Automated I18n Quality for Enterprise Platforms

Globalization Readiness

- Linguistic Quality
- Extensibility
- Maintainability
- Time to market
- Portability (standard based)

Reactive vs Proactive

re: fix bugs, correct translations, troubleshoot, but costumer will find issues before you

Prevent bugs, establish best practices that are global ready

Using AI out of the box

goose: Agentic vibe coding, but it dose not use ICU, does not deal with data ready for i18n.

LLM->most common, but statistically wrong.

- Not using standard region codes.
- Assumes only one language per region
- Assumes only two forms for plural
- Sloppy plural(s) construct in some languages
- No gender handling
- Embeds formatting and layout with content
- Content for all locales in a single file
- (not shown)
- Poor phone structure as raw text
- No attempt to find or use libraries for phone, address, or to CU or CLDR

Detect Issues in source content

- Before antering the translation pipeline
- Within Atlas, a plafform for managing localization workflows
- Rulebased linting
- Using 3rd party lib: ilib-lint

Github -> CI□□□□□□□□ -> AWS -> Management platform ->Github/CI/Translator vendor

Detect issues in source code

- Independent of translatable content
- Much larger dataset
- Build a custom scanner
- Static Analysis + AI
- Many programming language
- Custom integrations

i18n using AI + Self-Healing

Sourcecode I18n self healing using AI study

- Scan-train-refine
- Knowend and discovered

Going forward with AI

- i18n anti patter development
- Scanning tool development
- Fine tuning results
- AI Training
- Self-healing training
- CI/CD Intergration

Revision #1
Created 13 November 2025 17:06:16 by itsLittleKevin
Updated 13 November 2025 17:44:12 by itsLittleKevin