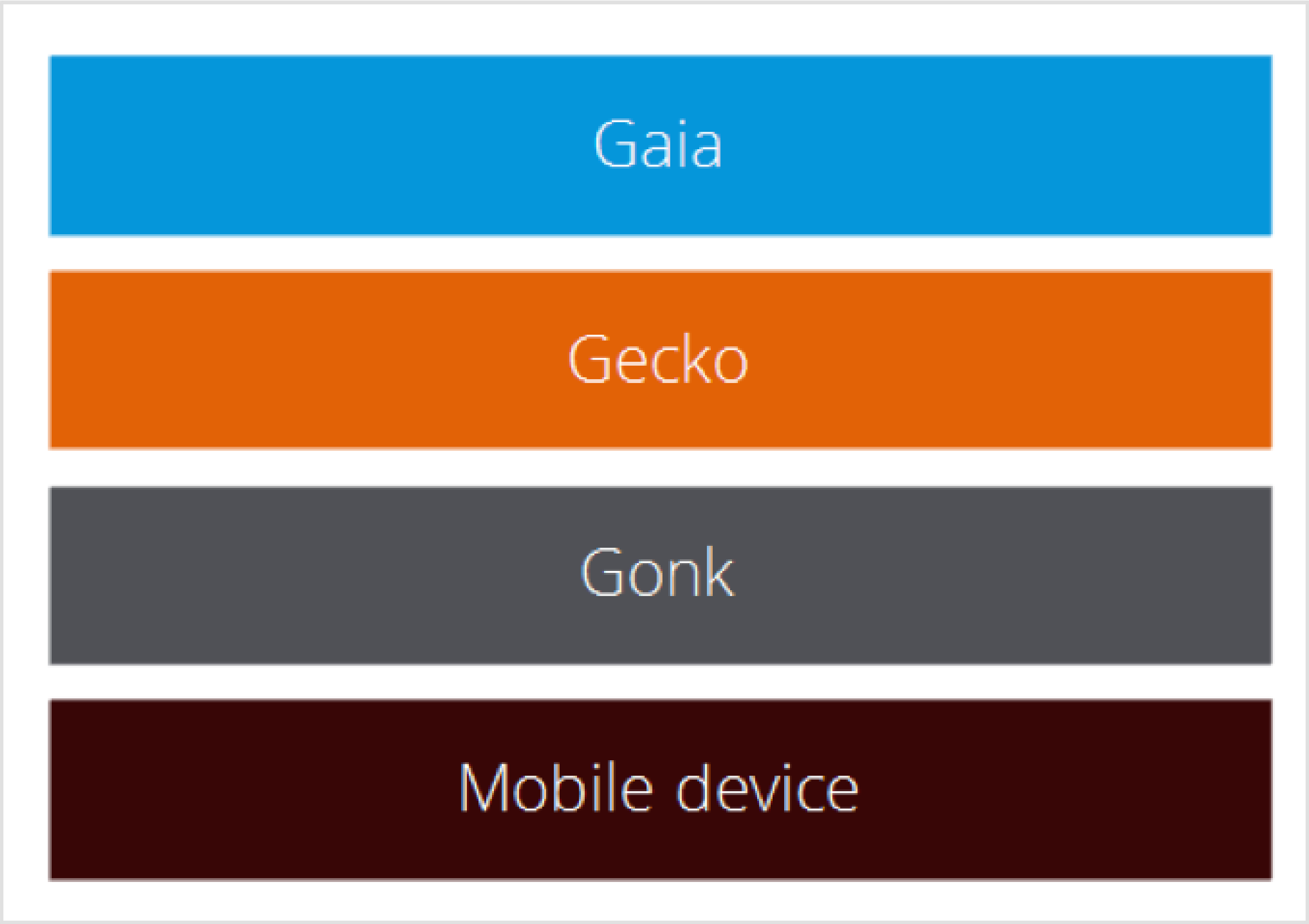
Secure Architecture



Gaia

Gecko

Gonk

Mobile device

Secure System Deployment

- The original system image is created by a known, trusted source – usually the device OEM – that is responsible for assembling, building, testing, and digitally signing the distribution package.

Secure System Updates

- Mozilla recommends and expects that updates are fetched over an SSL connection.
- Strong cryptographic verification is required before installing a firmware package.
- The complete update must be downloaded in a specific and secure location before the update process begins.
- The system must be in a secure state when the update process starts, with no Web apps running.
- The keys must be stored in a secure location on the device

Trusted and Untrusted Apps

- Certified
- Privileged
- Web (everything else)

Hosted Apps

- Located on a web server
- Loaded via HTTP

Goals and scope of the Firefox OS system security model

- Limit and enforce the scope of resources that can be accessed or used by a web application.
- Ensure several layers of security are being correctly used in the operating system.
- Limit and contain the impact of vulnerabilities caused by security bugs, from the Gonk layer.
- Web application permissions and any application related security feature is detailed in the Application Security model.
-
-

Enforcing permissions

- The Firefox OS core process, b2g, has very high privileges and has access to most hardware devices.
- Web applications run in a low-privileged content process and only communicate with the b2g core process using IPC, which is implemented using IPDL.
- The content process has no operating system level access to resources.
- Each Web API has one or more associated IPDL protocol declaration file(s) (*.ipdl)
- Firefox OS content processes can only communicate through the IPDL mechanism back to the core process, which will perform actions on behalf of content.

Risks

- Leak of information when spawning the web application's content process
- Possibility of accessing operating system resources, escalate to the same level of privileges as the b2g process
- Bypassing the content process initialization

Secure system updates Risks

- Compromised update package data, resulting in an untrusted update package being installed
- Compromised update check
    - User does not see new updates are available
    - User gets an out of date package as an update, which effectively downgrades the software on the device
- System state compromised or unknown during the installation of the update; this may (for example) lead to:
    - Missing elements during the installation, some of which may be security fixes
    - Security fixes reverted by the compromised system after upgrade
- Vulnerabilities in the update checking mechanism running on the device
- Lack of updates or tracking for a software component with a known vulnerability

App types

- Web Apps: Most third-party apps will be "Web" Apps, which is the default type, and doesn't grant the App any additional permissions besides those already exposed to the web. Web Apps can be installed from any website, without any further verification, but as a
- Privileged Apps: These Apps are allowed to request increased permissions, and as such Privileged Apps must be verified and signed by a Marketplace
- Certified Apps: Certified Apps can currently only be pre-installed on the device.

App Installation

- Hosted apps: Hosted apps are installed by calling navigator.mozApps.install(manifestURL), where manifestURL is a URL that specifies the location of the app. For further details, see Installing Apps.
- Packaged apps: For packaged apps, the main application manifest is stored inside the package itself, so that it can be signed. There is a second "mini-manifest" that is used to start the installation and update process on the marketplace. See Installing Packaged Apps and Packaged apps for more information.

Assumptions about users

- Data transfer is slow, expensive, and intentionally constrained; in other words, we assume that the user has a slow data connection and a limited amount of traffic permitted each month.
- We assume that the user has little or no access to WiFi; most updates will be performed over their cellular data connection.
- Devices are rarely roaming.
- Users keep their data service disabled by default, enabling it only to complete certain transactions.
- Users keep and use multiple SIM cards.

Design principles

- Updates should minimize impact to the user; don't interrupt the user any more than necessary, don't adversely impact their connection speed, and so forth.
- Don't charge the user to update their apps.
- Minimize the consequences of failed updates.
- Support backward compatibility for users who can't update their apps, or aren't able to update them often.
- Avoid presenting users with unneccessary technical details.