# Boot2Gecko

Chris Jones, Andreas Gal, and the B2G Team

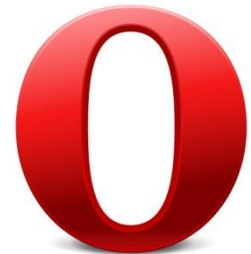Mozilla Vision 2012

Tokyo, Japan
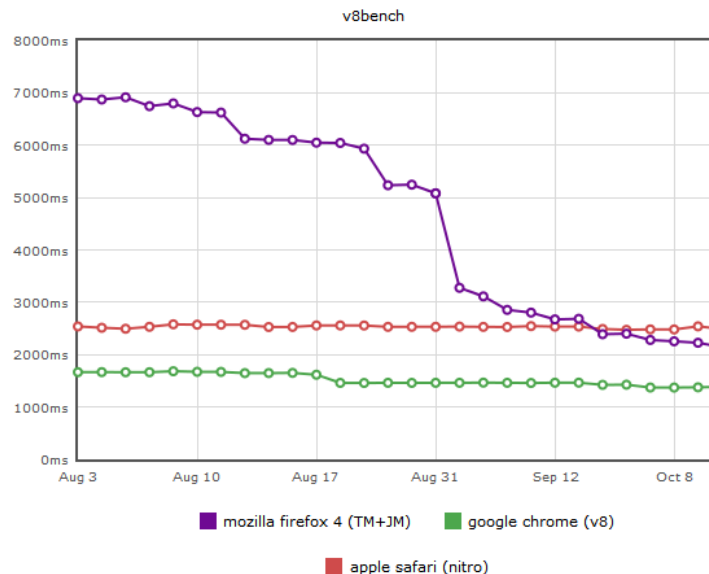
# The Web, 2002

# The Web, Today

Standards-based
web pages

Web standards

# Why did this change happen?

- **Mozilla!**

- How: ***Interoperable standards***

- Authors "write once", their code "runs everywhere"*
  - Happy authors!

- Browsers compete over quality of implementation
  - Better user experience!
  - Happy users!
  - Better standards!

v8bench

8000ms
7000ms
6000ms
5000ms
4000ms
3000ms
2000ms
1000ms
0ms

Aug 3    Aug 10    Aug 17    Aug 31    Sep 12    Oct 8

■ mozilla firefox 4 (TM+JM)    ■ google chrome (v8)

■ apple safari (nitro)

* mostly

# The Web on "desktop" and "mobile"

- **On desktop, web apps rule**
  - Mail clients, maps, office suites, video players, games, …
  - "Native apps" and "web apps" mostly have the same capabilities

- **On mobile, web apps** … where are they?
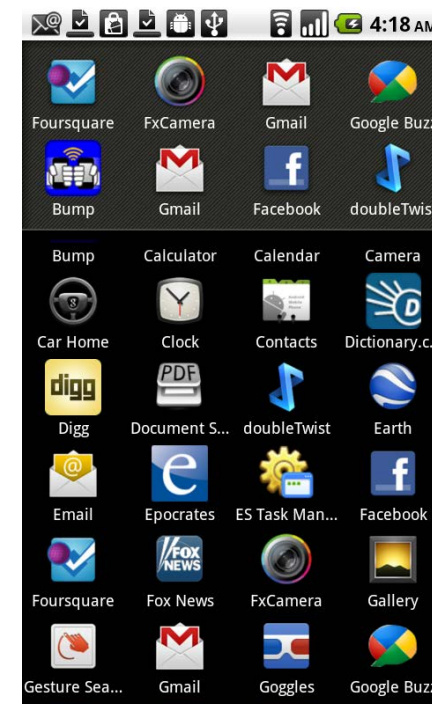
➔ *Why the difference?*

# Mobile ecosystems are closed "silos"

STORE     APPS     WEB

PLATFORM

DEVICE
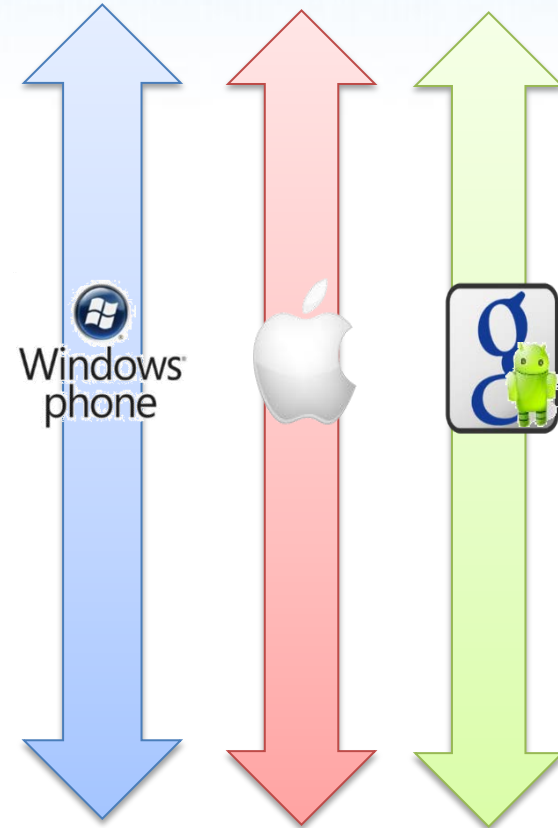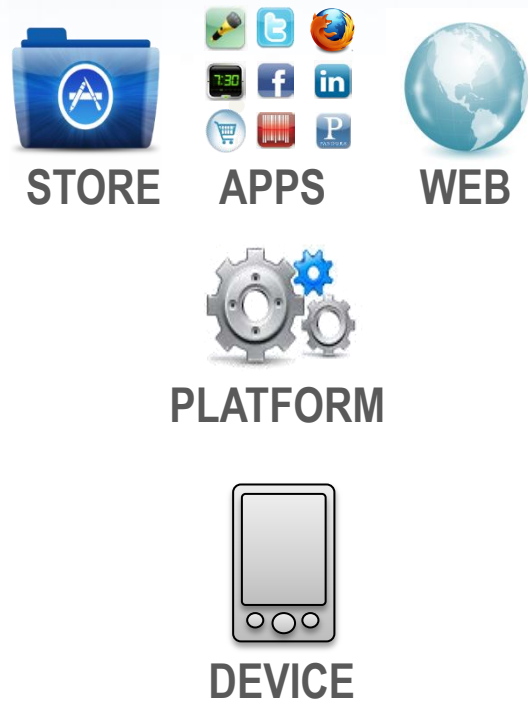
- Silos are bad for pretty much everyone.

- In these silos, the web platform is a second-class citizen.

➔ *We can change this*

# The Open Web platform needs more capabilities

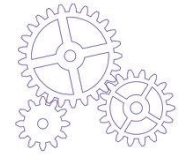**Telephony**     **Gestures**     **Audio**     **Location**     **Settings**

**Contacts**     **NFC**     **Camera**     **Bluetooth**     **SMS**

- APIs in iOS/Android/… but not in the web platform, are bugs.
➔ *We are fixing these bugs*

# Need an ecosystem without lock-in

**App Markets**

Built on interoperable standards

**Sync**

Sync data across devices

**BrowserID**

Control your identities

# Need to ship a phone: Boot2Gecko!



- **Competitive user experience**
  - Meet or exceed native app performance
  - Mozilla building *Gaia*

- **Build on open standards**
  - Drive device API standards
  - No Gecko-specific APIs

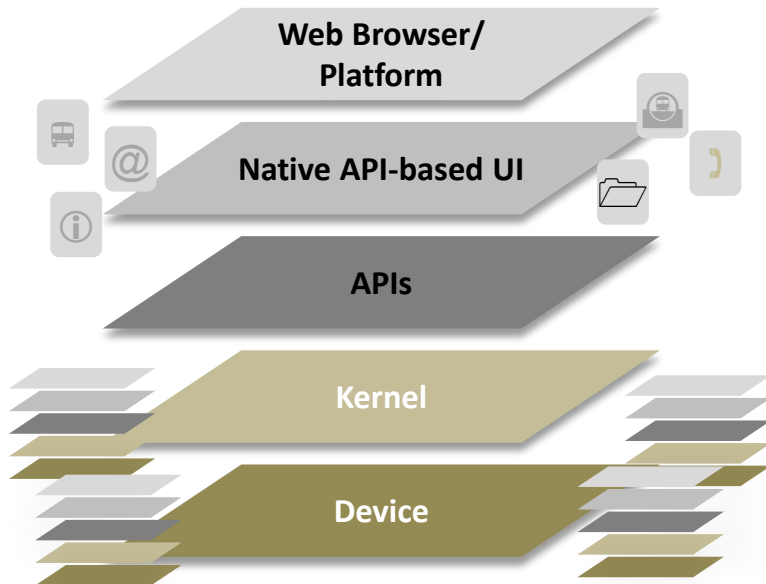- **Apps**
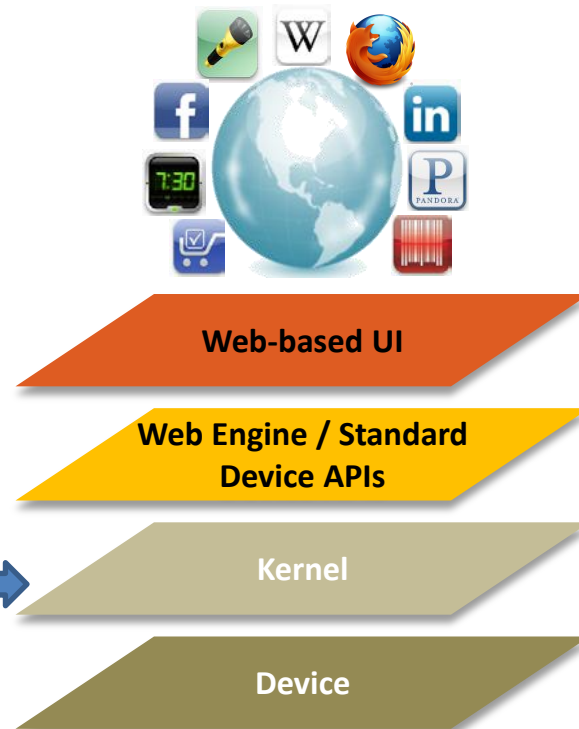  - Web is the App store
  - Cache locally and update from "the cloud"

# TIME FOR A DEMO

# Boot2Gecko removes a layer from phone stacks

**Existing Platforms**

**BOOT2GECKO**

| Web Browser/ Platform |
| Native API-based UI |
| APIs |
| Kernel |
| Device |

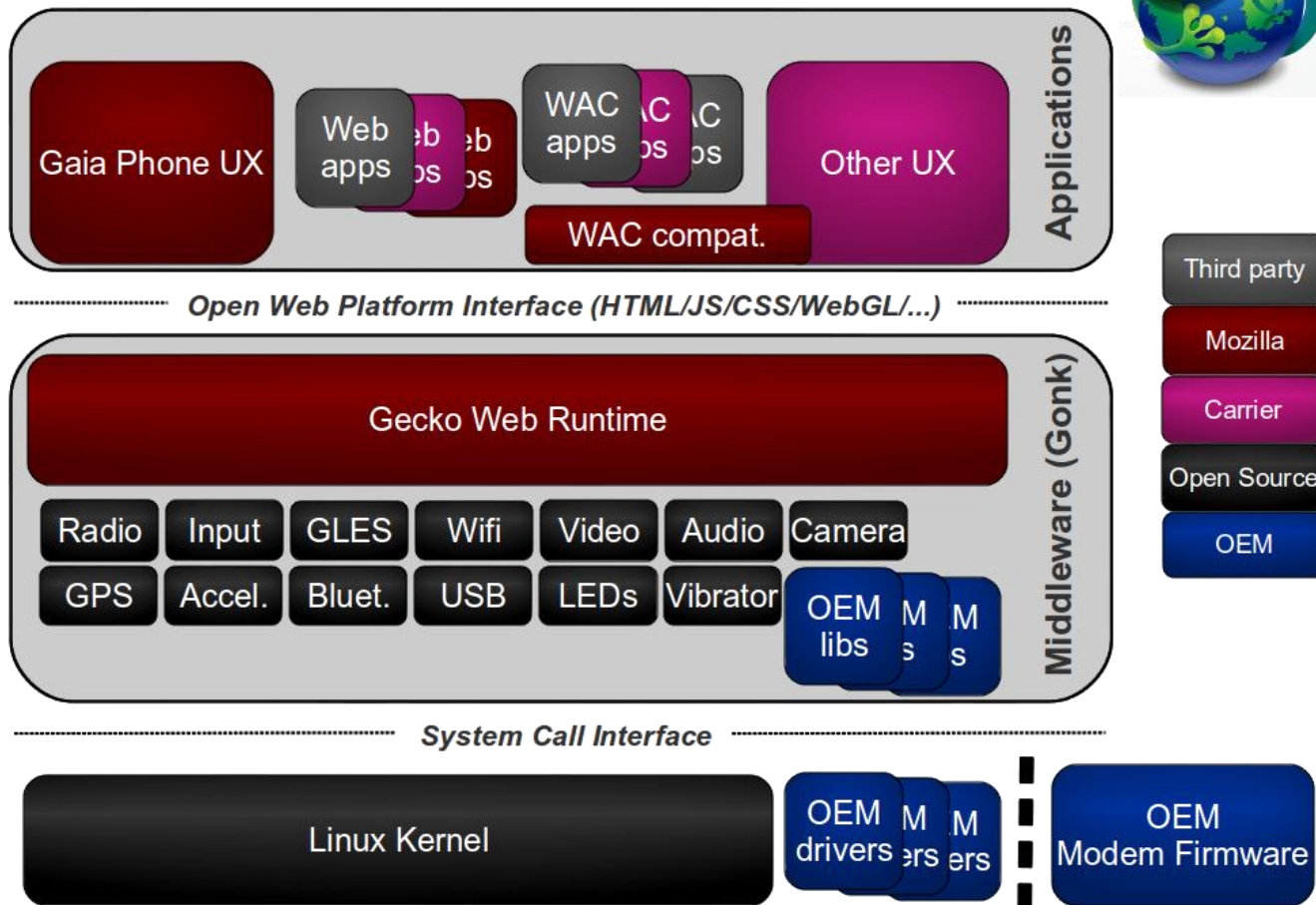| Web-based UI |
| Web Engine / Standard Device APIs |
| Kernel |
| Device |

Shorter stack: Use **less resources**, run **faster**

**Reuse existing code** when possible

# The guts of B2G

# Bring Mozilla values to mobile OSes

- Plan and develop in the open
  - Project roadmap is at https://wiki.mozilla.org/B2G/Roadmap
  - Source code always available https://github.com/andreasgal/B2G
- Standardize new APIs we create


- Create an open ecosystem


- Encourage interoperable implementations
  - We would love to see Boot2Webkit!
- Build user interface and apps on standards
  - We want our apps to run on Boot2Webkit too!

# Why (we think) B2G can succeed

- There are billions of web pages / apps
- There are millions of web developers

- There are billions of installations of the "B2G SDK" already
  – Standards-compliant web browsers!

- The web platform is richly capable already
  – Advanced layout, high-performance graphics, fast JavaScript, …
  – Missing pieces (e.g. telephony) are small in comparison

# Contribute

- We have a long way to go.

- We need apps
  - Keep writing web apps!
  - https://github.com/andreasgal/gaia

- We need to keep improving Gecko
  - Implement missing APIs.  Make Gecko faster.
  - https://bugzilla.mozilla.org

**EXTRA SLIDES**

# Will Mozilla ship a B2G phone?

- No, not directly

- We'll work with partners to ship phones to consumers

# What's the security model of B2G?

- **All Apps & Pages Assumed Hostile:**
  - Fully sandboxed
  - No access to sensitive APIs (e.g. Telephony)
  - That's all most apps need!

- **App developers & markets can decide**
  - "Bless" apps with more trust
  - Assign limited set of capabilities (Telephony but not USB, e.g.)

- **Users can change the settings:**
  - Grant privileges to unknown/untrusted apps UI (permanently or temporarily)
  - Grant of trust is natural and obvious

# What's a "web page" and what's a "web app"?

- They are the same thing.  Except
- Web apps have an application manifest, which specifies
  - Resources that need to be cached for offline use (appcache)
  - High-resolution icons (no more favico!)
  - Permissions that are needed/requested
  - Intents that are handled
  - [a long list of other configuration parameters]

- Web apps can run without a manifest
- Web pages can become web apps by adding a manifest