高德Inside

LBS技术解析与游戏结合实践

高德开放平台 资深研发经理

朴春植（帕科）

开放的高德

30万应用的共同选择

中国技术领先的地图LBS服务提供商

lbs.amap.com

即日起，注册成为高德开发者
就能免费使用域名、虚拟主机、企业邮箱服务1年。

lbs.amap.com

高德开放平台 lbs.amap.com　阿里云

高德用户专享

0元享阿里云产品使

域名 | 虚拟主机 | 企业邮

定 位

地图游戏结合

经验分享

定　位

中国大陆的定位方式及注意事项

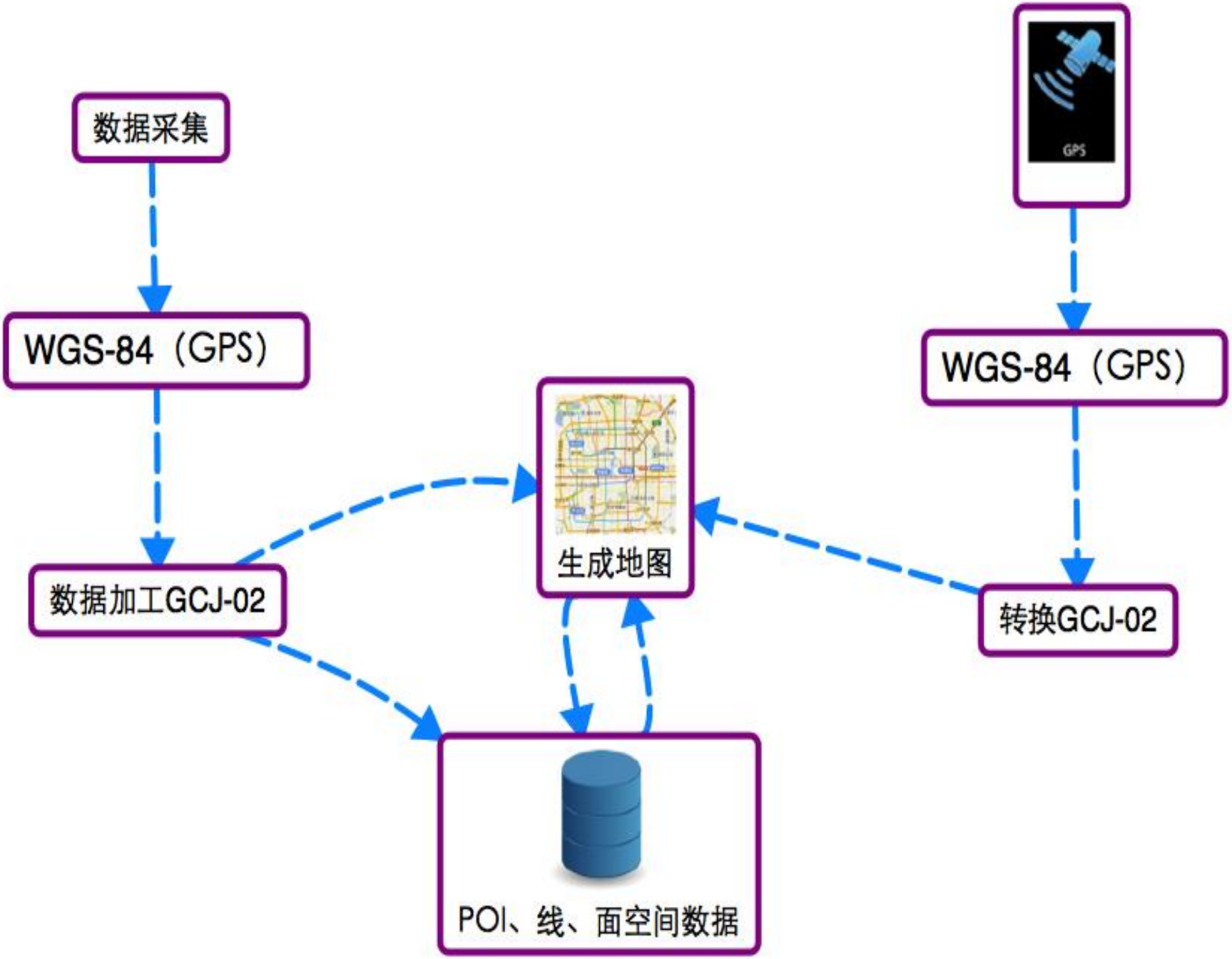
WGS-84坐标系、GCJ-02坐标加密

WGS-84坐标系

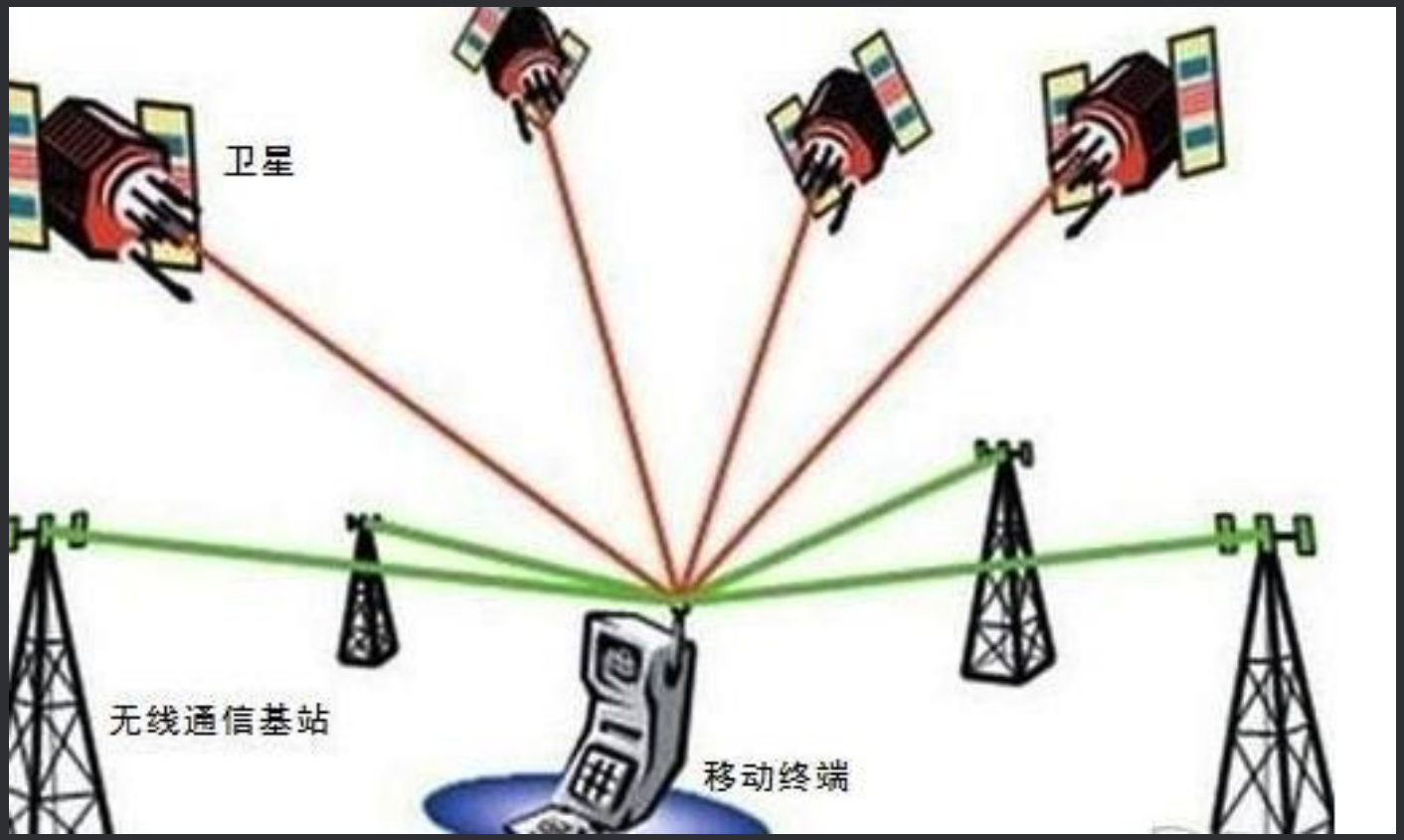
WGS84:World Geodetic System 1984，是为GPS全球定位系统使用而建立的坐标系统。

国测局的坐标加密(GCJ-02):

GCJ-02是由中国国家测绘局（国测局）制订的地理信息坐标系统。他是由一种对经纬度数据的加密算法，并加入了随即偏差。国内出版的各种地图系统（包括电子形式），必须至少采用GCJ-02对地理位置进行首次加密。


数据采集
WGS-84（GPS）
数据加工GCJ-02
GPS
WGS-84（GPS）
转换GCJ-02
生成地图
POI、线、面空间数据

定位 GPS、Wifi、基站、室内



GPS、Wifi、基站

分类	类型	精度	特点	调用方式
GPS	GPS/A*GPS	2-10m	准确、耗时长、耗电	定位SDK+系统接口
网络	基站	500-5000m	大致位置	定位SDK
	Wifi	20-200m	较为准确快速省电	定位SDK

定位 耗电量优化经验

选择适合产品需求的定位模式

减少定位频次（按时间、距离）

高德Inside Game

游戏和LBS如何结合？

Pokemon Go在中国？

LBS与游戏结合的需求及痛点

地图底图的自由定制化程度如何？

- 地图颜色是否可以可以改变？
- 楼快及文字是否能消失？

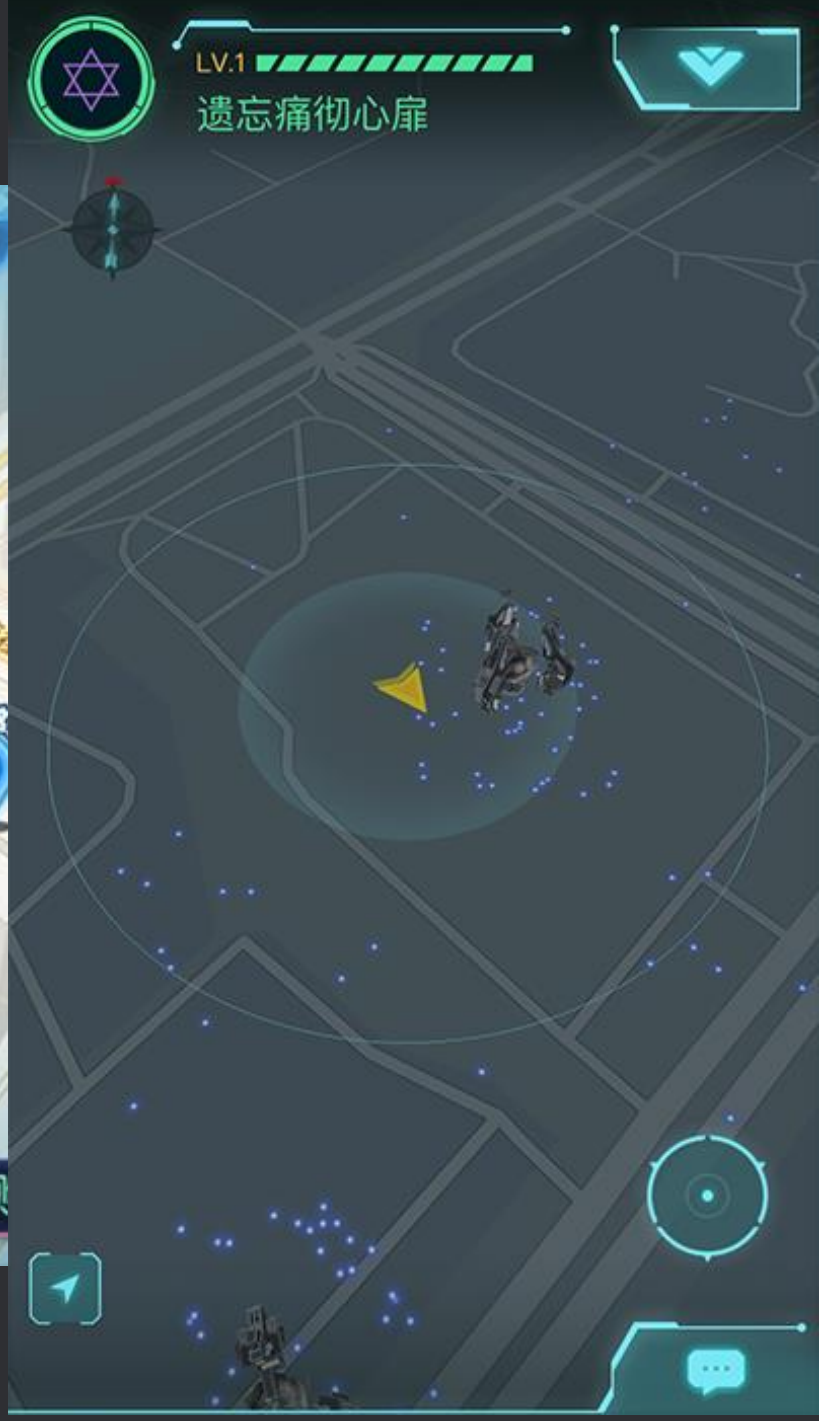
地图上如何实现游戏场景？

- 游戏元素如何叠加在地图上？
- 如何安排怪物，任务和副本？

渲染效率如何提升？

- 如何提升渲染效率，使千元安卓机流畅运行？
- 何时支持Unity3D和Cocos2D等游戏引擎？


任务
10级解锁
30500
30
商店
好快保
望京西园4区
夏都盈座
望京
C区
北京柠檬树
家庭旅馆
广顺北大街
B1
北京祥龙公
交客运有限
公司枢纽站
A1
广顺南大街
北
传输机
定位
收藏
北京市-新一城

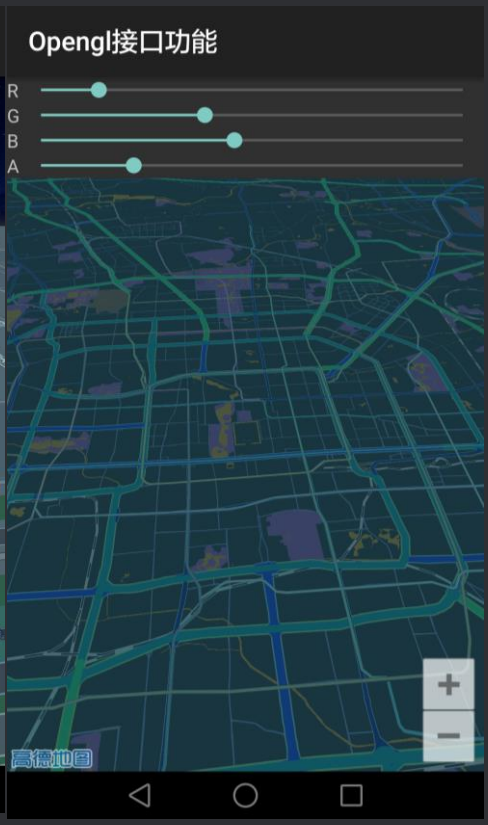

LV.1
遗忘痛彻心扉

自定义地图样式

普通

导航

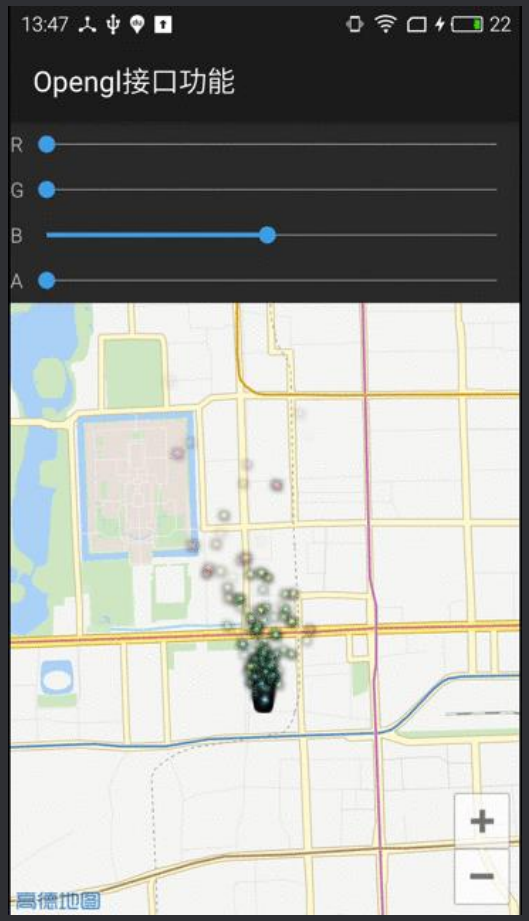
夜景

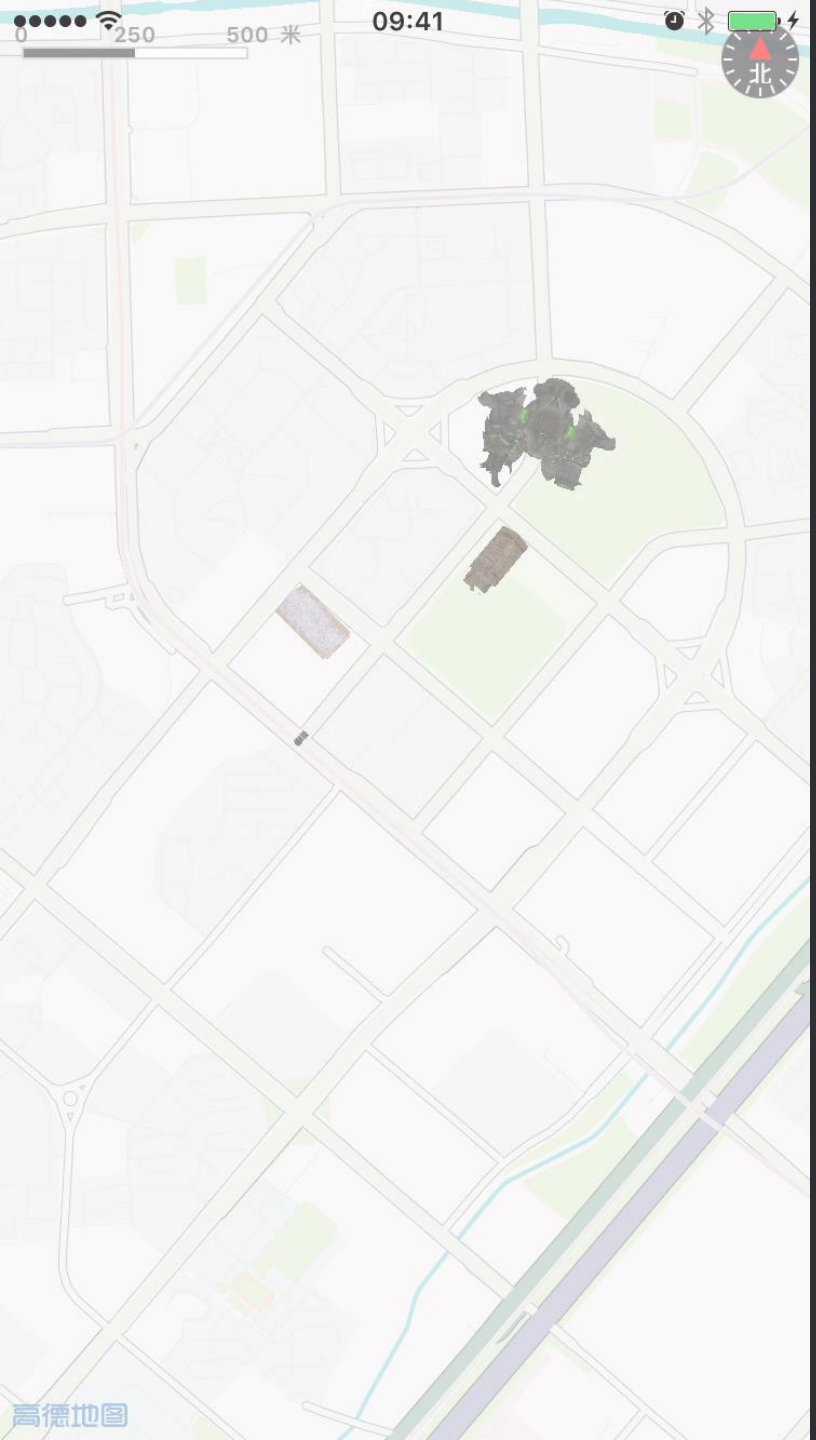


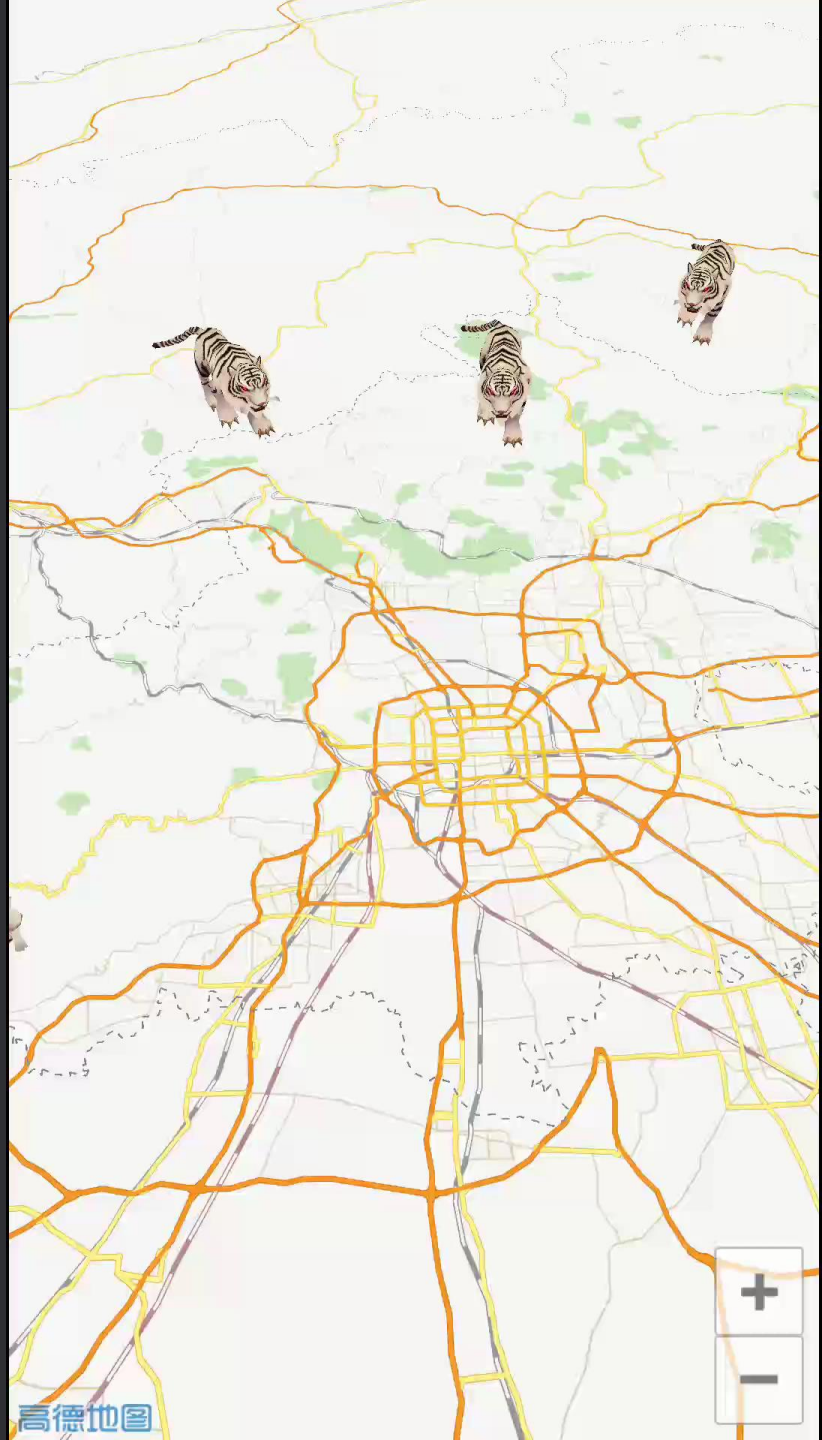
自定义

加载自定义模型—加载

用户可以加载自己的Overlay模型，在进行openGL绘制时，对模型Overlay进行平移（translate）、旋转（rotate）、缩放（scale）操作，可以直接简单地使用openGL接口，不需要自己建立矩阵。




09:41
0
250
500 米
北
高德地图


高德地图

主要技术原理：

1、地图开放每帧 Render 回调

2、开放 屏幕、经纬度、opengl坐标 互转

- 地图开放每帧回调及坐标转换

```
class MapRenderer implements CustomRenderer{
    TigerModel tiger;
    AMap aMap;
    ...
    @Override
    public void onDrawFrame(GL10 gl) {
        // 地图开放每帧Render 回调
        // 绘制模型
        tiger.draw();
    }

    @Override
    public void OnMapReferencechanged() {
        // 地图放到到一定级别 重新计算缩放比例
        // 开放屏幕、经纬度和OPENGL坐标互转
        aMap.getProjection().fromScreenLocation(screenPos);//屏幕坐标转经纬度
        aMap.getProjection().toScreenLocation(mapLatlng);//经纬度转屏幕坐标
        aMap.getProjection().toOpenGLLocation(mapLatlng);//经纬度转OPENGL坐标
        aMap.getProjection().toOpenGLWidth(screenWidth);//幕宽度转OPENGL宽度
    }
}
```

视频对应github地址，感兴趣的可以去下载

https://github.com/amapapi/Android_Map_Game.git

https://github.com/amapapi/iOS_Map_Game.git

即将发布

全新地图引擎上线

1、支持 opengl es 2.0

2、CPU 内存 消耗更低

3、自定义更佳灵活

……

高德Inside

开放平台 经验分享

什么事情是用户最关心的？

- ✓ 使用简单，API设计
- ✓ 稳定，崩溃率
- ✓ 响应速度，快
- ✓ 包大小，越小越好
- ✓ 隐私安全性

API 设计-场景

适配多元化应用场景

void	searchPOIAsyn() 查询POI异步接口。
PoiItem	searchPOIId(java.lang.String poiID) 已知poiid信息（点击地图底图），搜索POI的详细信息，同步

void	**setMockEnable**(boolean isMockEnable) 设置是否允许模拟位置, 默认为false 模式为低功耗模式(Battery_Saving)时无效

API 设计-扩展性

1.X版本

```
Locationmanager.requestLocationData(“lbs”, 2000, 10, listener);
```

2.X版本

```
//设置定位参数
aMapLocationClient.setLocationOption(mLocationOption);
//启动定位
aMapLocationClient.startLocation();
```

AMapLocationClientOption	**setGpsFirst**(boolean isGpsFirst) 设置是否优先返回GPS定位信息 默认值：false **只有在高精度定位模式下有效**
void	**setHttpTimeOut**(long httpTimeOut) 设置联网超时时间 单位：毫秒 默认值：30000毫秒 模式为**仅设备模式(Device_Sensors)**时无效
AMapLocationClientOption	**setInterval**(long interval) 设置发起定位请求的时间间隔 单位：毫秒 默认值：2000毫秒
AMapLocationClientOption	**setKillProcess**(boolean isKillProcess) 设置退出时是否杀死service 默认值:false, 不杀死 模式为**仅设备模式(Device_Sensors)**时无效
void	**setLocationCacheEnable**(boolean isLocationCacheEnable) 设置是否使用缓存策略, 默认为true 使用缓存策略
AMapLocationClientOption	**setLocationMode**(**AMapLocationClientOption.AMapLocationMode** locationMode) 设置定位模式
static void	**setLocationProtocol**(**AMapLocationClientOption.AMapLocationProtocol** amapLocationProtocol) 设置定位协议
void	**setMockEnable**(boolean isMockEnable) 设置是否允许模拟位置, 默认为false 模式为**低功耗模式(Battery_Saving)**时无效
AMapLocationClientOption	**setNeedAddress**(boolean isNeedAddress) 设置是否返回地址信息，默认返回地址信息 默认值：true, 返回地址信息 模式为**仅设备模式(Device_Sensors)**时无效
AMapLocationClientOption	**setOnceLocation**(boolean isOnceLocation) 设置是否只定位一次 默认值：false
void	**setOnceLocationLatest**(boolean isOnceLocationLatest) 设置单次定位是否等待设备AP源刷新 仅适用于单次定位，当设置为true时，会自动变为单次定位
void	**setWifiActiveScan**(boolean isWifiActiveScan) 设置是否主动刷新WIFI 默认值：true 主动刷新 默认值： 模式为**仅设备模式(Device_Sensors)**时无效

稳定性

1、合理的架构

2、代码扫描 Review机制

3、自动化

开放平台SDK演进

Mobile SDK

Android
地图SDK

iOS
地图SDK

- 2D地图SDK
- 3D地图SDK
- 搜索SDK
- 导航SDK
- 定位SDK

开放平台 SDK架构

SDK产品

定位 | 地图 | 搜索

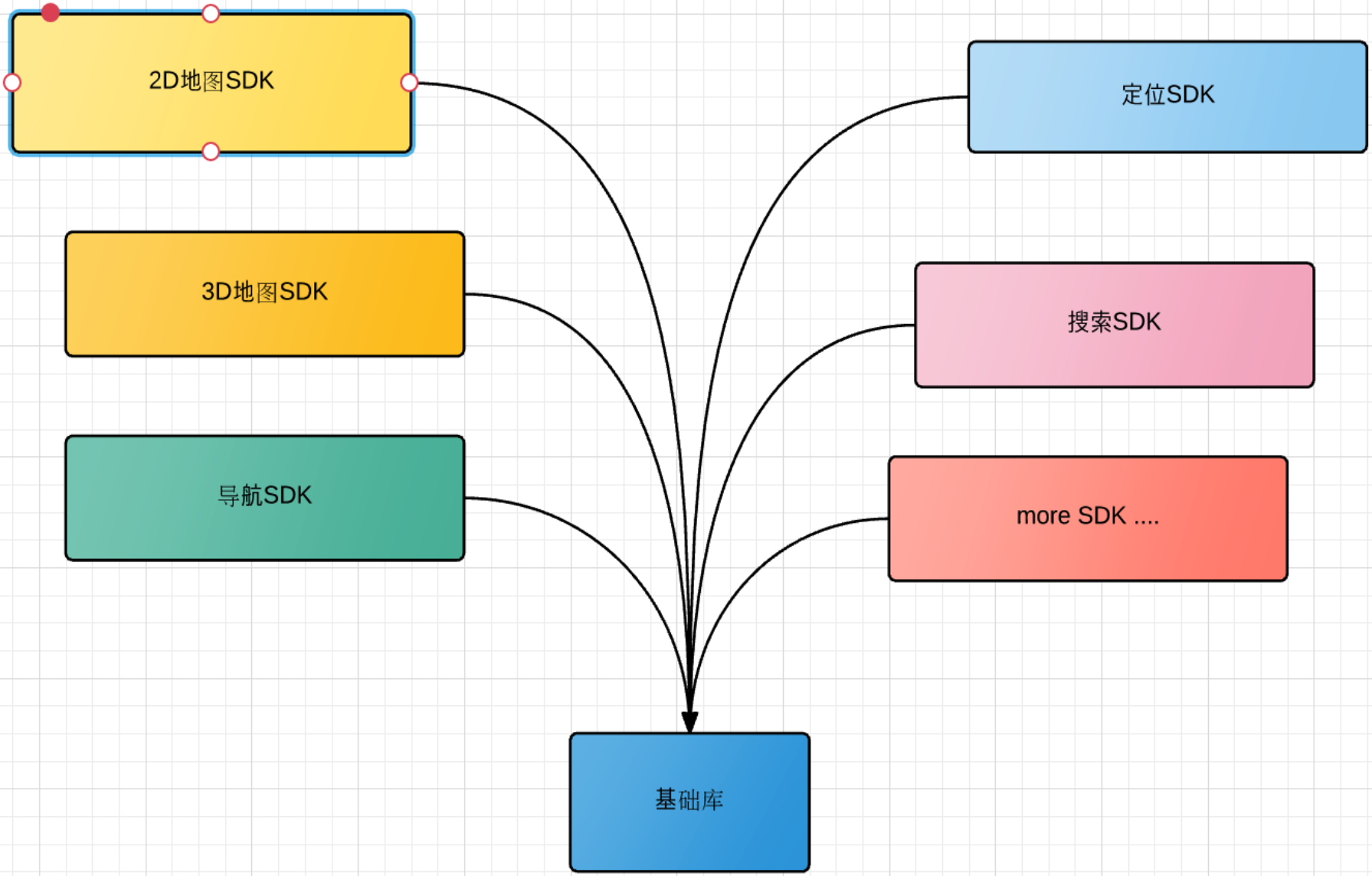
导航 | 室内 | …

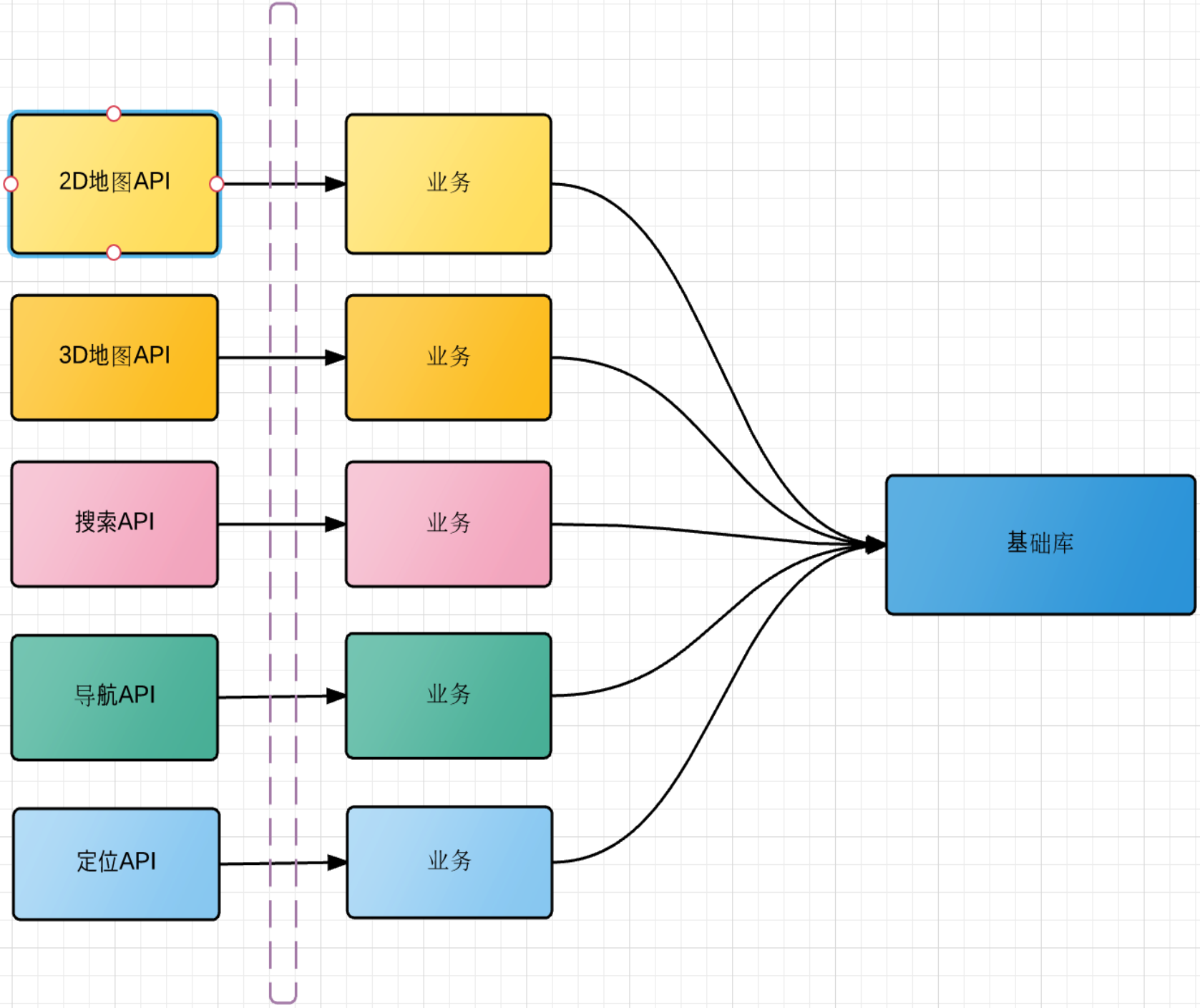
基础库

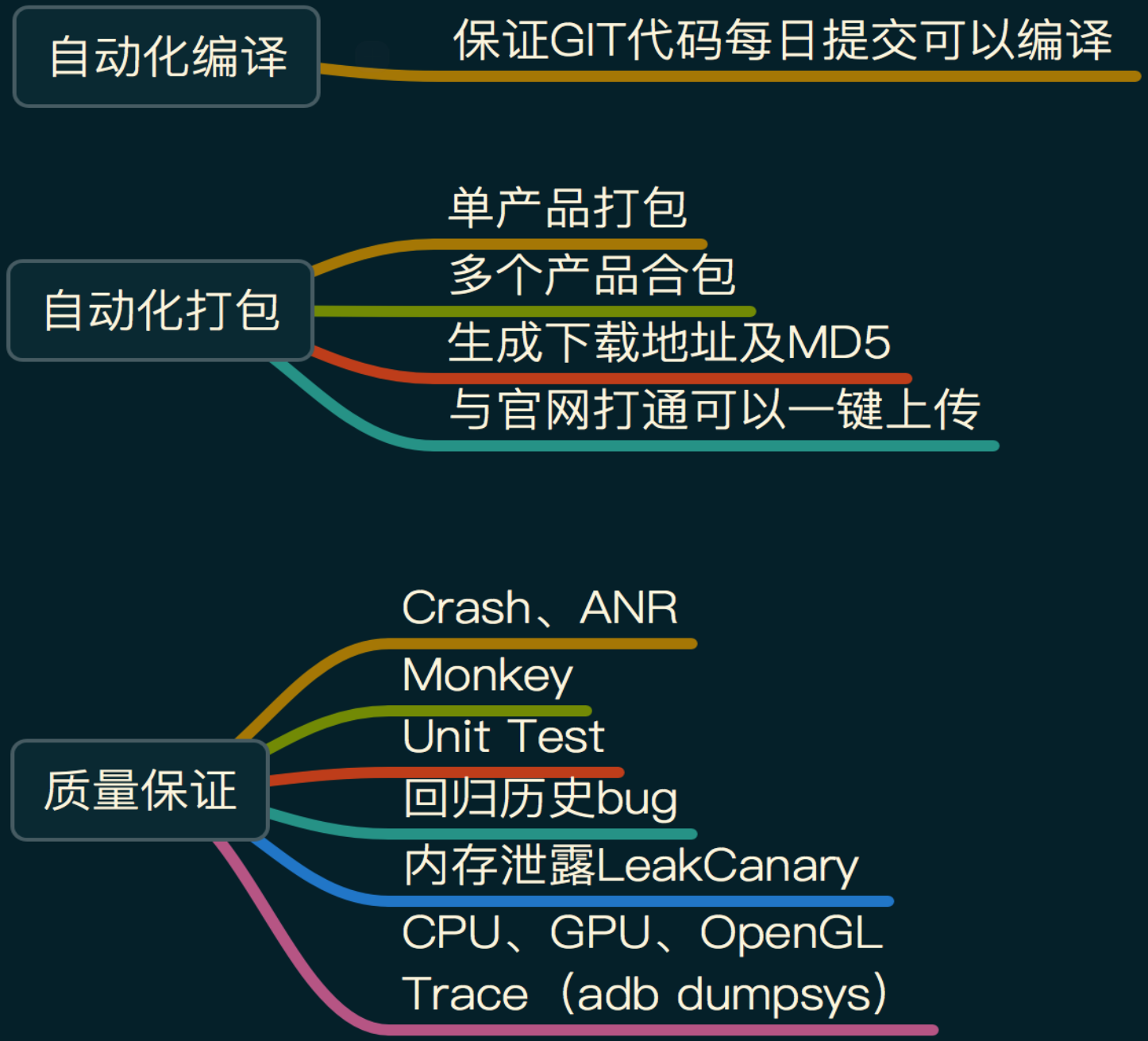
动态加速

公共库

网络模块 | 数据库模块 | 加密 | 基础功能

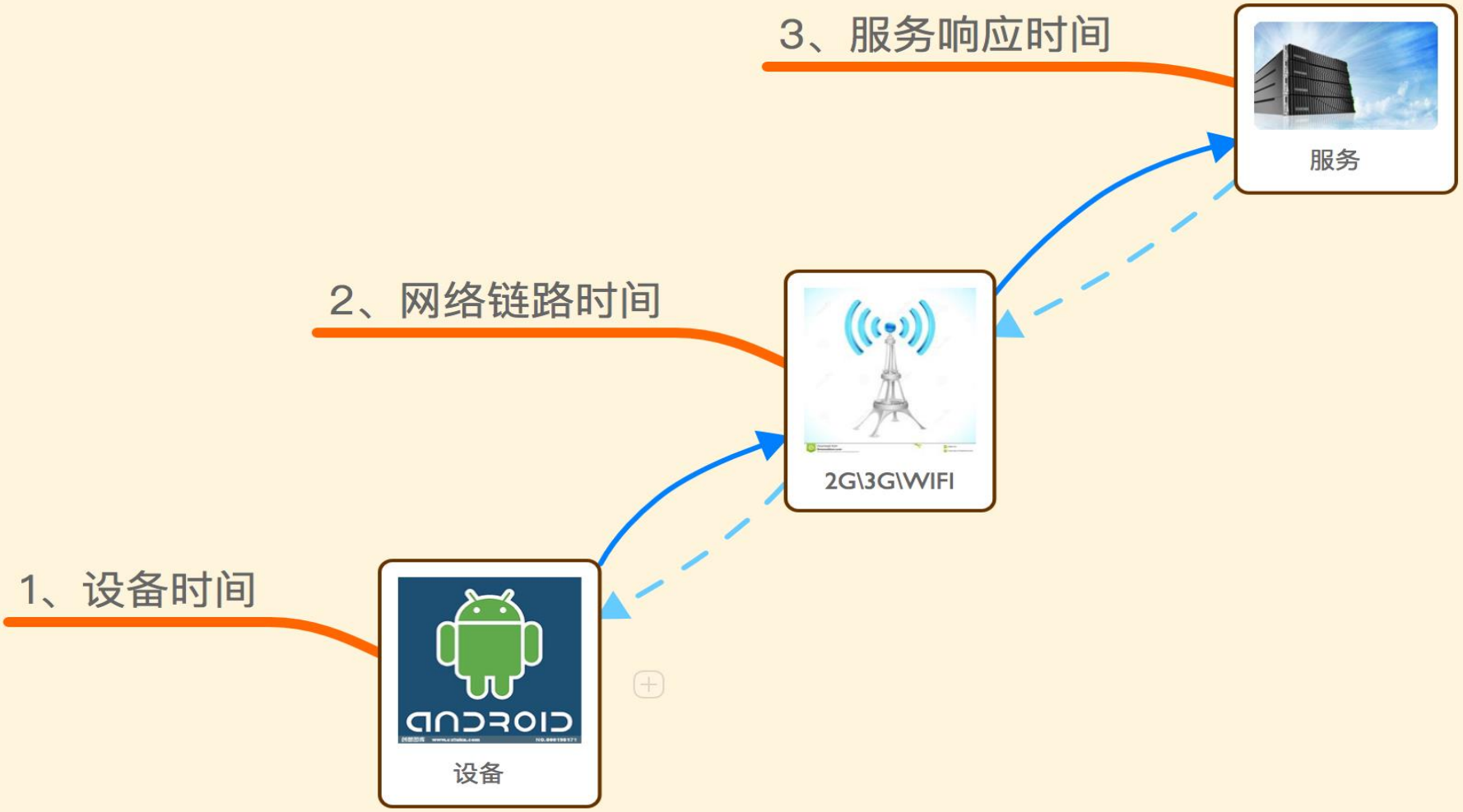

2D地图SDK
3D地图SDK
导航SDK
定位SDK
搜索SDK
more SDK ....
基础库

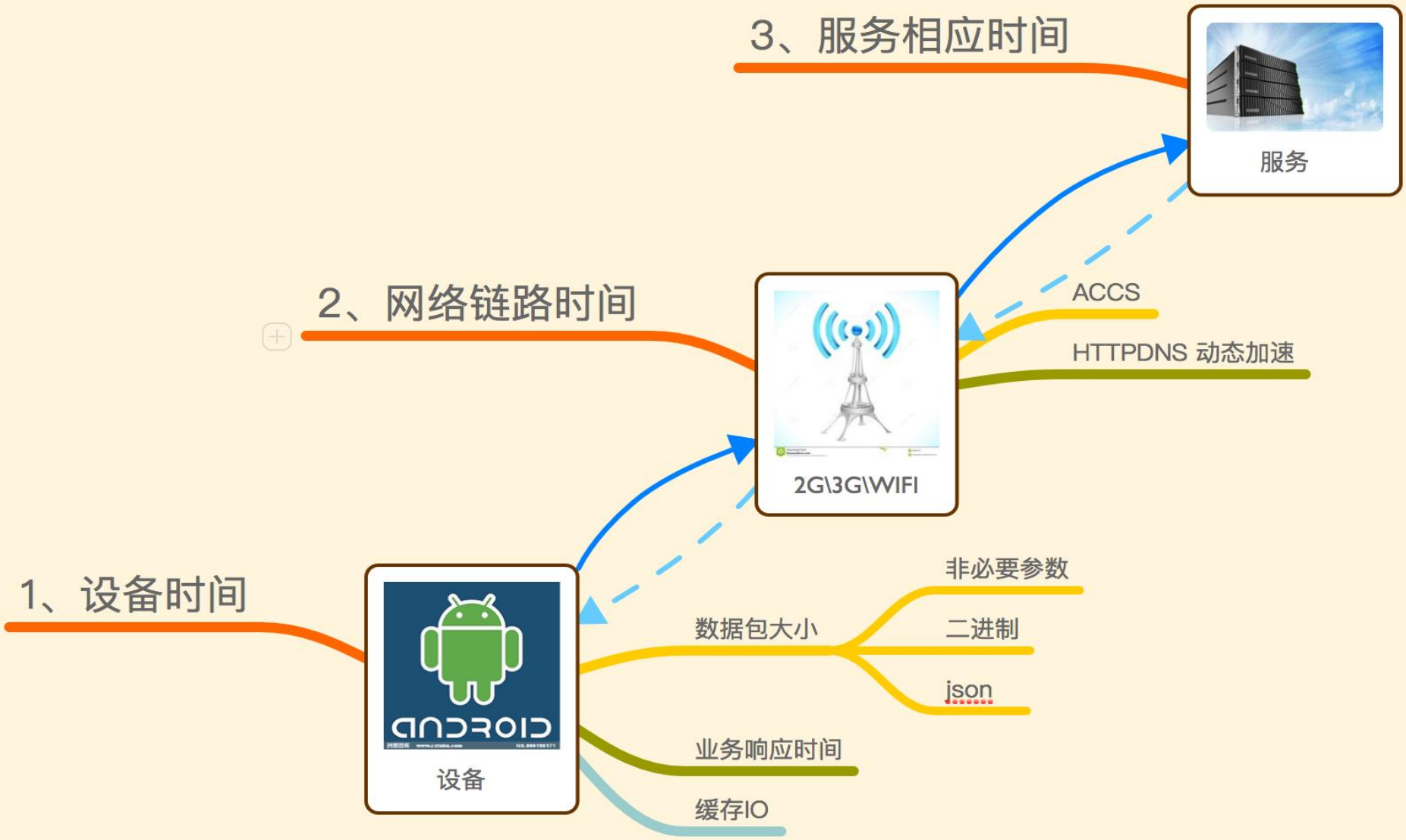

2D地图API
3D地图API
搜索API
导航API
定位API
业务
业务
业务
业务
业务
基础库


自动化编译
保证GIT代码每日提交可以编译
自动化打包
单产品打包
多个产品合包
生成下载地址及MD5
与官网打通可以一键上传
质量保证
Crash、ANR
Monkey
Unit Test
回归历史bug
内存泄露LeakCanary
CPU、GPU、OpenGL
Trace（adb dumpsys）

怎么做能更快？

- ✓ 网络请求
- ✓ 渲染速度
- ✓ 异步


3、服务响应时间
服务
2、网络链路时间
2G\3G\WIFI
1、设备时间
ANDROID
设备


3、服务相应时间
服务
2、网络链路时间
ACCS
HTTPDNS 动态加速
2G\3G\WIFI
1、设备时间
非必要参数
数据包大小
二进制
json
业务响应时间
缓存IO
设备


数据
二进制
加密
RSA加密KEY+AES
GZIP


数据
二进制
GZIP
加密
RSA加密KEY+AES

安全

- ✓ 编码安全
- ✓ 数据传输安全 二进制+AES+RSA
- ✓ https

手机预装厂商最低标准

~~DES~~

~~AES 128 || ECB模式~~

二进制+AES 256 CBC模式+RSA

解压缩漏洞：目录遍历风险

```
...
ZipEntry zipEntry = nul
while ((zipEntry = zipI
    String entryName =
    if (entryName.conta
        throw new Excep
    }
    ...
}
...
```

```
...
if (isZipFileValid(filePath)) {
    throw new Exception("unsecurity zipfile!");
} else {
    //进行解压操作
}
...

public static boolean isZipFileValid(String filePath) {
    boolean isValid = true;
    try {
        ZipFile zipFile = new ZipFile(filePath);
        Enumeration entrys = zipFile.entries();
        ZipEntry zipEntry;
        while (entrys.hasMoreElements()) {
            zipEntry = (ZipEntry) entrys.nextElement();
            if (zipEntry.getName().contains("../")) {
                return isValid = false;
            }
        }
        zipFile.close();
    } catch (Exception ex) {
        isValid = false;
        ex.printStackTrace();
    }
    return isValid;
}
```

✓ https

备忘录证书验证空实现，存在中间人攻击风险

```
import java.security.cert.CertificateException;

final class bw
  implements X509TrustManager
{
  public X509Certificate[] getAcceptedIssuers()
  {
    return null;
  }

  public void checkClientTrusted(X509Certificate[] paramArrayOfX509Certificate, String para
    throws CertificateException
  {
  }

  public void checkServerTrusted(X509Certificate[] paramArrayOfX509Certificate, String para
    throws CertificateException
  {
  }
}
```

✓ https

对域名进行强校验

对服务器证书域名进行强校验：

```
httpsURLConnection.setHostnameVerifier(SSLSocketFactory.STRICT_HOSTNAME_VERIFIER);
```

真正实现HostnameVerifier的verify()方法：

```
/**
 * https域名验证
 */
private HostnameVerifier hostnameVerifier = new HostnameVerifier() {
    @Override
    public boolean verify(String hostname, SSLSession session) {
        HostnameVerifier hv = HttpsURLConnection
                .getDefaultHostnameVerifier();
        return (hv.verify("*.amap.com", session));
    }
};
```

✓ https

WebView HTTPS安全

handler.proceed()

handler.cancel()

```
@Override
public void onReceivedSslError(WebView view, final SslErrorHandler handler, SslError error) {
    final AlertDialog.Builder builder = new AlertDialog.Builder(MyWebviewActivity.this);
    Log.d("error toString()", error.toString());
    Log.d("error getPrimaryError()", "" + error.getPrimaryError());
    SslCertificate sslCertificate = error.getCertificate();
    Log.d("sslCertificate", sslCertificate.toString());

    switch (error.getPrimaryError()) {
        case SslError.SSL_DATE_INVALID:
            Log.d("test", SslError.SSL_DATE_INVALID + " ssl date invalid");
            break;
        case SslError.SSL_IDMISMATCH:
            Log.d("test", SslError.SSL_IDMISMATCH + " hostname dismatch");
            break;
        case SslError.SSL_EXPIRED:
            Log.d("test", SslError.SSL_EXPIRED + " cert has expired");
            break;
        case SslError.SSL_UNTRUSTED:
            Log.d("test", SslError.SSL_UNTRUSTED + "cert is not trusted");
            break;
        case SslError.SSL_INVALID:
            Log.d("test", SslError.SSL_INVALID + "cert is invalid");
            break;
        case SslError.SSL_NOTYETVALID:
            Log.d("test", SslError.SSL_NOTYETVALID + "cert is not yet valid");
            break;
        }

    builder.setTitle("SSL证书错误");
    builder.setMessage("SSL错误码: " + error.getPrimaryError());
    builder.setPositiveButton("继续", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            handler.proceed();
        }
    });
    builder.setNegativeButton("取消", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
             handler.cancel();
        }
    });
    final AlertDialog dialog = builder.create();
    dialog.show();
}
```


Thanks