**Co daje reverse proxy (Matomo + Elastic RUM) – pod RODO / ryzyka**

1. **Pierwsza strona zamiast „third-party”**  
   • endpointy typu https://twoja-domena/analytics/... i https://twoja-domena/rum/... → brak zewnętrznych domen w przeglądarce.  
   • Łatwiejsze spełnienie zasady **minimalizacji ujawnień** i ograniczenie CORS.
2. **Gating po zgodzie** (CMP → cookie/flaga)  
   • RP może **blokować/odrzucać** żądania do intake, jeśli brak consent=analytics=1.  
   • Minimalizacja: zamiast 302/403 możesz zwracać **204 No Content** (ciche „dropnięcie” eventu).
3. **Anonimizacja na brzegu**  
   • **IP**: nie przekazujesz IP klienta (czyścisz X-Forwarded-For/X-Real-IP) → backend widzi IP proxy.  
   • **Query/URL**: automatyczne **obcinanie parametrów** (np. email=, phone=, utm\_\*) i skracanie ścieżek (np. /order/123456 → /order/:id).  
   • **Nagłówki**: allowlista—przepuszczasz tylko to, co potrzebne (np. Content-Type). Opcjonalnie „spłaszczasz” User-Agent.
4. **Twarde polityki nagłówków po stronie serwowania skryptów**  
   • Content-Security-Policy, Referrer-Policy: strict-origin-when-cross-origin, Permissions-Policy, Cross-Origin-Resource-Policy: same-site.  
   • Serwujesz **apm-rum.js**/**matomo.js** z własnego hosta (SRI i cache), co ogranicza wycieki meta-danych.
5. **Kontrola cookies i brak fingerprintingu**  
   • RP może usuwać Set-Cookie z odpowiedzi intake (Matomo/APM nie dostaną szansy ustawić nowych ciastek na tym torze).  
   • Wymuszasz flagi (Secure, SameSite=Lax) na własnych cookies CMP.
6. **Filtry/limity i retencja logów proxy**  
   • Rate-limit/DoS (ochrona i ograniczenie bezużytecznego zbierania).  
   • **Logi access** bez IP/bez query (lub z hashem/saltem rotowanym) → spójność z zasadą **privacy by default**.
7. **Scentralizowana zgodność**  
   • Jedno miejsce na: DPIA, listę czynności przetwarzania, retencję, inspekcję danych wysyłanych do Matomo/APM.  
   • Ułatwione **geofencing/data residency** (serwer w UE/PL), TLS/mTLS, WAF.

**Architektura – szkic**

**A. Matomo (self-host lub on-prem)**

Przeglądarka

├── GET /analytics/matomo.js ──► ReverseProxy (Nginx/IIS/Traefik)

└── POST /analytics/collect ──► [Consent Gate] ──► Matomo (matomo.php)

└─► (opc.) Ingest pipeline: drop/anon fields

* Serwujesz matomo.js lokalnie z RP; hit’y idą na /analytics/collect (RP proxy\_pass → matomo.php).
* RP: ciche dropnięcie gdy brak zgody, czyszczenie IP i parametrów URL.

**B. Elastic RUM (APM Server)**

Przeglądarka

├── GET /rum/agent.js ──► ReverseProxy (serwowanie skryptu)

├── POST /rum/intake ──► [Consent Gate] ──► APM Server /intake/v2/rum/events

└── GET /rum/config ──► APM Server /config/v1/rum/agents (opcjonalnie)

└─► (opc.) Ingest pipeline w Elasticsearch: drop source.ip, url.query,...

* W RUM ustawiasz endpoint na https://twoja-domena/rum/intake; RP kieruje do APM Servera.
* Ingest pipeline po stronie ES może dodatkowo wycinać pola (np. source.ip, url.query, user.\*).

**Minimalne przykłady konfiguracji**

**Nginx – wspólne motywy (Matomo/RUM)**

# 1) Flaga zgody z CMP (np. consent\_analytics=1)

map $http\_cookie $analytics\_ok {

default 0;

"~\*consent\_analytics=1" 1;

}

# 2) Nagłówki ochronne dla skryptów

location = /analytics/matomo.js {

root /var/www/static; # lub proxy do matomo hosta

add\_header Content-Security-Policy "default-src 'self'; connect-src 'self';" always;

add\_header Referrer-Policy "strict-origin-when-cross-origin" always;

}

location = /rum/agent.js {

root /var/www/static;

add\_header Content-Security-Policy "default-src 'self'; connect-src 'self';" always;

add\_header Referrer-Policy "strict-origin-when-cross-origin" always;

}

# 3) Intake – gating po zgodzie + anonimizacja IP + obcięcie cookies

location /analytics/collect {

if ($analytics\_ok = 0) { return 204; }

proxy\_pass http://matomo.internal/matomo.php;

proxy\_set\_header X-Forwarded-For "";

proxy\_set\_header X-Real-IP "";

proxy\_hide\_header Set-Cookie;

proxy\_pass\_request\_headers off; # ewentualnie włącz i zallowlistuj niezbędne

}

location /rum/intake {

if ($analytics\_ok = 0) { return 204; }

proxy\_pass http://apm.internal:8200/intake/v2/rum/events;

proxy\_set\_header X-Forwarded-For "";

proxy\_set\_header X-Real-IP "";

proxy\_hide\_header Set-Cookie;

# (opc.) limit\_req / limit\_except / body\_size itp.

}

**Obcinanie parametrów PII w URL (przykład)**

# Przykład: /analytics/collect?...&email=...&phone=...&utm\_source=...

# W praktyce zrób to przy pomocy subrequestu/lua/map, albo przyjmuj POST body bez referrer

# Alternatywa: wyłącz wysyłkę pełnych URL w samym kliencie i/lub użyj pola "page\_name"

Najskuteczniejsze jest **nie wysyłać** PII z frontu (konfiguracja agenta) oraz **nie logować** query w access logach RP.

**IIS/ARR (Windows) – skrót idei**

* **Inbound Rule:**  
  /analytics/collect → http://matomo.internal/matomo.php  
  /rum/intake → http://apm.internal:8200/intake/v2/rum/events
* **Server Variables:** usuń/wyczyść HTTP\_X\_FORWARDED\_FOR, REMOTE\_ADDR.
* **Outbound Rule:** usuń Set-Cookie z odpowiedzi intake.
* **Condition (zgoda):** jeśli brak consent\_analytics=1 w cookie → Response Action = 204.

**Ustawienia po stronie narzędzi (warto sparować z RP)**

**Matomo**

* Włącz **AnonymizeIP**, **Respect Do Not Track**, **cookieless tracking** lub przynajmniej skrócone cookies.
* **Wyłącz userId** i wszelkie custom dimensions z PII.
* **Referrer Policy** i **dozwolone domeny** ogranicz do własnych.
* Serwuj matomo.js z własnej domeny przez RP.

**Elastic RUM / APM**

* W RUM JS:
  + ustaw **serverUrl** na /rum/intake (RP),
  + ogranicz distributedTracingOrigins do własnych domen,
  + rozważ captureHeaders: false, sanitizeFieldNames (np. \*password\*, \*token\*),
  + unikaj user.id (nie ustawiaj).
* W APM Server/ES: ingest pipeline **drop**: source.ip, url.query, user.\* (jeśli pojawią się).
* Serwuj agenta z własnego hosta.

**Jak to realnie obniża ryzyko (skrót)**

* **Legalność (podstawa)**: łatwiej obronić **uzasadniony interes** lub **zgodę ograniczoną** do first-party, bo **nie dochodzi do zewnętrznej komunikacji** z przeglądarki (mniej podmiotów).
* **Minimalizacja**: IP, nagłówki i query są **anonimizowane „na brzegu”** – zmniejszenie zakresu danych.
* **Rozliczalność**: RP jest jednym, kontrolowanym punktem – możesz wykazać **konfiguracje, logikę zgody i retencję**.
* **Bezpieczeństwo**: TLS/mTLS, WAF, rate-limit, brak ciastek po stronie intake → mniejsza powierzchnia ataku i wycieków.

Uwaga: To nie „magiczna tarcza”. Nadal potrzebne są: polityka prywatności, DPIA, rejestr czynności, okresy retencji, umowy powierzenia (jeśli są podmioty zewnętrzne), obsługa praw osób.

**Szybka checklista wdrożeniowa**

1. Wystaw w RP ścieżki: **/analytics/matomo.js**, **/analytics/collect**, **/rum/agent.js**, **/rum/intake**.
2. Podłącz **CMP** → cookie consent\_analytics=1; dodaj gating (204 No Content bez zgody).
3. W RP: wyczyść **IP**, usuń **Set-Cookie** na intake, dodaj CSP/Referrer-Policy.
4. Na froncie: wskaż nowe endpointy; **wyłącz PII** (userId, custom context), ogranicz URL/referrer.
5. W Matomo/APM/ES: ingest pipeline **drop** pól wrażliwych; ustaw retencję.
6. Udokumentuj DPIA + rejestr czynności dla modułu „Analytics/Observability”.

**Podsumowanie (krótko)**

Reverse proxy daje Ci **pierwszo-partyjne** zbieranie danych, **gating po zgodzie**, **anonimizację IP/nagłówków/URL na brzegu** i **centralną kontrolę** nagłówków, cookies i CORS. To **istotnie obniża ryzyka RODO** (minimalizacja, rozliczalność, bezpieczeństwo), choć nie zastępuje podstaw prawnych i polityk.