

zero-cost

async stack traces

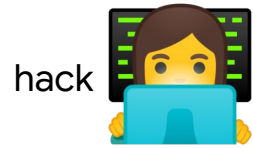


@bmeurer

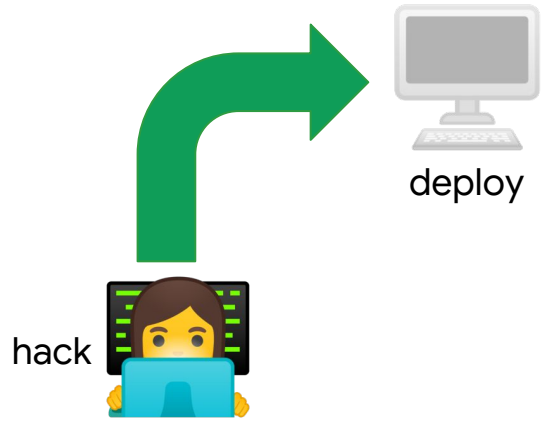


@v8js

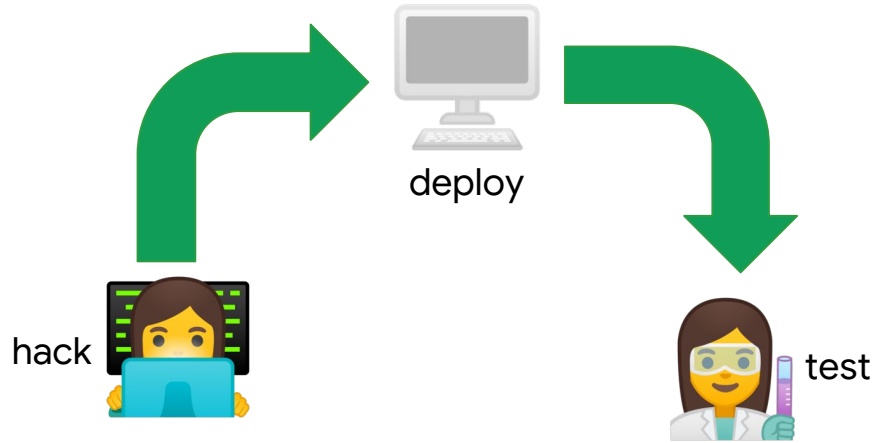
development cycle



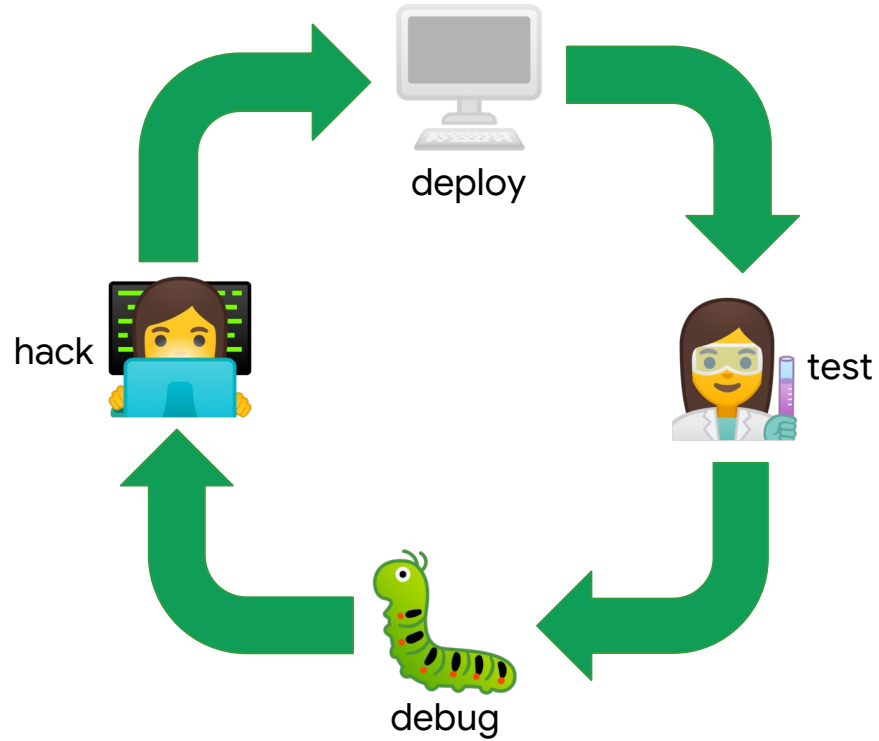
development cycle



development cycle



development cycle



async stack traces x

```
1 async function foo(x) {
2   await bar(x);
3 }
4
5 async function bar(x) { x = 1
6   await x;
7   throw new Error("Let's have a look...");
8 }
9
10 foo(1).catch(e => console.log(e.stack));
```

Line 7, Column 3

⌘+Enter

Pause on caught exceptions

⊗ **Paused on promise rejection**
Error: Let's have a look...

Threads

- Main paused
- newtab-serviceworker.js #103091 (activated)
- widget

Watch

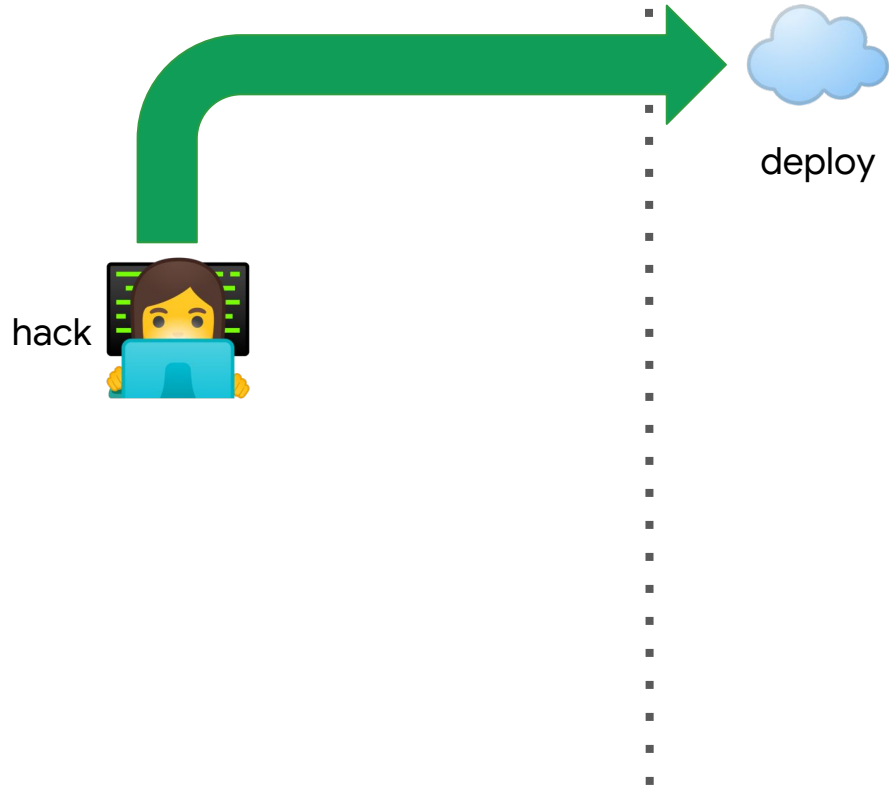
Call Stack

- bar async stack traces:7
- await in bar (async)
- foo async stack traces:2
- (anonymous) async stack traces:10

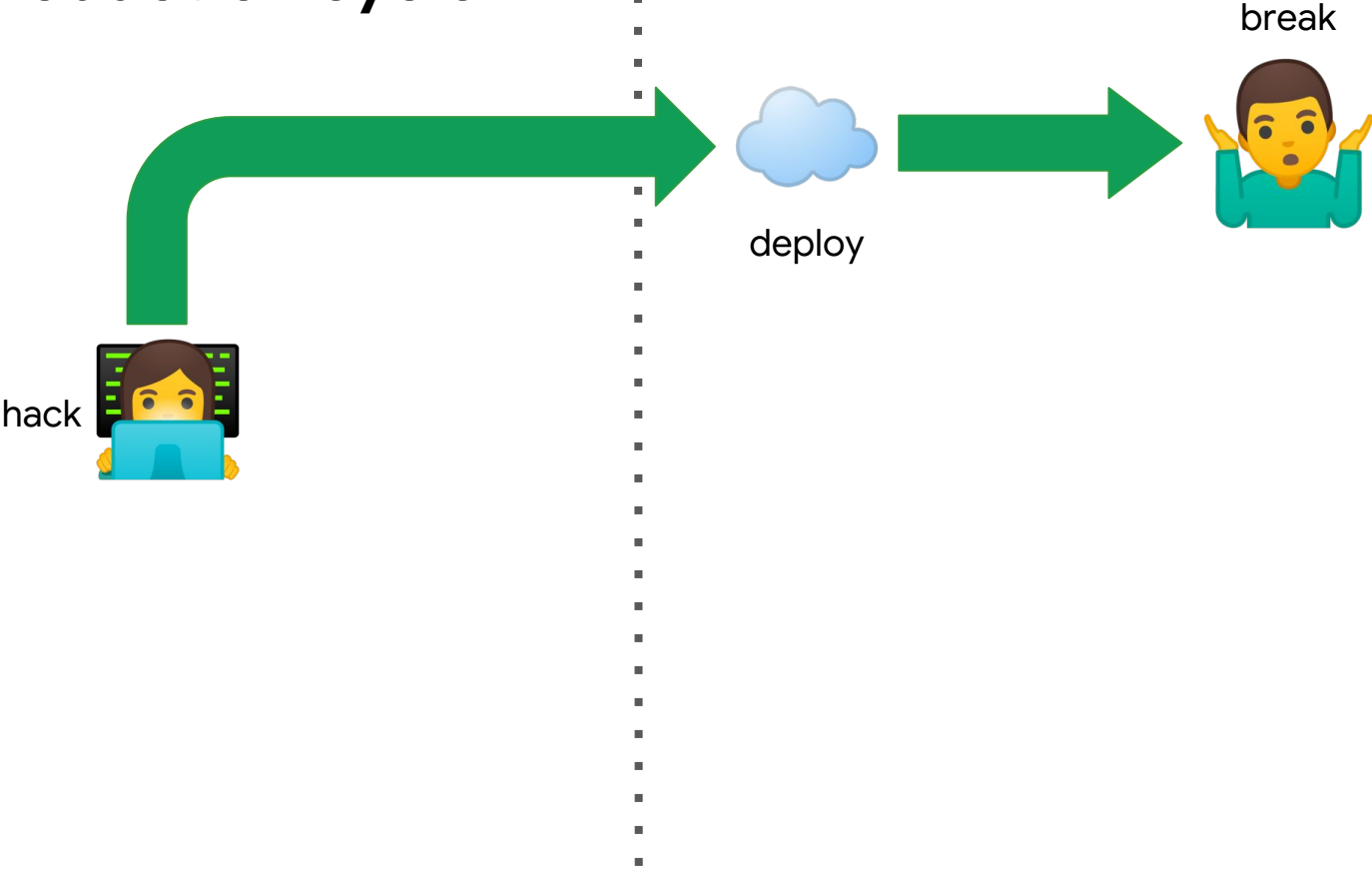
production cycle



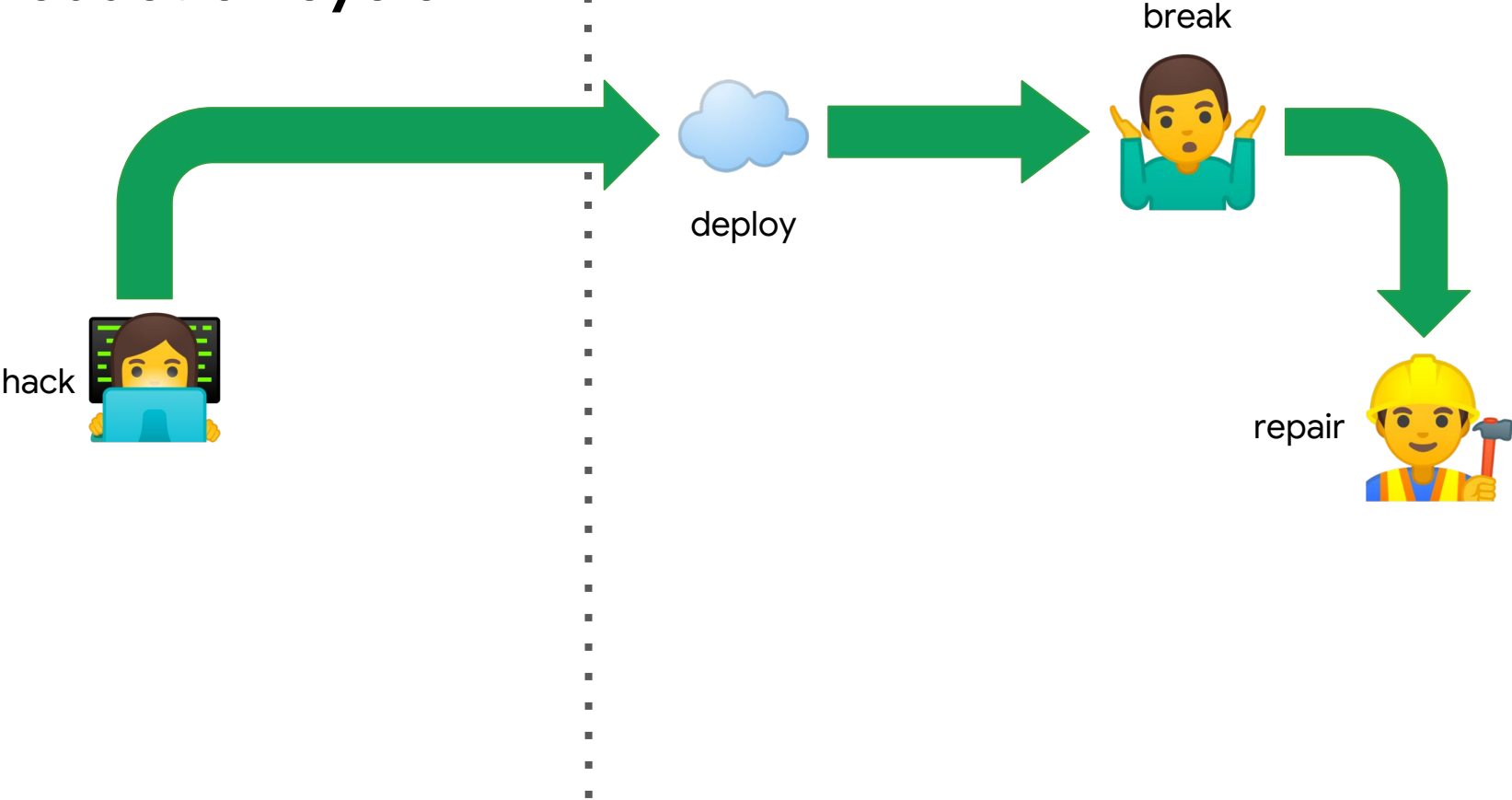
production cycle



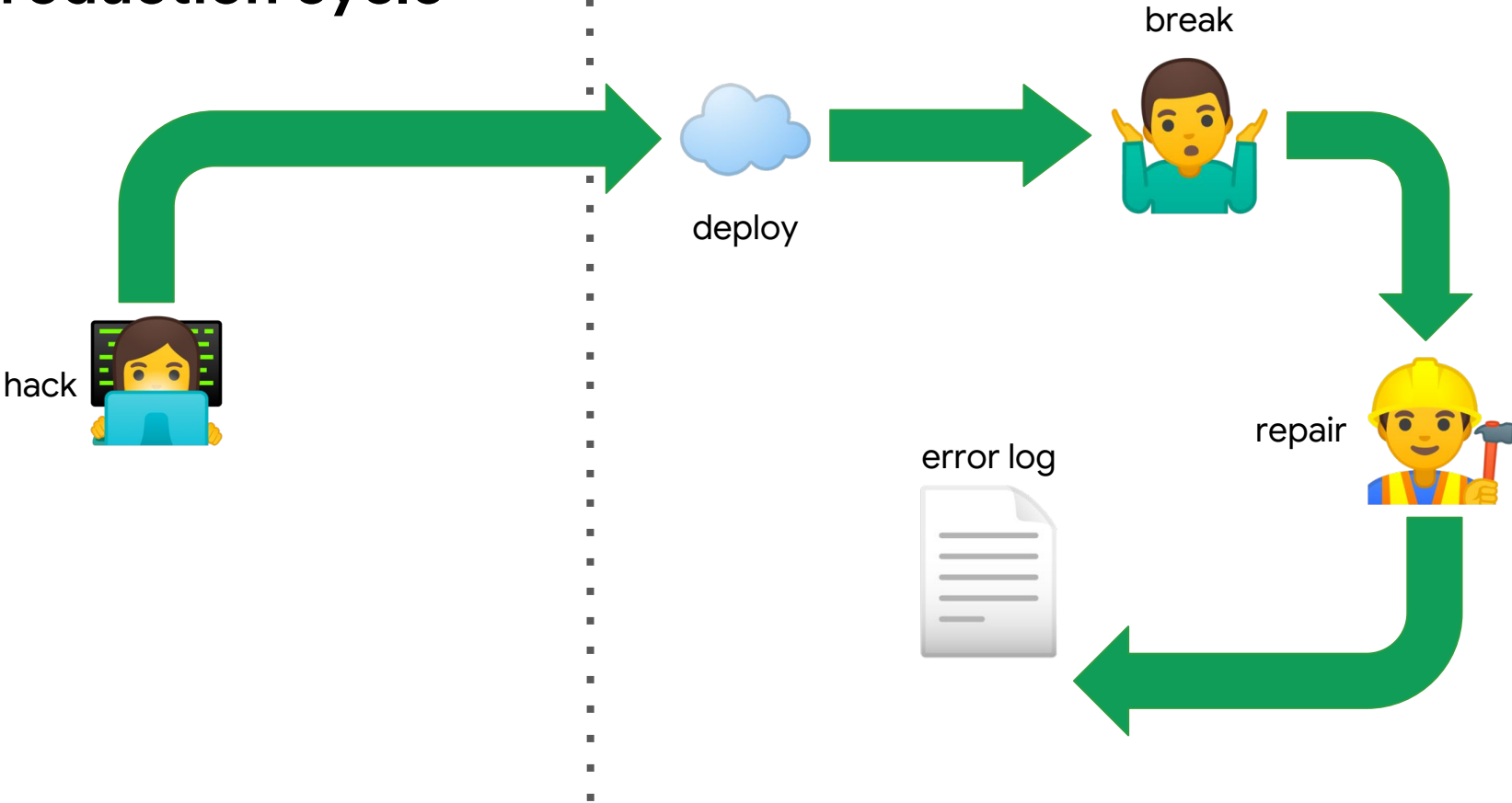
production cycle



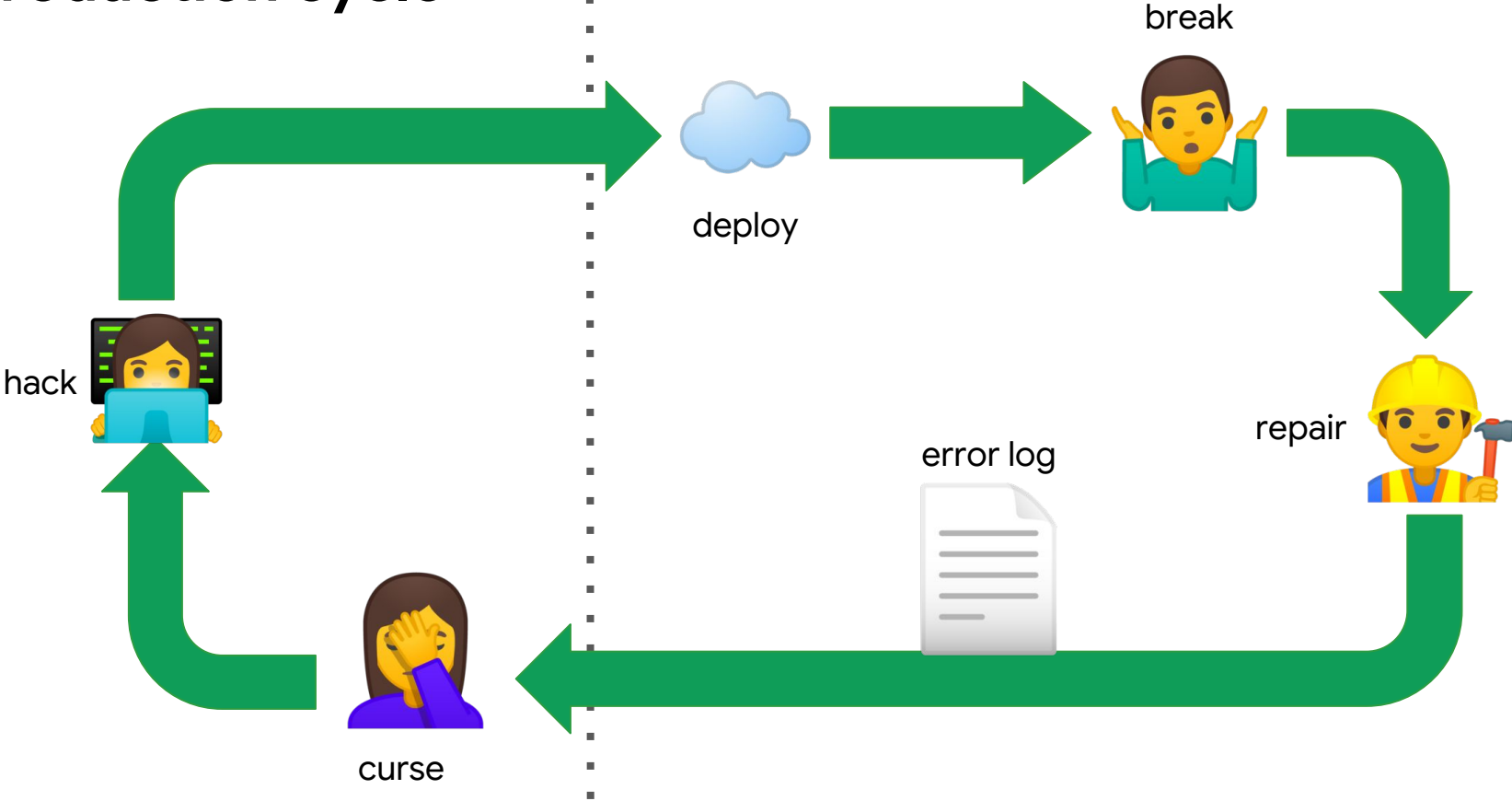
production cycle



production cycle



production cycle



```
1  async function foo() {
2    await bar();
3    return 42;
4  }
5
6  async function bar() {
7    await Promise.resolve();
8    throw new Error("BEEP BEEP");
9  }
10
11 foo().catch(e => console.log(e.stack));
```

```
1  async function foo() {
2    await bar();
3    return 42;
4  }
5
6  async function bar() {
7    await Promise.resolve();
8    throw new Error("BEEP BEEP");
9  }
10
11  foo().catch(e => console.log(e.stack));
```

```
1  async function foo() {
2    await bar();
3    return 42;
4  }
5
6  async function bar() {
7    await Promise.resolve();
8    throw new Error("BEEP BEEP");
9  }
10
11  foo().catch(e => console.log(e.stack));
```

```
1  async function foo() {
2    await bar();
3    return 42;
4  }
5
6  async function bar() {
7    await Promise.resolve();
8    throw new Error("BEEP BEEP");
9  }
10
11  foo().catch(e => console.log(e.stack));
```

```
$ node index.js
```

```
Error: BEEP BEEP
```

```
  at bar (index.js:8:9)
```

```
    at process._tickCallback (internal/process/next_tick.js:68:7)
```

```
    at Function.Module.runMain (internal/modules/cjs/loader.js:745:11)
```

```
    at startup (internal/bootstrap/node.js:266:19)
```

```
    at bootstrapNodeJSCore (internal/bootstrap/node.js:595:3)
```




Promise chain contains sufficient
information where to resume...





Promise chain contains sufficient information where to resume...

...in case of **await resume** and **suspend point coincides!**



```
1  async function foo() {
2    await bar();
3    return 42;
4  }
5
6  async function bar() {
7    await Promise.resolve();
8    throw new Error("BEEP BEEP");
9  }
10
11  foo().catch(e => console.log(e.stack));
```

```
1  async function foo() {
2    await bar();
3    return 42;
4  }
5
6  async function bar() {
7    await Promise.resolve();
8    throw new Error("BEEP BEEP");
9  }
10
11  foo().catch(e => console.log(e.stack));
```

```
$ node --async-stack-traces index.js
```

```
Error: BEEP BEEP
```

```
  at bar (index.js:8:9)
```

```
  at process._tickCallback (internal/process/next_tick.js:68:7)
```

```
  at Function.Module.runMain (internal/modules/cjs/loader.js:745:11)
```

```
  at startup (internal/bootstrap/node.js:266:19)
```

```
  at bootstrapNodeJSCore (internal/bootstrap/node.js:595:3)
```

```
  at async foo (index.js:2:3)
```

```
1  async function foo() {
2    await bar();
3    return 42;
4  }
5
6  async function bar() {
7    await Promise.resolve();
8    throw new Error("BEEP BEEP");
9  }
10
11  foo().catch(e => console.log(e.stack));
```

```
$ node --async-stack-traces index.js
```

```
Error: BEEP BEEP
```

```
  at bar (index.js:8:9)
```

```
    at process._tickCallback (internal/process/next_tick.js:68:7)
```

```
    at Function.Module.runMain (internal/modules/cjs/loader.js:745:11)
```

```
    at startup (internal/bootstrap/node.js:266:19)
```

```
    at bootstrapNodeJSCore (internal/bootstrap/node.js:595:3)
```

```
  at async foo (index.js:2:3)
```

```
1  async function foo() {
2    await bar();
3    return 42;
4  }
5
6  async function bar() {
7    await Promise.resolve();
8    throw new Error("BEEP BEEP");
9  }
10
11  foo().catch(e => console.log(e.stack));
```

```
$ node --async-stack-traces index.js
```

```
Error: BEEP BEEP
```

```
  at bar (index.js:8:9)
```

```
  at process._tickCallback (internal/process/next_tick.js:68:7)
```

```
  at Function.Module.runMain (internal/modules/cjs/loader.js:745:11)
```

```
  at startup (internal/bootstrap/node.js:266:19)
```

```
  at bootstrapNodeJSCore (internal/bootstrap/node.js:595:3)
```

```
  at async foo (index.js:2:3)
```

```
1  async function foo() {
2    await bar();
3    return 42;
4  }
5
6  async function bar() {
7    await Promise.resolve();
8    throw new Error("BEEP BEEP");
9  }
10
11  foo().catch(e => console.log(e.stack));
```

```
$ node --async-stack-traces index.js
```

```
Error: BEEP BEEP
```

```
  at bar (index.js:8:9)
```

```
  at process._tickCallback (internal/process/next_tick.js:68:7)
```

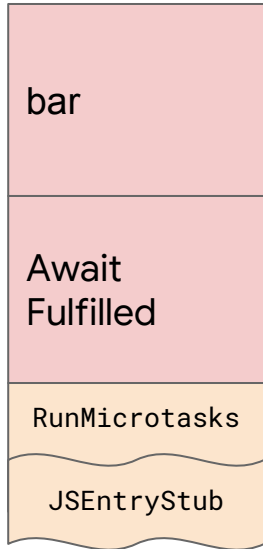
```
  at Function.Module.runMain (internal/modules/cjs/loader.js:745:11)
```

```
  at startup (internal/bootstrap/node.js:266:19)
```

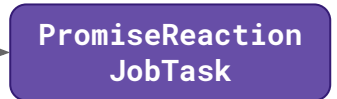
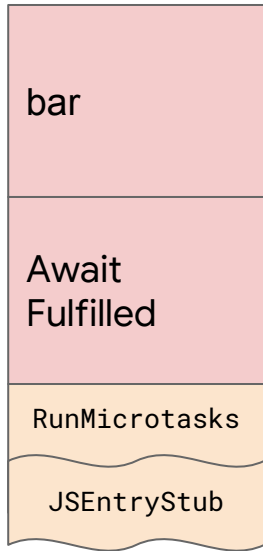
```
  at bootstrapNodeJSCore (internal/bootstrap/node.js:595:3)
```

```
  at async foo (index.js:2:3)
```

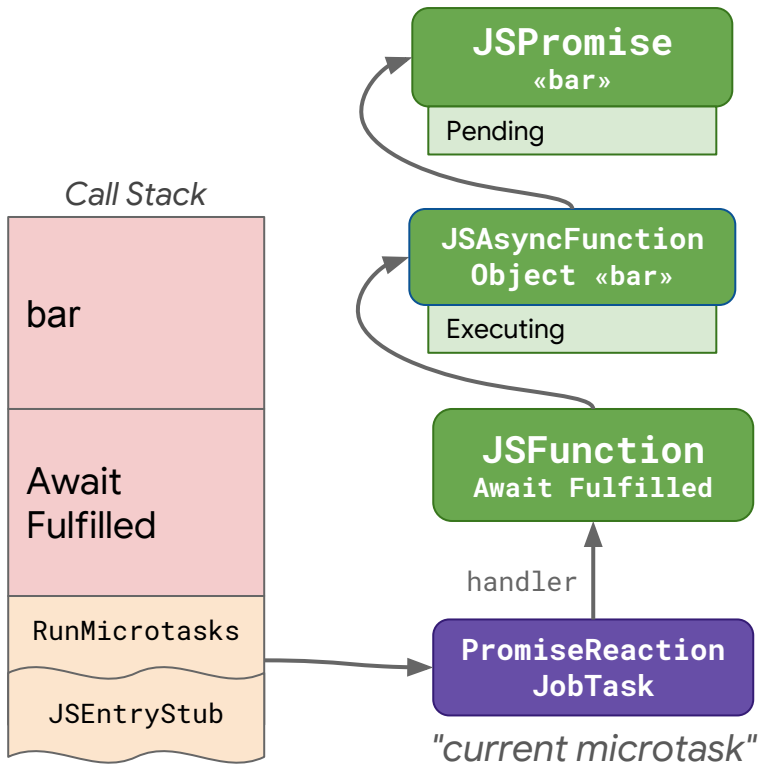
Call Stack

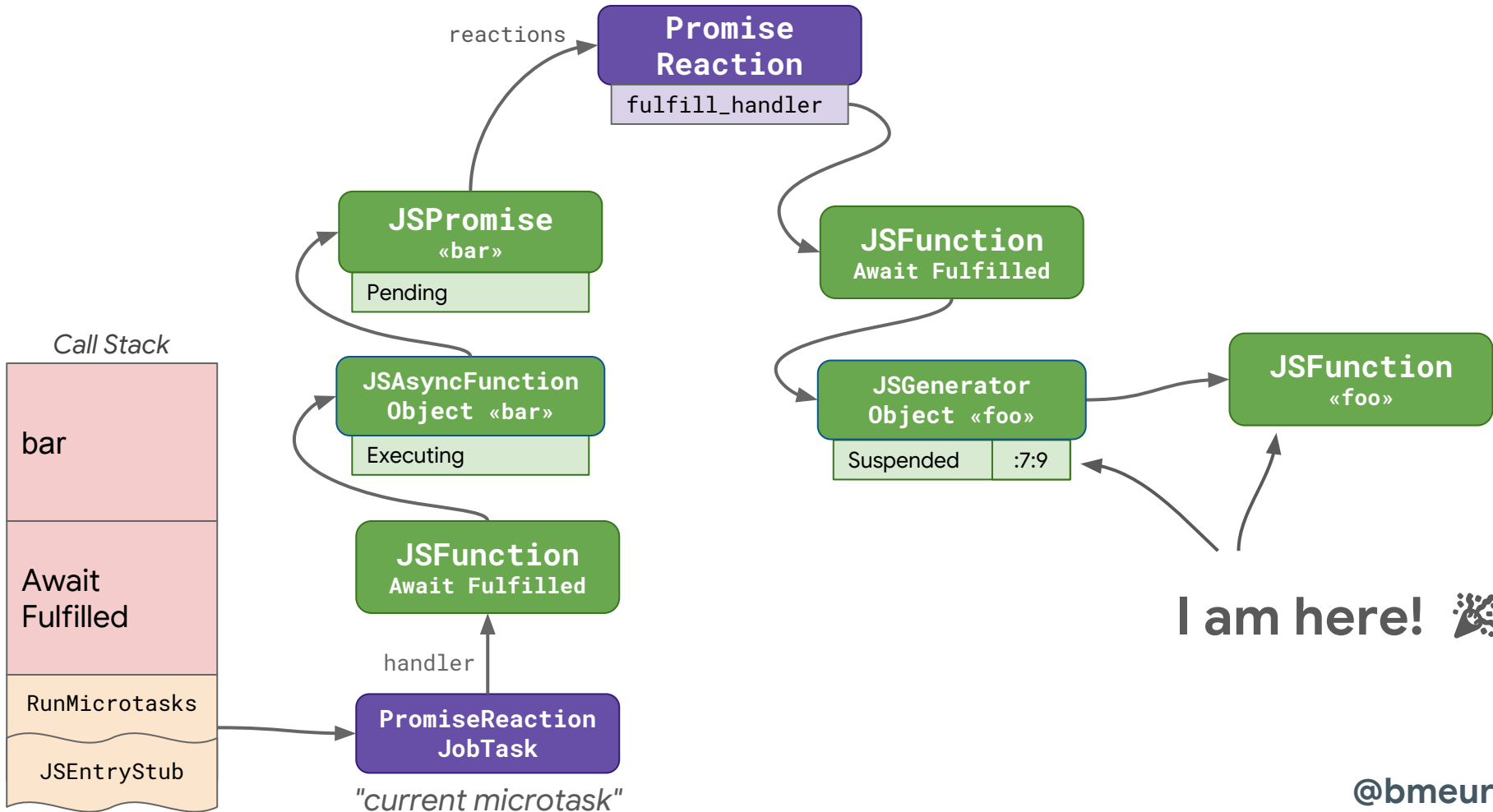


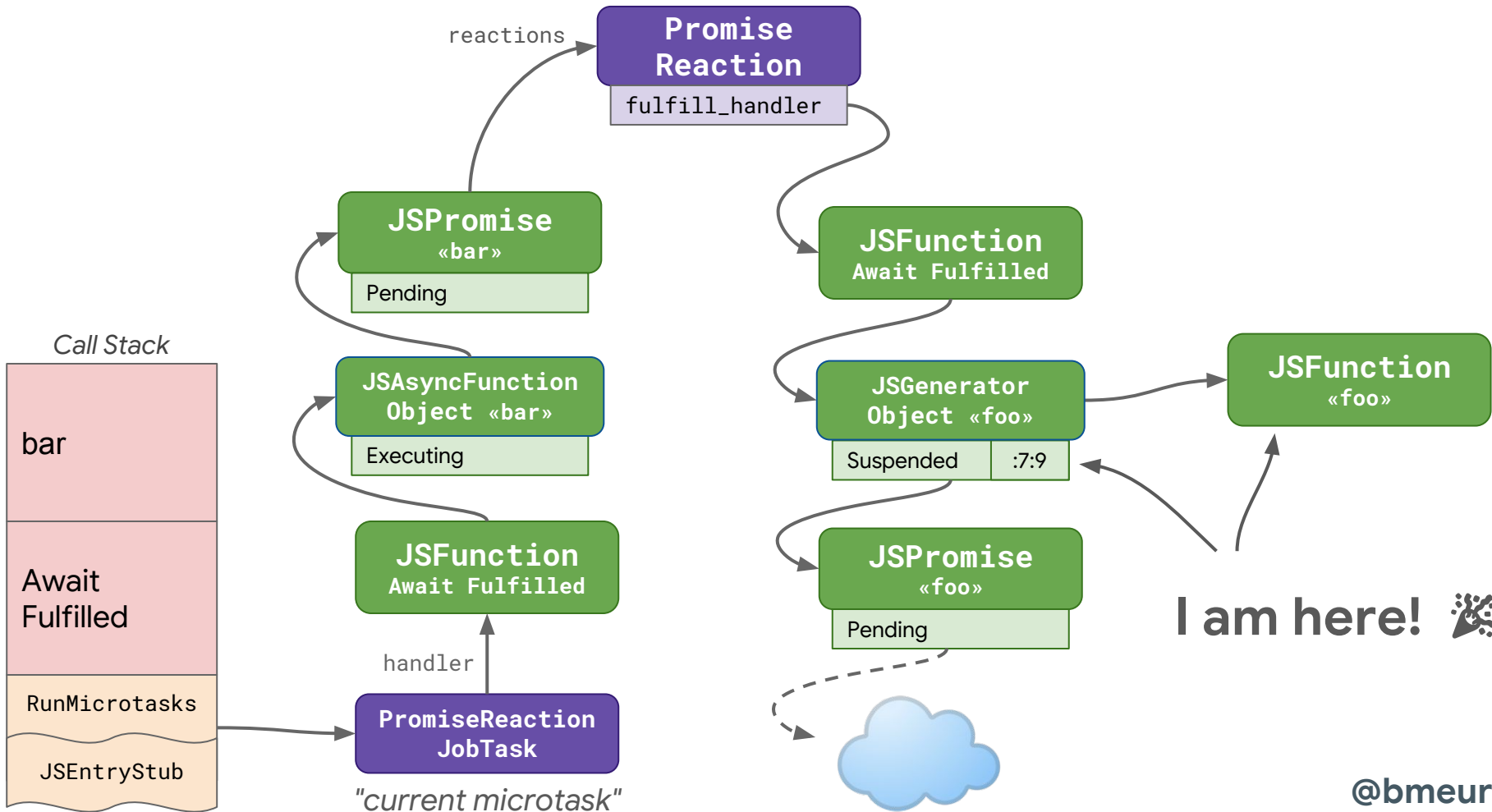
Call Stack



"current microtask"







Summary

- Limited to `async/await` and `Promise.all()`
- Can look through chained promises
- Shipped in V8 7.4

bit.ly/v8-zero-cost-async-stack-traces