Chapter 8

Integrating via Data Services

by Shashank Tiwari

So far, you know how to combine Flex and Java using HTTP and web services. The last chapter surveyed a bunch of alternative mechanisms to achieve this. Most of these mechanisms involve loosely coupled text-based data interchange. Most of them interact by pulling data. Only one of them, Hessian, transmits binary data. Only one, Hessian again (with additional infrastructure powered by the new Java IO), allows data push.

Now, we delve into more tightly coupled scenarios and efficient binary data transmission using AMF (Action Message Format). The AMF specification can be accessed online at http://opensource.adobe.com. This chapter looks at both pull- and push-based interactions—using data services and media servers. Adobe offers two alternatives for data services—the commercial LifeCycle Data Services (LCDS) and the open source BlazeDS—and it offers a suite of products for media servers: Flash Media Server (FMS) products. There are a few open source alternatives to these as well.

In this chapter, I will analyze these products in the context of their applicability to rich, engaging enterprise-grade applications. Functionally they can be divided into the following two topics:

- Remoting and RPC
- Messaging and data push

At this point of the book, remoting and RPC should be familiar territory, so let’s start there.

Remoting and RPC

Flex applications can access the Java server side using data services. They can access Java objects and invoke remote methods on them. The Flex framework includes a client-side component called
RemoteObject. This object acts as a proxy for a data service destination on the server. When configured properly, this object handle can be used to invoke RPCs. Before we get into the nitty-gritty of this object and destination configuration, let’s step back and look at the data services architecture.

Data Services Architecture

Figure 8-1 is a pictorial summary of the data services architecture. The view is biased to highlight the functional elements. It includes technical aspects but skips the internal details in many places. As you look deeper into the nuts and bolts in this chapter, many of these details will emerge.

Figure 8-1. Data services architecture: an overview

As Figure 8-1 depicts, data services includes the following:

- Gateway to intercept server-bound calls
- Parser to make sense of AMF messages
- Serializer and deserializer to transform objects between ActionScript 3.0 (AS3) and Java