

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. Un cas particulier de ces applications proposent aux utilisateur-rices de travailler ensemble pour la création d'un même contenu, en d'autres termes de collaborer. Nous appelons ces applications des *systèmes collaboratifs* :

Définition 1 (Système collaboratif). Un système collaboratif est un système supportant ses utilisateur-rices dans leurs processus de collaboration pour la réalisation de tâches.

De nos jours, ces systèmes font parties des applications les plus populaires du paysage internet, e.g. GoogleDocs compte XX millions utilisateur-rices, GitHub YY millions, Wikipedia 788 millions [1] ou Quora 300 millions [2]. De leur côté, d'autres plateformes fédèrent leur communautés en organisant ponctuellement des collaborations éphémères et généralement massives, e.g. r/Place [3] ou TwitchPlaysPokemon [4].

En raison de leur popularité, les systèmes collaboratifs doivent assurer plusieurs propriétés pour garantir leur bon fonctionnement et qualité de service : une haute disponibilité, tolérance aux pannes et capacité de passage à l'échelle.

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

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, que nous illustrons par la Figure 1.1.

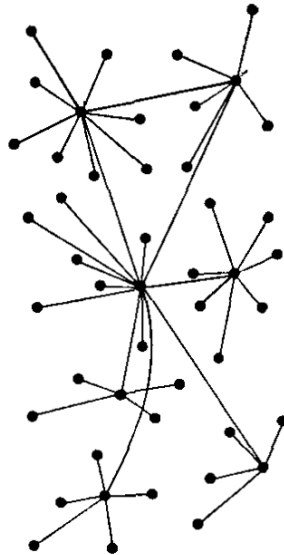


FIGURE 1.1 – Représentation d'une architecture décentralisée¹

Dans ce type d'architecture, les responsabilités, tâches et la charge travail sont réparties entre un ensemble de serveurs. Il convient toutefois de noter que les serveurs jouent de manière globale toujours un rôle central dans ces systèmes, malgré ce que le nom de cette architecture peut suggérer. En effet, ces systèmes reposent toujours sur leurs serveurs pour authentifier les utilisateur-rices, stocker les données de leurs utilisateur-rices ou encore fusionner les modifications effectuées par ces dernier-es.

Bien que cette architecture système permette de répondre aux problèmes d'ordre technique que nous présentons précédemment, elle souffre néanmoins de limites. Notamment, de part le rôle prédominant que jouent les serveurs dans les systèmes décentralisés, ces

1. Source : [5]

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.

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.

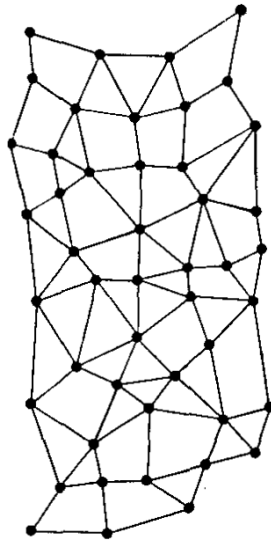
De plus, les serveurs ne sont pas une ressource libre. En effet, ils sont déployés et maintenus par la ou les organisations qui proposent le système collaboratif. Ces organisations font alors office d'*autorités centrales* du système, e.g. en se portant garantes de l'identité des utilisateur-rices, de l'authenticité d'un contenu ou encore de la disponibilité dudit contenu.

De part le fait que les autorités centrales possèdent les serveurs hébergeant le système, elles ont tout pouvoir sur ces derniers. Ainsi, les utilisateur-rices de systèmes collaboratifs prennent, de manière consciente ou non, le risque que les propriétés présentées précédemment soient transgressées par les autorités auxquelles appartiennent ces applications ou par des tiers avec lesquelles ces autorités interagissent, e.g. des gouvernements. Plusieurs faits d'actualités nous ont malheureusement montré de tels faits, e.g. la censure de Wikipedia par des gouvernements [6], la fermeture de services par les entreprises les proposant [7] ou encore la mise à disposition des données hébergées par des applications aux services de renseignement de différentes nations [8, 9]. Cependant, le coût conséquent de l'infrastructure nécessaire pour déployer des systèmes à large échelle équivalents entrave la mise en place d'alternatives, plus respectueuses de leurs utilisateur-rices.

Ainsi, il nous paraît fondamental de proposer des moyens technologiques rendant accessible la conception et le déploiement des systèmes collaboratifs alternatifs. Ces derniers devraient minimiser le rôle des autorités centrales, voire l'éliminer, de façon à protéger et privilégier les intérêts de leurs utilisateur-rices.

Dans cette optique, une piste de recherche que nous jugeons intéressante est celle des systèmes collaboratifs Pair-à-Pair (P2P). Cette architecture système, que nous illustrons par la Figure 1.2, place les utilisateur-rices au centre du système et relègue les éventuels serveurs à un simple rôle de support de la collaboration, e.g. la mise en relation des pairs.

Récemment, la conception de systèmes collaboratifs P2P a gagné en traction suite à [10]. Dans cet article, les auteurs définissent un ensemble de propriétés qui correspondent

FIGURE 1.2 – Représentation d'une architecture distribuée¹

à celles que nous avons établies précédemment, de la Définition 1 à la Définition 8. En utilisant ces propriétés comme critères, les auteurs comparent les fonctionnalités et garanties offertes par les différents types d'applications, notamment les applications lourdes et les applications basées sur le cloud.

Le résultat de cette comparaison est le suivant : alors que les applications basées sur le cloud permettent de nouveaux usages, notamment la collaboration entre utilisateur-rices ou la synchronisation automatique entre appareils, elles retirent à leurs utilisateur-rices toute garantie de pérennité, confidentialité des données et souveraineté des données. Ces dernières propriétés sont pourtant communément offertes par les applications lourdes. La Figure 1.3 détaille ce résultat.

Malgré ce que ce résultat pourrait suggérer, les auteurs affirment que les nouveaux usages offerts par les applications basées sur le cloud ne sont pas antinomiques avec les propriétés de confidentialité, souveraineté, pérennité.

Ainsi, ils proposent un nouveau paradigme de conception d'applications collaboratives P2P, nommées Local-First Softwares (LFS). Ce paradigme vise à la conception d'applications offrant le meilleur des approches existantes, c.-à-d. des applications cochant l'intégralité des critères de la Figure 1.3. Nous partageons cette vision.

Cependant, de nombreuses problématiques de recherche identifiées dans [10] sont encore non résolues et entravent la démocratisation des applications LFS, notamment celles à large échelle. Spécifiquement, les applications LFS se doivent de répliquer les données entre les 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.

Toutefois, compte tenu des propriétés visées par les applications LFS, plusieurs contraintes

Technology	Section	1. Fast (§ 2.1)	2. Multi-device (§ 2.2)	3. Offline (§ 2.3)	4. Collaboration (§ 2.4)	5. Longevity (§ 2.5)	6. Privacy (§ 2.6)	7. User control (§ 2.7)
<i>Applications employed by end users</i>								
Files + email attachments	§ 3.1.1	✓	—	✓	●	✓	—	✓
Google Docs	§ 3.1.2	—	✓	—	✓	—	●	—
Trello	§ 3.1.2	—	✓	—	✓	—	●	●
Pinterest	§ 3.1.2	●	✓	●	✓	●	●	●
Dropbox	§ 3.1.3	✓	—	—	●	✓	—	✓
Git + GitHub	§ 3.1.4	✓	—	✓	—	✓	—	✓
<i>Technologies employed by application developers</i>								
Thin client (web apps)	§ 3.2.1	●	✓	●	✓	●	●	●
Thick client (mobile apps)	§ 3.2.2	✓	—	✓	●	—	●	●
Backend-as-a-service	§ 3.2.3	—	✓	✓	—	●	●	●
CouchDB	§ 3.2.4	—	—	✓	●	—	—	—

FIGURE 1.3 – Évaluation d’applications et de technologies vis-à-vis des 7 propriétés visées par les Local-First Software²

restreignent le choix des méthodes de réplification 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.

Définition 9 (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 doivent donc adopter des méthodes de réplification dites optimistes [11]. Ces méthodes autorisent chaque noeud possédant une copie de la donnée à la consulter et à la modifier sans coordination au préalable avec les autres noeuds³. 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 [12], c.-à-d. obtenir à terme de nouveau des états équivalents.

Cependant, il convient de noter que les méthodes de réplification 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 que nous visons, c.-à-d. des applications LFS à large échelle, limitent les choix possibles concernant les mécanismes de résolution de conflits. Notamment, ces applications ne disposent d’aucun contrôle sur

3. Par opposition aux méthodes de réplification dites pessimistes, qui nécessitent une coordination préalable entre les noeuds avant toute modification de la donnée.

le nombre de noeuds qui compose le système, c.-à-d. le nombre d'appareils utilisés par l'ensemble de leurs utilisateur-rices. Ce nombre de noeuds peut donc croître de manière non-bornée. Les mécanismes de résolution de conflits choisis devraient donc rester efficaces, de manière indépendante à l'évolution de ce paramètre.

De plus, les noeuds composant le système 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* [13]. Ainsi, de part l'absence de garantie sur le nombre de noeuds connectés de manière stable, les applications LFS à large échelle 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 [14, 15].

Ainsi, pour permettre la conception d'applications LFS à large échelle, il convient de disposer de mécanismes de résolution de conflits pour l'ensemble des types de données avec une complexité algorithmique efficace peu importe le 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) [16, 17] 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, de la modifier sans aucune coordination préalable et d'assurer à terme la convergence des copies. 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, [18, 19] 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

Comme évoqué précédemment, la conception d'applications LFS à large échelle présente un ensemble de problématiques issues 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 LFS à large échelle, sûre et sans autorités centrales ?*

Pour étudier cette problématique, l'équipe Coast développe l'application Multi User Text Editor (MUTE)⁴ [20]. Il s'agit d'un Proof of Concept (PoC) d'éditeur de texte web collaboratif P2P temps réel chiffré de bout en bout.

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

Ce projet permet à l'équipe de présenter ses travaux de recherche portant sur les mécanismes de résolutions de conflits automatiques pour le type Séquence [21, 22, 23] et les mécanismes d'authentification des pairs dans les systèmes sans autorités centrales [24, 25].

De plus, en inscrivant ses travaux dans le cadre d'un système complet, ce projet permet à l'équipe d'identifier de nouvelles problématiques en relation avec les nombreux domaines de recherche nécessaires à la conception d'un tel système, c.-à-d. le domaine des protocoles d'appartenance aux groupes [26, 27], des topologies réseaux P2P [28] ou encore des protocoles d'établissement de clés de chiffrement de groupe [29].

Dans le cadre de notre thèse, nous avons contribué au développement de ce projet. Nous avons notamment implémenté plusieurs CRDTs pour le type Séquence [21, 23] et le protocole d'appartenance au réseau SWIM [26].

1.3 Plan du manuscrit

Ce manuscrit de thèse est organisé de la manière suivante :

Dans le chapitre 2, 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 réplication 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 nous associons un mécanisme de 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 3 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 [30], nous motivons le problème identifié et présentons l'idée de notre approche pour y répondre.
- (ii) Dans [31], nous détaillons une première partie de notre approche et présentons notre protocole d'évaluation expérimentale ainsi que ses premiers résultats.
- (iii) Dans [23], 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 [30]

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 [31]

Auteurs Matthieu Nicolas, Gérald Oster, Olivier Perrin

Article de workshop à 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 [23]

Auteurs Matthieu Nicolas, Gérald 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] WIKIMEDIA. *Wikimedia Statistics - English Wikipedia*. Last Accessed : 2022-10-06. URL : <https://stats.wikimedia.org/#/en.wikipedia.org>.
- [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] Taylor LORENZ. *Internet communities are battling over pixels*. Last Accessed : 2022-10-18. URL : <https://www.washingtonpost.com/technology/2022/04/04/reddit-place-internet-communities/>.
- [4] Matthew O'MARA. *Twitch Plays Pokémon a wild experiment in crowd sourced gameplay*. Last Accessed : 2022-10-18. URL : <https://financialpost.com/technology/gaming/twitch-plays-pokemon-a-wild-experiment-in-crowd-sourced-gameplay>.
- [5] Paul BARAN. « On distributed communications networks ». In : *IEEE transactions on Communications Systems* 12.1 (1964), p. 1–9.
- [6] WIKIPEDIA. *Censorship of Wikipedia*. Last Accessed : 2022-10-18. URL : https://en.wikipedia.org/wiki/Censorship_of_Wikipedia.
- [7] Cody ODGEN. *Google Graveyard*. Last Accessed : 2022-10-11. URL : <https://killedbygoogle.com/>.
- [8] Glen GREENWALD et Ewen MACASKILL. *NSA Prism program taps in to user data of Apple, Google and others*. Last Accessed : 2022-10-07. URL : <https://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data>.
- [9] Barton GELLMAN et Laura POITRAS. *U.S., British intelligence mining data from nine U.S. Internet companies in broad secret program*. Last Accessed : 2022-10-07. URL : https://www.washingtonpost.com/investigations/us-intelligence-mining-data-from-nine-us-internet-companies-in-broad-secret-program/2013/06/06/3a0c0da8-cebf-11e2-8845-d970ccb04497_story.html.
- [10] 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>.

- [11] 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>.
- [12] 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>.
- [13] 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>.
- [14] Leslie LAMPORT. « The part-time parliament ». In : *Concurrency : the Works of Leslie Lamport*. 2019, p. 277–317.
- [15] 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.
- [16] Marc SHAPIRO et Nuno PREGUIÇA. *Designing a commutative replicated data type*. Research Report RR-6320. INRIA, 2007. URL : <https://hal.inria.fr/inria-00177693>.
- [17] 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.
- [18] 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>.
- [19] 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>.
- [20] Matthieu NICOLAS, Victorien ELVINGER, G  rald OSTER, Claudia-Lavinia IGNAT et Fran  ois 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, ao  t 2017, p. 1–4. DOI : 10.18420/ecscw2017_p5. URL : <https://hal.inria.fr/hal-01655438>.

-
- [21] Luc ANDRÉ, Stéphane MARTIN, Gérard 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.
 - [22] 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>.
 - [23] 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>.
 - [24] 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>.
 - [25] 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>.
 - [26] 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.
 - [27] 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.
 - [28] 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.
 - [29] 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.

-
- [42] Paulo Sérgio 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.
 - [43] Paulo Sérgio ALMEIDA, Ali SHOKER et Carlos BAQUERO. « Delta state replicated data types ». In : *Journal of Parallel and Distributed Computing* 111 (jan. 2018), p. 162–173. ISSN : 0743-7315. DOI : 10.1016/j.jpdc.2017.08.003. URL : <http://dx.doi.org/10.1016/j.jpdc.2017.08.003>.
 - [44] Prince MAHAJAN, Lorenzo ALVISI, Mike DAHLIN et al. « Consistency, availability, and convergence ». In : *University of Texas at Austin Tech Report* 11 (2011), p. 158.
 - [45] Friedemann MATTERN et al. *Virtual time and global states of distributed systems*. Univ., Department of Computer Science, 1988.
 - [46] 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>.
 - [47] 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>.
 - [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] Giuseppe DECANDIA, Deniz HASTORUN, Madan JAMPANI, Gunavardhan KAKULAPATI, Avinash LAKSHMAN, Alex PILCHIN, Swaminathan SIVASUBRAMANIAN, Peter VOSSHALL et Werner VOGELS. « Dynamo : Amazon’s highly available key-value store ». In : *ACM SIGOPS operating systems review* 41.6 (2007), p. 205–220.
 - [50] Nico KRUBER, Maik LANGE et Florian SCHINTKE. « Approximate Hash-Based Set Reconciliation for Distributed Replica Repair ». In : *2015 IEEE 34th Symposium on Reliable Distributed Systems (SRDS)*. 2015, p. 166–175. DOI : 10.1109/SRDS.2015.30.
 - [51] Ricardo Jorge Tomé GONÇALVES, Paulo Sérgio ALMEIDA, Carlos BAQUERO et Victor FONTE. « DottedDB : Anti-Entropy without Merkle Trees, Deletes without Tombstones ». In : *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*. 2017, p. 194–203. DOI : 10.1109/SRDS.2017.28.
 - [52] 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>.

- [53] Carlos BAQUERO, Paulo Sérgio ALMEIDA et Ali SHOKER. « Making Operation-Based CRDTs Operation-Based ». In : *Distributed Applications and Interoperable Systems*. Sous la dir. de Kostas MAGOUTIS et Peter PIETZUCH. Berlin, Heidelberg : Springer Berlin Heidelberg, 2014, p. 126–140.
- [54] Vitor ENES, Paulo Sérgio ALMEIDA, Carlos BAQUERO et João LEITÃO. « 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.
- [55] Clarence A. ELLIS et Simon J. GIBBS. « Concurrency Control in Groupware Systems ». In : *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data*. SIGMOD '89. Portland, Oregon, USA : Association for Computing Machinery, 1989, p. 399–407. ISBN : 0897913175. DOI : 10.1145/67544.66963. URL : <https://doi.org/10.1145/67544.66963>.
- [56] Chengzheng SUN et Clarence ELLIS. « Operational transformation in real-time group editors : issues, algorithms, and achievements ». In : *Proceedings of the 1998 ACM conference on Computer supported cooperative work*. 1998, p. 59–68.
- [57] Matthias RESSEL, Doris NITSCHERUHLAND et Rul GUNZENHÄUSER. « An integrating, transformation-oriented approach to concurrency control and undo in group editors ». In : *Proceedings of the 1996 ACM conference on Computer supported cooperative work*. 1996, p. 288–297.
- [58] Chengzheng SUN, Yun YANG, Yanchun ZHANG et David CHEN. « A consistency model and supporting schemes for real-time cooperative editing systems ». In : *Australian Computer Science Communications* 18 (1996), p. 582–591.
- [59] David SUN et Chengzheng SUN. « Context-Based Operational Transformation in Distributed Collaborative Editing Systems ». In : *Parallel and Distributed Systems, IEEE Transactions on* 20 (nov. 2009), p. 1454–1470. DOI : 10.1109/TPDS.2008.240.
- [60] Chengzheng SUN, Xiaohua JIA, Yanchun ZHANG, Yun YANG et David CHEN. « Achieving convergence, causality preservation, and intention preservation in real-time cooperative editing systems ». In : *ACM Transactions on Computer-Human Interaction (TOCHI)* 5.1 (1998), p. 63–108.
- [61] Gérald OSTER, Pascal MOLLI, Pascal URSO et Abdessamad IMINE. « Tombstone Transformation Functions for Ensuring Consistency in Collaborative Editing Systems ». In : *2006 International Conference on Collaborative Computing : Networking, Applications and Worksharing*. 2006, p. 1–10. DOI : 10.1109/COLCOM.2006.361867.
- [62] Chengzheng SUN, Xiaohua JIA, Yanchun ZHANG, Yun YANG et David CHEN. « Achieving Convergence, Causality Preservation, and Intention Preservation in Real-Time Cooperative Editing Systems ». In : *ACM Trans. Comput.-Hum. Interact.* 5.1 (mar. 1998), p. 63–108. ISSN : 1073-0516. DOI : 10.1145/274444.274447. URL : <https://doi.org/10.1145/274444.274447>.

- [63] 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>.
- [64] Bernadette CHARRON-BOST. « Concerning the size of logical clocks in distributed systems ». In : *Information Processing Letters* 39.1 (1991), p. 11–16.
- [65] 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>.
- [66] 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>.
- [67] 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.
- [68] Paulo Sérgio ALMEIDA, Carlos BAQUERO, Ricardo GONÇALVES, Nuno PREGUIÇA et Victor FONTE. « Scalable and Accurate Causality Tracking for Eventually Consistent Stores ». In : *Distributed Applications and Interoperable Systems*. Sous la dir. de Kostas MAGOUTIS et Peter PIETZUCH. Berlin, Heidelberg : Springer Berlin Heidelberg, 2014, p. 67–81. ISBN : 978-3-662-43352-2.
- [69] Charbel RAHHAL, Stéphane WEISS, Hala SKAF-MOLLI, Pascal URSO et Pascal MOLLI. *Undo in Peer-to-peer Semantic Wikis*. Research Report RR-6870. INRIA, 2009, p. 18. URL : <https://hal.inria.fr/inria-00366317>.
- [70] Mehdi AHMED-NACER, Claudia-Lavinia IGNAT, Gérald OSTER, Hyun-Gul ROH et Pascal URSO. « Evaluating CRDTs for Real-time Document Editing ». In : *11th ACM Symposium on Document Engineering*. Sous la dir. d’ACM. Mountain View, California, United States, sept. 2011, p. 103–112. DOI : 10.1145/2034691.2034717. URL : <https://hal.inria.fr/inria-00629503>.
- [71] Stéphane WEISS, Pascal URSO et Pascal MOLLI. « Wooki : a P2P Wiki-based Collaborative Writing Tool ». In : t. 4831. Déc. 2007. ISBN : 978-3-540-76992-7. DOI : 10.1007/978-3-540-76993-4_42.
- [72] Ben SHNEIDERMAN. « Response Time and Display Rate in Human Performance with Computers ». In : *ACM Comput. Surv.* 16.3 (sept. 1984), p. 265–285. ISSN : 0360-0300. DOI : 10.1145/2514.2517. URL : <https://doi.org/10.1145/2514.2517>.

- [73] Caroline JAY, Mashhuda GLENCROSS et Roger HUBBOLD. « Modeling the Effects of Delayed Haptic and Visual Feedback in a Collaborative Virtual Environment ». In : *ACM Trans. Comput.-Hum. Interact.* 14.2 (août 2007), 8–es. ISSN : 1073-0516. DOI : 10.1145/1275511.1275514. URL : <https://doi.org/10.1145/1275511.1275514>.
- [74] Hagit ATTIYA, Sebastian BURCKHARDT, Alexey GOTSMAN, Adam MORRISON, Hongseok YANG et Marek ZAWIRSKI. « Specification and Complexity of Collaborative Text Editing ». In : *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*. PODC '16. Chicago, Illinois, USA : Association for Computing Machinery, 2016, p. 259–268. ISBN : 9781450339643. DOI : 10.1145/2933057.2933090. URL : <https://doi.org/10.1145/2933057.2933090>.
- [75] Hagit ATTIYA, Sebastian BURCKHARDT, Alexey GOTSMAN, Adam MORRISON, Hongseok YANG et Marek ZAWIRSKI. « Specification and space complexity of collaborative text editing ». In : *Theoretical Computer Science* 855 (2021), p. 141–160. ISSN : 0304-3975. DOI : <https://doi.org/10.1016/j.tcs.2020.11.046>. URL : <http://www.sciencedirect.com/science/article/pii/S0304397520306952>.
- [76] AUTOMERGE. *Automerge : data structures for building collaborative applications in Javascript*. Last Accessed : 2022-10-07. URL : <https://github.com/automerge/automerge>.
- [77] Loïck BRIOT, Pascal URSO et Marc SHAPIRO. « High Responsiveness for Group Editing CRDTs ». In : *ACM International Conference on Supporting Group Work*. Sanibel Island, FL, United States, nov. 2016. DOI : 10.1145/2957276.2957300. URL : <https://hal.inria.fr/hal-01343941>.
- [78] Weihai YU. « A String-Wise CRDT for Group Editing ». In : *Proceedings of the 17th ACM International Conference on Supporting Group Work*. GROUP '12. Sanibel Island, Florida, USA : Association for Computing Machinery, 2012, p. 141–144. ISBN : 9781450314862. DOI : 10.1145/2389176.2389198. URL : <https://doi.org/10.1145/2389176.2389198>.
- [79] 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>.
- [80] Matthew WEIDNER. *There Are No Doubly Non-Interleaving List CRDTs*. Last Accessed : 2022-10-07. URL : https://mattweidner.com/assets/pdf/List_CRDT_Non_Interleaving.pdf.
- [81] Stéphane WEISS, Pascal URSO et Pascal MOLLI. « Logoot-Undo : Distributed Collaborative Editing System on P2P Networks ». In : *IEEE Transactions on Parallel and Distributed Systems* 21.8 (août 2010), p. 1162–1174. DOI : 10.1109/TPDS.2009.173. URL : <https://hal.archives-ouvertes.fr/hal-00450416>.

-
- [82] 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>.
 - [83] 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>.
 - [84] Brice N  DELEC, Pascal MOLLI, Achour MOST  FAOUI 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.
 - [85] Brice N  DELEC, Pascal MOLLI et Achour MOST  FAOUI. « 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>.
 - [86] 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.
 - [87] Victorien ELVINGER, G  rald OSTER et Fran  ois 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.
 - [88] Sylvie NO  L 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.
 - [89] Jim GILES. « Special Report Internet encyclopaedias go head to head ». In : *nature* 438.15 (2005), p. 900–901.
 - [90] GOOGLE. *Google Docs*. Last Accessed : 2022-10-07. URL : <https://docs.google.com/>.
 - [91] OPENRELAY. *OpenRelay*. Last Accessed : 2022-10-07. URL : <https://openrelay.xyz/>.

- [92] Protocol LABS. *IPFS*. Last Accessed : 2022-10-07. URL : <https://ipfs.io/>.
- [93] 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>.
- [94] 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>.
- [95] 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>.