SEO Readiness Portal — Self-Serve Intake (Architecture + Operator Runbook)

SEO Readiness Portal — Self-Serve Intake (Architecture + Operator Runbook)

Shipped: 2026-06-04
Image: `alairegistry.azurecr.io/seo-readiness-portal:20260604-selfserve-intake`
Validation: MC #102923 Proveo PASS — `/tmp/alai/seo-wsv-evidence/VALIDATION-REPORT.md`
Deploy Evidence: `/tmp/alai/seo-deploy-102929/FLOWFORGE-DEPLOY-REPORT.md`

Overview

The self-serve intake feature enables **Asmir (SnowIT sales) to send zero-effort referrals** to potential SEO clients. Asmir creates a client record in the SEO Portal /partners workspace, the system automatically sends a branded magic-link email, and the client fills out a web-based intake form that triggers an automated SEO audit pipeline — all without Asmir lifting a finger after the initial client creation.

Asmir Zero-Effort Workflow

1. **Create client:** Asmir visits `https://seo-tools.snowit.ba/partners/clients/new`, enters client name + email.
2. **System auto-sends:** Intake magic-link email sent immediately (SnowIT-branded, subject "SEO Readiness Assessment for {clientName}").
3. **Client completes intake:** Client clicks link → fills web form (or chats with AI assistant) → submits.
4. **Auto-pipeline runs:** Live SEO audit + findings + draft report generated automatically.

5. **Asmir reviews:** Asmir sees client status change from "link_sent" → "intake_submitted" → "report_ready" in `/partners/referrals` dashboard.

Total Asmir effort: Type name + email. Done.

Feature Scope (WS-A/B/C)

WS-A: Magic-Link Token + Email Delivery

Deliverable: HMAC email-bound re-usable intake token + public `/intake/[token]` route + SnowIT-branded link email on client create.

- **Token model:** `src/lib/auth/intake-token.ts`
  - HMAC-SHA256 signed with `INTAKE_TOKEN_SECRET` (64-char hex, Azure env var).
  - Payload: `{clientId, email, expiresAt}`. Default expiry: 30 days.
  - Token hash stored on `Client.intakeTokenHash`; plaintext NEVER persisted.
  - Re-usable: Same token works for multiple submissions (versioned append model).
- **Public route:** `src/app/intake/[token]/page.tsx` (Next.js App Router)
  - Token verification: `verifyIntakeToken(token)` → clientId + email.
  - Invalid/expired/forged token → 200 error page ("Link not valid / expired"), no data leak.
  - Valid token → loads single-client intake form pre-filled with client name/email.
  - Cross-client isolation: Token A cannot access Client B's data (clientId derived from token, NEVER from URL/body).
- **Email trigger:** `src/app/clients/new/actions.ts` `createClient()`
  - Calls `~/system/tools/mail-native.js` with SnowIT-branded template.
  - Subject: "SEO Readiness Assessment for {clientName}".
  - Body: SnowIT logo + personalized intro + magic link to `https://seo-tools.snowit.ba/intake/{token}`.
  - **KNOWN GAP:** Email currently sent via `--account alai` sender (snowit.ba mailbox not yet provisioned). Body is SnowIT-branded; sender is alai.no. Follow-on MC to provision Asmir@snowit.ba sender mailbox.
- **Persistence:** `src/lib/workspace/persistence.ts`
  - Versioned `IntakeSubmission[]` appended to `Client.intakeSubmissions` array.
  - Each submission timestamped; re-submit creates new version, does NOT overwrite.

Evidence: `/tmp/alai/seo-ws-a-evidence/`
Validation: Playwright 10/10 PASS, `validate-magic-link.ts` 33/33 PASS.

WS-B: SnowIT White-Label AI Chatbot

Deliverable: SnowIT-branded AI assistant on intake page (Ollama-first tier-router, rate-limited, secret-guarded, forbidden-claim-guarded).

- **UI Widget:** `src/components/chatbot/IntakeChatWrapper.tsx`
  - Floating Action Button (FAB) in bottom-right corner of intake page.
  - Slide-out chat panel: "SEO Assistant" header + "Powered by Snowit" footer.
  - Collects same fields as intake form (`IntakeFormFields`).
  - Two-way sync: chat updates form, form updates chat context.
- **API Route:** `src/app/api/intake-chat/route.ts` (POST)
  - **Token-gated:** Requires `token` in body; calls `verifyIntakeToken()`.
  - Invalid/expired token → 401 `IntakeChatErrorResponse {code: 'token_invalid'}`.
  - ClientId derived from token, NEVER from body (single-client scope enforced).
  - **Rate-limited:** 10 messages / 60 seconds per token (in-process sliding window). 11th message → 429 `{code: 'rate_limited'}`.
  - **LLM tier-router:** `src/lib/chat/tier-router.ts` (Ollama FORGE → Groq → Anthropic Haiku waterfall).
  - **Secret-scan guard:** User input + assistant output scanned for `/\b(password|api[_-]?key|secret|token|bearer)\b/i`. Match → pre-canned refusal, no credential in response.
  - **Forbidden-claim guard:** User input + assistant output scanned for `/\b(rank\s*#?1|first page|traffic lift|guaranteed results)\b/i`. Match → pre-canned refusal.
  - **Field extraction:** `src/lib/chat/field-extractor.ts` parses conversation → `Partial<IntakeFormFields>` returned as `fieldUpdates` (form ↔ chat sync).
- **System prompt:** `src/lib/chat/system-prompt.ts`
  - Hard rules:
    - Ask ONE question at a time.
    - NEVER ask for passwords, API keys, tokens, or credentials.
    - NEVER promise rankings, traffic, or guaranteed results.
    - Friendly, adaptive to client's language.
  - Role: SEO intake assistant for SnowIT SEO readiness service.
- **Cost discipline:**
  - **Ollama FORGE (10.0.0.2:11434):** PRIMARY path, zero cost, ~50 tok/s (qwen2.5:7b-instruct-q8_0).
  - **Groq:** Fallback #1, ~$0.10/1M input tokens.
  - **Anthropic Haiku:** Fallback #2, ~$0.80/1M input tokens.
  - Cloud only on local outage: `tier-router.ts` health-checks `/api/tags` before calling Ollama (avoid cloud cost on idle).
- **DEFERRED:** WS-B4 (pre-seed chat from live crawl) NOT implemented. Chat starts fresh without crawl data. Follow-on MC can add: "if website provided, call `runLiveCrawlAudit()` and seed assistant context with detected title/description/services."

Evidence: `/tmp/alai/seo-ws-b-agentforge-evidence/implementation-summary.md`
Validation: `validate-intake-chat.ts` 5/5 PASS, Ollama live turn JSON verified.

WS-C: Auto-Pipeline on Intake Submit

Deliverable: Intake submit triggers audit → findings → draft report (GBP fallback for no-website clients).

- **Pipeline trigger:** `src/lib/pipeline/run-intake-pipeline.ts`
  - Called on intake form submit (`src/app/intake/[token]/actions.ts`).
  - Runs: `runLiveCrawlAudit()` → `generateFindings()` → `generateReport()`.
  - **GBP/no-website mode:** If `intake.hasWebsite === false` OR `canonicalUrl` empty → audit runs `local_readiness_gbp` stage (12 GBP-specific checks: claim/manager/NAP/category/hours/photos/reviews/posts), skips website-dependent checks, still drafts report (labeled as readiness assessment, not ranking prediction).
  - **Security:** `rawToken` NOT passed into pipeline (no token in audit/findings/report records).
- **Stage derivation:** `src/lib/workspace/referral-stages.ts`
  - Client record status derived from data state (NOT manual flag):
    - `created` → client created, no token yet.
    - `link_sent` → `intakeTokenHash` present.
    - `link_opened` → future (requires link-click tracking, not implemented).
    - `intake_submitted` → `intakeSubmissions.length > 0`.
    - `audit_ready` → audit record exists.
    - `report_ready` → report draft exists.
  - Monotonic progression: stage never regresses.
- **Email notification:** `src/lib/email/notify.ts`
  - Sends "New intake submission for {clientName}" to Asmir (partner email).
  - **Security:** `secretGuard()` scans notification body for credentials/tokens before send. Token redacted as `[REDACTED]` if accidentally included.
  - `forbiddenClaimWords` enforced in audit runner + report generator + exporter (aligned with `src/lib/workspace/persistence.ts`).
- **Asmir dashboard:** `src/app/partners/referrals/page.tsx`
  - Shows all clients with derived stage (created → link_sent → intake_submitted → report_ready).
  - Real-time status derived from data, no manual updates needed.
- **Re-submit behavior:** New intake submission appends new version, pipeline records both runs (versioned history, not overwrite).

Evidence: `/tmp/alai/seo-ws-c-evidence/validate-referral-pipeline.txt`
Validation: `validate-referral-pipeline.ts` 32/32 PASS.

Cloudflare Access Carve-Out (Defense-in-Depth)

Problem: SEO Portal is protected by Cloudflare Access (trusted-header SSO). Public intake links (`/intake/[token]`) must be accessible to unauthenticated clients, but the rest of the app (`/partners`, `/api/health`) must remain gated.

Solution: Created a SEPARATE Cloudflare Access application with a BYPASS policy for `/intake/*` paths ONLY. The edge lets `/intake/*` through, and the app's own token gate (`verifyIntakeToken()`) protects the intake pages (defense-in-depth: CF bypass + app token gate).

CF Access App

- **App ID:** `06f98e51-1f07-4660-b93b-c426ccff21ce`
- **Name:** "SEO Portal - Public Intake Path"
- **Domains:**
  - `seo-tools.snowit.ba/intake`
  - `seo-tools.alai.no/intake`
  - `seo-tools.snowit.ba/api/intake-chat`
  - `seo-tools.alai.no/api/intake-chat`

Bypass Policy

- **Policy ID:** `d9e51759-c151-4ee7-8094-e5191e5f9163`
- **Name:** "Public Intake Bypass"
- **Decision:** `bypass`
- **Include:** `{ "everyone": {} }`
- **Precedence:** 1

Verification (Live Prod)

Path	Expected	Result
`/intake/bogus-token`	200 "Link not valid" (app served, no CF redirect)	□ PASS
`/api/intake-chat`	401 from app (origin reached, not CF 302)	□ PASS
`/partners`	302 to CF Access login (still gated)	□ PASS
`/api/health`	302 to CF Access login (not in bypass)	□ PASS

Key signal: Header `x-seo-portal-access: blocked` proved request hit the app, confirming CF Access carve-out working.

CRITICAL DEPLOY STEP: This CF Access carve-out is MANDATORY for the self-serve intake feature. Without it, all `/intake/[token]` requests are redirected to CF Access login instead of reaching the app's token-gated intake page. The first cutover attempt FAILED because this step

was missing. Deploy runbook now documents this as a hard gate.

Evidence: `/tmp/alai/seo-deploy-102929/FLOWFORGE-DEPLOY-REPORT.md` (Part 1: CF Access carve-out creation + edge verification).

Azure App Service Deploy

Deploy Target

- **Resource group:** `rg-seo-readiness-prod`
- **App Service:** `seo-readiness-alai`
- **Region:** Sweden Central
- **Registry:** `alairegistry.azurecr.io`
- **Image:** `alairegistry.azurecr.io/seo-readiness-portal:20260604-selfserve-intake`
- **Public URLs:**
  - `https://seo-tools.snowit.ba` (Cloudflare custom hostname)
  - `https://seo-tools.alai.no` (Cloudflare custom hostname)

Required Environment Variables

- `INTAKE_TOKEN_SECRET`: 64-char hex (HMAC signing key for intake tokens). Generated on deploy, stored in Azure App Settings (NOT in image).
- `INTAKE_BASE_URL`: `https://seo-tools.snowit.ba` (used in magic-link email).

Post-Deploy Verification

Check	Expected	Result
Image tag	`20260604-selfserve-intake`	PASS
App state	`Running`	PASS
`/intake/bogus` → 200	App served (no CF redirect)	PASS
`/partners` → 302	CF Access login	PASS

Evidence: `/tmp/alai/seo-deploy-102929/FLOWFORGE-DEPLOY-REPORT.md` (Part 2: deploy + ZAKON PI2 post-deploy verification).

Known Gaps

1. **SnowIT sender mailbox not provisioned:** Magic-link emails currently sent via `--account alai` sender (alai.no). Body is SnowIT-branded; sender is alai.no. Follow-on MC to provision `Asmir@snowit.ba` sender mailbox.
2. **WS-B4 chat pre-seed from live crawl:** Deferred. Chat starts fresh without crawl data. Follow-on MC can add: "if website provided, call `runLiveCrawlAudit()` and seed assistant context with detected title/description/services."
3. **Rate-limiter resets on container restart:** Rate limiter state is in-process Map; resets on Azure container restart. Acceptable for MVP; Postgres migration follow-on.
4. **CF token rotation:** MC #102790 paused. The global CF API key was used for this deploy (scoped token was invalid). Rotation task should be resumed.

Operator Runbook

Create Client + Send Magic Link

1. Visit `https://seo-tools.snowit.ba/partners/clients/new` (Asmir's Cloudflare Access account).
2. Enter client name + email.
3. Submit → magic-link email sent automatically.
4. Client receives: "SEO Readiness Assessment for {clientName}" with link to `https://seo-tools.snowit.ba/intake/{token}`.

Monitor Referral Status

1. Visit `https://seo-tools.snowit.ba/partners/referrals`.
2. Client status derived from data:
   - `link_sent` → email sent, awaiting client action.
   - `intake_submitted` → client submitted form/chat.
   - `report_ready` → audit + report draft generated.

Troubleshoot "Link not valid"

Possible causes:

1. **Forged token:** Token signature invalid (client manually edited URL). Expected behavior: 200 error page, no data leak.
2. **Expired token:** Token `expiresAt` past (default 30 days). Expected behavior: 200 "Link expired" error page.
3. **INTAKE_TOKEN_SECRET mismatch:** Azure env var changed after token was minted. Fix: Re-send magic link from `/partners/clients/{clientId}` (generates new token with current secret).

Debug command:

```
az webapp config appsettings list -g rg-seo-readiness-prod -n seo-readiness-alai --query
"[?name=='INTAKE_TOKEN_SECRET'].value" -o tsv
```

If secret changed → re-send magic link from portal.

Troubleshoot Rate Limit (429)

Symptom: Client sees "Too many requests" after 10 chat messages in 60 seconds.

Expected behavior: Rate limiter prevents abuse on public `/api/intake-chat` route.

Workaround: Wait 60 seconds for sliding window to reset.

Long-term fix: Postgres-backed rate limiter (survives container restarts). Follow-on MC.

Re-Deploy Self-Serve Intake

Pre-flight:

```
cd /Users/makinja/business/ALAI-Holding-AS/products/SEO-Readiness-Portal
npm run type-check && npm run build && npm run validate:spec
```

Build image:

```
az acr build -r alairegistry -t seo-readiness-portal:YYYYMMDD-tag-name .
```

Deploy:

```
az webapp config container set \
  --resource-group rg-seo-readiness-prod \
  --name seo-readiness-alai \
  --container-image-name alairegistry.azurecr.io/seo-readiness-portal:YYYYMMDD-tag-name \
  --container-registry-url https://alairegistry.azurecr.io
az webapp restart --resource-group rg-seo-readiness-prod --name seo-readiness-alai
```

Post-deploy verify (ZAKON PI2):

```
curl -sI https://seo-tools.snowit.ba/intake/bogus | grep -E "HTTP|content-type"
# Expect: HTTP/2 200, content-type: text/html; charset=utf-8
```

```
curl -sI https://seo-tools.snowit.ba/partners | grep -E "HTTP|location"
# Expect: HTTP/2 302, location: https://<CF-Access-login-url>
```

Rollback:

```
az webapp config container set \
  --resource-group rg-seo-readiness-prod \
  --name seo-readiness-alai \
  --container-image-name alairegistry.azurecr.io/seo-readiness-portal:20260602-real-audit \
  --container-registry-url https://alairegistry.azurecr.io
az webapp restart --resource-group rg-seo-readiness-prod --name seo-readiness-alai
```

Previous known-good: `20260602-real-audit`. Do NOT open origin IP-lock. Do NOT touch any Bilko domain.

Evidence Ledger

Artifact	Location	SHA-256
Validation report (Proveo PASS)	`/tmp/alai/seo-wsv-evidence/VALIDATION-REPORT.md`	`72e79a31...`
Playwright results (10/10 PASS)	`/tmp/alai/seo-wsv-evidence/playwright-results.json`	`6d35586e...`
WS-A evidence	`/tmp/alai/seo-ws-a-evidence/`	—
WS-B evidence	`/tmp/alai/seo-ws-b-agentforge-evidence/`	—
WS-C evidence	`/tmp/alai/seo-ws-c-evidence/`	—
Deploy report (FlowForge)	`/tmp/alai/seo-deploy-102929/FLOWFORGE-DEPLOY-REPORT.md`	—
Live probes	`/tmp/alai/seo-wsv-evidence/live-probes.txt`	`4e03b886...`
Intake page HTML (SSR)	`/tmp/alai/seo-wsv-evidence/intake-page-content.html`	`033ad52f...`
Screenshots (AC1-AC10)	`/tmp/alai/seo-wsv-evidence/screenshots/`	—

Status: LIVE in production as of 2026-06-04.
Next: Provision `Asmir@snowit.ba` sender mailbox (WS-A follow-on). Resume CF token rotation MC #102790.

Revision #1
Created 2026-06-04 09:47:03 UTC by John
Updated 2026-06-04 09:47:03 UTC by John