

# Google Chrome OS

Christopher Maurer

13. Februar 2012

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>1</b>
<b>1 Entwicklungsstand und Verwendungszweck</b>	<b>2</b>
1.1 Erste Ankündigung 2009 . . . . .	2
1.2 Entwicklungsstand 2009 - 2011 . . . . .	2
1.2.1 Die Vorabversion <i>Chromium OS</i> . . . . .	2
1.2.2 <i>Chrome OS</i> als genutzte Version . . . . .	3
1.3 Möglicher Einsatzbereich . . . . .	3
1.4 Speziell verwendete Hardware . . . . .	3
1.4.1 Der Prototyp <i>Cr-48</i> . . . . .	3
1.4.2 <i>Chromebooks</i> als verwendete Hardware . . . . .	4
1.5 Stand Anfang 2012 . . . . .	4
<b>2 Architektur</b>	<b>5</b>
2.1 Angepasste Firmware . . . . .	5
2.1.1 <i>Recovery Firmware</i> . . . . .	5
2.1.2 <i>Verified Boot</i> . . . . .	6
2.2 Linux-Kernel . . . . .	6
2.3 Benutzeroberfläche . . . . .	7
2.3.1 Login . . . . .	7
2.3.2 <i>Chromium</i> . . . . .	8
2.4 Weitere Eigenschaften . . . . .	9
2.4.1 Arbeitsspeicherverwaltung . . . . .	9
2.4.2 <i>Google Cloud Print</i> . . . . .	9
2.4.3 Shell . . . . .	10
<b>3 Nutzen</b>	<b>11</b>
3.1 Vorteile . . . . .	11
3.1.1 Das Konzept von <i>Chrome OS</i> . . . . .	11
3.1.2 Leichtgewichtiges, schnelles System . . . . .	11
3.1.3 Sicherheitsmechanismen . . . . .	12
3.2 Nachteile . . . . .	12
3.2.1 Datenschutz . . . . .	12
3.2.2 Systemzwänge . . . . .	12
<b>Literatur</b>	<b>13</b>
<b>Abbildungsverzeichnis</b>	<b>14</b>

# 1 Entwicklungsstand und Verwendungszweck

## 1.1 Erste Ankündigung 2009

»It's been an exciting nine months since we launched the Google Chrome browser. Already, over 30 million people use it regularly. We designed Google Chrome for people who live on the web — searching for information, checking email, catching up on the news, shopping or just staying in touch with friends. 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.[...]« [1]

Mit diesen Worten wurde das Betriebssystem *Google Chrome OS* am 7.Juli 2009 seitens Google auf dem hauseigenen Blog angekündigt. Es soll damit die Lücke zwischen dem bereits veröffentlichten Webbrowser *Google Chrome* und dem 2009 bereits erfolgreichen Betriebssystem für Smartphones *Android* schließen. *Chrome OS* soll dabei eine natürliche Weiterentwicklung des Webbrowsers *Chrome* sein, und soll somit auch die grundlegende Sicht auf die Betriebssysteme verändern. Schließlich seien die bisherigen, etablierten Betriebssysteme alle zu einer Zeit entwickelt worden, als es kein Internet gab. Laut der Ankündigung sollte der Sourcecode bereits Ende 2009 offen vorliegen; damit folgt *Chrome OS* dem Betriebssystem für Smartphones und Tablets *Android* als Open-Source-Betriebssystem. Mitte 2010 sollten schon die ersten Endgeräte mit *Chrome OS* auf dem Markt erhältlich sein [2].

## 1.2 Entwicklungsstand 2009 - 2011

### 1.2.1 Die Vorabversion *Chromium OS*

Mit der Veröffentlichung des Quellcodes einer Vorabversion von *Chrome OS*, die Google als *Chromium OS* [3] bezeichnet, wurde das Betriebssystem der Öffentlichkeit zugänglich gemacht. Dabei wird erst das fertige Betriebssystem als *Chrome OS* angesehen, die Vorabversion, die für Entwickler und Interessierte zugänglich ist, wird als *Chromium OS* bezeichnet. Hier wendet Google die gleiche Nomenklatur wie beim entwickelten Webbrowser *Chrome* an, der als Open-Source-Variante *Chromium* erhältlich ist. Google stellt den Sourcecode, Anleitungen sowie eine Übersicht des Systems auf [www.chromium.org](http://www.chromium.org) zur Verfügung.

*Chromium OS* ist seit der Veröffentlichung von *Chrome OS* weiterhin erhältlich, und lässt sich im Gegensatz *Chrome OS* auf beliebiger Hardware installieren; es sind auch virtuelle Images für *Virtual Box* und *VMWare* verfügbar<sup>1</sup>.

---

<sup>1</sup>*Chrome OS* builds by Hexxeh - <http://chromeos.hexxeh.net/>

### 1.2.2 *Chrome OS* als genutzte Version

Auch wenn Google den Quellcode von *Chromium OS* veröffentlicht hat, so bleibt letztlich *Chrome OS* das Betriebssystem, das auch auf der entsprechenden Hardware, den *Chromebooks*, verwendet wird. Der Entwicklungsprozess ist ähnlich wie bei *Android*, wo Änderungen in den veröffentlichten Quellcode einfließen, und dann als finale Version veröffentlicht wird. *Chrome OS* ist damit das Ergebnis der Open-Source-Arbeit mit *Chromium OS*.

### 1.3 Möglicher Einsatzbereich

Der mögliche Einsatzbereich wird von den Entwicklern schon 2009 umrissen, *Chrome OS* soll nach Möglichkeit unabhängig von der Hardwarearchitektur sein; so sollen neben den bereits etablierten Prozessoren der x86-Architektur auch die ARM-Architektur unterstützt werden. Der Unterschied zu *Android* ist, dass nicht vorgesehen ist, *Chrome OS* als Betriebssystem ohne Bindung an eine spezielle Hardware zu nutzen. Der Einsatzbereich von *Chrome OS* liegt (bei heutiger Betrachtung) im Bereich der sog. *Netbooks*, also besonders flache und kleine Notebooks. Google selbst bezeichnet seine Hardware als *Chromebooks* [4], die damit in direkter Konkurrenz zu den *Netbooks* stehen. Die anfängliche ARM-Unterstützung ist zumindest aufgeschoben worden, zunächst konzentriert sich Google auf die x86-Architektur. Nach Bekanntgabe der Entwicklung 2009 wurden auf dem offiziellen Blog einige Hersteller benannt, mit denen Google für die Entwicklung zusammenarbeiten wird [5].

Google tritt mit dieser Positionierung einerseits gegen die etablierten Betriebssysteme sowie Herstellern von *Netbooks* an, distanziert sich aber andererseits von seinen eigenen Produkten wie *Android*. Die Ausrichtung auf eine bestimmte Hardware mit speziellen Anforderungen ist aber ein Novum gegenüber der bisherigen Strategien. Das Konzept des cloud-orientierten Betriebssystems, bei dem die Daten in der Cloud gespeichert, und nur bei Benutzerabruf auf die lokale Festplatte kopiert wird, unterscheidet sich stark von den bisher genutzten Betriebssystemen. Google will so die Nutzung des Internets über das eigene Betriebssystem noch weiter vereinfachen, wobei Dateien, Verzeichnisse, etc. wie in klassischen Betriebssystemen keine Rolle mehr spielen sollen [6].

### 1.4 Speziell verwendete Hardware

#### 1.4.1 Der Prototyp *Cr-48*

Am 7. Dezember 2010 gab Google auf einer Pressekonferenz bekannt, dass eine Referenzhardware für die späteren *Chromebooks* erscheint. Nutzen konnten die Hardware vor allem Entwickler in den USA, die sich bei Google für einen Prototyp bewerben konnten; ein Verkauf war nicht vorgesehen. Die Hardware selbst kündigte Google noch als Netbook an, ausgestattet ist es u.a. mit einem 12"-Display, einer SSD-Festplatte, WLAN, einem UMTS-Modem, einer Webcam, sowie einem USB-Anschluss ausgerüstet. Die Tastatur ist für die Bedürfnisse des Webbrowsers angepasst, so fehlen etwa die Funktionstasten,

und wurden durch browserspezifische Tasten ersetzt. Auf dem *Cr-48* war bereits eine Vorabversion von *Chrome OS* installiert.

#### 1.4.2 *Chromebooks* als verwendete Hardware

Seit dem 15. Juni 2011 sind insgesamt *Chromebooks* der Hersteller Acer und Samsung verfügbar [7]; der Preis liegt je nach Ausstattung bei ca. 200 - 400 Euro. Die Ausstattung orientiert sich stark am Prototyp *Cr-48*, wobei die Größe der SSD-Festplatte mittlerweile 16GB beträgt. Ein UMTS-Modem ist nur bei Samsungs teurerem *Chromebook* erhältlich. Alle *Chromebooks* besitzen nur wenige externe Anschlüsse, mit 16GB Speicher eine kleine Festplatte für die Speicherung von Daten während einer Sitzung des Betriebssystems, und kein optisches Laufwerk. Alle Daten werden generell in einer Cloud gespeichert, es ist nicht möglich, Daten auf die Festplatte zu kopieren; so ist eine Internetverbindung für den normalen Betrieb zwingend notwendig. Beispielsweise werden Dokumente nur über *Google Docs* gelesen oder geschrieben; ein Zugriff auf die Daten ist ohne Internetverbindung dann nicht ohne weiteres möglich. Jede zusätzliche Funktionalität werden bei *Chrome OS* als *Apps* realisiert; allerdings sind viele dieser *Apps* nicht offline verfügbar, obwohl sie gerade den Nutzen des Systems ausmachen.

Google bietet die *Chromebooks* auch leihweise an, z.B. für Schulen und andere Einrichtungen zu einem monatlichen Mietpreis.

#### 1.5 Stand Anfang 2012

Über die genauen Verkaufszahlen der *Chrome OS* und der *Chromebooks* schweigen sich Google sowie die Hersteller aus; von genauen Verkaufszahlen spricht die Fachpresse ebenfalls nicht.

Der Hersteller Samsung hat bekannt gegeben, eine neue Ausgabe des *Chromebooks* auf den Markt zu bringen [8]. Die Hardware wird dabei nur minimal verändert. Zusätzlich wird ein neues Set-Top-System namens *Chromebox* vertrieben, das im Prinzip einem Mac Mini ähnelt; die Hardware soll in etwa der des *Chromebooks* entsprechen. Der Verkaufsstart beider Geräte wird für das 2. Quartal 2012 angegeben; bisher ist der Verkauf nur in den USA vorgesehen.

## 2 Architektur

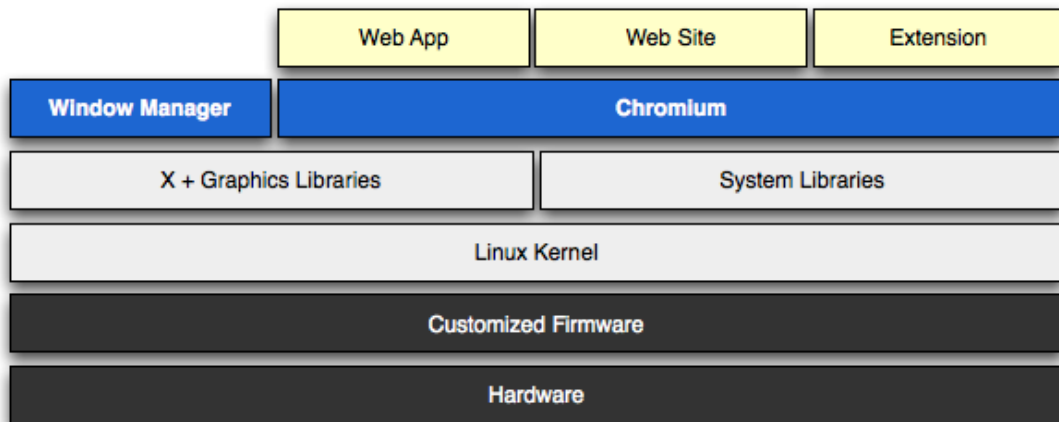


Abbildung 1: Übersicht der Architektur von *Chrome OS*

### 2.1 Angepasste Firmware

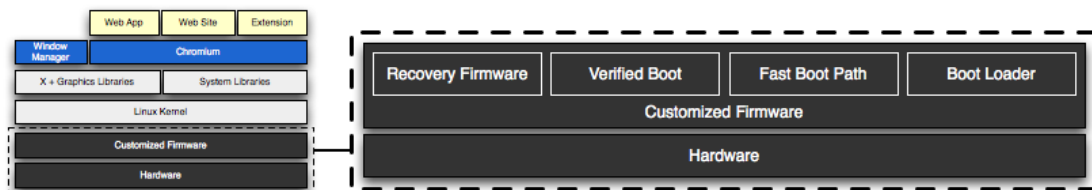


Abbildung 2: Detaillierte Firmware-Struktur *Chrome OS*

Grundlage der Architektur [9] von *Chrome OS* ist eine angepasste Firmware, die zwischen der Hardware und dem Linux-Kernel liegt. Die Firmware ist speziell an die Bedingungen von *Chrome OS* angepasst, und ist demnach viel schlanker als eine gewöhnliche Firmware eines Notebooks oder PCs. So werden Diskettenlaufwerke beim Systemstart nicht geprüft, da sie generell nicht benötigt werden. Zwei der vier Unterabschnitte sind dabei genauer zu betrachten.

#### 2.1.1 *Recovery Firmware*

*Recovery Firmware* [10] ermöglicht es, eine eventuell schädliche Variante der eigenen Firmware schon beim Systemstart zu erfassen und die vorherige Version wiederherzustellen, z.B. nach einem fehlgeschlagenen Firmwareupdate oder einem gezielten Angriff.

Grundsätzlich ist der EEPROM, auf dem die Firmware gespeichert ist, in zwei Abschnitte unterteilt: der erste Abschnitt ist ein schreibbarer Abschnitt, und beinhaltet mehrere Versionen der Firmware (mit einem Backup der aktuellen Version). Diese Firmware kann überschrieben werden, etwa durch ein Update. Der zweite Abschnitt ist schreibgeschützt, und kann nicht verändert werden. Diese Firmware wird in den o.g. Szenarien benutzt, um die offensichtlich beschädigte Firmware zu überschreiben. Technisch wird der Schreibschutz mit einem externen Pin des EEPROM realisiert, wird der Pegel des Pin auf 0 gesetzt, so existiert ein Schreibschutz für die höheren Ebenen.

Es können mit dieser Funktion ebenfalls Fehler in der Software beseitigt werden, die u.U. eine schädliche Wirkung haben könnten. So können etwa fehlerhafte Softwareupdates, und fehlerhafte Software, die den Systemstart beeinflussen kann, beim Bootvorgang beseitigt werden; potenzielle Angriffe, wie etwa Schreibzugang auf das root-Verzeichnis werden ebenfalls vom System beim Bootvorgang erkannt. So kann das komplette Betriebssystem notfalls auf den vorherigen Stand zurückgesetzt werden, falls die Überprüfung beim Bootvorgang fehlschlägt.

### 2.1.2 *Verified Boot*

Die Aspekte der *Recovery Firmware* dienen als Voraussetzung, damit Entwickler auch trotz unterschiedlicher Firmwareversionen bei der Nutzung von *Chrome OS* die gleichen Voraussetzungen haben wie die reinen Nutzer. *Verified Boot* [11] soll sicherstellen, dass der persistente Speicher der Firmware nicht verändert wurde; besonders der Aufruf des Kernels liegt im Fokus der Entwickler. Zuerst generiert die *read-only-Firmware* einen Hashwert, der Informationen über die veränderbare *read-write-Firmware* enthält. Aus diesem Schlüssel wird dann mittels RSA-Verschlüsselung ein dauerhafter *public key* erzeugt. Die *read-write-Firmware* generiert jeweils Hashwerte für die einzelnen Kernel-Module und den persistenten Speicher. Diese Hashwerte enthalten selbst einen sog. *Subkey* und eine Liste der einzelnen Kernel-Module, wie `kernel`, `initrd`, etc. Die Hashwerte werden mit dem *Subkey* verrechnet, so dass nicht der Schlüssel der *read-only-Firmware* verwendet wird. Damit kann auf diesen Schlüssel kein Zugriff erfolgen, z.B. muss dieser Schlüssel nicht für Update-Funktionen verwendet werden, was sonst einen Angriff erleichtern könnte.

Entscheidend ist der Schritt in Richtung des Kernels - hat der Bootvorgang diesen erreicht, so ist der *Verified Boot*-Vorgang vollzogen, und *Chrome OS* kann sicher gestartet werden.

## 2.2 Linux-Kernel

Ähnlich wie *Android* basiert auch *Chrome OS* auf verschiedenen vorhandenen Entwicklungen von Linux, *Chrome OS* nutzt hier den Linux-Kernel 2.6.32 als Kernel. Als Basis dient der Ubuntu-Kernel, der aber in vielen Bereichen angepasst wurde; es werden bisher nur vorhandene Module innerhalb des Kernels unterstützt, und keine neuen Module zugelassen. Weiterhin wird der Befehl `initrd` bzw. `initramfs`, also das Laden der Bootvorgänge aus dem RAM nicht unterstützt. Auf einen Swap-Speicher zur

Auslagerung verzichten die Entwickler, auch wenn die Auswirkungen gerade in Hinblick auf den SSD-Speicher positiv sind. So könnte auch die Nutzbarkeit des Browsers erhöht werden, denn ohne Swap ist die Performance allein von der Größe des RAM abhängig. Eine Aktivierung des Swap ist aber für die Zukunft vorgesehen; auch wird der Kernel momentan nur auf x86-Architektur entwickelt, die ARM- und x64-Unterstützung sind in Planung.

Auf den Kernel folgen beim Bootvorgang die System- und Benutzerprozesse, die durch den Daemon *Upstart* verwaltet werden. So ist es möglich, Prozesse parallel, oder bei Störungen neu zu laden. Dies beschleunigt den gesamten Bootvorgang.

## 2.3 Benutzeroberfläche

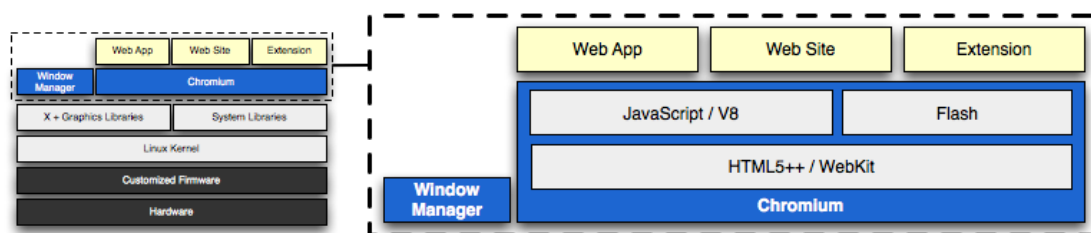


Abbildung 3: Detaillierte Struktur der Benutzeroberfläche von *Chrome OS*

Die Basis der *Window Manager* bildet ein *X-Window-Manager*, der allerdings stark modifiziert wurde, und leichtgewichtig ist. So werden keine Fensterpositionen mehr ermittelt, da der Browser generell nicht geschlossen wird und immer im Vollbildmodus dargestellt wird. Als zentrales und einziges Fenster tritt der Browser (hier: *Chromium*) in Erscheinung.

### 2.3.1 Login

Beim Login [12] möchte Google sicherstellen, dass auch Nutzer ohne einen Google-Account mit einem Gastaccount das Betriebssystem voll nutzen können. Für Nutzer mit einem Account ist der Login wichtig, damit die zuletzt genutzten Einstellungen geladen werden können. Hier soll vor allem die Identifikation mit dem Google-Servern gesichert werden, so dass ein sicherer *Single Sign-On* auf den Cloud-Ressourcen erfolgen kann. Der Browser benutzt zur Authentifizierung die eigenen Implementierungen der Google Account-HTTP(s)-API. Erfolgt die Anmeldung online, wird ein Hash des Passwortes und der *Trusted Platform Module (TPM)*<sup>2</sup> zu einem *magic string* gebildet. Bei einem Offline-Login wird der Benutzer erneut nach dem Passwort gefragt, um damit den *magic string* aufzulösen. Um diesen Prozess zu beeinflussen, müsste ein Angreifer sowohl das Passwort kennen, wie auch den im Gerät verbauten TPM-Chip manipulieren.

<sup>2</sup>[http://de.wikipedia.org/wiki/Trusted\\_Platform\\_Module](http://de.wikipedia.org/wiki/Trusted_Platform_Module)



Der *Single Sign-On* bewirkt, dass alle relevanten Dienste der Cloud mit *einer* Anmeldung gestartet werden. So entfallen weitere Programmstarts, wie etwa dem E-Mail-Clienten.

### 2.3.2 *Chromium*

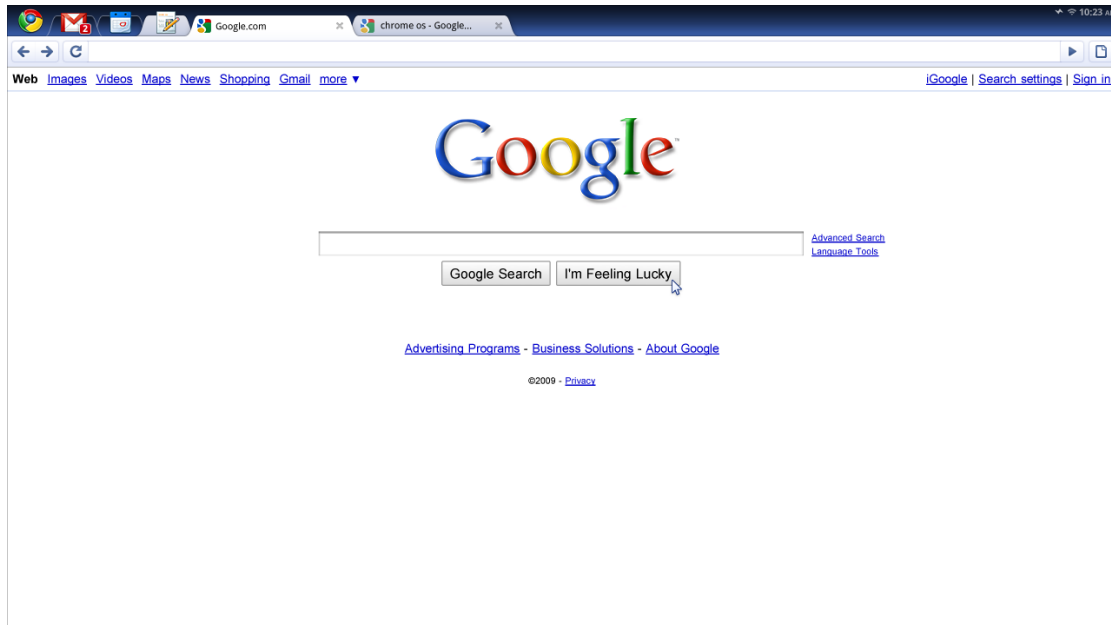


Abbildung 4: Browser *Chrome*

Google greift bereits auf die eigenen Erfahrungen des im Dezember 2008 erschienenen Browsers *Chrome* zurück; so fließen die Entwicklungen [13] in das Betriebssystem ein. Einige Veränderungen sind vor allem mit Blick auf die genutzten *Chromebooks* wichtig. Die obere Tableiste wurde weiter verkleinert, so dass der 12" kleine Bildschirm voll ausgenutzt werden kann. Die Menüleiste wurde zu einem einzigen Icon zusammengefasst, das ein Menü mit allen systemrelevanten Einstellungen bietet. Besonderes Augenmerk legen die Entwickler dabei auf die Integration von *Apps* in den Browser. Die *Apps* werden nach der kurzen Installation als Shortcuts auf den Desktop (es ist ein initialer Start-Tab von *Chrome*) abgelegt und wirken zunächst wie externe Programme. Sie öffnen sich aber nicht in einem neuen externen Fenster, sondern wieder als neuer Tab im Browserfenster. Weiterhin ist die Tableiste verschiebbar, so dass sie als Taskleiste aus bekannten Betriebssystemen fungieren kann. Zu den technischen Neuerungen zählen die Verwendung von HTML 5, so können die *Apps*, die meist online genutzt werden, auch offline weiter genutzt werden. Auch steigt dank HTML 5 die Performance des Browsers, der Berechnungen im Hintergrund ausführt, bevor eine Website geladen wird.

## 2.4 Weitere Eigenschaften

### 2.4.1 Arbeitsspeicherverwaltung

Bei *Chrome OS* wird zunächst auf einen swap-Auslagerungsspeicher verzichtet. Damit ist nicht mehr gewährleistet, dass bei intensiver RAM-Nutzung das Betriebssystem stabil bleibt. Um dies zu gewährleisten, überprüft der *Chrome* die Arbeitsspeichernutzung und ruft bei Bedarf einen zweistufigen Mechanismus im Kernel auf. Zuerst reserviert der Kernel vor dem sog. *soft wall* einen Speicherbereich von etwa 10 MB. Die geöffneten Tabs werden in folgende Prioritätsreihenfolge gebracht:

1. ungeöffnete Tabs
2. geöffnete Tabs ohne Benutzerinteraktion
3. geöffnete Tabs mit Benutzerinteraktion der letzten 10 Minuten bzw. nach Intensität der Speichernutzung
4. geöffnete Tabs im Vordergrund

Dabei wird die Liste von oben nach unten bearbeitet, so dass die Speicherbeschaffung möglichst wenig den Nutzer beeinflusst.

In der zweiten Stufe, dem sog. *hard wall*, wird der OOM-Prozess des Kernels aufgerufen, der den Speicher (nach Gesichtspunkten des Kernels) möglichst effizient freigibt. Dabei werden die vom *soft wall* eingeteilten Prioritäten nicht beachtet. In dem vom OOM-Prozess ausgewählten Tab von *Chrome* findet der Benutzer einen entsprechenden Hinweis, dass der Tab geschlossen wurde.

### 2.4.2 Google Cloud Print

Google nutzt einen eigenen Service, um das Drucken aus der Cloud zu ermöglichen. Dabei muss der Nutzer sein Google-Konto mit einem entsprechenden Drucker verknüpfen, damit der Drucker für den Dienst freigeschaltet wird. Dies kann ein lokaler Drucker sein, der an einen PC angeschlossen ist, oder ein speziell geeigneter *Cloud Ready Printer*, also ein Drucker, der für genau diesen Service eine geeignete Schnittstelle besitzt.

Für die Nutzung eines lokalen Druckers muss dieser an einen PC mit Internetverbindung angeschlossen sein; beide Geräte müssen eingeschaltet sein. Für den lokalen Drucker stellt Google einen Proxy zur Verfügung, der beim Hochfahren des jeweiligen Betriebssystems die Druckaufträge an den verbundenen Drucker weiterleitet. So ist von jedem mobilen Endgerät, auch von einem Smartphone, der Drucker sichtbar, und nimmt Druckaufträge entgegen. Das verwendete Protokoll *Extensible Messaging and Presence Protocol* (XMPP) folgt dem weit verbreiteten XML-Standard, und stammt ursprünglich vom Messaging-Clients Jabber. Bei einem *Cloud Ready Printer* wird der Drucker direkt mit Druckaufträgen versorgt, dies kann über eine eigene E-Mail-Adresse funktionieren (bei HP).

Für *Chrome OS* ist der Dienst essentiell, denn *Cloud Print* ersetzt hier die normale Druckfunktion des Betriebssystems; bisher funktioniert nur das Drucken von Text und

Tabellen mit *Google Docs* reibungslos. Die Zusammenarbeit mit den Druckerherstellern ist demnach sehr wichtig, damit Google diese Funktion erweitern kann; so ist der Erfolg des Dienstes abhängig von der Verfügbarkeit von Cloud-Druckern. Mittlerweile (Stand Januar 2012) haben HP, Kodak und Epson entsprechende Geräte auf den Markt gebracht.

### 2.4.3 Shell

*Chrome OS* besitzt, da es auf Linux basiert, eine Shell; sie ist aber stark in ihrer Funktionalität eingeschränkt. Die Konsole ist über die grafische Oberfläche nur durch Drücken von **STRG+ALT+T** erreichbar, die Benutzung erfolgt über insgesamt 3 Benutzerebenen, wobei die erste Ebene über keinerlei bash-Befehle verfügt. Hier werden nur einfache ssh-Funktionen bereitgestellt. Mit dem Kommando **shell** erreicht man die zweite Ebene, die dann als wirkliche shell einige Funktionen bereitstellt. So lassen sich einige der Standardbefehle wie **sudo**, **vim** etc. nutzen, andere Befehle wie etwa **chmod** fehlen aber ganz, da generell das gesamte Dateisystem nur lesbar sein soll. In der dritten Ebene arbeitet man wie gewohnt mit root-Rechten in der shell.

## 3 Nutzen

### 3.1 Vorteile

#### 3.1.1 Das Konzept von *Chrome OS*

*Chrome OS* basiert auf einem Konzept, dass das Endgerät ein sog. *Thin Client* ist. Anwendungen werden so nicht auf dem Endgerät gespeichert, sondern über ein vorhandenes Netzwerk abgerufen. An der Umsetzung dieses Konzepts, dass fast 30 Jahre alt ist, scheiterte man zunächst am Netzwerk. Mit dem Internet steht aber ein leistungsfähiges Netzwerk zur Verfügung, und auch die Endgeräte besitzen genug Leistung. Dieses Konzept wird aber laut Google zunächst nur für die *Chromebooks* gelten, da Smartphones noch nicht leistungsfähig genug sind.

Das Konzept bricht mit der traditionellen Arbeitsweise eines Betriebssystems, da bei *Chrome OS* kein Wert auf generell sichtbare Datenverwaltung gelegt wird. So spielt lediglich der Browser die übergeordnete Rolle, während die Administration des Betriebssystems entfällt. So werden Dateien und Verzeichnisse zur Nebensache, da sie in der Cloud gespeichert sind, und auf Abruf auf die Festplatte kopiert werden, die eigentlichen Daten verbleiben in der Cloud.

#### 3.1.2 Leichtgewichtiges, schnelles System

Durch die verwendete Architektur ist *Chrome OS* ein leichtgewichtiges System auf Grundlage von Linux; dabei wurde der Bootvorgang verkürzt, so dass das System innerhalb weniger Sekunden verfügbar ist. Gleichzeitig wurde der Login auf den *Single Sign-On*-Vorgang gebündelt; so sind alle benötigten Dienste sofort verfügbar. Die gute Arbeitsgeschwindigkeit resultiert vor allem auf der verwendeten Hardware. Die *Chromebooks* nutzen den Umstand der verwendeten kleinen SSD-Festplatte aus, dass die Lesegeschwindigkeit wesentlich höher liegt als die Schreibgeschwindigkeit. Es werden zwar Daten aus der Cloud auf die Festplatte geschrieben; da diese Daten aber über das Internet oder eine Netzwerkverbindung übertragen werden (im besten Falle mit 100 mBit/s), fällt die geringere Schreibgeschwindigkeit nicht ins Gewicht. Ohnehin wird auch bei *Chrome OS* im Betrieb gewöhnlich mehr von der Festplatte gelesen als geschrieben; so arbeitet das Betriebssystem schneller als mit einer herkömmlichen Festplatte. Der Akku wird durch den geringeren Stromverbrauch der SSD-Festplatte geschont, so sind lange Akkulaufzeiten gewährleistet.

*Chrome OS* wurde in vielen Bereichen für die Bedürfnisse des Cloud Computing angepasst, so ist die Oberfläche auf den Browser *Chrome* und einen kleinen Dateimanager reduziert worden. Ebenfalls wurde die Oberfläche von *Chrome* noch weiter optimiert, damit möglichst der gesamte Bildschirm genutzt werden kann. Durch die Verwendung des Linux-Kernels sind die wenigen verwendeten Prozesse leichtgewichtig und verbrauchen wenig Arbeitsspeicher.

### 3.1.3 Sicherheitsmechanismen

*Chrome OS* weist einige Sicherheitsmerkmale auf, so sind die Benutzerdaten komplett verschlüsselt, die Firmware wird bei jedem Systemstart überprüft, und *Chrome* nutzt Sandboxing. Weiterhin löscht der Gastaccount alle Daten beim Beenden des Systems; automatische Updates werden automatisch installiert, so dass eine Wartung entfällt. Da sämtliche Daten in der Cloud gespeichert sind, kann bei einem Virenbefall einfach das System zurückgesetzt werden, und die Daten gehen nicht verloren.

## 3.2 Nachteile

### 3.2.1 Datenschutz

Schon kurz nach der Bekanntgabe von *Chrome OS* kamen von vielen Seiten Bedenken über die Einhaltung von Datenschutzrichtlinien. Richard Stallman, Gründer der *Free Software Foundation*, kritisiert beispielsweise die grundsätzliche Haltung zu Cloud Computing, das generell die Menschen zum *Careless Computing* verleitet [17]. Die Benutzer speichern leichtsinnig ihre Daten auf den fremden Servern, und werden so zum Leichtsinns im Umgang mit ihren Daten verleitet. So entziehen sich die Nutzer selbst der Kontrolle ihrer eigenen Daten. Rechtliche Bedenken schließt Stallmann ebenfalls nicht aus, da die abgespeicherten Daten z.B. in der USA anderen Datenschutzrichtlinien unterliegen als etwa in Europa. Die grundsätzliche Kritik an *Chrome OS* weitet sich auf verschiedene Bereiche aus, die von Google als Sicherheitsmerkmale oder Vereinfachungen für Benutzer deklariert. Da die Datenspeicherung und Verwaltung generell auf den Servern von Google stattfindet, kann Google diese persönlichen Daten nutzen, um z.B. noch gezielter Anzeigen zu platzieren, und so den Umsatz des Unternehmens zu steigern. Da ein *Chromebook* permanent mit dem Internet verbunden sein muss, um den vollen Funktionsumfang zu nutzen, lassen sich anhand der ausgetauschten Daten leicht Benutzerprofile erstellen. Ähnliches leistet jetzt schon der Browser *Chrome* auf einem PC, der z.B. die Suchfunktion in Echtzeit ausführt, und so die Suche erleichtern soll.

### 3.2.2 Systemzwänge

Die *Chromebooks* sind nur für die Nutzung mit *Chrome OS* entwickelt worden. Es existiert zwar durch Umlegen eines Schalters ein Entwicklermodus, der eine Installation weiterer Betriebssysteme zulässt; jedoch kann dann *Chrome OS* nicht mehr genutzt werden. So werden Nutzer gezwungen, nur jeweils ein Betriebssystem zu verwenden. Gegen Veränderungen ist *Chrome OS* gesichert, was eine Modifikation wie etwa unter Linux nicht einfach macht; das Dateisystem ist zudem nicht schreibbar. Google fasst mit diesen Mechanismen Fuß auf dem Betriebssystem-Sektor, der vor allem von Windows-Betriebssystemen kontrolliert wird. Dabei geht das Unternehmen den Weg vom Suchmaschinenanbieter über Betriebssystemanbieter hin zum Verwalter von persönlichen Daten und Einstellungen.

Weiterer Zwang ist eine dauerhafte Internetverbindung, da keine Daten von der Cloud dauerhaft auf die Festplatte kopiert werden können, und somit immer in der Cloud verbleiben. Um die Daten immer synchron zu halten, ist die Internetverbindung zwingend notwendig; auch besitzen viele *Apps* trotz HTML 5 nur mit einer Internetverbindung den vollen Funktionsumfang.

Der Benutzer von *Chrome OS* kann zwar *Apps* entwickeln, und damit den Funktionsumfang erweitern; letztlich ist der Benutzer von *Chrome OS* aber ein reiner Konsument, insofern er nicht an der Entwicklung von *Chromium OS* beteiligt ist. Aber auch *Chromium OS* besitzt viele restriktive Mechanismen, da die Entwicklung letztlich von Google gesteuert wird, und schließlich in das fertige *Chrome OS* einfließen werden.

## Literatur

- [1] Google. *Introducing The Google Chrome OS*. <http://googleblog.blogspot.com/2009/07/introducing-google-chrome-os.html/>
- [2] Heise Online. *Googles Chrome OS soll gegen Ende des Jahres verfügbar sein*. <http://www.heise.de/newsticker/meldung/Googles-Chrome-OS-soll-gegen-Ende-des-Jahres-verfuegbar-sein-1014494.html>
- [3] The Chromium Projects. *Chromium OS - The Chromium Projects*. <http://www.chromium.org/chromium-os/>
- [4] Heise Online. *Google greift im Notebookmarkt an*. <http://www.heise.de/newsticker/meldung/Google-greift-im-Notebook-Markt-an-1241653.html>
- [5] Google. *Google Chrome blog: Google Chrome OS - FAQ*. <http://chrome.blogspot.com/2009/07/google-chrome-os-faq.html>
- [6] Heise Online. *Chrome OS: Cloud Computing, aber kein Touch*. <http://www.heise.de/newsticker/meldung/Chrome-OS-Cloud-Computing-aber-kein-Touch-1149335.html>
- [7] Heise Mobil. *Ausprobiert: Samsungs Netbook mit Chrome OS*. <http://www.heise.de/mobil/artikel/Was-bei-Googles-Cloud-Betriebssystem-noch-fehlt-1267237.html>
- [8] Heise Online. *Samsungs Chromebook und Chromebox nicht in Deutschland*. <http://www.heise.de/newsticker/meldung/Samsungs-Chromebook-und-Chromebox-nicht-in-Deutschland-1416806.html>
- [9] The Chromium Projects. *Software architecture*. <http://www.chromium.org/chromium-os/chromiumos-design-docs/software-architecture>
- [10] The Chromium Projects. *Firmware boot and recovery*. <http://www.chromium.org/chromium-os/chromiumos-design-docs/firmware-boot-and-recovery>

- [11] The Chromium Projects. *Verified boot*. <http://www.chromium.org/chromium-os/chromiumos-design-docs/verified-boot>
- [12] The Chromium Projects. *Login*. <http://www.chromium.org/chromium-os/chromiumos-design-docs/login>
- [13] The Chromium Projects. *User experience*. <http://www.chromium.org/user-experience>
- [14] The Chromium Projects. *Out of memory handling*. <http://www.chromium.org/chromium-os/chromiumos-design-docs/out-of-memory-handling>
- [15] c't Heft 9/2011. *Wolken-Druckerei, S.108-111*
- [16] Technology Review. *Nicht so ganz PC*. <http://www.heise.de/tr/artikel/Nicht-so-ganz-PC-989891.html>
- [17] The Guardian. *Google's ChromeOS means losing control of data, warns GNU founder Richard Stallman*. <http://www.guardian.co.uk/technology/blog/2010/dec/14/chrome-os-richard-stallman-warning>

## Abbildungsverzeichnis

1	Architekturübersicht, Quelle: siehe [9] . . . . .	5
2	Firmwareübersicht, Quelle: siehe [9] . . . . .	5
3	Oberfläche, Quelle: siehe [9] . . . . .	7
4	<i>Chrome</i> mit Tableiste, Quelle: siehe [13] . . . . .	8