Helder Orchestration Model

Implementatiedocument voor Hugoherbots.ai (Chat Engine)

Versie: v1.0 | Datum: 19 January 2026

Dit document beschrijft hoe we de chat engine logischer aansturen door orchestration expliciet te maken in configuratie (coach_overlay_v3.json), met context-layers, roleplay-gates en artifacts als brug tussen fases 2-3-4.

1. Doel en scope

Doel: de chat engine logischer aansturen door de oefenvorm per techniek expliciet te maken. Dit voorkomt artificiele rollenspellen op micro-technieken en maakt geloofwaardige, sequentiele training over EPIC-fases (2-3-4) mogelijk.

- Scope: configuratie + backend orchestration (server/v2) + sessie-opslag.
- Niet in scope: UI design details; wel de benodigde UX-berichten bij gates.
- Compatibiliteit: V3 moet naast V2 kunnen draaien (feature flag).

2. Probleem dat we oplossen

In de huidige V2 flow is orchestration te grof: alles is ofwel coach chat, ofwel (te snel) roleplay. Daardoor ontstaan vier structurele problemen.

1. Te vroeg roleplay: deeltechnieken zoals 2.1.1 (feitgerichte vragen) zijn geen natuurlijke rollenspellen op zich.
2. Geen gates: fase 3 roleplay kan starten zonder dat discovery (fase 2) is gedaan of samengevat.
3. Geen cross-fase continuiteit: de klant in fase 3/4 voelt als een nieuwe persona, los van de eerdere discovery.
4. Onvoldoende context voor realisme: pains, baten, typische bezwaren en lastige openingsvragen ontbreken vaak.

3. Designprincipes

- Orchestration hoort in configuratie (overlay), niet verspreid in hardcoded if/else in engines.
- Progressive disclosure: verzamel alleen context die nodig is voor de volgende stap; niet alles upfront.
- Roleplay is een macro-oefenvorm: pas starten wanneer context en prerequisites aanwezig zijn.
- Artifacts zijn de brug tussen fases: discovery_brief (2->3) en offer_brief (3->4).
- Backwards compatible: V2 velden blijven geldig; V3 velden zijn optioneel tot feature flag aan staat.

4. Concepten en definities

4.1 Modi

- CONTEXT_GATHERING: Hugo verzamelt context via LSD (Luisteren, Samenvatten, Doorvragen).
- COACH_CHAT: Hugo coacht (Socratisch) en helpt de vertaalslag naar de realiteit van de verkoper.
- ROLEPLAY: Hugo verdwijnt volledig; AI speelt de klant/prospect.
- FEEDBACK: nabespreking na roleplay (observatie, reflectieve vraag, micro-experiment).

4.2 Learning functions (oefenvormen)

- COACH_TRANSLATE: uitleg + vertaling naar de specifieke context van de gebruiker (geen roleplay).
- MICRO_DRILL: korte oefening op formulering/flow binnen 1 techniek (optioneel met mini-simulatie).
- ROLEPLAY_DRILL: rollenspel op 1 techniek met ingestelde klanthouding (bv. bezwaar).
- ROLEPLAY_INTEGRATED: rollenspel over meerdere technieken (bundel/fase) met EPIC-progressie.

4.3 Context layers

Context wordt opgebouwd in lagen. Elke laag bestaat uit concrete slots/velden die de engine kan verzamelen en vervolgens gebruiken in coaching en roleplay.

Layer	Doel	Voorbeeldslots
base	Personalisatie en setting	sector, product, klant_type, verkoopkanaal, ervaring
scenario	Gespreksrealiteit (start van gesprek)	afspraak_type, rol gesprekspartner, lastige_openingsvragen, DMU
value_map	Probe/Impact laten landen op baten	pains, baten, koopredenen_recent, usp_naar_baat_map
objection_bank	Realistische afritten/closing	bezwaren_typisch, twijfels_typisch, uitstelredenen_typisch
offer_map	Fase 3/4 coherent maken	oplossing_kern, voordelen, bewijsvoering, prijsrange

4.4 Artifacts

Artifacts zijn gestructureerde samenvattingen die door de engine worden opgebouwd en doorgegeven tussen fases. Ze maken sequentiele training mogelijk zonder dat context telkens opnieuw moet worden opgevraagd.

- scenario_snapshot: bewaart persona-seed en scenario-parameters zodat dezelfde klant hergebruikt kan worden.
- discovery_brief: output van fase 2 (pains, baten, criteria, urgentie). Vereist voor fase 3 roleplay.
- offer_brief: output van fase 3 (OVB en next step). Vereist voor fase 4 roleplay.

5. Coach overlay V3 - wat komt erbij

coach_overlay_v3.json behoudt de V2 velden (default_mode, roleplay_capable, roleplay_default) en voegt een orchestrator sectie toe per techniek.

Belangrijkste nieuwe velden:

- context_depth: LIGHT | STANDARD | DEEP (hoeveel context minstens nodig is).
- learning_function: COACH_TRANSLATE | MICRO_DRILL | ROLEPLAY_DRILL | ROLEPLAY_INTEGRATED.
- context_layers_required: lijst van context layers die aanwezig moeten zijn.
- artifacts_in/artifacts_out: prerequisites en outputs.
- roleplay_gates: expliciete voorwaarden voor mode switch naar ROLEPLAY.
- persona_policy: wanneer persona hergebruiken vs nieuw genereren.
- recommended_bundle: verwijs micro-technieken naar de bundel waar integratie-oefening zinvol is.

Voorbeeld (conceptueel):

```
{
```

```
    "2.1.1": {
      "default_mode": "COACH_CHAT",
      "roleplay_capable": false,
      "orchestrator": {
        "learning_function": "COACH_TRANSLATE",
        "context_depth": "LIGHT",
        "recommended_bundle": "2.1"
      }
    }
  }
```

6. Orchestration flow per fase

Dit hoofdstuk beschrijft de gewenste standaardflow. De orchestrator kan hiervan afwijken op basis van gebruikersintentie (bv. ervaren gebruiker die direct wil oefenen).

Fase	Primaire oefenvorm	Minimale prerequisites	Outputs (artifacts)
0 (pre-contact)	COACH_TRANSLATE / MICRO_DRILL	base (prospect/markt info)	scenario_snapshot (optioneel)
1 (opening)	MICRO_DRILL	base + scenario (light)	scenario_snapshot
2 (discovery / EPIC)	ROLEPLAY_INTEGRATED	base + scenario + value_map + objection_bank (standard/deep)	discovery_brief + scenario_snapshot
3 (aanbeveling / OVB)	ROLEPLAY_INTEGRATED	discovery_brief + offer_map	offer_brief
4 (closing)	ROLEPLAY_DRILL + ROLEPLAY_INTEGRATED	offer_brief + objection_bank	n.v.t. (feedback + next steps)

7. Roleplay gates

Roleplay mag alleen starten als gates slagen. Gates zijn expliciet en user-friendly: de engine blokkeert niet stil, maar legt uit wat ontbreekt en biedt een korte weg om het aan te vullen.

- Gate type A: context gate (vereiste context layers / depth).
- Gate type B: artifact gate (artifacts_in aanwezig).
- Gate type C: technique gate (learning_function staat roleplay niet toe).
- Gate type D: phase gate (initial_epic_phase consistent met fase en artifacts).

Voorbeeld gate-check (conceptueel):

```
    if learning_function in ["COACH_TRANSLATE"]:
        deny_roleplay("Deze oefening is bedoeld als vertaalslag, niet als rollenspel.")

    if "discovery_brief" in artifacts_in and not session.artifacts.discovery_brief:
        deny_roleplay("We missen nog de discovery samenvatting uit fase 2. Wil je eerst fase
2 oefenen of zal ik gericht context vragen?")

    if not has_context_layers(context_layers_required):
        ask_context_for_missing_layers()
```

8. Artifact lifecycle en opslag

8.1 Waar komen artifacts vandaan?

- scenario_snapshot: bij start van een roleplay (of na context gathering), serialize persona seed + scenario parameters.
- discovery_brief: na een ROLEPLAY_INTEGRATED in fase 2, automatisch genereren op basis van de sessie (gesprekhistoriek + evaluator signals).
- offer_brief: na fase 3 roleplay, automatisch genereren op basis van OVB segmenten en overeengekomen next steps.

8.2 Persistente opslag (cross-session)

Als gebruikers fase 2 vandaag doen en fase 3 morgen, moet discovery_brief persistent zijn. Daarom onderscheiden we:

- Session artifacts: geldig binnen dezelfde sessie (v2_sessions.artifacts).
- User artifacts/context: optioneel per user (users.persistent_context) voor hergebruik over sessies.

9. Implementatieplan (gefaseerd)

Aanbevolen aanpak: begin met V3.0 (gates + config parsing) en breid daarna uit met artifacts en extra context layers. Dit minimaliseert risico en houdt V2 stabiel.

Release	Doel	Belangrijkste changes	Risico
V3.0	Orchestrator leest V3 + enforce basis gates	orchestrator service, checkRoleplayGates(), initial_epic_phase support	Medium
V3.1	Session artifact storage	DB: v2_sessions.artifacts jsonb + CRUD + scenario_snapshot	High

V3.2	discovery_brief generatie	post-roleplay summarizer + evaluator ondersteuning + gate op fase 3	High
V3.3	Extended context layers	context_engine uitbreiden: value_map + objection_bank + offer_map (progressive)	Medium
V3.4	Cross-session persistence	users.persistent_context jsonb + select/reuse policy	Medium
V3.5	UI hints	recommended_bundle: banners/suggesties + video deep links	Low

10. Backend wijzigingen (concreet)

10.1 Nieuwe modules

- server/v2/orchestrator.ts: laad coach_overlay_v3, bepaal next mode, enforce gates, select context prompts.
- server/v2/artifact-manager.ts: opslaan/lezen van session artifacts en (optioneel) user artifacts.
- server/v2/context-layers.ts: definieer slots per layer + missing slots detection.

10.2 Aanpassingen aan bestaande engines

- context_engine.ts: uitbreiden met progressive context voor value_map/objection_bank/offer_map.
- roleplay-engine.ts: respecteer initial_epic_phase en persona_policy (reuse scenario_snapshot).
- coach-engine.ts: support auto-offer policy (turn counter) en start roleplay pas na gates.
- evaluator.ts (optioneel): structured outputs gebruiken om discovery_brief betrouwbaarder te genereren.

11. Data model (DB) wijzigingen

Minimale DB wijziging voor V3.1: artifacts opslaan per sessie.

```
ALTER TABLE v2_sessions
  ADD COLUMN artifacts jsonb DEFAULT '{}'::jsonb;
```

Optioneel voor cross-session (V3.4):

```
ALTER TABLE users
  ADD COLUMN persistent_context jsonb DEFAULT '{}'::jsonb;
```

12. UX/Copy voor gate-berichten

Gate-berichten moeten concreet zijn en altijd een next step bieden. Voorbeelden:

- "Voor deze oefening is een rollenspel pas zinvol als we je pains en baten scherp hebben. Zullen we dat eerst kort in kaart brengen?"
- "Ik kan fase 3 pas realistisch spelen als ik weet wat we in fase 2 ontdekt hebben. Wil je eerst fase 2 oefenen of zal ik je 3 gerichte vragen stellen om die samenvatting te maken?"
- "Deze techniek is een micro-vaardigheid. Ik stel voor: eerst samen 3 sterke varianten formuleren, en daarna oefenen we het in de Explore-bundel."

13. Teststrategie

- Unit tests: checkRoleplayGates() combinatoriek (context missing, artifact missing, learning_function restrictions).
- Integration test: fase 2 integrated roleplay -> discovery_brief opgeslagen -> fase 3 roleplay toegestaan en persona hergebruikt.
- Regression: V2 gedrag blijft identiek wanneer feature flag uit staat.

14. Appendix - suggested artifact schemas

14.1 scenario_snapshot

```
{
  "persona_seed": "string",
  "segment": "SME|midmarket|enterprise",
  "sector": "string",
  "role": "decision_maker|influencer|user",
  "dynamics": {"rapport": 0.3, "tension": 0.2, "commit": 0.1},
  "created_at": "ISO-8601"
}
```

14.2 discovery_brief

```
{
  "as_is": "samenvatting huidige situatie",
  "pains": ["..."],
  "desired_outcomes": ["..."],
  "baten": ["..."],
  "criteria": ["budget", "timing", "risk"],
```

```
  "urgency": "low|medium|high",
  "proof_points_needed": ["..."],
  "notes": "vrije notities"
}
```

14.3 offer_brief

```
{
  "solution": "...",
  "ovb": [
    {"expectation": "...", "oplossing": "...", "voordeel": "...", "baat": "..."}
  ],
  "risks_remaining": ["..."],
  "next_step": "meeting|proposal|order",
  "timeline": "..."
}
```

Einde document.