

OOP KONFERENZ 2022-02-01

Hilfe, wir syncen!

INNOQ



LUCAS DOHMEN
@MOONBEAMLABS



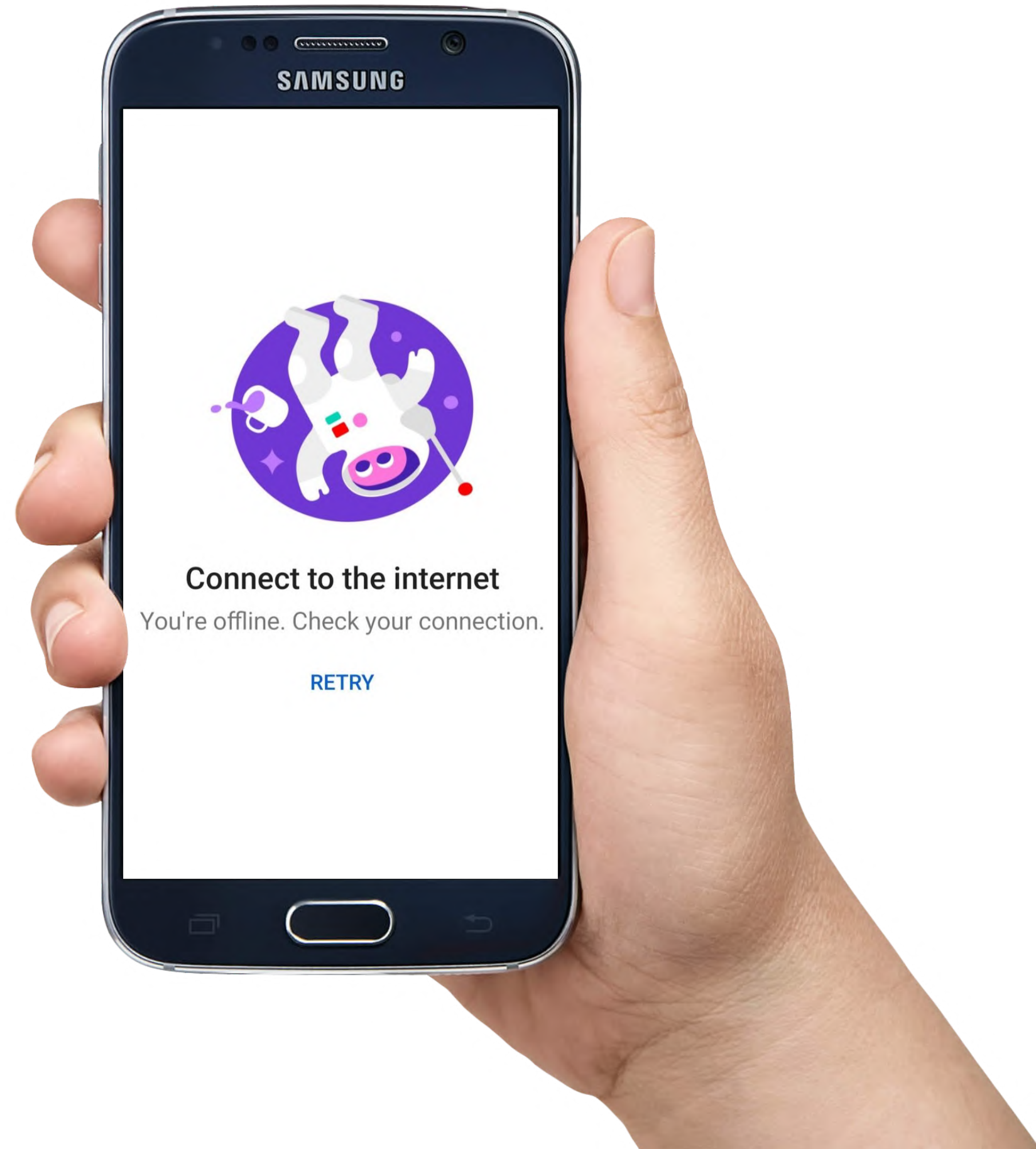
LARS HUPEL
@LARSR_H





Wenn alles in der Cloud ist ...

- ... sind wir abhängig vom Internet
- ... können wir nur eingeschränkt arbeiten
- ... schließen wir Menschen aus



Offline arbeiten?

- früher™ hat man auch nicht ständig Internet gehabt
- lokale Datenspeicherung, manuelle Synchronisierung



Offline-first ist Standard in der Entwicklung

- wir arbeiten gerne offline
- niemand möchte mehr Subversion benutzen
- Warum verlangen wir Konnektivität von unseren User:innen?





```
graph TD; A[Hello world!] --> B[Hello world]; A --> C[Hello, world!]
```

Hello
world!

Hello
world

Hello,
world!

Document1 - Microsoft Word

File Home Insert Page Layout References Mailings Review View

Spelling & Grammar Research Thesaurus Word Count Proofing

Translate Language Language

New Comment Delete Previous Next Comments

Track Changes Reviewing Pane Tracking

Accept Reject Previous Next Changes

Compare Compare Block Authors Restrict Editing Protect

TITLE

8 Support for Track Changes in the OpenDocument (ODF) format

Visualization and printing of comments:

Improvements to printing of comments inside the margin: Bug 36815. Please also see the OpenOffice.org bugs: i#94514, i#767, i#61644.

Ability to apply comments to text selections: Bug 38244 (This is partly done and will be integrated in LibreOffice 3.7. Limitation: Text ranges cannot span over more than one paragraph)

User interface for comments: Bug 36896, Bug 38247, Bug 38246. Further design ideas of the Comments Ruler Control are collected

1 **Introduction**

2 Related issues and requirements

3 Solution Proposals

4 Further ideas for improvements

5 Google Summer of Code 2009: Improve Writer's compare function, 7 Related OpenOffice/LibreOffice extensions

9 Manuals on using Writer's existing track changes features

10 Press Coverage and blogs on track changes

11 Please also see

Track changes is a possibility for Writer users to keep track of the changes a user makes or other users have made to a document. All changes are recorded and are visualized in order to ease the

This. It also aims to collect some ideas on potential improvements in this area in Writer.

The standard view always displays the most recent version and changes are not marked. However, it is possible to access the So-called "History" which makes it possible to easily navigate on a time scale through all changes made by all authors. Insertions are underlined in the colour of the respective co-author. Deletions are displayed as "Interactive markers" that display the deleted text if the mouse

Deleted: Track changes¶ Contents¶ [hide]

Formatted: Font: 18 pt, Bold

Moved (insertion) [1] Go

Moved (insertion) [2] Go

Markup Area

Comment [g1]: This is a comment, but there is nothing to say

Deleted: ¶ 6 Competitive Analysis¶

Moved up [1]: 8 Support for Track Changes in the OpenDocument (ODF) format¶ Go

Deleted: .page aims to collect some information on the track changes feature of Writer.

Page: 1 of 6 Words: 1,390 English (U.S.) 116%

$$\frac{dN}{dt} = \frac{1}{qV_{act}} - q_0(N-N_0)(1-\epsilon S)S + \frac{N_e}{\tau_n} - \frac{N}{\tau_p}$$

$$\frac{dS}{dt} = T_0 q_0(N-N_0)(1-\epsilon S)S + \frac{I_0 N}{\tau_n} - \frac{S}{\tau_p}$$

$$\frac{S}{P_k} = \frac{\tau_p \chi_0}{V_{act} n_{ph}} = 0$$

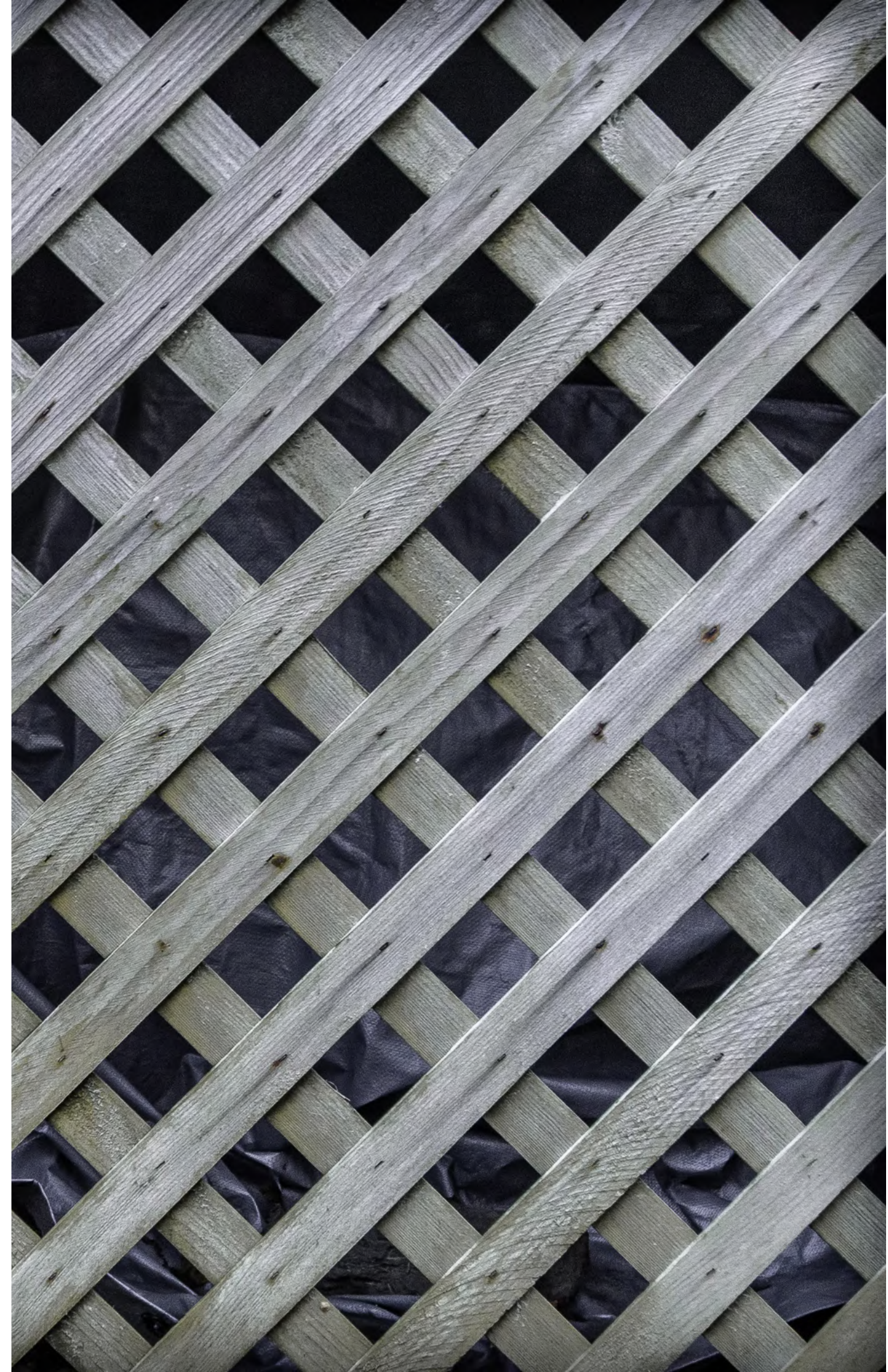
$$S \leq \frac{1}{\epsilon}$$

$$N = N_0$$

$$P_f = (m$$

Lattices

- mathematische Struktur
- abstrakte Definition einer Ordnungsrelation



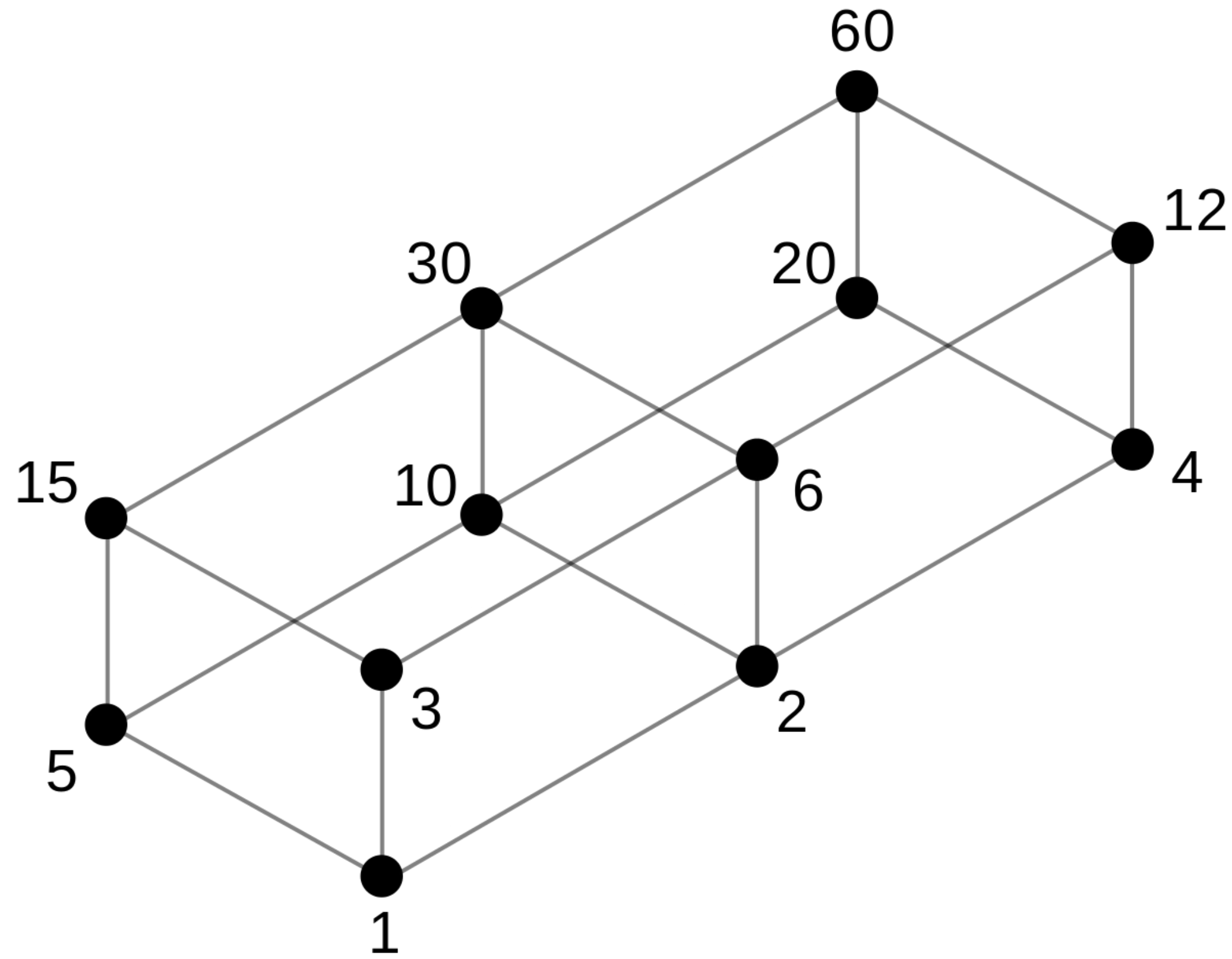
```
interface Lattice<T> {  
    join(left: T, right: T): T;  
    lteq(left: T, right: T): boolean;  
}
```

```
// 1. join(x, y) == join(y, x)
```

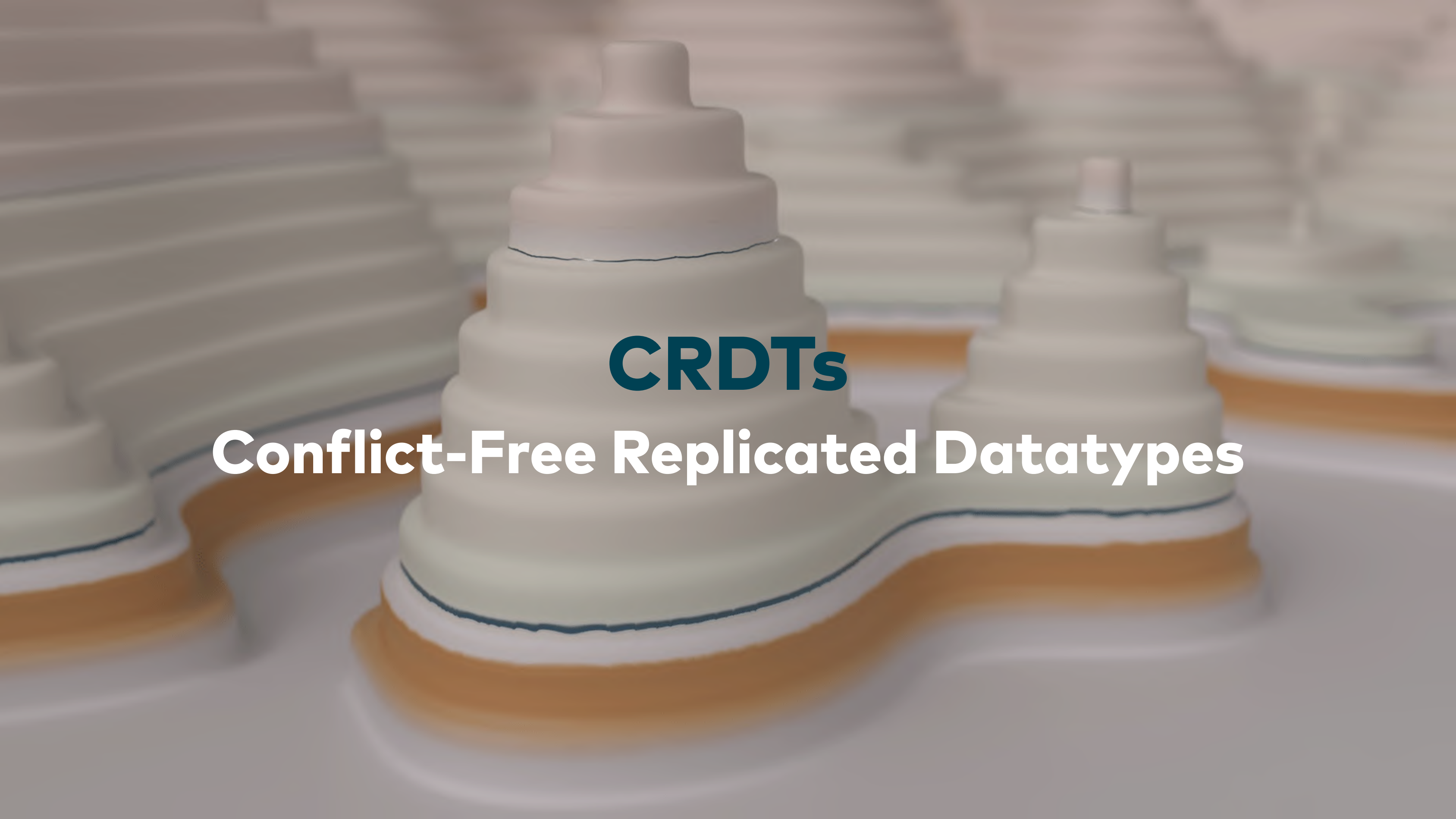
```
// 2. join(x, x) == x
```

```
// 3. join(x, join(y, z)) == join(join(x, y), z)
```

```
// ...
```



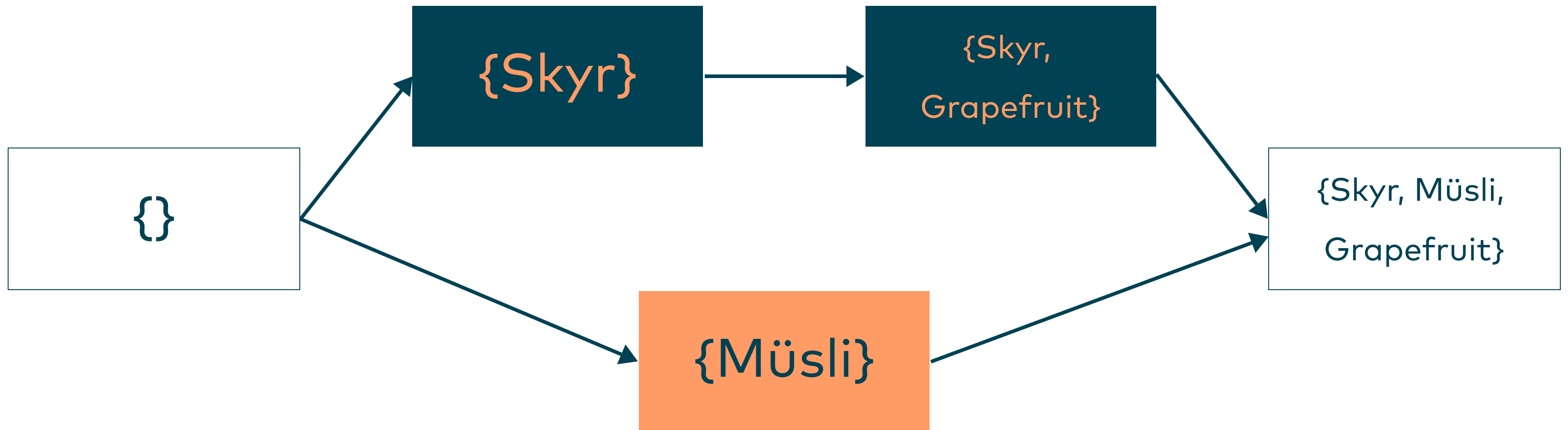
```
const divisionLattice: Lattice<number> = {  
  join(left: number, right: number) {  
    return leastCommonMultiple(left, right);  
  },  
  lteq(left: number, right: number) {  
    return right % left == 0;  
  }  
}
```



CRDTs

Conflict-Free Replicated Datatypes

Beispiel: G-Sets

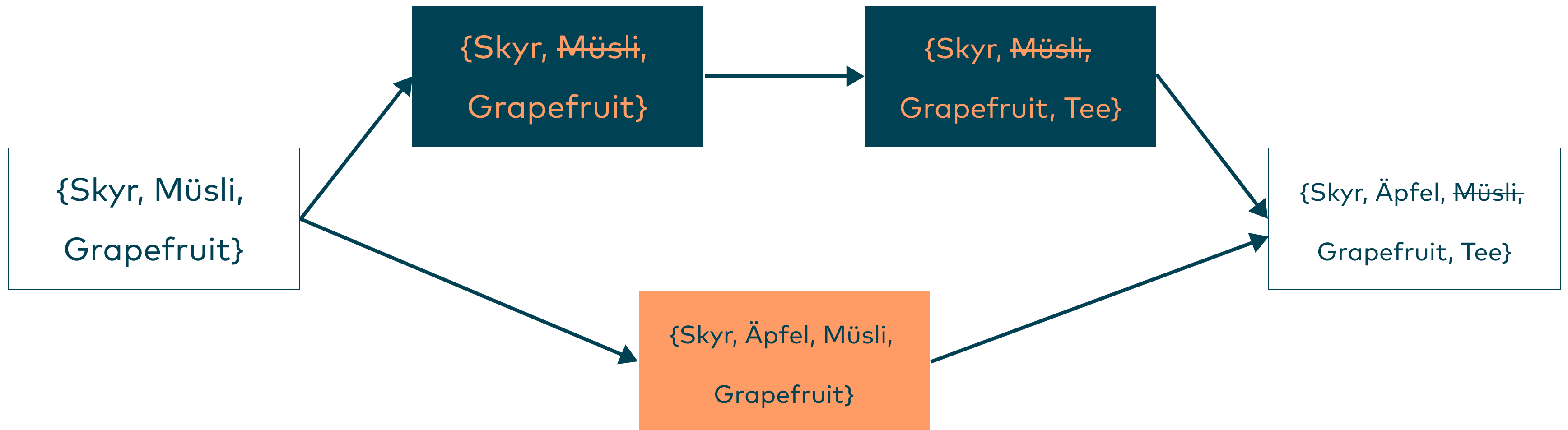


Tombstones

- Löschen ist nicht monoton 😞
- Lösung: als „gelöscht“ markieren



Beispiel: 2P-Sets

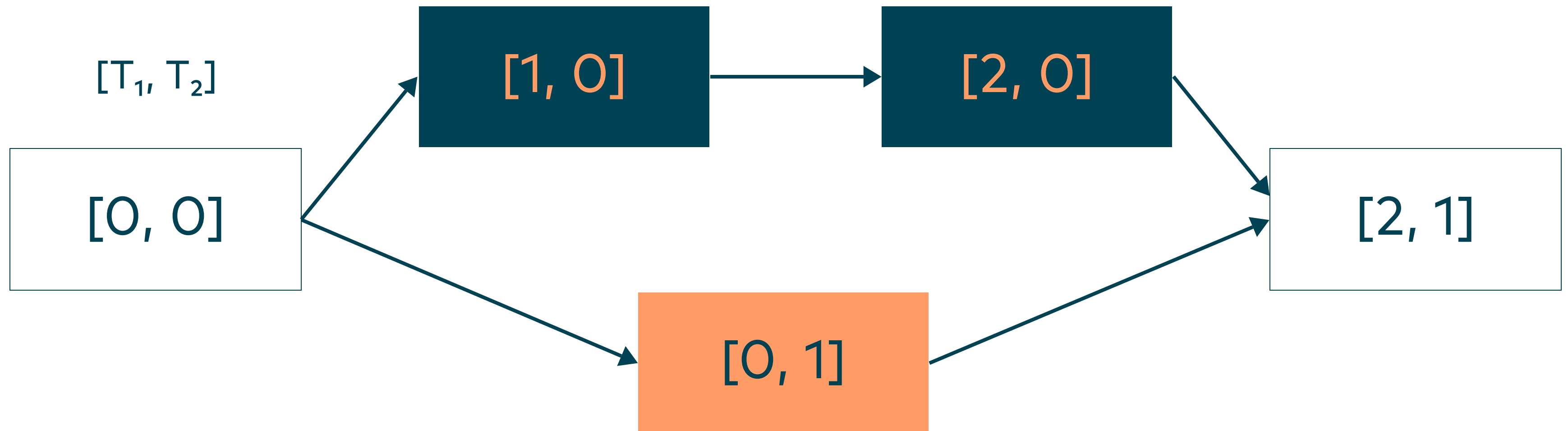


Vector Clocks

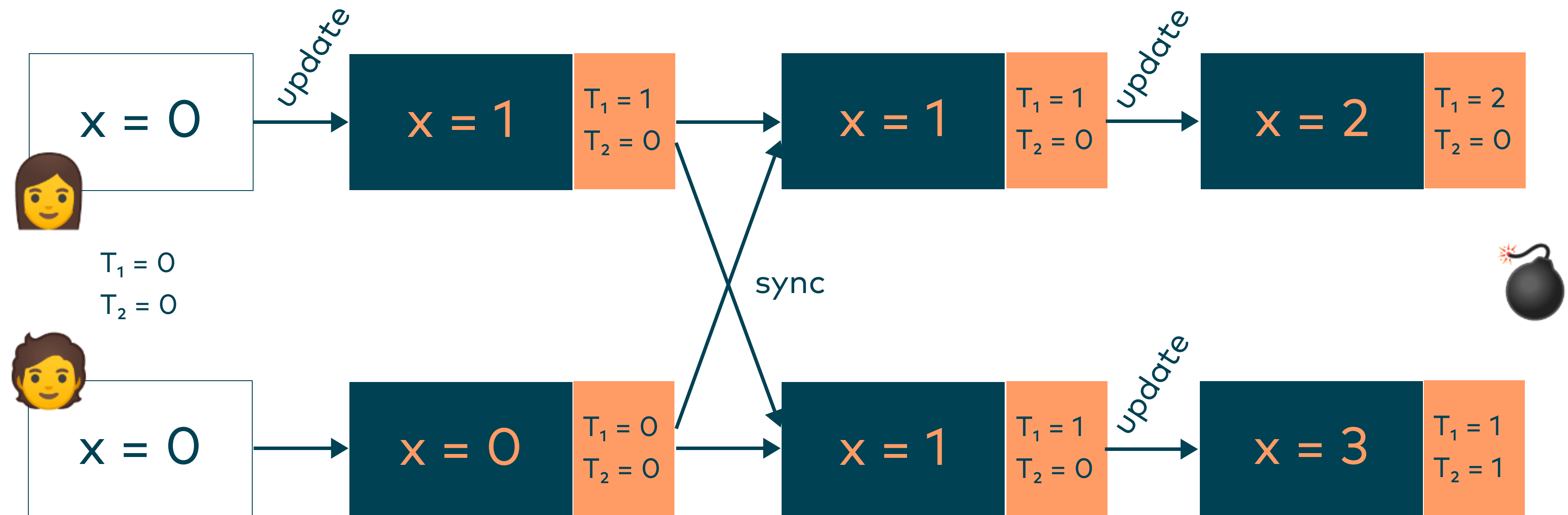
- jedes der n Geräte bekommt einen eigenen Zeitstempel
- ein Gesamt-Zeitstempel setzt sich aus n Geräte-Zeitstempeln zusammen
- repräsentiert letzten bekannten Zustand



Vector Clocks



Beispiel: MV-Register



**"Any two object replicas of a CvRDT
eventually converge, assuming the system
transmits payload infinitely often between
pairs of replicas over eventually-reliable
point-to-point channels."**

```
var d=this,e=this
```

```
$.getJSON(c.router.themePath+"/api/v1/devices",function(r){
```

```
if(r.length>0){
```

```
if(r[0].status=="ready"),a(document.body,"ready"),c.router.selectedDevice=r[0].id
```

```
},this.undelegateEvents)
```

```
$("#collapse").toggleClass("collapsed")
```

```
$("#togglePreviewDeviceButtons").click(function(){
```

```
if($("#collapse").hasClass("collapsed")){
```

```
maybeRequestFilesystem()
```

```
},this.undelegateEvents)
```

```
$.extend(this,{
```

```
listenTo(c.collection,"change",function(){
```

```
length),c.announceSearchResults(r.length)
```

```
function(){c.overlay.toggleClass("hidden")}
```



Automerge

Automerge

- JavaScript-Bibliothek für Browser & Node.js
- bietet als Datentyp JSON-Dokumente
- 💡 JSON-Dokument sind CRDTs



Verwendung

```
doc1 = Automerge.from({ todos: [] })
```

```
doc2 = Automerge.change(doc1, 'Add muesli', doc => {  
  doc.todos.push({ text: 'Müsli', done: false })  
})
```

```
doc3 = Automerge.change(doc1, 'Add skyr', doc => {  
  doc.todos.push({ text: 'Skyr', done: false })  
})
```

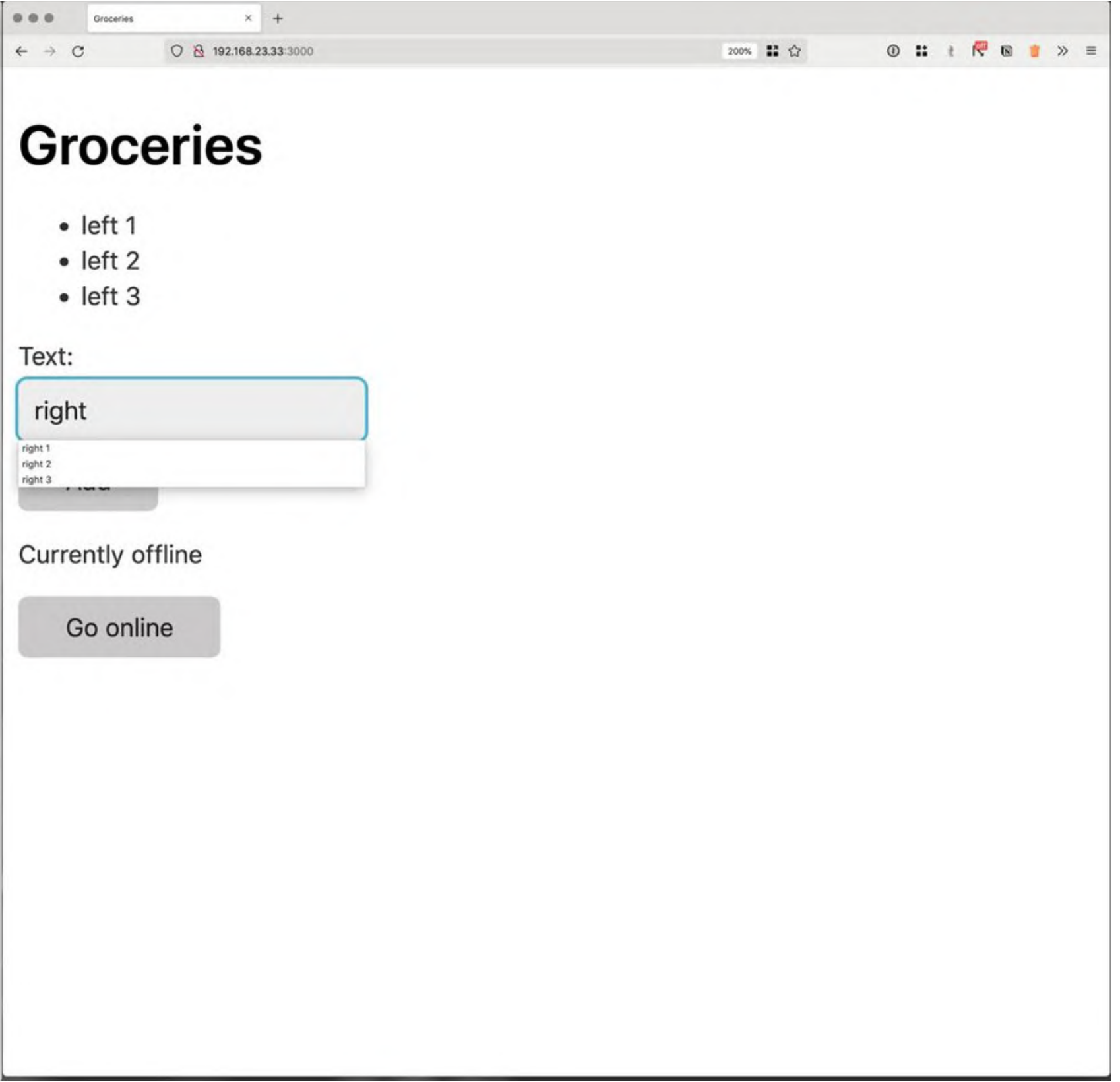
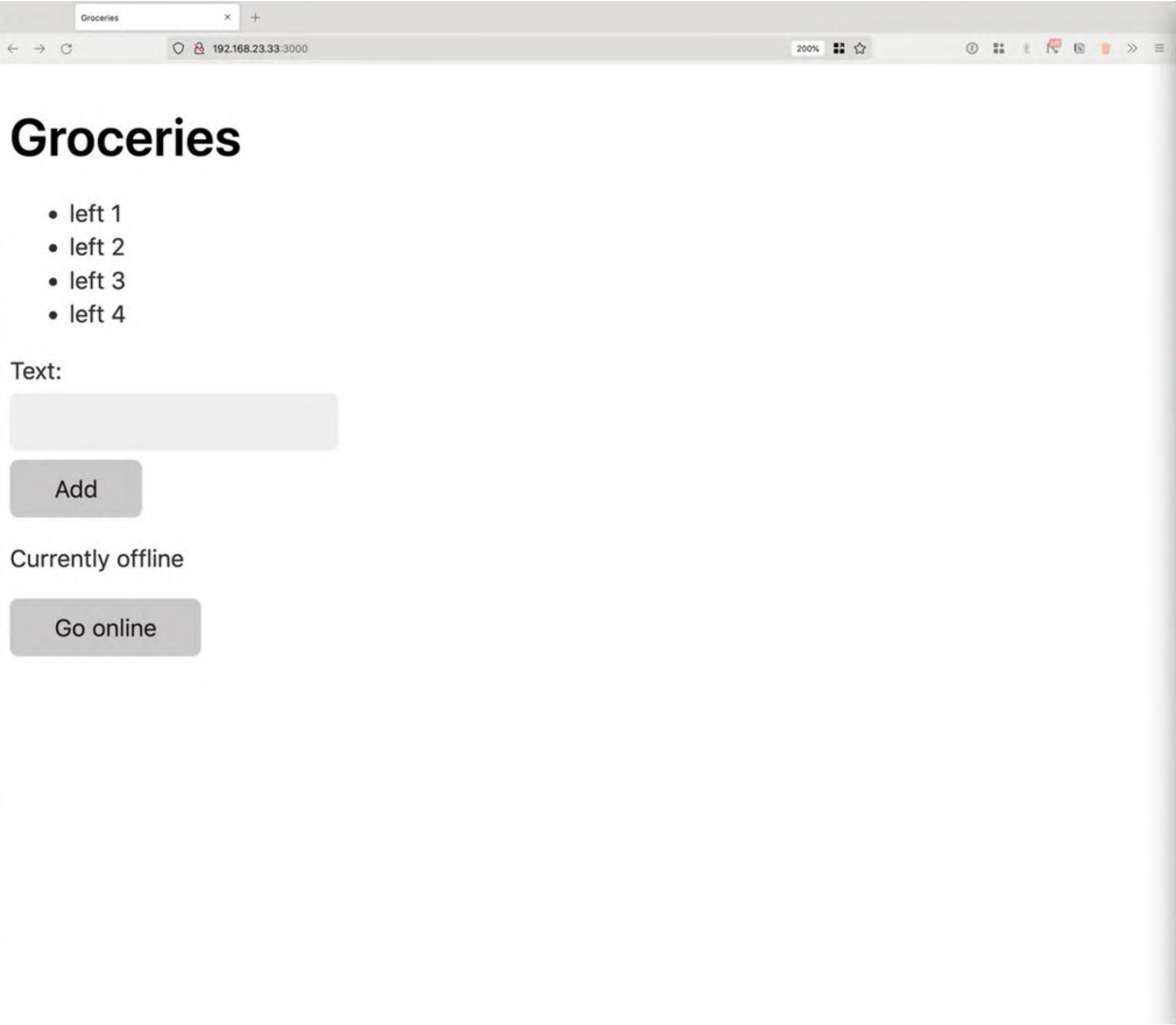
```
Automerge.merge(doc2, doc3)
```

```
// enthält Spuren von Müsli & Skyr
```

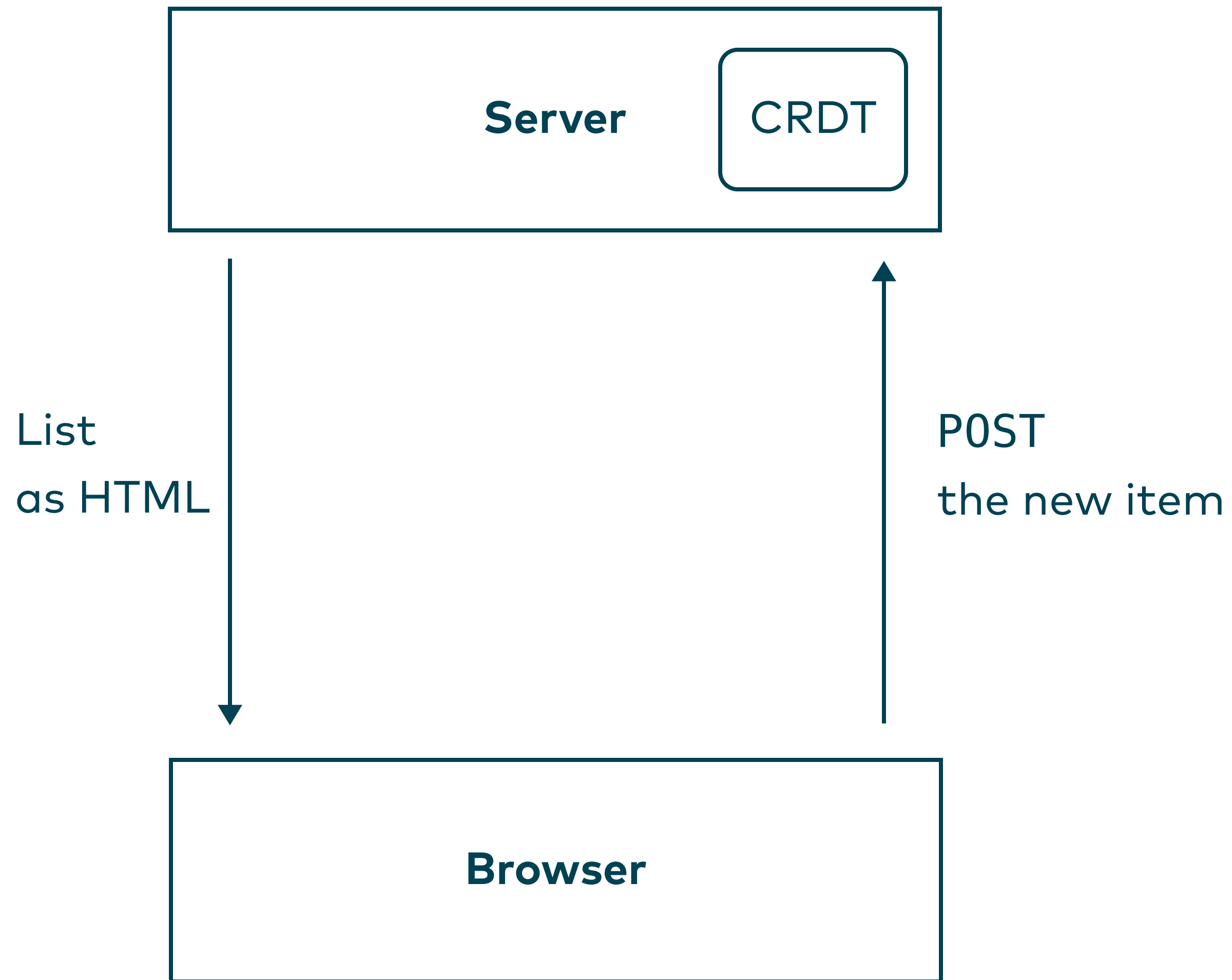


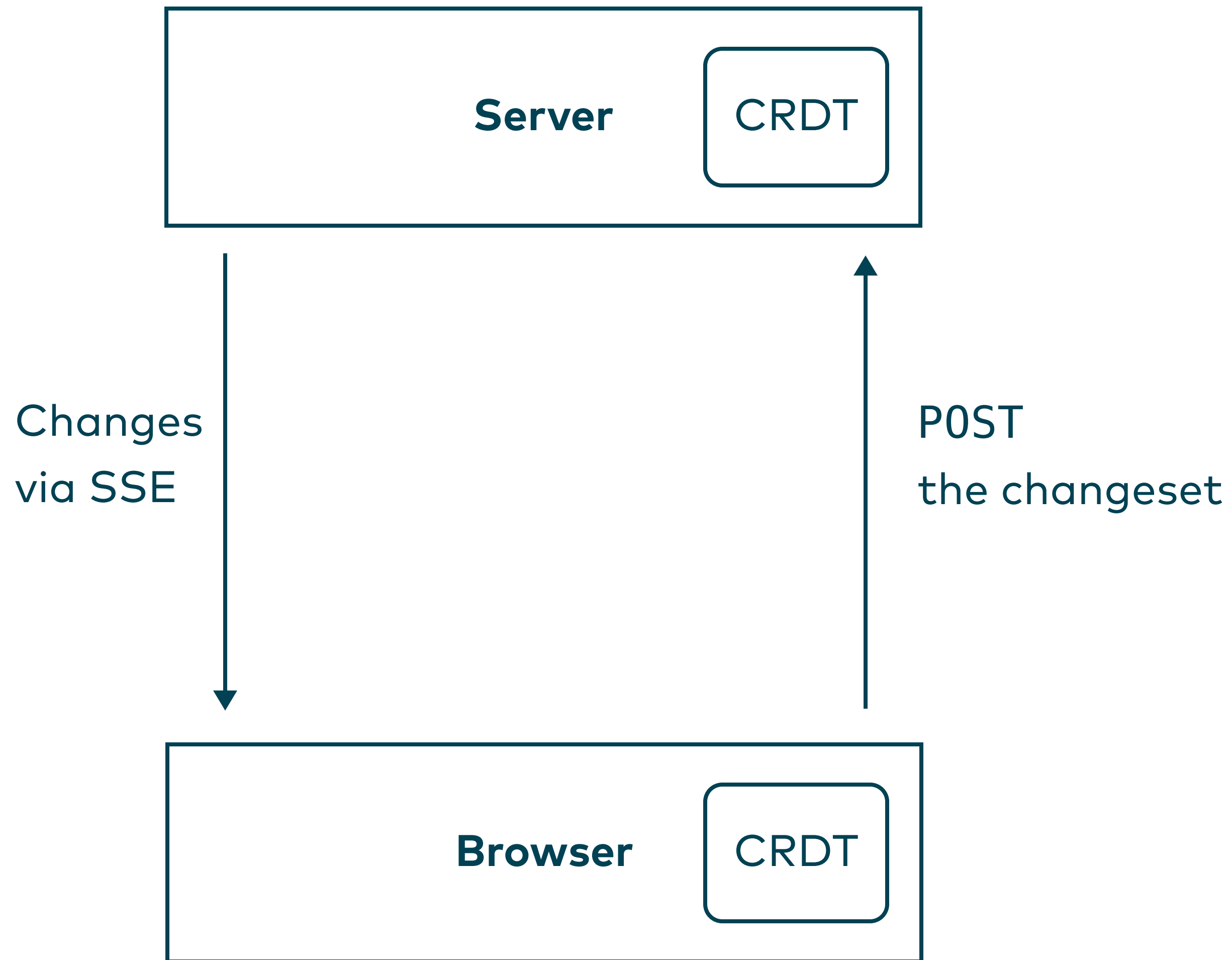
Demo

Eine Einkaufsliste



<https://github.com/innoq/groceries>





Subscribe via SSE

```
const source = new EventSource( "/stream" );  
source.addEventListener( "message", (ev) => {  
    list.applyChanges( ev.data );  
});
```

Push Changes

```
class List {  
  constructor() {  
    // ...  
    this.changes = [];  
    this.interval =  
      setInterval(this.pushChanges.bind(this), 1000);  
  }  
}
```

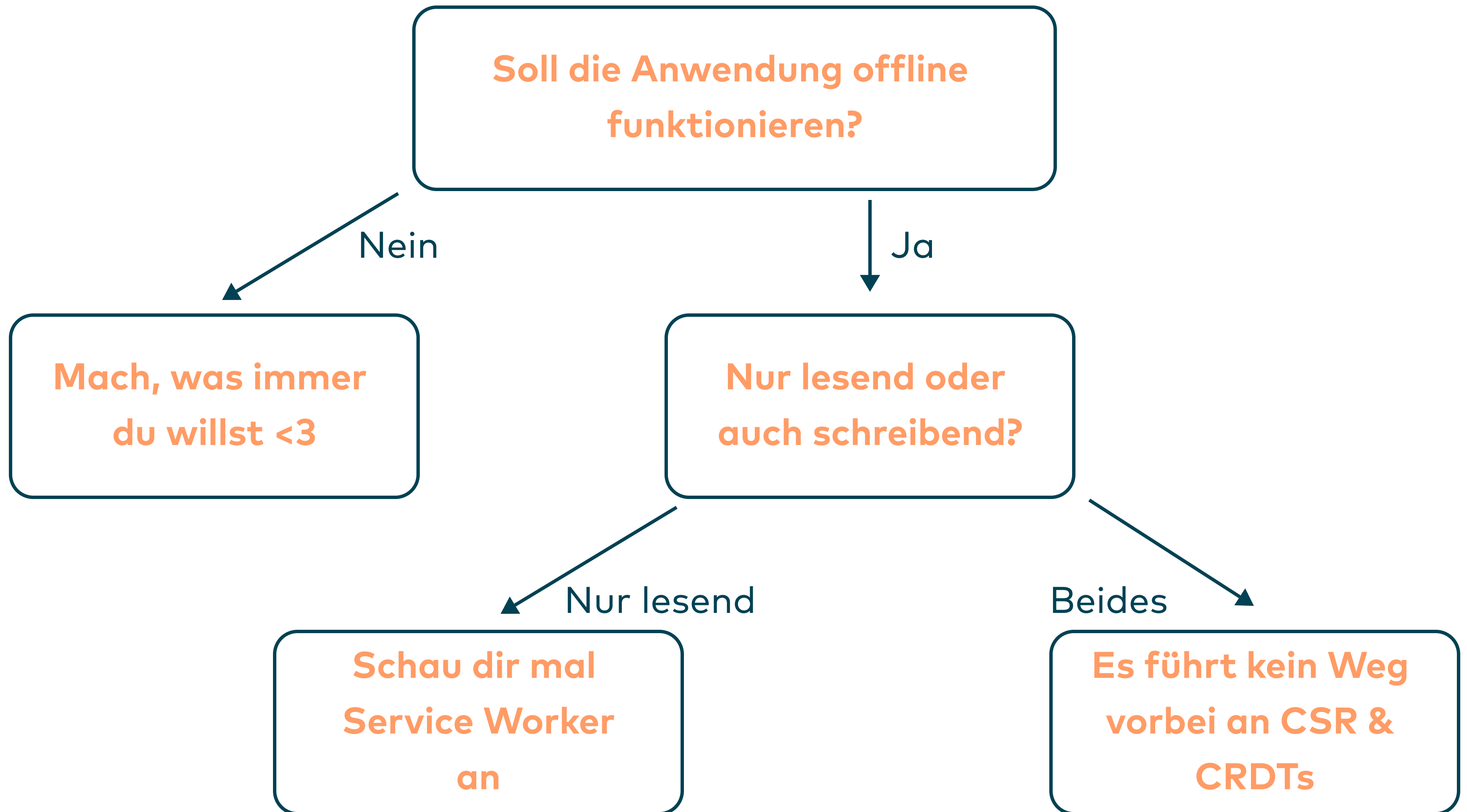
Push Changes

```
async pushChanges() {  
  try {  
    await fetch("/", {  
      method: "POST",  
      headers: { "Content-Type": "application/json" },  
      body: JSON.stringify({ changes: this.changes }),  
    });  
    this.changes = [];  
  } catch(e) { /* ... */ }  
}
```

Weitere Features

- mehr Datentypen:
 - Counter
 - Text (kollaboratives Editing)
 - Peritext
 - Table
- P2P-Anwendungen mit Hypermerge





Q&A



Lucas Dohmen

lucas.dohmen@innoq.com

@moonbeamlabs

Lars Hupel

lars.hupel@innoq.com

@larsr_h

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 333660

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Hermannstr. 13
20095 Hamburg

Erftstr. 15-17
50672 Köln

Königstorggraben 11
90402 Nürnberg

