

# R1 – Fiches de Cadrage E2E (livrées) · v1.0 (R2)

Dossier unique regroupant les fiches E2E pour chaque fonctionnalité livrée en R1 (M1→M3). Format = **fiche-cadrage-e2e\_v1.0** condensée. Gate : **agp-e2e-ready**.

## FICHE 1 – Prompt Builder MVP (UI + Back)

Meta · Feature: Prompt Builder MVP · Owner: <...> · AGP: <...> · UX: <...> · Codex: <...> · QA/ARC: <...> · Date:

### 1) Parcours E2E

- Affichage : **/console/prompt-builder** (liste blocs, badge "versionné" si owner), états empty/loading/error.
- Action : ajout/édition/suppression bloc.
- Résultat : bloc persistant + (si owner) **version++** et snapshot.

### 2) RBAC

- viewer: lecture seule · operator: édition **sans** version · owner: édition **avec** version.

### 3) Contrats

| Ty<br>pe | Méthode        | URL                            | Requê<br>te    | 200            | 4xx      | Idempotenc<br>e | Sécurité        |
|----------|----------------|--------------------------------|----------------|----------------|----------|-----------------|-----------------|
| API      | GET/POST/PATCH | <b>/api/prompt-blocks/:id]</b> | <b>{block}</b> | <b>{block}</b> | version` | 400/401/403/404 | idempotency_key |

Trace : **trace\_id** propagé UI→API→logs.

### 4) Observabilité & Budgets

- KPIs suivis (si parcours crée du trafic) : `ttft_ms/error_rate_percent`.
- Budgets (p95) :  $TTFT \leq 2000$  ms ; erreurs < 1%.

## 5) Données & Mémoire

- Tables : `prompt_blocks`, `prompt_block_versions` (append-only).

## 6) Preuves QA attendues

- `screenshot_prompt_builder.png`, `logs_api_promptblocks.json`, `coverage_report.html`.

Gate agp-e2e-ready : [ ] PASS / [ ] FAIL · Remarque: <...>

---

# FICHE 2 – Observabilité API (KPIs + runs)

Meta · Feature: API Observabilité · ...

## 1) Parcours E2E

- Affichage : page Observabilité consomme API.
- Action : filtre Lot/Sprint.
- Résultat : KPIs + table runs maj.

## 2) RBAC

- viewer/operator/owner: lecture ; 401/403 si non auth / rôle insuffisant (admin-only si config).

## 3) Contrats

| Type | Méthode | URL | Requête | 200 | 4xx | Idempotence | Sécurité |
|------|---------|-----|---------|-----|-----|-------------|----------|
|------|---------|-----|---------|-----|-----|-------------|----------|

|      |     |                   |                  |  |              |     |     |
|------|-----|-------------------|------------------|--|--------------|-----|-----|
| AP I | GET | /api/metrics/kpis | ?lot&sprint      | {ttft_ms, rtt_ms, error_rate_percent}<br>(arrondi 1 déc) | 400/401 /403 | n/a | JWT |
| AP I | GET | /api/metrics/runs | ?lot&sprint&page | {items:[...], page_size=20}                              | 400/401 /403 | n/a | JWT |

Trace : `trace_id` corrélé aux runs.

#### 4) Budgets & Observabilité

- Budgets p95 : TTFT $\leq$ 2000 ; RTT voix $\leq$ 2500 ; erreurs $<$ 1%.

#### 5) Données & Mémoire

- `metrics_raw, metrics_runs`.

#### 6) Preuves QA

- `logs_api_metrics.json`, (perf) `metrics_runs.csv` + `kpis_summary.json`, `screenshot_observabilite.png`.

Gate : [ ] PASS / [ ] FAIL

## FICHE 3 – Observabilité UI (KPIs + table)

Meta · Feature: UI Observabilité · ...

### 1) Parcours E2E

- Affichage : `/console/observabilite` → 3 KPI cards (TTFT, RTT, % erreurs) + table 20/pg, filtres Lot/Sprint.
- Action : changer filtre, pagination.
- Résultat : valeurs réaffichées avec arrondi 1 décimale.

## 2) RBAC

- viewer/operator/owner: lecture (masquer actions non autorisées).

## 3) Contrats

| Typ<br>e | Métho<br>de | URL               | Requête              | 200                              | 4xx             | Idempoten<br>ce | Sécuri<br>té |
|----------|-------------|-------------------|----------------------|----------------------------------|-----------------|-----------------|--------------|
| API      | GET         | /api/metrics/kpis | ?lot&spri<br>nt      | valeurs<br>ms/%<br>arrondi<br>es | 400/401/<br>403 | n/a             | JWT          |
| API      | GET         | /api/metrics/runs | ?lot&spri<br>nt&page | 20<br>lignes<br>par<br>défaut    | 400/401/<br>403 | n/a             | JWT          |

## 4) Preuves QA

- `screenshot_observabilite.png`, `logs_api_metrics.json`, tests UI (unités, arrondi, pagination).

Gate : [ ] PASS / [ ] FAIL

---

# FICHE 4 – Webhooks Sécurité (GitHub)

Meta · Feature: Webhooks sécurité · ...

## 1) Parcours E2E

- Action : GitHub envoie un webhook → back vérifie HMAC + idempotence.
- Résultat : 200 (sync) / 202 (async) ou 400/403.

## 2) RBAC

- N/A côté UI ; JWT/HMAC côté API.

### 3) Contrats

| Type | Méthode | URL                 | Requête  | 200   | 4xx     | Idempotence | Sécurité          |
|------|---------|---------------------|--|-------|---------|-------------|-------------------|
| API  | POST    | /api/webhook/github | {headers : X-Hub-Signature, X-GitHub-Delivery} | {ack} | 400/403 | delivery_id | HMAC + horodatage |

### 4) Observabilité

- Logs JSON sécurité, métriques d'erreur webhook.

### 5) Preuves QA

- logs\_webhooks.json, screenshot\_webhooks.png (optionnel), tests HMAC/idempotence.

Gate : [ ] PASS / [ ] FAIL

---

## FICHE 5 — Cross-Observabilité (trace\_id)

Meta · Feature: Cross-Obs trace\_id · ...

### 1) Parcours E2E

- Affichage : action UI génère trace\_id → API → DB metrics\_raw.
- Résultat : corrélation prouvée (UI↔API↔DB) sur un run donné.

### 2) Contrats

| Type | Méthode | URL | Requête | 200 | 4xx | Idempotence | Sécurité |
|------|---------|-----|---------|-----|-----|-------------|----------|
|------|---------|-----|---------|-----|-----|-------------|----------|

|     |   |   |            |       |     |     |     |
|-----|---|---|------------|-------|-----|-----|-----|
| API | * | * | header     | { ... | 4xx | n/a | JWT |
|     |   |   | x-trace-id | }     |     |     |     |

### 3) Preuves QA

- `logs_traceid.json` (triplet UI/API/DB), capture UI montrant l'ID.

Gate : [ ] PASS / [ ] FAIL

---

## FICHE 6 – Documents (UX complet : Upload · Liste · Preview · Download)

Meta · Feature: Documents UX · ...

### 1) Parcours E2E

- **Affichage** : `/console/documents` (liste paginée 20/pg ; états empty/loading/error).
- **Actions** :
  1. **Upload** fichier → progress → item apparaît en tête de liste.
  2. **Preview** d'un document → modal/route `/console/documents/:id` (métadonnées, actions).
  3. **Download** → lien signé/secure.
- **Résultat** : fichier stocké + indexé ; aperçu lisible ; téléchargement sécurisé.

### 2) RBAC

- viewer: lecture/preview/download ; operator: + upload ; owner: + suppression/retag.

### 3) Contrats (mini-contrat)

| Ty<br>pe | Métho<br>de | URL | Requ<br>ête | 200 | 4xx | Idempote<br>nce | Sécur<br>ité |
|----------|-------------|-----|-------------|-----|-----|-----------------|--------------|
|----------|-------------|-----|-------------|-----|-----|-----------------|--------------|

|     |      |                        |        |                           |              |     |                 |
|-----|------|------------------------|--------|---------------------------|--------------|-----|-----------------|
| API | POST | /api/projects/:id/docs | {file} | {doc}                     | 400/401 /403 | n/a | JWT             |
| API | GET  | /api/projects/:id/docs | ?page  | {items, page_size=20}     | 400/401 /403 | n/a | JWT             |
| API | GET  | /api/docs/:id          | n/a    | {doc, preview_url?}       | 401/403 /404 | n/a | JWT             |
| API | GET  | /api/docs/:id/download | n/a    | 302 Location:<signed_url> | 401/403 /404 | n/a | JWT + URL signé |

Trace : `trace_id` corrélé (upload→index→liste→preview/download).

#### 4) Observabilité & Données

- Tables : `docs`, `docs_index` ; métriques : `upload_latency_ms`, `error_rate_percent`.

#### 5) Preuves QA

- Captures `screenshot_documents_list.png`, `screenshot_documents_preview.png` ; `logs_api_docs.json` ; test E2E upload→preview→download.

Gate : [ ] PASS / [ ] FAIL

---

## FICHE 7 – Health & RBAC de base

Meta · Feature: Health + RBAC · ...

### 1) Parcours E2E

- Affichage : badge santé ; routes protégées selon rôle.
- Action : `/api/health` ; accès `/console/*` selon rôle.
- Résultat : 200 `{status:"ok", ts}` ; 401/403 si non autorisé.

## 2) Contrats

| Type | Méthode | URL         | 200          | 4xx     | Sécurité              |
|------|---------|-------------|--------------|---------|-----------------------|
| API  | GET     | /api/health | {status, ts} | 4xx     | JWT                   |
| API  | *       | /console/*  | n/a          | 401/403 | JWT + middleware rôle |

## 3) Preuves QA

- `logs_api_health.json`, tests RBAC routes, screenshot badge rôle UI.

Gate : [ ] PASS / [ ] FAIL

---

## FICHE 0 — Auth / Login → Console (E2E)

Meta · Feature: Auth/Login → Console · Owner: <...> · AGP: <...> · UX: <...> · Codex: <...> · QA/ARC: <...> · Date:

### 1) Parcours E2E

- **Affichage** : `GET /login` → page de connexion (états: empty/loading/error), champs email+password, bouton SSO (si activé), message RGPD.
- **Action** : `submit` formulaire (ou clic SSO). Pour formulaire : `POST /api/auth/login` avec `{email,password}`.
- **Résultat** :
  - **Succès** : 200 + `Set-Cookie: jwt=<...>; HttpOnly; Secure; SameSite=Lax` → **redirect /console** ; l'UI affiche l'utilisateur et le rôle.
  - **Échec** : 401 (identifiants invalides) → message d'erreur UI.
  - **Interdit** : 403 (rôle insuffisant pour /console) → message dédié + lien support.

### 2) RBAC & navigation protégée



- Middleware protège `/console/*` : si **pas de JWT** → 401 + redirect `/login` ; si **rôle insuffisant** → 403 UI.
- Rôles applicables après login : `viewer operator owner` (affichage conditionnel des CTA).

### 3) Contrats (mini-contrat)

| Typ<br>e | Méthod<br>e | URL                           | Requête                        | 200   | 4xx     | Idempoten<br>ce | Sécurit<br>é    |
|----------|-------------|-------------------------------|--------------------------------|---|---------|-----------------|-----------------|
| API      | POST        | <code>/api/auth/login</code>  | <code>{email, password}</code> | <code>{user:{id,email}, roles:[...], exp} + Set-Cookie JWT</code> | 400/401 | n/a             | JWT, rate-limit |
| API      | GET         | <code>/api/auth/me</code>     | n/a                            | <code>{user, roles}</code>  | 401     | n/a             | JWT             |
| API      | POST        | <code>/api/auth/logout</code> | n/a                            | 204   | 401     | n/a             | JWT             |

**Trace** : `trace_id` généré à l’affichage de `/login` et propagé jusqu’à `/console`.

### 4) Observabilité & budgets

- **Logs** : `auth_event={page_displayed|request_sent|success|failed}` avec `trace_id`.
- **KPIs** : `auth_latency_ms_p95 ≤ 1000` ; `auth_error_rate_percent < 1`.

### 5) Données & mémoire

- **JWT** signé (clé ≥32) ; pas de session serveur par défaut ; table `users` / `roles`.
- **Sécurité cookies** : `HttpOnly, Secure, SameSite=Lax` ; rotation clé si fuite.

### 6) Preuves QA attendues

- `screenshot_login.png` (form + états) ;  
`screenshot_console_after_login.png` (atterrissage).
- `logs_api_auth.json` (POST /auth/login → redirect /console avec même `trace_id`).
- Tests E2E : **login OK & login KO** (401/403), redirect testé.

Gate agp-e2e-ready : [ ] PASS / [ ] FAIL · Remarque : <...>

---

## FICHE 8 — Console Navigation (Shell + Routes Guard)

Meta · Feature: Console Shell/Navigation · ...

### 1) Parcours E2E

- **Affichage** : `/console` (header + sidebar : **Chat, Documents, Prompt Builder, Observabilité** ; badge rôle).
- **Actions** : click nav → route change (`/console/chat`, `/console/documents`, `/console/prompt-builder`, `/console/observabilite`), deep-link support ; back/forward navigateur.
- **Résultat** : page cible rendue, breadcrumb/onglet actif, 404 → page NotFound, états empty/loading/error gérés.

### 2) RBAC & visibilité

- Rôles : viewer/operator/owner ; **masquer** liens non autorisés ; middleware protège `/console/*` (401→/login, 403 UI).

### 3) Contrats (mini-contrat)

| Type | Méthode | URL  | 200                        | 4xx         | Sécurité         |
|------|---------|--|----------------------------|-------------|------------------|
| API  | GET     | <code>/api/auth</code><br><code>/me</code> | <code>{user, roles}</code> | 401         | JWT              |
| API  | *       | <code>/console/</code><br><code>*</code>   | n/a                        | 401/403/404 | JWT + guard rôle |

**Trace** : événements `nav_event={route_entered, route_left}` avec `trace_id`.

#### 4) Budgets & Observabilité

- **Budgets** : TTI ≤ 2000 ms (premier rendu console) ; nav intra-console ≤ 1000 ms p95.
- **Logs** : `route_change` avec durée.

#### 5) Preuves QA

- Captures `screenshot_console_nav.png` (menu), `screenshot_route_chat.png`, etc. ; `logs_route.json` ; test E2E garde 401/403/404.

**Gate** : [ ] PASS / [ ] FAIL

---

## FICHE 9 — Chat (UI + Stream/Fallback)

**Meta** · Feature: Chat · ...

### 1) Parcours E2E

- **Affichage** : `/console/chat` (historique thread, input, bouton **Envoyer**, bouton **Stop** ; états empty/loading/error).
- **Action** : saisie message → **Send** ;
  - **Chemin A (stream)** : UI ouvre `GET /api/chat/stream?thread_id=...` (SSE) ; tokens défilent.
  - **Chemin B (fallback)** : `POST /api/chat` (sync) ; réponse complète.
- **Résultat** : message assistant rendu ; compteur tokens facultatif ; erreurs visibles (timeout/retry/backoff).

### 2) RBAC

- viewer: lecture historique ; operator: + envoi message ; owner: + paramètres modèle.

### 3) Contrats (mini-contrat)

| Ty<br>pe | Méth<br>ode | URL                       | Requête                 | 200                       | 4xx                 | Idempoten<br>ce     | Sécu<br>rité |
|----------|-------------|---------------------------|-------------------------|---------------------------|---------------------|---------------------|--------------|
| SS<br>E  | GET         | /api/chat/<br>stream      | ?thread_id<br>&message= | event<br>:data<br>(delta) | 4xx/5xx             | n/a                 | JWT          |
| AP<br>I  | POST        | /api/chat                 | {thread_id<br>,message} | {repl<br>y}               | 400/401/4<br>03/429 | idempote<br>ncy_key | JWT          |
| AP<br>I  | POST        | /api/threads/<br>:id/(pin | unpin                   | abort)`                   | n/a                 | 200/409/4<br>04     | 4xx          |

Trace : `trace_id` + `thread_id` sur input, stream et stockage message.

### 4) Observabilité & Budgets

- KPIs : `chat_ttft_ms_p95` ≤ 2000, `error_rate_percent` < 1 ; log `stream_retry_count`.

### 5) Données & Mémoire

- Tables : `threads`, `messages` ; conservation **append-only** ; export JSONL optionnel.

### 6) Preuves QA

- `screenshot_chat_stream.png` / `screenshot_chat_fallback.png` ; `logs_api_chat.json` ; test E2E stream + fallback.

Gate : [ ] PASS / [ ] FAIL

---

## ANNEXE – Déclencheur d'audit (coller tel quel dans ticket QA)

QA-ARC-Review:

lot: R1-M3

scope: "Audit E2E fonctionnalités livrées (Prompt Builder, Observabilité API/UI, Webhooks, Cross-Obs, Documents, Health/RBAC)"

evidence\_expected:

- screenshot\_prompt\_builder.png
- logs\_api\_promptblocks.json
- screenshot\_observabilite.png
- logs\_api\_metrics.json
- metrics\_runs.csv
- kpis\_summary.json
- logs\_webhooks.json
- logs\_traceid.json
- logs\_api\_docs.json

decision\_gate: agp-e2e-ready

outputs:

- qa\_report.md
- agp\_review.yaml