

# Alternatives for Systems Integration in the NoSQL Era

Kai Wähner

kwaehner@talend.com

@KaiWaehner

www.kai-waehner.de

9/6/2013





Consulting  
Developing  
Coaching  
Speaking  
Writing

## Main Tasks

Requirements Engineering  
Enterprise Architecture Management  
Business Process Management  
Architecture and Development of Applications  
Service-oriented Architecture  
Integration of Legacy Applications  
Cloud Computing  
Big Data

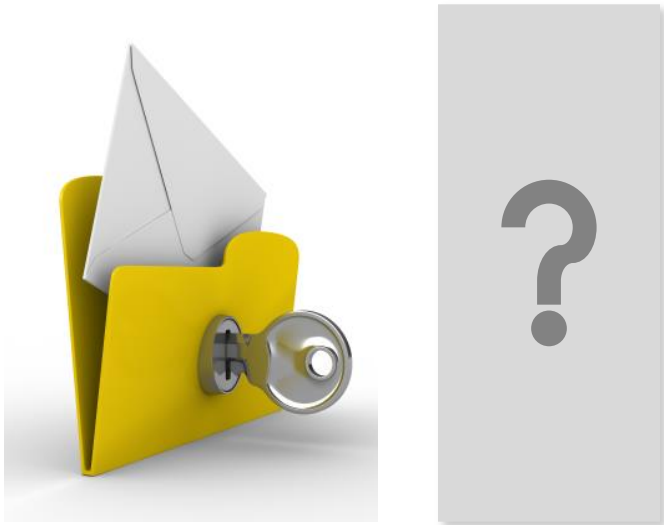
## Contact

Email: [kwaehner@talend.com](mailto:kwaehner@talend.com)  
Blog: [www.kai-waehner.de/blog](http://www.kai-waehner.de/blog)  
Twitter: @KaiWaehner  
Social Networks: Xing, LinkedIn

# What is the key message?



# Key messages



NoSQL cannot be avoided, and must be integrated!

NoSQL integration is already possible!

Different APIs, Frameworks and Products helps a lot!

# Agenda

- 1) Introduction to NoSQL
- 2) Systems Integration
- 3) API
- 4) Integration Framework
- 5) Enterprise Service Bus
- 6) Integration Suite
- 7) Custom Components



## Live Demos

# Agenda

## 1) Introduction to NoSQL

## 2) Systems Integration

## 3) API

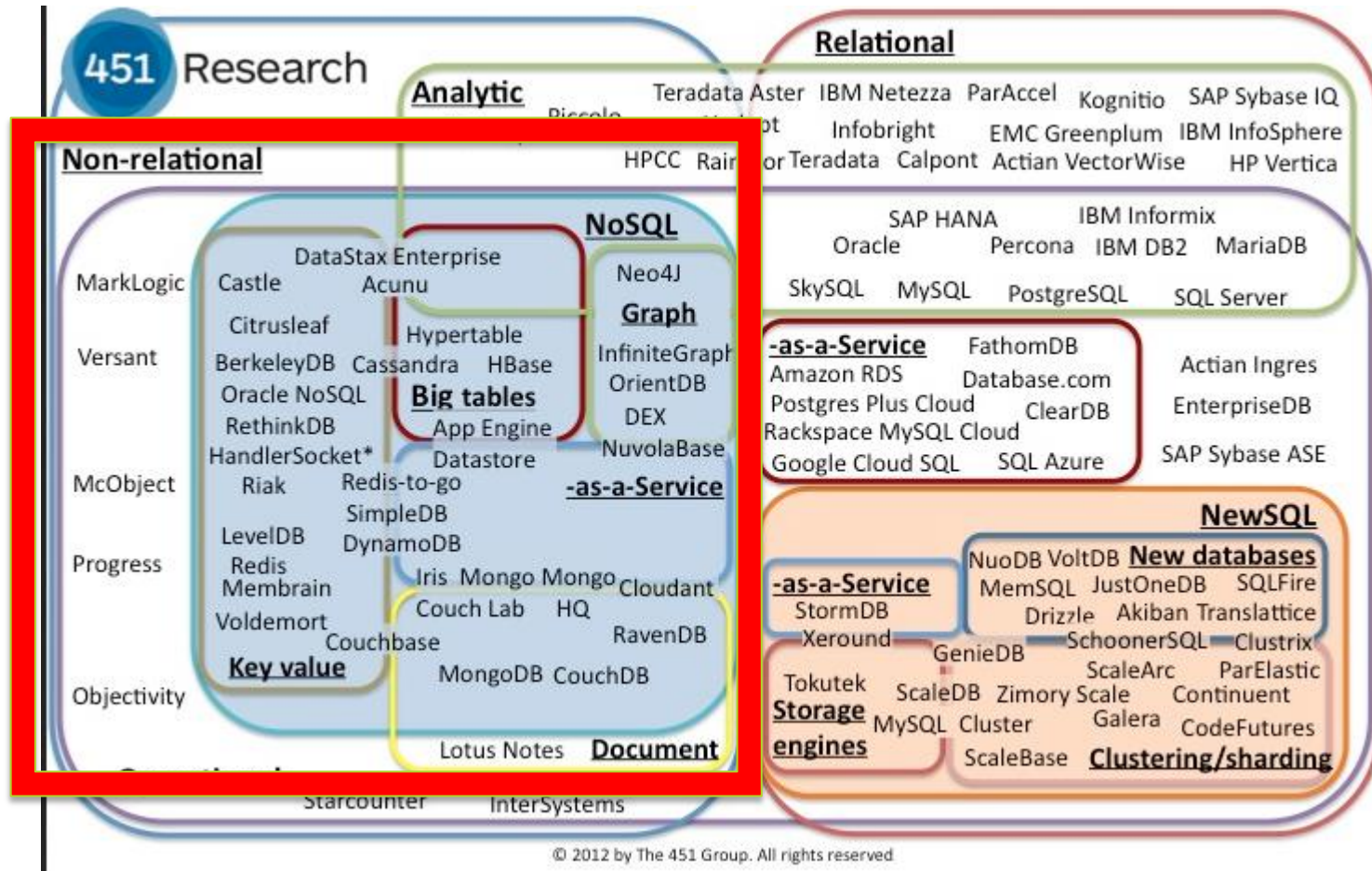
## 4) Integration Framework

## 5) Enterprise Service Bus

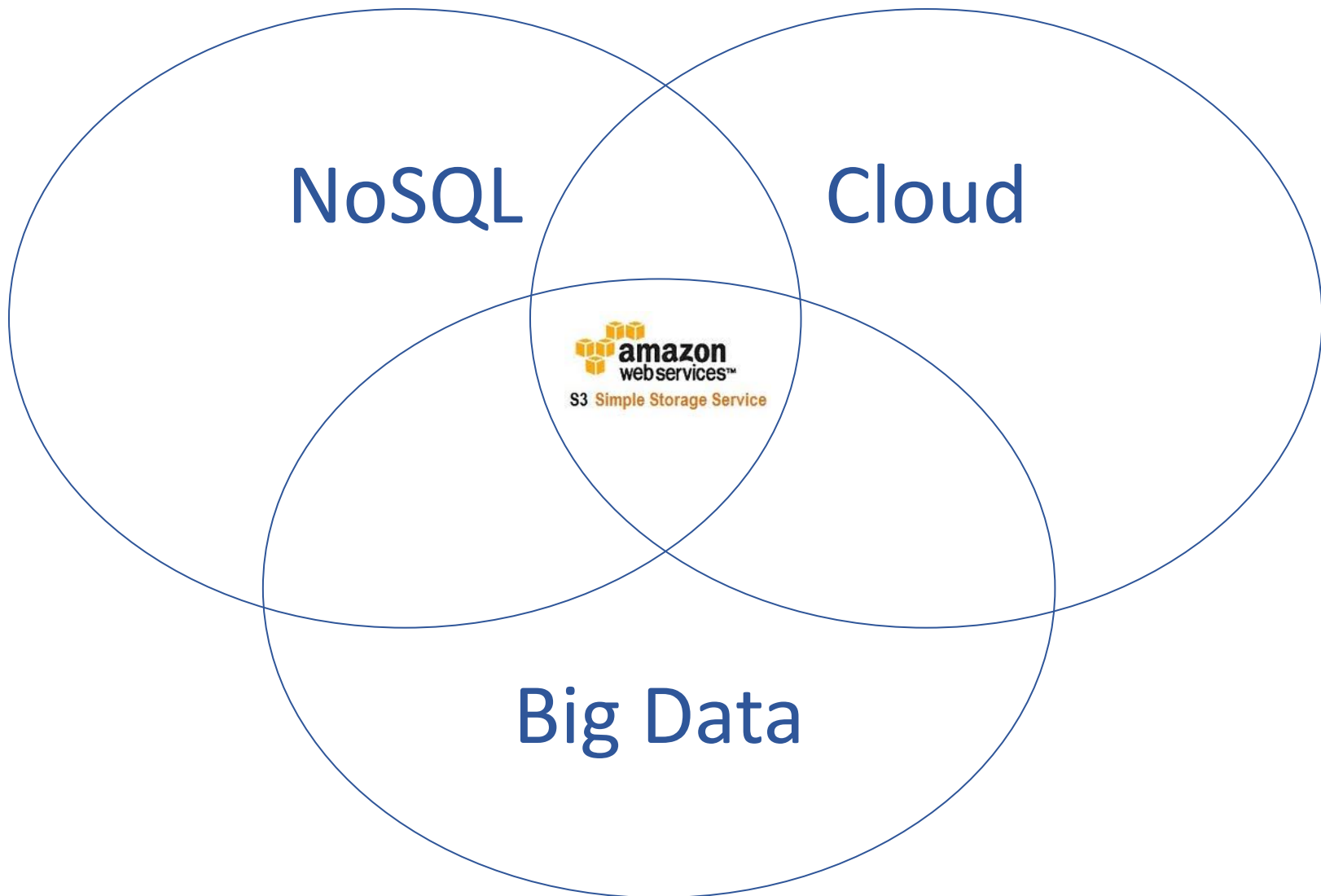
## 6) Integration Suite

## 7) Custom Components

# The evolving database landscape



# Complementary concepts





# Agenda

- 1) Introduction to NoSQL
- 2) Systems Integration**
- 3) API
- 4) Integration Framework
- 5) Enterprise Service Bus
- 6) Integration Suite

# A new era: NoSQL



# What is the problem?



→ „Spaghetti communication“

# What is the problem?



## Growth

- Applications
- Interfaces
- Technologies
- Products

# Solution: Systems integration



All Roads lead  
to Rome ...

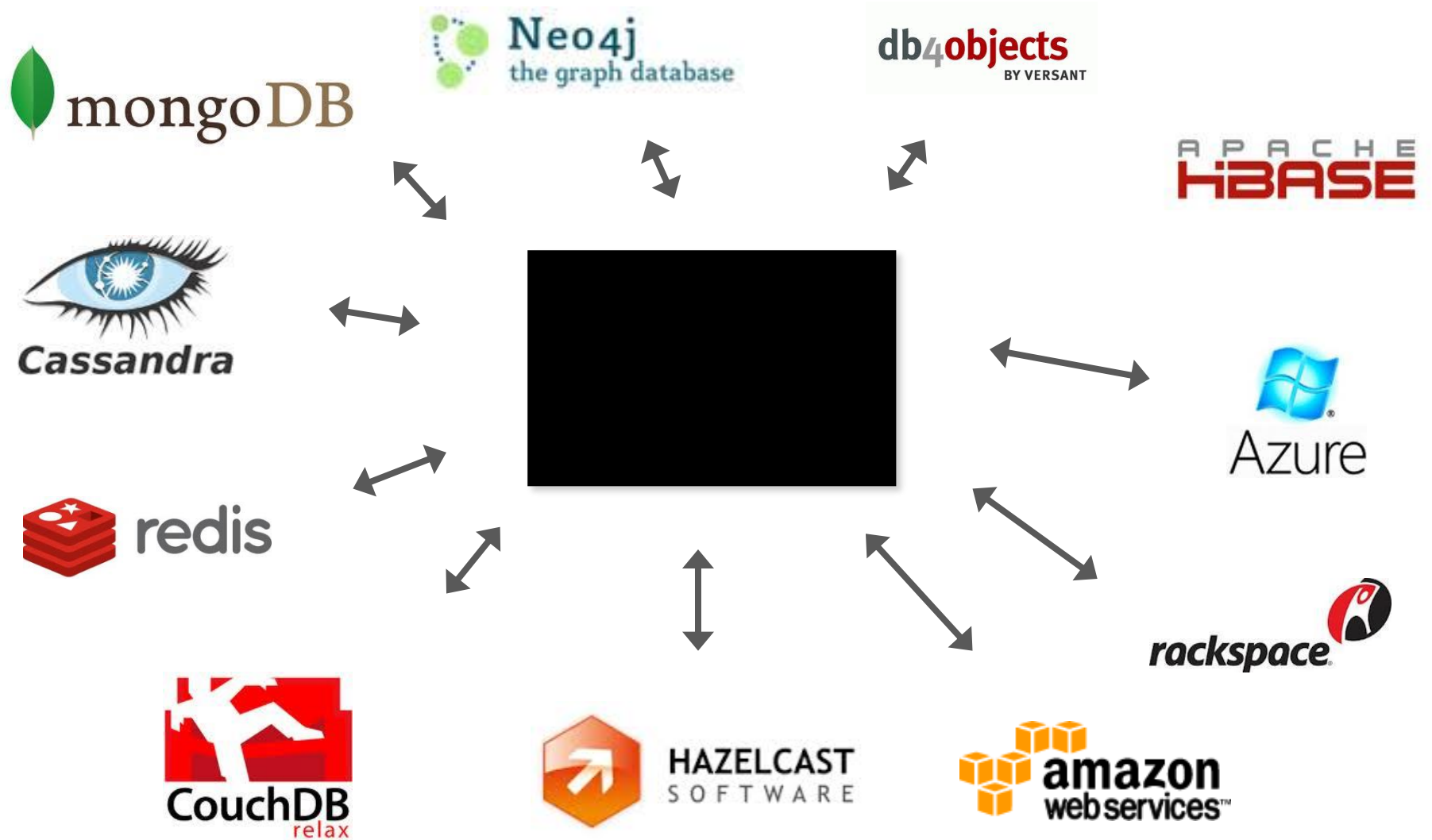
# Wishes for integrators



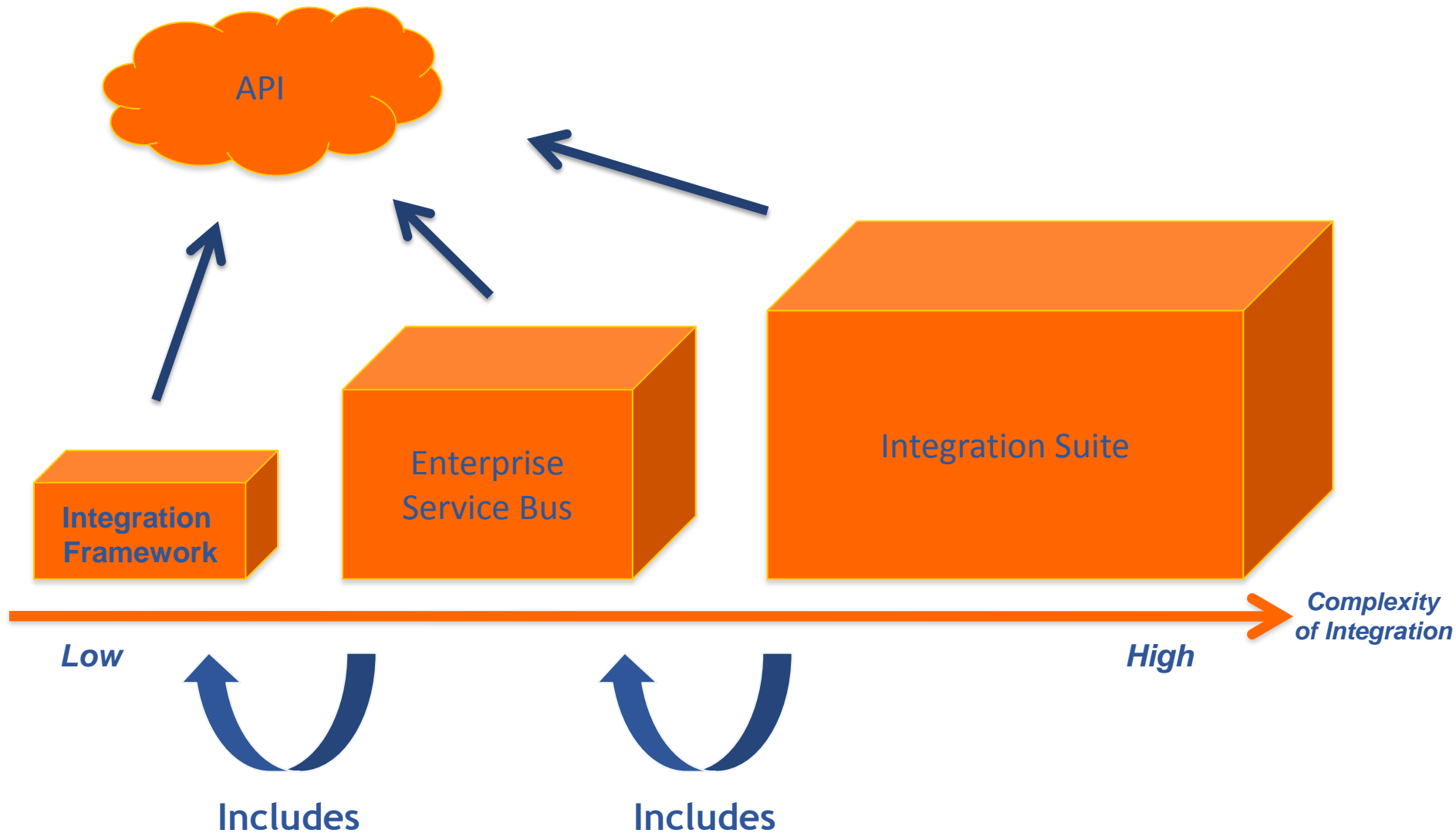
- Standardized Modeling
- Efficient Realization
- Automatic Testing



# Systems integration in the NoSQL era



# Alternatives for Systems Integration

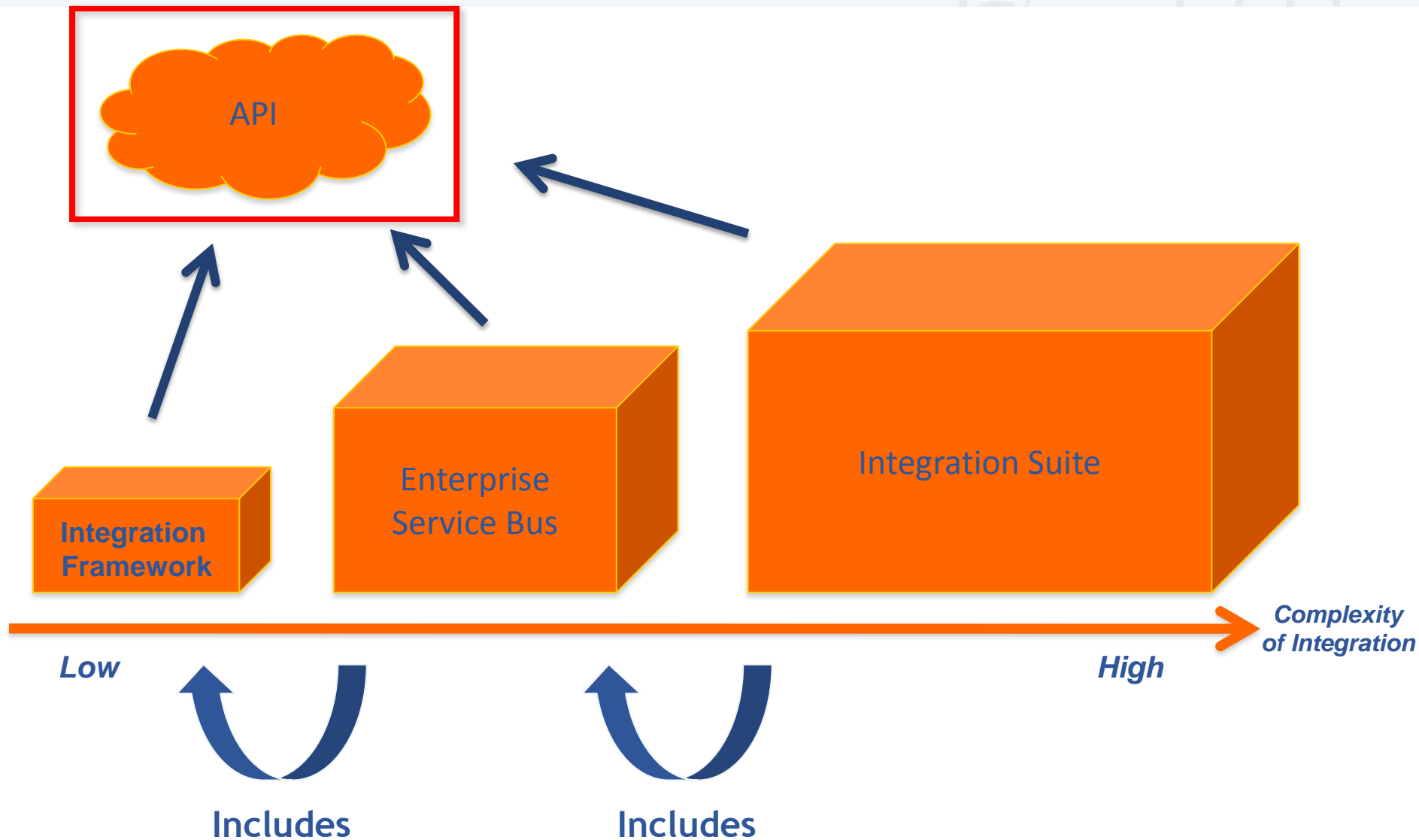




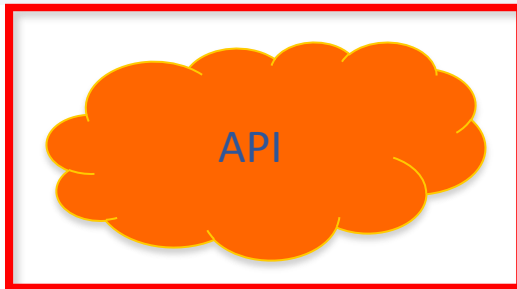
# Agenda

- 1) Introduction to NoSQL
- 2) Systems Integration
- 3) API**
- 4) Integration Framework
- 5) Enterprise Service Bus
- 6) Integration Suite
- 7) Custom Components

# Alternatives for Systems Integration



# Alternatives for Systems Integration



- **Proprietary API**
  - Vendor API
  - Generic API
- **Web Service Interface**
  - REST Web Service
  - SOAP Web Service

## → Vendor APIs



# IaaS - AWS S3 (Java API)

```
AmazonS3 s3 = new AmazonS3Client(new PropertiesCredentials(  
    S3Sample.class.getResourceAsStream("AwsCredentials.properties")));  
  
String bucketName = "my-first-s3-bucket-" + UUID.randomUUID();  
String key = "MyObjectKey";  
  
try {  
  
    s3.createBucket(bucketName);  
    s3.putObject(new PutObjectRequest(bucketName, key, createSampleFile()));  
  
    S3Object object = s3.getObject(new GetObjectRequest(bucketName, key));  
  
    ObjectListing objectListing = s3.listObjects(new ListObjectsRequest()  
        .withBucketName(bucketName)  
        .withPrefix("My"));  
  
    s3.deleteObject(bucketName, key);  
    s3.deleteBucket(bucketName);  
  
} catch (AmazonServiceException ase) {  
    // error handling...  
} catch (AmazonClientException ace) {  
    // error handling...  
}
```

# IaaS - AWS S3 (Ruby API)

```
require File.expand_path(File.dirname(__FILE__) + '/../samples_config')

(bucket_name, file_name) = ARGV
unless bucket_name && file_name
  puts "Usage: upload_file.rb <BUCKET_NAME> <FILE_NAME>"
  exit 1
end

# get an instance of the S3 interface using the default configuration
s3 = AWS::S3.new

# create a bucket
b = s3.buckets.create(bucket_name)

# upload a file
basename = File.basename(file_name)
o = b.objects[basename]
o.write(:file => file_name)

puts "Uploaded #{file_name} to:"
puts o.public_url
```

# IaaS – Microsoft Azure NoSQL Table Storage (C# API)

```
// Retrieve storage account from connection-string
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
    CloudConfigurationManager.GetSetting("StorageConnectionString"));

// Create the table client
CloudTableClient tableClient = storageAccount.CreateCloudTableClient();

// Get the data service context
TableServiceContext serviceContext = tableClient.GetDataServiceContext();

// Create a new customer entity
CustomerEntity customer1 = new CustomerEntity("Harp", "Walter");
customer1.Email = "Walter@contoso.com";
customer1.PhoneNumber = "425-555-0101";

// Add the new customer to the people table
serviceContext.AddObject("people", customer1);

// Submit the operation to the table service
serviceContext.SaveChangesWithRetries();
```

## → Generic Storage APIs

jclouds



sharpbox   
a cloud storage programming interface



# jClouds (Generic API)

## Generic API for IaaS

### JCLOUDS DOCUMENTATION

Below you will find the documentation for jclouds.org including user guides, Examples, FAQs, and References. Find information about jclouds.org, browse all documentation, or help to improve the documentation by [contributing](#).

#### API and Providers

There are many differences between cloud providers. However, there is a common domain among them, and some of them use very similar interfaces (APIs). For instance, Amazon Web Services (AWS) S3 and Google Storage use the same dialect or API.

A **provider** means the real instance and the real endpoint. Google Storage and AWS S3 use the same API (S3 API) but have different properties, e.g. endpoints.

In jclouds structure, there are two different packages API and provider, but they are related to each other.

Our API allows you the freedom to use portable abstractions or cloud-specific features. We support many cloud providers including *Amazon*, *GoGrid*, *Azure*, *vCloud*, and *Rackspace*.

**jclouds provides two abstraction APIs at the moment: Compute and Blobstore.**

- [Compute API](#) helps you bootstrap machines in the cloud.
- [Blobstore API](#) helps you manage key-value data.

#### User Guides

- [Using Blob Store API](#)
- [Using Compute API and Tools](#)
- [Google App Engine](#)

#### Getting Started

- [Installation](#)
- [Examples](#)

#### Quick Start Guides

- [Amazon Web Services](#)
- [Elastic Block Store Models](#)
- [Azure Storage Service](#)
- [BlueLock vCloud](#)
- [Cloud Sigma](#)
- [Eucalyptus](#)
- [File System](#)
- [Go Grid](#)
- [HP Cloud Services](#)
- [IBM Developer Cloud](#)
- [OpenStack](#)
- [Rackspace](#)
- [RimuHosting](#)
- [Terremark eCloud](#)
- [Terremark vCloud Express](#)

# jClouds (Generic API)

## JCLOUDS DOCUMENTATION

Below you will find the documentation for jclouds.org including user guides, Examples, FAQs, and References. Find information about jclouds.org, browse all documentation, or help to improve the documentation by [contributing](#).

### API and Providers

There are many differences between cloud providers. However, there is a common domain among them, and some of them use very similar interfaces (APIs). For instance, Amazon Web Services (AWS) S3 and Google Storage use the same dialect or API.

A **provider** means the real instance and the real endpoint. Google Storage and AWS S3 use the same API (S3 API) but have different properties, e.g. endpoints.

In jclouds structure, there are two different packages API and provider, but they are related to each other.

Our API allows you the freedom to use portable abstractions or cloud-specific features. We support many cloud providers including *Amazon*, *GoGrid*, *Azure*, *vCloud*, and *Rackspace*.

**jclouds provides two abstraction APIs at the moment: Compute and Blobstore.**

- [Compute API helps you bootstrap machines in the cloud.](#)
- [Blobstore API helps you manage key-value data.](#)

### User Guides

- [Using Blob Store API](#)
- [Using Compute API and Tools](#)
- [Google App Engine](#)

### Getting Started

- [Installation](#)
- [Examples](#)

### Quick Start Guides

- [Amazon Web Services](#)
- [Elastic Block Store Models](#)
- [Azure Storage Service](#)
- [BlueLock vCloud](#)
- [Cloud Sigma](#)
- [Eucalyptus](#)
- [File System](#)
- [Go Grid](#)
- [HP Cloud Services](#)
- [IBM Developer Cloud](#)
- [OpenStack](#)
- [Rackspace](#)
- [RimuHosting](#)
- [Terremark eCloud](#)
- [Terremark vCloud Express](#)



**Blobstore API**

# jClouds (Generic API)

## JCLOUDS DOCUMENTATION

Below you will find the documentation for jclouds.org including user guides, Examples, FAQs, and References. Find information about jclouds.org, browse all documentation, or help to improve the documentation by [contributing](#).

### API and Providers

There are many differences between cloud providers. However, there is a common domain among them, and some of them use very similar interfaces (APIs). For instance, Amazon Web Services (AWS) S3 and Google Storage use the same dialect or API.

A **provider** means the real instance and the real endpoint. Google Storage and AWS S3 use the same API (S3 API) but have different properties, e.g. endpoints.

In jclouds structure, there are two different packages API and provider, but they are related to each other.

Our API allows you the freedom to use portable abstractions or cloud-specific features. We support many cloud providers including *Amazon*, *GoGrid*, *Azure*, *vCloud*, and *Rackspace*.

**jclouds provides two abstraction APIs at the moment: Compute and Blobstore.**

- [Compute API](#) helps you bootstrap machines in the cloud.
- [Blobstore API](#) helps you manage key-value data.

### User Guides

- [Using Blob Store API](#)
- [Using Compute API and Tools](#)
- [Google App Engine](#)

### Getting Started

- [Installation](#)
- [Examples](#)

### Quick Start Guides

- [Amazon Web Services](#)
- [Elastic Block Store Models](#)
- [Azure Storage Service](#)
- [BlueLock vCloud](#)
- [Cloud Sigma](#)
- [Eucalyptus](#)
- [File System](#)
- [Go Grid](#)
- [HP Cloud Services](#)
- [IBM Developer Cloud](#)
- [OpenStack](#)
- [Rackspace](#)
- [RimuHosting](#)
- [Terremark eCloud](#)
- [Terremark vCloud Express](#)

Several different  
NoSQL providers  
supported



# jClouds (Generic API) – AWS S3 Blobstore (Java)

```
// get a context with amazon that offers the portable BlobStore API  
BlobStoreContext context = new BlobStoreContextFactory().  
    createContext("aws-s3", accesskeyid, secretkey);
```

```
// create a container in the default location  
BlobStore blobStore = context.getBlobStore();  
blobStore.createContainerInLocation(null, bucket);
```

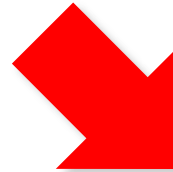
```
// add blob  
Blob blob = blobStore.newBlob("test");  
blob.setPayload("test data");  
blobStore.putBlob(bucket, blob);
```

```
// when you need access to s3-specific features,  
// use the provider-specific context  
AWSS3Client s3Client =  
    AWSS3Client.class.cast(context.getProviderSpecificContext().getApi());
```

```
// make the object world readable  
String publicReadWriteObjectKey = "public-read-write-acl";  
S3Object object = s3Client.newS3Object();
```

```
object.getMetadata().setKey(publicReadWriteObjectKey);  
object.setPayload("hello world");  
s3Client.putObject(bucket, object, withAcl(CannedAccessPolicy.PUBLIC_READ));
```

```
context.close();
```



Use another blobstore?  
Just change this line!

## → Generic NoSQL specific APIs



## Kundera

# EclipseLink (JPA) NoSQL Support

As of EclipseLink 2.4, EclipseLink has **added JPA support for NoSQL databases**, initially with support for MongoDB and Oracle NoSQL.

EclipseLink's NoSQL support allows the JPA API and JPA annotations/xml to be used with NoSQL data. EclipseLink also supports several NoSQL specific annotations/xml including `@NoSQL` that defines a class to map NoSQL data.

```
@Entity
@NoSql(dataFormat=DataFormatType.MAPPED)
public class Order
```

```
@Id
@GeneratedValue
@Field(name="_id")
private String id;
```

```
@Basic
private String description;
@Basic
private double totalCost = 0;
@Embedded
private Address billingAddress;
@Embedded
private Address shippingAddress;
@ElementCollection
private List<OrderLine> orderLines = new ArrayList<OrderLine>();
@ManyToOne(fetch=FetchType.LAZY)
private Customer customer;
```

```
Query query = em.createQuery("Select o from Order o where o.totalCost > 1000");
List<Order> orders = query.getResultList();
```

<http://wiki.eclipse.org/EclipseLink/Examples/JPA/NoSQL>

# Hibernate OGM (Object/Grid Mapper)

Hibernate Object/Grid Mapper (OGM) aims at providing Java Persistence (JPA) support for NoSQL solutions. It reuses Hibernate Core's engine but persists entities into a NoSQL data store instead of a relational database. It reuses the Java Persistence Query Language (JP-QL) to search their data.

That's the grand scheme of things and we will roll out functionalities over time. The short term target is:

- support for Infinispan (done)
- support Hibernate Search full-text queries (done)
- support simple JP-QL queries (restrictions and many-to-one joins)

The medium target is:

- support other key/value stores
- support other NoSQL families
- support complex joins and aggregations



<http://www.hibernate.org/subprojects/ogm.html>



The idea behind Kundera is to make working with NoSQL Databases drop-dead simple and fun. Kundera is being developed with following objectives:

- To make working with NoSQL as simple as working with SQL
- To serve as JPA Compliant mapping solution for NoSQL Datastores.
- To help developers, forget the complexity of NoSQL stores and focus on Domain Model.
- To make switching across data-stores as easy as changing a configuration.

## Currently Supported Datasources

---

- Cassandra
- MongoDB
- HBase
- Redis
- OracleNoSQL
- Neo4j
- Relational databases

<https://github.com/impetus-opensource/Kundera>



## → REST / SOAP APIs



# SaaS – Salesforce (REST API)

GET AVAILABLE OBJECTS / RECENT / QUERY / SEARCH:

```
curl -H 'Authorization: OAuth XYZ' -H "X-PrettyPrint:1"  
https://na14.salesforce.com/services/data/v20.0/recent
```

#####

QUERY EXAMPLE:

```
curl -H 'Authorization: OAuth XYZ' -H "X-PrettyPrint:1"  
https://na14.salesforce.com/services/data/v20.0/query?q=SELECT+name+from+Article__c
```

#####

UPDATE:

```
curl -H 'Authorization: OAuth ' -H "X-PrettyPrint:1" -H "Content-Type: application/json" --data-binary  
@test.json -X PATCH  
https://na14.salesforce.com/services/data/v20.0/subjects/Article__c/a00d0000002NJ0sAAG
```



- SOAP
- REST

# SaaS – Salesforce (REST API)

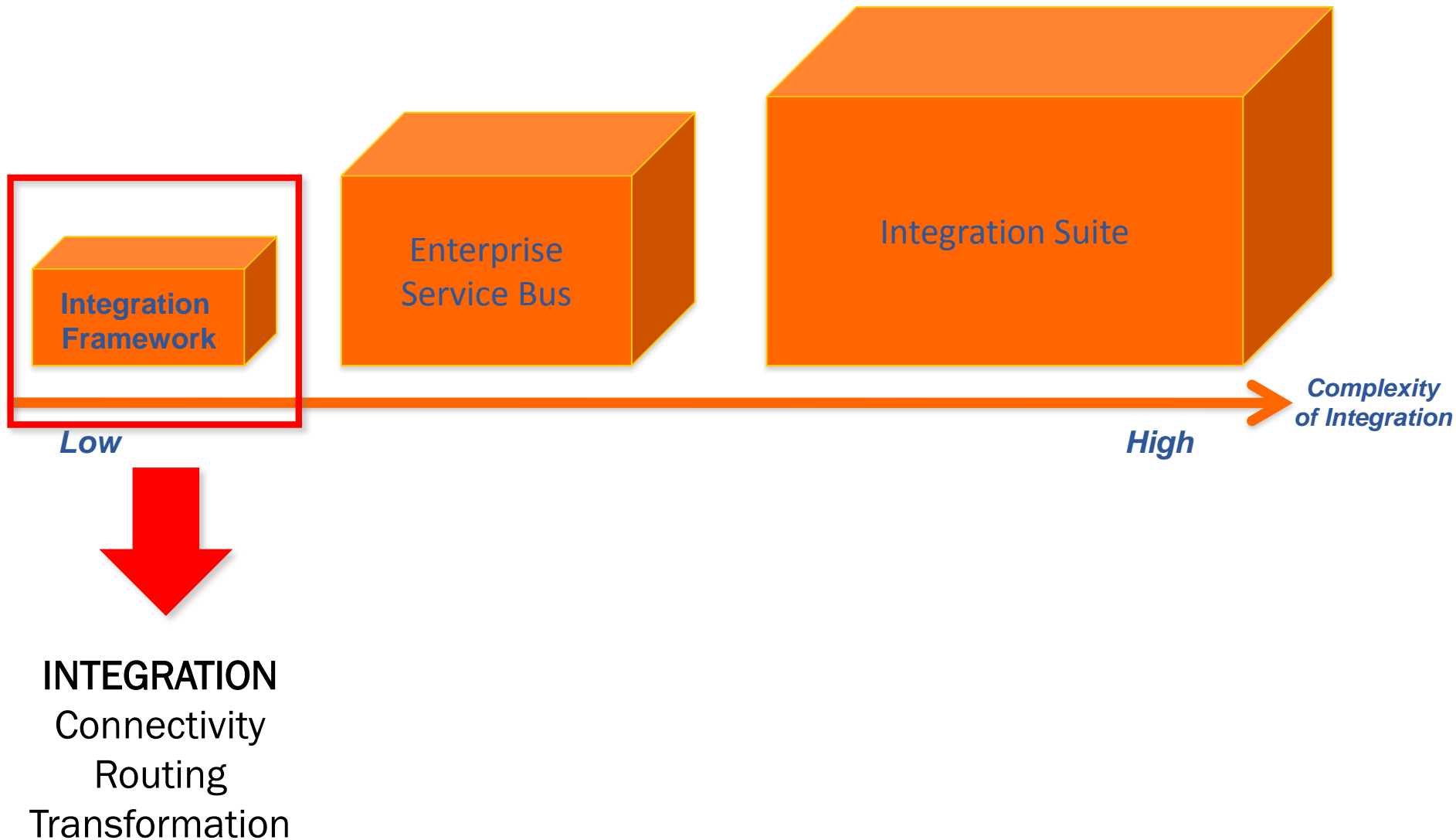


- SOAP
- REST

# Agenda

- 1) Introduction to NoSQL
- 2) Systems Integration
- 3) API
- 4) Integration Framework**
- 5) Enterprise Service Bus
- 6) Integration Suite
- 7) Custom Components

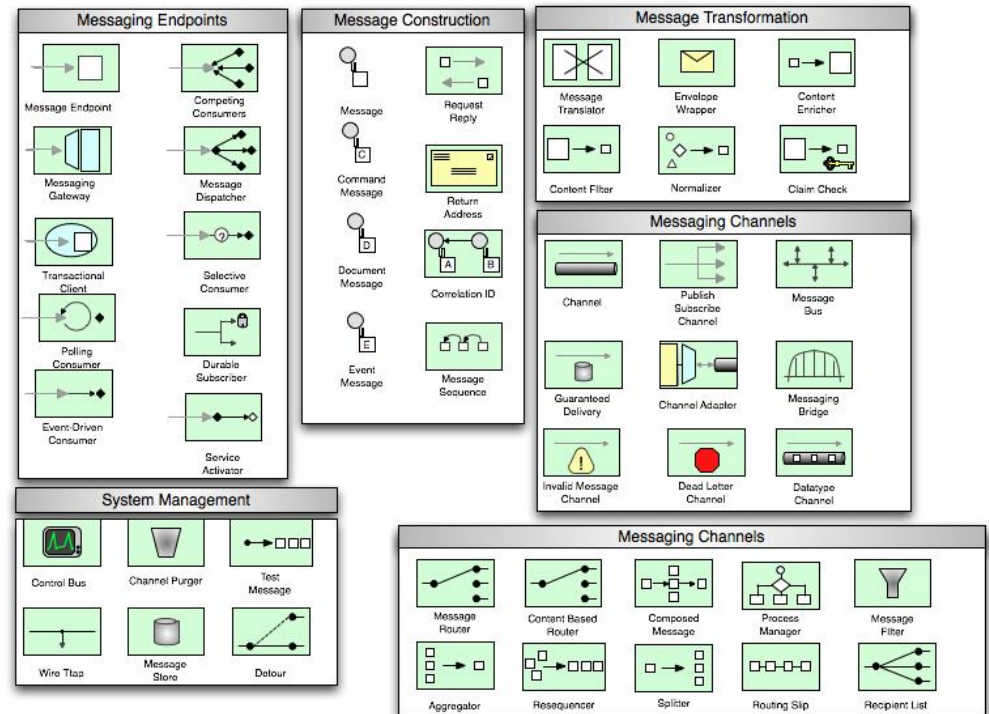
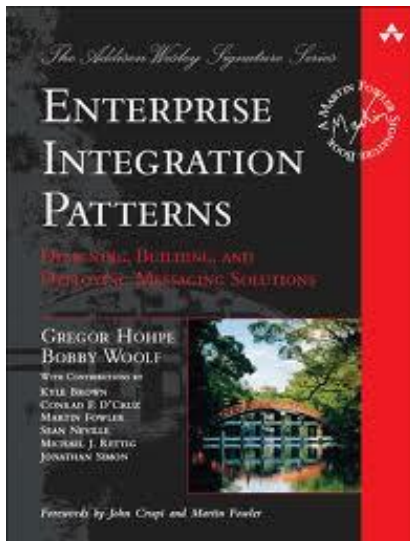
# Alternatives for Systems Integration



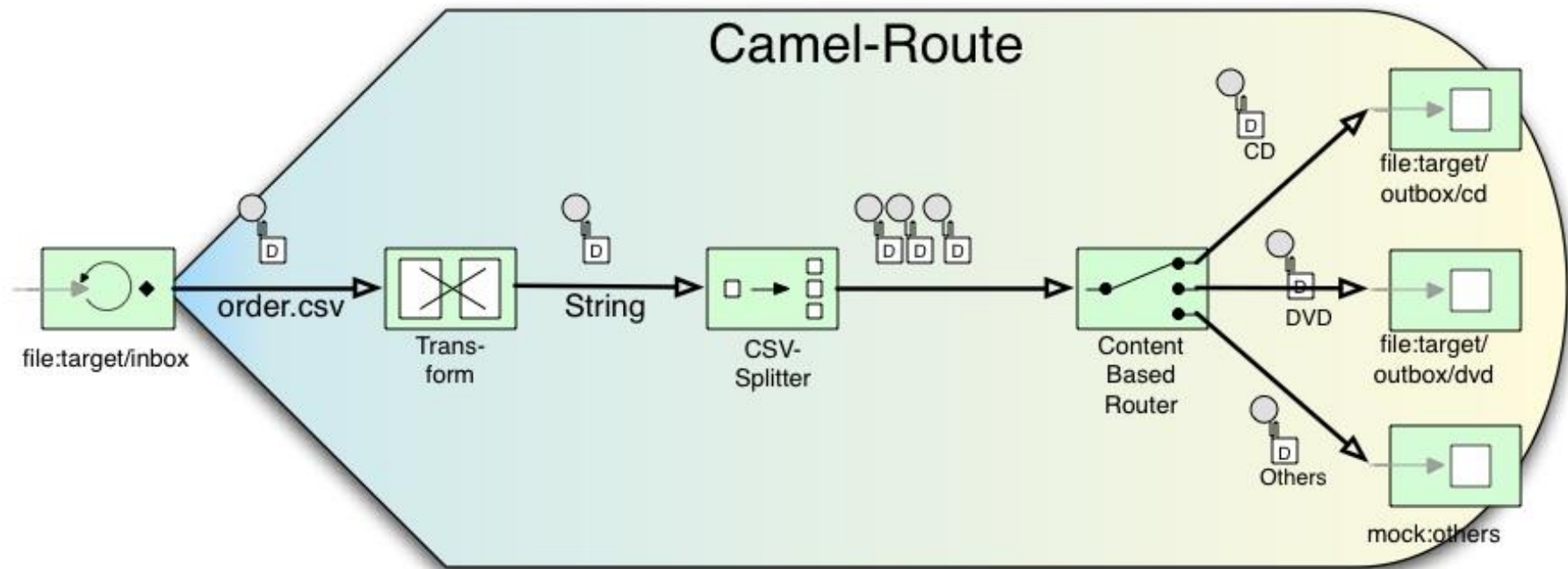
# Integration Frameworks



# Enterprise Integration Patterns



# Enterprise Integration Patterns





# Deployment

Standalone

Cloud

Spring  
Container

Web Container

Application  
Server



# No longer „glue code“

```
AmazonS3 s3 = new AmazonS3Client(new PropertiesCredentials(
    S3SampleClass.getResourceAsStream("AwsCredentials.properties")));

String bucketName = "my-first-s3-bucket-" + UUID.randomUUID();
String key = "MyObjectKey";

try {

    s3.createBucket(bucketName);
    s3.putObject(new PutObjectRequest(bucketName, key, createSampleFile()));

    S3Object object = s3.getObject(new GetObjectRequest(bucketName, key));

    ObjectListing objectListing = s3.listObjects(new ListObjectsRequest()
        .withBucketName(bucketName)
        .withPrefix("My"));

    s3.deleteObject(bucketName, key);
    s3.deleteBucket(bucketName);

} catch (AmazonServiceException ase) {
    // error handling...
} catch (AmazonClientException ace) {
    // error handling...
}
```

# Domain Specific Language (Camel)

// Producer

```
from("ftp:toS3")  
    .setHeader(S3Constants.KEY, simple("order.txt"))  
    .to("aws-s3://myBucket?accessKey=" + a + "&secretKey=" + s)
```

// Consumer

```
from(„salesforce://orders__c?user=dummy1“)  
    .filter(„attributeType==,dvd“)  
    .to(„ibm-database:orderData“)
```

# NoSQL with an Integration Framework



# Document-oriented database



# Document-oriented database



- 10gen
- stores structured data as JSON-like documents with dynamic schemas
- REST API and several SDKs (Java, .NET, Ruby, PHP, Python, etc.)
- Ad hoc queries, indexing, replication, load balancing
- Powerful, but also easy to use and flexible
- Example: Disney persists state information of online games in a common object repository.

# Code example: MongoDB Java Driver

```
// connect to the local database server
MongoClient mongoClient = new MongoClient();

// get handle to "mydb"
DB db = mongoClient.getDB("mydb");

// Authenticate - optional
// boolean auth = db.authenticate("foo", "bar");

// get a list of the collections in this database and print them out
Set<String> collectionNames = db.getCollectionNames();
for (String s : collectionNames) {
    System.out.println(s);
}

// get a collection object to work with
DBCollection testCollection = db.getCollection("testCollection");

// drop all the data in it
testCollection.drop();

// make a document and insert it
BasicDBObject doc = new BasicDBObject("name", "MongoDB").append("type", "database").append("count", 1)
    .append("info", new BasicDBObject("x", 203).append("y", 102));

testCollection.insert(doc);

// get it (since it's the only one in there since we dropped the rest earlier on)
DBObject myDoc = testCollection.findOne();
System.out.println(myDoc);
```

# Code example: camel-mongodb component

**// Producer**

```
from("jms:FlightDocumentQueue")  
    .to("mongodb:myDb?database=flights  
        &collection=tickets  
        &operation=insert");
```

**// Consumer**

```
from("mongodb:myDb?database=flights  
    &collection=cancellations  
    &tailTrackIncreasingField=departureTime")  
    .to("jms:CancelledFlightsQueue");
```



# Live demo



Integration of a document-oriented database in action...

# Some more...



# Key-Value database



# Code example: camel-aws component

// Producer

```
from(„jms:toS3Queue")  
    .setHeader(S3Constants.KEY, simple("order.txt"))  
    .to("aws-s3://myBucket?accessKey=" + a + "&secretKey=" + s)
```

// Consumer

```
from("aws-s3://myBucket?accessKey=" + a + "&secretKey=" + s)  
    .to("log:S3logging")
```

# Code example: camel-jclouds component

```
from("direct:toJcloudsAwsS3")  
    .setHeader(JcloudsConstants.BLOB_NAME, "jclouds-demo-tutorial.txt")  
    .setHeader(JcloudsConstants.CONTAINER_NAME, "kw-s3-data")  
    .to("jclouds:blobstore:aws-s3")
```

```
from("direct:toJcloudsMicrosoftAzure")  
    .setHeader(JcloudsConstants.BLOB_NAME, "jclouds-demo-tutorial.txt")  
    .setHeader(JcloudsConstants.CONTAINER_NAME, "kw-s3-data")  
    .to("jclouds:blobstore:azureblob")
```

# Graph-oriented database



# Code example: camel-neo4j component

// Producer

```
from("jms:createNewNeo4jNode")  
    .to("neo4j:http://Neo4jServer:7474/data");
```

// Consumer

```
from(„neo4j://todo)...
```

*Not implemented in current Camel release (2.11) ☹️ **TODO 2.12?***

*→ Use Camel's REST components (shown in some minutes...)*

# In-memory database





# Code example: camel-hazelcast component

// Producer

```
from("direct:add")  
    .setHeader(HazelcastConstants.OPERATION, „add“)  
    .to("hazelcast:queue:foo");
```

// Consumer

```
from("hazelcast:queue:foo")  
    .log("content of object foo: ${body}");
```

# Column-oriented database



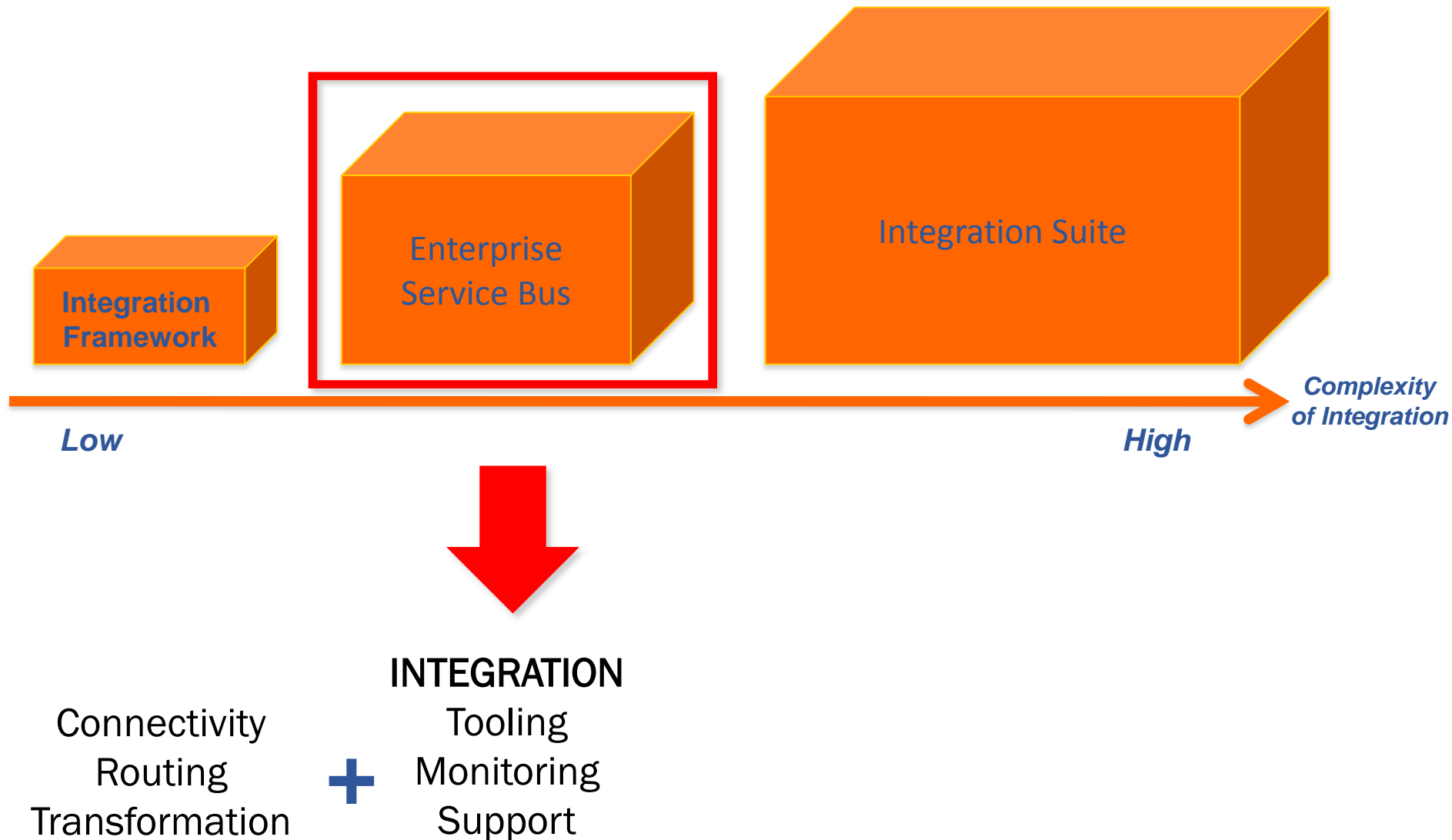
# Code example: camel-hbase component

```
<route>
  <from uri="direct:in" />
  <!-- Set the HBase Row -->
  <setHeader headerName="CamelHBaseRowId">
    <el>${in.body.id}</el>
  </setHeader>
  <!-- Set the HBase Value -->
  <setHeader headerName="CamelHBaseValue">
    <el>${in.body.value}</el>
  </setHeader>
  <to uri="hbase:mytable?operation=CamelHBasePut&family=myfamily&qualifier=myqualifier" />
</route>
```

# Agenda

- 1) Introduction to NoSQL
- 2) Systems Integration
- 3) API
- 4) Integration Framework
- 5) Enterprise Service Bus**
- 6) Integration Suite
- 7) Custom Components

# What is an Enterprise Service Bus (ESB)?



# Oracle Service Bus

Oracle Technology Network > Middleware > Service Bus > Overview

Fusion Middleware Home

AIA Foundation Pack

Business Intelligence

Coherence

Developer Tools

Event-Driven Architecture

GlassFish Server

Identity Management

JRockit

SOA Suite

TopLink

Tuxedo

WebCenter

WebCenter Content

WebCenter Sites

WebCenter Portal

Social Network

WebLogic Server

Event Processing

Overview

Downloads

Documentation

Customers

Learn More

**ORACLE**  
SERVICE BUS

## Oracle Service Bus Overview

Oracle Service Bus transforms complex and brittle architectures into agile integration networks by connecting, mediating, and managing interactions between services and applications. Oracle Service Bus delivers low-cost, standards-based integration for mission critical SOA environments where extreme performance and scalability are requirements.

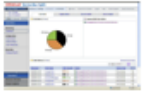

Oracle Service Bus 11g extends the performance and scalability leadership of [Oracle SOA Suite 11g](#) and [Event Driven Architecture Suite](#).

Oracle Service Bus 11g introduces new capabilities such as Service Result Cache, Automated Lifecycle Service Governance, and improved performance and availability for organizations using enterprise datacenters, as well as mobile and cloud environments.

### Overview

Getting Started with Oracle Service Bus

If you are new to Oracle Service Bus or simply want to brush up on a new feature, watch these short videos for a quick lesson.



Click for larger view

# Three IBM ESBs TODO Rebranded ???

IBM Software > WebSphere >

## WebSphere Enterprise Service Bus

ESB for quick integration of applications and processes

 [Add to My interests](#)



**Smart SOA Solution  
WESB Registry & Repository**  
→ [Download Redbook](#)

### Overview

IBM WebSphere Enterprise Service Bus  
time-to-value across your organization

IBM Software > WebSphere > WebSphere Message Broker product Line >

## WebSphere Message Broker

A simple yet powerful ESB for any size project

 [Add to My interests](#)



**Connecting Your Business Using  
WebSphere Message Broker  
as an ESB**  
→ [Download Redbook](#)

### Overview

WebSphere Message Broker is a  
of data sources from a wide range  
environments.

IBM Software > WebSphere >

## WebSphere DataPower SOA Appliances



**Analyst Report: A Competitive  
review of SOA Appliances**  
→ [Get the white paper](#)



**DataPower SOA Appliances  
strategic overview**  
→ [Download Redbook](#)

### What we offer


#### WebSphere DataPower Integration Appliance XI52

The XI52 is a security & integration gateway  
appliance, built for simplified deployment &  
hardened security, bridging multiple protocols  
& performing conversions at wire speed.

#### WebSphere DataPower Integration Appliance XI50B & XI50z

The XI50B, and XI50z deliver common  
message transformation, integration, and  
routing in a blade form factor cuts costs and  
improves performance.

# Mule ESB

**MuleSoft™**  
*connecting the new enterprise*

CommunityBlogSupport loginContact 1-866-228-9989

ProductsSolutionsServicesResourcesPartnersCompany


Home > Products > Mule ESB

## Mule ESB

The world's most widely used integration platform for connecting applications

- 3,200 companies in production
- 35% of the Global 500
- 103,000+ developers

Download



The diagram illustrates the Mule ESB acting as a central hub for integration. It connects various external services and internal systems. At the top, it connects to Web services, Email, REST, and Portal / CMS. It also integrates with a Cloud Platform (AWS, PaaS, Google). Below the Mule ESB logo, it connects to Enterprise Systems (SAP, IBM), Custom Apps, and Databases.

OverviewFeaturesConnectorsSupport



## ESB

[Home](#) ► [Products](#)



### Reliable, Scalable and Secure

Talend's reliable and scalable enterprise service bus (ESB) products simplify the connection, mediation and management of services and applications so more time is spent making decisions instead of integrating systems.

[Download](#)[Request Info](#)



**FuseSource**  
A Progress Software Company



Three ESBs (JBoss ESB, Switchyard, Fuse ESB)  
Two BPMs (jBPM, Polymita)  
No unified platform (yet)

# Many important players ...



Gartner Quadrant 2013 for Application Integration

# NoSQL with an Enterprise Service Bus



[Download](#)[About Us](#)[Blog](#)[Contact](#)[My account](#)[Products](#)[Solutions](#)[Resources](#)[Services](#)[Ecosystem](#)

## ESB

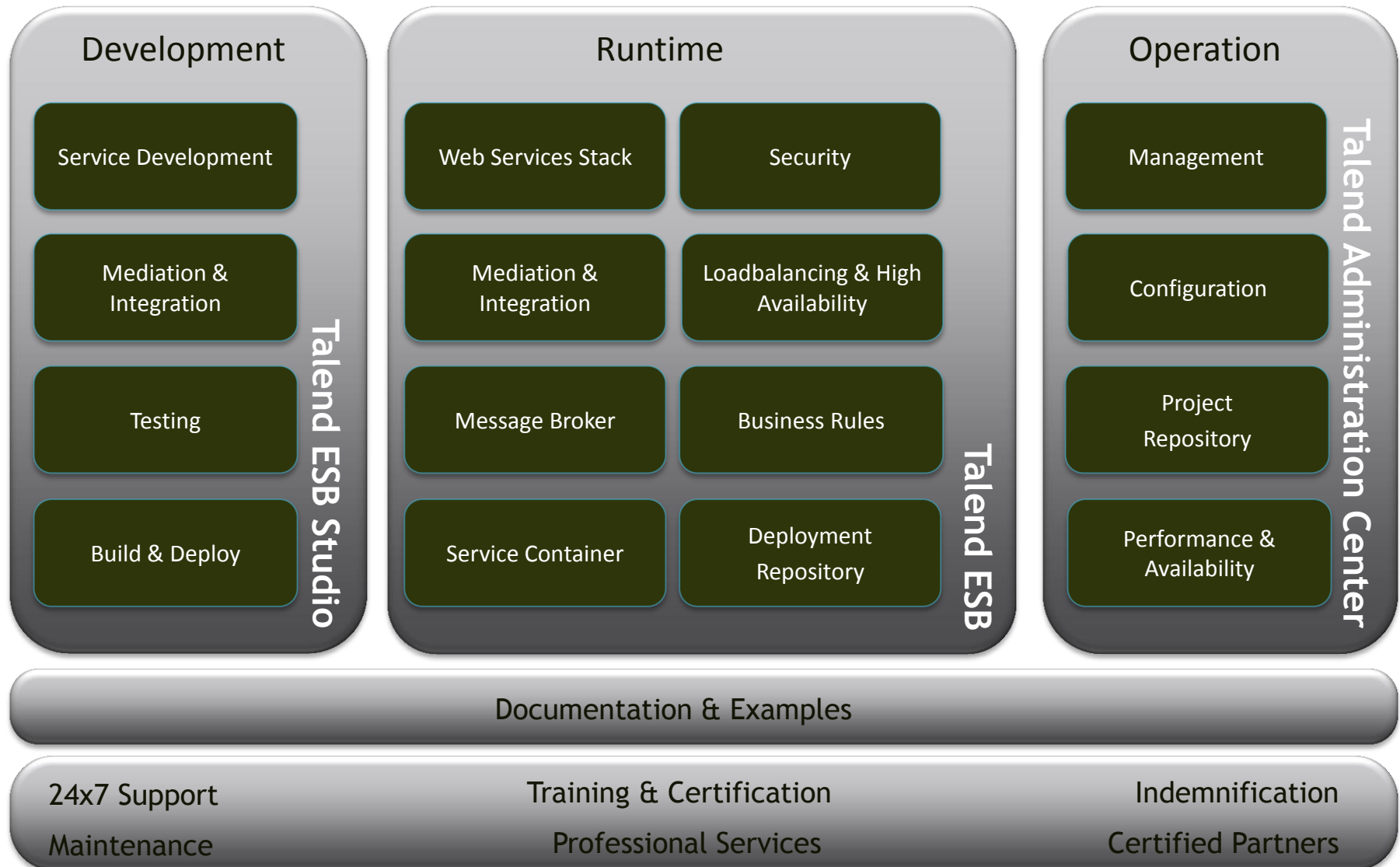
[Home](#) ► [Products](#)

### Reliable, Scalable and Secure

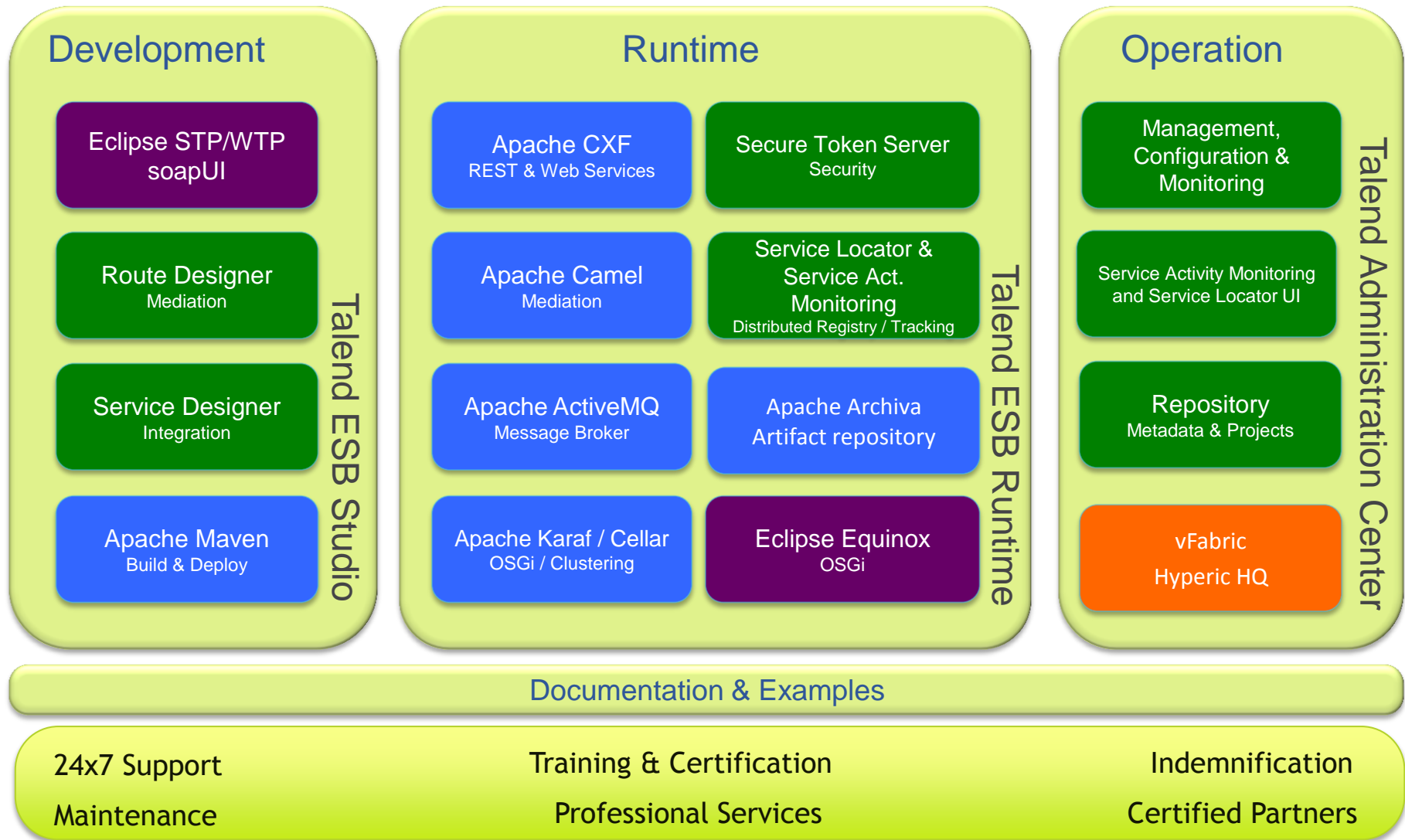
Talend's reliable and scalable enterprise service bus (ESB) products simplify the connection, mediation and management of services and applications so more time is spent making decisions instead of integrating systems.

[Download](#)[Request Info](#)

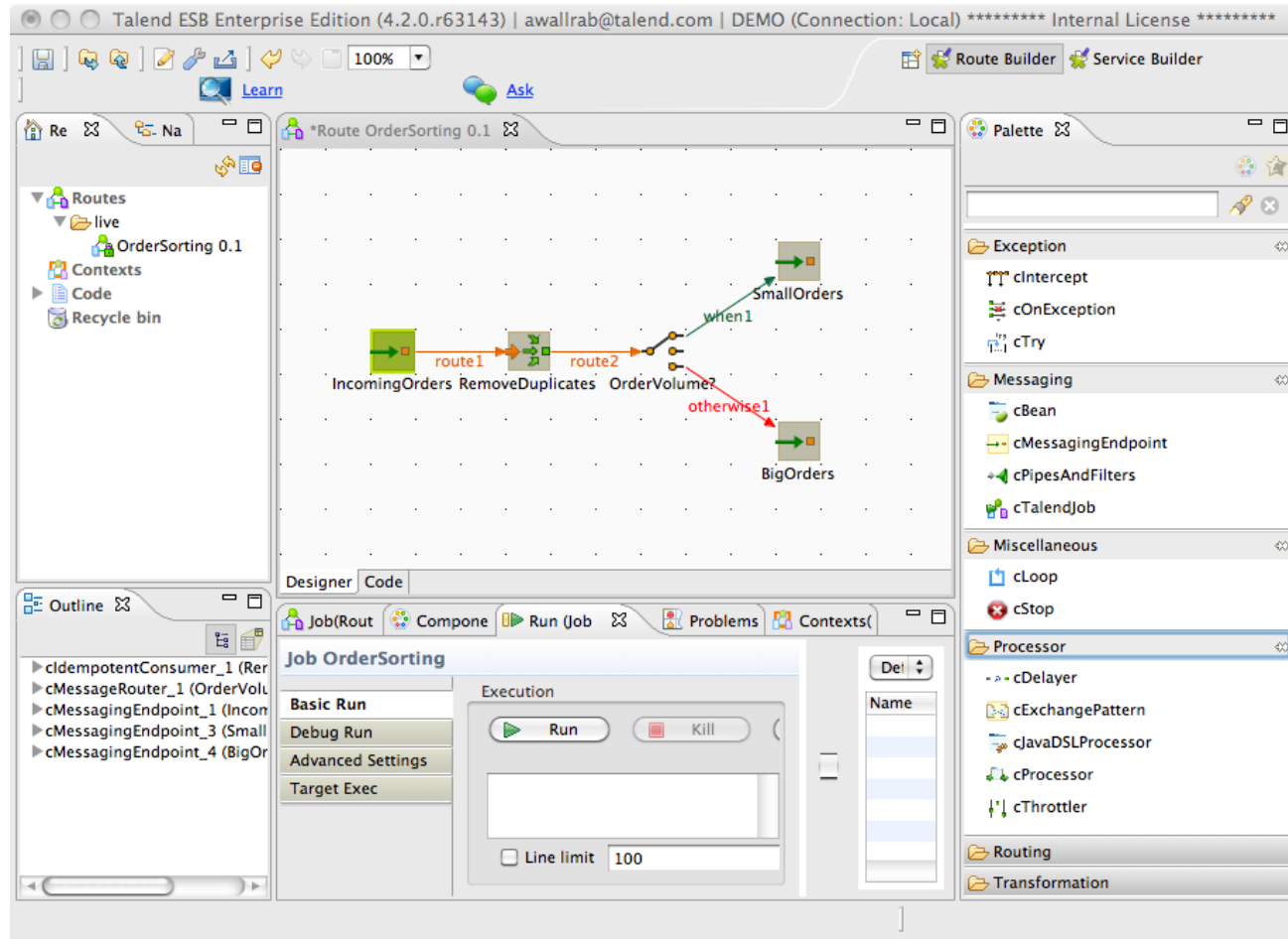
# Tooling on top of Camel: Talend ESB



# Tooling on top of Camel: Talend ESB



# Talend ESB Studio



## Route Builder

- Endpoints
- EIPs
- Processors
- Custom components

## Configuration

- Components
- Endpoints

## Code Generation

- 100% Java
- Camel Code
- Packaged as OSGi Bundles

## Execution in the IDE

- Debugging
- Live statistics
- Short dev cycles



# Live demo

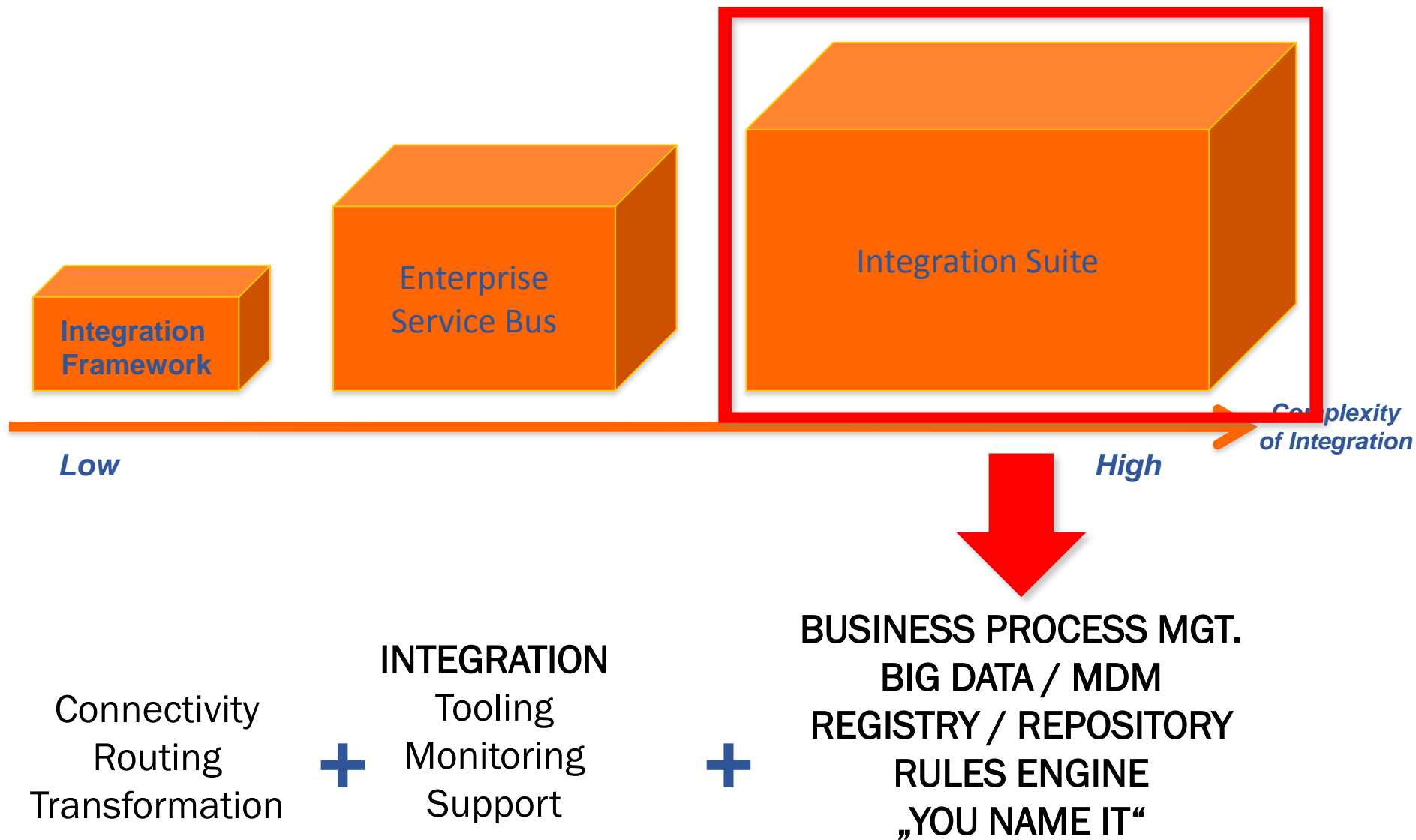


Integration of a key-value database in action...

# Agenda

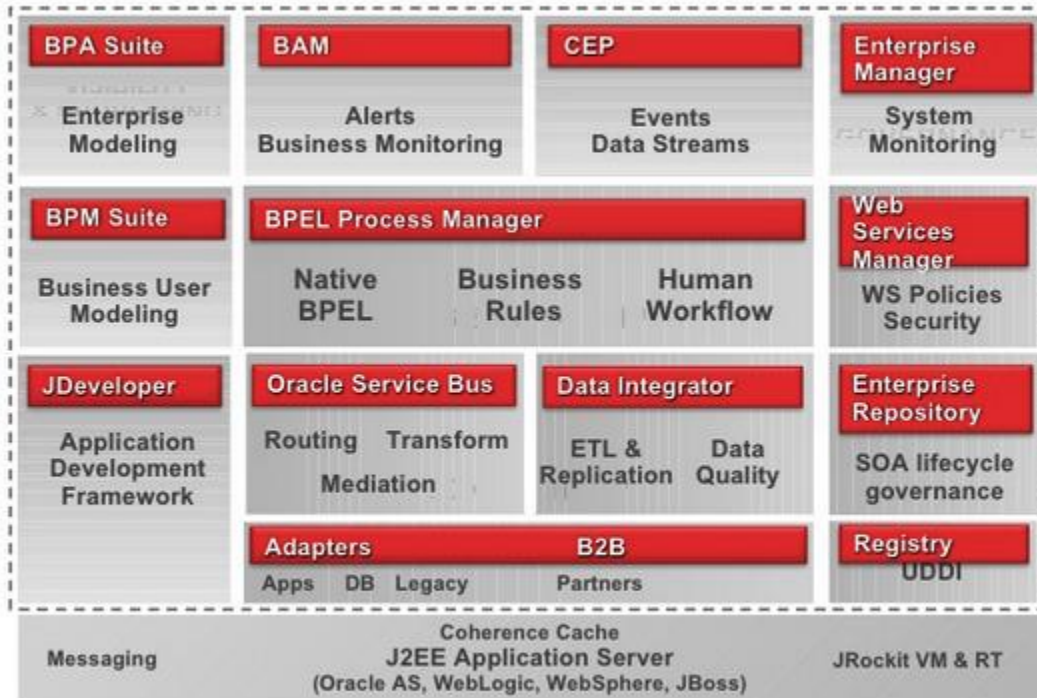
- 1) Introduction to NoSQL
- 2) Systems Integration
- 3) API
- 4) Integration Framework
- 4) Enterprise Service Bus
- 6) Integration Suite**
- 7) Custom Components

# What is an Integration Suite?



# Oracle Platform

## Oracle SOA Suite



ORACLE

# IBM WebSphere Stack

## IBM WebSphere software

See also: [List of IBM WebSphere products](#)

The following co

IBM has also cla

### Application

#### Main Product

- WebSphere
- WebSphere
- IBM Workloa
- WebSphere

#### All products

Application found

- Application F
- Debug Tool f
- Embedded V
- Fault Analyze
- File Manager
- IBM XWork S
- WebSphere

- Applicatio
- Express I
- Commun

#### • Network Deployment

- WebSphere
- WebSphere
- WebSphere
- WebSphere
- Workload S

Application opti

- IBM Expres
- WebSphere
- for Hybr
- for Publi
- WebSphere
- for Deve

WebSphere

- WebSphere
- WebSphere
- WebSphere
- WebSphere
- WebSphere
- WebSphere

Cloud platform

- IBM Worklo
- WebSphere
- Hypervis
- WebSphere
- Extende

#### • for z/OS

• WebSphe

Elastic cachi

- WebSphe
- WebSphe

Transaction

- CICS Ex
- CICS To

- CICS
- CICS
- CICS
- CICS
- CICS
- CICS

- CICS
- CICS
- CICS
- CICS
- CICS
- CICS

- CICS
- CICS
- CICS
- CICS
- CICS
- CICS

- CICS
- CICS
- CICS
- CICS
- CICS
- CICS

- CICS Tra
- Desk
- for M

- for z/
- CICS Tra

- for z/
- for z/
- IBM CL/C
- IBM CL/S
- TXSeries

#### • WebSphere Transaction Cluster Facility

### Application infrastructure areas of focus

Operational decision management

- WebSphere
- WebSphere
- WebSphere
- WebSphere
- WebSphere

- WebSp
- WebSp
- WebSp
- WebSp

- WebSp
- WebSp
- WebSp

- WebSphere
- WebSphere
- WebSphere

- WebSp
- WebSp
- WebSp

- WebSp
- WebSp
- WebSp

- WebSphere
- Operat

Process autom

- IBM Busin
- IBM Integr
- WebSphere
- WebSphere

- WebSphere

### Conne

### Mobile

### All prod

Mobile ap

- IBM M
- Ration

- for
- St
- Ra

Mobile int

- IBM C
- IBM S
- IBM V

- Lotus
- WebS
- M

# WSO2 Carbon Platform



- WSO2 API Manager
- WSO2 Application Server
- WSO2 Business Activity Monitor
- WSO2 Business Process Server
- WSO2 Business Rules Server
- WSO2 Complex Event Processor
- WSO2 Data Services Server
- WSO2 Elastic Load Balancer
- WSO2 Enterprise Service Bus
- WSO2 Gadget Server
- WSO2 Governance Registry
- WSO2 Identity Server
- WSO2 Mashup Server
- WSO2 Message Broker



# Talend Unified Platform



- Commercial license
- Subscription model
- Support included

- Open source license
- Free of charge
- Optional support
- Based on open source projects such as Eclipse or Apache Camel, CXF, Hadoop



## Integration Suite in Action...



# ESB Vendor == Integration Suite Vendor



Proprietary



Open Source



# ESB Vendor == Integration Suite Vendor



**ESB**



**BPM**



**Big Data**

# Custom Combination of ESB, BPM, etc.



- A lot of glue code
- Testing
- Bugfixing
- No support

Some other people already had the problems you would have!

# Agenda

- 1) Introduction to NoSQL
- 2) Systems Integration
- 3) API
- 4) Integration Framework
- 5) Enterprise Service Bus
- 6) Integration Suite
- 7) Custom Components**

# Custom NoSQL components



Easy to realize for all  
integration alternatives \*

- Integration Framework
- Enterprise Service Bus
- Integration Suite

*\* At least for open source solutions*

# Live demo (Example: Apache Camel)



Custom NoSQL components in action...

# Alternative for custom NoSQL components



Sluggish Boy ???



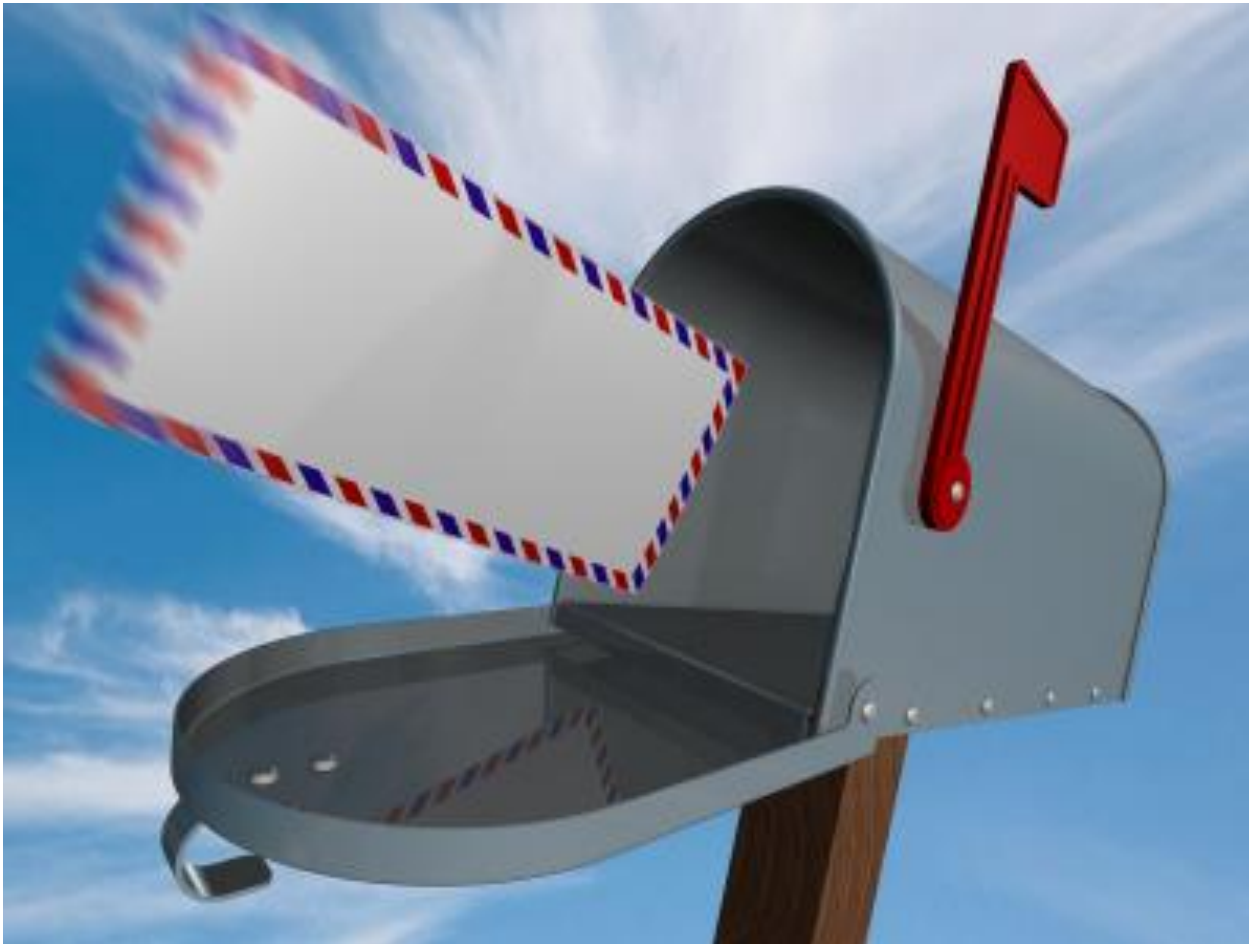
# Code example: REST API for Salesforce object store

```
// Salesforce Query (SOQL) via REST API
from("direct:salesforceViaHttpLIST")
    .setHeader("X-PrettyPrint", 1)
    .setHeader("Authorization", accessToken)
    .setHeader(Exchange.CONTENT_TYPE, "application/json")
.to("https://na14.salesforce.com/services/data/v20.0/query?q=SELECT+name+from
    +Article__c")
```

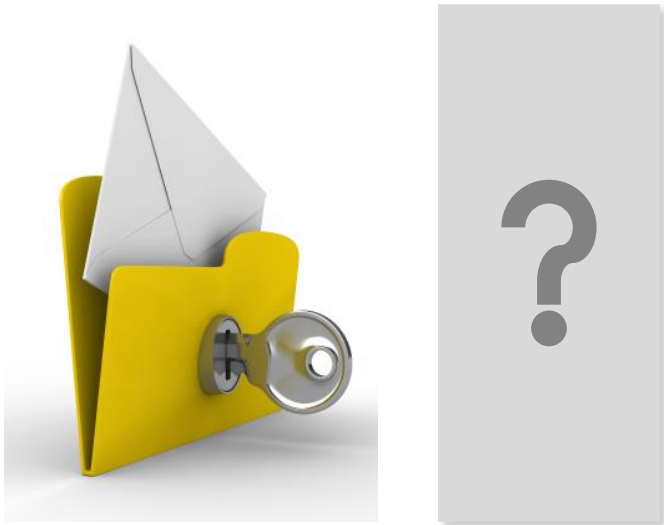
```
// Salesforce CREATE via REST API
from("direct:salesforceViaHttpCREATE")
    .setHeader("X-PrettyPrint", 1)
    .setHeader("Authorization", accessToken)
    .setHeader(Exchange.CONTENT_TYPE, "application/json")
.to("https://na14.salesforce.com/services/data/v20.0/subjects/Article__c")
```



# Did you get the key message?



# Key messages

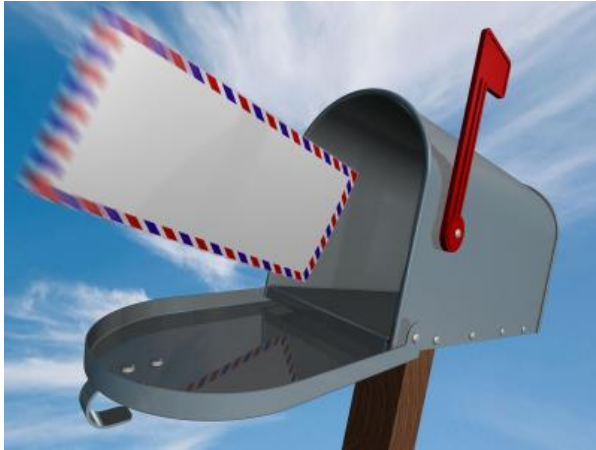


NoSQL cannot be avoided, and must be integrated! 

NoSQL integration is already possible! 

Different APIs, Frameworks and Products helps a lot! 

# Did you get the key message?



Thank you for your attention. Questions?

[kwaehner@talend.com](mailto:kwaehner@talend.com)

[www.kai-waehner.de](http://www.kai-waehner.de)

LinkedIn / Xing

@KaiWaehner

