



Electronics and Telecommunications
Research Institute

Introduction to Complex Event Processing

CEP with Esper, Real-time Event Stream

18 March 2014

Sung-Soo Kim

sungsoo@etri.re.kr

ETRI

References and Slides Credit

2

1. 원종석, 오픈 소스로 구현하는 Real-time CEP, Tedwon 데이터지능팀, 2013.
2. http://en.wikipedia.org/wiki/Complex_event_processing
3. http://en.wikipedia.org/wiki/Event-driven_architecture
4. Complex Event Processing Blog <http://www.complexevents.com/>
5. thecepblog <http://www.thecpblog.com/>
6. Event Processing in Action <http://www.manning.com/etzion/>
7. ESP and CEP with Esper <http://goo.gl/LodWe>
8. EDA – ESP & CEP ... with Java <http://goo.gl/KW72p>
9. Event Processing in Java A Financial Services Case Study <http://goo.gl/6iPbt>
10. 실시간 빅 데이터(Real-time Big Data) 프로세싱 맛보기 <http://goo.gl/IGaHB>
11. Esper Documentation <http://goo.gl/isNb6>
12. JBoss Drools Fusion <http://goo.gl/XloGJ>
13. Applying CEP Drools Fusion <http://goo.gl/MsYCH>
14. JBoss RHQ <https://docs.jboss.org/author/display/RHQ>

Outline

3

- CEP Concepts and Use Cases
- Background
- Event Driven Architecture and Processing Paradigm
- Event and Event Stream Concepts
- What's the Real-time?
- Esper for Java
- CEP Development Process
- Event Processing Language
- Input Adapters and Output Adapters
- Major Implementation Steps
- Summary

Concept of CEP (Complex Event Processing)

4

- 교회에서 ...
- 벨이 울린다.
- 턱시도를 입은 남자와 꽃을 든 여자가 함께 걷는다.
- 두 사람 위로 꽃 잎이 날리고 폭죽이 터진다.

Concept of CEP

5



CEP 기술은 이러한 복합적인 이벤트(Event)를 참조하여
결혼식이라는 패턴(Pattern)을 알아 차린다.

Use Case : Algorithmic Stock-Trading

6

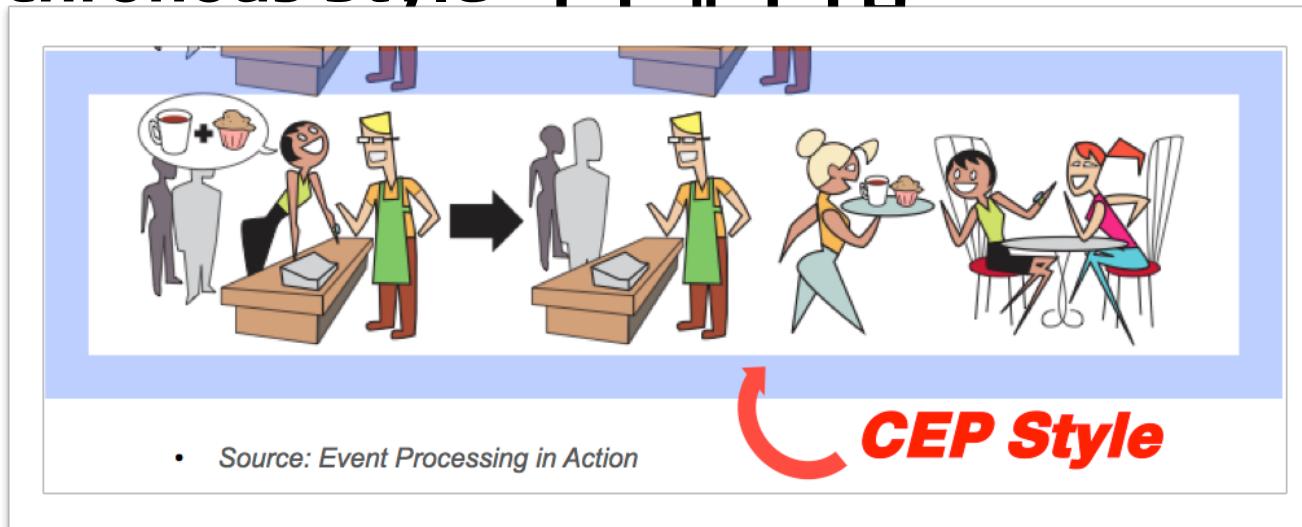
- 대표적인 CEP 적용 사례
- 주식 알고리즘 트레이딩
- 실시간 주식 틱 모니터링
- 실시간 자동 주식 매도 / 매수



Understanding the CEP Technology

7

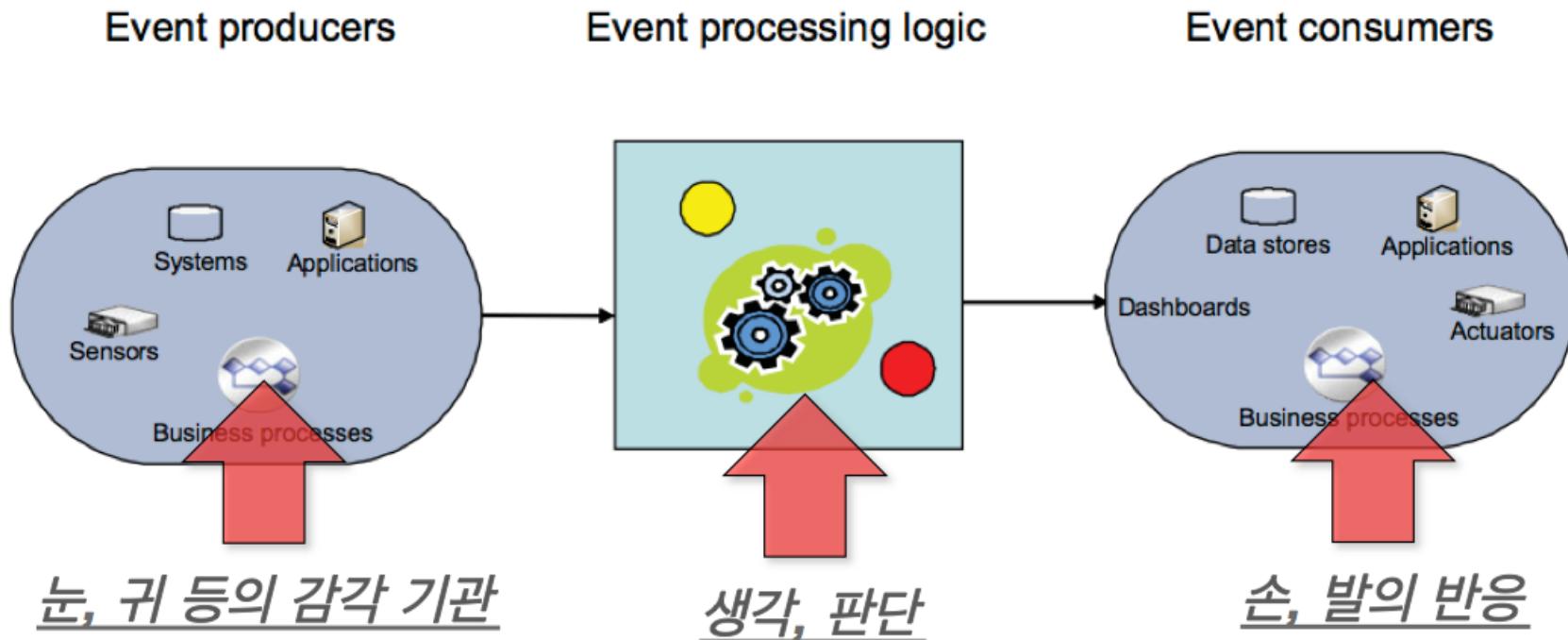
- 1990년 중반부터 학계로 부터 연구가 시작됨
- 실시간 이벤트 처리 기술
- 수많은 이벤트에서 의미있는 이벤트를 찾아내고 필터링
- EDA(Event-Driven Architecture) 기반 시스템
- Intelligence System
- Loosely-coupled 구조적 특징
- Asynchronous Style 처리 패러다임



Event Driven Architecture

8

- Event의 생산, 감지, 소모, 반응을 위한 아키텍처 패턴
- 사람의 감각 기관 및 반응 구조와 유사



EDA Event Processing Paradigm

9

- 선처리 후저장 패러다임
- Asynchronous Style Processing
- Response가 아닌 Reaction하는 시스템
- CEP가 이러한 스타일을 그대로 물려 받음

- EDA의 Event 처리 스타일 (3가지)
 - SEP - Simple Event Processing
 - JMS (Java Messaging Service)
 - ESP - Event Stream Processing
 - 일반적인 모든 Event 스트림 계산 기능 (Filtering, Aggregation, Join)
예: “지난 30분간의 구글 주식 평균값 계산”
 - CEP - Complex Event Processing
 - ESP & 패턴 감지 기능

Complex Event Processing

10

- Detect **patterns** among events
- Define event **correlation**: 이벤트간의 인과관계

“야후! 주식이 10% 이상 떨어진 이후,
구글 주식이 5% 이상 2번 연속으로 오른다면...”

- Open Source Projects
 - EsperTech Esper
 - *SQL-like Script Language*
 - JBoss Drools Fusion
 - *Rule Script Language*

CEP Applied Fields

11

- 기존의 영역에 첨가됨으로써 보다 인텔리전트한 시스템으로 발전

분야	설명	
Finance	Algorithmic Stock-Trading	금융 관리, fraud detection
OI	Operational Intelligence	장비 고장 예측 SLA 모니터링, 침입 탐지
Analytics	Real-time analytics	데이터 분석
	Predictive analytics	예측 분석
EBPM	Event-driven business process management	BPM 자동화
BAM	Business Activity Monitoring	프로세스 모니터링
Computing	Stream computing	스트림 컴퓨팅
ETL	Extract, Transform, Load	데이터 전처리 작업

CEP Products

12

벤더	제품명	설명
TIBCO	TIBCO BusinessEvents	업계 최고 제품
StreamBase Systems	StreamBase CEP	정교한 CEP 제품
SAP	Sybase Event Stream Processor	Sybase 제품 인수
Oracle	Oracle CEP	Esper 기반
IBM	InfoSphere Streams	테러 방지 목적으로 시작 멀티미디어 스트림 프로세싱 가능
	WebSphere Business Events	-
Microsoft	StreamInsight	-
Informatica	RulePoint	-

What is Event?

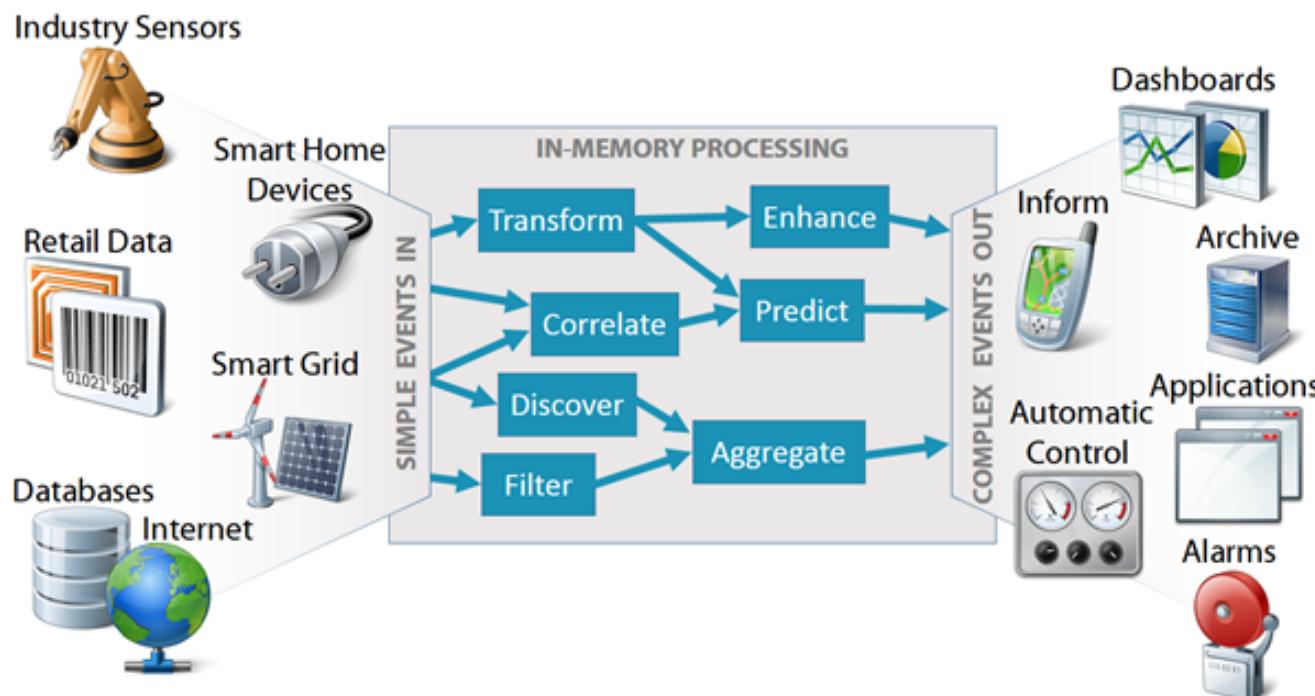
13

- Event : Something that happens
 실제로 발생한 사건, 일, 메세지
- Event란 어떤 것의 **변화**이다!
- 변화한다는 것은 어떠한 상태를 가지고 있다는 의미
- 상태를 가지고 변화하는 모든 것이 Event로 정의될 수 있다.
- 이 세상의 모든 것은 변화한다.
- Event Examples
 - Stock tick
 - 비밀번호 변경
 - 10초마다 실내 온도 측정
 - 위치 이동 이벤트

Event Stream Concept

14

- 정해진 Channel을 통해 시간의 순서대로 연속되는 Event의 흐름
- 시작과 끝이 없는 Event의 연속된 흐름
- Examples
 - 수도관의 물 흐름
 - 차도의 자동차 흐름



Definition of Real-Time System

15

- **Low latency**
 - 낮은 수준의 자연 시간
- **Regularities in response time**
 - 일정한 응답 속도의 보장
- **Providing predictable performance**
 - 예측 가능한 성능 제공

결국 낮은 자연 시간을 일정하게 최대한 오래 유지 시켜주는 것이 핵심

Real-time Three Classes

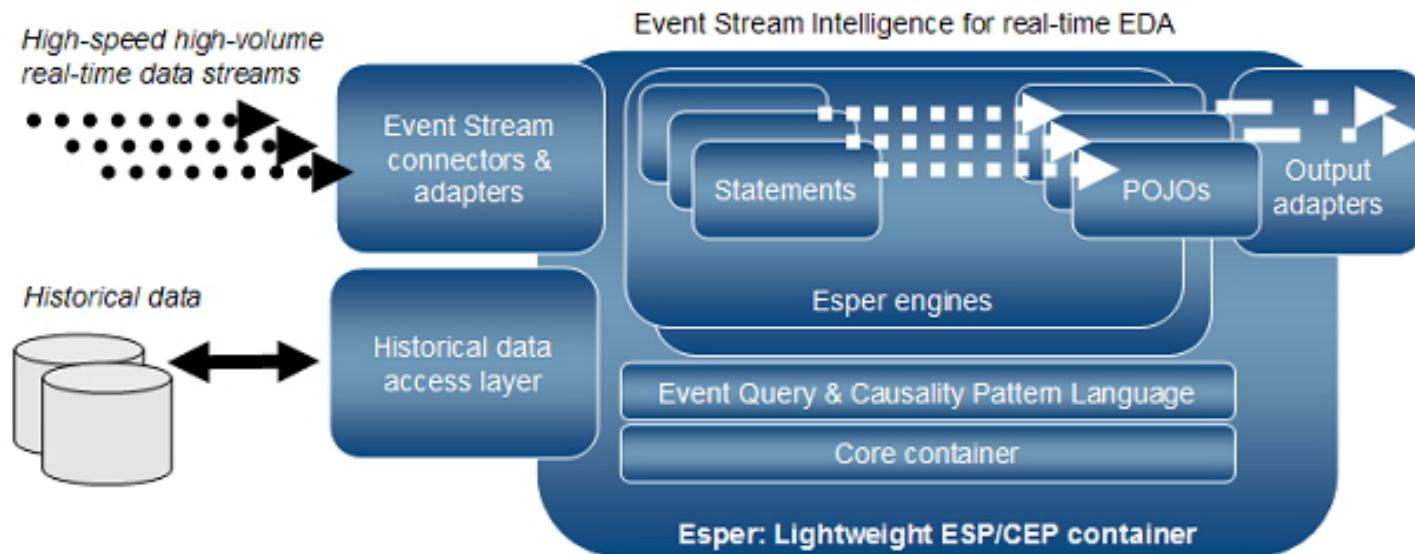
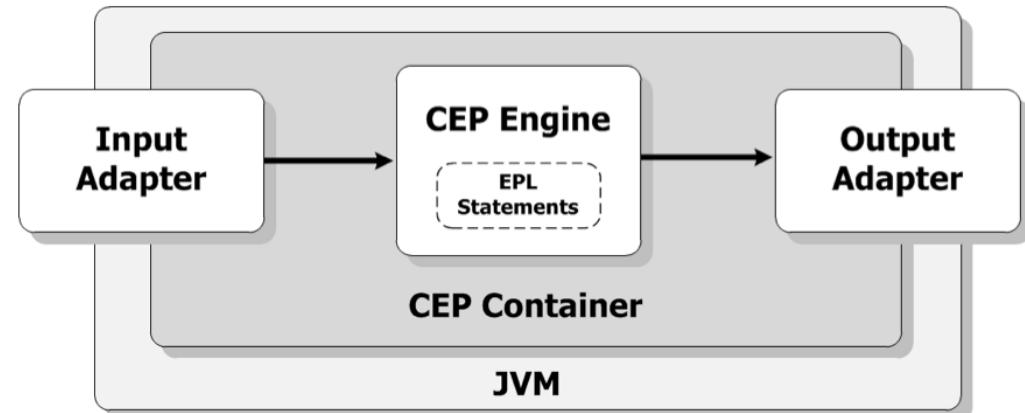
16

- 구체적인 범위 정의는 비즈니스 SLA 요구 수준에 따라 결정됨
- Near Real-time
 - Seconds 수준의 자연 시간 보장
- Real-time
 - Milliseconds 수준 보장
- Real Real-time
 - Microseconds 수준 보장

Esper for Java

17

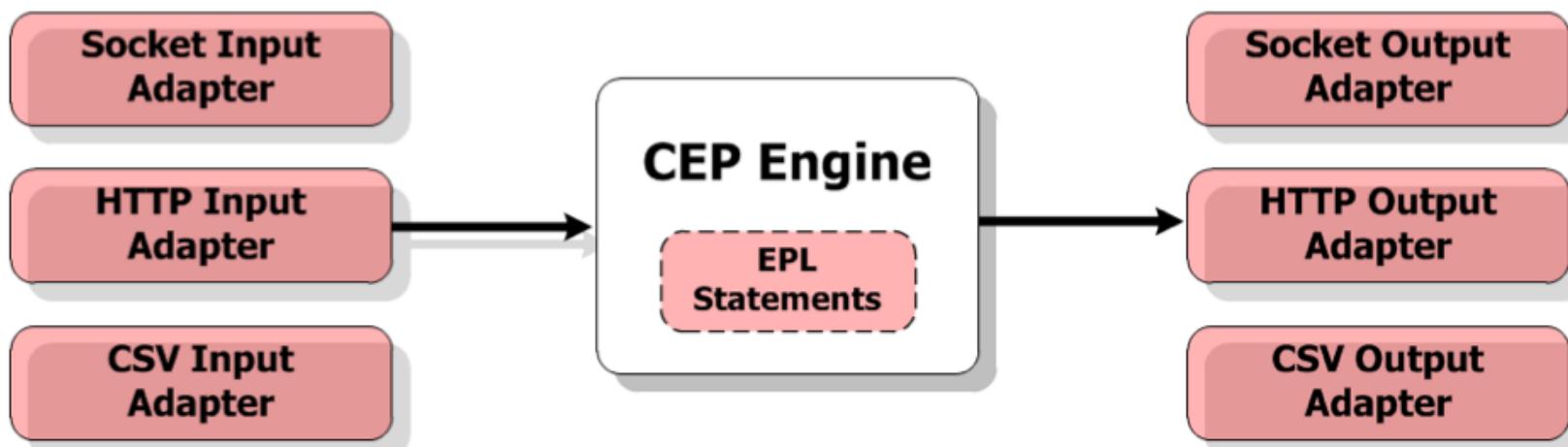
- **ESP/CEP Engine**
- **EPL(Event Processing Language)**
- **Good Reference Documentation**
- **In-Memory Based System**
- **Lightweight & Embeddable**
- **No Runtime Support**



CEP Development Process

18

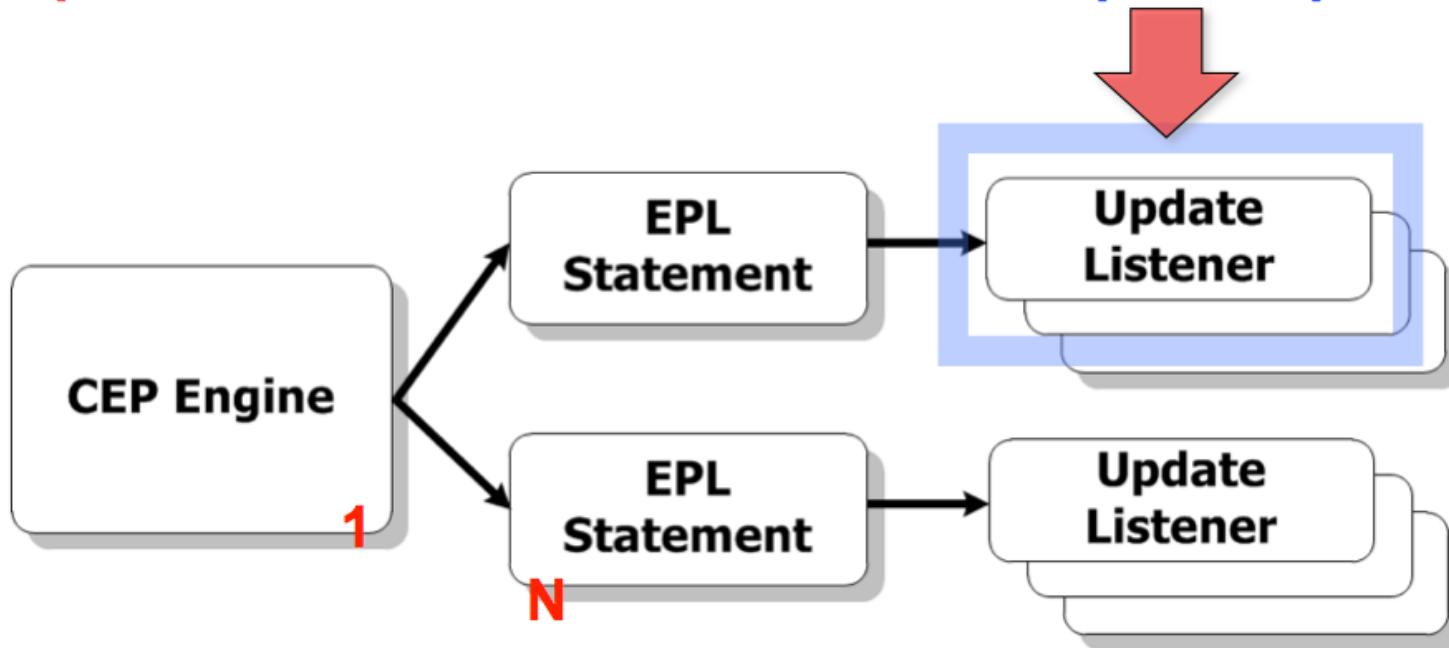
1. 이벤트 정의
2. 이벤트 타입(포맷) 정의
3. EPL 정의
4. Output Adapter 개발 및 등록
5. Input Adapter 개발 및 등록



UpdateListener Interface

19

✓ UpdateListener 인터페이스를 구현하여 Output Adapter 구현



Method Summary

void	<u>update</u> (<u>EventBean</u> [] newEvents, <u>EventBean</u> [] oldEvents) Notify that new events are available or old events are removed.
------	--

What is Event Processing Language (EPL)?

20

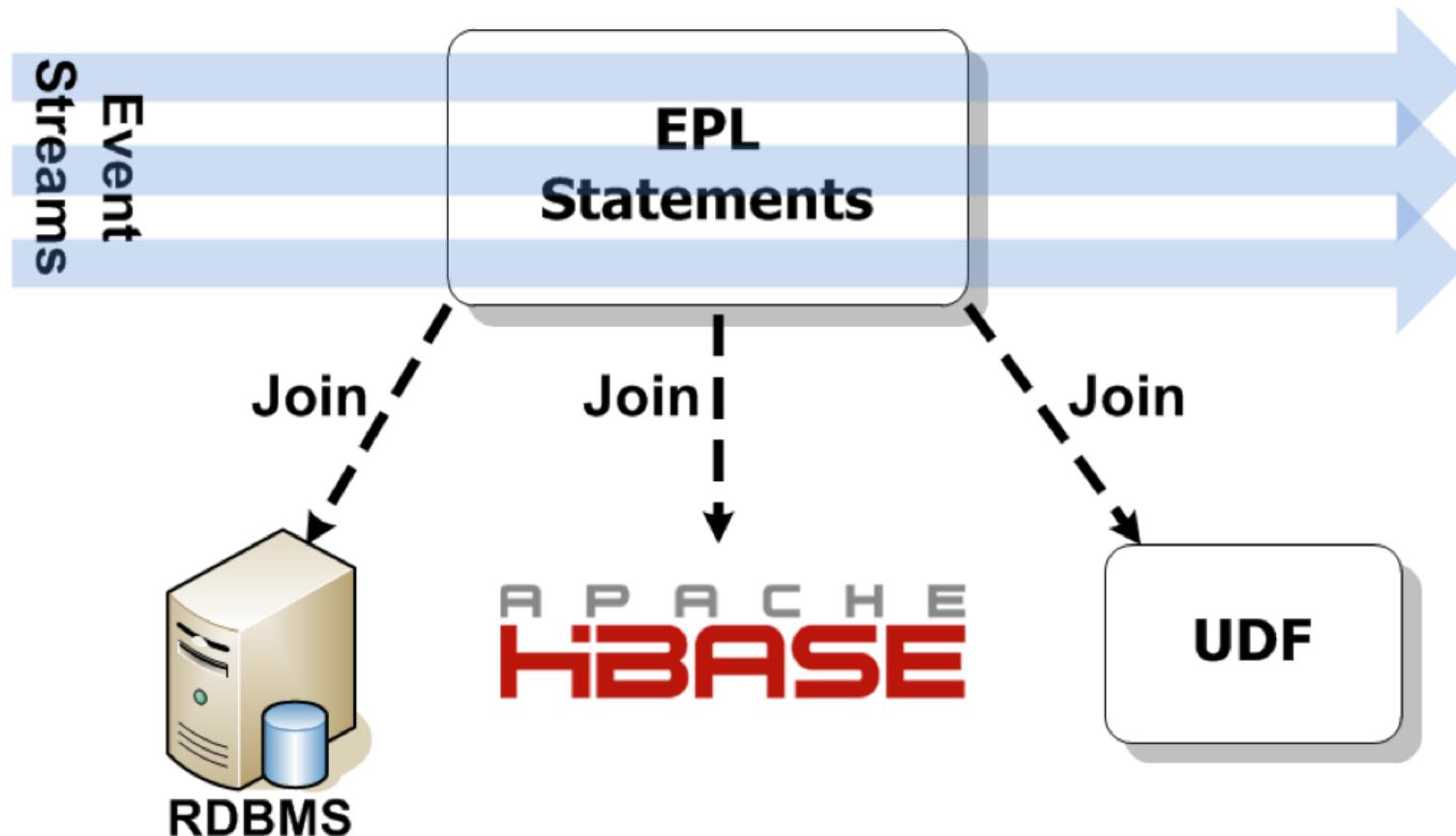
- SQL 형식의 Event 처리 스크립트 언어
- Data windows 제공
- Pattern matching 기능 제공
- Example
 - Continuous query
 - “온라인 쇼핑몰의 지난 30분간 실시간 주문 통계 Top 10”

```
select itemId, count(*) as cnt
from OrderEvent.win:time(30 min)
group by itemId
limit 10
```
- Functional Features
 - Event filtering
 - Sliding data windows and aggregation
 - Grouped windows and output rate limiting
 - Joins and Other Joins
 - Subquery
 - Historical or reference data

Join between EPL and External Resource

21

- 다양한 외부 리소스(RDBMS)와 동시에 조인 가능



Example : Join between EPL and External Resource

22

최근 1시간내 강남역에서 사용자가 찾은
“장소의 이름!!”

select assetId, **zone_name**

from LR.std:unique(assetId).**win:time(1 hour)** as lr,

sql:db[select zone_name from ZoneName

where zone in ('강남역 code')

EPL 조인 조건



where zone=\${lr.zone}]

장소 이름 마스터 테이블

MySQL 네이티브 쿼리

고객 마스터 정보

상품 추천 정보 등등...



Major Implementation Steps

23

- Step 1: CEP 엔진 인스턴스 초기화

```
EPSupplier epEngine =  
    EPSupplierManager.getProvider("MyID", configuration);
```

- **MyID**: Engine 아이디 등록
- **configuration**: Engine 튜닝 값 정의

- Step 2: EPL 등록하기

```
EPStatement statement =  
    epEngine.getEPAdministrator().createEPL("EPL string");
```

- **EPL string**: EPL 질의문 (e.g., "select * from tagEventData")

Major Implementation Steps

24

- Step 3: Output Adapter 구현하기

```
statement.addListener(new UpdateListener() {  
    public void update(EventBean[] newEvents) {  
        // event processing code: file writing, email sending, ...  
    }  
});
```

- Step 4: Input Adapter 구현하기, 이벤트 생성 및 전달

```
VTSGTagDataEvent event =  
    new VTSGTagDataEvent(assetID, timestamp, tagName, tagValue, ...);
```

```
epEngine.getEPRuntime().sendEvent(event);
```

- Engine으로 Event를 전달

Summary

25

- CEP Concepts and Use Cases
- Background
- Event Driven Architecture and Processing Paradigm
- Event and Event Stream Concepts
- What's the Real-time?
- Esper for Java
- CEP Development Process
- Event Processing Language
- Input Adapters and Output Adapters
- Major Implementation Steps

