

Re: Just tell me why ? so confuse...

Re: Just tell me why ? so confuse...

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet.webservices/2005-11/msg000>

- *From:* "Peter Kelcey" <Peter.Kelcey@xxxxxxxxx>
 - *Date:* 11 Nov 2005 12:07:38 -0800
-

Scott,

I agree completely that vendor independent services are preferred solutions in large enterprises that are looking to integrate multiple applications and systems. I would also agree that a quality architecture in that case would have a presentation layer communicating with a service façade that calls various business objects. (Just as you suggested). In these cases, we are willing to live with the performance hit caused by web services because we are gaining so much more towards our overall enterprise architecture.

This is where I potentially deviate from the architecture you suggested. It's my opinion that an application should only have a services layer if it is going to be integrated with one or more other systems or you know that you may need to change vendors in the future. In your post you stated that you had these requirements in your projects.

Without these requirements I don't believe that a self contained application should put a services layer between its business components and its presentation layer. I also think that service oriented architectures are often incorrectly incorporated within application architecture and that a lot of times, service layers only end up creating a fatter and less responsive solution for no reason. When looking at a system that doesn't have those requirements, web services have as many shortfalls as they do benefits. I think that the loss of any real transaction mechanism, the lack of binary communications, the lack of any true version control and the increased performance overhead are significant enough reasons to rule out web services as a communication mechanism within the internals of a single application. Therefore, I'd take something like .NET Remoting, RMI etc in order to get the additional benefits. It's in a case such as this where I meant that a proprietary solution is preferred and recommended.

Now I've have worked on solutions that implement a hybrid of these 2 architectures. A presentation layer uses .NET remoting to connect to business facades. We also had a small number of web services that

Re: Just tell me why ? so confuse...

Re: Just tell me why ? so confuse...

exposed a subset of those facades to other applications that needed to access our data. We get our performance boost, our transaction control and we get our system interoperability.

It always comes down to it is you need to accomplish and what your requirements are.

Based on what Serge says he's trying to do. I'd recommend the same thing you did.

IMHO as well.

Peter

- **Follow-Ups:**

- ◆ **Re: Just tell me why ? so confuse...**
◇ From: Scott M.

- **References:**

- ◆ **Re: Just tell me why ? so confuse...**
◇ From: Scott M.
- ◆ **Re: Just tell me why ? so confuse...**
◇ From: serge calderara
- ◆ **Re: Just tell me why ? so confuse...**
◇ From: Peter Kelcey
- ◆ **Re: Just tell me why ? so confuse...**
◇ From: Scott M.

- Prev by Date: **Re: Just tell me why ? so confuse...**
- Next by Date: **Re: Just tell me why ? so confuse...**
- Previous by thread: **Re: Just tell me why ? so confuse...**
- Next by thread: **Re: Just tell me why ? so confuse...**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**