# Semantic Web Policy Systems

Matthew Dunlop

# Semantic Web Policy Systems

- Using Semantic Web Technologies for Policy Management on the Web

**Using Semantic Web Technologies for Policy Management on the Web***

**Lalana Kagal** and **Tim Berners-Lee** and **Dan Connolly** and **Daniel Weitzner**
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Lab
Cambridge, MA 02139
{lkagal, timbl, connolly, djweitzner}@csail.mit.edu

**Abstract**

With policy management becoming popular as a means of providing flexible Web security, the number of policy languages being proposed for the Web is constantly increasing. We recognize the importance of policies for securing the Web and believe that the future will only bring more policy languages. We do not, however, believe that users should be forced to conform the description of their policy relationships to a single standard policy language. Instead there should be a way of encompassing different policy languages and supporting heterogeneous policy systems. As a step in this direction, we propose Rein, a policy framework grounded in Semantic Web technologies, which leverages the distributed nature and linkability of the Web to provide Web-based policy management. Rein provides ontologies for describing policy domains in a decentralized manner and provides an engine for reasoning over these descriptions, both of which can be used to develop domain and policy language specific security systems. We describe the Rein policy framework and discuss how a Rein policy management systems can be developed for access control in an online photo sharing application.

**Introduction**

The Web is one of the most important ways for disseminating information across global boundaries. Though it is a simple and convenient framework for searching and retrieving information, the Web suffers from the lack of easy-to-use and adaptable security required by website administrators, application developers, and people in charge of web content. Several approaches for access control to Web resources have been proposed such as WS-Policy (IBM *et al.* 2006), PeerTrust (Gavriloaie *et al.* 2004), Rei (Kagal, Finin, & Joshi 2003), and XACML (Lockhart, Parducci, & Anderson 2005). Each approach has its own policy language that can be used to develop policies over shared ontologies. This causes not only an interoperability issue between domains that use different languages but also forces users to conform their description of their policy relationships to the system's policy language. Instead of requiring all users to adopt a single policy language for their policy requirements, we instead leverage the power of the Semantic Web to reason across the various languages (such as RDF-S (Brickley & Guha 2004), OWL (Bechhofer *et al.* 2004), and rule languages) that are used to describe policies. Rein is a unifying framework that will help the Web preserve maximum expressiveness for policy communities by allowing users to define policies in their own languages but still use the same mechanisms for deploying policy domains.

Rein is a Web-based policy management framework, which exploits the inherently decentralized and open nature of the Web by allowing policies, meta-policies, and policy languages to be combined, extended, and otherwise handled in the same scalable, modular manner as are any Web resources. Resources, their policies and meta-policies, the policy languages used, and their relationships together form *Rein policy networks*. Rein allows entities in these policy networks to be located on local or remote Web servers as long as they are accessible via Hypertext Transfer Protocol (HTTP). It also allows these entities to be defined in terms of one other using their Uniform Resource Identifiers (URI) (Berners-Lee, Fielding, & Masinter 2005). Rein policy networks are described using Rein ontologies and these distributed but linked descriptions are collected off the Web by the Rein engine and are reasoned over to provide policy decisions.

Rein **does not propose a single policy language** for describing policies. It allows every user to potentially have her own policy language or re-use an existing language and if required, a meta policy. Rein provides ontologies for describing Rein policy networks and provides mechanisms for reasoning over them. The ontologies and reasoning mechanisms work with any policy language and domain knowledge defined in RDF-S, OWL, or supported rule languages.

Though authentication is an important part of access control, Rein does not enforce a particular kind of authentication but leaves it up to the individual policy to describe the authentication it requires, if any. This allows domains to either combine authentication and authorization into their access control policies or decouple them and provide the authentication information (such as statements in Security Assertion Markup Language (SAML) (Mishra *et al.* 2006)) as input to the authorization policies. We have developed examples that combine authentication and authorization and rely on simple cryptography techniques and other examples that use

*This work is sponsored by the National Science Foundation Awards 0427275 and 0524481.

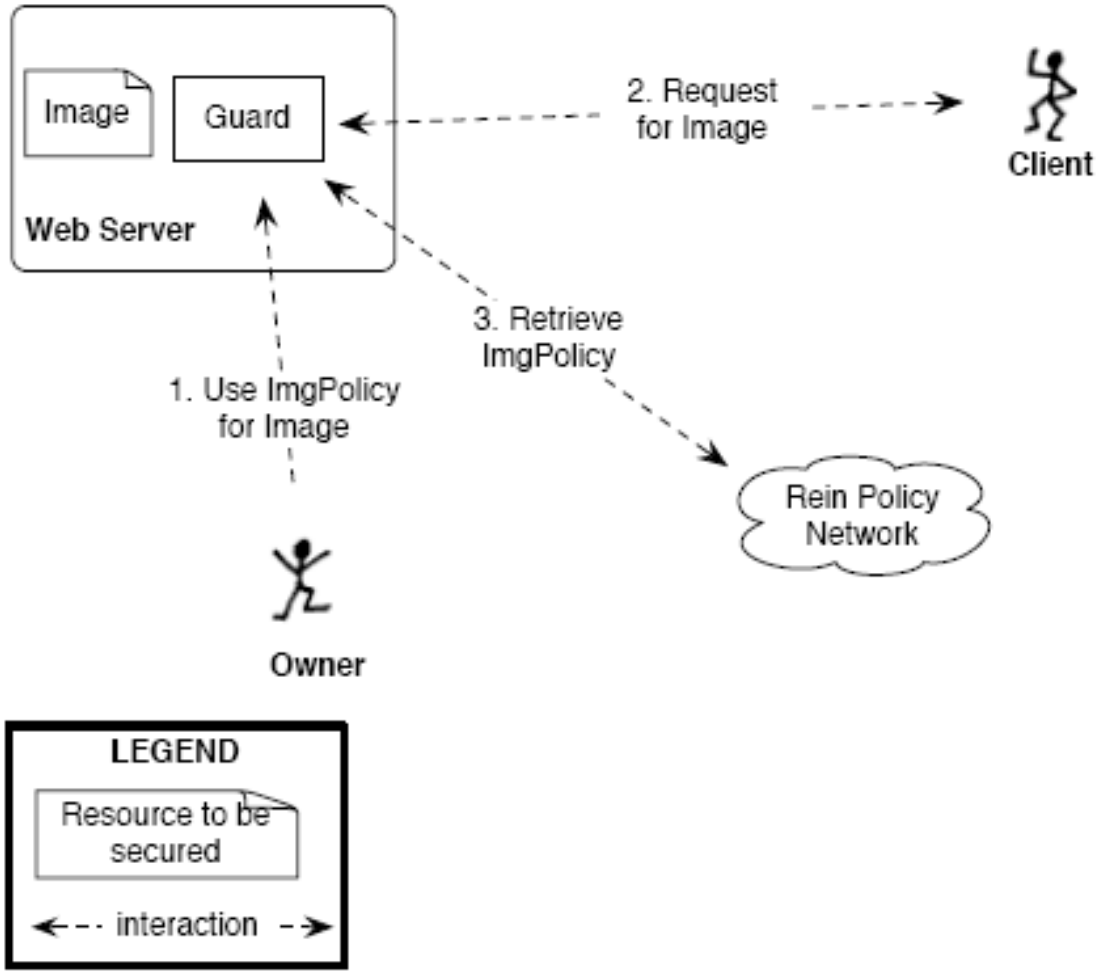
# Purpose

- Develop a policy framework (Rein) that leverages the semantic web
- Allow users to define policies in their own language
- Provide mechanisms for reasoning over any supported rule languages

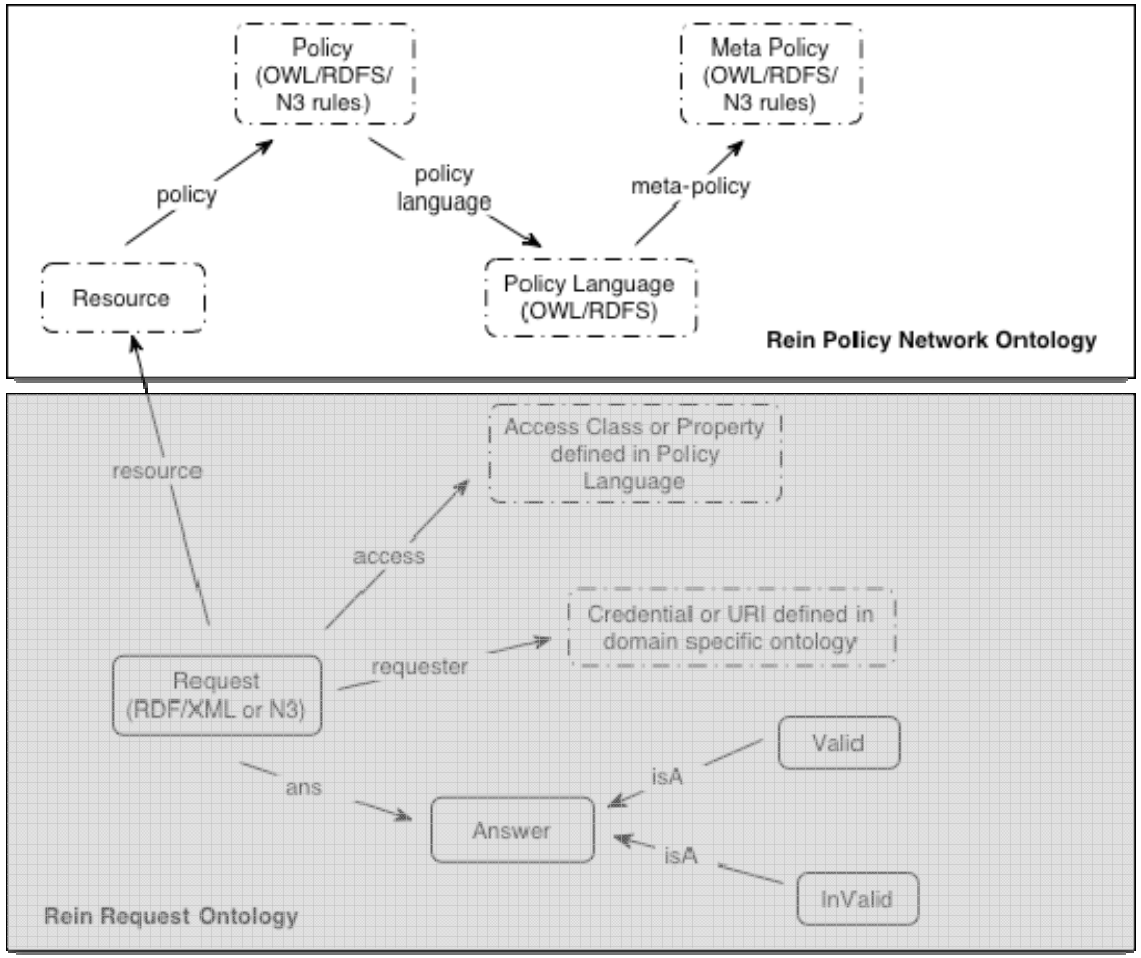
# Contributions

- Web-based approach for representing and reasoning over policies for web resources
- Flexible sophistication or expressiveness of policies
- Provides unified mechanism for reasoning
- Supports compartmentalized policy development
- Self-describing framework

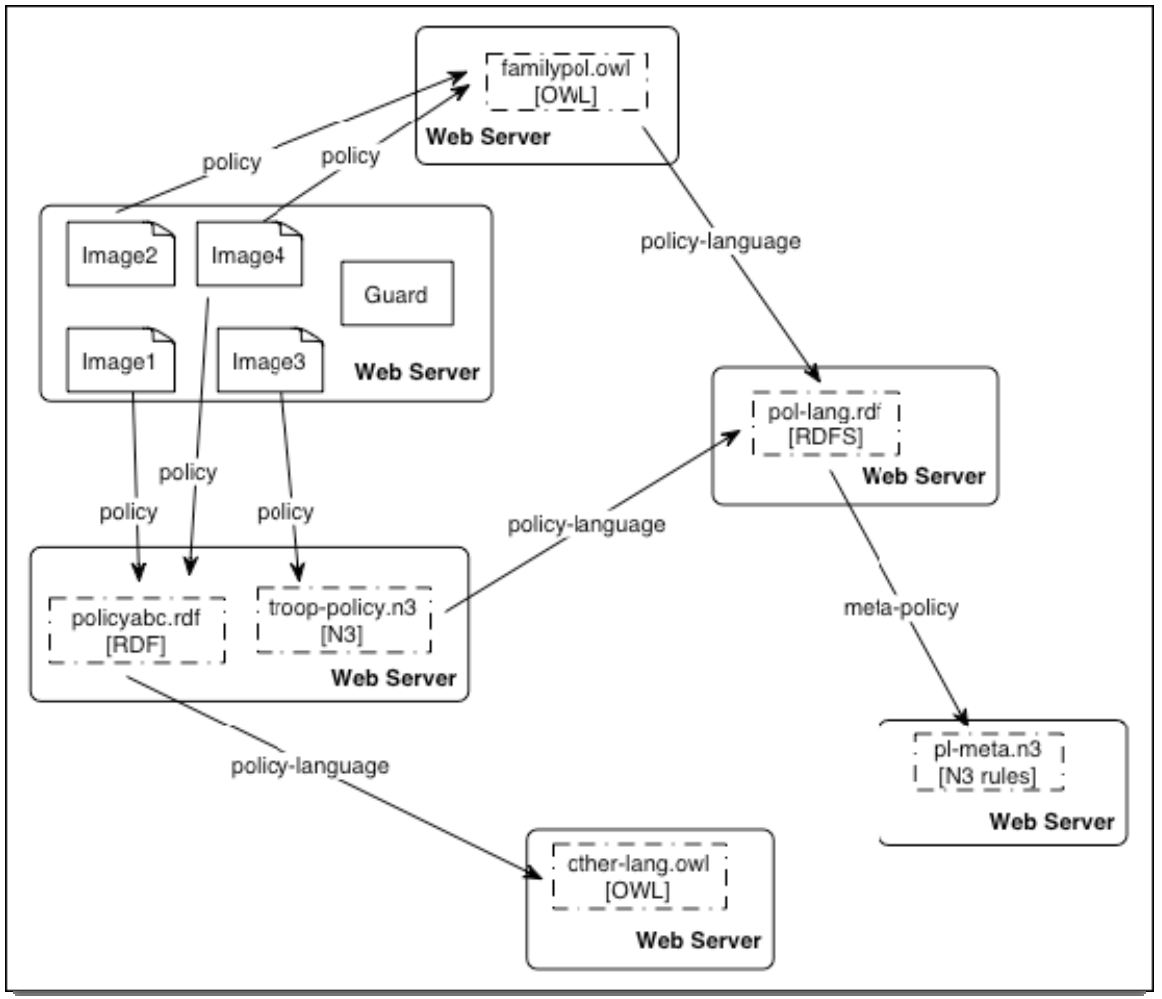
# Access control model



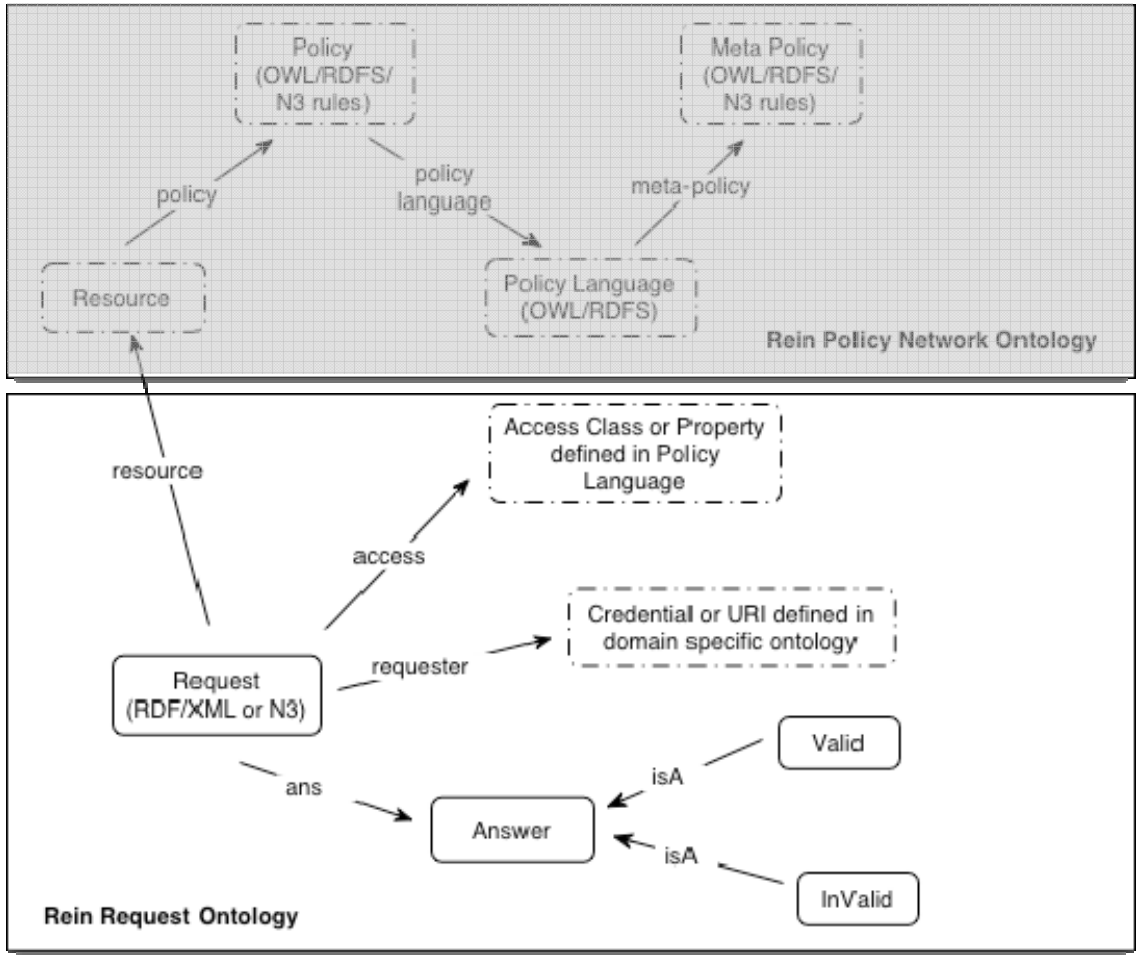
# Rein Ontology



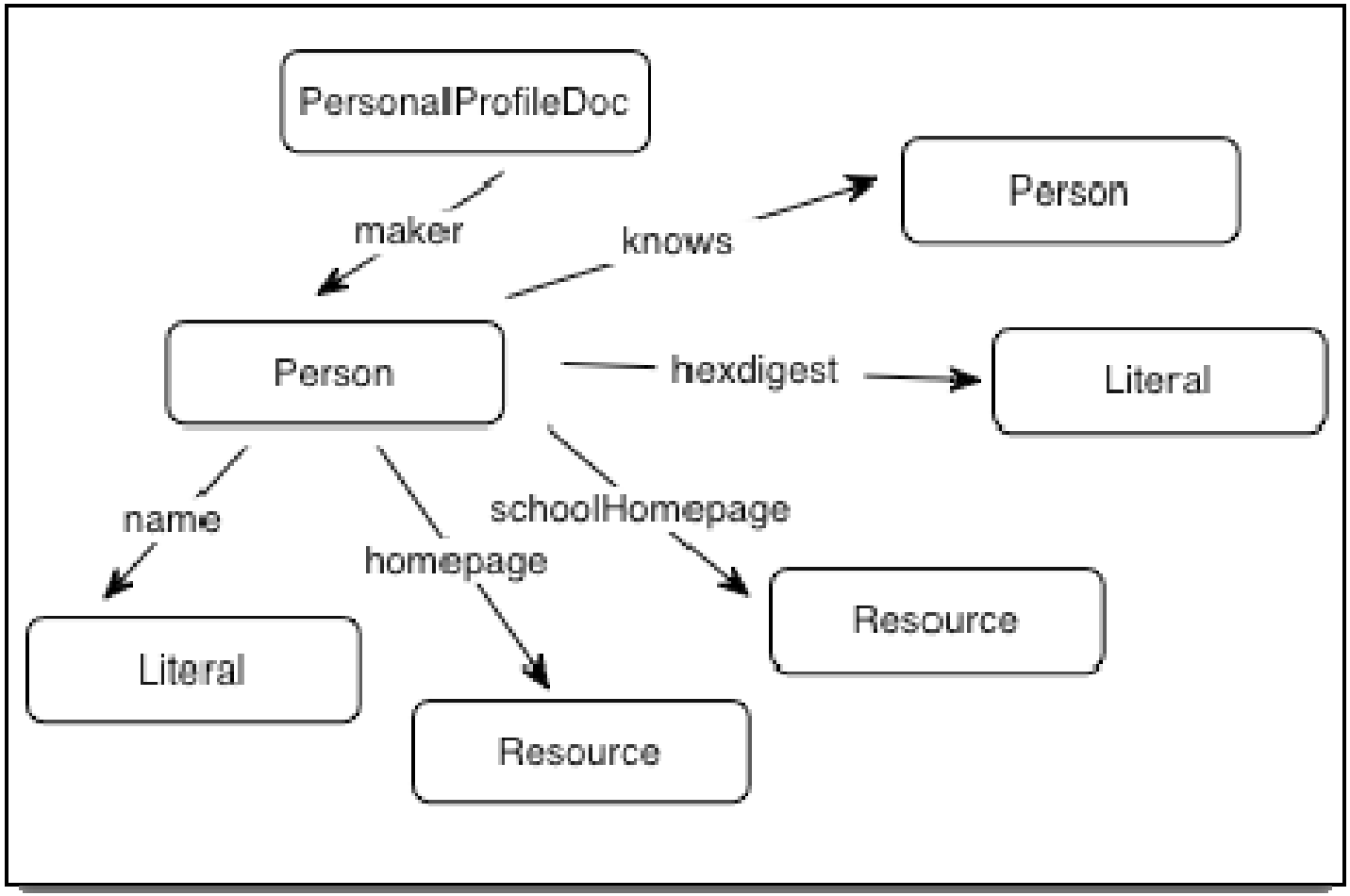
# Example Rein Policy Network



# Rein Ontology



# Sample Ontology



# Reasoning Engine

- Accepts requests for resources
- Collects relevant information
- Answers questions about access rights

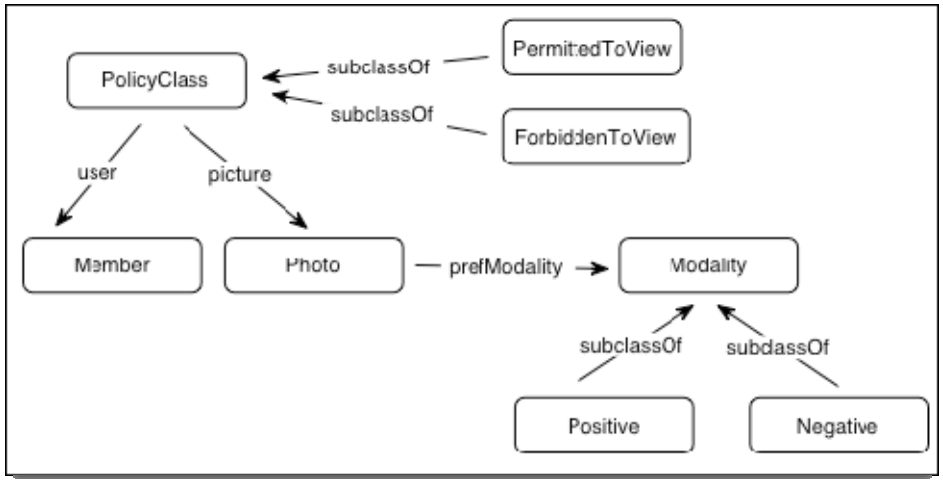
# Using Rein

- Rein used by guard
- Rein used by client
- Hybrid approach

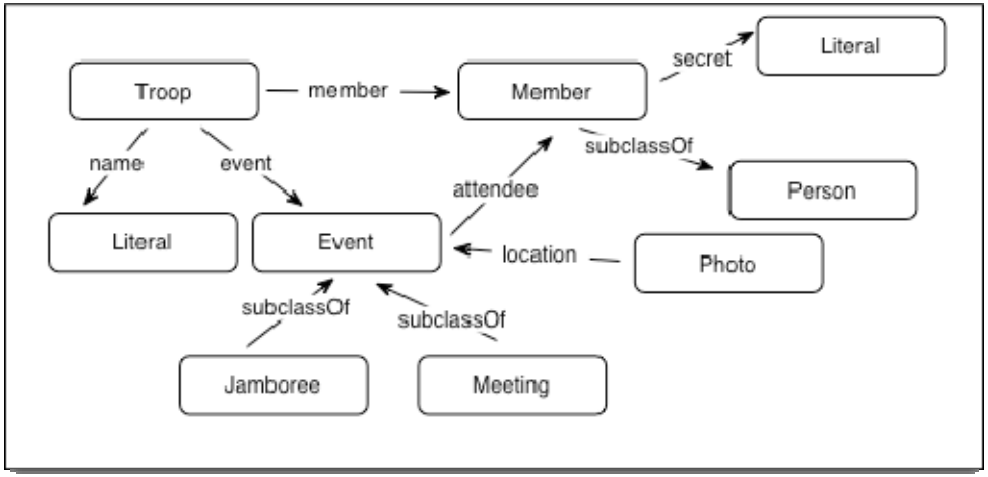
# Implementation Requirements

- Reasoners for RDF-S
- Engine for supported rule language(s)
- Programming language capable of:
  - Accessing web
  - Working with chosen reasoners and engines

# Photo Sharing Example

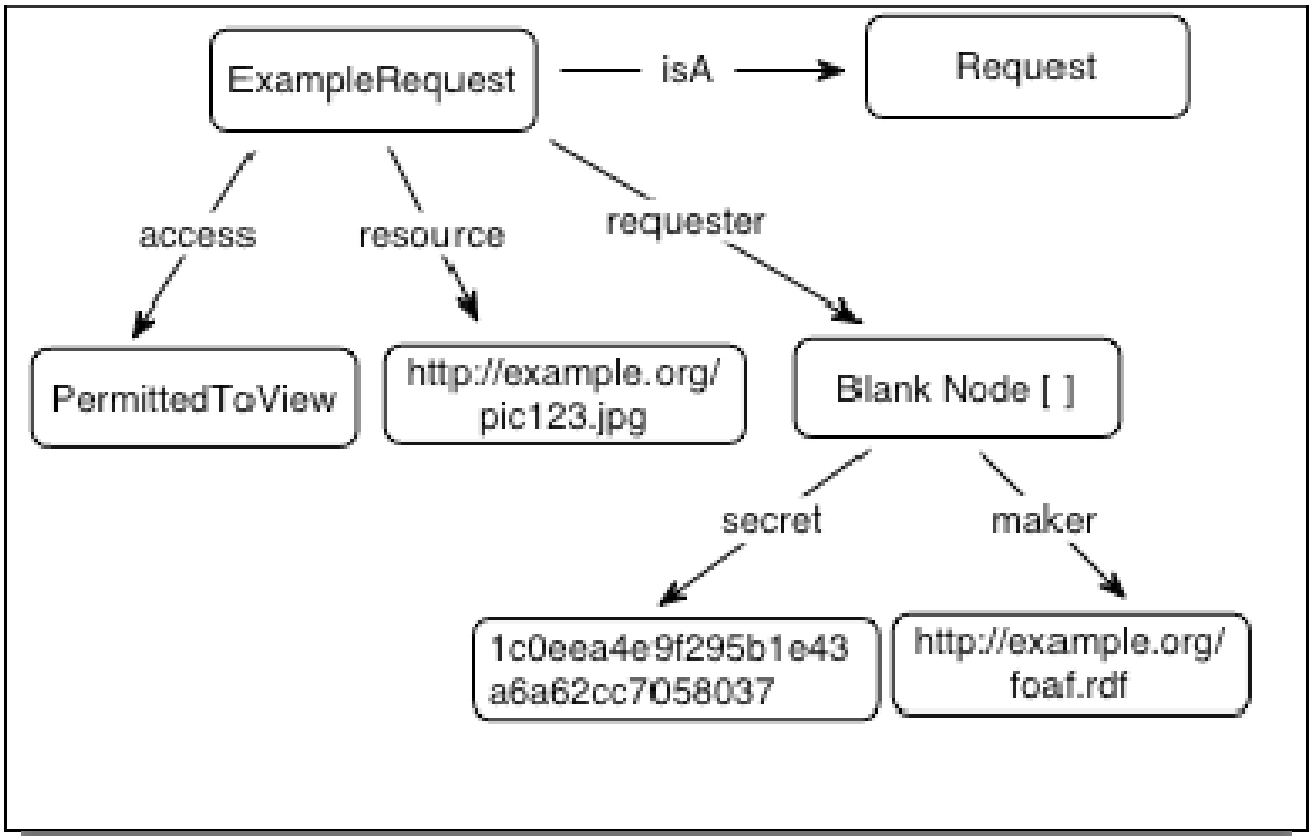



Policy Language

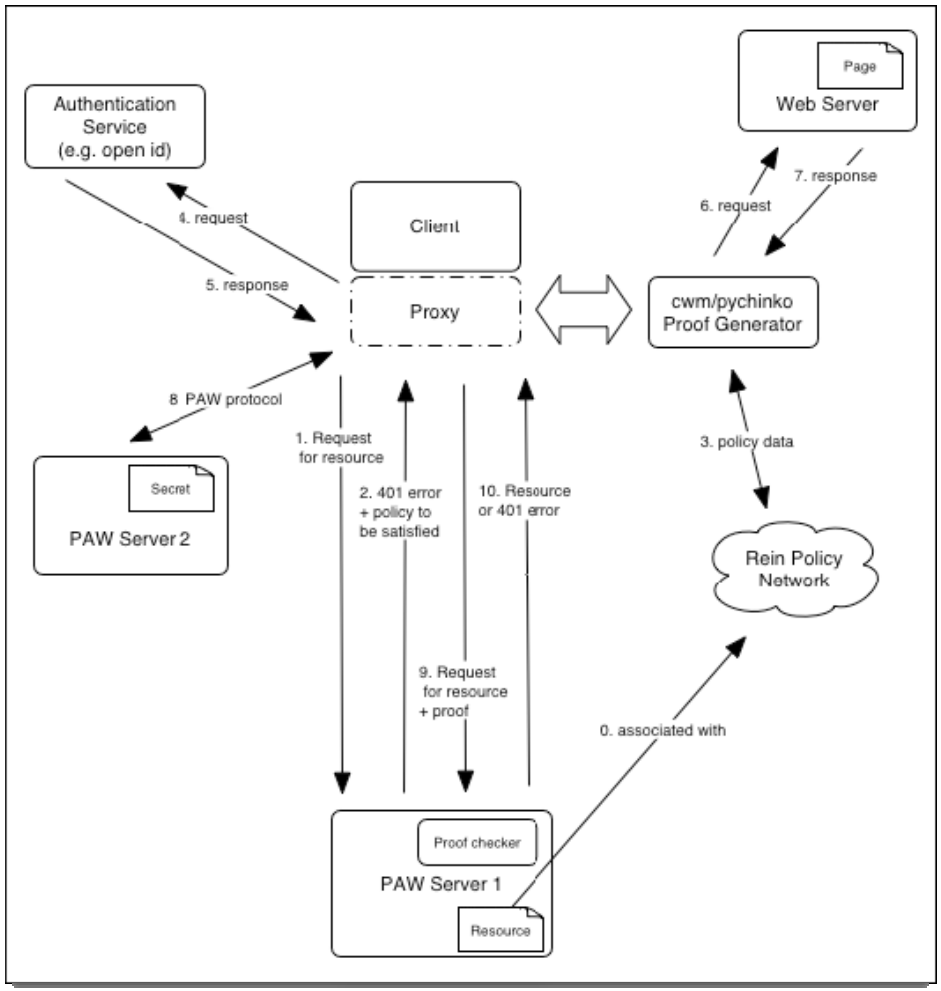



Troop Ontology

# Sample Request



# Rein implemented in Policy Aware Web



# Discussion

- There is no discussion of time to process a request. The overhead for processing a complex request could be non-negligible.
- There are no security mechanisms in place. It seems pretty easy to perform a DOS attack, yet difficult to prevent it.
- Rein does not allow querying of what resources a requester has access to, only whether a requester can access a resource.