

Flex体系架构深度剖析



Adobe

马鉴

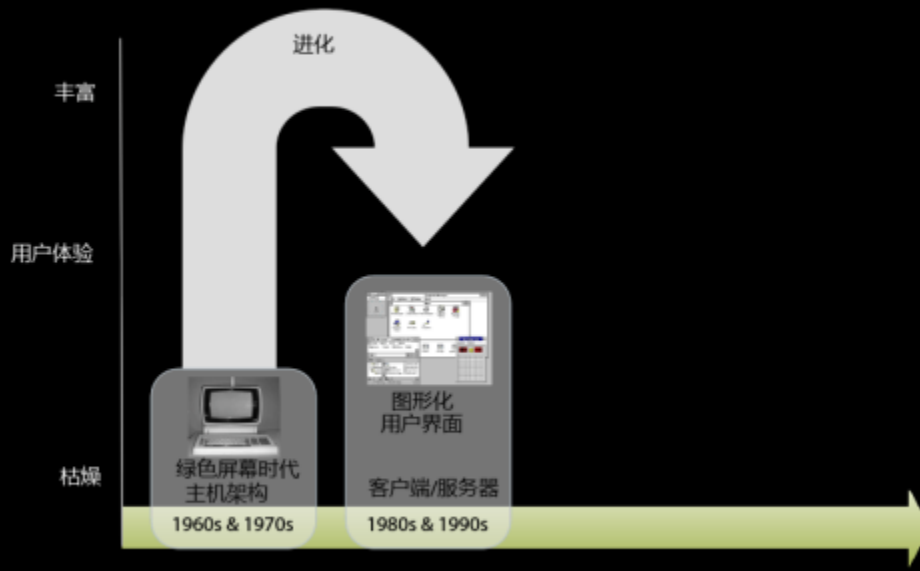
zma@adobe.com

Platform Technology Evangelist

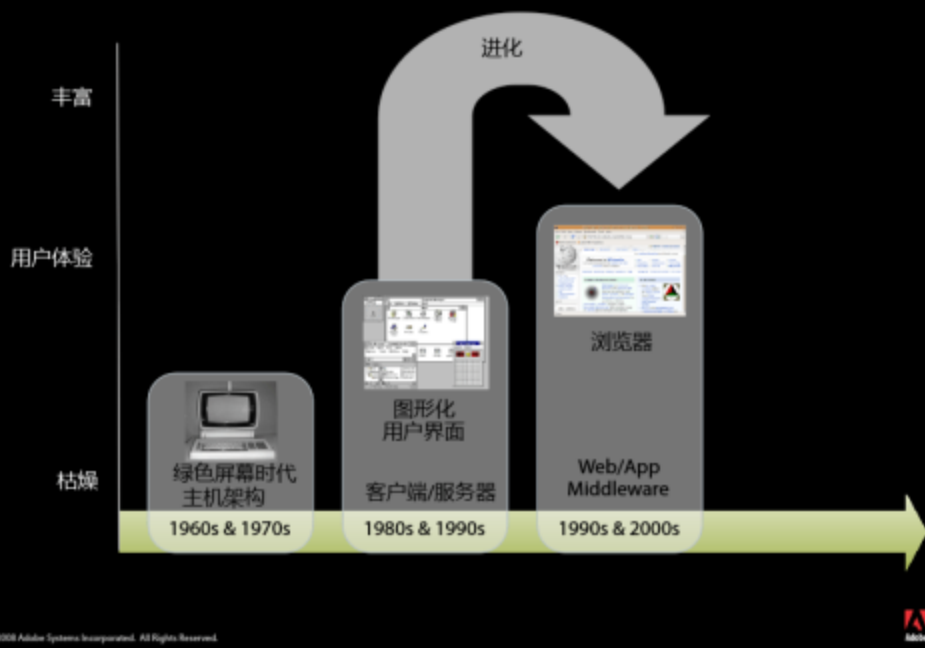
Blog: www.7yue.com



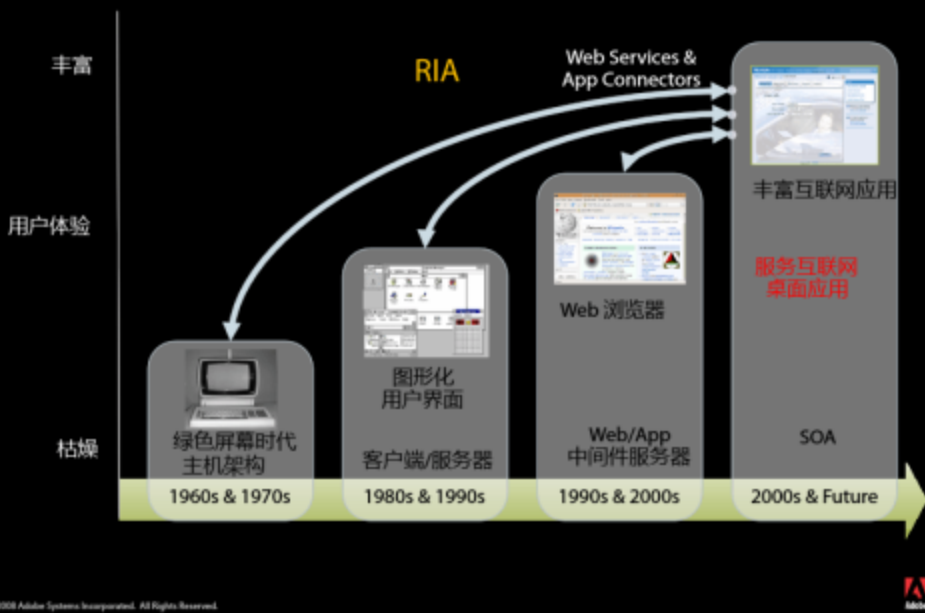
企业应用进化之路



企业应用进化之路



企业应用进化之路



Flash Platform & Flex 产品体系架构



© 2008 Adobe Systems Incorporated. All Rights Reserved.



Adobe Flash 平台技术家族

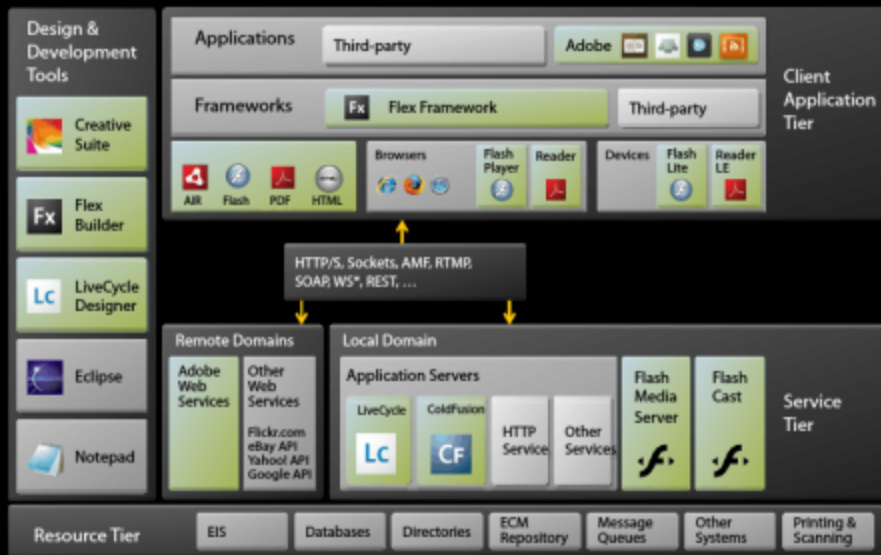


<p>设计开发工具集合</p> <ul style="list-style-type: none">Fl Adobe Flash CS4 ProfessionalFC Adobe Flash CatalystFx Adobe Flex Builder	<p>框架</p> <ul style="list-style-type: none">Fx Adobe Flex®	<p>客户端</p> <ul style="list-style-type: none">Adobe Flash® PlayerAdobe AIR™	<p>服务器端</p> <ul style="list-style-type: none">Adobe Flash Media Server familyData Services
--	---	---	---

© 2008 Adobe Systems Incorporated. All Rights Reserved.



针对RIA的Adobe技术整体架构



© 2008 Adobe Systems Incorporated. All Rights Reserved.



Flex UI + SOA = SOC

Front End

- 数据模型
- 服务
- 消息
- 持久性
- 跨操作系统
- 简洁的模式

Back End

- 数据模型
- 服务
- 消息
- 持久性
- 跨操作系统
- 简洁的模式



© 2008 Adobe Systems Incorporated. All Rights Reserved.



Flex体系架构划分

产品



工具及服务



应用开发



© 2008 Adobe Systems Incorporated. All Rights Reserved.



Adobe Flex 产品架构



Adobe®

Flex™ 3 SDK

Open source Flex SDK

MXML and ActionScript 3.0

框架和类库

Command-line 编译

© 2008 Adobe Systems Incorporated. All Rights Reserved.



Flex 3 Open Source

■ Flex 3 开源

- SDK源文件基于MPL开源许可，包括framework, compiler, debugger
- 公开的bug系统（基于JIRA），路线图及每日构建下载
- opensource.adobe.com/flex
- 三种形式
 - Free Adobe Flex SDK
 - Open Flex SDK
 - Flex UI Component
 - AIR SDK
 - Open Flex SDK
 - Compiler+Frameworks
 - Flex SDK add-ons for Open Flex SDK
 - UI Component
 - AIR SDK



Flex核心 - MXML

```
<mx:Application>
  <mx:WebService id="ws" wsdl="catalog.wsdl"/>
  <mx:Button label="Get Data" click="ws.getProducts()"/>
  <mx:DataGrid dataProvider="{ws.getProducts.result}"/>
  <mx:LineChart dataProvider="{ws.getProducts.result}"/>
</mx:Application>
```



MXML – 组件

```
<mx:Application>
  <mx:WebService id="ws" wsdl="catalog.wsdl"/>
  <mx:Button label="Get Data" click="ws.getProducts()"/>
  <mx:DataGrid dataProvider="{ws.getProducts.result}"/>
  <mx:LineChart dataProvider="{ws.getProducts.result}"/>
</mx:Application>
```



MXML – 对象实例

```
<mx:Application>
  <mx:WebService id="ws" wsdl="catalog.wsdl"/>
  <mx:Button label="Get Data" click="ws.getProducts()"/>
  <mx:DataGrid dataProvider="{ws.getProducts.result}"/>
  <mx:LineChart dataProvider="{ws.getProducts.result}"/>
</mx:Application>
```



MXML – 属性

```
<mx:Application>
  <mx:WebService id="ws" wsdl="catalog.wsdl"/>>
  <mx:Button label="Get Data" click="ws.getProducts()"/>
  <mx:DataGrid dataProvider="{ws.getProducts.result}"/>
  <mx:LineChart dataProvider="{ws.getProducts.result}"/>
</mx:Application>
```



MXML – 事件

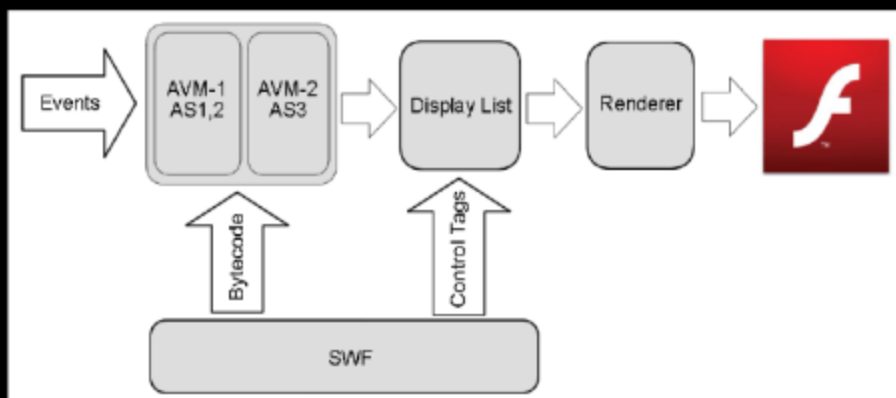
```
<mx:Application>
  <mx:WebService id="ws" wsdl="catalog.wsdl"/>
  <mx:Button label="Get Data" click="ws.getProducts()"/>
  <mx:DataGrid dataProvider="{ws.getProducts.result}"/>
  <mx:LineChart dataProvider="{ws.getProducts.result}"/>
</mx:Application>
```



MXML – 数据绑定

```
<mx:Application>  
  <mx:WebService id="ws" wsdl="catalog.wsdl"/>  
  <mx:Button label="Get Data" click="ws.getProducts()"/>  
  <mx:DataGrid dataProvider="{ws.getProducts.result}"/>  
  <mx:LineChart dataProvider="{ws.getProducts.result}"/>  
</mx:Application>
```

Flex核心-基于AVM2的编译器及Flash Player



Flash Player 一些内部机制 2种渲染模式

▪ Retained模式

- 高端模式，AS3是Retained Mode
- 能够成为跨越Adobe产品功能的标准
- 分离渲染本身和功能依赖要求，如GPU，Cell等
- 能够分离处理的类型，如video, audio
- 在渲染计算中以表面特征出现（Surfaces）
 - BitmapData, Filters, cacheAsBitmap都是表面特征
 - Bitmap本身会被独立保留
- 解决了渲染模式中的不连续性
- 体系庞大
- 难于维护和保持兼容性



Flash Player 一些内部机制 2种渲染模式

▪ Immediate模式

- 底层模式
- 灵活
- 易于维护及保持兼容

▪ 缺点

- 绑定了功能到底层语言与硬件驱动

▪ 结论

- Pixel Bender使用Immediate模式
- GPU加速使用Immediate模式
- 分离为表象的部分使用Immediate模式
- Bitmap和其他fills填充为Retained模式
- AS3是Retained模式
- 两者混合可以优化Binary下的Retained效率



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。
如要下载或阅读全文，请访问：<https://d.book118.com/977113045154006114>