



Best Practices for Integrating Applications Development

By

Atanu Majumdar
VP Engineering

Kovair Software, Inc.
4699 Old Ironsides Drive,
Unit 190,
Santa Clara, CA 95054
Sales: 1.408.262.0200 Extn. 1
sales@kovair.com
www.kovair.com

Document Version History		
Release	Date	Reason
Version 1.0 (Initial Release)	30/11/2011	Initial Release

Introduction 4
Goals 4
Integration Approach 4
Technical Considerations 5
Functional Requirements 5
Non Functional Requirements 6
User Management 6
Project/Template Management 7
Adapters 7
Tool Versions 8
Deployment 9
Kovair Omnibus Integration Platform 9
Conclusion 11

Introduction

Enterprise application integration remains an age-old challenge. Emergence of new technologies for hosting infrastructure, platform and software across locations including on-premise, managed and cloud based data centers makes integration even more demanding and complicated. Traditional application integration approaches are not meeting the challenges of changing business needs and posing threats to the agility and flexibility of an organization. Application integration, if done properly with the right choice of technology and best practice considerations, can deliver immense strategic and technical value. How enterprises pursue application integration can make the difference between pitfalls and positive possibilities.

This paper discusses best practices to be adopted by the stakeholders while implementing an effective application integration solution within an organization.

Goals

Defining the goals of your integration project is a key step towards its successful implementation. Stakeholders need to understand:

- Business drivers for integrating applications
- Information under management
- Existing applications and services under management
- Core business processes and their dependencies
- Integration scenarios, such as Application-to-Application (A2A), Business-to-Business (B2B), and Cloud based applications (Software-as-a-Service – SaaS)

and then define

- High level objectives
- Requirements that address organization's actual near-term, midterm, and long-term integration needs
- Road map for the features and functions to be deployed

Integration Approach

Selecting the right integration technology is the next important step once you have set your goals. Choose an Integration Platform instead of multiple ad hoc point-to-point solutions that:

- Gives you one-stop solution for all your integration needs
- Provides fast and simple deployment options
- Requires limited IT Support
- Offers improved adaptability and agility
- Supports Functional Reusability
- Has independent Change Management system
- Delivers Interoperability – loosely coupled framework
- Helps to orchestrate rather than just integrate
- Meets the needs for enterprise class security
- Supports multiple versions for the same tool or application

- Can easily accommodate in the future, replacement of any application or addition of new applications to your application ecosystem in a plug and play mode

Select an Integration Technology that is:

- Based on Service Oriented Architecture (SOA)
- Built around Enterprise Service Bus (ESB) with a hub like architecture connecting applications through Adapters or Connectors
- Supported with Open APIs for Adapters – Organizations can thus build the adapter for their homegrown tools or contract them to third parties
- Extensible and configurable through drag-drop visual interface without coding
- Easily maintainable
- Capable of providing control or governance through business rules and workflows
- Cloud compatible - Cloud-to-Cloud, Cloud-to-On-premise and On-premise to On-premise
- Mobile ready

Technical Considerations

Before starting the implementation, Integration project stakeholders need to consider and analyze the specific technical requirements for:

- Semantic and Metadata Management
- Validation of data for correct structure and format
- Standard and Advanced Transformations
- Information Consumption, Processing and Delivery
- Intelligent Routing
- Connectivity and Adapter Management
- Governance across all applications

Functional Requirements

Project stakeholders need to identify:

- Applications or tools to be integrated
- Information/data required to be synchronized across tools
- Mapping of metadata between applications or tools
 - Avoid meaningless data replication if it does not address any specific business need
- Relationships between different information entities to be synchronized
- Integration rules to follow
 - Identify which tools and items need bi-directional data integration and which need one way integration
 - Identify the conditions for which data would flow from one tool to another
- Data latency requirements
 - For some tool it may be sufficient to synchronize the data at a daily frequency and for some other tool it may be required to synchronize every minute

Non Functional Requirements

The non-functional requirements to be considered include:

- Performance
 - Determine the data volume that would be transported on a regular basis – Define performance metrics upfront
 - Determine the permitted latency limits
 - Determine the growth of data over the years
 - Plan hardware configuration to support probable data volume up to at least two years
- Scalability
 - Understand the scalability options
 - Plan for load balanced servers based on your data flow load
- Security
 - Understand and document security requirements
 - Ensure compliance with organizational data security policies
 - Encrypted storage and transport of security credentials
- Reliability
 - Understand and document reliability measures needed
 - Check whether the integration framework will provide the required reliability measures
- Availability - Failover
 - Understand the availability and failover options
 - Plan for configuring these options based on your availability needs
- Migration
 - Define your data migration needs
 - Check with the vendor – how to implement data migration
- Monitoring
 - Failures and faults at the services/ adapters level
 - Failures and errors at the data synchronization level
 - Notifications/Reporting for failures/ errors
 - Integration Health Dashboard

User Management

User Management is done differently in different applications or tools. Some tools support LDAP based user management while others have their own user management functionalities. Application integration will require user synchronization between different tools. An Integration Platform should provide following options for User Synchronization:

- LDAP/ Active Directory based User Management support for all applications
- Automated User Mapping Policies between applications
- Manual User Mapping between applications
- Any central User Management system shared by all applications

Document your organizational User Management Policies and decide on the best options for your case. It could be a combination of the above options.

Project/Template Management

Organizations deal with hundreds of projects. Reusing the configuration setup done for integrating several applications across projects is key for the success of an Integration Platform deployment. An Integration Platform should support the following best practices for integration setup:

- Automated Integration Setup for large number of projects
- Integration Templates for different project types
 - Set of applications to be integrated
 - Metadata mapping between objects from different applications
 - Set of Integration Flows or Rules
- New projects inherited from the above templates based on the project type
- Templates administered by Process/ Tools Group
- Project specific Integration Setups are managed by Project Managers

Adapters

Stakeholders need to identify the integration use cases upfront and ensure that the adapters can support easy and effective implementation of those use cases.

- Identify the tools, 3rd party and homegrown, for which Integration Adapters would be needed
- For standard applications, discuss and define:
 - Common Use Cases - Examples:
 - Unidirectional and bi-directional synchronization
 - Support for different field types – simple text, rich text, number, date, single/multiple select lookup list, cascade select list, choice list and others
 - Support for field data type and size validation
 - Ability to detect new custom (user defined) field created in a tool and mapping to fields in other tools for data transfer
 - Tool specific Integration Use Cases – Examples:

Use Case	Tool	Event	Action
1.	IBM RequisitePro – Requirements Management Tool	Requirement is approved for implementation in a particular release.	
	HP Quality Center – Test Management Tool		RequisitePro Requirement is replicated in Quality Center as a Test Requirement.

Use Case	Tool	Event	Action
2.	HP Quality Center	Test Cases created by Tester in Quality Center to test the functionality provided by the above Requirement. Test Cases are executed and Test Results are recorded in Quality Center.	
	IBM RequisitePro		Test Results for the Requirement are replicated in RequisitePro.

- Edge Case Scenarios – Examples:
 - Default behavior in the target tool when mandatory field value not supplied from the source tool
 - Default behavior in the target tool when lookup list value does not exist or is not mapped
- For homegrown applications, discuss and document:
 - Integration Use Cases
 - API availability for the homegrown tool
 - Data Storage for the homegrown tool – Database or File System based
 - Adapter development and testing options. Integration Platform should provide the option for building the adapter in case API is not available for the homegrown tool.
- Establish a Deployment Road Map for different Adapters.

Tool Versions

Often, organizations need to deal with several versions of the same tool. In addition, as the tool vendors come up with newer versions, upgrade is done. Integration Platform may need to provide upgrades for the newer versions of the tools. Managing and planning for these upgrades is essential to maintain the integrity and stability of your integrated tools ecosystem.

- Document a tool-wise list of different versions that need to be supported.
- Ensure that proper planning is done to bring the support for newer versions of tools and adapters are ready on time for deployment.
- Newer versions of Adapters may expose new objects from the tools. Planning is needed to use those features in your integration scenarios.
- Standardize on tool versions used across the organization
 - Supporting numerous versions of the same tool across the organization should be minimized as much as possible because it increases the complexity of the integration scenario and makes it difficult to maintain.

Deployment

Follow the standard best practices for deployment of an integration platform.

- Staging environment
 - Set up Development environment
 - Set up Test environment
- Testing
 - Configure Test Cases to verify your Integration Use Cases
 - Execute Tests on Development and Test servers
 - Perform load testing
 - Perform security testing
 - Perform reliability testing – check the scenario when tools are down or offline
 - Perform availability and failover testing
- Production
 - After successful testing, deploy the integration platform and adapters on Production servers.
 - Ensure failure/exception alerts are sent to right people.
 - Setup regular backup of the relevant integration components.

Kovair Omnibus Integration Platform

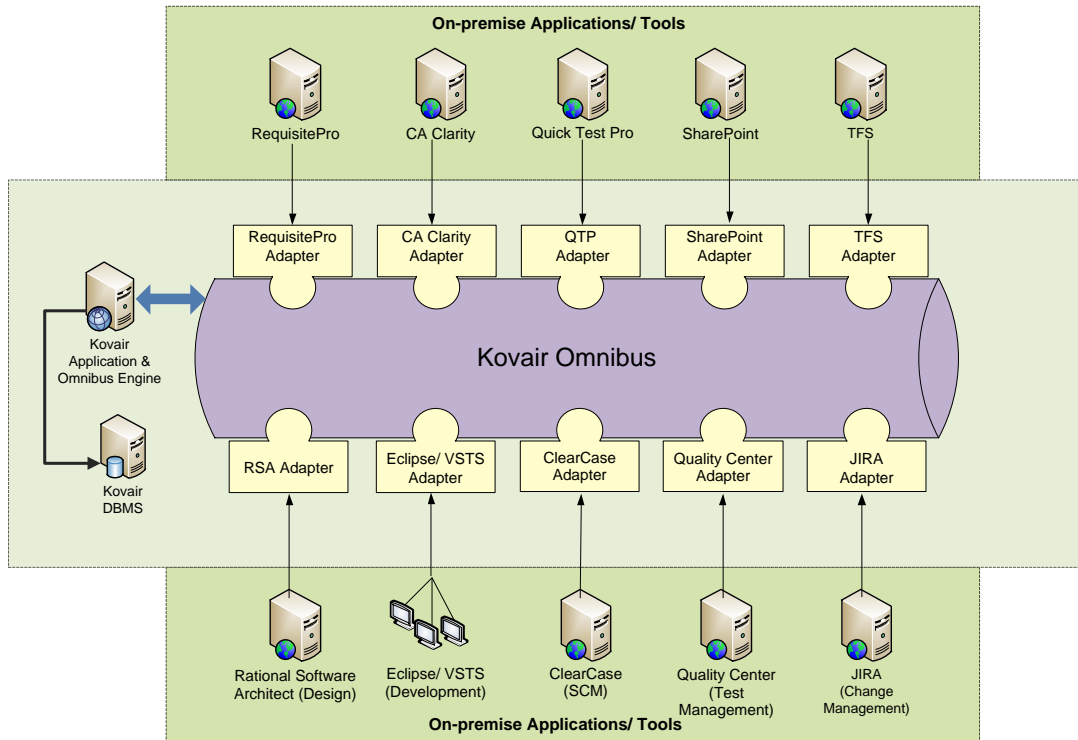
Kovair Omnibus is the leading Enterprise Service Bus (ESB) integration technology available for Application Development and IT tools. With standardized SOA based tools specific Omnibus adapters, one can create one’s own tools ecosystem. These tools can be from any vendor or any legacy data or any custom homegrown application development and IT tools.

Kovair’s Omnibus integration technology has major advantages that are not typically found in the vendor specific point tools or even a suite of tools.

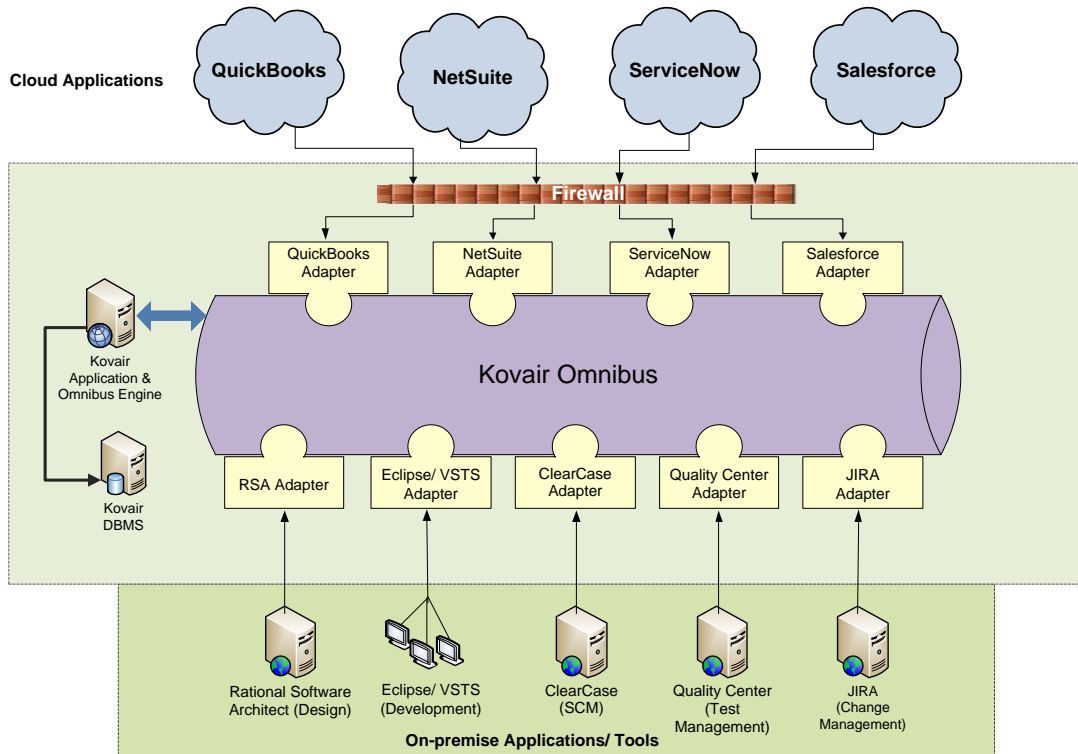
The following table lists the advantages of Kovair’s Omnibus integration technology:

Bus or Hub versus point to point integrations bringing considerable savings in development and maintenance costs	Leverages existing tools investment for customers – no need to rip and replace tools
Supports multi-directional Synchronization and Federation	Synchronization of data and relations between lifecycle artifacts in different tools
Data validation and transformation	Conflict Detection and Mediation
Customizable Integration Rules & Data Mapping independent of Adapters – No rule hardcoded in the Adapter	Supports integrations of multiple tools for the same functional area
Augment Integrated Tools’ capabilities for Process & Traceability	Connects tools behind firewalls through SOA – needed for globally distributed team
Reusable Integration Templates for rapid configuration – time-to-value measured in weeks	Compliance with organizational security requirements

Omnibus Platform – Integrating On-premise Applications/ Tools



Omnibus Platform – Integrating Both On-premise and Cloud Applications



Conclusion

Organizations are continually challenged with finding out ways to leverage more ROI out of existing infrastructure and applications. Having an effective Application Integration strategy is critical to taking on this daunting challenge. Choose the right integration technology and follow the best practices to establish an agile, reliable and scalable platform that is equipped to extend the life of your existing proven applications and ready to meet your future application development demands.