

Applying real-time analytics to data streams in digital health

Reducing costs, detecting data quality issues and improving patient care

Dr Zoran Milosevic
zoran@deontik.com

Agenda

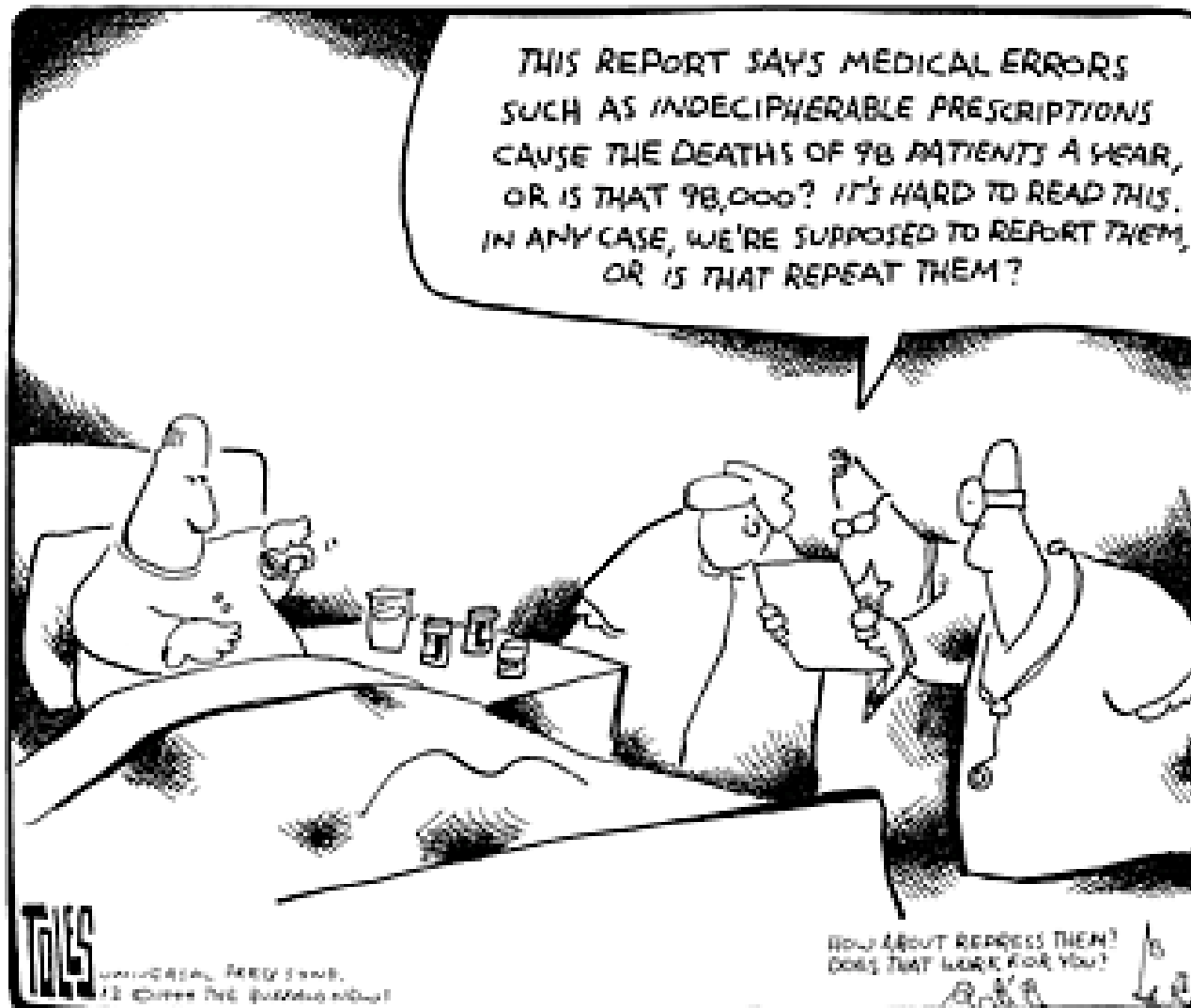
- Digital health requirements
- Use cases
- EventSwarm CEP Architecture
- Implementation
- Future research
- Demo

Improved quality of health care



Better access to information

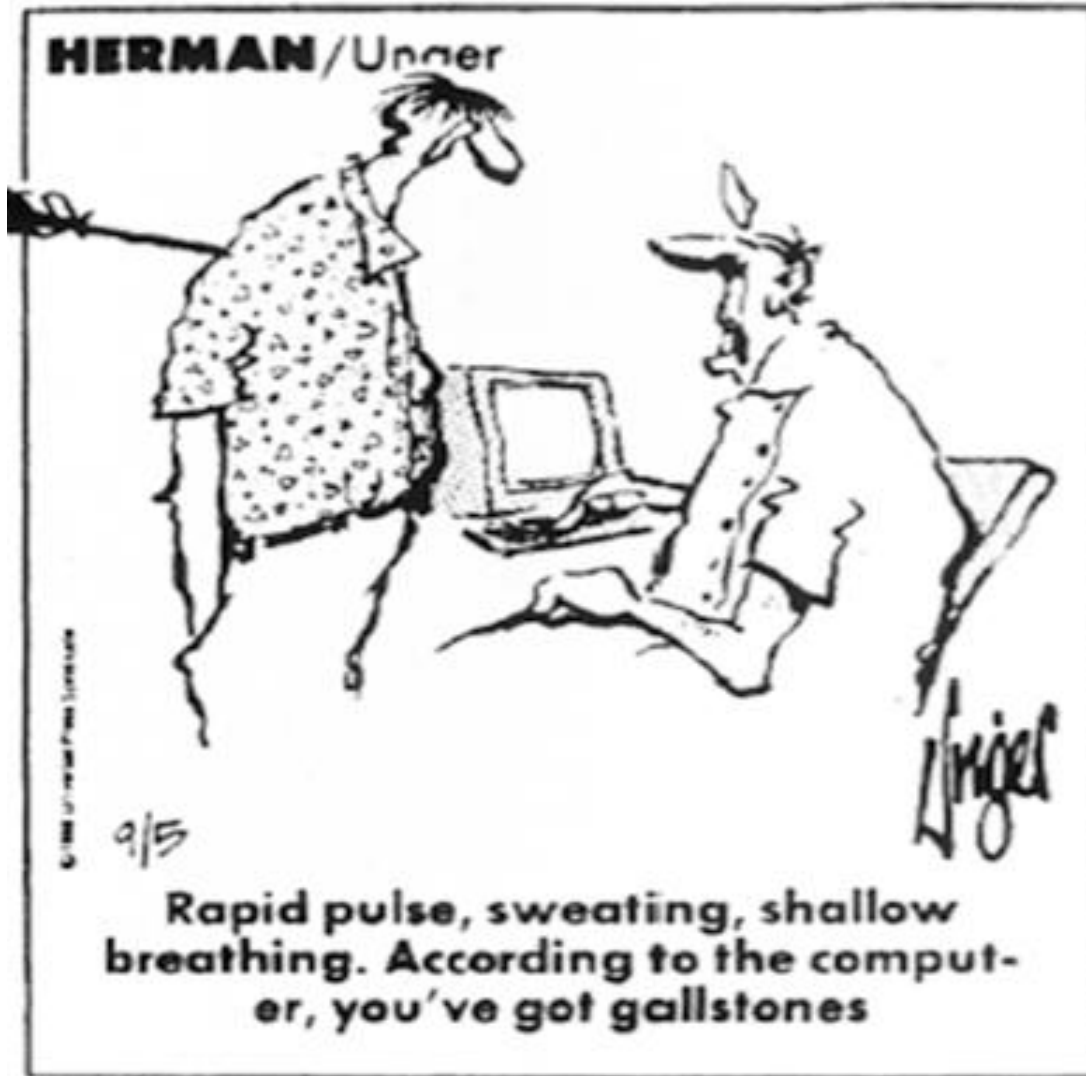
Reduced medical errors



Data Quality
!

TOLES©1999 The Buffalo News. Reprinted with permission of UNIVERSAL PRESS SYNDICATE. All rights reserved.

Patient Safety first !



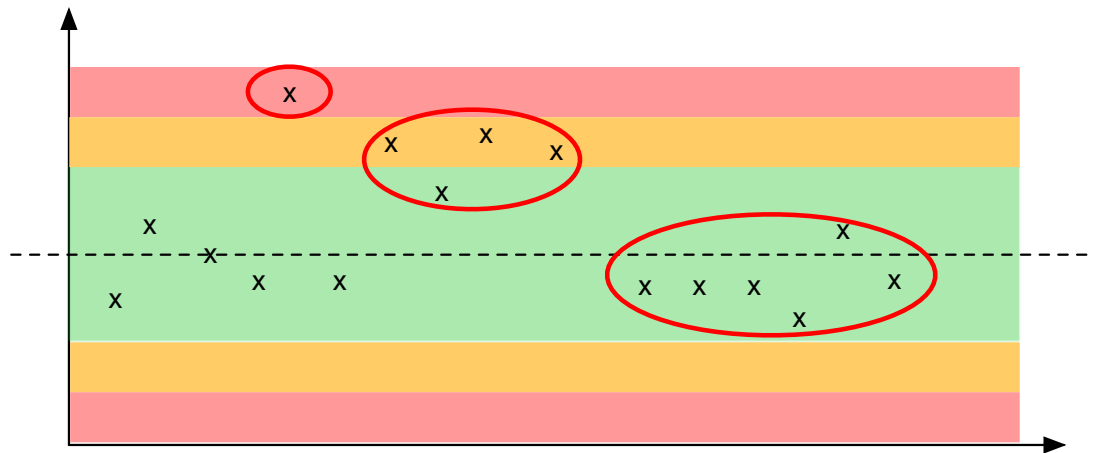
Clinician in the loop !

Use Cases

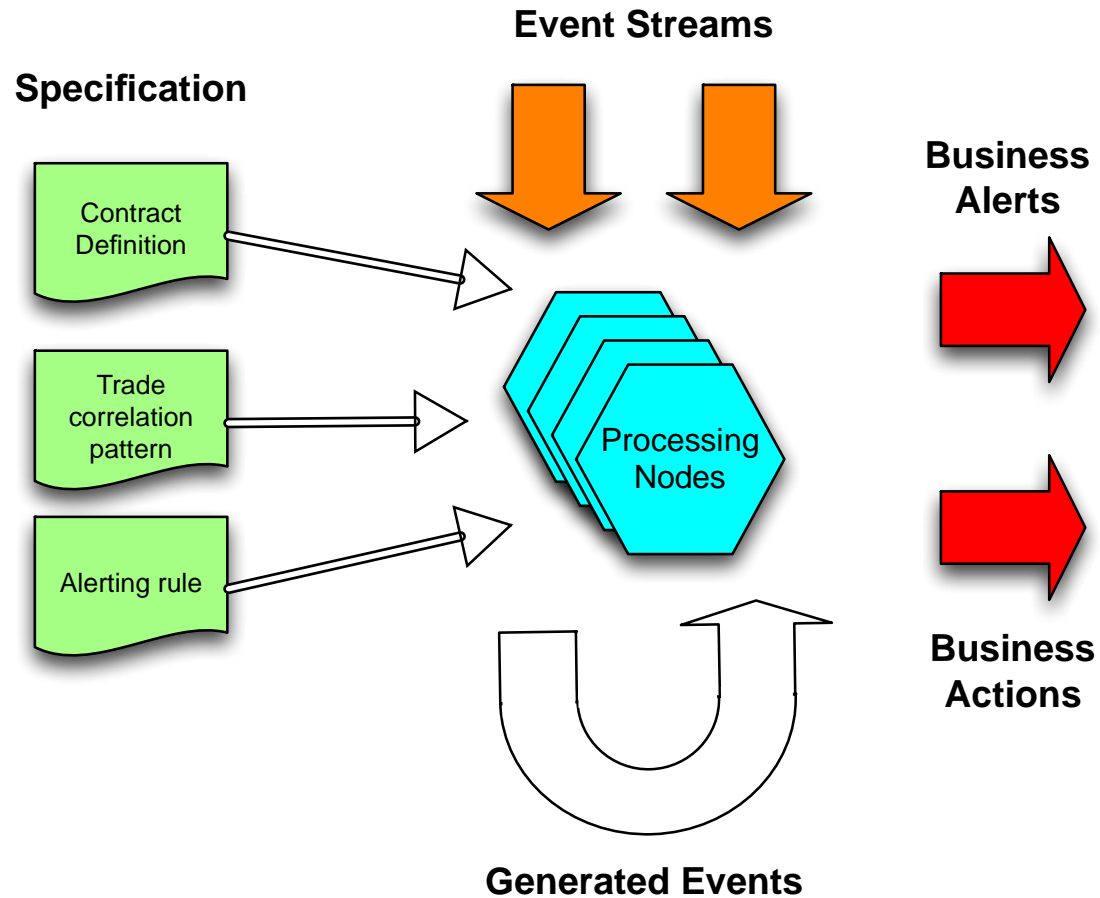
- **UC1: Reduce unusual/duplicate lab orders**
 - Problem: cost and convenience – useful life time limited by time windows
 - Solution: find duplicates in configurable window
- **UC2: Data quality issues**
 - Problem: detecting potential IT failures that may harm care delivery
 - Solution: syndromic surveillance algorithm to detect anomalies – via outliers
- **UC3: Context-specific lab result alerts – personalisation**
 - Problem: real-time detection of arising clinical conditions
 - E.g. Hemoglobin A1C (blood sugar/diabetes), Serum creatinine (kidney function)
 - Solution: compare event streams with data from EHRs – cascading rules
 - clinical decision support (CDS) application

Syndromic surveillance

- UC2: Research by Prof Enrico Coiera et al, UNSW
 - applies techniques previously used for disease outbreak
 - aiming to detect quality issues in Health IT systems
 - outliers may suggest equipment failures – use of SD measure



EventSwarm CEP Architecture



CEP Benefits

Operational Costs



*Contract violations,
network problems ...*



**Real-time
Visibility**

**Immediate
response**



New opportunity



**Identify quickly
through prediction**



**Scalability
through automation**

*Dealing with growing data
volume and velocity*



**Monitoring and
Pattern Detection**

*Humans focus:
decision making*



**Automated monitoring
improves accuracy**

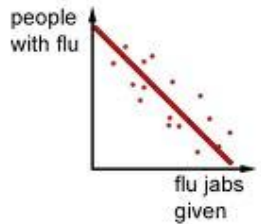
*e.g. patient condition
indicators*



**Event correlation improves
detection capability**

*e.g. twitter chats for detecting
disease outbreaks*

O
p
e

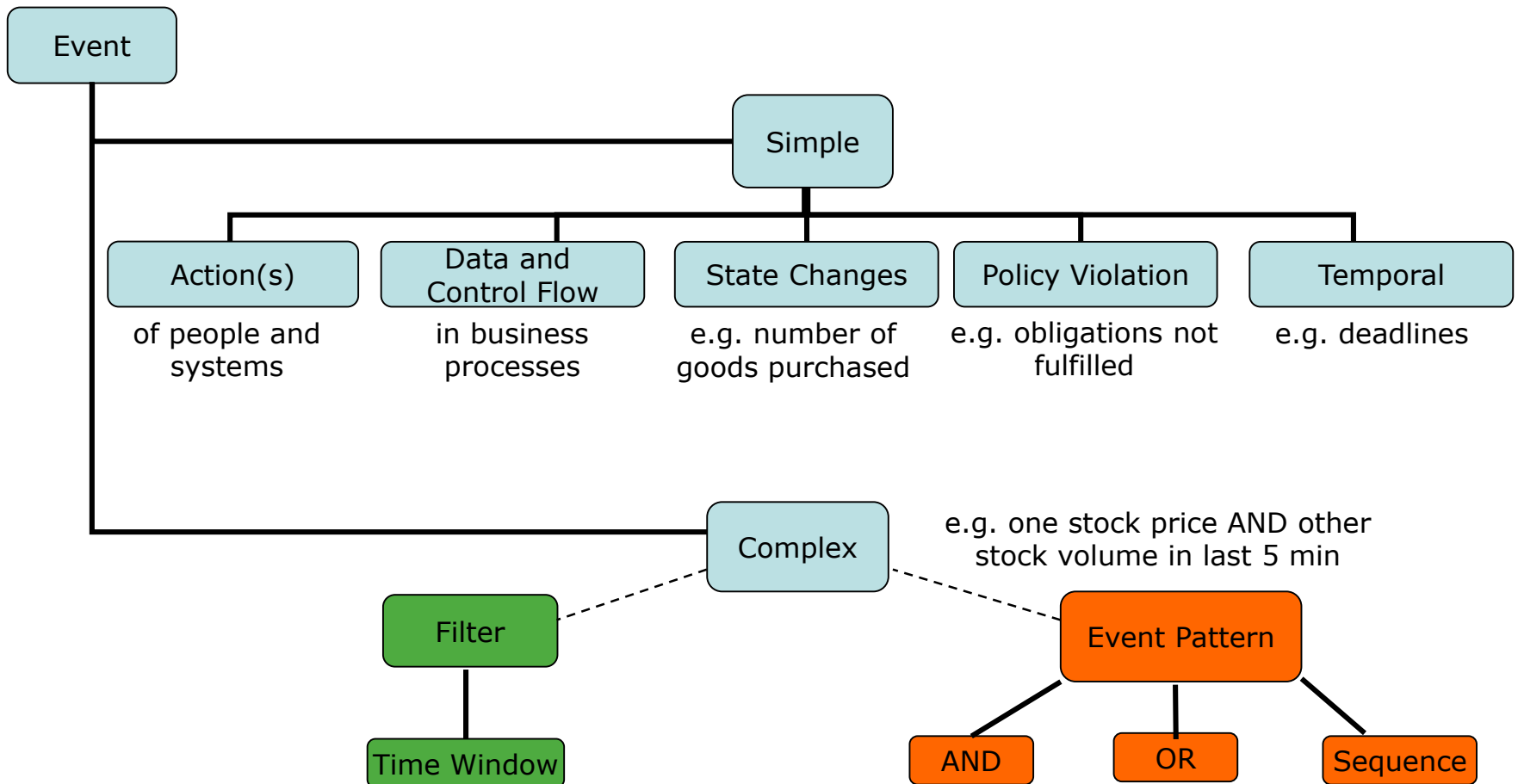


NEGATIVE CORRELATION
 • when more jobs are given the
number of people with flu falls.
 • flu jobs prevent flu.

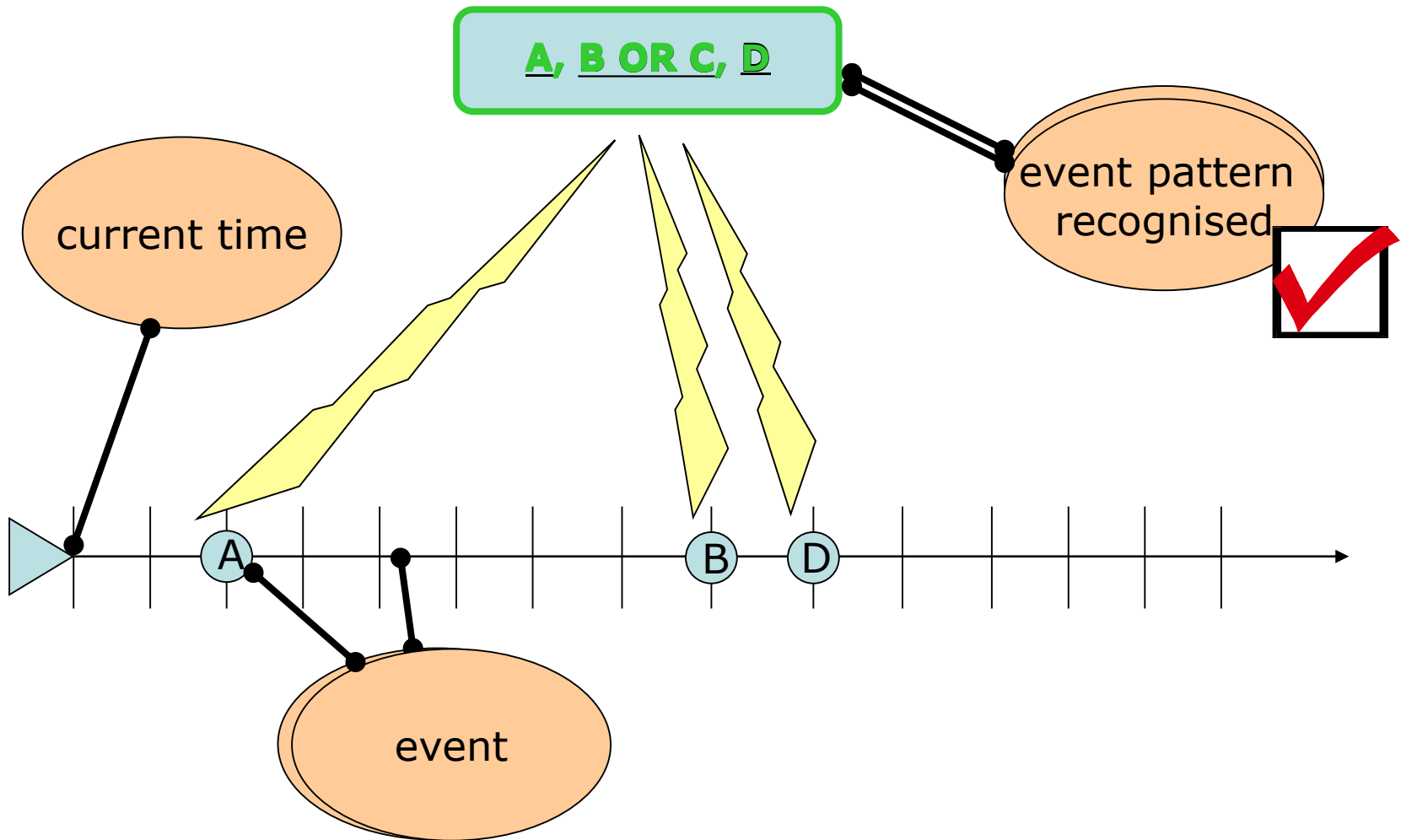
Event and Event Pattern

An event – *signifies an occurrence of interest*

Has structured content, and includes standard fields e.g. id and timestamp

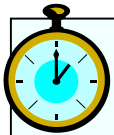
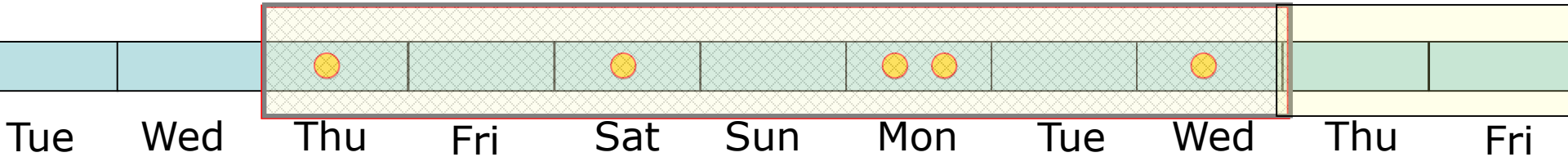


Event pattern processing



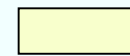
Time window example

Count(downtimeEvent): 5



Five sets of downtime in any seven days

- Width:
One Week
- Condition:
`Count(downtimeEvent) = 5`



Active Time Window



Window with condition found.



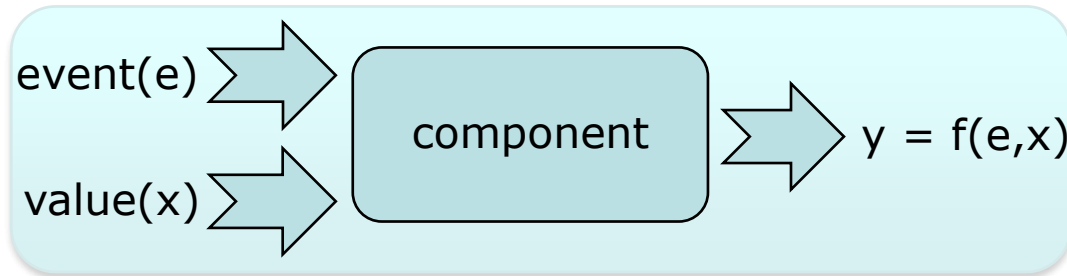
Downtime event



Event in match of window condition

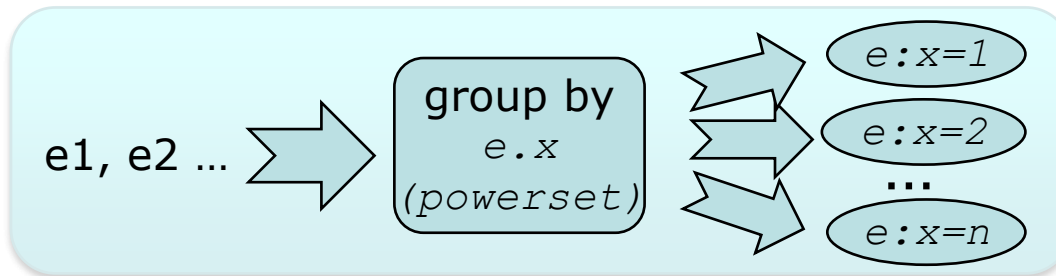
EventSwarm: distinguishing features

Event Driven



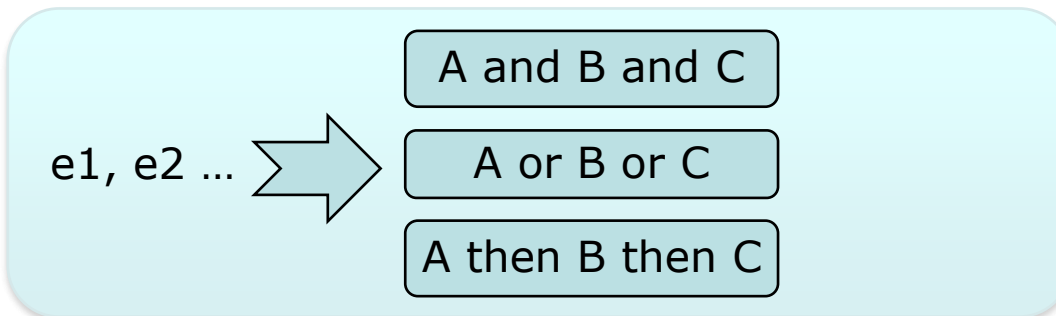
- Scalable
- Distributable
- Time consistent

Sets and Powersets



- Naturally Parallel
- Strong Semantics

Logically Compose

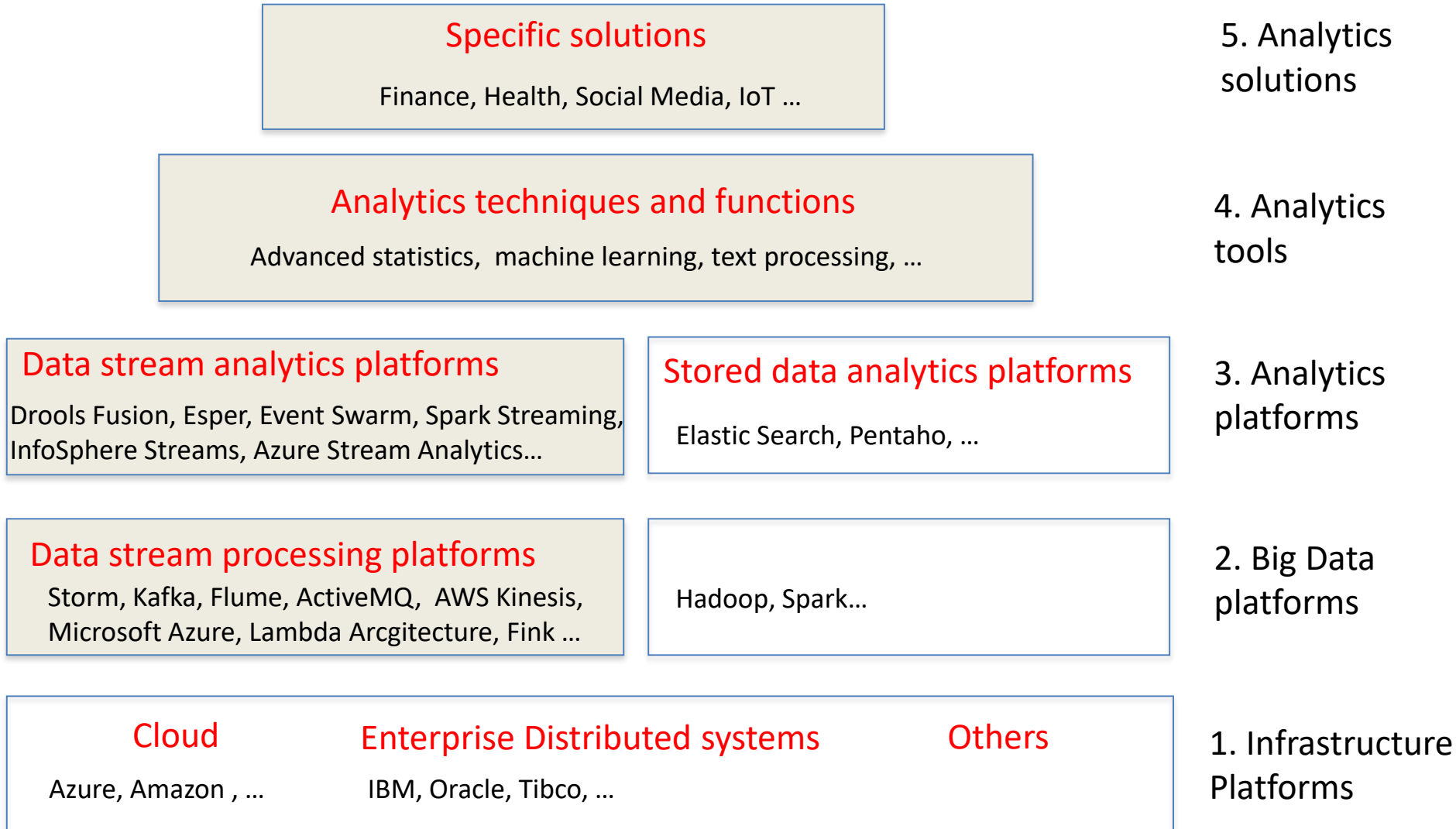


- Expressive

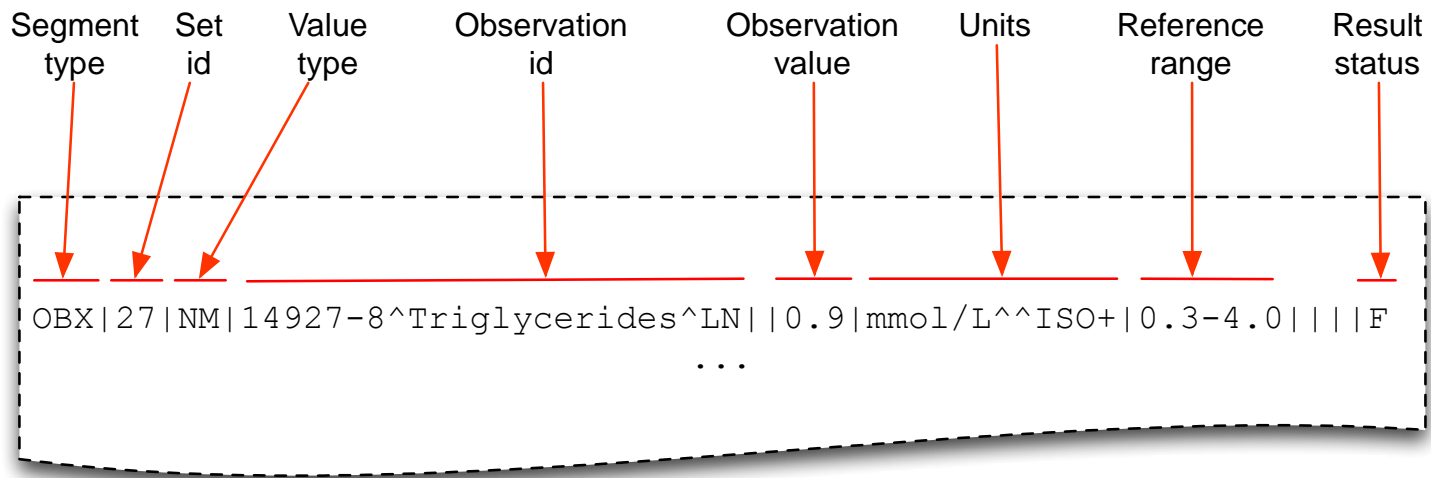
Implementation features

- Lightweight, in-memory, CEP library for Java
 - designed for embedding in applications, including mobile
- Components form a data-driven processing graph
 - graph segments can be distributed using any distribution infrastructure (e.g. Storm, Kafka)
- Latency, memory usage, distribution and threading
 - controlled through component selection and graph structuring
 - allows pushing your processing all the way out to the data sources - further improves latency

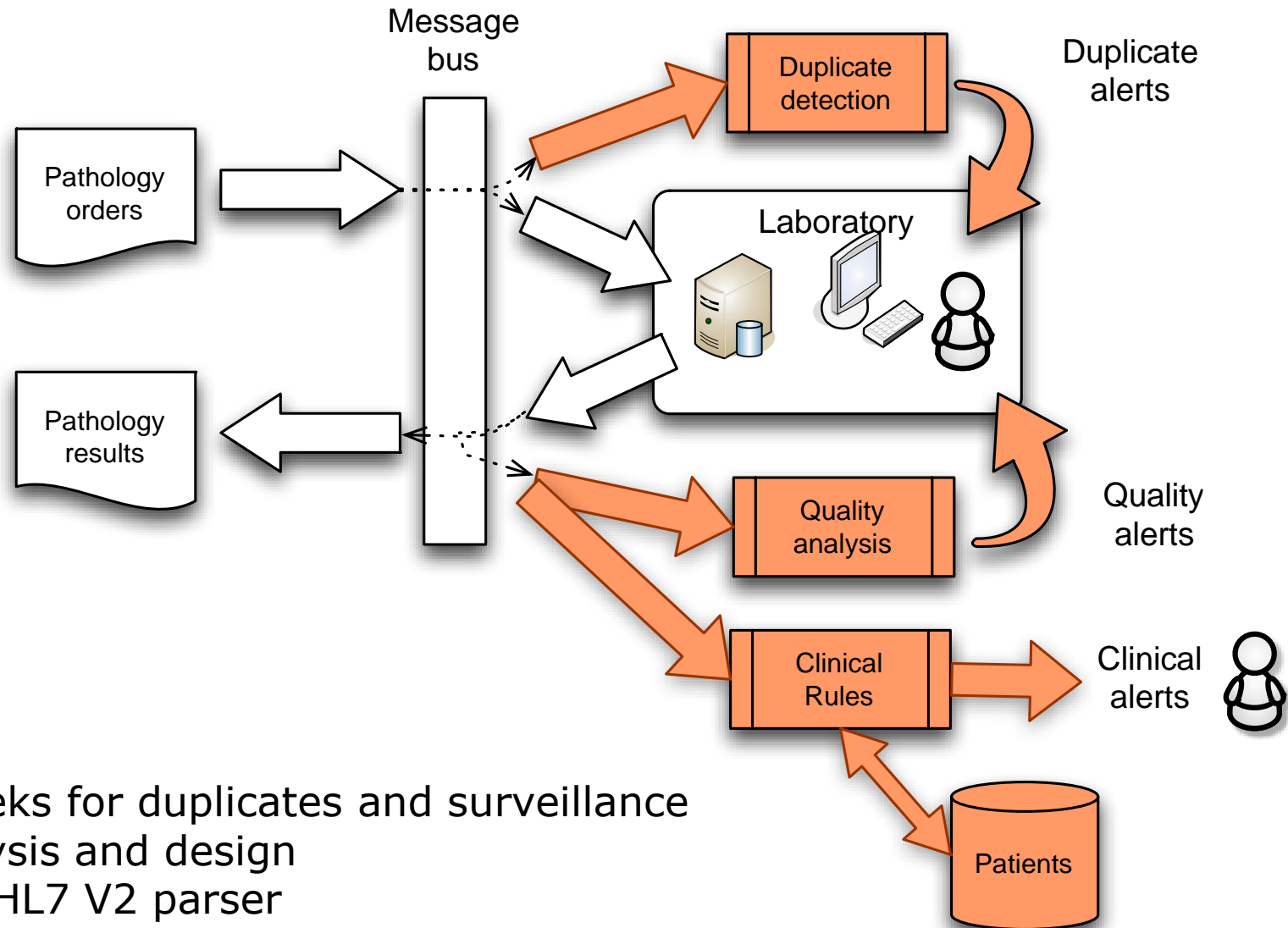
Event processing stack



HL7 message structure



EventSwarm implementation

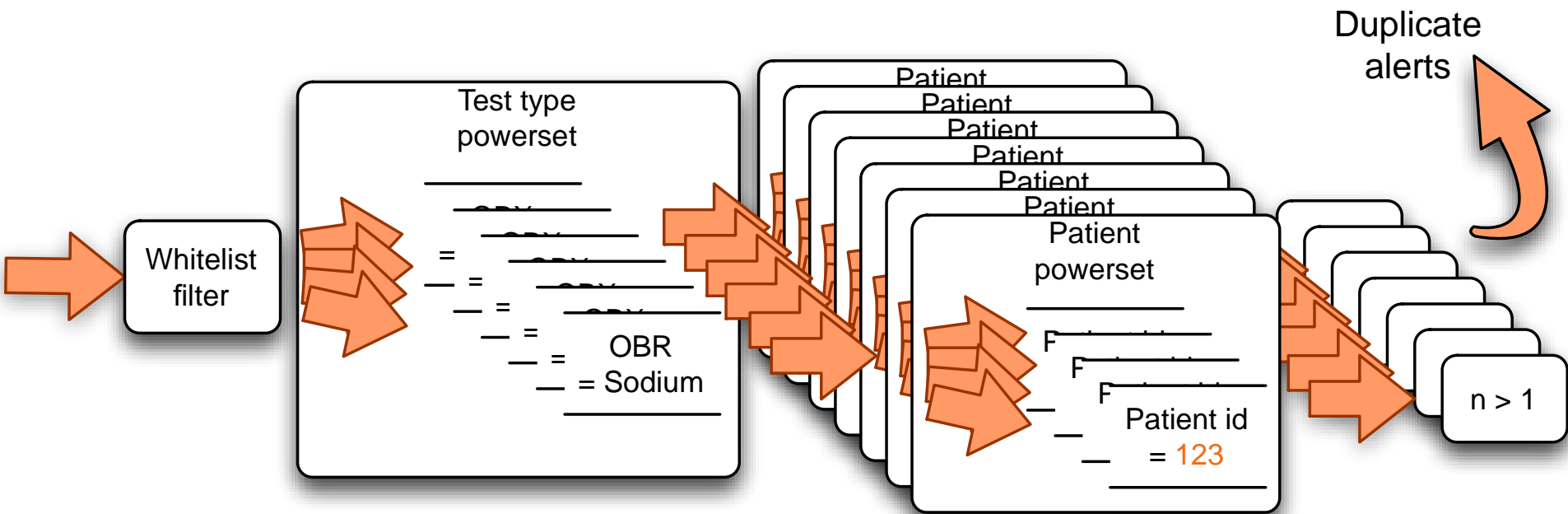


Two weeks for duplicates and surveillance

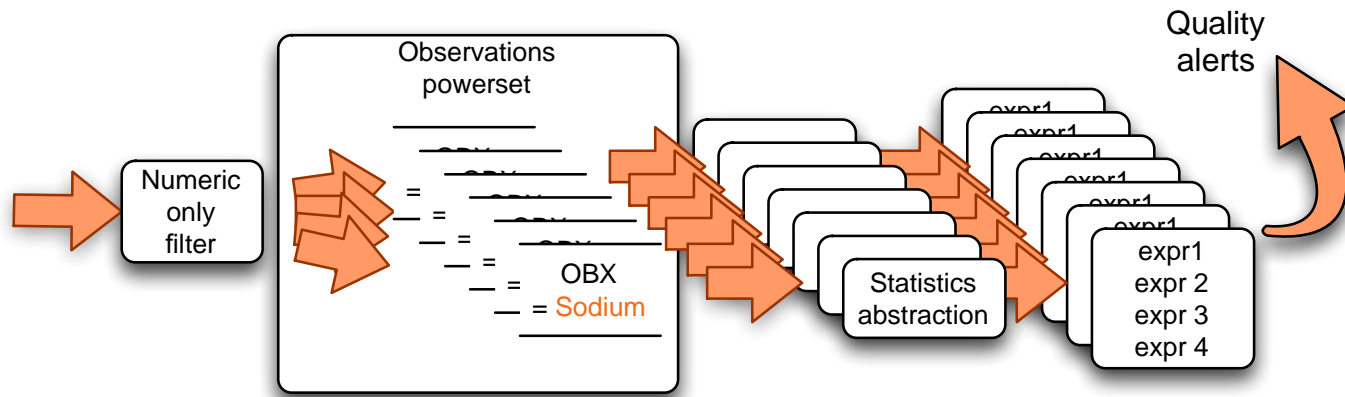
- Analysis and design
- Fast HL7 V2 parser

One week for clinical use cases

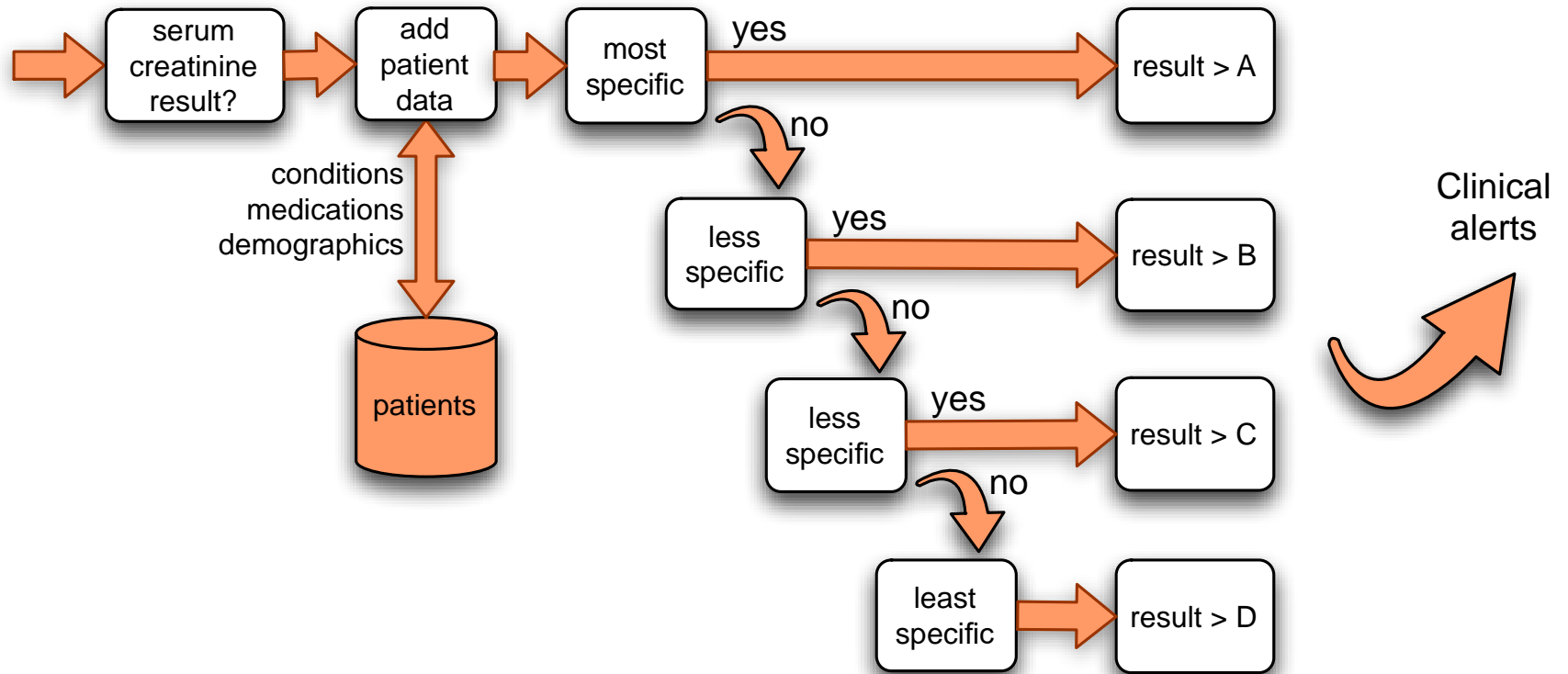
Duplicates



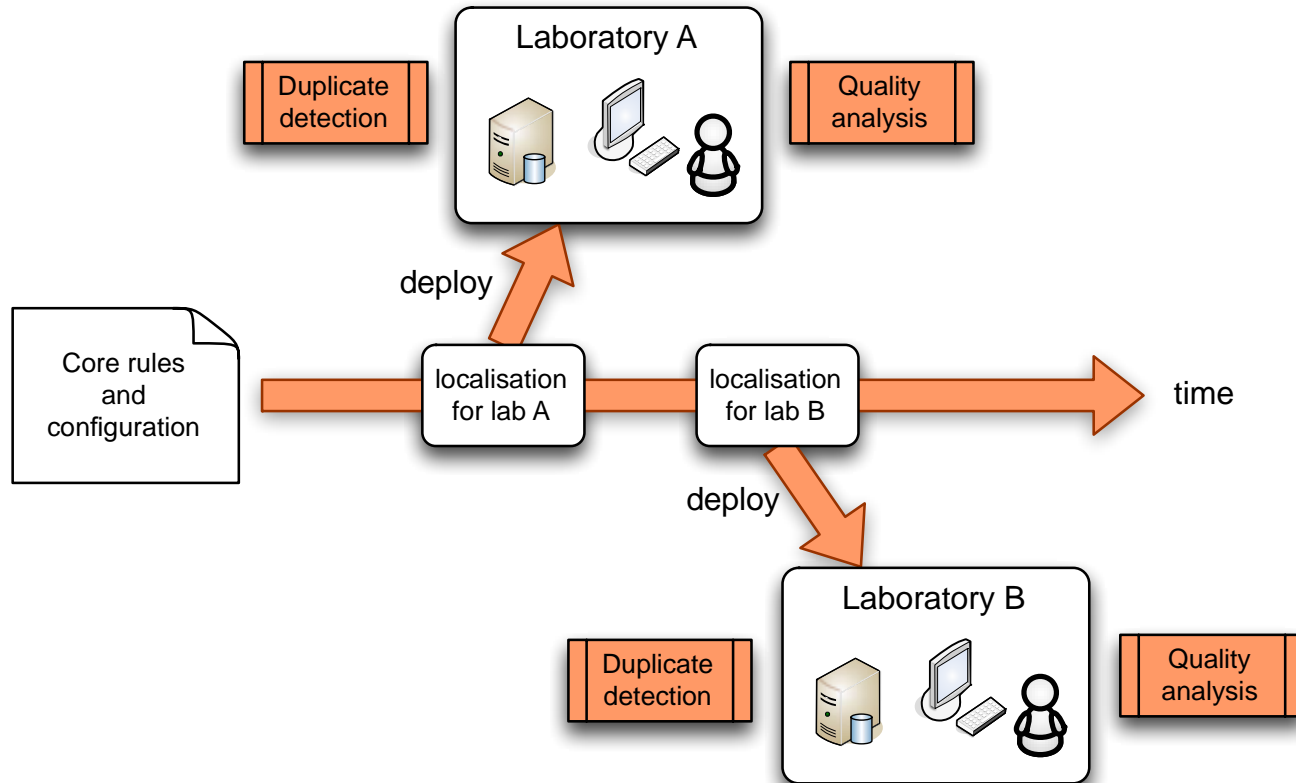
Data Quality



Clinical Alerts



Deployment - distributed



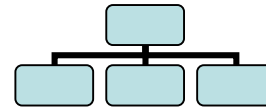
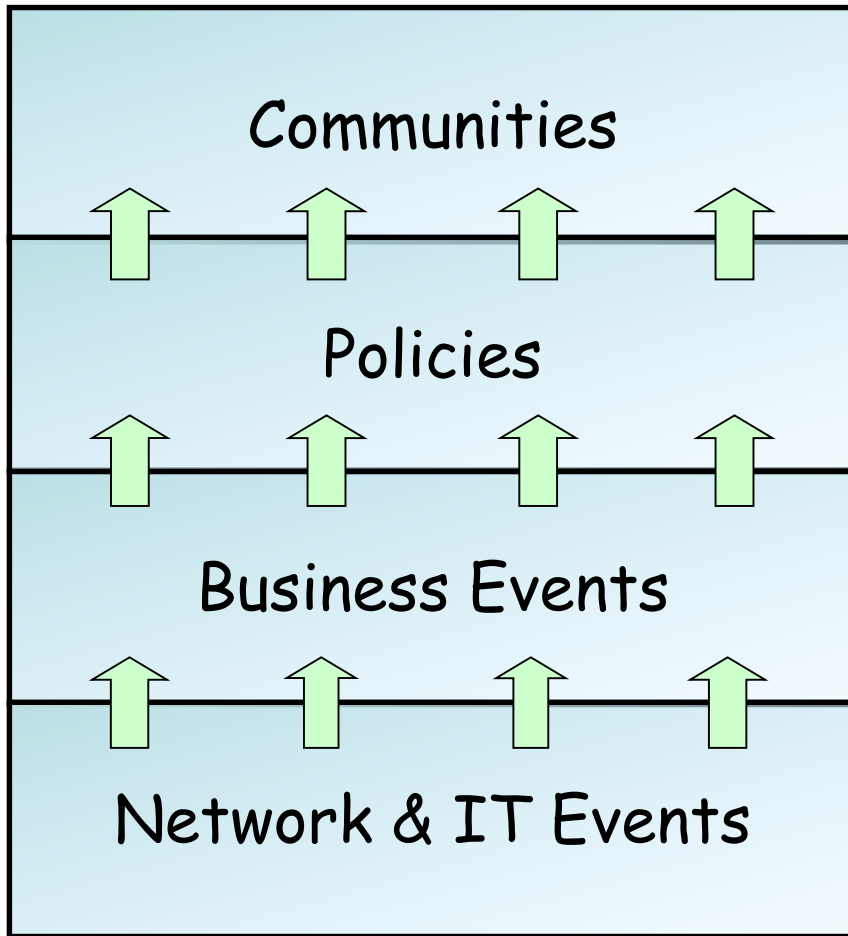
Horizontal scalability

High performance

Links to RegTech

- RegTech - use of IT for
 - Regulatory monitoring, reporting and compliance
- Today's approaches:
 - digitization of manual reporting and compliance,
 - e.g. know-your-customer
- Further benefits – real-time monitoring
 - faster detection of risks and more efficient compliance
 - much experience from our early research – contract monitoring
 - Business contract language (BCL) – use of event-patterns
 - “Compliance checking between business processes and business contracts”, widely cited paper
- Consider impact of advanced analytics, ML/AI, blockchain

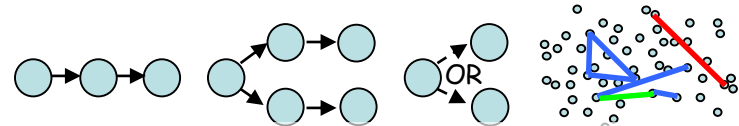
Monitoring Stack: real-time enterprise



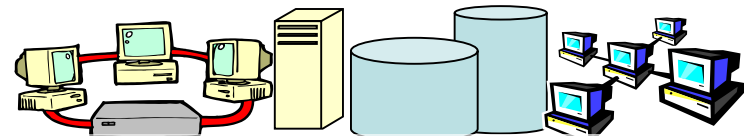
Assign/Revoke Roles, Policies, Fill Roles etc

"The supplier must ensure that all goods have been delivered within 10 days of receiving the purchase order"

Permissions, Prohibitions and Obligations



Event Patterns: sequences, parallel, alt, correlation/causality



Programs, Databases, Network, Legacy, Middleware

Policies & Events

The parties **must not** *[divulge the financial details]* of this contract to any third parties

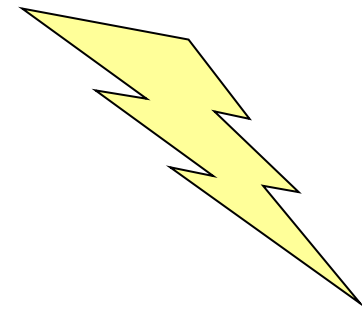
The supplier **must** ensure that all *[goods have been delivered]* *[within 10 days of receiving the purchase order]*

The Purchasing director is **entitled to** *[cancel an order]* if *[written notice is given]* within *[24 hours of order lodgement]*.

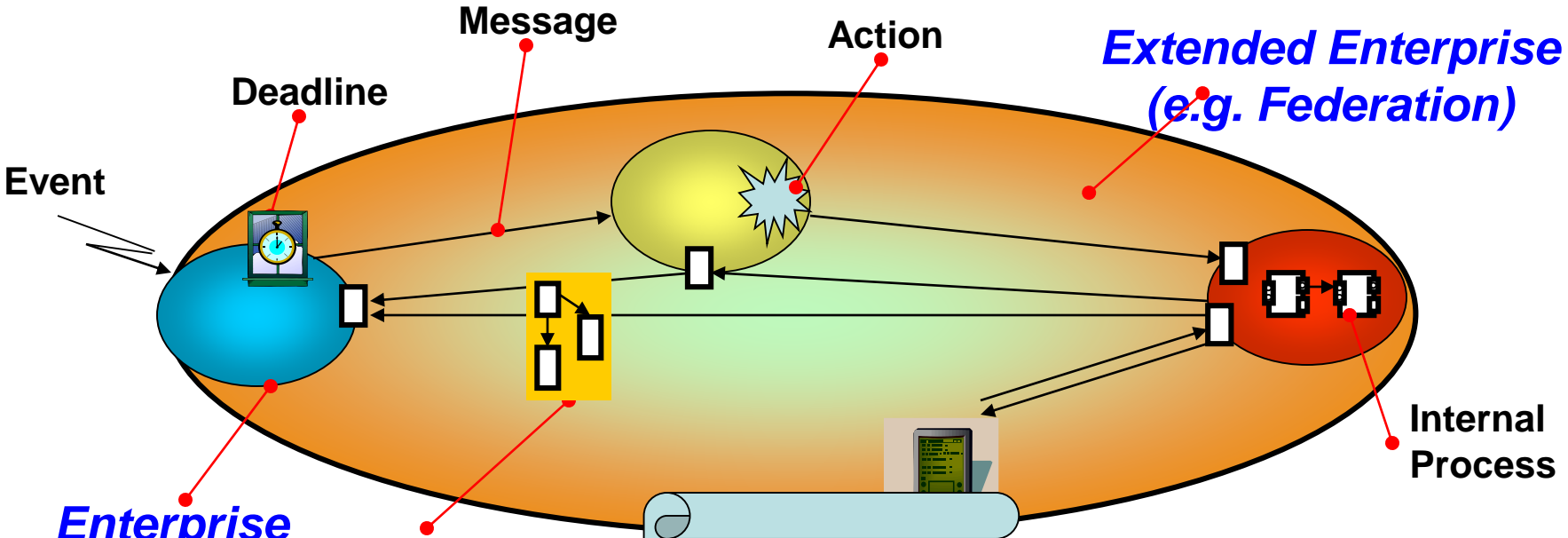
Policies are evaluated in response to events

Key:

[cancel an order]
- Event



Events, Policies, Community: powerful formalism

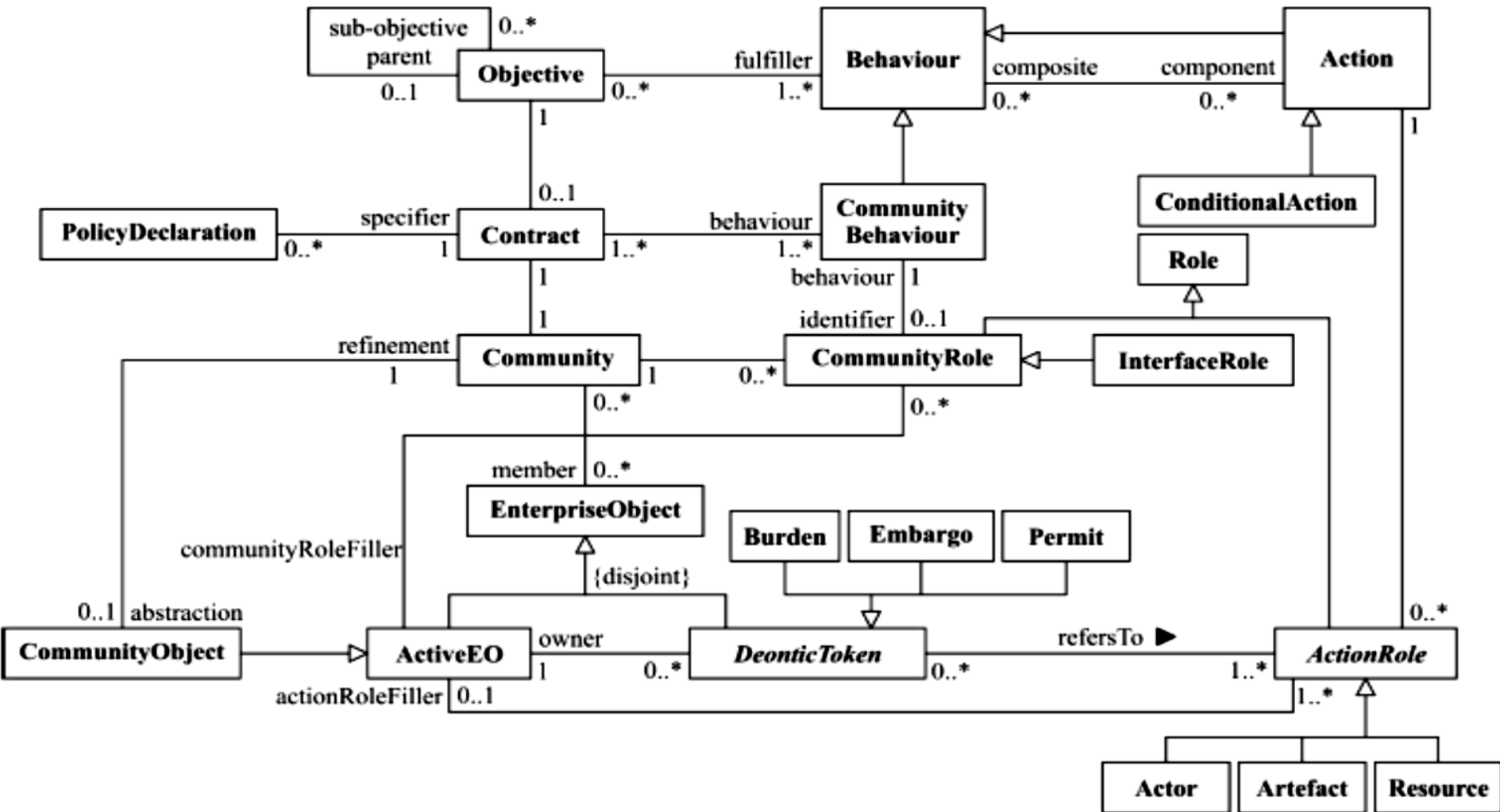


Obligations, Permissions,
Prohibitions, Violation measures ...

Context for Smart Contracts !

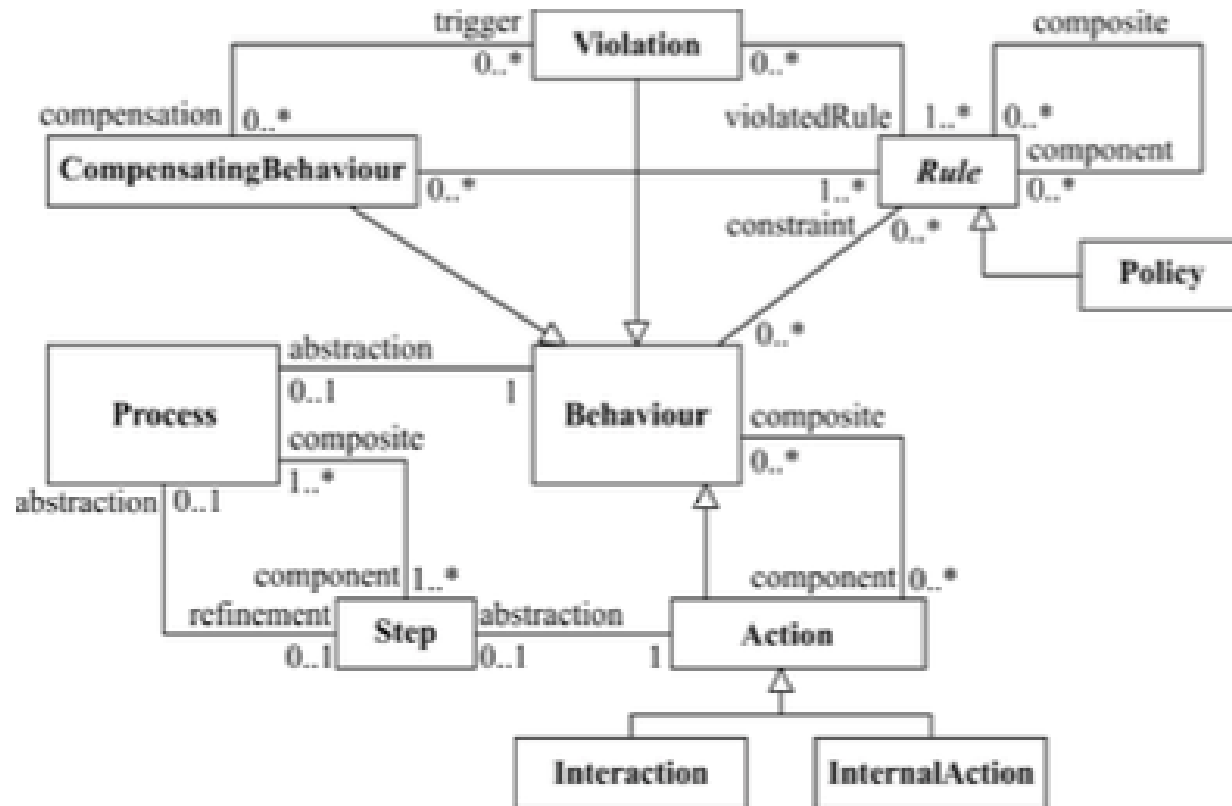
Community Model

RM-ODP enterprise language standard



Source: ISO/IEC 15414 -Information technology – Open distributed processing – Reference model – Enterprise language

Policy and behaviour



Contracts

- Legal contracts
 - Deontic constrains
 - Roles, policies, context (community)
 - Computable expressions
 - Monitoring, reasoning, ...
 - Event-oriented expressions
- Smart contracts – blockchain
 - What is needed to ensure legal validity ?
 - Contract life cycle
 - Contract language – capture deontic semantics
 - New paper submitted - leverages our previous research
 - “On Legal Contracts, Imperative and Declarative Smart Contracts, and Blockchain Systems”

Governance

- Governance → an important challenge
 - Legislative rules and regulatory compliance,
 - Compliance to internal/contractual policies, accounting principles ...
- Implies defining a scope or a domain
 - where policies apply
 - where monitoring is required
 - where enforcement is needed
- Contract is a special kind of governance
 - risk minimization in the presence of uncertainty
 - SLAs are special kind of contract
- A framework for modelling governance is needed
 - including the related (cross-)organizational and policy constraints
 - our approach → **Community Model**
 - contexts for enterprise and security policies, interactions, information exchange, interoperability ...

Future work

- Digital health application
 - new rules, e.g. personal medical devices
 - social media correlation with medical data streams
 - incident detection and emergency management
- General CEP platform
 - enriching with support for deontic logic token objects
 - based on the recent ODP Enterprise Language standard
 - useful for compliance and RegTech applications
 - AI/ML topics - prediction after event pattern detection
 - Long Short Term Memory (LSTM)
 - automated learning of CEP rules
 - porting to recent Lambda architecture
 - Formalizing EventSwarm concepts in Maude
- **Collaborations proposals welcome !**
 - Student projects etc.

Demo and Questions
