


阿里巴巴在线技术峰会  
Alibaba Online Technology Summit

# 电商互动营销的技术实现

天猫技术部-互动平台  
继勋

 Alibaba Group  
阿里巴巴集团

 阿里云  
aliyun.com

互动营销案例

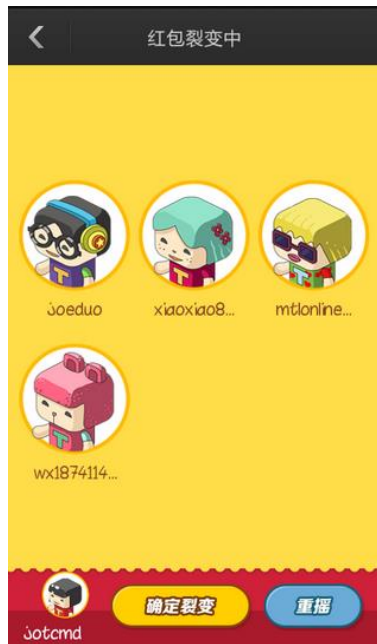
互动开放体系

Hilo引擎

互动创意大赛

# 互动营销案例







2013狂欢城



2014狂欢城



2015狂欢城





当互动游戏开始的时候，页面能自动切换出押宝PK模块

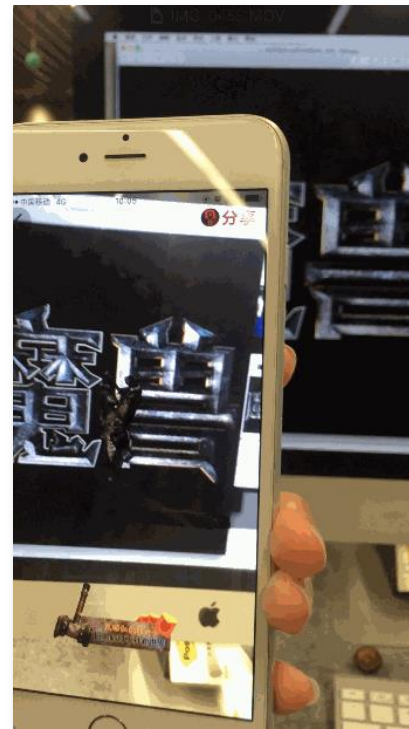
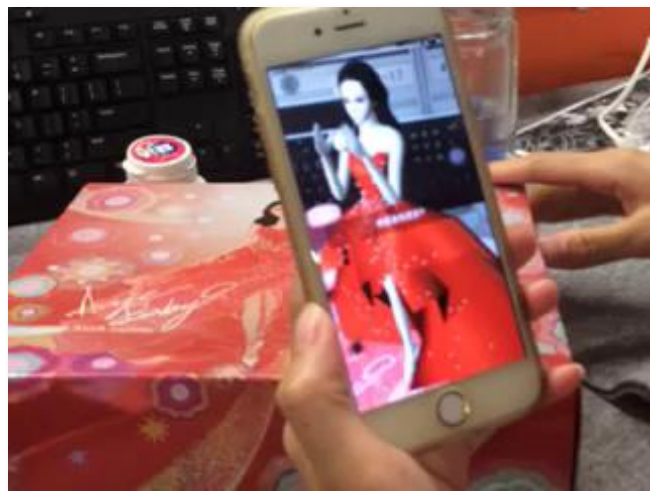


当明星表演的时候，页面能自动切换出对应的品牌模块

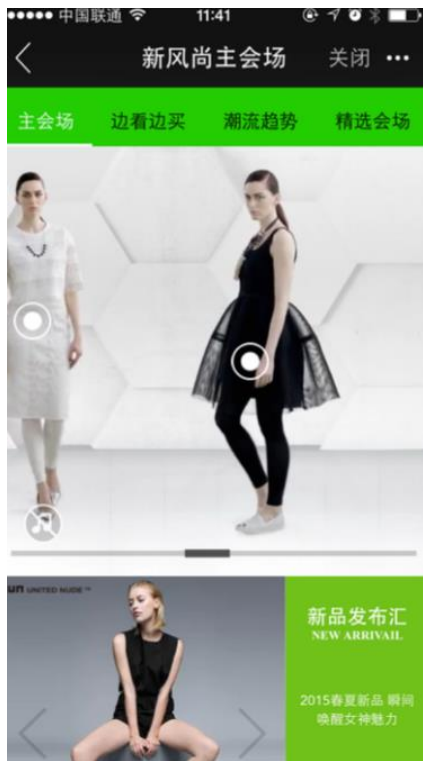


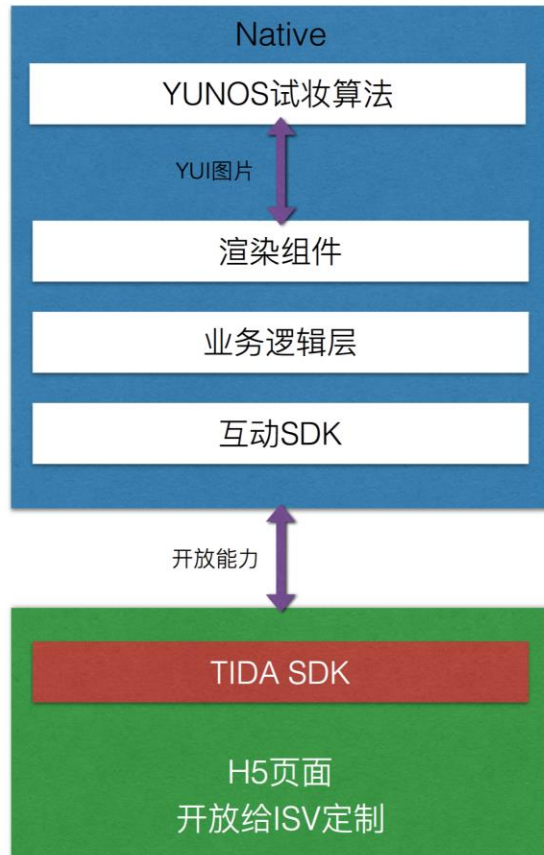
当播放广告的时候，页面能自动切换出广告模块，并与电视广告同步播放











阿里汽车

新车 独家首发  
不去现场也能360度看车  
下定抢超值大礼包

新美式旗舰CT6 精英男人尊享礼遇

- 1 精英男人礼  
订购CT6  
¥52000元
- 2 免费贷款  
¥12-18月
- 3 全家享乐  
家人同享CT6  
一年使用权!

点击查看更多细节

哈弗H7

【新车首发】哈弗H7

- 订购减2000元购车款



3D

49平米小户型绿色风格  
— 轻松体验3D环绕 —

美家特色，每日一更新

# 全民疯砍

**1212 品牌盛典**

**砍商品, 抢底价**  
爆品每5分钟一波

游小刃\*\*\*砍到ipdmini 游小刃\*\*\*砍到ipdmini 游小刃\*\*\*砍到ipdmini

**全民疯砍**

**MARUBI**  
丸美保湿系列五件套装  
底价 原价: 8888.00  
**¥888.00**

**BATTERY!**  
**GOODJOB! -5%**  
**PERFECT! -10%**

**MARUBI**  
丸美保湿系列五件套装  
底价 原价: 8888.00  
**¥888.00**

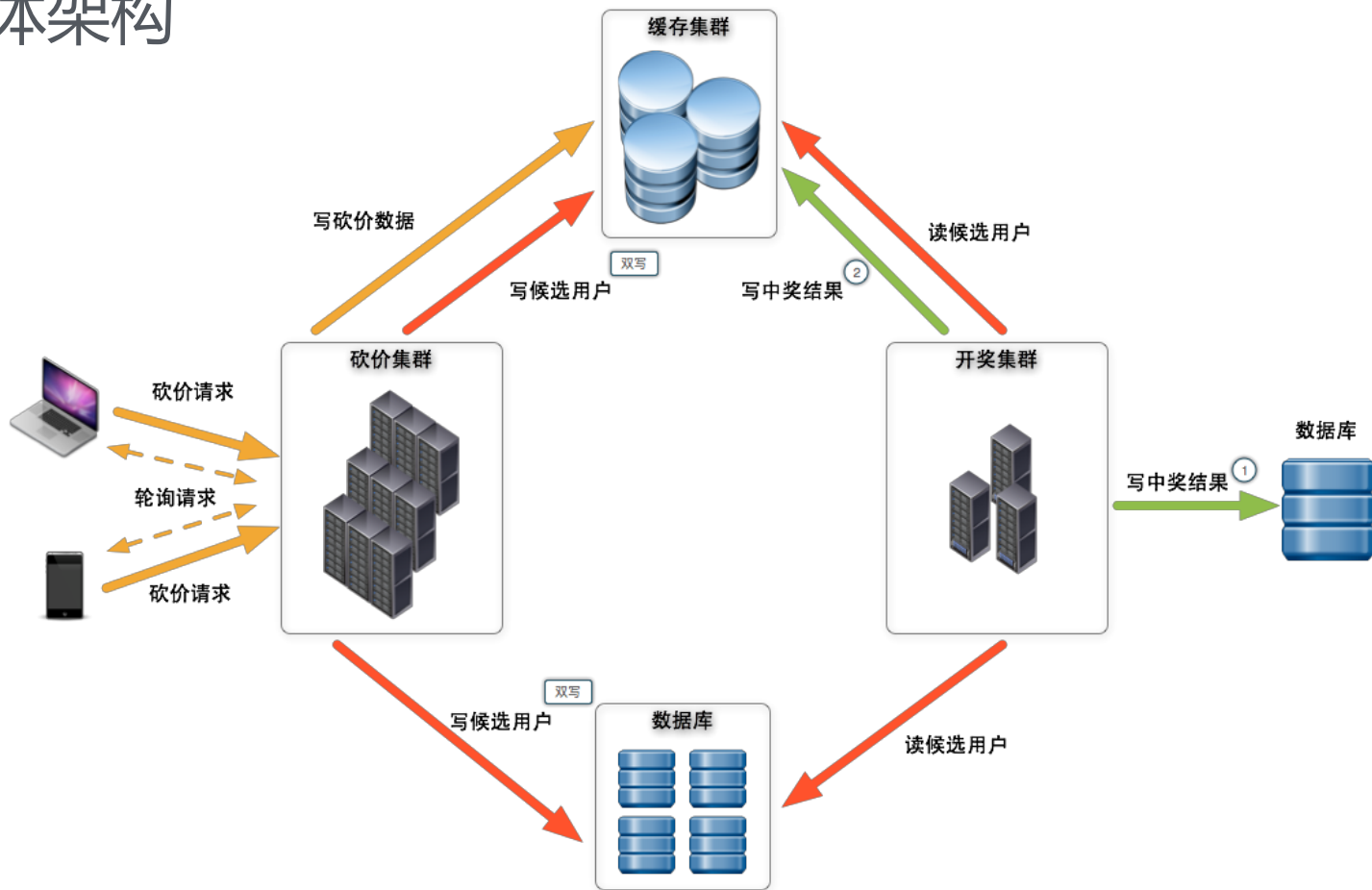
先砍完 就归你

现价: 6666.00

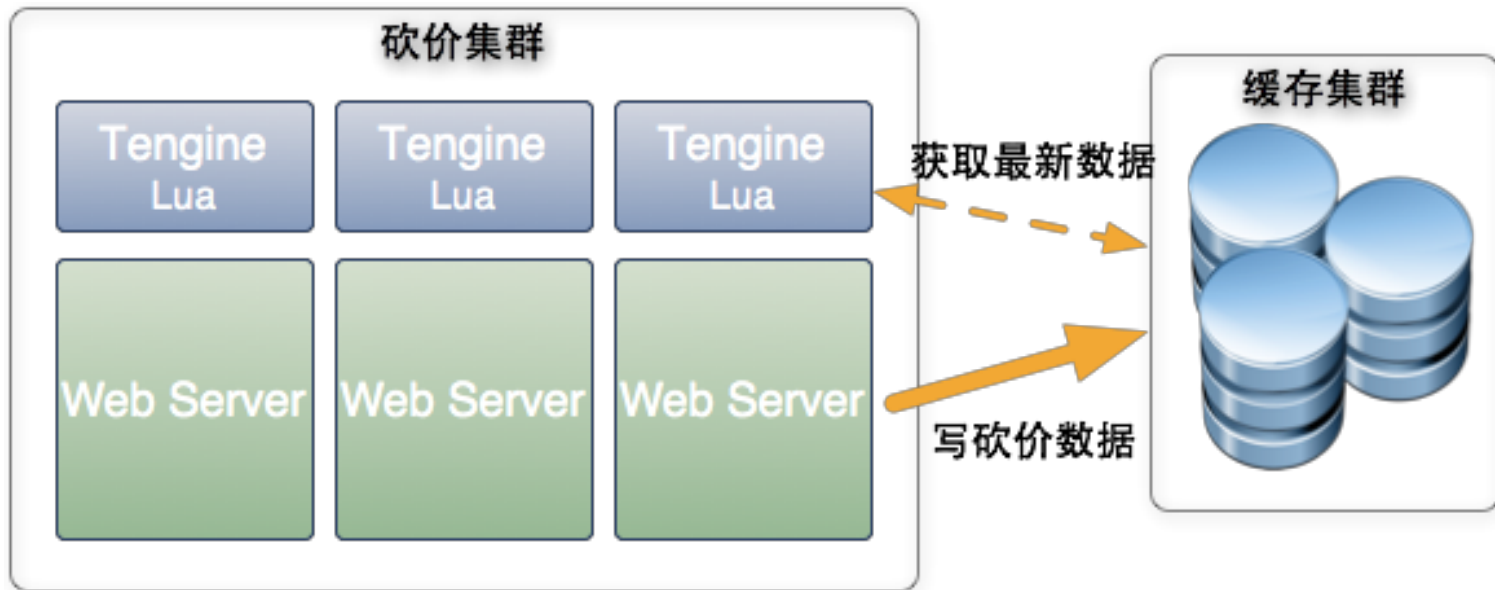
现价: 6666.00

**10:00 - 21:00 全民疯砍**

# 系统整体架构

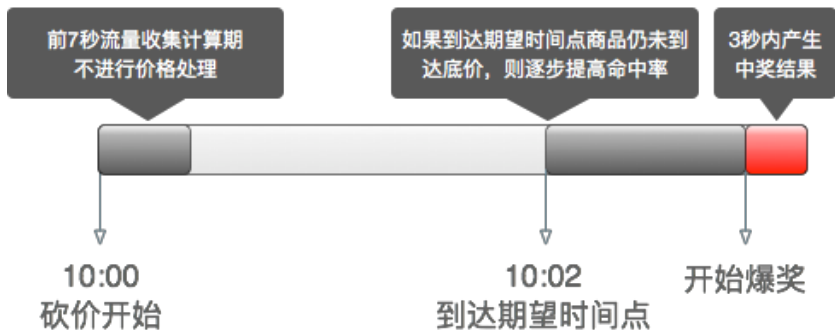


# 利用Tengine的能力

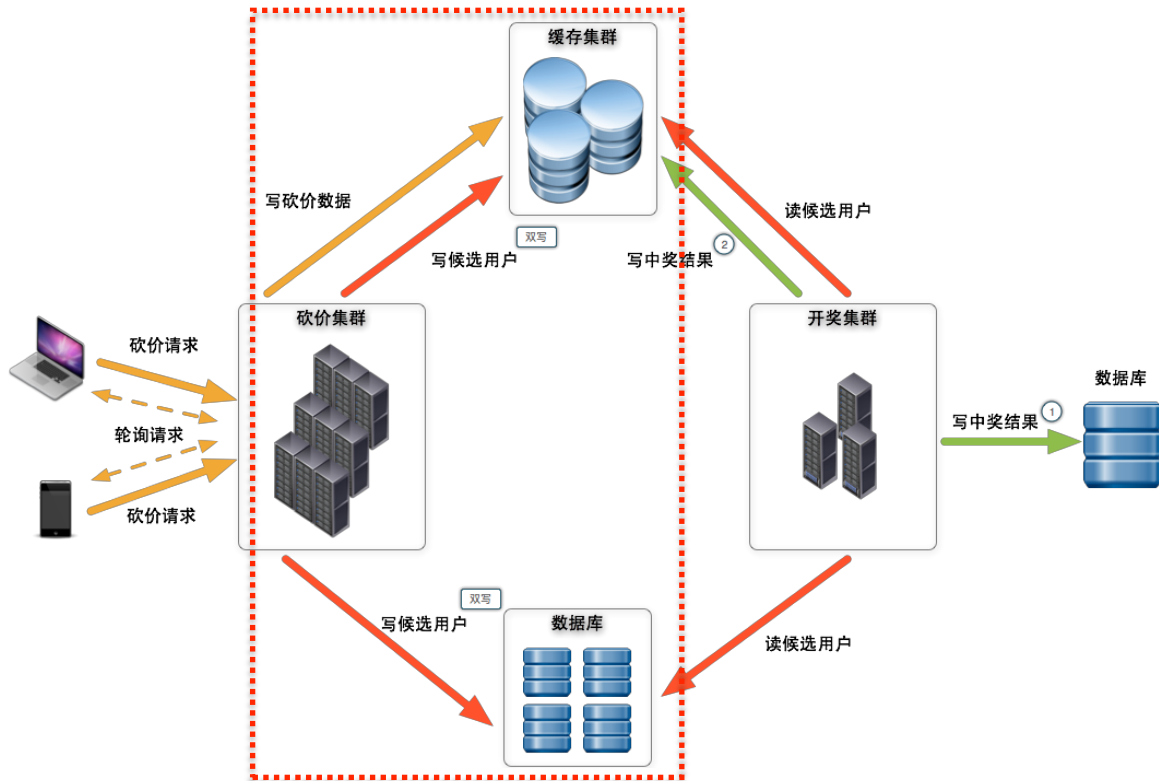


# 控制砍价时间

命中率 = (原价 - 底价) / (流量 \* 期望砍爆时间)  
例如:  $(9999 - 999) / (9000000 * 120) = 0.000008333$

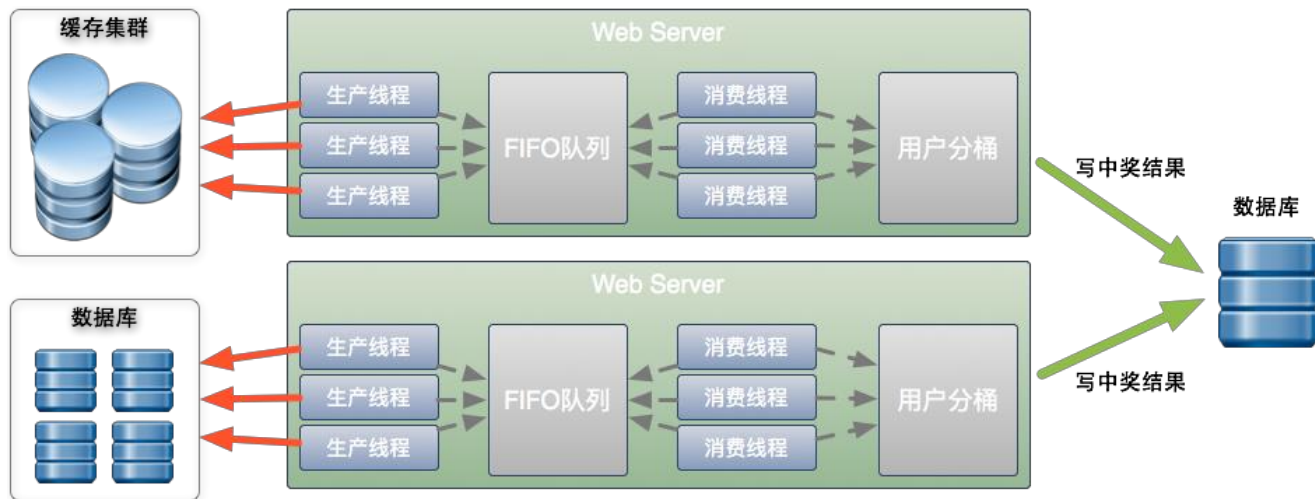


# 产生候选名单





# 集群内竞争开奖



# 赛车竞速赢汽车

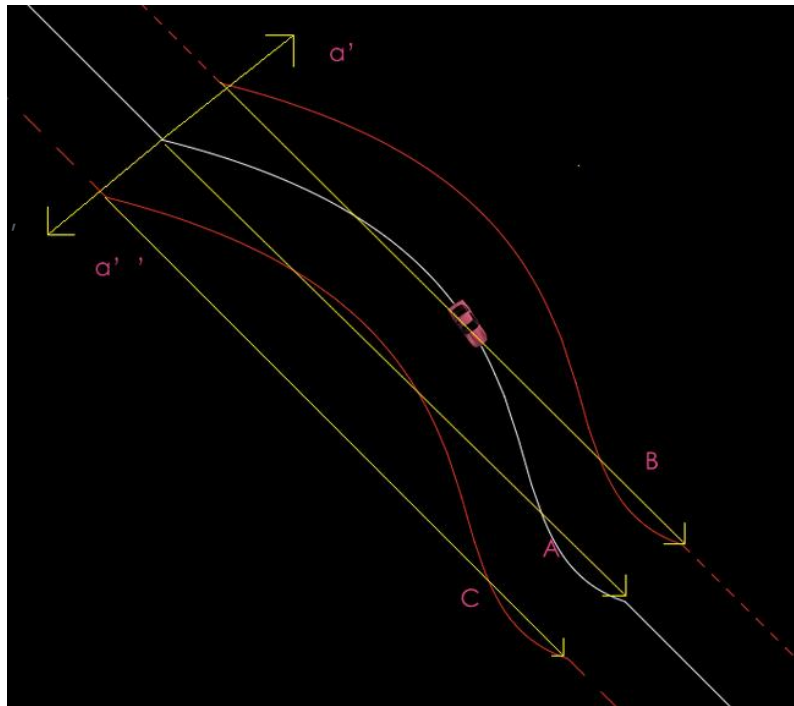




# 绘制赛道地图

## 直线和贝塞尔曲线组合

先画出白线，白线由直线和贝塞尔曲线组成，找到起始点，很容易画出  
第二步平移画出边线，直线和贝塞尔的平移方法类似，找到当前直线（贝塞尔）  
的起始点M，N，找到MN的垂直向量，通过MN，方向和距离（即赛道的宽度一半）  
算出另一个点即可



# 绘制赛道地图

## 直线和圆弧组合

### 优点

前端工程师和设计师更方便确定参数，使用圆弧只需要知道角度、半径，逆时针还是顺时针画圆

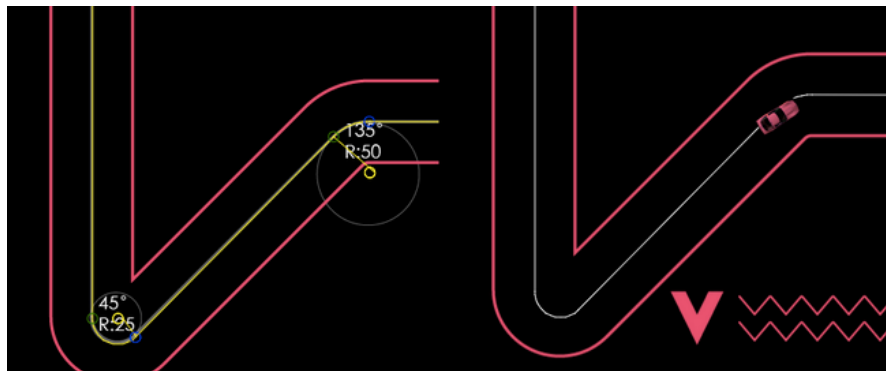
处理赛道和赛道叠加时候，不会出现贝塞尔曲线形变剧烈的情况，视觉感受上更容易接受

赛道的最后连接，使用圆弧切线相比贝塞尔曲线切线看起来更自然

### 缺点

直线+贝塞尔曲线的方式，我们可以在设计师设计阶段就导出地图的数据，但是圆弧需要记录的数据偏多，导出相对困难，而且如果日后需求改动，前端工程师无法对设计师完成的地图进行有效的数据校验

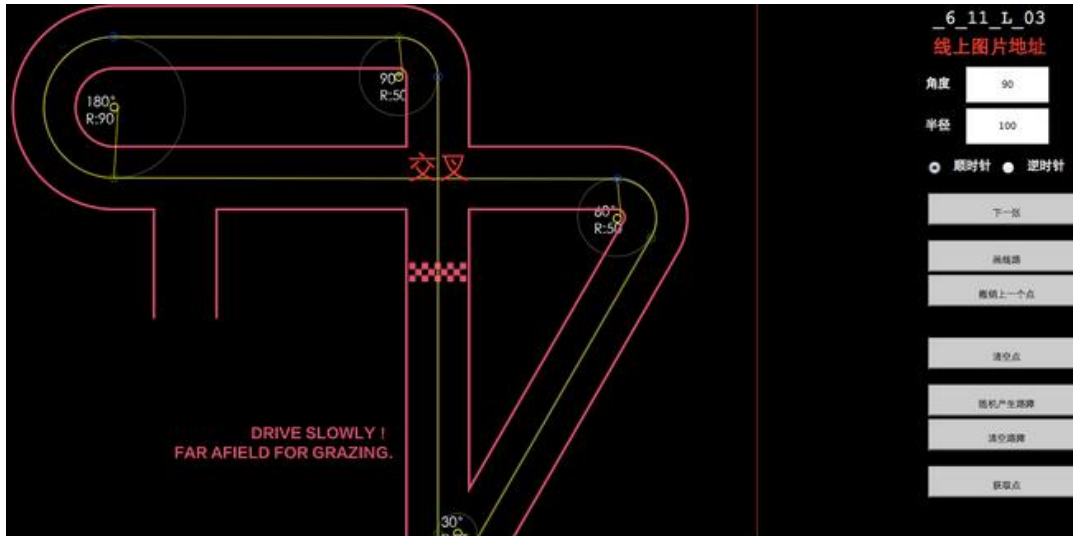
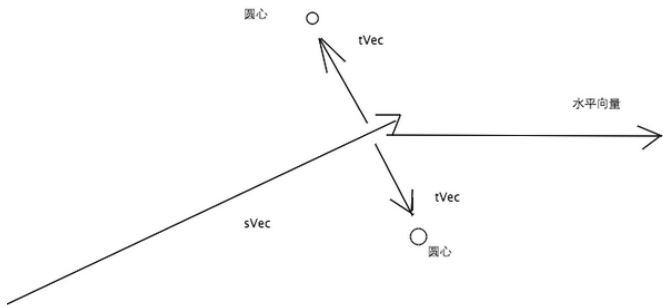
相比直线和圆弧组合，直接和贝塞尔曲线组合，画出的地图更丰富，变化更多



# 赛道地图编辑器

为了解决上面提到的缺点，我们写了一个赛道地图编辑器编辑器的主要作用就是完成地图的描点和相关的数据校验并生成前后端统一的数据格式然后导出

下图为圆弧生成的计算方法，根据当前直线，找到切线的圆心，Canvas的arc API在画线的时候需要确定开始角度和结束角度，因此，相对于顺时针圆弧，逆时针圆弧需要做一次角度转换



# 赛道地图编辑器

设计师在出图的时候，一张地图会出两张图，一张标注了圆弧角度，而另一张用于页面展示，相对于将数据和展现分离

将数据与展现分离，我们就可以随意替换页面展现风格，类似换肤的功能



# 防作弊策略 —— 服务端重放技术

## 数据收集和比对

在H5游戏过程中收集用户所有操作，并在游戏结束时将游戏过程中的所有操作和最终成绩上传给后端服务器

后端服务器根据用户的操作再结合正确的前端游戏逻辑，将用户操作作为输入，快速地将游戏过程重放一次得到一个正确的游戏成绩

将这个正确的成绩与用户上报的成绩作对比，如果不一致说明该用户篡改了游戏逻辑或者上报了虚假数据

## 压缩重放时间

验证过程中不再以绝对的时间比如小时、分钟、秒来计时，而是通过帧数来计时

游戏中1秒大概有60帧，前端只需要记录用户的操作和对应的帧数，比如用户在游戏的第100帧点击了屏幕

后端服务器来说，可以加速播放帧，比如可以1秒播放60000帧，相当于在看电影的时候按下了快进按钮，只是快进的速度为1000倍



# 防作弊策略 —— 随机路障

为了解决脚本录制的问题，我们必须在游戏过程中增加随机性，让操作不可预测  
为此，我们在赛道上设计了路障

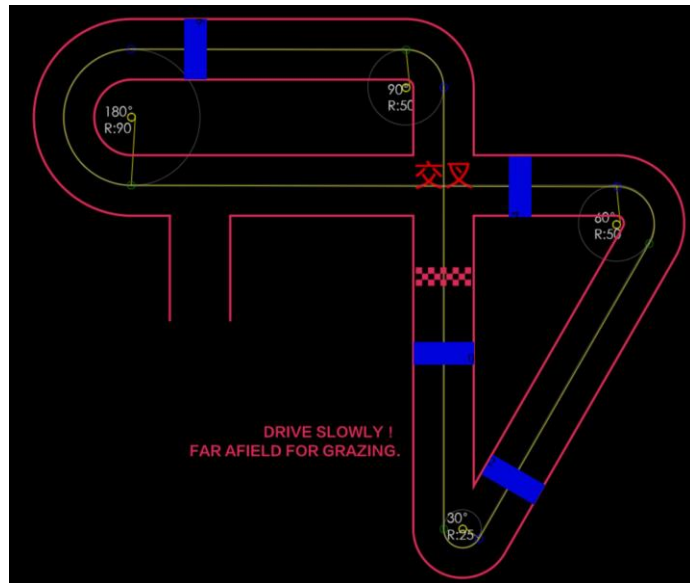
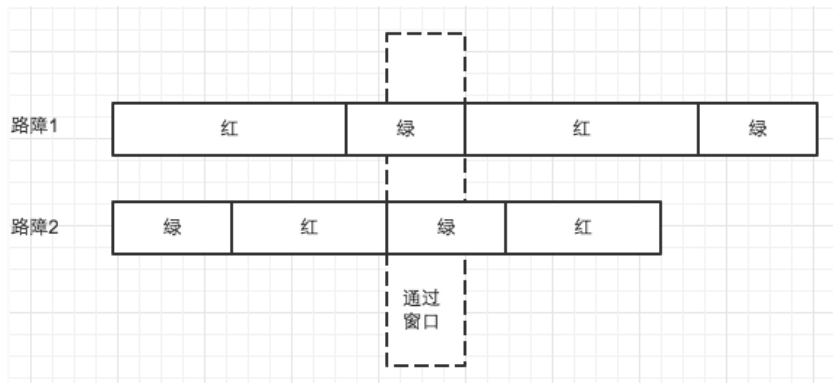
## 路障的设计原则

为了不影响视觉展示，路障不能在十字路口出现

路障由**服务端**生成并下发给前端，并且需要具备一定的散列性

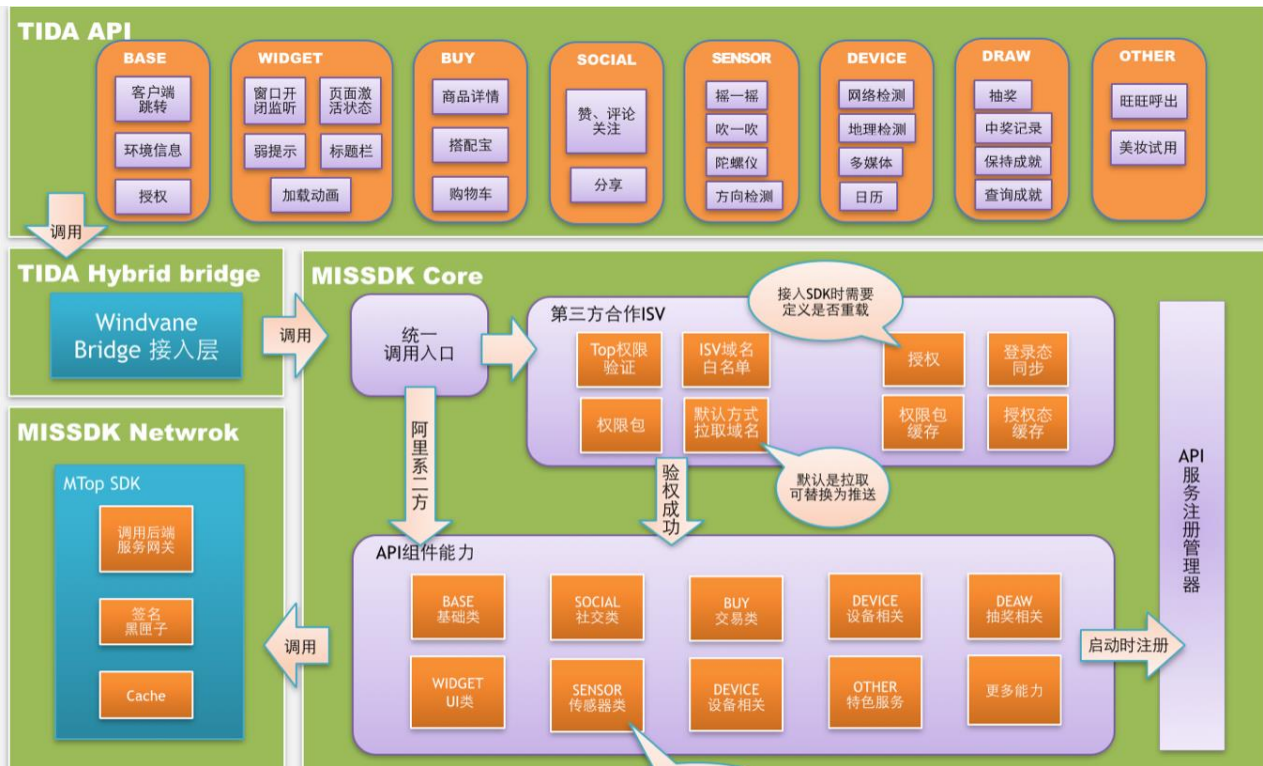
一个开闭周期内，通过窗口是可以调节的

**通过两个路障必须至少有一次操作（减速）**

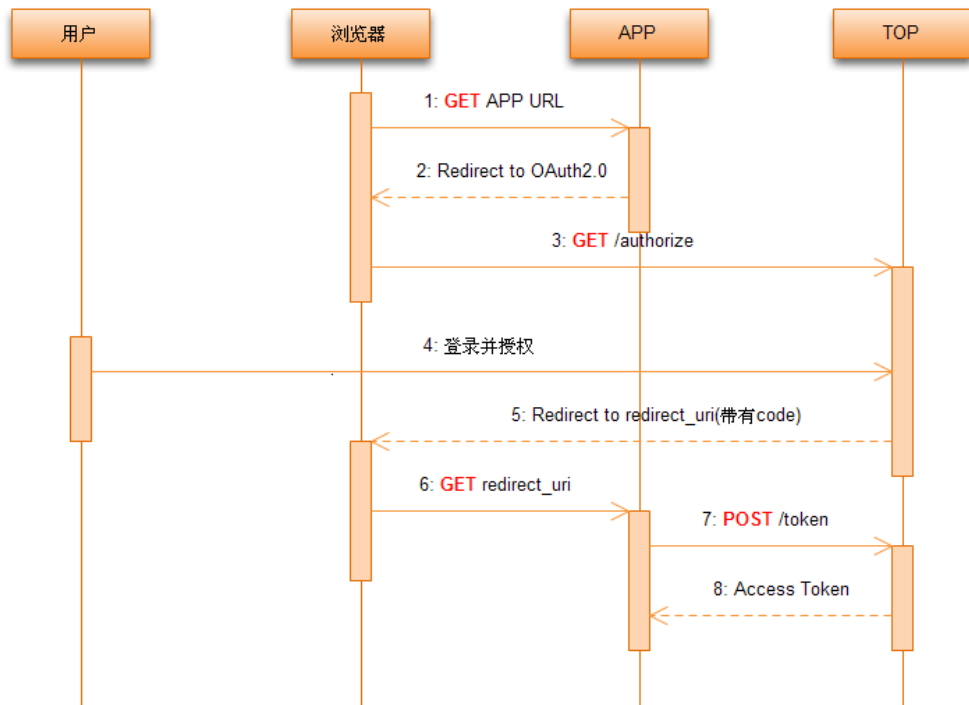


# 互动开放体系

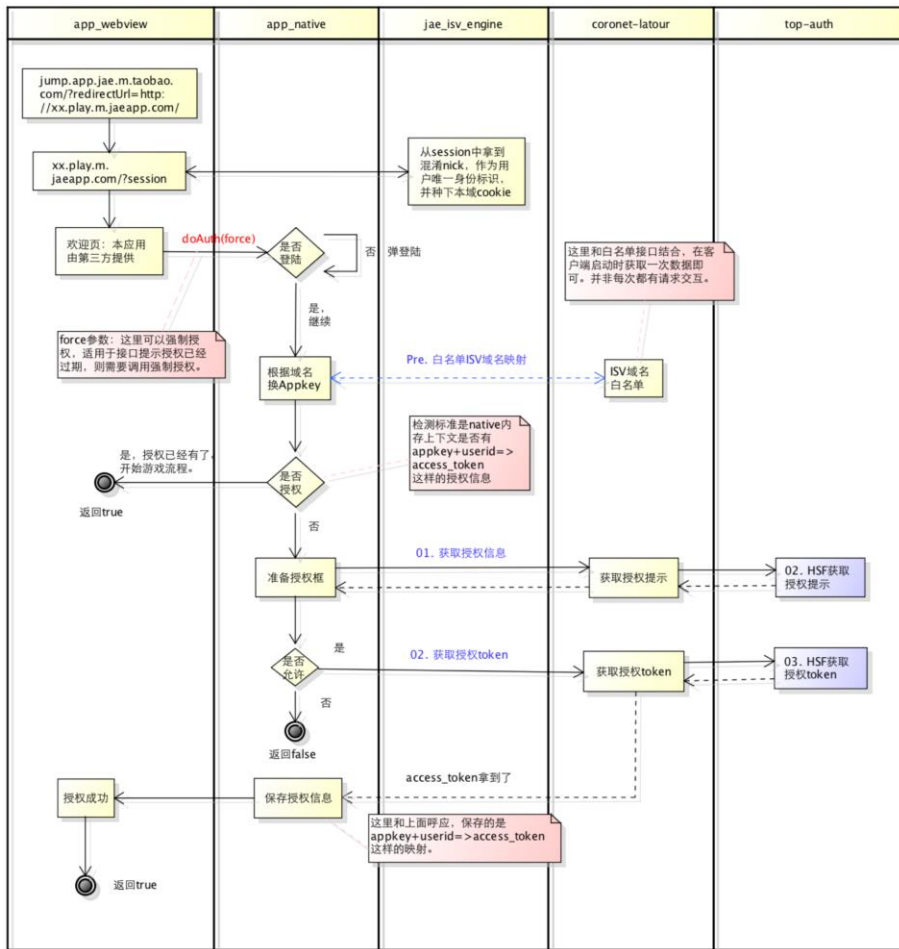
# 互动开放体系



# 传统授权方案



# 优化后的授权方案



```
/**
 * @param options 参数, 支持boolean型 function 或object 或者不传第1个参数 直接为function表示不强制授权
 * @param [options.refresh=false] 是否重新授权 true: 即使已经有授权, 会清掉之前的授权信息重启授权流程, false: 授权有效直接返回授权成功
 */
Tida.doAuth(options, function(data){
    if(data.finish){
        // 授权成功 可以顺利调用需要授权的接口了
    }else {
        // 未能成功授权
    }
});

// 另一种调用
Tida.doAuth(function(data){
    if(data.finish){
        // 授权成功 可以顺利调用需要授权的接口了
    }else {
        // 未能成功授权
    }
});

// 强制授权 每次都会弹出授权框
Tida.doAuth(true, function(data){
    if(data.finish){
        // 授权成功 可以顺利调用需要授权的接口了
    }else {
        // 未能成功授权
    }
});
```

data 格式:

```
{
  errorCode: 0, //通用错误码
  errorMessage: "用户已授权",
  finish: true //true为授权成功 false授权失败 授权失败请参考data.errorCode
}
```



# Tida SDK

## 监听摇一摇

```
var watchId = Tida.shake.watch({
  sensitivity: 20, // 振动幅度, 加速度变化超过这个值后触发shake
  frequency: 150, // 采样间隔(毫秒), 指每隔多长时间对加速度进行一次采样, 然后对比前后变化, 判断是否触发shake
  callbackDelay: 3000 // 触发『摇一摇』后的等待时间(毫秒), 防止频繁调用
}, function (data) {
  if (data.errorCode) {
    console.error(data.errorCode, data.errorMessage);
  } else {
    // errorCode为0 摇一摇成功触发
    alert('Shaked!');
  }
});
```

<http://g.alicdn.com/tmapp/tida-doc/docs/part1/index.html>

## 监听重力传感器

```
var watchId = Tida.motion.watch({}, function (data) {
  if (data.errorCode) {
    alert('error');
  } else {
    var acc = data.accelerationIncludingGravity;
    // x、y、z对应各方向上的加速度
    console.log(acc.x, acc.y, acc.z);

    var acc2 = data.acceleration;
    // x、y、z不包含重力加速度
    console.log(acc2.x, acc2.y, acc2.z);
  }
});
```



# 电商互动游戏引擎 —— Hilo

<https://github.com/hiloteam/Hilo>

# Hilo引擎的特点

## 极简内核

Hilo核心模块极精简，保留了2D游戏引擎最必要的模块，同时采用模块化管理

## 完善接入&扩展

Hilo 支持多种模块范式的包装版本，包括AMD，CMD，Standalone多种方式接入。另外，你可以新增和扩展需要的模块和类型

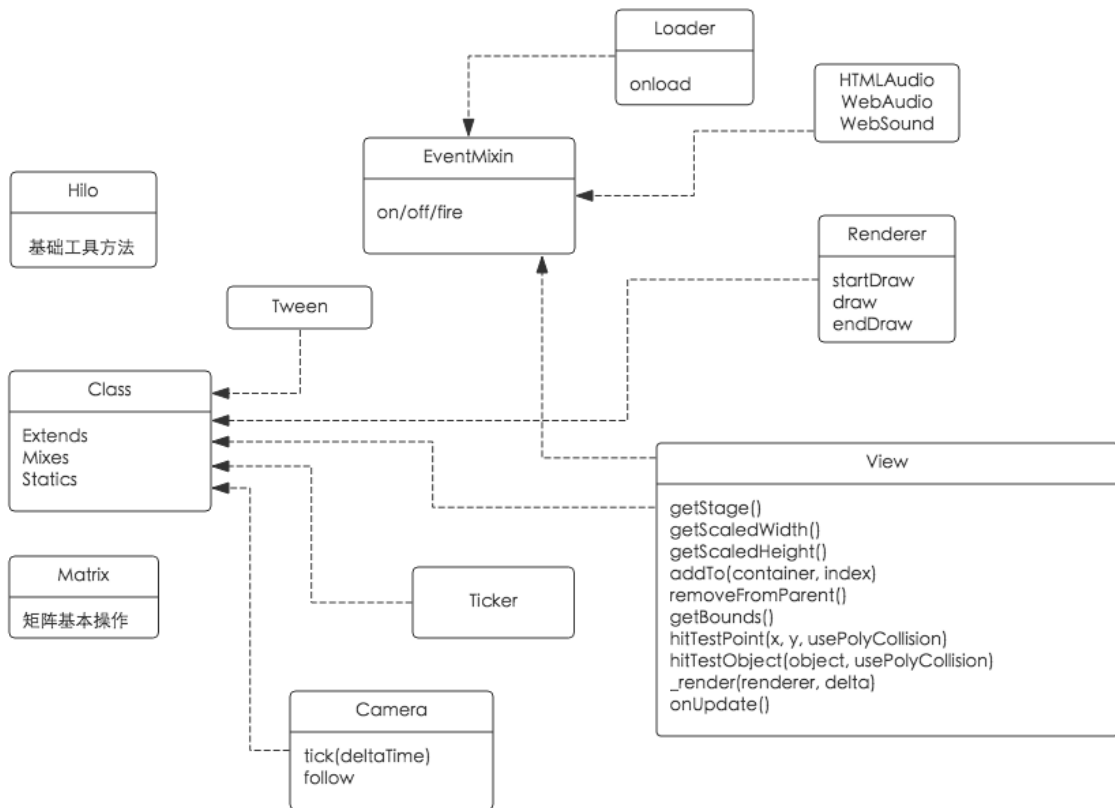
## 多种渲染方式

提供DOM，Canvas，Flash，WebGL等多种渲染方案，可以做到跨全端，高性能的要求

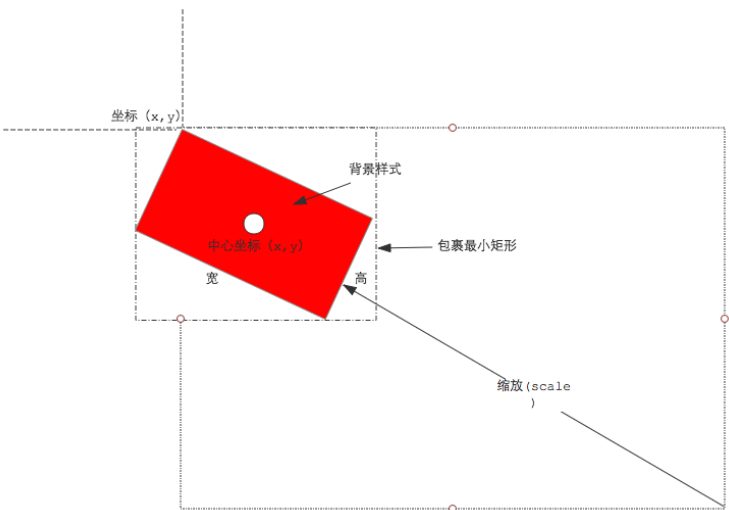
## 完善的周边工具

提供动画编辑器，Yeoman脚手架及典型案例产出的辅助开发工具

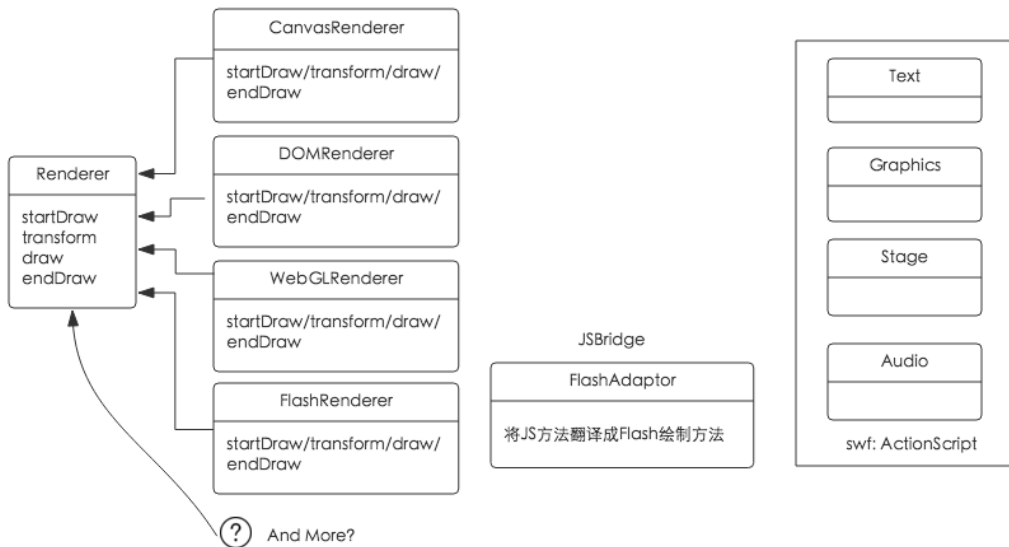
# Hilo整体结构



# Hilo的多渲染方式



```
render: function(renderer, delta){  
    //不同的View renderer方式可以多样, 相同的View也可以根据renderer的类型不同而呈现差异化  
    ...  
}
```



# 如何接入Hilo

## 传统模块定义范式

```
define(function(require, exports, module){  
  var a = require('a'),  
      b = require('b');  
  
  //something code here  
  return someModule;  
});
```

## 多种模块范式版本

```
hilo/  
├── build/  
│   ├── standalone/  
│   ├── amd/  
│   ├── commonjs/  
│   ├── kissy/  
│   └── cmd/
```

## Hilo模块定义范式

```
/**  
 * @module hilo/view/Sprite  
 * @requires hilo/core/Hilo  
 * @requires hilo/core/Class  
 * @requires hilo/view/View  
 * @requires hilo/view/Drawable  
 */
```



```
define(function(require, exports, module){  
  var Hilo = require('hilo/core/Hilo');  
  var Class = require('hilo/core/Class');  
  var View = require('hilo/view/View');  
  var Drawable = require('hilo/view/Drawable');  
  
  //some code here  
  return Sprite;  
});
```

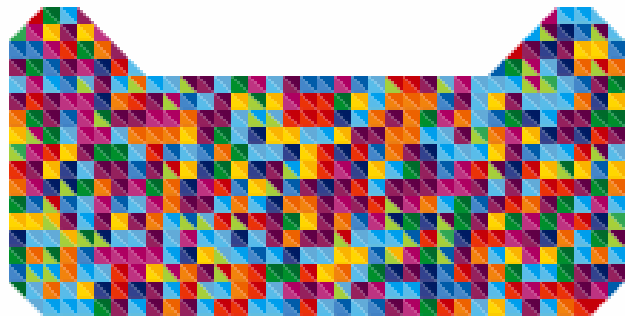
# 扩展Hilo

```
var SomeClass = Class.create({  
  Extends: ParentClass,  
  Mixes: SomeMixin,  
  Statics: SomeStatics,  
  constructor: Constructor,  
  
  propertyName: propertyValue,  
  methodName: methodValue  
});
```

使用 Class.mix(target, [mixinObject])方法

```
var EventMixin = {  
  on: function(type, handler){ },  
  off: function(type, handler){ },  
  fire: function(type, detail){ }  
}  
  
Class.mix(object, EventMixin);
```

# Hilo实现粒子特效



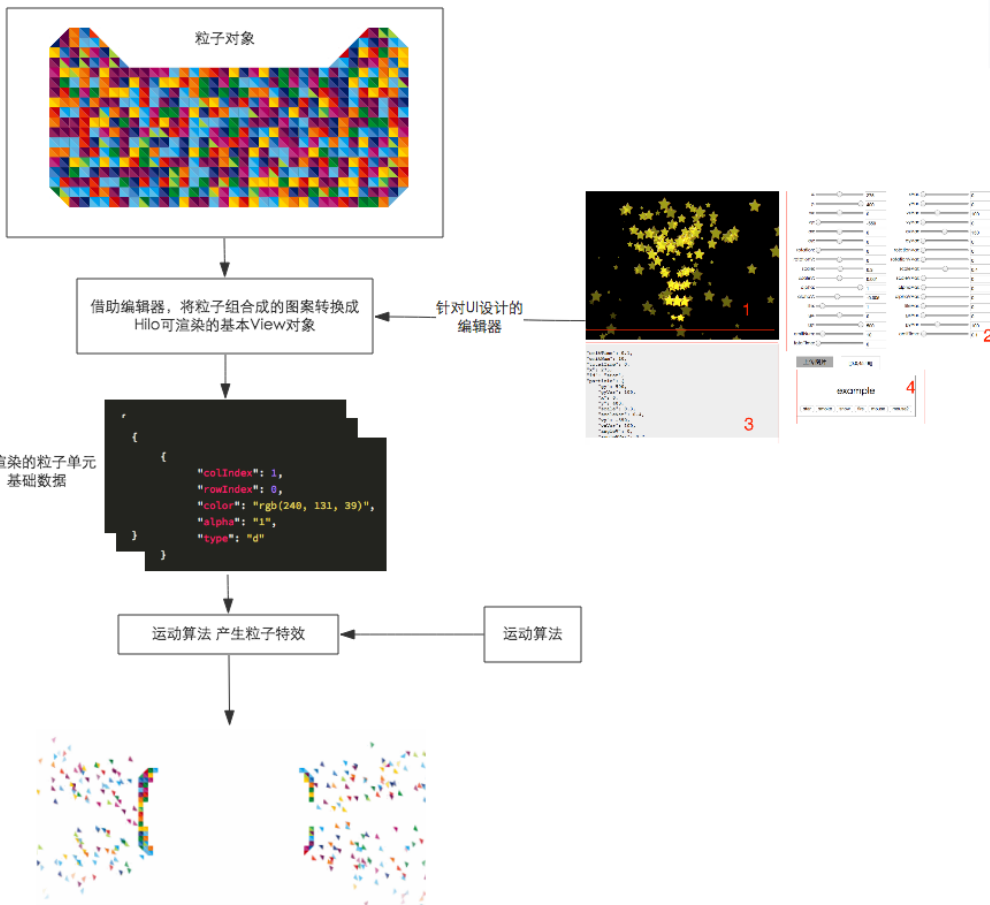
# Hilo的粒子特效

```
onUpdate = function(){
  if(this.isStart){
    this.x += this.vx;
    this.y += this.vy;
    this.rotation += Math.random() * 10;
    this.alpha -= Math.random() * 0.1;
    this.scaleX -= Math.random() * 0.01;
    this.scaleY -= Math.random() * 0.01;

    ...
  }
}
```

```
var distance = Math.sqrt((p.x - centerX)*(p.x - centerX) + (p.y - centerY)*(p.y - centerY))
var delay = ( distance)* 5; //延迟时间和距离有关
```

```
p.close = function(){
  p.isStart = false;
  Hilo.Tween.to(p, {
    x:p.target.x, //缓动到开合前的状态
    y:p.target.y,
    rotation:p.target.rotation,
    alpha:1,
    scaleX:1,
    scaleY:1
  },{
    ease:Hilo.Ease.Bounce.EaseInOut,
    duration:1000,
    delay:p.target.delay
  })
};
```

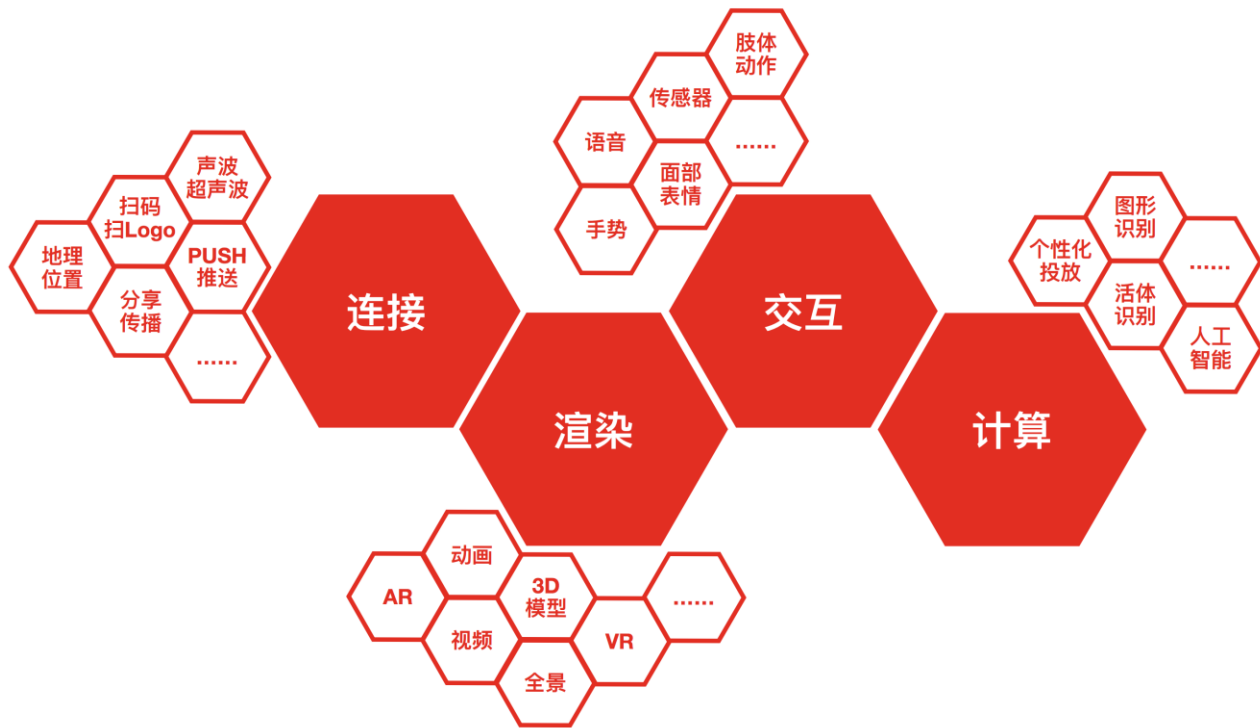




# Hilo实现补间动画



# 互动创意大赛





天猫技术  
TMALL TECHNOLOGY

# 天猫喵喵

## 互动创意大赛

创意独特 技术出众 携手共赢 | 2016.7.20-2016.8.31

我要参赛

<http://hudong.tmall.com>

The background of the slide is a deep space scene. In the center, there is a large, circular nebula with a dense core of bright orange and red light, surrounded by a diffuse cloud of blue and purple particles. The rest of the background is a dark, starry field with various celestial objects, including a prominent bright blue star in the lower right and some faint, wispy structures in the upper left.

Thank you !