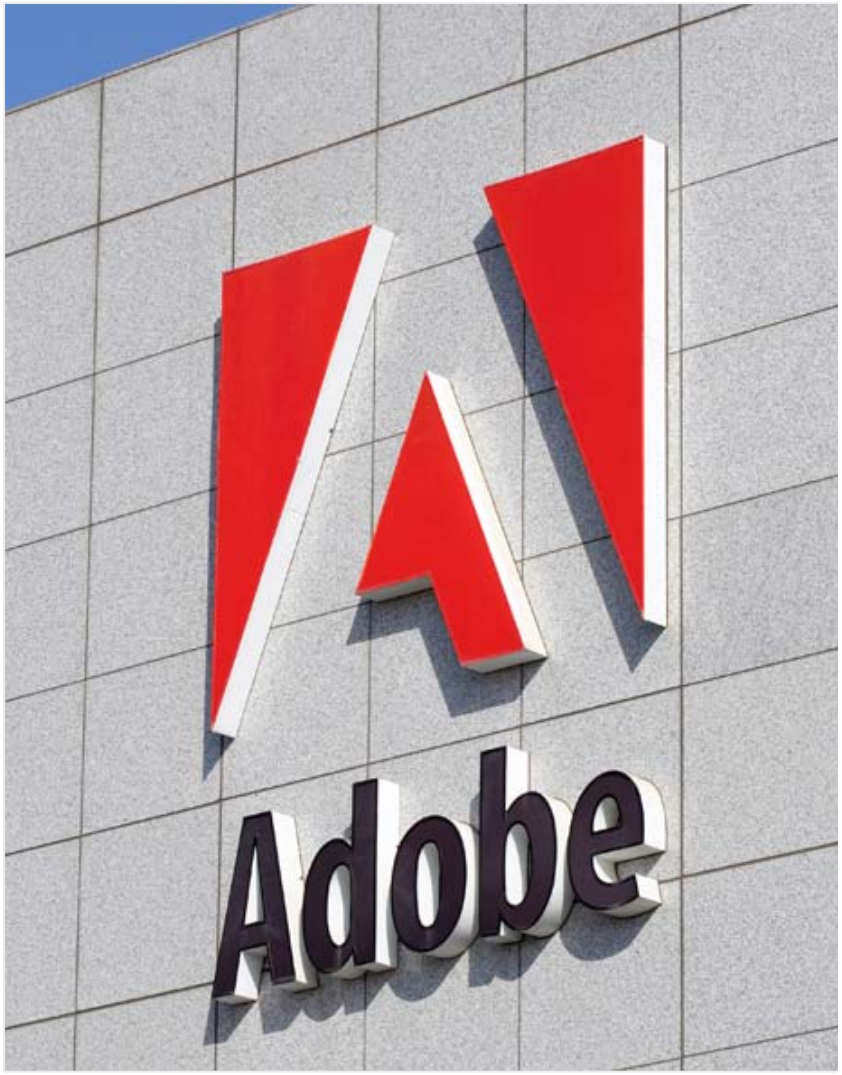
Flex Data Services for Component Developers

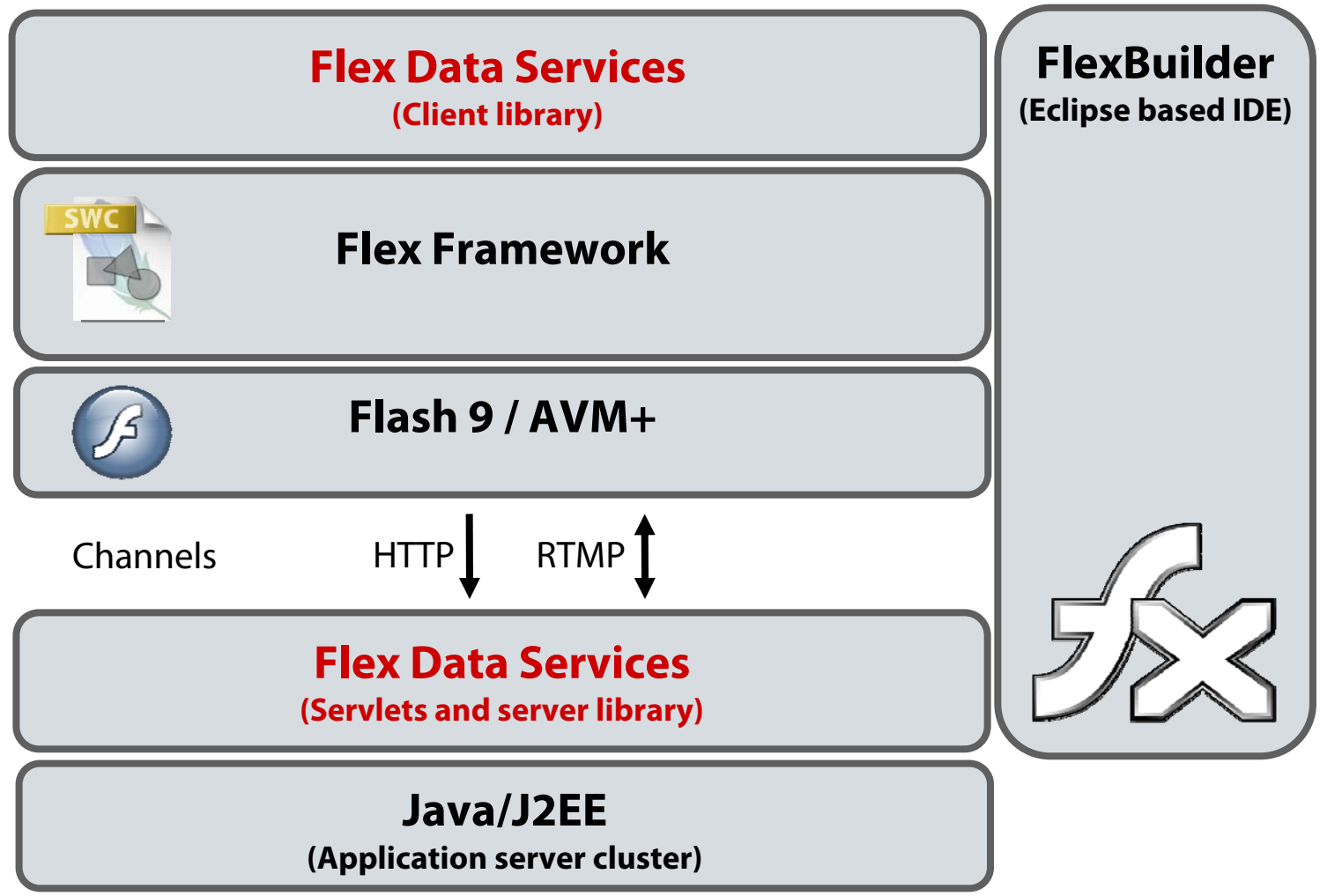
Jeff Vroom

Adobe Systems



Adobe

Flex Data Services in Flex 2



Adobe

Flex Client/Server Without Flex Data Services

- Supports clients with HTTP (using Soap, or XML)
- Requires no custom server code for writing Flex clients
- Not optimal for tight client/server coupling
  - Coding and runtime overhead for XML formatting and parsing
  - Server is unable to push messages to the client
- Not able to directly compile mxml files to swf from browser

Flex Data Services 2.0 Goals

- Attract J2EE programmers to Flex development
- Replace HTML centric technology (JSP, Struts, JSF, …)
- Work with backend technology (Hibernate, EJB, Spring, …)
- Make distributed Flex programming efficient
  - Minimize “glue code” in the development process
  - Provide high level apis for rapid development
  - Binary protocols for runtime and development efficiency
- Extend J2EE to include “server push”
  - Flash/Flex supports RTMP for real time server push to clients
  - Make it easy for J2EE programmers to adopt RTMP
- Scalable and reliable

Adobe

Three Main Components

- Remote procedure call
  - ActionScript calls Java using binary AMF protocol
- Publish/subscribe messaging
  - General bi-directional async messaging for Flex -> J2EE
  - Extend J2EE's messaging (JMS) to Flex clients
  - Clients use RTMP or HTTP transparently (HTTP uses polling for pushed messages)
- Data Management Services – declarative persistence programming for RIAs
  - Synchronizes client object model with server object model
  - Detect client changes, send to server, push to other clients
  - Conflict detection and resolution
  - Paging and lazy-loading of referenced objects for incremental loading of data

Advanced Concepts - Remote Object

- Scope and attribute-id for managing components
  - Scope: request, session, application – control lifecycle of your remote object
  - Attribute-id: stores component under this name in ServletContext or HttpSession to share with JSP or other destinations
- FlexFactory interface
  - Java interface you implement to lookup a component for remote object or FDMS assembler
  - Example configuration:

```
<destination name="SpringComponent">
    <properties>
        <factory>spring</factory>
        <source>MySpringComponentId</source>
    </properties>
</destination>
```

- FlexSession
  - Retrieved from FlexContext
  - Uses HTTP session if you are on HTTP, emulates most of the HttpSession functionality for RTMP
- [Transient] metadata
  - Annotation for AS property, do not serialize to server

Adobe

Advanced Concepts – Messaging

- Subtopics
  - Allows you to create dynamic groups of consumers
  - Supported by the AS messaging adapter
  - Must be enabled explicitly for a destination (<allow-subtopics>true</allow-subtopics>)
  - Product and Consumer expose a "subtopic" property
- Send messages to Flex Clients from Java code
  - Use JMS – no Java code needed
  - Use Messaging API (see dashboard/Feed.java for an example)
- Custom messaging adapter
  - Override invoke (message sent from client)
  - Override allowSubscribe, allowSend: specific authorization hooks for subtopics
  - Can also handle subscription messages and override routing logic
  - Methods to push messages to clients by destination, or by client id and also to route messages to other servers in the cluster

Advanced Concepts - Data Management Services

- Paging
  - Query results pulled down from client a page at a time as requested
- Managed associations
  - Supports replicating general object graphs from server to client (with cycles)
  - Tracks changes to relationships and pushes them to the server
  - Ensures a single instance of the client object for each id
  - Optionally associations can be “lazy” – referenced item is fetched as needed
- Writing assemblers
  - Implement these methods: getItem, createItem, updateItem, deleteItem, fill, count
  - Source to HibernateAssembler is included