# BankERRM™

**PBORM™**

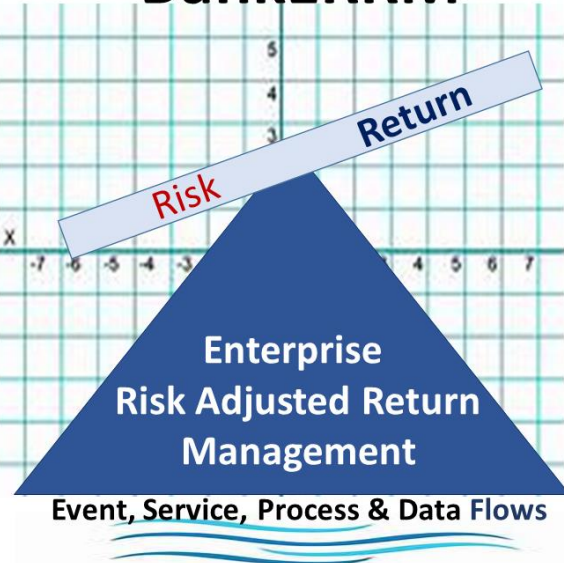Process Based Approach
to managing Operations and
Operational Risk Management

# Enterprise Risk Adjusted Return Management
## Interviews & Discussions

An interview (Q&A) format with Mr Guy Rackham
**Lead Architect** at Banking Industry Architecture Network (BIAN)
January 2023

## An interview (Q&A) format with Mr Guy Rackham
## Lead Architect at Banking Industry Architecture Network (BIAN)

Guy Rackham has worked in finance and technology for 40 years, specializing in business and technical architecture development and approaches for more than 25 years. Guy has worked in senior technology positions at international banks including Goldman Sachs and Standard Chartered and at several major software and consulting firms including Logica, Coopers & Lybrand, IBM and PriceWaterhouse. For the last seven years Guy has worked as an independent specialist spending the majority of his time as the lead architect at BIAN.

Prior to BIAN Guy led the development of the Component Business Modeling (CBM) approach that was used widely within IBM consulting. This ground-breaking work was recognized with Guy being the first strategy consultant promoted to IBM Distinguished Engineer. Through BIAN and other client assignments he continues exploring how advanced technologies and working practices can be best leveraged in commerce.
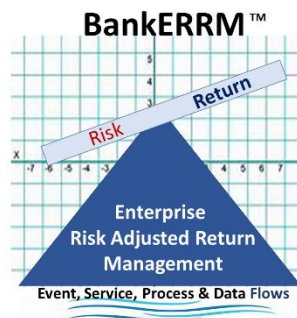
**Background:** Enterprise Architecture: Services Oriented Architecture, Micro Services Architecture and Event Data Architecture

| SOA | Microservices |
|-----|---------------|
| **It is based on the service sharing principle** | Each service is a single entity that can do only one thing, which is based on a single data unit |
| **It is based on the concept of business functionality reuse.** | It uses the bounded context concept. |
| **It uses the entire subsystem as a service and focused on coupling for reusing the service.** | It is focused on service decoupling and logic separation. |
| **Communication takes place using an enterprise service bus or message queuing.** | Communication takes place using the REST/SOAP API layer. |
| **Used in enterprise projects.** | The projects are smaller in scope (e.g. Treasury / Loans-Front Office). |
| **SOA supports the propagation of the heterogeneous protocol used in multiple subsystems.** | It reduces the choice for integrating the subsystem/services, it supports integration for the service which uses common communication protocol. |
| **DevOps and CICD are yet to get popular with SOA based architecture.** | It is focused on DevOps, cloud-based, container-based deployment with the CICD pipeline. |
| **Support Multithreaded with multiple parallel IO operations.** | Most of the application is single-threaded and in comparison, has less IO operations. |
| **SOA is based on model programming.** | Microservices is based on component-based programming. |
| **It can be developed by using MuleESB, MS BizTalk as a middleware component.** | It can be developed by using the NETTY or Spring cloud library. |

**Event driven architecture** is a software architecture approach that promotes the awareness of the production, consumption and response to events. Event-driven architecture works well for changing market conditions, customer behaviour and business conditions.

The three important aspects of EDA are (a) Architecture & Technology (b) Operations management and (c) Governance.
Ref:
https://developer.ibm.com/technologies/messaging/articles/advantages-of-an-event-driven-architecture/
https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/event-driven

**Q1:** The traditional approach in banking automation has been to use large, discrete product-centric / Line of Business centric systems. The resulting  processing silos are barriers for maximising risk-adjusted returns. How does BIAN's component based enterprise architecture provide the real-time architectural integration across Retail, Corporate, Treasury, Finance and Support Functions?

**A1:** BIAN defines 'product fulfillment' Service Domains that orchestrate the product specific activities, but that externalize or delegate to more general Service Domains that support both more generic activities for which a common solution can be used (such as accounting, document handling, payments, custody) and cross-product activities (such as credit management, risk management and relationship development). Those cross product activities, where appropriate can be implemented to support real-time perspectives, perhaps integrating event driven communications (though BIAN does not prescribe any particular implementation approach – it's only a conceptual design).

**Q2:** Can you highlight the key architectural design elements of the BIAN's Enterprise Architecture framework that would enable the harmonious working of SOA, Microservices and Event Driven Architecture components?

**A2**: BIAN's Service Domains are intended to relate one to one with a service center in a SOA. Microservices can be a little more confusing as some people look at the definition of a micro service (autonomous, encapsulated, service enabled etc.) and say that's the same as a Service Domain.

I think of a Service Domain more as an organizational construct – an assignable responsibility partition like a contained business unit. In contrast I think most consider a micro service to be an application/SW partition. So they can be aligned one to one, but be aware that they are really different concepts one defining organizational partitions the other SW application components.

I would expect micro services would more often relate to finer grained partitions of activity than Service Domains, and may support specific SW requirements such as implementing SW utilities.

Event Driven Architecture is a pattern of interaction that we are keen to enable between Service Domains in BIAN and are doing a lot of work right now to determine how that is properly defined in detail. Though many exchanges between Service Domains can follow a more traditional request/response flow, clearly for the more collaborative/networked business interactions the notification dependency supported through EDA is critical.

Look out for e BIAN Application Development Practitioner's Guide later this year to expand on this.

**Enterprise Data model**
**Background**
A taxonomy is a high level business structure that groups related business components together and defines their relationships.

Ontologies comprise classes, properties and instances. Ontology as a domain model is:
- Explicit, as it could be a sum of units and entities. It also states how they relate to each other;
- Shared, as there is a consensus on model with an agreed vocabulary (such as a standard called EDM Council's FIBO Ontology;
- A specification for the "conceptualisation" of a system. It has logical theory and is characterised by specific formal properties and has defined objectives;
- A specification for a logical model.

**ACTUS-BIAN work**
The ACTUS Financial Research Foundation has been established as the single IPR-holder of the ACTUS Standards.
https://www.actusfrf.org/about
The ACTUS project aims to remedy this weakness by creating a global standard for the consistent representation of financial instruments. The centrality of expected cash flows for financial analysis is obvious and undisputed. Less obvious is the role of the financial contracts. Financial contracts are mutual agreements between counterparties to exchange cash flows. The agreements are written by lawyers in different languages, legal terminology, and jurisdictions. This leaves us with a plethora of terms and different contracts.
The goal of ACTUS is to **break down the diversity in financial instruments into a manageable number of cash flow patterns** – so called Contract Types (CT). ACTUS defines two standards: Data and Algorithmics.

Questions:

Q3: Can you briefly explain how the BIAN enterprise framework facilitates the management of (a) Enterprise Master Data (b) Metadata and (c) Reference Data?

A3:

a) In BIAN there is no shared centralized data – every Service Domain governs its own view of the world in its own information base and information is exchanged between Service Domains by 'value'. Meaning one Service Domain can obtain information from another, but it has to interpret the value provided and relate it to its own information perspective as it feels is appropriate. The role of some Service Domains is to maintain general information that can be retrieved and interpreted by others – such as market information, directory or reference details, bank policies and guidelines etc.

b) Meta Data – all BIAN content is semantic and descriptive of the information concepts it represents. So for example BIAN will note a **customer reference** is required to indicate that some unique way to isolate a customer using some identifier applies, without stipulating any specific physical implementation of that identification. The BIAN **BOM** (Business Object Modeling) provides ever expanding detail of the conceptual specification of all information content, but to be fair that's not really meta data as much as definitional content.

c) Reference Data – as noted, there is no concept of a central shared reference database – Instead there are specific directory or catalog type Service Domains that maintain information structured in a manner for others to reference through their offered services – for example Party Reference Data Directory.

**Q4**: Can you outline the main points (principles, standards) of BIAN that enables a bank to build an enterprise data management capability ?

**A4:** The BIAN BOM, (that we are actively mapping to ISO20022 by the way) provides an increasingly rich conceptual business object model aspiring to cover all business information content. Service Domain Control Records define the selection of business information needed to support their specific responsibility and this information content is cross referenced back to the BOM. Note that a Service Domain's Control Record may contain information attributes selected from multiple business objects in the BOM, and may apply context specific names in cases (for example "party type: employee" as defined in the BOM may relate to the "Customer Relationship Manager" field in a Control Record). Also at this time though the goal is that every attribute in every control record is represented in the BOM, both views are being extended in parallel and the formal mapping between them is partial (but being improved daily!). So to answer your question, the combination of the BOM as a global business information specification, mapped to the business context specific extracts defined by the Control Records provides a framework for enterprise data management. Furthermore the service dependencies between Service Domains and the fact that each Service Domain 'governs' its own discrete collection of business information can be used to trace information governance obligations across activities within the bank.

**Q5**: Are you working with bank-data management initiatives like the EDM Council? Or with entities like International Swap & Derivatives Association?

**A5:** Our main coordination is aligning with ISO, though BIAN has MOUs with many other standards bodies and our policy is not to create anything that competes with any other standard in general. Unfortunately we have limited capacity to maintain active collaboration with too many other standards bodies.

**Background: Data as a Service (DaaS),Data Virtualization**

**DaaS** is an architectural framework that can be used for designing and developing a set of reusable data services. The enterprise banking DaaS framework can seamlessly route data across front, middle and back office functions.

The significant advantages that DaaS offers include reusability and data democratisation over the Cloud.

DaaS is for an enterprise approach to data management and is most appropriate for ERRM implementation.
Data as a service (DaaS) is a SOA-based service that transforms raw data into assets that can be used for decision-making. The data assets are delivered as requested via a standard connectivity protocol in a pre-determined, configurable format and frequency for internal and external consumption.

Banks and other financial institutions that have implemented service-oriented architecture, process automation are implementing DaaS. DaaS enables banks to identify, access, manage, secure and deliver information in real time regardless of the type of information or the platform on which it is stored.

**Data Virtualization** is a data management approach that allows an application to retrieve and manipulate data without requiring the details about its physical location and the way it is formatted at source.

Question:

**Q6a**: Can you briefly explain how the BIAN enterprise framework (SOA, MSA) facilitates implementing Data as a Service?

**A6a:** At the risk of sounding too radical – in my view data is an artificial construct – invented so we can automate processing logic in computers. Like with algebra – if take "x" or "y "out of context of the activity being represented by the equation they become meaningless. So one thing I think it important when discussing data and Service Domains is to repeat that there is no concept of shared data in BIAN, each Service Domain maintains its own knowledge base, and may interpret this as data for its internal processing at it sees fit. But any exchange of information is by value with a clear definition of what the value represents in business terms. Back to your question, at this stage our focus is on transactional business information exchanges (through service operation interactions).

We recognize that there is a second type of information exchange – a reporting/analytics capability for each Service Domain to provide read only "query" access to its historical information. Each Service Domain has an analytical object responsible for supporting this reporting capability. Data events (as opposed to business events that trigger service operation interactions) are likely to apply in this analysis and reporting capability of the Service Domain – but we are currently expanding this aspect of the BIAN specification. If you are into CQRS – the business event/transactional exchanges fit best to the "C" and data events and reporting better align with the "Q" space.

**Q6b**: How does BIAN's Event Driven Architecture leverage/support data as a service or Data Virtualization?

**A6b**: I'm not sure it does… We assume that any Service Domain obtains business information from another in the context of a business requirement (I am avoiding describing the mechanism as that would be a distraction – suffice to say its state change and event driven). The Service Domain is then responsible for interpreting that information for its own use. We would expect that an aspect of the offered service specification would describe the information in sufficient detail for the receiver to interpret the values and if appropriate the properties of that information (such as its age/integrity etc.) to confirm it is fit for the receiver's business purpose.

Note also that we believe that all information exchanges should be a design time arrangement. Information is not broadcast to all comers. Instead, for security and performance purposes, each allowed exchange between 2 Service Domains is pre-determined and specified – including the run-time authentication and authorization requirements when appropriate (we call these First Order Connections and are adding them to the specification during 2023 and beyond).

# Event Driven Architecture-System Dynamics

**Q7**: Can you briefly educate a reader on BIAN's draft standard state transitions?
**A7:** These will be documented in the upcoming BIAN Application Development Practitioner Guide, but for now we have isolated the different types of resources that need to be in place/and set-up for the Service Domain to perform (its "Production" resources/assets) and the types of asset or resource that it generates (its "Produced" assets). We have defined the general business triggers and possible states for all of these resource/asset types (different combinations of which are needed depending on a Service Domain's Functional Pattern). We have also developed finite state machine views that detail allowed 'pre state, event, post state' combinations as the basis for implementing a state machine and within this supporting business event driven communications.

**Q8**: Enterprise Risk Adjusted Return model can be used using systemic dynamics. Will BIAN's state transitions support configured and intelligence-based dynamic state transition?
**A8:** I'm not sure as yet – other than to say we do model the state of all involved assets/resources with the Service Domain so a number of interesting things might be possible – but its early days to predict what innovations members will develop.

**Q9**: Is there any working alliance with an entity specialising in System Dynamics?
**A9**: Not to my knowledge – but we do hear of all manner of uses of the BIAN model beyond the membership, so I wouldn't be surprised if there was some BIAN related incursion into that space – it would be interesting to find out.

Background - **Enterprise Standardised Operating Model**

The resolutions for correcting the high degree of variations in business delivery can be corrected by (i) improved product definition (ii) better alignment of I.T. processes with Business Processes (iii) improving staff skill and organisation structure and (iv) implementing an improved enterprise control framework with a focus on preventive controls.

BIAN has placed an emphasis on SOM as *standardization paves the way for easier reuse of functions in heterogeneous business environments.*

**Q10:** Is there a BIAN framework implementation recommendation that encourages banks to identity and approve variations in business delivery and document the standard enterprise operating model?
**A10:** There is a lot of work going with BIAN looking at tooling support, adoption requirements and more recently recognising the need to help define how the BIAN conceptual designs might be interpreted in different logical and even physical designs. It is really in the logical and physical interpretations that these business and operational variations occur, but its an area where as more people apply the model that BIAN might be a clearing house for practical suggestions.

**Q11**: If so, can you please share some tasks for the recommendation? E.g. Identification of Business delivery variations, Banking product definitions, Key Financial data-item definitions as per EDMC & ISDA data definition standards.
**A11**: As the previous answer implies – this would be great to do, but we're not there yet.

**Background – BASEL Risk Management recommendations**

The banking industry view is that the following BASEL requirements cannot be met correctly/accurately unless a bank has an Enterprise Architecture and enterprise data management capability:

- Enterprise Stress Testing BCBS 427, BCBS 450
- Intraday Enterprise Liquidity Management-BCBS 248  (requires a real-time **Enterprise Liquidity Hub**)
- Fundamental Review of the Trading Book – BCBS 265
- Interest Rate Risk in the Banking Book-IRRBB Principles of BCBS 108 and BCBS 368
- Net Stable Funding Ratio BCBS 295 and
- Liquidity Coverage Ratio BCBS 272

The following evolving requirements also  require an Enterprise Architecture & Enterprise Data Management capabilities:

- Credit risk– constraints on use of internal model approaches (BCBS 362 & 424)
- Revisions to the securitisation framework (BCBS 374 & 424)
- Counterparty credit risk SA counterparty credit risk (BCBS 279 & 424)
- CVA risk Review of the CVA risk framework (BCBS 325 & 424)
- Step-in risk (BCBS 398)
- Capital floors (BCBS 306 & 424)

*BCBS – Please refer www.bis.org

**Background**: SOA, MSA and EDA will certainly help a bank meet the evolving BASEL risk-return management requirements.

**Q12** – Has BIAN made an assessment of its framework's ability to support a bank with the following:

- A. Real Time Treasury Management
**A12A:** A very interesting use case is under consideration using the ACTUS standard integrated with BIAN. It is being developed in the Investment and Trading (I&T) WG – we'll have to see how far we get.

- B. Enterprise liquidity hub (across cash flows from Retail, Corporate and Treasury)
**A12B:** The same I&T use case will cover liquidity perspectives and the concept of Service Domains that act as consolidators, analysers and reporters of transactional information is included in the use case.

- C. Real time nostro management (implementing SWIFT's solution for BCBS 248)
**A12C:** The Service Domains can provide a high level framework to frame this out, but its probably more of an implementation issue (i.e. in the logical and physical interpretation of the conceptual BIAN Service Domain designs).

- D. Event based automation that provides the capability to *identify and mitigate* risks simultaneously
**A12D:** This I would consider a specific use case to define and develop – rather like the I&T use case. I do believe the Service Domain structures would provide some unique and powerful options should anyone want to tackle it. The Fraud WG has put together an extremely powerful risk management wireframe that would provide a foundation for this – it should be published some time this year.

- E. Omni channel platform
**A12E:** The BIAN External Access Framework includes Service Domains that specifically addresses this requirement (channel agnostic access), in line with the more specific FAPI requirements arising from initiatives such as PSD2.

**Background: Semantic API for Open Banking**

BIAN has enriched Semantic APIs with a full linkage to the BIAN's Business Object Model. BIAN Semantic API specifications are often misunderstood as ready-to-use specification.

**Q13**: What are the main tasks for a bank's team to make BIAN's semantic APIs operational?

**A13:** To adopt BIAN in general the most common pattern I've seen is that a core of committed business and technical architects, with management support, invest in learning the BIAN model and its underlying architectural properties. This group can then seek out pilot initiatives to aid adoption through proving value and integrating the various tools and techniques needed to support adoption. For the BIAN Semantic APIs – this is perhaps the primary focus of the BIAN membership to determine how and to develop tools and guides that support adoption. Already several powerful capabilities and training options are available, but these are likely to be expanded rapidly. I'd browse the BIAN site and portal to see what's there. We hope to add tooling to help users create their own compliant wireframes and business scenarios later this year which we hope to aid adoption.

**Q14**: Does BIAN emphasise on the design principles of REST i.e. representational state transfer architectural style?
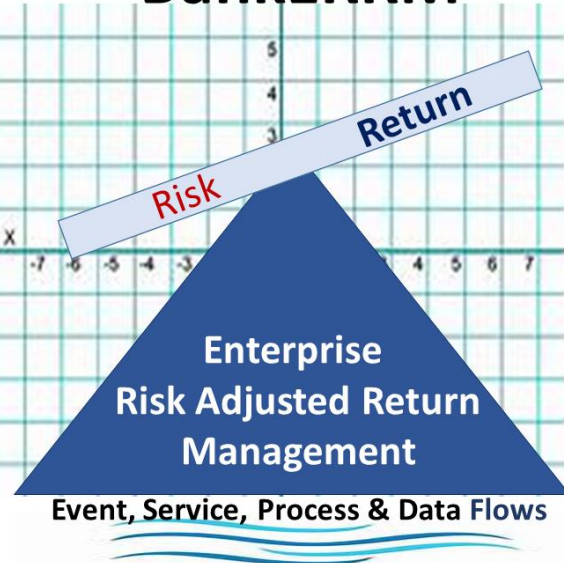
**A14**: This is a mystery to me to be honest – REST is an architectural style defined to support the efficient exchange of web pages. Given REST is the prevailing standard for APIs and open banking solutions more generally, BIAN has found it necessary to publish its service operations in a REST format (pseudo code) that we call **Semantic** APIs, in recognition that they are really more requirements specifications than complete implementation level designs as mentioned already. But hopefully the semantic APIs are sufficient to define unambiguous service exchanges that can be mapped to end points consistently (even if they are defined at a higher that ideal level in many cases).

REST is really a synchronous protocol (though I know an asynch version is now being promoted). Though BIAN is implementation agnostic, we do anticipate that all service exchanges between Service Domains should be loose coupled and asynchronous amongst other things. So yes BIAN Service Operations can map to REST endpoints, but this works for the definition of end user applications, and it gets a little more tricky when the same service operations are mapped to application to application interfaces where REST may not be the optimum architectural style. Fortunately I do all my programming in Powerpoint, so I don't have to resolve the sync/async debate myself!

**Q15:** Does BIAN have some Open Banking case studies?
**A15:** We will be developing these specifically as an aspect of the upcoming Coreless 3.0 initiative – with likely focus on the UK Open Banking (OBIE) PSD2 standards.

**BankERRM**™

Risk — Return

**Enterprise Risk Adjusted Return Management**

**Event, Service, Process & Data Flows**

**PBORM**™

Process Based Approach to managing Operations and Operational Risk Management

# Many Thanks BIAN
# Many Thanks Guy.

Kannan Subramanian R
kannansubramanianr@bankerrm.org
January 2023