饿了么

中生代技术 FRESHMAN TECHNOLOGY | IT大咖说 不止于技术

饿了么Android端APM演进

耿万鹏@饿了么移动

2017.3.11

拉扎斯网络科技（上海）有限公司
Rajax Network & Tehnology Co

目录

1 背景

2 挑战

3 原理

4 展示

5 展望

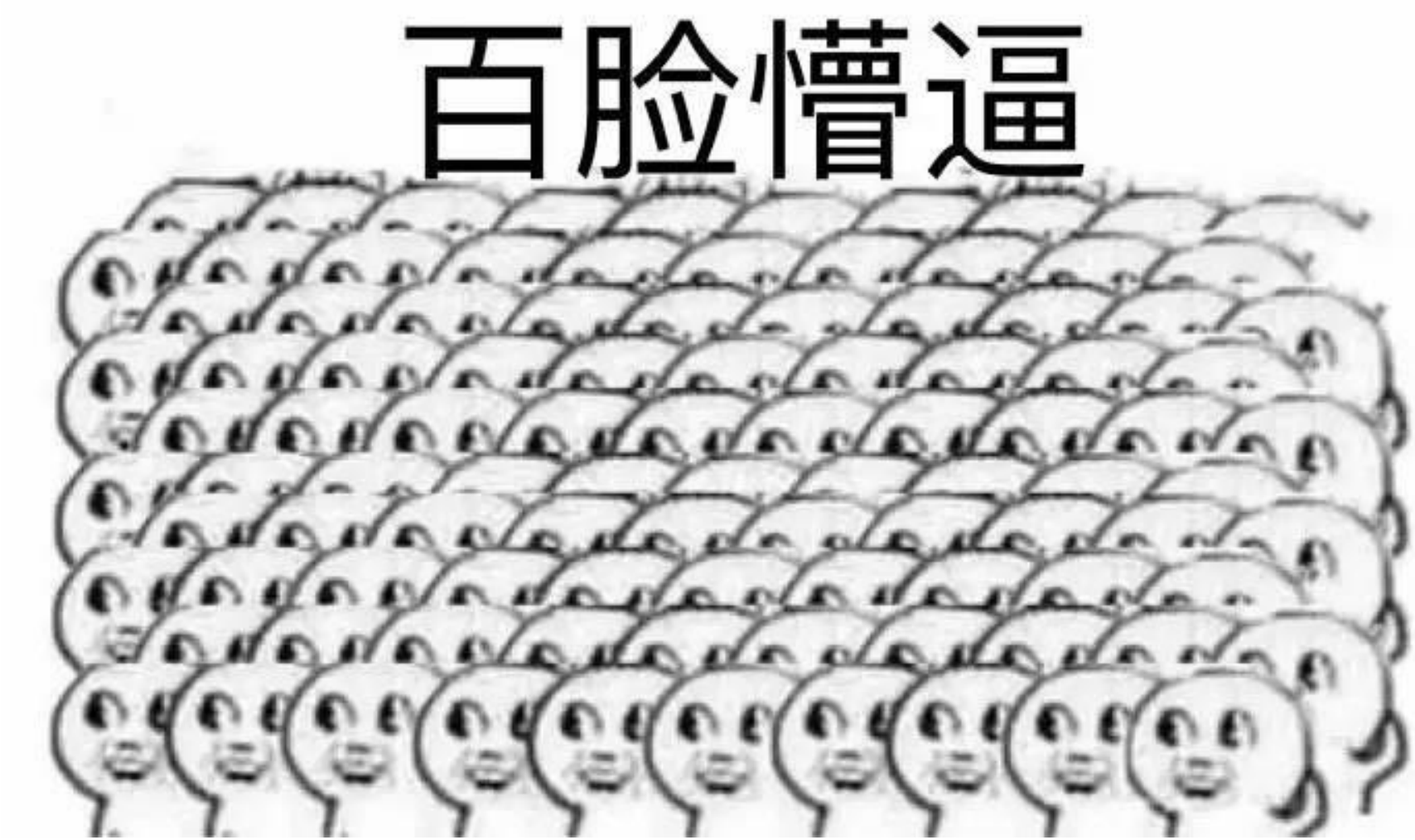
背景

背景

日订单过900w
移动端的业务量
占比高达98%

背景

出现问题难以定位



背景

提高应用性能，发现性能瓶颈

挑战

挑战

移动端:

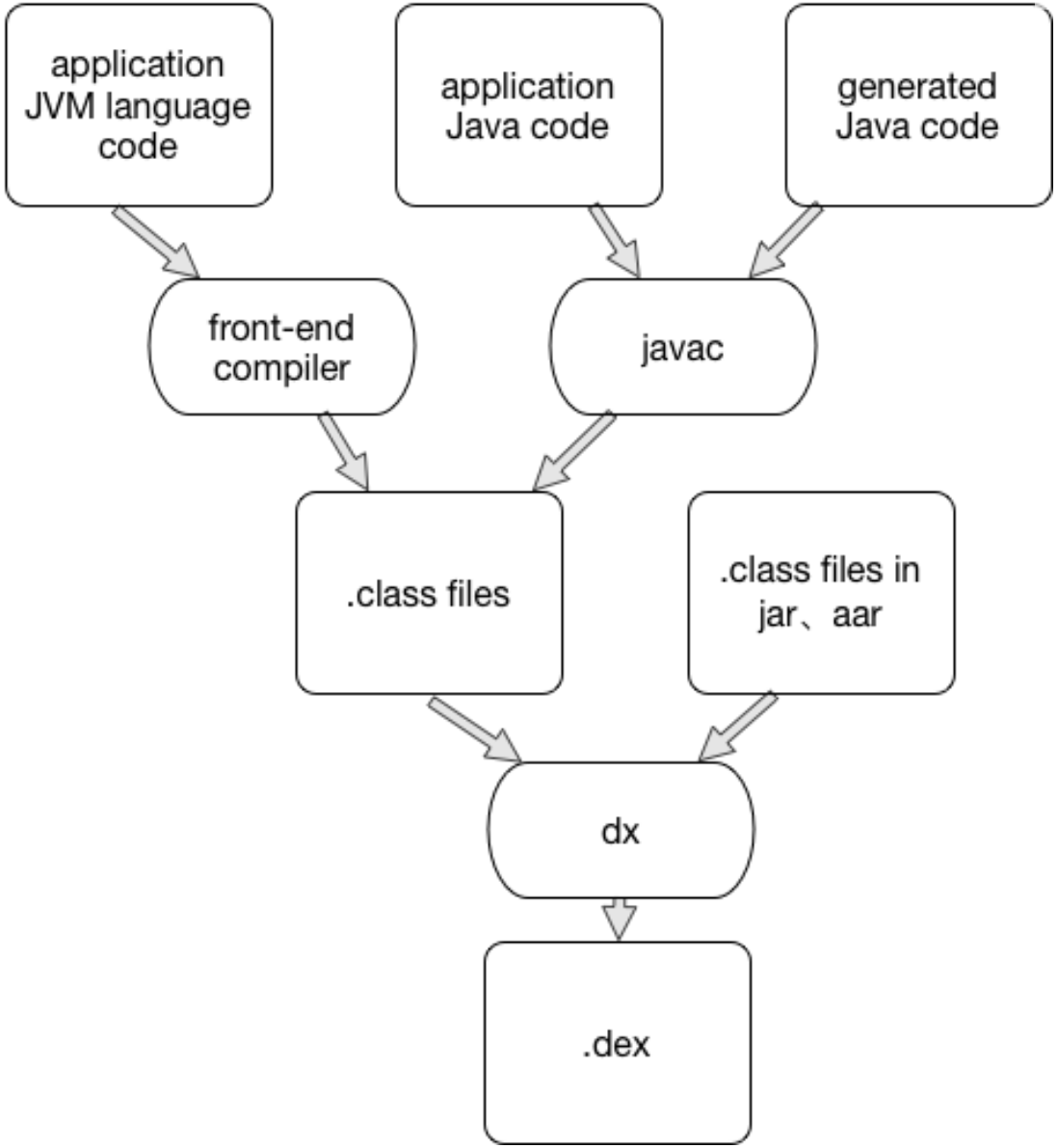
无侵入
低影响
高兼容

原理

3

实现原理

打包流程


application JVM language code
application Java code
generated Java code
front-end compiler
javac
.class files
.class files in jar、aar
dx
.dex

实现原理

JVM Bytecode

```
ClassFile {
    u4              magic;
    u2              minor_version;
    u2              major_version;
    u2              constant_pool_count;
    cp_info         constant_pool[constant_pool_count-1];
    u2              access_flags;
    u2              this_class;
    u2              super_class;
    u2              interfaces_count;
    u2              interfaces[interfaces_count];
    u2              fields_count;
    field_info      fields[fields_count];
    u2              methods_count;
    method_info     methods[methods_count];
    u2              attributes_count;
    attribute_info attributes[attributes_count];
}
```

```
Code_attribute {
    u2 attribute_name_index;
    u4 attribute_length;
    u2 max_stack;
    u2 max_locals;
    u4 code_length;
    u1 code[code_length];
    u2 exception_table_length;
    {   u2 start_pc;
        u2 end_pc;
        u2 handler_pc;
        u2 catch_type;
    } exception_table[exception_table_length];
    u2 attributes_count;
    attribute_info attributes[attributes_count];
}
```

实现原理

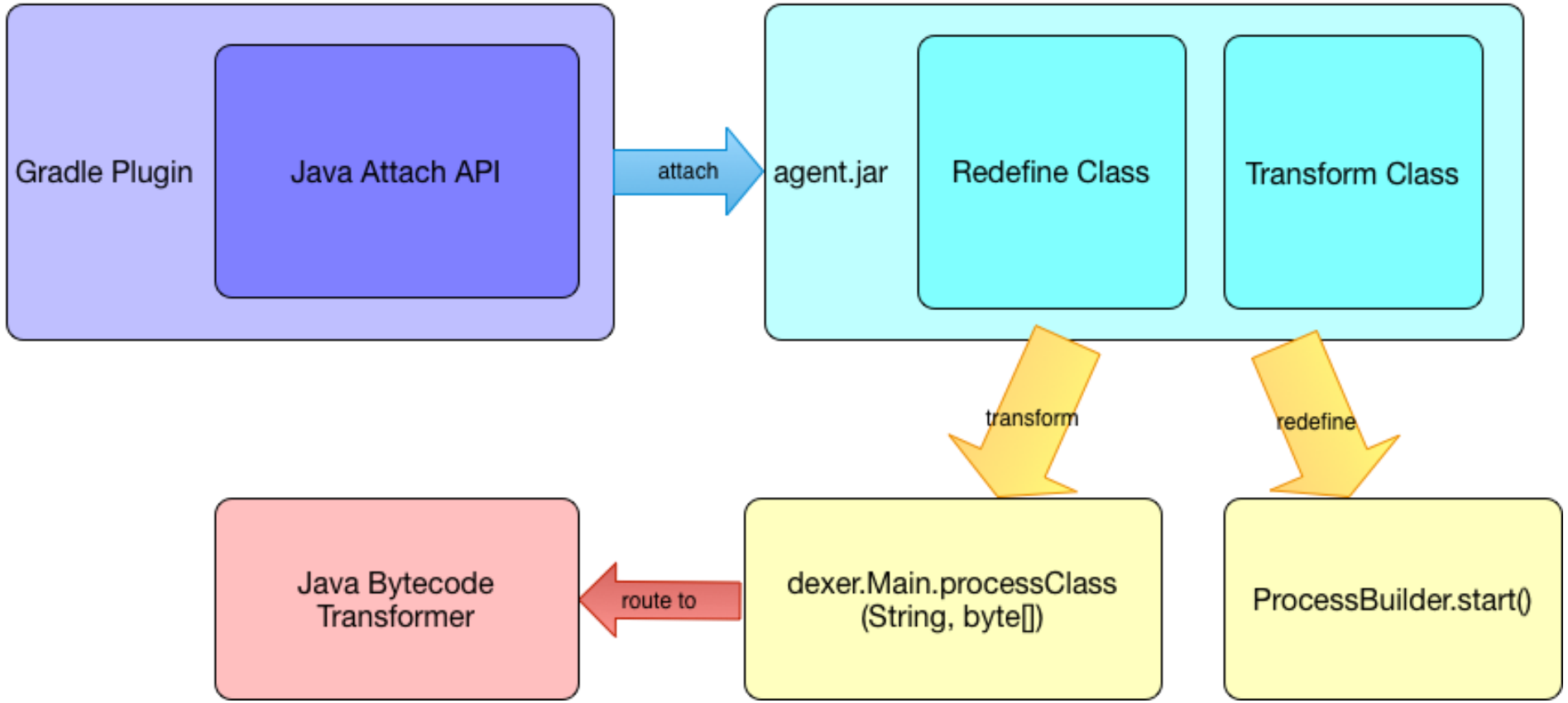
JVM Bytecode

```java
/**
 *
 * Created by gengwanpeng on 2017/2/7.
 */
public class Main {
    public static void main(String[] args) {
        int sum = new Main().add(1, 2);
        System.out.println(sum);
    }

    public int add(int a, int b) {
        return a + b;
    }
}
```

```
Compiled from "Main.java"
public class me.ele.test.Main {
  public me.ele.test.Main();
    Code:
       0: aload_0
       1: invokespecial #1                  // Method java/lang/Object."<init>":()V
       4: return

  public static void main(java.lang.String[]);
    Code:
       0: new           #2                  // class me/ele/test/Main
       3: dup
       4: invokespecial #3                  // Method "<init>":()V
       7: iconst_1
       8: iconst_2
       9: invokevirtual #4                  // Method add:(II)I
      12: istore_1
      13: getstatic     #5                  // Field java/lang/System.out:Ljava/io/PrintStream;
      16: iload_1
      17: invokevirtual #6                  // Method java/io/PrintStream.println:(I)V
      20: return

  public int add(int, int);
    Code:
       0: iload_1
       1: iload_2
       2: iadd
       3: ireturn
}
```

实现原理

传统做法


Gradle Plugin
Java Attach API
attach
agent.jar
Redefine Class
Transform Class
transform
redefine
Java Bytecode Transformer
route to
dexer.Main.processClass (String, byte[])
ProcessBuilder.start()

实现原理

Transform API

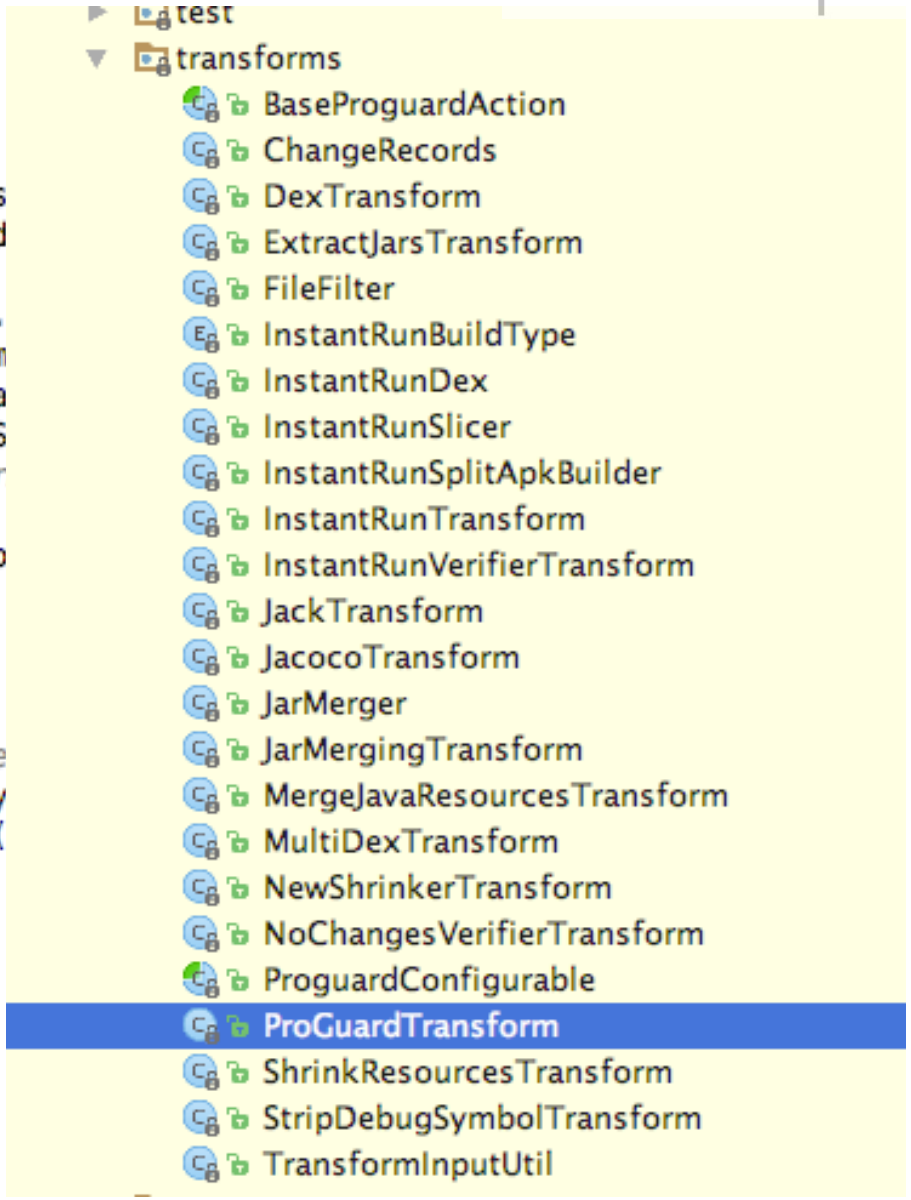
对比Attach API方案

Android 插件原生支持，可增量编译

无须处理混淆

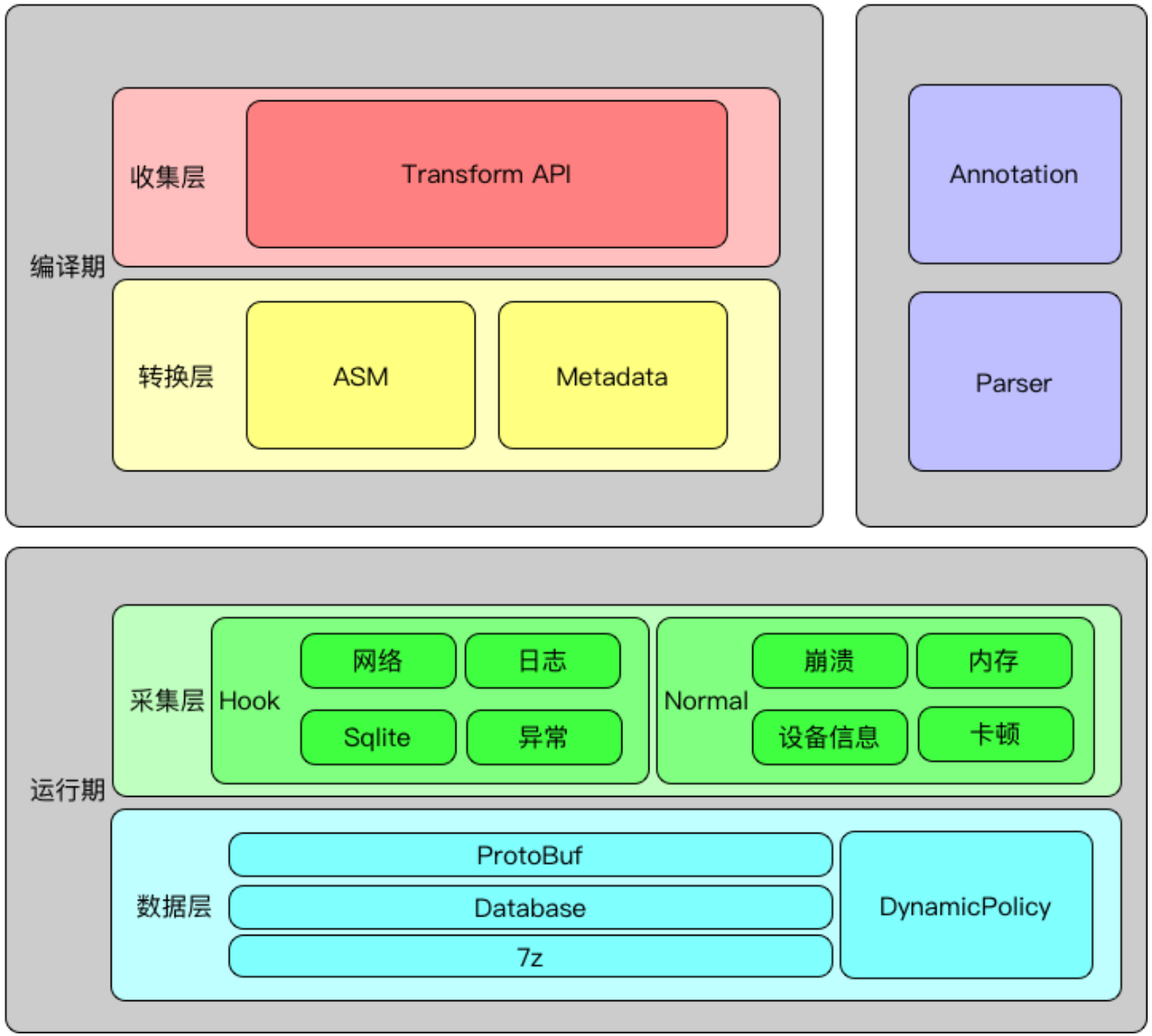
兼容 InstantRun MultiDex

```
// ----- External Transforms -----
// apply all the external transforms.
List<Transform> customTransforms = extens
List<List<Object>> customTransformsDepend

for (int i = 0, count = customTransforms.
    Transform transform = customTransform
    AndroidTask<TransformTask> task = tra
    .addTaskDependencies(tasks, variantS
    // task could be null if the transfor
    if (task != null) {
        List<Object> deps = customTransfo
        if (!deps.isEmpty()) {
            task.dependsOn(tasks, deps);
        }

        // if the task is a no-op then we
        if (transform.getScopes().isEmpty
            variantScope.getAssembleTask(
        }

    }
}

// ----- Minify next -----
```



实现原理

SDK整体架构



展示

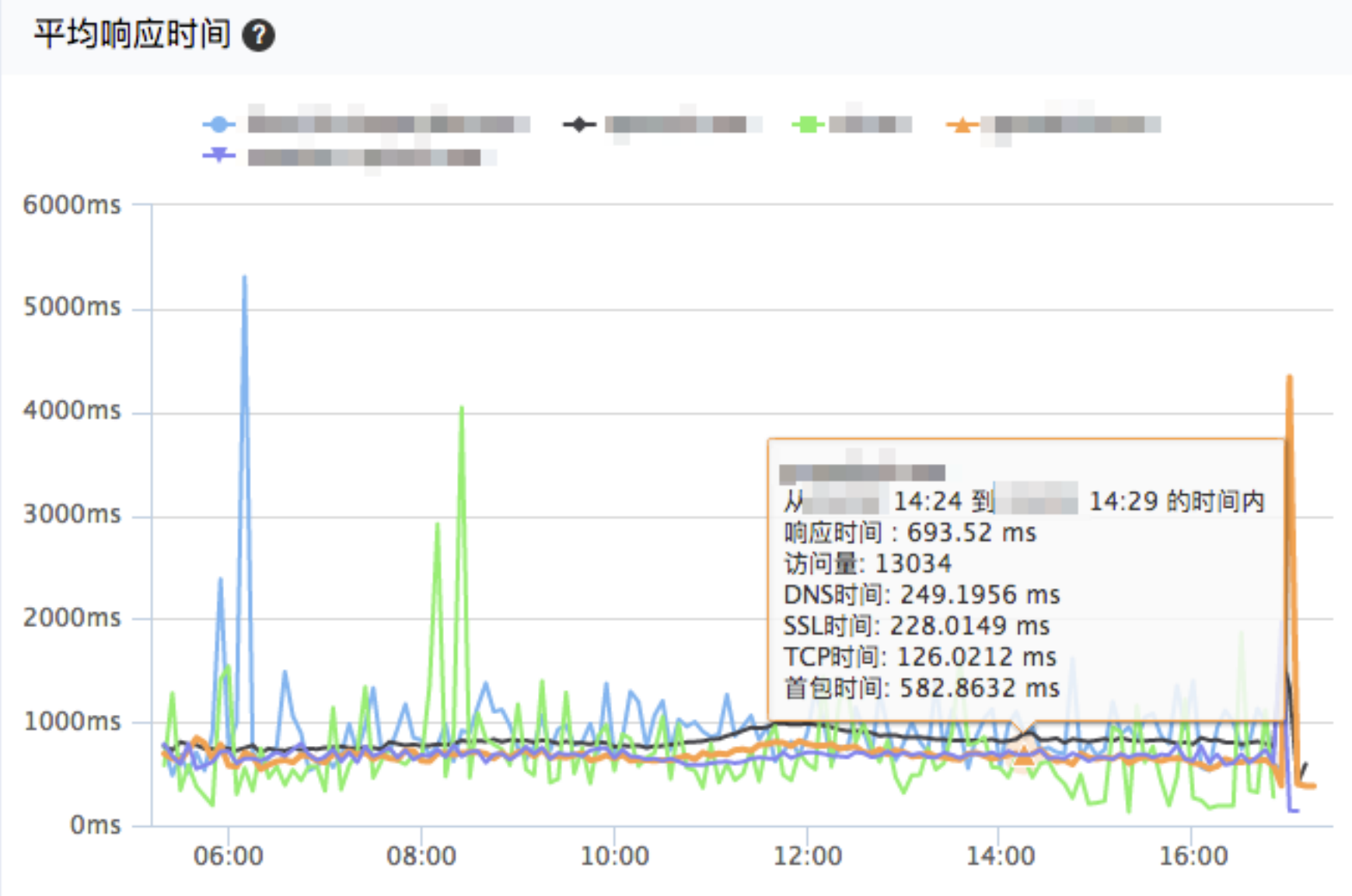
4.1

展示

响应时间	吞吐率	平均数据传输量

[illegible]	698.71ms
[illegible]	281.45ms
[illegible]	582.48ms
[illegible]	223.86ms
[illegible]	624.80ms
[illegible]	516.42ms
[illegible]	396.55ms
[illegible]	355.56ms
[illegible]	2.95sec
[illegible]	358.62ms
[illegible]	647.50ms
[illegible]	580.25ms
[illegible]	530.88ms
[illegible]	492.44ms
[illegible]	420.43ms
[illegible]	404.32ms

展示


平均响应时间
6000ms
5000ms
4000ms
3000ms
2000ms
1000ms
0ms
06:00
08:00
10:00
12:00
14:00
16:00
从 14:24 到 14:29 的时间内
响应时间 : 693.52 ms
访问量: 13034
DNS时间: 249.1956 ms
SSL时间: 228.0149 ms
TCP时间: 126.0212 ms
首包时间: 582.8632 ms

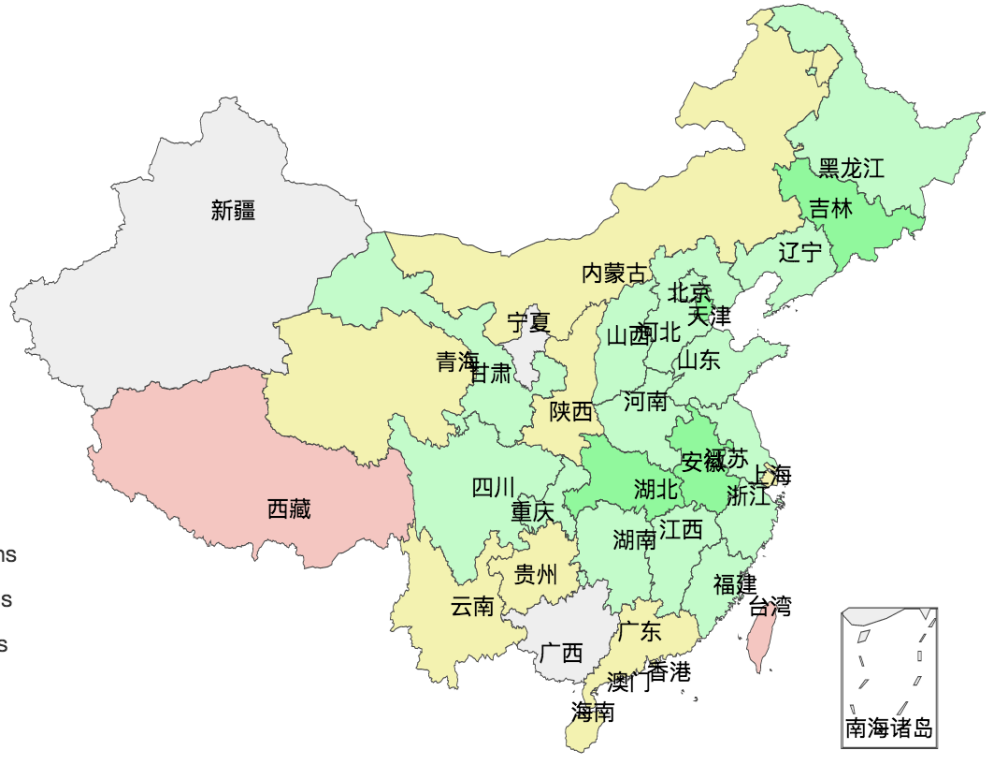
展示

所有版本 | 最近1天 | 所有运营商 | 所有网络

查询 响应时间

天津	560.19ms
内蒙古	817.88ms
青海	932.26ms
四川	739.20ms
澳门	2.16sec
台湾	1.40sec
广东	813.11ms
贵州	967.24ms
安徽	596.49ms
河南	653.38ms
山东	668.76ms
云南	853.91ms
山西	706.65ms
湖南	705.65ms
上海	925.98ms

响应时间



- > 4000ms
- 2500 ~ 4000ms
- 1500 ~ 2500ms
- 1000 ~ 1500ms
- 800 ~ 1000ms
- 600 ~ 800ms
- 400 ~ 600ms
- < 400ms

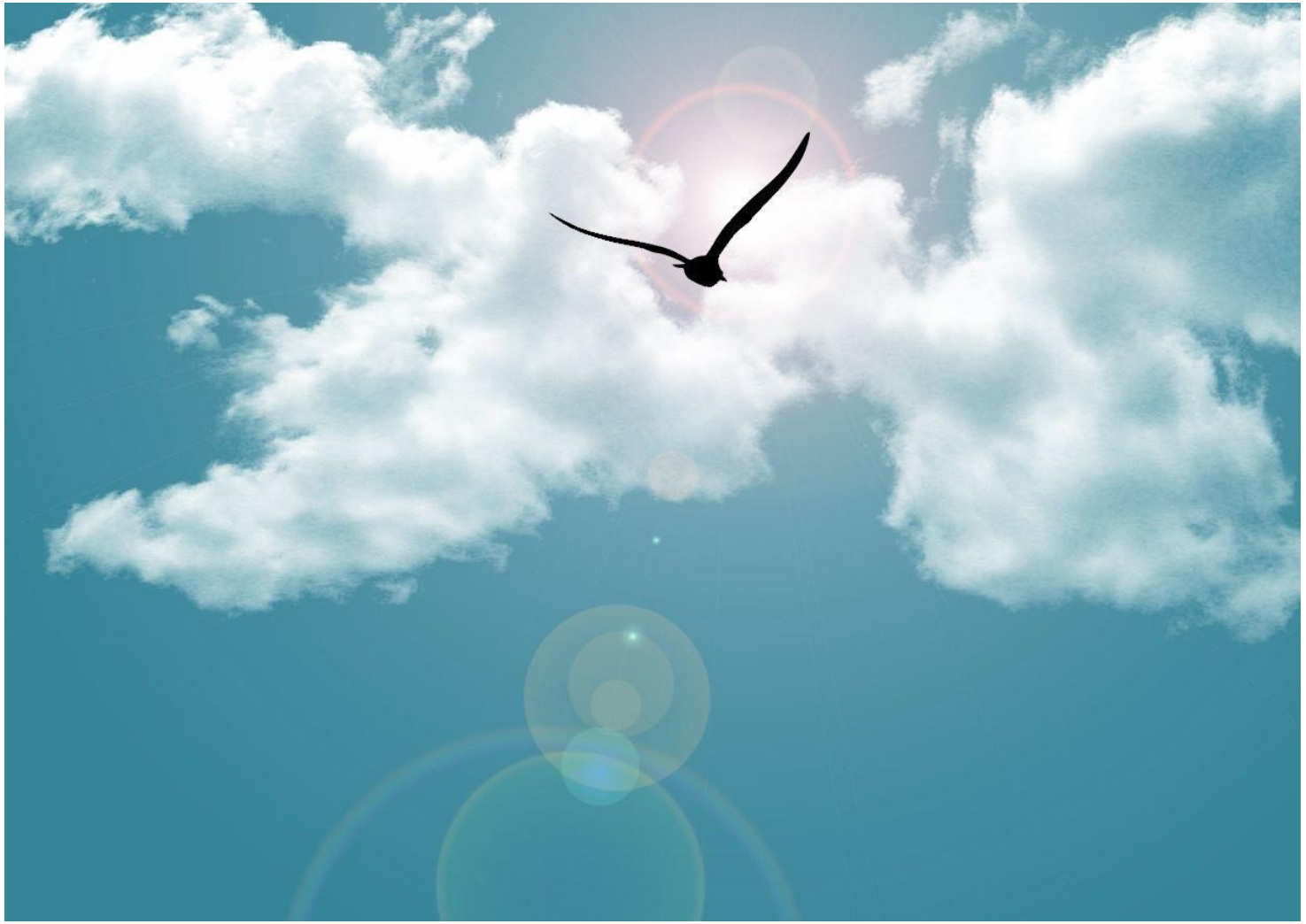
展望

展望

- 基于dex转换

定点排障

其他功能



Q&A

中生代技术
FRESHMAN TECHNOLOGY
IT大咖说
不止于技术

THANKS !