Micro-Apps und Micro-Services

Eine neue Ära bricht an.

von **Matthias Eger**
https://matthias-eger.de

Stand:

Weiden, 25.02.2026

Matthias Eger
Pfarräckerstr. 14
92637 Weiden

Tel. +49 (0) 961 634 32 61
Mobil +49 (0) 175 585 73 05

Mail mail@matthiaseger.de
Web www.matthias-eger.de

Vorwort

Ich schreibe das nicht aus der Perspektive eines „KI-Fans“, der alles automatisieren will, sondern aus der Perspektive eines Managers für angewandte KI-Transformation, der täglich sieht, was passiert, wenn Menschen KI mit Verantwortung verwechseln.

Es entstehen Mikroapps wie Pilze nach einem warmen Regen. Das geht heute schnell, kreativ, oft ist nützlich und dabei erschreckend häufig gefährlich, teuer und nicht überlebensfähig.

Das ist keine Rechtsberatung und ersetzt keine individuelle Sicherheits- oder Compliance-Bewertung. Es ist ein strategisches, praxisorientiertes Security-Playbook, welches sich an etablierten Rahmenwerken und empirischen Befunden orientiert.

Warum dieser Text unbequem sein muss! ☺

Weil sich gerade eine gefährliche Normalität mit Formel bildet:

„Kannst du mal schnell…?“ + KI-Tool + Low-Code-Plattform = „fertige App“.

Und niemand fragt:

- Wer haftet?
- Wer betreibt?
- Wer patcht?
- Wer löscht Daten?
- Wer prüft Berechtigungen?

Genau diese Lücke beschreibt bereits meine These aus meiner Publikation „KI ersetzt keine Entwickler – sie verschiebt die Verantwortung“.

KI kann Code liefern – aber eben keine Haftung, keine Prioritäten, keinen Betrieb, keine Governance übernehmen. Mal abgesehen davon, dass immer noch die meisten kleinen Unternehmen keine etablierte Risikostrategie geschweige denn Infrastruktur haben, um Kunden im Ernstfall überhaupt zu informieren.

Mikroapps im KI-Zeitalter

Der Begriff „Mikroapp“ wird je nach Anbieter unterschiedlich benutzt. Der Kern besteht aber aus kleiner Scope, klare Aufgabe, geringe Interaktionskosten. In der Praxis sind Mikroapps häufig kleine Prozesse, welche gezielte Aktionen in einem größeren Unternehmenskontext auslösen. Darunter fallen z. B. Genehmigungen, Tickets, Statuswechsel, Reports.

Ein anschauliches, herstellergeprägtes Beispiel liefert Citrix. Mikroapps nutzen APIs aus SaaS-, Web-, Legacy- oder Eigenentwicklungen, um Inhalte/Funktionen bereitzustellen und Tasks zu automatisieren, ohne dass Nutzer „große Anwendungen“ öffnen müssen.

Nicht Mikroapps selbst, sondern der Fertigungsprozess ist neu an der „neuen Ära“.

Drei Kräfte wirken gleichzeitig:

1. Low-Code ist Mainstream und wirtschaftlich relevant. Gartner prognostiziert für 2021 bereits ein Marktvolumen von 13,824.2 Mio. USD und beschreibt Low-Code ausdrücklich als soziale/technologische Bewegung, getrieben durch Remote-Entwicklung, Hyperautomation und „Composable Enterprise“. (Quelle https://www.gartner.com/en/newsroom/press-releases/2021-02-15-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-23-percent-in-2021)

2. Citizen Development skaliert real – nicht als Buzzword. Gartner definiert einen „Citizen Developer“ als Mitarbeiter, der Anwendungen für sich oder andere erstellt, mit Tools, die nicht aktiv durch IT/Business verboten sind. Das ist eine Persona, keine Job-Rolle.

3. KI-Code-Assistenten demokratisieren Implementierung weiter. Gartner erwartet, dass bis 2028 75 % der Enterprise-Software-Engineers KI-Code-Assistenten nutzen werden (von <10 % in 2023), und berichtet zugleich, dass 63 % der befragten Organisationen bereits pilotieren/deployen oder deployed haben.

Dieses Dreieck aus Low-Code-Plattformen + Citizen Developers + KI-Code-Assistenten erzeugt eine logische Konsequenz, dass ein Unternehmen nicht „ein App-Projekt“ bekommt, sondern ein App-Ökosystem. Und Ökosysteme brauchen Regeln, nicht Motivation.

Die meisten Mikroapps scheitern nicht auf der Oberfläche, weil z. B. die „UI unschön“ ist, sondern in den Schichten, über die überhaupt niemand spricht, weil sie nicht viral sind.

Dazu gehören Identity, Data, Interfaces, Betrieb, Audit und eben die Haftung.
Genau dort entstehen Kosten, die man in keiner Demo sieht.

Mikroapps vergrößern die Angriffsfläche vor allem über Identitäten und APIs

Mikroapps sind fast immer API-getrieben. Wenn die „kleine App“ etwas tut, dann ruft sie Systeme auf, bewegt Daten, triggert Workflows. Genau deshalb sind API-Risiken heute keine Spezialdisziplin mehr.

OWASP listet 2023 u. a. „Broken Object Level Authorization“, „Broken Authentication“ und weitere API-Top-Risiken als zentrale Muster (Quelle: https://owasp.org/API-Security/editions/2023/en/0x00-header/?utm_source=chatgpt.com).

Was das in Mikroapp-Realität bedeutet?

Eine Mikroapp, die „nur schnell“ Statusdaten aus einem System holt, kann bei falscher Objekt-Autorisierung zu einem Datenabfluss-Multiplikator werden.

Ein Endpoint, tausend Requests, riesiger Schaden.

Es geht noch härter. Viele erfolgreiche Angriffe sind keine „0-Day-Magie“, sondern Berechtigungs- und Credential-Realität. Die 2025er DBIR-Kommunikation von Verizon ordnet „Basic Web Application Attacks“ stark dem Muster „stolen credentials“ zu. In diesem Pattern wurden ca. 88 % der Breaches mit gestohlenen Credentials in Verbindung gebracht.

Das ist nur unangenehm ehrlich. Wenn du Hunderte Mikroapps zulässt, ohne Identity-Standards (SSO, MFA, Rollenmodell, Secrets-Handling, Logging), baust du nicht Innovation.

Du baust ein Buffet für Angreifer.

Kleine Apps aber mit großen Pflichten

Mikroapps sind oft Daten-„Abkürzungen“. Sie ziehen personenbezogene Daten aus System A, kombinieren sie mit Prozessdaten aus System B und drücken sie in ein Dashboard oder einen Flow. Sobald personenbezogene Daten verarbeitet werden, greifen Grundprinzipien wie Rechtmäßigkeit, Zweckbindung, Datenminimierung, Speicherbegrenzung, genau eben wie explizit im GDPR-Text.

Und „es war nur intern“ schützt nicht automatisch!

Die GDPR-Sanktionslogik reicht bei schweren Verstößen bis zu 20 Mio. EUR oder 4 % des weltweiten Jahresumsatzes, je nachdem, was höher ist.

Parallel wird die Regulierungslandschaft für Cybersecurity und Betriebsresilienz in der EU strenger. Die NIS-2-Richtlinie adressiert Maßnahmen für ein hohes gemeinsames Cybersicherheitsniveau und stärkt Pflichten rund um Risikomanagement und Incident-Handling.

Wer also Mikroapps als „kleine Tools“ behandelt, behandelt Compliance wie „Papierkram“ und wird irgendwann feststellen Compliance ist ein Kostenblock oder ein Existenzthema.

Der „AI Oversight Gap“ und neue Angriffsmuster

Die schnellste Eskalation entsteht dort, wo KI nicht nur baut, sondern auch ausführt, wie bei Agents, automatische Actions, „autonomous workflows“. Der 2025er „Cost of a Data Breach“ Report von IBM (Quelle: https://csrc.nist.gov/pubs/sp/800/218/final?utm_source=chatgpt.com) benennt genau die Lücke, die ich in Unternehmen praktisch beobachte: AI adoptiert schneller als Security/Governance. IBM berichtet u. a. eine globale durchschnittliche Breach-Kostenhöhe von 4,4 Mio. USD und zugleich, dass 63 % der Organisationen keine AI-Governance-Policies hatten, um AI zu managen oder „shadow AI“ zu verhindern.

Das ist nicht nur „Policy-Theater“. Es ist messbar teuer.

IBM nennt 1,9 Mio. USD Kosteneinsparung durch „extensive use of AI in security“ gegenüber Organisationen ohne diese Lösungen, also: Governance + Security-Automation sind Kostenfaktor, aber eben auch Kostenbremse. Inhaltlich zeigt OWASP mit der Top-10 für LLM-Applikationen konkrete Angriffsmuster, die für Mikroapps mit LLM-Komponenten sofort relevant sind: Prompt Injection, Insecure Output Handling, Supply-Chain-Risiken, Model DoS (Kosten-/Verfügbarkeitsangriff), eben unter anderem. Wenn dein Unternehmen „KI-Apps“ baut, ohne Prompt-/Output-Sicherheit, ohne Access-Controls, ohne Observability, dann ist es nur eine Frage der Zeit, bis du entweder Daten verlierst oder Kosten explodieren oder im schlimmsten Fall: beides!

Wartbarkeit ist kein Luxus, sie ist die eigentliche Rechnung am Ende

Sicherheitsvorfälle sind teuer. Aber das stillste Geld verbrennst du über Zeit durch Wartung, Fragmentierung und Technical Debt.

Consortium for IT Software Quality (Quelle: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf?utm_source=chatgpt.com) beziffert die Kosten schlechter Softwarequalität in den USA in 2022 auf mindestens 2,41 Billionen USD und die akkumulierte Software-Technical-Debt-Summe auf ca. 1,52 Billionen USD.

Warum gehört das in ein Mikroapp-E-Book?

Weil Mikroapps, wenn sie ungeregelt entstehen, Debt nicht reduzieren, sondern verteilen. Statt eines Monolithen hast du dann 300 Mini-Monolithen in Teams, mit je unbekannten Owner, unbekannten Datenflüssen, unbekannten Berechtigungen.

Der Preis von „KI schreibt Code“ ist, dass Code stärker geprüft werden muss, aber nicht weniger!

Die Sicherheitsforschung ist hier inzwischen klar genug, um nicht mehr zu diskutieren, ob das ein Problem ist, sondern wie groß. Die Studie „Asleep at the Keyboard? Assessing the Security of GitHub Copilot’s Code Contributions“ untersucht systematisch Bedingungen, unter denen KI-Assistenten unsicheren Code vorschlagen.

Neuere empirische Analysen zu Copilot-generiertem Code in realen GitHub-Projekten fanden ebenfalls „high likelihood of security weaknesses“ und beziffern in einem Datensatz z. B. Anteile betroffener Snippets, u. a. Python, JavaScript. KI senkt die Eintrittshürde, aber nicht das Sicherheitsniveau. Sie verschiebt die Arbeit von „Tippen“ zu „Auditen“. Wer das nicht akzeptiert, spart vorne und zahlt hinten.

Mikroapps als produktive Einheit, wenn man sie erwachsen betreibt

Mikroapps sind nicht „schlecht“. Sie sind eine legitime Antwort auf reale Probleme in der Organisation: Kontextwechsel, überladene Systeme, lange IT-Backlogs, fehlende Automatisierung.

In der Citrix-Logik sind Mikroapps gerade deshalb attraktiv, weil sie repetitiven Alltag reduzieren und Funktionen „dorthin bringen“, wo Nutzer arbeiten, statt Nutzer durch Landschaften voll mit Applikationen zu jagen.

Und es gibt einen zweiten, strategischen Vorteil: Low-Code/Composable-Ansätze passen zu dem, was Gartner als Treiber beschreibt (Hyperautomation, Composable Enterprise, steigender Delivery-Druck). Das ist nicht „nice to have“, das ist reale Wettbewerbsdynamik.

Der entscheidende Unterschied lautet daher nicht „Mikroapps: Ja/Nein“, sondern:

Mikroapps als Produktportfolio
oder
Mikroapps als Shadow-IT-Wildwuchs

Genau hier gilt ein Prinzip aus meiner E-E-A-T-Publikation:

Vertrauen ist das Fundament und entsteht durch Nachvollziehbarkeit, Transparenz und Konsequenz, nicht durch Geschwindigkeit.

Die Strategie: Governance-, Security- und Engineering-Blueprint für Mikroapps

Ich lege hier bewusst kein „Framework-Feuerwerk“ hin. Ich lege eine Strategie hin, die in Unternehmen funktioniert, weil sie konkret ist und sich an etablierten Standards orientiert: NIST SSDF, NIST CSF 2.0, Zero Trust, OWASP.

Die drei Grundsätze, die alles entscheiden

1. **Jede Mikroapp ist ein Systemteil mit realer Tragweite.**
   Wenn sie Daten liest/schreibt, Rechte nutzt oder Prozesse auslöst, ist sie nicht „klein“. Sie ist ein Angriffspfad, ein Compliance-Risiko und ein Wartungsobjekt.

2. **Geschwindigkeit ist kein KPI, wenn Betrieb nicht mitgedacht ist.**
   „Time to Value“ ist nur dann Wert, wenn der Wert nicht durch Incident-Kosten, Audit-Kosten oder Debt-Kosten gefressen wird.
3. **KI ist ein Produktionswerkzeug – kein Governance-Ersatz.**
   Das ist die Kernlinie meiner Publikation zur Operator-Rolle: KI kann liefern, aber nicht führen. Deshalb braucht es Orchestrierung, Dokumentation, Entscheidungsdisziplin.

Risiko-Klassen statt Bauchgefühl (Mikroapp-Gate-Modell)

Du brauchst ein einfaches, hartes Gate.

Kein Theater, sondern vier Klassen und jede Klasse hat Minimalanforderungen:

- **Klasse A – Persönliche Produktivität (lokal, keine sensiblen Daten, kein System-Write):** Erlaubt mit Baseline-Policies (Accountability/Owner, Inventar, Logging light).
- **Klasse B – Team-App (interne Daten, begrenzte User-Gruppe, moderate Prozesswirkung):** SSO/MFA, Rollenmodell, Data-Classification, Audit-Logs, Backup/Export, definierte Decommission-Regel.
- **Klasse C – Business-kritisch (Write in Kernsysteme, relevante Entscheidungen, operativer Impact):** Secure SDLC (SSDF-aligned), Test-Automatisierung, Security-Scanning, Incident-Runbooks, Monitoring/SLOs, Change-Management.
- **Klasse D – Extern oder regulatorisch sensibel (Kunden/Dritte, hohe personenbezogene Datenlast, regulierte Prozesse):** DPIA/DSFA-Logik wo nötig, starke Governance (inkl. Auditfähigkeit), Contracting/Third-Party-Risk, strengste Data-Minimization und Zweckbindung, plus Compliance-Mapping (GDPR, ggf. NIS2/DORA, ggf. AI-Act-Relevanz).

Dieses Gate-Modell ist nicht „Bürokratie“. Es ist ein Übersetzungsmechanismus zwischen Innovationsdruck und Sicherheitsrealität – genau das, was NIST CSF 2.0 als Zielsetzung beschreibt: Risiken verstehen, priorisieren, kommunizieren.

Was jedes Unternehmen sofort festziehen muss, die Minimum-Viable-Governance

Wenn du nur eine Seite mit Regeln einführen darfst, dann diese:

Jede Mikroapp braucht

- einen Owner (Person, nicht Team),
- einen Zweck, eine Datenklassifikation,
- eine Berechtigungslogik,

- einen Lifecycle-Status (Pilot/Prod/Deprecated) und
- eine Abschaltregel.

Ohne das ist es Shadow-IT per Definition.

Für Low-Code-Plattformen existieren dafür praxistaugliche Governance-Referenzen.

Beispiel: Microsoft beschreibt das Power Platform CoE Starter Kit als Referenz-Implementierung für Governance, Monitoring und Adoption, und betont zugleich: Ein CoE ist nicht „nur Tools“, sondern braucht People/Process/Requirements. (Quelle: https://learn.microsoft.com/en-us/power-platform/guidance/coe/overview)

Tools sind nie Governance; Tools sind Telemetrie und Durchsetzung.
Governance ist Entscheidung und Verantwortungsstruktur.

Die Leitplanke, die Mikroapps überleben lassen, nennt sich Minimum-Viable-Security

Ich binde das bewusst an etablierte, auditierbare Quellen, die ich recherchiert habe:

- **Secure SDLC nach NIST SSDF:** NIST SP 800-218 beschreibt einen Kernsatz sicherer Entwicklungspraktiken, die in SDLCs integrierbar sind, um Vulnerability-Risiken zu senken.
- **Secure-by-Design-Prinzipien:** Cybersecurity and Infrastructure Security Agency fasst „Secure-by-Design“ als Hersteller-/Anbieterprinzip zusammen und betont Ownership, Transparenz und Verantwortung für Security-Outcomes. Auch wenn das primär an Hersteller adressiert ist, ist die Logik für interne Plattformteams identisch: Wer eine interne Entwicklungsplattform betreibt, ist faktisch Hersteller für interne Apps.
- **Zero Trust:** NIST SP 800-207 beschreibt Zero-Trust-Architektur als Paradigmenwechsel von Perimeter-Denken zu Fokus auf User/Assets/Resources. Genau das ist Mikroapp-Realität (viele kleine Ressourcen, viele Identitäten, viele Policies).

Keine App ohne SSO/MFA, keine Secrets in Klartext, keine direkten DB-Credentials in Flows, keine unklassifizierten Connector-Wildwüchse, keine externen Freigaben ohne Security-Review, kein Go-Live ohne Logging.

Die harten Regeln gegen naive Prompt-Romantik

Wenn KI Code/Flows/UI erzeugt, behandle Output wie **Third-Party-Code**. Das ist keine Meinung, das ist eine Konsequenz aus empirischen Befunden zu unsicheren Vorschlägen und Code-Weaknesses in KI-Outputs.

Und wenn du LLMs in Mikroapps integrierst, musst du OWASP-LLM-Risiken operationalisieren, nicht diskutieren: Prompt Injection, Insecure Output Handling, Supply Chain, Model DoS sind keine „Edge Cases“, sondern dokumentierte Risikoklassen.

Das minimale Kontrollset für KI-gestützte Mikroapps:

- **Prompt-/Policy-Trennung**
  System-Policies sind nicht „Text“, sondern Policy-Artefakte) und Versionierung der Prompts/Policies
- **Output-Validation**
  z. B. strukturelle Validierung, erlaubte Aktionen, Escaping, „Insecure Output Handling“ ist ein Risiko
- **Kosten-/Rate-Guards**
  Model DoS ist eine OWASP-Klasse;
  das ist auch ein Kostenangriff
- **Observability**
  Wer hat wann welche KI-Aktion ausgelöst,
  welche Daten wurden berührt (Auditfähigkeit)
- **Klare Human-Override-Punkte**
  Gerade bei riskanten Aktionen, z. B. Freigaben, Exporte, Löschungen

Ohne Operations keine Mikroapp, sondern ein Experiment

ENISA ordnet 2024 mehrere zentrale Bedrohungsfelder ein (u. a. Verfügbarkeit, Ransomware, Datenbedrohungen) basierend auf Tausenden öffentlich berichteten Sicherheitsereignissen. Das ist keine theoretische Bedrohung, das ist das Umfeld, in dem Mikroapps laufen müssen.

Mikroapps brauchen ein „Mini-Produktionssystem“: Monitoring, Incident-Pfad, Ownership, Patch-/Change-Rhythmus. Ohne das kannst du nicht seriös behaupten, dass du „Wert geschaffen“ hast, du hast nur „Code veröffentlicht“.

Verantwortung orchestrieren statt Apps feiern

Ich beziehe mich hier bewusst auf meine eigene Verantwortungs-Linse aus meiner Publikation „Der Fluss in den Nebel“. Das Modell beschreibt drei Verantwortungslogiken (Komfort/Stabilität, Gestaltung und systemische Verantwortung) und betont:

Mehr Tragweite heißt mehr Komplexität, aber nicht „mehr Wert“.

Die Mikroapp-Explosion ist psychologisch und organisatorisch ein Zone-2-Phänomen: Menschen gestalten. Sie automatisieren. Sie bauen. Und das ist gut.

Das Problem entsteht, wenn Zone-2-Energie ohne Zone-3-Rahmen skaliert.

Dann kippt Gestaltung in systemisches Risiko. Dann werden Mikroapps zu Datenpfaden ohne Verantwortung, zu Steuerungslogik ohne Audit, zu Workflows ohne Sicherheitsmodell.

Genau hier positioniere ich mich nicht als „noch ein Tool“, nicht als „Prompt-Coach“, sondern als jemand, der Mikroapps auf Schienen setzt:

- Governance so, dass Innovation nicht erstickt,
  aber auditierbar wird (CoE-Denke: People/Process/Tools)

- Security so, dass du nicht auf Glück angewiesen bist
  (SSDF, Secure-by-Design, Zero Trust, OWASP)

- KI so, dass sie Produktivität hebt, ohne Kontrolle zu zerstören
  (OWASP-LLM-Risiken + empirische Evidenz zu unsicheren Code-Outputs)

Das ist unbequem! Ja, klar.

Denn es bedeutet, das Lieblingsnarrativ dieser Zeit zu zerstören:

„Jeder kann jetzt Software bauen,
also brauchen wir keine Struktur mehr.“

Doch genau das Gegenteil ist wahr!

Je leichter das Bauen wird, desto wichtiger wird derjenige, der Bauen in Betrieb übersetzt.

Bereit, KI richtig einzusetzen – statt dich von ihr überrollen zu lassen?

Wenn du diesen Text bis hierhin gelesen hast, dann hast du vermutlich erkannt,
dass der Unterschied zwischen *„KI nutzen“* und *„KI beherrschen“* riesig ist.
Genau da setze ich an.

Ich helfe dir, aus Chaos wieder Kontrolle zu machen –
ob als Unternehmen oder als Einzelperson mit einer Idee, die einfach funktionieren soll.
Gemeinsam bringen wir Struktur in dein Projekt, schaffen Klarheit über nächste Schritte und sorgen dafür, dass KI für dich arbeitet – nicht gegen dich.

📩 **Melde dich jetzt unverbindlich über mein Kontaktformular:**
👉 **Kontakt aufnehmen**

Oder direkt über https://magmacore.software, wenn du wissen willst, wie Magmacore© dein Informationsmanagement auf das nächste Level bringt.

Ich freue mich, von dir zu hören –
Matthias Eger
Operator. Entwickler. Möglichmacher.

Matthias Eger Design Studio
Inhaber Matthias Eger
Pfarräckerstr. 14
92637 Weiden

Tel. +49 (0) 961 634 32 61
Mobil +49 (0) 175 585 73 05

Mail mail@matthiaseger.de
Web www.matthias-eger.de