Microsoft

SQL Unplugged

StreamInsight를 통한 실시간 이벤트 데이터 처리

홍 세 환
㈜ 인브레인

발표자 소개

홍세환

- ㈜ 인브레인
  - http://www.inbrein.com
  - 컨설팅사업부 부장
  - 개발 PM
  - C#

목차

- StreamInsight 소개
  - Complex Event Processing
  - StreamInsight Architecture
- Application Component
  - Event (Flow, Control and Definition)
  - LINQ & Window
  - Adapters
  - Development Model
  - Deployment Model
  - Monitoring & Event Flow Debugger
- Usage Example

StreamInsight 소개
(Complex Event Processing)

Event Driven Architecture (Event Processing Styles) – Wikipedia

- Simple event processing
  - Simple events can be created by a sensor detecting changes in tire pressures or ambient temperature.
- Event stream processing
  - Stream event processing is commonly used to drive the real-time flow of information in and around the enterprise, which enables in-time decision making.
- **Complex event processing**
  - CEP is commonly used to detect and respond to business anomalies, threats, and opportunities.

Brenda M. Michelson, Event-Driven Architecture Overview, *Patricia Seybold Group*, February 2, 2006

StreamInsight 소개
(Complex Event Processing)

Complex Event Processing

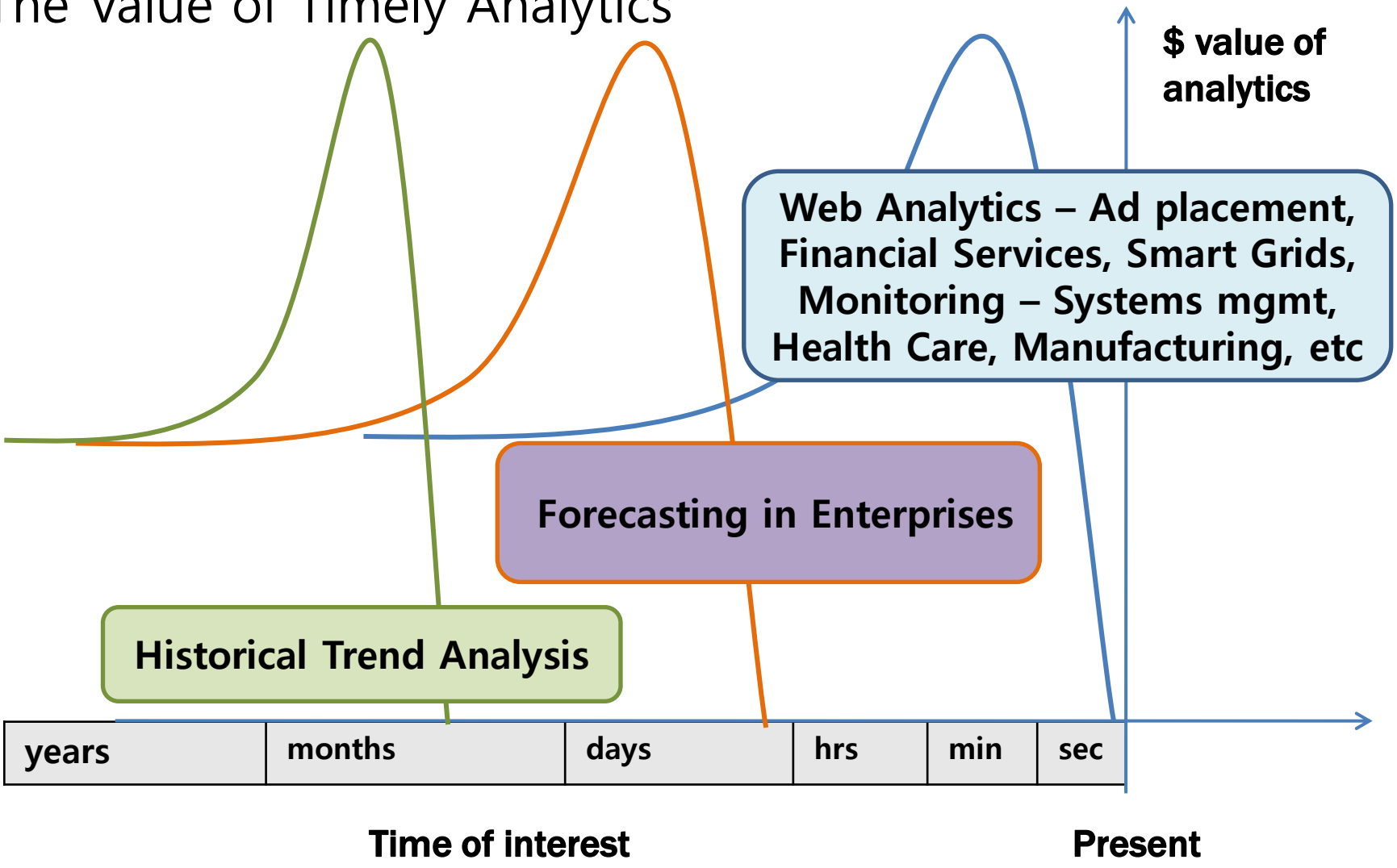
- Complex Event Processing (CEP) is the **continuous** and incremental processing of event streams from **multiple sources** based on declarative query and pattern specifications with **near-zero latency**

The Goals of CEP

- Identify and detect from seemingly unrelated multiple events:
  - Meaningful patterns, Relationships
  - Trends, Gaps (expected events that did not occur)
  - Abstractions
  - Exceptions, Opportunities
- Analyze data without storing it first
- Trigger immediate response actions
- Mine events for new business KPIs

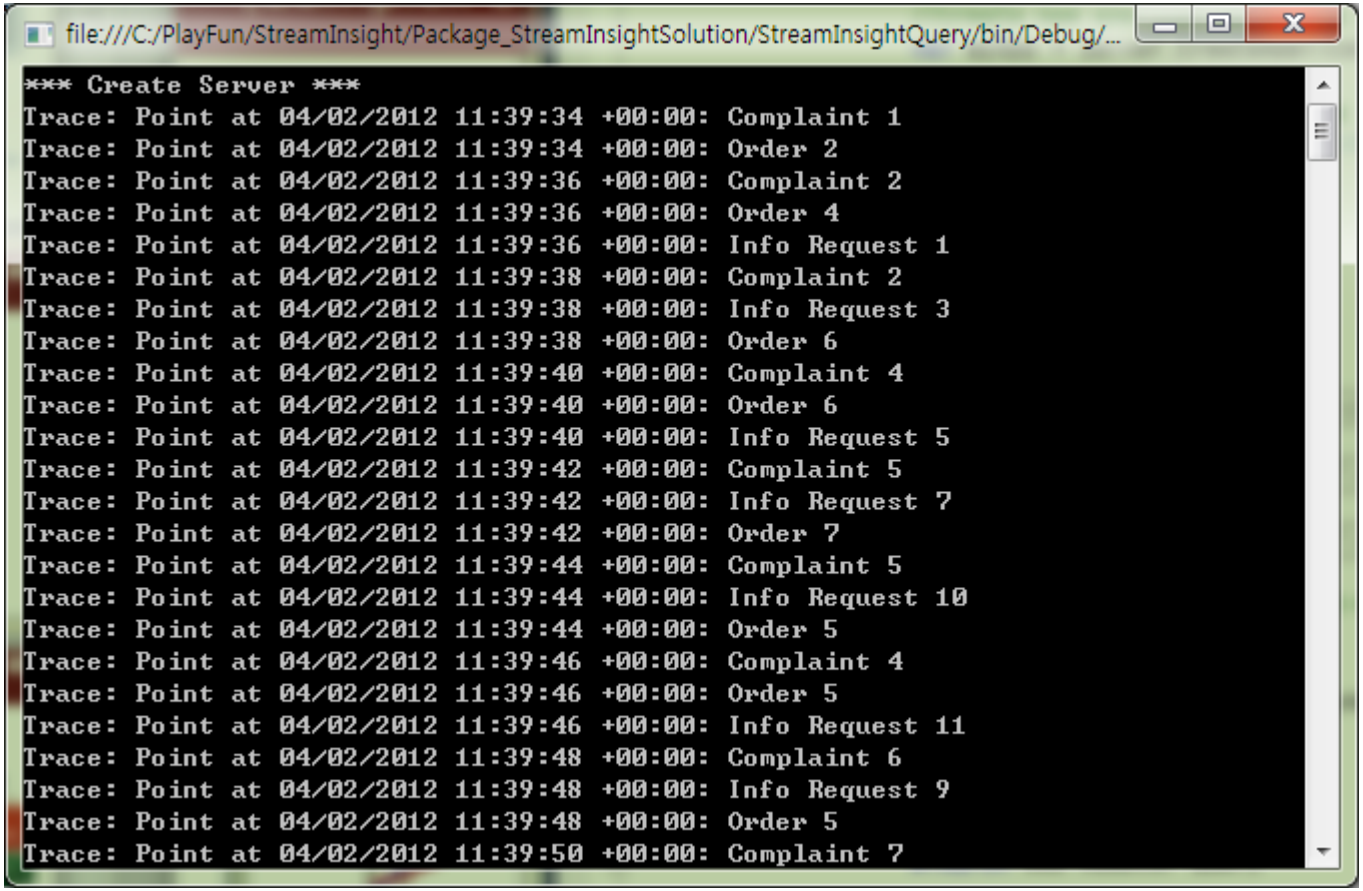
StreamInsight 소개
(Complex Event Processing)

The Value of Timely Analytics



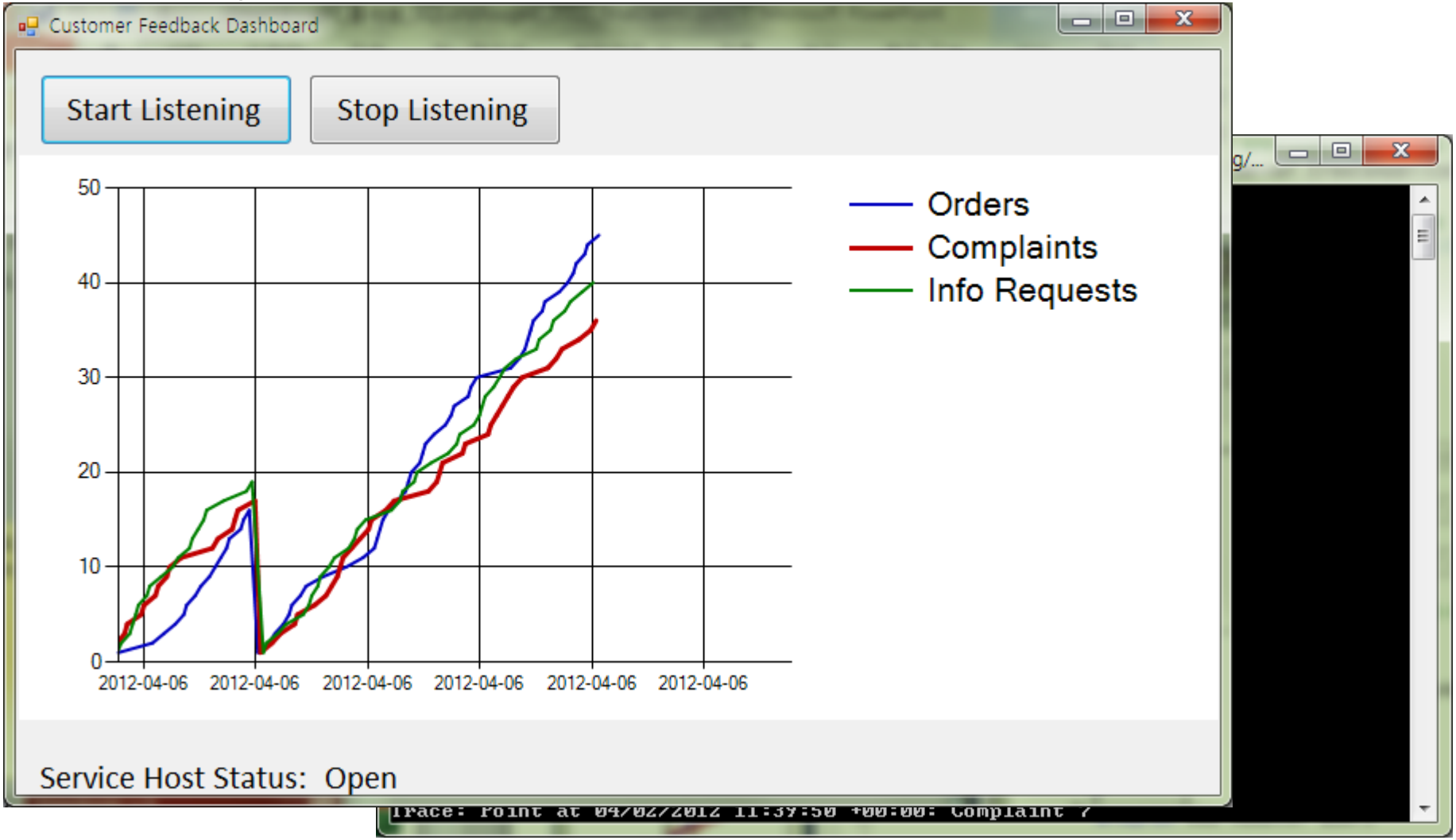
StreamInsight 소개
(Complex Event Processing)

Web Analytics Demo (Package_StreamInsightSolution)



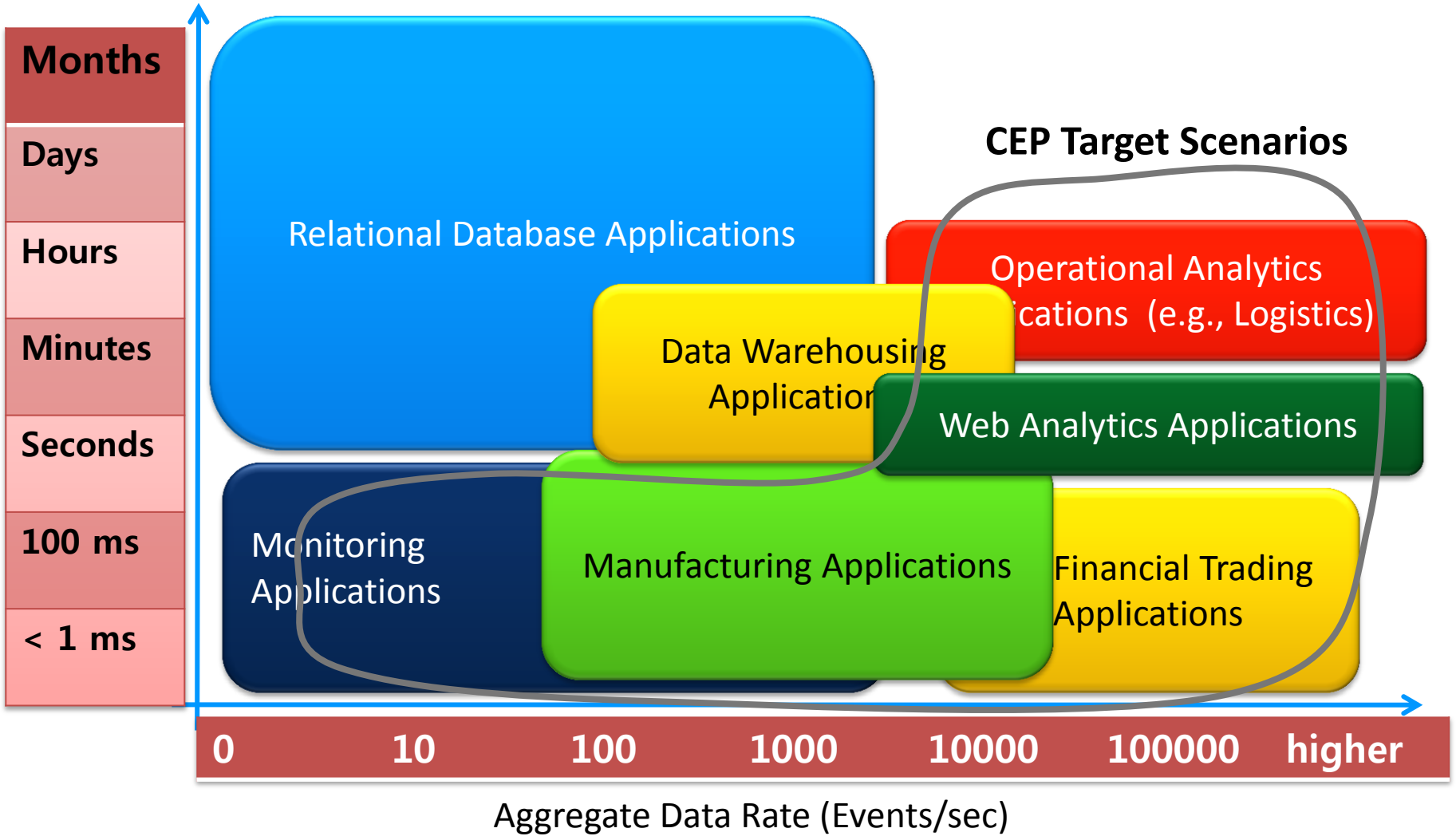
StreamInsight 소개
(Complex Event Processing)

Web Analytics Demo (Package_StreamInsightSolution)

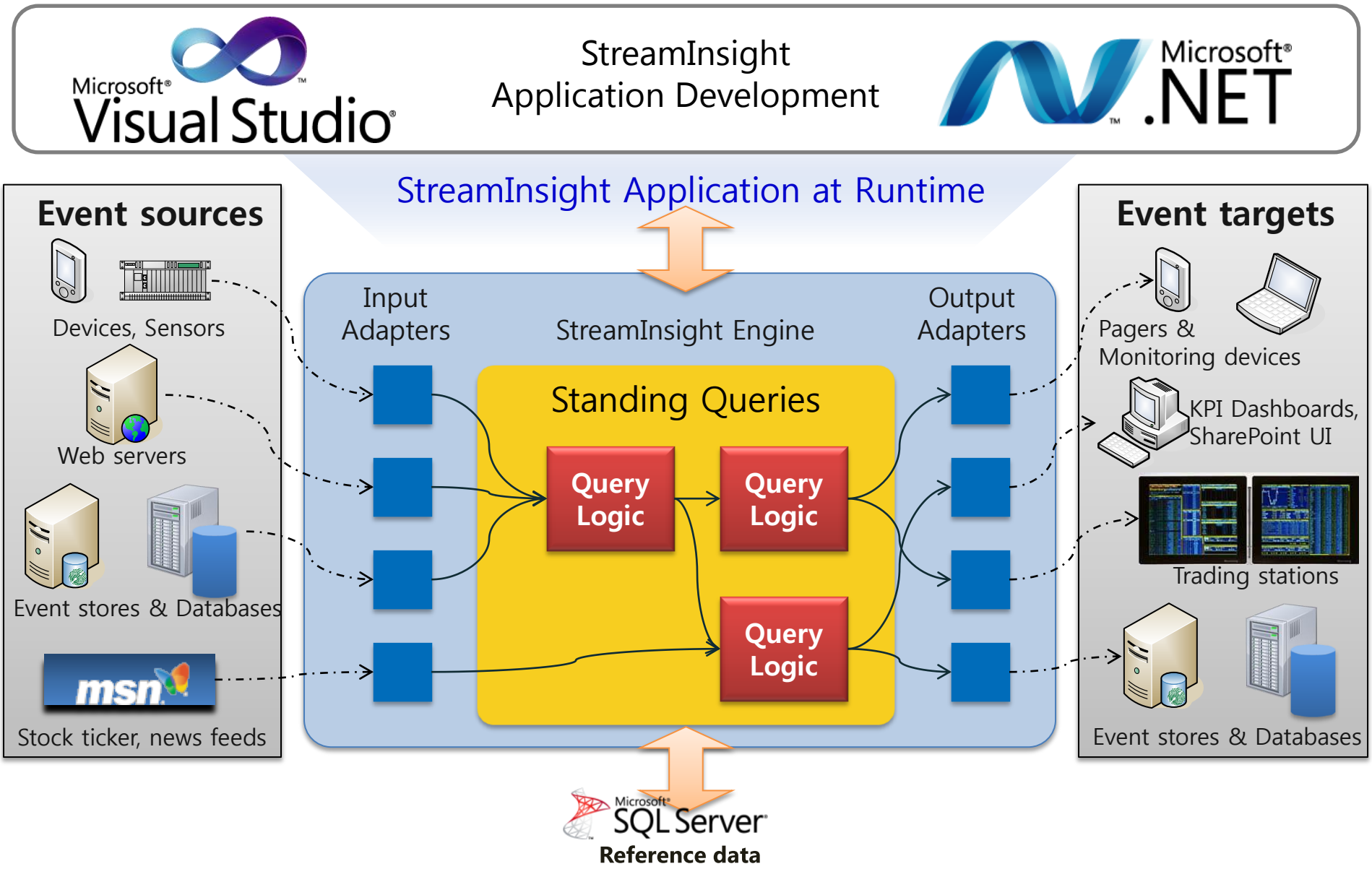


StreamInsight 소개
(Complex Event Processing)

CEP Sweet Spot



StreamInsight 소개
(StreamInsight Architecture)

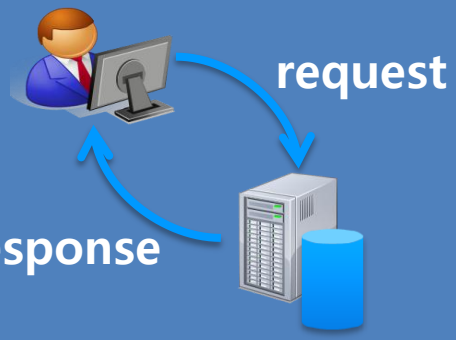


StreamInsight 소개
(StreamInsight Architecture)

StreamInsight VS. DBMS

Analytical results need to reflect important changes in business reality immediately and enable responses to them with minimal latency

	Database	CEP
Queries	Ad hoc on stored data	Continuous standing queries
Latency	Seconds	Milliseconds
Data Rate	Hundreds per Second	Tens of thousands per second
Query Semantics	Declarative relational analytics	Declarative relational *and* ***temporal*** analytics



StreamInsight 소개
(StreamInsight Architecture)

SQL Server Capabilities by Edition

Workload	Standard	Enterprise	Datacenter	Parallel Data Warehouse
Custom/Packaged OLTP Apps	4 procs, 64GB RAM, Backup Compression	8 procs, 2TB RAM, Adv. Security, Backup Compression	>8 procs, OS Max, Adv. Security, Backup Compression	N/A
Server Consolidation	1 VM/license	4 VMs/license, Resource Governor App & Multi-Server Mgmt (up to 25 instances)	Unlimited Virtualization, Resource Governor, App & Multi-Server Mgmt (> 25 instances)	N/A
Data Warehousing		Scale-Up DW, Data Compression 10s of TBs, Up to 30 TB with FastTrack	Scale-Up DW, Data Compression 10s of TBs	Scale-Out DW 10s - 100s of TBs
Business Intelligence	Dept/Team BI	Enterprise-Scale BI, Master Data Services, PowerPivot Mgmt	Enterprise-Scale BI, Master Data Services, PowerPivot Mgmt	Integrated with SSIS, SSAS and SSRS
StreamInsight	**<5000 events/sec & > 5 sec latency**	**<5000 events/sec & > 5 s latency**	**>5000 events/sec & < 5 s latency**	**Future coverage**

Application Component

- Event Stream Concept
- .NET 3.5 SP1, 4.0 and CLR for the runtime
- Visual Studio for developer productivity (IntelliSense)
- C# for apps development
  - Adapter, Object Model
  - Diagnostic APIs
  - IEnumerable – Pull Model
  - IObservable – Push Model
- LINQ for Query Language surface
  - Anonymous Method
  - Lambda Expression
  - SQOs (Standard Query Operators)
- Event Flow Debugger

Application Component
Event (Flow, Control and Definition)

Streams

- All data organized into Streams, Potentially unending
- Data may change over time
- Often represent the same value over time

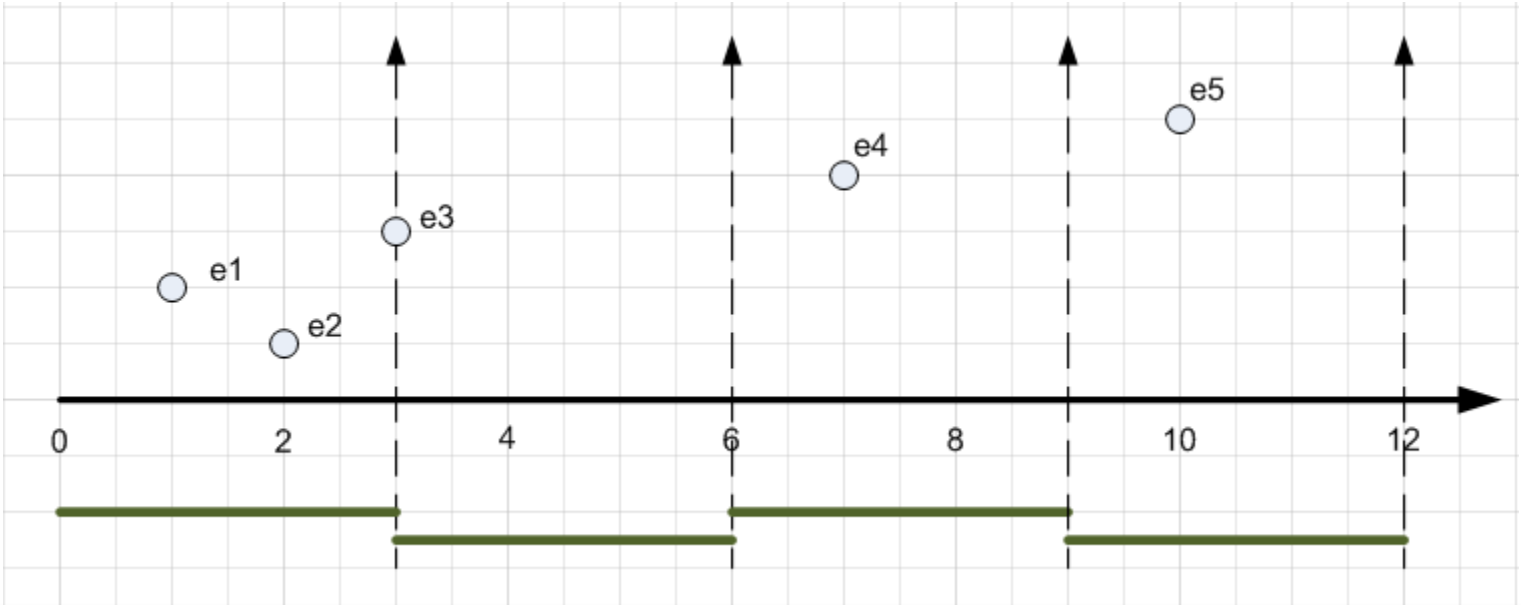
Events

- Data in streams is packaged into events
- Two parts to an event
  - Header (kind of event and timestamps)
  - Payload (.NET data structure)
- Timestamps are DateTimeOffset data type
  - All times normalized to UTC in server

Timestamps/ MetadataC	Long pumpID	String Type	String Location	Double flow	Double pressure
…	…	…	…	…	…

Application Component
Event (Flow, Control and Definition)

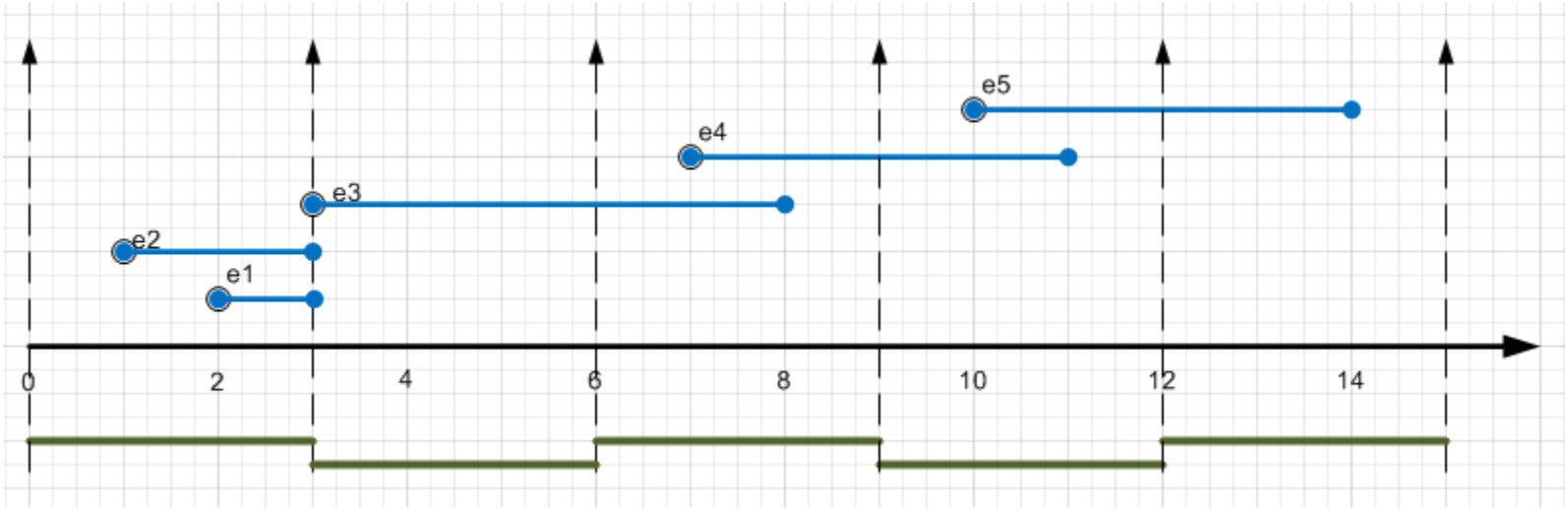
Event Model (Event Shapes) - Point Event



EventKind	StartTime	EndTime	Payload
INSERT	2009-12-27 02:04:00.213	2009-12-27 02:04:00.213 + c	EU-23423-12
INSERT	2009-12-27 02:04:04.329	2009-12-27 02:04:04.329 + c	EU-23423-15
INSERT	2009-12-27 02:04:04.234	2009-12-27 02:04:04.234 + c	EU-23423-18

Application Component
Event (Flow, Control and Definition)

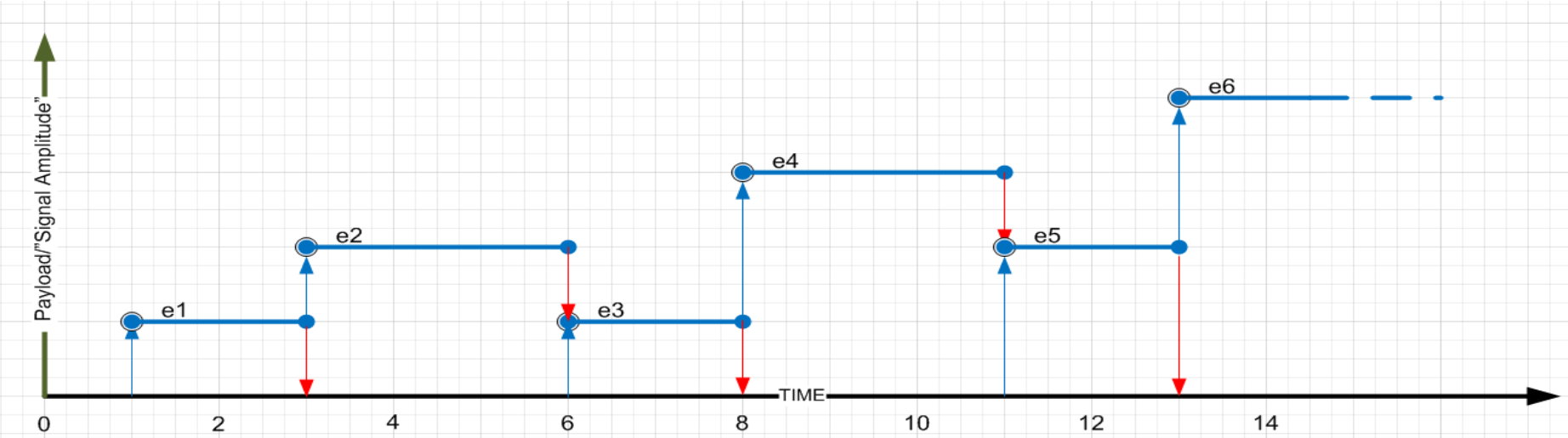
Event Model (Event Shapes) - Interval Event



EventKind	StartTime	EndTime	Payload
INSERT	2009-12-27 02:04:00.213	2009-12-27 02:04:04.329	EU-23423-12
INSERT	2009-12-27 02:04:04.329	2009-12-27 02:04:08.234	EU-23423-15
INSERT	2009-12-27 02:04:04.234	2009-12-27 02:04:04.523	EU-23423-18

Application Component
Event (Flow, Control and Definition)

Event Model (Event Shapes) - Edge Event



EventKind	EdgeType	StartTime	EndTime	Payload
INSERT	Start	2009-12-27 02:04:00.213	∞	EU-23423-12
INSERT	End	2009-12-27 02:04:00.213	2009-12-27 02:04:04.329	EU-23423-12
INSERT	Start	2009-12-27 02:04:04.234	∞	EU-23423-18
INSERT	End	2009-12-27 02:04:04.234	2009-12-27 02:04:08.238	EU-23423-18

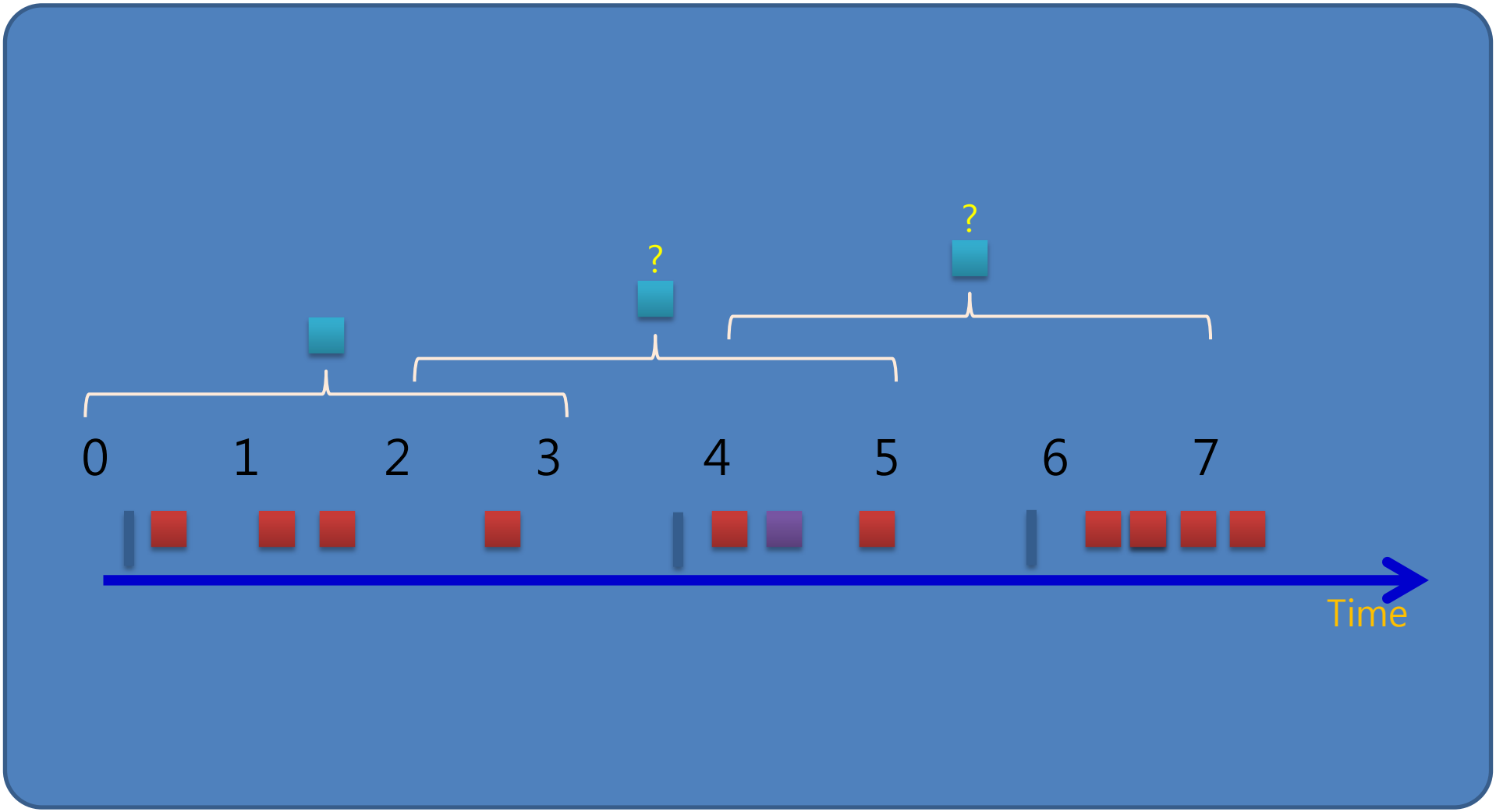
Application Component
Event (Flow, Control and Definition)

Event Kind

- Two event kinds
  - INSERT (new data for the stream)
  - CTI (Current Time Increment)
- C T I
  - Added to input stream
    - Typically added by input adapter
    - Can be declaratively added in query binding
  - Processing out of order data
  - Responsive stream
- Update – in Edge Event but Insert EndTime
- Delete – no concept of deleting, ReleaseEvent

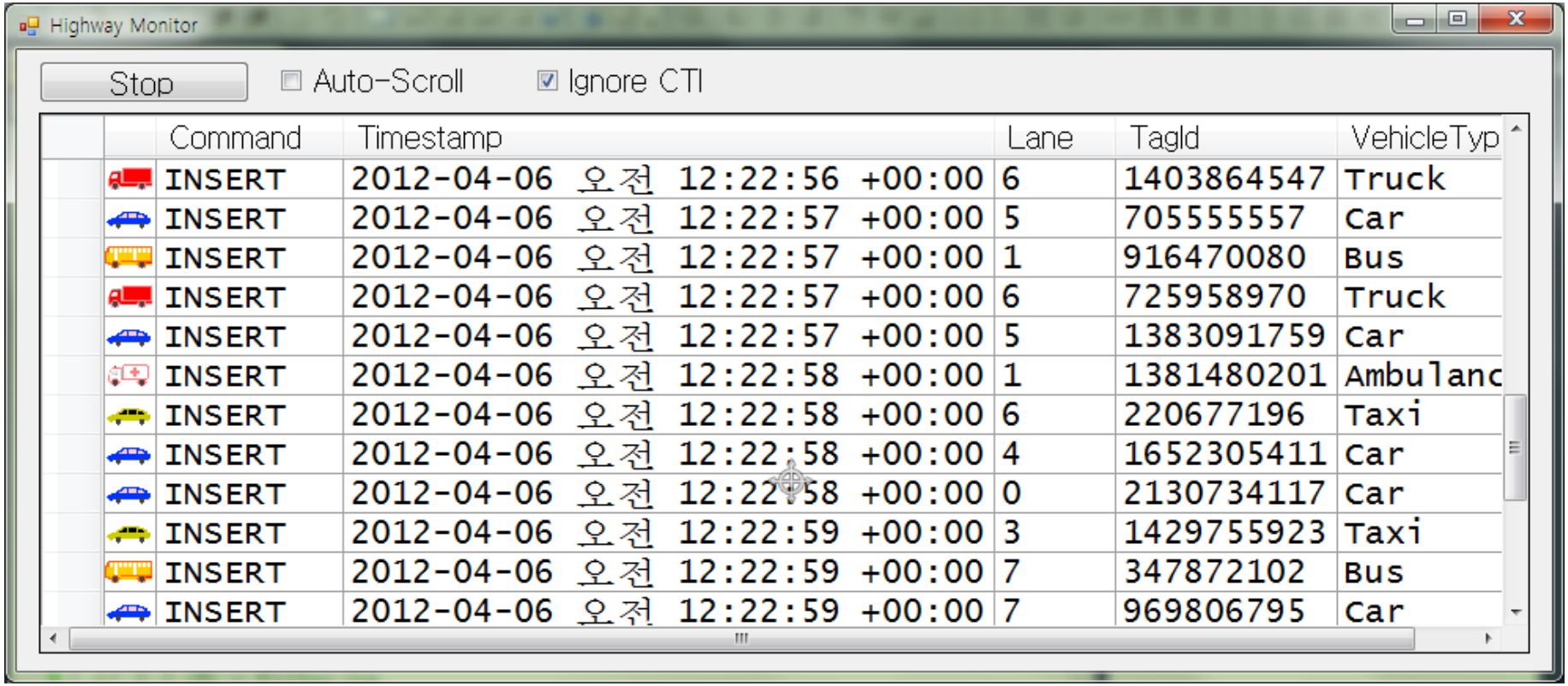
Application Component
Event (Flow, Control and Definition)

Stream Liveliness



Application Component
LINQ & Window

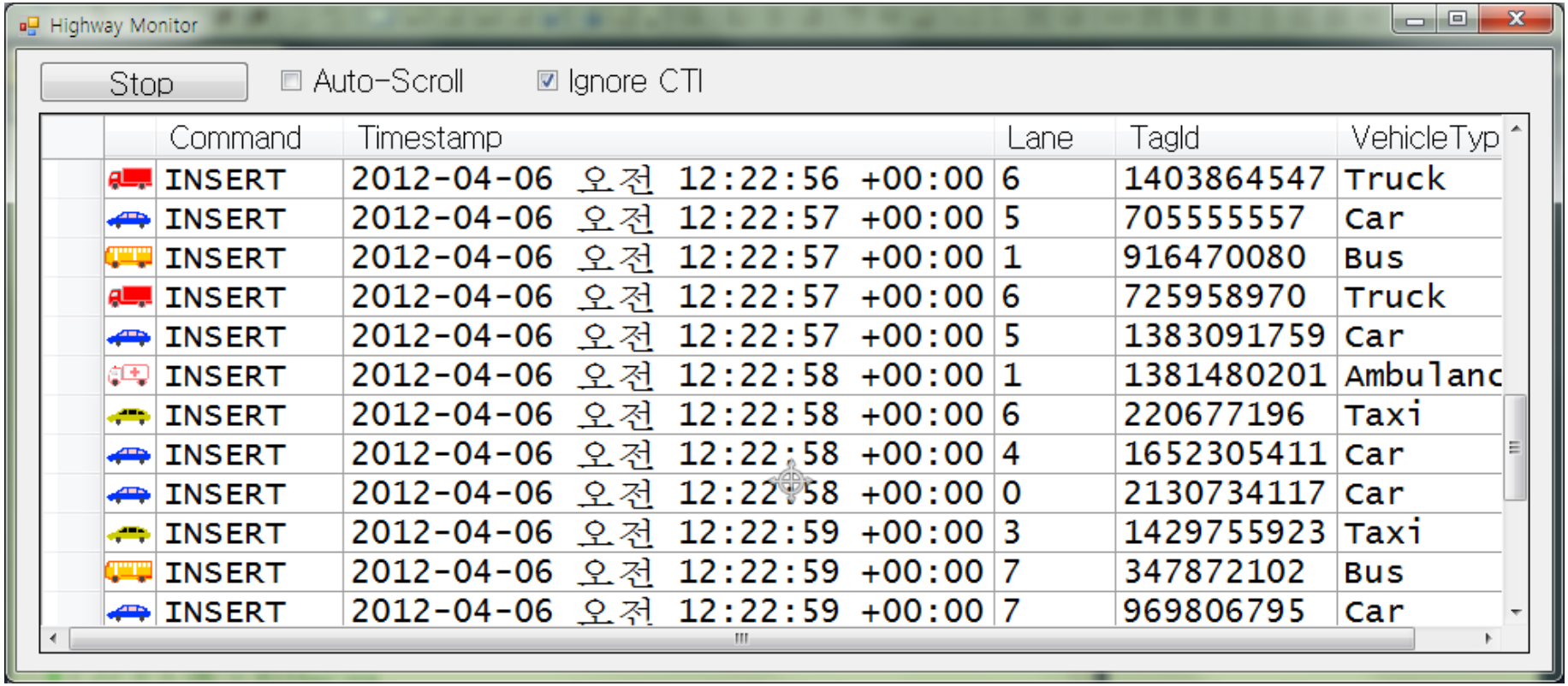
LINQ Demo (MSSQL2008R2DevelopersTrainingKit₩Labs₩SQL10R2UPD05-HOL-01₩Source₩Assets₩DisplayQueryVersion)




```
// 5 - filtering
var queryOutput = from e in input
                  //where e.VehicleTypeId == 2
                  select new { e.Lane, e.TagId, VehicleType = TollPointEvent.VehicleTypeName(e.VehicleTypeId) };
```

Application Component
LINQ & Window

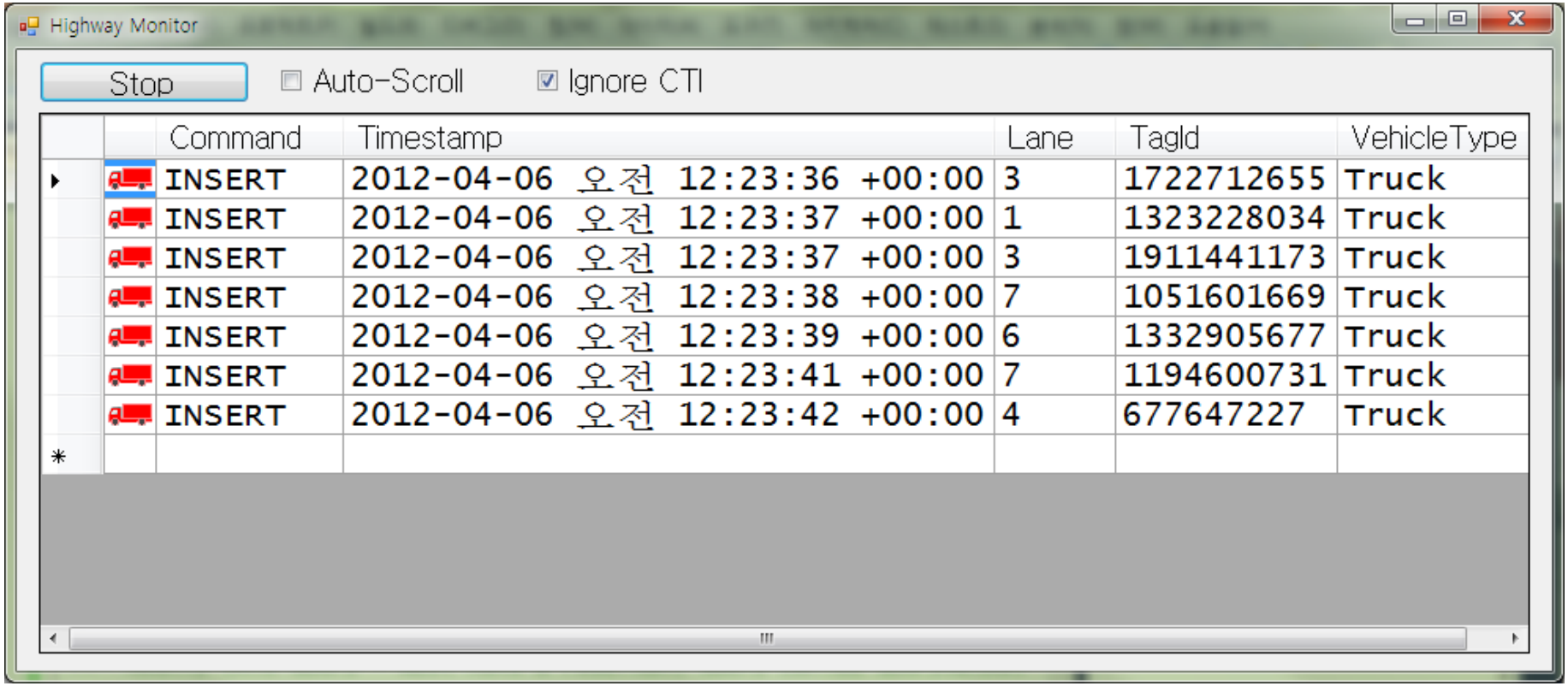
LINQ Demo (MSSQL2008R2DevelopersTrainingKit₩Labs₩SQL10R2UPD05-HOL-01₩Source₩Assets₩DisplayQueryVersion)




```
// 5 - filtering
var queryOutput = from e in input
                  where e.VehicleTypeId == 2
                  select new { e.Lane, e.TagId, VehicleType = TollPointEvent.VehicleTypeName(e.VehicleTypeId) };
```

Application Component
LINQ & Window

LINQ Demo (MSSQL2008R2DevelopersTrainingKit₩Labs₩SQL10R2UPD05-HOL-01₩Source₩Assets₩DisplayQueryVersion)




```
// 5 - filtering
var queryOutput = from e in input
                  where e.VehicleTypeId == 2
                  select new { e.Lane, e.TagId, VehicleType = TollPointEvent.VehicleTypeName(e.VehicleTypeId) };
```

LINQ Query Example

LINQ Example – JOIN, PROJECT, FILTER:

```
from e1 in MyStream1
join e2 in MyStream2
on e1.ID equals e2.ID
where e1.f2 = “foo”
select new { e1.f1, e2.f4 };
```

Join

Filter

Project

LINQ Example – GROUP&APPLY, WINDOW:

```
from e3 in MyStream3
group e3 by e3.i into SubStreams
from s4 in SubStreams
from e4 in s4.HoppingWindow(FiveMinutes,ThreeSeconds)
select new {
        pl = new MyNewPayload(e4.i, e4.f)};
```

Grouping

Window

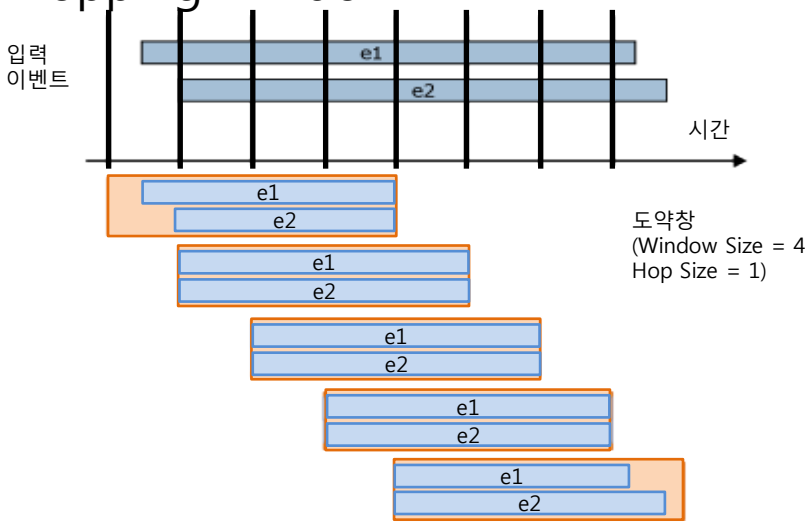
Application Component
LINQ & Window

CEP Query Features

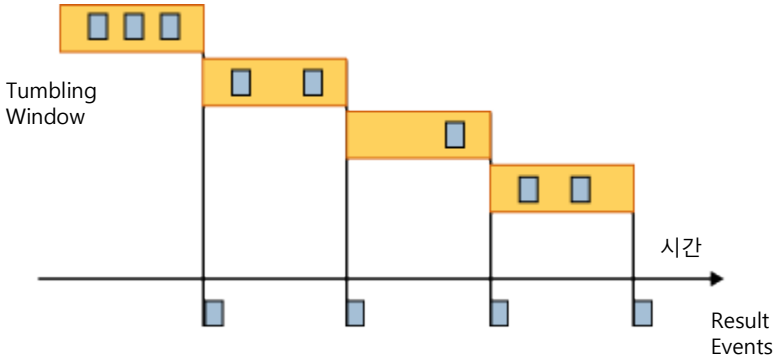
- Operators over streams
  - PROJECT
  - JOIN
  - EXISTS
  - FILTER
  - GROUP & APPLY
  - SUM, COUNT, …
  - TOP-K
  - **Temporal operations - window**
- Extensibility – to add new domain-specific operators
- Queries are written over specific event types
- Support for streaming data, reference data (lookup), and historical data (replay)

Query Elements : Event Windows

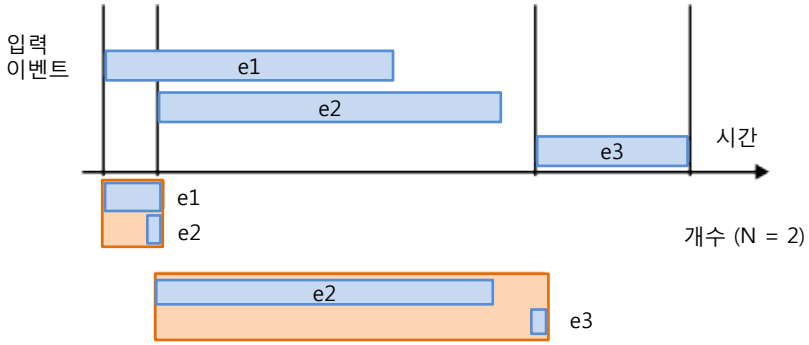
Hopping Window



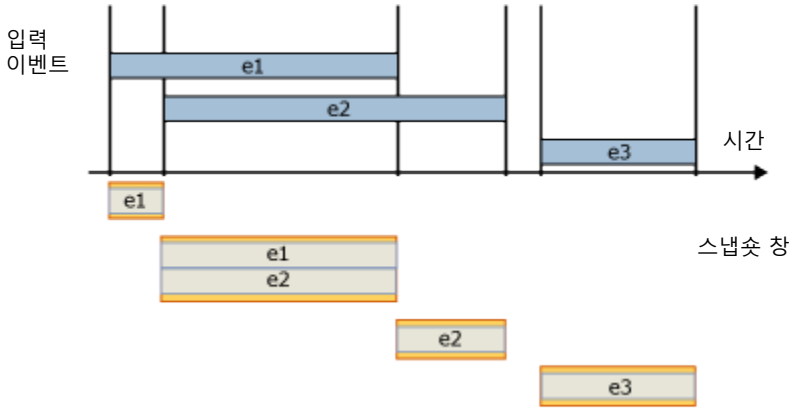
Tumbling Window



Count Window



Snapshot Window



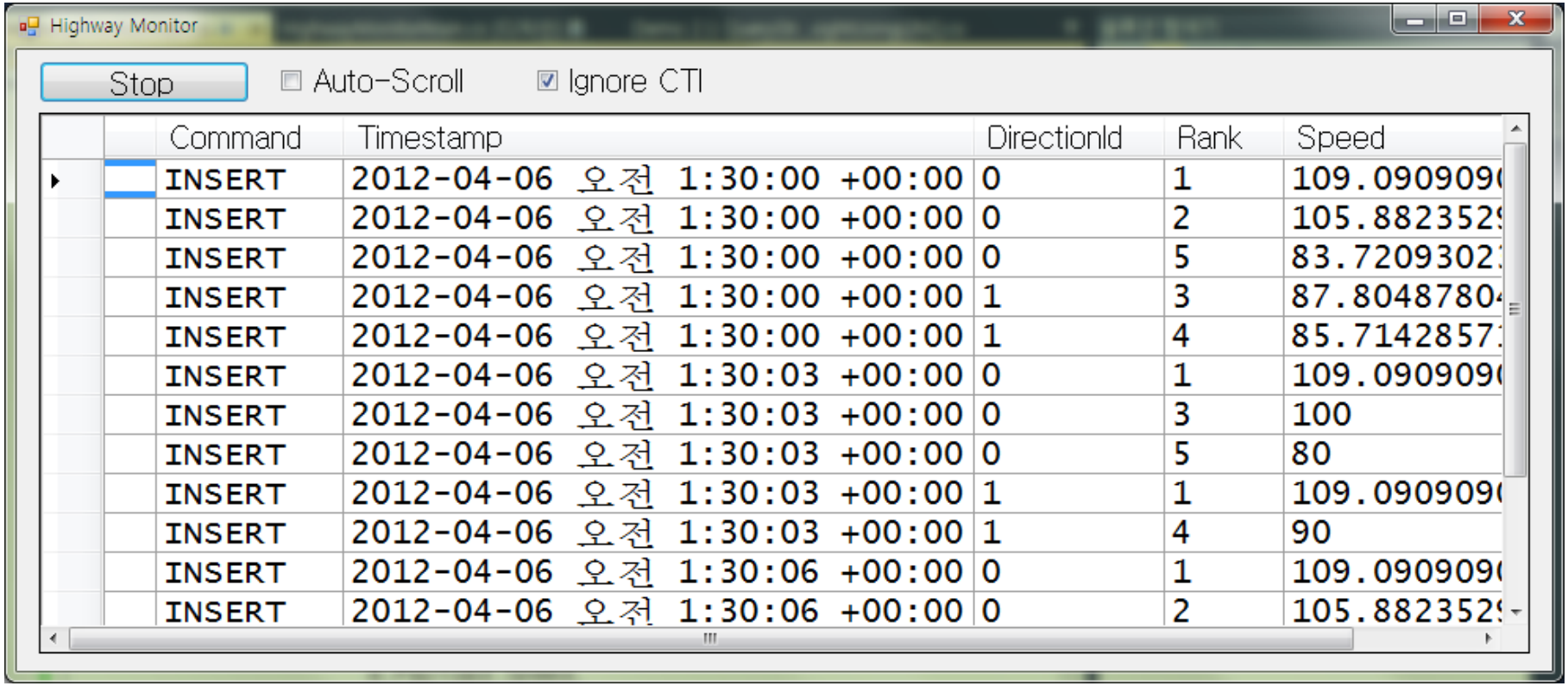
Query Elements : Event Window Demo

```
var vehicleSpeeds = from e in input
                    select new
                    {
                        e.DirectionId,
                        Speed = 60.0 * 60.0 / e.MillisecondsToPas
                    };

var topSpeeds = (from w in vehicleSpeeds.TumblingWindow
                  (TimeSpan.FromSeconds(10))
                 from e in w
                 orderby e.Speed descending
                 select e).Take(5, e => new
                 {
                     e.Payload.DirectionId,
                     e.Payload.Speed,
                     e.Rank
                 });
```

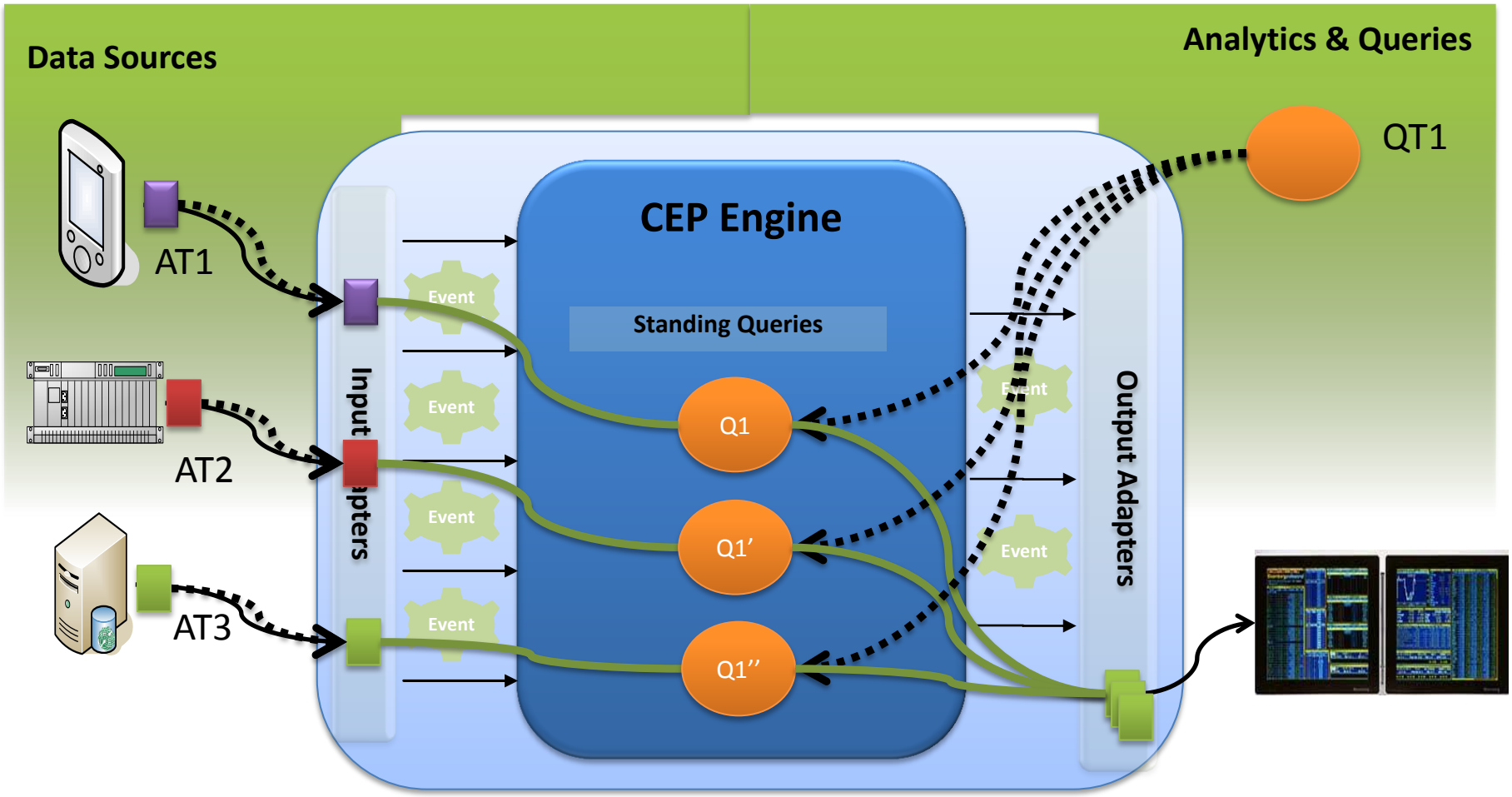
Application Component
LINQ & Window

Query Elements : Event Window Demo




```
            e.Payload.DirectionId,
            e.Payload.Speed,
            e.Rank
        });
```

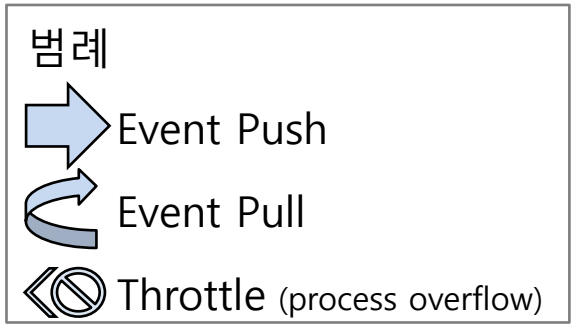
Query Binding

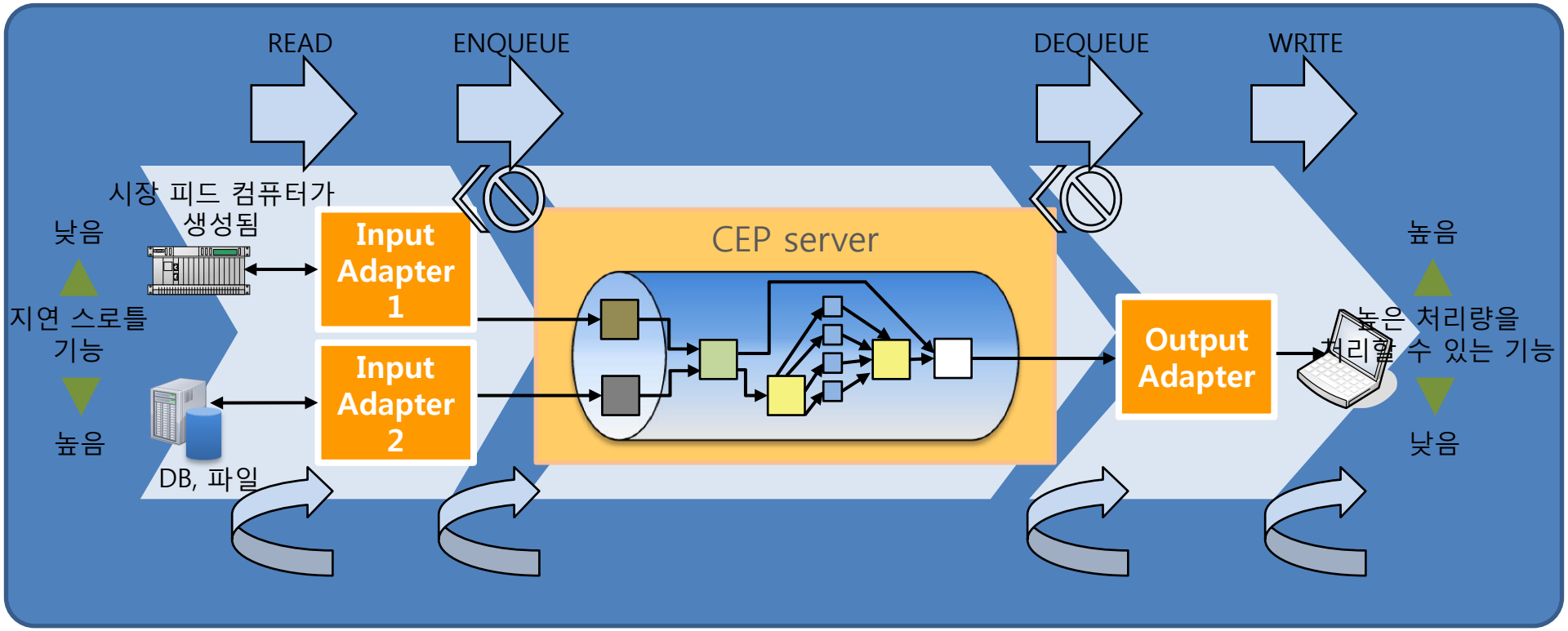


Application Component
Adapters

Event Stream Adapters : Push vs. Pull Model

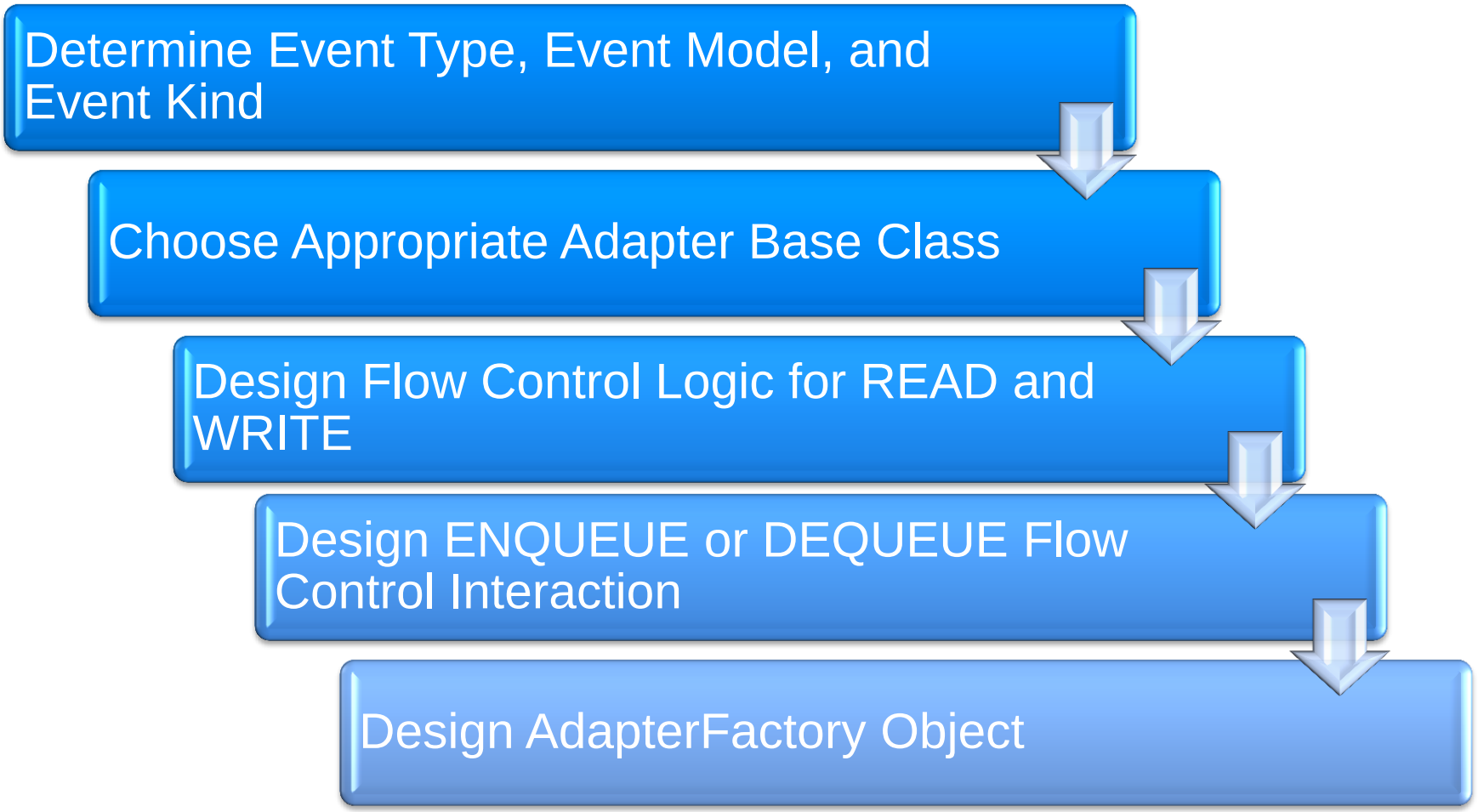
- Interaction points support Push and Pull
- Consumer can throttle input





Application Component
Adapters

Adapter Development Tasks



Application Component
Development Model

SQLa Unplugged

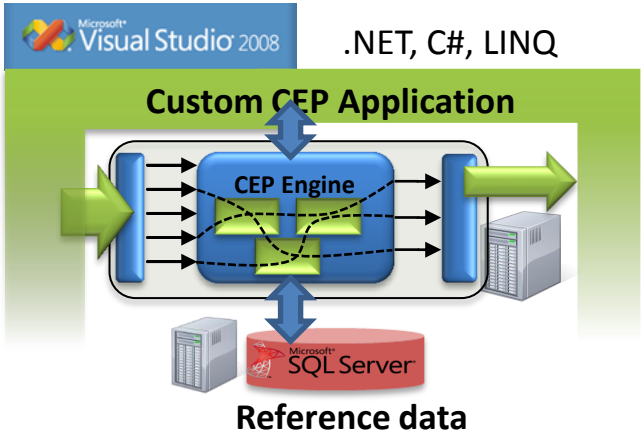
StreamInsight Development Two Models

Implicit Server	Explicit Server
Easiest	Most Flexible
Hides most complexity	Provide complete control of StreamInsight application, Development Environment
Allows developers to focus on query logic	Allows for reuse Queries, Adapters, Event Types, Third-party Query Templates
Query is automatically hosted	Code creates all objects and registers them into server
Stored in memory, not on stable storage (disk)	Server can store metadata in memory or persist to disk
	Server can run locally or remotely via Web service

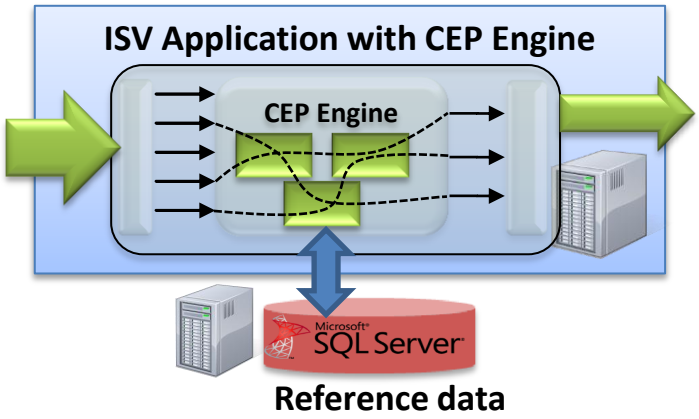
Application Component
Deployment Model

CEP Deployment Scenarios

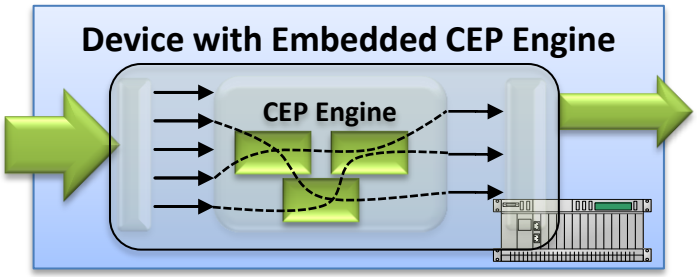
Scenario 1: Custom CEP Application Dev



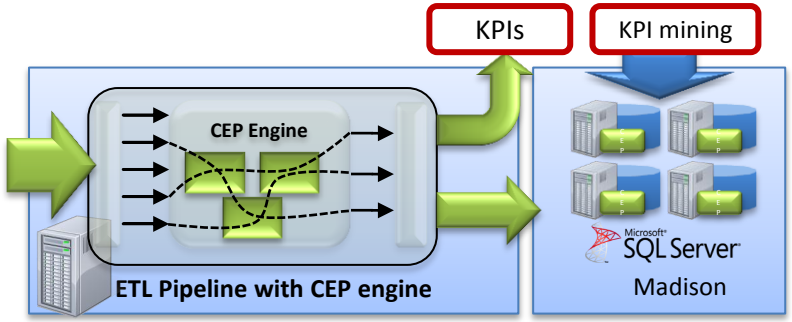
Scenario 2: Embed CEP in Application



Scenario 3: CEP Enabled Device

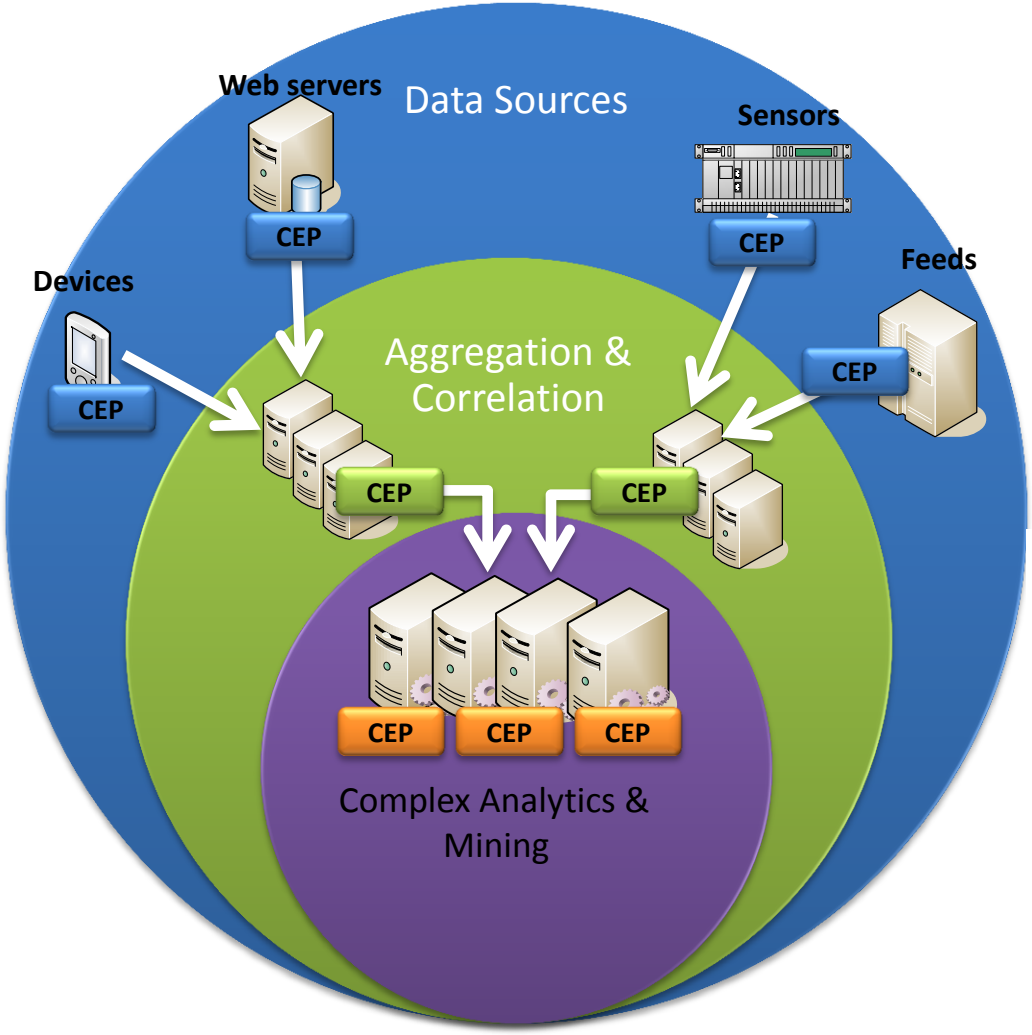


Scenario 4: Operational Intelligence w/ CEP



Application Component
Deployment Model

CEP Deployment alternatives



Event processing engines are deployed at multiple places on different scales

- At the edge – close to the data source
- In the mid-tier – consolidate related data sources
- In the data center – historical archive, mining, large scale correlation.

CEP for lightweight processing and filtering

CEP for aggregation and correlation of in-flight events

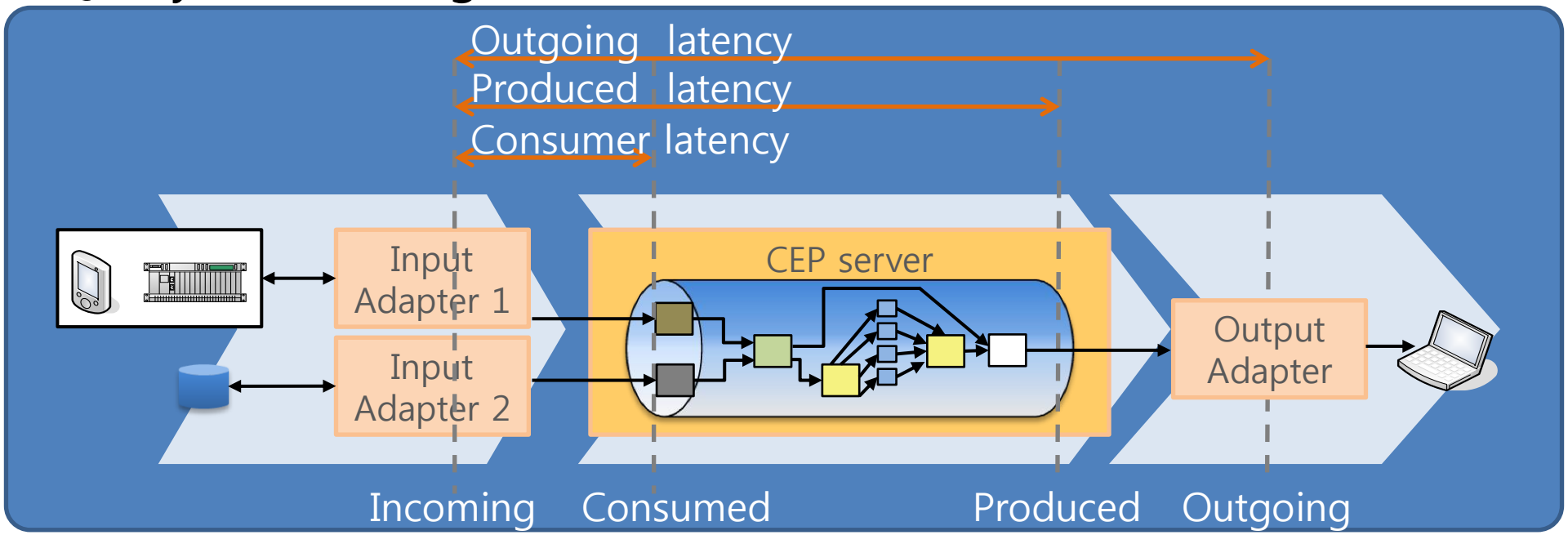
CEP for complex analytics including historical data

Application Component
Monitoring & Event Flow Debugger

Monitoring StreamInsight

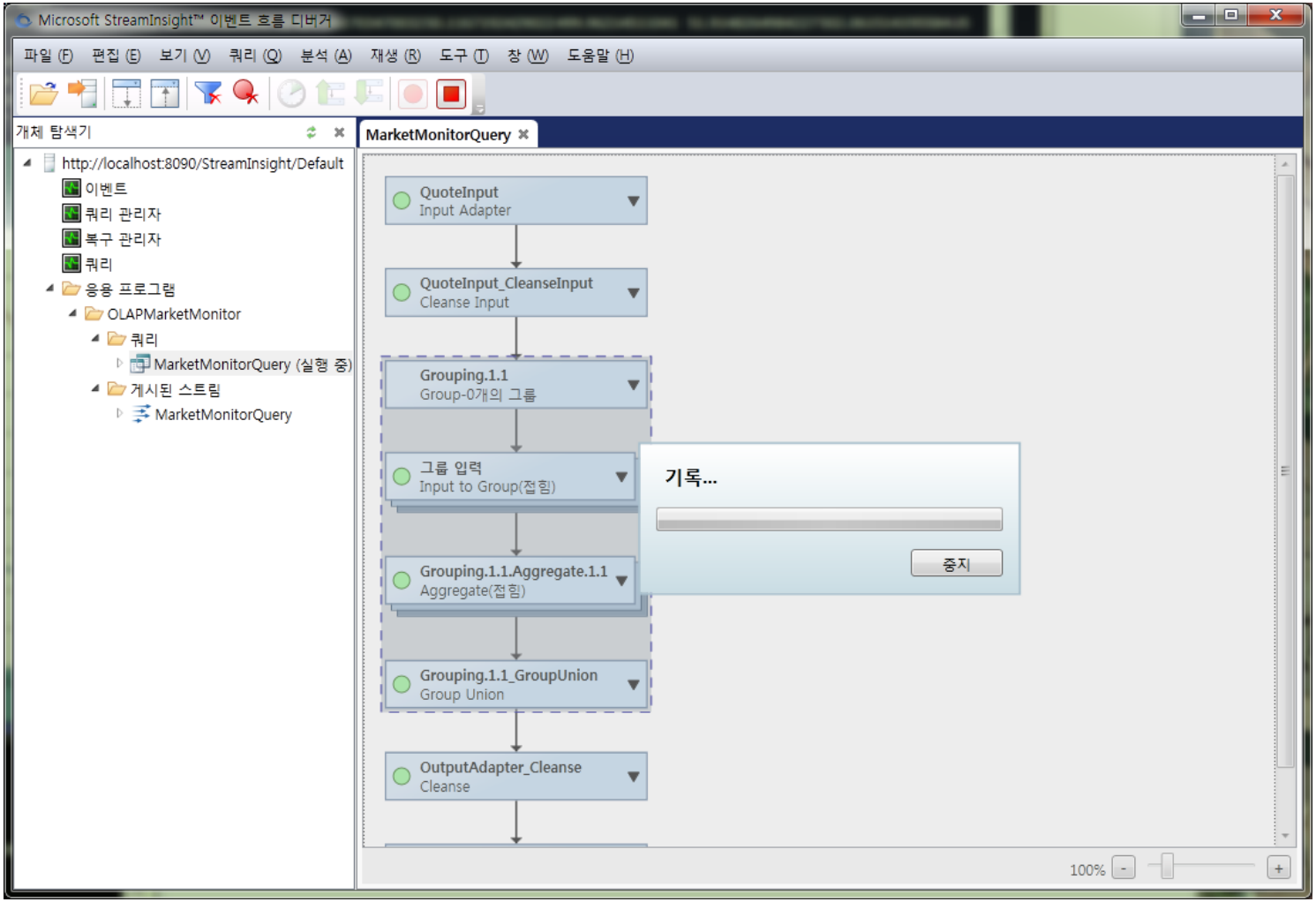
- Need to track : Overall health of system, Query performance
- Diagnostic views using ManagementService API
  - Use URIs for resource naming
  - GetDiagnosticView(), SetDiagnosticSettings(), ClearDiagnosticSettings()
  - Can be accessed via PowerShell

Query Monitoring Points



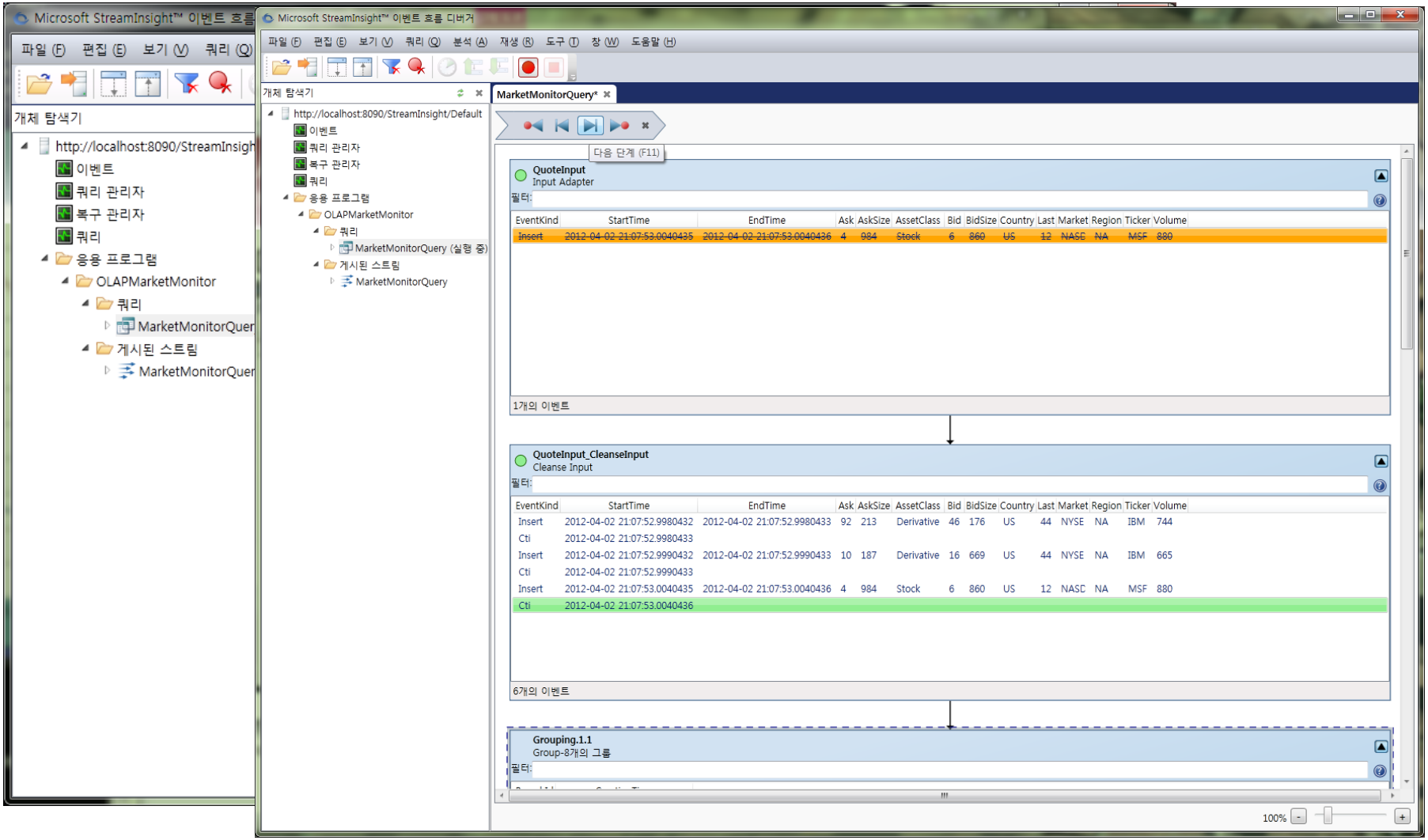
Application Component
Monitoring & Event Flow Debugger

Event Flow Debugger Demo (TechEdUS2010)



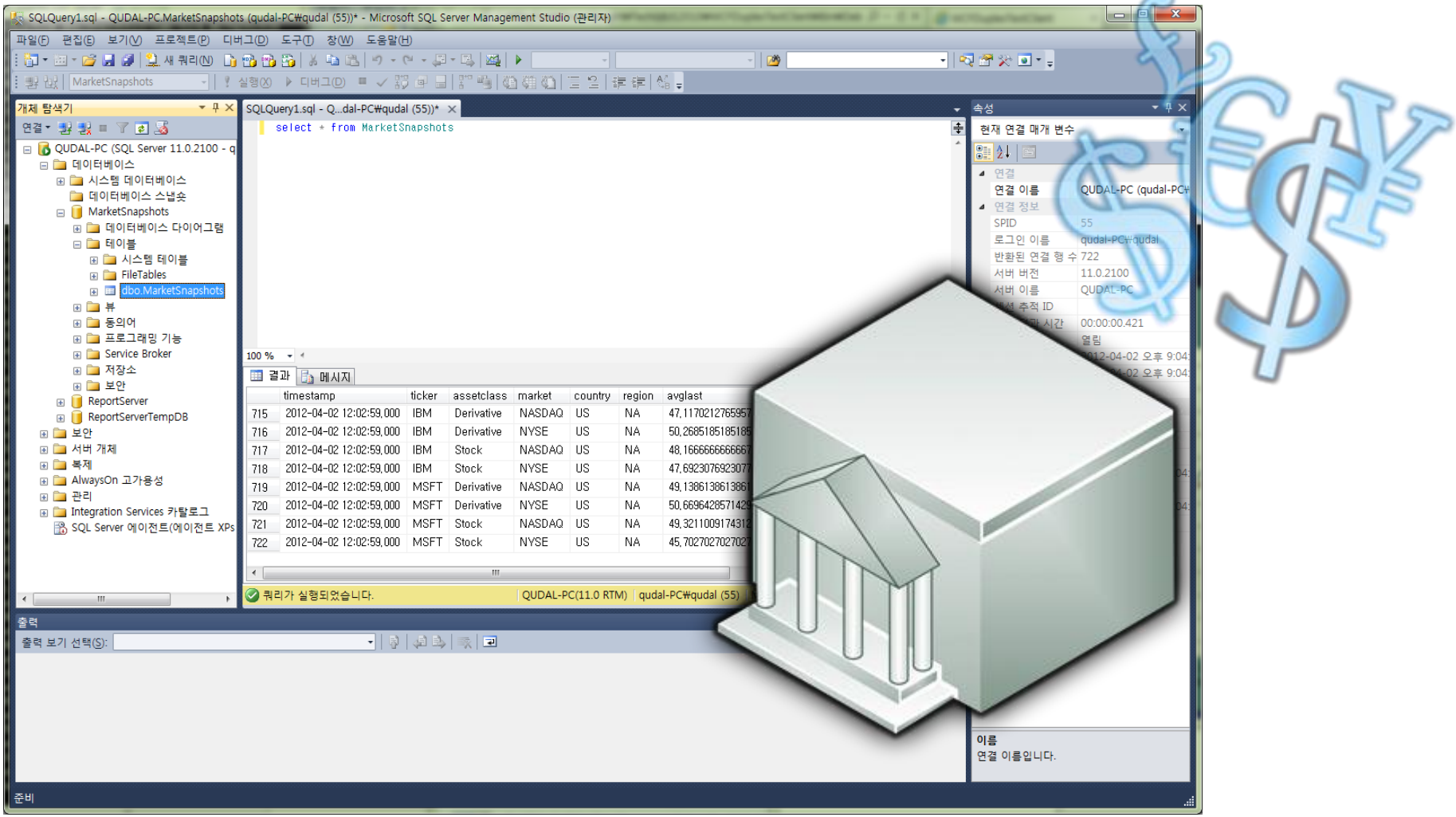
Application Component
Monitoring & Event Flow Debugger

Event Flow Debugger Demo (TechEdUS2010)



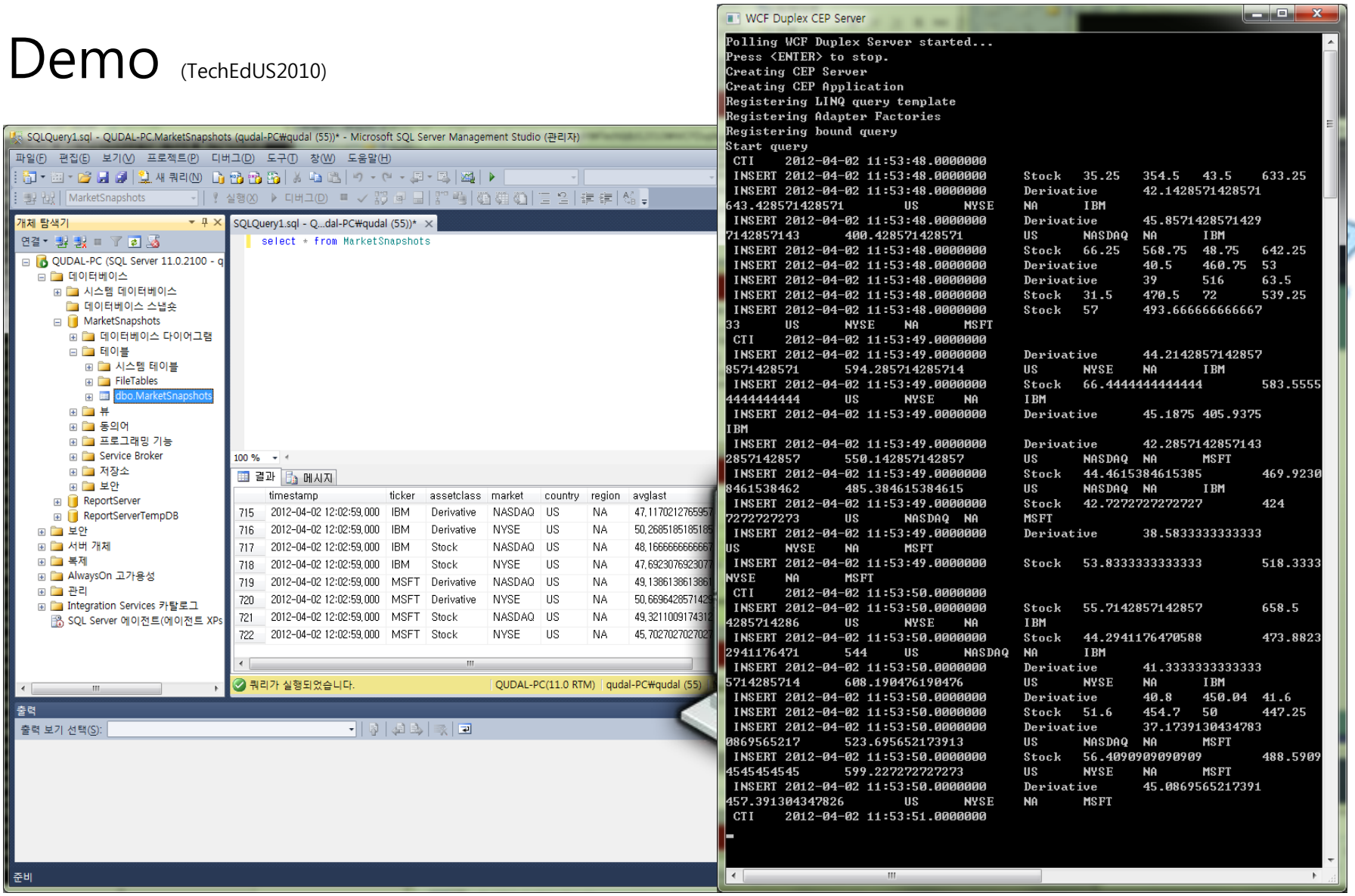
Usage Example
Capital Markets

Demo (TechEdUS2010)



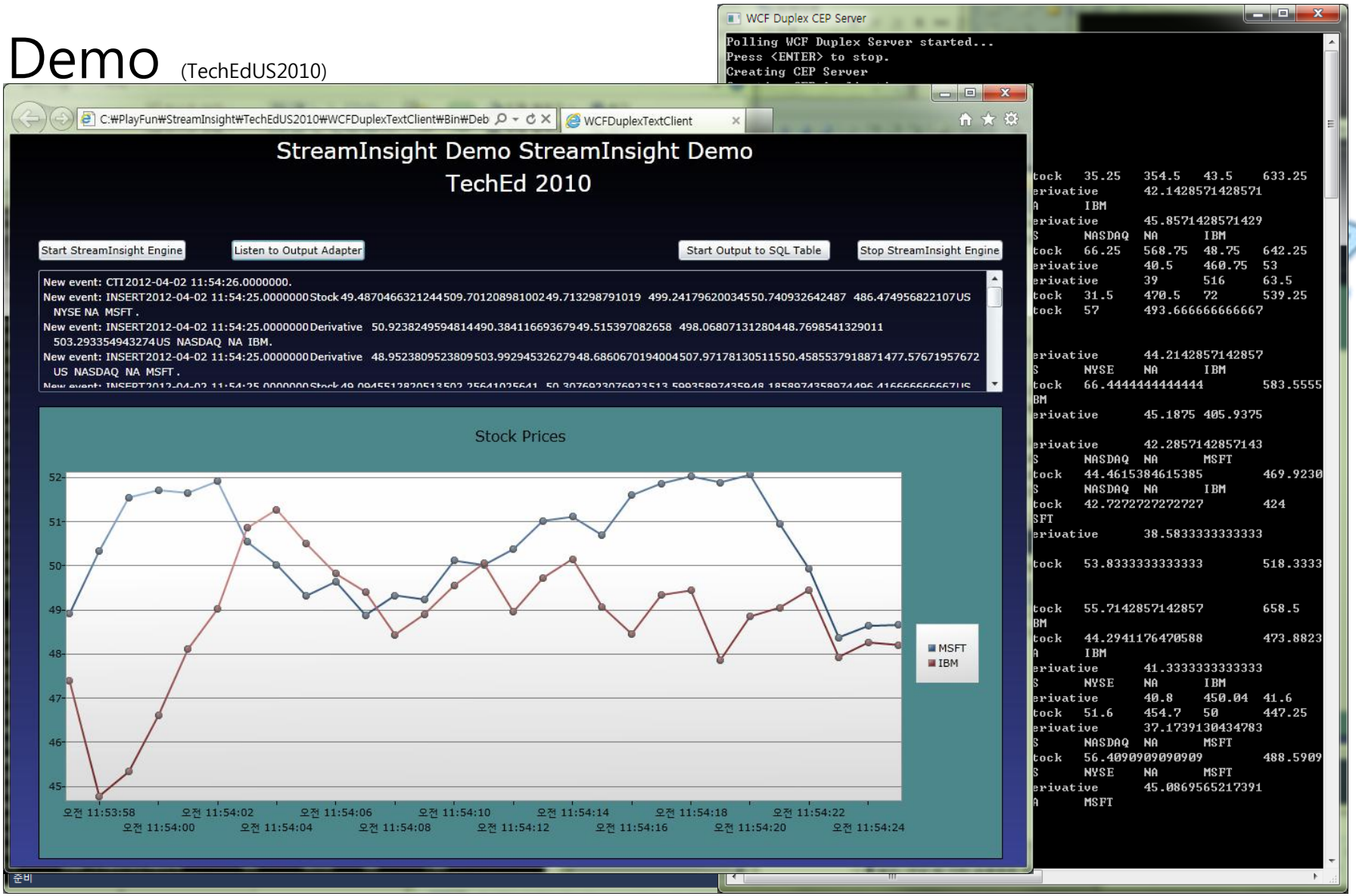
Usage Example
Capital Markets

Demo (TechEdUS2010)



Usage Example
Capital Markets

Demo (TechEdUS2010)



Usage Example

Manufacturing:
- Sensor on plant floor
- React through device controllers
- Aggregated data
- 10,000 events/sec

Web Analytics:
- Click-stream data
- Online customer behavior
- Page layout
- 100,000 events /sec

Financial Services:
- Stock & news feeds
- Algorithmic trading
- Patterns over time
- Super-low latency
- 100,000 events /sec

Power, Utilities:
- Energy consumption
- Outages
- Smart grids
- 100,000 events/sec

Asset Instrumentation for Data Acquisition, Subscriptions to Data Feeds

Data Stream

Visual trend-line and KPI monitoring
Batch & product management
Automated anomaly detection
Real-time customer segmentation
Algorithmic trading
Proactive condition-based maintenance

Data Stream

Stream Data Store & Archive

Asset Specs & Parameters

Lookup

Event Processing Engine
- Threshold queries
- Event correlation from multiple sources
- Pattern queries

Competitive Landscape

Progress – Apama	IBM – Open ESB
West Global – Total Insight	Coral8
Bristol Technology – SenActive	GmbH – Real Time Monitoring
Streambase	Agent Logic
Inetco	Tibco - BusinessEvents
Oracle – Fusion Middleware	Event Zero
Syndera	Rulecare
LG CNS Event Pro	Sybase Aleri Event Stream Processor

Industry Forum: http://complexevents.com

Resources

- StreamInsight Website
  - http://www.microsoft.com/sqlserver/en/us/solutions-technologies/business-intelligence/complex-event-processing.aspx
- StreamInsight Books Online
  - http://msdn.microsoft.com/ko-kr/library/hh750619(v=sql.10).aspx
- StreamInsight Forums
  - http://social.msdn.microsoft.com/Forums/en-US/streaminsight/threads
- StreamInsight Official blog
  - http://blogs.msdn.com/b/streaminsight/
- SQL Server 2008 R2 Update for Developers Training Kit (May 2011 Update)
  - http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=16281

SQL Unplugged 2012 : 쇼케이스