Event-Driven Architecture

Dominic Duggan
Stevens Institute of Technology

1

EVENT-DRIVEN

2

Time-Driven

Time-Driven

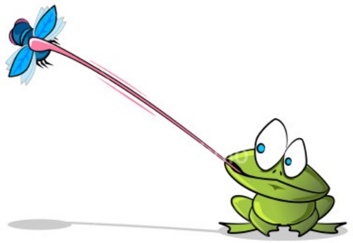
Customer places order on their iPhone
We convert the order to a telephone call, e-mail or fax
You handle the order as you normally would

Request-Driven

Time-Driven

Customer places order on their iPhone
We convert the order to a telephone call, e-mail or fax
You handle the order as you normally would

Request-Driven



Event-Driven

5

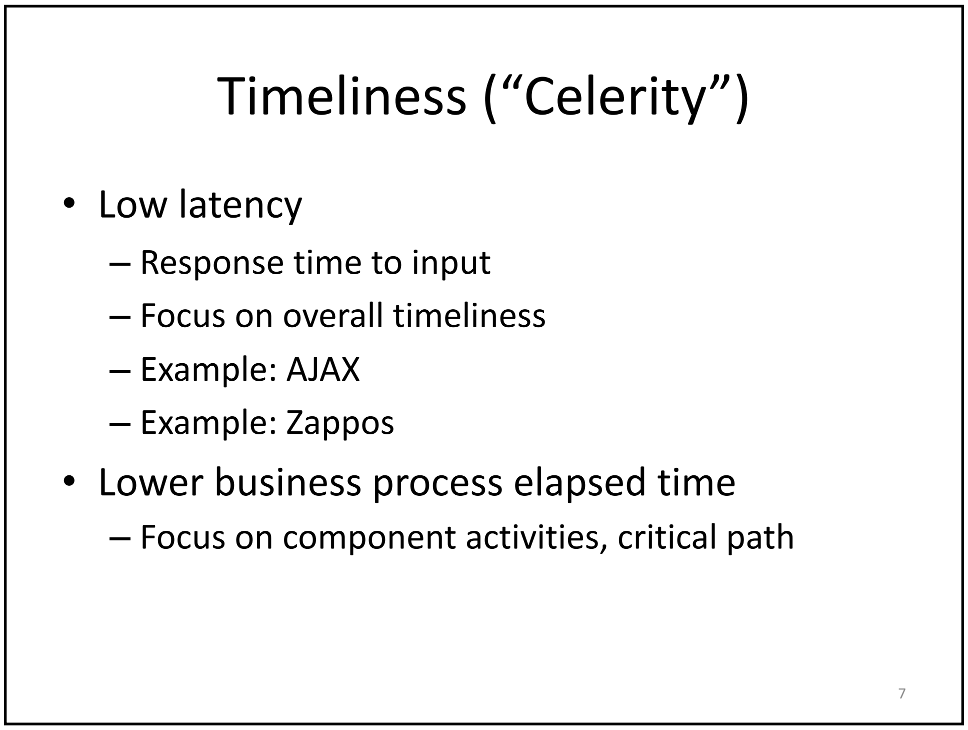
5

Event Processing

- Modern world is dynamic, competitive, global
- EP motivated by modern enterprise requirements
  - Timeliness
  - Agility
  - Information availability

6

6

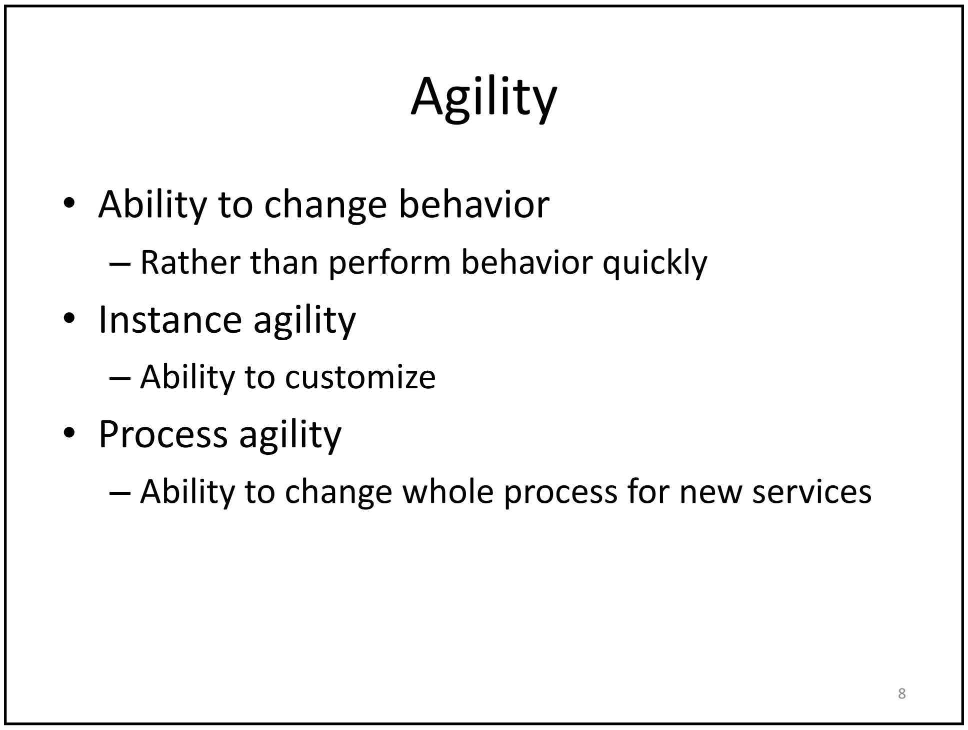


Timeliness (“Celerity”)

- Low latency
  - Response time to input
  - Focus on overall timeliness
  - Example: AJAX
  - Example: Zappos
- Lower business process elapsed time
  - Focus on component activities, critical path

7

7



Agility

- Ability to change behavior
  - Rather than perform behavior quickly
- Instance agility
  - Ability to customize
- Process agility
  - Ability to change whole process for new services

8

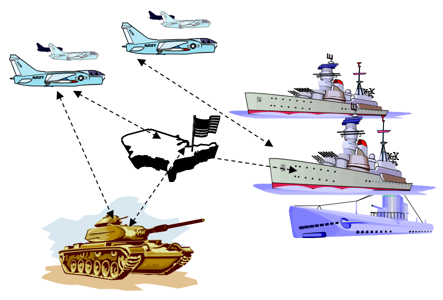
8

Information Availability

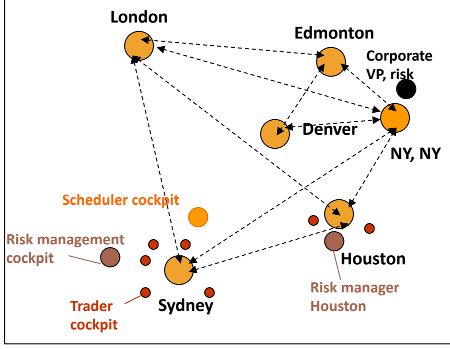
- Data consistency
  - Multiple redundant data repositories
  - How to synchronize contents?
- Information dissemination
  - MOM, text messages, tweets, alerts
- Situational awareness
  - Constant awareness of running operational activities
  - “Cut the fog:” identify and respond rapidly to new developments

Situational Awareness

Military situational awareness



Corporate situational awareness

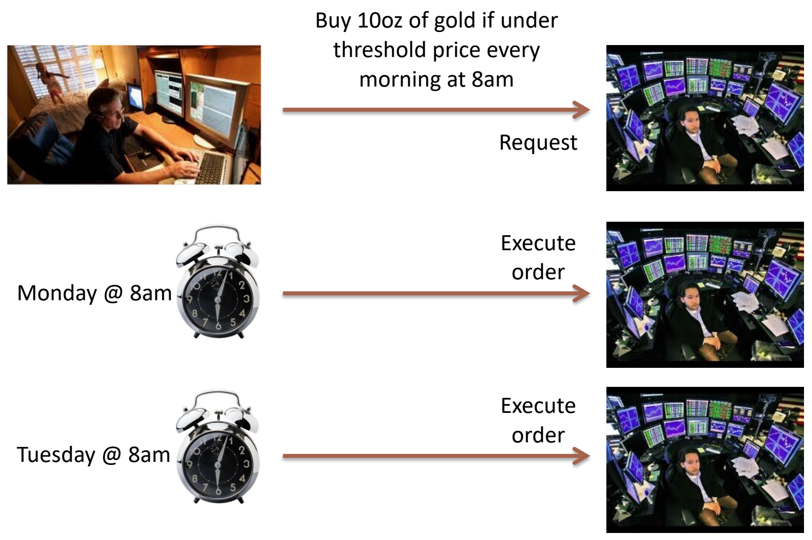


INTERACTIONS

11

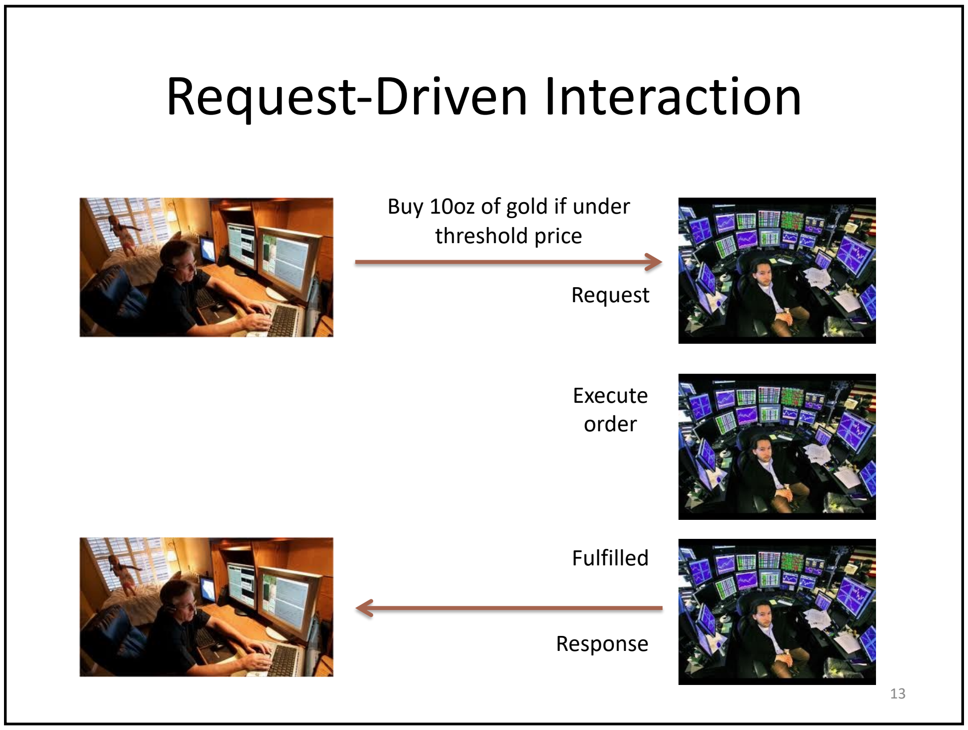
11

Time-Driven Interaction


Buy 10oz of gold if under threshold price every morning at 8am
Request
Monday @ 8am
Execute order
Tuesday @ 8am
Execute order

12

12



Request-Driven Interaction

Buy 10oz of gold if under threshold price

Request

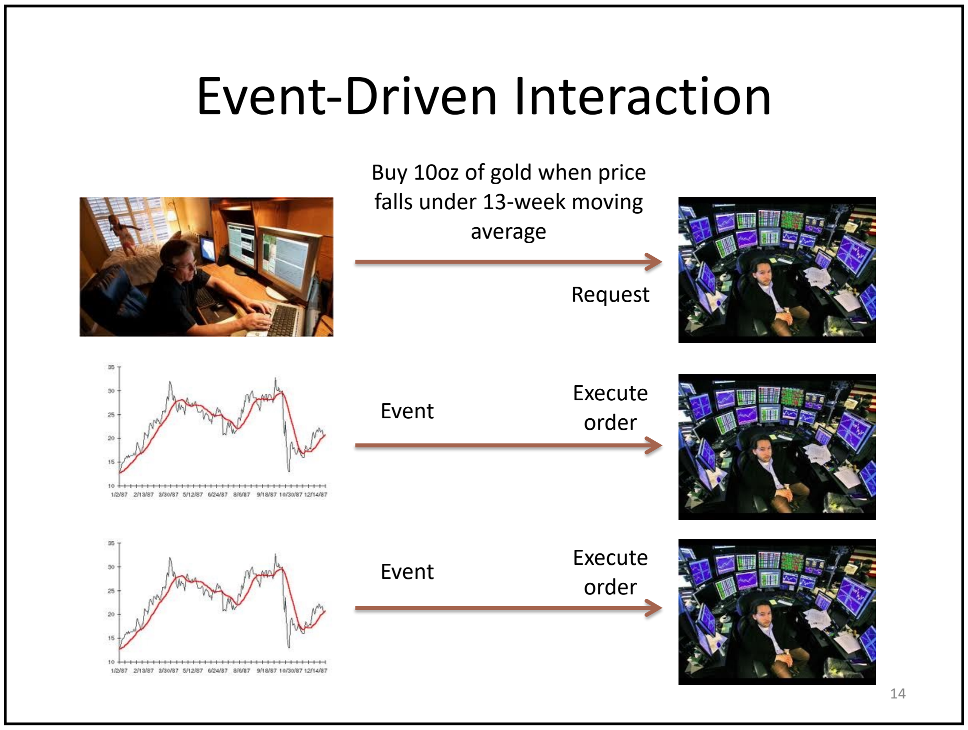
Execute order

Fulfilled

Response

13

13



Event-Driven Interaction

Buy 10oz of gold when price falls under 13-week moving average

Request

Event

Execute order

Event

Execute order

14

14

Time-Driven Interaction

- Advantages
  - Scheduled interaction: efficient sharing
  - Heartbeat mechanism
  - Regular measurements produce time series for analysis
  - Energy saving
- Not sustainable across large enterprises
  - Hybrid: event-driven with local time-driven

15

15

Request-Driven Interaction

- Short-term, perhaps stateless, interaction
- Clearly defined initiation and termination points
  - Client focus on a particular service
- Celerity: others may need timely response
  - Hybrid of request-driven and event filtering
  - Personal information manager

16

16

Event-Driven Interaction

- Long-term stateful interaction
- Bottom-up notification
- Drawing timely data from disparate sources

17

17

Contracts

- Time-driven:
  - Relation between pre-conditions and post-conditions of all participating agents
- Request-driven:
  - Relation between client request and server reply *(logic coupling)*
  - Maybe real-time constraint on server, not on client
- Event-driven:
  - *“When-then” rules*

18

18

Event-Driven Contract

- Contract is for an interval of time
- Contract is between agent and rest of system
  - Identity of notifier not important
- When-then rules
  - When pet is sick, call the vet
  - When child is sick, if fever then call doctor
- Rules should be executed in timely fashion
  - *Absence* of messages conveys information (cf situational awareness)

19

19

Hybrid Systems

- Example: Personal information manager
  - Celerity: notifies you of significant events
  - Prevents interruptions otherwise
- Example: E-mail
  - Receipt is event-driven
  - Deposited in mail folder
  - Retrieval is request-driven

20

20

Hybrid Systems

- Example: Location-based services
  - Location based on GPS, cell, wifi fingerprinting
  - Services based on location, profile
- Example: Emergency response
  - Emergency detection is event-driven
  - Dispatch of services is request-driven (SOA)

21

21

EVENT PROCESSING

22

22

Components of Event Processing

- Events
- Business events
- Event objects: discrete reports
  - “A message with an attitude”
- Event-driven
- Event-driven architecture
  - Loosely coupled: interaction between components is based only on events

23

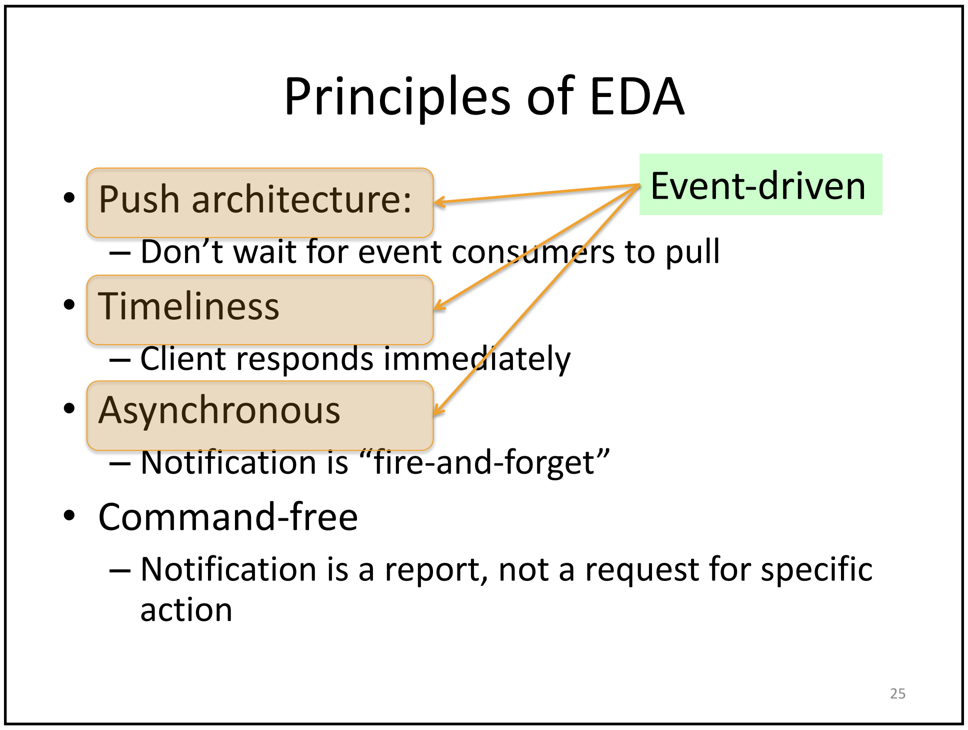
23

Principles of EDA

- Push architecture:
  - Don’t wait for event consumers to pull
- Timeliness
  - Client responds immediately
- Asynchronous
  - Notification is “fire-and-forget”
- Command-free
  - Notification is a report, not a request for specific action

24

24



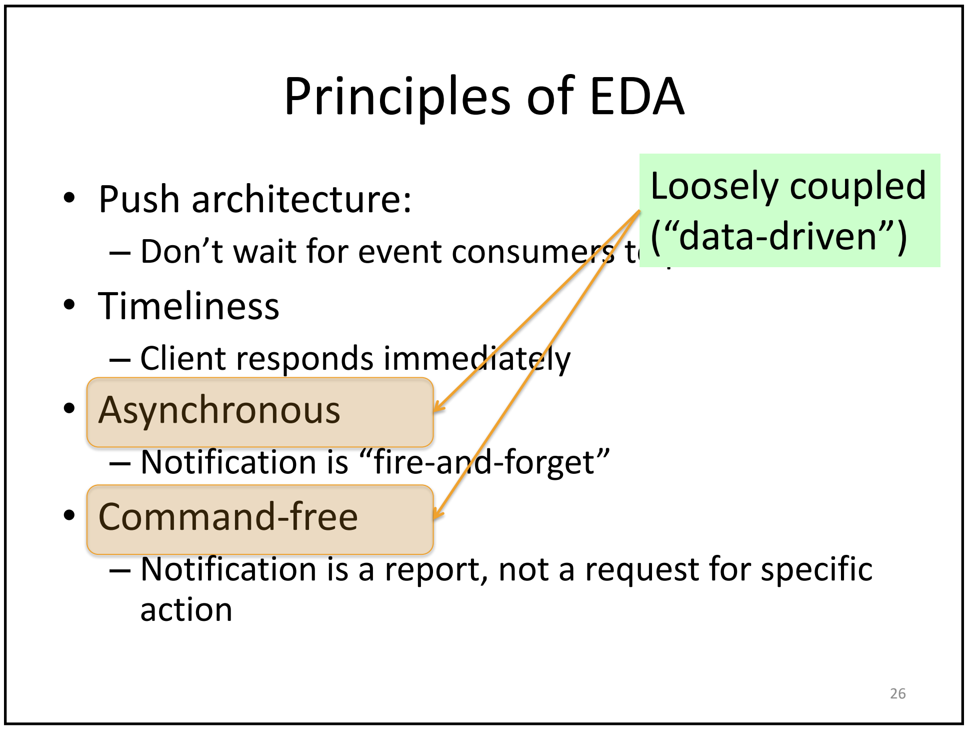
Principles of EDA

Event-driven

- Push architecture:
  - Don’t wait for event consumers to pull
- Timeliness
  - Client responds immediately
- Asynchronous
  - Notification is “fire-and-forget”
- Command-free
  - Notification is a report, not a request for specific action

25

25



Principles of EDA

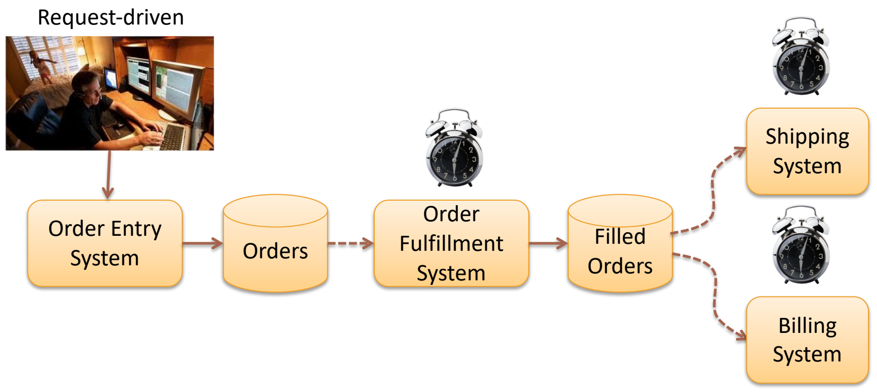
Loosely coupled (“data-driven”)

- Push architecture:
  - Don’t wait for event consumers to pull
- Timeliness
  - Client responds immediately
- Asynchronous
  - Notification is “fire-and-forget”
- Command-free
  - Notification is a report, not a request for specific action

26

26

Timeliness

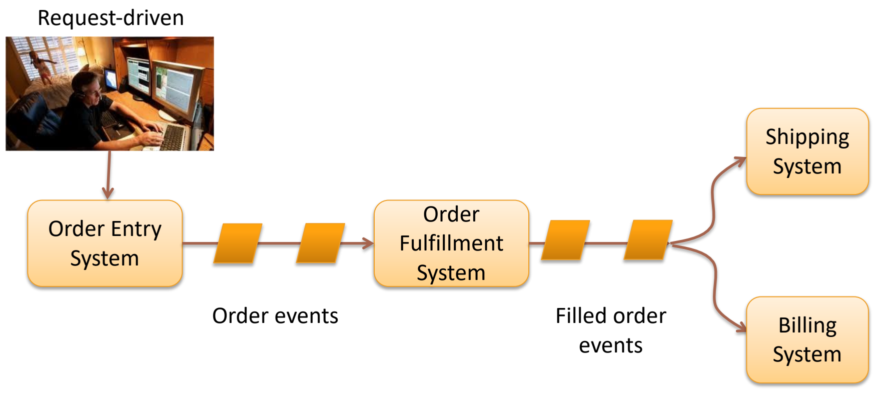


Time-driven batch process

27

27

Timeliness

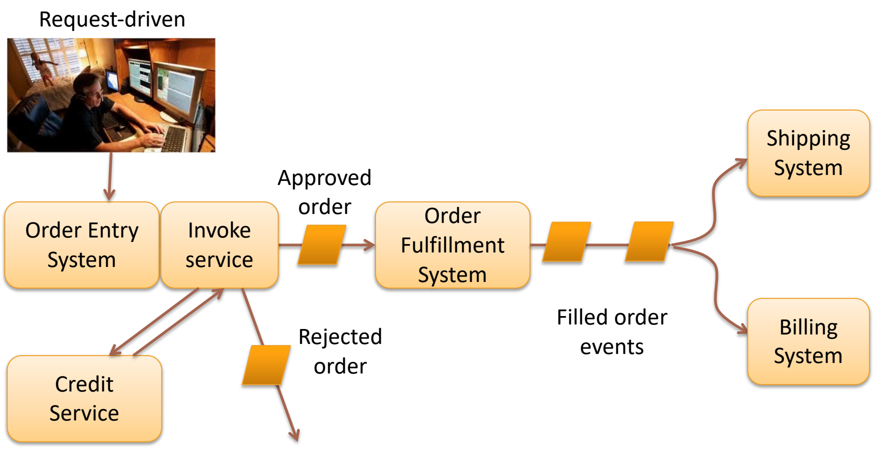


Event-driven continuous process

28

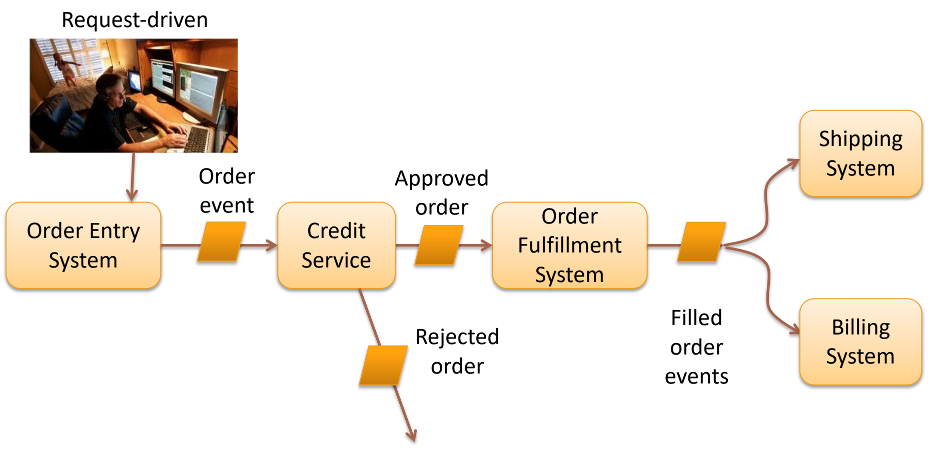
28

Agility



Inserting a request-driven process

Agility



Inserting an event-driven service

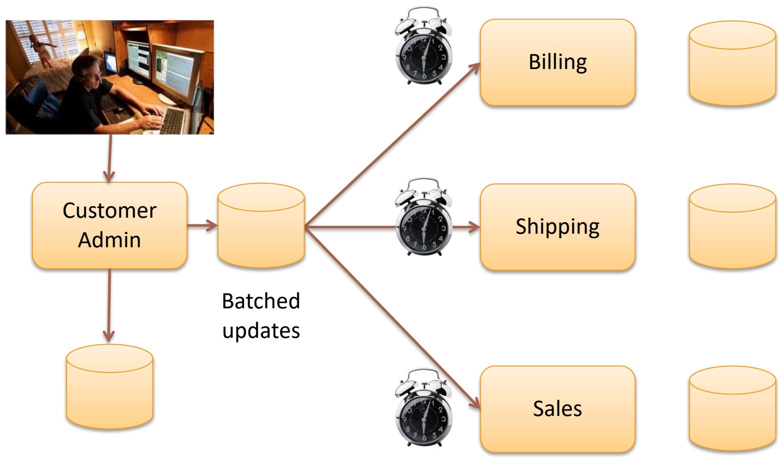
Agility

- Event-driven services:
  - Loose coupling
  - Accommodate piecemeal change
  - Reduce inter-component dependencies
- Request-driven services:
  - Accommodate feedback in interaction
  - Ex: notification of credit problem

31

31

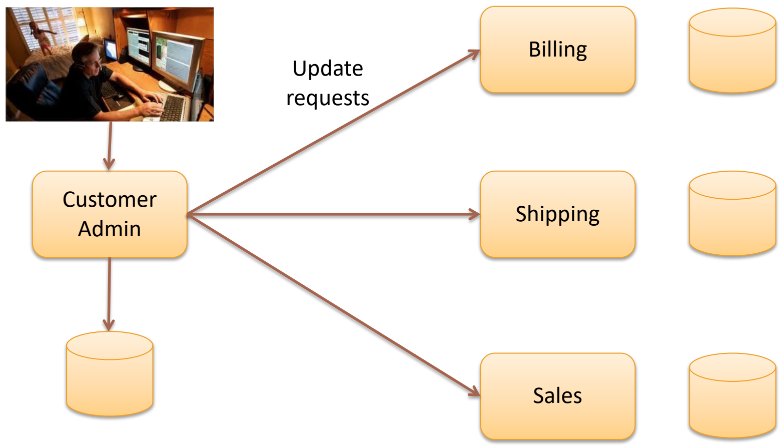
Information Availability



32

32

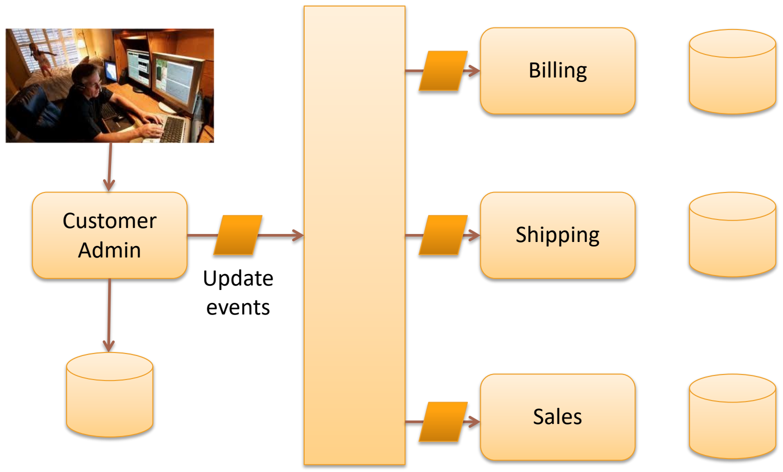
Information Availability



33

33

Information Availability



34

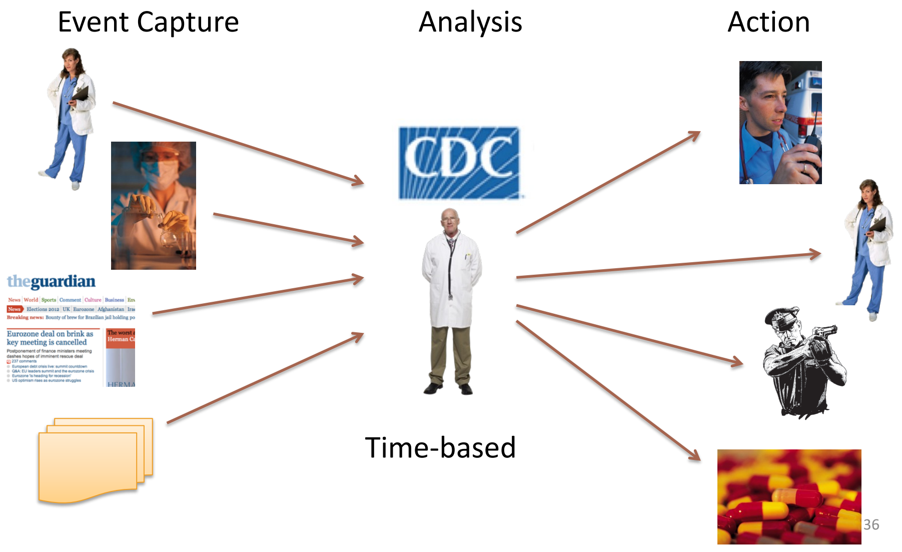
34

COMPLEX EVENT PROCESSING

35

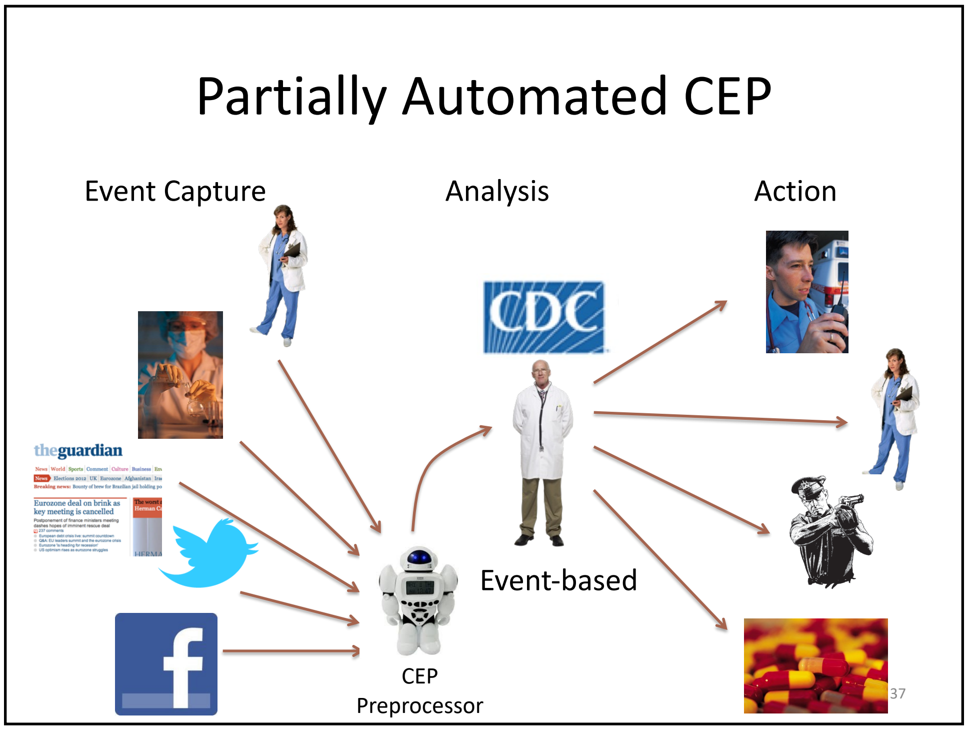
35

Manual CEP



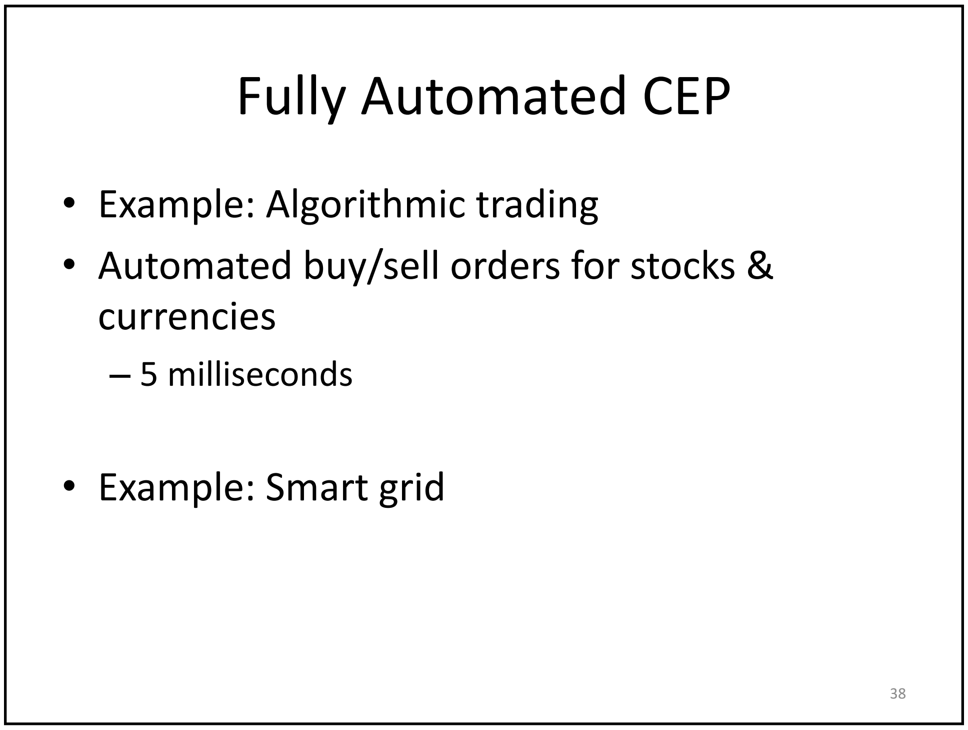
36

36



Partially Automated CEP

37



Fully Automated CEP

- Example: Algorithmic trading
- Automated buy/sell orders for stocks & currencies
  - 5 milliseconds
- Example: Smart grid

38

38

CEP Pattern Detection

- Example: correlating two event streams

```
NewsArticle(About Stock X) followed by
StockPriceRise(Stock X, > 5%) within 3
minutes.
```

- Note: StockPriceRise is itself result of CEP-based continuous analytic pre-processing
- Causality:
  - Vertical: Agent_1 Login $\rightarrow$ Agent Available
  - Horizonal: Agent_1 Login $\rightarrow$ Agent_1 Logout

39

39

Variations on Complex Events

- Uncertain events
  - Diagnostic
  - Predictive
- Absent events
- Complexity is relative
  - Example: stock trade reports
  - Short summary of complex interaction

40

40

EVENT PROCESSING ARCHITECTURE

41

41

Contracts

Request-driven

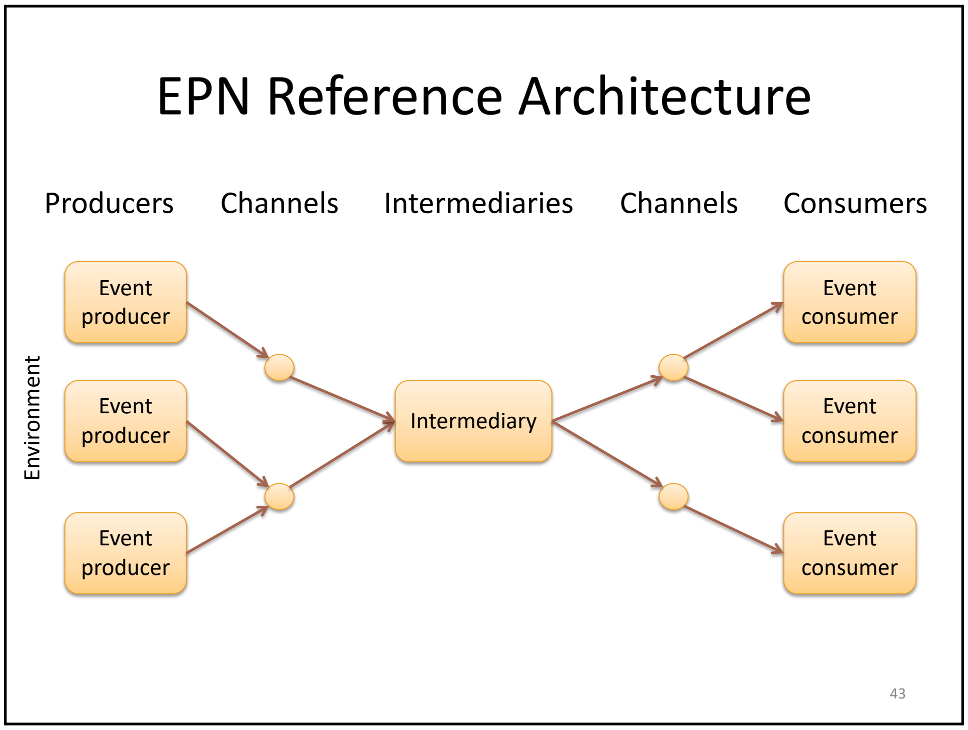
- Client request initiates, “consumed” by provider
- Service provider expected to respond to client requests
- Service client has no obligation to provider

Event-driven

- Event notification initiates, “consumed” by consumer
- Event producer expected to emit event whenever something happens
- Event consumer has no obligation to producer
  - Obligation at higher level

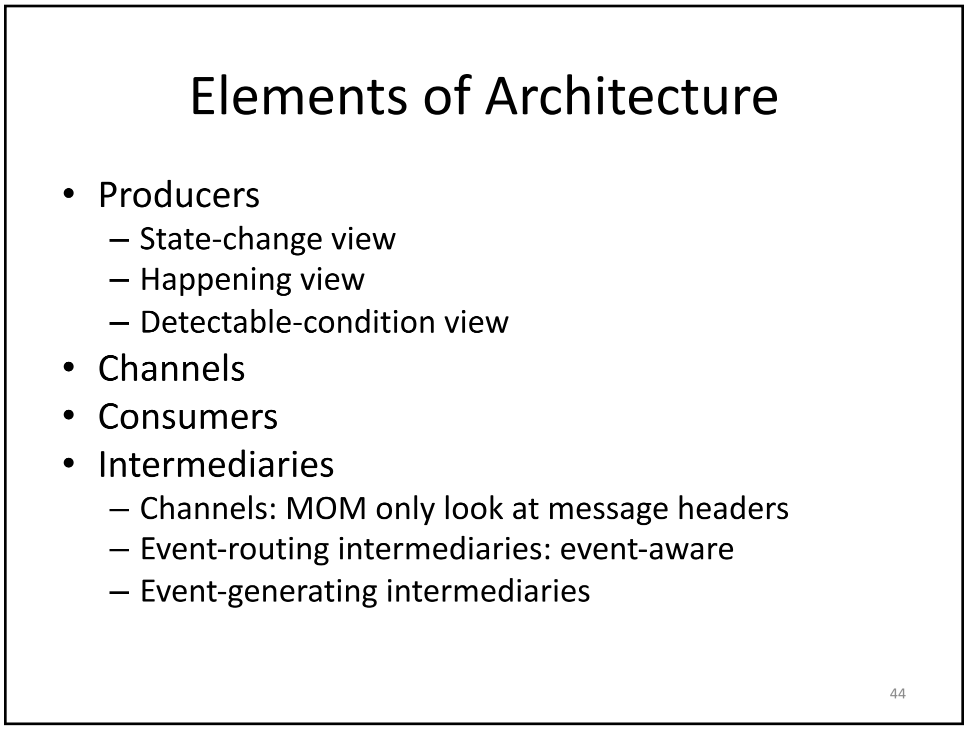
42

42



EPN Reference Architecture

43



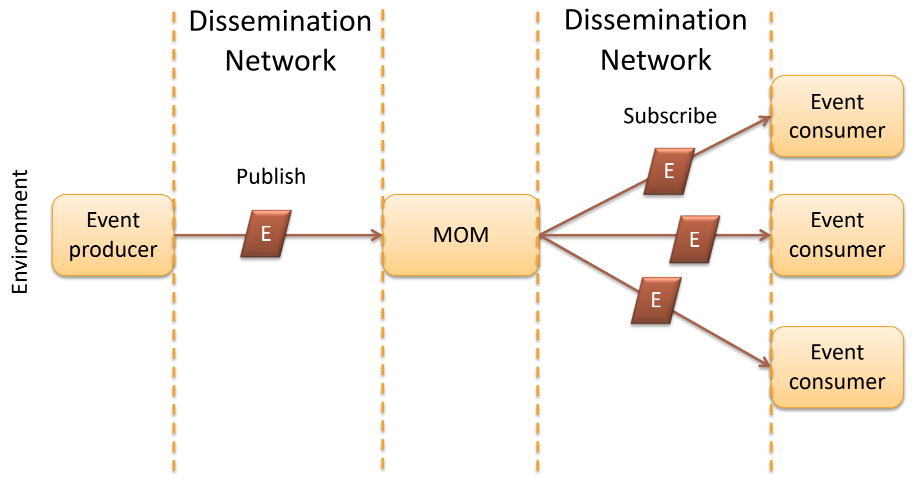
Elements of Architecture

- Producers
  - State-change view
  - Happening view
  - Detectable-condition view
- Channels
- Consumers
- Intermediaries
  - Channels: MOM only look at message headers
  - Event-routing intermediaries: event-aware
  - Event-generating intermediaries

44

44

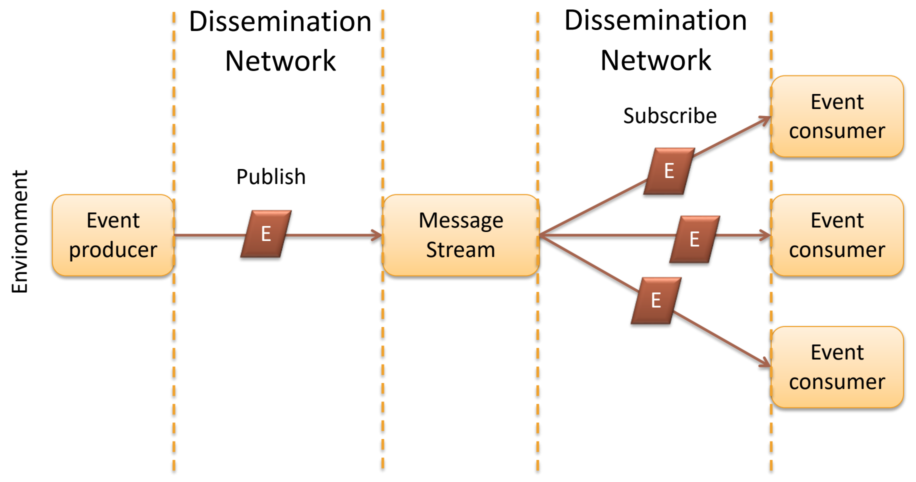
EPN for Information Dissemination



45

45

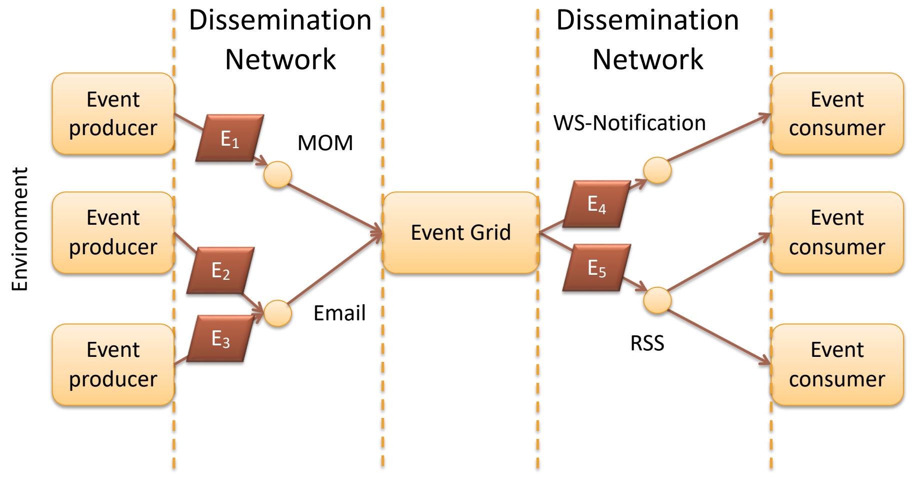
EPN for Information Dissemination



46

46

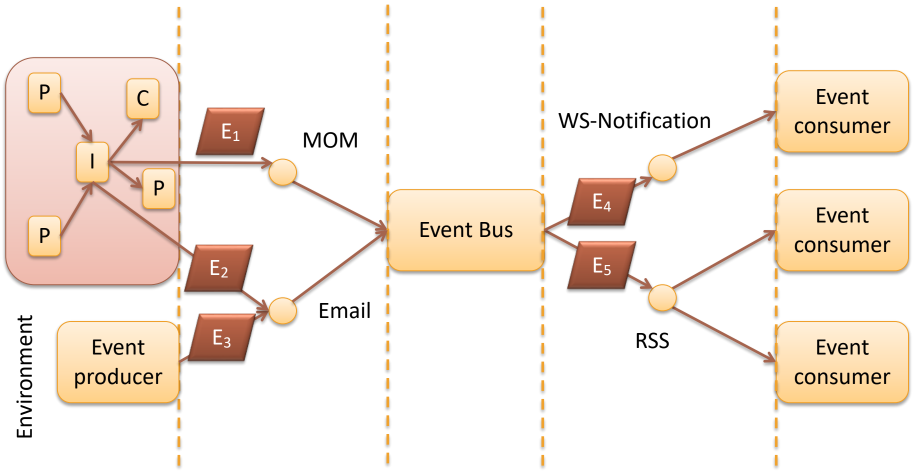
EPN for Situation Awareness



47

47

EPN for Application Integration



48

48