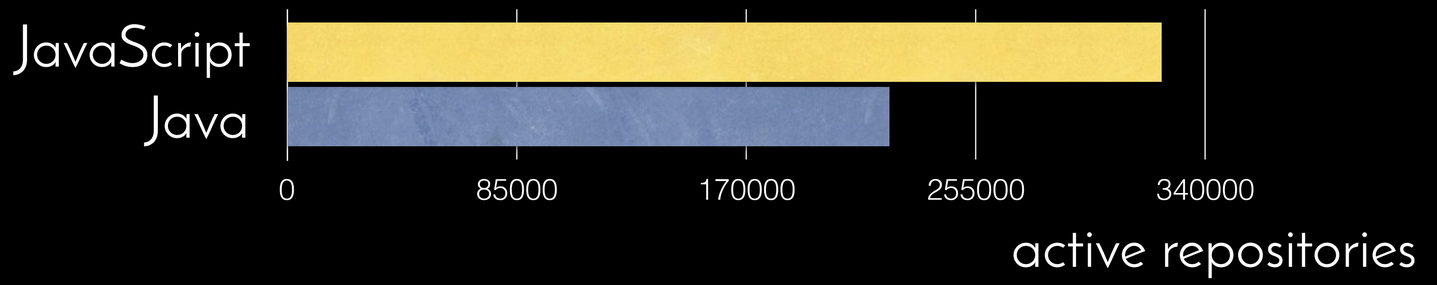
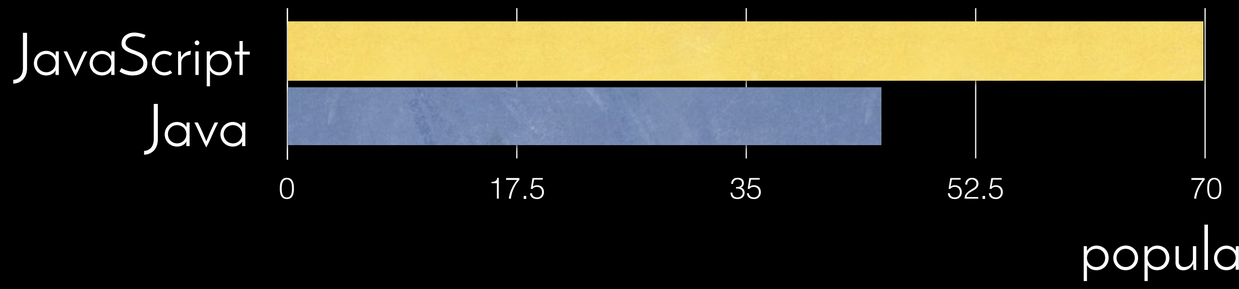
Finding Broken Promises In Asynchronous JavaScript Programs

Saba Alimadadi, Di Zhong, Magnus Madsen, and Frank Tip

Northeastern University

AARHUS UNIVERSITET


JavaScript
Java
0
85000
170000
255000
340000
active repositories


JavaScript
Java
0
17.5
35
52.5
70
popularity %

JS

Asynchronous JavaScript

```
fs.readdir(source, function (err, files) {
    files.forEach(function (fileName, fileIndex) {
        gm(source + fileName).size(function (err, values) {
            widths.forEach(function (width, widthIndex) {
                this.resize(w, h).write(nameName, function (err) {
                })
            })
        })
    })
})
```

little pyramid of doom!

// example from callbackhell.com (simplified)

Promise

JavaScript Promises


```
let p = new Promise(function (resolve, reject) {
});

let p2 = p.then(handleSuccess)  ;
           .then(soMuchSuccess, handleRejection)  ;
           .catch(moreRejection);
```

```
tcpPortUsed.check(port, 'localhost')
    .then(function (inUse) {
        if (inUse) {
            printErrorAndExit();
        } else {
            printSuccessMessage();
            startServer(port);
        }
})
```

simplified and modified example from:
https://github.com/facebook/Docusaurus

```
tcpPortUsed.check(port, 'localhost')
    .then(function (inUse) {
        if (inUse) {
            printErrorAndExit();
        } else {
            printSuccessMessage();
            startServer(port);
        }
})
```

> *Currently any exceptions thrown during server startup cause the process to silently exit. This makes tracking down bugs in config files super frustrating.*

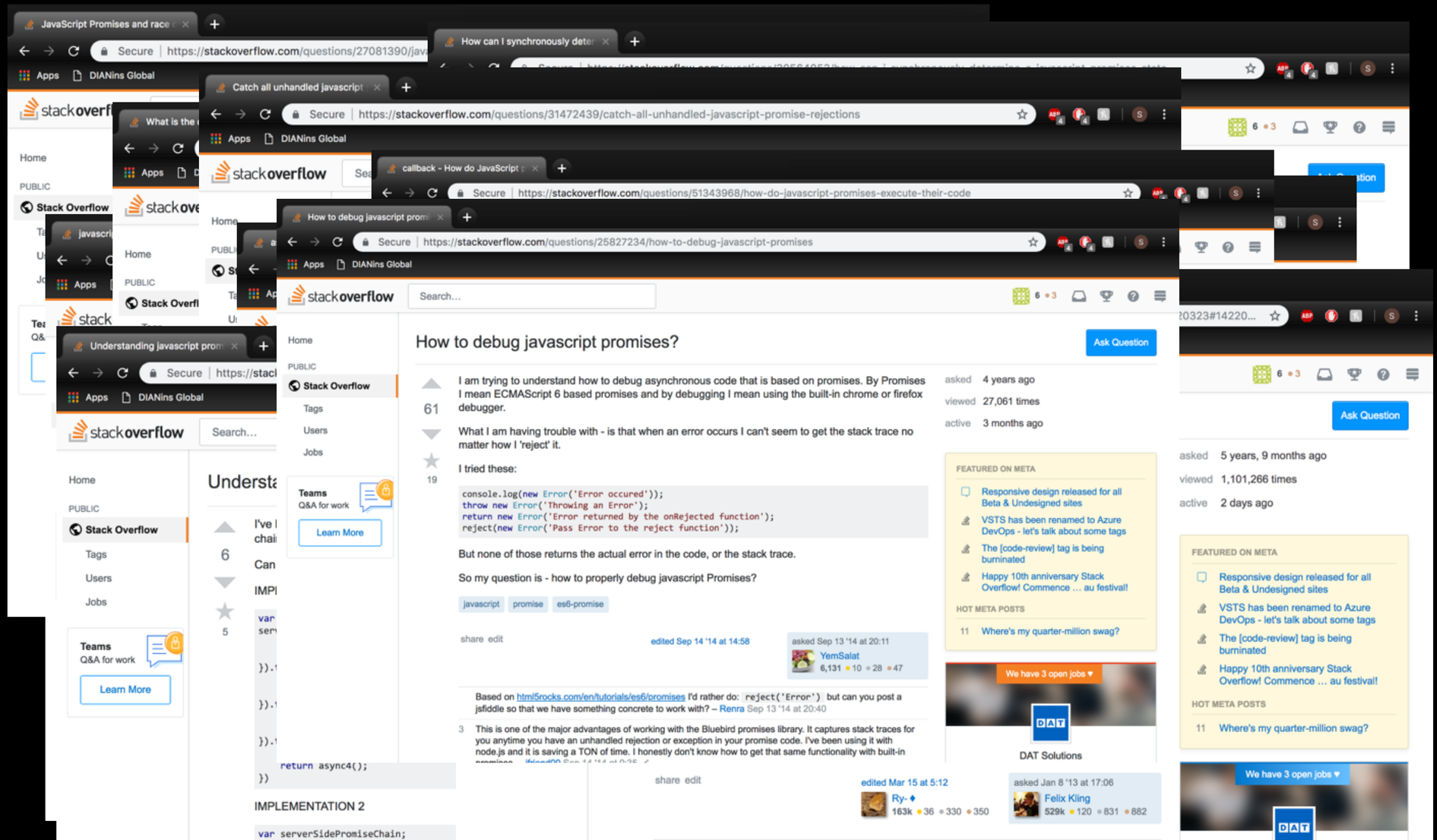
simplified and modified example from:
https://github.com/facebook/Docusaurus

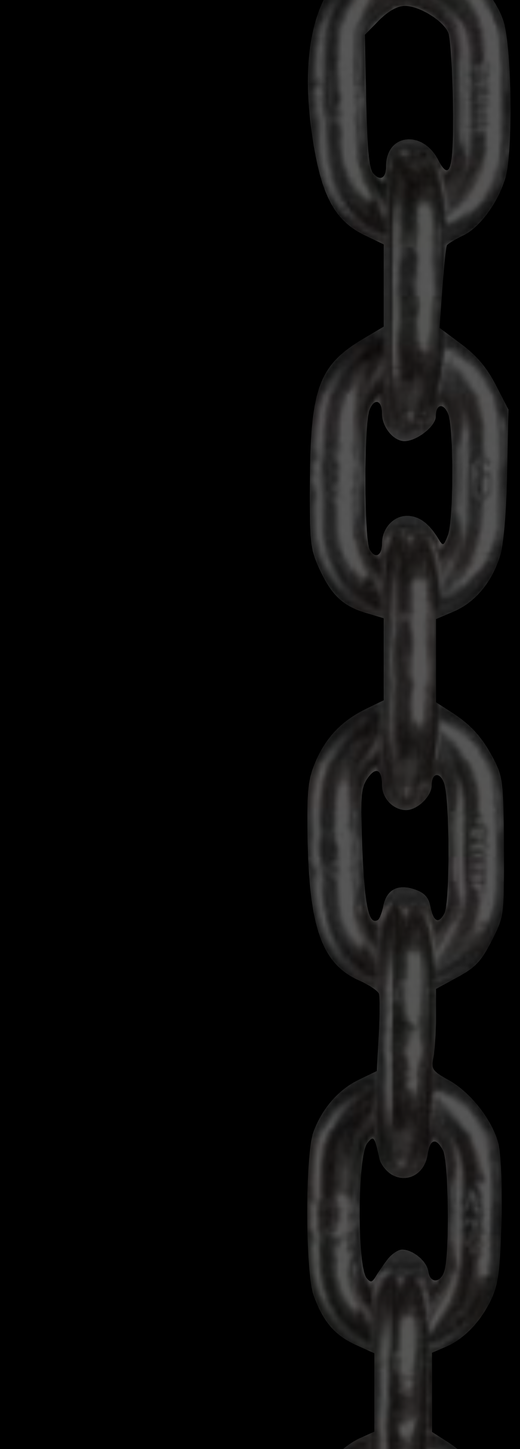
```
tcpPortUsed.check(port, 'localhost')
    .then(function (inUse) {
        if (inUse) {
            printErrorAndExit();
        } else {
            printSuccessMessage();
            startServer(port);
        }
})
+ .catch(function (ex) {
+     handleException(ex);
+ });
```

> *Currently any exceptions thrown during server startup cause the process to silently exit. This makes tracking down bugs in config files super frustrating.*

simplified and modified example from:
https://github.com/facebook/Docusaurus

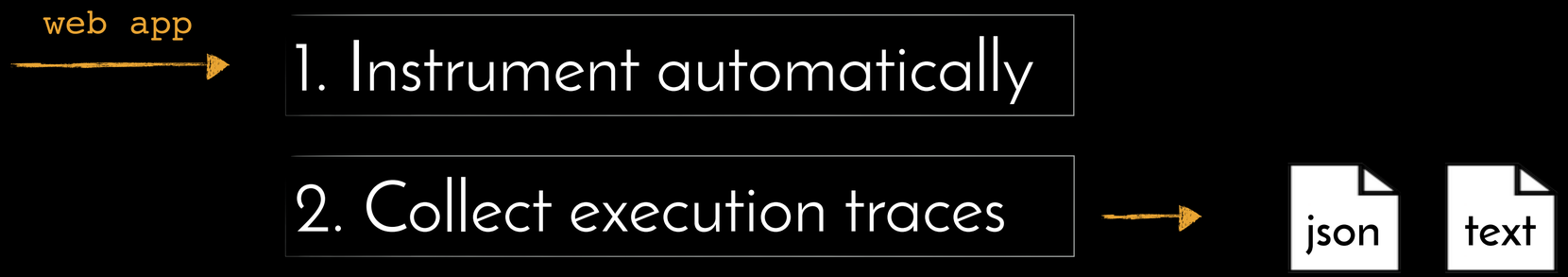
In Practice


How to debug javascript promises?
Ask Question
I am trying to understand how to debug asynchronous code that is based on promises. By Promises I mean ECMAScript 6 based promises and by debugging I mean using the built-in chrome or firefox debugger.
What I am having trouble with - is that when an error occurs I can't seem to get the stack trace no matter how I 'reject' it.
I tried these:
console.log(new Error('Error occured'));
throw new Error('Throwing an Error');
return new Error('Error returned by the onRejected function');
reject(new Error('Pass Error to the reject function'));
But none of those returns the actual error in the code, or the stack trace.
So my question is - how to properly debug javascript Promises?
javascript
promise
es6-promise
asked 4 years ago
viewed 27,061 times
active 3 months ago
FEATURED ON META
Responsive design released for all Beta & Undesigned sites
VSTS has been renamed to Azure DevOps - let's talk about some tags
The [code-review] tag is being burninated
Happy 10th anniversary Stack Overflow! Commence ... au festival!
HOT META POSTS
11 Where's my quarter-million swag?
We have 3 open jobs
DAT Solutions
asked 5 years, 9 months ago
viewed 1,101,266 times
active 2 days ago
IMPLEMENTATION 2
var serverSidePromiseChain;

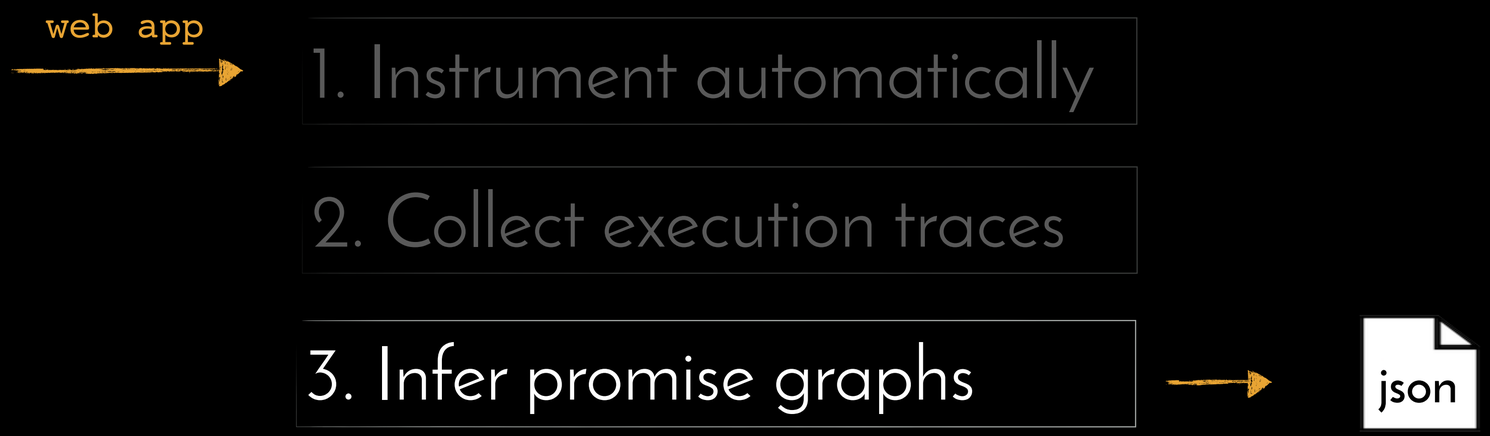


PromiseKeeper

PromiseKeeper



PromiseKeeper



[Madsen et al., 2017]

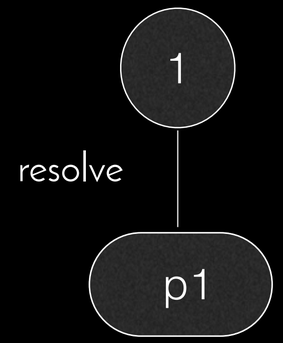
PromiseKeeper

web app

1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs




```
let p1 = Promise.resolve(1);
```

PromiseKeeper

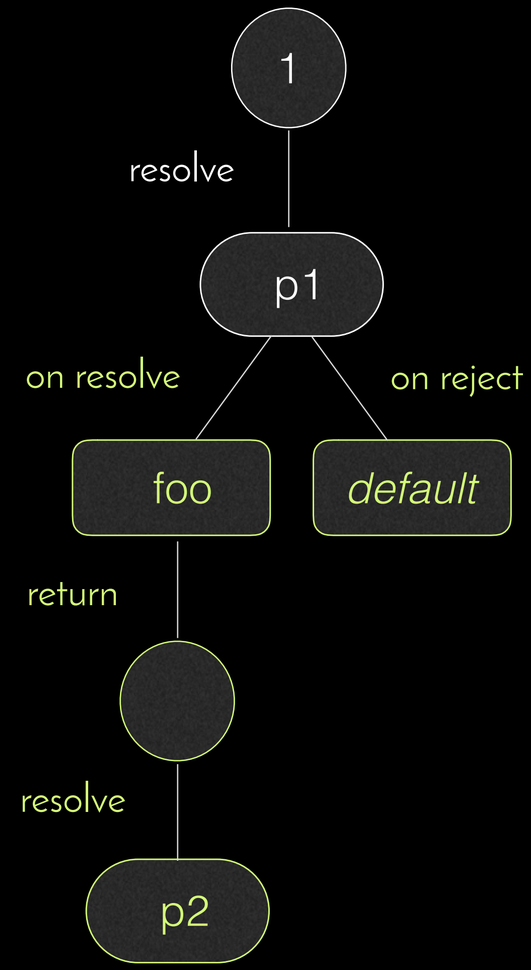
web app

1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs

```
let p1 = Promise.resolve(1);
let p2 = p1.then(foo);
function foo (x) {
}
```



PromiseKeeper

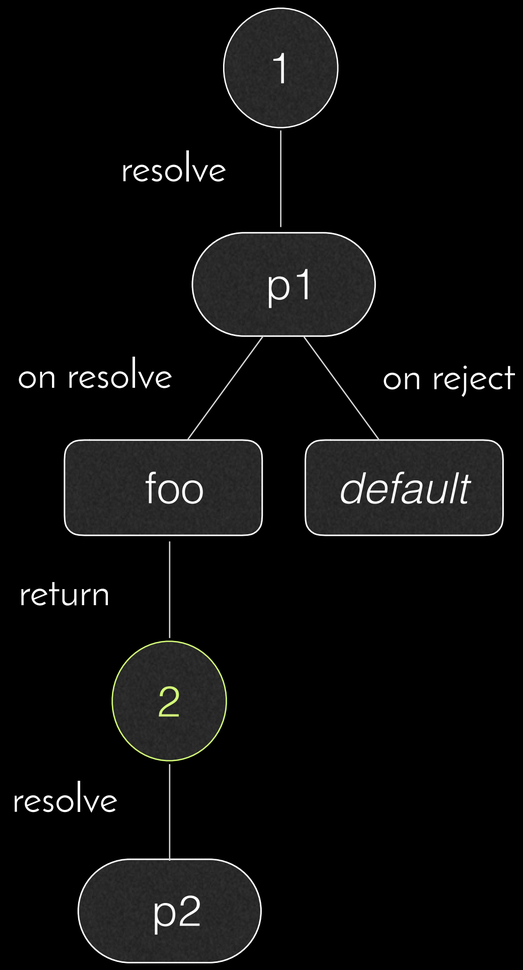
web app

1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs

```
let p1 = Promise.resolve(1);
let p2 = p1.then(foo);
function foo (x) {
  return x + 1;
}
```



PromiseKeeper

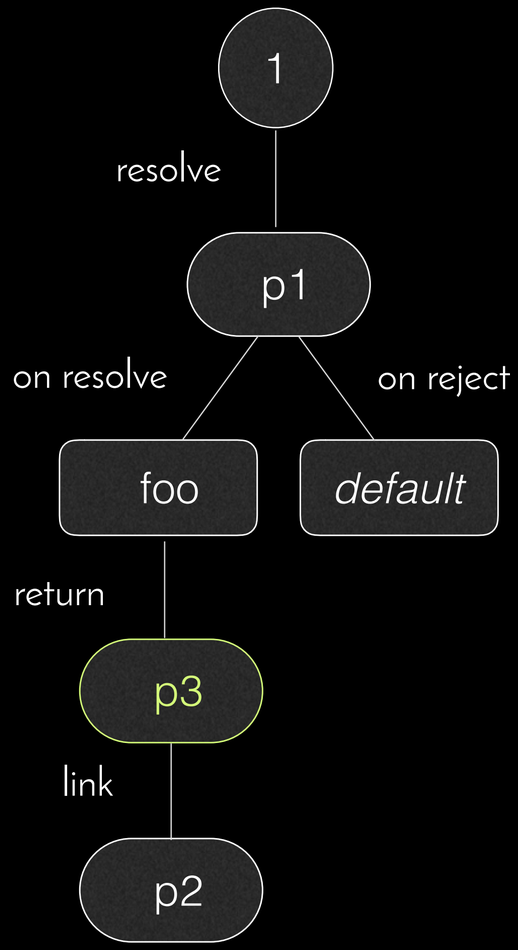
web app

1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs

```
let p1 = Promise.resolve(1);
let p2 = p1.then(foo);
function foo (x) {
   let p3 = Promise.resolve(x);
   return p3;
}
```



PromiseKeeper

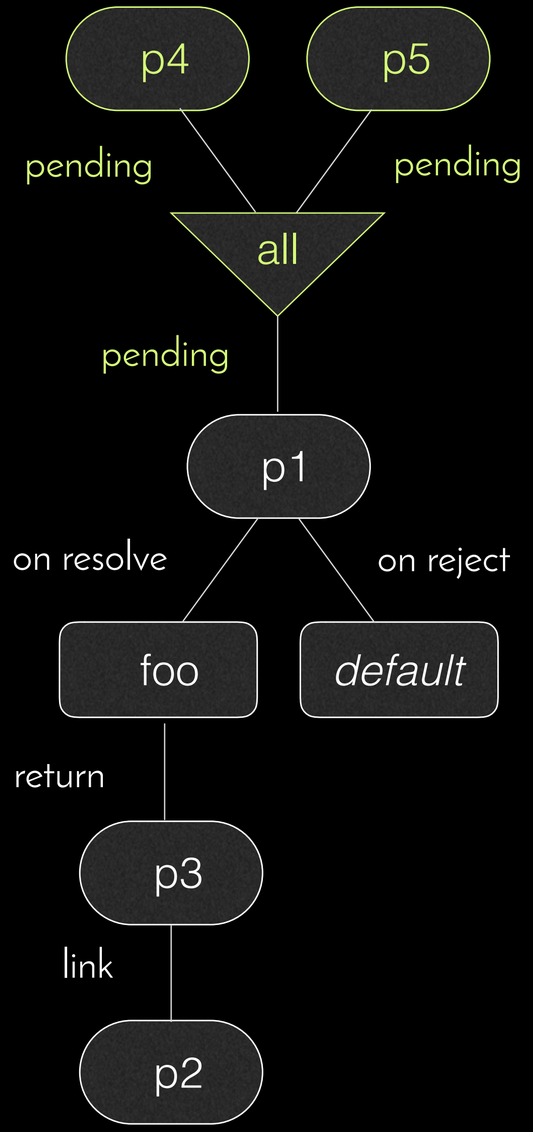
web app

1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs

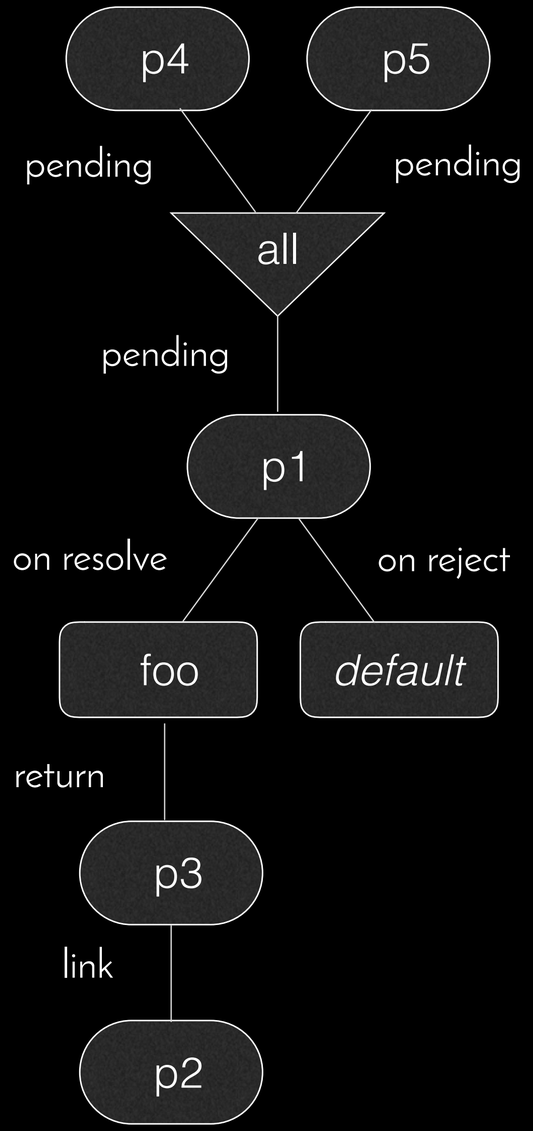
```
let p1 = Promise.all(p4, p5);
let p2 = p1.then(foo);
function foo (x) {
   let p3 = Promise.resolve(x);
   return p3;
}
```



PromiseKeeper

web app

1. Instrument automatically
2. Collect execution traces
3. Infer promise graphs
4. Analyze anti-patterns

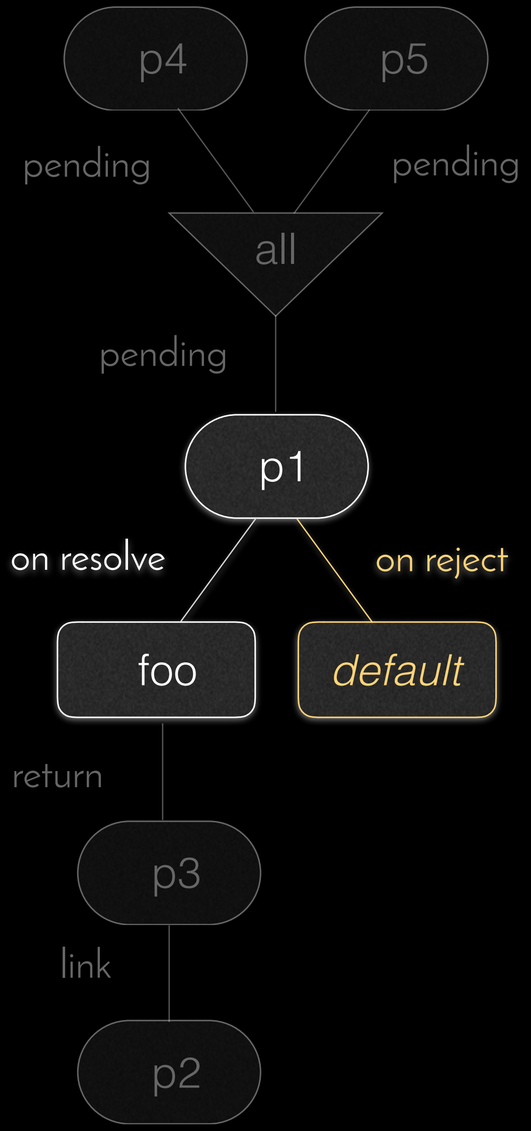


PromiseKeeper

web app

1. Instrument automatically
2. Collect execution traces
3. Infer promise graphs
4. Analyze anti-patterns

unhandled promise rejections

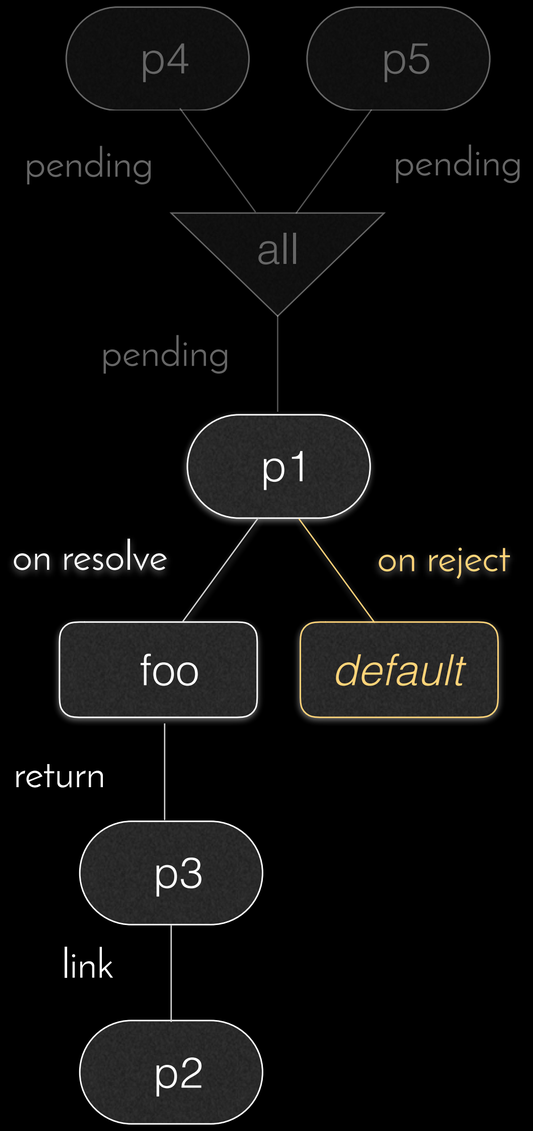


PromiseKeeper

web app

1. Instrument automatically
2. Collect execution traces
3. Infer promise graphs
4. Analyze anti-patterns

unhandled promise rejections

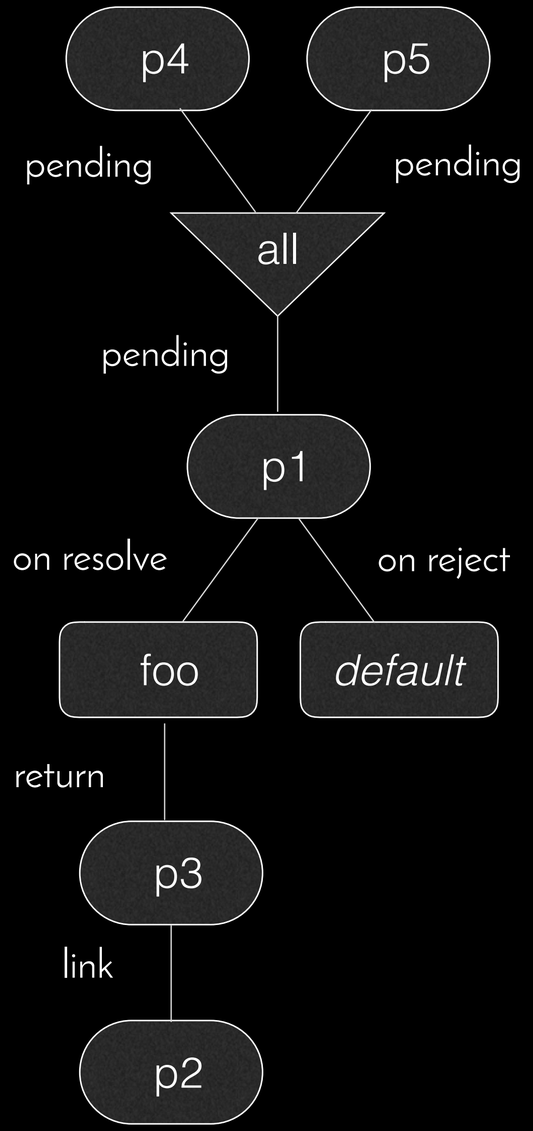


PromiseKeeper

web app

1. Instrument automatically
2. Collect execution traces
3. Infer promise graphs
4. Analyze anti-patterns

unsettled promises



PromiseKeeper

web app

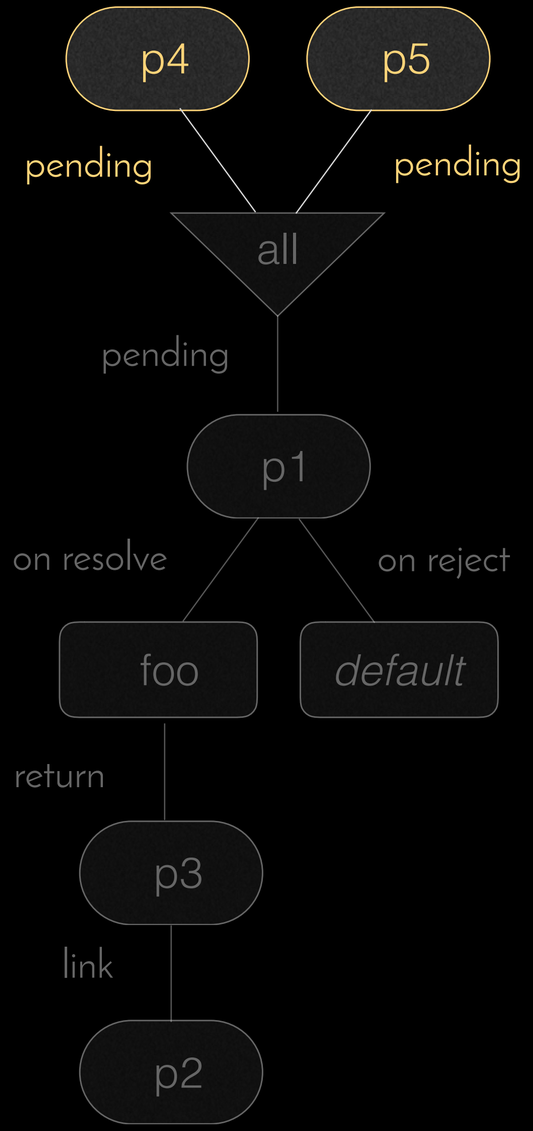
1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs

4. Analyze anti-patterns

unsettled promises

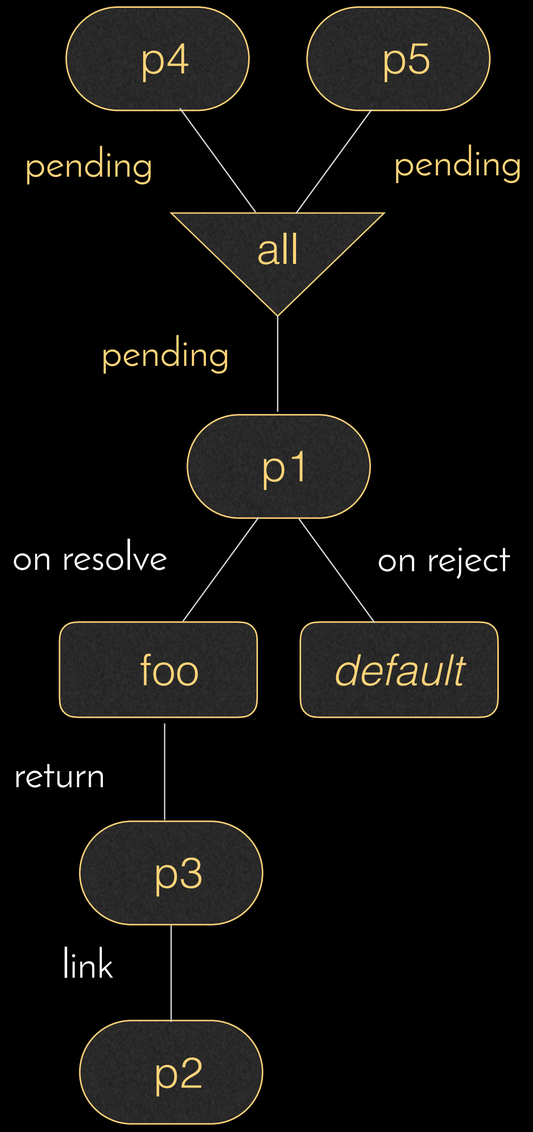


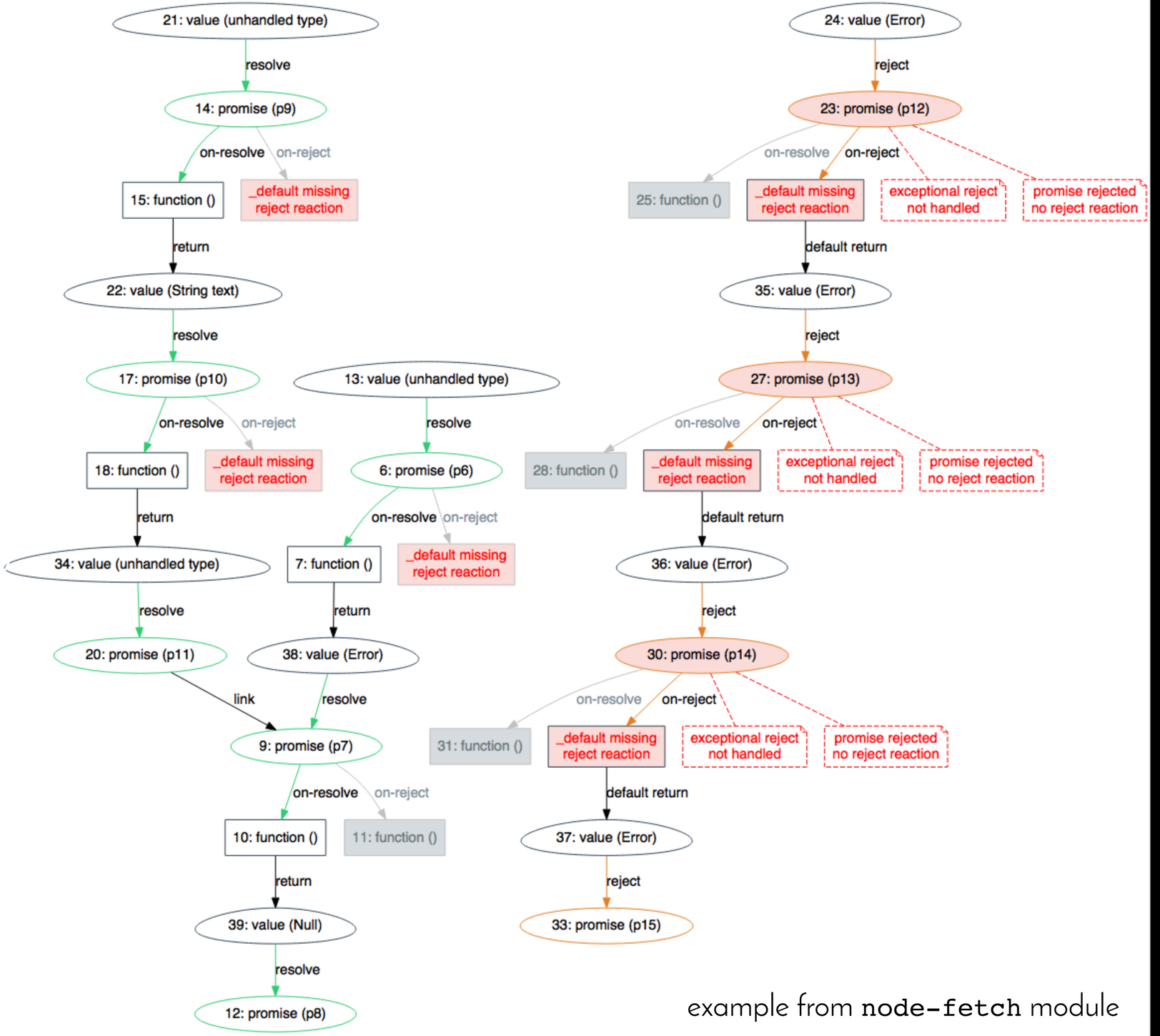
PromiseKeeper

web app

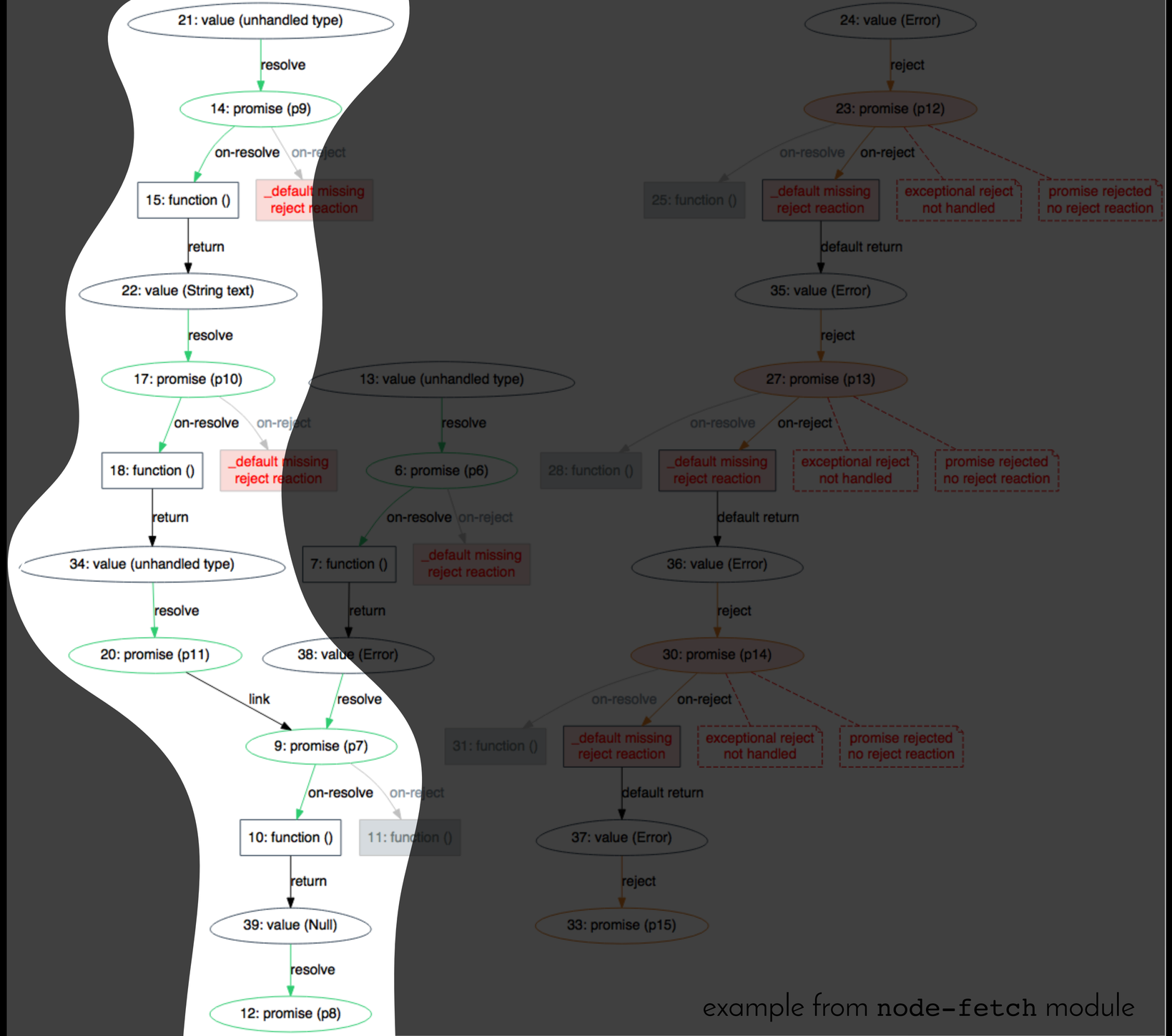
1. Instrument automatically
2. Collect execution traces
3. Infer promise graphs
4. Analyze anti-patterns

unsettled promises

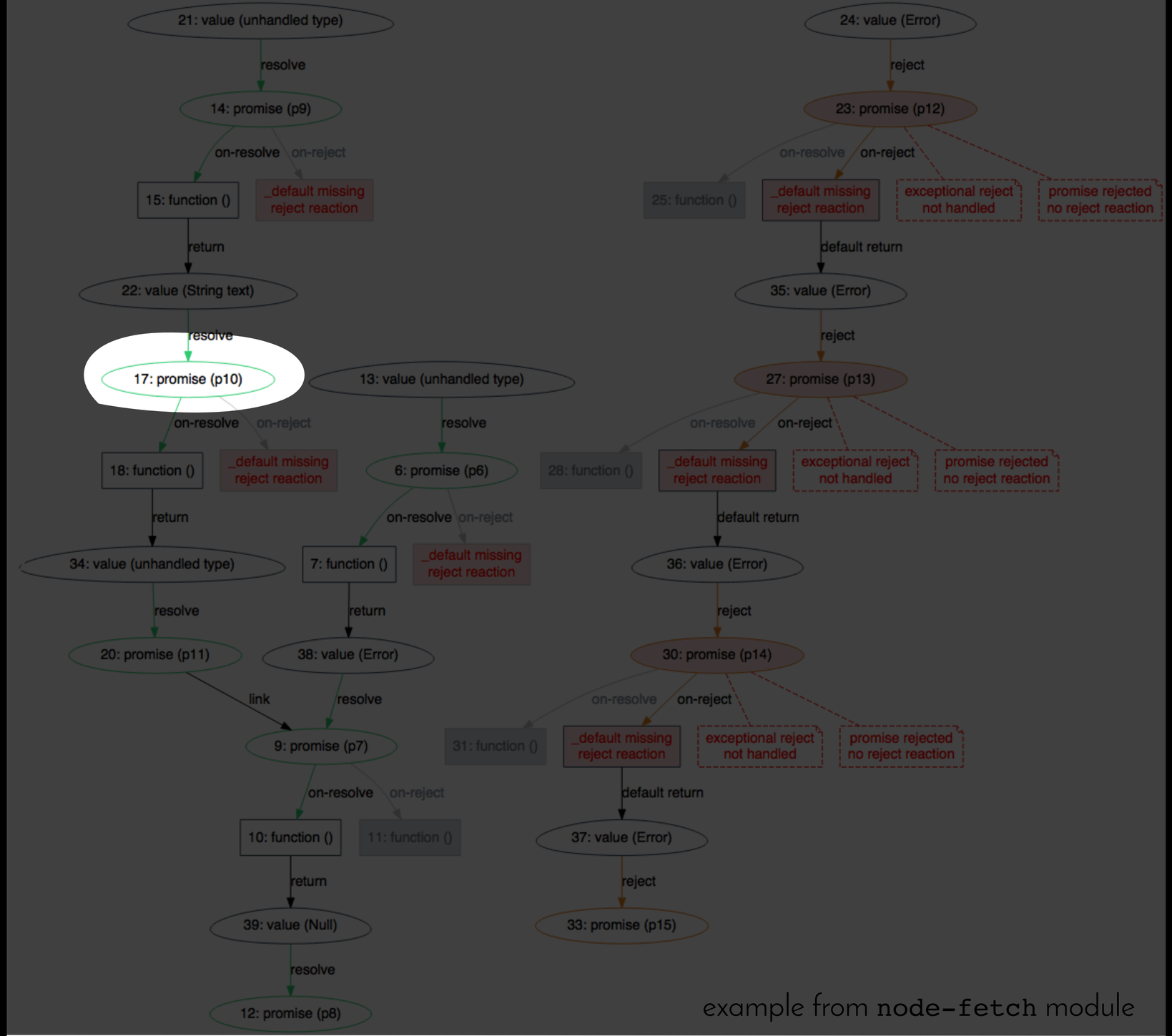




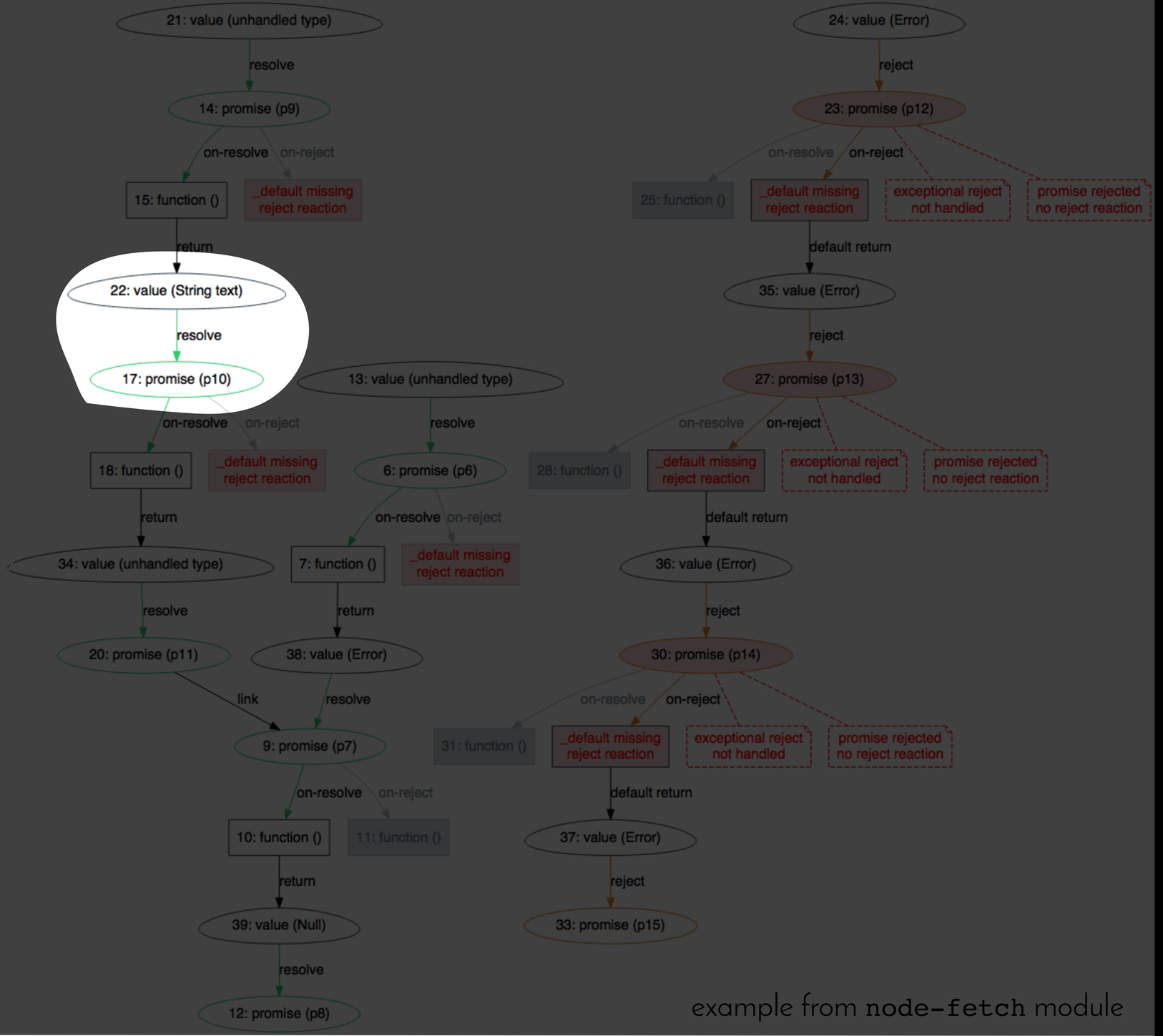
example from `node-fetch` module



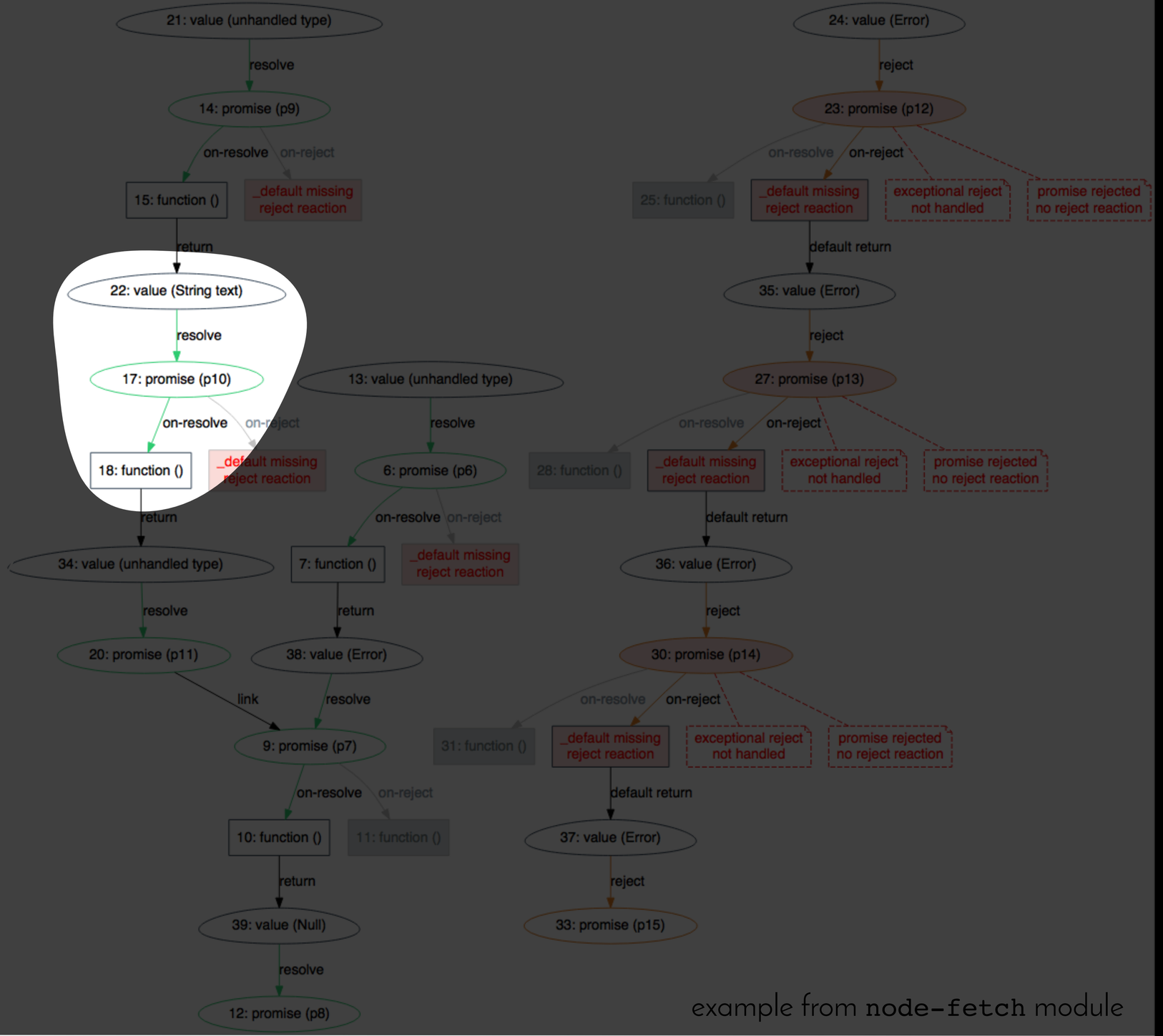
example from `node-fetch` module



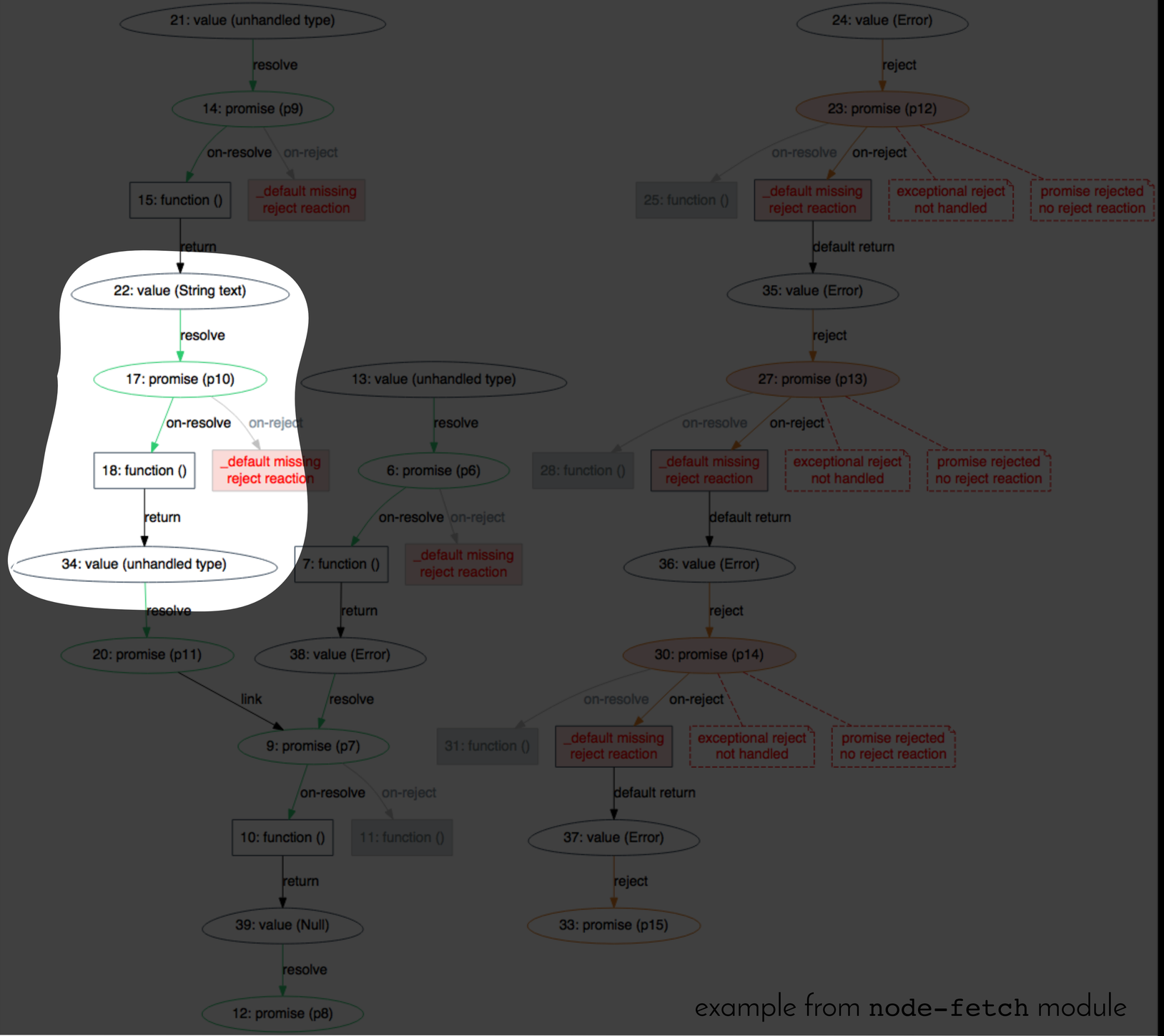
example from `node-fetch` module



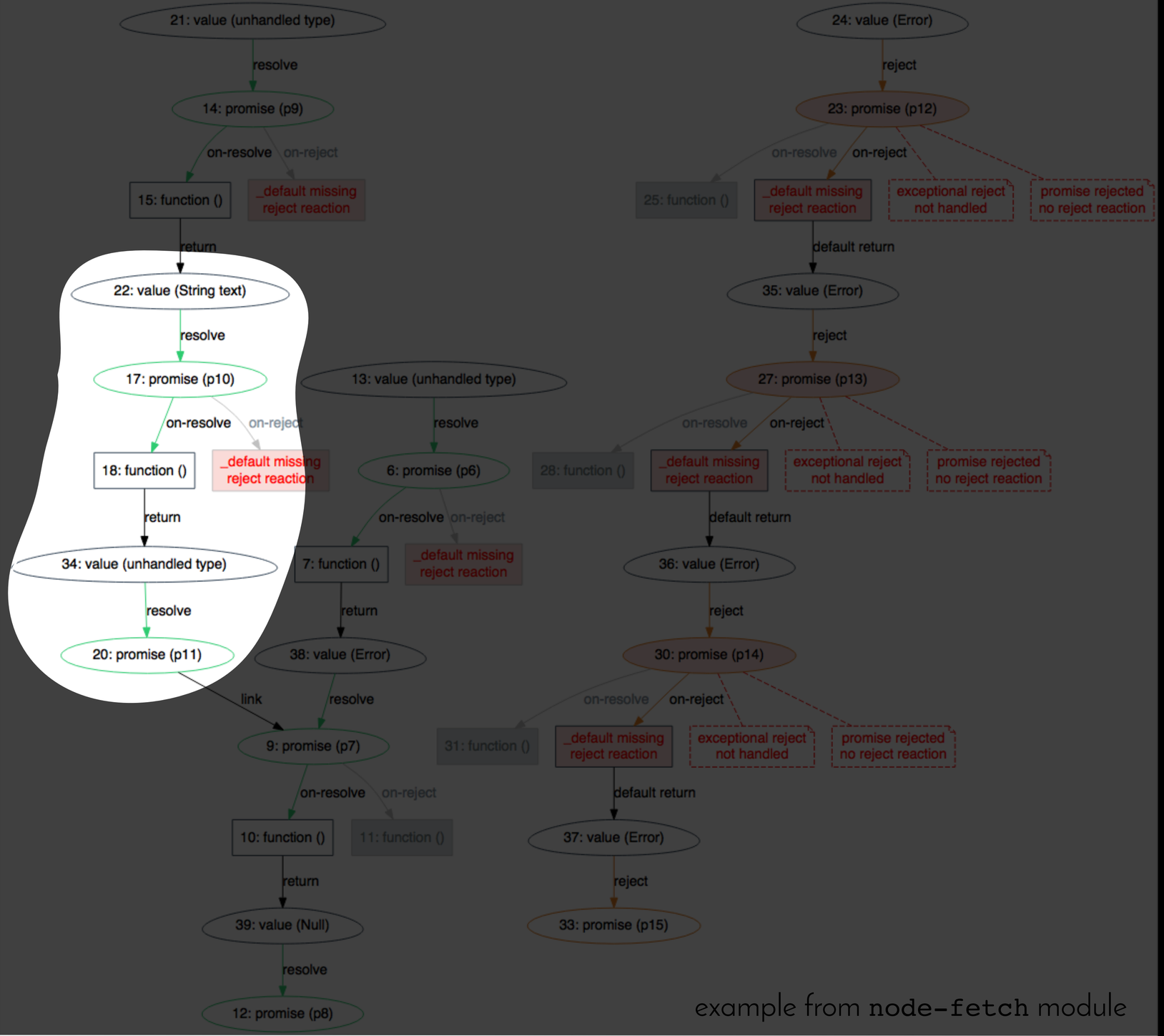
example from `node-fetch` module



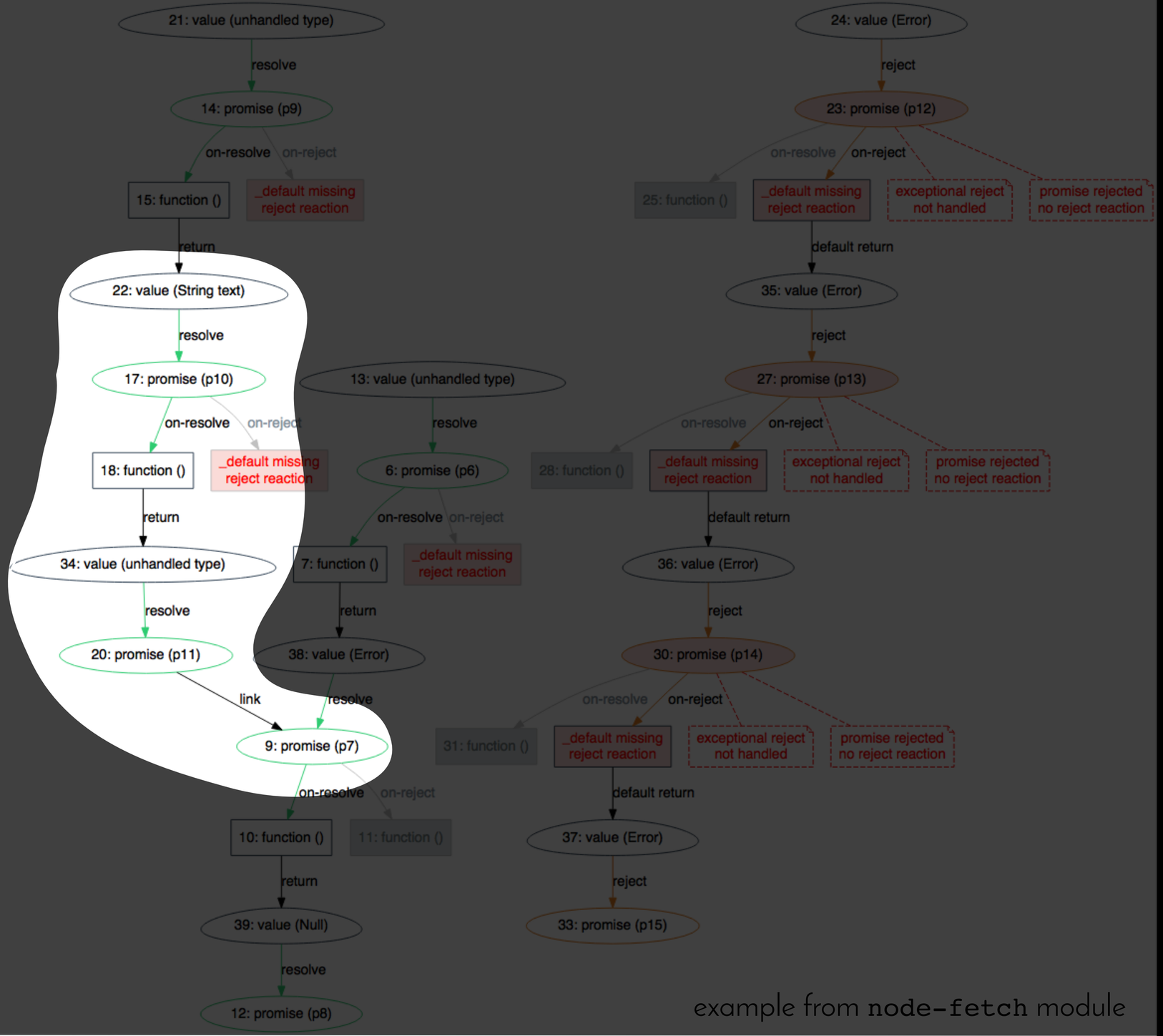
example from `node-fetch` module



example from `node-fetch` module

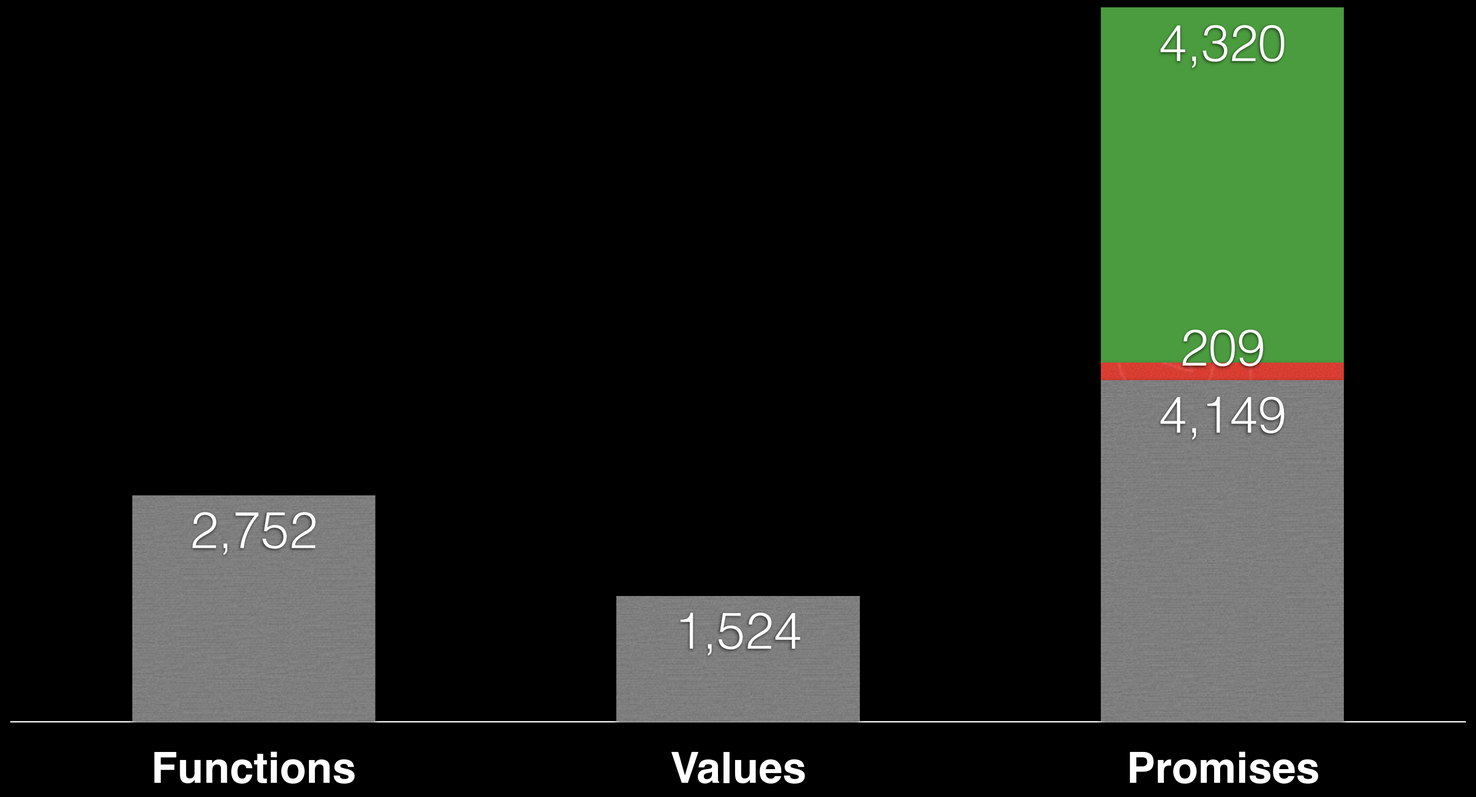


example from `node-fetch` module



example from `node-fetch` module

Broken Promises



115,000
Lines of

209

rejected promises

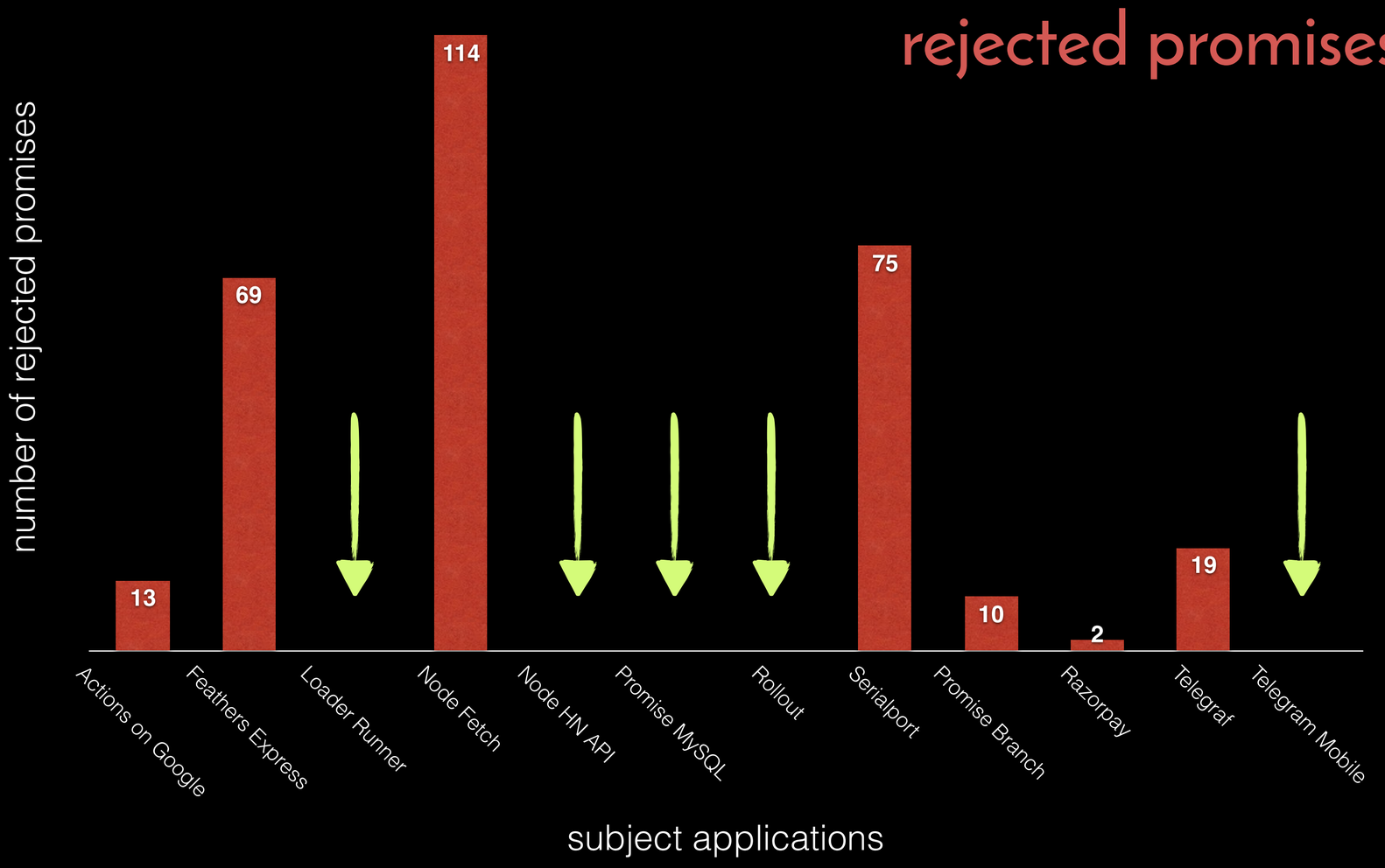
209

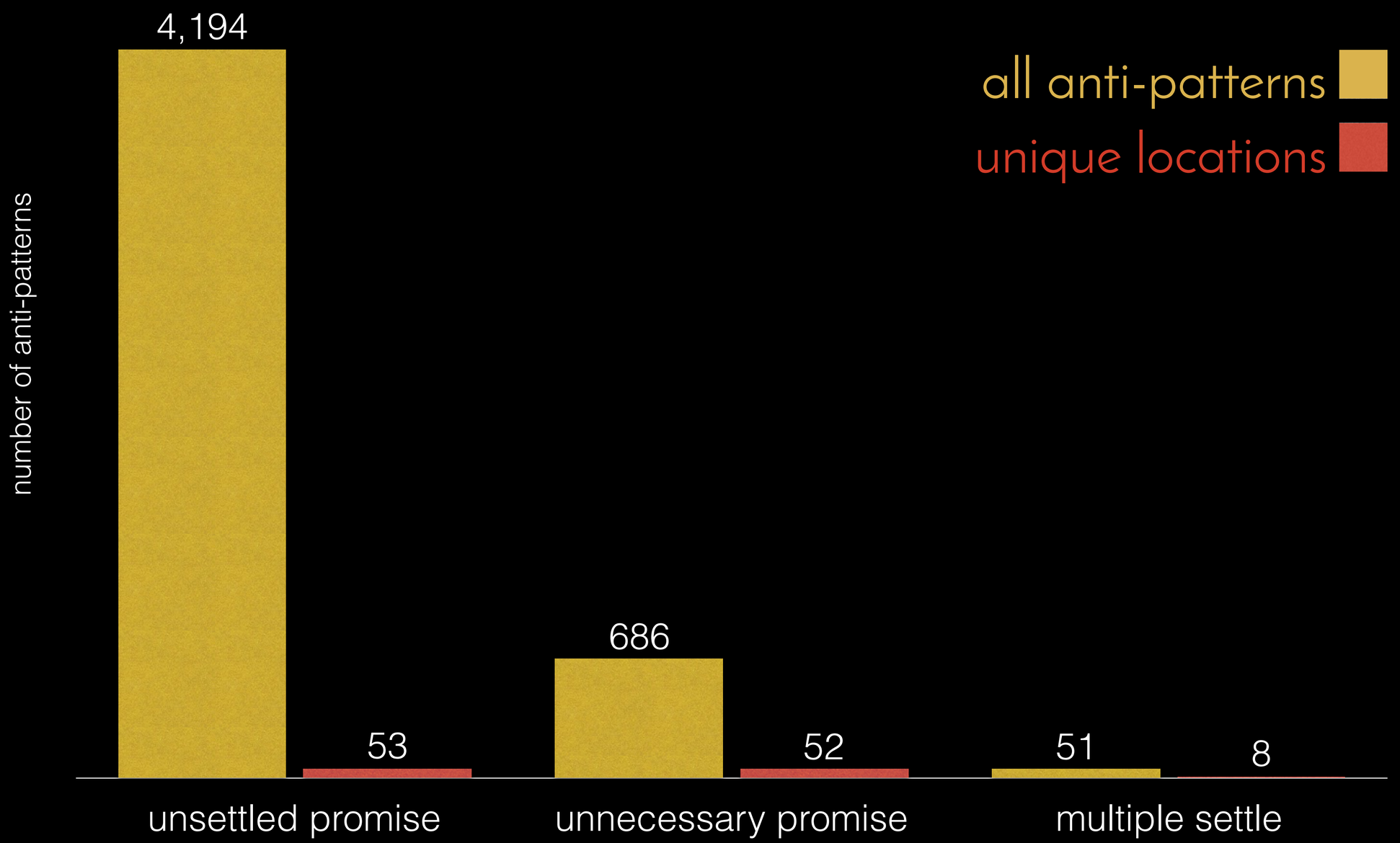
rejected promises

total

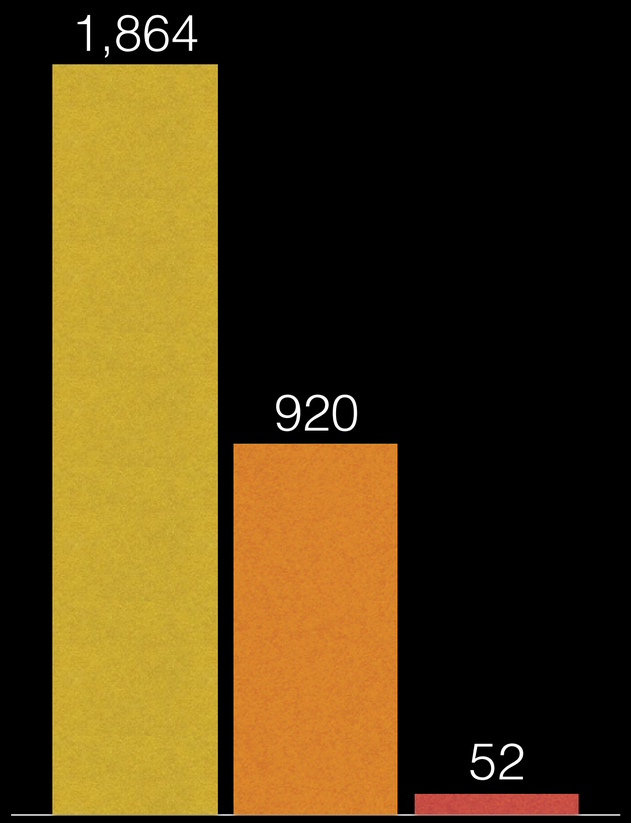
8698

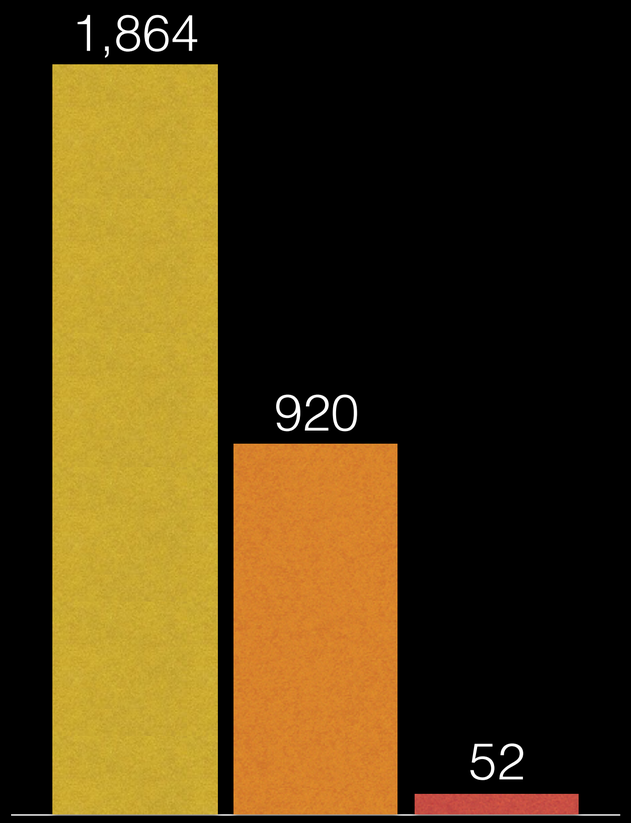
rejected promises



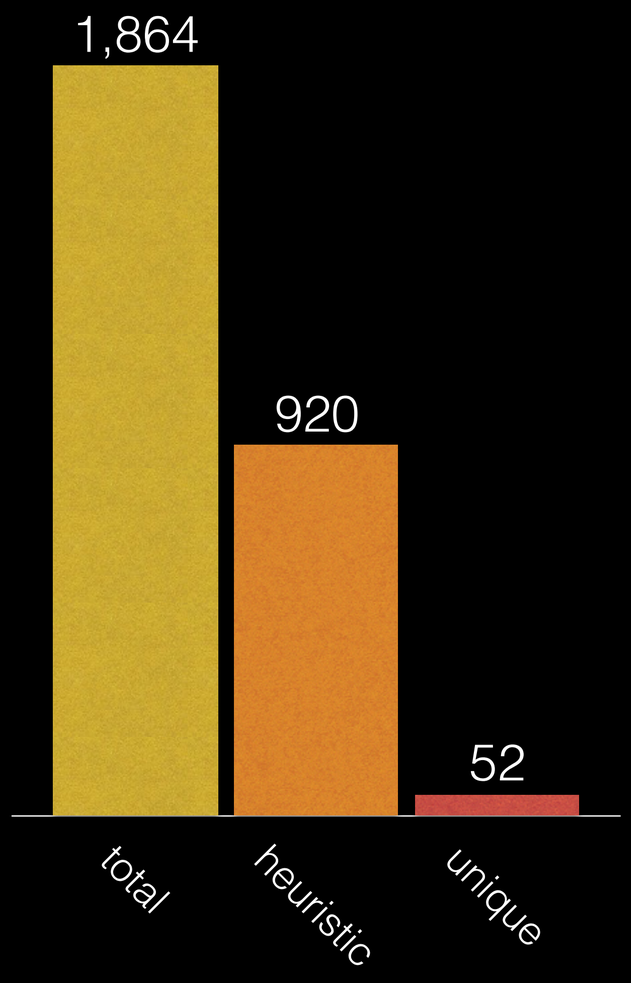

4,194
53
686
52
51
8
all anti-patterns
unique locations
number of anti-patterns
unsettled promise
unnecessary promise
multiple settle


infer anti-patterns
apply heuristics
detect unique locations
number of anti-patterns
2,224
185
88
1,864
920
52
632
456
143
unreachable reaction
missing reject reaction
broken chain

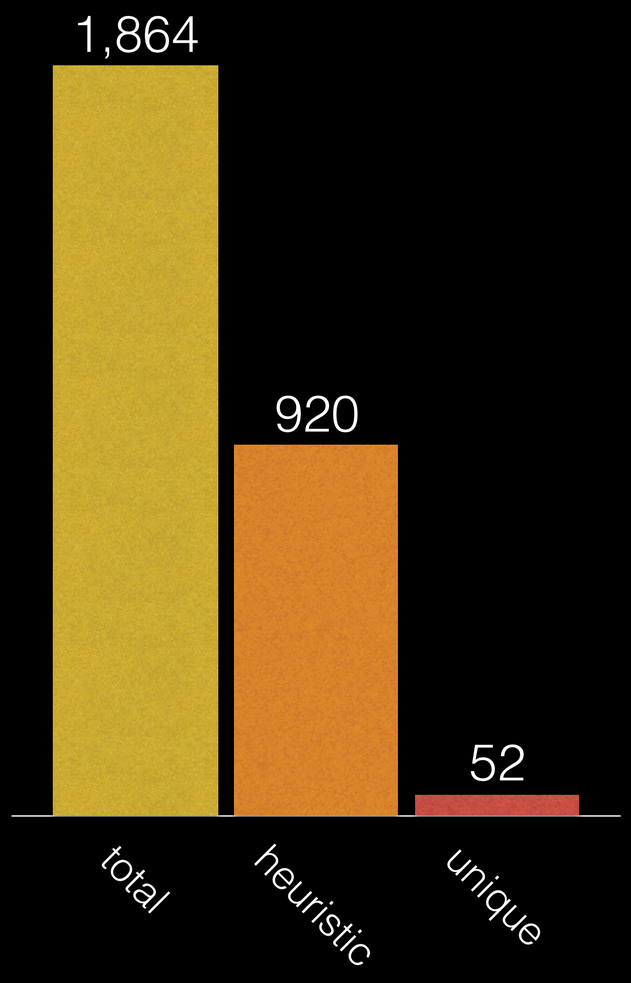

1,864
920
52
missing reject reaction


1,864
920
52

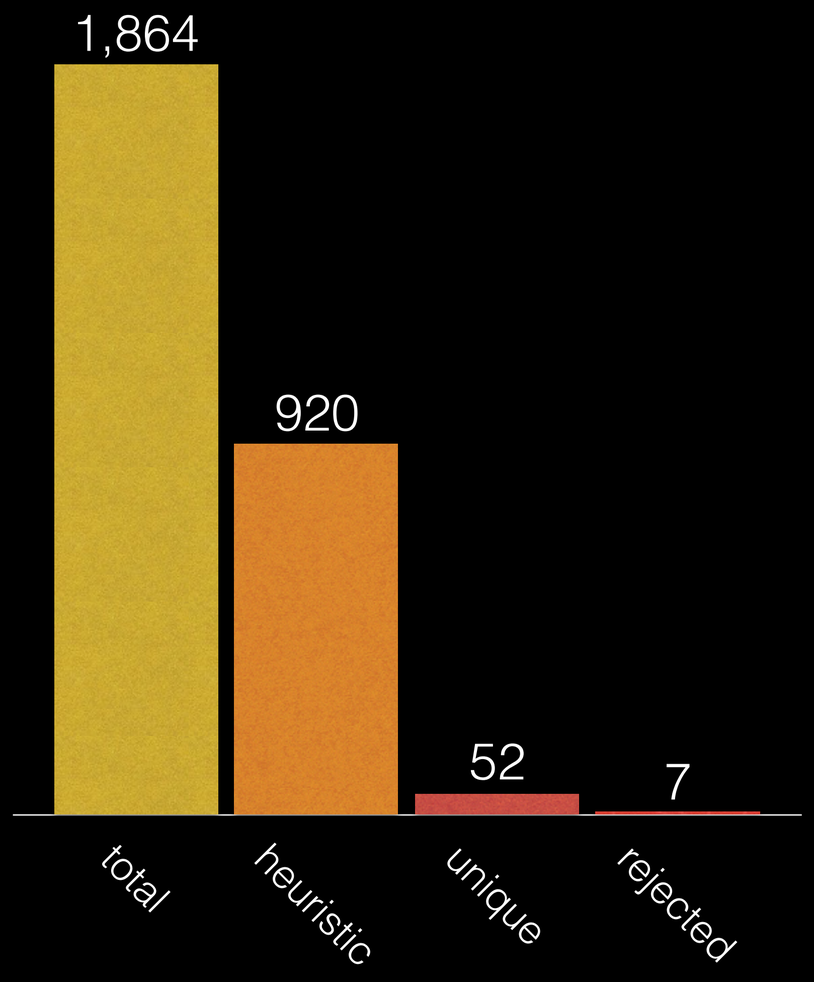
missing reject reactions

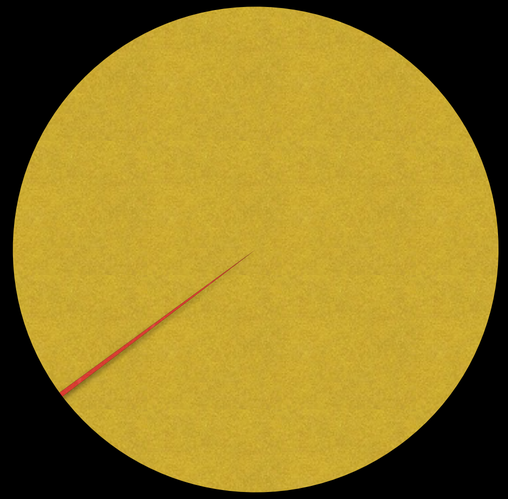


missing reject reactions

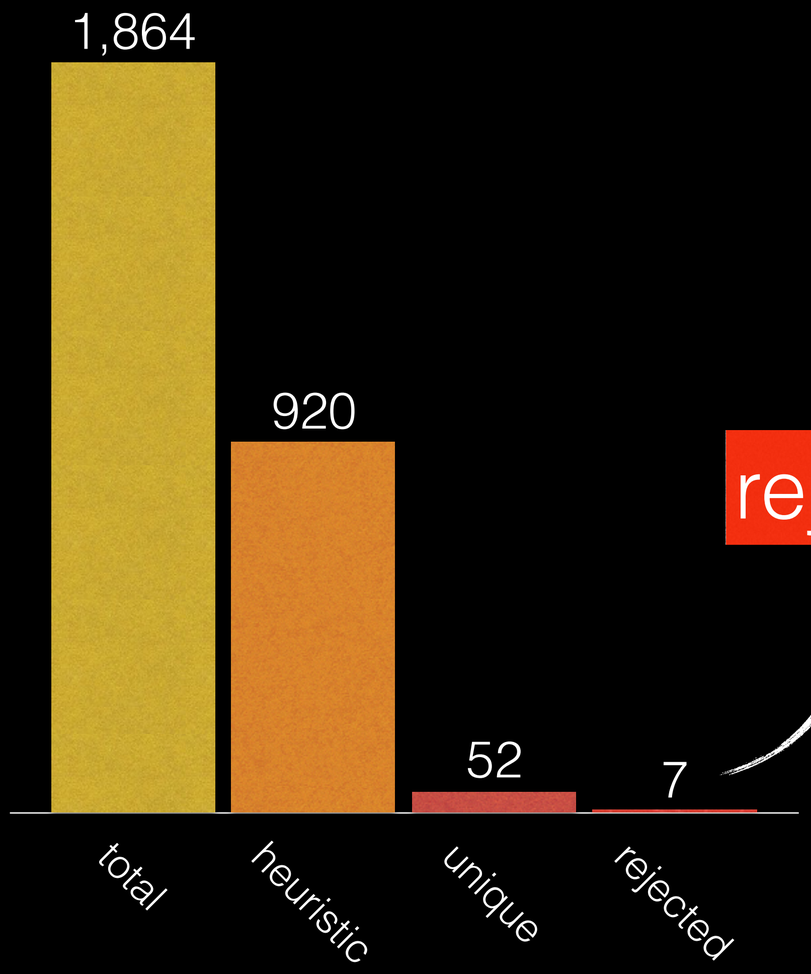


missing reject reactions

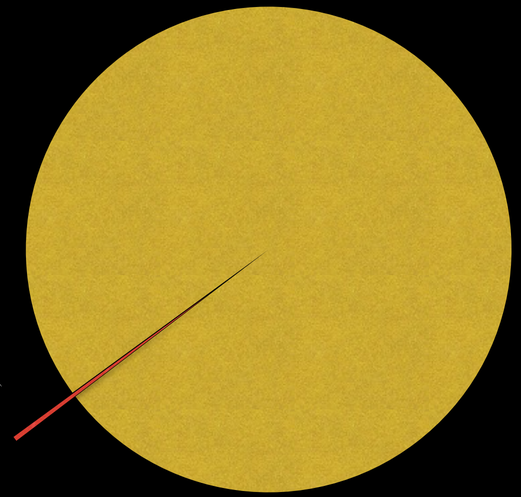




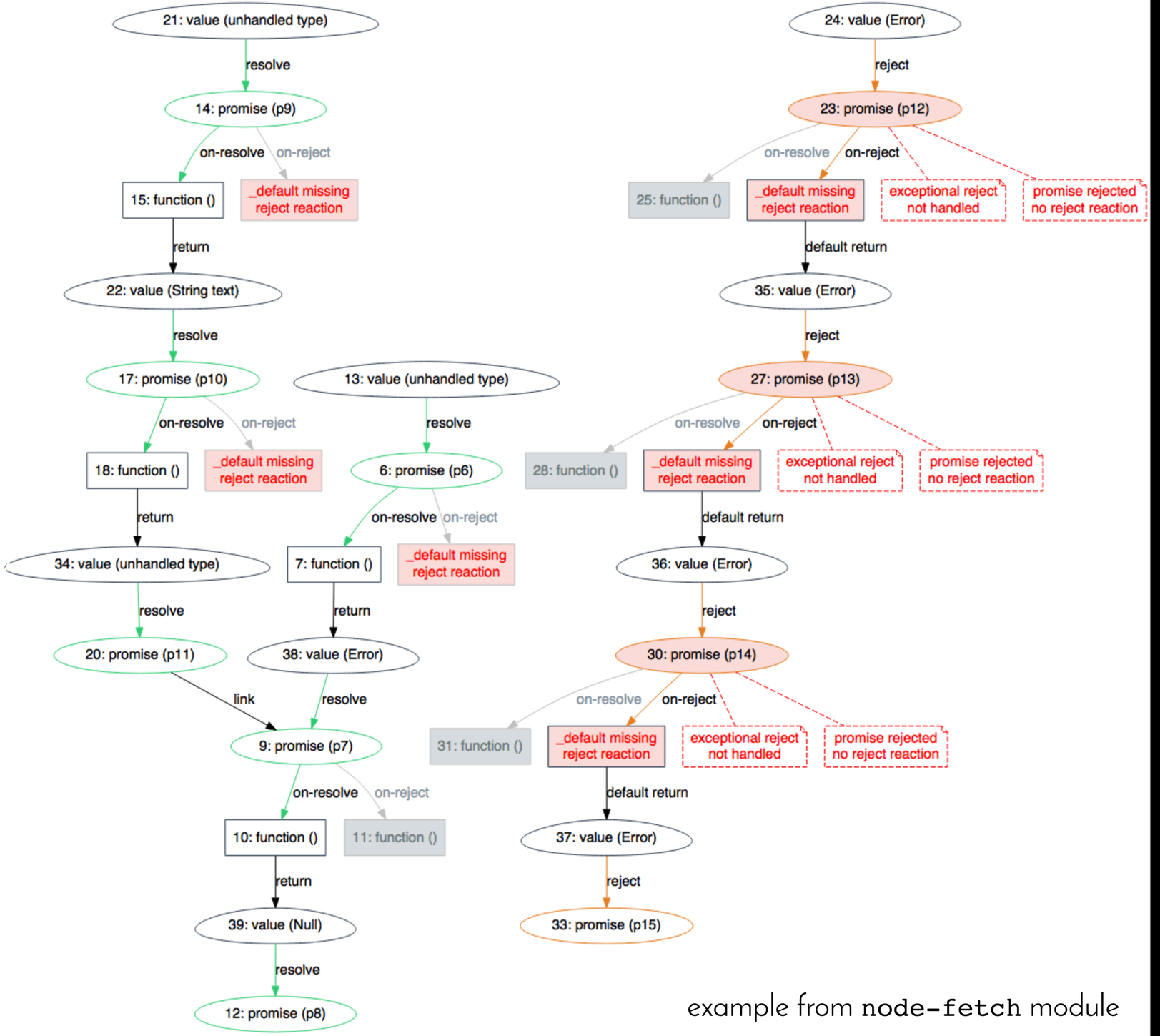
missing reject reactions



rejected

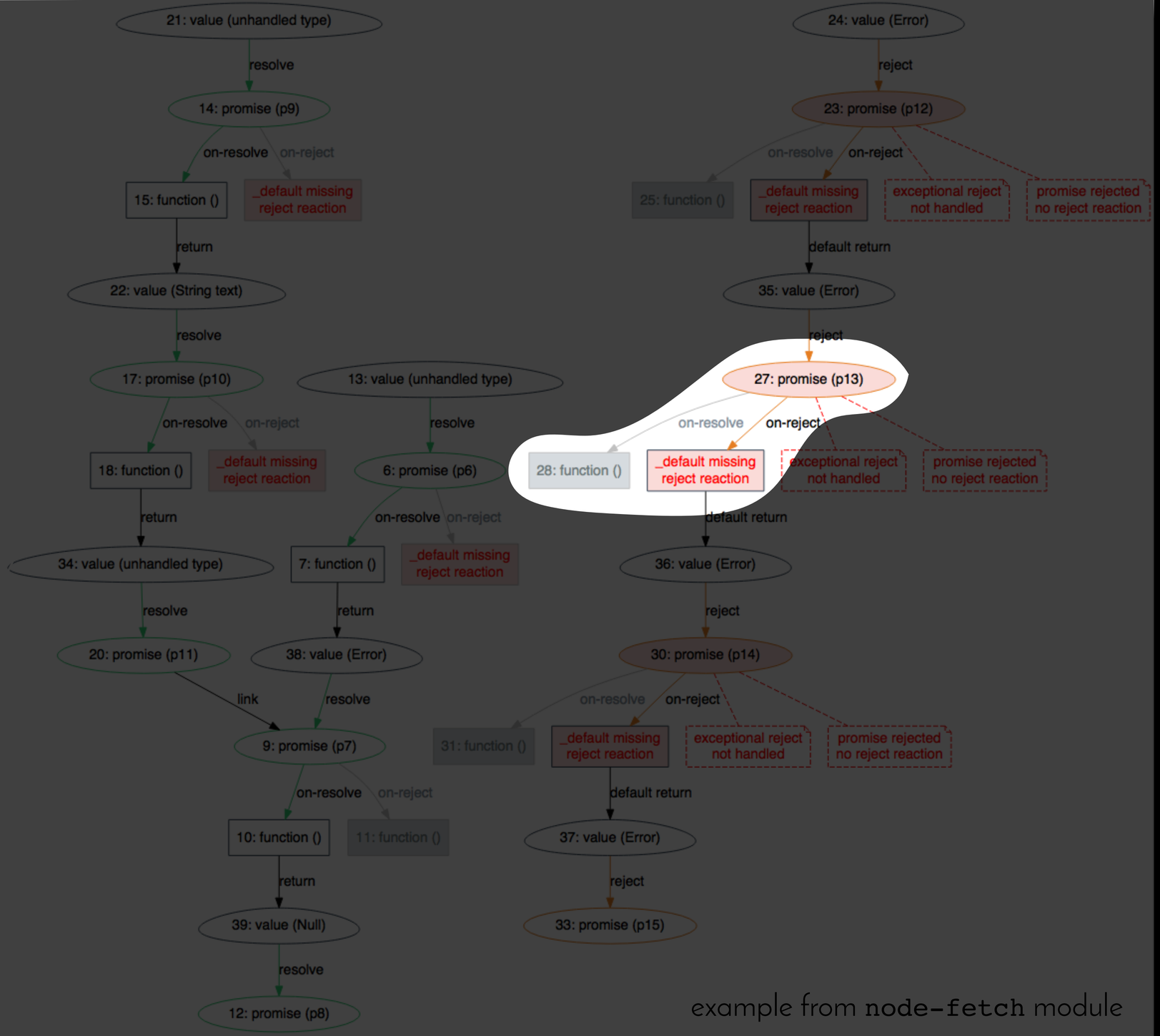


node-fetch
> 4,632,000 weekly downloads

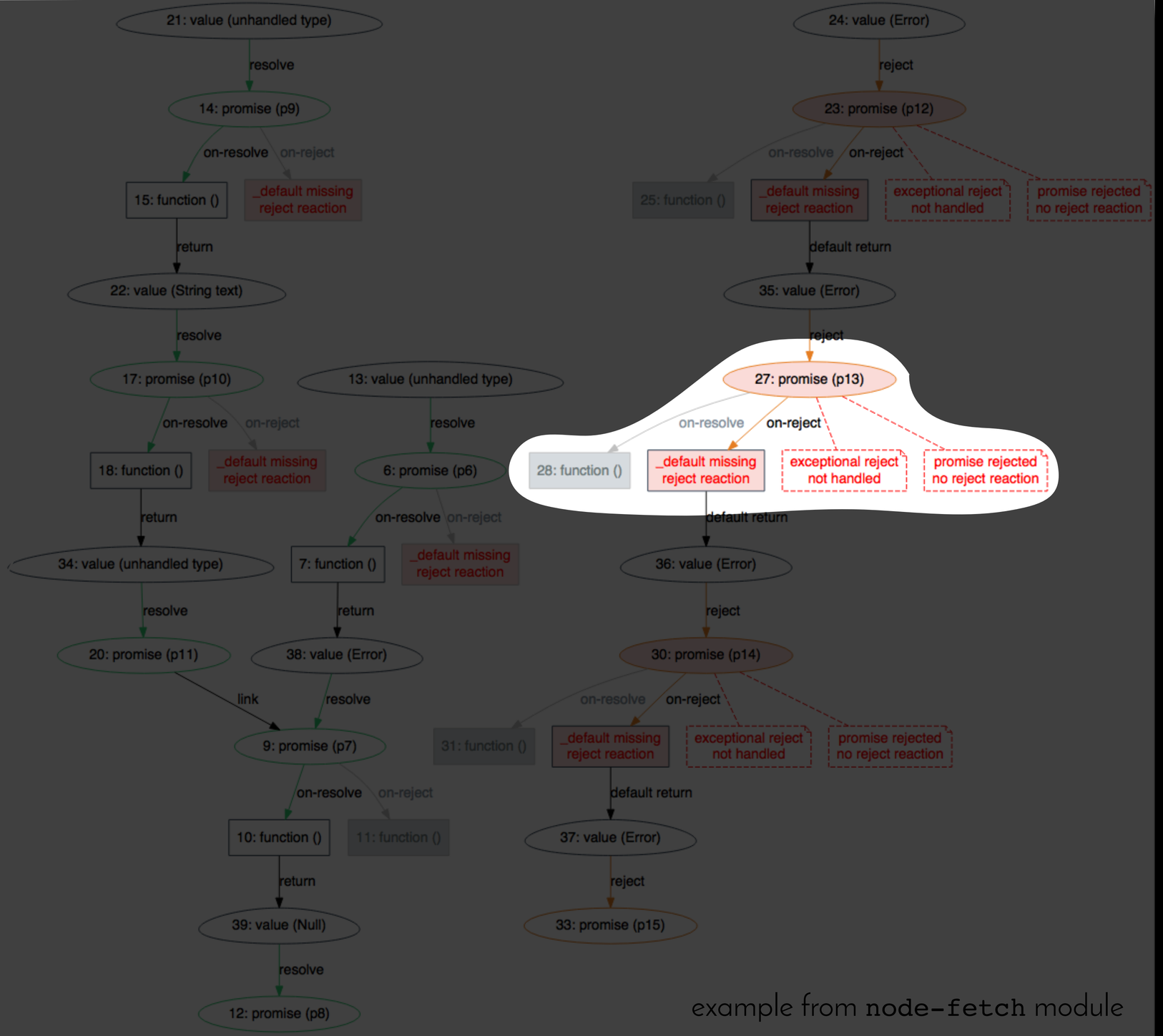


example from `node-fetch` module

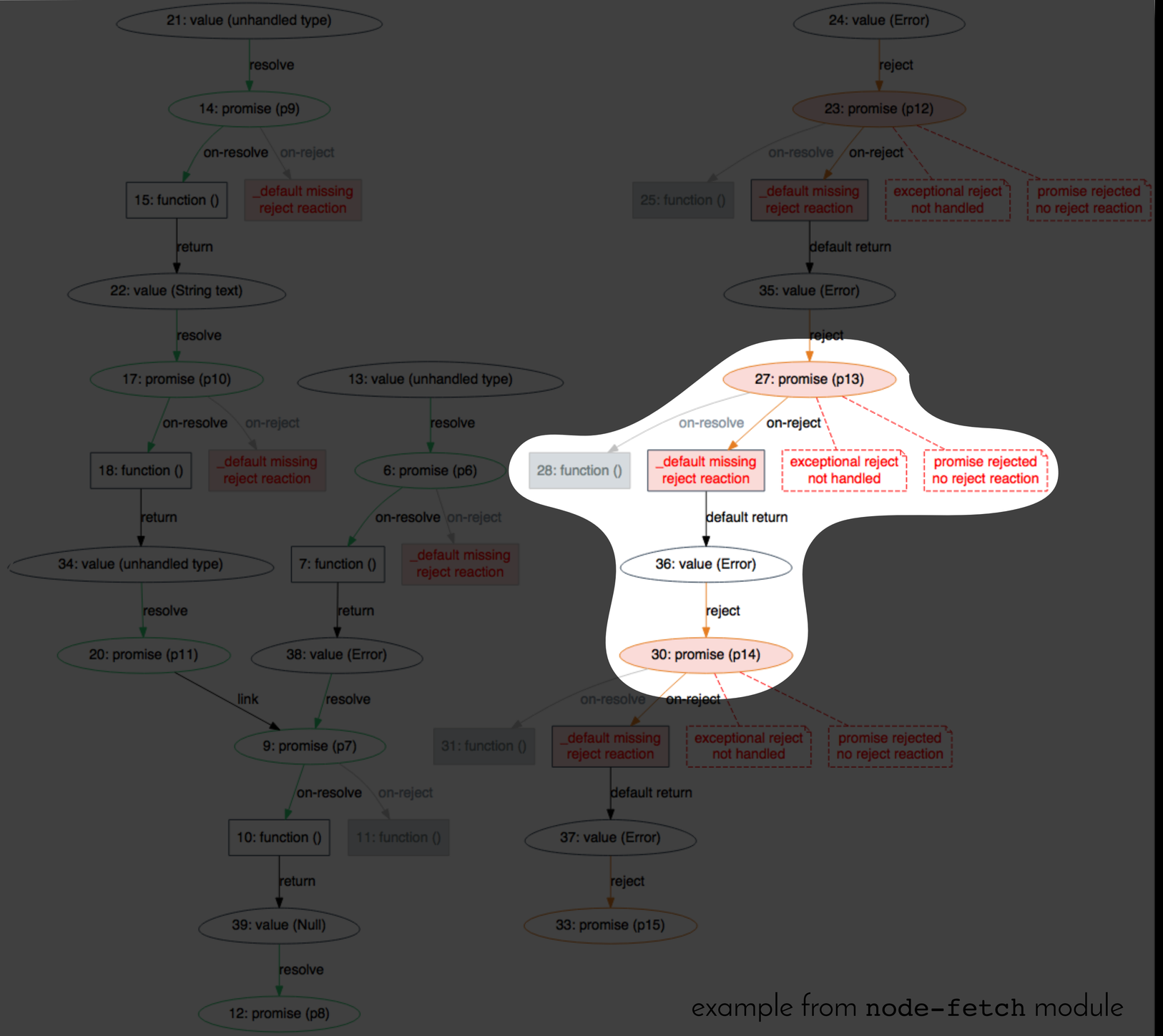
example from `node-fetch` module



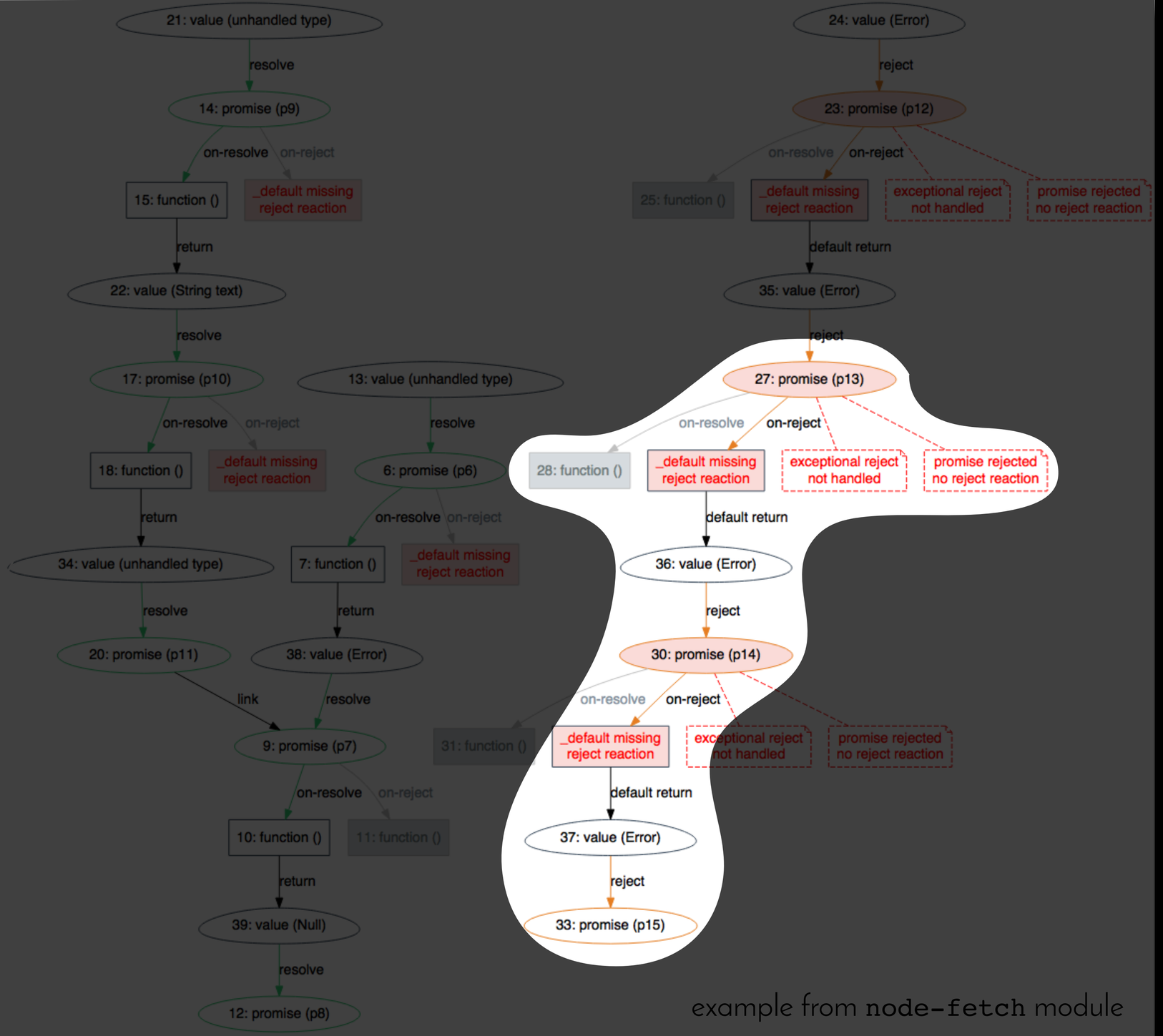
example from `node-fetch` module



example from `node-fetch` module

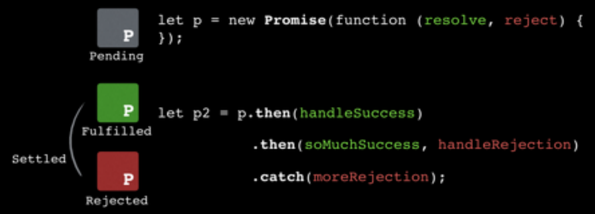


example from `node-fetch` module



example from `node-fetch` module

JavaScript Promises

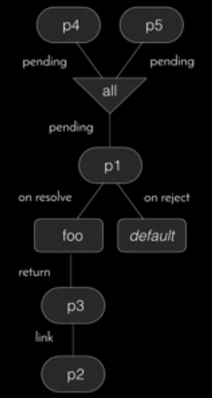


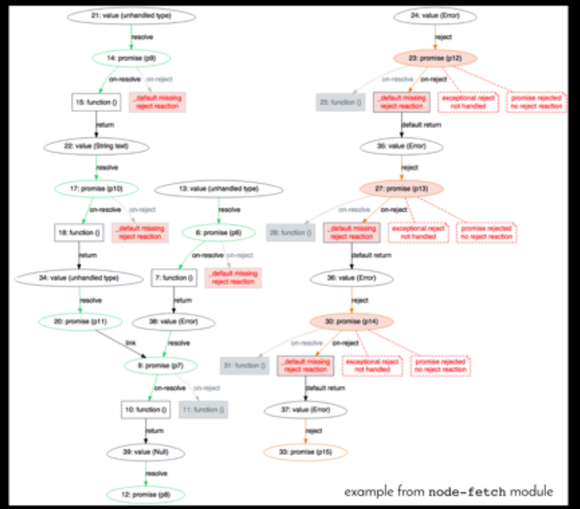

```
let p = new Promise(function (resolve, reject) {
});

let p2 = p.then(handleSuccess)
          .then(soMuchSuccess, handleRejection)
          .catch(moreRejection);
```

PromiseKeeper

web app →

1. Instrument automatically
2. Collect execution traces
3. Infer promise graphs
4. Analyze anti-patterns





HIRE ME!

@saba_a