系统架构

# 概述:

解决方案包括四个主要组件: Windows Phone客户端应用程序, Web Role, Worker Role, 以及非托管组件.

Windows Phone

Web Role

Worker Role

非托管组件

Blob

Queue

正如上图中所示,Windows Phone客户端应用程序连接到Web Role(上传短影到云). Web Role和Worker Role不直接相互沟通.相反地, 他们通过Queue进行通信. 解决方案中频繁使用Blob存储.直接使用Web Role和Worker Role对Blob的访问存储帐户键, 而手机客户端通过SAS访问.

在Web/Worker Role显然也使用数据表存储. 为节省空间在此不再演示. 最终, Worker Role 使用非托管组件编码视频.

将来, 我们将创建一个单独的托管的特效库项目以及一个单独的非托管的特效库项目. 但在此版本中, 我们把托管的特效库项目放在Windows Phone 项目中, 把非托管的特效库项目放在非托管代码项目中.

# Windows Phone 客户端:

Windows Phone客户端应用程序是终端用户使用的主要程序.下面是其主要组成部分的清单:

## 页面:

* MainPage: 主页. 包含用户创建新的短影, 或者打开现有短影的按钮.
* ComposePage: 应用程序最重要的页面. 用户可以在此合成短影.
* PreviewPage: 用户可以使用此页在已编码成视频的短影上传到云之前预览. 此页面使用Silverlight 动画提供预览. 它也使用相关特效库显示特效预览.
* ChooseStoryPage:显示已保存的短影的列表,并允许用户从中选择一个开始工作的简单页面.
* ChooseMediaPage: 当用户需要添加新的照片到当前的短影时显示此页面. 它使用Windows Phone MediaLibrary API 查询手机上的照片.

## 变换组件:

* TransitionFactory: 基于名称创建具体的特效类的组件. 但是它返回ITranstion 而不是具体的类.
* ITransition: 主应用程序和特效类使用的接口.
* TransitionBase: 提供ITransition一些默认实现的虚拟类.
* FadeTransition: 实现简单淡出效果的特效.
* FlyInTransition: 实现飞入效果的特效.
* FlyInTransition\_Design: 关联到飞入特效的简单的设计界面. 当在合成页面渲染时提供一个额外的设计界面.

## 其他组件:

* StoryServiceLocator: 负责调用云服务的服务定位.
* IsolatedStorageHelper: 与Windows Phone 孤立存储互动的类.
* PersistenceHelper: 负责序列化/反序列化短影的类.
* BitmapHelper: 用以重置图片大小的类.

同时还有其他值转换, 模型, 视图模型等等.

# Web Role:

Web Role 包含两个主要组件:

* StoryService: 使用WCF Web API建立的REST服务. 提供了上传短影, 以及列表显示已编码视频的特性.
* Story viewer: 页面调用REST 服务列表显示已编码视频, 同时使用 HTML5 视频标签渲染已编码视频的简单的HTML5 页面.

# Worker Role:

Worker Role用以编码视频. 它委托实际编码工作给非托管组件, 同时处理所有与Azure 存储之间的交互.

# 非托管组件:

非托管组件负责编码视频. 它包括下列主要组件:

* VideoEncoder: 编码视频的主组件. 它同时负责载入照片位图文件. 我们使用Media Foundation编码视频. 特别指出, 我们使用Windows 7 与Windows Server 2008 R2的Media Foundation带来的新特性: sink writer.
* XmlParser: 自制简单的xml解析器. 用来解析短影配置文件.
* 特效相关组件: 与Windows Phone 项目的特效库类似. 但是以非托管代码实现.