Centralized Authorization and SOA: Defy the Laws of Tradition

Lori MacVittie, 2008-07-04

I was listening to some ***Primus*** yesterday - To Defy The Laws of Tradition, to be precise- and it got me thinking about architectures and decisions that *defy the laws of (IT) tradition.*

One IT tradition that seems extremely difficult to overcome is that applications should authorize users. After all, the application should control, based on some kind of policy, what users can and cannot do while interacting with it. In fact it's almost a law within IT that while applications may accept the authentication of a user from a trusted source, it is still the authoritative source for authorizing access to specific *functions* or *actions* performed within that application.

The very essence of SOA defies that law, and yet it seems difficult for architects to *go against tradition* even when faced with an architectural model like SOA that *begs* for tradition to be thrown out the door.

Traditional applications need to make authorization-based decisions because of the way traditional applications keep all their logic hidden under the covers. Application flow between two disparate functions occurs within the code and there's really no good way - or reason - to ask an external source to authorize the user to execute the next piece of logic. It doesn't make sense from an architectural viewpoint, and from a performance standpoint it makes even less sense.

But a SOA application comprises disparate and hopefully distributed pieces of logic. The flow between functions that, in a traditional application, is entirely hidden beneath the hood is now exposed, with each service comprising a single unit of business logic. That piece of logic, *that service*, should not make decisions on authorization. Indeed, in a well designed SOA *the service cannot make that decision* because it does not have enough context on which to base its decision. In a SOA authorization is still often based on a combination of the application and the user or user's role, and that information is something the service does not have.

But let's assume it did. Perhaps the service can extrapolate context or retrieve it from the request. The authorization to execute that service is now hard-coded into the service and duplicated - once for each application that might make use of the service. And when a new application makes use of that service the service must be updated to include a way to authorize users for *that* application, assuming of course that the service can extract context from every request and for every application in which it will be included.

That's the traditional way to handle authorization and it's time we defy that tradition and move to a centralized authorization model for SOA; that is, we should use an external, policy-based security access solution to provide authorization of services based on the application, the contextual data in the request, and the service being invoked. This decouples authorization from the service and restores the **reusability** of that service without compromising security. It enables **flexibility** (i.e. **agility**) in the architecture because you can modify the authorization model without affecting services, and the addition of a new application requires integration or configuration of a new policy for the centralized authorization mechanism rather than modification of already deployed and tested services.

A centralized authorization architecture also improves the ability of the organization to audit access because all authorization events can be logged and monitored centrally rather than pulled from multiple sources which can cause headaches for those attempting to correlate events or track down problems.

Consider carefully how you currently authorize access to applications and how that architect fits with a SOA and servcies. You may find that your solution is a traditional one, and that you're following that tradition simply because that's the way it's always been done.

Imbibing: Mountain Dew

Technorati tags: MacVittie, F5, SOA, security, service oriented application delivery, authorization

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com