



INTEGRATION ARCHITECTURE

GENERAL INFORMATION

Who Can Benefit?

The IASA Integration Architecture course is a three-day in-depth integration architecture discovery for:

- Practicing and aspiring integration architects.
- Senior Developers
- Senior technology resources that need to support integration architectures
- Senior Architecture Managers, and
- Technical Leaders
- Anyone that needs to understand integration architecture and work with architects

Course Objectives

By the end of the course, students should be able to:

1. Explain the role of integration architecture in meeting corporate business objectives.
2. Explain to different stakeholders the key decisions surrounding the adoption of integration architecture and how those decisions impact the business mission.
3. Explain key technology decisions that impact the choice of business, software and infrastructure/cloud implementation technology choices.
4. Develop an end to end integration architecture for a corporate environment that based on sound architectural principles, processes and patterns.
5. Communicate about integration architecture with the right level of information for each different stakeholder.
6. Have a better understanding of how to apply the IASA Global Information Technology Body of Knowledge to real situations and come away with an intuitive feel and comfort about integration architecture practices.

Background

IASA's program reflects how international organizations work, which is often more hierarchical and has a unique architectural approach. The certifications are therefore suitable for those who wish to obtain an international qualification.

The flexibility, quality, changeability, scalability and durability of a system depends on the underlying architecture. And good architecture is based on the correct foundation work. Lack of architecture work not only

results in costly and ineffective solutions but also in long-term issues that get stuck in the activities that the solutions will support.

This program leverages the underpinnings of good architecture practice as defined by the IASA Global ITaBoK (the free Information Technology Architecture Body of Knowledge). Often the missing step in many courses of this nature is practical workshops demonstrate integration:

- The architectural artifacts and techniques used to build an integration architecture,
- The software implications and choices that go into the adoption of modern integrations, including API development, microservices and other approaches.
- How to earn and justify a seat at the executive table through integration delivery and by preparation targeted to each senior stakeholder and executive. We also discuss how to deal with significant business decisions that impact the integration architecture.

Overview of Methods

The IASA Integration Architecture course is a 3-day exploration of modern integration techniques. In today's technology landscape of hybrid cloud, IoT, API/Gateways, service mesh, serverless, microservices and the ever more complicated set of legacy systems and packaged SaaS solutions, integration could not be more important. However, due to the complexity of these systems, we must grow and new kind of integration architecture, which can deal with this complexity dependably and rationally. This course covers the primary integration patterns and methods in modern architectures including hands-on components such as cloud, microservices with traditional integration methods. Also, the course covers information structure and dependencies and techniques with a focus on shared dependencies in transactional and non-transactional formation across information entities.

In developing a real-world integration architecture, several key concerns include:

- Business objectives and value delivery are essential regardless of the technology choices
- Software and information concerns drive out the integration "application" structure and design, while
- The implementation will drive out infrastructure concerns, including implementation vehicles such as containerization versus serverless delivery.

This course provides an end to end understanding of these architectural aspects to leave the student with a solid understanding of the key issues.

Prerequisites

There are no formal prerequisites; however, an understanding of architecture patterns and practices, as presented by the ITaBoK, are valuable. The detailed software delivery and tools use eclipse integrated development with Java.

Also, the core foundations course material provides an important background (but is not essential if you are familiar with the ITaBoK)

Course Reading

The following texts build the foundation of this course:

[Enterprise Integration Patterns](#) by Gregor Hohpe and Bobby Wolf

[Information Technology Architecture Body of Knowledge](#) and core principles presented on the <https://iasaglobal.org/> website, by IASA Global

Other texts that serve as good course references include:

[A seat at the Table](#) by Mark Schwartz describes how agile business approaches, integration and the table work together.

[Cloud Design Patterns](#) by Alex Homer et al. describes some of the practical cloud design patterns (that cloud and hybrid base integration build on).

Continuous API Management by Mehdi Medjaoui et al. drives down into some of the key agile architecture issues and API architecture and management.

Integrating Serverless Architecture by Rami Vemula provides a great overview of serverless integration using Azure functions.

Expectations and Goals

IASA maintains and developed the integration architecture course and certification to:

- Develop and maintain a body of knowledge on the best practices in integration
- Provide tools for evaluating techniques, tools and technologies in the integration space
- Guide technical teams implementing both coarse-grained and fine-grained integrated systems
- Ensure that the right priorities are used to make architectural choices that balance costs against utility values
- Optimize project plans to achieve the right balance between resource consumption and project time

Course Delivery

The course is delivered both in-person and in an online instructor-led format with printed or online materials and practical workshops. The test for the certification is online after completion of the course. The course and certification are in English with English course material.

Certification

The fee for Integration Architecture certification is USD 600, and it is not included in the course fee. Course participants that want to certify need to state this intent at the beginning of the course.

INSTRUCTOR

About the Author

Brice is an acknowledged integration architecture thought leader, architecture practice leader and CITA-P. He has worked for companies such as IBM, Microsoft, and private industry to build effective and practical integration architecture practices. Many of the challenges of integration architecture stem from the diverse sets of architectural skills that are required to deliver an effective architecture that delivers on business value.

Brice has developed this course to represent and explore the integration architecture landscape. It has evolved along with the practical knowledge of how people, technology and architecture can be applied. At the same time, the integration principles, practices and processes are described in this course and use to present this material according to a solid theoretical foundation. The author used his direct experience with integration to identify and address the most common integration issues and problems.

One of the most interesting lessons is that only with the full participation of key stakeholders will the integration solution meet everyone's needs. From technology to business decisions and the varied array of stakeholders, Brice has developed a course that not only captures the technical details of integration

architecture, and it also describes scarce lessons in relationship stakeholder management that are often fundamental to the business success behind corporate integrations.

Contact

Brice Ominski is an IASA Global Principal Architect.

Email: Brice.Ominski@iasaoffice.org

I am always interested in IT architecture dialogue, especially as it relates to IASA Global and the integration of Architecture course work. Please feel free to forward me your comments and questions about this course.

COURSE MATERIALS

Integration Architecture

The Integration architecture course is the representation of the integration architecture best practices housed in the ITABoK Integration Topic Area and includes:

Architecture Practice, Management, Monitoring – From understanding architecturally significant requirements to management and monitoring of integration architecture, the course will address the processes, products and techniques for integration across multiple projects or products in the enterprise.

Previous and Theoretical Integration Architecture: EAI, ESB, SOA – As a primary method of integration, the course will review previous patterns, tools and techniques used to integrate systems.

Integration Implementation: On-premise, cloud and hybrid – cloud technologies, including SaaS and PaaS, have complicated integration architecture significantly and have created hybrid cloud-based integration strategies and Cloud integration patterns including data and services, will be covered in the course. Integration implementation choices need to be addressed and include functional /serverless vs. containerization for microservice implementation and API management tools. There is a broad range of tool kits, frameworks and management suites addressed in this course.

Information Integration – Information architecture is a separate course in the IASA catalogue, but this course addresses areas of concern, non-relational and data.

Information Integration – Information architecture is a separate course in the IASA catalogue, but this course addresses critical integration aspects of relational, non-relational and data dependencies.

LEARNING TOOLBOX

Exam Information

The core and foundations material found in the ITaBoK along with the suggested reading form preparatory course material.

How to Prepare for the Workshops

The class workbook is available in advance. Students should work through the software set up examples in advance. Also, a review of how the canvases and other artifacts are applied can be found in the ITaBoK and also available through IASA Global learning shots. Their student should also be familiar with the Tinkleman Corp. case studies before the class.

Certification

The Certified Integration Architect credential is awarded to those who qualify based on a combination of criteria including education, experience and test-based examination of professional knowledge of architectural skills and management.

The Certified Integration Architect credential is awarded by achieving a 70% or higher on the Certified Integration Architect examination. The exam consists of 75 multiple-choice/true/false questions.

The Foundation exam is available online, anytime, via IASA's Learning Management System. If attending an onsite course, the exam is proctored on the last day. If attending an online course, access is given on the last day of the course as well. Students will be given 2.5 hours each to complete the exam.

DETAILED SYLLABUS

Module 1 – Introduction

Module 1 provides an introduction, overview and context for the course and the other modules. The module begins by presenting background information that is relevant to understand how modern integration has evolved from a waterfall monolithic single function integration world to the current focus on business enablement and digital transformation. Business value drives the need for increased capability, whereas architecture strategies and provide options about the best ways to achieve that increasing capability. After a review of some fundamental concepts, we get right into the underlying architecture integration principles. We provide a brief overview of how each module sets the stage to meet the course objectives. Finally, the following lessons describe the theoretical underpinnings of integration architecture:

Integration Elements Overview describes the integration and the key architectural elements that create business value through integration.

Introduction to Integration Architecture Principles establishes the core principles to organize and understand the integration architecture pattern taxonomy. These patterns have extended the architecture fundamentals and evolved to support the current digital eco-system and deal with increasing integration complexity.

Integration Business Drivers helps provide an overview of integration in terms of business enablement, business values, and business tools. This section helps sets the context around executive integration discussions.

An Evolution describes the increasing complexity that integration patterns have evolved to solve. APIs, Microservices and other messaging patterns have evolved to deal with increasing complexity. This

evolution is by no means over as integration that is the very heart of digital transformation today will become the basis for a whole new foundation for distributed computing and application development.

Module 2 – Practical integration

Module 2 focuses on modern integration technologies that are the heart of digital transformation. Despite the increasing focus on technologies like IoT, Artificial intelligence, and others, all these technologies are only useful in ways that they can be combined and work together. The integration enables architectures where all these components can work synergistically to deliver old business capabilities better and new innovative capabilities. Then it's about how products wrap these capabilities to provide value. The following lessons describe the theoretical underpinnings of integration architecture:

Integration for Digital Agility describes the modern principles stemming from agility and how they impact integration. Change and rapid change are inevitable. The pace of change is increasing. This section introduces key in technologies to support change, including APIs, API gateways, cloud adoption, and how to connect all those technologies, including those that have not yet been invented.

API as a Product AaaP describes the development of these high technology-driven products. Products must provide business functionality, be delivered internally / externally to provide value and show a real ROI in some way. In this section, we show how a traditional tool, like the business model canvas, can be adapted to illustrate the key factors in API product development. It also describes the key business models that have been enabled through API deployment to help provide a taxonomic context. It is in this module that we begin to exam Tinkleman's online and the difficulties their company is facing. In the end, its about understanding customer adoption of your API as a product.

Effective [API] Decisions Making describes approaches to good API governance. These governance approaches apply to any organization to effectively manage API as a product. The distinction between centralized and decentralized decision making is key driving in understanding how to tailor your organization's API decision processes. This section also discussed known traps and pitfalls.

API Concepts shifts our thinking to focus more on the practical software architecture aspects that we need to understand to build real APIs. These concepts include strategy, design/documentation, development/testing, deployment/monitoring, security, discovery and change management. In this section, we discuss how waterfall vs. agile and modern dev-ops concepts are applied. This section's workshop looks at defining API using tools like OpenAPI (Swagger). This section also describes why decoupling the API interface from its implementation is so important, along with other elements like API security.

API Lifecycle describes the lifecycle: the creation of specific deliverables, who is responsible, and how the team can work together to develop complex API based products. Issues such as version control take a lead role in this section. As well, it's important to understand API deployment in the greater business context and how we treat our customers.

API Integration Assets recognizes the importance of managing API endpoints as business assets to be managed as such. API catalogues, service technology and the chaotic impact of unmanaged APIs are all managed. As API management is ever more complicated, we see the emergence of service fabrics.

API Landscapes recognizes that APIs don't come one at a time; APIs are typically managed as groups and by API management tools. This section examines how to manage APIs operationally and as key contributors to revenue by focusing on the underlying infrastructure needed to support these revenue-producing APIs. The choice in API management systems impacts how these APIs are delivered.

Module 3 – Modern integration: Microservices

This section describes the microservices and service orientation perspective. These services can stand alone to provide value or supply services to the APIs that we have discussed in the previous module.

Microservices 101 describes the fundamental architectural principles that support building good microservice-based architectures. With microservices come a greater expectation of customer service delivery, and these architectures need to manage these expectations fully. These approaches have subsumed the development of monoliths and even ESB based architectures. However, the reality is that we live in a world where these styles often need to co-exist and the course will detail some of the issues. For example, how do microservices fit into a transactional world is a topic that is still highly relevant when microservices meet systems of record.

Service Fabrics: Microservice Platforms, Containers and the Cloud looks at fabrics as the next evolutionary step in microservice deployment. Microservices in fabric transition thinking about single messages to dealing with message swarms and the emergent properties of those swarms.

Microservice Patterns / in-depth revisits the technical aspects of building microservices. In this section, we revisit the patterns in more detail and describe the essential features in software. The workshop introduces messaging and how to implement a microservice pattern. The section starts by understanding messaging and the event-driven request /reply pattern and then builds from there. The workshop describes ways to architect and to implement these messaging patterns at a fundamental level by using Camel Apache. Of course, we look at microservice and transactional connect as well as event-driven architecture using Kafka.

API and Microservice Governance extends the governance fundamentals to deal with both APIs and microservices. Who owns what, and how we make changes are important and often introduce the notion of a center of excellence to deal with these issues until integration becomes part of the mainstream fabric of the organization.

Module 4 – Messaging

In this module, we revisit the fundamentals of messaging and look in-depth at how to use messaging tools such as Apache Camel and pre-existing messaging libraries. After this module, the student will understand

how to establish a messaging system and its endpoints based on the Apache camel messaging library that is, in turn, based on Gregor Hohpe's messaging patterns.

Frameworks, Tools and Suites in Depth continue to evaluate the continuum of implementation options based on the level of developer support. This section focuses on the types of vendor offerings that will enable you to build your messaging / integration offering. Here we also discuss the practical adoption choices as we survey the main vendor-provided tools.

Messaging Style is one of the key elements identified in Hohpe's messaging book. While integration evolves, a deep understanding of messaging patterns provides the theoretical basis to understand these evolutions at a fundamental level.

Popular Integration messaging Protocols examines the technology of what makes messaging work across some of the most popular messaging protocols. This section also examines the popular MS message queue and fundamental properties that have business concerns (durability, reliability, transaction support etc.)

Popular Integration Messaging Protocols continue the detailed assessment by looking at some of the popular integration messaging protocols, including JMS, HTTPs, SQS / SNS (Amazon), MQTT etc.

Messaging Design Taxonomy looks at key message patterns and approaches. This section elaborates on the pipes and filters architecture, message routing, message translation, system management considerations and testing patterns. It provides an idea of the messaging design and how thorough the taxonomy of pattern has become. This section examines message pattern selection based on quality attributes and some of the considerations that are involved. The workshop in this section examples architecture differences between RPC and REST styles messaging approaches.

The Message provides an in-depth discussion around the message contents itself. Messages transmit discrete units of data between a sender and receiver. Marshalling these packets ensures that the receiver gets and understands and reconstructs the message stream in the intended order.

Pipes Filters Revisits describes this architectural approach and the types of problems that are suited. These include more real-time problems than your typical digital enterprise integration and this section is more suited to enabling IoT, streaming or high-performance messaging architectures.

Message Router and "hub and spoke" type architectures are a different style to pipes n filters and provide more a basis for many of the enterprise / digital integration patterns. This section describes some of the important distinctions between choosing pipes and filters versus message router style messaging architectures.

More Advanced Messaging Patterns and Quality Tradeoffs in Messaging explores some of the additional more complex patterns and their characteristics: channels, construction, routing, transformation and additional system management patterns.

Module 5 – the Integration Taxonomy

This module provides an overview of the integration taxonomy and how we look and evaluate different integration architectures as a business, software, infrastructure and information systems.

Integration Patterns / Anti-patterns explores common aspects of what to do and what not to do in an integration architecture. This section also describes how the social complexity of the business problem drives out many of the integration characteristics. Here we explore an array of styles from point to point integration, synchronous vs asynchronous properties etc. We also examine some of the legacy architecture approaches, like file transfer.

About Architecture Method discusses the adoption of integration patterns for various types of business operating models that describe the enterprise architecture. Unification, Replication and Diversification each have characteristics that tend to suggest certain integration patterns.

Integration Style provides a delineation of patterns based on intent, connection approach and interaction mechanism. In this section, we also summarize implications for the four layers of the enterprise architecture: the business perspective, the information perspective, the application perspective and the technology perspective. Integrations necessarily address each of these perspectives.

The Intent Taxonomy delves into the types of intent and classification of integration intents. I. Intent maps the solution to the social complexity of the integration problem.

Connection examines the need to transfer information between endpoints and ultimately between systems and people. The functional layer can transform data, and the connection provides one place where we can make this transformation as well as incorporate error checking.

Interaction mechanics look at the dynamics of messaging in more detail. Here we need to decide between push/pull and synchronous versus asynchronous patterns. This section is more advanced discussed across several patterns.

Module 6 – Information Integration

This module discusses how to align information across systems and through integration by looking at the information layer in the enterprise model. This area is quite complex in general, and this course only addresses some of the issues, such as data transformation for different uses.

Information Sharing Challenges touches on some of the challenges that are more information oriented. Here we discuss applicable canonical data patterns and ETL/ ELT tools. We also discuss different types of data formats that are in common use and the implications for integration architecture. Database replication is also discussed as one possible pattern to share information that abstracts some of the other layers.

Workshop Approach

There are three categories of workshops: general discussion, architecture and coding examples. The general discussion includes areas in the module that may apply to specific situations and how we can take that module material and adapt it to the situation. The coding workshops examine how to set up API code, and messaging code using Java in an Eclipse development environment. Finally, the architecture example follows Tinkleman Online. We examine the solution end to end for one problem and then run through groups with a different area. As we identify requirements and how to deal with quality attributes to meet those requirements, we see how the architecture views and canvases help us to make architectural decisions and chose an appropriate architecture that meets the problem's quality attributes.

The result is the student has walked through messaging creation, discussed how to make architecture real in his/her situation and finally, the steps that are needed to make key architectural decisions in building out a real-life architecture.