IKASAN

Open Source Enterprise Integration

Ikasan ESB Reference Architecture Review

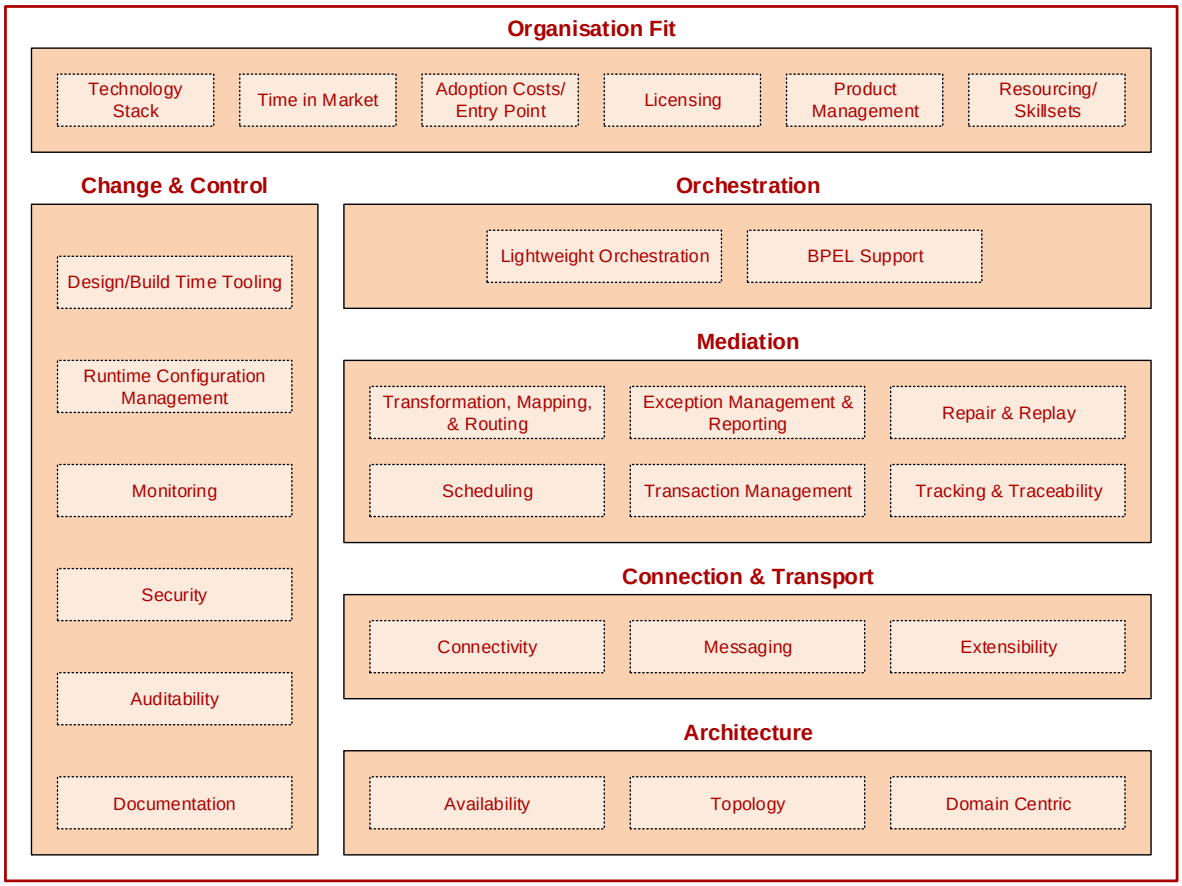
EXECUTIVE SUMMARY

This paper reviews the Ikasan Enterprise Integration Platform within the construct of a typical ESB Reference Architecture model showing Ikasan's coverage within the ESB integration space.

Ikasan has strength in the overall architecture, connectivity, and mediation areas of functionality which is a product of its growth and maturity within the vertical financial sector. Orchestration is only lightly covered by Ikasan in its current offering; however, it does lend itself to interoperability with other players in this area allowing this gap to be closed. Overall, Ikasan provides a robust ESB platform with strong architectural and mediation offerings. Where areas of immaturity exist there is opportunity for Ikasan to grow and fill this space.

ESB REFERENCE ARCHITECTURE MODEL

A typical ESB Reference Architecture Model is shown below grouped into functional areas within which categories of operation are covered.



IKASAN ESB FUNCTIONAL COVERAGE SUMMARY

This table summarises the functional areas with detailed splits of the categories allowing these to be scored based on the following score card and weighting criteria.

Score Card	
-2	Unacceptable
-1	Poor
0	Little to no offering, but acceptable
1	Good, positive direction
2	Almost perfect

Weighting	
5	Very Important
4	
3	
2	
1	Doesn't matter

Functional Area / Category	Weighting	Score	Weighted Score
Architecture	-	-	-
Availability	-	-	-
Fault Tolerance	*5*	*2*	*10*
Scalability (Horizontal)	*5*	*2*	*10*
Scalability (Vertical)	*4*	*2*	*8*
Resource Footprint	*4*	*1*	*4*
Throughput	*5*	*2*	*10*
Service Federation	*1*	*0*	*0*
Topology	-	-	-
Intra-business Entity	*5*	*2*	*10*
Inter-business Entity	*5*	*1*	*5*
Domain Centric	-	-	-
Canonical Model Support	*5*	*2*	*10*
Sub-total	**-**	**14**	**67**
Connection & Transport	-	-	-
Connectivity	-	-	-
Communications Protocols	*4*	*2*	*8*
Off-the-shelf Connectivity	*3*	*1*	*3*
Messaging	-	-	-
Messaging Standards	*4*	*2*	*8*
Extensibility	-	-	-
Supporting New Connectivity	*5*	*2*	*10*
Sub-total	**-**	**7**	**29**
Mediation	-	-	-
Transformation, Mapping, & Routing	-	-	-
Enrichment	*5*	*2*	*10*
Conversion	*5*	*2*	*10*
Mapping	*5*	*2*	*10*
Routing	*5*	*1*	*5*
Sequencing	*5*	*2*	*10*
Filtering	*5*	*2*	*10*

Splitting	*5*	*2*	*10*
Aggregation	*5*	*2*	*10*
Broker	*5*	*2*	*10*
Scheduling	5	1	5
Exception Management & Reporting	5	1	5
Repair & Replay	4	1	4
Transaction Management	5	2	10
Tracking & Traceability	5	1	5
Sub-total		***23***	***114***
Orchestration	-	-	-
Lightweight Orchestration	3	1	3
BPEL Support	3	0	0
Sub-total		***1***	***3***
Change & Control	-	-	-
Design/Build Time Tooling	-	-	-
Standards Adoption	*3*	*2*	*6*
Developer Lead Time Productivity	*4*	*1*	*4*
SDLC (IDE's & Scaffolding)	*4*	*2*	*8*
Runtime Configuration Management	5	2	10
Monitoring	-	-	-
Technical Monitoring	*5*	*2*	*10*
Business Monitoring	*4*	*1*	*4*
Security	-		
Authentication	*5*	*2*	*10*
Authorisation	*5*	*1*	*5*
Encryption	*3*	*1*	*3*
Auditability	4	1	4
Documentation	3	-1	-3
Sub-total		**14**	**61**
Organisation Fit	-	-	-
Technology Stack	3	2	6
Time in Market	4	1	4
Adoption Costs/Entry Point	5	2	10
Licensing	5	2	10
Product Management	-	-	-
Features & Fixes	3	*2*	*6*
Releases	*4*	*1*	*4*
Support	*5*	*1*	*5*
Resourcing/Skillsets	5	2	10
Sub-total		**14**	**55**

ARCHITECTURE

Architecture refers to how well the platform has been architected for fault tolerance, scalability, throughput, and service federation. Key to this are the topologies the platform can fit and how well it focuses on the business domain it services.

Availability

Fault Tolerance

Ikasan has the ability to auto-recover in the event of technical failures. In addition to this the Integration Modules are able to hook into the Application Server runtime container high availability (HA) features to leverage a container's HA clustering failover facility.

Scalability

ESB scalability can have a natural throttle enforced by the requirement of maintaining the end to end order of messages. This can limit aspects of parallel processing; however, Ikasan makes the most of scalability options available by allowing parallel (multi-threaded) processing to be configurable operations within flows.

Flows themselves can be logically grouped as performance requirements dictate and deployed across a number of Application Server runtime containers. The Application Server runtime containers can be scaled out both horizontally (in number) and vertically (on host specification).

Resource Footprint

The resource footprint of any Ikasan integration solution is completely configurable to ensure that services that are not of concern to that solution are not packaged and deployed with that solution. An Ikasan integration solution can be stripped down to a bare bones deployment as requirements dictate.

Throughput

Ikasan's core engine has currently been benchmarked at **3 million messages per second** (single-threaded, message order maintained, run on an 8GB 4 core machine) within the scope of a simple integration solution based on an object consumer, converter, router, and publisher.

Service Federation

Ikasan does not currently support automatic service federation. This feature is being reviewed as a potential planned roadmap item.

Topology

Ikasan is agnostic to the topology adopted within an organization. There are recommended topologies which promote performance, visibility, and supportability, however, these are not dictated by Ikasan as an Enterprise Integration Platform.

Domain Centric

Ikasan lends itself to a business domain centric approach to ensure the integration "*mess*" observed with point to point connectivity outside of an ESB solution are not simply taken and replicated within the ESB.

CONNECTION & TRANSPORT

Connection and transport cover how well a platform supports messaging standards, wire protocols, vendor API offerings, and how easily an unsupported protocol can be added.

Connectivity

Ikasan isolates the connectivity protocols of source and target systems within the scope of consumer and producer abstractions. This allows for swapping of protocols with minimal impact to the flow supporting that consumer/producer, and also provisions a very flexible model for supporting any kind of protocol. A number of protocol connectors have been developed based on the Java Connector Architecture (JCA) standard to allow maximum portability and extensions for hooks into container managed services such as Transaction Management and Life cycle management.

Messaging

Ikasan is agnostic to the adopted messaging standards and implementations being modelled. It will simply support the messaging standard adopted by the organization.

The architecture of the ESB underpinned by Ikasan promotes the concept of a well-defined business domain model defining industry domain message standards and extensions therein as appropriate.

Any third-party transformation tools can additionally be used within an Ikasan runtime component for complex transformations based on industry domain standard definitions.

Extensibility

Ikasan's model allows for any connector protocol to be developed as a simple API program or in-line with the standard Java Connector Architecture (JCA) specification to leverage Transaction Management and other Application Server container services.

Third-party JCA connectors or API libraries can be used within an Ikasan component.

MEDIATION

Transformation, Mapping, & Routing

Ikasan supports the Enterprise Integration Patterns as standard contract based implementations. A number of implementations are available off-the-shelf, whilst creating custom implementations to suit a specific requirement is quite trivial. Importantly Ikasan does not restrict or overlay framework wrappers around the message – the message is free from Ikasan constructs and anything can be transported.

Ikasan currently supports the following Enterprise Integration Patterns as standard contracts.

Enrichment

Ikasan provides a dedicated contract for the transformation of any message specifically without mutating the incoming type.

Conversion

Ikasan provides a dedicated contract for the conversion of any message type into any other type.

Mapping
Ikasan provides mapping services in a number of forms. Coded mapping can be accomplished with static lookup files, or via some other persistent store. A Mapping Configuration Service feature within Ikasan exposed through the Dashboard also allows runtime mapping changes to be made dynamically.

Routing
Ikasan provides a dedicated contract for single recipient routers (exclusive OR routing) as well as and multi-recipient routers (AND routing).

Sequencing
Ikasan provides a dedicated contract for (re)-sequencing of messages.

Filtering
Ikasan provides a dedicated contract for the filtering of messages backed by an optional persistent store.

Splitting
Ikasan provides a dedicated contract for the splitting of messages.

Aggregation
Ikasan provides a dedicated contract for the aggregation of messages. This can be optionally backed with any type of in-memory or persistent store.

Broker
Ikasan provides a dedicated contract for brokering of messages in support of operations such as request/reply or the invocation of an external system within the scope of a flow.

Scheduling

Ikasan has dedicated schedulers for managing both simple relative schedules as well as cron expression based complex schedules.

Exception Management & Reporting

Ikasan supports a configurable exception management service in support of reporting technical or business message failures. A flow can be configured to report the error and either auto-recover and retry; exclude the message for potential repair and resubmission; or to simply halt and alert operations staff.

Repair & Replay

Ikasan's Dashboard provides an optional message repair (subject to authentication/authorization) and replay service for any failed messages within a flow.

Transaction Management

Ikasan binds to the underlying Transaction Manager provided by the Application Server runtime container. Ikasan utilizes non-transactional, local-transaction, and distributed-transaction semantics as supported through the transaction manager to allow coordination of resources for robust and guaranteed message delivery.

Tracking & Traceability

Ikasan ensures that every message initially received is stamped with an immutable message life identifier to allow tracking of this message regardless of mutations throughout a business stream. The Ikasan dashboard provides search facilities to allow any message transported via Ikasan to be traced throughout its life.

ORCHESTRATION

Orchestration refers to the higher level process/workflow used to orchestrate business events within the domain.

Lightweight Orchestration

Ikasan supports the construct of business streams which provision the coordination of any number of integration modules to make up a logical end to end work flow. This aspect is being further built out as part of the dashboard to provide a more intuitive business domain view around user workflows.

BPEL Support

Ikasan does not provide a BPEL solution. Other provider BPEL solutions can be utilized within Ikasan runtime to support more complex work flows.

CHANGE & CONTROL

Change and control refers to a number of concerns that cross cut through the platform as well as build and runtime management of the platform.

Design/Build Time Tooling

Ikasan is based on Java and XML, therefore, has immediate support within the popular IDEs. Git, Maven, and Nexus underpin Ikasan's source control; build and dependency management; and binary repository, respectively.

Ikasan development is currently undertaken directly in XML and Java. There is no GUI for creating integration solutions; however, this is something currently under review based on the Eclipse Graphical Editing Framework (GEF).

Runtime Configuration Management

Ikasan is deployed as a standard JEE artefact within an Application Server runtime container.

Ikasan Consoles expose a number of management aspects within Integration Modules all the way down to configuring the runtime operation of flows and components therein. Flow operations can be started, paused, and stopped dynamically without impacting the rest of the application (this is equivalent to stopping/starting a single thread in the JVM). This provides for very dynamic runtime management of integration modules and the business streams they support.

Monitoring

Technical Monitoring

Ikasan has default monitors which push any change in the runtime status of a component to any registered technical monitor platforms. This ensures real-time notification of any change in runtime state per flow.

Business Monitoring

Ikasan has some basic business monitoring and statistics allowing breaches in SLAs to be notified to any business monitor consumer. NOTE: This feature is in beta. Alternatively, third-party business monitors can be plugged in to report and alert on these statistics.

Security

Ikasan takes a standard approach to authentication and authorization of principals within roles to assigned security policies. This can be managed locally, via LDAP, or through a combination of both.

All sensitive data being transported over Ikasan can be encrypted over the wire based on the selected encryption policy of the organization.

Auditability

Any change in runtime state or configuration is recorded within the Ikasan platform. A full record of who changed what and when is available through the Ikasan Console.

Documentation

Ikasan currently has Console User and Administration Guides; and a Getting Started Guide for developers. Ikasan is currently working on providing a full developer guide for the Ikasan platform.

ORGANISATION FIT

The organization fit is very subjective as each organization differs. The following categories have been highlighted as common concerns within organizations when adopting solutions.

Technology Stack

Ikasan is developed in Java and therefore will run on any platform which can support a Java Runtime Environment. It typically runs within an Application Server runtime container and has proven support against both JBoss and WebLogic. Ikasan never deviates from standard specifications, so it remains portable to any other Application Server.
Ikasan uses a SQL database for its configuration persistence. Any database the Application Server can support can be used for this.

Time in Market

Ikasan has been running and supporting investment bank business since 2005. It has undergone many facets of change in keeping up with latest trends in technologies and methodology approaches to where it is now – a robust and proven platform supporting front, middle, and back office integration requirements within the financial vertical market.

Adoption Costs/Entry Point

Ikasan is very flexible and does not require a big bang approach to adoption. Single connectors from Ikasan can be used in alternate ESB stacks, however, to get the real power of Ikasan it is recommended that at least the platform is deployed with a starting integration module solution. As an Open Source project Ikasan has no license fees or maintenance/support costs – you are free to adopt the IkasanEIP Open Source Enterprise Integration Platform as needs fit.

Licensing

Ikasan is distributed under the Open Source Modified BSD license.

Product Management

Features & Fixes

Ikasan is an Open Source product and as such any feature or fix changes to the platform can be undertaken by any developer wishing to contribute to the platform (with appropriate signed agreements and disclosures). Any feature or fix submission is managed and reviewed by one of core committers before being accepted and merged into the code base. All committers are long term members of the Ikasan project.

Releases

Ikasan releases have been on-demand in the past, although they are starting to review streamlining this in to releases every month with potentially ad-hoc feature/fix build releases for early adopters.

Support

Ikasan has a number of community support channels. Ikasan Jira is the most commonly used to raise queries, feature, requests or bugs. Alternatives to this include IRC or direct interaction with the development team, where appropriate.

Resourcing/Skillsets

Ikasan is based purely on Java and XML. It does not utilize custom domain specific languages (DSL) and therefore benefits from a wealth of standard market skillsets. It follows the Enterprise Integration Patterns closely. Once a developer is familiar with these patterns they become productive very quickly.