



ORACLE®

Complex Event Processing with Oracle

Faris Al-Badri, Oracle Switzerland
faris.al-badri@oracle.com

Agenda

- Oracle Event Processing Overview
- Continuous Query Language Concepts
- Tooling and Visualization
- Fast Data and Big Data
- References and Resources

Oracle Event Processing Overview

Streaming Event-Driven Architecture (EDA)

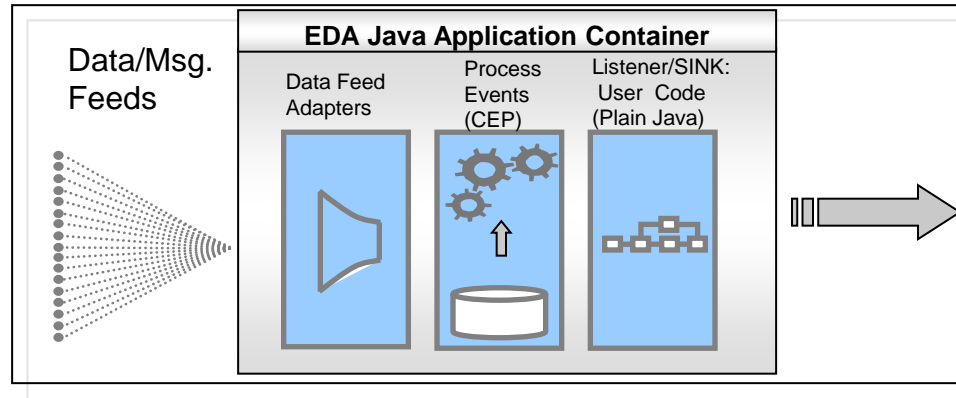
► Enriched Streams

- From any source: data streams, web services, Java, Database

► Adapters

- Translate external events/data into java objects for processing

Oracle Event Processing



► Processors

- Set of queries applied to the streams

► Listeners

- Handle triggers raised by the processors

► Events

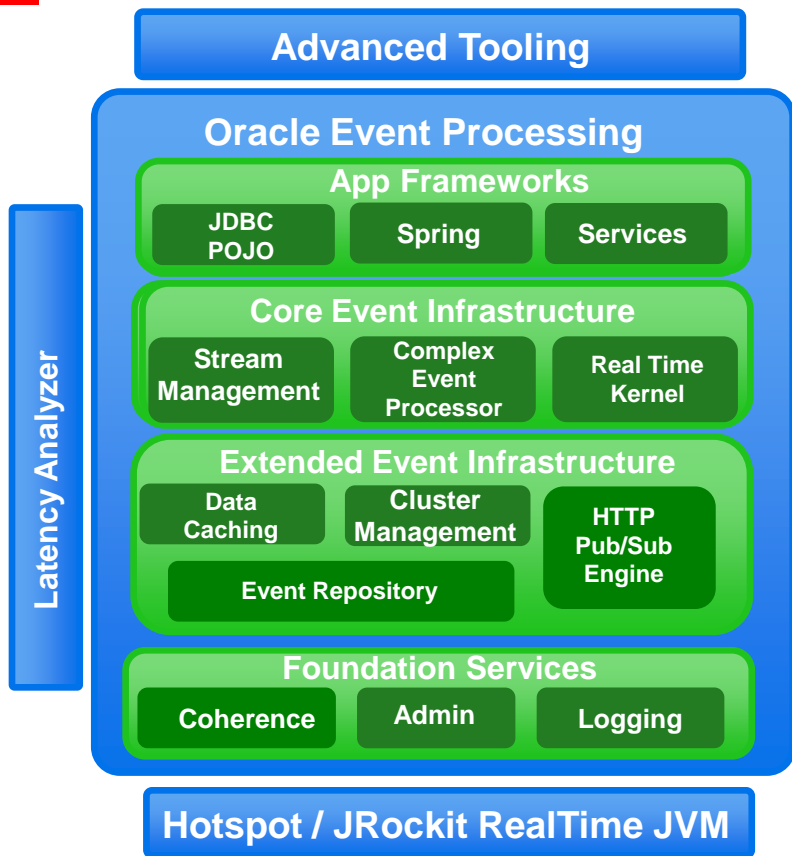
- Implemented as JavaBean or Map

Incoming Data Streams

Latency

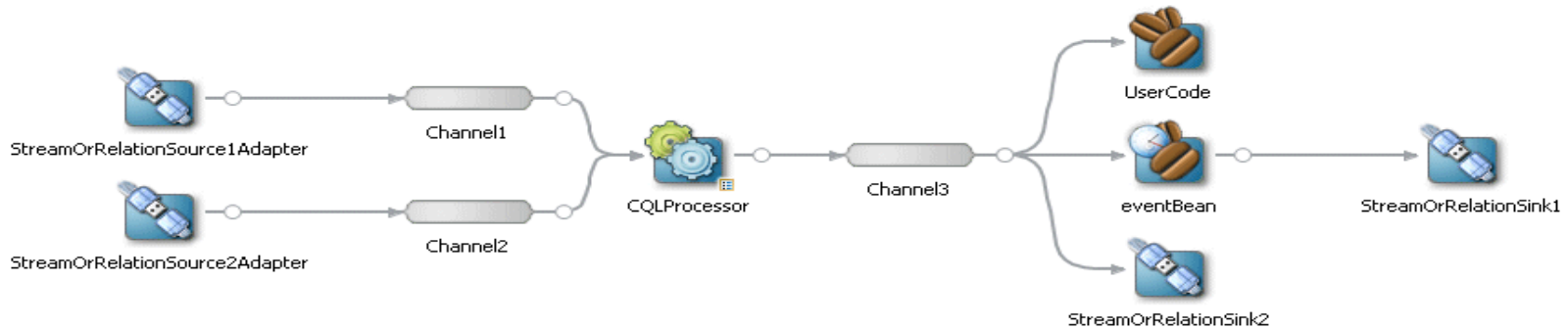
Resulting Data

OEP Product Architecture



- Lightweight Java Application Server (Jetty)
 - ▶ Full environment for building and running Java applications
 - ▶ Services -- Security, Logging, User Mgmt
- Designed for High Throughput
 - ▶ Hundreds of thousands of events/second
- Event Processing Infrastructure
 - ▶ CEP Engine
 - ▶ Event Processing programming constructs and services
 - ▶ Continuous Query Language
- Easy-to-Use Development Environment
 - ▶ OSGi, Spring Framework, POJO
- Eclipse-based Tooling
 - ▶ Plugins for EDA Application Development

OEP Application



Adapter

interfaces directly to the event stream source and sink, understands inbound/outbound protocol, converts event data into normalized form

Channel

event processing connection point, can queue data, multi-threadable

Processor

consumes normalized data from channel, executes CQL queries, may generate new events to output channel

Event Bean / User Code

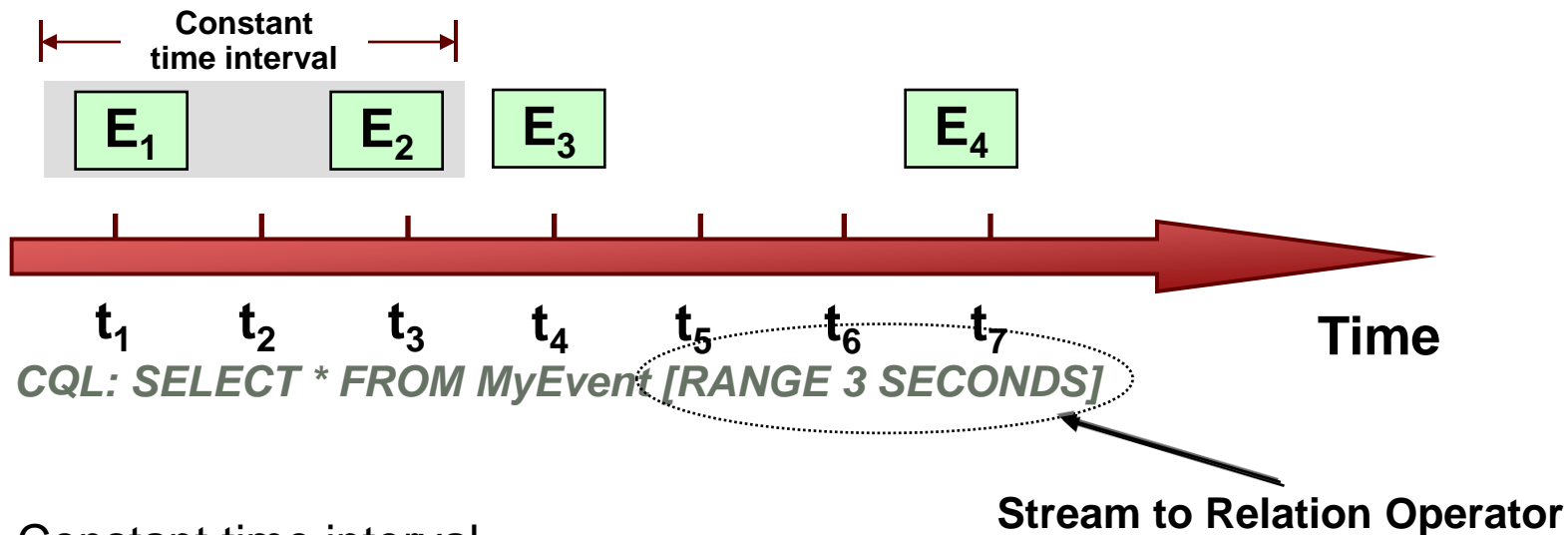
Java classes that register to listen to output channel, EventBean uses API to be managed by OEP container

Continuous Query Language Concepts

Continuous Query Language (CQL)

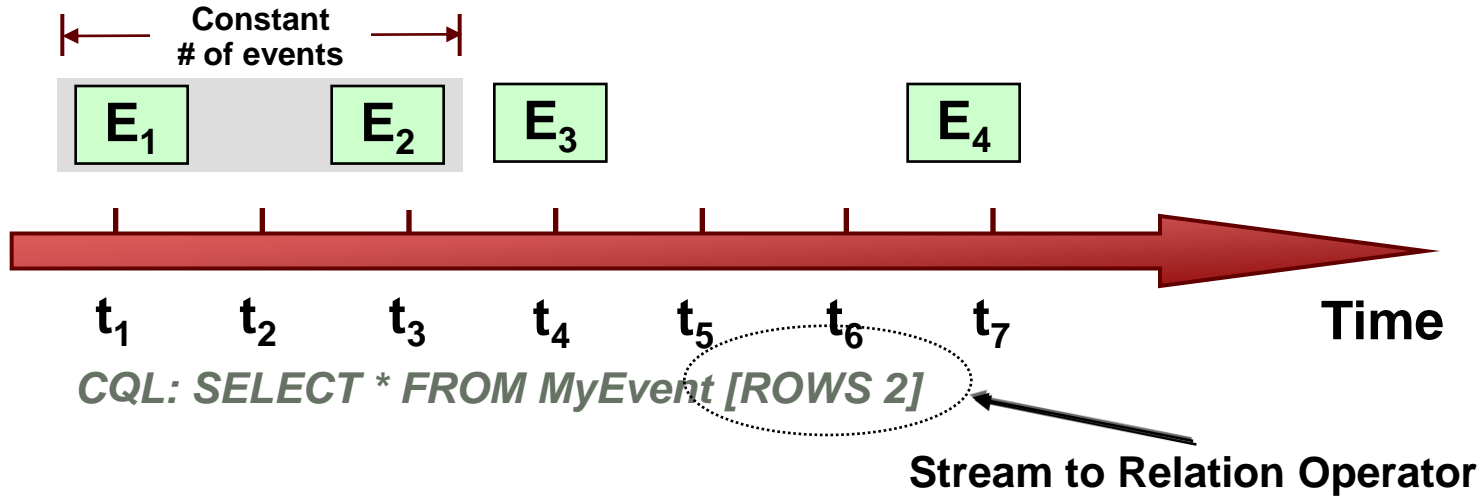
- Extends the relational model to support “continuous” queries
 - Formal model for describing Time, Streams, and Relations
 - Adds extensions to SQL-99
- Enables construction of “windows” over event streams
 - Bounded by time or count
 - Partitioned based on values
 - Processed incrementally or in batches
- Defines a “working set” of events to apply queries to
 - Filtering, aggregation, and correlation of events
- Adds pattern matching
 - Supports track-and-trace scenarios (e.g. detecting missing events)
 - accepted as ANSI SQL standard in 2012, implemented in Oracle Database 12c

Time-based event windows



- Constant time interval
- Number of events in Relation will vary
- Events will expire from Relation over time

Row-based event windows



- Constant number of events in Relation
- Events will not expire from Relation until replaced

Pattern Recognition with MATCH_RECOGNIZE

Example: Missing Event Detection

```
<query active="true" id="DetectMissingEvent">
  <![CDATA[
    SELECT
      "Event has not occurred in 5 seconds" AS detail,
      "ALERT ALERT" AS alertType,
      EventMiss.tStamp AS tStamp
    FROM RelayChannel MATCH_RECOGNIZE (

      MEASURES A.ELEMENT_TIME AS tStamp include timer events
      PATTERN( A B ) DURATION 5 seconds
      DEFINE
        A AS A.psi > 0,
        B AS B.psi > 0
    ) AS EventMiss
  ]]>
</query>
```

The **MEASURES** clause gives a name to the value in the events timestamp so it can be used in the select statement.

The **PATTERN** clause defines that you are looking for an 'A' event followed by the 'B' event which are defined in the **DEFINE** clause.

The **DURATION** clause defines how long in between events can occur.

Extending CQL with Data Cartridges

Allows to integrate other data types and functions to use within CQL:

Oracle Java Data Cartridge

Use Java types, methods, fields, and constructors in CQL queries

Oracle Spatial

An option for Oracle Database providing advanced spatial features to support high-end geographic information systems and location-enabled business intelligence solutions

Oracle JDBC Data Cartridge

Execute SQL queries against databases and use returned results in CQL queries

Oracle NoSQL Database Data Cartridge / Oracle Hadoop Data Cartridge

Integrate an existing NoSQL or Hadoop data source into an event processing network

Tooling and Visualization

Eclipse OEP App Development

The screenshot shows the Eclipse IDE interface for developing an Oracle Event Processing (OEP) application. The Package Explorer on the left displays the project structure for 'oracle.cep.sample.signalgen', including source files and build artifacts. The main editor shows a flow diagram with the following components and connections:

- socketAdapter (Input)
- signalgenerationProcessor (Central Processor)
- AlgoTradingBean (Output)
- cache1 (Output)

The Servers view shows the 'Oracle CEP v11' server is started and synchronized. The Console view displays performance metrics for three runs:

Throughput (msg per second)	Average latency (microseconds)
9873	71
10431	66
10370	64
9373	71

Monitoring and Managing OEP Applications

The screenshot displays the Oracle CEP Visualizer web application. The browser window title is "D:\weblogic\dev\wlevs_cq\modules\visualizer\flex\bin-debug\Visualizer.html - Windows Internet Explorer". The address bar shows the file path: "file:///D:/weblogic/dev/wlevs_cq/modules/visualizer/flex/bin-debug/Visualizer.html#app=11cb8f961-selectedIndex=18&1a0".

The application interface includes a navigation menu with "Home", "Security", "Dashboard", "ViewStream", and "Logout". The main content area is titled "Processor: cqIProc @NonClusteredServer" and features tabs for "General", "Record", "Playback", "Query Wizard", "CQLRules", and "Query Plan".

The "Query Plan" tab is active, showing a flow diagram with the following stages: 1:Source, 2:Window, 3:Source, 4:Window, 5:Join, 6:Filter, 7:Select, 8:Aggregate, and 9:Output. The diagram shows two parallel paths from sources 1 and 2, and 3 and 4, both leading to a join stage (5). The flow then proceeds through filter (6), select (7), aggregate (8), and finally output (9).

On the right side, there is a "CQL Constructs" panel with "Templates". It lists three templates: "Join Template", "Pattern Match Template", and "View1 Template". The "Join Template" is currently selected and highlighted.

The bottom of the screen shows the Windows taskbar with the "start" button, several application icons, and the system tray displaying "Internet" and "100%" zoom level. The system clock shows "12:45 PM".

Visualizing Events: Oracle Business Activity Monitoring

➤ Monitor business processes & services in real-time

- Key Performance Indicators (KPIs)
- Service-Level Agreements (SLAs)

➤ Analyze events as they occur

- Correlate events & KPIs
- Identify trends as they emerge
- Alert users to bottlenecks & solutions

➤ Act on current conditions

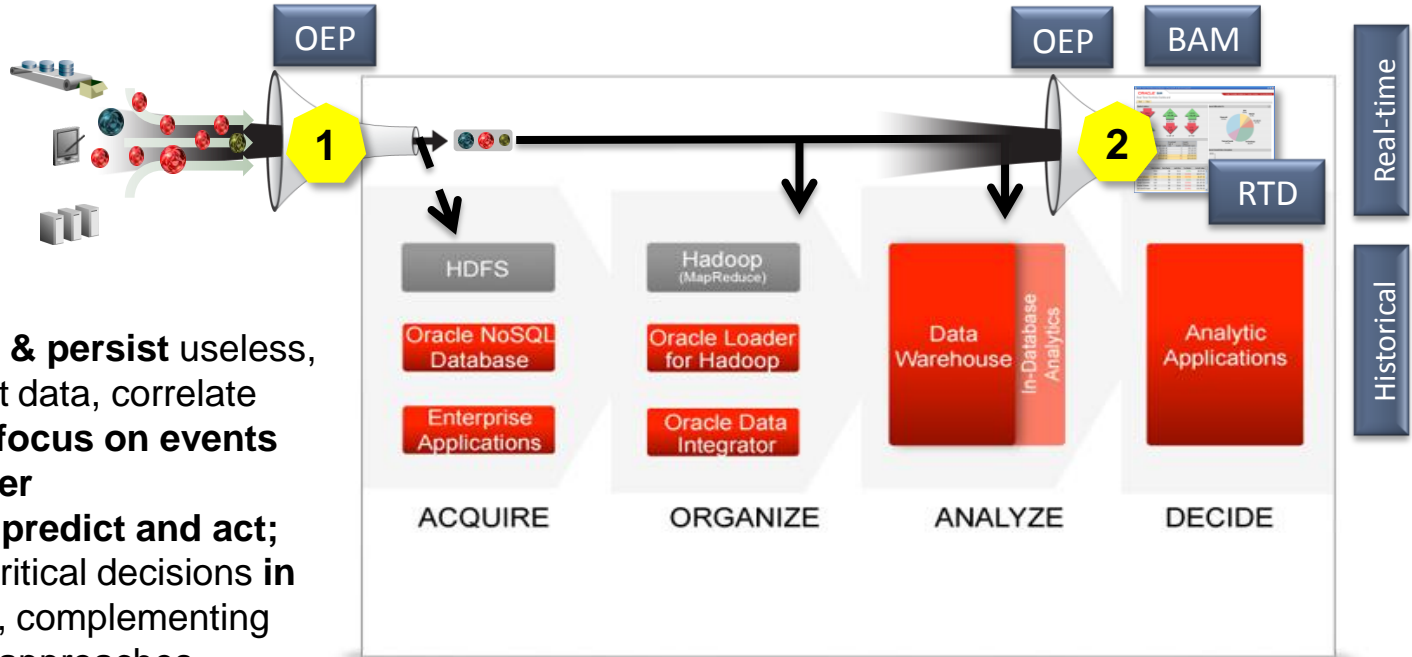
- Event-driven alerts
- Real-time dashboards
- BPEL processes & web services integration



Fast Data and Big Data

Fast Data & Big Data

Adding real-time value/capabilities with OEP, BAM* & RTD**



1 Filter out & persist useless, redundant data, correlate find and focus on events that matter

2 Analyze, predict and act; execute critical decisions in real-time, complementing historical approaches

*BAM: Oracle Business Activity Monitoring **RTD: Oracle Real-Time Decisions

References and Resources

Customer Adoption of Oracle's Fast Data

NTT
docomo



FedEx



The background of the slide is an abstract digital composition. It features a central horizontal band of dark grey, which serves as a backdrop for the text. Above and below this band, there are dynamic, flowing patterns of light. These patterns consist of numerous thin, overlapping lines in shades of deep blue, vibrant red, and bright white, creating a sense of motion and energy, reminiscent of data streams or light trails in a futuristic setting.

Where to Get More Information?

www.oracle.com/fastdata

Hardware and Software

ORACLE®

Engineered to Work Together