

AI Low Code Plattform

A12 Backstage Session - September 2026

Sneak-Peek in ausgewählte Entwicklungsstränge

openCode 

GitHub 

Grundlegende Informationen zu A12

<https://www.a12.ai>

Was ist A12?

A12 ist eine Open Source-Plattform für die Entwicklung von Unternehmensanwendungen in komplexen IT-Landschaften. Sie setzt auf modellbasiertes Software Engineering (MDSE) und erschließt das Low Code-Prinzip für die Welt der Enterprise Software. Als offene Plattform vereinfacht A12 die Integration von Best-of-Breed-Lösungen und den Einsatz von KI auf allen Ebenen. Die Modellierungsumgebung von A12 stellt Werkzeuge be-

reit, um Teile einer Anwendung ohne Programmierkenntnisse zu erstellen und als unabhängige Geschäftslogik-Module langfristig zu pflegen. Die Laufzeitplattform von A12 bietet die nötige Flexibilität, um geschäftskritische Applikationen mit professioneller Individualsoftwareentwicklung, KI-Unterstützung und Systemintegration zu voll integrierten Unternehmensanwendungen zu entwickeln.

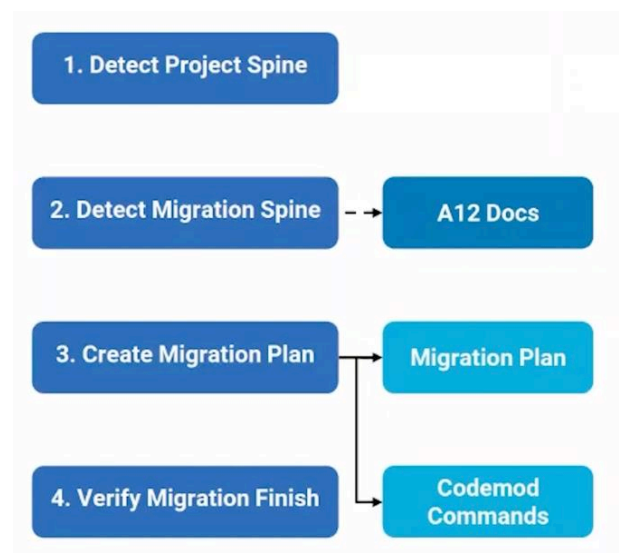
CLAUDE

OPENCODE



Agentic Engineering – Verbesserte Migrationsunterstützung, Plugins für OpenCode und mehr

Das Tooling-Angebot für den Einsatz von KI-Agenten rund um A12-Projekte wächst. So wurde die Unterstützung für Migrationen ausgebaut. Das seit Juli bestehende Migrations-Plugin für Claude Code hilft jetzt nicht nur bei Major-Releases, einen Migrationsplan aufzustellen, Codemod-Befehle zu generieren und die Migration auf Vollständigkeit zu prüfen. Es unterstützt nun auch Minor-Release-Migrationen wie von Version 2026.06-ext0 auf 2026.06-ext1. In kleineren Projekten wurden Migrationen bereits komplett autonom durchgeführt. Ergänzend zu den Claude Code-Plugins gibt es künftig auch Plugins für OpenCode. Damit wird es möglich, selbstgehostete LLMs für KI-Agenten zu nutzen – eine Voraussetzung für digital souveräne Entwicklungsprozesse. Neben der Dokumentation von A12 wurde das A12 Discourse-Forum in den a12-knowledge-support integriert. KI-Agenten können so auf das gesamte Wissen der Community zurückgreifen. Seit der Bereitstellung der ersten Plugins entwickelt mgm Benchmarks, um die Zeit, Qualität und Kosten von KI-Tasks zu bewerten. Neben den umfangreichen Benchmarks für Modellierungsaufgaben sind Benchmarks für weitere Aufgaben geplant – z. B. rund um die Architektur der Anwendung. Neue Benchmarks orientieren sich dabei am User Feedback.



Business Scripting – Erste experimentelle Version mit Full-Stack-Support

Mit Business Scripting erschließen sich für Fachexperten und Business Analysten ergänzend zur Modellierung weitere Funktionalitäten von A12 – und zwar in Form von kleinen Skripten in TypeScript mit fixer Struktur und einer speziell dafür entworfenen API. Seit Juni ist eine erste Version, die zunächst Backend- und Workflow-seitige Operationen unterstützt, mgm-intern verfügbar im Rahmen der Forschung und Entwicklung. Mit dem A12-Release 2026.06-ext2 wird diese interne Entwicklungsversion um den Frontend-Bereich erweitert. So ist es mit den Skrip-

ten zum Beispiel auch möglich, Aktionen für den Klick eines Buttons zu definieren. Das Business Scripting bietet damit erstmals die Unterstützung für den gesamten Full Stack. Darüber hinaus wird die Backend-seitige Unterstützung des Business Scripting bereits Teil des regulären Releases 2026.06-ext2. Es steht als experimentelles Feature allen Interessenten zur Verfügung, die es testen und Feedback geben möchten. Die experimentelle Full-Stack-Version wird vrs1. Anfang 2027 Teil des Releases 2026.06-ext4.

```
1 import type { Script, ScriptContext } from "@com.mgmt.a12.client/client-scripting";
2 import type { ScriptClientApi } from "@com.mgmt.a12.client/client-scripting/client-api";
3
4 type Args = { contractId: string };
5
6 export const openContractDetail: Script<Args, string> = {
7   name: "openContractDetail",
8   description: "Opens the selected contract in its own detail form.",
9
10  async main({ api, args }: ScriptContext<ScriptClientApi, Args>) {
11    const activity = await api.activity.launcher();
12    if (!activity) return "no activity to open from";
13
14    const result = await api.activity.startSub(
15      activity.id,
16      { model: "Contract_DM", instance: args.contractId },
17      { views: [{ name: "ContractDetailForm" }] },
18    );
19
20    return result.outcome;
21  },
22 };
```

Markdown-Support: Rich Text für Menschen und KI-Agenten

Um eine einheitliche Unterstützung für formatierte Texte zu bieten, führt A12 die durchgängige Unterstützung für die Auszeichnungssprache Markdown ein. Dadurch wird es für Menschen künftig noch einfacher, Informationen strukturiert und verdichtet in A12-Anwendungen zu editieren und zu sichten. Da auch LLMs sehr gut mit Markdown umgehen können, eröffnen sich zudem neue Möglichkeiten für KI-Agenten, Informationen aus Feldern zu lesen und in Felder zu schreiben. Neben zwei neuen Widgets – Core Markdown Widget und Markdown Area Form

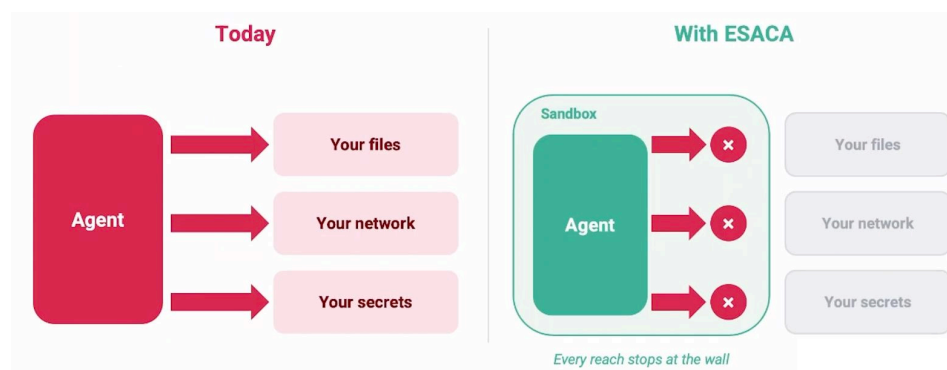
Widget – sind Integrationen in die Form und Content-Engine in Arbeit. Das Core Markdown Widget baut als Standalone React-Komponente auf dem Open Source Text Editor Framework Lexical auf. Es unterstützt die Modi "view", "edit" sowie "source" und ist durch ein Plugin SDK erweiterbar. Für die Zukunft sind weitere Ausbaustufen geplant – zum Beispiel für die Unterstützung kollaborativer Editiermöglichkeiten in Echtzeit sowie die Integration mit der A12 Print-Engine.



ESACA: Agentic Coding mit europäischer Datenhoheit

Coding-Agents beschleunigen die Softwareentwicklung, sind aber sicherheitstechnisch heikel: Wer einen Agenten auf der Host-Maschine laufen lässt, gibt ihm vollen Zugriff auf das Dateisystem, das Netzwerk und hinterlegte Secrets. Genau hier setzt ESACA an – die [European Sovereign Agentic Coding Appli-ance](#). Der Agent läuft in einer Sandbox, isoliert vom Host. Keine Daten verlassen die Sandbox unkontrolliert: Jeder ausgehende Request läuft über einen Egress-Proxy mit Allowlist nach dem Default-Deny-Prinzip. Tokens und Zugangsdaten verbleiben in einem Secret Vault auf der lokalen Maschine. Sie gelangen nie in die Sandbox selbst, sondern werden erst zur Laufzeit durch den

Proxy in die ausgehenden Requests gebracht. Code, Zugangsdaten und Daten bleiben unter Kontrolle. ESACA funktioniert mit verschiedenen Coding-Agents - initial unterstützt werden Claude Code und OpenCode. A12 wird nativ unterstützt: Wenige Klicks erzeugen eine fertig konfigurierte Sandbox mit den A12-Plugins, dem A12 Knowledge MCP und passenden A12-Egress-Regeln. ESACA gibt es als Desktop-App und als CLI, für Windows, macOS und Linux. Beide erzeugen dieselbe Sandbox. Teams können zwischen grafischer Oberfläche und Kommandozeile wählen



A12 App Factory: Von der Infrastruktur zur kompletten Projektumgebung

A12-Anwendungen schneller und einfacher in die Wertschöpfung bringen – das ist das Kernziel der A12 App Factory. Mit einer Reihe neuer Features deckt die Infrastruktur- und Verwaltungsplattform jetzt den gesamten Weg von der Geschäftsmodellierung bis zum skalierbaren Deployment ab. Während bisher Einzelanwendungen im Mittelpunkt standen, rückt künftig die Verwaltung ganzer Anwendungslandschaften in den Fokus – inkl. sicherer KI-Unterstützung auf Basis der ESACA-Sandbox. Schon länger lassen sich per Knopfdruck auf Basis des A12 Projekt-Templates (mit oder ohne Workflows) Umgebungen für Entwicklung und Qualitätssicherung anlegen, Zugangsberechtigungen für Teammitglieder konfigurieren, Build-Prozesse steuern und der Status der deployten Umgebungen

kontrollieren. Neu hinzu kommen Produktions-Deployments, individuelle Projekt-Templates und eine webbasierte Instanz des Simple Model Editors für die Modellierung; mit GitLab wird neben Bitbucket ein zweites Repository unterstützt, wovon besonders Open-Source-Projekte profitieren. Noch in diesem Jahr folgt eine webbasierte IDE für klassische Entwicklungsarbeiten: Teams können dann komplett in der Cloud-Umgebung arbeiten; Konfigurationen an lokalen Maschinen werden überflüssig. Für 2027 stehen erweiterte KI-Funktionen auf der Roadmap, etwa zur Optimierung des Betriebs, sowie neue Governance-Funktionen für die Anwendungslandschaft – zum Beispiel für die Einrichtung eines Datenkatalogs und die modellgetriebene Pflege von Datenflüssen zwischen Anwendungen.

Im Auftrag des Nutzers: Wie UAA mit Personal Access Tokens und OAuth Discovery Maschinen sicher angemeldet

Bislang kannte UAA zwei grundlegende Arten der Authentifizierung: die nutzerinteraktive Anmeldung über OAuth2, LDAP/Local oder SAML sowie die Anmeldung ohne Benutzeroberfläche per API-Key oder Zertifikat. Mit der wachsenden Verbreitung von KI-Agenten rückt jedoch ein dritter Fall in den Vordergrund: Wie meldet sich eine Maschine an, die im Namen eines bestimmten Nutzers handeln soll? UAA schließt diese Lücke mit zwei neuen Features. Das Personal Access Token (PAT) ist ein neuer Authentifizierungstyp: ein langer, zufällig erzeugter String, den der Nutzer selbst generieren und an einen Client übergeben. Der Zugriff bleibt dabei auf ein per Whitelist definiertes Set an Endpunkten beschränkt, und der Token verfällt entweder nach festgelegtem Zeitplan oder wird bei Bedarf widerrufen. Die zweite Neuerung, OAuth Discovery, geht den umgekehrten Weg: Statt dass der Nutzer ein Token erzeugt und weitergibt, holt sich der Client den Zugang selbst. Dafür veröffentlicht der Server unter einer standardisierten Adresse maschinenlesbare Metadaten, aus denen der Client erfährt, welcher Identity Provider zuständig ist, wo die Endpunkte für Autorisierung und Token-Ausgabe liegen und welche Verfahren unterstützt werden. Auf dieser Basis startet der Client den Login-Flow, öffnet ein Browserfenster für die Anmeldung und erhält anschließend ein kurzlebiges, an den jeweiligen Nutzer gebundenes Token. So lassen sich Clients wie Claude oder Cursor nahtlos einbinden, ohne dass dauerhaft gültige Zugangsdaten im Umlauf sind.

New token

Token name

Valid for (days)

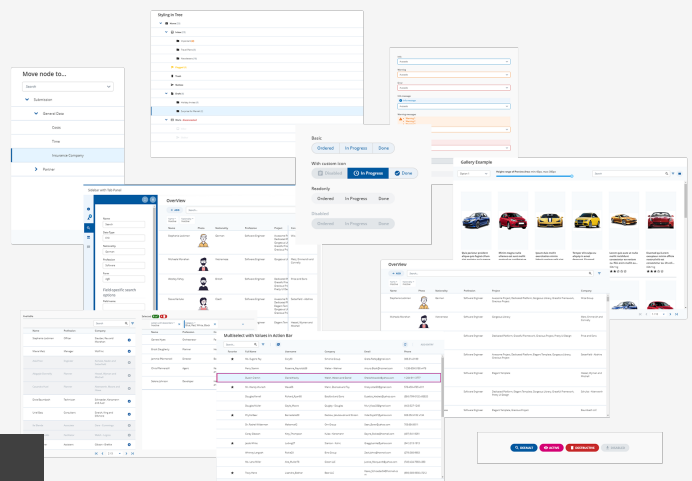
30

CANCEL

GENERATE TOKEN

A12 Widgets

A12 Widgets ist eine umfassende Sammlung von vorgefertigten Komponenten, die eine ansprechende und intuitive Benutzererfahrung ermöglichen.



A12-Widget-Showcase
<https://www.mgm-tp.com/a12.htmlshowcase/>