Zero-Knowledge Architecture for ARCO Automation

Cómo automatizar derechos ARCO+ sin almacenar datos personales

Junio 2026

Subtítulo: Cómo automatizar derechos ARCO+ sin almacenar datos personales — una arquitectura para el cumplimiento de la Ley 21.719 chilena

Versión: 1.0 — Junio 2026

Autor: PrivacyEngine

Resumen Ejecutivo

La Ley 21.719 de Protección de Datos Personales chilena, vigente desde diciembre de 2026, exige que las empresas procesen los derechos ARCO+ (Acceso, Rectificación, Cancelación, Oposición, Portabilidad, Olvido) de sus clientes en plazos determinados, con la obligación de acreditar dicho cumplimiento.

PrivacyEngine propone una arquitectura **zero-knowledge** para la automatización de derechos ARCO+ en e-commerce. Esta arquitectura permite procesar solicitudes de datos personales en múltiples sistemas simultáneamente, generar un ledger de auditoría criptográficamente firmado (SHA-256) y descartar inmediatamente los datos procesados, sin almacenamiento persistente en ningún momento del ciclo.

Este whitepaper describe la arquitectura, los componentes del sistema, las consideraciones de seguridad y los casos de uso para e-commerce en Shopify, WooCommerce y PrestaShop.

1. Introducción

1.1 Contexto Regulatorio

La Ley 21.719 (publicada en diciembre de 2023, vigente desde diciembre de 2026) establece un marco integral de protección de datos personales en Chile. Entre sus disposiciones más relevantes para el e-commerce se encuentran:

- **Derechos ARCO+:** Acceso, Rectificación, Cancelación, Oposición, Portabilidad y Olvido
- **Plazos de respuesta:** 5-10 días hábiles según el tipo de derecho
- **Obligación de acreditar:** El responsable del tratamiento debe poder demostrar el cumplimiento
- **Sanciones:** Multas de hasta 20.000 UTM (~$1.300.000 USD)

1.2 El Problema

Los e-commerce chilenos almacenan datos de clientes en múltiples sistemas: plataforma de e-commerce (Shopify, WooCommerce), herramientas de marketing (Klaviyo, Mailchimp), CRM, ERP y sistemas de analítica. Procesar una solicitud ARCO requiere actuar sobre todos estos sistemas de forma coordinada, con trazabilidad completa.

Las soluciones actuales presentan limitaciones críticas:

| Solución | Limitación |

| Procesamiento manual | 1 hora/solicitud, propenso a errores, sin trazabilidad |

| Scripts caseros | No orquestan multi-sistema, sin ledger auditable |

| Plataformas enterprise (OneTrust, MineOS) | 10-100x más caras, sin cobertura Ley 21.719, almacenan datos |

1.3 Nuestra Propuesta

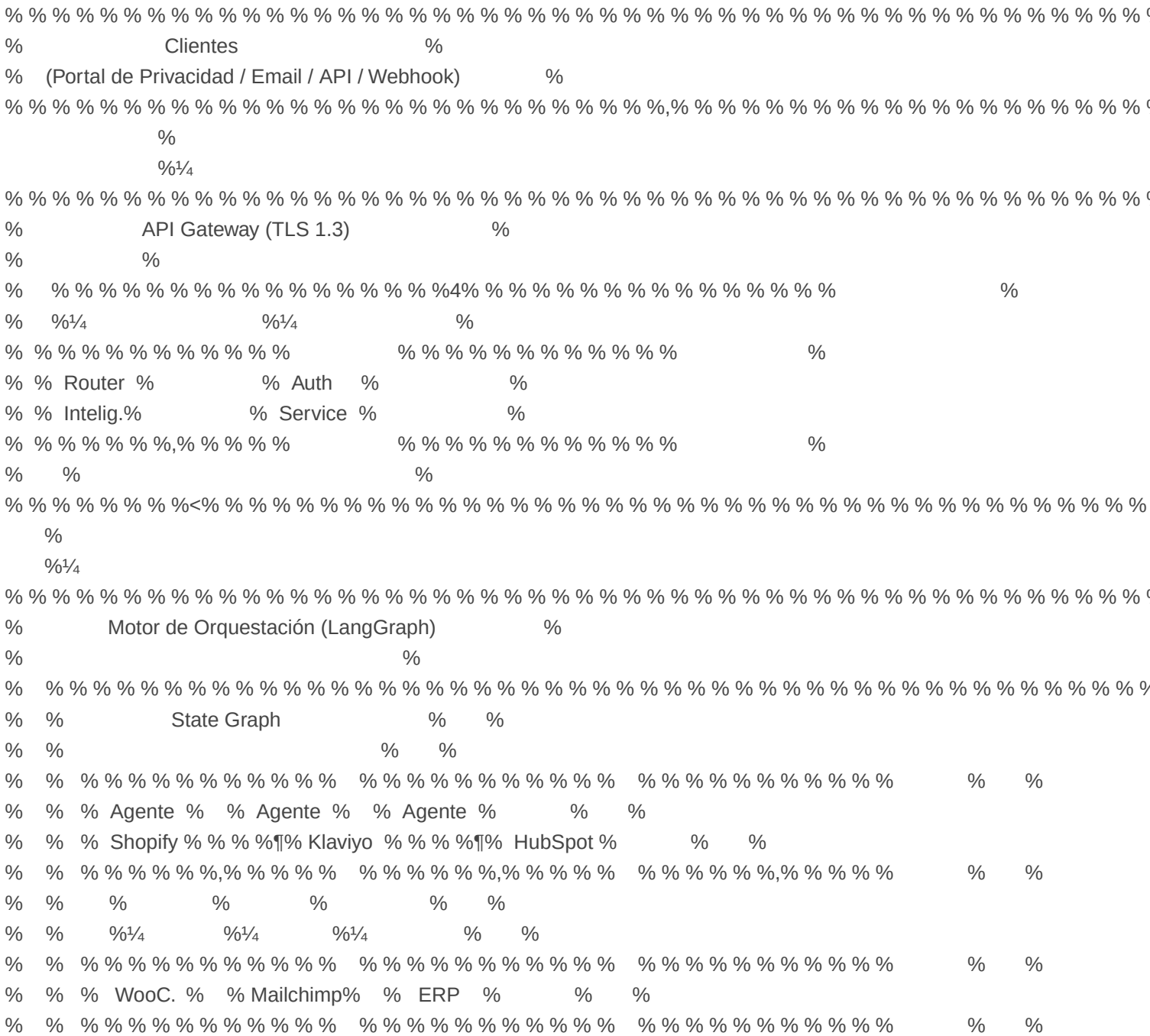
PrivacyEngine propone una arquitectura zero-knowledge que:

1. Recibe solicitudes ARCO a través de un Portal de Privacidad embeddable o API REST
2. Orquesta agentes de IA (LangGraph) para ejecutar acciones en todos los sistemas conectados
3. Genera un ledger SHA-256 inmutable de cada operación
4. Descarta los datos procesados sin almacenamiento persistente

2. Arquitectura del Sistema

2.1 Visión General

PrivacyEngine opera como middleware stateless entre las solicitudes ARCO de los clientes y los sistemas donde residen los datos. La arquitectura sigue el principio de **cero confianza** (zero-trust): ningún componente del sistema confía implícitamente en otro, y todos verifican cada operación.



```
%   %                                         %     %
%   % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
                  %
                  %¼
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%            Ledger SHA-256 (Inmutable)            %
%                                             %
%   Bloque N-1 !• Bloque N !• Bloque N+1 (hash encadenado)      %
%                                             %
%   Cada bloque contiene: timestamp, acción, sistemas, firma     %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
```

2.2 Componentes Clave

2.2.1 API Gateway

Punto de entrada único para todas las solicitudes. Implementa:

- TLS 1.3 para todas las comunicaciones
- Rate limiting por cliente (100 solicitudes/minuto por defecto)
- Validación de formato de solicitud
- Enrutamiento al servicio de autenticación

2.2.2 Servicio de Autenticación y Autorización

Gestiona:

- Tokens OAuth 2.0 para conexiones salientes (Shopify, WooCommerce, APIs de terceros)
- API Keys para conexiones entrantes (sistemas externos que envían solicitudes a PrivacyEngine)
- Scope mínimo: cada token tiene solo los permisos necesarios (leer perfil, eliminar perfil)

2.2.3 Router Inteligente (Clasificador ARCO)

Servicio de clasificación que recibe una solicitud entrante y determina:

1. **Tipo de derecho:** Acceso, Rectificación, Cancelación, Oposición, Portabilidad u Olvido
2. **Identidad del cliente:** Extrae identificador (RUT, email, customer ID) de la solicitud
3. **Sistemas destino:** Determina qué sistemas conectados pueden tener datos de este cliente
4. **Prioridad:** Calcula urgencia basada en plazos legales y SLAs del cliente

2.2.4 Motor de Orquestación (LangGraph)

El corazón del sistema. LangGraph permite definir un **grafo de estado** donde cada nodo es un agente autónomo responsable de un sistema específico.

El flujo del grafo:

1. **Nodo de inicio:** Router despacha la solicitud a agentes relevantes
2. **Nodos de ejecución:** Cada agente se conecta a su sistema, ejecuta la acción y retorna un proof firmado
3. **Nodo de consolidación:** Recibe todos los proofs, verifica consistencia, genera el ledger
4. **Nodo de finalización:** Almacena el ledger, descarta datos en memoria, notifica al solicitante

Características del grafo:

- **Ejecución paralela:** Todos los agentes se ejecutan simultáneamente
- **Aislamiento de fallos:** Si un agente falla, los demás continúan; el ledger registra el fallo
- **Reintentos automáticos:** Hasta 3 reintentos con backoff exponencial
- **Timeout por agente:** Cada agente tiene un timeout configurable (default: 30s)

2.2.5 Agentes de Sistema

Cada agente es un servicio especializado que sabe cómo comunicarse con un sistema específico:

| Agente | Sistema | Protocolo | Acciones Soportadas |

Shopify	Shopify REST/GraphQL API	OAuth 2.0	Leer perfil, eliminar cliente, anonimizar pedidos
WooCommerce	WooCommerce REST API	API Key + Consumer Key	Leer perfil, eliminar cliente, anonimizar pedidos
Klaviyo	Klaviyo API v3	API Key	Buscar perfil, eliminar suscriptor, eliminar de listas
Mailchimp	Mailchimp API v3	OAuth 2.0	Buscar miembro, eliminar miembro, limpiar audiencia
HubSpot	HubSpot CRM API	OAuth 2.0 / API Key	Buscar contacto, eliminar/archivar contacto, eliminar datos
ERP	API REST personalizada	API Key	Buscar cliente, anonimizar facturación
Genérico	API REST	Bearer Token	Configurable por el cliente

2.2.6 Ledger SHA-256

Registro inmutable de todas las operaciones. Cada entrada tiene la siguiente estructura:

```
{
  "ledger_entry": {
    "version": "1.0",
    "entry_id": "ledger_20260615_001234",
    "previous_hash": "sha256$a1b2c3d4e5f6...",
    "timestamp": "2026-06-15T14:30:00.000Z",
    "solicitud": {
      "id": "REQ-2026-06-001234",
      "tipo": "cancelacion",
      "cliente_hash": "sha256$e5f6g7h8...",
      "canal": "portal_privacidad"
    },
    "acciones": [
      {
        "sistema": "shopify",
        "status": "completado",
        "accion": "eliminar_cliente",
        "timestamp": "2026-06-15T14:30:01.000Z",
        "proof_hash": "sha256$i9j0k1l2..."
      },
      {
        "sistema": "klaviyo",
        "status": "completado",
        "accion": "eliminar_suscriptor",
        "timestamp": "2026-06-15T14:30:02.000Z",
        "proof_hash": "sha256$m3n4o5p6..."
      }
    ],
    "firma": "sha256$firma_digital_nodo",
    "nonce": "a1b2c3d4"
  }
}
```

2.3 Zero-Knowledge: Ausencia de Almacenamiento de Datos

La característica central de PrivacyEngine es la arquitectura zero-knowledge. Esto se implementa mediante:

1. **Procesamiento en memoria volátil:** Todos los datos de clientes recibidos se mantienen exclusivamente en RAM durante el procesamiento.

2. **Descarte inmediato:** Una vez generado y firmado el ledger, los datos en memoria se descartan explícitamente.
3. **Sin escritura en disco:** Los datos de clientes nunca se escriben en disco, base de datos, caché persistente ni logs del sistema.
4. **Ledger sin datos personales:** El ledger contiene hashes SHA-256 de los identificadores de clientes, nunca los identificadores en texto plano.
5. **Cifrado en tránsito:** Todos los datos se transmiten cifrados (TLS 1.3). No hay endpoints sin cifrar.

3. Seguridad

3.1 Modelo de Amenazas

Amenaza	Mitigación
Acceso no autorizado a la API	Tokens OAuth 2.0 + rate limiting + validación
Interceptación de datos en tránsito	TLS 1.3 obligatorio en todas las comunicaciones
Compromiso del servidor PrivacyEngine	Zero-knowledge: no hay datos de clientes que robar
Modificación del ledger	Cadena de hashes SHA-256 + firmas digitales
Fuga de datos por logging	Logs sin datos personales (solo hashes y metadatos)
Ataque de repetición	Nonce + timestamp en cada solicitud

3.2 Principios de Seguridad por Diseño

1. **Mínimo privilegio:** Cada conexión a sistemas externos usa tokens con los permisos mínimos necesarios
2. **Defensa en profundidad:** Múltiples capas de seguridad (TLS, auth, validación, logging)
3. **Privacidad por defecto:** Sin almacenamiento de datos personales
4. **Auditable:** Cada operación deja traza criptográfica verificable

4. Casos de Uso

4.1 Solicitud de Cancelación

Escenario: Un cliente solicita la eliminación de todos sus datos personales.

Flujo:

1. El cliente envía la solicitud a través del Portal de Privacidad (2 min)
2. PrivacyEngine recibe: tipo=cancelación, identificador=cliente@email.com
3. Router clasifica: sistemas destino = {Shopify, Klaviyo, HubSpot}
4. Motor despacha agentes en paralelo a los 3 sistemas
5. Cada agente ejecuta la eliminación y retorna proof
6. Agente de consolidación verifica los 3 proofs
7. Ledger generado con timestamp y firmas
8. Notificación al cliente: "Solicitud #1234 procesada exitosamente"

Tiempo total: ~2 segundos (sin contar el tiempo del cliente para enviar la solicitud)

4.2 Solicitud de Acceso

Escenario: Un cliente solicita conocer todos los datos que la empresa tiene de él.

Flujo:

1. El cliente envía la solicitud de acceso
2. PrivacyEngine recibe: tipo=acceso, identificador=cliente@email.com
3. Motor despacha agentes para leer datos en todos los sistemas
4. Cada agente lee los datos del cliente y los devuelve cifrados

5. Agente de consolidación compila el reporte
6. Reporte se envía al cliente por email seguro
7. Ledger registra: "Solicitud de acceso completada — datos entregados"

5. Integración con E-commerce

5.1 Shopify

- **Método:** OAuth 2.0 vía Shopify App Store
- **Datos requeridos:** customer_id, email
- **Acciones:** DELETE /admin/api/2024-10/customers/{id}.json, anonimizar pedidos asociados
- **Tiempo de implementación:** 5 minutos (instalar app + autorizar)

5.2 WooCommerce

- **Método:** Plugin + API Key + Consumer Key
- **Datos requeridos:** customer_id, email
- **Acciones:** DELETE /wp-json/wc/v3/customers/{id}, anonimizar pedidos
- **Tiempo de implementación:** 15 minutos (instalar plugin + configurar)

5.3 Integraciones de Marketing

- **Klaviyo:** API Key v3 — buscar perfil por email, eliminar perfil y de todas las listas
- **Mailchimp:** OAuth 2.0 — buscar miembro por email, eliminar de audiencia
- **HubSpot:** OAuth 2.0 — buscar contacto por email, archivar + eliminar datos personales
- **Genéricas:** API REST configurable con webhooks

6. Sobre los Autores

PrivacyEngine es un middleware de automatización ARCO+ para e-commerce chileno, desarrollado con arquitectura zero-knowledge y motor multi-agente LangGraph. Más información en [privacyengine.dev]().

Este whitepaper es un documento vivo. Para la versión más reciente, visite privacyengine.dev/docs/whitepaper

Descargo de responsabilidad: Este documento tiene fines informativos y no constituye asesoría legal. Para cumplimiento normativo específico, consulte a un abogado especializado en protección de datos.