

Context-Aware Management Domains

Ricardo Neisse¹, Maarten Wegdam, Patrícia Dockhorn Costa, and Marten van Sinderen

CTIT, University of Twente, The Netherlands
{r.neisse, m.wegdam, p.dockhorncosta, m.j.vansinderen}@utwente.nl

Abstract. In this paper we extend the concept of management domains to a new concept called Context-Aware Management Domains (CAMDs). CAMDs enable context-aware management of policies allowing the grouping of entities based on context information. Since context is dynamic, so is the domain membership. As a consequence, the association of policies with the entities in the domain also becomes dynamic. In this paper we provide CAMD examples and an information model together with a discussing on our ongoing implementation for our target context-aware service platform.

1 Introduction

Context aware services adapt themselves to the current user's situation. An example of this is a tourist service which uses the current user location, activity, and preferences to personalize tourist advices. In order to support context awareness, service platforms have been designed to support context information acquisition, reasoning and distribution [1].

Typical context-aware service platforms have thousands or millions of entities (users, service providers, context providers, etc.) and different types of policies have to be managed. Policies are required, for instance, to control access to context information, to enforce user's privacy, and to manage trust relationships among the entities. Due to the complexity, dynamicity, and large number of entities, the specification of these policies can easily become unmanageable.

Standard policy management tools ease the policy management, however, the problem with these tools is that they provide either static management capabilities (e.g. management domains [2]), or, if there is some form of dynamic management, this is limited to one specific area (e.g., X-RBAC [3], [4], and [5] for access control and COMITY [6] for trust management). For this reason, these policy management tools do not fulfill the dynamic policy requirements of context-aware service platforms.

In a context-aware service platform, policies are defined based on the context of the entities. One example is a privacy policy stating that "Bob's identity should not be anonymized for nearby persons". In this case, "nearby persons" refers to a set of

¹ Supported by CNPq Scholarship – Brazil

entities not known at policy specification time, because it is not possible to determine beforehand which entities are likely to approach Bob.

In this paper we address context-aware policy management by extending management domains to a new concept called Context-Aware Management Domains (CAMDs). CAMDs are management abstractions that provide dynamic grouping of entities based on common context situations and, as a result, context-aware management of different types of policies. We provide an information model for CAMDs and discuss an implementation strategy for CAMDs in the scope of our target context-aware service platform [1].

This paper is organized as follows. Section 2 introduces our context-aware service platform and describes the policy deployment scenario with examples. Section 3 presents our new concept called Context-Aware Management Domains, the information model, and our ongoing implementation efforts. Section 4 compares our work with related work on context-aware management tools and Section 5 ends this paper with conclusions and future work.

2 Policy Management in a Context-Aware Service Platform

Figure 1 presents our target context-aware service platform considering a single administrative domain and illustrates the main roles we distinguish regarding the platform *management* and *operation* layers.

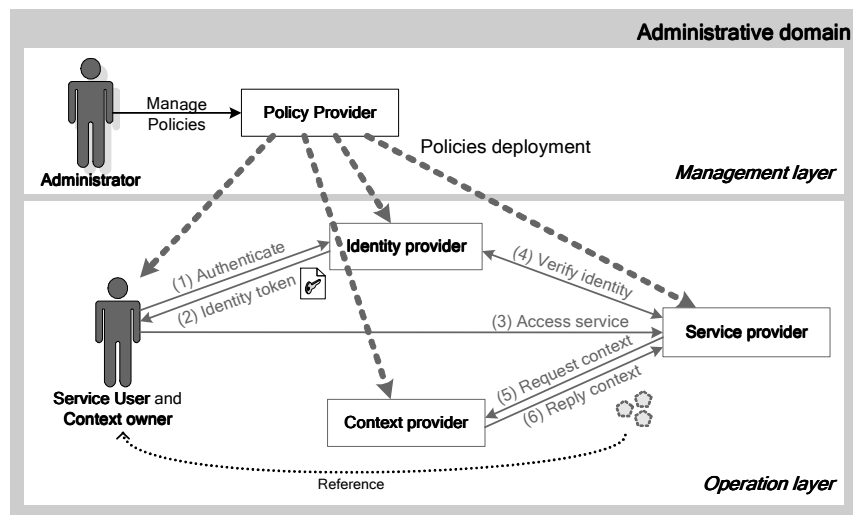


Fig. 1. Policy deployment in a context-aware service platform

Within the *operation layer* a user receives an authentication token (2), after authenticating with an identity provider (1), which is used to access a service provider (3). The service provider verifies the user's identity (4) and retrieves context information to adapt the service (5 and 6). This information can be, for instance, the

current activity or location of the user; however, it can also include context information about other entities (context owners) that are relevant for the context-aware service used (e.g. service provider). For details about the operation layer see [1].

Within the management layer an administrator accesses the policy provider in order to manage operation policies. Policies are rules that define a choice in the behavior of the system and can be of different types such as obligation, authorization, refraining, filtering, delegation, and meta-policies [7]. Policies of different types can also focus on different management areas, for example, access control, privacy enforcement and trust management. In this paper we support context-aware management of policies of different types and different areas, however, due to space limitations, we only exemplify obligation policies focusing on privacy enforcement.

Obligation privacy policies describe actions that subjects must perform on target entities under certain conditions [7]. Such a policy could state, for instance, that “15 minutes after getting Bob’s location, Alice (Bob’s colleague) should delete it” or “Bob’s identity should be anonymized by the identity provider when provided to Alice”. In the examples above the policy subjects and targets (Bob, Alice, and Bob’s identity provider) are individually specified in the policies and do not easily allow Bob to specify policies for a set of entities, for instance, all his colleagues.

In order to allow the deployment of policies for a collection of entities, as opposed to individual entities, management domains [2] can be used as a grouping abstraction (e.g. Bob’s colleagues). Management domains reduce the management complexity in large systems because it is hard to specify and apply policies individually for each entity on a large scale. However, one problem with management domains is that they are static, and the inclusion and removal of entities from a domain must be done manually. In this paper we go one step further by defining management domains based on context situations [8] in a new concept called Context-Aware Management Domains (CAMDs).

3 Context-Aware Management Domains

In order to illustrate our new concept we present in Figure 2 an example where the Context-Aware Management Domain (CAMD) “colleagues currently at work” is mapped to every colleague in which the current activity status is “at work”. When persons change their activities domain membership also changes. As a result, any association of domain policies with entities also changes, as entities can leave/enter the domain dynamically according to changes in the context information.

