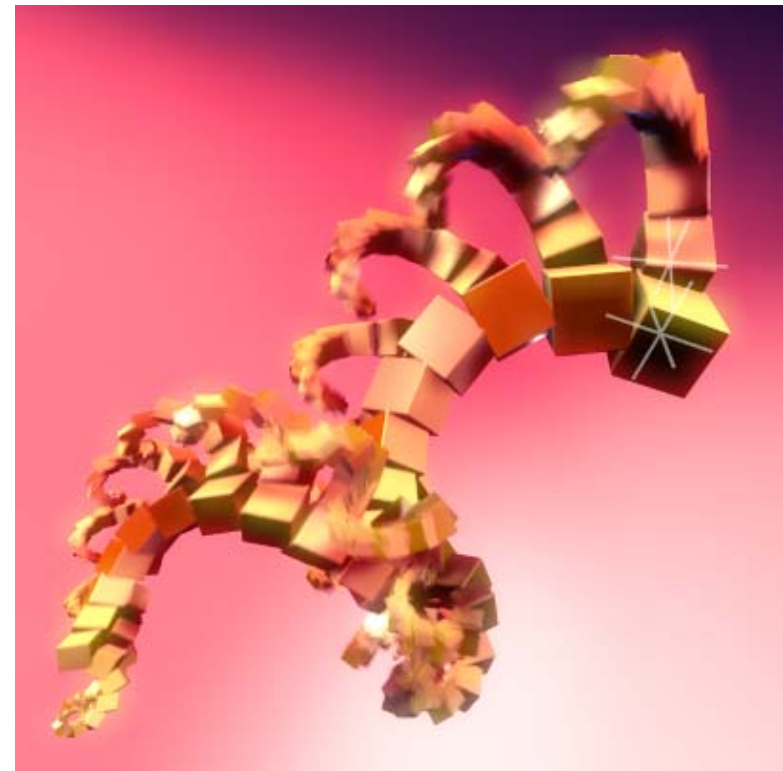




# Lessons for the Future of Event Processing

John Bates



# Apama – History

- 1990s - Research at University of Cambridge, UK
- 2000 - 2005 – Event processing startup, funded by the Carlyle Group
- 2005 – Acquired by Progress Software (\$500m NASDAQ-listed US corp)
- Today – Apama is the fastest-growing division of Progress Software



# Agenda

- Our **academic vision** for event processing in 1990s
- **Lessons learned** from real business applications
  - What did we omit from the academic vision?
  - How much of the vision has come to pass so far
- **Architecture** of a commercial event processing platform
- **Killer applications** of event processing
- **Predictions** for the event processing market

# Pre-1990s Event Processing Prior Art

- Active database
  - ECA rules
- Hard/soft real-time systems
  - Handling events within specific latencies
- Operating systems
  - Event handlers in response to registration of interest
- Window systems
  - Callbacks based on registration of interest
- Rules engines/expert systems/prolog
  - Inference “intelligence”
  - Rete optimizes which rules to run in response to queries

# Our Research Focus [1]

- Building event-driven applications
  - Extending events as “first class citizens” in distributed programs
  - Making it possible to specify and detect composite/complex event sequences to drive actions
  - Using event-based programming for rapid application development
- Making event processing high performance
  - Multi-dimensional matching and complex event sequencing
- Complex event languages
  - Event Processing Language (EPL) – event expressions and actions
  - Embedding into other languages, e.g. Perl, Python
  - Extending SQL with EPL extensions for event storage/retrieval/analysis
  - Accessing from C++ and Java

# Our Research Focus [2]

- **Event-driven architectures**
  - Event-driven components as **reusable building blocks** for distributed applications
  - Components don't have to know about each other
  - Composite event-based **federators** provided the “glue” to bind applications/services together in various different ways
- **Event-Services that “follow-me” and migrate “in the cloud”**
- **Storing and retrieving events**
  - Time series event storage, retrieval and replay

# Our Research Focus [3]

- Application areas
  - Tracking large numbers of moving objects, detecting complex scenarios and responding
    - Location and context aware computing
    - Virtual and augmented reality
    - “Follow-me” multimedia, e.g. the “Intelligent Hospital”
  - Memory prosthesis



# Agenda

- Our **academic vision** for event processing in 1990s
- **Lessons learned** from real business applications
  - What did we omit from the academic vision?
  - How much of the vision has come to pass so far
- **Architecture** of a commercial event processing platform
- **Killer applications** of event processing
- **Predictions** for the event processing market



# Lessons Learned in Real Business Applications [1]

- The world is often “not ready”
  - Ideas from 10-15 years ago are only just starting to happen
  - Example: **location and context aware** applications
- The lowest hanging fruit are applications that generate enhanced profit from competitive advantage in existing systems
  - **Trading in Capital Markets** is top of the list !!
- Initially firms are unlikely to fully re-architect their entire systems as event-driven
  - Must **connect non-intrusively** to their existing systems



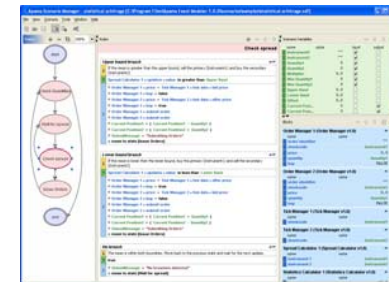
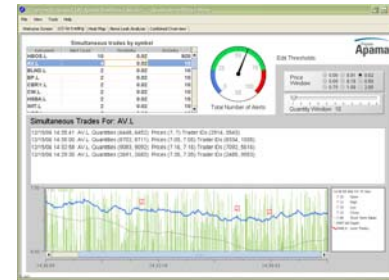
# Lessons Learned in Real Business Applications [2]

- Event-driven development requires customers to “think differently”
  - Complex event expressions can be **complex** for many users
  - Testing event-driven applications can be **complex** – there is a need for **special tools**
- Some users will use cutting edge techniques, like **Event Processing Languages (EPLs)** but others want to use **in-house standards** like Java
- Offering just an event processing/CEP development platform can make it too daunting
  - Offering key **vertical applications/accelerators** makes starting less daunting
  - Pre-built **connectors** accelerate usage



# Lessons Learned in Real Business Applications [3]

- **Business users** are as interested as IT users
- “**Business agility**” (i.e. being able to quickly create or change complex logic in a real-time business) is as important as performance and latency
- **Business analysts** – don’t want to “program” but demand **graphical tools** to model and change event processing scenarios
- **Business executives** want **parameterizable dashboards** for visibility into what is happening in their business

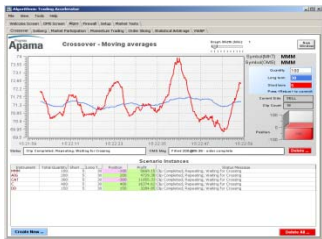


# Agenda

- Our **academic vision** for event processing in 1990s
- **Lessons learned** from real business applications
  - What did we omit from the academic vision?
  - How much of the vision has come to pass so far
- **Architecture** of a commercial event processing platform
- **Killer applications** of event processing
- **Predictions** for the event processing market

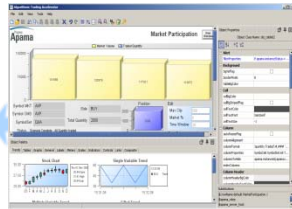
# Apama Platform Overview

## Dashboards

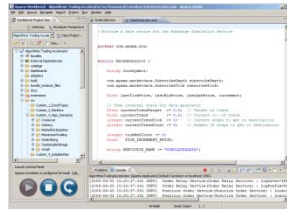


## Developer Tools

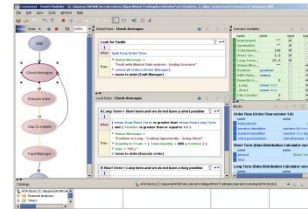
### Dashboard Builder



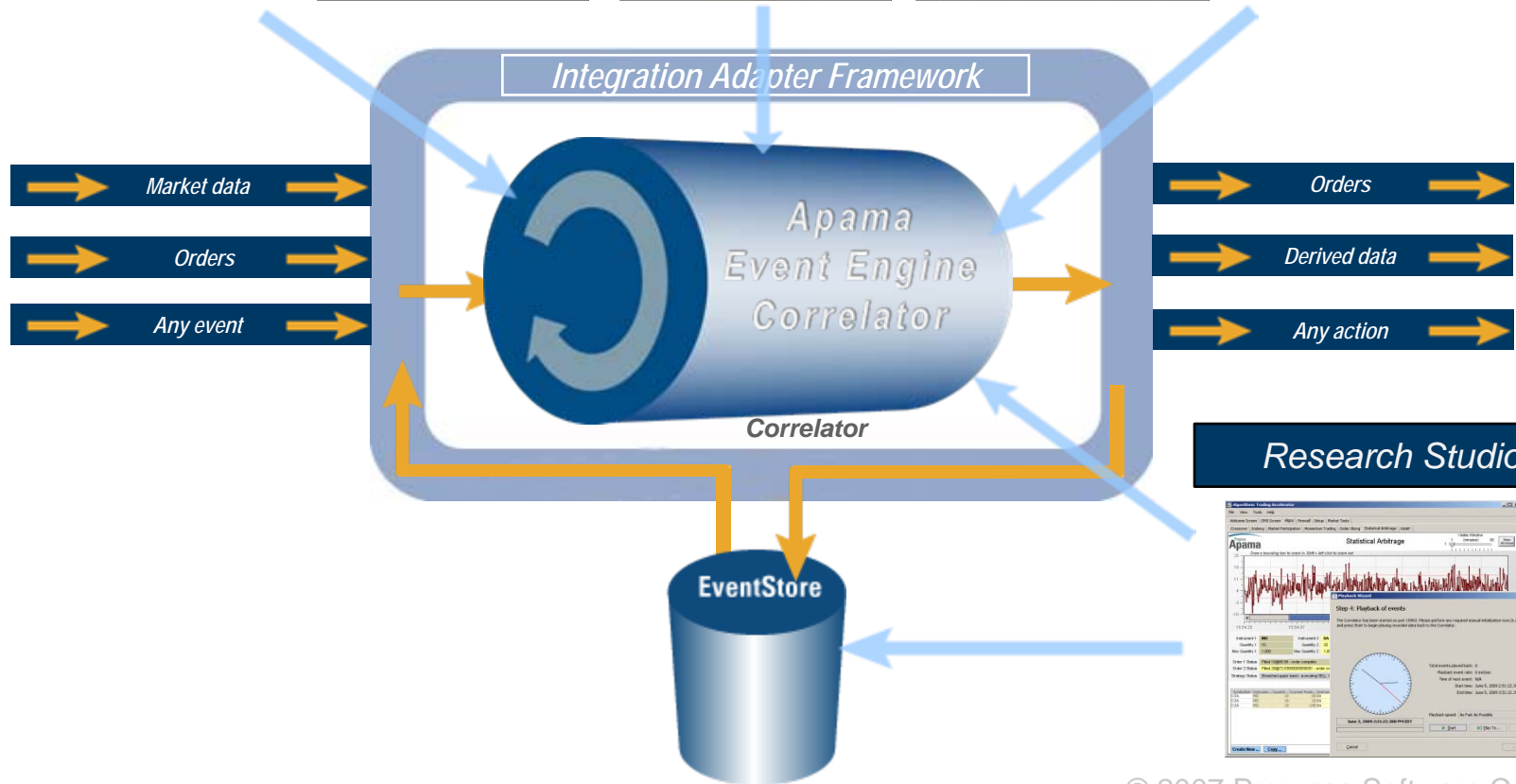
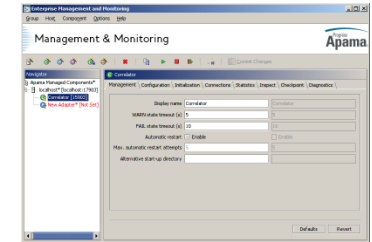
### Apama Studio



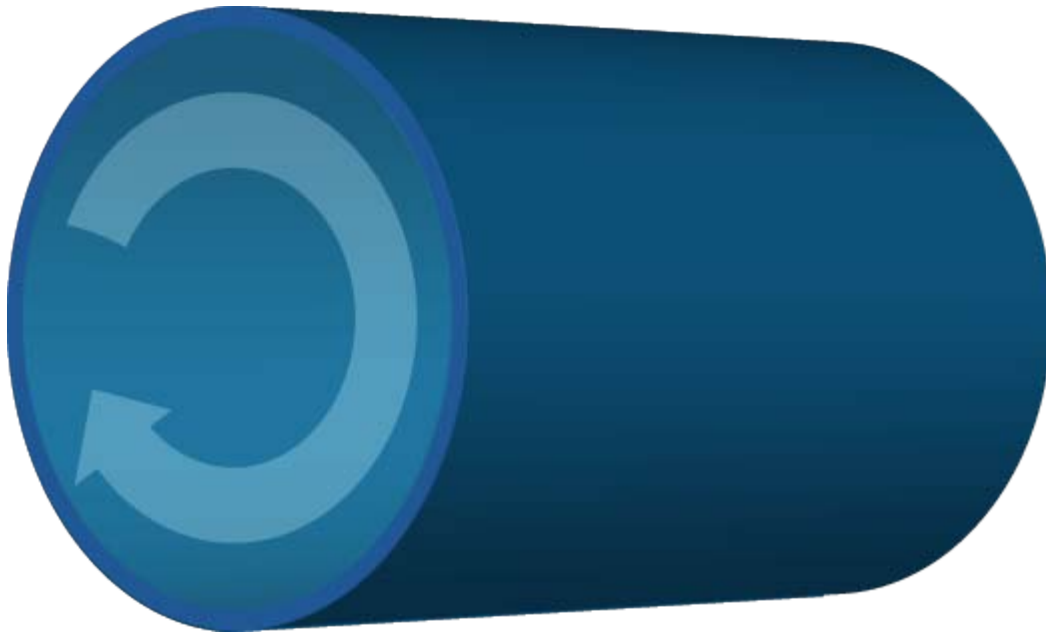
### Event Modeler



## Manage/Monitor

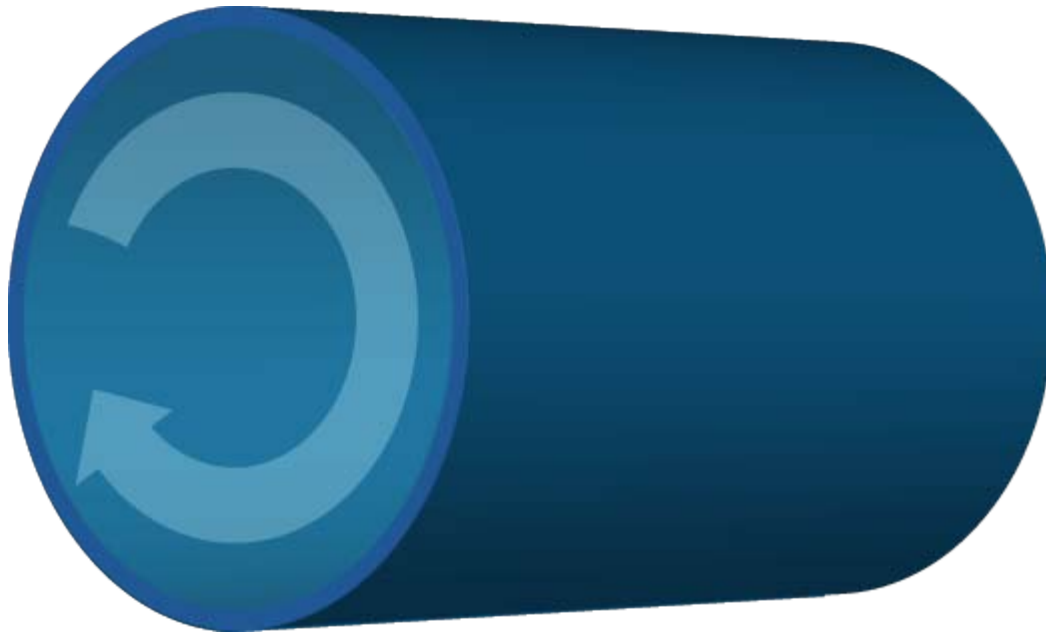


# Inside the Correlator



- *Hypertree:  
Dynamic Fishing Nets*
- *Deterministic with  
Micro Threading Architecture  
“mThreads”*
- *Imperative EPL Model  
with declarative event expressions*
- *Manage Application State  
within the engine*
- *Deploy and Un-Deploy  
EPL on the fly*
- *Internal Clocking =  
Back-Testing and Simulation*

# Inside the Correlator



- *Multiple event contexts to support a multi-core architecture*
- *Abstraction of Contexts to Threads (Thread Pool)*

# Example EPL

---

- Logical operators: and, or, not

```
on StockTick("IBM.N", price >100.0):t1 and
  (StockTick("MSFT.O", price <65.0):t2 or
   StockTick("MSFT.O", price >70.0):t2) and
  not StockTick("MSFT.O", price >80.0) ...
```

- Specify temporal sequencing and constraints using 'followed-by' and 'within' operators

```
on all NewsArticle(symbol="IBM.N") ->
  StockTick(symbol="IBM.N", price<100.0) ->
  Trade("IBM.N", volume >5000, *) within(2.0) doStuff;
```

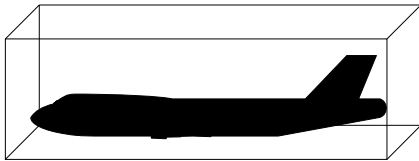
- Take actions, e.g. "Emit" events to channels

```
on all StockTick(symbol="IBM.N"):tick {
  sterlingPrice := tick.price * exchangeRate;
  emit tick to "dollar_prices";
  emit StockTick("IBM.N", sterlingPrice) to "sterling_prices";
}
```



# Event Monitoring is Multi-dimensional

- Support Cartesian coordinate system
- *Location* type
  - Encapsulate spatial container ( X1 , Y1 , X2 , Y2 ) or ( X1 , Y1 , Z1 , X2 , Y2 , Z2 )



```
event MovingObject {  
    string name;  
    location loc;  
}  
MovingObject ( `Plane 1` ,  
    30.12,23.67,41.96 ,  
    32.23,25.45,43.34 );
```

- Provide operations and behaviour, e.g.
  - *Intersection*
  - *Within*

# Example EPL Snippet – Monitor Collisions

```
event MovingObject {                               // The event definition
    string name;
    location loc;
}

Monitor CollisionDetector {                        // The main monitor
    MovingObject mo, co;                          // Store the latest events for
                                                    // "Plane 1" and any nearby objects

    action onload {
        // Look for all updates of "Plane 1" followed by an update from an
        // object that is within 2.0 Meters and take an action. However,
        // if an update is from "Plane 1" do not take an action and stop
        // using "Plane 1's" old position. Restart with the new location.

        on all MovingObject (name = `Plane 1`):mo {
            on all MovingObject (loc = mo.loc.expand(2.0)):co and not
                MovingObject (name = `Plane 1`) {

                // Take whatever action necessary
                print co.name + " is near " + mo.name;
            }
        }
    }
}
```

# Example 2 – Monitor ALL objects

```
event MovingObject {                               // The event definition
    string name;
    location loc;
}

Monitor CollisionDetector {                         // The main monitor
    MovingObject mo, co;                            // Store the latest events for

    action onload {

        // Look for all updates of any object followed by a second update of
        // an object that is within 2.0 Meters and take an action. However,
        // if the second update is from the same object do not take an
        // action and stop using the old position of the first object;
        // instead restart with the new location.

        on all MovingObject ():mo {
            on all MovingObject (loc = mo.loc.expand(2.0)):co and not
                MovingObject (name = mo.name) {

                // Take whatever action necessary
                print co.name + " is near " + mo.name;
            }
        }
    }
}
```

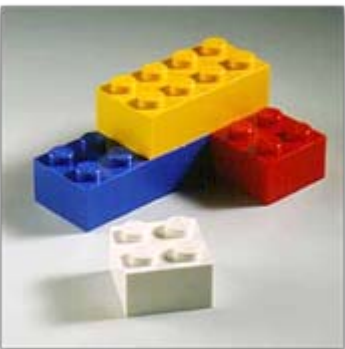
# User Roles



**Business Executives**



**Business Analysts**



**IT Users**

**Dashboards**

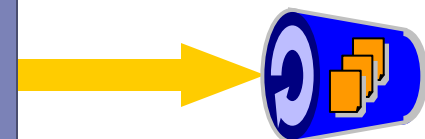
- Business visibility
- Opportunities & threats
- Create instances of scenarios

**Strategy Modeling, Dashboard Creation, Back-testing, Deployment, Evolution**

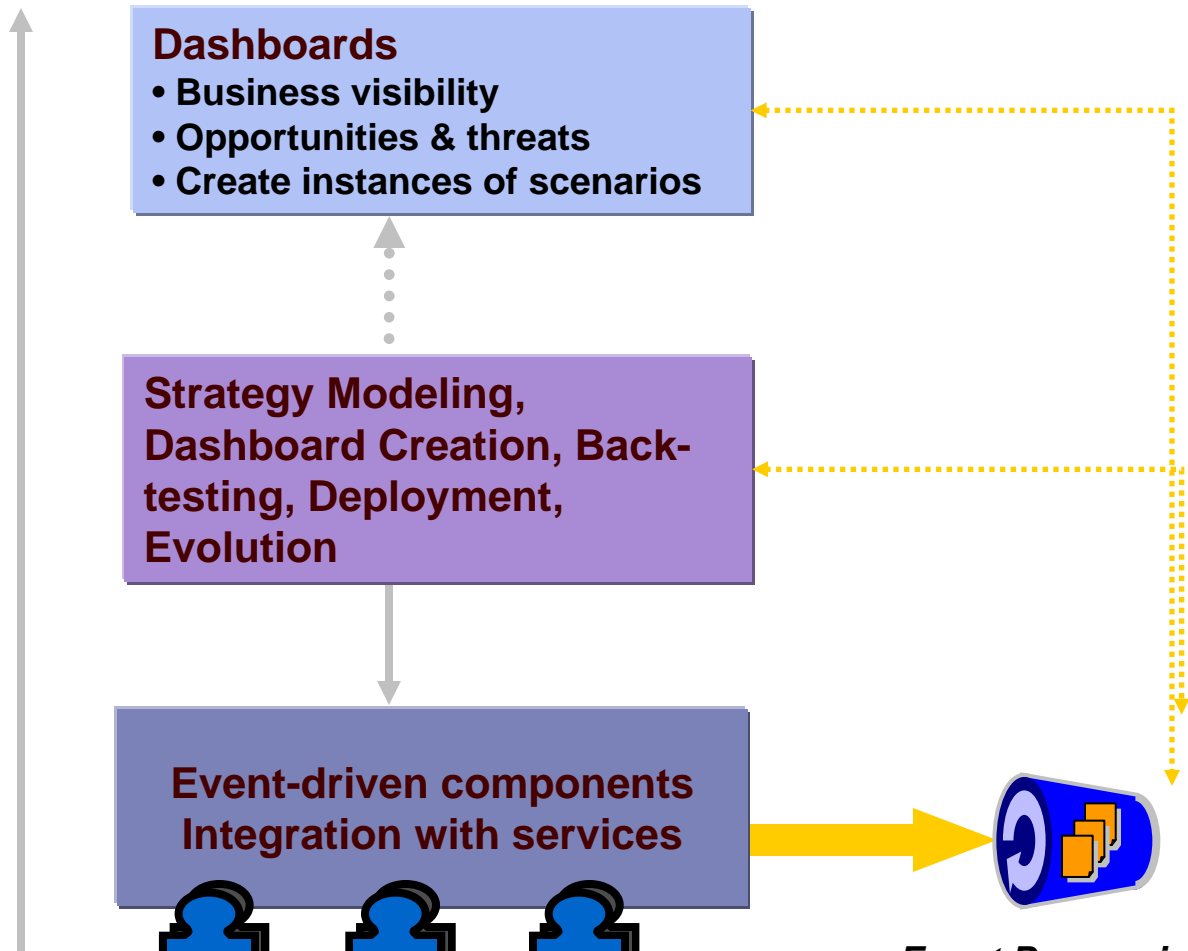
**Event-driven components Integration with services**



Third-party Analytics    In-house functions    Enterprise services



**Event Processing Platform**



# Event Modeler Overview

The screenshot displays the Apama Scenario Manager interface. The main workspace shows a 'Block Wiring' diagram with the following components and connections:

- Order Flow** block sends `trade >>> data` to both **Short Term** and **Long Term** blocks.
- Market Depth** block is positioned to the left of the **Iceberg Timer** and **Order Timer** blocks.
- Order Manager** block sends `order execution >>> order status` to the **Position Calculator** block.
- The **Position Calculator** block sends `position >>> position` to the **P&L Calculator** block.

On the right side, there are two tool windows:

- Scenario Variables**: A table listing variables with their values and input/output status.
- Blocks**: A list of available blocks with their parameters.

At the bottom, the **Catalogs** window shows the details for the **Market Depth v1.0** block:

**Market Depth**  
Manages subscriptions to market depth feeds (bid, mid, ask price and quantity)  
v1.0, 2 June 2004 (C. Reed)  
Intended for use with Event Modeler 2.1 or higher

**blocks [C:\Apama\EventModeler\blocks]**

- Market Depth v1.0
  - input feeds
  - output feeds
  - parameters
  - operations

“SmartBlocks” encapsulate pre-packaged modules made available to non-programmers.

# Event Modeler Overview

Complex event rules – composed of **WHEN** expression **THEN** action

Input and Output Variables – enable business executives to create “instances” of complex scenarios via dashboards

Finite state machine state flow. Each state can have different active rules.

The screenshot displays the Apama Scenario Manager interface with several key components:

- States Panel:** A finite state machine diagram showing states: start, end, Check Averages, Execution Decision, Depth Check, Timed Order, Time Slice Order, Time Delay, and Clip Complete. Arrows indicate the flow between these states.
- Check Averages Rule:**

```

Check Averages
If Short Term > Long term and we do not have a long position
{
  ( Short Term » statistics » mean is greater than
  Long Term » statistics » mean ) and ( Position
  is less than or equal to 0.0 )
  !
  • Status Message =
  "Position is Short - Trading Opportunity - Going Long"
  • Quantity to Trade = ( Clip Quantity + ABS ( Position ) )
  • Side = "BUY"
  • move to state [Execution Decision]
}

If Long Term > Short term and we do not have a short position
{
  ( Long Term » statistics » mean is greater than
  Short Term » statistics » mean ) and ( Position
  is greater than or equal to 0.0 )
  !
  • Status Message =
  "Position is Long - Trading Opportunity - Going Short"
  • Quantity to Trade = ( Clip Quantity + ABS ( Position ) )
  • Side = "SELL"
  • move to state [Execution Decision]
}
            
```
- Scenario Variables Panel:** A table for defining input and output variables.
 

name	value	input	output
Instrum...	"PRGS"	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Clip Qua...	100	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Short Te...	5.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Long Te...	20.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Executio...	"Tim..."	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Number ...	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Time Sli...	5.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Order Ti...	5.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Status ...	""	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Position	Positio...	<input type="checkbox"/>	<input checked="" type="checkbox"/>
OMS Sta...	Order ...	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Quantity...	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Time Sli...	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
- Blocks Panel:** Lists available blocks such as Market Depth (Market Depth version 1.0) and Order Flow (Order Flow version 1.0) with their respective parameters.
- Catalogs Panel:** Shows a tree view of installed blocks and catalogs.
- Market Depth Block Details:**

```

Market Depth
Manages subscriptions to market depth feeds
(bid, mid, ask price and quantity)

v1.0, 2 June 2004 (C. Reed)
Intended for use with Event Modeler 2.1 or higher
            
```

# Dashboard Studio

Apama Dashboard Studio RevenueAssurance.rtv

File Edit View Tools Help

LBS Service QoS Cinema Service QoS Music Service QoS

Revenue Loss % Over Billed %

Service Latencies (sec)

Alerts

Object Palette

Attach to Apama

Attach to: Scenario instance table

Correlator: default

Scenario: RevenueAssurance

Timestamp variable:

Display variables: LBS Throughput

Filter: LBS Throughput

By variable: MUSIC Overcharge, MUSIC Revenue Loss, MUSIC Throughput, OverchargePC

Where value equals: QoS-Cinema, QoS-LBS, QoS-Music

Using time interval:

OK Cancel

- Real-time variables and analytics can be visualized using graphs, charts, tables etc.

- Enables event-driven logic to be visualized in real-time

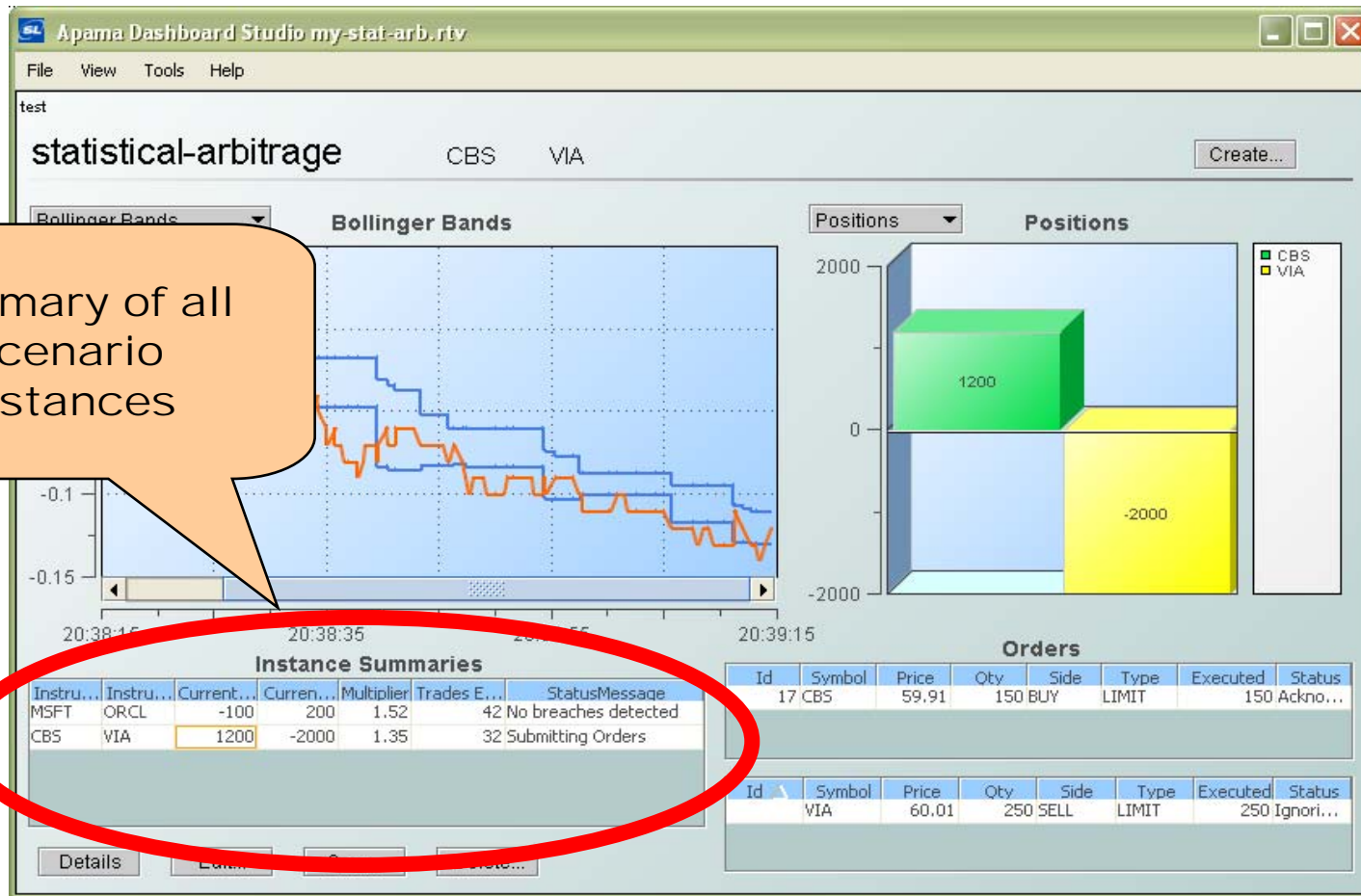
- Variety of deployment options (web, app, ...)

Select from a palette of graphical objects. Each object can be laid out in a graphical dashboard and bound to Apama events

Users can customize the look and feel of all widgets, and specify which Apama event scenarios' output variables to visualize.

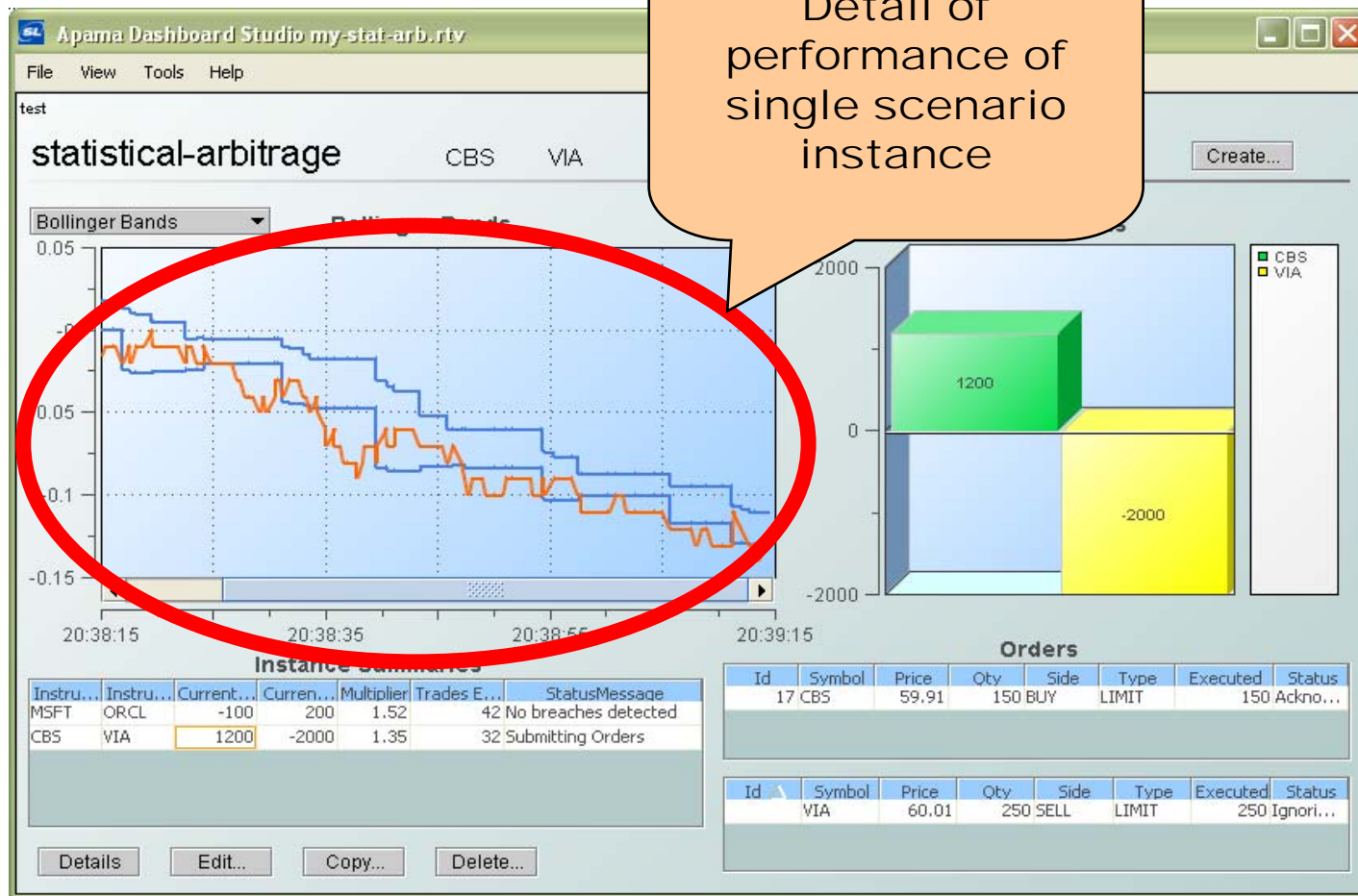


# Sample Dashboard Summaries

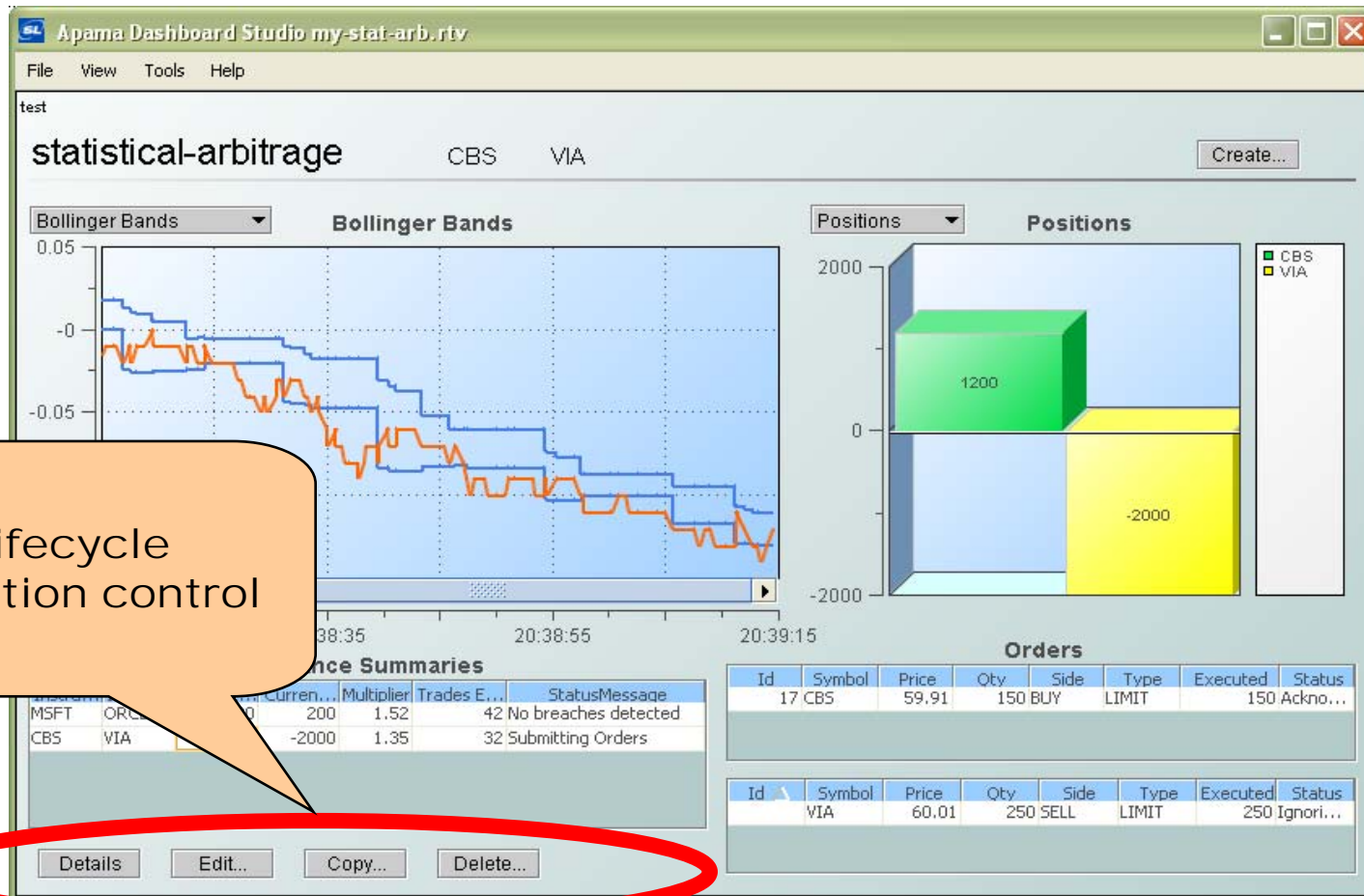




# Sample Dashboard Details



# Sample Dashboard Controls



# Sample Dashboard Controls

The screenshot displays the Apama Dashboard Studio interface. The main window shows a dashboard titled "statistical-arbitrage" with a Bollinger Bands chart and a table of instance summaries. A configuration dialog box is open in the foreground, containing several input fields for defining instance parameters. A red circle highlights the dialog box. A speech bubble points to the dialog box with the text "Forms for defining instance parameters".

Apama Dashboard Studio my-stat-arb.rtv

File View Tools Help

test

statistical-arbitrage

Bollinger Bands

0.05

-0

-0.05

-0.1

-0.15

20:38:15 20:38:35

Instance Summary

Instru...	Instru...	Current...	Curren...	Multip
MSFT	ORCL	-100	200	1.
CBS	VIA	1200	-2000	1.

Details Edit... Copy...

Apama Dashboard Studio my-stat-arb-crea

Instrument1

Instrument1

Quantity1 0

Quantity2 0

MaxQuantity1 0

MaxQuantity2 0

Multiplier 0.0

OK Clear Cancel

Conditions

CBS

VIA

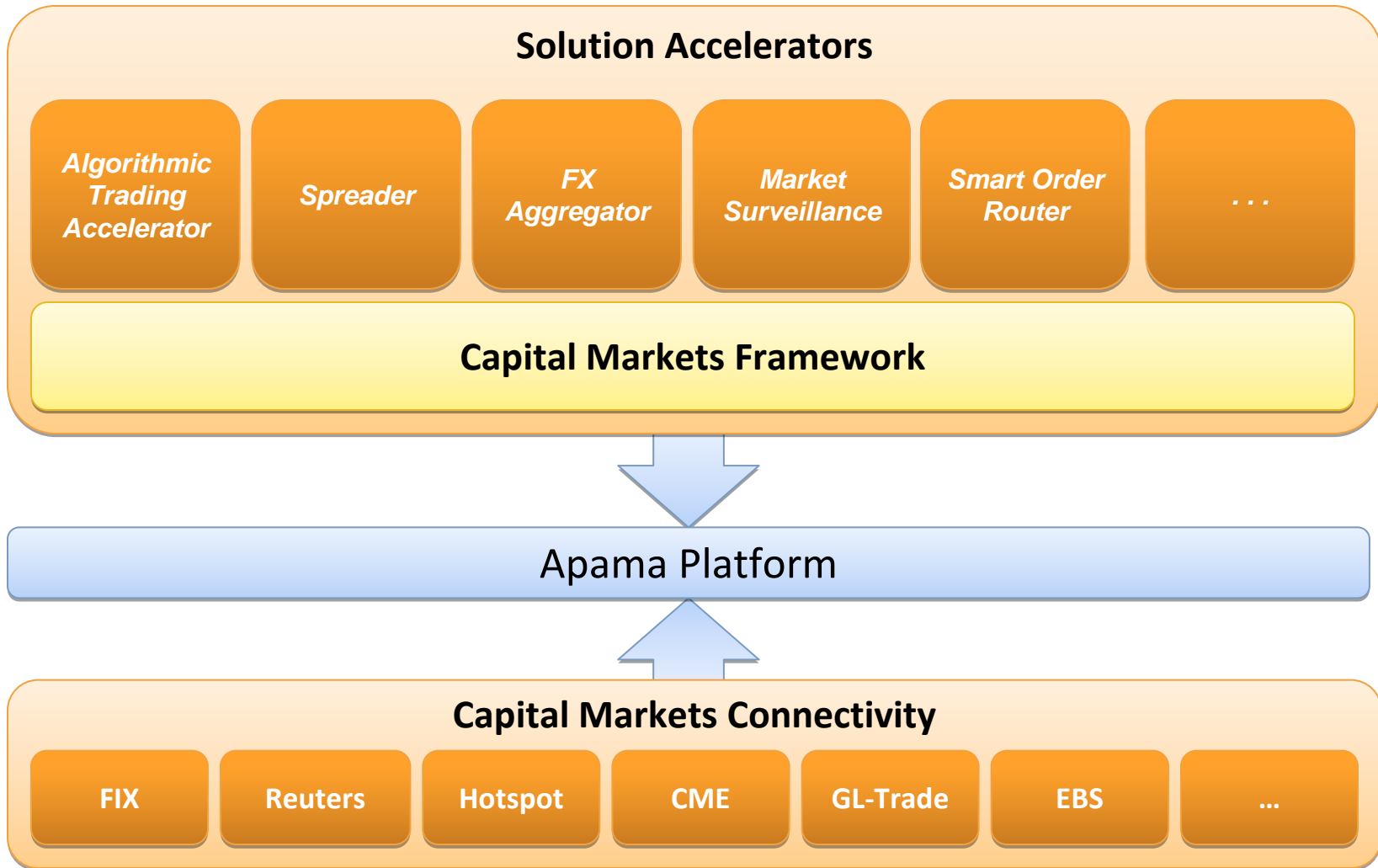
-2000

Type	Executed	Status
LIMIT	150	Ackno...

Side	Type	Executed	Status
LL	LIMIT	250	Ignori...

Forms for defining instance parameters

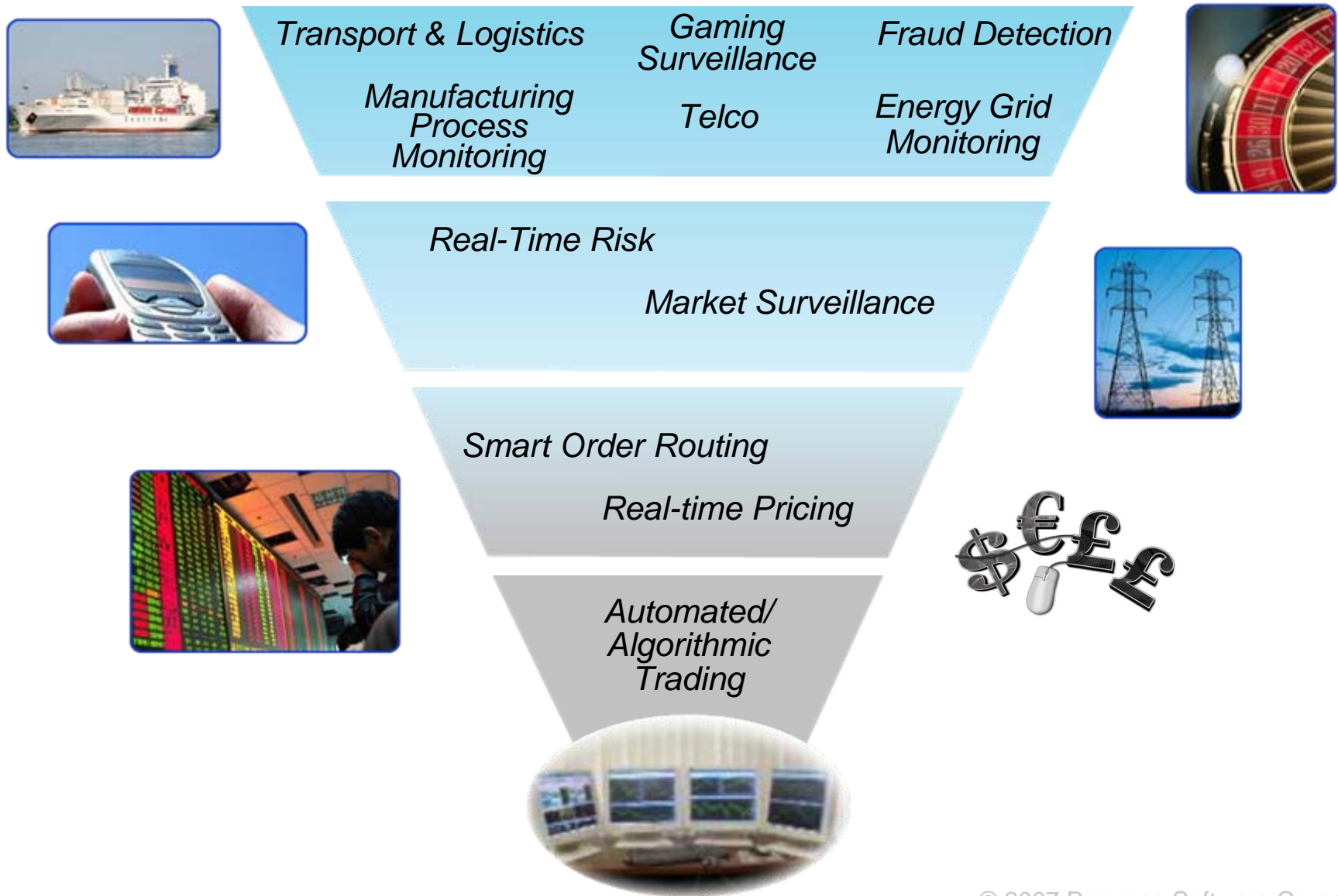
# The Need for Pre-Built Apps and Connectors



# Agenda

- Our **academic vision** for event processing in 1990s
- **Lessons learned** from real business applications
  - What did we omit from the academic vision?
  - How much of the vision has come to pass so far
- **Architecture** of a commercial event processing platform
- **Killer applications** of event processing
- **Predictions** for the event processing market

# Evolution of Applications





# High Frequency Automated Trading

## WHEN

**MSFT** price moves outside 2% of MSFT Moving Average

## FOLLOWED-BY (

My Basket moves up by 0.5%

## AND (

**HPQ's** price moves up by 5%

## OR

**MSFT's** price moves down by 2%

)

)

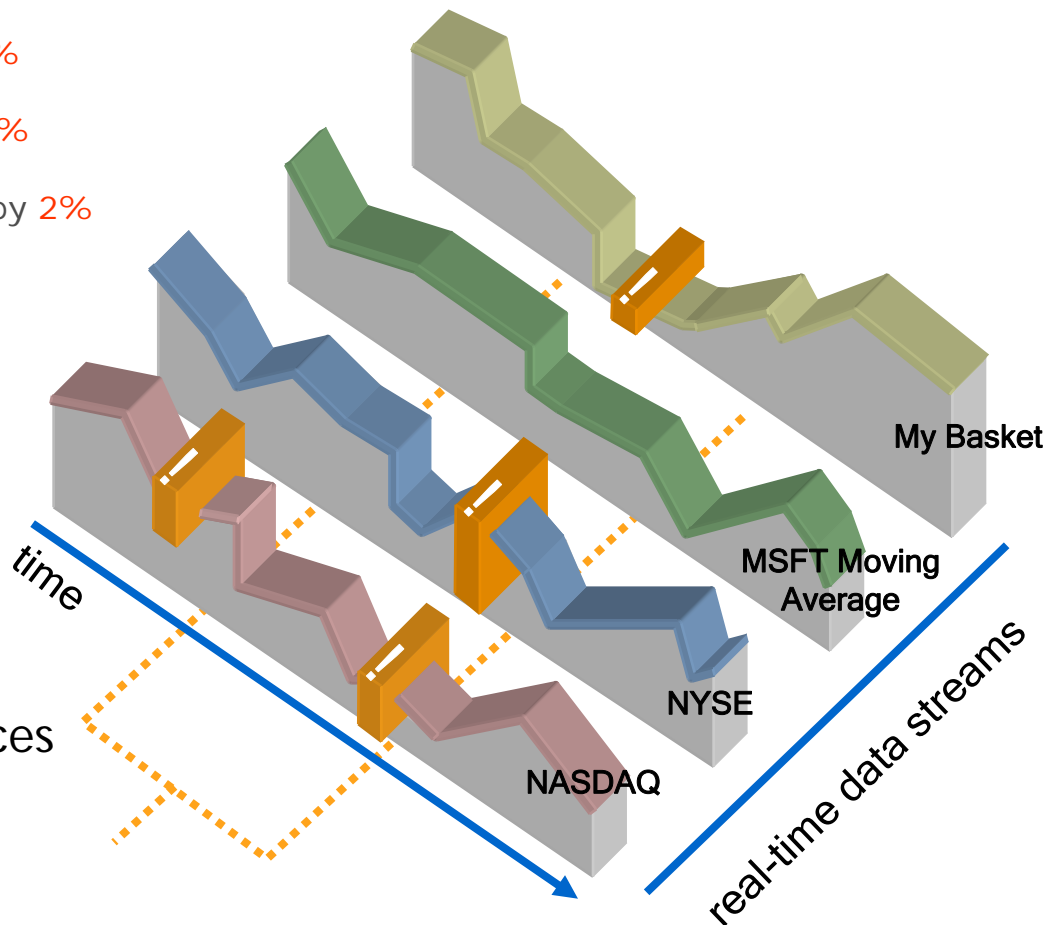
## ALL WITHIN

any 2 minute time period

## THEN

BUY **MSFT**  
SELL **HPQ**

- multiple data streams
- temporal sequencing
- complex event sequences
- real-time constraints
- automated actions



# Market Surveillance

## WHEN

TRADE1 where size > large-trade-size

## FOLLOWED-BY (

TRADE2 where counterparty =

## AND (

TRADE1.TRADER

ABS (TRADE1.PRICE – TRADE2.PRICE)  
LESS THAN wash-trade-threshold

)  
)

## ALL WITHIN

Any 10-second time window

## THEN

ALERT MARKET-CONDUCT (TRADER, YELLOW)

## FOLLOWED-BY

MARKET-CONDUCT (TRADER)

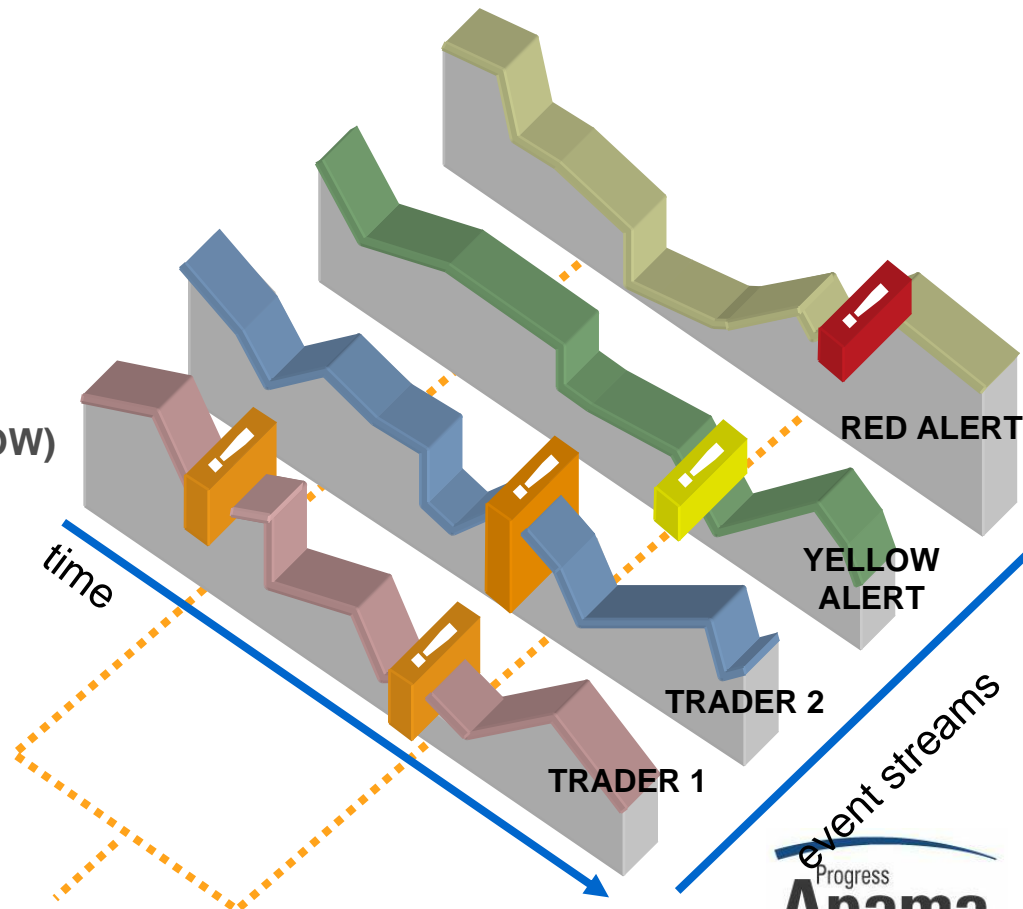
## ALL WITHIN

2 SECONDS

## THEN

ALERT MARKET-CONDUCT (TRADER, RED)

- multiple data streams
- event-driven
- complex event sequences
- temporal constraints
- automated actions





# Energy Grid Monitoring

## Monitoring Rule

**IF**

15 minute Wattage Moving Average decreases by 5%

**FOLLOWED-BY** (

Remote Equipment Alarm

**AND** (

Sub-Station Stability Warning

**OR**

Wattage Spikes Beyond Threshold

)  
)

**ALL WITHIN**

any 30 minute time period

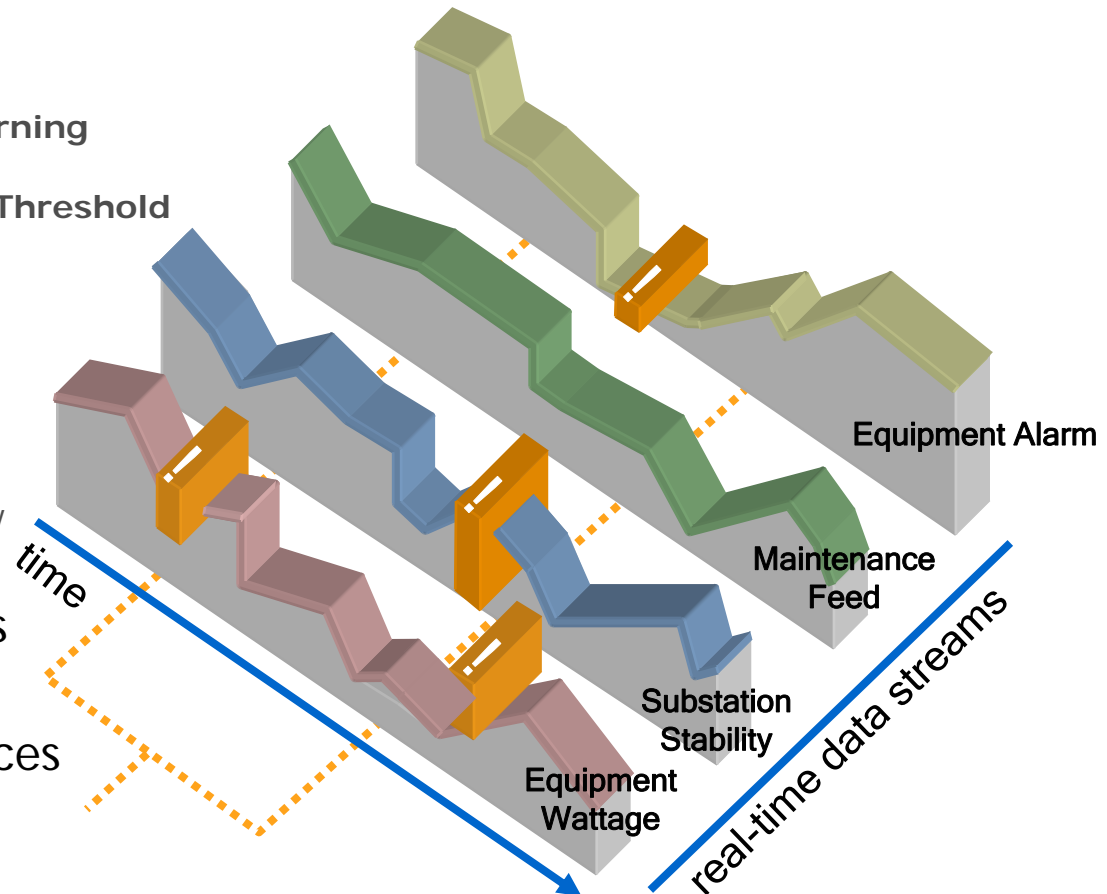
**THEN**

Send an E-Mail

Display on Dashboard

Invoke Resolution Work-Flow

- Real-time data streams
- Temporal sequencing
- Complex event sequences
- Real-time constraints
- Automated actions



# Fraud Detection – Online Loans

## Monitoring Rule

**IF**

The requested loan amount exceeds conventional limits

**FOLLOWED-BY** (

A loan application with same address

**AND** (

Another loan for borrower is denied

**OR**

A loan status inquiry is made

)

)

**ALL WITHIN**

any 4 hour time period

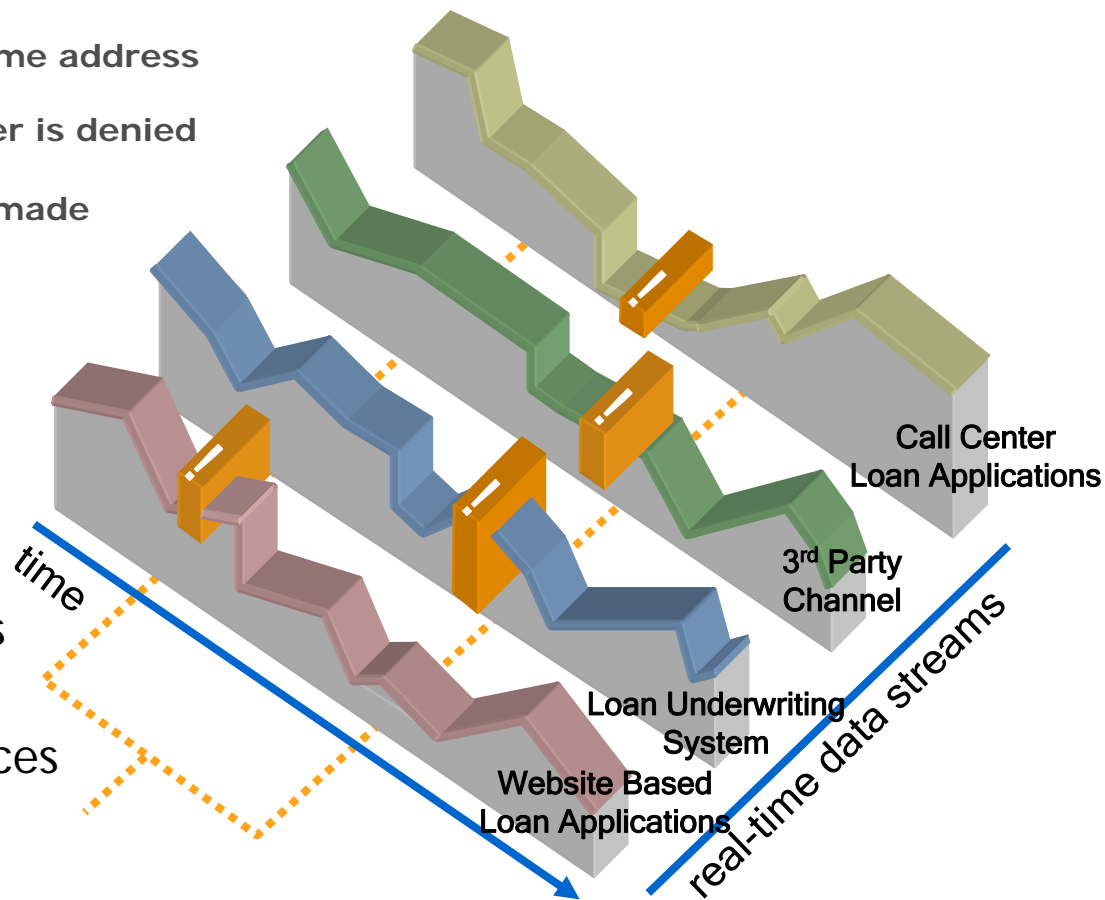
**THEN**

Create an Alert on Dashboard

Deny the Loan

Setup an Investigation

- Real-time data streams
- Temporal sequencing
- Complex event sequences
- Real-time constraints
- Automated actions



# Telco - Location Based Services

## Real-Time Location Based Service Rule: Combining Elements of Time and Location

**IF**

Starbucks has an active promotion

**AND**

The subscriber consents to advertising

**FOLLOWED-BY (**

A subscriber is within 10 minutes  
of a subscriber location

**OR**

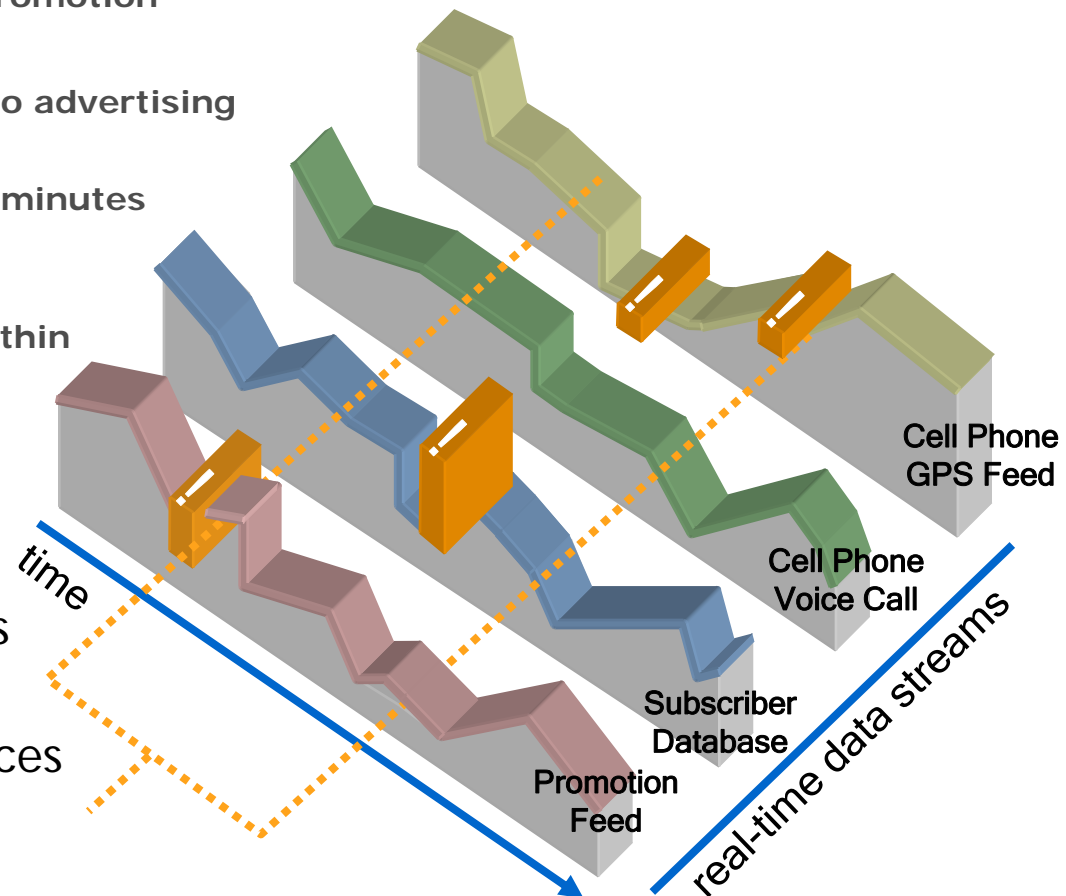
The subscriber lingers within  
the fenced location for  
10 minutes

**)**

**THEN**

Send an SMS Text Message

- Real-time data streams
- Temporal sequencing
- Complex event sequences
- Real-time constraints
- Automated actions



# Telco - Dynamic Location-Based Services

Real-Time Location Based Service Rule:  
Combining Elements of Time and Location

IF **Starbucker** has an active promotion

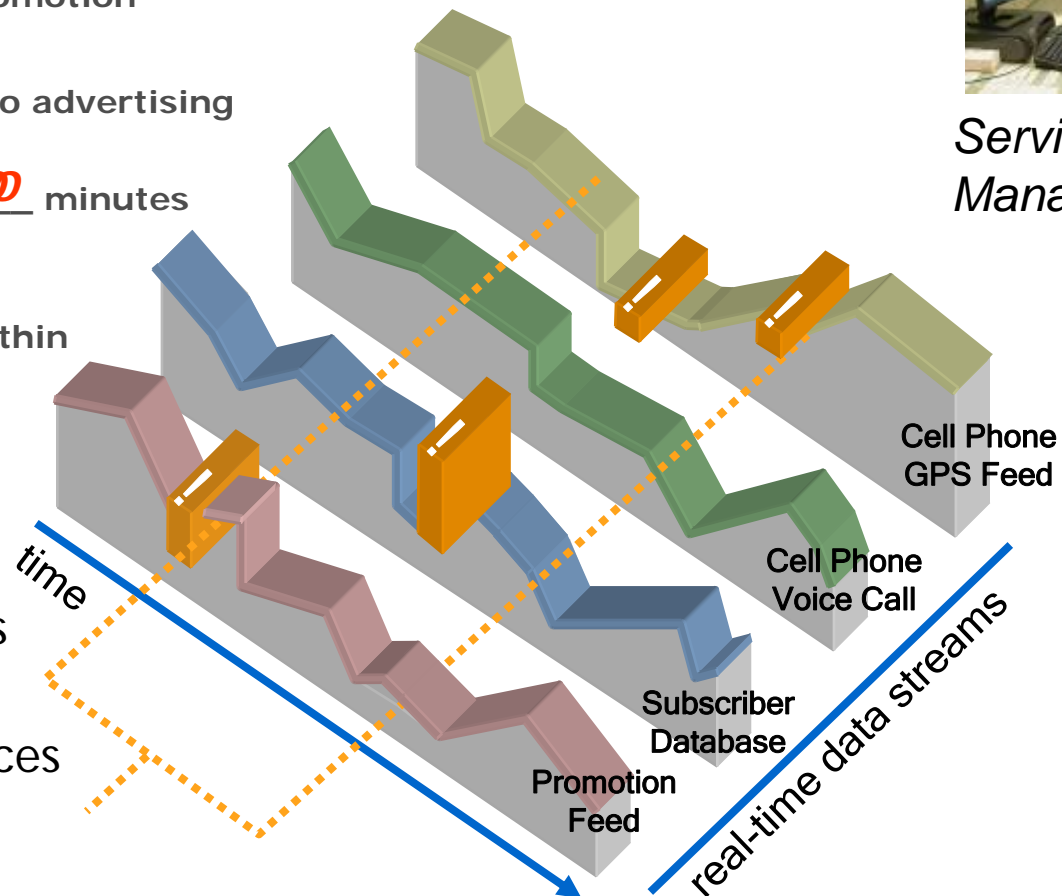
AND  
The subscriber consents to advertising

FOLLOWED-BY ( A subscriber is within 200 minutes of a subscriber location

OR  
The subscriber lingers within the fenced location for 120 minutes

)  
THEN  
Send an SMS Text Message

- Real-time data streams
- Temporal sequencing
- Complex event sequences
- Real-time constraints
- Automated actions



Service Manager X



Service Manager Y

# Transport & Logistics

## Monitoring Rule

### IF

A certain shipment is marked as a "Hot Load"

### FOLLOWED-BY

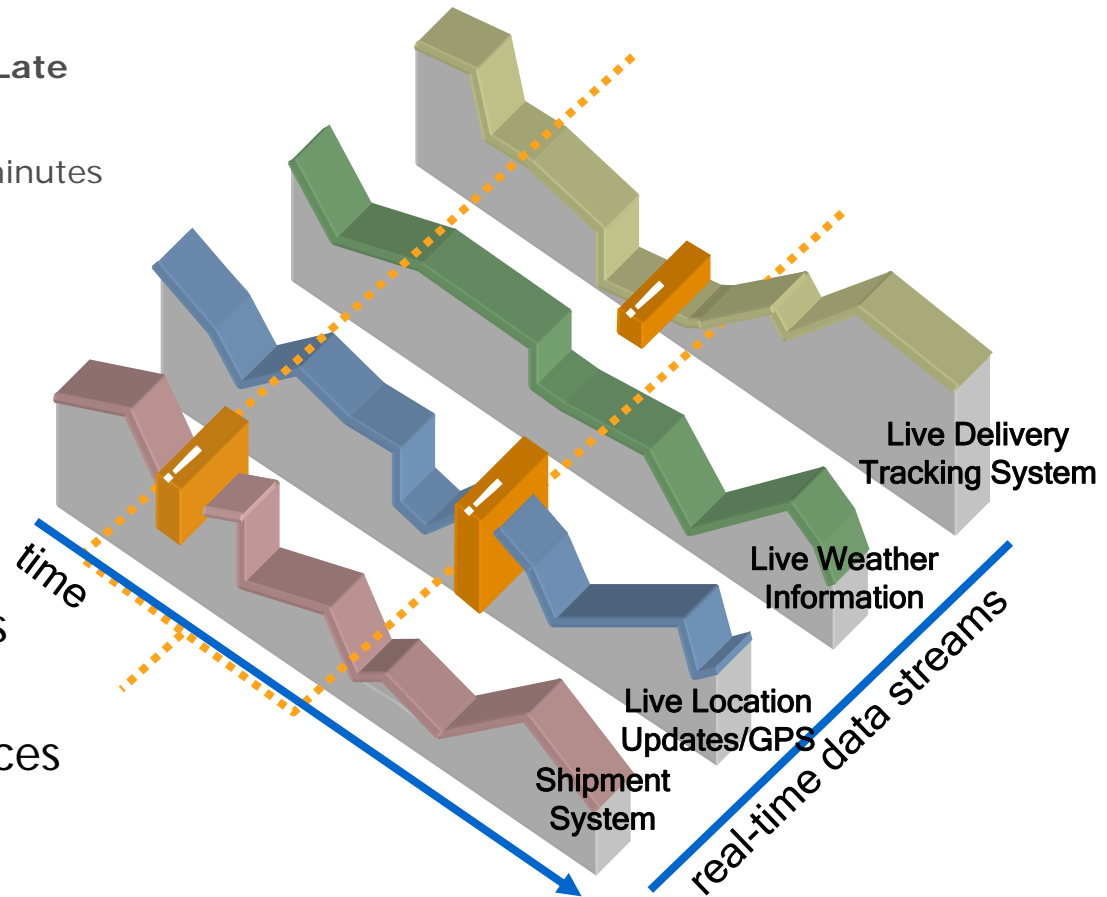
"Hot Load" is Potentially Late

### FOLLOWED-BY

"Hot Load" is **WITHIN** 40 minutes of destination

### THEN

Call and warn customers  
Allocate more capacity



- Real-time data streams
- Temporal sequencing
- Complex event sequences
- Real-time constraints
- Automated actions

# Maritime Logistics

## Monitoring Rule

IF

Vessel **ABC** is marked as "Important" with a critical cargo

FOLLOWED-BY

The vessel is potentially late by **60** minutes

FOLLOWED-BY

The vessel is **WITHIN 25** minutes of destination

THEN

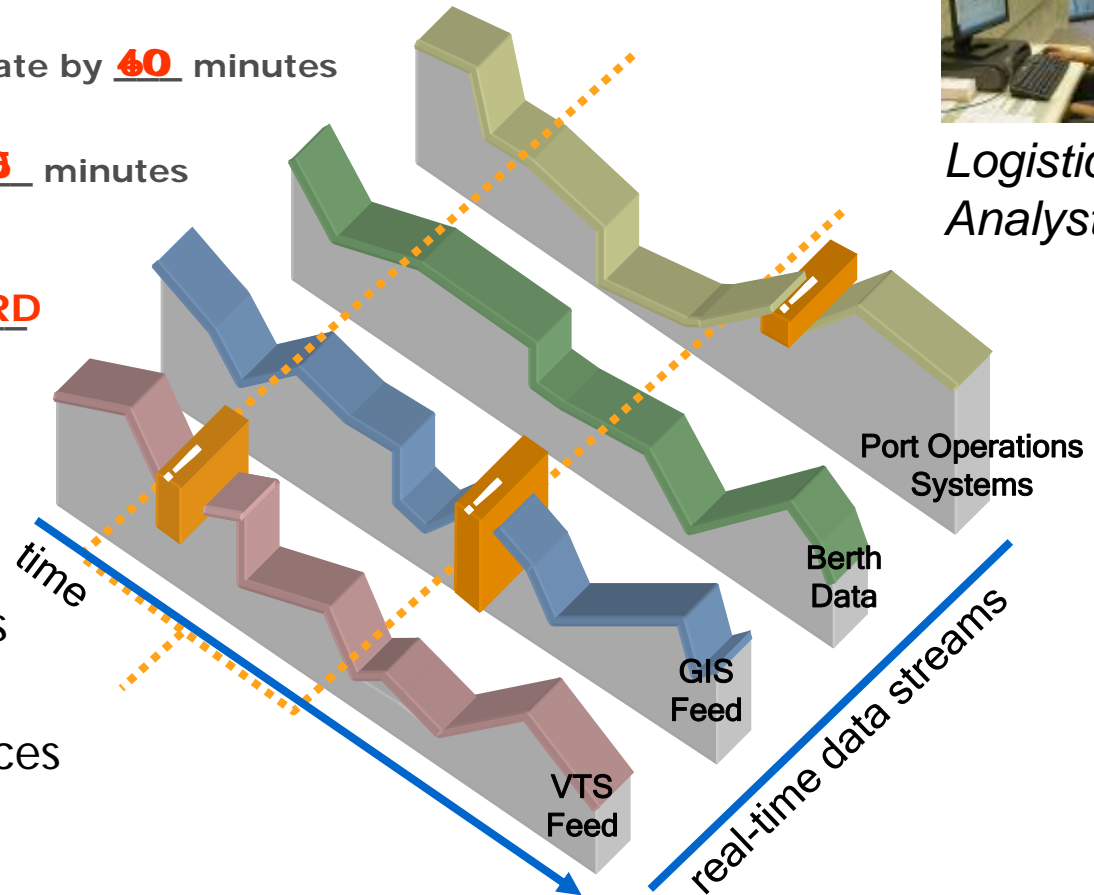
Send Alert to **DASHBOARD**



Operations  
Manager



Logistics  
Analyst



- Real-time data streams
- Temporal sequencing
- Complex event sequences
- Real-time constraints
- Automated actions

# Agenda

- Our **academic vision** for event processing in 1990s
- **Lessons learned** from real business applications
  - What did we omit from the academic vision?
  - How much of the vision has come to pass so far
- **Architecture** of a commercial event processing platform
- **Killer applications** of event processing
- **Predictions** for the event processing market



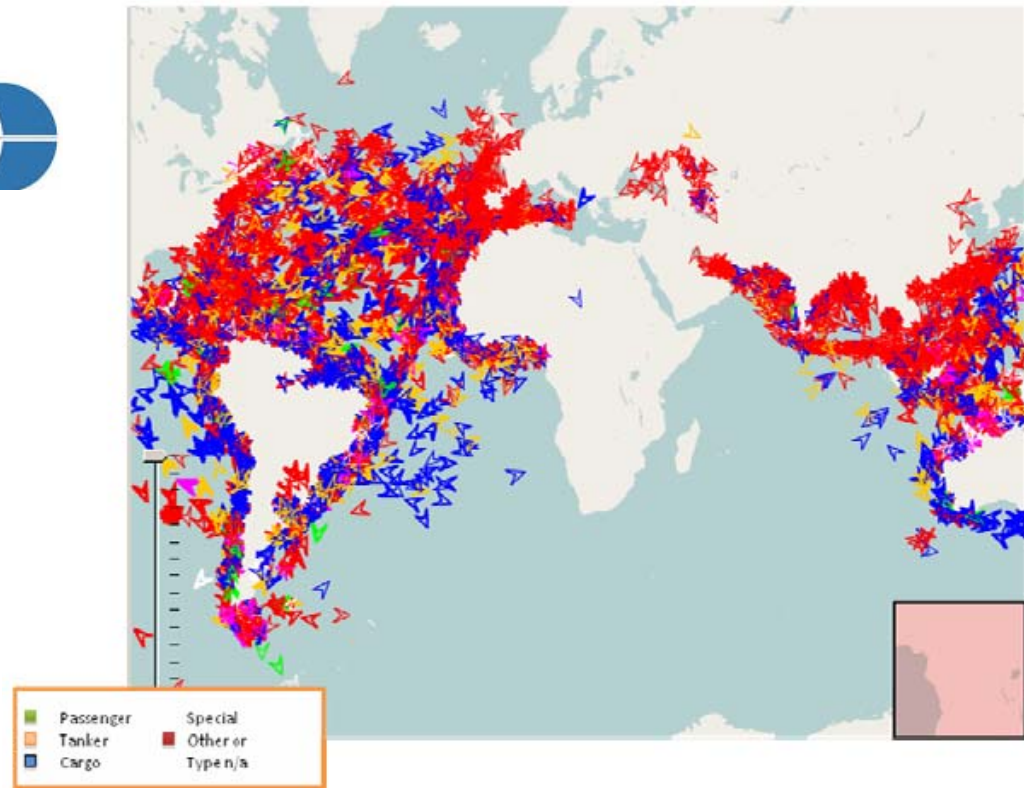
# The Next 5 Years [1]

- *Killer Applications* - tracking everything on the planet
  - with the ability to react instantly
  - E.g. every car, plane, truck, train, plane, bag, package against event-driven business rules

*Example: Dirkzwager*



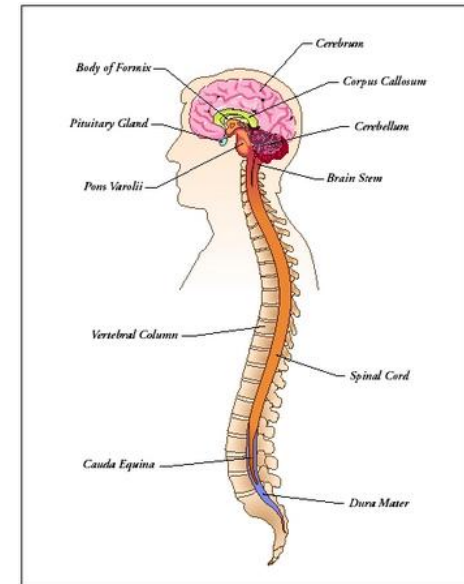
*How to optimize global ship movement and berth space for a lean and green approach?*





# The Next 5 Years [2]

- Event-driven architectures will achieve wider adoption
  - Federating services in the enterprise
  - Providing an agile enterprise “nervous system”
- Event processing *everywhere*
  - CEP rules in “the cloud”



# Predictions for the Market

- There will be no standalone event processing/CEP market
- Event processing/CEP will be a key part of other markets:
  - Part of a bigger “event-driven business process management” market
    - BPM, rules, CEP/business events, integration will merge
  - Embedded in vertical industry apps, e.g. trading platforms



# Conclusions

- Our research vision from the 1990s **anticipated many of today's business requirements**
  - Rapid application development, low latency response to events, event storage and retrieval
  
- However, a number of **real world lessons have been learned**
  - Start as an add-on to existing systems rather than a complete event-driven re-architecture
  - Graphical tools for business users are critical
  
- Within the next 5 years
  - More **truly event-driven architectures** will be implemented
  - Event processing will be a **critical part** of the business process management platforms and many business applications of the future