



J-Fall

11 november 2009 Spant!



Event Driven Architecture with GlassFish ESB

Eugene Bogaart
Solution Architect
Sun Microsystems



Objectives

- Introduce background on **event-driven architecture** (EDA) and Complex Event Processing (CEP)
- Overview GlassFish ESB Intelligent Event Processor (IEP) service engine
- **Demonstrate** a simple, but interesting, event processing application
- Jumpstart you, get you started!



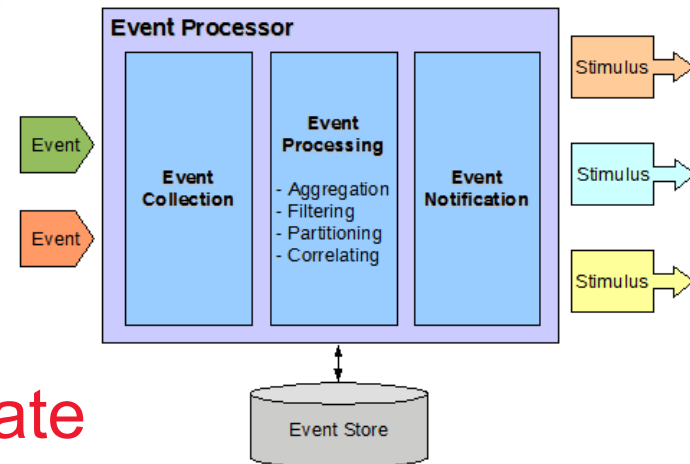
Agenda

- Event-Driven Architecture
 - Fundamentals
 - Theory
- Demonstration
- Summary



Event-Driven Architecture Fundamentals (Terminology)

- EDA is a **software architectural pattern** focusing on **producing, detecting, consuming and acting upon events**
- Event is a **significant change in state** of something of interest
- Three event processing styles:
 - Simple: **events directly related to specific change of state**
 - Event stream: **events screened to identify important ones**
 - Complex: **events analyzed for patterns and inferencing (casual, temporal, spatial) determines decision-impelling situations**

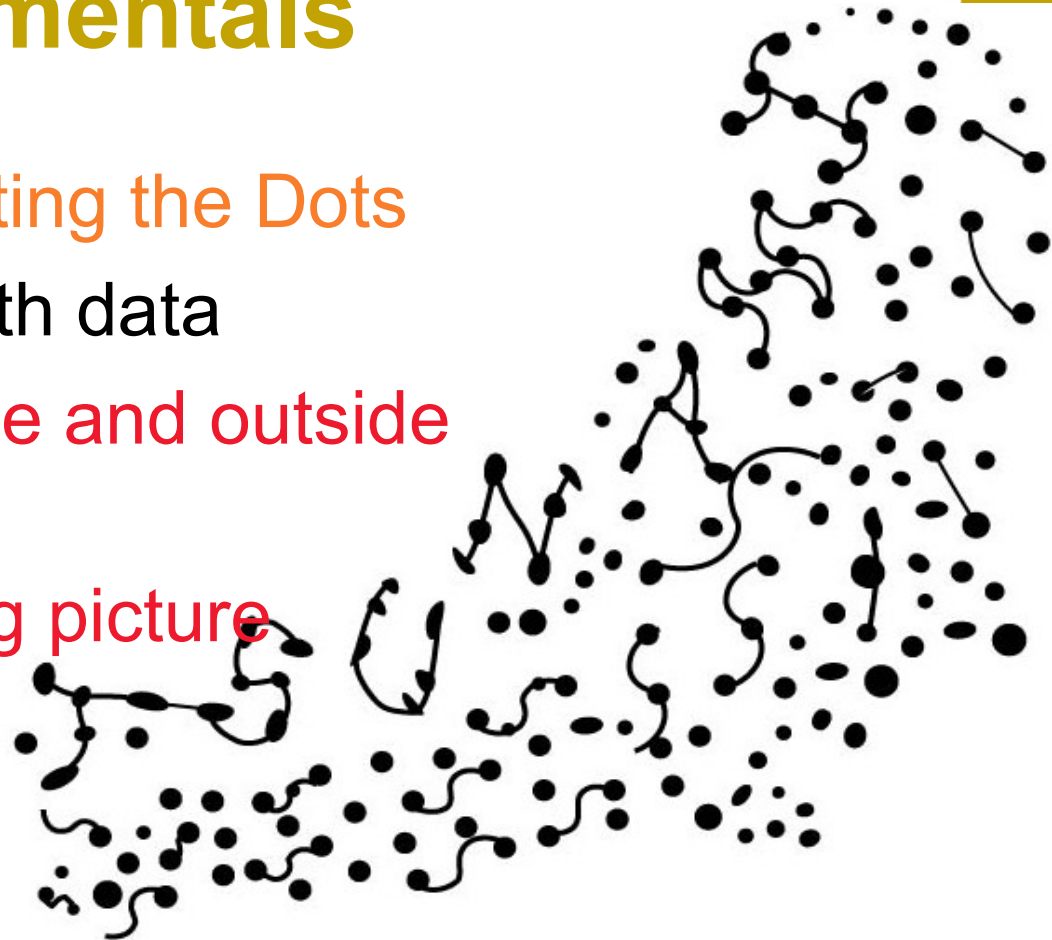




Event-Driven Architecture Fundamentals

Inspiration: Connecting the Dots

- We are flooded with data
- Must monitor **inside and outside** the enterprise
- Understand the **big picture**
 - **Threats** and
 - **opportunities,**
 - **context**
- We **need all the dots** but need help in connecting the dots

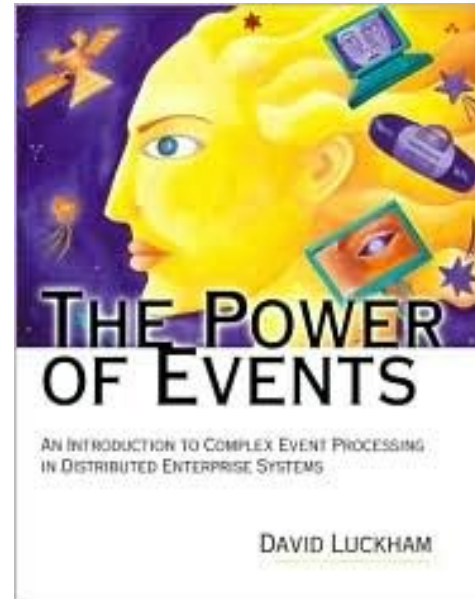




Event-Driven Architecture Fundamentals

Terminology: CEP History

- Developed in the **early 90s** to analyze real-time events in distributed systems (air traffic control, radar tracking, telco networks, electronic markets)
- David Luckham, “Father of CEP”
 - **Stanford electrical engineering professor (Emeritus)**
 - **Co-founder of Rational Software**





Event-Driven Architecture Fundamentals

Terminology: CEP and ESP

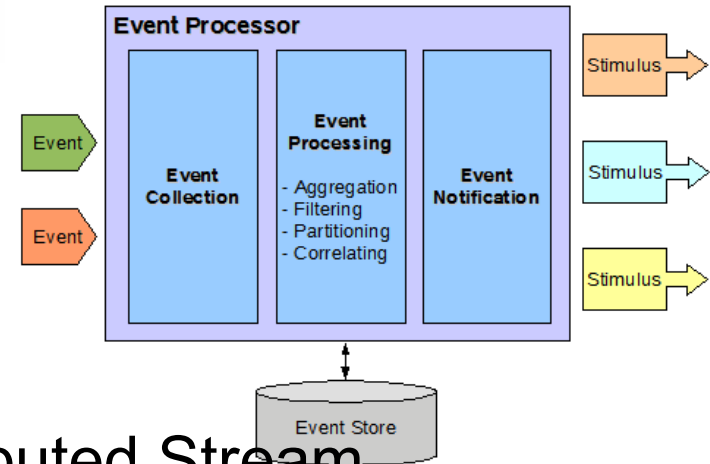
- ESP focuses on **streams**
 - Sequence of events ordered by time
 - Simpler algorithms as data is analyzed as received, analyzed, passed along and forgotten (Prototypical examples: Stock market feeds, RFID)
- CEP focuses on **clouds**
 - Collection of events generated at different places
 - Cloud could contain many streams
 - Events don't arrive in order
 - Focus on detecting patterns in independent events
- Some consider ESP a subset of CEP,
 - but domains merge as ESP vendors evolve



Intelligent Event Processor Overview

DBMS vs. DSMS Perspectives

- Focus on **data streams (no tables)**
- Example domains: network monitoring, sensors, telecom call records, financial markets
- Distributed Data Base Management Systems
 - Persistent data
 - One-time queries
 - Random access
 - Access plan determined by queue analyzer
 - Implemented by SQL



- Distributed Stream Management Systems
 - Transient and persistent data
 - Continuous queries
 - Sequential access
 - Unpredictable data characteristics and arrival patterns
 - Implemented by CQL



Event-Driven Architecture Fundamentals

Comparing EDA and SOA

- Event-Driven SOA

- Notable thing inside or outside enterprise happens
- Infrastructure detects state change & issues event
- Event initiates one or more SOA services

- SOA Service as Event Generator

- Service generates an event signaling a problem, opportunity or notable deviation
- Infrastructure delivers event to event infrastructure
- Event is propagated to all subscribers

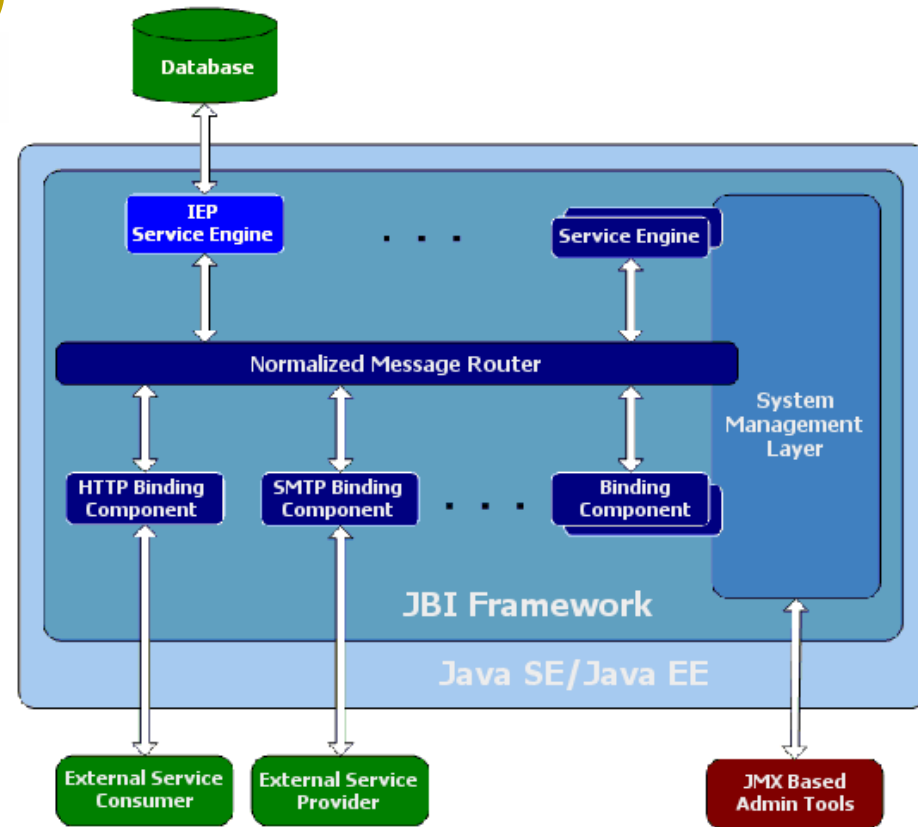
“SOA and EDA are peers and complements. I disagree with the SOA evangelists who say EDA is merely a subset (child) of SOA.”

- Brenda Michelson  Patricia Seybold Group
Trusted Advisors to Customer-Centric Executives



Intelligent Event Processor Big Picture (I)

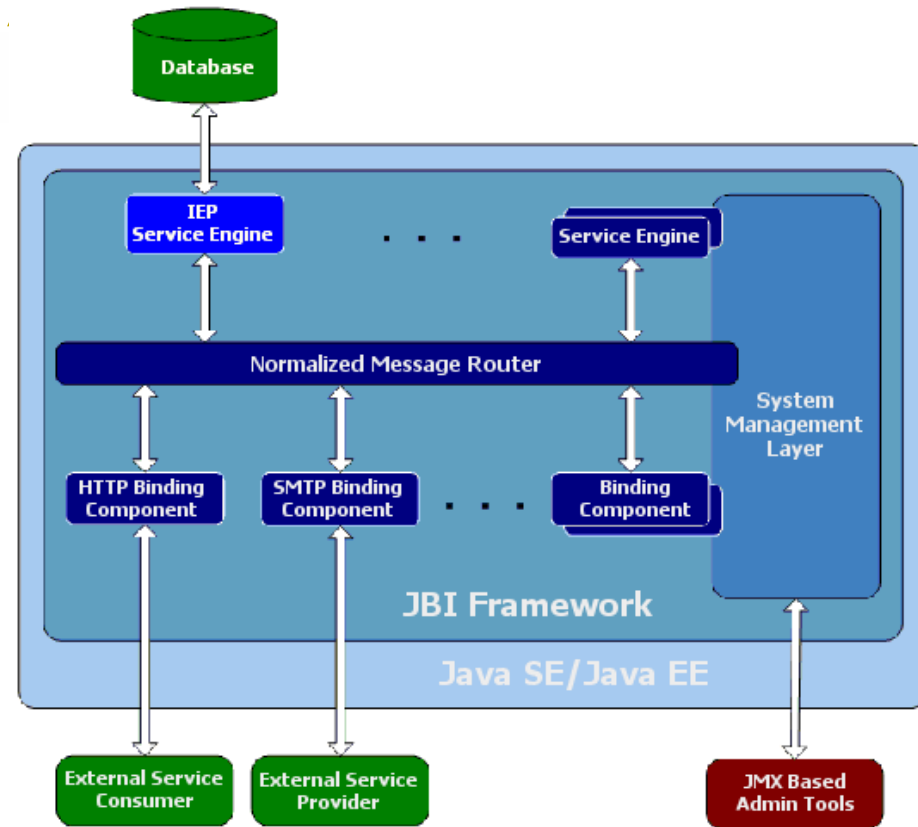
- Real time **business event**
 - Collection,
 - Processing
 - Aggregation,
 - Filtering, partitioning
 - and correlation
- **Event process editor**
- Real time business event notification and event triggers





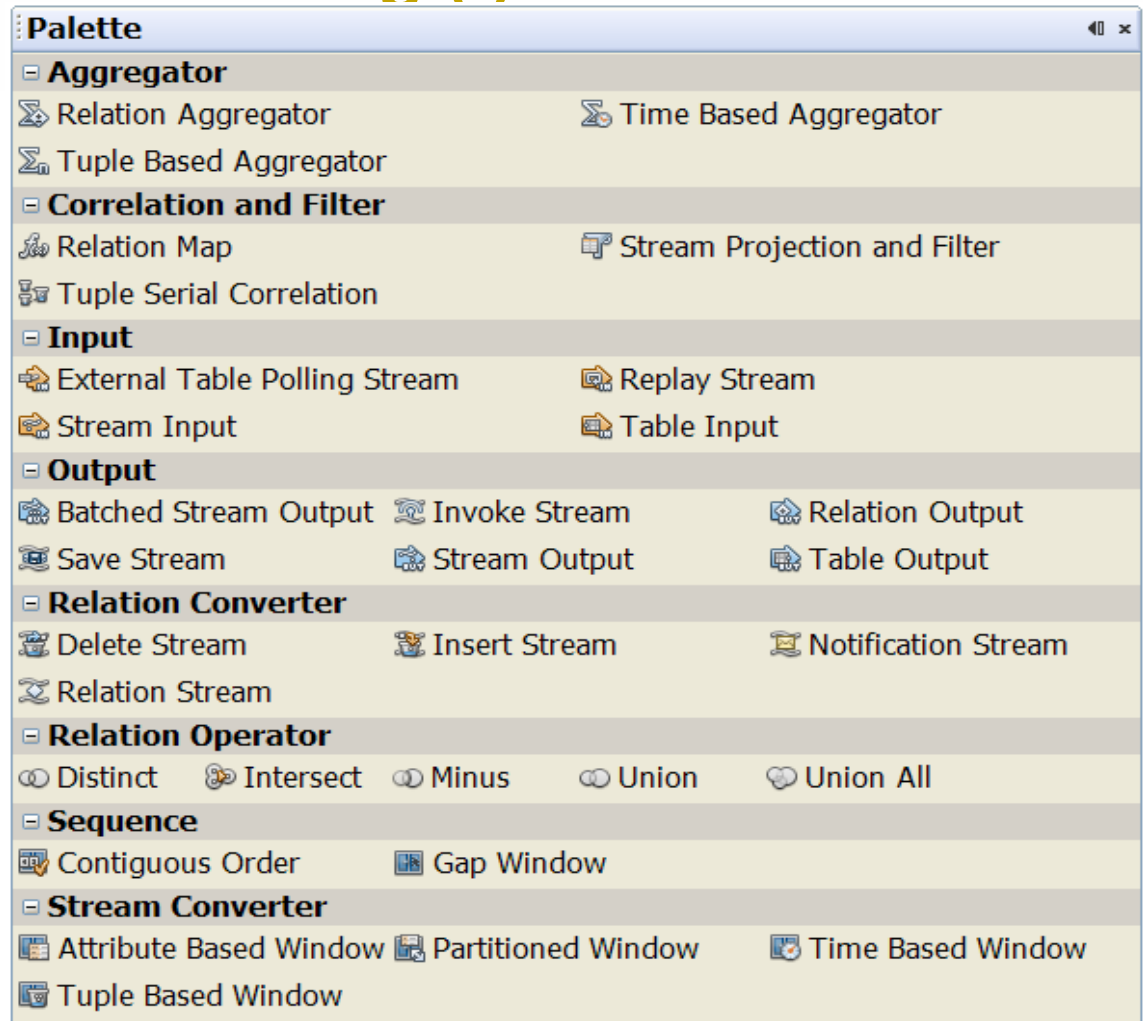
Intelligent Event Processor Big Picture (II)

- Event provider to other tools such as BI and reporting tools
- Runtime service engine based on Continuous Query Language (CQL)
- Framework to create visual representation of IEP based aggregated data like **dashboard**, charts and reports



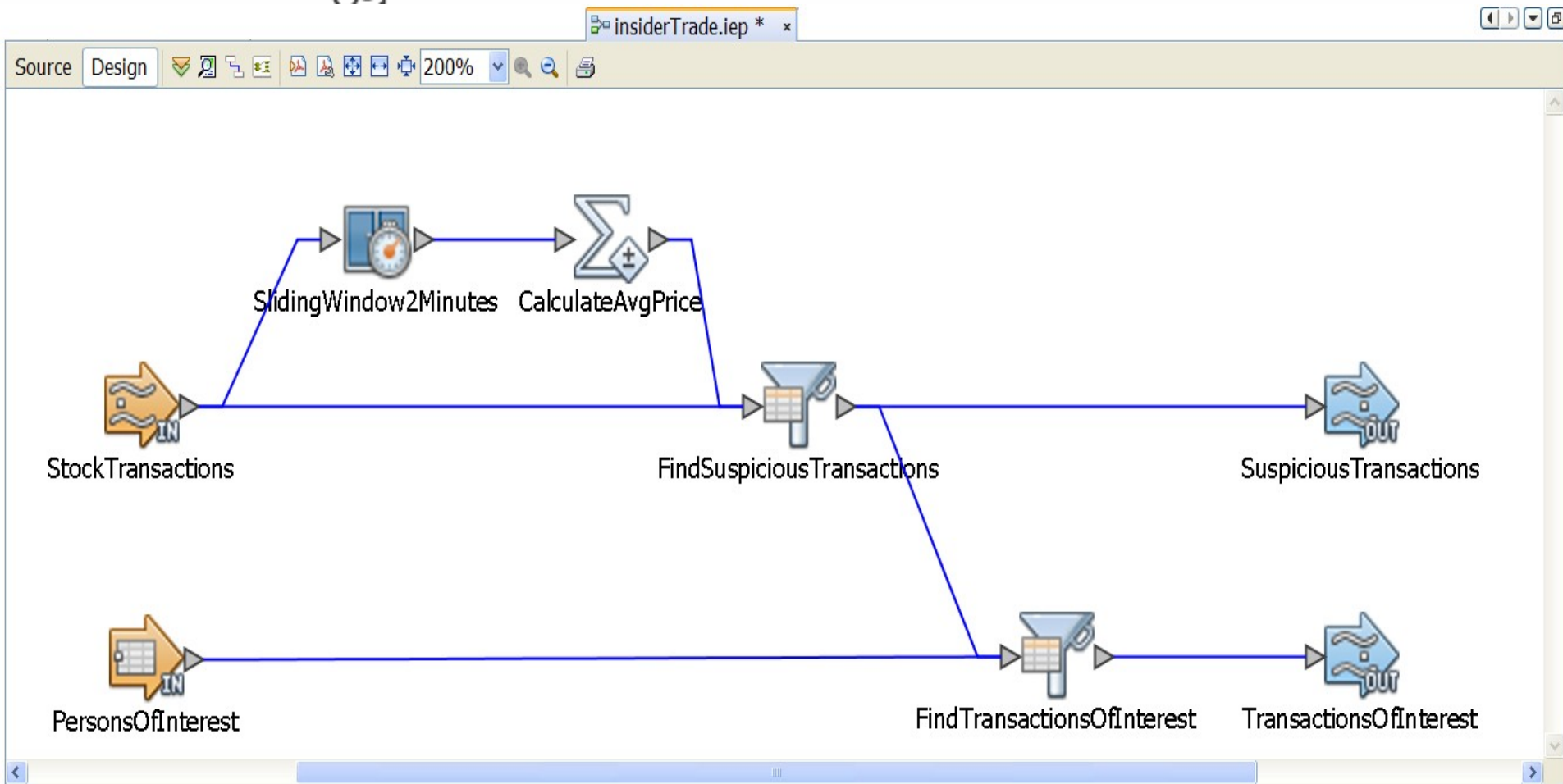


Intelligent Event Processor NetBeans Tooling (I)





Intelligent Event Processor NetBeans Tooling (II)





Demo Scenario Stock Fraud Detection



- Stock exchange interested in **suspicious trading activity**
- An example of “suspicious” is a trade whose price is significantly different
- “Significantly different” is defined as **price at least 10% above or below the prevailing price**
- If detected, an alert is to be generated
- For each alert, a matching operation is to be performed against a possible suspects table to produce a second – a known suspects notification



Demonstration



Demo Scenario

Lotto money laundry



- Penny Lotto is interested in **suspicious activity**
- An example of “suspicious” is a money deposit and withdrawal without significant activity
- “Significantly activity” is defined as **deposit, no betting and withdrawal within 24 hours**
- If detected an alert is to be generated
- Each alert needs to send to different end-points.



Summary



Summary

- Just SOA in your **toolkit** isn't enough for today's challenges
- Event-Driven Architecture (EDA) is an **orthogonal to SOA**
- **EDA** focused on producing, detecting, consuming and acting upon events
- Stock Fraud detection is a very simple **complex event processing** (CEP) demo based real life on a fraud detection scenario
- **GlassFish ESB IEP** is you friend for CEP.



Learning more



Learning more ...

- Glassfish ESB website
 - <https://glassfish.dev.java.net>
- Intelligent Event Processing website:
 - <https://open-esb.dev.java.net/IEPSE.html>
- IEP BluePrints
 - See NetBeans Samples
- IEP Tutorial
 - <http://wikis.sun.com/display/OpenESBTutor/Tom+Barrett%27s+Open+ESB+and+Mural+Tutorials>



Q & A Thanks

Eugene Bogaart
eugene.bogaart@sun.com