

Bilko ADR-016: E-Invoice Adapter

Author: ALAI, 2026

ADR-016: E-Invoice Adapter & UBL 2.1 Canonical Model

Status: Accepted **Date:** 2026-04-21 **Author:** ALAI, 2026 **Related:** ADR-015 (Four-Jurisdiction Plugin), ADR-019 (Integration Adapter Registry) ---

Context

Each of Bilko's four tax jurisdictions mandates **electronic invoicing** via government-operated fiscal platforms: | Jurisdiction | Platform | Format | Mandatory Since | Inbound Support | |
----- | ----- | ----- | ----- | ----- | | RS (Serbia) | SEF (efaktura.gov.rs) | UBL 2.1 SEF envelope | 2023 | Yes (B2B) | | HR (Croatia) | HR-FISK / FINA eRacun | UBL 2.1 FINA XML | Jan 2026 | Yes (B2B) | | BA-FED (Bosnia FBiH) | CPF | Unspecified (stub) | TBD 2027 | Unknown | | BA-RS (Bosnia RS) | UINO | Unspecified (stub) | TBD | Unknown |

Problem Statement

Bilko's **invoice domain** must: 1. Generate invoices in a **canonical format** independent of any single jurisdiction 2. **Serialize** to jurisdiction-specific XML/JSON for submission 3. **Submit** to fiscal platforms with retry/error handling 4. **Poll** for status updates (async platforms) 5. **Parse** inbound invoices received from suppliers (B2B procurement) **Without:** Embedding SEF/FINA-specific logic in the core `Invoice` model Duplicating invoice validation across 4 jurisdictions Tight coupling to any single platform's API changes

Design Decision Drivers

Vendor patterns adopted: SAP B1 Electronic Filing Manager (EFM): Pluggable adapter layer per jurisdiction canonical UBL 2.1 core **EN 16931 European e-invoicing standard:** Defines minimal invoice semantic model **UBL 2.1 (ISO/IEC 19845):** OASIS Universal Business Language —

XML serialization **Key insight from SAP:** The `_canonical model_` should reflect **accounting semantics**, not platform wire formats. Adapters translate from canonical → platform-specific. ---

Decision

1. Canonical Invoice Model — EN 16931 Subset

Bilko's internal invoice representation is a **Kotlin data class** implementing the **EN 16931 Core Invoice Model** (mandatory BT fields optional BG groups). **File:** `CanonicalInvoice.kt`` (see `~/ALAI/products/Bilko/scratch-api/src/main/kotlin/no/alai/bilko/einvoice/CanonicalInvoice.kt``) **Key design rules:** 1. All monetary amounts: `BigDecimal`` with 4 decimal places (ADR-002) 2. All dates: `kotlinx.datetime.LocalDate`` 3. VAT categories: UN/ECE 5305 codes (``S`,`Z`,`E`,`AE`,`O`,`K`,`G``) 4. Invoice types: UNTDID 1001 codes (380 = commercial invoice, 381 = credit note, 383 = debit note) 5. Currency codes: ISO 4217 (RSD, EUR, BAM) 6. Country codes: ISO 3166-1 alpha-2 (RS, HR, BA) **Mandatory fields (EN 16931 BT numbering in KDoc):** BT-1: Invoice number BT-2: Issue date BT-3: Invoice type code BT-5: Currency code BT-9: Due date BG-4: Supplier party (name, tax ID, address) BG-7: Buyer party (name, tax ID, address) BG-23: VAT breakdown per category BG-25: Invoice lines (minimum 1 line) **Extension point:** `val adapterMetadata: Map = emptyMap()` For jurisdiction-specific fields not covered by EN 16931 (e.g., ``sef.requestId`,`hr.fiscalCode``). Namespaced by adapter.

2. EInvoiceAdapter Interface

All jurisdiction-specific e-invoice integrations **must** implement this contract. **File:** `EInvoiceAdapter.kt`` (see `~/ALAI/products/Bilko/scratch-api/src/main/kotlin/no/alai/bilko/einvoice/EInvoiceAdapter.kt``) **Four methods (ADR-019 lifecycle contract):** `interface EInvoiceAdapter { val jurisdiction: TaxJurisdiction val lifecycleState: AdapterLifecycleState // STUB | SANDBOX_VERIFIED | PRODUCTION_CERTIFIED @Throws(AdapterException::class) fun serialize(invoice: CanonicalInvoice): ByteArray @Throws(AdapterException::class) fun submit(serializedInvoice: ByteArray, invoice: CanonicalInvoice): SubmitResult @Throws(AdapterException::class) fun pollStatus(submissionId: String, invoice: CanonicalInvoice): EInvoiceStatus @Throws(AdapterException::class) fun parseIncoming(rawPayload: ByteArray): CanonicalInvoice }` **Lifecycle states (ADR-019):** ``STUB``: Not implemented — all methods throw `AdapterException(NOT_IMPLEMENTED)`` ``SANDBOX_VERIFIED``: Tested against fiscal platform sandbox (e.g., SEF demo env) ``PRODUCTION_CERTIFIED``: Audited and approved for live traffic **Error normalization (ADR-019):** All adapters throw `AdapterException(code, market, retryable, rawPayload)`` — never platform-native exceptions. Canonical error codes: ``NETWORK_TIMEOUT`,`AUTH_TOKEN_EXPIRED`,`VALIDATION_SCHEMA_ERROR`,`PLATFORM_MAINTENANCE``, etc.

3. Adapter Implementations — Phase 1

Priorities

| Jurisdiction | Adapter Class | Lifecycle State (2026-04-22) | Notes | | ----- | ----- |
----- | ----- | | RS | `SEFInvoiceAdapter` |
SANDBOX_VERIFIED | Outbound inbound implemented (MC #8682) | | HR |
`HRFISKEInvoiceAdapter` | STUB | Blocked on FINA certificate (CEO decision pending) | | BA-FED |
`CPFInvoiceAdapter` | STUB | CPF spec not published | | BA-RS | `UINOInvoiceAdapter` | STUB |
UINO spec unavailable | **SEF Serbia (RS) — Reference Implementation:** Serialization: UBL 2.1
XML SEF JSON envelope Submission: HTTPS POST to
`https://efaktura.mfin.gov.rs/api/publicApi/invoice` Authentication: X.509 certificate API key Polling:
GET `/api/publicApi/invoice/{id}/status` Inbound: Webhook `/api/invoices/incoming` polling
`/api/publicApi/inbox` **SEF sandbox certification (Task 1.5):** 5 invoice types tested in sandbox:
1. B2B outbound invoice 2. B2G outbound invoice (to government buyer) 3. Credit note 4.
Cancelled invoice 5. Inbound invoice received from supplier **Evidence:** Acknowledgement IDs
returned by real SEF demo environment (MC #8682 DoD).

4. Canonical ? Platform Serialization Flow

sequenceDiagram participant Core as Invoice Service participant Plugin as CountryPlugin (RS)
participant Adapter as SEFInvoiceAdapter participant SEF as SEF Platform Core->>Plugin:
generateInvoiceXml(invoice) Plugin->>Adapter: serialize(invoice) Adapter->>Adapter: Map
CanonicalInvoice → UBL 2.1 XML Adapter-->>Plugin: ByteArray (XML) Plugin->>Adapter:
submit(xml, invoice) Adapter->>SEF: POST /api/publicApi/invoice SEF-->>Adapter: HTTP 200
platformInvoiceId Adapter-->>Plugin: SubmitResult(platformInvoiceId, PENDING) Plugin-->>Core:
FiscalSubmissionHandle **Key invariant:** The `Invoice` model in `packages/database` **never**
contains SEF-specific fields. All jurisdiction logic flows through the plugin and adapter layers. ---

Consequences

Positive

- Platform independence:** When SEF changes XML schema (happened 2024), only `SEFInvoiceAdapter` changes. Core untouched.
- Testability:** Each adapter has isolated unit tests. Mock platforms for regression.
- Incremental rollout:** HR/BA adapters can remain stubs until fiscal platforms are ready.
- B2B procurement:** Inbound invoices parse through `parseIncoming()` → canonical model → standard invoice creation flow.

Negative

- Mapping complexity:** UBL 2.1 has 200+ optional fields. CanonicalInvoice supports ~40.

Adapter must decide what to include. 2. **Round-trip fidelity:** Parsing inbound invoice → canonical → serialize may lose platform-specific metadata. Use `adapterMetadata` map. 3. **Certification burden:** Sandbox testing required per adapter before production (Phase 1 Task 1.5).

Risks

1. **SEF/FINA breaking changes:** Government platforms have no SLA, no deprecation policy. **Mitigation:** Adapter feature flags (ADR-019 Task 4.5) — disable broken adapter without redeploy.
2. **CPF/UINO spec delays:** BiH platforms have no published tech specs. **Mitigation:** Stub adapters document `NOT_IMPLEMENTED`; GA proceeds without BiH e-invoicing.
3. **UBL 2.1 extensions:** Some platforms extend UBL with custom namespaces. **Mitigation:** `adapterMetadata` carries extension data; serialization handles custom namespaces. ---

Implementation Notes

Validation Strategy

EN 16931 validation (core): Invoice must have ≥ 1 line `sum(lines.lineTotal) == invoice.totalAmountExclVat` `totalAmountExclVat totalVatAmount` == `totalAmountInclVat` All amounts ≥ 0 (credit notes use negative `quantity`, not negative `unitPrice`) Jurisdiction validation (adapter): SEF Serbia: Buyer VAT ID must match PIB format (9 digits) HR Croatia: Supplier must be FINA-registered (OIB validation) Validation errors throw AdapterException(VALIDATION_BUSINESS_RULE, retryable=false)``

Async Submission Pattern (Transactional Outbox)

SEF and HR-FISK are **asynchronous**. Recommended pattern: // 1. Persist invoice to DB (transaction T1) // 2. Write to {market}_outbox table (still in T1) // 3. Commit T1 // 4. Background worker polls outbox, calls adapter.submit() // 5. Update outbox row with platformInvoiceId // 6. Background worker polls adapter.pollStatus() until APPROVED/REJECTED **Do NOT block invoice creation** on fiscal platform availability.

Sandbox Environments

Platform	Sandbox URL	Credential Type	-----	-----
SEF (RS)	https://demo-efaktura.mfin.gov.rs	Test X.509 cert API key		HR-FISK (HR)
https://demo.fiskalizacija.hr	Test FINA certificate		CPF (BA-FED)	N/A Not available
UINO (BA-RS)	N/A Not available	Sandbox credential convention (ADR-019): Secret store path: `Bilko/sandbox/{market}/{secret-name}` Example: `Bilko/sandbox/RS/sef-api-key` ---		

References

EN 16931 spec: <https://ec.europa.eu/digital-building-blocks/sites/display/DIGITAL/Compliance+with+eInvoicing+standard> **UBL 2.1 spec:** <http://docs.oasis-open.org/ubl/UBL-2.1.html> **SEF Serbia docs:** <https://www.efaktura.gov.rs/en> **HR-FISK Croatia:** https://www.porezna-uprava.hr/HR_Fiskalizacija/Stranice/Naslovna.aspx
Implementation: `~/ALAI/products/Bilko/scratch-api/src/main/kotlin/no/alai/bilko/einvoice/`
Master plan: `~/system/specs/bilko-multi-market-architecture-plan.md` (Phase 1 Task 1.4, Task 1.5) **Related ADRs:** - ADR-015: Four-Jurisdiction Plugin Architecture - ADR-019: Integration Adapter Registry ---

Approval

Approved: 2026-04-21 by CEO Alem Basic **Execution:** Phase 0 Task 0.2, Phase 1 Task 1.4 (completed 2026-04-22, MC #8682)

Revision #2

Created 2026-04-22 10:36:34 UTC by John

Updated 2026-05-31 20:06:24 UTC by John