

# Cloud Server Infrastructure Management and Visualization

The required strategies for managing the resources efficiently

# What is covered in the presentation?

- AWS Organization
  - Identity and Access Management (IAM)
  - Single Sign-On (SSO) with Google Workspace
- AWS Resources and Workloads
  - Applications
  - Data Processing
- Infrastructure Team Methodologies and Toolings
  - Graphical and Visualization Exchange
    - Slack
  - Documentation
    - Git Repository
    - Confluence

# **AWS Organization**

The foundation of a well-managed infrastructure

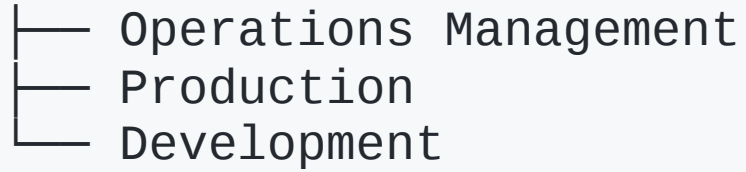
## **Identity and Access Management (IAM)**

Is a web service that helps you securely control access to AWS resources

# Managing Organization Units (OUs)

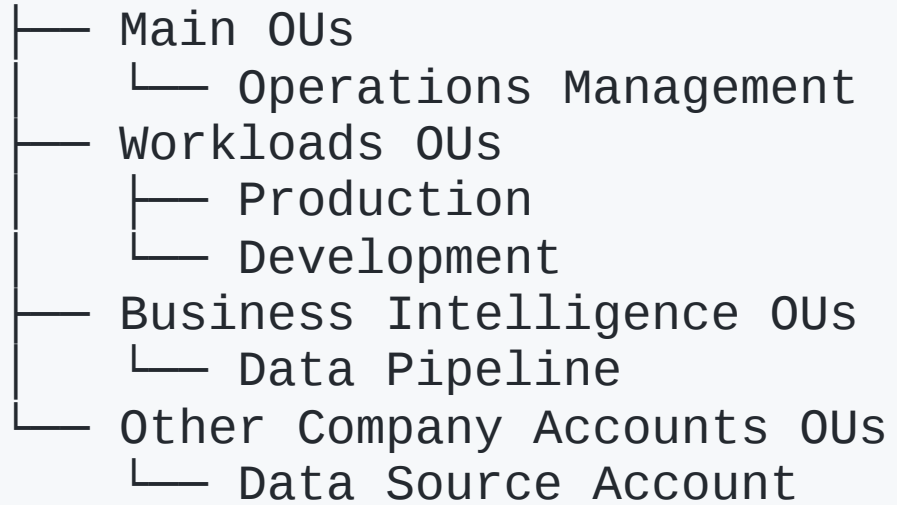
## Scenario 1

Root OUs

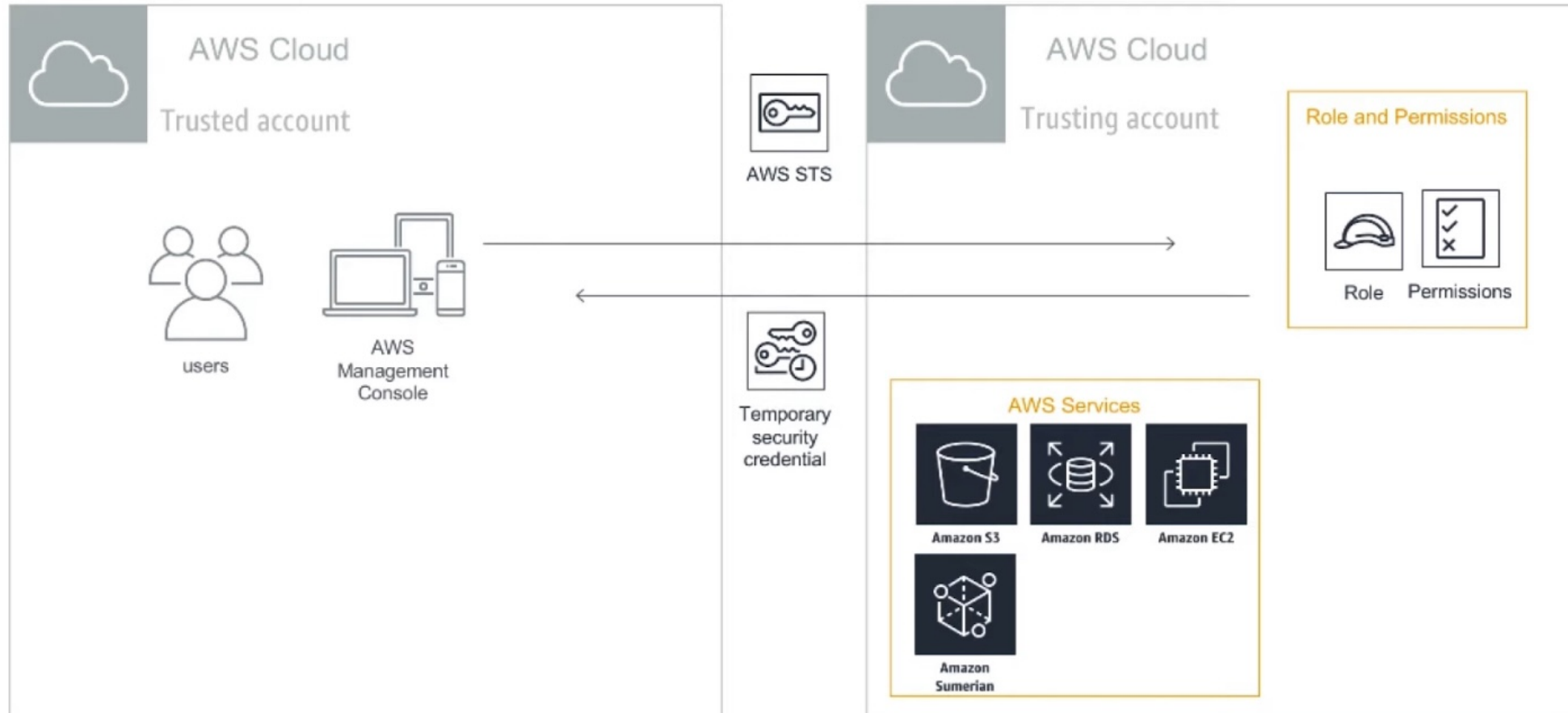


## Scenario 2

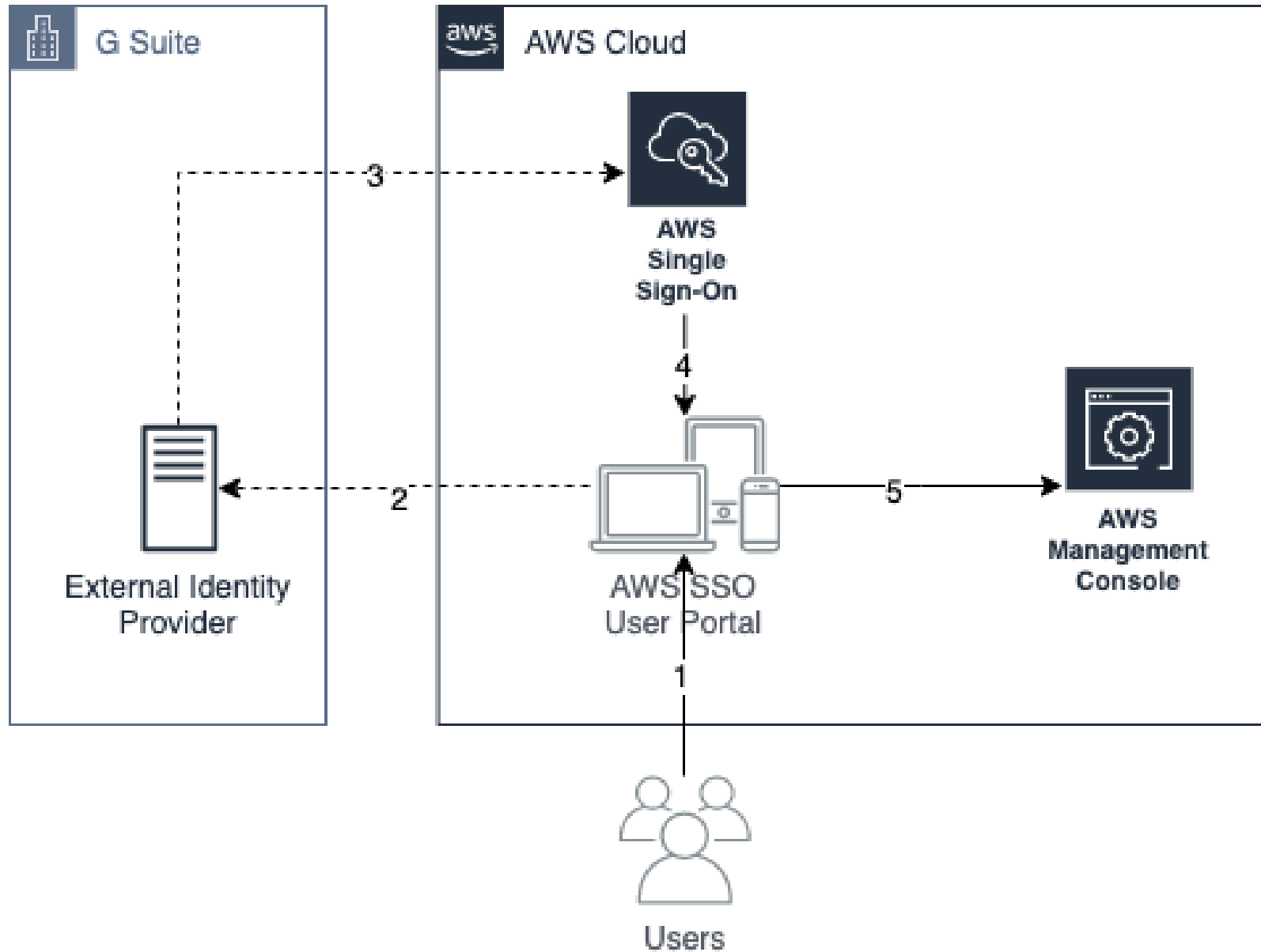
Root OUs



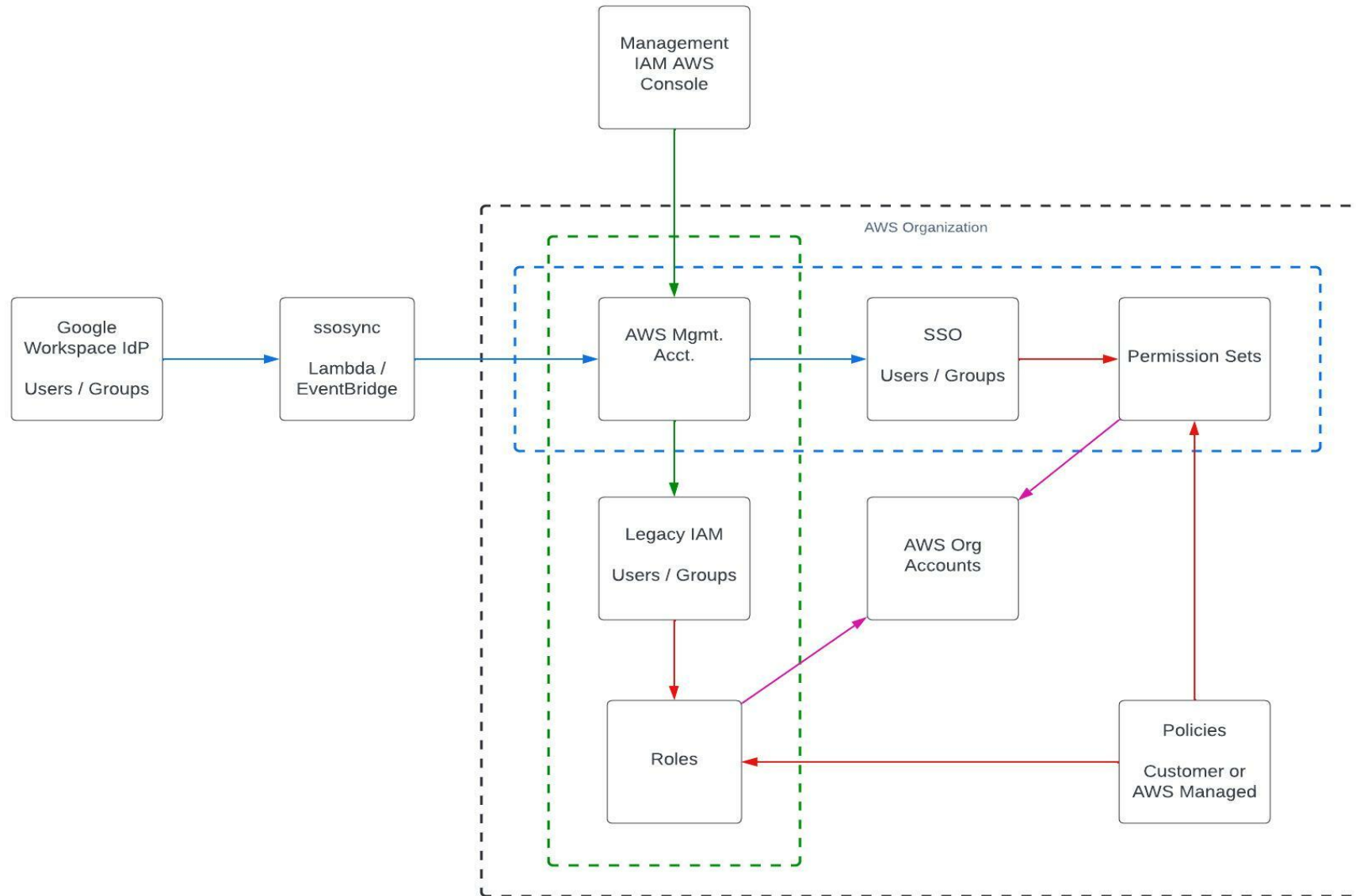
# Assume Role Across Organization Units Account



# Google Workspace as an external identity provider for AWS SSO



## IAM and SSO in a Whole Picture





# Master the IAM Policy Structure

```
{  
  "Statement": [{  
    "Effect": "effect",  
    "Principal": "principal",  
    "Action": "action",  
    "Resource": "arn",  
    "Condition": {  
      "condition": {  
        "key": "value" }  
      }  
    }  
  ]  
}
```

**P**incipal – The entity that is allowed or denied access

*"Principal": "AWS": "arn:aws:iam::123456789012:user/username"*

**A**ction – Type of access that is allowed or denied access

*"Action": "s3:GetObject"*

**R**esource – The Amazon resource(s) the action will act on

*"Resource": "arn:aws:sqs:us-west-2:123456789012:queue1"*

**C**ondition – The conditions under the access defined is valid

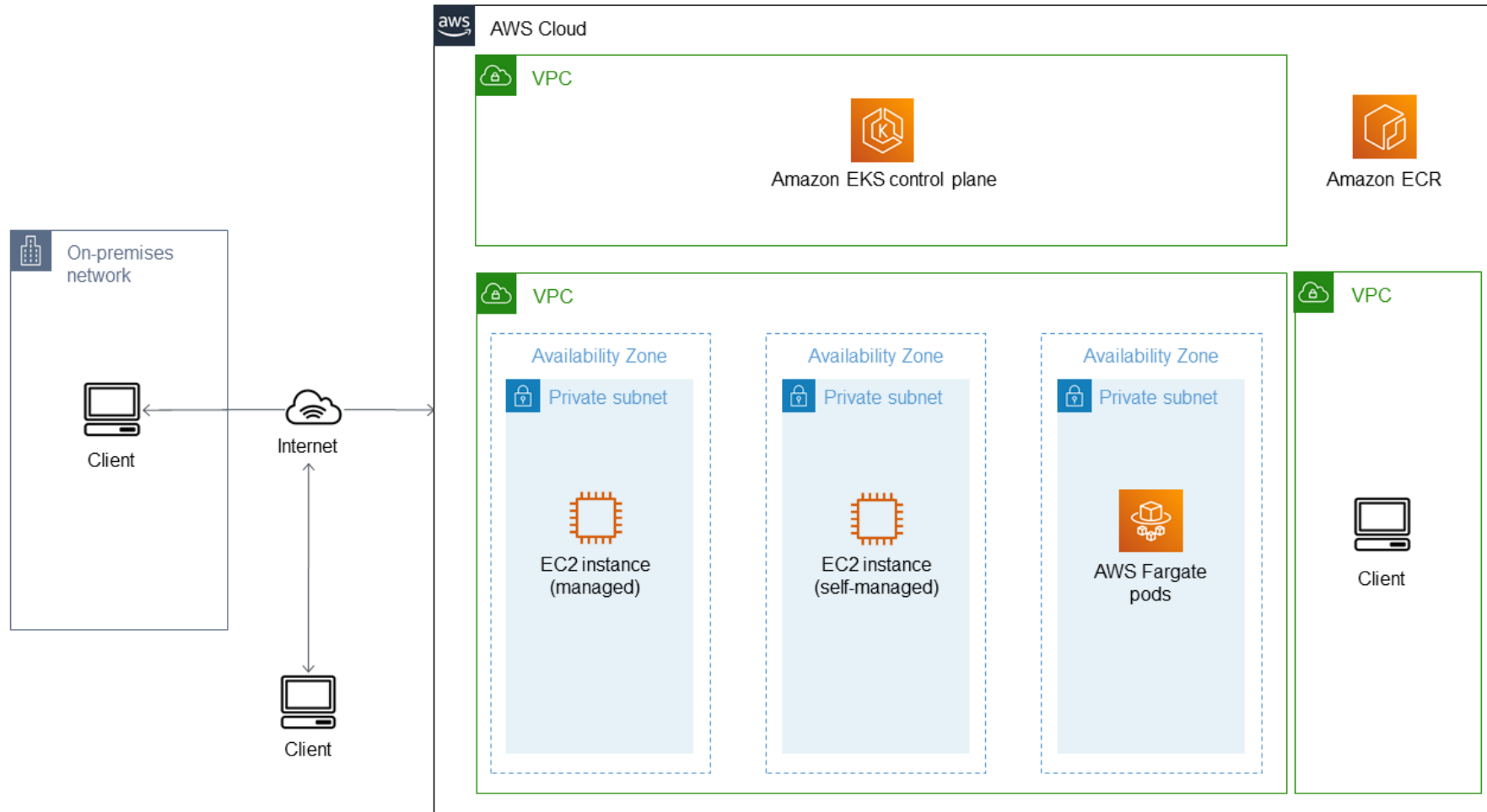
*"StringEqualsIfExists": {"aws:RequestTag/project": ["Pickles"]}*

## **Orchestrated Services**

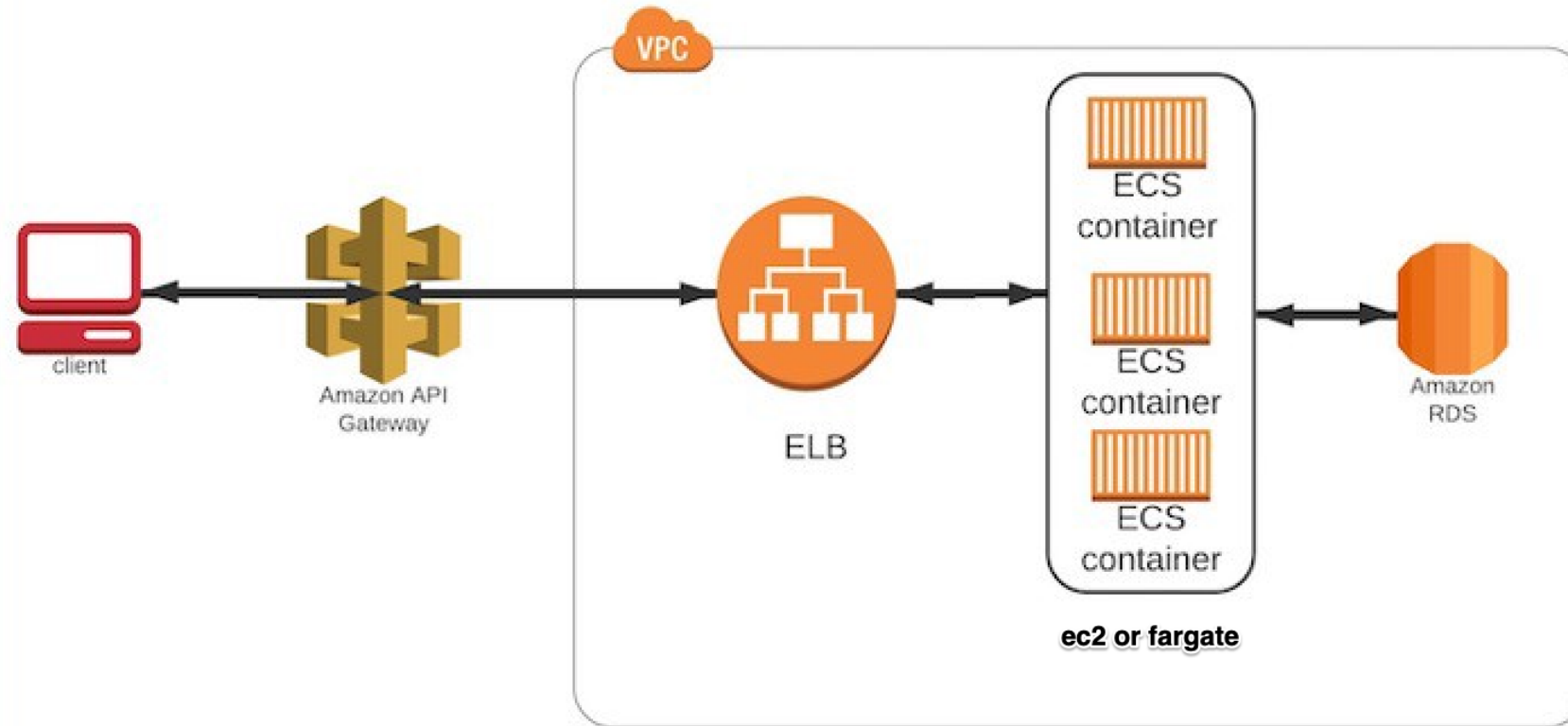
**Amazon Kubernetes Service (Amazon EKS)**

**Amazon Elastic Container Service (Amazon ECS)**

# Amazon Kubernetes Service (Amazon EKS)



# Amazon Elastic Container Service (Amazon ECS)



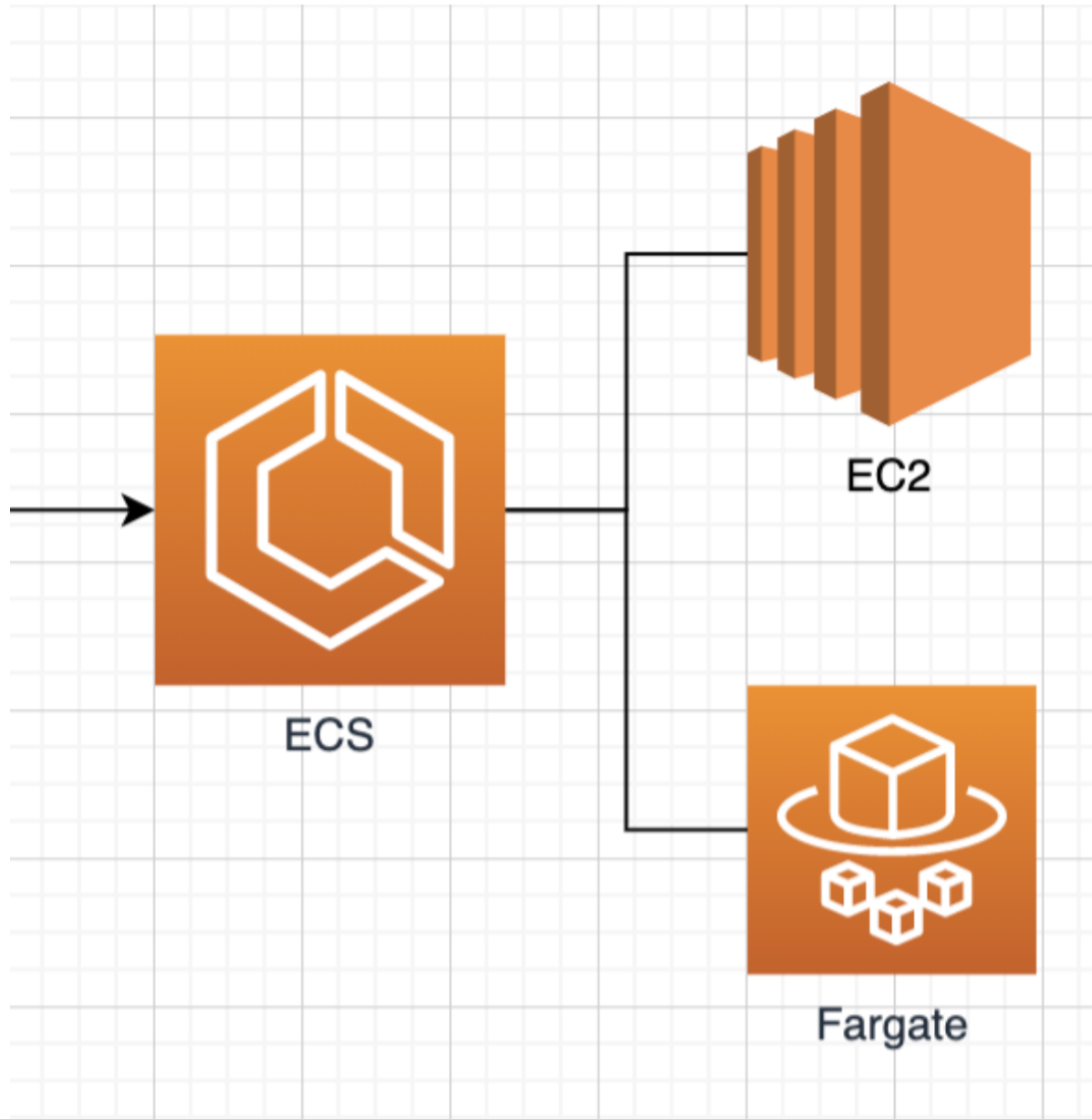
- EC2 Compute, or
- ECS/Fargate

# **Microservice and Serverless Architecture**

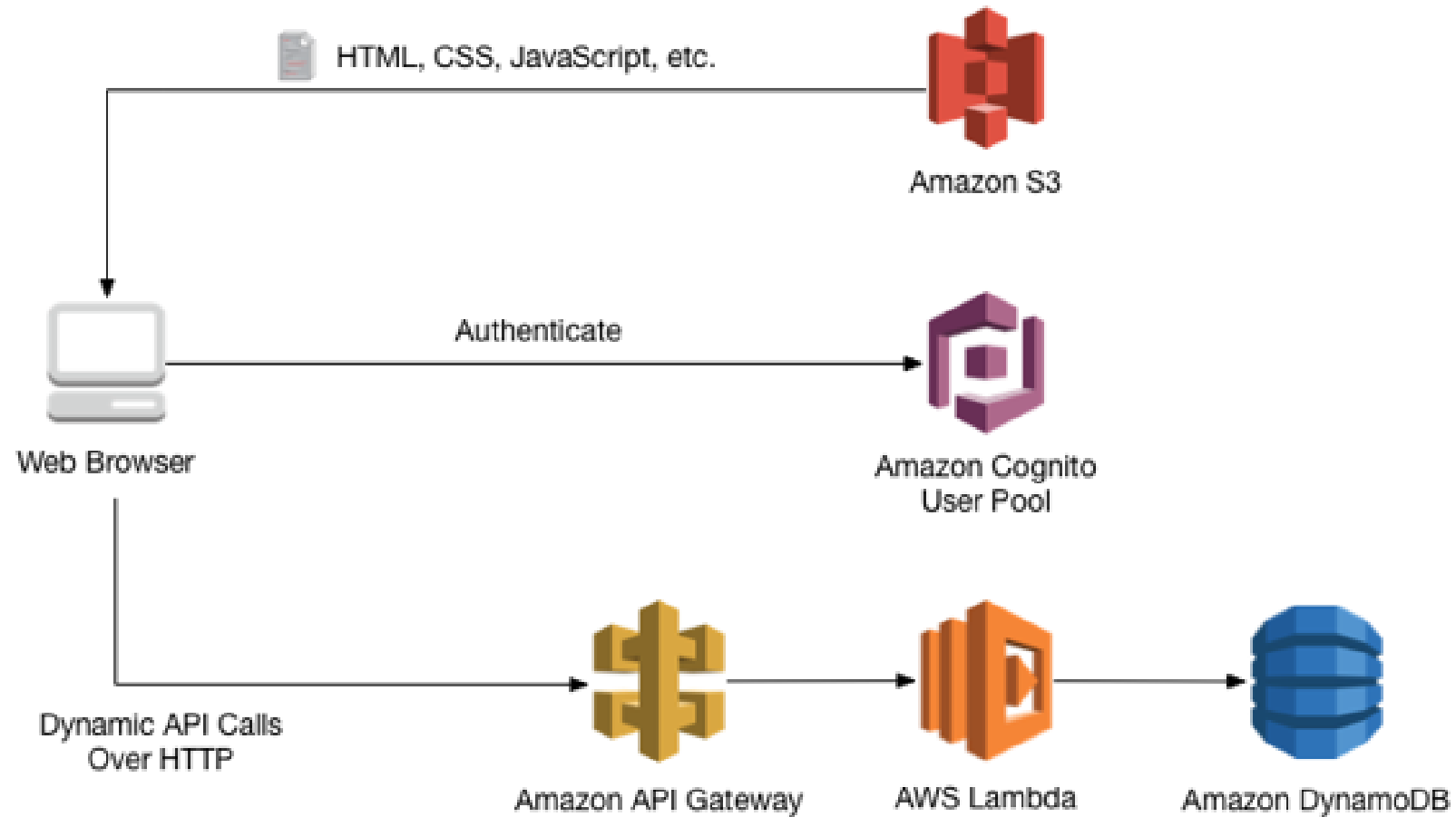
**Amazon ECS/Fargate**

**Amazon Lambda**

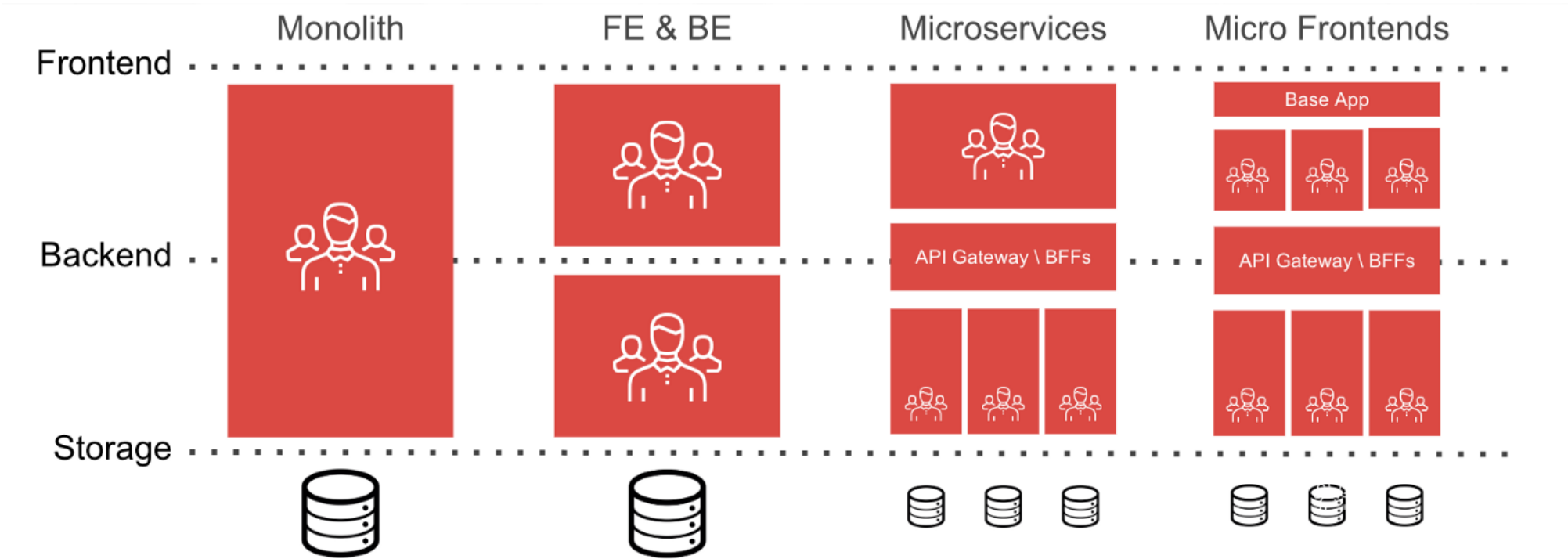
# Amazon ECS/Fargate



# Amazon Lambda

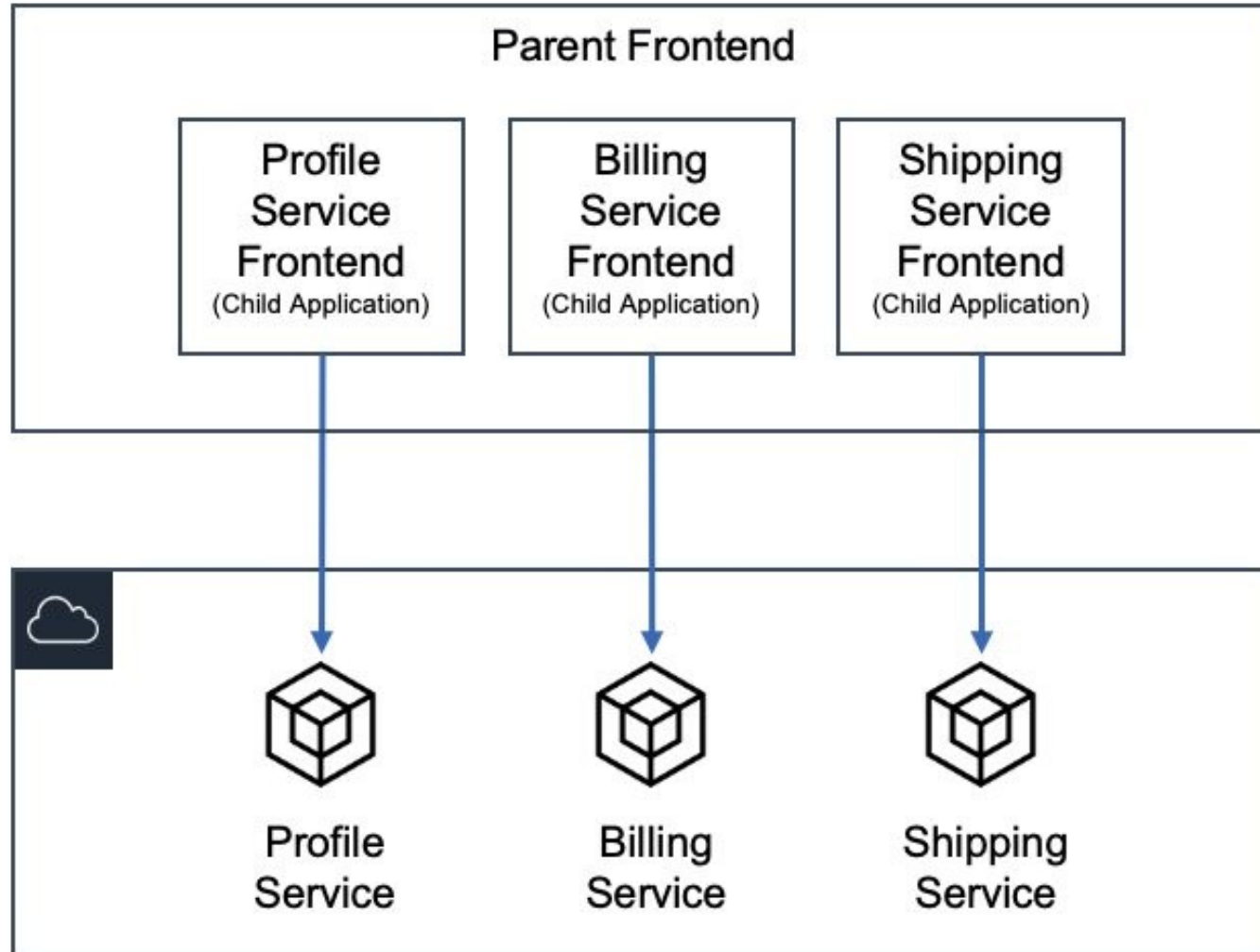


# Evolution of Software Architecture

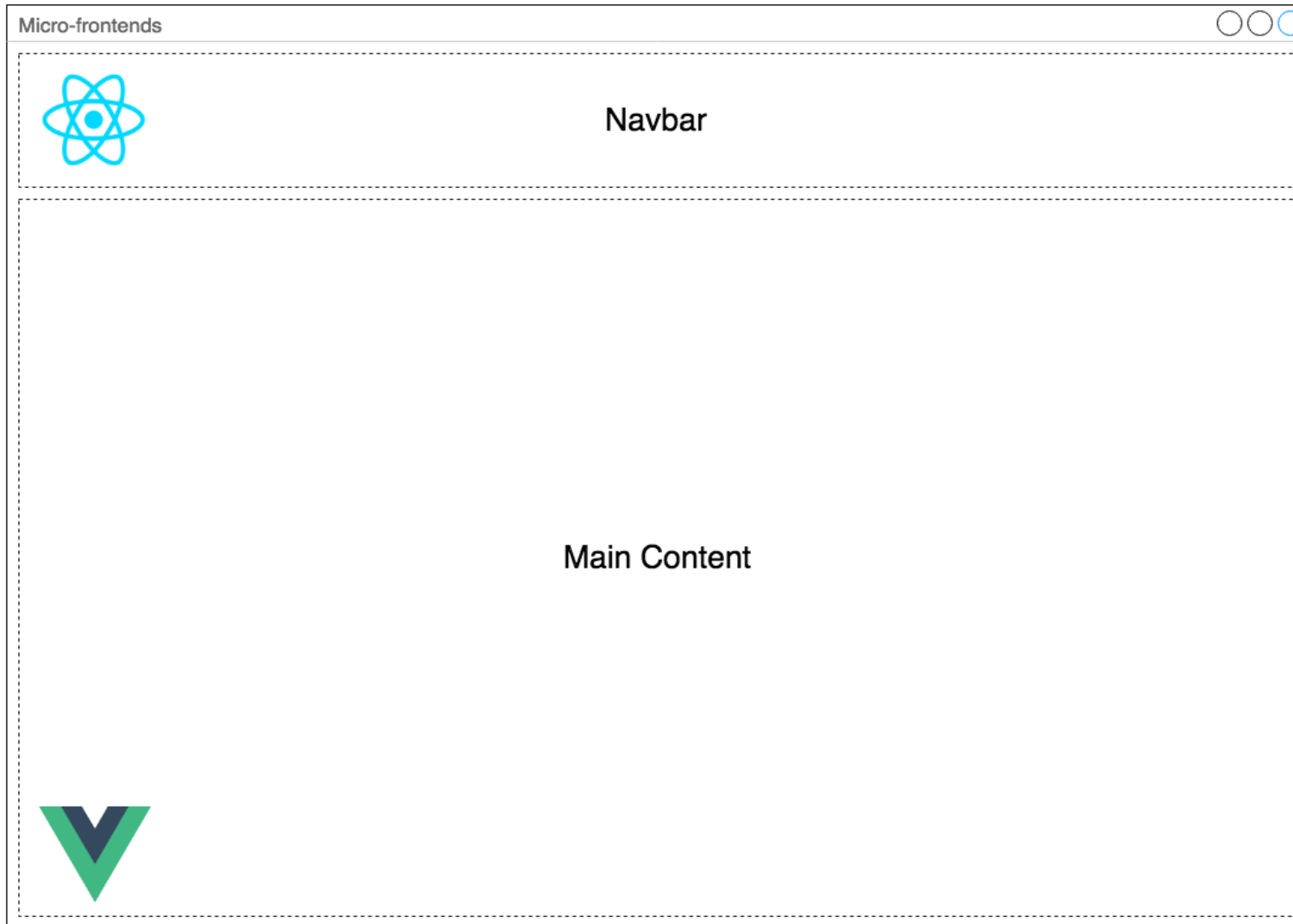




## Example: Micro-Frontend Architecture



# Example: Micro-Frontend UIs



# **Infrastructure Team Methodologies and Toolings**

**DEMO**

# And we have this Meme!

The Developer

Apr 30 · 🌐

Boss: "Write better comments."

```
14
15 import Foundation
16 import Markdown
17 import SwiftUI
18
19 // Link resolution works in two parts:
20 //
21 // 1. When DocC compiles a documentation bundle and encounters an "external" reference it will call out to
22 //    resolve that reference using the external resolver that's been registered for that bundle identifier.
23 //    The reference may be a page in another documentation bundle or a page from another source.
24 //
25 // 2. Once DocC has finished compiling the documentation bundle it will summarize all the pages and on-page
26 //    elements that can be linked to.
27 //    This information is external when another documentation bundle resolves a reference for that page.
28 //
29 //
30 //
31 //
32 //
33 //
34 //
35 //
36 //
37 //
38 //
39 //
40 //
41 //
42 //
43 //
44 //
45 //
46 //
47 //
48 //
49 //
50 //
51 //
52 //
53 //
54 //
55 //
56 //
57 //
58 //
59 //
60 //
61 //
62 //
63 //
64 //
65 //
66 //
67 //
68 //
69 //
70 //
71 //
72 //
73 //
74 //
75 //
76 //
77 //
78 //
79 //
80 //
81 //
82 //
83 //
84 //
85 //
86 //
87 //
88 //
89 //
90 //
91 //
92 //
93 //
94 //
95 //
96 //
97 //
98 //
99 //
100 //
```

```
66 /// A summary of an element that you can link to from outside the documentation bundle.
67 ///
68 /// The non-optional properties of this summary are all the information needed when another bundle references this element.
69 ///
70 /// Various information from the summary is used depending on what content references the summarized element. For example:
71 /// - In a paragraph of text, a link to this element will use the "title" as the link text and style the title in code font if the "kind" is a type of
72 ///   in a task group, the "title" and "abstract-and-contents" is displayed together to give more context about this element and the element may
73 ///   based on the values of its "platforms" and other metadata about the current versions of the platforms.
74 public struct LinkDestinationSummary: Codable, Equatable {
75     /// The kind of the summarized element.
76     public let kind: DocumentationNodeKind
77
78     /// The language of the summarized element.
79     public let language: SourceLanguage
80
81     /// The relative path to this element.
82     public let path: String
83
84     /// The resolved topic reference URL to this element.
85     public var referenceURL: URL?
86
87     /// The title of the summarized element.
88     public let title: String
89
90     /// An abstract in a single paragraph of rendered inline content.
91     public let abstract: Abstract < [RenderedInlineContent]>
92 }
```

Like

Comment

Share

1.1K