HACKTIVITY 2018

CHEATS FOR EVERYTHING ANDROID

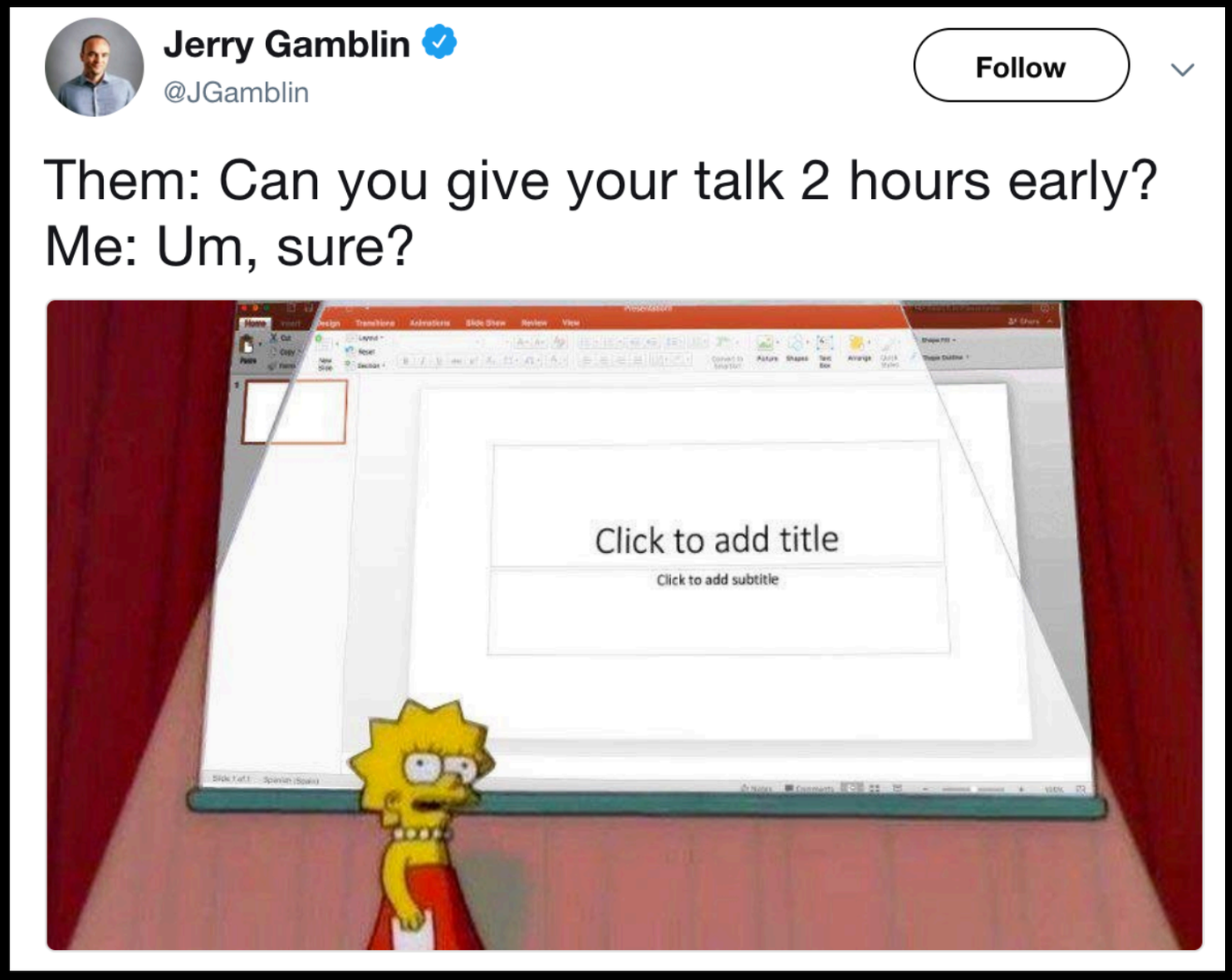
'PROGRAMMING IS SUPER BORING.
GAME CHEATS ARE THE ONLY COOL THINGS
ABOUT CODING.'

- an 11-year-old kid

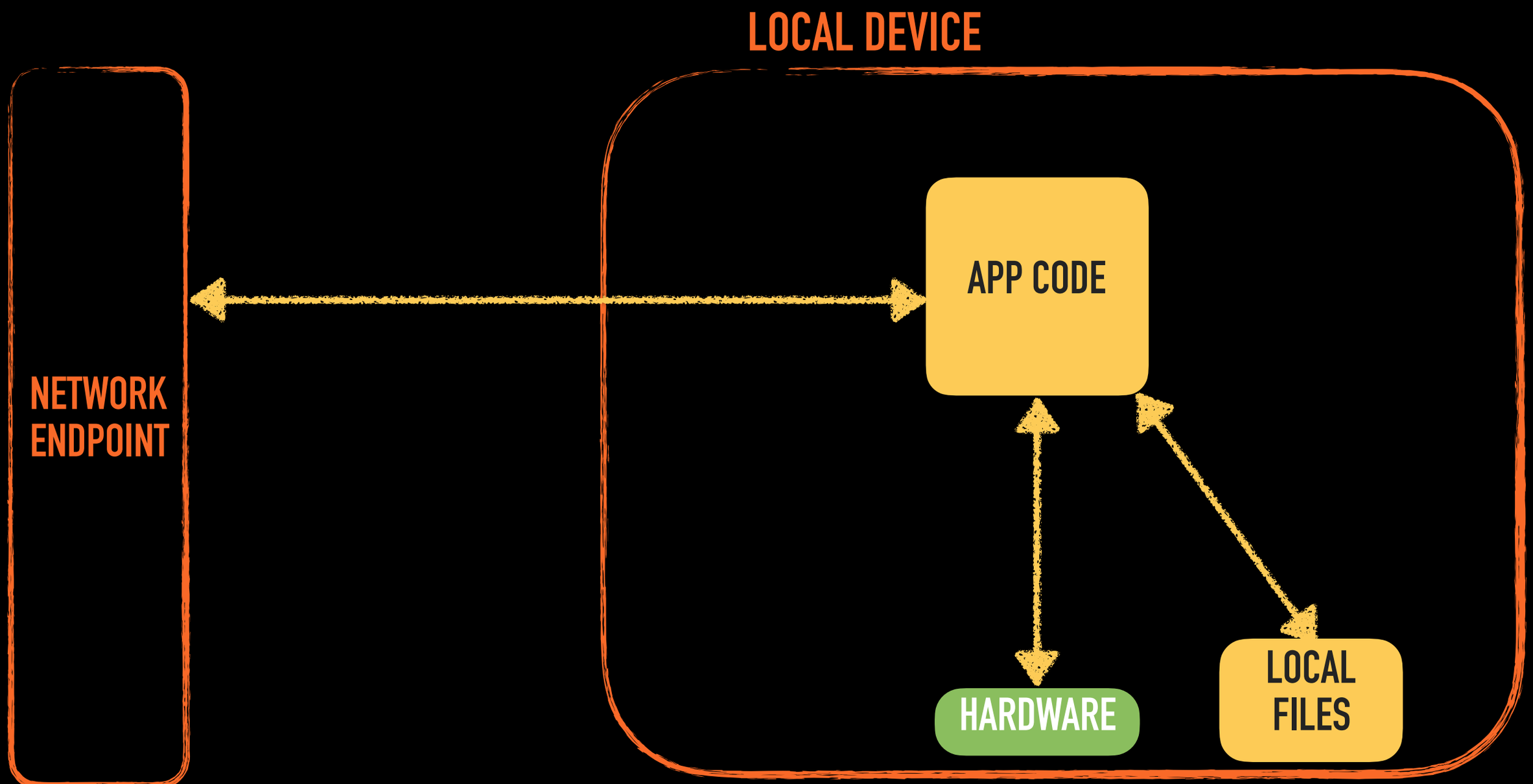
WRITING CHEATS IS SEXY.

WHOAMI

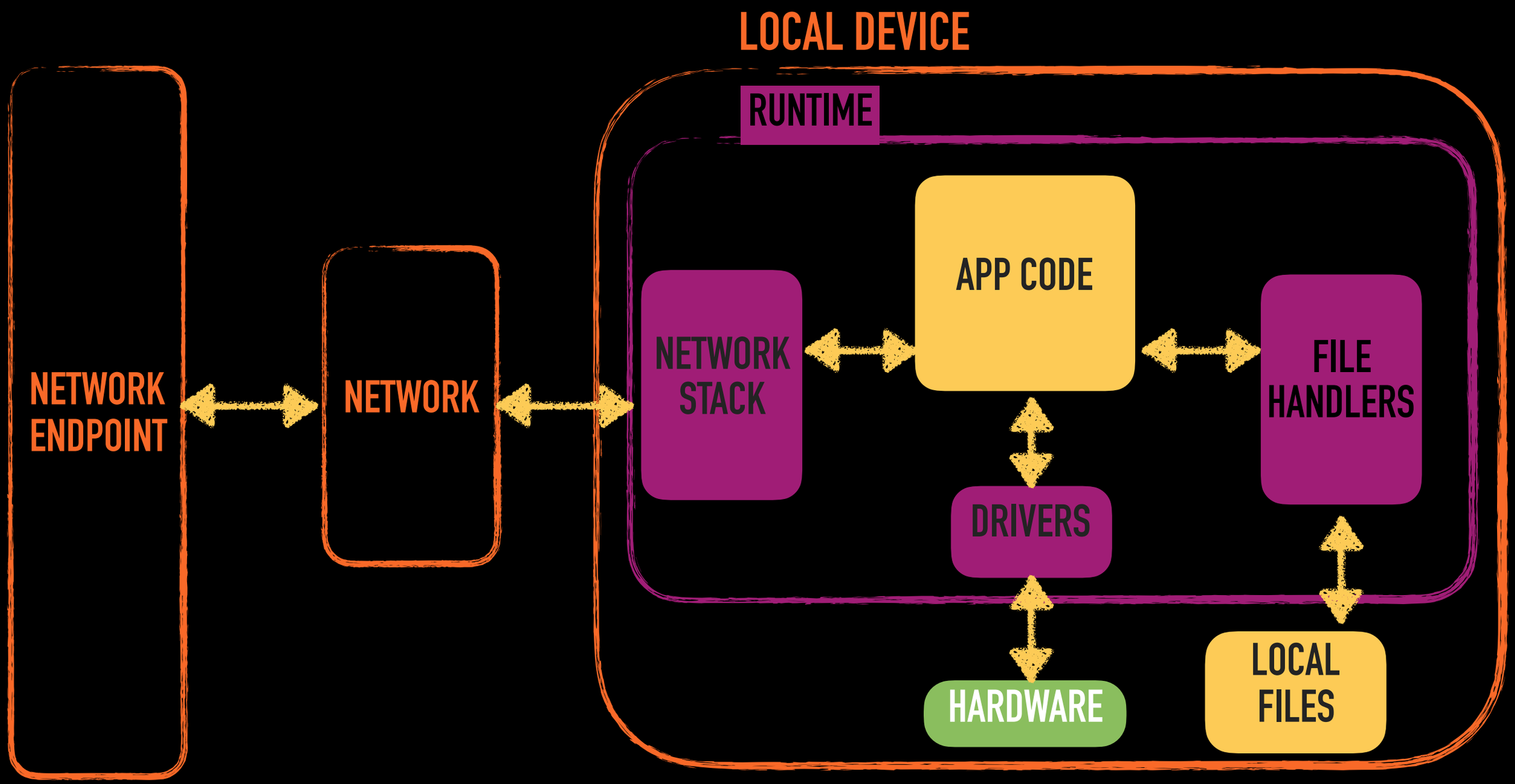
- Zsombor Kovacs CISSP, OSCP, OSWP, OSCE
- Researcher, testing pens for a decade
- Director of Research @MRG Effitas
- Founder of Hackersuli



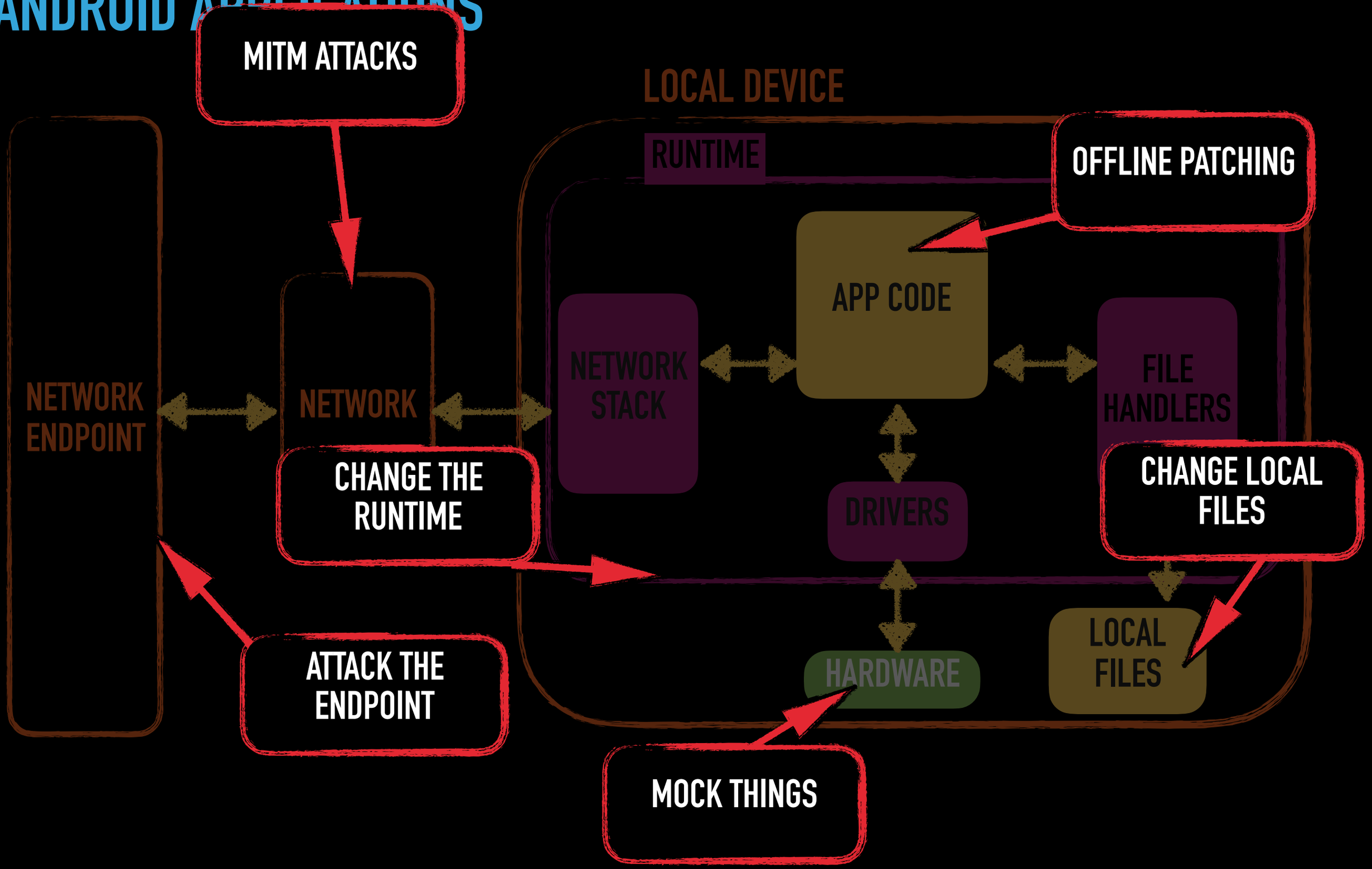
ANDROID APPLICATIONS



ANDROID APPLICATIONS



ANDROID APPLICATIONS



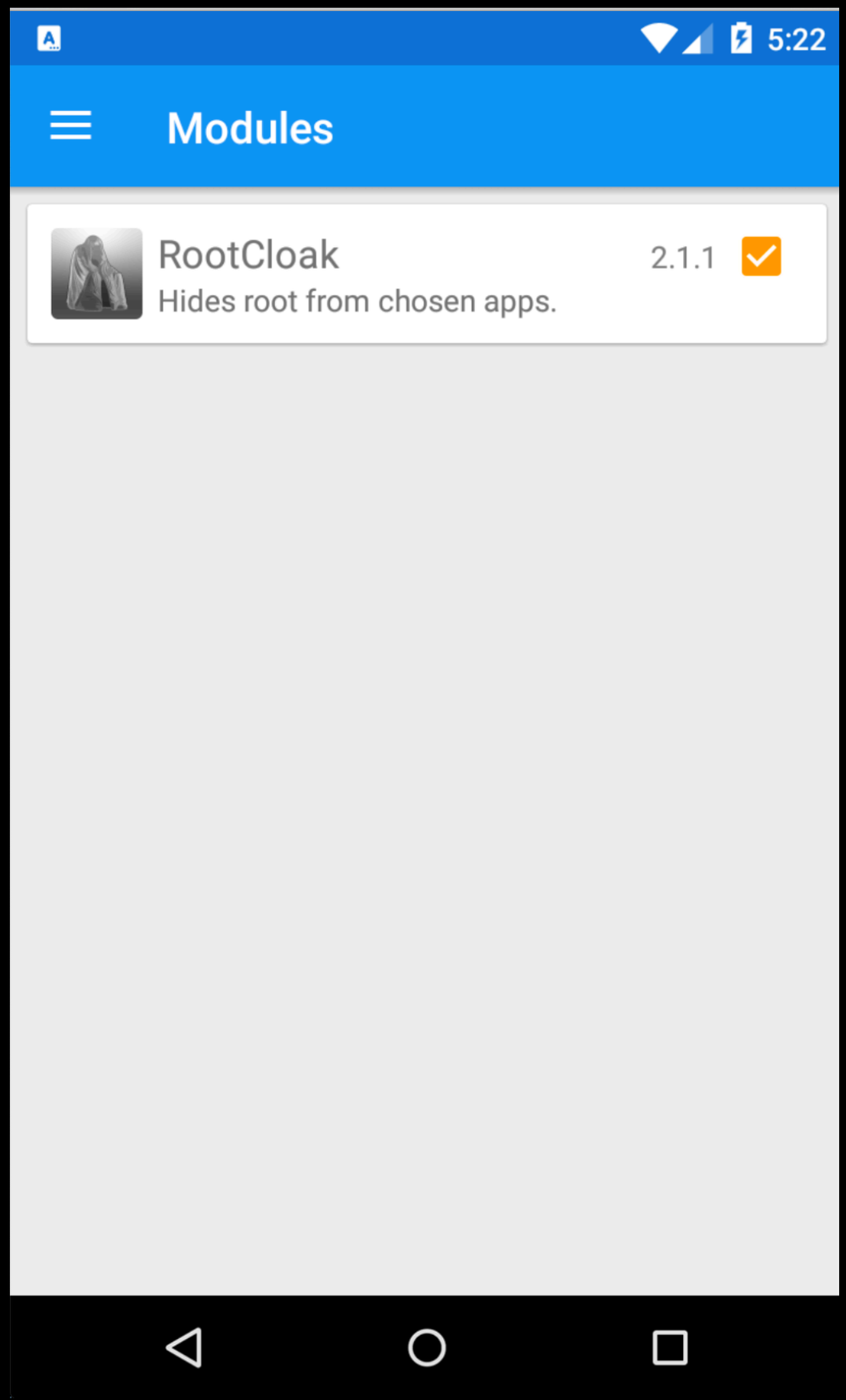
TAMPERING WITH THE RUNTIME

- Run-time manipulation of... *everything*
  - Intercept and change method call parameters
  - Overwrite methods
- ...and go home like nothing happened

TOOLS OF THE TRADE

XPOSED FRAMEWORK

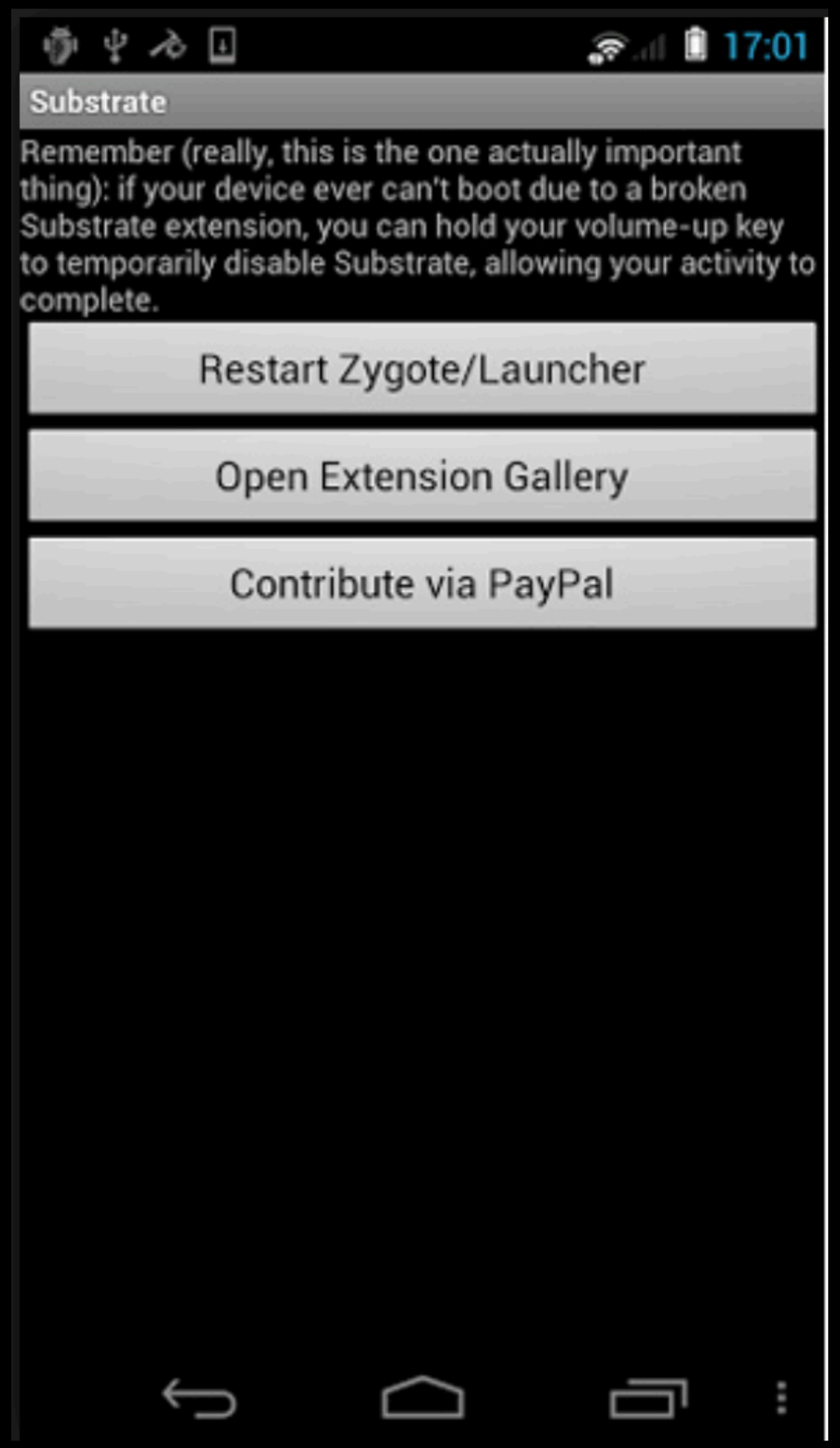
- From KitKat on
- Designed for permanent changes
- Lots of modules



TOOLS OF THE TRADE

CYDIA SUBSTRATE

- Lots of modules
- Supported only on KitKat :(



TOOLS OF THE TRADE

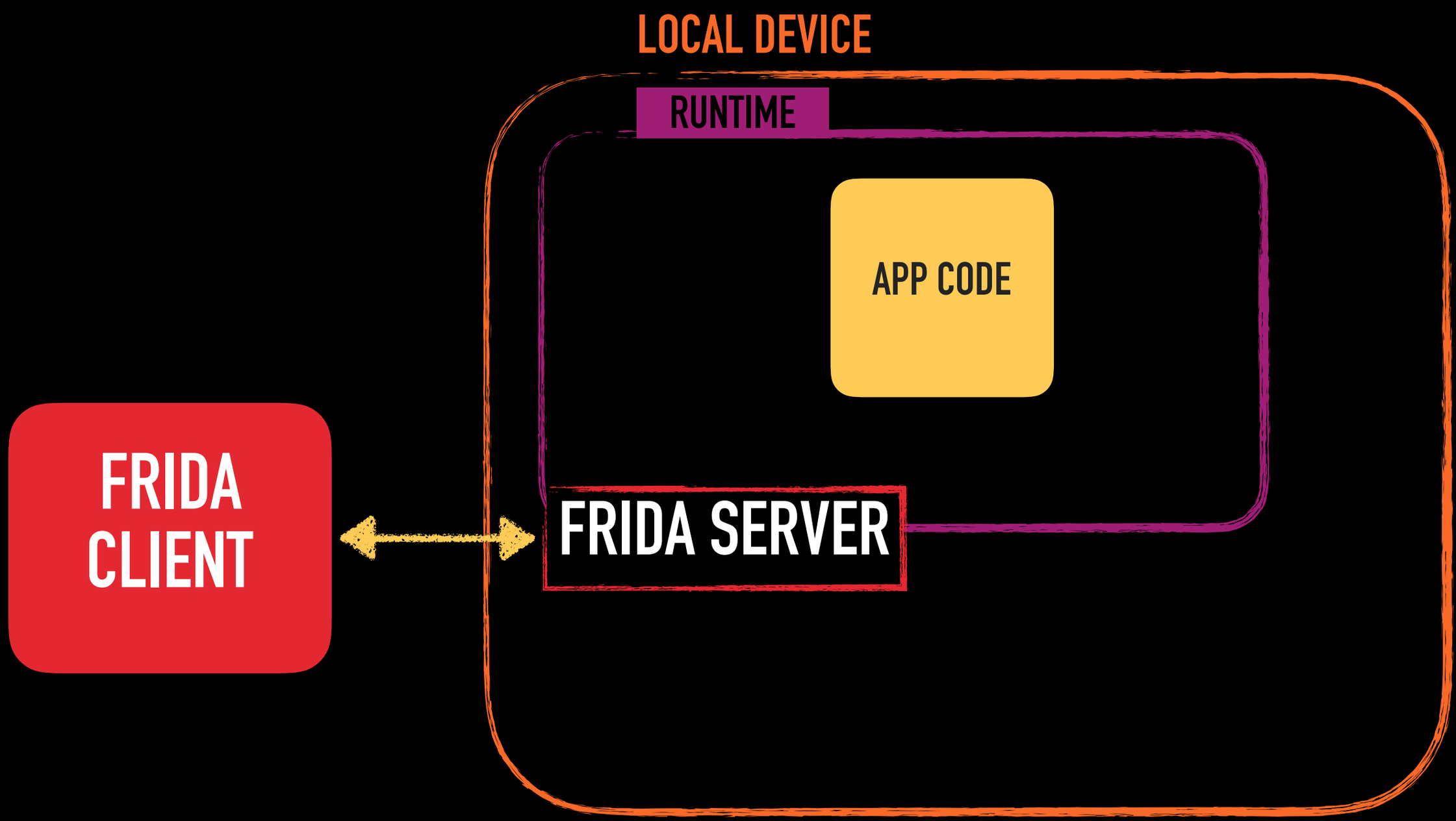
FRIDA

- Perfect for pentesters
- Scriptable in JavaScript
- Rock solid architecture

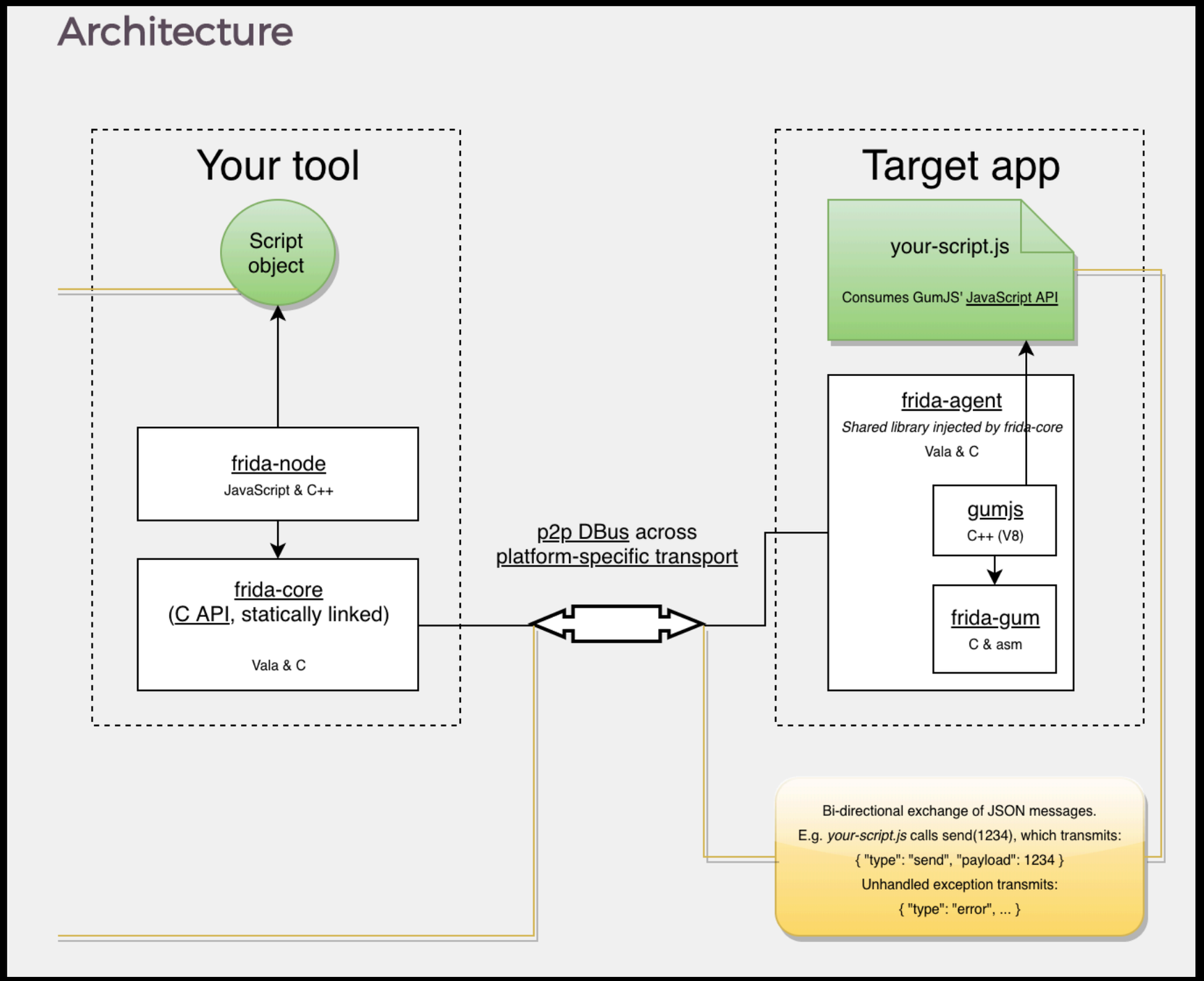
A TYPICAL WORKFLOW WITH FRIDA

- Decompile the application.
- Pinpoint the relevant bits in the code.
- Create a frida script to tweak the code while the app is being run.
- Run app and the script.
- Profit.

FRIDA INTERNALS



FRIDA INTERNALS



APPLICATION DECOMPILATION

- Gives an approximation of the actual code (no re-compilation)
- Tools of the trade
  - dex2jar (from .apk to .jar)
  - Your Favourite Java Decompiler (from .jar to actual code)
  - http://apkdecompilers.com

FRIDA CHEAT SHEET

TO TWEAK METHODS

```
1  package packagename;
2  import stuff;
3
4  public class MainActivity extends AppCompatActivity {
5      [...]
6      public static Boolean method(String param1, String param2) {
7          [stuff]
8      }
9  }
```

```
1  Java.perform(function () {
2      console.log("Start");
3      var Act = Java.use("packagename.MainActivity"); //full class name
4      Act.method.implementation = function(a,b){
5          [...stuff...]
6          return this.method(a,b); //to return the original value
7          return false; //to return something else
8      }
9  });
```

FRIDA CHEAT SHEET TO DUMP OBJECT DATA

```
1  package packagename;
2
3  import stuff;
4
5  public class MyObject implements Serializable
6  {
7    private transient String[] a; //the secret parameter
8    [...]
9  }
```

```
1  Java.perform(function () {
2      var Act = Java.choose("packagename.MyObject", {
3          onMatch: function (instance) {
4              console.log("Found instance: " + instance);
5              var x = instance.a.value;
6              console.log(x);
7          },
8
9          onComplete: function () {
10         }
11     });
12   });
```

FRIDA CHEAT SHEET TO DUMP OBJECT DATA

```
1   package packagename;
2
3   import stuff;
4
5   public class MyObject implements Serializable
6   {
7     private transient String[] a; //the secret parameter
8
9     public void a(){
10      [...]
11    }
12  }
```

NOTE THE _

```
1   Java.perform(function () {
2       var Act = Java.choose("packagename.MyObject", {
3               onMatch: function (instance) {
4                   console.log("Found instance: " + instance);
5                   var x = instance._a.value;
6                   console.log(x);
7               },
8
9               onComplete: function () {
10              }
11      });
12    });
```

DEMO – FIND THE KEY

DEMO – FIND THE KEY

```
public static String encrypt(String key, String value) {
    try {
        SecretKeySpec skeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
        cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
        byte[] encrypted = cipher.doFinal(value.getBytes());
        String my = new String(encrypted);
        return my;
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return null;
}
```

```
Java.perform(function () {
    console.log("Start");
    var Act = Java.use("com.example.zsombor.cryptodemo.MainActivity");
    Act.encrypt.implementation = function(a,b){
        console.log("Key:");
        console.log(a);
        console.log("Ciphertext:");
        console.log(b);
        return this.encrypt(a,b);
    }
});
```

DEMO – FIND THE KEY

DEMO – BRUTE FORCE LOCAL PIN SCREEN

DEMO – BRUTE FORCE LOCAL PIN SCREEN

```
public class MainActivity extends AppCompatActivity {
    [...]

    protected boolean isPinCorrect(String pin){
        [....]
    }

    protected void onCreate(Bundle savedInstanceState) {
        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                if(isPinCorrect(pinField.getText().toString())){
                    [show success screen]
                } else {
                    [show failure screen]
                }
            }
    }
}
```

```
Java.perform(function () {
    var Act = Java.choose("com.example.zsombor.pinapp.MainActivity", {
            onMatch: function (instance) {
                console.log("Found instance: " + instance);
                console.log("Starting brute force...")
                for (var i = 1000; i < 9999; i++) {
                    if(instance.isPinCorrect(i+"") == true){
                        console.log("Result of " + i + ":" + instance.isPinCorrect(i+""));
                    }
                }
            },
            onComplete: function () { }
    });
});
```

DEMO – BRUTE FORCE LOCAL PIN SCREEN

DEMO – BYPASS ROOT DETECTION

DEMO – BYPASS ROOT DETECTION

```
public class MainActivity extends AppCompatActivity {
    [...]
    protected void onCreate(Bundle savedInstanceState) {
        [...]
        checkButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                    if(isRooted()){[show rooted message]} else
                    {[show clean message]}
                }
            });
    }

    public boolean isRooted() {
        [...]
    }
```

```
Java.perform(function (){
    console.log("Start");
    var Act = Java.use("com.example.zsombor.rootdetectorapp.MainActivity");
    Act.isRooted.overload().implementation = function(){
        console.log("isRooted invoked");
        return false;
    }
});
```

DEMO – BYPASS ROOT DETECTION IN A MOBILE BANKING APP

DEMO – A CHEATER FOR A SIMPLE MINESWEEPER GAME

DEMO – A CHEATER FOR A BLACKJACK GAME

DEMO – A CHEATER FOR A BLACKJACK GAME

WolframAlpha computational intelligence

1.2Quintillion / 1Billion

Browse Examples Surprise Me

Input:

$$\frac{1\,200\,000\,000\,000\,000\,000}{1\,000\,000\,000}$$

Open code

Exact result:

1 200 000 000

Step-by-step solution

Try it!