# Building Your Own Collaborative Editing Web Forms For Free

By David Johnson
djohn89@gmail.com
www.djohn89.com

Outline

- How does collaborative editing with Google Docs work?
  - What's the Google Realtime API?
  - Why use Operational Transformations vs. CRDTs?
- Intermission: JK Wrangler and JT Scrambler (new pickup truck)
- How can you build your own collaborative editing system from existing web forms?
 - Examples of my own work with collaborative editing

Collaborative Editing

Client C1	Client C2	Client C3
Lorm ipsum dolor ...	Lorem ipsum dolor ...	Lorem ipsum dolor ...
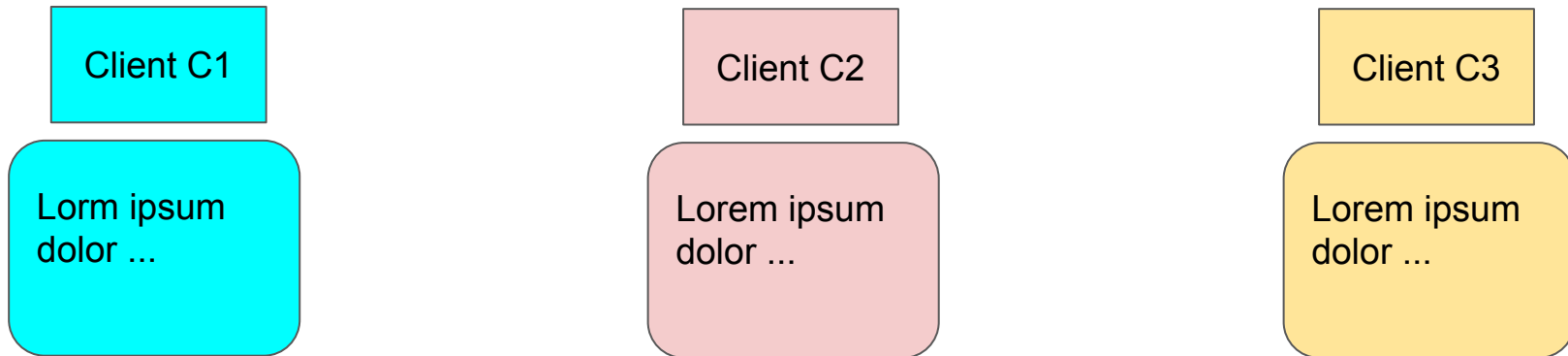
Basic concepts: the same document is edited by multiple users on different devices simultaneously (e.g., Google Docs, Sheets, Slides)
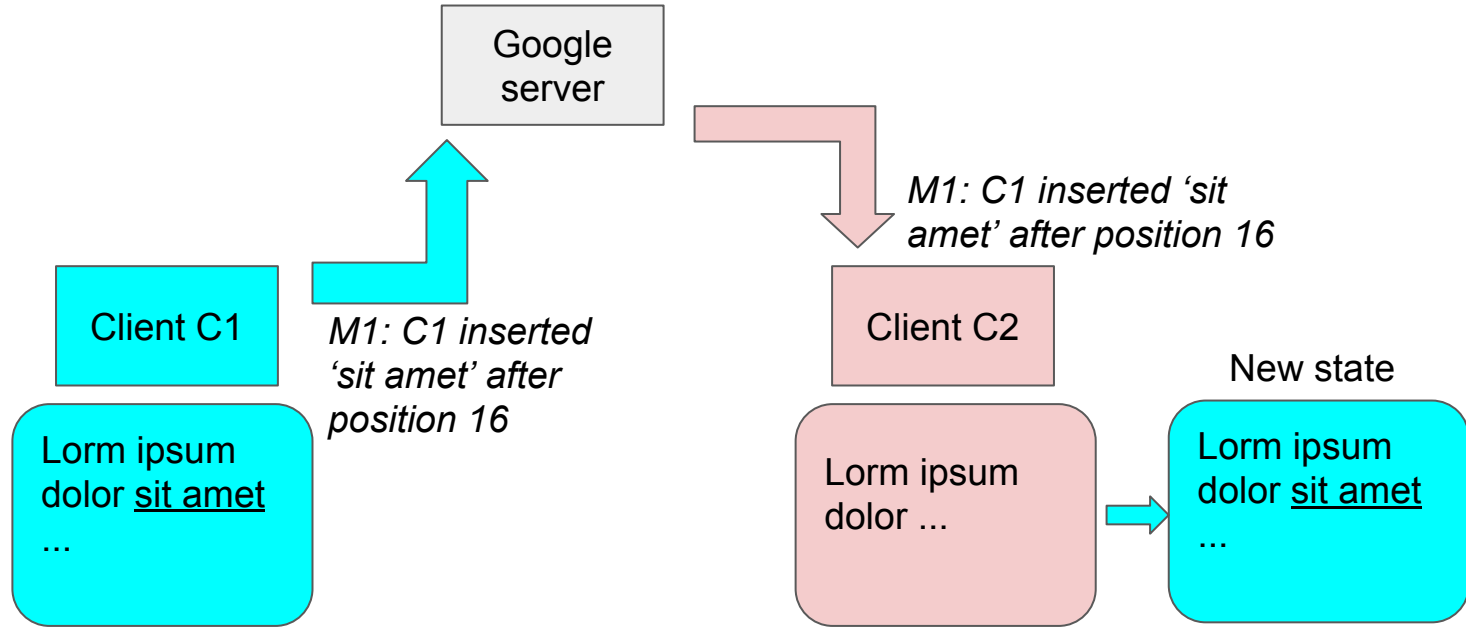
The users' actions include: insertions, deletions, formatting, selection, copy, paste, etc. Unfortunately, these operations don't commute, making it hard to keep the documents the same on every device.

Most operations don't commute

Source
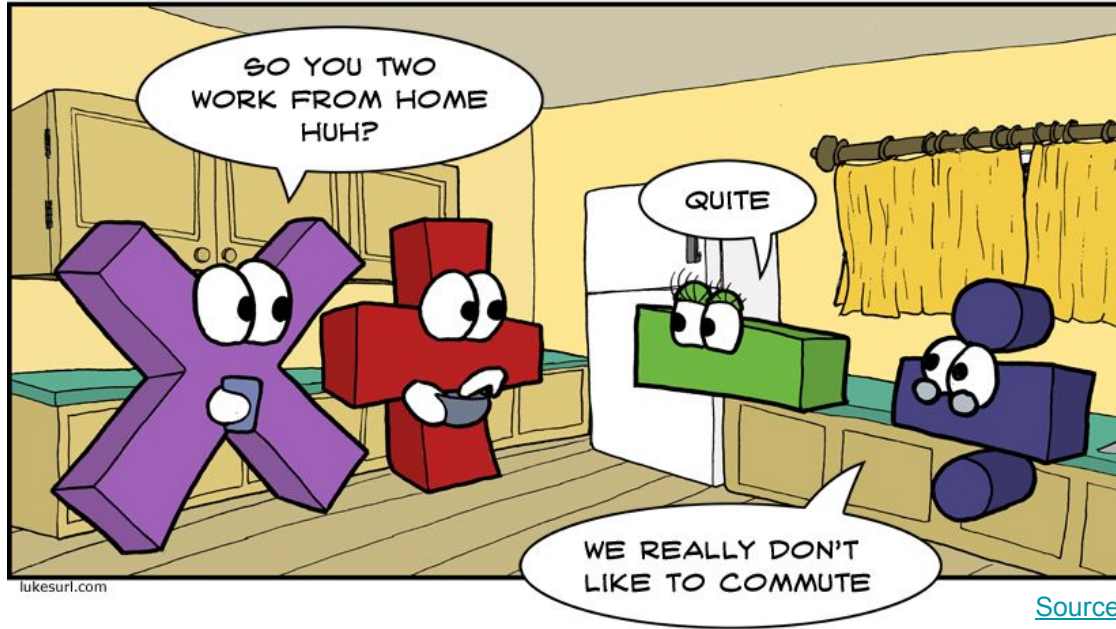
- Commute: A+B = B+A
- But A-B ≠ B-A
- Real life: wash clothes then dry

- Commute: editing different text
- Doesn't commute: editing same text
- Real life: appending text

How does collaborative editing with Google Docs work?

Google server

Client C1

M1: C1 inserted 'sit amet' after position 16

Lorm ipsum dolor sit amet ...

M1: C1 inserted 'sit amet' after position 16

Client C2

New state

Lorm ipsum dolor ...

Lorm ipsum dolor sit amet ...

The shared state (a text document) is maintained by sending messages that contain individual operations (inserting or deleting text). These operations are transformed such that every client will have the same state after all messages are received (eventual consistency).
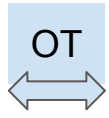
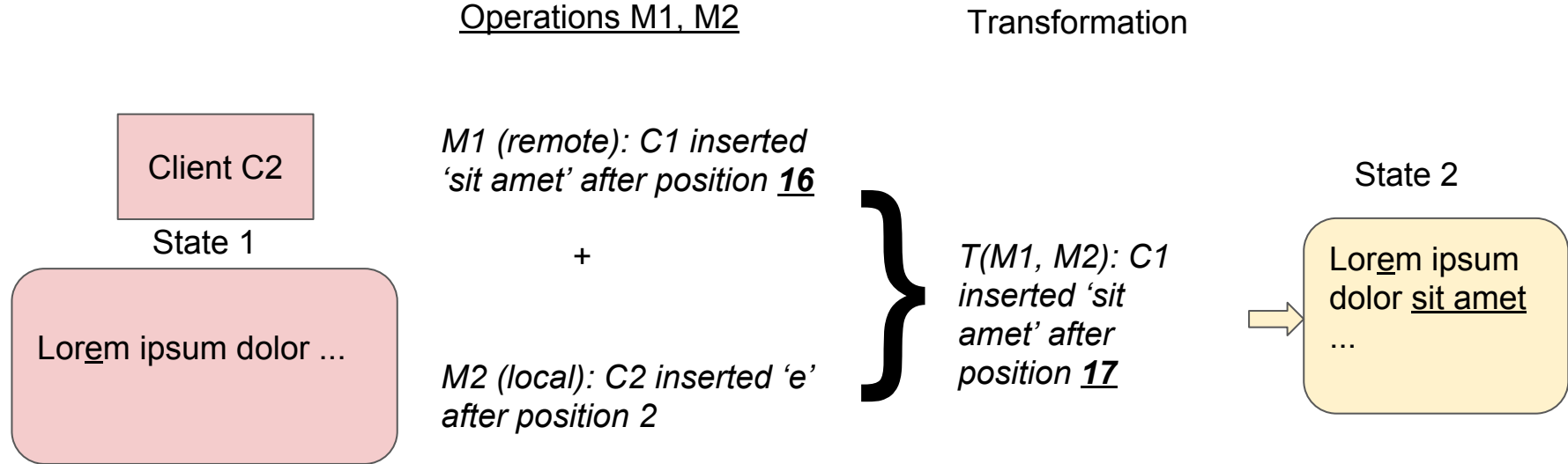# 3-way merge version control (diff and patch)
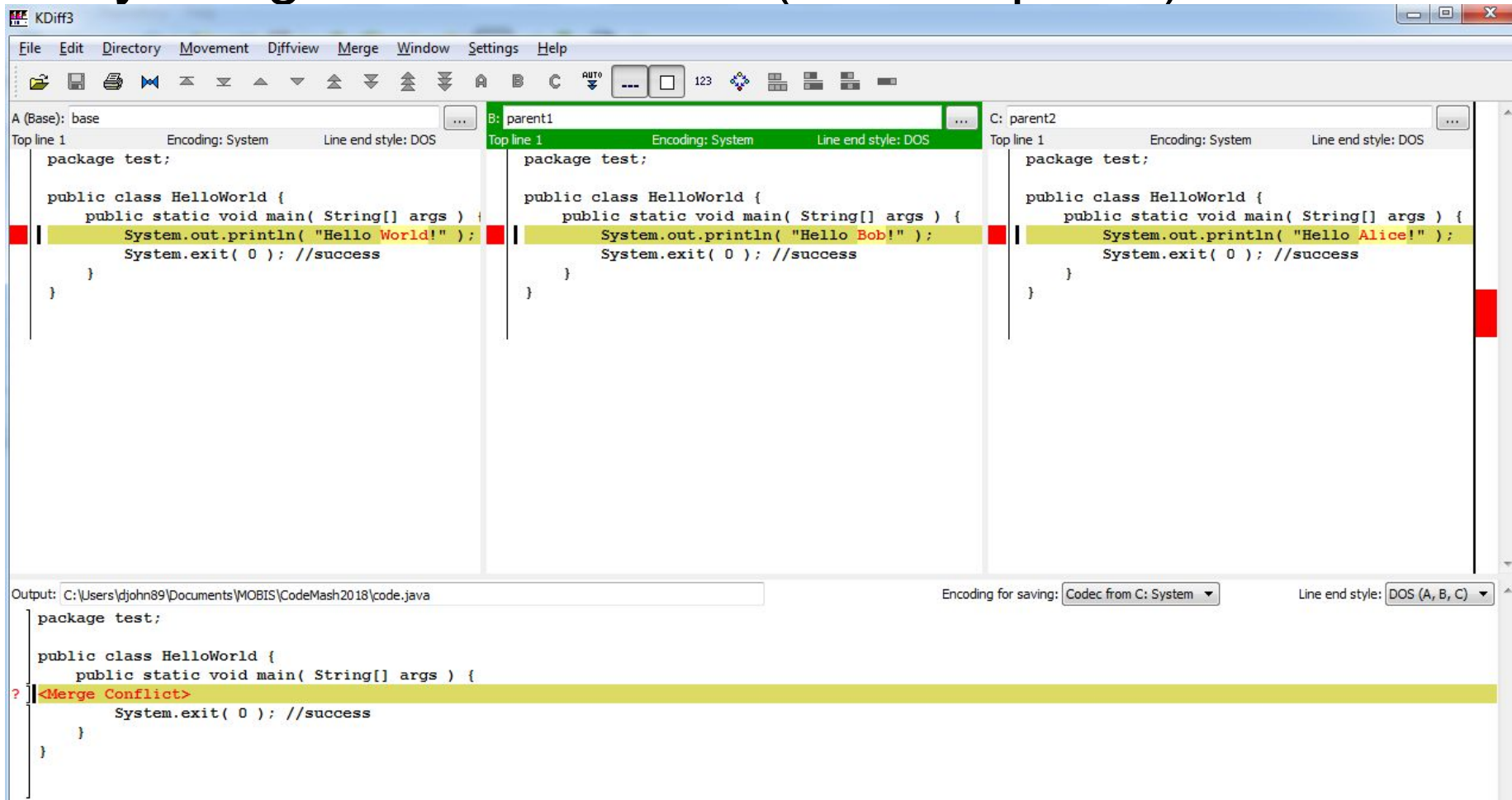
Google Realtime API

- Client side javascript library implementing operational transformations (OT)
- Requires Google-based authentication and storage in Google Drive
- Used by Google Docs, Slides, Sheets, etc.

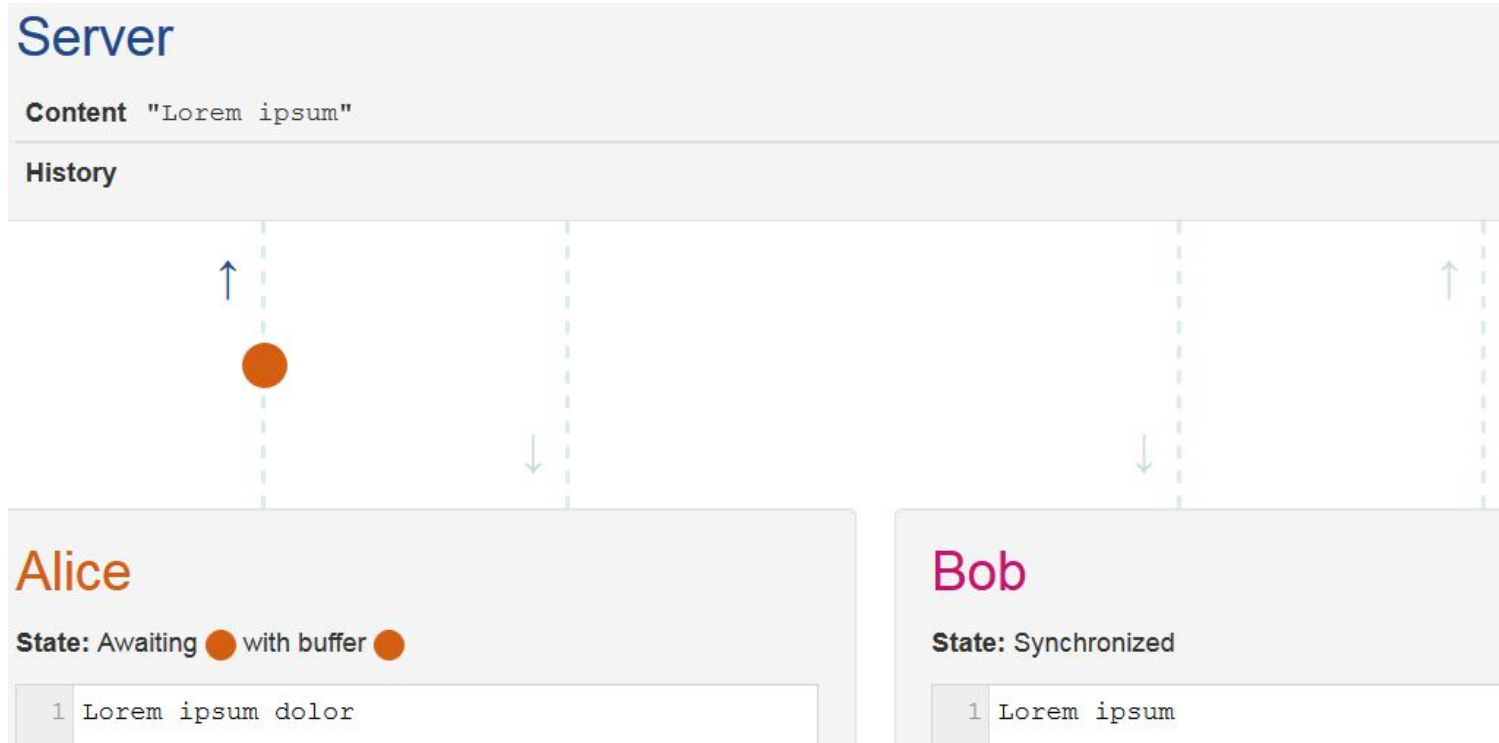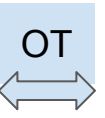https://developers.google.com/google-apps/realtime/overview

Operational Transformation (OT)

Dating back at least to the 1989 Grove System, OT provided the first practical way to maintain a complex shared state with eventual consistency.

Operations M1, M2

Transformation

Client C2

State 1

Lorem ipsum dolor ...

M1 (remote): C1 inserted 'sit amet' after position 16

+

M2 (local): C2 inserted 'e' after position 2

}

T(M1, M2): C1 inserted 'sit amet' after position 17

State 2

Lorem ipsum dolor sit amet ...

OT Visualization Demo

https://operational-transformation.github.io/visualization.html

Google I/O 2013 presentation

"The Secrets of the Drive Realtime API" video

OT Overview

- Operations must be transformed so that they commute
- Message delivery is assumed to be reliable and in-order (*)
- Every client and also the server must implement OT in the same way
- Preservation of user intention is difficult (but solved)

Transformation Functions: T(x, y) must implement every possible pair of operations

|	Op1	Op2	Op3
Op1	T(Op1, Op1)	T(Op1, Op2)	...
Op2	T(Op2, Op1)		
Op3	...		...

OT Challenges
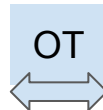
- OT functions mutate the operations (insertion, deletion, selection, copy, paste, etc.) so that all operations will commute and eventually converge
  - Unfortunately, many OT algorithms are flawed or very difficult to implement correctly
  - A single failed transformation will ruin the whole scheme
  - Transaction Property 2 (TP2) is very hard to achieve
- Plain text editing is the easiest to get right, but formatting operations add a lot of business value (and complexity)
- Google Wave was discontinued in 2010 due to lack of user interest
- OT is still used by Google Drive applications (Docs, Sheets, etc.)
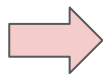
Newer Alternative - CRDTs
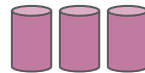
Conflict-free Replicated Data Type (CRDT): a distributed data type that is guaranteed to converge to a shared state. No math! E.g., Riak, Voldemort, or Cassandra

Added set

Position	Character
0	L
1	o
2	r
3	m

"Lorm"

M2 (local): C2 inserted 'e' after position 2
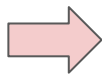
Added set

Position	Character
0	L
1	o
2	r
2.1	**e**
3	m

"Lorem"

CRDTs - tombstone function

CRDT
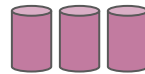
Deletions are handled by creating a second set of deletions ("tombstones"). The final string as rendered by sorting the sets and removing the elements that are in the Remove-only set.

Added set

Position	Character
0	L
1	o
2	r
2.1	**e**
3	m

"Lorem"

M3 (local): C2 deleted 'o' after position 1

➡

Added set

Position	Character
0	L
1	o
2	r
2.1	**e**
3	m

Removed set

Position	Character
1	o

"Lrem"

Tombstones: not for the faint of heart

Personality Type Tombstones
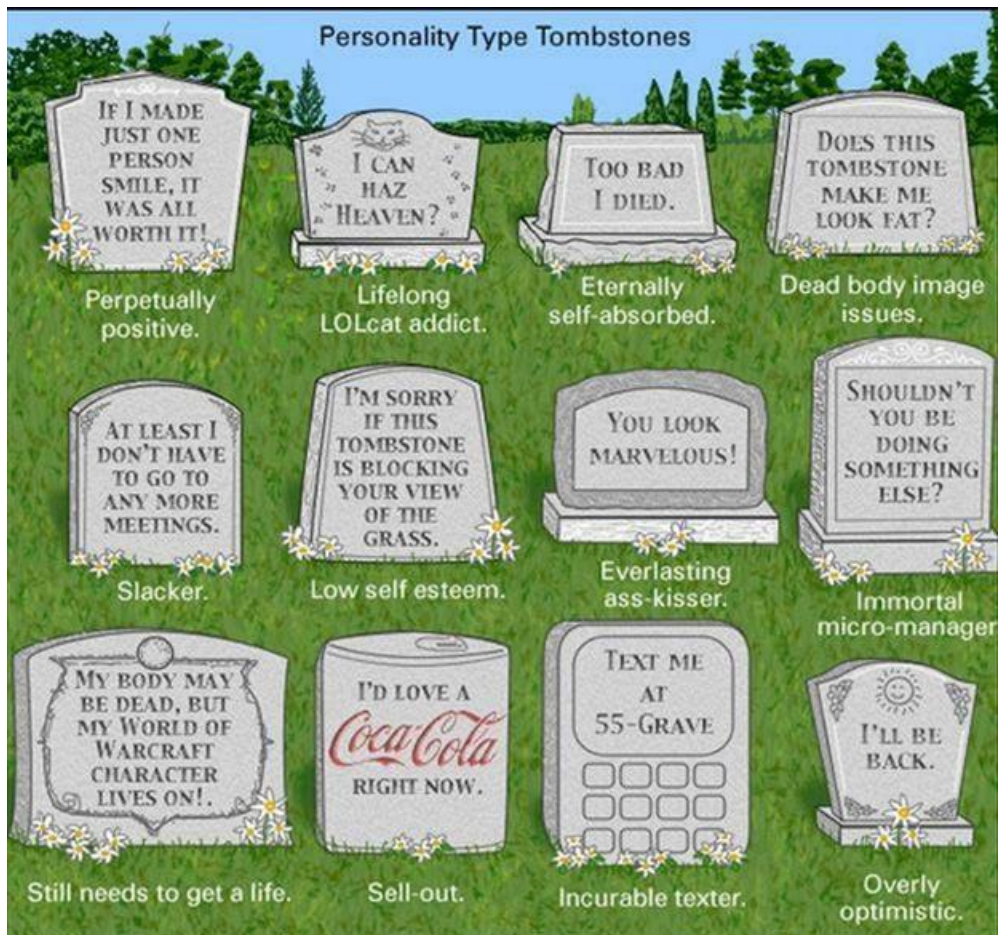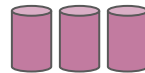
- Append only sets (Added, Removed) provide correctness guarantees
- The Remove-only set is a form of garbage collection (*)
- There are serious performance issues in distributed NoSQL databases with the accumulation of tombstones

Source

CRDT details

- Real life CRDTs (Cassandra): Netflix, Twitter, Wikimedia

- Implementation details: vector clocks - every character is assigned an identifier based on its character position, editor ID, and monotonic local operation counter (strict, total ordering)

- Multiple editors are handled by taking the union of each Added set and the union of each Removed set

- Convergence is guaranteed because no identifier is ever deleted or reused.

- No server is required to limit concurrency or assign an order to the operations

- Downsides: the length of the identifiers is unbounded, sorting can become slow, and the size of the Removed set is unbounded

Summary: OT vs CRDTs
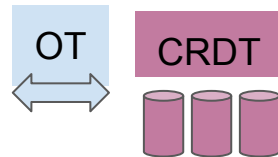
- OT has a long history with some well-known problems and very difficult implementations, but it generally works (*) and uses little memory or computation
- CRDTs are much newer, offer better guarantees of correctness, and can be simpler to implement (*), but they can become slow due to memory usage and computation required
- Some equivalence exists for these two ideas at a theoretical level, but the implementations differ significantly
- Next: intermission for Jeep video, then practical implementations with free software

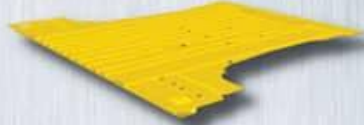# Intermission - JK Wrangler conversion to truck

[Youtube - Jeep Wrangler Unlimited Pickup Conversion Revealed](#)

Enthusiasts have been converting their Wranglers to trucks for years. Conversion kits cost between $6,000 to $21,000+ depending on features.
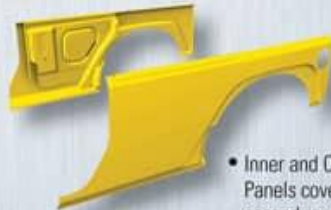
Example conversion kit: Mopar JK-8

- Half Hard Top works with Freedom Top Panels to enclose the new cabin
- Bed Floor extends the truck bed to 50 inches
- Bulkhead provides 44 inches between the wheel wells
- Inner and Outer Body Panels cover the second row entrance

- Released at Moab 2011
- Dealership price: $5,500 parts, $5,000 installation (varies)
- Time: 30 day parts delivery, ~2 weeks installation (35 hours shop time: welding, painting)
- 3 year, 36,000 mile warranty only applies if installed by dealership technicians
- Other kits: AEV Brute DC350 (retains rear seats), JKUte, Crew

Intermission - JT Wrangler Pickup (Scrambler)

1981-1985 CJ-8 Scrambler Wikipedia - fuel economy and public perception doomed it

Youtube: 2019 Jeep Scrambler Pickup (JT)

Most of my time is going towards the new Wrangler Pickup (Scrambler). Possible options: longer frame, new domestic diesel engine, even bigger axles, and better towing capacity.

End of Intermission

- How does collaborative editing with Google Docs work?
  - What's the Google Realtime API?
  - Why use Operational Transformations vs. CRDTs?
- Intermission: JK Wrangler and JT Scrambler (new pickup truck)
- **How can you build your own collaborative editing system from existing web forms?**
  - **Examples of my own work with collaborative editing**

Practical implementations of Collaborative Editing

- All the work is done for you, but it's not free:
  - Google Real-time API (Docs, Forms, etc.), Microsoft Office 365
  - FireBase, Backendless, Parse
  - Google Drive, Dropbox, Microsoft OneDrive
- Free but not real-time:
  - Version control software (e.g., Git)
  - Wikipedia, reddit, digg
- Free but you have to do some work:
  - Editors:
    - CKEditor, CodeMirror, Ace Editor
    - Collabedit, Etherpad
  - Data models:
    - TogetherJS
    - ShareJS / ShareDB

Why not use an existing solution?

- Money: $10 per user per month (Google Suite), $20/user/month (Microsoft E3)
- Off-site dependency: only works if public Internet works
- Legal liability: off-site server could lose data or disclose it inappropriately
- Loss of control: unscheduled downtime, breaking API changes, server maintenance, data format changes, browser incompatibilities
- Obsolescence: Product is discontinued (e.g., Google Wave, Chat, Labs, Trends, etc.) or reduces functionality with no migration path
- No native app access: must use web browser and javascript (Google RT-API)

CKEditor

Heading 2 ▾ Font ▾ Size ▾ **B** *I* U S I_x A▾ A▾

Lorem Ipsum Dolor Sit Amet

Welcome letter
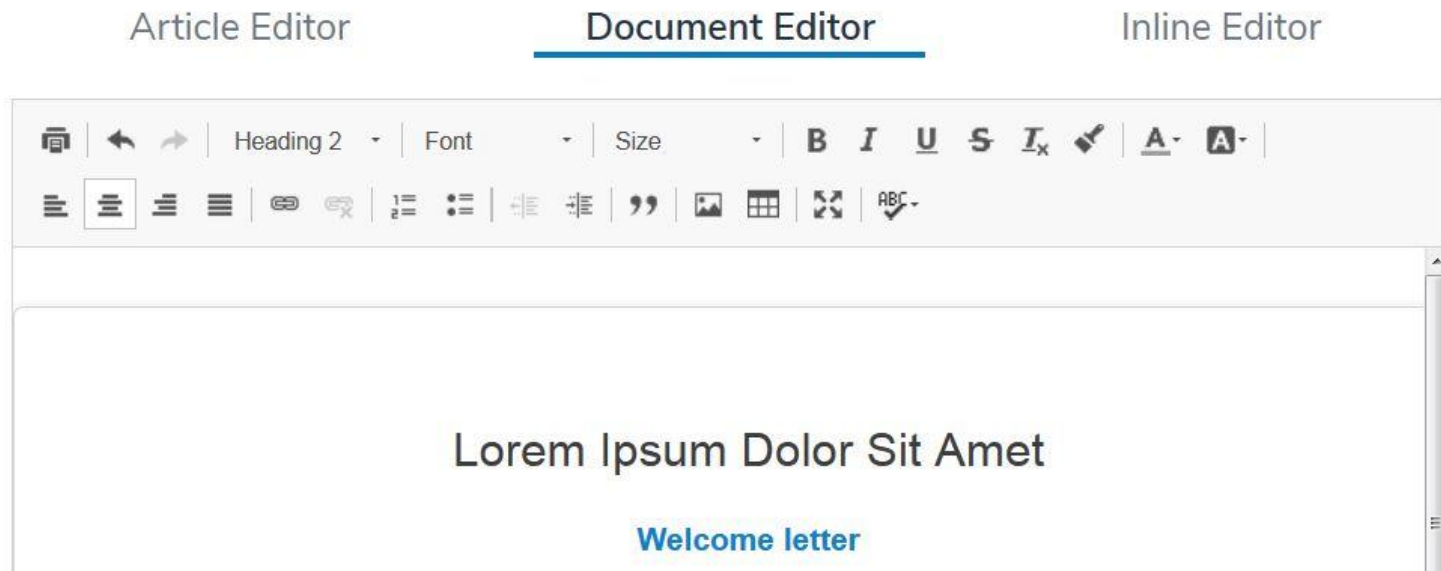
CKEditor 4 is a free, full featured HTML editor. Works great for a single user. Image uploading is not free.

CKEditor 5 is a rewrite with OT and some collaborative features ("Letters") - but it's currently in alpha and the collaboration is not free.

TogetherJs

TogetherJs is a free, open source javascript library made by Mozilla in 2013 that:

- Synchronizes the DOM between browsers
- Performs OT on text fields
- Uses a simple message relay server
  - Locally hosted server is freely available (NodeJs with websockets)
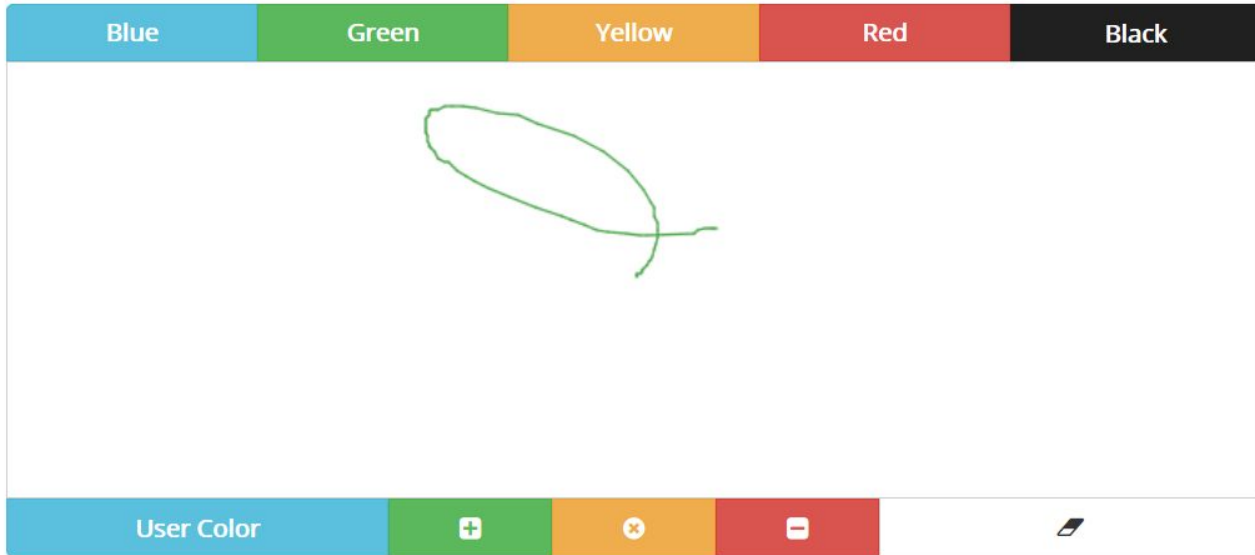- Provides chat functionality and user operation feedback

It does not provide storage or authentication. It works with almost any existing webapps by design (see architect's blog post).

TogetherJS demos

My Examples of Collaborative Editing

Systems I've implemented:

- Time and attendance (Manning)
- Consolidated Shift Notes

These were real webapps that needed to be created or enhanced with collaborative editing functionality

Ex. 1: Manning Reports

Workers for Production team on 01.01.1917 [Shift: 2]

This report: *Modify* "

Add a missing employee: *Register*

Add all missing employees for the Production team. *Register*

Filter by Reason: Any

Total : 143

Delete this report: ✖ Delete

Area	Station	Name	Notes	Reason	Value	Entry	Exit	HR Notes	HR Reason	HR Value	Net Hours Worked
Frame	2 R	Victor Zuppa		-	0.0				-	0.0	
Frame	2 L	Robert Zwak		-	0.0				-	0.0	
Frame	3 L	Albert Brown		-	0.0				-	0.0	
Frame	3 R	Edgar Ellis		-	0.0				-	0.0	
Frame	4 R	Walter Sparkman		-	0.0				-	0.0	

Twice a day, the Production department must record attendance for approximately 200 operators and support staff across various teams.

Human Resources must enforce policies (excused vs. unexcused absences).

When HR and Production supervisors edit the same report, hilarity ensues!

More Manning Reports

Area	Station	Name	Notes	Reason	Value	Entry	Exit	HR Notes	HR Reason	HR Value	Net Hours Worked
Tool Tech ▾	- ▾	Riley White 📅	Unexcused Absent	Absent ▾	1.0			Validated by S.K. 1/3/1917	FMLA-Full ▾	0.0	
Maintenance ▾	- ▾	Theron Jewell 📅	Medical	Medical Leave ▾	0.0			Rejected by S.K. (no note provided)	Absent ▾	1.0	
Maintenance ▾	- ▾	Carl Mcnaughton 📅	Excused per policy	LOA/NSCH ▾	1.0				LOA/NSCH ▾	1.0	
TEAM LEADER ▾	- ▾	William Salyers 📅	VNS (pre-approved)	VNS ▾	1.0			Validated by S.K.	VNS ▾	1.0	
Maintenance ▾	- ▾	Guy Zurbrugg 📅		- ▾	0.0				- ▾	0.0	

Example of attendance review process: employee does not show up for scheduled shift. Supervisor assigns 1 attendance point for Unexcused absence and calls in another employee (involuntarily coverage). Employee files grievance contesting the attendance point and claims FMLA exception (take care of sick family member). HR investigates and requests doctor's note. Employee provides note; HR verifies accuracy and removes point if approved.

Collaborative Editing for Manning Reports

Area	Station	Name	Notes	Reason	Value
Frame ▼	2 R ▼	████████		- ▼	0.0

SELECT boxes: onChange, save

Short text (no OT)

Longer text (needs OT)

Different fields need different event handlers and types of synchronization

Attendance point values cause other reports to change

(This used to be an Excel file on a network drive, which had a long history of causing problems.)

TogetherJS integration for Manning Reports

- Run node.js server on internal production network using hub\server.js:

```
C:\Users\Administrator\togetherjs-develop>\nodejs\node.exe hub\server.js --port 8081 --host 0.0.0.0
```

- Load together.js in client side webpage:

```
<script>
var TogetherJSConfig_hubBase  = "wss://myserver.internal.net:8081/ws";
// more config...
</script>
<script src="js/togetherjs-min.js"></script>
```

- DOM events will automatically propagate changes between browsers
- Last user submits form or automatically saves state via AJAX to server

Ex. 2: Consolidated Shift Notes

- Each department fills out a shift note: a summary of all important issues occurring during that shift
- The other shift reads the shift notes so that they know the situation they are walking into, then writes second shift notes
- At a daily management meeting, both sets of notes are reviewed
- Goals: accountability, problem recurrence prevention, various Key Performance Indicators (KPIs)
- Previously this was an email process. Many emails were generated then lost over the years; the emails were poorly formatted and difficult to search; attachments clogged up inboxes

Production & Quality Shift Notes

MOBIS-NA-OH PRODUCTION SHIFT LETTER

✖ Close

Top of Form

| DATE | | | | SHIFT | | 1st | |

MANPOWER

WHOLE PLANT ABSENTEEISM		Full Time	Supplemental	Temp			Full Time	Supplemental	Temp
	Absent	2.0	0.0	1.0	Suspension		0.0	0.0	0.0
	Vacation	4.2	0.6	0.0	Open Jobs		2.0	0.0	0.0
	Late/Leave	0.3	0.6	0.0	VNS		11.0	2.0	0.0
	Exc Policy	9.0	4.0	0.0	Light Duty		1.0	0.0	0.0
	FMLA-Full	10.0	0.0	0.0	Training		0.0	3.0	5.7
	FMLA-Part	1.1	0.0	0.0	Penalty Heads		1.0	0.0	0.0
	Medical Leave	8.0	1.0	0.0	Union Bus		1.0	0.0	0.0
	LOA/NSCH	6.4	0.0	1.0	Special Project		1.0	0.0	0.0
	TOTAL ABSENT	54.0	8.2	2.0	Total present but not in RTR:		4.0	3.0	5.7

Manpower Utilization – Actual / Required To Run

ACTUAL / RTR	Department	Full Time		Supplemental		Temp		Total		Target	
		Present	RTR	Present	RTR	Present	RTR	Present	RTR	RTR	Delta
	Production	102.8	98.8	22.4	20.4	32.0	26.3	157.2	145.5	145.0	0.5
	Materials	19.0	19.0	9.0	8.0	1.0	1.0	29.0	28.0	27.0	1.0
	Quality	10.2	10.2	2.4	2.4	0.0	0.0	12.6	12.6	11.0	1.6
	Maintenance	7.0	7.0			0.0	0.0	7.0	7.0	10.0	-3.0
							TOTAL	205.8	193.1	193.0	0.1

SAFETY ISSUES	WHAT/WHERE/WHO/WHEN/WHY
	Reportable injury (3+ days off)

PRODUCTIVITY

1st Hour JPH	39	Inspection Issues	41.0
Trestle Count @ End of Shift	46	FTC Percentage	96.8

✉ Message ✳ 20161220_073636.jpeg (894 KB)

1st Shift Quality Supervisor Notes:

Safety:
- No issues to report

Significant Issues At Chrysler:

- Nothing to report

Significant Issues At Mobis:
Last APAC picture taken: **3740**

- Seq. 8590 no yellow balance marks on the transfer case to the rear drive shaft.
- Seq. 8670 bent motor mount bracket making decking difficult.(see picture of repair)
-

Sorts At Mobis:
PIC\Dana – axle sort for leaking fluid
PIC\Trenton – engine sort for damaged vent tube on the plenum
PIC – clickering upper control arm, front to axle and frame

Tools in Branch:

- Supervisors kept forgetting to send the emails
- Too many emails and too many attachments
- Other emailed notes: materials, maintenance, tooling, IT, safety

New Collaborative Editing Website for Shift Notes

Information Technology (IT)

Environment, Health, and Safety (EHS)

More Shift Notes

Quality Notes

Create New Note

Search Options

Inquiry

Search:

Modify	Date	Shift	Created	Modified	Email Sent At	Delete
Edit	20171019	2	2017-10-20 02:49:06 by Mark Epley	2017-10-20 03:35:24 by Mark Epley	2017-10-20 03:55:00	Delete
Edit	20171019	1	2017-10-19 07:42:38 by Adam Szymanski	2017-10-19 15:00:44 by Russell Baskett	2017-10-19 16:30:01	Delete

Maintenance Notes

Create New Note

Search Options

Inquiry

Modify	Date	Shift	Created	Modified	Email Sent At	Delete
Edit	20171020	1	2017-10-20 09:45:04 by Paul Worden	2017-10-20 09:46:01 by Paul Worden		Delete
Edit	20171019	1	2017-10-19 17:13:55 by Randy Howell	2017-10-20 02:17:01 by Tom Green		Delete

Tooling Notes

Create New Note

Search Options

Inquiry

Search:

Modify	Date	Shift	Created	Modified	Email Sent At	Delete
Edit	20171020	1	2017-10-20 05:41:40 by Dylan Boudrie			Delete
Edit	20171019	2	2017-10-19 17:50:23 by Dave Semenovich	2017-10-19 23:29:06 by Dave Semenovich	2017-10-20 03:55:00	Delete

Materials Notes

Create New Note

Search Options

Inquiry

Search:

Modify	Date	Shift	Created	Modified	Email Sent At	Delete
Edit	20171019	2	2017-10-20 03:03:33 by David James		2017-10-20 03:55:00	Delete
Edit	20171018	2	2017-10-19 02:49:57 by David James		2017-10-19 03:55:00	Delete

Example of Editing - Quality Shift Note

Supervisor types note into CKEditor (HTML editor in <textarea>)

Example of Photo Upload - Quality Shift Note

- Basic jQuery Mobile site with custom photo uploading servlets
- Photos are automatically resized and attached to emails
- Responsive layout for mobile and desktop browsers

Collaborative Editing for Shift Notes

- TogetherJS doesn't work with HTML editors (CKEditor) and CKEditor 5.0 is not ready yet
- My solution for now: use [ShareJS](#)

To make changes to a document, you can call one of these three methods:

- doc.create(type, [data], [context], [callback]): Create the document on the server with the given type and initial data.
- doc.submitOp(op, [context], [callback]): Submit an operation to the document. The operation must be valid for the given OT type of the document.
- doc.del([context], [callback]): Delete the document on the server.

Warnings to others: don't try to use diff/patch on HTML documents! Google tried this with [Differential Synchronization](#), and there were endless bugs [due to DOM parsing errors and browser incompatibilities](#).

Discussion and Conclusion

- Operational Transformation and CRDTs are the two main methods of synchronizing data in between web browsers
- Google Docs uses OT, but you have to use Google servers
- With an editor and a data model, you can build your own solution for free!

End

The strongest drive

is not love or hate.

It is one person's need
~~modify, revise alter~~

to ~~change~~ another's copy.
~~rewrite~~ amend ~~enoptopieces~~
change

Source: Arthur Plotnik's *The Elements of Editing: A Modern Guide for Editors and Journalists* (New York: Macmillan, 1982).

Where do irreconcilable conflicts come from?

- Multiple users
- Same data
- Spread out over time
- Loss of order of operations
- Result: Large, irreconcilable diffs