Open Source Enterprise Integration Platform

Jeff Mitchell
ikasan.org
July 2010

Introduction

- The Problem Domain
- Architecture Strategies
- Translating the Architecture
- Technology Stack
- Features
- Development Lifecycle
- Roadmap
- Summary

Problem Domain

- **[Domain]** Enterprise Application Integration (EAI) space
  - Integration of business systems in a landscape of interwoven and often complex finance processes
- **[Issue]** EAI is complex, costly, & not the primary business concern
  - Greater adoption of “Best of Breed” specialist applications
  - Business applications distributed across disparate platforms
  - Business data distributed across isolated silos
  - Legacy data repositories
  - Data duplication and integrity issues
  - No clear business data owner
  - Exponential integration requirements
  - Greater complexity of business demand on data orchestration
  - EAI is more than simply connecting applications
- **[Goal]** To provide simple, robust, configurable commoditised solutions
  - Expose business artifacts whilst isolating the integration specifics

Architecture Strategies

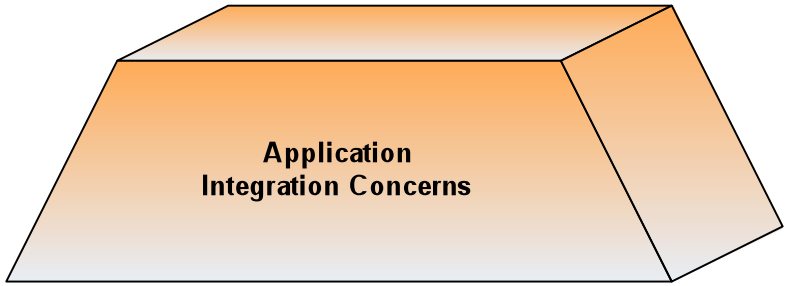
- Simplicity
  - Must not require in-depth proprietary knowledge
  - Avoidance of overly complex or heavyweight frameworks
  - Highly testable simple constructs
- Commonality
  - Reusable interchangeable constructs
  - Standard contracts of interaction
  - Simple repeatable implementation steps
- Adaptability
  - Ability to support any type of business entity
  - Ability to integrate any type of Enterprise Information System
  - Without breaking the first two strategies of Simplicity & Commonality
- Robust & Guaranteed operation
  - Maintain data integrity in business delivery and failure scenarios
  - Require minimal manual intervention i.e. failure / automated recovery

Architecture Strategies

- Clear contract definition and separation of concerns

Architecture Strategies

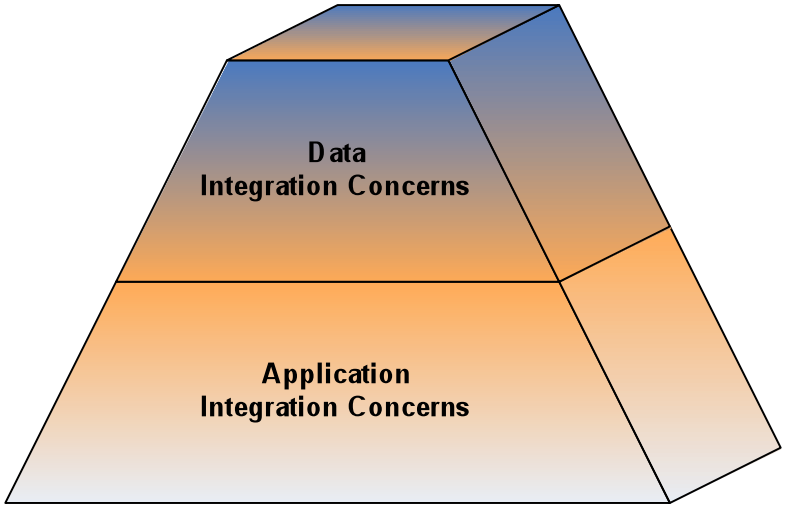
- Clear contract definition and separation of concerns
  - Application concerns



- **[Application]** performant, robust & guaranteed operation

Architecture Strategies

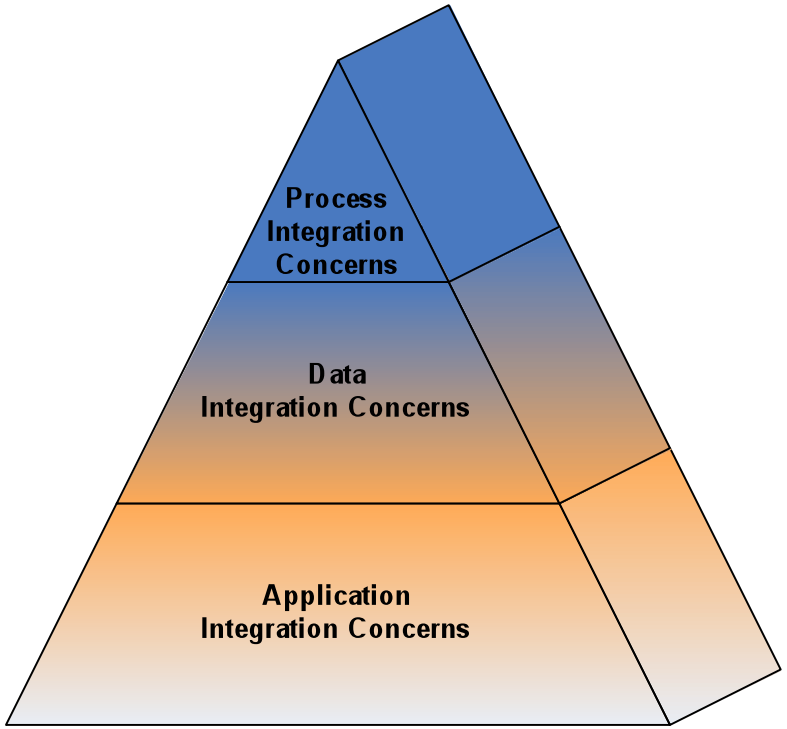
- Clear contract definition and separation of concerns
  - Application concerns
  - Data concerns



- **[Data]** presentation of standard, meaningful business data
- **[Application]** performant, robust & guaranteed operation

Architecture Strategies

- Clear contract definition and separation of concerns
  - Application concerns
  - Data concerns
  - Business process concerns



- **[Process]** business event & entity orchestration (STP is the goal)
- **[Data]** presentation of standard, meaningful business data
- **[Application]** performant, robust & guaranteed operation

Architecture Strategies

- Loose coupling
  - Integrations should only require knowledge of each other through exchange of business artifacts
- Tight cohesion
  - An integration's specifics must not bleed out unnecessarily
- High visibility
  - Business data tracking
  - Fail scenarios
  - Operational audit
- Single logical point of integration
  - Geographically agnostic / cross platform support
- Open standards aligned with proven design patterns
  - Adoption of industry standard Enterprise Design Patterns
  - Avoidance of vendor lock-in

Architecture Strategies

[Key Strategies]

▪ Single Point of Integration	▪ Loose Coupling	▪ Variety of Standard Contracts
▪ Separation of Concerns	▪ Tight Cohesion	▪ Standard Design Patterns
▪ Interchangeable Constructs	▪ Highly Testable Constructs	▪ Data Integrity
▪ Control & Management	▪ Monitor & Alerting	▪ Minimal Manual Intervention
▪ Support any Protocol/Entity	▪ Business Event Tracking	▪ Operational Audit

- Ensure traceability from architecture through to implementation
  - All too often architecture fails to translate into the real world
  - The right architecture can reduce complexity

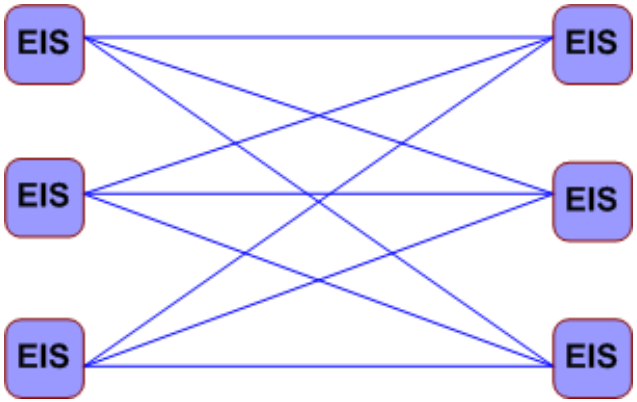
Translating the Architecture

[Key Strategies]

▪ Single Point of Integration	▪ Loose Coupling	▪ Variety of Standard Contracts
▪ Separation of Concerns	▪ Tight Cohesion	▪ Standard Design Patterns
▪ Interchangeable Constructs	▪ Highly Testable Constructs	▪ Data Integrity
▪ Control & Management	▪ Monitor & Alerting	▪ Minimal Manual Intervention
▪ Support any Protocol/Entity	▪ Business Event Tracking	▪ Operational Audit

[Issue] Commonly agreed that “spaghetti integration” is bad

- Integration can be exponentially complex
- Bleed out of APIs, data syntax and business semantics
- Ripple effect of change is massive



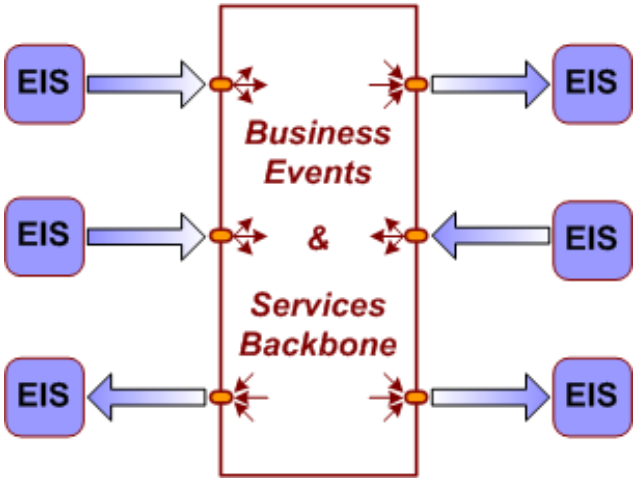
Translating the Architecture

[Key Strategies]

▪ Single Point of Integration	▪ Loose Coupling	▪ Variety of Standard Contracts
▪ Separation of Concerns	▪ Tight Cohesion	▪ Standard Design Patterns
▪ Interchangeable Constructs	▪ Highly Testable Constructs	▪ Data Integrity
▪ Control & Management	▪ Monitor & Alerting	▪ Minimal Manual Intervention
▪ Support any Protocol/Entity	▪ Business Event Tracking	▪ Operational Audit

[Concept] Standard Enterprise Application Integration (EAI) approach

- Provision of an Event/Service backbone
- Single point of integration for EIS business flow
- Ripple effect of change is localised *(assuming best practice)*



Translating the Architecture

[Key Strategies]

▪ Single Point of Integration	▪ Loose Coupling	▪ Variety of Standard Contracts
▪ Separation of Concerns	▪ Tight Cohesion	▪ Standard Design Patterns
▪ Interchangeable Constructs	▪ Highly Testable Constructs	▪ Data Integrity
▪ Control & Management	▪ Monitor & Alerting	▪ Minimal Manual Intervention
▪ Support any Protocol/Entity	▪ Business Event Tracking	▪ Operational Audit

[First Step] Modelling the Domain

- Defines fundamental entities, constructs, and their relationships
- Leverage existing domain knowledge
- External domain knowledge – we are not the first…
  - Enterprise Integration Design Patterns
    - [Ref. Enterprise Integration Patterns, Gregor Hohpe & Bobby Woolf, 2003]
  - Enterprise Application Design Patterns
    - [Ref. Patterns of Enterprise Application Architecture, Martin Fowler]

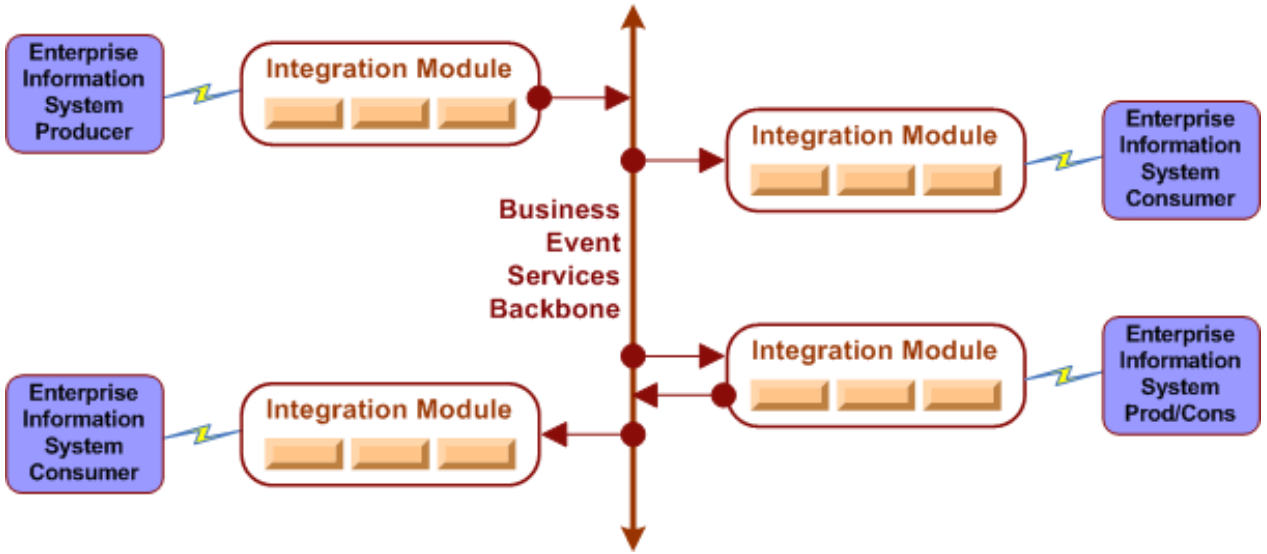
Translating the Architecture

[Key Strategies]

✓ Single Point of Integration	✓ Loose Coupling	✓ Variety of Standard Contracts
▪ Separation of Concerns	▪ Tight Cohesion	▪ Standard Design Patterns
▪ Interchangeable Constructs	▪ Highly Testable Constructs	▪ Data Integrity
▪ Control & Management	▪ Monitor & Alerting	▪ Minimal Manual Intervention
▪ Support any Protocol/Entity	▪ Business Event Tracking	▪ Operational Audit

[Construct] Integration Modules

- Provides a logical grouping of business operations as a single integration point
- Provide either a source, target, or bi-directional business flow.



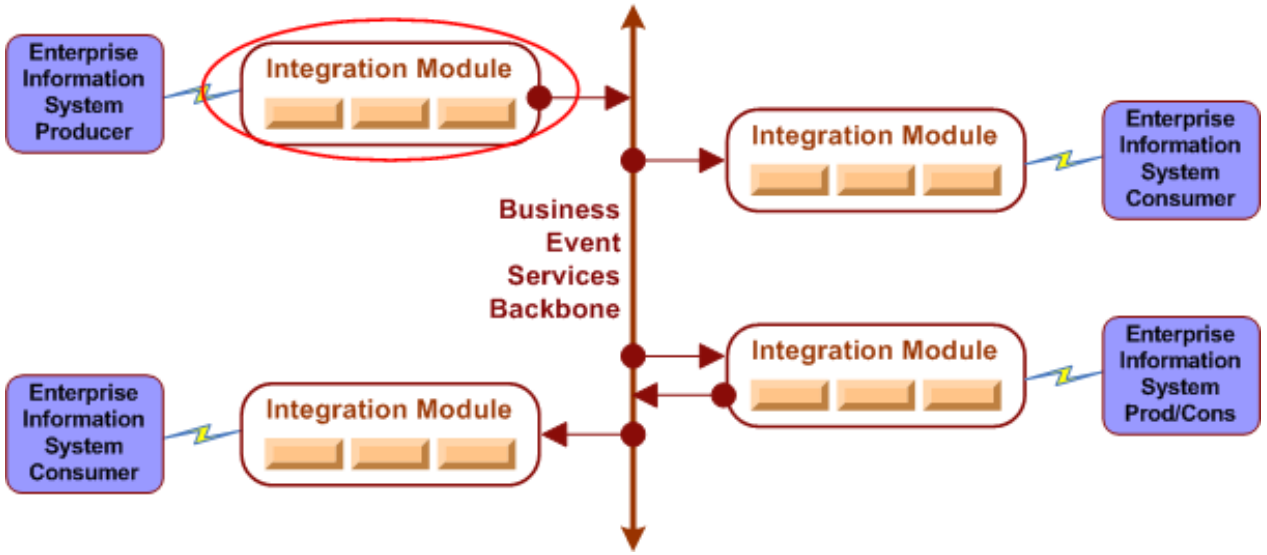
Translating the Architecture

[Key Strategies]

✓ Single Point of Integration	✓ Loose Coupling	✓ Variety of Standard Contracts
▪ Separation of Concerns	▪ Tight Cohesion	▪ Standard Design Patterns
▪ Interchangeable Constructs	▪ Highly Testable Constructs	▪ Data Integrity
▪ Control & Management	▪ Monitor & Alerting	▪ Minimal Manual Intervention
▪ Support any Protocol/Entity	▪ Business Event Tracking	▪ Operational Audit

[Construct] Integration Modules

- Provides a logical grouping of business operations as a single integration point
- Provide either a source, target, or bi-directional business flow.



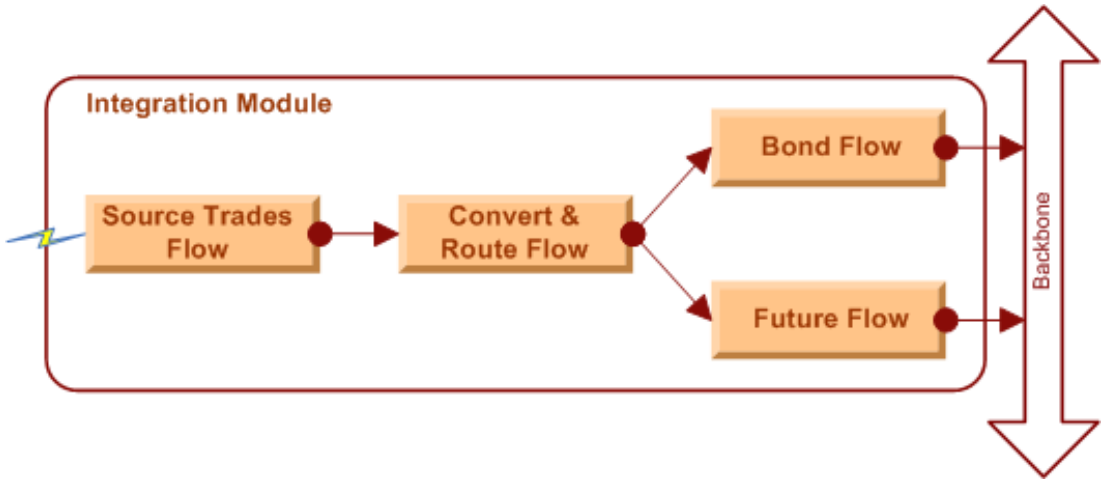
Translating the Architecture

[Key Strategies]

✓ Single Point of Integration	✓ Loose Coupling	✓ Variety of Standard Contracts
✓ Separation of Concerns	✓ Tight Cohesion	✓ Standard Design Patterns
▪ Interchangeable Constructs	▪ Highly Testable Constructs	▪ Data Integrity
▪ Control & Management	▪ Monitor & Alerting	▪ Minimal Manual Intervention
▪ Support any Protocol/Entity	▪ Business Event Tracking	▪ Operational Audit

[Construct] Flows

- Provide cohesive operations on a business artifact as a synchronous operation
- Multiple flows can be chained to isolate concerns
- Standard event container allows any data type to be transported



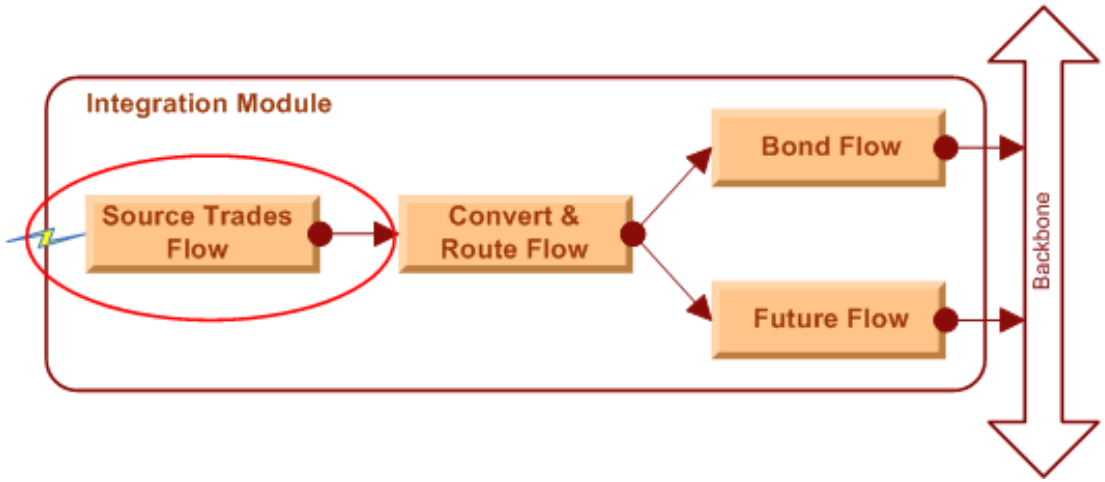
Translating the Architecture

[Key Strategies]

✓ Single Point of Integration	✓ Loose Coupling	✓ Variety of Standard Contracts
✓ Separation of Concerns	✓ Tight Cohesion	✓ Standard Design Patterns
▪ Interchangeable Constructs	▪ Highly Testable Constructs	▪ Data Integrity
▪ Control & Management	▪ Monitor & Alerting	▪ Minimal Manual Intervention
▪ Support any Protocol/Entity	▪ Business Event Tracking	▪ Operational Audit

[Construct] Flows

- Provide cohesive operations on a business artifact as a synchronous operation
- Multiple flows can be chained to isolate concerns
- Standard event container allows any data type to be transported



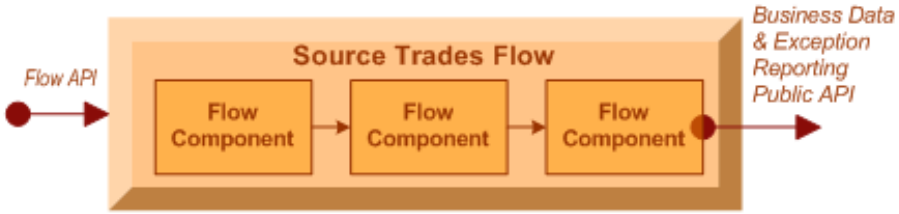
Translating the Architecture

[Key Strategies]

✓ Single Point of Integration	✓ Loose Coupling	✓ Variety of Standard Contracts
✓ Separation of Concerns	✓ Tight Cohesion	✓ Standard Design Patterns
✓ Interchangeable Constructs	✓ Highly Testable Constructs	✓ Data Integrity
▪ Control & Management	▪ Monitor & Alerting	▪ Minimal Manual Intervention
▪ Support any Protocol/Entity	▪ Business Event Tracking	▪ Operational Audit

[Construct] Flow Components

- POJO implementations provide independent, interchangeable operations
- Isolated component white box testing and chained black box testing
- All flow components within an atomic operation guaranteeing data integrity



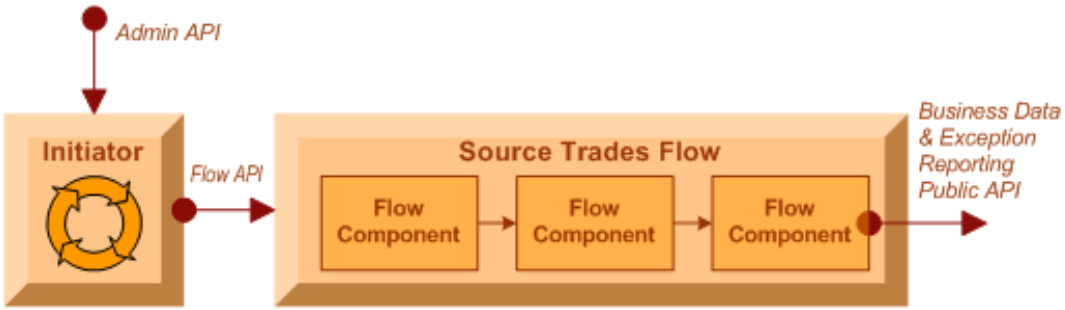
Translating the Architecture

[Key Strategies]

✓ Single Point of Integration	✓ Loose Coupling	✓ Variety of Standard Contracts
✓ Separation of Concerns	✓ Tight Cohesion	✓ Standard Design Patterns
✓ Interchangeable Constructs	✓ Highly Testable Constructs	✓ Data Integrity
✓ Control & Management	✓ Monitor & Alerting	✓ Minimal Manual Intervention
▪ Support any Protocol/Entity	▪ Business Event Tracking	▪ Operational Audit

[Construct] Initiators

- Business flow control (start & stop)
- Notification of flow status (running, stopped, recovery, stoppedInError)
- Configurable exception handling for automated flow recovery
- Public API for external client management



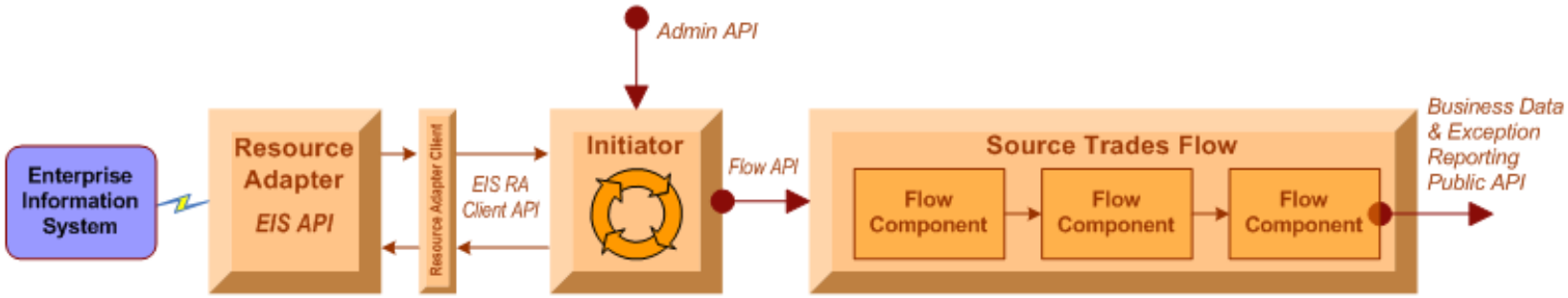
Translating the Architecture

[Key Strategies]

✓ Single Point of Integration	✓ Loose Coupling	✓ Variety of Standard Contracts
✓ Separation of Concerns	✓ Tight Cohesion	✓ Standard Design Patterns
✓ Interchangeable Constructs	✓ Highly Testable Constructs	✓ Data Integrity
✓ Control & Management	✓ Monitor & Alerting	✓ Minimal Manual Intervention
✓ Support any Protocol/Entity	▪ Business Event Tracking	▪ Operational Audit

[Construct] Resource Adapters

- Support any type of protocol / API
- Extend guaranteed integrity to non-guaranteed EIS sources/targets
- No proprietary knowledge required; only EIS API
- Resource adapters pluggable re-usable components (SUN JCA spec)
- 3rd party off-the-shelf or internally developed



Translating the Architecture

[Key Strategies]

✓ Single Point of Integration	✓ Loose Coupling	✓ Variety of Standard Contracts
✓ Separation of Concerns	✓ Tight Cohesion	✓ Standard Design Patterns
✓ Interchangeable Constructs	✓ Highly Testable Constructs	✓ Data Integrity
✓ Control & Management	✓ Monitor & Alerting	✓ Minimal Manual Intervention
✓ Support any Protocol/Entity	✓ Business Event Tracking	✓ Operational Audit

[Construct] Administration Console

- Initiator management
  - runtime status (stopped, running, recovery, stoppedInError)
  - runtime control (start, stop, start-up type {auto, manual, disabled})
- Flow management
  - flow component chain view and drill down
  - audit log of flow actions (stop/start)
- Wiretap management
  - dynamic creation/deletion of real-time in-flight data capture
- User management
  - creation, deletion, and permission maintenance

Translating the Architecture

[Key Strategies]

✓ Single Point of Integration	✓ Loose Coupling	✓ Variety of Standard Contracts
✓ Separation of Concerns	✓ Tight Cohesion	✓ Standard Design Patterns
✓ Interchangeable Constructs	✓ Highly Testable Constructs	✓ Data Integrity
✓ Control & Management	✓ Monitor & Alerting	✓ Minimal Manual Intervention
✓ Support any Protocol/Entity	✓ Business Event Tracking	✓ Operational Audit

[Construct] User Console

- In-flight business event searches
  - by module(s)
  - by flow(s)
  - by business stream(s)
  - parameter filters
    - business content, date/time, names, etc
- Event “Life Identifier” linking
  - flow component chain view and drill down
  - audit log of flow actions (stop/start)
- User management
  - creation, deletion, and permission maintenance

Translating the Architecture

[Key Strategies]

✓ Single Point of Integration	✓ Loose Coupling	✓ Variety of Standard Contracts
✓ Separation of Concerns	✓ Tight Cohesion	✓ Standard Design Patterns
✓ Interchangeable Constructs	✓ Highly Testable Constructs	✓ Data Integrity
✓ Control & Management	✓ Monitor & Alerting	✓ Minimal Manual Intervention
✓ Support any Protocol/Entity	✓ Business Event Tracking	✓ Operational Audit

[Construct] Monitor Clients

- Big Brother / Hobbit client
- Email client
- Log scrapper
- HTTP client

Translating the Architecture

[Key Strategies]

✓ Single Point of Integration	✓ Loose Coupling	✓ Variety of Standard Contracts
✓ Separation of Concerns	✓ Tight Cohesion	✓ Standard Design Patterns
✓ Interchangeable Constructs	✓ Highly Testable Constructs	✓ Data Integrity
✓ Control & Management	✓ Monitor & Alerting	✓ Minimal Manual Intervention
✓ Support any Protocol/Entity	✓ Business Event Tracking	✓ Operational Audit

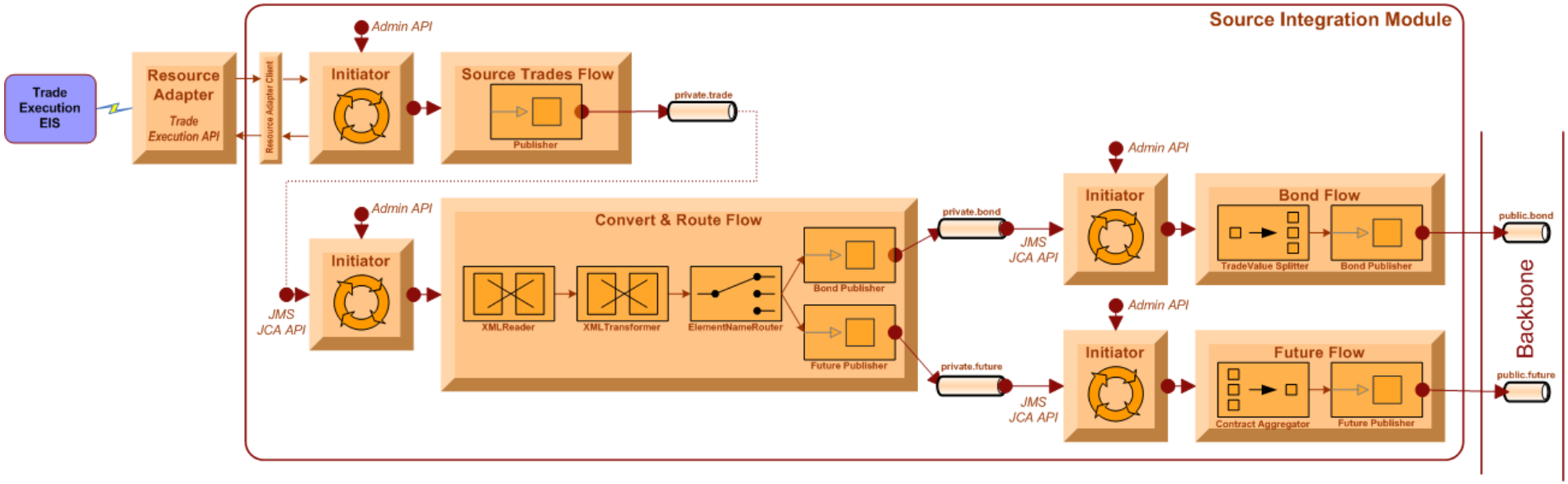
[Construct] OSS Test Frameworks & Ikasan FlowTestHarness

- Flow component automated unit testing
  - Junit, JMock, XMLUnit
- Initiator automated unit testing
  - JUnit, JMock
- Flow automated unit (integration) testing
  - Spring-test, Ikasan FlowTestHarness, HSQL in memory database
- Module integration testing
  - Currently manual
  - Future: automation through maven plugins

Translating the Architecture

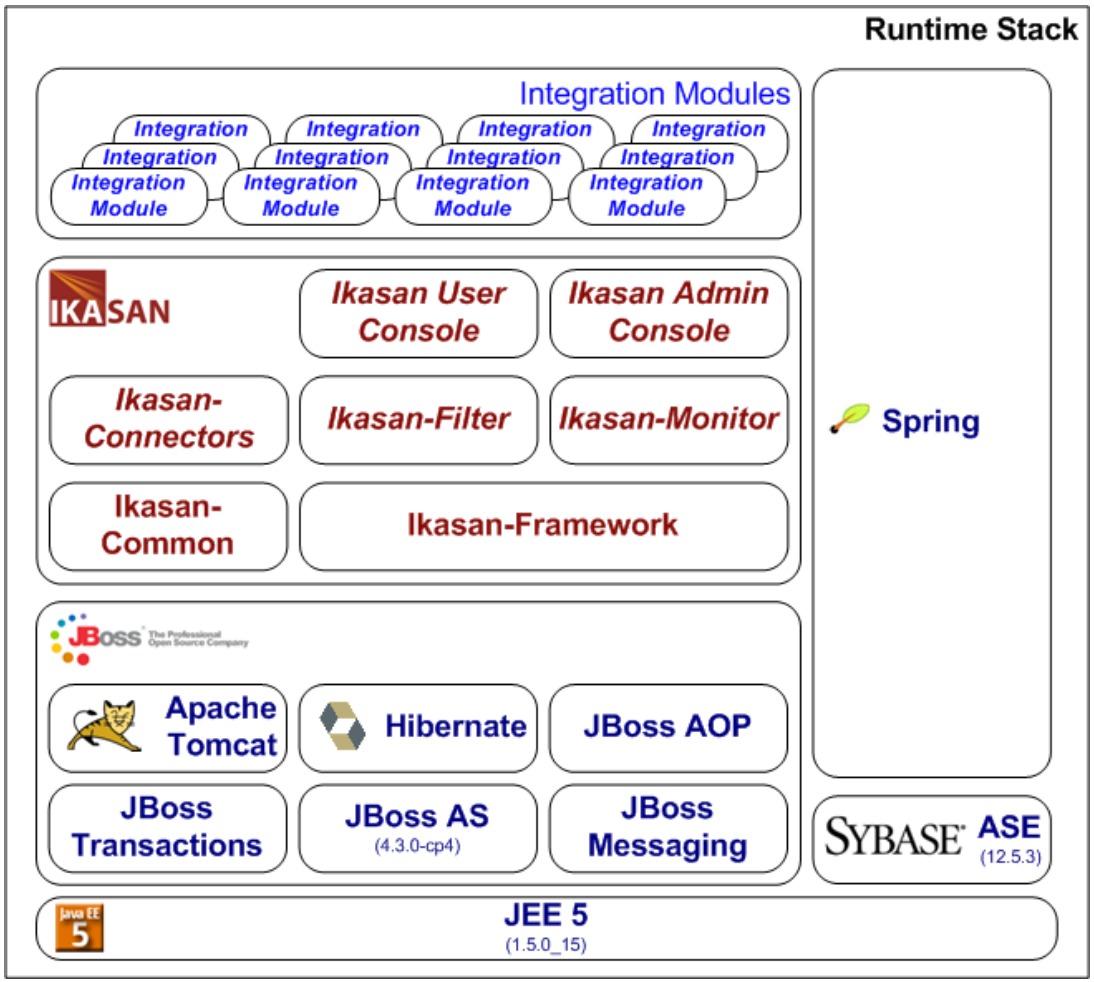
- Problem Domain
- Architecture Strategies
- **Translating Architecture**
- Technology Stack
- Features
- Development Lifecycle
- Roadmap
- Summary

- Sample Source Integration Module
  - Source Trades Flow
    - Get trades (CSV format) from EIS with minimal failure potential
  - Convert & Route Flow
    - Transform CSV -> XML syntax; Transform XML semantics; Route on type
  - Bond Flow
    - Split one bond event into many bond events based on trade size rule
  - Future Flow
    - Aggregate multiple future contract events until *x* then push as single event



Technology Stack

- Runtime Stack



Technology Stack

- Development Stack
  - Core technologies
    - Java 1.5+ / JEE
    - XML
    - Spring Framework 2.6
  - Test Frameworks
    - JUnit
    - JMock
    - SpringTest
    - Ikasan Flow Test Harness 0.7.10
  - Scaffolding
    - Source Management – Subversion
    - Requirements and Issue Management – JIRA
    - Peer Review and Acceptance Management – Crucible, FishEye
    - Project Build Management – Maven
    - Project Artefact Management – Sonotype Nexus
    - Continuous Integration – Hudson

Features

- Version 0.7.10 current production release
- Interoperability
  - Protocols
    - JMS
    - FTP (transactional & chunking support)
    - SFTP (transactional & chunking support)
    - RDBMS (via Hibernate ORM)
    - SMTP (Email)
    - HTTP(s)
  - Vendor APIs
    - Trax Messaging
    - PATS Trade Feed
    - Espeed Trade feed
    - Tradeweb Trade Feed
    - Bloomberg Data License Per Security
  - Data Constructs
    - Event – Ikasan internal entity container
    - Payload – actual entity in-flight

Features

- Functionality
  - Initiators – business flow control
    - Allows a client to stop, start, enable, disable, query runtime status
    - Event Driven Type
      - Any resource adapter protocol
    - Schedule Driven Type
      - Quartz Scheduler for absolute and relative invocation scheduling
  - Flows
    - Full Transactional Semantics
      - Non-transactional
      - Local transactions
      - 2-Phase transactions
      - Last resource commit optimisation (LRCO)
        - incorporate non-transactional resources within a distributed transaction
    - Configurable Automated Recovery
      - Rollback operations & stop
      - Rollback operations and retry in x seconds
      - Rollback operations and retry a number of times or indefinitely
      - Report the exception and continue

Features

- Functionality
  - Flow Components
    - Transformers
      - XSLT
      - XML Reader
      - XML Validator
      - any POJO implementing Transformer interface
    - Routers
      - Recipient list
      - Content
      - Xpath
      - Any POJO implementing Ikasan router interface
    - Sequencers
      - Aggregator
      - Splitter
      - Any POJO implementing Ikasan sequencer interface
    - Endpoints
      - Loggers
      - Hard wiretap
      - Any POJO implementing Ikasan endpoint interface
      - Any resource adapter client API

Features

- Functionality
  - Dynamically capture runtime data at any point in the flow
  - Users able to search on business content across a flow

Development Lifecycle

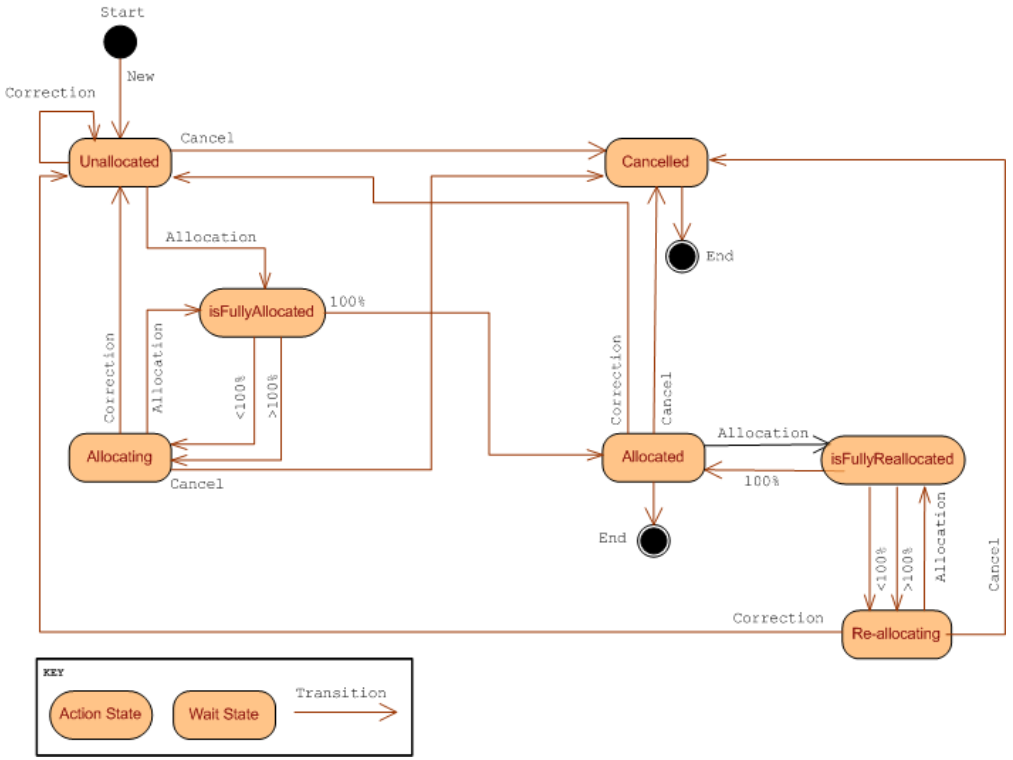
- Approach based on TDD/BDD Agile dogma
  - Gather/refine user stories and create use cases
  - Sketch the stories into a module/flow design
  - Create/refine the Flow Test Harness reflecting the design
  - Create modules/flows/components to meet test harness acceptance
  - Peer/pair walk through (10 minute daily stroll)
    - Design reviews
    - Code reviews
  - Continuous Integration (CI) test coverage increases
  - Deliver functionality to users for proofs
  - Repeat!
- Recent Example Implementation Times
  - 3 days – Completed module delivery for transformed XML; real-time and batch aggregation; delivered over HTTP(S) to downstream application
- Demo

Roadmap

- Version 0.8.0 due shortly, currently in beta
- New Features
  - Event hospital
    - Resubmission of failed business events
    - Public API invocable over HTTP
    - Full history of exceptions on a given event
  - Simplified Domain Model
    - Support any user defined entity in the domain model
    - Support of any user defined domain properties

Roadmap

- Version 0.8.x or 0.9.0
- New Feature
  - State Modeling
    - Standard Event Condition Action (ECA) State Pattern supporting
      - wait states
      - action states
      - conditional & default transitions



Roadmap

- Future Releases
  - Functionality
    - event replay
    - business subscription streams as an exposed registration service
  - Monitoring & Management
    - Flow status notification management service
      - metrics and stats
      - active monitoring rules
  - Documentation
    - Quick Start Guide
    - Developer Guide

Summary

- Ikasan Project Resource Summary
  - Web Site: http://www.ikasan.org
  - SVN: http://open.jira.com/svn/IKASAN
  - Wiki: http://sourceforge.net/apps/mediawiki/ikasaneip/index.php?title=Main_Page
  - JIRA: http://open.jira.com/browse/IKASAN
  - Mailing List: http://lists.sourceforge.net/lists/listinfo/ikasaneip-user
  - IRC: http://sourceforge.net/apps/mediawiki/ikasaneip/index.php?title=IRC

Open Source Enterprise Integration Platform

Jeff Mitchell
ikasan.org
July 2010