



DrupalCon
GLOBAL 2020
JULY 14-17

Real-Time Collaboration and the Future of Web Applications

Peta Hoyes
Tag1 Consulting
Chief Operating Officer
Peta@Tag1.com
drupal.org/u/phoyes

Fabian Franz
Tag1 Consulting
VP Software Engineering
Fabian@Tag1.com
drupal.org/u/fabianx



Tag1 Consulting

The Team Global Leaders Trust to Ensure Their Success with Technology



Peta Hoyes

Partner/COO

- Leads Delivery, Operations, Finance
- Mechanical Engineer with a passion for great product design and art
- Created one of the first hyper-local communities, New York Online in 1993

Fabian Franz

VP, Software Engineering

- 1 of 3 Drupal (D7) Framework Managers
- Performance & Scalability Expert
- Architected code used by 2% of the internet & 10% of top 100k sites



What We'll Cover

- The Collaboration Spectrum
- Core Features Required to Enable Collaboration in Applications
- How This Applies to Content Management Systems
- How we Integrated Collaborative Editing in Drupal and WP
- Challenges Going From Single to Multi User Collaborative Apps
- Collaboration and The Future of Web Applications
- Your Questions

We'll Demo Working Collaborative Applications Along the Way
Including Shared Editing in Both Drupal and WordPress



Collaborative Applications





Collaboration is Fundamental

Working Together is How People & Orgs Get Most Things Done

- Collaborative Applications Revolutionize The Way We Work
- Technology Reduces Costs & Friction While Increasing Productivity
- Critical to Traditional Workforce, Essential to Modern Workers

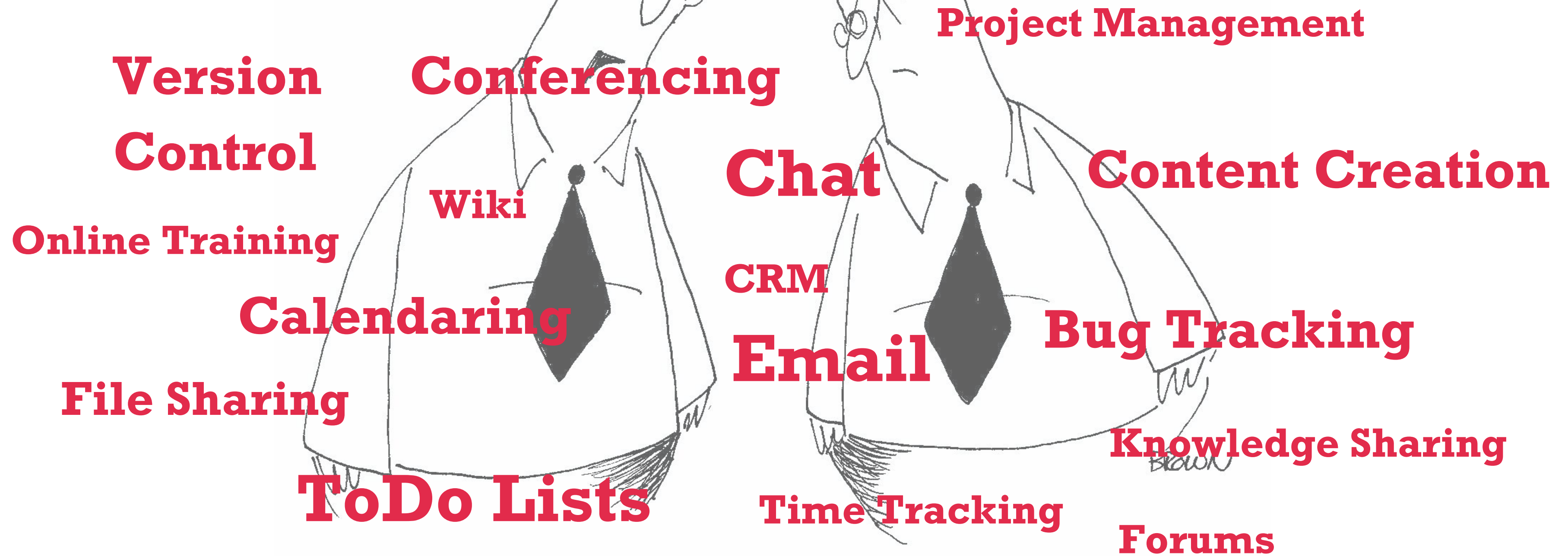
18% of Workers Are Fully Remote, 52% Work From Home at Least Once a Week

smallbizgenius.net/by-the-numbers/remote-work-statistics





An Ecosystem of Needs and Tools



"Maybe we've been collaborating too much."



The Collaboration Spectrum

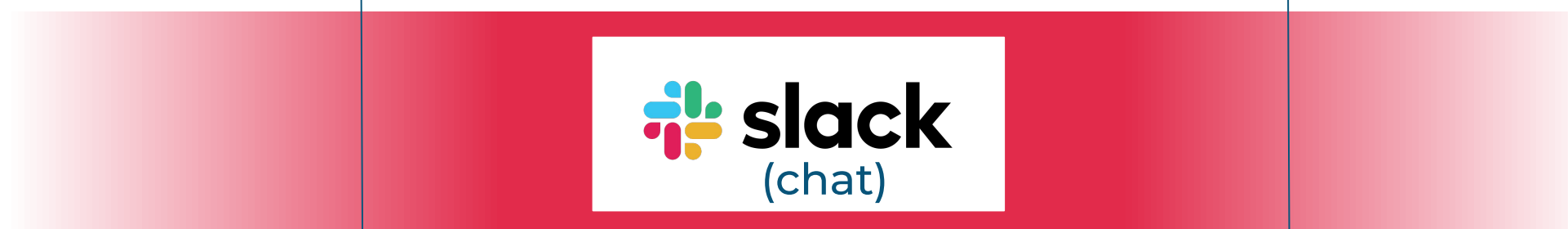
Synchronous vs. Asynchronous Collaboration



Synchronous



Synchron-ish



Asynchronous





The Next Episode

Shifting The Paradigm from Single User to Multi User Collaborative Apps

- Asynchronous Collaboration Tools are 20~ Years Old
Drupal, WordPress, Jive, SharePoint, etc.
- Synchronous Content Tools are ~10 Years Old
Google Docs and Quip
- Why Isn't Synchronous Collaboration Part of
Collaboration Software Suites, CMSs, etc.?



"You said I'd be part of a team."



Synchronous Collaboration & The CMS

Content Management Systems Aren't Awesome For Creating Content

- Content Creation is Core, But It Has Always Lagged Behind
- We Used to Author in MS Word Then Cut and Paste
- Better Editors Improved the Experience, Some
- Now People Use Google Docs and Similar Tools
- No CMS on the Market Works The Way We Do - Collaboratively

Until Now... Let's Talk About How We Are Making It Happen!



Requirements for Efficient Collaboration





Presence Awareness (PA) Indicators

Distributed Collaboration Requires Timely Insight Into Availability & Status

- SLACK: Online/Active? Snooze Notification? Typing Back?
- Zoom: Who's Speaking, Non Verbal Communication
- Google Docs: Who's Active, Where, What You Are Doing
- Jira: Can't See Changes Until They Are Made

As Awareness Decreases, Friction Increases





Change Management

Editorial Workflow Facilitates The Collaborative Content Creation Process

- In-line Commenting
- Suggest Changes & Track Changes
- Versioning and Reverting
- Local Undo Redo



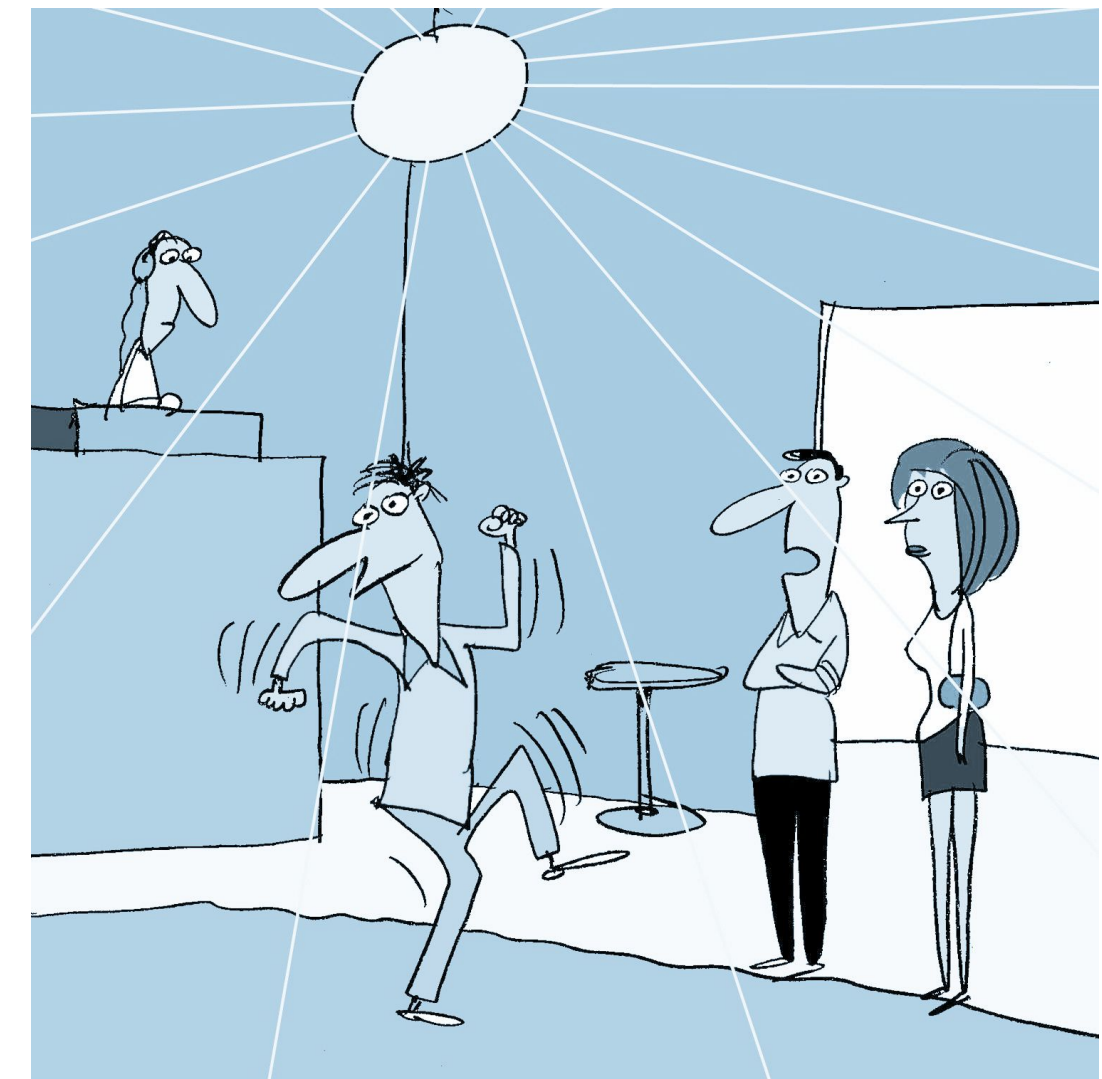


Algorithms

It's Complex & Requires Substantial Investment

- How Can Multiple People Edit Content And It All Makes Sense?
Conflict-free Replicated Data Types (CRDT)
Operational Transformation (OT)
- How Do Peers Communicate?
Networking: Client Server and or p2p
Protocol: TCP/IP, WebRTC

Why Build ALL Requirements In Each App?



*“Having algorithm and having rhythm
are 2 different things.”*



The Yjs Real-Time Collaboration Framework





Yjs

A Framework That Enables Collaboration in Any Application

- Indepth Research Identified Yjs as the Best Choice for Our Needs
- Open Source and It Works With Any Structured Data Types
- Provided PA, Workflow, and is Protocol Independent
- Kevin Jahns, The Project Lead, Willingness To Collaborate

We Successfully Implemented Shared Editing In Drupal





Real-Time Shared Editing Framework

*Collaborative Rich Text
Editing Demo*



How does that all even work?

Some theory to empower you!



What the heck is a CRDT?

Hint: It's a conflict-free replicated data type

<tag¹>



What is a CRDT?

I don't know what Conflict Free Replicated Data type means :(

Text editors work like this:

→ Insert H at position 15

CRDTs work like this:

→ Insert H after object with ID 1,12345

<tag¹>



What is a CRDT?

New York, New York!

Analogy in New York -- would you rather have instructions:

→ Go 3 blocks left, then 2 blocks up, then 4 blocks right, then ...

OR

→ Go to: "271 Bleecker St. New York, NY 10014"



Why 271 Bleecker St., New York?



**Apparently they
have the best
neapolitan Pizza!
:D**

(or so I heard)

<tag¹>



What is a CRDT?

Less Pizza, more code ...

- Yjs gives us CRDTs on common data types
- Example:

TODO LIST [every Vue / React / Svelte / Alpine / ...] example ever

"It's the new Hello World"



How can you use a CRDT?

Todo Lists are the new Hello World!

TODO LIST [every Vue / React / Svelte / Alpine / ..] example ever

- Data is stored in a Javascript Array
- It's an interactive app => If the array changes, the todo list changes
- Put it in a YArray, connect to a server and it's collaborative and real-time!

<tag¹>



How can you use a CRDT?

Todo Lists are the new Hello World!

TODO LIST [every Vue / React / Svelte / Alpine / ..] example ever

- Put it in a YArray, connect to a server and it's collaborative and real-time!
- It is as simple as that to make the list-part real-time (edits to the text are not)
- This ease of integration => The future of web applications

<tag¹>



How can you use a CRDT?

And Rich Text?

- For rich text the primitive is a YXMLDocument.
- We partnered with Kevin Jahns (author of Yjs) on helping us bring Yjs to Drupal as we love open source.
- And here an exciting journey began for me to learn all about collaborative text editing



Challenges for collaborative text editors

One tech to solve them all



Personal Undo/Redo

Undo alllllll the things ...

- Personal Undo/Redo is really important
- **Future idea:** Moderator takes over for a user and can undo on their behalf (after booting them from the session)



Personal Undo/Redo

Undo alllllll the things ...

→ Many challenges: create table, undo table, someone had written into the table in the meantime

=> What to do?



**Perfect conflict resolution is not
needed**



Personal Undo/Redo

Perfect conflict resolution is not needed!

- You don't need perfect conflict resolution
 - Because: Google Docs does not have it!
- => "Good enough" is enough! :)



Personal Undo/Redo

Testing the big players against Yjs!

Test case 1 -- Paste the largest Wikipedia document 7 times
+ undo / redo:

- Google docs chokes (7 seconds and more)
- Quip (by Salesforce) completely freezes
- Yjs is happily snappy

That was the moment when Kevin sold me on the technology ;)



Personal Undo/Redo

Testing the big players against Yjs!

Test case 2 -- Break a paragraph in two AND make it a list at the same time:

- Google docs offline mode has some really funny behavior
- Gets into a state where the list items are more left indented than the text, like this:

```
Text
* Indented
* Indented
Text
```

<tag¹>



Personal Undo/Redo

Testing the big players against Yjs!

Summary:

If it's good enough for Google Docs to be imperfect, it's good enough for us!

Awareness solves most conflict resolution problems as users tend to be careful not to step on each others' feet.



What about offline mode?

Airplanes in the Skyyyyy!

- Yjs can easily sync back after working offline on a document on e.g. a flight => no problem
- Yjs is offline-first
- If network connection between Americas and Europe broke
 - => Two teams could continue working in Silos, then sync back

<tag¹>



What about offline mode?

Robots vs. Humans

Automatic conflict resolution is tricky:

- They are perfectly "correct" mathematically and yet make no sense for human beings
- **Future of that:** Tools to track changes since you last viewed and selectively apply them where you want them.

(Foundation is there, but more work on it needs sponsors)

<tag¹>



Implementing Collaboration in Drupal

Editors, Workflows, Revisions and more



How did we implement all this?

Editor selection

- Don't needed to reinvent the wheel
- Yjs already supports several editors
- But CKEditor4 is not one of them :(



How did we implement all this?

Editor selection

- If your web applications uses Prosemirror or QuillJS you are set
- Yjs and prosemirror is a really great fit
- Journey to integrating a modern editor into Drupal



Editor integration

The largest challenge at first

Drupal's editor supports:

- Arbitrary HTML (Full HTML) mode
- Editor is fed with HTML
- Result is stored again as HTML



Editor integration

The largest challenge at first

Modern editors:

- Define a schema
- Editor is fed with JSON
- Store result as JSON



Editor integration

The largest challenge at first

Good choice: Atlassian editor (used in Confluence, Open Source based on Prosemirror):

- Editor is fed with ADF-JSON
- Store result as ADF-JSON
- We build upon that!



Editor integration

Modern vs Legacy - Two worlds clash

2 ways to solve this:

- - Render everything client-side
 - Use a JS render-based approach
 - Use NodeJS for SSR and search indexing
[good fit for decoupled applications].
- Render the editor contents to a hidden textarea
 - Textarea contains JSON that includes a html key
eg `{'html': "", 'json': {}}`
 - HTML is the rendered HTML (like from legacy editors!)

<tag¹>



Editor integration

Field Formatters save the day!

We used #2

- Use a custom field formatter - returns the 'html' part of the json
- Load just the JSON part into the Editor
- Export rendered HTML+JSON from the editor

=> This also let's search api work out of the box.

<tag¹>



YEAH!

<tag¹>



What about Full HTML?



What about Full HTML?

A schema clashes with full html support

Easiest way:

→ Just don't use it

<tag¹>



What about Full HTML?

A schema clashes with full html support

Other way:

- Use sparingly (it usually decreases accessibility)
- Create rich widgets instead
- Create a catch-all block type that displays raw HTML as unformatted code in the editor

*[optionally with syntax highlighting via CodeMirror!]
(<https://prosemirror.net/examples/codemirror/>)*

<tag¹>



Biggest Challenge



Non Technical



**How do many users
work in a system
designed for a single
user?**



Revisions + Content workflow + Track changes

- How do you integrate that with multiple users?
- Who is owning the document?

Fun fact: Wikipedia could have had simple text-based collaboration since years ago, but philosophical challenges, not technical are what has been preventing that so far.

<tag¹>



Revisions + Track changes

A technical solution

Yjs has the full history embedded in it's YDocument (minus some garbage collection)

- Every state of the document can be accessed by a "state vector"
- Whenever a new revision is created => Store not only HTML+JSON, but also the state vector
- Frequently save the (large) YDocument to persistent storage (auto-save ala Google Docs)

<tag¹>



Revisions + Track changes

A technical solution

Yjs document + JSON representation + HTML

- Revisions in Drupal are mapped to Revisions in the collaborative editor (YEAH!)
- A revision has multiple authors, but one person who published it
- Saving and Publishing is **explicit** (someone pushes the Publish button once ready)

<tag¹>



Revisions + Track changes

A technical solution

Open questions:

→ What about several drafts of content (git branches)?

=> In theory Yjs can merge those as it is offline-first, in practice no one tried, yet

→ What about garbage collection? (at one point older history needs to be pruned)

=> Organizational decision how to archive and prune.

<tag¹>



More challenges



Authentication + Access control

A technical solution

How do you ensure your document stays safe:

- Simplest way: hash_hmac an access token that is per entity UUID
- Entity UUID => `hash_hmac($UUID, Secret Key)` => Secret hash per document

Only they - who haveth the hash - can access the document!

<tag¹>



Authentication + Access control

So many more possibilities ...

Future work:

- Viewer only access
- Comment-only access

=> *Need different hashes and a way to authenticate editors vs. viewers*



Speaking of comments



Implementing comments

Construction in New York - again!

How do you store comments?

→ Out of band as a selection on a portion of text

Think: *Construction work from 781 Fifth Avenue to 795 Fifth Avenue*

→ The actual comment data is stored as a normal comment in Drupal, the selection only contains a reference to the comment entity id

<tag¹>



How does the infrastructure needed look like?



Infrastructure

What do you need to implement this?

Two ways to integrate:

- Enterprise: NodeJS with a redis backend and persistent backing store (MySQL database) => For our enterprise client
- Personal: Signaling server, WebRTC + Peer2Peer communication (No NodeJS needed to add to your stack)

=> Solution being developed for Wordpress (!)

Full podcast: <https://tag1.com/yjs/WebRTC>

<tag¹>



WordPress is ahead in the race

At least: right now

Anyone wants to bring Drupal on-par or start a Core initiative?

=> Contact us (yjs@tag1.com)

=> Watch our podcasts: tag1.com/yjs

=> Happy to share more details

→ **WordPress Gutenberg demo already looks really impressive**

Maybe Gutenberg in Drupal is a way forward?



Implementing Collaboration In WordPress

Let's take a look!

Yjs WP & Gutenberg

Yjs was integrated with Gutenberg in WordPress as a proof of concept to demonstrate it is possible to implement shared editing well ahead of the planned Gutenberg roadmap.



Yjs WP & Gutenberg

Yjs was integrated with Gutenberg in WordPress as a proof of concept to demonstrate it is possible to implement shared editing well ahead of the planned Gutenberg roadmap.





Gutenberg Prototype in < 24 hours!

Kevin rocks!

- Kevin Jahns (Yjs) created this integration with Gutenberg in less than a day (WOW!)
- With a modern editor integration of Yjs is pretty easy.

See for yourself:

<https://github.com/WordPress/gutenberg/pull/18357>

<tag¹>



Open Source is eating the world

Pizza anyone?

→ Since recording the demo, a better implementation by a former Automattic employee has appeared:

- * Comments

- * Awareness of other users (user list, shared cursors)

→ Check out some animated GIFs and screenshots here:

<https://github.com/WordPress/gutenberg/pull/23129>

<tag¹>



WordPress has collab on the roadmap!

It's a first class citizen

- Goal: Enable every WordPress instance
 - from the tiniest shared hosting accounts to the largest enterprise sites –
- to support p2p real-time shared editing without the sites installing any additional software.



WordPress has collab on the road-map!

It's a first class citizen

There are two components to achieving this:

- Integrating WordPress & Gutenberg with Yjs (in-progress!)
- Setting up and Integrating a Peer2Peer backend infrastructure



Peer2Peer is hard

Many commercial VoIP systems still use relay servers

Without too many details:

- Signaling server to even find other users
(or long-poll or ajax request or ...)
- STUN and TURN to push through firewalls
- WebRTC to bring it all together



**But: We will hopefully be
working on it :)**



Collaboration & The Future of Applications

Imagine the possibilities!



Every Application Will Offer Collaboration

The Shift Towards Multi User Collaborative Applications is Happening Now

- Advent of Collaboration Frameworks Will Foster Rapid Adoption
- Robust Tooling Will Enable More Effective Collaboration
- Better Tooling Will Enable Applications to be More Robust



CMS Collaboration Goes Beyond Text

Collaboration Will Permeate Content Management Systems

- Shared Editing of Documents Is Only The Beginning
- Layout Building & Theming
- Site Building and Administrative Tasks
- Enhanced Features e.g., Whiteboarding, Shared Lists in a Block



What about Drupal specifically?



Imagine the possibilities

Real-time collaboration everywhere

Just imagine:

- Real-time layout builder
- Real-time text editing in Drupal
- Real-time (admin) notifications
- Seeing who else is online



Imagine the possibilities

Real-time collaboration everywhere

BUT :(

For real-time in Drupal so far the options have been mostly:

- Firebase (commercial, costly)
- NodeJS with channels
(pretty hard to use, needs NodeJS added to the Stack)



Imagine the possibilities

Real-time collaboration everywhere

- But with the peer2peer support we could have a real-time enabled Drupal out-of-the box -- not only for the editor, but for the whole system (!).
- It's a completely new way to think once we have these possibilities.



**Let's make the future of web
applications happen!**

<tag¹>



It has already started!



Q&A

Questions? yjs@tag1.com



Want to Learn More? [Tag1.com/Yjs](https://tag1.com/yjs)

Yjs Real-Time Collaboration Framework Documentation & Resources

Code: [Yjs.dev](https://yjs.dev)

Blogs, Podcasts, etc:

[Tag1.com/Yjs](https://tag1.com/yjs)

Free Open Source Support:

[Github.com/yjs](https://github.com/yjs)

Paid Support & Development

Contact us at Yjs@Tag1.com

Demos:

→ Yjs 2 Min. Demo Video

https://youtu.be/_OwHt1lsyKk

→ YJs & Gutenberg in WP Demo

<https://gutenberg.yjs.dev/>

→ Live Demos for Other Editors

<https://yjs.dev/#demos>

→ Whiteboarding

(see “drawing” tab on live demos, above)



DrupalCon^{GLOBAL}

The Open Source Digital Experience Conference

ONLINE EVERYWHERE

2020 JULY 14-17