

ImperiusProposal

Abstract

Policy systems allow administrators to define policies that guide the automated administration of resources. Such automation saves time and effort by simplifying resource-management.

We proposed to develop a policy-based management infrastructure that automates administrative tasks by executing policies written in the "Simplified Policy Language" (SPL), a standards-based policy language.

Proposal

This proposal seeks to create a project within the Apache Software Foundation to (1) develop a policy evaluator for the SPL policy language, and (2) develop bindings between the policy evaluator and projects such as Apache Geronimo or Apache Tomcat.

Background

As computer systems become more complicated, they continue to become increasingly complex and costly to manage. Various studies have shown that the cost of managing systems often exceeds the cost of purchasing them. The goal of the Imperius project is to reduce management burden by allowing administrators to specify policies that replace manual administrative actions.

Rationale

Policy-based management is one approach to reducing the cost of systems administration. Administrators define policies that express a "management intent", and the policy-based management system executes the policy, thereby reducing the burden on the administrator. For example, a policy might state "backup the database daily at 1am." A policy management system would interpret the policy, and trigger the database to perform a backup at the assigned time, offloading that task from the administrator.

To allow more flexible semantics, such policies are often expressed as "If-Then" (also called "Condition-Action") rules. For example, "If the packet comes from subnet 6.7.8.*, Then place it on the high-priority outgoing queue" or "If the data in the database have changed by 10%, Then perform an incremental backup." Such policies allow for more complex automation.

To implement these kinds of rule-based policies, the policy system must interact with the system's APIs. In the second example above, the system relies on the ability to measure the percentage of data that have changed since the last backup (or to compute that value from measurable data). Similarly, to execute the "then" clause, the policy system must be able to start an incremental backup on the database using the database's API. Expressed generally, a policy system must have a binding to the "instrumentation layer" or API for the system under management.

Thus, policy systems consist of two major components: an evaluation engine and a binding to an instrumentation environment. The evaluation engine (1) accepts policies expressed according to a well-defined grammar, (2) collects the data needed to evaluate the policies, and (3) actuates the "Then" section as appropriate.

This project would build an SPL evaluation engine and bindings to various Apache and non-Apache APIs.

The design of SPL, a Preliminary DMTF standard, is inspired by existing policy languages and models including PDL (policy definition language) from Bell Laboratories, the Ponder policy language from Imperial College, and other policy languages.

The basic unit of a SPL policy is a policy rule. An SPL policy rule consists of a condition, an action, and other fields. Multiple policy rules can be grouped into a policy group. Policy groups can be nested – i.e., a policy group can contain another policy group. For the specification of the policy condition, SPL provides a rich set of operators that act on the following types: signed and unsigned short, regular and long integers, float and long float, character, string, Boolean, and calendar.

We expect the community to develop a number of bindings between the evaluation engine and APIs. Examples include the API for Geronimo or Tomcat, and standard interfaces such as WSDM and CIM. The community may choose to develop a wide range of bindings, all leveraging the common SPL engine and policy syntax.

The Imperius Policy Engine will be implemented in the form of a Java application. The SPL application will provide the following functionality:

1. [PolicyManager](#) - The [PolicyManager](#) acts as an orchestrator

delegating tasks to its subcomponents.

2. [PolicyDataStore](#) - The [PolicyDataStore](#) is responsible for two major tasks: Policy Storage and the creation of Internal Policy Objects:

1. Policy Storage: This involves persisting of the policies

into a repository along with caching of the Internal Policy Objects for use in evaluation

b. Internal Policy Object Creation: This involves syntax, semantic analysis of the textual policy stored in the repository and the creation of abstract syntax trees and symbol tables for use in the evaluation of the policies.

3. Policy Evaluator - The [PolicyEvaluator](#) is responsible for the actual evaluation of the policies. It uses the Internal Policy Object created by the [PolicyData Store](#). It iterates over the list of all object instances that satisfy the anchor object criterion (refer to the SPL specification for more information) evaluating conditions and performing actions.

Development will focus on features, performance, scalability, quality, and extensibility.

Why ASF

The developers of Imperius are interested in joining the Apache Software Foundation because:

- we would like to encourage binding of the SPL language to numerous product APIs, especially existing Apache projects, in an open development environment.
- We would like development of Imperius to take place within the ASF legal, licensing and oversight structure.

Initial Goals

The initial goals are to develop an SPL evaluation engine and bindings to the APIs for:

1. Apache Tomcat
2. Apache HTTP server
3. WSDM (<http://www.oasis-open.org/committees/wsdm/>)
4. CIM

Current Status

IBM has a prototype SPL engine, and a binding to CIM APIs.

Meritocracy

The Imperius project will be meritocratic. The project will follow the guidelines (<http://apache.org/foundation/how-it-works.html#meritocracy>) of the Apache Software Foundation. We plan to proactively recruit individuals in the Community to: (1) provide function, performance and stability enhancements to the core engine; (2) encourage bug fixing; (3) improve code structure; and (4) provide binding to other APIs. We believe that open, meritocratic development will result in a stronger engine with a broader range of binding than we could develop internally.

Community

The initial community will be comprised of developers from Sun and IBM, both of which have substantial interest in the project. The project will be actively communicated within the ASF and to both the DMTF and SNIA communities, encouraging growth in the Imperius community.

Core Developers

The initial core developers will come from IBM and Sun. One Sun committer has experience with [OpenPegasus](#) and Open Solaris. Two IBM developers have worked on [OpenPegasus](#). Other strong developers have expressed interest, and we expect them to earn committer status quickly.

Alignment

As noted above, Imperius can supplement the existing management of Apache products.

In addition the code has been implemented in Java, and uses other OSS such as ANTLR and JUNIT. ANT may be used as the code evolves.

Known Risks

Orphaned products

This proposal is not the result of an orphaned or abandoned IBM internal project. This is a proposal we're making with a full commitment to construct an active and successful project. Initial committers are users of this package, and have an interest in its ongoing viability.

Inexperience with open source

The initial committers have varying degrees of experience with open source projects, as noted above. All have been involved with source code that has been released under an open source license, but they lack experience developing code specifically with the ASF development process. We do not, however, expect any difficulty in executing under normal meritocracy rules.

Homogeneous Developers

The project will begin with developers from both Sun and IBM, and we expect that developers from the CIM-SPL community and from other ASF projects to join. In addition, we have received design comments from developers associated with the [OpenGroup](#).

We do acknowledge the risk that the project will not garner interest beyond IBM and Sun, though we believe that interest from communities involved in the DMTF reduces the risk.

Reliance on salaried developers

It is expected that Imperius development will occur by developers on both salaried time and on volunteer time, after hours. While there is currently a reliance on salaried developers (currently from IBM, but it's expected that other company's salaried developers will also be involved), IBM will continue to support the project in the future by dedicating 'work time' to Imperius.

Relationships with other Apache Products

A Excessive Fascination with the Apache Brand

The SPL language and prototype were developed to address the need for a common policy engine that can be used by a number of others projects, some part of the ASF. We believe that such synergy, along with the shared philosophy, and process and licensing consideration, make the ASF the proper community.

Scope

The scope of the Imperius project at ASF would be to continue the project development, including adding new features and bindings, and improving overall code quality, scalability and performance.

Documentation

The CIM-SPL specification: http://www.dmtf.org/standards/published_documents/DSP0231.pdf

Initial Source

The initial source, a prototype SPL engine, has been developed.

Source and Intellectual Property Submission Plan

The source comes from IBM. IBM will submit a CCLA if the incubator proposal is accepted.

Information about DMTF policies can be found in references [1], [2] and [3].

External Dependencies

None

Cryptography

None implemented in the code.

Resources Required

(3.1) mailing list(s)

- imperius-private (with moderated subscriptions)
- imperius-dev
- imperius-commits
- imperius-user

(3.2) Subversion or CVS repositories

Imperius would like to use a SVN repository.

(3.3) Jira (issue tracking)

Since Imperius would have its own release cycle, it should have its own JIRA project

- Project Name: Imperius
- Project Key: Imperius

Initial Committers

- Mike Nunez <Michael.Nunez at Sun.COM>
- Neeraj Joshi <jneeraj at us.ibm.com>
- Mark Carlson <Mark.Carlson at Sun.COM>
- Prashant Baliga <prabalig at in.ibm.com>

Sponsors

Champion

- Bill Stoddard(stoddard at apache.org)

Nominated Mentors

- Bill Stoddard
- Filip Hanik
- Craig Russell
- Kevan Miller (non-binding)

Sponsoring Entity

- Incubator PMC

References

[1] <http://www.dmtf.org/about/policies>

[2] http://www.dmtf.org/standards/published_documents/DSP4004.pdf

[3] <http://www.dmtf.org/about/policies/patent-10-18-01.pdf>