Jozef de Vries & Joseph Leong – IBM Rational Integrations Gearbox Team

IBM Rational OSLC Adapter for Atlassian JIRA

© 2012 IBM Corporation

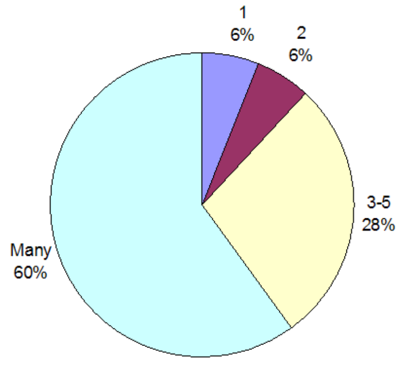
Topics

- Rational’s strategy for 3rd party integrations
- Demo of the Rational OSLC adapter for Atlassian JIRA
- Development design and strategy for the adapter

© 2012 IBM Corporation

Lifecycle tool environments are becoming increasingly heterogeneous…

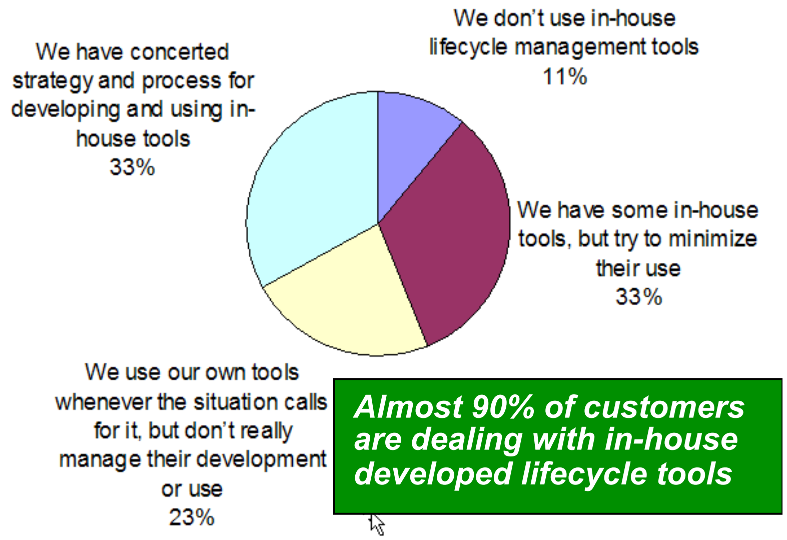
# of commercial lifecycle tool vendors



Almost all are managing 3 or more lifecycle tool vendors

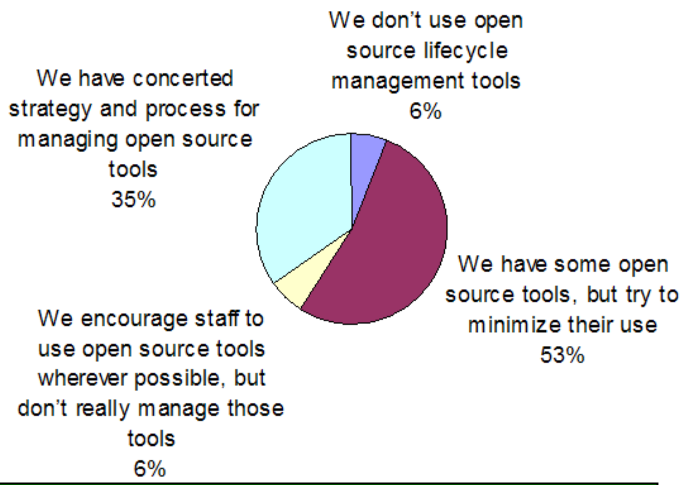
Source: Rational Voice of the Customer Event, November 2010

Role of in-house developed tools



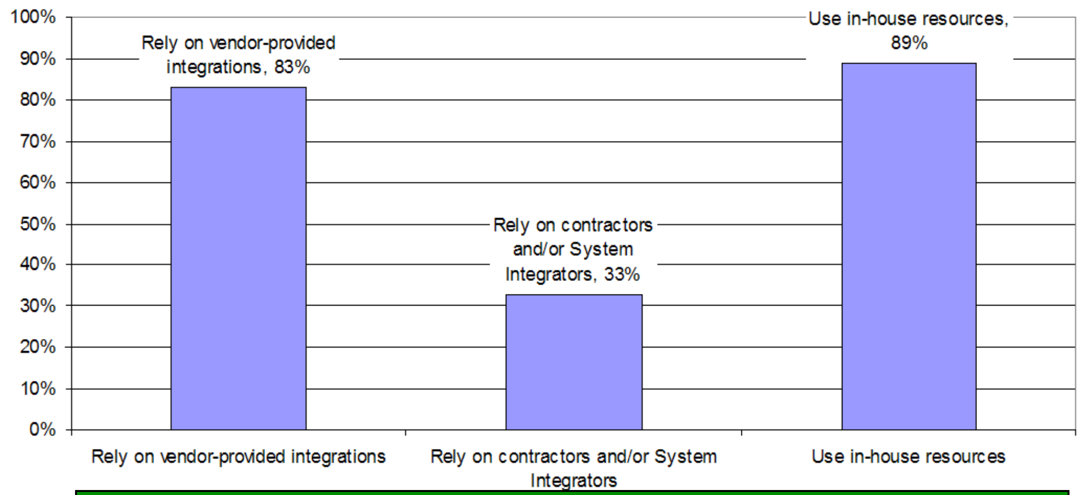
Almost 90% of customers are dealing with in-house developed lifecycle tools

Role of open source lifecycle tools



Almost all are dealing with open source lifecycle tools, but 2/3 say not very well

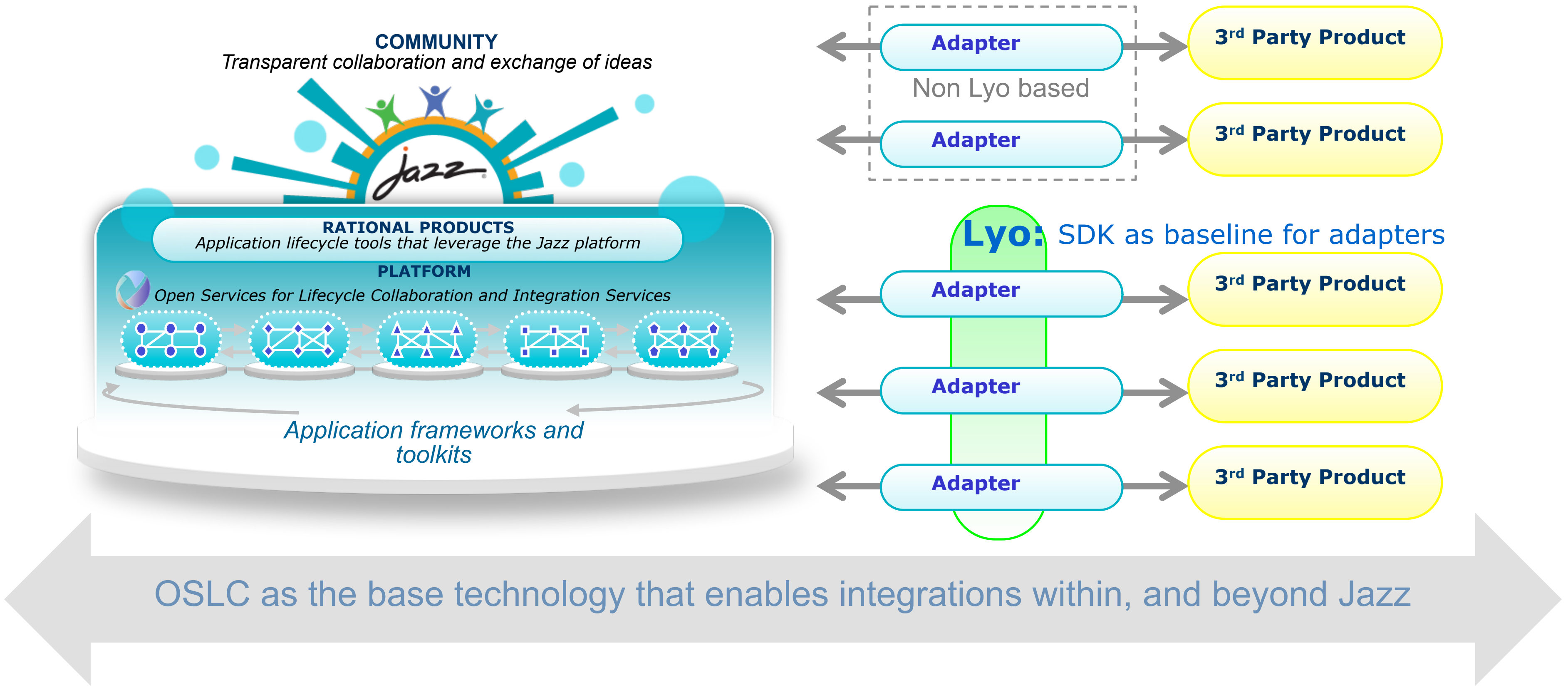
What resources do you use to develop and maintain lifecycle tool integrations?



Vast majority supplement vendor-provided integrations with in-house effort

© 2012 IBM Corporation

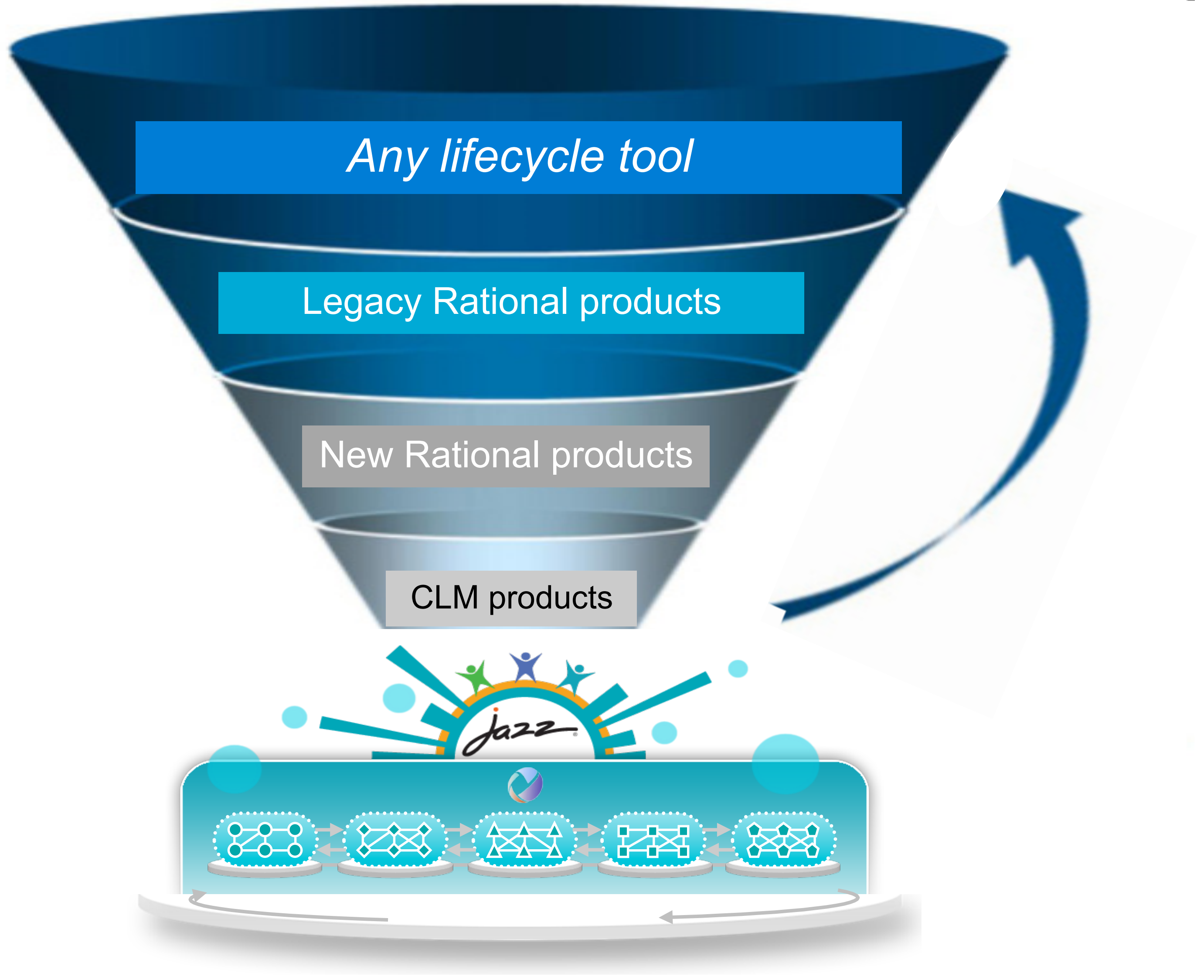
Rational Integrations and OSLC


COMMUNITY
Transparent collaboration and exchange of ideas
Jazz
RATIONAL PRODUCTS
Application lifecycle tools that leverage the Jazz platform
PLATFORM
Open Services for Lifecycle Collaboration and Integration Services
Application frameworks and toolkits
Adapter
Non Lyo based
Adapter
3rd Party Product
3rd Party Product
Lyo: SDK as baseline for adapters
Adapter
Adapter
Adapter
3rd Party Product
3rd Party Product
3rd Party Product
OSLC as the base technology that enables integrations within, and beyond Jazz

- Jazz provides an integration platform
- Project Lyo provides an OSLC SDK
- Adapters are built in-house, by partners, by community, and as service assets

© 2012 IBM Corporation

Rational's vision - integrate tools in the entire lifecycle whether they are from Rational, third-parties or home-grown



© 2012 IBM Corporation

Rational Integrations Gearbox

3rd party integrations

- Work with Product Mgt on governance and models
- Build adaptors/integrations/assets
- Act as dev-side in partner engagements
- Develop multi-product scenarios

OSLC assets

- Reference implementation across final specs
- Test suites
- Sample code
- Tutorials/primers/best practices
- In tight concert w/ OSLC team around evangelizing community

Integration assets

- Examine existing integrations to improve use cases for customers
- Determine based on pain pts or strategy where to bolster w/ assets
- Shape future requirements for nextgen versions of integrations
- Leverage ISSR, support, sales, JS, ULL

OSLC

Jazz Integration Services

Dedicated Development organization focused on 3rd party Integrations

© 2012 IBM Corporation

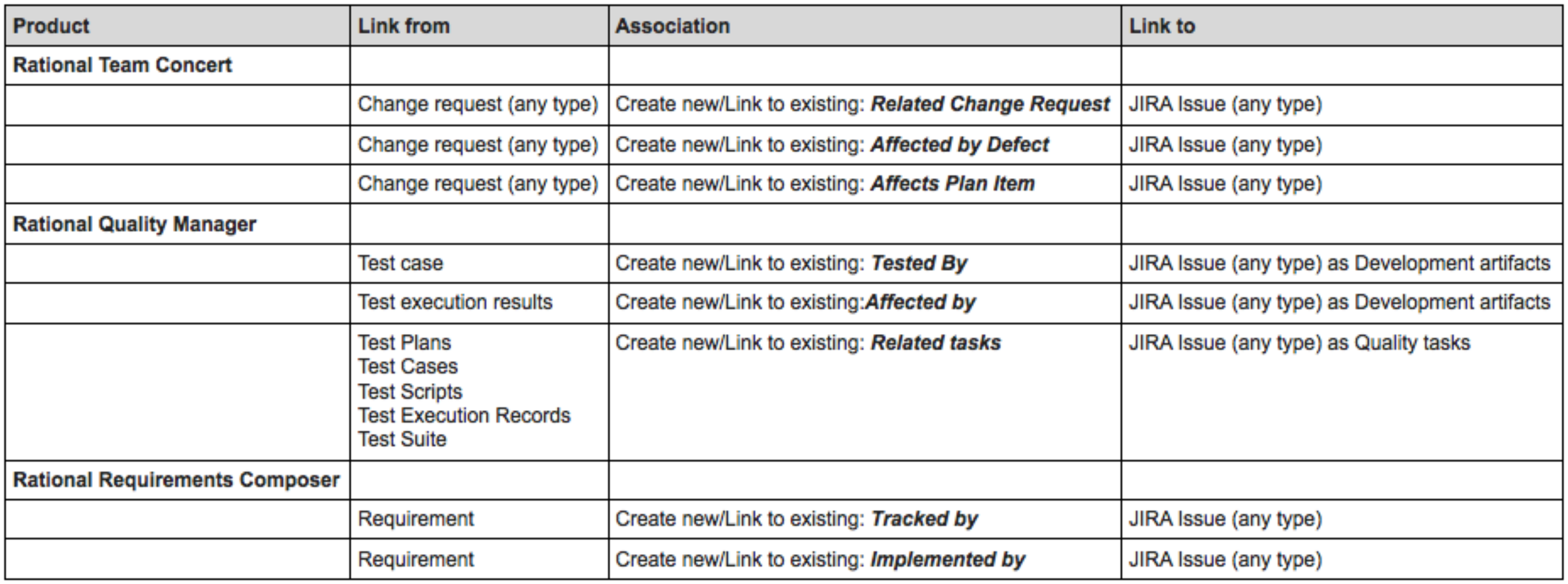
Typical customer situations that we want to address

- "RTC seems interesting, but I have a large investment in HPQC: How do I manage traceability? Where do I manage defects"
- "We would like to use RRC for requirements, but need to ensure that our testers in HP QC can provide test coverage for all the requirements in a release"
- "We contract out component development to teams that track defects with JIRA – how can I connect my RQM testing efforts to these groups?"
- "I have a large pool of users using GIT, and we're interested in using RTC for CM activities. How can I ensure proper governance over code changes?"

© 2012 IBM Corporation

Rational OSLC Adapter for JIRA - Demo

- JIRA in place of, or in addition to the CM component in RTC/Jazz Foundation



Product	Link from	Association	Link to
Rational Team Concert			
	Change request (any type)	Create new/Link to existing: ***Related Change Request***	JIRA Issue (any type)
	Change request (any type)	Create new/Link to existing: ***Affected by Defect***	JIRA Issue (any type)
	Change request (any type)	Create new/Link to existing: ***Affects Plan Item***	JIRA Issue (any type)
Rational Quality Manager			
	Test case	Create new/Link to existing: ***Tested By***	JIRA Issue (any type) as Development artifacts
	Test execution results	Create new/Link to existing:***Affected by***	JIRA Issue (any type) as Development artifacts
	Test Plans Test Cases Test Scripts Test Execution Records Test Suite	Create new/Link to existing: ***Related tasks***	JIRA Issue (any type) as Quality tasks
Rational Requirements Composer			
	Requirement	Create new/Link to existing: ***Tracked by***	JIRA Issue (any type)
	Requirement	Create new/Link to existing: ***Implemented by***	JIRA Issue (any type)

© 2012 IBM Corporation

Development design and strategy for the adapter

© 2012 IBM Corporation

What are the fundamental tools involved?

- **Rational solution for Collaborative Lifecycle Management (CLM)**
  - Rational Team Concert (RTC) – CM, Planning, SCM, Build Automation
  - Rational Quality Manager (RQM) - QM
  - Rational Requirements Composer (RRC) - RM

- **Atlassian JIRA**
  - Issue tracking system
  - Change management system

© 2012 IBM Corporation

What is the objective of our development?

- **Rational CLM Products ←--- Integration ---→ Atlassian JIRA**
  - Ex 1: RTC Work Item → *Create/Link* → JIRA Issue
  - Ex 2: RQM Test Execution Record → *Create/Link* → JIRA Issue
  - Ex 3: RRC Requirement → *Create/Link* → JIRA Issue
- **How do we go about this integration?**
  - Let's use a **specification** that describes what we are trying to communicate

© 2012 IBM Corporation

OSLC Provider & Consumer

- Roles that describe how an application interacts with the OSLC specification

- OSLC specifications provide a principle to describe data in different domains
  - Domain Examples: Change Management, Requirements, and Quality Management
  - Association with each other
  - Linked data model

- An OSLC provider is responsible for exposing domain data in accordance with the OSLC specification that creates exposure to creating, updating and querying linked data

- An OSLC consumer is responsible for consuming the OSLC provider services so that it can manifest access to the domain data through delegated interfaces and service calls

© 2012 IBM Corporation

What Roles are the Products Playing?

- OSLC CM Provider:
  In a nutshell, hosts the CM Data and surfaces it through the specification
- OSLC CM Consumer:
  Again, in a nutshell, provides a way to get at the CM Data through the specification

- Rational CLM is an OSLC CM Consumer
  - Get this for free because Rational CLM already implements this

- Atlassian JIRA is an OSLC CM Provider
  - In other words, the objective of the integration is to implement an OSLC CM Provider in JIRA!

© 2012 IBM Corporation

Specifications

- Basically a set of expectations you and I understand
  - Example: Tell me about a car you want to sell (English, Car Vocabulary, …)

- How to go about creating the Rational CLM and Atlassian JIRA integration?
  - The integration point between the two products is creating and accessing Change Management (CM) domain data
  - What is Change Management data?
  - Rational CLM products, as a whole, consume CM data
  - Atlassian JIRA produces CM data (Such as Bugs, Tasks, New Features…)
  - Leverage **OSLC Change Management Specification**
    - Provides a set of expectations for how to describe CM data

© 2012 IBM Corporation

Delegated Interfaces / Services

- Some concrete examples of the OSLC CM Specification (Recall from the Demo)
  - Creation Dialog / Selection Dialog
    - Specification provided the location to access this
  - UI Preview
    - Specification provided the consumer the understanding of this capability
  - Backlinks in JIRA back to CLM
    - Service layer underneath
    - Allow REST calls for discovery / creation / updates

© 2012 IBM Corporation

Recap

- OSLC technology is the basis to enable this integration through providers and consumers
- Objective: Integrate Rational CLM with Atlassian JIRA
- Ability for Rational CLM to create and access CM data in Atlassian JIRA
- Leverage OSLC Change Management Specification to set the expectation for what the communication of CM data between products will look like
- **Questions?**

© 2012 IBM Corporation

Design Decisions

- Not going to cover the OSLC CM spec itself
  - Details of the attributes and shape of the data can be found at http://open-services.net
  - Focus on Design Decisions that materialized the integration

- Outline
  - Embodiment of this implementation
  - Mechanisms to surface the OSLC CM Provider data
  - Data representation
  - Security
  - Specification Validation / Compliance Level
  - Retrospective and Improvements (LYO)

© 2012 IBM Corporation

Embodiment of Implementation

Embedded Atlassian JIRA Plugin or Standalone Server Integration?

- **Embedded Plugin**
  - **+** Closer to JIRA data (aggregate data / computations locally, less HTTP overhead)
  - **+** Ability to extend and manipulate the JIRA web interface (backlinks)
  - **+** A much richer API (reuse query language, leverage existing security mechanism)

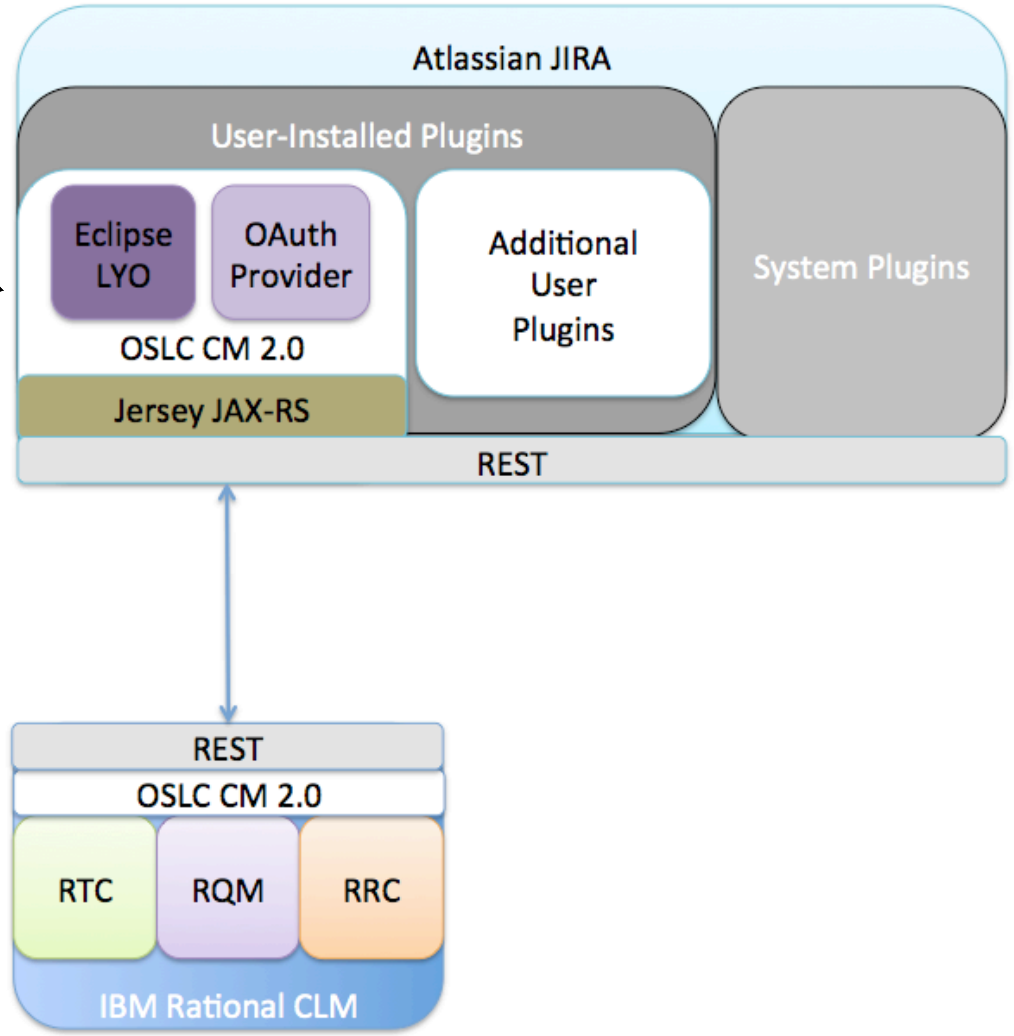
- **Standalone Server**
  - **+** Creating a more reusable OSLC CM provider architecture for other integrations
  - **+** Working with JIRA REST API is a lot less complex
  - **+** Potentially less admin access is required

- **Conclusion:** Embedded Plugin approach, primarily because of the need to access a richer API and extend the web interface

© 2012 IBM Corporation

Embodiment of Implementation

Embedded Plugin



© 2012 IBM Corporation

Surfacing the OSLC CM Provider data

How to handle serving all the different web contexts and services?

- OSLC is a RESTful service

- **JAX-RS**
  - Great to delegate serving different contexts and routing different HTTP verbs driven requests
  - Utilizes JAX-RS framework Jersey, which also serves their JIRA REST API
  - No additional binary footprint to get this because Jersey was already included in JIRA

© 2012 IBM Corporation

Data Representation

- Example artifacts of the specification
  - Service Catalog
  - Service Provider
  - ChangeRequest etc…
- Created POJOs to represent our change management domain data
  - Created a base set of POJOs to model the OSLC CM resources
  - Extended base POJOs to capture additional data JIRA Issues provide
- High frequency of converting POJOs to various data formats
  - Leverage an automated process
    - JAXB XML Marshalling / Unmarshalling of POJOs (Annotations to describe XML)
    - JSON Marshalling / Unmarshalling of POJOs

© 2012 IBM Corporation

Security

Who is allowed to access a JIRA issue?

Who is allowed to access a Change Management resource?

- Delegate security to respective product + OAuth
  - CLM Users must login to access their resources (nothing new)
  - When attempting to create/link artifacts in JIRA
    - Ex: Surface dialogs from JIRA (Creation / Selection)

    - Ex: When CLM is updating backlinks in JIRA

- JIRA Permission manager to authenticate requests in a session

© 2012 IBM Corporation

Specification Validation / Compliance Level

- How to validate your OSLC implementation?
  - Lyo Test Suite
    - Assesses the level of compliance your implementation offers
    - Can be used in conjunction with build testing
    - Generates compliance reports
    - Available on eclipse project site

- Compliance Reports and Levels
  - Different levels of integrations (MUST / SHOULD / MAY etc…)
  - For example, Rational CLM products tend to implement more parts of the specification
    - Provide richer integrations (Ex: OAuth, RDF/XML Data representations, etc..)
  - The compliance report helps assess these compliance levels
    - Generated graph for quick view
    - Summary
    - Break down of validation tests

© 2012 IBM Corporation

Retrospective and Improvements (LYO)

- Previously, only took advantage of OSLC Constants, Basic POJOs, Test Suite

- Good News!

- Lyo SDK now has provided a lot more to help facilitate implementations
  - OAuth Provider framework
  - Annotations schema on top of POJOs to generate OSLC documents
  - Out of the box JSON / RDF+XML marshalling / unmarshalling of OSLC documents
  - RDF storage utilities to persist global metadata
  - More underway – Eclipse Lyo project roadmap
  - Get involved to help us understand what you need

© 2012 IBM Corporation

Questions?

© 2012 IBM Corporation

Resources

- Rational OSLC adapter for JIRA: https://jazz.net/library/article/766
- Open Services: http://open-services.net/
- OSLC CM V2 Specifications:

  http://open-services.net/bin/view/Main/CmHome?sortcol=table;up=#2_0_Finalized
- Eclipse LYO OSLC SDK / Test Suite: http://www.eclipse.org/lyo/

© 2012 IBM Corporation