

# Von Miro bis Minecraft Herausforderung Real-Time Collaboration

# Über mich

Ronald Brill  
Principal



**G E B I T**  
Solutions



<https://github.com/rbri>

<HtmlUnit/>

- [@HtmlUnit](#)
- <https://github.com/HtmlUnit>
- <https://github.com/SeleniumHQ/htmlunit-driver>
- [Mozilla Rhino, ....](#)

## BingoBongoTesting

- [www.bingobongotesting.org/](http://www.bingobongotesting.org/)



- [www.wetator.org/](http://www.wetator.org/)

# **Von Miro bis Minecraft**

## **Herausforderung Real-Time Collaboration**

# Wie alles begann....



<https://www.sueddeutsche.de/leben/weihnachten-wunschzettel-kinder-1.4262820>

Neue Version einer bestehenden erfolgreichen Software zur Datenanalyse (BI)

## Anforderungen

- ...Cloud... Browser... (Micro)Services.... universelle Schnittstellen...
- Unterstützung für **kollaboratives Arbeiten!**

# *Gleichzeitig gemeinsam arbeiten...*

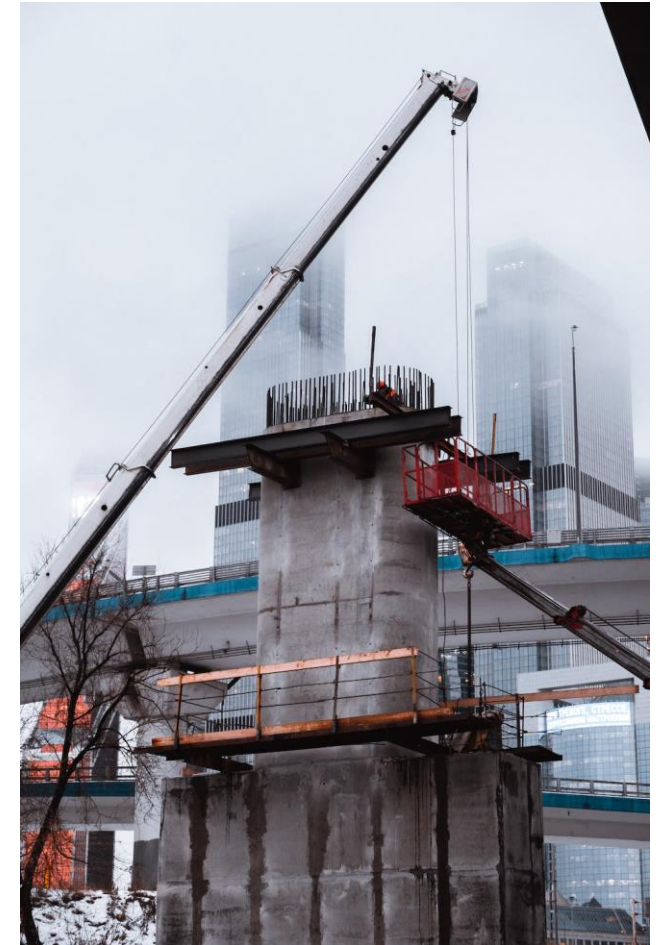
## Was bedeutet **gleichzeitiges kollaboratives Arbeiten**?

Es muss etwas entstehen.... **Artefakt**

- Dokument, Präsentation, Kalkulation (Office 365)
- eine Analyse (Datei)
- oder das was eure Applikation so macht

Out of scope

- Bildschirm freigeben - Steuerung übergeben
- Branch und Mergen



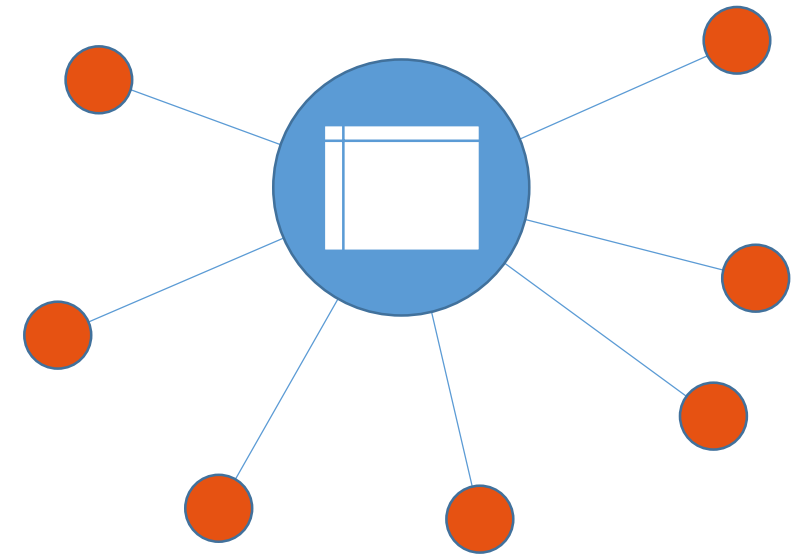
# Was muss Software für kollaboratives Arbeiten können?



# Was muss Software für kollaboratives Arbeiten können?

## Gemeinsamer Zugriff auf das Artefakt

- führt zur Aufhebung der **örtlichen** Trennung
  - keine physikalische Präsenz
- führt zur Problemen bei der **zeitlichen** Konsistenz
  - Reihenfolge wird von den Knoten unterschiedlich wahrgenommen
  - Latenz

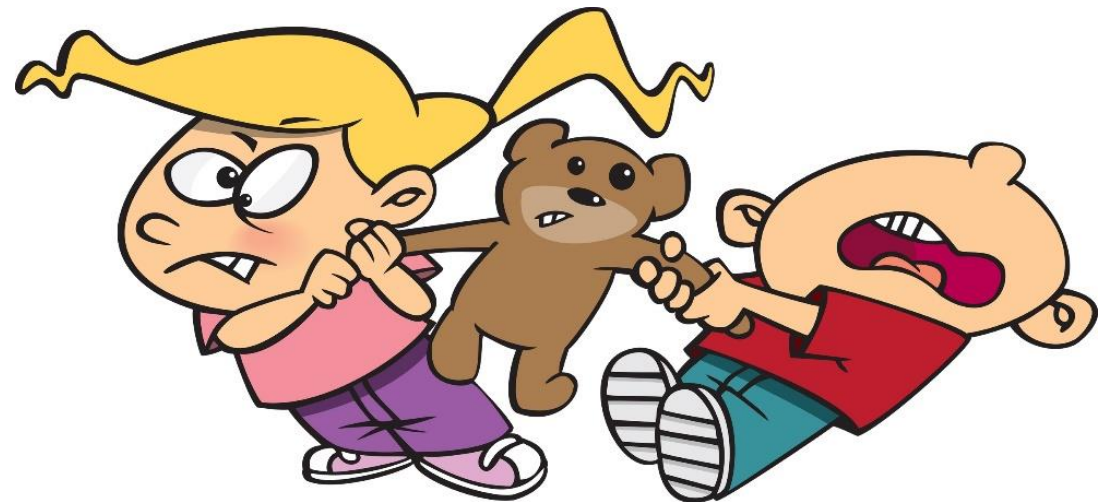


# Was muss Software für kollaboratives Arbeiten können?

## Gemeinsamer Zugriff auf das Artefakt

- führt zur Aufhebung der **örtlichen** Trennung
- führt zur Problemen bei der **zeitlichen** Konsistenz

Notwendig ist die **Vermeidung** von **Konflikten**  
(oder deren automatische Auflösung)





# *Was muss Software für kollaboratives Arbeiten können?*

## **Vermeidung von Konflikten**

- Wir brauchen eine Strategie

Bei welchen Lösungen können wir ‚abkupfern‘?

# Was muss Software für kollaboratives Arbeiten können?

## Vermeidung von Konflikten



Datenbanken hatten / lösen das gleiche Problem

**Be consistent... exclusive lock,  
serialize changes, last wins**

- Transaktion
- Isolation
- Session

# *Was muss Software für kollaboratives Arbeiten können?*

## Vermeidung von Konflikten



Online Games hatten / lösen das gleiche Problem

**Es passiert genau JETZT / one World**

- alle Spieler ‚leben‘ zum gleichen Zeitpunkt
- und (simulierte) Physik

# *Was muss Software für kollaboratives Arbeiten können?*

## Vermeidung von Konflikten

*Keiner hat die Absicht ... den Anderen zu ärgern  
gemeinsam*

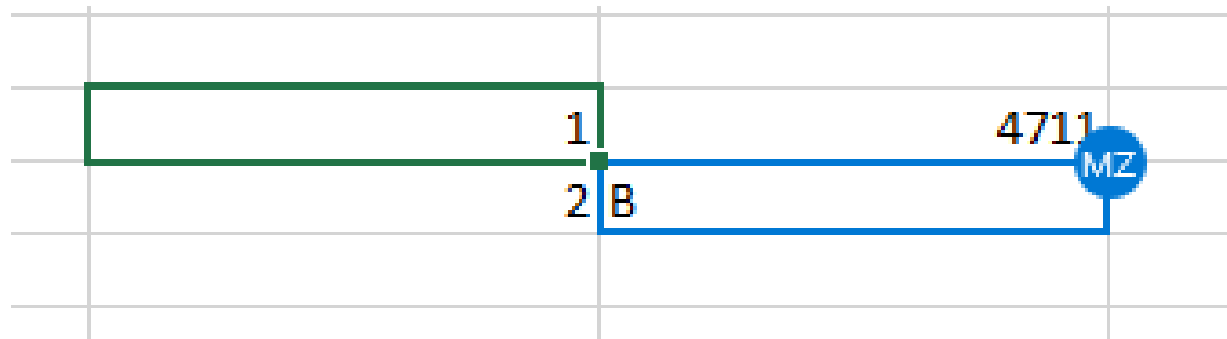
***Konstruktiv***



# Was muss Software für kollaboratives Arbeiten können?

## Vermeidung von Konflikten

- bewusste örtliche Trennung

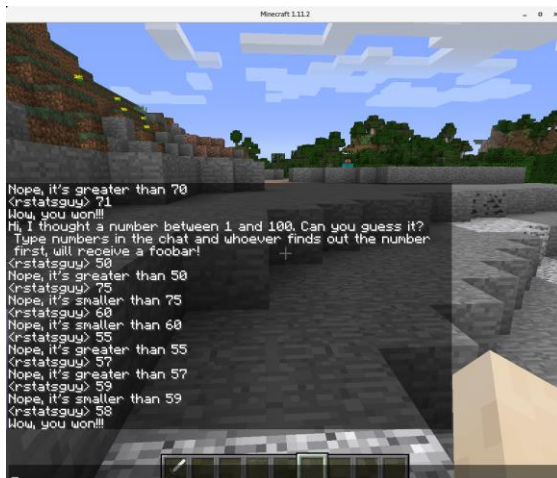


**Konstruktiv**

# Was muss Software für kollaboratives Arbeiten können?

## Vermeidung von Konflikten

- Bewusste örtliche Trennung
- Kommunikation / Abstimmung
- Serialisierung / Aufteilung



teamspeak

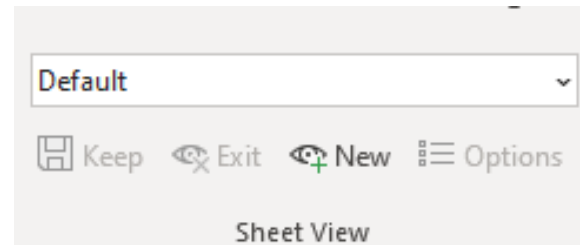


**Konstruktiv**

# Was muss Software für kollaboratives Arbeiten können?

## Vermeidung von Konflikten

- Bewusste örtliche Trennung
- Kommunikation / Abstimmung
- Erfahrung
  - generell
  - mit dem Tool



**Konstruktiv**

# *Was muss Software für kollaboratives Arbeiten können?*

## **Vermeidung von Konflikten**

- Bewusste örtliche Trennung
- Kommunikation / Abstimmung
- Erfahrung
  - generell
  - mit dem Tool
- Vertrauen

***Konstruktiv***



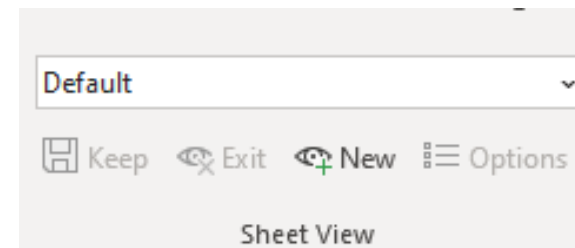
# Was muss Software für kollaboratives Arbeiten können?

## Vermeidung von Konflikten

- Bewusste örtliche Trennung



- übergreifende Funktionen
  - alles Markieren....
  - Filtern...
  - ....



# Was muss Software für kollaboratives Arbeiten können?



***Konstruktiv***

**Vermeidung von Konflikten durch**

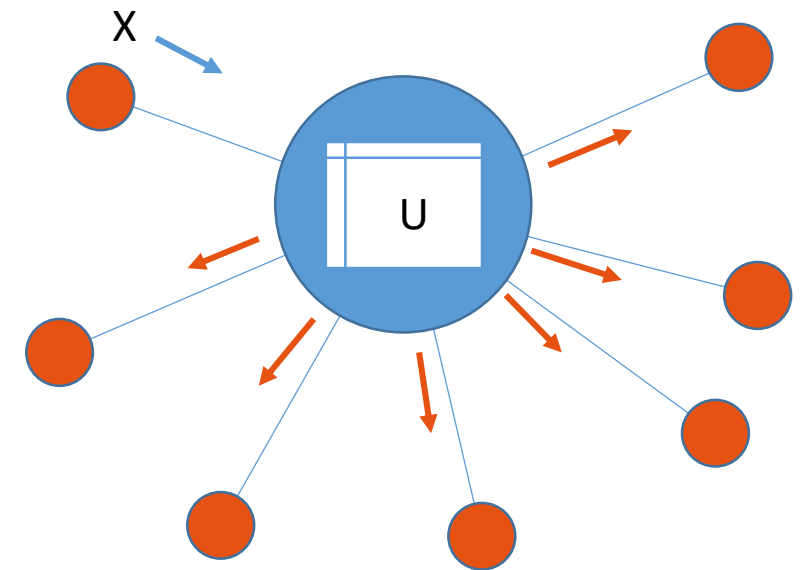
- Bewusste örtliche Trennung

# Was muss Software für kollaboratives Arbeiten können?

## Vermeidung von Konflikten

- zeitlich konsistenter Ablauf

**Latenz**, adj. **latent** (von lateinisch *latere* ‚verborgen sein‘) steht für:  
die *Latenzzeit* oder *Verzögerungszeit* – also den Zeitraum zwischen einem verborgenen Ereignis und dem Eintreten einer sichtbaren Reaktion darauf



# Was muss Software für kollaboratives Arbeiten können?

## Vermeidung von Konflikten

- zeitlich konsistenter Ablauf
- kein Save!
- real time sync
- alle sehen den gleichen (aktuellen) Zustand

es passiert genau JETZT



# Was muss Software für kollaboratives Arbeiten können?

## Vermeidung von Konflikten

- zeitlich konsistenter Ablauf
- kein Save!
- real time sync
- alle sehen den gleichen (aktuellen) Zustand



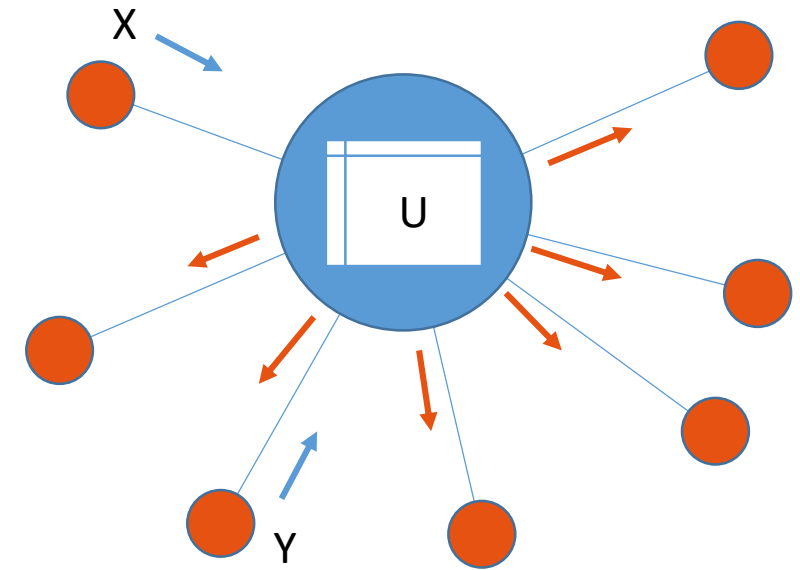
es passiert genau JETZT



# Was muss Software für kollaboratives Arbeiten können?

## Vermeidung von Konflikten

- Beim Schreiben auf die gleichen Daten (keine örtliche Trennung) kann die Latenz zur Konflikten/Inkonsistenzen führen



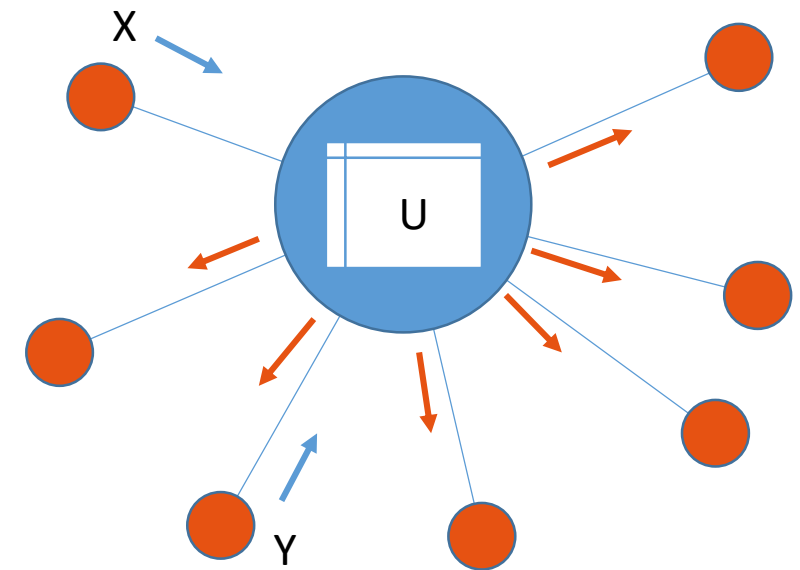
# Was muss Software für kollaboratives Arbeiten können?

## Vermeidung von Konflikten

- Beim Schreiben auf die gleichen Daten (keine örtliche Trennung) kann die Latenz zur Konflikten/Inkonsistenzen führen



***Konstruktiv***



# Was muss Software für kollaboratives Arbeiten können?



abcd

xyz

4711





# Was muss Software für kollaboratives Arbeiten können?



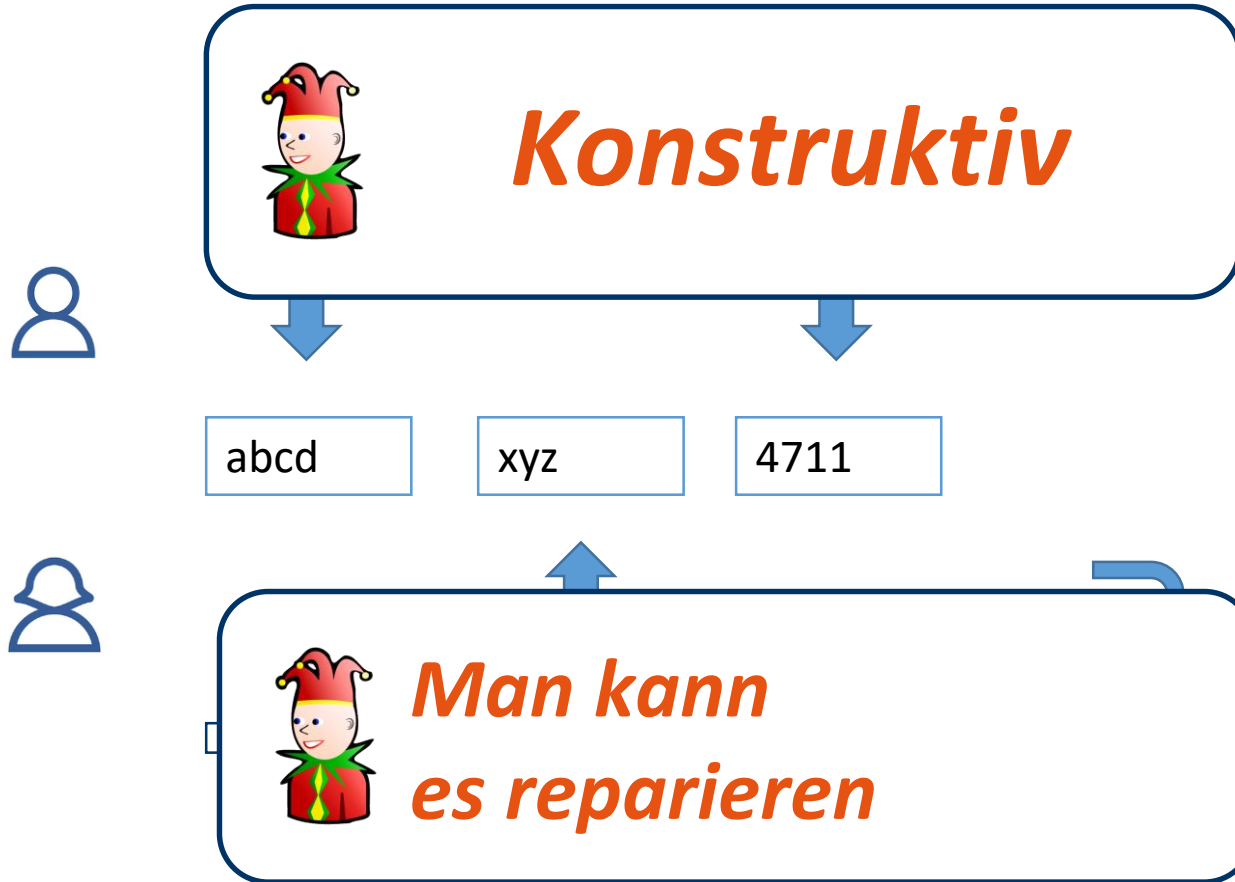
abcd

xyz

4711



# Was muss Software für kollaboratives Arbeiten können?



# Was muss Software für kollaboratives Arbeiten können?



***Konstruktiv***

**Vermeidung von Konflikten durch**

- bewusste örtliche Trennung
- zeitlich konsistenter Ablauf

# *Was muss Software für kollaboratives Arbeiten können?*

## Technische Ansätze

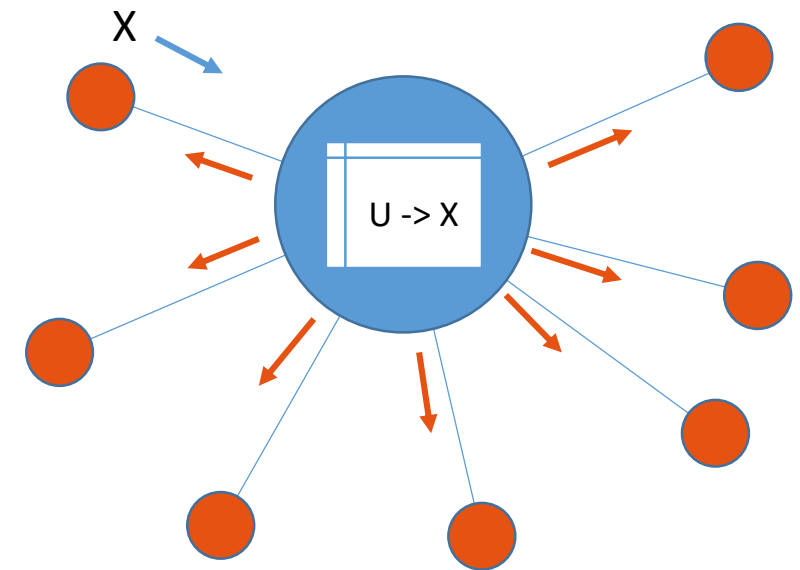
- Terminal
- Operational Transformation
- Conflict free replicated data types



# Terminals

## Terminals – kein lokaler State

- Änderung wird auf dem Server verarbeitet
- Ergebnis wird verteilt
- Alle Anzeigen werden aktualisiert
- Server-seitige Serialisierung
- Latenz bleibt als Problem bestehen



# *Operational Transformation (OT)*

## Wikipedia

**Operational transformation (OT)** is a technology for supporting a range of collaboration functionalities in advanced collaborative software systems. OT was originally invented for **consistency maintenance** and **concurrency control** in **collaborative editing of plain text documents**. Its capabilities have been extended and its applications expanded to include group undo, locking, conflict resolution, operation notification and compression, group-awareness, HTML/XML and tree-structured document editing, collaborative office productivity tools, application-sharing, and collaborative computer-aided media design tools.

In 2009 OT was adopted as a core technique behind the collaboration features in Apache Wave and Google Docs.

# Operational Transformation (OT)

## OT

- Jeder Client hat eine eigene Kopie
  - Lokale Änderungen sind stets erlaubt – kein Locking
  - Änderungen werden dann publiziert
  - Jeder Empfänger von Änderungen muss sie vor der ‚Anwendung‘ noch anpassen (mit seinen Änderungen ‚Mergen‘)
  - Sehr komplexe Algorithmen selbst wenn nur wenige Operationen unterstützt werden
- Group Undo

# *Operational Transformation (OT)*

[https://en.wikipedia.org/wiki/Operational\\_transformation#Critique\\_of\\_OT](https://en.wikipedia.org/wiki/Operational_transformation#Critique_of_OT)

...Joseph Gentle who is a former Google Wave engineer .... wrote,

"Unfortunately, implementing OT sucks. There's a **million algorithms** with **different tradeoffs**, mostly trapped in academic papers. [...] Wave took 2 years to write and if we rewrote it today, it would take almost as long to write a second time."

But later he amends his comment with "I no longer believe that wave would take 2 years to implement now - mostly because of advances in web frameworks and web browsers."



# *Conflict free replicated data types (CRDT)*

## Wikipedia

In distributed computing, a **conflict-free replicated data type (CRDT)** is a data structure that is replicated across multiple computers in a network, with the following features:

1. The application can update any replica independently, concurrently and without coordinating with other replicas.
2. An algorithm (itself part of the data type) automatically resolves any inconsistencies that might occur.
3. Although replicas may have different state at any **particular point in time**, they are guaranteed to **eventually** converge.

The CRDT concept was formally defined in 2011 by Marc Shapiro, Nuno Preguiça, Carlos Baquero and Marek Zawirski. Development was initially motivated by collaborative text editing...

# *Conflict free replicated data types (CRDT)*

## CRDT

- lösen das Problem das durch Latenz entsteht
- auch längere Offline Zeiten werden unterstützt
- brauchen keinen zentralen Moderator / Server
- brauchen keinen ‚Takt‘;  
Reihenfolge der Änderungen (Merges) ist egal

# *Conflict free replicated data types (CRDT)*

## CRDT

- 2 Ansätze
  - Operation based
    - Übertragung von Änderungen durch Operation (die dann beim Empfänger ausgeführt werden)
    - brauchen Garantie für die Zustellung jeder Nachricht
  - State based
    - es wird der komplette Status übertragen und beim Empfänger gemerged

# Conflict free replicated data types (CRDT)

## CRDT

- diverse Typen
  - Grow-only Counter
  - Grow-only Set
  - Last-Write-Wins-Element-Set
  - Observed-Remove Set
  - ...



... Vector Clocks ...

# *Conflict free replicated data types (CRDT)*

## Wikipedia

- Riak is a distributed NoSQL key-value data store based on CRDTs. **League of Legends** uses the Riak CRDT implementation for its in-game chat system, which handles 7.5 million concurrent users and 11,000 messages per second. Bet365, stores hundreds of megabytes of data in the Riak implementation of OR-Set.
- **Fluid Framework** is an open-source collaborative platform built by Microsoft that provides both server reference implementations and client-side SDKs for creating modern realtime web applications using CRDTs.

# *Conflict free replicated data types (CRDT)*

<https://github.com/xi-editor/xi-editor/issues/1187#issuecomment-491473599>

I took on investigation of OT/CRDT with the hope that there was a "right answer" to collaborative editing, only needing to be discovered....

In hindsight, I believe these were reasonable assumptions, but I believe turned out to be wrong.

Indeed, the literature of CRDT does specify a *mathematically correct* answer...  
But this does **not always line up with what humans would find the most faithful...**

# Conflict free replicated data types (CRDT)

 **Automerger** <https://automerger.org/>

Automerger is a library which provides fast implementations of several different CRDTs, a compact compression format for these CRDTs, and a sync protocol for efficiently transmitting those changes over the network.

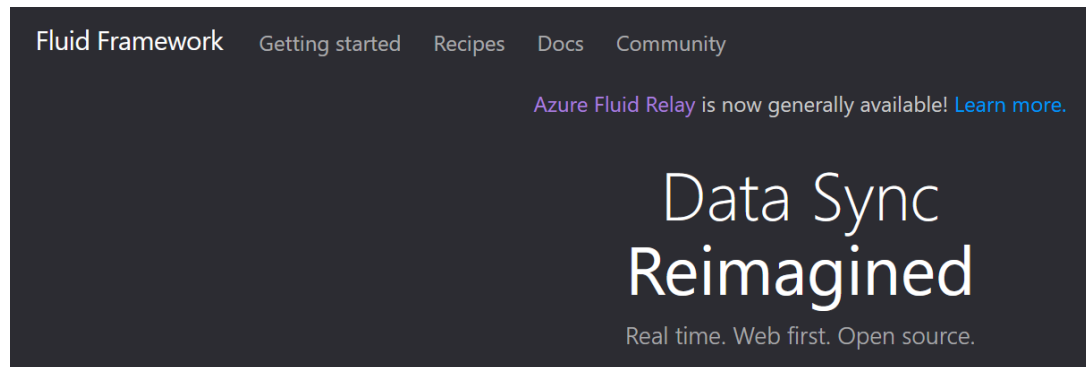
A JSON-like data structure (a CRDT) that can be modified concurrently by different users, and merged again automatically.

 <https://github.com/yjs/yjs>

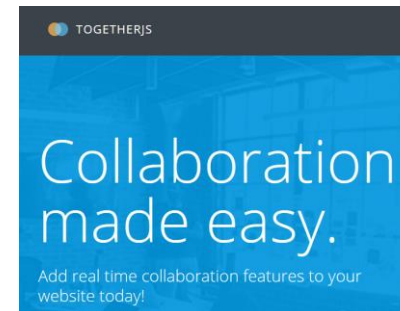
Yjs is a CRDT implementation that exposes its internal data structure as shared types. Shared types are common data types like Map or Array with superpowers: changes are automatically distributed to other peers and merged without merge conflicts.

# Gleichzeitig gemeinsam arbeiten...

## Frameworks



<https://fluidframework.com/>



<https://togetherjs.com/>



<https://github.com/yjs/yjs>



# *Gleichzeitig gemeinsam arbeiten...*

## **... ist schwierig!**

- technische Lösungen stehen ~~bereit~~ in den Startlöchern
- wir haben es (noch) nicht geschafft, gemeinsam mit unseren Anwendern Workflows konkret zu definieren
- nicht alle Nutzer teilen die Auffassung, dass sie im vorgestellten Sinn kollaborativ arbeiten wollen



# *Gleichzeitig gemeinsam arbeiten...*

## **... ist schwierig!**

- wir haben alle unsere Workflows sequentiell gestaltet
  - verschicken von Dokumenten
  - Formulare
  - Laufzettel
  - Scanner & Fax



# *Gleichzeitig gemeinsam arbeiten...*

## **... ist schwierig!**

- Wo anfangen?
- Zentrale Artefakt Ablage
  - sie dann zur Kollaborations-Zentrale ausgebaut werden kann
  - Zugriffsrechte, Artefakte finden, Verschlagwortung
  - Versionierung



# *Gleichzeitig gemeinsam arbeiten...*

## Zukunft – wir stehen am Anfang

- Real-Time Collaboration wird in den Arbeitsalltag einziehen
- unser IT Systeme werden das unterstützen (müssen)
- Umstellung der Use Cases braucht Zeit
- Super spannendes Thema  
Es gibt noch viel zu lernen



# Referenzen

- [https://en.wikipedia.org/wiki/Collaborative\\_real-time\\_editor](https://en.wikipedia.org/wiki/Collaborative_real-time_editor)
- [https://en.wikipedia.org/wiki/Operational\\_transformation](https://en.wikipedia.org/wiki/Operational_transformation)
- [https://en.wikipedia.org/wiki/Conflict-free\\_replicated\\_data\\_type](https://en.wikipedia.org/wiki/Conflict-free_replicated_data_type)
  
- <https://ckeditor.com/blog/Lessons-learned-from-creating-a-rich-text-editor-with-real-time-collaboration/>
- <https://www.tiny.cloud/blog/real-time-collaboration-ot-vs-crdt/>
- <https://github.com/xi-editor/xi-editor>
- <https://technology.riotgames.com/news/determinism-league-legends-implementation>
- <https://technology.riotgames.com/news/determinism-league-legends-unified-clock>
  
- <https://fluidframework.com/>
- <https://togetherjs.com/>
- <https://github.com/yjs/yjs>
- <https://automerge.org/>

# Referenzen

## Bilder und Cliparts

- <https://www.pexels.com/>
- <https://openclipart.org>
- <https://de.freepik.com>