



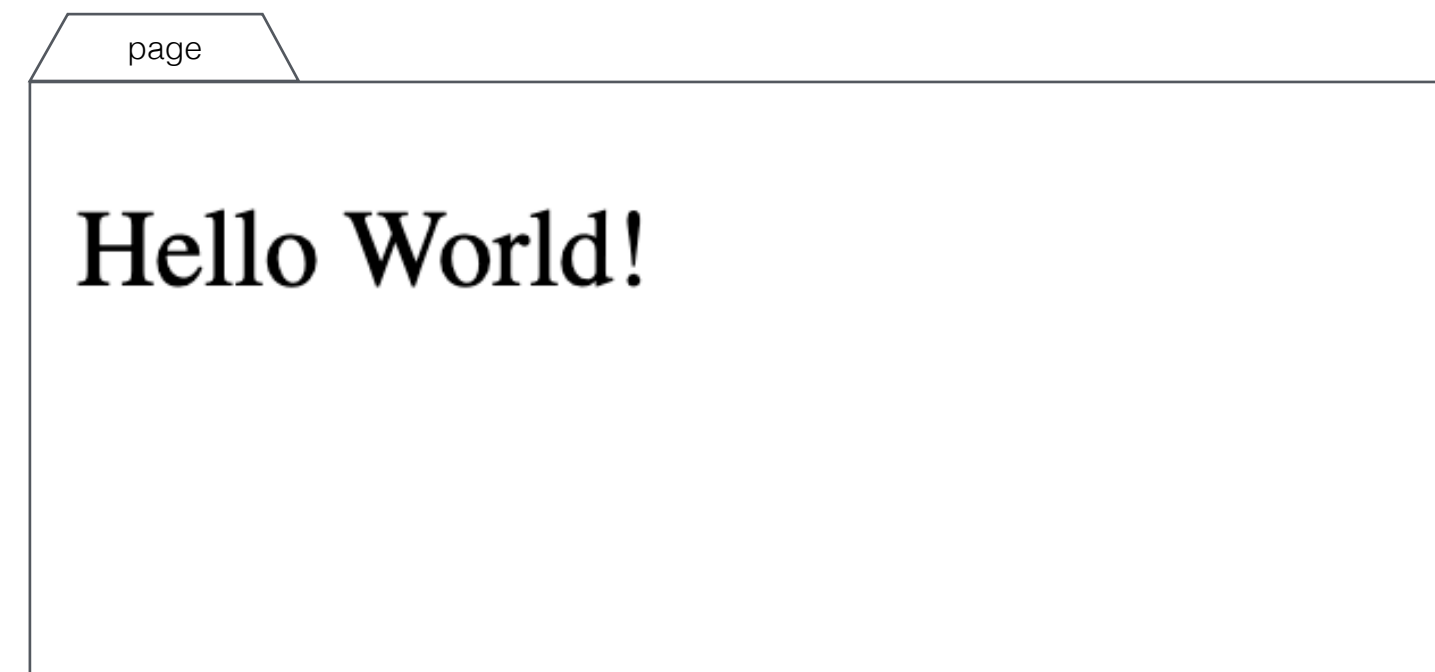
The Fellowship of the String

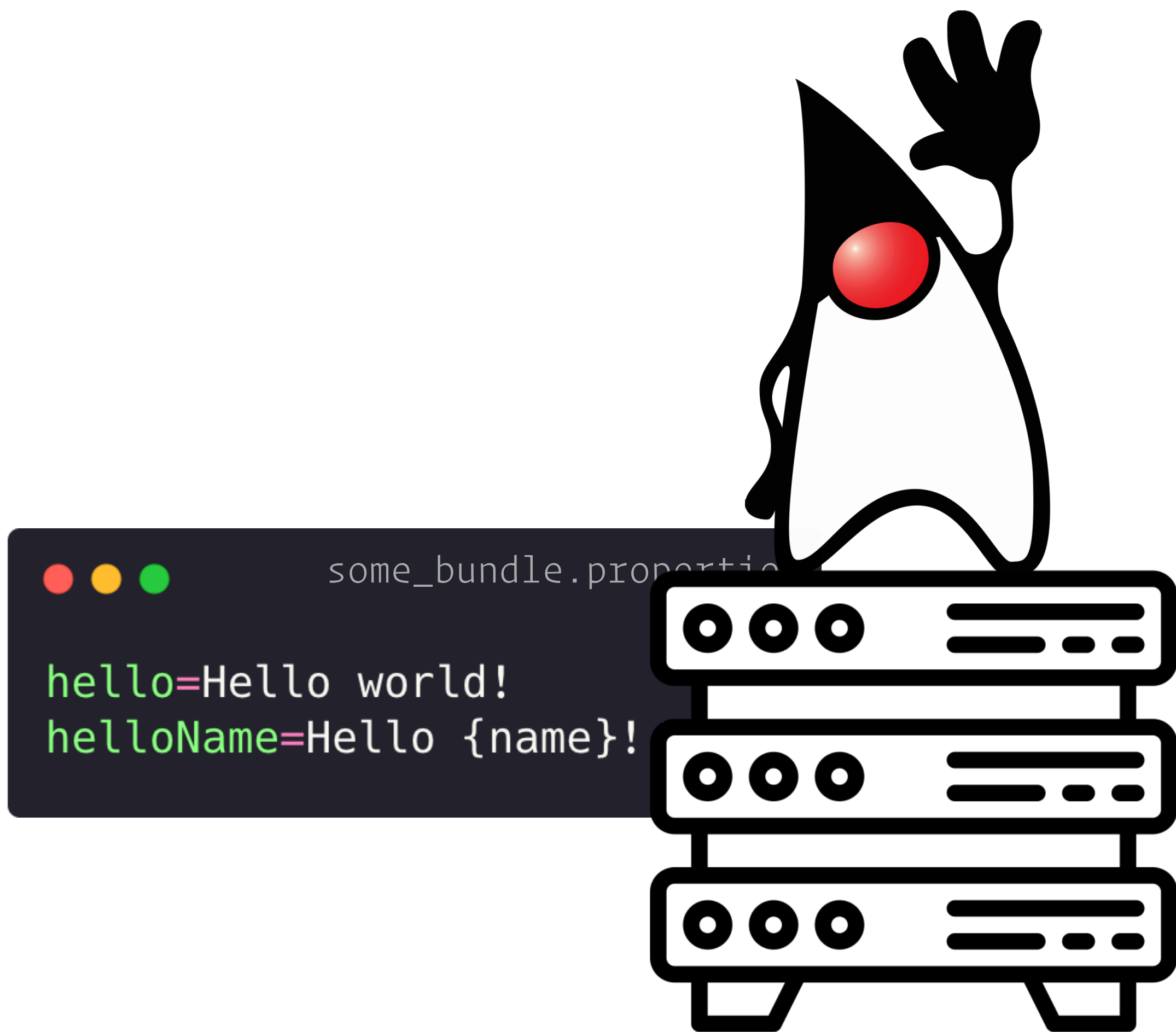
Successful refactoring with empathy and small bites

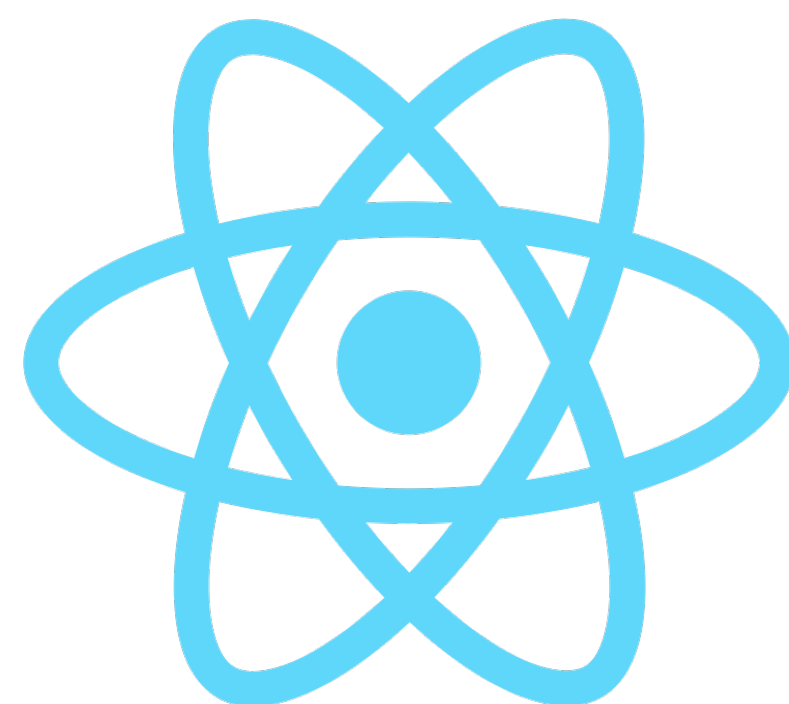
```
getI18nString( 'bundle' , 'key' );  
    "Hello Word!"
```

```
const Hello = () => {  
  // Get getI18nString from a global context  
  const { getI18nString } = useContext(MyContext);  
  // Pull a translated version of the hello string out of some_bundle.js  
  return (  
    <p>{ getI18nString('some_bundle', 'hello') }</p>  
  );  
};
```

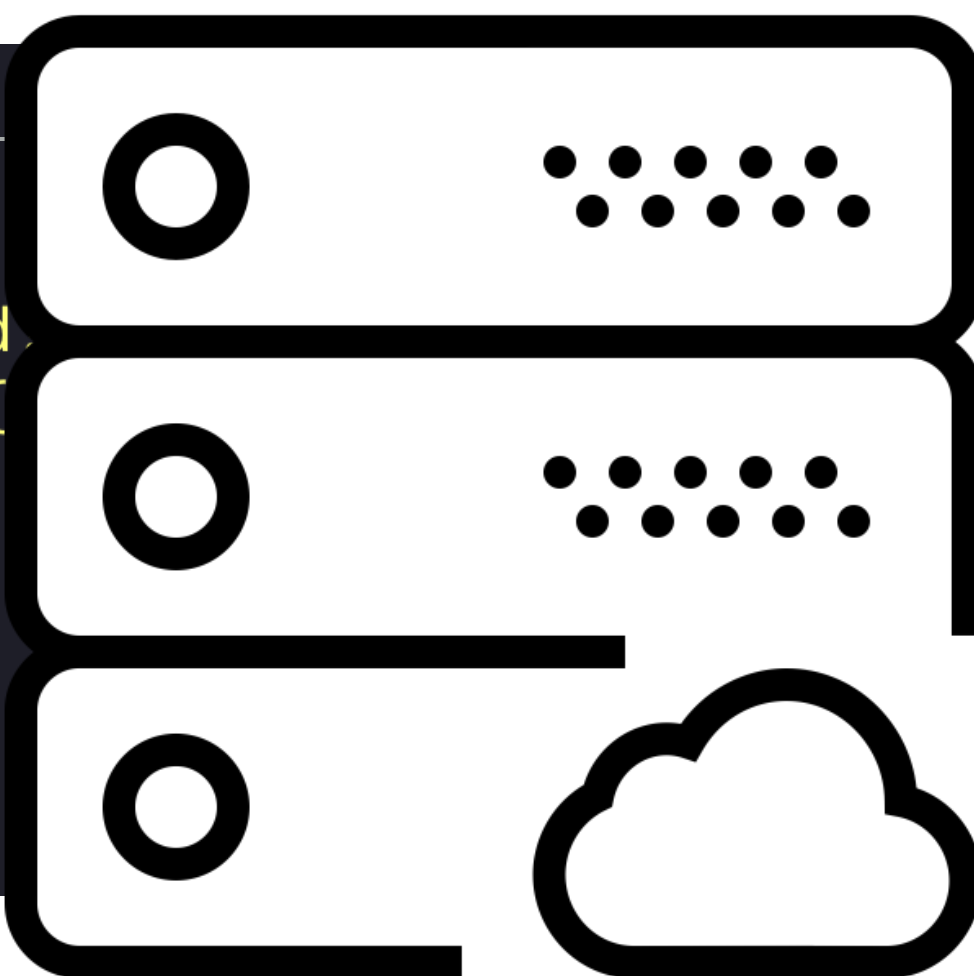
```
some_bundle.json  
  
{  
  "hello": "Hello World!",  
  "helloName": "Hello {name}!",  
}
```







```
some_
{
  "hello": "Hello World",
  "helloName": "Hello {",
  "never": "gonna",
  "give": "you",
  "up": "never",
  "gonna": "let"
}
```



?

```
const Hello = () => {
  // Get getI18nString from a global context
  const { getI18nString } = useContext(MyContext);
  // Pull a translated version of the hello string out of some_bundle.js
  return (
    <p>{ getI18nString('some_bundle', 'hello') }</p>
  );
};

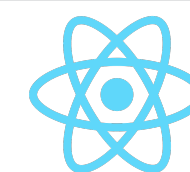
// Declare which strings we want to use in this component
Hello.__meta__ = {
  strings: [
    {
      bundle: 'some_bundle',
      ids: ['hello'],
    },
  ],
};
```

```
some_bundle.json

{
  "hello": "Hello World!",
  "helloName": "Hello {name}!",
}
```

page

Hello World!



```
getI18nString( 'bundle', 'not_a_key' )  
  
== ""
```


getI18nString('not_a_key')

JS





```
// Test a key
if (getI18nString('my_bundle', 'maybe_a_key')) {
    // ...
}

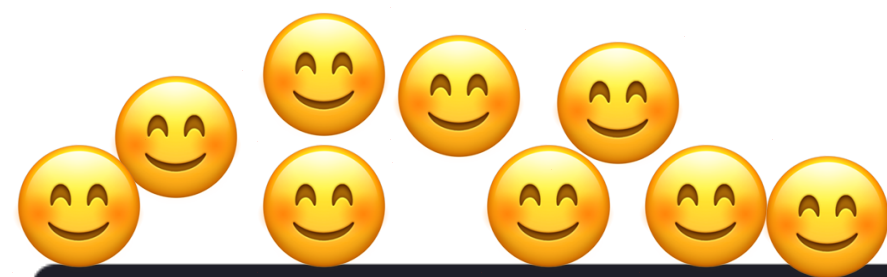
// Check if a UI string exists
for (const optionName of paymentOptions) {
    const key = 'payment.' + optionName;
    if (getI18nString('payment_bundle', key)) {
        return optionName;
    }
}

// try a server value and fall back to a default
return (
    getI18nString('payment_bundle', serverValue) ||
    getI18nString('payment_bundle', 'default')
);
```

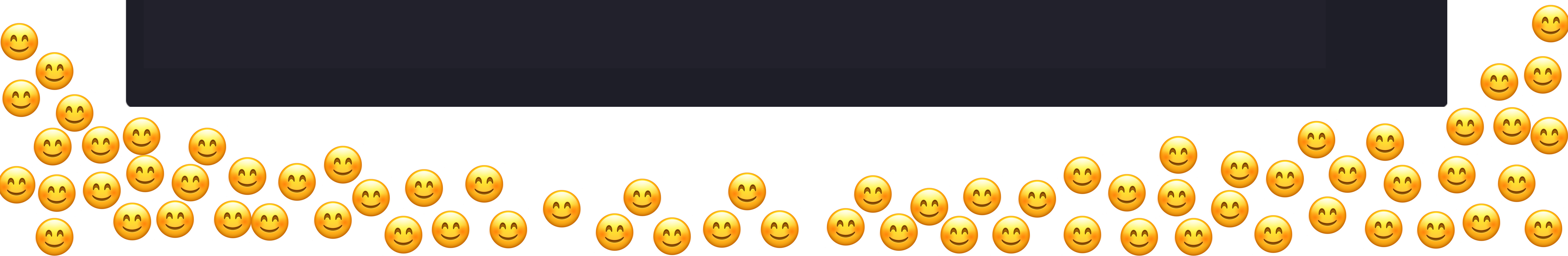
BUT

IT

IS



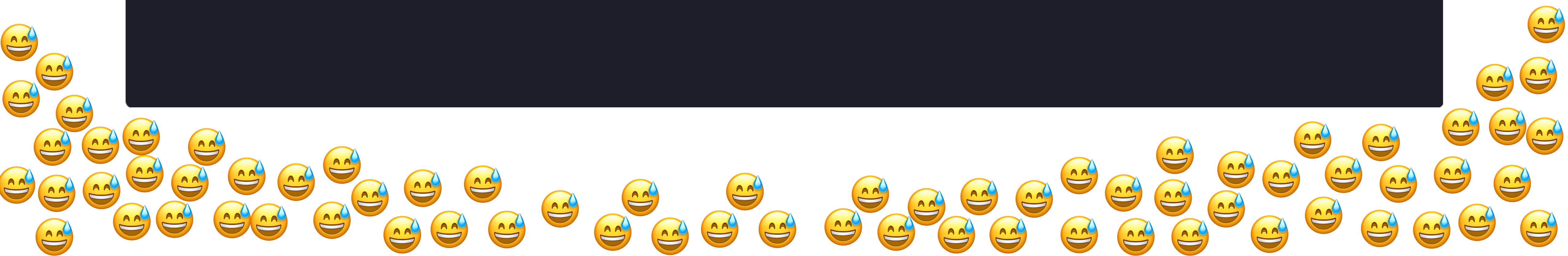
```
// Using the falsy blank string fallback
const myString =
  getI18nString('bundle', someVariable) || // if this is an empty string
  getI18nString('bundle', 'fallback');    // fall back to this
```





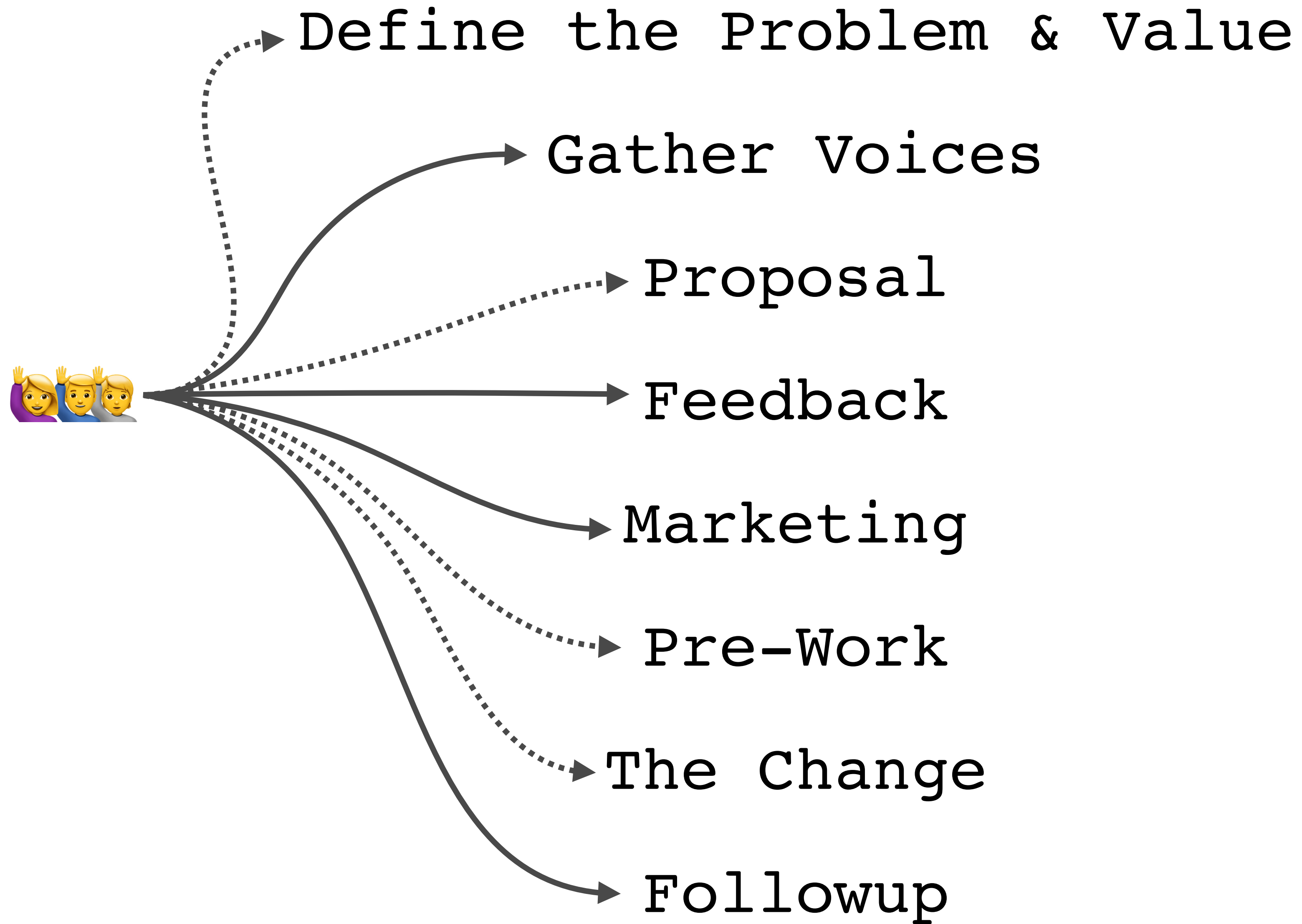
```
// Using the falsy blank string fallback
const myString =
  getI18nString('bundle', someVariable) || // if this is an empty string
  getI18nString('bundle', 'fallback');    // fall back to this
```

```
// Oops! 🐛🐛🐛
const bug = getI18nString('myBundle', 'oops this key is wrong!'); // Blank!
```





Proposal
The Change





Define the Problem & Value

Gather Voices

Proposal

Feedback

Marketing

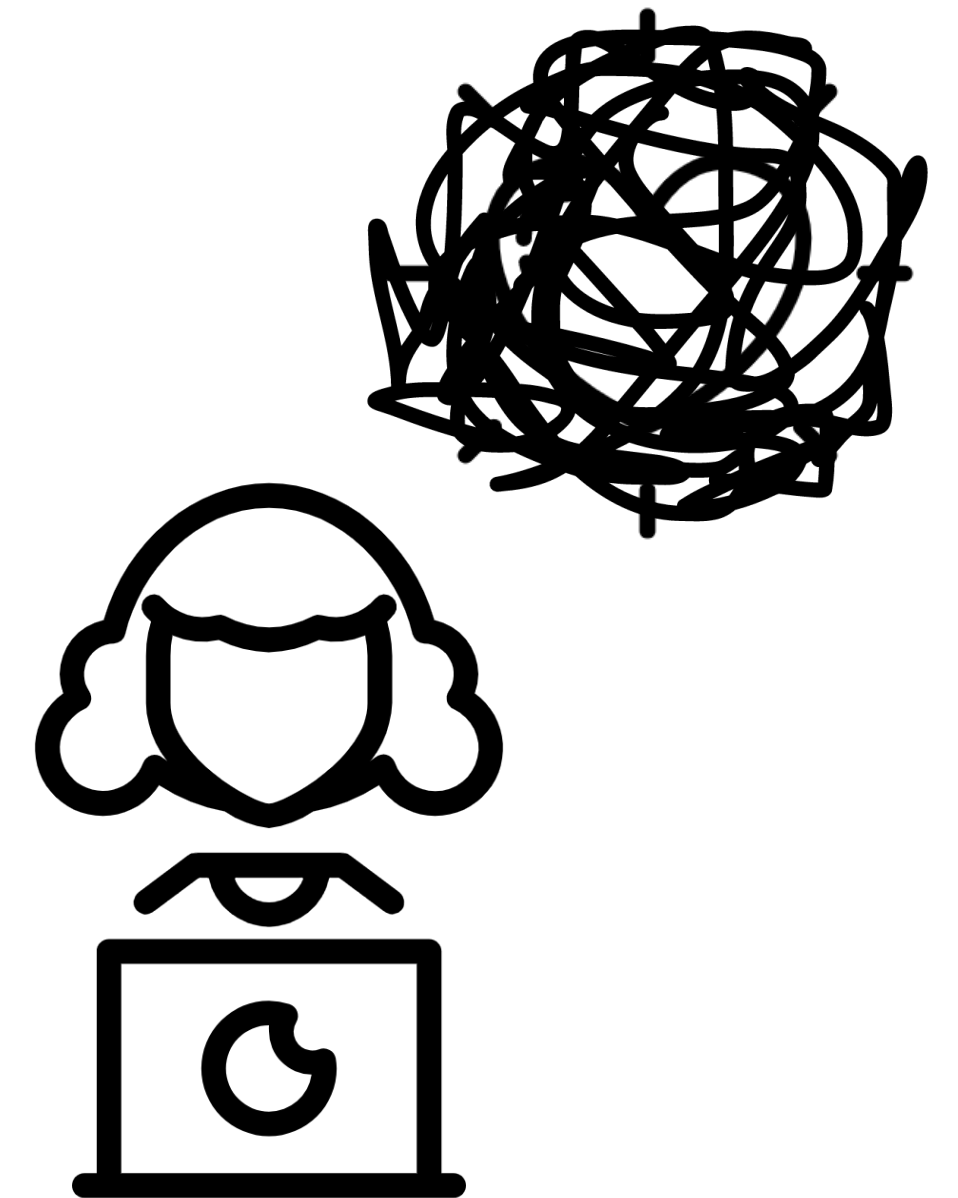
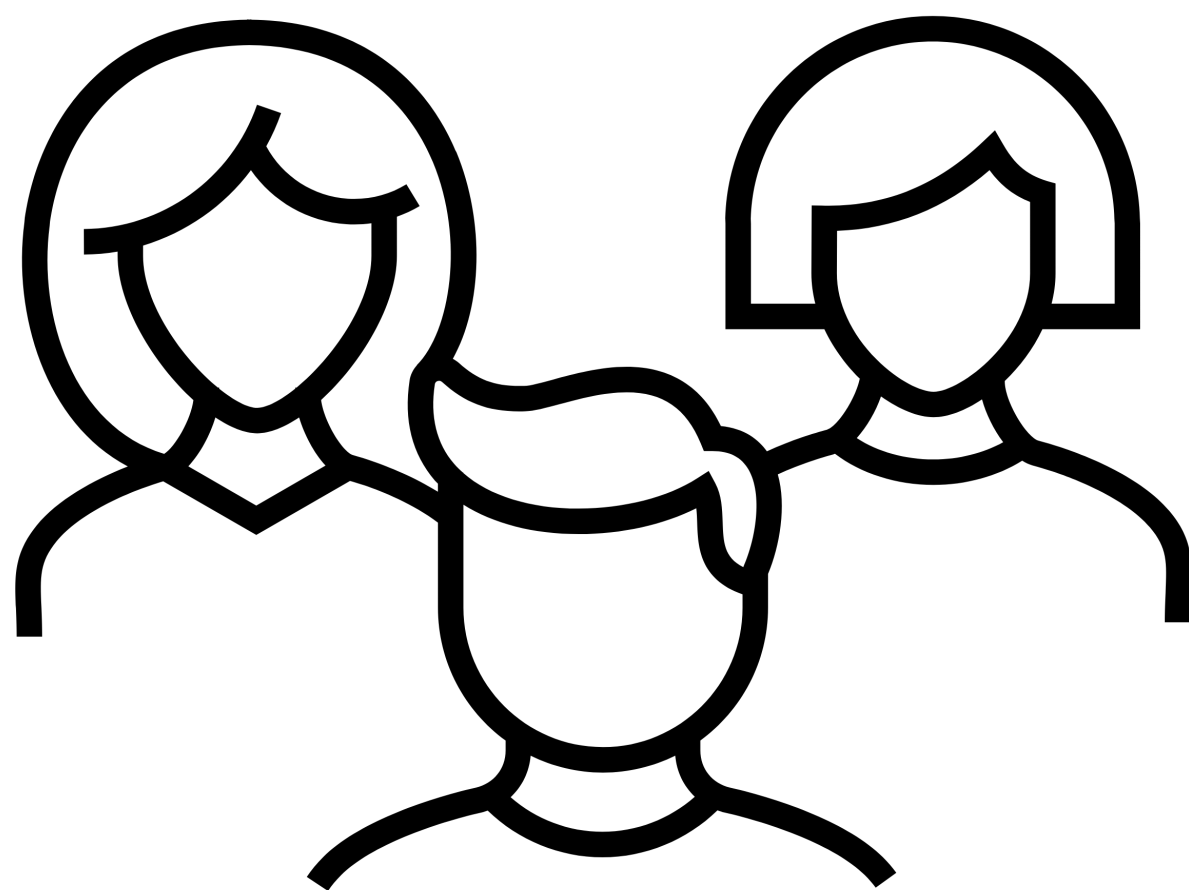
Pre-Work

The Change

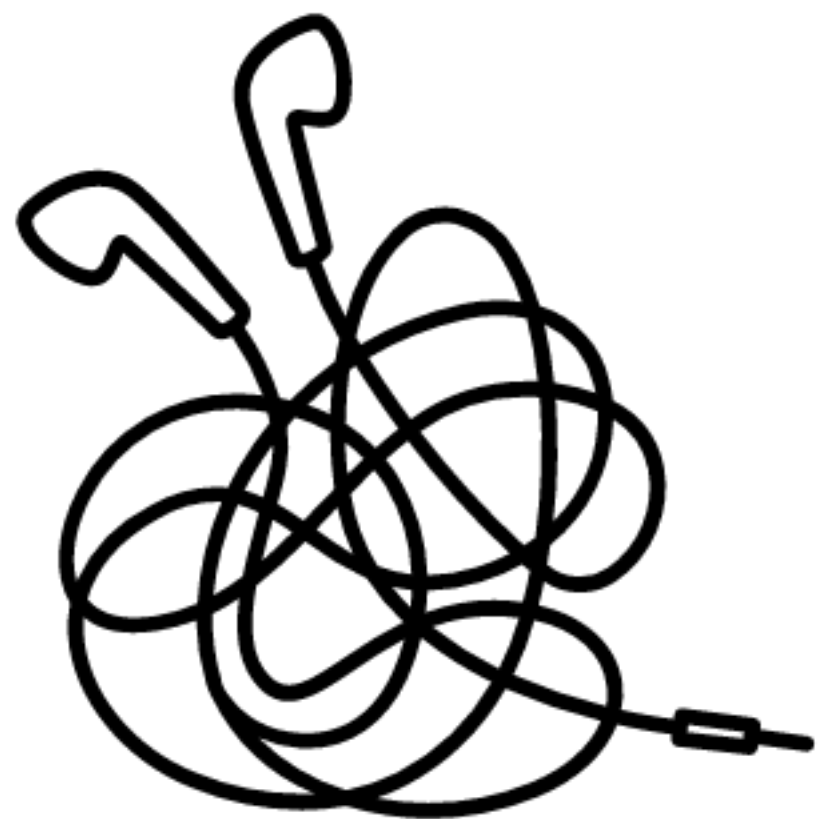
Followup

Define the Problem and
the Value

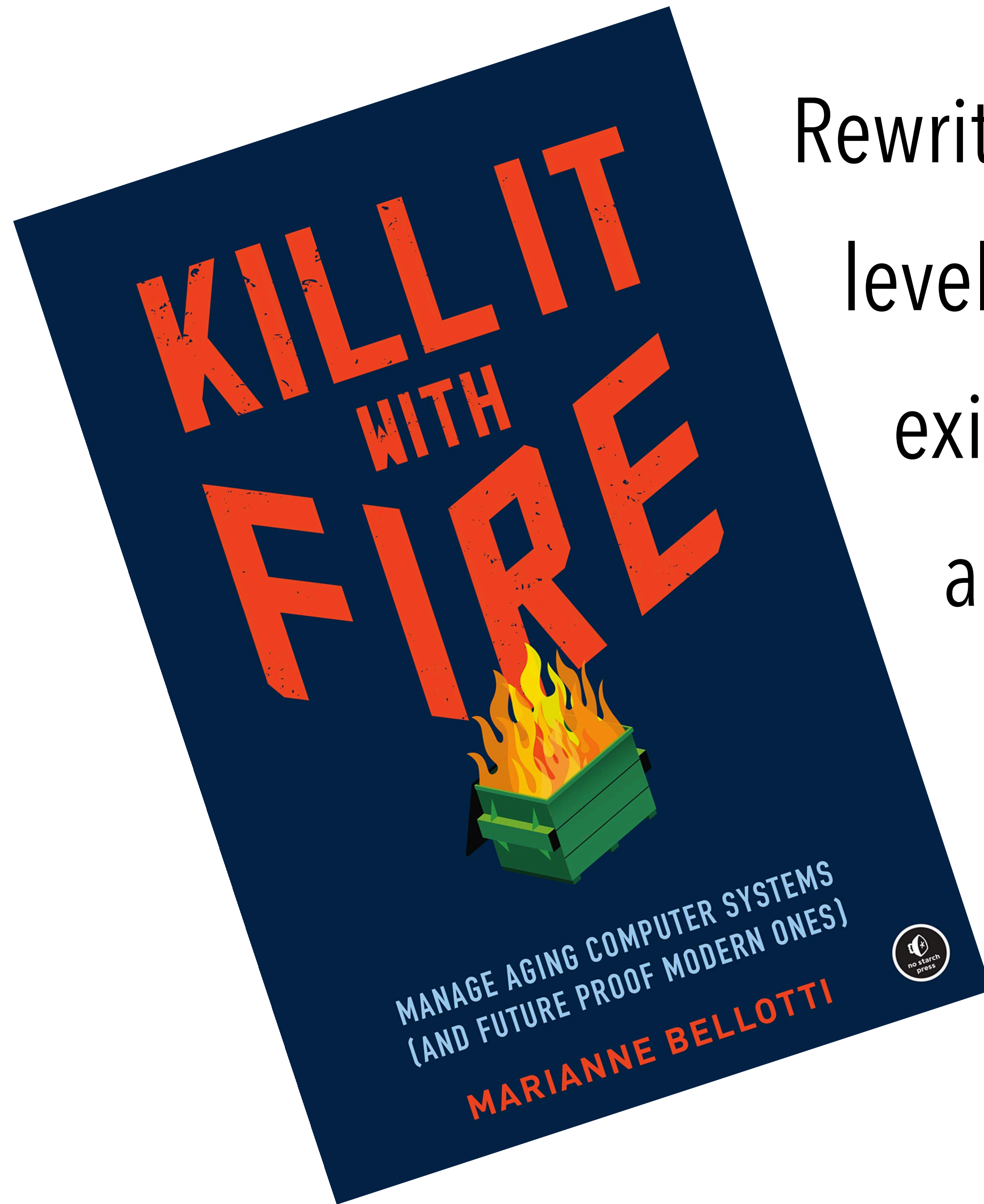
Gather Your Voices



Technical Solution







Rewrites maintain an attractive level of ambiguity while the existing systems are well known and therefore boring.

Marianne Bellotti



```
// Declare we want to use a string
const helloStringObject = requireString('bundleName', 'key');

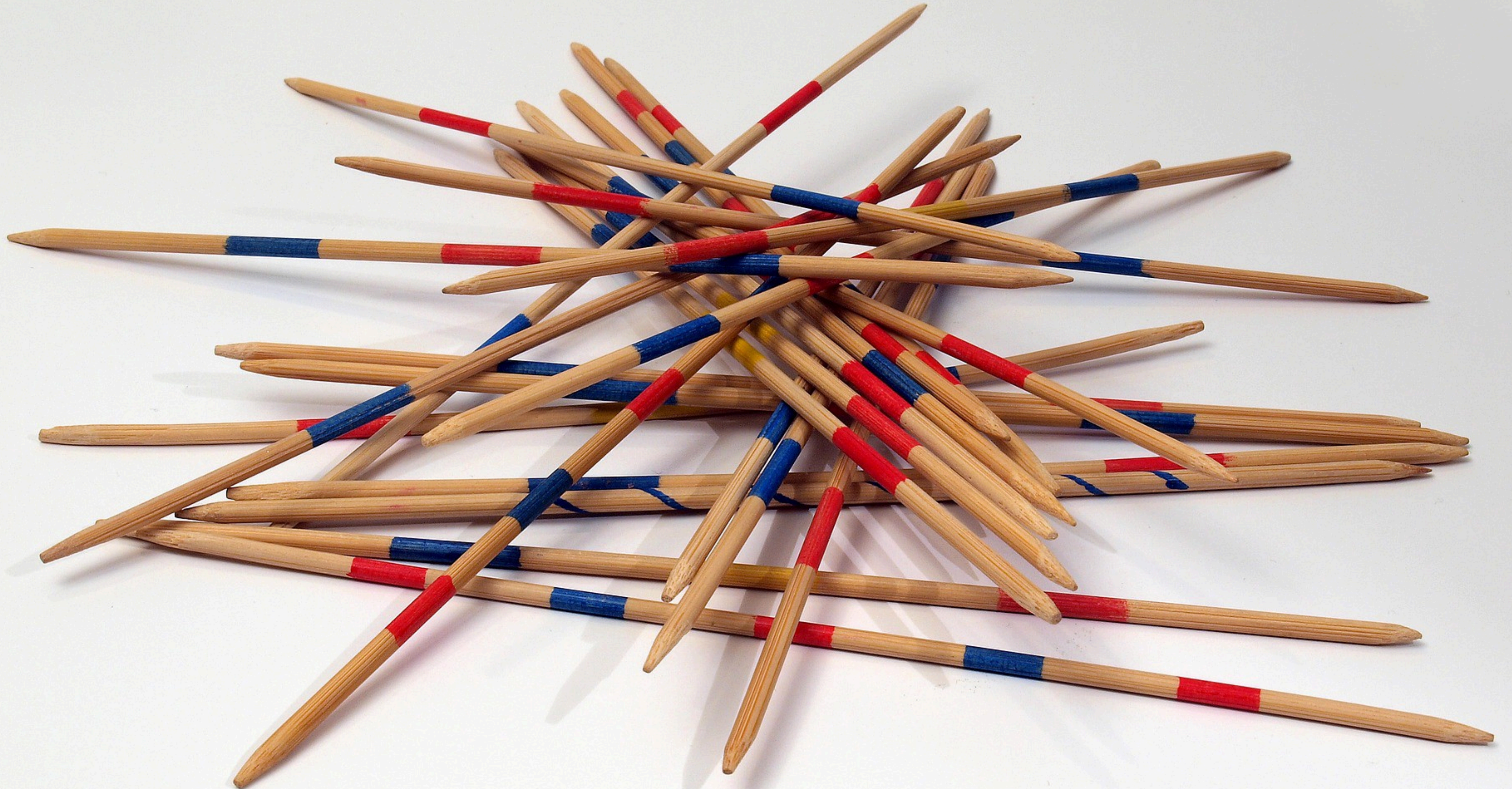
// Use the string at runtime
const translatedHello = formatString(helloStringObject);
```


Technical (P) Review

Marketing

Technical Pre-Work

The “Mikado Method”



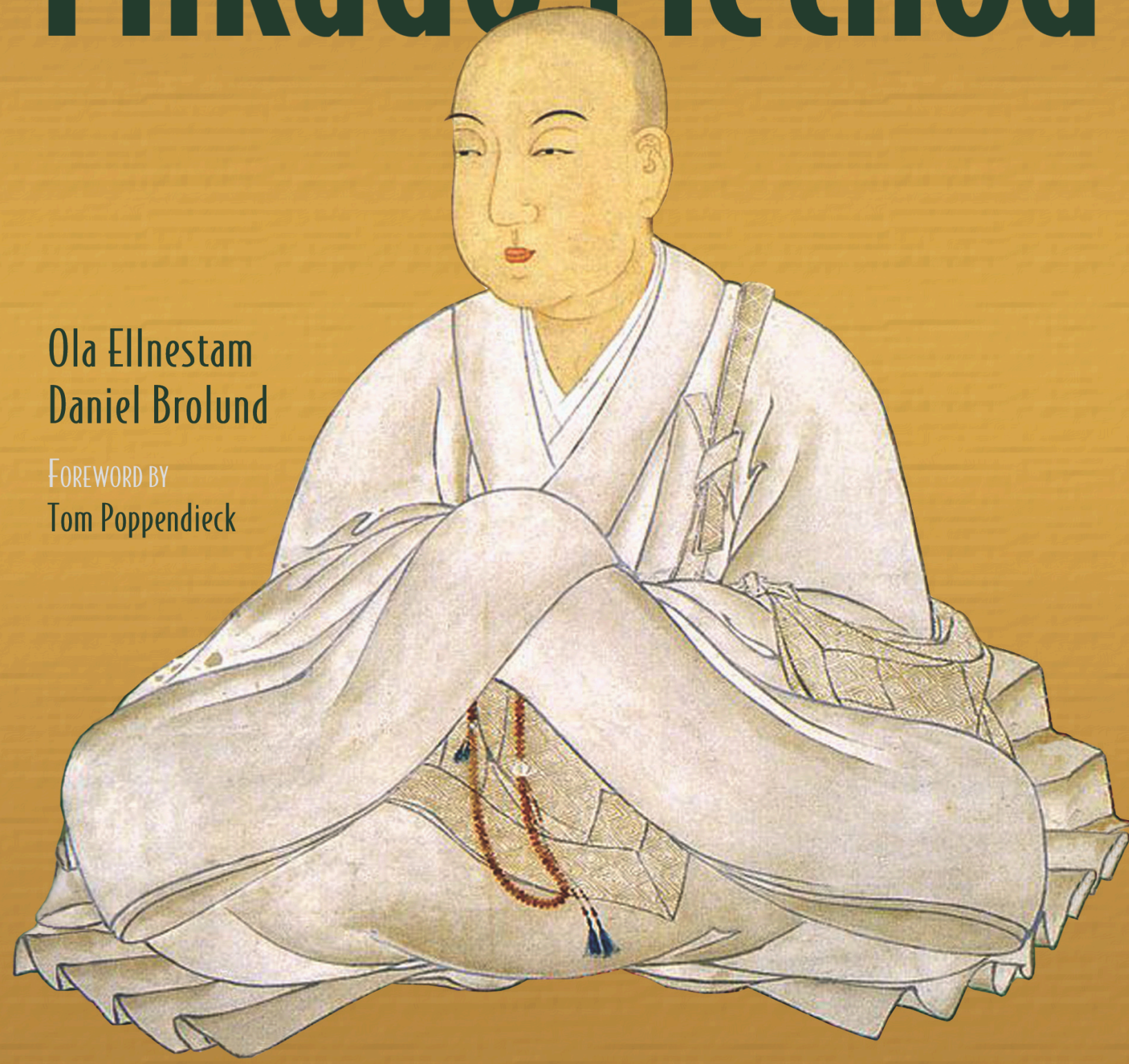


 MANNING

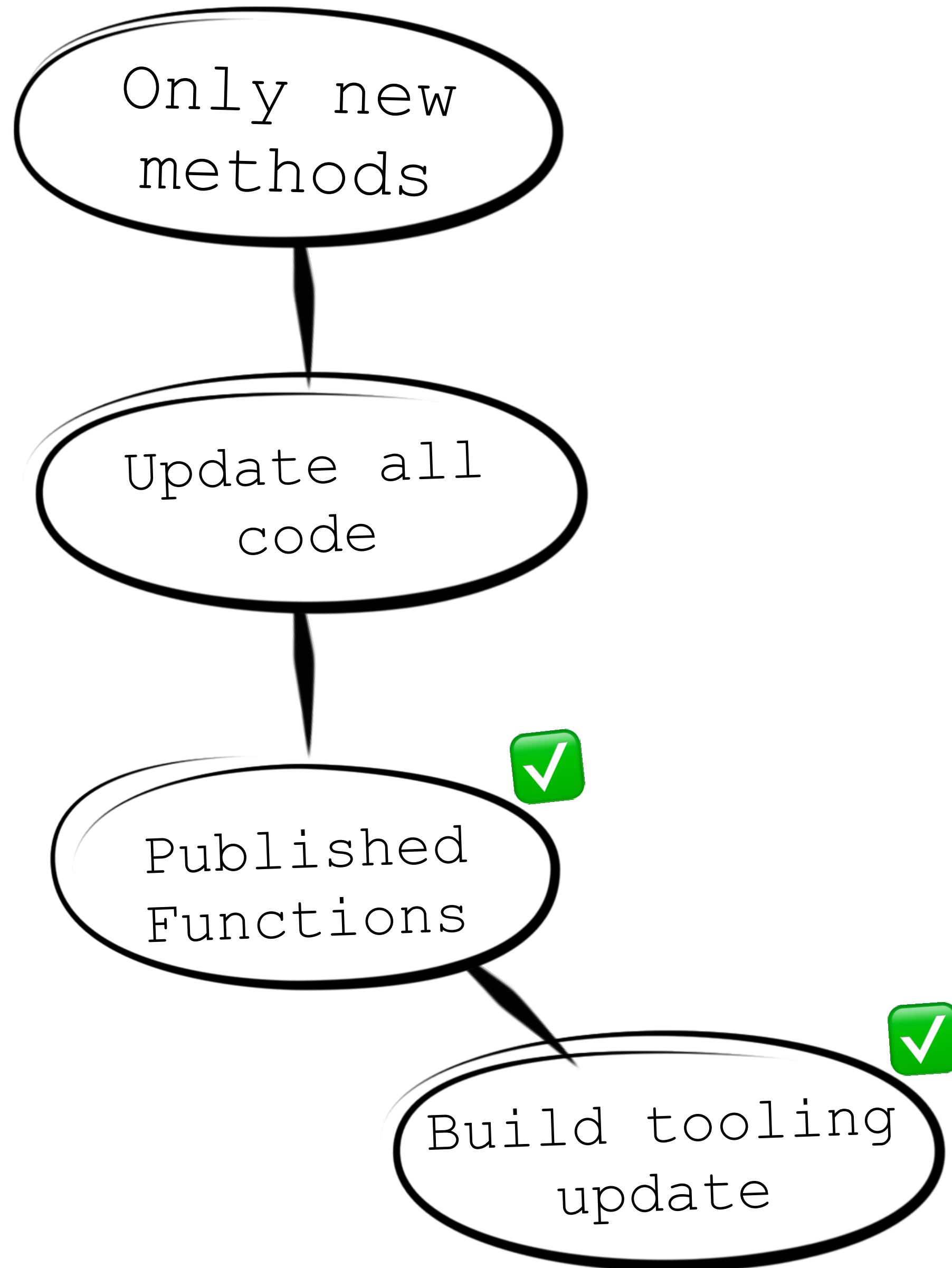
The Mikado Method

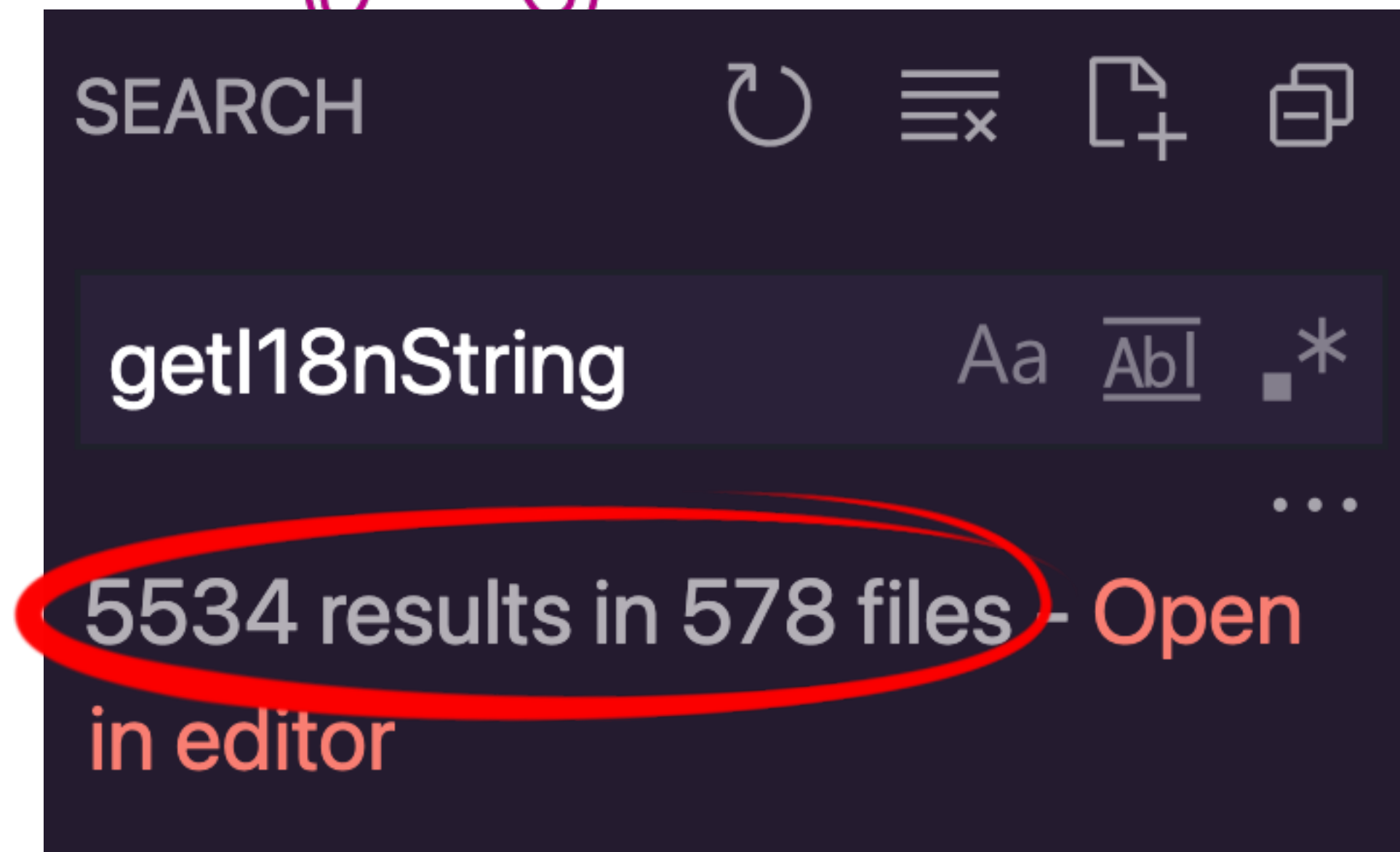
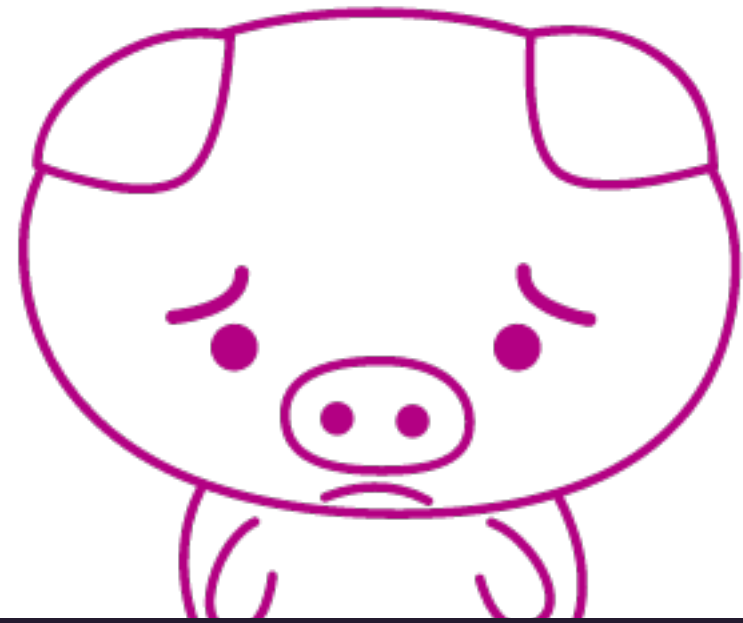
Ola Ellnestam
Daniel Brolund

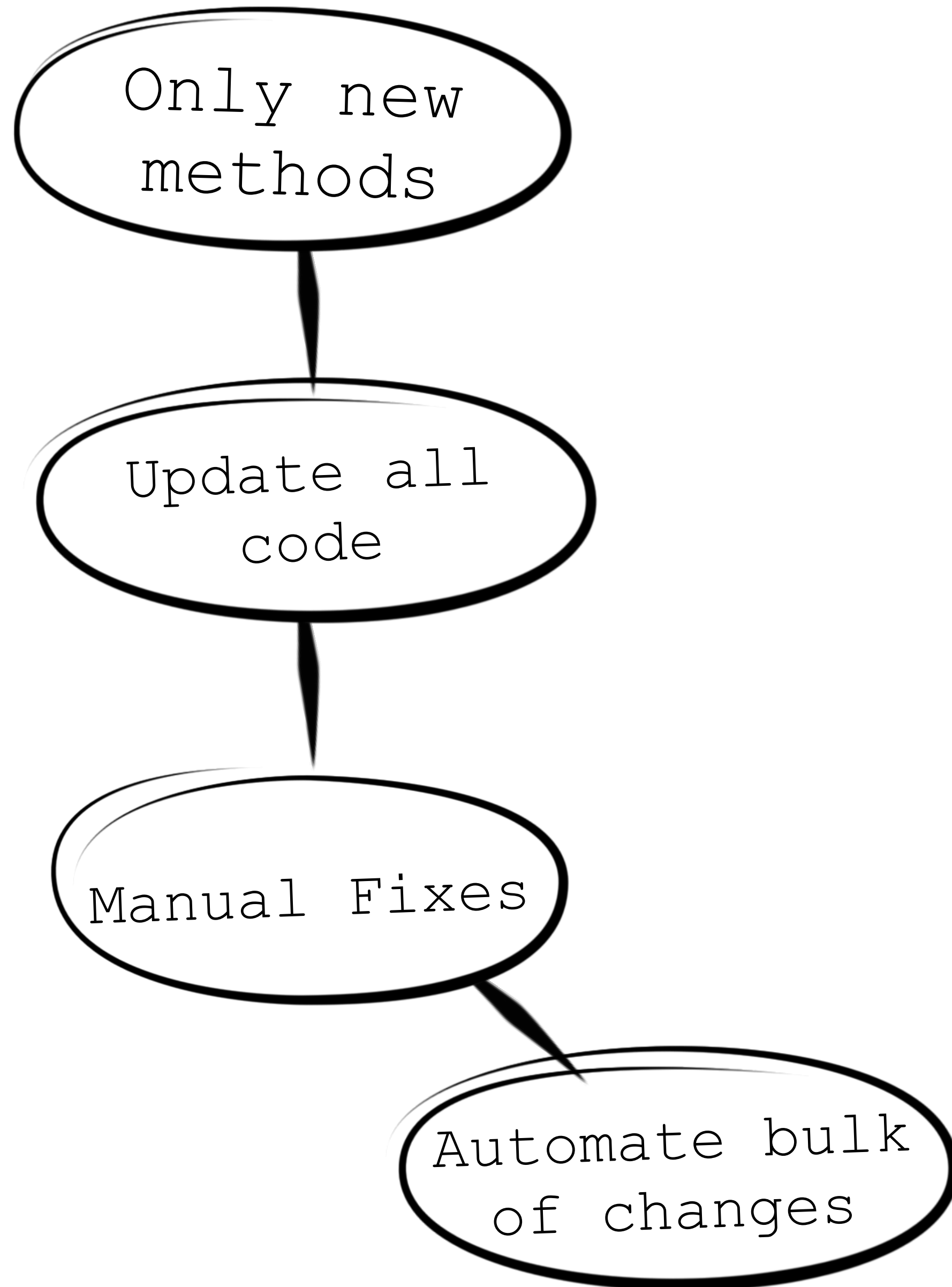
FOREWORD BY
Tom Poppendieck



bit.ly/mikado-method







Codemods

 [facebook](#) / [jscodeshift](#) Public

A JavaScript codemod toolkit.

BABEL



Codemods

AST Explorer



Snippet



JavaScript

@babel/parser



Transform



default



Parser: [@babel/parser-7.15.3](#)

```
1
2
3 const myConst = foo.bar;
4
5
```

```
- program: Program {
  sourceType: "module"
- body: [
  - VariableDeclaration {
    declarations: [
      - VariableDeclarator {
        id: Identifier {
          name: "myConst"
        }
        init: MemberExpression {
          object: Identifier {
            name: "foo"
          }
          computed: false
          - property: Identifier = $node {
            name: "bar"
          }
        }
      }
    ]
    kind: "const"
  }
]
```


AST Explorer

Snippet



JavaScript

@babel/parser



Transform



default



Parser: [@babel/parser-7.15.3](#)

Transformer: [babelv7-7.14.0](#)

```
1
2 // input
3 const myConst = foo.bar;
4
5
```

```
- body: [
  - VariableDeclaration {
    - declarations: [
      - VariableDeclarator {
        - id: Identifier {
          name: "myConst"
        }
        - init: MemberExpression {
          + object: Identifier {name}
          computed: false
          + property: Identifier {name}
        }
      }
    ]
  }
```

```
1 // Codemod or Transform
2 export default function (babel) {
3   return {
4     visitor: {
5       // Reverse the name of any identifier we find
6       Identifier(path) {
7         path.node.name = path.node.name.split('').reverse().join('');
8       }
9     }
10  };
11 }
12
```

Prettier

```
1 // output
2 const tsnoCym = oof.rab;
```





Codemods with Babel Plugins



AUTHOR
Laurie Barth



bit.ly/babel-codemod



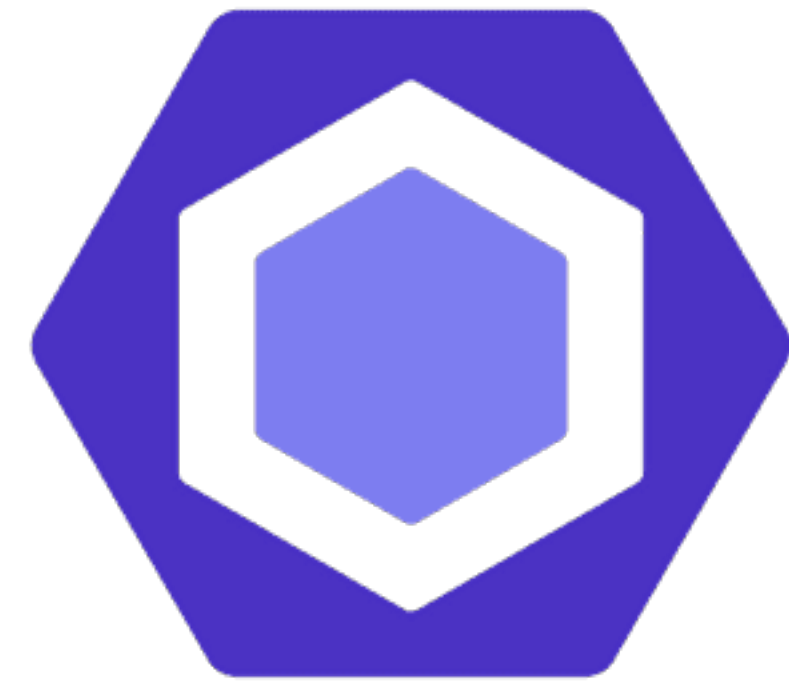
Compilers for Humans

New Course by John Otander & Laurie Barth

compilersforhumans.com



Let ' s Go !



ESLint

Unexpected 'debugger' statement. eslint(no-debugger)

[View Problem](#) [Quick Fix... \(⌘.\)](#)

debugger;



```
{
  "rules": {
    "no-restricted-syntax": [
      "warning",
      {
        "selector": "CallExpression[callee.name='getI18nString']",
        "message": "getI18nString has been deprecated and will be removed in
the next weeks. See bit.ly/string-update for documentation and examples on its
replacement and reach out to @jacques if you have any questions."
      }
    ]
  }
}
```

getI18nString has been deprecated and will be removed in the next weeks. See <https://bit.ly/string-update> for documentation and examples on its replacement and reach out to @jacques if you have any questions. `eslint(no-restricted-syntax)`

```
const getI18nString: (bundle: any, key: any) => string
```

View Problem Quick Fix... (⌘.)

```
getI18nString('bundle', 'key');
```


Manual Changes



ESLint



Localization API Upgrade

Created vs. Resolved Chart: String API Upgrades - Member Subtasks

Filter Results: Top-level string API upgrades

Key	Summary ↑	Sub-Tasks	Assignee
WEBUI-15305	Simplicity stringRegistry reducer	WEBUI-15306, WEBUI-15307, WEBUI-15308, WEBUI-15310, WEBUI-15311, WEBUI-15313, WEBUI-15314, WEBUI-15315, WEBUI-15316, WEBUI-15317, WEBUI-15318, WEBUI-15319, WEBUI-15320	Chris Hourihan
WEBUI-14293	String API upgrades in ui/js/account	WEBUI-14306, WEBUI-14307, WEBUI-14308, WEBUI-14309, WEBUI-14310, WEBUI-14311, WEBUI-14312, WEBUI-14313, WEBUI-14314, WEBUI-14315, WEBUI-14316, WEBUI-14317, WEBUI-14318, WEBUI-14319, WEBUI-14320, WEBUI-14321, WEBUI-14322, WEBUI-14323, WEBUI-14324, WEBUI-14325, WEBUI-14326, WEBUI-14327, WEBUI-14328, WEBUI-14329, WEBUI-14330, WEBUI-14331, WEBUI-14332	Stefan Heymanns
WEBUI-14294	String API upgrades in	WEBUI-14333, WEBUI-	Jacques

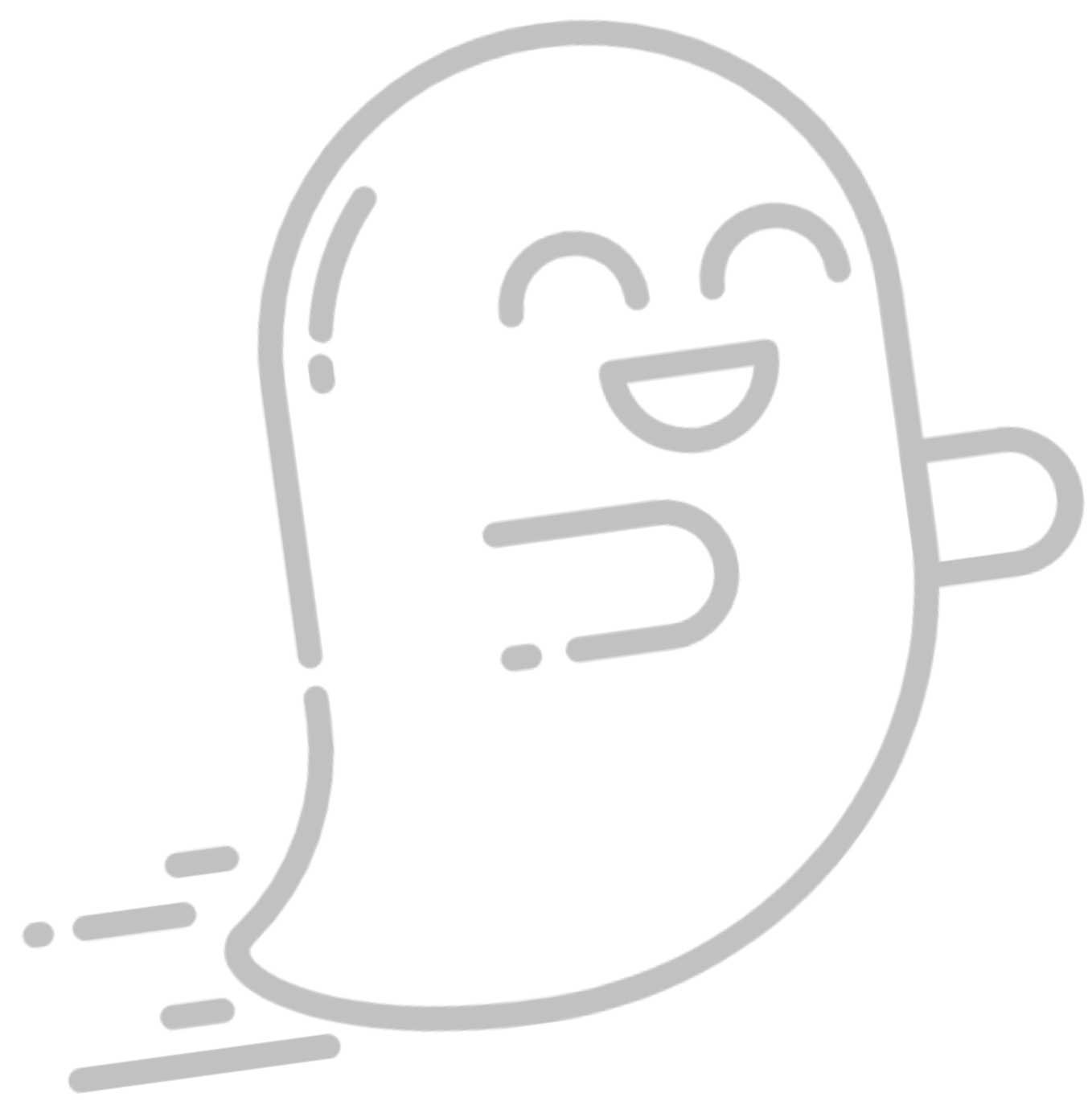
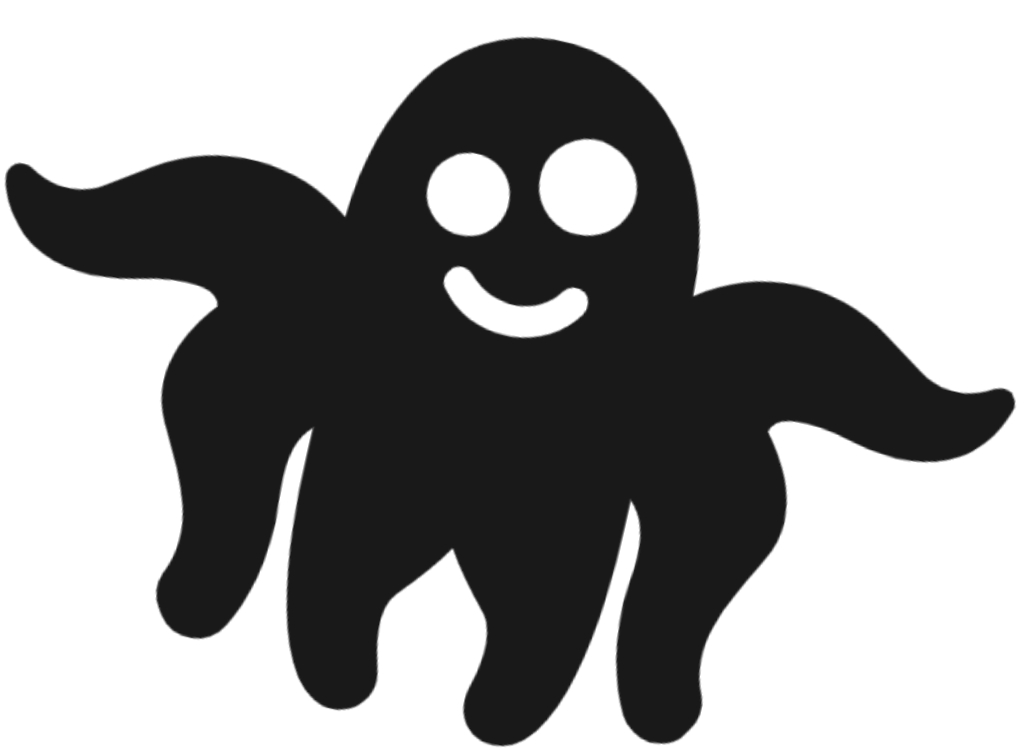
Created vs. Resolved Chart: String API Upgrades - AUI Subtasks

Issue Statistics: String API Upgrades - All Subtasks (Status)

Status	Count	Percentage
OPEN	1	0%
RESOLVED	186	67%
VERIFIED	11	4%
CLOSED	81	29%
Total	279	

Filter Results: String API Upgrades - All Subtasks

Key	Status ↑	Summary	Assignee
WEBUI-15304	OPEN	WEBUI-15068 / /common/i18n usage in ui/js/signup/simplicity/reducers/formFields.js	Unassigne
WEBUI-14306	RESOLVED	WEBUI-14293 / Deprecated localization APIs in ui/js/account/ageVerification.jsx	Jacques Favreau
WEBUI-14307	RESOLVED	WEBUI-14293 / Deprecated localization APIs in ui/js/account/billingDetails.jsx	Stefan Heymanns
WEBUI-14308	RESOLVED	WEBUI-14293 / Deprecated localization APIs in ui/js/account/cancelPlan.jsx	Stefan Heymanns
WEBUI-14309	RESOLVED	WEBUI-14293 / Deprecated localization APIs in ui/js/account/changeEmail.jsx	Jacques Favreau
WEBUI-14310	RESOLVED	WEBUI-14293 / Deprecated localization APIs in ui/js/account/changePassword.jsx	Jacques Favreau





```
// Are we on a developer box?  
if(isDevBox){  
    throw new Error('Deprecated! Please ping @jacques immediately if you see this.');
```



```
// Otherwise we're in prod  
logger.alert(`getI18nString called with ${bundle} and ${key}!`);
```