How to route AMF requests though one trusted server

Asked 15 years, 10 months ago Modified 15 years, 4 months ago

Viewed 967 times

0

We are planning to develop a Flex application that must make AMF calls to retrieve data from a secure server. However, we are not allowed to connect directly to that secure server with AMF. We can only make WSDL/SOAP or .net remoting calls to the secure server. Therefore, **all instances of the flex application running on a client machine must connect to a single trusted server, which would then connect to the secure server.**

These requirements are based on political reasons within the company and are not flexible (isn't that always the case?)

In case the requirements are not clear, here is an example. When you access your checking account via your bank web site the web site downloads a flex/flash application to your web browser. This application shows you your accounts, transactions, balance, etc. However, your actual checking account details are stored on a secure database server at the bank that doesn't accept inbound connections from clients. So, the flex application in your browser makes an AMF call to the banks web site, and it turns around can makes a WSDL/SOAP or .net remoting call to the database server. It gets the data and sends it back to the your browser via AMF.

Is there a "proxy" application we can run on the web server which will accept the AMF calls and turn around and make WSDL/SOAP or .net remoting calls to the secure server? Or do we have to write this by hand?

.net apache-flex remoting amf

Share

Improve this question

Follow

asked Feb 17, 2009 at 22:28

Chris Vesper

648 ● 7 ● 20

2 Answers

Sorted by: Highest score (default)

▲

0

▼

✓

If you happen to be using asp.net in your web server you can use either FlourineFx (open source) or WebOrb (commercial) as your server.

Both options can be hosted in IIS or act as windows services.

You can then just drop your dotnet class libraries for connecting to your secure server within the the bin directories of the servers.

Share Improve this answer

answered Aug 26, 2009 at 4:34

Follow

Nathan Smith
37.5k ● 6 ● 30 ● 25

0

I would make the BlazeDS server the proxy, with a custom java class (using the JavaAdapter). Your java class will then work as a facade against your main server.

The advantage of this setup, the blazeDS proxy server will take care of all the serialization/deserialization of the AMF request and response. And then you have the full power of java to call out to your main servers to use RMI, SOAP, HTTP, etc. This way the main server will not need to know anything about AMF.

Other options; 1. AMF is still passed over as binary data in a basic http requests. So a simple HTTP proxy should forward the request correctly. However, if you do this the final server will need to know how to process the AMF protocal.

2. Instead of AMF, use the mx:HTTPService tag and mark useProxy="true". This will proxy all of the http (json, rest) calls through a blazeDS server.

Share Improve this answer

Follow

answered Feb 18, 2009 at 4:22

Mike Nimer