z/TPF V1.1

TPF Users Group - Fall 2012

Title: z/TPF Business Events and Common Deployment

Name: Michael Shershin
Venue: Main Tent

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

Agenda

- **PJ39759 – Websphere Business Events on z/TPF**
  - Includes changes in TPF Toolkit
    - TPF Toolkit V.next scheduled for release in 4Q2012
- **PJ39754 – Common Deployment**

What is a Business Event?

- **An event is an item of significance that has occurred in the system that is usually related to a business process.**
- **A Business Event consists of:**
  - Trigger
  - Data gathering and enrichment
  - Data formatting
  - Data transmission
- **Example: For international travel, send changes in a manifest to government agencies that handle immigration and customs.**

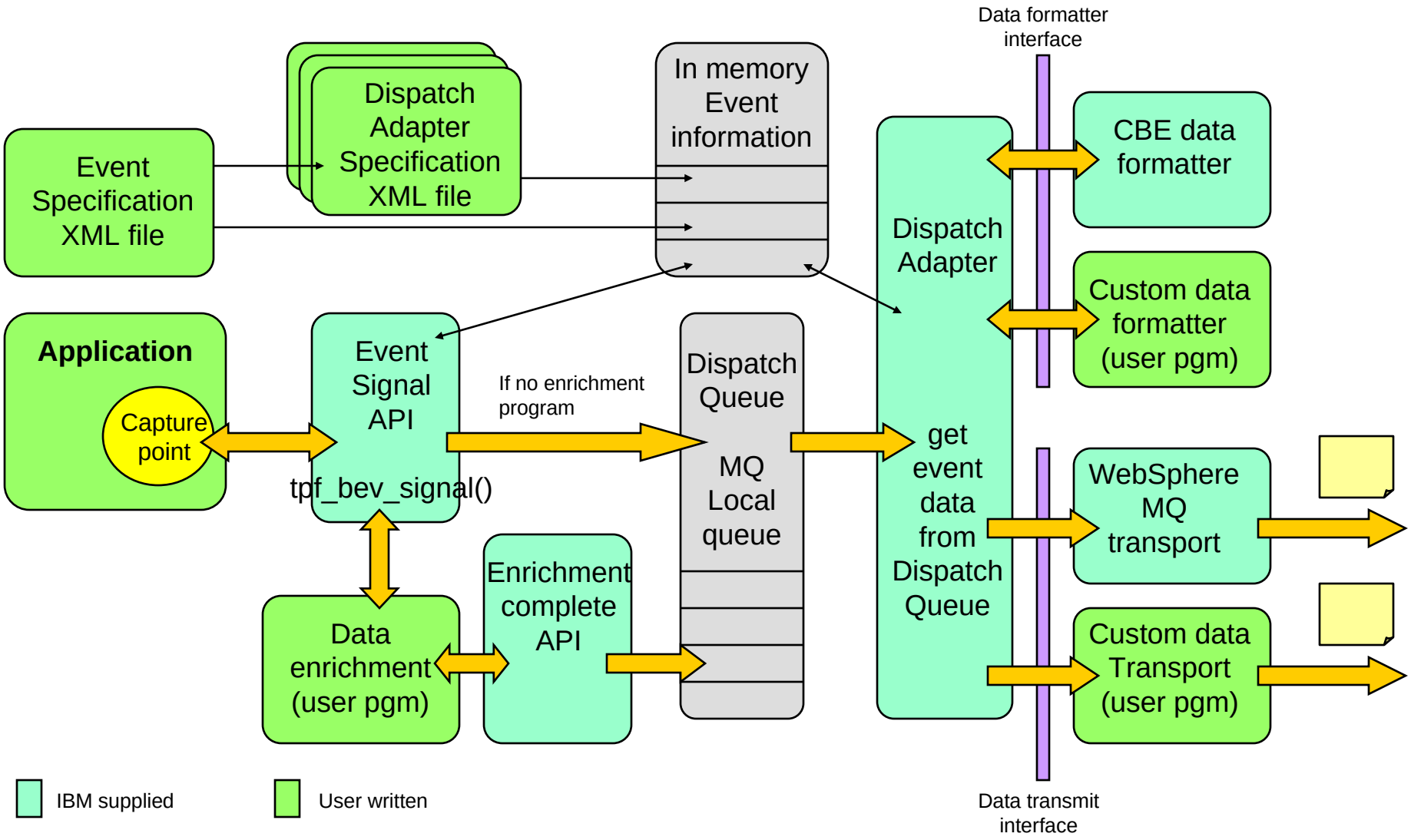
What is the value of a business event?

- **Send the right information at the right person at the right person.**
- **Facilitate real time observation of exceptional business behavior and notify the appropriate people.**
- **Provide data for dashboard display of real time business service availability.**

Business Events Glossary

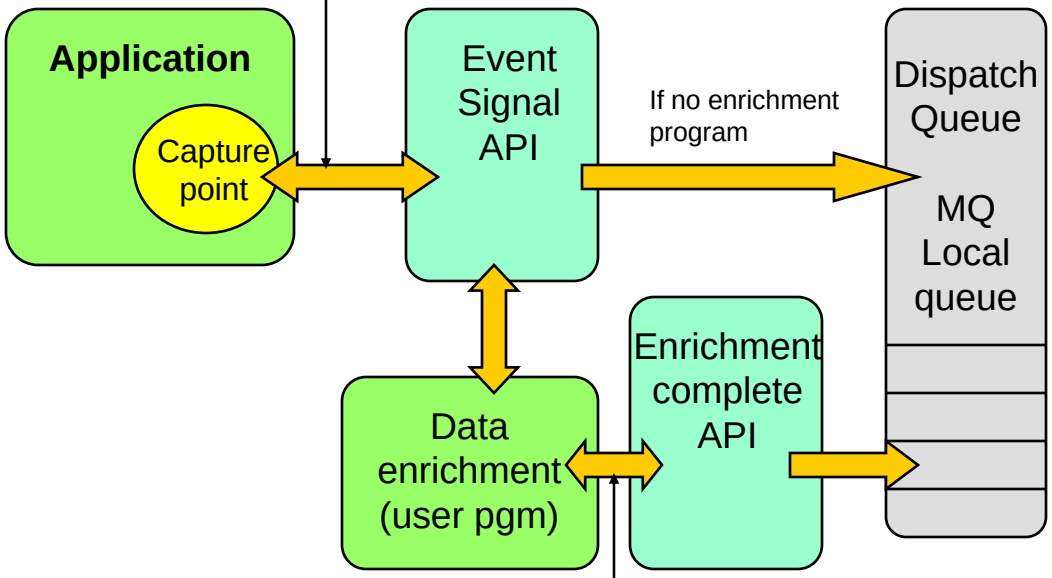
- **Business event consumer**
  - A location that receives data from a business event.
- **Business event dispatch adapter**
  - Transport mechanism that sends data to the business event consumer.
- **Business event specification**
  - An XML file that is the deployment descriptor for a single business event.
- **Business event dispatch adapter specification**
  - An XML file that is the deployment descriptor for a single business event dispatch adapter.
- **Business event dispatch queue**
  - A local MQ queue that allows asynchronous transmission of event data.

Business events architecture in z/TPF



Business Events APIs

```
tpf_bev_signal(char *eventName,
               int     dataLength,
               void *binaryData,
               char *interceptName)
```

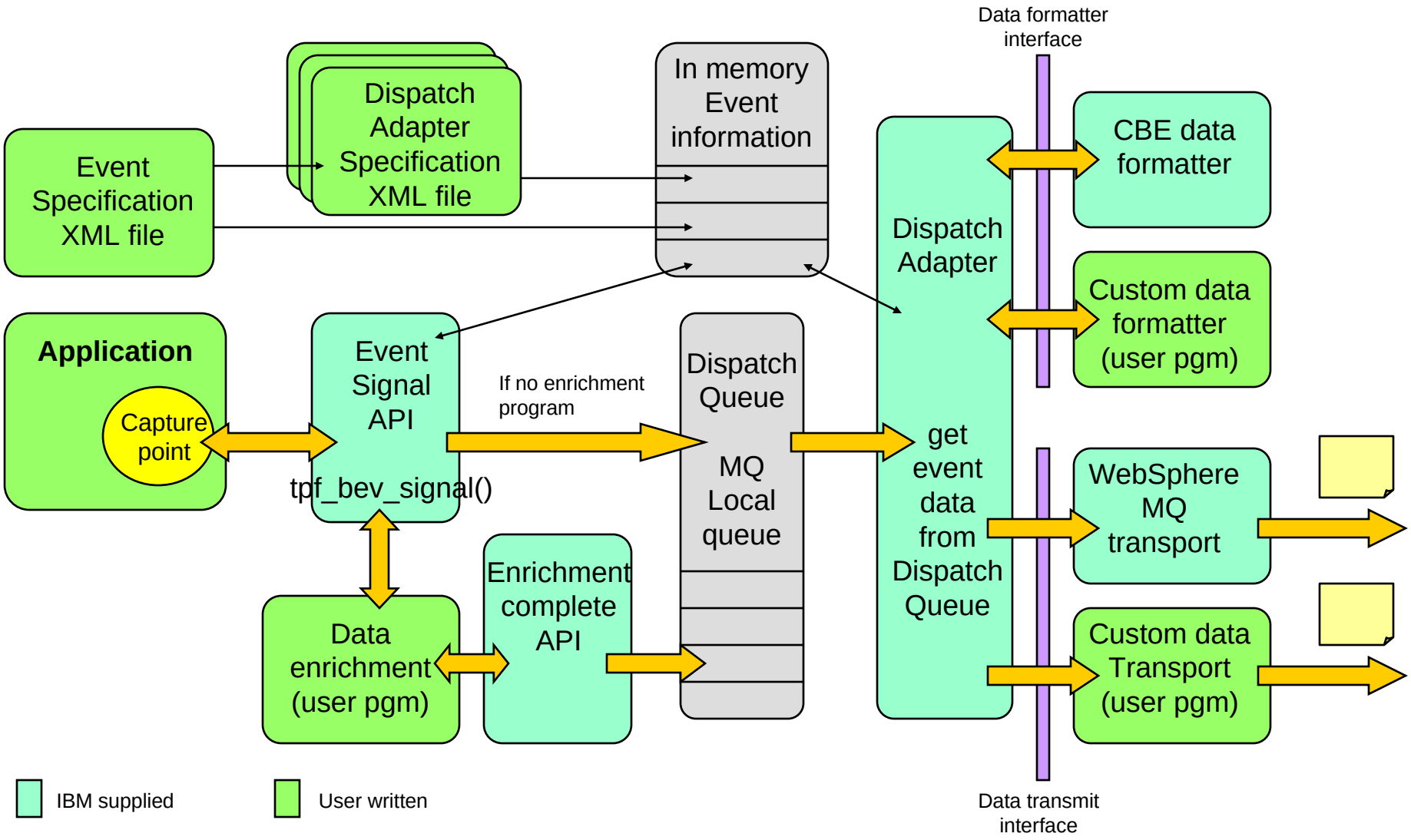


tpf_bev_signal_enrichment_complete()

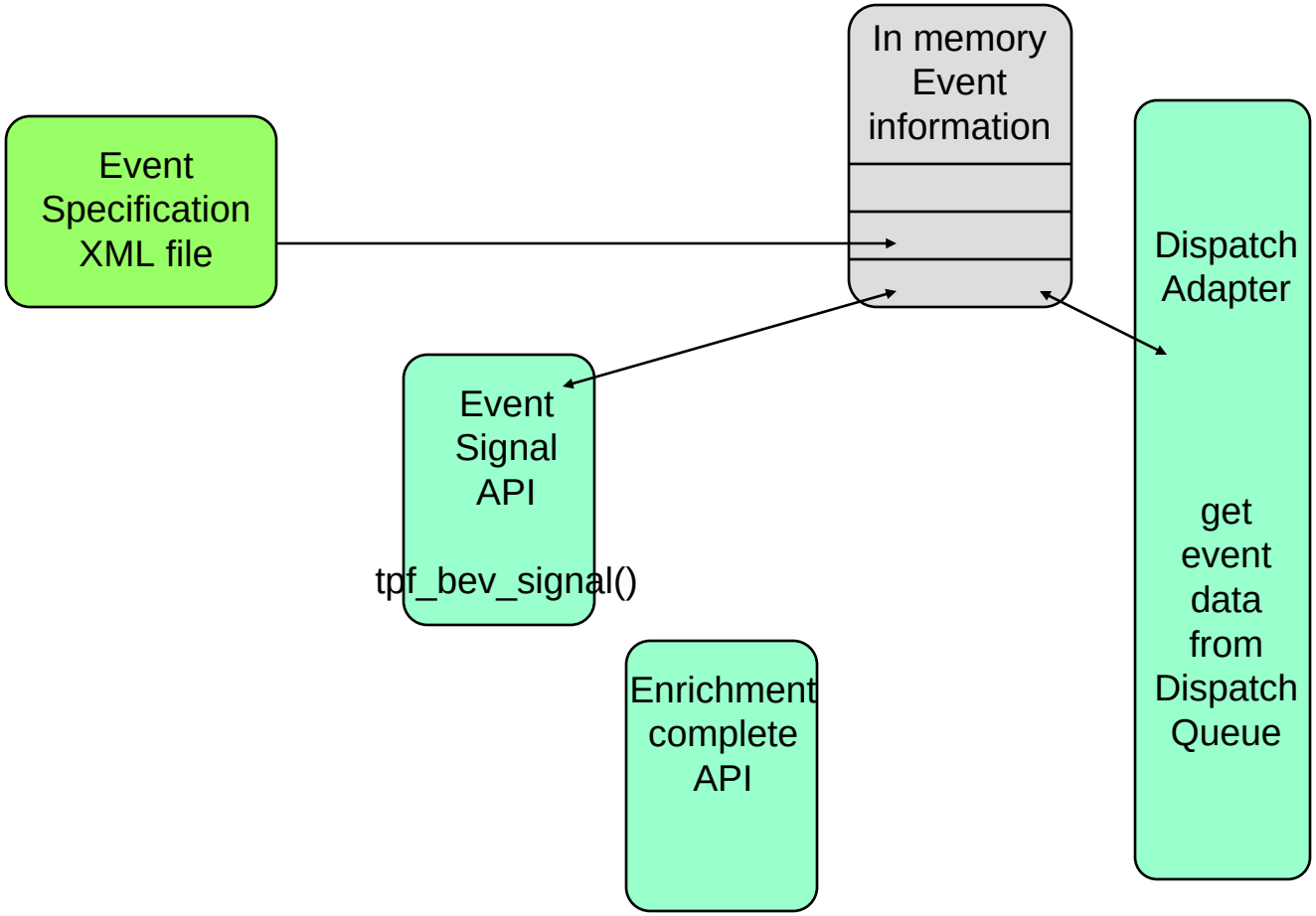
Business Events APIs

- **tpf_bev_signal()**
  - An event has happened.
  - Insert into your application process.
  - Event characteristics are in an event specification XML file
    - Event name is in event specification file
- **tpf_bev_signal_enrichment_complete()**
  - The enrichment program is complete. Send the event data to event consumers.
- **Event name provided on the APIs must match an event name in a deployed event specification file.**
- **Data can be passed on both APIs**
  - Data structure is defined in the event specification file.
  - Pass binary data such as a TPFDF LREC.
- **Intercept name is intended to identify a specific event call.**

Business events architecture in z/TPF



Event specification file



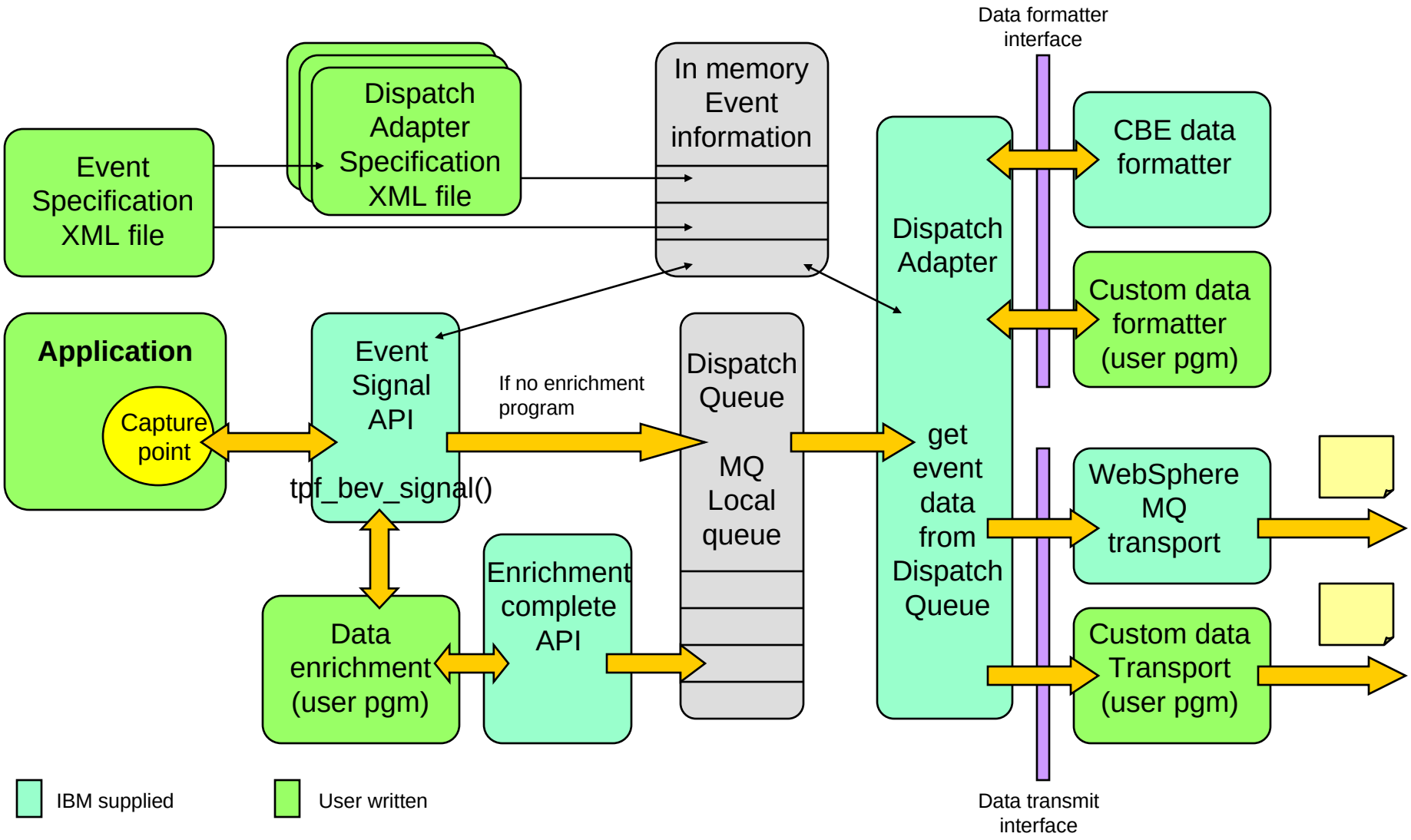
Event specification file

- **XML file that describes an individual business event.**
- **Managed by common deployment.**
  - Files loaded to TPF using E-type loader or image loader.
  - Files parsed once and results put into an in memory structure.
  - Must deploy an event specification file if this is the initial load of this file.
    - ZMDES DEPLOY FILE-*name*
- **Use TPF toolkit wizard to create Event Specification file.**

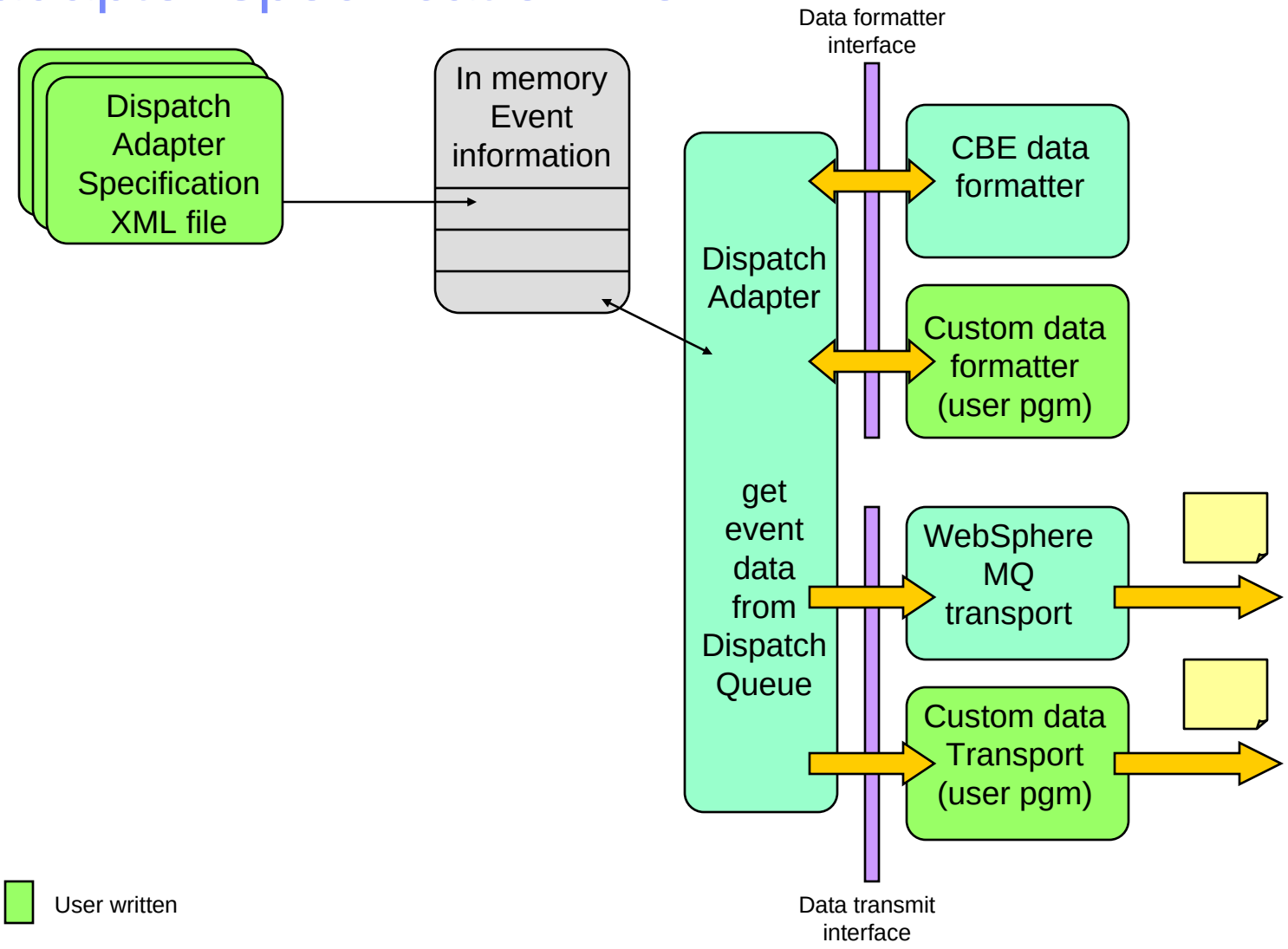
Event specification file contents

- **Event name**
- **Enrichment program (4 character name) (optional)**
  - If not specified, data provided on the tpf_bev_signal() will be added to the dispatch queue.
- **Dispatch queue name (optional)**
  - If not specified, the system default dispatch queue will be used.
- **Message properties**
  - Persistence
  - Priority
  - Expiry Time
- **Definition of data that is being captured.**
  - Represents the binary data that is being captured.
    - DSECT / C structure
  - Conforms to TPF Data Model (TDM).
- **List of dispatch adapters to use.**

Business events architecture in z/TPF



Dispatch adapter specification file



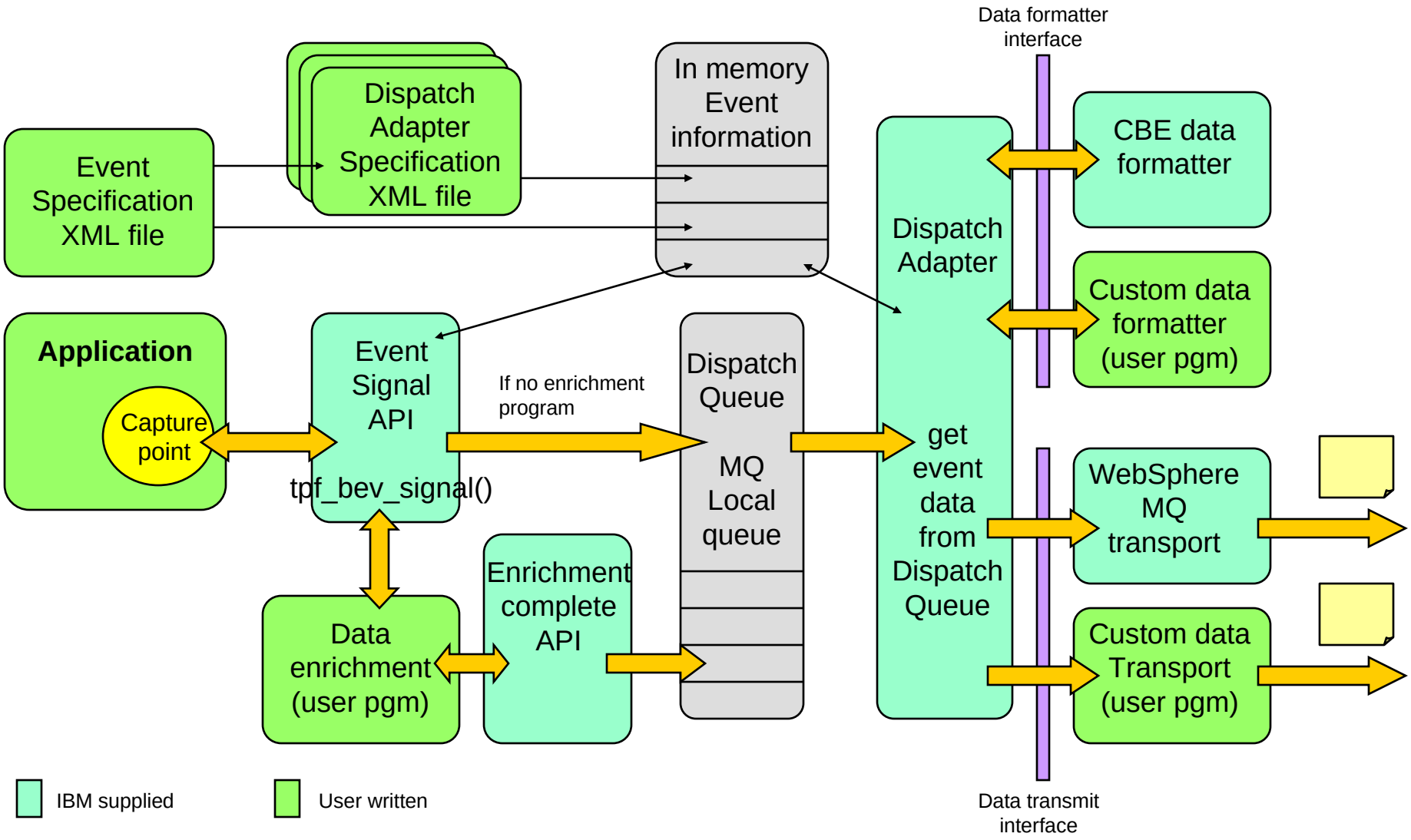
Dispatch adapter specification file

- **XML file that describes how to format data and where to send data.**
- **Managed by common deployment.**
  - Files loaded to TPF using E-type loader or image loader.
  - Files parsed once and results put into an in memory structure.
  - Dispatch adapter specification files are automatically deployed on the initial load of a specific file.
- **Dispatch adapter name is specified in the event specification file.**
- **Dispatch adapter is used after data is pulled from the dispatch queue.**
- **Use TPF toolkit wizard to create Dispatch Adapter Specification file.**

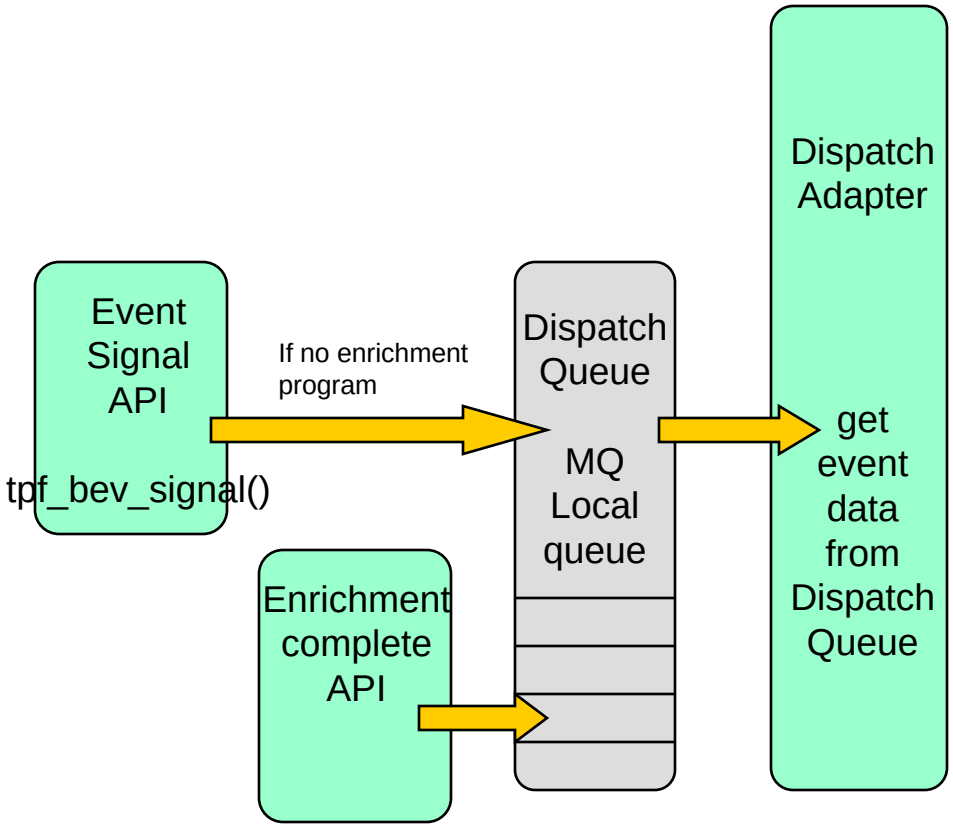
Dispatch adapter specification file contents

- **Dispatch adapter name**
- **Format**
  - Common base event (CBE) V1.0.1
  - Custom
    - 4 character program name
- **Adapter type**
  - WebSphere MQ
    - Queue name
  - Custom
    - 4 character program name
    - User unique data (i.e. IP address)

Business events architecture in z/TPF



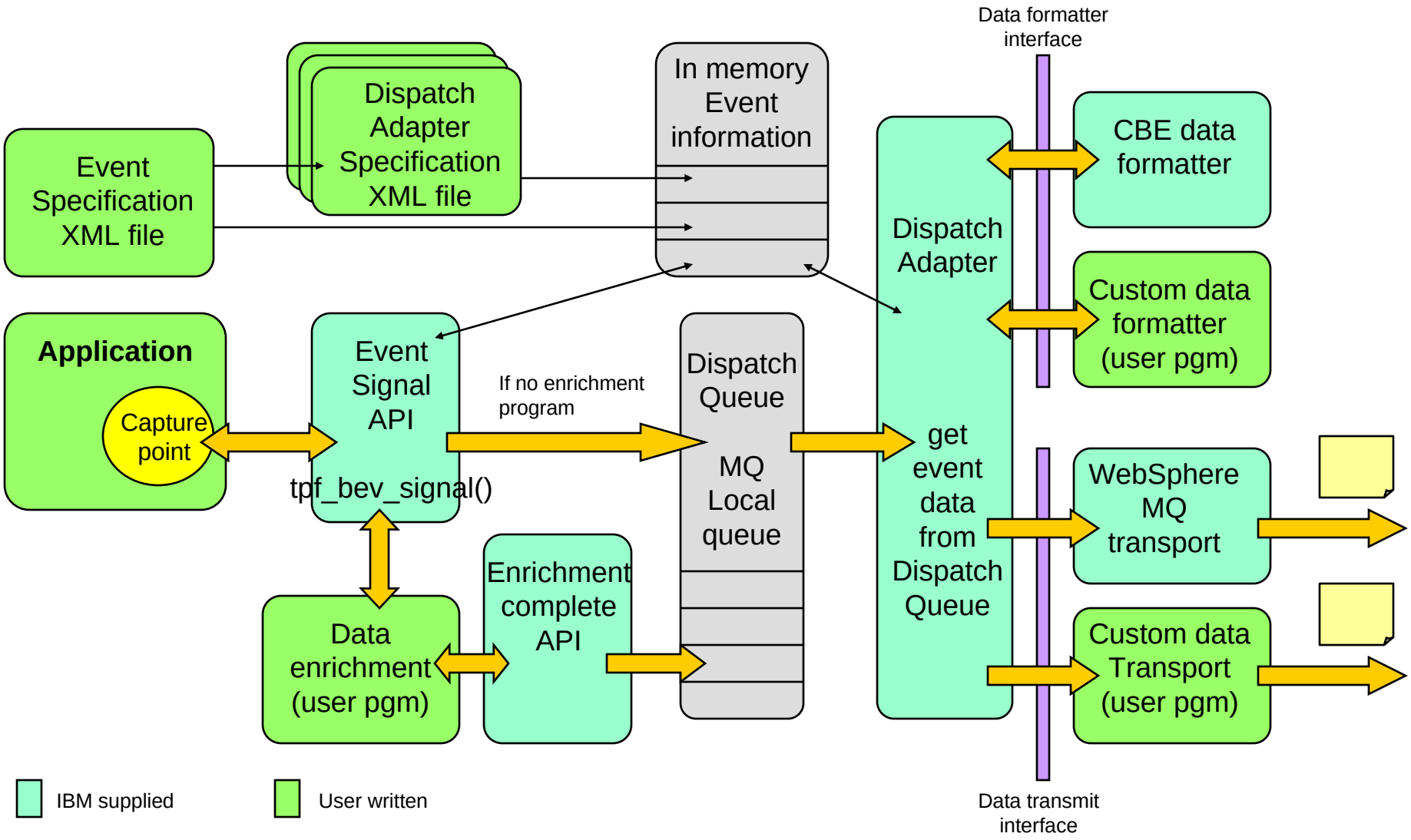
Dispatch Queue



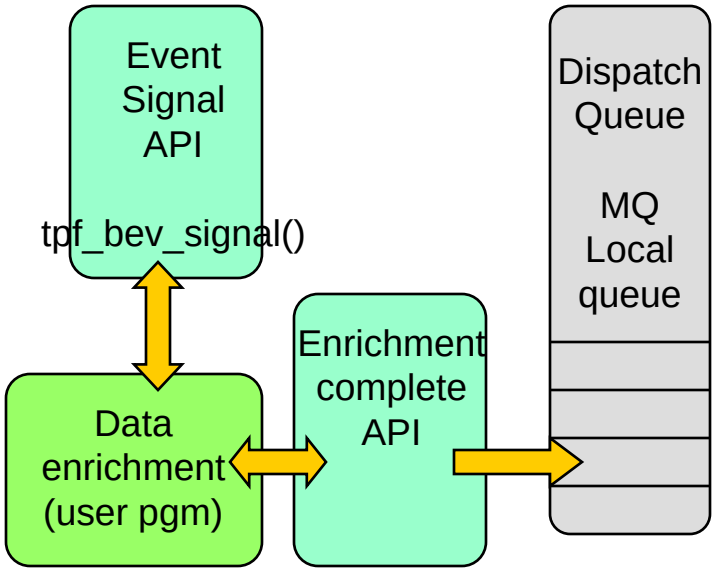
Dispatch Queue

- **Purpose is to disassociate the business event call from the transmission to the business event consumers.**
- **MQ local queue**
  - Allows the event processing to run in a commit scope when tpf_bev_signal() is called.
- **System default queue**
  - Used if dispatch queue is not specified in event specification file.
  - Does not guarantee order .
  - Multiple ECBs used to pull from dispatch queue.
- **User specified event dispatch queue.**
  - Specified in event specification file.
  - Guarantees order.
  - Only one ECB used to pull from each user specified dispatch queue.

Business events architecture in z/TPF



Enrichment program



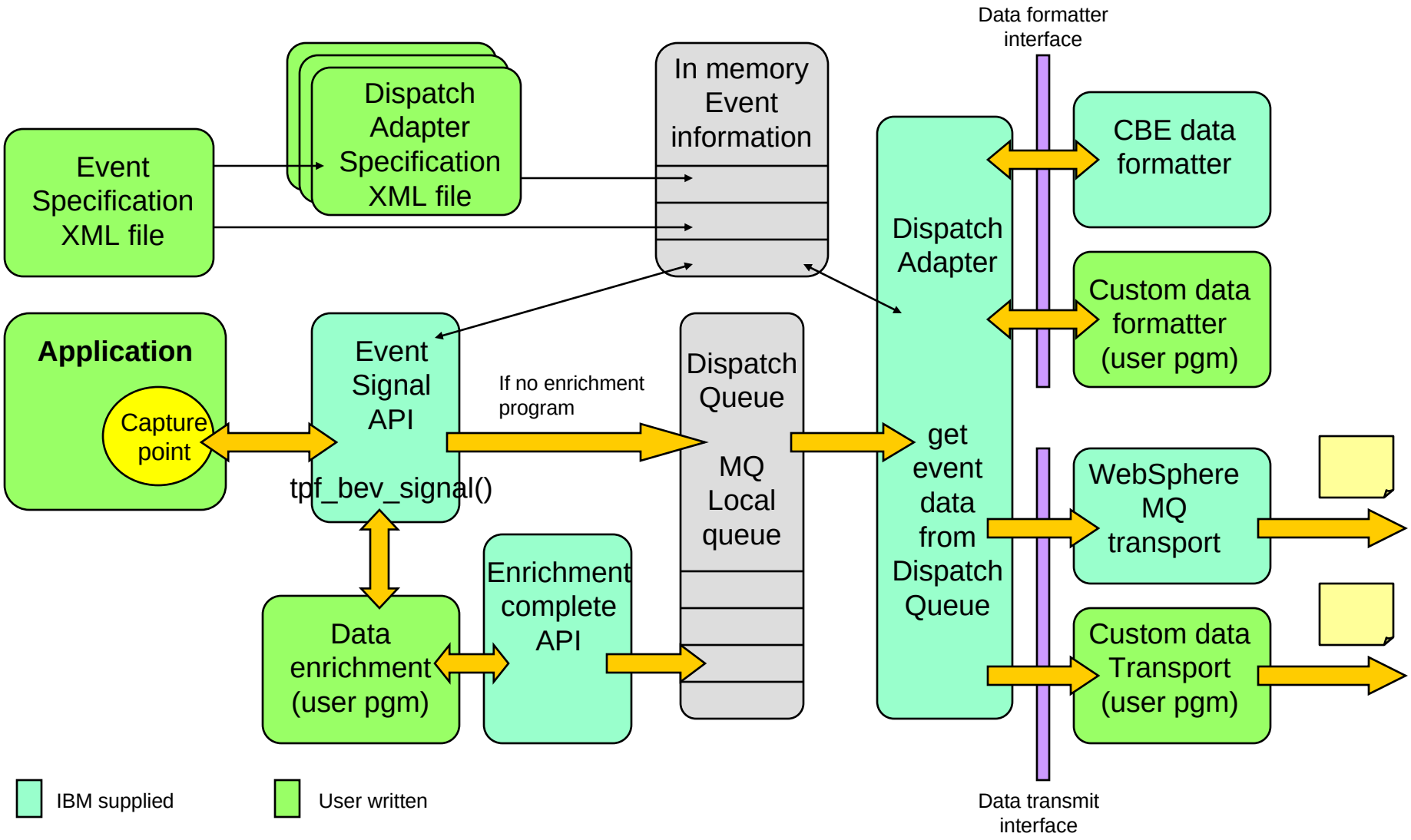
IBM supplied

User written

Enrichment program

- **Optional, user written program**
- **Used to gather additional information for the event that you want to pass to the event consumer.**
- **Entered by same ECB that called the tpf_bev_signal() API.**
- **Ability to modify message properties**
  - Expiry time, priority, persistence
- **Ability to identify which dispatch adapters to use**
- **Ability to pass information specify additional information to add to the CBE XML document**
- **When enrichment is complete, to continue event processing call API:**
  - tpf_bev_signal_enrichment_complete()

Business events architecture in z/TPF



Business Events Controls

- **Enable / disable business events by processor**
  - ZBEVF ENABLE
- **Maximum number of events that one ECB can signal in 1 second.**
  - ZBEVF SET MAXEVENTS-*value*
- **Maximum number of ECBs that can process events from the system default dispatch queue.**
  - ZBEVF SET MAXECBS-*value*
- **Maximum number of errors that can occur in 1 minute for a dispatch adapter before the dispatch adapter is automatically undeployed.**
  - ZBEVF SET MAXERRORS-*value*
  - User exit in UBEV allow users to decide how to handle errors.
- **System default dispatch queue depth warning value.**
  - ZBEVF SET QDEPTHWARN-*value*
- **Display control values**
  - ZBEVF DISPLAY SETTINGS
- **Use Format 2 Global: IBEVCNTL**

Various Business Events Displays

- **Display list of business events that are in memory**
  - ZBEVF DISPLAY EVENT
- **Display information about a specific business event**
  - ZBEVF DISPLAY EVENT *event_name*
- **Display list of dispatch adapters that are in memory**
  - ZBEVF DISPLAY DISPATCH ADAPTER
- **Display information about a specific dispatch adapter**
  - ZBEVF DISPLAY DISPATCH ADAPTER *adapter_name*
- **Display errors for both business events and dispatch adapters**
  - ZBEVF DISPLAY ERRORS
- **Display number of events over a specified period of time**
  - ZBEVF USAGE TIME-*value*

Business Events monitor

- **Verify that ECBs that are pulling items from the dispatch queue are still active.**
- **Warning messages**
  - If an error happened in the last minute, send a message indicating what error happened and the number of errors.
  - Once a minute if the number of messages on each dispatch queue is larger than QDEPTHWARN value, send a message to the console with the number of messages on the dispatch queue.

Common Deployment

Common deployment glossary

- **Common deployment**
  - A mechanism to make deployment descriptors available for use.
- **Common deployment configuration file**
  - A file that contains information about each piece of functionality that uses common deployment.
- **Deployment descriptor**
  - An XML file that is used to describe capabilities and options for a specific piece of functionality.

What is common deployment?

- **One use of XML files is to describe capabilities and options of functionality.**
  - Deployment descriptor
  - Requires XML file be deployed
  - Example:
    - Business event specification file
    - Business event dispatch adapter specification file
- **Deployed means:**
  - The XML file is parsed once
  - Results are put into a structure in memory
  - Mechanism to locate the in memory structure
- **Common deployment managed files are loaded to TPF using TPF loader.**
  - E-type loader (ZOLDR) and image loader (ZTPLD).
- **XML files are parsed:**
  - In restart
  - When a loadset that contains a deployment descriptor is activated (ZOLDR ACT)

What is common deployment?

- **Several common tasks that deployment for all deployment descriptors must do.**
  - Validate that the file exists
  - Maintain status of which files are deployed
  - After an IPL re-deploy the file
  - Provide a mechanism to find the structure in memory
  - Handle changes to the file (i.e. handle a file in a loadset that was activated or deactivated).

Who uses common deployment?

- **Business events**
  - Event specification file
  - Dispatch adapter specification file
- **WODM**
  - Endpoint group descriptor
  - TPF data model descriptor
  - WODM RuleApp descriptor

Common deployment rules

- **Deployment descriptors must be loaded to the system using the TPF loader (OLDR or TLDR).**
  - IBM delivered files are specified in control file
    - base/cntl/tpf.loadfile
- **A unique file extension must be used on the deployment descriptor.**
  - Used to identify the function (or application)
  - Comprised of "*.function_unique_name*.xml"
    - Business event specification is: ".evspec.xml"
    - Business event dispatch adapter specification is: ".evda.xml"
- **On z/TPF system deployment descriptors must be in directory**
  - /sys/tpf_pbfiles/tpf-fdes/.

Common deployment conventions

- **File extension is part of the name of the schema.**
  - The name of the schema must be tpf-*file_extension*.xsd.
- **On linux IBM delivered schema are in relative directory**
  - base/tpf-fdes/schema/.
- **On z/TPF the schema are in directory**
  - /sys/tpf_pbfiles/tpf-fdes/schema/.

How to control a deployment descriptor

- **Deploy**
  - ZMDES DEPLOY FILE-file_name
  - ZMDES DEPLOY FILE-file_name IPROC-x
- **Undeploy**
  - ZMDES UNDEPLOY FILE-file_name
  - ZMDES UNDEPLOY FILE-file_name IPROC-x
- **Display status**
  - ZMDES DISPLAY

Common deployment file system usage

- **Files loaded to the system using ZOLDR or ZTPLD**
  - Configuration File
    - /sys/tpf_pbfiles/tpf-fdes/fdes-config.csv
  - Deployment descriptors must be in directory:
    - /sys/tpf_pbfiles/tpf-fdes/
  - By convention schemas are put in directory:
    - /sys/tpf_pbfiles/tpf-fdes/schema/
- **Non-loaded file:**
  - Status file
    - /etc/tpf-fdes/.status_X where X is the CPUID of the processor

Common deployment configuration file

- **Identify each function that uses common deployment**
- **On linux, relative directory is:**
  - base/tpf-fdes/fdes-config.csv
- **On z/TPF, absolute directory is:**
  - /sys/tpf_pbfiles/tpf-fdes/fdes-config.csv
- **Contents**
  - Column 1: File extension (e.g. evspec.xml)
  - Column 2: 4-character program name that does function unique processing
  - Column 3: Auto-deploy indicator (e.g. Yes, means that a function descriptor will automatically be deployed when it is initially loaded_
  - Column 4: Permanently-deploy indicator (e.g. Yes means that a function descriptor can never be undeployed)

Function unique processing program

- **Common deployment cannot do some work for a deployment descriptor.**
- **Function unique processing program does this work**
- **Three actions:**
  - Build the in memory structure
    - Called by restart and ZOLDR ACT
    - Do XML parsing
    - Obtain system heap to keep results from parsing
    - System heap must have a standard header
  - Processing on cycle to norm
  - Clean up processing
    - Called by EAT cleanup
    - Return system heap

Common Deployment locate an in-memory structure

- **Use C function: tpf_fdes_find()**
- **Locates in-memory structure based on:**
  - Key (example: event name)
  - Type (example: event specification - BEVEVESTYPE)
  - Version
  - Activation number
- **Key, type, and version are supplied by the function unique processing program on return from the build option.**
  - Located in the standard header of the in-memory structure

The End

Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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.