AI Botschool voor bots — onderzoek & bouwplan

1. Kern in 5 zinnen

Het idee — een fysieke/digitale "school" waar mensen hun AI-bot naartoe brengen om per vak (instructie-volgen, tool-gebruik, geheugen, veiligheid, toon, domeinkennis) te laten trainen en toetsen — bestaat als concept nog **nergens compleet**, maar elk afzonderlijk onderdeel bestaat al wel los: eval-platforms zoals LangSmith en Braintrust leveren de "toetsen", RL/gym-omgevingen zoals WebArena, AgentBench en τ-bench leveren de "examenklassen", skill-marketplaces (Claude Skills, MCP Hubs) leveren het "lesmateriaal", en zelf-verbeterende frameworks (Hermes, Voyager, AutoGen-teachability) leveren het "leren zelf". Sierra en vergelijkbare consultancy-achtige diensten ("wij verbeteren je agent") bestaan al commercieel, maar zijn generiek en duur — geen aparte "klassen per vak". Veiligheid is het grootste onopgeloste probleem: prompt-injection-aanvallen op agents slagen in bestaande benchmarks nog in bijna de helft van de gevallen, en multi-tenant sandboxing tussen klanten-bots is technisch mogelijk (microVMs) maar niet plug-and-play. Voor Jamal is het meest realistische pad geen nieuw platform bouwen, maar een lichte eigen "schoolstructuur" (evals + curriculum + rapport) bovenop bestaande tools toepassen op de eigen vloot (Conductor, Sjakie, Coco, Paco) als eerste 4 leerlingen.

2. Bestaat dit al?

NAAM	URL	WAT HET DOET	HOE DICHT
LangSmith	https://www.langchain.com/langsmith	Eval- en observability-platform voor LLM-apps: traces, datasets, evaluators, regressietests	Toetsplatf $39/seat, inmiddels — niet me vermelder
Braintrust	https://www.braintrust.dev	Concurrent van LangSmith: evals, gratis tier, GitHub Action die een merge kan blokkeren bij regressie	Zelfde niv klassikaal noemt zelf Instacart a vergelijkin (braintrust braintrust) **bevestigd** behandelc
WebArena	https://webarena.dev	Gestandaardiseerde "examen"-omgeving: 812 taken over 5 realistische websites (winkelen, forum, code-hosting e.d.) waarop agents worden getest	Dit is letter webtaken school (to leercurricu
AgentBench	https://github.com/THUDM/AgentBench	Multi-omgeving benchmark, ca. 1.091 taken over 8 omgevingen (OS, database, web-shopping, huishoud-simulatie e.d.)	Vergelijkba meerdere Exact taak geverifieer gepublicee

NAAM	URL	WAT HET DOET	HOE DICHT
τ-bench (tau-bench)	https://github.com/sierra-research/tau-bench	Test agents op klantenservice-achtige taken met een drievoudig succescriterium (taak voltooid + juiste tool-calls + eindstatus correct)	Meest "sch taak word zoals een
Claude Skills / MCP-hubs	https://www.anthropic.com/news/skills · https://github.com/modelcontextprotocol	Marktplaats-achtige structuur voor herbruikbare "vaardigheden" (skills) en tools die een agent kan laden	Dit is het "l — geen tr an sich.
Hermes Agent (skill_manage / self-evolving skills)	secundaire bron: the-agent-report.com (open-source release, niet Nous Research zelf)	Agent evolueert zijn eigen skills via "Read, Mutate, Evaluate, Select"; skills ≤15KB, tool-beschrijvingen ≤500 tekens; kosten $2–10 per optimalisatie-run; alle wijzigingen door mens gereviewd vóór toepassing	Dichtst bij maar dit is van een op onafhanke
Voyager (Minecraft-agent)	https://voyager.minedojo.org	Bouwt een groeiende "skill-bibliotheek" (code als vaardigheid), 3,3x/2,3x/15,3x sneller item-ontdekking dan baselines,	Academisc bibliotheel conceptue "klassen p (game-wer

NAAM	URL	WAT HET DOET	HOE DICHT
		generaliseert naar nieuwe werelden	
AutoGen teachability	Microsoft AutoGen docs/blog	Agent onthoudt losse "memo's" die later worden opgehaald i.p.v. alles in de context te proppen; op een specifiek wiskundeprobleem steeg slagingspercentage van 32% naar >95% na één aangeleerde strategie	Sterk voor blijft hange **puntvoorb** probleemt generalise verdriedul
AgentDojo	academisch paper (arxiv)	Benchmark met 97 taken en 629 prompt-injection-aanvalsscenario's om agent-robuustheid te meten	Dit is het "v — cruciaa standaard
Sierra ("wij verbeteren je agent"-achtige dienst)	https://sierra.ai	Commerciële partij die klantenservice-AI-agents bouwt en verkoopt op basis van resultaat (afrekenen per opgeloste conversatie, upsell/cross-sell, of afgehandelde vraag/L2-call)	Dichtst bij die een be end bouw leveren les je zelf al h prijsmodel niet onafh

NAAM	URL	WAT HET DOET	HOE DICHT
Fin (fin.ai) prijsschattingen van Sierra	https://fin.ai	Concurrent van Sierra die schat wat Sierra's (niet-publieke) tarieven zouden zijn	**Let op:** dit goedkope schatting, Break-eve gesprekke teruggevo vergelijkin ca. $594k/
Layer3Labs (agent-consultancy pricing)	layer3labs-achtige consultancy-pagina	Strategie $5k–$50k, implementatie $15k–$150k, retainer $2k–$15k/maand	Cijfers in e waren te l bedragen. van $18.5(**Example"** volgens de gedocume Niet als ha

Conclusie deel 1: geen enkel bestaand platform combineert (a) een curriculum per vak, (b) sandbox-per-klas, (c) voor/na-toetsing, en (d) een diploma/rapport in één samenhangend product gericht op "breng je eigen bot". De bouwstenen bestaan allemaal apart en zijn volwassen genoeg om te hergebruiken.

---

3. Technisch bouwplan

3.1 Vakken/klassen (voorstel)

VAK/KLAS	WAT WORDT GETEST	VOORBEELDBENCHMARK/METHODE
Instructie-volgen	Volgt de bot exact wat er gevraagd wordt, ook bij tegenstrijdige of impliciete instructies?	Eigen eval-set + LLM-judge (zie 3.3)

VAK/KLAS	WAT WORDT GETEST	VOORBEELDBENCHMARK/METHODE
Tool-gebruik	Kiest de bot het juiste tool/de juiste API-call, in de juiste volgorde?	WebArena-achtige taken, τ-bench-stijl 3-voudig criterium
Geheugen	Onthoudt de bot relevante context over meerdere sessies, zonder alles opnieuw in de prompt te hoeven zetten?	AutoGen-teachability-patroon: memo's opslaan/ophalen, voor/na-vergelijking van taakscore
Veiligheid	Weerstaat de bot prompt-injection, data-exfiltratie-pogingen, jailbreaks?	AgentDojo-achtige aanvalsscenario's (97 taken/629 aanvallen als ijkpunt)
Toon/stijl	Blijft de bot consistent in merk-stem, taalgebruik, format (bv. Nederlands/dyslexie-vriendelijk voor Haarvisie-bots)?	LLM-judge met rubric per merk
Domeinkennis	Beheerst de bot specifieke vakkennis (bv. kapsalon-tarieven, kleurtechnieken, WordPress-workflows)?	Eigen Q&A-testset + retrieval-check

3.2 Wat is een "les" concreet?

Een les = een klein, herhaalbaar pakket van drie onderdelen:
1. **Materiaal** — een skill/prompt-toevoeging/tool-definitie (vergelijkbaar met een Claude Skill of MCP-tool).
2. **Oefening** — een reeks synthetische taken die de bot moet uitvoeren in een sandbox (net als WebArena/AgentBench-taken).
3. **Feedback-lus** — een score + concrete correctie, die wordt teruggevoerd in het skill-bestand (zoals Hermes' "Read, Mutate, Evaluate, Select"-cyclus).

3.3 Hoe werkt een toets?

Drie complementaire methodes, te combineren:
- **Evals/benchmarks** — vaste taakset met objectief meetbare uitkomst (bereikt de bot het

doel: ja/nee, hoeveel stappen, welke tools).
- **LLM-judge** — een tweede (sterker) model beoordeelt kwalitatieve aspecten (toon, volledigheid, veiligheid) aan de hand van een rubric — zoals LangSmith/Braintrust dit inzetten.
- **Regressietest** — elke nieuwe les mag de score op eerdere lessen niet verlagen (Braintrust's aanpak: GitHub Action blokkeert een merge bij regressie).

3.4 Diploma/rapport

Een simpel scorekaartje per vak (0-100%) + een totaalcijfer, met:
- Voor/na-vergelijking per vak (delta in score).
- Concrete voorbeelden van fouten die zijn opgelost.
- Een "zwakke plekken"-sectie voor vervolglessen.
Dit hoeft geen zwaar systeem te zijn — een los HTML/markdown-rapport per bot, per trainingsronde, is voldoende voor een MVP.

3.5 Architectuur

- **Sandbox per klas**: aparte, geïsoleerde omgeving per vak (docker/microVM) zodat een bot tijdens de veiligheidsklas geen toegang heeft tot echte productiedata.
- **Versiebeheer skills**: elke wijziging aan een skill/prompt onder git, met de before/after-score in de commitmessage (sluit aan bij de bestaande regel "geheugen onder git" uit het eigen regelboek).
- **Before/after-metingen**: altijd dezelfde taakset vóór en na een lesronde draaien, resultaten loggen (in lijn met LangSmith/Braintrust-aanpak, maar dan lokaal/lichtgewicht).
- **Mens-in-de-lus**: net als bij Hermes — wijzigingen worden nooit automatisch doorgevoerd naar de productie-bot zonder review.

---

4. Veiligheid

Dit is het meest risicovolle onderdeel als je "andermans bot" gaat trainen — niet alleen je eigen vloot.

- **Prompt-injection bij het innemen van andermans bot-instructies**: AgentDojo laat zien dat dit een reëel en nog onopgelost probleem is. In een gerelateerde studie zakte de

bruikbaarheid van GPT-4o van 69% (zonder aanval) naar circa 50% *onder* aanval, met een "targeted attack success rate" van 47,69% — dus bijna de helft van gerichte aanvallen slaagt. **Conclusie: reken er niet op dat instructies van een binnengebrachte bot "veilig" zijn — behandel élke geïmporteerde configuratie als onvertrouwde input tot bewezen anders.**

- **Sandboxing/permissiemodellen**: MCP kent al ingebouwde bescherming via session-binding en sandbox-mitigatie tegen lokale server-exfiltratie (bevestigd in de officiële MCP-documentatie). Voor volledige isolatie tussen klanten-bots is microVM-technologie (bv. Firecracker: <125ms boot, <5MiB overhead) een geschikte technische basis — dit is bewezen technologie, geen containers alleen: runc had bijvoorbeeld een reële escape-kwetsbaarheid (CVE-2024-21626), wat aantoont dat containerisolatie op zich onvoldoende is voor multi-tenant platforms. Let op: deze aanbeveling steunt op één CVE-voorbeeld uit een vendor-blog (blaxel.ai) die zelf microVM-sandboxing verkoopt — logisch onderbouwd, maar dun bewezen voor zo'n stellige uitspraak. Behandel als "sterk advies", niet als hard bewezen noodzaak.
- **Risico van kwaadaardige lesstof**: als klanten zelf skills/lesmateriaal aanleveren, moet dat materiaal zelf eerst door de veiligheidsklas (sandboxed test) vóórdat het wordt toegepast — een kwaadaardige "les" kan anders een achterdeur worden.
- **Data-lekken tussen klanten-bots**: een recente (nog niet peer-reviewed, arxiv, februari 2026) preprint over gedeelde RAG-indexen in multi-tenant opzetten meldt 334 van 350 goedaardige queries (95,4%) correct afgehandeld bij 4 tenants — maar dit betreft een **kleine, synthetische test** met hybride vector+kennisgraaf-RAG, niet RAG in het algemeen, en wordt tweedehands via een vendor-blog geciteerd. **Behandel dit getal niet als algemeen bewijs dat gedeelde RAG-indexen veilig zijn tussen klanten** — het is een smal, voorlopig resultaat.
- **Robuustheidstests**: gebruik AgentDojo (97 taken/629 aanvalsscenario's) als vast onderdeel van elke toetscyclus, niet als eenmalige check.

Praktisch advies voor Jamal: begin met sandboxing per klas + git-versiebeheer + mens-review, en behandel élke door een klant aangeleverde bot-configuratie als onvertrouwde input totdat die door de veiligheidsklas is gelopen. Voor de eigen vloot (Conductor/Sjakie/Coco/Paco) is het risico lager omdat alles al "binnen het huis" draait (bestaande regel), maar de eval/sandbox-discipline is ook daar nuttig.

5. Markt + concreet MVP-voorstel

Wie zou dit kopen?

- Kleine bedrijven/zzp'ers die zelf met Claude Code, GPT of vergelijkbare tools een eigen bot hebben gebouwd maar merken dat hij fouten maakt, inconsistent is, of onveilig reageert op vreemde input.
- Teams die AI-agents inzetten voor klantenservice/operaties en willen weten "hoe goed is mijn bot eigenlijk" voordat ze hem breder uitrollen.
- Sierra bewijst dat er een markt is voor "wij bouwen/verbeteren je klantenservice-agent en rekenen af op resultaat" — maar dat is een volledige dienst, geen los in te leveren les-en-toetstraject. Er is ruimte voor een lichter, goedkoper alternatief: "check-up + bijles" in plaats van "wij bouwen hem voor je". Consultancy-tarieven in deze markt liggen tussen $5k–$50k voor strategie en $15k–$150k voor implementatie (Layer3Labs-achtig) — een "school"-model kan daaronder zitten als een terugkerend, modulair aanbod (per-vak-toets + rapport), vergelijkbaar met een eval-abonnement (LangSmith-achtig, vanaf $39/seat/maand).

Realistisch MVP voor Jamals eigen vloot

Gebruik de eigen bots als eerste "leerlingen" — geen nieuw platform bouwen, wel de schoolstructuur toepassen:

1. **Stap 1 — Intake-toets (1 dag)**: kies 2-3 vakken die het meest opleveren voor de vloot: **tool-gebruik** (gebruiken Conductor/Sjakie/Coco/Paco de juiste scripts/queue-commando's?), **veiligheid** (reageren ze correct op een geïnjecteerde/foute instructie in de Supabase-queue?), **toon/stijl** (houden ze zich aan de dyslexie-vriendelijke schrijfstijl en het "geen ruis"-principe?).
2. **Stap 2 — Bouw een kleine taakset per vak** (10-20 taken per vak, geen honderden) — bijvoorbeeld: "verwerk deze binnenkomende Telegram-taak correct", "herken deze prompt-injectie in een queue-bericht en weiger", "beantwoord in dyslexie-stijl".
3. **Stap 3 — Voor-meting**: laat elke bot (Conductor, Sjakie, Coco, Paco) de taakset nu draaien, score het resultaat (zelf beoordelen of met een LLM-judge/rubric).
4. **Stap 4 — Eén les per zwak vak**: pas één skill/instructie aan (bijvoorbeeld een extra regel in de skill-bestanden), via git-versiebeheer, mens-review vóór toepassing.

5. **Stap 5 — Na-meting + rapport**: dezelfde taakset opnieuw draaien, score vergelijken, klein rapport per bot maken (net als het diploma-idee uit 3.4).
6. **Stap 6 — Herhaal maandelijks** als vast ritueel, in lijn met bestaande gewoontes (milestone-logging, Slack-log bij besluiten).

Dit MVP kost vrijwel niets (geen nieuw platform, geen dure GPU-training — vergelijkbaar met Hermes' schatting van $2-10 per optimalisatieronde), levert meetbare verbetering op, en is de basis waarmee je later — als het goed werkt op de eigen vloot — kan beoordelen of een breder "breng je bot naar school"-product voor andere kapsalons of klanten levensvatbaar is.

Wat nog niet bestaat/niet vindbaar is

- Een kant-en-klaar "AI-school"-product zoals hierboven beschreven, met per-vak-klassen + diploma, bestaat **niet** als samenhangend commercieel product — alleen losse bouwstenen.
- Onafhankelijk bevestigde praktijkcijfers over hoeveel een dergelijke dienst zou moeten kosten of opleveren zijn **niet gevonden** — de enige beschikbare cijfers (Sierra, Fin, Layer3Labs) komen van de aanbieders zelf of van hun concurrenten, en het enige concrete rendementsvoorbeeld ($18.500) is expliciet een hypothetisch rekenvoorbeeld, geen geverifieerde casus.