Web Policy High Level Design

This document is highly confidential! No Disclosure!

# Document History

|  |  |  |
| --- | --- | --- |
| Cheng Yi | 2010/12/15 | 1. Add Module and Module Dependency |
| Cheng Yi | 2010/12/21 | 1. Add Module Repository |

# Architecture

## Modules and Module Dependency



Utility

1. Web Utility

Common

1. Policy Event Interface
2. Action Interface

Data Source Engine

1. Maintain various data sources
2. Map from the data to Policy Event (including how to get the data)
3. Regularly (pull or push) fetch the data and map them to Policy Event.
4. Maintain the Policy Event Cache, if new/updated event found, trigger the policy engine.

Policy Engine:

1. Rule Maintenance
2. Event Firing and Action Triggering

Notification Engine:

1. Notification Template Definition
2. Notification Service

Studio-Common

1. Event Model Studio.
   1. Defining the classes.
2. General Action Studio.
   1. Defining the scripts, statements and expressions.

Studio-Data Source

1. Defining the data sources
2. Defining the mapping from data source to policy event.
   1. Steps how to perform the mapping.

For web data source define the navigation of the web.

Studio-Policy

1. Define the rules

Studio-Notification

1. Define the notification templates

Repository:

1. The interface between studio and engine.
2. Various studios update the Event definition, Action definition, Policy definition, Data Source Definition, Notification Template definitions and etc into the repository.
3. Various engines fetch these definitions from the repository and are notified of the updates.
4. Repository is able to do the version control, deployment, etc.