

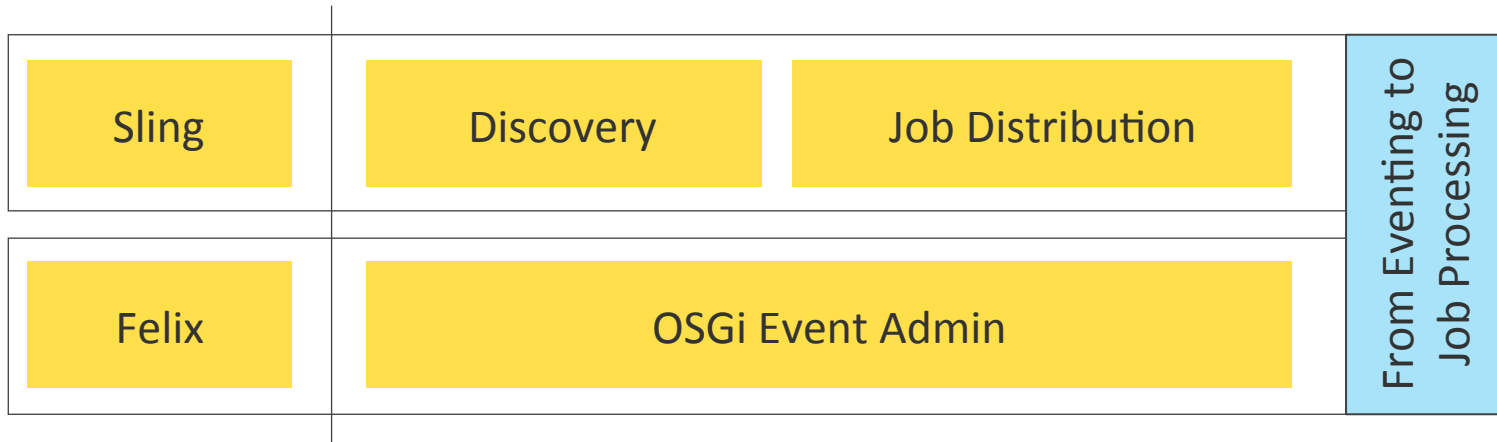
adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 23-25 SEPTEMBER 2013

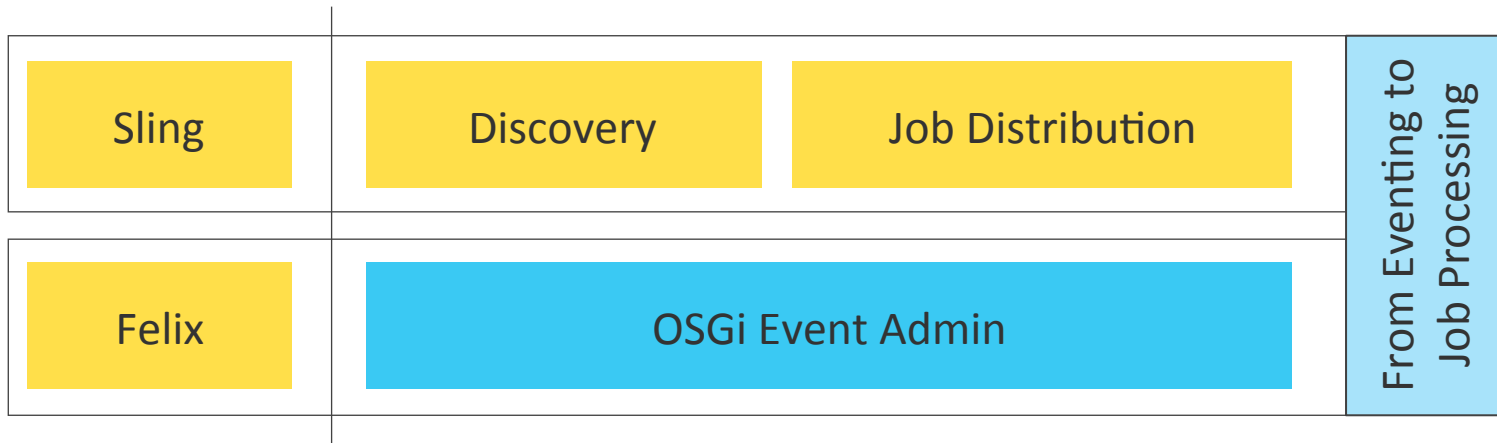
Distributed Eventing and Jobs

Carsten Ziegeler | Adobe Research Switzerland

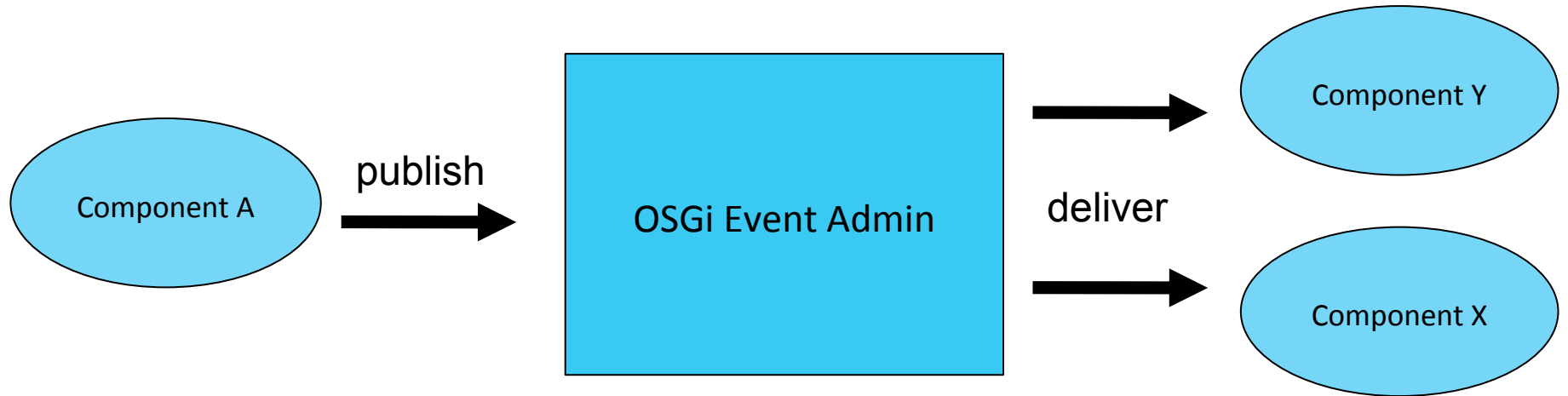
- RnD Team at Adobe Research Switzerland
 - Co-founder Adobe Granite
- OSGi Core Platform and Enterprise Expert Groups
- Member of the ASF
 - Current PMC Chair of Apache Sling
 - Apache Sling, Felix, ACE
- Conference Speaker
- Technical Reviewer
- Article/Book Author



Overview



OSGi Event Admin Publish Subscribe Model



OSGi Event Admin

Publish Subscribe Model

- OSGi event is a data object with
 - Topic (hierarchical namespace)
 - Properties (key-value-pairs)
- Resource Event
 - Topic:
org/apache/sling/api/resource/Resource/ADDED
 - Properties: path, resource type etc.

OSGi Event Admin

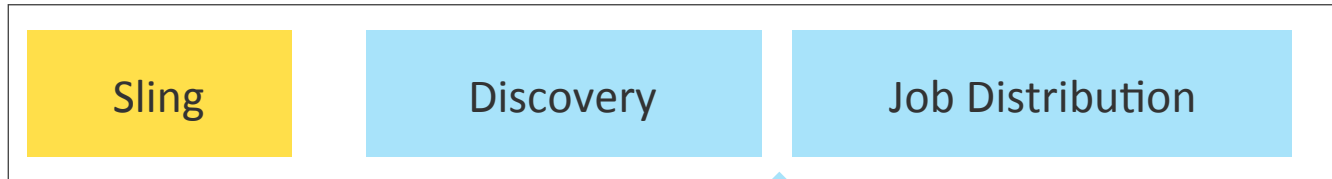
Publish Subscribe Model

- Publisher creates event object
 - Sends event through EventAdmin service
 - Either sync or async delivery
- Subscriber is an OSGi service (EventHandler)
 - Service registration properties
 - Interested topic(s)
 - `org/apache/sling/api/resource/Resource/*`
 - Additional filters (optional)
 - `(path=/libs/*)`

OSGi Event Admin

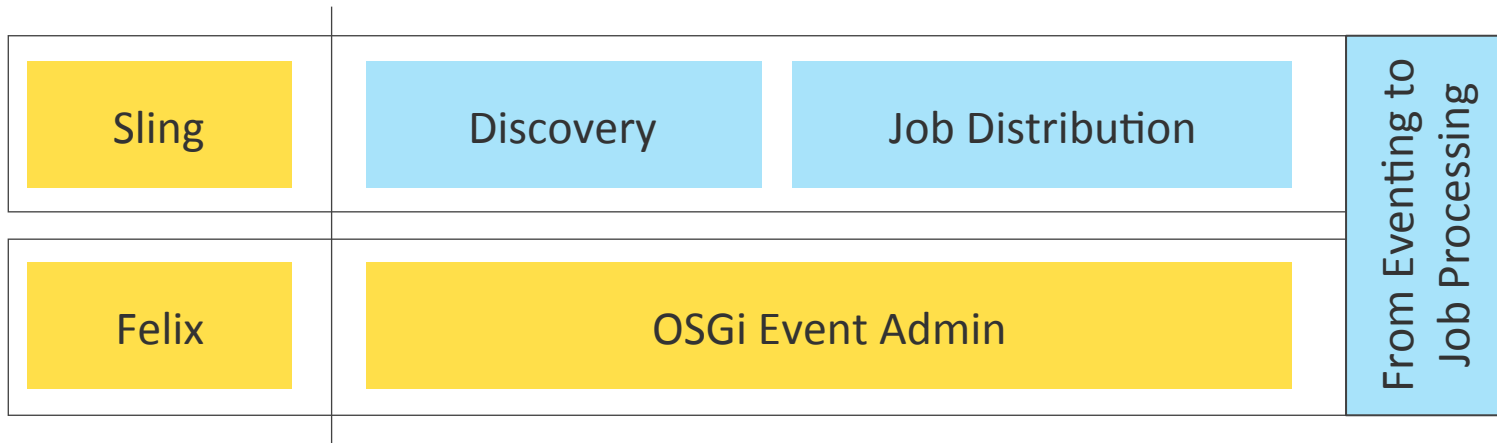
Publish Subscribe Model

- Immediate delivery to available subscribers
- No guarantee of delivery
- No distributed delivery

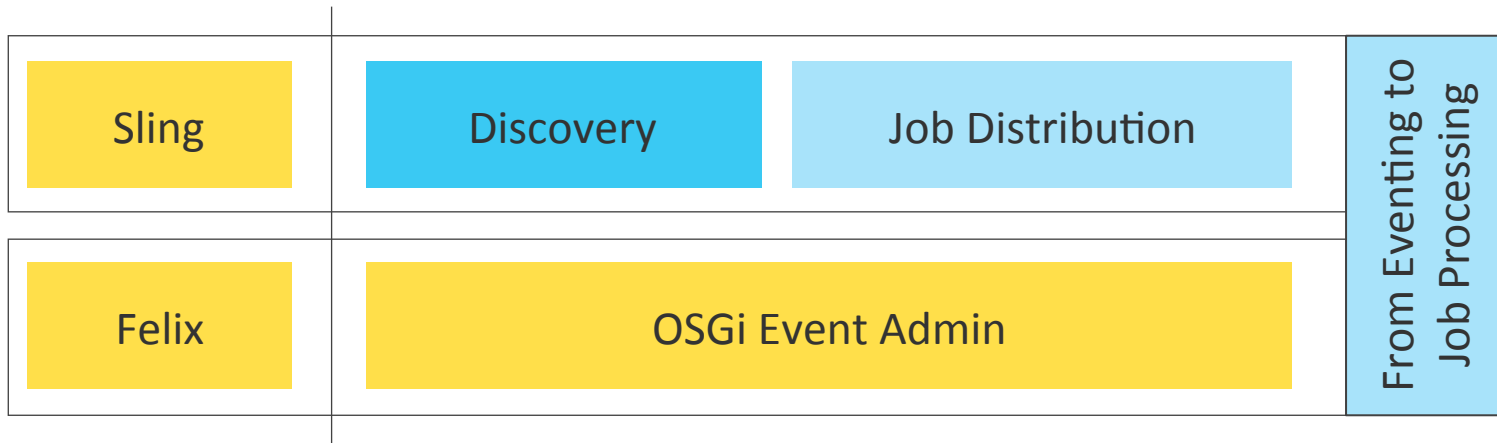


- Immediate delivery to available subscribers
- No guarantee of delivery
- No distributed delivery

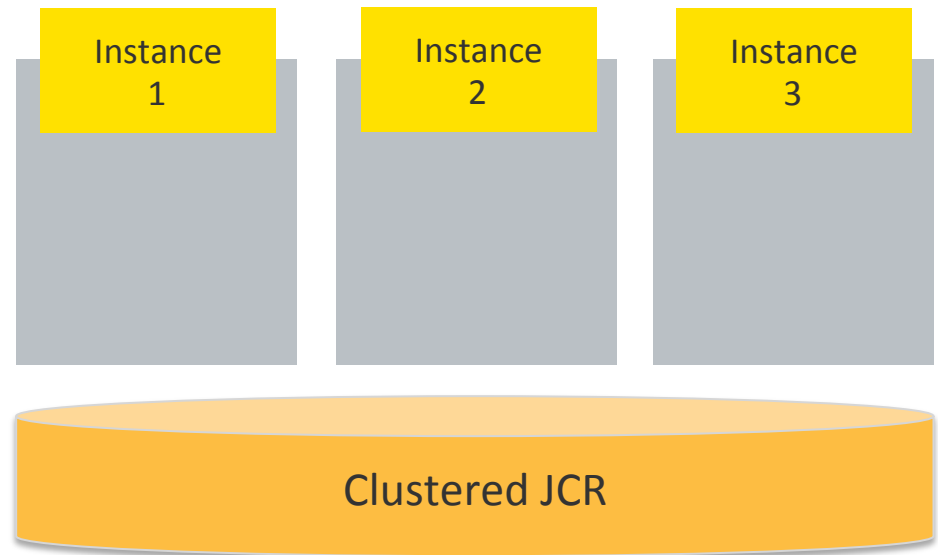
Overview



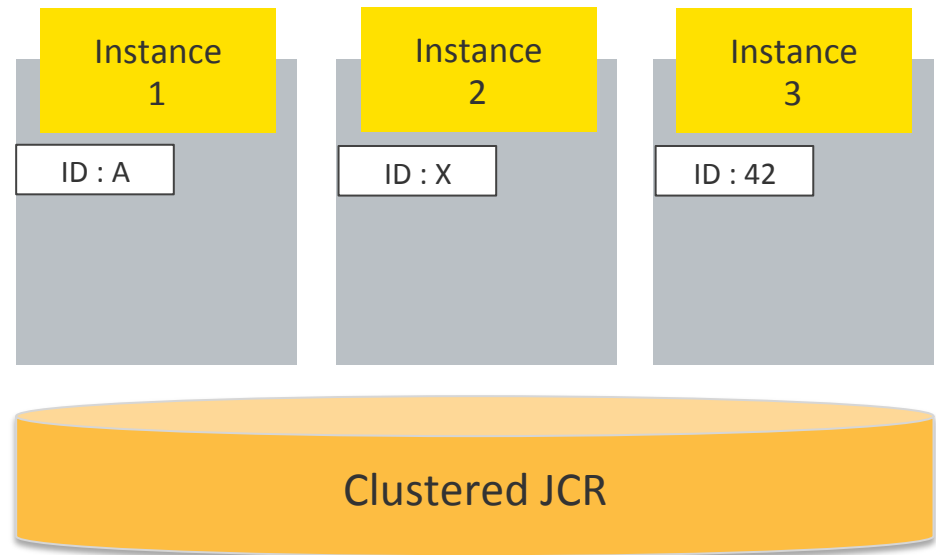
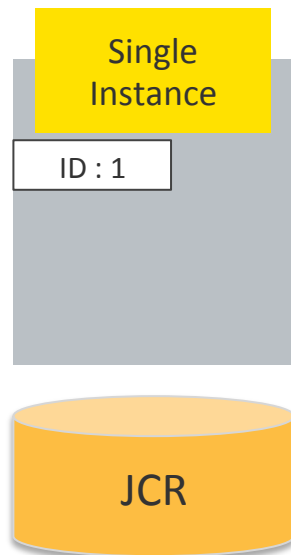
Overview



Installation Scenarios



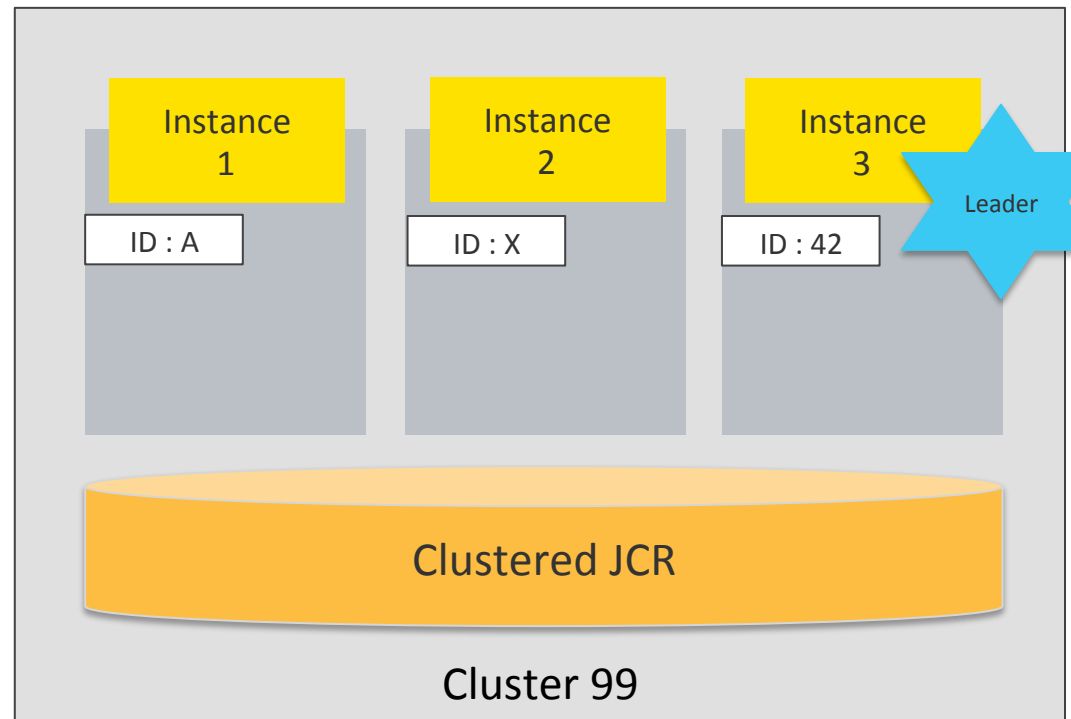
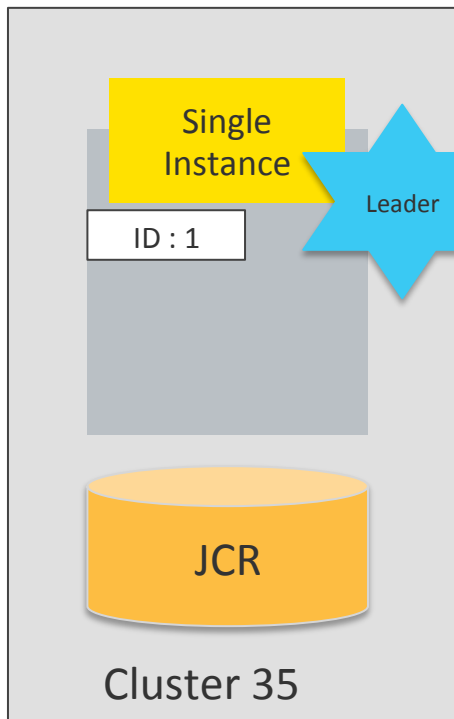
- Instance: Unique Id (Sling ID)



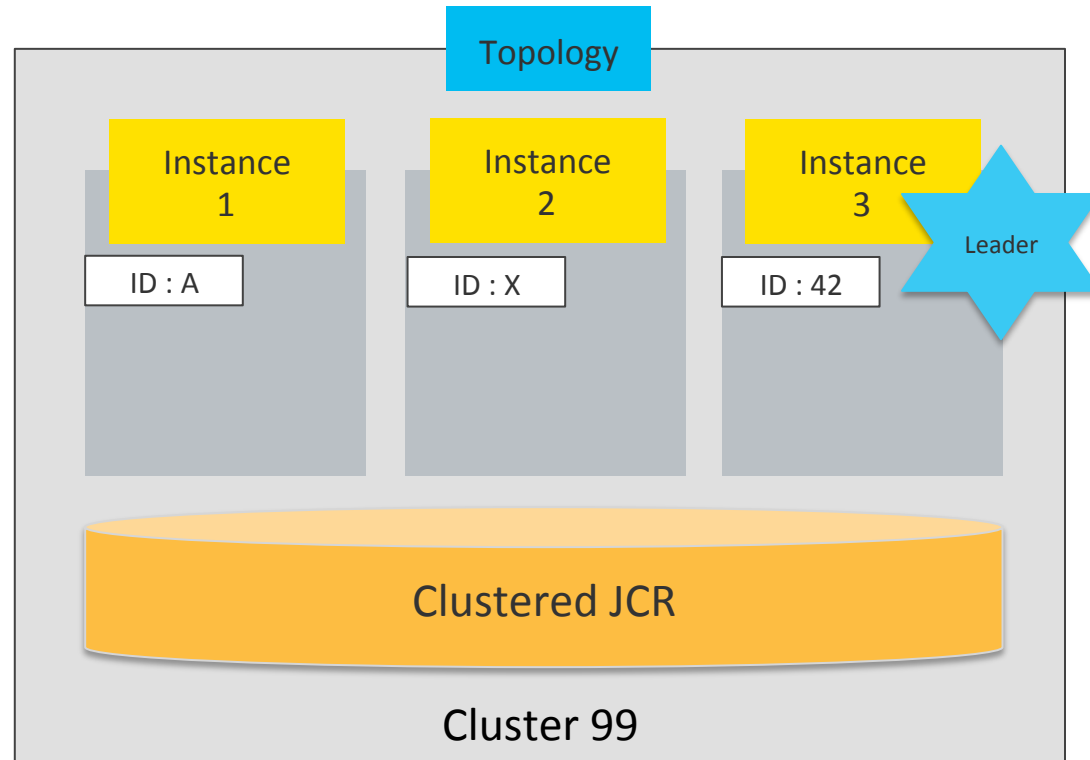
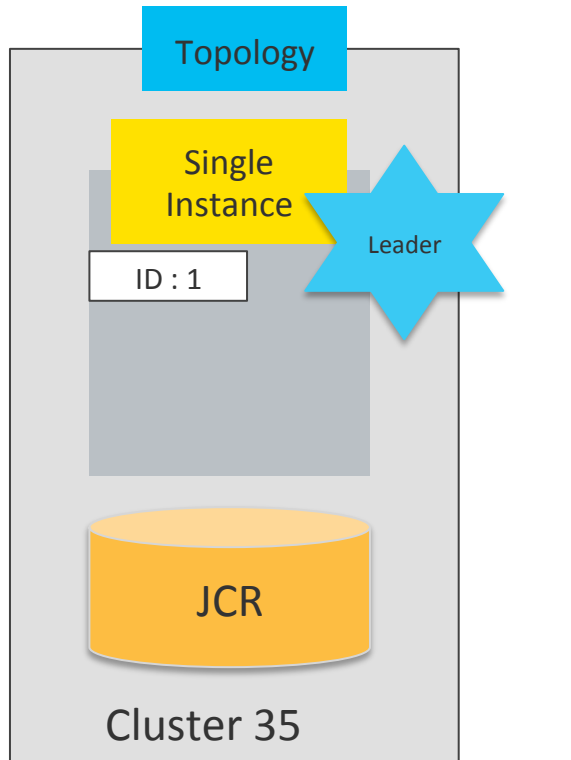
Topologies I

Apache Sling Discovery

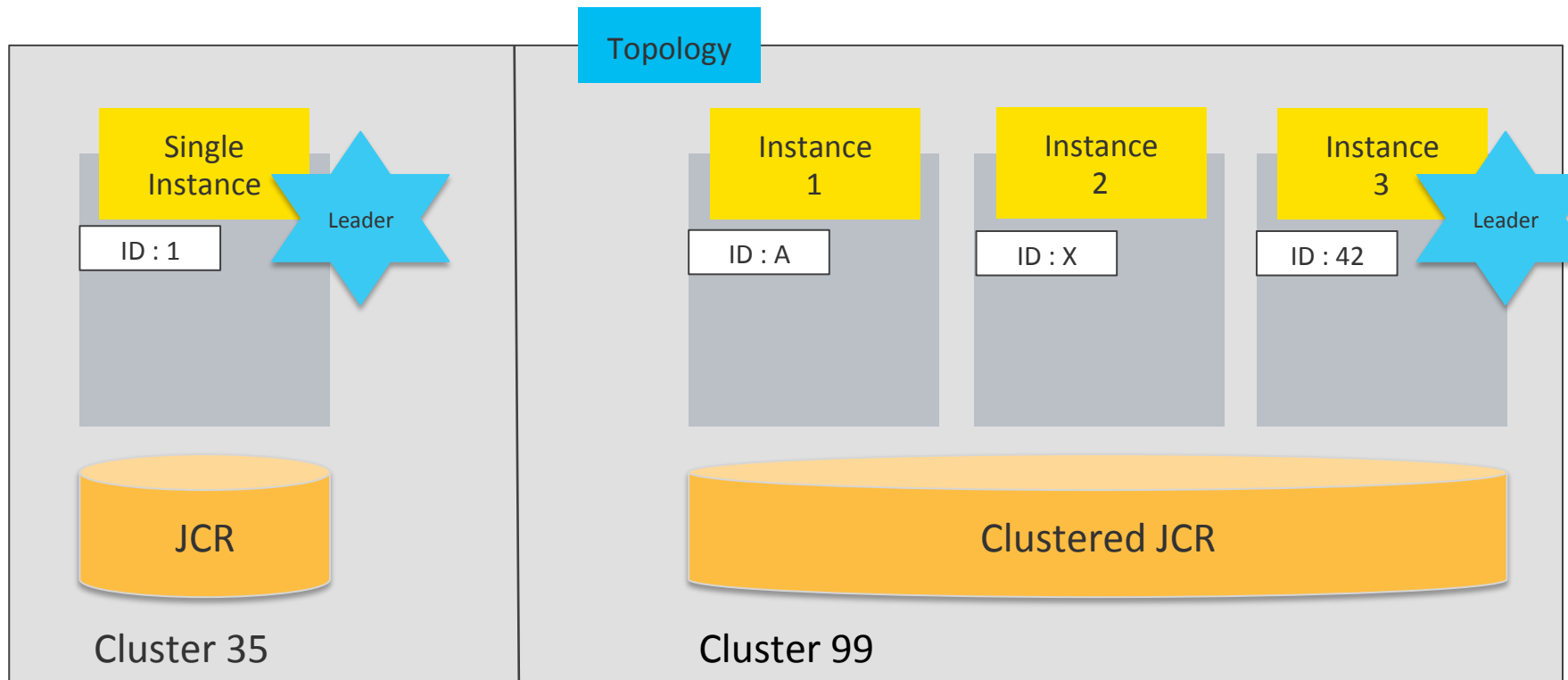
- Instance: Unique Id (Sling ID)
- Cluster: Unique Id and leader



- Topology: Set of clusters



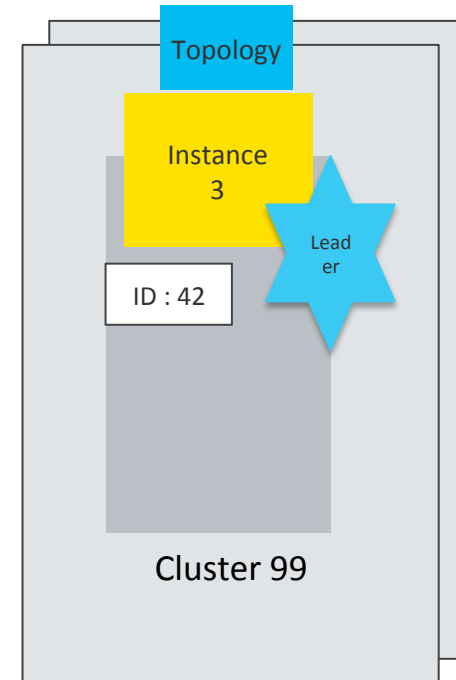
- Topology: Set of clusters



Topologies II

Apache Sling Discovery

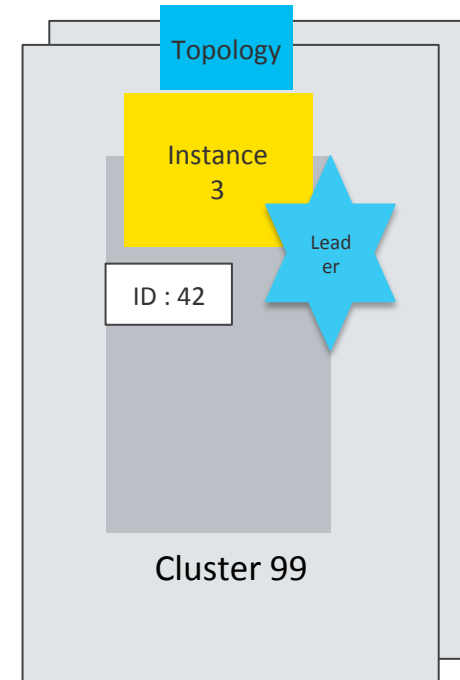
- Instance
 - Sling ID
 - Optional: Name and description
 - Belongs to a cluster
 - Might be the cluster leader
 - Additional distributed properties
 - Extensible through own services (PropertyProvider)
 - E.g. data center, region or enabled job topics



Topologies II

Apache Sling Discovery

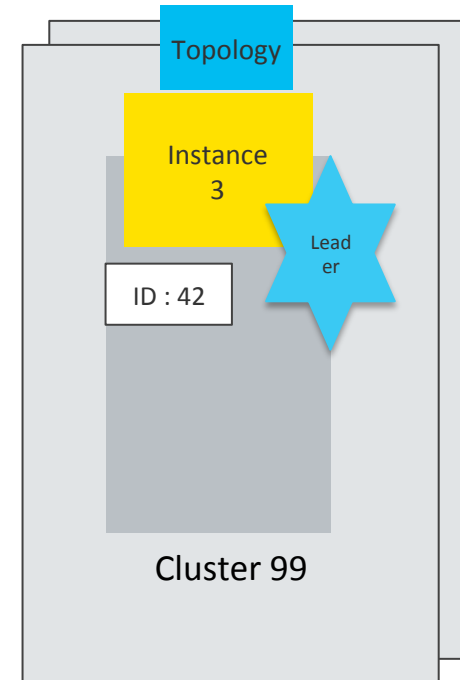
- Cluster
 - Elects (stable) leader
 - Stable instance ordering



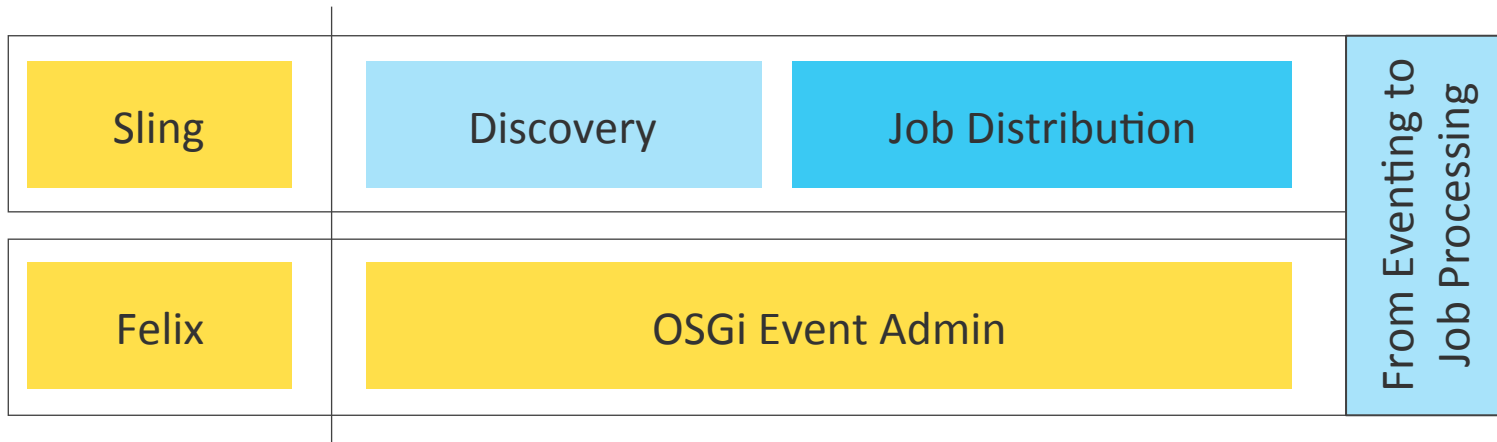
Topologies II

Apache Sling Discovery

- TopologyEventListener
- Receives events on topology changes
 - Topology is changing
 - Topology changed
 - Properties changed



Overview



Job Handling I

Apache Sling Job Handling

- Job : Guaranteed processing, exactly once
 - Exactly one job consumer
- Started by client code, e.g. for replication, workflow...
 - Job topic
 - Payload is a serializable map

Job Handling I

Apache Sling Job Handling

- Sling Job Manager handles and distributes jobs
 - Delivers job to a job consumer...
 - ...and waits for response
 - Retry and failover
- Notification listeners (fail, retry, success)

Starting / Processing a Job | Apache Sling Job Handling

Starting a job

```
public interface JobManager {  
    Job addJob(String topic, String optionalName, Map<String, Object> properties);  
    ...  
}
```

Processing a job

```
public interface JobConsumer {  
    String PROPERTY_TOPICS = "job.topics";  
  
    enum JobResult {  
        OK,  
        FAILED,  
        CANCEL,  
        ASYNC  
    }  
  
    JobResult process(Job job);  
}
```

Note: Starting/processing of jobs through Event Admin is ~~deprecated~~ but still supported

Starting / Processing a Job II

Apache Sling Job Handling

```
@Component
@Service(value={JobConsumer.class})
@Property(name=JobConsumer.PROPERTY_TOPICS, value="org/apache/sling/jobs/backup")
public class BackupJobConsumer
    implements JobConsumer {

    @Override
    public JobResult process(final Job job) {
        // do backup
        return JobResult.OK;
    }
}
```


Job Handling I

Apache Sling Job Handling

- New jobs are immediately persisted
- Jobs are “pushed” to the processing instance
- Processing instances use different queues
 - Associated with job topic(s)
 - Main queue
 - 0..n custom queues

Job Queue I

Apache Sling Job Handling

- Queue is configurable
- Queue is started on demand in own thread
 - And stopped if unused for some time

Job Queue II

Apache Sling Job Handling

- Queue Types
 - Ordered queue
 - Parallel queues: Plain and Topic Round Robin

Job Queue III

Apache Sling Job Handling

- Limit for parallel threads per queue
- Number of retries (-1 = endless)
- Retry delay
- Thread priority

Additional Configurations

Apache Sling Job Handling

- Job Manager Configuration = Main Queue Configuration
 - Maximum parallel jobs (15)
 - Retries (10)
 - Retry Delay
- Eventing Thread Pool Configuration
 - Used by all queues
 - Pool size (35) = Maximum parallel jobs for a single instance



Adobe Granite Web Console Jobs

Main OSGi Sling Status Web Console

Apache Sling Job Handling

Apache Sling Job Handling: Overall Statistics Restart! Reset Stats

Start Time	14:41:42:785 2013-Jul-18
Local topic consumers:	/,com/adobe/granite/workflow/offloading,com/day/cq/replication/job/*
Last Activated	-
Last Finished	-
Queued Jobs	0
Active Jobs	0
Jobs	0
Finished Jobs	0
Failed Jobs	0
Cancelled Jobs	0
Processed Jobs	0
Average Processing Time	-
Average Waiting Time	-

Topology Capabilities

com/day/cq/replication/job/*	local
com/adobe/granite/workflow/offloading	local
/	local

No active queues.

Adobe Granite Web Console Jobs



Main OSGi Sling Status Web Console

Apache Sling Job Handling

Apache Sling Job Handling: Overall Statistics Restart! Reset Stats

Start Time	14:41:42:785 2013-Jul-18
Local topic consumers:	/,com/adobe/granite/workflow/offloading,com/day/cq/replication/job/*
Last Activated	-
Last Finished	-
Queued Jobs	0
Active Jobs	0
Jobs	0
Finished Jobs	0
Failed Jobs	0
Cancelled Jobs	
Processed Jobs	
Average Processing	
Average Waiting Time	
Topology Capabilities	
com/day/cq/replication/job/*	
com/adobe/granite/workflow/offloading	
com/day/cq/replication/job/*	local
com/adobe/granite/workflow/offloading	local
/	local

No active queues.

Job Distribution I

Apache Sling Job Distribution

- Each instance determines enabled job topics
 - Derived from Job Consumers (new API required)
 - Can be whitelisted/blacklisted (in Job Consumer Manager)
 - Announced through Topology

Job Distribution II

Apache Sling Job Distribution

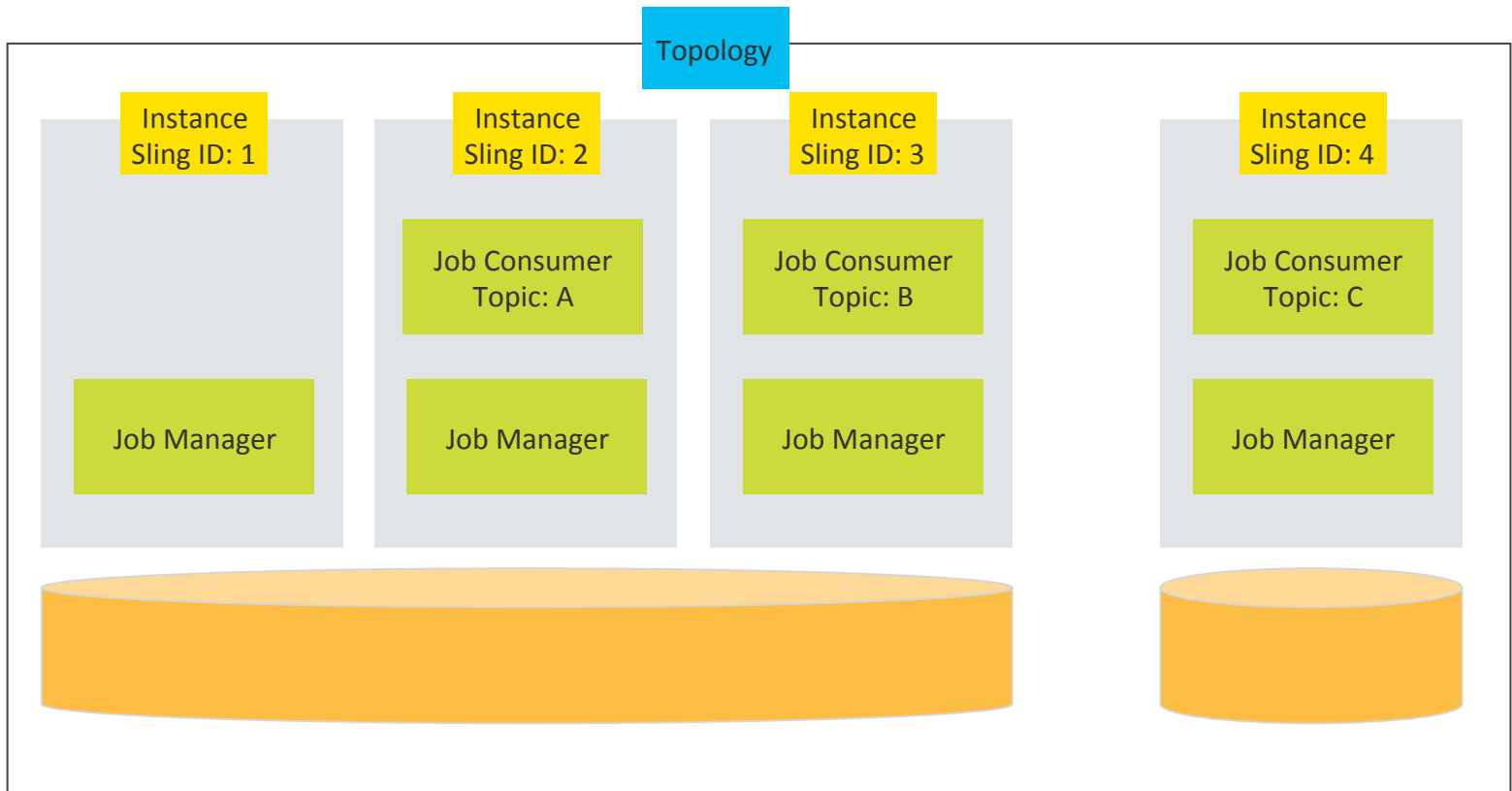
- Job Distribution depends on enabled job topics and queue type
 - Potential set of instances derived from topology (enabled job topics)
 - Ordered: processing on leader only, one job after the other
 - Parallel: Round robin distribution on all potential instances
 - Local cluster instances have preference

Job Distribution III

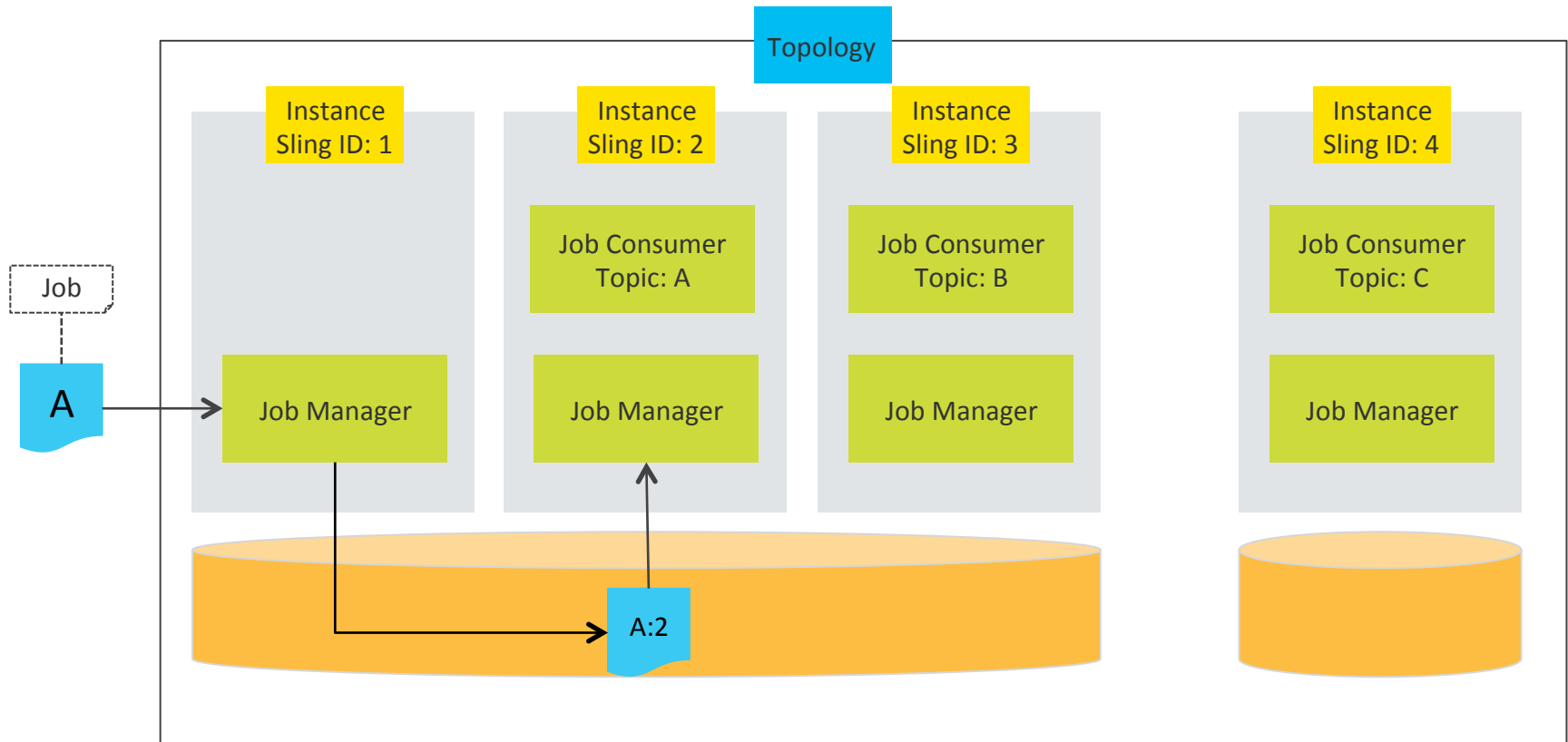
Apache Sling Job Distribution

- Failover
 - Instance crash: leader redistributes jobs to available instances
 - Leader change taken into account
- On enabled job topics changes: potential redistribution

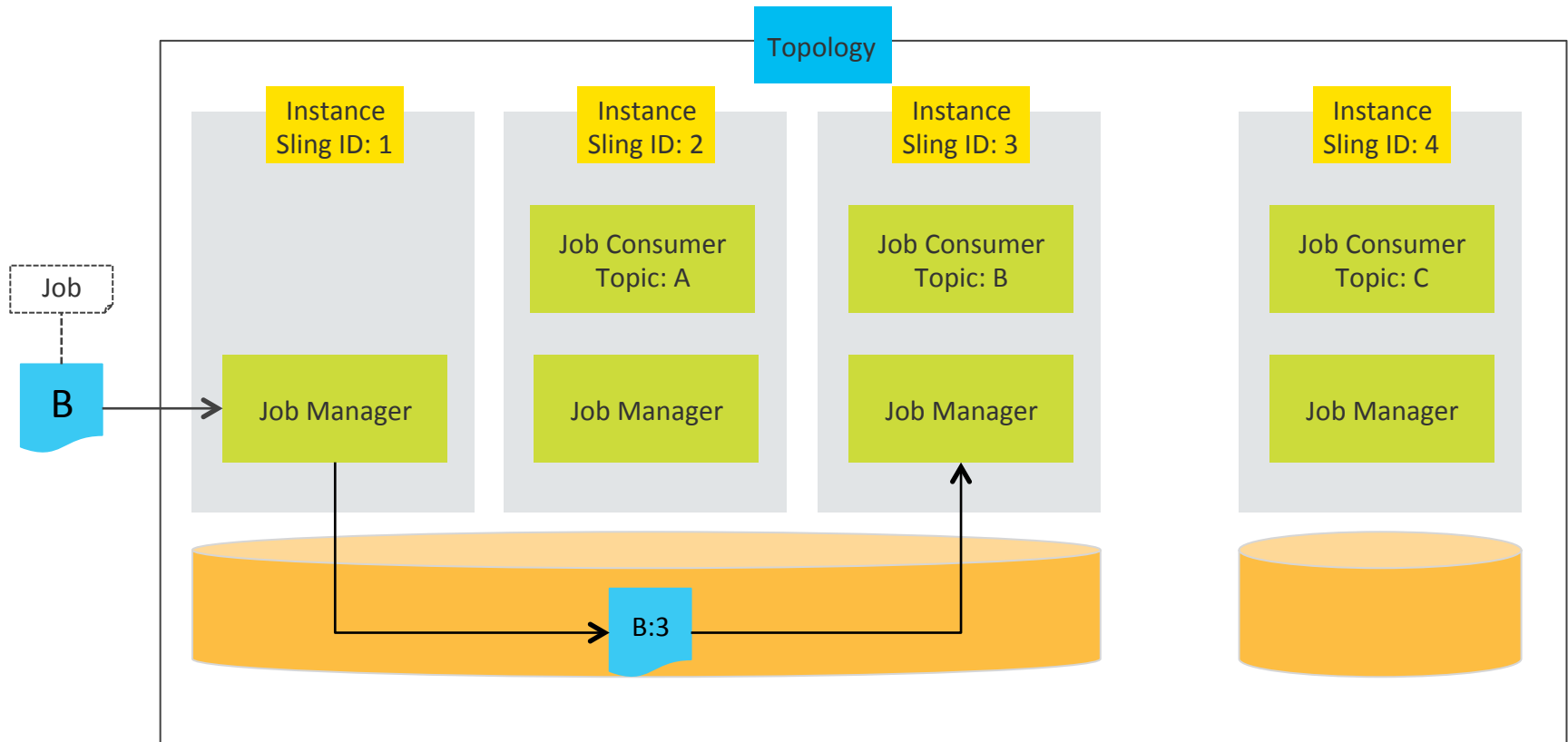
Sling Job Distribution



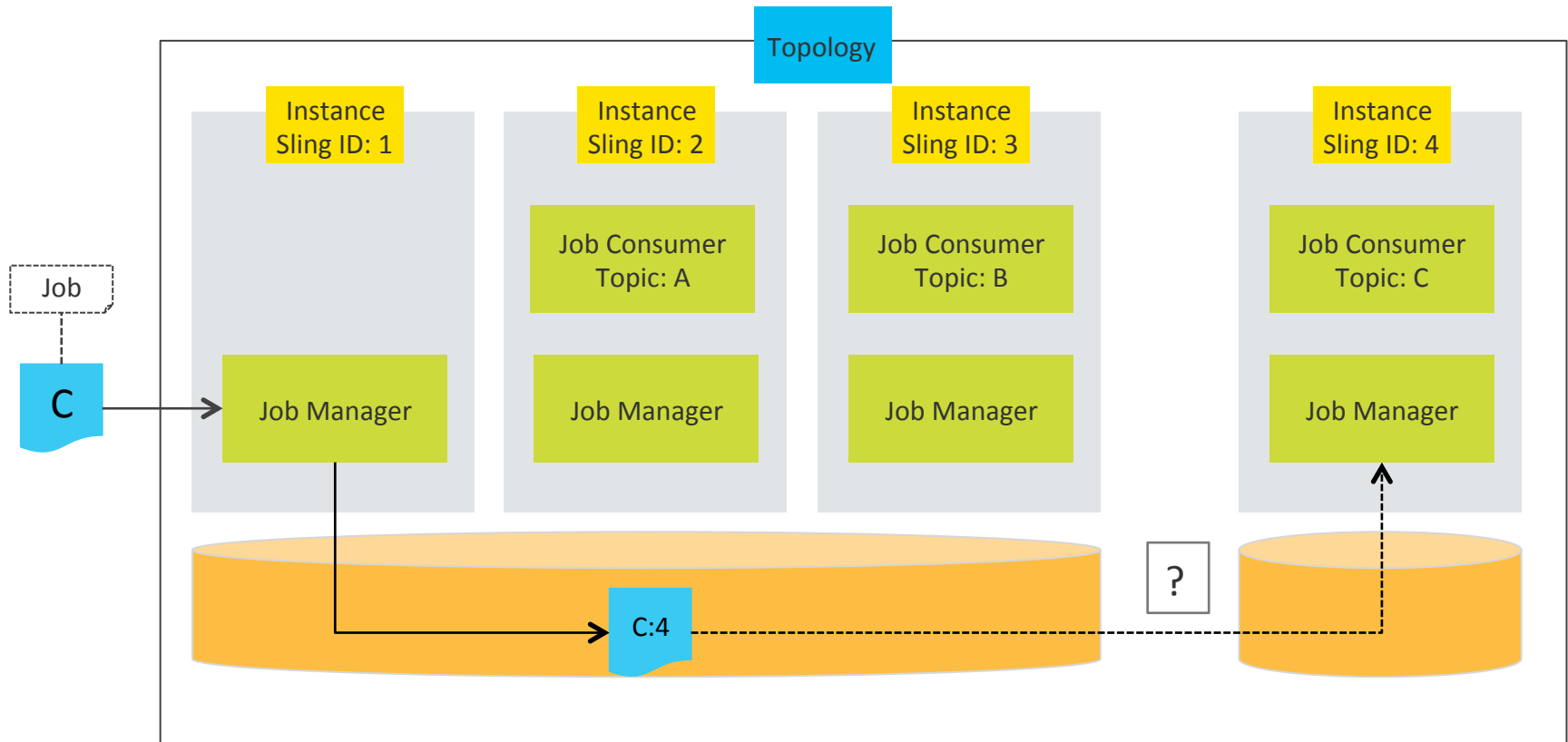
Sling Job Distribution



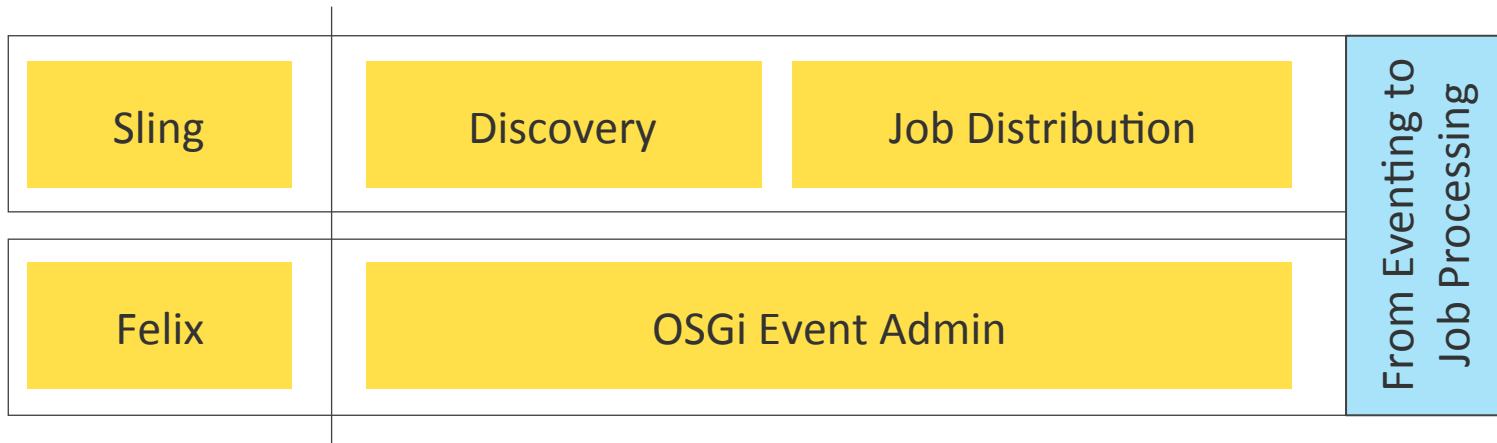
Sling Job Distribution



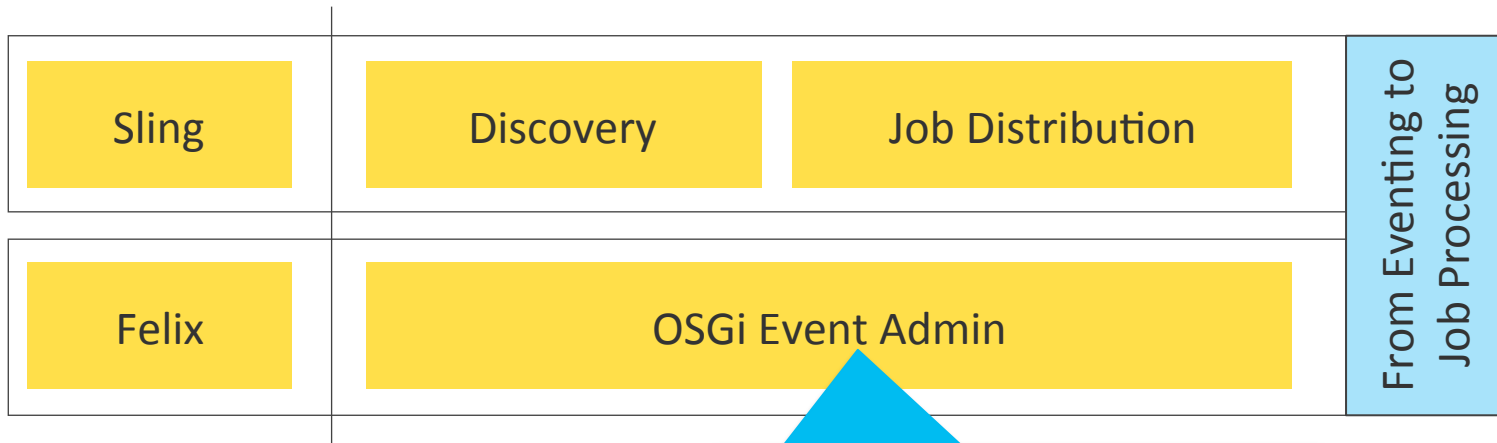
Sling Job Distribution



Discovery and Eventing



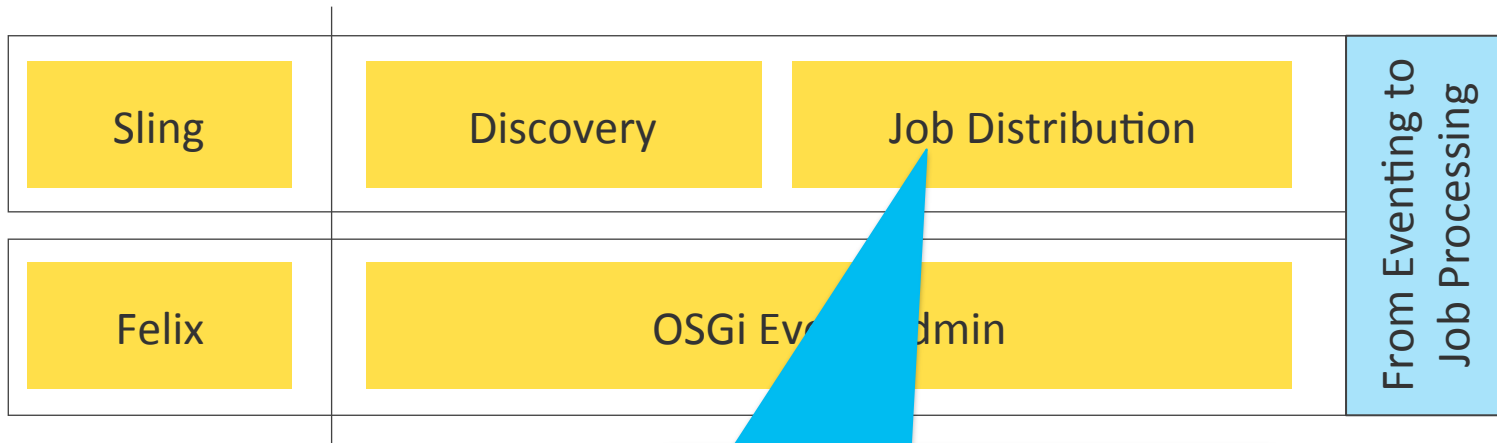
Discovery and Eventing – What’s Next?



New OSGi Specification

- Distributed Eventing
- Cloud Computing

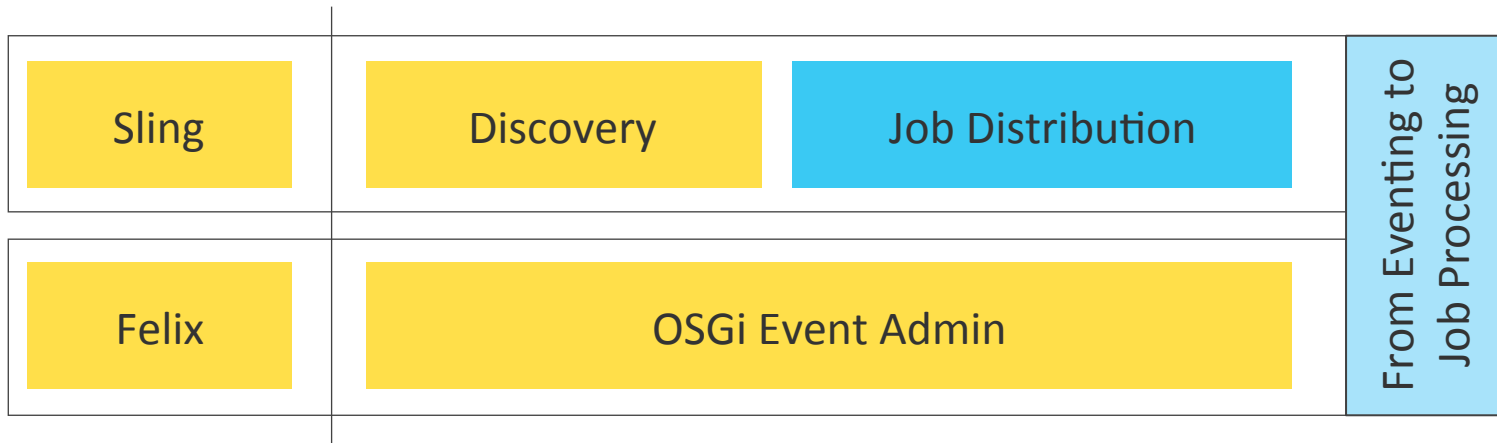
Discovery and Eventing – What’s Next?



Job Distribution

- Improved load balancing
- Pull based distribution

One More Thing...





- Jobs can inform about
 - Progress (percentage)
 - ETA
- Additional informational messages
- All information is persisted

Improved Failure Handling I

Apache Sling Job Processing

- Currently, no history of jobs
- Immediately removed once
 - Job succeeds
 - Job is cancelled
- What happened?
- What did go wrong?

- Cancelled jobs are kept
 - With a reason and log
 - Can be retried
- Successful jobs can be kept
 - With a message and log

Discovery and Eventing

