Local-First software

Local-First Software: Collaborative Spreadsheet Editing

Sandro Gössi & Fabian Gubler

Supervisors: George Zakhour & Dr. Pascal Weisenburger

University of St.Gallen

Our Team

Sandro Gössi

Master of Computer Science

University of St. Gallen

Fabian Gubler

Master of Computer Science

University of St. Gallen

Our Supervisors

George Zakhour

PhD Student, Programming Group

University of St. Gallen
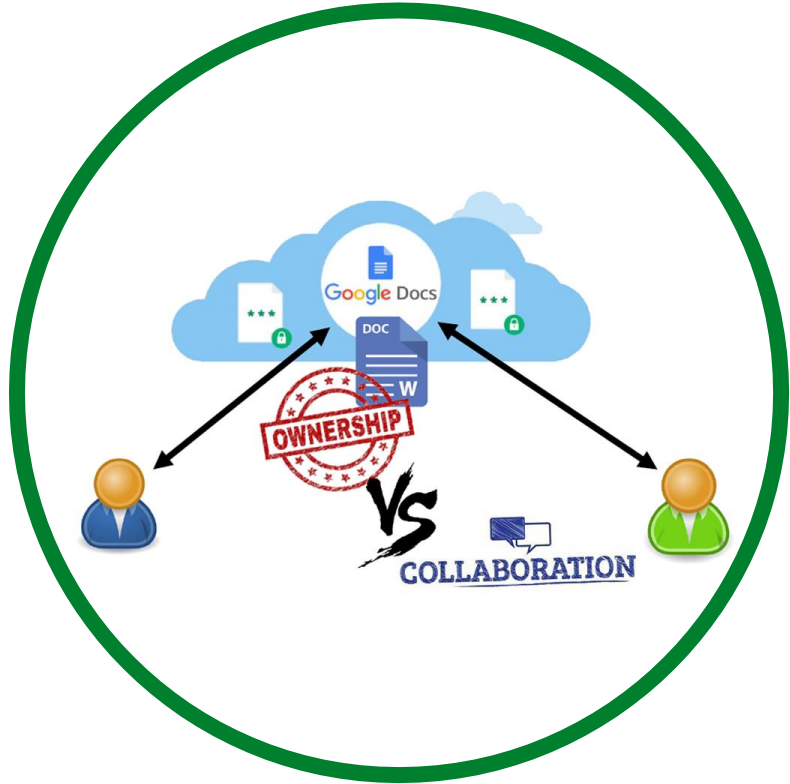*Alumni of EPFL and AUB*

Dr. Pascal Weisenburger

Postdoc, Programming Group

University of St. Gallen
Alumni of TU Darmstadt

Table Of Contents

1. Motivation

2. Project Description

3. Introduction to Local-First

4. Problem Statement

5. Methodological Approach

6. Progress & Findings

7. Project Goals, Roadmap & Milestones

Motivation

Why Local-First Software?

Platform Capitalism: «Enshittification»

Haven't you noticed, lately, as belts tighten across Silicon Valley, that your own personal internet feels <u>less abundant</u> than before?

That certain things are getting a <u>little more expensive</u>, or a <u>little less convenient</u>?

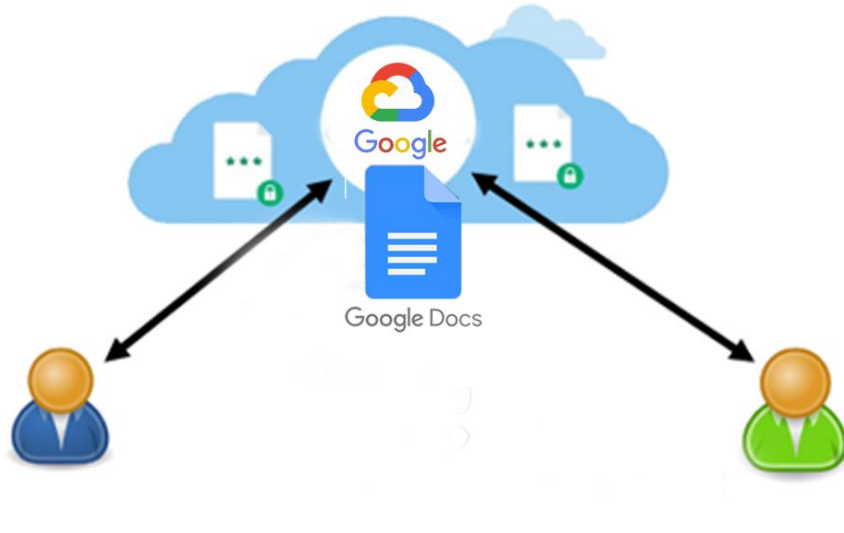# The Cloud Is A Prison

Data Privacy Concerns

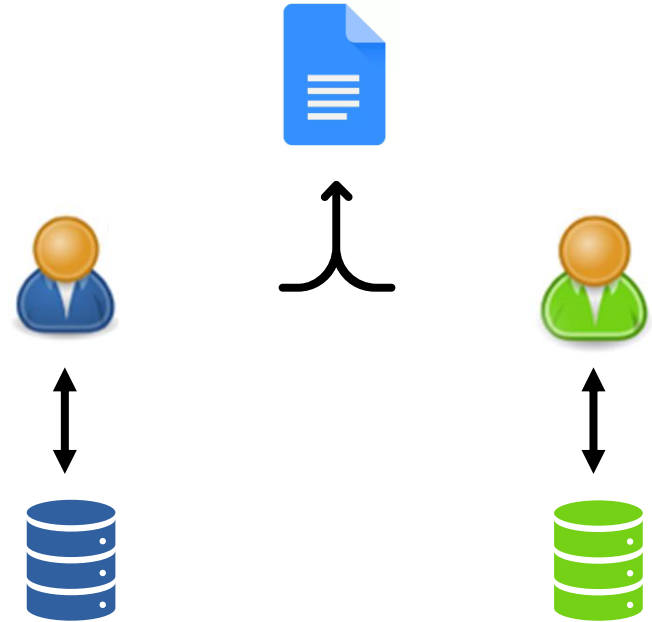**Network And Platform Dependencies**

Loss of Control & Cost Implications

Cut Out The Middleman
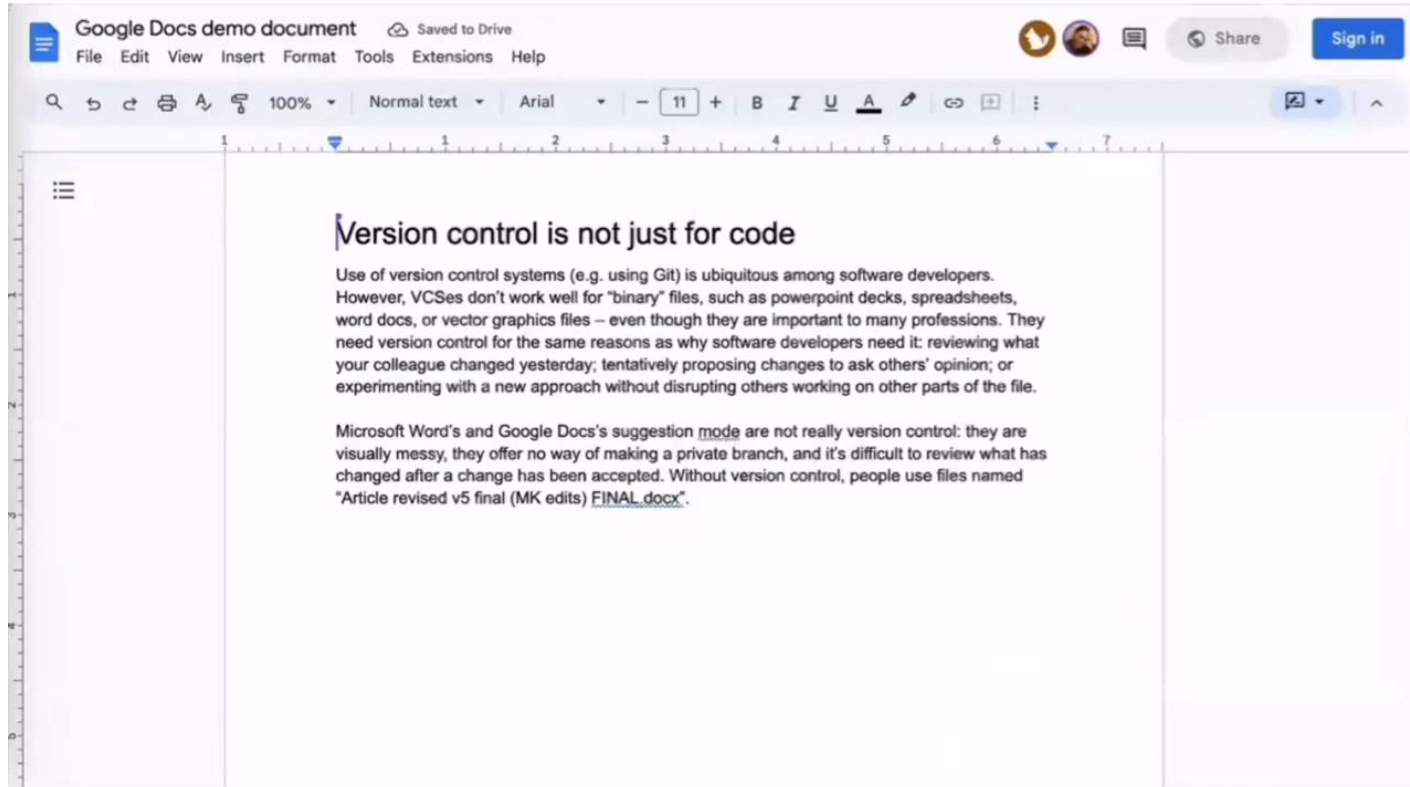
The Cloud

Local-First

Project Description

What Is Our Project About?

Status Quo: Real-Time Collaboration In The Cloud

Collaborative Local-First Spreadsheet?

Introduction to Local-First

What Is Local-First Software?

Definition: Local-First Software

✓ design approach / set of principles

✓ Prioritizing the user's <u>local device</u>.

✓ Enables both <u>collaboration</u> and <u>local data ownership</u>.

✓ Prioritizes the use of <u>local storage</u> and <u>local networks</u> over servers in remote data centers or the cloud.

Seven Principles/Ideals for Local-First Software (1)

1. **Speed:** Near-Instant Response Time

2. **Synchronization:** Multi- And Cross-Device Sync

3. **Offline First:** The Network Is Optional

4. **Collaboration:** Seamless Collaboration

Seven Principles/Ideals for Local–First Software (2)

5. **Longevity:** The Long Now

6. **Decentralization:** <u>Security</u> And <u>Privacy</u> By Avoiding Centralized Cloud Databases

7. **Data Ownership:** Retain Ultimate Ownership And Control Over Your Data

Various Technologies Analyzed By Local-First Ideals

Technology	1. Fast	2. Multi-device	3. Offline	4. Collaboration	5. Longevity	6. Privacy	7. User control
Applications employed by end users							
Files + email attachments	✓	—	✓	●	✓	—	✓

✓ means the technology meets the ideal,

— means it partially meets the ideal,

● means it does not meet the ideal.

Various Technologies Analyzed By Local-First Ideals

Technology	1. Fast	2. Multi-device	3. Offline	4. Collaboration	5. Longevity	6. Privacy	7. User control
Applications employed by end users							
Files + email attachments	✓	—	✓	●	✓	—	✓
Google Docs	—	✓	—	✓	—	●	—

✓ means the technology meets the ideal,

— means it partially meets the ideal,

● means it does not meet the ideal.

24 October 2023

Various Technologies Analyzed By Local-First Ideals

Technology	1. Fast	2. Multi-device	3. Offline	4. Collaboration	5. Longevity	6. Privacy	7. User control
Applications employed by end users							
Files + email attachments	✓	—	✓	●	✓	—	✓
Google Docs	—	✓	—	✓	—	●	—
Git + GitHub	✓	—	✓	—	✓	—	✓

✓ means the technology meets the ideal,

— means it partially meets the ideal,

● means it does not meet the ideal.

Technologies Powering Local-First Software

CRDTs (Conflict-free Replicated Data Types)

Data structure that allows multiple replicas to be updated independently without coordination and later be merged without conflicts, ensuring consistent data across devices.
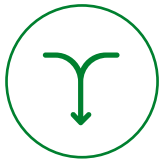
PROJECT FOCUS

P2P Protocols

Communication methods that allow software applications to interact directly with each other without the need for a central server, facilitating decentralized data exchange and collaboration.

Local Databases And Storage Solutions

Systems that store and manage data directly on a user's device, providing fast access and operations without relying on remote servers.

Synchronization And Merging Techniques

Algorithms and methods that ensure data consistency across devices by resolving differences and merging changes made in different locations or times.

PROJECT FOCUS

Problem Statement

What Is The Concrete Problem We Want To Solve?

Collaboration: Merging Two Excel Formulas

Excel Formula
Formula before any edits were made:

=SUMME(A2:A9)

User 1
Changes the Formula to:

=SUMME(A3:A10)

User 2
Changes the Formula to:

=SUMME(A2:A9)/B5

RESULT?
How do we merge this
two formulas?

Collaboration: Result After Merging

Excel Formula

Formula before any edits were made:

=SUMME(A2:A9)

User 1

Changes the Formula to:

=SUMME(A3:A10)

User 2

Changes the Formula to:

=SUMME(A2:A9)/B5

=SUMME(A3:A10)/B5

Methodological Approach

How Do We Plan To Solve The Problem?

Basic Text Merging Does NOT Work!

✗ **Loss of Formula Structure:** Merging text directly can disrupt the inherent structure of Excel formulas.

✗ **Syntax Errors:** Direct text concatenation risks creating syntactically incorrect formulas

✗ **Nesting Issues:** Text merging can disrupt nested functions, leading to broken formulas.

✗ **No Semantic Understanding:** Basic merging lacks a deep understanding of formula logic, potentially resulting in logic errors.

✗ **Redundancy and Duplication:** Simple text concatenation can introduce redundant or duplicate functions and operations.

Strings

char str[] = "HELLO";

| H | E | L | L | O | \0 |

String Concatenate

"Hello" + "World" = " HelloWorld "

String 1   String 2   Result

Abstract Syntax Tree (AST)
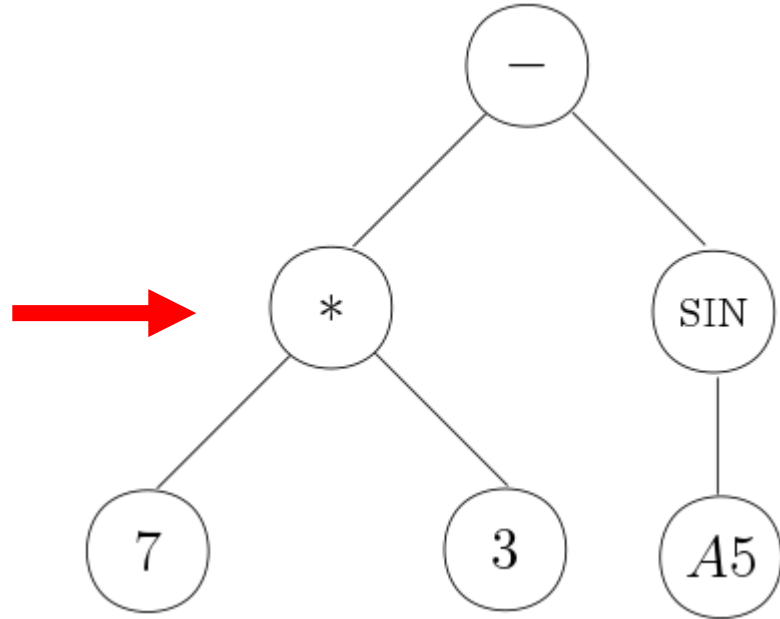
CRDTs For Abstract Syntax Tree (AST)

CRDTs (Conflict-free Replicated Data Types)

*Data structure that allows multiple replicas to be updated independently without coordination and later be **merged without conflicts**, ensuring **consistent data** across devices.*

- ✓ Commutative
- ✓ Idempotent
- ✓ Associative
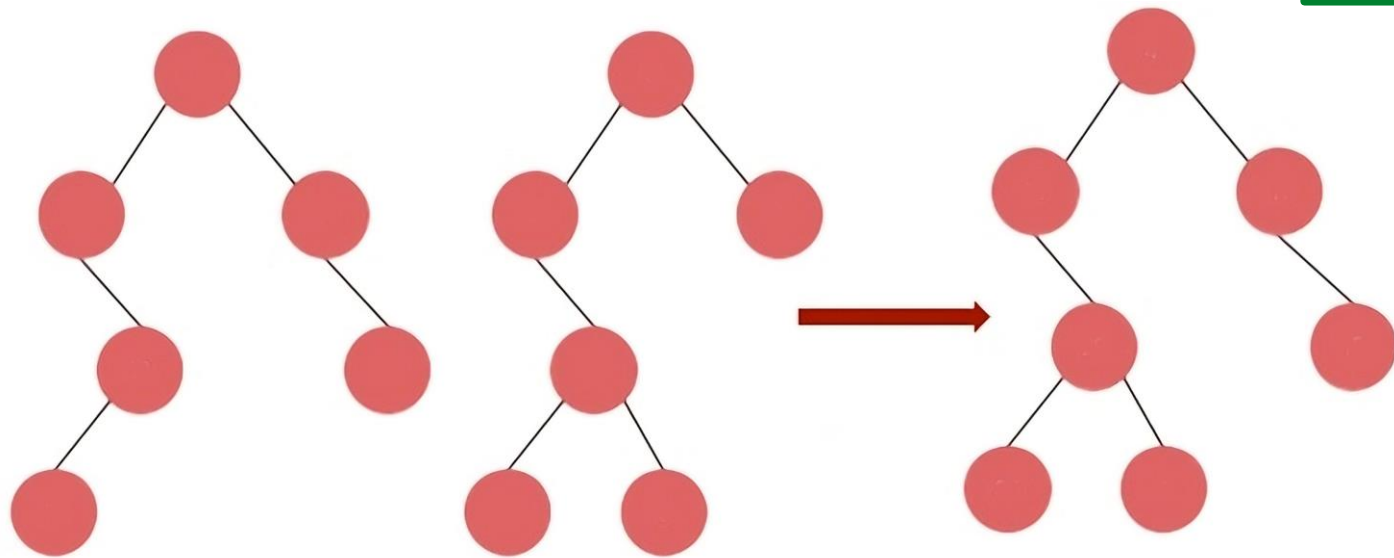
Why CRDTs? – Popularity In Academia & Industry

CRDTs solve distributed data consistency challenges

I was wrong. CRDTs are the future

CRDT — Collaboration Protocol of the future!

Autumn Semester 2023: Integrative Master's Projects – Programming Group
Sources: Gentle, 2020; Porwal, 2023; Stichbury, 2022

Two Types of CRDTs

Operation-Based CRDTs

Operation-based CRDTs are also called commutative replicated data types, or CmRDTs. CmRDT replicas propagate state by transmitting only the update operation.

The operations are <u>commutative</u>. However, they are <u>not</u> necessarily <u>idempotent</u>.

Example: *For example, a CmRDT of a single integer might broadcast the operations (+10) or (−20). Replicas receive the updates and apply them locally.*

State-Based CRDTs

State-based CRDTs are called convergent replicated data types, or CvRDTs. In contrast to CmRDTs, CvRDTs send their full local state to other replicas.

The states are merged by a function which must be <u>commutative</u>, <u>associative</u>, and <u>idempotent</u>.

Example: *Instead of broadcasting operations like "add item A" or "remove item B", it shares its entire state (i.e., the whole set) with other replicas.*

University of St.Gallen

Examples Of Known CRDTs

Different CRDTs

- G-Counter (Grow-only Counter)
- PN-Counter (Positive-Negative Counter)
- G-Set (Grow-only Set)
- 2P-Set (Two-Phase Set)
- LWW-Element-Set (Last-Write-Wins-Element-Set)
- OR-Set (Observed-Remove Set)
- Vector Clocks

Example 1: G-Counter

```
Node A Counter: {A: 3, B: 4}
Node B Counter: {A: 5, B: 2}
After merging:   {A: 5, B: 4}
```

Merge: Max of corresponding elements A: 5, B: 4.

TotalValue: Sum of all elements 5 + 4 = 9

Example 2: LWW-Register

```
Node A Register: ("apple", timestampA)
Node B Register: ("banana", timestampB)

If timestampA > timestampB:
Merged Register: ("apple", timestampA)
Else if timestampA < timestampB:
Merged Register: ("banana", timestampB)
Else:
Merged Register: Depends on tie-breaker
```

LWW-Register: Keeps the latest write in case of concurrent writes based on timestamps.

Progress & Findings

What Have We Done So Far?

Progress Integrative Masters Project 2023

Research

Local-First
Understand the concept of Local-First Software by studying recent publications, papers, and lectures.

CRDTs
Research different types of CRDTs and understand their operation and mechanisms.

Syntax Tree
Analyze why we cannot use text-based CRDTs and must use abstract syntax trees (AST) instead.

Coding

Existing Code
Examine existing implemented CRDTs and GitHub repositories to understand their functionality.

GitHub
Create GitHub Repository and readme files for project collaboration.

Example CRDTs
Code Basic CRDTs
- *GCounter*
- *PNCOunter*
- *LWWRegister*

Operation vs. State-based CRDTs
- Build Foundation and Understanding

Other

Planning
Create the roadmap, schedule, and milestones for our project to keep us on track.

Branding
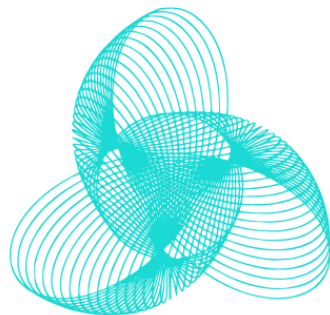Create a logo and brand name for our application for marketing purposes.

Slide Deck
Creation of the slide document for the IMP progress presentation to properly communicate our project.

Cellster: Empowering Offline Collaboration In Spreadsheets

Cellster

Empowering Offline Collaboration in Spreadsheets

Break free from the chains of constant connectivity with the power of local-first software. Dive into a revolutionary way of collaboratively editing spreadsheet formulas without the need for real-time online presence. Experience seamless merges using CRDTs and witness the power of structured conflict resolution.

Project Goal, Roadmap & Milestones

What Are The Next Steps?

Project Goal & Additional Milestones

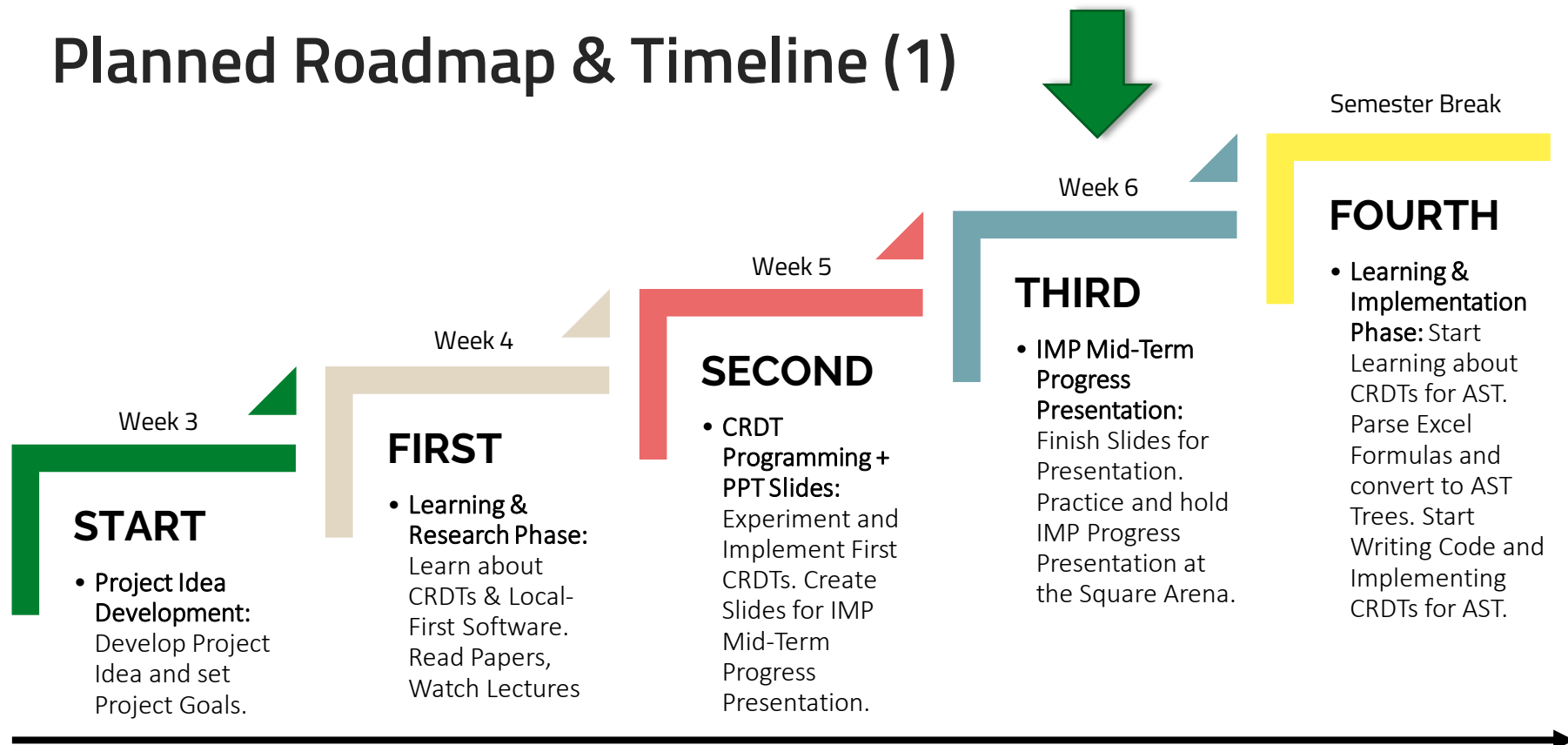🏆 **Milestone 1 (Project Goal):** "Excel formula CRDT" – Building CRDT for Merging an AST

🏆 **Milestone 2:** "Spreadsheet Grid CRDT" – Building an CRDT for Spreadsheets (i.e. Grid with Cells)
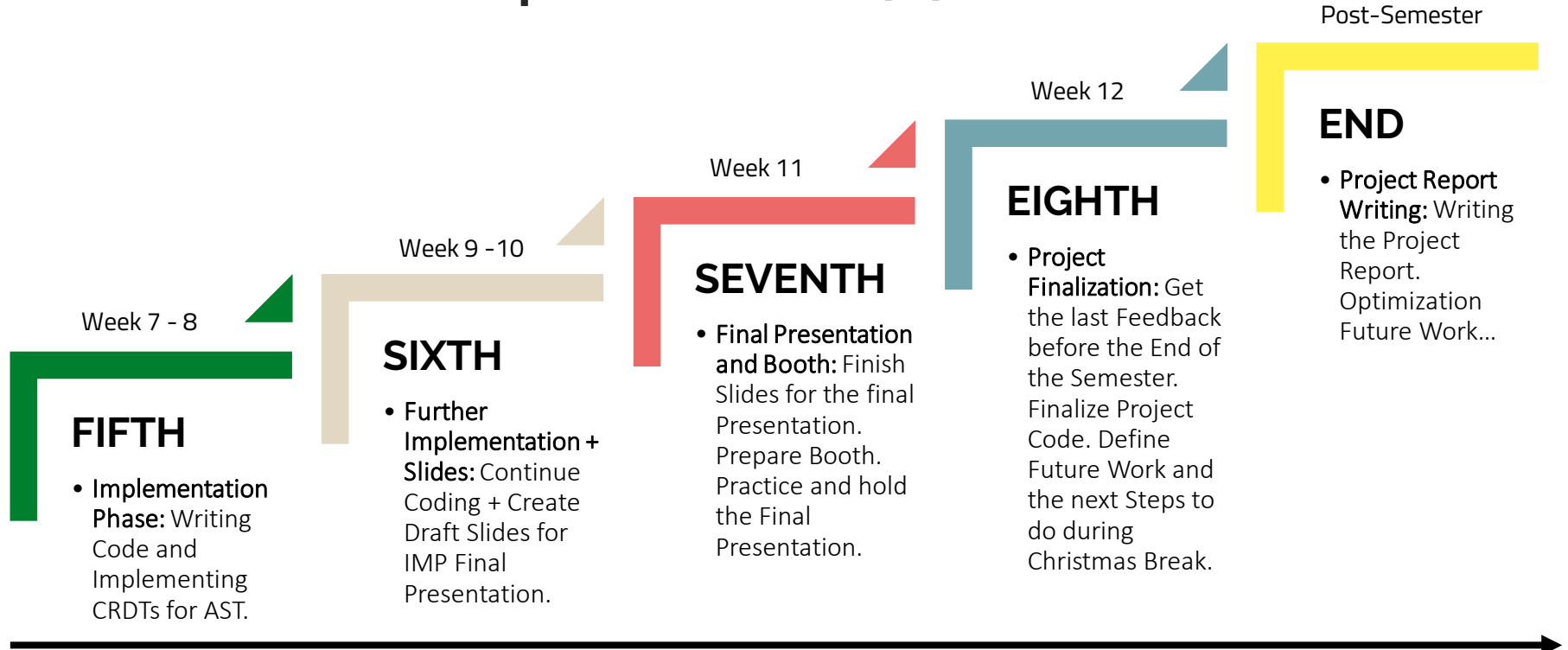
🏆 **Milestone 3:** "Cellster" - Collaborative Spreadsheet Application

Cellster

Planned Roadmap & Timeline (1)

Week 3

START

- **Project Idea Development:** Develop Project Idea and set Project Goals.

Week 4

FIRST

- **Learning & Research Phase:** Learn about CRDTs & Local-First Software. Read Papers, Watch Lectures

Week 5

SECOND

- **CRDT Programming + PPT Slides:** Experiment and Implement First CRDTs. Create Slides for IMP Mid-Term Progress Presentation.

Week 6

THIRD

- **IMP Mid-Term Progress Presentation:** Finish Slides for Presentation. Practice and hold IMP Progress Presentation at the Square Arena.

Semester Break

FOURTH

- **Learning & Implementation Phase:** Start Learning about CRDTs for AST. Parse Excel Formulas and convert to AST Trees. Start Writing Code and Implementing CRDTs for AST.

Planned Roadmap & Timeline (2)

Post-Semester

Week 12

Week 11

Week 9 -10

Week 7 - 8

FIFTH

- **Implementation Phase:** Writing Code and Implementing CRDTs for AST.

SIXTH

- **Further Implementation + Slides:** Continue Coding + Create Draft Slides for IMP Final Presentation.

SEVENTH

- **Final Presentation and Booth:** Finish Slides for the final Presentation. Prepare Booth. Practice and hold the Final Presentation.

EIGHTH

- **Project Finalization:** Get the last Feedback before the End of the Semester. Finalize Project Code. Define Future Work and the next Steps to do during Christmas Break.

END

- **Project Report Writing:** Writing the Project Report. Optimization Future Work...

Thank You For Your Attention!

Q&A: Any Questions?

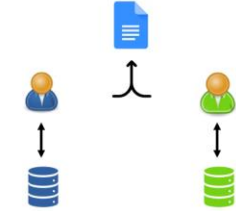# Summary

Local-First

Local First Principles
1. Fast
2. Multi-Device
3. Offline
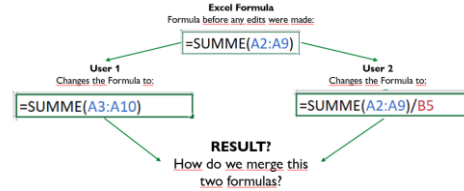4. Collaboration
5. Longevity
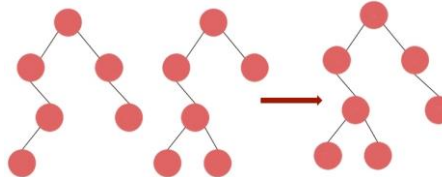6. Privacy
7. User Control
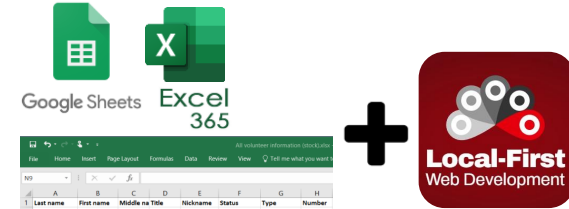
Problem & Solution
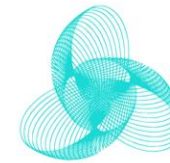

Abstract Syntax Trees (AST)

CRDTs For Merging ASTs

Project Goals

Milestone 1 (Project Goal): "Excel formula CRDT" – Building CRDT for Merging an AST

List of References & Resources

Barber, G. (2023). The cloud is a prison. Can the Local-First Software movement set us free? *WIRED*. https://www.wired.com/story/the-cloud-is-a-prison-can-the-local-first-software-movement-set-us-free/

Channel. (2021). *CRDT vs OT*. Channel.io. https://channel.io/ko/blog/crdt_vs_ot

Colyer, A. (2019). *Local-first software: you own your data, in spite of the cloud.* The Morning Paper. https://blog.acolyer.org/2019/11/20/local-first-software/

Gentle, J. (2020). I was wrong. CRDTs are the future. *Seph*. https://josephg.com/blog/crdts-are-the-future/

HyperFormula. (2022). *Key Concepts | HyperFormula (V2.6.0)*. https://hyperformula.handsontable.com/guide/key-concepts.html#high-level-design-diagram

Kleppmann, M., Wiggins, A., Van Hardenberg, P. & McGranaghan, M. (2019). Local-first software: you own your data, in spite of the cloud. *Onward!* https://doi.org/10.1145/3359591.3359737

Porwal, P. (2023). CRDT — Collaboration Protocol of the Future! - Piyush Porwal - medium. *Medium*. https://medium.com/@ppiyush/crdt-collaboration-protocol-of-the-future-c9990c1db748

Preguiça, N. (2018). *Conflict-free Replicated Data types: an overview*. arXiv.org. https://arxiv.org/abs/1806.10254

Stichbury, J. (2022). CRDTs solve distributed data consistency challenges. *Ably Realtime*. https://ably.com/blog/crdts-distributed-data-consistency-challenges

Thoughtworks. (2022). *Local-First Application*. https://www.thoughtworks.com/radar/techniques/local-first-application

rs. (2023). Conflict-free replicated data type. *Wikipedia*. https://en.wikipedia.org/wiki/Conflict-free_replicated_data_type

Code Example CRDTs

```python
# Implement LWWRegister Class
class LWWRegister:
    def __init__(self):
        self.value = None
        self.timestamp = {}  # node -> timestamp

    def write(self, node, value, timestamp):
        if node not in self.timestamp or timestamp > self.timestamp[node]:
            self.value = value
            self.timestamp[node] = timestamp

    def read(self):
        return self.value

    def current_timestamp(self):
        return max(self.timestamp.values()) if self.timestamp else None

    def merge(self, other):
        for node, timestamp in other.timestamp.items():
            if node not in self.timestamp or timestamp > self.timestamp[node]:
                self.value = other.value
                self.timestamp[node] = timestamp
```