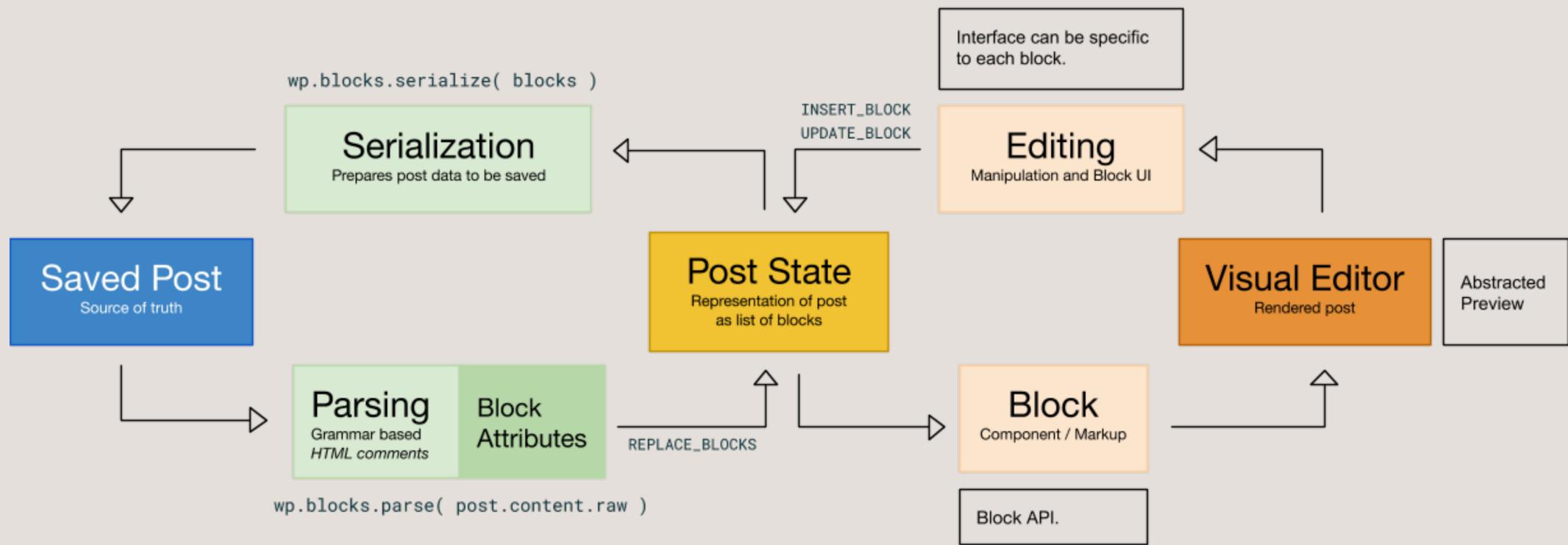
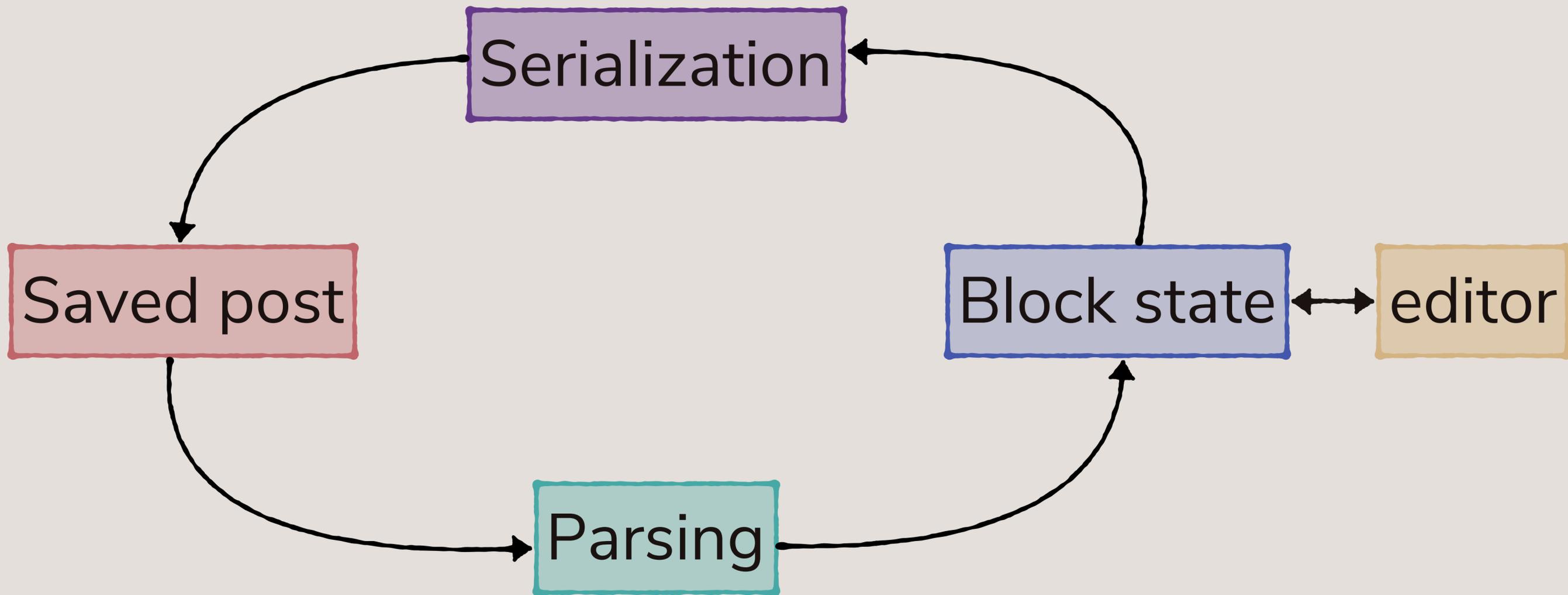


breaking apart block data

@ryelle





anatomy of a block

Demo

This is a test

[Cheese Babka](#)

December 31, 2018

[Blueberry Skillet Cornbread](#)

September 22, 2018

[Puff Pastry](#)

February 3, 2018



Baking bread in a cast
iron pot

Preview Publish... ⚙️ ⋮

View

- Top Toolbar
Access all block and document tools in a single place
- Spotlight Mode
Focus on one block at a time
- Fullscreen Mode
Work without distraction

Editor

- Visual Editor
- Code Editor ⌘⇧M

Tools

- Block Manager
- Manage All Reusable Blocks
- Keyboard Shortcuts ⌘⇧H
- Copy All Content

Options

anatomy of a block

Demo

This is a test

[Cheese Babka](#)

December 31, 2018

[Blueberry Skillet Cornbread](#)

September 22, 2018

[Puff Pastry](#)

February 3, 2018



Baking bread in a cast
iron pot

Editing Code

Exit Code Editor ✕

Demo

```
<!-- wp:heading {"level":3} -->
```

```
<h3>This is a test</h3>
```

```
<!-- /wp:heading -->
```

```
<!-- wp:latest-posts {"postsToShow":7,"displayPostDate":true} /-->
```

```
<!-- wp:image {"id":602,"sizeSlug":"large"} -->
```

```
<figure class="wp-block-image size-large"><figcaption>Baking bread in a cast iron
```

```
pot</figcaption></figure>
```

```
<!-- /wp:image -->
```

register a block type

```
registerBlockType( 'ryelle/recipe', {  
  title: __( 'Recipe', 'rmb-recipe-block' ),  
  icon: <Icon icon={ BlockSVG } />,  
  category: 'widgets',  
  attributes: {...},  
  edit() {},  
  save() {},  
} )
```

saving a block

```
save( { attributes } ) {  
    const { content } = attributes;  
  
    return (  
        <p>{ content }</p>  
    );  
}
```

```
<!-- wp:paragraph -->  
<p>This is your editable content.</p>  
<!-- /wp:paragraph -->
```

```
save( { attributes } ) {  
  const { content, level } = attributes;  
  const tagName = "h" + level;  
  
  return <RichText.Content  
    tagName={ tagName }  
    value={ content } />;  
}
```

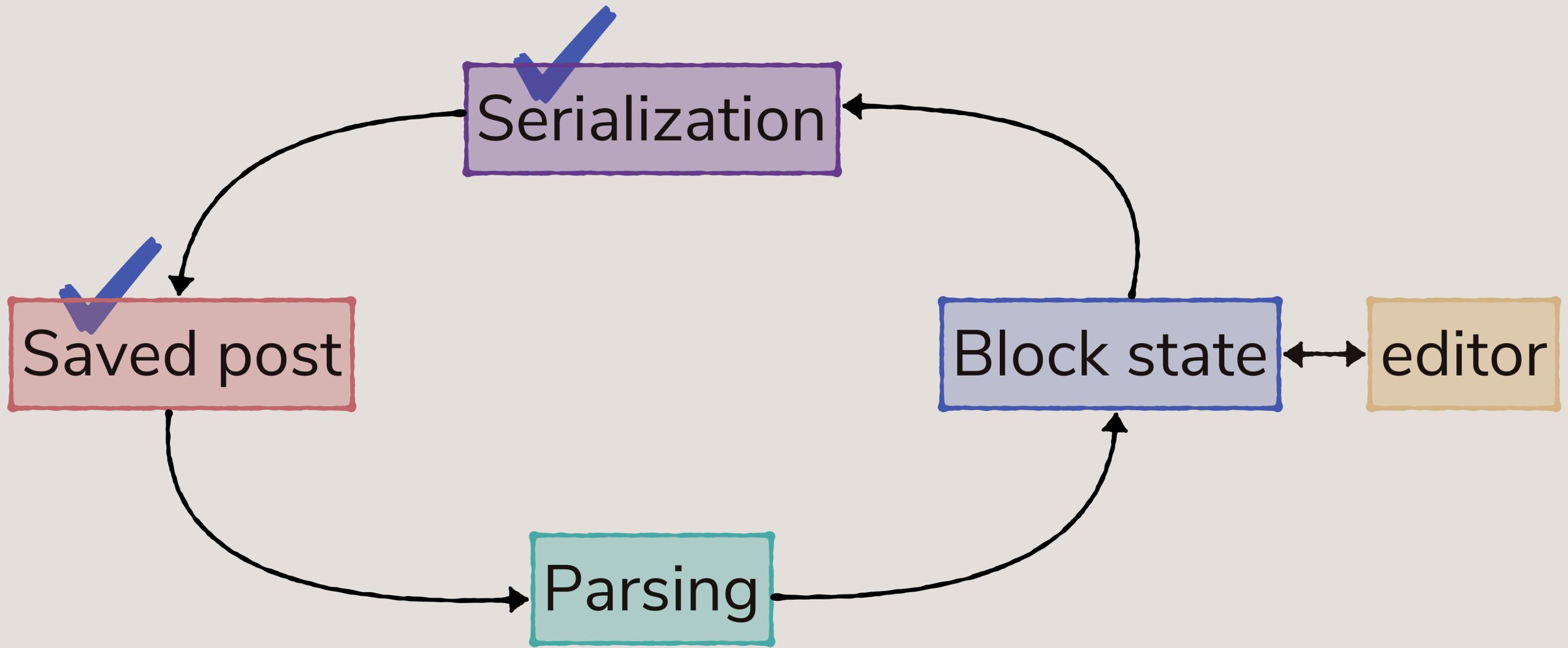
```
save( { attributes } ) {  
  const { categories = [] } = attributes;  
  return `[products on_sale=1 category="${ categories.join( ',' ) }"]`;  
},
```

```
<!-- wp:woocommerce/product-on-sale -->  
[products on_sale=1 category="5,11"]  
<!-- /wp:woocommerce/product-on-sale -->
```

```
save() {  
    return null;  
}
```

more detail

- [gutenberg/packages/blocks/src/api/serializer.js](#)
- `serializeBlock` given a block, return a string of content (runs `save`)
- [gutenberg/packages/element/src/serialize.js](#)
- `renderToString` given the result of `save`, convert this to a string



```
registerBlockType( 'ryelle/recipe', {
  title: __( 'Recipe', 'rmb-recipe-block' ),
  icon: <Icon icon={ BlockSVG } />,
  category: 'widgets',
  attributes: {...},
  edit() {},
  save() {},
} )
```

attributes

```
attributes: {
  url: {
    type: 'string',
    source: 'attribute',
    selector: 'a',
    attribute: 'href',
  },
  text: {
    type: 'string',
    source: 'html',
    selector: 'a',
  },
  backgroundColor: {
    type: 'string',
  },
}
```

attribute types

- string
- boolean
- integer or number
- object
- array

attribute sources

- default
- attribute
- text
- HTML
- query
- meta

source: default

```
attributes: {  
  level: {  
    type: "number",  
    default: 2,  
  },  
}
```

```
<!-- wp:heading {"level":3} -->  
<h3>Ingredients</h3>  
<!-- /wp:heading -->
```

```
attributes.level = 3
```

```
<!-- wp:heading -->  
<h2>Ingredients</h2>  
<!-- /wp:heading -->
```

```
attributes.level = 2
```

source: default

core block examples

- Heading block
- Latest Posts block

source: attribute



Attribute

```
<p class="editor-note">My cat is very grumpy</p>
```

The diagram illustrates the concept of an attribute in HTML. A white line with a vertical stem and horizontal bars at both ends points from the word "Attribute" to the "class" attribute in the HTML code snippet below. The code snippet is: `<p class="editor-note">My cat is very grumpy</p>`. The opening tag `<p` and closing tag `</p>` are in blue, while the attribute `class="editor-note"` is in yellow.

source: attribute

```
attributes: {  
  label: {  
    type: 'string',  
    default: '',  
    source: 'attribute',  
    selector: 'div',  
    attribute: 'aria-label',  
  },  
},
```

```
<!-- wp:ryelle/demo-attribute -->  
<div  
  aria-label="An editable label"  
  class="wp-block-ryelle-demo-attribute"  
>  
  Example block content. In a real situation,  
  this would probably be editable.  
</div>  
<!-- /wp:ryelle/demo-attribute -->
```

```
attributes.label = "An editable label"
```

source: attribute

```
attributes: {  
  hasCount: {  
    type: 'boolean',  
    default: true,  
    source: 'attribute',  
    selector: 'div',  
    attribute: 'data-has-count',  
  },  
}
```

```
<div data-has-count="false">
```

?

```
attributes.hasCount = true
```

```
<div data-has-count>
```

```
<div>
```

source: attribute

core block examples

- Audio block
- Button block

source: text

```
attributes: {  
  content: {  
    type: 'string',  
    default: '',  
    source: 'text',  
    selector: '.my-content',  
  },  
},
```

```
<!-- wp:ryelle/demo-text -->  
<div class="wp-block-ryelle-demo-text">  
  <p>This is unrelated content.</p>  
  <pre class="my-content">  
    This is the attribute value.  
  </pre>  
</div>  
<!-- /wp:ryelle/demo-text -->
```

```
attributes.content = "This is the attribute value."
```

source: text

core block examples:

- Code block

source: html

```
attributes: {  
  content: {  
    type: 'string',  
    default: '',  
    source: 'html',  
    selector: 'p',  
  },  
},
```

```
<!-- wp:ryelle/demo-html -->  
<div class="wp-block-ryelle-demo-html">  
  <p>  
    This is <em>content</em>  
    with <code>html</code> allowed.  
  </p>  
</div>  
<!-- /wp:ryelle/demo-html -->
```

```
attributes.content = "<p>This is <em>content</em>..."
```

source: html

```
<RichText
```

```
  tagName="p"
```

```
  value={ attributes.content }
```

```
  onChange={ ( content ) => setAttributes( { content } ) }
```

```
/>
```

```
<RichText
```

```
  tagName="div"
```

```
  multiline="p"
```

```
  value={ attributes.content }
```

```
  onChange={ ( content ) => setAttributes( { content } ) }
```

```
/>
```

source: html

```
attributes: {  
  content: {  
    type: 'string',  
    default: '',  
    source: 'html',  
    selector: 'p',  
  },  
},
```

```
<!-- wp:ryelle/demo-html -->  
<div class="wp-block-ryelle-demo-html">  
  <p>This is one line</p>  
  <p>and this is a second line</p>  
</div>  
<!-- /wp:ryelle/demo-html -->
```

source: html

```
attributes: {  
  content: {  
    type: 'string',  
    default: '',  
    source: 'html',  
    selector: 'div',  
    multiline: 'p',  
  },  
},
```

```
<!-- wp:ryelle/demo-html -->  
<div class="wp-block-ryelle-demo-html">  
  <p>This is one line</p>  
  <p>and this is a second line</p>  
</div>  
<!-- /wp:ryelle/demo-html -->
```

```
attributes.content = "<p>This is one line</p><p>and..."
```

source: html

core block examples

- Button block
- Quote block (both multiline and not)

source: query

```
attributes: {  
  content: {  
    type: 'array',  
    source: 'query',  
    selector: 'p',  
    query: {  
      children: {  
        type: 'string',  
        source: 'html',  
      },  
    },  
    default: [ {}, {} ],  
  },  
},
```

```
<!-- wp:ryelle/demo-query -->  
<div class="wp-block-ryelle-demo-query">  
  <div>  
    <p>This is one content item</p>  
  </div>  
  <div>  
    <p>This is another content item</p>  
  </div>  
</div>  
<!-- /wp:ryelle/demo-query -->
```

source: query

```
attributes: {  
  content: {  
    type: 'array',  
    source: 'query',  
    selector: 'p',  
    query: {  
      children: {  
        type: 'string',  
        source: 'html',  
      },  
    },  
    default: [ {}, {} ],  
  },  
},  
,
```

```
attributes.content = [  
  {  
    children: "This is one content item"  
  },  
  {  
    children: "This is another content item"  
  },  
]
```

source: query

core block examples

- Gallery block
- Text columns block (deprecated)

source: tag *

```
attributes: {  
  tag: {  
    type: 'string',  
    source: 'tag',  
    selector: '.my-content',  
    default: 'h2',  
  },  
},
```

```
<!-- wp:ryelle/demo-tag -->  
<h3 class="wp-block-ryelle-demo-tag my-content">  
  Alternative heading block  
</h3>  
<!-- /wp:ryelle/demo-tag -->
```

```
attributes.tag = "h3"
```

source: tag

core block examples

- Table block

source: meta

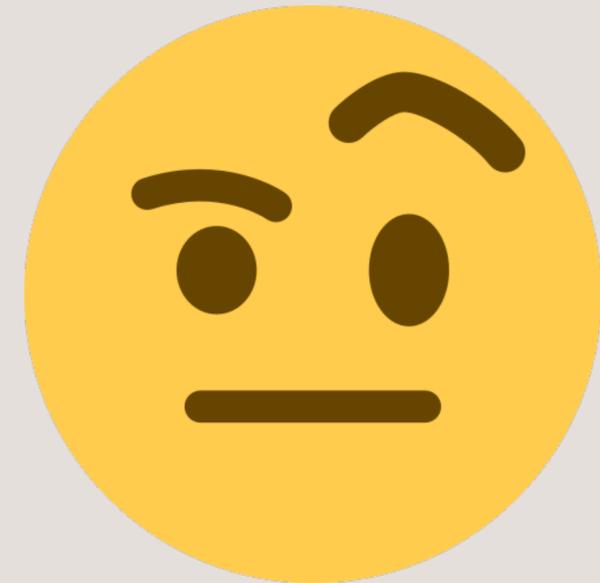
```
attributes: {  
  price: {  
    type: 'integer',  
    default: 0,  
    source: 'meta',  
    meta: 'demo_price',  
  },  
},
```

source: meta

```
function your_prefix_register_meta() {
    register_post_meta(
        'post', // post type
        'demo_price', // meta key
        array(
            'show_in_rest' => true,
            'single'       => true,
            'type'         => 'integer',
        )
    );
}
add_action( 'init', 'your_prefix_register_meta' );
```

source: meta

```
<!-- wp:ryelle/demo-meta -->  
<div class="wp-block-ryelle-demo-meta">  
    Cost $0  
</div>  
<!-- /wp:ryelle/demo-meta -->
```



source: meta

core block examples

- none!

custom sources

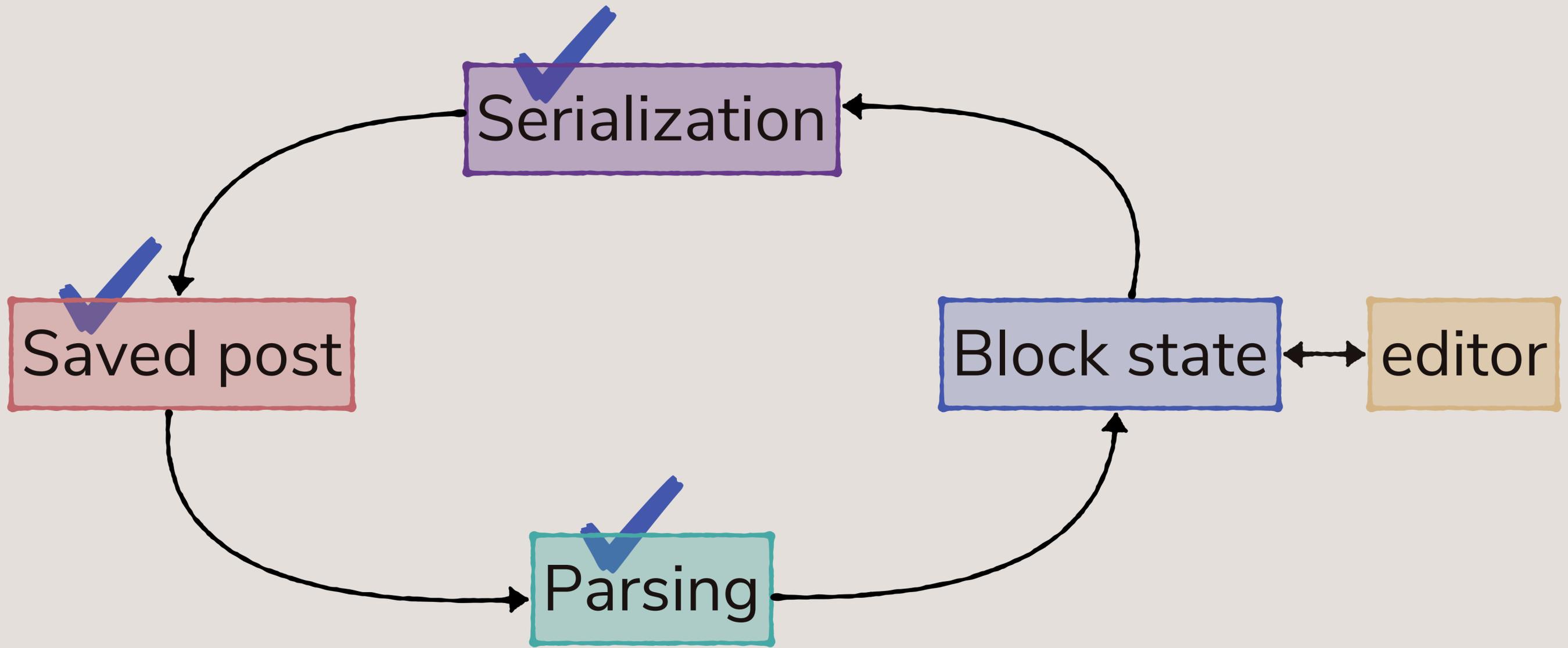
potentially, maybe, in the future?

`registerCustomSource?`

- does not exist (yet) #16282
- "custom sources" as an idea, for meta #16402

more detail

- [gutenberg/packages/blocks/src/api/parser.js](#)
- `getBlockAttribute` parses an attribute out of the block content, based on the attributes configuration
- hpq parser library: <https://aduth.github.io/hpq/>



Practice: poll block

Test Poll

What's your favorite fruit?

- Apple
- Orange
- Banana
- Strawberry
- Mango

Vote!

Document

Block

×



Example: Poll

Ask your readers for their opinions.

Settings



How many options are there?

5

Advanced



What's your favorite fruit?

- Apple
- Orange
- Banana
- Strawberry
- Mango

Vote!

<label>

<legend>

<button>

Test Poll

`attributes.question`
`html`

`attributes.count`
`default`

What's your favorite fruit?

`attributes.options`
`query`

- Apple
- Orange
- Banana
- Strawberry
- Mango

`options[].label`
`html`

Vote!

`attributes.submitLabel`
`text`

Example: Poll
Ask your readers for their opinions.

How many options are there?

```
count: {
  type: 'number',
  default: 3,
},
options: {
  type: 'array',
  source: 'query',
  selector: 'label',
  query: {
    label: {
      type: 'string',
      source: 'html',
    },
  },
  default: [ {}, {}, {} ],
},
```

```
question: {
  type: 'string',
  source: 'html',
  selector: 'legend',
},
submitLabel: {
  type: 'string',
  source: 'text',
  selector: 'button',
  default: 'Submit',
},
```

block validation

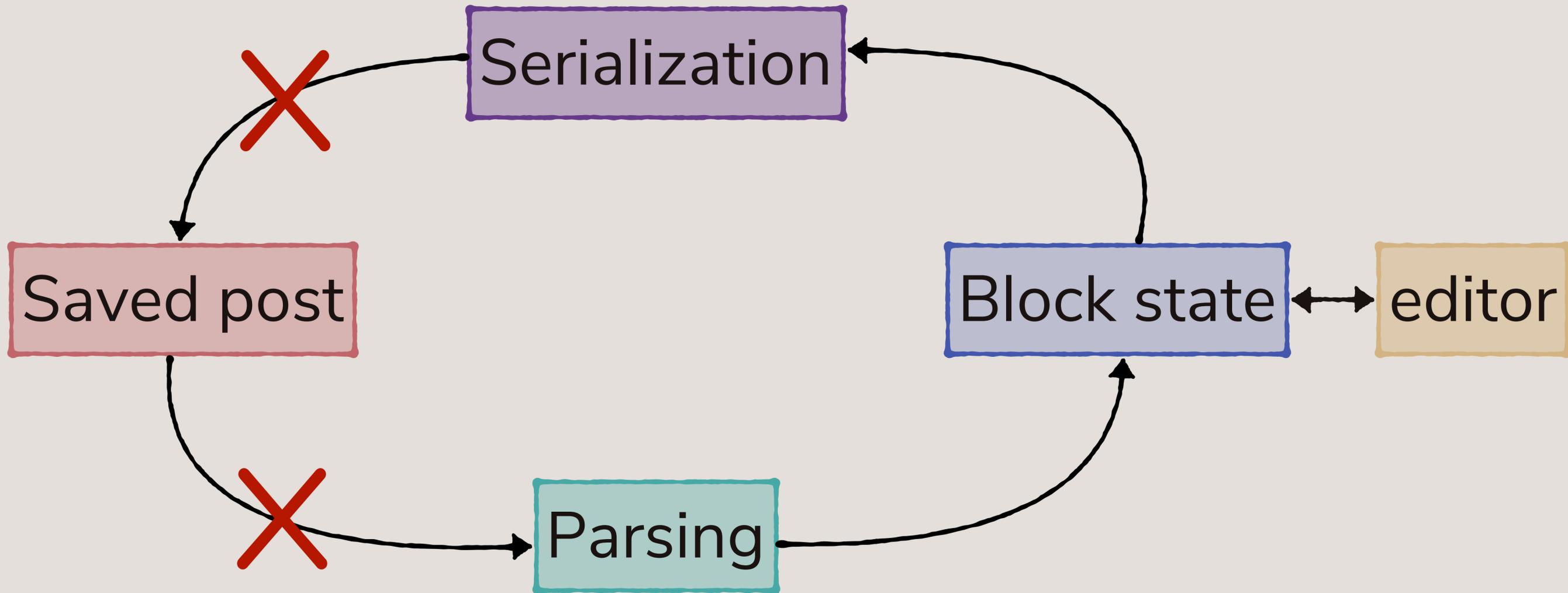
This block contains unexpected or invalid content.

Resolve

Convert to HTML



block content ~~≡~~ save



thank you!

@ryelle