



Common Workflow Service

Standards Based Solution for Managing Operational Processes

Galen Hollins

Jet Propulsion Laboratory,
California Institute of Technology



Overview



What is the Common Workflow Service (CWS)?

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

Developed under the Multimission Ground System and Services NASA program. MGSS is the NASA Program responsible for the management, implementation, maintenance and operation of the Advanced Multi-Mission Operations System ([AMMOS](#)).



Need for Workflow Solutions at JPL



- Automation and monitoring of operational processes
 - Scientific data pipelines
 - Other automation
- Standardized way to do things
 - Shared, collective benefit of productivity
- Understandability
- Maintainability across developers and projects



Points to Consider



- Distributed processing, performance constraints
- Both short and long-running tasks
- Wide variety of workflow types
- Agile process development/modification



Software Design Approach



1. *Adhere to current BPM industry standards*

- BPMN 2.0

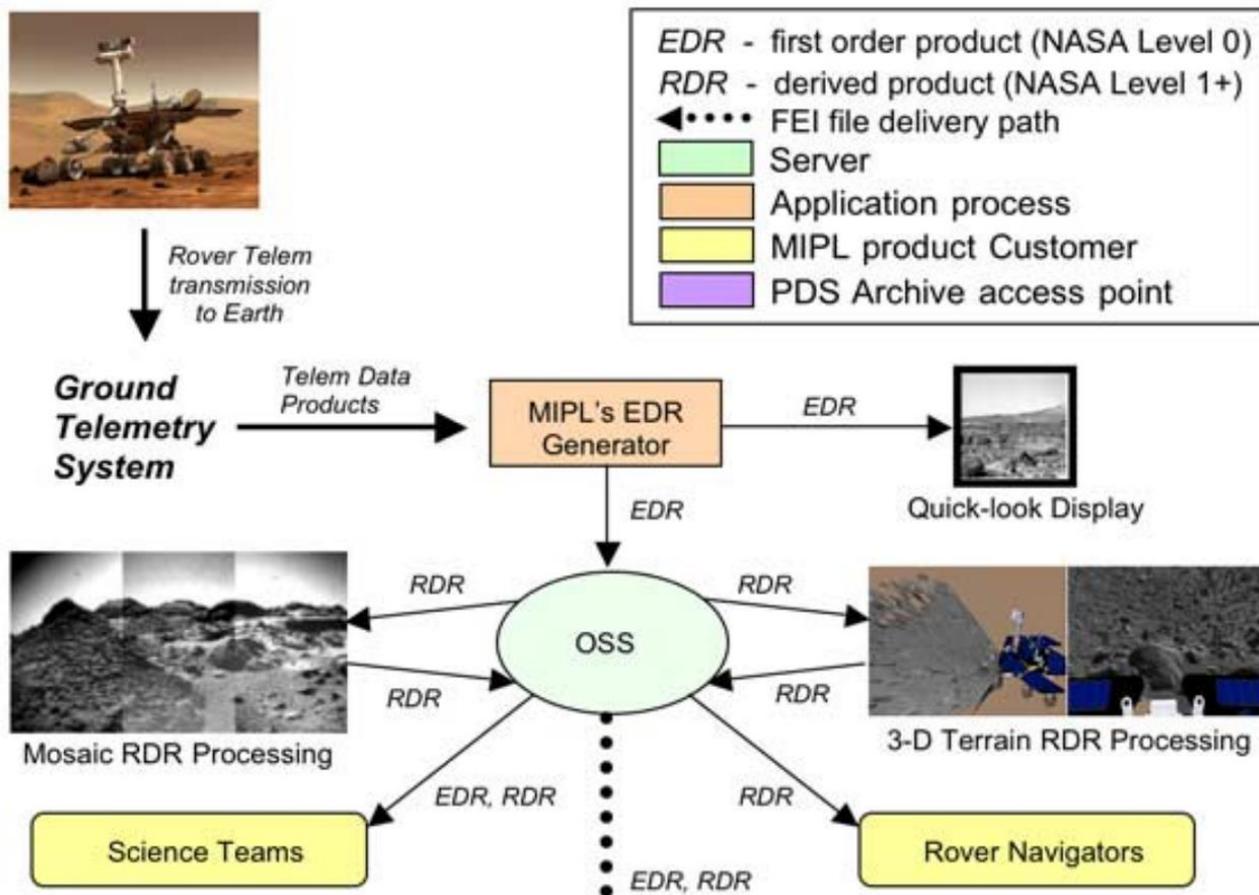
2. *Leverage existing BPMS solutions*

- 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

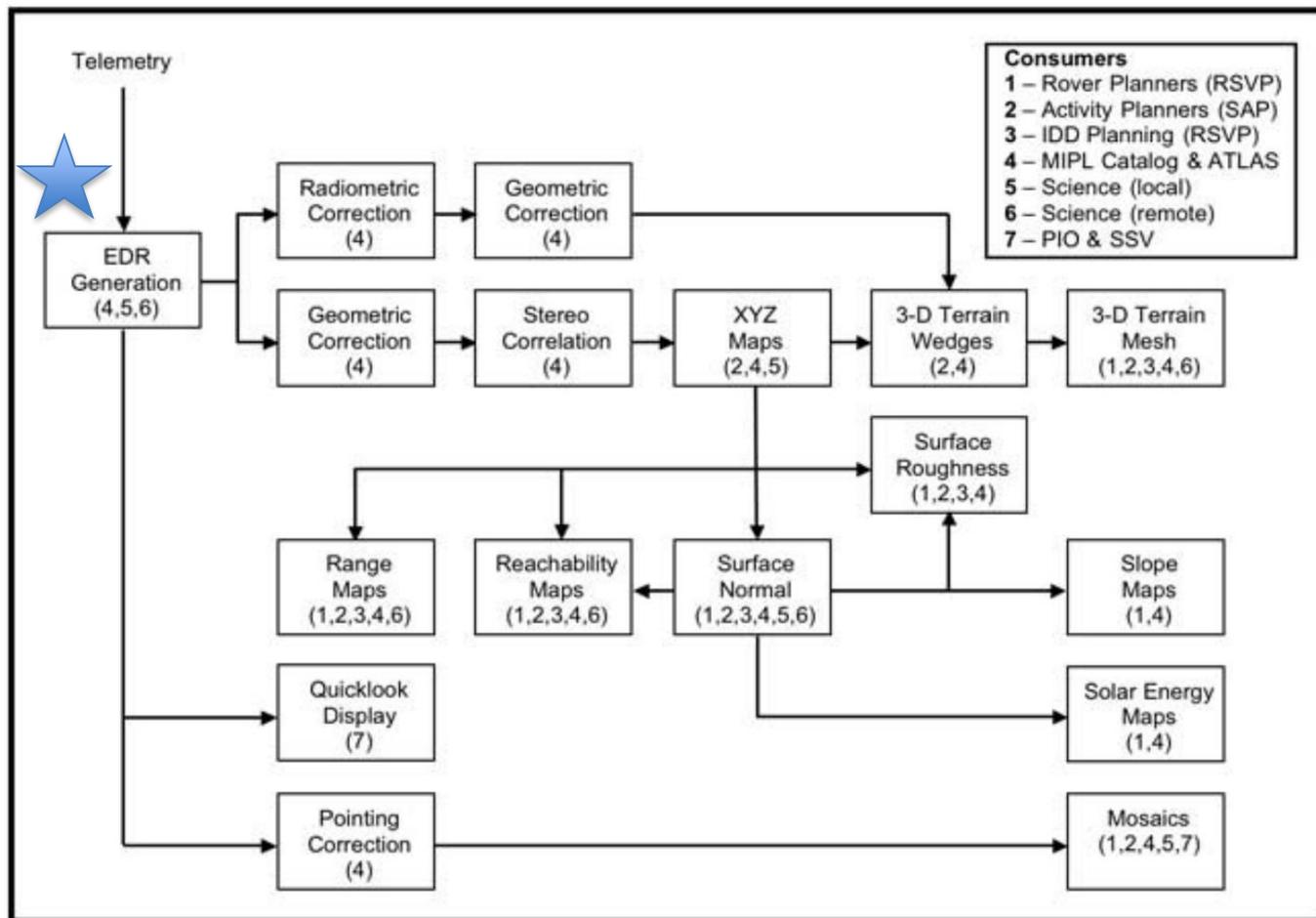
Example Mission Data Flow



Source: JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 111, E02S02, doi:10.1029/2005JE002462, 2006



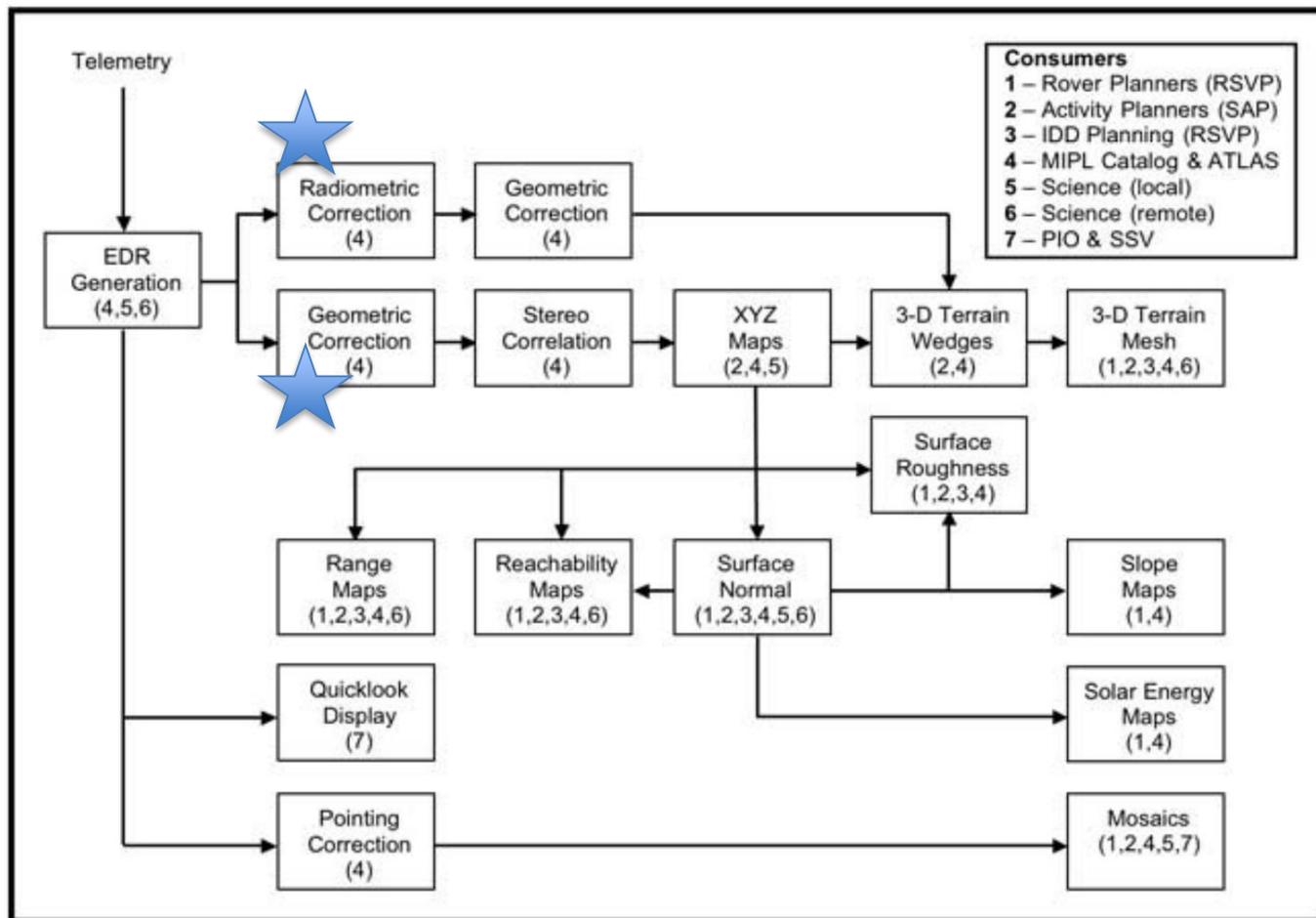
Typical Product Generation Pipeline



Source: JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 111, E02S02, doi:10.1029/2005JE002462, 2006



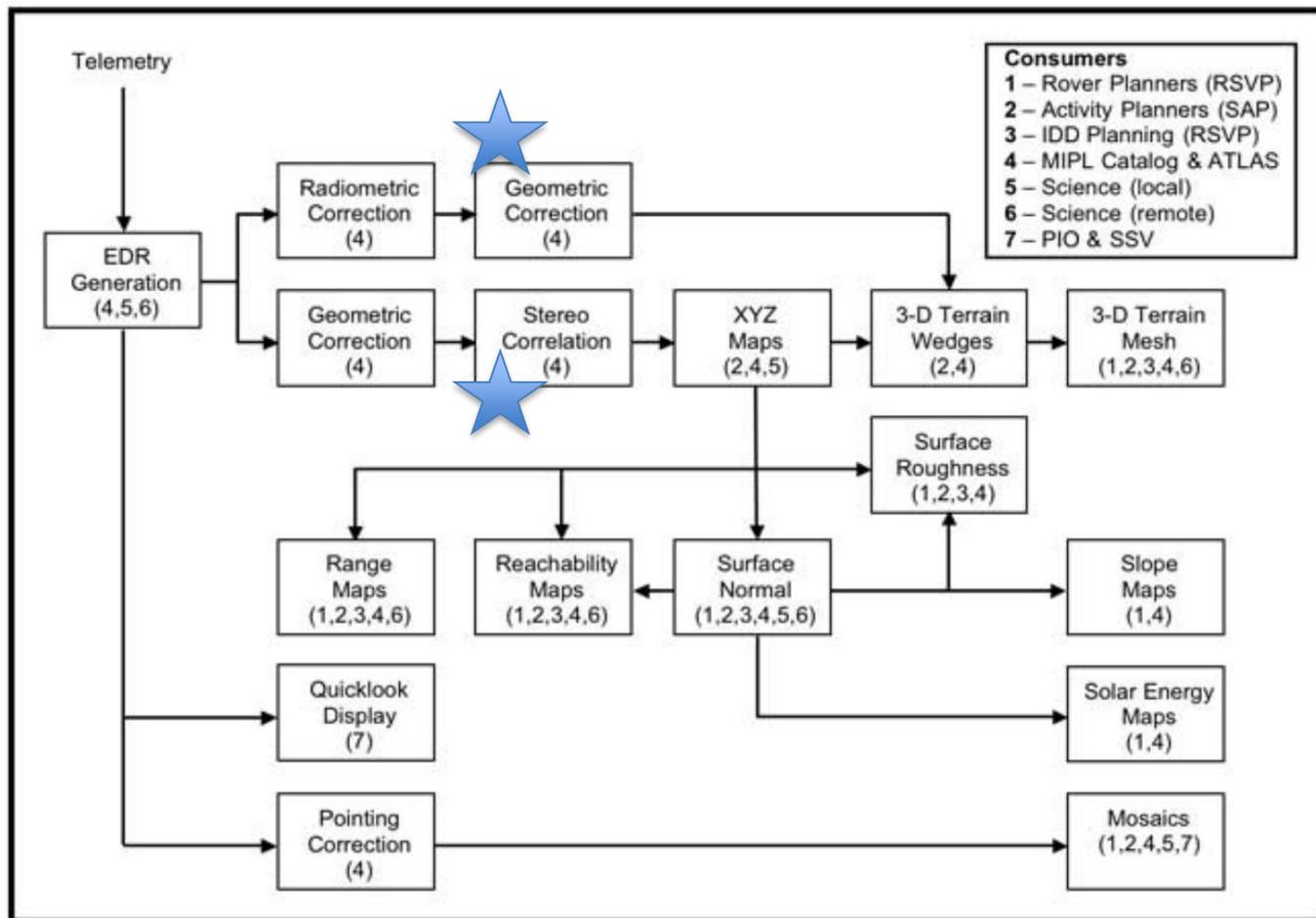
Typical Product Generation Pipeline



Source: JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 111, E02S02, doi:10.1029/2005JE002462, 2006



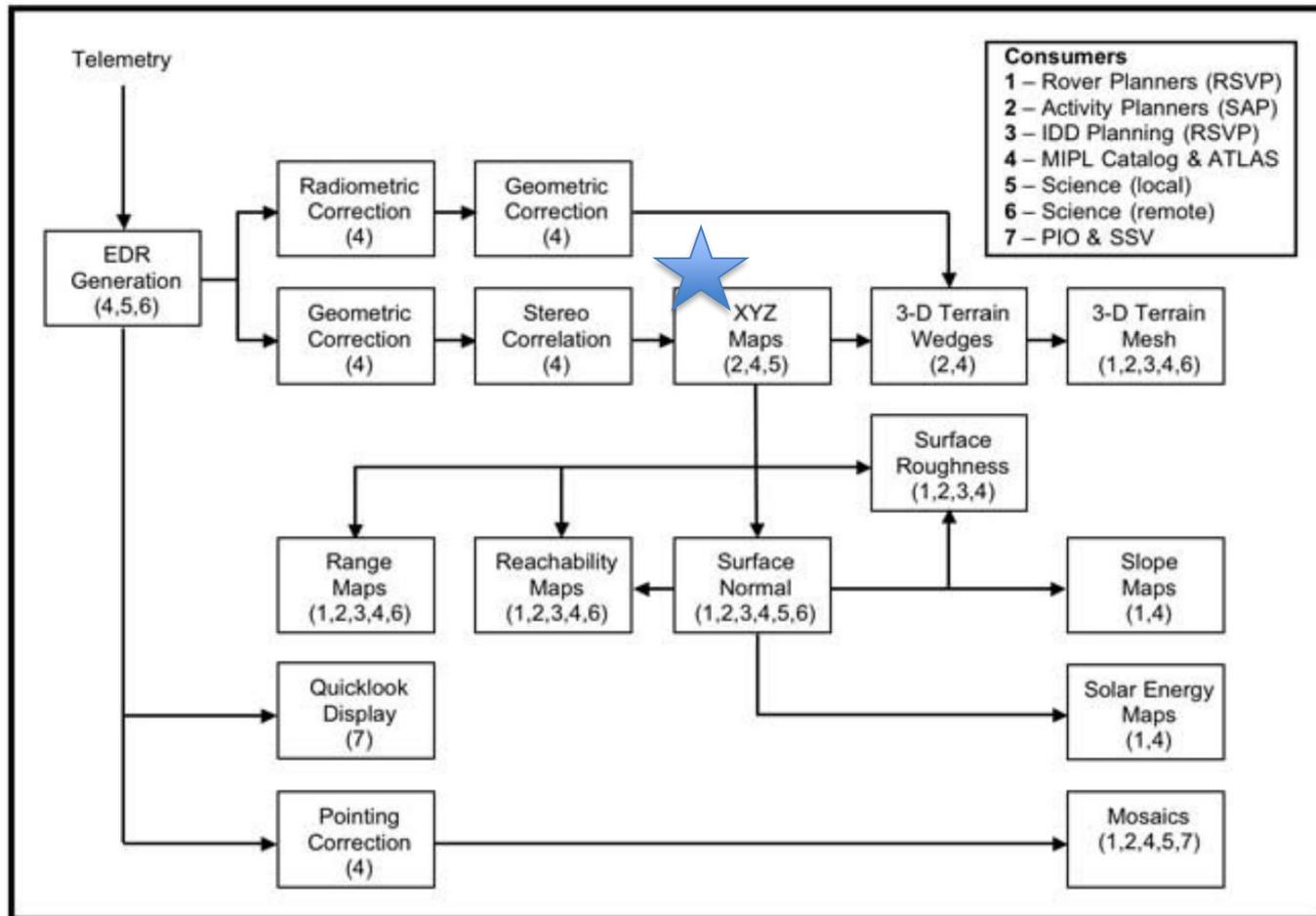
Typical Product Generation Pipeline



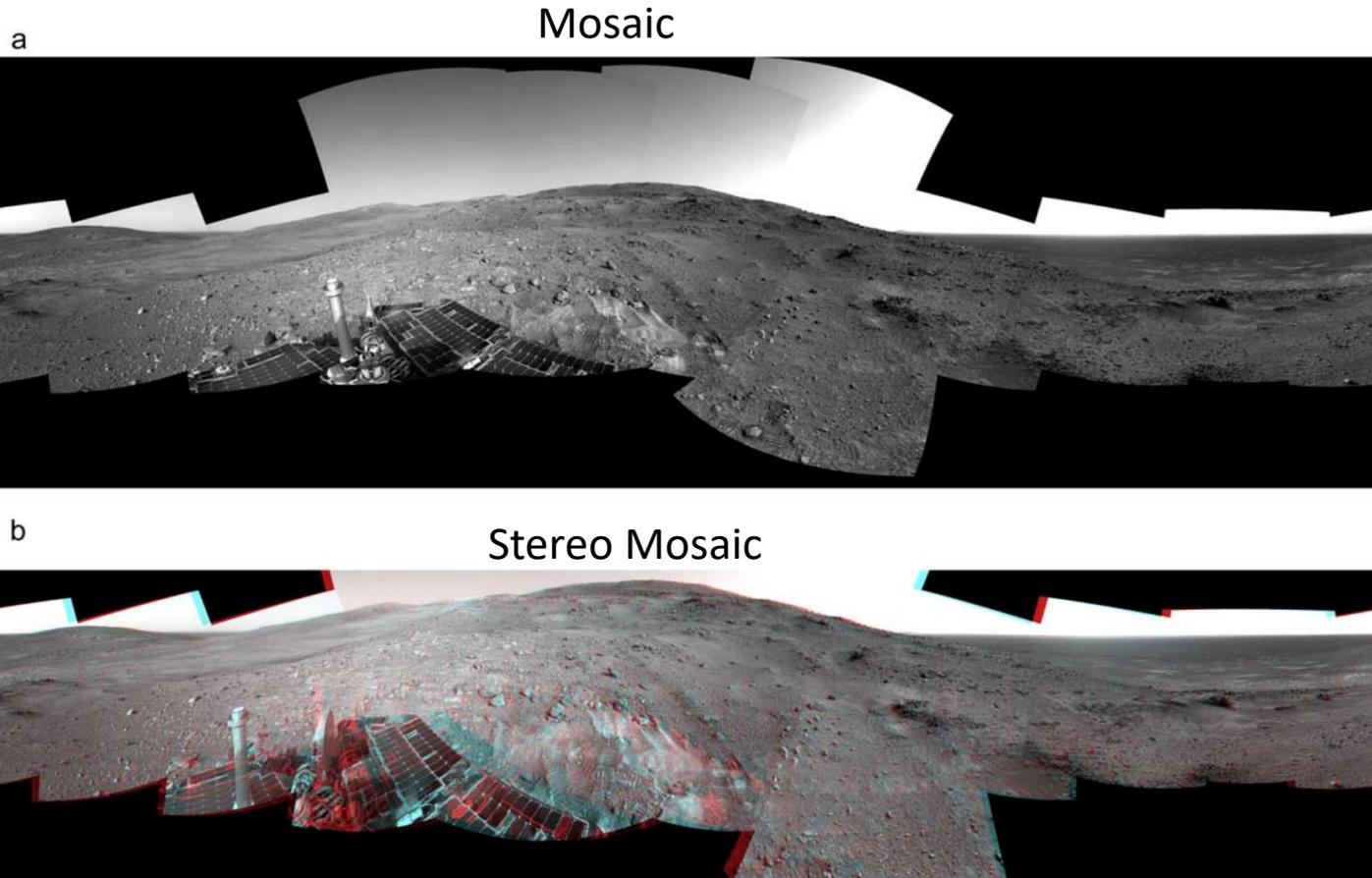
Source: JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 111, E02S02, doi:10.1029/2005JE002462, 2006



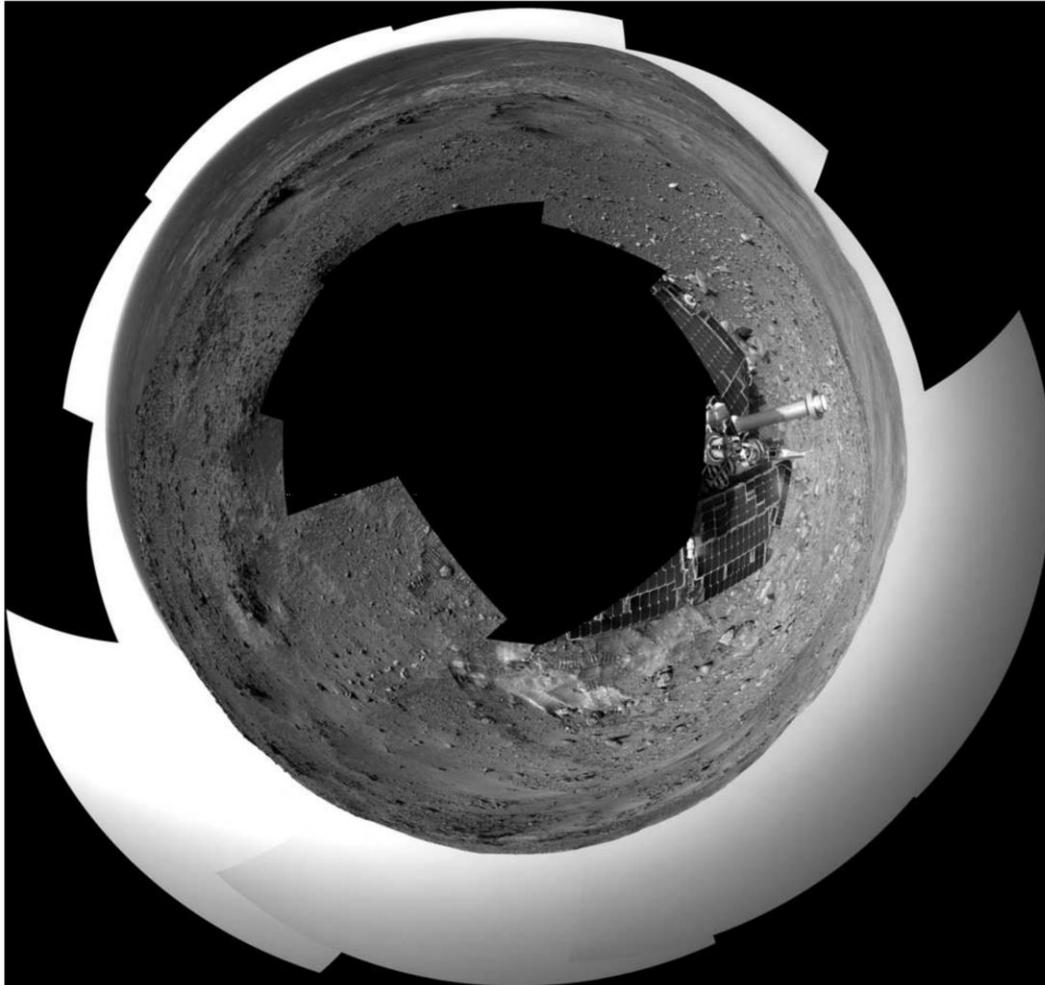
Typical Product Generation Pipeline



Source: JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 111, E02S02, doi:10.1029/2005JE002462, 2006



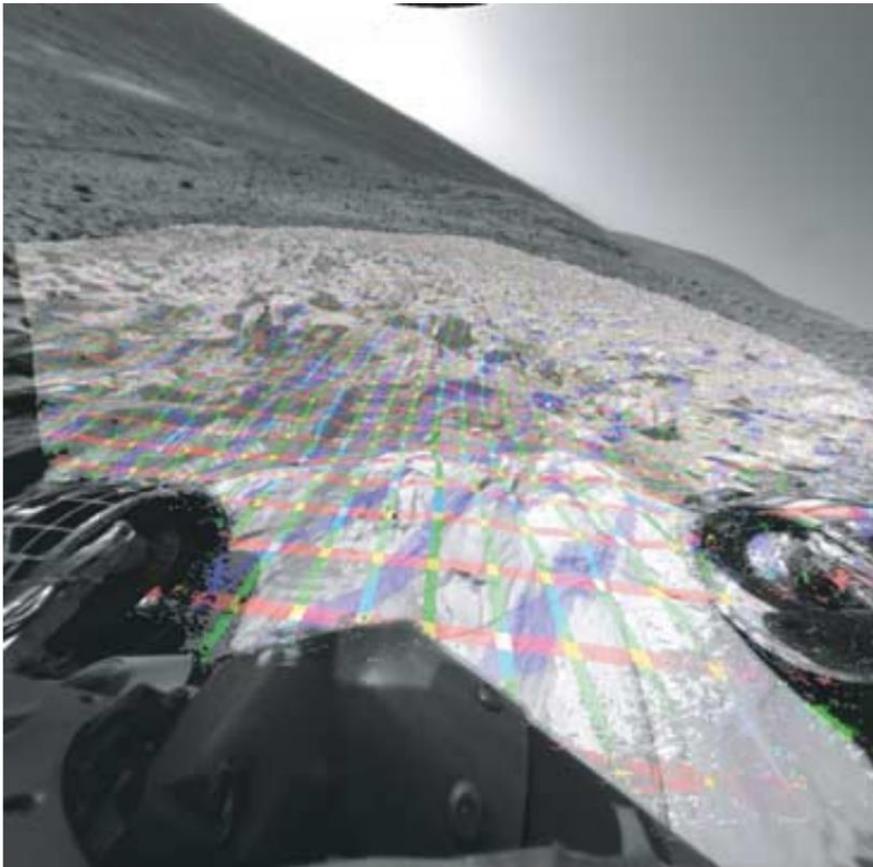
Source: JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 111, E02S02, doi:10.1029/2005JE002462, 2006



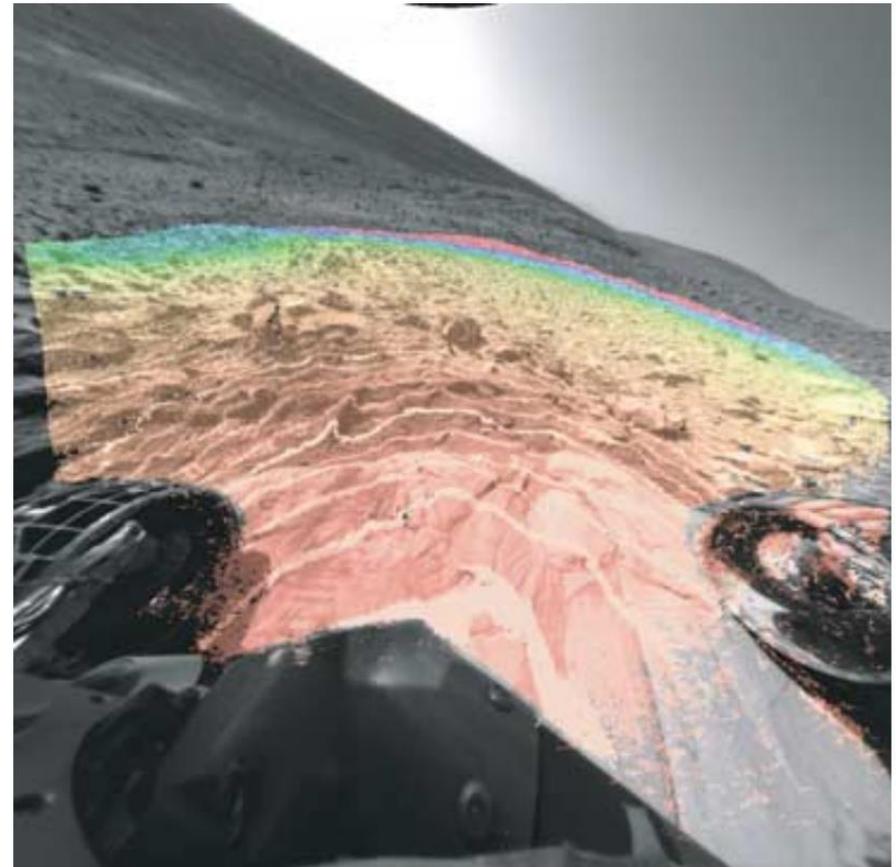
Polar Projection

Source: JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 111, E02S02, doi:10.1029/2005JE002462, 2006

XYZ Image



Range Image



Source: JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 111, E02S02, doi:10.1029/2005JE002462, 2006



What is the Common Workflow Service (CWS)?

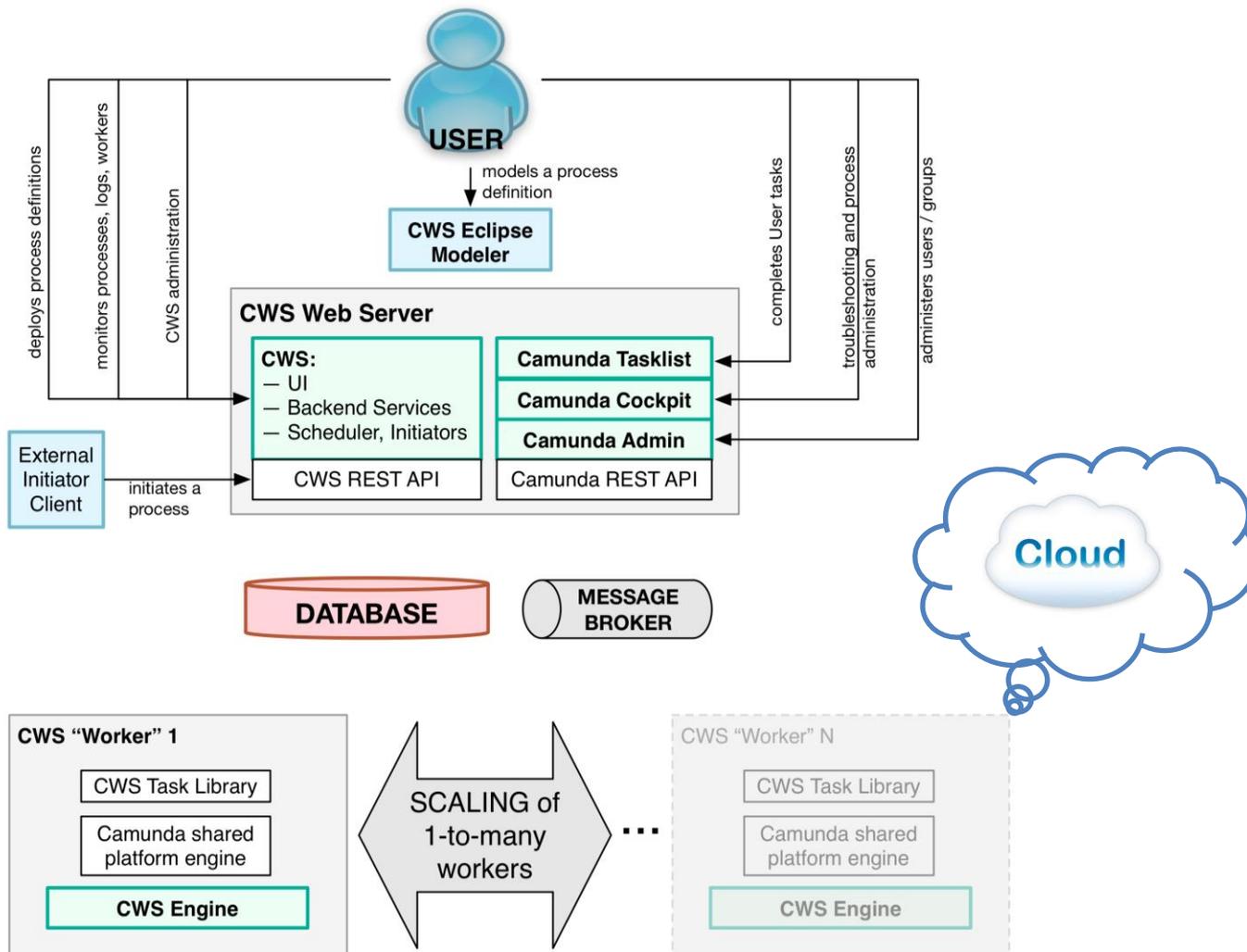
A **standards** based Business Process Management (BPM) solution for executing and managing mission operations processes.

$$\begin{aligned} \text{CWS} = & \\ & \text{BPMN Engine (Camunda BPM)} \\ & + \\ & \text{Customized BPMN 2.0 Modeler} \\ & + \\ & \text{Monitoring and Management Web Applications (CWS + Camunda)} \end{aligned}$$

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

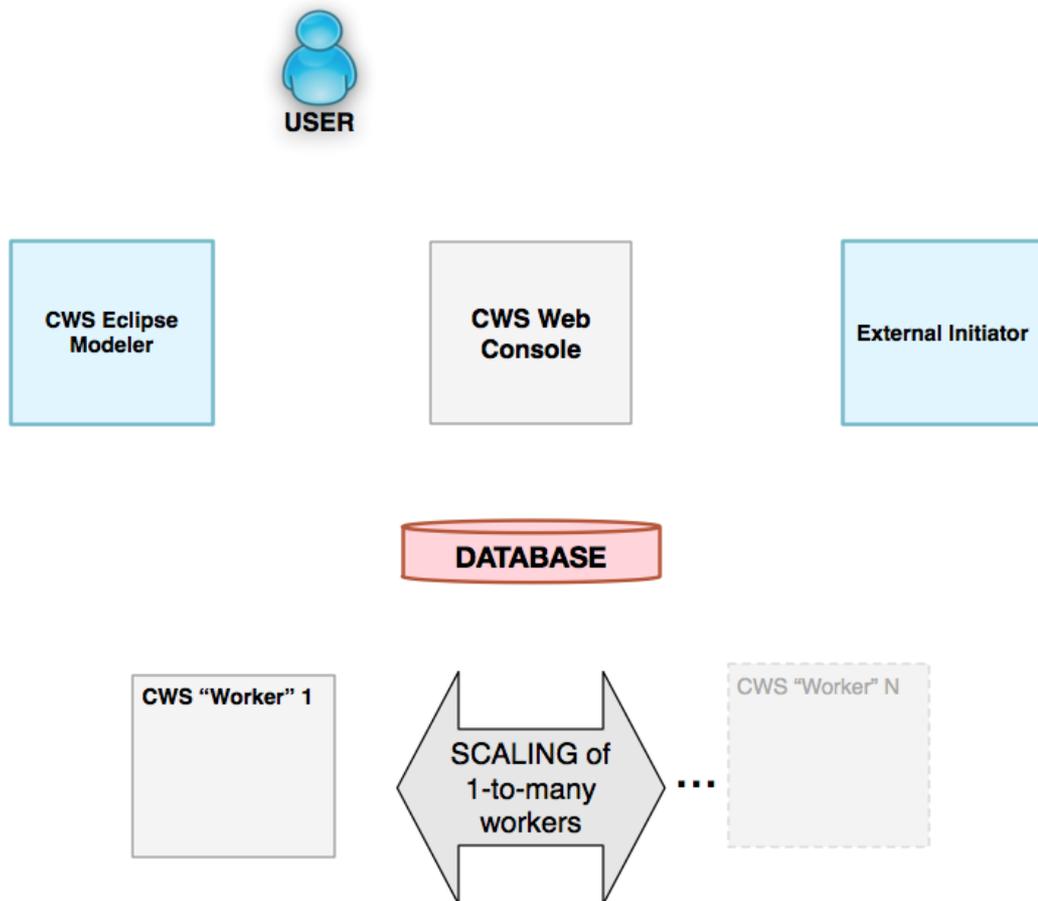


Architecture





Overview





Overview



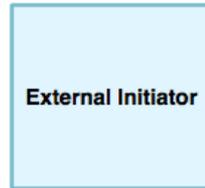
USER



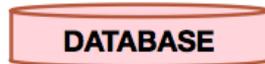
CWS Eclipse Modeler



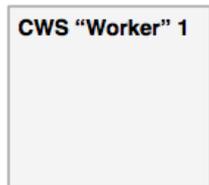
CWS Web Console



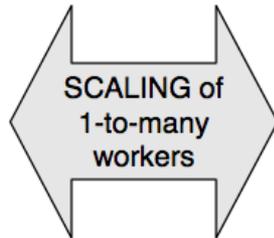
External Initiator



DATABASE



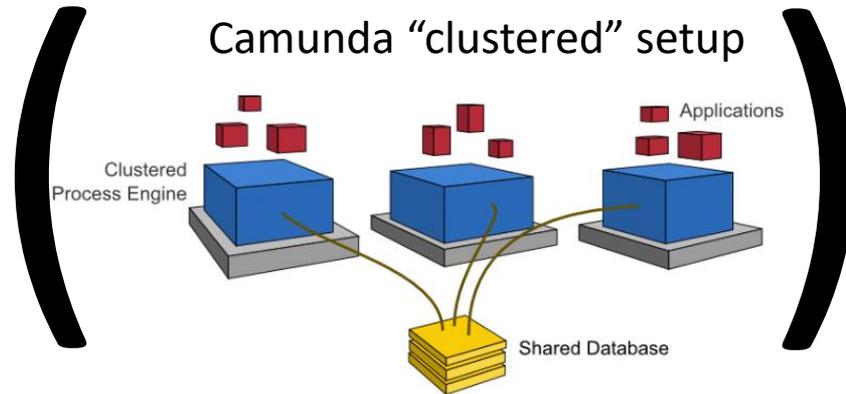
CWS "Worker" 1



SCALING of
1-to-many
workers



CWS "Worker" N



Camunda "clustered" setup

Clustered Process Engine

Applications

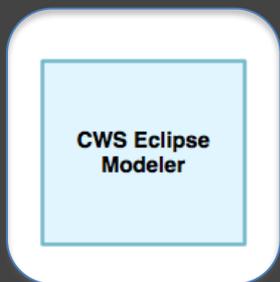
Shared Database



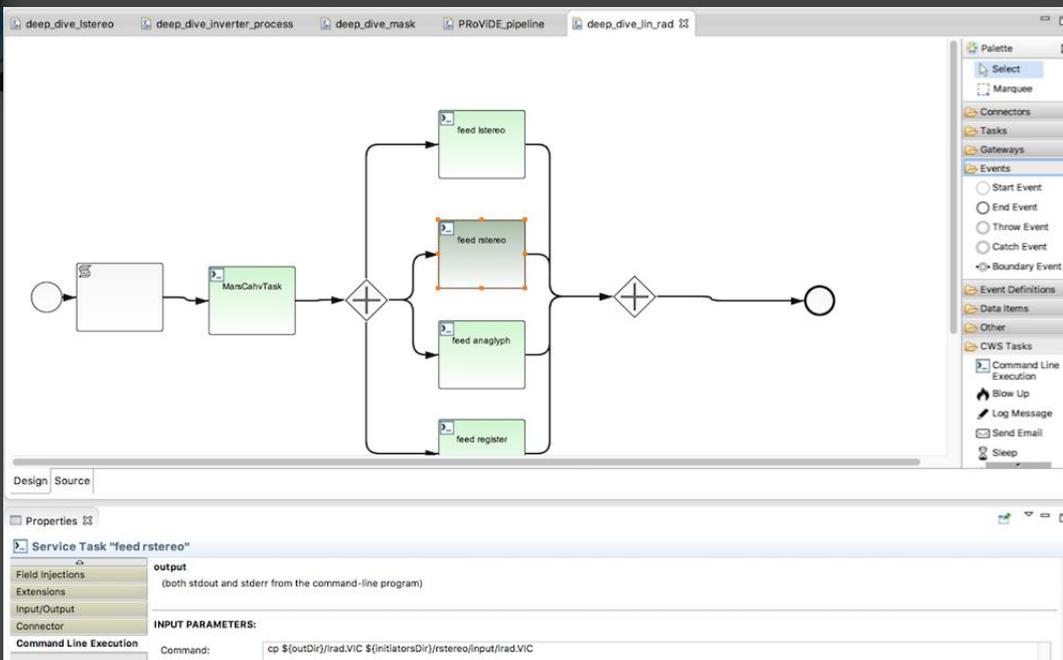
Modeling a Process Definition



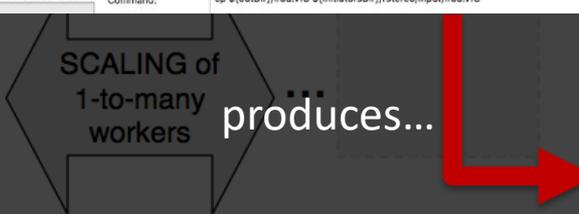
Model a process, using CWS modeler application



USE

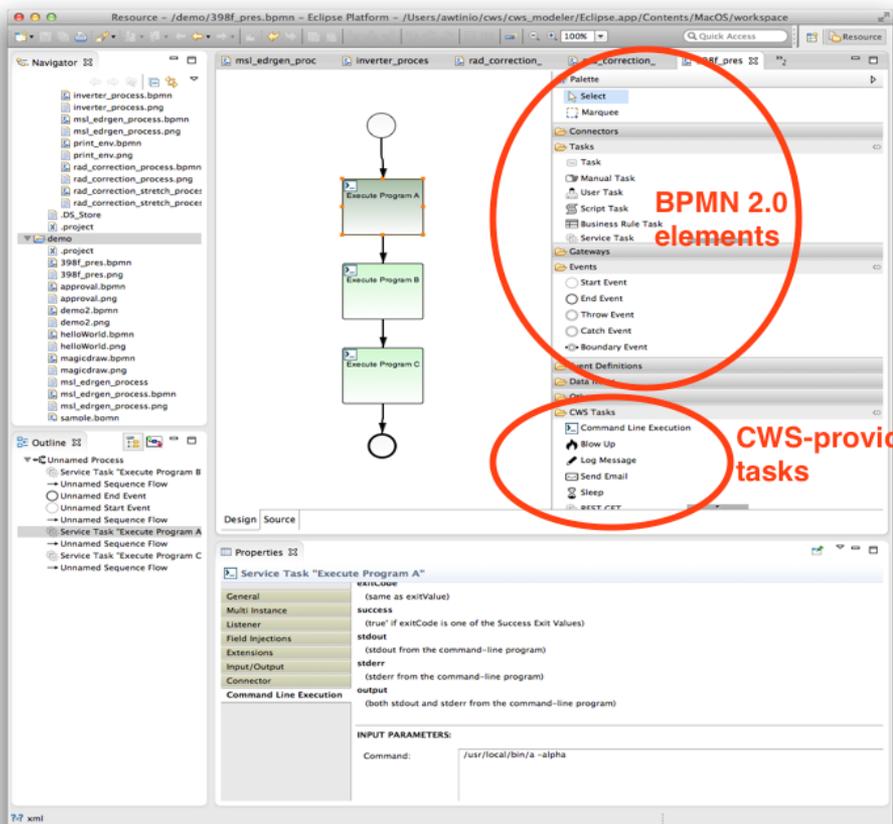


CWS "Worker" 1



XML BPMN 2.0 Process Definition File

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.CommandExecTask" 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:serviceTask>
  <bpmn2:sequenceFlow id="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.CommandExecTask" 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:serviceTask>
  <bpmn2:sequenceFlow id="SequenceFlow_2" name="" sourceRef="ServiceTask_1" targetRef="ServiceTask_2"/>
  <bpmn2:serviceTask id="ServiceTask_3" camunda:class="jpl.cws.task.CommandExecTask" 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"/>

```



CWS Task Properties



The screenshot shows a BPMN editor interface. At the top, there's a Project Explorer with several files. The main canvas displays a BPMN diagram with a Start Event, a Command Line Task, a Logging task, and an End Event. Below the canvas, the Properties window is open for the 'Service Task "Command Line Task"'. The properties are organized into sections: General, Multi Instance, Listener, Field Injections, Extensions, Input/Output, Connector, and Command Line Execution. The Command Line Execution section contains the following fields:

- INPUT PARAMETERS:**
- Command:** sleep 240
- Working Directory:** /Users/ghollins/temp
- Success Value(s):** 0
- throwOnFailures:** true
- Exit Event Map:**
- throwOnTruncatedOutput:** false
- Pre-condition:**
- onPreConditionFail:** ABORT PROCESS



Deploying a Process Definition

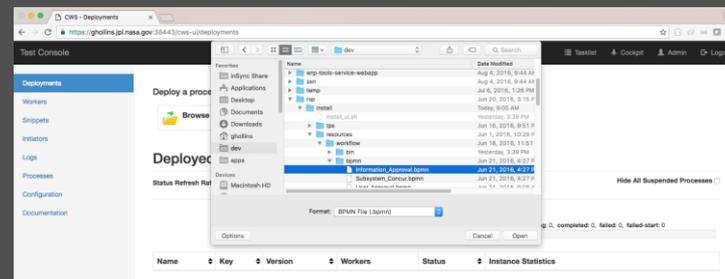


Choose BPMN model file from filesystem

CWS Eclipse
Modeler

CWS Web
Console

External Initiator



Test Console

Deploy a process definition:

Deployed Process Definitions

Status Refresh Rate: 5 seconds

Name	Key	Version	Workers	Status	Instance Statistics
Inverter Process	deep_dive_inverter_process	2	<input type="button" value="view"/>	active	pending: 0, running: 0, completed: 4, failed: 0, failed-start: 0
deep_dive_lin_rad	deep_dive_lin_rad	1	<input type="button" value="view"/>	active	pending: 0, running: 0, completed: 4, failed: 4, failed-start: 0
deep_dive_istereo	deep_dive_istereo	1	<input type="button" value="view"/>	active	pending: 0, running: 0, 3 Completed , failed: 0, failed-start: 0
deep_dive_mask	deep_dive_mask	2	<input type="button" value="view"/>	active	pending: 0, running: 0, completed: 2, failed: 0, failed-start: 0
RAD Process	deep_dive_rad	1	<input type="button" value="view"/>	active	pending: 0, running: 0, completed: 4, failed: 0, failed-start: 0
RAD Stretch Process	deep_dive_rad_stretch	1	<input type="button" value="view"/>	active	pending: 0, running: 0, completed: 4, failed: 0, failed-start: 0
deep_dive_register	deep_dive_register	1	<input type="button" value="view"/>	active	pending: 0, running: 0, completed: 3, failed: 0, failed-start: 0
deep_dive_rstereo	deep_dive_rstereo	1	<input type="button" value="view"/>	active	pending: 0, running: 0, completed: 3, failed: 0, failed-start: 0



Deploying a Process Definition



USER

Click on Deploy button

CWS Eclipse Modeler

CWS Web Console

External Initiator

Name	Key	Version	Workers	Status	Instance Statistics
Inverter Process	deep_dive_inverter_process	2	view	active	pending: 0, running: 0, completed: 4, failed: 0, failed-start: 0
deep_dive_lin_rad	deep_dive_lin_rad	1	view	active	pending: 0, running: 0, completed: 4, failed: 4, failed-start: 0
deep_dive_istereo	deep_dive_istereo	1	view	active	pending: 0, running: 0, 3 Completed , failed: 0, failed-start: 0
deep_dive_mask	deep_dive_mask	2	view	active	pending: 0, running: 0, completed: 2, failed: 0, failed-start: 0
RAD Process	deep_dive_rad	1	view	active	pending: 0, running: 0, completed: 4, failed: 0, failed-start: 0
RAD Stretch Process	deep_dive_rad_stretch	1	view	active	pending: 0, running: 0, completed: 4, failed: 0, failed-start: 0
deep_dive_register	deep_dive_register	1	view	active	pending: 0, running: 0, completed: 3, failed: 0, failed-start: 0
deep_dive_rstereo	deep_dive_rstereo	1	view	active	pending: 0, running: 0, completed: 3, failed: 0, failed-start: 0



Deploying a Process Definition



USER

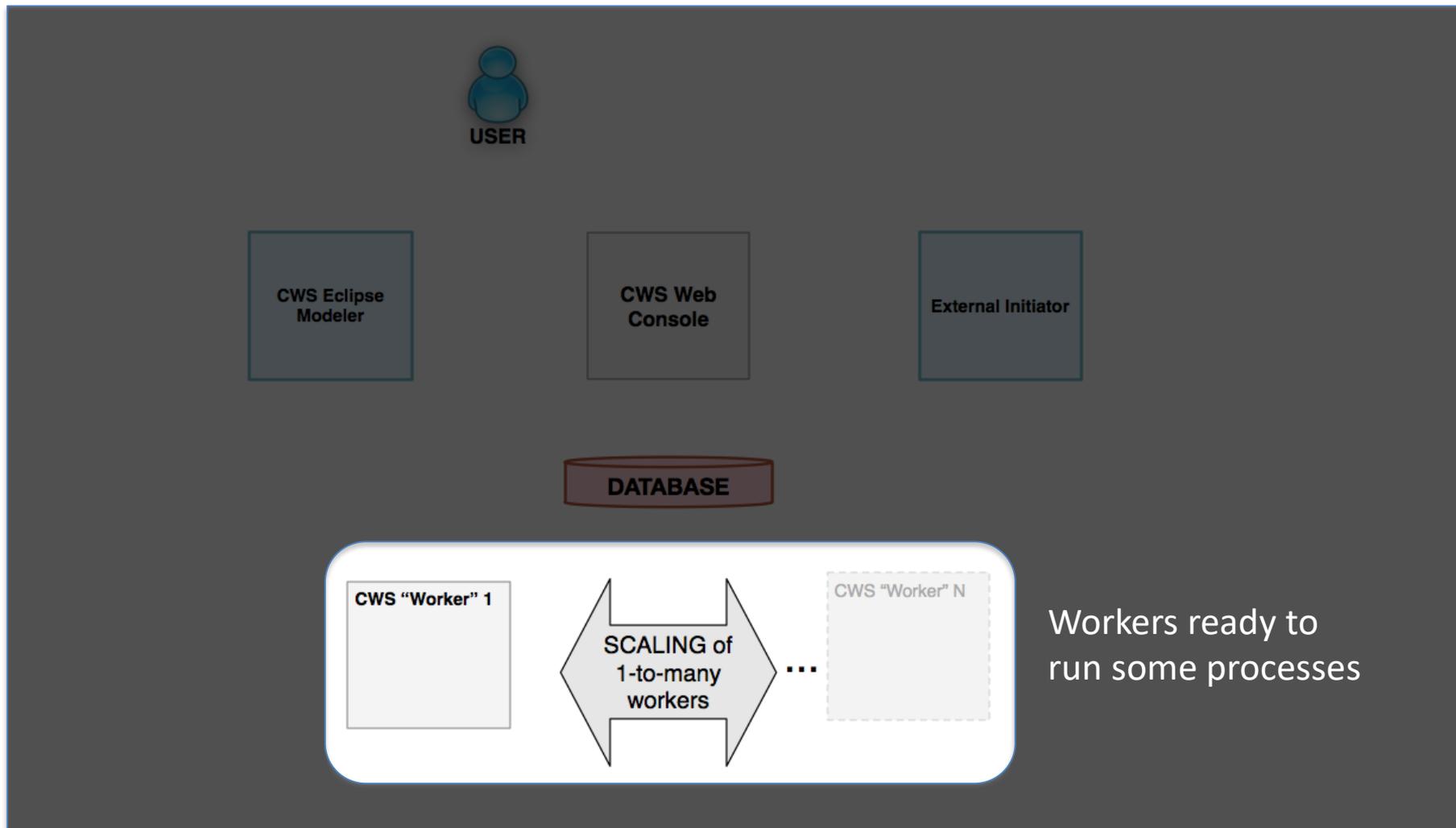
Process Definition is now deployed

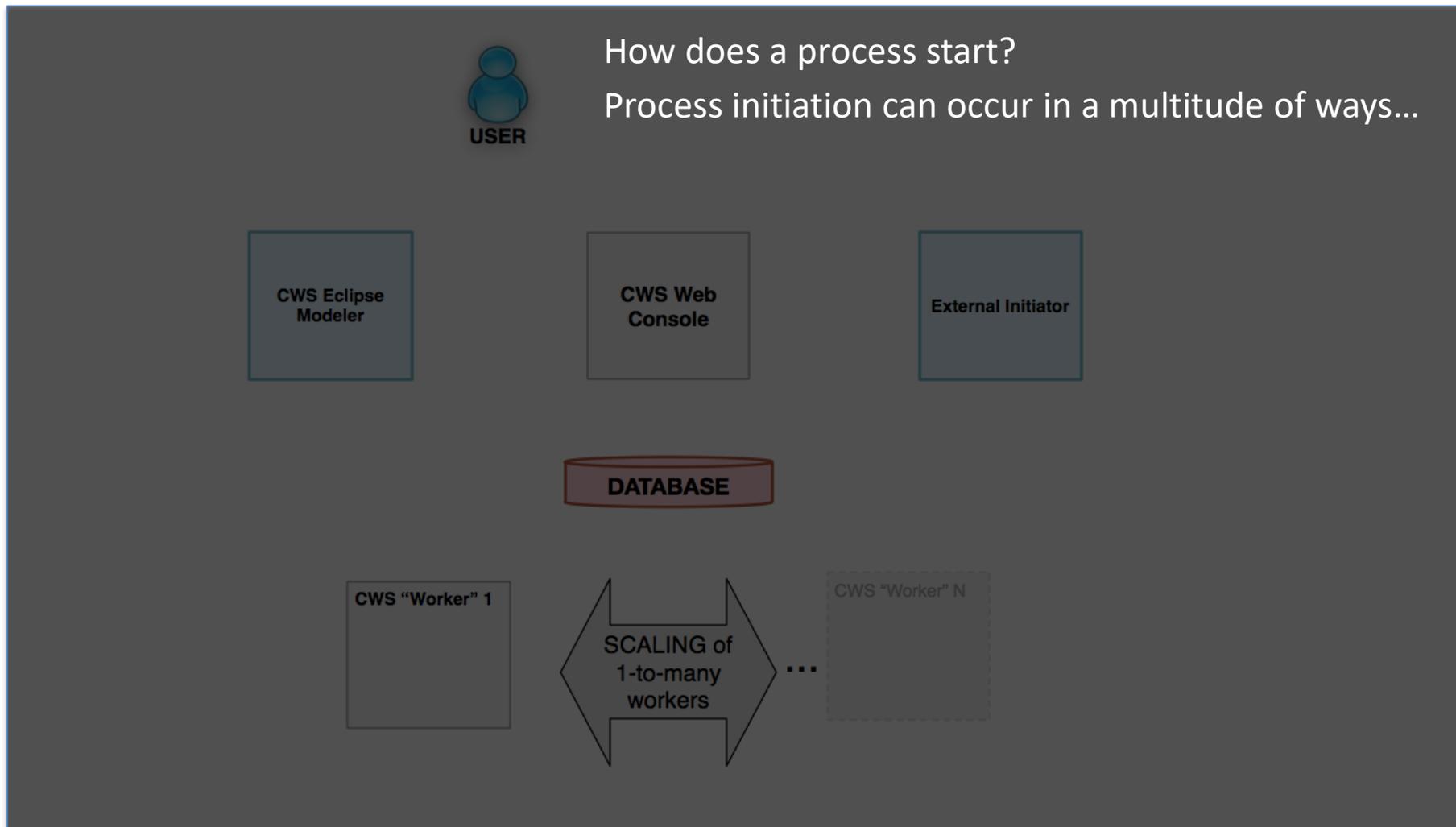
CWS Eclipse Modeler

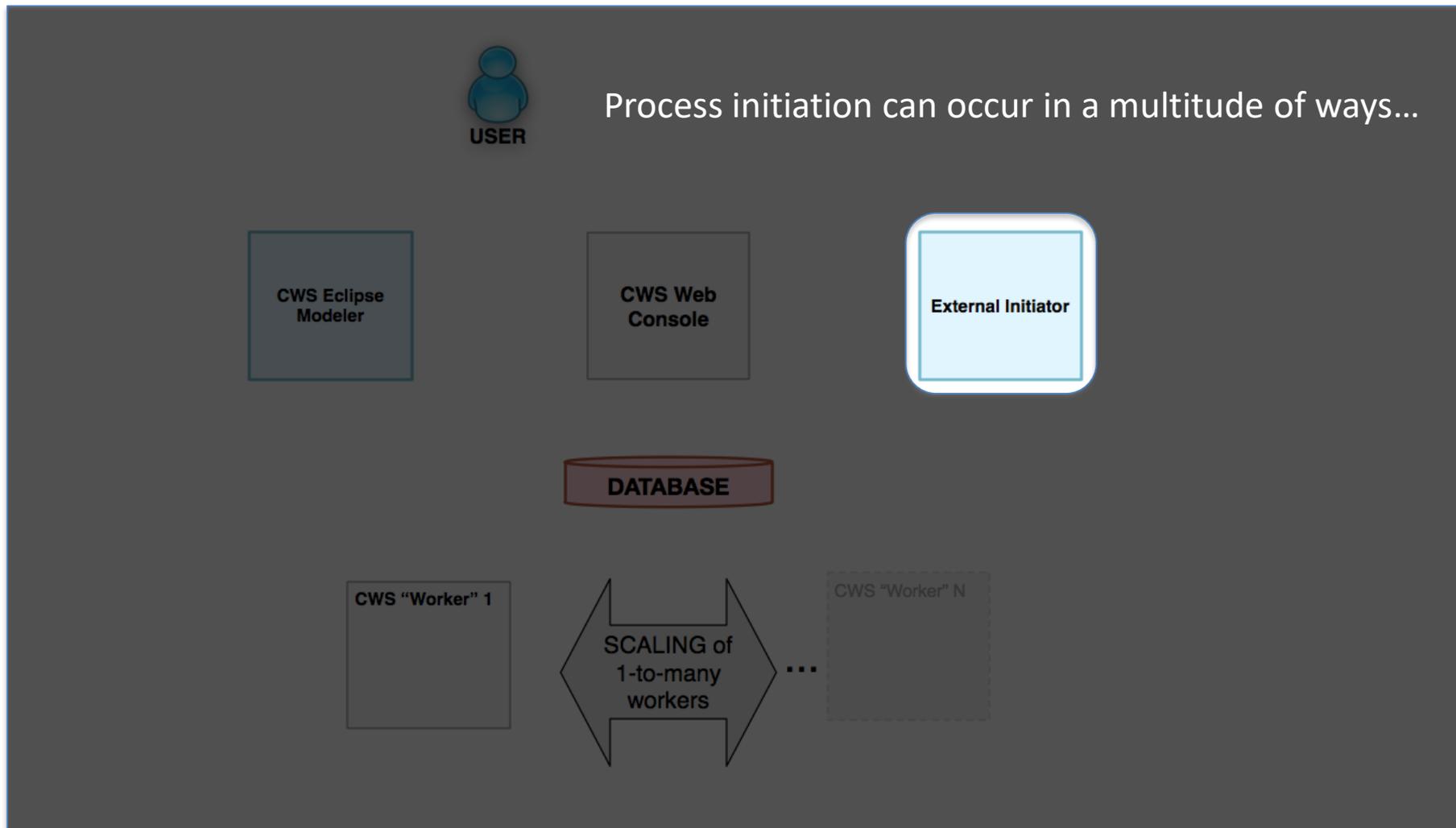
CWS Web Console

External Initiator

Name	Key	Workers	Status	Statistics
Inverter Process	deep_dive_inverter_process	2	view active	pending: 0, running: 0, completed: 4, failed: 0, failed-start: 0
deep_dive_in_rad	deep_dive_in_rad	1	view active	pending: 0, running: 0, completed: 4, failed: 4, failed-start: 0
deep_dive_istereo	deep_dive_istereo	1	view active	pending: 0, running: 0, completed: 3, failed: 0, failed-start: 0
deep_dive_mask	deep_dive_mask	2	view active	pending: 0, running: 0, completed: 2, failed: 0, failed-start: 0
RAD Process	deep_dive_rad	1	view active	pending: 0, running: 0, completed: 4, failed: 0, failed-start: 0
RAD Stretch Process	deep_dive_rad_stretch	1	view active	pending: 0, running: 0, completed: 4, failed: 0, failed-start: 0
deep_dive_register	deep_dive_register	1	view active	pending: 0, running: 0, completed: 3, failed: 0, failed-start: 0
deep_dive_rstereo	deep_dive_rstereo	1	view active	pending: 0, running: 0, completed: 3, failed: 0, failed-start: 0

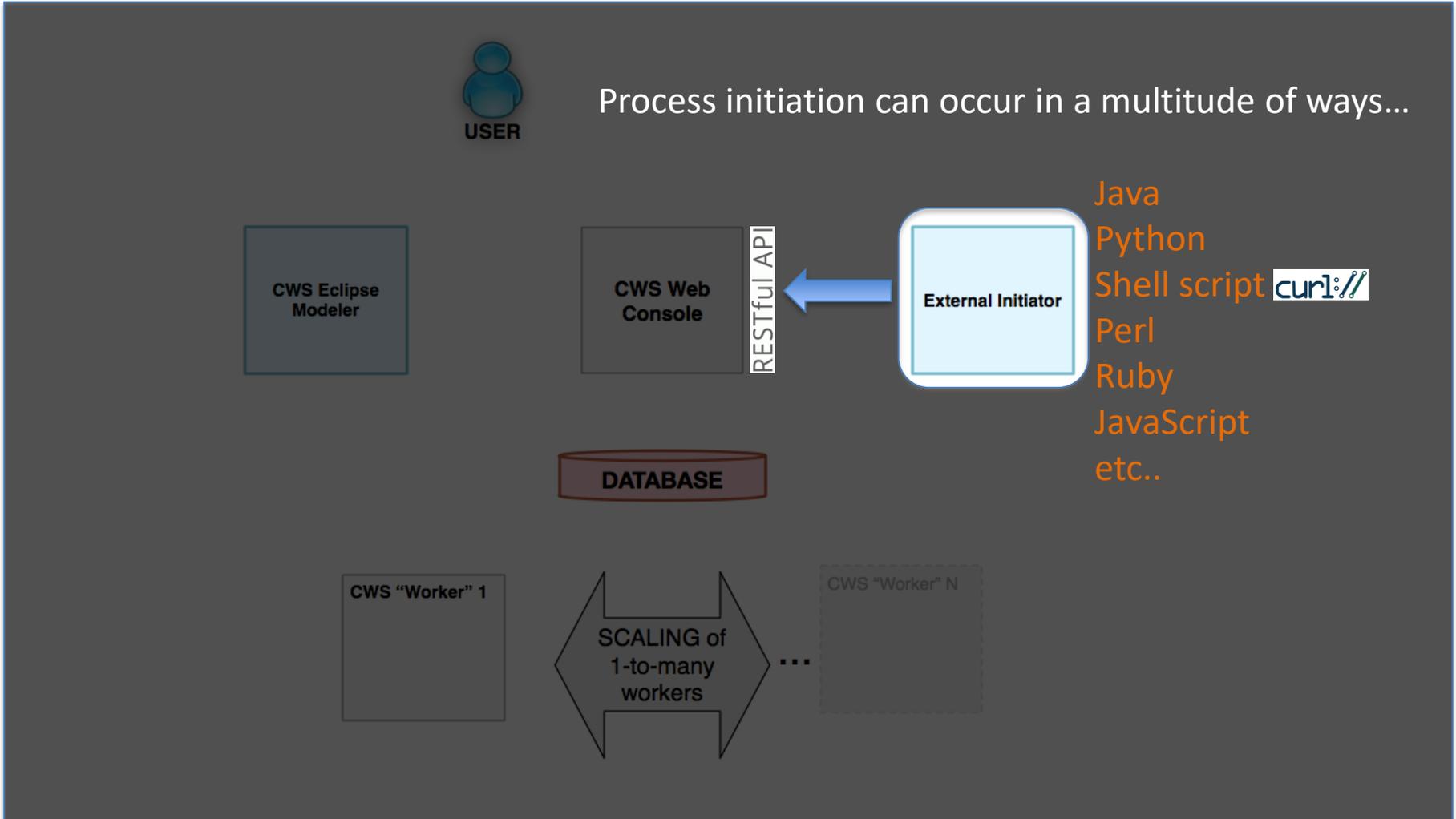






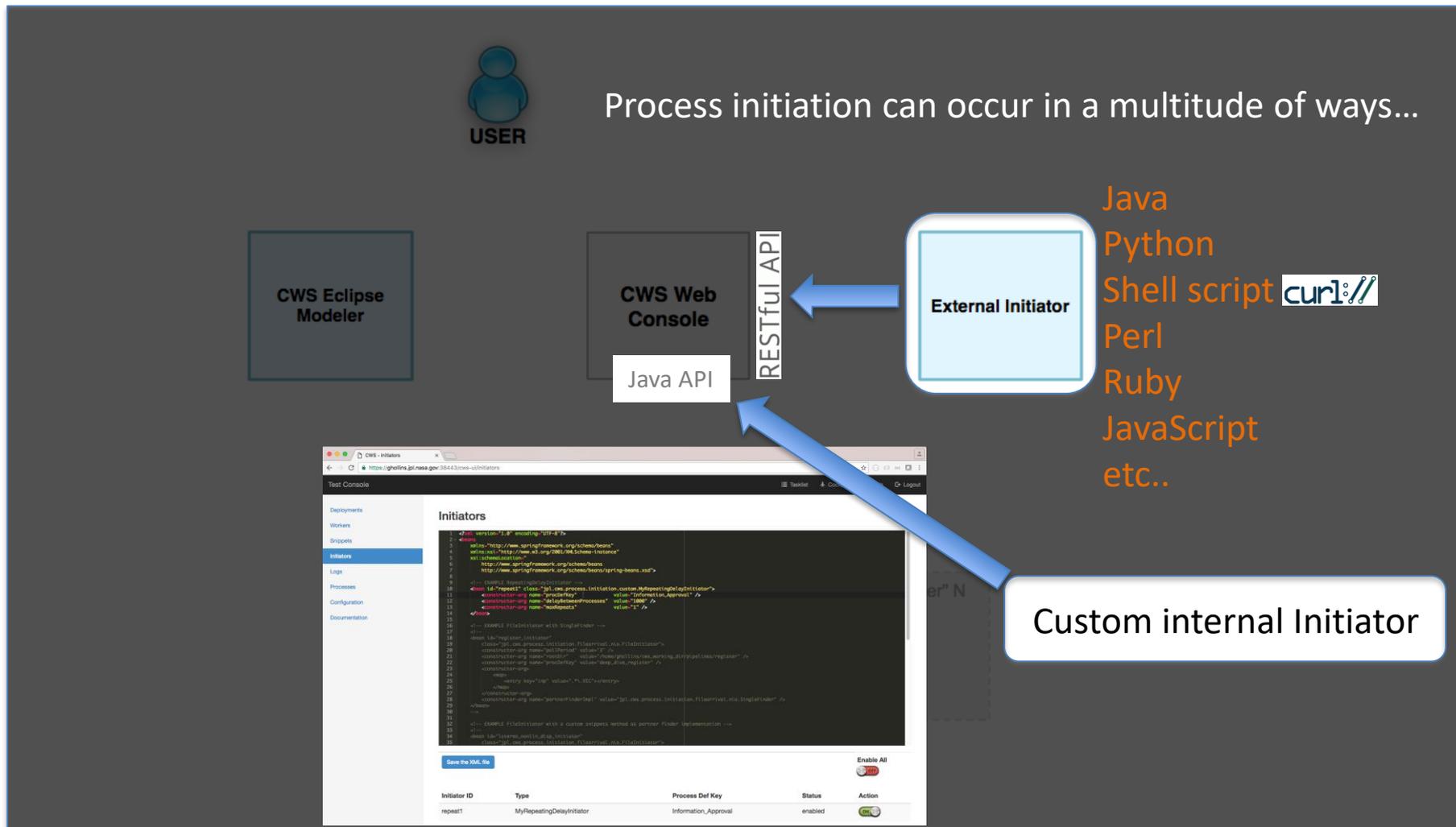


Process Initiation



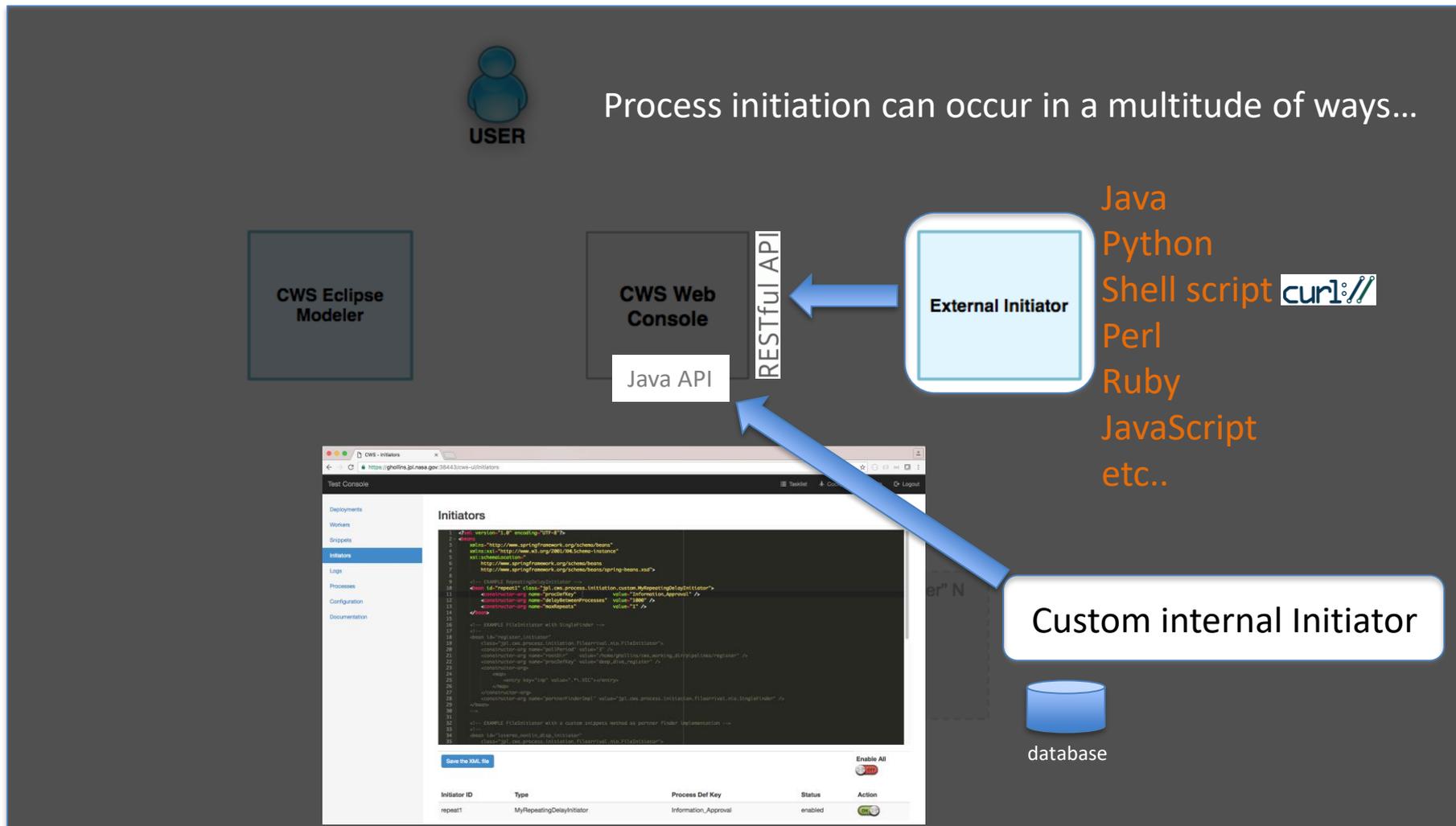


Process Initiation



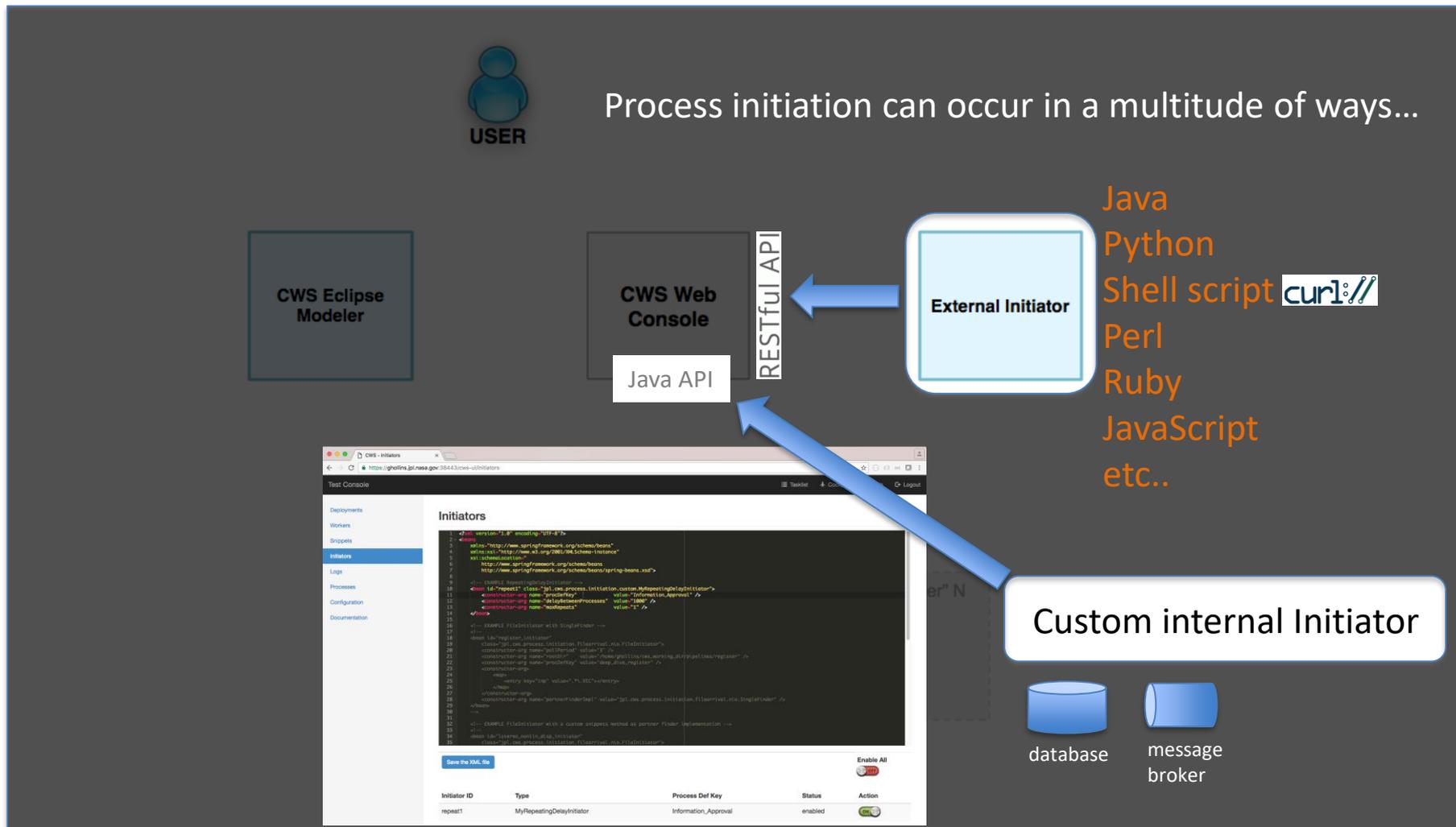


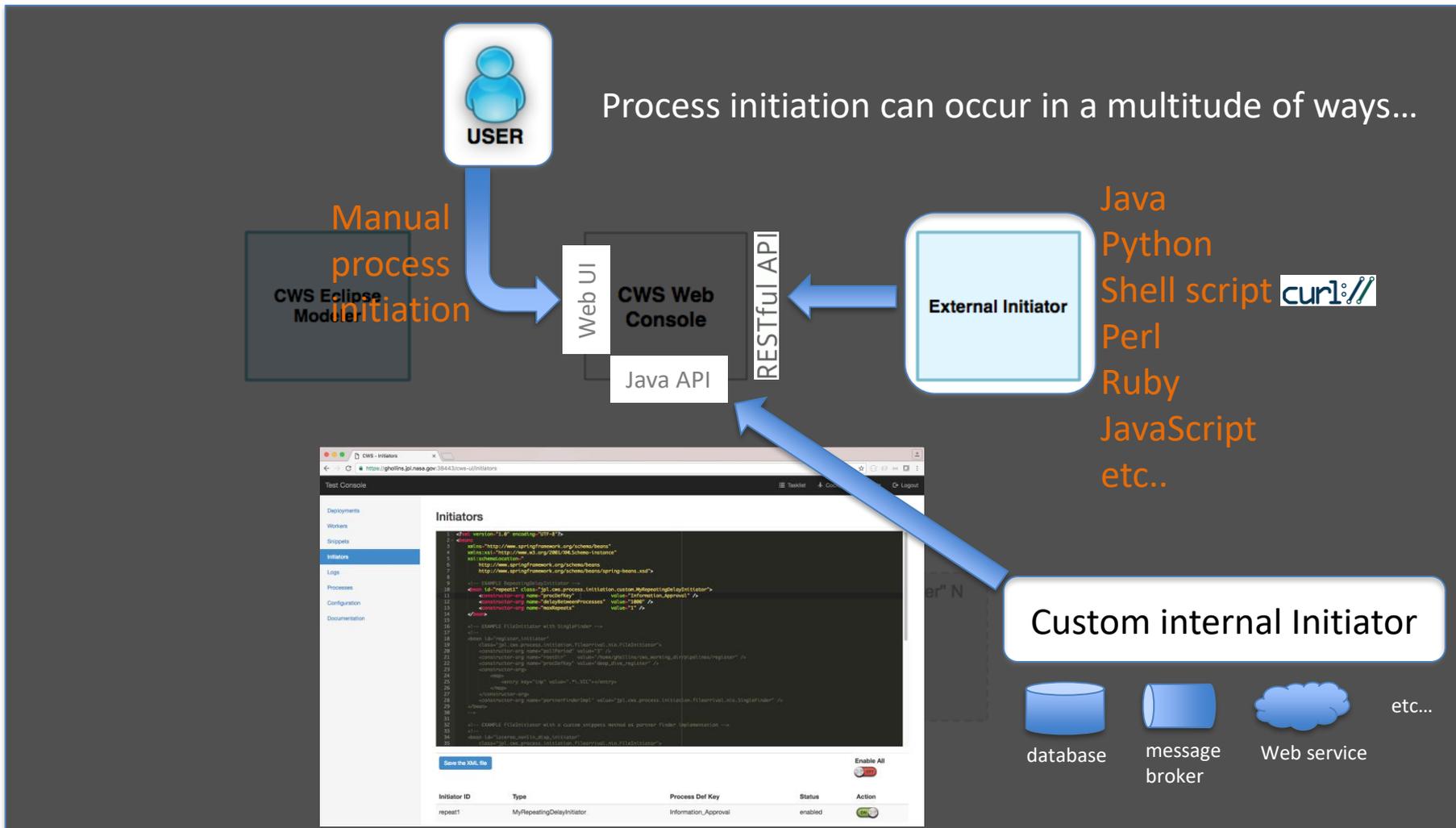
Process Initiation





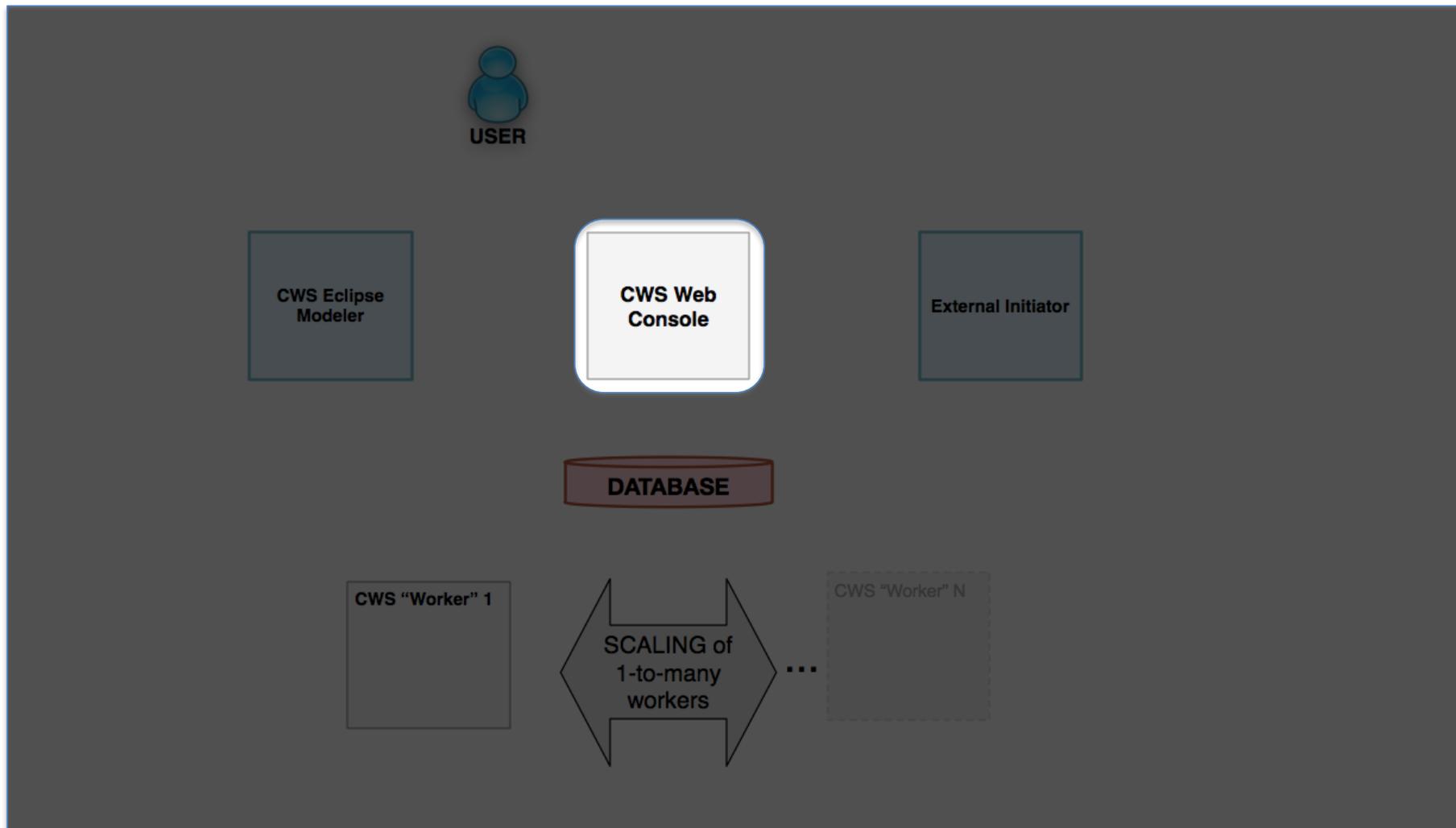
Process Initiation







Process Monitoring and Management





Deploying and Executing Process



Deploy a process definition:

Deployed Process Definitions

Status Refresh Rate: 5 seconds Hide All Suspended Processes

Totals:
pending: 65, running: 0, completed: 320376, failed: 23, failed-start: 16

Name	Key	Version	Workers	Status	Instance Statistics
CommandLineProcess	Process_1	11	enable	suspended	pending: 0, running: 0, completed: 109804, failed: 3, failed-start: 0
CommandLineProcess	SimpleTest_ID2	1	enable	suspended	pending: 4, running: 0, completed: 12, failed: 0, failed-start: 4
SimpleTest_ID3	SimpleTest_ID3	1	enable	active	pending: 0, running: 0, completed: 2, failed: 0, failed-start: 0
SimpleTest_ID4	SimpleTest_ID4	1	enable	active	pending: 6, running: 0, completed: 2, failed: 0, failed-start: 2
SimpleTest_ID5	SimpleTest_ID5	1	view	active	pending: 0, running: 0, completed: 8, failed: 0, failed-start: 2
SimpleTest_ID6	SimpleTest_ID6	1	enable	active	pending: 32, running: 0, completed: 26, failed: 0, failed-start: 2
SimpleTest_ID7	SimpleTest_ID7	3	enable	active	pending: 15, running: 0, completed: 56, failed: 7, failed-start: 0
SimpleTest_ID8	SimpleTest_ID8	1	view	active	pending: 0, running: 0, completed: 132, failed: 10, failed-start: 0
CommandLineProcess	SimpleTest_ID	1	enable	active	pending: 6, running: 0, completed: 3, failed: 0, failed-start: 4
Trigger_ID1	Trigger	1	enable	active	

Process definitions deployed to the CWS via the web based console

Start a process

Search by process name, case sensitive.

Click on the process to start.

- Geometric Correction Process
- Group 398F Sample Process
- Inverter Process
- MSL EDR Generation Process
- Radiometric Correction Process
- Radiometric Correction With Stretch Process

Close

Powered by camunda BPM / v7.2.0

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



Deployments

Workers

Snippets

Initiators

Logs

Processes

Configuration

Documentation

Processes

Filters:

Process Definition:

Select PD

Status:

- Failed
- Complete
- Running
- Pending
- Disabled

Created Date:

From...

To...

Filter

Filters Actions

Showing 1 - 100 (out of 100)

Select	Initiator	Definition Key	Proc Inst ID	Status	Schedule Queued Time	Started on Worker	Process Start	Process End
	priority = 10, timer fire #100	test	111194d3-9324-11e7-9d05-784f437c8140	fail	Sep 6, 2017 9:52:25 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:37 AM	Sep 6, 2017 9:54:38 AM
	priority = 10, timer fire #99	test	105c282a-9324-11e7-9d05-784f437c8140	fail	Sep 6, 2017 9:52:25 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:36 AM	Sep 6, 2017 9:54:36 AM
	priority = 10, timer fire #94	test	0ffa58c2-9324-11e7-9d05-784f437c8140	running	Sep 6, 2017 9:52:24 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:35 AM	
	priority = 10, timer fire #98	test	0fffd70e-9324-11e7-9d05-784f437c8140	running	Sep 6, 2017 9:52:24 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:35 AM	
	priority = 10, timer fire #93	test	0f9a8561-9324-11e7-9d05-784f437c8140	running	Sep 6, 2017 9:52:24 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:34 AM	
	priority = 10, timer fire #97	test	0f9aac7b-9324-11e7-9d05-784f437c8140	running	Sep 6, 2017 9:52:24 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:34 AM	
	priority = 10, timer fire #89	test	0e4e7cf2-9324-11e7-9d05-784f437c8140	fail	Sep 6, 2017 9:52:23 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:32 AM	Sep 6, 2017 9:54:33 AM
	priority = 10, timer fire #95	test	0df5fd9a-9324-11e7-9d05-784f437c8140	fail	Sep 6, 2017 9:52:24 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:32 AM	Sep 6, 2017 9:54:32 AM
	priority = 10, timer fire #96	test	0dff6bf4-9324-11e7-9d05-784f437c8140	fail	Sep 6, 2017 9:52:24 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:32 AM	Sep 6, 2017 9:54:32 AM
	priority = 10, timer fire #90	test	0c58ef43-9324-11e7-9d05-784f437c8140	fail	Sep 6, 2017 9:52:23 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:29 AM	Sep 6, 2017 9:54:30 AM
	priority = 10, timer fire #86	test	0b183175-9324-11e7-9d05-784f437c8140	complete	Sep 6, 2017 9:52:23 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:27 AM	Sep 6, 2017 9:54:48 AM
	priority = 10, timer fire #87	test	0b660363-9324-11e7-9d05-784f437c8140	fail	Sep 6, 2017 9:52:23 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:27 AM	Sep 6, 2017 9:54:28 AM
	priority = 10, timer fire #92	test	0b176e1b-9324-11e7-9d05-784f437c8140	complete	Sep 6, 2017 9:52:23 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:27 AM	Sep 6, 2017 9:54:47 AM
	priority = 10, timer fire #91	test	03fd584f-9324-11e7-9d05-784f437c8140	complete	Sep 6, 2017 9:52:23 AM	128_149_86_225_408224125_1504716288556	Sep 6, 2017 9:54:15 AM	Sep 6, 2017 9:54:36 AM



Monitoring and Management



- Deployments
- Workers
- Snippets
- Initiators
- Logs
- Processes
- Configuration
- Documentation

Logs

Filters:

Process Definitions: All Process Definitions

Process Instances: 0b183175-9324-11e7-9d05-78

Log Level:

- Trace
- Debug
- Information
- Warning
- Error

Search by Keyword:

Start Date:

End Date:

[Filter](#)

Filters

Refresh the logs every 10 seconds

Showing 8 most recent matching rows

Show/Hide Additional Columns: CWS Host CWS Worker ID Log Level Thread Name Process Definition Key Process Instance ID

Time Stamp	CWS Host	Thread Name	Proc Def Key	Message
2017-09-06T16:54:47.928Z	128.149.86.225	pool-2-thread-6	test	ENDING...
2017-09-06T16:54:43.922Z	128.149.86.225	pool-2-thread-6	test	Sleeping... (16000 out of 20000 ms)
2017-09-06T16:54:39.920Z	128.149.86.225	pool-2-thread-6	test	Sleeping... (12000 out of 20000 ms)
2017-09-06T16:54:35.917Z	128.149.86.225	pool-2-thread-6	test	Sleeping... (8000 out of 20000 ms)
2017-09-06T16:54:31.914Z	128.149.86.225	pool-2-thread-6	test	Sleeping... (4000 out of 20000 ms)
2017-09-06T16:54:27.911Z	128.149.86.225	pool-2-thread-6	test	Sleeping... (0 out of 20000 ms)
2017-09-06T16:54:27.910Z	128.149.86.225	pool-2-thread-6	test	Dice roll came up 0 / 10
2017-09-06T16:54:27.910Z	128.149.86.225	pool-2-thread-6	test	STARTING...



Monitoring and Management



Home > Another Failing Process

PROCESS DEFINITION: Another Failing Process Runtime

Version 1

Running Instances

- of the selected version: 4
- of all versions: 4

Filter

Select a filter

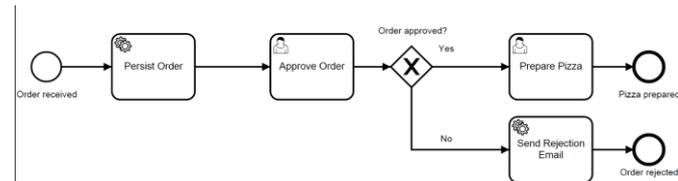
Instance counter

Incident indicator

Start Event → Service Task → Intermediate Timer Event

State	ID	Start Time	Business Key
●	6041fb0d-10e5-11e5-91b6-1e6c20524533	2015-06-12T11:28:00	
●	5e9c3a3e-10e5-11e5-91b6-1e6c20524533	2015-06-12T11:28:00	

State	Activity	Type	Configuration	Action
Active	Service Task	async-continuation		
Active	Intermediate Timer Event	timer-intermediate-transitio		



camunda tasklist

Create task Start process demo Add Comment

My Tasks 6

My Group Tasks

Accounting

John's Tasks

Mary's Tasks

Peter's Tasks

All Tasks

Search for Tasks

Assign Approver Invoice Receipt 50

8 minutes ago Demo Demo

Invoice Amount: 30€ Invoice Number: GPF-23232323

Filter results

Assign Approver Invoice Receipt 50

8 minutes ago Demo Demo

Invoice Amount: 30€ Invoice Number: GPF-23232323

Who should approve this invoice?

Amount: 30€

Invoice Number: GPF-23232323

Approver:

Who should approve this invoice?

Save Complete

Task view



Code Snippets



Test Console Tasklist Cockpit Admin Logout

- Deployments
- Workers
- Snippets**
- Initiators
- Logs
- Processes
- Configuration
- Documentation

Executable Code

CWS provides a mechanism to execute custom code snippets that return a String from BPMN definitions. This is useful, for example when you want to dynamically inject values such as email addresses, command-line arguments, etc.. into various aspects of your existing BPMN tasks.

In order to call a method you define, use the following syntax in your BPMN model:

```
/${cws.methodName([<method_params>...])}
```

For example, here's a command line execution task that uses dynamically generated arguments:

```
/path/to/program.exe -arg1=${cws.getArg1('data', processVar1)} -arg2=${cws.getArg2()}
```

Edit the below code to add or modify methods that are available to your BPMN processes:

```
1 package jpl.cws.core.code;
2
3 import java.util.*;
4 import java.util.regex.*;
5 import java.io.*;
6
7 //-----
8 // This class provides a place to define custom methods;
9 // Out of the box, the CwsCodeBase superclass provides access to the CWS
10 // installation hostname and port via variables:
11 // ${cws.hostname}
12 // ${cws.port}
13 //
14 // Also, provided by the superclass are these methods:
15 // String getEnv(String envVar)
16 //
17 // Example of calling a snippet from a BPMN model:
18 // ${cws.getEnv("JAVA_HOME")}
19 //-----
20
21 public class CustomMethods extends CwsCodeBase {
22
23     public String echo(String arg1) {
24         return arg1;
25     }
26
27     public String heyYou() {
28         return "hey you!";
29     }
30
31     public List<String> foo(String s3Object, String s3BucketName) {
32         List<String> partners = new ArrayList<String>();
33         return partners;
34     }
35
36     public String getRandomUUID() {
37         return UUID.randomUUID().toString();
38     }
39 }
```

[Validate & Save](#) [Reload Editor with Last Successfully Compiled Code](#)



Code Snippets Example Usage



In order to call a method you define, use the following syntax in your BPMN model:

```
`${cws.methodName( [<method_params>...])}
```

For example, here's a command line execution task that uses dynamically generated arguments:

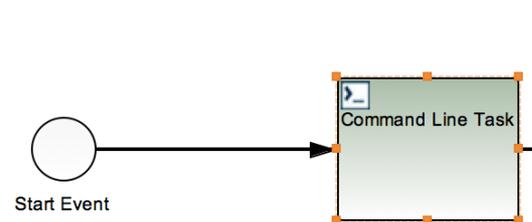
```
/path/to/program.exe -arg1=${cws.getArg1('data', processVar1)} -arg2=${cws.getArg2()}
```

Edit the below code to add or modify methods that are available to your BPMN processes:

```

1  package jpl.cws.core.code;
2
3  import java.util.*;
4  import java.util.regex.*;
5  import java.io.*;
6
7  //-----
8  // This class provides a place to define custom methods.
9  // Out of the box, the CwsCodeBase superclass provides access to the CWS
10 // installation hostname and port via variables:
11 //   ${cws.hostname}
12 //   ${cws.port}
13 //
14 // Also, provided by the superclass are these methods:
15 //   String getEnv(String envVar)
16 //
17 // Example of calling a snippet from a BPMN model:
18 //   ${cws.getEnv("JAVA_HOME")}
19 //
20 //-----
21 public class CustomMethods extends CwsCodeBase {
22
23     public String echo(String arg1) {
24         return arg1;
25     }
26
27     public String heyYou() {
28         return "hey you!";
29     }
30
31     public List<String> foo(String s3Object, String s3BucketName) {
32         List<String> partners = new ArrayList<String>();
33         return partners;
34     }
35     public String getRandUuid() {
36         return UUID.randomUUID().toString();
37     }

```



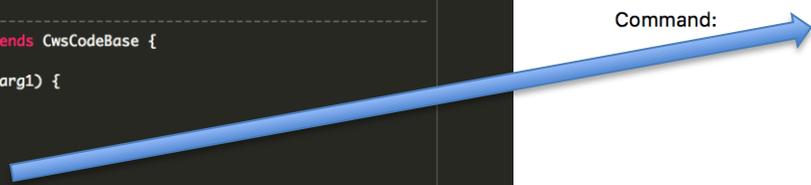
Design Source

Command Line Task"

INPUT PARAMETERS:

Command:

```
echo ${cws.heyYou()}
```





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
 - Secured connections



CWS Future Work



- *Move away from Eclipse modeler, and go with Camunda / Web modeler*
- *Out-of-the box components and installation for external tasks*



Demo



Contact Info



Galen Hollins (CWS Task Lead / Software Engineer)

Jet Propulsion Laboratory
California Institute of Technology Pasadena CA

Galen.A.Hollins@jpl.nasa.gov

Jim Wood (CWS Software Engineer)

Jet Propulsion Laboratory
California Institute of Technology Pasadena CA

James.F.Wood@jpl.nasa.gov

Adrian Tinio (MGSS IDS System Engineer)

Jet Propulsion Laboratory
California Institute of Technology Pasadena CA

Adrian.W.Tinio@jpl.nasa.gov



Questions/Comments