

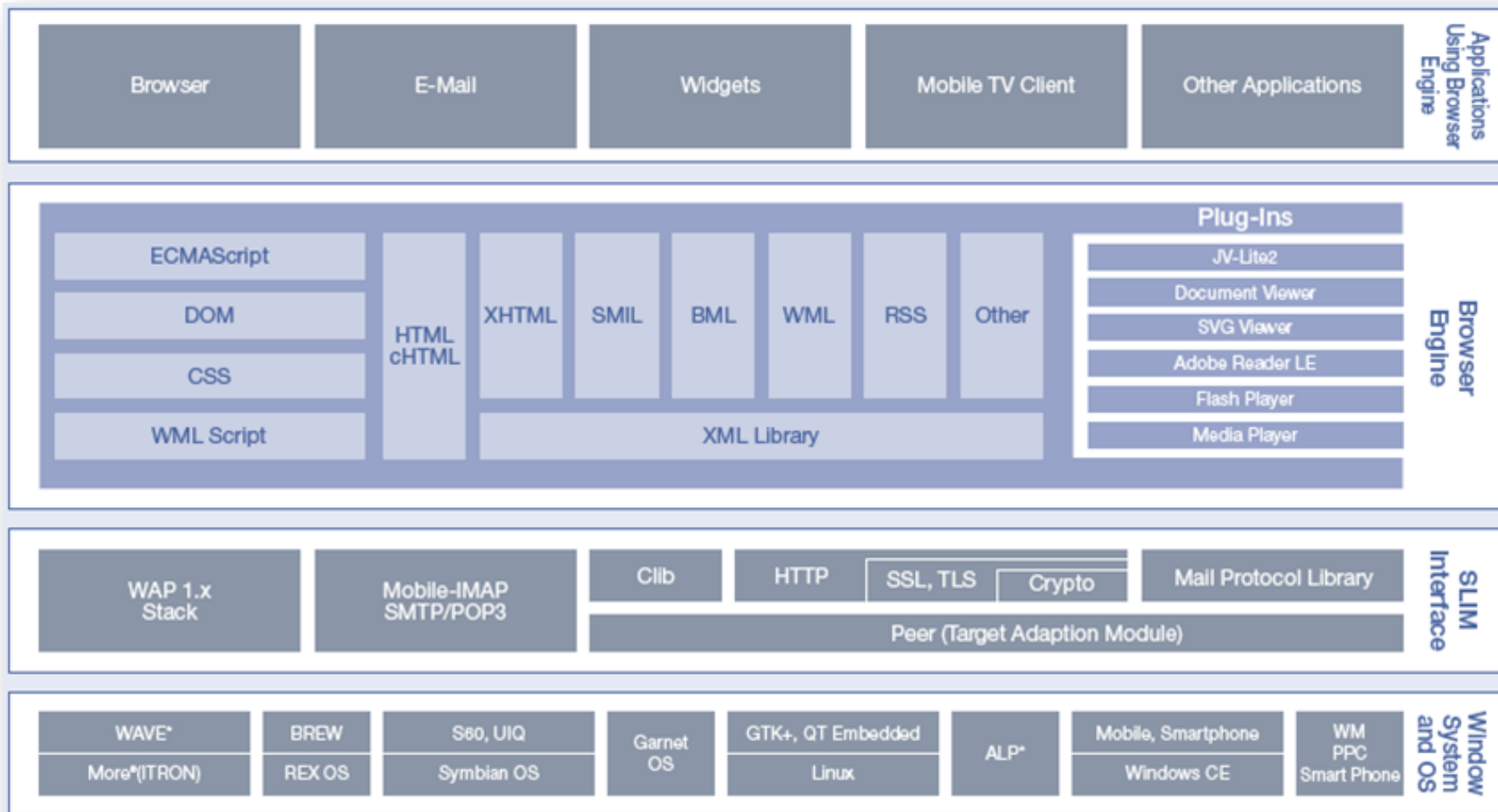
移动浏览器内核和HTML5

2012.07 iron

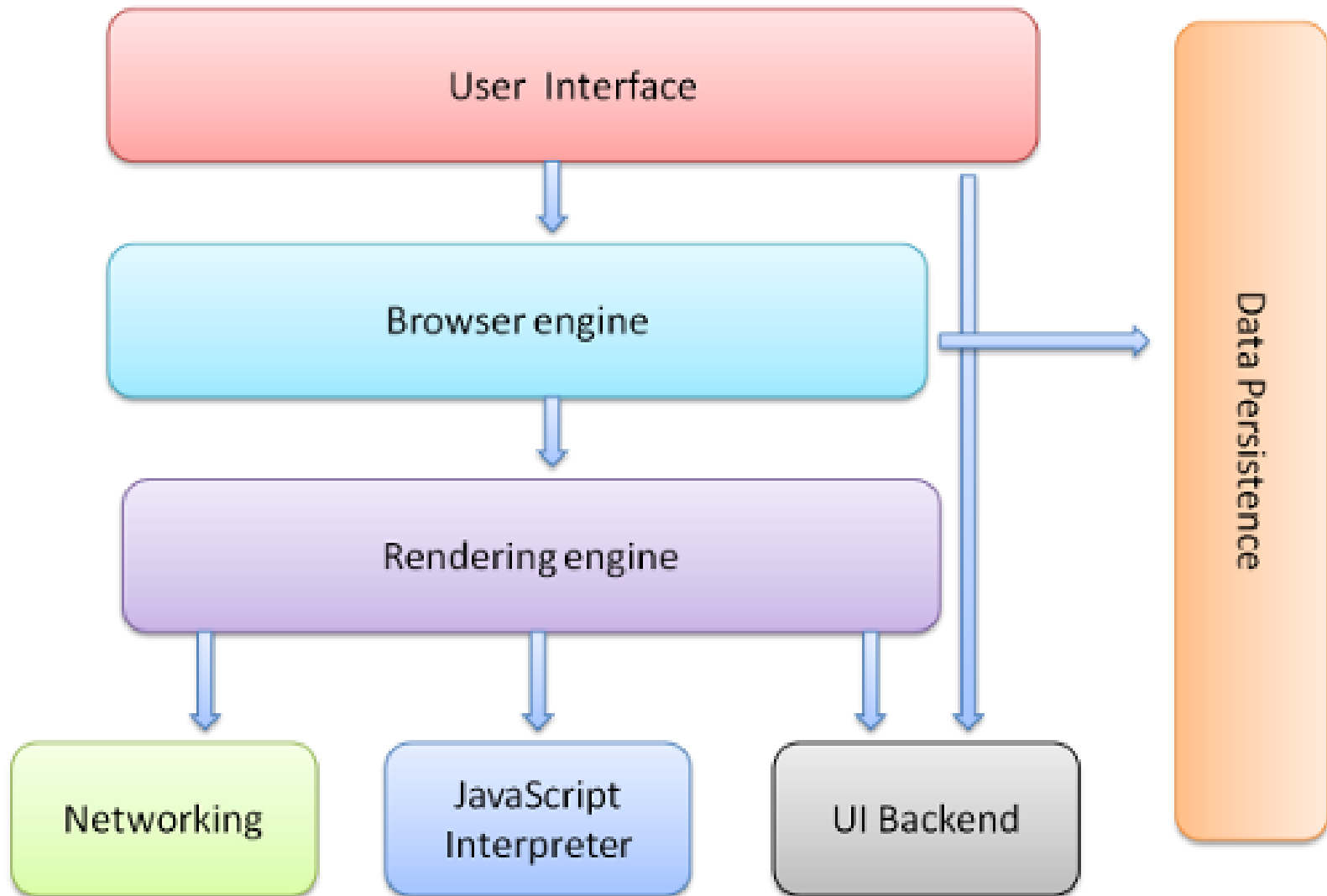
目录

- > **一、浏览器内核简介**
- > **二、移动浏览器开发特点**
- > **三、HTML新“5”台**
- > **四、QQ浏览器对HTML5的支持**
- > **五、案例分享**

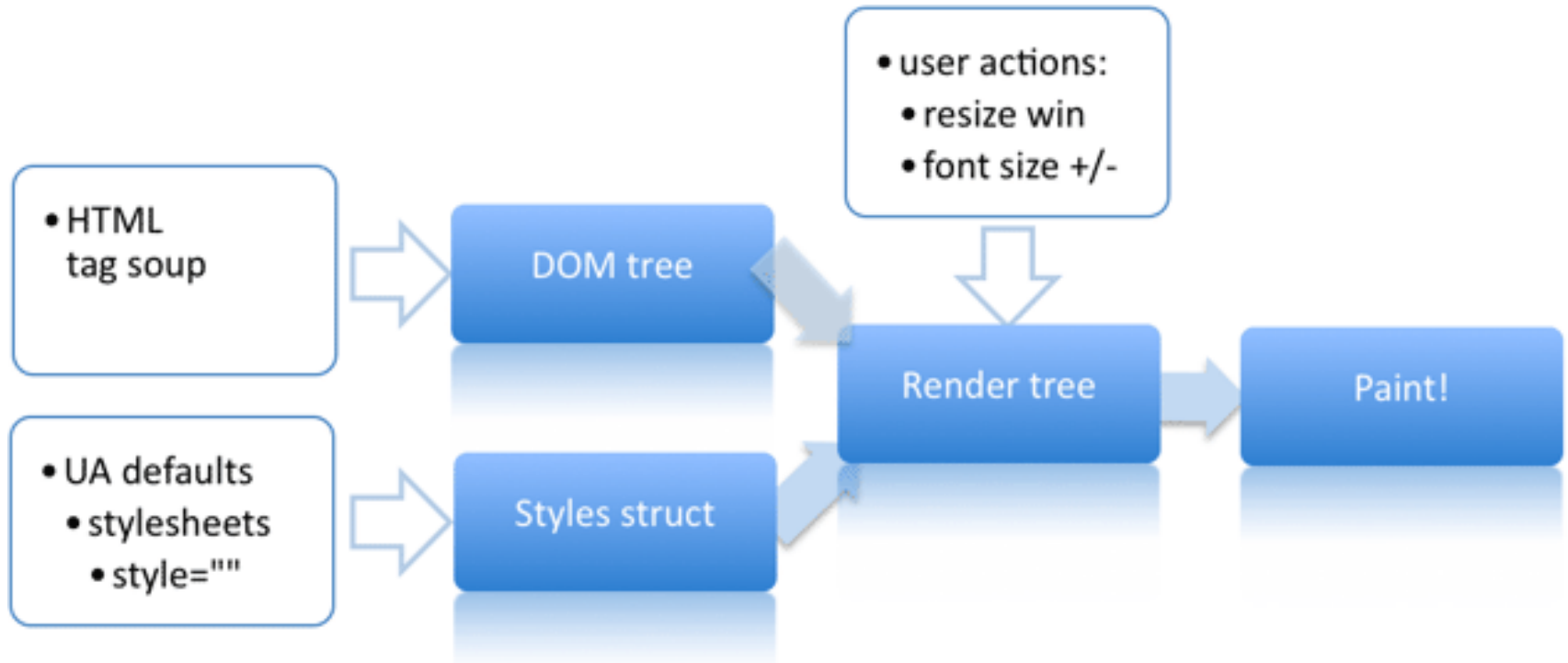
浏览器技术框架



浏览器的主要构成



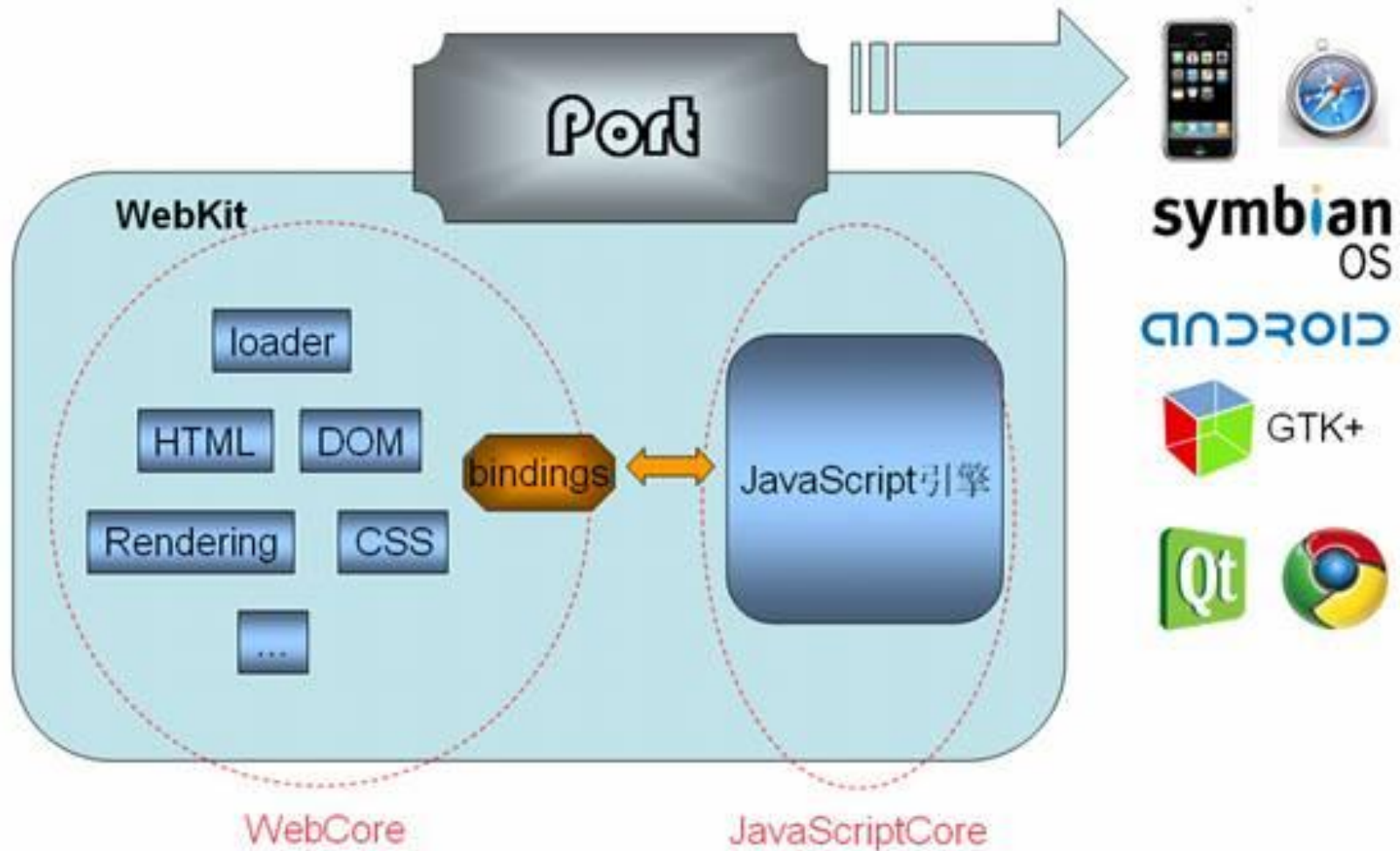
内核工作流程



浏览器内核流派



WebKit



目录

- > 一、浏览器内核简介
- > 二、移动浏览器开发特点
- > 三、HTML新“5”台
- > 四、QQ浏览器对HTML5的支持
- > 五、案例分享

平台兼容



平台兼容

> iOS :

- > **webkit**开源代码移植到iOS系统 ;
- > 输入法支持 ;
- > 优化ios渲染机制。

> Android :

- > 需要支持**2.1 – 4.0**Android系统
- > 设计了**os**中间层，这层代码负责适配各个平台的差异性，包括**skia**图形引擎，音视频播放，个别系统函数差异等;
- > 根据不同**cpu**类型，比如**arm5**，**arm7**分别调用**jit**和**js**解释器，最大限度的提升**js**性能

性能优化

- > Webkit和JS引擎的版本升级
- > 硬件加速
- > 平台相关优化
- > 用户体验优化

内存优化

- > **解决内存泄漏**
- > **消减内存最大峰值**
- > **根据系统内存状况释放资源等方法**

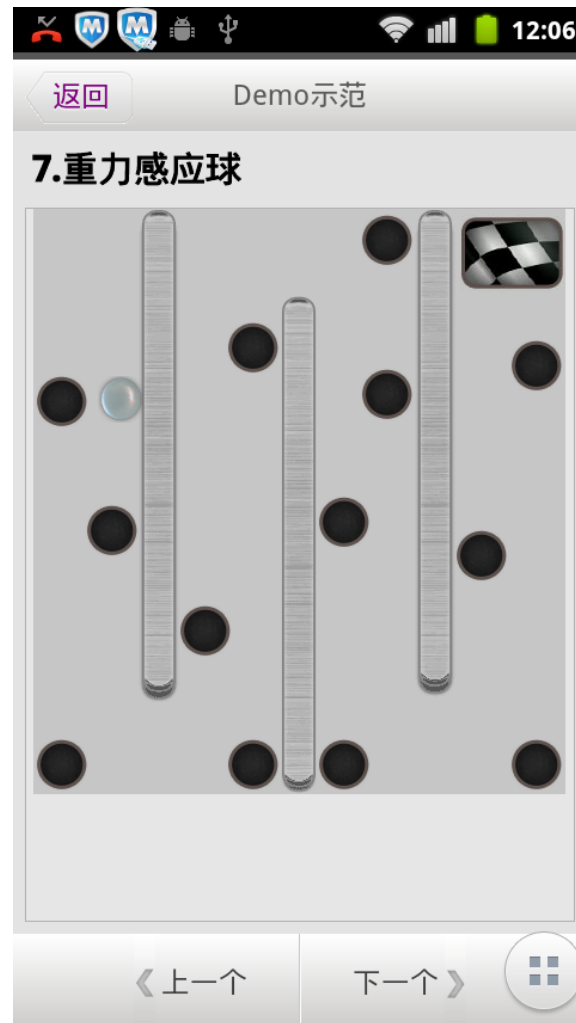
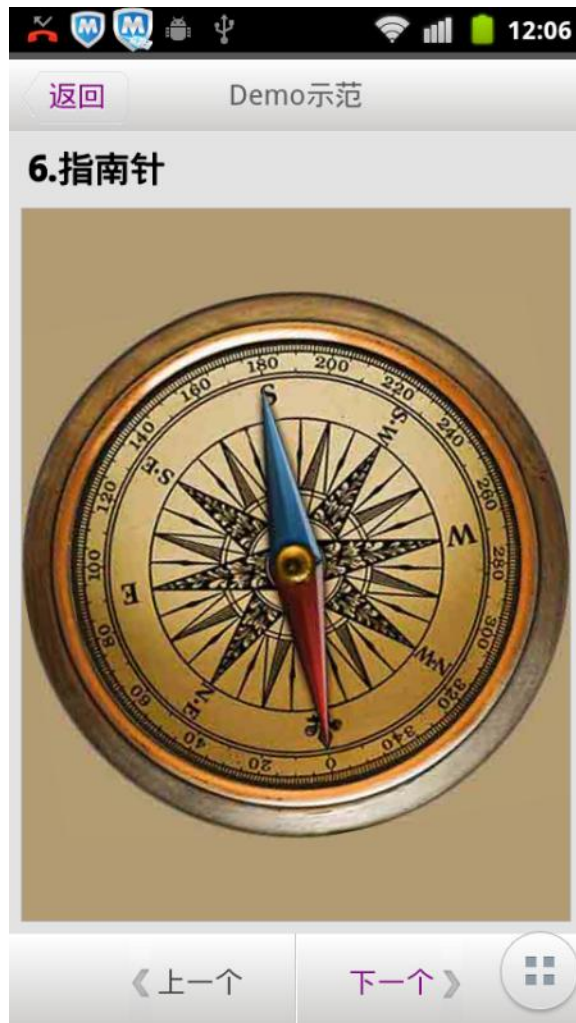
资源优化

- > 尽量调用系统的库，而不要静态编译到代码中;
- > 关闭代码中不需要的宏，比如：svg，xbl等
- > 图片多使用Nine Patch图片(绘画的是一个可以伸缩的位图图像，Android会自动调整大小来容纳显示的内容)或者webp格式
- > 链接器的选择：Gold链接器;
- > 编译参数的选择；
- > jsc和v8的选择，选择v8会导致安装包大500k左右

网速优化

- > 采用代理提升网络加载速度，并节省流量；
- > 采取了打包下载，图片降质，资源压缩，替换长连接，去广告等方法减少用户流量；
- > DNS优化；

HTML5支持



HTML5支持

- > 重点支持html5标准，大大提升html5能力；
- > 通过研发以下几个方面：
 - > **Form标签**
 - > **音视频支持**
 - > **Device API的支持**

移动APP开发特点

平台兼容

性能优化

内存优化

资源优化

网速优化

目录

- > 一、浏览器内核简介
- > 二、移动浏览器开发特点
- > 三、HTML新 “” 台
- > 四、QQ浏览器对HTML5的支持
- > 五、案例分享

HTML新 “” 台

- > **开发**—游戏和应用开发者的机遇
- > **框架**—工具化、组件化提高开发效率
- > **发行**—上线、发布、运营、渠道
- > **搜索**—其它发行渠道的遐想
- > **支撑**—手机QQ浏览器提供最佳体验

开发

> HTML5

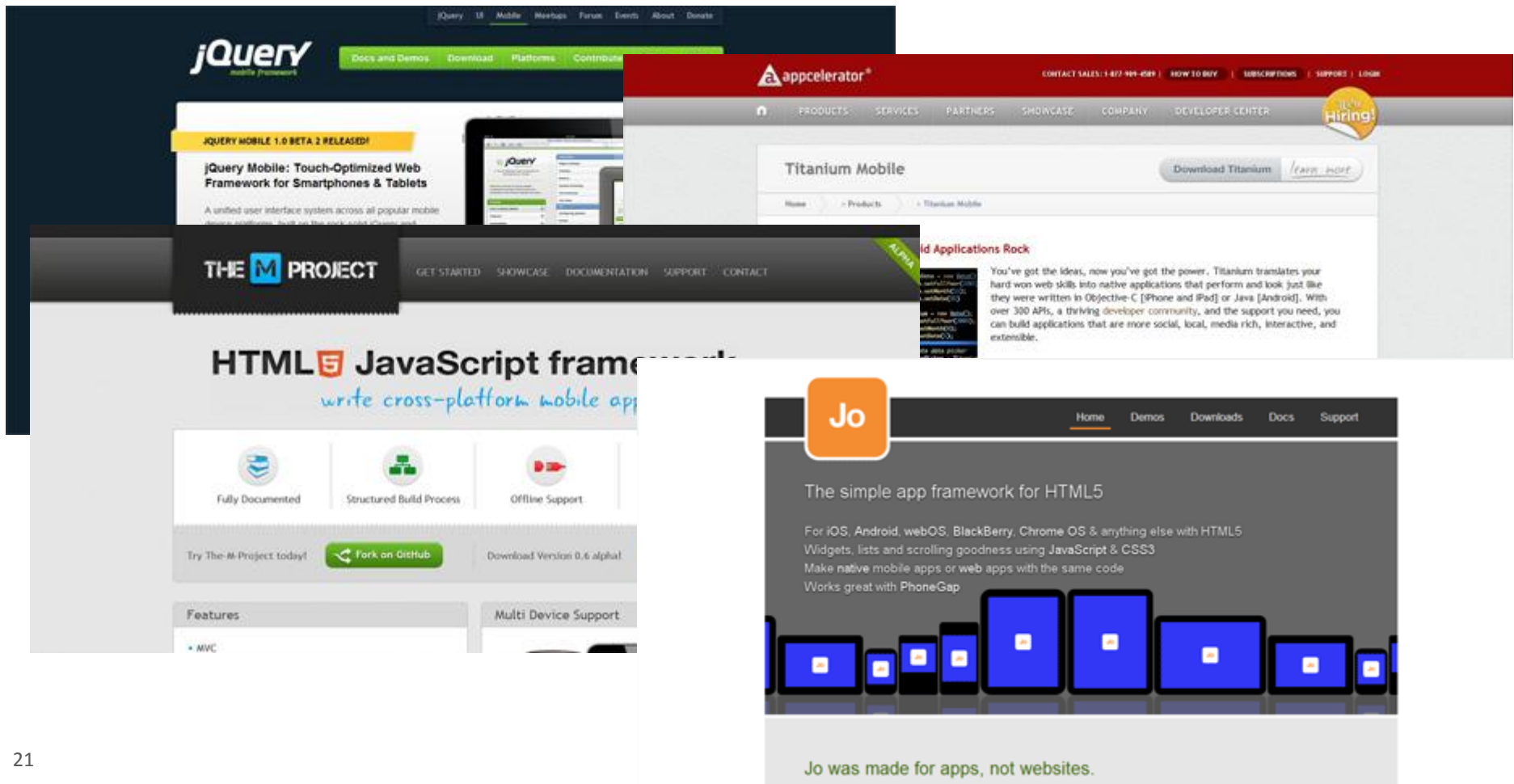
- > 和浏览器轻耦合, 应用快速发布
- > 开放性更高, 信息流互相关联
- > 跨平台, 相比Native App更易于适配
- > 突破AppStore的封锁
- > 轻量级应用
- > 结合移动设备特性

> 开发者提前储备

- > 团队能力积累
- > 游戏引擎、开放工具组件积累

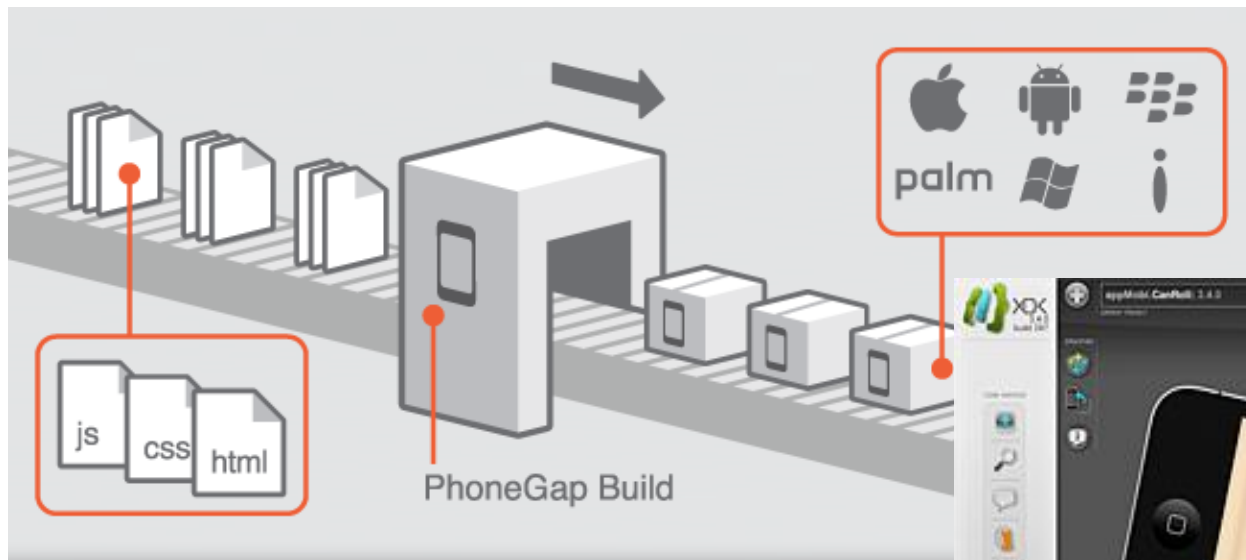
框架

> JS类—jQuery



框架

> 开发工具类- PhoneGap & AppMobi



jqMobi



框架

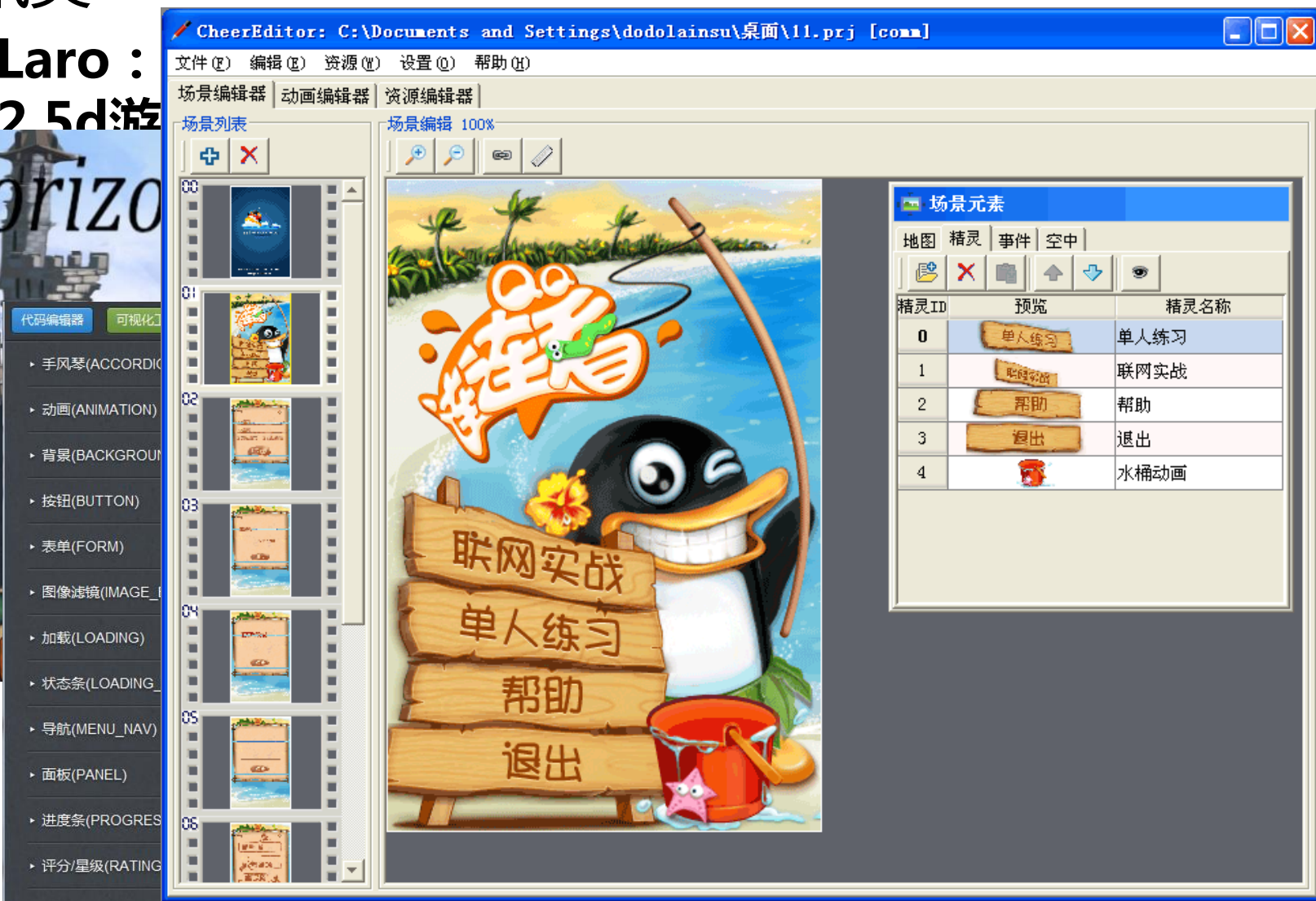
- > 游戏引擎类
 - > **Construct 2 the HTML5 Game Maker**



框架

> 腾讯类

> Laro : 2.5d游



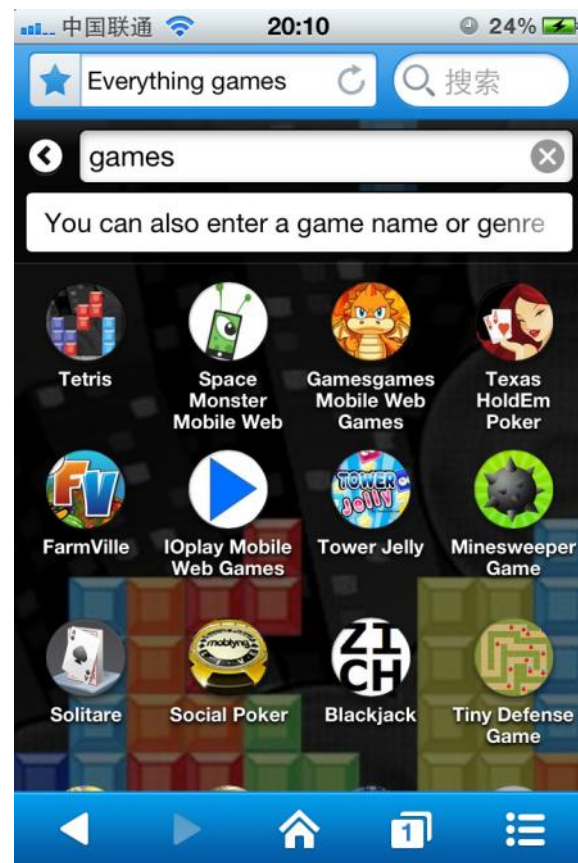
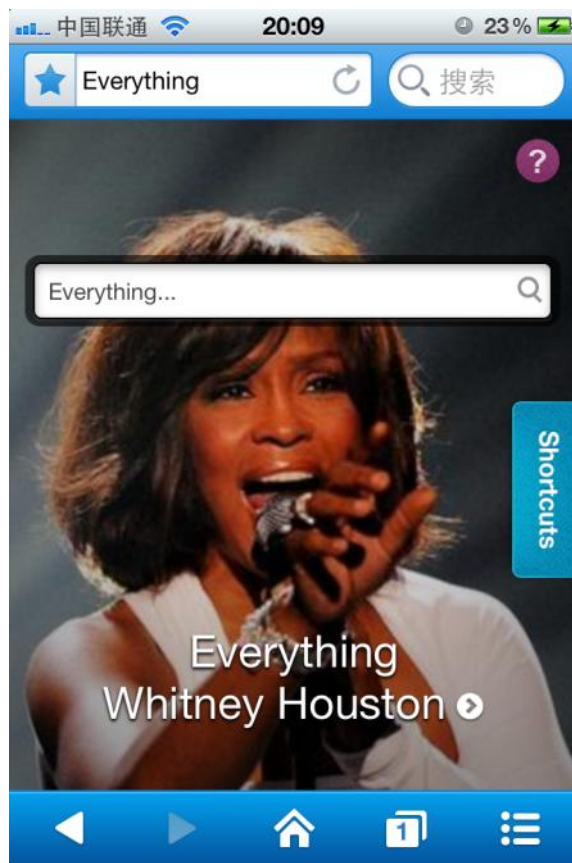
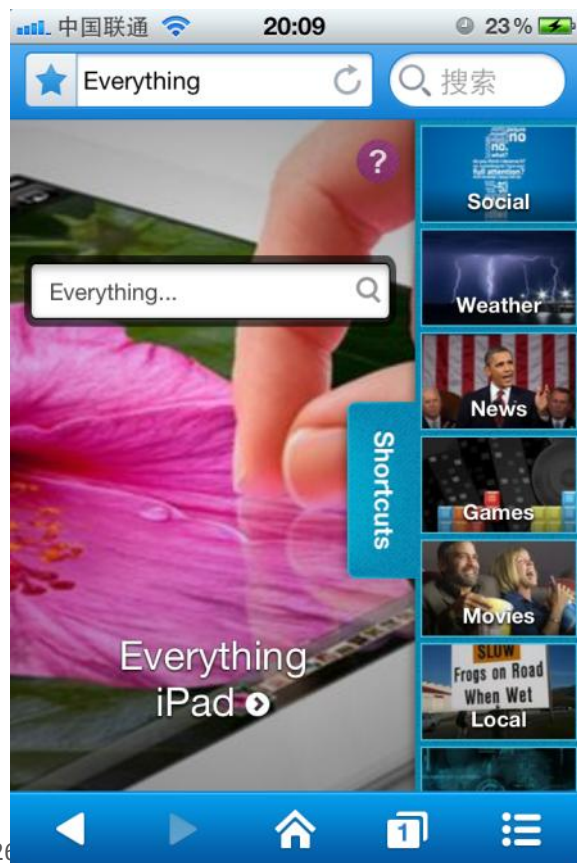
发行

- > **Chrome Web Store**
- > **移动运营商的WAC (Wholesale Applications Community)**
- > **Q+ & 腾讯开放平台**
- > **HTML5 QQ游戏大厅**

搜索

搜索—其它发行渠道的遐想

<http://everything.me>



支撑

> 手机QQ浏览器

> JS引擎性能

> 图像渲染算法

> GPU等硬件加速

> HTML5标准支持的完善

目录

- > 一、浏览器内核简介
- > 二、移动浏览器开发特点
- > 三、HTML新“5”台
- > 四、QQ浏览器对HTML5的支持
- > 五、案例分享

HTML5关键技术



HTML5技术 —— 画布（Canvas）



使用豌豆荚截图 wandoujia.com

```
var canvas = document.getElementById("compass");  
var ctx = canvas.getContext("2d");
```

HTML5技术 —— CSS3（动画）



使用豌豆荚截图 wandoujia.com

```
var el = document.querySelector('div');  
  
el.style['webkitTransitionDuration'] = '300ms';  
  
el.style['webkitTransform'] = 'translate3d(-100%, 0, 0)';
```

HTML5技术 —— WebSocket



```
var conn = new
WebSocket("ws://html5rocks.websocket.org/echo");
showMsg('WebSocket正在连接中...');
conn.onopen = function ($e) {
    showMsg('WebSocket已经连接成功');
};
conn.onmessage = function ($e) {
    showMsg(msg);
};
conn.onclose = function ($e) {
    showMsg('WebSocket已经关闭连接');
};
conn.onerror = function ($e) {
    showMsg('WebSocket连接错误');
};
```


HTML5技术 —— FileSystem

```
window.webkitRequestFileSystem(TEMPORARY, fileSize, function(fs) {
    fs.root.getFile('data.js', {create:true, exclusive:false}, function(fileEntry) {
        fileEntry.createWriter(function(fileWriter) {
            fileWriter.onwrite = function(e) {
                callback && callback();
            };
            fileWriter.onerror = function(e) { };
            var builderInstance = new WebKitBlobBuilder();
            builderInstance.append(JSON.stringify(data));
            fileWriter.write(builderInstance.getBlob('text/plain'));
        }, function() {
            alert('createWriter error');
        });
    }, function() {
        alert('getFile error');
    });
}, function() {
    alert('request filesystem error');
});
```

HTML5技术 —— 摄像头



```
<input type="file" onchange="handleCamera()"
accept="image/*" capture="camera" />
```

使用豌豆荚截图 wandoujia.com

HTML5技术 —— 地理位置



使用豌豆荚截图 wandoujia.com

```
if (navigator.geolocation) {  
    navigator.geolocation.getCurrentPosition(  
        function(position) {  
            ...  
        }, errorHandler);  
}
```

HTML5技术 —— 感应器



使用豌豆荚截图 wandoujia.com

```
if (window.DeviceOrientationEvent) {  
    window.addEventListener('deviceorientation', running,  
        false);  
} else {  
    alert("很抱歉，您的浏览器或者设备不支持传感器！");  
}
```

HTML5 技术 —— 本地存储

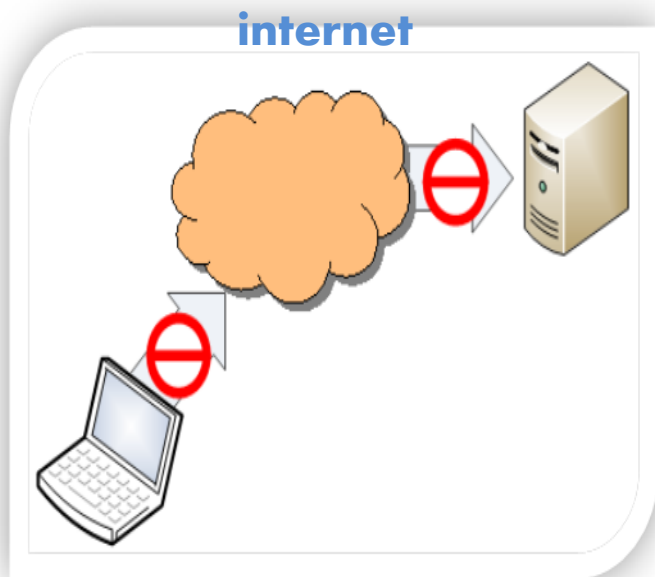
- > LocalStorage/SessionStorage
 - **SessionStorage:** 用于运行时数据临时存储
 - **localStorage:** 用于应用高速数据存取，加快运行速度
- > LocalStorage比较于Cookie

		Web Storage	
异	数据存储容量大小	5MB	4KB
	是否向服务器发出请求	否	是（包含在每个HTTP请求中）
同	都存储在客户端		
	数据不会因页面刷新或关闭等操作而改变		

减轻Web服务器的负载

节省网络的带宽，降低数据传输涉密风险

HTML5技术——离线应用



- > 保证离线状态下的Web应用能够正常使用
- > Web应用的相关资源文件会被缓存至本地



目录

- > 一、浏览器内核简介
- > 二、移动浏览器开发特点
- > 三、HTML新“5”台
- > 四、QQ浏览器对HTML5的支持
- > 五、案例分享

iOS系统相关

- > 刚开始在做ApplicationCache功能时，我们在初始化的时候打开某些功能（key-value字典的方式），但经常无缘故地失败，后来才发现，ios自带的webkit注册了像WebKitOfflineWebApplicationCacheEnabledPreferenceKey这样的键值(40多个相同键值)，我们键值的名字跟系统webkit一样，导致经常被更改了。
- > 解决：把webPreferences.mm里所有的key字符串全部更名处理。（如在前面加上QQ两个字）

H5.qq.com站点支持



解决方案

- > 分析该站点的html源代码，发现有用到ApplicationCache相关功能，所以，先适配ApplicationCache，这时候可以看到进度条，但进度条不走。
- > 再根据分析源代码，发现还有用到websocket的功能，然后把websocket适配好，就可以看到进度条正常工作，并正常下载相关游戏组件与数据了。

H5.qq.com站点支持



首屏绘制速度提升

- > 刚开始移植Mac版本webkit到ios平台后，首次绘制一个网页时间大概需要3-5分钟，分析整个流程，有三大问题导致绘制速度慢：
- > 问题一：网页有多大，就绘制多大的位图，不仅大量占用内存，还导致首次绘制超级慢。
- > 问题二：单线程导致非常卡。在解析和排版的时候，界面基本无法操作。
- > 问题三：发现Layout排版占用了1/3左右的时间

解决方案（做了大量的尝试和预研）

> 针对问题一：

- > 1). 实现TiledBackingStore机制，实现ViewPort仅为屏幕可视区域的宽高。
- > 保证每次只绘制当前屏幕的内容，而不需要绘制整个网页那么大。
- > 2). 实现分块绘制，提高页面滑动过程中，位图块的重用率。

> 针对问题二：

- > 修改为双线程，把解析和排版放在一个线程，UI绘制放在单独的线程，保证在解析或排版过程中，不影响用户的操作。

> 针对问题三：

- > 先分析Layout排版过程，哪些API或函数执行慢。发现加载字体花掉了大部分时间，而加载字体是与系统API相关。因此，优化为启动时预加载字体，减少排版时加载字体的时间。

内存优化相关

- > X5内核在多窗口情况下，深度浏览WWW页面时，RAM占用直线上升，很容易达到Android系统的内存报警阈值，因为Android系统没有交换分区机制，所以必须手工处理内存占用过大的情况。
 - 。
- > 基本思路：当APP收到内存报警消息后，通知各个窗口内存低需要清理，窗口收到消息后发现自己是在后台，则从内存中把窗口占用的空间全部释放，只保留前进后退列表，如果窗口收到消息后发现自己是在前台，则释放前进，后退内存的页面，当前页面保留，当多窗口切换到或者前进后退到释放的页面时，重新拉取页面。

关于iron

- > 邱会发
- > [Email: iron@qq.com](mailto:iron@qq.com)
- > QQ: 30637145
- > 腾讯微博: iron88
- > [MSN: toiron@hotmail.com](mailto:toiron@hotmail.com)

Q & A

谢谢！