# Hotwired Reactive Web Development

## How #LowJS can you go?

Josh Graham, Nov 2021

@delitescere

## About

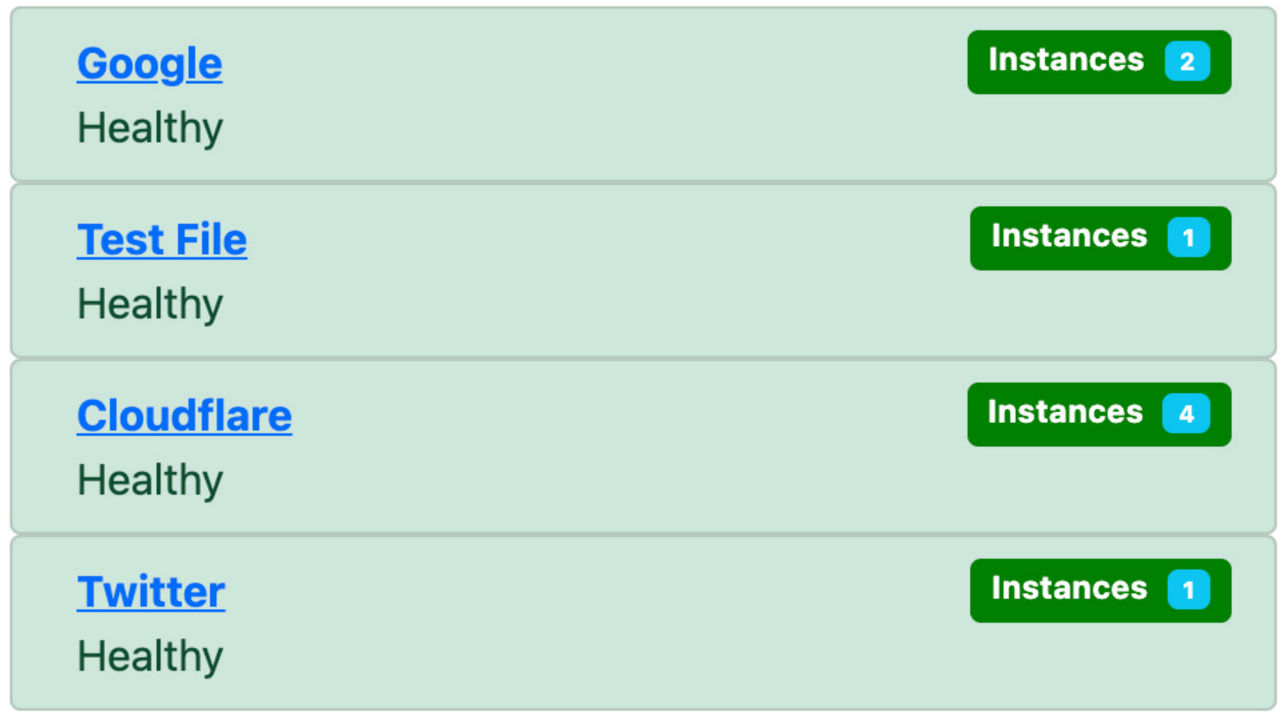
**Investor / Advisor / Change maker**

- **Canva** – *empowering everyone in the world to design anything and publish anywhere*
- **Atlassian** – *tools to unleash the potential of every team*
- **ThoughtWorks** – *global technology consultancy ("Agile")*
- **OzEmail** – *Australia's first major ISP*

# About

## Investor / Advisor / Change maker

- **Hydroplane** – *emission-free powered aviation & marine using hydrogen fuel cells*
- **Cookaborough** – *the platform for batch-based local food businesses to thrive*
- **Mass Dynamics** – *transforming mass spectrometry data into knowledge*
- **Secure Code Warrior** – *making secure coding a positive and engaging experience*



Google Instances 2
Healthy

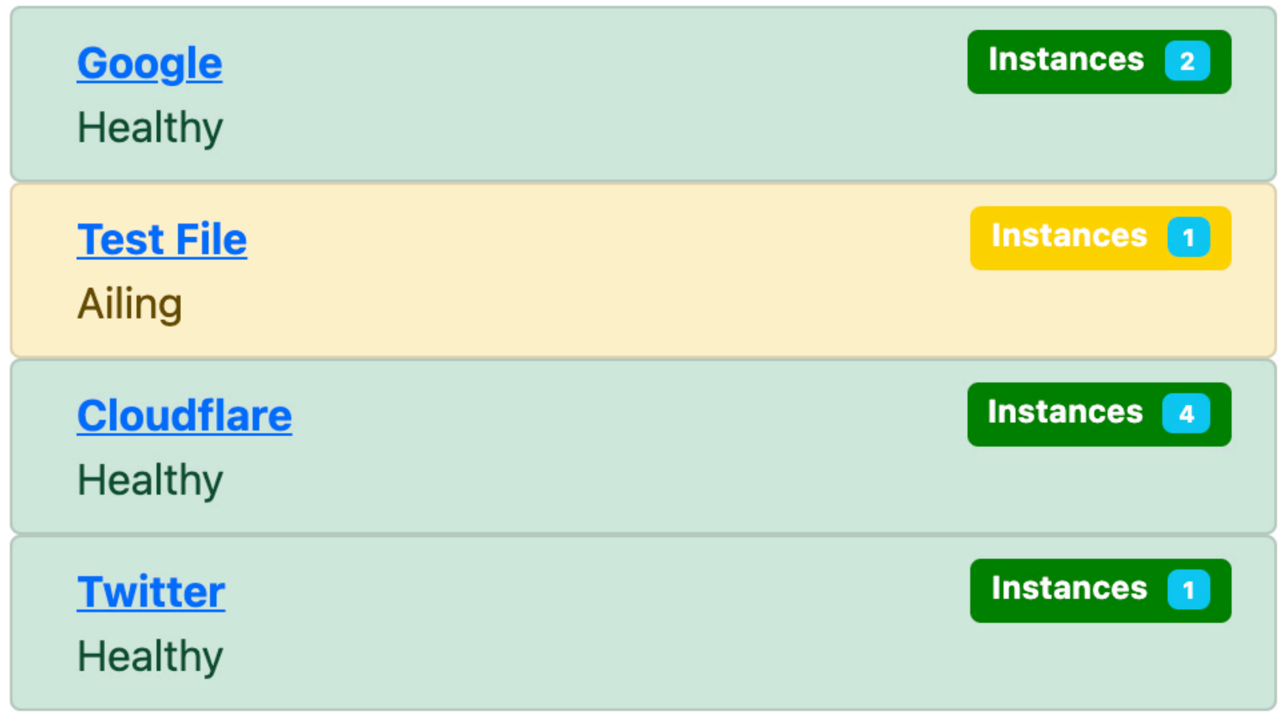
Test File Instances 1
Healthy

Cloudflare Instances 4
Healthy

Twitter Instances 1
Healthy

# status.gallery

## Simple information radiator

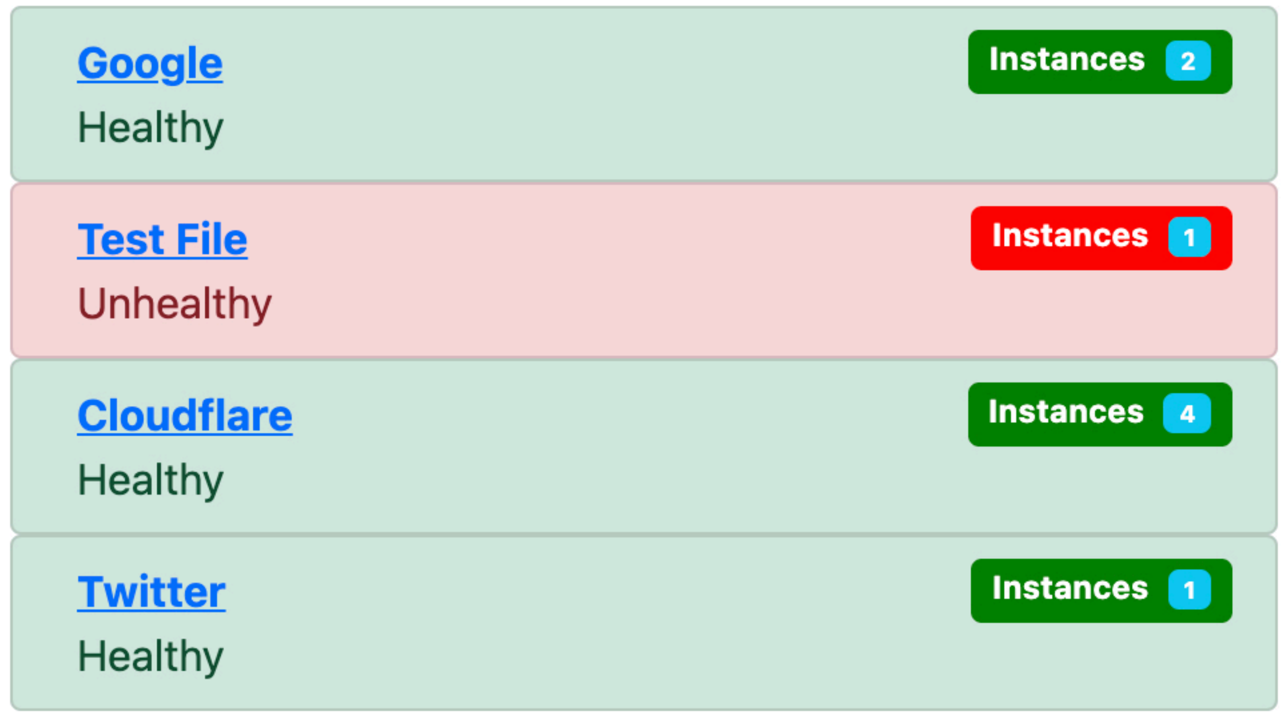


Google Instances 2
Healthy

Test File Instances 1
Ailing

Cloudflare Instances 4
Healthy

Twitter Instances 1
Healthy



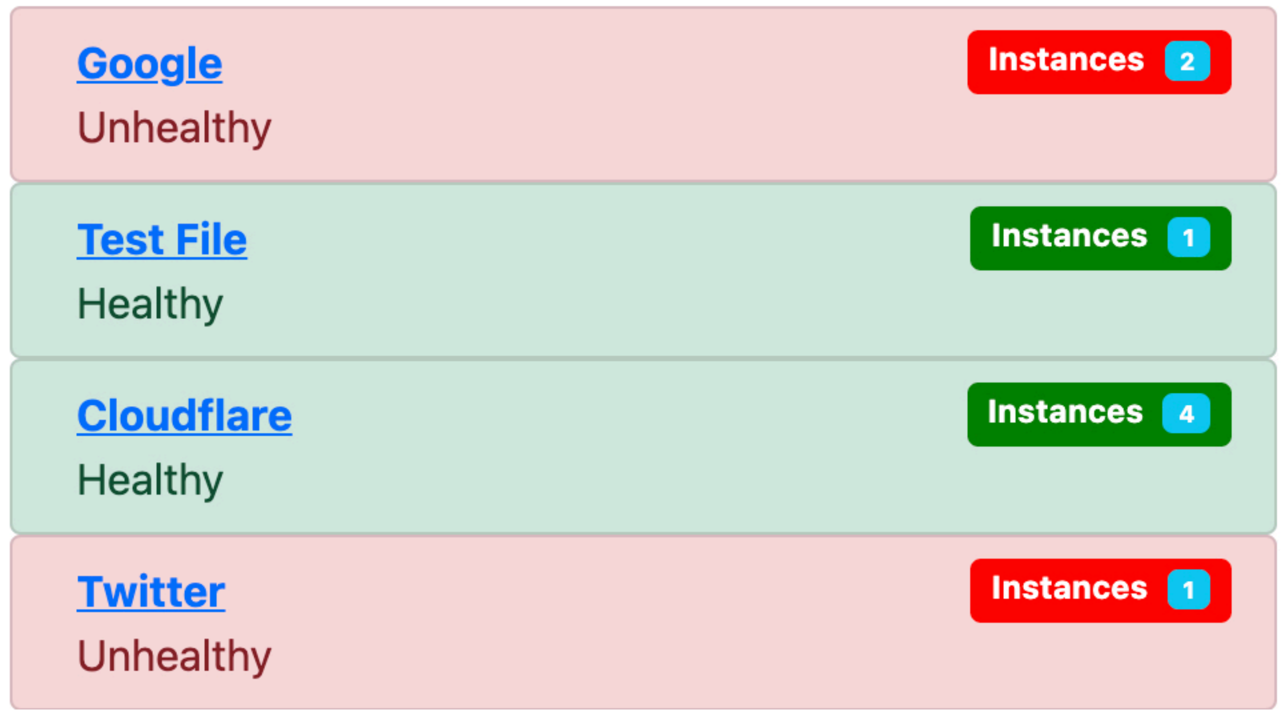
Google Instances 2
Healthy

Test File Instances 1
Unhealthy

Cloudflare Instances 4
Healthy

Twitter Instances 1
Healthy

## Healthy (Green)
## Ailing (Amber)
## Unhealthy (Red)



Google Instances 2
Unhealthy

Test File Instances 1
Healthy

Cloudflare Instances 4
Healthy

Twitter Instances 1
Unhealthy

# Motivation

**status.gallery**

- HTML, CSS
- JS ~120~~000~~ loc for progressive enhancement
  *(could be less in a better world)*
- Bootstrap, Hotwire
- SpringBoot, WebFlux, ThymeLeaf
- Kotlin Coroutines, Kotlin Flows.

## Motivation

# A slick, lightweight "frontend"

## Motivation

**Embrace the browser, using the principles of a Resource-Oriented Client Architecture**

`https://roca-style.org`

## Motivation

**Multi-channel**
**Low-latency**
**Highly concurrent**
**Asynchronous**
**On-demand**
**Responsive**
**Accessible**
**Testable**
**Only-what’s-needed**
**Server-pushable**
**Simple**

## Background

***Simple != Easy***

- Essential for reliability
- Aids in understanding
  - Problem domain
  - Solution
  - Technical underpinnings
- More accurate (and almost always faster) diagnosis of issues
- More precise (and often faster) changes (fewer "unintended consequences").

## Background

# No data* from server to browser

Background

# Dynamic HTML

## Background

AJAX

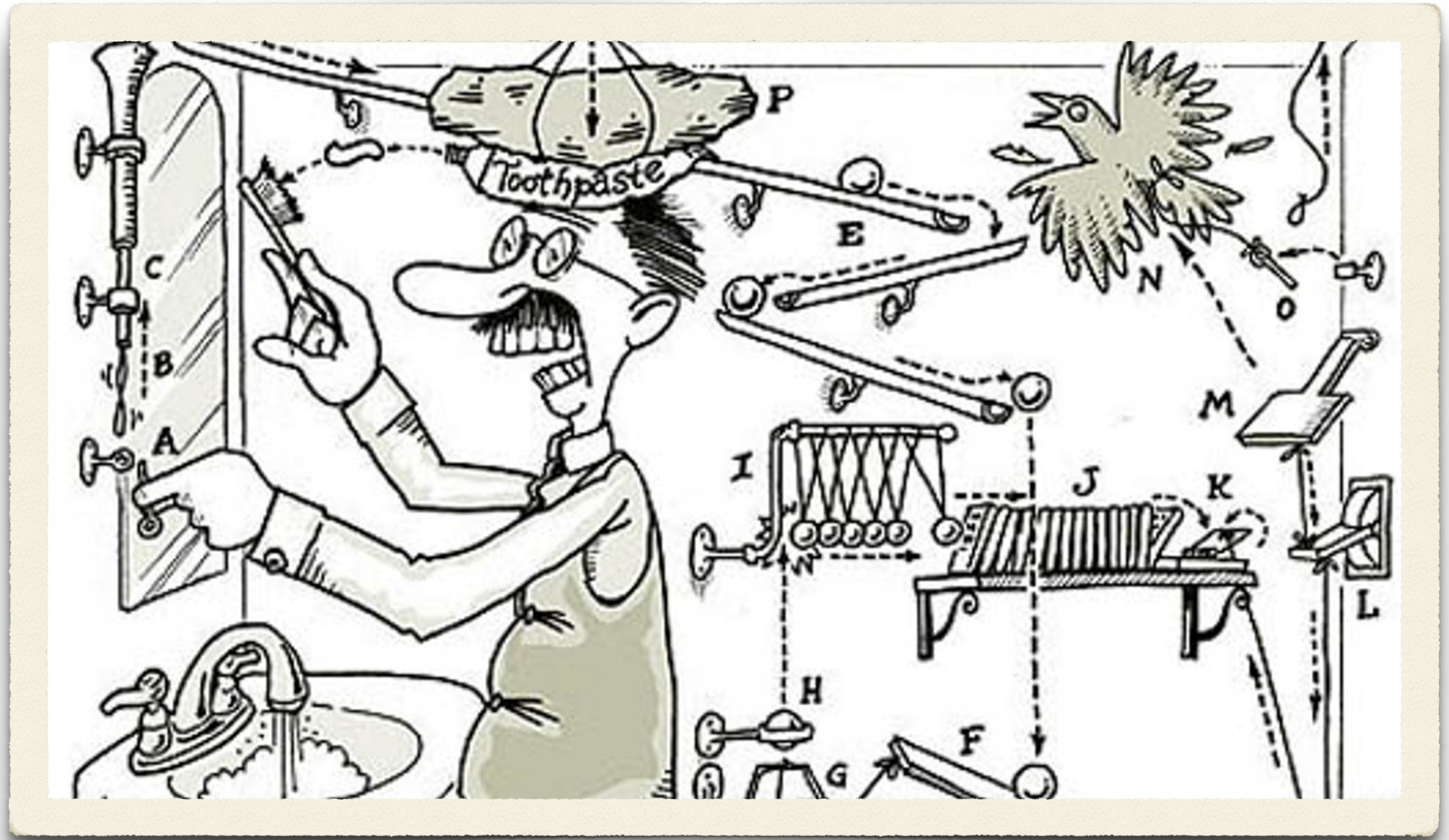
## Background

# XML + XSLT

## Background

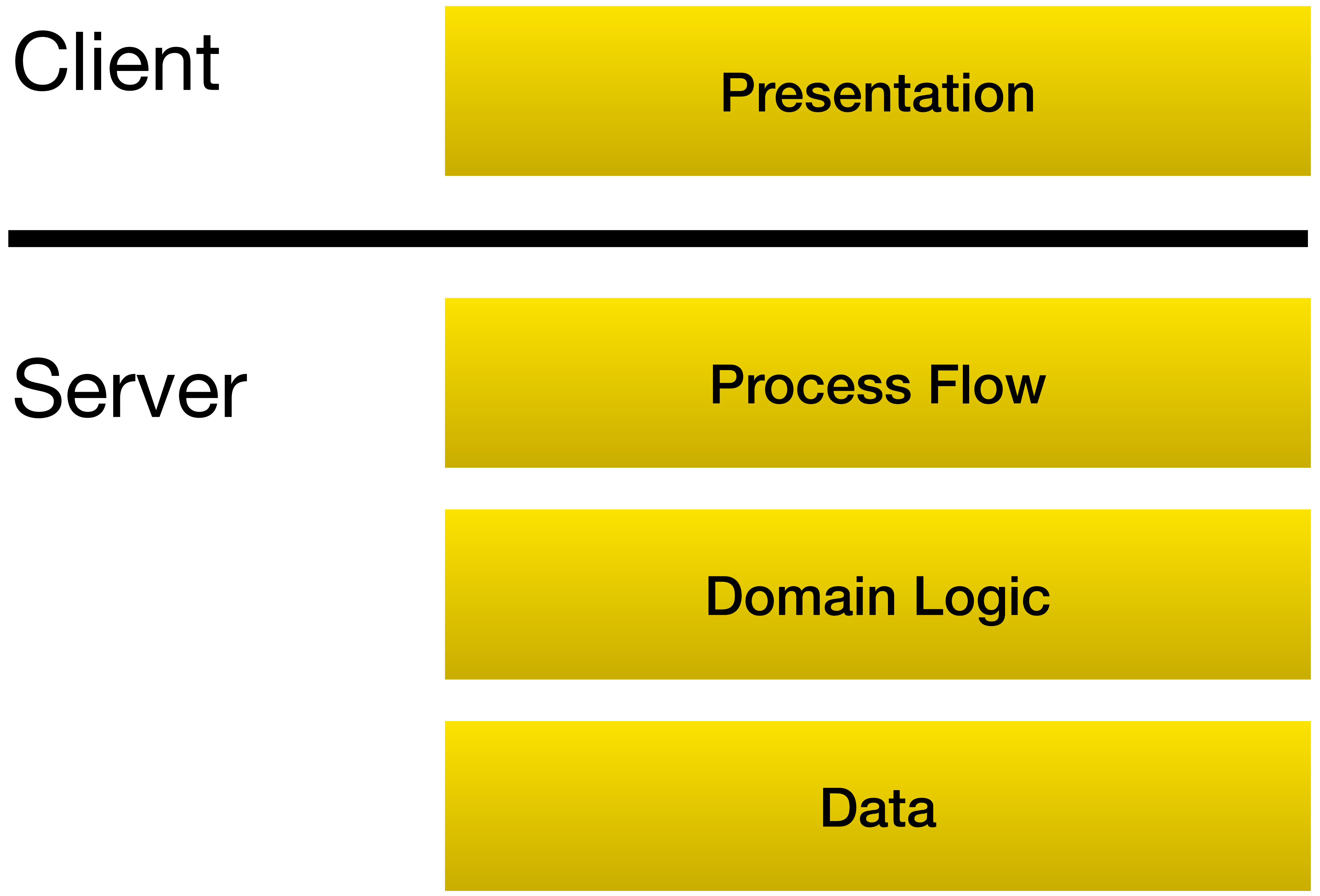
JSON

## Background



# The web-native way of distributing logic

**© Stefan Tilkov**



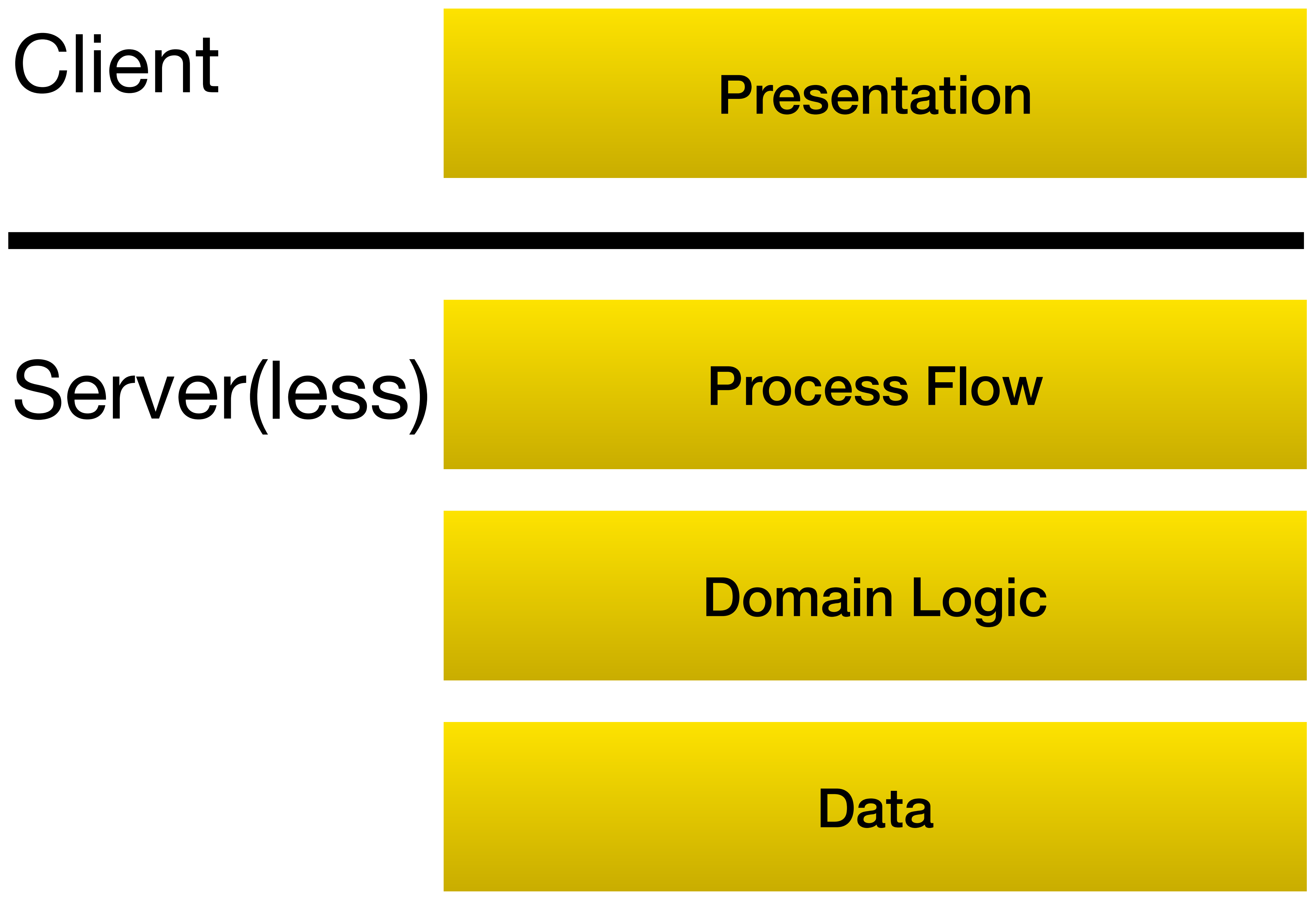
> Rendering, layout, styling on an *unknown* client

> Logic & state machine on server

> Client user-agent extensible via code on demand

# The web-native way of distributing logic

**© Stefan Tilkov**



> Rendering, layout, styling on an *unknown* client

> Logic & state machine on server

> Client user-agent extensible via code on demand

## Background

#LowJS

## Background

### Some #LowJS toolkits

- htmx htmx.org (née intercooler.js)
- Hotwire hotwired.dev
  - Turbo – responsiveness, components, streaming updates
  - Stimulus – HTML-centric state and wiring with "a dash of custom code"
  - Strada – progressively enhance web interactions with native replacements

**Hotwire** is an alternative approach to building modern web applications without using much JavaScript by sending HTML instead of JSON over the wire.

## **H**TML **o**ver **t**he **wire**

## Hotwire

### Wait. HTML as a data transfer format ?!?

*[after making a good case to do so]*

But we all know... you would never use HTML...

Just wait a year... maybe it'll be the thing of the future.

— Stefan Tilkov, GOTO 2014

## Hotwire

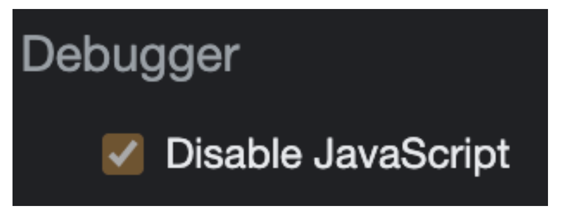
### Benefits

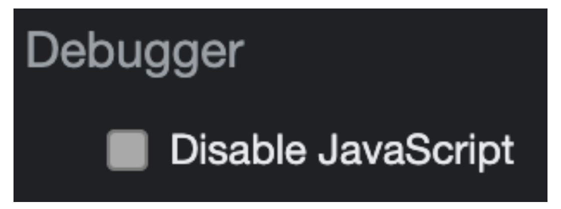
- Fast first-load pages
- Keeps logic on the server
- A simpler, more productive development experience in any programming language
- Without sacrificing any of the speed or responsiveness associated with a Single Page Application.

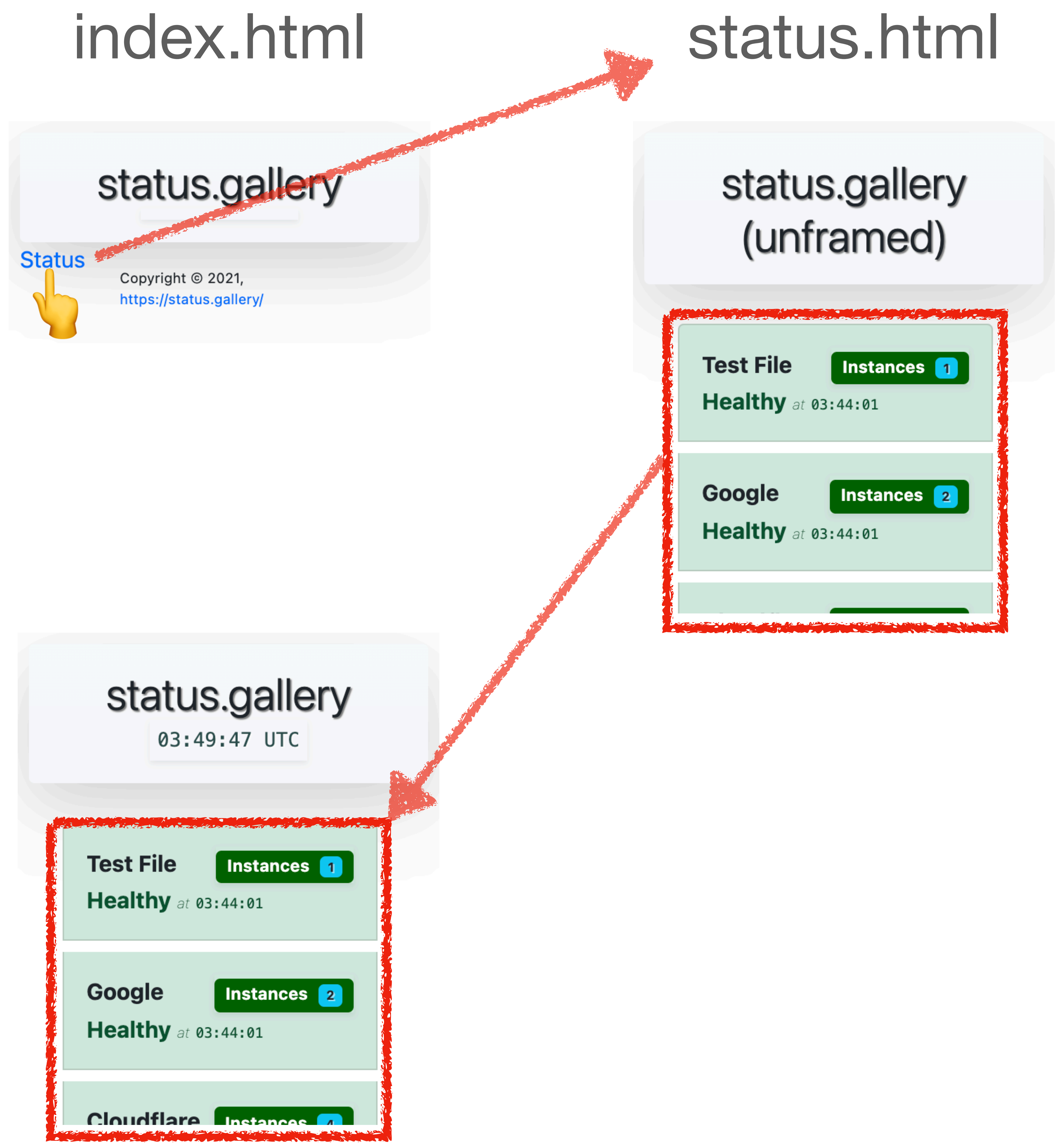
# Hotwire: Turbo Frames

## Transclusion

- If JS is not available, a regular navigation takes place
- Thus the transcluded resource should be well-formed HTML
- Otherwise, a fragment of the second page is used to update a portion of the first page
- Automatically transclude with a `click()` instruction.


Debugger
✓ Disable JavaScript


Debugger
Disable JavaScript



# index.html

```html
<!DOCTYPE html>
<html lang="${lang}"
      th:lang="${lang}"
      th:with="lang=${#locale.language}"
      xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
<head>
    <link th:replace="~{fragments/html-head}"/>
    <script src="/scripts/index.js" async defer></script>
    <script src="/scripts/status.js" async></script>
    <title>status.gallery</title>
</head>
<body>
<header class="bg-light bg-gradient p-3 rounded shadow-lg">
    <div class="display-4 text-center text-shadow">status.gallery</div>
    <div class="mx-auto small font-monospace text-center shadow-sm p-1" style="width: 8em">
        <time id="clock" datetime=""></time>
    </div>
</header>
<turbo-frame id="status_frame" autoscroll data-autoscroll-block="start">
    <a href="/status" id="status_frame_load">
        <!-- #ProgressiveEnhancement -->
        <script>document.currentScript.parentElement.hidden = true;</script>
        Status
    </a>
    </div>
</turbo-frame>
<link th:replace="~{fragments/footer}"/>
</body>
</html>
```

index.html

```html
<!DOCTYPE html>
<html lang="${lang}"
      th:lang="${lang}"
      th:with="lang=${#locale.language}"
      xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
<head>
    <link th:replace="~{fragments/html-head}"/>
    <script src="/scripts/index.js" async defer></script>
    <script src="/scripts/status.js" async></script>
    <title>status.gallery</title>
</head>
<body>
<header class="bg-light bg-gradient p-3 rounded shadow-lg">
    <div class="display-4 text-center text-shadow">status.gallery</div>
    <div class="mx-auto small font-monospace text-center shadow-sm p-1" style="width: 8em">
        <time id="clock" datetime=""></time>
    </div>
</header>
<turbo-frame id="status_frame" autoscroll data-autoscroll-block="start">
    <a href="/status" id="status_frame_load">
        <!-- #ProgressiveEnhancement -->
        <script>document.currentScript.parentElement.hidden = true;</script>
        Status
    </a>
    </div>
</turbo-frame>
<link th:replace="~{fragments/footer}"/>
</body>
</html>
```

# status.html

```html
<!DOCTYPE html>
<html lang="${lang}"
      th:lang="${lang}"
      th:with="lang=${#locale.language}"
      xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
<head>
    <link th:replace="~{fragments/html-head}"/>
    <title>status.gallery Status Page (unframed)</title>
    <!-- if Turbo is available, index.html won't have included this head, so the meta refresh won't happen (which is good!) -->
    <!-- if JavaScript is disabled, Turbo won't be available, so this meta refresh will take the place of SSE -->
    <meta content="3" http-equiv="refresh" name="refresh"/>
    <script src="/scripts/status.js" async></script>
</head>
<body>
<div class="bg-light bg-gradient p-3 rounded shadow-lg">
    <h1 class="display-1 text-center" style="text-shadow: 0 1px 0 gray">status.gallery (unframed)</h1>
</div>
<turbo-frame id="status_frame">
    <div class="box">
        <ul class="container list-group">
            <li th:class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
                th:each="observee : *{observees}"
                th:id="|observee*{id}|"
                th:object="${observee}">
                <div th:replace="~{fragments/observee-status-li :: observee-status-li}"></div>
            </li>
        </ul>
    </div>
</turbo-frame>
</body>
<link th:replace="~{fragments/footer}"/>
</html>
```

## status.html

```html
<!DOCTYPE html>
<html lang="${lang}"
      th:lang="${lang}"
      th:with="lang=${#locale.language}"
      xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
<head>
    <link th:replace="~{fragments/html-head}"/>
    <title>status.gallery Status Page (unframed)</title>
    <!-- if Turbo is available, index.html won't have included this head, so the meta refresh won't happen (which is good!) -->
    <!-- if JavaScript is disabled, Turbo won't be available, so this meta refresh will take the place of SSE -->
    <meta content="3" http-equiv="refresh" name="refresh"/>
    <script src="/scripts/status.js" async></script>
</head>
<body>
<div class="bg-light bg-gradient p-3 rounded shadow-lg">
    <h1 class="display-1 text-center" style="text-shadow: 0 1px 0 gray">status.gallery (unframed)</h1>
</div>
<turbo-frame id="status_frame">
    <div class="box">
        <ul class="container list-group">
            <li th:class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
                th:each="observee : *{observees}"
                th:id="|observee*{id}|"
                th:object="${observee}">
                <div th:replace="~{fragments/observee-status-li :: observee-status-li}"></div>
            </li>
        </ul>
    </div>
</turbo-frame>
</body>
<link th:replace="~{fragments/footer}"/>
</html>
```

# status.html

```html
<!DOCTYPE html>
<html lang="${lang}"
      th:lang="${lang}"
      th:with="lang=${#locale.language}"
      xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
<head>
    <link th:replace="~{fragments/html-head}"/>
    <title>status.gallery Status Page (unframed)</title>
    <!-- if Turbo is available, index.html won't have included this head, so the meta refresh won't happen (which is good!) -->
    <!-- if JavaScript is disabled, Turbo won't be available, so this meta refresh will take the place of SSE -->
    <meta content="3" http-equiv="refresh" name="refresh"/>
    <script src="/scripts/status.js" async></script>
</head>
<body>
<div class="bg-light bg-gradient p-3 rounded shadow-lg">
    <h1 class="display-1 text-center" style="text-shadow: 0 1px 0 gray">status.gallery (unframed)</h1>
</div>
<turbo-frame id="status_frame">
    <div class="box">
        <ul class="container list-group">
            <li th:class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
                th:each="observee : *{observees}"
                th:id="|observee*{id}|"
                th:object="${observee}">
                <div th:replace="~{fragments/observee-status-li :: observee-status-li}"></div>
            </li>
        </ul>
    </div>
</turbo-frame>
</body>
<link th:replace="~{fragments/footer}"/>
</html>
```

# status.html

```html
<!DOCTYPE html>
<html lang="${lang}"
      th:lang="${lang}"
      th:with="lang=${#locale.language}"
      xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
<head>
    <link th:replace="~{fragments/html-head}"/>
    <title>status.gallery Status Page (unframed)</title>
    <!-- if Turbo is available, index.html won't have included this head, so the meta refresh won't happen (which is good!) -->
    <!-- if JavaScript is disabled, Turbo won't be available, so this meta refresh will take the place of SSE -->
    <meta content="3" http-equiv="refresh" name="refresh"/>
    <script src="/scripts/status.js" async></script>
</head>
<body>
<div class="bg-light bg-gradient p-3 rounded shadow-lg">
    <h1 class="display-1 text-center" style="text-shadow: 0 1px 0 gray">status.gallery (unframed)</h1>
</div>
<turbo-frame id="status_frame">
    <div class="box">
        <ul class="container list-group">
            <li th:class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
                th:each="observee : *{observees}"
                th:id="|observee*{id}|"
                th:object="${observee}">
                <div th:replace="~{fragments/observee-status-li :: observee-status-li}"></div>
            </li>
        </ul>
    </div>
</turbo-frame>
</body>
<link th:replace="~{fragments/footer}"/>
</html>
```

# Hotwire: Turbo Streams

## WebSocket, Server-Sent Events, form submission

- HTML (fragments)
- In lieu of XML or JSON
- DOM directly updated (by Turbo library)
- *Only* DOM updates
- Use Stimulus for sprinkling of JS, e.g. reset form fields after response received
- Work smoothly with Thymeleaf / Spring integration's `ReactiveDataDriverContextVariable` and reactive collections (e.g. a Flux or a Flow)
- I prefer SSE.

## status.js

```
// #ProgressiveEnhancement
(window['EventSource'] && window['Turbo']) ?
  Turbo.connectStreamSource(new EventSource('/status.stream')) :
  console.warn('Turbo Streams over SSE not available');
```

## StatusController.kt

```kotlin
@FlowPreview
@ExperimentalTime
@Controller
internal class StatusController @Autowired constructor(private val supervisor: ObserveeSupervisor) {

    @GetMapping("/status", produces = [MediaType.TEXT_HTML_VALUE])
    suspend fun index(model: Model): String {
        val observees = ObserveeInfo.from(supervisor.observeeHealthFlows)
        model.addAttribute("observees", observees)
        return "status"
    }

    @GetMapping("/status.stream", produces = [MediaType.TEXT_EVENT_STREAM_VALUE, CustomMediaType.TURBO_STREAM_VALUE])
    suspend fun stream(model: Model): String {
        val observees = ObserveeInfo.from(supervisor.observeeHealthFlows.asFlow())
        model.addAttribute("observees", dataDrivenEach(observees))
        return "observee-status.turbo-stream"
    }

    private fun dataDrivenEach(stream: Flow<Any>) = ReactiveDataDriverContextVariable(stream, 1)
}
```

## StatusController.kt

```kotlin
@FlowPreview
@ExperimentalTime
@Controller
internal class StatusController @Autowired constructor(private val supervisor: ObserveeSupervisor) {

    @GetMapping("/status", produces = [MediaType.TEXT_HTML_VALUE])
    suspend fun index(model: Model): String {
        val observees = ObserveeInfo.from(supervisor.observeeHealthFlows)
        model.addAttribute("observees", observees)
        return "status"
    }

    @GetMapping("/status.stream", produces = [MediaType.TEXT_EVENT_STREAM_VALUE, CustomMediaType.TURBO_STREAM_VALUE])
    suspend fun stream(model: Model): String {
        val observees = ObserveeInfo.from(supervisor.observeeHealthFlows.asFlow())
        model.addAttribute("observees", dataDrivenEach(observees))
        return "observee-status.turbo-stream"
    }

    private fun dataDrivenEach(stream: Flow<Any>) = ReactiveDataDriverContextVariable(stream, 1)
}
```

## StatusController.kt

```kotlin
@FlowPreview
@ExperimentalTime
@Controller
internal class StatusController @Autowired constructor(private val supervisor: ObserveeSupervisor) {

    @GetMapping("/status", produces = [MediaType.TEXT_HTML_VALUE])
    suspend fun index(model: Model): String {
        val observees = ObserveeInfo.from(supervisor.observeeHealthFlows)
        model.addAttribute("observees", observees)
        return "status"
    }

    @GetMapping("/status.stream", produces = [MediaType.TEXT_EVENT_STREAM_VALUE, CustomMediaType.TURBO_STREAM_VALUE])
    suspend fun stream(model: Model): String {
        val observees = ObserveeInfo.from(supervisor.observeeHealthFlows.asFlow())
        model.addAttribute("observees", dataDrivenEach(observees))
        return "observee-status.turbo-stream"
    }

    private fun dataDrivenEach(stream: Flow<Any>) = ReactiveDataDriverContextVariable(stream, 1)
}
```

StatusController.kt

```kotlin
@FlowPreview
@ExperimentalTime
@Controller
internal class StatusController @Autowired constructor(private val supervisor: ObserveeSupervisor) {

    @GetMapping("/status", produces = [MediaType.TEXT_HTML_VALUE])
    suspend fun index(model: Model): String {
        val observees = ObserveeInfo.from(supervisor.observeeHealthFlows)
        model.addAttribute("observees", observees)
        return "status"
    }

    @GetMapping("/status.stream", produces = [MediaType.TEXT_EVENT_STREAM_VALUE, CustomMediaType.TURBO_STREAM_VALUE])
    suspend fun stream(model: Model): String {
        val observees = ObserveeInfo.from(supervisor.observeeHealthFlows.asFlow())
        model.addAttribute("observees", dataDrivenEach(observees))
        return "observee-status.turbo-stream"
    }

    private fun dataDrivenEach(stream: Flow<Any>) = ReactiveDataDriverContextVariable(stream, 1)
}
```

## observee-status.turbo-stream.html

```
<turbo-stream action="replace" data-th-each="observee : *{observees}" data-th-target="|observee${observee.id}|">
    <template>
        <li data-th-class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
            data-th-id="|observee*{id}|"
            data-th-object="${observee}">
            <div data-th-replace="~{fragments/observee-status-li :: observee-status-li}"></div>
        </li>
    </template>
</turbo-stream>
```

## observee-status.turbo-stream.html

```html
<turbo-stream action="replace" data-th-each="observee : *{observees}" data-th-target="|observee${observee.id}|">
    <template>
        <li data-th-class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
            data-th-id="|observee*{id}|"
            data-th-object="${observee}">
            <div data-th-replace="~{fragments/observee-status-li :: observee-status-li}"></div>
        </li>
    </template>
</turbo-stream>
```

# observee-status.turbo-stream.html

```html
<turbo-stream action="replace" data-th-each="observee : *{observees}" data-th-target="|observee${observee.id}|">
    <template>
        <li data-th-class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
            data-th-id="|observee*{id}|"
            data-th-object="${observee}">
            <div data-th-replace="~{fragments/observee-status-li :: observee-status-li}"></div>
        </li>
    </template>
</turbo-stream>
```

×
Headers
EventStream
Initiator
Timing
Cookies

| Id | Type | Data | Time |
|---|---|---|---|
| 0 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="file:src/sample/test.file">Test File</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark te… | 09:35:49.163 |
| 1 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="https://abc.xyz">Google</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark text-etched">… | 09:35:49.164 |
| 2 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="https://cloudflare.com">Cloudflare</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark text… | 09:35:49.165 |
| 3 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="https://twitter.com">Twitter</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark text-etc… | 09:35:49.165 |
| 4 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="https://cloudflare.com">Cloudflare</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark text… | 09:36:01.698 |
| 5 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="file:src/sample/test.file">Test File</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark te… | 09:36:02.674 |
| 6 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="https://twitter.com">Twitter</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark text-etc… | 09:36:11.788 |
| 7 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="https://abc.xyz">Google</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark text-etched">… | 09:36:21.122 |
| 8 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="https://cloudflare.com">Cloudflare</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark text… | 09:37:01.704 |
| 9 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="file:src/sample/test.file">Test File</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark te… | 09:37:02.769 |
| 10 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="https://twitter.com">Twitter</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark text-etc… | 09:37:12.440 |
| 11 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="https://abc.xyz">Google</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark text-etched">… | 09:37:22.746 |
| 12 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href="https://cloudflare.com">Cloudflare</a> </div> <div class="col-auto px-0 shadow-sm"> <span class="badge green">Instances <span class="badge bg-info ms-1 text-dark text… | 09:38:01.734 |

# observee-status.turbo-stream.html

```html
<turbo-stream action="replace" data-th-each="observee : *{observees}" data-th-target="|observee${observee.id}|">
    <template>
        <li data-th-class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
            data-th-id="|observee*{id}|"
            data-th-object="${observee}">
            <div data-th-replace="~{fragments/observee-status-li :: observee-status-li}"></div>
        </li>
    </template>
</turbo-stream>
```

× Headers **EventStream** Initiator Timing Cookies

| Id | Type | Data |
|---|---|---|
| 0 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 1 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 2 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 3 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 4 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 5 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 6 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 7 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 8 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 9 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 10 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 11 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 12 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 13 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |

# observee-status.turbo-stream.html

```html
<turbo-stream action="replace" data-th-each="observee : *{observees}" data-th-target="|observee${observee.id}|">
    <template>
        <li data-th-class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
            data-th-id="|observee*{id}|"
            data-th-object="${observee}">
            <div data-th-replace="~{fragments/observee-status-li :: observee-status-li}"></div>
        </li>
    </template>
</turbo-stream>
```

Headers | EventStream | Initiator | Timing | Cookies

| Id | Type | Data |
|---|---|---|
| 0 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 1 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 2 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 3 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 4 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 5 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 6 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 7 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 8 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 9 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 10 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 11 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 12 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |

# observee-status.turbo-stream.html

```html
<turbo-stream action="replace" data-th-each="observee : *{observees}" data-th-target="|observee${observee.id}|">
    <template>
        <li data-th-class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
            data-th-id="|observee*{id}|"
            data-th-object="${observee}">
            <div data-th-replace="~{fragments/observee-status-li :: observee-status-li}"></div>
        </li>
    </template>
</turbo-stream>
```

× Headers **EventStream** Initiator Timing Cookies

| Id | Type | Data |
|---|---|---|
| 0 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 1 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 2 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 3 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 4 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 5 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 6 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 7 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 8 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 9 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 10 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 11 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 12 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 13 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> |

# observee-status.turbo-stream.html

```html
<turbo-stream action="replace" data-th-each="observee : *{observees}" data-th-target="|observee${observee.id}|">
    <template>
        <li data-th-class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
            data-th-id="|observee*{id}|"
            data-th-object="${observee}">
            <div data-th-replace="~{fragments/observee-status-li :: observee-status-li}"></div>
        </li>
    </template>
</turbo-stream>
```

×
Headers
EventStream
Initiator
Timing
Cookies

| Id | Type | Data |
|---|---|---|
| 0 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 1 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 2 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 3 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 4 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 5 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 6 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 7 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 8 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 9 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |
| 10 | message | <turbo-stream action="replace" target="observee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class= |
| 11 | message | <turbo-stream action="replace" target="observee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="r |
| 12 | message | <turbo-stream action="replace" target="observee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="r |
| 13 | message | <turbo-stream action="replace" target="observee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class= |

## observee-status.turbo-stream.html

```html
<turbo-stream action="replace" data-th-each="observee : *{observees}" data-th-target="|observee${observee.id}|">
    <template>
        <li data-th-class="|my-1 py-3 d-flex list-group-item list-group-item-*{alert}|"
            data-th-id="|observee*{id}|"
            data-th-object="${observee}">
            <div data-th-replace="~{fragments/observee-status-li :: observee-status-li}"></div>
        </li>
    </template>
</turbo-stream>
```

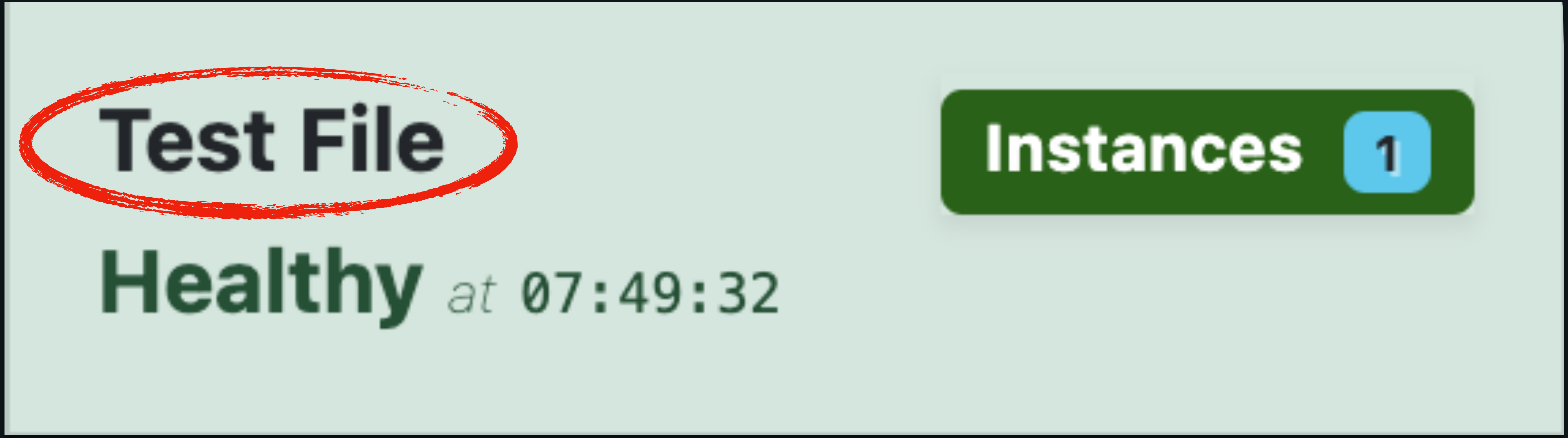
# observee-status-li.html

```html
<div class="container" data-th-fragment="observee-status-li">
    <div class="row">
        <div class="col px-0">
            <a class="fw-bold text-dark" data-th-href="*{location}" data-th-text="*{label}">
                An Observee
            </a>
        </div>
        <div class="col-auto px-0 shadow-sm">
            <span data-th-class="|badge *{color}|">Instances
                <span class="badge bg-info ms-1 text-shadow" data-th-text="*{count}">99</span>
            </span>
        </div>
    </div>
    <div class="row">
        <div class="col px-0">
            <span class="fw-bold" data-th-text="*{status}">Status</span>
            <span class="fw-lighter x-small fst-italic">at</span>
            <time class="fw-lighter x-small font-monospace"
                data-th-data-bounce="*{timestampBounce}" data-th-datetime="*{timestamp}"
                data-th-text="*{hhmmss}">HH:MM:SS
            </time>
        </div>
    </div>
</div>
```

```
ee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee187636958"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" hre
ee868023820"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee868023820"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href=
ee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href=
ee1170653260"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1170653260"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" hre
ee743535751"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee743535751"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" href=
ee1876369582"> <template> <li class="my-1 py-3 d-flex list-group-item list-group-item-success" id="observee1876369582"> <div class="container"> <div class="row"> <div class="col px-0"> <a class="fw-bold text-dark" hre
```

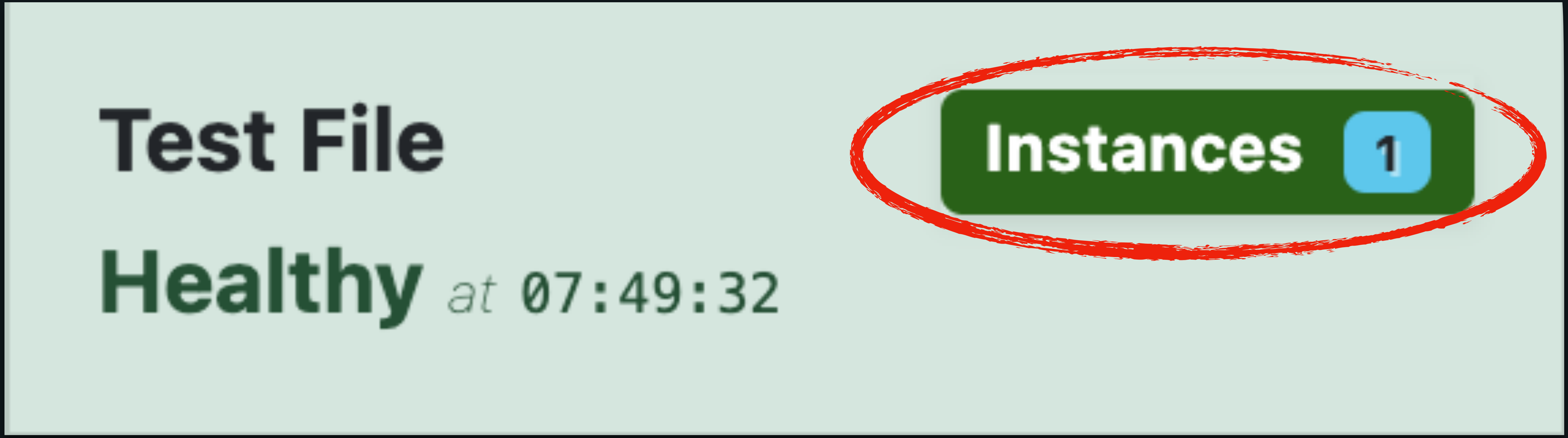
## observee-status-li.html

```
<div class="container" data-th-fragment="observee-status-li">
    <div class="row">
        <div class="col px-0">
            <a class="fw-bold text-dark" data-th-href="*{location}" data-th-text="*{label}">
                An Observee
            </a>
        </div>
        <div class="col-auto px-0 shadow-sm">
            <span data-th-class="|badge *{color}|">Instances
                <span class="badge bg-info ms-1 text-shadow" data-th-text="*{count}">99</span>
            </span>
        </div>
    </div>
    <div class="row">
        <div class="col px-0">
            <span class="fw-bold" data-th-text="*{status}">Status</span>
            <span class="fw-lighter x-small fst-italic">at</span>
            <time class="fw-lighter x-small font-monospace"
                data-th-data-bounce="*{timestampBounce}" data-th-datetime="*{timestamp}"
                data-th-text="*{hhmmss}">HH:MM:SS
            </time>
        </div>
    </div>
</div>
```



## observee-status-li.html

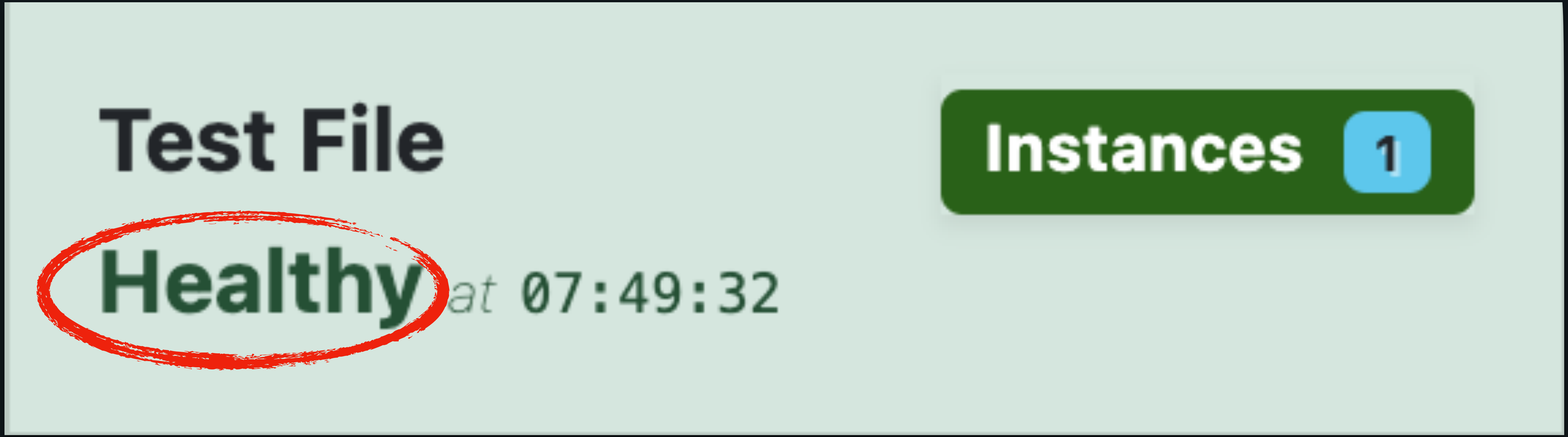
```html
<div class="container" data-th-fragment="observee-status-li">
    <div class="row">
        <div class="col px-0">
            <a class="fw-bold text-dark" data-th-href="*{location}" data-th-text="*{label}">
                An Observee
            </a>
        </div>
        <div class="col-auto px-0 shadow-sm">
            <span data-th-class="|badge *{color}|">Instances
                <span class="badge bg-info ms-1 text-shadow" data-th-text="*{count}">99</span>
            </span>
        </div>
    </div>
    <div class="row">
        <div class="col px-0">
            <span class="fw-bold" data-th-text="*{status}">Status</span>
            <span class="fw-lighter x-small fst-italic">at</span>
            <time class="fw-lighter x-small font-monospace"
                data-th-data-bounce="*{timestampBounce}" data-th-datetime="*{timestamp}"
                data-th-text="*{hhmmss}">HH:MM:SS</time>
        </div>
    </div>
</div>
```


Test File

Healthy *at* 07:49:32

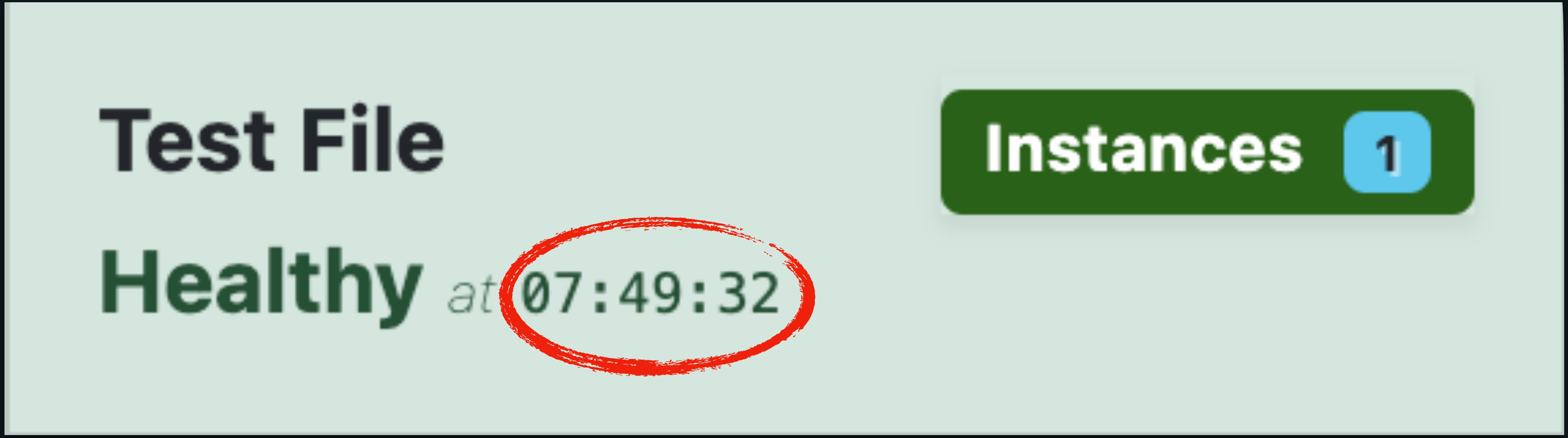
# observee-status-li.html

```html
<div class="container" data-th-fragment="observee-status-li">
    <div class="row">
        <div class="col px-0">
            <a class="fw-bold text-dark" data-th-href="*{location}" data-th-text="*{label}">
                An Observee
            </a>
        </div>
        <div class="col-auto px-0 shadow-sm">
            <span data-th-class="|badge *{color}|">Instances
                <span class="badge bg-info ms-1 text-shadow" data-th-text="*{count}">99</span>
            </span>
        </div>
    </div>
    <div class="row">
        <div class="col px-0">
            <span class="fw-bold" data-th-text="*{status}">Status</span>
            <span class="fw-lighter x-small fst-italic">at</span>
            <time class="fw-lighter x-small font-monospace"
                data-th-data-bounce="*{timestampBounce}" data-th-datetime="*{timestamp}"
                data-th-text="*{hhmmss}">HH:MM:SS
            </time>
        </div>
    </div>
</div>
```



# observee-status-li.html

```html
<div class="container" data-th-fragment="observee-status-li">
    <div class="row">
        <div class="col px-0">
            <a class="fw-bold text-dark" data-th-href="*{location}" data-th-text="*{label}">
                An Observee
            </a>
        </div>
        <div class="col-auto px-0 shadow-sm">
            <span data-th-class="|badge *{color}|">Instances
                <span class="badge bg-info ms-1 text-shadow" data-th-text="*{count}">99</span>
            </span>
        </div>
    </div>
    <div class="row">
        <div class="col px-0">
            <span class="fw-bold" data-th-text="*{status}">Status</span>
            <span class="fw-lighter x-small fst-italic">at</span>
            <time class="fw-lighter x-small font-monospace"
                data-th-data-bounce="*{timestampBounce}" data-th-datetime="*{timestamp}"
                data-th-text="*{hhmmss}">HH:MM:SS
            </time>
        </div>
    </div>
</div>
```



Test File

Instances 1

Healthy at 07:49:32

## Spring WebFlux

**Reactive web stack**

- Asynchronous, non-blocking, reactive back-pressure
- Concurrency with a small number of threads
- Scale with fewer hardware resources
- Functional programming model (Java 8+)
- Adapts to various Reactive Streams (Java 9+) libraries and Coroutines (Kotlin).

## Kotlin Coroutines & Kotlin Flows

**My name is not “Brian Goetz”, “Doug Lea”, “Erik Meijer”, “Viktor Klang”, “Jonas Boner”, …**

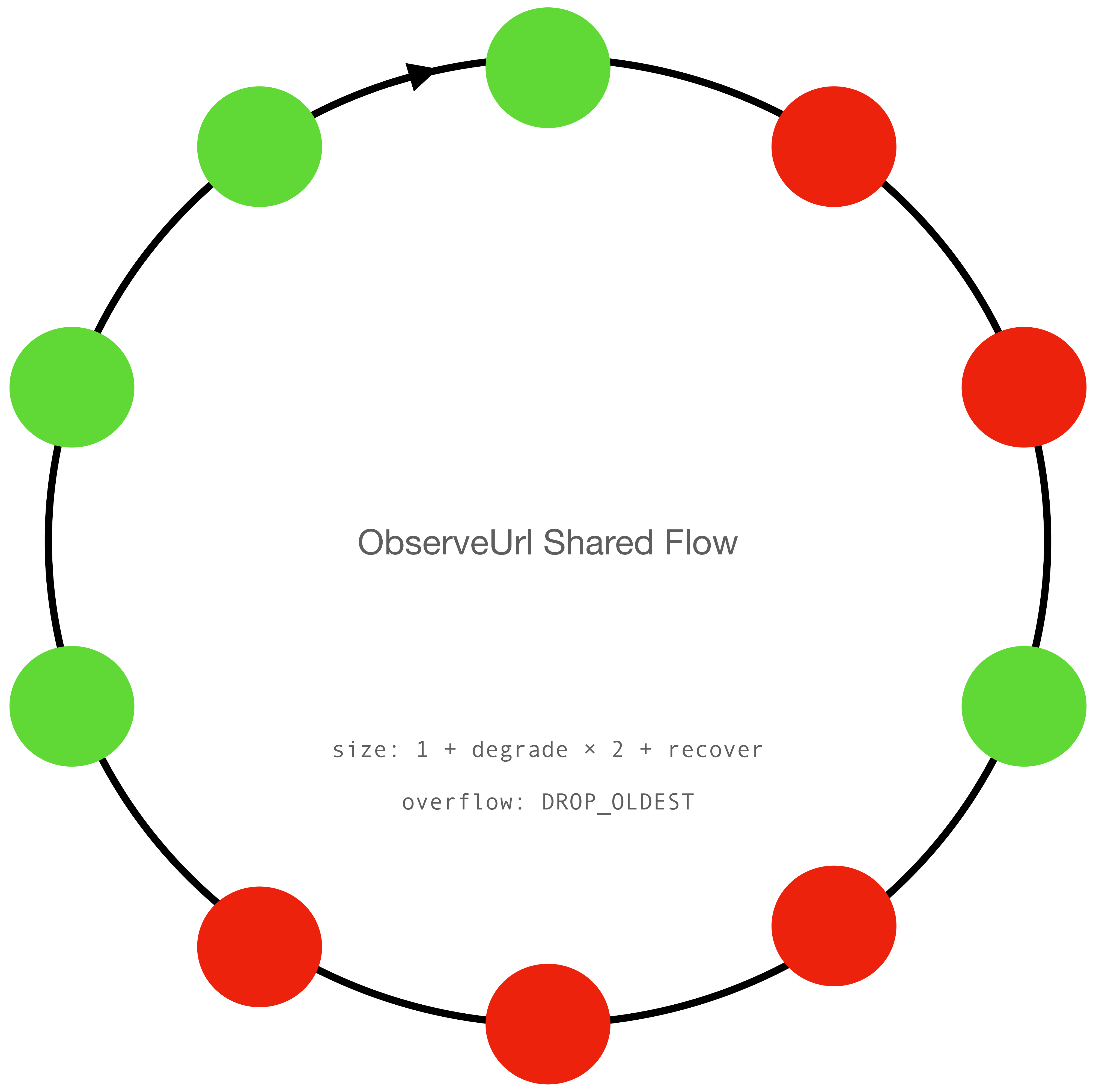
- Lightweight threads + Structured concurrency
- *Scoped* asynchronous operations and *scoped* parallel decomposition
- Write non-blocking, functional code in an imperative style
- Flow is Flux equivalent(-ish)
- Suitable for hot or cold, finite or infinite streams
    - Push-based
    - Suspending functions handle back-pressure
    - Easily add your own operators (coroutines)
    - `map` supports asynchrony (suspending function parameter).

## 3 domain model Flows, 1 view model Flow

- Domain model
  - Shared Flow per Instance: was “ping” Result an Error or Success?
  - State Flow per Instance: calculate health based on recent Results
  - State Flow per Observee: calculate overall health based on instance(s) state
- View model
  - Flow of each observee’s current state: interpolated into template

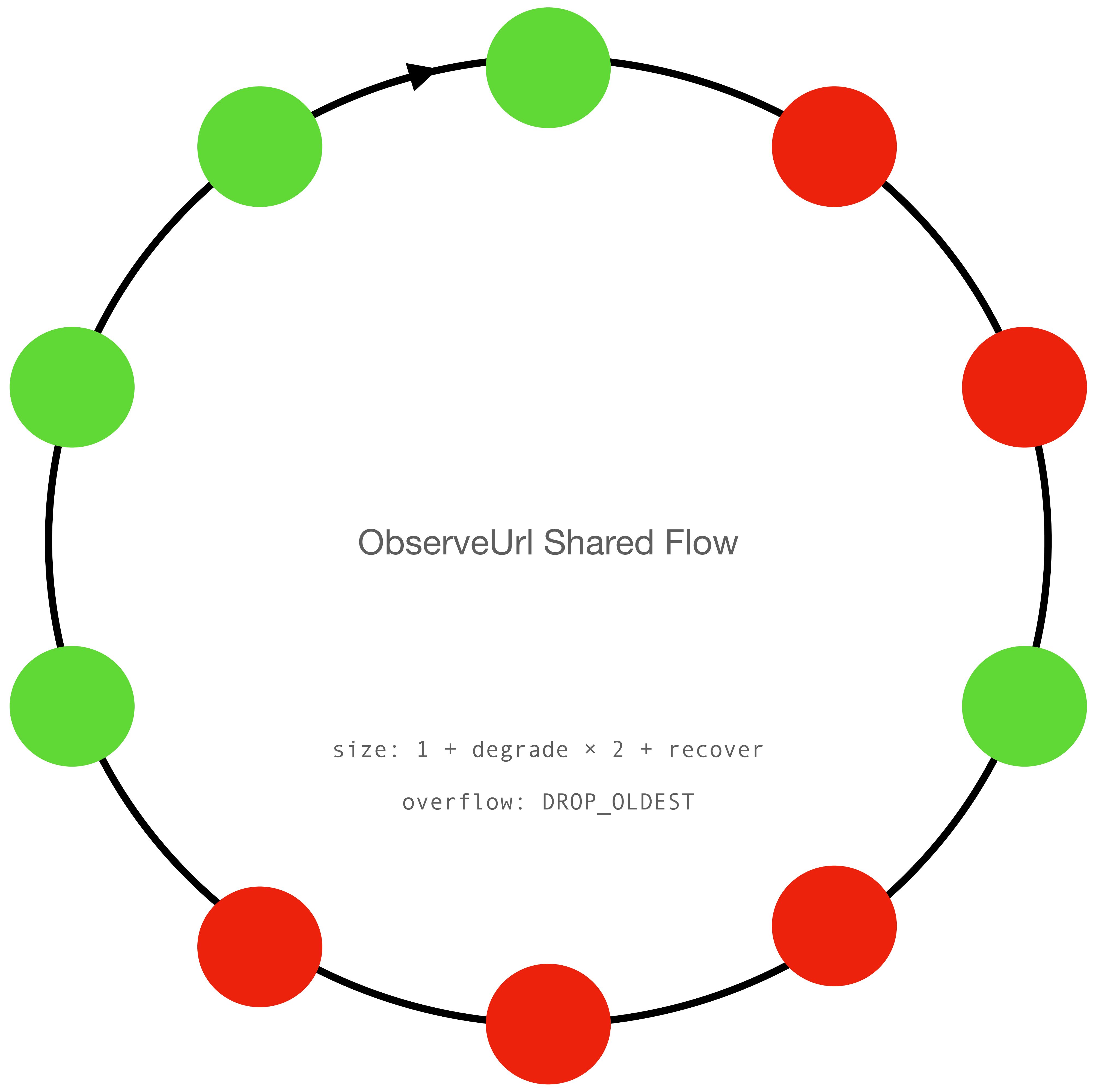
## ObserveUrl shared flow

**Acts like a Ring Buffer which doesn't wait for reads**



## ObserveUrl shared flow

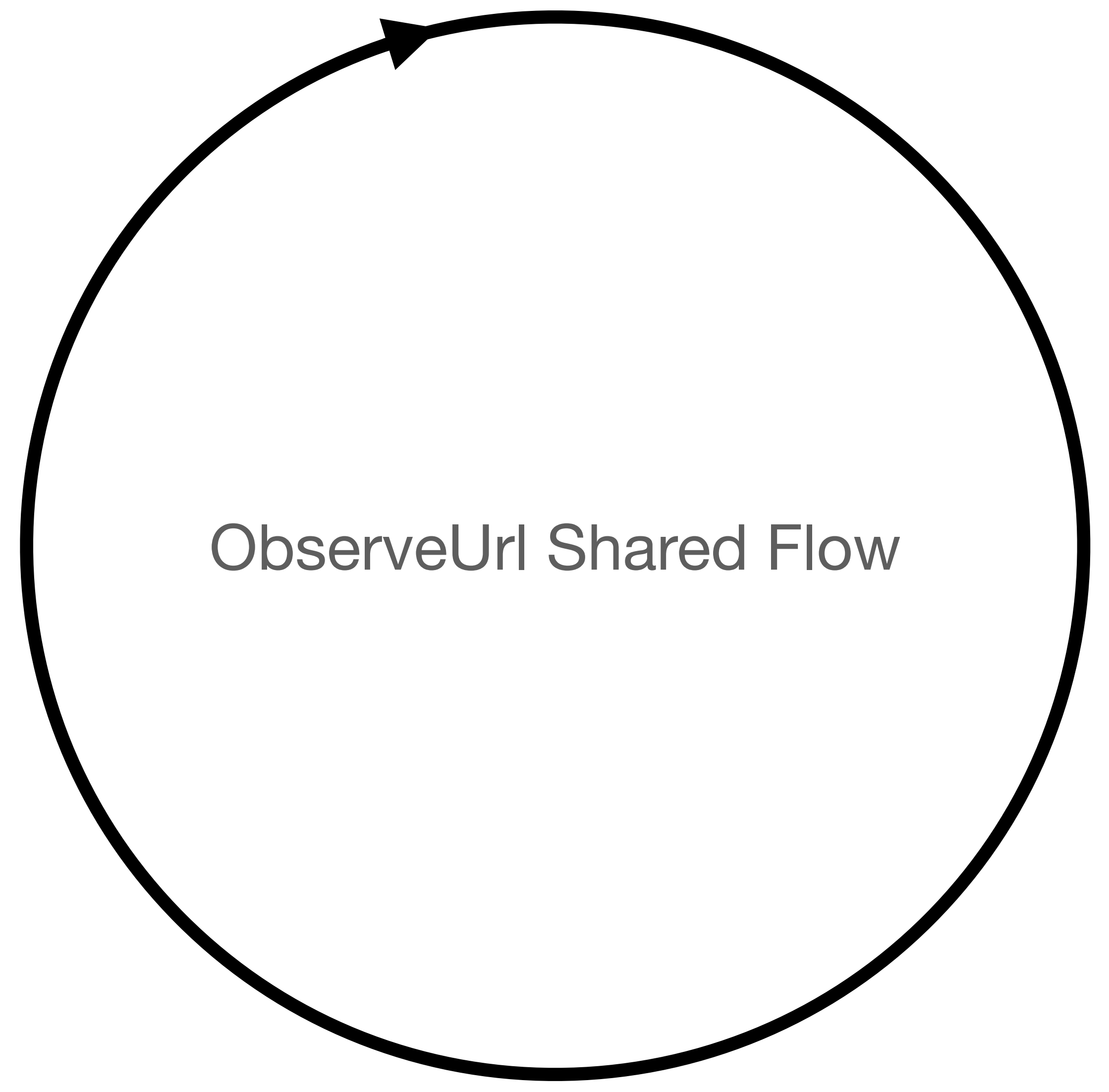
### Acts like a Ring Buffer which doesn't wait for reads



- Result = Either<Error, Success>
- Error
  - HTTP status
  - Timeout
  - Exception
    (typically IOException)
- Blocking check in IO dispatcher
- Checked every "period" by a coroutine

# Instance Health state flow

## Calculated from ObserveUrl shared flow

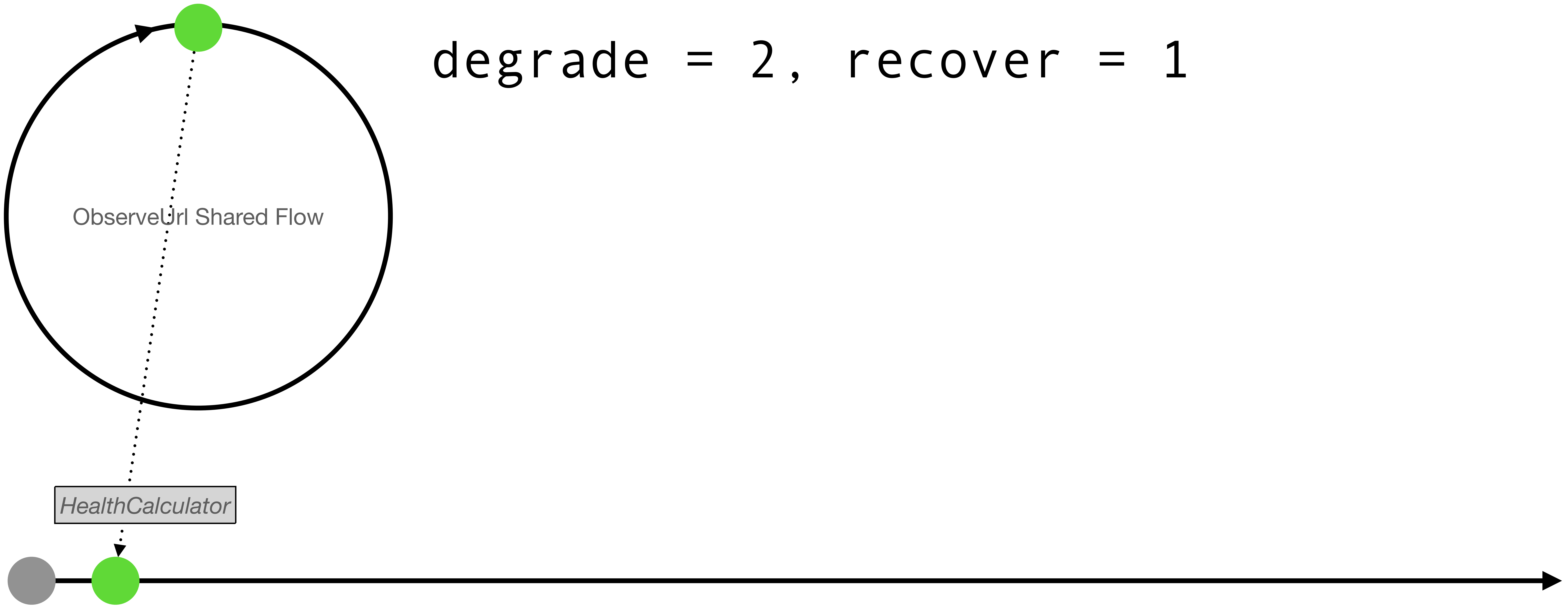


```
degrade = 2, recover = 1
```

The State Flow has an initial state of Unknown health

# Instance Health state flow

## Calculated from ObserveUrl shared flow

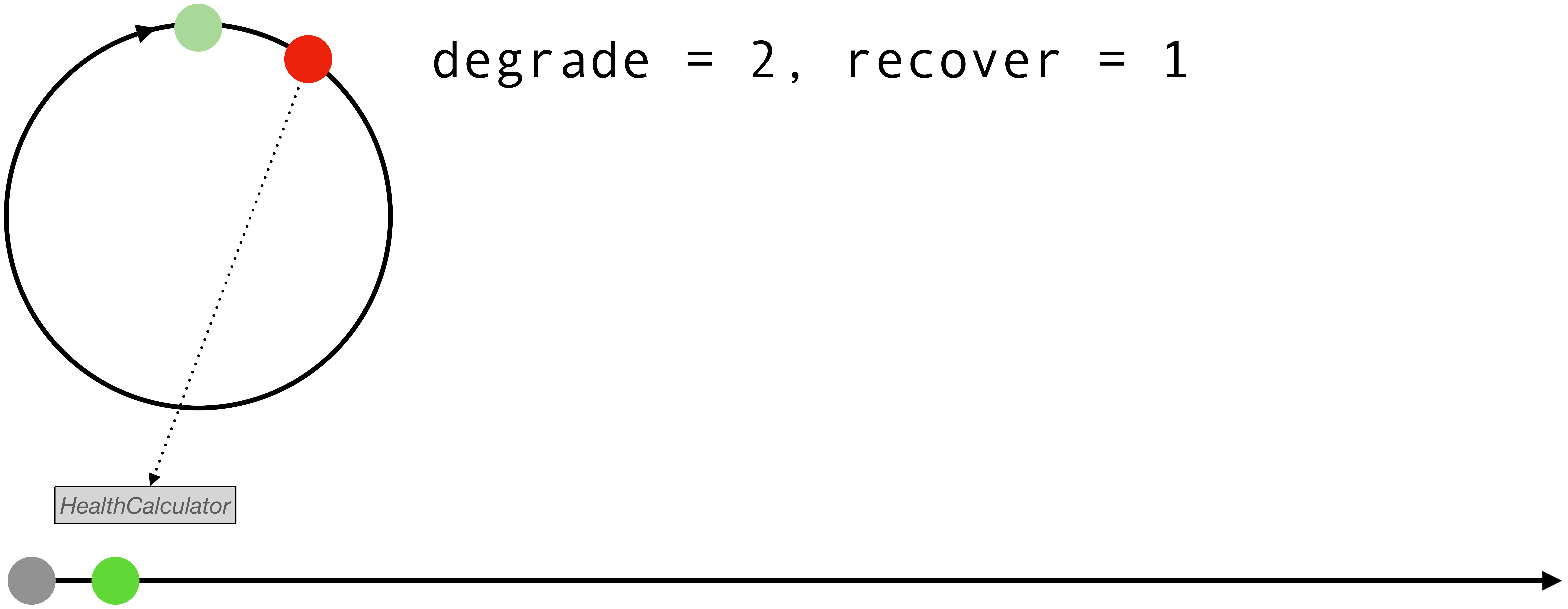


```
degrade = 2, recover = 1
```

The first ObserveUrl shared flow event is emitted, and a calculation is performed to determine that the Instance Health state is Healthy

# Instance Health state flow

## Calculated from ObserveUrl shared flow

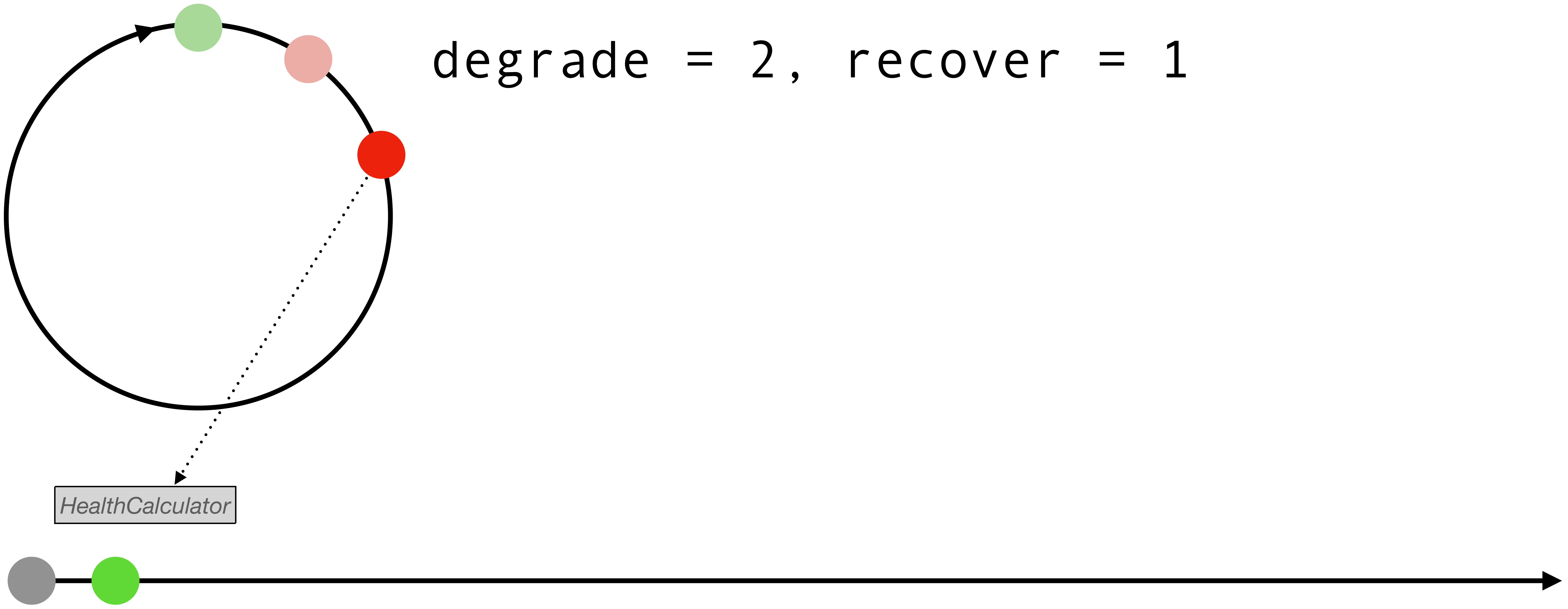


```
degrade = 2, recover = 1
```

Because “degrade after” is 2, the instance is still calculated as Healthy
Because there is no change in health, no event is emitted to the State Flow

# Instance Health state flow

## Calculated from ObserveUrl shared flow

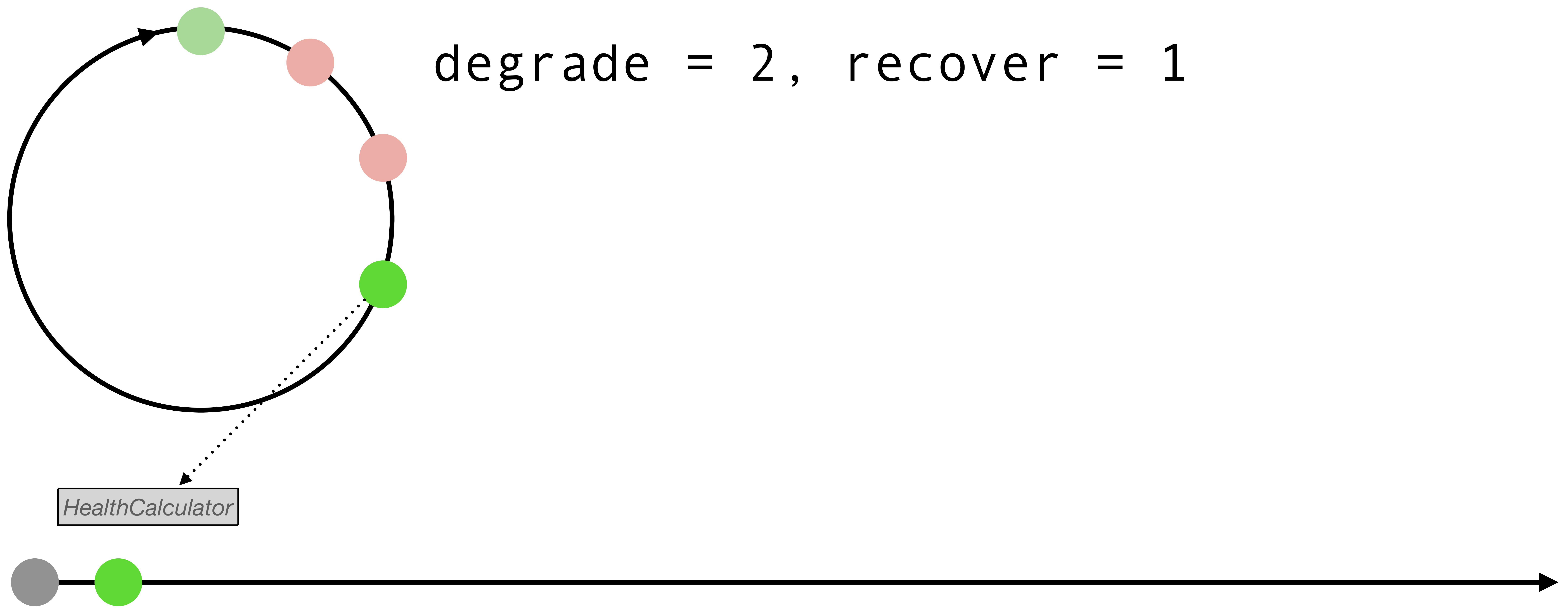


```
degrade = 2, recover = 1
```

Because degrade is 3, the instance is still calculated as Healthy
Because there is no change in health, no event is emitted to the State Flow

# Instance Health state flow

## Calculated from ObserveUrl shared flow

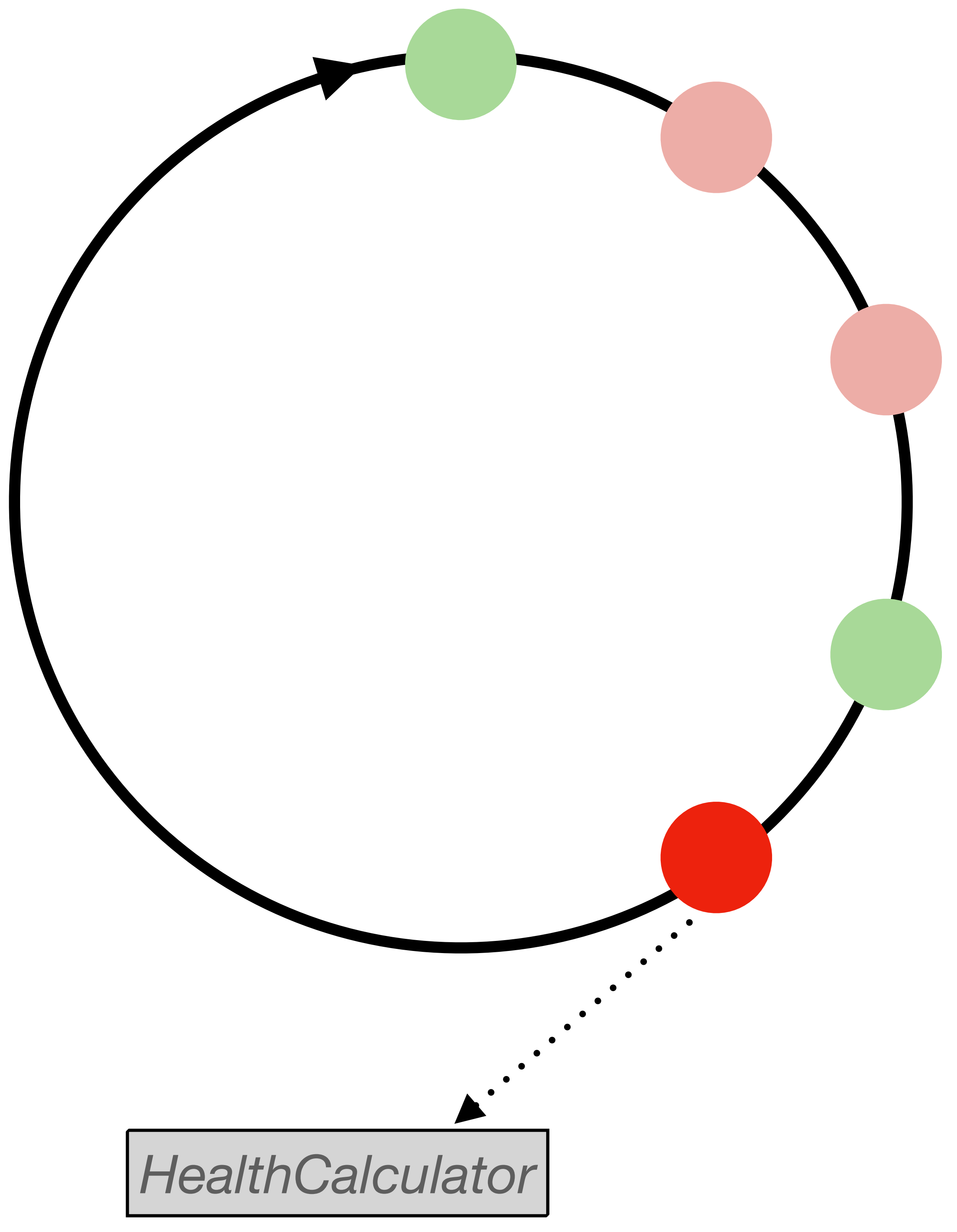


```
degrade = 2, recover = 1
```

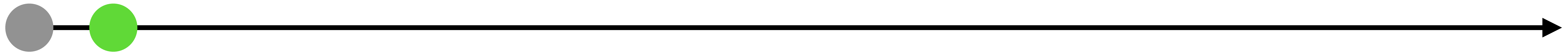
Because there is no change in health, no event is emitted to the State Flow

# Instance Health state flow

## Calculated from ObserveUrl shared flow



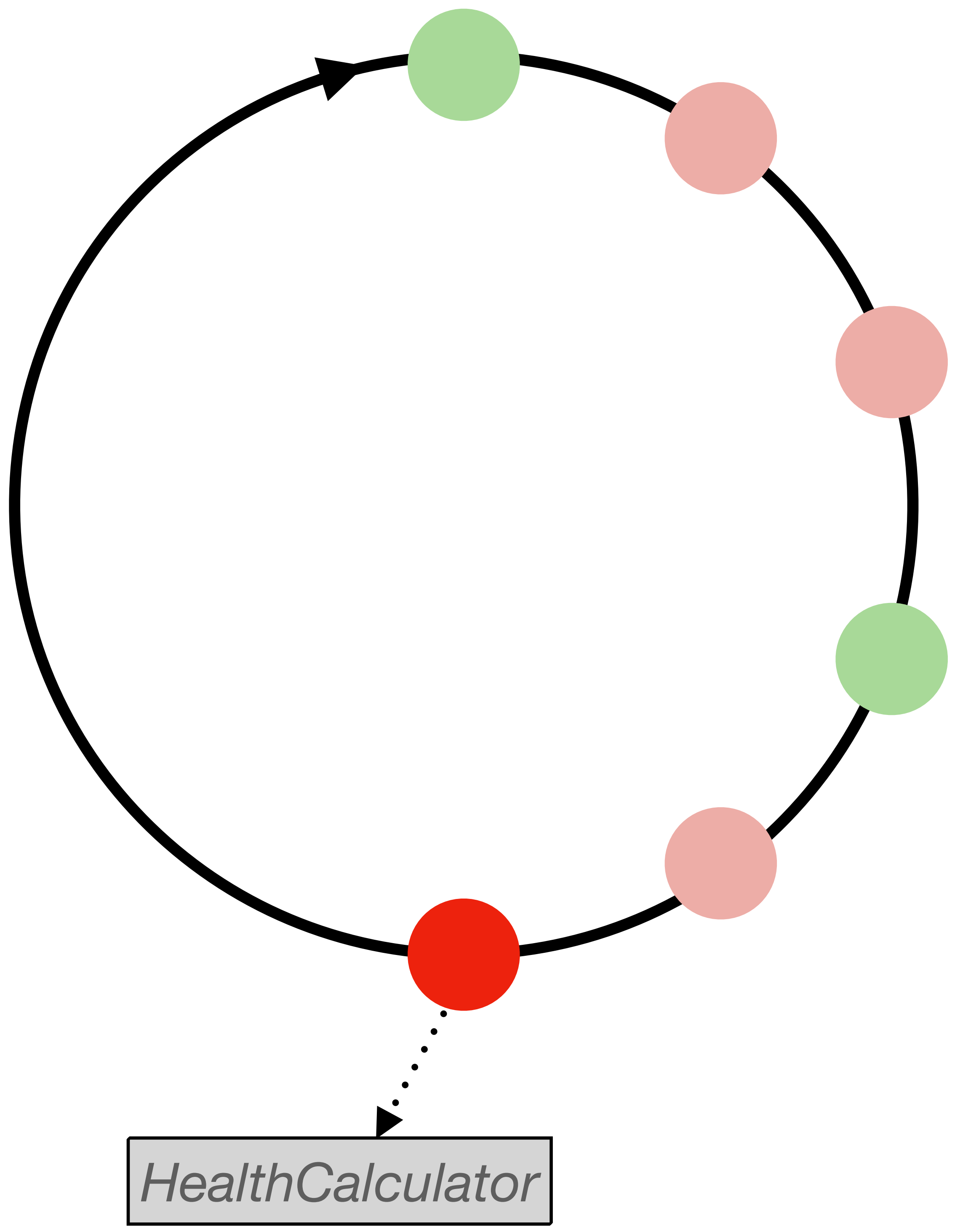
```
degrade = 2, recover = 1
```



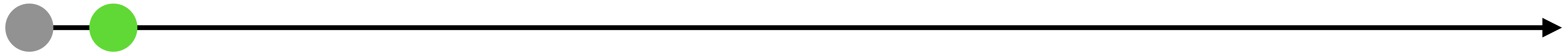
Because "degrade after" is 2, the instance is still calculated as Healthy
Because there is no change in health, no event is emitted to the State Flow

# Instance Health state flow

## Calculated from ObserveUrl shared flow



```
degrade = 2, recover = 1
```



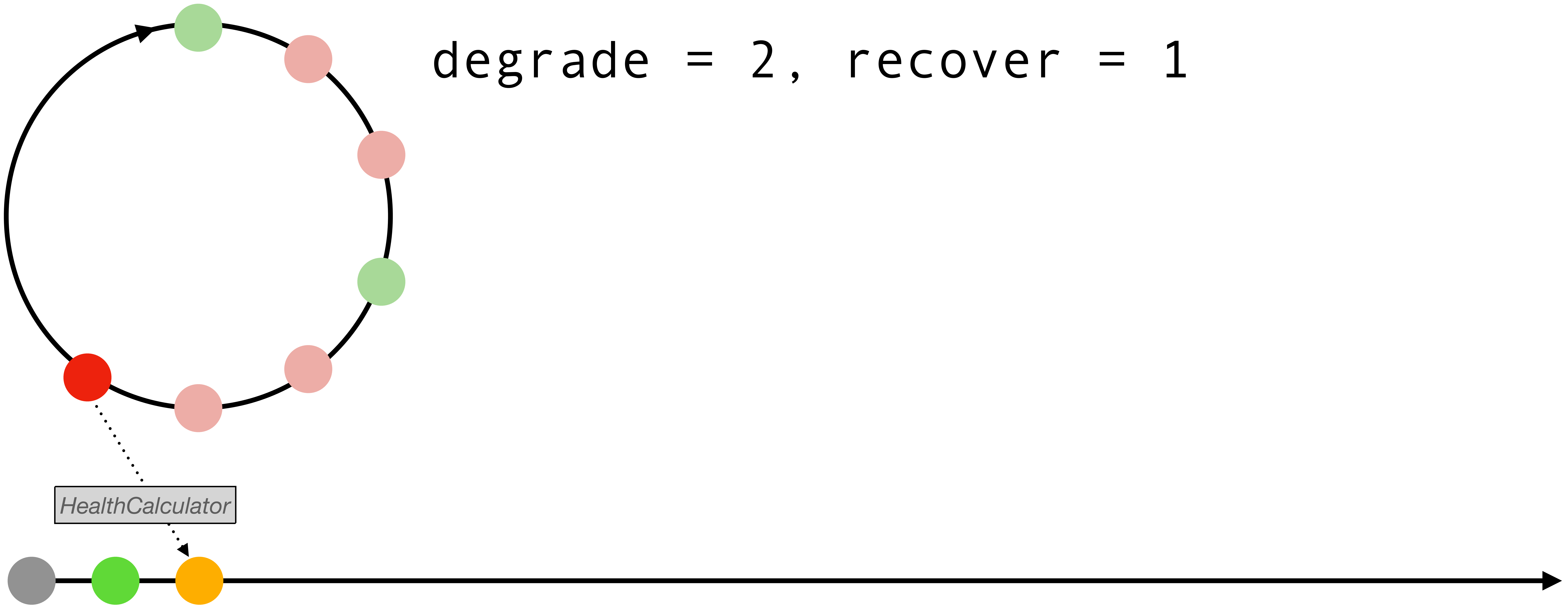
Because “degrade after” is 2, the instance is still calculated as Healthy
Because there is no change in health, no event is emitted to the State Flow

** The ring size is larger than needed to help visualize this example*

# Instance Health state flow

## Calculated from ObserveUrl shared flow

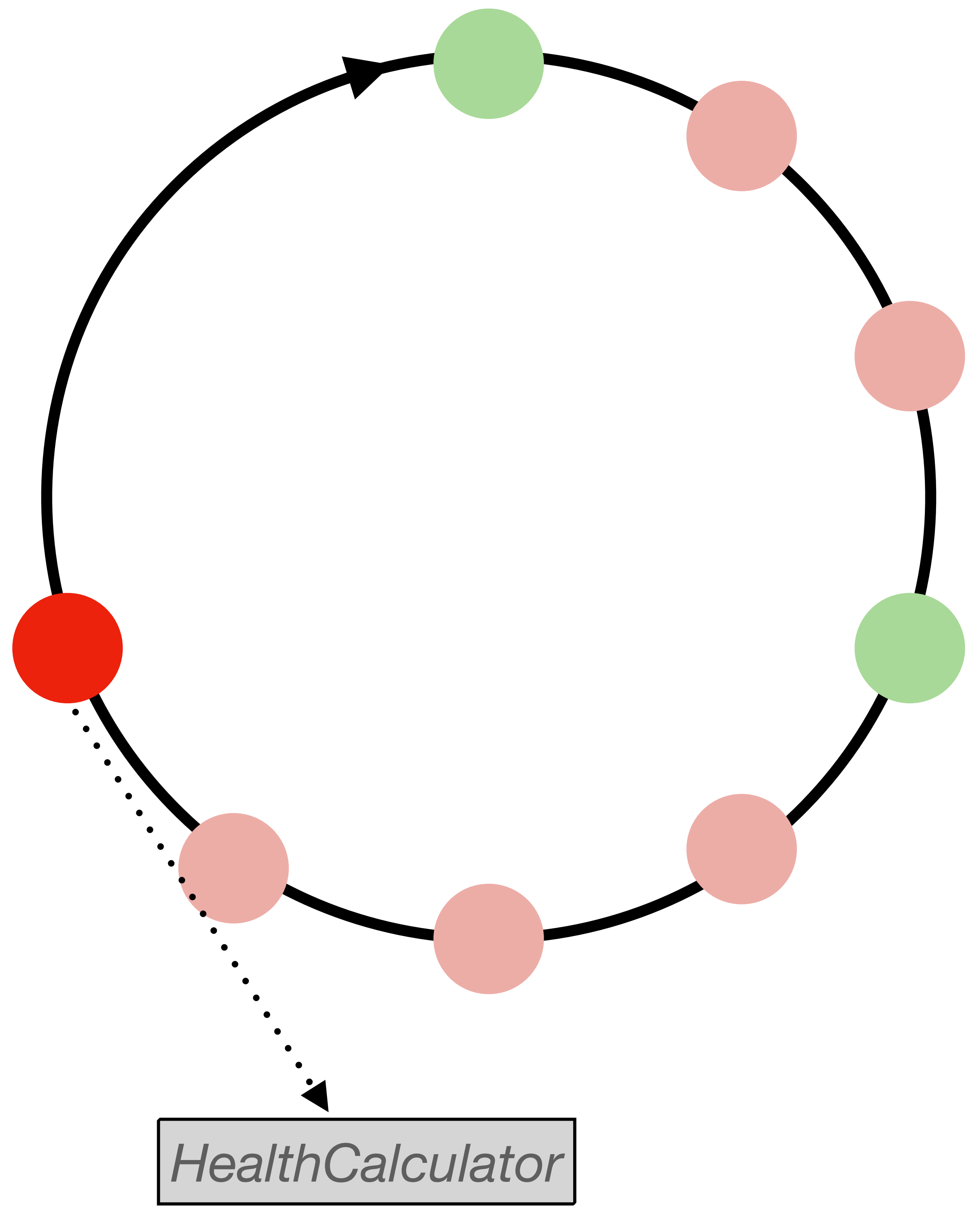
`degrade = 2, recover = 1`



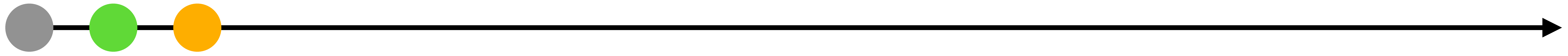
Because “degrade after” is 2, the instance is now calculated as Ailing
Because there is a change in health, an event is emitted to the State Flow

# Instance Health state flow

## Calculated from ObserveUrl shared flow



```
degrade = 2, recover = 1
```

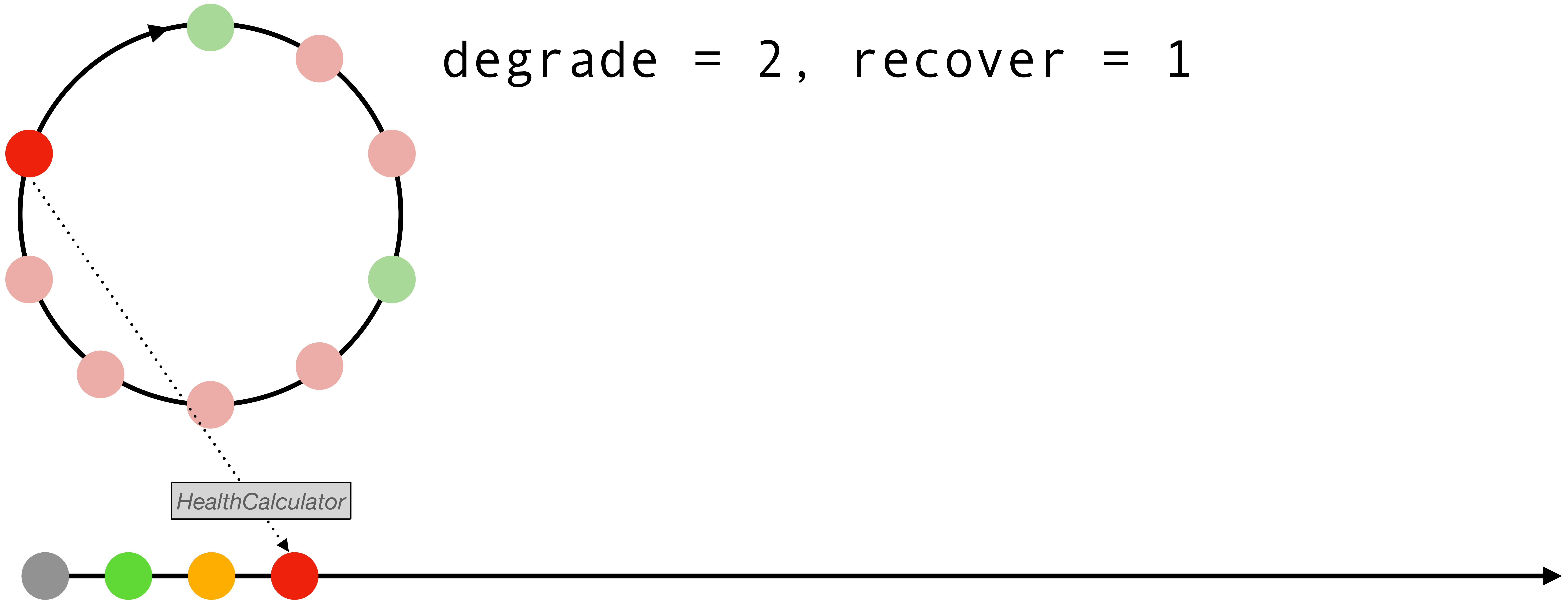


Because “degrade after” is 2, the instance is still calculated as Ailing
Because there is no change in health, no event is emitted to the State Flow

# Instance Health state flow

## Calculated from ObserveUrl shared flow

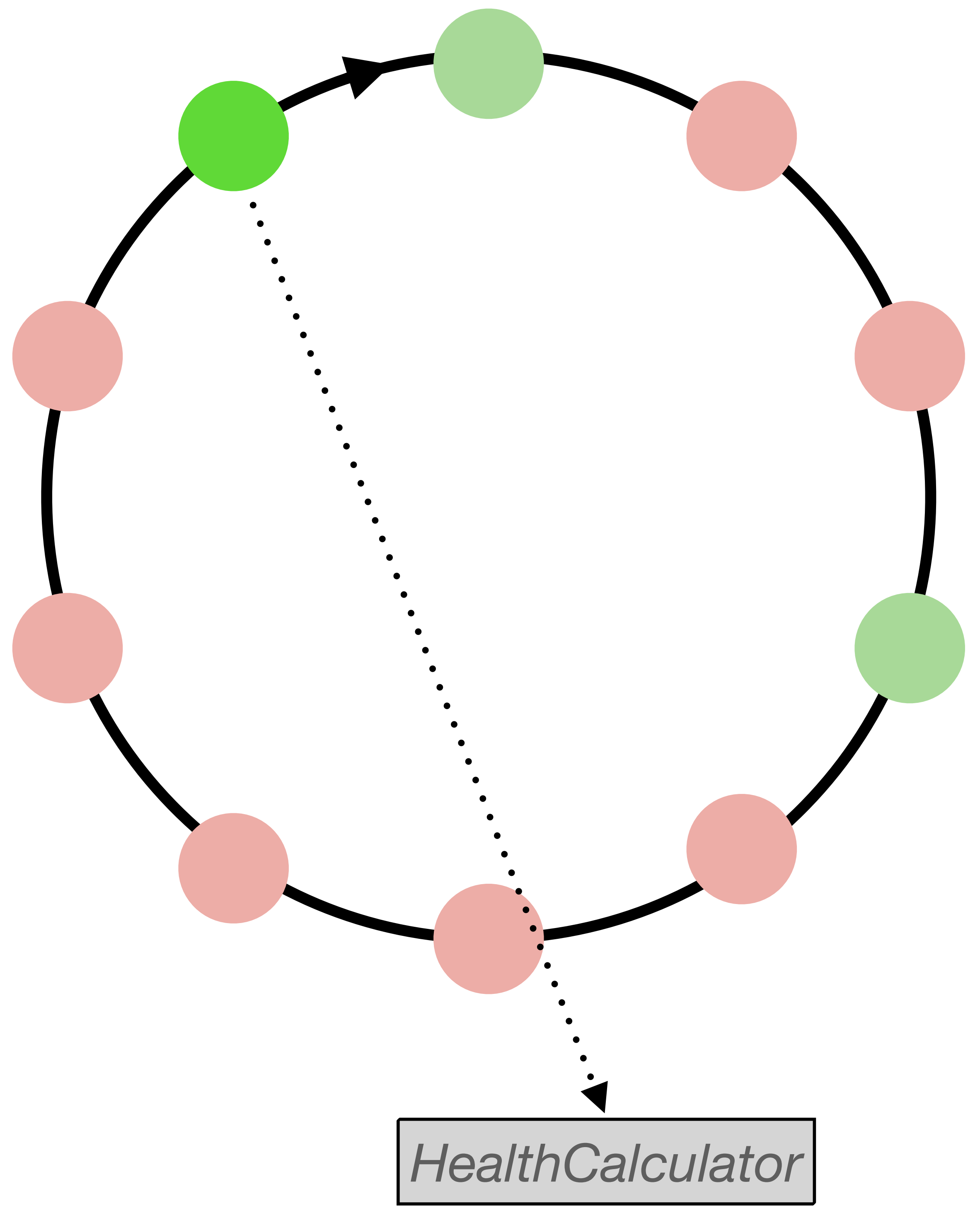
```
degrade = 2, recover = 1
```



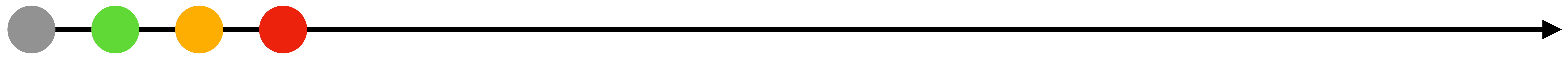
Because “degrade after” is 2, the instance is now calculated as Unhealthy
Because there is a change in health, an event is emitted to the State Flow

# Instance Health state flow

## Calculated from ObserveUrl shared flow



```
degrade = 2, recover = 1
```

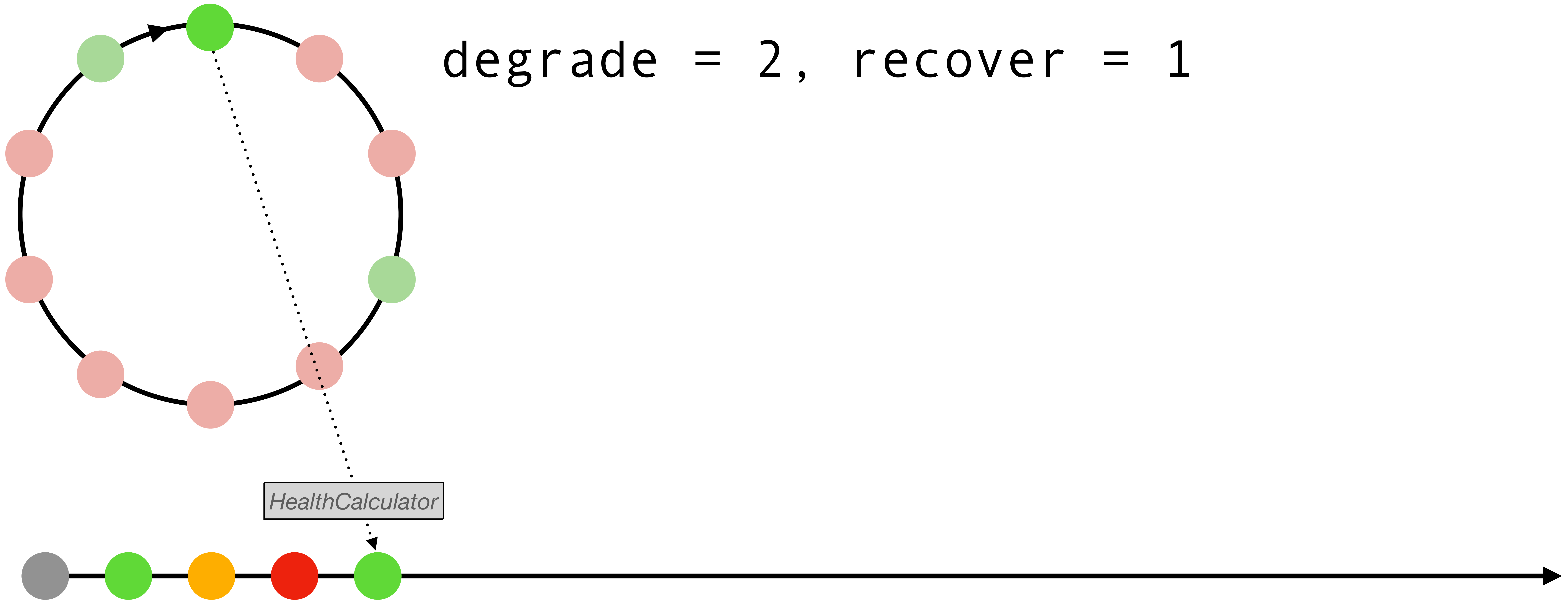


Because “recover after” is 1, the instance is still calculated as Unhealthy
Because there is a change in health, an event is emitted to the State Flow

## Instance Health state flow

### Calculated from ObserveUrl shared flow

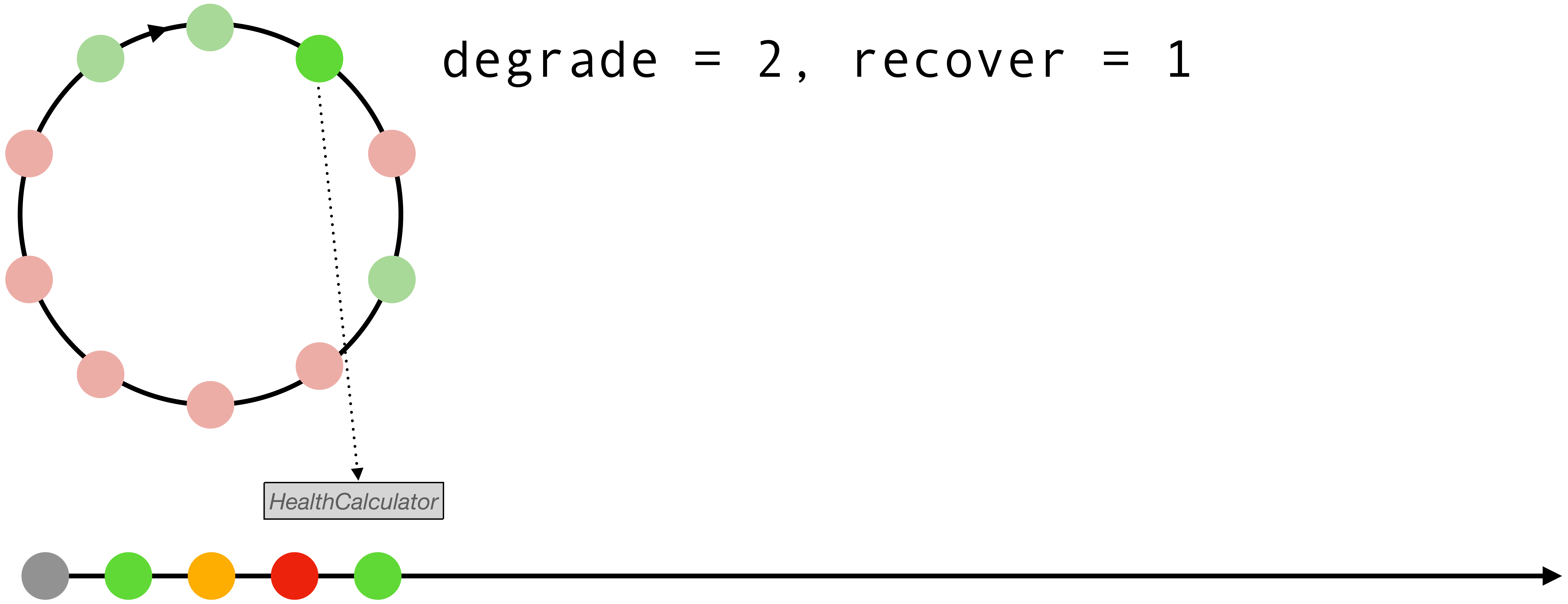
```
degrade = 2, recover = 1
```



Because “recover after” is 1, the instance is now calculated as Healthy
Because there is a change in health, an event is emitted to the State Flow

# Instance Health state flow

## Calculated from ObserveUrl shared flow

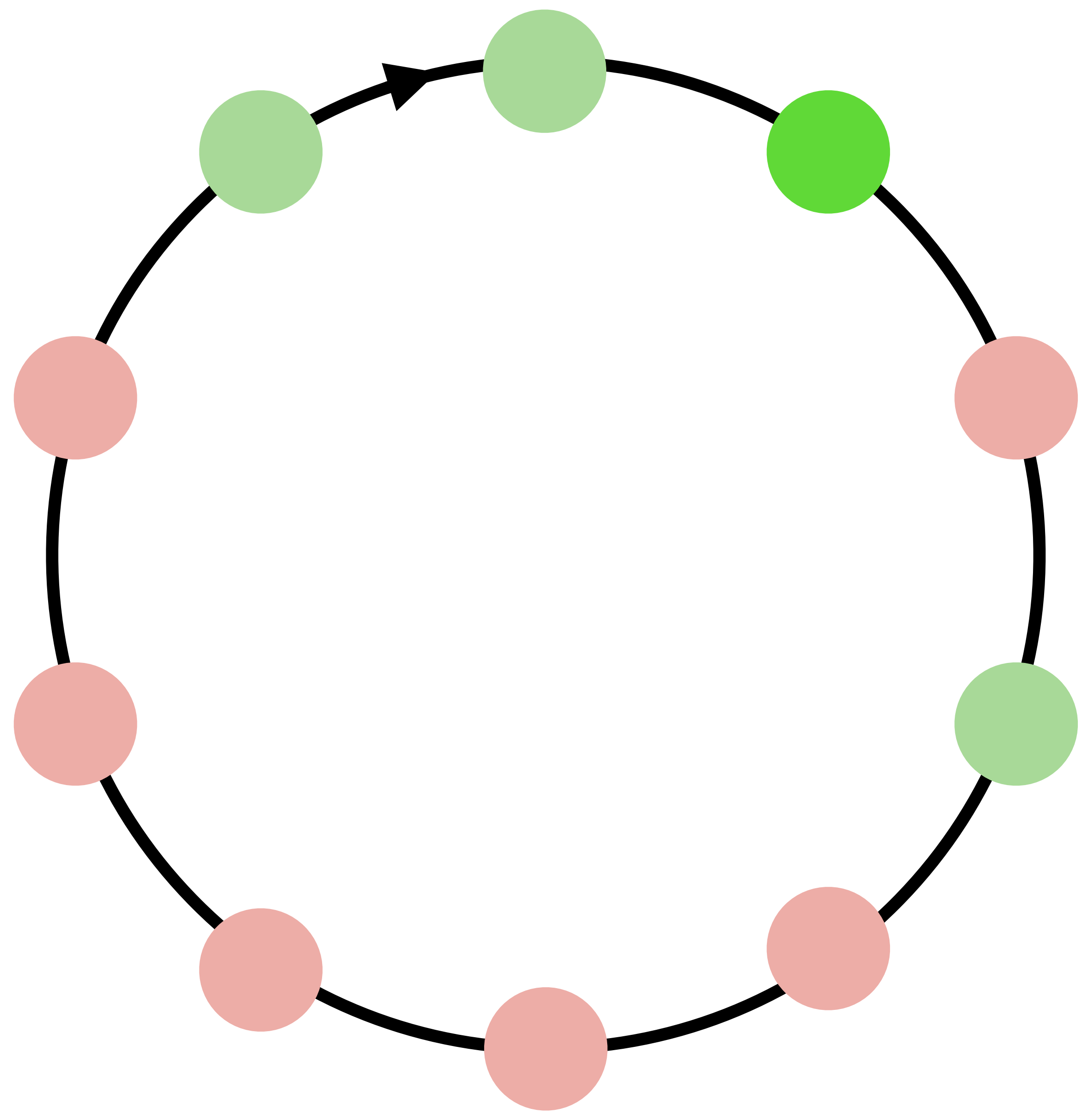


```
degrade = 2, recover = 1
```

Because there is no change in health, no event is emitted to the State Flow

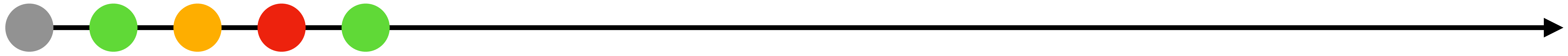
# Instance Health state flow

## Calculated from ObserveUrl shared flow



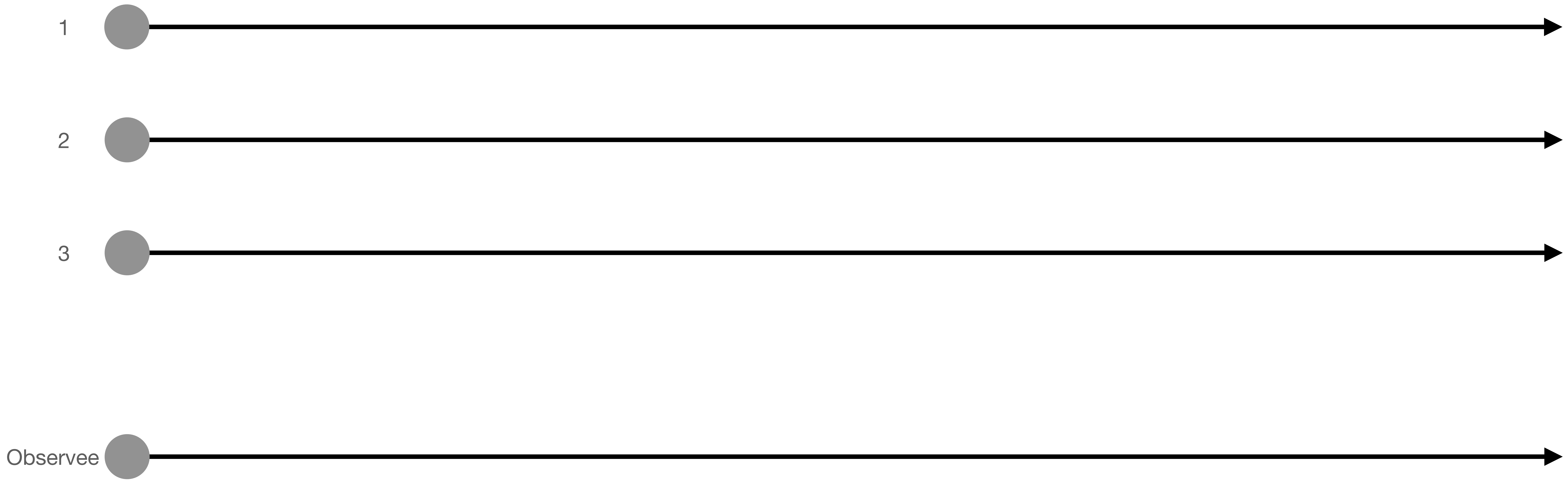
```
degrade = 2, recover = 1
```

And now, multiple State Flows feeding another State Flow



# Observee Health state flow

## Calculated from combined Instance Health state flows

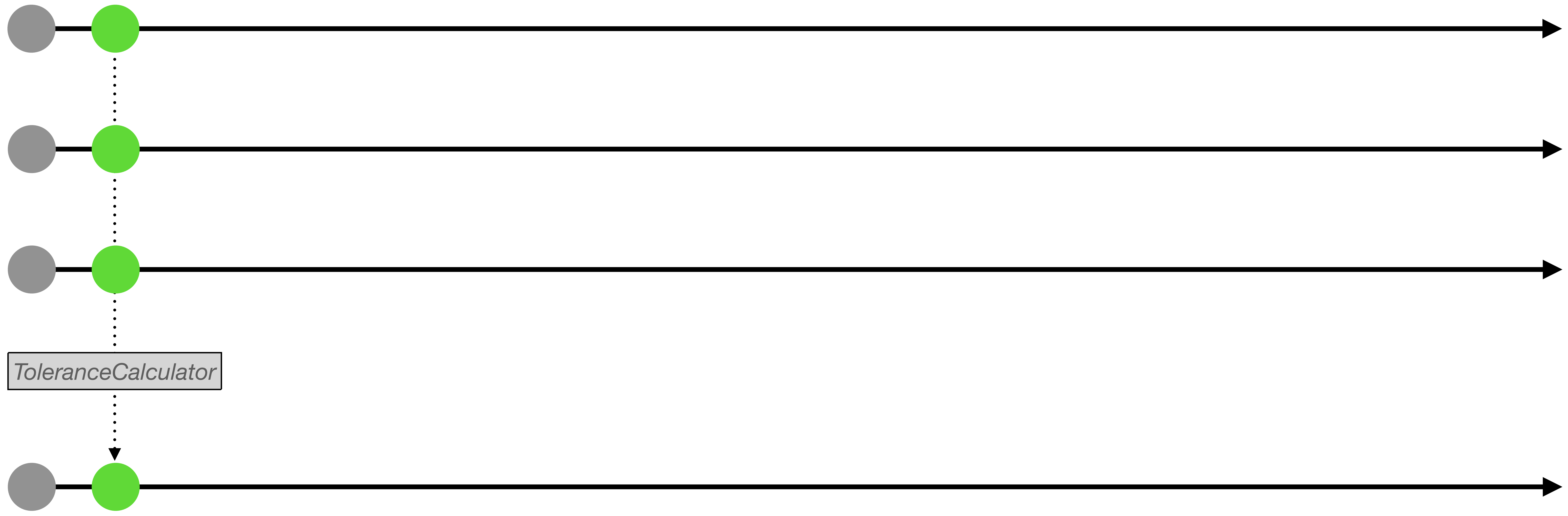


```
tolerance = 1
```

The State Flow has an initial state of Unknown health

# Observee Health state flow

## Calculated from combined Instance Health state flows

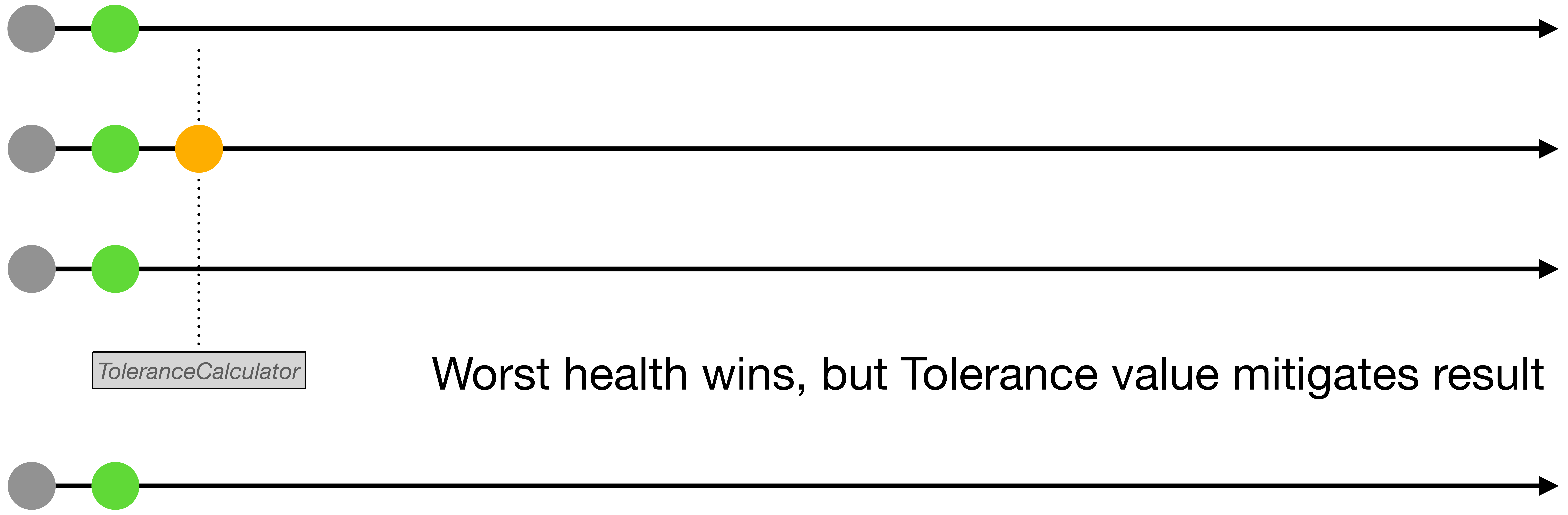


```
tolerance = 1
```

The first Instance Health state flow events are emitted, and a calculation is performed to determine that the Observee Health state is Healthy

# Observee Health state flow

## Calculated from combined Instance Health state flows



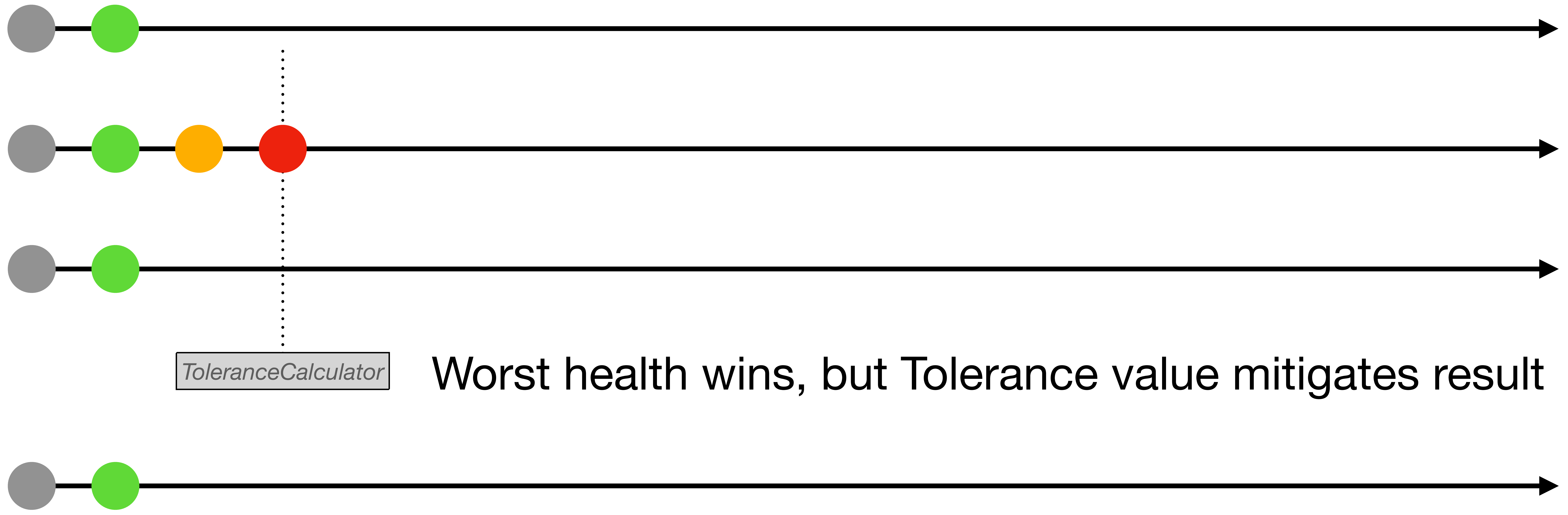
Worst health wins, but Tolerance value mitigates result

```
tolerance = 1
```

Because 1 not-healthy instance is tolerated, the system is still calculated as Healthy
Because there is no change in health, no event is emitted to the State Flow

# Observee Health state flow

## Calculated from combined Instance Health state flows



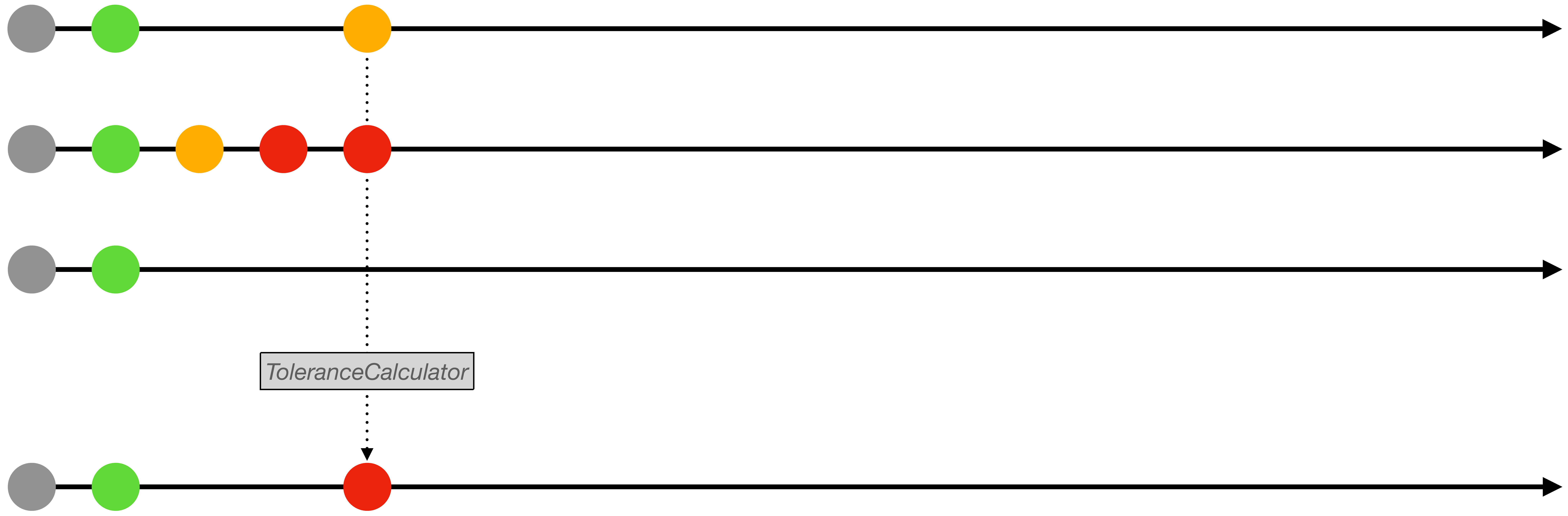
Worst health wins, but Tolerance value mitigates result

```
tolerance = 1
```

Because 1 not-healthy instance is tolerated, the system is still calculated as Healthy
Because there is no change in health, no event is emitted to the State Flow

# Observee Health state flow

## Calculated from combined Instance Health state flows

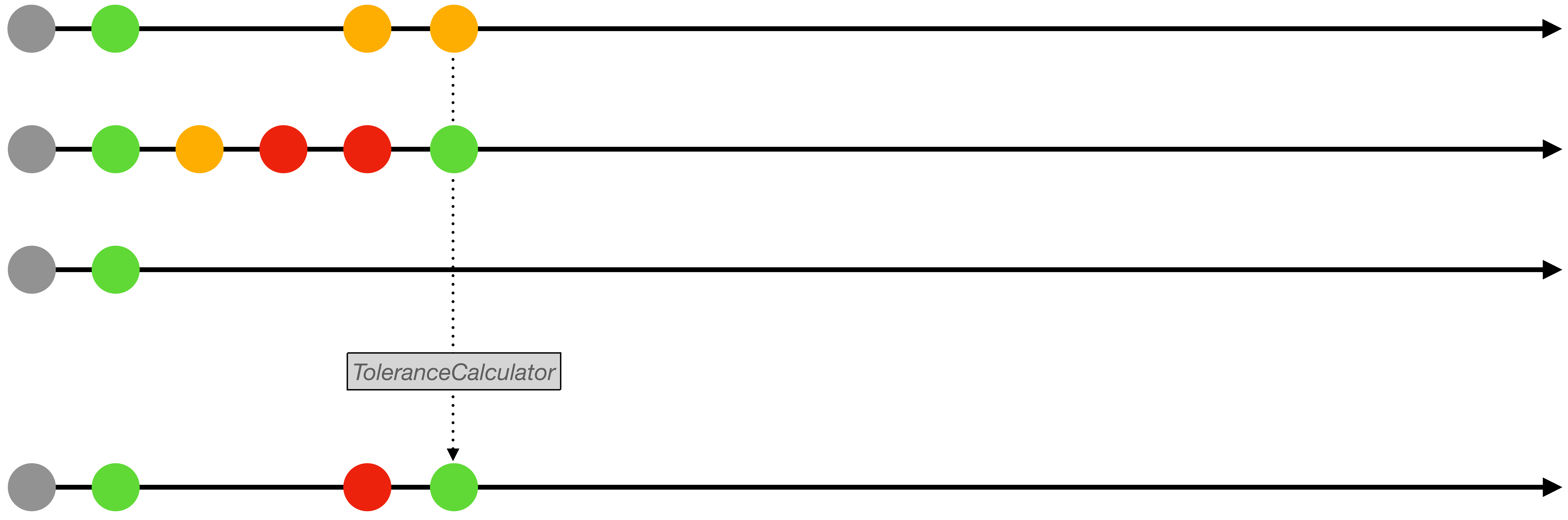


```
tolerance = 1
```

Because only 1 not-healthy instance is tolerated, and the worst not-healthy instance is Unhealthy, the system is calculated as Unhealthy
Because there is a change in health, an event is emitted to the State Flow

# Observee Health state flow

## Calculated from combined Instance Health state flows

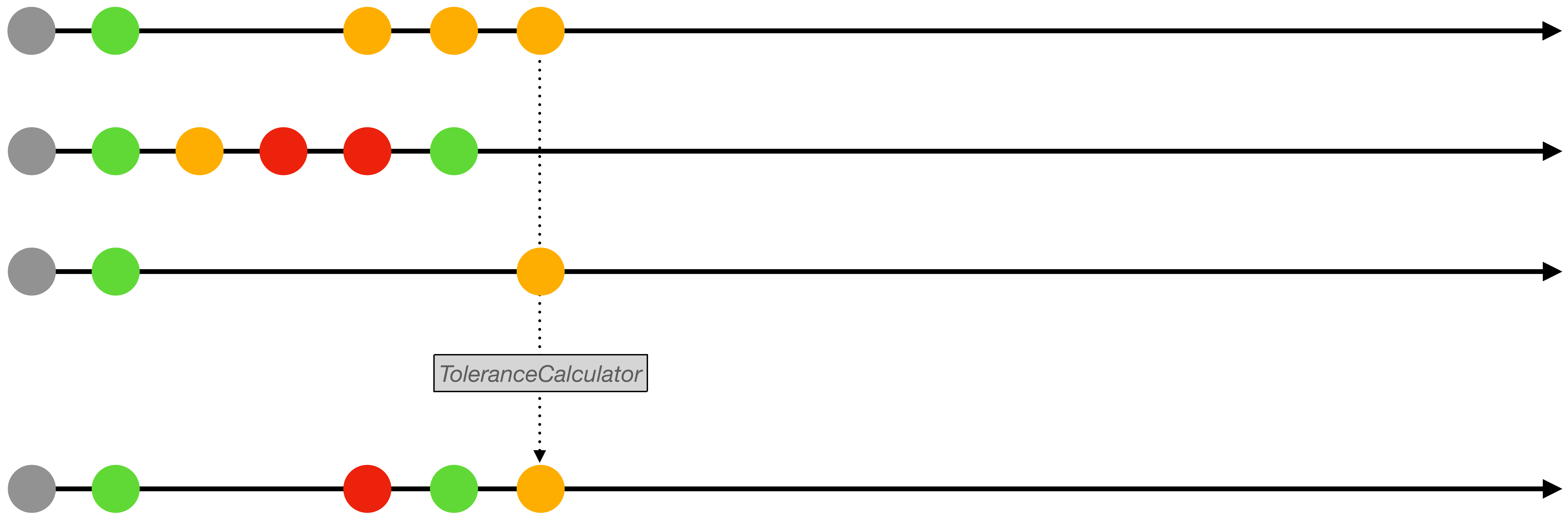


```
tolerance = 1
```

Because 1 not-healthy instance is tolerated, the system is now calculated as Healthy
Because there is a change in health, an event is emitted to the State Flow

# Observee Health state flow

## Calculated from combined Instance Health state flows

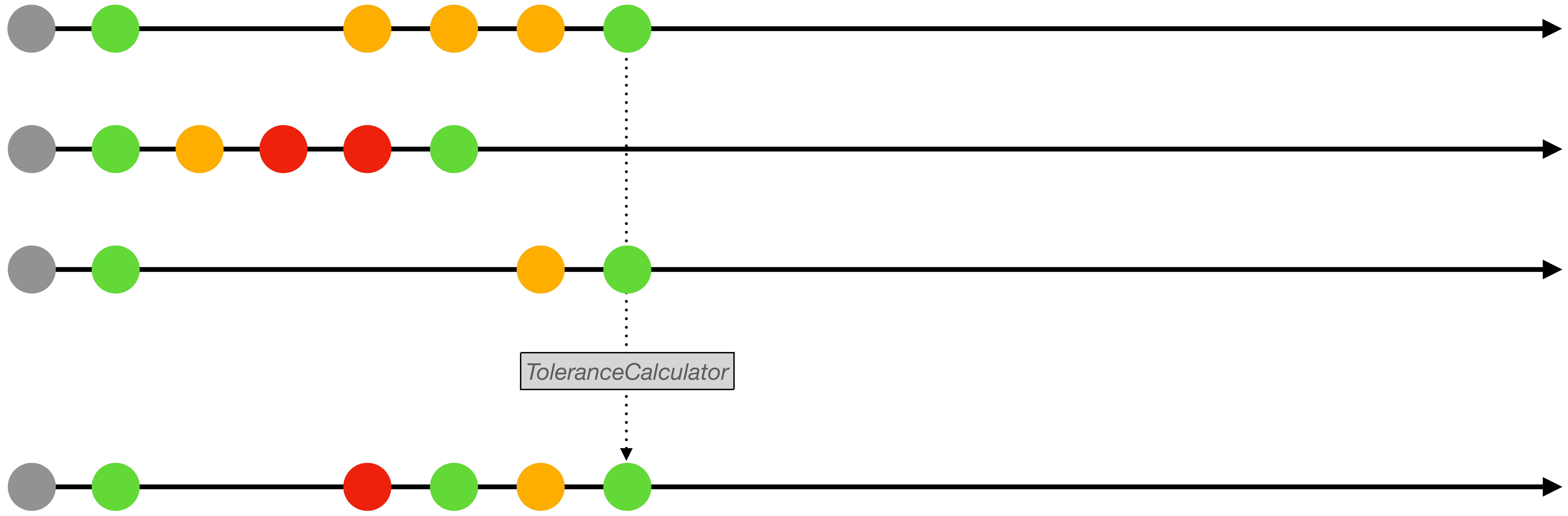


```
tolerance = 1
```

Because only 1 not-healthy instance is tolerated, and the worst not-healthy instance is Ailing, the system is calculated as Ailing
Because there is a change in health, an event is emitted to the State Flow

# Observee Health state flow

## Calculated from combined Instance Health state flows

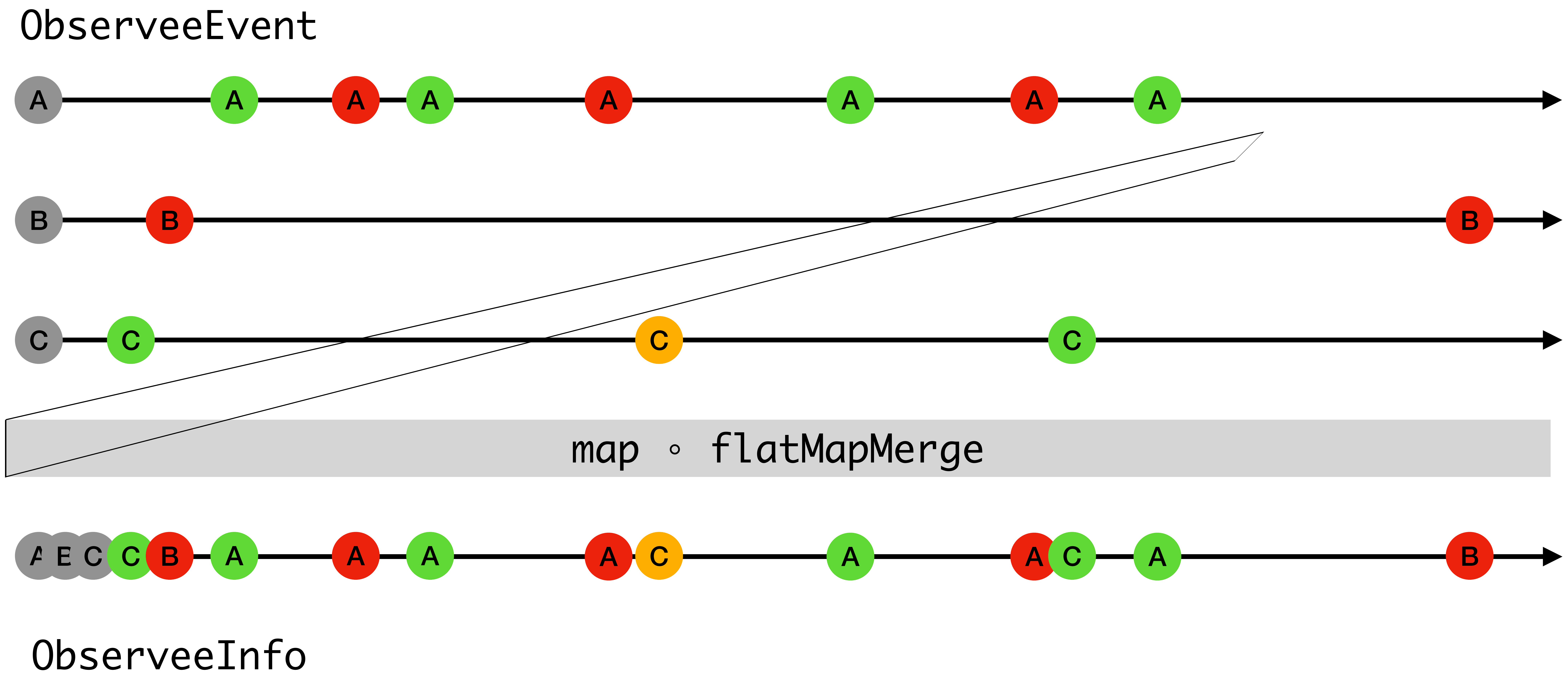


tolerance = 1

Because all instances are healthy, the system is now calculated as Healthy
Because there is a change in health, an event is emitted to the State Flow

/status.stream

Flow<Flow<ObserveeEvent>> → merge → map → Flow<ObserveeInfo>



## Merging Flows

**Turning Kotlin Flows into text/event-stream for Turbo Stream**

- Server-side reactive stream of domain model events
- Functional transformations
- Mapped to reactive stream of view model events
- Template interpolation
- Sent to browser as SSE messages
- Each event message contains HTML data
- DOM updated directly (no JSON deserialization & mapping).

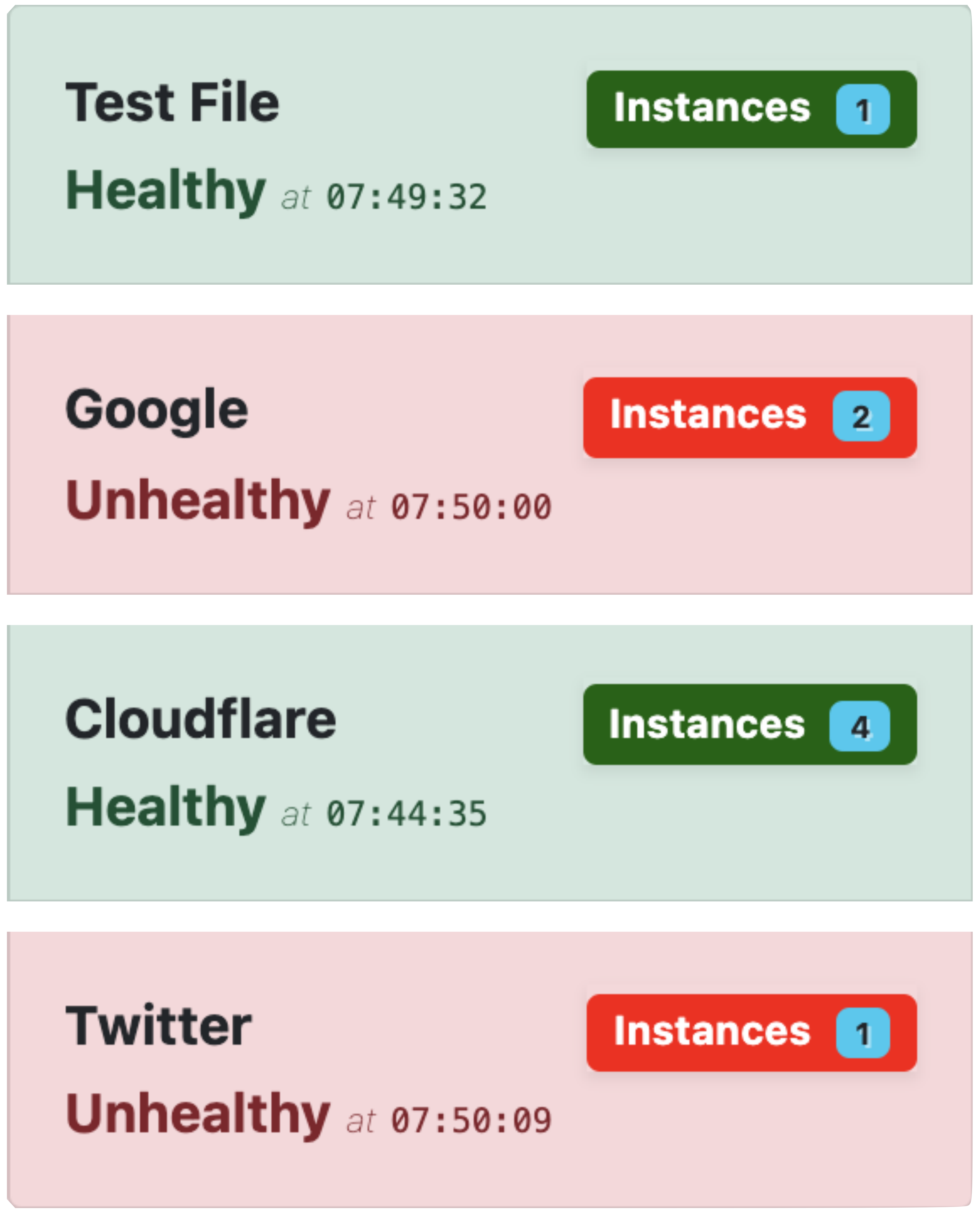
## Recap

- You don’t need a lot of JavaScript
    - for dynamic updates
    - to process data transferred from the server
- You don’t need an SPA
    - for responsive browser experiences
    - unless you have a situation that is both tolerable and unavoidable.

## Recap

- You can use HTML as an effective data transfer format
  - with a semantically rich content model
  - providing low-friction updates to web clients
- You can have a web-native reactive system that is
  - simple
  - fast
  - accessible
  - with a frontend using HTML, CSS, and a #LowJS toolkit.

## https://status.gallery


Test File
Instances 1
Healthy at 07:49:32

Google
Instances 2
Unhealthy at 07:50:00

Cloudflare
Instances 4
Healthy at 07:44:35

Twitter
Instances 1
Unhealthy at 07:50:09

**Josh Graham**
**@delitescere**