

# Re: What am I missing?

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2007-09/msg02275.html>

---

- *From:* BJ <[julie6232000@xxxxxxxxxx](mailto:julie6232000@xxxxxxxxxx)>
  - *Date:* Sun, 23 Sep 2007 23:47:26 -0700
- 

On Sep 23, 1:38 am, "Just Me" <[news.microsoft.com](mailto:news.microsoft.com)> wrote:

Well I don't think so really.

Which question?

I personally think it depends how complex your middle layer is likely to be. For example, if you were dealing with a business which had hundreds or protocols regarding how it did business then a business layer is definitely worth doing but if we are only talking about a few then why bother one could argue.

I just started looking at the existing application, but it seems that the middle tier is making calls to the data layer and passing it off to the client. And yes I do agree that if you have dissimilar systems trying to access the same data or a massive audit process that interfaces with multiple systems (IBM 3270 Mainframe, Solaris, Win NT) then a middle tier does make sense.

The purists will advocate multi tier and will even argue that the tiered must be firewalled using web services to prevent such terrible erroneous measures being carried out but errant developers. But that's a purist for you. The advantage of taking such an approach is that you then have a structured way to evolve your middle layer.

I can't do things because it's the way a purists says it has to be done a certain way. We'd all have dumb terminals and writing programs in COBOL or Assembler.

For me, its horses for courses, if it looks like I need to handle a complex situation, then I'd better have a very formal method of doing so, if its simple, why complicate it. And if it's simple, abut has been designed in a

## Re: What am I missing?

complex way, then is it easier to add to the existing complexity or should I simplify it, and in doing so where is the business case for it. ?

I think what you are saying is the middle tier simplifies a complex design? Please correct me if I understood that wrong.

The application is used in a very specific way, show me an image from my group of images. Currently, the client application makes a call to the service, which in turn utilizes the imaging API's to retrieve the image. There are some direct calls to the database by the middle tier. The middle tier calls are specific to that application. I truly do appreciate your response and I'm not arguing any fact other than by keeping the middle tier what do I gain? By losing the middle tier, what do I lose?

I guess I was hoping for a smack up side my head to say, "hey its 35% faster to use a middle tier" or "its more secure" or "it's the only way to support 100, 200, 300, 400+ users". I'm pretty sure invoking a middle tier is not faster. Instead of going from point A to Z, the process is going from A to M to Z utilizing a protocol with a lot of overhead (SOAP). One of the developers told me SOAP was the new way to invoke services and CORBA and RMI was the old school way. Well, I can say with certainty that CORBA and RMI are older than SOAP, but unless the number have changed over the past 7 years the evidence says SOAP is the slowest of those technologies. I'll use it but don't tell me CORBA and RMI have gone the way of the dinos.

As far as security, I don't know. Is having a middle tier more secure? If I make it a web form application or a client win form application, it will sit on the network behind the corporate firewall. Today security is maintained utilizing a security table and LDAP. They are slowly moving over to AD.

As far as being able to support a large group of users, I don't know the answer to that one. I do know that there is only one BEA domain and one Database for this application. If some other application utilizes access to that Database by way of the middle tier, then leave it up for them. I did over hear the supervisor say they were looking for a more supportable solution. Today, if a change has to happen in the middle tier there is one developer for all three BEA domains that writes those services. It took us two full days to find a problem in the VB6 code because we could figure out a Tuxedo error 12.

Again thank you for the reply, but so far for this application, I really don't see why I would need to maintain a middle tier.

.