

Work in Parallel, Without Fear

Conflict Tolerance for Local-First Collaboration

Jonas Schürmann

Local-First Cologne, Cologne, 2026-10-15

Chair for Programming Systems, TU Dortmund University

- Great data layer for local-first live collaboration:
 - Data is fully stored on-device
 - Unrestricted local editing without coordination
 - After syncing up, everybody will converge on the same document

¹<https://crdt.tech/>

- Great data layer for local-first live collaboration:
 - Data is fully stored on-device
 - Unrestricted local editing without coordination
 - After syncing up, everybody will converge on the same document
- CRDTs are decentralized and consensus-free
- Conflicts are resolved automatically in a deterministic fashion
- Eventual convergence is proven with lattice theory

¹<https://crdt.tech/>

DEMO

DEMO

- Live collaboration is awesome, but sometimes you want to work alone to...
 - ...avoid interruptions
 - ...work offline
 - ...keep ideas to yourself

- Live collaboration is awesome, but sometimes you want to work alone to...
 - ...avoid interruptions
 - ...work offline
 - ...keep ideas to yourself
- Working alone means concurrent edits pile up

- Live collaboration is awesome, but sometimes you want to work alone to...
 - ...avoid interruptions
 - ...work offline
 - ...keep ideas to yourself
- Working alone means concurrent edits pile up
- When CRDTs eventually merge them, automatic conflict resolution can silently discard my edits

- Branches provide isolated workspaces

- Branches provide isolated workspaces
- But: Upon merging, conflict resolution is a huge pain

- Branches provide isolated workspaces
- But: Upon merging, conflict resolution is a huge pain
- If you are the unlucky one to hit conflicts, you have to resolve them...
 - ...immediately
 - ...alone
 - ...without context

- Branches provide isolated workspaces
- But: Upon merging, conflict resolution is a huge pain
- If you are the unlucky one to hit conflicts, you have to resolve them...
 - ...immediately
 - ...alone
 - ...without context
- The result:
 - Interruptions
 - Premature resolution decisions
 - Dismissal of others' work
 - Introduction of errors

- Branches provide isolated workspaces
- But: Upon merging, conflict resolution is a huge pain
- If you are the unlucky one to hit conflicts, you have to resolve them...
 - ...immediately
 - ...alone
 - ...without context
- The result:
 - Interruptions
 - Premature resolution decisions
 - Dismissal of others' work
 - Introduction of errors

⇒ Despite its usefulness, branching & merging is rare in local-first apps

- Conflicts are first-class objects in the document, so you can
 - Defer the resolution decision
 - Share conflicts with others
 - Resolve conflicts collaboratively

²<https://www.inkandswitch.com/universal-version-control/>

- Conflicts are first-class objects in the document, so you can
 - Defer the resolution decision
 - Share conflicts with others
 - Resolve conflicts collaboratively
- Benefits [4]:
 - Preserve all contributions
 - Uncover disagreements
 - Make deliberate and well-informed resolution decisions

²<https://www.inkandswitch.com/universal-version-control/>

- Conflicts are first-class objects in the document, so you can
 - Defer the resolution decision
 - Share conflicts with others
 - Resolve conflicts collaboratively
- Benefits [4]:
 - Preserve all contributions
 - Uncover disagreements
 - Make deliberate and well-informed resolution decisions
- Lazy merging: Integrate syntax, history, and conflicts in one CRDT [1]

²<https://www.inkandswitch.com/universal-version-control/>

- Conflicts are first-class objects in the document, so you can
 - Defer the resolution decision
 - Share conflicts with others
 - Resolve conflicts collaboratively
 - Benefits [4]:
 - Preserve all contributions
 - Uncover disagreements
 - Make deliberate and well-informed resolution decisions
 - Lazy merging: Integrate syntax, history, and conflicts in one CRDT [1]
- ⇒ Is conflict tolerance one of the missing pieces for universal version control²?

²<https://www.inkandswitch.com/universal-version-control/>

Version Control



Git + lazy merge driver

Version Control



Git + lazy merge driver

Conflict Representation

```
...  
targetID: {  
  "tag": "choice",  
  "options": ["fed6", "2d4f"]  
}
```

...
First-class conflicts in the
abstract model syntax

Version Control



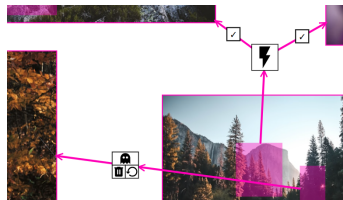
Git + lazy merge driver

Conflict Representation

```
...  
targetID: {  
  "tag": "choice",  
  "options": ["fed6", "2d4f"]  
}  
...
```

First-class conflicts in the
abstract model syntax

Conflict Visualization



Conflict projection in the
Cinco model editor

DEMO

Version Control



Automerge

Version Control



Automerge

Conflict Representation

```
Automerge.getConflicts(  
  cells,  
  0  
)  
  ||  
{  
  "1@a1b2": "#84cc16",  
  "2@c3d4": "#ef4444"  
}
```

Conflicts object

Version Control



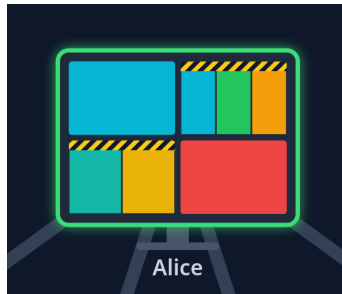
Automerge

Conflict Representation

```
Automerge.getConflicts(  
  cells,  
  0  
)  
  ||  
{  
  "1@a1b2": "#84cc16",  
  "2@c3d4": "#ef4444"  
}
```

Conflicts object

Conflict Visualization



Conflict projection in
React Flow

DEMO

Version Control



Automerge

Version Control



Automerge

Conflict Representation

```
Automerge.getConflicts(  
    node.inputs,  
    inputName  
)  
  
    ||  
  
{  
    "1@a1b2": "silhouette",  
    "2@c3d4": "checkerboard"  
}
```

Conflicts object

Version Control



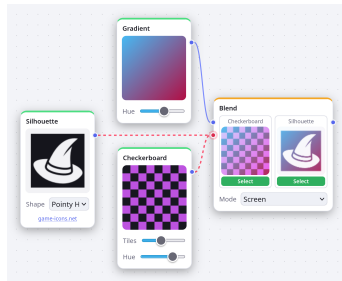
Automerge

Conflict Representation

```
Automerge.getConflicts(  
  node.inputs,  
  inputName  
)  
  
  ||  
  
{  
  "1@a1b2": "silhouette",  
  "2@c3d4": "checkerboard"  
}
```

Conflicts object

Conflict Visualization



Conflict projection in
React Flow

How to develop a conflict-tolerant local-first app



1. Implement a conflict-tolerant CRDT



2. Make the rest of the app conflict-tolerant

- Conflict tolerance is a pervasive cross-cutting concern
- Similar to error handling

³<https://www.inkandswitch.com/project/amsheets/live25/>

- Conflict tolerance is a pervasive cross-cutting concern
- Similar to error handling
- How can we make it easy?
 - Conflict-tolerant UI frameworks
 - Conflict-tolerant databases/query languages
 - Conflict-tolerant APIs/programming languages (see Ambsheets³)

³<https://www.inkandswitch.com/project/ambsheets/live25/>

Thank You!

Slides & Links



<https://jonas-schuermann.name/talks/2026-10-15-local-first-cologne/>

References

- [1] Jonas Schürmann and Bernhard Steffen. **“Lazy Merging: From a Potential of Universes to a Universe of Potentials”**. en. In: *Electronic Communications of the EASST* Volume 82: 11th International Symposium on Leveraging Applications of Formal Methods (2023). DOI: [10.14279/tuj.eceasst.82.1226](https://doi.org/10.14279/tuj.eceasst.82.1226).
- [2] Jonas Schürmann and Bernhard Steffen. **“Lazy Three-Way Model Merging in Cinco Cloud: A Lattice-Theoretic Approach”**. In: *Formal Methods for Industrial Critical Systems*. Ed. by Peter Gorm Larsen and Kristin Yvonne Rozier. Cham: Springer Nature Switzerland, 2026, pp. 118–136. ISBN: 978-3-032-36849-2. DOI: [10.1007/978-3-032-36849-2_7](https://doi.org/10.1007/978-3-032-36849-2_7).
- [3] Marc Shapiro et al. **A comprehensive study of Convergent and Commutative Replicated Data Types**. Research Report RR-7506. Inria – Centre Paris-Rocquencourt ; INRIA, Jan. 2011, p. 50. URL: <https://hal.inria.fr/inria-00555588>.
- [4] Konrad Wieland et al. **“Turning Conflicts into Collaboration”**. In: *Computer Supported Cooperative Work (CSCW)* 22.2-3 (Sept. 2012), pp. 181–240. ISSN: 1573-7551. DOI: [10.1007/s10606-012-9172-4](https://doi.org/10.1007/s10606-012-9172-4).