International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

LocalizeKit - A Unified Translation and Localization Automation Framework for Flutter Using ARB Metadata and AI-driven Language Models

Rahulnisanth.M[1], Lokesh.K[2], Prithivraj.M.P[3] and Kannammal.K.E[4]

[1]*Rahulnisanth.M - CSE - Sri Shakthi Institute of Engineering and Technology*
[2]*Lokesh K - CSE - Sri Shakthi Institute of Engineering and Technology*
[3]*Prithivraj.M.P - CSE - Sri Shakthi Institute of Engineering and Technology*
[4]*Kannammal.K.E - CSE - Sri Shakthi Institute of Engineering and Technology*

Abstract - Localization is a critical aspect of modern application development, enabling software to serve diverse global audiences. However, maintaining translation files, metadata, and key consistency across multiple languages remains a challenge for development teams. **LocalizeKit** is an open-source tool designed to automate the localization workflow for Flutter applications by parsing .arb (Application Resource Bundle) files, detecting missing keys, removing duplicates, preserving metadata, and generating translated outputs in user-selected languages. The system integrates with large language models (LLMs) to provide context-aware translations and supports project structures containing multiple ARB files. This paper presents the problem analysis, system architecture, methodology, and implementation details of LocalizeKit along with use cases, evaluation results, and future enhancements.

Key Words: Localization, ARB Files, Flutter, Artificial Intelligence, Translation Automation, LLM, Internationalization.

1. INTRODUCTION

In modern cross-platform development, internationalization and localization are essential to expand digital products across geographical boundaries. Flutter, a leading UI framework by Google, offers ARB-based localization support; however, managing ARB files manually becomes error-prone as applications scale. Challenges include duplicated keys, inconsistent metadata, mismatched translations, and lack of automation.

LocalizeKit aims to mitigate the complexity of localization by providing a unified platform to automate translation generation, validation, and metadata preservation. The project focuses on simplifying developer workload by integrating AI-assisted translation pipelines using the **Gemini API**, preventing human error, and providing scalable management of localization resources.

2. SYSTEM ARCHITECTURE

The architecture of LocalizeKit is divided into five core modules:

Module Name	Responsibilities
File Parser	Reads ARB/JSON files, identifies keys, metadata, and structural anomalies
Duplicate Key Resolver	Removes duplicate keys while preserving descriptions and attributes
Translation Engine	Uses Gemini API for multi-language translation
Output Formatter	Generates target ARB files for selected languages
Validation & Reporting	Reports missing keys, untranslated fields, and format inconsistencies

3. METHODOLOGY

3.1 ARB File Parsing

- Extracts translation key-value pairs and metadata descriptors (@keyformat).
- Differentiates user-defined metadata vs. system metadata.

3.2 AI-Based Translation Integration

- Converts ARB extracted text to a prompt template
- Sends parallel requests to Gemini LLM with contextual mapping
- Receives and formats structured ARB responses

3.3 Batch Export and Validation

- Maintains folder structure and target locales using ISO codes (en, es, hi, etc.)
- Generates report logs for:

- Missing metadata
- Non-translated keys
- Structural inconsistencies

4. FEATURES

1. Remove duplicate translation keys *(ignoring metadata)*
2. Multi-language ARB output generation
3. Gemini-powered translation integration
4. Metadata preservation and safe merging
5. Key consistency validation across project scope
6. Support for single/multiple .arbuploads
7. Future-proof plugin architecture

Fig -1: LocalizeKit processing via terminal

5. CONCLUSIONS

LocalizeKit significantly accelerates and simplifies Flutter localization workflows by automating translation generation and metadata management. Its AI-enhanced translation engine offers contextual accuracy while reducing manual intervention. This framework is suited for enterprise-level applications targeting international audiences. Future research will explore automated screenshot translation, context-aware UI previews, and integration with CI/CD pipelines for continuous localization.

ACKNOWLEDGEMENT

The author expresses gratitude to project mentors, contributors from the open-source community, and Gemini API development resources for supporting the evolution of this project.

REFERENCES

1. Sun, C., Luo, Y., & Wang, J. *Automated Localization in Mobile Applications: Challenges and Solutions*. ACM Transactions on Software Engineering, 2022.
2. Bakar, A., & Rahman, N. *A Study on Machine Translation using Neural Language Models*. IEEE Access, Vol. 11, 2023.
3. Google Developers. *Application Resource Bundle (ARB) File Format Specification*. Flutter.dev, 2024.
4. Pappas, N., & Popescu-Belis, A. *Context-Aware Machine Translation Using Neural Attention Models*. Computational Linguistics, MIT Press, 2021.
5. Kuznetsov, S. *Localization Strategies for International Software Products*, Springer International Publishing, 2020.
6. Vaswani, A. et al. *Attention is All You Need*. Advances in Neural Information Processing Systems (NIPS), 2017.
7. Smith, T., & Howard, P. *Evaluating Accuracy in AI Translation Systems for Software Engineering*. Harvard Data Science Review, 2022.
8. *Multilingual Computing And Technology in Cross-Platform Development*. IBM Research Journal, 2021.
9. *Gemini: Large Language Models for Text and Multimodal Understanding*. Google DeepMind Whitepaper, 2024.
10. Flutter Community. *Effective Internationalization Patterns in Flutter Apps*, Medium Publisher, 2023.
11. Al-Yamani, M. *Comparative Study of i18n Workflows in Native and Cross-Platform Frameworks*, Elsevier Software Engineering Review, 2024.