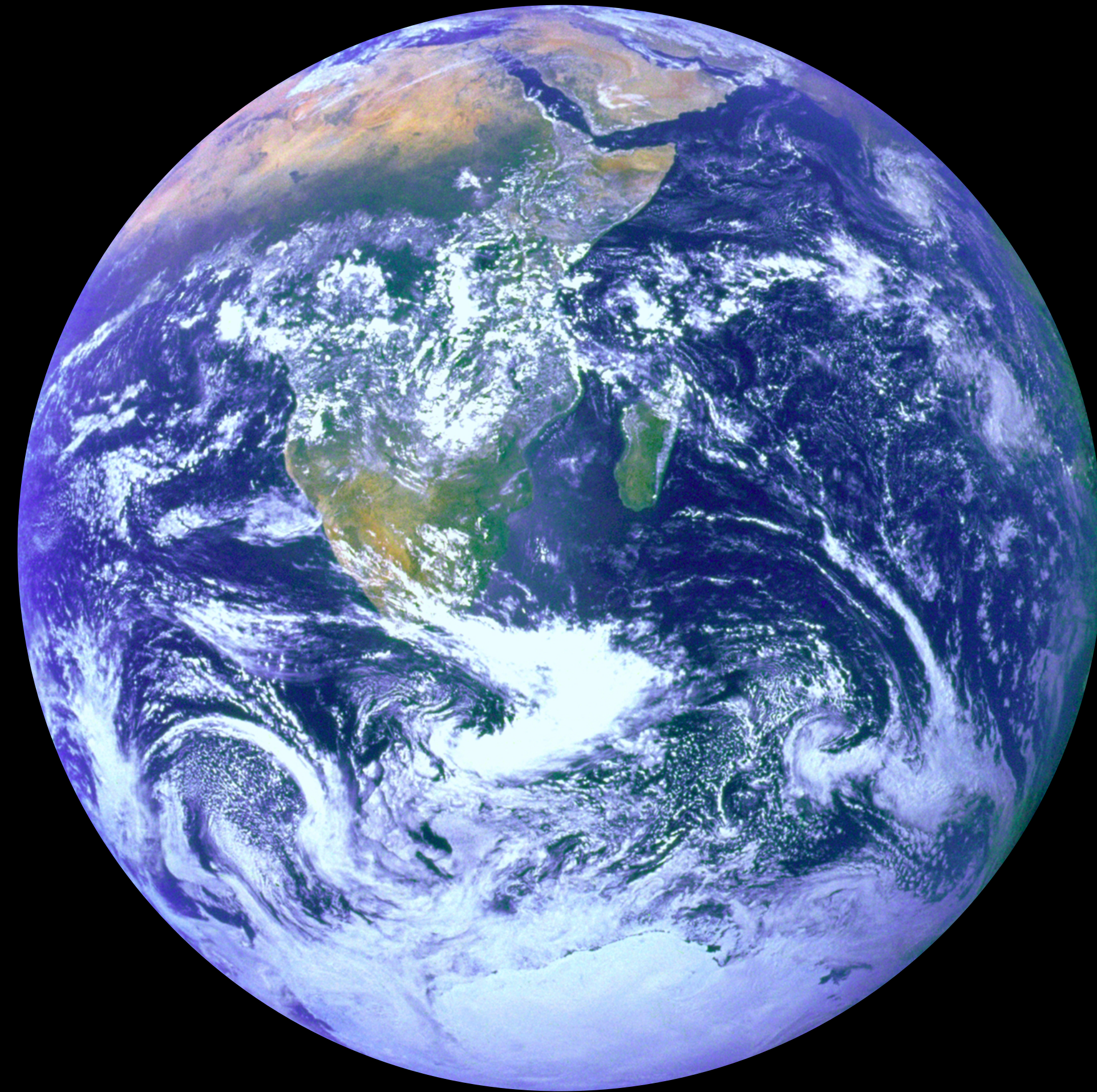


visual editor

The present and future of editing our wikis



Overview

Image: https://commons.wikimedia.org/wiki/File:The_Blue_Marble.jpg – public domain original from NASA; adjusted by User:Deglr6328

Why a visual editor?

Why a visual editor?

- Drop in new contributors is a major issue for Wikimedia

Why a visual editor?

- Drop in new contributors is a major issue for Wikimedia
- Wikitext stops people becoming contributors or staying

Why a visual editor?

- Drop in new contributors is a major issue for Wikimedia
- Wikitext stops people becoming contributors or staying
- Not an appropriate environment for the 21st Century

Why a visual editor?

- Drop in new contributors is a major issue for Wikimedia
- Wikitext stops people becoming contributors or staying
- Not an appropriate environment for the 21st Century
- Prevents us from providing other helpful editing tools

Development history

Development history

- Discussion for many years since 2004

Development history

- Discussion for many years since 2004
- Became a core goal in mid-2011 and we started work

Development history

- Discussion for many years since 2004
- Became a core goal in mid-2011 and we started work
- Working out if we can do it

Development history

- Discussion for many years since 2004
- Became a core goal in mid-2011 and we started work
- Working out if we can do it
- Iterative code development based on feedback & data

Deployment history

Deployment history

Available:

Deployment history

Available:

- ... on a wiki (mediawiki.org) from Jun 2012

Deployment history

Available:

- ... on a wiki (mediawiki.org) from Jun 2012
- ... for real edits by users (opt-in alpha) from Dec 2012

Deployment history

Available:

- ... on a wiki (mediawiki.org) from Jun 2012
- ... for real edits by users (opt-in alpha) from Dec 2012
 - English only at first; other Wikipedias since Apr 2013

Deployment history

Available:

- ... on a wiki (mediawiki.org) from Jun 2012
- ... for real edits by users (opt-in alpha) from Dec 2012
 - English only at first; other Wikipedias since Apr 2013
- ... for real edits by all users (opt-out beta) from Jul 2013

Deployment history

Available:

- ... on a wiki (mediawiki.org) from Jun 2012
- ... for real edits by users (opt-in alpha) from Dec 2012
 - English only at first; other Wikipedias since Apr 2013
- ... for real edits by all users (opt-out beta) from Jul 2013
 - English only at first; more Wikipedias since & to come



Architecture

Image: [https://commons.wikimedia.org/wiki/File:Arch_\(PSF\).png](https://commons.wikimedia.org/wiki/File:Arch_(PSF).png) – public domain original from Pearson Scott Foresman

How it works

How it works

VisualEditor:

- JavaScript client-side editor of our HTML+RDFa

How it works

VisualEditor:

- JavaScript client-side editor of our HTML+RDFa
- Unique needs – rich data model, generated content

How it works

VisualEditor:

- JavaScript client-side editor of our HTML+RDFa
- Unique needs – rich data model, generated content

Parsoid:

- Node.js server-side wikitext \longleftrightarrow HTML parser service

How it works

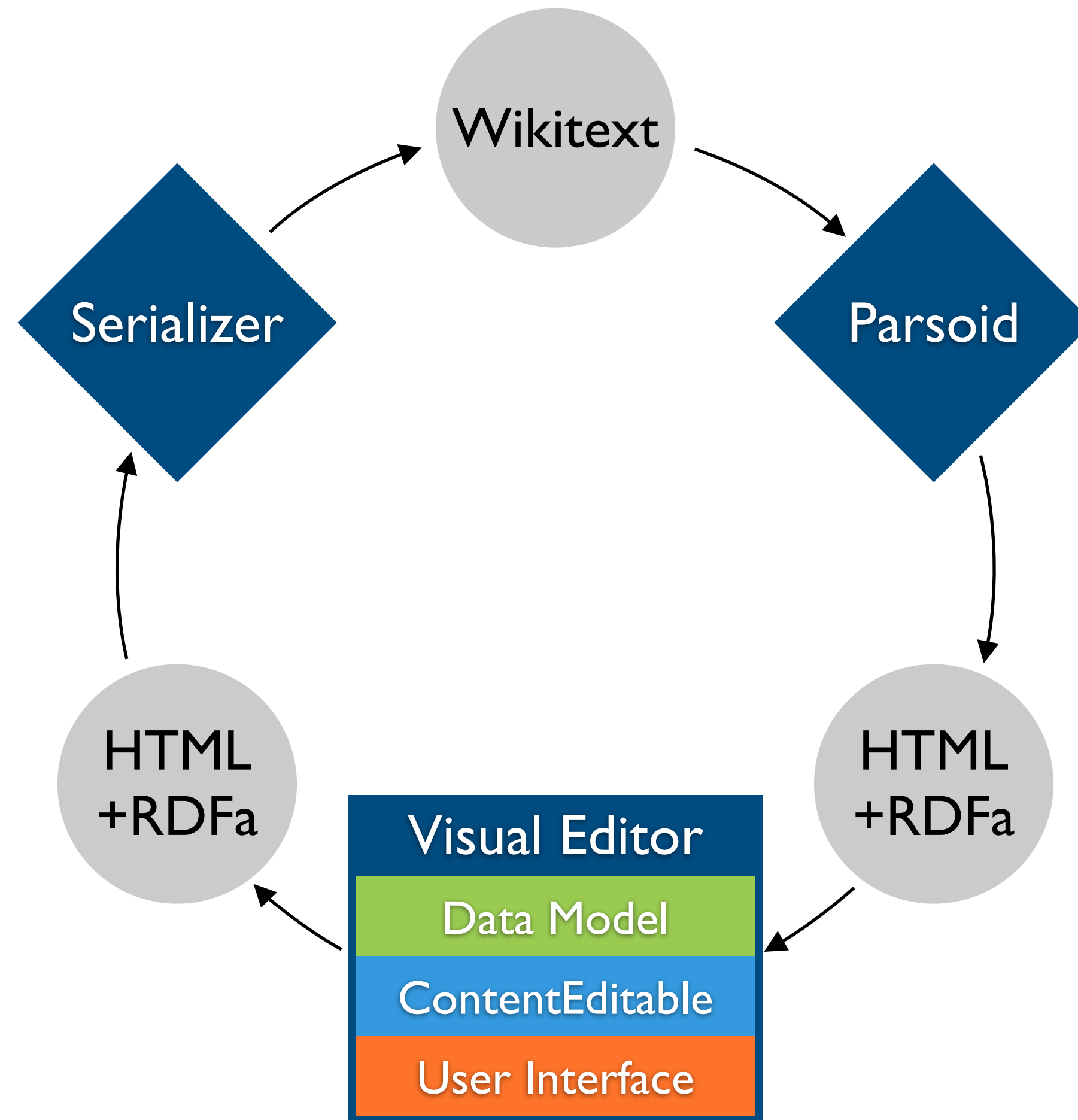
VisualEditor:

- JavaScript client-side editor of our HTML+RDFa
- Unique needs – rich data model, generated content

Paroid:

- Node.js server-side wikitext \longleftrightarrow HTML parser service
- Will eventually replace MediaWiki's core PHP parser

Anatomy of editing a page



Anatomy of editing a page

Anatomy of editing a page

- VisualEditor assets load when clicking Edit

Anatomy of editing a page

- VisualEditor assets load when clicking Edit
- Integration asks for the HTML from Parsoid

Anatomy of editing a page

- VisualEditor assets load when clicking Edit
- Integration asks for the HTML from Parsoid
- Data Model builds a linear model from this HTML

Anatomy of editing a page

- VisualEditor assets load when clicking Edit
- Integration asks for the HTML from Parsoid
- Data Model builds a linear model from this HTML
- Content Editable builds surface from the data model

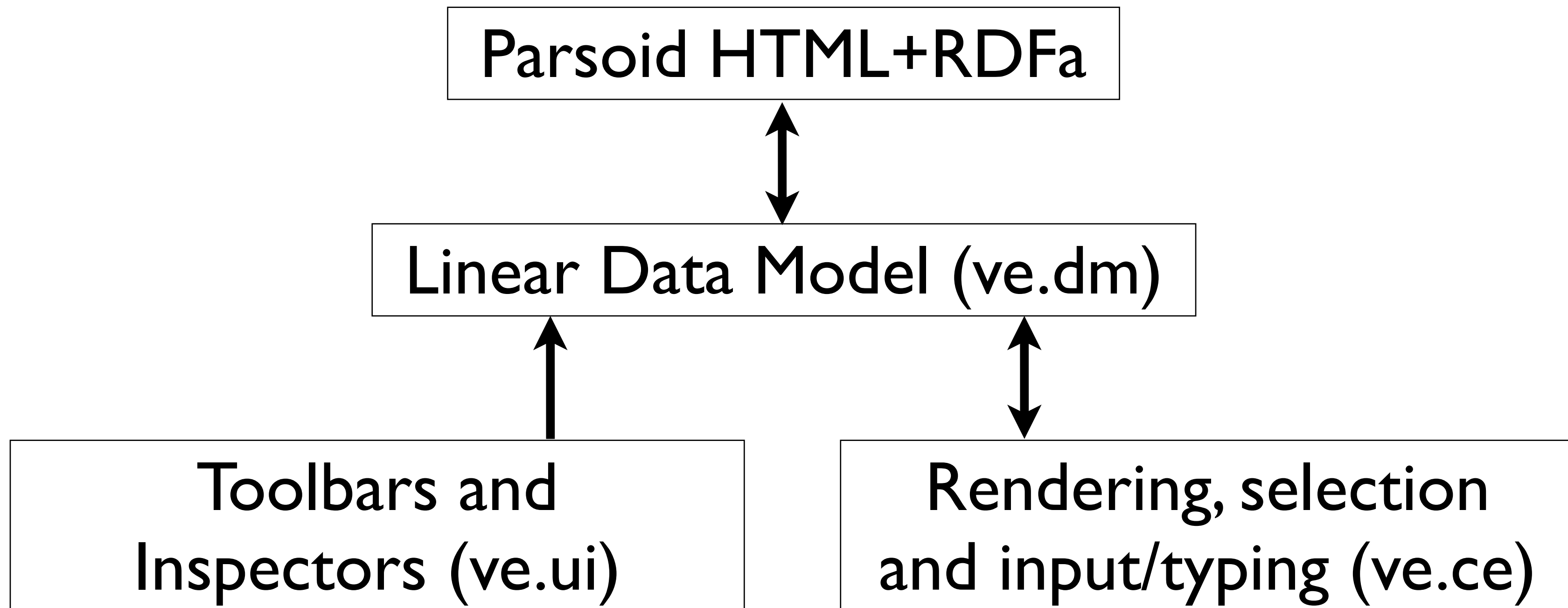
Anatomy of editing a page

- VisualEditor assets load when clicking Edit
- Integration asks for the HTML from Parsoid
- Data Model builds a linear model from this HTML
- Content Editable builds surface from the data model
- Editing tools give you a way to edit the surface

Anatomy of editing a page

- VisualEditor assets load when clicking Edit
- Integration asks for the HTML from Parsoid
- Data Model builds a linear model from this HTML
- Content Editable builds surface from the data model
- Editing tools give you a way to edit the surface
- The integration lets you save via Parsoid to MediaWiki

Transformations



Flexibility and modularity

Flexibility and modularity

- Designed to be very modular

Flexibility and modularity

- Designed to be very modular
- Can easily replace or extend existing tools

Flexibility and modularity

- Designed to be very modular
- Can easily replace or extend existing tools
- Even with a user script or a gadget

Flexibility and modularity

- Designed to be very modular
- Can easily replace or extend existing tools
- Even with a user script or a gadget
- Can expand for new content, uses and platforms



Data Model

Image: <https://commons.wikimedia.org/wiki/File:Xmatrix.png> – screenshot of XMatrix by Jamie Zawinski, licensed under the MIT/Expat licence



Outline

Outline

- We can't edit the Parsoid HTML+RDFA directly

Outline

- We can't edit the Parsoid HTML+RDFA directly
- Reliable conversion from HTML to a thing we can edit

Outline

- We can't edit the Parsoid HTML+RDFA directly
- Reliable conversion from HTML to a thing we can edit
- Round-trip back to HTML without corruption

Outline

- We can't edit the Parsoid HTML+RDFA directly
- Reliable conversion from HTML to a thing we can edit
- Round-trip back to HTML without corruption
- Synchronise with ContentEditable (both ways)

Outline

- We can't edit the Parsoid HTML+RDFA directly
- Reliable conversion from HTML to a thing we can edit
- Round-trip back to HTML without corruption
- Synchronise with ContentEditable (both ways)
- Needs to support real-time collaborative editing

Example

Example

- Gives us a linear view of the HTML tree:

```
<p>Hi</p>
```

→

```
[ {type: 'paragraph' }, 'H', 'i', {type: '/  
paragraph' } ]
```


Example

- Gives us a linear view of the HTML tree:

```
<p>Hi</p>
```

→

```
[ {type: 'paragraph' }, 'H', 'i', {type: '/  
paragraph' } ]
```

- Lets us create linear, fully reversible transactions (diffs)

Example

- Gives us a linear view of the HTML tree:

```
<p>Hi</p>
```

→

```
[ {type: 'paragraph' }, 'H', 'i', {type: '/  
paragraph' } ]
```

- Lets us create linear, fully reversible transactions (diffs)
- Keeps track of every change for undo/redo

Example

- Gives us a linear view of the HTML tree:

```
<p>Hi</p>
```

→

```
[{type: 'paragraph' }, 'H', 'i', {type: '/  
paragraph' } ]
```

- Lets us create linear, fully reversible transactions (diffs)
- Keeps track of every change for undo/redo
- ...and collaboration

Language challenges

Language challenges

- Data Model stores characters as an array of objects

Language challenges

- Data Model stores characters as an array of objects
- We changed from code units to grapheme clusters

Language challenges

- Data Model stores characters as an array of objects
- We changed from code units to grapheme clusters
- ãg (code units): ['n', '^', 'g']

Language challenges

- Data Model stores characters as an array of objects
- We changed from code units to grapheme clusters
- `ng` (code units): `['n', '^', 'g']`
- `ng` (grapheme clusters): `['n^', 'g']`

Language challenges

- Data Model stores characters as an array of objects
- We changed from code units to grapheme clusters
- `ng` (code units): `['n', '^', 'g']`
- `ng` (grapheme clusters): `['n^', 'g']`
- This doesn't work, so we're reverting to code units

Language issues

Language issues

- Supplementary characters: 𨳏

Language issues

- Supplementary characters: 𨳏
- Complex grapheme clusters: ༄འྐ

Language issues

- Supplementary characters: 𨋖
- Complex grapheme clusters: ༄འཁ
- Combining accents: façade

Language issues

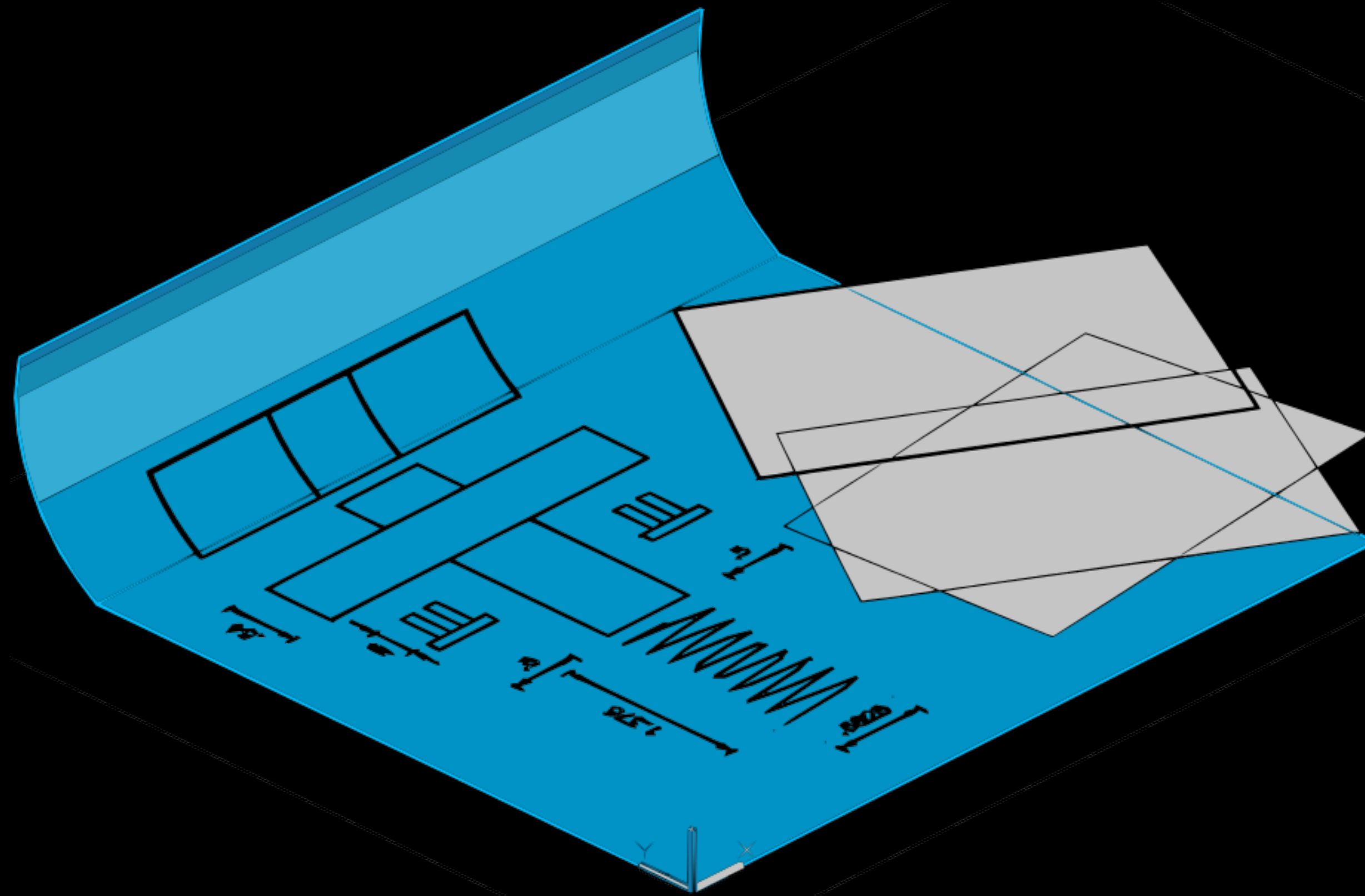
- Supplementary characters: 𨋖
- Complex grapheme clusters: ༄།།
- Combining accents: façade
- Bi-directional: עב2345|רית

Language issues

- Supplementary characters: 𨋖
- Complex grapheme clusters: ༄།།
- Combining accents: façade
- Bi-directional: עב2345|רית
- Input methods (East Asian, Indic, ...)

Language testing

We need *your* feedback!



Content Editable

Image: <https://commons.wikimedia.org/wiki/File:Polytechnic.svg> – Available under CC-BY-SA 3.0 by [[User:Μηχανικός I]]

Browser Content Editable

Browser Content Editable

- Messes up HTML

Browser Content Editable

- Messes up HTML
- Inconsistent key handling

Browser Content Editable

- Messes up HTML
- Inconsistent key handling
- Adds `
` tags

Browser Content Editable

- Messes up HTML
- Inconsistent key handling
- Adds `
` tags
- Native text insertion

Browser Content Editable

- Messes up HTML
- Inconsistent key handling
- Adds `
` tags
- Native text insertion
- Spellcheck

Browser Content Editable

- Messes up HTML
- Inconsistent key handling
- Adds `
` tags
- Native text insertion
- Spellcheck
- Selection & cursor

Browser Content Editable

- Messes up HTML
- Inconsistent key handling
- Adds `
` tags
- Native text insertion
- Spellcheck
- Selection & cursor
- Equal rendering

Programmatic cursor handling

Programmatic cursor handling

- Stop the browser's native cursor handling

Programmatic cursor handling

- Stop the browser's native cursor handling
- Prevent the cursor from entering protected elements

Programmatic cursor handling

- Stop the browser's native cursor handling
- Prevent the cursor from entering protected elements
- Allows us to skip over non-editable elements

Programmatic cursor handling

- Stop the browser's native cursor handling
- Prevent the cursor from entering protected elements
- Allows us to skip over non-editable elements
- Keyboard deleting is more predictable
(we have control)

SurfaceObserver

SurfaceObserver

- Poll for content changes and notify model

SurfaceObserver

- Poll for content changes and notify model
- Use compositionStart + compositionEnd

SurfaceObserver

- Poll for content changes and notify model
- Use `compositionStart` + `compositionEnd`
- May be replaced with native `MutationObservers`



Editing Tools

Image: https://commons.wikimedia.org/wiki/File:Blacksmith_anvil_hammer.svg – Public domain, uploaded by [[User: Midnightcomm]]

Actually editing stuff!

Actually editing stuff!

Text formatting

Actually editing stuff!

Text formatting

- “Annotations” – things like **bold**, *italic*, [link](#), ...

Actually editing stuff!

Text formatting

- “Annotations” – things like **bold**, *italic*, [link](#), ...
- Includes more complex ones which need inspectors

Actually editing stuff!

Text formatting

- “Annotations” – things like **bold**, *italic*, [link](#), ...
- Includes more complex ones which need inspectors

Generated content

Actually editing stuff!

Text formatting

- “Annotations” – things like **bold**, *italic*, [link](#), ...
- Includes more complex ones which need inspectors

Generated content

- References, templates, meta-data, media items, ...

Actually editing stuff!

Text formatting

- “Annotations” – things like **bold**, *italic*, [link](#), ...
- Includes more complex ones which need inspectors

Generated content

- References, templates, meta-data, media items, ...
- Where VisualEditor is unique – a generic capability

Current & future tools

Current & future tools

B ● Bold

Current & future tools

B ● Bold

I ● Italic

Current & future tools

B ● Bold

I ● Italic

↔ ● Link

Current & future tools

B ● Bold

I ● Italic

 ● Link

 ● Lists

Current & future tools

B ● Bold




I ● Italic

 ● Link

 ● Lists




 ● In/outdenting lists

Current & future tools

- B** ● Bold
- I* ● Italic
-  ● Link
-  ● Lists
-  ● In/outdenting lists

Coming soon:




Current & future tools

- B** ● Bold
- I* ● Italic
-  ● Link
-  ● Lists
-  ● In/outdenting lists

Coming soon:

- Aあ ● Language (`lang="en"`)

Current & future tools

- B** ● Bold
- I* ● Italic
-  ● Link
-  ● Lists
-  ● In/outdenting lists

Coming soon:

- Aあ* ● Language (`lang="en"`)
- U ● Underline/Strikethrough

Current & future tools

B ● Bold

I ● Italic

 ● Link

 ● Lists

 ● In/outdenting lists




Coming soon:

Aあ ● Language (`lang="en"`)


U ● Underline/Strikethrough

x^2 ● Superscript/Subscript

Current & future tools

- B** ● Bold
- I* ● Italic
-  ● Link
-  ● Lists
-  ● In/outdenting lists

Coming soon:

- Aあ* ● Language (`lang="en"`)
- U ● Underline/Strikethrough
- x^2 ● Superscript/Subscript
-  ● In/outdenting paragraphs

Current & future dialogs

Current & future dialogs

- Transclusion editing

Current & future dialogs

- Transclusion editing
- Reference editing/re-use






Current & future dialogs

- Transclusion editing
- Reference editing/re-use
- Reference list editing

Current & future dialogs

- Transclusion editing
- Reference editing/re-use
- Reference list editing
- Page settings

Current & future dialogs

-  ● Transclusion editing
-  ● Reference editing/re-use
-  ● Reference list editing
-  ● Page settings
-  ● Media inserting/settings

Current & future dialogs

- Transclusion editing
- Reference editing/re-use
- Reference list editing
- Page settings
- Media inserting/settings

Coming soon:

Current & future dialogs

- Transclusion editing
 - Reference editing/re-use
 - Reference list editing
 - Page settings
 - Media inserting/settings
- Coming soon:*
- Code (`<syntaxhighlight>`)

Current & future dialogs

 ● Transclusion editing


 ● Reference editing/re-use

 ● Reference list editing

 ● Page settings

 ● Media inserting/settings

Coming soon:

 ● Code (`<syntaxhighlight>`)

 ● Equation (`<math>`)

Current & future dialogs

 ● Transclusion editing


 ● Reference editing/re-use

 ● Reference list editing

 ● Page settings

 ● Media inserting/settings

Coming soon:

 ● Code (`<syntaxhighlight>`)

 ● Equation (`<math>`)

 ● Hieroglyphics (`<hieroglyphics>`)

Current & future dialogs

 ● Transclusion editing


 ● Reference editing/re-use

 ● Reference list editing

 ● Page settings


 ● Media inserting/settings

Coming soon:

 ● Code (`<syntaxhighlight>`)

 ● Equation (`<math>`)

 ● Hieroglyphics (`<hieroglyphics>`)

 ● Tables (`{| ... |- ... |}`)

Usability is key

Usability is key

- Editing tools are critical to using VisualEditor

Usability is key

- Editing tools are critical to using VisualEditor
- Continuous iteration for improvement

Usability is key

- Editing tools are critical to using VisualEditor
- Continuous iteration for improvement
- Unlike wikitext editing, discoverability is possible

Usability is key

- Editing tools are critical to using VisualEditor
- Continuous iteration for improvement
- Unlike wikitext editing, discoverability is possible
- Provide full functionality but push “norms”

Usability is key

- Editing tools are critical to using VisualEditor
- Continuous iteration for improvement
- Unlike wikitext editing, discoverability is possible
- Provide full functionality but push “norms”
- Balance ease of use with consistency

FUTURE

Next steps

What we're working on now

What we're working on now

- Main focus is stability, usability, and performance

What we're working on now

- Main focus is stability, usability, and performance
- Language features (IME, variants, section mark-up, ...)

What we're working on now

- Main focus is stability, usability, and performance
- Language features (IME, variants, section mark-up, ...)
- Editing of tables, galleries, formulæ, code blocks, ...

What we're working on now

- Main focus is stability, usability, and performance
- Language features (IME, variants, section mark-up, ...)
- Editing of tables, galleries, formulæ, code blocks, ...
- Support editing on mobile, as part of Flow, and others

What we're working on now

- Main focus is stability, usability, and performance
- Language features (IME, variants, section mark-up, ...)
- Editing of tables, galleries, formulæ, code blocks, ...
- Support editing on mobile, as part of Flow, and others
- Make available for other Wikimedia wikis & 3rd parties

visual editor

Questions?

visual editor

The present and future of editing our wikis