1. Objective

Provide a detailed engineering plan to extend Crayto's existing browser extension into a full Zero-Trust password-protection agent with licence management, corporate-domain enforcement, and admin-configured response policies.

2. Scope & Assumptions

- **Local policy store:** All policies, whitelist URLs, and corporate domain lists are persisted in `chrome.storage.local` and referenced for every runtime check. The extension then syncs with the Admin API **continuously**—either on a fixed polling interval (default 5 minutes) or via a push/web-socket channel—so any admin policy change is reflected on endpoints within seconds.
- **Browsers:** Chrome (MV3) is baseline; Edge, Firefox, Safari and Brave
- **Backend:** tbd
- **Security Posture:** All extension→backend calls use mTLS with short-lived JWTs; no password data leaves the client.
- Existing features—DOM scan for `<input type="password">` and badge injection—are stable and will be refactored but not re-written.

3. Functional Requirements

ID	Requirement	Priority
FR-1	On first run, extension registers device + tenant via backend `/register` API and stores signed licence token.	P0
FR-2	For every page load & SPA route change, scan DOM for password forms.	P0
FR-3	Inject "Protected by Crayto" badge adjacent to each detected form.	P0
FR-4	Evaluate active tab `origin` against **Global Whitelist** pulled from backend & cached	P0

FR-5	If origin whitelisted, verify `form.action` host === `origin` and perform client-side script analysis to flag inline/linked JS capable of reading password fields.	P0
FR-6	If origin not whitelisted, evaluate email domain on `input[type="email"]` against **Corporate Domain List** (from local storage).	P0
FR-7	For corporate domains + non-whitelisted origins, display policy-driven modal: **Block**, **Proceed w/ Caution**, or **Request Approval**.	P0
FR-8	Modal choice & metadata logged to backend `/events` API.	P1
FR-9	Admin portal allows policy selection per tenant and manages whitelist & domain list.	P1
FR-10	Extension auto-updates policies every 10 min or via `websocket push`.	P2

4. Non-Functional Requirements

- **Performance:** DOM scan < 2 ms per page; memory overhead < 10 MB.
- **Privacy:** Never transmit user passwords or full page HTML.
- **Resilience:** Operates offline using last-known policies; fails closed for corporate domains.
- **Telemetry:** Only event hashes + timestamps; no PII.

5. Workflow (Happy Path)

1. **Install → Register**: Extension posts `{device_fingerprint, tenant_id}` → receives `licence_jwt`, `policies`.
2. **Page Load**: `mutationObserver` + initial scan locate password forms.
3. **Badge Inject**: Adds SVG badge near password/submit button.
4. **Whitelist Check**: Compare `location.origin` with cached whitelist.
5. **Security Checks**: Validate `form.action`, run static heuristics on inline JS (`window.addEventListener('input', …)`,

`document.querySelector('[type=password]')`, etc.).

6. **Decision Engine**: Combines origin status + email-domain evaluation + JS risk score.
7. **User Prompt**: If needed, render modal; record action.
8. **Submit Gate**: Depending on modal choice & policy, allow, warn, or block `submit` event.

6. Detailed Task Breakdown

Epic	Task	Owner	Effort	Dependency
E1 Extension Bootstrap	Refactor registration flow, add JWT storage (chrome.storage.local)	FE	2d	BE API
	Implement licence validation on startup	FE	1d	BE
E2 Policy Sync	Whitelist + Corp domains fetch & cache	FE	2d	BE
	Websocket push listener	FE	1d	BE
E3 DOM Scanner 2.0	Migrate to `mutationObserver`; debounce scans	FE	2d	—
E4 Script Analyzer	Static regex heuristics (e.g., `PasswordHarvestPattern`)	FE	3d	—
E5 Decision Engine	Implement risk matrix & outcome mapping	FE	2d	Policies
E6 UI Components	Badge SVG + positioning	FE	1d	—
	Modal UX (Block / Warn / Approve)	FE	2d	Design
E7 Telemetry	`/events` schema & batching	FE	2d	BE
E8 Backend Enhancements	Licence /policy APIs, Websocket hub	BE	4d	—
	Admin portal pages (whitelist, policies)	FE	5d	BE

E9 QA & Security	Unit tests (Jest) for scanner & engine	QA	3d	—
	Pen-test extension (CSP bypass, XSS)	Sec	3d	FE
E10 Packaging & Release	CI build (WebExt + zip), publish to Chrome Web Store (internal)	DevOps	2d	QA pass

Estimated **Sprint Length:** 2 weeks → MVP in 4 weeks.

8. Milestones

Date	Milestone
+2 w	E1–E3 complete; extension registers & scans with badge
+4 w	E4–E7; risk engine operational, telemetry live
+6 w	E8–E10; admin portal alpha, Chrome store release

9. Acceptance Criteria (MVP)

- Install & registration succeed under 500 ms.
- Non-whitelisted + corporate email triggers modal ≥ 98 % of cases in test suite.
- No password field submission blocked on whitelisted origin with valid form/action/scripts in 1 000 test URLs.
- Extension passes Google MV3 compliance checks & content-security policy enforced.

10. Risk & Mitigations

- **False Positives** → Provide quick override + feedback loop to backend.
- **Performance Regression** → Benchmark on top-100 Alexa sites each release.
- **Policy Drift** → Signed policies with expiry; extension denies outdated policies > 24 h.

11. Future Enhancements

- Heuristic & ML-based script classification.
- Phishing-page screenshot upload (privacy-preserving hash) for SOC triage.
- Firefox & Safari (MV3 polyfill) support.
- SSO-based admin portal.

Prepared by: Engineering @ Crayto **Version:** v0.1 (2025-04-27)