JWT Security Reference — browser-context implementatiekaart

Transport · Storage · Resource types · Cross-domain / CDN · Geavanceerd · Server-side validatie · Security matrix

Referentiekaart voor browser-gebaseerde webapplicaties — context-afhankelijk. Buiten scope: OAuth/OIDC flows, PKCE, server-side identity, native/mobile apps. Geen vervanging voor threat modelling (STRIDE), key management en CSP-hardening.

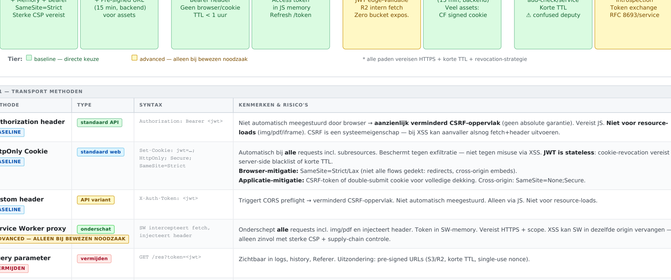
aanbevolen · web-standaard · krachtig patroon · conditioneel · geavanceerd · vermijden · beperkt

DEFAULT ARCHITECTURE — 90% VAN DE USE CASES (WEB SAAS)

Refresh token → HttpOnly; Secure; SameSite=Strict; __Host- cookie | **Access token** → JS memory, 15 min TTL | **API** → Authorization: Bearer | **Media/bestanden** → pre-signed URL (15 min, server-side) | **Rotatie** → family-invalidation bij hergebruik | **Revocation** → jti-blacklist voor kritieke operaties

SW proxy / DPoP / mTLS / localStorage+CSP → alleen toevoegen bij aantoonbare behoefte. Baseline first.

BESLISBOOM — KIES JE ARCHITECTUUR OP USE CASE TYPE



Tier: ☐ baseline — directe keuze ☐ advanced — alleen bij bewezen noodzaak

* alle paden vereisen HTTPS + korte TTL + revocation-strategie

01 — TRANSPORT METHODEN

METHODE	TYPE	SYNTAX	KENMERKEN & RISICO'S
Authorization header BASELINE	standaard API	`Authorization: Bearer <jwt>`	Niet automatisch meegestuurd door browser → **aanzienlijk verminderd CSRF-oppervlak** (geen absolute garantie). Vereist JS. **Niet voor resource-loads** (img/pdf/iframe). CSRF is een systeemeigenschap — bij XSS kan aanvaller alsnog fetch+header uitvoeren.
HttpOnly Cookie BASELINE	standaard web	`Set-Cookie: jwt=…; HttpOnly; Secure; SameSite=Strict`	Automatisch bij **alle** requests incl. subresources. Beschermt tegen exfiltratie — niet tegen misuse via XSS. **JWT is stateless**: cookie-revocation vereist server-side blacklist of korte TTL. **Browser-mitigatie:** SameSite=Strict/Lax (niet alle flows gedekt: redirects, cross-origin embeds). **Applicatie-mitigatie:** CSRF-token of double-submit cookie voor volledige dekking. Cross-origin: SameSite=None;Secure.
Custom header BASELINE	API variant	`X-Auth-Token: <jwt>`	Triggert CORS preflight → verminderd CSRF-oppervlak. Niet automatisch meegestuurd. Alleen via JS. Niet voor resource-loads.
Service Worker proxy ADVANCED — ALLEEN BIJ BEWEZEN NOODZAAK	onderschat	`SW intercepteert fetch, injecteert header`	Onderschept **alle** requests incl. img/pdf en injecteert header. Token in SW-memory. Vereist HTTPS + scope. XSS kan SW in dezelfde origin vervangen — alleen zinvol met sterke CSP + supply-chain controle.
Query parameter VERMIJDEN	vermijden	`GET /res?token=<jwt>`	Zichtbaar in logs, history, Referer. Uitzondering: pre-signed URLs (S3/R2, korte TTL, single-use nonce).
Request body BASELINE	beperkt	`{ "token": "<jwt>" }`	Alleen POST/PUT. JSON content-type triggert preflight → verminderd CSRF-oppervlak. Niet voor GET of resource-loads.
WebSocket (OTT) ADVANCED	specifiek	`ws://…/?ott=<token>` of eerste WS-frame	Browser ondersteunt geen custom headers bij upgrade. Wissel JWT in voor one-time token (30s TTL, single-use). OTT in query param is hier geaccepteerd patroon — het is geen long-lived credential.
EventSource / SSE BASELINE	beperkt	`Cookie of query param`	EventSource ondersteunt geen headers. Alternatief: Fetch + ReadableStream (volledige header-support).
DPoP (RFC 9449) ENTERPRISE	sender-constrained	`DPoP: <proof-jwt>` `Authorization: DPoP …`	Bindt token aan client keypair. Waardeloos bij diefstal **alleen als** private key niet uitlekt. Proof bevat htm/htu/ath. Concepteuel sterk, praktisch fragiel ecosystem. Vereist correcte implementatie op client + server.

02 — CLIENT-SIDE STORAGE

STORAGE TYPE	TYPE	SCOPE	XSS	EIGENSCHAPPEN
Memory (JS var) BASELINE	minimale exposure	tab	runtime risico	Verdwijnt bij refresh, nooit naar disk. **Niet XSS-safe**- XSS geeft directe runtime-toegang. **Vs localStorage:** memory → runtime misuse (sessie-gebonden). localStorage → exfiltratie + replay buiten sessie (persistente blast radius). Kortste exposure window, niet nul risico.
SW memory ADVANCED	geïsoleerd	origin	beperkt	Niet direct bereikbaar via page-JS. Maar: postMessage misuse, lifecycle attacks en compromised origin kunnen SW beïnvloeden. "Resistent" is te sterk — **geïsoleerd, niet immuun**.
SharedWorker memory ADVANCED	multi-tab	origin	resistent	Eén instantie over alle tabs. Cross-tab token-sync zonder localStorage-risico.
HttpOnly Cookie BASELINE	web-standaard	origin/domain	resistent	Beschermt tegen exfiltratie via JS. **Niet** tegen misuse: XSS kan requests doen met de cookie. Domain-attribuut voor subdomain-sharing. __Host- prefix = sterkste isolatie.
sessionStorage VERMIJDEN — GEBRUIK MEMORY	marginaal	tab	kwetsbaar	Tab-scoped, niet gedeeld. Verdwijnt bij sluiting. Nog steeds XSS-kwetsbaar — gebruik memory.
localStorage VERMIJDEN TENZIJ CSP + SANDBOXING	risicovol	origin	kwetsbaar	Persistent, cross-tab. Volledig JS-leesbaar → exfiltratie + **replay buiten sessie** mogelijk na compromise. Grotere blast radius dan memory. Risico te reduceren met strikte CSP + sandboxing — context-afhankelijk, niet absoluut verboden.
IndexedDB VERMIJDEN TENZIJ WEB CRYPTO	risicovol	origin	kwetsbaar	Async, persistent. Zelfde XSS-risico als localStorage. Alleen zinvol met Web Crypto.
IDB + Web Crypto ENTERPRISE	geavanceerd	origin	gedeeltelijk	extractable:false key. Beschermt exfil maar niet in-context misuse in dezelfde origin.
Cache API VERMIJDEN	vermijden	origin	kwetsbaar	Toegankelijk vanuit SW én page. Niet bedoeld voor credentials.

03 — RESOURCE TYPES & AUTH-STRATEGIE

RESOURCE TYPE	TYPE	HEADERS?	AUTH-STRATEGIE
XHR / Fetch BASELINE	volledig	ja	Volledige controle. Authorization: Bearer standaard. CORS preflight bij cross-origin.
img / video / audio BASELINE	beperkt	nee	**Contextafhankelijk:** (1) Pre-signed URL — primair in productie (native browser support, CDN-compatible). (2) Fetch → Blob URL — geen URL-lekkage, breekt caching + streaming. (3) HttpOnly cookie. (4) SW proxy (niche). Nooit **long-lived JWT** in URL.
PDF / download BASELINE	proxy / signed	nee	Inline: Fetch → ArrayBuffer → Blob URL. Download: Fetch → Blob → a.download. Direct: pre-signed URL (5–15 min, single-use, nonce). Nooit **long-lived JWT** in URL — short-lived signed URLs zijn wel toegestaan.
SW (interceptor) ADVANCED	universeel	ja — injecteert	Onderschept **alle** browser requests en voegt header toe. Token in SW-memory. Krachtig in media-heavy / offline-first apps; **geen default best practice** — XSS kan SW vervangen, cache poisoning is reëel risico.
iframe ADVANCED	complex	nee	Cookie: SameSite=None;Secure (cross-origin). postMessage voor token-doorgeven. Geen wildcard CORS combineren.
WebSocket ADVANCED	specifiek	alleen upgrade	Cookie bij upgrade. OTT als query param (invalideert direct). Of auth-message als eerste WS-frame.
EventSource / SSE BASELINE	beperkt	nee	Cookie of OTT query param. Vervang door Fetch+ReadableStream voor volledige header-support.
Web Worker fetch ADVANCED	geïsoleerd	ja	Worker roept fetch aan met headers. Token via postMessage van main thread. Token nooit in page-scope tijdens request.

04 — CROSS-DOMAIN & CDN (S3 · R2 · CLOUDFRONT · CLOUDFLARE)

SCENARIO	TYPE	MECHANISME	IMPLEMENTATIE & RISICO'S
Pre-signed URL BASELINE S3, R2, GCS, Azure	standaard CDN	`?X-Amz-Signature=…` `&X-Amz-Expires=900`	HMAC-gesigneerde URL met TTL. Aanbevolen 15 min. Geen JWT op client. **Risico URL-lekkage:** log sanitization vereist, `Referrer-Policy: no-referrer` instellen, CDN access logs reviewen op credential exposure. Single-use nonce voor eenmalighеid.
CF signed cookie BASELINE	CDN-brede auth	`CloudFront-Policy=…` `CloudFront-Signature=…`	Eén set cookies → wildcard-path toegang. Automatisch meegestuurd. Ideaal bij veel bestanden per sessie. Cross-origin: SameSite=None;Secure.
CF signed URL ADVANCED	per-resource	`?Expires=…&Signature=…` `&Key-Pair-Id=…`	Per-URL auth, deelbaar. S3-bucket nooit direct exposed (CloudFront als proxy). TTL server-side afdwingen.
Cloudflare R2 + Worker ADVANCED	R2-specifiek	`Worker valideert JWT → fetch R2 intern → stream naar client`	**CF Worker als auth-proxy** validеert JWT via Web Crypto op edge, zero-exposure van bucket credentials. Minimale latency.
Akamai / Fastly ENTERPRISE	enterprise CDN	`HMAC edge-token in URL of cookie`	Eigen token formats (niet JWT). Akamai EdgeAuth / Fastly token auth. TTL + IP-binding + path-scoping.
CORS + credentials BASELINE	cross-origin API	`Allow-Origin: exact` `Allow-Credentials: true`	Exacte origin vereist (geen *). credentials:'include' + SameSite=None;Secure. Authorization in CORS allowlist. Nooit met wildcard.
Subdomain cookie ADVANCED	multi-subdomain	`Domain=.example.com; Secure; HttpOnly`	Gedeeld over api./app./cdn. Risico: subdomain-XSS compromitteert allen. __Host- prefix voorkomt dit maar is incompatibel met Domain-attribuut.
Token exchange proxy ADVANCED	zero-trust	`Client → backend → RFC 8693 token → S3/R2/ext. API`	Client heeft **nooit** service-credentials. Backend valideert JWT, tekent requests (SigV4) of wisselt in voor scoped token.

05 — GEAVANCEERDE TECHNIEKEN

TECHNIEK	TYPE	CATEGORIE	WANNEER & WAAROM
DPoP (RFC 9449) ENTERPRISE	sender-constrained	transport	Proof-JWT met htm/htu/ath. Gestolen token waardeloos zonder private key. Mitigeert replay + exfiltratie. Vereist server-side validatie.
mTLS token binding ENTERPRISE	cert-gebonden	transport	Token aan TLS client-cert via cnf claim. Backend valideert thumbprint. Sterkste binding. Complexe PKI. Gebruikt in PSD2 / open banking.
Token rotation BASELINE	standaard	lifecycle	Refresh token eenmalig bruikbaar. Hergebruik → family-invalidatie. Families bijhouden in database voor diefstal-detectie.
Silent refresh BASELINE	moderne aanpak	lifecycle	Background fetch naar /auth/refresh. Cookie autoriseert → access token in response body → memory. Vervangt verouderd iframe silent refresh.
__Host- prefix BASELINE	hardening	cookie	Vereist: Secure, geen Domain, Path=/. Voorkomt subdomain-overschrijving. Sterkste isolatie. Incompatibel met Domain-attribuut.
__Secure- prefix BASELINE	tussenoptie	cookie	Vereist Secure flag. Domain-attribuut toegestaan. Voorkomt HTTP-injectie. Minder strikt dan __Host-.
CHIPS (partitioning) ADVANCED	2024+	cookie	Partitioned cookies scoped aan top-level + third-party origin. Reactie op afschaffing third-party cookies. Voor cross-site embedded widgets.
PAR (RFC 9126) ENTERPRISE	OAuth	OAuth flow	Auth request parameters server-side gepusht vóór redirect. Voorkomt parameter-tampering in URL. Bij complexe scopes of hoog-risico flows.
Introspection proxy ENTERPRISE	microservices	architectuur	API gateway valideert JWT centraal, stuurt claims door naar downstream. Vereenvoudigt sleutelrotatie + revocation. Let op: **confused deputy** — downstream service moet audience (aud) valideren en niet blind token forwarden. Bij microservices: overweeg service-to-service tokens (SPIFFE/SVID) naast user tokens.

07 — SERVER-SIDE JWT VALIDATIE (VEREISTE CHECKS)

CLAIM / ELEMENT	PRIORITEIT	VEREISTE VALIDATIE
alg BASELINE	kritiek	**Whitelist: RS256, ES256, PS256.** Weiger: none, HS* bij asymmetrische setup. Server bepaalt toegestane algoritmen — vertrouw nooit de alg-claim uit het token zelf.
iss BASELINE	vereist	Exact match met verwachte issuer. Weiger tokens zonder iss of met onbekende issuer. Voorkomt token-reuse vanuit andere systemen.
aud BASELINE	vereist	Exact match met service-identifier. **Kritiek bij microservices**: voorkomt confused deputy bij token forwarding. Elke service valideert eigen aud.
exp / nbf BASELINE	vereist	Strikte expiry-check. Klok-skew max 30–60 sec. nbf valideren indien aanwezig. **Weiger tokens zonder exp.**
kid + JWKS BASELINE	rotatie	kid → JWKS-lookup per key. Cache JWKS 5–15 min. Bij onbekende kid: refresh cache vóór afwijzen (rotatie-scenario). Nooit kid uit token vertrouwen zonder verificatie.
jti (replay) ADVANCED	optioneel	jti-registry voor refresh tokens en kritieke operaties. **JWT zonder jti-check is niet revoceerbaar binnen TTL.** Schaal: distributed cache (Redis) bij hoge load.
scope / claims BASELINE	autorisatie	Valideer per endpoint. RBAC/ABAC buiten JWT-validatie. Claims zijn assertions van issuer — verifieer vertrouwen van issuer alvorens te gebruiken.

RISICO-GEWICHTXSS = account takeoverCSRF = actie zonder leesrechtReplay = hergebruik geldig tokenXSS > CSRF > Replay — *header strategie verplaatst risico naar XSS; cookie + SameSite=Strict is robuuster dan het lijkt*

06 — SECURITY MATRIX

TRANSPORT × STORAGE

heuristisch — niet deterministisch

STORAGE → TRANSPORT ↓	MEMORY	HTTPONLY COOKIE	LOCALSTORAGE	SESSION STORAGE	INDEXEDDB	SW MEMORY
Authorization header	✓ ideaal	n.v.t.*	XSS	XSS	XSS	✓
HttpOnly Cookie	—	✓ ideaal	—	—	—	—
Custom header	✓	—	XSS	XSS	XSS	✓
Service Worker proxy	—	—	—	—	—	✓ native
Query param (OTT)	✓ OTT	✓	✗	✗	✗	✓ OTT

* Cookie transport leest token niet via JS. `✓/✗` zijn indicatief — uitkomst afhankelijk van threat model, attacker capabilities en mitigaties (CSP, sandboxing, origin isolation). **Revocation niet meegenomen**: elke ✓ geldt alleen tot token-expiry tenzij server-side intrekking is geïmplementeerd. Gebruik als startpunt, niet als beslisser. Patroon: cookie valideert op /auth/token → access token in response body → opslaan in memory → Authorization: Bearer.

TRANSPORT / METHODE × RESOURCE TYPE

heuristisch — niet deterministisch

METHODE ↓ / RESOURCE →	XHR/ FETCH	IMG/ VIDEO	PDF/ DOWNLOAD	IFRAME	WEB-SOCKET	SSE	S3/R2 DIRECT
Authorization header	✓	✗	✗	✗	✗	✗	✗
HttpOnly Cookie	✓	✓	✓	SameSite!	✓	✓	✗
Service Worker proxy	✓	✓	✓	scope!	—	—	via proxy
Pre-signed URL (S3/R2)	overhead	✓	✓	✓	✓	beperkt	✓
CF signed cookie	✓	✓	✓	✓	✓	✓	CDN ✓
Fetch → Blob URL	—	✓ best	✓ best	✓	—	—	via fetch
CF Worker proxy (R2)	✓	✓	✓	✓	—	—	✓

Productie-aanbeveling (defence-in-depth): Refresh token in **HttpOnly; Secure; SameSite=Strict; __Host- cookie** → Access token in **JS memory (15 min TTL)** — minimale exposure window, niet XSS-safe → API calls via **Authorization: Bearer** → Bevestigde media via **pre-signed URL (primair)** of Fetch+Blob (geen URL-lekkage) → R2 via **CF Worker proxy** bij complexe auth-logica → Token rotation met **family-invalidation** bij gedetecteerd hergebruik → **Revocation:** JWT is stateless — combineer korte TTL (access ≤15 min) met server-side blacklist of jti-registry voor kritieke operaties → **Replay-detectie:** jti-claim + server-side registry voor refresh tokens → **Device binding** (optioneel, hoog risico): DPoP of mTLS voor sender-constrained tokens

Ontbrekende context voor volledige architectuurbeslissing: Threat model (STRIDE / attacker capabilities / runtime threat behaviour) · Key management (JWKS, kid-rotatie, alg-whitelist, key escrow) · Revocation (stateless JWT + blacklist/jti-registry + TTL-strategie per token type) · Backend trust: introspection vs local validation · confused deputy · token forwarding risico · SPIFFE/SVID · End-to-end trust chain redenering · device binding · replay-detectie · rotation enforcement model