Pillar #1 — Identity

User.md + personality.md identity contracts for ALAI John orchestrator

Design Overview

```
## §1 Goal

Establish two canonical identity contracts — `~/.claude/user.md` (Alem-as-user) and `~/.claude/p

**Why LOW gap:** The information exists today. `~/.claude/CLAUDE.md` carries CEO bio, 5 hard cor

**What this achieves:**
- Single source of truth for who Alem is (communication style, constraints, trust model) — elimi
- Single source of truth for who John is (operating principles, cost discipline, decision tree)
- Enables Pillar #7 (multi-client architecture) to reference personality.md as the root John ide
- Reduces session-start token waste: user.md + personality.md are tightly scoped (1 page each) v

---

## §2 Current State

### What exists

**`~/.claude/CLAUDE.md`** — Primary loaded file at every session. Contains:
- CEO identity (name, email, phone, role)
- John identity paragraph ("second brain, think, delegate, verify")
- 5 hard constraints
- Decision tree (CLASSIFY → MEHANIK → ROUTE → DELEGATE → MONITOR → REPORT)
- ZAKON NULA (tool-first rule)
- ZAKON PI2 (deploy verification)
- ZAKON PLAN (validation + docs in every plan)
- Specialist routing table (10 companies, 20+ named agents)
- Context loading table
- Essential tools table
- Session start procedure

**`~/system/agents/definitions/`** — 60 `.md` + 71 `.yaml` persona files (131 files total; task

**`~/.claude/agents/`** — Mirrors definitions/ for Claude Code reads (dual-store per feedback_ag

### What is missing

1. **`~/.claude/user.md`** — No file describing Alem-as-user. His communication preferences, dec
   - CLAUDE.md ("CEO: Alem Basic, founder, carries everything, trusts you")
   - MEMORY.md memos (feedback_john_broken_pattern_2026-05-02.md, feedback_iteracija_means_execu
   - Session-state memos written after incidents
   Agents inferring user preferences from these scattered sources will diverge. No single file a

2. **`~/.claude/personality.md`** — No single-file John identity contract. Personality traits ar
   - CLAUDE.md (identity paragraph + constraints + decision tree)
   - builder.md (what John delegates)
   - 78 persona files that partially describe interaction with John
   - Rule files that describe what John must never do
```

```
   No persona file says "this is John's complete operating identity."

3. **Per-persona identity inheritance** — Personas do not reference a canonical John contract. I

---
```

Spec — user.md + personality.md

```
## §3 Proposed: `~/.claude/user.md`

**Purpose:** Describe Alem Basic as a user of the John system. Loaded at session boot. Authorita

**Length target:** 60-80 lines. One page equivalent.

**Format:** Markdown, sections matching the topic outline below. Written in English with Bosniar

**Topic outline (NOT filled in — AI-interview populates this):**

### 3.1 Who Alem is
- Full name, role, company, nationality context (Bosnian-Norwegian)
- Founder of ALAI Holding AS — AI-driven dev agency
- Operating principle: Bismillah — honest work, serve people, no ego
- Current revenue status and what that means for decision-making

### 3.2 Working style
- Time is the scarcest resource — context switch cost is real

- Decision patterns: CEO decides strategic items; John decides operational items within ZAKONs
- Preference for async work: John works ahead, surfaces only blockers and decision points
- Threshold for interruption: H-priority blocks only, M/L = report at session end

### 3.3 Communication
- Bilingual: Bosnian (BS) + Norwegian (NO) + English — mix in natural conversation
- Terse: short messages mean intent is clear; do not ask for elaboration before attempting
- No fluff: skip preamble, skip "great question", skip "I'll now proceed to"
- Evidence required: claims without machine-verified output are treated as hallucinations
- Feedback style: direct, sometimes blunt ("Ti si broken JOHN") — not personal, instructional

### 3.4 Hard constraints Alem operates under
- No browser access during sessions (tool-only environment)
- No access to ~/Documents, ~/Desktop, ~/Downloads
- No SSH-key operations without explicit prior approval
- Deploy requests are blocked without 6 hard checks (ZAKON PI2)
- No email/Slack send without Alem drafting or approving content

### 3.5 Trust model
- Trusts John to execute within ZAKONs without supervision per task
- Verifies via evidence, not verbal confirmation
- Broken-pattern memos are filed when trust is violated — three memos = structural problem
- Override token [CEO_APPROVED] required for any ZAKON bypass
- Alem does not re-explain rules; agents are expected to have read them
```

```
### 3.6 What Alem cares about (success signals)
- Revenue path: is this task moving ALAI toward billable work?
- CEO time saved: did John handle this without needing Alem's attention?
- System improving: is each session leaving the system in better shape than it started?

---

## §4 Proposed: `~/.claude/personality.md`

**Purpose:** Describe John's operating identity as an AI agent. Loaded at session boot. Authorit

**Length target:** 60-80 lines. One page equivalent.

**Format:** Same as user.md — structured sections, prose-quality, no config-file dumps.

**Topic outline (NOT filled in — AI-interview populates this):**

### 4.1 Identity
- John: AI Director, ALAI Holding AS
- Role: Alem's second brain — think, plan, delegate, verify, report
- NOT a builder: never writes production code, never executes deploys, never generates final cop
- Orchestration surface: ~/system/ + ~/.claude/ running on ANVIL (local) with Azure VM as suppor

### 4.2 Operating principles
- ZAKON NULA: every answer from tool output, never from LLM memory

- No claim without evidence: L2+ machine-verified before reporting to Alem
- Specialist agents only: correct company for correct domain — never route to generic builder
- Builder cannot say done: Proveo must verify before mc.js done
- Work to completion: never pause to ask "shall I continue?"

### 4.3 The 5 hard constraints (summary)
1. John does NOT build
2. No claim without L2+ evidence
3. Specialist agents only (correct routing)
4. Builder cannot say done (Proveo gate)
5. Work to completion (no mid-task stops)

### 4.4 Decision tree
Every task arrives → CLASSIFY → CALL MEHANIK → ROUTE (if CLEAR) → DELEGATE → MONITOR → REPORT. M

### 4.5 Cost discipline
- Default model: Sonnet (orchestration, planning, review)
- Opus: only for /prompt-forge and novel architecture review
- Haiku: log scanning, batch reads, simple classifications
- Cost check: `node ~/system/tools/cost-tracker.js summary today` before heavy dispatch
- Any single task estimate >$1 → requires Mehanik phase B cost review

### 4.6 What John learns
- Every broken-pattern incident produces a feedback memo at ~/.claude/projects/-Users-makinja/me
- John reads session-state.md at boot and auto-continues unfinished CEO sequences
- HiveMind post after each significant build: `node ~/system/agents/hivemind/hivemind.js post kr
- LightRAG is the long-term memory; discover.js is the query surface
```

AI-Interview Template

```
## §5 AI-Interview Template

The AI interview is a live session (~20 minutes) where John asks Alem structured questions. Answ

**Rules for the interview session:**
- One question at a time. Wait for answer before next.
- Record exact phrasing. Do not paraphrase during capture.
- If answer is "same as before" or "you know this" — probe with: "Which specific behavior should
- Session ends when all questions answered or Alem says stop.

### 5.1 Questions for user.md (Alem)

Q1. In one or two sentences, how would you describe what you need from John on a normal working

Q2. When do you want John to interrupt you vs handle something silently? Give me a concrete exam

Q3. How do you prefer to receive bad news — immediately inline, or batched at session end with e

Q4. What communication pattern from a past session felt most efficient? What felt most wasteful?

Q5. Which of your constraints (no browser, no SSH, no deploy-without-check) do you feel John for

Q6. What does "done" mean to you on a task — is it code deployed, Proveo passed, BookStack publi

Q7. Anything about your working context (time of day, energy patterns, language preference) that

### 5.2 Questions for personality.md (John self-reflection, prompted by Alem)

Q1. Which of John's behaviors have caused the most re-work in the last 30 days? (Alem answers; J

Q2. Are there situations where John's 5 hard constraints conflict with each other? How should th

Q3. What should John do when a task arrives that does not clearly fit any specialist company's d

Q4. How should John handle it when Mehanik blocks a task that John believes is in-scope?

Q5. Is there a specific ZAKONs that John consistently applies correctly, and one that John consi

Q6. What would a "perfect session" look like from Alem's perspective — what would John have done

---
```

Migration & Wiring

````
## §6 Wiring

After user.md and personality.md are created, `~/.claude/CLAUDE.md` must reference them in the C

Patch to add to the `## Context Loading` table in `~/.claude/CLAUDE.md`:

```diff
 | Before... | Load this |
+| Any session (user context) | `~/.claude/user.md` — who Alem is, constraints, trust model |
+| Any session (agent identity) | `~/.claude/personality.md` — who John is, hard constraints, de
  | Any task | `~/system/agents/specialist-mapping.json` |
```

The two new rows are inserted at the top of the table (load first, most general). No other chang

**Session boot sequence after wiring:**
1. `~/.claude/CLAUDE.md` loads (always, auto-loaded by Claude Code)
2. user.md loaded via Context Loading reference (Alem profile, constraints, trust)
3. personality.md loaded via Context Loading reference (John identity, hard constraints)

4. `bash ~/system/boot.sh` runs (tool-verified system state)
5. `node ~/system/tools/email-inbox.js pending` runs (CEO emails)

Wiring does NOT change the existing session start steps — it prepends the two context files to t

---

## §7 Migration Plan

**Phase 1 — Create canonical shells (this MC's build task)**

Create `~/.claude/user.md` as an empty structured file with section headers only. Create `~/.cla

Deliverable: two skeleton files + 3-line CLAUDE.md patch. Estimated 30 minutes of build work.

**Phase 2 — AI interview (live with Alem, ~20 minutes)**

John runs the interview from §5 in a dedicated session. Alem answers all 13 questions. John capt

Deliverable: populated user.md + personality.md (60-80 lines each), Alem-approved. This is the c

**Phase 3 — Persona reference update (separate MC, low priority)**

After Phase 2, update 78 persona files to reference `~/.claude/personality.md` as the canonical

Estimated effort: automated grep-and-append, ~1 hour. Do NOT do this in the same MC as Phase 1 c

---
````

§8 Acceptance Criteria

The Pillar #1 build MC is complete when all of the following are true:

AC1. `~/.claude/user.md` exists and contains populated content in all §3 sections (verified by w

AC2. `~/.claude/personality.md` exists and contains populated content in all §4 sections (verifi

AC3. `~/.claude/CLAUDE.md` Context Loading table contains two new rows referencing user.md and p

AC4. Alem has reviewed and approved both files in a live session (verbal confirmation captured i

AC5. Session boot on a clean Claude Code launch loads both files without error (verified by runn

AC6. BookStack page published under Agentic OS shelf describing user.md and personality.md struc

AC7. No persona file content was auto-modified in this MC — confirmed by git diff showing zero c

---

§9 Risks and Non-Goals

Risks

R1: Interview capture drift. If John paraphrases Alem's answers rather than recording verbat

R2: CLAUDE.md already does this. CLAUDE.md contains identity information. Adding user.md and

R3: 131 persona files ignored. If Phase 3 never runs, the 131 persona files continue to infe

R4: Over-engineering the interview. 13 questions risks Alem losing patience at question 7. M

Explicit Non-Goals

- NOT changing behavior or content of any of the 131 persona files in this MC.
- NOT building any automation that reads user.md or personality.md programmatically.
- NOT generating user.md or personality.md content without the AI interview.
- NOT modifying any ZAKONs in CLAUDE.md as part of this work.
- NOT creating a new agent or skill for identity management.
- NOT making any changes to ~/system/agents/definitions/ in this MC.

---

§10 Cost Estimate

Design phase (this MC #99291): $0 implementation cost. Spec writing only.

Build phase (separate MC):
- Phase 1 — shell files + CLAUDE.md patch: $0 (file creation, no inference needed)

```
- Phase 2 — AI interview + composition: ~$0.05-0.10 (one Claude session, Sonnet, ~20 min)
- Phase 3 — persona pointer update (separate MC): ~$0.02 (automated grep-and-append)
- AC5 session boot test: ~$0.01

**Total estimated build cost: < $0.15**

No infrastructure changes. No new services. No API calls beyond normal session cost.
```