Razorback Requirements Document

**Q4 Localities**

Version 0.3

Revision 1

10/28/2011

## 

Table of Contents

[Introduction 3](#_Toc310420156)

[Customer Focus Statement 3](#_Toc310420157)

[Requirements 3](#_Toc310420158)

[Localities 3](#_Toc310420159)

[Data Transfers 3](#_Toc310420160)

[Transfer Methods 3](#_Toc310420161)

[Implementation 4](#_Toc310420162)

[Message Format 4](#_Toc310420163)

[Message Data Types 4](#_Toc310420164)

[Metrics 5](#_Toc310420165)

[Impact 5](#_Toc310420166)

[Future Work 5](#_Toc310420167)

[Enhancements 5](#_Toc310420168)

## Introduction

A Razorback instance is capable of spanning a large network with more than one physical or logical location.

## Customer Focus Statement

Razorback will attempt to minimize the amount of network traffic that takes place between localities. This will help both performance of large scale deployments and cost related to data transfer over long distance links.

## Requirements

### Localities

* Must support more than one.
* Unconnected locality – No dispatcher is allowed to run in this locality.
* Locality preference – The API will provide a means of allowing an administrator to specify the preferred sequence of localities to contact when there is no dispatcher in its current locality.

### Data Transfers

* Submitted blocks will be transferred to the master dispatcher (MD) in the current locality, if there is no MD in the current locality then the API will search for dispatchers in the locality preference list in sequence. If no dispatcher is found in the locality list then one is picked at random from the currently online MDs from the remaining localities.
* Inspectors will always contact the MD in its current locality to transfer the block from, if there are no dispatcher online in the current locality then the same search algorithm is used to find a dispatcher to transfer the block from.
* When the dispatcher is contacted to retrieve a block that is not currently stored in its locality it will look up the list of localities that the block is currently stored in and transfer the block to its locality before returning the block to the requestor.

### Transfer Methods

Razorback will support the following methods of block transfer:

* Interior transfers will always take place over a shared file system.
* Exterior transfers will take place of one of the following modular transfer protocols
  + SSH
  + Other modules to be made available at a future date.

## Implementation

### Message Format

Messages will be placed in the MQ in JSON objects, this is slightly more efficient than using XML and just as well supported by a range of programming languages.

### Message Data Types

## Metrics

* Message size
* Message count

## Impact

* Requires changes to the dispatcher:
  + Event submission
  + Judgment processing
  + Log processing
  + Flags coping thread

## Future Work

### Enhancements

* TBD