Event Driven Architecture

Lecture-6

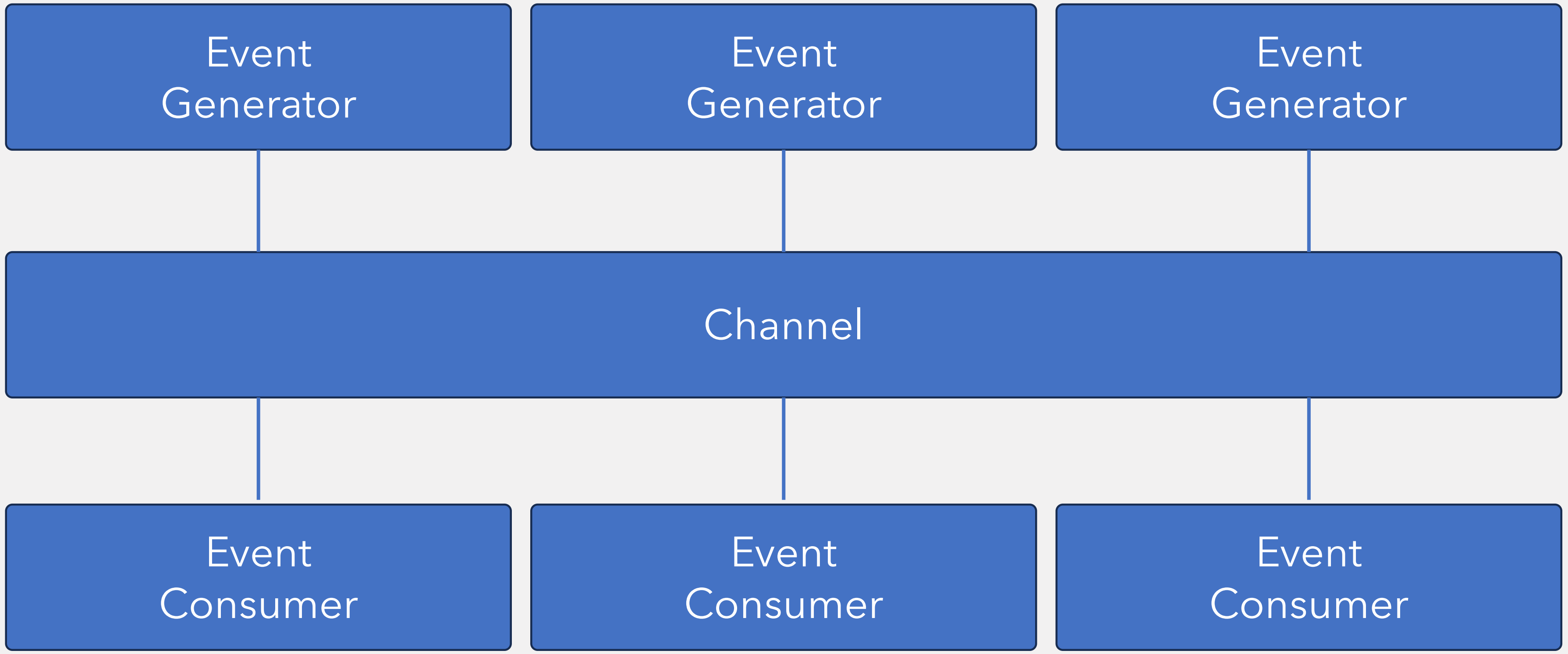
Styles of Interaction

Type Of Interaction	Initiator	Participants
Time-Driven	Time	The Specified Systems
Request-Driven	Client	Client & Server
Event-Driven	Event	Open Ended

What is Event Driven Architecture

- It’s a way of building solutions based on the generation, reception, processing and sending of events
- Events flow between decoupled services and components
  - *Generator and Consumers are obvilious to each other*
- Used for building extensible complex distributed systems which are governed by asynchronus events
  - There is no command and control structure
  - No request and response mechanisms

Main Component of EDA



Main Components

- **Event** – entity depicting something that happened
- **Event Generator** – entity that creates and pushes the event
- **Event Channel** – medium where the event lies before consumption
- **Event Consumer** – entity which consumes the events from the channel

Event

- An event is a piece of information that may be applicable to one or more objects at a given time
- An event would be
    1. self contained
    2. unique
    3. time relevant
- An event will propose
    1. a problem
    2. an opportunity
    3. a threshold
    4. a deviation, to its consumer
- Resulting action due to the occurrence of an event could be
    1. invocation of a service
    2. triggering a business process
    3. publishing of another event

Event Generators, Consumers and Channel

- Event Generators
    - publish events in the form of messages
    - have no idea who would consume the message
    - have no idea about the consequences of the messages generated
- Event Consumers
    - subscribe to published events
    - handles events in an asynchronous manner
    - oblivious to other consumers
- Event Channel
    - acts as the mediator
    - handles subscriptions
    - handles storage and delivery

Principles of Event-Driven Architecture (EDA)

1. “Real-time” events at the Producer:

    Events occur in real-time as they happen at the source or producer, providing timely and up-to-date information.

2. Push Notifications:

    Events are actively pushed to the consumers, ensuring immediate awareness and reducing latency in data dissemination

3. One-way “Fire-and-Forget”:

    Events follow a one-way, "fire-and-forget" approach, where the producer emits events without waiting for a response, simplifying the interaction and enhancing efficiency

4. Immediate Action at the Consumers:

    Consumers of events take immediate actions upon receiving the information, allowing for swift and responsive behavior in the system.

5. Informational, not Commands:

    Events convey informational messages such as "someone logged in" rather than issuing commands like "audit this." This principle emphasizes the broadcasting of relevant information rather than direct commands, promoting a decoupled and scalable architecture.

What EDP brings to an Architecture

- Event hierarchies and fine grainedness
- Self describing nature of the event payload
- No special packing for each receiver
- Multiple receivers for a single event
- Responsibility of handling state is with the consumer
- Anonymity when sending an event
- Real time sending of events
- Async nature of delivery
- Guaranteed delivery

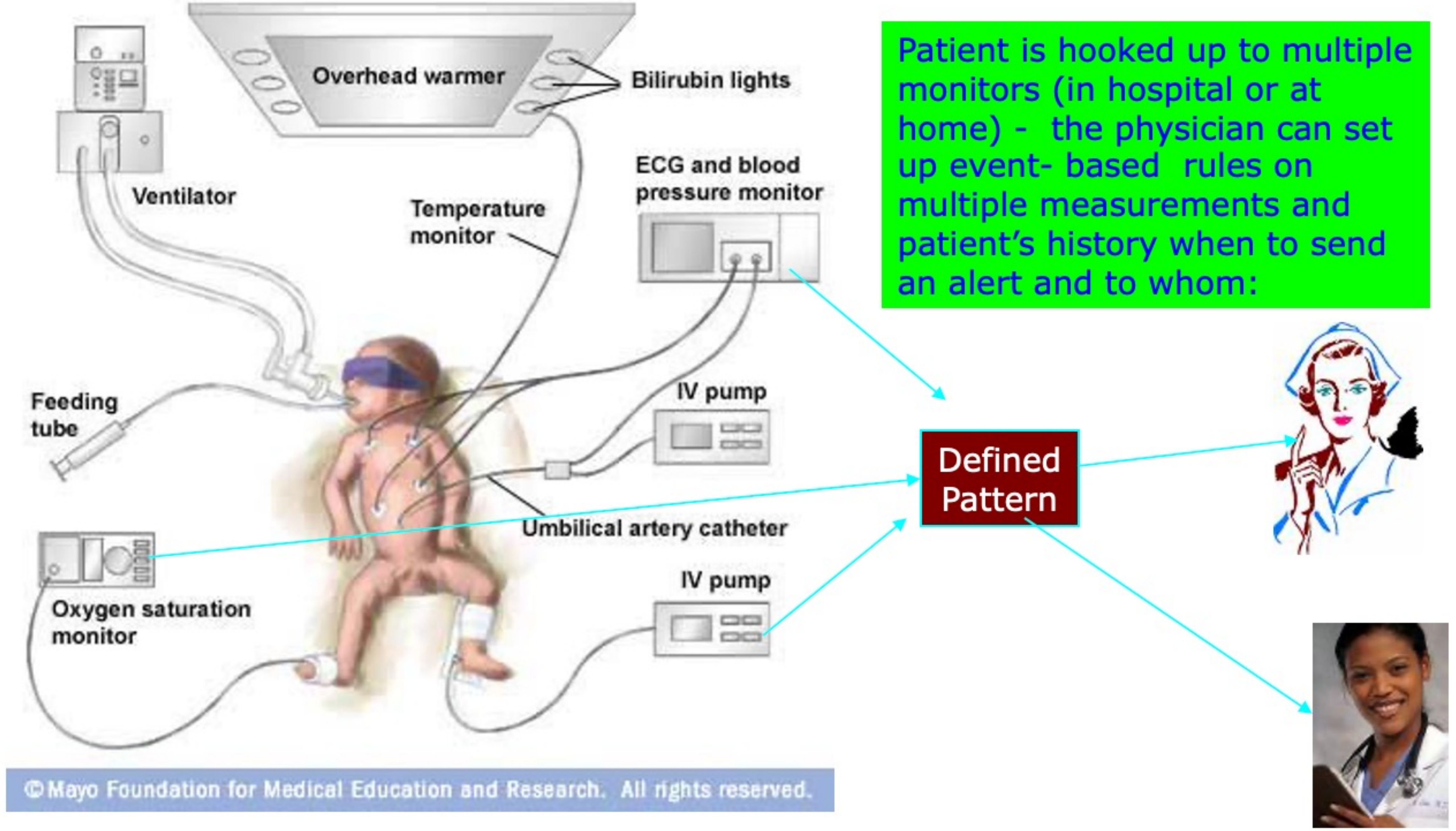
What EDP removes from an Architecture

- Co-ordination
- Pre-planned continuity
- Global context with the latest status

These only work where

  - the individual executions are fast
  - Actions only happen in a pre-defined order which does not change frequently
  - There is a fixed number of parties involved which the primary orchestrator knows of

Patient is hooked up to multiple monitors (in hospital or at home) - the physician can set up event- based rules on multiple measurements and patient’s history when to send an alert and to whom:


Overhead warmer
Bilirubin lights
Ventilator
Temperature monitor
ECG and blood pressure monitor
Feeding tube
IV pump
Umbilical artery catheter
Oxygen saturation monitor
IV pump
© Mayo Foundation for Medical Education and Research. All rights reserved.
Defined Pattern