Common Workflow Service

Jet Propulsion Laboratory,
California Institute of Technology

Advanced Multi-Mission Operations System

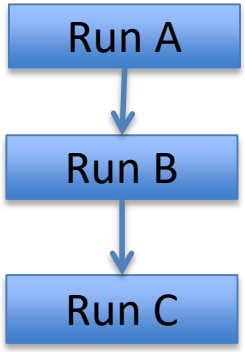
© 2016 California Institute of Technology. Government sponsorship acknowledged

Agenda

- Introduction
- CWS Features
- Demo
- Questions/Comments

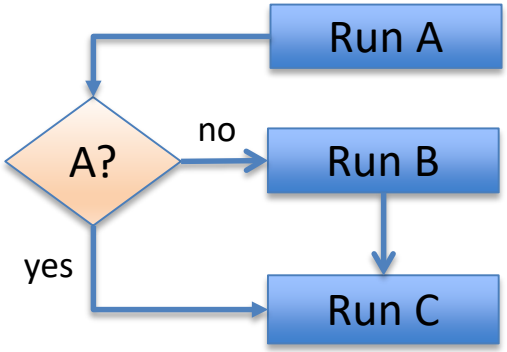
Workflow

```python
#!/usr/local/bin/python

from subprocess import call

def a():
    call("a")

def b():
    call("b")

def c():
    call("c")

a();
b();
c();
```



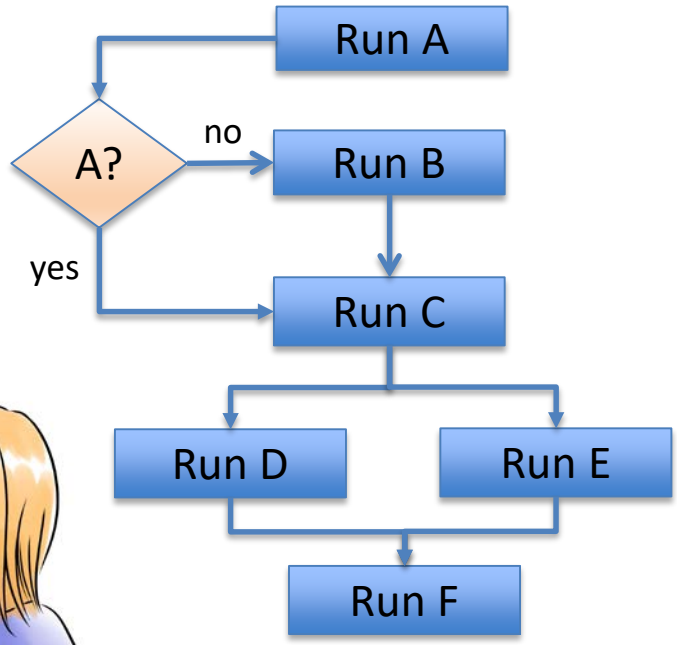
Workflow

```python
#!/usr/local/bin/python

from subprocess import call

def a():
    call("a")
    return True

def b():
    call("b")

def c():
    call("c")

aResult = a();

if not aResult:
    b();

c();
```



Workflow

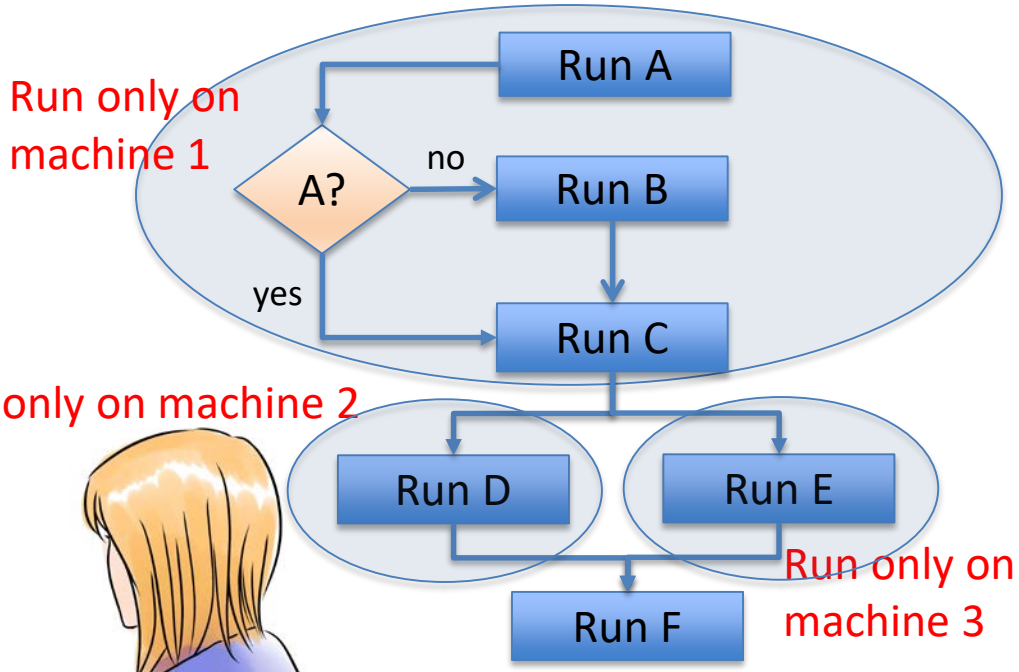
```python
#!/usr/local/bin/python

from subprocess import call
from multiprocessing import Process

def a():
    call("a")
    return True

def b():
    call("b")

def c():
    call("c")

def d():
    call("d")

def e():
    call("e")

def f():
    call("f")

aResult = a();

if not aResult:
    b();

c();

p1 = Process(target=d)
p1.start()
p2 = Process(target=e)
p2.start()
p1.join()
p2.join()

f();
```



Workflow

```python
#!/usr/local/bin/python

from subprocess import call
from multiprocessing import Process

def a():
    call("a")
    return True

def b():
    call("b")

def c():
    call("c")

def d():
    call("d")

def e():
    call("e")

def f():
    call("f")

aResult = a();

if not aResult:
    b();

c();

p1 = Process(target=d)
p1.start()
p2 = Process(target=e)
p2.start()
p1.join()
p2.join()

f();
```

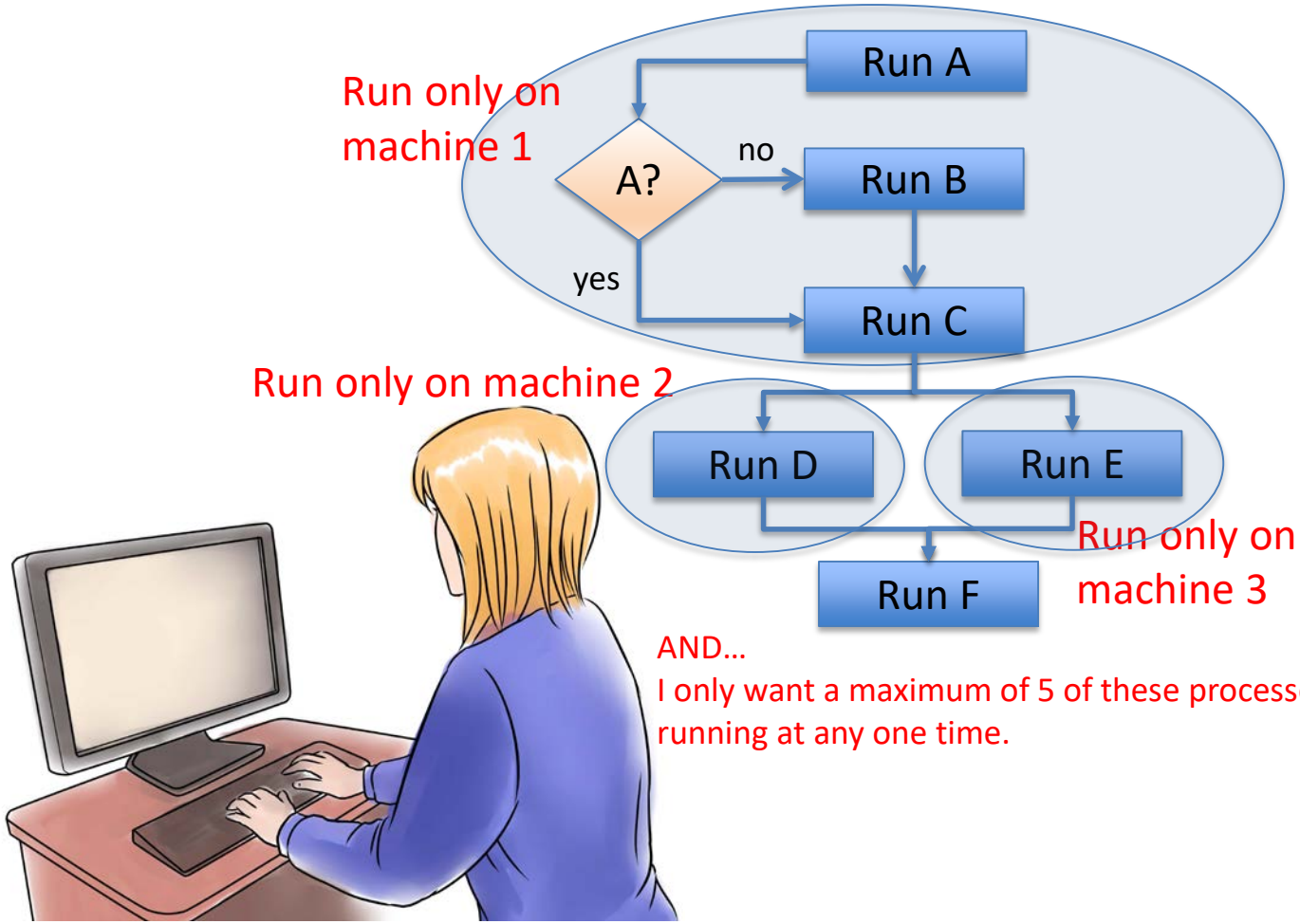
Distribution Code and Infrastructure…



Workflow

```python
#!/usr/local/bin/python

from subprocess import call
from multiprocessing import Process

def a():
    call("a")
    return True

def b():
    call("b")

def c():
    call("c")

def d():
    call("d")

def e():
    call("e")

def f():
    call("f")

aResult = a();

if not aResult:
    b();

c();

p1 = Process(target=d)
p1.start()
p2 = Process(target=e)
p2.start()
p1.join()
p2.join()

f();
```

Distribution Code and Infrastructure…

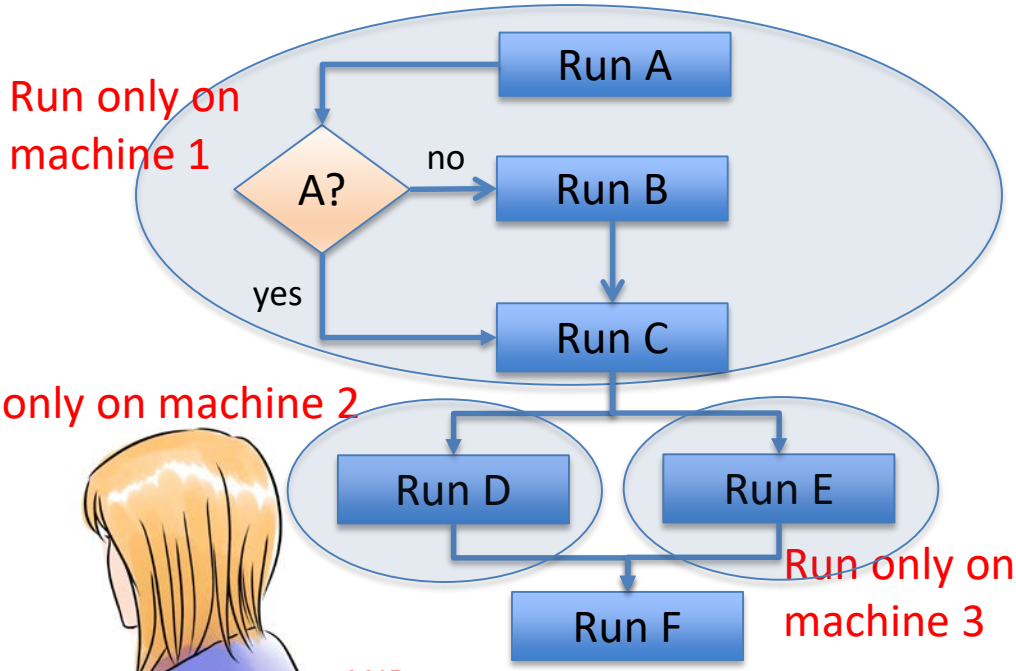


AND…
I only want a maximum of 5 of these processes running at any one time.

Workflow

```python
#!/usr/local/bin/python

from subprocess import call
from multiprocessing import Process

def a():
    call("a")
    return True

def b():
    call("b")

def c():
    call("c")

def d():
    call("d")

def e():
    call("e")

def f():
    call("f")

aResult = a();

if not aResult:
    b();

c();

p1 = Process(target=d)
p1.start()
p2 = Process(target=e)
p2.start()
p1.join()
p2.join()

f();
```

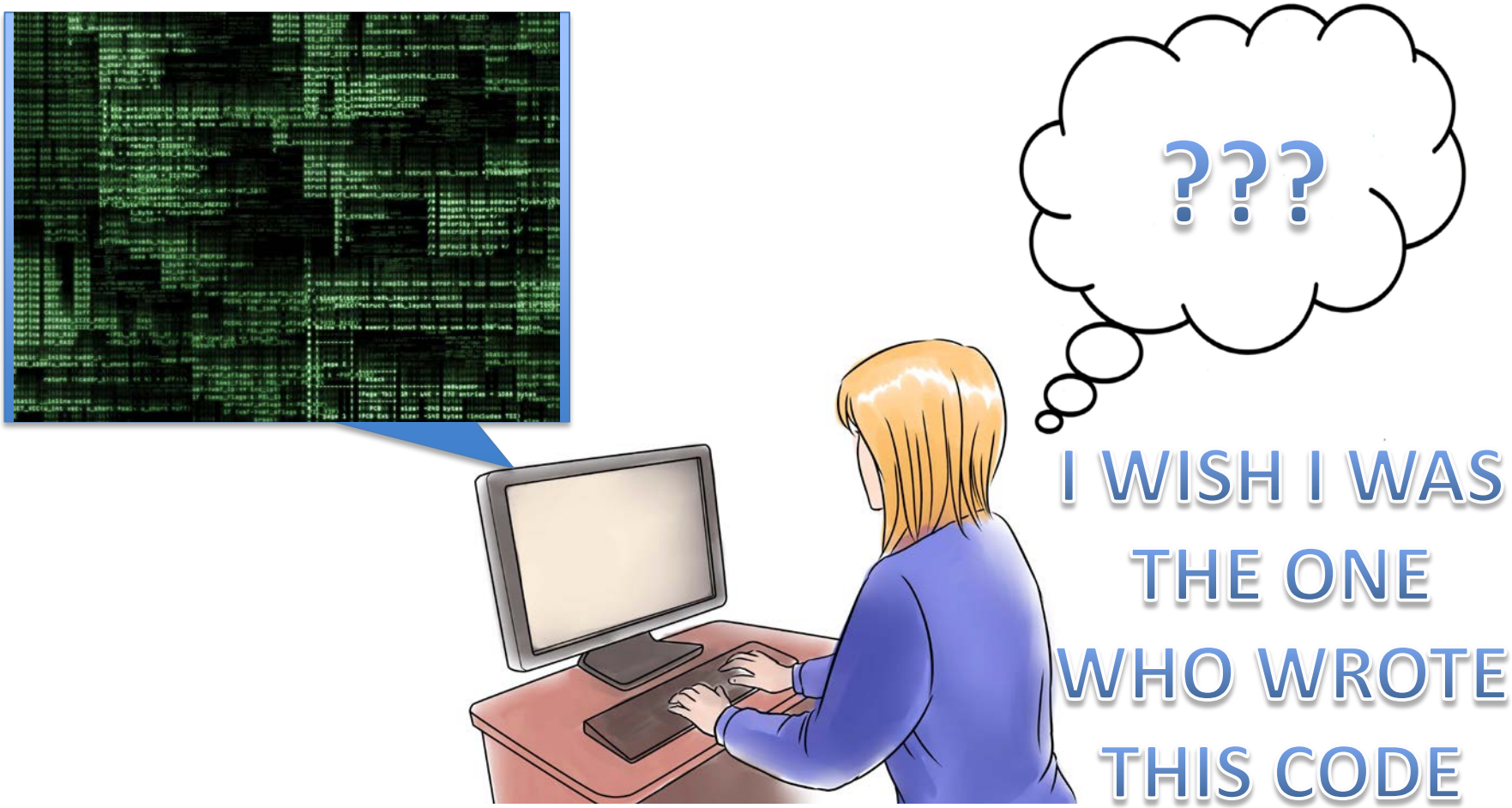
Distribution Code and Infrastructure…



~~AND…~~
~~I only want a maximum of 5 of these processes running at any one time.~~

I MEANT TO SAY “4”. Can we make that change right now? without restarting things, or updating code?

Workflow



Points to Consider

- How to do orchestration?
- Transparent and visible process model
- Agile process development/modification
- Distributed processing

Overview

What is the Common Workflow Service (CWS)?

A **standards** based Business Process Management (BPM) solution for executing and managing mission operations processes.

Overview

What is the C*Library of Common Tasks*

ommon Workflow Service (CWS)?

A **standards** based Business Process Management (BPM) solution for executing and managing mission operations processes.

CWS =

BPM Engine (Camunda BPM)

+

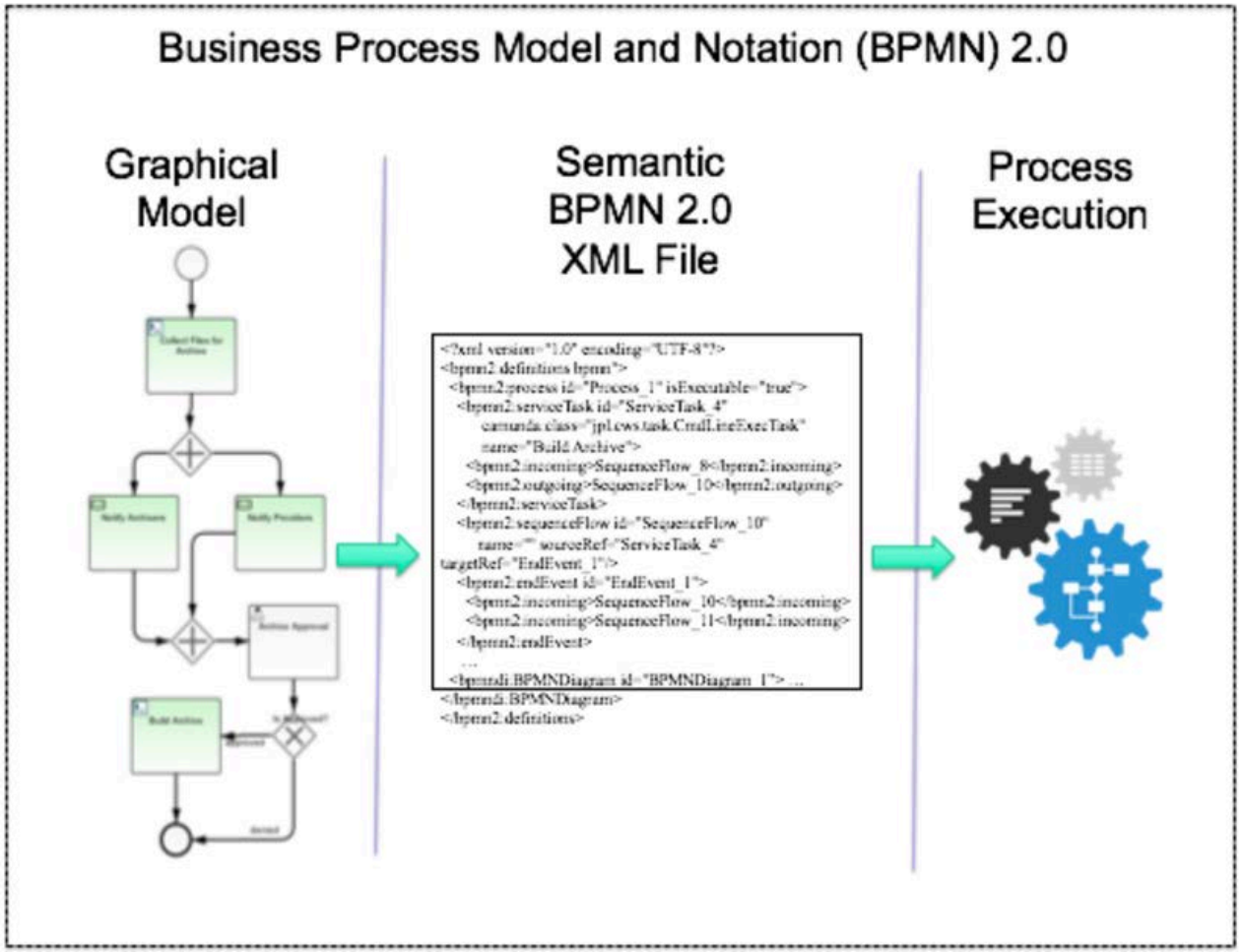
Customized BPMN 2.0 Modeler

+

Monitoring and Management Web Applications (CWS + Camunda)

*Applicable to planetary and science SDS processing pipelines, and many other use cases

The standard: BPMN 2.0

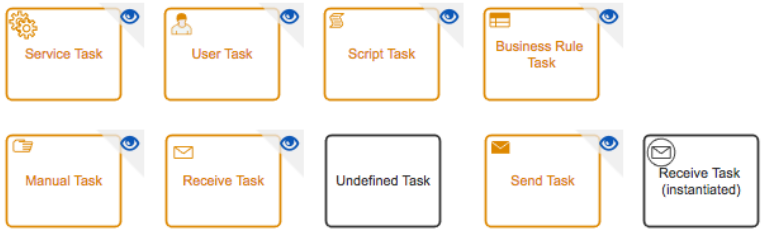


The standard: BPMN 2.0

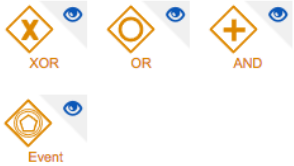
Participants

Subprocesses

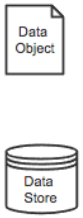
Tasks



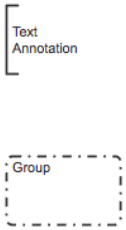
Gateways

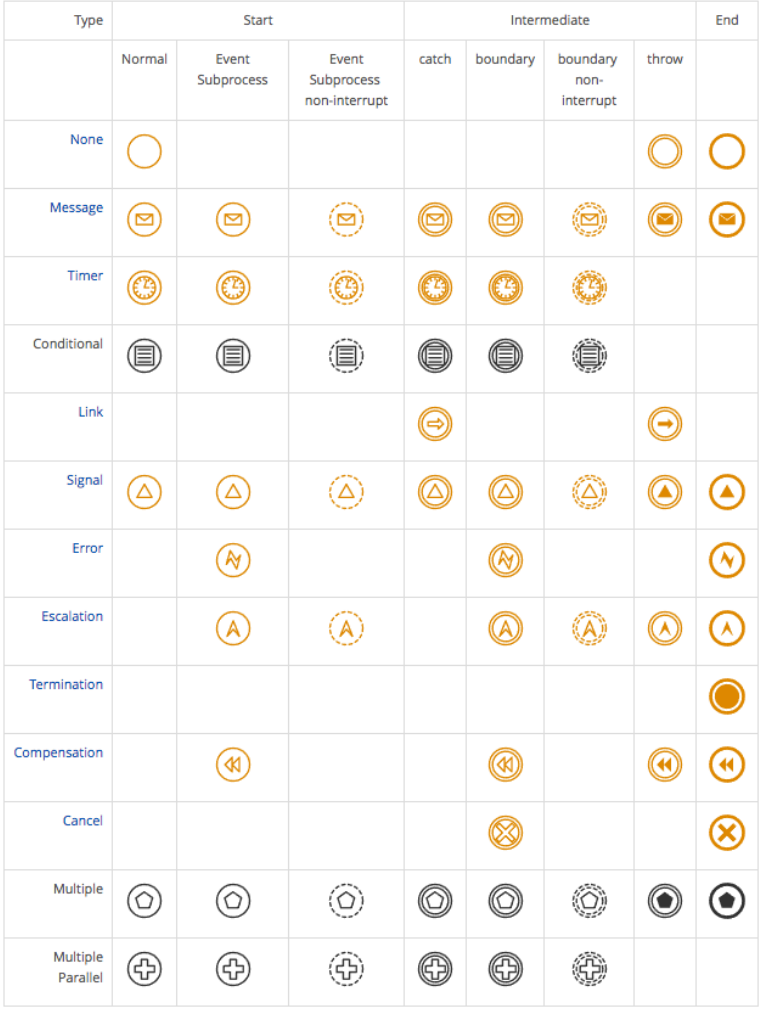


Data



Artifacts

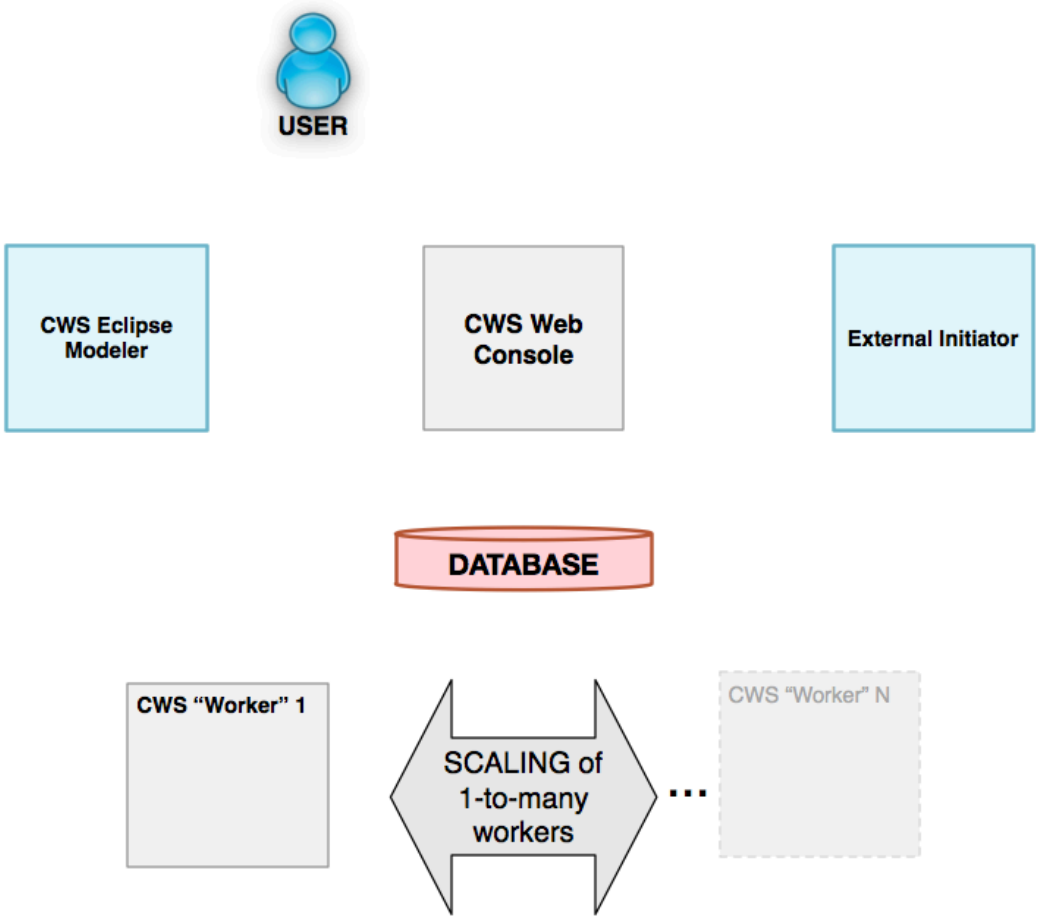




Type	Start			Intermediate				End
	Normal	Event Subprocess	Event Subprocess non-interrupt	catch	boundary	boundary non-interrupt	throw	
None								
Message								
Timer								
Conditional								
Link								
Signal								
Error								
Escalation								
Termination								
Compensation								
Cancel								
Multiple								
Multiple Parallel								

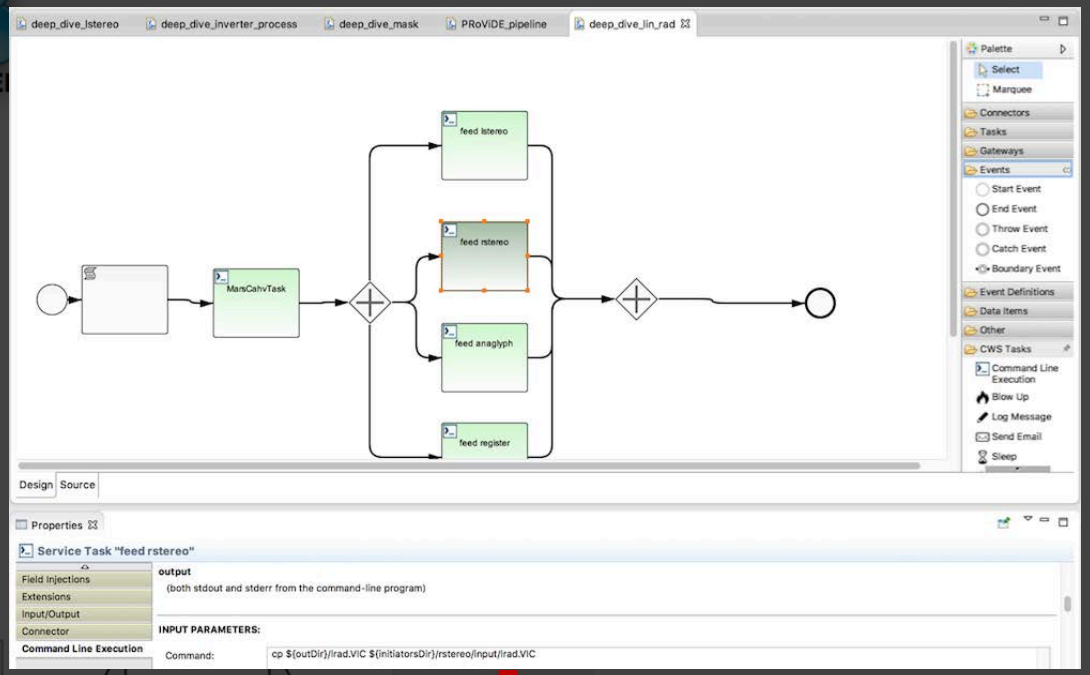
Pictures from open source Camunda BPM reference guide (https://docs.camunda.org/manual/7.5/reference/bpmn20/)

Overview


USER
CWS Eclipse Modeler
CWS Web Console
External Initiator
DATABASE
CWS "Worker" 1
SCALING of 1-to-many workers
...
CWS "Worker" N

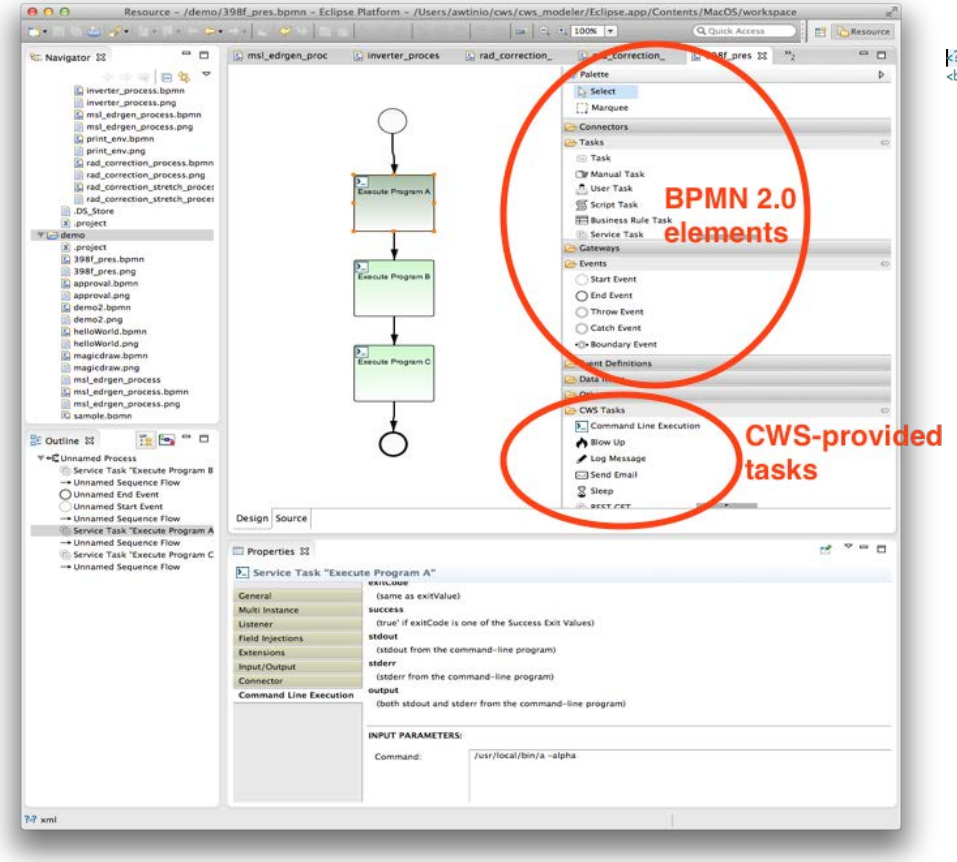
Modeling a Process Definition

Model a process, using CWS modeler application



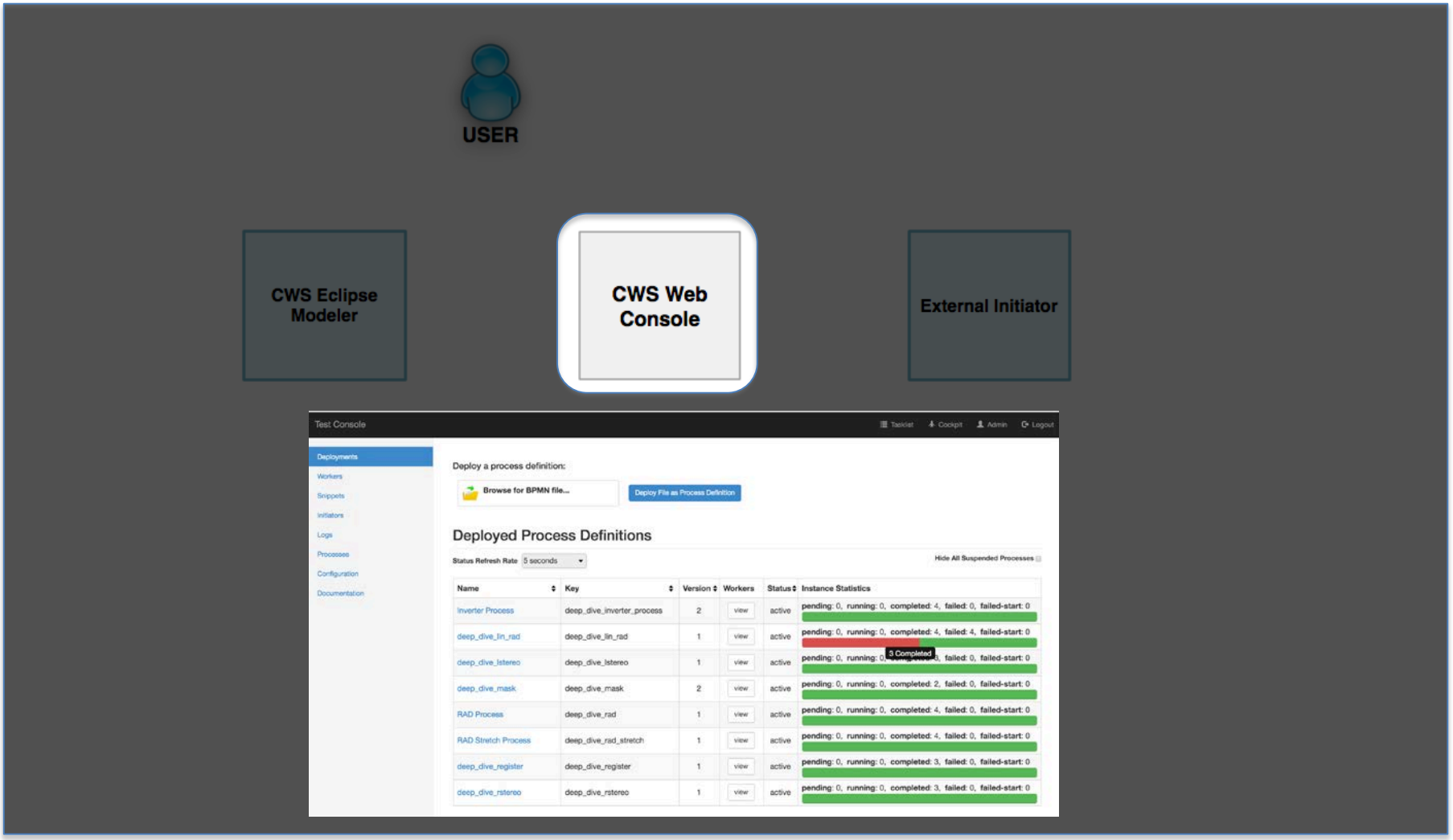
Creating a Process Definition

Process Modeler GUI – drag and drop process modeling

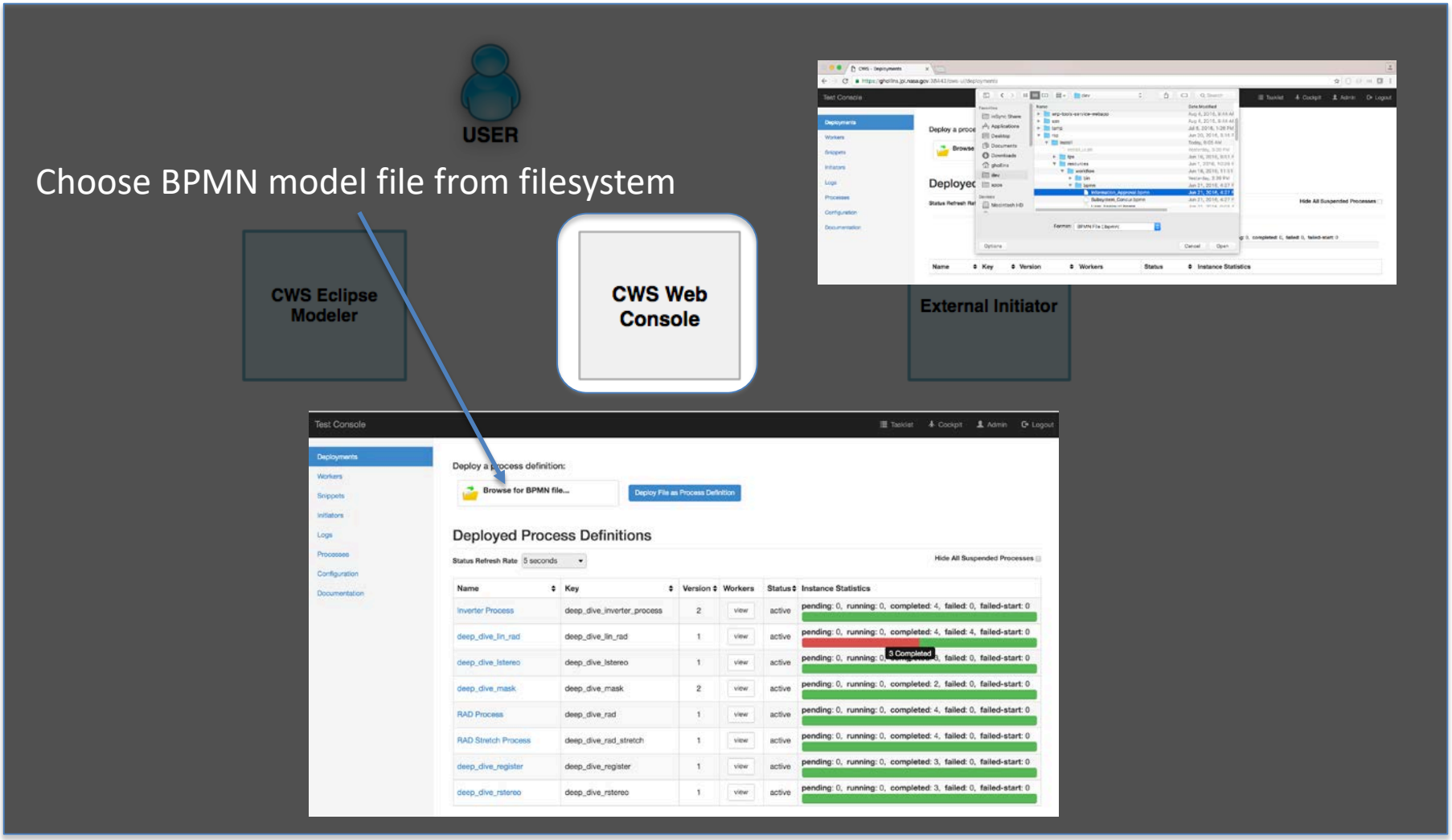



```
<?xml version="1.0" encoding="UTF-8"?>
<bpmn2:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:bpmn2="http://www.omg.org/spec/BPMN/201
  <bpmn2:process id="Process_1" isExecutable="false">
    <bpmn2:serviceTask id="ServiceTask_2" camunda:class="jpl.cws.task.CmdLineExecTask" name="Execute Program B">
      <bpmn2:extensionElements>
        <camunda:field expression="true" name="throwOnFailures"/>
        <camunda:field expression="false" name="throwOnTruncatedVariable"/>
        <camunda:field expression="ABORT_PROCESS" name="onPreConditionFail"/>
        <camunda:field name="cmdLine">
          <camunda:expression>/usr/local/bin/b -beta</camunda:expression>
        </camunda:field>
      </bpmn2:extensionElements>
      <bpmn2:incoming>SequenceFlow_2</bpmn2:incoming>
      <bpmn2:outgoing>SequenceFlow_3</bpmn2:outgoing>
    </bpmn2:serviceTask>
    <bpmn2:sequenceFlow id="SequenceFlow_3" name="" sourceRef="ServiceTask_2" targetRef="ServiceTask_3"/>
    <bpmn2:endEvent id="EndEvent_1">
      <bpmn2:incoming>SequenceFlow_4</bpmn2:incoming>
    </bpmn2:endEvent>
    <bpmn2:startEvent id="StartEvent_1">
      <bpmn2:outgoing>SequenceFlow_1</bpmn2:outgoing>
    </bpmn2:startEvent>
    <bpmn2:sequenceFlow id="SequenceFlow_1" name="" sourceRef="StartEvent_1" targetRef="ServiceTask_1"/>
    <bpmn2:serviceTask id="ServiceTask_1" camunda:class="jpl.cws.task.CmdLineExecTask" name="Execute Program A">
      <bpmn2:extensionElements>
        <camunda:field expression="true" name="throwOnFailures"/>
        <camunda:field expression="false" name="throwOnTruncatedVariable"/>
        <camunda:field expression="ABORT_PROCESS" name="onPreConditionFail"/>
        <camunda:field name="cmdLine">
          <camunda:expression>/usr/local/bin/a -alpha</camunda:expression>
        </camunda:field>
      </bpmn2:extensionElements>
      <bpmn2:incoming>SequenceFlow_1</bpmn2:incoming>
      <bpmn2:outgoing>SequenceFlow_2</bpmn2:outgoing>
    </bpmn2:serviceTask>
    <bpmn2:sequenceFlow id="SequenceFlow_2" name="" sourceRef="ServiceTask_1" targetRef="ServiceTask_2"/>
    <bpmn2:serviceTask id="ServiceTask_3" camunda:class="jpl.cws.task.CmdLineExecTask" name="Execute Program C">
      <bpmn2:extensionElements>
        <camunda:field expression="true" name="throwOnFailures"/>
        <camunda:field expression="false" name="throwOnTruncatedVariable"/>
        <camunda:field expression="ABORT_PROCESS" name="onPreConditionFail"/>
```

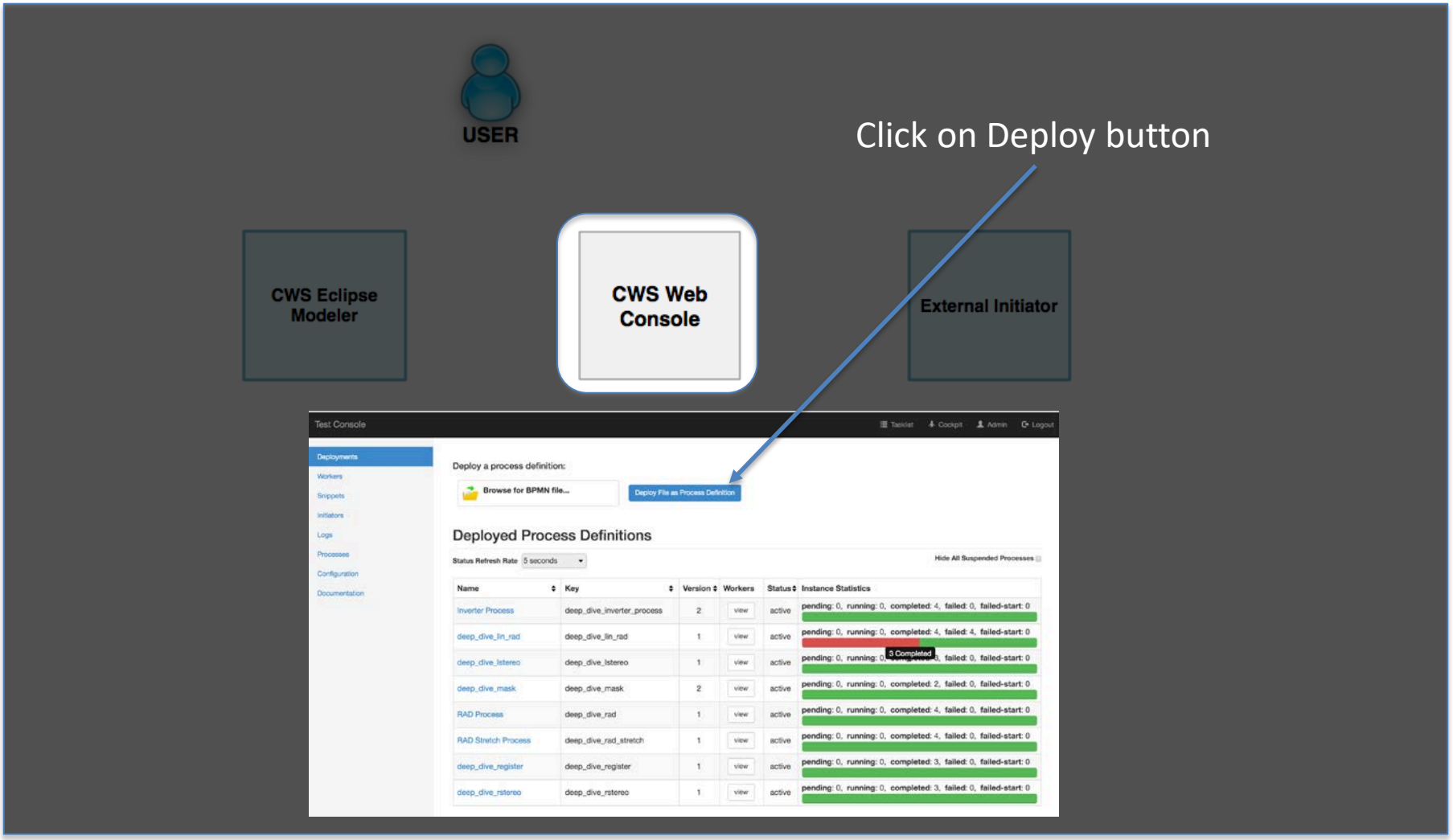
Deploying a Process Definition



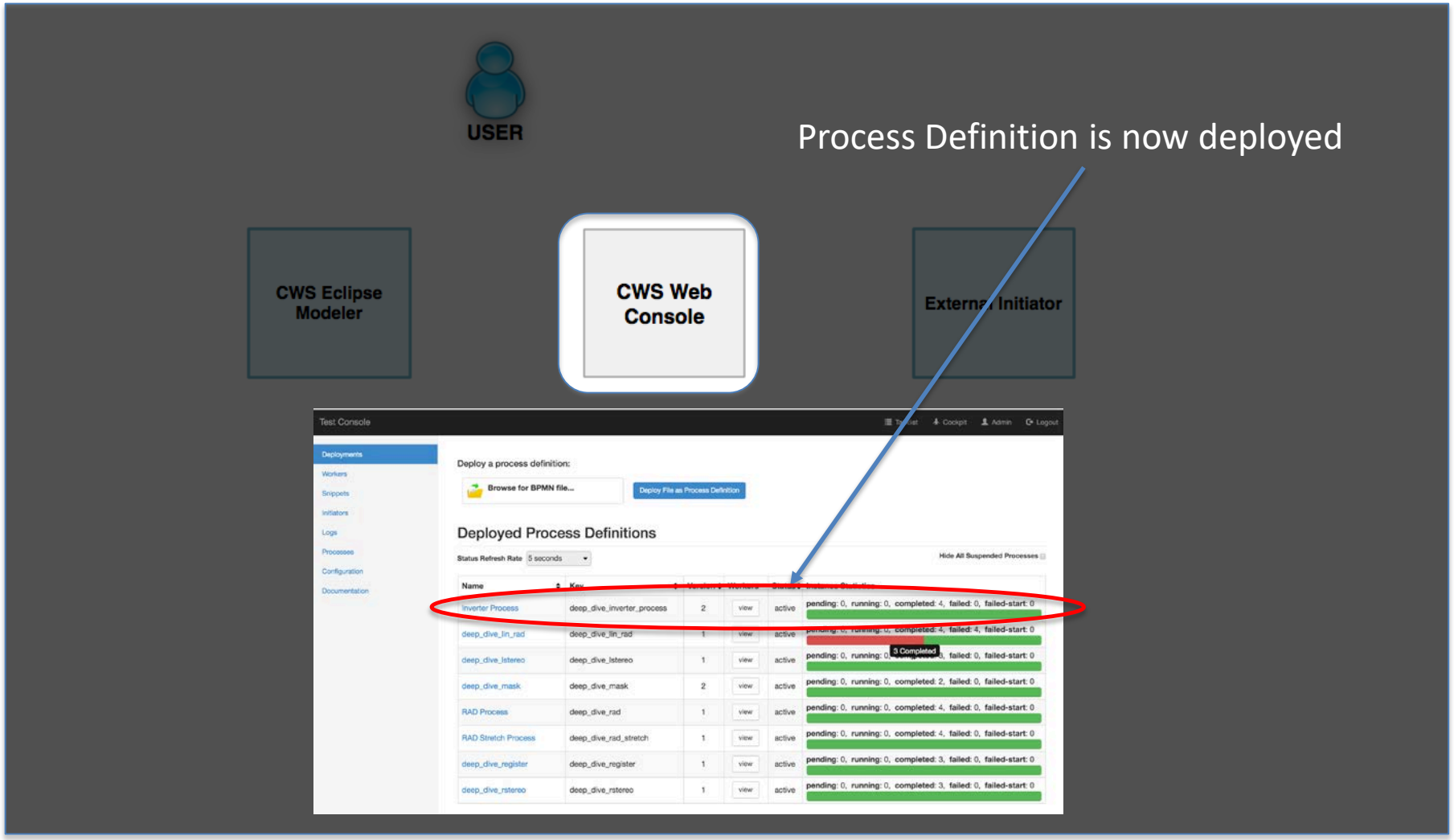
Deploying a Process Definition



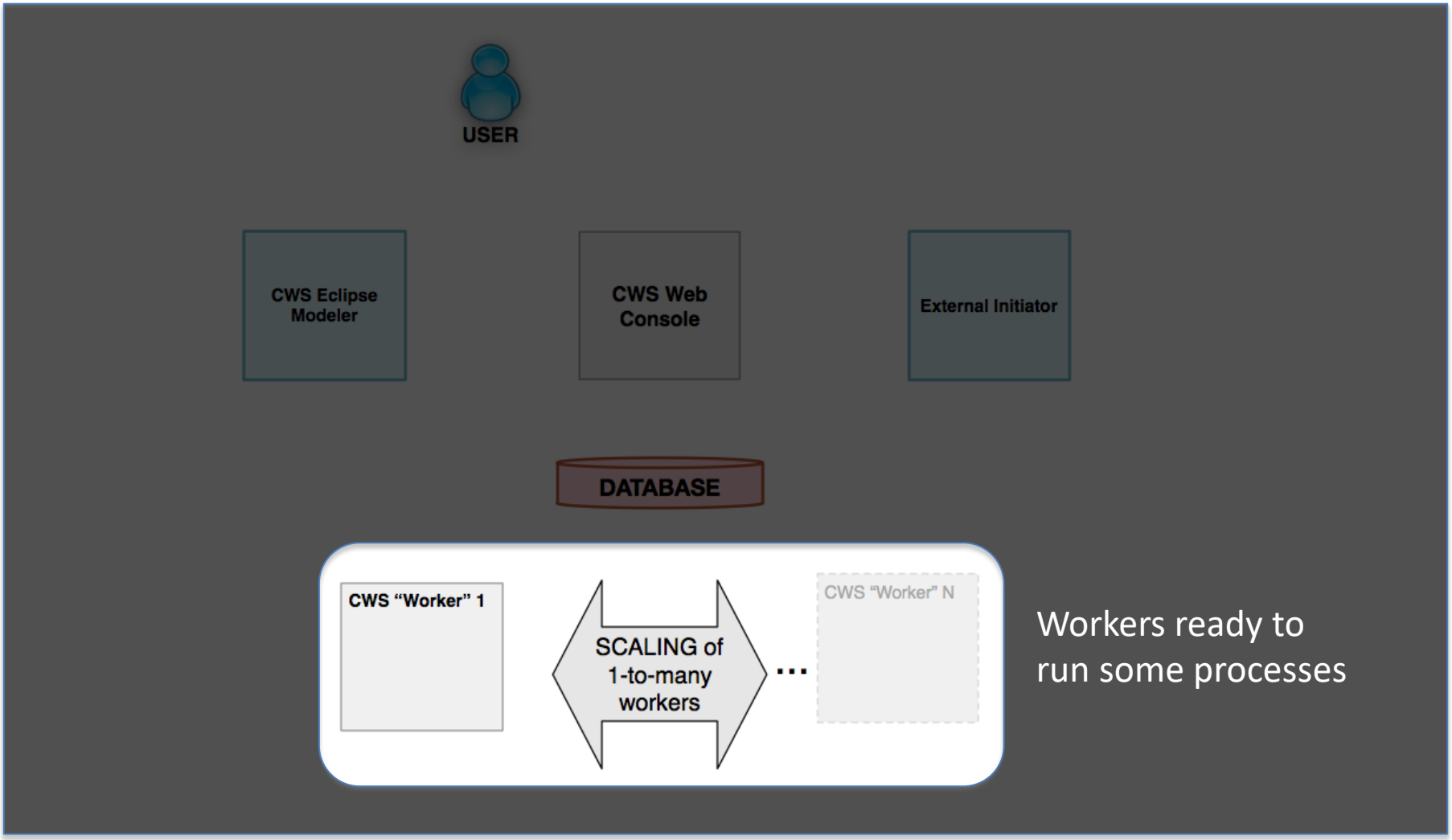
Deploying a Process Definition



Deploying a Process Definition

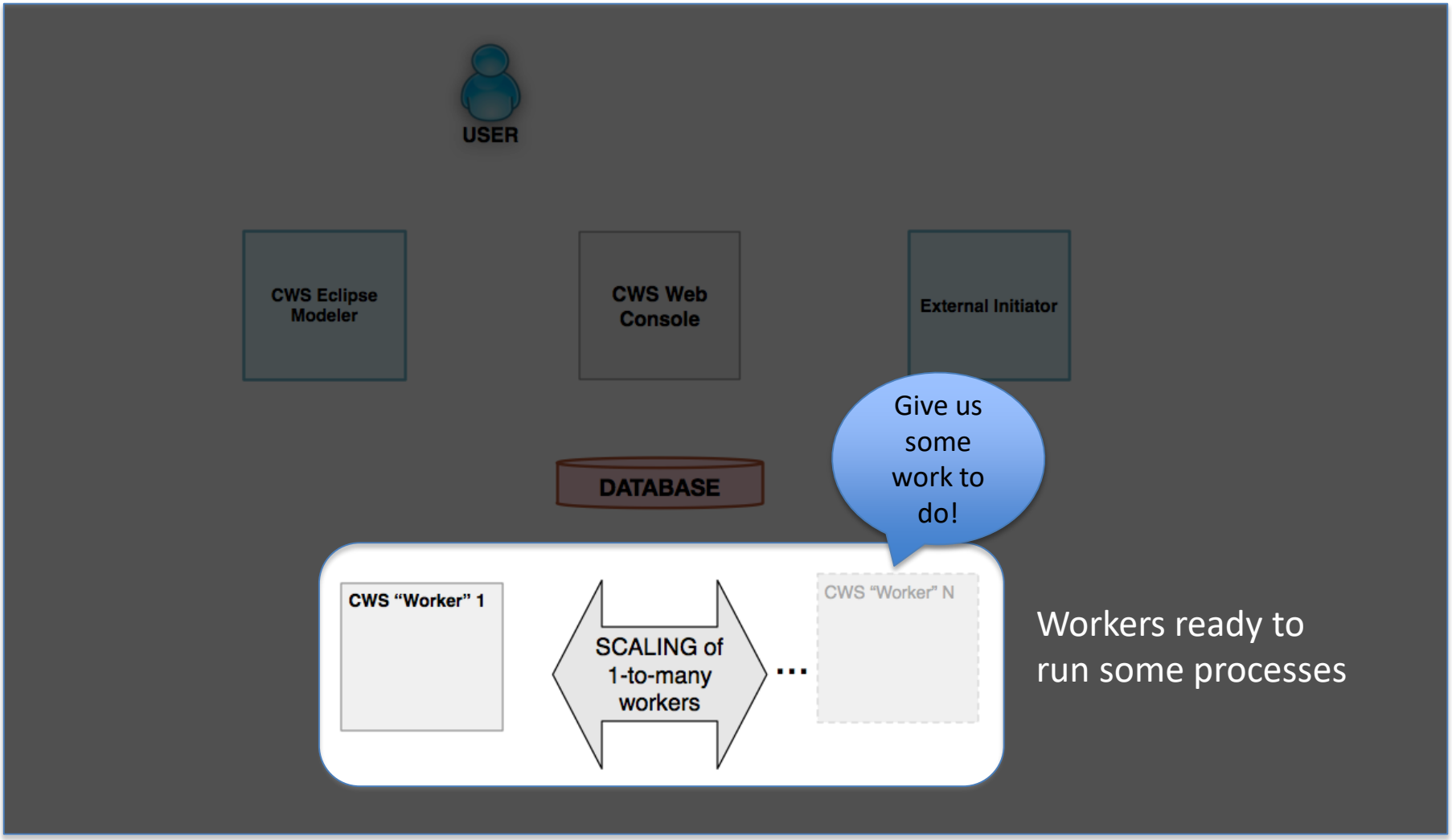


Workers

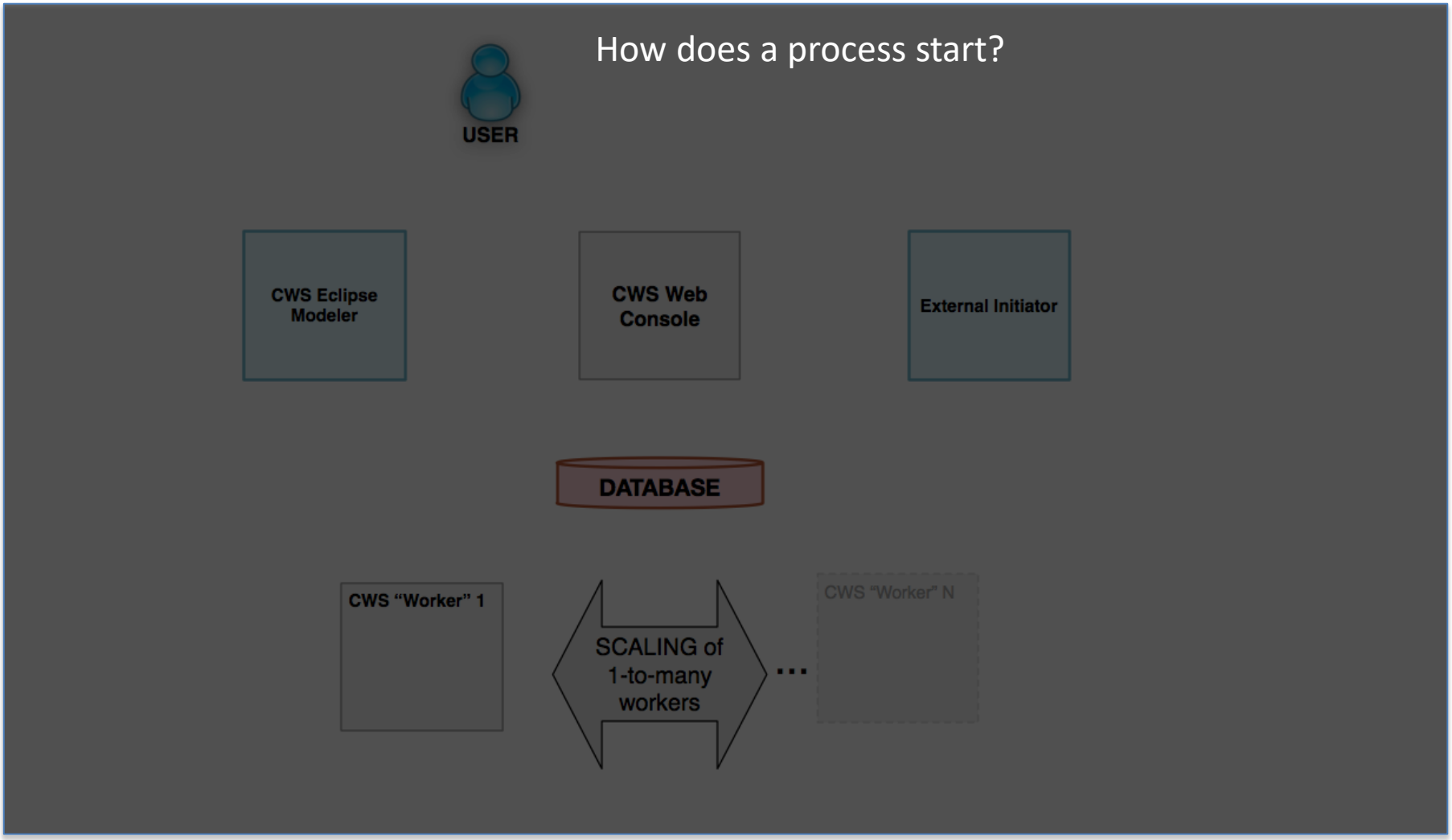


Workers ready to run some processes

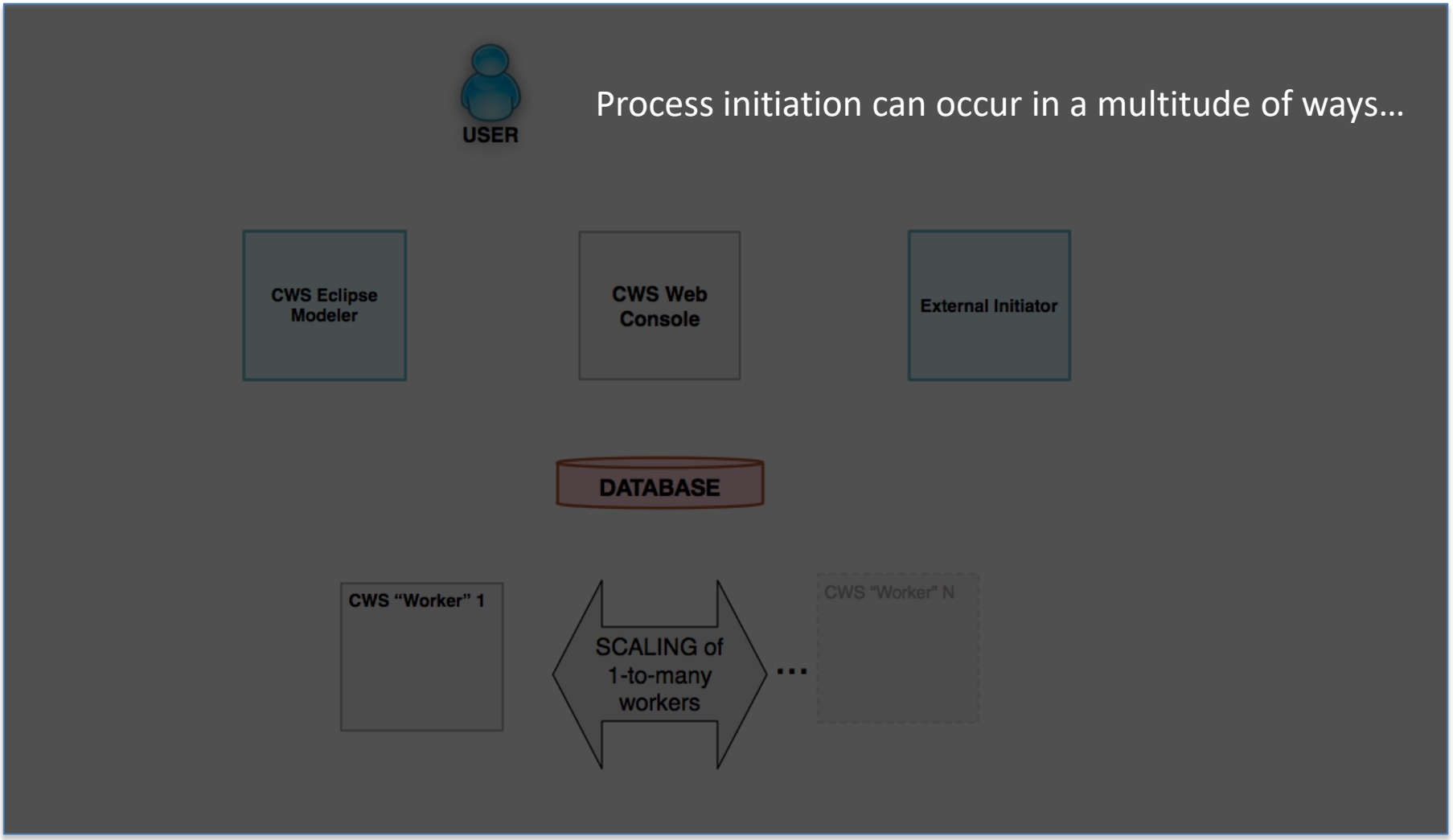
Workers


USER
CWS Eclipse Modeler
CWS Web Console
External Initiator
DATABASE
Give us some work to do!
CWS "Worker" 1
SCALING of 1-to-many workers
...
CWS "Worker" N
Workers ready to run some processes

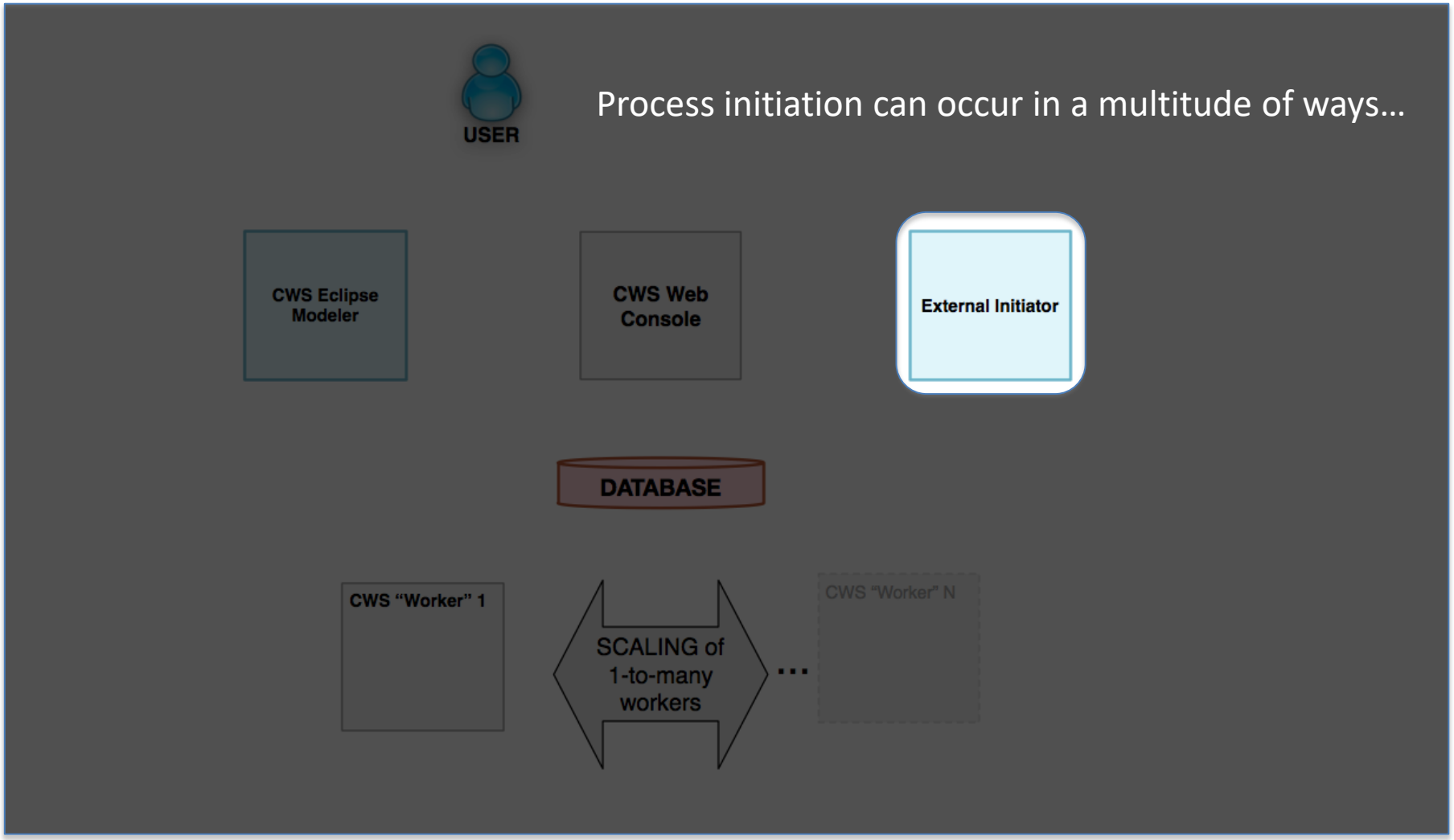
Process Initiation



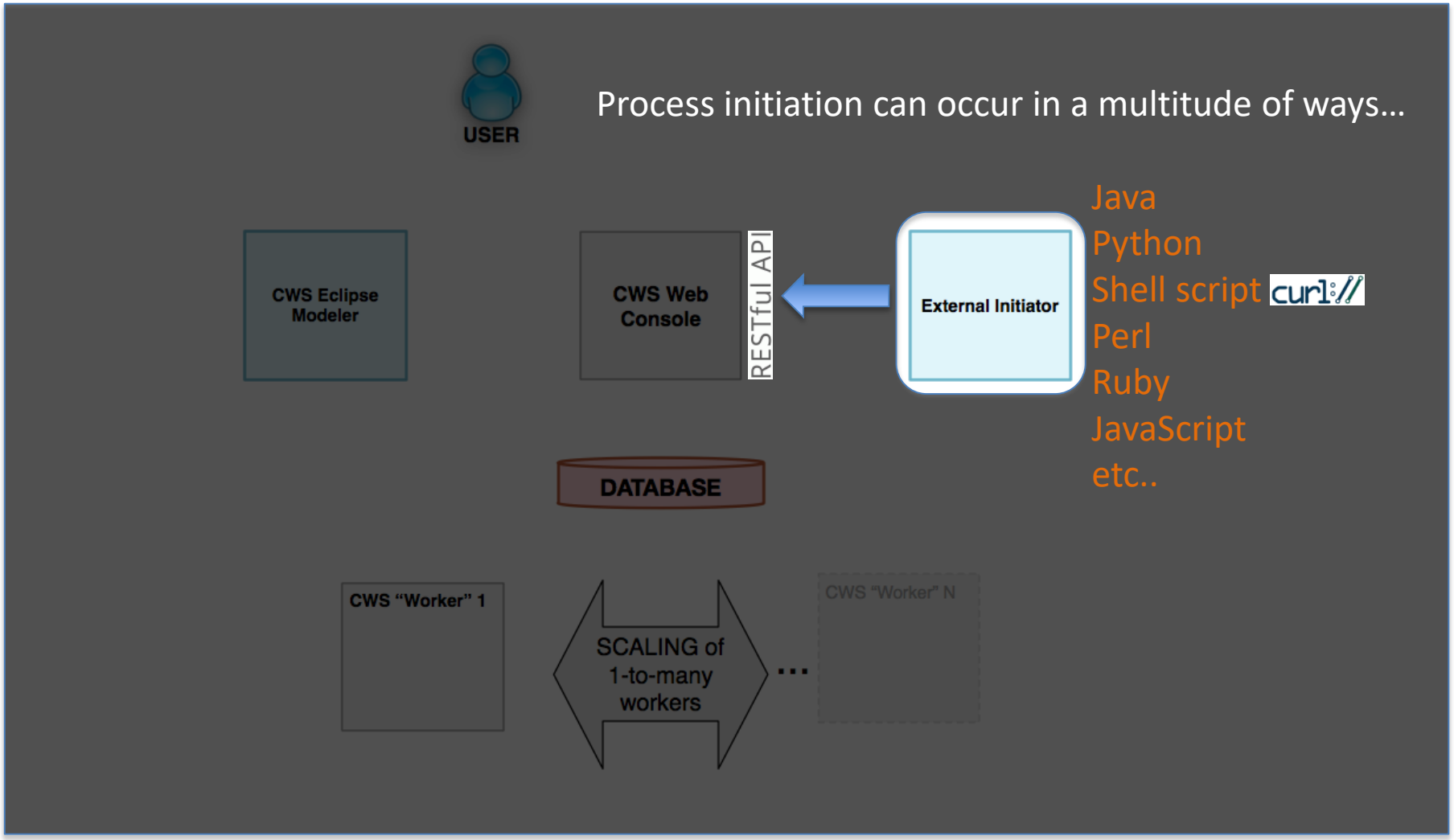
Process Initiation



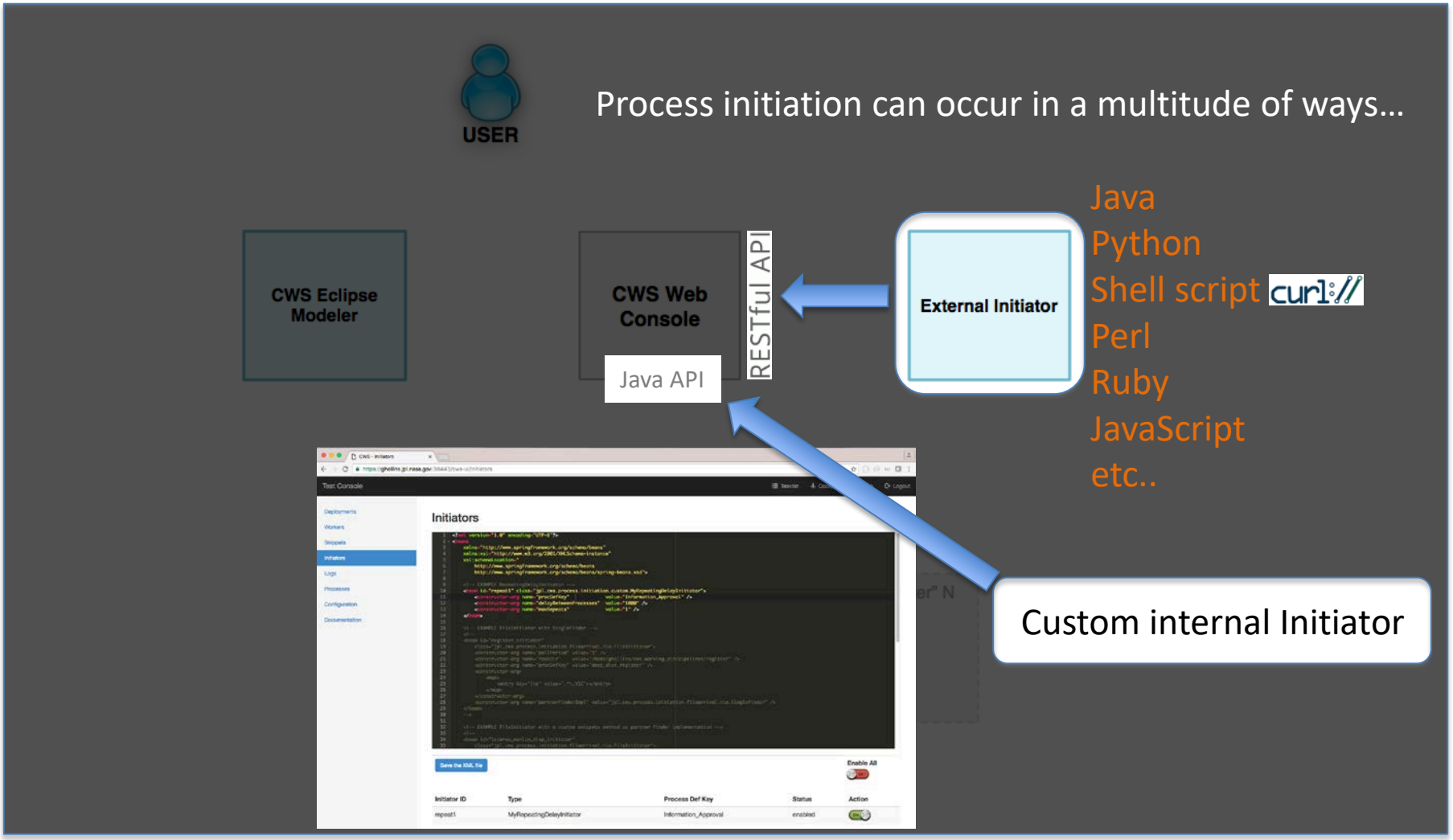
Process Initiation



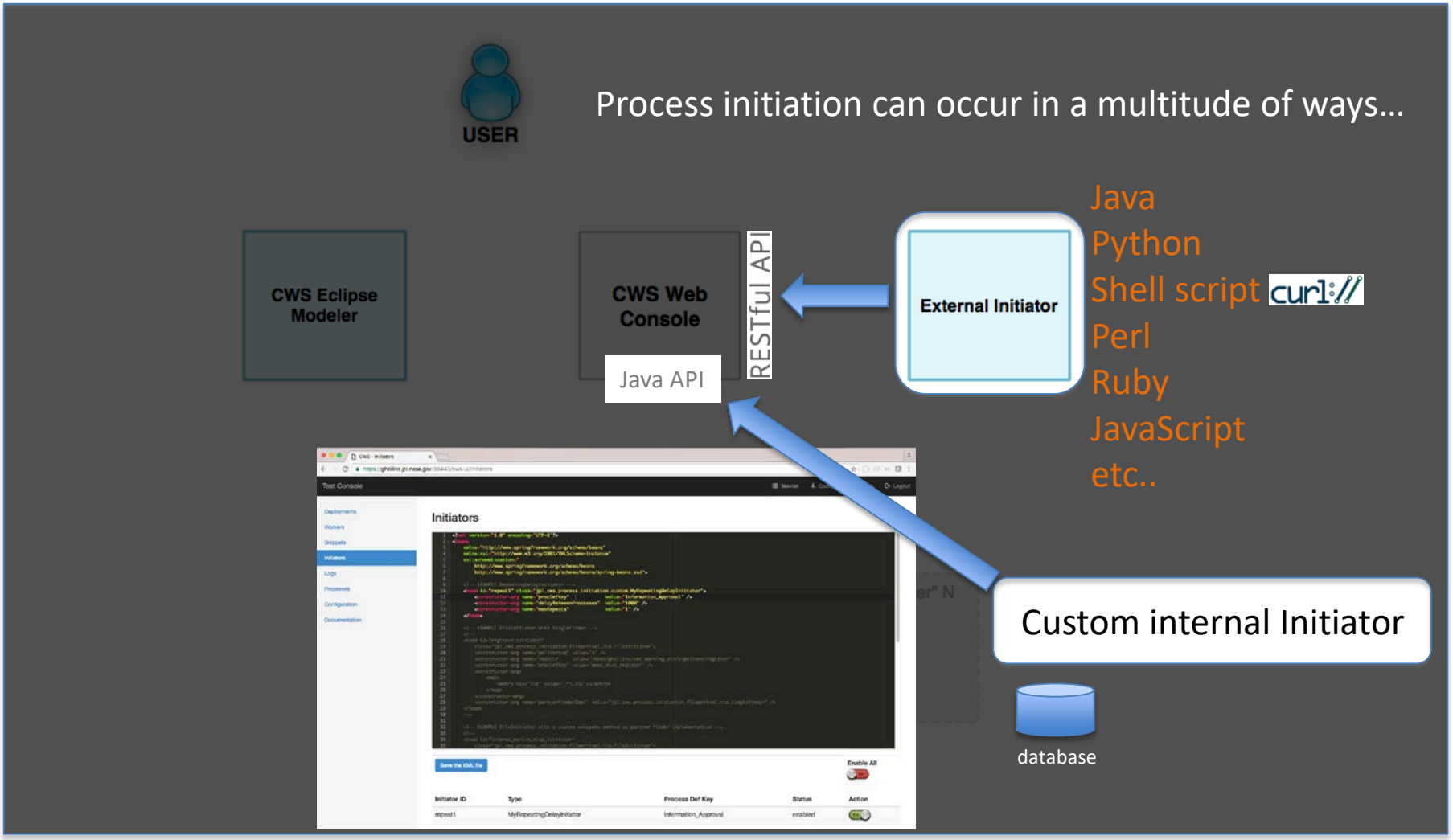
Process Initiation



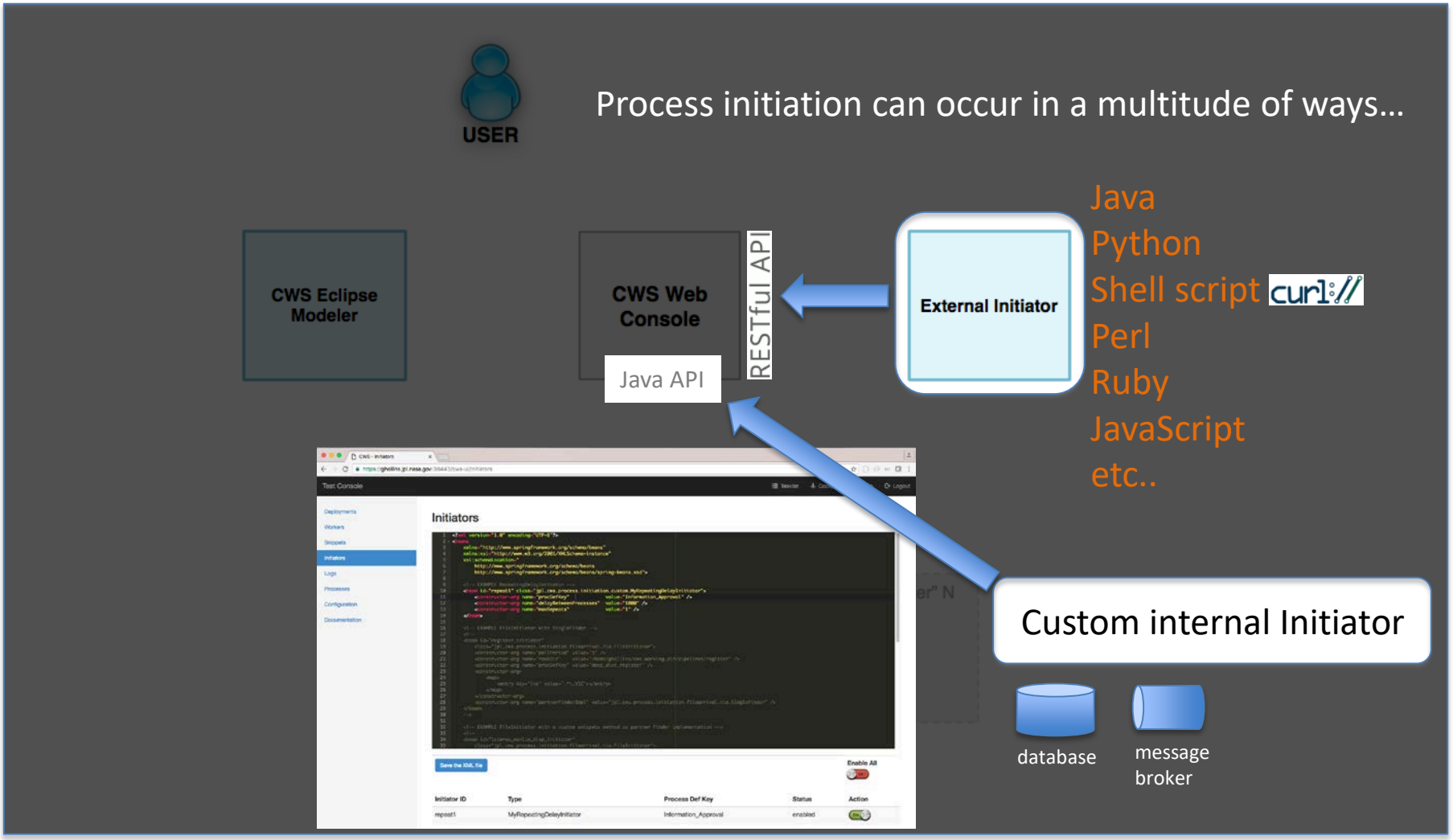
Process Initiation



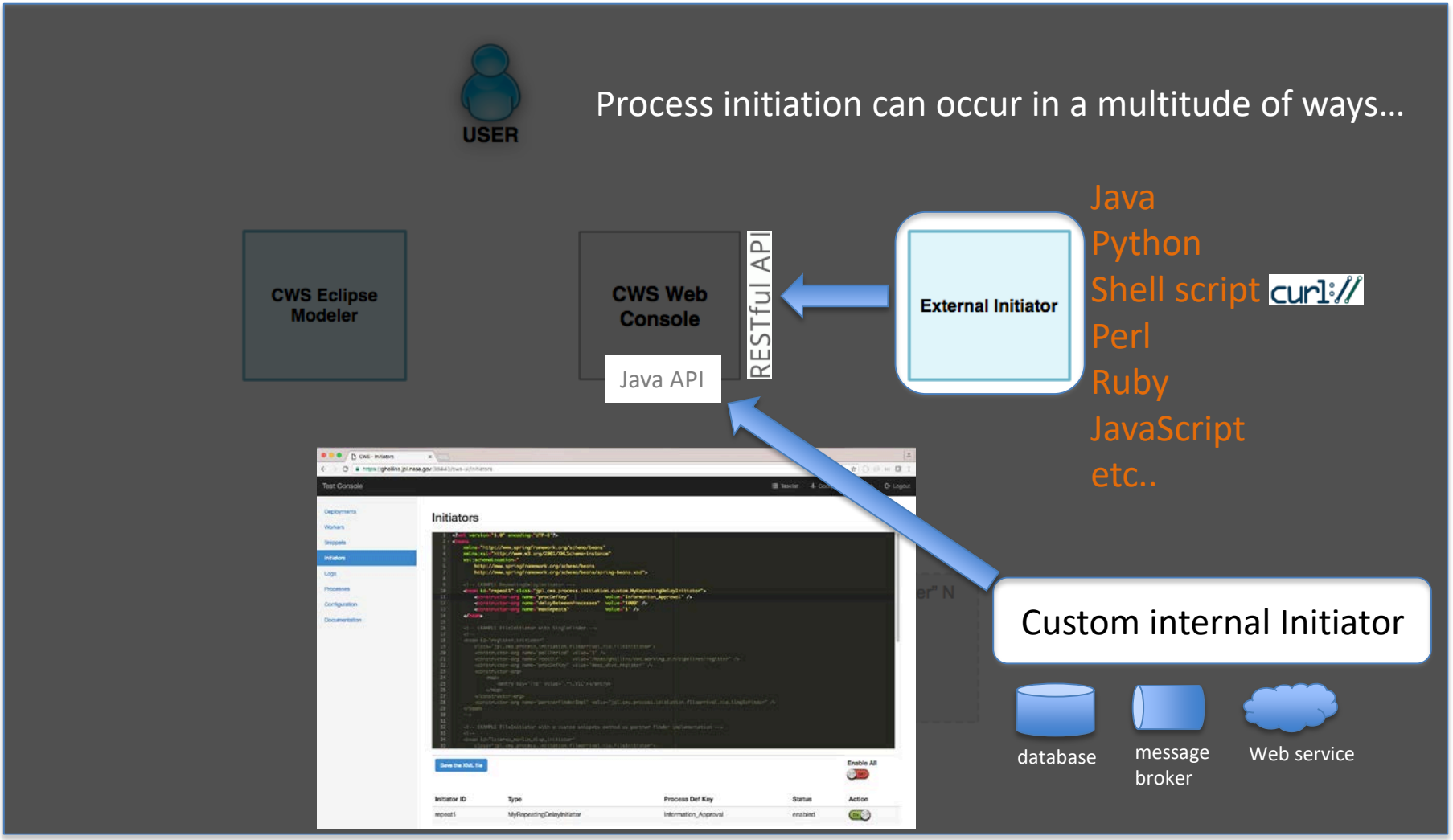
Process Initiation



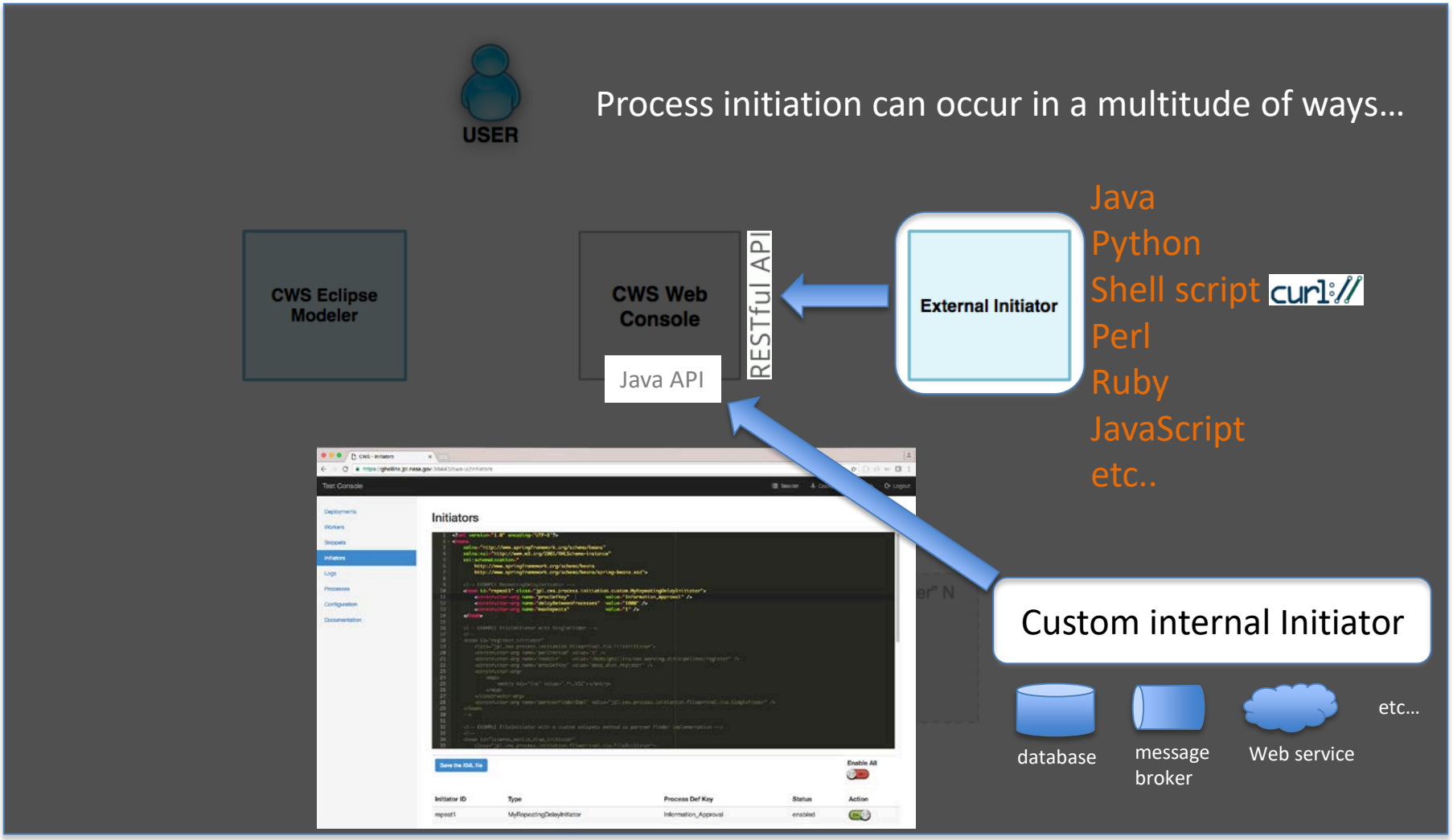
Process Initiation



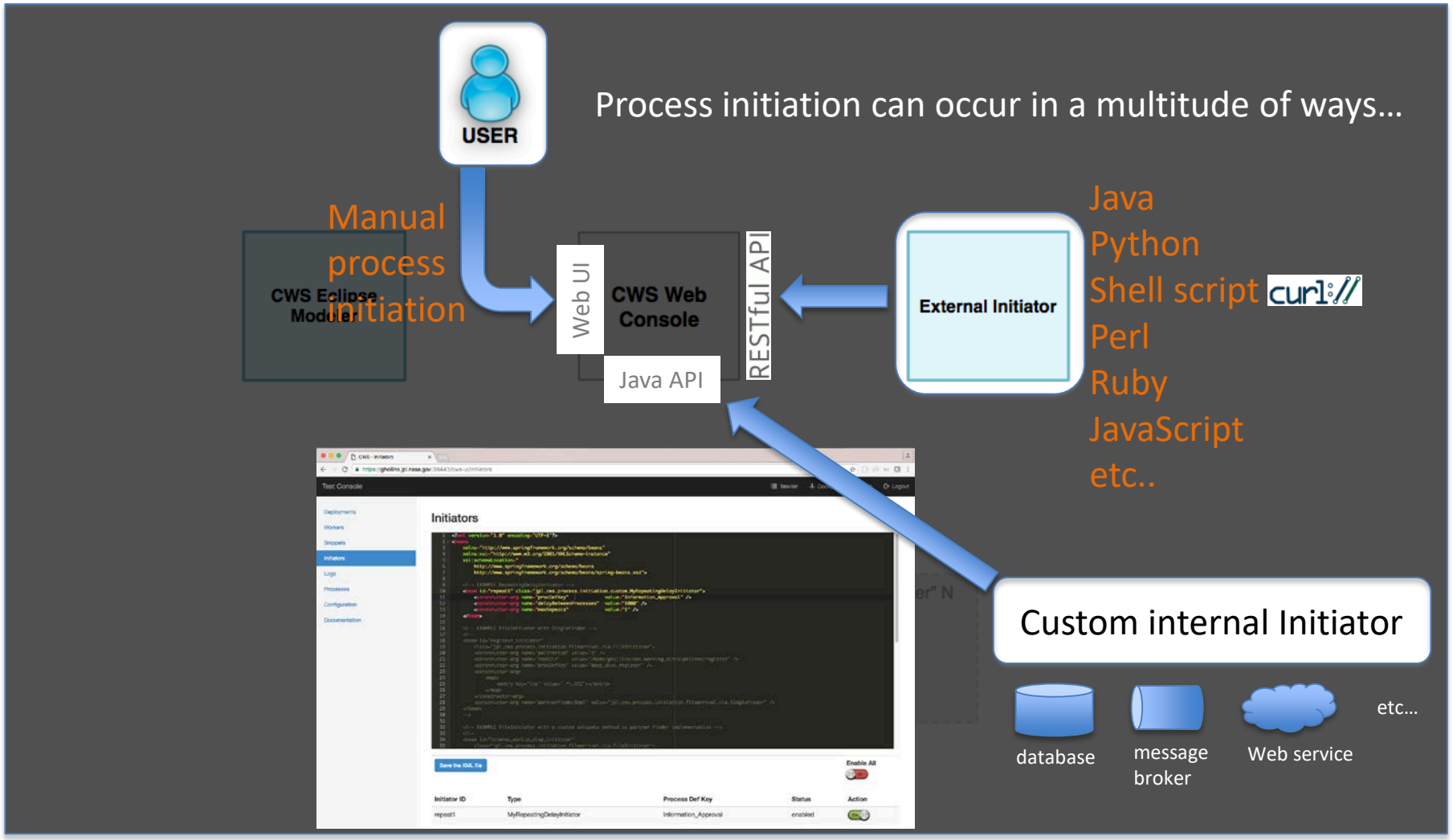
Process Initiation



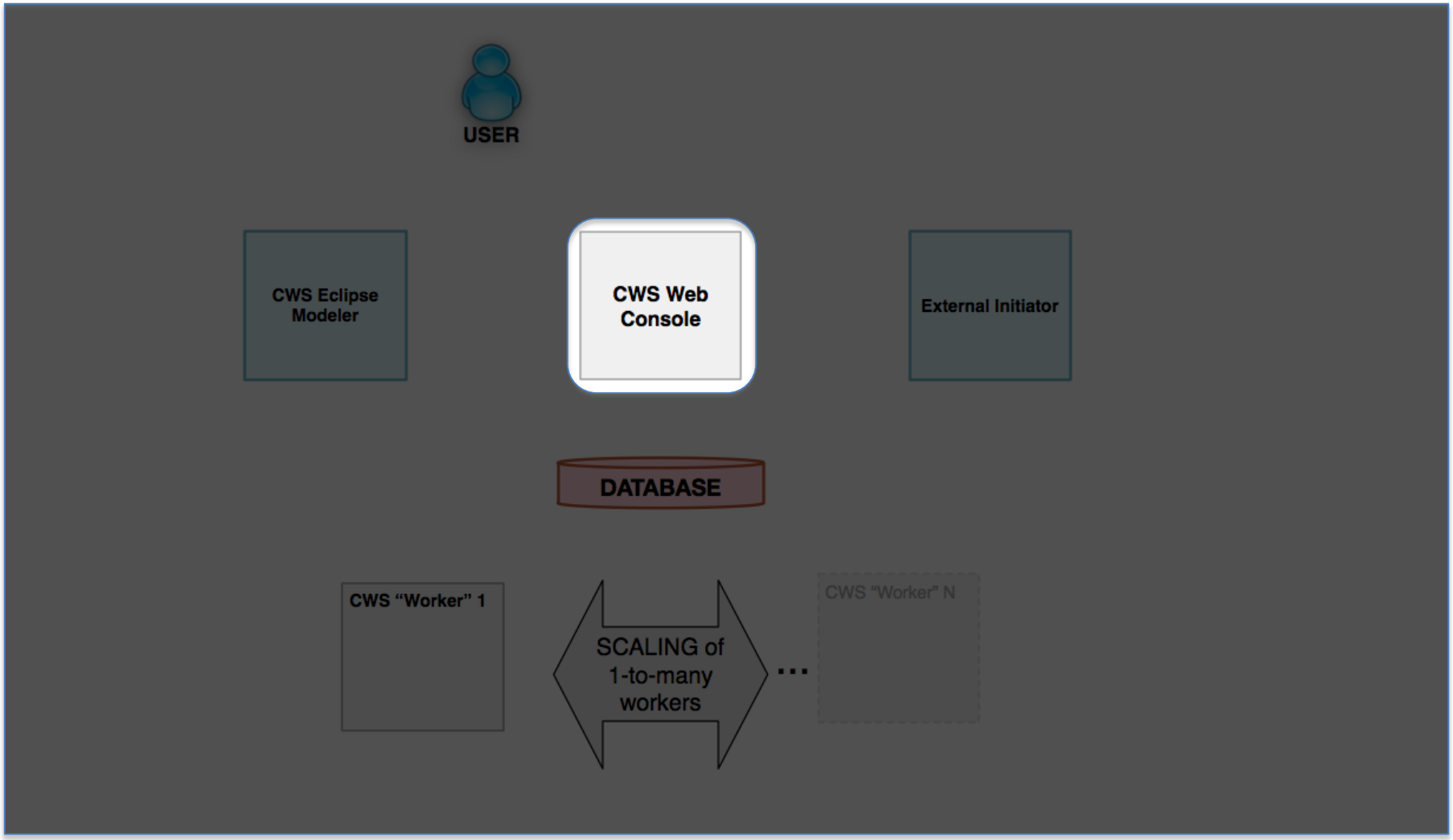
Process Initiation



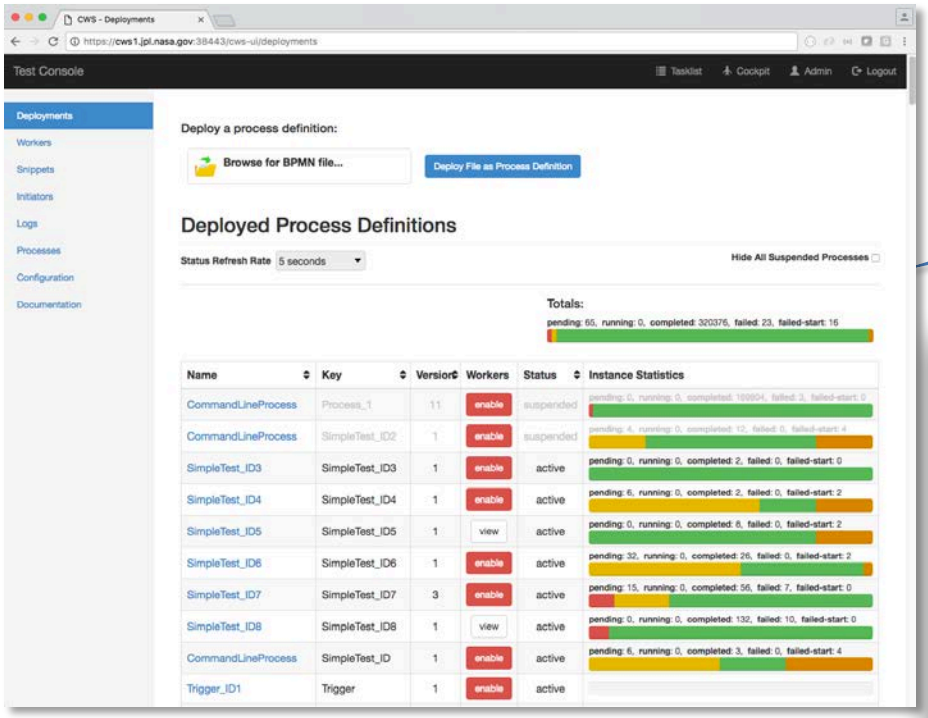
Process Initiation



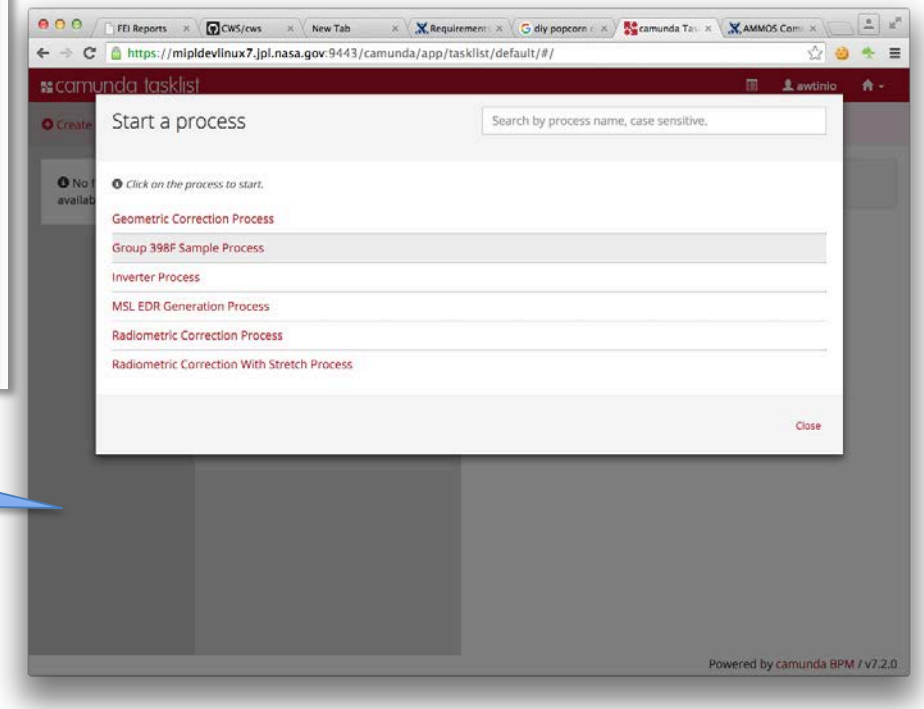
Process Monitoring and Management



Deploying and Executing Process



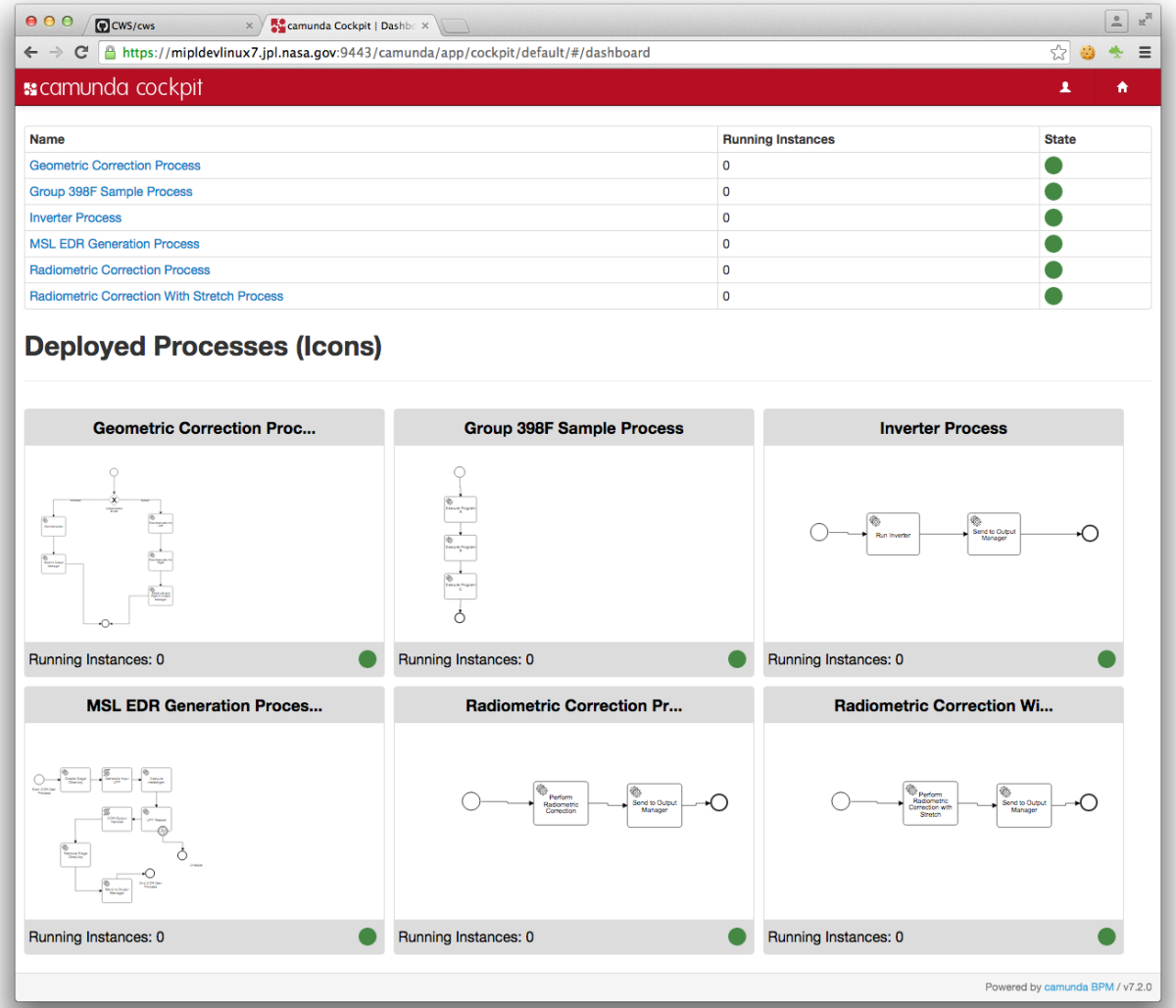
Process definitions deployed to the CWS via the web based console



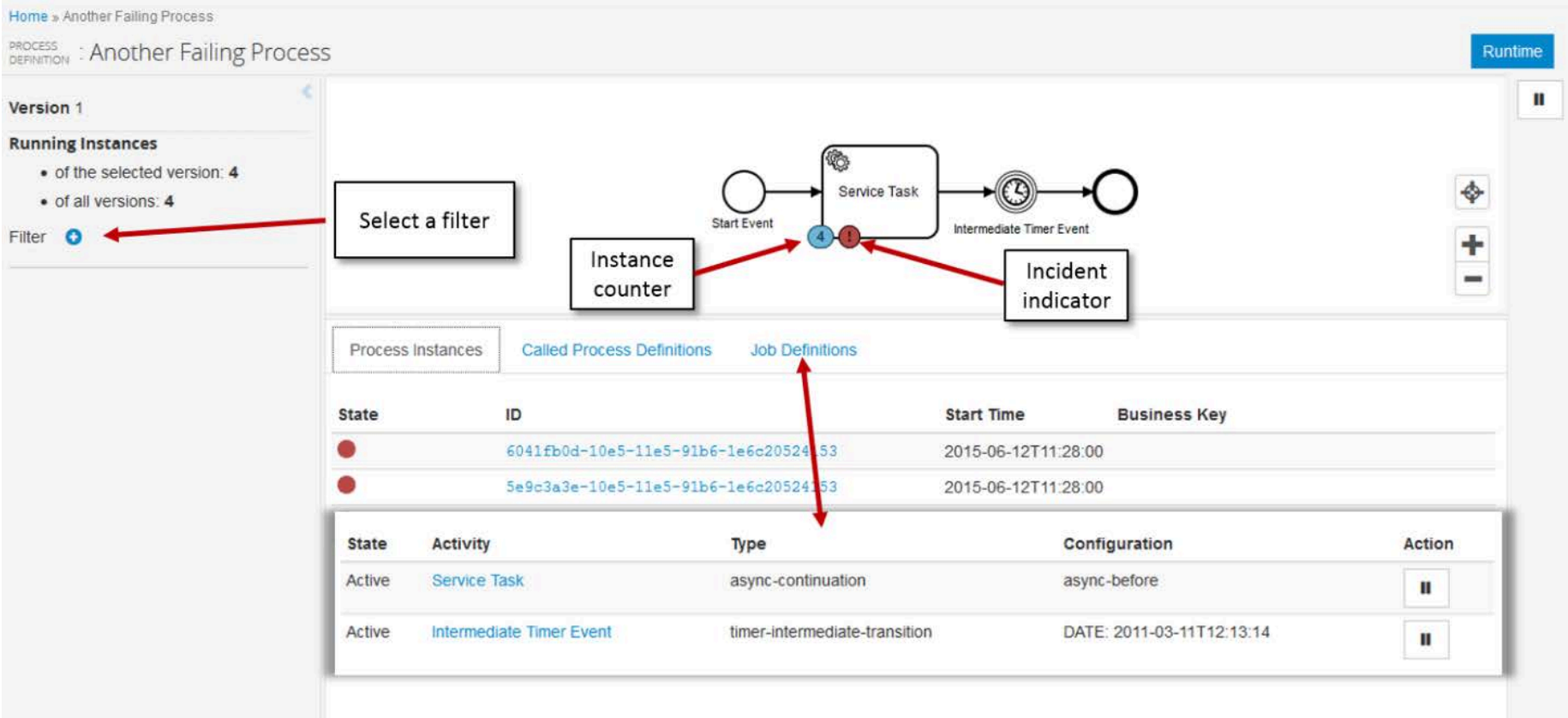
Manual execution of processes can be done through the web console as well**

** Typical process executions are triggered automatically via event listeners

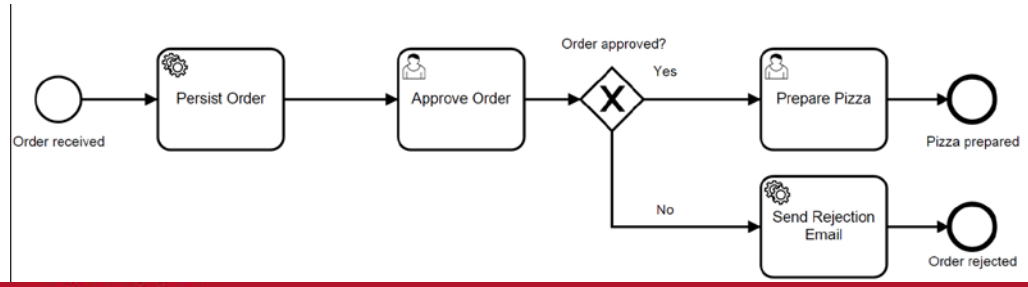
Monitoring and Management

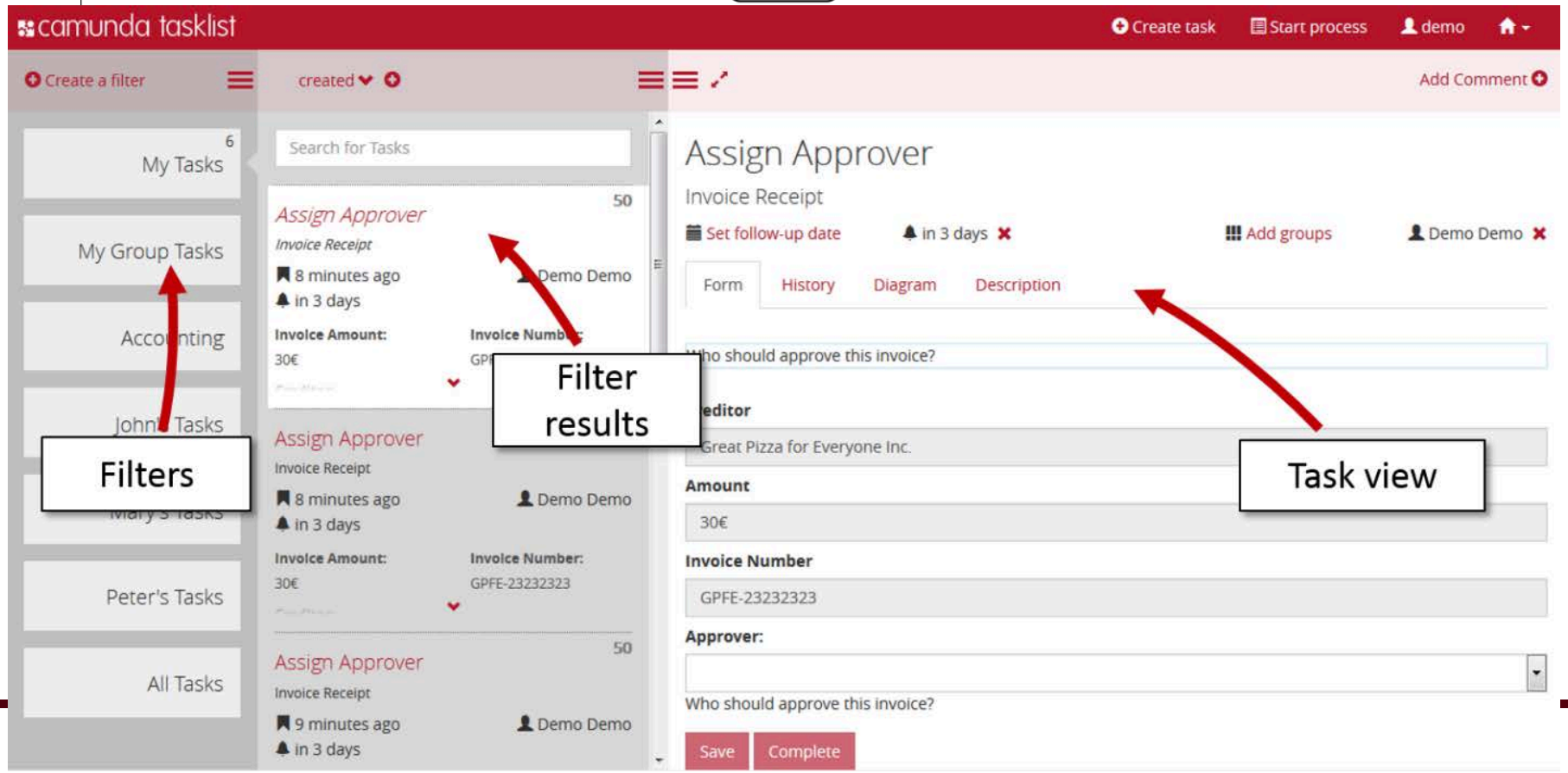


Monitoring and Management



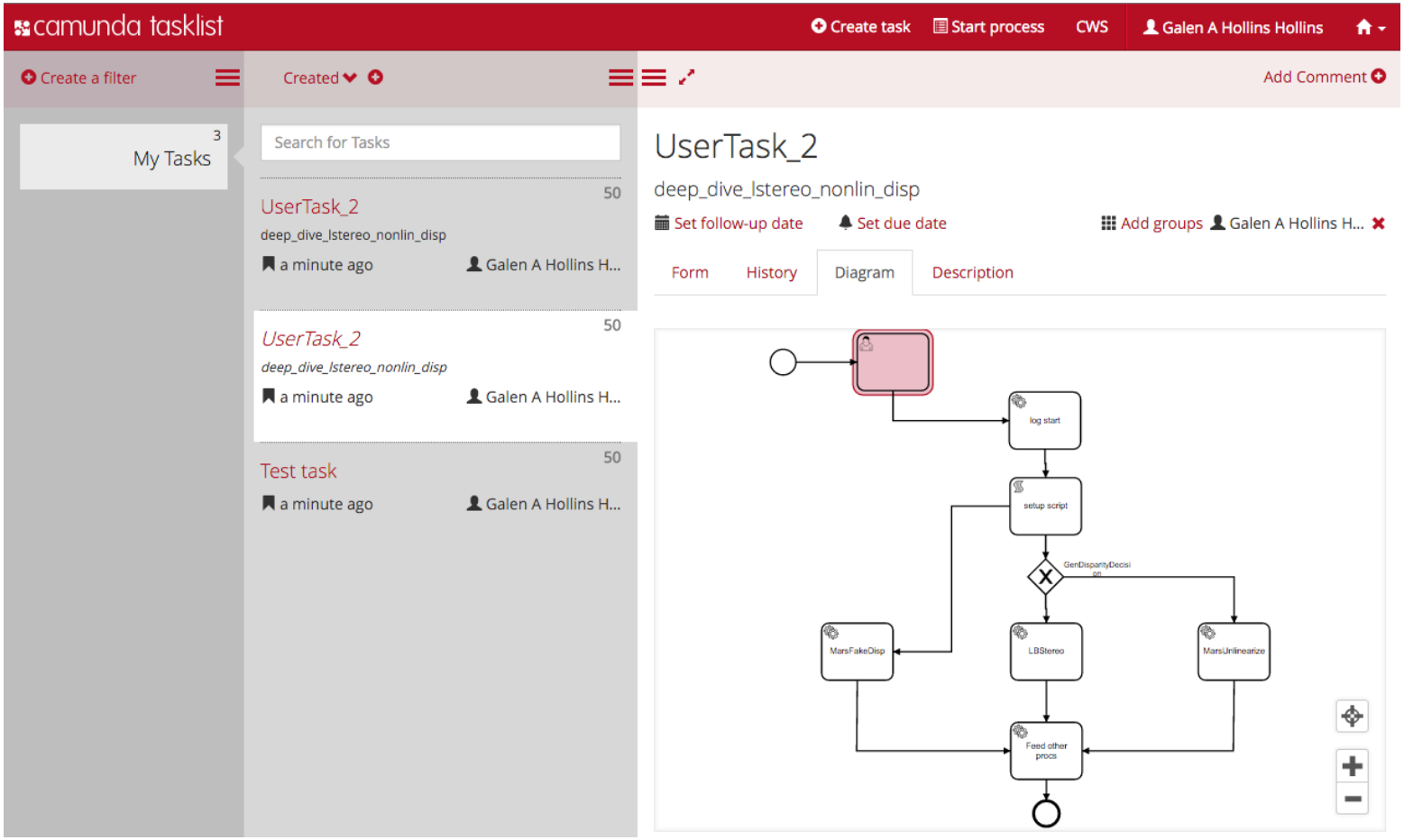
User Tasks


Order received
Persist Order
Approve Order
Order approved?
Yes
Prepare Pizza
Pizza prepared
No
Send Rejection Email
Order rejected


camunda tasklist
Create task
Start process
demo
Create a filter
created
Add Comment
My Tasks
My Group Tasks
Accounting
Johnny Tasks
Mary's Tasks
Peter's Tasks
All Tasks
Search for Tasks
Assign Approver
Invoice Receipt
8 minutes ago
in 3 days
Demo Demo
Invoice Amount:
30€
Invoice Number:
GPFE-23232323
9 minutes ago
Assign Approver
Invoice Receipt
Set follow-up date
in 3 days
Add groups
Demo Demo
Form
History
Diagram
Description
Who should approve this invoice?
Great Pizza for Everyone Inc.
Amount
30€
Invoice Number
GPFE-23232323
Approver:
Who should approve this invoice?
Save
Complete
Filters
Filter results
Task view

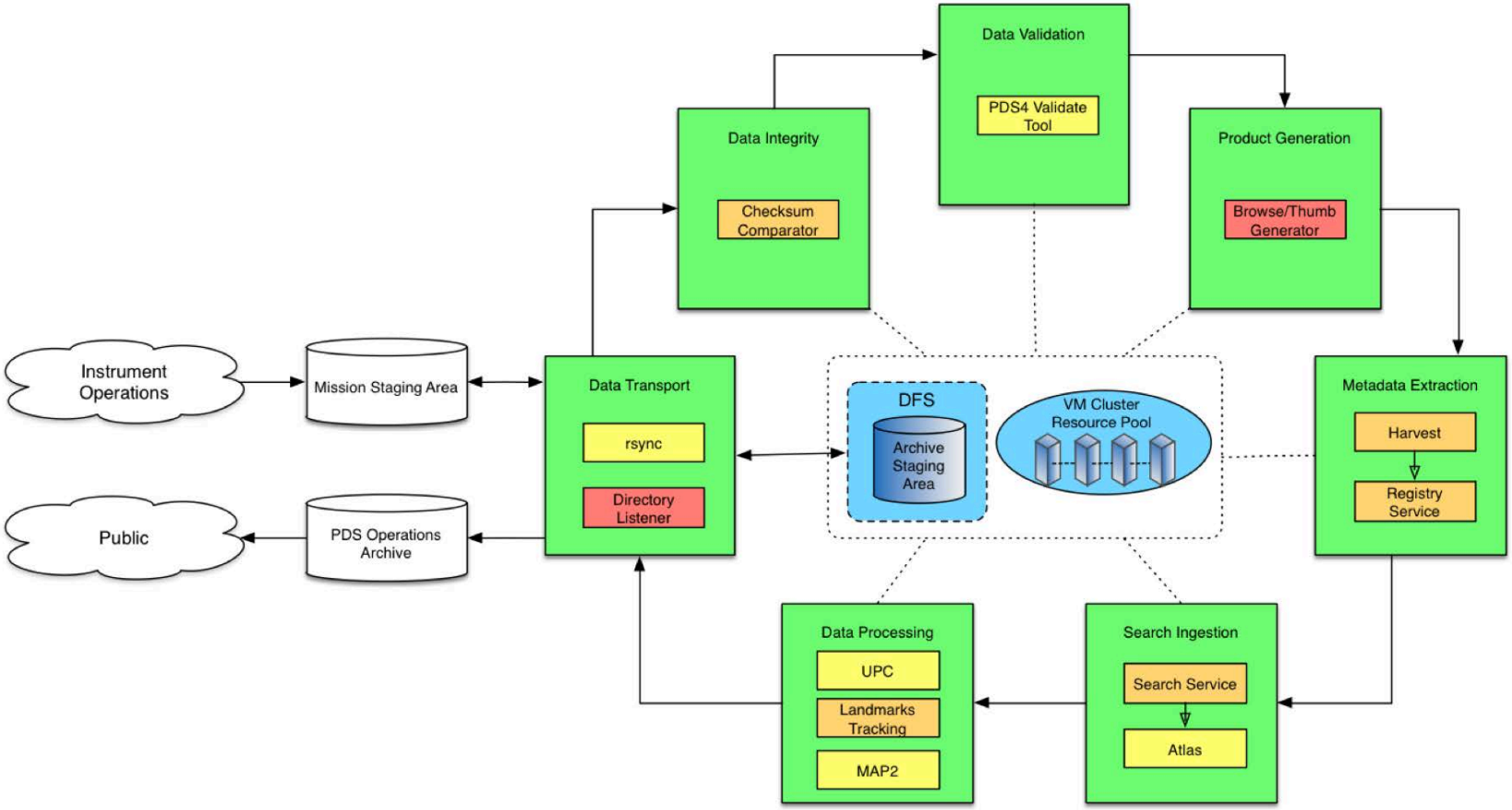
User Tasks

Contextual information about User Task…



PDS Example

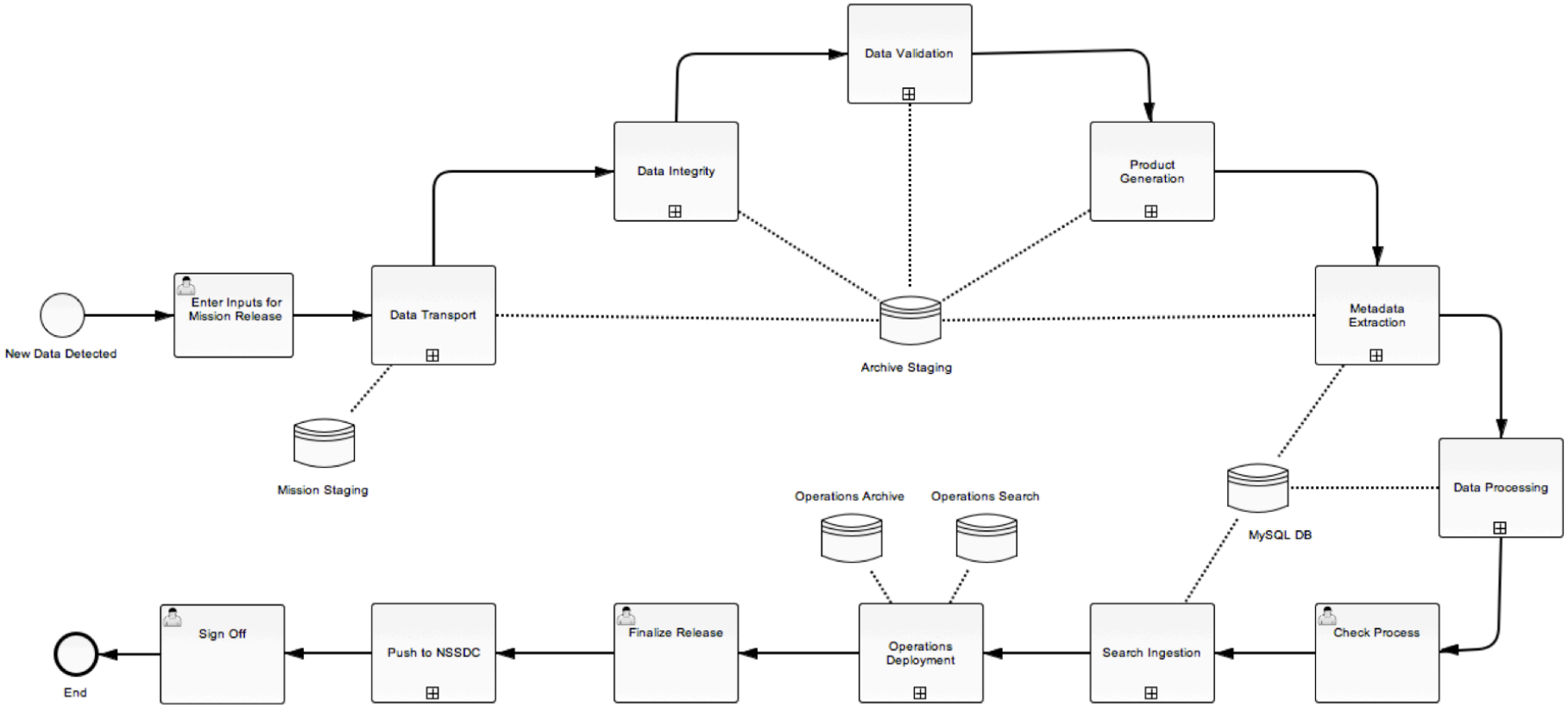
From concept…



This document has been reviewed for export control and it does NOT contain controlled technical data.

PDS Example

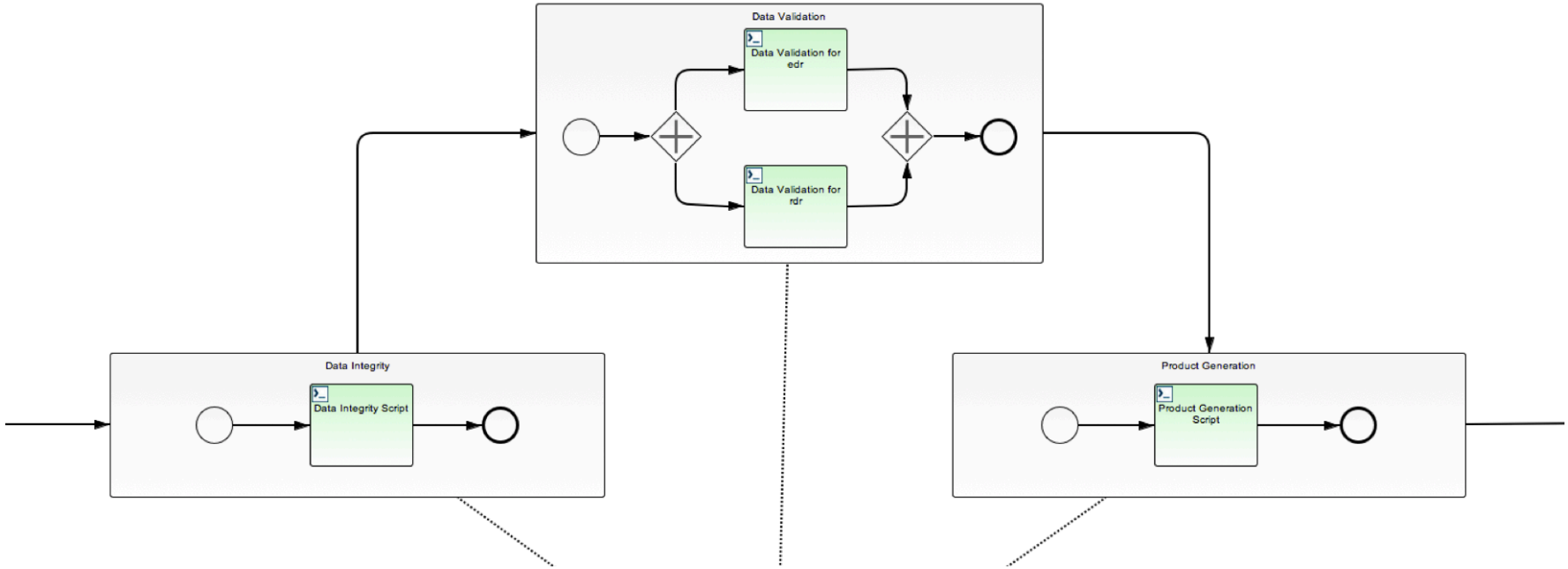
To reality:



This document has been reviewed for export control and it does NOT contain controlled technical data.

PDS Example

(zoomed in):



This document has been reviewed for export control and it does NOT contain controlled technical data.

Process execution and more…

CWS is more than just the execution of a workflow

Covers the entire process life cycle:

- Definition of a process model
- Deployment
- Initiation
- **Execution**
- Monitoring and Control
- Management
- Manual (User) Tasks
- Updates to and Re-deployment of process model

CWS Features Summary

- *Process Execution Engine*
  - BPMN 2.0 execution engine
  - Orchestration of **automated** tasks AND **user** tasks
- *Drag and Drop Process Modeler*
  - *Library of Common Tasks*
- *Human/User Task Management*
  - Handles life cycle of human tasks (assigning, claiming, completing, etc…)
- *Distributed and Load Balanced*
  - Execution of processes can be distributed across computing nodes/workers
- *Event-based Process Initiation*
  - *Plug-in architecture where custom event listeners can be implemented and plugged-in.*
- *Web-based Control and Management Interface*
- *Security Features*
  - Pluggable authentication model
  - Rule based authorization
  - Secured connections

Software Design Approach

1. *Adhere to current BPM industry standards*
   - BPMN 2.0

2. *Leverage existing BPMS solutions*
   - Investigated open-source solutions.
   - Selected Camunda BPM (http://camunda.org)

3. *The design must be flexible to support a wide variety of use cases*
   - Scalable architecture to allow for use cases with different performance requirements

CWS Technology Stack

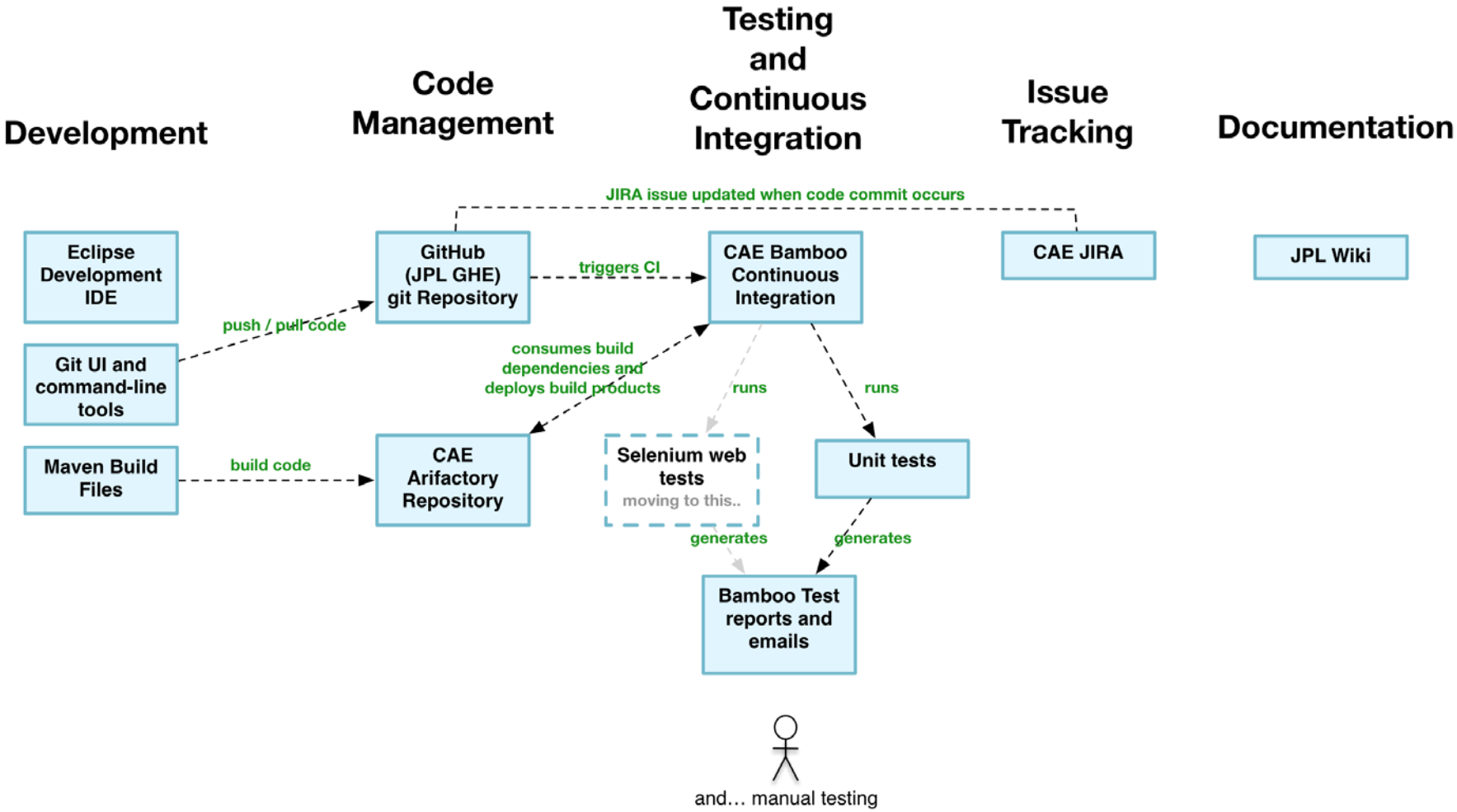
Front End

- **Bootstrap**, **jQuery**
- **Freemarker** (templating engine)
- **Angular.js** (Camunda apps)
- **Javascript** / **CSS** / **HTML**

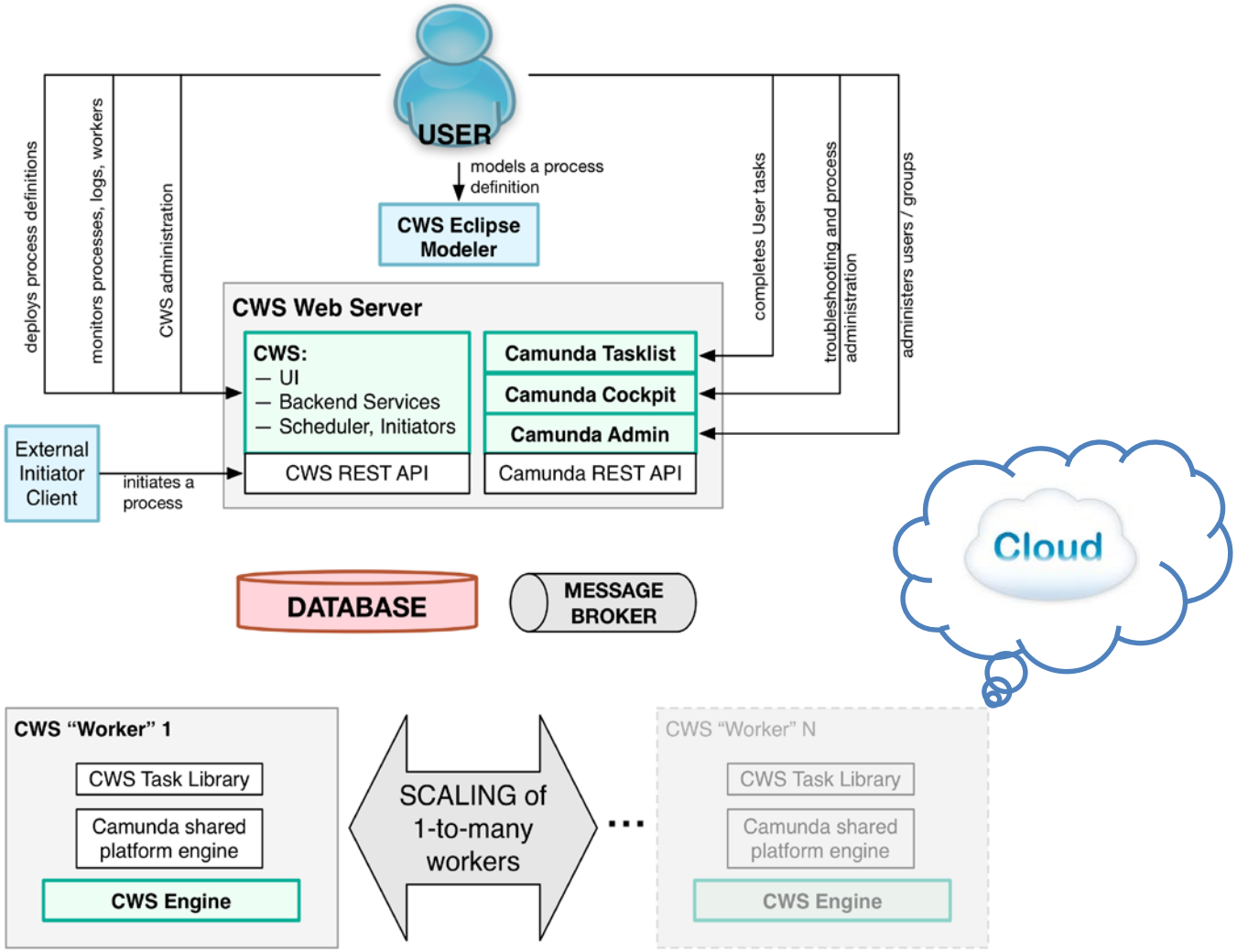
Back End

- Web Container: Apache **Tomcat**
- Core Service Code:
  - **Java** (portable to any platform)
  - **Spring Framework** (contexts, MVC, JMS, JDBC, etc..)
  - **Other various open source Java libraries**
- BPMN Engine and apps: **Camunda BPM**
- Database: **MySQL**
- Messaging: Apache **ActiveMQ**
- Log Gathering / Searching: **LogStash / ElasticSearch**
- Other: various **shell scripts**

Software Development Process



Cloud Demo


USER
models a process definition
CWS Eclipse Modeler
deploys process definitions
monitors processes, logs, workers
CWS administration
completes User tasks
troubleshooting and process administration
administers users / groups
CWS Web Server
CWS:
— UI
— Backend Services
— Scheduler, Initiators
CWS REST API
Camunda Tasklist
Camunda Cockpit
Camunda Admin
Camunda REST API
External Initiator Client
initiates a process
DATABASE
MESSAGE BROKER
Cloud
CWS "Worker" 1
CWS Task Library
Camunda shared platform engine
CWS Engine
SCALING of 1-to-many workers
...
CWS "Worker" N
CWS Task Library
Camunda shared platform engine
CWS Engine

Demo

Questions/Comments

Contact Info

Galen Hollins (CWS Task Lead / Developer)

Galen.A.Hollins@jpl.nasa.gov

Jim Wood (CWS Developer)

James.F.Wood@jpl.nasa.gov

Adrian Tinio (MGSS IDS System Engineer)

Adrian.W.Tinio@jpl.nasa.gov

CWS Wiki

http://goto.jpl.nasa.gov/cws