# MadEasy AI Browser – V3.00 Vibecoding Platform & Ecosystem

**Scope:** Bygg en innebygd, selvstendig **Vibecoding Multi‑Agent Platform** i MadEasy (Windows & Android/WASI), med eget utviklingsmiljø (editor/runner/preview/QA/git), et marketplace for plugins/profiler/templat­er og en Community Hub for deling og samarbeid.

**Elevator:** “Replit‑opplevelse inni nettleseren – men med et spesialisert AI‑team som leder prosjektet fra idé til deploy.”

## 0) Oppsummering (MVP → V1)

* **MVP:** Team‑roller (Leader/PM/Architect/Engineer/Data Analyst), lokal sandbox, Monaco‑editor, Node/Python runner, preview, QA‑gate, Git/PR, Vibe‑profiler v3, inter‑agent meldingsskjema, Policy Guard.
* **V1:** Multi‑stack (Next.js, FastAPI, .NET), WASI på Android, testmatrix, metrics‑SDK dashboards, Marketplace (plugins/profiler), Community Hub (playbooks/prosjekter), SLA‑klar.

## 1) Hva er nytt vs. V2

* **Egen plattform:** uavhengig av Replit/Lovable/Bolt/mgx – kjøres lokalt (og valgfritt i skyen).
* **Agent‑team m/ kontrakter:** klare objective functions, RACI og forhandlingsprotokoll.
* **Vibe‑profiler v3:** helhetlige rammer for stack, stil, kvalitetsbudsjett, sikkerhetskrav.
* **Sandbox‑sikkerhet:** kapabilitetsbasert (WASI/OCI), ressurskvoter, nettpolicy.
* **Marketplace 3.0:** typer: Tool‑plugins, Agent‑plugins, Profile Packs, Project Templates.
* **Community Hub 3.0:** squads, code‑review crews, badges/poeng, konkurranser.

## 2) Arkitektur (tekstlig)

User ──┬── Team Leader (Orchestrator)  
 ├── Product Manager (Stories/Acceptance)  
 ├── Architect (ADR/Structure)  
 ├── Engineer (Code/Patch/Test)  
 └── Data Analyst (Metrics/Dashboards)  
 │  
 └─ Tools: FS • Editor • Runner • Preview • QA • Git • Policy • Metrics • Secrets  
  
Runtime:  
 Windows: LocalSandbox.Win / WSL2Sandbox (Linux toolchain)  
 Android: WASI Sandbox (+ optional Cloud Sandbox)  
  
Ecosystem:  
 Marketplace (Plugins/Profiles/Templates) • Community Hub (Projects/Playbooks/Crews)

## 3) Agent‑roller & kontrakter (RACI + Objective Functions)

**Leader:** orkestrerer plan→deploy, eier risiko/avveininger.  
**PM:** verdi/akseptkriterier, backlog, DoD.  
**Architect:** rammeverk/struktur, ADR, cross‑cutting (i18n, a11y, perf/security).  
**Engineer:** patch/implementasjon, tester, fikser bygg/QA.  
**Data Analyst:** hendelser, målinger, dashboard, eksperimenter.

**Objective functions (eksempler):** - PM: maksimer «Acceptance Score» (oppfylte krav / krav).  
- Architect: minimer «Structure Debt» (antall brudd på guider).  
- Engineer: minimer tid‑til‑grønn QA/PR.  
- Analyst: maksimer målingsdekning uten å overskride budsjett.

**Forhandlingsprotokoll (kort):** hver oppgave gjennomgår *proposal → critic → approve → execute*. Konflikt → Leader avgjør med Policy Guard + brukerprompt.

## 4) Vibe‑profiler v3

Definerer stack, stil, kvalitetskrav, testpolicy, sikkerhet, performance‑budsjett.

name: next\_tailwind\_secure\_v3  
stack:  
 language: typescript  
 framework: nextjs  
 css: tailwind  
quality:  
 lighthouse: { desktop\_min: 90, mobile\_min: 80 }  
 a11y\_block: [critical]  
 test: { unit: jest, e2e: playwright }  
 perf\_budget: { lcp\_ms: 2500, tbt\_ms: 200, cls: 0.1 }  
security:  
 deps: { audit: snyk, fail\_on\_severity: high }  
 headers: [csp\_strict, hsts, xfo\_sameorigin]  
style:  
 rules: [no\_any, prefer\_const, early\_return, module\_boundaries]  
constraints:  
 deps\_max: 14  
 pages\_min: ["/", "/pricing", "/contact"]

**Profiler kan arves/overstyres** per prosjekt, og håndheves av QA/Policy Guard.

## 5) Sandbox & Runtime

* **Windows:** LocalSandbox.Win (job objects, low integrity), WSL2Sandbox for Linux‑verktøy.
* **Android:** WASI sandbox (WebAssembly), isolerte prosesser; tunge jobber kan «cloud‑burst».
* **Ressursgrenser:** CPU/RAM/IO/Net; per prosjekt og per agent‑jobb.
* **Hermetiske builds:** låsing av avhengigheter, reproduserbare artefakter.
* **Nettpolicy:** allowlist domener for runner.exec.

## 6) Verktøy‑APIer v3 (stabile kontrakter)

**FS:** read/write/patch/diff/search (scopet til workspace).  
**Editor:** open/close/applyPatch/annotate (Monaco).  
**Runner:** exec(cmd, args, timeout, env), streams (stdout/stderr).  
**Preview:** open(url|port), shareLink(), health().  
**QA:** run(profile) → metrics/artifacts/asserts.  
**Git:** init/branch/commit/PR/status; sign commits.  
**Policy:** evaluate(action, ctx) → allow/warn/block.  
**Metrics:** emit(event, payload), defineDashboard(spec).  
**Secrets:** request(scope) → one‑time lease.

## 7) Inter‑agent meldingsskjema v3

{  
 "role": "leader|pm|architect|engineer|analyst",  
 "intent": "plan|spec|design|implement|test|measure|review|merge|deploy",  
 "topic": "pricing\_page",  
 "inputs": { "files": ["/pages/pricing.tsx"] },  
 "constraints": { "perf\_budget": { "lcp\_ms": 2500 } },  
 "proposal": { "steps": ["create page", "add table", "write tests"] },  
 "diff": "patch://...",  
 "asserts": ["route:/pricing", "lighthouse.perf>=90"],  
 "evidence": [{ "type": "qa", "ref": "qa://run/77" }],  
 "decision": { "status": "approve|revise", "notes": "…" }  
}

## 8) Flyt (plan → design → implement → test → measure → review → merge → deploy)

**Gates:** QA (LH/a11y/tests), Policy (secrets/deploy), Code Review (PM/ARC), Security Scan.  
**Operate (post‑deploy):** SLO‑måling, sanntidsdash, rollback‑policy.

ASCII‑sekvens:

User → Leader: Charter  
Leader → PM: Stories/Acceptance  
PM → Architect: Structure/ADR  
Architect → Engineer: Tasks/Patches  
Engineer → QA: Run profile  
QA → Leader: Gate green  
Analyst → Leader: Metrics ready  
Leader → Git/CI: PR → Merge → Deploy

## 9) QA & Preview v3

* **Ephemeral env** pr. branch/PR, seed‑data, service‑virtualisering.
* **Testmatrix:** desktop/mobil, mørk/lys, locale.
* **Security:** Snyk/OWASP ZAP (lett profil) som gate.
* **Flake manager:** retry med heuristikk, karanteneliste.

## 10) GitOps & Policy

* **PR‑policy:** «DoD limes inn automatisk», DCO sign‑off, konvensjonelle commits.
* **Deploy:** require\_approval til prod; break‑glass med audit.
* **Secrets:** one‑time lease, renses i logs.

## 11) Metrics SDK v3

* **Spec:** metrics.yaml (eventnavn, felt, PII‑nivå).
* **API:** metrics.emit(name, payload); batch flush.
* **Dashboards:** minimale, genereres fra spec; sammenlign før/etter release.

## 12) Marketplace 3.0 (Plugins/Profiles/Templates)

**Typer:** - **Tool‑plugins:** nye tools (f.eks. crawler, translator, db‑client).  
- **Agent‑plugins:** spesialiserte agenter (DesignCritic, SecEngineer).  
- **Profile Packs:** samling av Vibe‑profiler (stack‑oppsett, policies).  
- **Project Templates:** ferdige repo‑skeletter.

**Manifest (plugin):**

{  
 "name": "mde-crawler-pro",  
 "version": "3.0.1",  
 "type": "tool",  
 "entry": "dist/index.wasm",  
 "permissions": ["fs.read", "network.fetch"],  
 "sandbox": { "wasi": true, "cpu\_ms": 60000, "mem\_mb": 256 },  
 "signing": { "alg": "ed25519", "sig": "…" }  
}

**Runtime:** WASI (WASM) som standard; Node/.NET plugins støttes i isolerte «job‑runners».  
**Security:** permission‑modell; statisk skanning (secrets/malware); runtime caps.  
**Distribusjon:** offentlig katalog + Private Marketplace (org‑intern).  
**Økonomi:** revshare, lisensnøkler, usage‑metrics til forfatter (anonymisert).

## 13) Community Hub 3.0

* **Projects & Playbooks:** vis profil, backlog, QA‑status, «Open in Vibecoding».
* **Crews:** code‑review squads og mentor‑ordninger; @mentions og inline‑kommentarer.
* **Konkurranser:** månedlige «build challenges» m/ premier.
* **Reputation:** poeng/badges for merges, QA‑fixes, templates.
* **Moderering:** rapporter, takedown, lisens/ToS‑sjekk.

## 14) Pris & SKUer (utkast)

* **Premium:** Vibecoding (enkel), Tool‑plugins, 1 samtidige sandbox, community.
* **Enterprise:** Agent‑plugins, Private Marketplace, 5+ sandboxes, Policy Guard pro, on‑prem opsjon.
* **Compute‑kreditter:** runner‑minutter, QA‑kjøringer, sikkerhetsskann.

## 15) Milepæler V3

* **M0:** Orchestrator/agents, Vibe‑profiler v3, sandbox (Node/Python), Monaco, basic QA.
* **M1:** Git/PR, preview links, testmatrix, metrics SDK.
* **M2:** Marketplace (tool+profile), Community (projects/playbooks), security scan.
* **M3:** Agent‑plugins, Private Marketplace, on‑prem opsjon.

## 16) Risiko & mitigasjon

* **Sandbox omfang:** start WASI først, utvid til OCI trinnvis.
* **Flaky QA:** flake‑manager + karantene.
* **Plugin‑sikkerhet:** WASM default + signering + skanning.
* **Ressurskost:** kredittmodell, throttling, lokal cache.

## 17) Åpne spørsmål

* Minimum WASI‑APIs for dev‑tools?
* Hvilke agent‑plugins prioriteres (DesignCritic vs. SecEngineer)?
* Hvilke templates først (Next/FastAPI/.NET/React Native)?

## 18) Versjonslogg (V3)

* **V3.00:** Første spesifikasjon for Vibecoding Platform, Marketplace 3.0 og Community Hub 3.0 (dette dokumentet).