UC3 Design

SDF Monitoring

Document Information

General	
Document Title:	UC3 Design
Area or Project:	SDF Monitoring
Document Code:	
Document Owner:	Simon Rosman
File Name	UC3 Design.docx

History

Version	Date	Status	Author	Affected pages:	Approved by:
v0.1	17-11-2009	New	Simon Rosman	All	
v1.0	18-11-2009	Final	Simon Rosman		

Distribution

Version	Name	Date	Publication Status

TABLE OF CONTENT

1. Introduction

1.1 Content

This document describes the solution that will be created for UC3. Topics dealt with are:

- The current monitoring environment
- The use case
- Two possible solutions

1.2 Audience

The intended audiences are:

- SaaS Deployment Framework team
- Operational management team

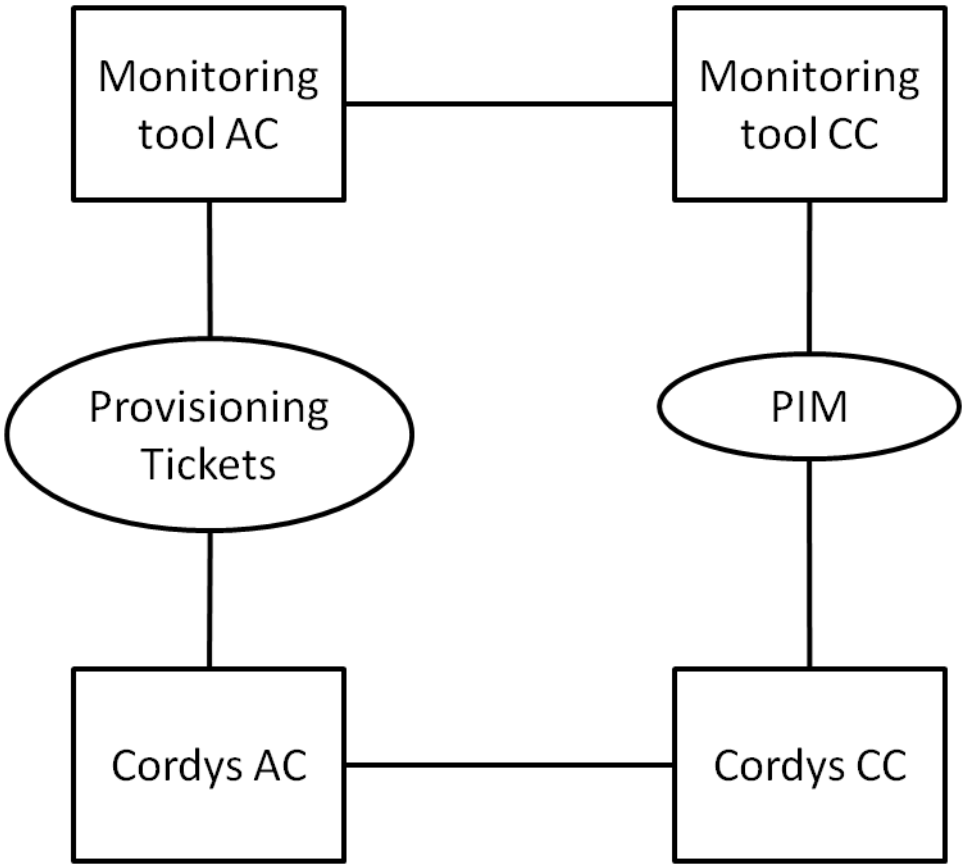
1.3 Purpose of this document

This document serves a number of purposes, namely:

- To describe the different scenarios for solving the described use case.

2. Monitoring Environment

The monitoring tool is supposed to monitor multi cluster provisioning. Provisioning consists of a provisioning ticket on the Admin Cluster, and of one or more Process Instances on a customer cluster.



3. UC3 Description

UC3: Engine task in 'waiting' state on Admin Cluster caused by error on Customer Cluster

Short description
If a provisioning task which is running on a Customer Cluster fails, this must be detected by the monitor.

Scope
Multi-cluster

Purpose
This use case can prove if the monitoring tool is able to provide previously missing information about the multi-cluster environment. E.g. it tests the complete environment.

Actors
- User / Global Operator
- Platform Operator

Scenario
1. A Global Operator starts a provisioning task by creating a user account + assigning an application to this user.
2. The task is sent to the customer cluster.
3. The customer cluster starts executing the provisioning task.
4. Create failure on customer cluster which is currently undetectable.
5. The monitor detects the error on customer cluster (e.g. by watching the Process Instance Manager).
6. The monitor updates the Engine task on the Admin Cluster with the error information.
7. The monitor informs the Platform Operator.

Success Conditions
This use case is successful if the Platform Operator gets notified within 2 minutes after the failure.

4. UC3 Solution

4.1 Scenario 1 – Monitor 'consistency' between Engine tasks and PIM

Scenario steps:

Admin Cluster:

1. The monitoring tool requests the list of the Provisioning engine tasks (tickets). This list contains the guids of the Process instances on the Customer Cluster
2. The monitoring tool sends the list to the Monitoring tool at the Customer Cluster.

Customer Cluster:

3. The monitoring tool reads the list of Process Instance guids from the list received from the Admin Cluster.
4. The monitoring tool requests the list of process instances from the Customer Cluster which are in the list.
5. The monitoring tool searches for aborted processes.
6. If any aborted processes are found, the monitoring tool orders the Admin Cluster to set the corresponding ticket on error status.

Requirements:

- SDF must store the relationship between Provisioning tickets and their corresponding BPM's.

Pros:

- Provisioning tickets can be updated according to the status of the BPM's on the Customer Cluster.

Cons:

- SDF must be modified to store the guid of the BPM related to a provisioning ticket, or the BPM must be modified to store the guid of the Provisioning ticket.

4.2 Scenario 2 – Monitor 'time window' from PIM

Scenario steps:

Customer Cluster:

1. The monitoring tool requests the list of Process Instances from the Customer Cluster.
2. The monitoring tool filters the list according to a configured time window.
3. The monitoring tool filters for SDF processes and instantiated sub processes.
4. The monitoring tool filters for aborted processes.
5. The monitoring tool goes in error state, if any aborted processes are found.

Pros:

- In this scenario, no alternations to SDF are necessary.

Cons:

- Unable to determine which individual BPM's are acknowledged.
- Unable to relate BPM's to their corresponding provisioning tickets.