Chromium Operating System

Karol Kański

21 stycznia 2010

Plan prezentacji

Kalendarium

- 7 lipca 2009 - prezentacja Google Chrome OS
- 19 listopada 2009 - Google ogłasza projekt Chromium OS
- druga połowa 2010 - pierwsze netbooki z Chrome OS

Google Chrome OS vs. Chromium OS

Google Chrome OS to system rozwijany w firmie Google, który będzie dostępny wraz z dedykowanymi urządzeniami w drugiej połowie 2010 roku. Chromium OS to projekt open source, do którego Google wniósł kod GC OS. Wiele cech Chrome OS będzie niedostępnych w Chromium OS (autoupdate, weryfikowane bootowanie). Google prosi, żeby swoje dystrybucje nazywać Chromium OS, a nie Google Chrome OS.

Motywacja wg. Google

Prezentacja Google Chrome OS, 7 lipca 2009

However, the operating systems that browsers run on were designed in an era where there was no web. So today, we're announcing a new project that's a natural extension of Google Chrome — the Google Chrome Operating System. It's our attempt to re-think what operating systems should be.

Wymagania użytkowników

Do Google docierały potrzeby użytkowników:

- chcą odbierać maile natychmiast
- chcą, by ich komputery działały zawsze tak szybko, jak wtedy, gdy je kupili
- chcą mieć swoje dane dostępne z każdego miejsca na Ziemi
- nie chcą więcej martwić się o utratę danych i ich backupy
- nie chcą więcej męczyć się z konfigurowaniem systemu- on ma po prostu działać

Inne powody

Powody, którymi Google mniej się chwali

- spożytkowanie sukcesu Chrome
- ale też zapewnić mu dalsze powodzenie
- pokazać Microsoftowi, kto jest najważniejszą firmą IT

Cele projektu

Stworzenie systemu opracyjnego:

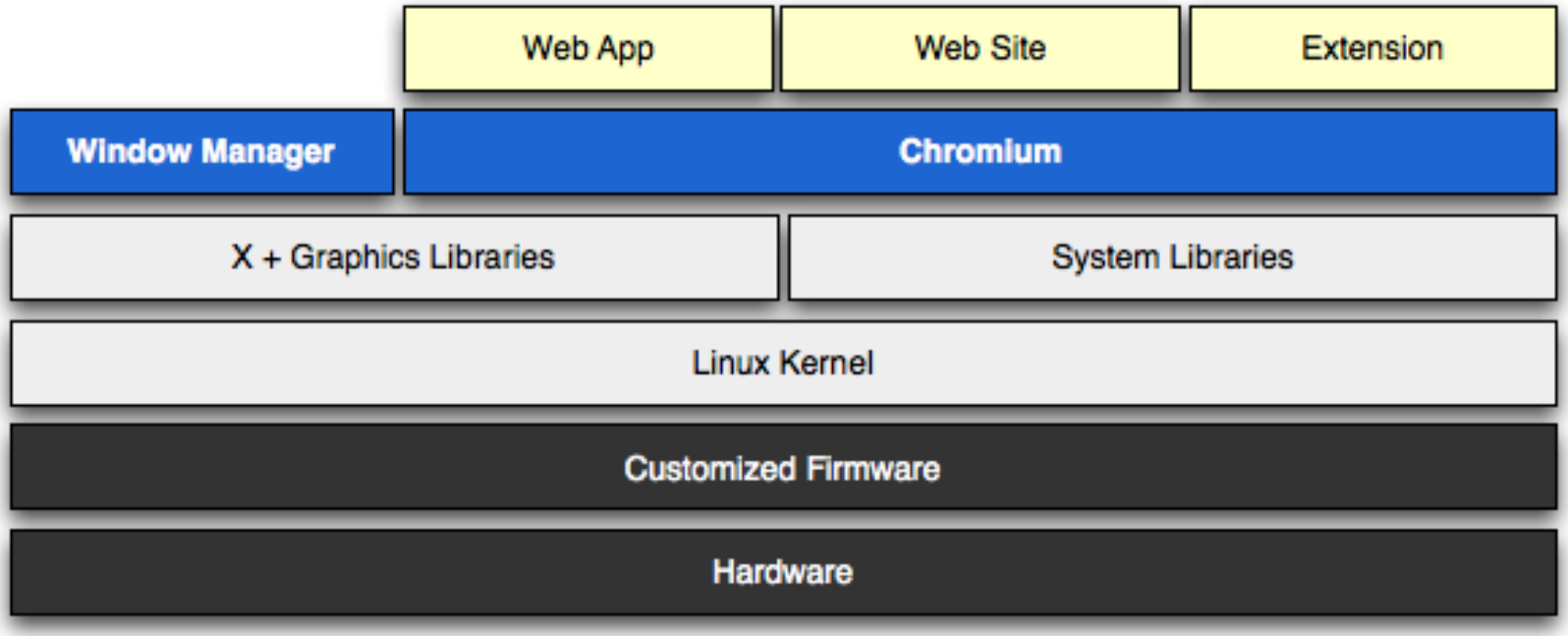
- bezpiecznego
- lekkiego i szybkiego
- prostego w obsłudze

Architektura ogólnie

W architekturze systemu można wyróżnić trzy główne komponenty:

- Firmware
- Jądro i usługi z przestrzeni użytkownika
- Przeglądarka oparta o Chromium i menadżer okien

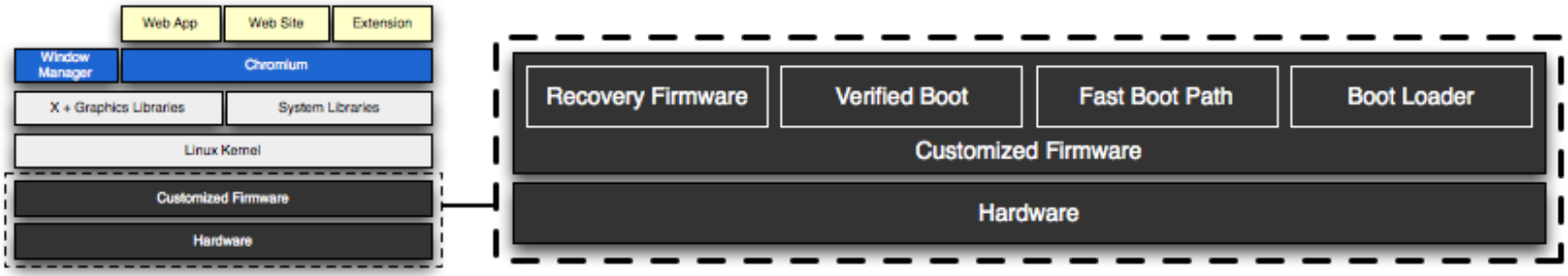
Architektura ogólnie



Firmware

- odgrywa kluczową rolę w tym, że system bootuje się szybciej i bezpieczniej
- wyrzucone niepotrzebne skladniki (nie trzeba być kompatybilnym wstecz)
- wsparcie dla odzyskiwania systemu

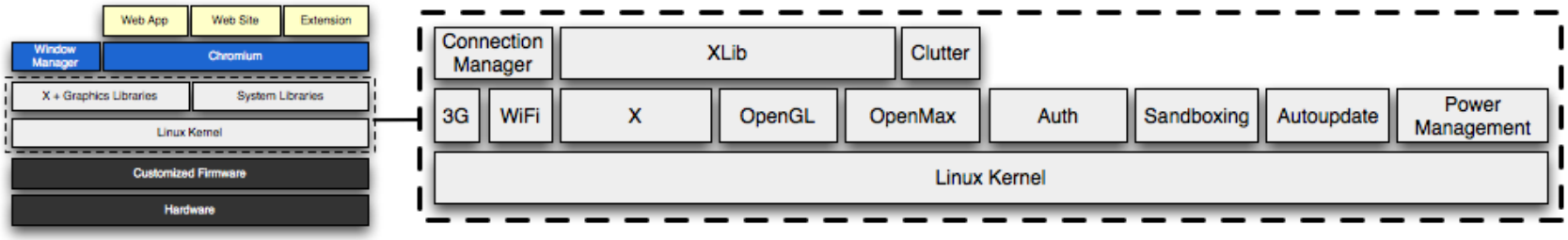
Firmware


Web App
Web Site
Extension
Window Manager
Chromium
X + Graphics Libraries
System Libraries
Linux Kernel
Customized Firmware
Hardware
Recovery Firmware
Verified Boot
Fast Boot Path
Boot Loader
Customized Firmware
Hardware

Jądro i usługi przestrzeni użytkownika

- zwykłe jądro Linuxa (2.6.30)
- kilka latek poprawiających wydajność
- odchudzenie procesu init (uruchamianie tylko krytycznych usług)
- procesy zarządzane przez Upstart

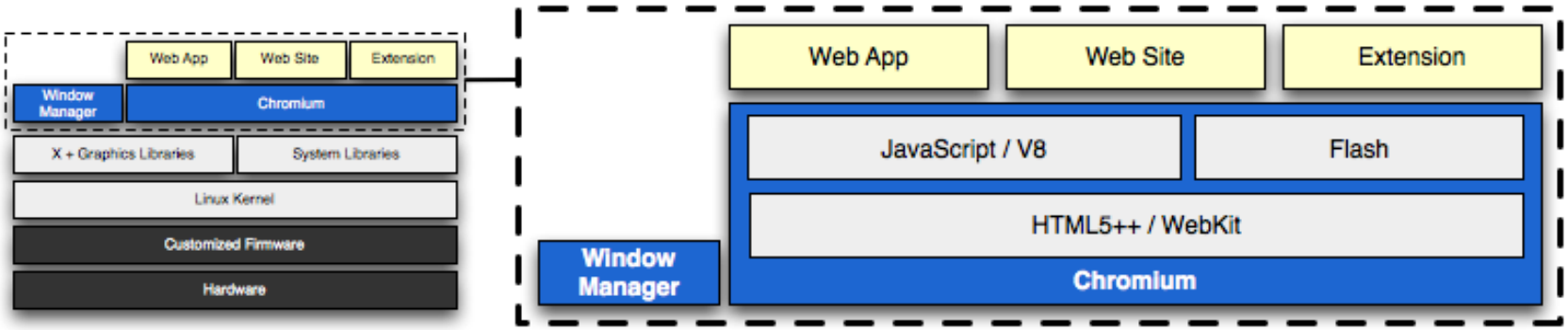
Jądro i usługi przestrzeni użytkownika



Chromium i menadżer okien

- standardowy menadżer okien
- przeglądarka oparta o Chromium
- tylko aplikacje webowe (działające wewnątrz przeglądarki)

Chromium i menadżer okien



Bezpieczeństwo ogólnie

Cztery główne myśli:

- Idealne jest wrogiem dobrego.
- Broń sie jak najgłębiej
- Niech urządzenia będą bezpieczne z definicji
- Nie obwiniaj użytkownika

Model zagrożeń

Typy przeciwników:

- przypadkowy
- specjalistyczny

W obecnej wersji główna uwaga skupia się na tym pierwszym.
Typy ataków:

- zdalny
- kradzież urządzenia

Główne pomysły

Chromium OS projektowano mając na względzie głównie bezpieczeństwo. Do najważniejszych pomysłów zapewniania bezpieczeństwa należą:

- utwardzanie systemu
- modularyzacja przeglądarki
- bezpieczne aktualizacje
- wsparcie firmwaru (weryfikowane bootowanie)
- zarządzanie kontami
- szyfrowanie danych (kradzież urządzenia)

Utwardzanie systemu

Bardzo duży nacisk położono na poprawę poziomu bezpieczeństwa samego systemu. W ostatnich latach działania podnoszące poziomo bezpieczeństwa w środowisk Linuksowych kręcą się wokół dwóch tematów:

- zasady najmniejszych przywilejów
- łagodzenia skutków ataków

Główne idee

Działania poprawiające bezpieczeństwo systemu:

- piaskownice (sandboxing)
- utwardzanie łańcucha narzędziowego (toolchain)
- utwardzanie jądra
- restrykcje nałożone na system plików

Technologie

Jądro Linuksa daje wiele możliwości pozwalających na realizację powyższych celów:

- cgroups
- namespacing
- dzielenie przywilejów root-a przy użyciu capabilities
- Linux Security Modules
- grsecurity

Plan wprowadzania zabezpieczeń

Wprowadzanie kolejnych poziomów zabezpieczeń do systemu będzie podzielone na trzy fazy

1. Zamiany na powierzchni.
2. Schodzimy głębiej.
3. Nie zapomnij fajki!

Zmiany na powierzchni

- firewall
- usługi i capabilities
- MAC

Schodzimy głębiej

- cgroupsd
- przeglądarki we wlasnych przestrzeniach nazw
- root nie jest właścicielem wszystkich plików
- używanie urządzeń - warstwa pośrednicząca
- monitorowanie

Nie zapomnij fajki

- /sbin/init - całkowite usunięcie roota
- sterowniki urządzeń
- utwardzenie zarządzania stosem jądra
- izolacja danych i procesów użytkownika per domena

Ochrona cachowanych danych użytkownika

Chcemy:

- trzymać pewną ilość maili, obrazków itp. na dysku, żeby komputer nie był bezużyteczny, kiedy nie ma połączenia z siecią
- współdzielić komputer
- i przy tym nie martwić się o bezpieczeństwo tych danych

Rozwiązanie:

- szyfrowanie

Szczegóły (1)

- zaszyfrowany katalogu home dla każdego użytkownika
- przy pierwszym logowaniu użytkownik dostaje zaszyfrowany obraz w ukrytym katalogu
- po zalogowaniu obraz jest odszyfrowywany i przeznaczony do użytku
- po wylogowaniu/rebootowaniu obraz jest znowu szyfrowany, czasem też kompresowany

Szczegóły (2)

Konkretniej:

- obraz przyłączony do systemu za pomocą pseudourządzenia loop device
- za pomocą device mapper subsystem nakładamy na to wirtualne urządzenie szyfrujące dm-crypt
- dm-crypt formowatowany w ext4 i kopiowany do katalogu home

Wydajność

Konkretniej:

- tworzenie nowego obraz- tanie, ale formatowanie systemu plików- drogie
- 144 GB to 2 minuty, nawet 1 GB to około 6 sekund
- z pomocą przychodzi ext4 i leniwa inicjalizacja inodów - 1GB to sekunda, 128 MB dużo mniej
- problemem jest zwiększanie obrazu - można to robić online, ale wtedy dysk będzie bezużyteczny- rozwiązaniem dodanie leniwej inicjalizacji inodów przy operacji resize

Bootowanie firmwaru i odzyskiwanie systemu

Firmware ma odgrywać olbrzymią rolę w tym, że system będzie się bootował szybko i bezpiecznie.
Podjęto decyzję, że ma mieć następujące cechy:

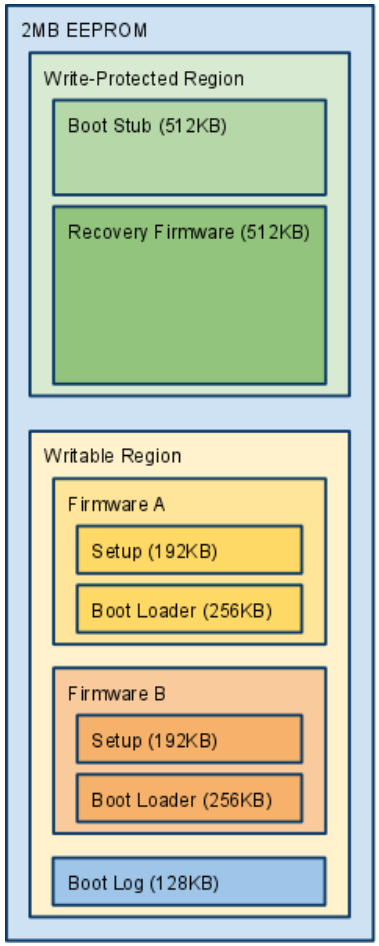
- bootowanie ma startować bezpiecznie
- zapisywalny firmware musi mieć backup
- firmware odzyskiwania systemu ma być tylko do odczytu
- firmware odzyskiwania systemu nie może korzystać z sieci
- firmware odzyskiwania systemu ma powiedzieć użytkownikowi, jak ten system odzyskać
- firmware odzyskiwania systemu może być wywołany przez użytkownika
- deweloperzy mają mieć możliwość instalowania własnego oprogramowania

Podział EEPROM

Pamięć EEPROM jest podzielona na następujące bloki:

- boot stub
- firmware A/B setup
- firmware A/B bootloader
- boot log
- recovery firmware

Mapa EEPROM


2MB EEPROM
Write-Protected Region
Boot Stub (512KB)
Recovery Firmware (512KB)
Writable Region
Firmware A
Setup (192KB)
Boot Loader (256KB)
Firmware B
Setup (192KB)
Boot Loader (256KB)
Boot Log (128KB)

Weryfikowane bootowanie

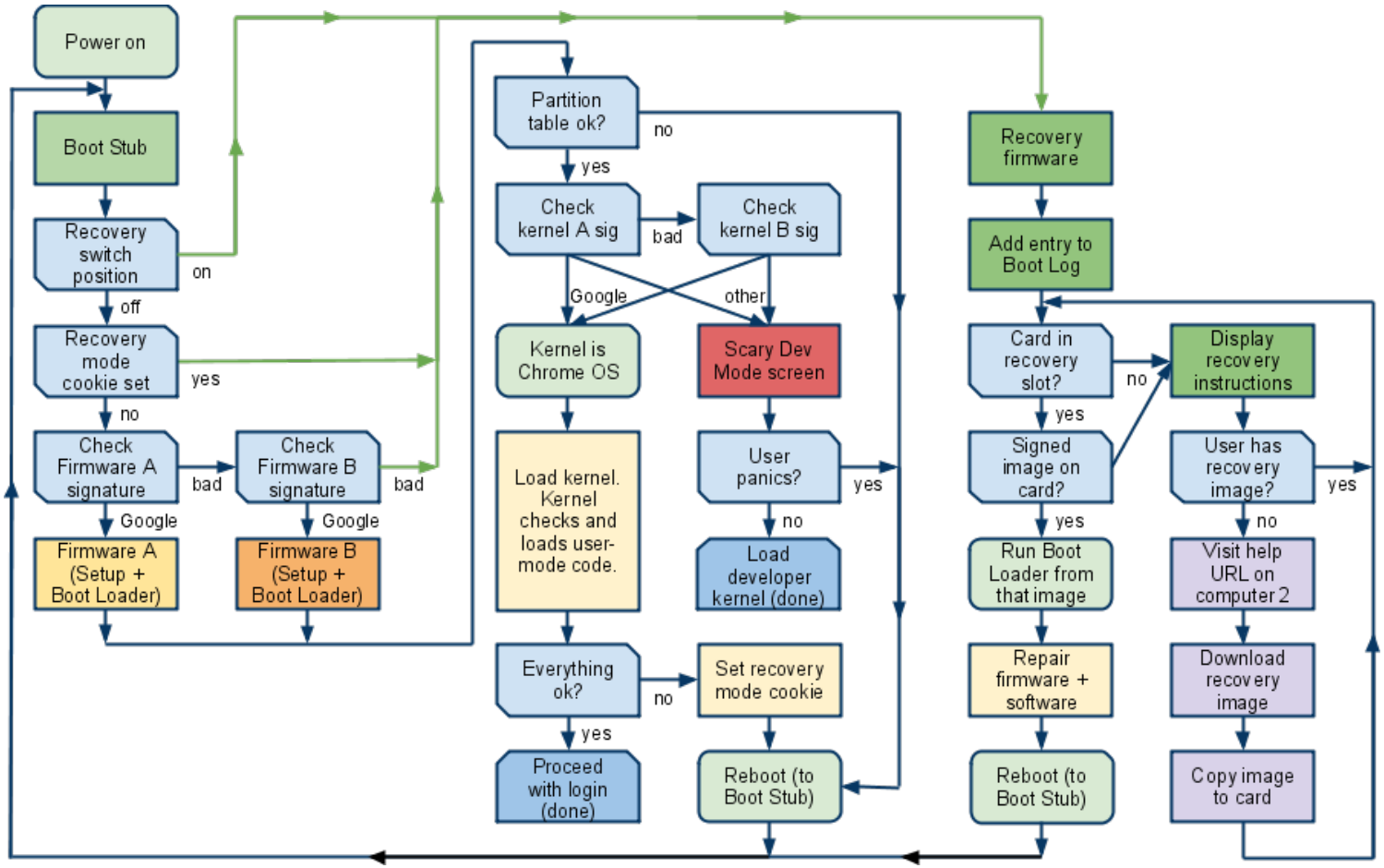
Kolejna technika mająca pomóc w zwiększeniu bezpieczeństwa. Zaczynając od kawałka kodu tylko do odczytu każdy kolejny krok na ścieżce bootowania jest weryfikowany przez poprzedni. Ma to zapewnić, że wykonywany kod pochodzi od Chromium OS.

Weryfikowane bootowanie

Kolejne kroki wyglądają następująco:

- boot stub weryfikuje firmware
- następnie ten firmware weryfikuje kolejny wykonywany kod i tak aż do jądra
- weryfikacja całej partycji root dodałaby ok 5 sekund do procesu bootowania- to zdecydowanie za dużo
- weryfikowanie bloków dyskowych będzie się odbywało przy dostępie do nich

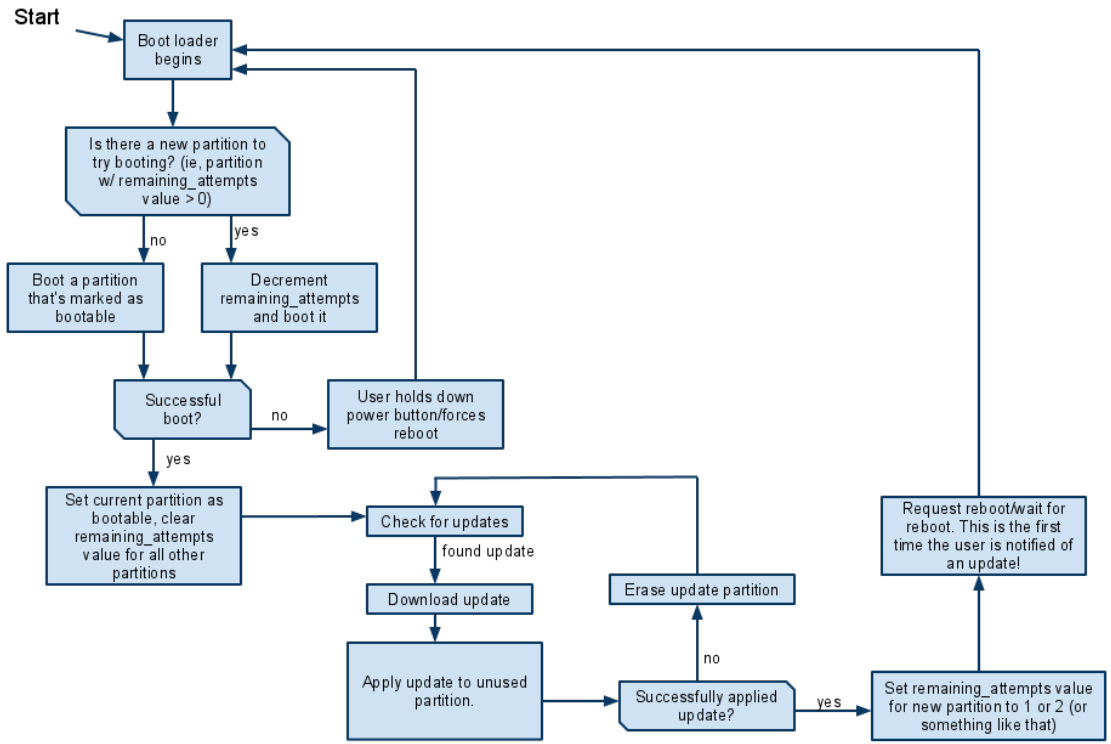
Proces bootowania



Aktualizacje systemu

- dwie partycje root, jedna partycja na dane, opcjonalnie jedna swap
- aktualizacje działają w tle
- aktualizowana jest partycja, która jest w danym momencie nieużywana
- po restarcie następuje zamiana partycji

Proces aktualizacji



Logowanie

Logowanie:

- jeśli offline to lokalnie(trzeba było już wcześniej przynajmniej raz być zalogowanym), jeśli online to do Google
- SSO dla usług Google
- auto-logowanie

W planach:

- SSO dla serwisów polegających na OpenID

Zarządzanie kontami

- jeden właściciel urządzenia
- lista użytkowników, którzy mogą korzystać z urządzenia
- tryby: Incognito, Promiscous

Zalety/ wady

Zalety:

- duży nacisk na bezpieczeństwo
- prostota obsługi
- szybki (na pokazie bootowanie w 7 sekund)

Wady:

- pełnia możliwości tylko na dedykowanym sprzęcie
- nic nowego
- brak wielu możliwości zwykłego OS

Pytania

Pytania?