



Integrating Spring Batch and Spring Integration

By

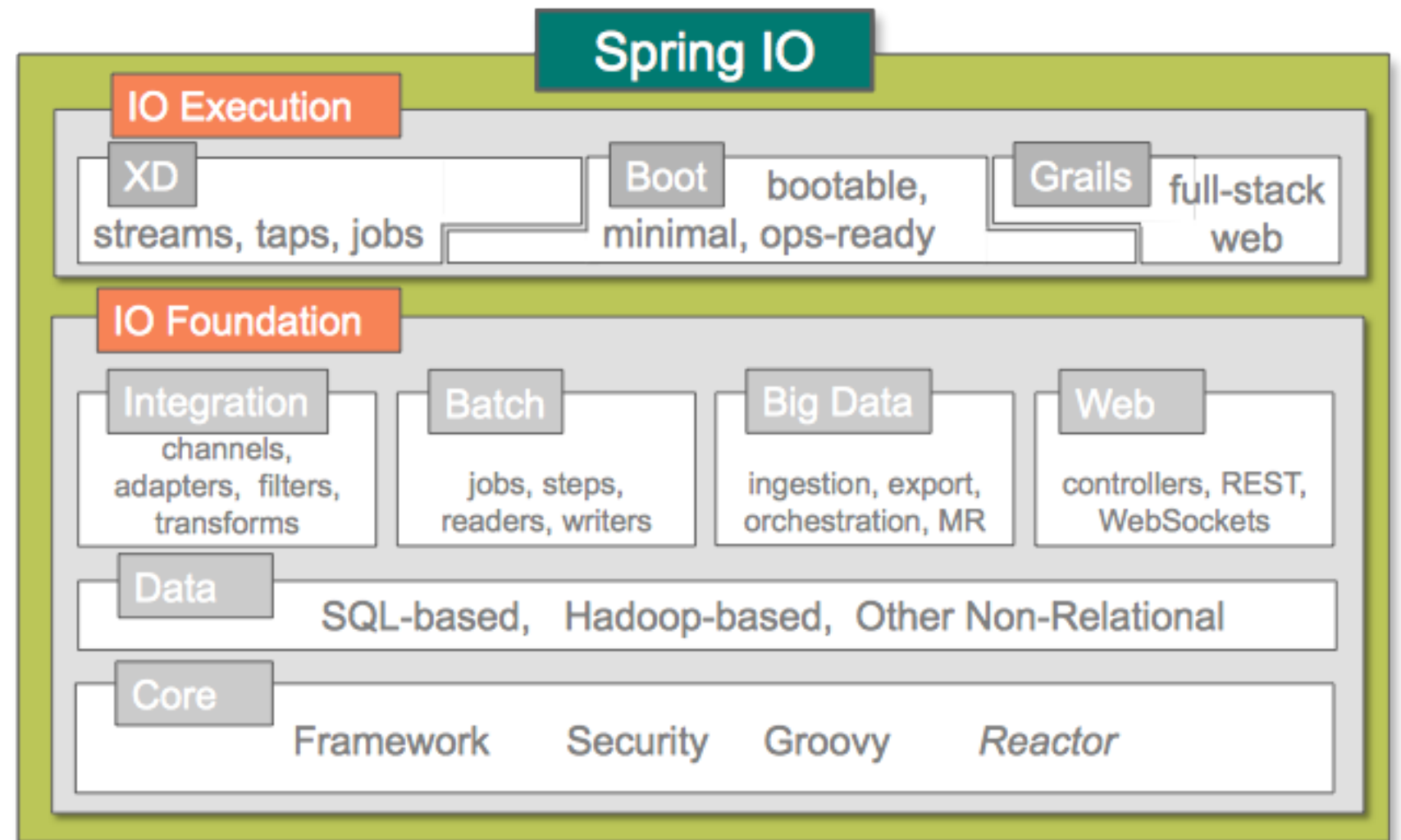
Gunnar Hillert - @ghillert

Michael Minella - @michaelminella

Pivotal™

What we will cover...

- Spring **Batch**
- Spring **Integration**
- Spring **Batch Integration**
- Spring **XD**



The Pivotal One Platform

- **Application Fabric**



- Languages, Frameworks, Services, Analytics



- **Data Fabric**



- High Capacity, Real-time, Ingest & Query, Scale-out, Storage

- **Cloud Fabric**



GemFire



- Automation, Service Registry, Cloud Independence

Spring Batch

<http://projects.spring.io/spring-batch/>

“

Batch processing ... is defined as the processing of data **without interaction or interruption.**

Michael T. Minella, Pro Spring Batch

Batch Jobs

- Long-running
 - Often outside office hours
- Non-interactive
 - Often include logic for handling errors or restarts
- Process large volumes of data
 - More than fits in memory or a single transaction

Batch and offline processing

- Close of business processing
 - Order processing, Business reporting, Account reconciliation
- Import/export handling
 - a.k.a. ETL jobs (Extract-Transform-Load)
 - Instrument/position import
 - Data warehouse synchronization
- Large-scale output jobs
 - Loyalty scheme emails, Bank statements
- Hadoop job orchestration

Features

- Transaction management
- Chunk based processing
- Declarative I/O
- Start/Restart/Skip capabilities
- Web administration interface
- Based on the Spring framework
- JSR 352: Batch Applications for the Java Platform

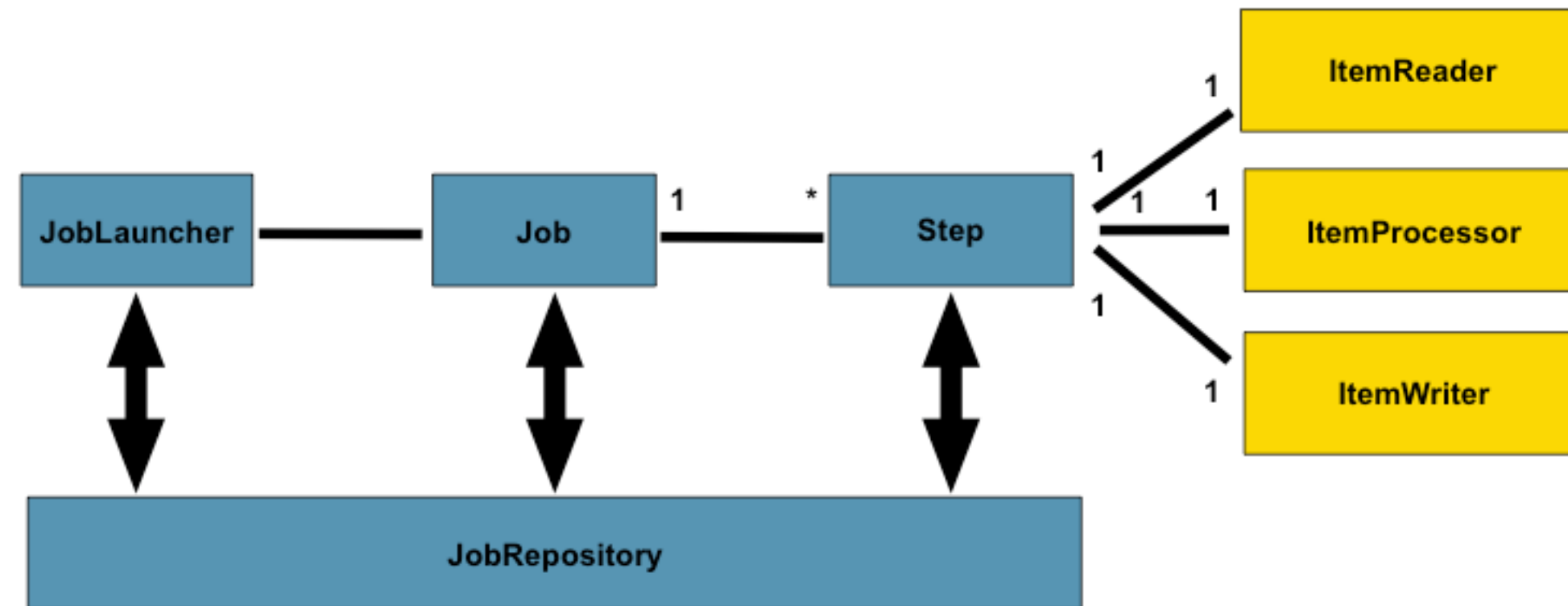
Demo

Hello Spring Batch

<https://github.com/ghillert/spring-batch-sample-hello>

Concepts

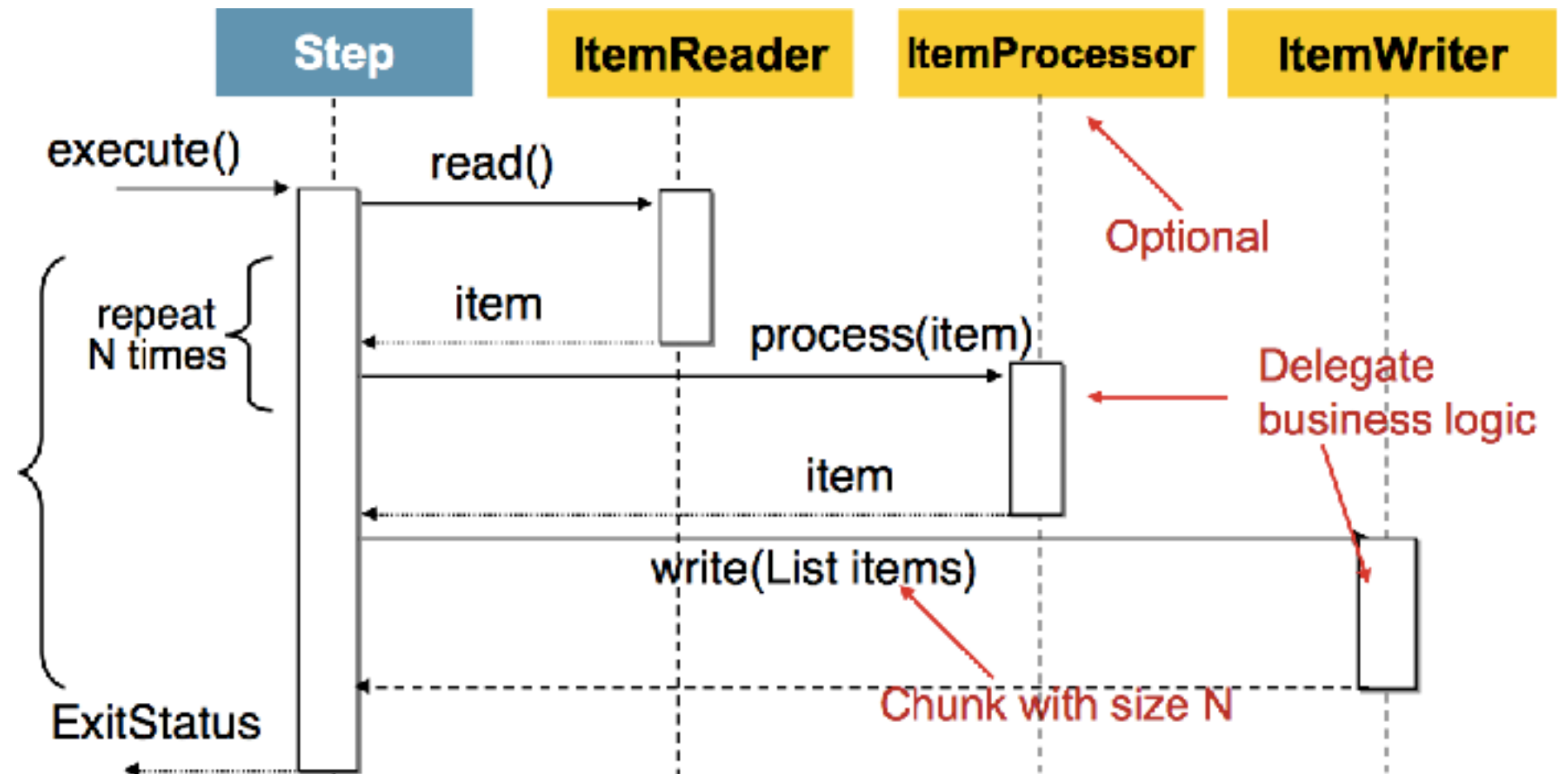
- Job
- Step
- Chunk
- Item



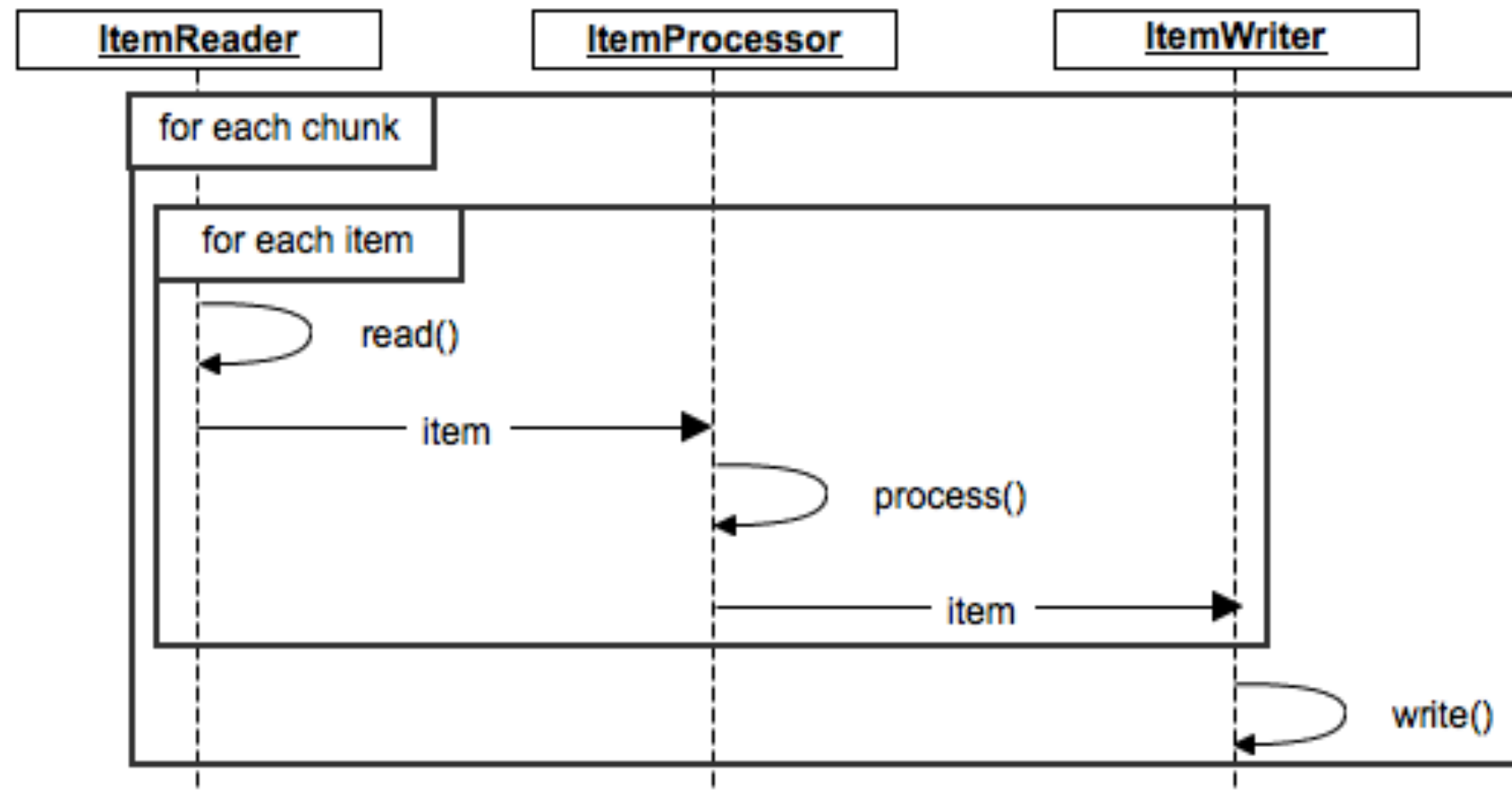
Repeat | Retry | Skip | Restart

Chunk-Oriented Processing

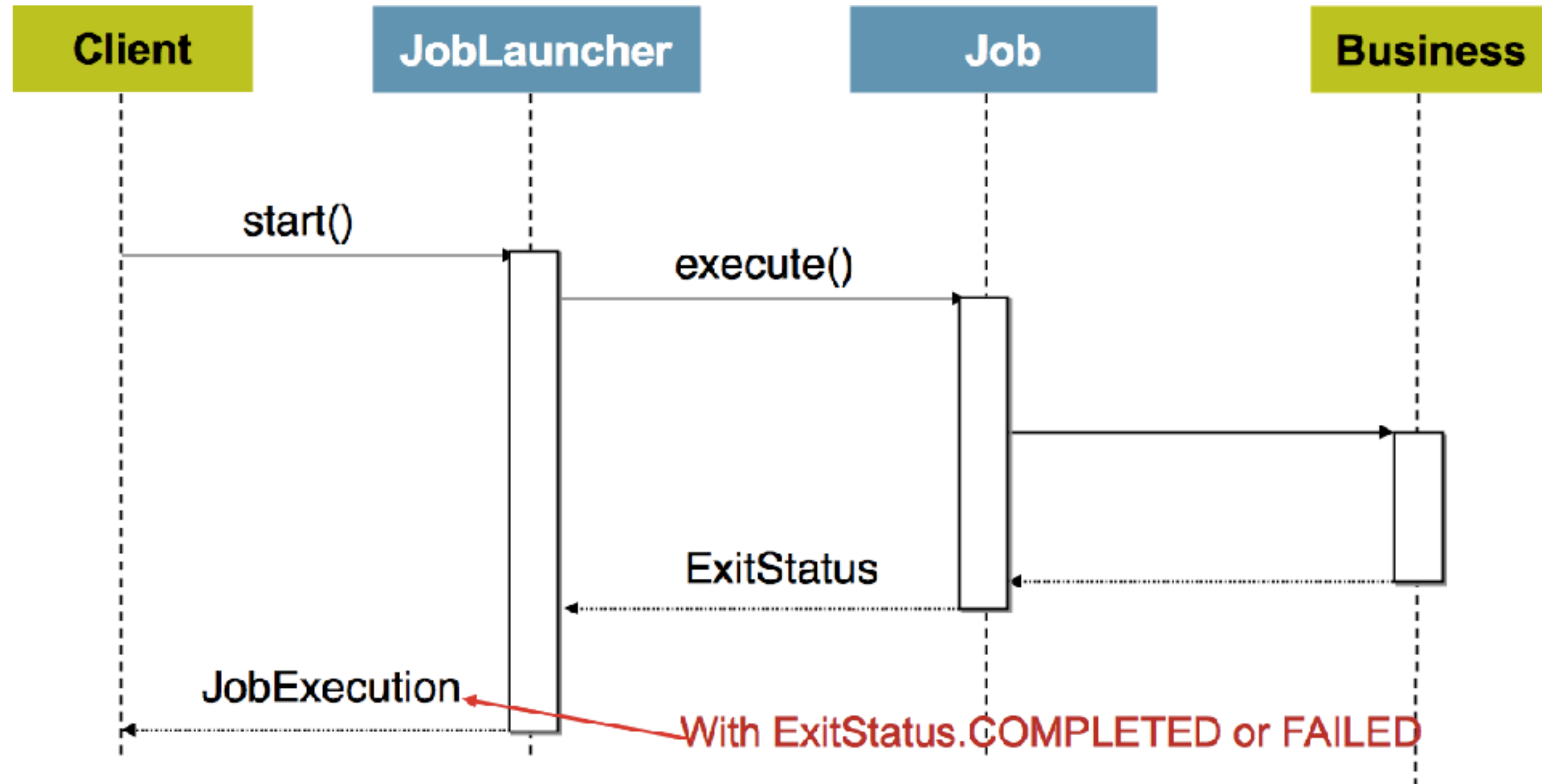
- Input-output can be grouped together
- Input collects Items before outputting: Chunk-Oriented Processing
- Optional ItemProcessor



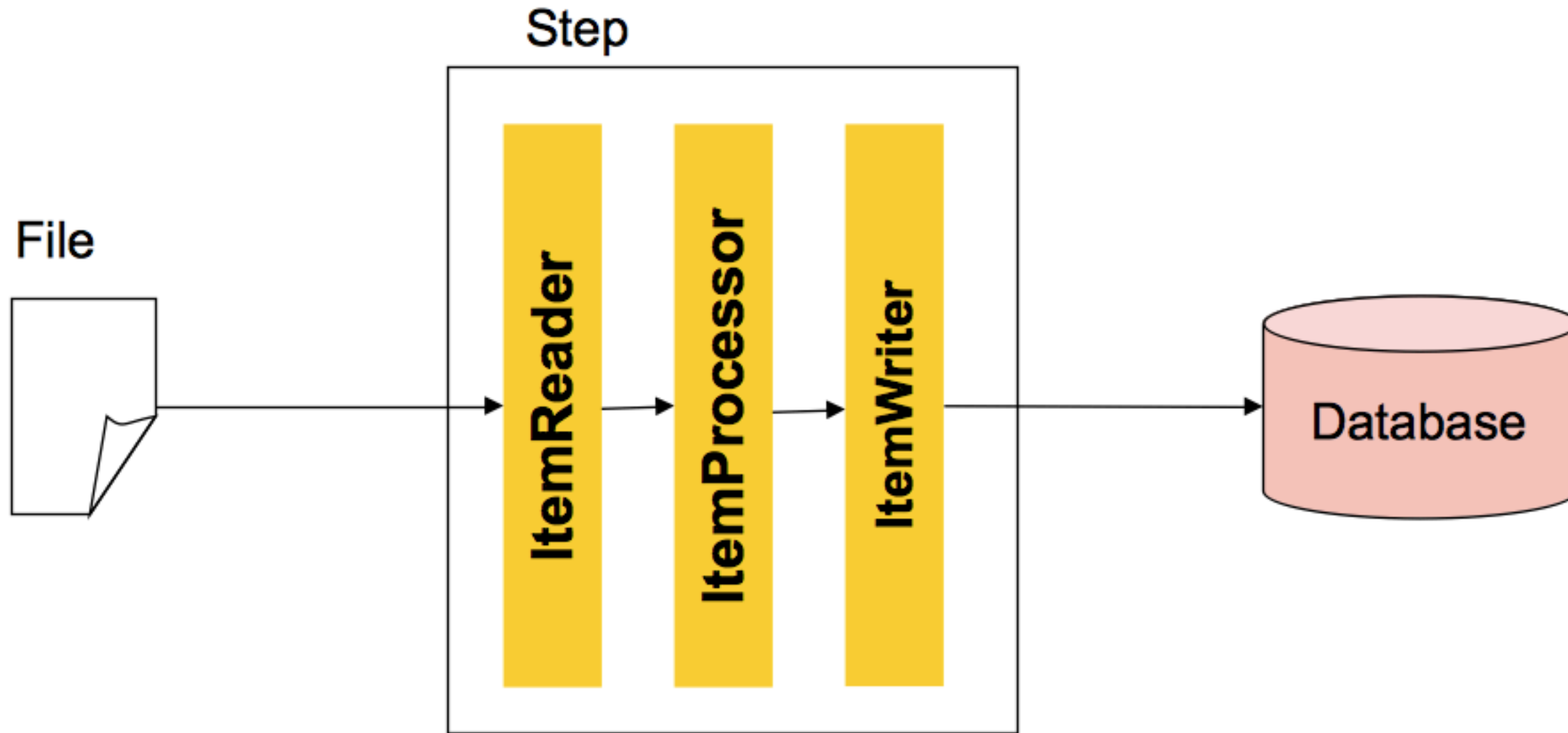
Chunk-Oriented Processing



JobLauncher



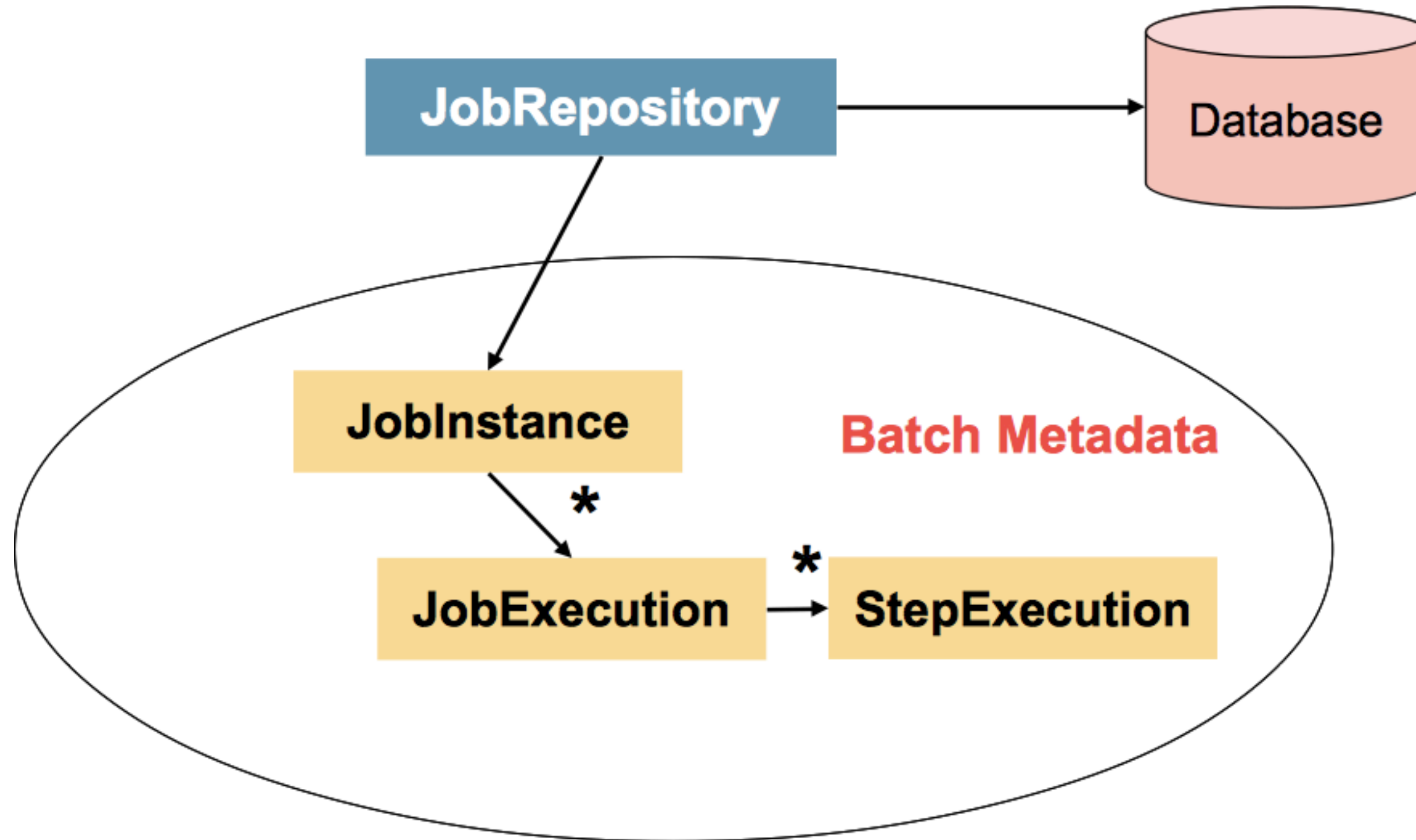
Simple File Load Job



ItemReaders and ItemWriters

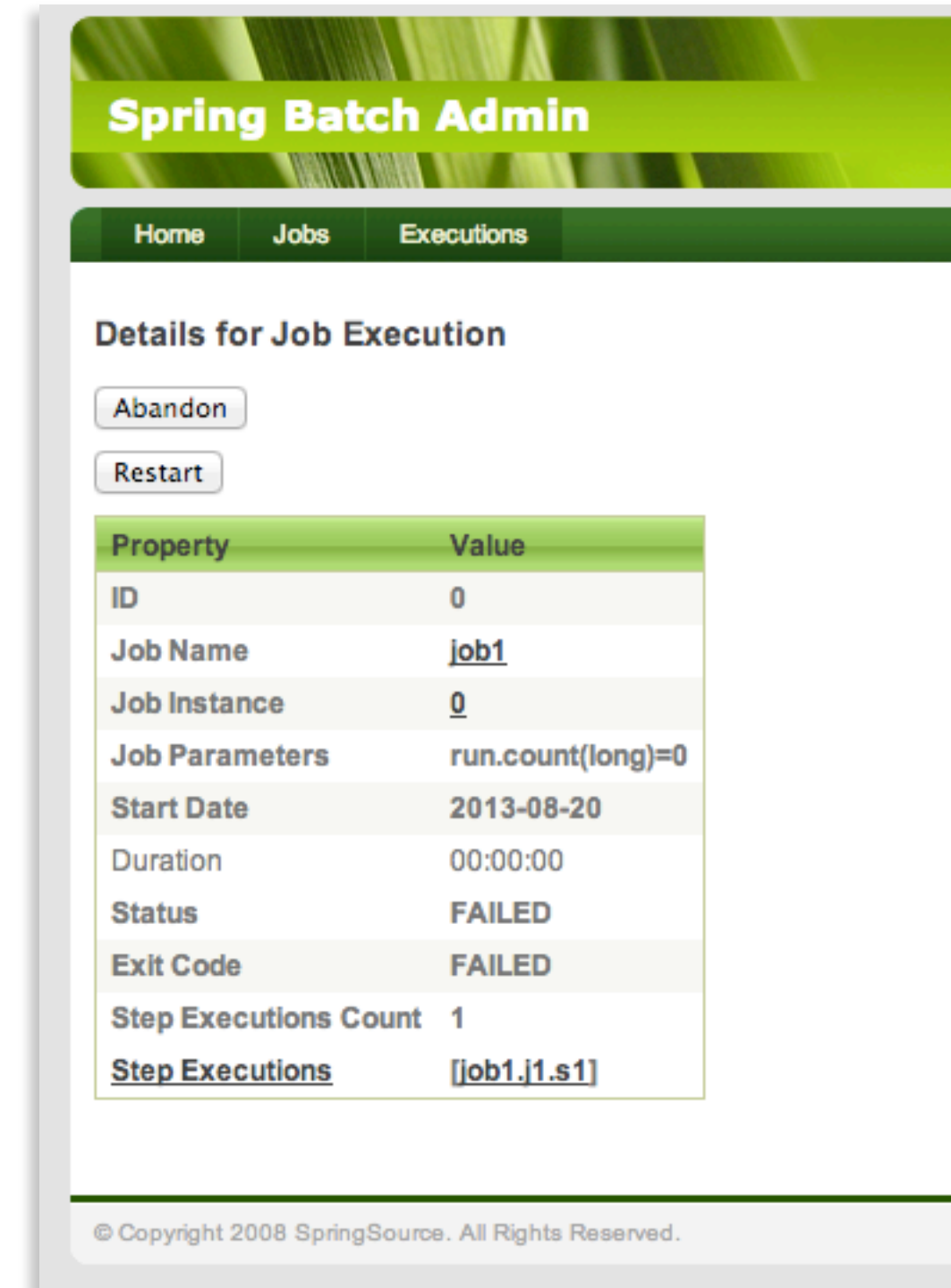
- Flat File
- XML (StAX)
- Multi-File Input
- Database
 - JDBC, JPA/Hibernate, Stored Procedures, Spring Data
- JMS
- Email
- Implement your own...

Job Repository



Spring Batch Admin

- Sub project of Spring Batch
- Provides Web UI and REST interface to manage batch processes
- Manager, Resources, Sample WAR
 - Deployed with batch job(s) as single app to be able to control & monitor jobs
 - Or monitors external jobs only via shared database



The screenshot displays the Spring Batch Admin web application. At the top, there is a green header with the title "Spring Batch Admin". Below the header is a navigation bar with three tabs: "Home", "Jobs", and "Executions". The "Jobs" tab is currently selected. The main content area is titled "Details for Job Execution". It contains two buttons: "Abandon" and "Restart". Below these buttons is a table with two columns: "Property" and "Value". The table lists various job execution details, including ID, Job Name, Job Instance, Job Parameters, Start Date, Duration, Status, Exit Code, Step Executions Count, and Step Executions. The status is "FAILED".

Property	Value
ID	0
Job Name	job1
Job Instance	0
Job Parameters	run.count(long)=0
Start Date	2013-08-20
Duration	00:00:00
Status	FAILED
Exit Code	FAILED
Step Executions Count	1
Step Executions	[job1.1.s1]

© Copyright 2008 SpringSource. All Rights Reserved.

Spring Integration

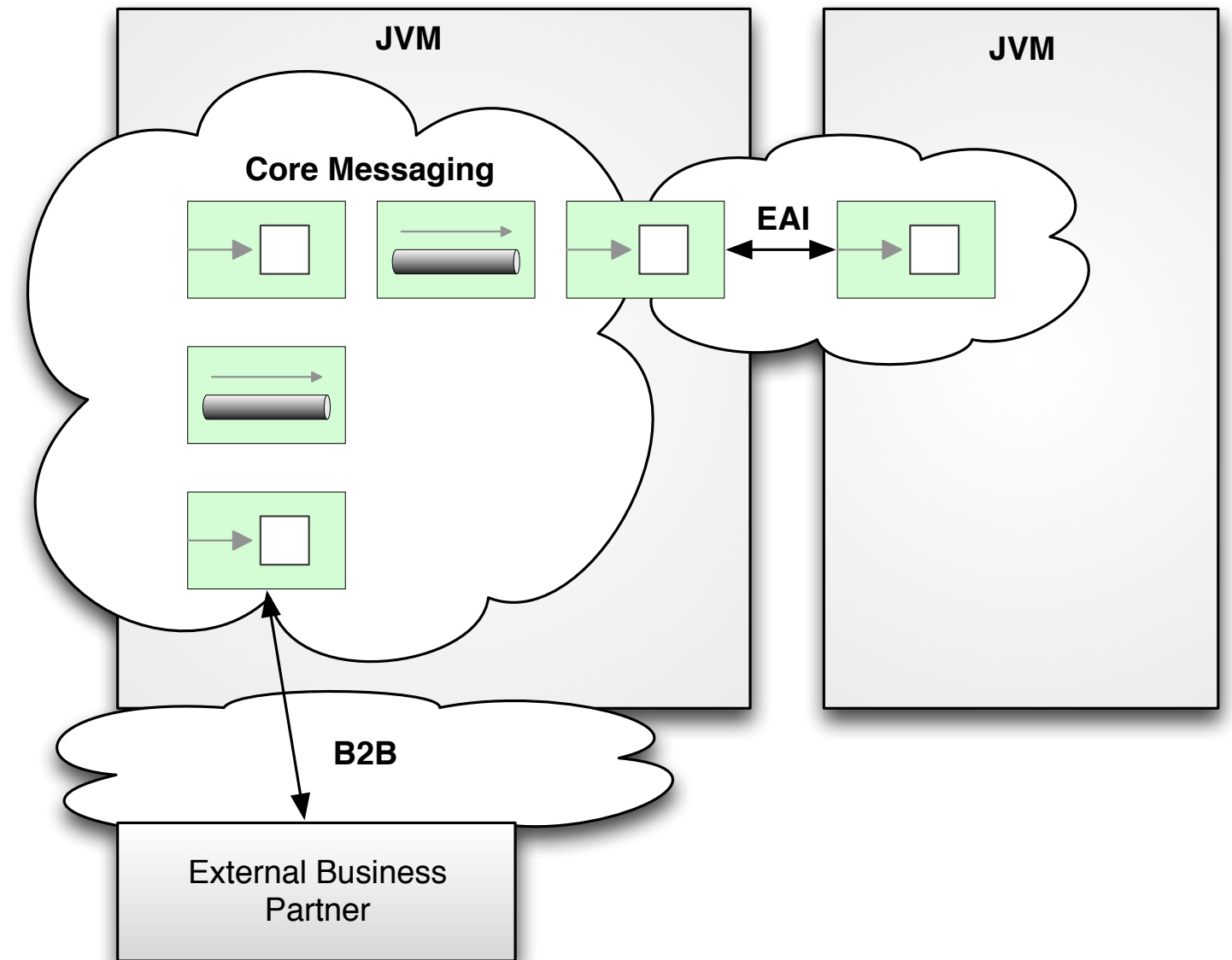
<http://projects.spring.io/spring-integration/>

Integration Styles

- File Transfer
- Shared Database
- Remoting
- Messaging

Integration Styles

- Business to Business Integration (B2B)
- Inter Application Integration (EAI)
- Intra Application Integration

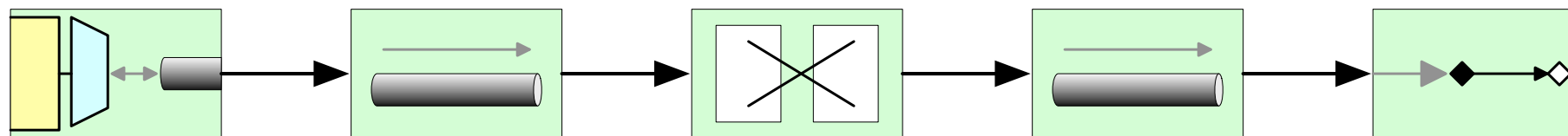


Common Patterns

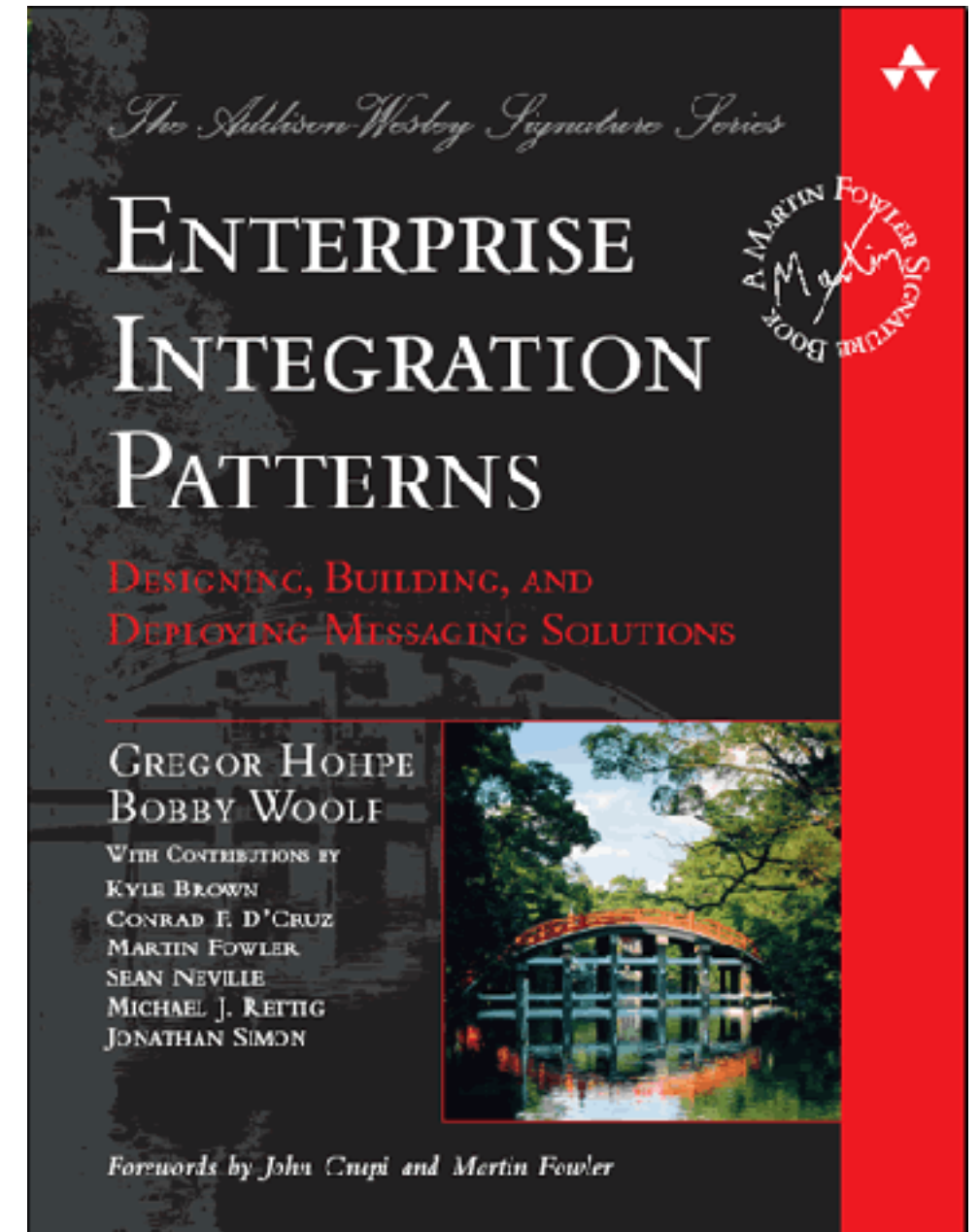


Enterprise Integration Patterns

- By Gregor Hohpe & Bobby Woolf
- Published 2003
- Collection of well-known patterns
- Icon library provided



<http://www.eaipatterns.com/eaipatterns.html>



“

Spring Integration provides an extension of the Spring programming model to support the well-known enterprise integration patterns.

Spring Integration Website

Spring Integration Components

- Claim Check (In/Out)
- Content Enricher
- Header Enricher
- Payload Enricher
- Control Bus
- Delayer
- JMX Support
- Message Handler Chain
- Messaging Bridge
- Resequencer

- Service Activator
- Scripting support (JSR 223)
- Ruby/JRuby, Javascript ...
- Groovy
- Message History
- Message Store
- JDBC, Redis, MongoDB, Gemfire
- Wire Tap
- ...

Adapters

- AMQP/RabbitMQ
- AWS*
- File/Resource
- FTP/FTPS/SFTP
- GemFire
- HTTP (REST)
- JDBC
- JMS
- JMX
- JPA

- MongoDB
- POP3/IMAP/SMTP
- Print
- Redis
- RMI
- RSS/Atom
- SMB
- Splunk
- Spring Application Events

- Stored Procedures
- TCP/UDP
- Twitter
- Web Services
- XMPP
- XPath
- XQuery
- ...

Samples

- <https://github.com/SpringSource/spring-integration-samples>
- Contains 50 Samples and Applications
- Several Categories:
 - Basic
 - Intermediate
 - Advanced
 - Applications

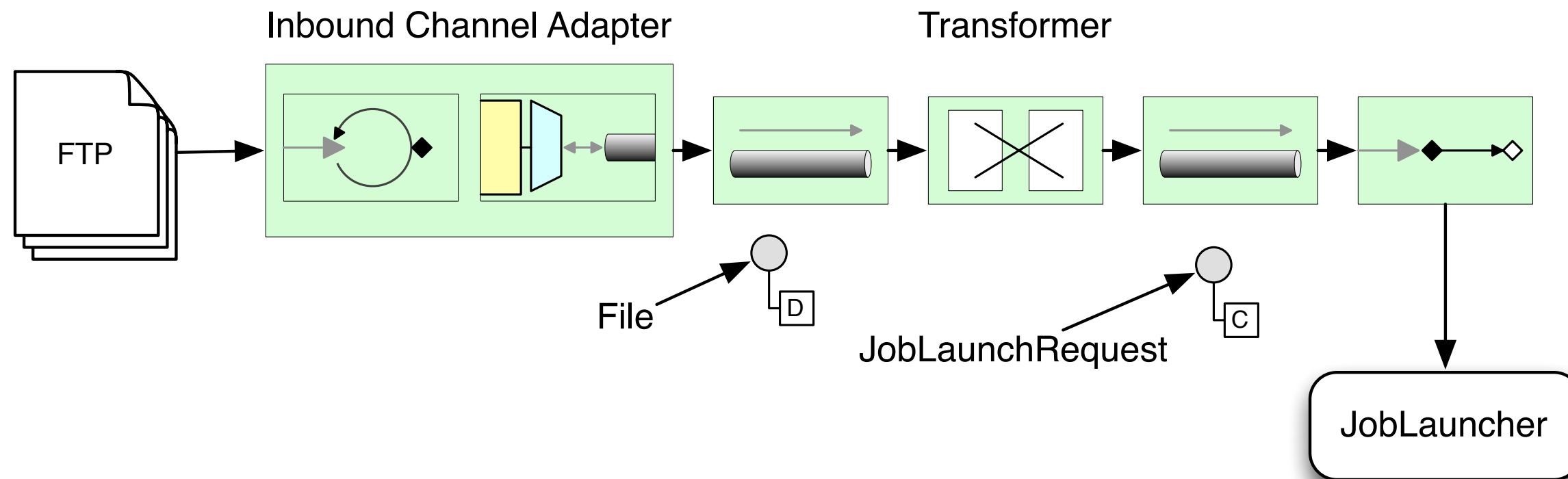
Spring Batch Integration

<https://github.com/SpringSource/spring-batch-admin/>



Launching batch jobs through messages

- Event-Driven execution of the JobLauncher
- Spring Integration retrieves the data (e.g. file system, FTP, ...)
- Easy to support separate input sources simultaneously



JobLaunchRequest

```
public class FileMessageToJobRequest {
    private Job job;
    private String fileParameterName;
    ...
    @Transformer
    public JobLaunchRequest toRequest(Message<File> message) {
        JobParametersBuilder jobParametersBuilder = new JobParametersBuilder();
        jobParametersBuilder.addString(fileParameterName,
                                      message.getPayload().getAbsolutePath());
        return new JobLaunchRequest(job, jobParametersBuilder.toJobParameters());
    }
}
```

DefaultJobParametersConverter

- Convert (textual) Properties/Maps to JobParameters
- Provide Typed Parameters
 - Date
 - String
 - Long
 - Double
- Provide Date+Number Format
- Define Identifying / Non-Identifying Parameters

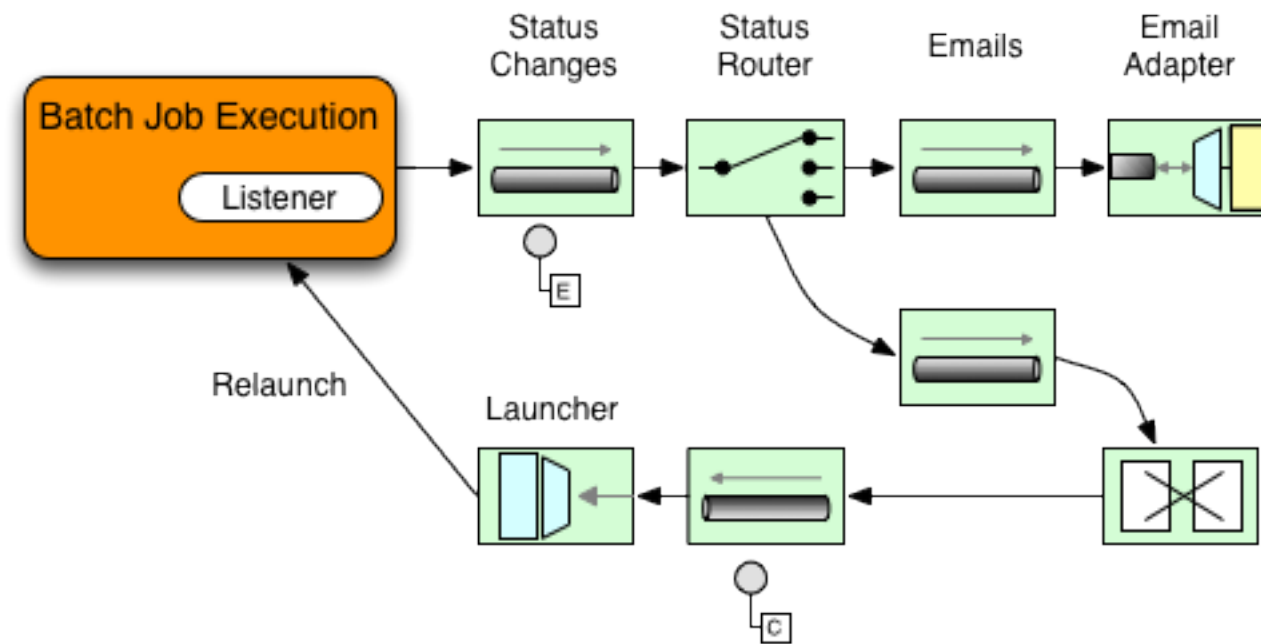
```
myDateParam(date)=2013/09/10  
aStringParameter=Hello Spring  
-stringParamNOTIdentifying=Hello Spring  
aNumberParameter(Long)=123456
```

JobLaunchRequest

```
<batch-int:job-launching-gateway    request-channel="requestChannel"  
                                   reply-channel="replyChannel"  
                                   job-launcher="jobLauncher" />
```


Get feedback with informational messages

- Spring Batch provides support for listeners:
 - StepExecutionListener
 - ChunkListener
 - JobExecutionListener



Get feedback with informational messages

```
<batch:job id="importPayments">
    ...
    <batch:listeners>
        <batch:listener ref="notificationExecutionsListener"/>
    </batch:listeners>
</batch:job>

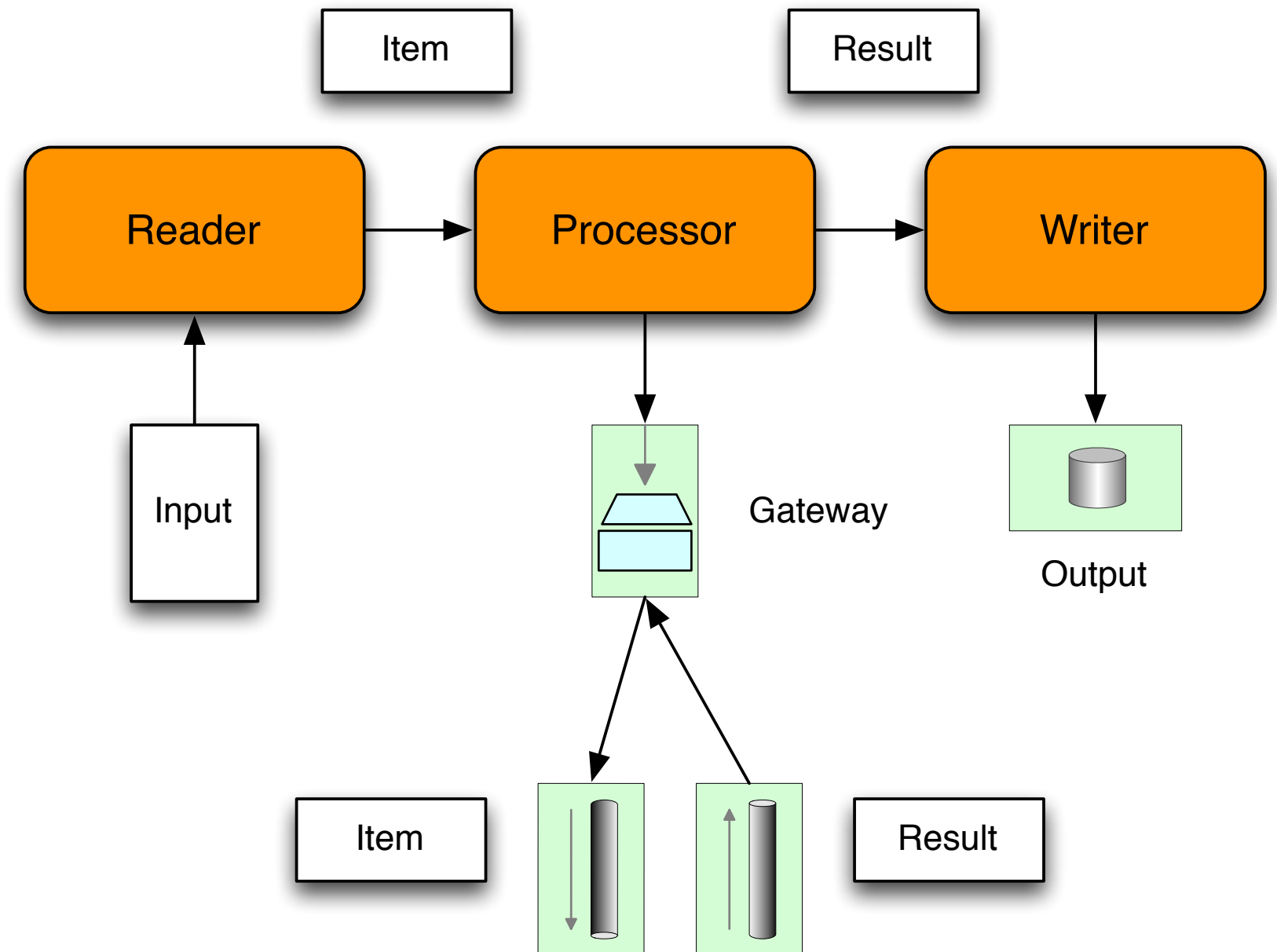
<int:gateway id="notificationExecutionsListener"
    service-interface="o.s.batch.core.JobExecutionListener"
    default-request-channel="jobExecutions"/>
```

Externalizing batch process execution

- Offload complex processing
- Use Spring Integration inside of Batch jobs
 - e.g. ItemProcessor + ItemWriter
- Asynchronous Item processing support
 - AsyncItemProcessor
 - AsyncItemWriter
- Externalize Chunk processing
 - ChunkMessageChanneItemWriter

Asynchronous Processors

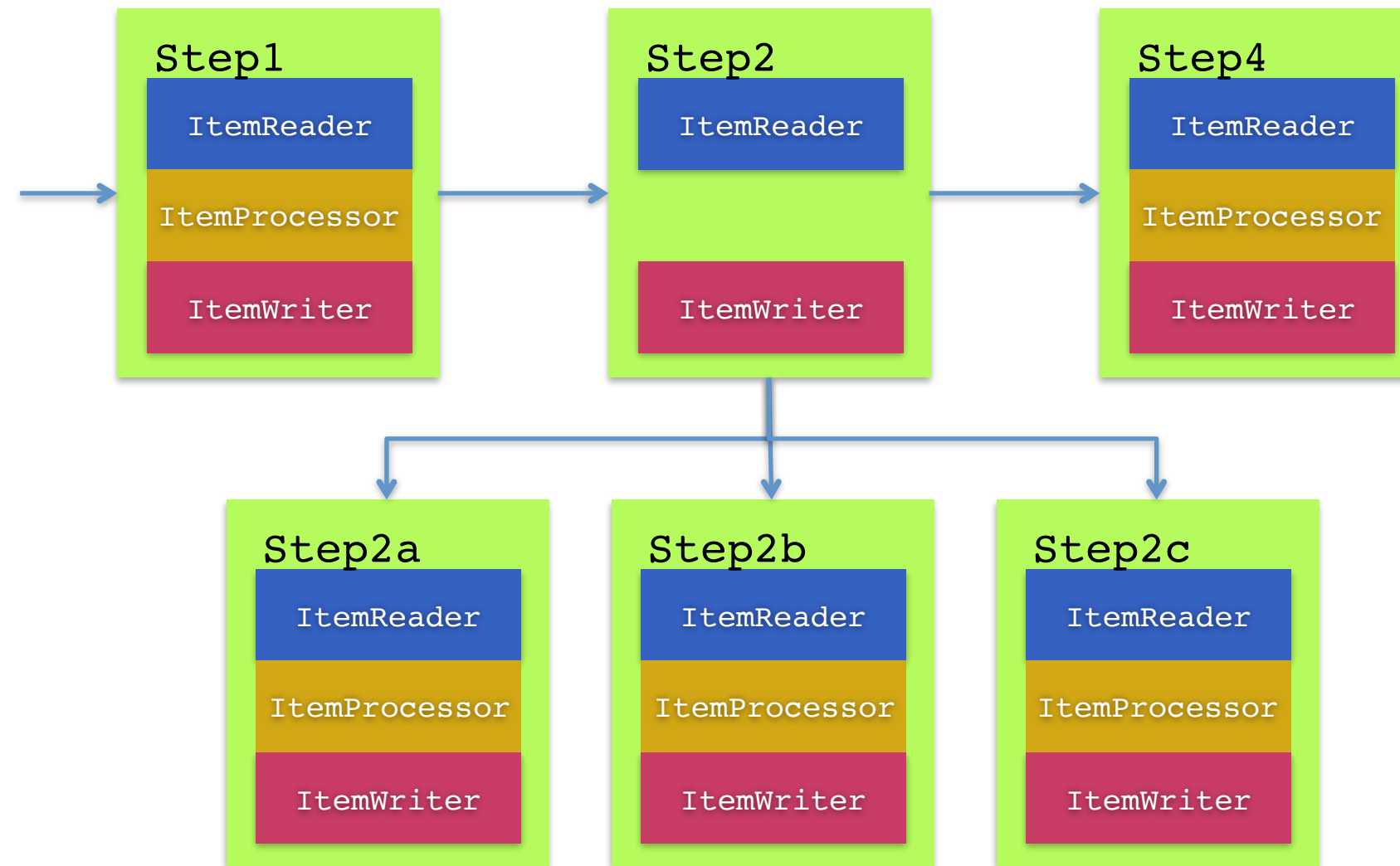
- AsyncItemProcessor
- AsyncItemWriter
- externalize the processing of **items**



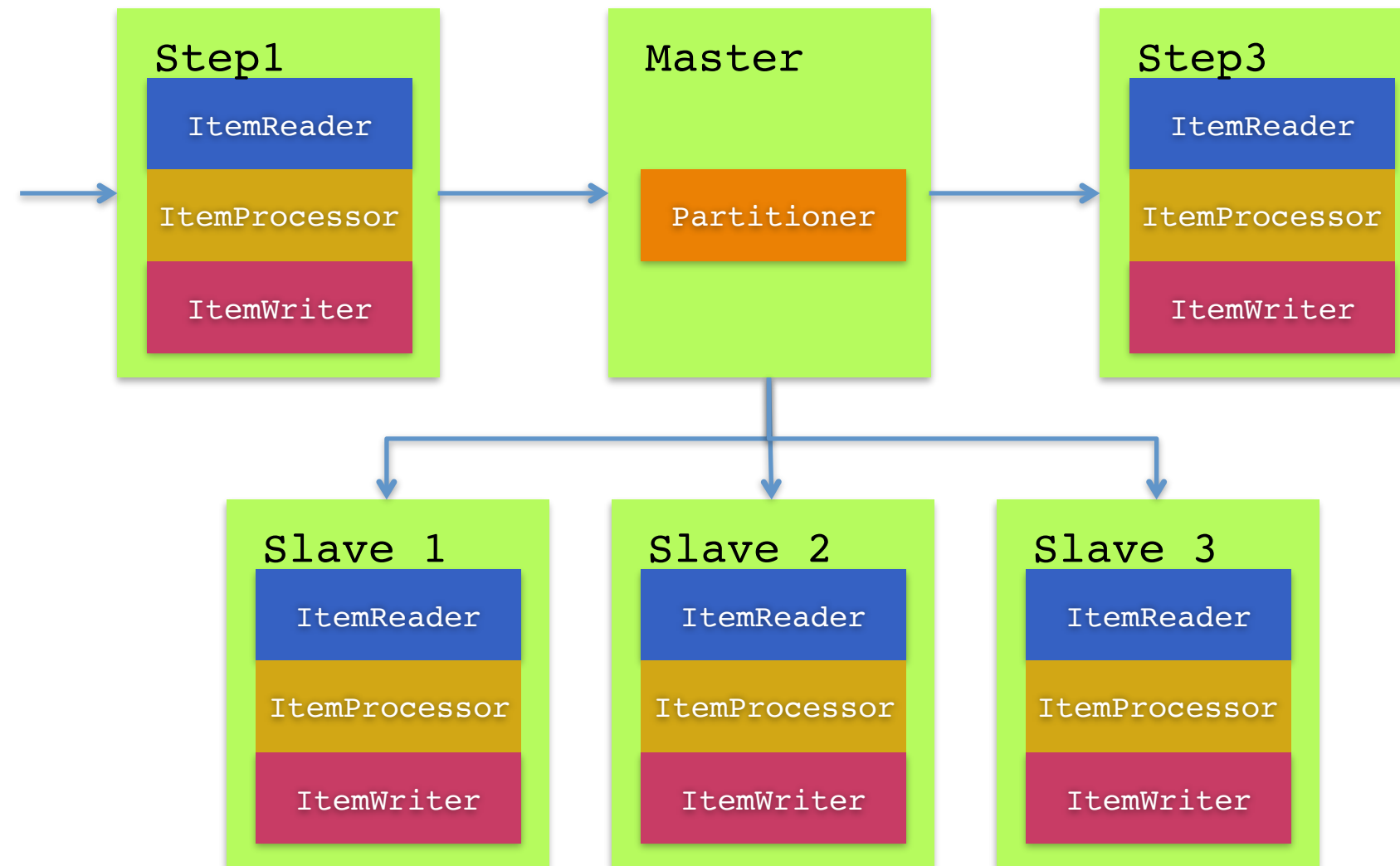
Concurrent Step vs. Async Processors

	Concurrent Step	Async Processors
Parallel Read/Write	X	
Restartability		X
Write in Sequence		X

Remote Chunking

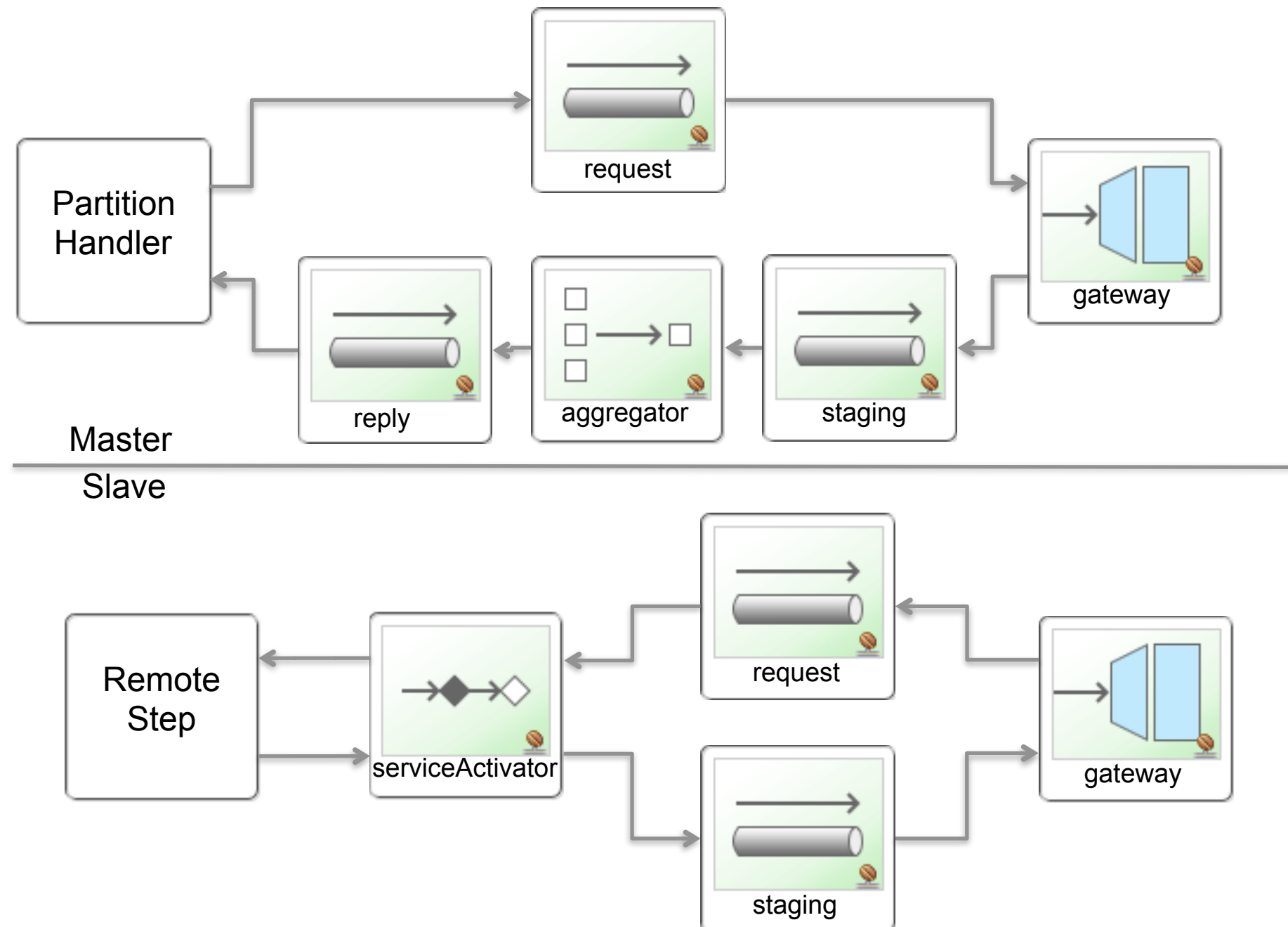


Remote Partitioning



Remote Partitioning

- MessageChannelPartitionHandler



Demo

Payment Import

<https://github.com/ghillert/spring-batch-integration-sample>

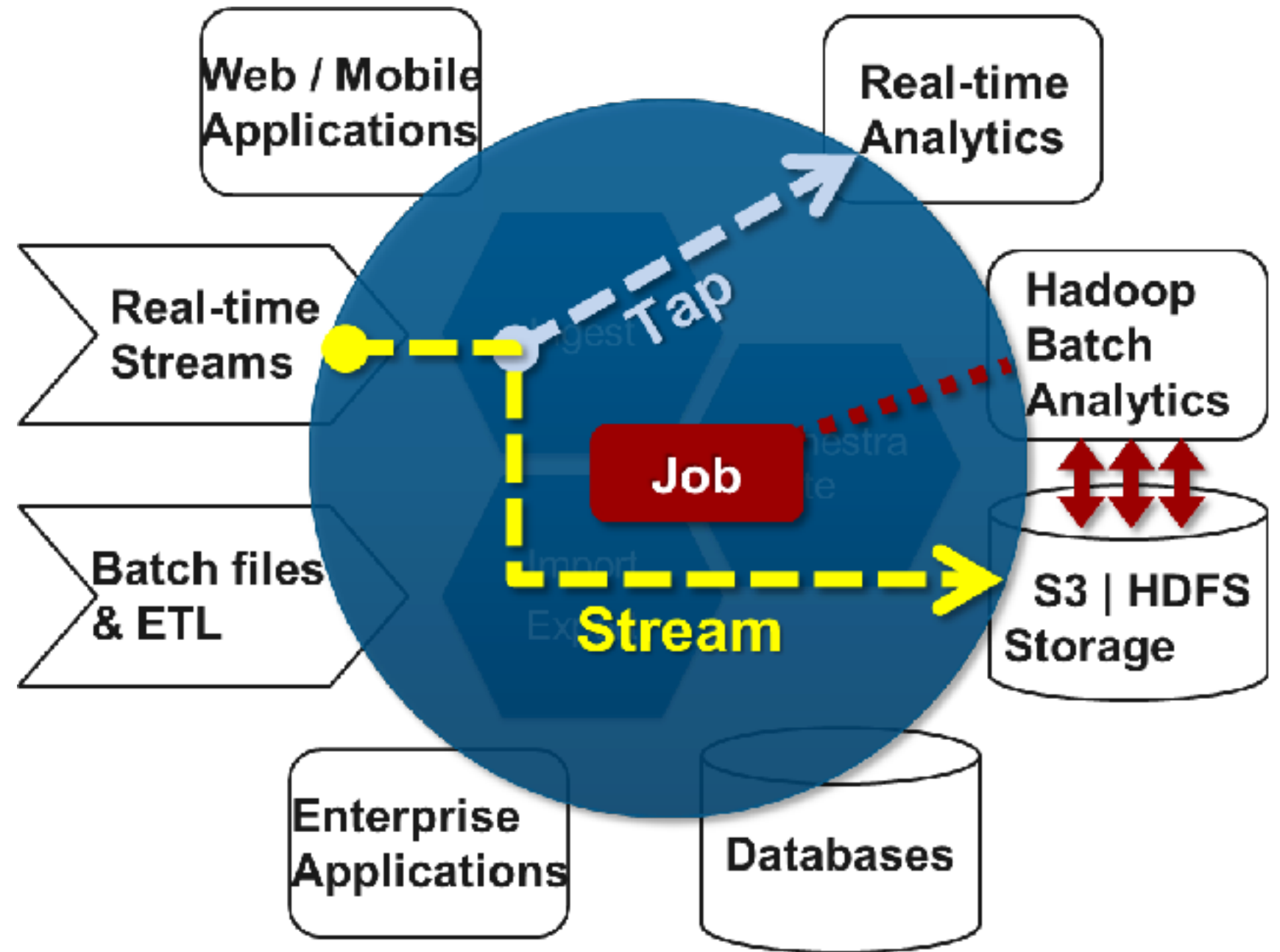
Spring XD

<http://projects.spring.io/spring-xd/>



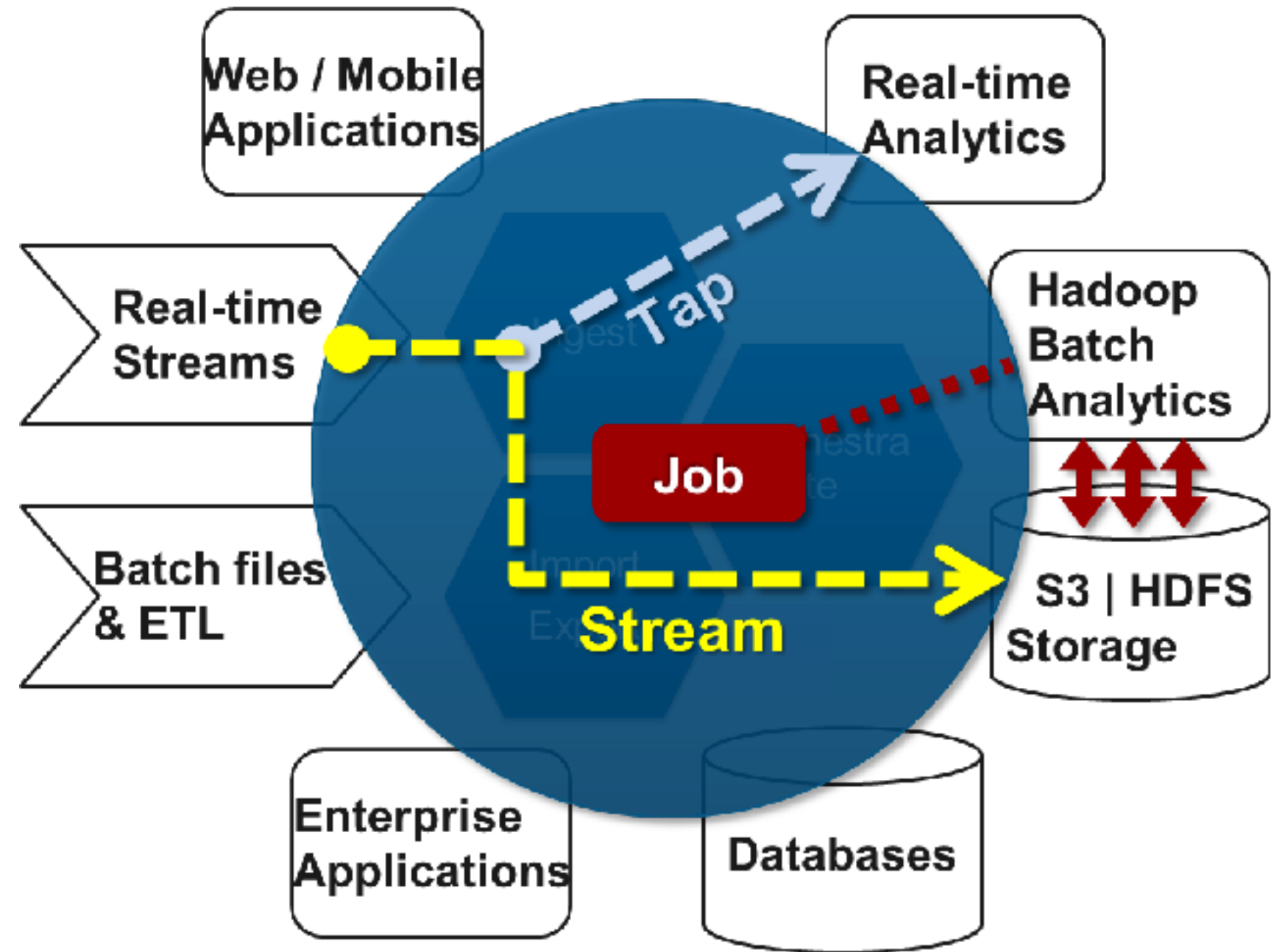
Tackling Big Data Complexity

- Unified agile experience for
- Data Ingestion
- Real-time Analytics
- Workflow Orchestration
- Data Export



Tackling Big Data Complexity cont.

- Built on existing assets
 - Spring Integration
 - Spring Batch
 - Spring Data
 - Redis, GemFire, Hadoop
- XD = 'eXtreme Data'



Data Ingestion Streams

- DSL based on Unix pipes and filters syntax

`http | file`

- Modules are parameterizable

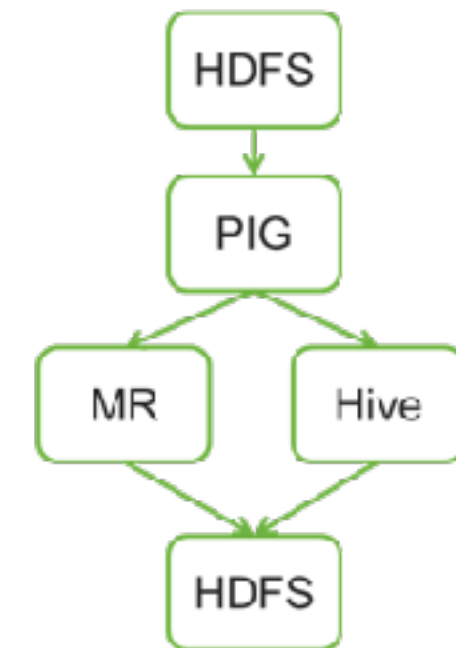
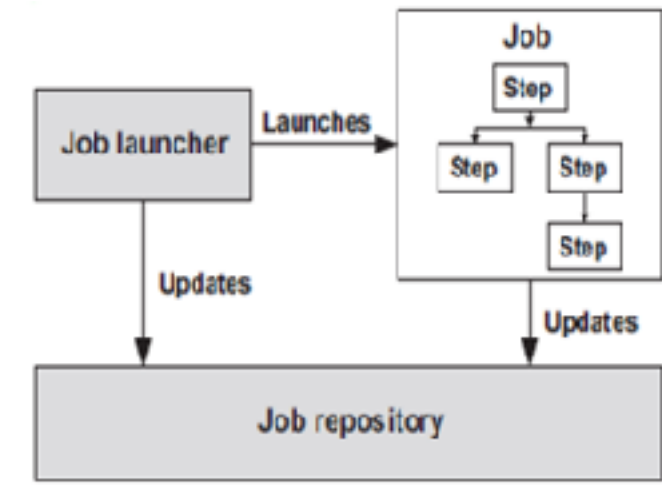
`twittersearch --query=spring | file --dir=/spring`

- Simple logic can be added via expressions or scripts

`http | filter --expression=payload=='Spring' | hdfs`

Hadoop workflow managed by Spring Batch

- Reuse Batch infrastructure and features to manage Hadoop workflows
 - Job state management, launching, monitoring, restart/retry policies, etc.
- Step can be any Hadoop job type or HDFS script
- Can mix and match with other Batch readers/writers
 - (e.g. JDBC for import/export use-cases)



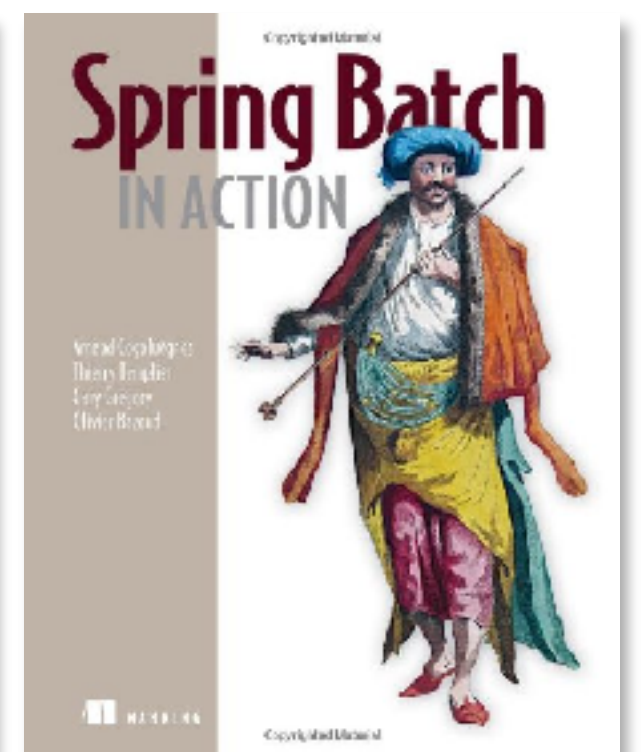
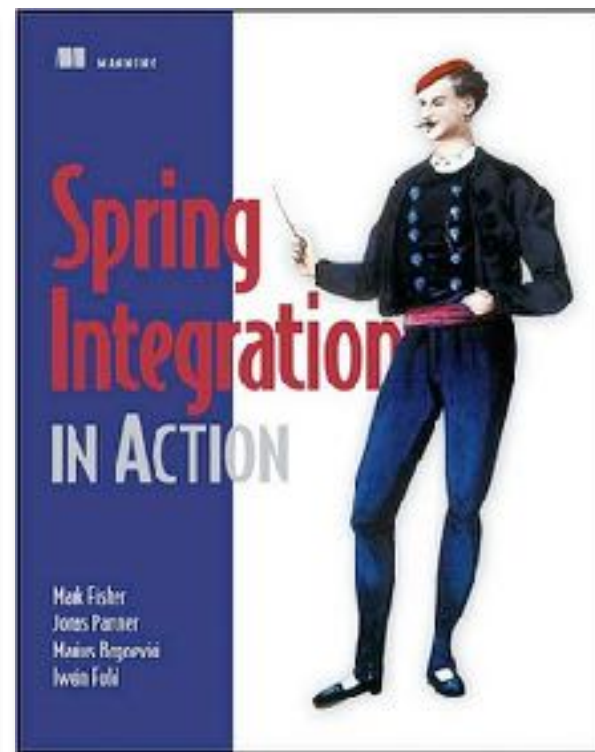
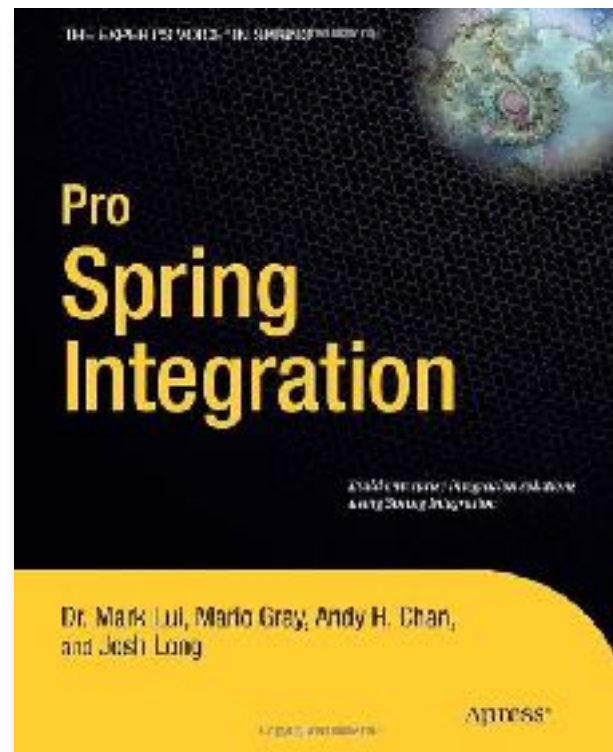
Demo

Spring XD

Batch word-count sample

<https://github.com/SpringSource/spring-xd-samples>

Books



Learn More. Stay Connected.



JSR-352, Spring Batch and You @14:30 (Tue)
Spring Integration Internals @12:45 (Wed)
Tackling Big Data Complexity with Spring @14:30 (Wed)
Real Time Analytics with Spring @16:30 (Wed)

Talk to us on Twitter: @springcentral

Find session replays on YouTube: spring.io/video