

Pure-native Architecture

赵世婚 @沪江

Geekbang>

极客邦科技

全球领先的技术人学习和交流平台

扫我，码上开启新世界



Geekbang>

InfoQ | EGO NETWORKS | StuQ

InfoQ

专注中高端技术人员
的社区媒体

EGO NETWORKS

EXTRA GEEKS' ORGANIZATION
高端技术人员
学习型社交网络

StuQ

实践驱动的IT职业
学习和服务平台



促进软件开发领域知识与创新的传播



实践第一 案例为主

时间：2015年12月18-19日 / 地点：北京·国际会议中心

欢迎您参加ArchSummit北京2015, 技术因你而不同



ArchSummit北京二维码



[北京站]

2016年04月21日-23日



关注InfoQ官方信息
及时获取QCon演讲视频信息

TABLE OF Contents

- Introduction to Pure-native
- Pure-native architecture
- The internal of Android NDK
- Pure-native Within Hujiang CCTalk

WHY

Pure-native

WHAT

Pure-native

Problem

Efficient

- Divide develop

Security

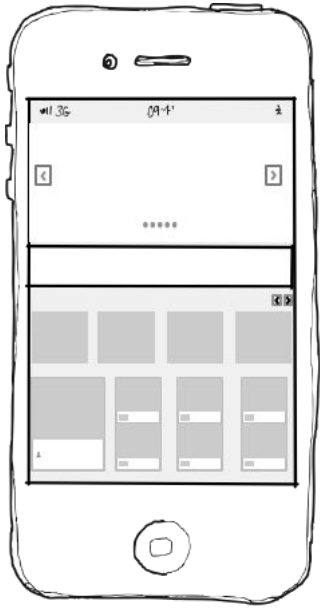
- Reflect、 Disassemble

Performance

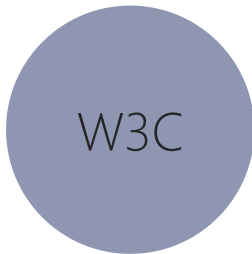
- Unzip raise to over 250%

Cost

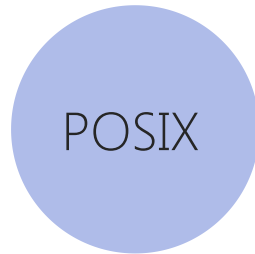
Pure-native



Hybrid App



W3C



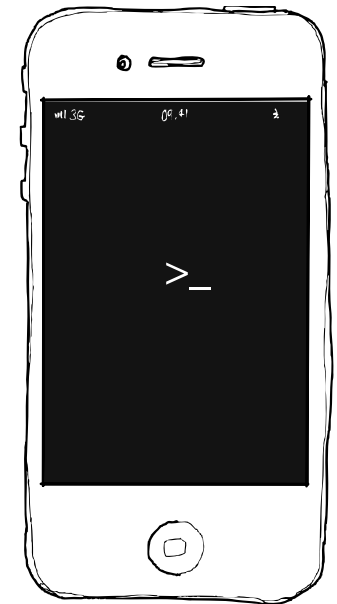
POSIX



Bionic



Darwin

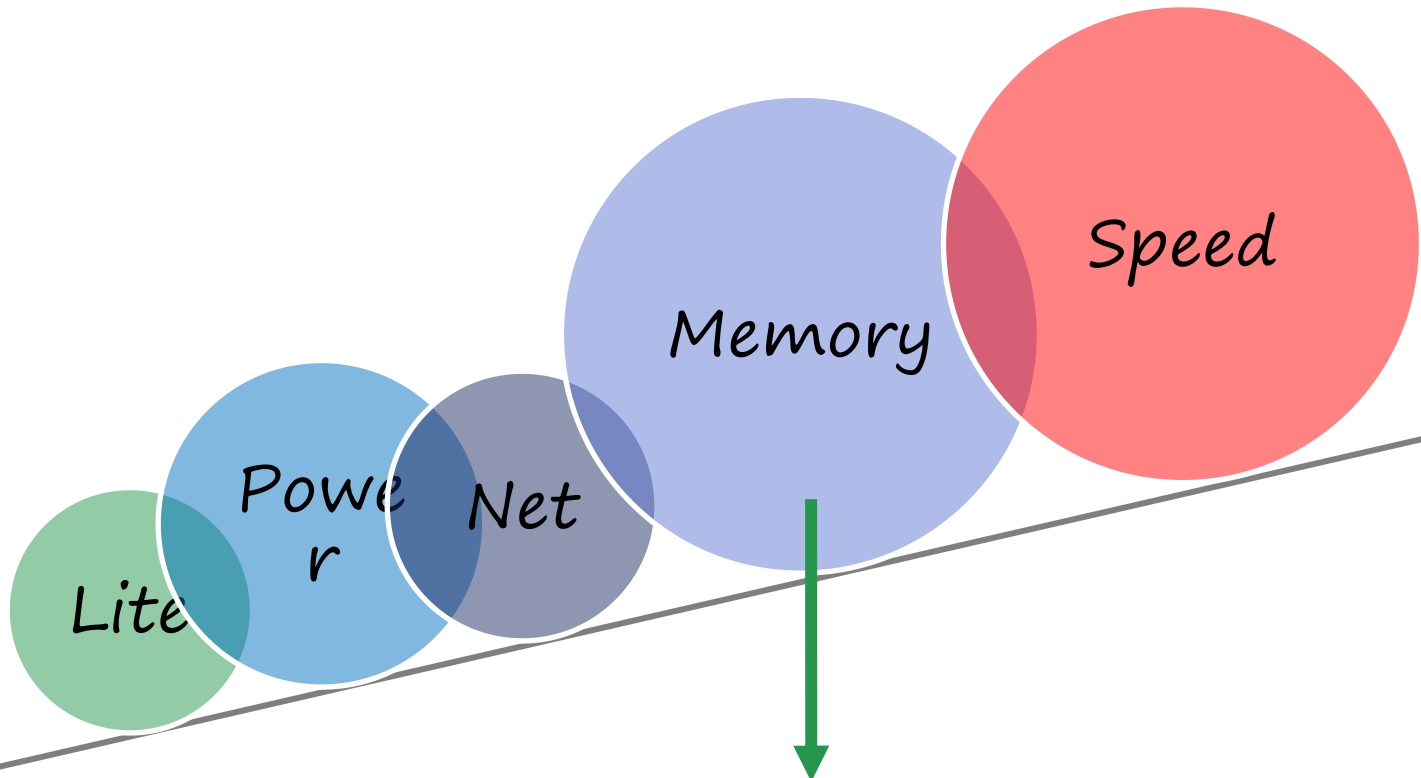


Pure-native
brought by **InfoQ**

HOW

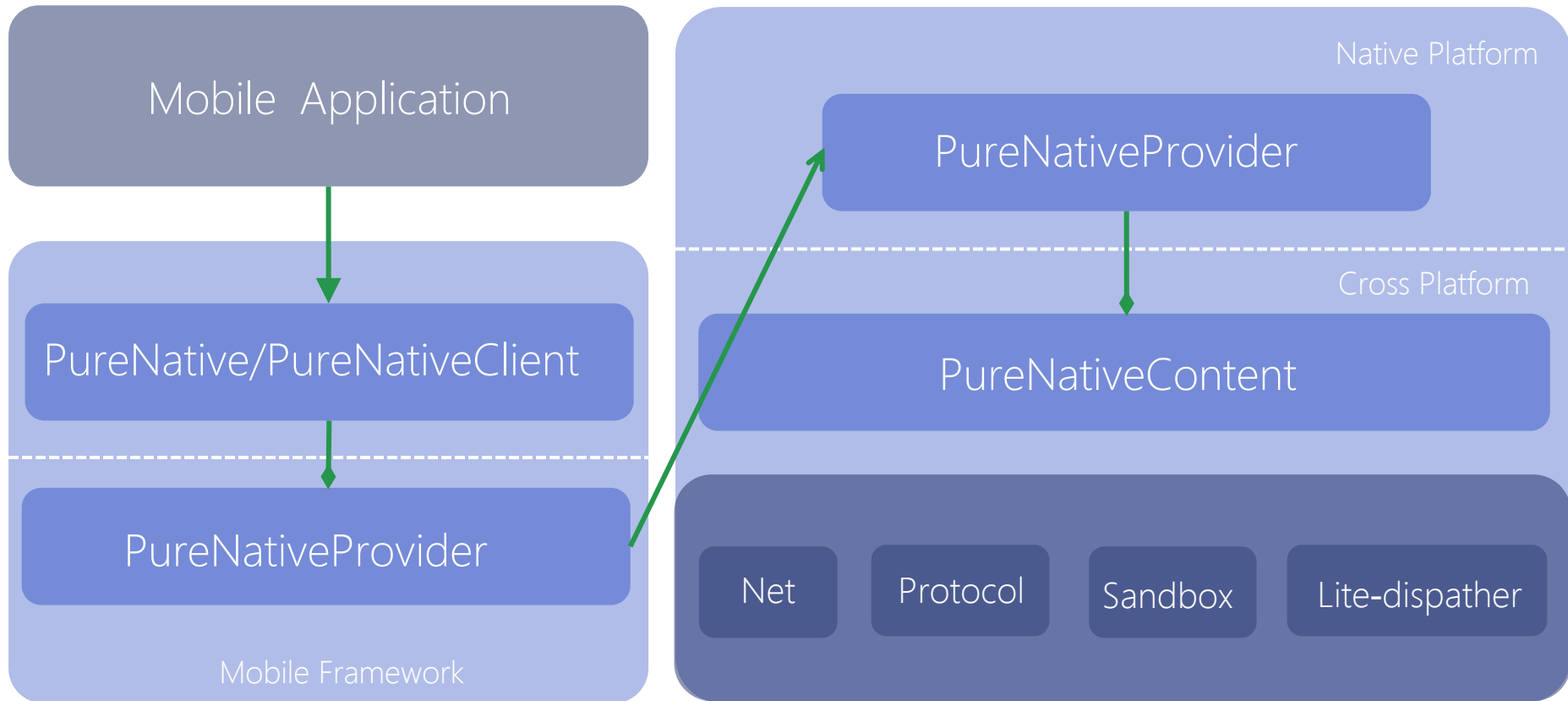
Pure-native

Mobile System Sensitivities

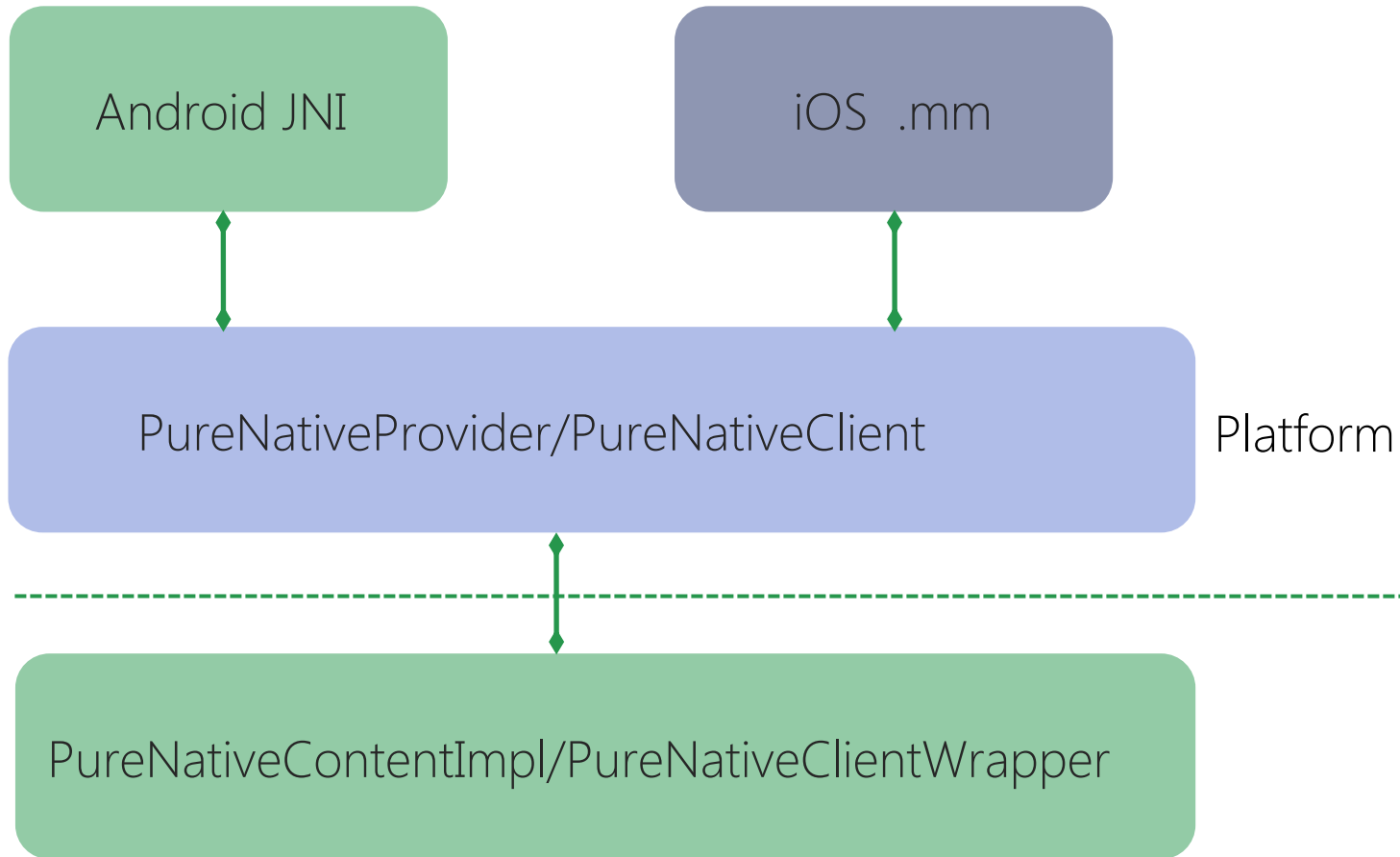


iOS8/iPhone6/646MB
Android 4.1/Mi2/96MB

Pure-native architecture

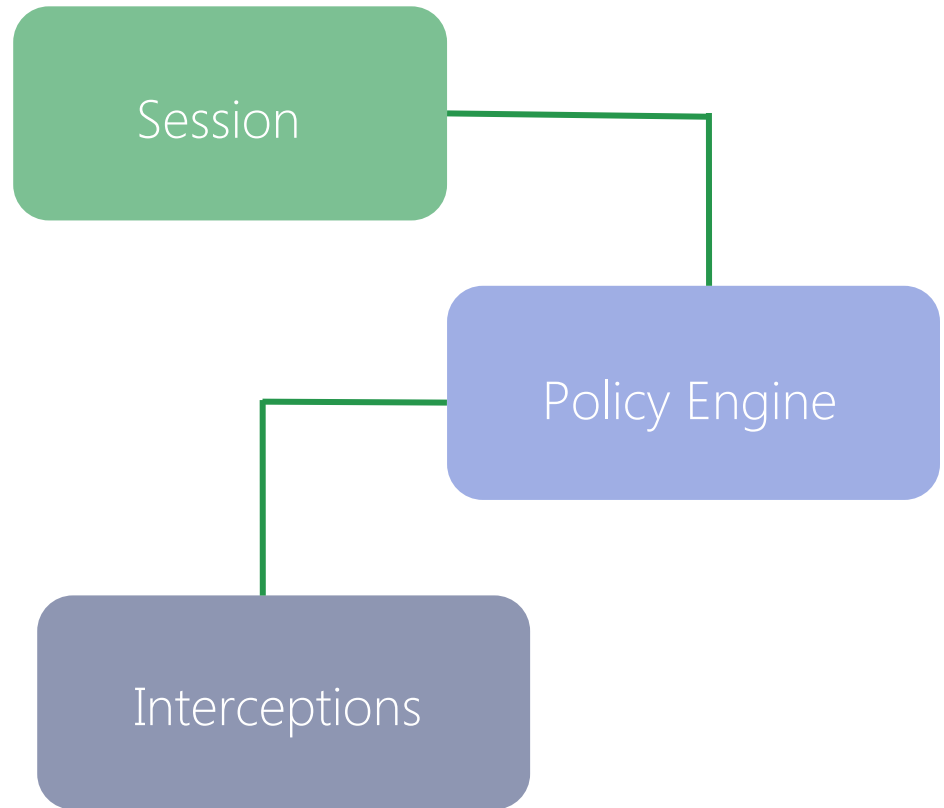


Pure-native platform



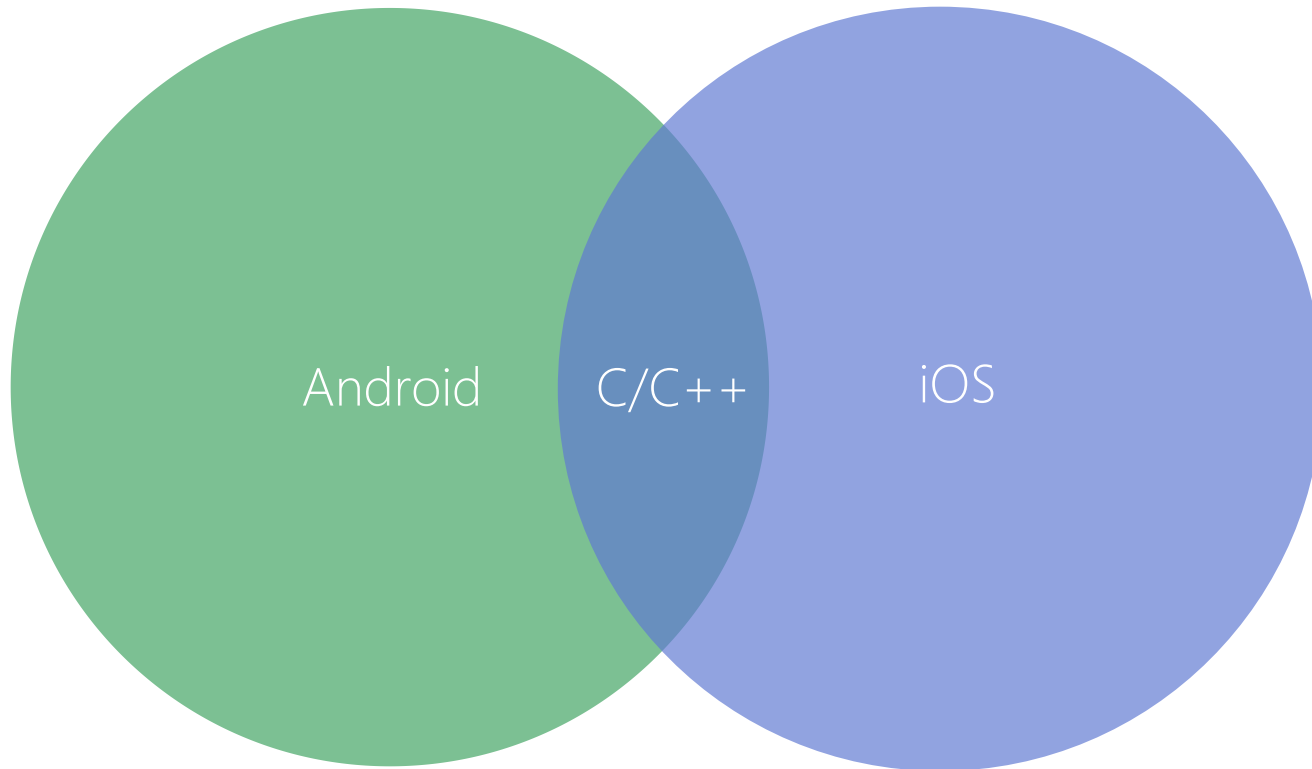
Sandbox Process

- Restrict write to disk
- Reflect Reject
- Forbid Access Sensitive info
 - Locations
 - Micro phone
 - Photo & Camera



DESIGNING CROSS- LANGUAGE INTERFACE

Cross-language



Cross-language interface design

Python & GYP Auto Generate Code

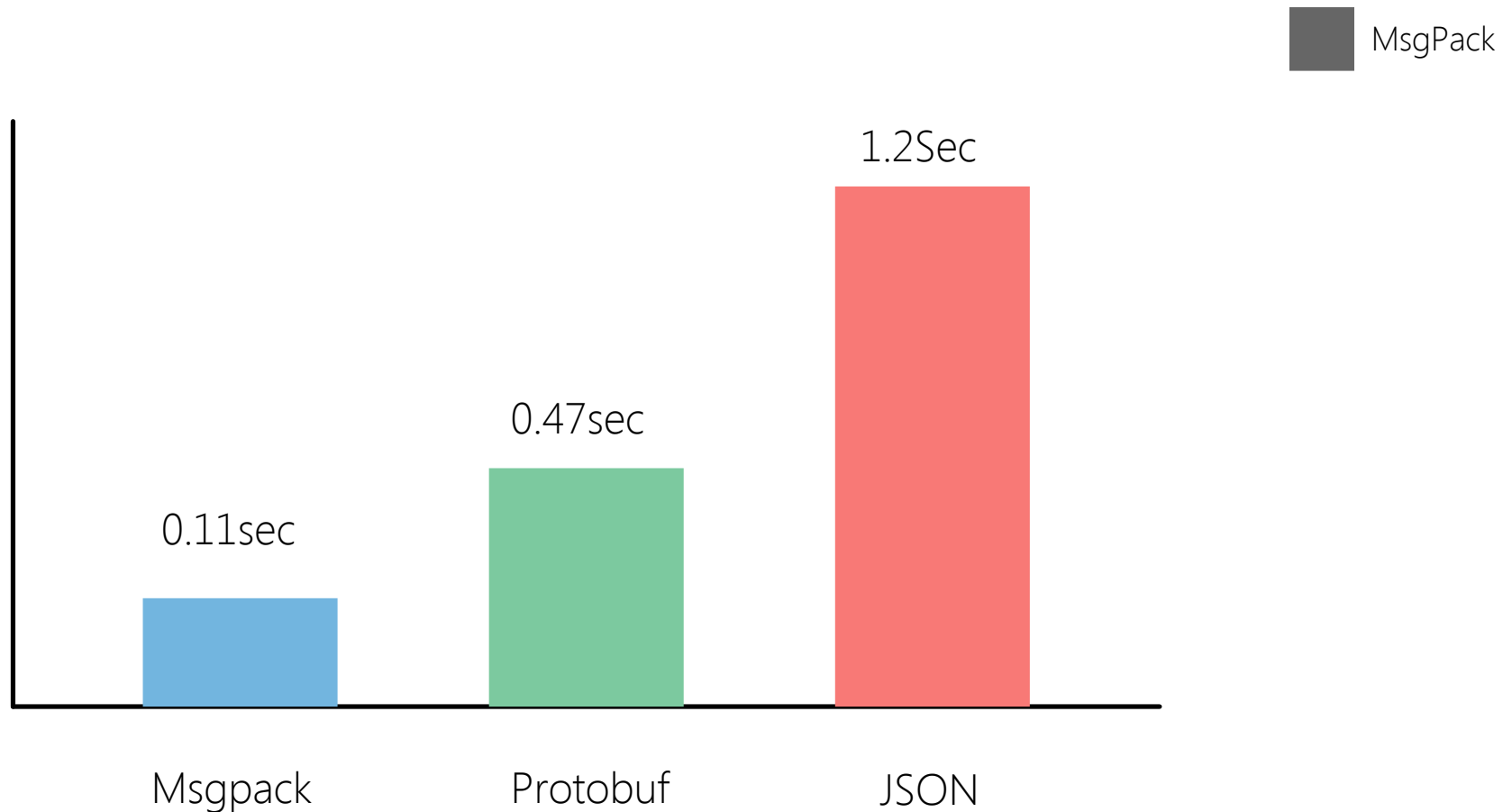
- @CalledByNative, native

Unify Language Interface

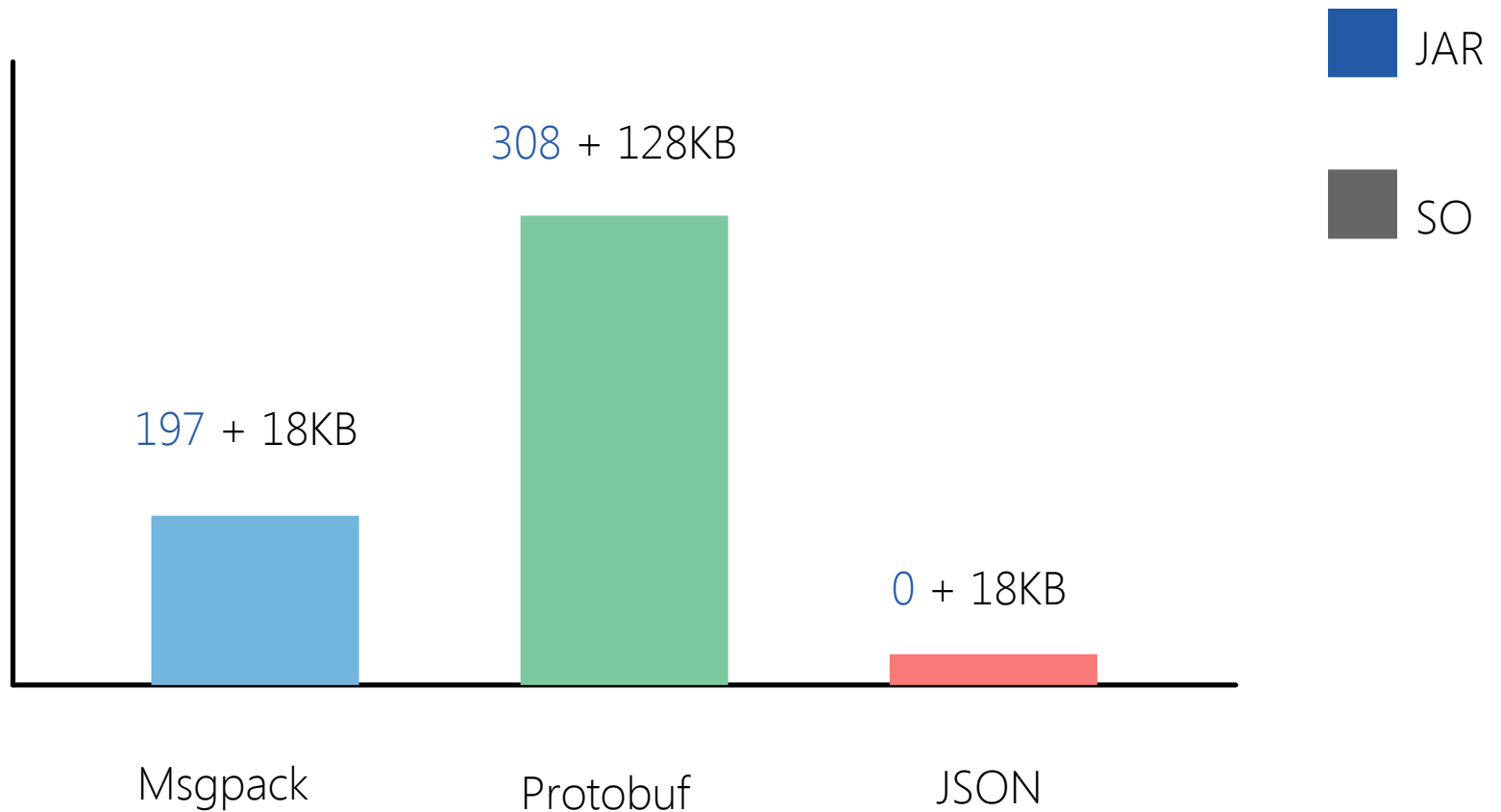
- Notify, RPC

Benchmark for speed

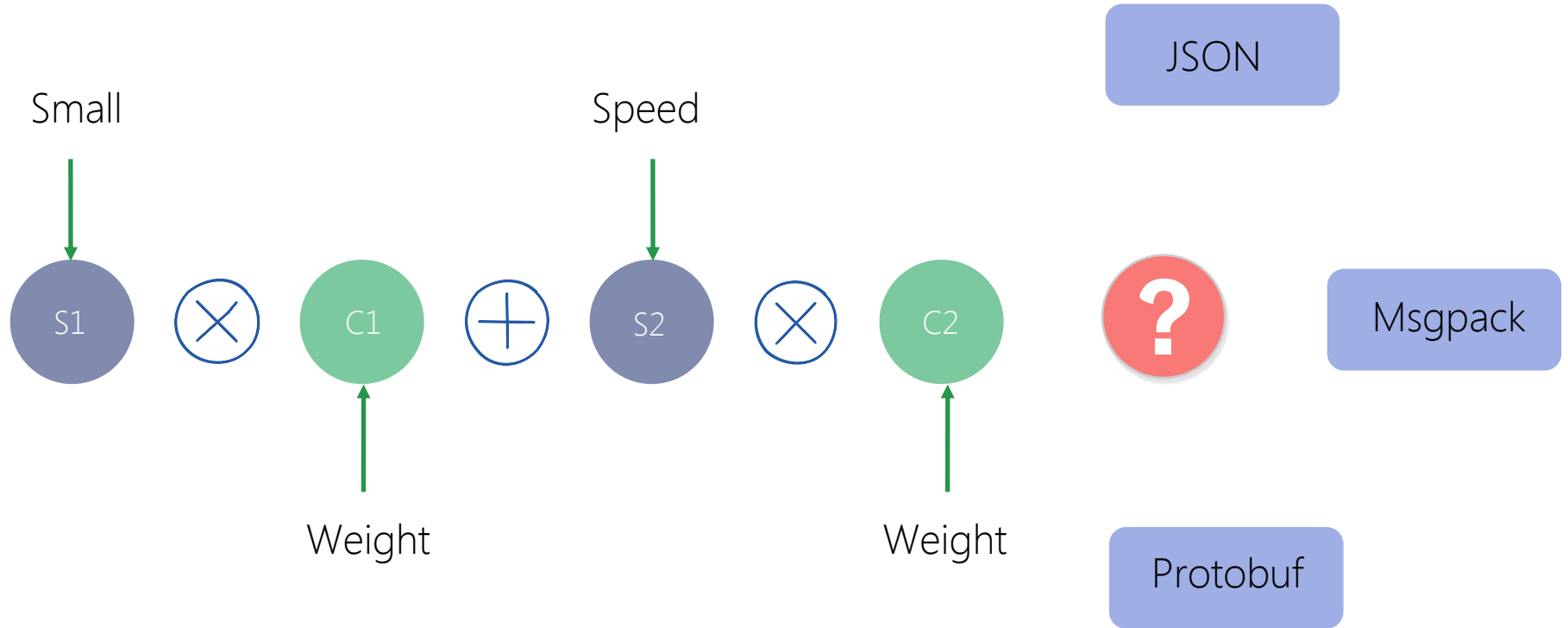
Serialization + DeSerialization



Benchmark for size

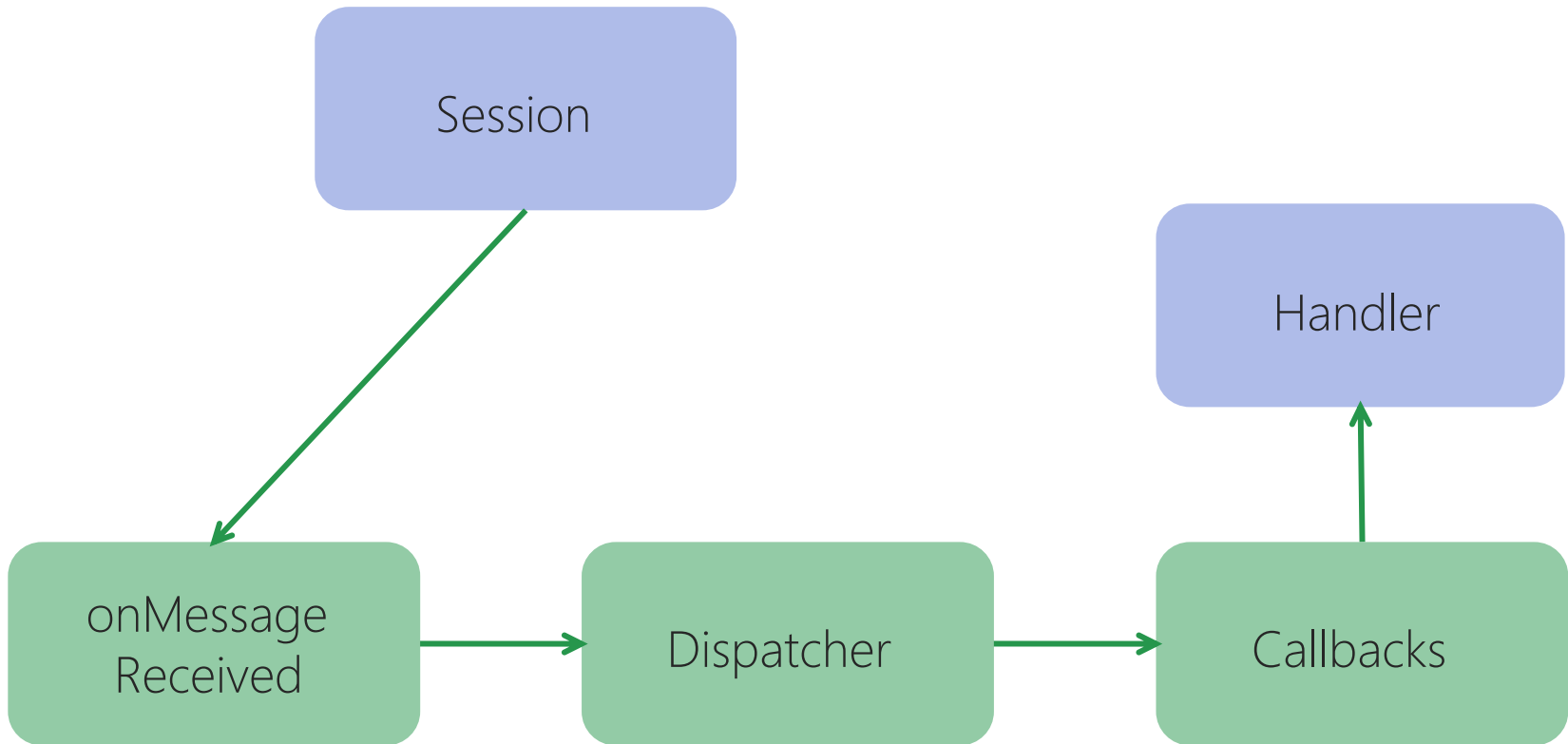


The Select of Data Exchange



Design Lite-dispatcher

Lite-dispatcher design analysis



Easy Use

Declaring messages

```
CC_MESSAGE0_0(Param0, 0)
CC_MESSAGE1_0(Param1, 1, bool)
CC_MESSAGE2_0(Param2, 2, bool, int)
CC_MESSAGE3_0(Param3, 3, bool, int, float)
CC_MESSAGE4_0(Param4, 4, bool, int, float, double)
CC_MESSAGE5_0(Param5, 5, bool, int, float, double, const char*)
```

Easy Use

Implement Listener

```
class Listener {  
public:  
    // Called when a message is received. Returns true if the message was handled.  
    virtual bool OnMessageReceived(const Message&) = 0;  
  
protected:  
    virtual ~Listener() {}  
};
```

Easy Use

Implement Handler

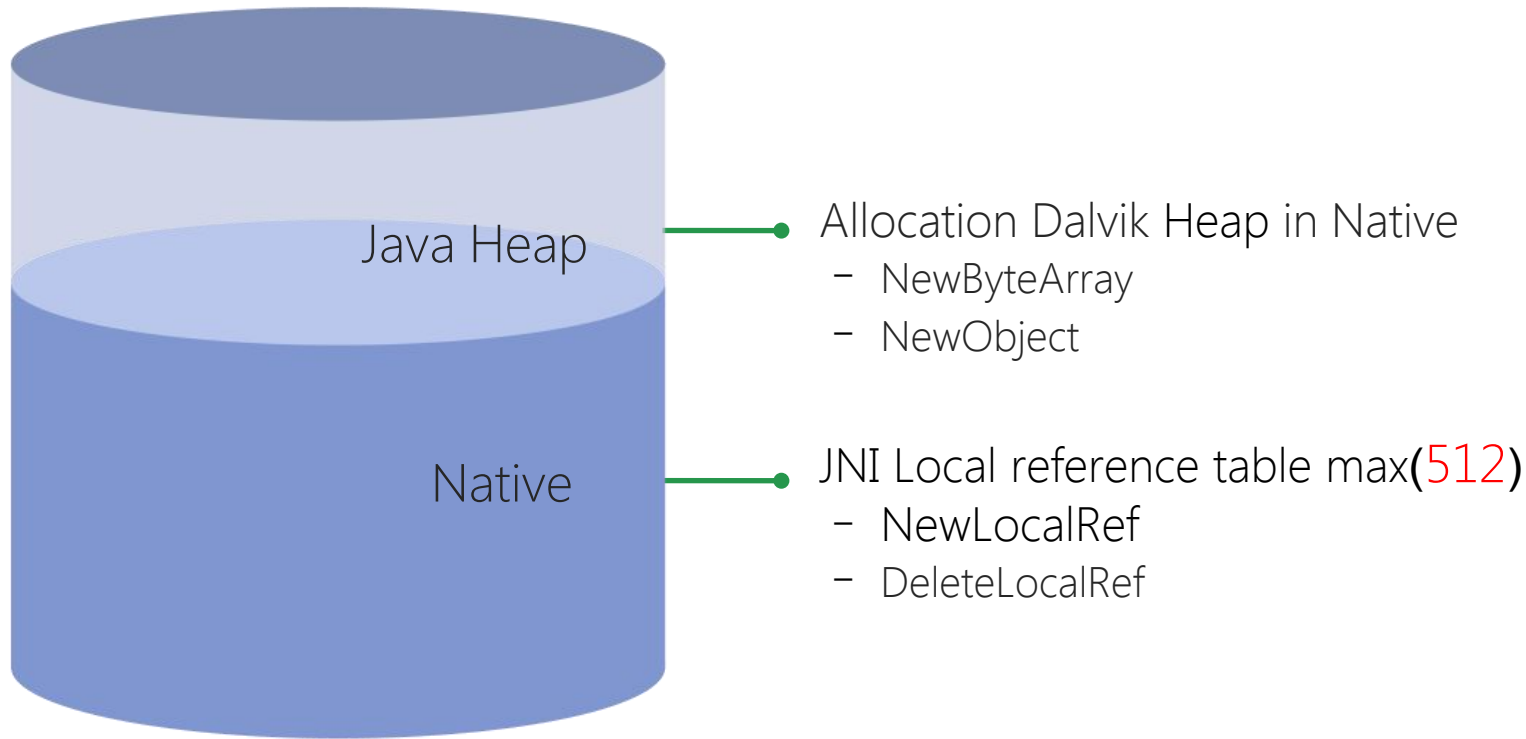
```
bool MessageReceiveListener::OnMessageReceived(const base::Message& message) {
    bool handled = true;

    CC_BEGIN_MESSAGE_MAP(MessageReceiveListener, message)
        CC_MESSAGE_HANDLER(Param0, OnMethod0)
        CC_MESSAGE_HANDLER(Param1, OnMethod1)
        CC_MESSAGE_HANDLER(Param2, OnMethod2)
        CC_MESSAGE_HANDLER(Param3, OnMethod3)
        CC_MESSAGE_HANDLER(Param4, OnMethod4)
        CC_MESSAGE_HANDLER(Param5, OnMethod5)
        CC_MESSAGE_UNHANDLED(handled = false)
    CC_END_MESSAGE_MAP

    return handled;
}
```

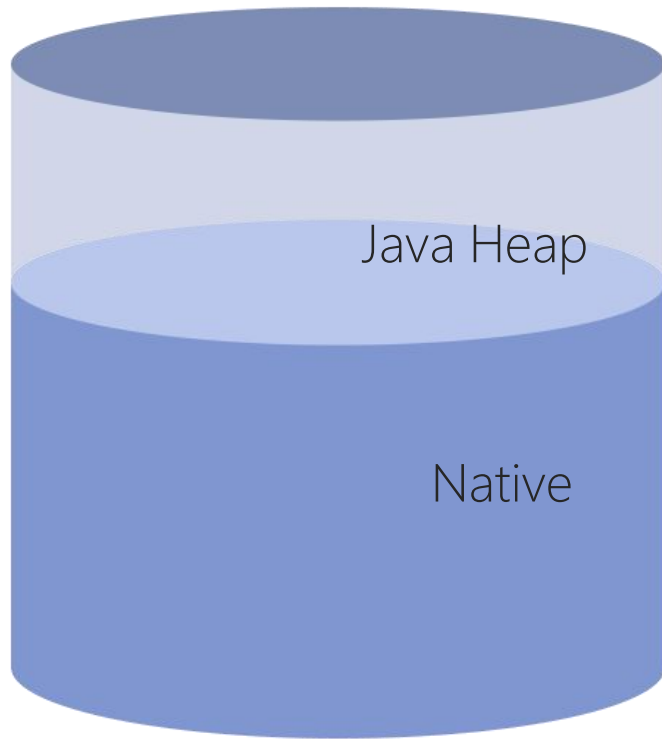

Internal of Android NDK

Android JNI Features



- GetByteArrayRegion instead of GetByteArrayElement&ReleaseByteArrayElements
- Provide Scope Object & JNI Helper Function to avoid memory leak

Android JNI Features



Java Heap

• Android Threads
– Struct Thread

Native

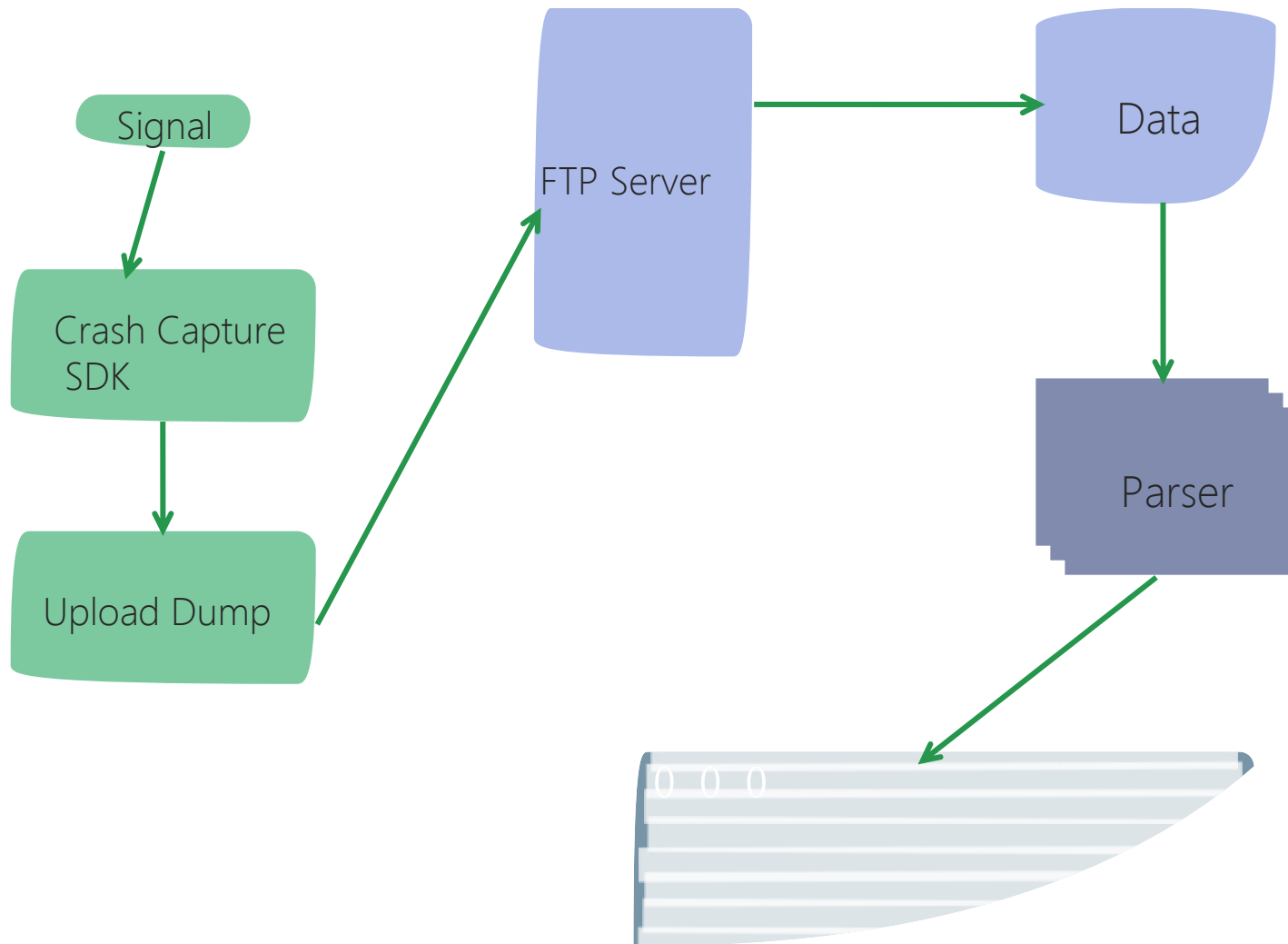
• BionicThreads
– VMThread Objects

- AttachCurrentThread
- DetachCurrentThread

Android Native Crash Capture

Google Breakpad

Native Crash Analysis Platform



Operating system: Android 4.4.4 Linux 3.4.42-ga98722d #1 SMP PREEMPT Thu Jun 11 12:50:29 CDT 2015 armv7l
CPU: arm
ARMv7 ARM part(0x4100c070) features: swp,half,thumb,fastmult,vfpv2,edsp,neon,vfpv3,tls,vfpv4,idiva,idivt
4 CPUs

Crash reason: SIGABRT
Crash address: 0x24da
Process uptime: not available

Thread 40 (crashed)

```
0 libc.so + 0x3b764
  r0 = 0x00000000    r1 = 0x000025b9    r2 = 0x00000006    r3 = 0x00000000
  r4 = 0x9f89ddd8    r5 = 0x00000006    r6 = 0x0000000b    r7 = 0x0000010c
  r8 = 0x00000000    r9 = 0xb86df9b0    r10 = 0xb910b3f0   r12 = 0x000025b9
  fp = 0x00000001    sp = 0x9f88da50    lr = 0xb6f3a691    pc = 0xb6f5f764
Found by: given as instruction pointer in context
1 libc.so + 0x172a1
  sp = 0x9f88da68    pc = 0xb6f3b2a3
Found by: stack scanning
30 libtalk.so!platform::android::ToJavaByteArray [jni.h : 901 + 0x5]
  sp = 0x9f88dc40    pc = 0xa4f6eafd
Found by: stack scanning
31 libtalk.so!platform::android::CCNativeClient::onNativeContent [ccnative_client.cpp : 41 + 0xd]
  r4 = 0xb90527d8    r5 = 0x9f88dc60    r6 = 0x00080039    r7 = 0x002007fd
  r8 = 0x0000000c    sp = 0x9f88dc58    pc = 0xa4f6e8ab
Found by: call frame info
32 libtalk.so!platform::android::AndroidHandlerClient::notify [android_handler_client.cpp : 106 + 0x9]
  r4 = 0xa500d1f0    r5 = 0xb8955990    r6 = 0xb8f8e030    r7 = 0x00080039
  r8 = 0x0000000c    r9 = 0x00000008    sp = 0x9f88dc88    pc = 0xa4f6a697
Found by: call frame info
33 libtalk.so!content::HandlerClientWrapper::notify [handler_client_wrapper.cpp : 23 + 0x5]
  r4 = 0xa4f6a655    r5 = 0xb8a97528    r6 = 0xb8f7d4d0    r7 = 0x9f88dd84
  r8 = 0xb910b0c8    r9 = 0x00000008    sp = 0x9f88dca0    pc = 0xa4f79ee3
Found by: call frame info
34 libtalk.so!network::CmdDispatcher::Recv [cmddispatcher.cpp : 191 + 0x7]
  r4 = 0xa4f76bbd    r5 = 0xb8a97528    r6 = 0xb8f7d4d0    r7 = 0x9f88dd84
  r8 = 0xb910b0c8    r9 = 0x00000008    sp = 0x9f88dca8    pc = 0xa4f6e433
Found by: call frame info
35 libtalk.so!network::CmdDispatcher::Run [cmddispatcher.cpp : 90 + 0xd]
  r4 = 0xb8f7d4d0    r5 = 0x9f88dd88    r6 = 0x0000000d    r7 = 0xa500cdd0
  r8 = 0x9f88dd04    r9 = 0x0000003c    r10 = 0x9f88dcfc   fp = 0x9f88dcf8
  sp = 0x9f88dce0    pc = 0xa4f6e60d
Found by: call frame info
```

Crash Backtrace

Thread 0

```
0 libc.so + 0x126b8
  r0 = 0xb8e92c8     r1 = 0x00000000    r2 = 0x00000062    r3 = 0x00000000
  r4 = 0x00000000    r5 = 0x00000000    r6 = 0x00000000    r7 = 0x000000f0
  r8 = 0x00000062    r9 = 0xb8e92c8     r10 = 0xb8e92c8     r12 = 0xb8e92c8
  fp = 0xb5176b40    sp = 0xb8e92c8     lr = 0xb4f11e4f     pc = 0xb6f366b8
```

Other Backtrace

```
0xb6c73000 - 0xb6c76fff libETC1.so ???
0xb6c7b000 - 0xb6d35fff libstlport.so ???
0xb6efd000 - 0xb6f00fff libstdc++.so ???
0xb6f24000 - 0xb6f7dfff libc.so ??? (WARNING: No symbols, libc.so, 11DB1157052C7A1422DDB89347B8313B0)
0xb6fa0000 - 0xb6fa1fff app_process32 ??? (main) (WARNING: No symbols, app_process32, AB70C4D488DCC368EF43882516E3C6C90)
```

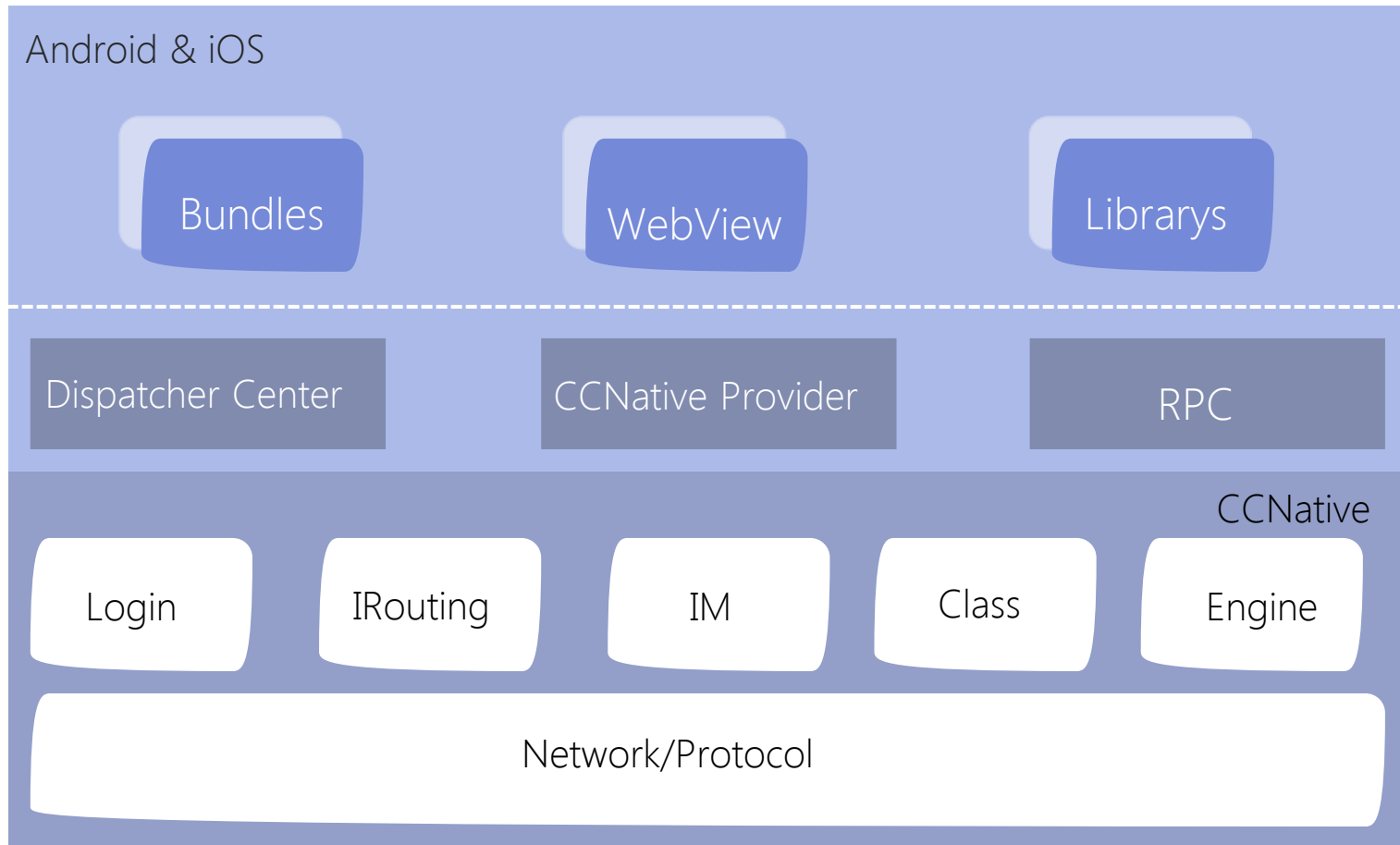
Why ???

Pure-native

WITHIN



CCTalk Architecture



沪江-移动&前端技术沙龙



THANKS