TPF Users Group - 2011

Making better business decisions with z/TPF

Instrumenting your z/TPF applications with business event publication APIs

Jason Keenaghan
Open Source Subcommittee

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0
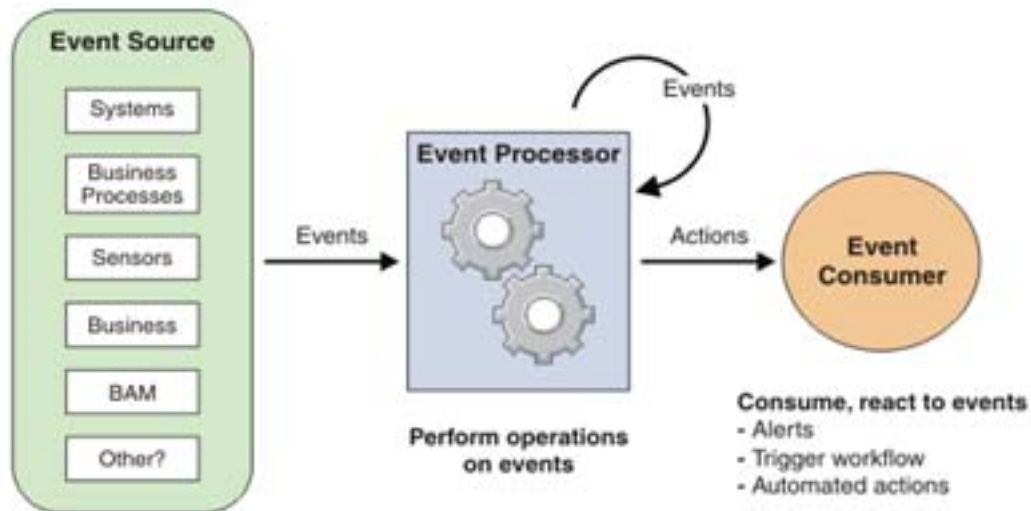
Agenda

- **What is business event processing?**

- **Defining business events for consumption**

- **Business events as a path to modernization and agility**

- **z/TPF business event processing**

  - Statement of direction and design preview

Event processing architecture

An event is anything that happenthat is significant to a system.

Event Source
- Systems
- Business Processes
- Sensors
- Business
- BAM
- Other?

Events →

Event Processor

Perform operations on events

Actions →

Events

Event Consumer

Consume, react to events
- Alerts
- Trigger workflow
- Automated actions

Event processing consists of:

- Event capture
- Data enrichment
- Formatting and emission
- Routing and further processing of emitted events
- Consumption of processed events

How event processing can help your business

- **You can use event processing to enhance governance for standards compliance, and to monitor business processing.**

- **Event processing enables detection of business situations, providing early and intelligent insight to assist you in making timely and effective business decisions.**

- **Event processing can help your business in these ways:**

  - Obtaining the right information at the right level of detail for the right person at the right time.

  - Facilitating quick observation of exceptional business behavior and notification to the appropriate people.

  - Diagnosing problems based on symptoms and resolving them.

  - Providing data for dashboard display of real-time business service availability.

So who will get these events? And what will they do?

- **WebSphere Business Events (WBE)**
  - Also now part of WebSphere Operational Decision Management (WODM) along with ILOG
  - Provides timely insight and response to emitted events
  - Allows business users to detect, evaluate, and respond effectively to the impact of business events
- **IBM WebSphere Business Monitor**
  - Provides business activity monitoring (BAM) capabilities
  - Measure business performance, monitor real-time and completed processes, detect and resolve problems in the execution of business processes, and report on business operations to enable cyclical improvements
- **Non-IBM vendor offerings**
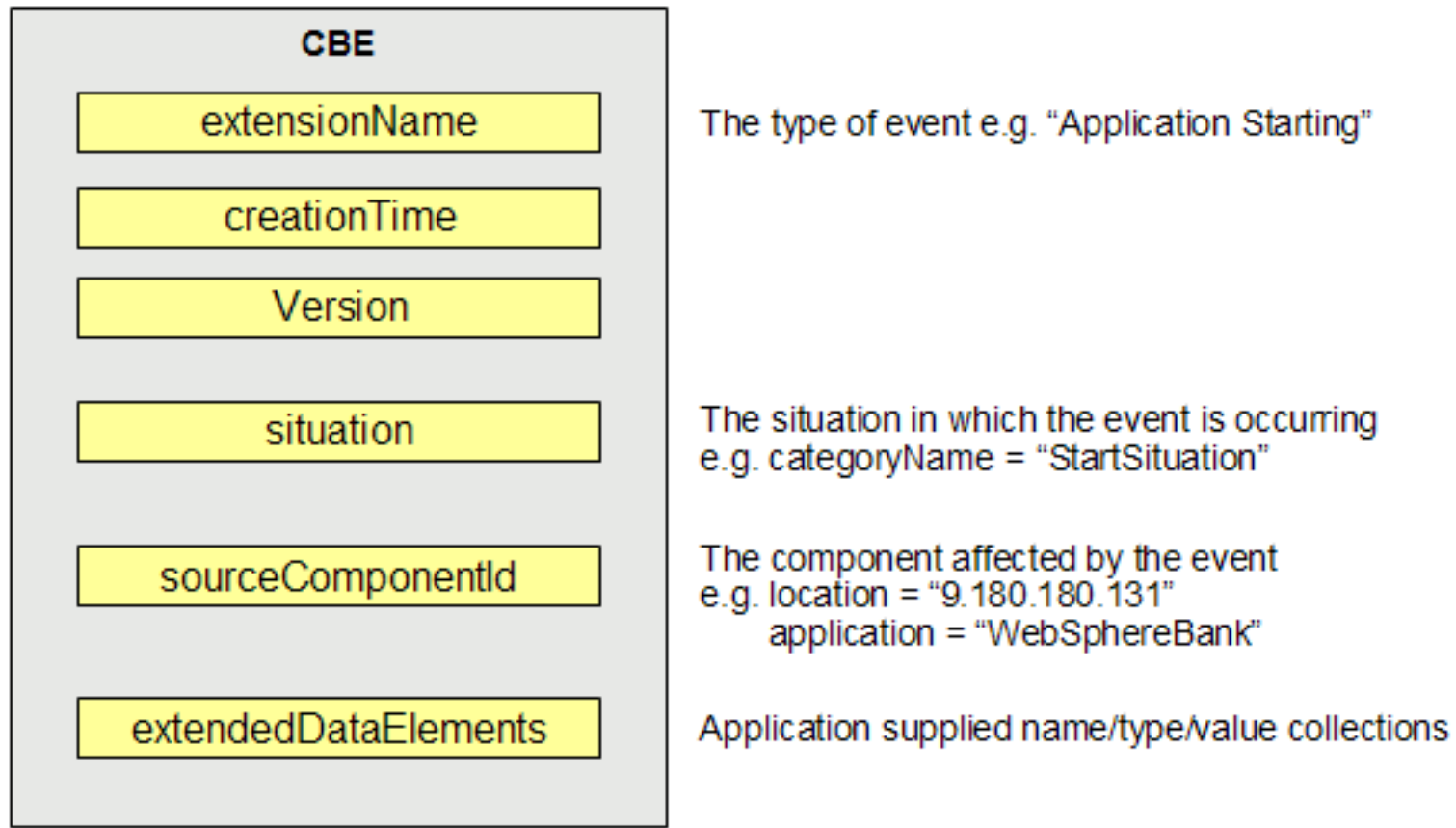- **Custom event consumers**

Many different products that are able to consume emitted events and take actions

Event message formats

- **Common Base Event (CBE)**

  - Standard event format
    - Proposed as new standard to OASIS
    - Included as part of Web Services Distributed Management (WSDM) standard
  - Supports both IT and business events

- **WebSphere Business Event (WBE)**

  - Event format unique to IBM WBE product
  - XML-based, like CBE, but with fewer elements

- **Common Base Event REST (CBER)**

  - Simplified XML format that can be sent using HTTP 1.1
  - Works with products like IBM WebSphere Business Monitor

Common Base Event (CBE) structure

Consists of a set of properties describing the event:

CBE	
extensionName	The type of event e.g. "Application Starting"
creationTime	
Version	
situation	The situation in which the event is occurring e.g. categoryName = "StartSituation"
sourceComponentId	The component affected by the event e.g. location = "9.180.180.131" application = "WebSphereBank"
extendedDataElements	Application supplied name/type/value collections

Defining an event

- **Determine what information is required to capture the event**

- **Decide which CBE properties are relevant and map them to the event data**

- **e.g. Loan Request Event:**

  - extensionName="Banking:LoanRequest"

  - situation:
    - categoryName="RequestSituation"

  - sourceComponentId:
    - application="BankingServices"

  - extendedDataElements:
    - customerName=<name of customer requesting the loan>
    - loanAmount=<amount of requested loan>

Sample Common Base Event (CBE) format

```
<CommonBaseEvent creationTime="2006-03-29T09:11:15.812Z" extensionName="Banking:LoanRequest"
                globalInstanceId="CE11DABF03F9D4A240C83B937DF9A21171" sequenceNumber="3" version="1.0.1">
```

```
    <extendedDataElements name="customerName" type="string"> <values>Gary Chapman</values> </extendedDataElements>
    <extendedDataElements name="loanAmount" type="string"> <values>64000</values> </extendedDataElements>
```

```
    <sourceComponentId application="BankingServices"
                component="WPS#Platform 6.0 [ND 6.0.2.3 cf30542.05] [WBI 6.0.1.0 gm0548.30] "
                componentIdType="ProductName"
                executionEnvironment="Windows XP[x86]#5.1"
                instanceId="widCell\widNode\server1"
                location="9.173.176.120"
                locationType="Hostname"
                processId="2508"
                subComponent="J2EE_Application"
                threadId="WebContainer : 1"
                componentType="http://www.ibm.com/namespaces/autonomic/WebSphereApplicationServer"/>
```

```
    <situation categoryName="RequestSituation">
            <situationType xsi:type="RequestSituation"  reasoningScope="EXTERNAL"
                    successDisposition="SUCCESSFUL" situationQualifier="START COMPLETED"/>
    </situation>
```

```
</CommonBaseEvent>
```

Disclaimer: Statement of direction

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.   Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

z/TPF support for business event processing

- **An API that will allow applications to signal that a business event has occurred**

- **Infrastructure that will support enrichment of event data and asynchronous emission of events, decoupled from application logic**

- **Extensible infrastructure that allows for custom event data formats and custom transport mechanisms**

- **Tooling to help create the necessary artifacts for application use and event management**

Event binding file: Defining business event to z/TPF

- **Determine the data that is important to emit as part of the event**

- **Specify data format of events using a tool that will create the data structures for application use (C/C++ and HLASM)**

  - Initially: Linux-based command line tool

  - Long term includes: integrate into TPF Toolkit for graphical user interface

- **Specify if events of this type must have their order preserved as they are emitted**
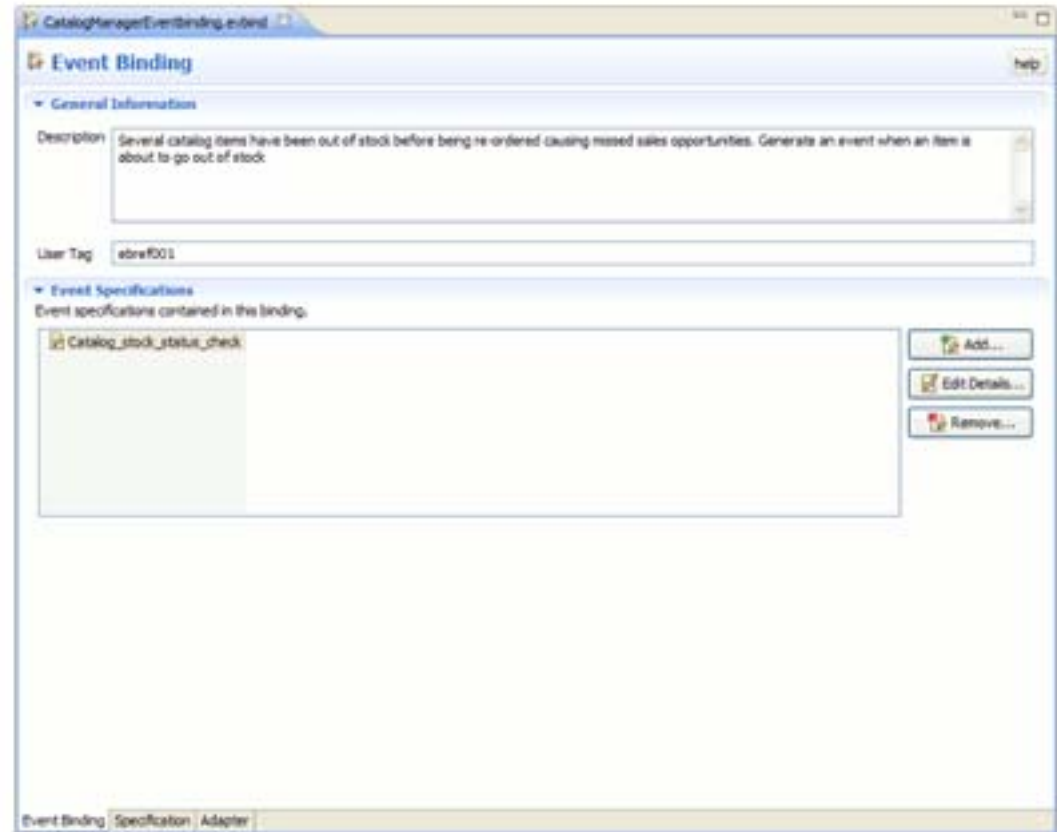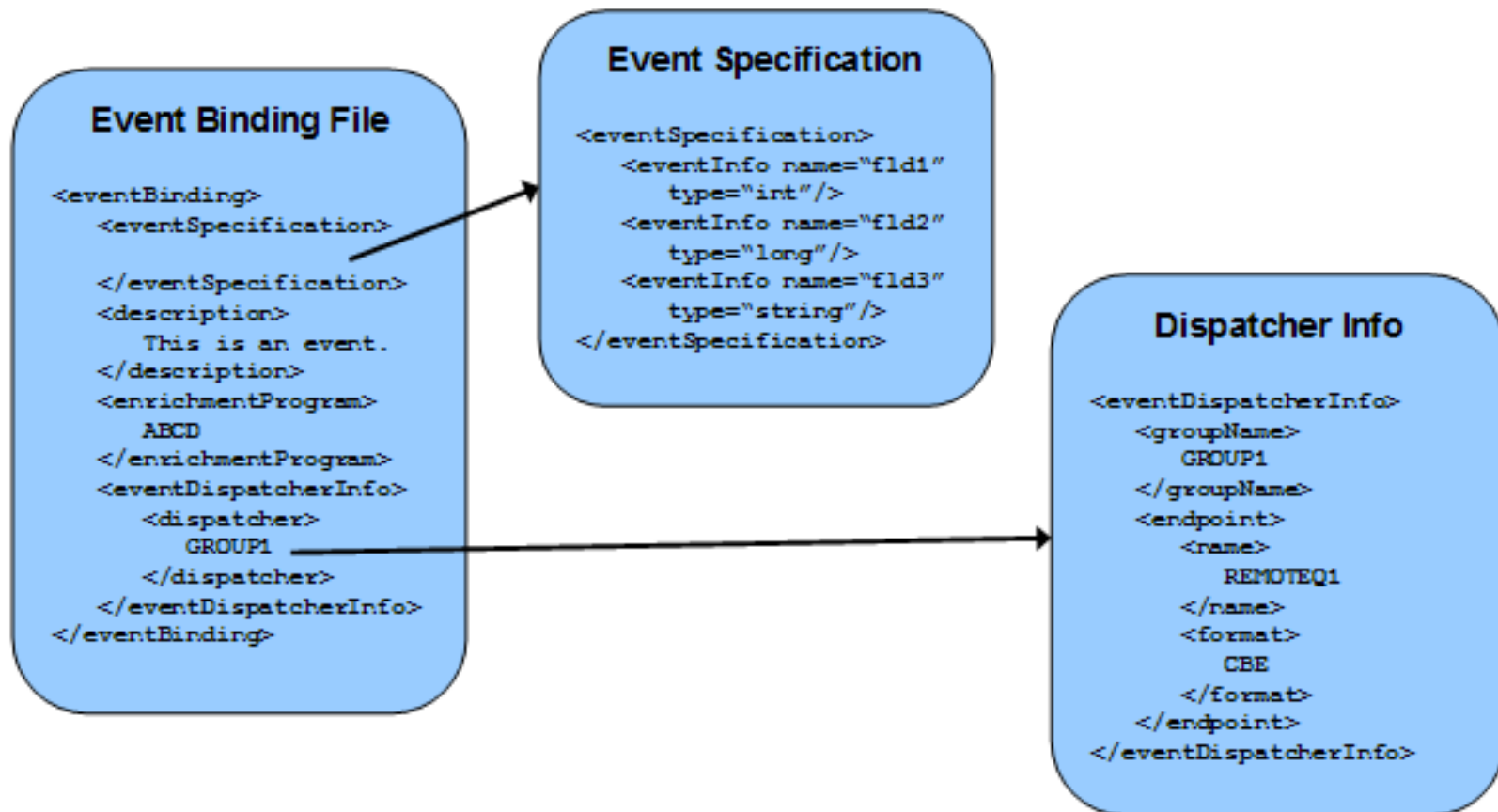

Note: This is not an actual screen shot from the TPF Toolkit but just an example of how an event binding editor might look

Event binding file: Defining event dispatch information

- **For any single event you must specify one or more endpoints (event consumers) to whom the event will be sent**

- **Multiple endpoints can be grouped together**
  - For each endpoint specify: communication information and expected format of event data
  - Example: If you have 15 events all going to the same 3 endpoints, specify endpoint information once and have all events reference it. If you need to add a new endpoint to the group, only have to update in one place and make it visible to all

- **Transport mechanisms to be supported:**
  - Initially: WebSphere MQ and custom
  - Long term includes: HTTP

- **Event data formats to be supported:**
  - Initially: CBE and custom
  - Long term includes: CBER and WBE

Logical representation of event binding files

Event Binding File

```
<eventBinding>
    <eventSpecification>

    </eventSpecification>
    <description>
        This is an event.
    </description>
    <enrichmentProgram>
        ABCD
    </enrichmentProgram>
    <eventDispatcherInfo>
        <dispatcher>
            GROUP1
        </dispatcher>
    </eventDispatcherInfo>
</eventBinding>
```

Event Specification

```
<eventSpecification>
    <eventInfo name="fld1"
        type="int"/>
    <eventInfo name="fld2"
        type="long"/>
    <eventInfo name="fld3"
        type="string"/>
</eventSpecification>
```

Dispatcher Info

```
<eventDispatcherInfo>
    <groupName>
        GROUP1
    </groupName>
    <endpoint>
        <name>
            REMOTEQ1
        </name>
        <format>
            CBE
        </format>
    </endpoint>
</eventDispatcherInfo>
```

For illustration purposes only: The different components are depicted here as separate XML files. The actual implementation is subject to change.

Enabling event bindings: Common deployment

- **Increasing popularity in using XML files as *function descriptors***
  - Describe capabilities and options of functionality in an easy to create, easy to read format: event binding file is one example
  - However, don't want to access the XML files themselves while processing transactions

- **Common deployment mechanism**
  - In general: Parse the XML function descriptor and place the relevant information into a structure in memory for fast access
  - Deployment and management of function descriptors can be done without regard to actual content of files:
    - Validate that file exists
    - Maintain status of which files are deployed
    - After an IPL, re-deploy previously active files
    - Provide a mechanism to locate the structure in memory given a key
   - Handle changes to the deployed file

Common deployment (continued)

- **Integrating common deployment with z/TPF loaders support for loading files into the file system (e.g., ZOLDR or ZTPLD)**

  - Function descriptors are usually tied to programs that are going to make use of the underlying functionality

  - ECBs running with one version of a program should have access to the corresponding function descriptor that represents the same version

- **Configuration file determines actions to take and specific programs to call based on file extensions**

  - Should files of this type be deployed automatically when loaded?

  - What program should be called to parse and validate the function descriptor?

Instrumenting applications to emit business events

- **Applications can be updated to signal the occurrence of a business event**

  - tpf_bev_signal() API

  - Specify event name, pointer to data and length, and intercept name

  - Data is specified using generated C structures and HLASM DSECTs

- **If business eventing support is not enabled for the system or for this event, no event will actually be generated**

z/TPF event processing components
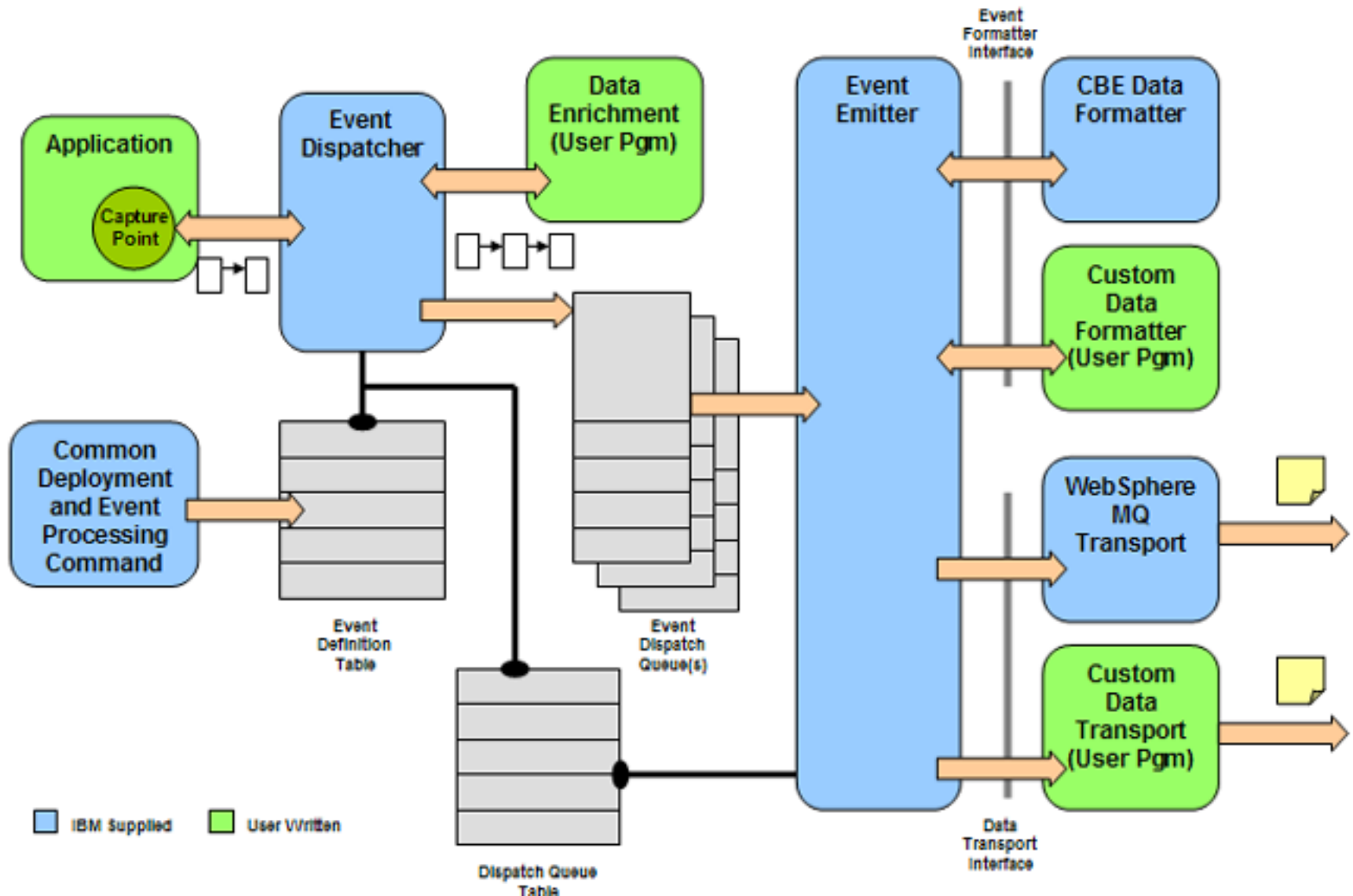
Summary

- **z/TPF applications can easily be instrumented to emit business events:**

    - Do not need to know the exact format of the data to be sent

    - Do not need to know where the event will be sent to

- **Easily extract and communicate critical business events in real time as part of transaction processing**

- **Enables you to drive business process management (BPM), apply complex event processing (CEP) technologies, and create business events dashboards:** *all based on information coming out of your z/TPF system*

Trademarks

- **IBM is a trademark of International Business Machines Corporation in the United States, other countries, or both.**

- **Apache is a trademark of The Apache Software Foundation.**

- **Other company, product, or service names may be trademarks or service marks of others.**

- **Notes**

- **Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.**

- **All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.**

- **This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.**

- **All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.**

- **Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.**

- **Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.**

- **This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.**