



BAMA und REA: Mac-Apps nachbauen *mit Claude*

Die Links zu BAMA und REA, wie du beide installierst, was sie wirklich können, was sie kosten und wo die rechtliche Grenze liegt. Plus ein Ausblick auf ArtCraft, das Thema meines nächsten Reels.

// 01

Auf einen Blick

FRAGE	KURZ GESAGT
BAMA	21 Skills für Claude Code und Codex: Die App ansehen, jeden Screen fotografieren, einen Plan schreiben, nativ in Swift nachbauen.
REA	Ein Werkzeugkasten für deinen KI-Agenten: Er untersucht eine App ohne ihren Quellcode und erklärt, wie ein Feature funktioniert. Über 414.000 Sterne auf GitHub.
Kosten	Beide sind Open Source und kostenlos. Für Claude Code brauchst du ein bezahltes Claude-Abo, für das Klonen nativer Apps Claude Pro oder Max.
Grenze	Nur eigene oder erlaubte Software, Forschung und Kompatibilität. Keine fremden Marken, Icons oder Designs 1:1 übernehmen.

// AUS DEM VIDEO



Was das Video nicht sagt

- **ArtCraft hat nichts mit REA zu tun.** Es sind zwei getrennte Projekte. ArtCraft hat seine Apps nicht „kopiert“, sondern als **Clean-Room-Nachbau** neu geschrieben: eigener Code, gebaut aus öffentlichen Spezifikationen und aus Beobachtung. Das ist ein großer Unterschied, auch rechtlich.
- **„Kostenlos“ stimmt nur halb.** REA und BAMA kosten nichts. Der Agent, der damit arbeitet, schon: Claude Code läuft nur mit einem bezahlten Claude-Abo oder über die API. Und BAMA's Klon-Schritt braucht für native Mac-Apps Computer Use, also Claude Pro oder Max.
- **BAMA ist brandneu.** Das Repo ist seit dem 24.10.2026 online, also eine Woche alt (Stand 31.10.2026). Rechne mit Fehlern und häufigen Updates.
- **BAMA braucht macOS 26.** Dazu Xcode 26 für den Swift-6.2-Compiler. Öffnen musst du Xcode nie, installiert sein muss es trotzdem.
- **Die Baupläne sind keine geklonten Fremd-Apps.** Dave Ebbelaar hat die 12 Specs aus seiner eigenen Menüleisten-App „Onyx“ herausgelöst. Sie beschreiben, was eine App tun soll, nicht wie eine fremde App aussieht.
- **Die Sternzahl von REA ist ungewöhnlich.** 414.956 Sterne in gut sechs Monaten, aber nur rund 1.600 Beobachter. Ob alle Sterne echt sind, kann niemand von außen prüfen. Die Zahl ist die, die GitHub anzeigt.



DER MENTOR SAGT

Nachbauen heißt verstehen und neu schreiben. Kopieren heißt Ärger. Bleib auf der ersten Seite.

Die Links

BAMA • daveebbelaar/bama

Build Any Mac App: 21 Skills, die eine native Mac-App vom Screenshot bis ins Dock bringen.

[github.com](https://github.com/daveebbelaar/bama) ➔

REA • morluto/rea

Reverse Engineer Anything: dein Agent untersucht Apps bis auf Binär-Ebene, mit Belegen.

[github.com](https://github.com/morluto/rea) ➔

REA installieren

Du brauchst Node.js 22 (oder neuer) und npm. Dann im Terminal:

```
npx rea-agents setup
```

Wähle deine Agenten aus (Claude Code, Codex, Cursor, Gemini CLI ...), prüfe die vorgeschlagenen Änderungen und bestätige. Danach den Agenten neu starten. Für die Analyse nativer Programme braucht REA zusätzlich ein Analyse-Werkzeug wie Ghidra (kostenlos), Hopper oder IDA. JavaScript- und Electron-Apps gehen ohne.

BAMA in Claude Code installieren

```
/plugin marketplace add daveebelaar/bama  
/plugin install bama@bama
```

Danach `/bama` eingeben: Der Router schaut in dein Projekt und wählt den nächsten Skill. Für den Klon-Schritt Computer Use einschalten: in Claude Code über `/mcp` → `computer-use` → Enable, in der Claude-Desktop-App unter Einstellungen.

Einen Bauplan bauen

- 1. Einen Spec aus dem Ordner `specs/` im BAMA-Repo in einen leeren Ordner kopieren, zum Beispiel `window-snapper.md`.
- 2. Claude Code in diesem Ordner starten.
- 3. Diesen Satz schicken:

```
Build the app in window-snapper.md with BAMA.
```

// 04

BAMA und REA im Vergleich

PROJEKT	WAS ES MACHT	FÜR WEN
REA	Versteht fremde Apps: Funktionen, Aufrufe, Daten, mit Belegen	Entwickler, Sicherheitsforscher, Leute, die eine Schnittstelle bauen
BAMA	Baut neue native Mac-Apps aus Idee, Spec oder Screenshots	Indie-Entwickler, Gründer, Power-User mit Mac

Die 12 Baupläne in BAMA

BAUPLAN	WAS DIE APP MACHT
Window Snapper	Fenster per Tastenkürzel auf Hälften oder Drittel legen
Window Switcher	Mit Option-Tab jedes Fenster mit Live-Vorschau wählen
Launcher	Apps und Dateien aus einem Suchfeld öffnen
Clipboard History	Alles Kopierte durchsuchen und einfügen
Text Snippets	Kurzes Kürzel tippen, ganzer Text erscheint
Color Picker	Farbe irgendwo am Bildschirm aufnehmen und speichern
Screenshots	Bildschirm, Bereich oder Fenster aufnehmen und markieren
Screen Recorder	Bildschirm als MP4 mit Mikro, Systemton und Webcam
Menu Bar Stats	CPU, Speicher, Netz und Akku in der Menüleiste
Toolbox	Mac wach halten, versteckte Dateien zeigen, Audio umschalten
Creator Tools	Tastendruck-Overlay, Cursor-Highlight, Teleprompter
Dev Toolkit	Projekte öffnen, JSON und Base64 umwandeln, Port freigeben

// 05

Was es wirklich kostet

TEIL	PREIS
BAMA, REA	kostenlos, Open Source (MIT)
Claude Code	nur mit bezahltem Claude-Abo (Pro, Max, Team) oder API-Guthaben
Klonen nativer Mac-Apps	Computer Use: Claude Pro oder Max
Mac	macOS 26 und Xcode 26 (kostenlos im App Store)
Analyse-Werkzeug für REA	Ghidra kostenlos, Hopper und IDA kostenpflichtig

Ein Tipp gegen Überraschungen: Starte mit dem kleinsten Bauplan (Window Snapper). Laut BAMA ist jeder Spec „an afternoon of agent time“. Das zieht spürbar an deinem Nutzungslimit.

Prompts zum Kopieren

Deine eigene App als native Mac-App nachbauen (BAMA):

Klone meine eigene Web-App [URL] als native Mac-App mit BAMA. Mach von jedem Screen einen Screenshot, schreib die User Stories und bau Milestone 1.

Erst verstehen, dann bauen (REA):

Untersuche mit REA, wie [Funktion] in [App, die mir gehört oder die ich analysieren darf] funktioniert. Zeig mir die Belege und erkläre es in einfachen Worten.

Ein Feature für dein eigenes Projekt:

Ich mag, wie [App] [Funktion] löst. Beschreibe das Verhalten nur aus der Bedienung, ohne Code, Texte, Icons oder Design der App zu übernehmen, und schreib daraus User Stories für meine eigene App [Name].

Eine App aus einer Idee statt aus einer Vorlage:

/bama Ich will eine kleine Menüleisten-App, die [Aufgabe]. Schreib zuerst ein App-Brief, dann die Stories in Milestones.

Den Bauplan anpassen:

Lies window-snapper.md. Ändere den Namen in [Name], die Tastenkürzel auf [Kürzel] und streiche alles, was nicht in Version 1 muss. Dann bau es mit BAMA.

Wer davon profitiert

WER	PROBLEM	WAS DU MACHST
Indie-Entwickler	Neue Mac-App dauert Wochen	Idee als Spec, BAMA baut den ersten Milestone
Gründer	Klick-Prototyp sieht nicht echt aus	Native App für die Demo, mit echten Daten lokal
Agentur	Kunde will eine kleine interne Mac-App	Kundenbriefing als Spec, Bau in Commits, sauber nachvollziehbar
Creator	Werkzeug, das genau zum eigenen Workflow passt	Teleprompter oder Kapitelmarken als eigene App
Entwickler mit Altlast	Eigenes altes Tool ohne Quellcode	Mit REA verstehen, mit BAMA neu bauen

Was du daraus machen kannst

- **Ein Service:** kleine native Mac-Tools für Teams bauen, die ein bestimmtes Problem haben (Menüleisten-Helfer, interne Werkzeuge).
- **Content:** eine Serie „Ich baue eine Mac-App in einem Nachmittag“, jede Folge ein Bauplan.
- **Dein eigener Spec:** schreib deine App-Idee im BAMA-Format (Brief, How it works, Stories, Gotchas). Damit wird jede Idee baubar.
- **Lernen:** lass REA erklären, wie eine eigene oder offene App ein Problem löst, bevor du es selbst baust.

Grenzen und Risiken

- **Recht:** REA ist laut eigenem Hinweis nur für legale Forschung, Bildung, Interoperabilität, Archivierung und Software, die dir gehört oder die du untersuchen darfst. Viele Lizenzverträge (EULA) verbieten Reverse Engineering. Lies sie vorher.
- **Marken und Design:** Namen, Logos, Icons, Texte und das genaue Aussehen fremder Apps sind geschützt. Nachgebaute Funktionen mit eigenem Design sind etwas anderes als eine Kopie. Für die Rechte bist du selbst verantwortlich.
- **Datenschutz:** Beim Klonen sieht der Agent alles in der App. BAMA empfiehlt ein Demo-Konto mit Beispieldaten.
- **Sicherheit:** Du installierst fremden Code und gibst Computer Use Zugriff auf deinen Bildschirm. Nur aus den offiziellen Quellen installieren, REA warnt selbst vor Kopien mit ähnlichem Namen.
- **Reife:** BAMA ist eine Woche alt. Nicht jeder Bau klappt beim ersten Mal, und Updates kommen oft.

Deine erste Stunde

- ☐ Prüfen: läuft dein Mac mit macOS 26? Xcode 26 aus dem App Store installieren.
- ☐ BAMA in Claude Code installieren (zwei Befehle oben).
- ☐ Den Window-Snapper-Spec in einen leeren Ordner kopieren.
- ☐ Den Satz „Build the app in window-snapper.md with BAMA.“ schicken und Milestone 1 bauen lassen.
- ☐ Die fertige App aus dem Dock starten und testen.
- ☐ Optional: REA einrichten (Befehl oben) und an einer eigenen oder offenen App ausprobieren.

// 11

Mein Tipp

Fang nicht mit „klone App X“ an, sondern mit deiner eigenen Idee als Spec. Das ist rechtlich sauber, und du lernst, wie BAMA denkt: Brief, Stories, ein Commit pro Story. Wenn das sitzt, baust du in einem Nachmittag Werkzeuge, für die du sonst jeden Monat zahlst.

Alle Links, Befehle und Zahlen am 31.10.2026 auf den offiziellen Seiten geprüft.

// 12

Als Nächstes: ArtCraft

Im Video kurz erwähnt, im nächsten Reel ausführlich: Das ArtCraft-Team baut mit KI-Agenten Open-Source-Apps als Clean-Room-Nachbau, zum Beispiel PhotoCraft als Neubau von Photoshop, dazu Apps für Fotos, Videoschnitt, Vektorgrafik und PDF. Wer nicht warten will: getartcraft.com.



DER MENTOR SAGT

Erst REA, jetzt BAMA, und bald ArtCraft. Ich bin gespannt, was du baust.

— @reposprites

Du willst auch so animierte Videos machen? Dann besuch meine Webseite und hol dir die Reel-Engine.

reposprites.com ↗

@reposprites • reposprites.com

BAMA