

REACT VIENNA / 2022-06-21

Mayday, we're syncing!

INNOQ



LUCAS DOHMEN
@MOONBEAMLABS



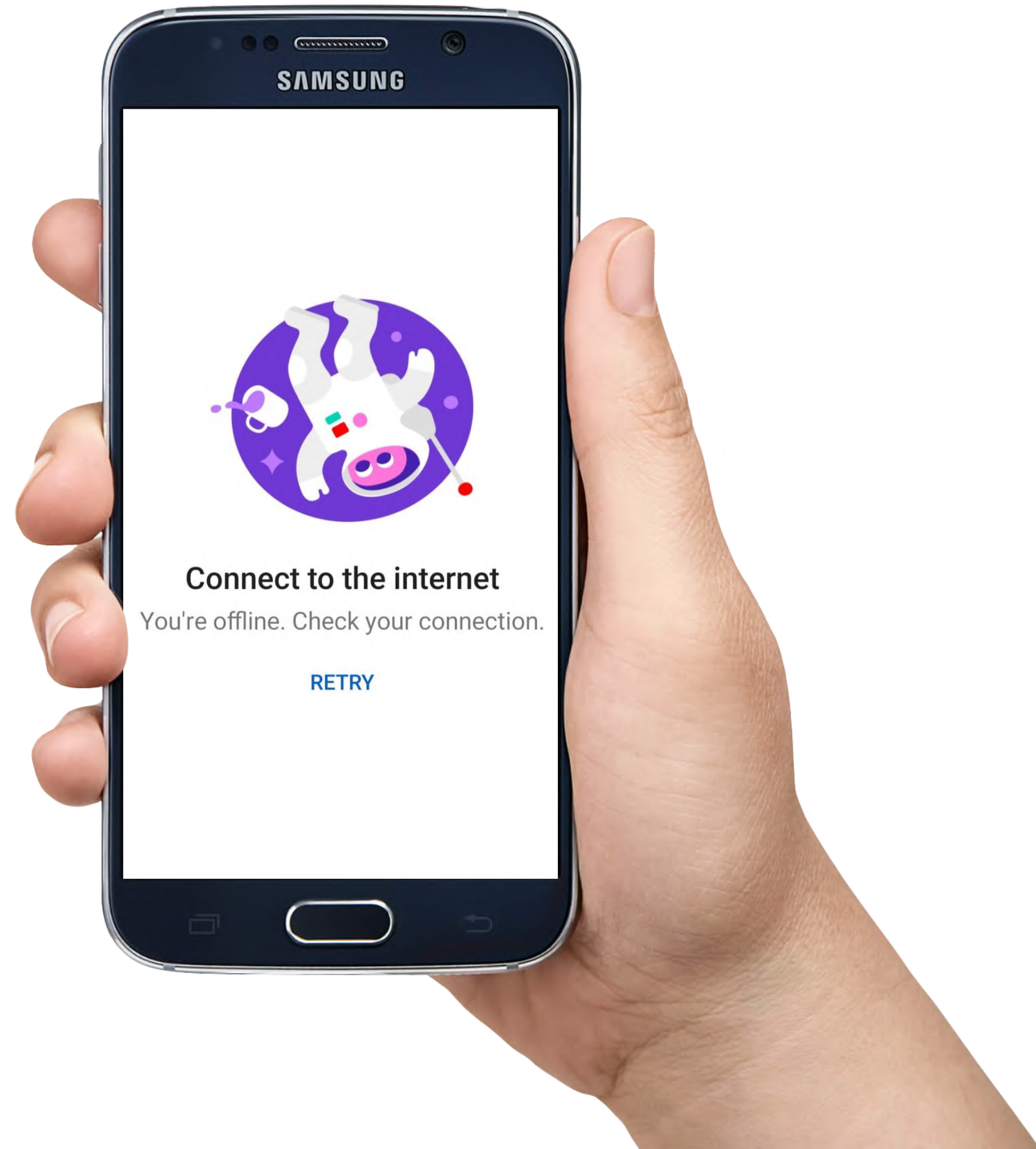
LARS HUPEL
@LARSR_H





When everything's in the cloud

- ... we are dependent on the Internet
- ... we are limited in how we can work
- ... we are excluding people



Working Offline?

- in the Olden Days, we didn't always have a connection
- local file storage, manual sync

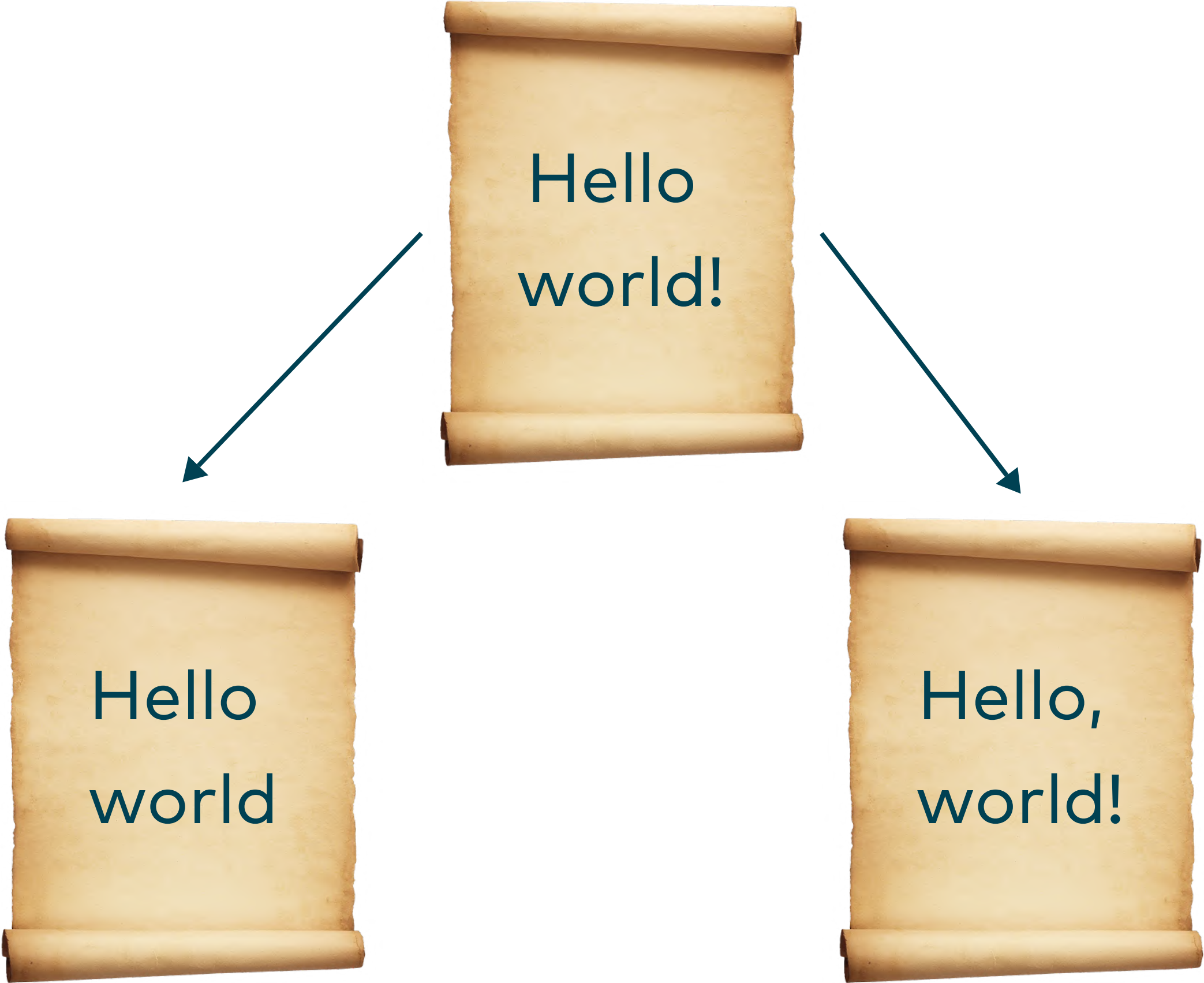


Offline-first is the standard for development

- we like working offline
- no one wants to use Subversion nowadays
- why do we ask our users for connectivity?







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

Diagram illustrating a branching structure. A top scroll contains the text "Hello world!". Two arrows point from this scroll to two bottom scrolls. The left bottom scroll contains the text "Hello world", and the right bottom scroll contains the text "Hello, world!".

Hello
world!

Hello
world

Hello,
world!

RD Merge Revisions for D:\Projects\Git\Demo\Nancy\src\Nancy\DependencyContextAssemblyCatalog.cs

↑

↓

Highlight words

5 changes. 1 conflict

Local Changes (Read-only)		Result		Changes from Server (revision 9d4b6795...
ⓘ using System.Reflecti using Microsoft.Exter	7	7	using System.Reflection;	7
	8	8	using Microsoft.Extension	8
	×	9		9
/// <summary>	10	10	/// <summary>	10
/// Default implement	11	11	/// Default implementati	11
/// retrieving <see c	12	12	/// retrieving <see cref	12
/// </summary>	13	13	/// </summary>	13
public class Depend	14	14	public class DependencyC	14
{	15	15	{	15
private static re	16	16	private static readon	16
private readonly	17	17	private readonly Dep	17
	18	18		18
/// <summary>	19	19	/// <summary>	19
/// Initializes a	×	20	/// Initializes a new	20
/// using <see cl	21	21	/// </summary>	21
/// </summary>	22	22	public DependencyCon	22
public Dependency	23	23	{	23
: this(Assembl	×	24	var entryAssembl	24
{	25	25		25
}	×	26	this.dependencyC	26
	27	27	}	27
/// <summary>	28	28	/// <summary>	28
/// Initializes a	29	29	/// Gets all <see cref	29
/// using <param	30	30	/// </summary>	30
/// </summary>	31	31	/// <returns>An <see	31
public Dependency	32	32	public IReadOnlyColl	32
{	33	33	{	33
dependencyCor	34	34	var results = new	34
}	35	35		35

Accept Left

Accept Right

Apply

Abort

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%

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

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

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

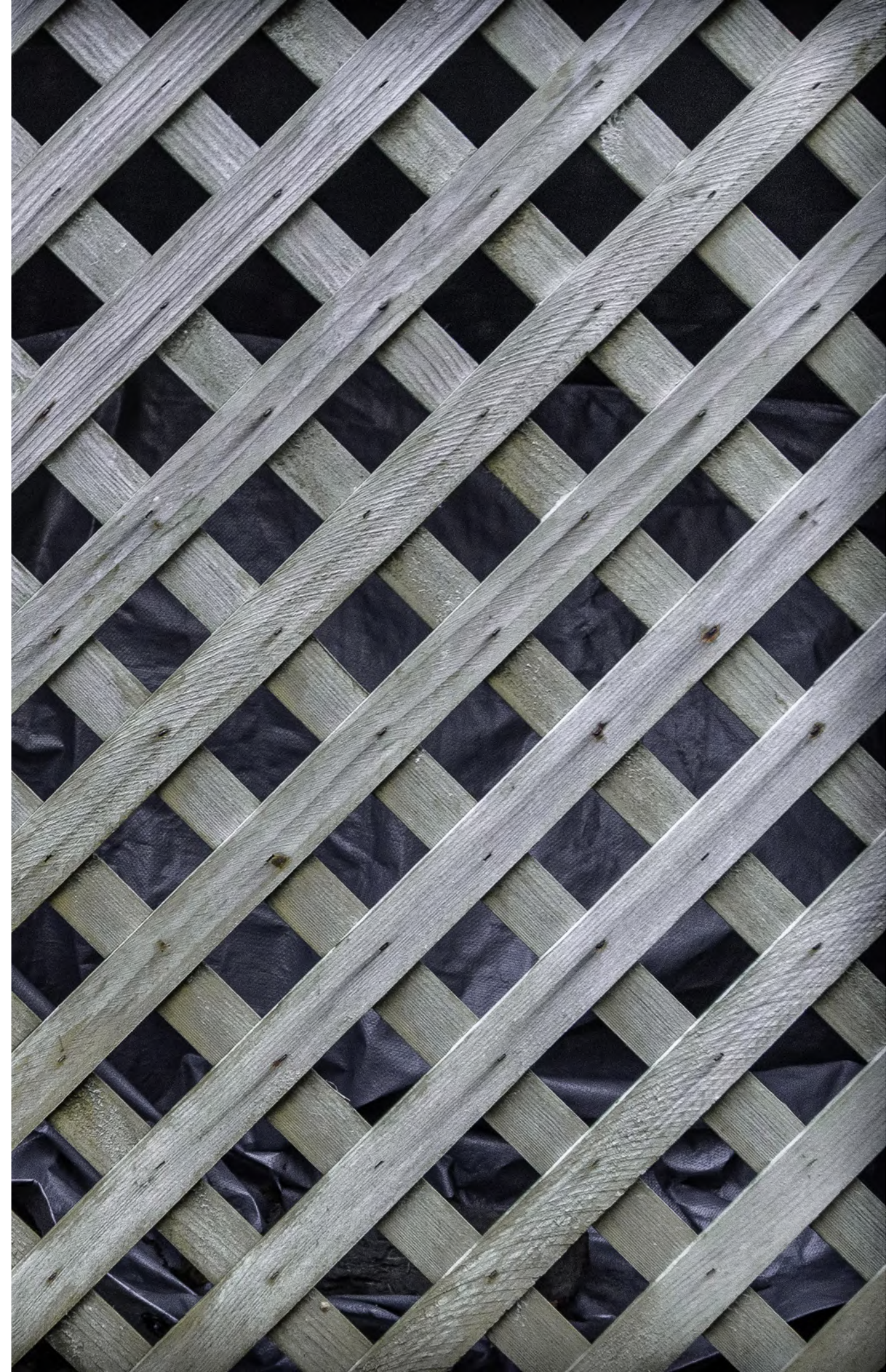
$$S \leq \frac{1}{\epsilon}$$

$$N = N_0$$

$$P_t = (m$$

Lattices

- mathematical structure
- abstract definition of ordering



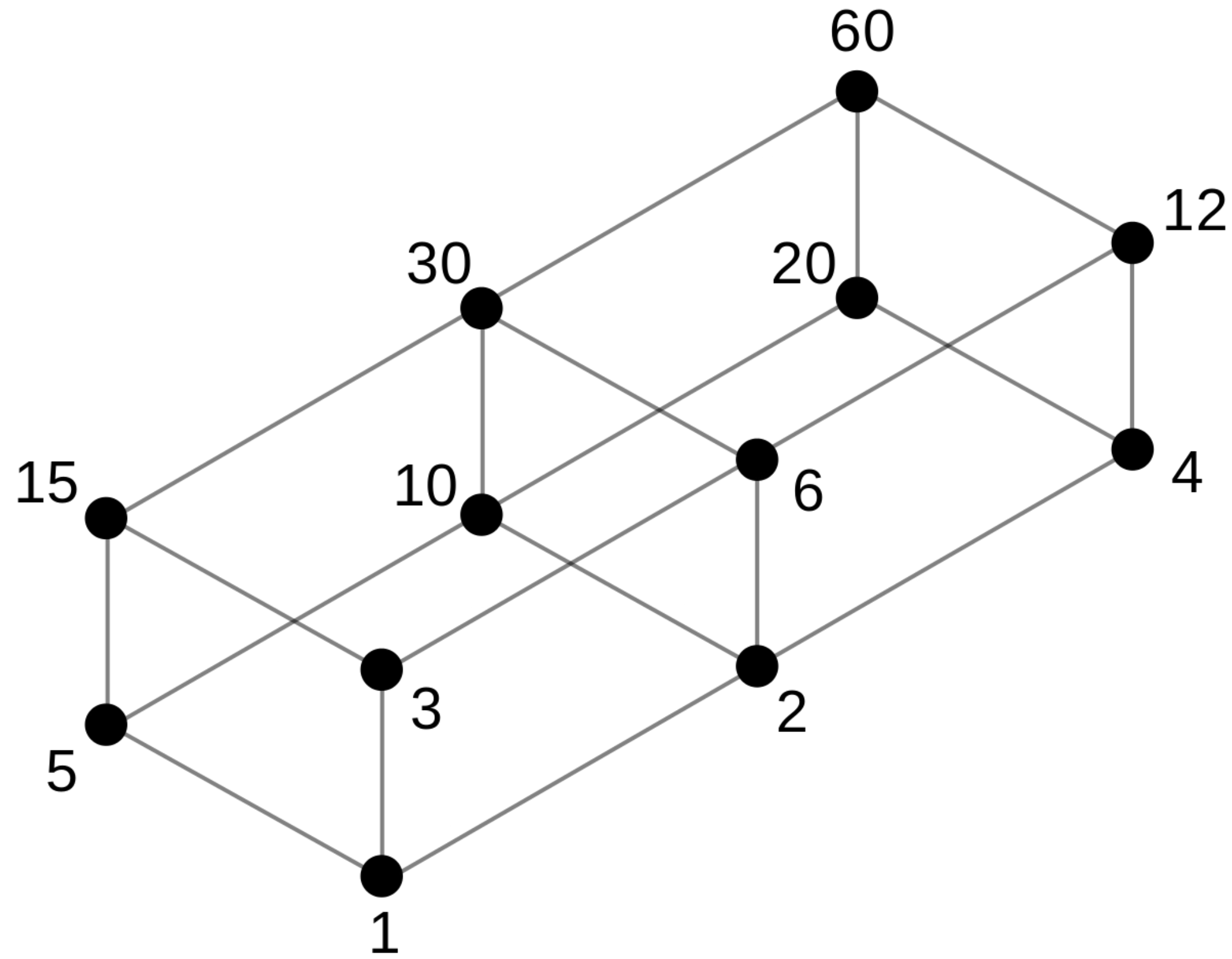

```
interface Lattice<T> {  
    join(left: T, right: T): T;  
    lteq(left: T, right: T): boolean;  
}
```

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

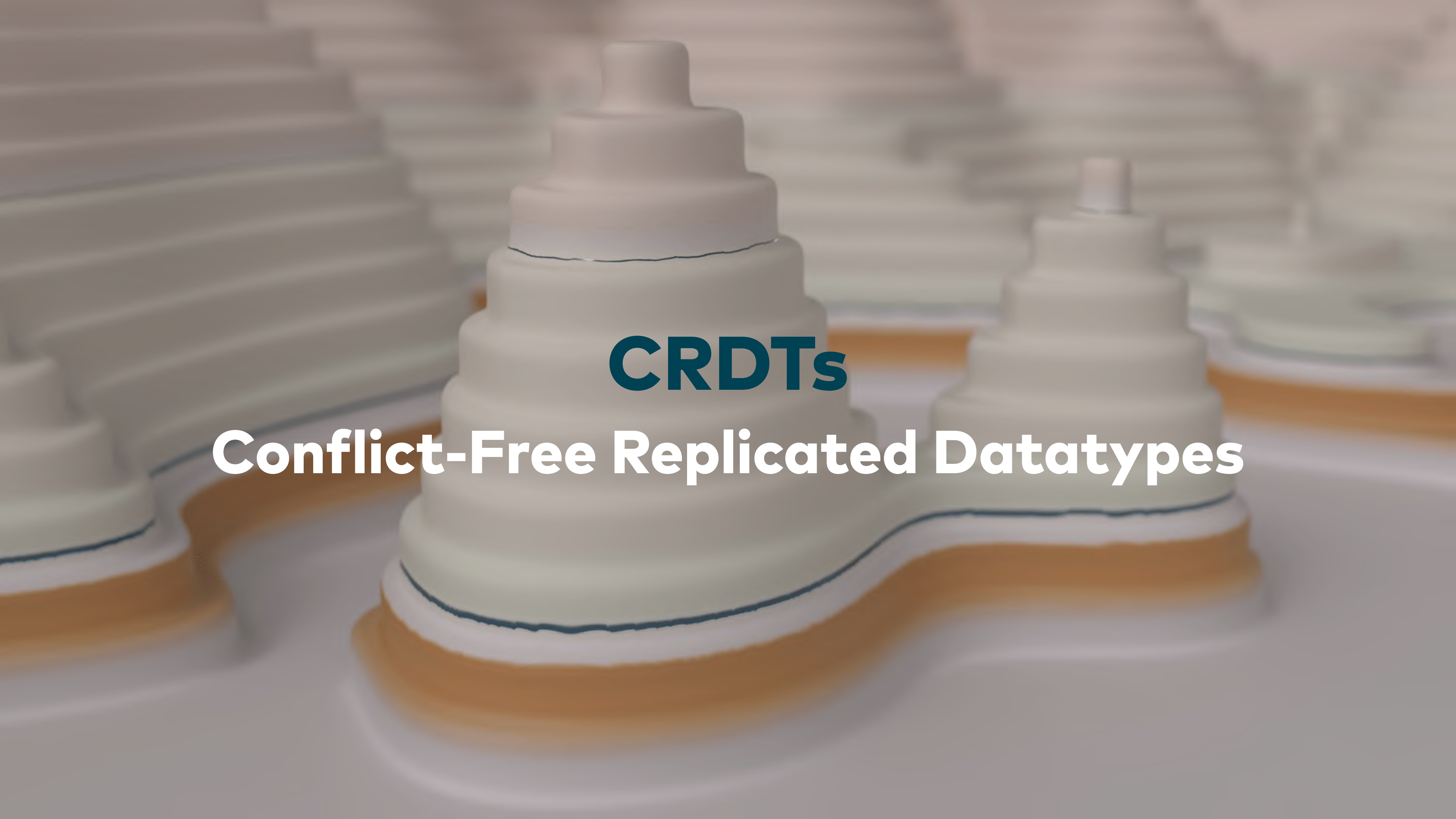
```
// 2. join(x, x) == x
```

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

```
// ...
```

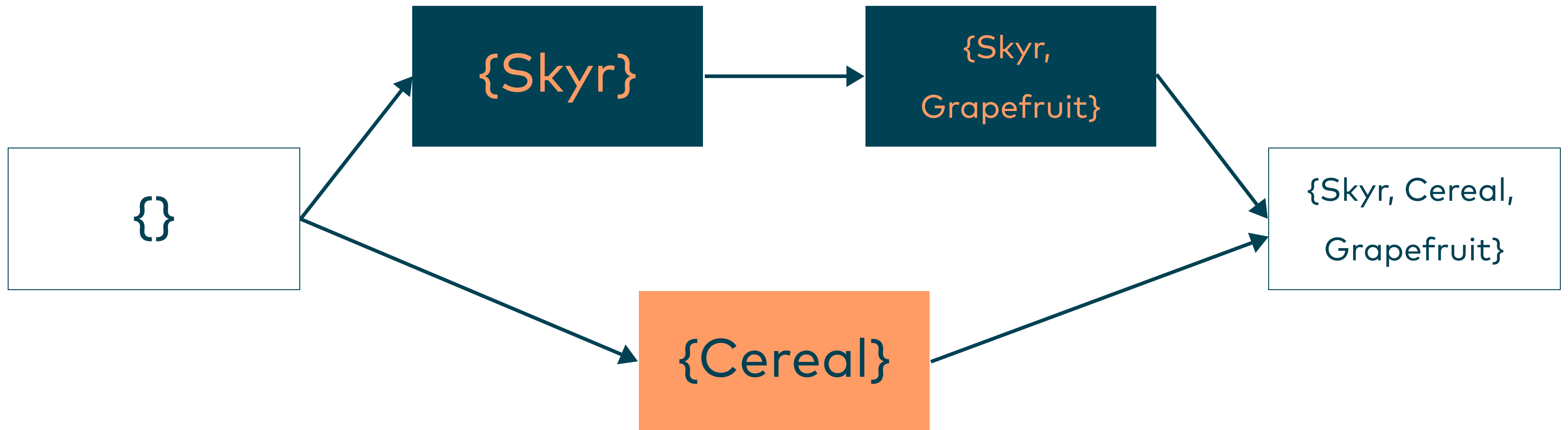

```
const divisionLattice: Lattice<number> = {  
  join(left: number, right: number) {  
    return leastCommonMultiple(left, right);  
  },  
  lteq(left: number, right: number) {  
    return right % left == 0;  
  }  
}
```

CRDTs

Conflict-Free Replicated Datatypes

Example: G-Sets

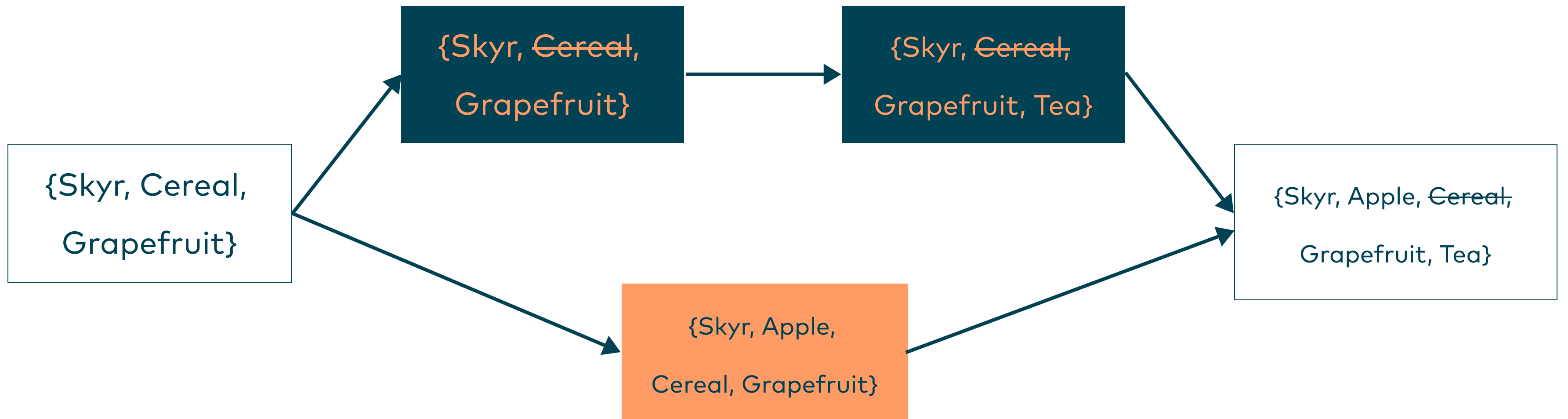


Tombstones

- deletion is not monotonic 😞
- solution: mark as deleted



Example: 2P-Sets

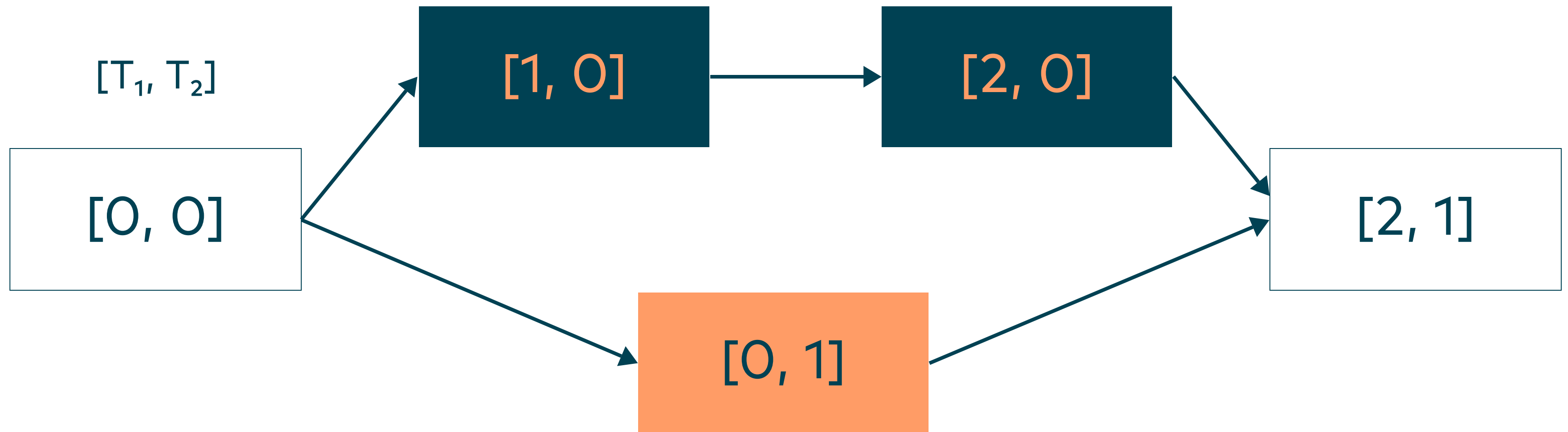


Vector Clocks

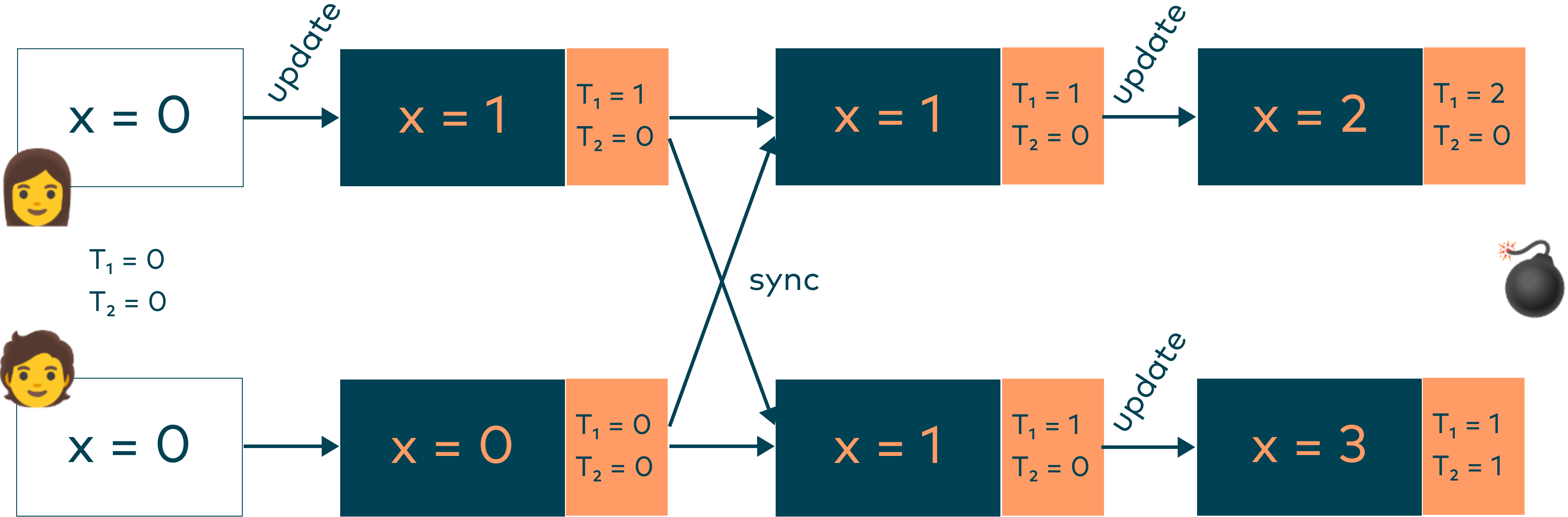
- each of the n devices gets its own timestamp
- the global timestamp for all devices is composed of those n timestamps
- it represents the last known state



Vector Clocks



Example: 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("active")}
```



Automerge

Automerge

- JavaScript Library for Browser & Node
- offers JSON documents as datatype
 - 💡 they are CRDTs



Usage

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

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

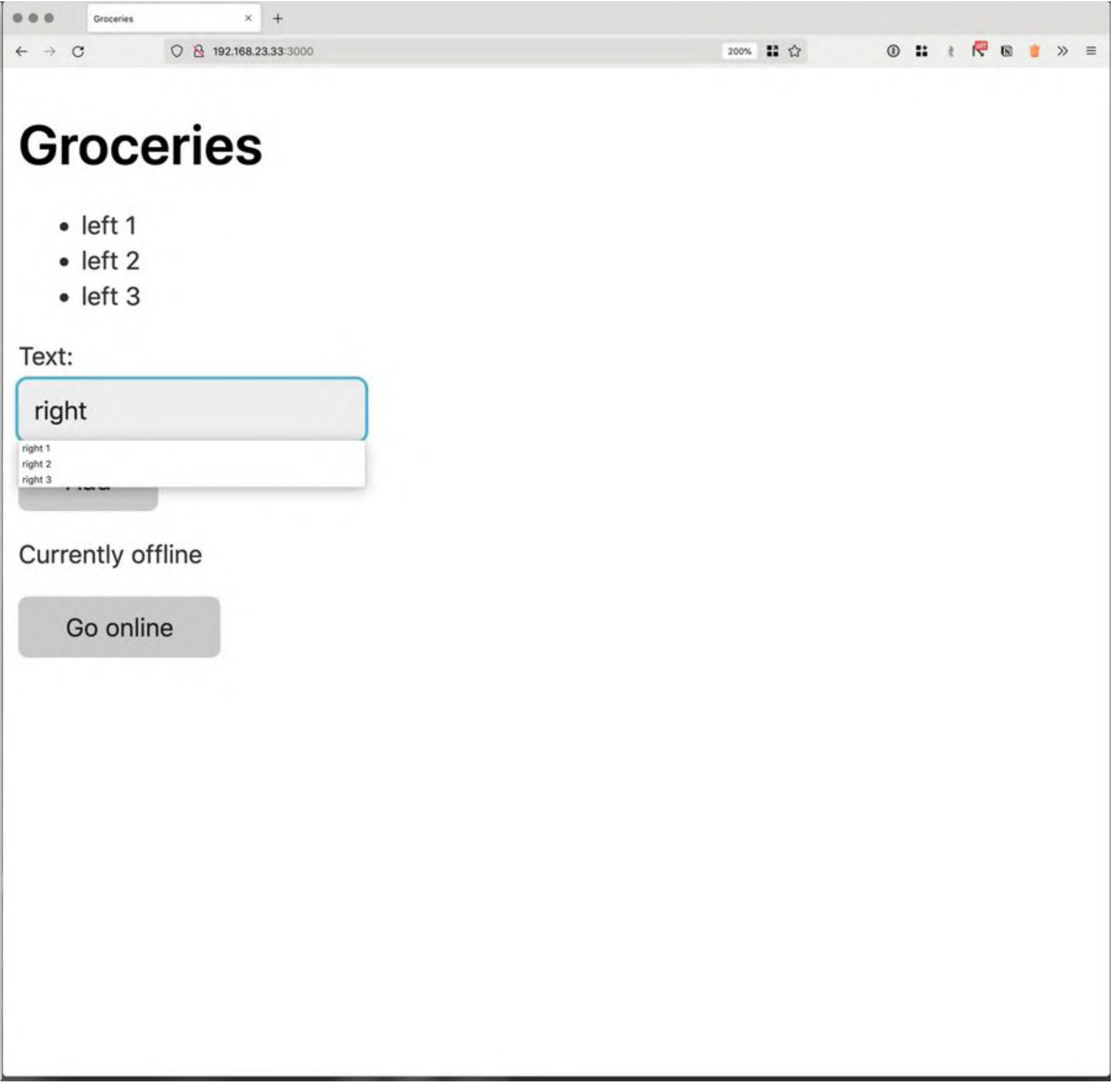
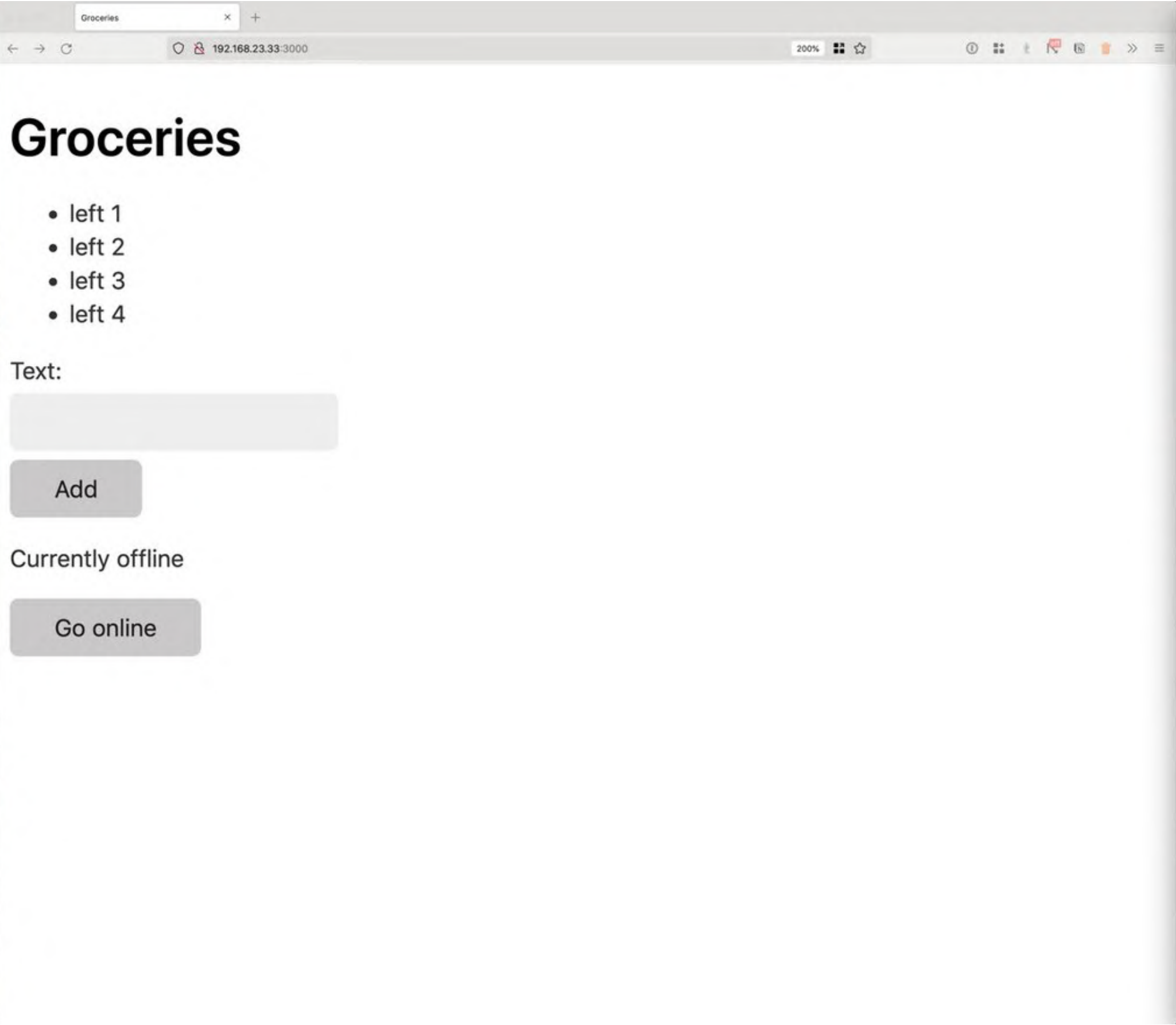
```
doc3 = Automerge.change(doc1, 'Add skyr', doc => {  
  doc.todos.push({ text: 'Skyr', done: false })  
})
```

```
doc4 = Automerge.merge(doc2, doc3)  
// contains traces of cereal & skyr
```

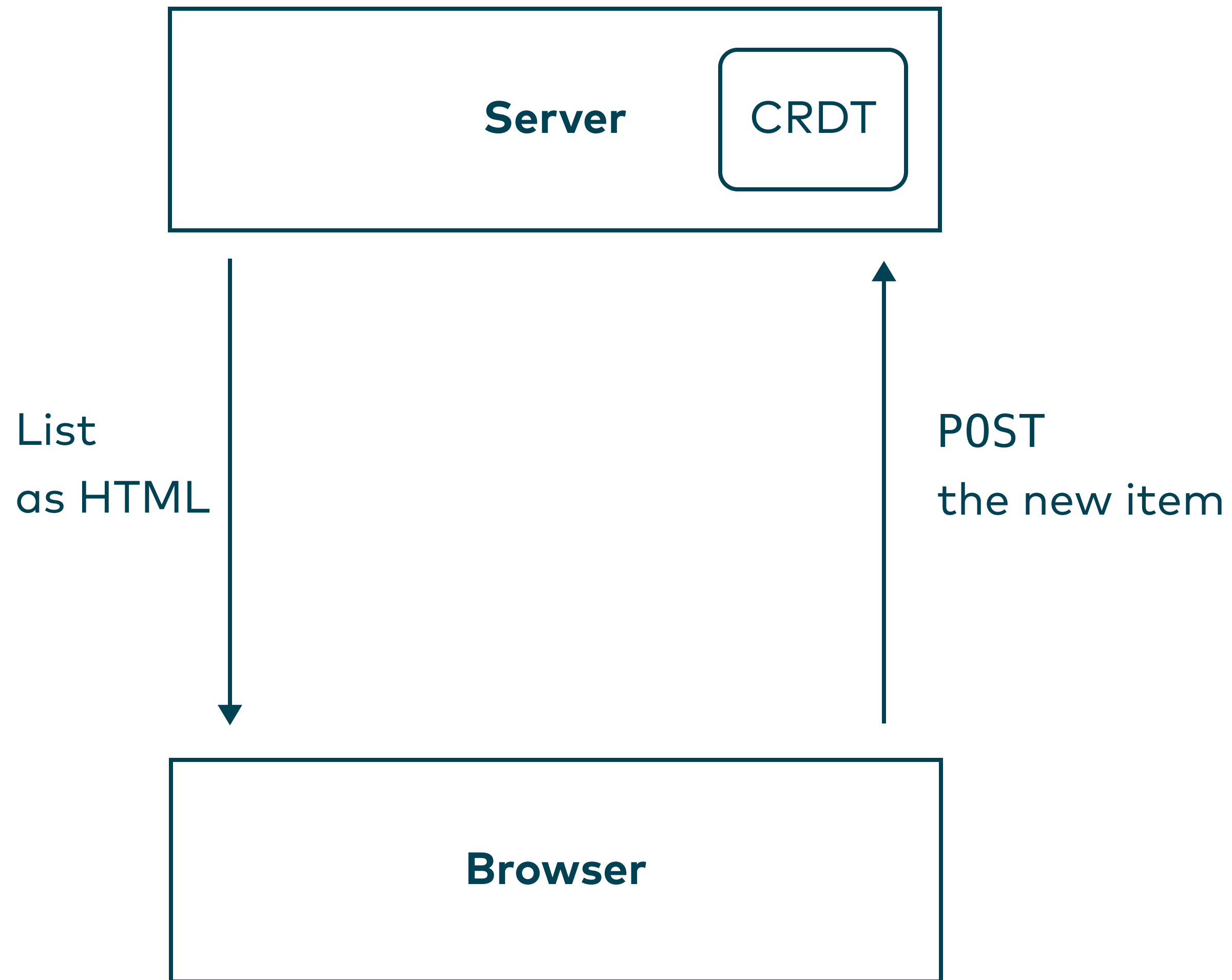


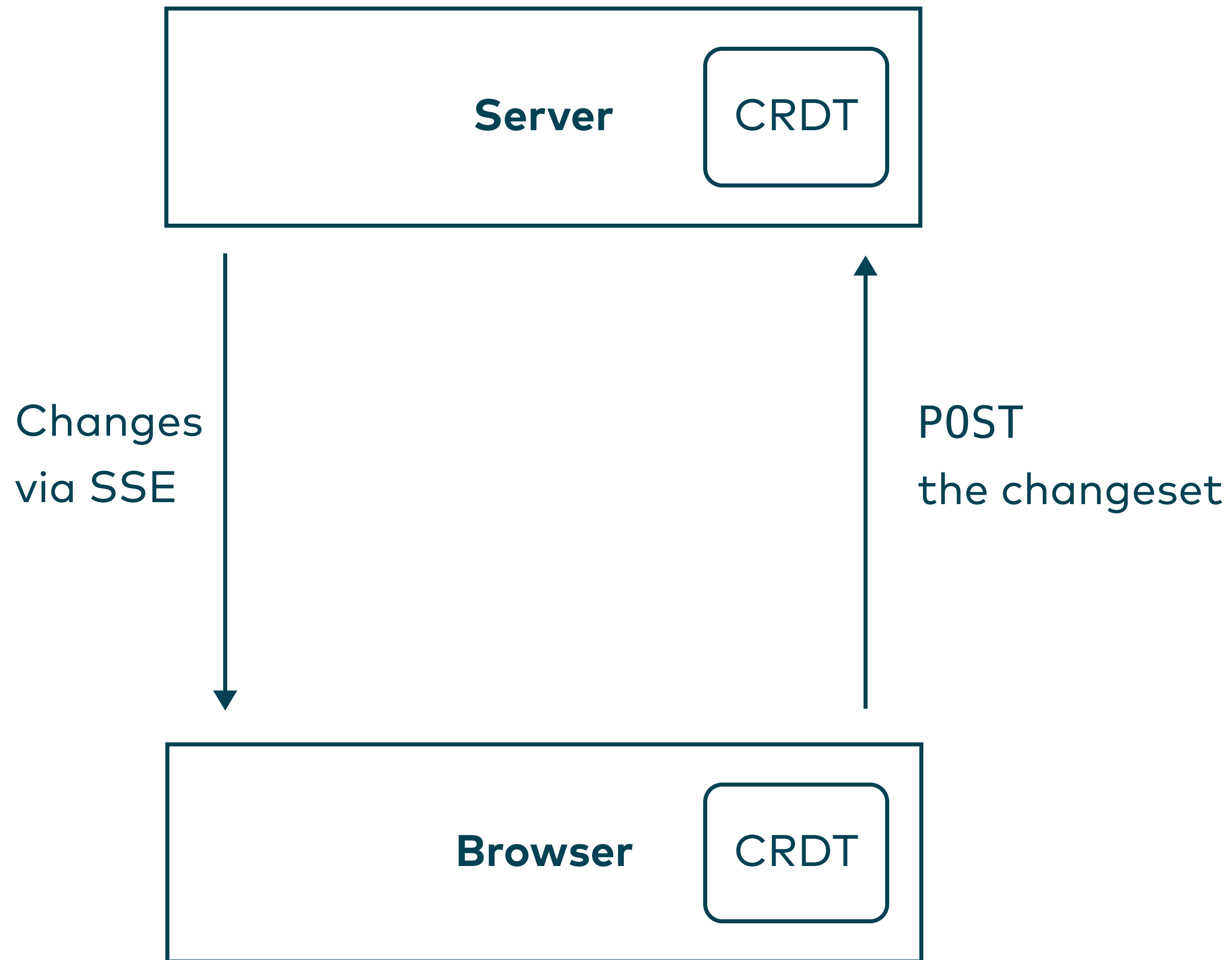

Demo

Shopping List



<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);  
  }  
}
```

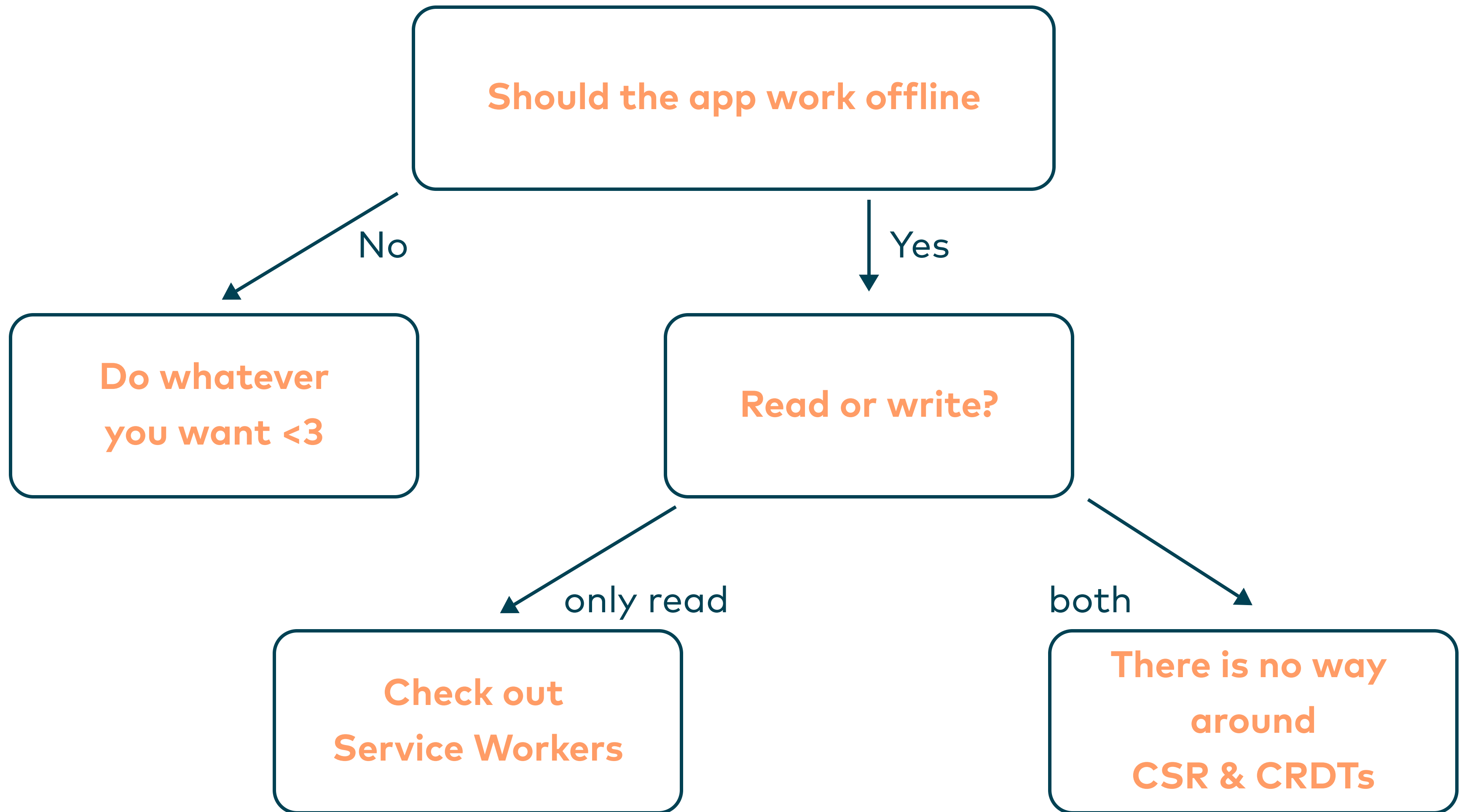
Push Changes

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


More Features

- other datatypes
 - Counter
 - Text (collaborative editing)
 - Peritext
 - table
- P2P with 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