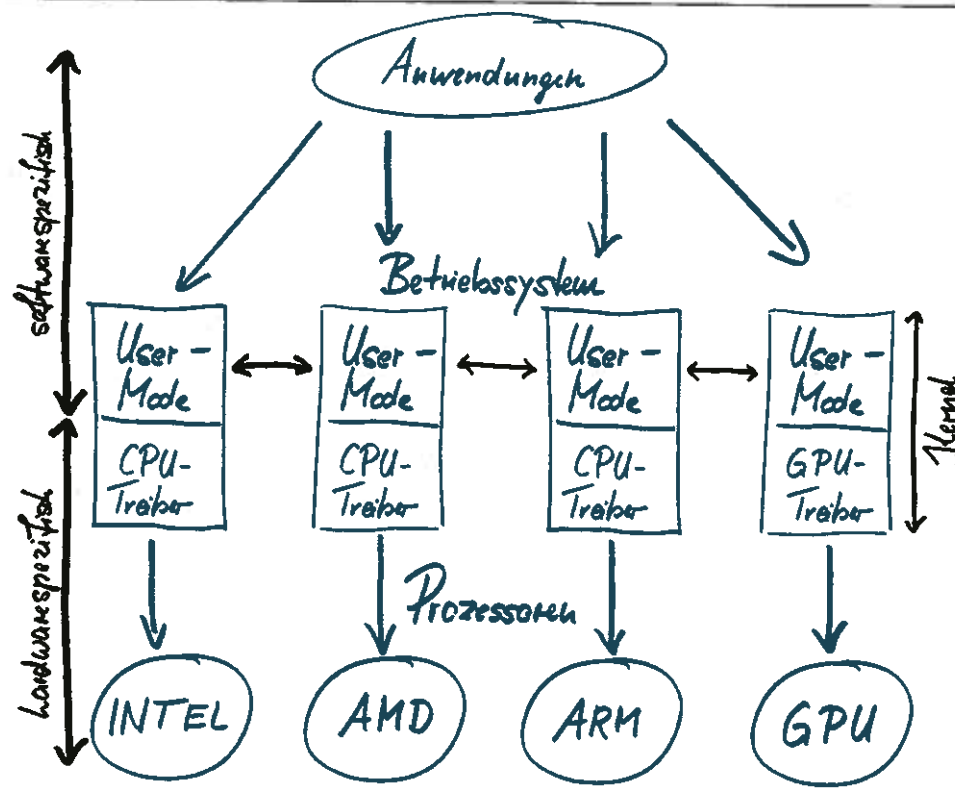


**BARRELFISH: OPTIMAL FÜR MULTICORE**

Das Betriebssystem hat pro Prozessorkern einen Kernel und pro CPU-Typ einen speziellen Treiber. Das hat zwei Vorteile: Erstens läuft es gleichzeitig auf unterschiedlichen CPU-Typen, zweitens steigt im Gegensatz zu herkömmlichen Systemen wie Windows die Rechenleistung linear mit der Zahl der Prozessorkerne an



**PROGRAMME AUSFÜHREN** Startet eine Software mehrere Threads oder Prozesse, holen sich die einzelnen Kernel diese Aufgaben selbstständig ab - wenn möglich im Batch-Modus, sodass sie mehrere hintereinander ausführen. Dadurch sind alle Kernel gleichmäßig ausgelastet.

**MULTI-KERNEL** Das Betriebssystem richtet für jeden Prozessorkern einen eigenen Kernel ein. Der Softwareteil mit dem User-Mode ist dabei immer identisch. Per Messaging verständigen sich die einzelnen Kernels darüber, wer gerade welche Aufgaben abarbeitet.

**VERSCHIEDENE PROZESSOREN** Die vorhandene Hardware steuert jeder Kernel über einen angepassten CPU-Treiber an. Dadurch kann Barrelfish gleichzeitig auf unterschiedlichen Prozessoren rechnen und auch die GPU der Grafikkarte gezielt für spezielle Operationen einsetzen.

# Erste Skizzen für WINDOWS 8

Microsoft kann mit Windows 8 ein großer Wurf gelingen. In seinen **Forschungslabors** rund um die Welt entstehen schon jetzt spannende Technologien für das künftige Betriebssystem

VON MARKUS MANDAU

**D**ie Stimmung auf der Microsoft-Konferenz PDC (Professional Developer Conference) war gut. Doch dann versteinerte das Gesicht von Steven Sinofsky, Chef der Entwickler von Windows. „Was planen Sie für das neue Windows 8?“, hatte ein Reporter gefragt. Sinofsky schwieg. Der Reporter wiederholte die Frage. Dann endlich antwortete Sinofsky: „Ich habe nichts gesagt. Windows 8? Das waren Ihre Worte, nicht meine.“ Bei diesem Schweigen und Abwiegen von Microsoft ist es geblieben, wenn auch mittlerweile einige Details an die Öffentlichkeit gedrungen sind (► S. 20). Doch die beschäftigen sich nur mit Änderungen an der Oberfläche, nicht mit der Technik dahinter.

So ruhig alles nach außen scheint, so konzentriert arbeitet Microsoft intern bereits an Windows 8. Vorschnelle Mitarbeiteräußerungen lassen schon jetzt zwei mögliche Veröffentlichungs-Termine erahnen: Ein Entwickler hat in seinem Blog den ersten Juli 2011 als voraussichtlichen Beginn des Lifecycle-Supports angegeben - er arbeitet inzwischen nicht mehr für den Konzern. Andere Quellen, zumeist aus PowerPoint-Präsentationen, setzen den RTM-Zeitpunkt (Release to Manufacturing) auf 2012. Einer der Gründe für Microsofts beharrliches Schweigen zu Windows 8 dürfte der immer noch nachwirkende Vista-Schock sein. Das Betriebssystem, das auf der Boom-Plattform Netbook nicht läuft.

Der Zwang, die richtige Technologie zum passenden Zeitpunkt auf den Markt zu bringen, verschärft sich bei Windows 8. Gerade weil Windows 7 so erfolgreich ist. Umfragen zeigen, dass die Kunden ein verbessertes Windows 7 nicht wollen. Laut Forrester Research werden nur ein Prozent aller Firmen in den USA und Europa das aktuelle Windows 7 überspringen, um auf das nächste zu warten. Den restlichen 99 Prozent muss Microsoft aber einen Anreiz bieten, um später doch von Windows 7 auf 8 zu wechseln.

Kein Wunder, dass ein Microsoft-Mitarbeiter in einem - inzwischen gelöschten - Blogbeitrag andeutet, dass Windows 8 komplett anders wird und die Handhabung des PCs revolutionieren soll. So viel Wille zur Veränderung tut auch Not, denn die

Hardware-Plattform für Windows droht jetzt schon empfindlich zu schrumpfen. Kleine Rechner - von den Smartphones bis zum iPad - erobern den Markt und viele nutzen nicht Intel-, sondern ARM-Prozessoren, die Windows nicht ansteuern kann.

Für Windows 8 muss Microsoft größere Veränderungen vornehmen. Das dafür nötige Know-how dazu steht dem Konzern dank seiner weltweiten Forschungsprojekte zur Verfügung, er muss es nur noch umsetzen. Wobei vier Projekte aus den Microsoft-Labors für die Zukunft von Windows besonders interessant sind.

**Betriebssystem Barrelfish: Es läuft auf jeder Hardware**

„Wir wollten nur einmal schauen, ob so etwas funktioniert“, resümiert der Entwickler Andrew Baumann über das Betriebssystem Barrelfish, das das Unternehmen zusammen mit der Uni Zürich entwickeln lässt. Das Betriebssystem

sondere an Barrelfish: Es hat mehrere Kernel. Und ist dazu gedacht, auf sehr vielen sowie unterschiedlichen Prozessoren zu laufen - darunter auch die GPU der Grafikkarte. Die Entwickler nennen ihr Konzept ein Multikernel-OS, im Gegensatz zu den Monokernen von Windows, Mac OS und Linux.

Barrelfish ordnet jedem Prozessorkern einen Kernel zu. Dabei arbeiten die einzelnen Kernel als Knotenpunkte (Node) in einem Netzwerk aus Rechenkernen. Jeder einzelne Node besteht aus zwei Elementen: Ein Teil arbeitet im User-Mode und organisiert die Aufgaben für die und zwischen den CPUs - er ist in jedem Kernel identisch. Der andere übernimmt die Rolle des Hardware-Interfaces und dient als CPU-Treiber. Das Interface ist auf die Hardware abgestimmt, also auf die Befehlsätze und Eigenschaften einer x86- oder ARM-CPU beziehungsweise einer GPU.

Die Ressourcen-Zuteilung der CPUs zu den Prozessen wird in Barrelfish anders organisiert als in Windows. In Windows 7 hat Microsoft den für das Prozessmanagement zuständigen zentralen Dispatcher Lock so weit optimiert, dass er kein Nadelöhr mehr darstellt, wenn das System mit acht oder 16 Kernen rechnen muss. Windows 7 kann daher auf einem Rechner mit bis zu 256 CPUs laufen. Doch je mehr Kerne hinzukommen, umso geringer ist ihr Leistungsgewinn.



„Windows 8? Dazu sage ich gar nichts.“  
Steven Sinofsky, Chef der Windows-Programmer

Barrelfish geht das Problem systematischer an. Hier stimmen sich mehrere Dispatcher, einer pro Kernel, miteinander ab. Die Kommunikation zwischen den einzelnen Kernen findet mittels Remote Procedure Calls statt, für die ein gemeinsamer Speicherbereich vorgesehen ist. Dank dieser Mechanismen sind der Zahl und der Art der beteiligten CPUs keine Grenzen gesetzt. Die Performance von Barrelfish steigt gleichmäßig mit der Anzahl der CPUs.

**DAS BRINGT ES WINDOWS 8:** Viele Ideen in Barrelfish greifen weit in die Zukunft, doch zwei Fähigkeiten sind interessant: Die Portierung von Barrelfish auf andere CPU-Typen wie ARM-Prozessoren ist leicht. Zudem nutzt Barrelfish die GPU über einen eigenen Kernel, wann immer es geht. Hier lockt ein großer Performancegewinn für spezielle Rechenoperationen, der bisher nur schlecht zu realisieren ist. Nur für die Portierung von in C oder C++ geschriebener Software auf andere Prozessortypen bietet Barrelfish keine automatisierte Lösung.

**Betriebssystem Singularity: Ein System, das nicht abstürzt**

Gedanklich gehört der Bluescreen fest zu Windows – auch wenn er in Vista und Windows 7 sein hässliches Gesicht nur noch selten zeigt. Inzwischen hat Microsoft mit Singularity sogar ein System entwickelt, das keinen Bluescreen kennt, da es als Ganzes nicht abstürzen kann. Es sollte laut Microsoft-Entwickler Galen Hunt „auch in einer feindseligen Umgebung bestehen“. Singularity gelingt dieser Trick, weil das System die ablaufenden Prozesse per Software von einander und vom Zugriff auf die Hardware isoliert.

Die Software-Isolierung funktioniert nur bei Programmen, die in einer Laufzeitumgebung (Runtime) wie Java und .NET-Anwendungen ausgeführt werden. Die Runtime sorgt dafür, dass Code nicht „ausbricht“ und einen Pufferüberlauf erzeugt, der einen Absturz zur Folge hat. Damit umgeht Singularity die Gefahr, die von schlecht programmierten Anwendungen oder gar Malware ausgeht.

Leider lässt sich die Idee nicht eins zu eins für Windows umsetzen. Die meisten Programme, mit denen Anwender zu tun haben, sind in Sprachen wie C oder C++ geschrieben. Da sie direkt in der CPU ausgeführt werden, haben sie prinzipiell eine höhere Performance. Windows schützt sich vor den potenziellen Gefahren, die in diesen Programmen schlummern, indem es sie im User-Mode der CPU ausführen lässt – er hat weniger Rechte. Diese Schutzmaßnahme mindert den Performance-Vorsprung dieser Programmiersprachen allerdings wieder.

Während Windows den Kernel-Mode für die wichtigen Systemfunktionen reserviert, kann Singularity alle Anwendungen geschützt im Kernel-Mode laufen lassen. Es erreicht dadurch trotz der Laufzeitumgebung eine ähnlich gute Performance wie C-Programme unter Windows 7 (► CHIP 09/2009, S. 70 ff.).

**DAS BRINGT ES WINDOWS 8:** Im Kernel-Mode betreibt Windows bisher nur die hardwarenahen Teile des Betriebssystems, hauptsächlich Gerätetreiber. Mittels Software-Isolierung könnten aber auch Systemdienste wie etwa die COM-Schnittstelle, die unter anderem für ActiveX zuständig ist, in diesem Modus laufen. Damit das klappt, müssten diese Systemkomponenten beispielsweise in das .NET-Framework überführt werden. Der Effekt wäre ein stabileres System, das auch für Malware-Angriffe weniger anfällig ist.

**Browser Gazelle: Geschützt gegen Webattacken**

Der Browser ist die zentrale Anwendung im Betriebssystem, für viele User sogar die einzige. Kein Wunder, dass der Wettlauf um den

schnellsten sowie sichersten Browser voll entbrannt ist. So wird der fertige IE9 schon für Ende 2010 erwartet, deutlich früher als das neue Betriebssystem. Der Browser für Windows 8 heißt daher IE10.

Die Generation um den IE9 holt technisch erst mal das auf, was die Browser in den letzten Jahren versäumt haben: Sie erhöhen die Performance, indem sie zusätzliche Ressourcen nutzen, etwa die Rechenkraft der Grafikkarte. Und sie sichern die Browserarchitektur ab, indem sie die Plugins in eigene Prozesse auslagern. Aber sie schützt nicht vor Attacken, die einen Datendiebstahl zum Ziel haben – etwa Cross Site Scripting, das im Zusammenspiel zwischen Browser und Website stattfindet. Kein Wunder, meint Helen J. Wang, die Microsofts Browserprojekt Gazelle leitet, schließlich sei die jetzige Generation dafür gar nicht entwickelt worden.

Deshalb berücksichtigt Gazelle nicht nur die Art des Codes, sondern auch seine Quelle. Gazelle kann JavaScript einer vertrauenswürdigen Website wie [www.chip.de](http://www.chip.de) normal rendern. Wird jedoch Skript-Code von exter-

**Windows 8 ist reif für neue Technologien**

Mit dem nächsten Betriebssystem erhält Microsoft die ideale Gelegenheit, um unsichere und veraltete System-Komponenten teilweise oder sogar komplett auszutauschen – endlich.

**64 statt 32 Bit:** Windows 8 dürfte nur noch als 64-Bit-System ausgeliefert werden. Etwas anderes ist auf moderner Hardware nicht sinnvoll. Zumal 32-Bit-Software normal weiterfunktioniert.

**EFI statt BIOS:** Der BIOS-Nachfolger EFI ist jetzt schon unter der 64-Bit-Version von Vista und Windows 7 möglich. Zu Windows 8 sollte Microsoft von den Hardware-Hersteller EFI-kompatible Treiber für ihre Boards einfordern.

**.NET statt COM:** Programme nutzen bisher das COM-Interface, um mit Systemkomponenten zu kommunizieren. Eine Übertragung sensibler Komponenten wie ActiveX ins .NET-Framework würde die Systemsicherheit deutlich erhöhen.

**MF statt DirectShow:** Ab Version 8 sollte Windows zur Multimedia-Wiedergabe nur noch die Media Foundation (MF) statt DirectShow nutzen. Sie garantiert bessere Stabilität und Performance dank Hardware-Beschleunigung per GPU.

nen Quellen auf die CHIP-Website nachgeladen, lässt ihn Gazelle in einem getrennten Prozess mit weniger Rechten laufen.

Von der Architektur her ist Gazelle wie ein kleines Betriebssystem aufgebaut. Deshalb firmiert es mittlerweile unter dem Namen ServiceOS. Dadurch kann Gazelle auch EXE-Dateien, also nativen Code, ausführen und externe Hardware wie eine Webcam oder das GPS direkt ansprechen.

**DAS BRINGT ES WINDOWS 8:** Das Sicherheitskonzept von Gazelle geht den Schritt weiter, der in Browsern noch nicht implementiert ist, und macht das Web zu einer sicheren Zone. Die Architektur lässt sich auch für den IE10 umsetzen, ohne den Browser gleich zum Neben-Betriebssystem zu machen, wie es Gazelle eigentlich vorsieht.

**Paketverwaltung CoApp: Keine Softwareprobleme mehr**

Die meisten Programme werden für die Windows-Plattform entwickelt, denn das Betriebssystem läuft auf fast 95 Prozent aller PCs. Trotzdem könnte die Software-Unterstützung noch besser werden, besonders im

Bereich der Open-Source-Anwendungen. Viele Entwickler bezeichnen die Portierung ihrer Programme, die sie meistens für die Linux-Plattform schreiben, auf Windows als einen Alptraum. Der beginnt oft damit, dass die kompilierte Software einige Funktionen in bestimmten DLLs nutzen will. Ob der Programmierer diese in der passenden Version auf jedem Windows findet, weiss er nicht.

Das Projekt CoApp soll das Problem lösen. Der Microsoft-Programmierer Garrett Serack plant, einen Packet Manager für die Windows-Plattform zu entwickeln, wie er in der Linux-Welt zum Standard gehört. CoApp startet als Open-Source-Projekt, das laut Serack in den nächsten ein bis zwei Jahren umgesetzt werden soll. Es wäre damit passend zu Windows 8 verfügbar, wird aber auch auf XP, Vista und Windows 7 laufen.

Zentraler Ansatzpunkt von CoApp ist Windows Side-by-Side (WinSxS). Dank dieser Systemkomponente arbeiten verschiedene Versionen einer DLL reibungslos miteinander. Wenn sich eine Software installiert, kann sie über ein Manifest beschreiben, welche DLL-Version sie benötigt. Fehlt das Manifest, nimmt Windows einfach die Standardversion der DLL.

Am Ende soll CoApp eine Plattform bieten, um Programme, die im Source-Code vorliegen, problemlos für Windows zu kompilieren. Für den User bedeutet das, dass ihm mehr Open-Source-Programme in der neuesten Version zur Verfügung stehen – und dass er ein zentrales Modul hat, über das er sie einfach und sicher aktualisieren kann.

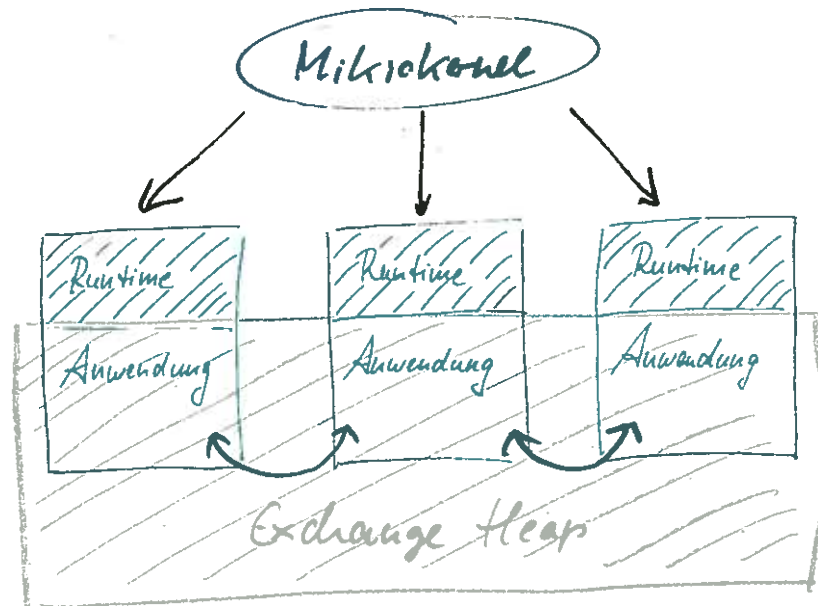
**DAS BRINGT ES WINDOWS 8:** Die nächste Windows-Version wird das CoApp-Projekt nicht nur adaptieren, sondern auch auf kommerzielle Software ausweiten. Vielleicht dient es sogar als Grundlage für eine eigene App-Plattform. Dies lässt eine Stellenausschreibung vermuten: Für Windows 8 suchte Microsoft Anfang des Jahres einen Programmierer, um den Update Client weiterzuentwickeln. Anscheinend soll Windows 8 auch Updates von vertrauenswürdigen Software-Anbietern einspielen, so wie es jetzt schon mit den Gerätetreibern geschieht. Die neuesten Sicherheitslücken im Adobe Reader wären damit sofort gestopft.

**FAZIT:** Microsoft fördert eine Reihe von Projekten, die Windows im Kern verbessern, es schneller, vielseitiger und sicherer machen. Das sind genau die Versprechen, die Windows 8 einlösen sollte. Möglicherweise schafft Microsoft damit sogar ein Betriebssystem für alle Plattformen vom Smartphone bis zum Großrechner – damit wäre es dann wieder einmal konkurrenzlos. ☐

MARKUS.MANDAU@CHIP.DE

**SINGULARITY: STABILES SYSTEM**

Singularity kennt keinen Bluescreen und entschärft Malware. Der Kernel startet Anwendungen nur in isolierten Runtimes, die dann über den Exchange Heap kommunizieren. Ein Pufferüberlauf ist unmöglich

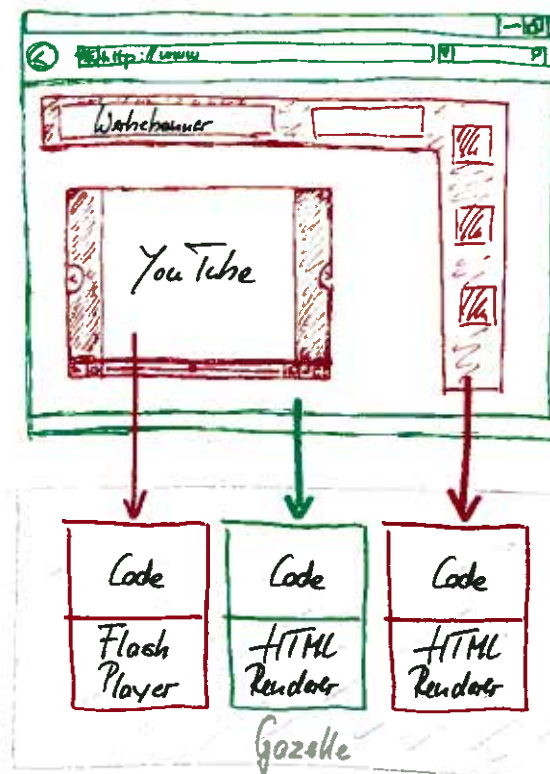


**PROZESSE STARTEN**  
Der Kernel startet jede Anwendung per Runtime in einem isolierten Prozess. Nutzen mehrere Anwendungen dieselben Ressourcen, geben sie diese im Exchange Heap wieder frei.

**PROZESSE WERDEN ISOLIERT**  
Die Runtime isoliert den Programmcode der Anwendung auf Software-Ebene. Das klappt schon für Java- und .NET-Anwendungen, nur C-Programme laufen in Singularity nicht.

**GAZELLE: SICHERER BROWSER**

Gazelle ermöglicht angstfreies Surfen, da es im Gegensatz zu aktuellen Browsern den Ursprung von Webinhalten analysiert und in getrennten Prozessen ausführt. Unsicherer Code läuft nur mit verminderten Rechten



**INHALTE TRENNEN**  
Gazelle trennt den Web-Code in unterschiedliche Prozesse auf, wenn er von verschiedenen Quellen geladen wird.

**RECHTE ZUWEISEN**  
Die Prozesse erhalten unterschiedliche Rechte. Code aus unbekanntem Quellen wird isoliert und mit weniger Rechten ausgeführt.

rot: eingeschwänkte Rechte  
grün: hohe Rechte