FOSDEM

Yjs 14 & BlockNote update

Suggestions and Attributions

Nick Perez, Kevin Jahns, Yousef El-Dardiry

FOSDEM

Yjs 14 & BlockNote update

Suggestions and Attributions

Nick Perez, Kevin Jahns, Yousef El-Dardiry

Zen DiS

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

LaSuite Docs

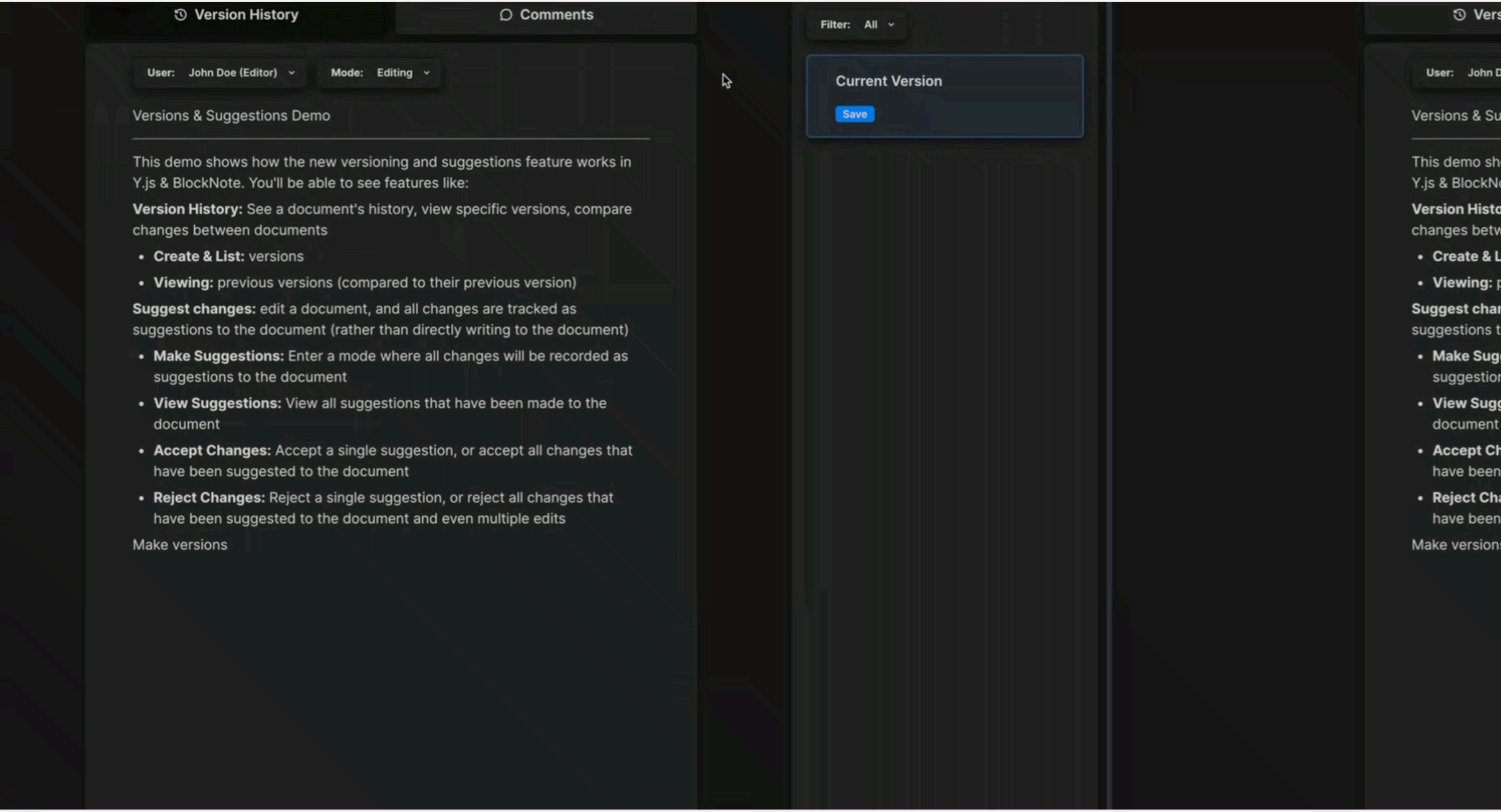
Overview

- Introduction
- Demo
- Architectural overview
- Updates in Yjs, y-prosemirror and BlockNote
- Looking ahead

Introduction

Adding support to Yjs, y-prosemirror and BlockNote for:

- **Suggestions (track changes):** Proposing changes to a document without applying them immediately.
- **Attributions (detailed version history):** Tracking who made what change and when it happened.


Version History
Comments
User: John Doe (Editor)
Mode: Editing
Versions & Suggestions Demo
This demo shows how the new versioning and suggestions feature works in Y.js & BlockNote. You'll be able to see features like:
Version History: See a document's history, view specific versions, compare changes between documents
Create & List: versions
Viewing: previous versions (compared to their previous version)
Suggest changes: edit a document, and all changes are tracked as suggestions to the document (rather than directly writing to the document)
Make Suggestions: Enter a mode where all changes will be recorded as suggestions to the document
View Suggestions: View all suggestions that have been made to the document
Accept Changes: Accept a single suggestion, or accept all changes that have been suggested to the document
Reject Changes: Reject a single suggestion, or reject all changes that have been suggested to the document and even multiple edits
Make versions
Filter: All
Current Version
Save

Anti-pattern: Store suggestions directly

Bad idea, because we want:

Decentralized permissions

Enable suggestion-only users without centrally inspecting every update.

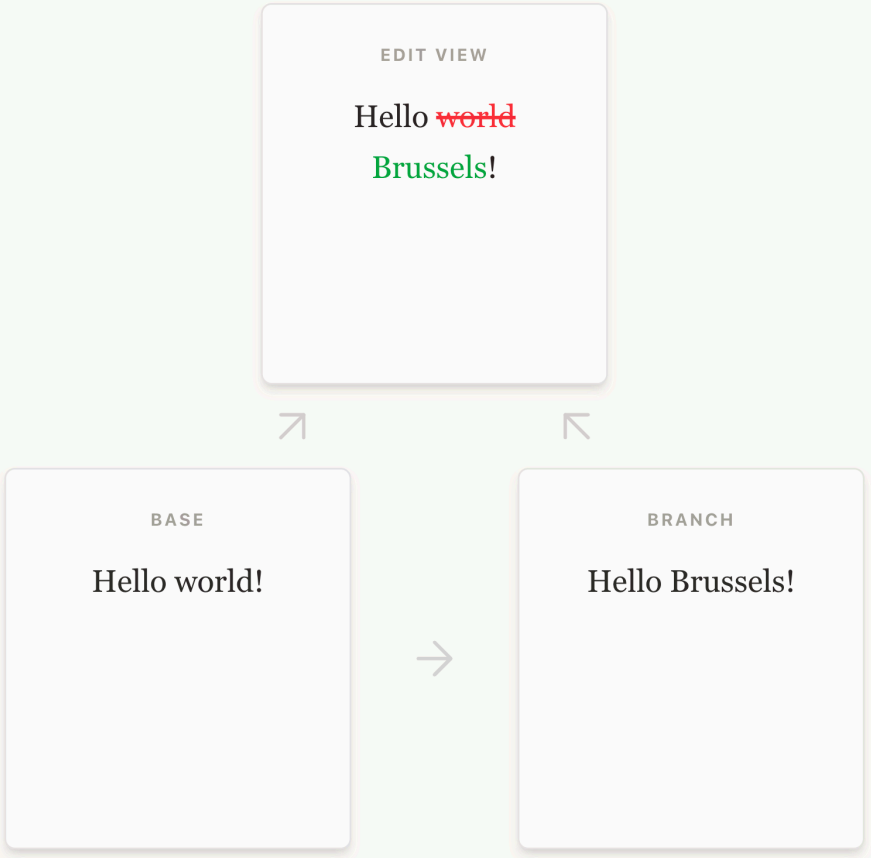
Decoupling of versions

Allow novel collaborative user experiences.

Solution: model similar to Git branches

Suggestions are like branches, **except:**

- Changes to the base are always shown live in the branch
- Different user modes:
  - **Suggestion mode:** edits go to fork
  - **Edit mode without suggestions:** edits go to base
  - **Edit mode + show suggestions:** edits go to ???


EDIT VIEW
Hello world
Brussels!
BASE
Hello world!
BRANCH
Hello Brussels!

Attributions: character-level metadata

In Yjs, every character is uniquely addressable. With attributions, we can attach metadata to specific ranges.

- **IDMap<ID, metadata>:** a new data structure that efficiently maps IDs to attributes (metadata).

Metadata can be application-defined:

- Author information
- Timestamps
- Current weather 🌤️

What's new in Y.js 14

A deep dive into the architectural changes

Unified Y.Type
One type to rule them all + Schema

Attribution Manager
Track who wrote what

Content Renderer
Diffs & attribution display

@y/prosemirror
The editor bridge, reimagined

An update **6 years** in the making

Unified Y.Type

MAPPING TO XML

Y.Text	<>text content</>
Y.Array	<><item /><item /></>
Y.Map	<map key="value" />
Y.XmlElement	<name key="val">...</name>

BEFORE (V13)

Y.Text

Y.Array

Y.Map

Y.XmlElement

AFTER (V14)

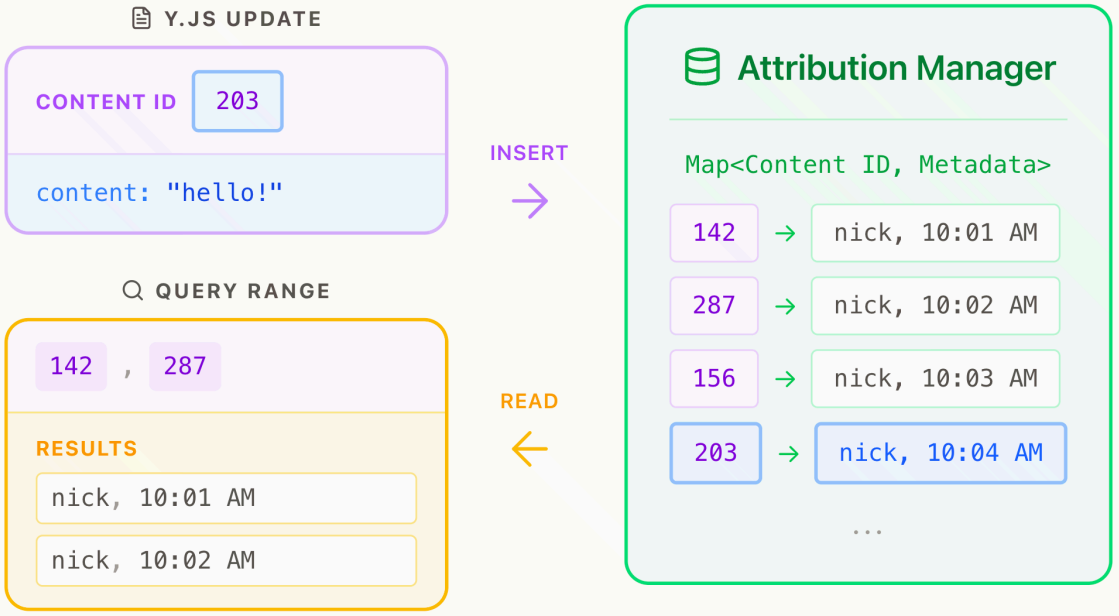
Y.Type + Schema

Deltas: A Universal Change Format

Deltas can...

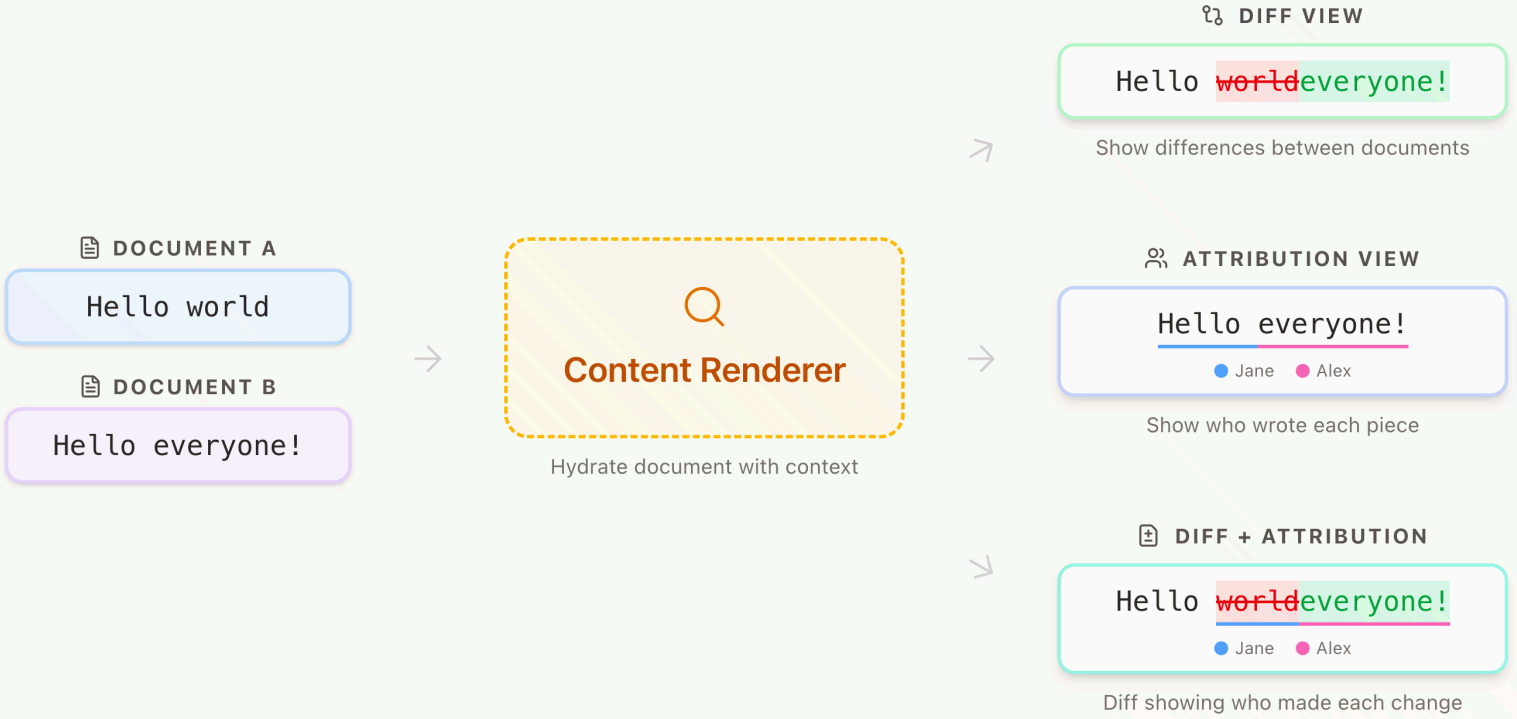
- be diffed
- be merged
- describe changes

Attribution Manager

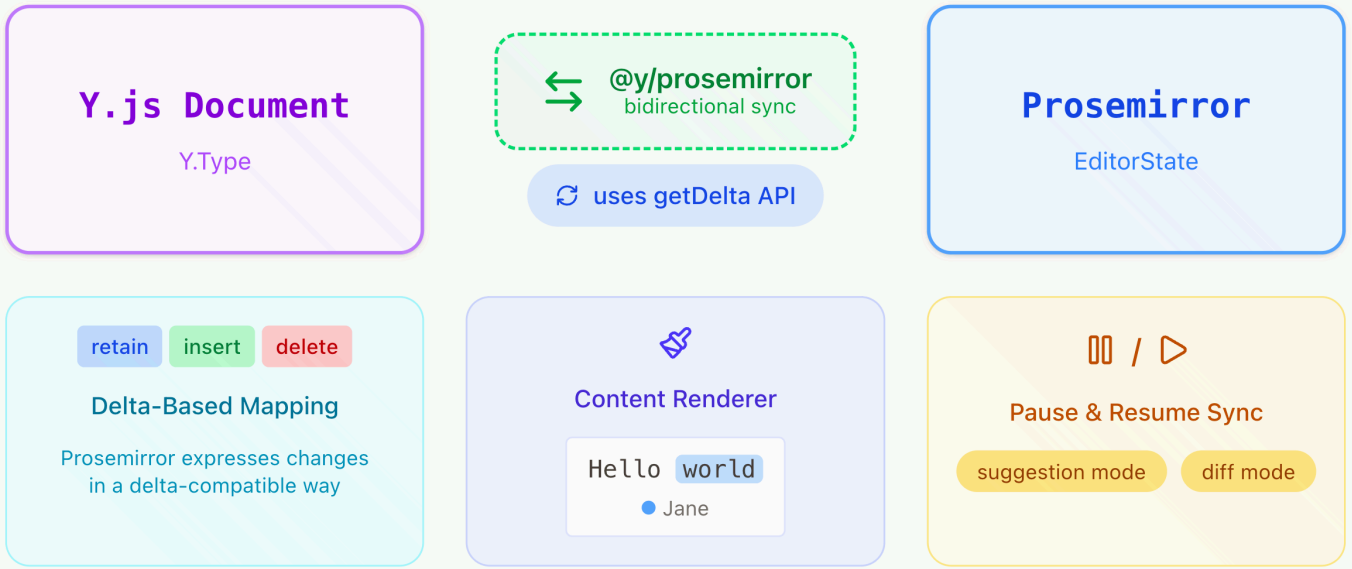


A **Content ID** is a set of inserted and deleted Lamport timestamps that uniquely identify a piece of content

Content Renderer: Views on documents



@y/prosemirror: The Bridge



Thank you!

Yjs 14 preview available under `@y/y`

y-prosemirror and BlockNote updates later this year

team@blocknotejs.org