MadEasy AI Browser – V2.02 Komplett Plan og Arkitektur

---

1) Mål og visjon

- **Automasjon på neste nivå:** Multi-agent orkestrering med spesialiserte roller.
- **Enterprise readiness:** sikkerhet, policyer, revisjon og flerbrukerstøtte.
- **Marketplace:** åpne for deling og installasjon av playbooks og plugins.
- **Mobil paritet:** Android med voice-first og raske workflows.
- **Kvalitet:** Preview-drevet QA som fanger feil tidlig, inkl. ytelse, a11y og visuell regresjon.

---

2) Epics

1. **Multi-agent orkestrering 2.0** – Planner, Critic, Executor, Researcher, Fixer.
2. **Watched Workflows & Scheduler** – repeterende playbooks, diff-deteksjon, varsler.
3. **QA Suite Pro** – Lighthouse-diff, axe-core, visuell regresjon, console error gates.
4. **Selector Studio + læringsprofiler** – stabilitets-score, fallback-læring pr. domene.
5. **Marketplace (Playbooks & Plugins)** – signerte pakker, vurderinger, sandbox-policy.
6. **Collaborative Mode** – delte sessions, kommentarer, review-regler.
7. **Security & Compliance+** – policy-simulator, redaksjon i logger, revisjonsspor.
8. **Android Paritet v2** – voice-first, hurtigplaybooks, Keystore/StrongBox Vault.
9. **MadEasy Voice v2** – hotword + push-to-talk, meeting mode med diarization, barge-in.
10. **Data & Observability v2** – KPI-dashboard, event-DAG, eksport til BI.

---

3) Arkitekturforsterkninger

- **Shared Core** via gRPC: orchestrator, qa, vision, voice, market.
- **Policy Guard**: evaluerer alle handlinger mot regler/scopes.
- **Task Graph**: avhengigheter, retries, kompensasjoner.
- **CAS-lagring**: content-addressable storage for skjermbilder/rapporter.
- **Telemetry**: strukturert logging + anonymisering.

Arkitekturdiagram V2

MadEasy V2 Architecture

Merk (compliance): «Lead Data Vault» er admin-styrt og isolert fra sluttbrukere. Dersom dataene inneholder **personopplysninger** om personer i EU/EØS, vil **GDPR normalt fortsatt gjelde** for behandlingsansvarlig uavhengig av at sluttbrukere ikke kontrollerer lagringen. V2 leverer derfor **konfigurerbare «compliance modes»**: - *GDPR-kontrollert:* RLS, rettslig grunnlag (konfig), privacy-ledger, slett/innsyn, DPA/SCC dokumentasjon. - *Pseudonymisert/Hash:* sensitifiserte nøkler; opplåsing krever admin-prosess. - *Public-source only:* begrens til åpne kilder og felter uten privat karakter.

---

4) API-oppdateringer

- `qa_suite`: støtte for profiler og baseline-diff.
- `watch_workflow`: RRULE + triggers (content, element, status).
- `market.install` / `market.publish`: signering, manifest.
- `selector.stabilityScore(element)`: returnerer 0–100.
- `voice.start(mode, lang)`: streaming events.

---

5) Milepæler

- **M0 (uke 0–2):** Arkitektur, feature flags, migreringsplan.
- **M1 (uke 3–6):** Multi-agent 2.0, Selector Studio v2, QA Suite Pro del 1.
- **M2 (uke 7–10):** Watched Workflows, Marketplace (beta), Voice v2 grunnlag.
- **M3 (uke 11–14):** Collaborative Mode, Security+, Android paritet.
- **RC (uke 15–16):** Stabilisering, dokumentasjon, demo.

---

6) Sprintplan

Sprint 1 (2 uker): - Orchestrator v2 (task graph + policy hook). - Selector Studio v2 (stabilitets-score). - QA Suite Pro (LH-diff, console gate). - Voice v2 (hotword + PT-T). - KPI-instrumentering.

Sprint 2 (2 uker): - Watched Workflows (RRULE, diff). - Marketplace alpha (lokal installasjon + sandbox). - QA Suite Pro (visuell regresjon). - Collaborative grunnlag (read-only sessions).

---

7) KPIer

- **Automasjonsrate:** ≥70% playbooks uten manuell intervensjon.
- **Stabilitet:** ≥40% færre selector-feil.
- **Kvalitet:** ≥90% reduserte kritiske a11y-funn før merge.
- **Produktivitet:** 25–40% færre iterasjoner til grønn gate.
- **Engasjement:** ≥30 community-playbooks i Marketplace.

---

8) Risiko & mitigasjon

- **Anti-bot/ToS:** pacing, HIL-sjekkpunkter.
- **Ytelse:** lokal akselerasjon, batch, throttling.
- **Sikkerhet:** strenge scopes, release-once secrets, policy-simulator.
- **Android begrensning:** lite-profiler, tydelig støtte-matrise.

---

9) Multi-agent orkestrering 2.0 (detaljer)

Roller: Planner (plan), Critic (risiko/kvalitet), Executor (handling), Researcher (kildeinnhenting), Fixer (feilretting).

Tilstandsmodell: plan → propose → review → execute → validate → fix* → done/escalate.

Meldingsskjema: JSON med rolle, intent, proposal, asserts, evidence, decision.

Consensus: Planner+Critic kjører «short debate» (≤3 runder). Ved konflikt → human-in-the-loop.

---

10) Policy Guard (regelmotor)

Formål: Evaluere hver handling mot policyer (domene, risiko, bruker-scope).

Regel-skjema: YAML med scopes og policies. Simulator for dry-run.

---

11) Selector Studio v2

Stabilitets-score (0–100): basert på ARIA, synlig tekst, data-attributes, dom-dybde, sibling-varians.

Profiler: per domene med preferred selectors, fallbacks, anti-patterns.

---

12) QA Suite Pro (profiler & diff)

Profiler: definert i YAML. LH-min, axe-block-levels, routes, forms, visual regression baseline, console gates.

Rapport: LH diff, visuelle deltas, konsollfeil. Publiseres i PR.

---

13) Watched Workflows & Scheduler

RRULE: f.eks. ukentlig hver mandag 07:00. **Triggers:** content change, element change, status change. **Actions:** kjør playbook, opprett PR, send varsel.

---

14) Marketplace (Playbooks & Plugins)

Manifest: JSON med name, version, author, permissions, entry, signatur. **Policy:** sandbox krever samsvar mellom manifest og runtime. **Publishing:** signering → scanning → listet i katalog.

---

15) Collaborative Mode

- Live cursors, delte sessions, kommentarer med @mentions.
- Review-regler: 2 øyne på kritiske endringer.
- Sessions logges med events og deltagere.

---

16) Android Paritet v2

- Voice-first UI, Quick Playbooks.
- Vault koblet til Keystore/StrongBox.
- Lite QA: enkle sjekker lokalt, LH kjøres server-side.

---

17) MadEasy Voice v2

- STT: Whisper (DirectML Windows, lite Android).
- Hotword/PTT + barge-in.
- Diarization i møte-modus, oppsummering til arkiv.
- Events: partial_transcript, final_transcript, intent.

---

18) CI/CD maler

GitHub Actions PR-gate: kjør QA runner i pipeline. **Release pipeline:** bygg artefakter, signering (MSIX/electron), opplasting.

---

19) Telemetry & personvern

- Event DAG, sampling.
- Privacy ledger: retention, anonymisering.
- Eksport: CSV/Parquet for BI.

---

20) Migrering V1 → V2

- Kompatibilitetslag for playbooks.
- Policy Guard: starter som advarsel, går til blokk.
- QA profiler per domene.

---

21) Åpne spørsmål

- Betalte pakker i Marketplace V2 eller V2.1?
- Hvor strenge skal default policyer være?
- Minstestøtte for Android WebView features?

22) Lead Data Vault (admin-only, global, Non-EU default)

Default (Non-EU drift): Lead Vault kjører som en **admin-kontrollert database uten sluttbrukerrettigheter**. Alle leads (scrapet/samlet) lagres her i tillegg til at brukeren kan lagre/eksportere lokalt som vanlig. Vaulten fungerer dermed som et **sentralisert arkiv for admin** med enrichment, dedup og scoring. Brukere har aldri lesetilgang – kun admin via Read-Access Layer. GDPR-mekanismer er ikke aktivert som standard.

Opsjonelle compliance-moduser: - **GDPR-kontrollert:** for EU/EØS-marked, med RLS, privacy ledger, slett/innsyn, rettslig grunnlag. - **Pseudonymisert/Hash:** fingerprint-lagring; PII krever admin-prosess. - **Public-source only:** begrens til åpne kilder.

22.1 Arkitektur

Browser → Ingestion API (mTLS+signatur) → Enrichment → Dedup/Scoring → Lead Data Vault (Postgres RLS, Admin-only) → Read-Access Layer (eksport) → Audit/Compliance.

22.2 Datamodell

- **lead** (id, fingerprint, source_url, collected_at, collector_id, raw_blob, normaliserte felter)
- **enrichment** (lead_id, provider, payload, confidence)
- **score** (lead_id, fit, reach, intent, credibility, total)
- **audit_log** (event_id, actor, action, hash_prev, hash_curr)

22.3 Dedup & scoring

Fingerprint + fuzzy-match; merge policy; retention (konfigurerbar TTL).

22.4 Tilgang & policy

- RLS: kun admin leser, ingestor kan kun insert.
- Policy Guard: blokkerer alle klient-reads.
- Eksport via Admin-lag, ikke direkte DB.

22.5 Enrichment

Verifisering (MX/SMTP, HLR), firmografi, geo, normalisering, prioritering.

22.6 Sikkerhet

mTLS+signatur på ingest, PII-maskering i logger, audit hash-kjede.

22.7 Integrasjon

Playbooks sender alltid via ingest-step.

22.8 Admin-verktøy

Søk/filtrer, merge, manuell enrichment, eksport til CRM.

22.9 Overvåkning

Contract tests, dedup-rapporter, alarmer på ingest/RLS-brudd.

22.10 Sekvens: Dobbel lagring (lokalt + Vault)

Dual Storage Sequence

Forklaring: Når en playbook kjører, eksporterer browseren data **lokalt** (CSV/XLSX/JSON) for brukeren *og* sender samtidig et normalisert, signert payload via **Ingestion API** (mTLS) til Lead-tjenestene for enrichment/dedup/scoring før lagring i **Lead Data Vault (Admin-only, RLS)**.

Eksempel – Playbook-steg (YAML):

```
name: leads_cast_iron_eu
inputs:
  query: "cast iron cookware wholesaler EU"
  take: 50
steps:
  - goto: https://www.google.com
  - search: ${query}
  - scrape_results:
      take: ${take}
      fields: [company, url, email, phone, country]
  - normalize:
      map:
        email: ${item.email | to_lower | trim}
        phone: ${item.phone | e164}
        domain: ${item.url | domain}
        company: ${item.company | title}
  - export:
      type: xlsx
      path: /exports/leads_${now:yyyyMMdd}.xlsx
      columns: [company, domain, email, phone, country]
  - ingest:
      endpoint: https://vault.example.com/ingest/lead
      auth:
        mtls: true
        client_id: ${profile.client_id}
        sign:
          alg: ed25519
          key_ref: vault_signing_key
      payload:
        source_url: ${item.url}
        playbook: ${playbook.name}@${playbook.version}
        collector_id: ${profile.instance_id}
        normalized:
          email: ${item.email}
          phone: ${item.phone}
          company: ${item.company}
```

```
        domain: ${item.domain}
        country: ${item.country}
```

Notater: - `export` skjer alltid lokalt for brukerens arbeidskopi. - `ingest` går alltid via mTLS + signert payload til Vault, uavhengig av bruker/modus. - Feil i `ingest` skal **ikke** blokkere lokal eksport; de logges og re-queues for retry.

---

23) Vanlige nettleserfunksjoner

MadEasy AI Browser skal også støtte **standard nettleser-opplevelse** slik at brukeren kan bruke den som primær browser: - Faner, bokmerker, historikk. - Nedlastinger, filleser. - Autofyll (passord, kort, adresser) via Vault-integrasjon. - Utvidelsesstøtte (extensions kompatible med Chrome/Edge API der mulig). - Incognito/privat-modus. - Developer Tools (inspiser DOM, nettverkslogg).

---

24) Brukerregistrering og tilgang

- **Registrering:** E-post + passord eller SSO (Google/Microsoft/GitHub).
- **Gratis vs. Full tilgang:**
- Gratis: begrenset antall playbooks/oppgaver.
- Full: ubegrenset, inkl. DevBridge, Lead Vault-integrasjon.
- **Profilsystem:** brukerens preferanser, playbook-bibliotek, tokens.
- **Abonnement:** monthly/annual, admin kan tildele roller.

---

25) Community plass

- **Forum/Hub:** innebygd i browseren (webview til community.madeasy.ai).
- **Deling:** playbooks, plugins, erfaringer.
- **Rangering:** upvotes/downvotes, badges.
- **Moderering:** policy-styrt, rapporteringssystem.
- **Docs & læring:** tutorials, API-eksempler, video-demoer.

---

26) Brukerlagde plugins

- **Plugin SDK:** TypeScript/Node-basert, definert API for browsertool, scraping, devbridge.
- **Sikkerhet:** sandbox, deklarative tillatelser (manifest.json).
- **Distribusjon:** via Marketplace (signert + policy-skannet).
- **Eksempel manifest:**

```
{
  "name": "my-scraper-plugin",
  "version": "0.1.0",
  "permissions": ["browser.navigate", "scrape"],
```

```
  "entry": "index.js"
}
```

- **Eksempel API-kall:**

```
const tab = await madEasy.browser.open("https://example.com")
const data = await tab.scrape({ selector: "table tr" })
```

---

23) Vanlige nettleser-funksjoner (Core Browser)

Faner & vinduer: tab groups, pin, dupliser, tilbakestill lukkede faner. **Historikk & bokmerker:** mapper, tags, synk mellom enheter (opt-in), import/eksport (Chrome/Edge/Firefox HTML/JSON). **Nedlastinger:** kø, pause/gjenoppta, automatisk mappesortering etter filtype/domenepolicy. **Søk/URL-felt:** forslag (lokal historikk + tilkoblede søkemotorer), hurtigkommandoer (`:settings` , `:playbook`), privat søk. **Skjemaautofyll:** adresser, navn, telefon (kryptert lokalt), profilstyring. **Passord/Secrets:** egen vault (DPAPI/Keystore), generator, brudd-sjekk (lokale lister), 2FA-støtte. **Personvern/Blocking:** tracker-liste, tredjepartskapsler, fingeravtrykk-reduksjon, per-site rettigheter. **Ytelse:** ressursmåler per fane, "sovende faner", throttling av bakgrunnsarbeid. **Utklipp/lesevisning:** snapshot til notater, justérbar lesevisning, oversettelse (lokal/sky-modell). **DevTools:** DOM/Network/Console, HAR-eksport, element-inspektør, "Explain Element".

---

24) Brukere, pålogging & tilgangsnivåer

Registrering/pålogging: e-post+passord, SSO (Google/Microsoft), magic link. MFA (TOTP/WebAuthn). **Profiler:** Free (basis), Pro (playbooks, anonym/opptak), Enterprise (policyer, CI-integrasjon, Lead Vault admin-tilgang). **Onboarding:** veiviser → profiler, scopes, første playbook. **Fakturering:** Stripe (kort), støtte for teams/seat-basert. Kvoter per plan (kjøretid, eksport, lagring). **Tillatelser:** scopes per domene/handling (les/skriv/opplasting/commit/deploy). Pro/Ent kan definere egne policyer.

Flyt (høy nivå): 1) Sign-up → e-postverifisering / SSO → valgt plan. 2) Første innlogging → onboarding-playbook. 3) Ved farlige handlinger → policy prompt + evt. admin-godkjenning.

---

25) Community Hub

Formål: Samle brukere, playbooks, diskusjoner og showcases. - **Playbook-galleri:** søk/filtrer, rating, versjoner, changelog. - **Diskusjoner/Q&A:** tråder per playbook, tips, feilsøking. - **Showcase:** del resultater (skjermbilder/rapporter) med masking av PII. - **Konkurranser:** månedlige "automation challenges". - **Moderering:** rapporter, takedown, lisens/ToS etterlevelse.

Creator-profil: bio, badges, statistikk (downloads, stars), donasjon/affiliates.

---

26) Plugin/Playbook-SDK (utvidelser)

Typer: - **Playbooks:** YAML-definerte arbeidsflyter (full tilgang via definert sett med tools). - **Plugins:** kjørbar utvidelse (Node/.NET) som eksponerer nye **tools**; lastes sandkasset.

Manifest (plugin):

```
{
  "name": "mad-scraper-pro",
  "version": "0.2.1",
  "author": "your@domain",
  "permissions": ["browser.navigate","network.fetch","file.write"],
  "entry": "index.js",
  "sandbox": { "cpu_ms": 30000, "mem_mb": 256 },
  "signing": { "alg": "ed25519", "sig": "…" }
}
```

API-overflater: - `browser`: navigate, click, type, evaluate, screenshot, network log. - `system`: file read/write (scoped), secrets, temp storage, timers. - `qa`: lighthouse, axe, visual-diff, console gate. - `lead`: normalize, verify, ingest (mTLS/signatur) → **Vault**.

Sikkerhet: - Sandkasse (job runner) med ressursgrenser og filsystem-scopes. - Strict permission-modell: kun det manifestet ber om blir tilgjengelig. - Signering og integritetsjekk ved install/oppdatering.

Distribusjon: via **Marketplace** eller lokalt (dev-modus). Versjonspinning og rollback.

---

27) Plugin-utvikleropplevelse

Dev-CLI: `mde plugin init|dev|pack|publish`. **Hot-reload:** lokal dev med test-browser, mock-scopes og logs. **Testing:** kontraktstester mot tools, snapshot-tester av output. **Eksempelskjelett:**

```
// index.ts (Node plugin)
import { ToolKit, definePlugin } from 'mde-sdk'

export default definePlugin({
  name: 'mad-scraper-pro',
  setup: (kit: ToolKit) => ({
    async run(ctx) {
      const page = await kit.browser.newPage()
      await page.goto(ctx.inputs.url)
      const data = await page.$$eval('a.result', els => els.map(e => ({ t:
e.textContent, href: e.href })))
      await kit.system.writeFile(ctx.outputs.path, JSON.stringify(data,
null, 2))
      return { count: data.length }
```

```
    }
  })
})
```

---

28) Governance for Community & Marketplace

- **Innsending:** automatisk skanning (secrets, malware, farlige rettigheter).
- **Review:** rask manuell gjennomgang for «featured»; ellers automatisk publisering m/ rate-limit.
- **Policy brudd:** takedown, sperring av nøkler, varsel til forfatter.
- **Versjonering:** semver, deprecations, sikkerhets-bulletiner.
- **Analytics:** nedlastinger, aktive installasjoner, krasj-rapporter (anonymisert).

---

29) Produktpakker og prising (utkast)

- **Free:** core browser, 3 playbooks, basis blokkering, lokale eksporter.
- **Pro:** ubegrenset playbooks, anonym/profiler, QA Suite (basis), Community Creator, plugin-dev.
- **Enterprise:** Policy Guard, Selector Studio v2, QA Pro, Watched Workflows, SSO/MFA, CI-integrasjon, **Lead Vault Admin**, dedikert support.

---

30) Veikart – tillegg

- **Leseliste & notater** synk (opt-in) med kryptering.
- **Side-opptak til tutorial** (makro til playbook).
- **Team-spaces** i Community med private delinger.
- **App-innlogging for Android ↔ Windows** sesjonsdeling.

---

31) Lead Vault Access & Pricing

Modell: Tilgang til Lead Vault krever **Enterprise-plan**. I tillegg prises uttrekk etter **kombinasjon av land og kategori**.

31.1 Prisingsmodell

- **Enterprise-lisens:** fast kostnad pr. organisasjon (årlig/månedlig).
- **Per lead:** uttrekk av leads fra Vault belastes med enhetlig kredittsystem.
- **Differensiering:**
- **Kategori:** f.eks. Fitness = 0.10, Beauty = 0.12, Wellness = 0.15.
- **Land/region:** Tier 1 (US, DE, UK, NO, FR) høyest, Tier 2 middels, Tier 3 lavest.
- Endelig pris = Enterprise-kost + (kategori-rate + land-rate) × antall leads.

31.2 Teknisk implementasjon

- **DB-felter:** `category`, `country`, `enrichment_level` pr. lead.
- **Pricing Engine:** beregner kredittkost basert på tabell.

- **Credits Table:** admin kjøper pakker; trekk skjer ved eksport.
- **Access Layer:** sjekker kreditt før eksport og logger forbruk.
- **Billing Logs:** alle uttrekk logges (hash-chain audit).

31.3 Eksempel (YAML-policy)

```yaml
pricing:
  enterprise_fee: 2000   # fast per år (eksempel)
  by_category:
    fitness: 0.10
    beauty: 0.12
    wellness: 0.15
  by_country:
    US: 0.20
    NO: 0.18
    IN: 0.06
```

31.4 Diagram (oversikt)

Bruker (Enterprise) → Access Layer (kredittkontroll) → Vault → Eksport.

---

32) Idébank – Feature Matrix

En oversikt over potensielle funksjoner delt i tre hovedområder. Kolonnene viser verdi og om funksjonen passer best for **Premium** eller **Enterprise**.

Område	Funksjon	Verdi for bruker	Plan
Nettleser	Isolerte arbeidsprofiler	Parallell surfing med egne cookies, VPN og fingerprint	Enterprise
Nettleser	Auto-workflows i tabs	Automatisk refresh, scrape, koble data mellom faner	Premium
Nettleser	Integrert VPN/Proxy manager	Velg utgangsnode per tab for research eller compliance	Enterprise
Nettleser	Smart tab memory	Suspend/restore grupper, reduser ressursbruk	Premium
Nettleser	Content Diff Mode	Marker endringer i DOM/tekst på en side	Premium
Nettleser	PDF/Web Capture	Høyfidelitets snapshots til søkbare PDFer	Premium
Nettleser	Compliance Browsing	Maskér/eksporter kun whitelistede felter	Enterprise
Dev	Auto-Test Generation	Generer Cypress/Playwright-tester fra playbooks	Premium
Dev	Multi-Env Preview	Sammenlign PR/Staging/Prod side-by-side	Enterprise

Område	Funksjon	Verdi for bruker	Plan
Dev	Security Scan Integration	Kjør OWASP/Snyk scanning i pipeline	Enterprise
Dev	Visual Git Timeline	Grafisk commit/PR/preview-oversikt	Premium
Dev	Experiment Mode	Sandbox for patch-testing uten å røre repo	Premium
Dev	Auto-Changelog	Automatisk changelog fra merges og QA	Premium
Dev	Container Integration	Start/stop Docker Compose fra playbook	Enterprise
Leads	Smart Lead Enrichment	AI parsing og berikelse av fritekst/kataloger	Enterprise
Leads	Lead Dedup Dashboard	Visualiser overlapp og merge-forslag	Premium
Leads	Intent Mining	ML analyserer sider for kjøpsintensjon	Enterprise
Leads	Competitor Tracking	Overvåk konkurrent-sider og samle leads	Enterprise
Leads	Automated Lead Nurture	Integrasjon med CRM for lead-sekvenser	Enterprise
Leads	Data Quality Score	Sanntidsvalidering: valid, risky, enriched	Premium
Leads	Lead Flow Policies	Regler per land/kategori før eksport	Enterprise
Leads	Trend Analytics	Grafer på volum, kvalitet, kategori, kilde	Premium

33) Utvidet Idébank – Nye Funksjoner

Nye forslag lagt til i tabellen, utvidet med avanserte nettleser-, dev- og lead-funksjoner samt enterprise-tillegg.

Område	Funksjon	Verdi for bruker	Plan
Nettleser	AI Smart Summaries	Automatisk sammendrag av sider og PDF-er	Premium
Nettleser	Contextual Voice Control	Stemmekommandoer i naturlig språk («finn PDF-er med ‘pricing’»)	Enterprise
Nettleser	Cross-Device Continuity	Fortsett sesjoner sømløst PC ↔ Android	Premium
Nettleser	Multi-View Mode	Split screen / grid-visning av flere sider	Premium
Nettleser	Stealth Mode 2.0	Fingerprint/proxy-rotasjon per request	Enterprise
Nettleser	Workflow Recorder	Record-clicks → auto-generer playbook	Premium
Dev	AI Code Refactorer	Analyse & forbedring av kode (ytelse/sikkerhet)	Enterprise
Dev	Dependency Health Monitor	Overvåk npm/nuget for sårbarheter og stale libs	Premium
Dev	Cross-Platform Deployment	Deploy til Vercel, Netlify, Azure, AWS	Enterprise

Område	Funksjon	Verdi for bruker	Plan
Dev	Issue Auto-Triage	Automatisk kategorisering av issues (GitHub/ Jira)	Premium
Dev	Code Review Assistant	AI-kommentarer i PR-er med DoD-sjekk	Enterprise
Dev	Instant Sandbox	Start isolert container for repo/PR m/ dev-AI	Enterprise
Leads	Predictive Lead Scoring	AI for konverteringssannsynlighet	Enterprise
Leads	Lead Route Engine	Automatisk ruting av leads til riktige team/CRM	Enterprise
Leads	ABM Mode	Grupper leads etter selskaper (account-based)	Enterprise
Leads	Dark Web Lead Monitor	Overvåk mørke nett for kontakt-/firmadata	Enterprise
Leads	Real-Time Lead Alerts	Push-varsler når leads matcher kriterier	Premium
Leads	Lead Privacy Shield	Dynamisk maskering av PII etter rolle	Enterprise
Enterprise	Custom Policy Engine	Last egne YAML-policyer for overstyring	Enterprise
Enterprise	Air-Gapped Mode	Kjør browser/Vault helt lokalt, uten sky	Enterprise
Enterprise	On-Prem Deployment	Containerisert Vault + orchestrator	Enterprise
Enterprise	SSO & SCIM	Integrasjon med IdP/HR-systemer	Enterprise
Enterprise	Enterprise Audit Trail	Eksport av revisjon til eksterne systemer	Enterprise

34) Roadmap Visualisering

Visuell oversikt over planlagte funksjoner fordelt på **versjonsutgaver**. Hver versjon må ferdigstilles og publiseres før neste starter.

```
ROADMAP – MadEasy Browser

V2.1 (første utvidelse etter MVP/V2):
[■■■■■■■■■] Workflow Recorder – record-clicks til playbook
[■■■■■■■■■] AI Smart Summaries – sammendrag av sider/PDF
[■■■■■■■■■] Lead Dedup Dashboard – oversikt over duplikater
[■■■■■■■■■] Dependency Health Monitor – overvåk libs
[■■■■■■■■■] Real-Time Lead Alerts – push-varsler på leads
```

```
V2.2:
[■■■■    ] Cross-Device Continuity – sesjoner PC ↔ Android
[■■■■    ] Multi-View Mode – split screen / grid tabs
[■■■■    ] Auto-Test Generation – generer testkode fra playbooks
[■■■■    ] Code Review Assistant – AI-kommentarer i PR
[■■■■    ] Lead Route Engine – regler for routing av leads
[■■■■    ] Predictive Lead Scoring – AI-basert konverteringsscore

V2.3:
[■■      ] Stealth Mode 2.0 – fingerprint/proxy per request
[■■      ] Instant Sandbox – isolert container for repo/PR
[■■      ] ABM Mode – account-based marketing grupper
[■■      ] Dark Web Lead Monitor – overvåk mørke nett
[■■      ] On-Prem Deployment – kjør Vault/Core internt
[■■      ] Air-Gapped Mode – drift helt uten sky
```

Merk: - V2.1 = første "Premium Release" med brukerrettede features. - V2.2 = utvidelser for samarbeid, dev-integrasjon og avansert scoring. - V2.3 = enterprise-nivå sikkerhet, on-prem og avanserte lead-funksjoner.

---

35) Tekniske Detaljer per Hovedelement

35.1 Lead Data Vault (inkl. Access & Pricing)

Formål & verdi
- Sentralisert, admin-only database som samler **alle leads** uavhengig av bruker og modus.
- Sikrer datakvalitet, deduplisering, berikelse og scoring.
- Tilgang er begrenset til Enterprise-kunder, med prising per land/kategori + fast Enterprise-kost.

Brukeropplevelse (UX-flow)
- Vanlig bruker: får lokale eksporter (CSV/XLSX/JSON) fra playbooks, helt uavhengig av Vault.
- Admin (Enterprise): logger inn i Access Layer og kan søke, filtrere, berike og eksportere leads.
- Kredittsystem: admin kjøper kreditter → velger leads → system beregner pris (land+kategori) → eksport genereres.

Systemkrav
- **API:** `/ingest/lead` (mTLS + signatur), `/admin/query`, `/admin/export`.
- **Database:** PostgreSQL 15+ med Row-Level Security.
- **Autentisering:** mTLS for ingest; OIDC/SSO for admin.
- **Dependencies:** Lead Enrichment services (MX/SMTP check, HLR, firmografi, geo).

Datamodell
- `lead(id uuid, fingerprint text, source_url text, collected_at ts, collector_id text, raw jsonb, email text, phone text, company text, domain text, country text)`
- `enrichment(id, lead_id, provider, payload jsonb, confidence, ts)`
- `score(lead_id, fit int, reach int, intent int, credibility int, total int)`

- `audit_log(event_id, actor, action, lead_id, hash_prev, hash_curr, ts)`
- `pricing(category text, country text, rate numeric)`
- `credits(org_id, balance numeric, last_topup ts)`

Integrasjoner
- **Playbooks:** alltid med `export` + `ingest` steg.
- **Policy Guard:** blokkerer alle ikke-admin read-forsøk.
- **Marketplace:** plugins kan bare sende til ingest, aldri lese.

Eksempel – ingest request

```
POST /ingest/lead
Content-Type: application/json
X-Client-Id: abc123
X-Signature: ed25519(<body>)
{
  "source_url": "https://example.com",
  "collector_id": "me-42",
  "normalized": {
    "email": "a@b.com",
    "phone": "+4711223344",
    "company": "Acme",
    "domain": "acme.com",
    "country": "NO"
  },
  "playbook": "leads_cast_iron_eu@2.1"
}
```

Eksempel – pricing policy (YAML)

```yaml
pricing:
  enterprise_fee: 2000
  by_category:
    fitness: 0.10
    beauty: 0.12
    wellness: 0.15
  by_country:
    US: 0.20
    NO: 0.18
    IN: 0.06
```

Done-criteria
- Alle ingest-requests logges i audit_log med hash-kjede.
- Deduplication implementert (fingerprint + fuzzy).
- Kredittsystem sjekkes ved eksport.
- Admin kan søke/filter + eksport til CSV/XLSX.
- Policy Guard blokkerer all ikke-admin lesing.

35.2 Policy Guard

Formål & verdi
- Sentral motor for å validere **alle handlinger** mot definerte regler og scopes.
- Hindrer uautoriserte deploys, commits, opplastinger og eksporter.
- Gjør systemet enterprise-klar med sporbarhet og simulering av policyer.

Brukeropplevelse (UX-flow)
- Når en handling initieres (f.eks. `deploy prod`), Policy Guard sjekker om brukeren og domenet har riktige scopes.
- Dersom policy blokkerer, vises et **Policy Prompt** i UI med forklaring og evt. behov for admin-godkjenning.
- Admin kan bruke Policy Simulator for å teste endringer i regler før de aktiveres.

Systemkrav
- **API:** `/policy/evaluate`, `/policy/simulate`, `/policy/update`.
- **Regelfiler:** YAML/JSON lastes fra database eller repo.
- **Database:** `policy_rules(id, scope, conditions jsonb, effect, created_at)`.
- **Autentisering:** krever admin-rolle for å oppdatere regler.

Datamodell (regel)

```
- id: prod_deploy_requires_approval
  when:
    action: deploy
    env: prod
  allow: false
  require: [ human_approval ]
- id: external_upload_block
  when:
    action: upload
    domain: not_in(allowed_domains)
  allow: false
```

Integrasjoner
- **Orchestrator:** alle steg rutes via Policy Guard før eksekvering.
- **Lead Vault:** policy sjekker at kun ingest (ikke read) er tillatt for klienter.
- **Marketplace:** plugins må samsvare med manifest-permissions.

Eksempel – evaluate call

```
POST /policy/evaluate
{
  "action": "deploy",
  "env": "prod",
  "actor": "user123",
  "context": { "repo": "acme/app" }
```

```
}

Response: { "allowed": false, "require": ["human_approval"] }
```

Done-criteria
- Alle handlinger evalueres mot Policy Guard før kjøring.
- Policy Simulator kan kjøres i UI med trace-eksport.
- Admin kan opprette, endre og publisere regler.
- Avviste handlinger logges i audit_log med årsak.

35.2 Policy Guard (Regelmotor & Håndheving)

Formål & verdi
- Sørger for at **hver handling** (navigate, click, type, upload, commit, deploy, export, vault.read) evalueres mot eksplisitte regler før den kjøres.
- Reduserer risiko (utilsiktede deploys, datalekkasjer) og muliggjør **enterprise-kontroll** pr. domene, miljø, rolle og plan.

Brukeropplevelse (UX-flow)
- **Silent allow** når handlingen er trygg (grønn indikator i Logg).
- **Warn/Human-in-the-loop** når policy krever eksplisitt bekreftelse (modal med begrunnelse + «Request approval»).
- **Block** med tydelig melding (regel-ID, «contact admin»).
- **Simulator** i Settings → «Dry-run policy» mot en valgt session/trace for å se hvorfor noe ble blokkert.

Håndhevingspunkter (enforcement points)
- **Executor** før verktøy-kall: `BrowserTool`, `FileTool`, `DevBridge`, `LeadTool`, `ExportTool`.
- **Network** (fetch/upload/download) – sjekk domene, content-type og størrelse.
- **Vault Access Layer** – `vault.read/export` krever enterprise + kreditt + policy.
- **CI/CD hooks** – blokker `deploy:prod` uten godkjenning.

Regelmotor (evaluering)
- Policy Guard tar en **PolicyContext** og et **ActionRequest** og evaluerer mot en regel-liste.
- Første match ender evalueringen (prioritetsrekkefølge); flere regler kan kombineres med `all/any`.

PolicyContext (eksempel)

```
{
  "user": {"id":"u_12","plan":"enterprise","roles":["admin"],"org":"org_7"},
  "session": {"id":"s_99","mode":"automate","profile":"anon-eu"},
  "env": {"platform":"windows","app":"MadEasy","version":"2.1.0"},
  "resource": {"domain":"github.com","path":"/org/repo","env":"prod"}
}
```

ActionRequest (eksempel)

```
{ "action":"deploy", "params": {"target":"prod", "pr":"#142"} }
```

Regel-skjema (YAML)

```
version: 1
scopes:
  - id: github.write
    description: "Open PR, merge, push"
  - id: vault.read
    description: "Read/export from Lead Vault"

policies:
  - id: prod_deploy_requires_approval
    description: "Prod deploy krever admin-godkjennelse"
    when:
      action: deploy
      resource.env: prod
    effect: require_approval
    approvers: [ role:admin ]

  - id: deny_external_upload
    description: "Blokker opplasting til ukjente domener"
    when:
      action: upload
      resource.domain: not_in(allowed_upload_domains)
    effect: deny

  - id: vault_read_enterprise_only
    when:
      action: vault.read
      user.plan: not_eq(enterprise)
    effect: deny
```

Pseudokode (evaluering)

```
PolicyDecision Evaluate(PolicyContext ctx, ActionRequest act) {
  foreach (var rule in OrderedRules) {
    if (rule.When.Matches(ctx, act)) {
      switch(rule.Effect) {
        case Allow: return Allow();
        case Deny:  return Deny(rule.Id, rule.Description);
        case RequireApproval: return Pending(rule.Id, rule.Approvers);
      }
    }
  }
```

```
  return DefaultDeny();
}
```

API-overflate
- `POST /policy/evaluate` – for simulator og ekstern kall i CI.
- `GET /policy/active` – aktiv policy med versjon/hash.
- `POST /policy/approve` – signert godkjenning (approver-rolle, TOTP/WebAuthn).

Datamodell
- `policy_bundle(id, version, hash, raw_yaml, created_at)`
- `approval(id, rule_id, actor, act_digest, status, ts)`
- `decision_log(id, session_id, rule_id?, action, effect, reason, ts)`

Integrasjoner
- **Lead Vault:** `vault.read/export` håndheves her + kredittkontroll.
- **DevBridge:** blokker merge/deploy/secret-bruk uten riktig scope/approval.
- **Marketplace:** plugins får **kun** de permissions manifestet ber om; Policy Guard verifiserer mismatch.

Eksempler
- **Deploy blokkert:** bruker forsøker `deploy:prod` uten godkjenning → modal med «Request approval» → admin godkjenner → handling re-tries og kjøres.
- **Opplasting blokkert:** plugin prøver å laste opp til ukjent domene → «deny» med regel-ID → logg.
- **Vault-lesing:** ikke-enterprise bruker prøver eksport → avvist (regel: `vault_read_enterprise_only`).

Done-criteria
- Alle handlinger går gjennom Policy Guard; logger inneholder beslutning og regel-ID.
- Simulator kan reprodusere en sesjonsbeslutning deterministisk (policy-versjon+hash).
- Approval-flow med signering/MFA er på plass.
- CI-hook blokkerer prod-deploy uten godkjenning.

---

36) Grunnfunksjoner (Core Must-Haves)

36.1 Nettleser

Formål: sikre at MadEasy oppfører seg som en fullverdig browser, med robusthet og daglig brukbarhet. **Funksjoner:** - Oppdateringssystem (auto-update, rollback ved feil). - Synkronisering (opt-in) av bokmerker, innstillinger, playbooks. - Crash Recovery: gjenopprett tabs, playbooks, workflows. - Offline-modus: kjør playbooks lokalt mot HTML-filer eller cached data. - Flere brukerprofiler (privat, jobb, research). **Done-criteria:** auto-update med rollback testet, tab-recovery fungerer, minst 2 profiler støttet.

36.2 Utvikling

Formål: gi devs og avanserte brukere kontroll og transparens.
Funksjoner: - Logging & Debug Panel for AI/Executor-beslutninger. - Replay Mode: kjør tidligere playbook-run på nytt. - Element Inspector med «copy selector» for manuelle overstyringer. - Unit Test

Runner for YAML-playbooks. - Mock Data/Env: test playbooks uten prod-trafikk. **Done-criteria:** logs kan eksporteres, replay fungerer, minst 1 playbook-test per release.

36.3 Leads

Formål: sikre at selv basisbrukere får verdi fra leads-funksjoner uten Vault.
Funksjoner: - Basic Lead Export (CSV/XLSX lokalt, uten enrichment). - Lead Validation Lite (regex for e-post/tlf). - Duplicate Warning (viser % overlapp med Vault). - Lead Tagging (hot, cold, supplier, partner). **Done-criteria:** eksport til CSV/XLSX fungerer, regex-validering stopper åpenbare feil, tagging lagres lokalt.

36.4 Sikkerhet

Formål: grunnsikring for alle brukere.
Funksjoner: - Permission Prompts for sensitive handlinger (login, upload, delete). - Secret Manager UI for API-nøkler (Google, GitHub). - Sandboxed Downloads (karantene + virusscan). - Privacy Mode Toggle (auto-slett historikk/logger ved exit). **Done-criteria:** minst én prompt per sensitiv handling, secrets krypteres i Vault, downloads scannes.

36.5 Android (mobil)

Formål: gjøre mobilappen nyttig alene.
Funksjoner: - Share-to-MadEasy (fra LinkedIn, Gmail, Chrome). - Quick Actions / widgets for playbooks. - Offline Export: kjør små scraping-playbooks og lagre til Files. **Done-criteria:** «Share to MadEasy» fungerer, minst én widget støttet, offline scraping testet.

35.3 QA Suite Pro (Kvalitetssikring)

Formål & verdi
- Gi enhetlig kvalitetssikring av web-apper, playbooks og dev-leveranser før merge/deploy.
- Reduserer risiko for feil i produksjon og øker tillit til automatisering.

Brukeropplevelse (UX-flow)
- Bruker/PM ser et QA-kort i browseren eller GitHub PR med grønn/rød status per assert.
- Rapporter genereres automatisk ved kjøring av playbook eller preview-URL.
- Mulighet for «drill-down» i detaljer (Lighthouse metrics, a11y-funn, skjermbilder).

Systemkrav
- `qa_suite` modul som kjører via Playwright/Puppeteer.
- Integrasjon med Lighthouse CLI og axe-core for a11y.
- Visual regression testing med skjermbilde-diff.
- Console log capture (max severity).

Datamodell
- `qa_result(id, project_id, commit_sha, profile, metrics jsonb, created_at)`
- `qa_assert(id, qa_result_id, type, status, details jsonb)`
- `qa_artifact(id, qa_result_id, type, path/url, created_at)`

Profiler (YAML-eksempel)

```
profile: web_app_default
lighthouse:
  desktop_min: 85
axe:
  block_levels: [critical]
routes:
  - "/"
  - "/contact"
forms:
  - selector: "form#contact"
    fields:
      email: "test@invalid"
    expect_errors: ["email"]
visual_regression:
  baseline: s3://artefacts/baseline/
  threshold: 0.03
console_gate:
  max_severe: 0
```

Integrasjoner
- **DevBridge:** kjør QA etter build/preview, returner rapport i chat.
- **GitHub Action:** automatisk PR-sjekk (grønn gate = merge).
- **Marketplace:** playbooks kan inkludere QA-steg som assertions.

Eksempler
- Preview bygger med score 78 (<85) → QA Suite markerer rødt, foreslår lazy-load.
- Skjema lar ugyldig e-post passere → assert feiler, QA kommenterer i PR.
- Visuell diff >3% → rapport med før/etter skjermbilder.

Done-criteria
- Lighthouse-score ≥ definert min på alle profiler.
- Ingen kritiske a11y-funn.
- Alle definerte ruter returnerer 200.
- Forms valideres iht. definisjon.
- QA-rapport genereres og kobles til PR/Playbook run.

35.4 Selector Studio v2 (Stabile UI-selectors)

Formål & verdi
- Minimere automasjonsfeil ved DOM-endringer.
- Gi utviklere og brukere innsikt i hvor robuste selectors er, og automatisk lære fallback-strategier pr. domene.

Brukeropplevelse (UX-flow)
- Når en playbook kjører, vises selector-score i logg (0–100).
- Brukere kan åpne «Selector Studio» i DevTools for å se alternative selectors.
- Advarsler vises dersom en valgt selector er under terskel (<50).

Systemkrav
- Selector-analyser kjøres via DOM-parser med heuristikker.
- ML-modell for fallback-valg trenes på tidligere runs.
- Integrasjon med VisionTool (OCR) for å finne visuelle labels hvis DOM endrer seg.

Datamodell
- `selector_profile(id, domain, preferred_selectors jsonb, fallbacks jsonb, antipatterns jsonb, stability_score float)`
- `selector_log(id, run_id, selector, score, chosen bool, ts)`

Stabilitets-score (beregning)
Formel: `score = 40*ARIA + 25*visibleText + 15*dataAttr + 10*(1-depthNorm) + 10*(1-variance)`
- ARIA (rolle/label tilstede)
- Visible text (stabil tekst)
- Data attributes (eks. data-test)
- DepthNorm (DOM-dybde normalisert)
- Variance (endringsfrekvens på siblings)

Profiler (JSON-eksempel)

```
{
  "domain": "replit.com",
  "preferredSelectors": ["aria/Run", "[data-test=run]"],
  "fallbacks": ["text=Run", "button:has-text('Run')"],
  "antiPatterns": ["div:nth-child(…)"]
}
```

Integrasjoner
- **Executor:** bruker Selector Studio API for å velge beste selector ved runtime.
- **Policy Guard:** kan blokkere automasjon hvis selector-score < terskel.
- **DevTools overlay:** viser score + alternativer når bruker inspiserer elementer.

Eksempler
- Run-knapp på Replit endrer fra `button#run` til `aria/Run` → fallback trigges automatisk.
- Element med `div:nth-child` gir score 20 → Studio foreslår mer stabile alternativer.
- Bruker kan manuelt velge fallback og lagre i profil.

Done-criteria
- Alle selectors logges med score.
- Automatisk fallback fungerer for minst 80% av endringer.
- DevTools-overlay viser alternativer og score.
- Profiler kan lagres og gjenbrukes pr. domene.

35.4 Selector Studio v2 (robuste automasjons-selectorer)

Formål & verdi
- Generere og vedlikeholde **stabile selectorer** som overlever UI-endringer.
- Redusere «flakiness» i automatisering (færre 404/timeout på elementer).
- Lære **domeneprofiler** (Lovable, Bolt, Replit, mgx, osv.) med prefererte mønstre og fallbacks.

Brukeropplevelse (UX-flow)
- **Explain Element-overlay:** når du holder musepekeren over et element, vises kandidat-selectorer, stabilitets-score og "Copy selector".
- **Driftvarsler:** når en playbook feiler pga. «element not found», viser UI en snackbar med «Auto-repair applied (v2)» + link til diff.
- **Profiler per domene:** sidepanel som viser helse (suksessrate, nylige brudd) og top-selectorer.
- **Manuelle overstyringer:** dev kan feste en egendefinert selector med begrunnelse; denne blir del av profilen.

Systemkrav
- **Runtime:** CDP-driver (WebView2/Chromium) for DOM/ARIA/Attrs og screenshot.
- **Vision fallback:** ONNX/Tesseract for OCR av synlig tekst + enkel komponentdeteksjon (knapper, input, tabs).
- **Lagring:** SQLite (per installasjon) for profiler, telemetry og bruddlogger.
- **API:** `selector.generate`, `selector.validate`, `selector.score`, `selector.profile.get/set`.

Datamodell (tabeller)
- `selector_profile(domain, preferred json, fallbacks json, anti_patterns json, updated_at)`
- `element_fingerprint(id, domain, page, role, inner_text_hash, attrs json, path, created_at)`
- `selector_candidate(id, fingerprint_id, selector, score, features json, created_at)`
- `run_event(id, playbook, domain, selector, status, latency_ms, ts)`
- `breakage_event(id, domain, selector, reason, context json, ts)`

Algoritme (scoring)
Mål: rangere kandidater etter robusthet.
Funksjoner: - `ARIA` (rolle + tilgjengelighetsnavn), `visibleText` (OCR + DOM), `dataAttr` (data-*, testid), `structural` (dybde, søskenvariasjon), `stability` (historisk flakiness), `uniqueness` (match count på siden), `actionability` (klikkbar, enabled, i viewport).

Formel (skisse):

```
score = 35*ARIA + 20*visibleText + 15*dataAttr + 10*(1-depthNorm) +
        10*(1-siblingVariance) + 5*uniqueness + 5*stability + 0*actionability
```

Skaleringen (0–1) per feature kalibreres mot historikk. Terskler: `score>=0.70` = **grønn**, `0.50–0.69` = gul (fallback), `<0.50` = rød (kun nød-bruk).

Genererings-pipeline
1) **Oppdag** mål via hint (role/text/near label) eller heuristikk.
2) **Ekstrahér features** fra DOM + OCR.
3) **Bygg kandidater** (ARIA → text → data-attr → CSS/XPath).
4) **Ranger** med scoring.
5) **Valider** top-N (vent på stabil layout, sjekk `isIntersecting`).
6) **Persistér** vinner + alternativer i profilen.
7) **Overvåk** i kjøretid (latens, feil), oppdater `stability` og «promoter» gode alternativer.

Visuell fallback
- Hvis DOM-selectorer feiler, bruk **screenshot-matching** rundt forventet område (anker-element: overskrifter/labels).
- OCR-tekstmatch (`≈` fuzzy) for knapper («Run», «Build», «Preview»).
- Returner en «visual handle» som siste ledd i fallback-kjeden, og logg som `breakage_event`.

Drift & selvreparasjon
- Ved «element not found»:
a) prøv neste kandidat i profilen →
b) generér nye kandidater i sanntid →
c) visuell fallback →
d) *HIL* (human-in-loop) om alt feiler.
- Etter vellykket fallback, lagre ny kandidat med høyere «stability».
- Hvis et domene bryter ofte, foreslå **profiloppdatering** (PR-lignende flyt i UI).

API (skisser)

```
// selector.generate
POST /selector/generate { domain, pageUrl, hint: { role?, text?, near? } }
→ { candidates: [{ selector, score, features }] }

// selector.validate
POST /selector/validate { selector, timeoutMs }
→ { ok: boolean, latencyMs, viewportRect }

// selector.profile.get
GET /selector/profile?domain=replit.com
→ { preferred: [...], fallbacks: [...], anti_patterns: [...] }
```

Domeneprofil (YAML-eksempel)

```
replit.com:
  preferred:
    - "aria/Run"
    - "[data-test=run]"
  fallbacks:
    - "text=/^Run$/"
    - "button:has-text('Run')"
  anti_patterns:
```

```
    - "div:nth-child(…)"
  thresholds:
    min_score: 0.65
    timeout_ms: 4000
```

Pseudokode (C#)

```
SelectorCandidate[] GenerateCandidates(Element e) {
  var list = new List<SelectorCandidate>();
  list.Add(AriaSelector(e));
  list.Add(TextSelector(e));
  list.AddRange(DataAttrSelectors(e));
  list.Add(CssPath(e));
  return list.Select(c => Score(c, e)).OrderByDescending(x =>
x.Score).ToArray();
}

ElementHandle FindElement(TargetHint hint) {
  var profile = LoadProfile(hint.Domain);
  var cands = GenerateCandidates(Discover(hint));
  foreach (var c in Prefer(profile, cands)) {
    var ok = Validate(c.Selector, timeoutMs: profile.TimeoutMs);
    if (ok) return Handle(c);
  }
  return VisualFallback(hint) ?? throw new NotFoundException(hint);
}
```

Telemetry & kvalitet
- Mål: suksessrate pr. domene/selector, median latens, antall `breakage_event` per uke.
- Rapporter «Top offenders» for prioritering.

Done-criteria
- ≥40% færre selector-relaterte feil i playbooks (mot V1 baseline).
- Profiler for minst 4 målplattformer (Lovable, Bolt, Replit, mgx).
- Explain-overlay med «Copy selector» i DevTools.
- Auto-repair løser ≥60% av brudd uten HIL.

35.5 Vibecoding Multi-Agent Platform (innebygd, selvstendig)

Formål & verdi
- Gi en **Replit-lignende** opplevelse direkte i MadEasy – men drevet av et **spesialisert agent-team** (Leader, Product Manager, Architect, Engineer, Data Analyst) som kan planlegge, kode, teste, måle og levere i én sammenhengende flyt.
- Fungerer **uten** eksterne code-tjenester. Støtter lokal kjøring (Windows/Android) + valgfri «cloud burst» for tunge jobber.

Tekstlig arkitektur (høy nivå)

```
User ↔ Team Leader (Orchestrator)
          │
          ├─ Product Manager (Backlog, Acceptance)
          ├─ Architect (ADR, struktur, rammeverk)
          ├─ Engineer (kode, patch, test)
          └─ Data Analyst (metrics, logging, dashboards)

Agents ↔ Tools: FS, Editor, Runner, Preview, QA Suite, Git, Policy Guard
Runtime: Local Sandbox (Windows/Android) + optional Cloud Sandbox
```

UX-flow
1) «New Vibecode Session» → velg **Vibe-profil** (stack, stil, regler).
2) Team Leader oppretter **Project Charter** (mål, akseptkriterier, scope).
3) PM bryter ned i **user stories**; Architect genererer **mappestruktur + ADR**.
4) Engineer implementerer **patches**; QA Suite kjører automatisk.
5) Data Analyst legger inn **instrumentering** (telemetry events), lager **dashboards**.
6) Team Leader **merger/deployer** når policy-gates er grønne.

Vibe-profil (YAML)

```yaml
name: next_tailwind_fast
language: typescript
framework: nextjs
style: minimal|tailwind|shadcn
quality:
  lighthouse_min: 85
  a11y_block: [critical]
  test: jest
constraints:
  deps_max: 12
  ui: wcag_aa
coding_guides: ["no any", "prefer const", "early return"]
```

Project Charter (YAML)

```yaml
goal: "Build a product landing with pricing + contact"
acceptance:
  - "/pricing loads <2s and CLS<0.1"
  - "Contact form rejects invalid email"
scope:
  must: ["landing hero", "pricing tables", "contact form"]
  out: ["blog", "auth"]
```

Runtime/Environment
- **Sandbox** - Windows: `LocalSandbox.Win` (job objects, low IL, fs-jail) + `WSL2Sandbox` (Ubuntu) for Linux-tooling. - Android: `IsolatedProcess` + **WASI** (WebAssembly) for lettvekts-CLI; fallback til sky

for tyngre bygg. - **Runner**: Node.js, Python, .NET, Go støttet (konfigurable baser).
- **Editor**: Monaco med AI-forslag; side-panel for ADR, stories, teststatus.
- **Preview**: intern dev-proxy (port-mapping), HTTPS lokalt; «Open in tab» integrert.
- **FS**: prosjekt-workspace, artefakter, cache; snapshot/restore per story.

Agentroller & RACI
- **Team Leader (TL)**: orkestrerer sprint, prioriterer, tolker Policy Guard-avgjørelser, godkjenner merge.
- **Product Manager (PM)**: skriver user stories/acceptance, oppdaterer backlog og DoD, validerer verdi.
- **Architect (ARC)**: velger rammeverk, lager ADR, definerer mappestruktur og cross-cutting concerns (i18n, a11y).
- **Engineer (ENG)**: implementerer patch, genererer test, fikser build.
- **Data Analyst (DA)**: definerer events, legger inn målepunkter, bygger dashboards og tolker data.

Meldingsskjema (inter-agent)

```
{
  "role": "leader|pm|architect|engineer|analyst",
  "intent": "plan|spec|design|implement|test|measure|review|merge|deploy",
  "topic": "pricing_page",
  "inputs": {"charterId":"c_12","files":["/pages/pricing.tsx"]},
  "proposal": {"steps":["create /pages/pricing","add table","write test"]},
  "diff": "patch://...",
  "asserts": ["route:/pricing","lighthouse.perf>=85"],
  "evidence": [{"type":"qa","ref":"qa://r_77"}],
  "decision": {"status":"approve|revise","notes":"…"}
}
```

Team-loop (kontrollflyt)

```
charter → plan (PM/TL) → design (ARC) → implement (ENG) → test (QA) → measure
(DA)
  → review (TL/PM/ARC) → merge → deploy → retro (metrics)
```

Verktøy (tooling-API)
- `fs.*` (read, write, patch, diff, search)
- `runner.exec(cmd, timeout)` (isoleringspolicy)
- `preview.open()/url()`
- `qa.run(profile)` (LH, axe, routes, forms, visual)
- `git.*` (init, branch, commit, PR)
- `policy.evaluate(action, ctx)` (block/warn/allow)
- `metrics.emit(event, payload)` / `dashboard.create(spec)`

Eksempel – Engineer patch

```
{
  "role": "engineer",
```

```
  "intent": "implement",
  "topic": "contact_form",
  "diff": "diff --git a/pages/contact.tsx b/pages/contact.tsx
+ add HTML5 email validation ...",
  "asserts": ["form_validate:email","route:/contact"]
}
```

QA-kobling
- Hver patch trigger `qa.run(profile)`; status må være grønn før merge.
- Team Leader kan trigge «debatt» (Leader↔Architect) om valg faller under terskel.

Data & analyser
- DA oppretter `metrics.yaml` (event-skjema), binder i kode via SDK, oppretter mini-dashboards.
- Post-deploy: samle baseline vs. ny måling, lag «Impact Note».

Policy & sikkerhet
- Policy Guard på `runner.exec`, `git.push`, `deploy` og `vault.read`.
- Secrets leveres «release-on-use» fra Vault; renses fra logs.
- Sandboxes får ressursgrenser (CPU, RAM, nett).

Sekvens (tekstlig)

```
User → TL: "Bygg pricing"
TL → PM: "Lag stories"
PM → ARC: "Aksept + struktur"
ARC → ENG: "Lag /pricing, table, styles"
ENG → QA: "Run profile:web_app_default"
QA → TL: "LH=89, a11y ok"
DA → TL: "Evt 'view_pricing' event lagt inn"
TL → Git: "PR #12" → Merge → Deploy (policy ok)
```

Done-criteria (MVP → V1)
- Lokal sandbox (Win + Android WASI) støtter Node + Python; preview fungerer.
- Full agent-loop fra charter til merge med QA-gate.
- Vibe-profil styrer stil/rammeverk og valideres i QA.
- Metrics kan defineres og vises i et enkelt dashboard.
- Policy Guard håndhever push/deploy-regler.
