

# How does SOA service discovery (UDDI) work in practice?

Asked 16 years, 3 months ago   Modified 15 years, 7 months ago

Viewed 6k times



8

I'm just reading up on SOA and the service registry / UDDI get mentioned regularly. It sounds nice but how is used in reality?



- Is the registry meant to decouple a logical service from its' physical implementation (port, url etc)?
- Is the registry meant to be browsed by a human looking for an interesting service to play with?
- Would it be 'wrong' to hard-wire an application to the services it uses?



architecture

soa

uddi

Share

Improve this question

Follow

asked Sep 24, 2008 at 11:59



paul

13.5k ● 24 ● 84 ● 154

3 Answers

Sorted by:

Highest score (default)





10

I find it to be more theoretically useful than practically useful. It is infrequently implemented and infrequently used. In reality, DNS provides a sufficient abstraction tool for the location of resources on the network.

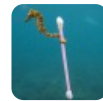


Share Improve this answer

answered Sep 24, 2008 at 16:50



Follow



**MattMcKnight**

8,260 ● 31 ● 35



4

A service registry stores and publishes information about all available services, mainly their interface description and their current URI (ip, port, whatever). This way the application can simply ask the registry for the needed service and will get the details of a fitting service implementation, and can connect.



UDDI is not the only way to get a registry for you services. But remember that UDDI is intended for webservices only, so it's only usefull if your SOA consists only of webservices.

1) Correct.

2) No, it's not really meant for human eyes. Sure, there are tools to browse the directory, but they are mainly for looking if the registry got the services you need etc. The real usage happens directly between your application/service and the registry.

3) That depends on what you want to accomplish. If you want to build a SOA it think it would be 'wrong' because this contradicts the loose coupling paradigm of SOA. If this is your only service, the only application that uses it and it's likely that the service won't change it's URI there's definitely no problem in hard-wiring it - but then there's propably no need to separate this service :)

Share Improve this answer

edited Sep 24, 2008 at 16:47

Follow

answered Sep 24, 2008 at 12:27



Fabian

811 ● 2 ● 9 ● 17



1



how about using multicast to discover the service? Like using jgroups or SLP? All the services will discover each other and inject the one they need into a proxies. Then building abstraction over the actual transport implementation. (e.g. rest, soap, rmi)



Share Improve this answer

answered May 4, 2009 at 9:10



Follow



michael