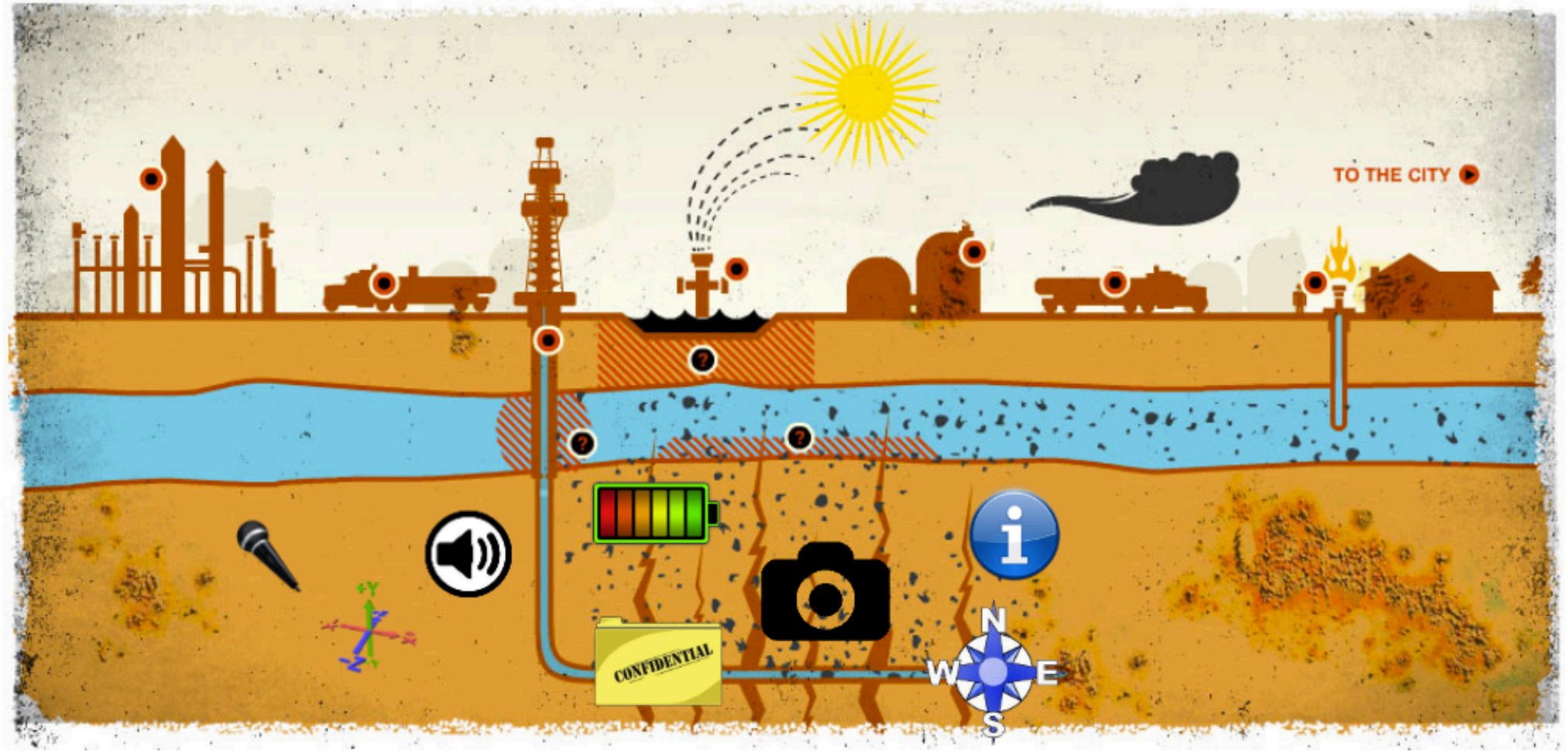
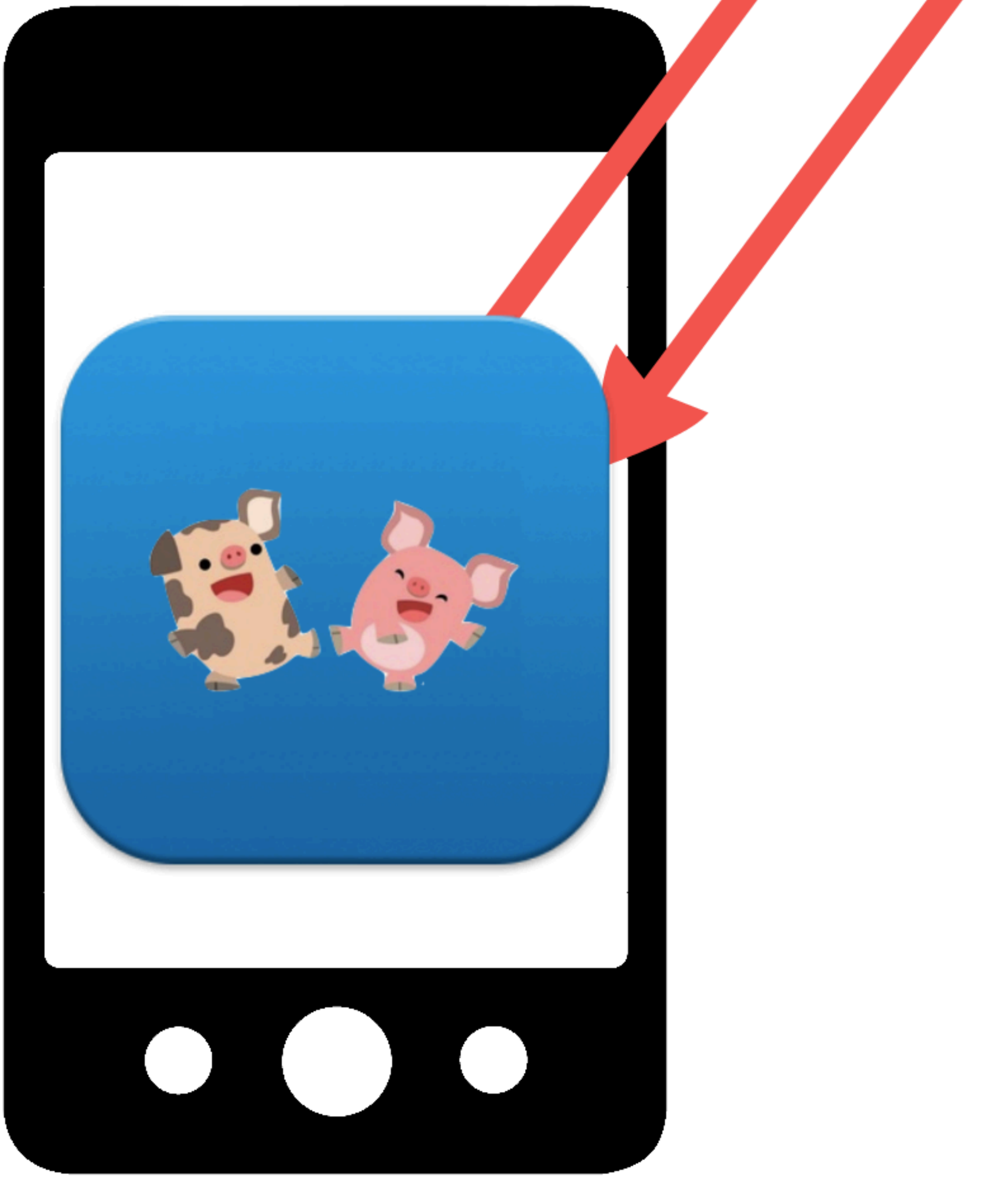
Breaking and Fixing Origin-Based Access Control in Hybrid Web/Mobile Application Frameworks

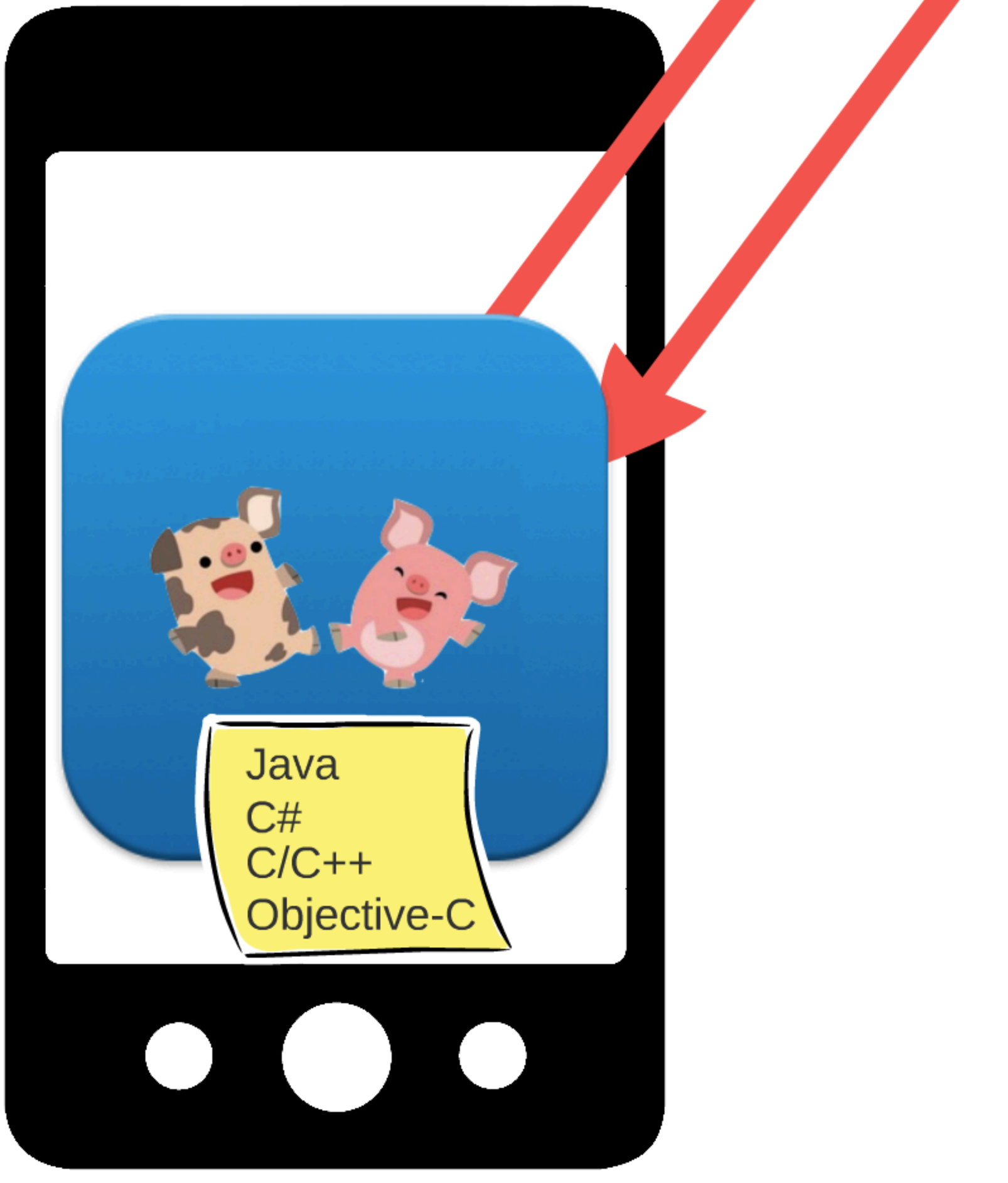
Martin Georgiev **Suman Jana** **Vitaly Shmatikov**

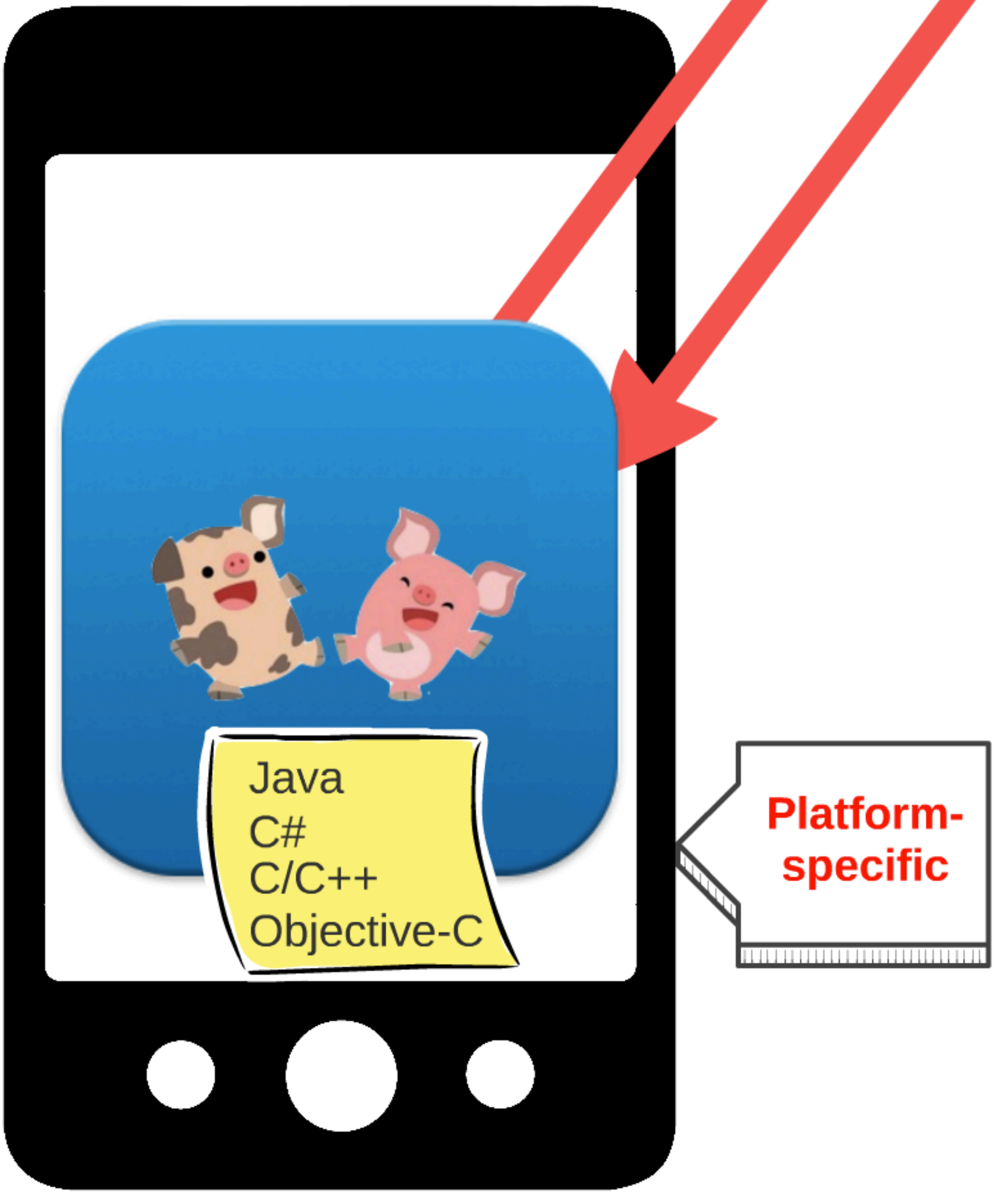
The University of Texas at Austin

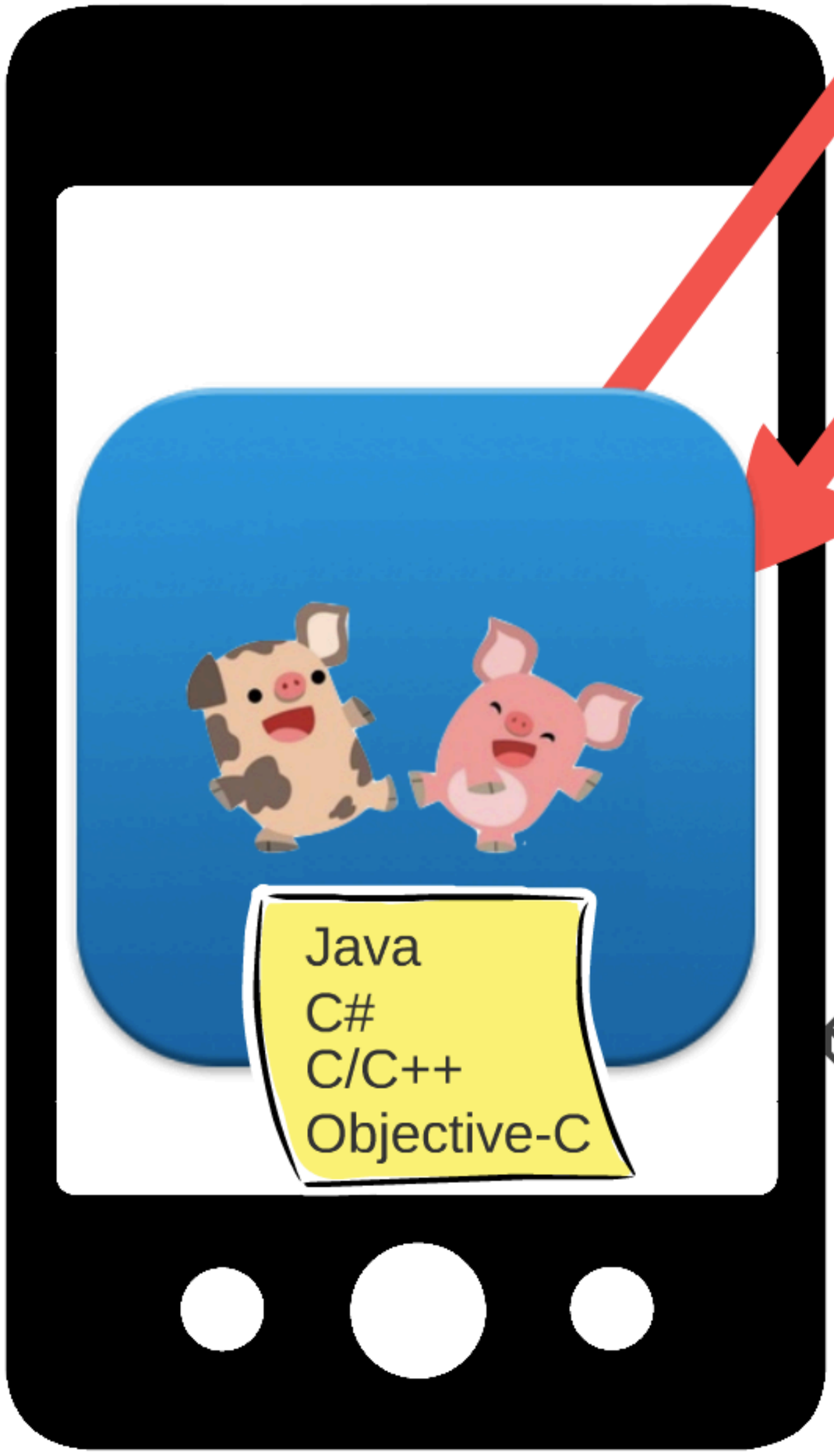


Once Upon A Time ...



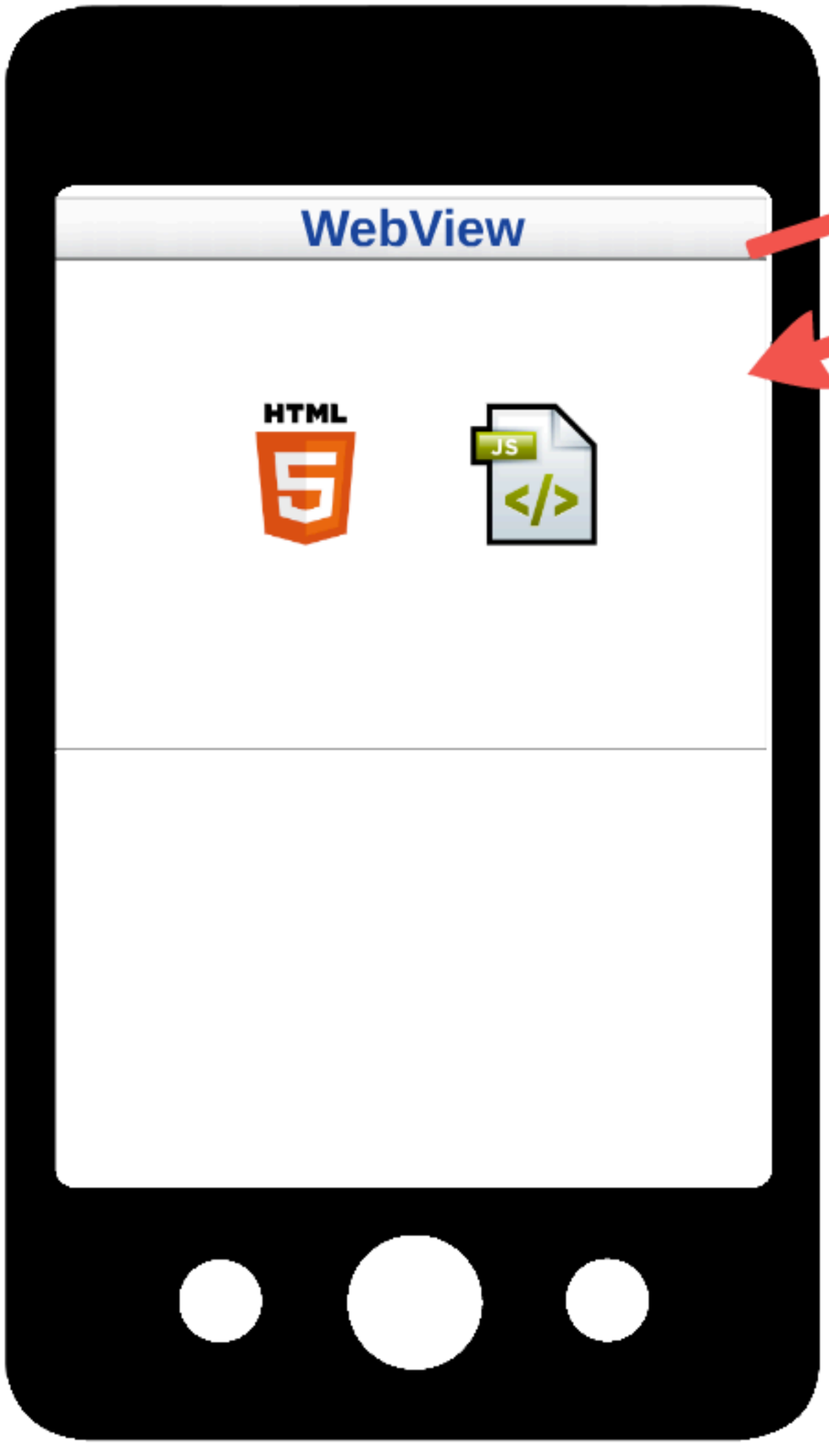

Java
C#
C/C++
Objective-C


Java
C#
C/C++
Objective-C
Platform-specific

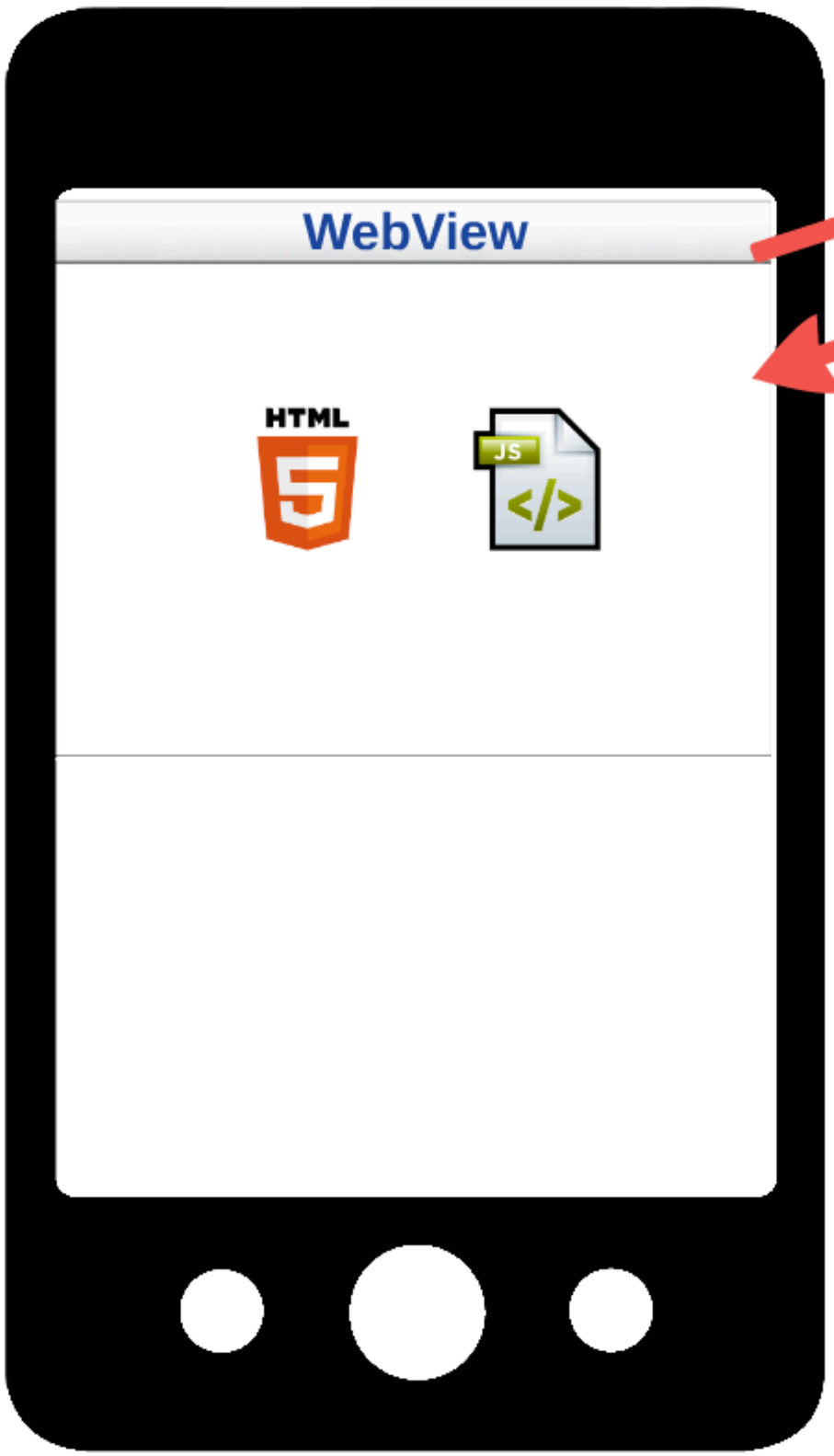


Platform-specific

Woulda been nice if...

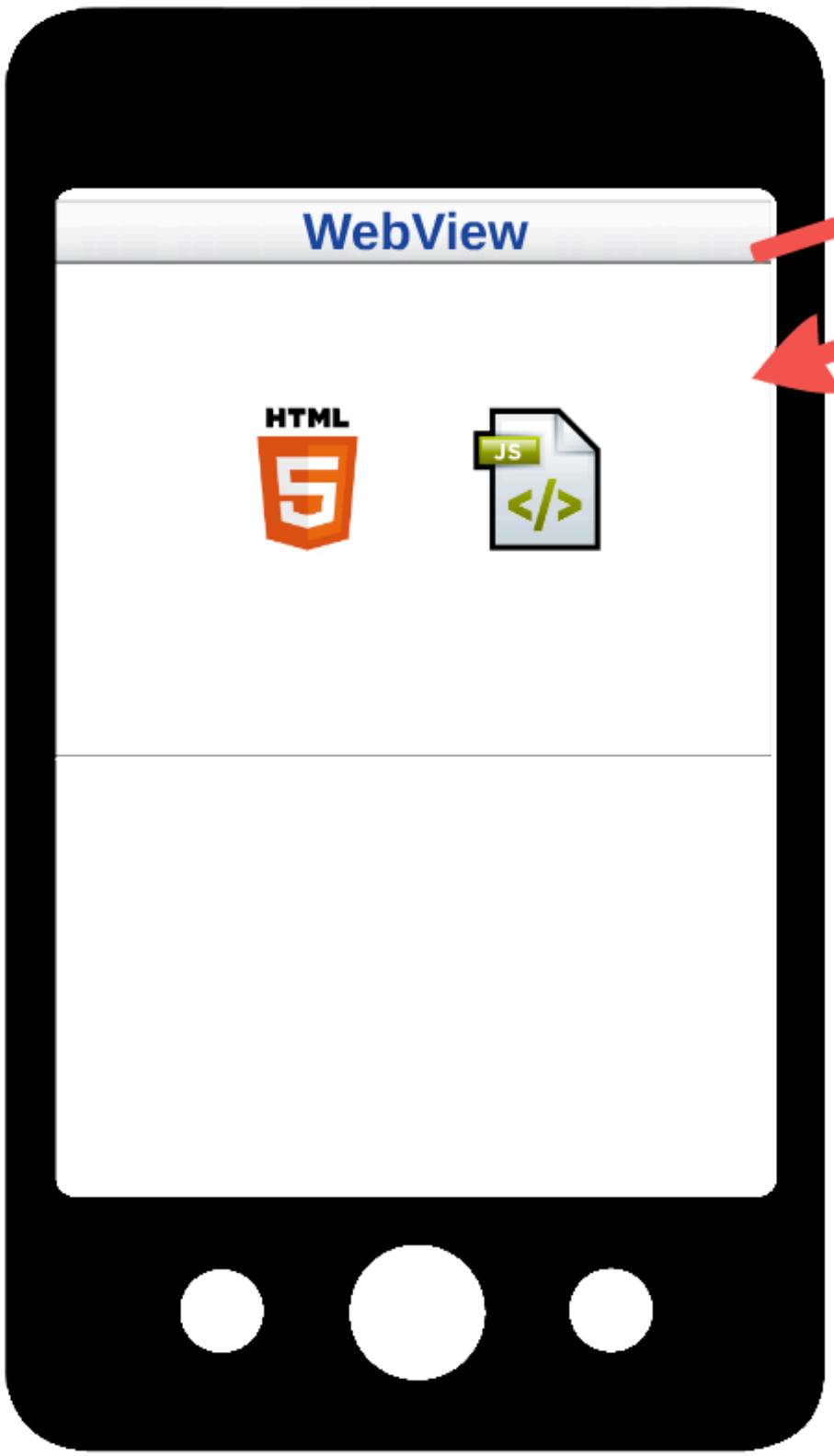


Woulda been nice if...



- Platform-independent
- Portable
- Reuse old Web apps
- Low maintenance

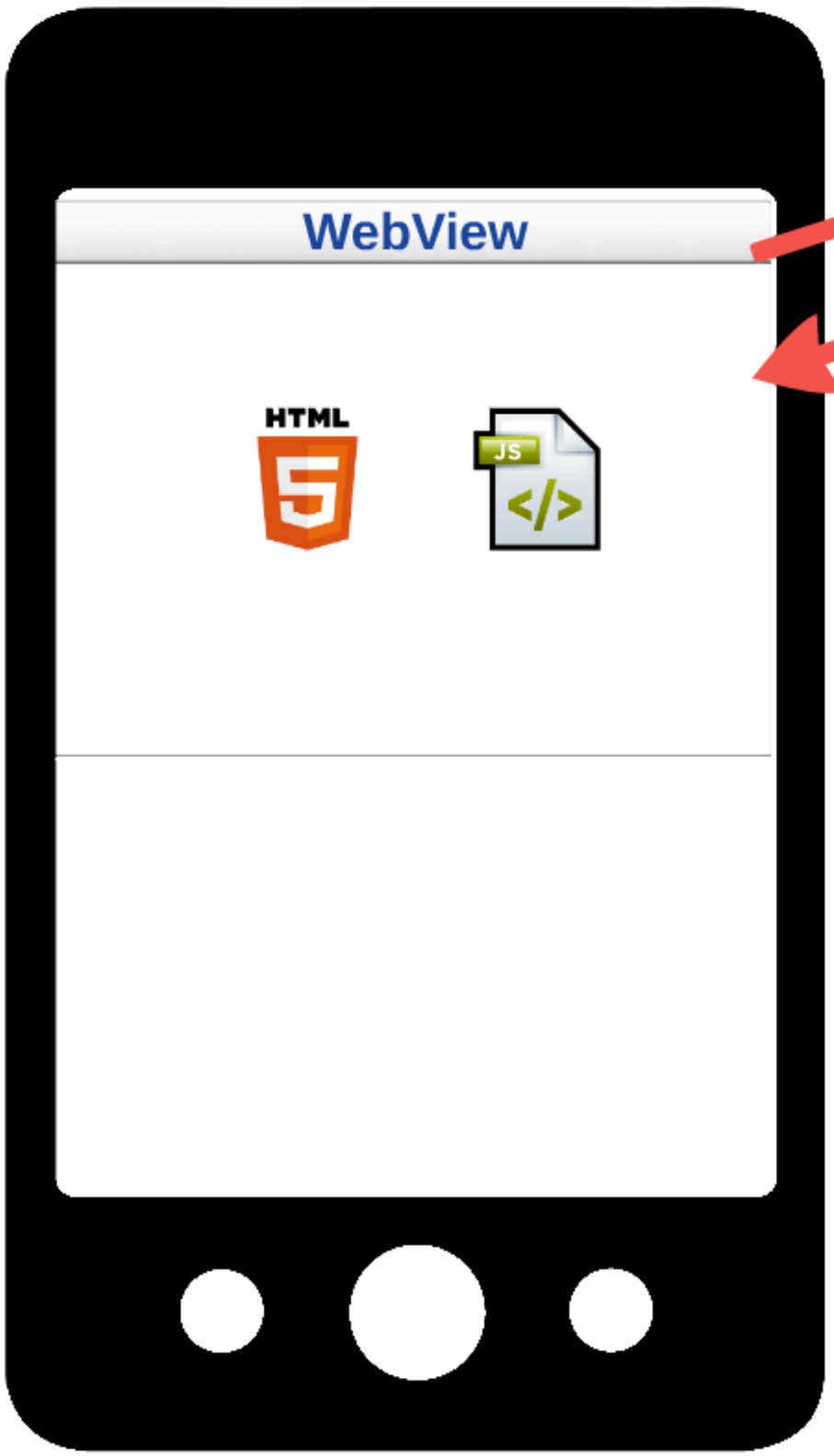
Woulda been nice if...



- Platform-independent
- Portable
- Reuse old Web apps
- Low maintenance

Hybrid App Model

Woulda been nice if...

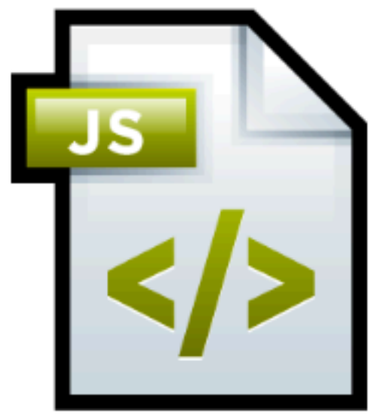


- Platform-independent
- Portable
- Reuse old Web apps
- Low maintenance

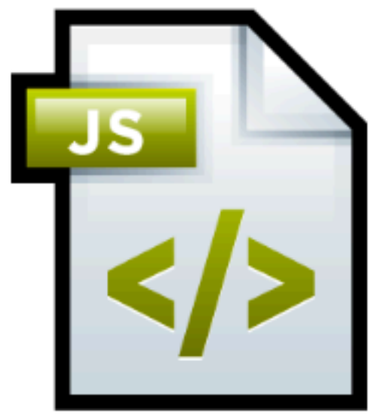
Over 400,000 developers worldwide

Hybrid App Model

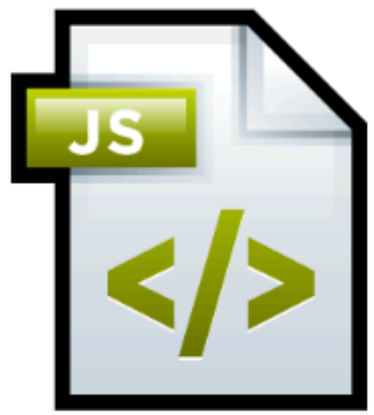
WebView

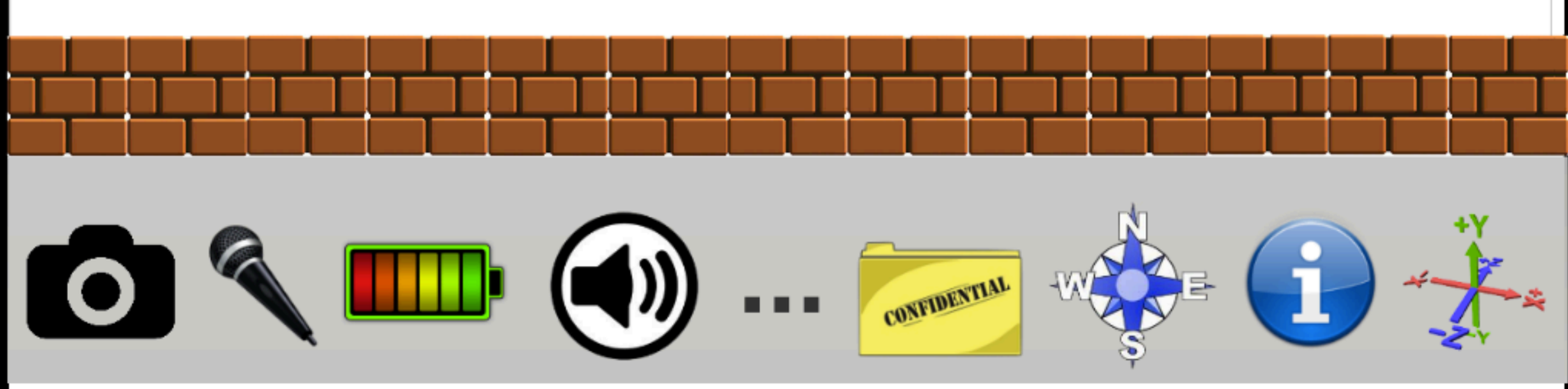


WebView



WebView



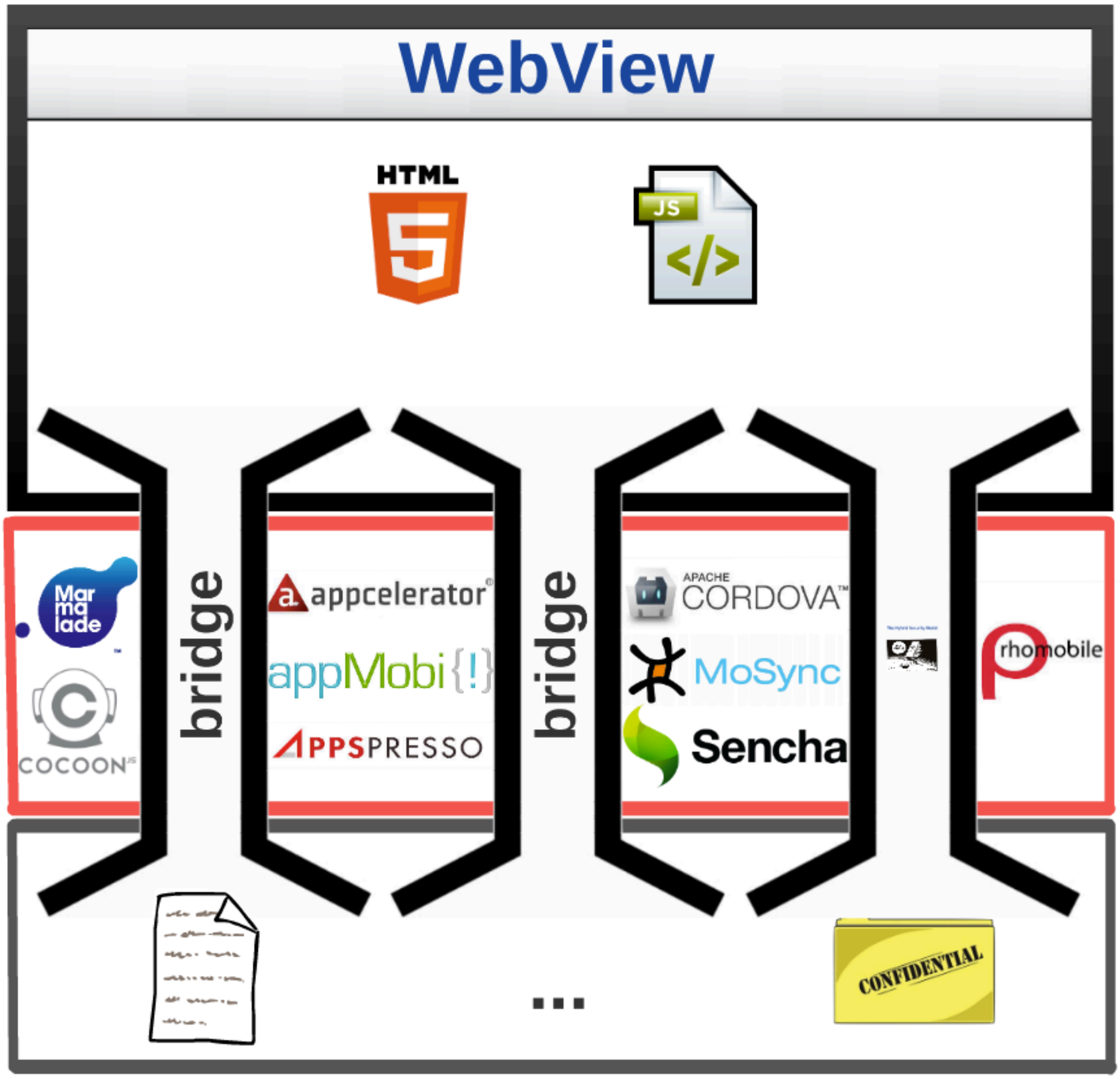


WebView

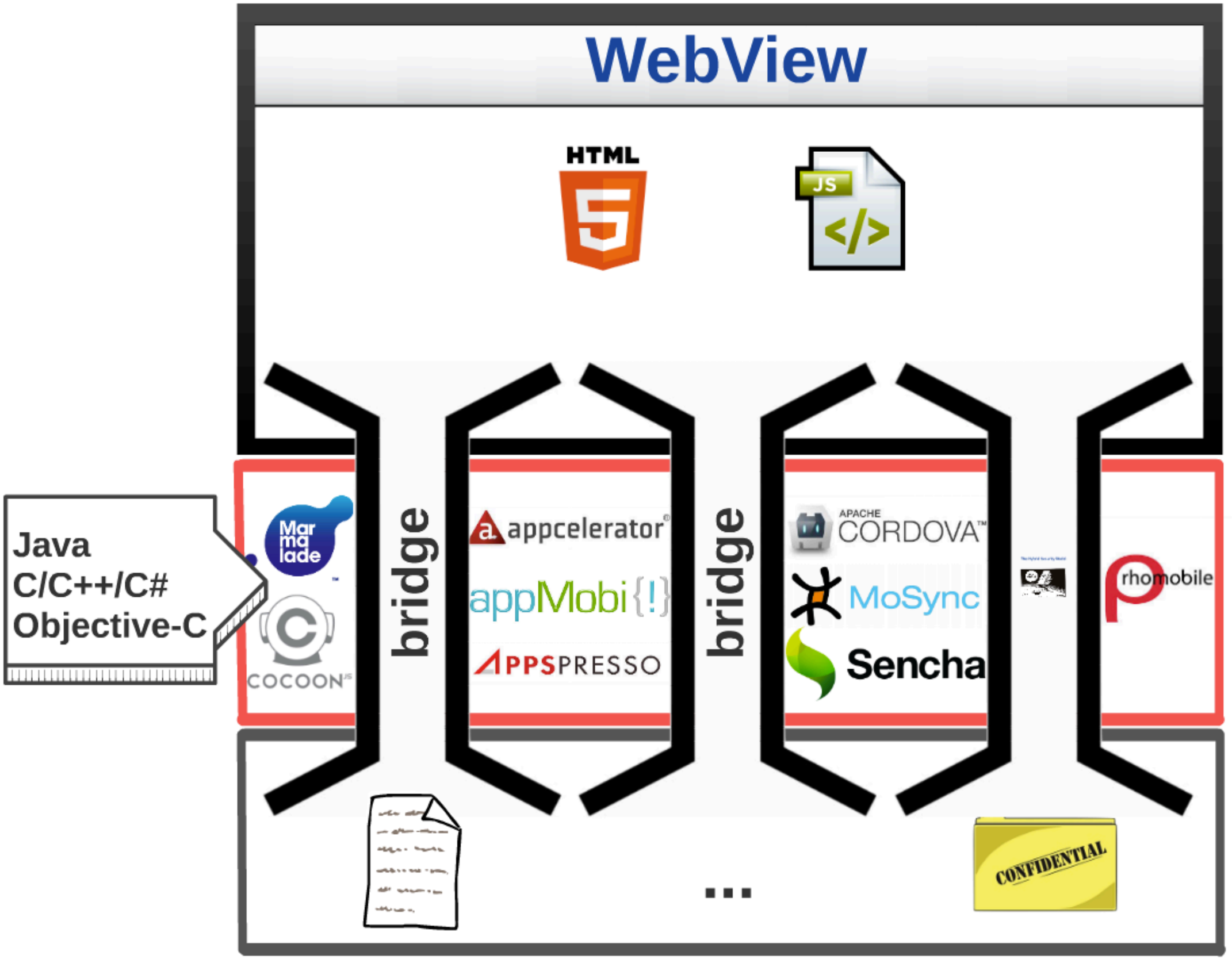
Problem:

How to equip Web apps with access to native resources?

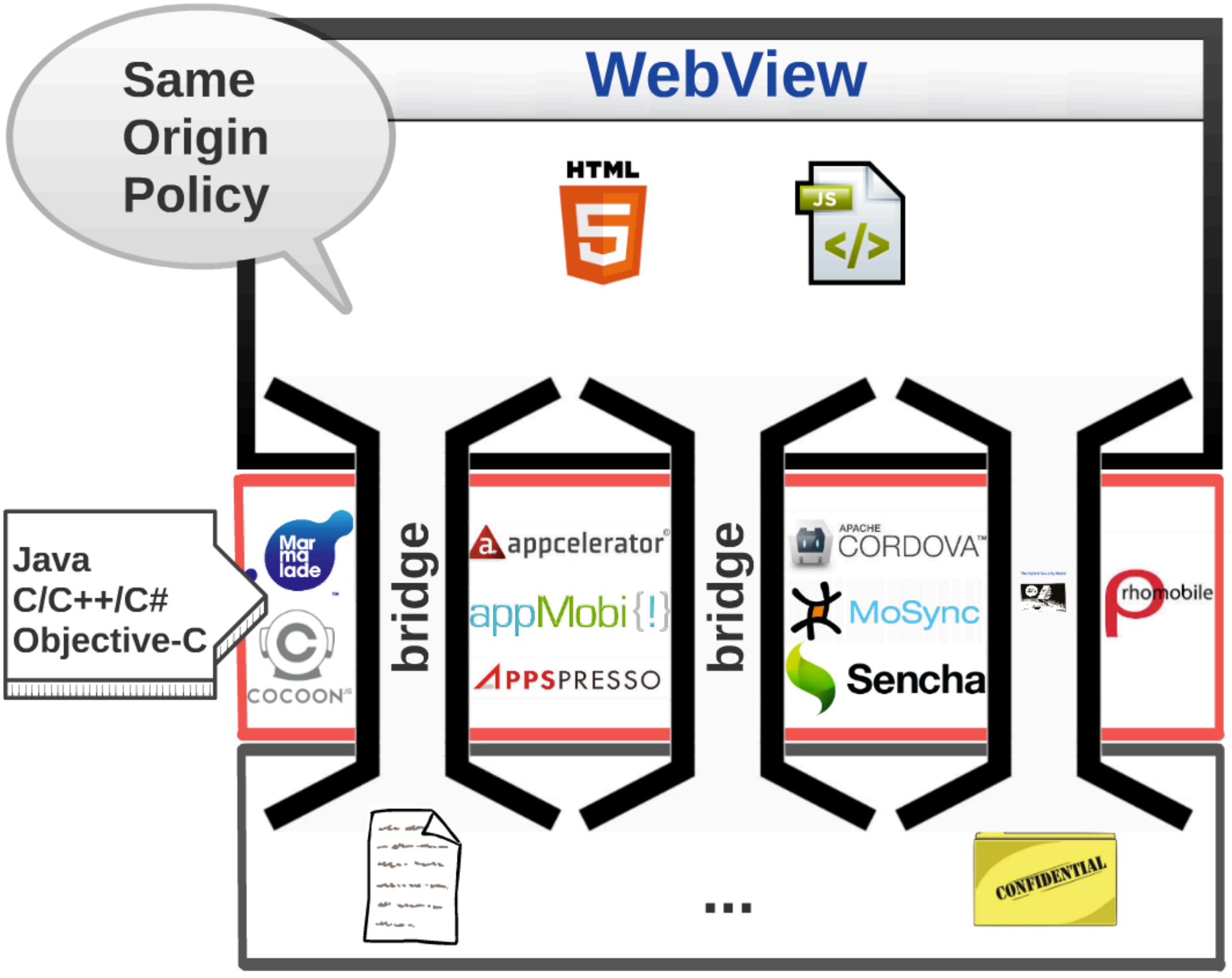
The World Of Hybrid Frameworks



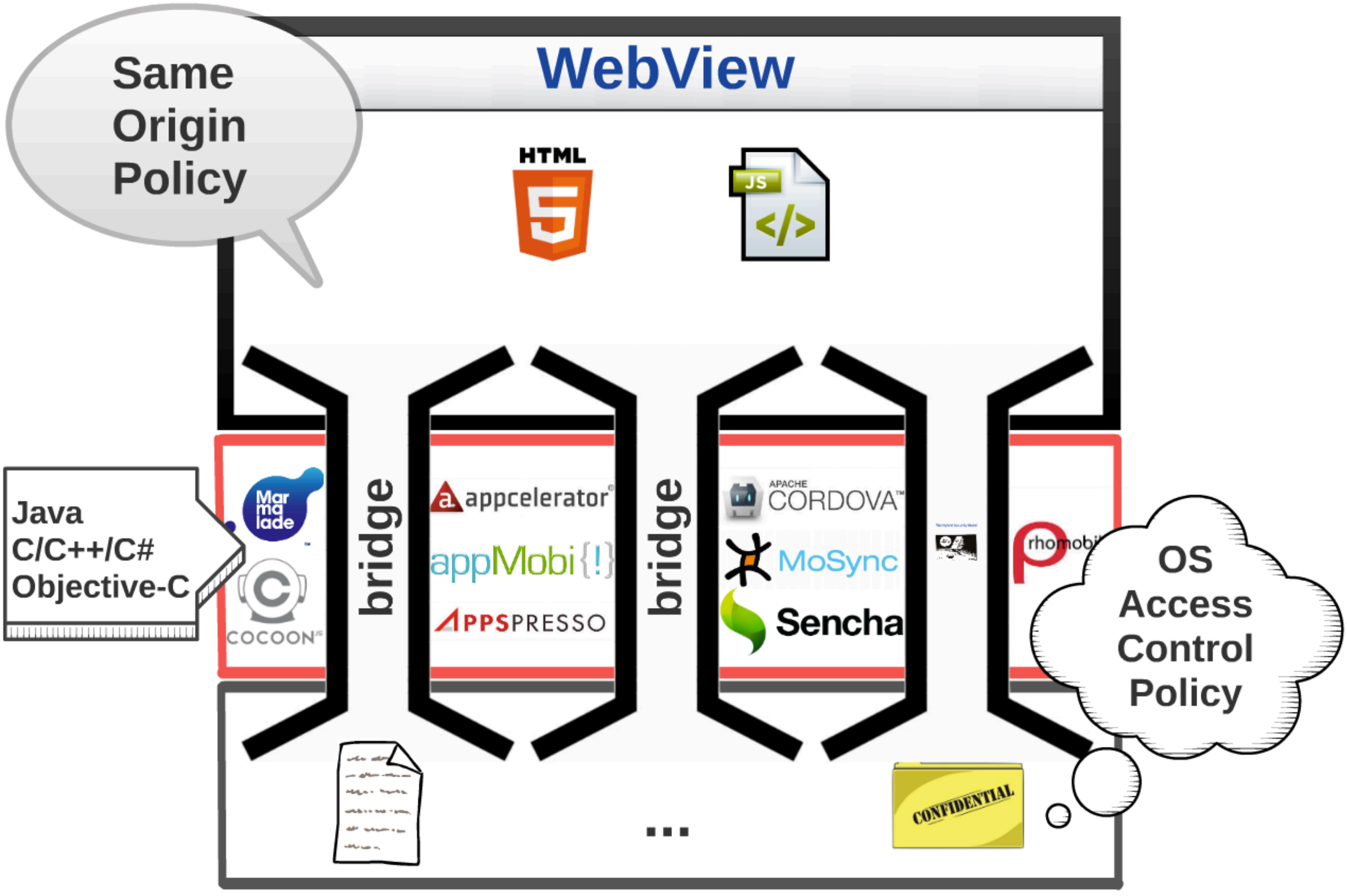
The World Of Hybrid Frameworks



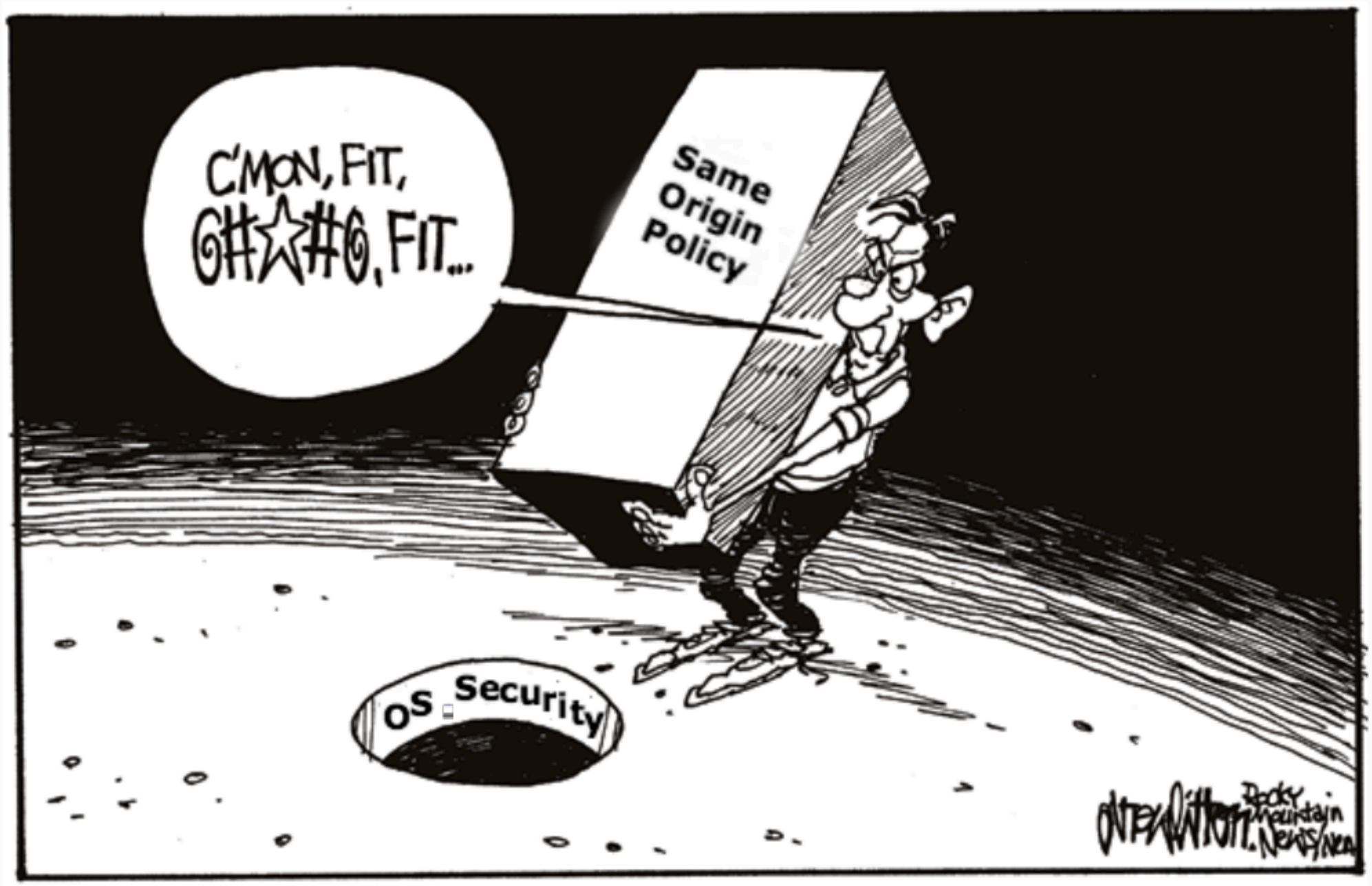
The World Of Hybrid Frameworks



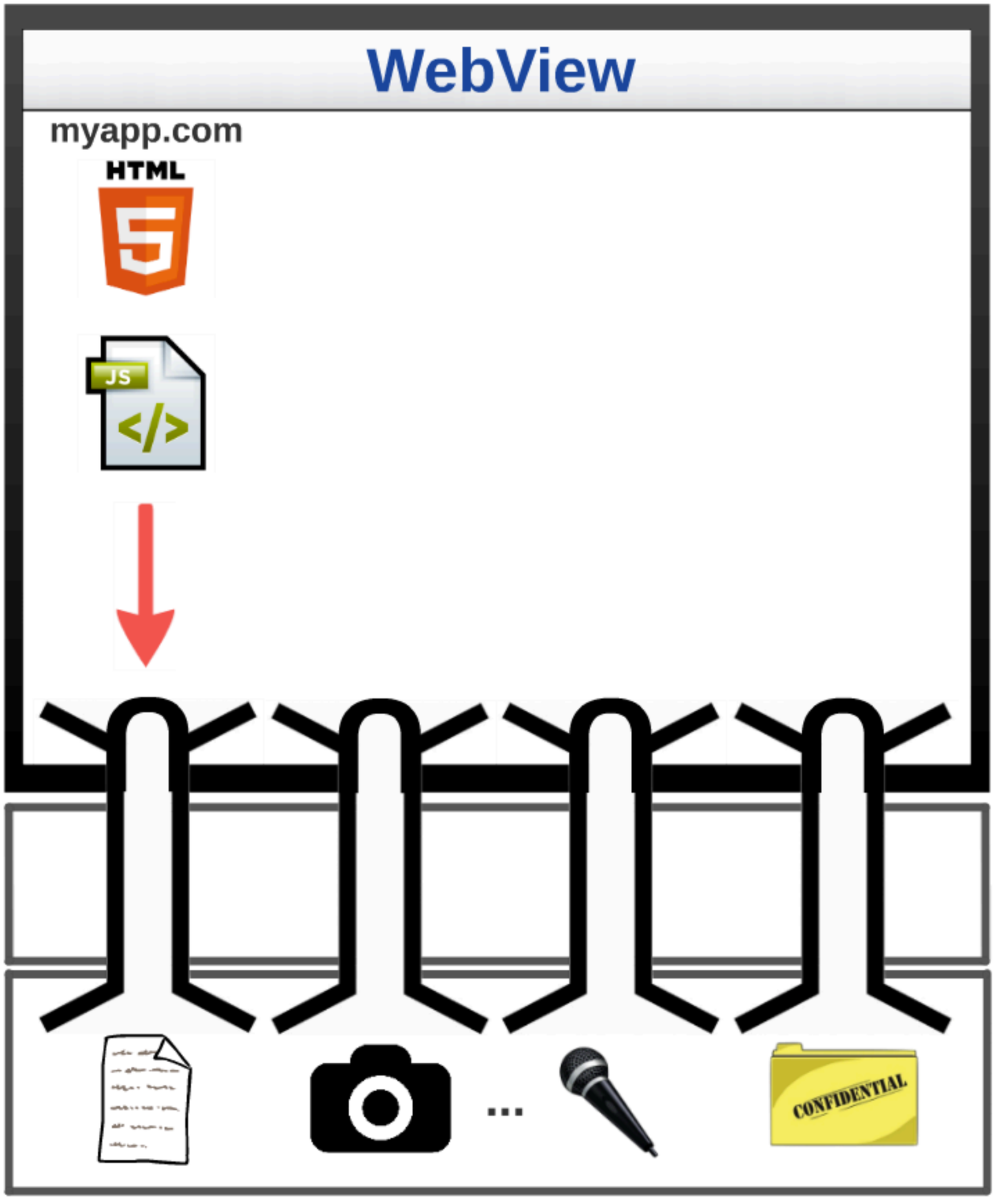
The World Of Hybrid Frameworks



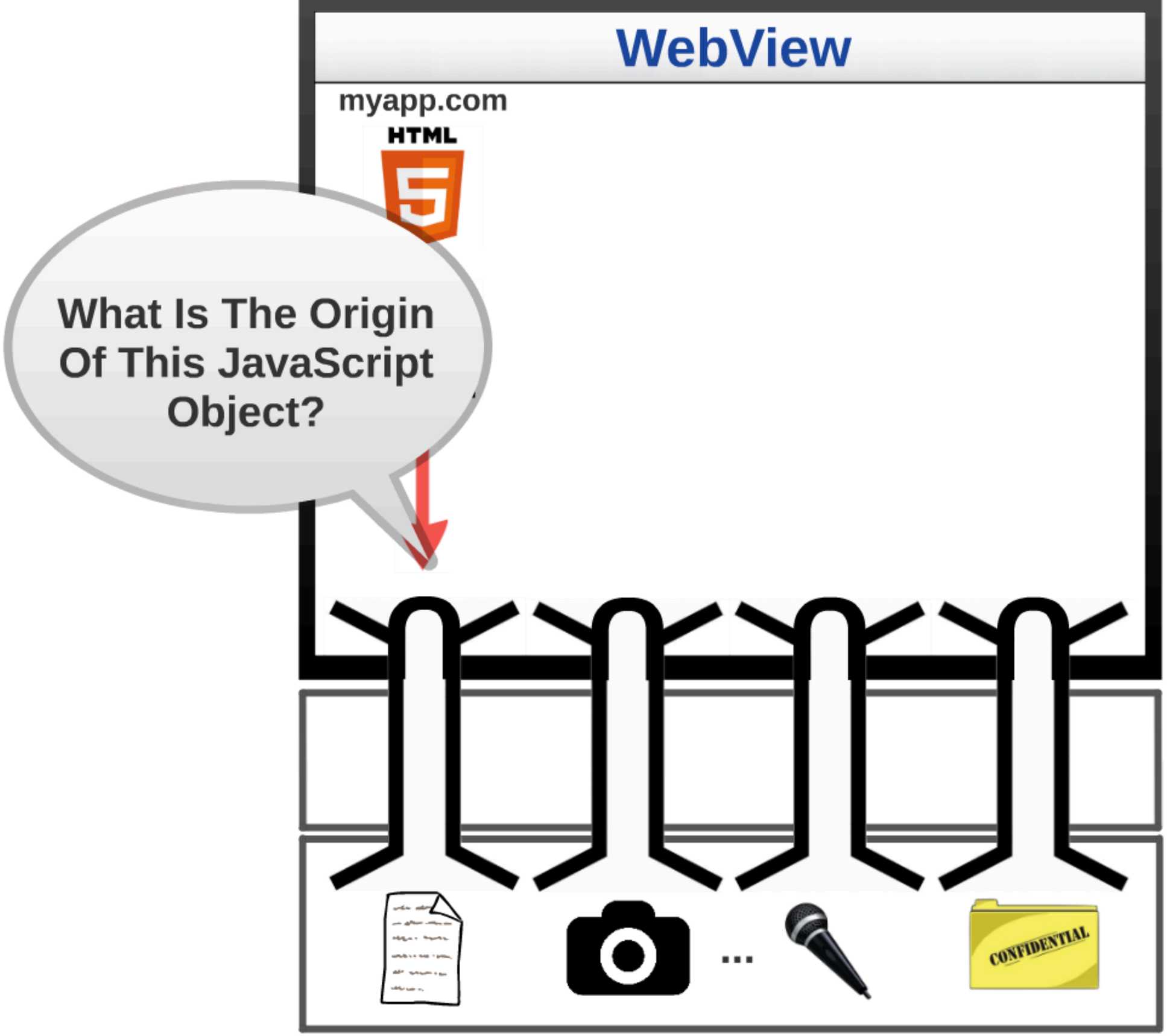
The Hybrid Security Model



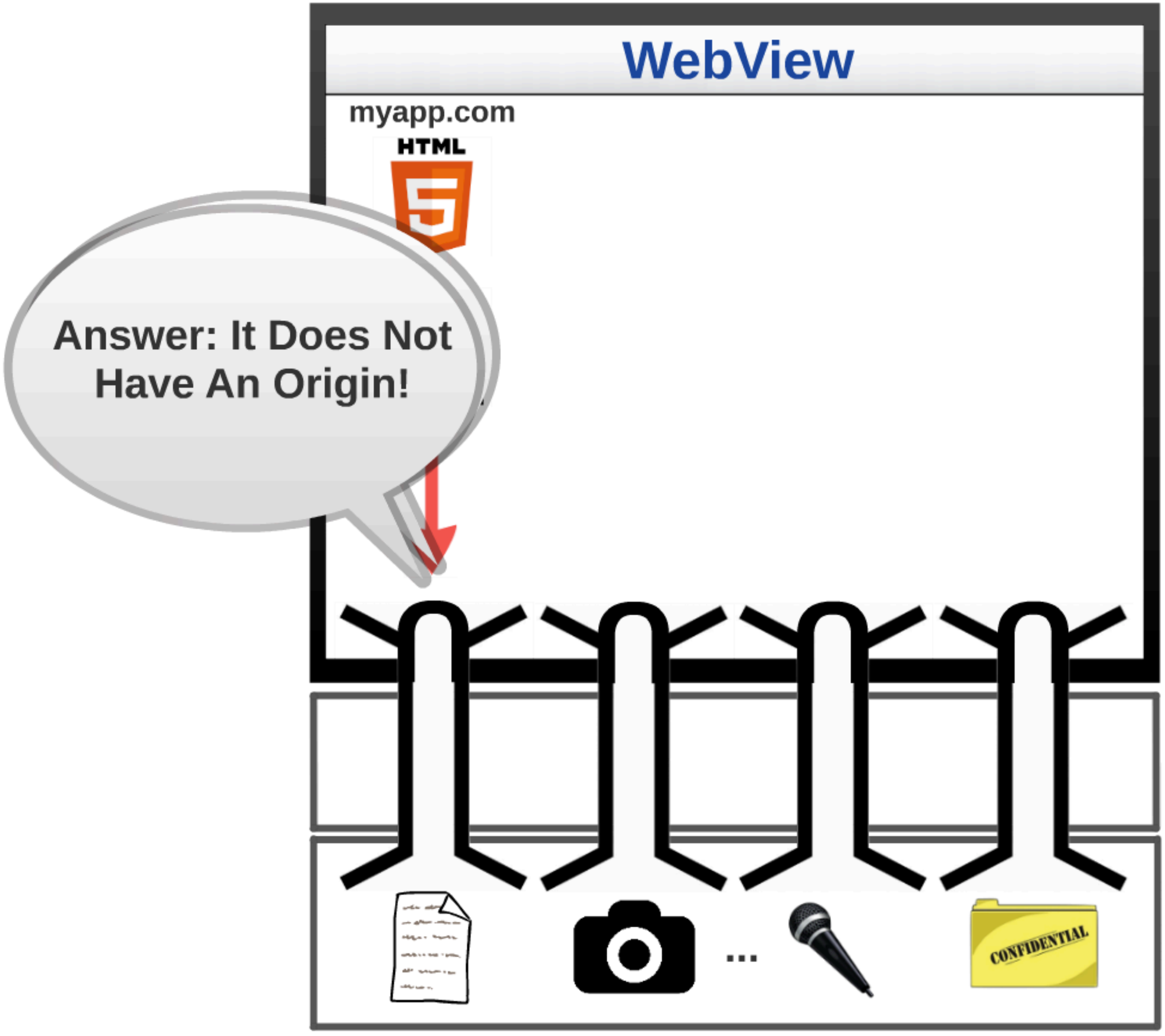
Attack Model



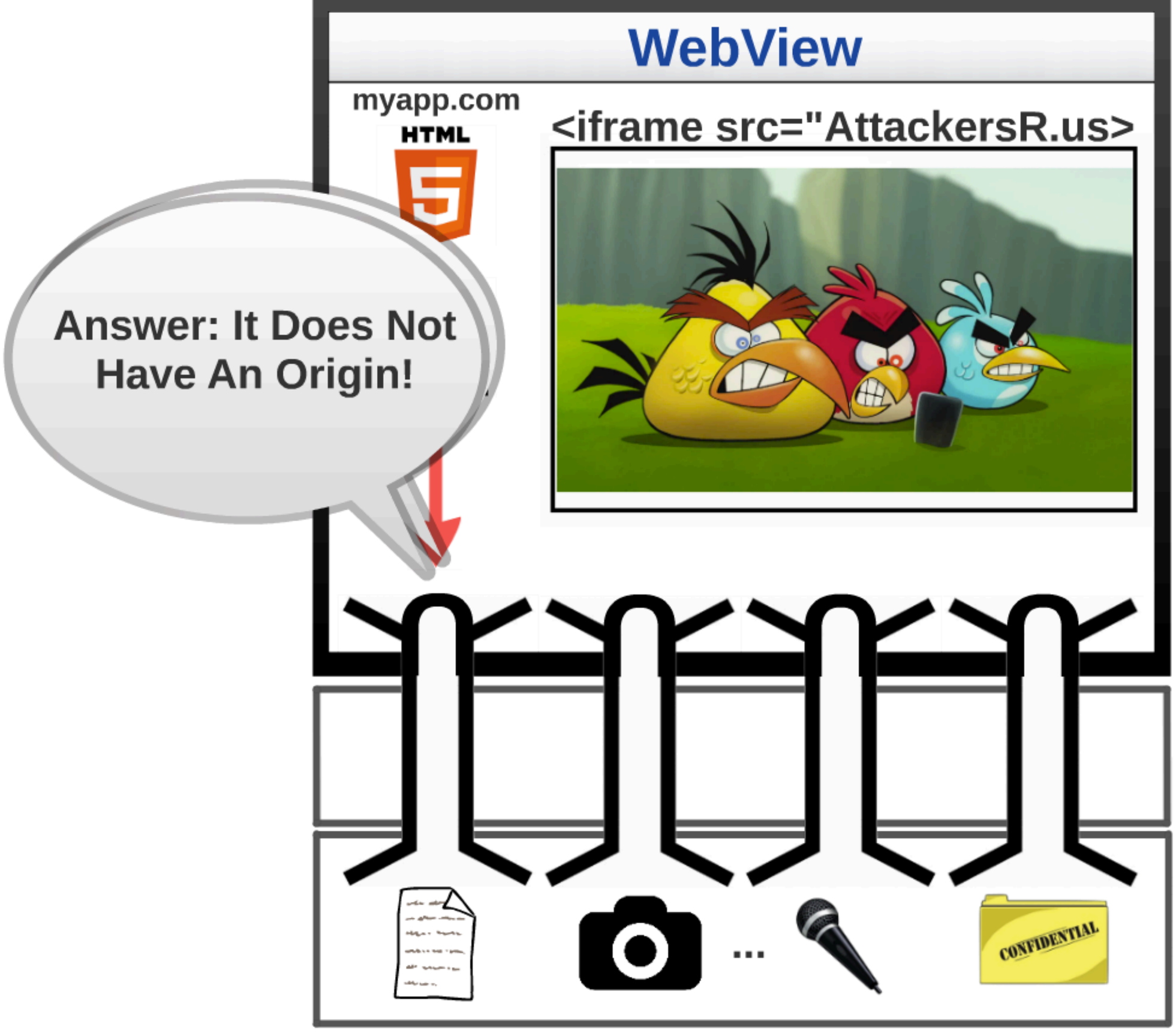
Attack Model



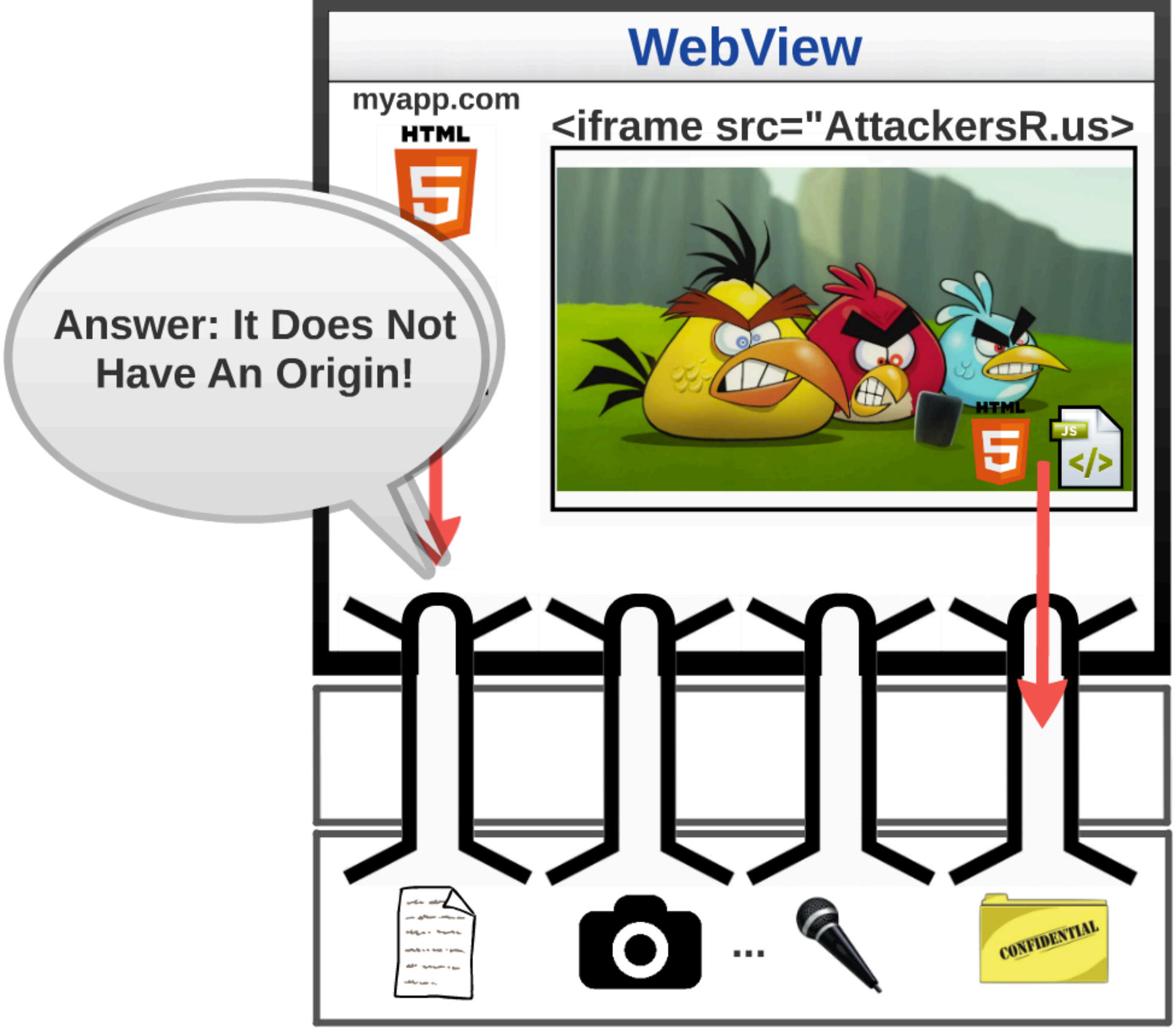
Attack Model



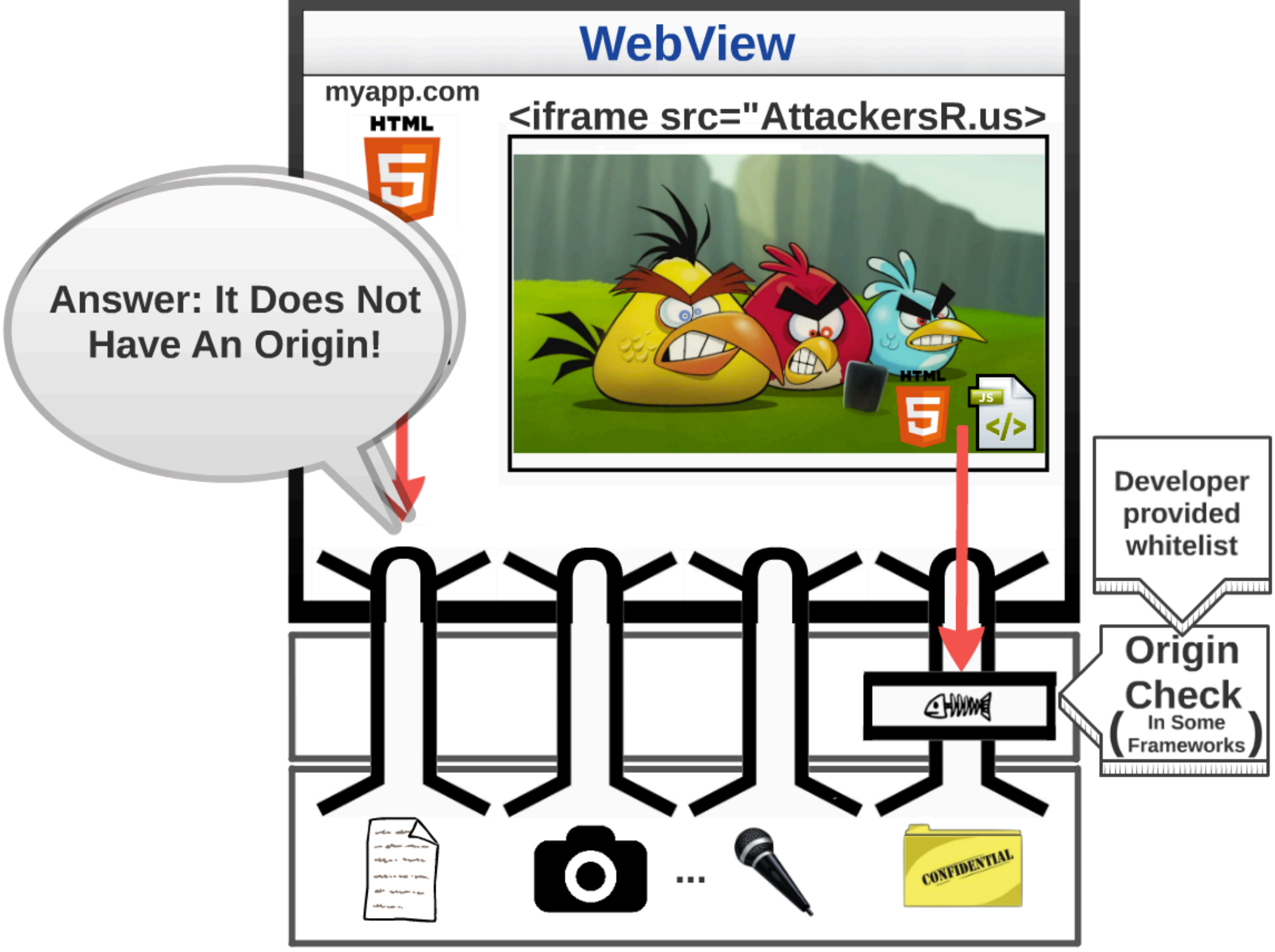
Attack Model



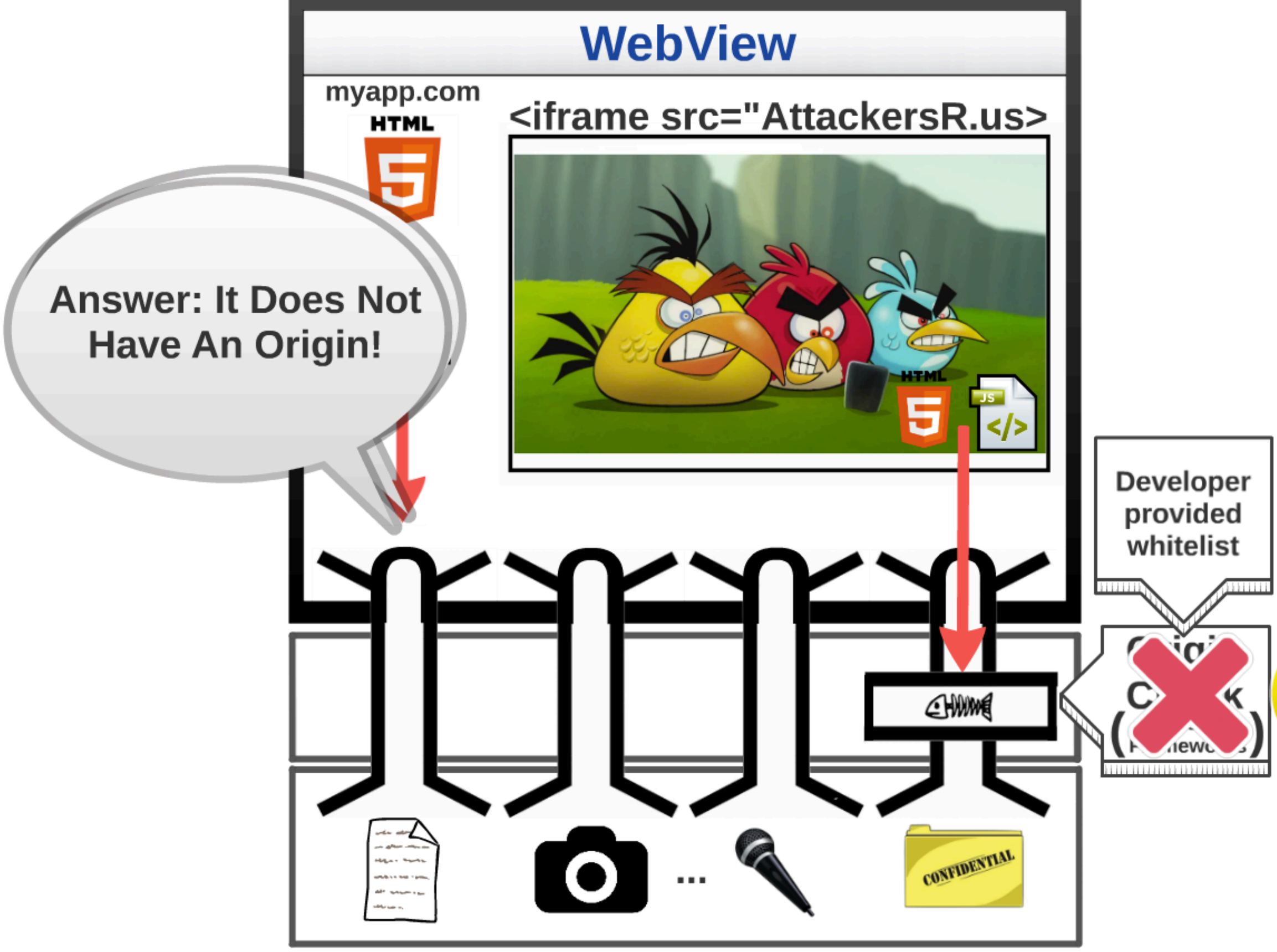
Attack Model



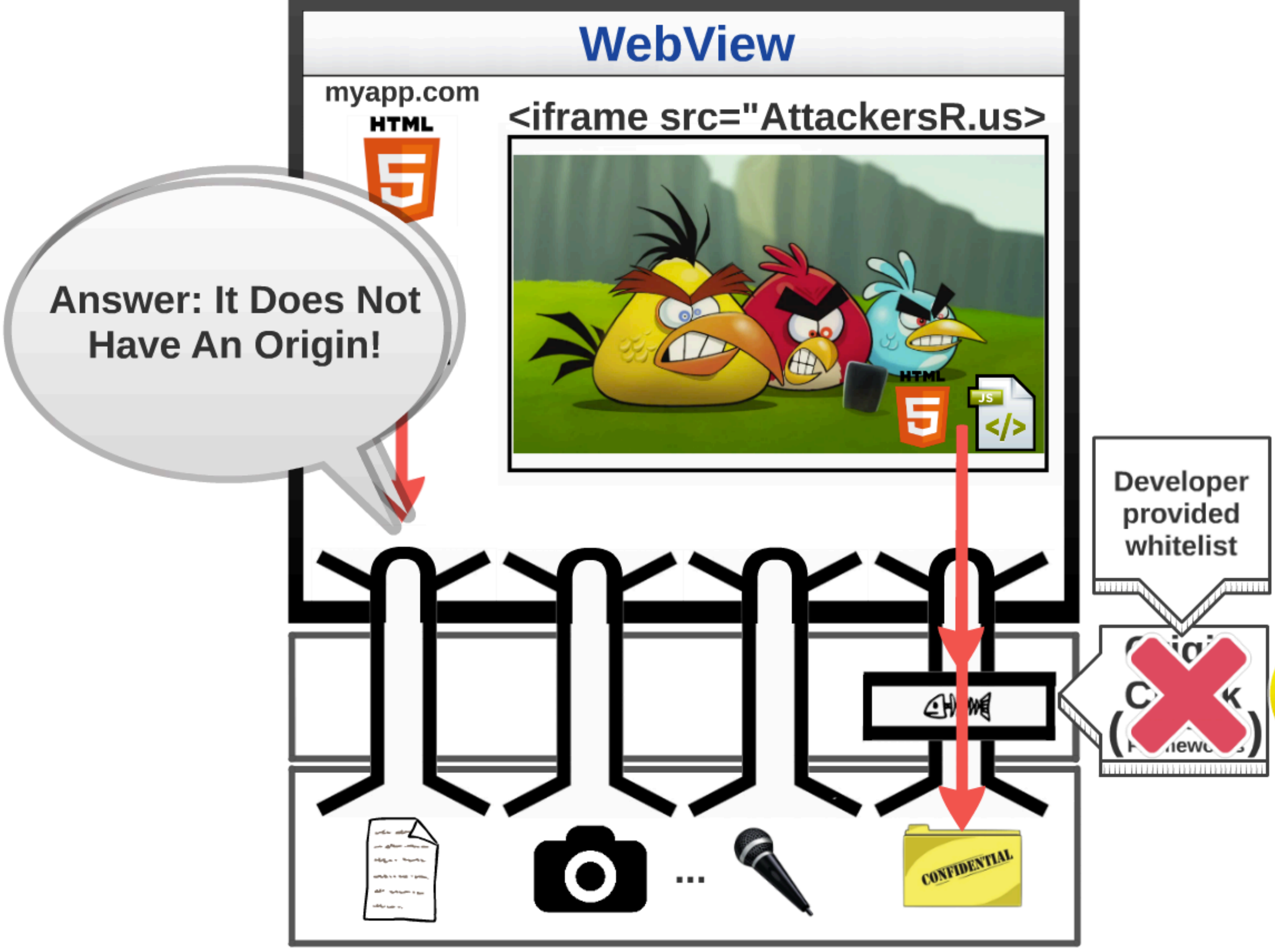
Attack Model



Attack Model



Attack Model



Attack Model

WebView

myapp.com

HTML

<iframe src="AttackersR.us>

All Your Secrets Are Belong To Us!

Answer: It Does Not Have An Origin!

Developer provided whitelist

CONFIDENTIAL

Attack Model

WebView

myapp.com

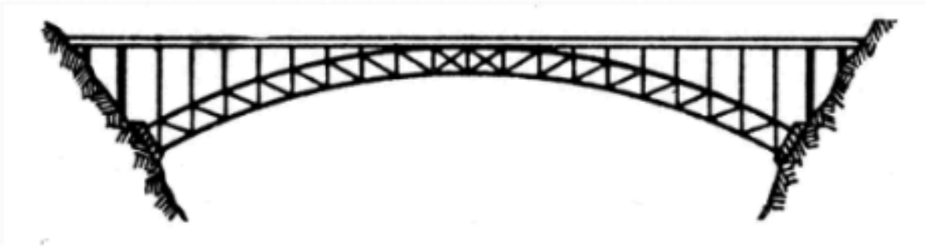
HTML

<iframe src="AttackersR.us>

All Your Secrets Are Belong To Us!

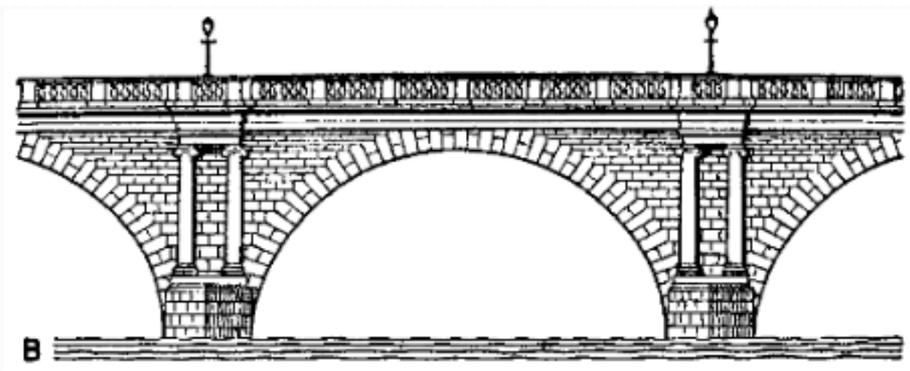
Bridges to Native Resources

addJavascriptInterface



prompt / alert / confirm

loadUrl / executeScript / InvokeScript



cookies

Home-made

Bridges to Native Resources

addJavascriptInterface

prompt / alert / confirm

loadUrl / executeScript / InvokeScript

cookies

Home-made

Bridges to Native Resources

addJa

onfirm

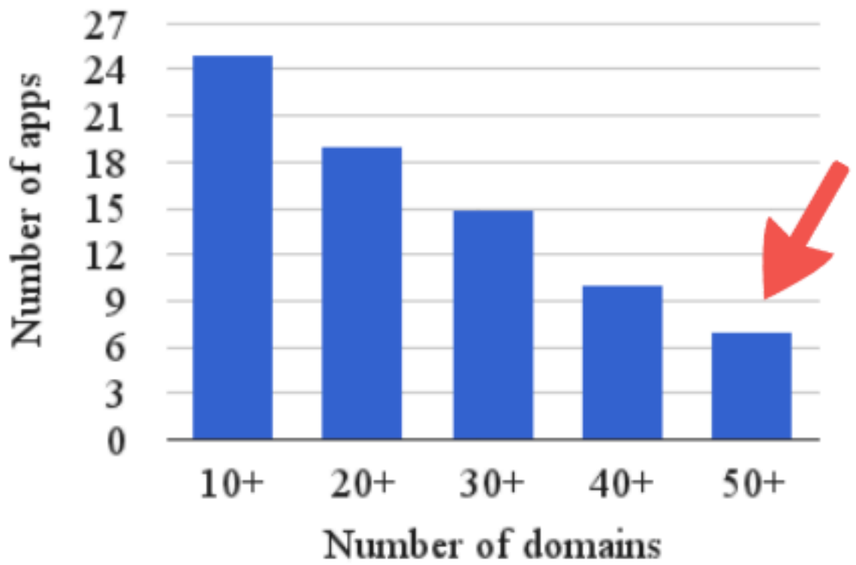
loadUrl / ex

s

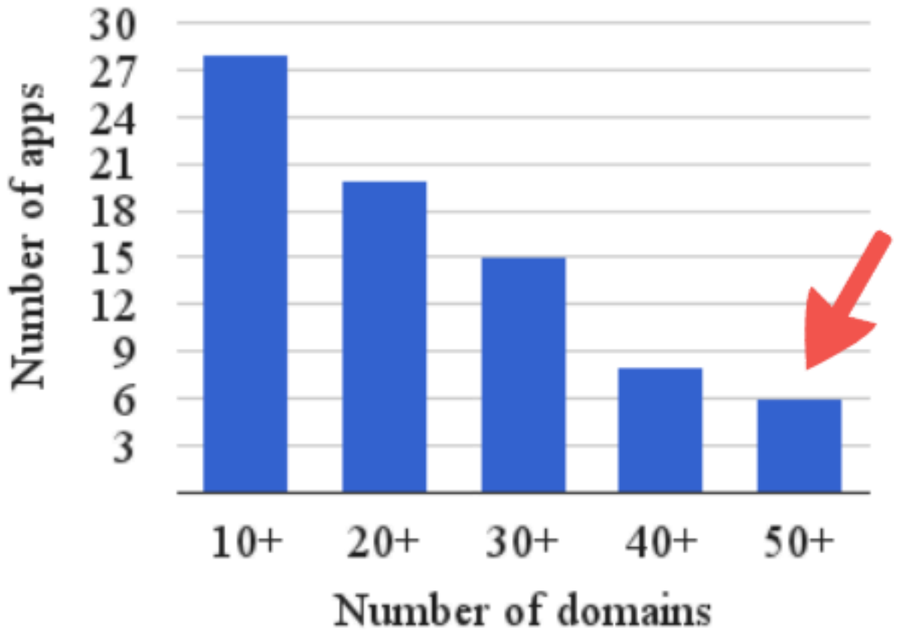
- **wrong regular expressions**
- **wrong URL interception**
- **lack of origin propagation**
- **frame confusion**
- **wrong scheme**

JS

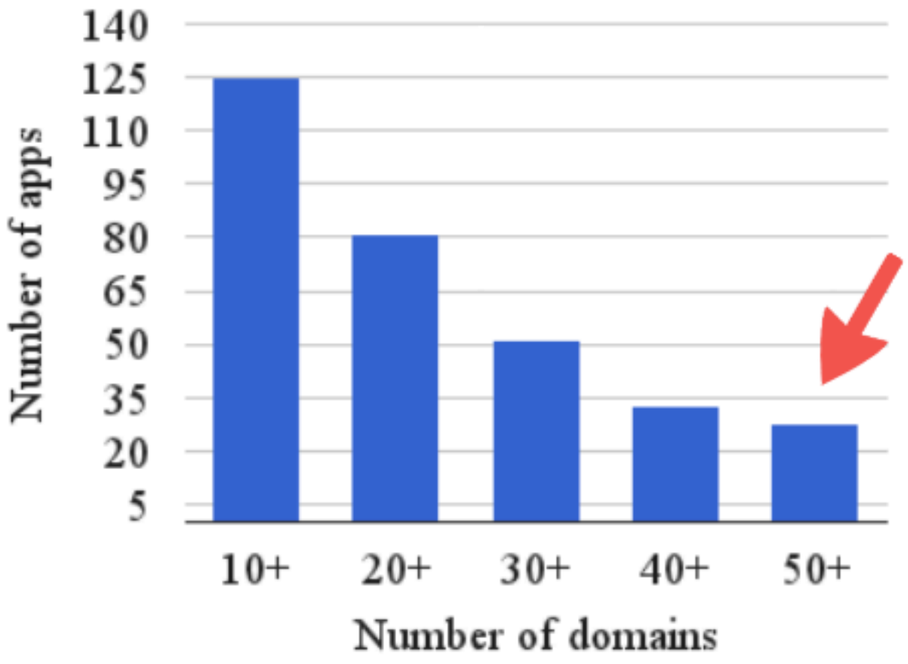
Prevalence



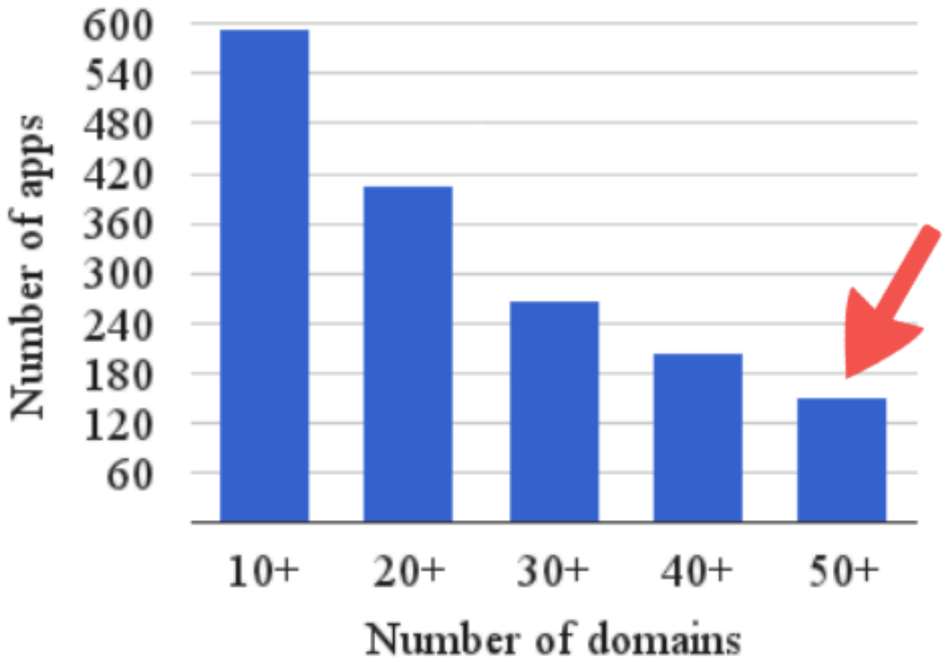
WRITE_CONTACTS



READ_CONTACTS

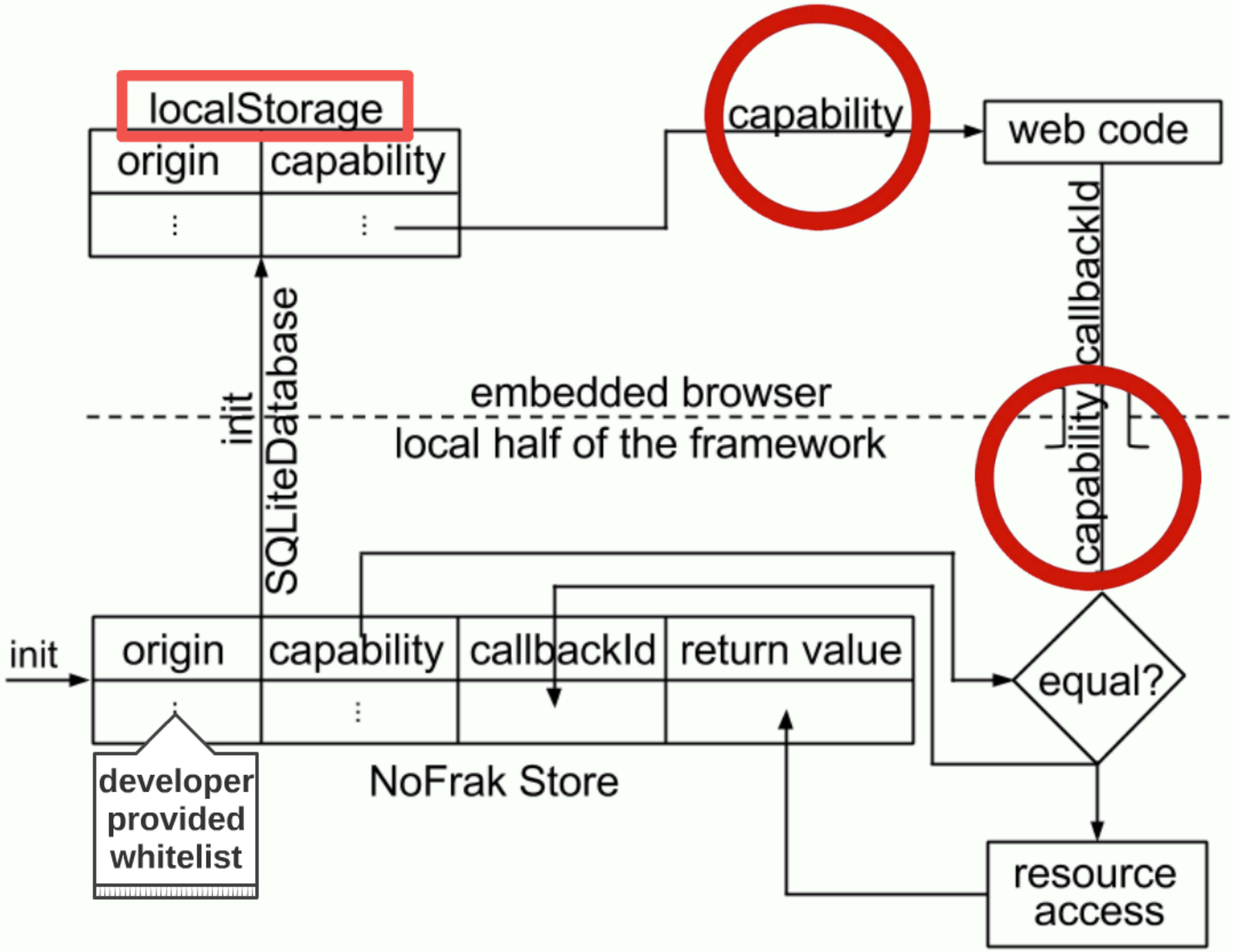


WRITE_EXTERNAL_STORAGE



ACCESS_FINE_LOCATION

Downlink


localStorage
origin
capability
capability
web code
callbackId
embedded browser
local half of the framework
init
SQLiteDatabase
capability
callbackId
init
origin
capability
callbackId
return value
equal?
developer provided whitelist
NoFrak Store
resource access

100% API compatibility

0.24 - 5.8% overhead

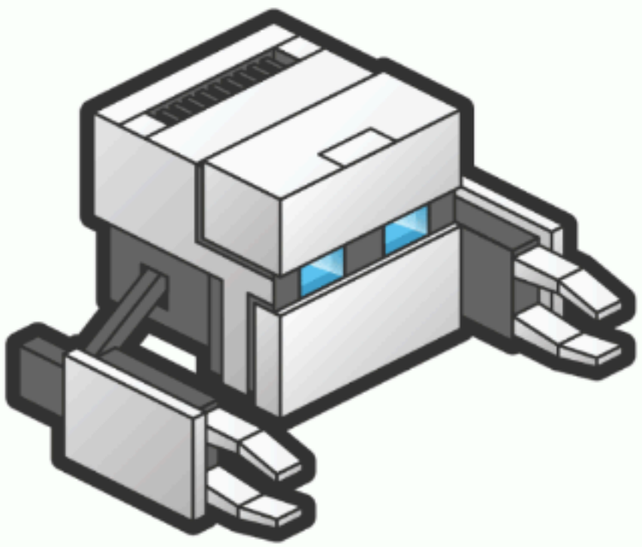
No code changes in existing apps

Same mechanism for all bridges / all platforms

They Are Fixing It Now!

"Adding a SecureToken [NoFrak] is a good idea and we should implement this"

dev@cordova.apache.org



PhoneGap

+

=

It's 11pm. Do you know if your hybrid framework is secure?

- **No way for the app developer to restrict access to native resources (unlike PhoneGap)**
- **No internal framework defenses, either**
- **All apps built on these frameworks are generically insecure**