

September 26, 2025

Wenn Systeme Grenzen setzen, ist es Zeit für ein Neues.

Ein Whitepaper über das Betriebssystem der nächsten Generation

Jenning Schäfer

jenningschaefer@gmx.de

ABSTRACT

Die digitale Welt ruht auf Fundamenten, die zunehmend an ihre Grenzen stoßen.

x86 ist leistungsfähig, aber überkonstruiert und ineffizient.

ARM verspricht Effizienz, jedoch nur im Tausch gegen Lizenz und Abhängigkeit.

Linux dominiert Server und eingebettete Systeme, scheiterte jedoch im mobilen Bereich an proprietären Treibern und Fragmentierung.

Android ist „offen“ nur im Kern – das Ökosystem bleibt fest an Google gebunden.

Apple wiederum liefert Perfektion, aber nur im Käfig.

Gleichzeitig entstehen neue Grundlagen:

RISC-V als offene Architektur.

Rust als sichere Systemprogrammiersprache.

Mikrokern als belastbares Fundament.

Es sind Technologien im Aufbruch, nicht im Niedergang. Genau hier liegt die Chance, ein neues Ökosystem zu schaffen.

Open Nexus OS baut auf diesen Prinzipien: ein quelloffenes, modulares Betriebssystem für viele Geräteklassen, das Offenheit, Sicherheit und Benutzerfreundlichkeit vereint. Eine Basis, auf der sowohl Entwickler als auch Unternehmen bauen können – ohne neue Mauern, ohne neue Abhängigkeiten.

Dieses Projekt ist mehr als eine technische Alternative. Es ist der Versuch, die digitale Zukunft nicht länger auf brüchigen Fundamenten zu errichten, sondern auf einem Fundament, das offen, nachhaltig und zukunftsfähig ist.

Inhaltsverzeichnis

1	Wenn Technologie im Weg steht, ist sie nicht mehr smart.	4
1.1	Analyse der bestehenden Ökosysteme	4
1.1.1	Apple – Perfektion im Käfig	4
1.1.2	Microsoft – Verwaltung statt Erlebnis	4
1.1.3	Google & Android – Offenheit im Schaufenster	4
1.1.4	Linux – Stark im Serverraum, schwach in der Hosentasche	5
1.2	Huawei OpenHarmony – unterschätzt auf eigene Gefahr	5
2	Warum zwischen offen und einfach wählen?	5
3	RISC-V als Hardware-Basis	6
4	Mikrokern-Design – Ein Fundament, das nicht bricht	6
5	Das fehlende Puzzleteil – Warum Betriebssysteme ohne Werkzeug scheitern	6
6	From Spark to Ecosystem	7
7	Community ohne Mauern	8
8	Die richtige Person zur richtigen Zeit	8
9	Beyond Funding: Mitbauen statt Zuschauen	8
10	Ein nachhaltiger Weg nach vorn	9
11	Schlussfolgerung	9

1 Wenn Technologie im Weg steht, ist sie nicht mehr smart.

Wir leben in einer Welt, in der beinahe jedes Gerät mit dem Etikett smart versehen wird. Smartphone, Smartwatch, Smart Speaker, Smart Home. Ein inflationär gebrauchter Begriff – und doch bedeutet er in der Praxis meist nur: mehr Funktionen, mehr Menüs, mehr Konfigurationsaufwand.

Wahrhaft „intelligent“ ist nicht das Gerät, das uns mit Optionen überfrachtet, sondern jenes, das verschwindet. Das uns nichts mehr abverlangt. Wenn das Notebook sich von selbst entsperrt, sobald die Uhr am Handgelenk ist. Wenn Kopfhörer unaufgefordert zum richtigen Gerät wechseln. Wenn die gewünschte Funktion genau in dem Moment präsent ist, in dem sie gebraucht wird – nicht früher, nicht später.

Diese Form der Selbstverständlichkeit ist möglich. Doch sie bleibt ein Privileg der geschlossenen Ökosysteme. Apple gewährt sie – allerdings nur innerhalb seines Käfigs. Microsoft bietet sie – allerdings nur für jene, die sich der Logik seiner Konten und Richtlinien unterwerfen. Google ermöglicht sie – allerdings stets im Tausch gegen die intimsten Daten seiner Nutzer.

So entstehen Inseln des Komforts, perfekt poliert – und gleichzeitig umschlossen von Mauern.

Was fehlt, ist ein Gegenentwurf: ein Ökosystem, das dieselbe Mühelosigkeit bietet, ohne sich in Abhängigkeit zu verwandeln. Offen, überprüfbar, erweiterbar. Ein Fundament, das nicht einer Firma gehört, sondern allen, die es nutzen und gestalten.

Das ist der Anspruch von Open Nexus OS: Intelligenz, die nicht zum Preis der Freiheit erkauft wird.

1.1 Analyse der bestehenden Ökosysteme

1.1.1 Apple – Perfektion im Käfig

Apple beherrscht die Kunst der nahtlosen Integration wie kein Zweiter. Geräte erkennen einander, Dienste greifen ineinander, die Oberfläche wirkt wie aus einem Guss. Es ist die Eleganz der geschlossenen Form.

Doch die Vollkommenheit hat ihren Preis: Sie existiert nur, solange man sich dem Käfig beugt. Die Freiheit des Nutzers endet an der Mauer des Ökosystems. Innovation von außen findet nicht statt. Apples Welt ist nicht offen – sie ist nur schön eingerichtet.

1.1.2 Microsoft – Verwaltung statt Erlebnis

Microsofts Reich bleibt der Desktop. Die Integration funktioniert – allerdings in erster Linie für IT-Abteilungen. Konten, Policies, Active Directory, Azure: eine Architektur der Kontrolle, nicht der Leichtigkeit.

Für Unternehmen mag das eine Stärke sein. Für den Alltag bleibt es ein Flickenteppich. Mobile Nutzung? Halbherzig. Cross-Device? Stückwerk. Microsoft liefert kein Erlebnis, sondern Verwaltung – effizient für Administratoren, frustrierend für Menschen.

1.1.3 Google & Android – Offenheit im Schaufenster

Android trat einst mit dem Versprechen der Offenheit an. In der Praxis ist daraus ein Flickenteppich geworden: Jeder Hersteller baut seine eigene Variante, Updates bleiben uneinheitlich, Schnittstellen brechen auseinander. Ein gemeinsames Ökosystem entsteht so nicht – nur ein paar zusammengenähte Cloud-Anwendungen, die von Google zusammengehalten werden.

Echte Integration findet man fast ausschließlich bei Samsung, das sich sein eigenes Sub-Ökosystem geschaffen hat. Der Rest ist Stückwerk.

Und im Zentrum bleibt Google: das Nadelöhr, durch das fast alles läuft. Wer Android ohne Google-Dienste nutzt, verliert den Großteil der Funktionalität. Offenheit gibt es hier wie im Schaufenster – sichtbar, aber nicht wirklich zugänglich.

1.1.4 Linux – Stark im Serverraum, schwach in der Hosentasche

Linux ist das Rückgrat des Internets. Stabil, vielseitig, unverzichtbar. Doch was im Serverraum glänzt, scheitert auf Geräten, die wir täglich in die Hand nehmen.

Der Grund liegt im Fundament: ein monolithischer Kernel, in dem Treiber im privilegierten Modus laufen – und damit das ganze System gefährden. Für Prozessoren und Grafikkarten existieren brauchbare Treiber. Für Modems, Antennen, Energieverwaltung? Proprietäre Black Boxes.

Darum läuft Linux nicht auf unseren Handys. Darum brauchen Autos Speziallösungen. Nicht, weil es an Talent mangelt. Sondern weil die Architektur nie für diese Realität geschaffen war.

1.2 Huawei OpenHarmony – unterschätzt auf eigene Gefahr

Es gibt bereits einen Kandidaten für das „eine Betriebssystem für alles“: Huaweis OpenHarmony. Modularer Kernel, offene Lizenz, geräteübergreifendes Konzept – auf dem Papier wirkt es wie die Antwort auf all die Fragen, die sich Europa seit Jahren stellt.

Doch die frei verfügbare Version ist stark beschnitten, zentrale Verwaltungsfunktionen fehlen – und am Ende bleibt die Abhängigkeit von einem einzigen Anbieter. Ausgerechnet China.

Wer glaubt, das sei keine Bedrohung, sollte sich erinnern: Auch bei Autos hat niemand geglaubt, dass chinesische Hersteller einmal deutsche Ikonen überholen. Auch Bell, IBM, Palm und Blackberry schienen unantastbar – bis sie es nicht mehr waren.

Die größte Gefahr ist nicht, dass OpenHarmony scheitert. Die größte Gefahr ist, dass es funktioniert.

2 Warum zwischen offen und einfach wählen?

Seit Jahrzehnten scheint die Entscheidung klar: Geschlossene Systeme bieten Bequemlichkeit – zum Preis der Freiheit. Offene Systeme bieten Freiheit – oft zum Preis der Nutzbarkeit.

So wurde es uns verkauft: Man könne nur das eine oder das andere haben. Apple steht für Perfektion im Käfig. Linux für Freiheit im Rohzustand. Dazwischen bleibt wenig Raum.

Doch die Wahrheit ist: Diese Wahl ist eine Illusion. Sie ist das Produkt veralteter Architekturen und kurzsichtiger Geschäftsmodelle – nicht ein Naturgesetz der Technik.

Warum sollte ein Gerät nicht zugleich nahtlos und offen sein? Warum sollte ein Ökosystem entweder mühelos oder frei sein? Warum sollte der Nutzer gezwungen sein, sich zwischen Komfort und Autonomie zu entscheiden?

Open Nexus OS lehnt diese falsche Alternative ab. Es strebt ein System an, das die Selbstverständlichkeit geschlossener Plattformen mit der Offenheit freier Software verbindet. Ein Betriebssystem, das nicht zwischen Komfort und Freiheit unterscheidet – weil beides Teil desselben Designs ist.

Das Ziel ist keine weitere Plattform im Konkurrenzkampf. Das Ziel ist ein Fundament, auf dem alle bauen können: ohne neue Mauern, ohne neue Abhängigkeiten, ohne die alten Kompromisse.

3 RISC-V als Hardware-Basis

„Alte Architekturen bauen keine neue Zukunft.“

x86 hat uns den Personal Computer gebracht – und sich dabei selbst in eine Sackgasse manövriert. Jede Generation wurde komplizierter, schwerer, ineffizienter. Ein Bauwerk aus Flickwerk: mächtig, aber überladen.

ARM versprach Leichtigkeit. Doch Leichtigkeit unter Lizenz bleibt Abhängigkeit. Wer ARM nutzt, mietet sein Fundament – und lebt mit Regeln, die andere diktieren.

Beide Architekturen sind Ergebnisse einer Vergangenheit, die nicht mehr unsere Zukunft ist. Sie tragen die Narben ihrer Herkunft: x86 als überkonstruierter Veteran, ARM als Eigentum weniger Konzerne.

RISC-V hingegen ist frei. Eine Architektur ohne Ketten, modular, offen, erweiterbar. Kein Fossil, sondern junges Fundament. Kein Mietobjekt, sondern Baugrund in eigenem Besitz.

Es ist nicht „billiger Ersatz“ für die bestehenden Systeme, sondern eine neue Chance: Technologien im Aufbruch, nicht im Niedergang. Und genau deshalb ist RISC-V die Basis von Open Nexus OS – weil die Zukunft nicht auf dem schnellsten Relikt von gestern steht, sondern auf dem offensten Fundament von morgen.

4 Mikrokern-Design – Ein Fundament, das nicht bricht

Monolithische Kernel sind Meisterwerke vergangener Zeiten – und zugleich ihre größte Hypothek. Alles, was das System steuert, läuft im privilegiertesten Raum: Treiber, Dateisysteme, Netzwerke. Ein einziger Fehler genügt, und das ganze Konstrukt stürzt ein. Auf Servern mag man das tolerieren, weil die Kernhardware offen dokumentiert ist. Doch in der mobilen Welt sieht es anders aus: Modems, Antennen, Energiemanagement – fast alles proprietäre Black Boxes. Und niemand weiß, was darin geschieht, bis es zu spät ist.

Darum wurde Linux nie das universelle Mobile-OS. Nicht wegen mangelnder Brillanz, sondern weil die Architektur blind vertraut, wo sie skeptisch sein müsste.

Ein Mikrokern dreht dieses Verhältnis um. Er reduziert das, was im Kern läuft, auf das absolut Notwendige: Speicherverwaltung, Prozessplanung, Interprozesskommunikation. Alles andere – Treiber, Dateisysteme, Dienste – läuft isoliert im Userspace. Fehler werden eingekapselt, nicht vererbt. Ein Kartenhaus wird zum Fundament.

Und wenn dieser Mikrokern in Rust geschrieben ist, fällt auch die letzte Schwachstelle: unsichere Speicherzugriffe. Sicherheit entsteht nicht mehr durch Hoffnung, sondern durch Architektur.

Das ist kein neues Feature. Es ist die Umkehrung eines Denkfehlers, der Betriebssysteme seit Jahrzehnten prägt.

5 Das fehlende Puzzleteil – Warum Betriebssysteme ohne Werkzeug scheitern

Ein Betriebssystem allein genügt nicht. Ohne die richtigen Werkzeuge bleibt es eine leere Hülle. Denn was nützt die beste Plattform, wenn niemand darauf bauen kann?

Die großen Player haben das verstanden. Apple gab den Entwicklern Xcode. Microsoft stellte Visual Studio bereit. Google schuf Android Studio. Mit diesen Werkzeugen öffneten sie die Tür

zu ihren Ökosystemen – allerdings immer nur unter einer Bedingung: Wer hineinwill, muss die Regeln des Eigentümers akzeptieren.

Linux dagegen hatte nie diese Tür. Stattdessen: ein Flickenteppich aus Editoren, Compiler-Ketten, Paketmanagern. Genial für Experten, unbrauchbar für alle anderen. Deshalb blieb Linux trotz seiner Stärke ohne ein wirkliches App-Ökosystem für den Massenmarkt.

Und genau hier setzt Nexus Studio an.

Es ist mehr als ein Editor. Es ist das fehlende Bindeglied. Ein Werkzeug, mit dem eine Anwendung nicht zehnmals entwickelt werden muss – sondern einmal, für alle Geräte. Ein Werkzeug, das offen genug ist, um Gemeinschaft zu schaffen, und zugleich flexibel genug, um auch geschlossene Anteile zu erlauben, wenn Unternehmen das brauchen.

Denn Offenheit heißt nicht, alle Türen auszuhängen. Offenheit heißt, die Wahl zu lassen. Wer Offenheit zur Ideologie macht, baut neue Mauern. Wer sie pragmatisch versteht, baut Brücken.

Nexus Studio ist genau diese Brücke. Das Puzzleteil, das bisher gefehlt hat. Ohne sie bleibt jedes „offene“ Betriebssystem Stückwerk. Mit ihr wird ein Ökosystem möglich.

6 From Spark to Ecosystem

„Kein Flicken, kein Feature – ein Fundament.“

Wir bauen nicht das nächste Feature, wir bauen die nächste Grundlage. Ein Betriebssystem entsteht nicht auf einmal, sondern Schicht für Schicht – jede tragfähig genug, um benutzt zu werden, bevor die nächste entsteht.

1. Der Funke – Proof of Life Alles beginnt mit dem Sichtbarsten: ein System, das startet, ein Login, ein Launcher. Desktop und Mobile wechseln ineinander. Nicht perfekt, aber lebendig. Der Beweis: Es funktioniert.

2. Das Fundament – Kern und Ordnung Danach folgt das Unsichtbare: Ressourcenverwaltung, erste Bibliotheken, Standards. Keine Abkürzungen, keine Hacks. Nur das, was ein Betriebssystem trägt – gebaut für Dauer, nicht für Demo.

3. Die Sprache – Ein Framework für Anwendungen Ein System lebt nicht ohne Software. Deshalb entsteht eine deklarative Schicht, mit der sich Anwendungen sauber und konsistent entwickeln lassen. Moderne Schnittstellen, modernes Design – nicht improvisiert, sondern geplant.

4. Das Werkzeug – Nexus Studio Hier fällt die Entscheidung, ob aus einer Idee ein Ökosystem wird. Nexus Studio ist die Entwicklungsumgebung, die Entwicklern die Tür öffnet: einmal programmieren, überall einsetzen. Offen genug für Gemeinschaft, flexibel genug für Unternehmen.

5. Die Essentials – Was Nutzer erwarten Medien, Konten, Speicher, Netzwerke. All das, was Menschen als selbstverständlich ansehen – aber konsequent offen und konsistent umgesetzt.

6. Das Unsichtbare – Sicherheit und Zuverlässigkeit Updates, Paketverwaltung, Sicherheitsarchitektur. Dinge, die man erst bemerkt, wenn sie fehlen – und die im Kern über Vertrauen entscheiden.

7. Die Reife – Geschwindigkeit und Präzision Messen, optimieren, stabilisieren. Keine Versprechen, keine Showeffekte, sondern echte Performance. Das System wird nicht nur nutzbar, es wird verlässlich.

8. Die Skalierung – Vertrauen im großen Maßstab Geräteverwaltung, Compliance, IoT, maschinelles Lernen. Die Ebene, auf der Unternehmen, Behörden und Bildungseinrichtungen das System im Alltag einsetzen können.

Und so wächst das Projekt nicht in Sprüngen, sondern Schichten. Vom ersten Funken, der zeigt, dass es lebt – bis zum Ökosystem, das getragen wird von Entwicklern, Unternehmen und Institutionen.

7 Community ohne Mauern

„Offenheit ist wertlos, wenn niemand hineinfindet.“

Die meisten Projekte sterben, bevor sie leben – nicht aus Mangel an Ideen, sondern weil sie hinter verschlossenen Türen verharren. Repositories bleiben unsichtbar, Diskussionen unauffindbar, Strukturen undurchlässig.

Open Nexus geht einen anderen Weg. Alles ist von Beginn an sichtbar. Ein GitHub-Repository mit Diskussionsforum und Dokumentation für die Öffentlichkeit. Ein GitLab, in dem die eigentliche Entwicklung stattfindet. Eine Website, die die Vision erklärt. Ein Discord, der die ersten Gespräche trägt.

Kein unsichtbares Labor, sondern ein offener Raum. Und mehr noch: Der erste Prototyp lebt bereits. Nicht poliert, nicht fertig – aber funktional. Proof of Life. Der entscheidende Unterschied zwischen einer Idee und einer Realität.

8 Die richtige Person zur richtigen Zeit

„Bewegungen beginnen nicht mit Komitees. Sie beginnen mit Menschen.“

Große Ideen sind wertlos, wenn niemand den ersten Schritt macht. Die meisten warten – auf Ressourcen, auf Strukturen, auf Erlaubnis. Doch Veränderung beginnt nicht mit Warten. Sie beginnt mit Handeln.

Ich habe Systeme entwickelt, Projekte geleitet, internationale Teams geführt. Ich kenne die Praxis des Bauens ebenso wie die Notwendigkeit der Organisation. Und als ich die Risse im Alten und die Konturen des Neuen sah, habe ich nicht gewartet, bis jemand anders begann. Ich habe begonnen.

Darum existiert dieses Projekt nicht als Folie oder Blogpost, sondern als funktionierender Kern. Nicht als Plan, sondern als laufendes System.

Denn Bewegungen entstehen nie durch die, die zögern. Sie entstehen durch die, die anfangen.

9 Beyond Funding: Mitbauen statt Zuschauen

„Kapital beschleunigt. Menschen gestalten.“

Geld allein baut keine Zukunft. Geld beschleunigt, es kauft Zeit, es kauft Kapazität – aber es ersetzt keine Vision, keine Entschlossenheit, keine Mitstreiter.

Was wir brauchen, sind keine Investoren am Spielfeldrand, die Renditen erwarten. Wir brauchen Mitbauer. Partner, die bereit sind, in Standards zu denken statt in Quartalen. Menschen und Unternehmen, die nicht nur Kapital geben, sondern Verantwortung übernehmen.

Es geht nicht um Schecks. Es geht darum, ob man den Mut hat, Teil einer Bewegung zu werden, die größer ist als jede einzelne Bilanz.

10 Ein nachhaltiger Weg nach vorn

„Offenheit ist kein Idealismus. Sie ist die härteste Form von Strategie.“

Offen und frei heißt nicht beliebig oder naiv. Es heißt skalierbar, überprüfbar, anpassbar – und damit belastbarer als jedes proprietäre Modell.

Darauf baut eine zweite Ebene: Nexus OS. Ein kontrollierter Layer für Organisationen, die Zertifizierungen, Compliance oder Flottenverwaltung brauchen. Nicht, um Mauern zu ziehen, sondern um Verlässlichkeit zu garantieren.

Damit entsteht ein klares Modell:

Unternehmen können zertifizierte Hardware und Treiber einsetzen, ohne Black Boxes zu fürchten.

Bildungseinrichtungen erhalten sichere, kontrollierte Cloud-Umgebungen, ohne sich an externe Monopole zu binden.

Entwickler können frei veröffentlichen oder über einen kuratierten Store vertreiben.

Firmen haben die Wahl, offene oder geschlossene Komponenten zu kombinieren – je nach Bedarf.

Das ist kein Idealismus. Es ist ein Geschäftsmodell. Eines, das die besten Entwickler anzieht. Eines, das mit den Bedürfnissen der Industrie wächst. Eines, das langfristig konkurrenzfähig bleibt – gerade weil es offen ist.

11 Schlussfolgerung

„Das ist der Moment.“

Jede Epoche hat ihre Wendepunkte.

Als der Personal Computer das Labor verließ und auf den Schreibtisch wanderte. Als das Internet die Universitäten hinter sich ließ und die Welt verband. Als das Telefon aufhörte, ein Telefon zu sein, und zum Zentrum unseres Lebens wurde.

Heute stehen wir erneut an einer solchen Schwelle.

Denn die Fundamente, auf denen wir noch immer bauen, sind überlebt: x86 ist ein überkonstruierter Anachronismus. ARM bleibt ein Mietvertrag. Linux erstickt an proprietären Treibern. Android ist eine offene Hülle mit geschlossenem Kern. Apple liefert Perfektion – im Käfig.

Wir können diese Systeme weiterflicken. Wir können uns im Komfort einrichten und die Mauern ignorieren. Oder wir können warten – bis jemand anders die Lücke füllt. Wahrscheinlich nicht aus Europa, sondern aus China. Und dann werden wir uns wieder fragen, warum wir das Offensichtliche übersehen haben – so wie einst bei Bell, IBM, Palm oder Blackberry.

Die Alternative ist klar: ein Fundament, das offen, modular und resilient ist. Werkzeuge, die Entwickler befähigen statt ausschließen. Eine Architektur, die nicht auf Gnade beruht, sondern auf Prinzipien.

Es gibt keinen Mittelweg. Keine elegante Kompromissformel. Entweder wir bauen neu – oder wir leben weiter im Überkommenen.

Dies ist der Moment, in dem sich entscheidet, ob wir noch einmal ein Jahrzehnt verwalten – oder endlich beginnen, die Zukunft zu gestalten.