

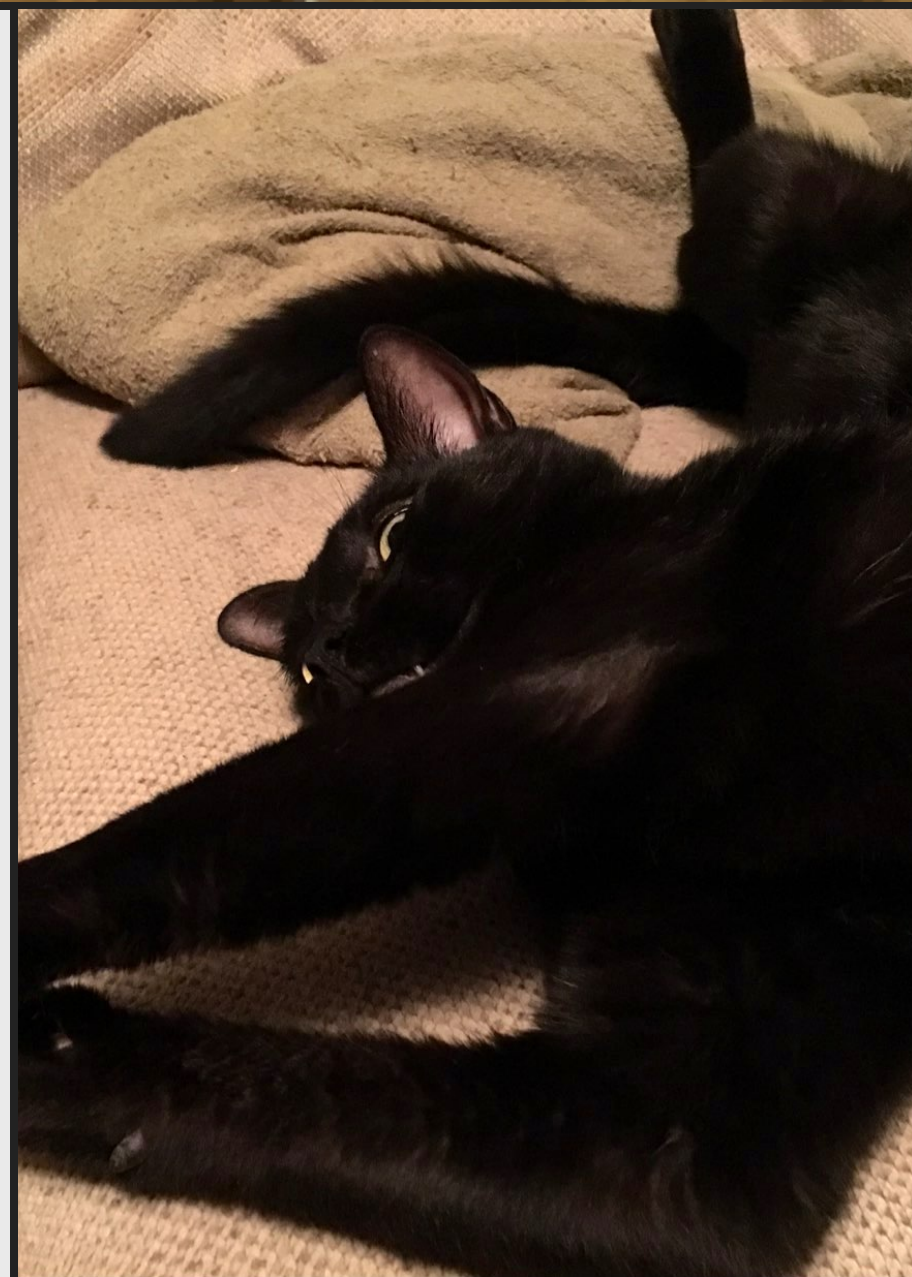
PHONEGAP DAY EU 2016

COMMON PHONEGAP GOTCHAS





[Photo by Bill Bertram](#)



MY NAME IS KERRI

HELLO!

PURPOSE

- IDENTIFY PAIN POINTS
- LEARN THE "EASY" WAY
- HAPPIER USERS
- KEEP BRAIN CELLS

COMMON PAIN POINTS*

*NOT SCIENTIFICALLY DETERMINED

- PLUGIN ISSUES
- XHR / AJAX FAILURES / CSP / CORS
- CLI INSTALLATION
- ICONS / LAUNCH SCREENS
- TREATING PHONEGAP AS A BROWSER / SERVER
- MISUNDERSTANDING PHONEGAP TOOLS
- DATA STORAGE
- RESPONSIVE DESIGN
- SLOW
- DIDN'T EXPECT PLATFORM QUIRKS

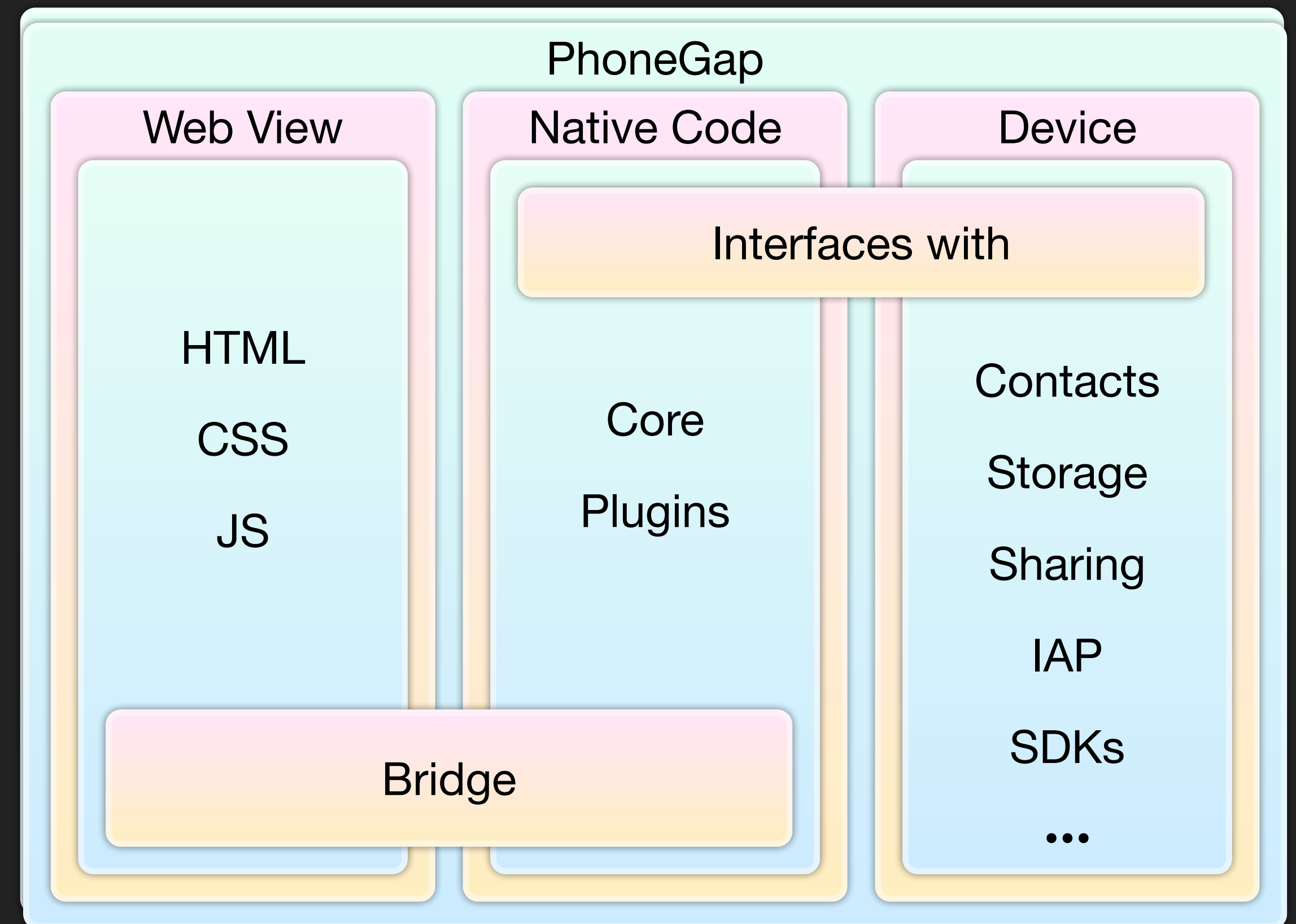
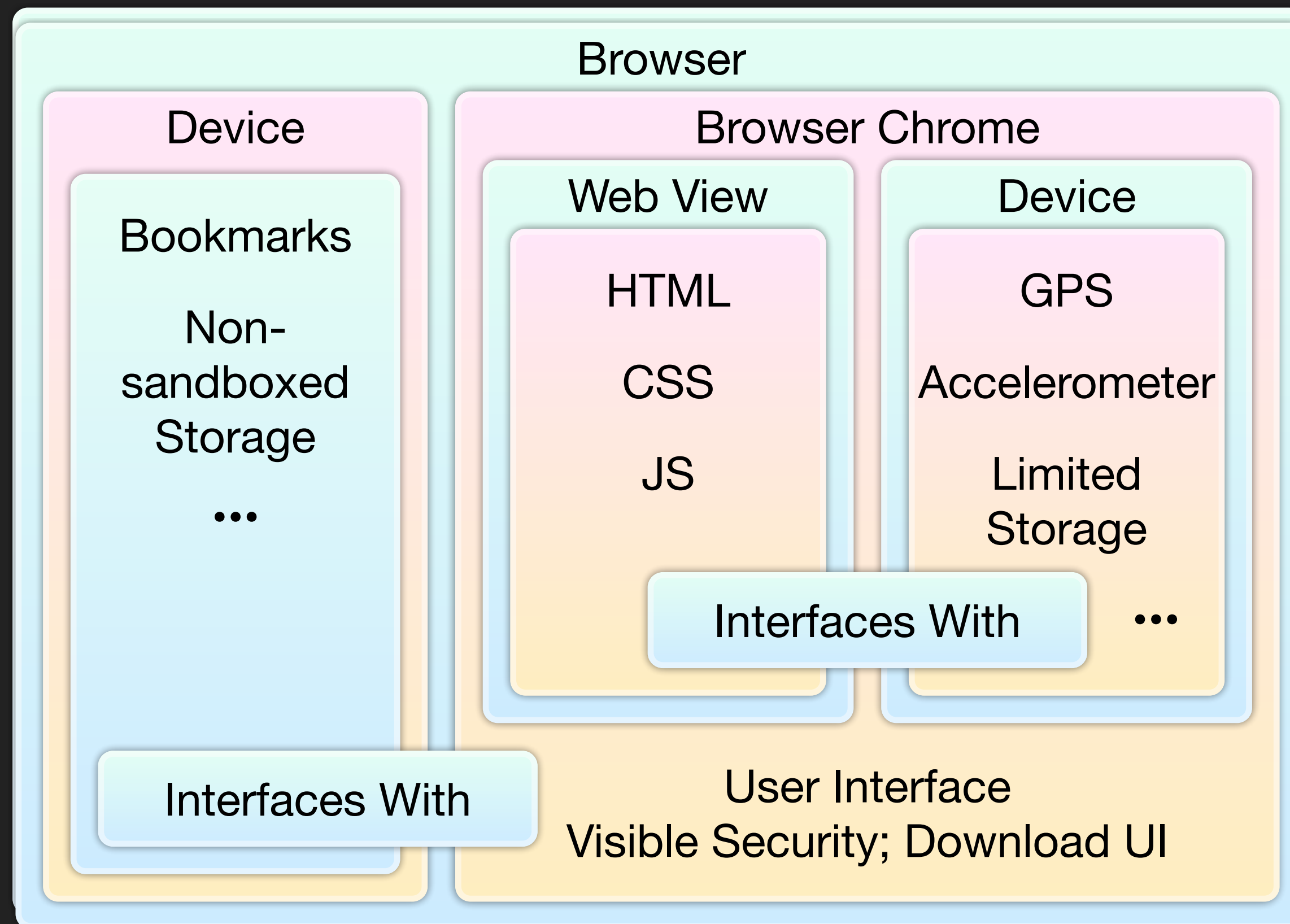
CLI INSTALLATION

- WINDOWS SEEMS TO BE PROBLEMATIC FOR USERS
- NETWORK RESTRICTIONS / PROXIES GET IN THE WAY
- OFTEN HAVE DIFFICULTY CONFIGURING ENVIRONMENT VARIABLES CORRECTLY
- SOMETIMES SECURITY PROGRAMS CAUSE INSTALL PROBLEMS
- NPM WARNINGS LOOK SCARY
- SOME GRUMBLING ABOUT NUMBER OF PREREQUISITES

NOT A BROWSER

- **DON'T WRAP A WEBSITE**
- **KEEP CODE LOCAL**



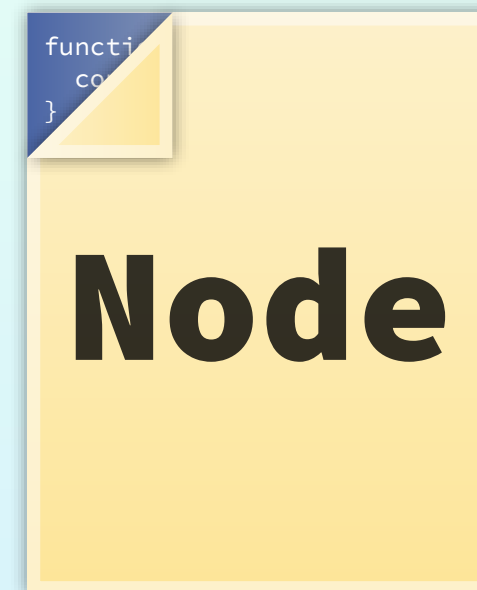


SINGLE PAGE APP (SPA)

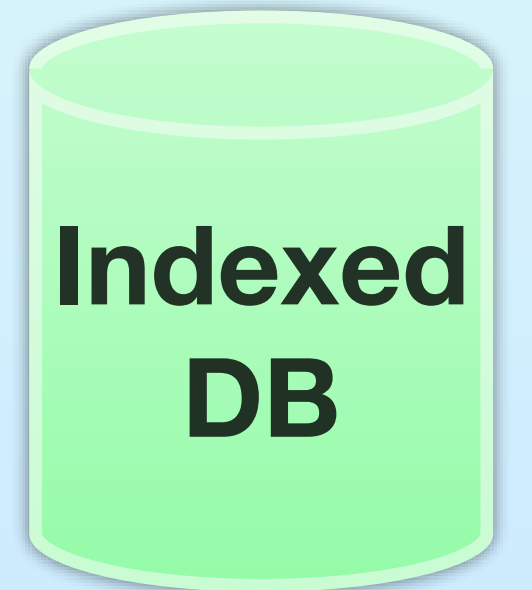
- DYNAMIC DOM
- EASIER ANIMATION
- EASIER APP STATE
- FASTER

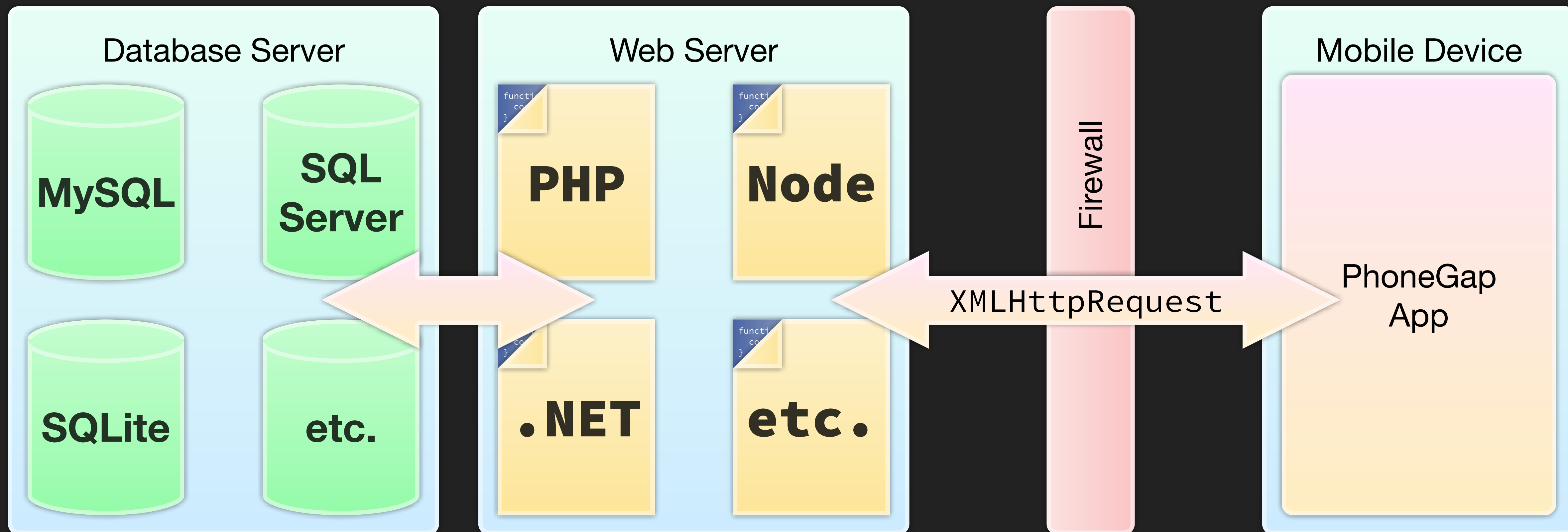
**NOT A
SERVER**

Server



PhoneGap





**CLI, DESKTOP,
DEVELOPER, BUILD,
OH MY!**

- **ONE OF THE FIRST QUESTIONS I ASK: WHICH TOOL ARE YOU USING?**
- **DEVELOPER DOESN'T SUPPORT EVERY PLUGIN**
- **BUILD HAS DIFFERENT STRUCTURE THAN CLI**
- **DESKTOP'S TEMPLATES NEED UPDATED**

	CLI	Desktop App	Developer App	Build
What	Command line interface for creating, building, managing, and running PhoneGap apps	Graphical user interface for creating, managing, and quickly testing PhoneGap apps	Mobile app that connects to the Desktop app or `phonegap serve` for rapid iteration purposes	Cloud service that builds apps for you without needing a local development environment
Benefits	Complete control	Quickly create, manage, and serve apps	Rapid iteration with quick reload	Don't need a local dev environment Upload from CLI, .zip, GitHub Easy distribution
Disadvantages	Need to install prereqs & SDKS More maintenance	Template is out-of-date Have to use CLI to add platforms/plugins Connection difficulties	Only core + select plugins lots of edge cases not present in built apps Connection difficulties	Different project structure PGB vs NPM plugin repo

PUT A PIN IN IT

Platform

```
$ cordova platform add ios@4.1.0 --save  
  
(config.xml)  
<engine name="ios" spec="4.1.0" />
```

Plugins

```
$ cordova plugin add cordova-plugin-device@1.1.1  
--save  
  
(config.xml)  
<plugin name = "cordova-plugin-device"  
spec = "1.1.1" />
```

PhoneGap Build CLI Version

```
<preference name = "phonegap-version"  
value = "cli-6.0.0"/>
```

PLUGIN ISSUES

- NOT INSTALLED OR INSTALLED INCORRECTLY
- NOT WAITING FOR DEVICEREADY
- PASSING INCORRECT/MALFORMED PARAMETERS
- PAY ATTENTION TO ERROR CALLBACK!
- DOCUMENTATION IS OFTEN AN ISSUE
- CAN LEAD TO FEELINGS OF POOR QUALITY

DATA STORAGE

- **LOCALSTORAGE IS TEMPTING, BUT A POOR CHOICE**
- **SYNC IS A BIG QUESTION (COUCHDB & POUCHDB)**
- **USE THE BEST STORAGE OPTION FOR THE DATA**

localStorage

WHAT IS IT GOOD FOR?

NOT MUCH

	localStorage	<u>File API</u>	<u>Web SQL</u>	<u>IndexedDB</u>	SQLite (<u>1</u> , <u>2</u> , <u>3</u>)	<u>App Prefs</u>	<u>Keychain</u>
Quota	~5mb	n/a	5 - 50mb	5 - 50mb	n/a	platform specific	platform specific
Async?	✗	✓	✗	~	~	✓	✓
Format	Key / String	Text / Binary	Structured data	Store / Key / Object	Structured data	Key / String	Key / String
Good for	Not much Caching Temporary data	Saving and loading documents	Storing & querying structured data	Storing semi- structured JavaScript objects	Large amounts of structured data	User Settings	Sensitive Information
Note	Not persistent, No callback	Persistent <i>and</i> temporary file systems	Deprecated	Bugged on iOS (buggyfill)	Keep result set small!	Not good for lots of data	Not good for lots of data

**DON'T STORE SENSITIVE
INFORMATION**

**YOUR USER'S DATA AND
PRIVACY ARE PRICELESS**

ALWAYS USE HTTPS



[Blog](#)

[Technology ▾](#)

[Sponsors ▾](#)

[Support ▾](#)

[About ▾](#)

Let's Encrypt is a new Certificate Authority:

It's free, automated, and open.

[Get Started \(Public Beta\)](#)

DON'T TRUST YOUR DATA

- VALIDATE EVERYWHERE
- DOM INJECTION
- SQL INJECTION

// DOM INJECTION:

```
let untrustedData = `<button onclick="doSomethingNefarious();"/>`;
el.innerHTML = untrustedData;    // SUSCEPTIBLE
el.textContent = untrustedData; // SAFE
```

// SQL INJECTION:

```
{
  let sql = "select * from user where username = " + username;
  db.transaction(tx => tx.executeSql(sql));
} // SUSCEPTIBLE
```

```
{
  let sql = `select * from users where username = ?`;
  db.transaction(tx => tx.executeSql(sql, [username]));
} // SAFE
```

NETWORK CONNECTIVITY

- LOTS OF MISUNDERSTANDING ABOUT WHITELIST AND CONTENT-SECURITY-POLICY

**CONFIGURE THE
WHITELIST**

```
$ cordova plugin add cordova-plugin-whitelist --save
```

Network Access

```
<access origin="https://www.example.com" />  
<access origin="https://*.example.com" />  
<access origin="*" /> <!-- NOT SECURE -->
```

Top-level Navigation (location .href)

```
<allow-navigation href="https://*.example.com/*" />
```

CSP META TAG

- TIGHTEN AS MUCH AS POSSIBLE
- DEV WILL BE MORE PERMISSIVE

<!-- Development CSP -->

```
<meta http-equiv="Content-Security-Policy"
      content="default-src 'self' gap: https://ssl.gstatic.com;
              connect-src 'self' ws: https://test.example.com;
              img-src     'self' data:;
              style-src   'self' 'unsafe-inline';
              script-src  'self' 'unsafe-eval' 'unsafe-inline'
              gap:;"/>
```

<!-- Production CSP -->

```
<meta http-equiv="Content-Security-Policy"
      content="default-src 'self' gap: https://ssl.gstatic.com;
              connect-src 'self' https://www.example.com;
              img-src     'self' data:;
              style-src   'self';
              script-src  'self' gap:;"/>
```

'self'	Required to load local resources
gap:	UIWebView only (iOS) — used by web-native bridge
ws:	Web socket support (needed for live reload)
<u>https://ssl.gstatic.com</u>	Android TalkBack (Accessibility)
data:	Inline images, including SVG use
'unsafe-eval'	Try to avoid, but not always possible (live reload / plugins)
'unsafe-inline'	Try to avoid, but not always possible (live reload / plugins)

CROSS-ORIGIN RESOURCE SHARING (CORS)

- SERVER-SIDE
- REQUIRED FOR WKWEBVIEW, PHONEGAP SERVE, AND BROWSER

**CONNECTIVITY IS NOT
GUARANTEED**

```
if (!navigator.onLine) {  
    throw new OfflineError(); // offline; don't bother  
}  
let xhr = new XMLHttpRequest();  
xhr.open(method, URI, true);  
xhr.timeout = 30000; // 30 second timeout  
xhr.ontimeout = function() {  
    throw new TimeoutError(); // might be offline or unreachable. If online, try  
                                // again (throttle); error after too many failures  
                                // Watch for long-running queries!  
}  
xhr.onerror = function() {  
    throw new XHRError(); // try again; error after too many failures  
}  
xhr.onload = function() {  
    if (this.status >= 200 && this.status <= 299) { /* success! */ }  
    else { throw new HTTPError(this.status); }  
}  
xhr.send(data);
```

ICONS & LAUNCH SCREENS

- **PATHS ARE RELATIVE TO CONFIG.XML**
- **USE MODERN `<SPLASH>` AND `<ICON>` TAGS**
- **MAKE SURE THE IMAGE IS SAME SIZE AS SPECIFIED IN CONFIG.XML**
- **KEEP LAUNCH SCREEN SIMPLE**

PERCEIVED AS SLOW

- 300MS DELAY ****SIGH****
- JANKY ANIMATIONS
- PROFILE!

BAN SLOW INTERACTIONS

- FAST CLICK OR GESTURE
LIBRARY (1, 2)
- AFFORDANCES
- DELAY WORK UNTIL
AFTER ANIMATIONS

PROFILE

- TRACK MEMORY, FRAMES, EXECUTION TIME
- CHROME / SAFARI + INSTRUMENTS
- SCREEN RECORDING

PERFORMANCE MATTERS

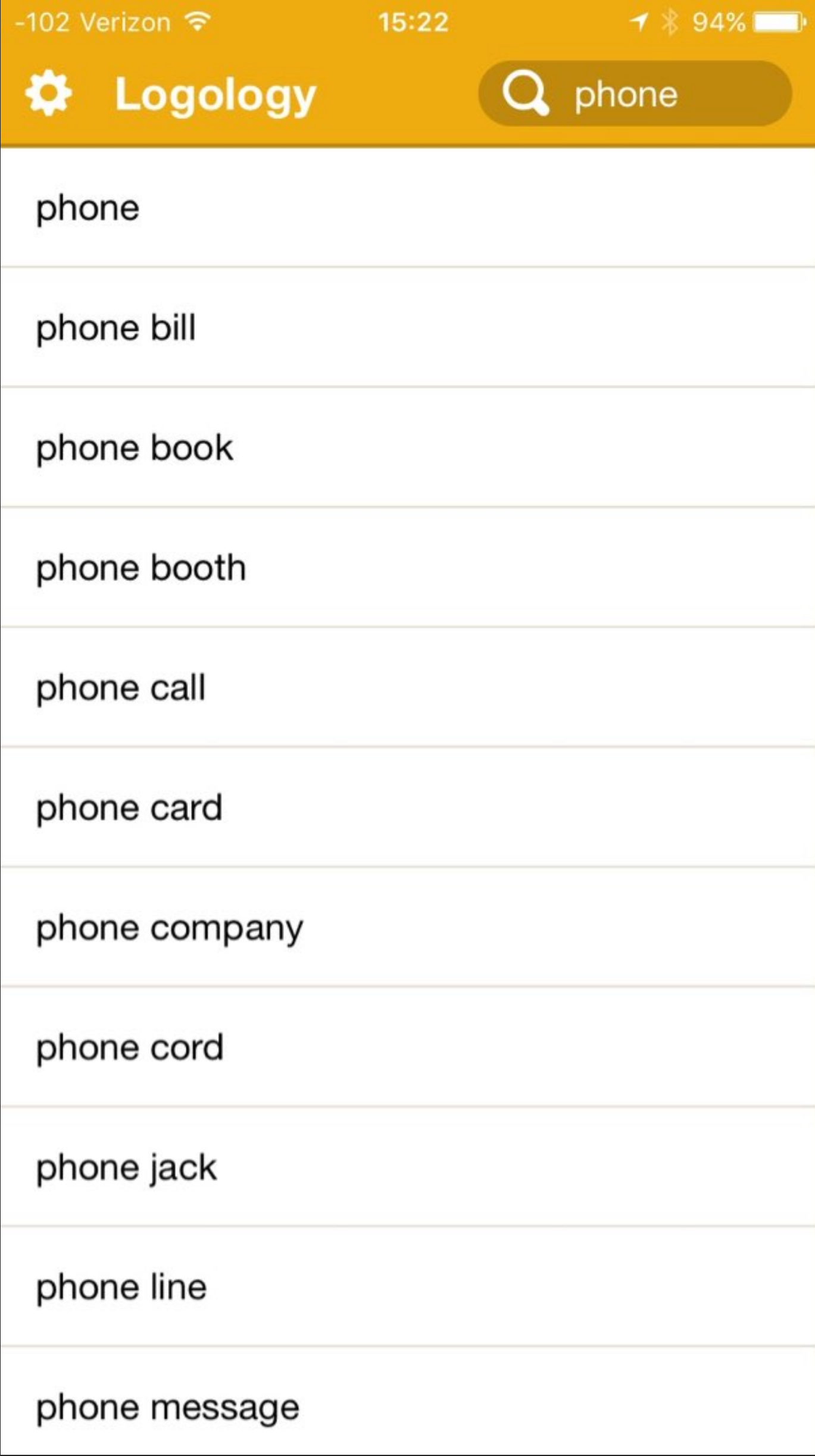
Quantifiable Performance

Bruce Lefebvre

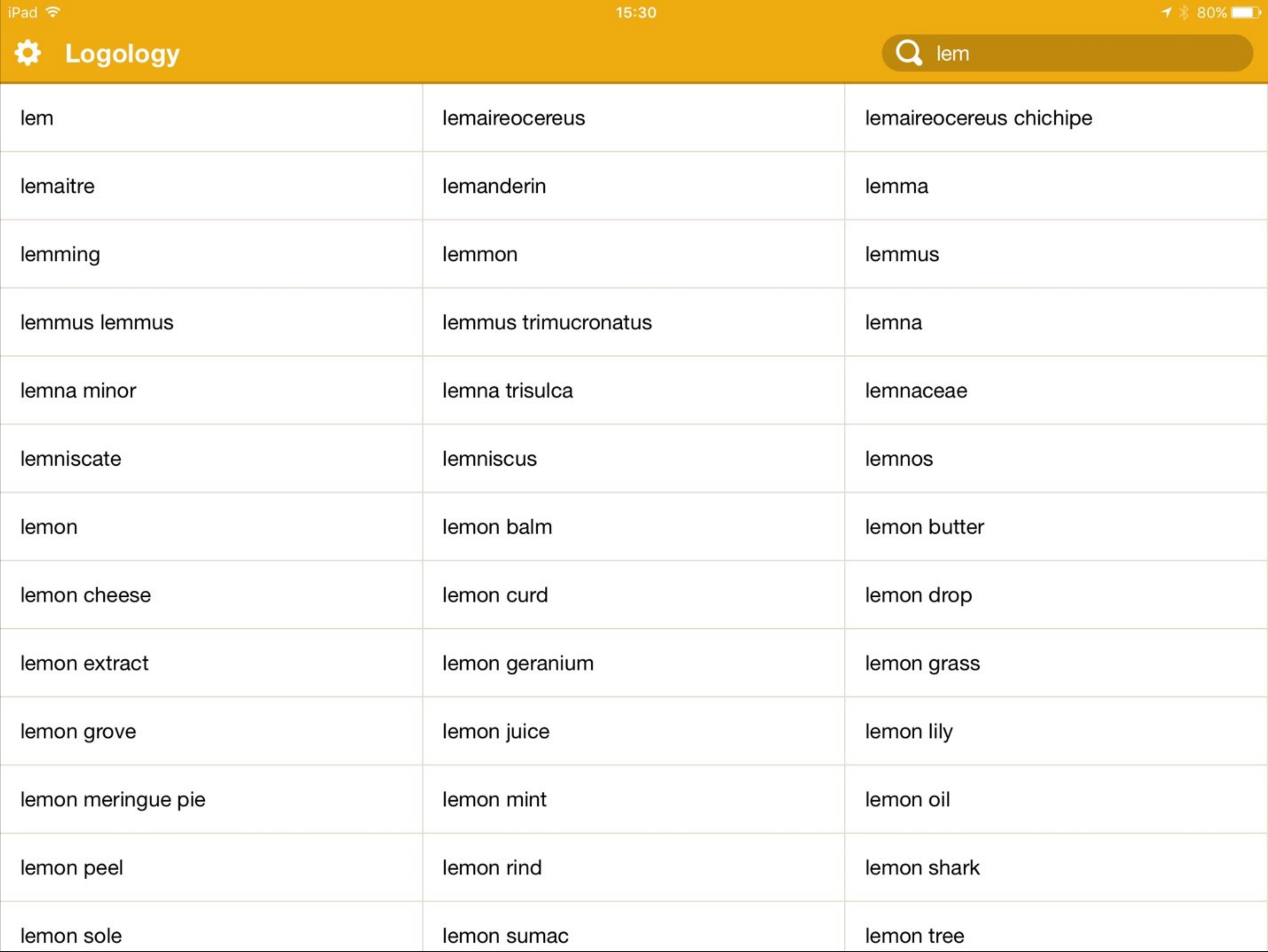
11:20am

DESIGN RESPONSIVELY

- HTML IS ALREADY RESPONSIVE
- RESPONSIVE UNITS (% , REM,
VIEWPORT
UNITS:VH,VW,VMAX,VMIN)
- MEDIA QUERIES & DEVICE
PLUGIN
- FLEXBOX LAYOUT



iPhone

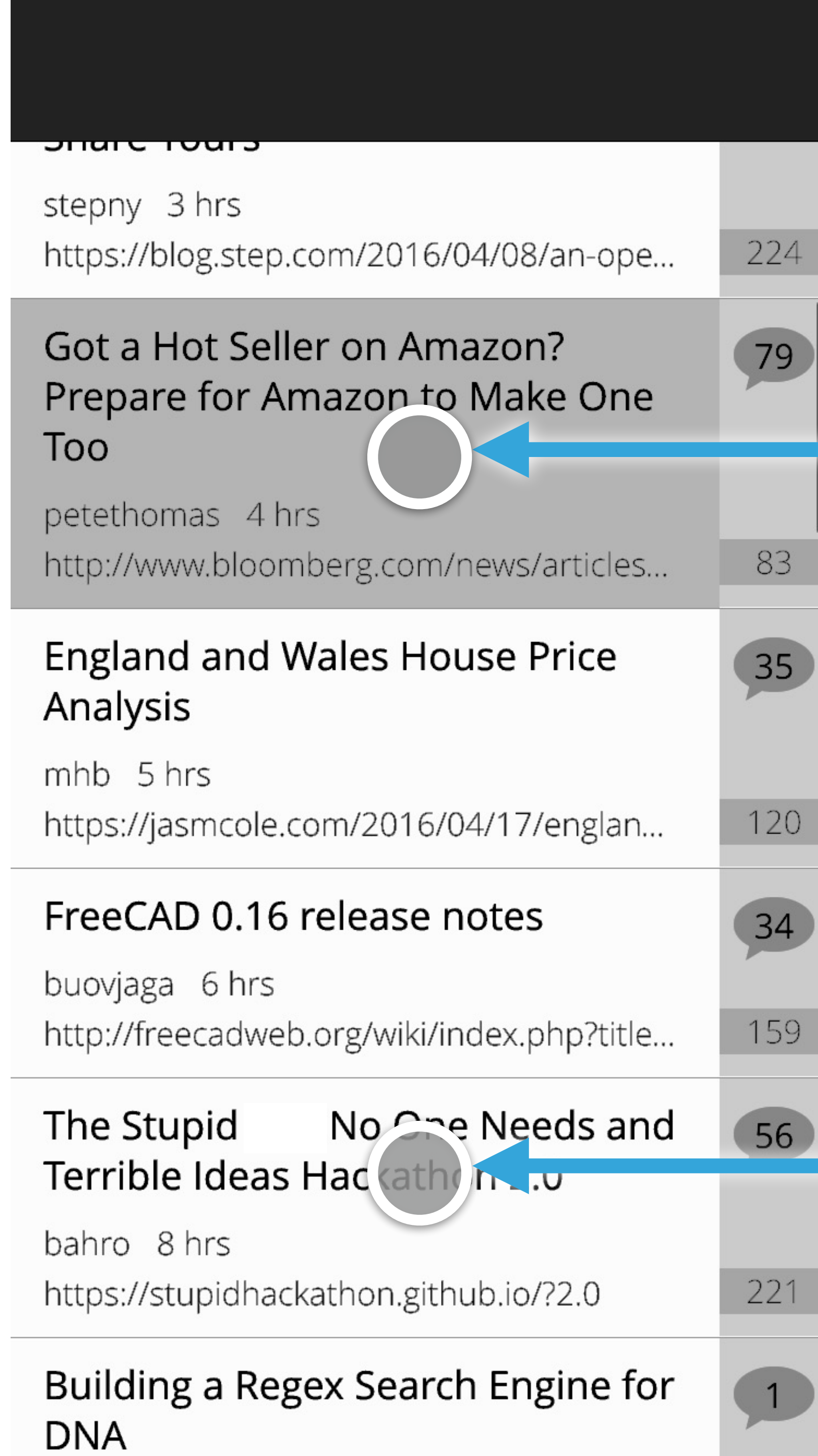


iPad (landscape)

MOBILE IS QUIRKY

- MOBILE WEB IS QUIRKY
- ACCEPT DEVICE & PLATFORM-SPECIFIC WORKAROUNDS
 - DEVICE PLUGIN
 - CROSSWALK
 - WKWEBVIEW

iOS UIWebView Scroll Offset Bug



Touch is registered *here*

My finger is actually here

```
# iOS 9 (UIWebView on iOS 8)
$ cordova plugin add cordova-plugin-wkwebview-engine --save

# iOS 8 & 9
$ cordova plugin add https://github.com/apache/cordova-
plugins.git#master:wkwebview-engine-localhost --save
# Then, edit config.xml
<platform name="ios">
  <content src="http://localhost:49000"/>
</platform>

# Android
$ cordova plugin add cordova-plugin-crosswalk-webview --save
```

MANAGE THE KEYBOARD

Handling the keyboard in
hybrid applications

Tim Lancina

10:15am

THANK YOU!

@photokandy
www.photokandy.com