

# Chapitre 1

## Introduction

### Sommaire

1.1	Contexte . . . . .	1
1.2	Questions de recherche et contributions . . . . .	6
1.2.1	Ré-identification sans coordination synchrone pour les CRDTs pour le type Séquence . . . . .	6
1.2.2	Éditeur de texte collaboratif P2P temps réel chiffré de bout en bout . . . . .	7
1.3	Plan du manuscrit . . . . .	8
1.4	Publications . . . . .	9

### 1.1 Contexte

- L'évolution des technologies du web a conduit à l'avènement de ce qui est communément appelé le Web 2.0. La principale caractéristique de ce média est la possibilité aux utilisateur-rices non plus seulement de le consulter, mais aussi d'y contribuer.
- Ces nouvelles fonctionnalités ont permis l'apparition d'applications incitant les utilisateur-rices à créer et partager leur propre contenu, ainsi que d'échanger avec d'autres utilisateur-rices à ce sujet. [1] définit ce type d'applications, c.-à-d. les *réseaux sociaux*, de la manière suivante :

**Définition 1.** Un réseau social est une application respectant les critères suivants :

- (i) Elle prend la forme d'une application interactive.
- (ii) Elle permet à ses utilisateur-rices de produire et de partager leur propre contenu. Ce contenu généré par les utilisateur-rices constitue le contenu principal de l'application.
- (iii) Elle permet à ses utilisateur-rices de posséder et de maintenir leur propre profil sur la plateforme.
- (iv) Elle encourage ses utilisateur-rices à étendre leur réseau social en se connectant à d'autres utilisateur-rices, voire en créant ou en s'intégrant à des communautés.

De nos jours, les réseaux sociaux représentent les applications les plus populaires du paysage internet, e.g. Facebook compte 2,9 milliards d'utilisateur-rices par mois [2], Youtube 2,5 milliards [2], Wikipedia 788 millions [3] ou encore Quora 300 millions [2].

- La démocratisation de ces applications présente plusieurs bienfaits. Notamment, nous notons que les réseaux sociaux permettent, en diminuant le coût individuel pour tout à chacun-e de contribuer, d'améliorer la diffusion de l'information [4, 5]. Aussi, ils contribuent à la création de communautés [6]. Finalement, en permettant à chacun-e de partager son savoir, ils permettent la création de bases de connaissances complètes [7, 8].
- Cependant, la conception de réseaux sociaux fait face à de nombreux défis. Notamment, ces applications doivent assurer leur haute disponibilité, tolérance aux pannes et capacité de passage à l'échelle en raison de leur popularité et importance dans notre quotidien.

**Définition 2** (Disponibilité). La disponibilité d'un système indique sa capacité à répondre à tout moment à une requête d'un-e utilisateur-rice.

**Définition 3** (Tolérance aux pannes). La tolérance aux pannes d'un système indique sa capacité à continuer à répondre aux requêtes malgré l'absence de réponse d'un ou plusieurs de ses composants.

**Définition 4** (Capacité de passage à l'échelle). La capacité de passage à l'échelle d'un système indique sa capacité à traiter un volume toujours plus conséquent de requêtes.

Pour cela, ces systèmes adoptent une architecture décentralisée<sup>1</sup>, c.-à-d. une architecture reposant sur un ensemble de serveurs qui se répartissent la charge de travail et les tâches. Malgré ce que le nom de cette architecture peut suggérer, il convient de noter que de manière globale les serveurs jouent toujours un rôle central dans les systèmes décentralisés. Par exemple, les serveurs servent à authentifier les utilisateur-rices, à stocker leurs données ou encore à assurer la communication entre utilisateur-rices.

- Additionnellement, il convient de préciser que ces serveurs ne sont pas une ressource libre. En effet, ils sont mis à disposition et maintenus par la ou les organisations qui proposent le réseau social. Ces organisations font alors office d'*autorités centrales* du système, e.g. en se portant garantes de l'identité des utilisateur-rices ou encore de l'authenticité d'un contenu.
- De part le rôle que jouent les serveurs dans les systèmes décentralisés, ces derniers échouent à assurer un second ensemble de propriétés, que nous jugeons néanmoins fondamentales :

**Définition 5** (Confidentialité des données). La confidentialité des données d'un système indique sa capacité à garantir à ses utilisateur-rices que leurs données ne seront pas accessibles par des tiers non autorisés ou par le système lui-même.

---

1. Nous utilisons la classification présentée dans [9] pour distinguer les architectures systèmes centralisées, décentralisées et distribuées.

**Définition 6** (Souveraineté des données). La souveraineté des données d'un système indique sa capacité à garantir à ses utilisateur-rices leur maîtrise de leurs données, c.-à-d. leur capacité à les consulter, modifier, partager, exporter ; supprimer ou encore à décider de l'usage qui en est fait.

**Définition 7** (Pérennité). La pérennité d'un système indique sa capacité à garantir à ses utilisateur-rices son fonctionnement continu dans le temps.

**Définition 8** (Résistance à la censure). La résistance à la censure d'un système indique sa capacité à garantir à ses utilisateur-rices son fonctionnement malgré des actions de contrôle de l'information par des autorités.

Ainsi, les utilisateur-rices des réseaux sociaux prennent, de manière consciente ou non, le risque que ces propriétés soient transgressées par les autorités auxquelles appartiennent ces applications ou par des tiers avec lesquelles ces autorités interagissent, e.g. des gouvernements.

- Qui plus est, il convient de noter que les autorités auxquelles appartiennent ces applications, e.g. des entreprises, possèdent leurs propres intérêts, e.g. leur propre profit. De nombreux faits d'actualité ont malheureusement montré que ces autorités ont tendance à privilégier leurs intérêts, quitte à prendre des décisions s'avérant nocives pour une partie de leurs utilisateur-rices. Nous pouvons par exemple noter la mise en avant de contenu à caractère raciste [10], la non-modération de contenu misogyne [11], l'encouragement à la production de contenu toxique [12] ou l'inaction et même entrave à la correction des effets délétères de son réseau social sur la santé mentale de ses utilisatrices [13].
- Ainsi, la présence d'autorités centrales dans les réseaux sociaux représente un danger pour les utilisateur-rices. Il nous paraît alors fondamental de proposer des moyens technologiques pour concevoir et déployer des réseaux sociaux alternatifs qui minimisent le rôle des autorités centrales, voire l'éliminent.
- Dans cette optique, une piste de recherche que nous jugeons intéressante à étudier est celle présentée dans [14]. Dans ce travail, les auteurs proposent un nouveau paradigme de conception d'applications, nommées Local-First Softwares (LFS). Ce type d'applications se démarque de ceux existants, e.g. les applications basées sur le cloud, par la place centrale donnée aux utilisateur-rices et leurs propres appareils, les éventuels serveurs étant relegués qu'à de simples rôles de support.
- [14] identifie 7 propriétés que doivent satisfaire les applications LFS, illustrées dans la Figure 1.1. Ces propriétés se recoupent en grande partie avec celles que nous

	1. Fast	2. Multi-device	3. Offline	4. Collaboration	5. Longevity	6. Privacy	7. User control
???	✓	✓	✓	✓	✓	✓	✓

FIGURE 1.1 – Liste des propriétés visées par les LFS<sup>2</sup>

avons identifiées précédemment :

- (i) Le fonctionnement en mode hors-ligne et le fonctionnement avec une latence minimale sont tous deux assurés en privilégiant la disponibilité [15] (Définition 2).
- (ii) Le respect de la vie privée des utilisateur-rices correspond à la propriété de confidentialité (Définition 5).
- (iii) Le contrôle des utilisateur-rices sur leurs données correspond à la propriété de souveraineté (Définition 6).
- (iv) La longévité de l'application correspond aux propriétés de pérennité (Définition 7) et de résistance à la censure (Définition 8).

Notons qu'il découle de ces propriétés que les applications LFS sont fondamentalement P2P.

- De manière similaire, observons que ce paradigme met en lumière une dernière propriété des applications LFS :

**Définition 9** (Collaborativité). La collaborativité d'un système indique sa capacité à supporter ses utilisateur-rices dans leurs processus de collaboration pour la réalisation de tâches.

Nous précisons que nous considérons dans ce manuscrit qu'une collaboration peut prendre bien des formes. Une collaboration peut ainsi prendre la forme d'un échange entre utilisateur-rices, e.g. un fil de discussion sur une plateforme de questions et réponses, ou d'une édition collaborative d'un contenu, e.g. la rédaction d'une page de wiki.

- Ainsi, [14] établit un paradigme de conception d'applications correspondant à notre vision. Cependant, de nombreuses problématiques de recherche identifiées dans ce travail sont encore non résolues et entravent la démocratisation des applications LFS.
- Notamment, les applications LFS se doivent de répliquer les données entre appareils pour permettre :
  - (i) Le fonctionnement en mode hors-ligne et le fonctionnement avec une faible latence.
  - (ii) Le partage de contenu entre appareils d'un-e même utilisateur-ric.e.
  - (iii) Le partage de contenu entre utilisateur-rices pour la collaboration.
- Cependant, compte tenu des propriétés visées par les applications LFS, plusieurs contraintes restreignent le choix des méthodes de réplication possibles. Ainsi, pour permettre le fonctionnement en mode hors-ligne de l'application, c.-à-d. la consultation et la modification de contenu, les applications LFS doivent relaxer la propriété de cohérence des données.

---

2. Source : <https://www.inkandswitch.com/local-first/#towards-a-better-future>

**Définition 10** (Cohérence). La cohérence d'un système indique sa capacité à présenter une vue uniforme de son état à chacun de ses utilisateur-rices à un moment donné.

Les applications LFS ne peuvent donc pas reposer sur des méthodes de réplication dites pessimistes, c.-à-d. qui empêchent toutes modifications concurrentes d'une même donnée.

- Les applications LFS doivent donc adopter des méthodes de réplication dites optimistes [16]. Ces méthodes autorisent chaque noeud possédant une copie de la donnée de la consulter et de la modifier sans coordination au préalable avec les autres noeuds. L'état des copies des noeuds peut donc diverger temporairement. Un mécanisme de synchronisation permet ensuite aux noeuds de partager les modifications effectuées et de les intégrer de façon à converger à terme [17], c.-à-d. obtenir à terme de nouveau des états équivalents.
- Cependant, il convient de noter que les méthodes de réplication optimistes autorisent la génération en concurrence de modifications provoquant un conflit, e.g. la modification et la suppression d'une même page dans un wiki. Un mécanisme de résolution de conflits est alors nécessaire pour assurer la convergence à terme des noeuds.
- De nouveau, le modèle du système des applications LFS limitent les choix possibles concernant les mécanismes de résolution de conflits. Notamment, les applications LFS ne disposent d'aucun contrôle sur le nombre de noeuds qui compose le système, c.-à-d. le nombre d'appareils utilisés par l'ensemble de leurs utilisateur-rices. Le nombre de noeuds peut donc croître de manière non-bornée. La complexité algorithmique des mécanismes de résolution de conflits doit donc être indépendante de ce paramètre, ou alors en être fonction uniquement de manière logarithmique.
- De plus, ces noeuds n'offrent aucune garantie sur leur stabilité. Des noeuds peuvent donc rejoindre et participer au système, mais uniquement de manière éphémère. Ce phénomène est connu sous le nom de *churn* [18]. Ainsi, de part l'absence de garantie sur le nombre de noeuds connectés de manière stable, les applications LFS ne peuvent pas utiliser des mécanismes de résolution de conflits reposant sur une coordination synchrone d'une proportion des noeuds du système, c.-à-d. sur des algorithmes de consensus [19, 20].
- Ainsi, pour permettre la conception d'applications LFS, il convient de disposer de mécanismes de résolution de conflits pour l'ensemble des types de données avec une complexité algorithmique efficace par rapport au nombre de noeuds et ne nécessitant pas de coordination synchrone entre une proportion des noeuds du système.

## 1.2 Questions de recherche et contributions

### 1.2.1 Ré-identification sans coordination synchrone pour les CRDTs pour le type Séquence

- Les Conflict-free Replicated Data Types (CRDTs) [21, 22] sont des types de données répliqués. Ils sont conçus pour permettre à un ensemble de noeuds d'un système de répliquer une donnée et pour leur permettre de la consulter et de la modifier sans aucune coordination préalable. Dans ce but, les CRDTs incorporent des mécanismes de résolution de conflits automatiques directement au sein leur spécification.
- Cependant, ces mécanismes induisent un surcoût, aussi bien en termes de métadonnées et de calculs que de bande-passante. Ces surcoûts sont néanmoins jugés acceptables par la communauté pour une variété de types de données, e.g. le Registre ou l'Ensemble. Cependant, le surcoût des CRDTs pour le type Séquence constitue toujours une problématique de recherche.
- En effet, la particularité des CRDTs pour le type Séquence est que leur surcoût croît de manière monotone au cours de la durée de vie de la donnée, c.-à-d. au fur et à mesure des modifications effectuées. Le surcoût introduit par les CRDTs pour ce type de données se révèle donc handicapant dans le contexte de collaborations sur de longues durées ou à large échelle.
- De manière plus précise, le surcoût des CRDTs pour le type Séquence provient de la croissance des métadonnées utilisées par leur mécanisme de résolution de conflits automatique. Ces métadonnées correspondent à des identifiants qui sont associés aux éléments de la Séquence. Ces identifiants permettent de résoudre les conflits, e.g. en précisant quel est l'élément à supprimer ou en spécifiant la position d'un nouvel élément à insérer par rapport aux autres.
- Plusieurs approches ont été proposées pour réduire le coût induit par ces identifiants. Notamment, [23, 24] proposent un mécanisme de ré-assignation des identifiants pour réduire leur coût a posteriori. Ce mécanisme génère toutefois des conflits en cas de modifications concurrentes de la séquence, c.-à-d. l'insertion ou la suppression d'un élément. Les auteurs résolvent ce problème en proposant un mécanisme de transformation des modifications concurrentes par rapport à l'effet du mécanisme de ré-assignation des identifiants.
- Cependant, l'exécution en concurrence du mécanisme de ré-assignation des identifiants par plusieurs noeuds provoque elle-même un conflit. Pour éviter ce dernier type de conflit, les auteurs choisissent de subordonner à un algorithme de consensus l'exécution du mécanisme de ré-assignation des identifiants. Ainsi, le mécanisme de ré-assignation des identifiants ne peut être déclenché en concurrence par plusieurs noeuds du système.
- Comme nous l'avons évoqué précédemment, reposer sur un algorithme de consensus qui requiert une coordination synchrone entre une proportion de noeuds du système est une contrainte incompatible avec les systèmes P2P à large échelle sujets au churn. Notre problématique de recherche est donc la suivante : *pouvons-nous proposer un*

*mécanisme sans coordination synchrone de réduction du surcoût des CRDTs pour Séquence, c.-à-d. adapté aux applications LFS ?*

- Pour répondre à cette problématique, nous proposons RenamableLogootSplit, un nouveau CRDT pour le type Séquence. Ce CRDT intègre un mécanisme de ré-assignation des identifiants, dit de renommage, directement au sein de sa spécification. Nous associons au mécanisme de renommage un mécanisme de résolution de conflits automatique additionnel pour gérer ses exécutions concurrentes. Ainsi, nous proposons un CRDT pour le type Séquence dont le surcoût est périodiquement réduit par le biais d'un mécanisme n'introduisant aucune contrainte de coordination synchrone entre les noeuds du système.

### 1.2.2 Éditeur de texte collaboratif P2P temps réel chiffré de bout en bout

- Les systèmes collaboratifs permettent à plusieurs utilisateur-rices de collaborer pour la réalisation d'une tâche. Les systèmes collaboratifs actuels adoptent principalement une architecture décentralisée, c.-à-d. un ensemble de serveurs avec lesquels les utilisateur-rices interagissent pour réaliser leur tâche, e.g. Google Docs [25]. Par rapport à une architecture centralisée, cette architecture leur permet d'améliorer leur disponibilité et tolérance aux pannes, notamment grâce aux méthodes de réplication de données. Cette architecture à base de serveurs facilite aussi la collaboration, les serveurs permettant d'intégrer les modifications effectuées par les utilisateur-rices, de stocker les données, d'assurer la communication entre les utilisateur-rices ou encore de les authentifier.
- De part le rôle qui leur incombe, ces serveurs occupent une place primordiale dans ces systèmes. Il en découle plusieurs problématiques :
  - (i) Ces serveurs manipulent et hébergent les données faisant l'objet de collaborations. Ces systèmes ont donc connaissance des données manipulées et de l'identité des auteur-rices des modifications. Les systèmes collaboratifs décentralisés demandent donc à leurs utilisateur-rices d'abandonner la souveraineté et la confidentialité de leur travail.
  - (ii) Ces serveurs sont gérés par des autorités centrales, e.g. Google. Les systèmes collaboratifs devenant non-fonctionnels en cas d'arrêt de leurs serveurs, les utilisateur-rices de ces systèmes dépendent de ces autorités centrales. Ainsi, de part leur pouvoir de vie et de mort sur les services qu'elles proposent, les autorités centrales représentent une menace pour la pérennité de ces systèmes, e.g. [26].
- Pour répondre à ces problématiques, c.-à-d. confidentialité et souveraineté des données, dépendance envers des tiers, pérennité des systèmes, un nouveau paradigme de conception d'applications propose de concevoir des applications LFS, c.-à-d. des applications mettant les utilisateur-rices et leurs appareils au coeur du système. Dans ce cadre d'applications P2P, les serveurs sont relégués seulement à un rôle de support à la collaboration.

- Dans le cadre de ses travaux, notre équipe de recherche étudie notamment la conception d'applications respectant ce paradigme. Ce changement de modèle, d'une architecture décentralisée appartenant à des autorités centrales à une architecture P2P sans autorités centrales, introduit un ensemble de problématiques de domaines variés, e.g.
  - (i) Comment permettre aux utilisateur-rices de collaborer en l'absence d'autorités centrales pour résoudre les conflits de modifications ?
  - (ii) Comment authentifier les utilisateur-rices en l'absence d'autorités centrales ?
  - (iii) Comment structurer le réseau de manière efficace, c.-à-d. en limitant le nombre de connexions par pair ?
- Cet ensemble de questions peut être résumé en la problématique suivante : *pouvons-nous concevoir une application collaborative P2P à large échelle, sûre et sans autorités centrales ?*
- Pour étudier cette problématique, l'équipe Coast développe l'application Multi User Text Editor (MUTE)<sup>3</sup> [27]. Il s'agit d'un éditeur de texte web collaboratif P2P temps réel chiffré de bout en bout. Ce projet nous permet de présenter les travaux de recherche de l'équipe portant sur les mécanismes de résolutions de conflits automatiques pour le type Séquence [28, 29, 30] et les mécanismes d'authentification des pairs dans les systèmes sans autorités centrales [31, 32]. Puis, ce projet nous donne l'opportunité d'étudier la littérature des nombreux domaines de recherche nécessaires à la conception d'un tel système, c.-à-d. le domaine des protocoles d'appartenance aux groupes [33, 34], des topologies réseaux P2P [35] ou encore des protocoles d'établissement de clés de chiffrement de groupe [36]. Ce projet nous permet ainsi de valoriser nos travaux et d'identifier de nouvelles perspectives de recherche. Finalement, il résulte de ce projet le Proof of Concept (PoC) le plus complet d'applications LFS, à notre connaissance. *Matthieu: TODO : Vérifier du côté des applis de IPFS*

## 1.3 Plan du manuscrit

- Ce manuscrit de thèse est organisé de la manière suivante :
- Dans le ??, nous introduisons le modèle du système que nous considérons, c.-à-d. les systèmes P2P à large échelle sujets au churn et sans autorités centrales. Puis nous présentons dans ce chapitre l'état de l'art des mécanismes de résolution de conflits automatiques utilisés dans les systèmes adoptant le paradigme de la réplique optimiste. À partir de cet état de l'art, nous identifions et motivons notre problématique de recherche, c.-à-d. l'absence de mécanisme adapté aux systèmes P2P à large échelle sujets au churn permettant de réduire le surcoût induit par les mécanismes de résolution de conflits automatiques pour le type Séquence.
- Dans le ??, nous présentons notre approche pour présenter un tel mécanisme, c.-à-d. un mécanisme de résolution de conflits automatiques pour le type Séquence auquel

---

3. Disponible à l'adresse : <https://mutehost.loria.fr>



nous associons un mécanisme de Garbage Collection (GC) de son surcoût ne nécessitant pas de coordination synchrone entre les noeuds du système. Nous détaillons le fonctionnement de notre approche, sa validation par le biais d’une évaluation empirique puis comparons notre approche par rapport aux approches existantes. Finalement, nous concluons la présentation de notre approche en identifiant et en détaillant plusieurs de ses limites.

- Dans le ??, nous présentons MUTE, l’éditeur de texte collaboratif temps réel P2P chiffré de bout en bout que notre équipe de recherche développe dans le cadre de ses travaux de recherche. Nous présentons les différentes couches logicielles formant un pair et les services tiers avec lesquels les pairs interagissent, et détaillons nos travaux dans le cadre de ce projet, c.-à-d. l’intégration de notre mécanisme de résolution de conflits automatiques pour le type Séquence et le développement de la couche de livraison des messages associée. Pour chaque couche logicielle, nous identifions ses limites et présentons de potentielles pistes d’améliorations.
- Finalement, nous récapitulons dans le chapitre 2 les contributions réalisées dans le cadre de cette thèse. Puis nous clotûrons ce manuscrit en introduisant plusieurs des pistes de recherches que nous souhaiterons explorer dans le cadre de nos travaux futurs.

## 1.4 Publications

Notre travail sur la problématique identifiée dans la sous-section 1.2.1, c.-à-d. la proposition d’un mécanisme ne nécessitant aucune coordination synchrone pour réduire le surcoût des CRDTs pour le type Séquence, a donné lieu à des publications à différents stades de son avancement :

- (i) Dans [37], nous motivons le problème identifié et présentons l’idée de notre approche pour y répondre.
- (ii) Dans [38], nous détaillons une première partie de notre approche et présentons notre protocole d’évaluation expérimentale ainsi que ses premiers résultats.
- (iii) Dans [30], nous détaillons notre proposition dans son entièreté. Nous accompagnons cette proposition d’une évaluation expérimentale poussée. Finalement, nous complétons notre travail d’une discussion identifiant plusieurs de ses limites et présentant des pistes de travail possibles pour y répondre.

Nous précisons ci-dessous les informations relatives à chacun de ces articles.

### Efficient renaming in CRDTs [37]

**Auteur** Matthieu Nicolas

**Article de position** à Middleware 2018 - 19th ACM/IFIP International Middleware Conference (Doctoral Symposium), Dec 2018, Rennes, France.

**Abstract** *Sequence Conflict-free Replicated Data Types (CRDTs)* allow to replicate and edit, without any kind of coordination, sequences in distributed systems. To ensure convergence, existing works from the literature add metadata to each element but they do not bound its footprint, which impedes their adoption. Several approaches were proposed to address this issue but they do not fit a fully distributed setting. In this paper, we present our ongoing work on the design and validation of a fully distributed renaming mechanism, setting a bound to the metadata’s footprint. Addressing this issue opens new perspectives of adoption of these CRDTs in distributed applications.

## Efficient Renaming in Sequence CRDTs [38]

**Auteurs** Matthieu Nicolas, G  rald Oster, Olivier Perrin

**Article de workshop**    PaPoC 2020 - 7th Workshop on Principles and Practice of Consistency for Distributed Data, Apr 2020, Heraklion / Virtual, Greece.

**Abstract** To achieve high availability, large-scale distributed systems have to replicate data and to minimise coordination between nodes. Literature and industry increasingly adopt Conflict-free Replicated Data Types (CRDTs) to design such systems. CRDTs are data types which behave as traditional ones, e.g. the Set or the Sequence. However, unlike traditional data types, they are designed to natively support concurrent modifications. To this end, they embed in their specification a conflict-resolution mechanism.

To resolve conflicts in a deterministic manner, CRDTs usually attach identifiers to elements stored in the data structure. Identifiers have to comply with several constraints, such as uniqueness or belonging to a dense order. These constraints may hinder the identifiers’ size from being bounded. As the system progresses, identifiers tend to grow. This inflation deepens the overhead of the CRDT over time, leading to performance issues.

To address this issue, we propose a new CRDT for Sequence which embeds a renaming mechanism. It enables nodes to reassign shorter identifiers to elements in an uncoordinated manner. Experimental results demonstrate that this mechanism decreases the overhead of the replicated data structure and eventually limits it.

## Efficient Renaming in Sequence CRDTs [30]

**Auteurs** Matthieu Nicolas, G  rald Oster, Olivier Perrin

**Article de journal** dans IEEE Transactions on Parallel and Distributed Systems, Institute of Electrical and Electronics Engineers, 2022, 33 (12), pp.3870-3885.

**Abstract** To achieve high availability, large-scale distributed systems have to replicate data and to minimise coordination between nodes. For these purposes, literature and industry increasingly adopt Conflict-free Replicated Data Types (CRDTs) to design such systems. CRDTs are new specifications of existing data types, e.g. Set or Sequence. While CRDTs have the same behaviour as previous specifications in sequential executions,

they actually shine in distributed settings as they natively support concurrent updates. To this end, CRDTs embed in their specification conflict resolution mechanisms. These mechanisms usually rely on identifiers attached to elements of the data structure to resolve conflicts in a deterministic and coordination-free manner. Identifiers have to comply with several constraints, such as being unique or belonging to a dense total order. These constraints may hinder the identifier size from being bounded. Identifiers hence tend to grow as the system progresses, which increases the overhead of CRDTs over time and leads to performance issues. To address this issue, we propose a novel Sequence CRDT which embeds a renaming mechanism. It enables nodes to reassign shorter identifiers to elements in an uncoordinated manner. Experimental results demonstrate that this mechanism decreases the overhead of the replicated data structure and eventually minimises it.

# Bibliographie

- [1] Jonathan A OBAR et Steven S WILDMAN. « Social Media Definition and the Governance Challenge - An Introduction to the Special Issue ». In : *Obar, JA and Wildman, S.(2015). Social media definition and the governance challenge : An introduction to the special issue. Telecommunications policy* 39.9 (2015), p. 745–750.
- [2] STATISTA. *Biggest social media platforms 2022*. Last Accessed : 2022-10-06. URL : <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>.
- [3] WIKIMEDIA. *Wikimedia Statistics - English Wikipedia*. Last Accessed : 2022-10-06. URL : <https://stats.wikimedia.org/#/en.wikipedia.org>.
- [4] David MEEK. « YouTube and Social Movements : A Phenomenological Analysis of Participation, Events and Cyberplace ». In : *Antipode* 44.4 (2012), p. 1429–1448. DOI : <https://doi.org/10.1111/j.1467-8330.2011.00942.x>. eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8330.2011.00942.x>. URL : <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8330.2011.00942.x>.
- [5] Yannis THEOCHARIS. « The wealth of (occupation) networks? Communication patterns and information distribution in a Twitter protest network ». In : *Journal of Information Technology & Politics* 10.1 (2013), p. 35–56.
- [6] José VAN DIJCK et Thomas POELL. « Understanding social media logic ». In : *Media and communication* 1.1 (2013), p. 2–14.
- [7] Jim GILES. « Special Report Internet encyclopaedias go head to head ». In : *nature* 438.15 (2005), p. 900–901.
- [8] Lada A ADAMIC, Jun ZHANG, Eytan BAKSHY et Mark S ACKERMAN. « Knowledge sharing and yahoo answers : everyone knows something ». In : *Proceedings of the 17th international conference on World Wide Web*. 2008, p. 665–674.
- [9] Paul BARAN. « On distributed communications networks ». In : *IEEE transactions on Communications Systems* 12.1 (1964), p. 1–9.
- [10] Safiya Umoja NOBLE. *Algorithms of Oppression : How Search Engines Reinforce Racism*. NYU Press, 2018. ISBN : 9781479849949.
- [11] Amnesty INTERNATIONAL. *#Toxictwitter : Violence and abuse against women online*. Last Accessed : 2022-10-07. URL : <https://www.amnesty.org/en/documents/act30/8070/2018/en/>.

- [12] Wall Street JOURNAL. *Facebook Tried to Make Its Platform a Healthier Place. It Got Angrier Instead*. Last Accessed : 2022-10-07. URL : <https://t.co/P6JohMdhQE>.
- [13] Wall Street JOURNAL. *Facebook Knows Instagram Is Toxic for Teen Girls, Company Documents Show*. Last Accessed : 2022-10-07. URL : <https://t.co/JAvzKFc61q>.
- [14] Martin KLEPPMANN, Adam WIGGINS, Peter van HARDENBERG et Mark MCGRANAGHAN. « Local-First Software : You Own Your Data, in Spite of the Cloud ». In : *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*. Onward! 2019. Athens, Greece : Association for Computing Machinery, 2019, p. 154–178. ISBN : 9781450369954. DOI : 10.1145/3359591.3359737. URL : <https://doi.org/10.1145/3359591.3359737>.
- [15] Daniel ABADI. « Consistency Tradeoffs in Modern Distributed Database System Design : CAP is Only Part of the Story ». In : *Computer* 45.2 (2012), p. 37–42. DOI : 10.1109/MC.2012.33.
- [16] Yasushi SAITO et Marc SHAPIRO. « Optimistic Replication ». In : *ACM Comput. Surv.* 37.1 (mar. 2005), p. 42–81. ISSN : 0360-0300. DOI : 10.1145/1057977.1057980. URL : <https://doi.org/10.1145/1057977.1057980>.
- [17] Douglas B TERRY, Marvin M THEIMER, Karin PETERSEN, Alan J DEMERS, Mike J SPREITZER et Carl H HAUSER. « Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System ». In : *SIGOPS Oper. Syst. Rev.* 29.5 (déc. 1995), p. 172–182. ISSN : 0163-5980. DOI : 10.1145/224057.224070. URL : <https://doi.org/10.1145/224057.224070>.
- [18] Daniel STUTZBACH et Reza REJAIE. « Understanding Churn in Peer-to-Peer Networks ». In : *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*. IMC '06. Rio de Janeiro, Brazil : Association for Computing Machinery, 2006, p. 189–202. ISBN : 1595935614. DOI : 10.1145/1177080.1177105. URL : <https://doi.org/10.1145/1177080.1177105>.
- [19] Leslie LAMPORT. « The part-time parliament ». In : *Concurrency : the Works of Leslie Lamport*. 2019, p. 277–317.
- [20] Diego ONGARO et John OUSTERHOUT. « In search of an understandable consensus algorithm ». In : *2014 USENIX Annual Technical Conference (Usenix ATC 14)*. 2014, p. 305–319.
- [21] Marc SHAPIRO et Nuno PREGUIÇA. *Designing a commutative replicated data type*. Research Report RR-6320. INRIA, 2007. URL : <https://hal.inria.fr/inria-00177693>.
- [22] Marc SHAPIRO, Nuno M. PREGUIÇA, Carlos BAQUERO et Marek ZAWIRSKI. « Conflict-Free Replicated Data Types ». In : *Proceedings of the 13th International Symposium on Stabilization, Safety, and Security of Distributed Systems*. SSS 2011. 2011, p. 386–400. DOI : 10.1007/978-3-642-24550-3\_29.

- 
- [23] Mihai LETIA, Nuno PREGUIÇA et Marc SHAPIRO. « Consistency without concurrency control in large, dynamic systems ». In : *LADIS 2009 - 3rd ACM SIGOPS International Workshop on Large Scale Distributed Systems and Middleware*. T. 44. Operating Systems Review 2. Big Sky, MT, United States : Assoc. for Computing Machinery, oct. 2009, p. 29–34. DOI : 10.1145/1773912.1773921. URL : <https://hal.inria.fr/hal-01248270>.
  - [24] Marek ZAWIRSKI, Marc SHAPIRO et Nuno PREGUIÇA. « Asynchronous rebalancing of a replicated tree ». In : *Conférence Française en Systèmes d'Exploitation (CFSE)*. Saint-Malo, France, mai 2011, p. 12. URL : <https://hal.inria.fr/hal-01248197>.
  - [25] GOOGLE. *Google Docs*. Last Accessed : 2022-10-07. URL : <https://docs.google.com/>.
  - [26] Cody ODGEN. *Google Graveyard*. Last Accessed : 2022-10-11. URL : <https://killedbygoogle.com/>.
  - [27] Matthieu NICOLAS, Victorien ELVINGER, Gérald OSTER, Claudia-Lavinia IGNAT et François CHAROY. « MUTE : A Peer-to-Peer Web-based Real-time Collaborative Editor ». In : *ECSCW 2017 - 15th European Conference on Computer-Supported Cooperative Work*. T. 1. Proceedings of 15th European Conference on Computer-Supported Cooperative Work - Panels, Posters and Demos 3. Sheffield, United Kingdom : EUSSET, août 2017, p. 1–4. DOI : 10.18420/ecscw2017\\_p5. URL : <https://hal.inria.fr/hal-01655438>.
  - [28] Luc ANDRÉ, Stéphane MARTIN, Gérald OSTER et Claudia-Lavinia IGNAT. « Supporting Adaptable Granularity of Changes for Massive-Scale Collaborative Editing ». In : *International Conference on Collaborative Computing : Networking, Applications and Worksharing - CollaborateCom 2013*. Austin, TX, USA : IEEE Computer Society, oct. 2013, p. 50–59. DOI : 10.4108/icst.collaboratecom.2013.254123.
  - [29] Victorien ELVINGER. « Réplication sécurisée dans les infrastructures pair-à-pair de collaboration ». Theses. Université de Lorraine, juin 2021. URL : <https://hal.univ-lorraine.fr/tel-03284806>.
  - [30] Matthieu NICOLAS, Gerald OSTER et Olivier PERRIN. « Efficient Renaming in Sequence CRDTs ». In : *IEEE Transactions on Parallel and Distributed Systems* 33.12 (déc. 2022), p. 3870–3885. DOI : 10.1109/TPDS.2022.3172570. URL : <https://hal.inria.fr/hal-03772633>.
  - [31] Hoang-Long NGUYEN, Claudia-Lavinia IGNAT et Olivier PERRIN. « Trusternity : Auditing Transparent Log Server with Blockchain ». In : *Companion of the The Web Conference 2018*. Lyon, France, avr. 2018. DOI : 10.1145/3184558.3186938. URL : <https://hal.inria.fr/hal-01883589>.
  - [32] Hoang-Long NGUYEN, Jean-Philippe EISENBARTH, Claudia-Lavinia IGNAT et Olivier PERRIN. « Blockchain-Based Auditing of Transparent Log Servers ». In : *32th IFIP Annual Conference on Data and Applications Security and Privacy (DBSec)*. Sous la dir. de Florian KERSCHBAUM et Stefano PARABOSCHI. T. LNCS-10980.

- Data and Applications Security and Privacy XXXII. Part 1 : Administration. Bergamo, Italy : Springer International Publishing, juil. 2018, p. 21–37. DOI : 10.1007/978-3-319-95729-6\\_2. URL : <https://hal.archives-ouvertes.fr/hal-01917636>.
- [33] Abhinandan DAS, Indranil GUPTA et Ashish MOTIVALA. « SWIM : scalable weakly-consistent infection-style process group membership protocol ». In : *Proceedings International Conference on Dependable Systems and Networks*. 2002, p. 303–312. DOI : 10.1109/DSN.2002.1028914.
- [34] Armon DADGAR, James PHILLIPS et Jon CURREY. « Lifeguard : Local health awareness for more accurate failure detection ». In : *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE. 2018, p. 22–25.
- [35] Brice NÉDELEC, Julian TANKE, Davide FREY, Pascal MOLLI et Achour MOSTÉFAOUI. « An adaptive peer-sampling protocol for building networks of browsers ». In : *World Wide Web* 21.3 (2018), p. 629–661.
- [36] Mike BURMESTER et Yvo DESMEDT. « A secure and efficient conference key distribution system ». In : *Advances in Cryptology — EUROCRYPT’94*. Sous la dir. d’Alfredo DE SANTIS. Berlin, Heidelberg : Springer Berlin Heidelberg, 1995, p. 275–286. ISBN : 978-3-540-44717-7.
- [37] Matthieu NICOLAS. « Efficient renaming in CRDTs ». In : *Middleware 2018 - 19th ACM/IFIP International Middleware Conference (Doctoral Symposium)*. Rennes, France, déc. 2018. URL : <https://hal.inria.fr/hal-01932552>.
- [38] Matthieu NICOLAS, Gérald OSTER et Olivier PERRIN. « Efficient Renaming in Sequence CRDTs ». In : *7th Workshop on Principles and Practice of Consistency for Distributed Data (PaPoC’20)*. Heraklion, Greece, avr. 2020. URL : <https://hal.inria.fr/hal-02526724>.
- [39] Gérald OSTER, Pascal URSO, Pascal MOLLI et Abdessamad IMINE. « Data Consistency for P2P Collaborative Editing ». In : *ACM Conference on Computer-Supported Cooperative Work - CSCW 2006*. Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work. Banff, Alberta, Canada : ACM Press, nov. 2006, p. 259–268. URL : <https://hal.inria.fr/inria-00108523>.
- [40] Hyun-Gul ROH, Myeongjae JEON, Jin-Soo KIM et Joonwon LEE. « Replicated abstract data types : Building blocks for collaborative applications ». In : *Journal of Parallel and Distributed Computing* 71.3 (2011), p. 354–368. ISSN : 0743-7315. DOI : <https://doi.org/10.1016/j.jpdc.2010.12.006>. URL : <http://www.sciencedirect.com/science/article/pii/S0743731510002716>.
- [41] Nuno PREGUICA, Joan Manuel MARQUES, Marc SHAPIRO et Mihai LETIA. « A Commutative Replicated Data Type for Cooperative Editing ». In : *2009 29th IEEE International Conference on Distributed Computing Systems*. Juin 2009, p. 395–403. DOI : 10.1109/ICDCS.2009.20.

- [42] Stéphane WEISS, Pascal URSO et Pascal MOLLI. « Logoot : A Scalable Optimistic Replication Algorithm for Collaborative Editing on P2P Networks ». In : *Proceedings of the 29th International Conference on Distributed Computing Systems - ICDCS 2009*. Montreal, QC, Canada : IEEE Computer Society, juin 2009, p. 404–412. DOI : 10.1109/ICDCS.2009.75. URL : <http://doi.ieeeecomputersociety.org/10.1109/ICDCS.2009.75>.
- [43] Brice NÉDELEC, Pascal MOLLI, Achour MOSTÉFAOUI et Emmanuel DESMONTILS. « LSEQ : an adaptive structure for sequences in distributed collaborative editing ». In : *Proceedings of the 2013 ACM Symposium on Document Engineering*. DocEng 2013. Sept. 2013, p. 37–46. DOI : 10.1145/2494266.2494278.
- [44] Brice NÉDELEC, Pascal MOLLI et Achour MOSTÉFAOUI. « A scalable sequence encoding for collaborative editing ». In : *Concurrency and Computation : Practice and Experience* (), e4108. DOI : 10.1002/cpe.4108. eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.4108>. URL : <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.4108>.
- [45] Carlos BAQUERO, Paulo Sergio ALMEIDA et Ali SHOKER. *Pure Operation-Based Replicated Data Types*. 2017. arXiv : 1710.04469 [cs.DC].
- [46] Victorien ELVINGER, Gérald OSTER et Francois CHAROY. « Prunable Authenticated Log and Authenticable Snapshot in Distributed Collaborative Systems ». In : *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*. 2018, p. 156–165. DOI : 10.1109/CIC.2018.00031.
- [47] Sylvie NOËL et Jean-Marc ROBERT. « Empirical study on collaborative writing : What do co-authors do, use, and like ? ». In : *Computer Supported Cooperative Work (CSCW)* 13.1 (2004), p. 63–89.
- [48] D. S. PARKER, G. J. POPEK, G. RUDISIN, A. STOUGHTON, B. J. WALKER, E. WALTON, J. M. CHOW, D. EDWARDS, S. KISER et C. KLINE. « Detection of Mutual Inconsistency in Distributed Systems ». In : *IEEE Trans. Softw. Eng.* 9.3 (mai 1983), p. 240–247. ISSN : 0098-5589. DOI : 10.1109/TSE.1983.236733. URL : <https://doi.org/10.1109/TSE.1983.236733>.
- [49] OPENRELAY. *OpenRelay*. Last Accessed : 2022-10-07. URL : <https://openrelay.xyz/>.
- [50] Protocol LABS. *IPFS*. Last Accessed : 2022-10-07. URL : <https://ipfs.io/>.
- [51] Friedemann MATTERN et al. *Virtual time and global states of distributed systems*. Univ., Department of Computer Science, 1988.
- [52] Colin FIDGE. « Logical Time in Distributed Computing Systems ». In : *Computer* 24.8 (août 1991), p. 28–33. ISSN : 0018-9162. DOI : 10.1109/2.84874. URL : <https://doi.org/10.1109/2.84874>.



- [53] Ravi PRAKASH, Michel RAYNAL et Mukesh SINGHAL. « An Adaptive Causal Ordering Algorithm Suited to Mobile Computing Environments ». In : *Journal of Parallel and Distributed Computing* 41.2 (1997), p. 190–204. ISSN : 0743-7315. DOI : <https://doi.org/10.1006/jpdc.1996.1300>. URL : <https://www.sciencedirect.com/science/article/pii/S0743731596913003>.
- [54] Martin KLEPPMANN, Victor B. F. GOMES, Dominic P. MULLIGAN et Alastair R. BERESFORD. « Interleaving Anomalies in Collaborative Text Editors ». In : *Proceedings of the 6th Workshop on Principles and Practice of Consistency for Distributed Data*. PaPoC '19. Dresden, Germany : Association for Computing Machinery, 2019. ISBN : 9781450362764. DOI : 10.1145/3301419.3323972. URL : <https://doi.org/10.1145/3301419.3323972>.
- [55] Quang Vinh DANG et Claudia-Lavinia IGNAT. « Quality Assessment of Wikipedia Articles : A Deep Learning Approach by Quang Vinh Dang and Claudia-Lavinia Ignat with Martin Vesely as Coordinator ». In : *SIGWEB Newsl.* Autumn (nov. 2016). ISSN : 1931-1745. DOI : 10.1145/2996442.2996447. URL : <https://doi.org/10.1145/2996442.2996447>.
- [56] Claudia-Lavinia IGNAT, G  rald OSTER, Meagan NEWMAN, Valerie SHALIN et Fran  ois CHAROY. « Studying the Effect of Delay on Group Performance in Collaborative Editing ». In : *Proceedings of 11th International Conference on Cooperative Design, Visualization, and Engineering, CDVE 2014, Springer 2014 Lecture Notes in Computer Science*. Proceedings of 11th International Conference on Cooperative Design, Visualization, and Engineering, CDVE 2014. Seattle, WA, United States, sept. 2014, p. 191–198. DOI : 10.1007/978-3-319-10831-5\_29. URL : <https://hal.archives-ouvertes.fr/hal-01088815>.
- [57] Claudia-Lavinia IGNAT, G  rald OSTER, Olivia FOX, Fran  ois CHAROY et Valerie SHALIN. « How Do User Groups Cope with Delay in Real-Time Collaborative Note Taking ». In : *European Conference on Computer Supported Cooperative Work 2015*. Sous la dir. de Nina BOULUS-RODJE, Gunnar ELLINGSEN, Tone BRATTETEIG, Margunn AANESTAD et Pernille BJORN. Proceedings of the 14th European Conference on Computer Supported Cooperative Work. Oslo, Norway : Springer International Publishing, sept. 2015, p. 223–242. DOI : 10.1007/978-3-319-20499-4\_12. URL : <https://hal.inria.fr/hal-01238831>.
- [58] Paulo S  rgio ALMEIDA, Ali SHOKER et Carlos BAQUERO. « Efficient State-Based CRDTs by Delta-Mutation ». In : *Networked Systems*. Sous la dir. d'Ahmed BOUAJJANI et Hugues FAUCONNIER. Cham : Springer International Publishing, 2015, p. 62–76. ISBN : 978-3-319-26850-7.
- [59] Nuno M. PREGUI  A, Carlos BAQUERO et Marc SHAPIRO. « Conflict-free Replicated Data Types (CRDTs) ». In : *CoRR* abs/1805.06358 (2018). arXiv : 1805.06358. URL : <http://arxiv.org/abs/1805.06358>.
- [60] Vitor ENES, Paulo S  rgio ALMEIDA, Carlos BAQUERO et Jo  o LEIT  O. « Efficient Synchronization of State-Based CRDTs ». In : *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 2019, p. 148–159. DOI : 10.1109/ICDE.2019.00022.

- 
- [61] Leslie LAMPORT, Robert SHOSTAK et Marshall PEASE. « The Byzantine Generals Problem ». In : *Concurrency : The Works of Leslie Lamport*. New York, NY, USA : Association for Computing Machinery, 2019, p. 203–226. ISBN : 9781450372701. URL : <https://doi.org/10.1145/3335772.3335936>.
- [62] Jim BAUWENS et Elisa Gonzalez BOIX. « Flec : A Versatile Programming Framework for Eventually Consistent Systems ». In : *Proceedings of the 7th Workshop on Principles and Practice of Consistency for Distributed Data*. PaPoC '20. Heraklion, Greece : Association for Computing Machinery, 2020. ISBN : 9781450375245. DOI : 10.1145/3380787.3393685. URL : <https://doi.org/10.1145/3380787.3393685>.
- [63] Jim BAUWENS et Elisa Gonzalez BOIX. « Improving the Reactivity of Pure Operation-Based CRDTs ». In : *Proceedings of the 8th Workshop on Principles and Practice of Consistency for Distributed Data*. PaPoC '21. Online, United Kingdom : Association for Computing Machinery, 2021. ISBN : 9781450383387. DOI : 10.1145/3447865.3457968. URL : <https://doi.org/10.1145/3447865.3457968>.