



The future runs on System z

CICS Transaction Server for z/OS V4.1 and Events



Session Agenda

- **Introduction to Event Processing**
- **CICS and Events**
 - Event specification
 - Capture specification
 - Event Binding
- **Event Binding Editor**
- **CICS Event Adaptors**
- **Summary**

What is an event?

- **An event is**

- *Anything that happens (or is contemplated as happening)*
- An event has a name and usually some data (its payload)
- Produced and responded to asynchronously

- **Simple event**

- A single event, meaningful in itself
 - Order placement, bank account update, stock trade

- **Complex event processing**

- Detect and respond to patterns of events
 - Three orders from customer A in 2 days
 - ATM withdrawal after PIN change update
 - Unusual stock trading pattern

- **Business Event Processing**

- Detect and respond to events that have a business impact
- Extend event processing capabilities to business users
 - IBM WebSphere Business Events (WBE)

Event Processing – Why now?

- **Event processing is not new**
 - Systems management and monitoring
 - Pub/Sub messaging systems
- **The *business* value of event processing is decreased latency in**
 - Obtaining insights
 - Making decisions based on those insights
 - Executing the decisions
- **Multiple business factors have accelerated event processing requirements:**
 - Compliance with regulations
 - Demand for cost reduction leading to more automation
 - Technology developments such as RFID
 - Desire for greater awareness of business behavior

The gap between LOB and IT limits

Business users know which event patterns are relevant and what actions are required, but aren't equipped to implement themselves



IT people *may* have event-based tools and technologies but can't respond quickly enough to changing business requirements



Business Event Processing Bridges the Gap

Puts power in the hands of the business user

- No coding required for defining business event patterns
- Tasks performed via tooling



Provides unique convergence of power, flexibility, and ease of use

- IT configures endpoints and payloads
- Leverages SOA infrastructure



CICS TS V4.1 is aimed at helping users to

Compete for new opportunity by gaining insight into business processes and responding by modifying key business applications quickly and with confidence

– Business Flexibility and Innovation

Comply with corporate, industry and government policies to manage business risk of critical business applications

– Governance and Compliance

Event Processing

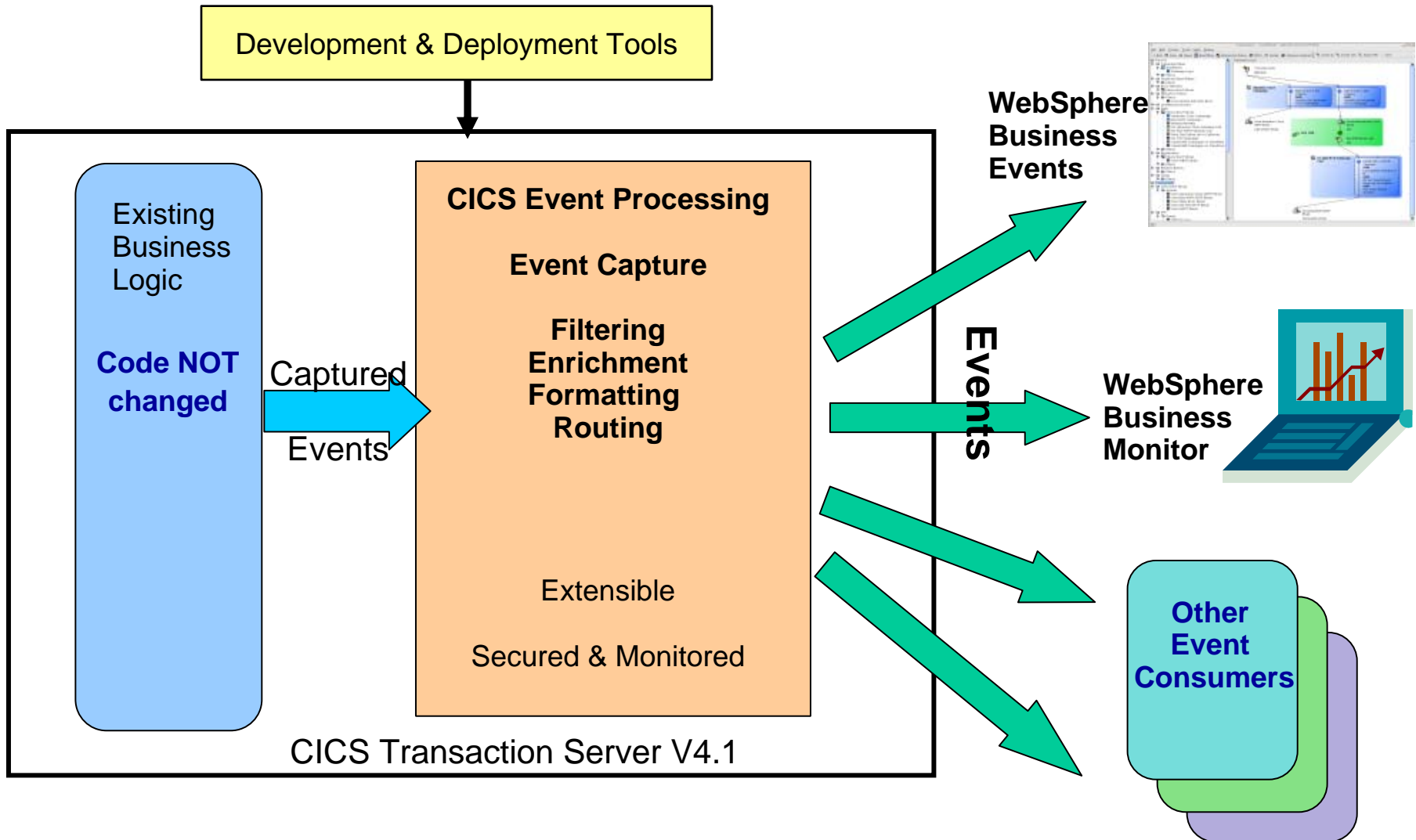
Control costs by simplifying IT infrastructure and improving development and operations productivity through easier-to-use interfaces and functions

– IT Simplification

CICS and Business Events

- **Event processing addresses the need for agility**
 - Modern businesses must react quickly to circumstances
 - Decision makers need reliable, timely information
- **CICS systems run an enormous amount of existing business logic**
- **With an Event-based approach**
 - Potential to gain insight into the processing in CICS
 - Introduce additional extensions to applications
- **CICS TS V4.1 allows you to emit business events from existing applications**
 - Supporting changing corporate policies
 - Without the need to change the applications
 - In a dynamic, de-coupled fashion
 - And driving your choice of destination
 - WebSphere Business Monitor
 - WebSphere Business Events
 - CICS application
 - MQSeries, ...

An Overview of CICS Event Processing

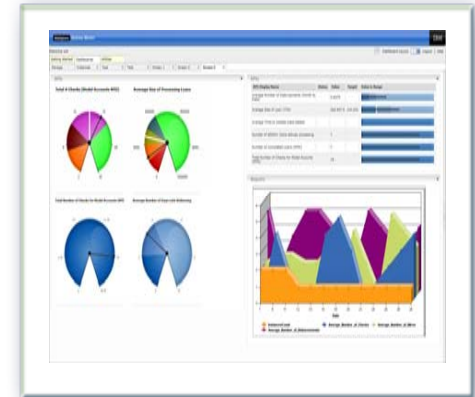


The event solution for CICS

- Simple events can be emitted *without change to existing business logic*
- Events are triggered when the predefined conditions are met:
 - Specified EXEC CICS Command, and
 - Specified file/program/service (for example), and/or
 - Running under specified transaction or program (for example)
- Filter conditions or predicates relating to event data can also control the event capture (for example “account balance must be less than 100 dollars”)
- Once conditions are met, CICS will
 - Capture application data specified for the event
 - Apply any policy relating to the event
 - Emit the event through its associated Event Processing Adapter
- Events are specified using the Event Binding Editor in the CICS Explorer
- CICS Explorer is used to deploy the event binding to CICS within a CICS bundle

Observe business processing scenario

- **Identify key points in order processing business logic**
 - order requested, order placed, order confirmed, order dispatched, order cancelled
 - Collect relevant contextual data associated with the event, including a way to correlate events for the same order, and emit event
 - Events sent e.g. to WebSphere Business Monitor
 - Observe orders being received, processed, cancelled
 - Study Key Performance Indicators
 - Numbers of orders received per week, time to process and dispatch orders, etc.
 - Take action when thresholds exceeded, when value of a customer's orders exceeds a certain amount
- **Application continues processing independently:**
 - Event instrumentation is 'non-invasive' to the application



Complex event scenario



- **Collect events relating to bank card usage**
- **Check for unusual patterns of behaviour**
 - New card ordered within a week of an address change request
 - Multiple online purchases
 - 2 or more cash withdrawals in quick succession
 - Purchases in different geographical locations in short period of time
- **Emit simple events to WebSphere Business Events**
 - Specify actions to take in WBE
 - Suspend card
 - Contact card holder

The Role-Based CICS Event Workflow



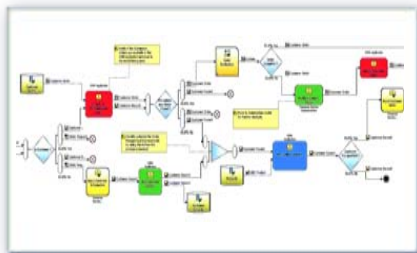
**LoB
Manager**



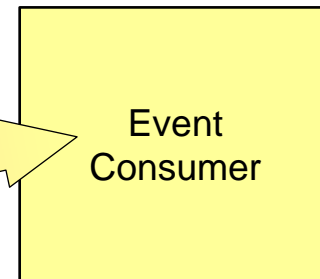
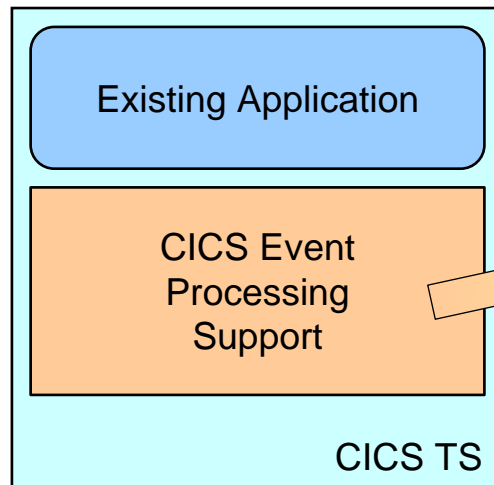
**Application
Analyst**



**Systems
Programmer**

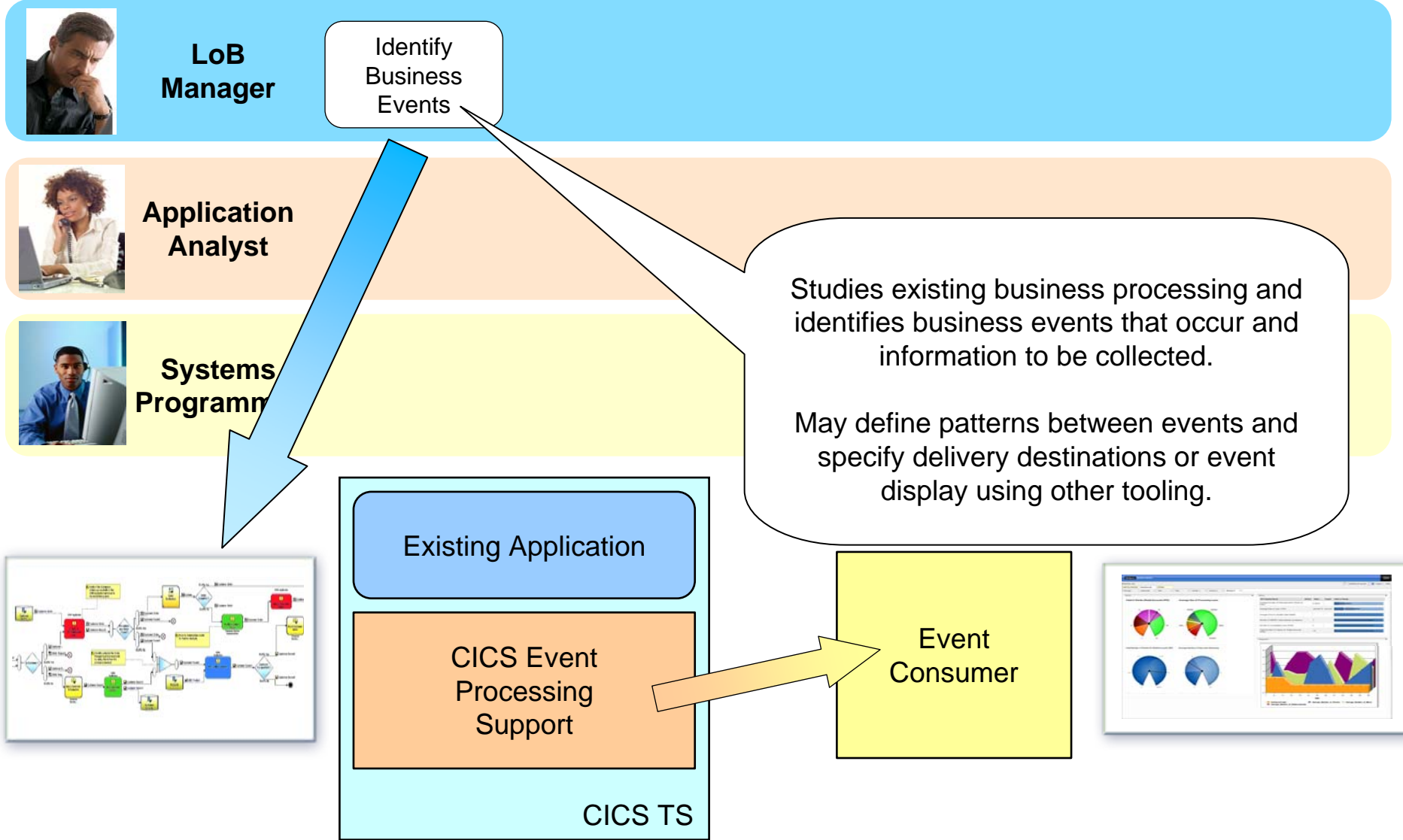


Business Modeler

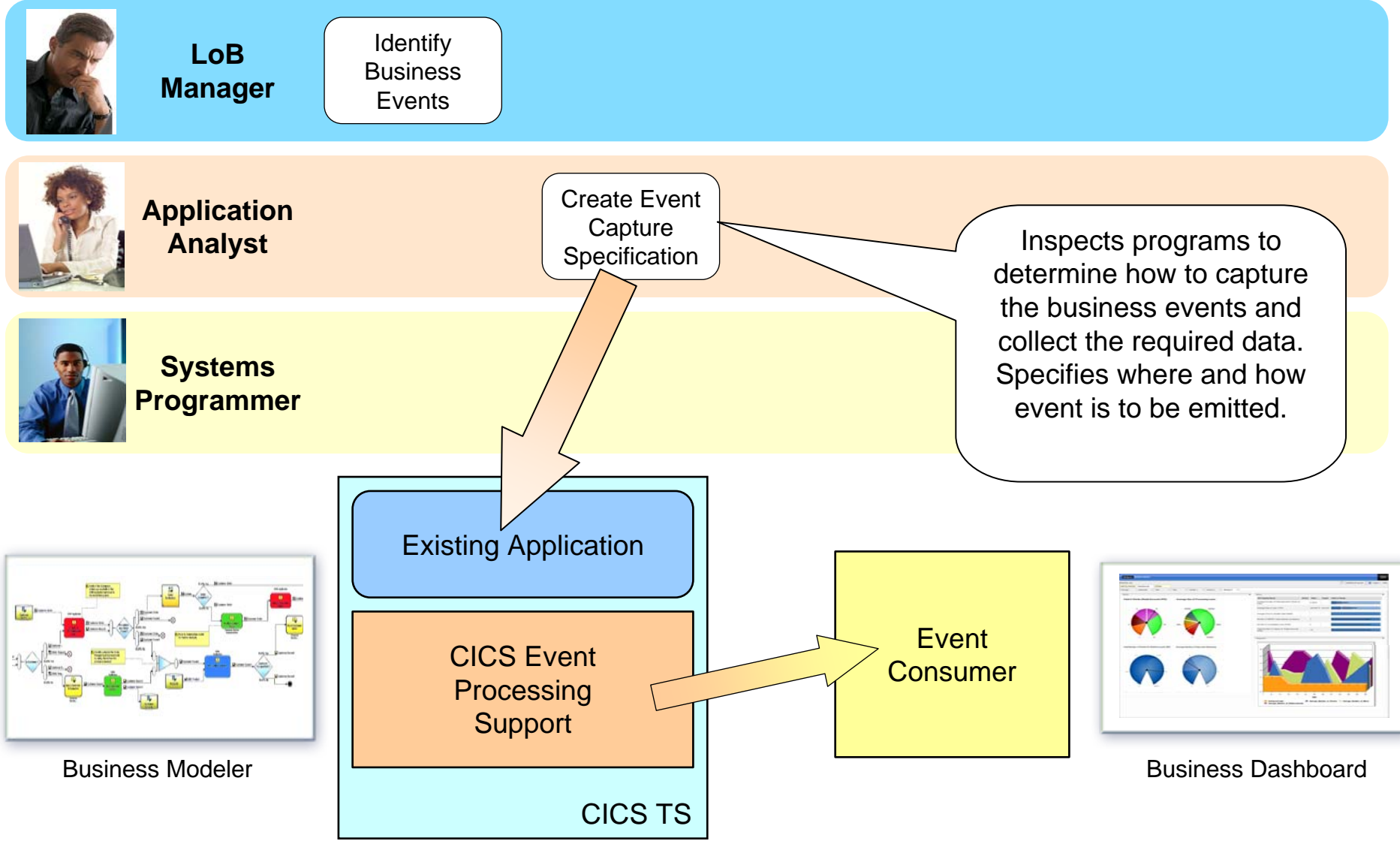


Business Dashboard

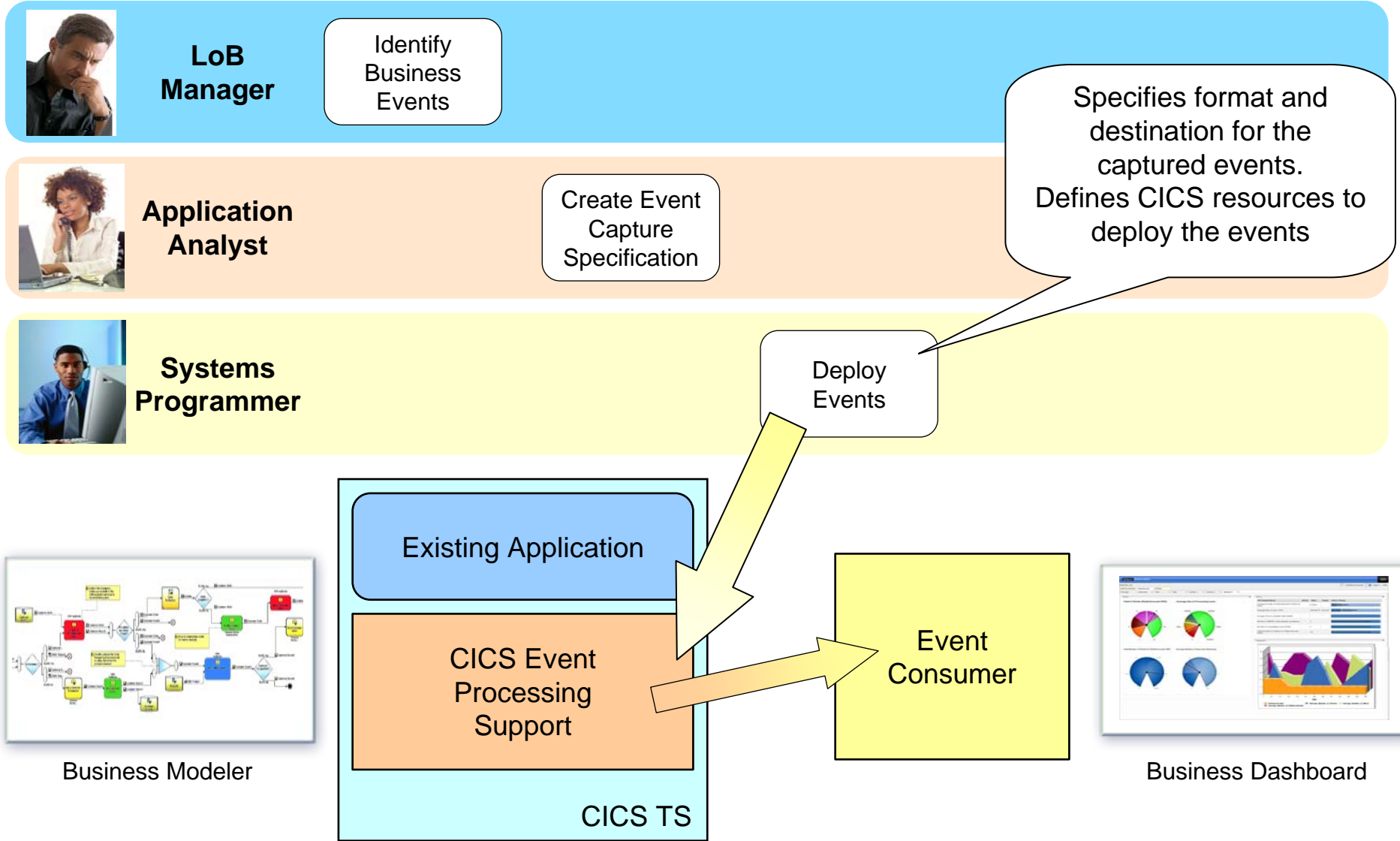
LoB defines the business events



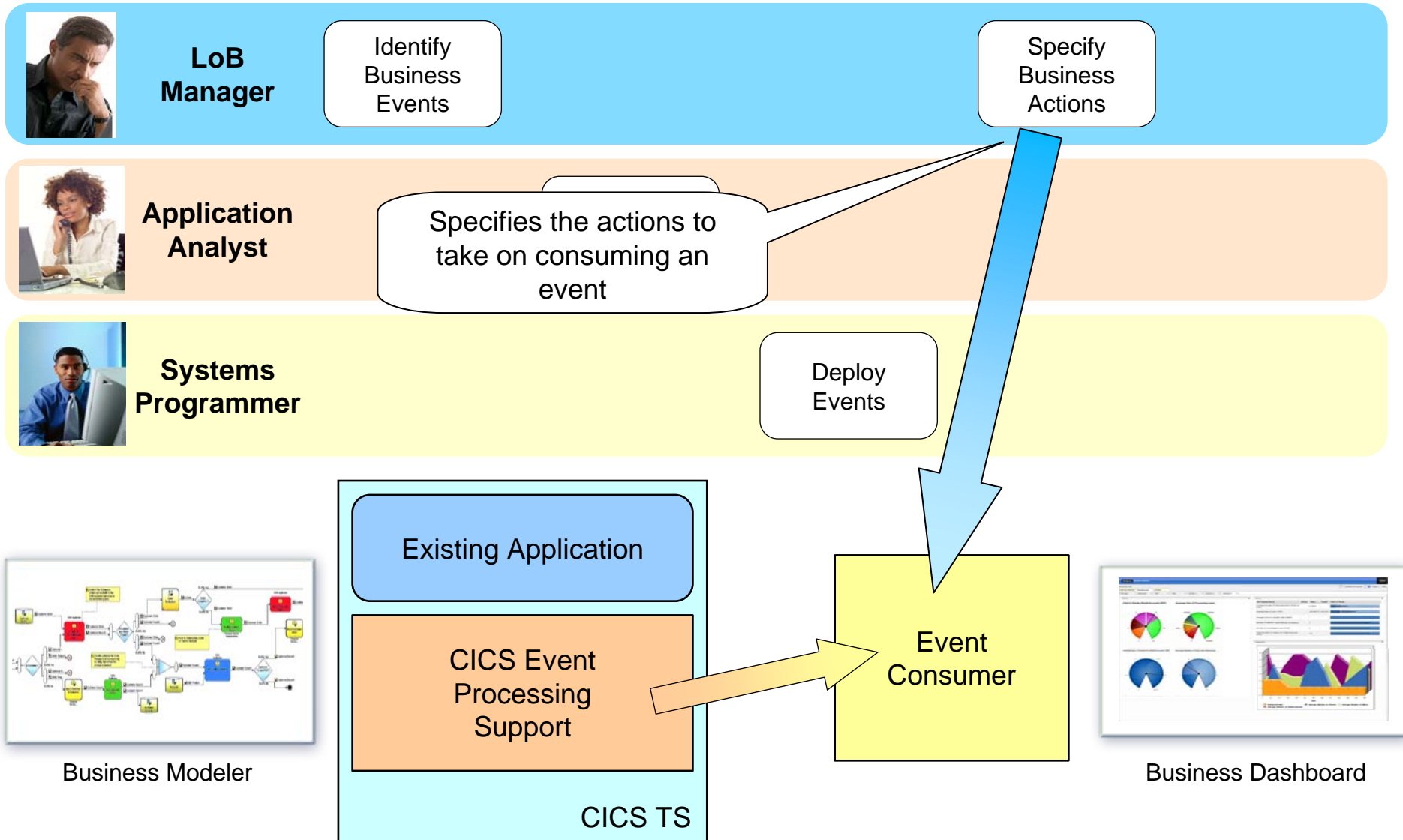
Application Analyst identifies events in application



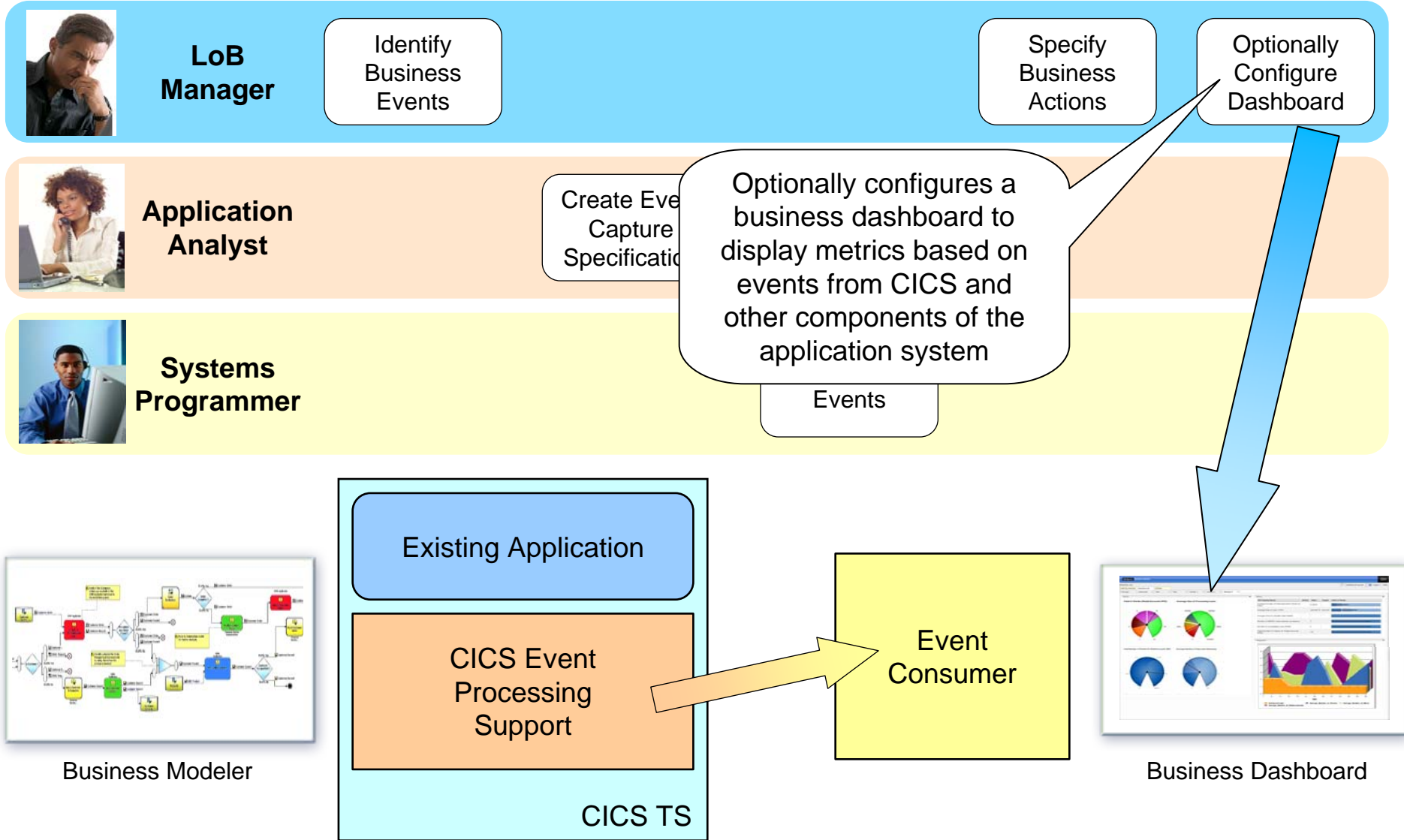
Systems Programmer configures the infrastructure



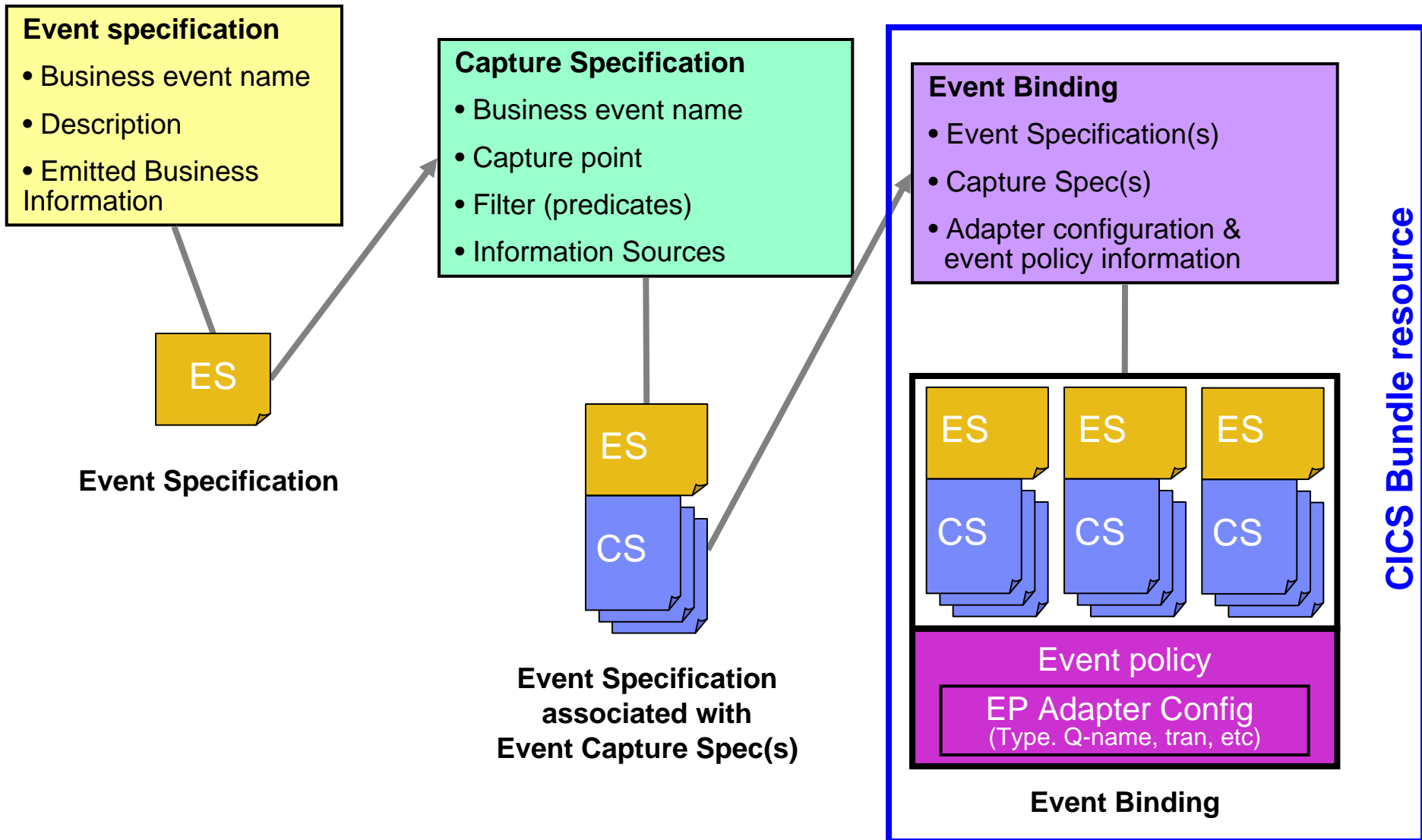
LoB defines the business actions



LOB may configure a dashboard



CICS Event Specification



CICS Event Specification Example

Event Specification:

Event name: `order_OverTenThousand_Received`

Event description: Whenever an order is processed that is for over 10 thousand, this event is triggered for display on dashboard

Emitted business information: `customer ID, OrderNumber`

CICS Event Specification Example...

Capture Specification:

Before EXEC CICS LINK command

← **Capture Point**

to PROGRAM(OrderDB)

← **Application Command Options Predicate**

from current_program = OrderUI

← **Application Context Predicate**

where OrderVal > 10K

← **Application Data Predicate**

OrderVal is in the Commarea or channel passed on the LINK

How to provide event data from data available from application and context ← **Information Sources (Captured Data)**

This identifies an EXEC CICS command and some filters, it does not 'point' directly at a specific location in the application code, this can be made more or less specific by use of filtering predicates

CICS Event Specification Example

Event Binding:

order_OverTenThousand_Received,

order_FromMajorCustomer_Received, ...

← **Event binding includes other related events**

EP Adapter = CICSTransaction

← **how events in this Binding are emitted**

The Event Binding

- **XML specification of one or more related business events**
 - Multiple business events can be grouped together
 - These events can be quite different from each other, but will share
 - The same Event Processing adapter, with the same configuration
 - The same event processing policy (e.g. transactionality)

- **Each event specification may have multiple capture specifications**
 - Typically, each event specification will be associated with one capture specification
 - A particular event may have several manifestations
 - e.g. “credit card usage” event may occur through several channels (Internet, in-store purchase, ATM machine)
 - Each capture specification must be able to provide the data to be emitted in the event and in the same format

Deployment of Event Bindings

- **The Event Binding is deployed into CICS**
 - Event Bindings are deployed via inclusion in a CICS Bundle
 - A CICS Bundle resource contains a collection of related CICS resources
 - An archive file containing resources, artifacts, etc. plus a manifest
 - Used for a number of new resources, including Event Bindings
 - Installing a BUNDLE into CICS will install the included resources
 - Create bundle and export to zFS from CICS Explorer
 - Define bundle resource including the zFS location
 - Deploying the Event Binding into a particular CICS region will resolve the capture specifications
 - CICS enables the capture specifications in the Event Binding so that they can be intercepted during runtime processing

CICS Event Capture options

- **Non-invasive**

- Declare event points in application logic without changing the application
- Use application knowledge to map business event onto points in the logic

- **Explicit API**

- **EXEC CICS SIGNAL EVENT**

- EVENT supplies an event identifier
- Data can be supplied as either FROMCHANNEL or Data area and length
- Identifier to be used in event specification

- Explicit way of adding a capture point to an application

- Allows exact pinpointing of the event point, and exact selection of relevant data
- Use to “event-enable” the application

- Define as event within an event binding

- Allows filtering and selection of data to use for different business events
- Allows event to be enabled and disabled
- ‘Fast path’ in tooling to simplify specification of explicit events

Eventable CICS Commands

- **Focus is on events of interest in business terms, so commands relating to system activity not eventable**
 - e.g. not ABEND, DUMP TRANSACTION, HANDLE CONDITION, SPI commands
- **Anything that starts work has a good likelihood of mapping to business events**
 - e.g. START, START TRANSID, LINK, INVOKE WEBSERVICE
 - Also enable event capture for program initiation via whatever means (e.g. Web services pipeline, entering transid at a terminal)
- **Getting data into or out of CICS can be a good way of finding out about business events**
 - e.g. RECEIVE MAP, RECEIVE, SEND MAP
- **Writes to CICS data resources (files, queues) may often occur when processing business events**
 - e.g. WRITE FILE, WRITEQ TS
- **Reads of CICS data resources are also interesting, as events do not only occur when data is updated**
 - e.g. READ FILE, READQ TD
- **No plans to event enable data oriented commands**
 - DB2, IMS, MQ
 - In a future release, might event enable these commands to get the additional application context
 - **but** only limited information about the command could be available to CICS & included in capture specs

Eventable CICS Commands

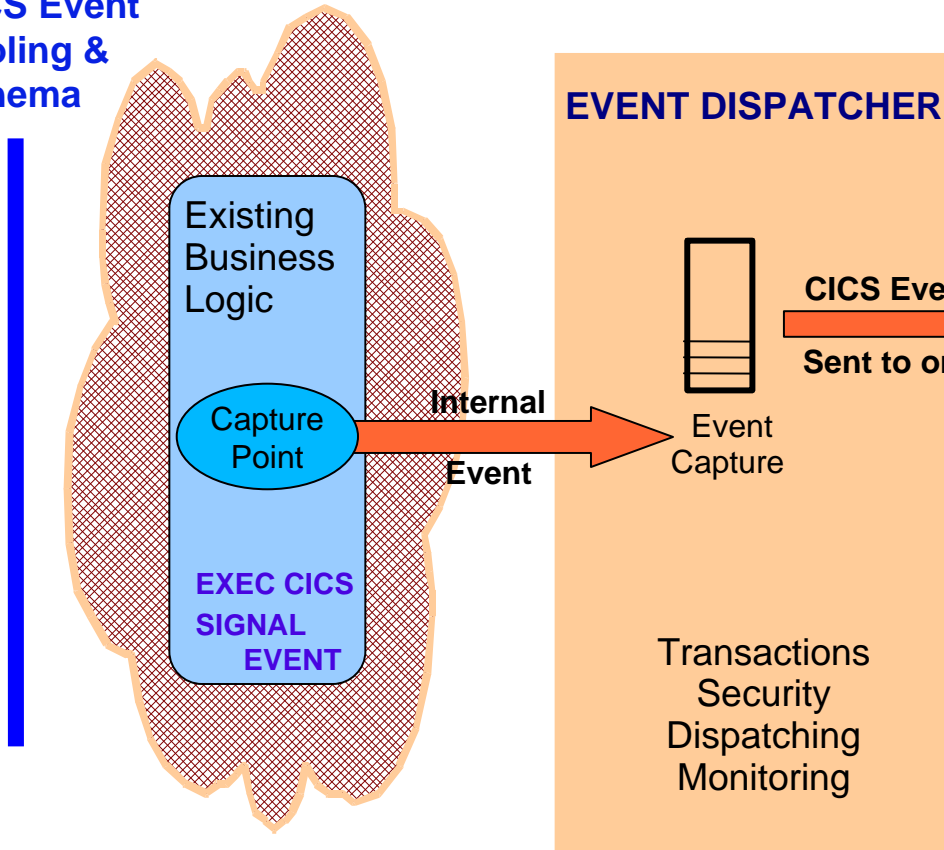
- **Channel commands**
 - PUT CONTAINER, START (TRANSID)
- **File Control**
 - WRITE, REWRITE, DELETE
 - READ, READNEXT, READPREV
- **Interval Control**
 - START, RETRIEVE
- **Program Control**
 - LINK, RETURN, XCTL
- **Scheduling Services**
 - START (ATTACH)
- **Temporary Storage**
 - WRITEQ TS, READQ TS, DELETEQ TS
- **Transient Data**
 - WRITEQ TD, READQ TD, DELETEQ TD
- **Web support**
 - INVOKE (WEB)SERVICE
 - WEB READ, WEB READNEXT
- **BMS**
 - RECEIVE MAP
 - SEND MAP
 - SEND TEXT
- **Terminal Control**
 - CONVERSE, RECEIVE, SEND
- **New APIs**
 - SIGNAL EVENT, INVOKE SERVICE
- **Program initiation**
 - Enable event when program starts

Filterable and Capturable Data

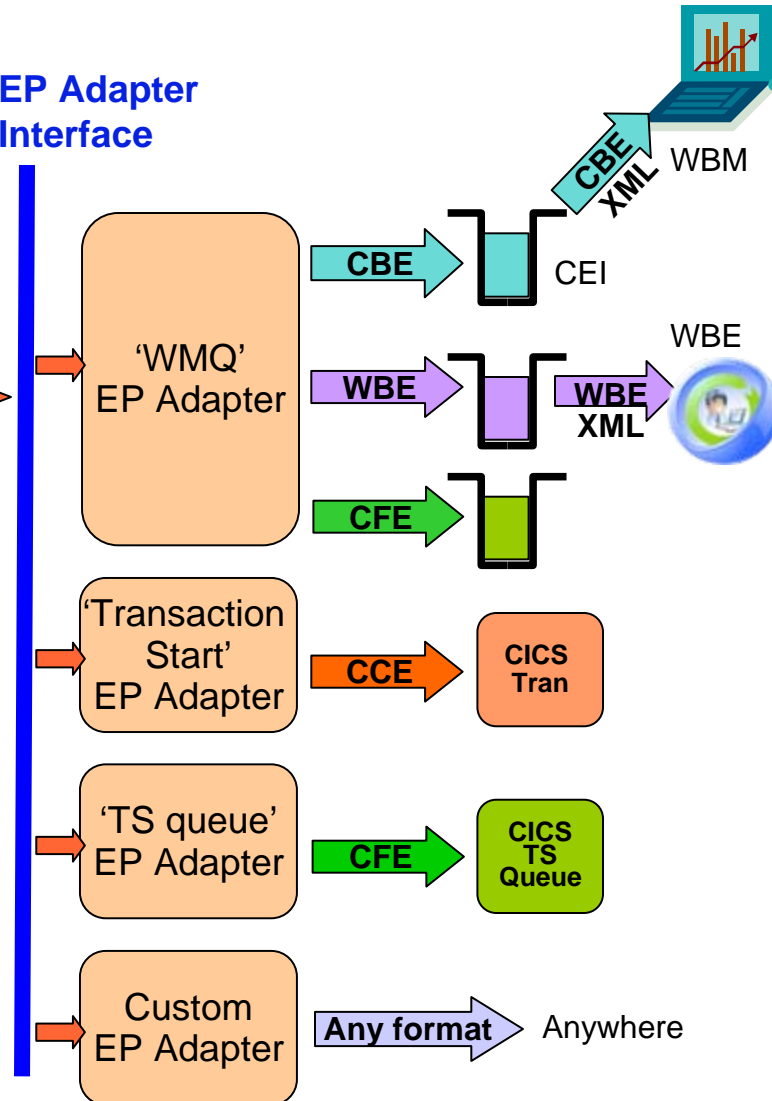
- **Application Context – applies to all commands**
 - Filterable (can be included in a predicate):
 - Tranid, Current program, Userid, Command response (OK/not OK)
 - Captured automatically:
 - UOWid, Network applid qualifier & CICS applid, Date & time
 - Capturable (can be information source for an item of emitted business information):
 - Tranid, Current program, Userid
- **Application command options and application data – Command-specific**
 - e.g. For RECEIVE MAP
 - Filterable and capturable: MAP (*primary predicate*), MAPSET, EIBaid, EIBCposn
 - **Primary Predicate** for each command is the data item on which filtering is *strongly recommended* for performance
 - e.g. For LINK
 - Filterable & Capturable: Program (*primary predicate*), Data from channel or Commarea
- **Most commands will be captured *after* they occur, some offer the option to capture *before* e.g. LINK**

CICS Event Processing Adapters

CICS Event Tooling & Schema



EP Adapter Interface



EP Adapters Summary

EP Adapter Type	EP Adapter	Event format	Transport	Intended consumer
Message Queue	WMQ Queue EP Adapter	CBE (XML)	WebSphere MQ	WebSphere Business Monitor or any application which processes a CBE
		WBE (XML)		WebSphere Business Events or any application which processes a WebSphere Business Events connector
		CFE (structure)		Application written to get the CFE from the message queue
CICS Transaction	Transaction Start EP Adapter	CCE (containers in channel)	Program start, with channel	A CICS application which processes the event data from the containers (or just takes action as a result of being driven).
CICS TS Queue	TS Queue EP Adapter	CFE (structure)	Temporary storage queue write	Test and debug
Custom	User-written			

Event Transactionality

- **Transactional option on the event definition**
 - Part of the advanced adapter options on an event binding
 - When set, causes CICS to wait for syncpoint completion before either emitting or discarding event (depending on syncpoint outcome)
 - For many events, will not want transactionality e.g. attempt to write to file could be as interesting as succeeding

- **Note**
 - Transactional events are not emitted until the UOW reaches syncpoint – for a long-running transaction, this could mean the events are not very close to real-time

Other event processing policy attributes

- **Dispatch Priority**
 - Specify priority of events in the event binding as Normal or High
- **Userid the EP Adapter is to run under**
 - Specify a userid under which the EP adapter will run
 - e.g. might be needed to allow access to required WMQ queue, or for actions carried out by custom EP adapter
 - ‘Use context userid’ will run EP adapter under the same userid as that running when the event was captured
 - By default, EP adapters run under CICS default userid
- **Transaction ID the EP adapter is to run under**
 - Normally runs under a default tranid
- **Can specify a different tranid or userid for charging**
- **Some performance implications of specifying tranid or userid**

CICS EP Tooling

- **Event Binding Editor is a plug-in to the CICS Explorer**
 - Create event binding(s) within a **Bundle Project**
 - CICS Explorer provides support to deploy bundles containing event bindings
- **Event Binding Editor**
 - Eclipse editor feature , operating on event binding files (type .evbind)
- **Event binding can be built up in stages**
 - Validated against schema each time it is saved, other validation as information added
 - A problems pane keeps a running record of all errors

Event Binding Editor

The screenshot shows the IBM CICS Explorer interface. The main window is titled 'Catalog_stock_status_check.evbind' and displays the 'Specifications' dialog for the event 'Catalog_stock_status_check'. The 'General' section is expanded, showing the event name and description: 'The stock level is low and there is no re-order in place.' An 'Add Capture Specification' dialog is open in the foreground, allowing the user to define a new capture specification. This dialog has a 'General Information' section with fields for 'Name' (set to 'Check_stock_status_on_rewrite') and 'Description' (set to 'The number in stock and number on order are available in the FROM parameter of the REWRITE command'). In the background, the 'Capture Specifications' section of the main dialog is visible, containing a table of existing specifications.

Length	Description
8	Program Name
4	Item reference number
40	Item description in catalog
4	Current number of items in stock
3	Number of items on order

Deploying an Event Binding

- Export bundle from CICS Explorer Resource Perspective onto zFS, where it will be unpacked in a directory
- Create BUNDLE resource definition with the location of the bundle directory on zFS as BUNDLEDIR
 - via CICS Explorer, CICSplex SM, CEDA etc.
- Install BUNDLE resource
 - Will install the event binding(s) in the bundle
 - Error is reported if the event binding fails validation checks carried out at install
 - Most validation errors will be prevented by Event Binding Editor

Summary

- **Non-invasive** emission of business events from CICS applications *without need to change* existing business logic
- SIGNAL EVENT API for explicit instrumentation of events
- **Event Binding Editor** tooling within CICS Explorer to create event specifications
- Event specifications deployed to CICS via bundles containing event bindings
 - Specifies event and the emitted business data, and how it can be detected and captured by the CICS runtime
 - Specify event capture points as EXEC CICS command (a subset of the EXEC CICS API) plus filtering on command parameters and data
- Events dispatched to specified EP adapter for formatting and emission to event consumer consumers including **WBE** and **WBM**
 - CICS provided EP adapters plus capability for custom EP adapters