

## Re: middle tier recommendations

**Source:** <http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2004-11/1639.html>

---

**From:** Ben Strackany (*infoNOSPAM\_at\_developmentnow.nospam.com*)

**Date:** 11/26/04

Date: Fri, 26 Nov 2004 12:48:14 -0600

Nick -

Along the lines of what Param said -- remoting & web services would be useful if your middle tier were on its own machine, but IMO the MT should only be its own box if you are doing some serious CPU-intensive tasks in your middle tier (which I am assuming you are not) or if you have some other reason.

So IMO your middle tier should be .NET assemblies/DLLs & be on the web server (i.e. the same machine as the ASP.NET pages), referenced & called from ASP.NET just like other assemblies. No web services, no soap, no remoting. That keeps it simpler, faster, more reliable.

We did the whole 3 physical tier thing years ago & it was not worth it. :)

--

Ben Strackany

[www.developmentnow.com](http://www.developmentnow.com)

<http://www.developmentnow.com>>dn</a>

"Param R." <pr@nospam.com> wrote in message

news:eI363t80EHA.2624@TK2MSFTNGP11.phx.gbl...

> Nick, to answer your questions:-

>

> 1. 2 Web Servers & 1 Database

> 2. Starting out 2000+ users.

> 3. Continuous business use M-S 8-7.

> 4. 99.99% (8 a.m. - 10 p.m.)

> 5. During peak times we could see a sudden spike in usage.

> 6. YES. Realtime modification is a necessity.

> 7. Limited communication with existing systems - in realtime.

>

> It is all about speed for this project. If the web pages are slow and the

> apps take time to execute then the business starts losing \$\$...

>

> thanks!

>

> "Nick Malik" <nickmalik@hotmail.nospam.com> wrote in message

> news:FKqpd.565167\$mD.516047@attbi\_s02...

> > What is driving the architecture? What are the key constraints?

> >

> > Honestly, most applications are fine with an ASP.NET layer that calls a

> > simple middle layer, written as DLLs, that call SQL. That said, most

> > applications have fewer than 100 concurrent users. I'm going to venture a

>

Re: middle tier recommendations

## microsoft.public.dotnet.framework: Re: middle tier recommendations

> > guess that this doesn't apply to you.  
> >  
> > How many servers have you set aside for this application?  
> > How many users do you plan to serve with this application?  
> > What is the nature of their use (continuous use for business day,  
> > occasional  
> > light use, occasional heavy use, receiving a stream of information)?  
> >  
> > In addition, you didn't provide the key constraints that drives the  
> > architecture.  
> > Do you have high uptime requirements (99.9% or better)?  
> > Do you have variable scalability issues (sudden spikes that increase  
> > traffic  
> > by an order of magnitude or more for a sustained period)?  
> > Do you need to be able to modify the behavior of the system while it is  
> > running due to the nature of competition in your business?  
> > Do you have existing systems that you need to communicate with? If so,  
> > are  
> > these systems designed for real-time communication or do you need to  
batch  
> > things up?  
> >  
> > Without at least a little of this information, my answer would be too  
> > vague  
> > to be useful.  
> >  
> > As for .NET remoting, it is a useful mechanism for designs that need to  
> > partition the execution of the application onto multiple servers. The  
> > marshalling is far more efficient than with web services, but it is  
still  
> > marshalling... and if you are sending data sets across a marshalling  
> > boundary, you are probably not designing your interfaces correctly.  
> >  
> > An excellent book: Advanced .NET Remoting by Ingo Rammer.  
> >  
> > HTH,  
> > ---- Nick  
> >  
> > "Param R." <pr@nospam.com> wrote in message  
> > news:%23kiwNQj0EHA.804@TK2MSFTNGP12.phx.gbl...  
> >> Hi all, we are in the process of architecting a new application that  
will  
> >> have an asp.net front end & sql back end. In the past we have used  
> >> webservices as a middle tier solution but in terms of performance it  
has  
> > not  
> >> been upto the mark. Besides with the latest .net version there are some  
> >> known issues with calling web services (keepalives etc.). What other  
> > middle  
> >> tier solution does .net have to offer? What is .net remoting and how  
does  
> > it  
> >> work? Is it similar to DCOM? Does it have to run under IIS? I would  
like  
> >> something that is not dependent on IIS preferably. In our solution we  
> >> will  
> >> be passing custom objects back and forth with the middle tier  
interacting  
> >> with the database.  
> >>  
> >> Any help and guidance here is much appreciated!  
> >>

microsoft.public.dotnet.framework: Re: middle tier recommendations

> >> thanks!  
> >>  
> >>  
> >  
> >  
>  
>