

Zone-based Layer Activation

Context-specific Behavior Adaptations across
Logically-connected Asynchronous Operations

Stefan Ramson, Jens Lincke, Harumi Watanabe, and Robert Hirschfeld

Software Architecture Group, Hasso Plattner Institute

Workshop on **Context-Oriented Programming**
and Advanced Modularity | 21 July 2020

Motivational Example

GitHub Application



- » display the most recently changed **repositories** of a user
- » including timestamp and message of the **latest commit**

Motivational Example

GitHub Application



- » display the most recently changed **repositories** of a user
- » including timestamp and message of the **latest commit**

```
const repos = await GitHub.repos()
```

Motivational Example

GitHub Application

- » display the most recently changed **repositories** of a user
- » including timestamp and message of the **latest commit**

```
const repos = await GitHub.repos()  
for (let repo of repos) {
```

Motivational Example

GitHub Application

- » display the most recently changed **repositories** of a user
- » including timestamp and message of the **latest commit**

```
const repos = await GitHub.repos()
for (let repo of repos) {
  const commits = await repo.commits()
```

Motivational Example

GitHub Application

- » display the most recently changed **repositories** of a user
- » including timestamp and message of the **latest commit**

```
const repos = await GitHub.repos()
for (let repo of repos) {
  const commits = await repo.commits()
  display(repo, commits.last)
}
```

Motivational Example GitHub Application

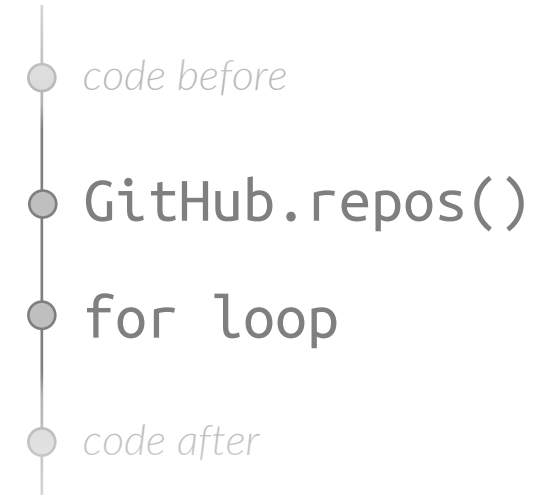
- » display the most recently changed **repositories** of a user
- » including timestamp and message of the **latest commit**

```

async function displayRepos() {
  const repos = await GitHub.repos()
  for (let repo of repos) {
    const commits = await repo.commits()
    display(repo, commits.last)
  }
}

```

displayRepos()



only public repositories



Motivational Example

Behavior Adaptation: Accessing Private Repositories

- » display the most recently changed **repositories** of a user
- » including timestamp and message of the **latest commit**
- » provide an option to include **private** repositories

```

async function displayRepos() {
  const repos = await GitHub.repos()
  for (let repo of repos) {
    const commits = await repo.commits()
    display(repo, commits.last)
  }
}

```



Motivational Example

Intended Behavior Adaptation (with Authentication)

- » display the most recently changed **repositories** of a user
- » including timestamp and message of the **latest commit**
- » provide an option to include **private** repositories



```

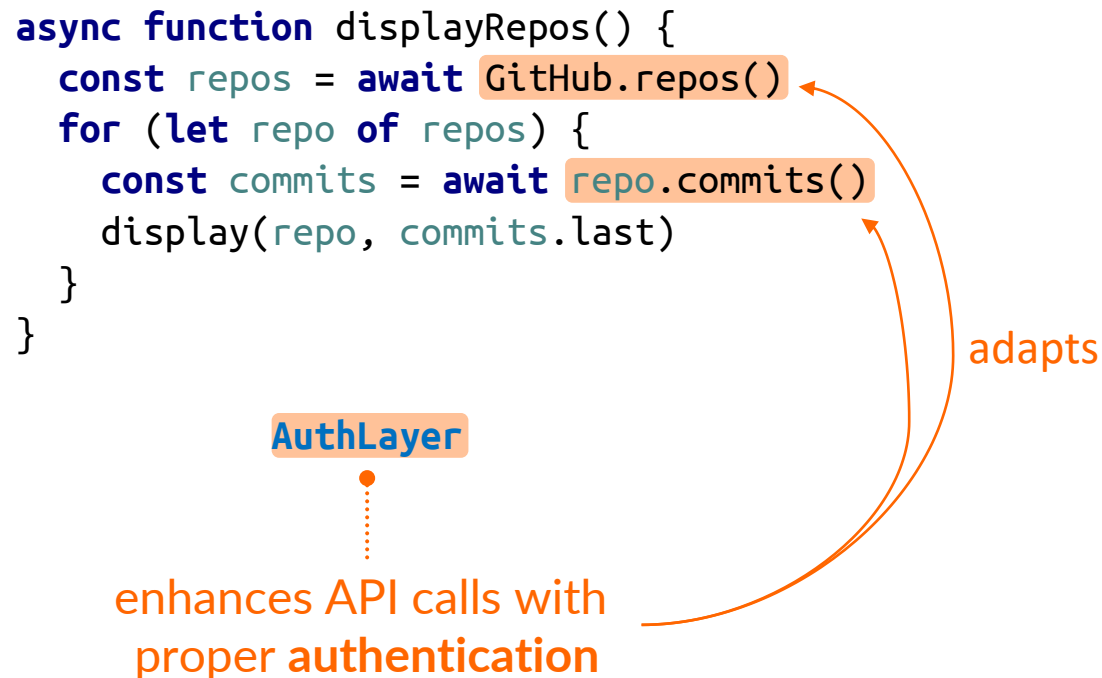
async function displayRepos() {
  const repos = await GitHub.repos()
  for (let repo of repos) {
    const commits = await repo.commits()
    display(repo, commits.last)
  }
}

```

AuthLayer

enhances API calls with proper authentication

adapts



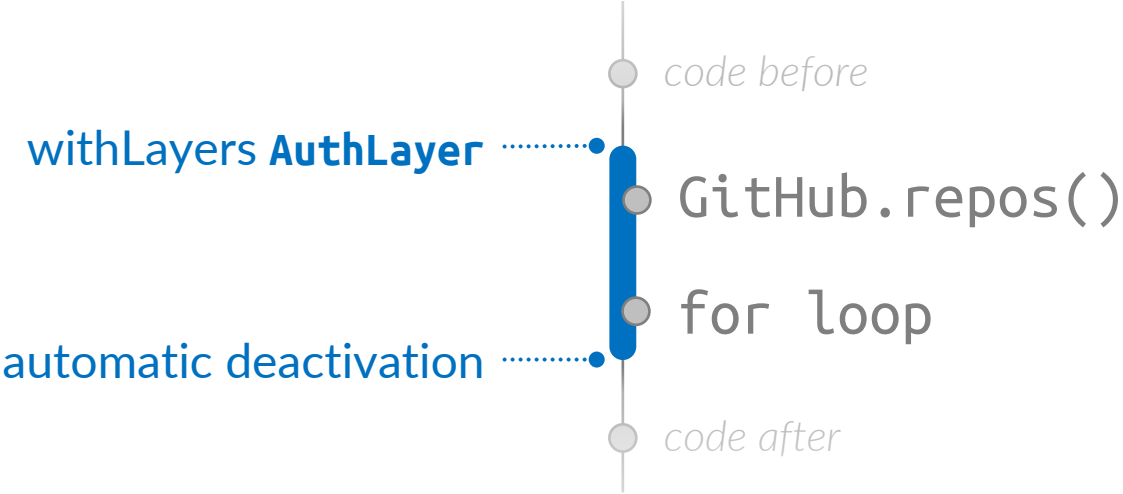
Motivational Example

Intended Behavior Adaptation (Dynamic Extent)

```
async function displayRepos() {  
  const repos = await GitHub.repos()  
  for (let repo of repos) {  
    const commits = await repo.commits()  
    display(repo, commits.last)  
  }  
}
```

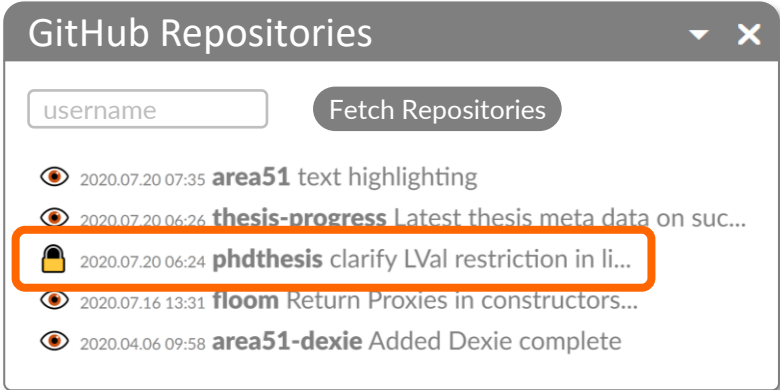
```
withLayers([AuthLayer], displayRepos)
```

enhances API calls with proper authentication



adapts

behavior adaptation

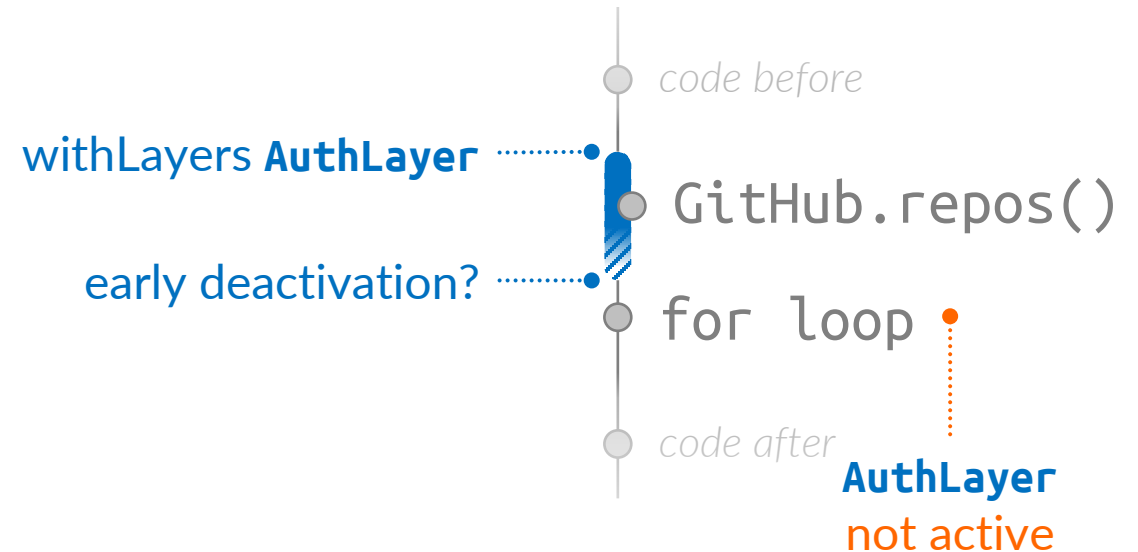


Motivational Example

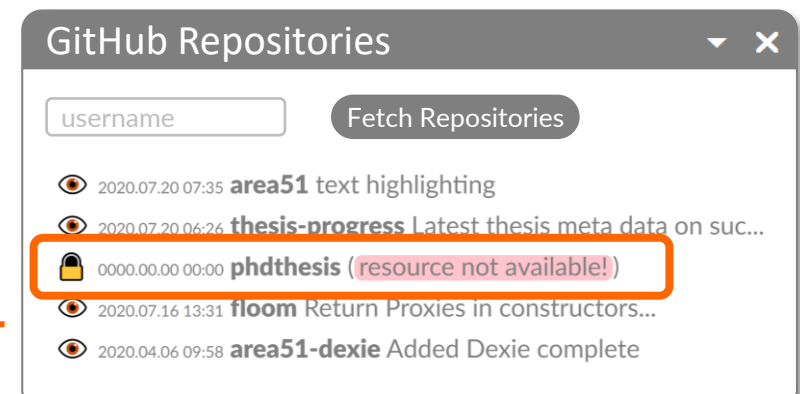
Actual Behavior (Dynamic Extent)

```
async function displayRepos() {  
  const repos = await GitHub.repos()  
  for (let repo of repos) {  
    const commits = await repo.commits()  
    display(repo, commits.last)  
  }  
}
```

```
withLayers([AuthLayer], displayRepos)
```



authentication error



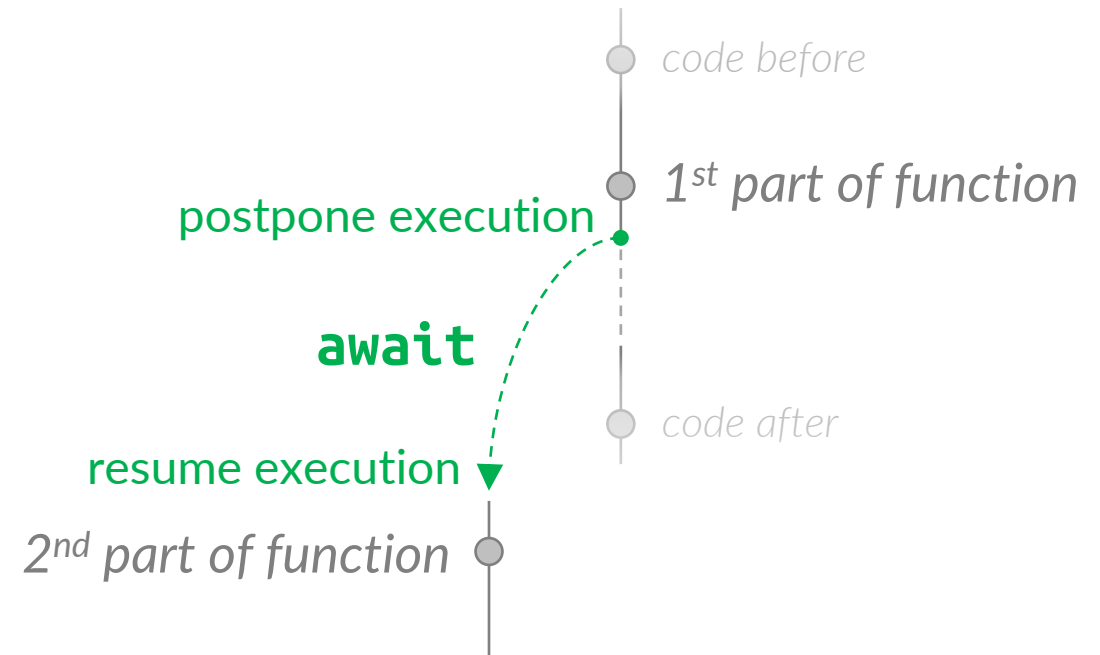
Motivational Example

Asynchronous Execution



```
async function displayRepos() {  
  const repos = await GitHub.repos()  
  for (let repo of repos) {  
    const commits = await repo.commits()  
    display(repo, commits.last)  
  }  
}
```

```
withLayers([AuthLayer], displayRepos)
```



Motivational Example

Asynchronous Execution

```
async function displayRepos() {  
  ● GitHub.repos()  
}
```

breaks the function into multiple pieces

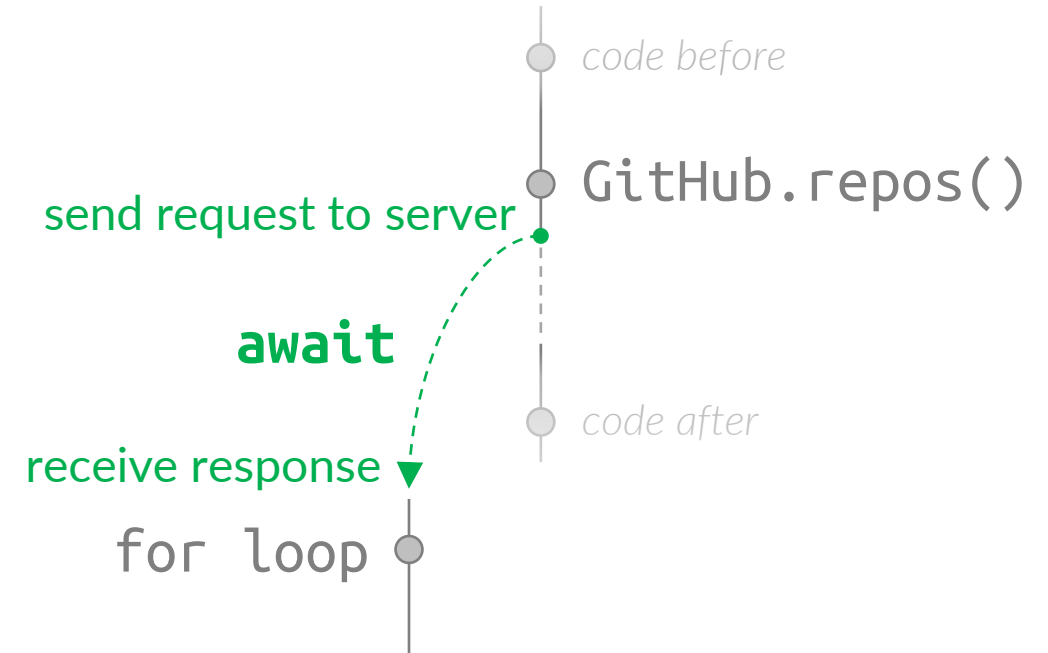
await

executed immediately

postponed until receiving a response

```
const repos =  
for (let repo of repos) {  
  const commits = await repo.commits()  
  display(repo, commits.last)  
}  
}
```

```
withLayers([AuthLayer], displayRepos)
```



Motivational Example

Asynchronous Execution

```
async function displayRepos() {  
  ● GitHub.repos()  
}
```

breaks the function into multiple pieces

await

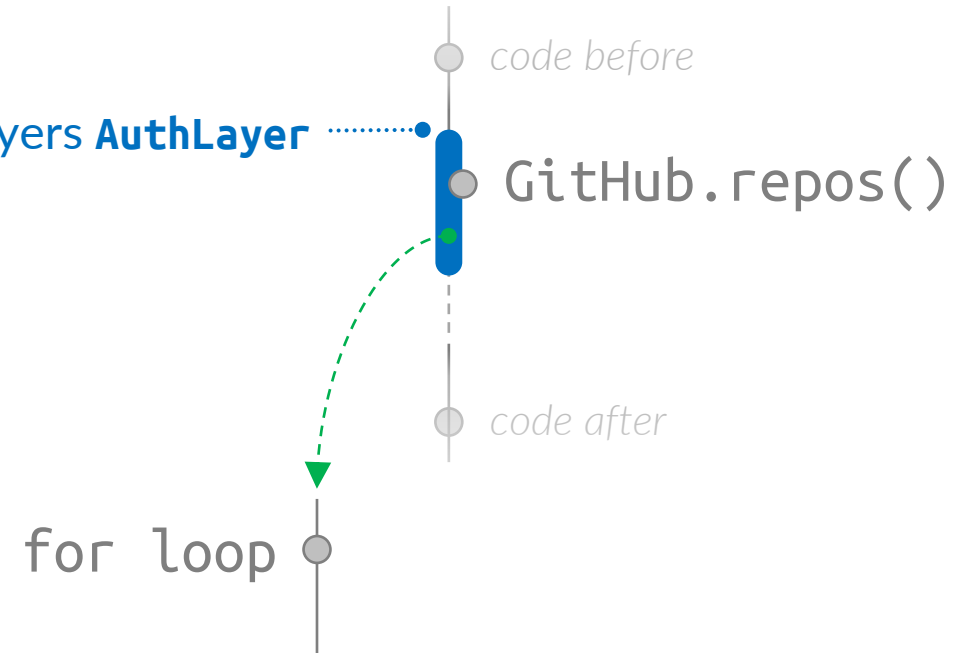
executed immediately

postponed until receiving a response

```
const repos =  
for (let repo of repos) {  
  const commits = await repo.commits()  
  display(repo, commits.last)  
}  
}
```

```
withLayers([AuthLayer], displayRepos)
```

withLayers AuthLayer



Motivational Example

Asynchronous Execution

```
async function displayRepos() {  
  ● GitHub.repos()  
}
```

breaks the function into multiple pieces

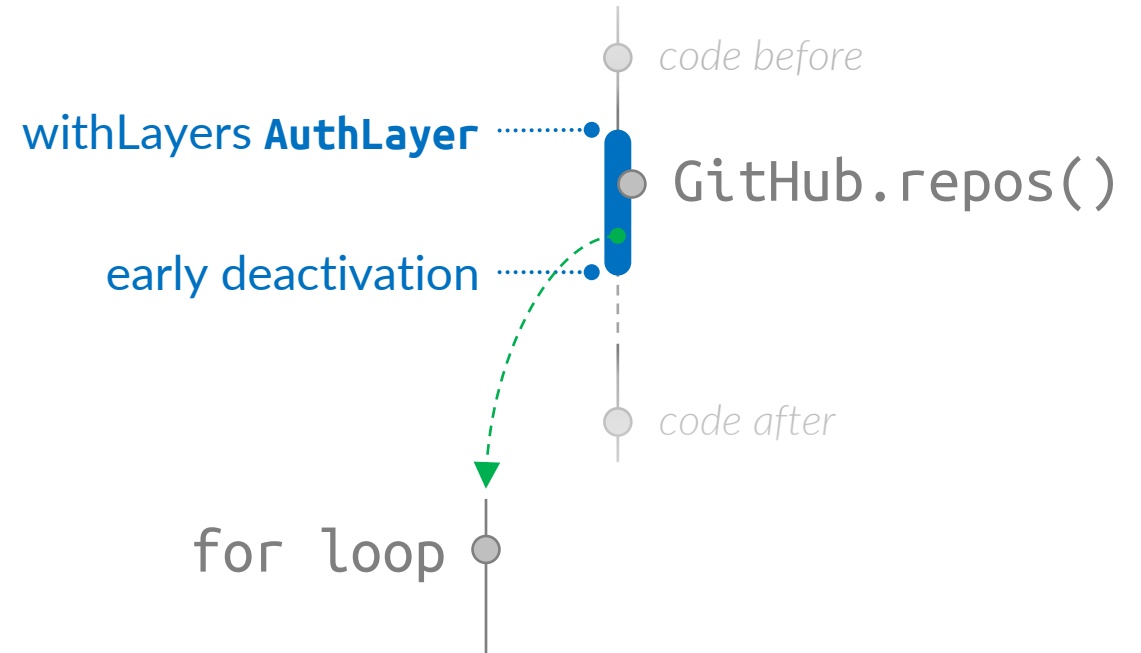
await

executed immediately

postponed until receiving a response

```
const repos =  
for (let repo of repos) {  
  const commits = await repo.commits()  
  display(repo, commits.last)  
}  
}
```

```
withLayers([AuthLayer], displayRepos)
```



Motivational Example

Asynchronous Execution

```
async function displayRepos() {  
  ● GitHub.repos()  
}
```

breaks the function into multiple pieces

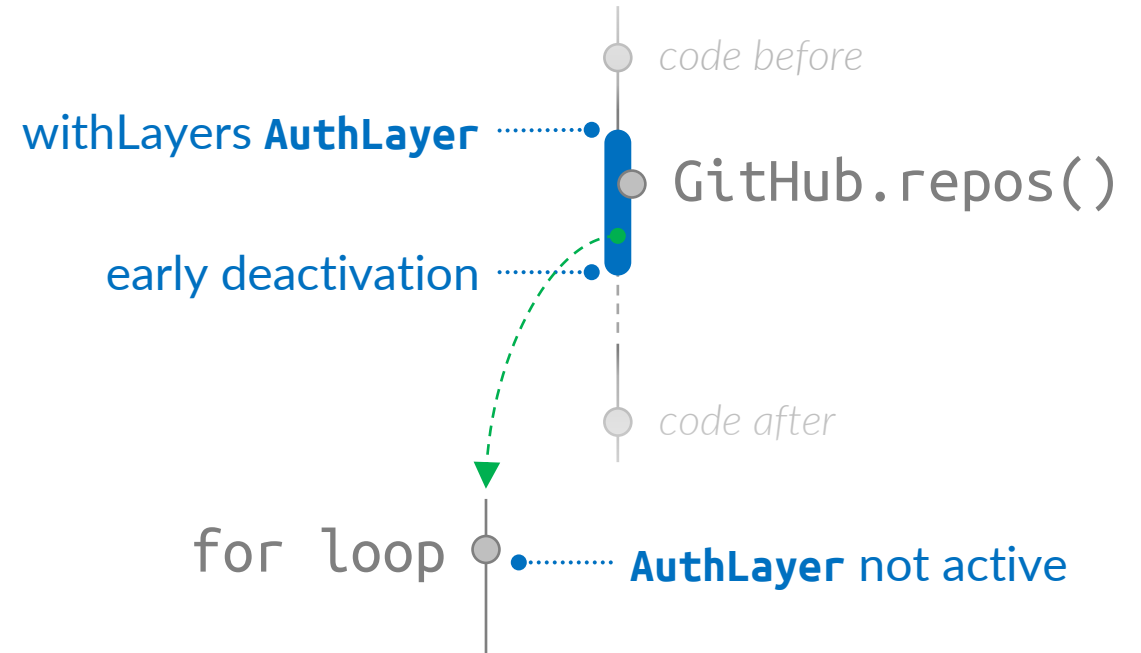
await

executed immediately

postponed until receiving a response

```
const repos =  
for (let repo of repos) {  
  const commits = await repo.commits()  
  display(repo, commits.last)  
}
```

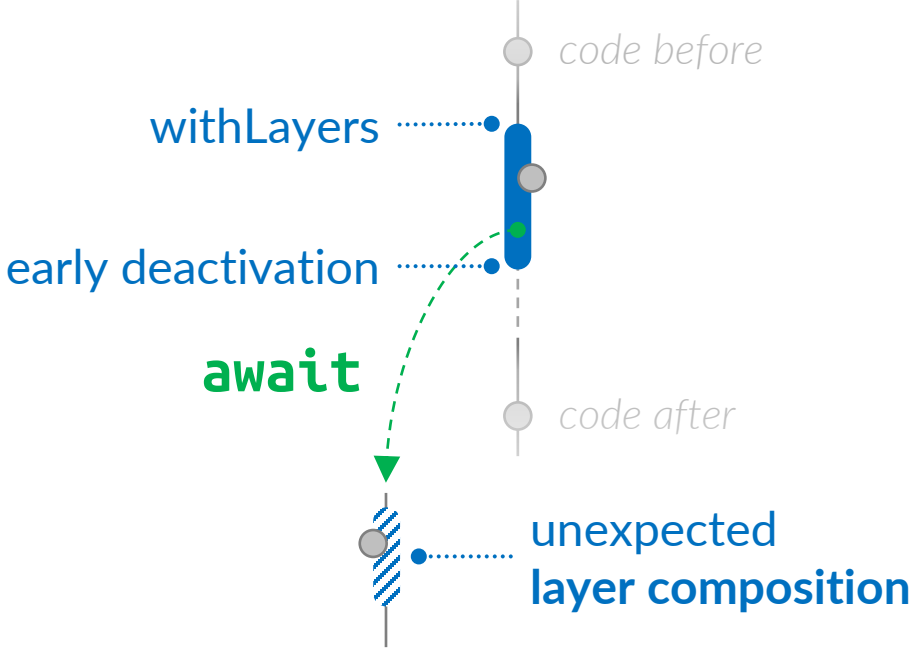
```
withLayers([AuthLayer], displayRepos)
```



Problem Statement

Problem:

- » the **safe semantics** of the dynamic extent activation means **cannot be guaranteed** for asynchronous execution models



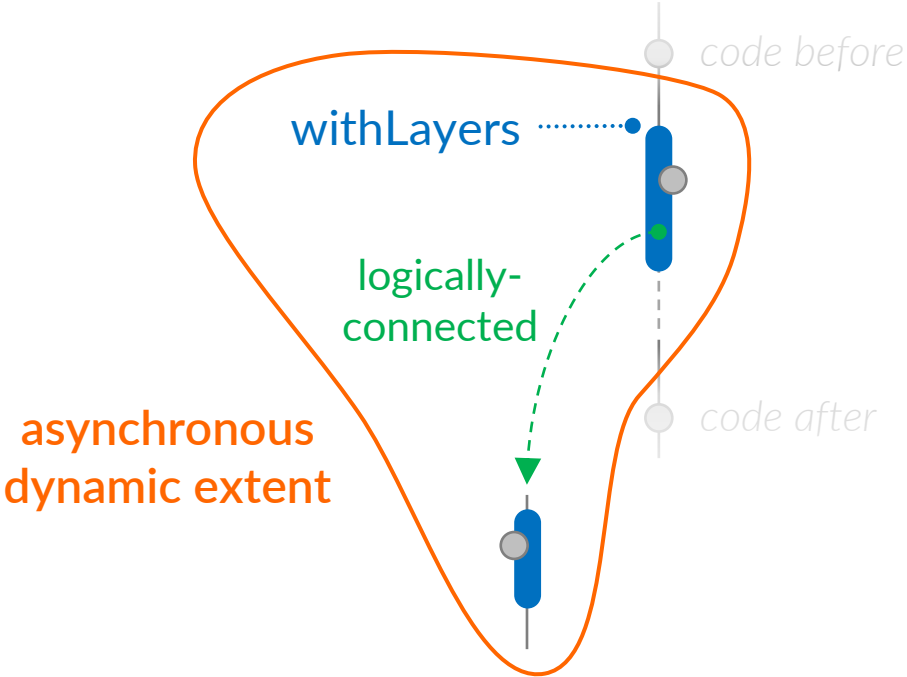
Problem Statement

Problem:

- » the **safe semantics** of the dynamic extent activation means **cannot be guaranteed** for asynchronous execution models

Goals:

- » expand the notion of dynamic extent to **all asynchronous operations** logically-connected to the given block
- » provide a **consistent layer composition** across all these operations

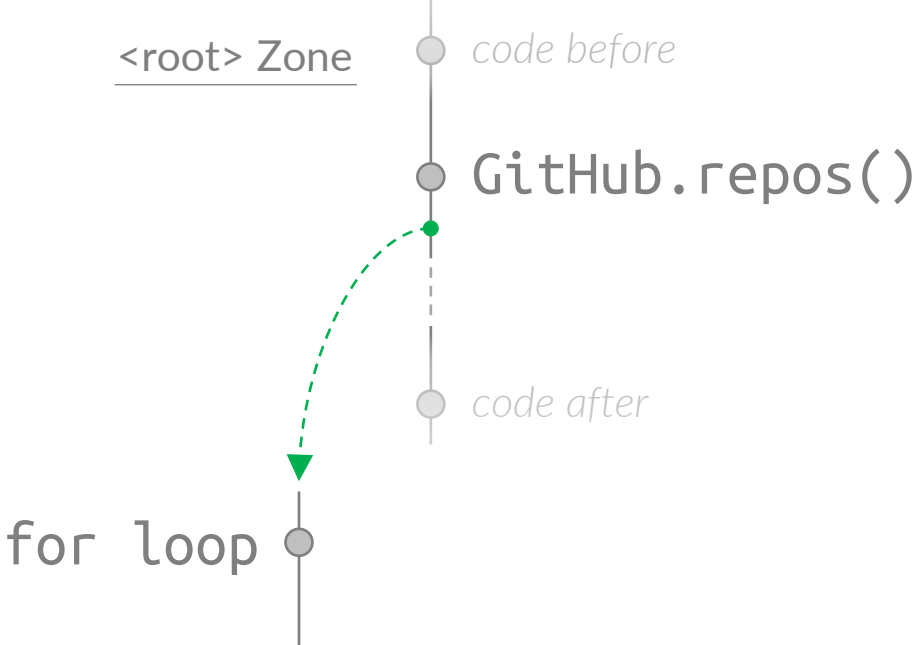


Asynchronous Dynamic Extent

Background: Zones

» a **persistent context** across logically-connected asynchronous operations

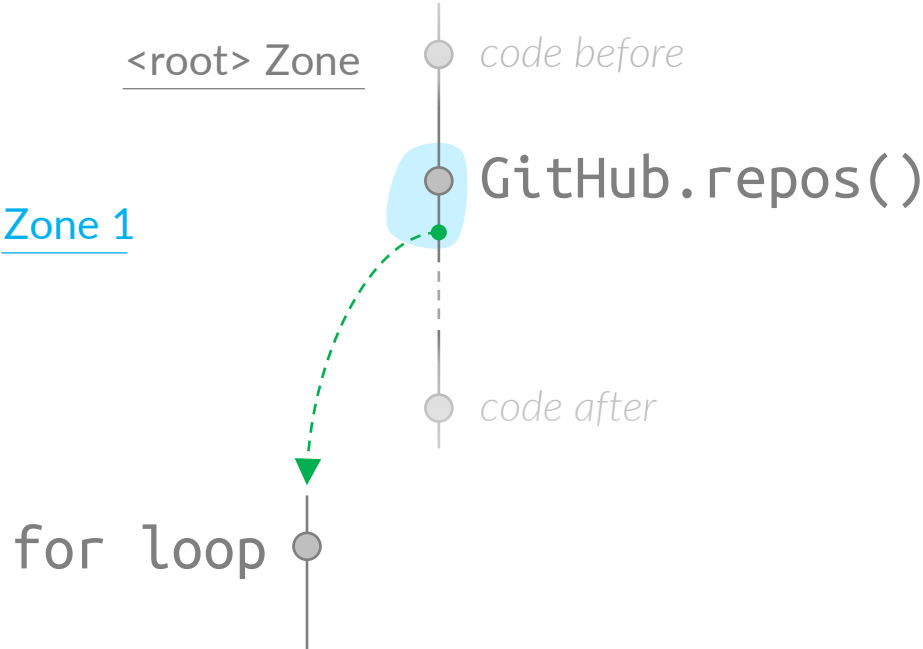
displayRepos()



Background: Zones

» a **persistent context** across logically-connected asynchronous operations

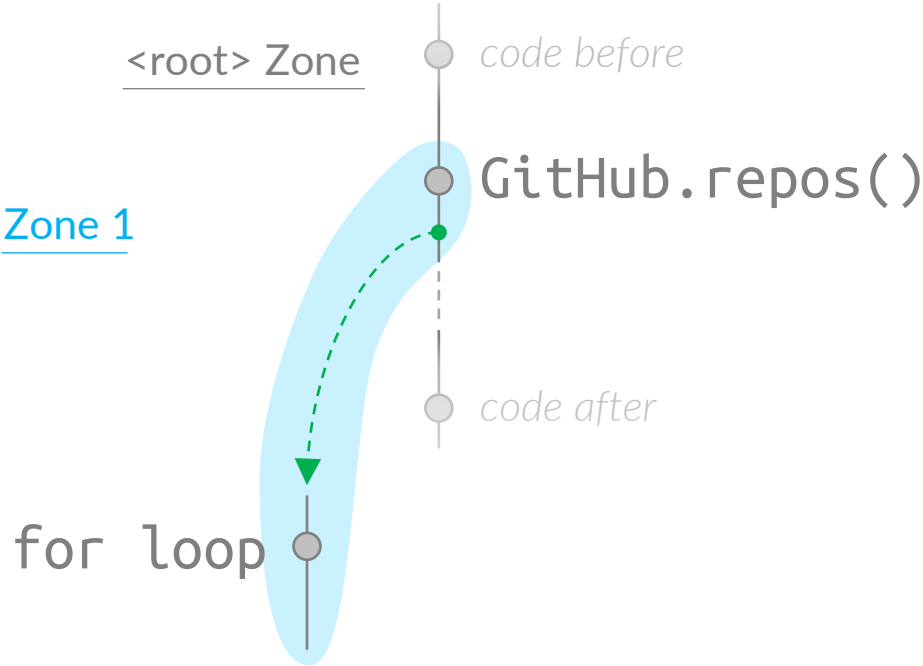
```
runZoned(displayRepos)
```



Background: Zones

- » a **persistent context** across logically-connected asynchronous operations
- » code **scheduled** within a zone is **executed** in that zone

```
runZoned(displayRepos)
```



Background: Zones

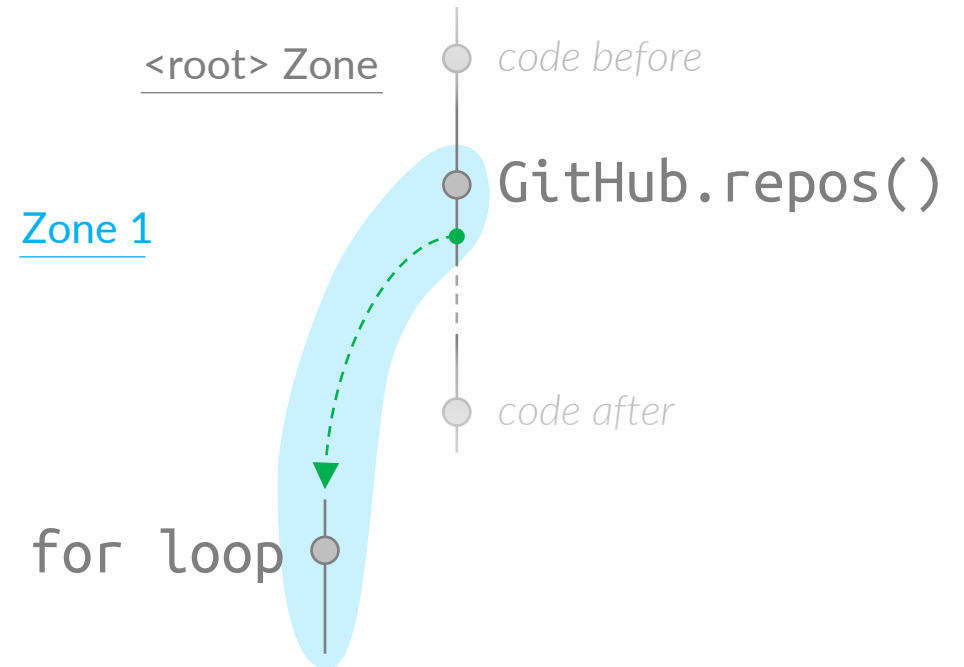
Zone Properties



similar to thread-local storage and dynamic scope

```
runZoned(displayRepos)
```

```
class Github {  
  static async repos () {  
    const json = await this.api('/repos')  
    return json.map(r => new Repository(r))  
  }  
  static async api (path) {  
    const url = GH_API + user + path  
    const response = await fetch(url)  
    return response.json()  
  }  
}
```



Background: Zones

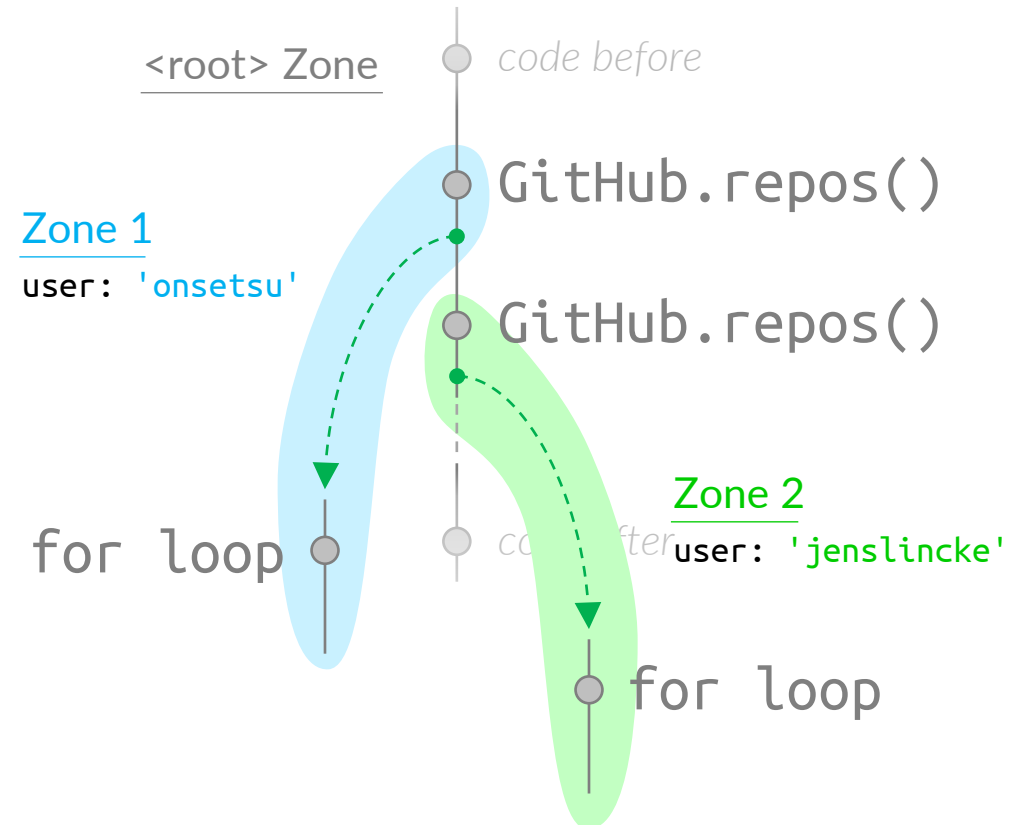
Zone Properties



similar to thread-local storage and dynamic scope

```
runZoned(displayRepos, { user: 'onsetsu' })
runZoned(displayRepos, { user: 'jenslincke' })
```

```
class Github {
  static async repos () {
    const json = await this.api('/repos')
    return json.map(r => new Repository(r))
  }
  static async api (path) {
    const url = GH_API + Zone.current.user + path
    const response = await fetch(url)
    return response.json()
  }
}
```



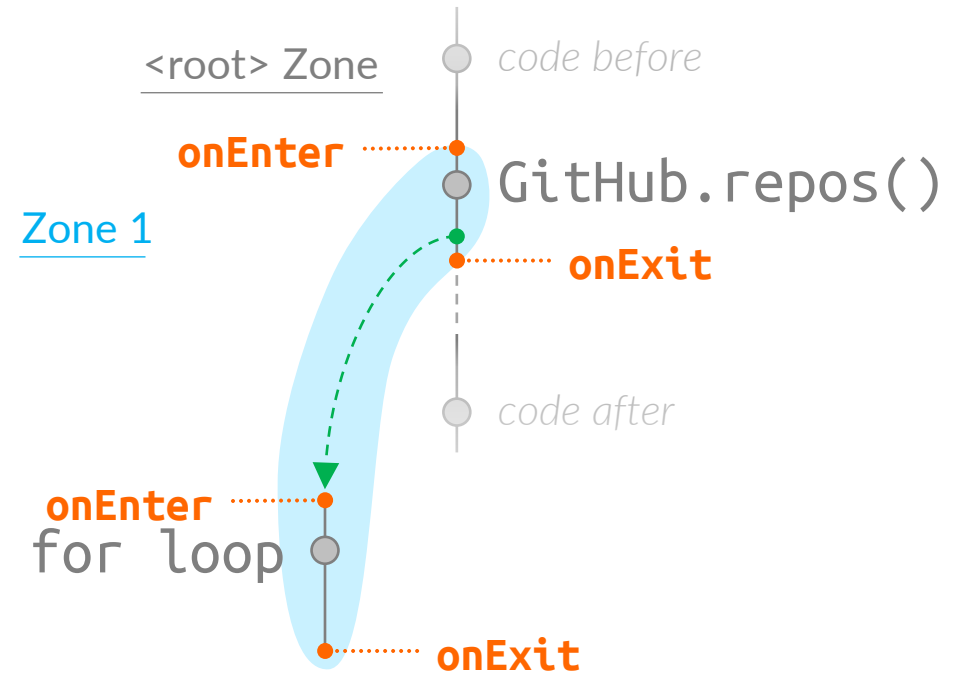
Background: Zones

Life-cycle Callbacks



reflection on the execution of asynchronous tasks

```
runZoned(displayRepos, {  
  onEnter() { [...] },  
  onExit() { [...] },  
  onError() { [...] }  
})
```

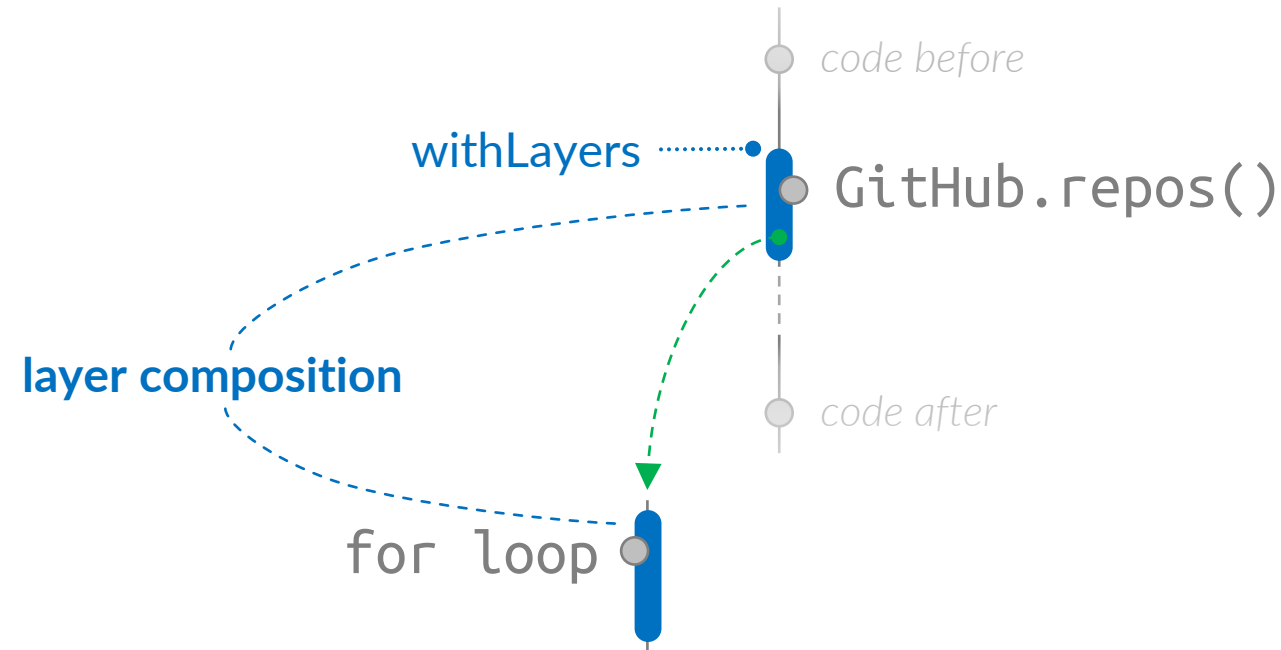


Zone-based Layer Activation

Zone-based Layer Activation

» Goal

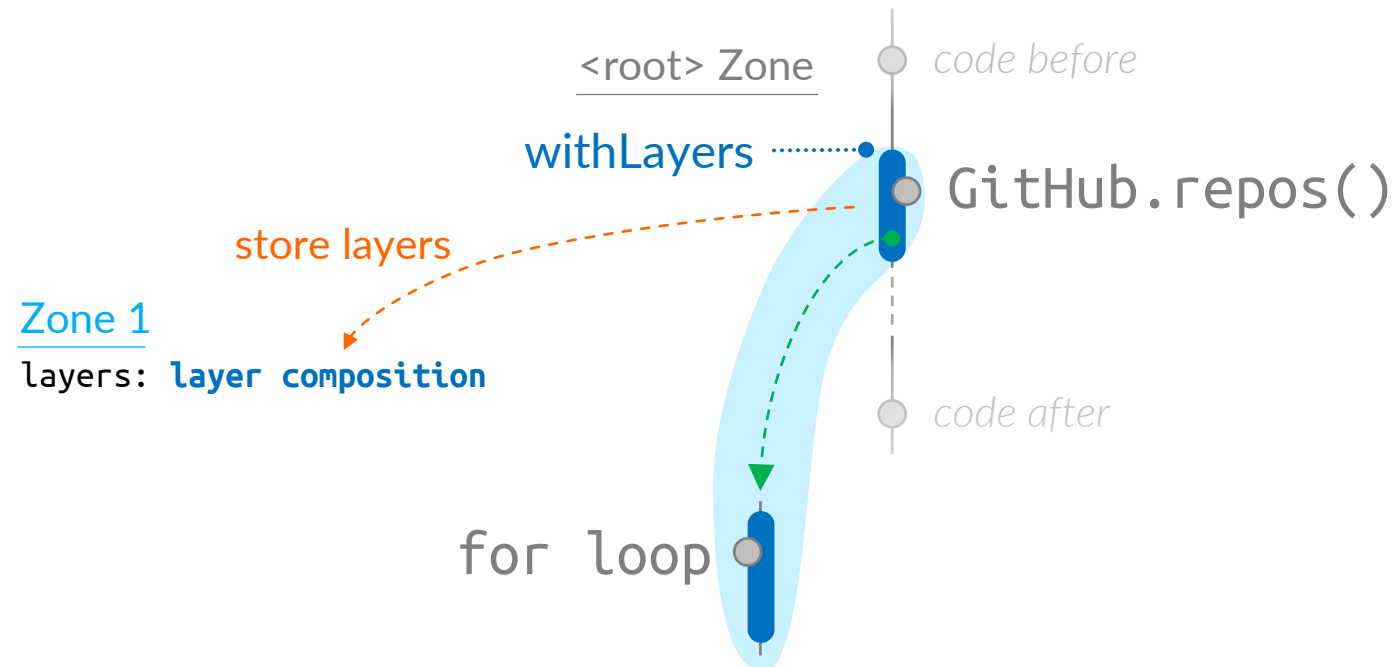
- > **consistent layer composition** across **asynchronous operations**



Zone-based Layer Activation

» Intuition

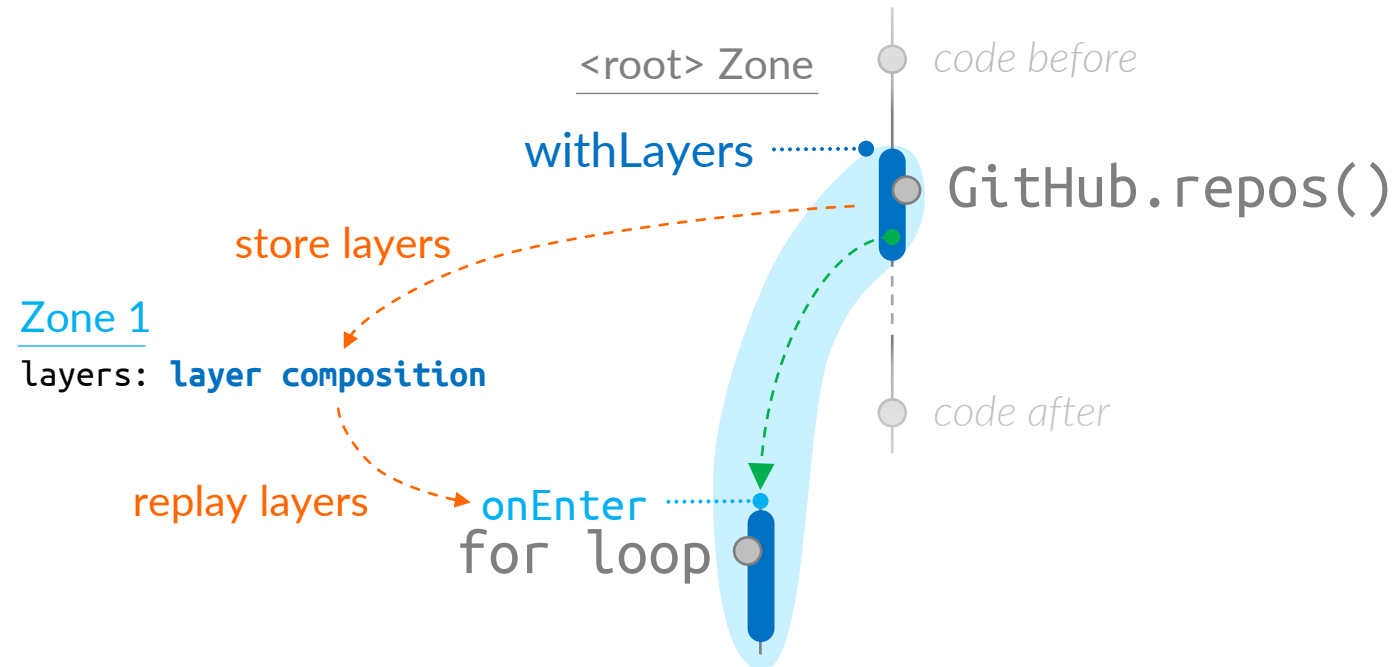
- › **store** a layer composition **per zone**



Zone-based Layer Activation

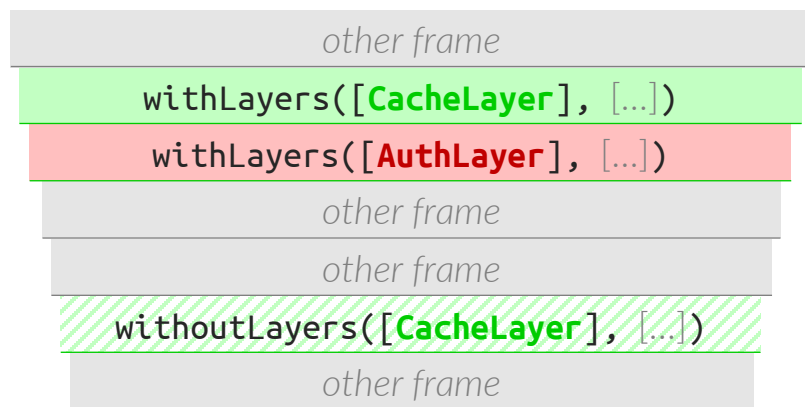
» Intuition

- › **store** a layer composition **per zone**
- › **restore** layer composition when **re-entering a zone**

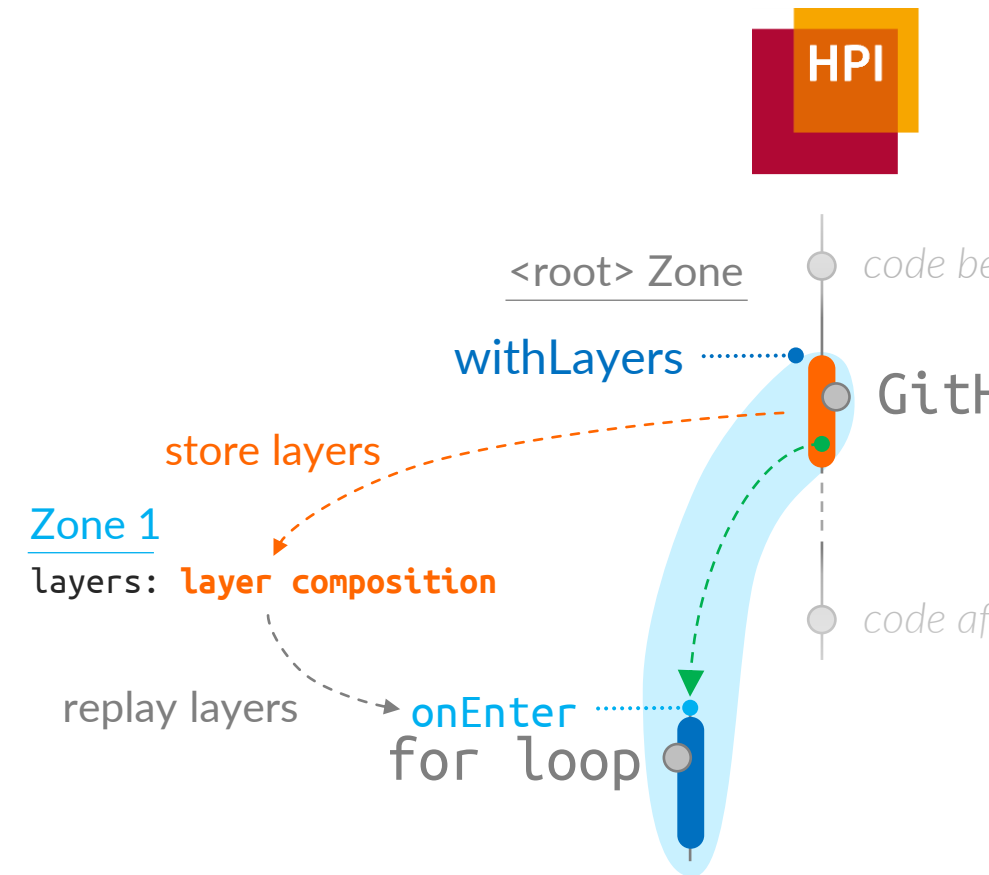


Zone-based Layer Activation Layerstack

- » dynamic extent activates layers based on **local execution context**

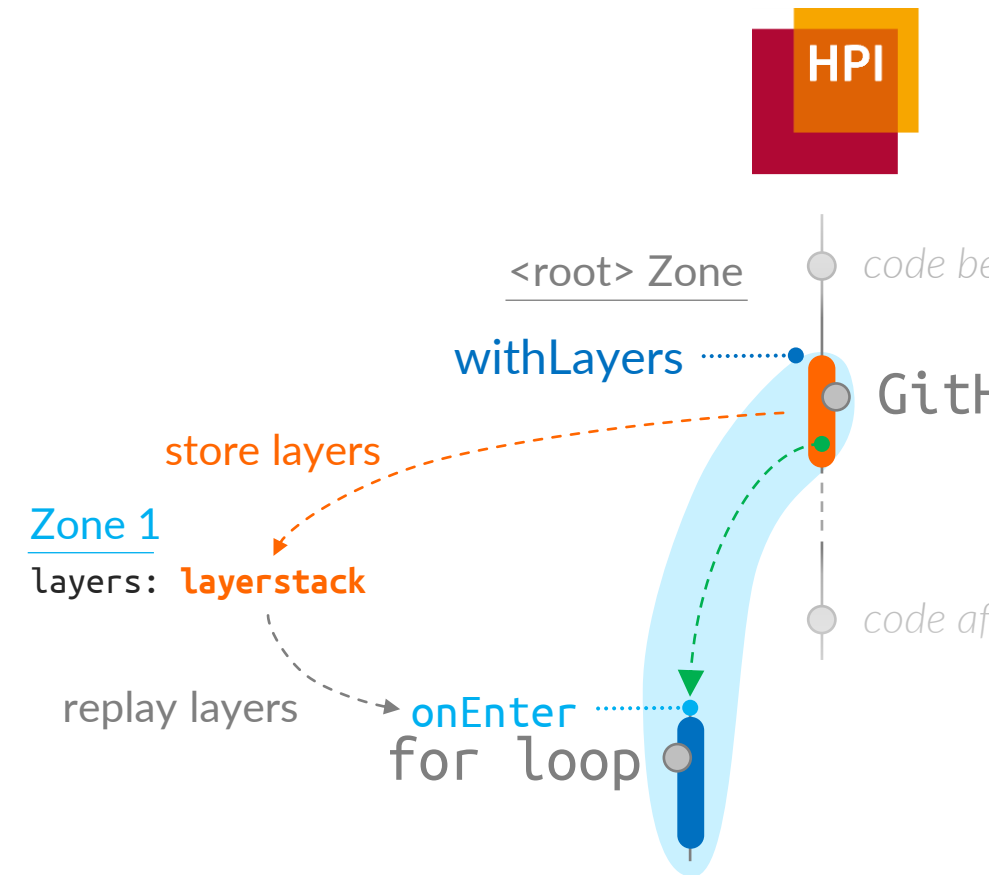
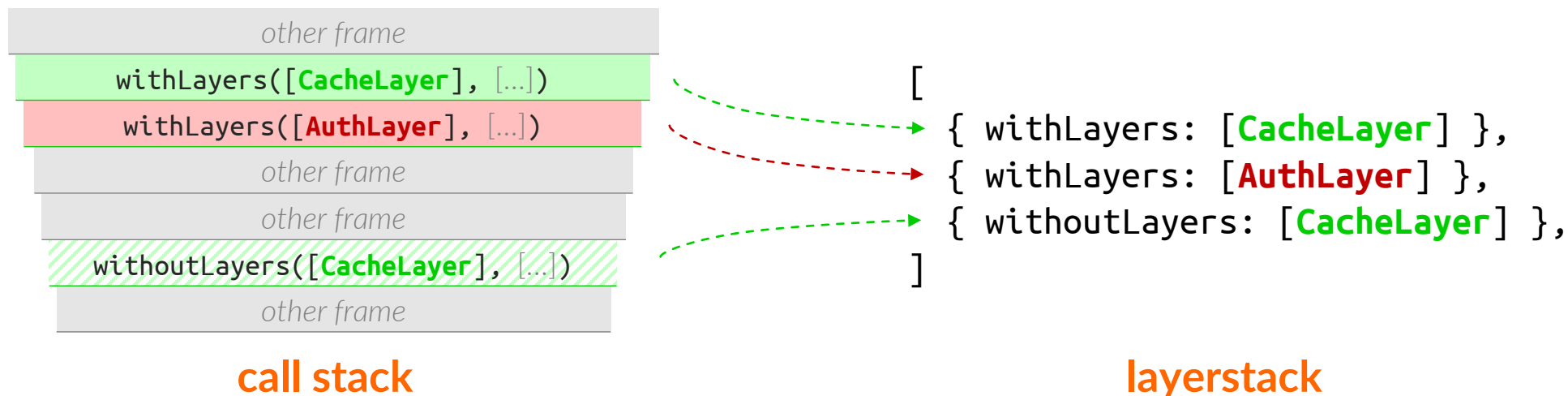


call stack



Zone-based Layer Activation Layerstack

- » dynamic extent activates layers based on **local execution context**
- » only store layerstack as context



Zone-based Layer Activation

Replaying a Layerstack



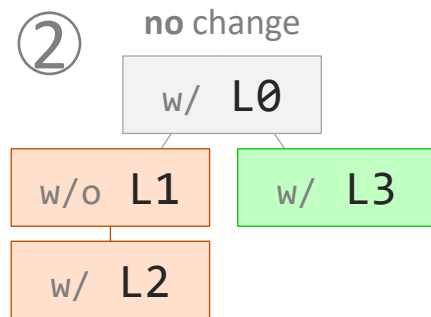
» dedicated scheme to handle layer **life-cycle callbacks**

```
[
  { withLayers: [L0] },
  { withoutLayers: [L1] },
  { withLayers: [L2] },
] .replay(
  [
    { withLayers: [L0] },
    { withLayers: [L3] },
  ]
)
```

①

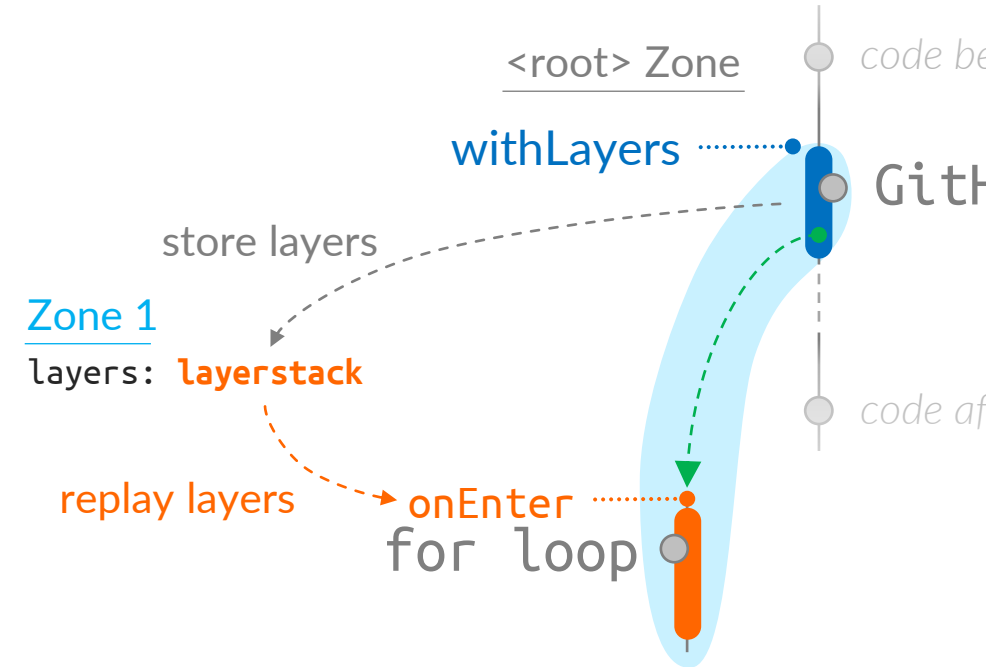
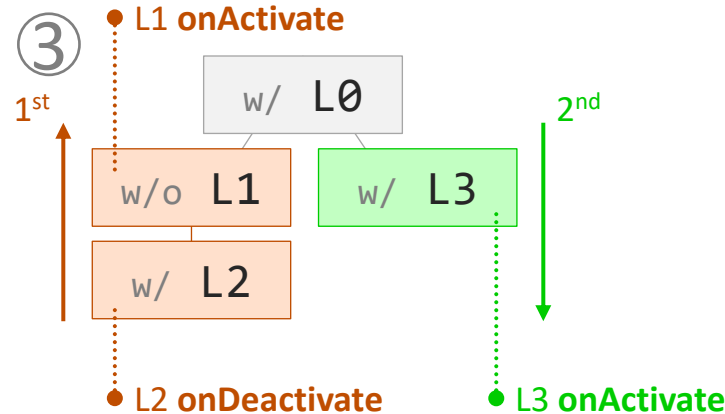


②



remove from LayerStack add to LayerStack

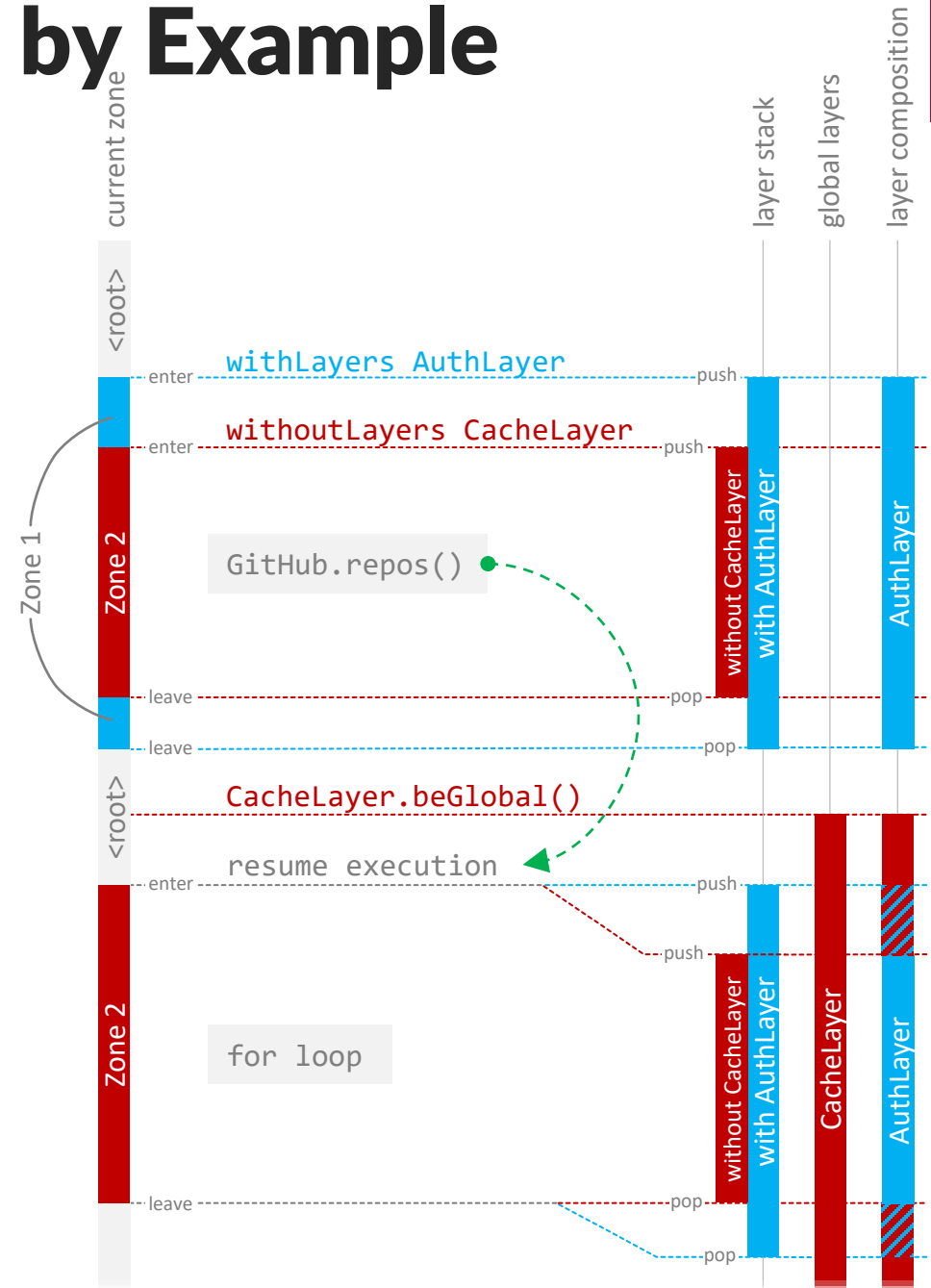
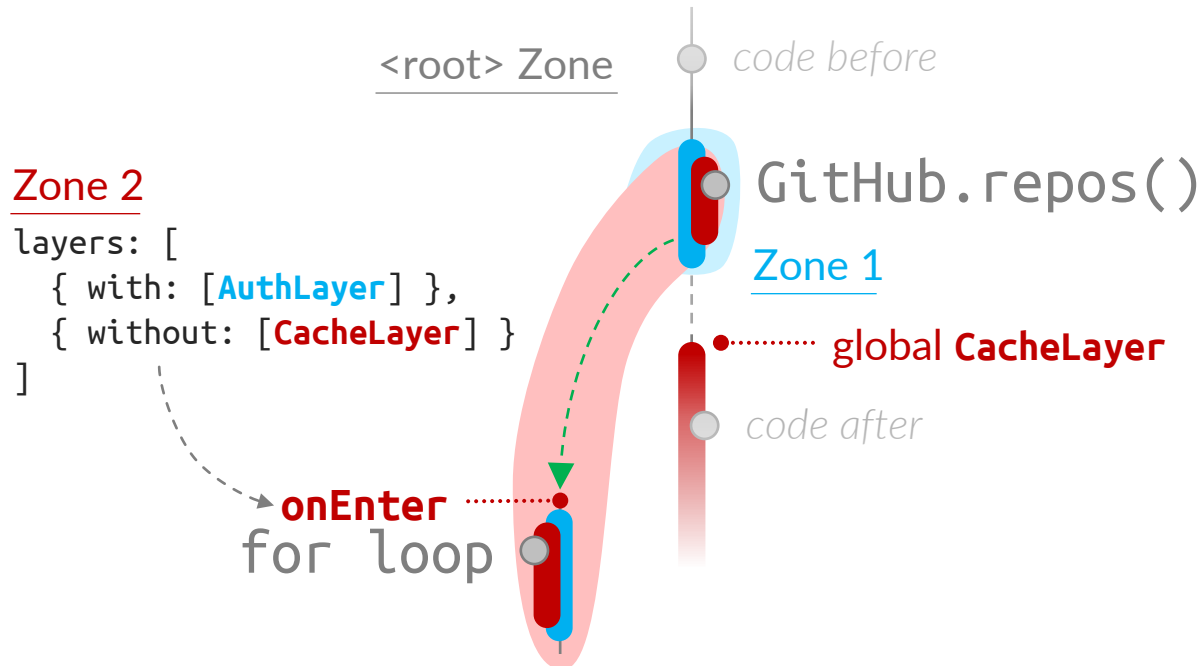
③



Zone-based Layer Activation by Example



```
withLayers([AuthLayer], () =>
  withoutLayers([CacheLayer],
    displayRepos
  )
)
CacheLayer.beGlobal()
```



Implementation

library **area51**

<https://github.com/onsetsu/area51>

- » extension to ContextJS
- » extended Dexie.Promise with zone life-cycle callbacks

Please select a variant for code execution:

- base behavior** (`displayRepos()`)
- dynamic extent** (`withLayers([AuthLayer], displayRepos)`)
- asynchronous dynamic extent** (`withLayersZoned([AuthLayer], displayRepos)`)

onsetsu

GitHub access token

username

Your token is only required when you want to access your private repositories.

Fetch Latest Repositories

Clear List

- 👁️ floom 2020.07.16 13:31 Return Proxies in constructors...
- 👁️ area51 2020.07.01 13:14 example text
- 👁️ thesis-progress 2020.06.22 08:00 Latest thesis meta data on suc...
- 🔒 phdthesis (resource not available!)
- 👁️ area51-dexie 2020.04.06 09:58 Added Dexie complete
- 👁️ floom 2020.07.16 13:31 Return Proxies in constructors...
- 👁️ area51 2020.07.01 13:14 example text
- 👁️ thesis-progress 2020.06.22 08:00 Latest thesis meta data on suc...
- 🔒 phdthesis 2020.06.22 07:58 zones stub
- 👁️ area51-dexie 2020.04.06 09:58 Added Dexie complete

life example: <https://onsetsu.github.io/area51/experiments/github-access.html>

Asynchronous Programming beyond async/await



- » asynchronous mechanisms:
 - › **async/await keyword**
 - › timeouts
 - › animations
 - › event handlers
 - › ...

- » zones apply to all
 - › subject to Zone-based Layer Activation

Summary

Problem

Notion of **dynamic extent** limited to synchronous parts of code fragments

Motivational Example Asynchronous Execution

```
async function displayRepos() {
  return GitHub.repos()
}

const repos =
  for (let repo of repos) {
    const commits = await repo.commits()
    display(repo, commits.last)
  }

withLayers([AuthLayer], displayRepos)
```

Breaks the function into multiple pieces

executed immediately

suspended until receiving a response

code before

withLayers AuthLayer

early deactivation

code after

for Loop

AuthLayer not active

GitHub.repos()

2020-07-21 Stefan Ramson | Hasso Plattner Institute, Software Architecture Group 14

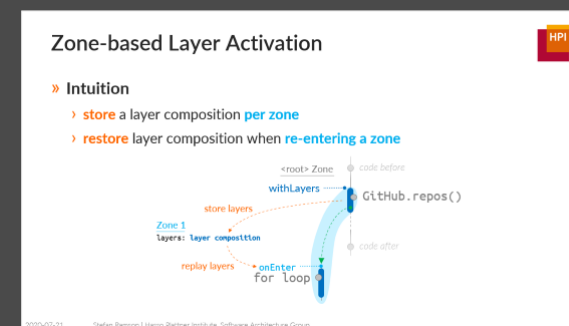
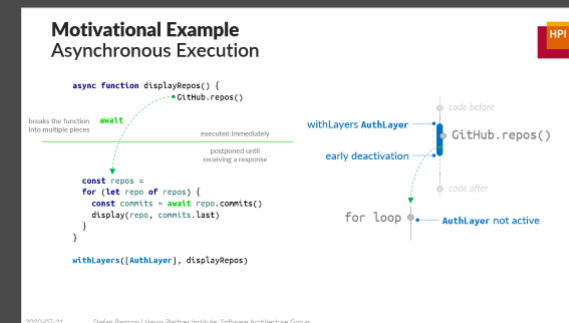
Summary

Problem

Notion of **dynamic extent** limited to synchronous parts of code fragments

Approach

Use **zones** to intercept asynchronous execution and manipulate layer composition accordingly



Summary

Problem

Notion of **dynamic extent** limited to synchronous parts of code fragments

Approach

Use **zones** to intercept asynchronous execution and manipulate layer composition accordingly

Impact

Consistent layer composition across logically-connected asynchronous operations

Motivational Example Asynchronous Execution

```

async function displayRepos() {
  => GitHub.repos()
}

const repos =
for (let repo of repos) {
  const commits = await repo.commits()
  display(repo, commits.last)
}

withLayers([AuthLayer], displayRepos)

```

Breaks the function into multiple pieces

- executed immediately
- suspended until receiving a response

code before

withLayers AuthLayer

early deactivation

code after

for Loop

AuthLayer not active

2020-07-21 Stefan Ramson | Hasso Plattner Institute, Software Architecture Group 15

Zone-based Layer Activation

» Intuition

- » store a layer composition **per zone**
- » restore layer composition when **re-entering a zone**

code before

code after

Zone 1

store layers

layers: layer composition

replay layers

onEnter

for Loop

2020-07-21 Stefan Ramson | Hasso Plattner Institute, Software Architecture Group 29

Implementation

library **area51**
<https://github.com/onsetu/area51>

» extension to **ContextJS**

» extended **Dexie.Promise** with zone life-cycle callbacks