

Libraries are heavily used in modern software development by developers to bring in functionality already implemented by other developers into an application.

Such dependence on third-party code brings issues regarding maintenance, compatibility, and evolution for both the client and library software developer.

In this thesis, we analyze the relation and frontiers that exist between libraries and client software.

A particularity of applications using libraries is the presence and consumption of Application Programming Interfaces (APIs).

These APIs are the main interaction frontier the developer uses to consume a specific library feature.

This frontier allows for data to be transferred in both directions from the client to the library, and from the library to the client.

Library developers often lack usage knowledge on the uses of their libraries, preventing proper and safe evolution.

In contrast, client developers often face difficulties finding a balance between updating to the latest version of their dependencies and retaining compatibility without breaking their code.

Indeed, updating dependencies exposes clients to the risk of Breaking Changes (BCs) coming from their dependencies.

While some work already exists on some types of compatibility issues, others remain unexplored.

We propose, based on the existence of such frontiers, new methods to help libraries understand their uses, and clients to understand the behavioral changes they face.

**Keywords:** Libraries, Software evolution, Software ecosystems

Les bibliothèques logicielles sont largement utilisées dans le développement logiciel moderne par les développeurs afin d'intégrer dans une application des fonctionnalités déjà implémentées par d'autres développeurs.

Une telle dépendance à du code tiers pose des problèmes de maintenance, de compatibilité et d'évolution, tant pour les développeurs de clients que pour ceux de bibliothèques.

Dans cette thèse, nous analysons la relation et les frontières qui existent entre les bibliothèques et les clients.

Une particularité des applications utilisant des bibliothèques est la présence et la consommation d'interfaces de programmation d'applications (APIs).

Ces APIs constituent la principale frontière d'interaction que le développeur utilise pour consommer une fonctionnalité spécifique d'une bibliothèque.

Cette frontière permet le transfert de données dans les deux sens, du client vers la bibliothèque, et de la bibliothèque vers le client.

Les développeurs de bibliothèques manquent souvent de connaissances sur les usages réels de leurs bibliothèques, ce qui empêche une évolution correcte et sûre.

À l'inverse, les développeurs de clients rencontrent souvent des difficultés à trouver un équilibre entre la mise à jour vers la dernière version de leurs dépendances et le maintien de la compatibilité sans casser leur code.

En effet, la mise à jour des dépendances expose les clients au risque de changements cassants (BCs) provenant de leurs dépendances.

Bien que certains travaux existent déjà sur certains types de problèmes de compatibilité, d'autres restent inexplorés.

Nous proposons, en nous appuyant sur l'existence de telles frontières, de nouvelles méthodes pour aider les bibliothèques à comprendre leurs usages et les clients à comprendre les changements de comportement auxquels ils sont confrontés.

**Mots-clés :** Bibliothèques logicielles, Évolution logicielle, Écosystèmes logiciels