

Finding Broken Promises In Asynchronous JavaScript Programs

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



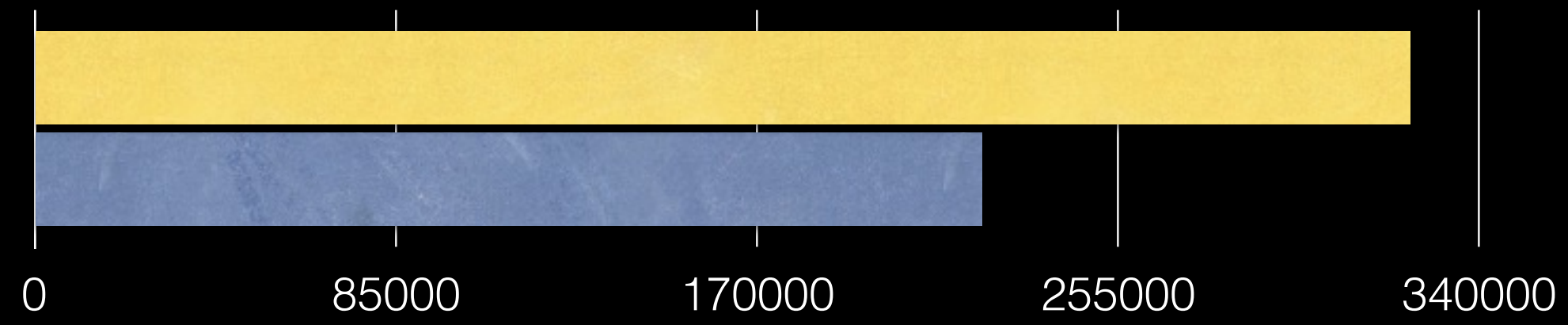
Northeastern
University



AARHUS
UNIVERSITET



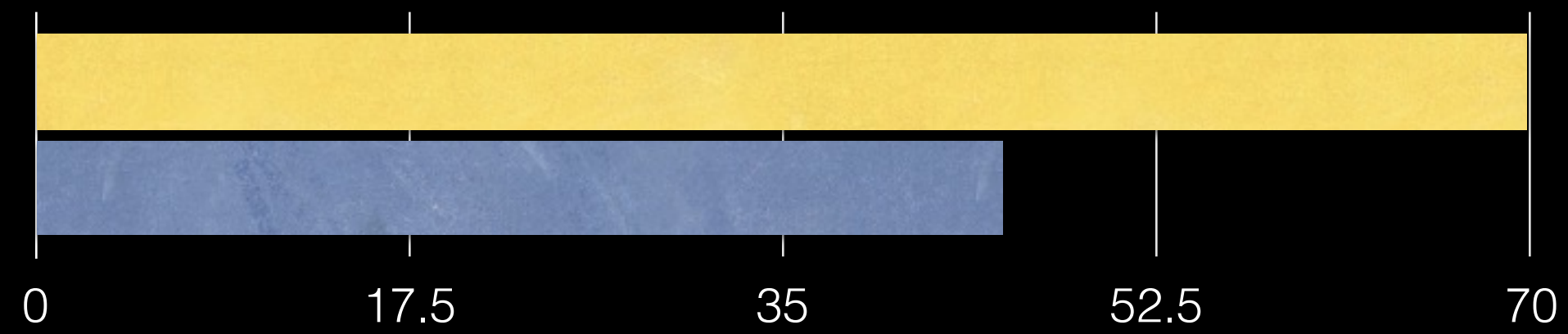
JavaScript
Java



active repositories



JavaScript
Java

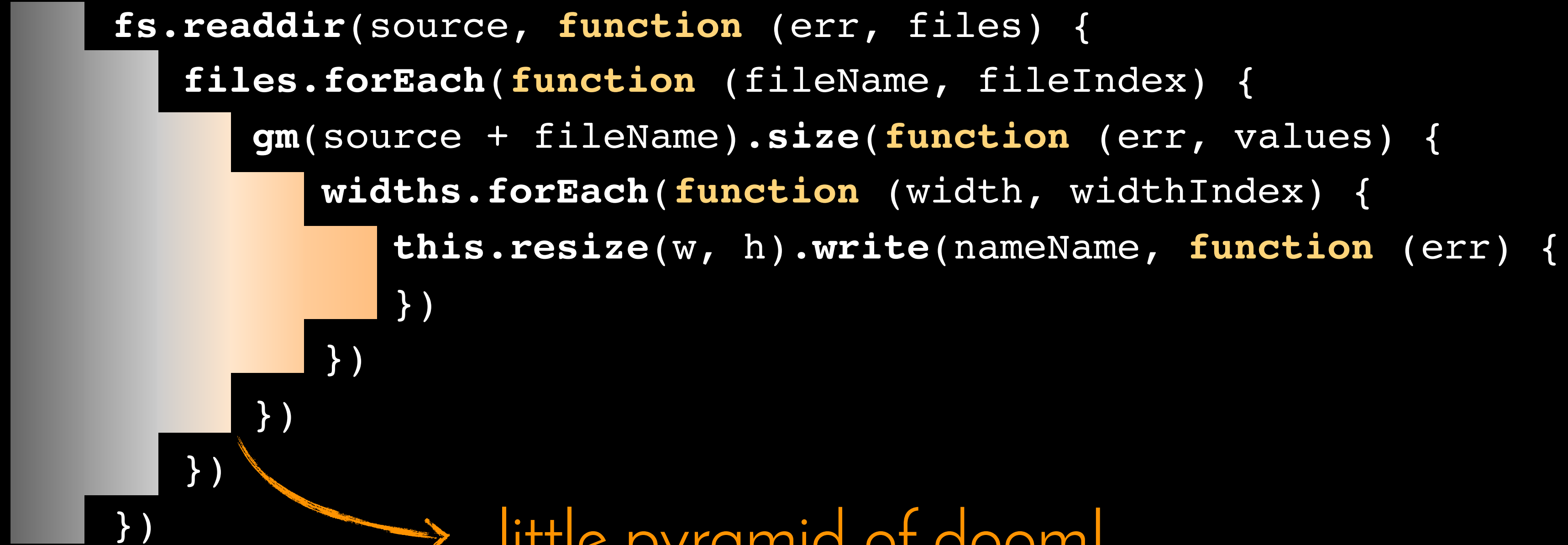


popularity %

J S

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



Pending

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



Fulfilled

```
let p2 = p.then(handleSuccess);
```

```
.then(soMuchSuccess, handleRejection);
```



Rejected

```
.catch(moreRejection);
```

Settled

```
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);
    }
  })
+ .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

The image displays a collage of overlapping browser windows from Stack Overflow. The primary window in the foreground shows a question titled "How to debug javascript promises?". The question text reads: "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 occurred')); 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?". The question has 61 votes and was asked 4 years ago, viewed 27,061 times, and is active for 3 months. The answer section shows a code snippet:

```
return async4();
```

 and another snippet:

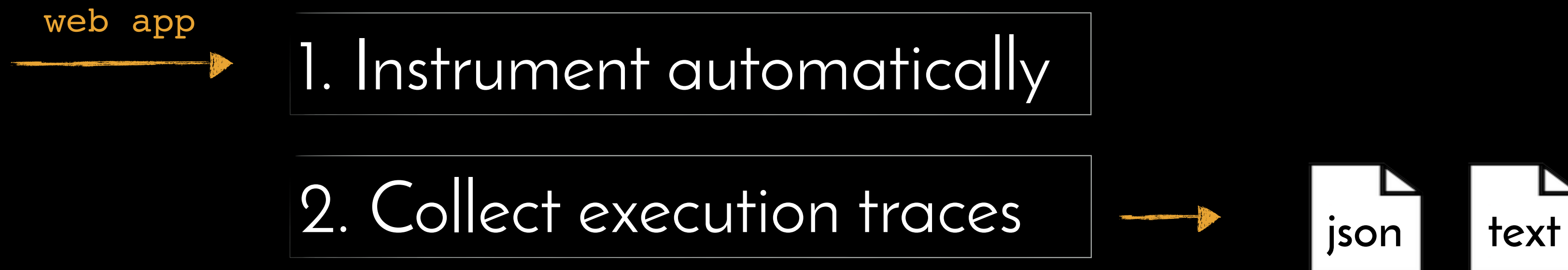
```
var serverSidePromiseChain;
```

. Other visible windows show questions like "Catch all unhandled javascript promise rejections" and "How can I synchronously determine if a promise is resolved?". The interface includes navigation menus, search bars, and various UI elements typical of a Q&A website.

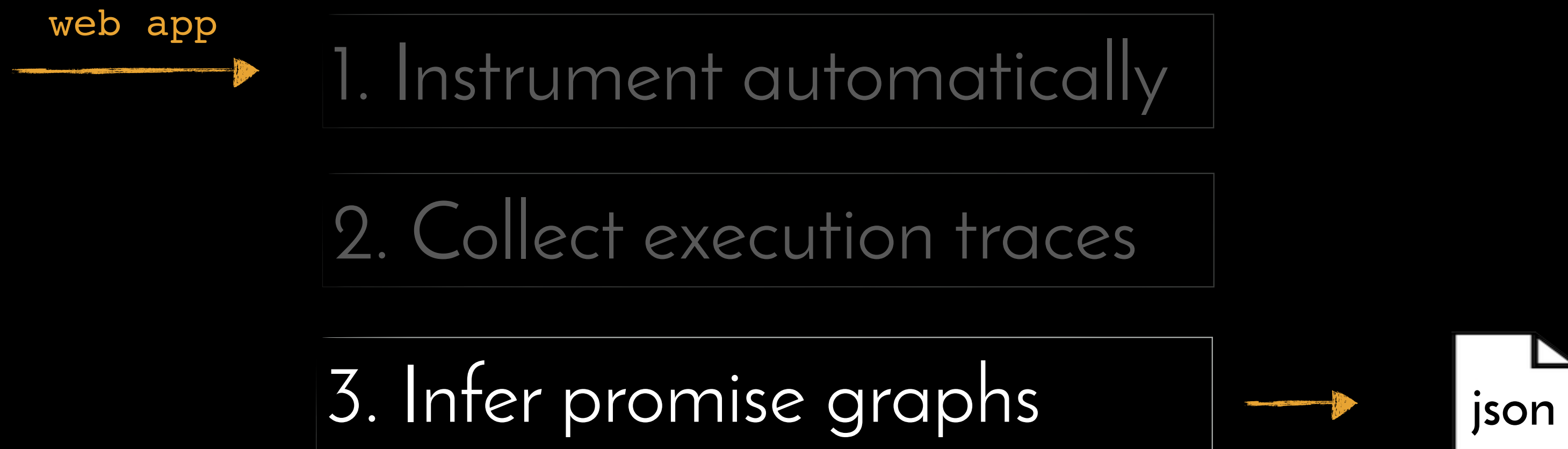


PromiseKeeper

PromiseKeeper



PromiseKeeper



[Madsen et al., 2017]

PromiseKeeper

web app

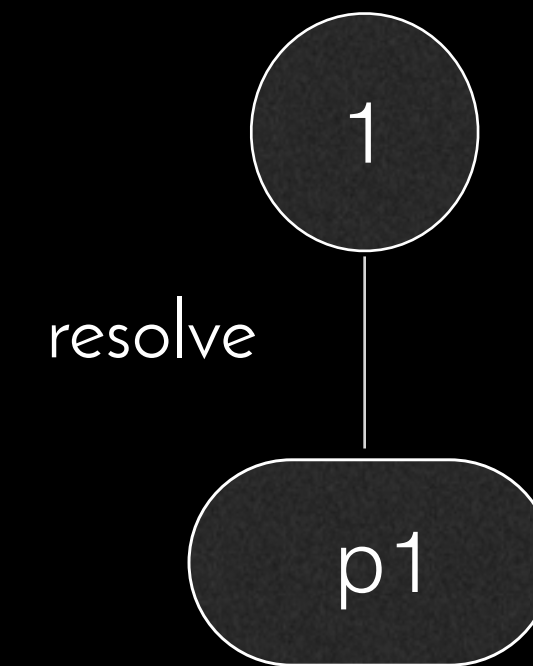


1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs

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



PromiseKeeper

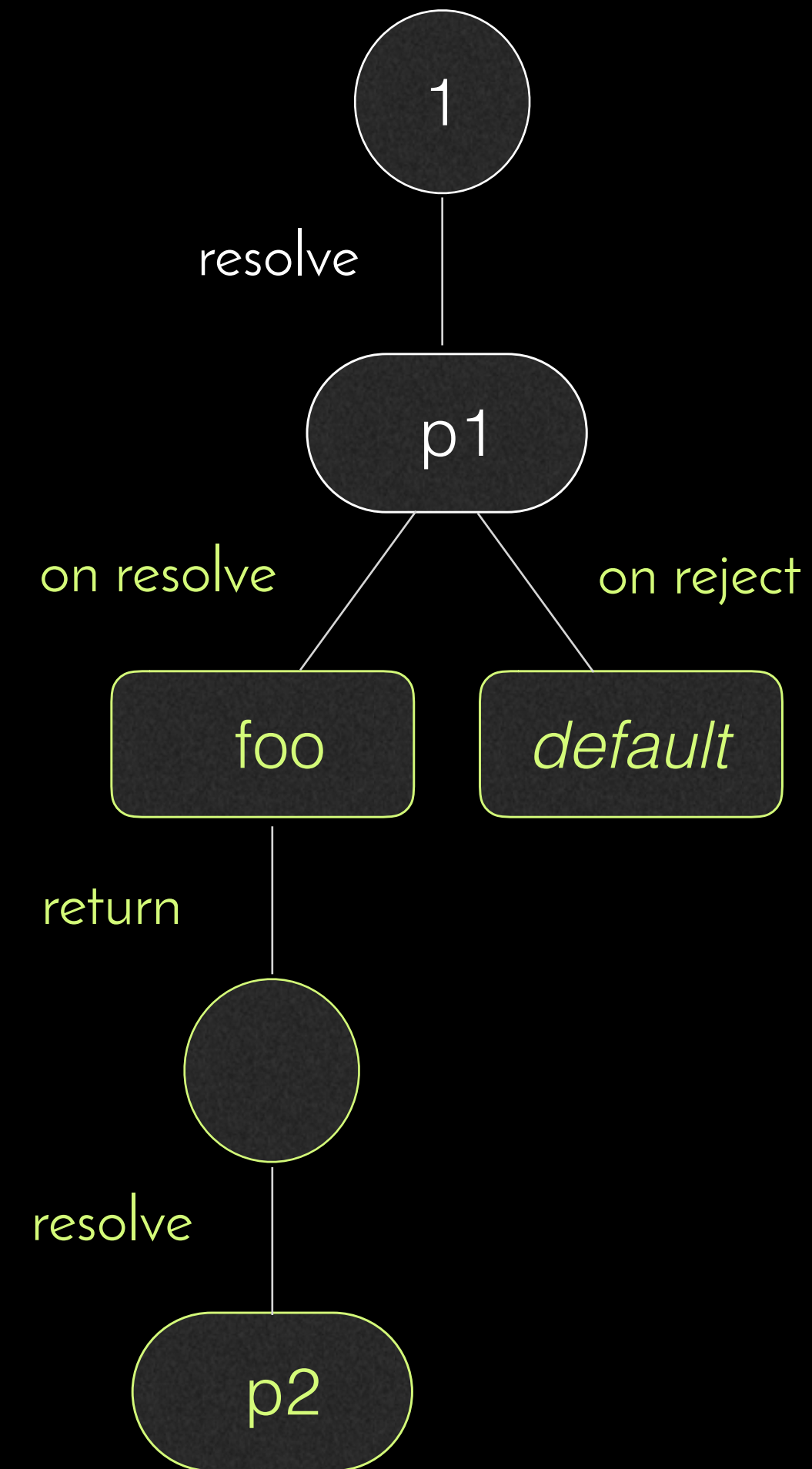
web app

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

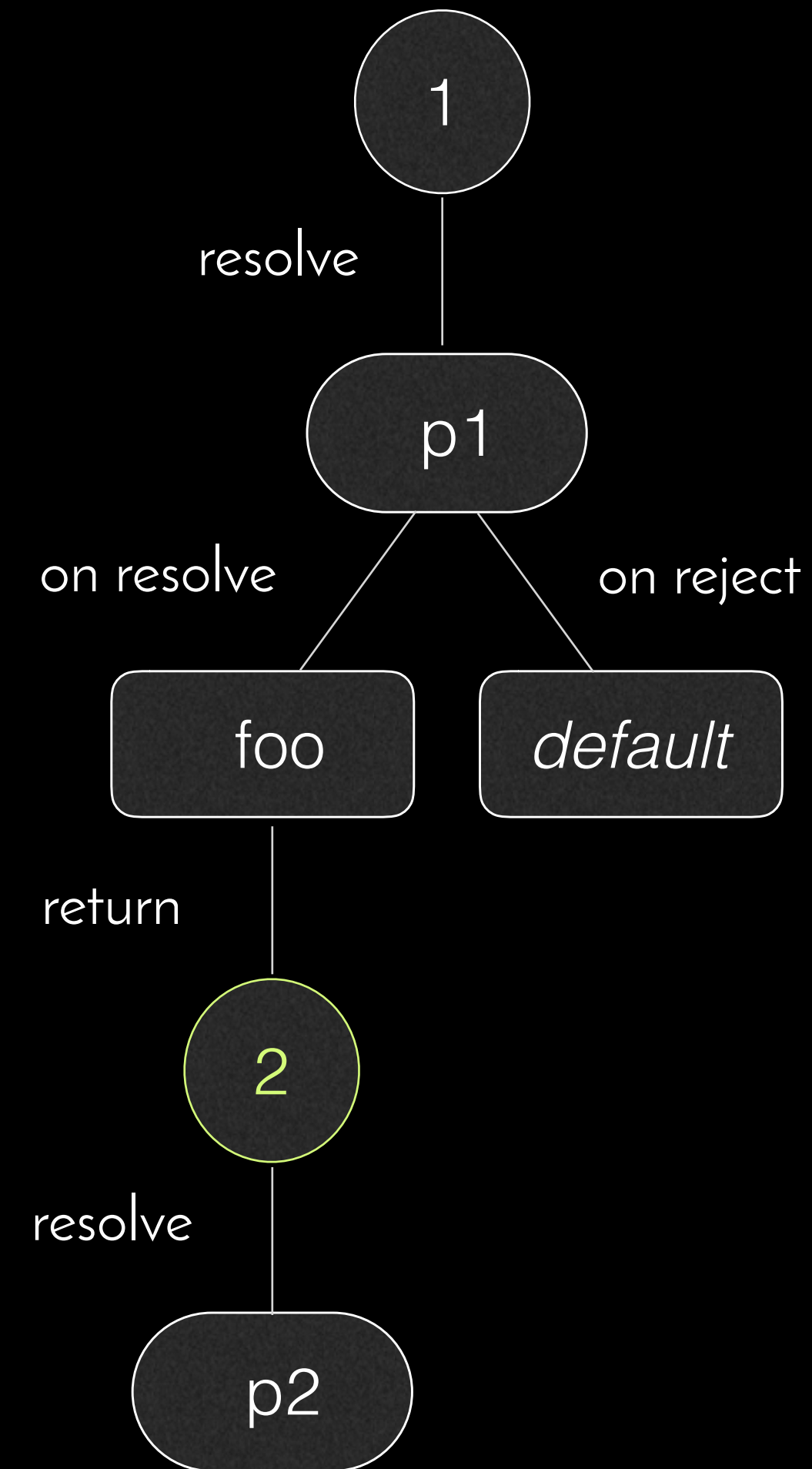
web app

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

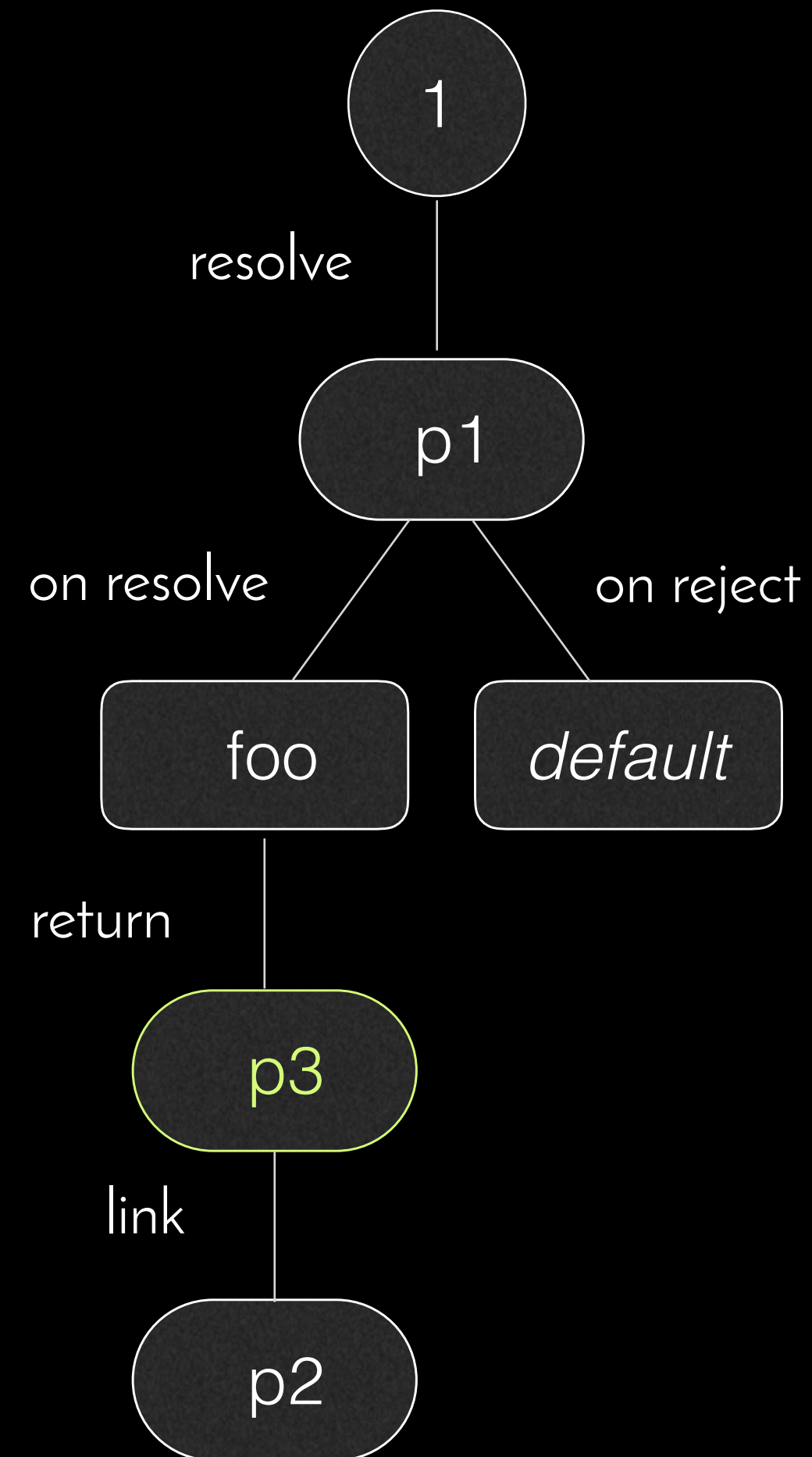
web app

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

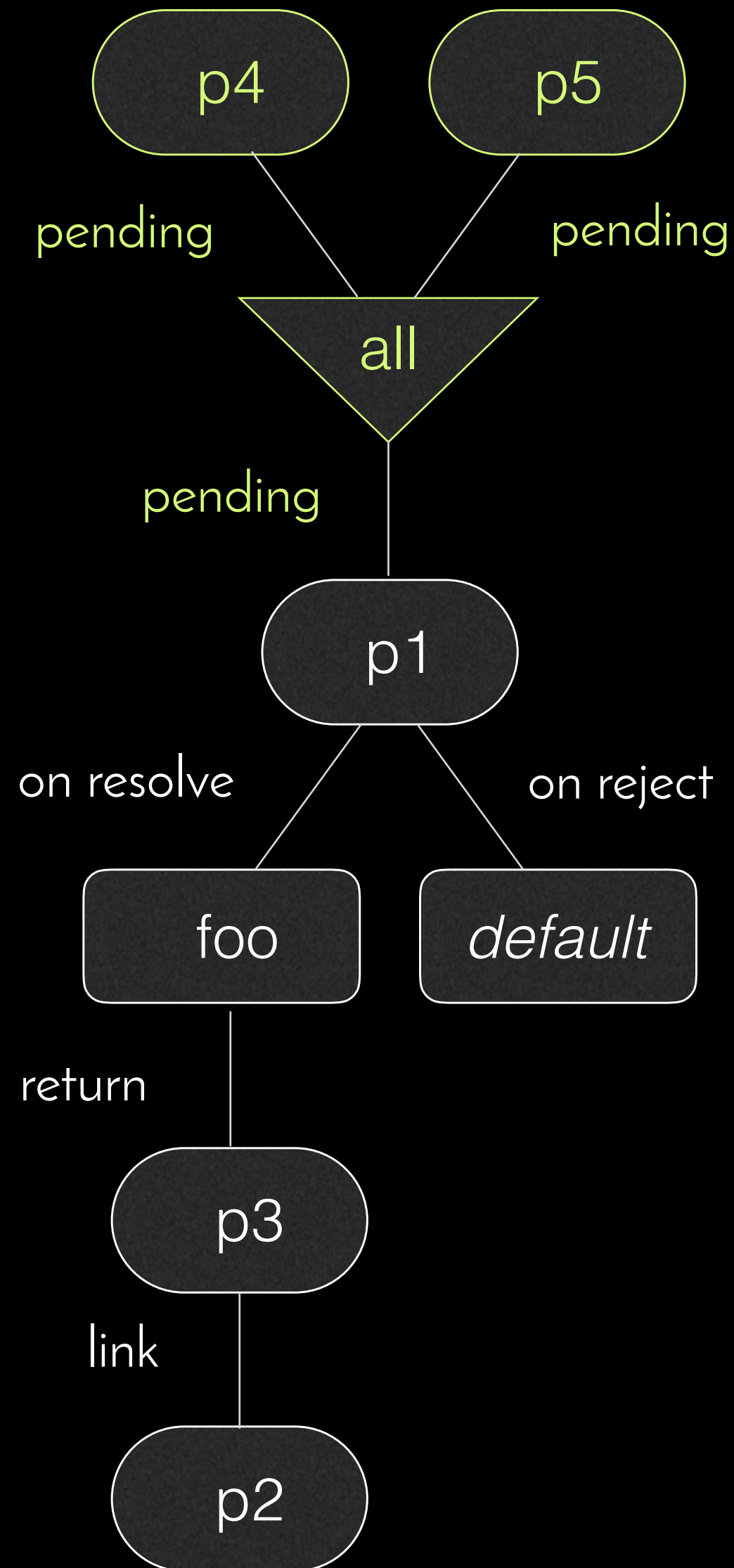
web app

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

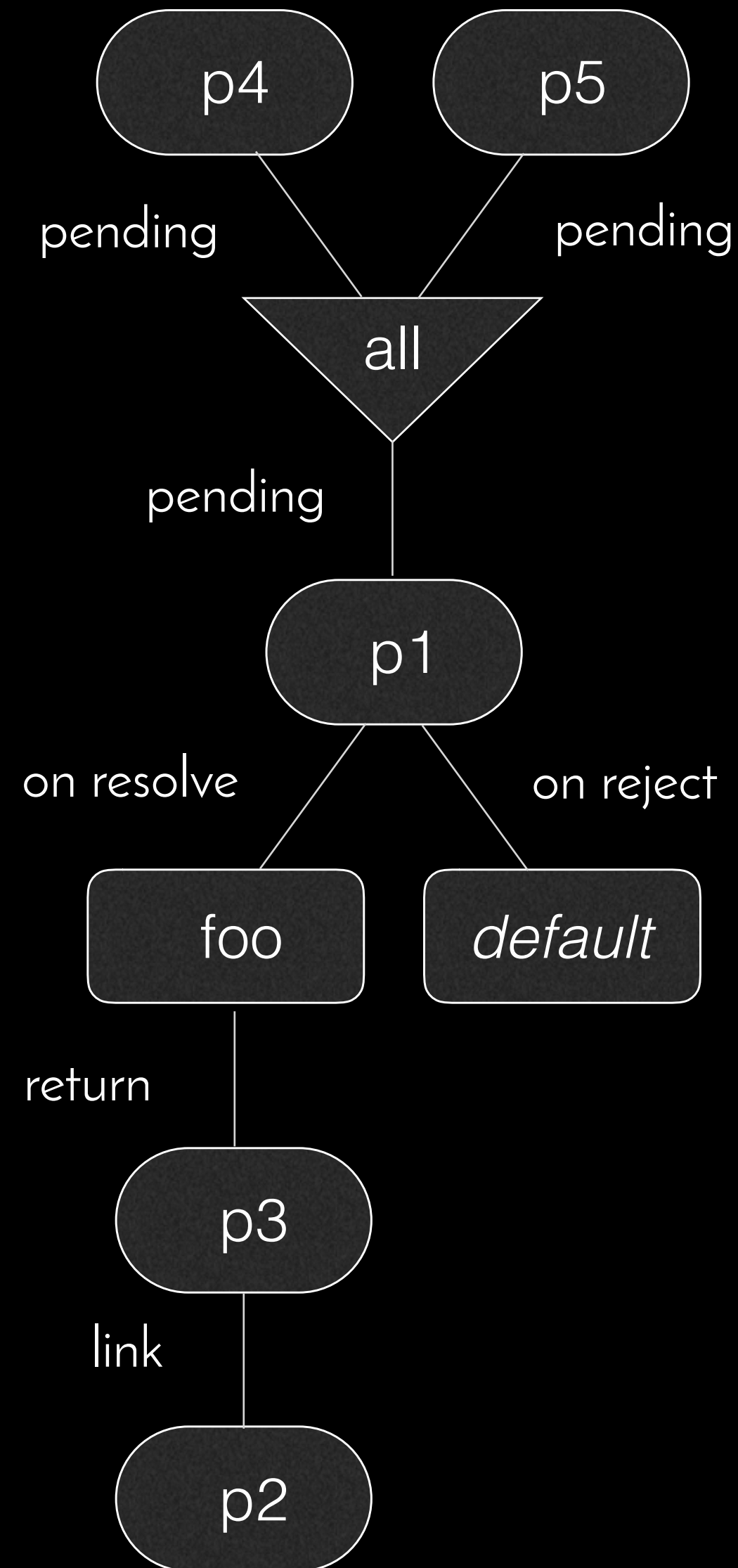
web app

1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs

4. Analyze anti-patterns



PromiseKeeper

web app

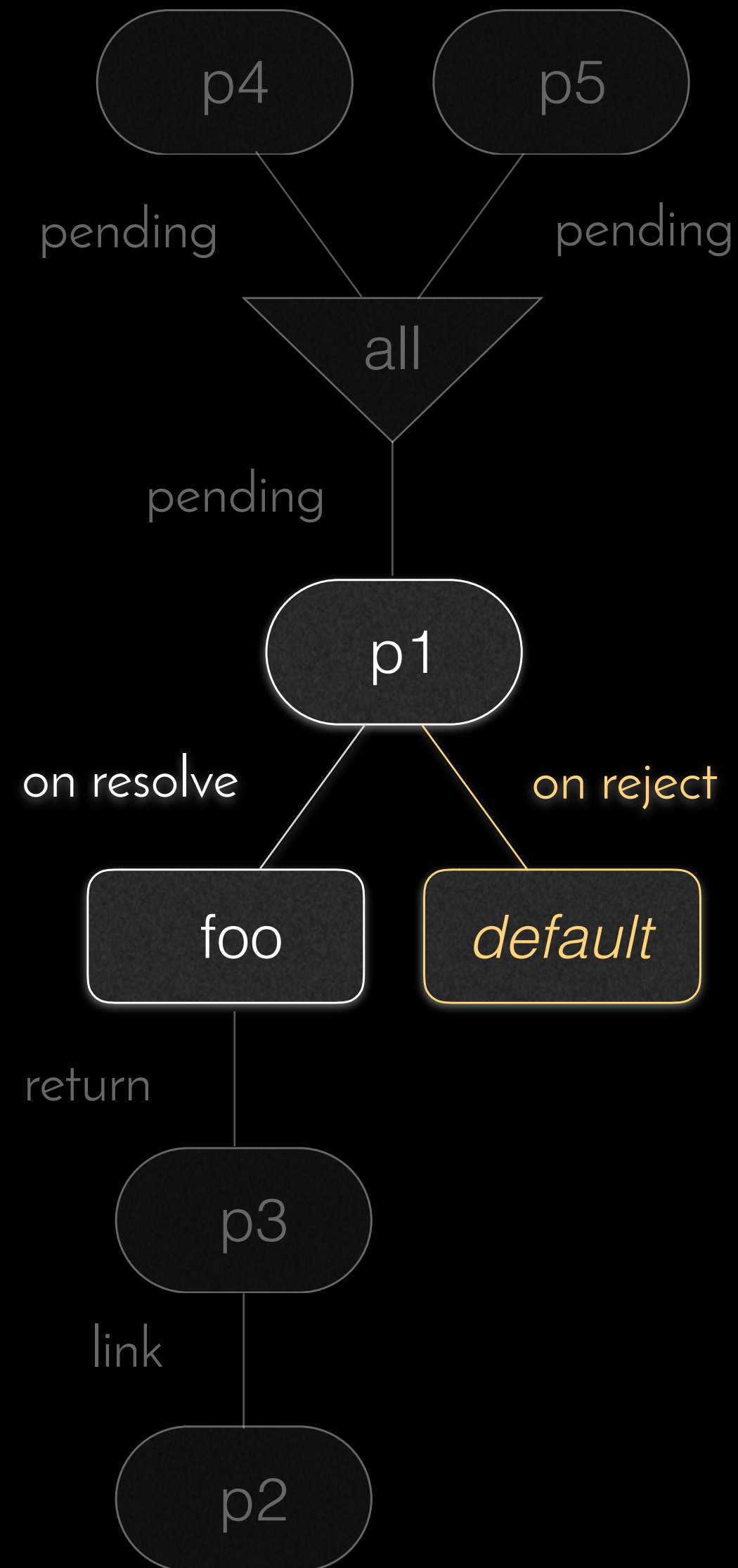
1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs

4. Analyze anti-patterns

unhandled promise rejections



PromiseKeeper

web app

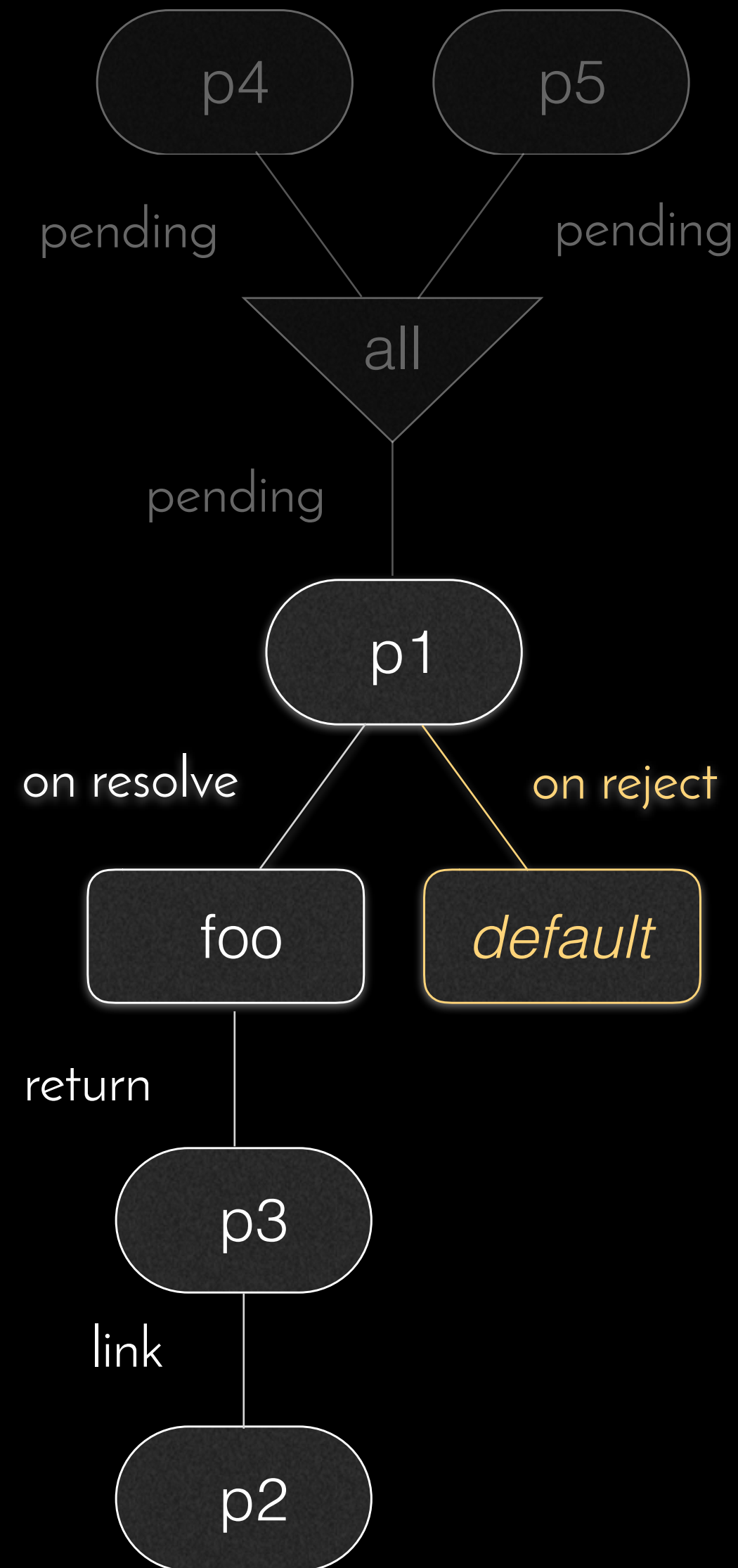
1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs

4. Analyze anti-patterns

unhandled promise rejections



PromiseKeeper

web app

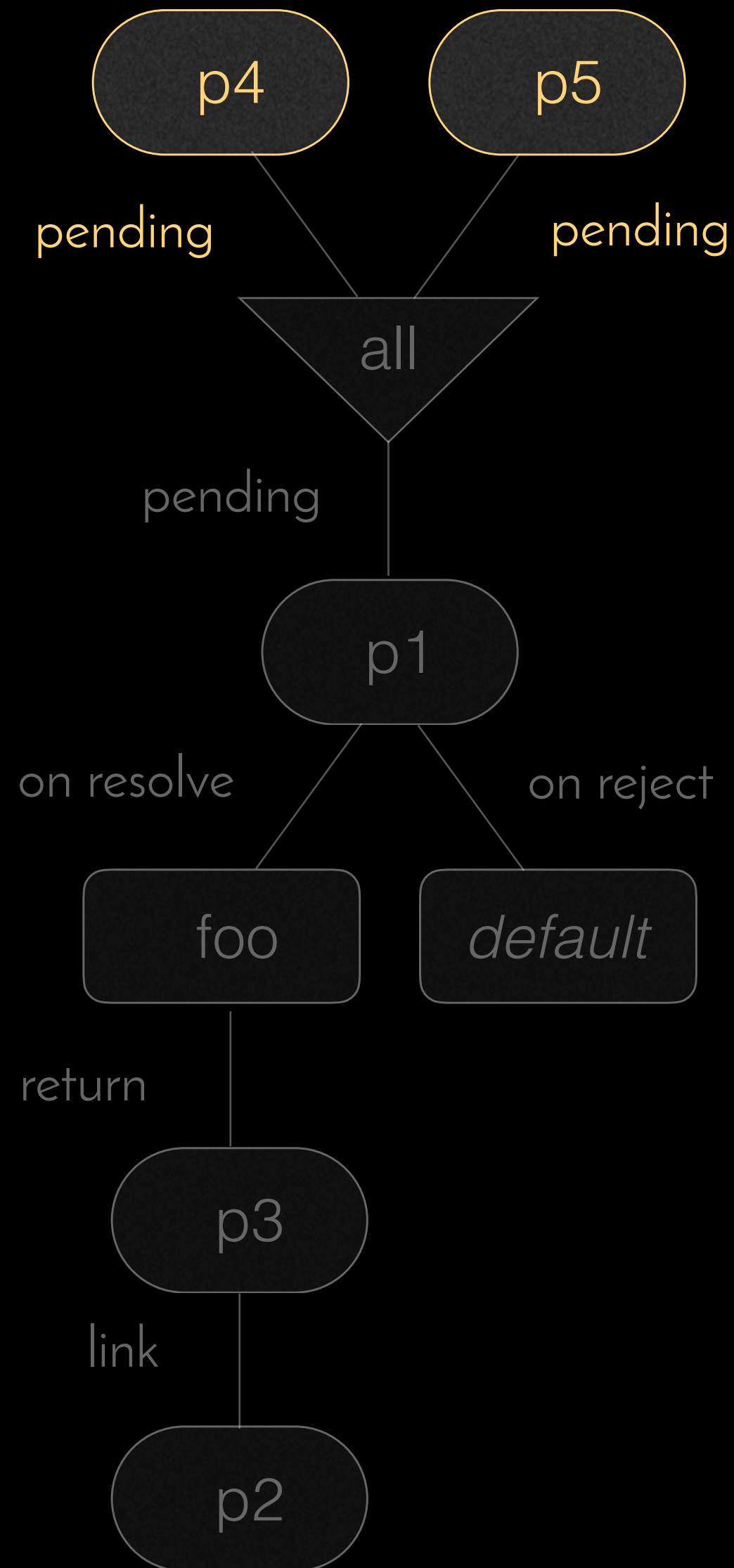
1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs

4. Analyze anti-patterns

unsettled promises



PromiseKeeper

web app

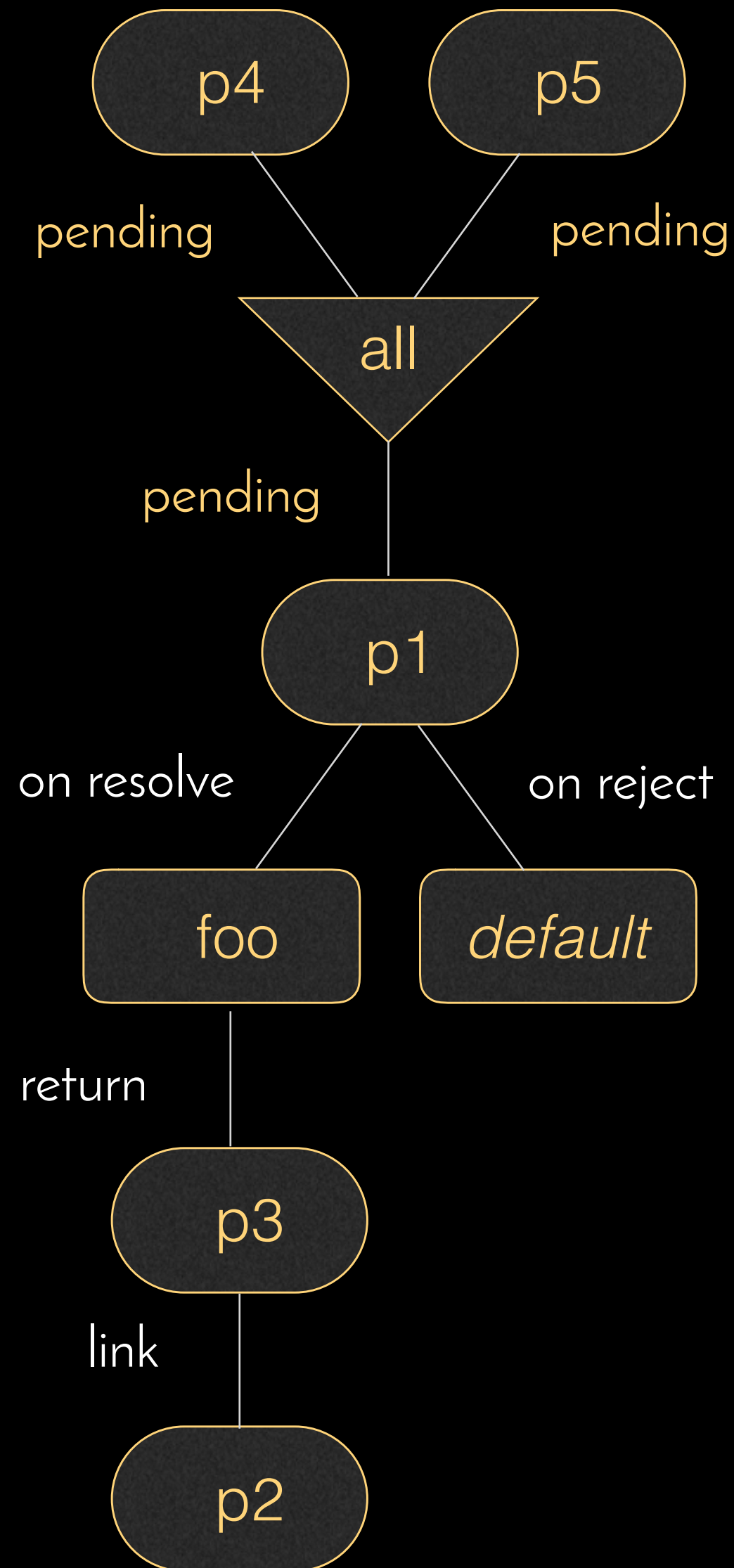
1. Instrument automatically

2. Collect execution traces

3. Infer promise graphs

4. Analyze anti-patterns

unsettled promises



PromiseKeeper

web app



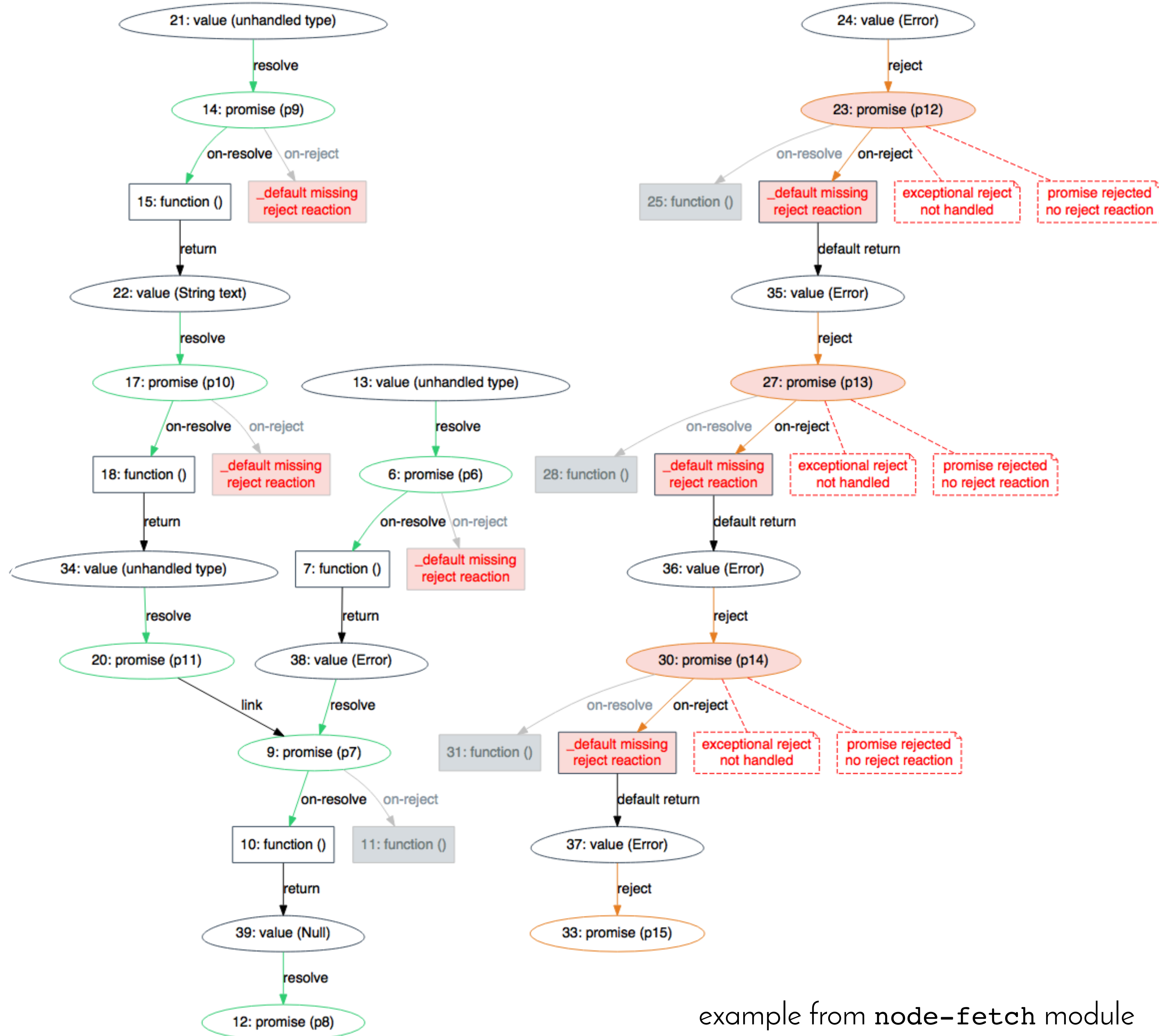
1. Instrument automatically

2. Collect execution traces

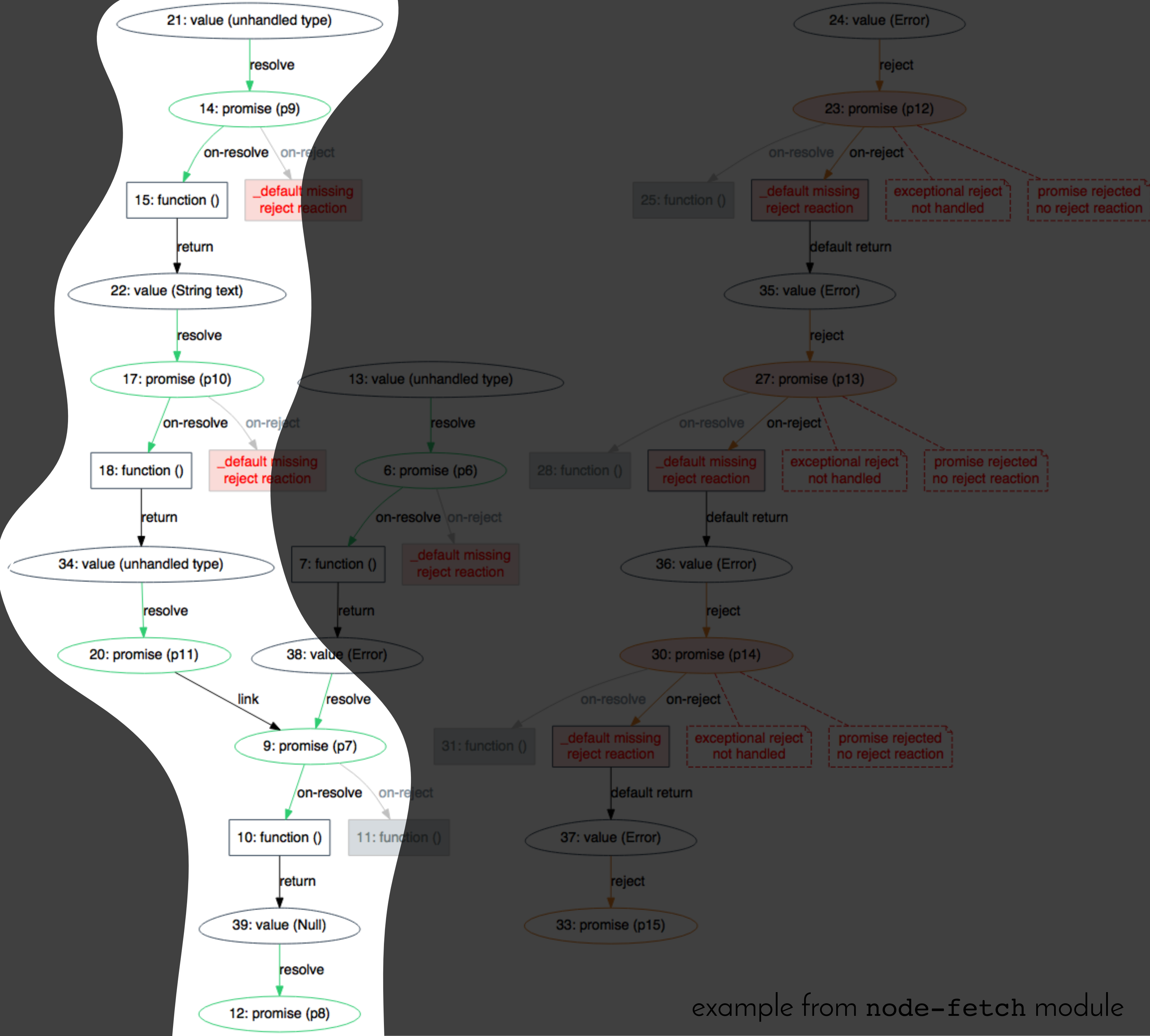
3. Infer promise graphs

4. Analyze anti-patterns

5. Visualize the graphs
and generate reports



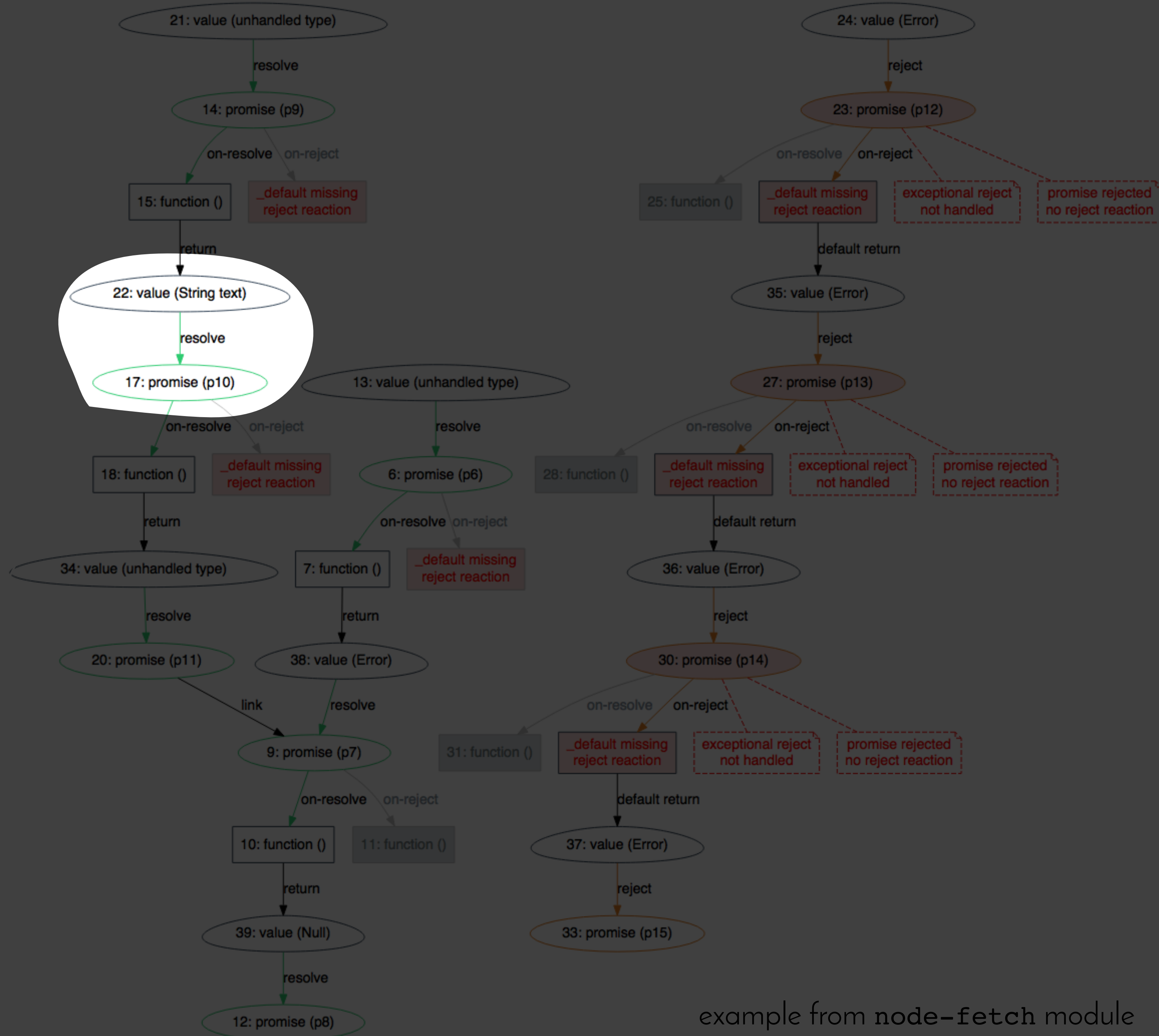
example from node-fetch module



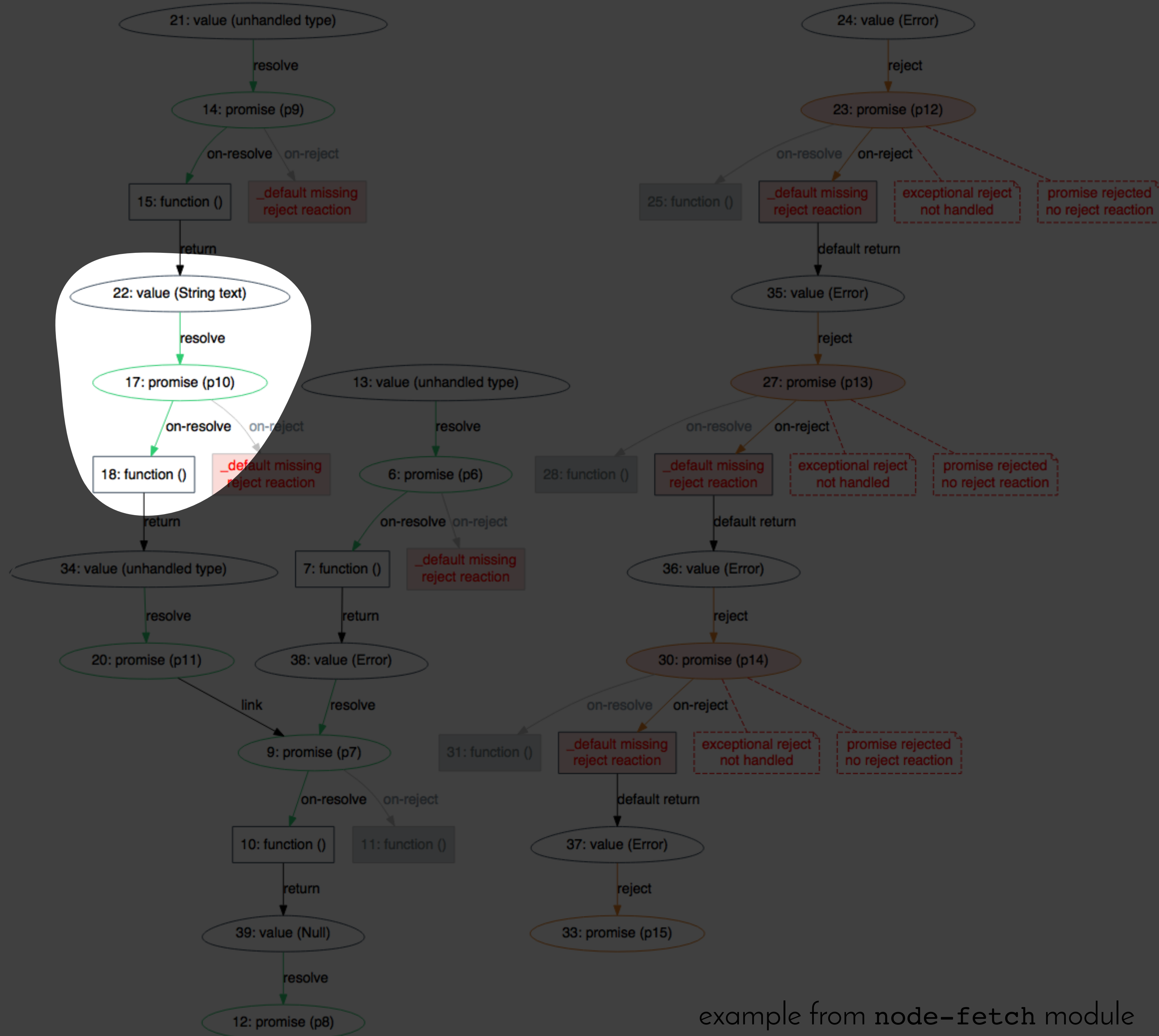
example from node-fetch module



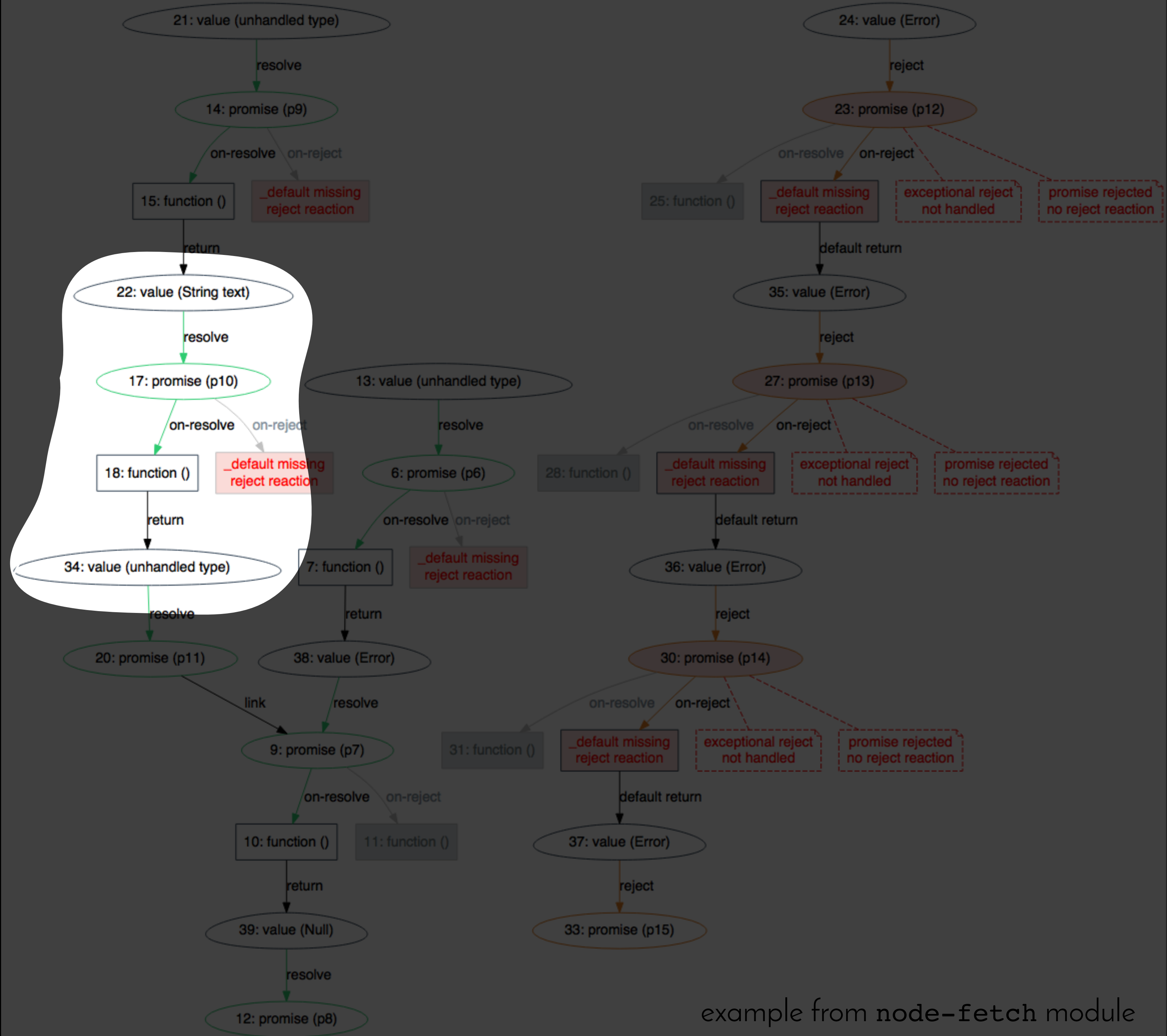
example from node-fetch module



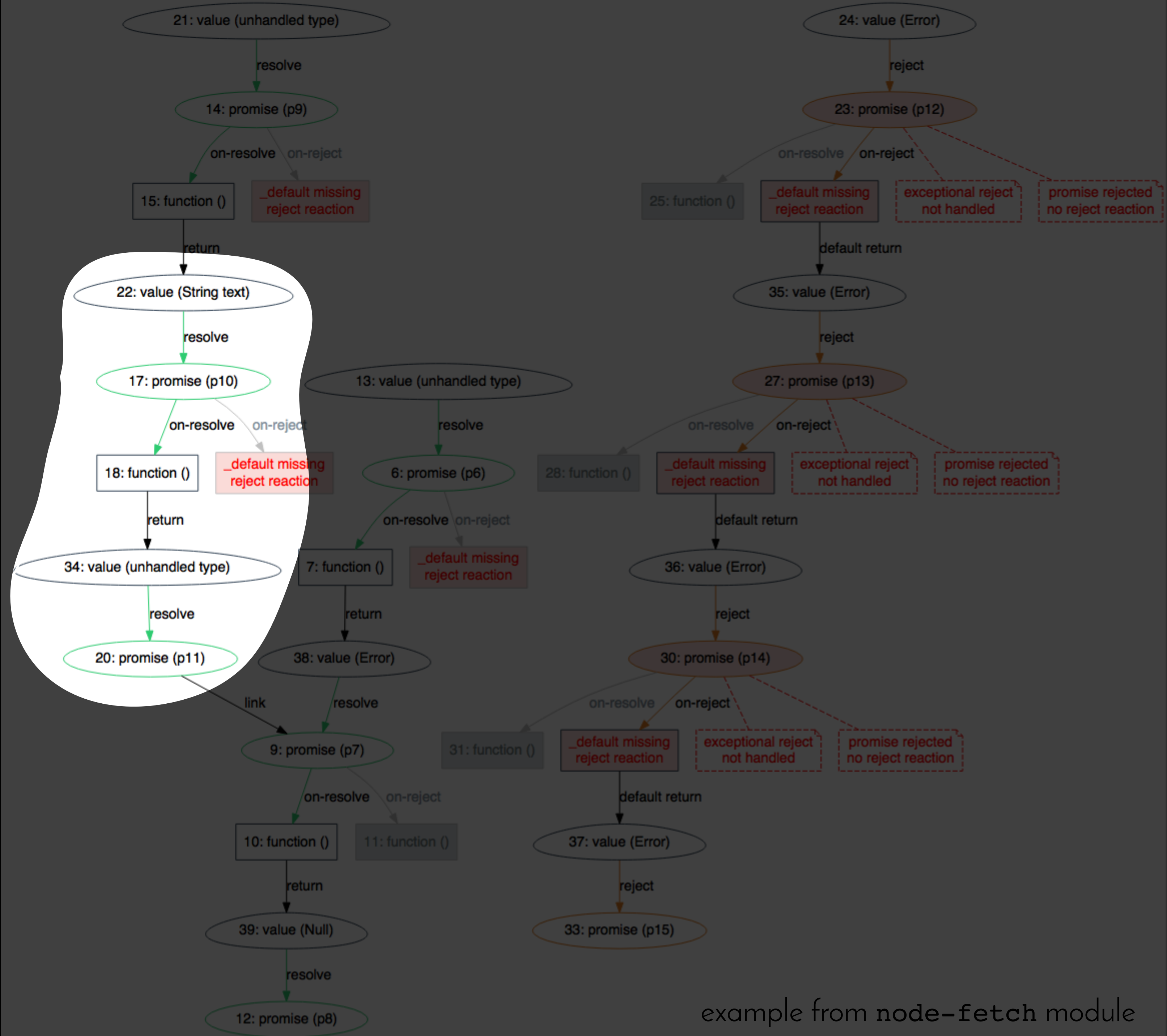
example from node-fetch module



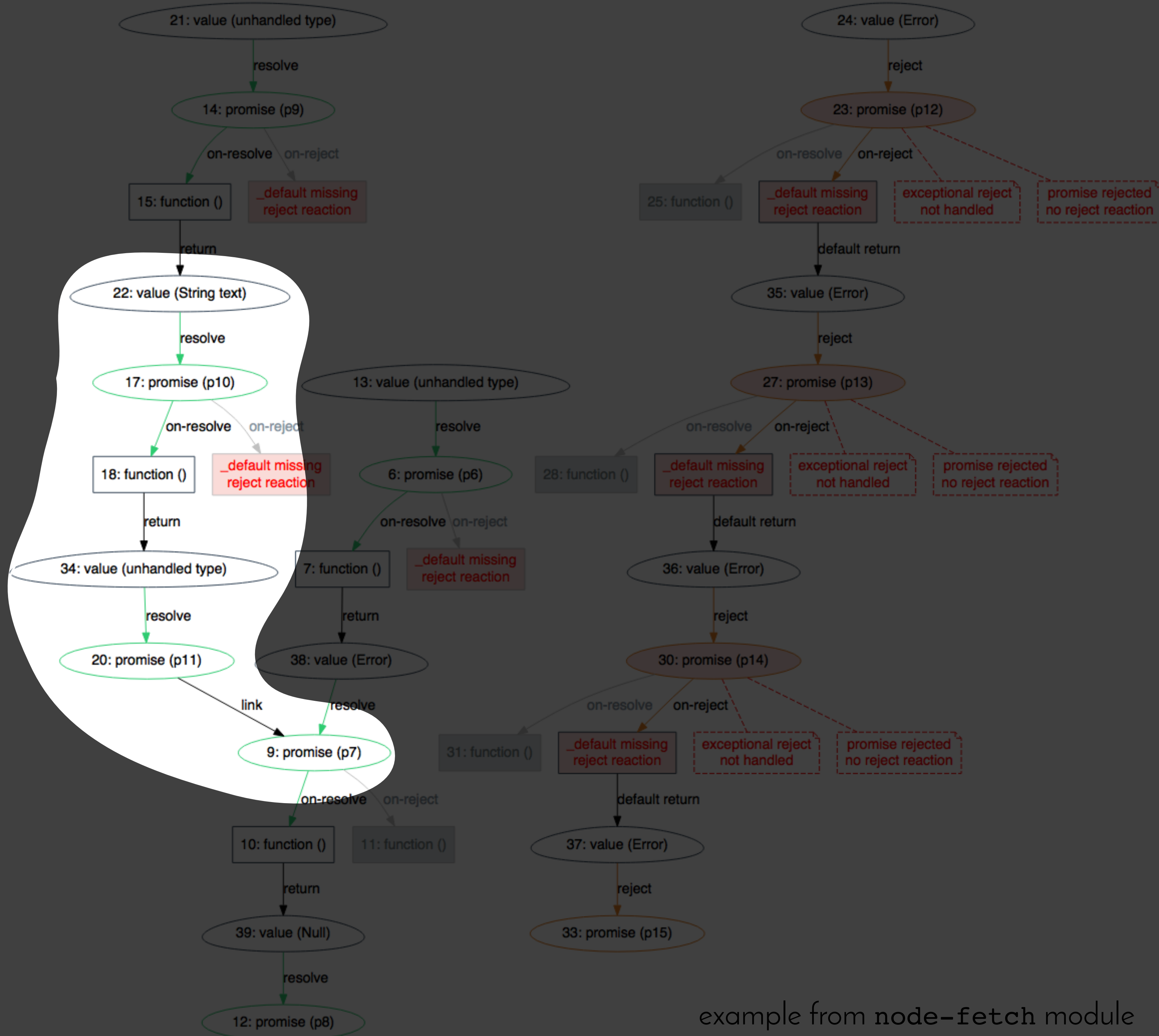
example from node-fetch module



example from node-fetch module



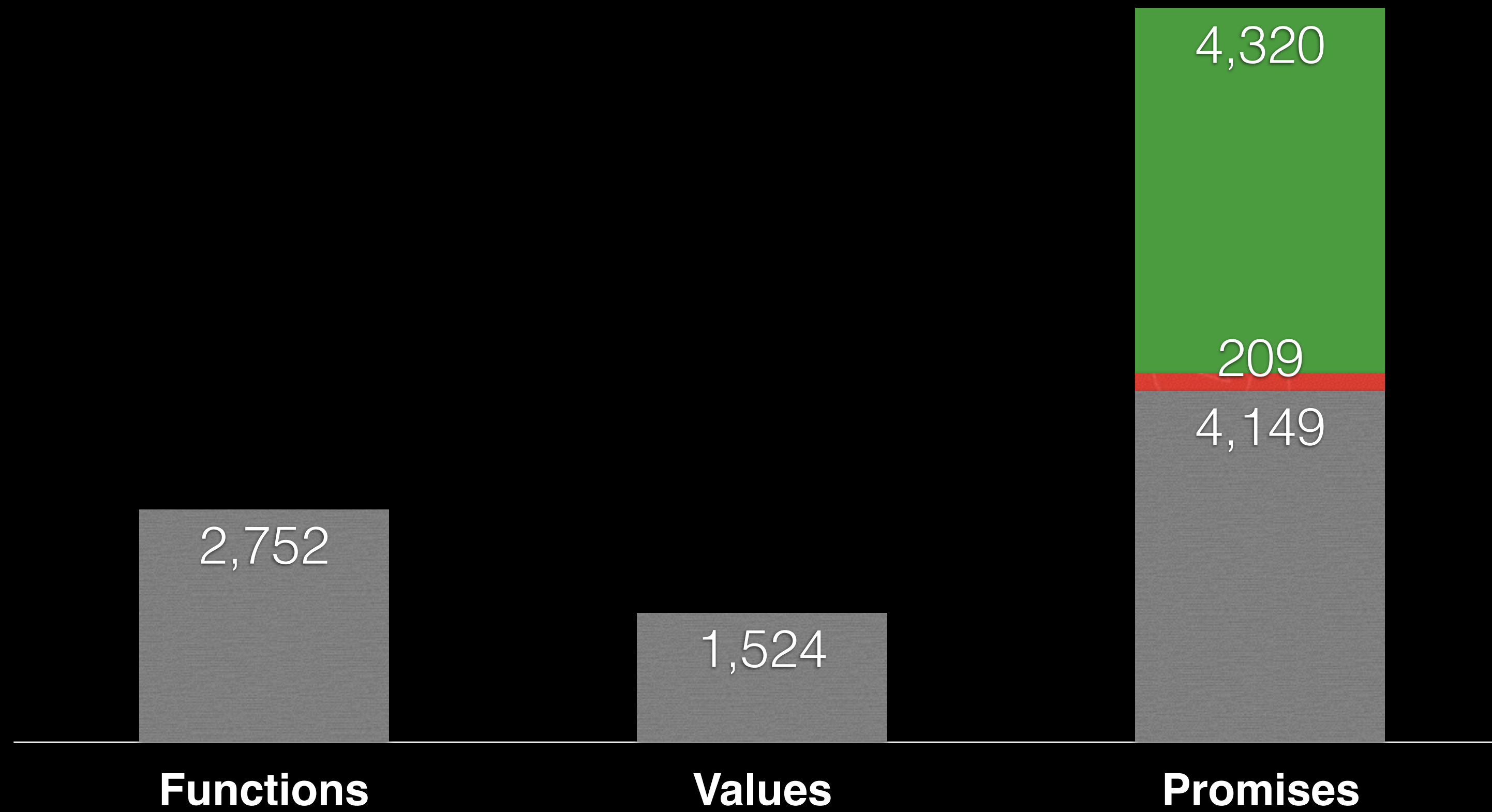
example from node-fetch module



example from node-fetch module

Broken Promises





115,000
Lines of



209

209
rejected promises

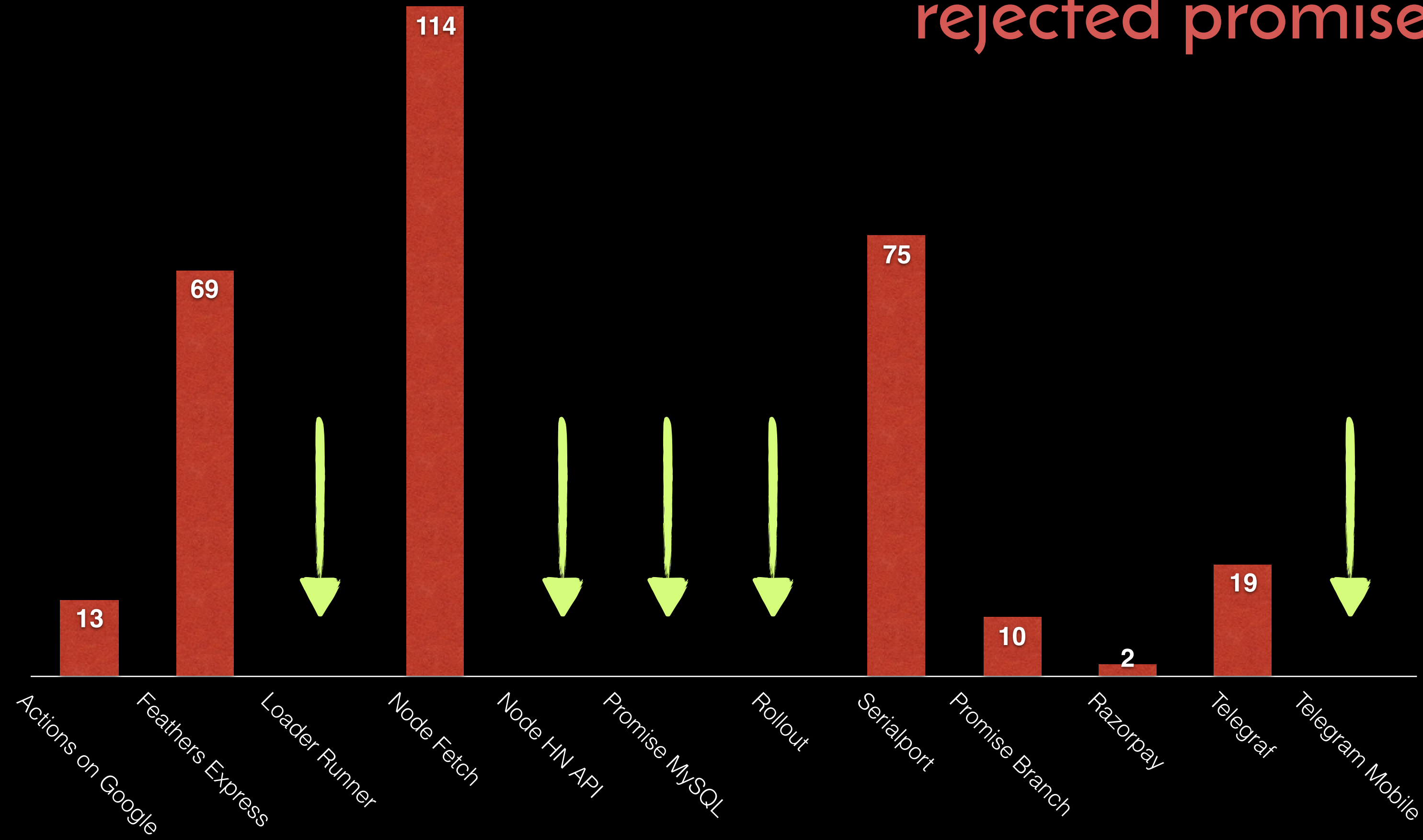
209

rejected promises

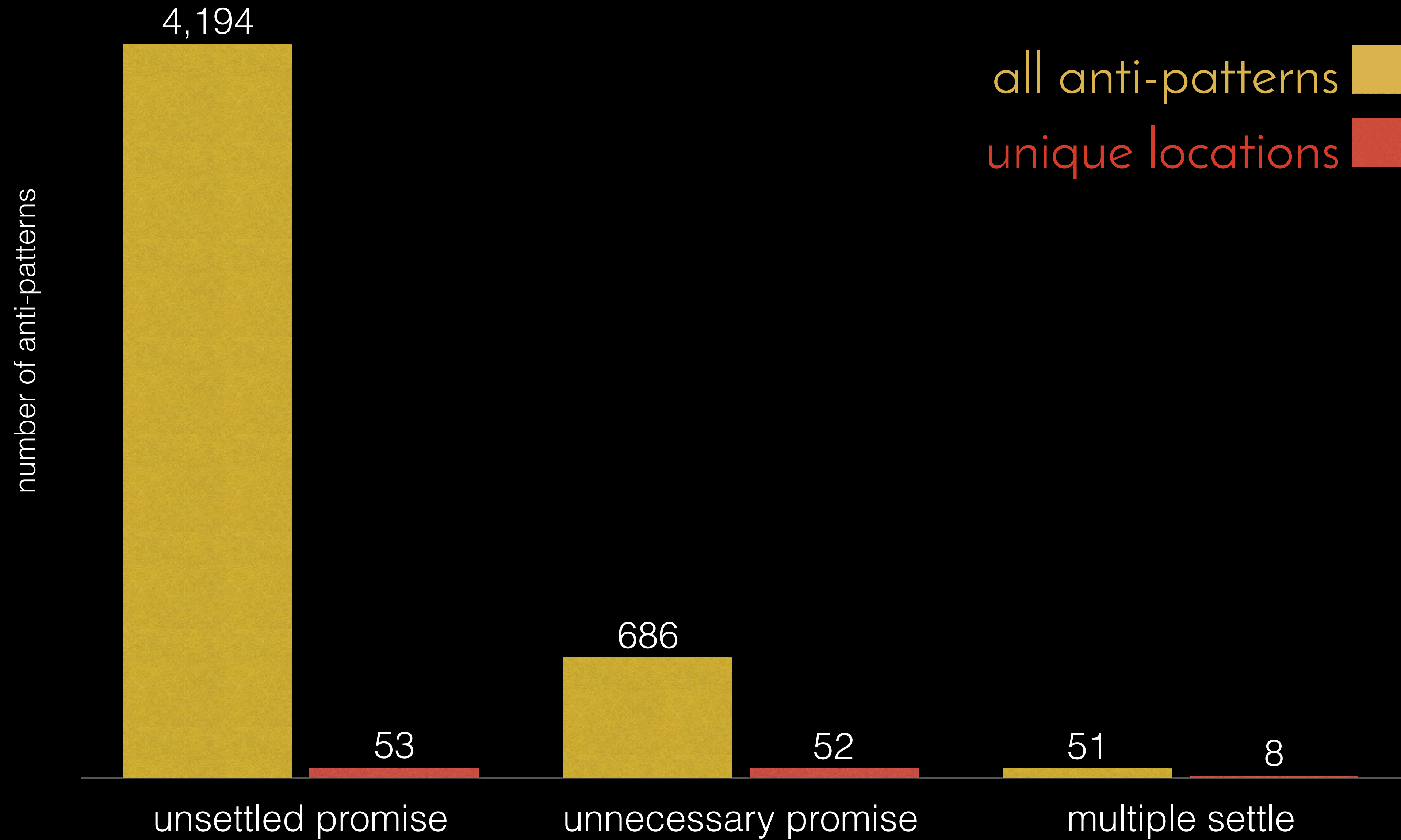
total
8698

rejected promises

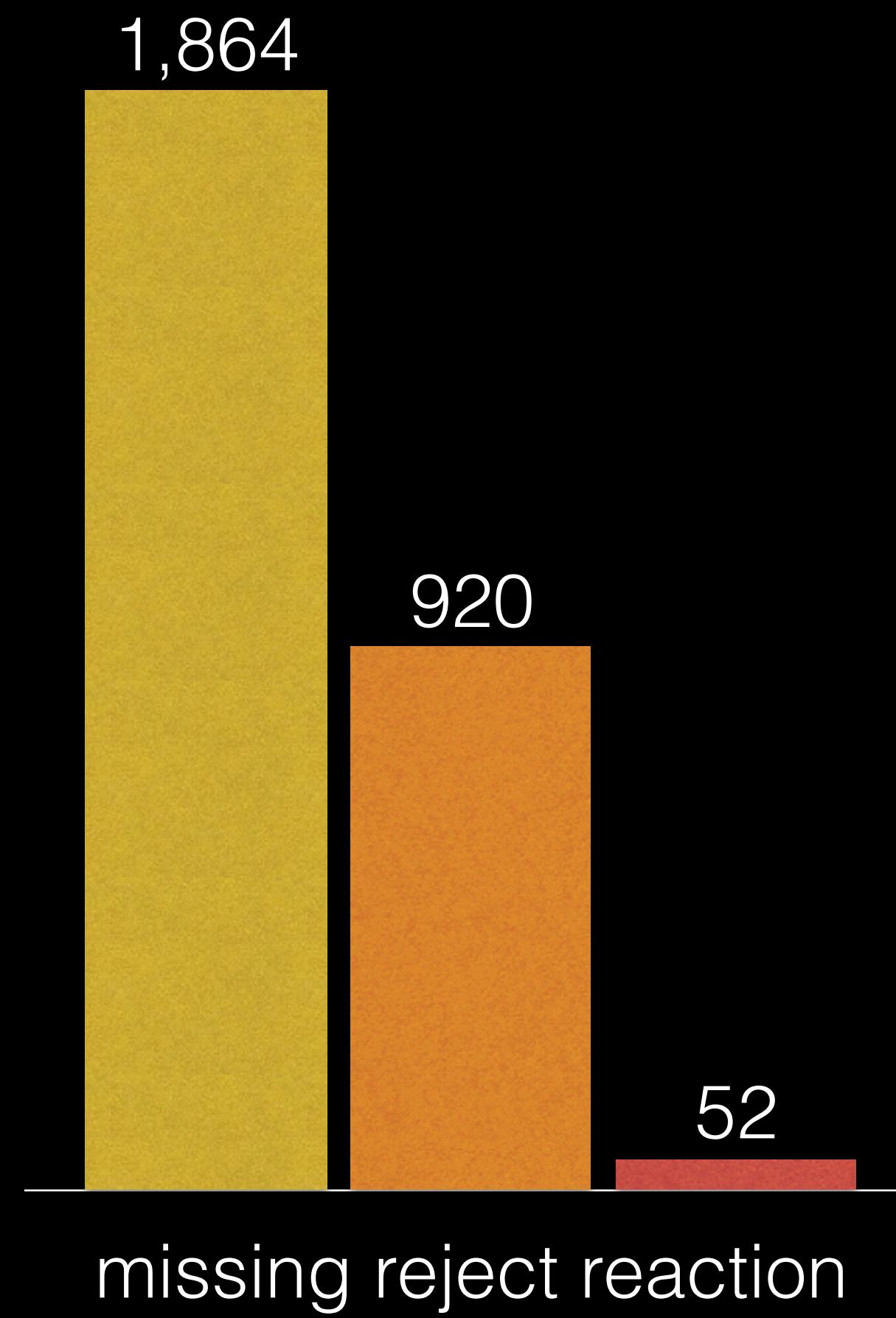
number of rejected promises

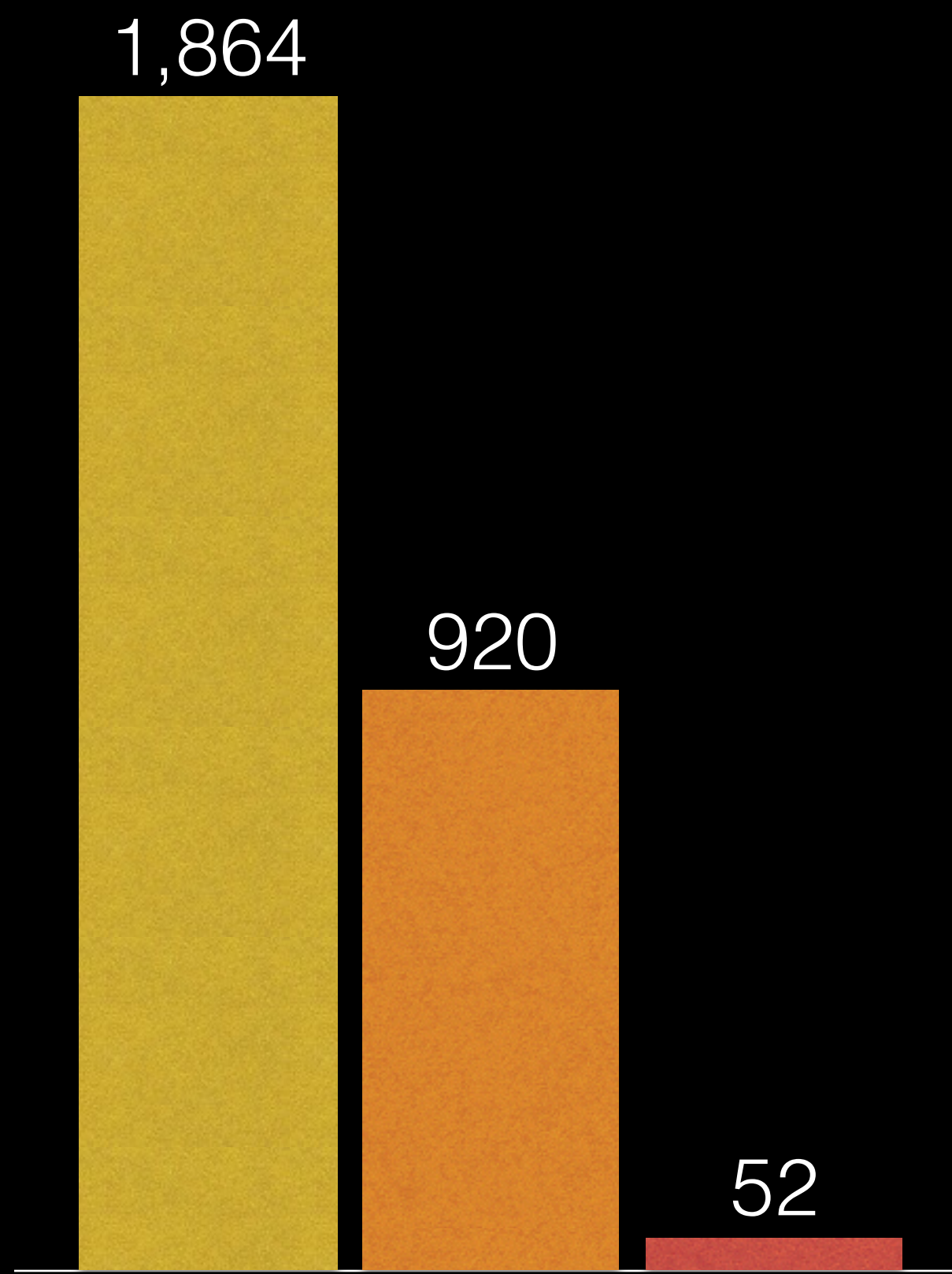


subject applications

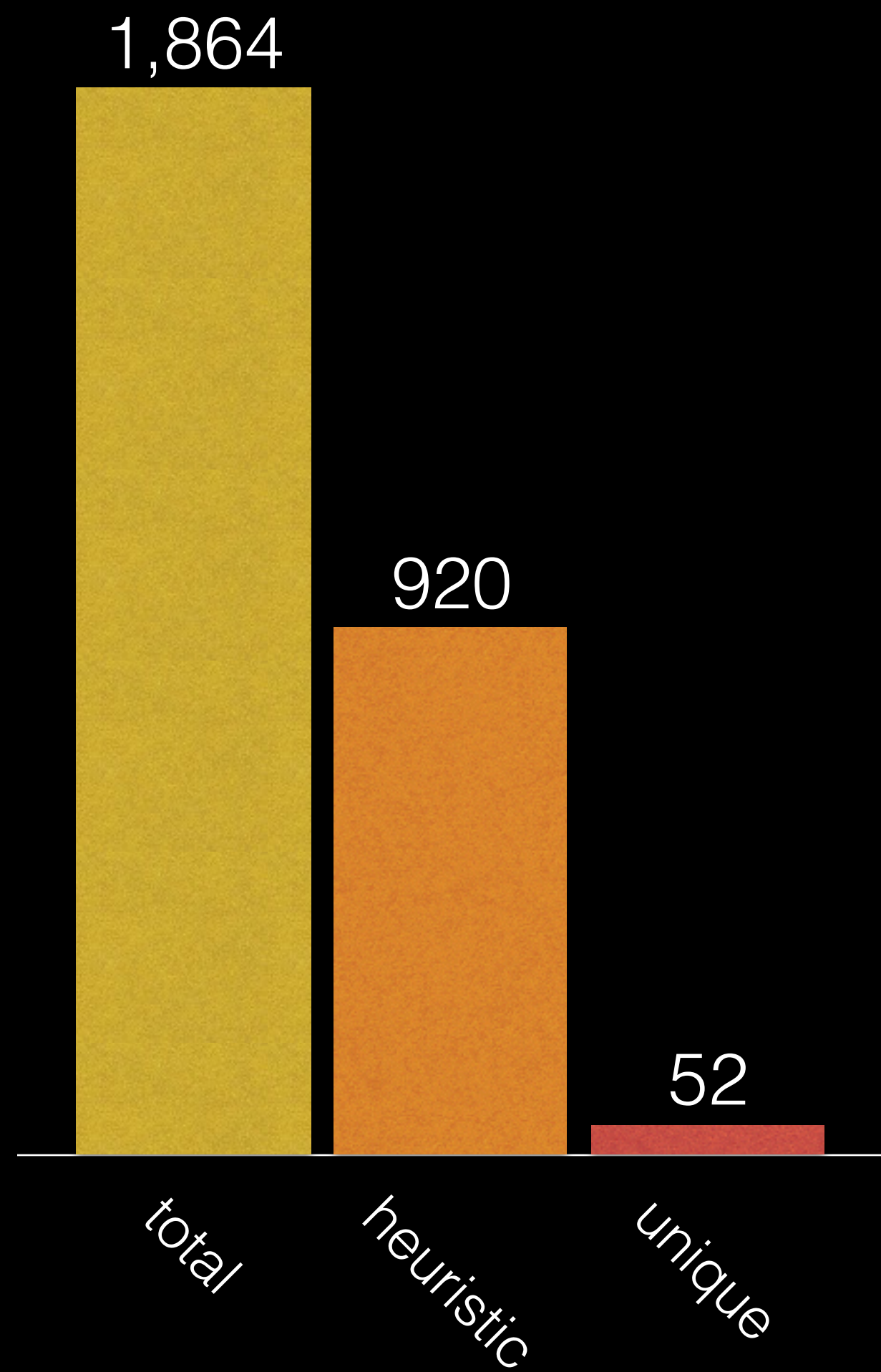




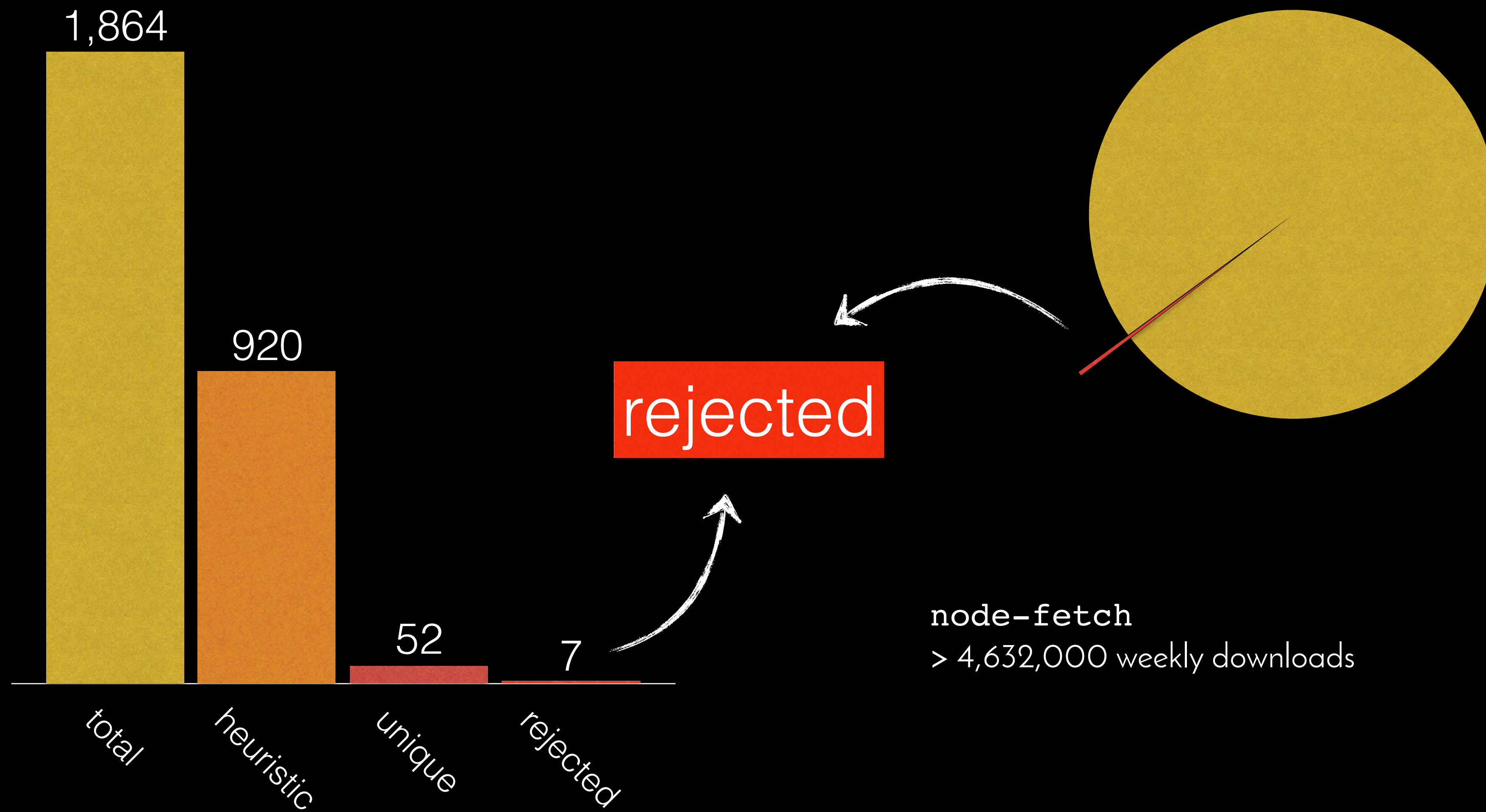


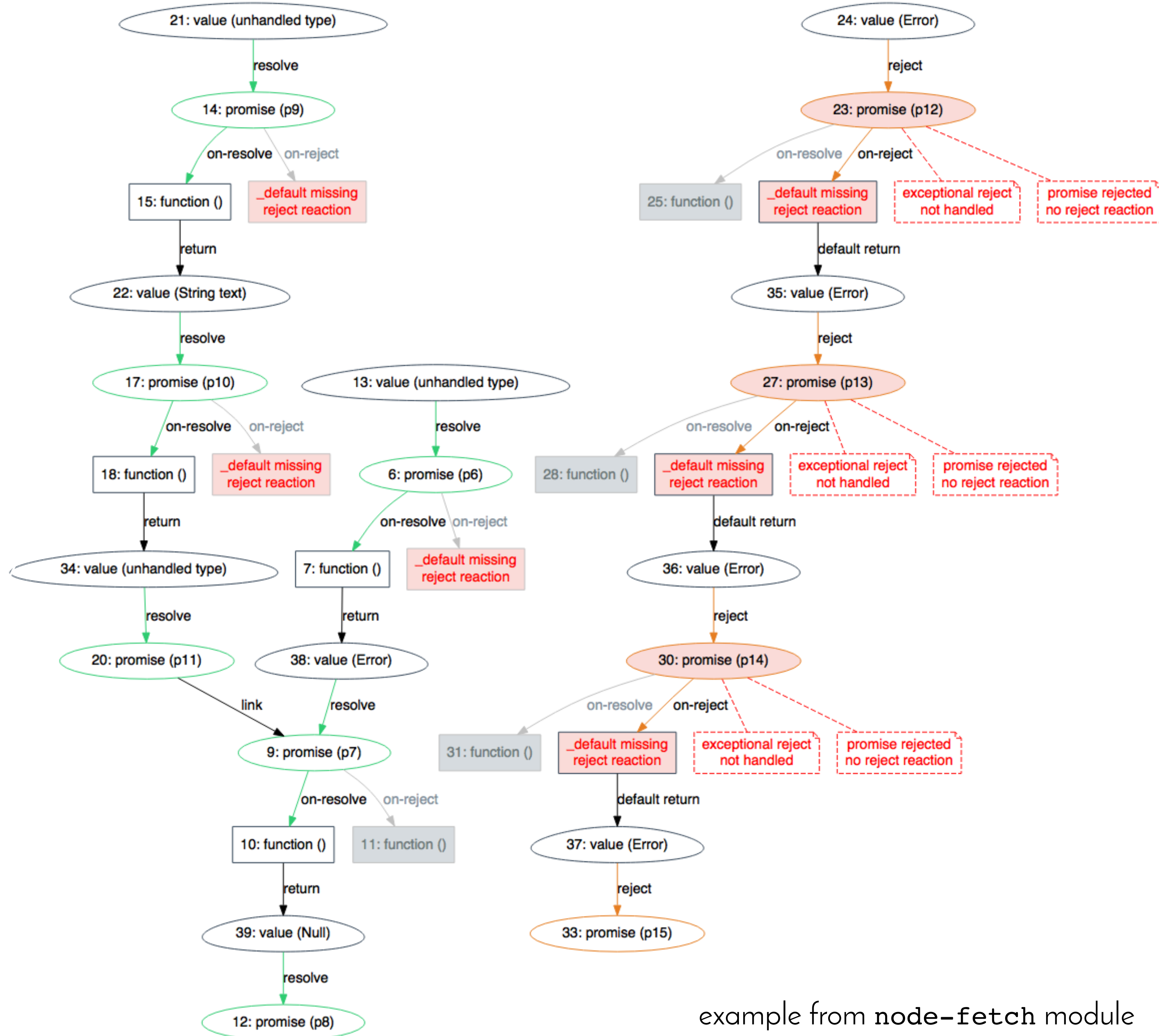


missing reject reactions



missing reject reactions

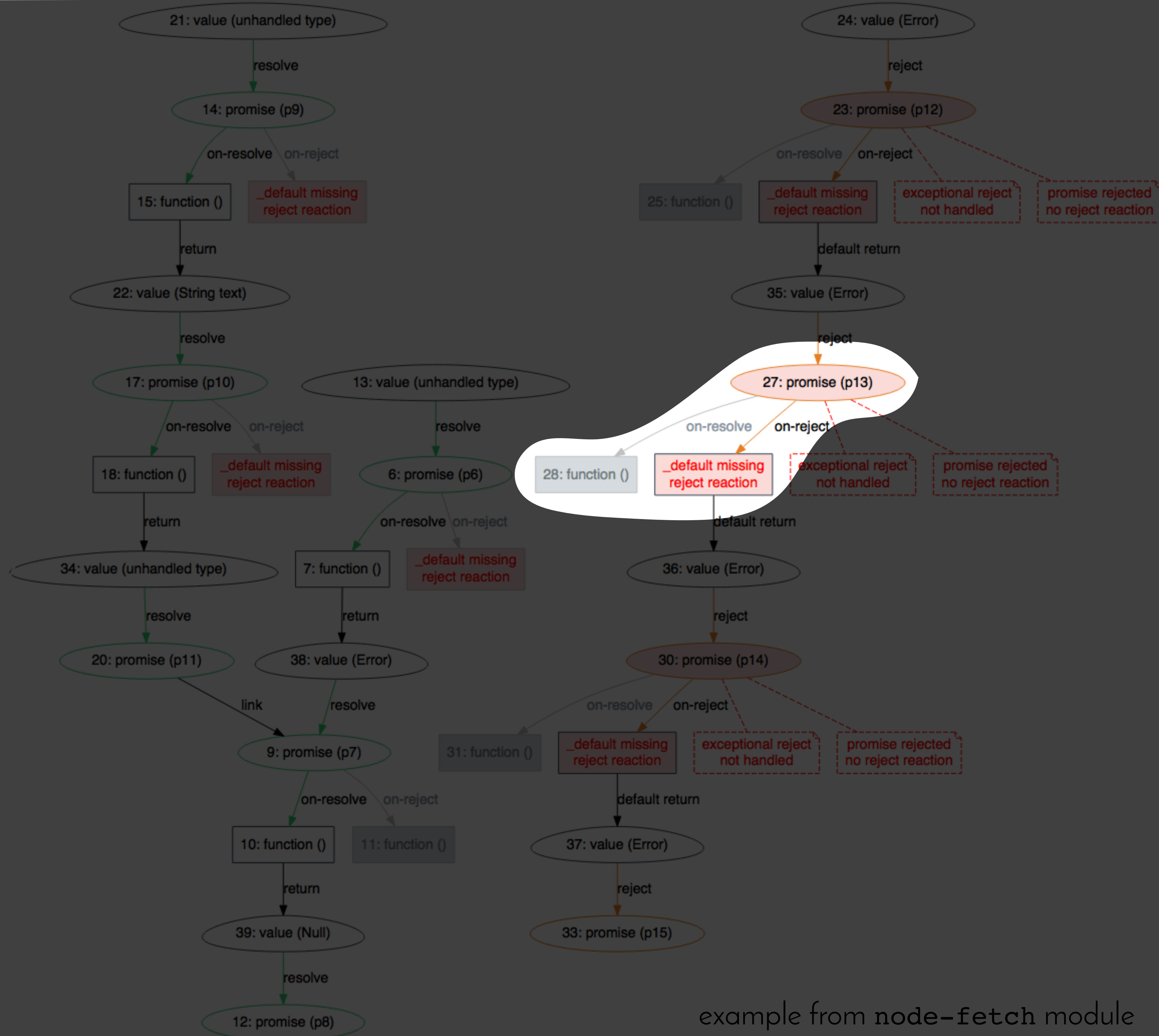




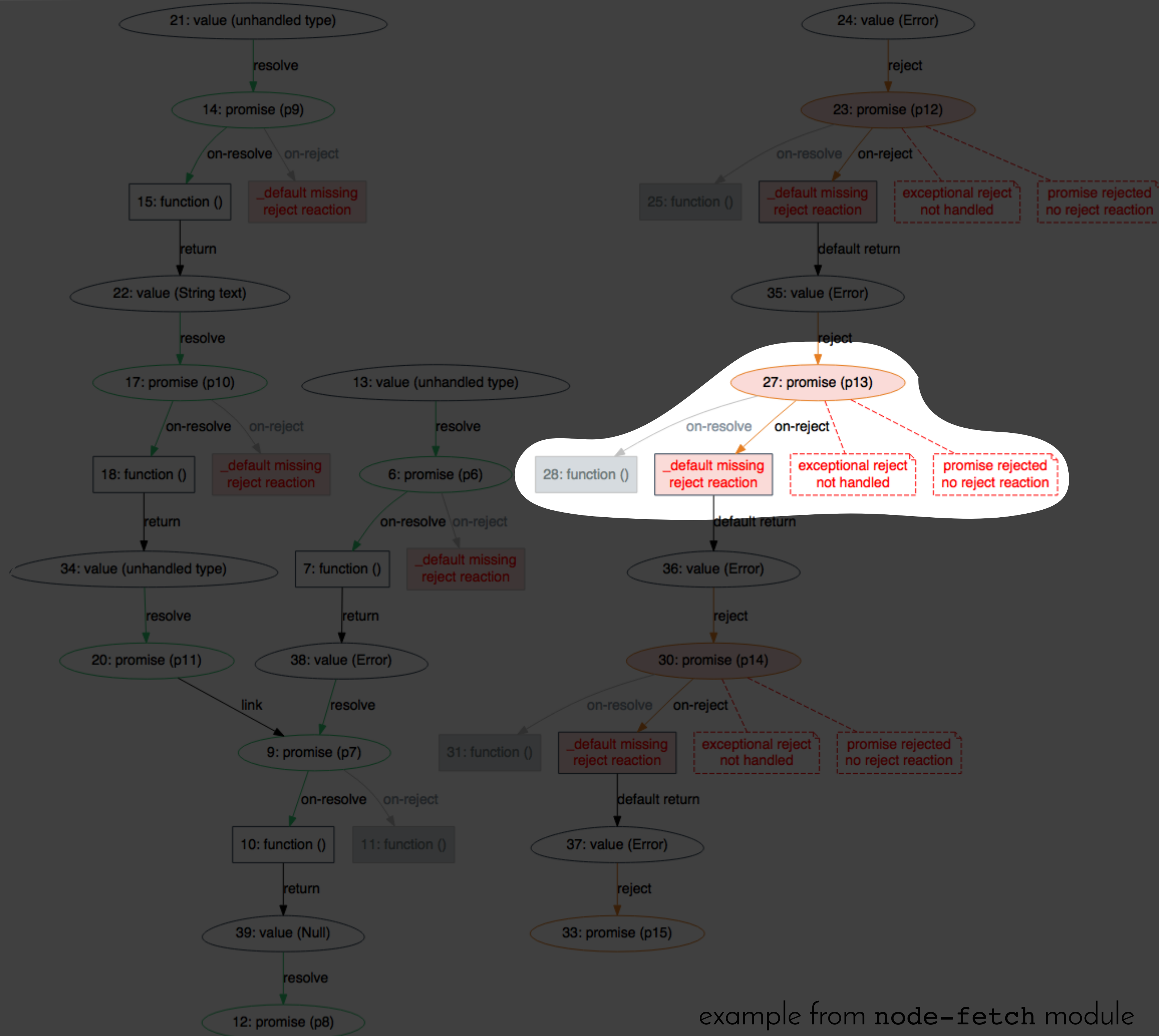
example from node-fetch module



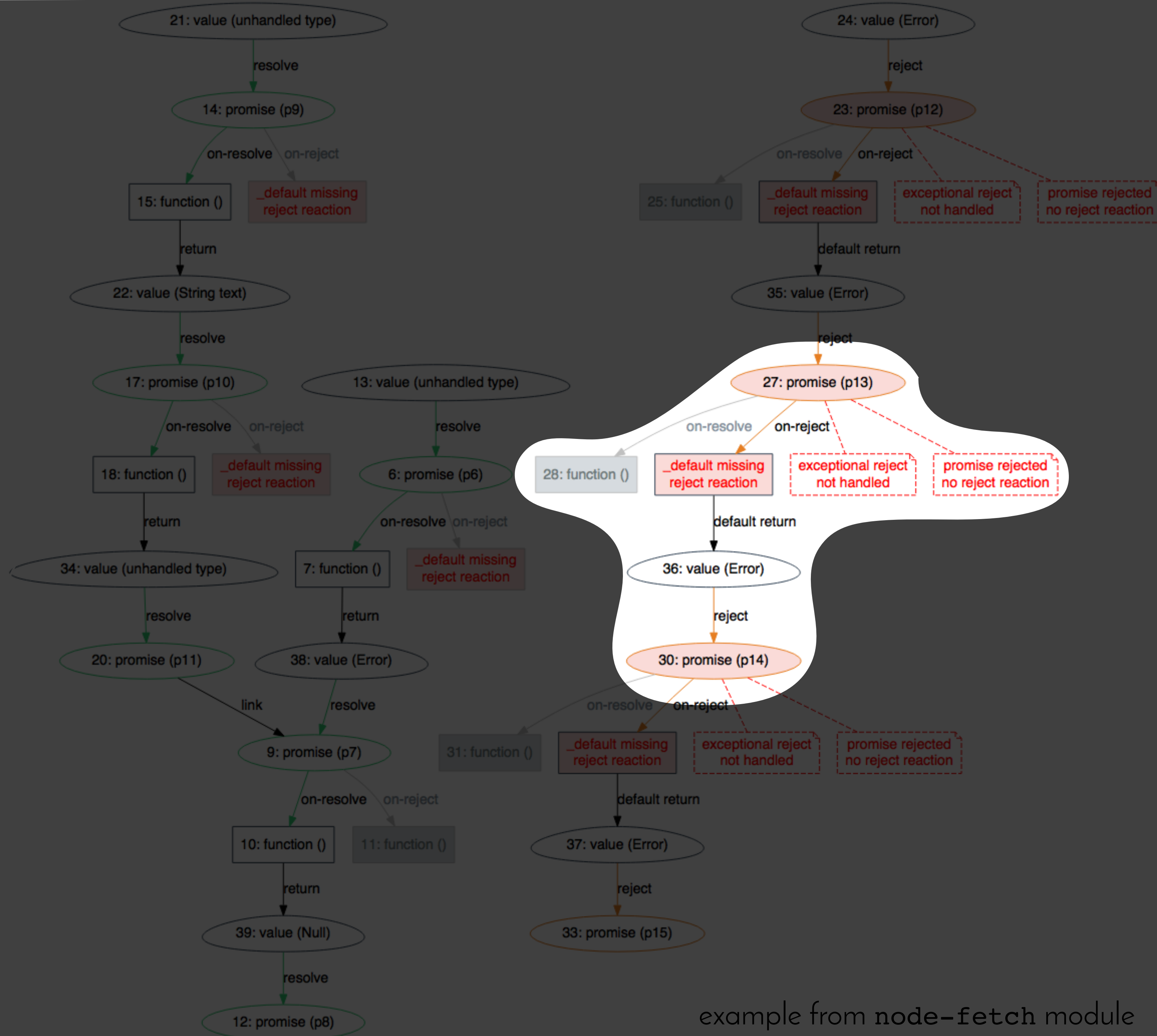
example from node-fetch module



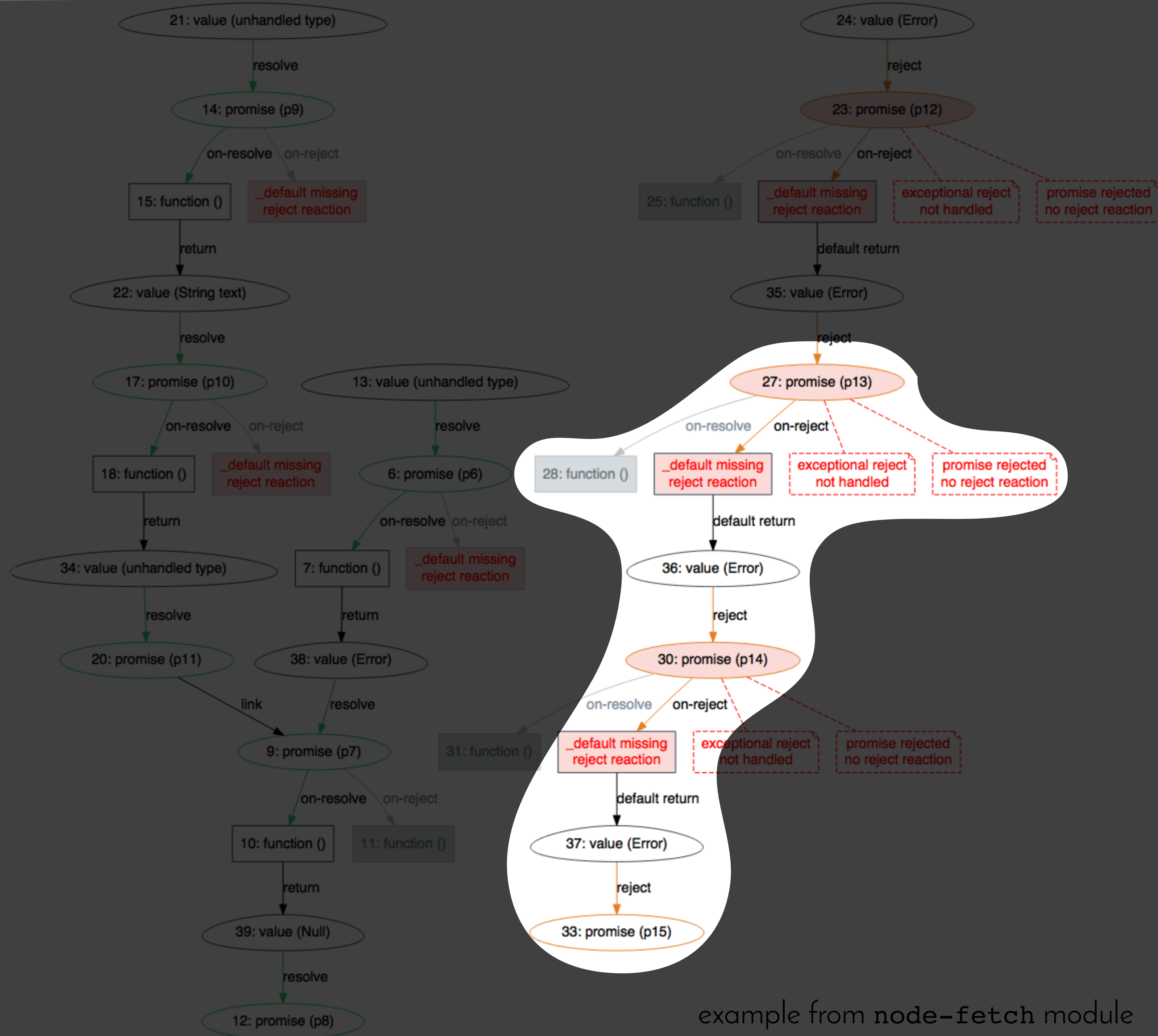
example from node-fetch module



example from node-fetch module

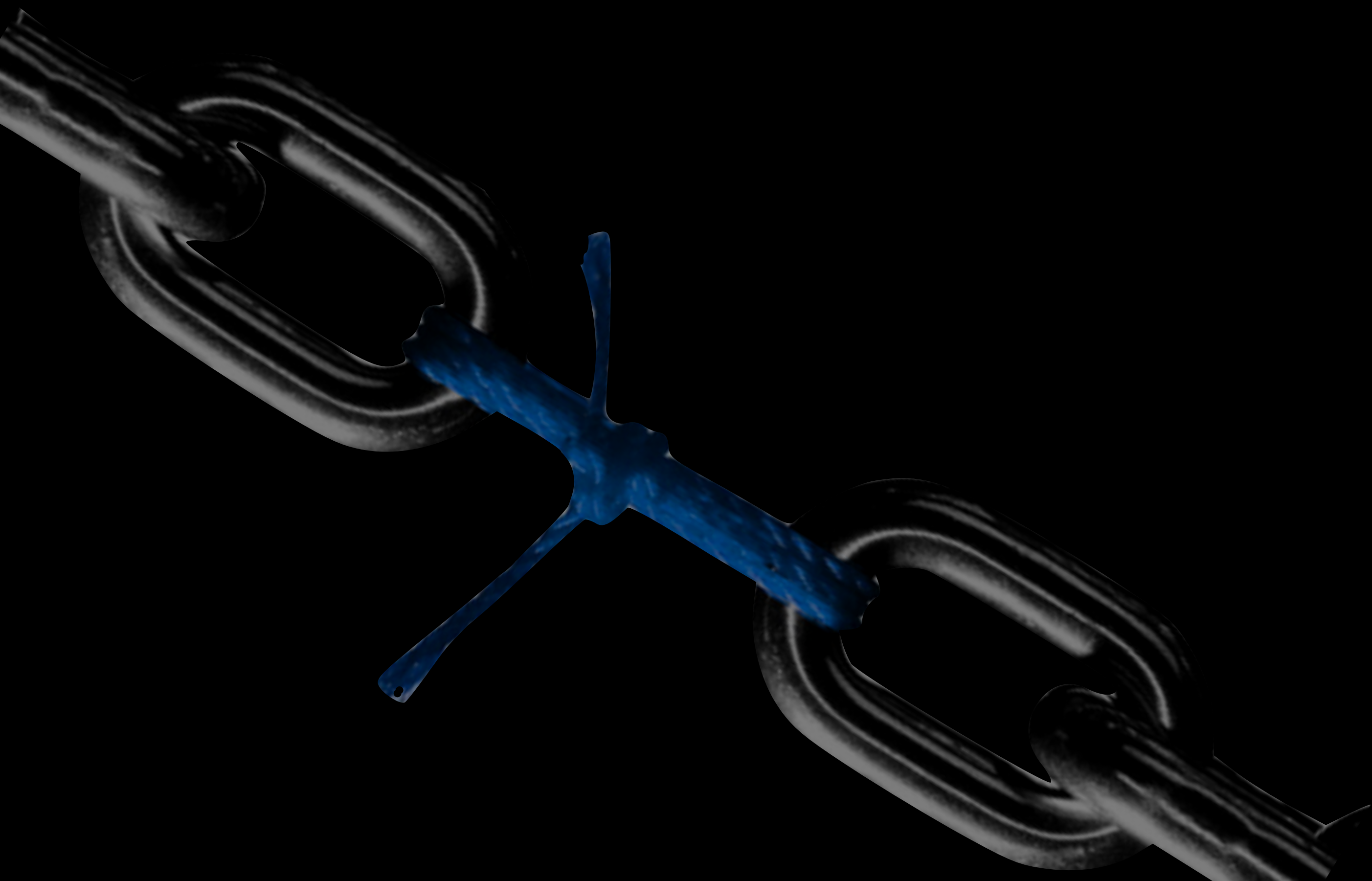


example from node-fetch module



example from node-fetch module

Common Developer Practices



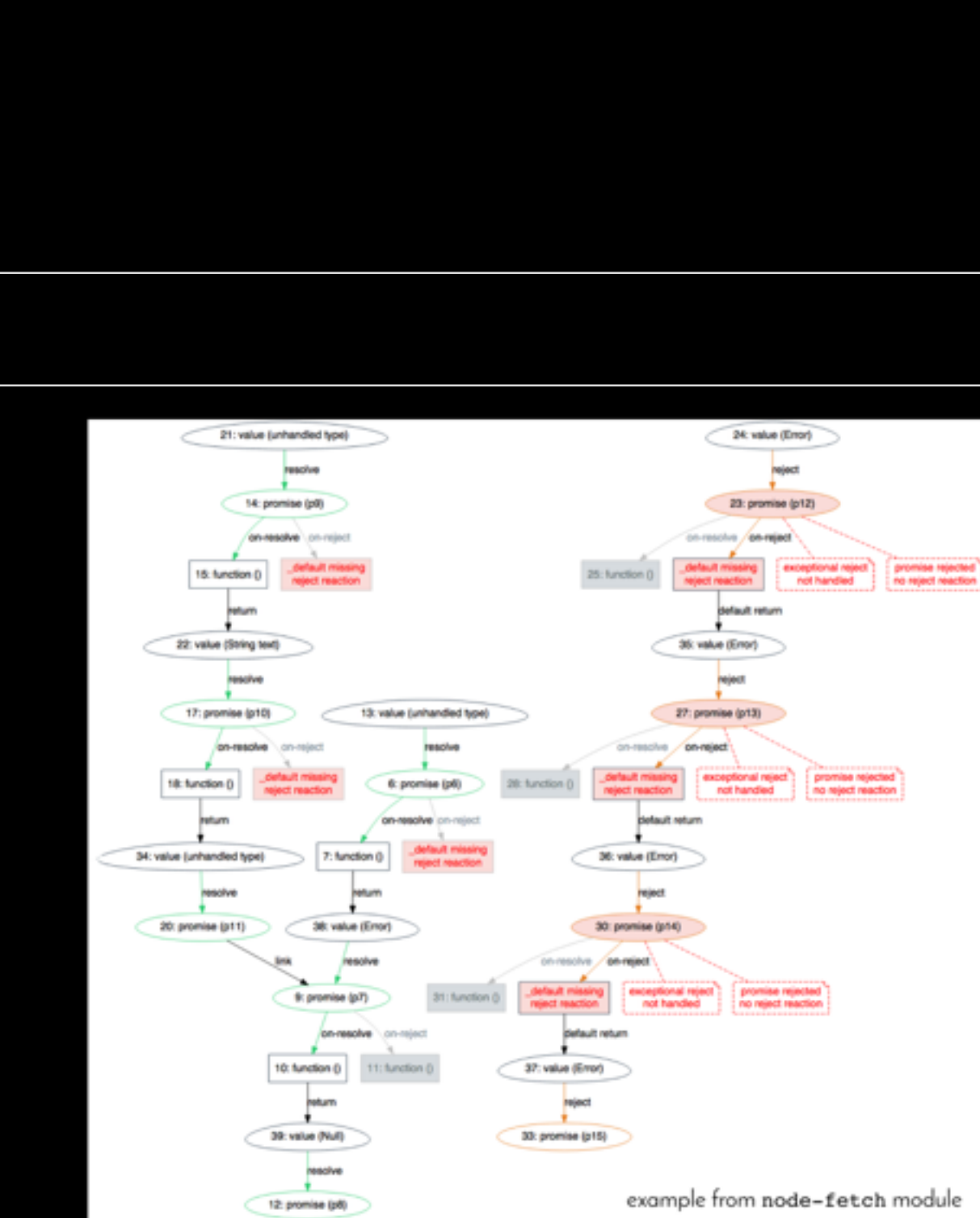
JavaScript Promises

```

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

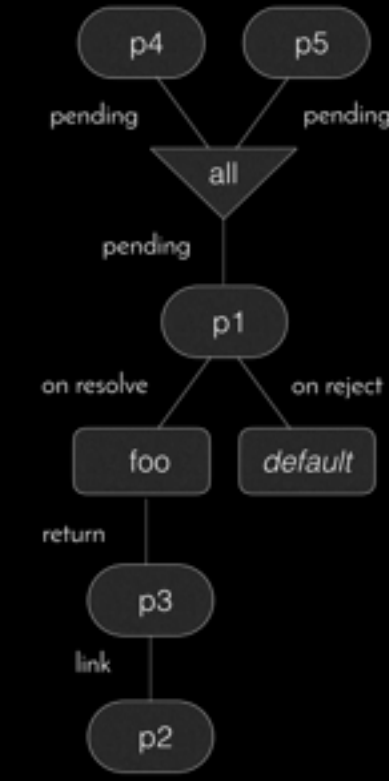
    P
    Fulfilled
    let p2 = p.then(handleSuccess)
                .then(soMuchSuccess, handleRejection)
                .catch(moreRejection);

    P
    Rejected
  
```



PromiseKeeper

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



<https://github.com/nuprl/PromiseKeeper>

@saba_a