

Event Processing: Two Ways to Capture the Value of Events

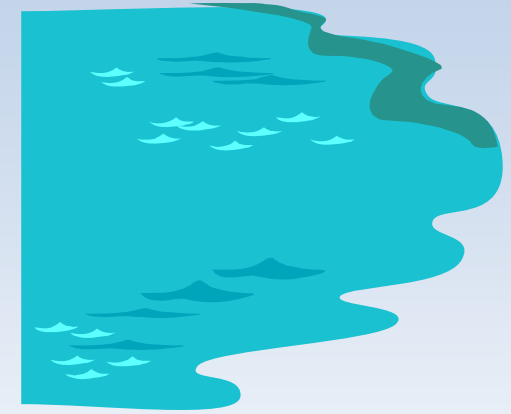
Regina Casonato

Timely Sense-and-Respond Behavior Is Essential in an Event-Driven World

Situation Awareness



Threats



Opportunities

Key Issues

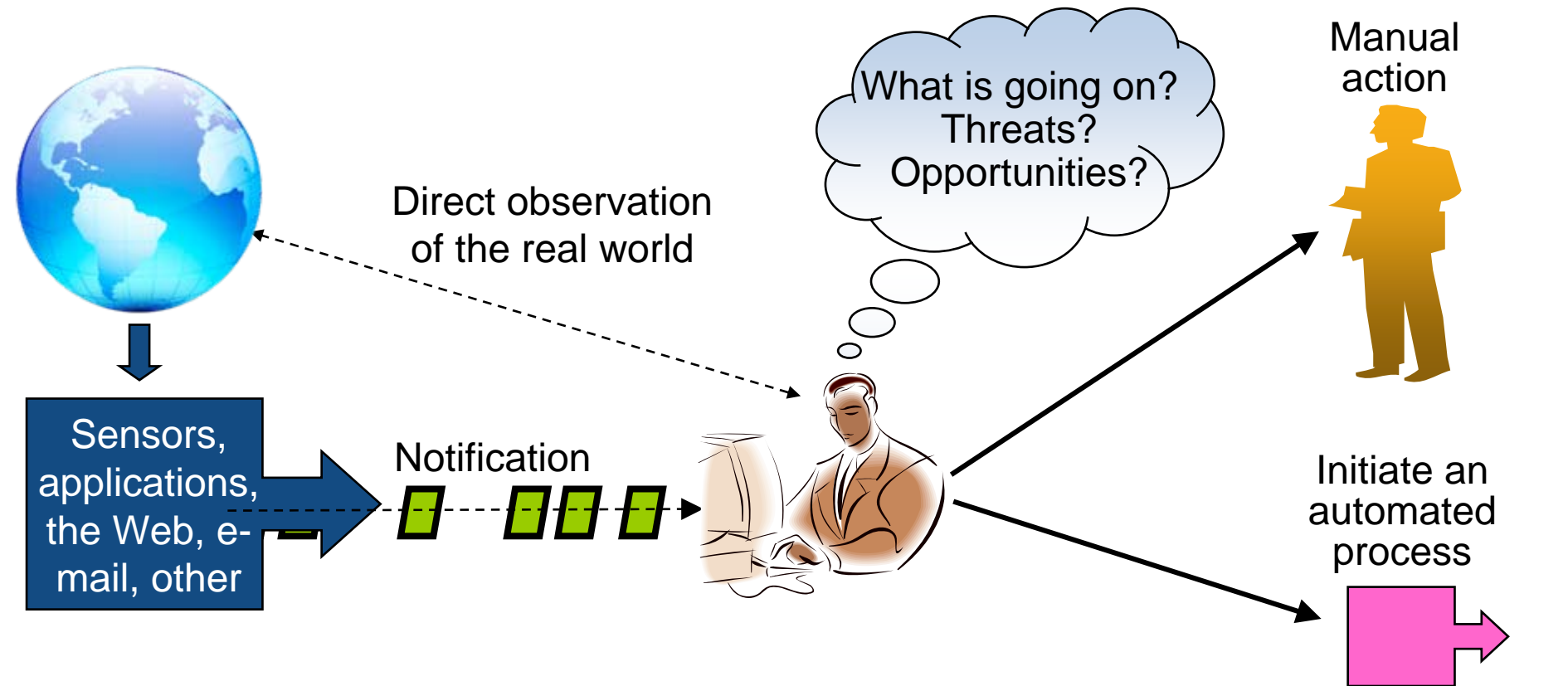
1. How will organizations derive tangible benefits from business event processing?
2. What tactics and best practices will mainstream companies employ to pursue successful event-processing strategies?

Companies Have Always Been Event-Driven but the Event Processing Was Done by People

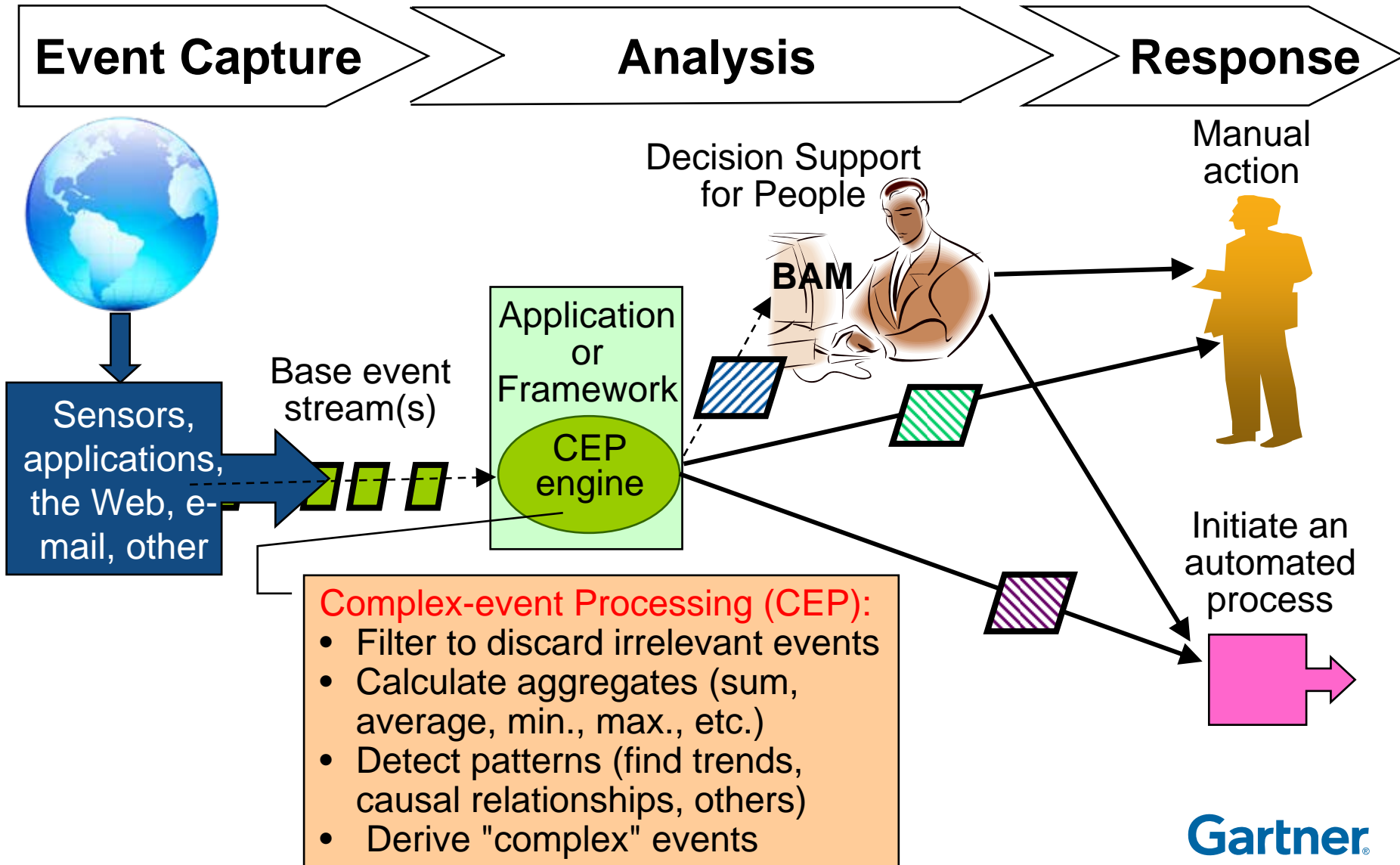
Event Capture

Analysis

Response



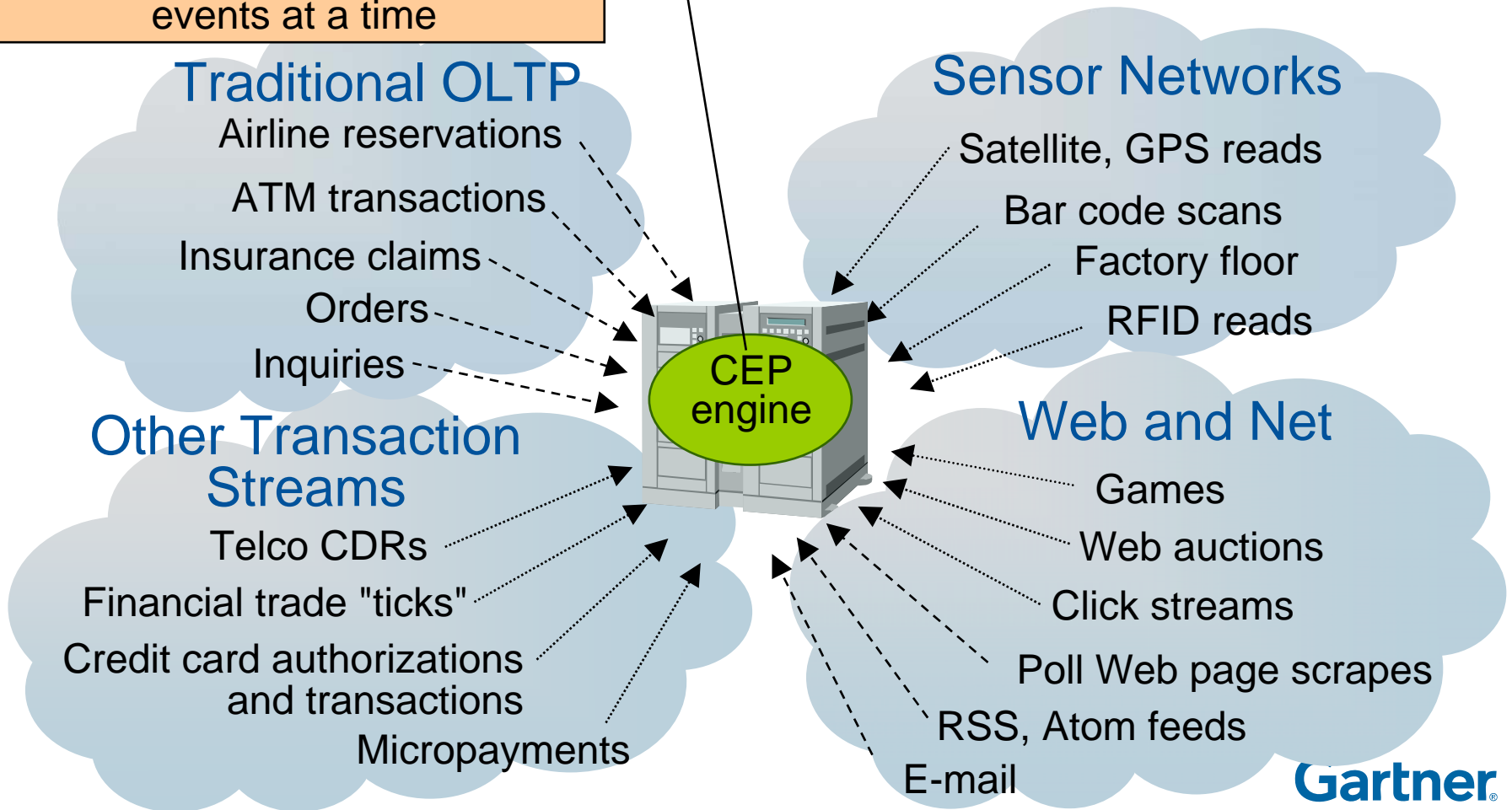
CEP Systems Automate Some Portion or All of the Analysis Phase of Event Processing



The Business Value of CEP Systems:

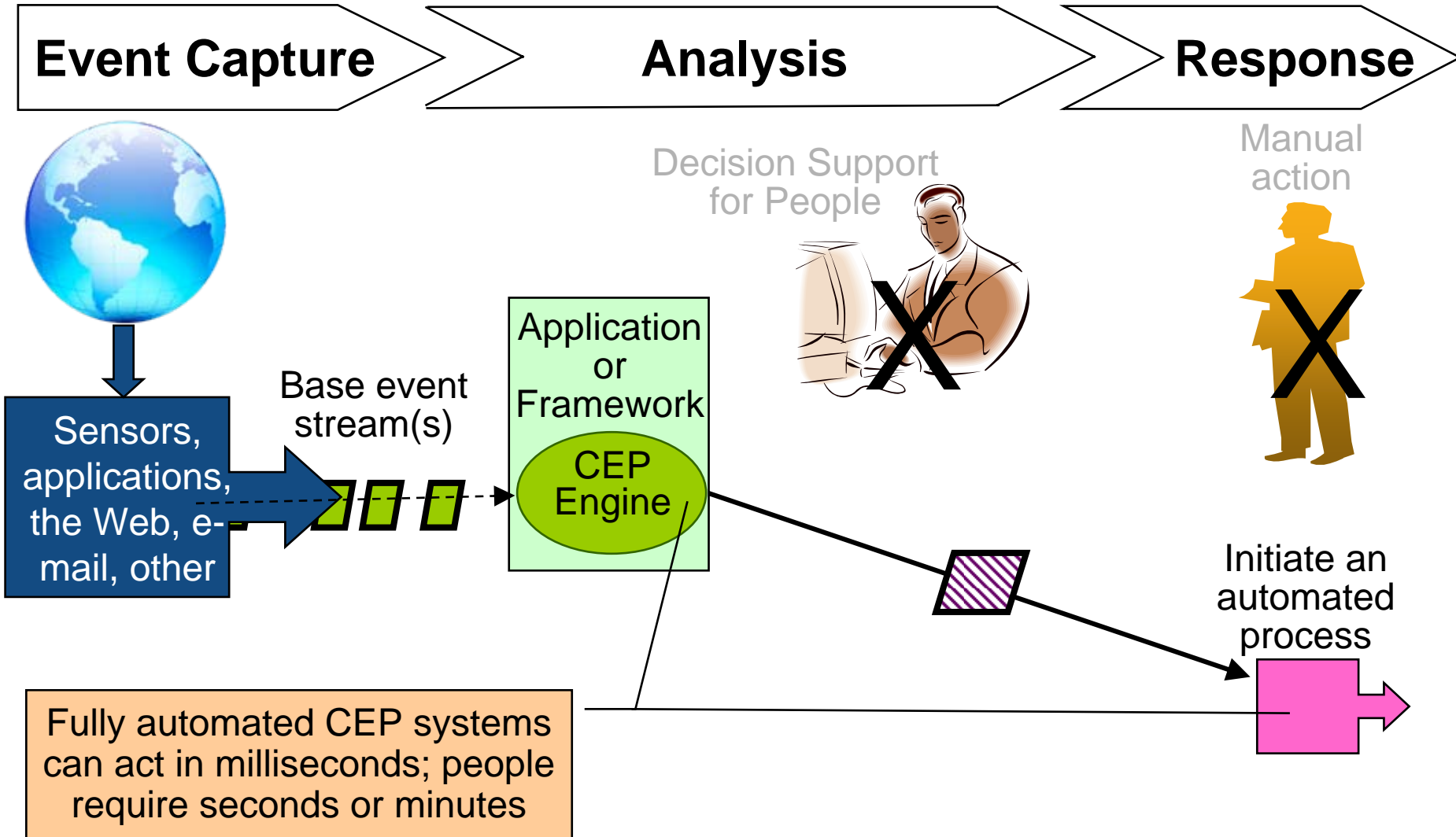
1. Better Decision Quality

CEP systems can handle up to 100,000's of events per second, people can consider only a few events at a time



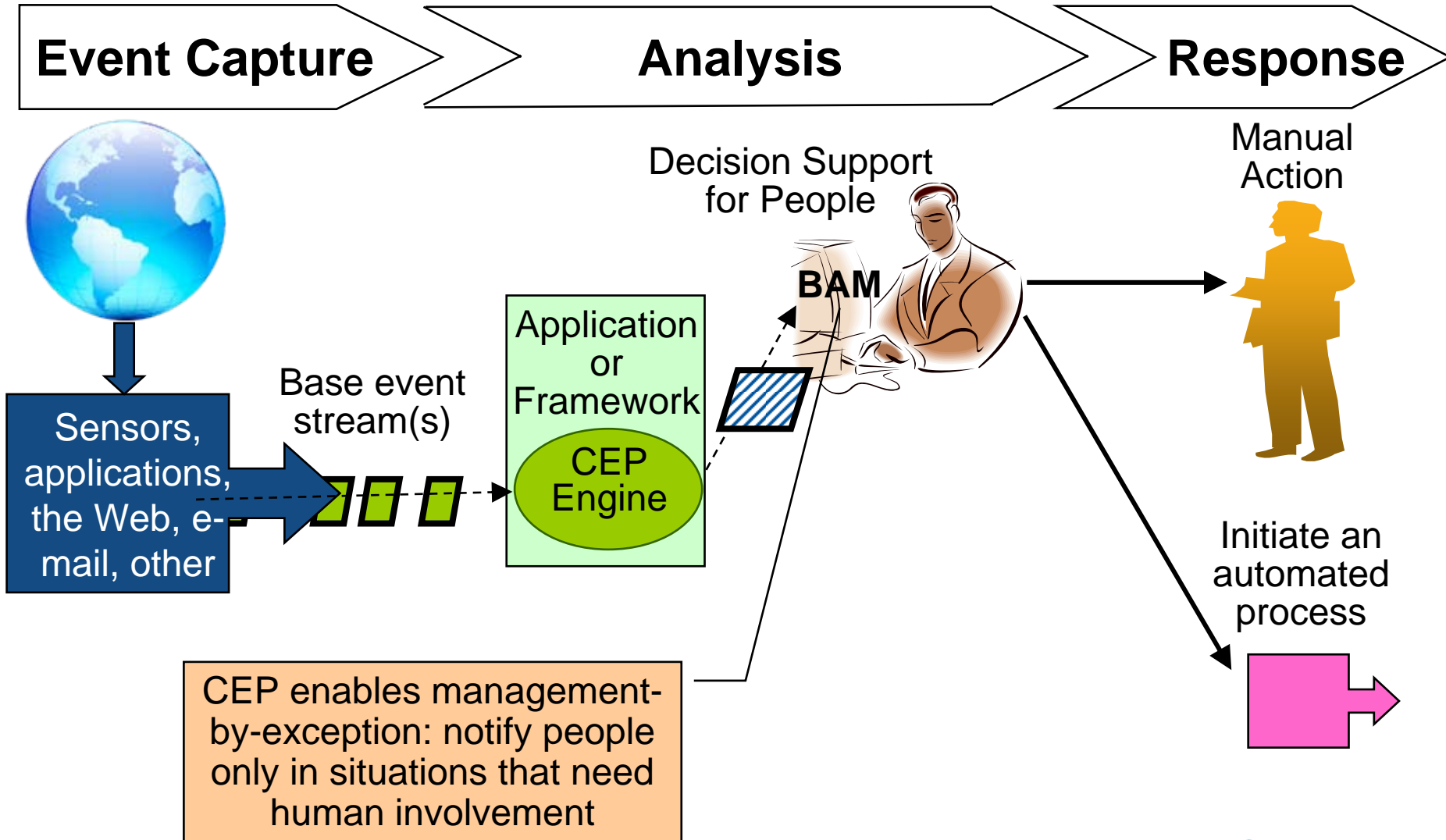
The Business Value of CEP Systems:

2. Faster Response



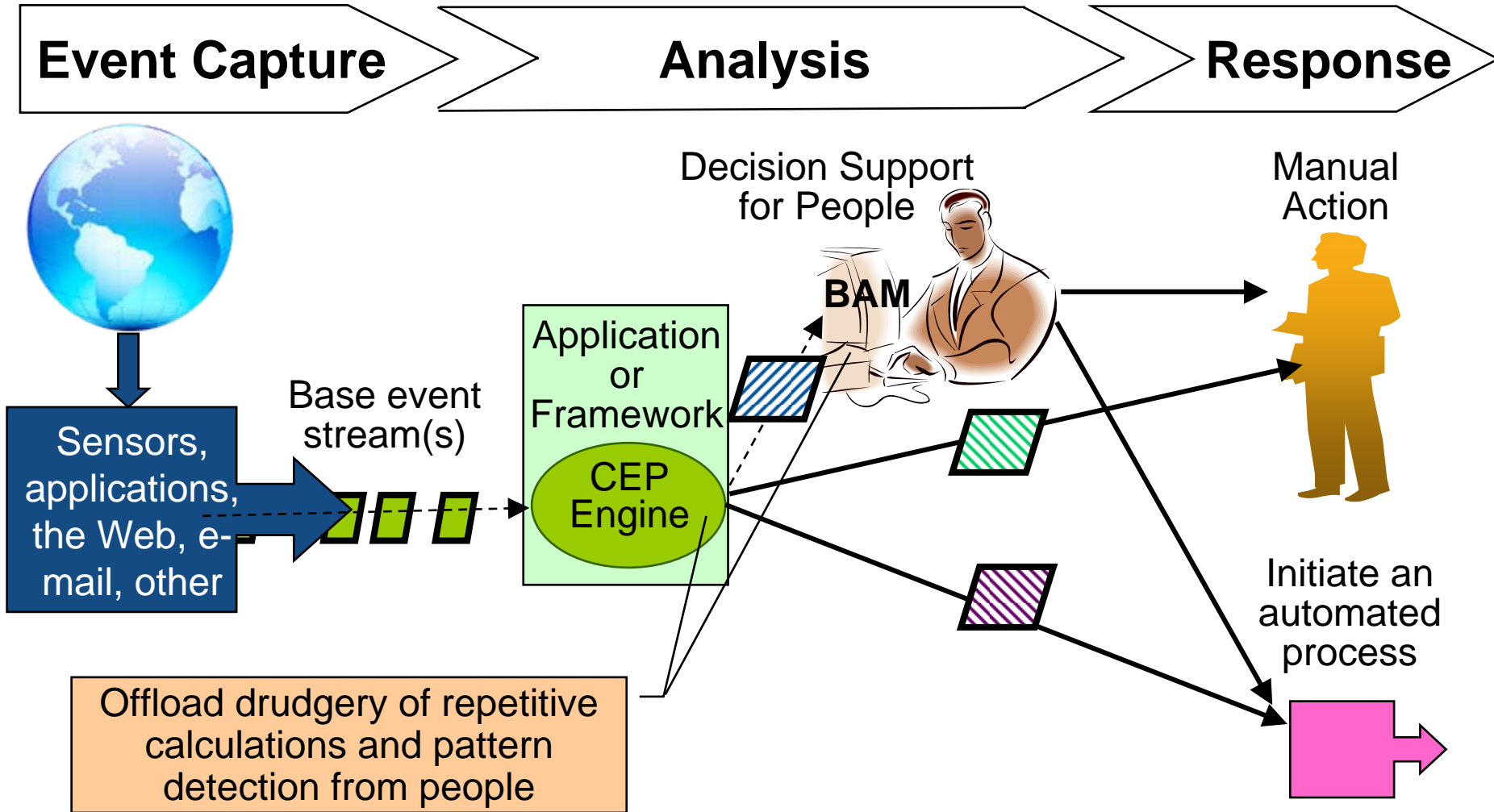
The Business Value of CEP Systems:

3. Reduce Information Overload



The Business Value of CEP Systems:

4. Reduce Cost

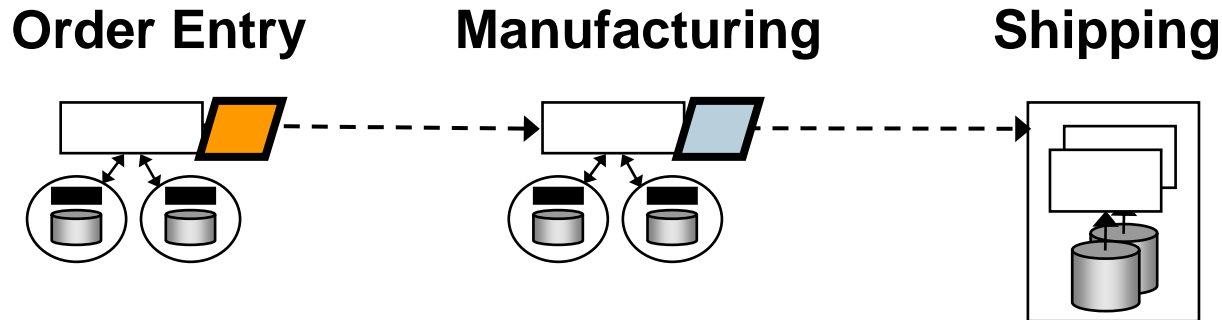


Key Issues

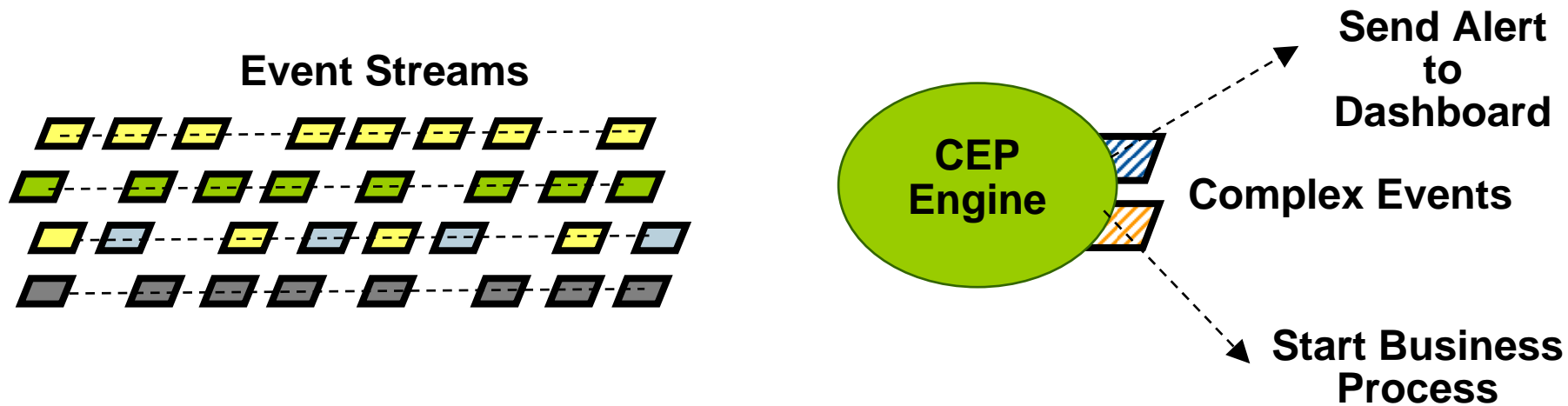
1. How will organizations derive tangible benefits from business event processing?
2. What tactics and best practices will mainstream companies employ to pursue successful event-processing strategies?

Two Aspects of Event Processing

Communication Aspect: Asynchronous Messaging for IT Flexibility



Computation Aspect: Analyzing Events for BAM and Other CEP



Event-Driven Architecture Is a Natural Variation of SOA

Client/server SOA is paving the way for EDA because:

- SOA teaches architects and developers about distributed computing
- SOA makes event-capable middleware, such as ESBs, commonplace

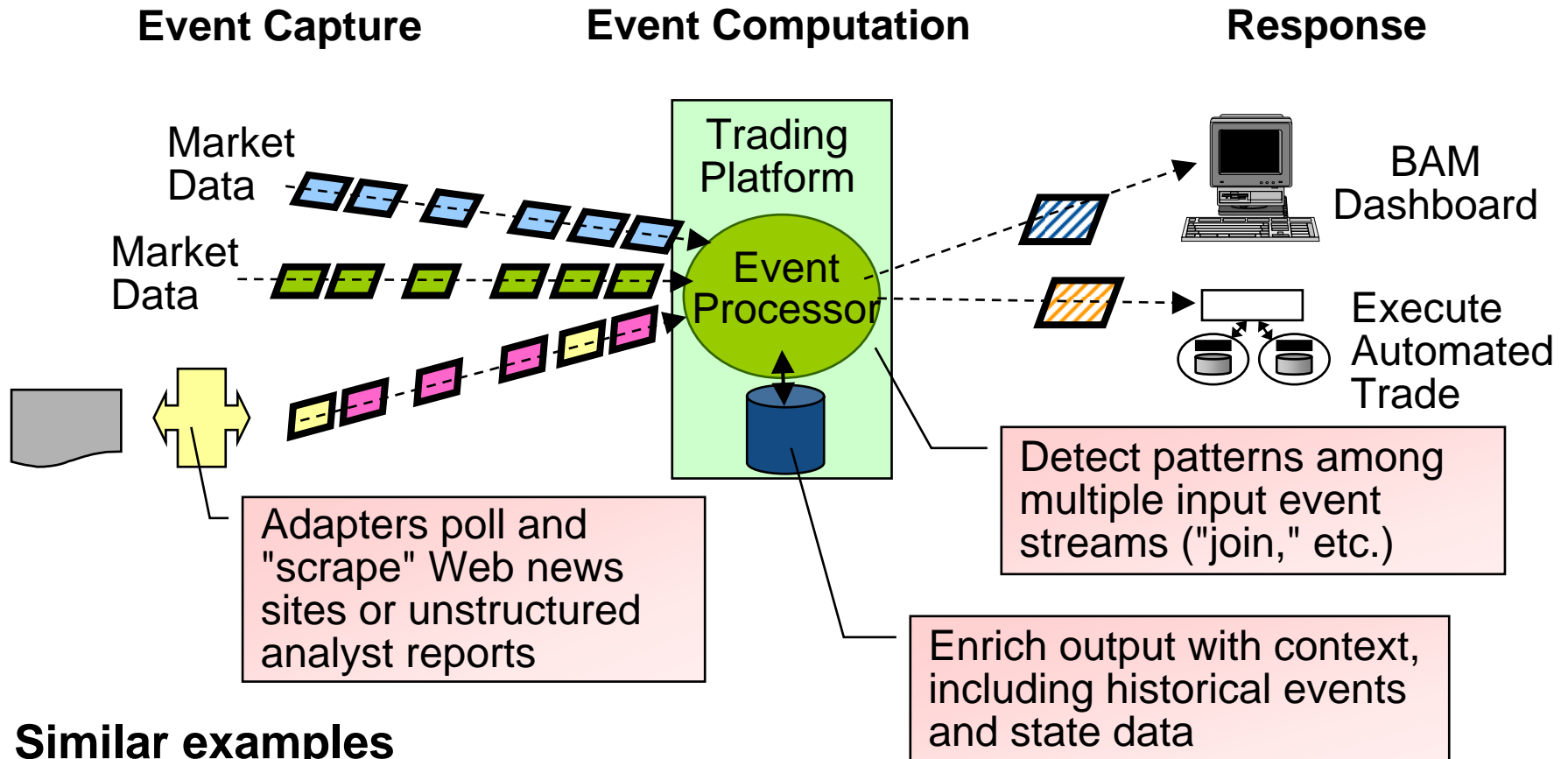
When EDA is implemented using SOA, it:

- Conforms to the five principles of SOA: Modularity, distributability, formal interface, separate interface from implementation, shareability
- Can leverage Web and Web services standards, such as URI, XML, WSDL and SOAP
- Should be coordinated by the SOA COE
- Coexists with other (non-EDA) types of SOA in the same applications
- Should share the same ESB, registry/repository, BPM, BI and other tools

However, EDA differs from conventional client/server SOA applications; it:

- Employs notification messages rather than request/reply interactions
- Can be used for asynchronous event-at-a-time staged business processes
- Can be used for sense-and-respond through complex-event processing

Example of Multiple Input Event Streams: Algorithmic Trading



Similar examples

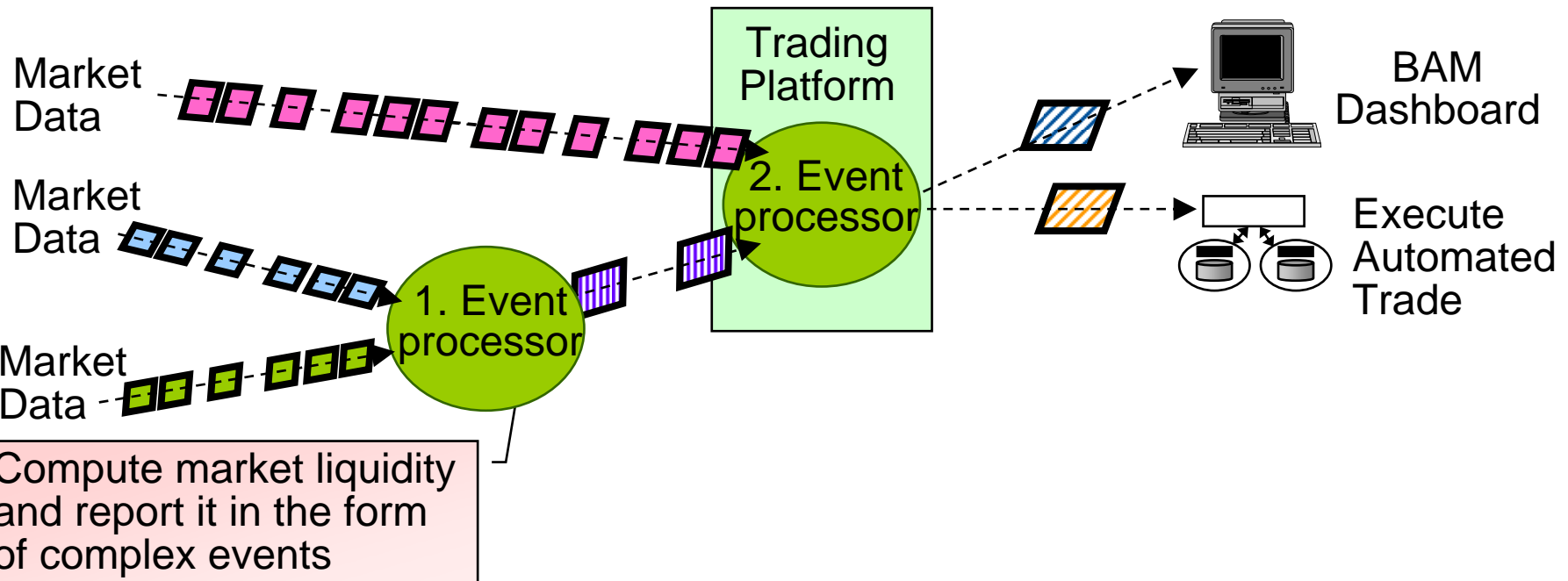
- Anti-money-laundering
- Market surveillance
- Market "best execution"
- Compliance, RegNMS, MiFID
- Hospital emergency room monitoring
- Fraud detection
- Electricity trading
- Battlefield operations

Example of Cascading Event Processing: Multistage Capital Markets Trading

Event Capture

Event Computation

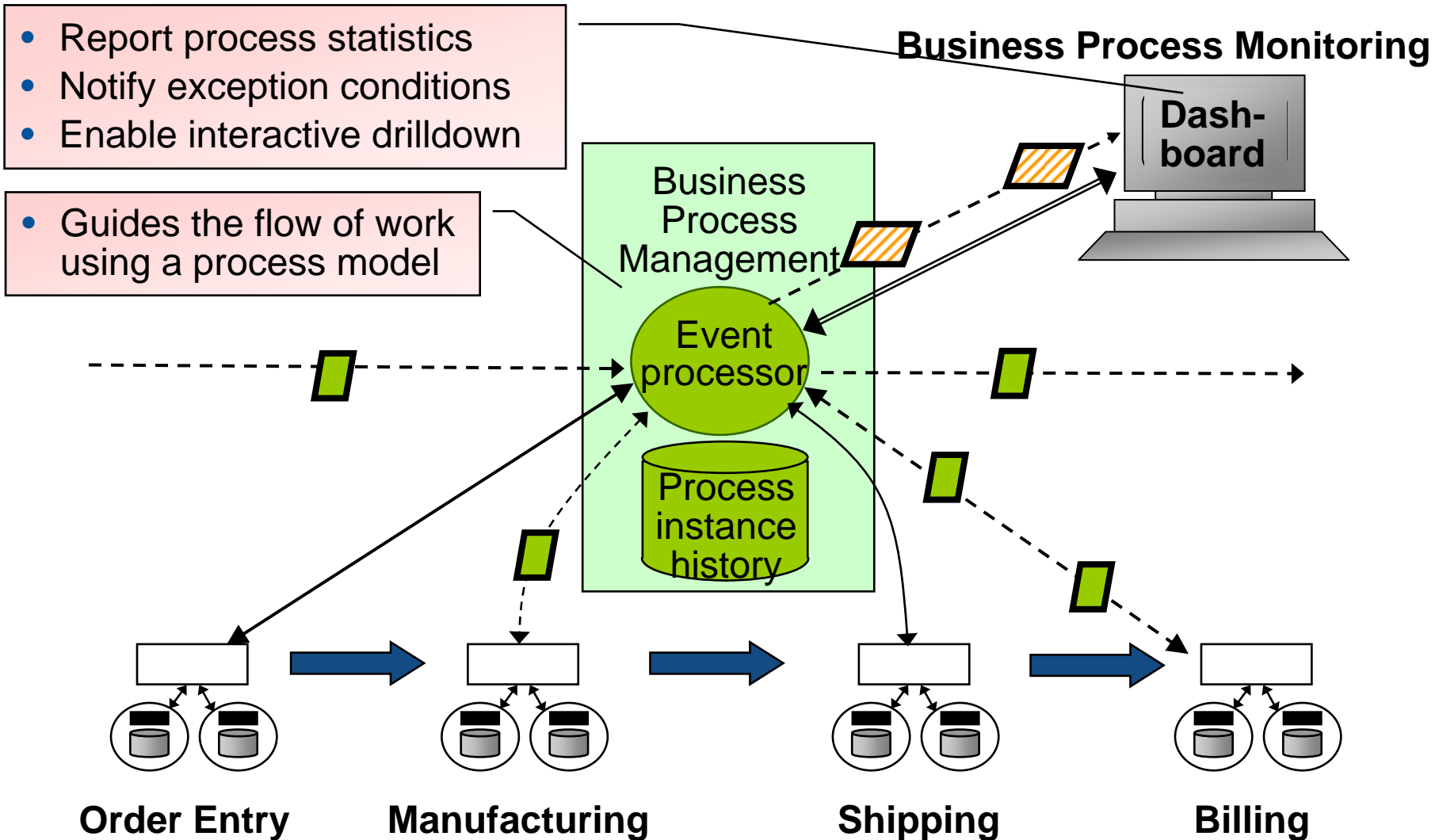
Response



Similar Examples

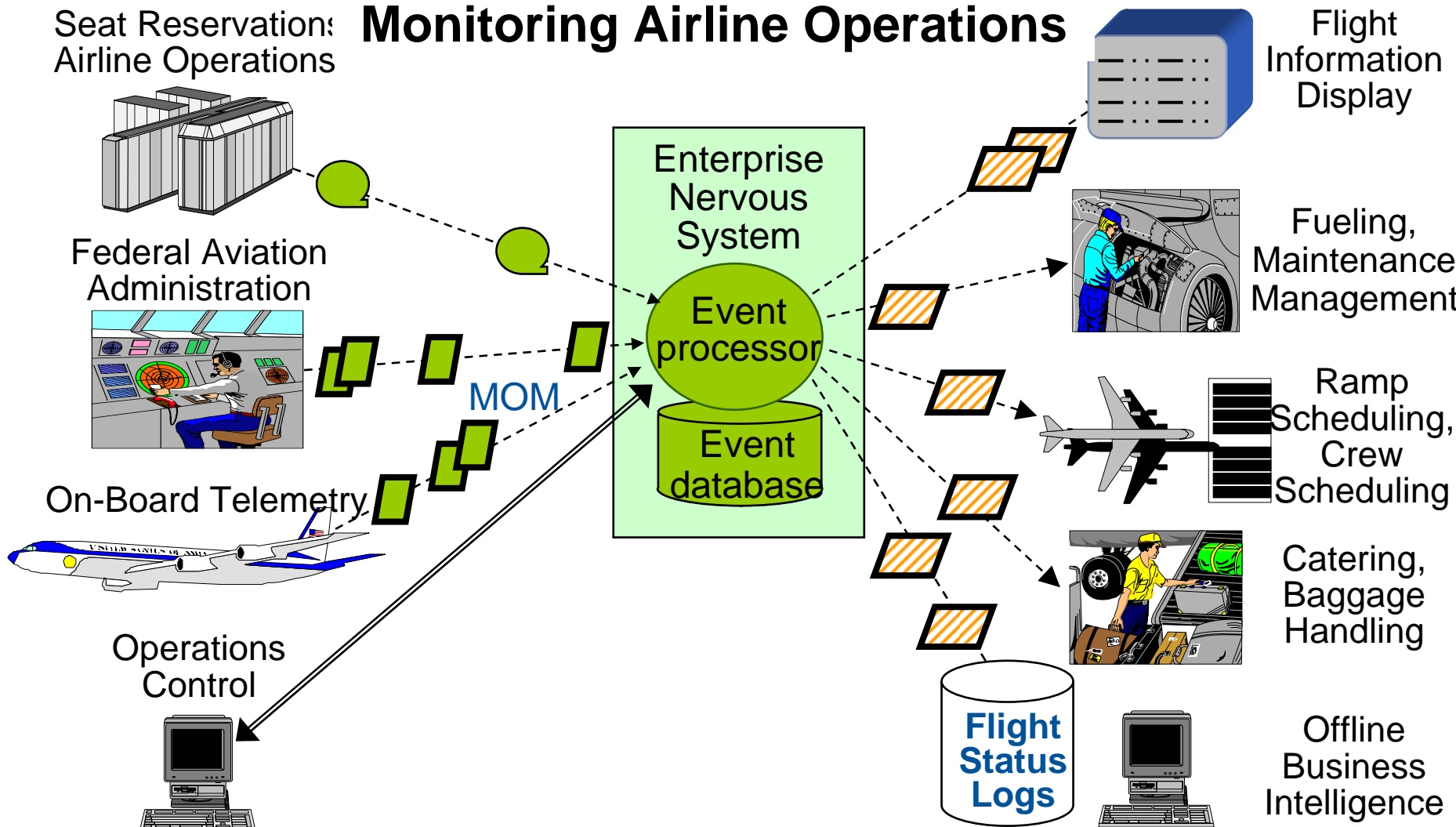
- (Stage 1) Cleanse raw market data feeds and pass to (Stage 2) trading platform
- (Stage 1) Analyze HTTP Web traffic and send reports to (Stage 2) rules-driven customer experience management application
- (Stage 1) Consolidate raw RFID reader events and report results to (Stage 2) supply chain management systems

Events Trigger Business Processes and Are Also Used to Monitor Them



Event Processing Is Often an Overlay on Top of Established Processes and Systems

Monitoring Airline Operations



Bottom Line

- SOA architects, business analysts and developers should enhance their event processing skills because event processing is significantly different from conventional client/server (C/S) and data-centric design patterns.
- Companies should make EDA part of their SOA strategy before starting their first major SOA application project.
- Teach business analysts how to identify business situations where derived events and CEP would bring tangible benefits
- Acquire CEP, BAM and dashboard software on a project-by-project basis, but reuse the same products in multiple projects where feasible