BIAN
BIAN Global Summit
December 8th 2021

Extending BIAN to Support an Event Driven Architecture

Guy Rackham
BIAN Lead Architect
BIAN – Extending BIAN for ED^BA – 2021

Outline

BIAN - ED^BA - Overview

Topics Covered in the next 20 minutes…

1. What is an Event Driven Architecture (EDA) for BIAN? - ED^BA
   a. Business events can engage message queues and events to call on services
   b. There are also business event’s that result in less obvious ‘notification’ traffic
   c. Any event may trigger a ‘cascading’ multi-threaded reaction across the portfolio
   d. Events can result from internal scheduled actions as well as external requests

2. What are the key technical challenges?
   a. What types of business event are there to consider?
   b. Who needs to know? – BIAN will be expanding ‘First Order Dependencies’
   c. To scale, declared interest and interpretation is handled external to the source

3. Where are we up to now?
   a. We have draft standard state transitions matched to Service Domain types…
   b. … and have defined the key design aspects for the First Order connections…
   c. …and we are trialing a deployment approaches in the EDA task force under AF&F

20 Minutes, 3 Topics
1. What do we mean by ED^BA?

**Event Driven Architecture (ED^BA) – Transactional Message Exchanges**

The term event driven is often associated with a messaging mechanism. A business event causes the request. The receipt and response to the request can be considered as another business event. Platform events trigger the message exchanges.

The exchange shown involves two discrete but linked/associated 'messaging' events - the *call* and the *response* - over a pre-established connection.

BIAN anticipates the use of this type of message exchange.
1. What do we mean by ED^B^A?

Detection and notification of ‘business events’ defines a second type of exchange traffic.

Some refer to “pull” (called) and “push” (notification) traffic.
1. What do we mean by ED^BA?

When Service Domain service dependencies are fully implemented in the application architecture it can support a highly reactive type of business choreography...

...the response to an event in one Service Domain results in a 'cascade' of (independent) concurrent interactions expanding throughout the component network.

Predefined Dependencies Determine the Collective Response...
1. What do we mean by EDBA?

Event Driven Architecture (EDBA) – Note: The Trigger can be External or Internal

Whatever the source, the processing results in a material state change…

Handling a service request and internally scheduled activity can both lead to notifications

The association between handling offered services, internal scheduled actions and any business events generated within the Service Domain is fully encapsulated.

Service Domains Operate Autonomously and are Loose Coupled
1. What is an Event Driven Architecture (EDA) for BIAN? - ED³A
   a. Business events can engage message queues and events to call on services
   b. There are also business event’s that result in less obvious ‘notification’ traffic
   c. Any event may trigger a ‘cascading’ multi-threaded reaction across the portfolio
   d. Events can result from internal scheduled actions as well as external requests

2. What are the key technical challenges?
   a. What types of business event are there to consider?
   b. Who needs to know? – BIAN will be expanding ‘First Order Dependencies’
   c. To scale, declared interest and interpretation is handled external to the source

3. Where are we up to now?
   a. We have draft standard state transitions matched to Service Domain types…
   b. … and defined the key design aspects for the First Order connections…
   c. …and we are trialing a deployment approaches in the EDA task force under AF&F
2. What are the Key Architectural Elements?

### Event Driven Architecture (ED\textsuperscript{BA}) – Types of Business Event

Types of Control Record have their own Types of Business Event.

<table>
<thead>
<tr>
<th>Functional Pattern</th>
<th>Produced Asset</th>
<th>Function Specific Internals</th>
<th>Event Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRECT</td>
<td>IP Plan (CREATE)</td>
<td>Define the organizational roles and responsibilities, the work streams and activities and their associated metrics, scheduling, dependencies and outputs/deliverables for development and deployment of intellectual property (IP)</td>
<td>Stages in the assignment, development &amp; deployment of IP</td>
</tr>
<tr>
<td>MANAGE</td>
<td>Processing (PERFORM)</td>
<td>Support forced sequential and concurrent (event driven) processing activities. (Can be multi-threaded and include start/end dependency handling)</td>
<td>Stages in fulfilling the arrangement/workflow</td>
</tr>
<tr>
<td>ADMINISTER</td>
<td>Directory (REGISTER)</td>
<td>Classify, verify and maintain structured reference information and associations (e.g. memberships)</td>
<td>Stages in capture and qualification</td>
</tr>
<tr>
<td>DESIGN</td>
<td>Evaluation (EVALUATE)</td>
<td>Apply tests and analytical models to consolidated information to qualify, certify and assess entities and their activity</td>
<td>Stages in the evaluation</td>
</tr>
<tr>
<td>DEVELOP</td>
<td>Allocation (PROVIDE)</td>
<td>Inventory, allocate and track assignable resources</td>
<td>Assignment stages</td>
</tr>
</tbody>
</table>

The business events are characterized by the associated ‘produced asset’

BIAN anticipates the use of this type of message exchange.
2. What are the Key Architectural Elements?

Event Driven Architecture (EDBA) – Who needs to Know (First Order Connection)

Pre-ordained links exist between Service Domains

It should theoretically be possible to identify all required Service Domain interactions (both ‘pull’ & ’push’ in nature)

BIAN Wireframe and Business Scenario artifacts contain the required service connections between Service Domains

The collection of all identified called and offered Service Domains can be cataloged for each Service Domain (its “First Order Connections”)

Discovery will continue as evolving practices and new capabilities cause new interactions

BIAN is adding First Order Connections to the specification
2. What are the Key Architectural Elements?

Event Driven Architecture (ED^3A) – To Scale the Receiver Interprets the Message

The source is in control of what changes trigger events, but not who subscribes

The ‘source Service Domain handles its internal operations that trigger notification...

...but as it has no control over the number and needs of consumers, scalable messaging middleware must support communications and the receiver must interpret the content.

The communication mechanism ensures ‘delivery’
1. What is an Event Driven Architecture (EDA) for BIAN? - ED^B^A
   a. Business events can engage message queues and events to call on services
   b. There are also business event’s that result in less obvious ‘notification’ traffic
   c. Any event may trigger a ‘cascading’ multi-threaded reaction across the portfolio
   d. Events can result from internal scheduled actions as well as external requests

2. What are the key technical challenges?
   a. What types of business event are there to consider?
   b. Who needs to know? – BIAN will be expanding ‘First Order Dependencies’
   c. To scale, declared interest and interpretation is handled external to the source

3. Where are we up to now?
   a. We have draft standard state transitions matched to Service Domain types…
   b. … and defined the key design aspects for the First Order connections…
   c. …and we are trialing a deployment approaches in the EDA task force under AF&F
3. Where are we up to now?

Event Driven Architecture (EDBA) – Standard State Transitions

Rather than triggering on create and update, more specific state transitions can be defined

The types of resources involved in supporting production and the Service Domain’s “produced asset” provide the context to define standard states and transitions

More specific event types improve the EDA design
3. Where are we up to now?

Event Driven Architecture (EDBA) – Specifying First Order Connections

The Meta Model has been updated to record these and we will develop techniques.

Different considerations exist if deploying First Order Connections in specific situations.

To date most attention has been on using BIAN aligned APIs to wrap existing systems.

In the future systems assembled from BIAN aligned ‘components’ will also be important.

First Order Connections – SD to SD connections – both internal within and external to an Application Cluster (Stand-alone system)

<table>
<thead>
<tr>
<th>Called SD</th>
<th>Service Type</th>
<th>SD/CR/BQ Trigger</th>
<th>Event Type(s)</th>
<th>Message I/O Payload</th>
<th>Access/Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifies the called Service Domain</td>
<td>(A – Instantaneous) B – Real-time C – Best-time D – Delayed response E – Notification F – Scheduled</td>
<td>Defines the trigger scope for notification create/update</td>
<td>Selected events defined by FP</td>
<td>SD/CR/BQ extracted message content (I &amp; O as appropriate). NB: Scope defined by sender</td>
<td>Suitable access/security controls that are applied at the time of exchange. Pre and Post conditions</td>
</tr>
</tbody>
</table>

Note: In an Application Cluster connections must also be provided for synchronization between local proxy/host and external proxy/host Service Domains

SERVICE TYPES:

(A. Instantaneous – Supports concurrent data access. Note: does not use queue/event mechanism)
B. Real-time – quick response expected – Caller will wait for response
C. Best-time – timely response expected – Caller may multi-thread activity
D. Delayed response – response expected later – Caller must either poll for response or be notified

E. Notification on update – Caller subscribes to state change notification
F. Scheduled synchronization – Caller subscribes to scheduled updates

Our focus should expand to consider new solution development.
We are trialing a scalable approach in the EDA task force under AF&F

Twelve technical challenges to supporting standalone component application deployments:

1. Scope: Application Cluster
2. First Order Connections
3. EDA Notifications
4. Service Access Control
5. User Access Screens
6. Command & Control
7. Preconfigured Containers
8. Analysis & Reporting
9. Organizational Deployments
10. Platform Optimization
11. Versions & Releases
12. Functional Variations