black hat ASIA 2020

OCTOBER 1-2, 2020
BRIEFINGS

There are Apps in Apps Here is How to Break Them

Ronny Xing

Tencent 腾讯

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

> Whoami

- RonnyXing (@RonnyX2017)
- Web & Android Security Research
- Pentester
- At Tencent Security Xuanwu Lab

> Agenda

1. What is “Instant App” outside and inside
2. Architecture of WebView based instant app
3. Vulnerabilities in the concrete implementation
4. Architecture of native android instant app
5. Vulnerabilities in the Google Play Instant

key terms definition

Instant app $\longrightarrow$ There are Apps in Apps Here is How to Break Them

key terms definition

There are

Apps in Apps ⟵ Supervisor app

Here is

How to Break Them

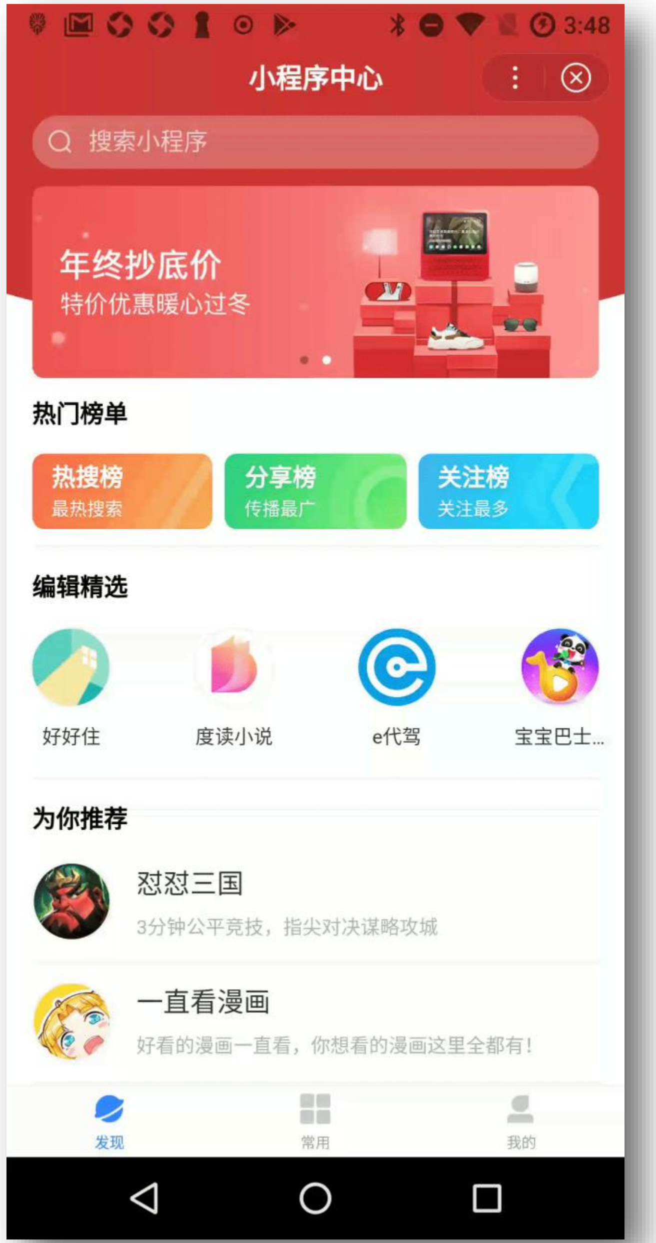
> Agenda

Background

- Mobile Web adaptation: Facebook Instant Articles / Google AMP / …
- Mobile Hybrid dev: React Native / …

- “Instant App”
  - Google I/O 2016: release Google Instant App
  - Various Webview Based Instant Apps released
  - Apple WWDC 2020: release App Clips

What Instant Apps look like



小程序中心

搜索小程序

年终抄底价
特价优惠暖心过冬

热门榜单

热搜榜 最热搜索
分享榜 传播最广
关注榜 关注最多

编辑精选

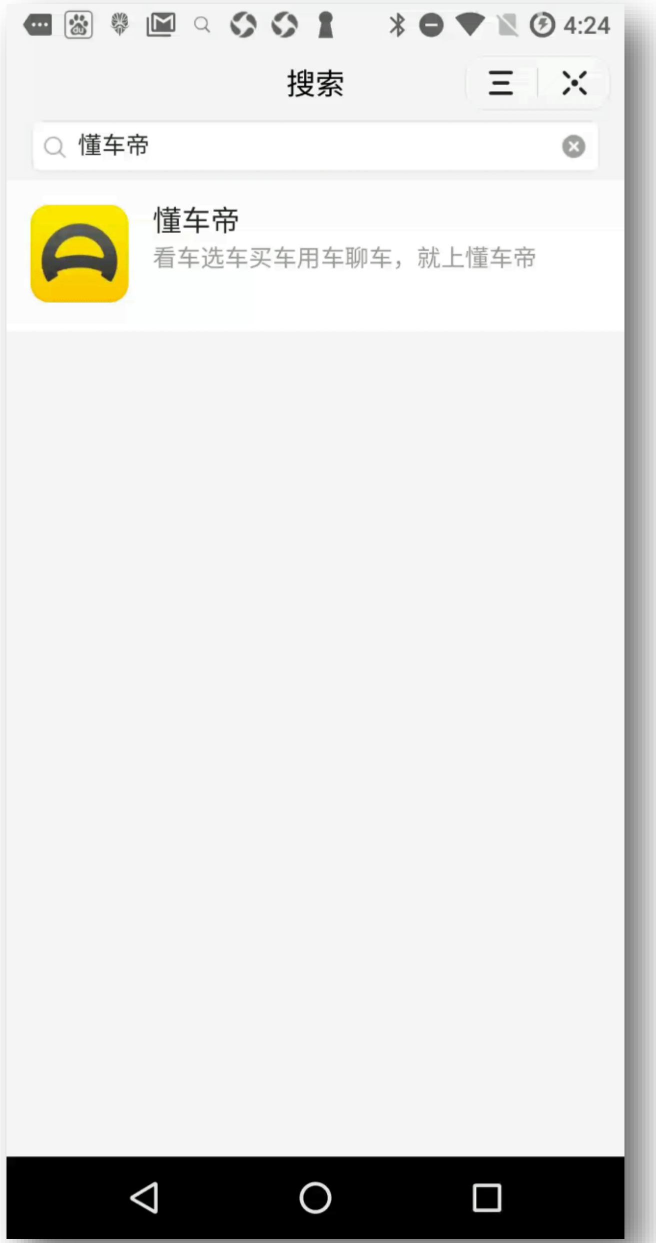
好好住
度读小说
e代驾
宝宝巴士...

为你推荐

怼怼三国
3分钟公平竞技，指尖对决谋略攻城

一直看漫画
好看的漫画一直看，你想看的漫画这里全都有！

发现 常用 我的

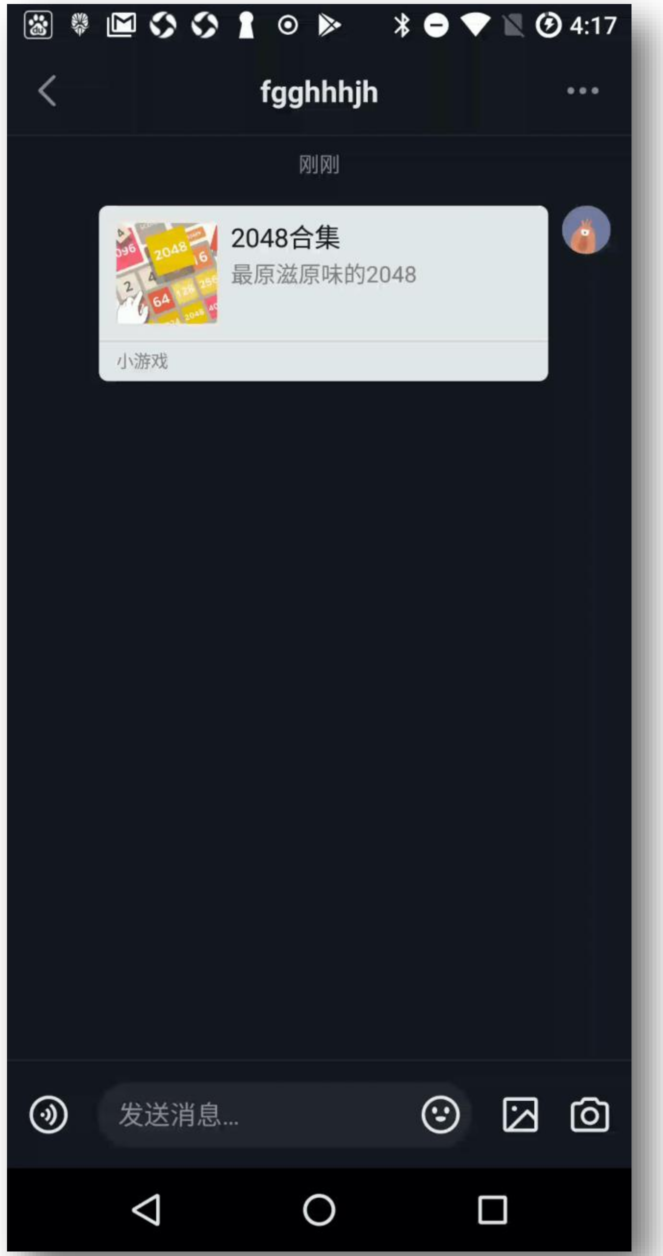


搜索

懂车帝

懂车帝
看车选车买车用车聊车，就上懂车帝

What Instant Apps look like



4:17

fgghhhjh

刚刚

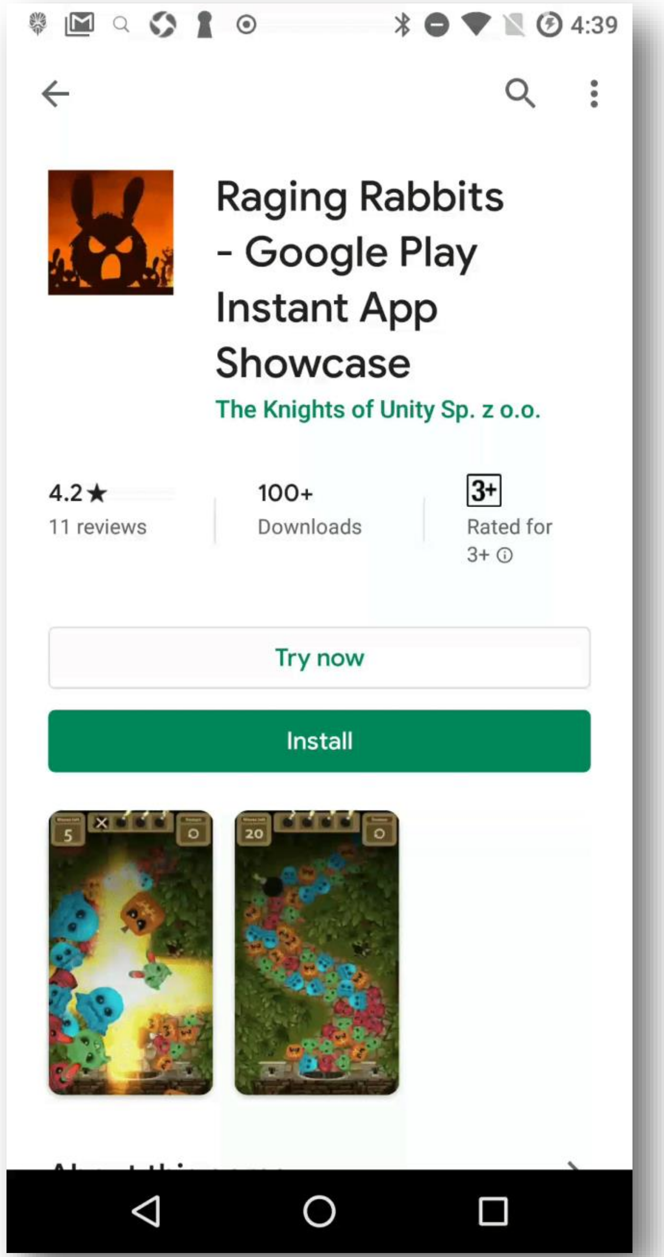
2048合集

最原滋原味的2048

小游戏

发送消息...

What Instant Apps look like



4:39

Raging Rabbits - Google Play Instant App Showcase

The Knights of Unity Sp. z o.o.

4.2★	100+	3+
11 reviews	Downloads	Rated for 3+ ⓘ

Try now

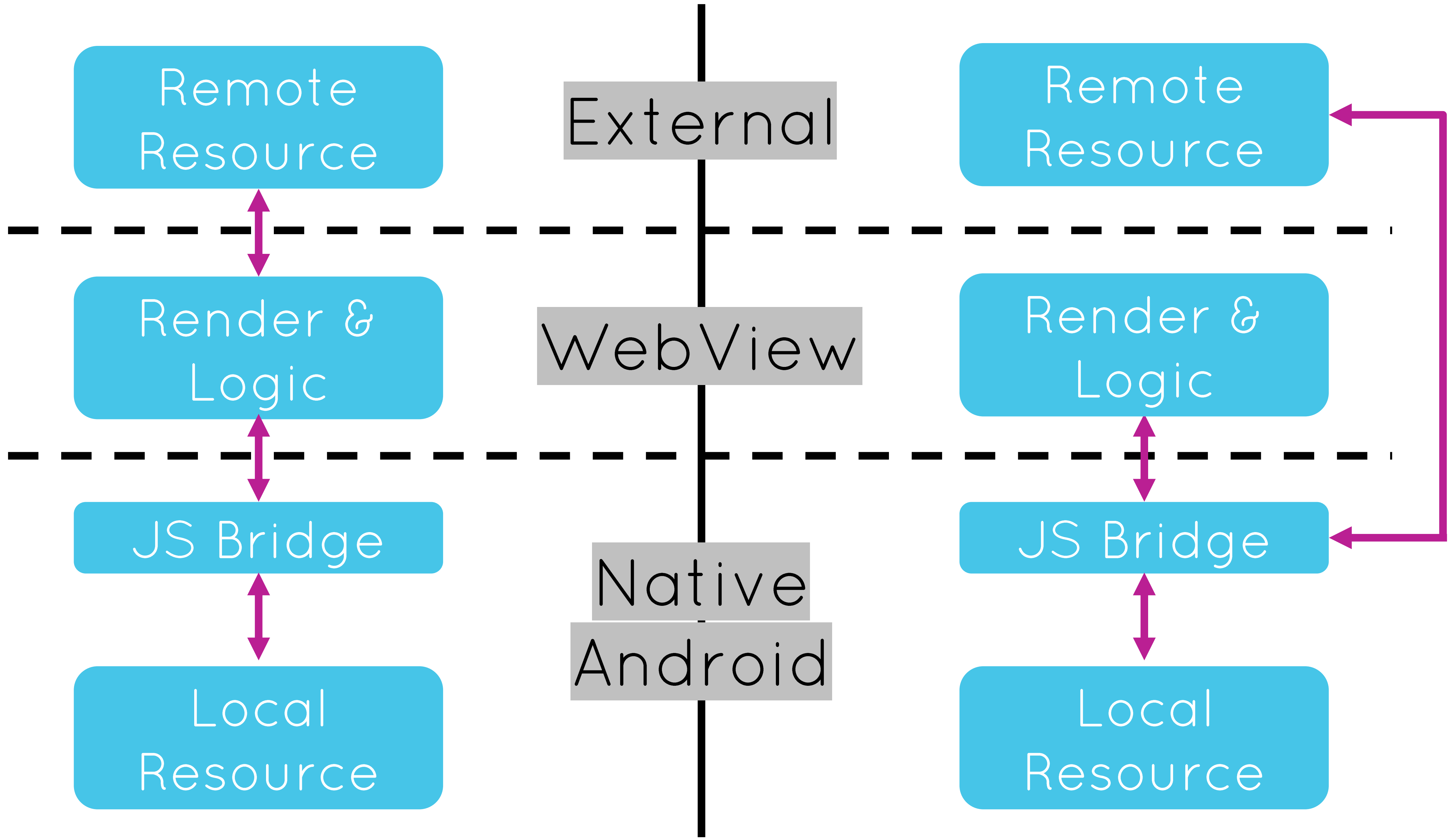
Install

What is inside

- WebView-based architecture
- Hybrid Mobile App Development
- Module Loading
  - Dynamic
  - Remote
  - From third party

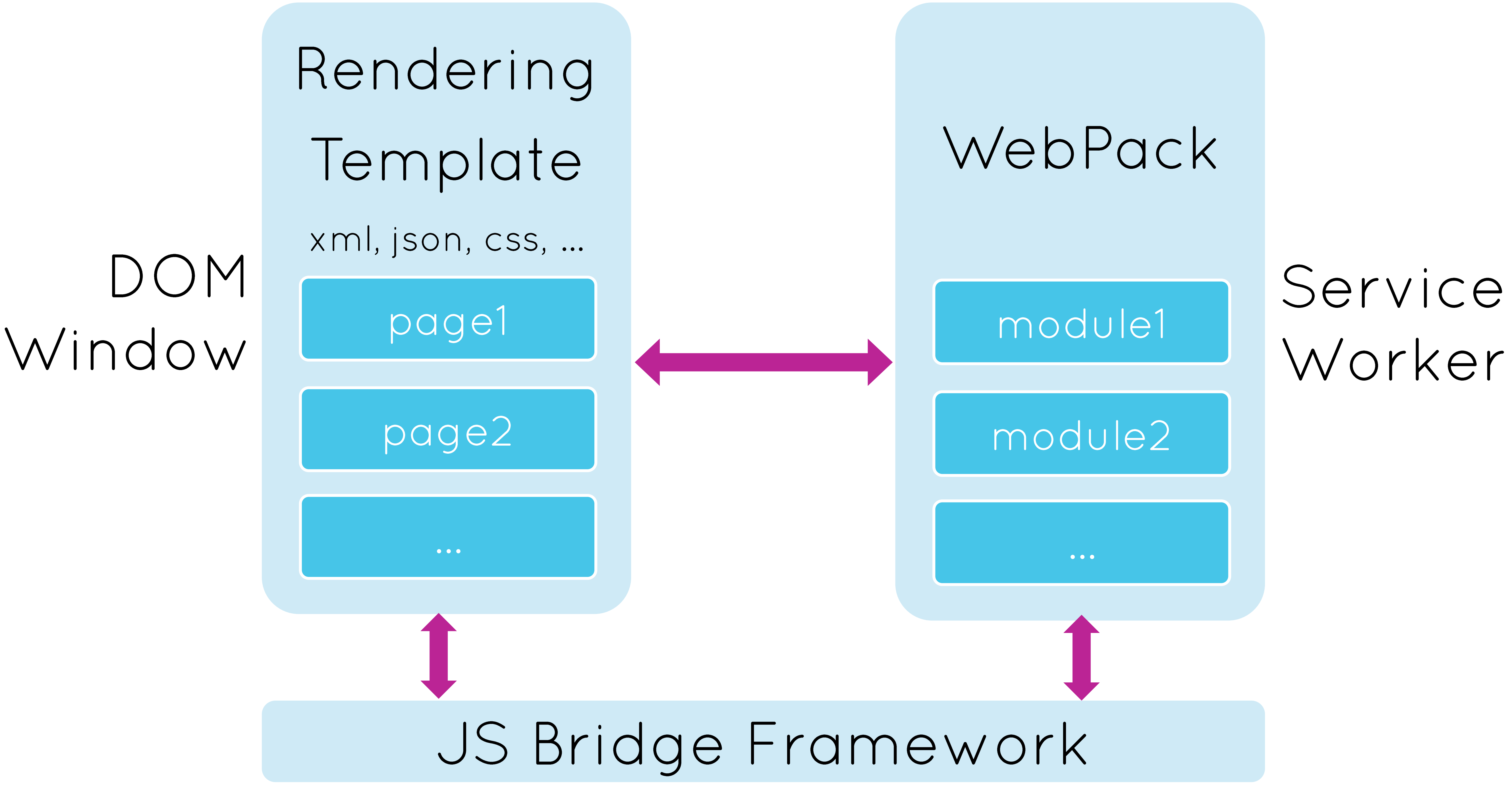
WebView
VS
WebView-Based Instant App

Webview vs WBIA



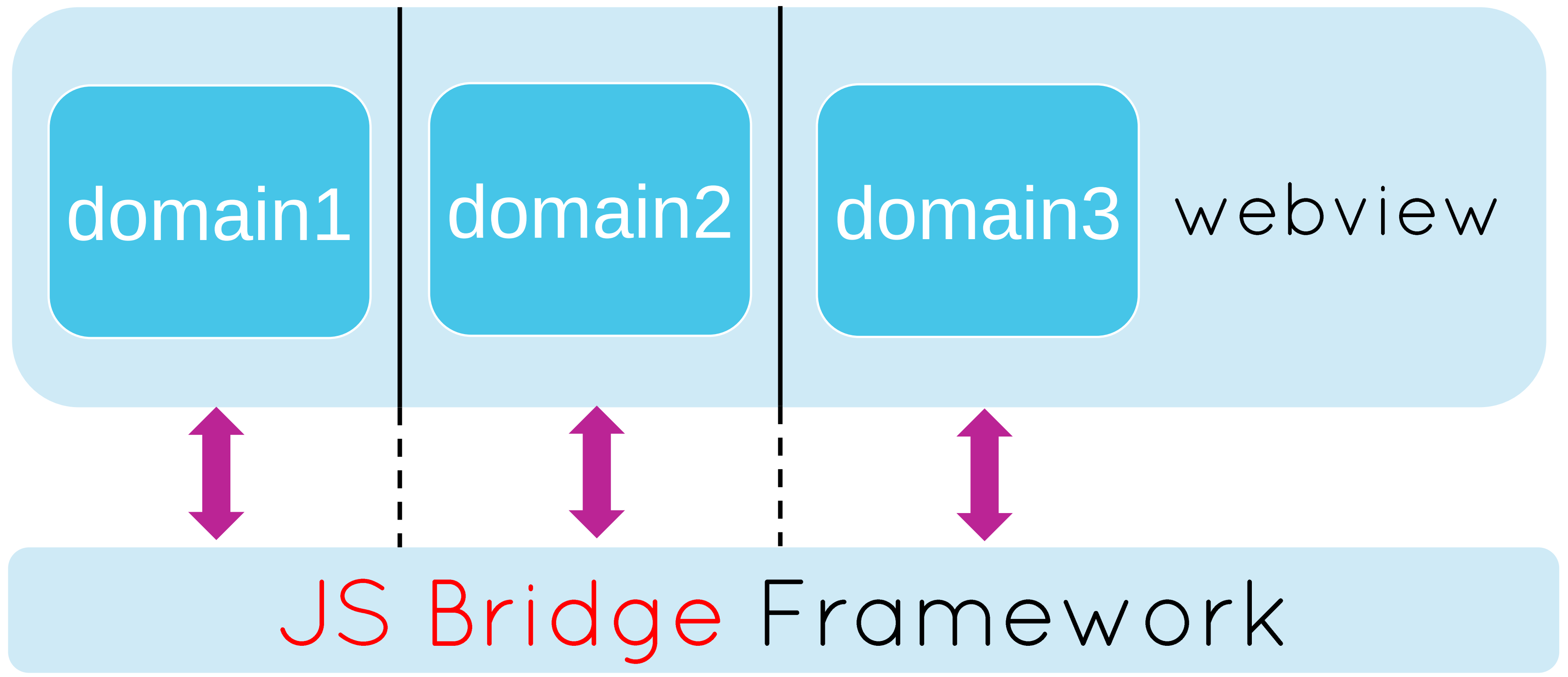
> Agenda

WebView in WBIA



One Instant App is One Domain

One Instant App is One Domain



Classic WebView JS Bridge

- JavaScript Interface
- WebView events handler

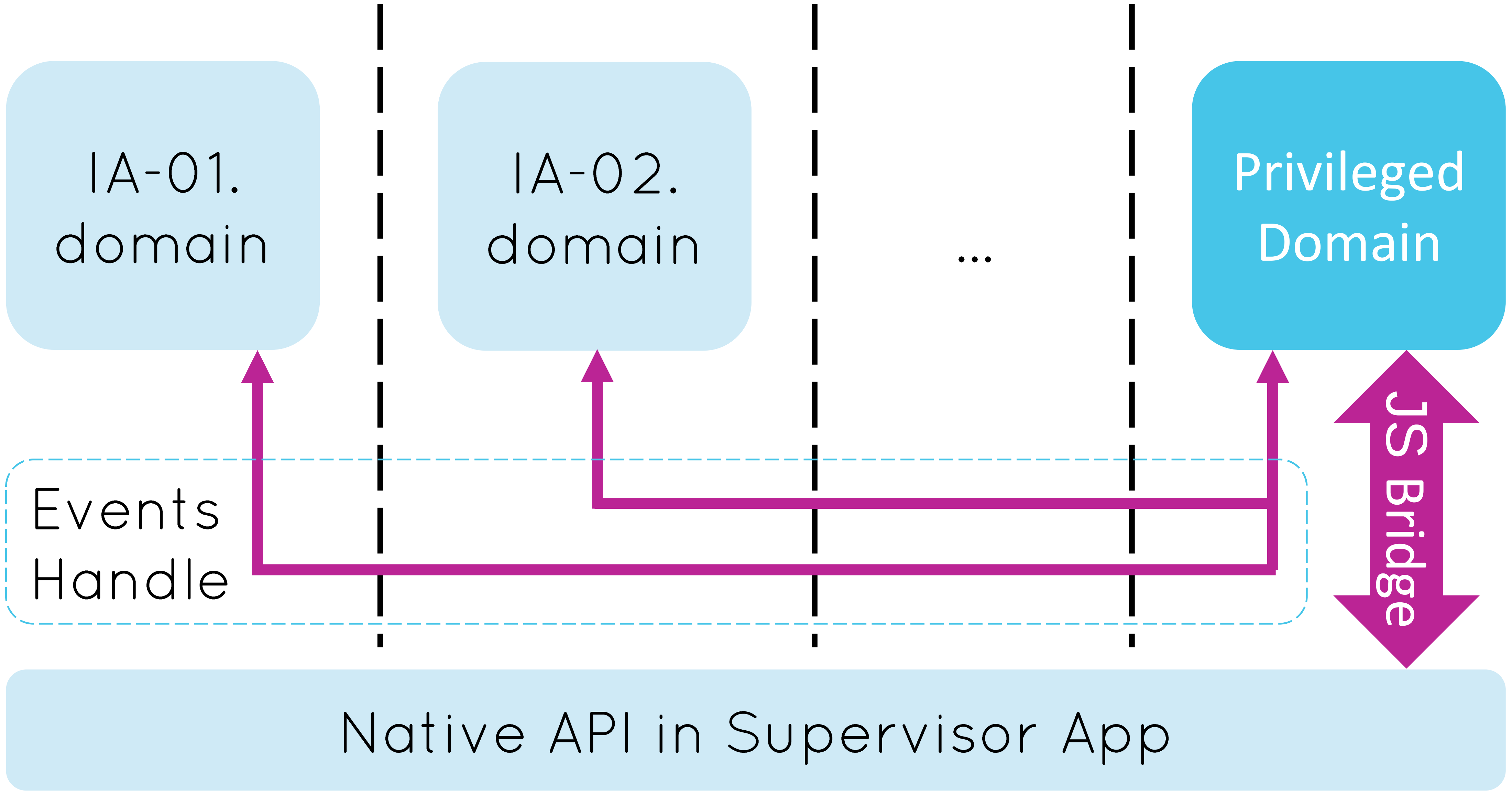
Hard to implement in Classic WebView

- Multiple domains with multiple pages run together
- Isolated process
- Asynchronous communication
- External resource restriction

Solutions in WBIA

- Cross domain communication with privileged domain
- Cross domain events handling framework

WBIA JS Bridge

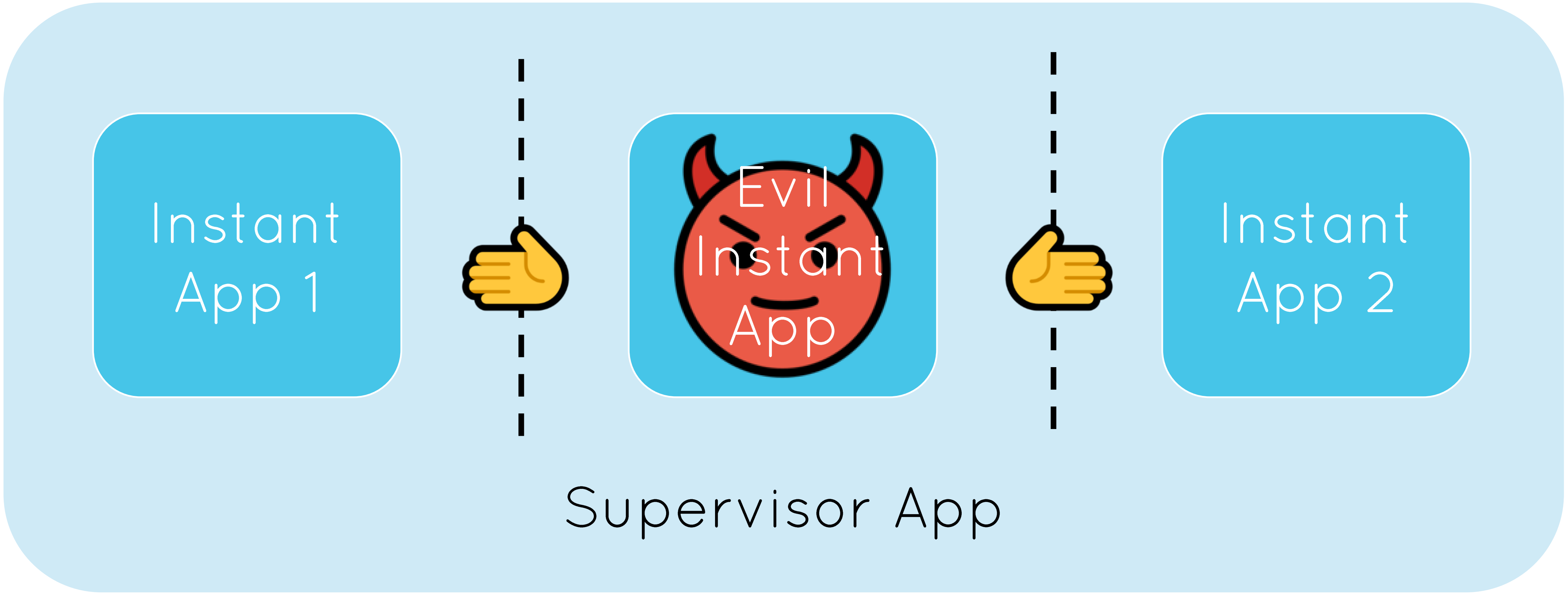


> Agenda

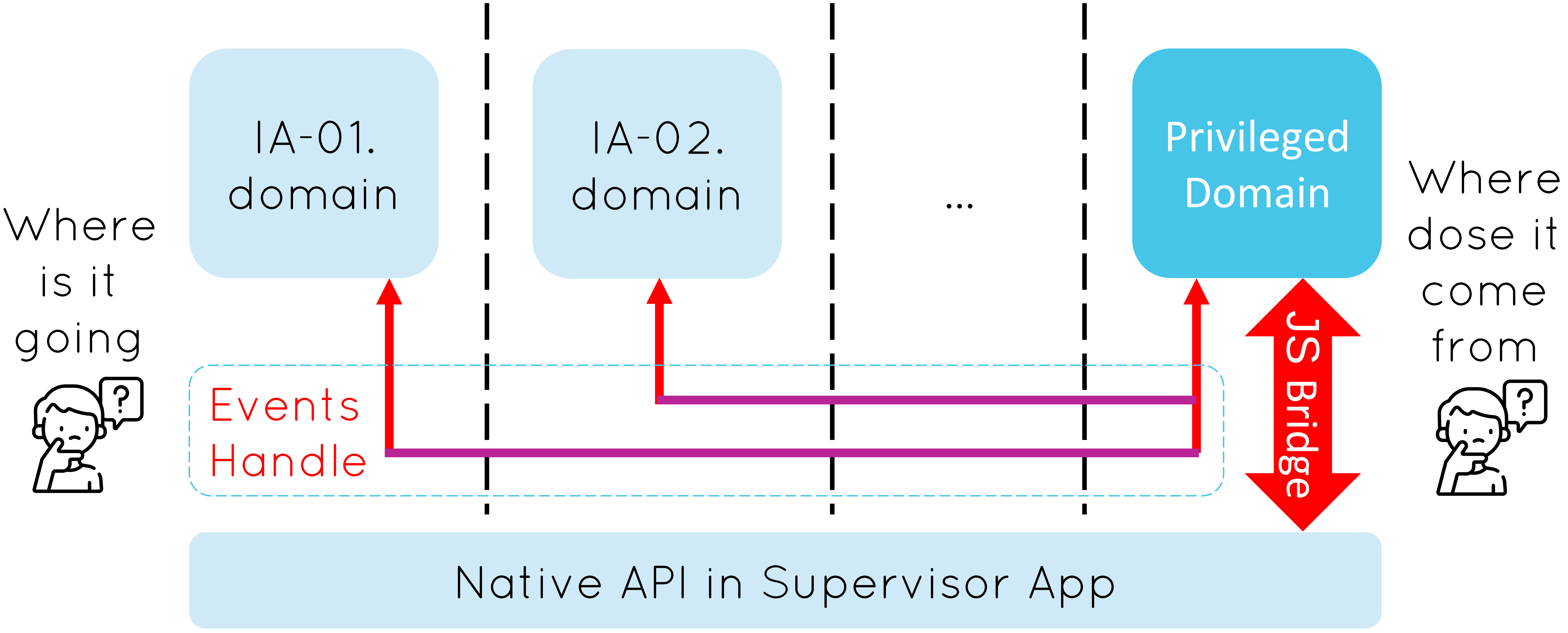
1. What is “Instant App” outside and inside
2. Architecture of WebView based instant app
3. Vulnerabilities in the concrete implementation
4. Architecture of native android instant app
5. Vulnerabilities in the Google Play Instant

Target

- Lateral movement:
  Compromise isolation between instant apps



WBIA JS Bridge

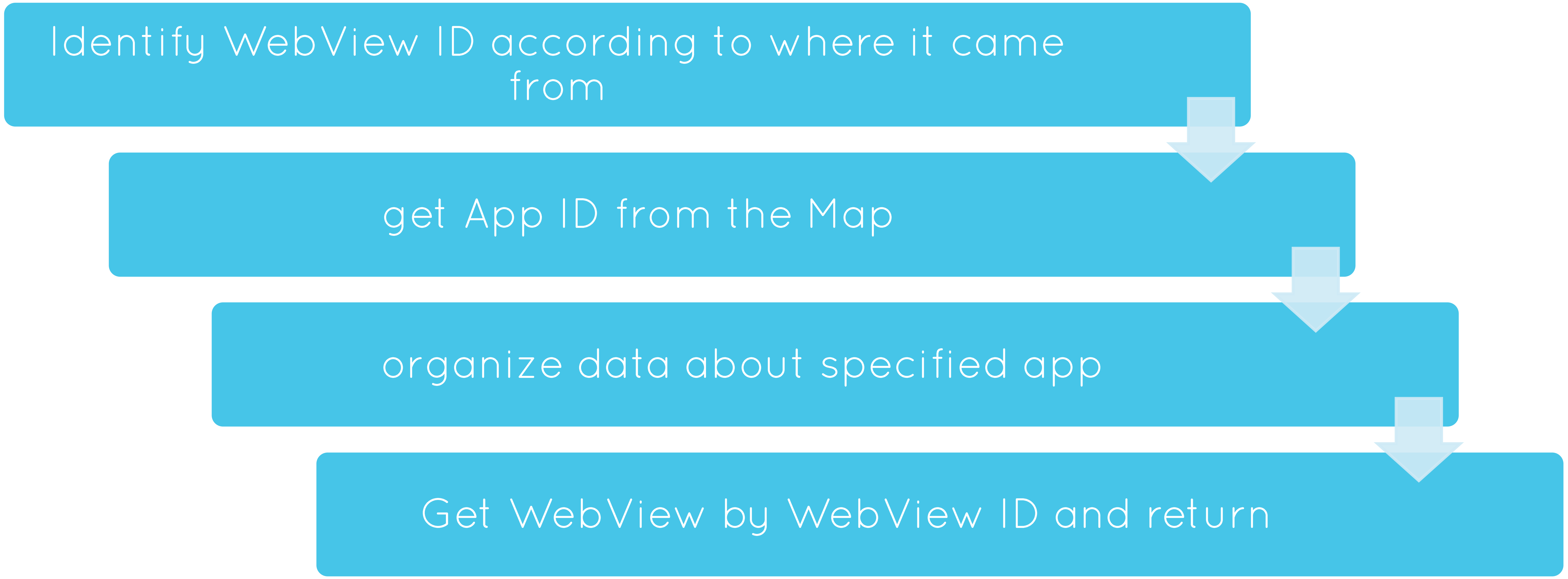


Attack Surfaces

- JS Bridge requests' source identification
- Events callback

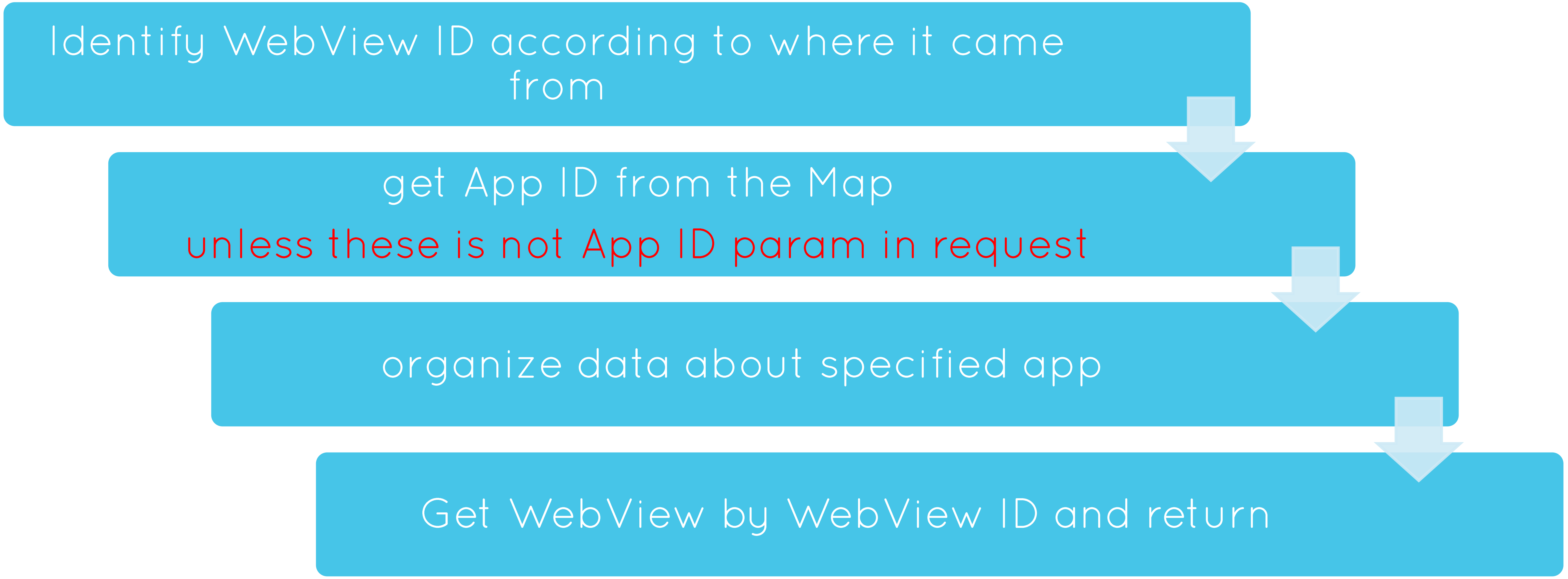
Identification

- Key structure:
  - APP ID – WebView ID Map


Identify WebView ID according to where it came from
get App ID from the Map
organize data about specified app
Get WebView by WebView ID and return

Identification

- Key structure:
  - APP ID – WebView ID Map



Identify WebView ID according to where it came from

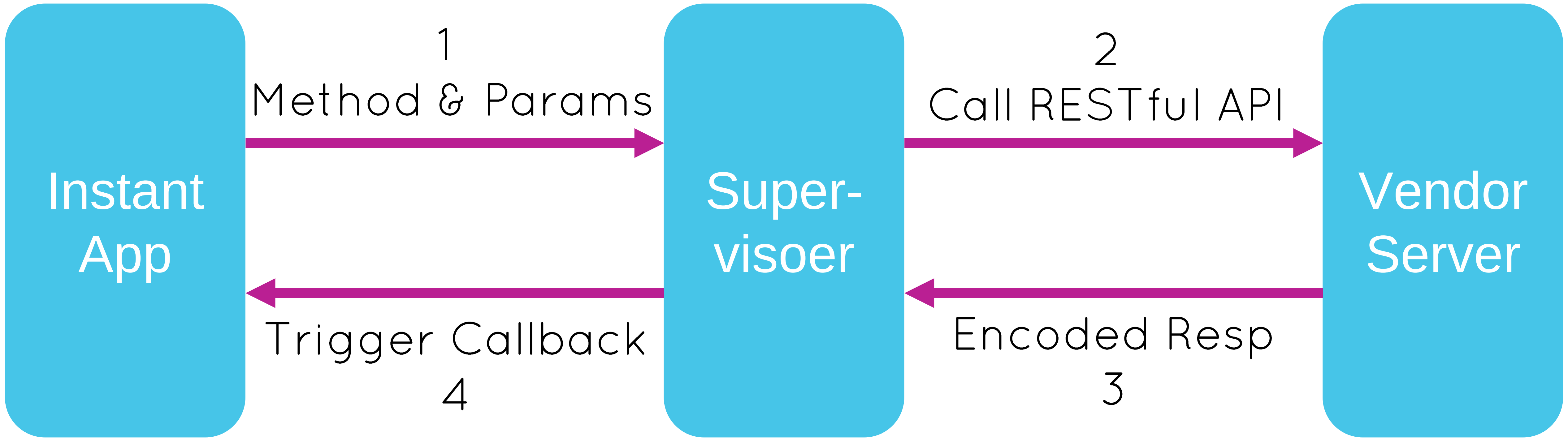
get App ID from the Map

unless these is not App ID param in request

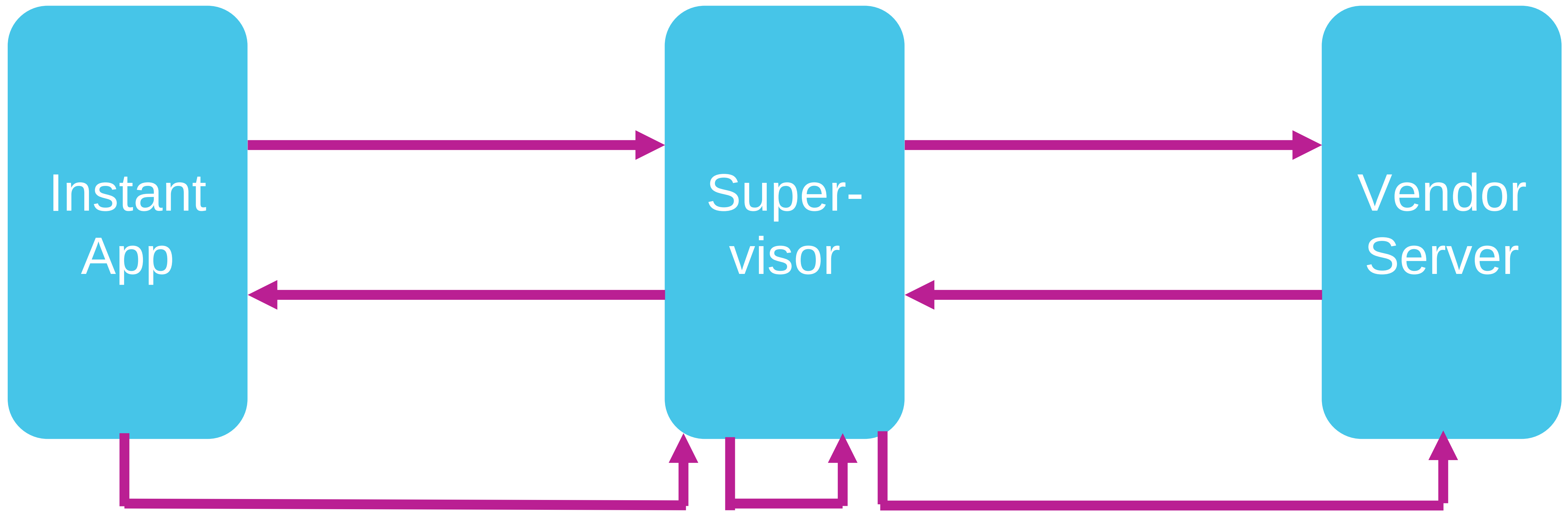
organize data about specified app

Get WebView by WebView ID and return

RPC


Instant App
Super-visoer
Vendor Server
1
Method & Params
2
Call RESTful API
Encoded Resp
3
Trigger Callback
4

XRPC

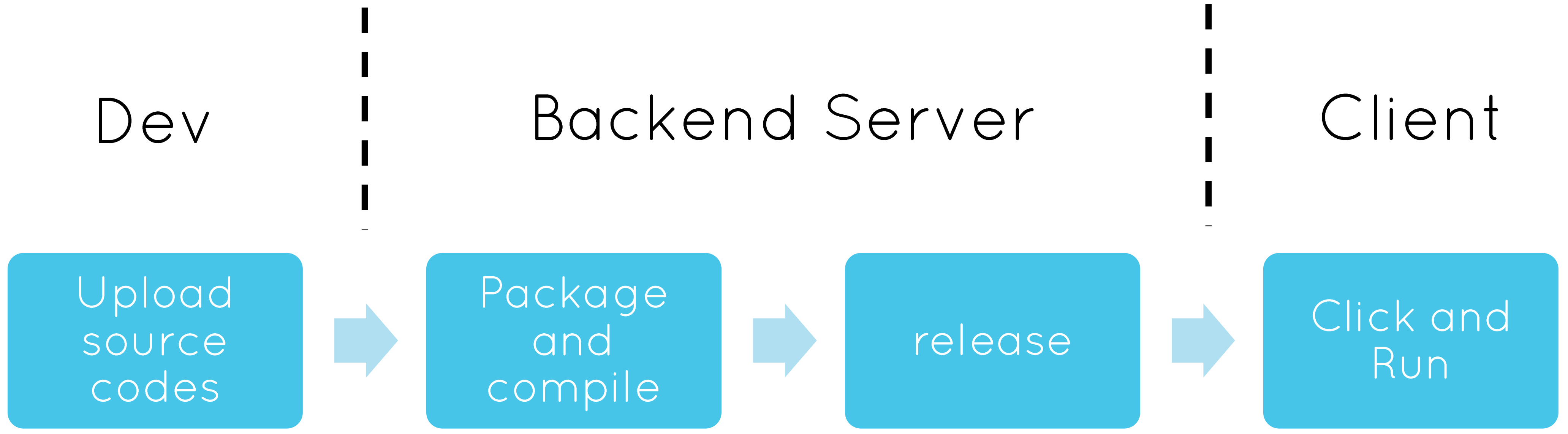


```
Method: XRPC
Params: {
  optType: RPC_interface
  data: ...
}
```

Instantiate RPC interface

Call RESTful API

Exploit



- Cloud Packager
  - Package the payload in the remote black box
  - Security check and filter

- JS Sandbox
  - Module in webpack
  - Limits for cross domain request

Cross Domain Request

Fetch

- In DOM Window / Service Worker
- Cant get response data
    - callback event cant be registered by attacker

importScripts

- Only in Service Worker
- Need special format response data to trigger callback function

Security Measures

- Project codes are packaged as some modules in webpack (like bundle.js)

- Blacklist for sensitivity function and object, they will be replace to (void 0)

- More mitigation:
  - Objects localization

Black List

Sensitive functions:

- eval / fetch / importscripts / Function

Sensitive objects:

- *Global Object* has all we need
  - this / self / window / global / thisGlobal

Bypass Black List

How webpack organize bundle.js ?

A simple Webpack demo

webpack.config.js

```
module.exports = {
    entry: './app/index.js', // enter file
    output: {
        path: path.resolve(__dirname, 'build'), // output dir
        filename: "bundle.js", // output file name
        publicPath: 'build/' // pack dir
    },
    module: {...}
}
```

./app/index.js

```
import sum from './sum'
import './addImage'
console.log(sum(1, 2))
```

./app/sum.js

```
export default (a, b) => {
    return a + b
}
```

bundle.js

```
(function(modules) { // webpackBootstrap
    var installedModules = {}; // The module cache
    function __webpack_require__(moduleId) { // The require function
        ...
        if(...)
            return installedModules[moduleId].exports;
        ...
    }
})
([
    Module0,
    Module1,
    ...
]);
```

bundle.js

```
([
    /* 0 */
    (function(module, exports, __webpack_require__) {
        "use strict";
        var _sum = __webpack_require__(1);
        var _sum2 = _interopRequireDefault(_sum);
        __webpack_require__(2);
        function _interopRequireDefault(obj) { ... }
        console.log((0, _sum2.default)(1, 2));
    }),
    /* 1 */
    (function(module, exports, __webpack_require__) {
        "use strict";
        Object.defineProperty(exports, "__esModule", { value: true });
        exports.default = function(a, b) {
            return a + b;
        };
    }),
    ... // 2, 3, 4 ...
]);
```

Search Modules exports

- Obscure after package
  - Local variable → a, b, c …
  - arguments[2] → __webpack_require__

- Find a module export *global, self, window* , etc.

```
for (var index = 0; index < 200; index++) {
    if(arguments[2](index)["impo"+"rtSc"+"ripts"]){
      globalIndex = index;
      break;
    }
}
```

Objects Localization

- sensitive functions & objects will be moved into a private module, become local variable, when webpack bootstrap.

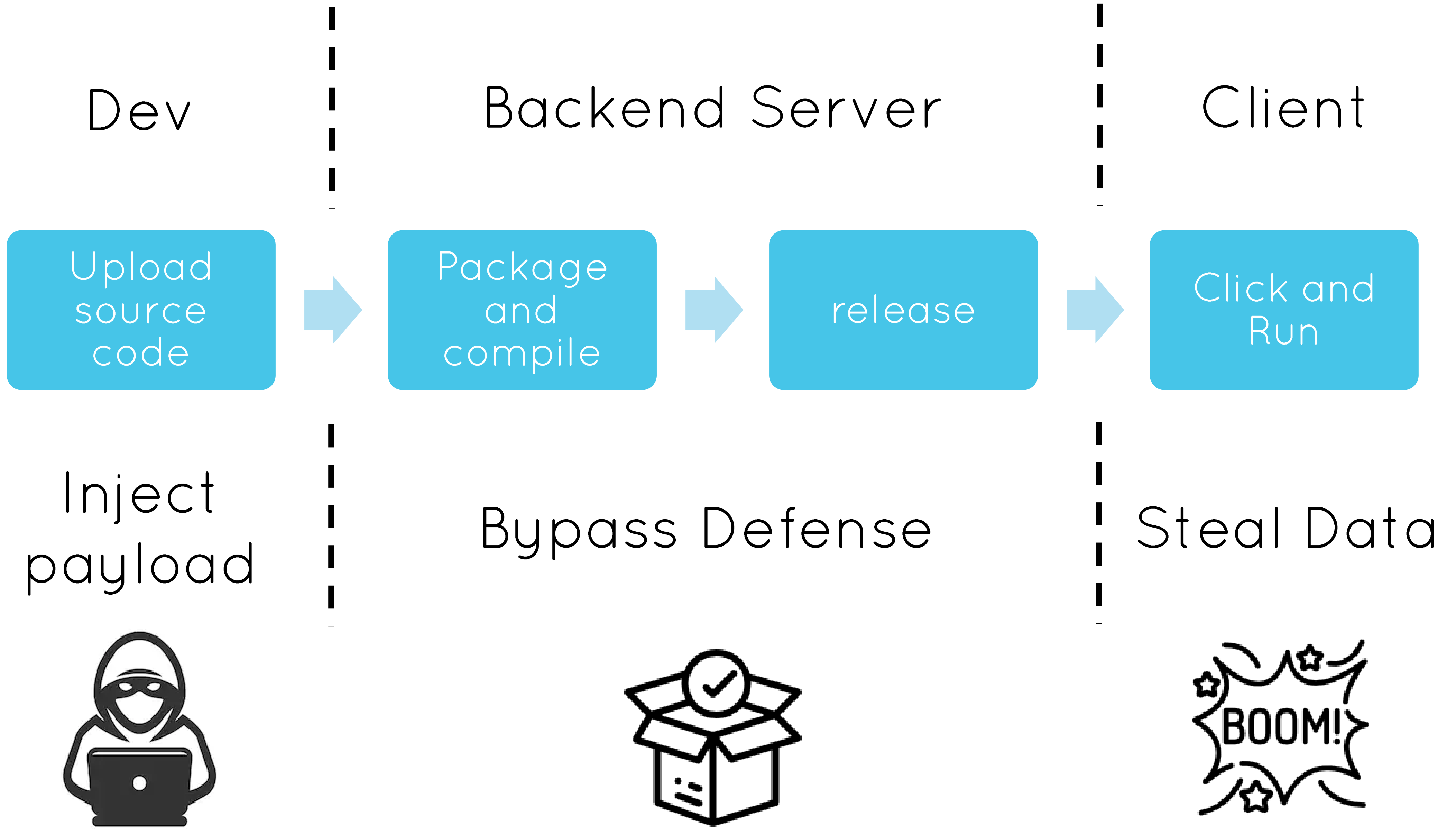
```
function blank(){ }
exports.c = (function(){
    let a = {};
    a.a = globalThis.importScripts;
    globalThis.__proto__.importScripts = blank
    return function(x, y, z){
        a.a(x);
        ...
    }
})();
```

Key Objects Localization

- this / globalThis / self in ServiceWorker is *ServiceWorkerGlobalScope*
- *importscripts* is inherited from *WorkerGlobalScope*
- Recover function from prototype:

```
arguments[2](globalIndex)["importScripts"] = WorkerGlobalScope.prototype.importScripts;
```

Exploit


Dev
Backend Server
Client
Upload source code
Package and compile
release
Click and Run
Inject payload
Bypass Defense
Steal Data
BOOM!

> Agenda

black hat ASIA 2020

1. What is “Instant App” outside and inside
2. Architecture of WebView based instant app
3. Vulnerabilities in the concrete implementation
4. Architecture of native android instant app
5. Vulnerabilities in the Google Play Instant

Google Play Instant

- Native Android apps, without the installation

- With Google Play Instant, people can tap to try an app
  without installing it first.

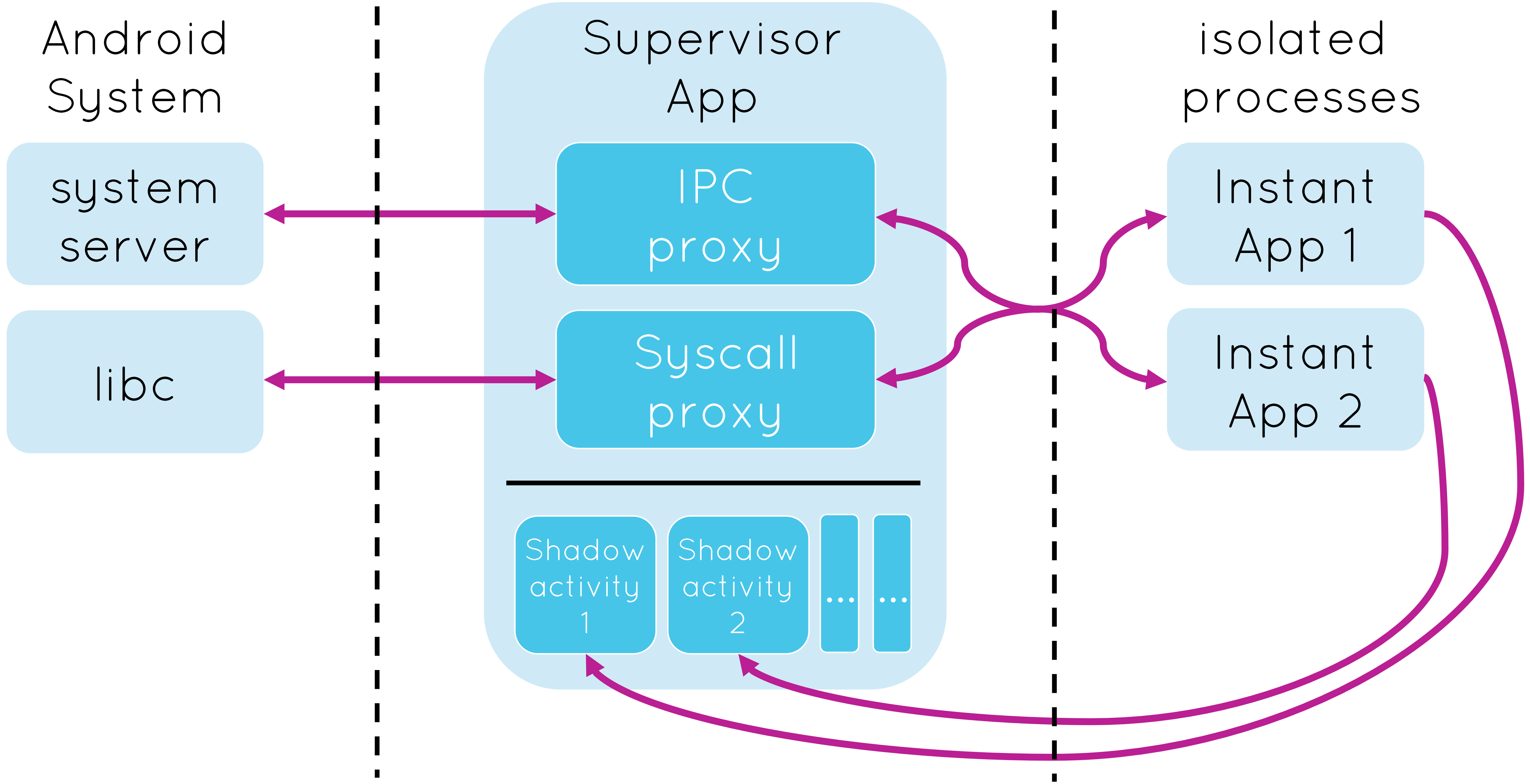
- Increase engagement with your Android app and gain more installs by surfacing your instant app across ..., ..., ...
  anywhere you share a link.

Native & App Bundle! Cool~

- API level >= 26: Google Play Instant app is supported by AOSP(Package Manager Service)
  - Break out? Maybe more difficult than LPE from apps.

- API level < 26:
  - com.google.android.instantapps.supervisor

Inside supervisor



Setup Isolated Process

```
Trace.beginSection("IChildProcessConnection.setupWithApplicationI
nfo");
```

```
com.google.android.instantapps.supervisor.isolatedservice.Isolate
dService.setupWithApplicationInfo(...)
```

Register 3 binder:

① .ipc.ServiceManagerForwarderProxy	→ IPC Proxy
② .syscall.SyscallService	→ Syscall Proxy
③ .event.EventReceiver	→ Events Handler

IPC proxy: IPC Whitelist

- DNA DATA:
  - Location: cache/dna_data/com.google.android.instantapps.dna.archive:39
  - Protobuf in a zip:

```
    // aidl items
    2 <chunk> = message:
        1 <chunk> = "android.app.IActivityManager" // aidl class name
        2 <chunk> = "activity" // aidl alisa name
        3 <chunk> =
"com.google.android.instantapps.supervisor.ipc.proxies.handler.ActivityManagerProxyHandler"
// ProxyHandler
        8 <varint> = 4
        9 <varint> = 1
        // IPC method items
        10 <chunk> = message:
            ……
```

IPC proxy: IPC Whitelist

```
// IPC method items
10 <chunk> = message:
    // method signatures
    1 <chunk> = message:
        1 <chunk> = "getIntentSender" // method name
                3 <chunk> = message:
                    2 <chunk> = message(1 <varint> = 5) // int
                3 <chunk> = message:
                    2 <chunk> = message(1 <varint> = 9) // String
                    3 <chunk> = 5  // parser typo or a flag
                3 <chunk> = message:
                    2 <chunk> = message(1 <varint> = 13) // IBinder
                // other params
                …
                // return type
                4 <chunk> = message:
                    1 <varint> = 16 // No-Predefined class
                    2 <chunk> = "android.content.IIntentSender"
                // flags or typo, but I don’t care
                7 <chunk> = message: …
     // method type
     2 <varint> = 2
```

IPC proxy: IPC Whitelist

The 2nd item of *IPC method item* is the type of IPC method

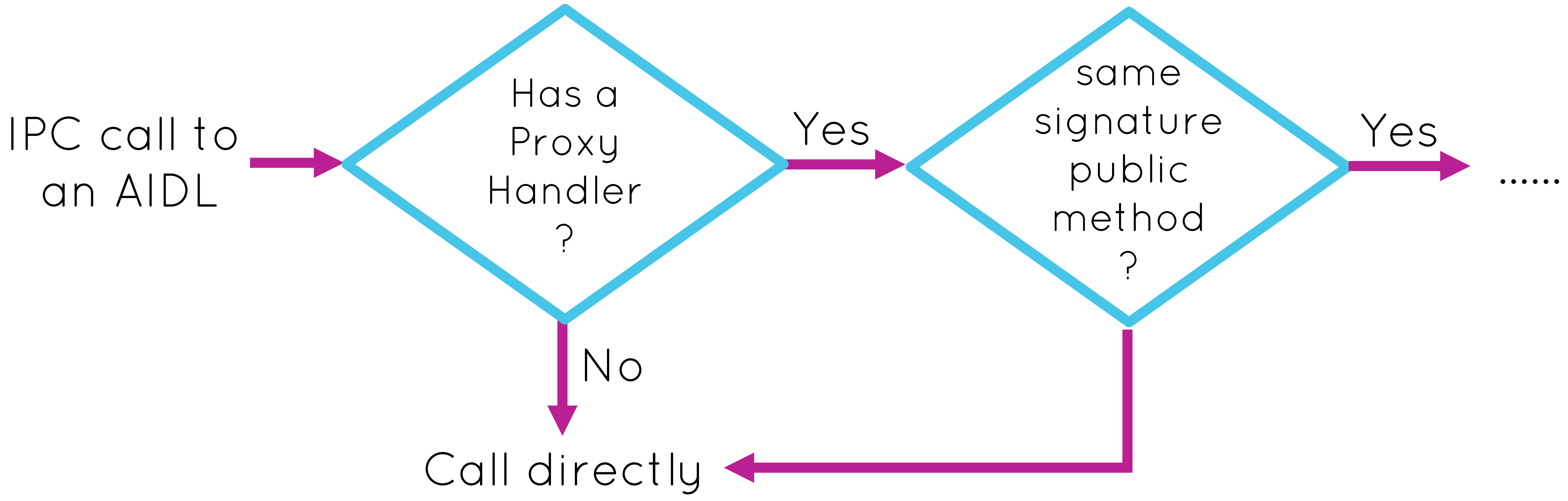
3 <= *type* <= 14

↓

permission check failed

IPC proxy: IPC Whitelist

- The 3th item of *AIDL item* is the class name of `ProxyHandler` to check and forward IPC call

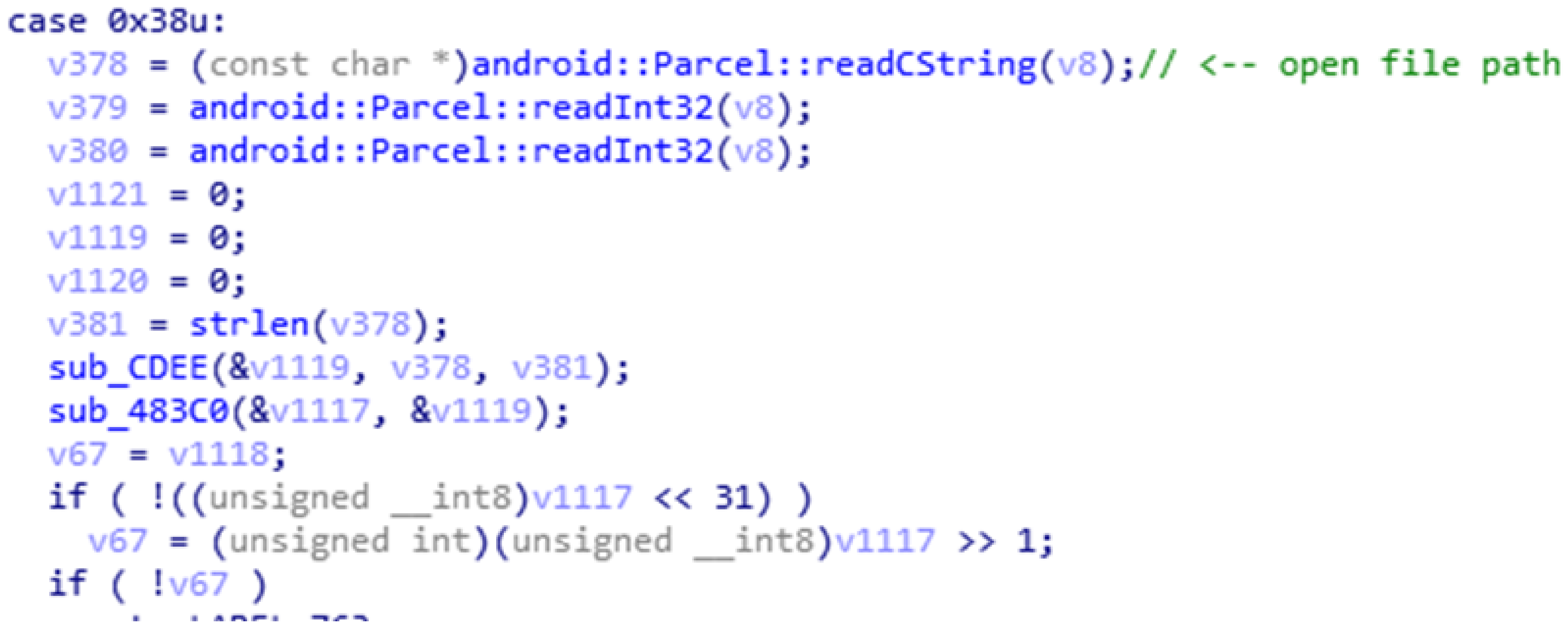


Syscall(libc) Proxy

- Implemented in libsyscall.so
- Some libc calls will be forwarded to the proxy
- Find the `onTransact` method (for example *sub_364E0*)
- There are 88 transact codes for IPC call

Syscall(libc) Proxy

- Example: code 0x38 is open



```
case 0x38u:
  v378 = (const char *)android::Parcel::readCString(v8);// <-- open file path
  v379 = android::Parcel::readInt32(v8);
  v380 = android::Parcel::readInt32(v8);
  v1121 = 0;
  v1119 = 0;
  v1120 = 0;
  v381 = strlen(v378);
  sub_CDEE(&v1119, v378, v381);
  sub_483C0(&v1117, &v1119);
  v67 = v1118;
  if ( !((unsigned __int8)v1117 << 31) )
    v67 = (unsigned int)(unsigned __int8)v1117 >> 1;
  if ( !v67 )
```

> Agenda

1. What is “Instant App” outside and inside
2. Architecture of WebView based instant app
3. Vulnerabilities in the concrete implementation
4. Architecture of native android instant app
5. Vulnerabilities in the Google Play Instant

Components Access

Sandbox for access external components

- Activity: intent forwarded by supervisor, rewrite flags and extra data.
- Service & BroadcastReceiver: only can access Internal components and specific in supervisor
- Content Provider: Internal Only

Target

Bypass sandbox to

Access external components without limits

IntentSender

`android.app.IActivityManager.getIntentSender` will be forwarded to the `ActivityManagerProxyHandler`

Check and forward the intent with flags and extra data rewrote

But, pay attention to the parameters of the method `IntentSender.sendIntent`

IntentSender

```
public void sendIntent (Context context,
        int code,
        Intent intent,
        IntentSender.OnFinished onFinished,
        Handler handler)
```

intent: Additional Intent data. It will be passed to `Intent#fillIn`.

IntentSender

```
public int fillIn (Intent other,
        int flags)
```

Copy the contents of other in to this object, but only where fields are not defined by this object.

Bypass Sandbox

Get an `IntentSender` in whitelist by IPC call

↓

Initialize a new intent with target component

↓

Use send or `sendIntent` to
update the intent of the original `IntentSender`

Other Vulns in Supervisor

- Duplicate provider authority
- ……

Thank You

Q & A

@RonnyX2017

Th3s3v3n.shell@gmail.com