



Firefox OS



Sam Foster

sfoster@mozilla.com



[samfosteriam](#)



[sfoster](#)

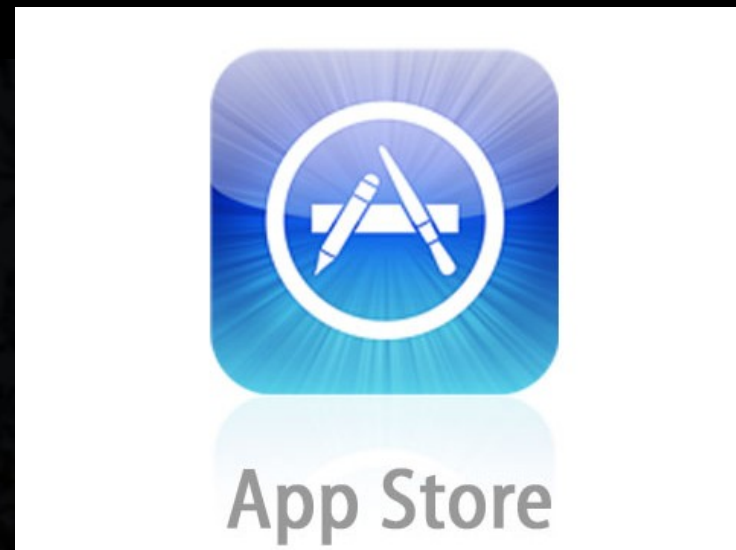
Firefox OS



- Open source mobile OS (Apache 2.0)
- 14 operators / 28+ countries
- Built by Mozilla + partners + contributors

Why Firefox OS?

- Web under threat from parallel content universes
- Walled gardens and gatekeepers
- Duopoly (Monopoly?)
- The web *is* the platform



Firefox OS Architecture

The diagram illustrates the Firefox OS architecture as a stack of four layers. From top to bottom, the layers are: Gaia (blue), Gecko (orange), Gonk (grey), and Mobile device (dark red). Each layer is represented by a horizontal bar with its name centered inside.

Gaia

Gecko

Gonk

Mobile device

Architecture

- **Gaia:** Application Layer (HTML + JS + CSS)
- **Gecko:** Web Platform Layer (Communication, Data Mgmt, Hardware access and other APIs)
- **Gonk:** Infrastructure (Linux kernel, OSS/OEM libs, drivers + firmware)

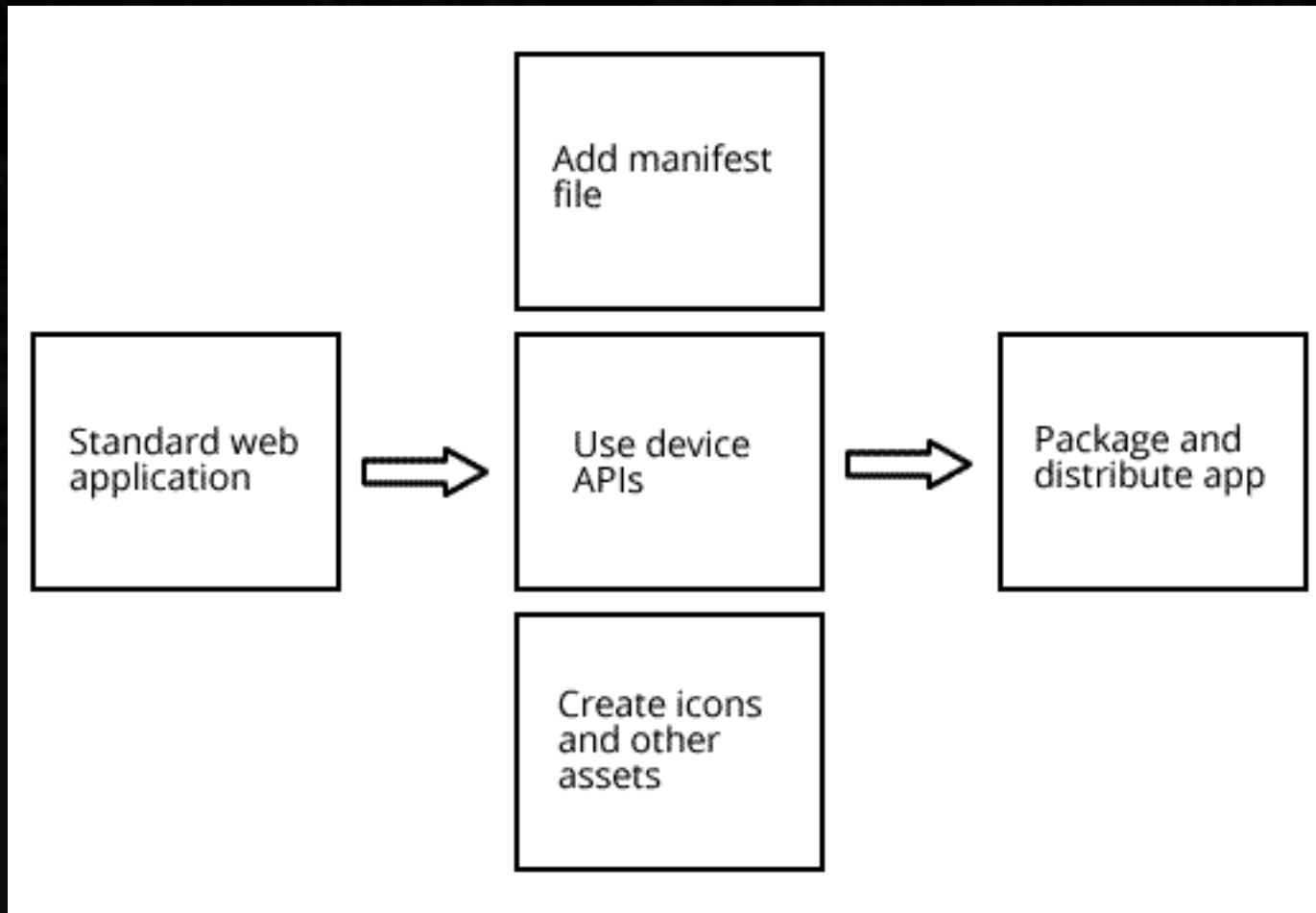
State of Play

- v1 – 1.4 Shipped, handsets available in markets in Europe, Latin America, Asia
- V2.0 Shipping
- V2.1 Locked, testing and certification
- V2.2 WIP
- V3 Planning underway

Features: Quick Tour

- Dialer, SMS etc. Dual SIM support
- Camera, Gallery
- Bluetooth, NFC
- FM Radio
- And much more...

Making Apps



API Permissions

The screenshot shows the Mozilla Developer Network (MDN) documentation for Web APIs, categorized into three main sections based on permission levels: Communication APIs, Data management APIs, and Other APIs. Each API entry includes a description, a permission icon (Common, Privileged, or Certified), and a status label (Certification required or Non-standard).

Communication APIs

- Network Information API**
Provides basic information about the current network connection, such as connection speed.
- Bluetooth** (Certification required, Non-standard)
The WebBluetooth API provides low-level access to the device's Bluetooth hardware.
- Mobile Connection API** (Certification required, Non-standard)
Exposes information about the device's cellular connectivity, such as signal strength, operator information, and so forth.
- Network Stats API** (Certification required, Non-standard)
Monitors data usage and exposes this data to privileged applications.
- TCP Socket API** (Privileged, Non-standard)
Provides low-level sockets and SSL support.
- Telephony** (Certification required, Non-standard)
Lets apps place and answer phone calls and use the built-in telephony user interface.
- WebSMS** (Certification required, Non-standard)
Lets apps send and receive SMS text messages, as well as to access and manage the messages stored on the device.
- WiFi Information API** (Certification required, Non-standard)
A privileged API which provides information about signal strength, the name of the current network, available WiFi networks, and so forth.

Data management APIs

- FileHandle API** (Non-standard)
Provides support for writable files with locking support.
- IndexedDB**
Client-side storage of structured data with support for high-performance searches.
- Contacts API** (Privileged)
Provides access to the user's contacts database, with support for adding, reading, and modifying contact information.
- Device Storage API** (Privileged, Non-standard)
Allows apps to create, read, and change files stored in a central location on the device, such as the "pictures" folder on modern desktop platforms or the photo storage on mobile devices.
- Settings API** (Certification required, Non-standard)
Lets apps examine and change system-wide configuration options that are permanently stored on the device.

Other APIs

- Alarm API**
Lets apps schedule notifications. Also provides support for automatically launching an app at a specific time.
- Simple Push API**
Lets the platform send notification messages to specific applications.
- Web Notifications**

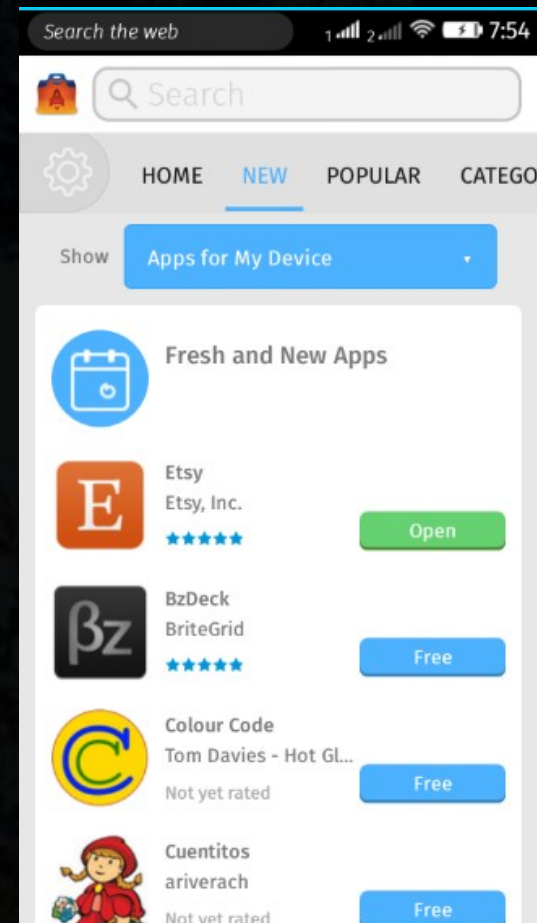
Common, Privileged, Certified

Dev Tools

The image shows a screenshot of the Firefox DevTools interface. The top navigation bar includes tabs for Inspector, Console, Debugger, Style Editor, Performance, Network, and Scratchpad. The Inspector panel is active, showing the DOM tree for the page. The selected element is a `div` with the ID `windows` and the class `app`. The DOM tree shows a hierarchy starting from `html` to `body`, then `theme-settings`, `screen`, and finally `windows`. The `windows` element contains a `before` pseudo-element and several `div` elements representing application windows, such as `CallScreenWindow_0`, `AppWindow_1`, `AppWindow_5`, `AppWindow_6`, `AppWindow_7`, and `AppWindow_8`. The `AppWindow_5` element is currently selected and highlighted in blue. The CSS Rules panel on the right shows the styles for the selected element. It includes a `system` class rule from `system.css:241` with properties like `position: absolute`, `width: 100%`, and `height: 100%`. Below that, it shows the `body` rule from `system.css:15` with `color: #FFF`. The `Inherited from html` section shows the `root` rule from `gaia-theme.css:6` with various color variables like `color-alpha: #333`, `color-beta: #FFF`, and `color-gamma: #4D4D4D`.

Marketplace

- Curated, packaged apps
- Open & Non-proprietary
- Monetization options



Resources

- How to do everything:
https://developer.mozilla.org/Firefox_OS
- Get the code:
<https://github.com/mozilla-b2g/>
- File a bug: <https://bugzilla.mozilla.org/>
- Good first bug?
<http://www.joshmatthews.net/bugsahey/?b2g=1>

Thank You!

