JSR 352 Expert Group

Working Session

29 February 2012

Agenda

- Review: Launcher, Manager Interfaces
- First Look: Launch & Packaging
- Examples: Batch Container Use
- Discussion: Readers/Writers
- List for Next Meeting

Review: Launcher, Manager

```
public interface BatchLauncher {
    JobId execute(String jobName,Object[] parameters);
}
```

```
public interface BatchManager {
    void stop(JobId jobInstance);
    void restart(JobId jobInstance, Object[] parameters);
    void remove(JobId jobInstance);
    JobResults getResults(JobId jobInstance); // exitstatus/job RC
    void getJobLog(JobId jobInstance,Writer jobLogWriter);
    List<JobInfo> getJobs(JobFilter filter);
}
```

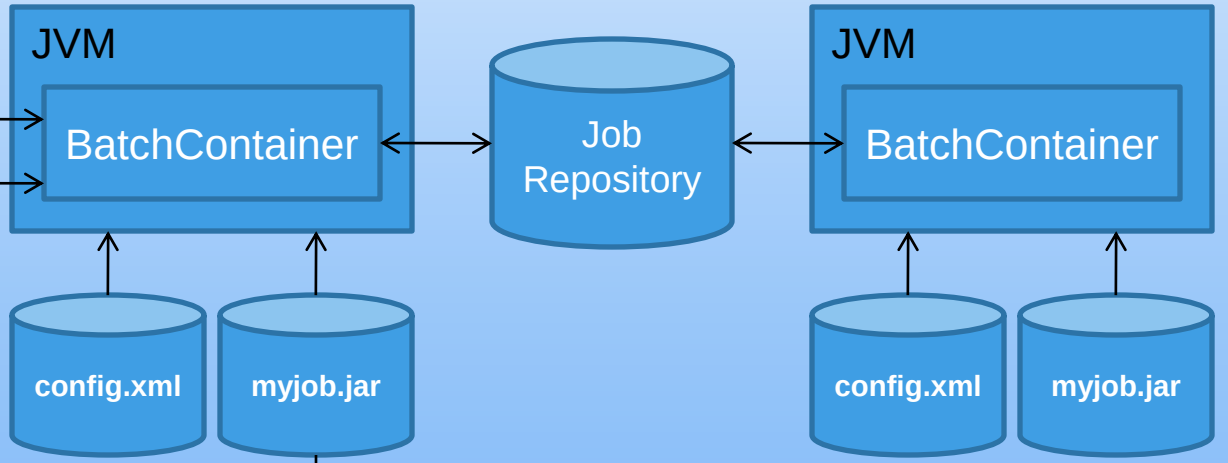
Note: actual names and signatures TBD

First Look: Launch & Packaging

container operations:

launcher.execute("Job1")

manager.getJobs()



"Job" annotation processor writes jobs.xml

class files:

```
@Job(name="Job1")
public class MyJob {}
```

```
@Job(name="Job2")
public class YourJob {}
```

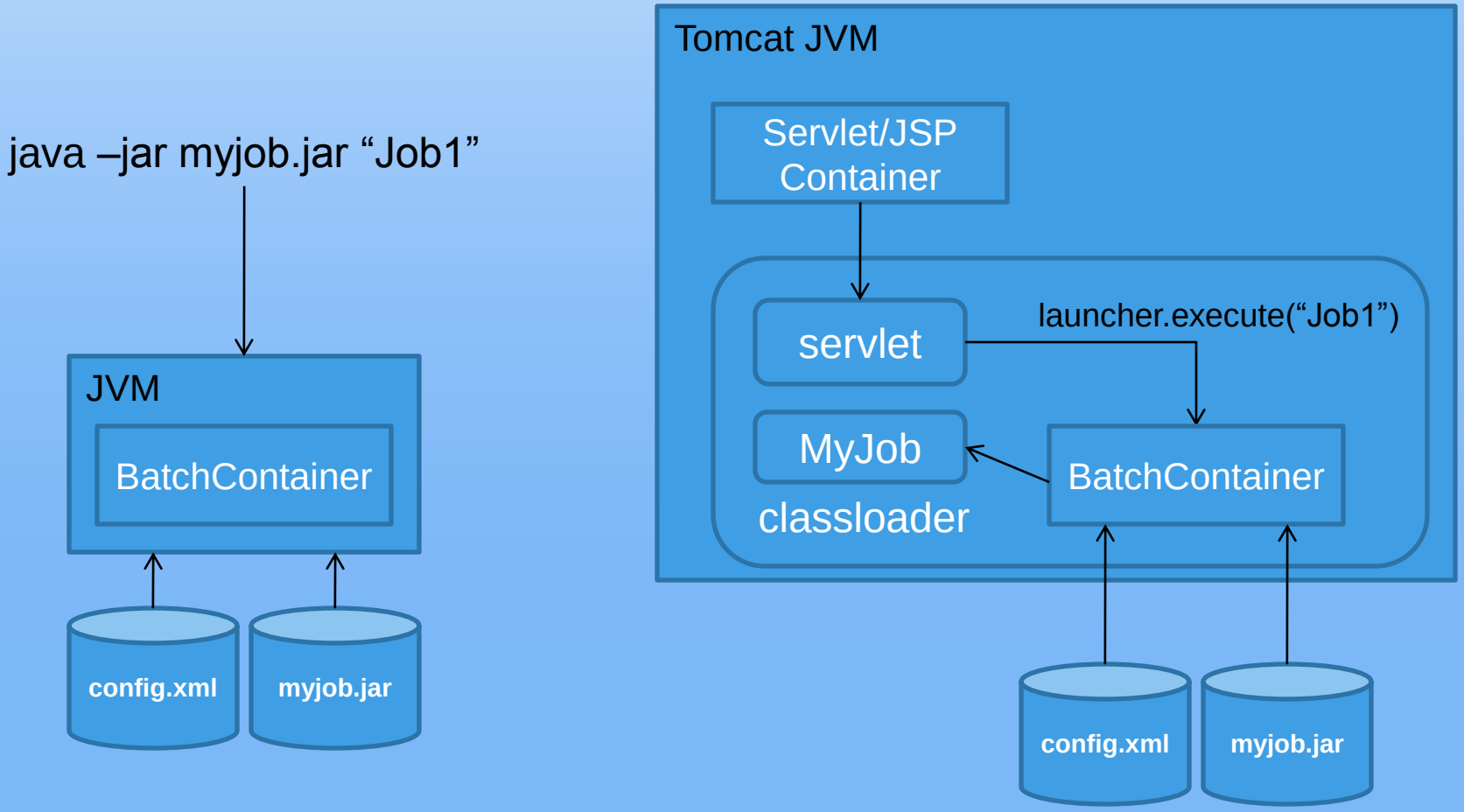
META-INF/jobs.xml:

```
<job name="Job1"
classname="com..MyJob" />

<job name="Job2"
classname="com..YourJob" />
```

Examples: BatchContainer Use

Command line and web servers:

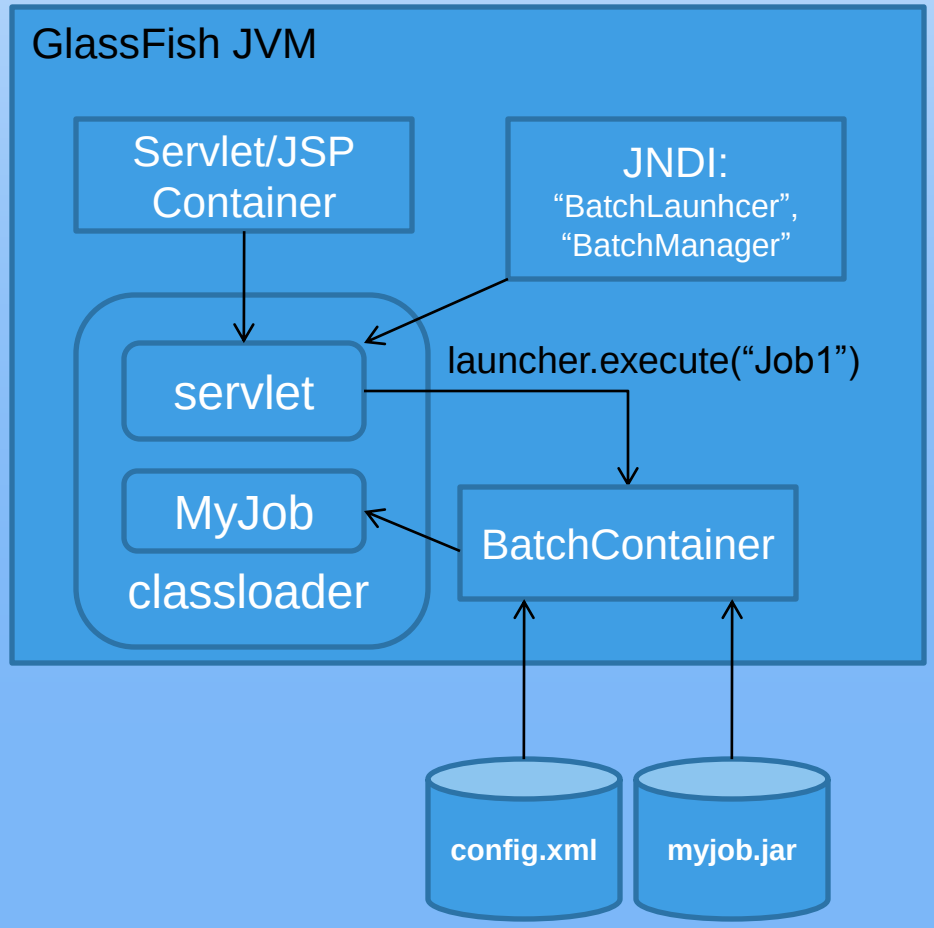


Examples: BatchContainer Use

Java EE Servers:

More likely that batch container is a singleton runtime service.

BatchLauncher and BatchManager references available through JNDI lookup.



Discussion: Runtime Configuration

- BatchContainer config.xml
  - List of plugin implementations:
    - TransactionManager **<<< More discussion?**
    - JobRepositoryManager
    - JobLogManager
    - ExecutionManager (i.e. task executor)
    - CommandLineHandler
    - ParameterResolver?
  - Properties bags to configure each plugin

Discussion: Readers/Writers

Idea: combine ItemReader(Writer) and ItemStream semantics

- Annotations

```
@Reader (or @Writer)
public  class MyReader {
    @Open void open(CheckpointInfo chkpt) {…}
    @Close void close() {…}
    @ItemReader {Type} read() {…}
    @GetCheckpointInfo CheckpointInfo getCheckpt() {…}
}

@ItemWriter void write({Type} record) {…}
```

Discussion: Readers/Writers

- Usage

```
@Step(…)
@Reader(ref="MyFileReader")
@Writer(ref="MyFileWriter")
public class MyStep {

@ItemProcessor {Type1} process({Type2} rec) {…}

}
```

List for Next Meeting

- Listeners
- Concurrency
- Future
  - Repeat
  - Exit codes
  - Step conditions
  - Execution Context
  - Metrics
  - Java EE