

# Bilko ADR-018: Market vs Locale

Author: ALAI, 2026

```
# ADR-018: Market vs Locale Separation

**Status:** Accepted
**Date:** 2026-04-21
**Author:** ALAI, 2026
**Related:** ADR-015 (Four-Jurisdiction Plugin), ADR-017 (RLS Multi-Tenancy)

---

## Context

Bilko serves 4 tax jurisdictions (markets) but must support 5+ locales (languages/scripts). These are orthogonal concerns but were conflated in early prototypes.

Problem example:

- A company in Belgrade, Serbia (RS market) may want UI in Serbian Latin (`sr-Latn`)
- A company in Banja Luka, Bosnia RS entity (BA-RS market) also wants UI in Serbian Latin (`sr-Latn`)
- Same locale, different markets – different tax rates, different filing authorities

Old mistake: Hardcoding `market = locale` (e.g., `if locale == 'sr' then jurisdiction = RS`) breaks when:

1. Diaspora companies (Serbian company in Norway uses `nb` locale, `RS` market)
2. Multilingual jurisdictions (BiH supports `bs`, `sr`, `hr` locales, 2 markets)
3. English-speaking accountants working on local entities

Goal: Separate MarketContext (tax jurisdiction, currency, CoA, fiscal platform) from LocaleContext (UI language, number format, date format).

---

## Decision

### 1. Two Independent Contexts

MarketContext:

- Source of truth: `Organization.taxJurisdiction` (enum: `RS | HR | BA_FED | BA_RS`)
- Read from: JWT claim `org.taxJurisdiction` at request time
- Controls: VAT rates, currency, e-invoice adapter, fiscal platform, CoA version, retention policy
- Injected via: `CountryPlugin` interface (ADR-015)

LocaleContext:

- Source of truth: User preference (`User.preferredLocale`) or browser `Accept-Language` header
- Supported locales: `{sr-Latn, sr-Cyrl, hr, bs, en}` (Phase 3 Task 3.1)
- Controls: UI strings (i18n), number formatting, date formatting, currency symbol display
- Injected via: Next.js `next-intl` provider

Key invariant: Market and locale are independent variables. A user can select any locale regardless of organization market.

### 2. Locale Matrix (Supported Combinations)

| Locale | Script | Markets Supporting | Notes
```

`sr-Latn` RS entity)	Latin	RS, BA-RS	Serbian Latin (default for Serbia, common in RS entity)
`sr-Cyrl` common in practice)	Cyrillic	RS, BA-RS	Serbian Cyrillic (official in Serbia, less common in practice)
`hr` BiH)	Latin	HR, BA-FED	Croatian (Croatia + Croat-majority areas of BiH)
`bs` Federation)	Latin	BA-FED, BA-RS	Bosnian (official in BiH Federation)
`en` accountants)	Latin	All	English (for international accountants)

**\*\*User flow:\*\***

1. User logs in ? JWT contains `org.taxJurisdiction` (e.g., `BA\_FED`)
2. User selects UI locale ? stored in `User.preferredLocale` (e.g., `bs`)
3. Frontend: `MarketContext.value = BA\_FED`, `LocaleContext.value = bs`
4. Invoice wizard shows BiH-specific VAT fields (17% flat rate) in Bosnian language

### ### 3. Routing Decision – Path Prefix, Not Subdomain

**\*\*Options evaluated:\*\***

- **\*\*Option A (subdomain):\*\*** `bilko.rs`, `bilko.hr`, `bilko.ba` ? different deployments per market
- **\*\*Option B (path prefix):\*\*** `bilko.io/rs/...`, `bilko.io/hr/...`, `bilko.io/ba/...` ? single deployment

**\*\*Decision: Option B (path prefix)\*\***

**\*\*Rationale:\*\***

1. **\*\*Single deployment\*\*** – reduces infra complexity (one Docker image, one DB, one domain cert)
2. **\*\*Locale independence\*\*** – user can switch locale without changing URL market segment
3. **\*\*SEO flexibility\*\*** – `/en/pricing` vs `/sr/cene` share same market routing
4. **\*\*Cloudflare Transform Rule\*\*** injects `X-Market` header from path prefix (Task 4.2)

**\*\*URL structure:\*\***

```

...
bilko.io/           ? Landing page (market preselection)
bilko.io/rs/...    ? Serbia market
bilko.io/hr/...    ? Croatia market
bilko.io/ba-fed/... ? BiH Federation market
bilko.io/ba-rs/... ? BiH RS entity market
bilko.io/en/pricing ? English pricing page (market-agnostic)
...

```

**\*\*Backend ignores `X-Market` header when JWT present\*\*** (Kelsey security rule – Task 4.2). Only use path prefix for **\*\*unauthenticated\*\*** flows (landing pages, marketing).

### ### 4. Frontend Implementation (Phase 3)

**\*\*MarketProvider (Task 3.1):\*\***

```

````tsx
// apps/web/lib/market-context.tsx
const MarketContext = createContext<TaxJurisdiction | null>(null)

export function MarketProvider({ children }: { children: ReactNode }) {
  const jwt = useJWT() // Read from httpOnly cookie
  const market = jwt?.org?.taxJurisdiction ?? null

  return <MarketContext.Provider value={market}>{children}</MarketContext.Provider>
}

export function useMarket() {
  const market = useContext(MarketContext)
  if (!market) throw new Error('useMarket must be within MarketProvider')
  return market
}
...

```

**\*\*LocaleProvider (next-intl):\*\***

```

````tsx
// apps/web/app/[locale]/layout.tsx
import { NextIntlClientProvider } from 'next-intl'

```

```
import { getMessages } from 'next-intl/server'

export default async function LocaleLayout({
  children,
  params: { locale },
}: {
  children: ReactNode
  params: { locale: string }
}) {
  const messages = await getMessages(locale)

  return (
    <NextIntlClientProvider locale={locale} messages={messages}>
      {children}
    </NextIntlClientProvider>
  )
}

```

**\*\*Dynamic component import based on market:\*\***

```
````tsx
// apps/web/app/[locale]/dashboard/vat-return/page.tsx
import { useMarket } from '@lib/market-context'
import dynamic from 'next/dynamic'

export default function VATReturnPage() {
  const market = useMarket()

  const VATReturnComponent = dynamic(
    () => import(`@bilko/country-${market.toLowerCase()}/components/VATReturn`),
  )

  return <VATReturnComponent />
}

```

Each jurisdiction package exports market-specific components:

- `@bilko/country-rs/components/VATReturnRS` (Serbia PPPDV form)
- `@bilko/country-hr/components/VATReturnHR` (Croatia PDV-S form)
- `@bilko/country-ba-fed/components/VATReturnBAFED` (BiH FBiH PDV-1 form)
- `@bilko/country-ba-rs/components/VATReturnBARS` (BiH RS PDV form)

### 5. Formatters – Injected from CountryPlugin

**\*\*Problem:\*\*** Number and date formatting is locale-dependent **\*\*and\*\*** market-dependent:

- Serbia uses `.` as decimal separator (123.456,78)
- Croatia uses `.` as decimal separator (123.456,78)
- US English uses `,` as decimal separator (123,456.78)

**\*\*Solution:\*\*** `JurisdictionFormatters` interface (ADR-015) provides formatters per market.

**\*\*Kotlin side (CountryPlugin):\*\***

```
````kotlin
interface JurisdictionFormatters {
  fun formatAmount(amount: BigDecimal, currency: Currency): String
  fun formatDate(date: LocalDate): String
  fun formatTaxRate(rate: BigDecimal): String
}

class PluginRS : CountryPlugin {
  override fun getFormatters() = object : JurisdictionFormatters {
    override fun formatAmount(amount: BigDecimal, currency: Currency) =
      "%.2f %s".format(Locale("sr", "RS"), amount, currency.currencyCode)
    // Output: "123.456,78 RSD"
  }
}

```

**\*\*Frontend side (React):\*\***

```
````tsx
// packages/ui/components/molecules/AmountDisplay.tsx
import { useMarket } from '@lib/market-context'

export function AmountDisplay({ amount, currency }: { amount: number; currency: string }) {
  const market = useMarket()
}

```

```

const formatter = getFormatterForMarket(market) // Fetched from API or config

return <span>{formatter.formatAmount(amount, currency)}</span>
}
...

**Zero market-specific logic in `packages/ui`** - all formatting logic injected from country
plugin.

---

## Consequences

### Positive

1. **True internationalization:** Diaspora companies (e.g., Serbian company in Germany) can
use `de` locale with `RS` market
2. **User choice:** Same org can have users in different locales (accountant in English, CEO
in Serbian)
3. **Marketing flexibility:** Landing pages can be localized without locking market
selection
4. **Clean abstractions:** UI components never contain market conditionals (`if market ==
RS`)

### Negative

1. **Complexity for users:** Market vs locale distinction may confuse non-technical users
("Why do I choose country twice?")
2. **Translation burden:** 5 locales × 287 UI strings (Task 3.4) = 1,435 translation units
3. **Formatter duplication:** Each market needs formatters for each locale (e.g., `RS + en`
vs `RS + sr-Latn`)

### Risks

1. **Locale confusion:** User selects `hr` locale on `RS` market ? sees Croatian UI for
Serbian taxes. **Mitigation:** Locale picker shows recommended locale per market.
2. **Formatter bugs:** Incorrect decimal separator (`,` vs `.`) can cause accounting errors.
**Mitigation:** Unit tests for all formatter combinations.

---

## Implementation Notes

### Market Badge (Visual Cue)

Gold accent badge in top-bar:

```tsx
// packages/ui/components/atoms/MarketBadge.tsx
export function MarketBadge({ market }: { market: TaxJurisdiction }) {
  const labels = {
    RS: 'Srbija',
    HR: 'Hrvatska',
    BA_FED: 'BiH Federacija',
    BA_RS: 'BiH Republika Srpska',
  }

  return <div className="bg-gold/10 text-gold px-3 py-1 rounded-full text-
sm">{labels[market]}</div>
}
...

**Design constraint:** Badge is **gold accent** (`#F2C87A`), not market-specific color.
Branding remains singular (plum `#8B6BBF`).

### i18n Coverage Audit (Task 3.4)

**Current state (2026-04-10 audit):**

- 287 hardcoded strings in `apps/web` components
- 0 strings in `messages/{locale}.json`

**Target state (Phase 3 completion):**

```bash
npm run i18n:audit
# Expect: 0 hardcoded strings
```

**Tooling:** ESLint rule `no-hardcoded-strings` (enforced in CI after migration).

```

---

## ## References

- **Next.js i18n:** <https://next-intl-docs.vercel.app/>
- **ISO 639-1 locale codes:** [https://en.wikipedia.org/wiki/List\\_of\\_ISO\\_639-1\\_codes](https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)
- **Master plan:** `~/system/specs/bilko-multi-market-architecture-plan.md` (Phase 3 Tasks 3.1-3.4)
- **Related ADRs:**
  - ADR-015: Four-Jurisdiction Plugin Architecture (defines TaxJurisdiction enum)
  - ADR-017: RLS Multi-Tenancy (country\_code column, separate concern from locale)

---

## ## Approval

**Approved:** 2026-04-21 by CEO Alem Basic

**Execution:** Phase 3 Tasks 3.1-3.4 (not yet started – blocked on Phase 1 completion)

---

## Revision #2

Created 2026-04-22 10:36:59 UTC by John

Updated 2026-05-31 20:06:26 UTC by John