# 七牛云存储调研

# 产品特性

七牛云储存提供非结构化数据的一站式托管服务，产品特性如下：

* 安全存储
  + 云存储
  + 多副本、多数据中心
  + 安全机制
  + 弹性空间
  + ...........
* 分发加速
* 上传加速
  + - * 就近上传
      * 断点续上传
      * 并发分块上传
* 下载加速
  + - * CDN
* 数据处理
  + - 图片处理
      * 缩略、裁减、水印、等等
    - 音视频处理
      * 视频转码、截帧、打⽔印、等等
    - 文档处理
      * 格式转转(转PDF、转图片等等)
    - 数据分析
      * 文件存储纪录、下载量、API请求数、等等
* 直播服务
  + 智能调度
  + 实时转码
  + 丰富SKD
  + 质量监控

# 基本概念

## 2.1资源

资源是七牛云存储服务中的逻辑存储单元。对于每一个账号，该账号里存放的每个资源都有唯一的[空间](http://developer.qiniu.com/article/newbie-guide.html" \l "bucket" \o "空间)与[键(Key)](http://developer.qiniu.com/article/glossary/" \l "j)标识。资源键名是一个字符串，例如：level1/level2/example1.jpg，它可以包含任意字符（包括UTF-8编码形式的Unicode字符）。

使用者可以在上传资源时为其指定一个方便管理的键名，通过前缀来达到类似于文件目录的分类和层次效果。例如对于一个网站的资源，我们可以用如下键名命名资源：

index.html

features/index.html

imgs/features/feature1.png

about.html

假设这些资源都位于某个绑定了域名 example.com 的公开空间中，则用户可以在浏览器里输入如下URL访问这些资源：

<http://www.example.com/features/index.html>

## 2.2空间

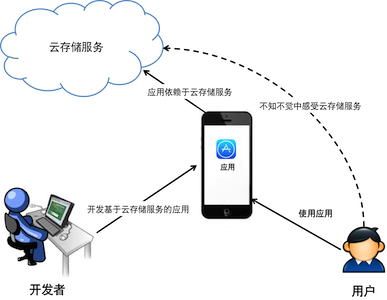
空间是资源的组织管理单位，一个资源必然位于某个空间中。可以为每个空间设置一系列的属性，以对资源提供合理的管理动作。 常见的属性有：

* 将空间设置为公开或私有，以控制对空间内资源的访问权限。
* 设置资源的数据处理样式，以便于用简短方式对资源进行处理。



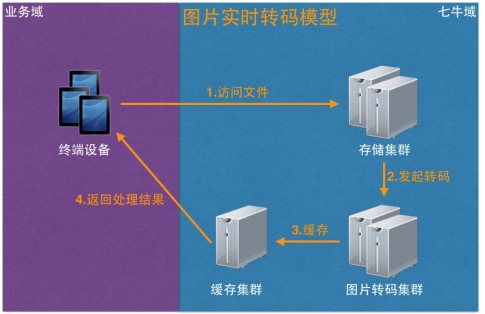
## 2.3开发者和用户

开发者是七牛云存储服务的直接使用者，用户是开发者所推出产品的使用者即七牛云存储服务的间接使用者。



# 应用场景

## 3.1图片展示类及图片UGC应用



### 3.1.1 客户案例之穷游网

穷游的在网页端和移动端 APP 的静态资源链接都使用了七牛，APP主要使用的是图片链接，网页端还包括了 html、css、js 文件，所以他们在七牛的使用过程中有这么几种方式：

1. 网站建设工作人员修改网站代码后，将静态资源从业务服务器直接同步上传 到七牛的空间里，利用七牛的 CDN 加速优化网站的呈现速度。
2. 穷游用户在编写游记后，直接将资源从客户端上传到七牛，然后通过七牛对 业务服务器的回调，让业务服务器获取上传端上传的资源信息，进行入库 处理。
3. 上面两种类型的上传，最后在呈现的时候都会经过七牛提供的 CDN 服务进 行加速，另外图片资源大多数情况下还会使用一些图片处理操作。

### 3.1.2 客户案例之视觉中国

视觉中国在七牛主要存放的是新闻图片、创意照片，使用的数据存储方案是网页端上传+镜像存储对资源的拉取。对于较冷的资源，使用的是镜像存储拉取 的方式，在七牛的空间中设置镜像源为自己图床，通过在七牛的域名访问资源去图床自动拉取资源，大大减轻了冷数据迁移的压力.



# 编程模型

## 4.1基本结构

基于七牛云存储服务构建的应用，建议使用如下基本架构：



# 数据处理

## 5.1数据处理机制

七牛云存储内建了一个非常高效易用的数据处理框架。数据处理框架可以管理和执行一系列符合规范的[数据处理](http://developer.qiniu.com/article/index.html#dora-api-handbook)操作。开发者可以在访问资源时制定执行一个或多个数据处理指令，以直接获取经过处理后的结果。

典型场景: 客户端可以上传一张高精度的图片，然后在查看图片的时候根据屏幕规格生成一张大小适宜的缩略图，例如为iPhone生成一张960x640大小的缩略图。这样既可以明显降低网络流量，而且可以提高图片显示速度，还能降低移动设备的内存占用。

原图 （[链接](http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg)）：

<http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg>

针对该原图获取一个适合iPhone5屏幕尺寸的图片（[链接](http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg?imageView/2/w/640/h/960)）：

<http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg?imageView/2/w/640/h/960>

## 5.2管道

七牛云存储的数据处理功能规格如下:

[GET] url?<fop>

即基于文件的URL通过问号传参来实现数据处理，<fop>表示数据处理指令及其所需要的参数，是File Operation的缩写，表示文件处理。那么，将一个资源经由多个 <fop> 链式处理，各 <fop> 之间用|分割，我们称之为管道(pipeline)。

管道操作的规格如下：

[GET] url?<fop1>|<fop2>|<fop3>|<fopN>

示例:将一个原图缩略，然后在缩略图上打上另外一个图片作为水印：

#原图 <http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg>

#基于原图生成缩略图

<http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg?imageView/2/h/200>

#在生成的缩略图之上打水印

[http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg?imageView/2/h/200|watermark/1/image/aHR0cDovL3d3dy5iMS5xaW5pdWRuLmNvbS9pbWFnZXMvbG9nby0yLnBuZw==](http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg?imageView/2/h/200%7Cwatermark/1/image/aHR0cDovL3d3dy5iMS5xaW5pdWRuLmNvbS9pbWFnZXMvbG9nby0yLnBuZw==)

## 5.3样式

如果觉得 url?<fop1>|<fop2>|<fop3>|<fopN> 这样的形式够冗长，还可以为这些串行的 <fop> 集合定义一个友好别名。使用七牛 [qrsctl](http://developer.qiniu.com/code/v6/tool/qrsctl.html) 工具的命令行：

定义 url 和数据处理之间的分隔符为 separator：

qrsctl separator <bucket> <separator>

定义数据处理的别名为 aliasName ：

qrsctl style <bucket> <aliasName> <fop>

示例:

// 定义 url 和数据处理之间的分隔符为 "-"

qrsctl separator <bucket> "-"

// 定义该管道数据处理样式名为 "pipeline"

qrsctl style <bucket> "pipeline" "imageView2/2/h/200|watermark/1/image/aHR0cDovL3d3dy5iMS5xaW5pdWRuLmNvbS9pbWFnZXMvbG9nby0yLnBuZw=="

那么，以下两个 URL 则等价:

原始URL

http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg?imageView/2/h/200|watermark/1/image/aHR0cDovL3d3dy5iMS5xaW5pdWRuLmNvbS9pbWFnZXMvbG9nby0yLnBuZw==

友好风格URL

http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg-pipeline

# 安全机制

在使用七牛云存储服务的过程中，需要考虑安全机制的场景主要有如下几种：

* 上传资源
* 访问资源
* 管理和修改资源

这三个场景需要考虑不同的安全因素，因此七牛针对性的提供了三种安全机制：[上传凭证](http://developer.qiniu.com/article/developer/security/upload-token.html)、[下载凭证](http://developer.qiniu.com/article/developer/security/download-token.html)和[管理凭证](http://developer.qiniu.com/article/developer/security/access-token.html)。

推荐模型:



密钥（AccessKey/SecretKey）

密钥用于以上凭证的生成。以 SecretKey 为参数，配合适当的签名算法，可以得到原始信息的数字签名，防止内容在传递过程中被伪造或篡改。

密钥通常是成对创建和使用，包含一个 AccessKey 和一个 SecretKey。其中 AccessKey 会在传输中包含，而用户必须保管好 SecretKey 不在网络上传输以防止被窃取。

# 数据分析

