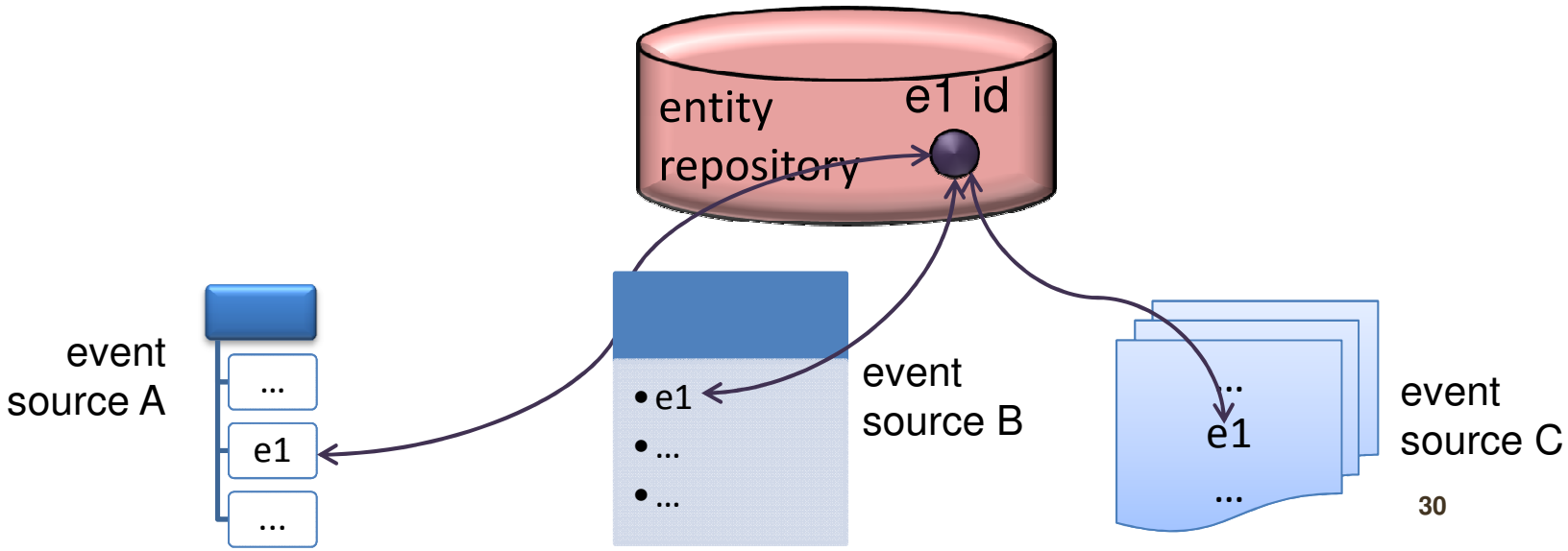
- the Entity Name System (see References [1][2])
  - scalable infrastructure for assigning and managing unique global identifiers for named entities

- basic ideas
  - any description of an entity is “resolved” into its global ID
  - not a universal knowledge base about entities
    - stores minimal amount of information about an entity for
      - distinguishing entities one from another
      - finding entities and their identifiers as a result of a query
  - similar to lightweight master data management system

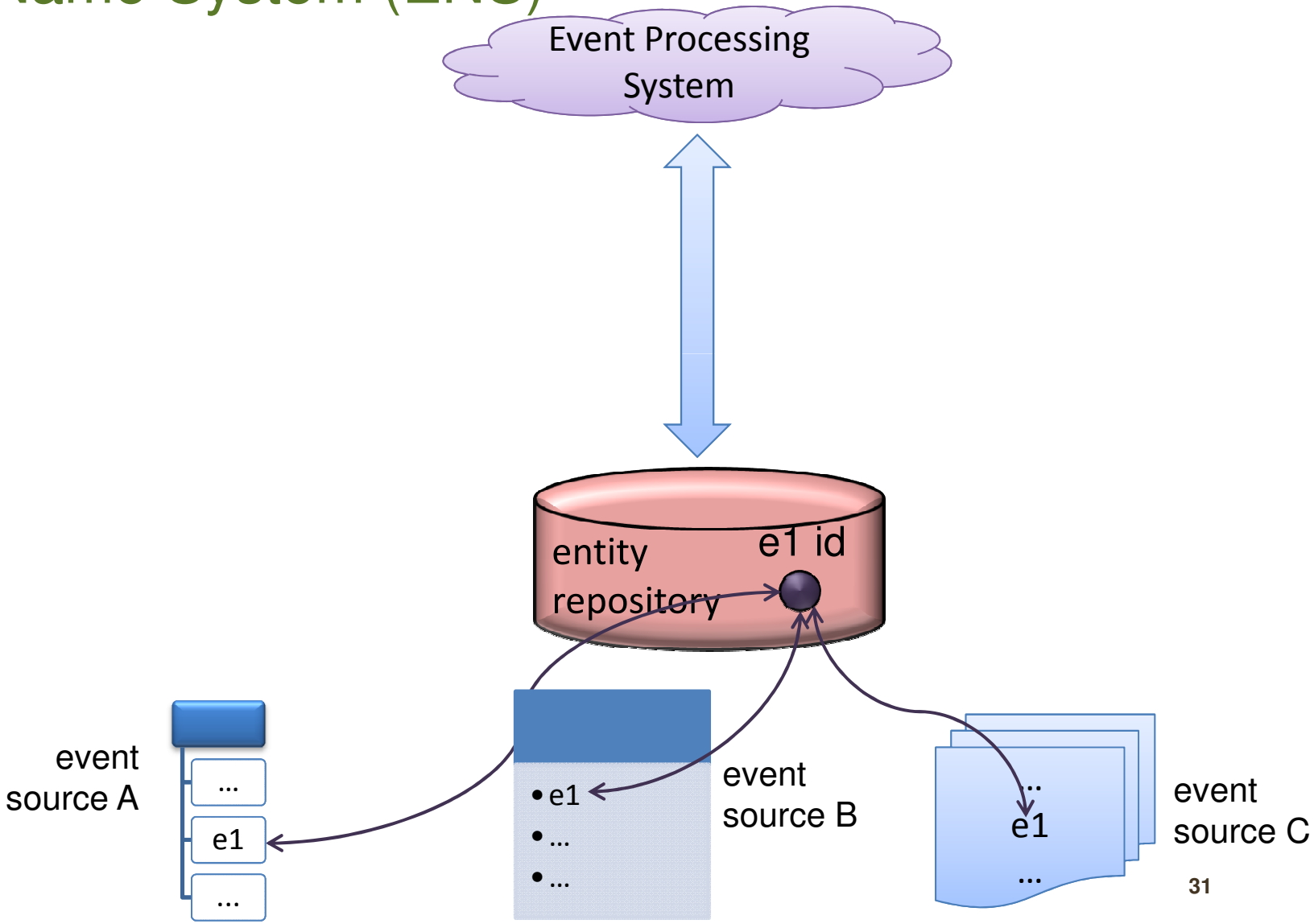
Entity Name System (ENS)



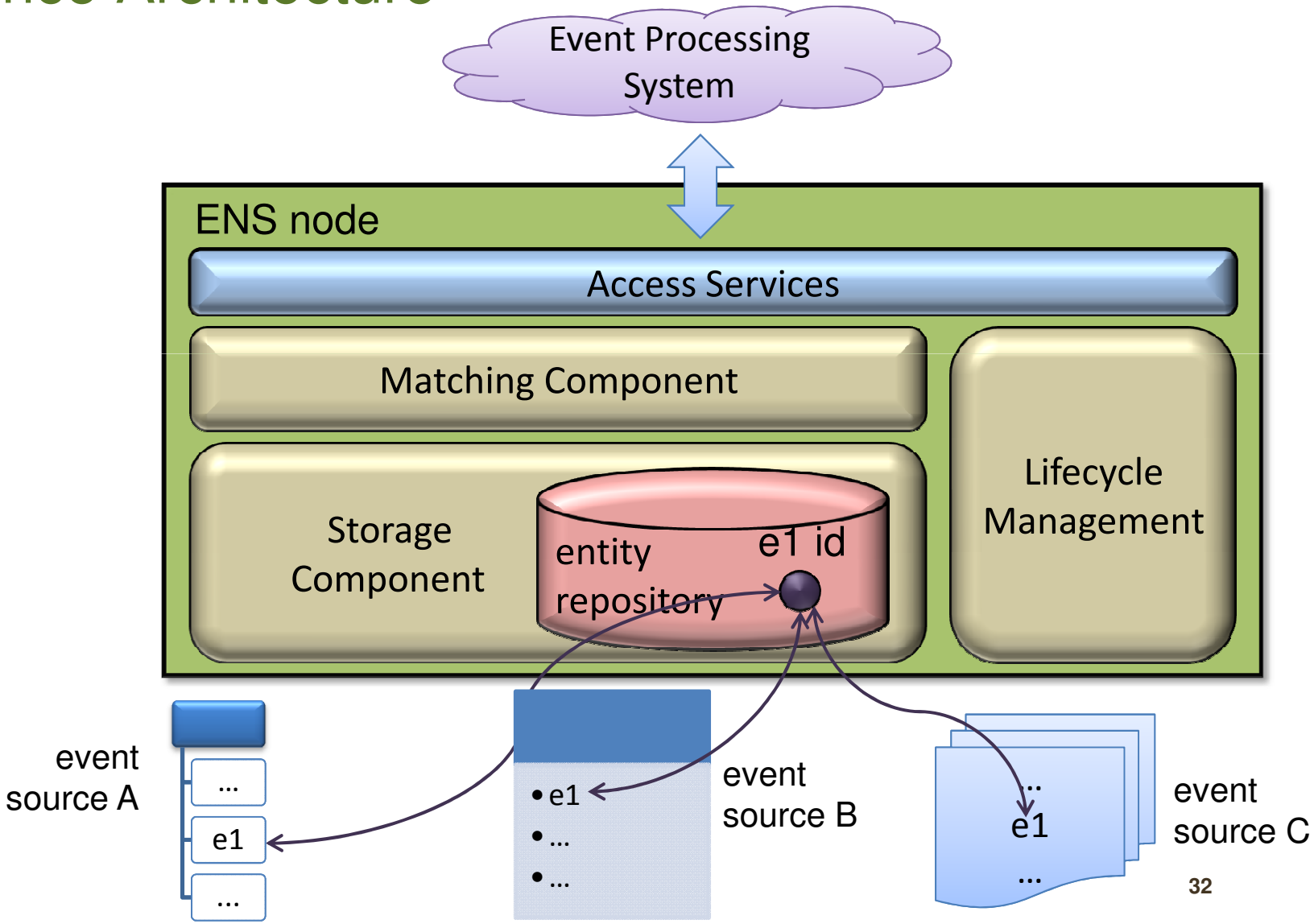
Entity Name System (ENS)



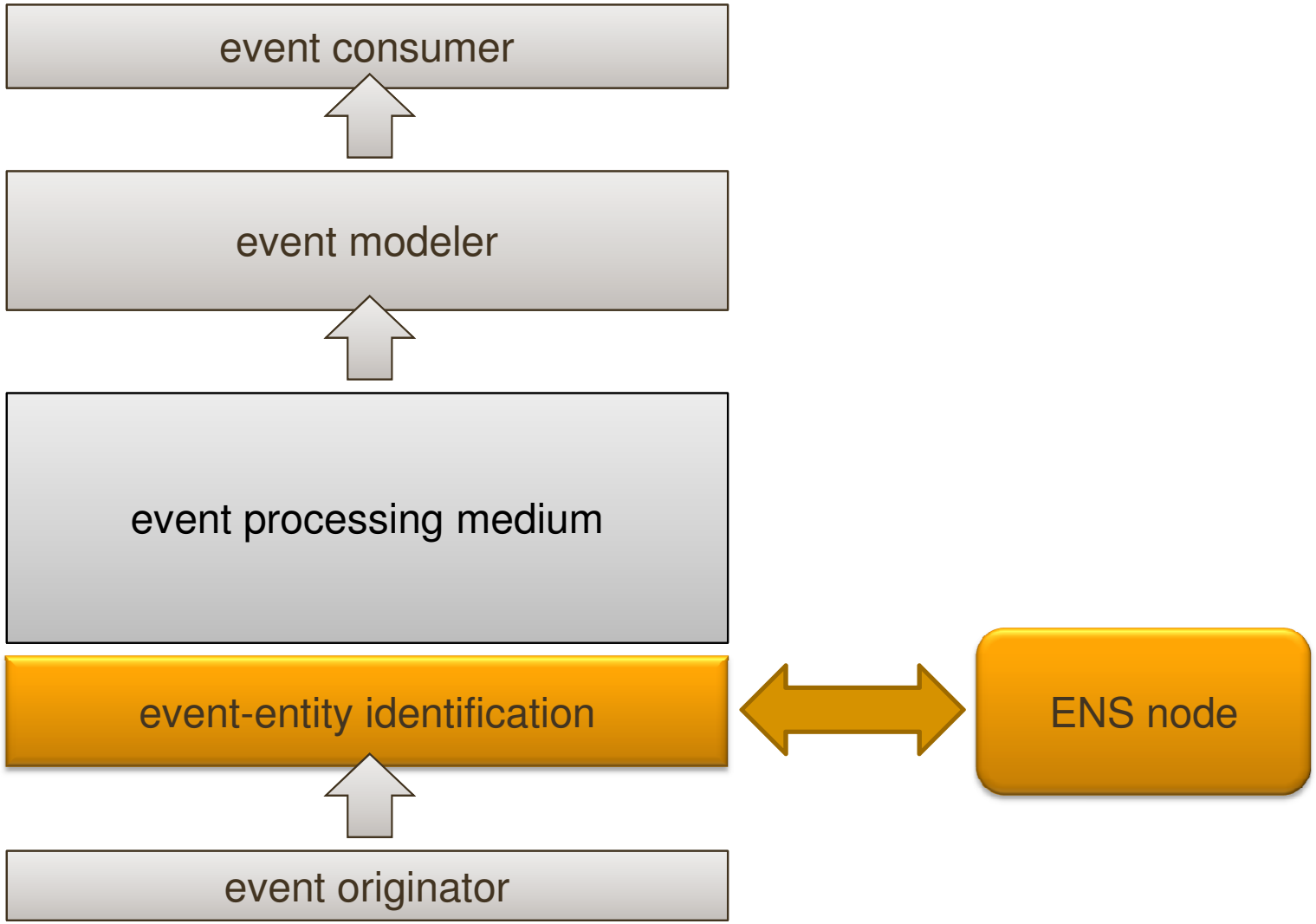
Entity Name System (ENS)



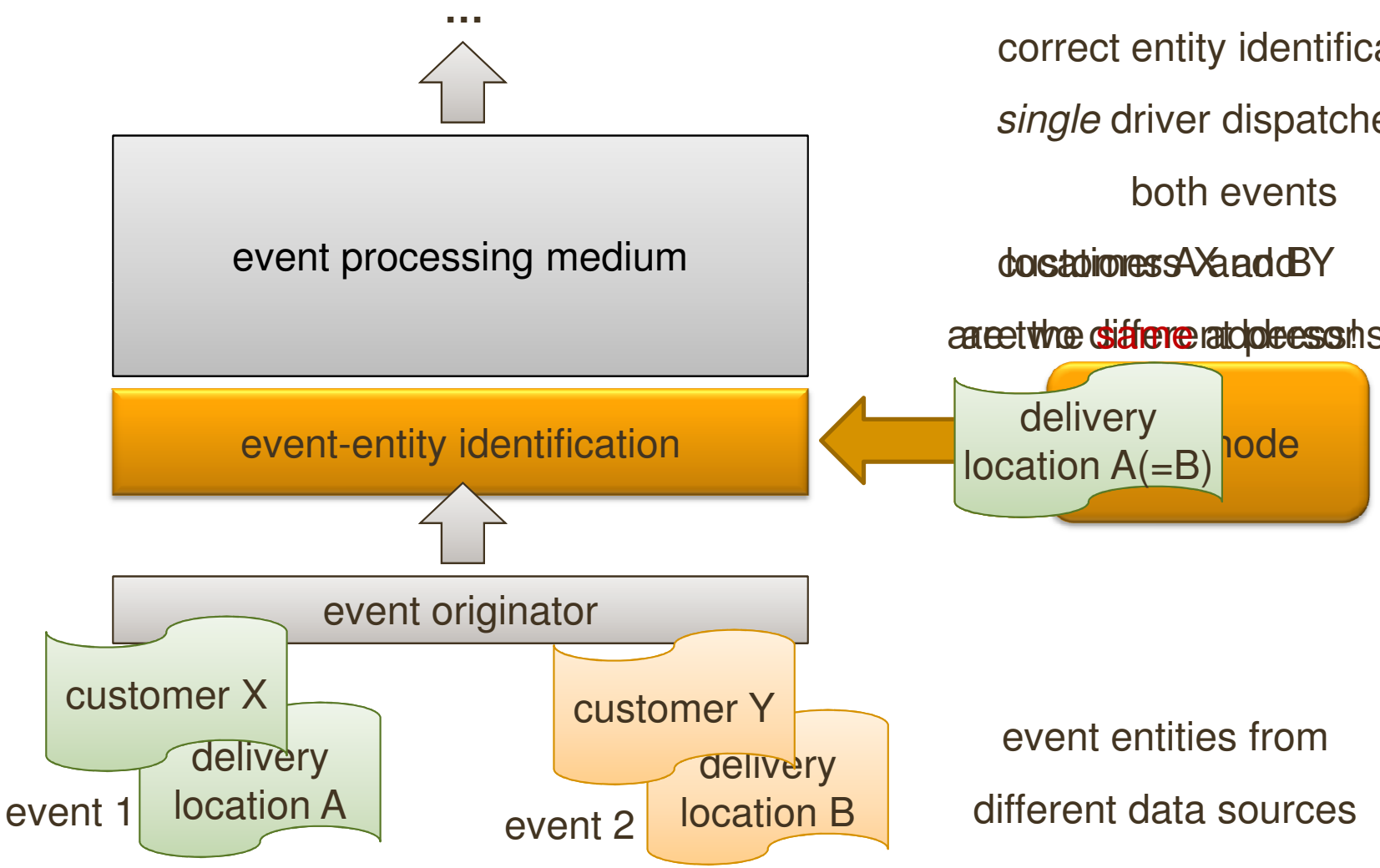
Entity Name System (ENS): Reference Architecture



Entity Name System Service



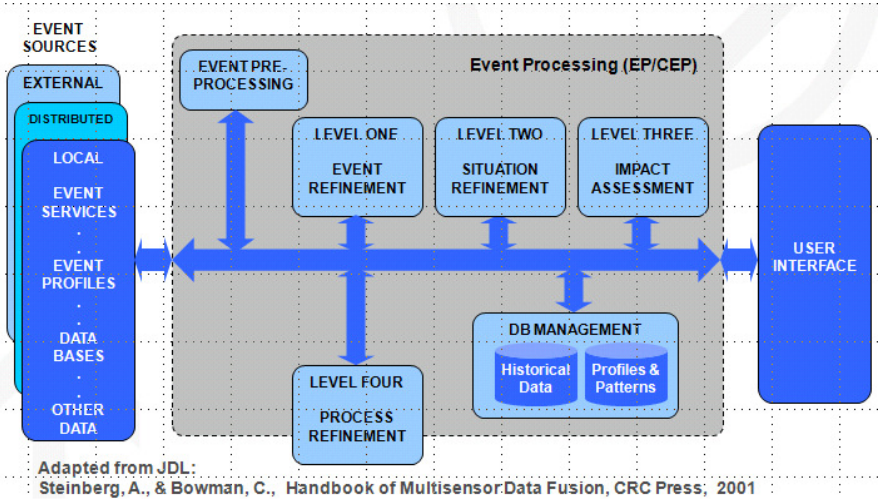
Entity Name System Service: Use Case

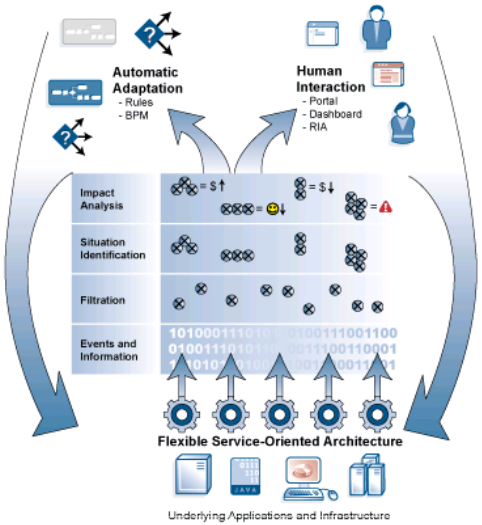


Overview – Sample Member Architectures

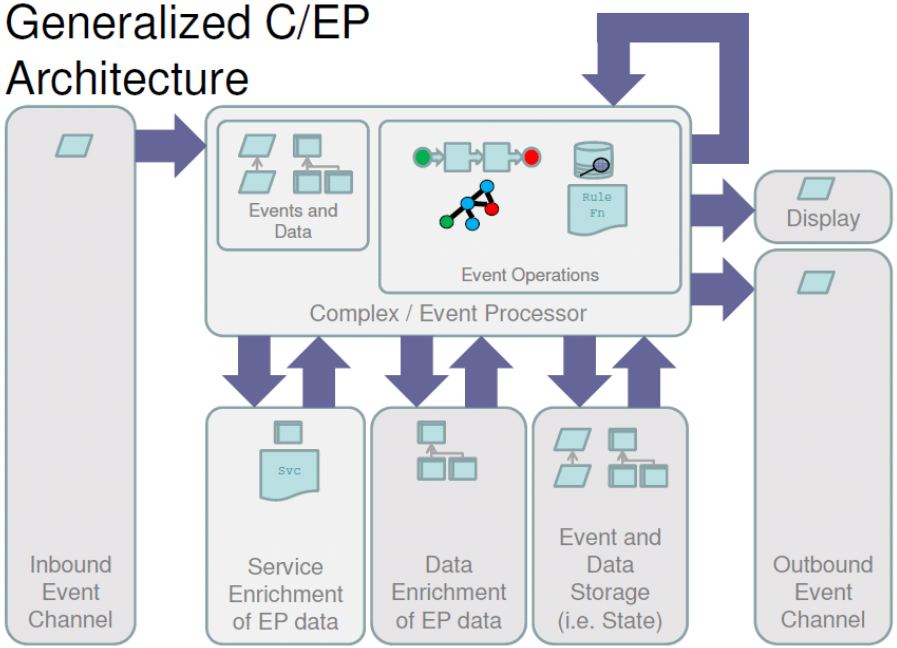
- University of Trento - Themis Palpanas
- **TIBCO – Paul Vincent**
- Oracle - Alex Alves
- IBM - Catherine Moxey

TIBCO Reference Architecture(s)

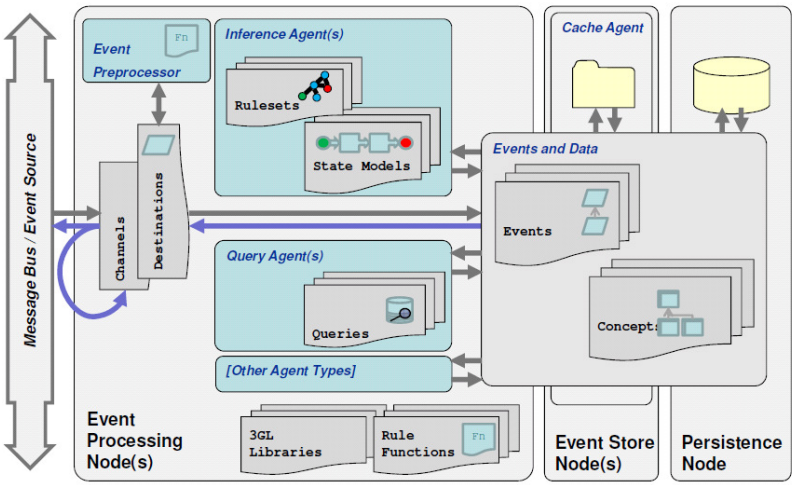




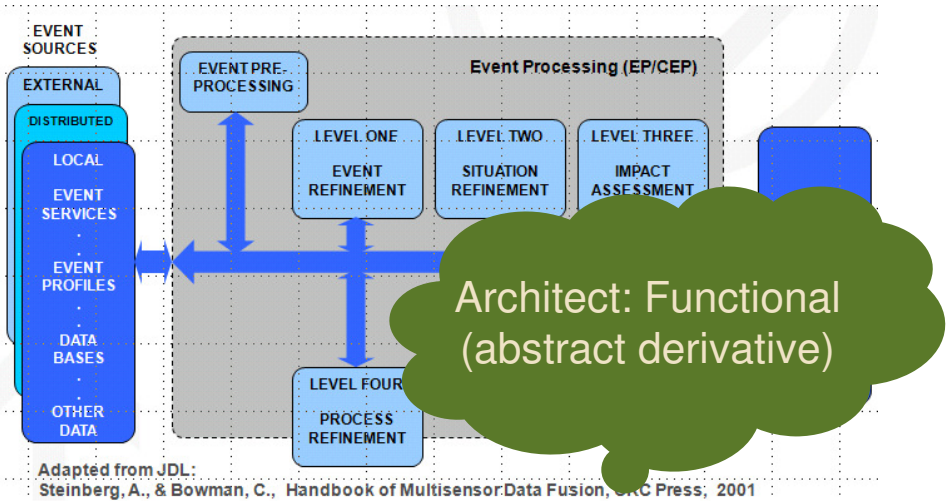
Generalized C/EP Architecture



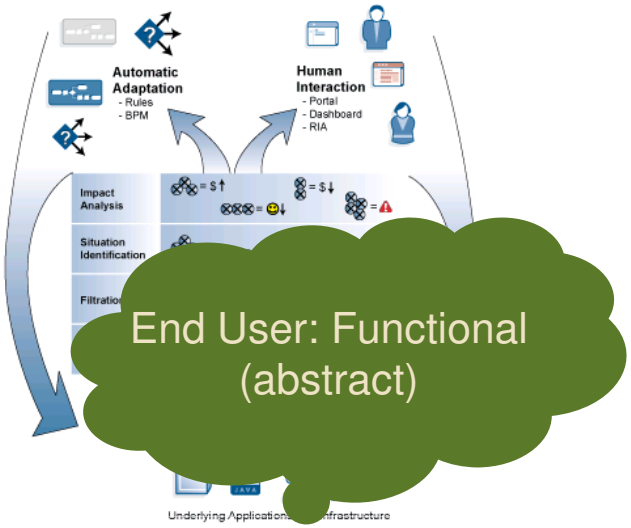
CEP Agent Architecture



TIBCO Reference Architecture(s)

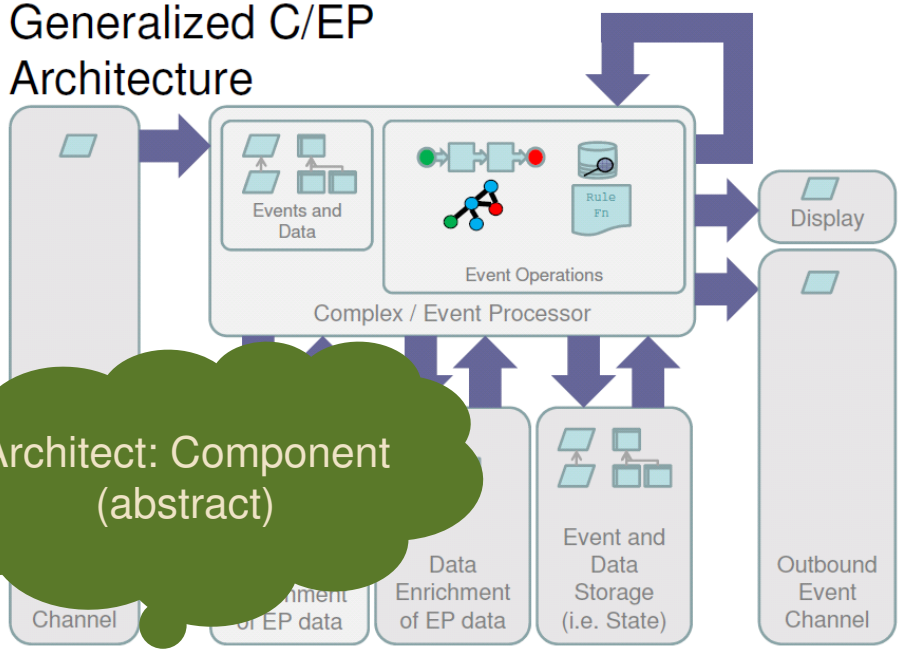


Architect: Functional (abstract derivative)



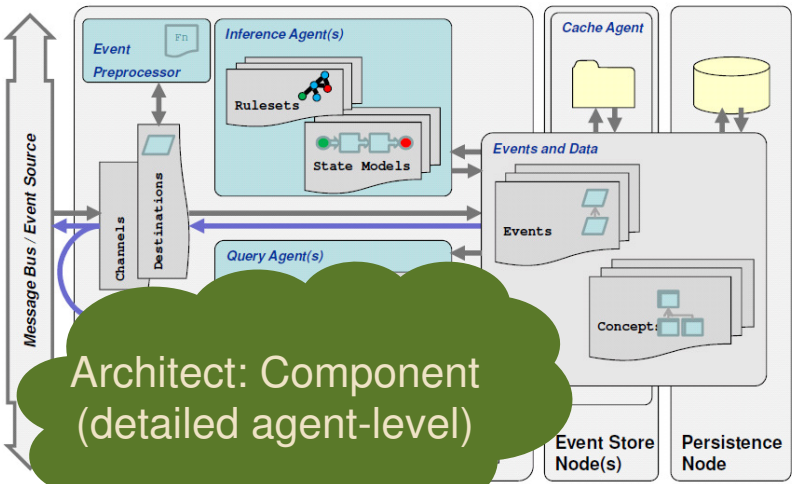
End User: Functional (abstract)

Generalized C/EP Architecture



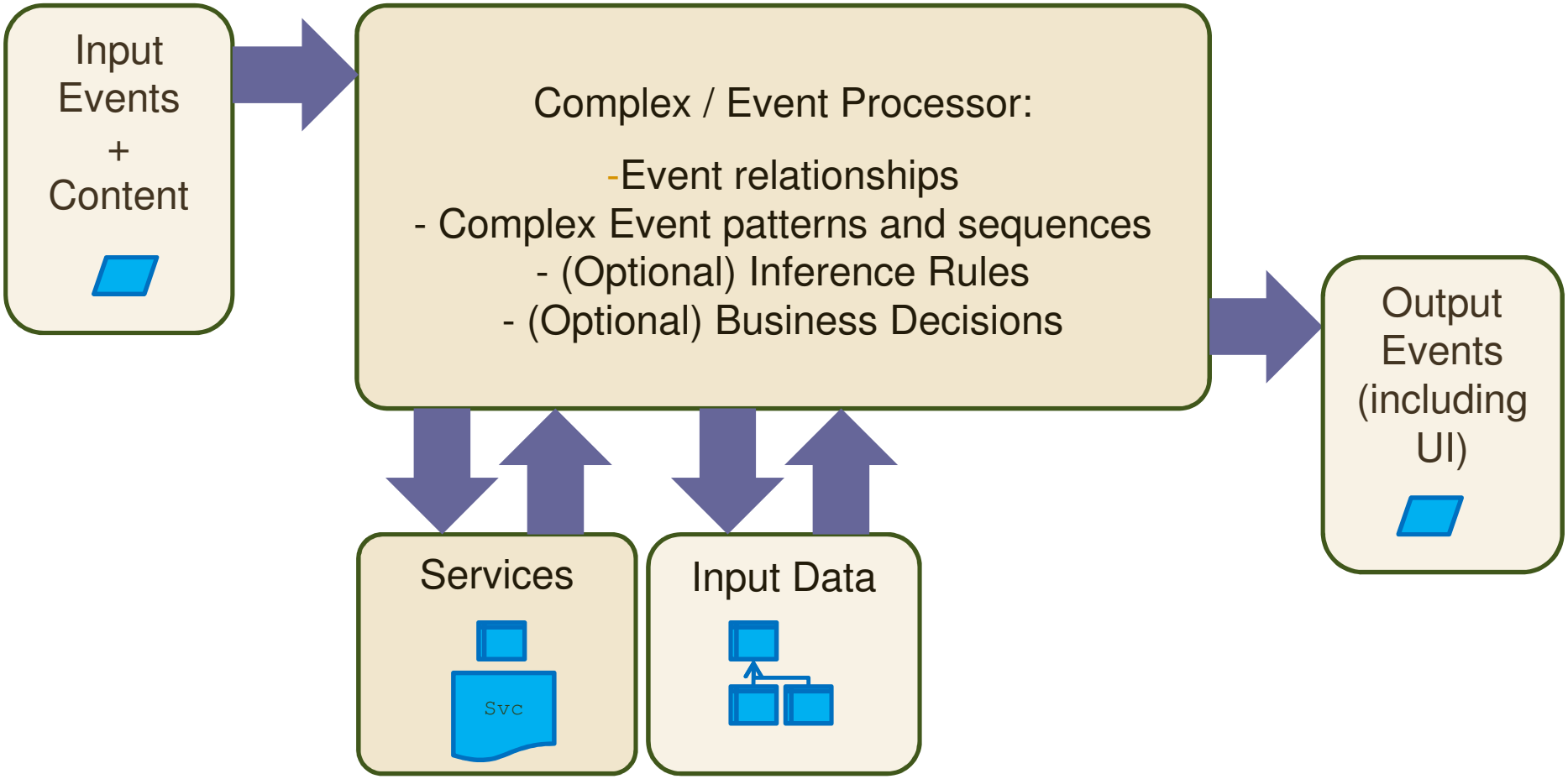
Architect: Component (abstract)

CEP Agent Architecture

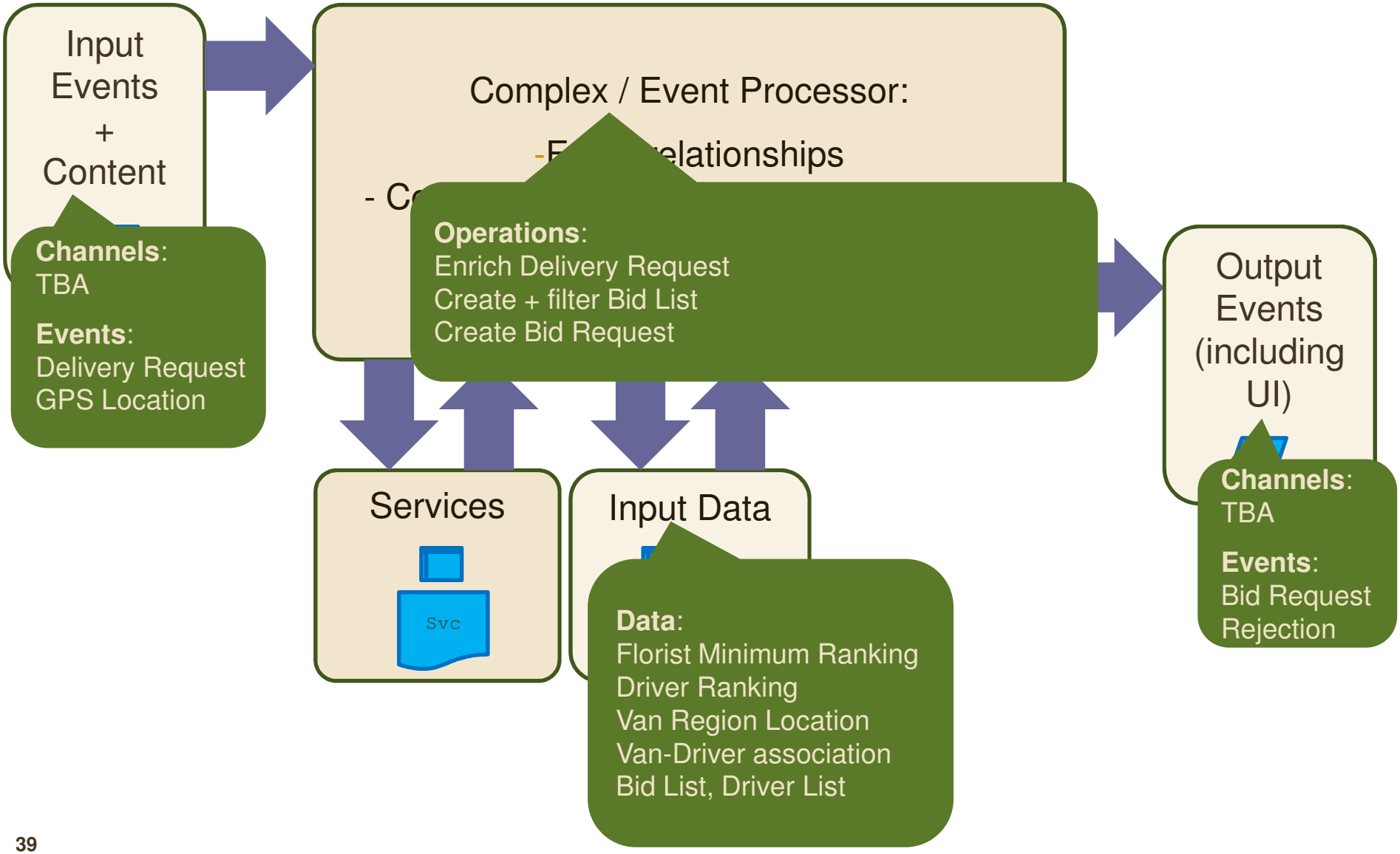


Architect: Component (detailed agent-level)

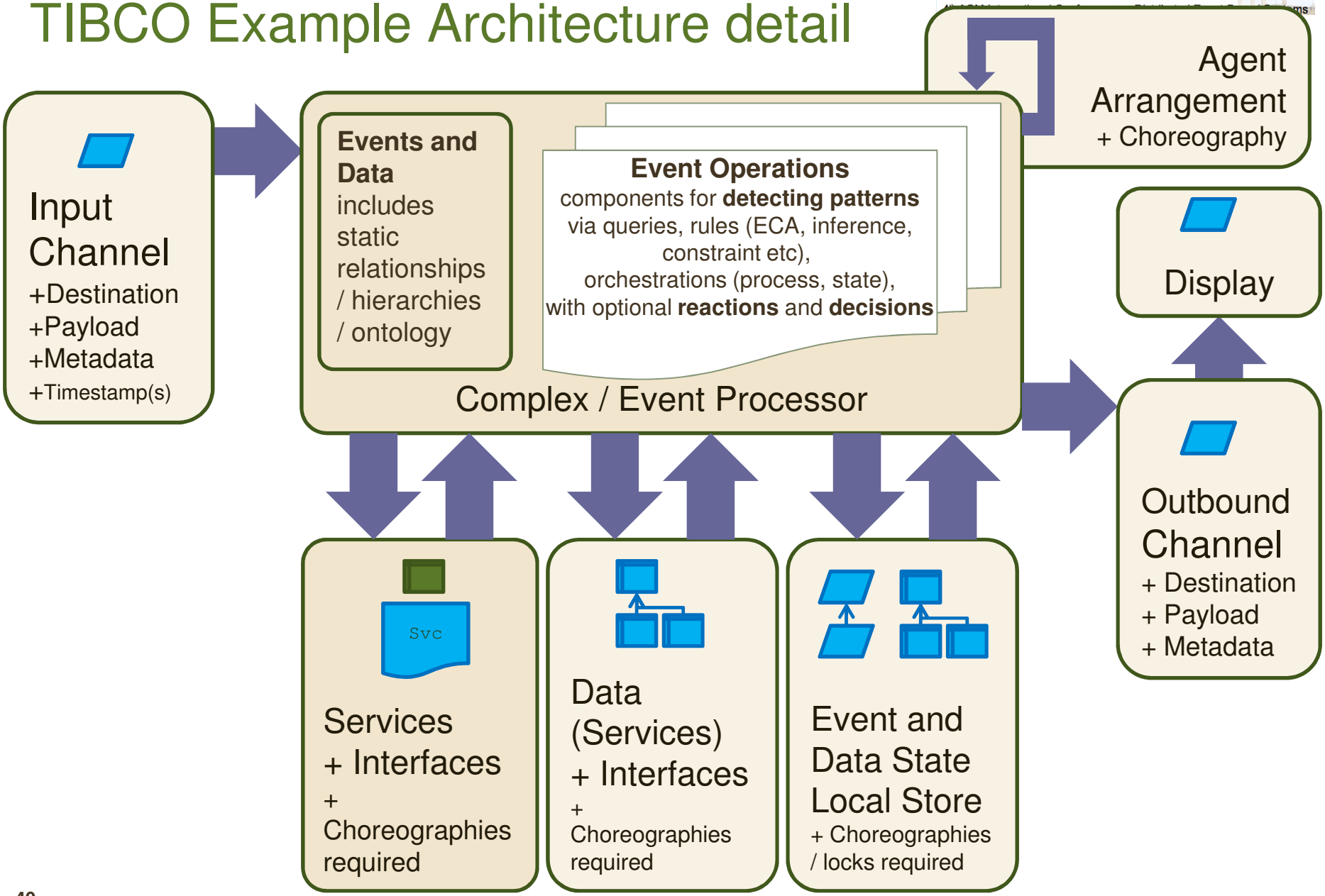
TIBCO Example Architecture



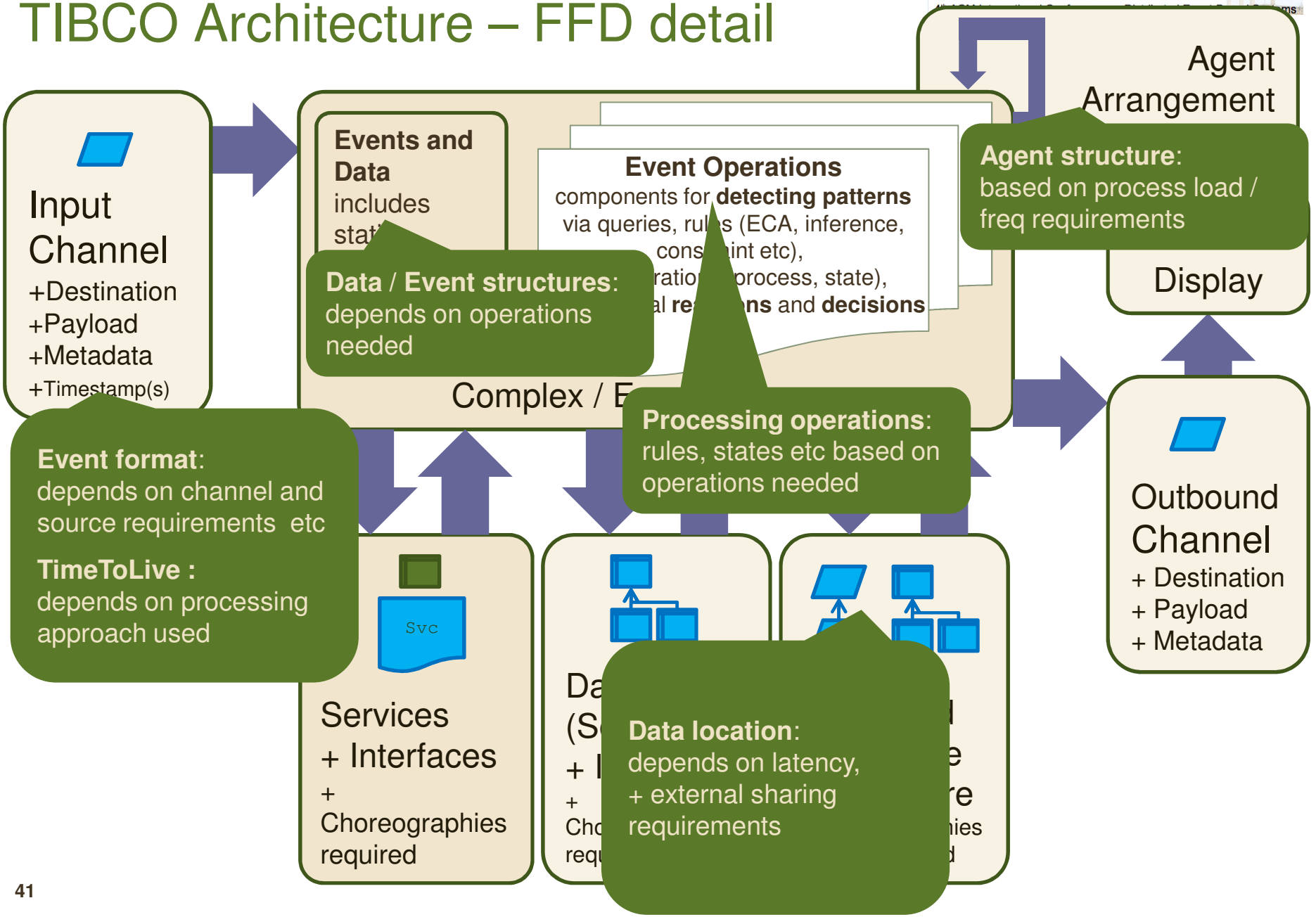
TIBCO Example Architecture for FFD



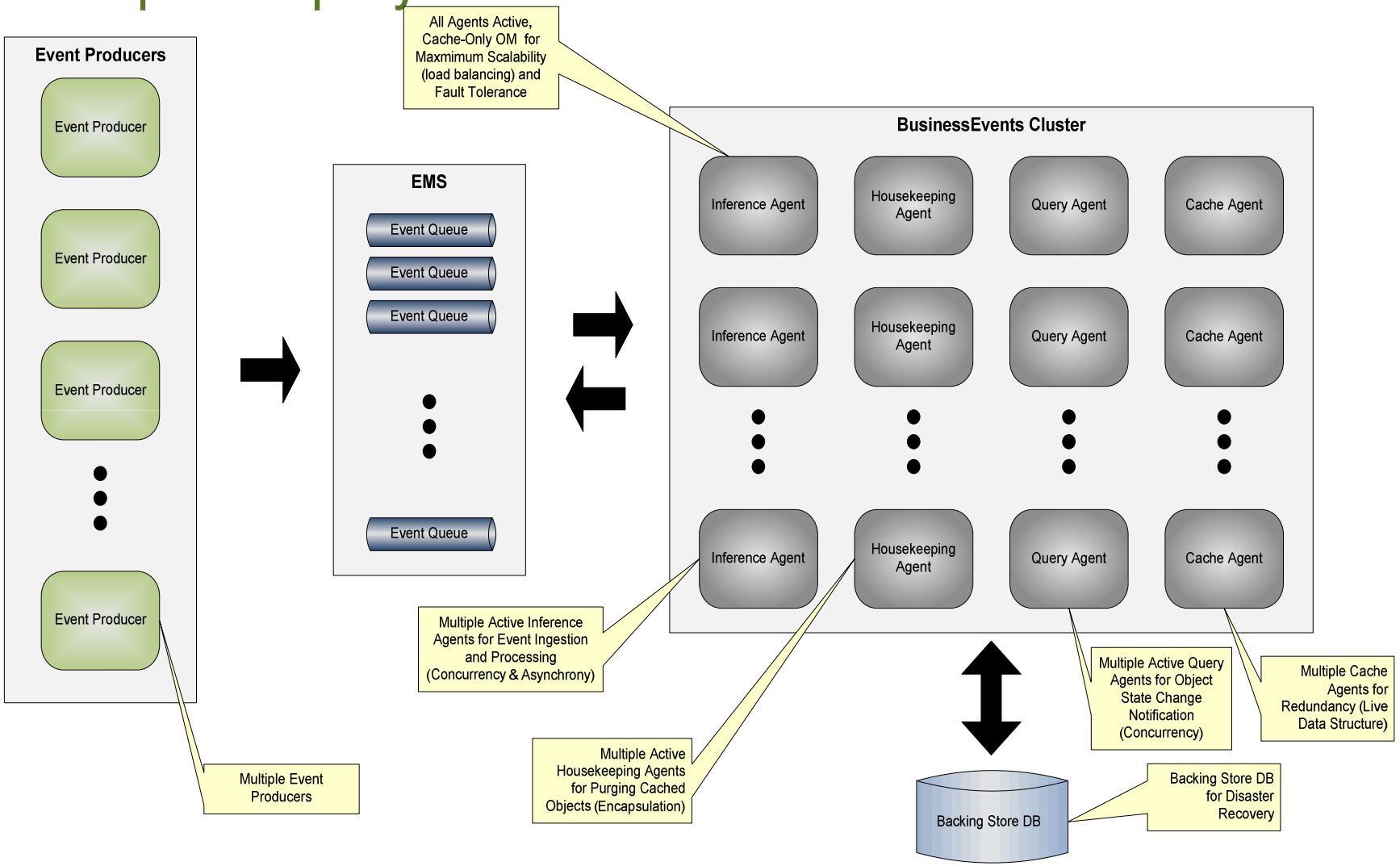
TIBCO Example Architecture detail



TIBCO Architecture – FFD detail



Example Deployment Architecture

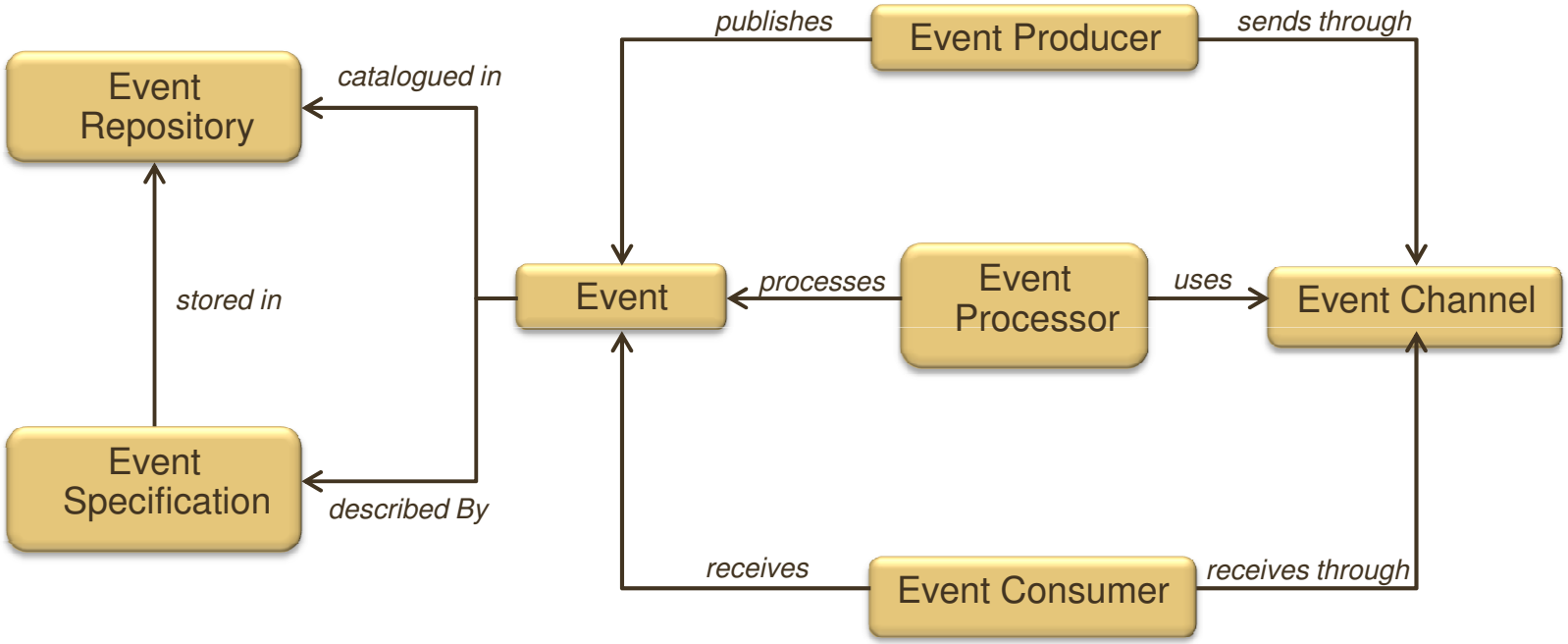


Distributed Event Processing Agents – see ref.

Overview – Sample Member Architectures

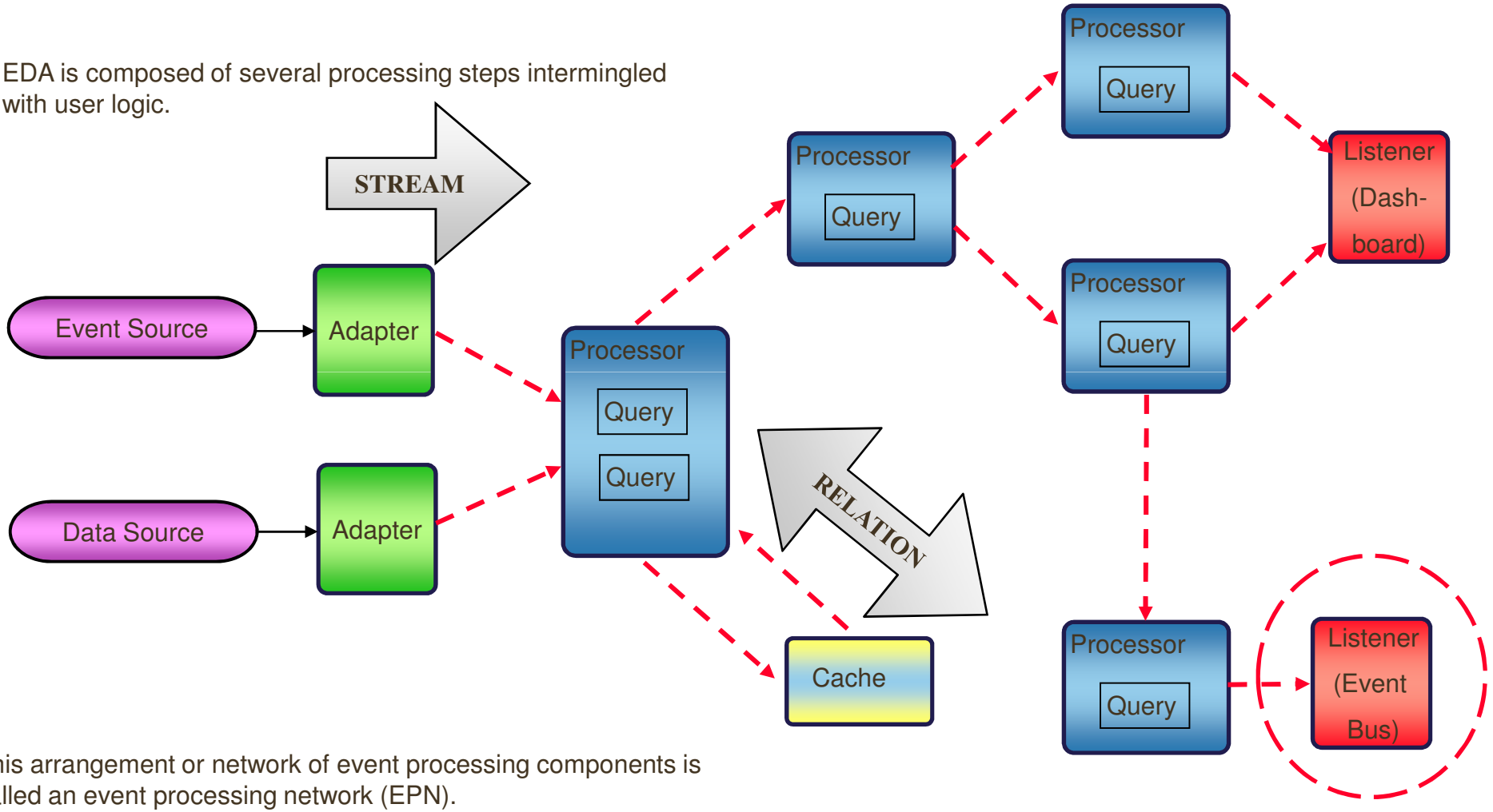
- University of Trento - Themis Palpanas
- TIBCO – Paul Vincent
- **Oracle - Alex Alves**
- IBM - Catherine Moxey

Conceptual View



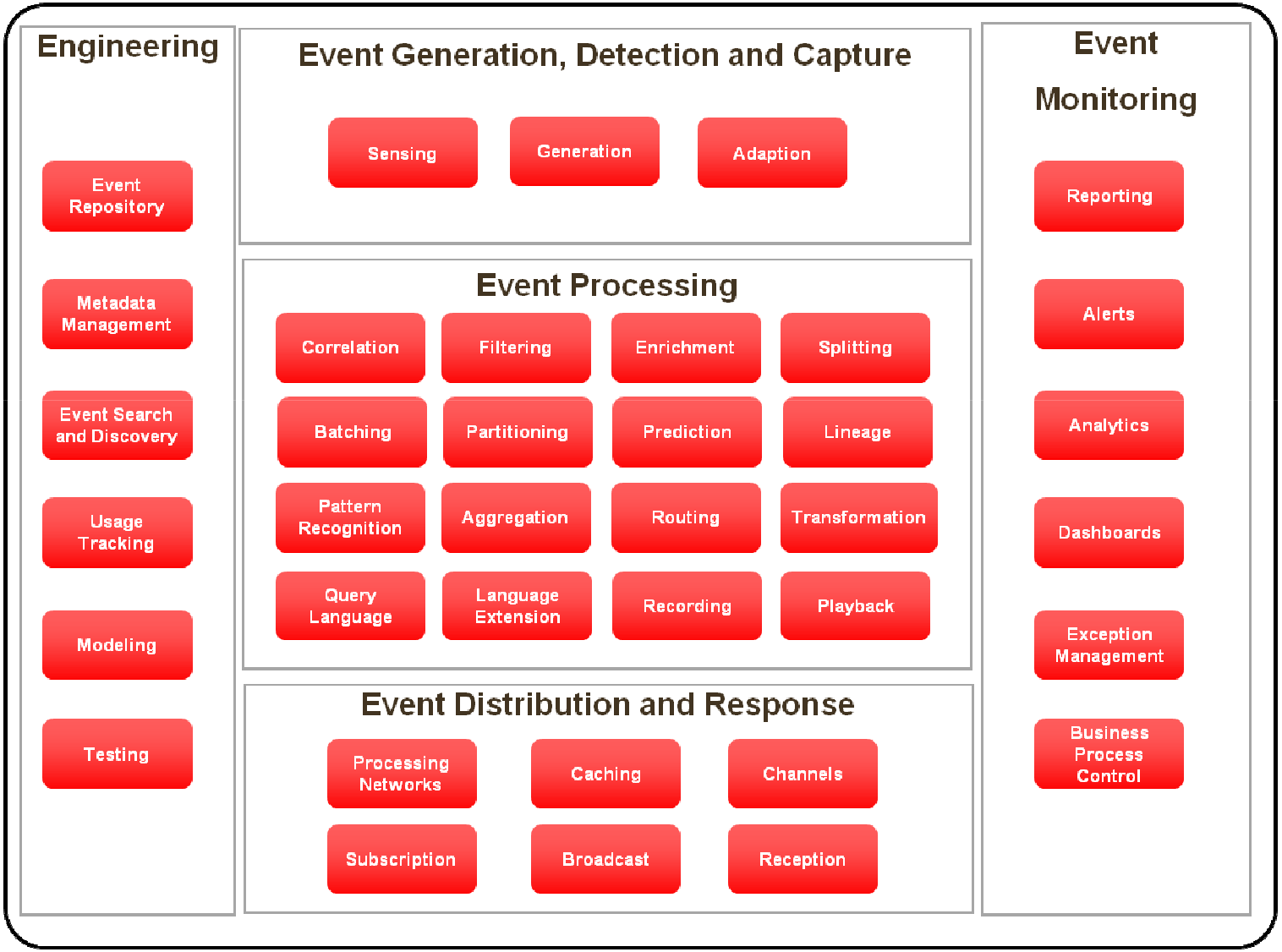
Logical View

EDA is composed of several processing steps intermingled with user logic.

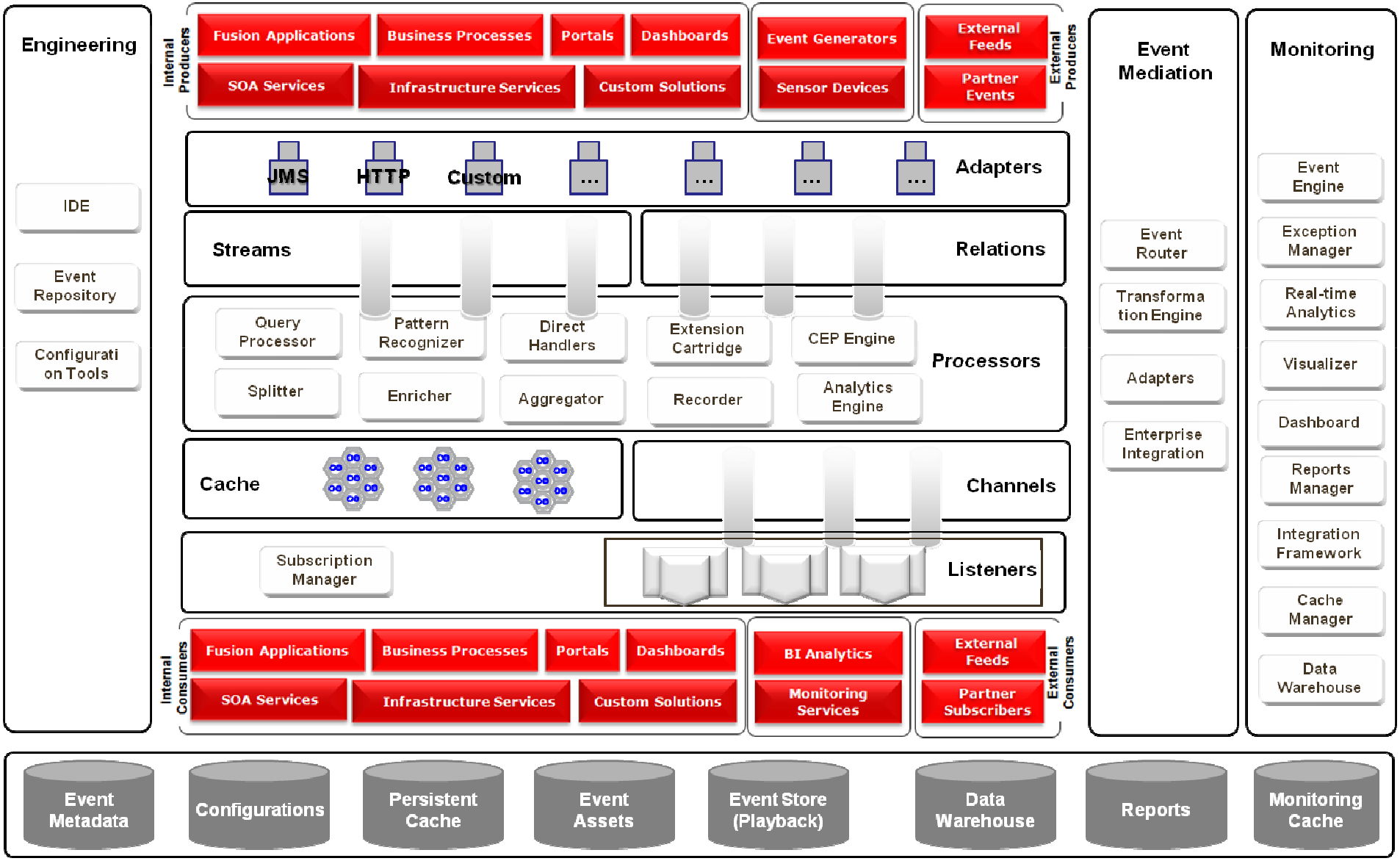


This arrangement or network of event processing components is called an event processing network (EPN).

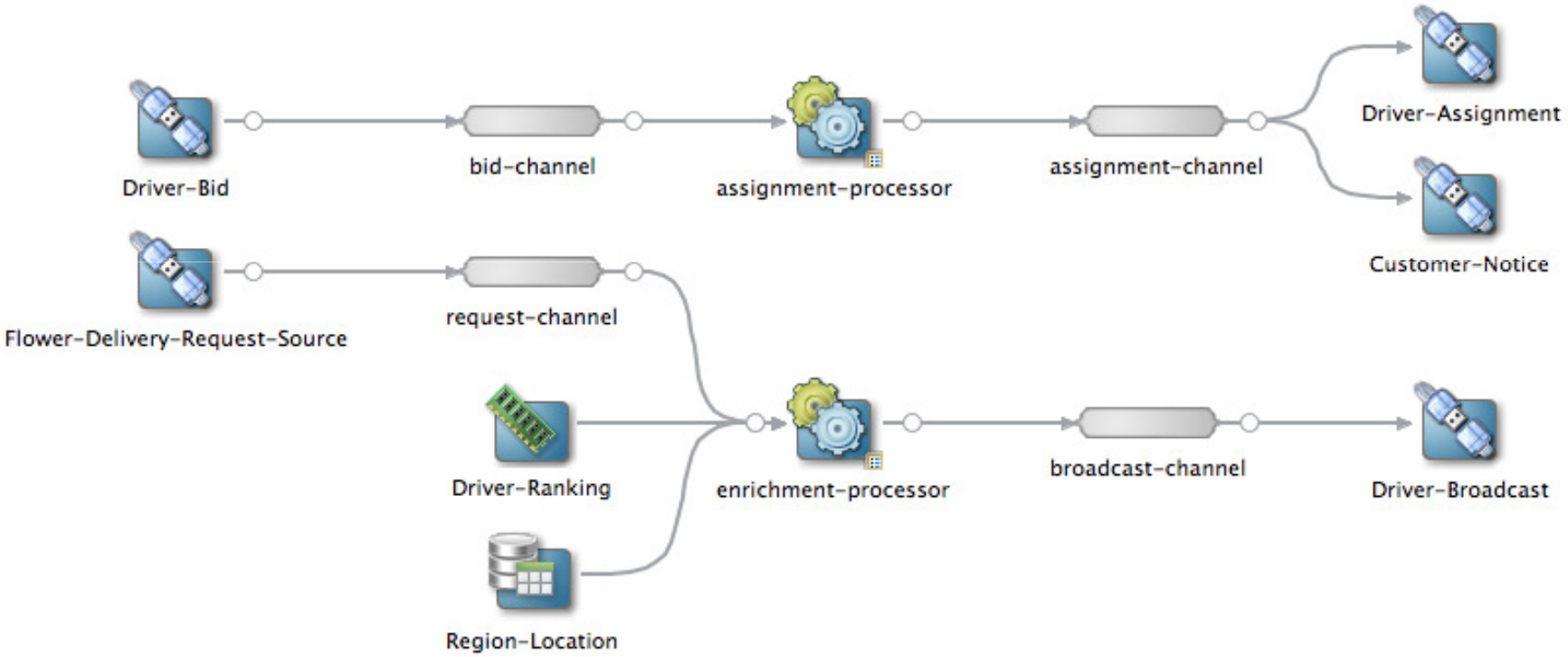
Functional View



Deployment View



Fast-Flower-Delivery Use-case



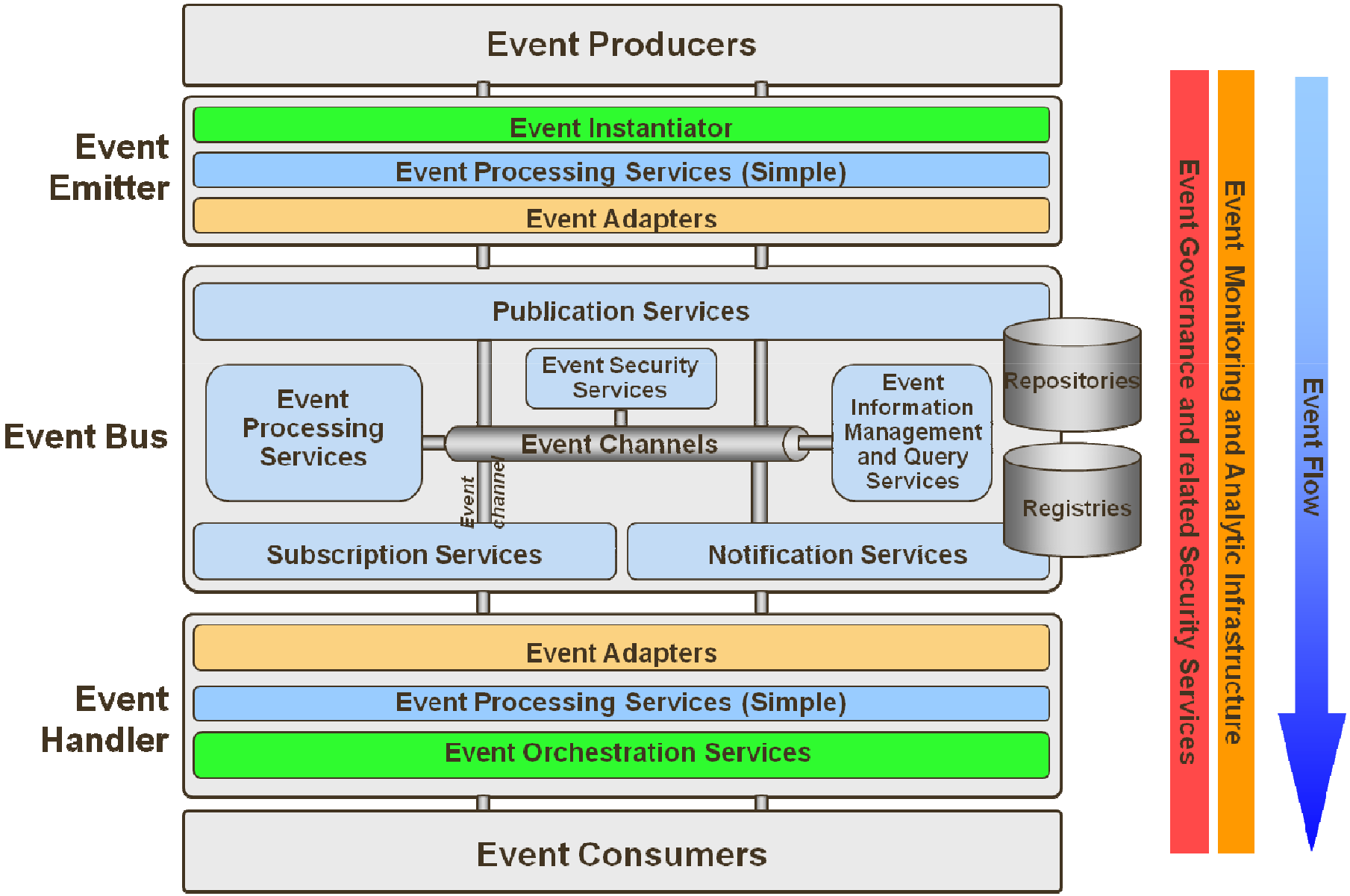
epts
event processing technical society

DEBS2010
4th ACM International Conference on Distributed Event-Based Systems

Overview – Sample Member Architectures

- University of Trento - Themis Palpanas
- TIBCO – Paul Vincent
- Oracle - Alex Alves
- **IBM - Catherine Moxey**

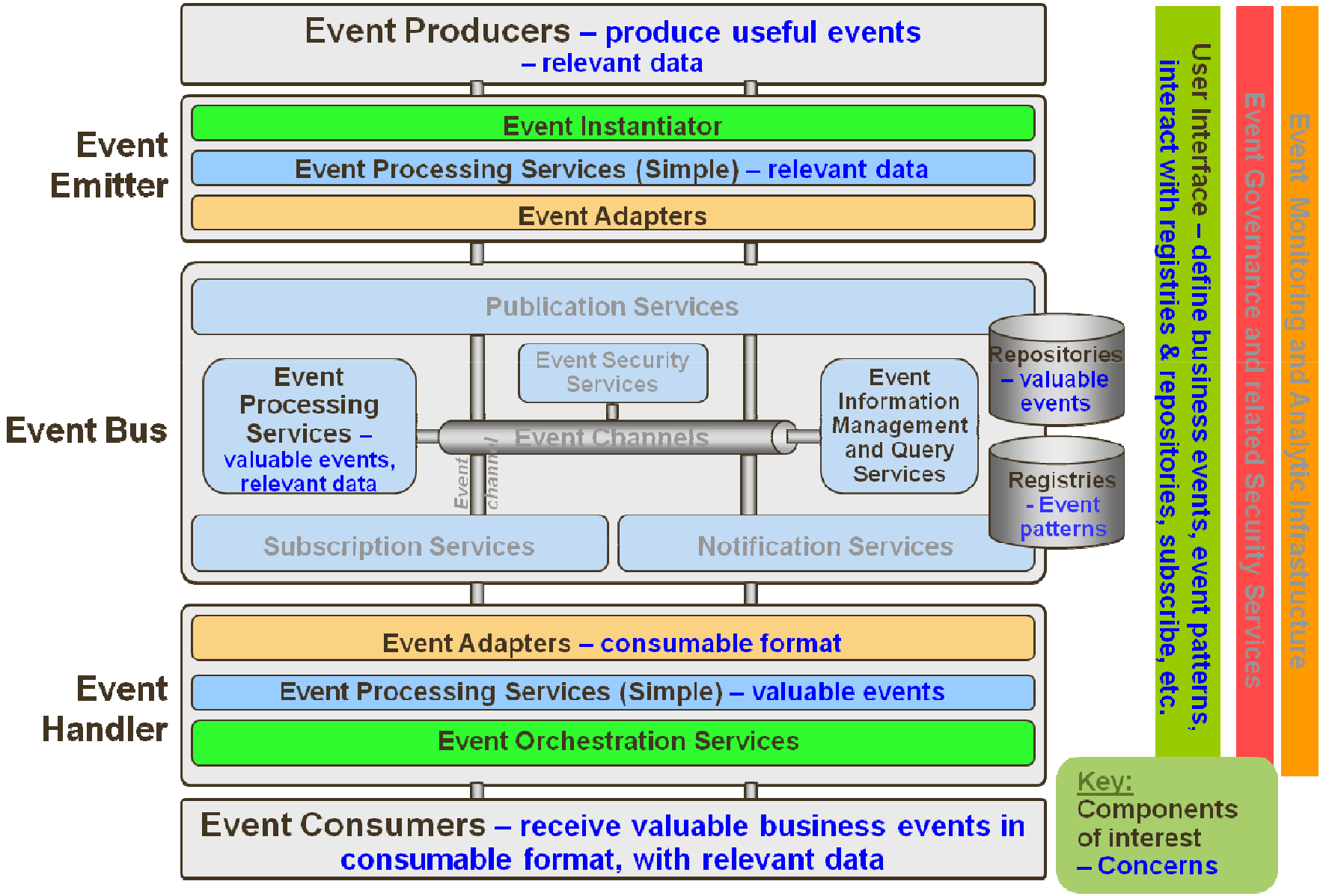
IBM Conceptual Architecture for EP



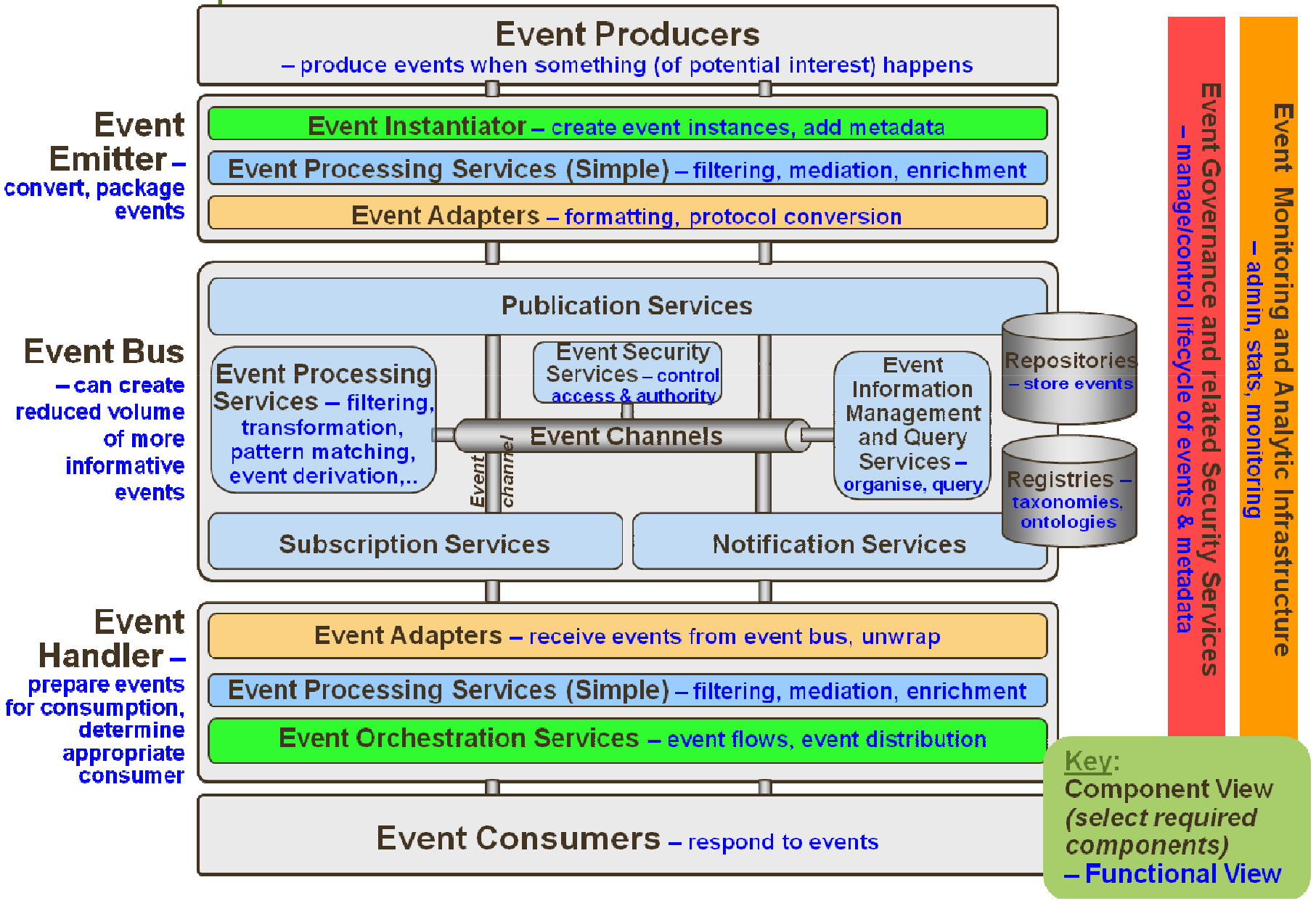
IBM EP Architecture – mapping to methodology

System:	event processing system – event emitter, event bus, event handler	
Environment:	initial producers of events and their ultimate consumers	
Stakeholders	Concerns	Architectural View
Event consumers – use events to get information, take action, detect problems, etc.	Receive correct events in consumable format, with relevant data	Event Handler (adapters, consumer-side EP services, orchestration), Event Processing Services, Repositories, Event Emitters. Publication, Subscription, Notification services to receive events
Event "Analysts" – create and understand event specifications and definitions, how events should be processed, etc.	Means to specify the events to be produced, the patterns to be identified, event enrichment etc.	Event Instantiator, Event Processing Services, Event Information Management and Query Services, Event Security Services, Event Registries. Event Repositories to do pattern-matching over time
Event System managers – manage and control the EP system	Security, Performance, Configurability	Event Security Services, Event Governance and related Security Services
Event System operators – operate the EP system	Abilities to monitor and configure system, diagnose and solve problems	Event Monitoring and Analytic Infrastructure, Event Channels, Publication, Subscription, Notification Services
Event Producers	Generated events are made use of in some way	Event Instantiator, producer-side EP services, Event Adapters, Event Bus…

IBM Conceptual Architecture for *Decision Maker* stakeholder



IBM Conceptual Architecture for *Architect* stakeholder

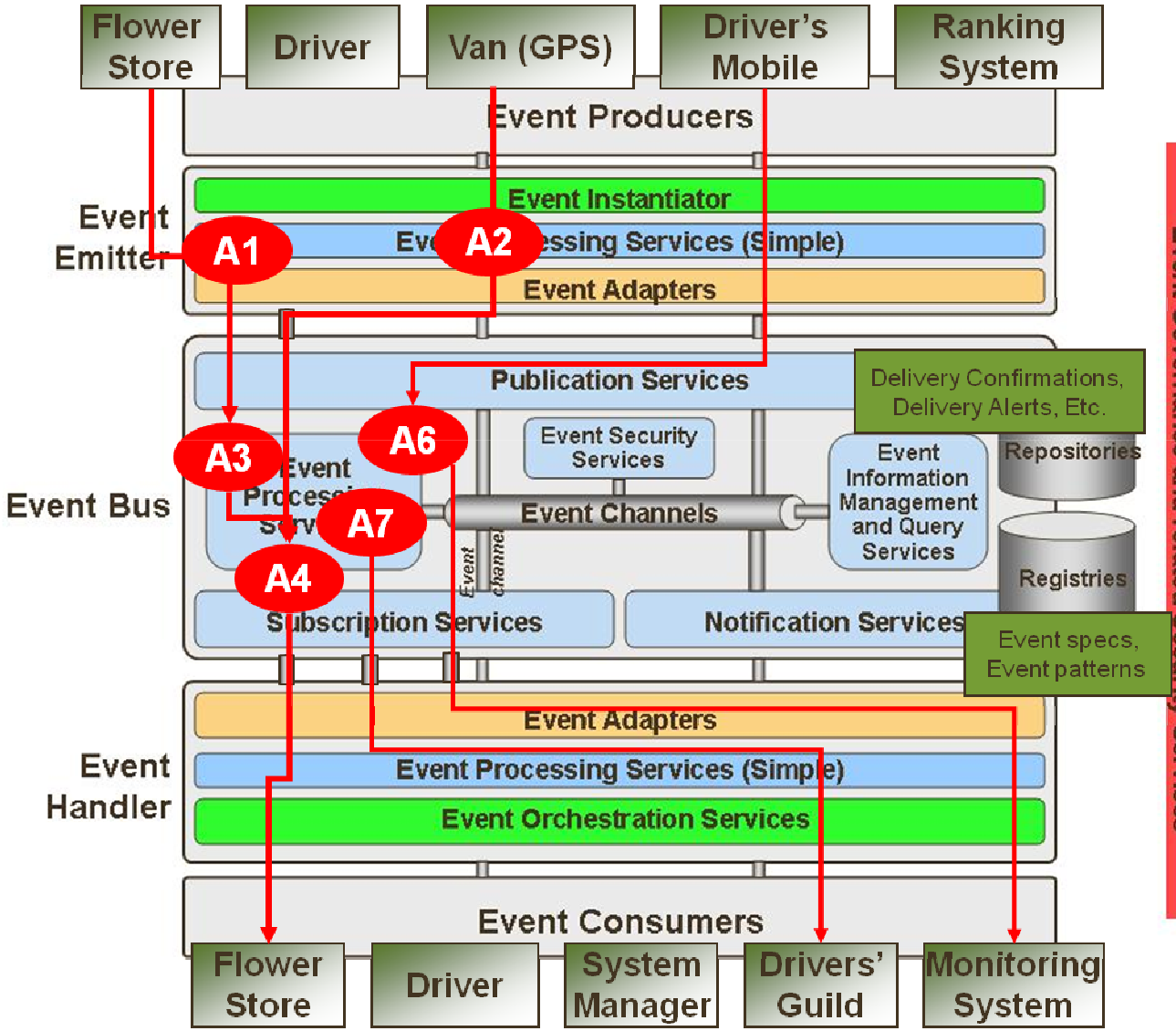


epts
event processing technical society

DEBS2010
4th ACM International Conference on Distributed Event-Based Systems

FFD Use Case mapped to IBM RA

Showing some of the Event Flows and Event Processing Agents



A1: Agent Property	Specification
Agent Name	Enrich Delivery Request
Agent Type	Enrichment
Agent Context	Always, By flower store
Input Event(s)	Delivery Request
Agent Specification	Flower store's minimum acceptable ranking
Output Event(s)	Enriched Delivery Request

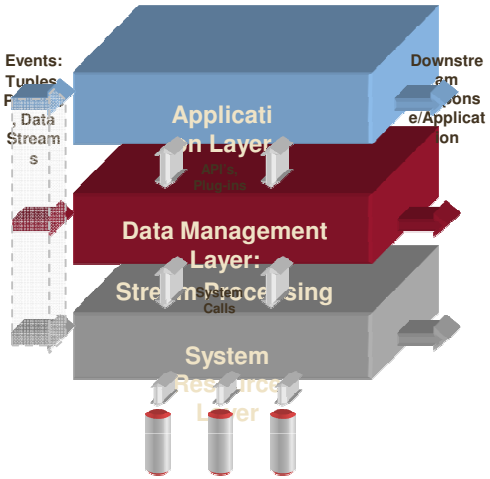
Agent ID	Agent Name
A1	Enrich Delivery Event
A2	Translate Location (from GPS to region of city)
A3	Filter Authorized drivers (who meet ranking)
A4(m)	Assign (manual)
A4(a)	Assign (automatic)
A5	Generate pick-up alert
A6	Generate Delivery alert
A7	Adjust Ranking

A7: Agent Property	Specification
Agent Name	Adjust Ranking
Agent Type	Pattern
Agent Context	By Driver
Input Event(s)	Last 20 deliveries, Delivery Alerts
Agent Specification	No alerts, increase ranking; more than 5, decrease ranking
Output Event(s)	Ranking Increase, Ranking Decrease

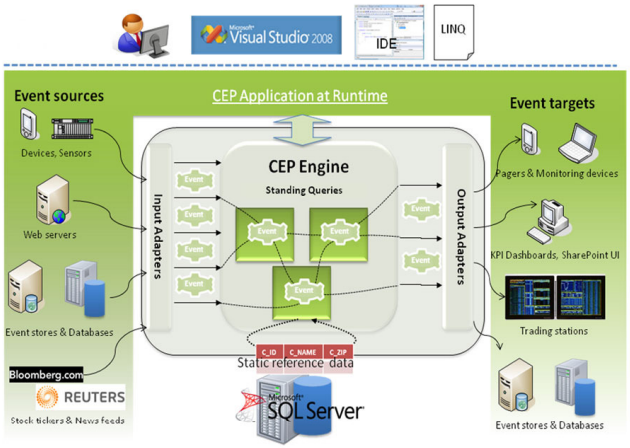
Agenda

- Introduction to architectures, architecture methodologies, and event processing
- Member architectures and salient features
- **Skeleton reference architecture from EPTS Reference Architecture Working Group**
- Summary and future work of the EPTS Reference Architecture Working Group

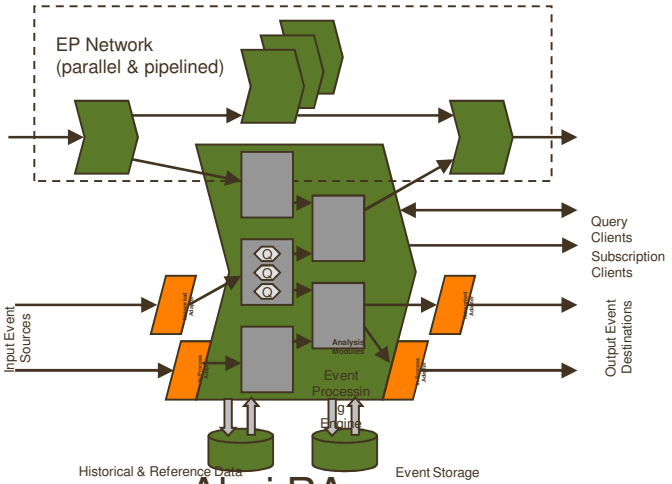
Step 1: collect Event Processing Reference Architectures…



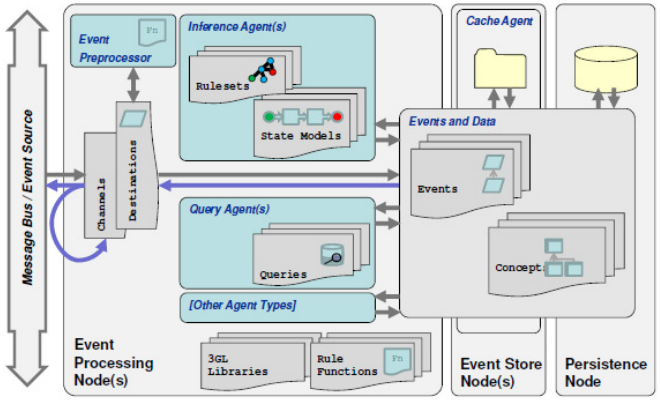
Streambase RA



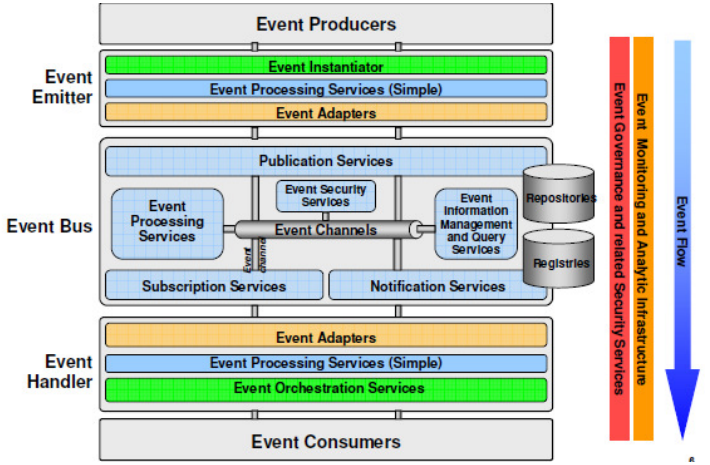
Microsoft RA



Aleri RA

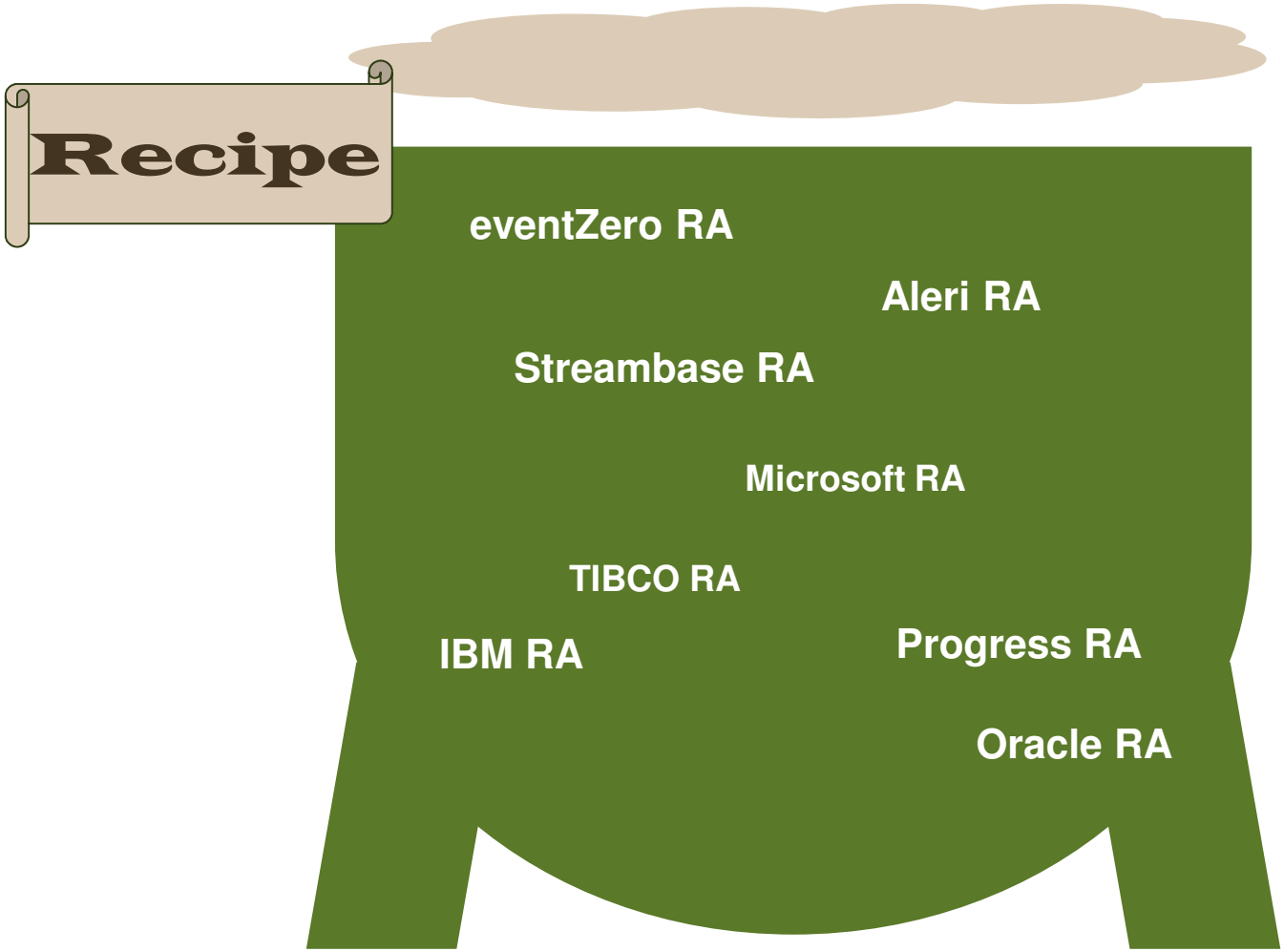


TIBCO RA

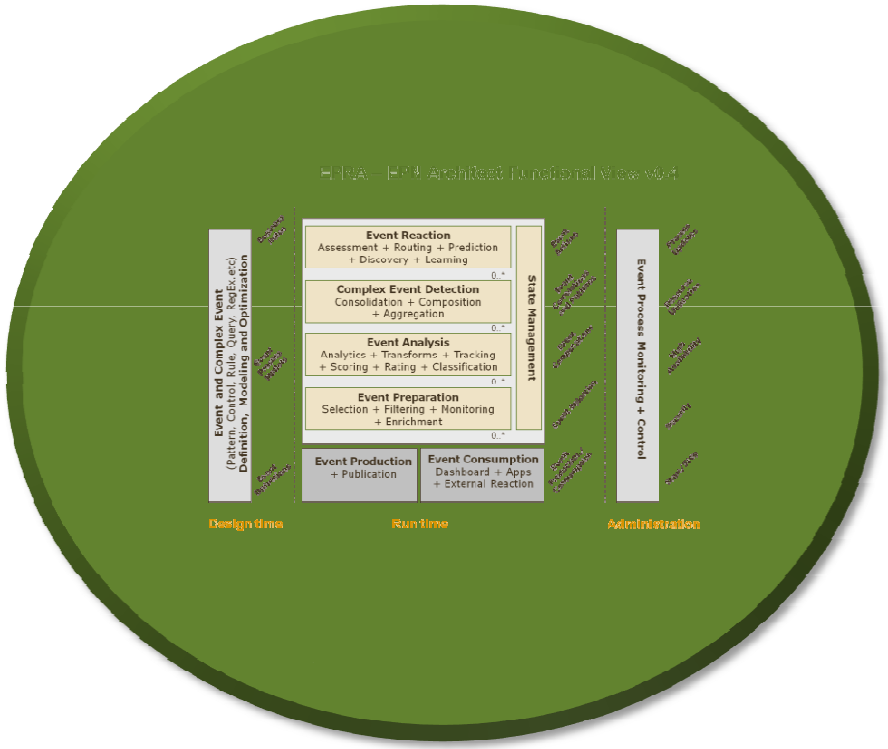


IBM RA

Step 2: identify methodology to abstract an EPTS version…



Step 3: distil the common EPTS Reference Architecture(s)



Reference Architecture Viewpoints recap

Viewpoint Element	**Viewpoint**		
	Engineering EP Architecture	*Managing EP Architecture*	*Business with EP Architecture*
Concepts	How to implement?	How to apply?	How to utilize / sell / own?
Stakeholders	Architects / Engineers	Project Manager	Decision Maker, Customer, Provider
Concerns	Effective construction and deployment	Operational Management	Strategic and tactical management
Techniques / Languages	Modeling, Engineering	IT (service/appl) management, project management	Monitoring, Enterprise Decision Management, Governance

Stakeholders and Viewpoints covered

	Viewpoints (for CEP / stateful EP)	
Stakeholder	*Component*	*Functional*
Decision Maker / End User	[Not applicable]	Inputs, Outputs and Processing Requirements View
Architect / Designer	Solution Components View	Functions carried out in CEP View

Stakeholders and Viewpoints covered

	Viewpoints (for CEP / stateful EP)	
Stakeholder	*Component*	*Functional*
Decision Maker / End User	[Not applicable]	Inputs, Outputs and Processing Requirements View
Architect / Designer	Solution Components View	Functions carried out in CEP View

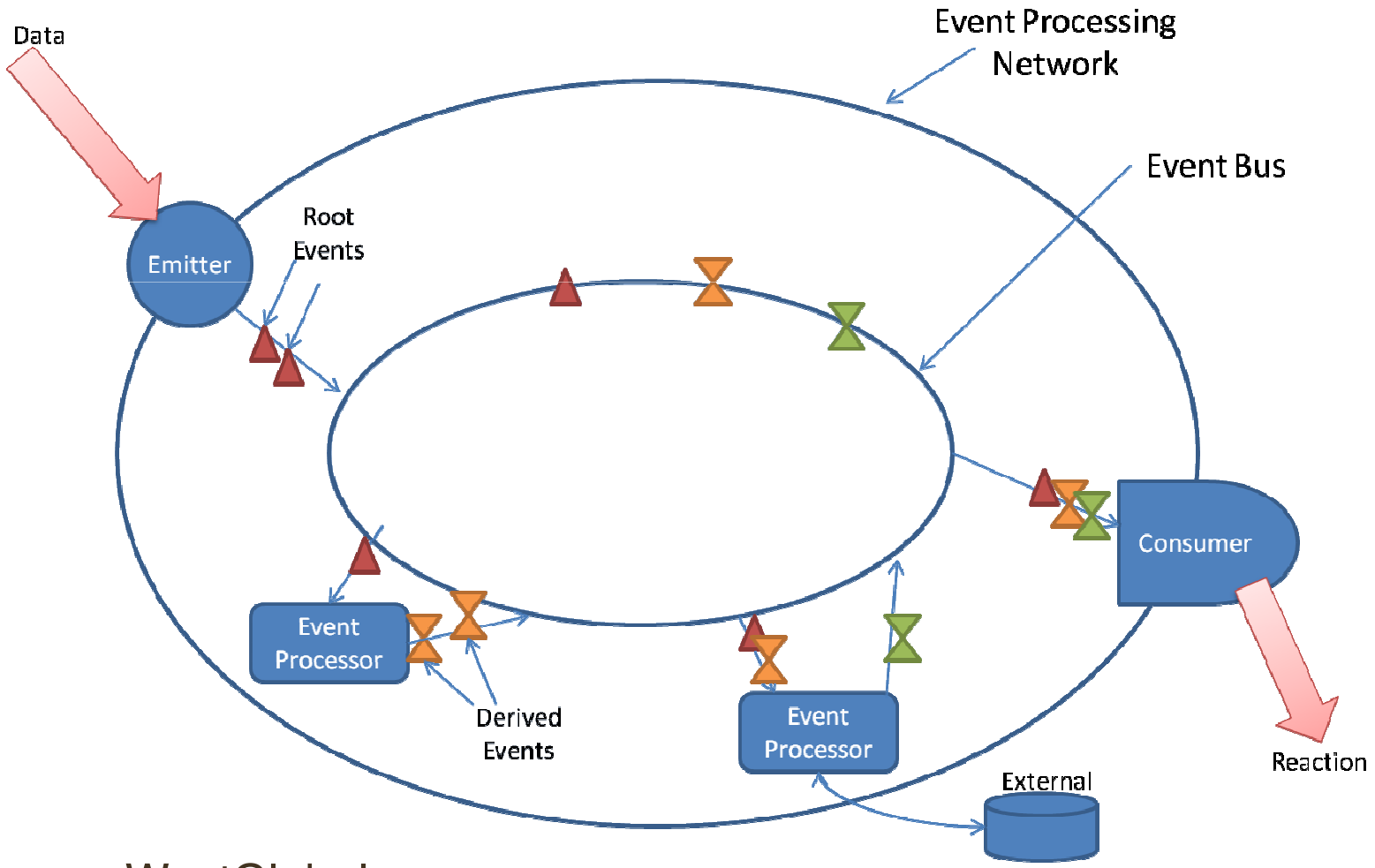
epts
event processing technical society

DEBS2010
4th ACM International Conference on Distributed Event-Based Systems

Logical View

- View of the Event Processing Network?
- A Network Diagram?
- *Currently under consideration!*

Example Logical View



Source: WestGlobal

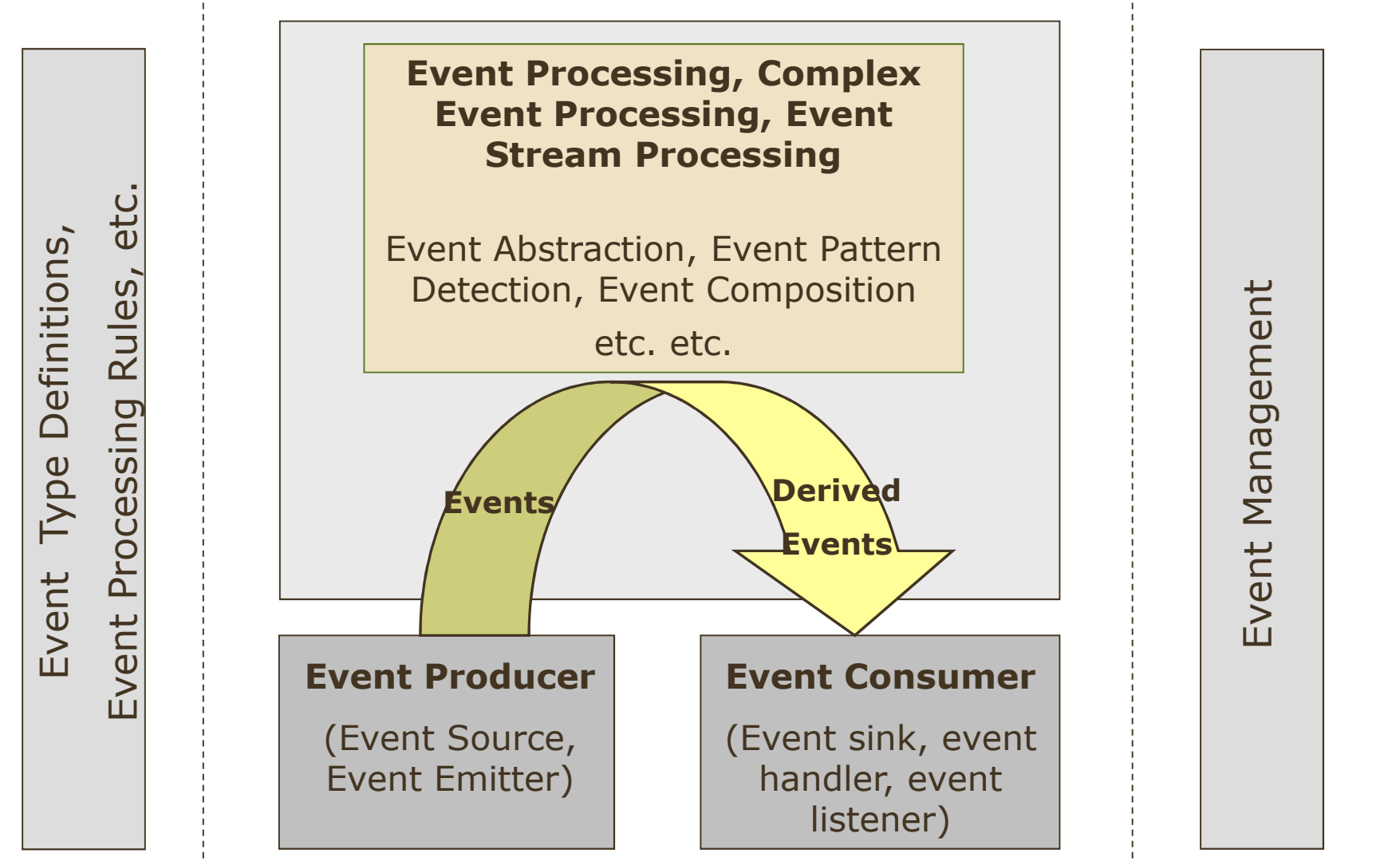
Stakeholders and Viewpoints covered

	Viewpoints (for CEP / stateful EP)	
Stakeholder	*Component*	*Functional*
Decision Maker / End User	[Not applicable]	Inputs, Outputs and Processing Requirements View
Architect / Designer	Solution Components View	Functions carried out in CEP View

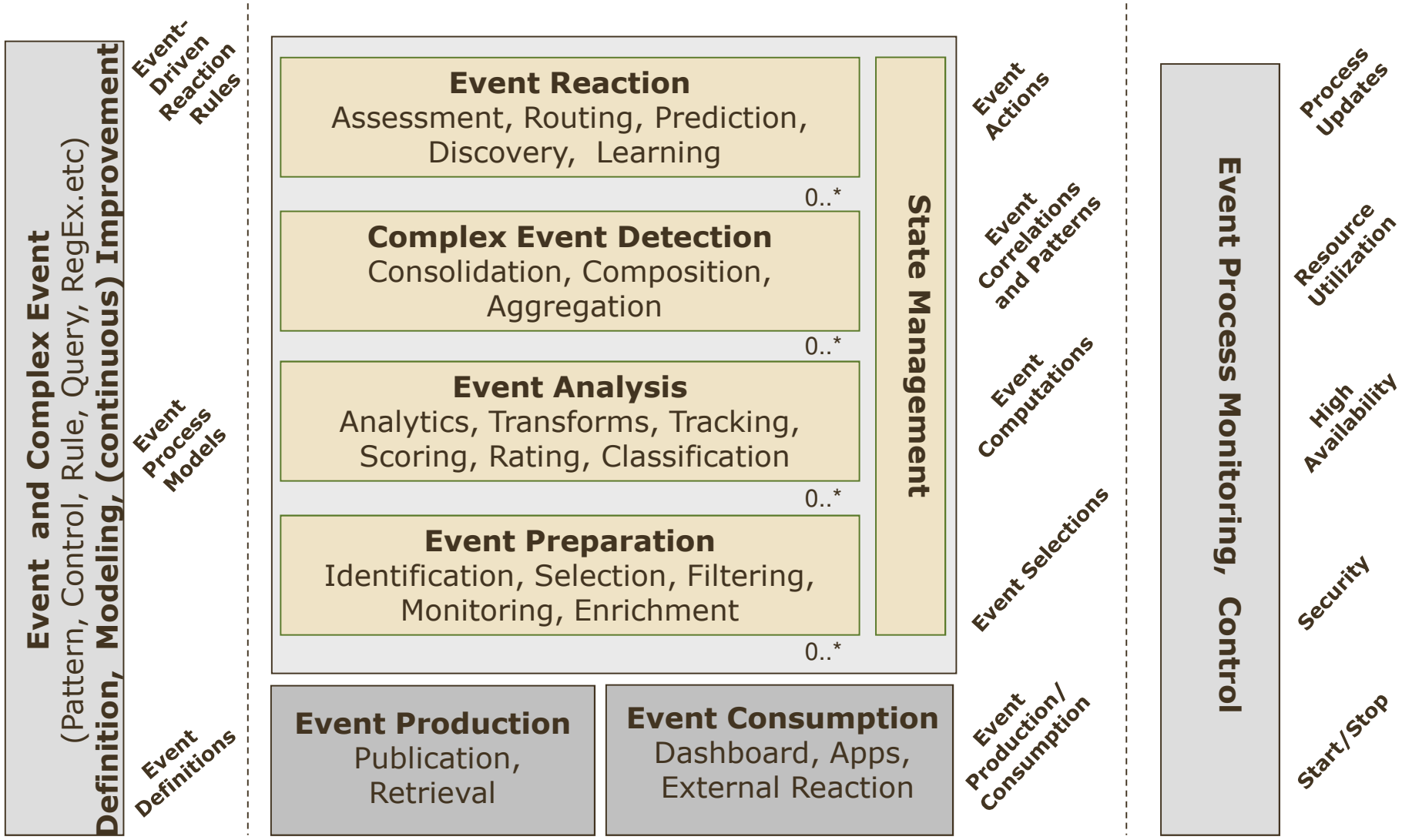
Functional View

- **Architect and Developer perspective**
  - includes the 3 main functions (development, run-time and administration),
  - targets primarily the automated event processing operations
- **Run-time functions in 2 main groups:**
  - the event infrastructure (sources and consumers) external to the event processor under consideration,
  - the event processor.

Functional View source: definitions of EP

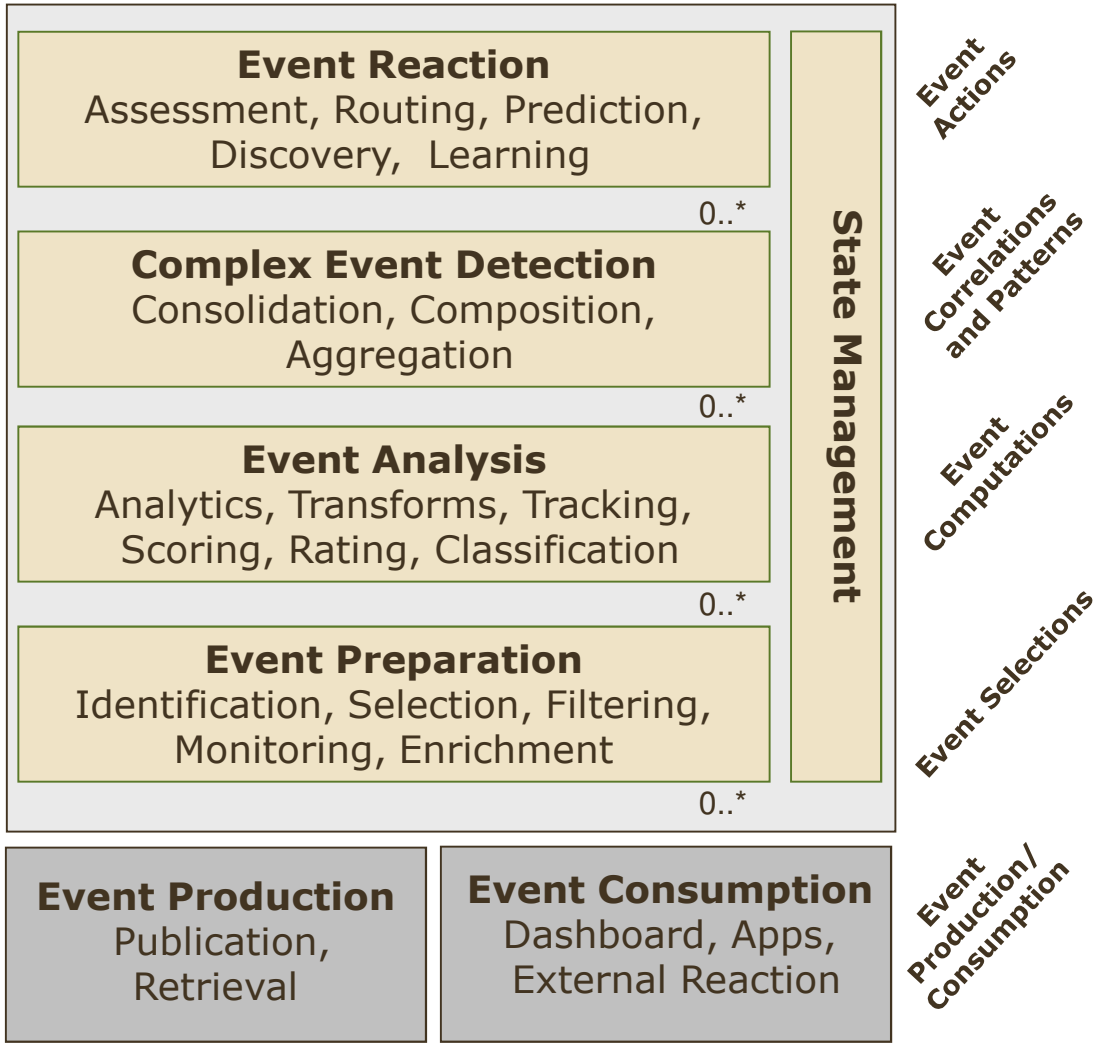


Reference Architecture: Functional View



Design time **Run time** **Administration**

Reference Architecture: Functional View / Runtime



Run time

Reference Architecture: Functional View / Runtime

Event Production: the source of events for event processing.

- Event Publication: As a part of event production, events may be published onto a communication mechanism (eg event bus) for use by event consumers (including participants in event processing). This is analagous to a "push" system for obtaining events.
- Event Retrieval: As a part of event production, events may be explicitly retrieved from some detection system. This is analagous to a "pull" system for obtaining events.

Reference Architecture: Functional View / Runtime

Event Consumption: the process of using events from event publication and processing. Event processing itself can be an event consumer, although for the purposes of the reference architecture, event consumers are meant to indicate downstream consumers of events generated in event processing.

- Dashboard: a type of event consumer that displays events as they occur to some user community.
- Applications: a type of event consumer if it consumes events for its own processes.
- External Reaction: caused through some event consumption, as the result of some hardware or software process.

Reference Architecture: Functional View / Runtime

Event Preparation: the process of preparing the event and associated payload and metadata for further stages of event processing.

- **Entity Identification**: incoming events will need to be identified relative to prior events, such as associating events with particular sources or sensors.
- **Event Selection**: particular events may be selected for further analysis. Different parts of event processing may require different selections of events. See also event filtering.
- **Event Filtering**: a stream or list of events may be filtered on some payload or metadata information such that some subset is selected for further processing.
- **Event Monitoring**: particular types of events may be monitored for selection for further processing. This may utilise specific mechanisms external to the event processing such as exploiting event production features.
- **Event Enrichment**: events may be "enriched" through knowledge gained through previous events or data.

Event Preparation
Identification, Selection, Filtering, Monitoring, Enrichment

Reference Architecture: Functional View / Runtime

Event Analysis: the process of analysing suitably prepared events and their payloads and metadata for useful information.

- **Event Analytics:** the use of statistical methods to derive additional information about an event or set of events.
- **Event Transforms:** processes carried out on event payloads or data, either related to event preparation, analysis or processing.
- **Event Tracking:** where events related to some entity are used to identify state changes in that entity.
- **Event Scoring:** the process by which events are ranked using a score, usually as a part of a statistical analysis of a set of events. See also *Event Analytics*
- **Event Rating:** where events are compared to others to associate some importance or other, possibly relative, measurement to the event.
- **Event Classification:** where events are associated with some classification scheme for use in downstream processing.

Event Analysis
Analytics, Transforms, Tracking, Scoring, Rating, Classification

Reference Architecture: Functional View / Runtime

Complex Event Detection: the process by which event analysis results in the creation of new event information, or the update of existing complex events.

- **Event Consolidation:** combining disparate events together into a "main" or "primary" event. See also event aggregation.
- **Event Composition:** composing new, complex events from existing, possibly source, events.
- **Event Aggregation:** combining events to provide new or useful information, such as trend information and event statistics. Similar to event consolidation.

Complex Event Detection
Consolidation, Composition, Aggregation

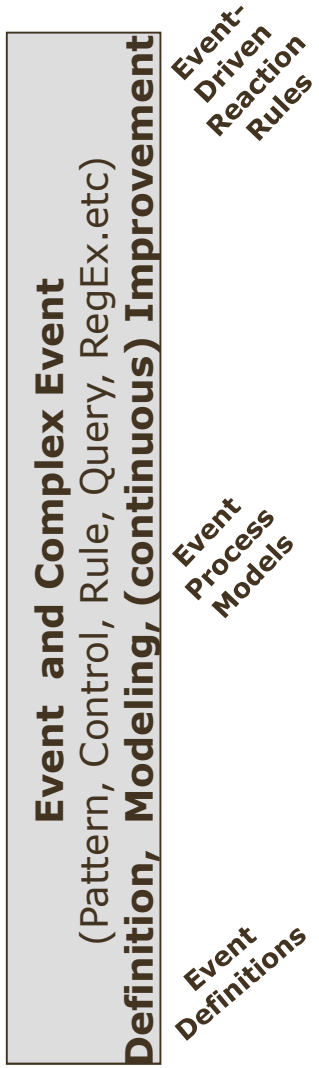
Reference Architecture: Functional View / Runtime

Event Reaction: the process subsequent to event analysis and complex event detection to handle the results of analysis and detection.

- **Event Assessment:** the process by which an event is assessed for inclusion in some process, incorporation in some other event, etc.
- **Event Routing:** the process by which an event is redirected to some process, computation element, or other event sink.
- **Event Prediction:** where the reaction to some event processing is that some new event is predicted to occur.
- **Event Discovery:** where the reaction to some event processing is the disclosure of a new, typically complex, event type.
  - Note that event prediction is predicting some future event, usually of a known type, whereas event discovery is the uncovering of a new event type. See also event-based learning.
- **Event-based Learning:** the reaction to some event processing that uses new event information to add to some, typically statistical-based, understanding of events.
  - Note that event-based learning is a specialisation of general machine learning and predictive analytics.

Event Reaction
Assessment, Routing, Prediction, Discovery, Learning

Reference Architecture: Functional View / Design time



Covers the definition, modeling, improvement / maintenance of the artifacts
used in event processing:

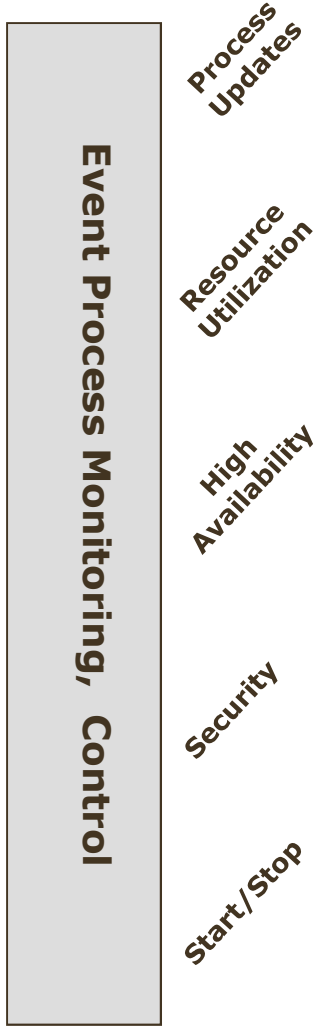
- event definitions, including event metadata and payloads,
- event and event object organisations and structures,
- event processing transformations / queries / rules / procedures / flows / states / decisions / expressions (although these can sometimes be considered as administrative updates in some situations)

Design time

Reference Architecture: Functional View / Administration

Administrative concepts of monitoring and control. This may involve

- starting and stopping the application and event processing elements, including application monitors
- providing and updating security levels to event inputs and outputs (also can design-time)
- management of high availability and reliability resources, such as hot standby processes
- resource utilisation monitoring of the event processing components
- process updates, such as how-swapping of event processing definitions to newer versions.



Agenda

- Introduction to architectures, architecture methodologies, and event processing
- Member architectures and salient features
- Skeleton reference architecture from EPTS Reference Architecture Working Group
- **Summary and future work of the EPTS Reference Architecture Working Group**

Current Works

- **Terminology and Methodology**
  - ANSI/IEEE Std 1471 :: ISO/IEC 42010 Terminology
  - EPTS-RA Terminology
  - Concepts from the EPTS Glossary
- **Reference Architecture Discovery**
- **Reference Architecture Comparison**
  - Identify commonalities in proposed architectures
  - Collect core and additional CEP system functions and components
- **Member Reference Architecture Descriptions**
  - Using the EPTS-RA Methodology and Terminology
- **Application of RA Descriptions to Use Cases**

Next Steps

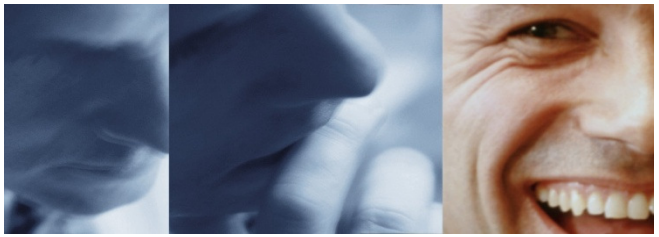
- **Generalize the EPTS Reference Architecture descriptions**
  - addressing important stakeholders and their views
  - using rigorous RA methodology and terminology (+ glossary)
- **Evaluate the EPTS-RA descriptions**
  - apply on selected EPTS use cases and compare to member experience
  - compare with Best Practice Guidelines
- **Review Logical Architecture requirements for Reference purposes**
- **Possible Outputs:**
  - **EPTS-RA Description** document
  - **EPTS-RA Design Patterns and Best Practice Guidelines** document
  - **Wiley Book** "Pattern Oriented Software Architecture: Architectures, Models and Patterns for Event Processing" (already in preparation)
  - Input for **Event Processing Metamodels and associated standards**

Summary

- **Reference Architecture provides a common set of EP Functions that may be included in**
  - EP systems
  - EP-related tools
- **Describes the mapping from Glossary of Terms to Implementation**
- **Provides the basis for EP agents, operators and languages**

epts
event processing technical society

DEBS2010
4th ACM International Conference on Distributed Event-Based Systems



Thank you !

Acknowledgment to the Event Processing Technical Society Reference Architecture working group: David Tucker (Event Zero), John Morrell (Coral8, Inc.), Baden Hughes (Event Zero), Dieter Gawlick (Oracle), Alex Alves (Oracle), Anand Srinivasan (Oracle), Shailendra Mishra (Oracle), Zbigniew Jerzak (TU Dresden), Hans-Arno Jacobsen (University Toronto), Albert Mavashev (Nastel Technologies Inc.), Simon Courtenage (University of Westminster), Catherine Moxey (IBM United Kingdom Limited), Richard Tibbetts (Streambase), Brian Connell (WestGlobal), Guy Sharon (IBM), Themis Palpanas (University of Trento)

References

- EPTS Glossary
  http://www.ep-ts.com/component/option,com_docman/task,doc_download/gid,66/Itemid,84/
- University of Trento Architecture section:
  1. Heiko Stoermer, Themis Palpanas, George Giannakopoulos. *The Entity Name System: Enabling the Web of Entities.* International Workshop on Data Engineering meets the Semantic Web (DESWeb), in conjunction with the IEEE International Conference on Data Engineering (ICDE), Long Beach, CA, USA, March 2010
  2. Zoltan Miklos, Nicolas Bonvin, Paolo Bouquet, Michele Catasta, Daniele Cordioli, Peter Fankhauser, Julien Gaugaz, Ekaterini Ioannou, Hristo Koshutanski, Antonio Mana, Claudia Niederee, Themis Palpanas, Heiko Stoermer. *From Web Data to Entities and Back*. International Conference on Advanced Information Systems Engineering (CAiSE), Hammamet, Tunisia, June 2010
- IBM Architecture section:
  Conceptual Architecture paper: http://www.redbooks.ibm.com/redpapers/pdfs/redp4642.pdf
- TIBCO Architecture section:
  Sergio Gonik & Dan Buchko, *BusinessEvents Distributed Processing Implementation Patterns, Tips, and Techniques*, TUCON TIBCO User Conference, Las Vegas, May 2010