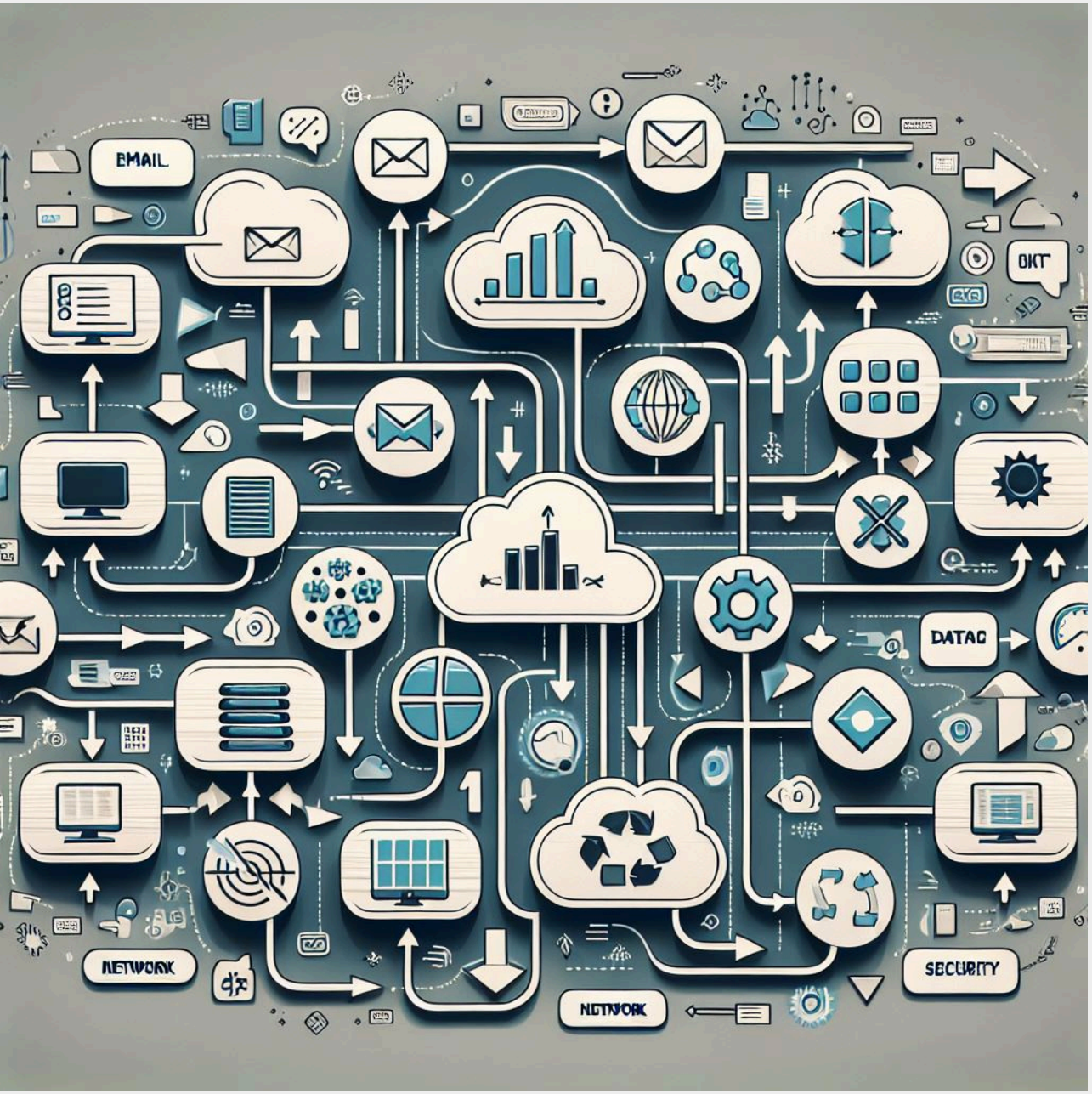
One Auth
to rule them all

Centralizing Authentication with Azure and API Gateway

Filippos Karailanidis
@filkaris

Who are we?

- Leading fintech company
- Specialize in Currency trading
- Offices in Cyprus, Greece, London, Dubai, Japan, USA
- Support 30 languages
- 10+ Million clients worldwide
- 1400+ employees / 600+ IT



Internal Services Landscape

- Older and newer systems
- Different UIs
- Different Technologies

Let's create a standard!

Set a standard

Spring of 2023 - common standard for internal applications

Set a standard

Spring of 2023 - common standard for internal applications

- Appearence

Set a standard

Spring of 2023 - common standard for internal applications

- Appearence
- Decoupled frontend/backend

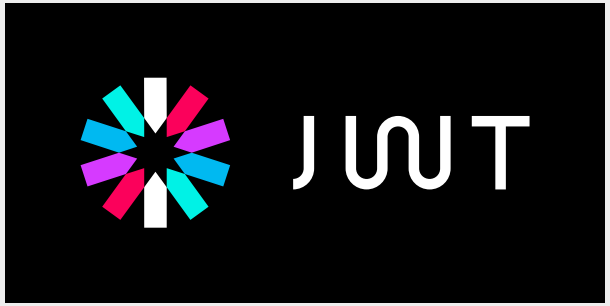
Set a standard

Spring of 2023 - common standard for internal applications

- Appearence
- Decoupled frontend/backend
- Authentication/Authorization

Why do you care?

AWS API Gateway

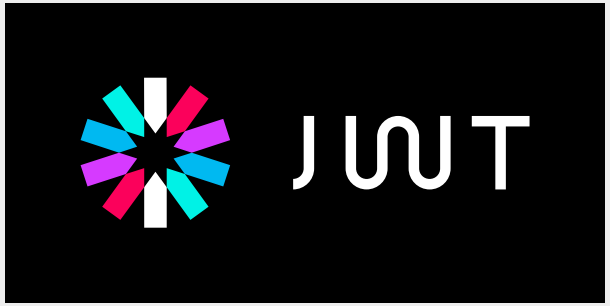


JWT

Azure Active Directory

Why do you care?

AWS API Gateway

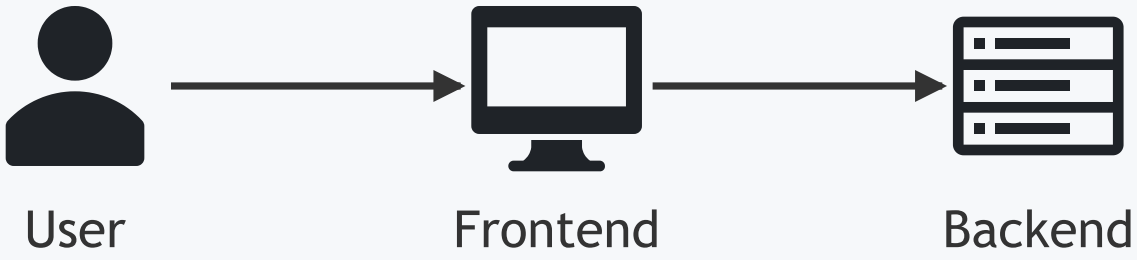


JWT

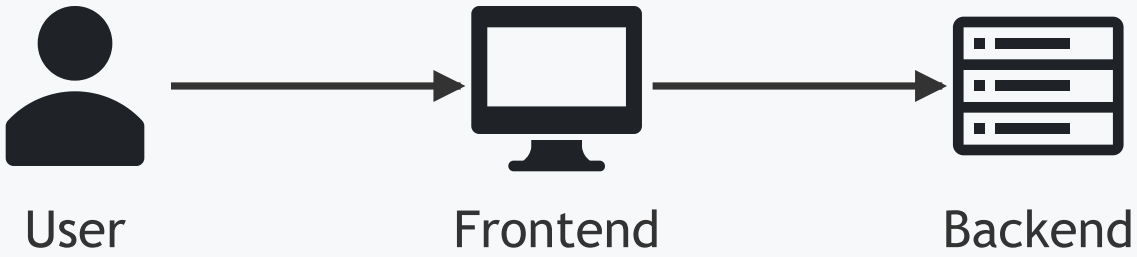
Azure Active Directory

Look out for 💡

Where we are now



Where we are now



Auth???

Quick Reminder,

Quick Reminder,

Authentication vs Authorization

Quick Reminder,

Authentication vs Authorization

Authentication = key = Can you enter the house?

Quick Reminder,

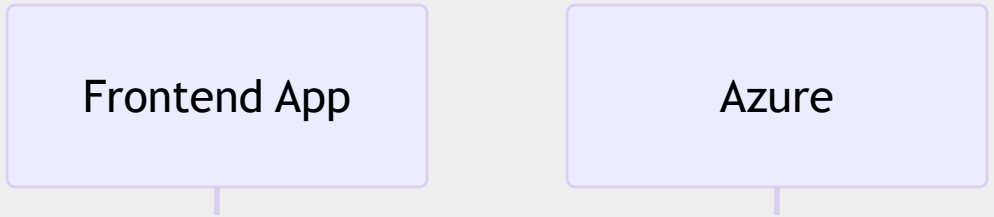
Authentication vs Authorization

Authentication = key = Can you enter the house?

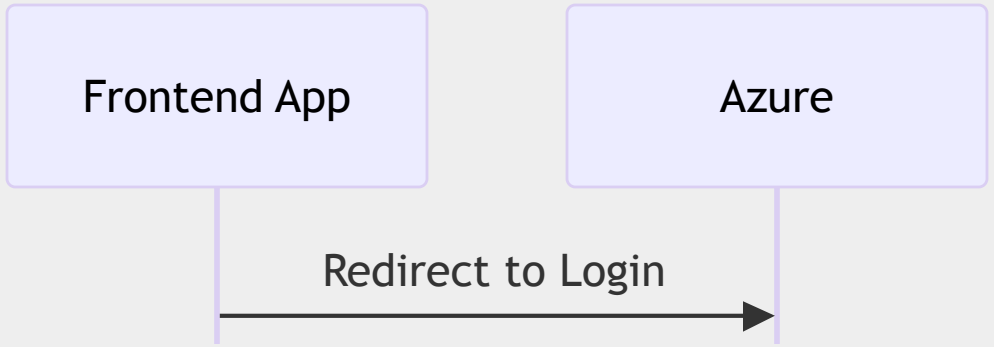
Authorization = identity = Can you pet the cat?


"YOU SHALL
NOT PASS"

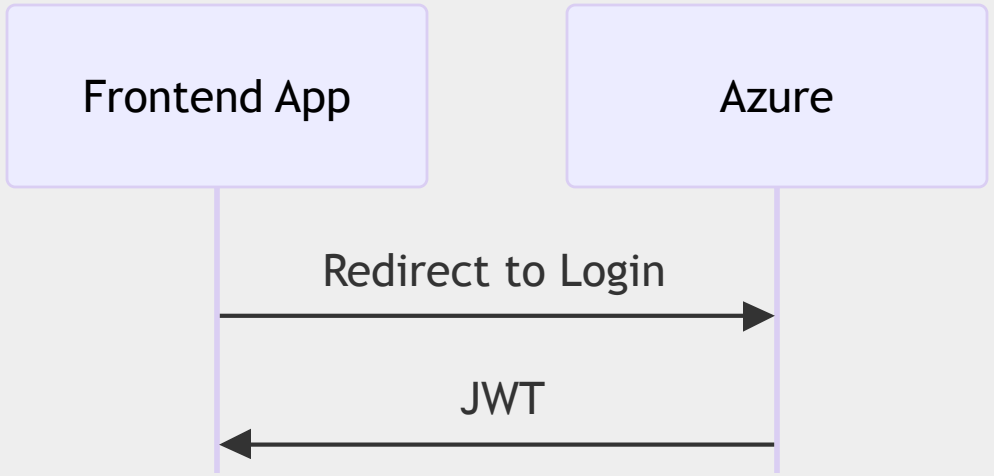
Authentication Flow



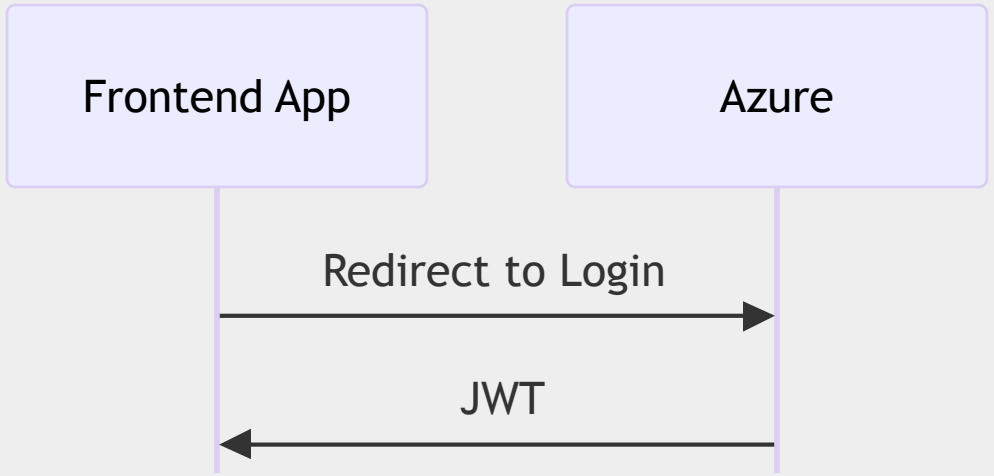
Authentication Flow



Authentication Flow

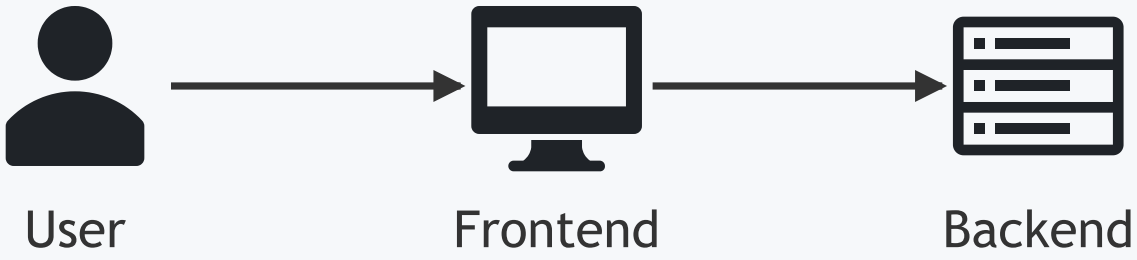


Authentication Flow

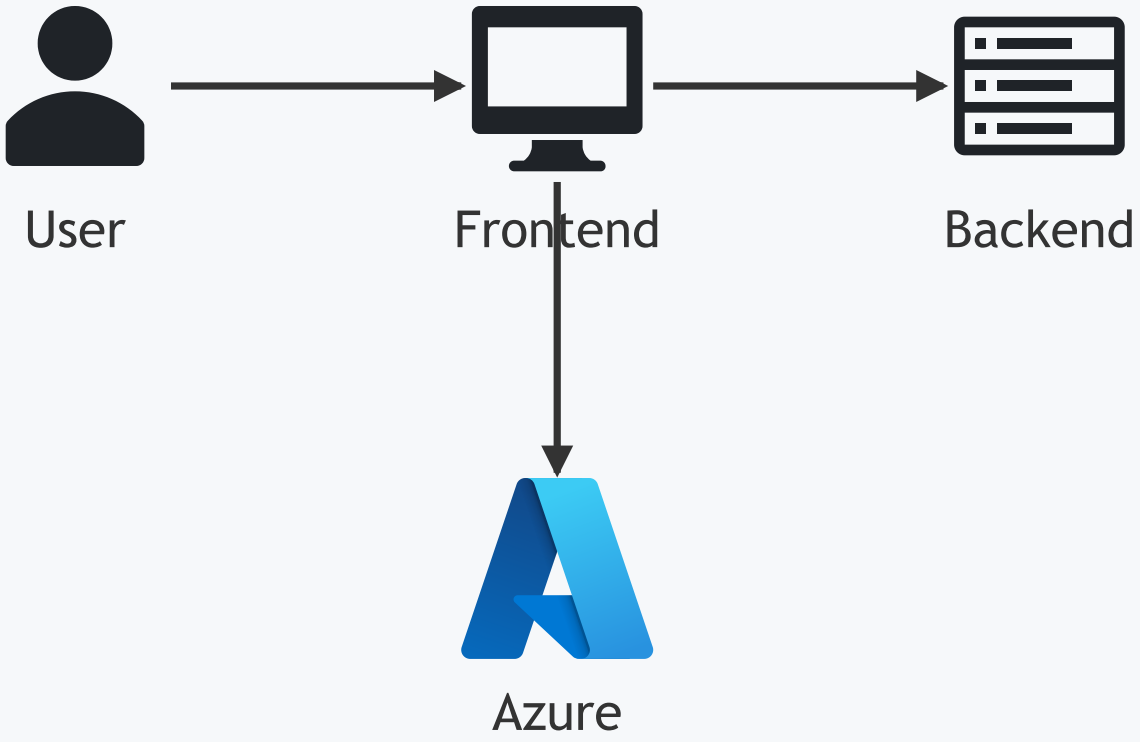


✅ Authentication

Where we are now

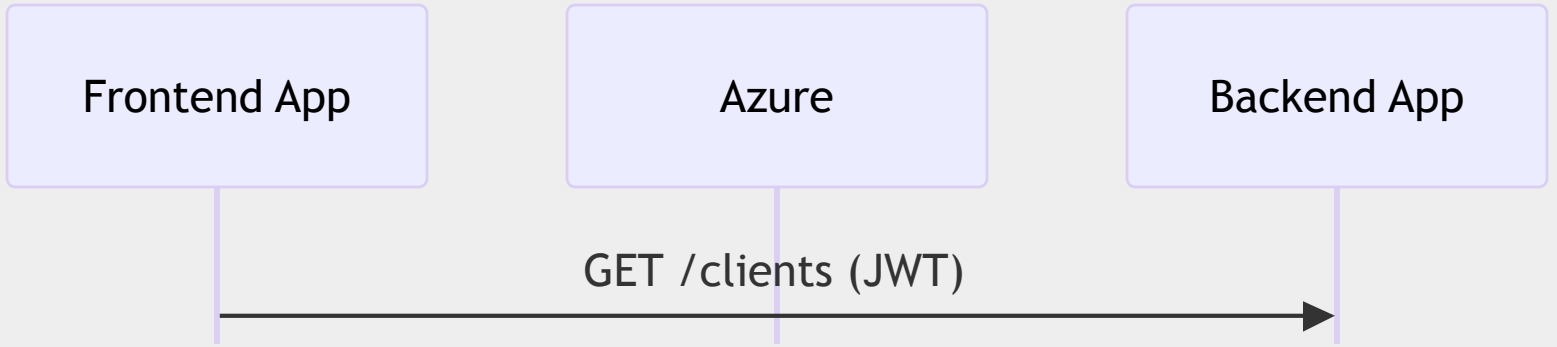


Where we are now + authentication

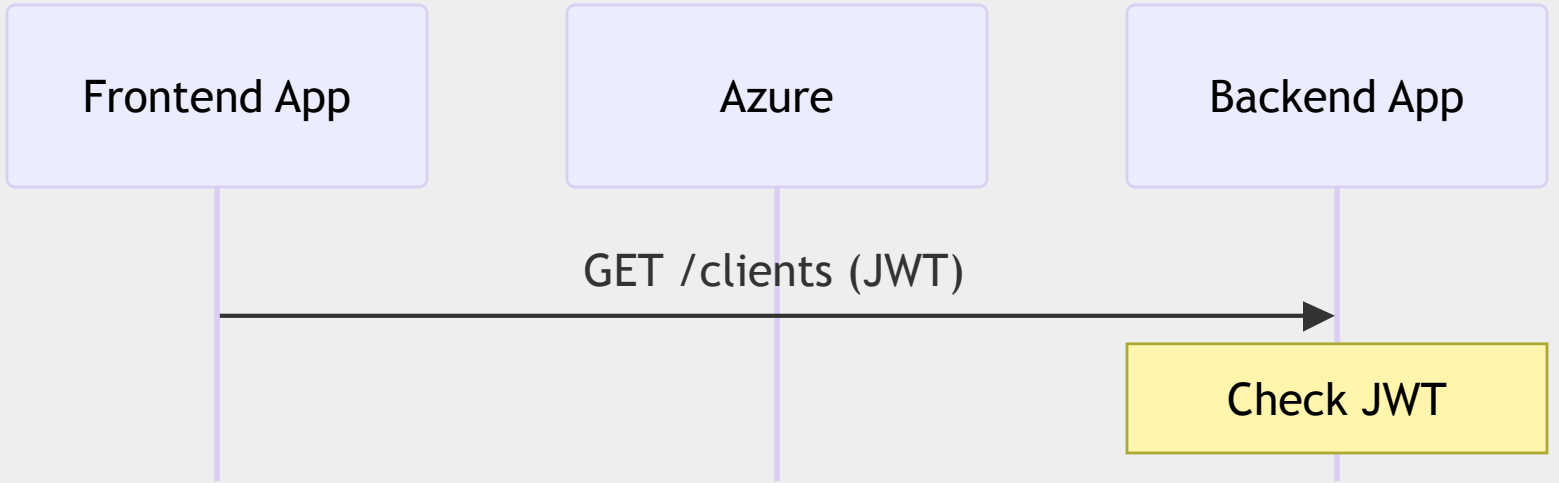


Azure

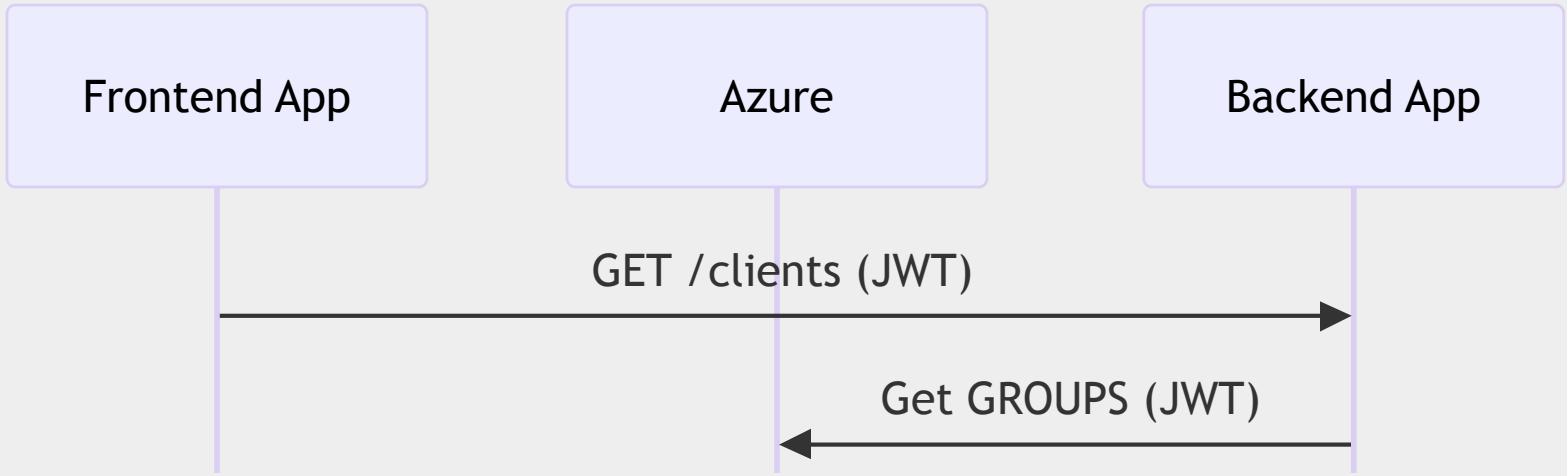
Request Flow



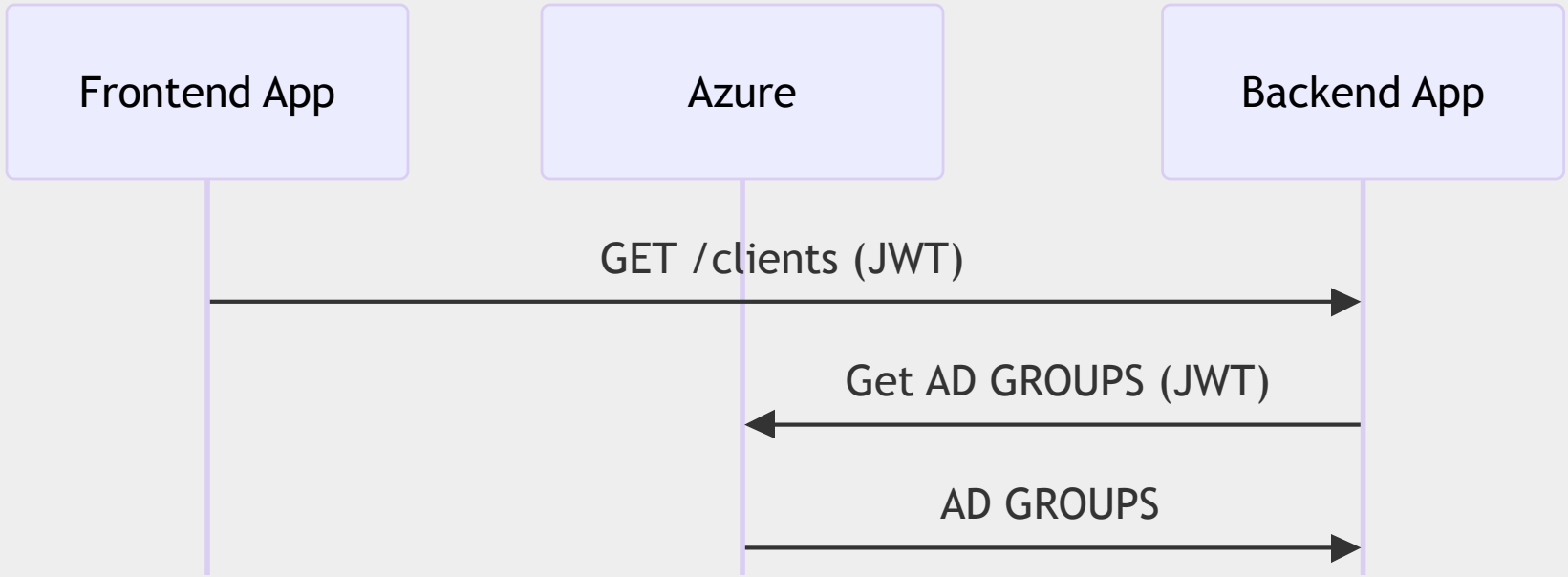
Request Flow



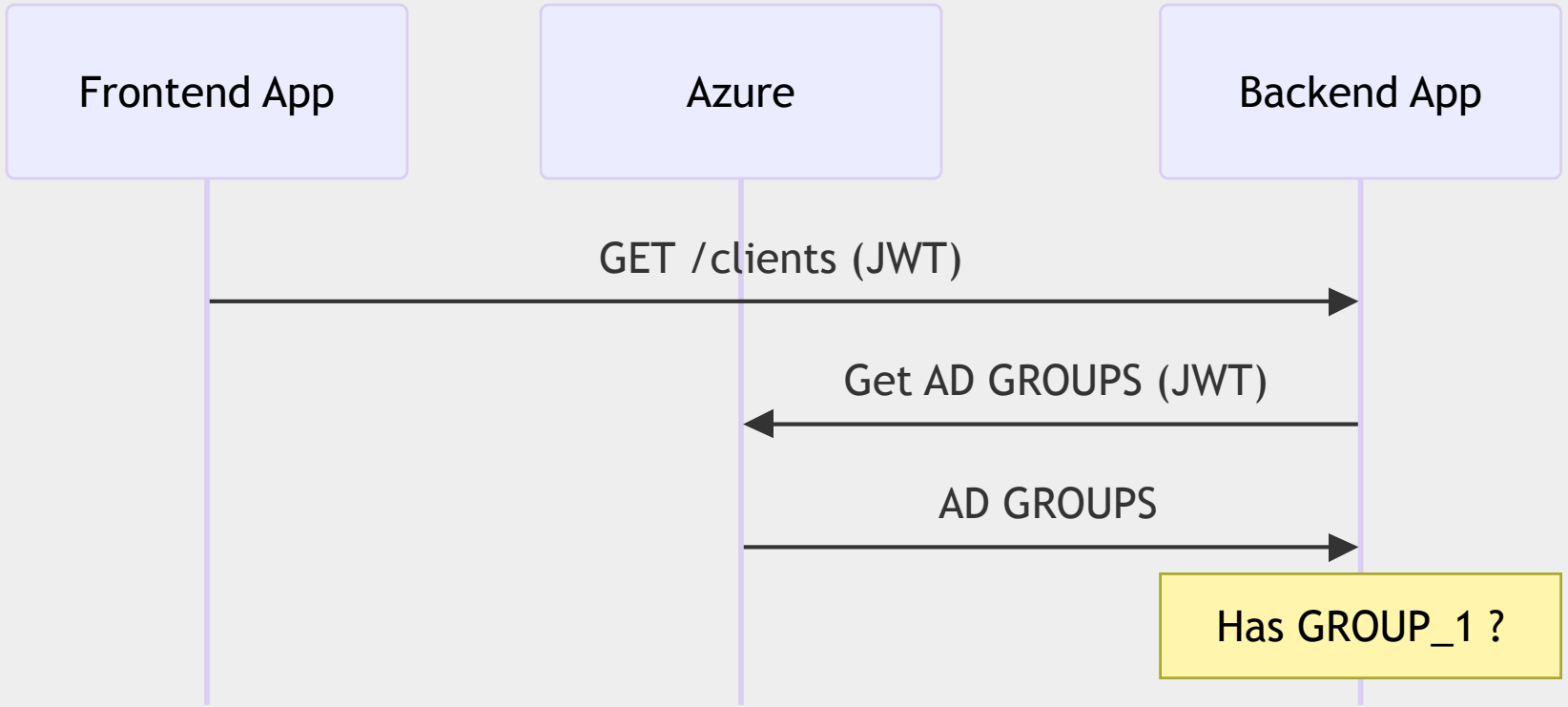
Request Flow



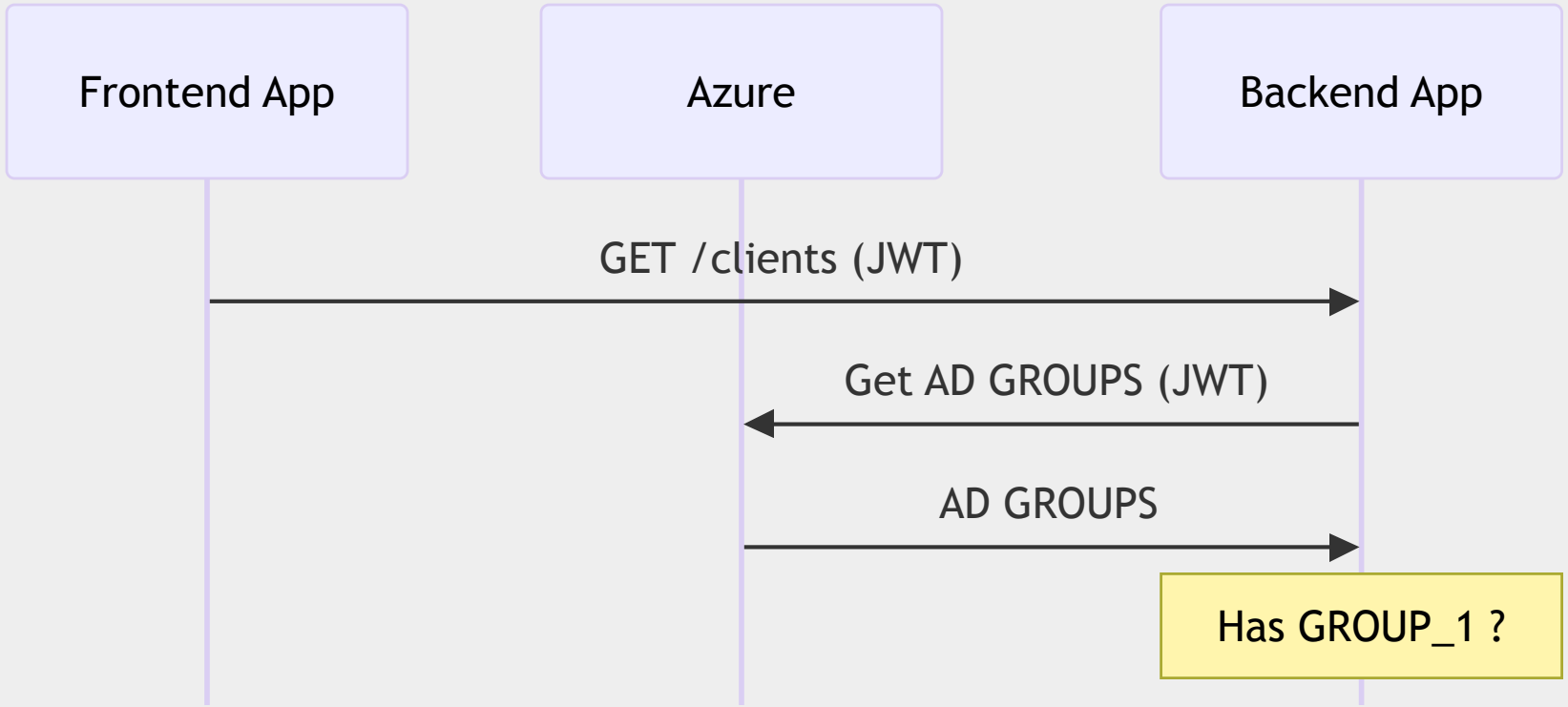
Request Flow



Request Flow

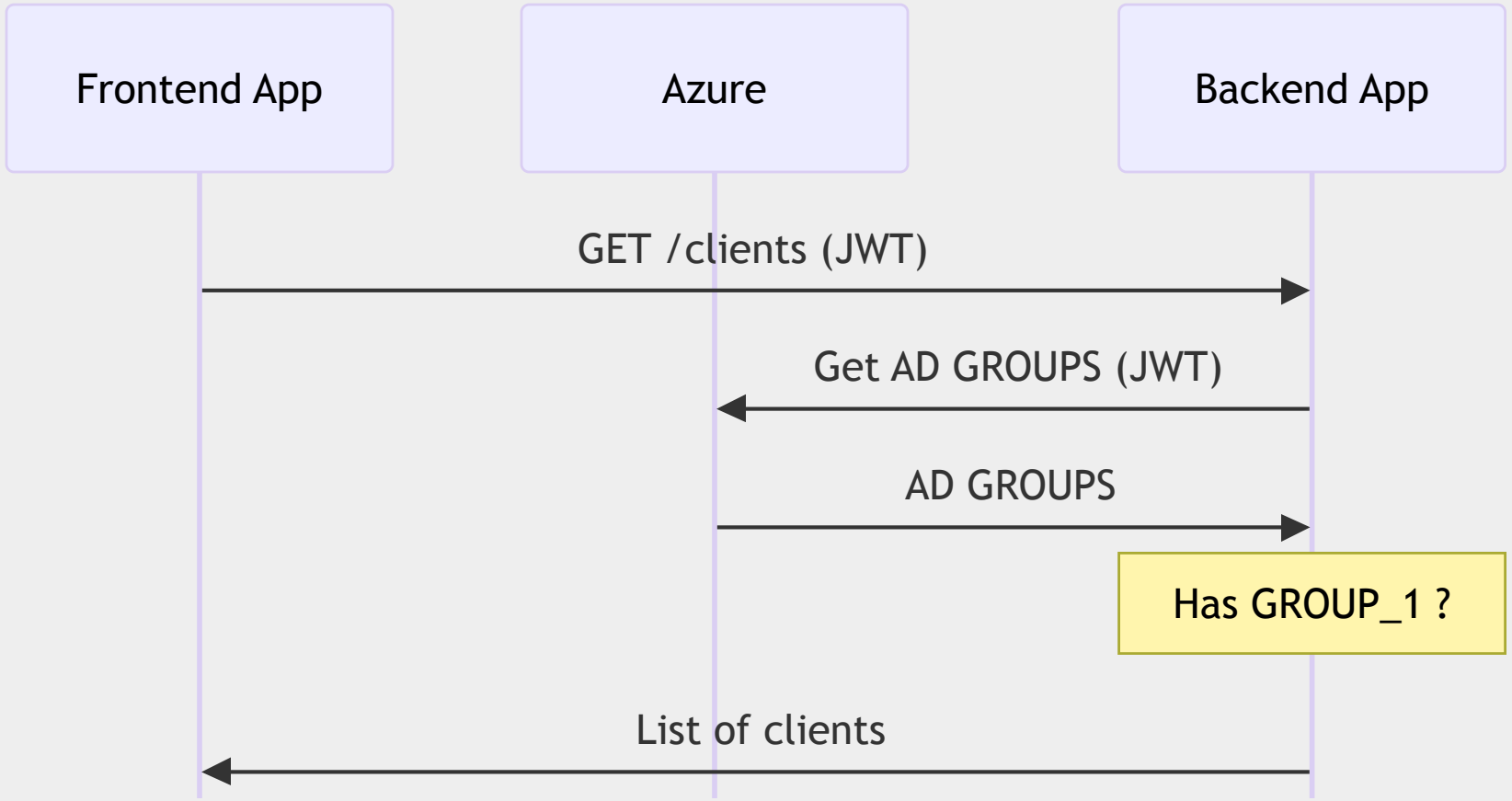


Request Flow

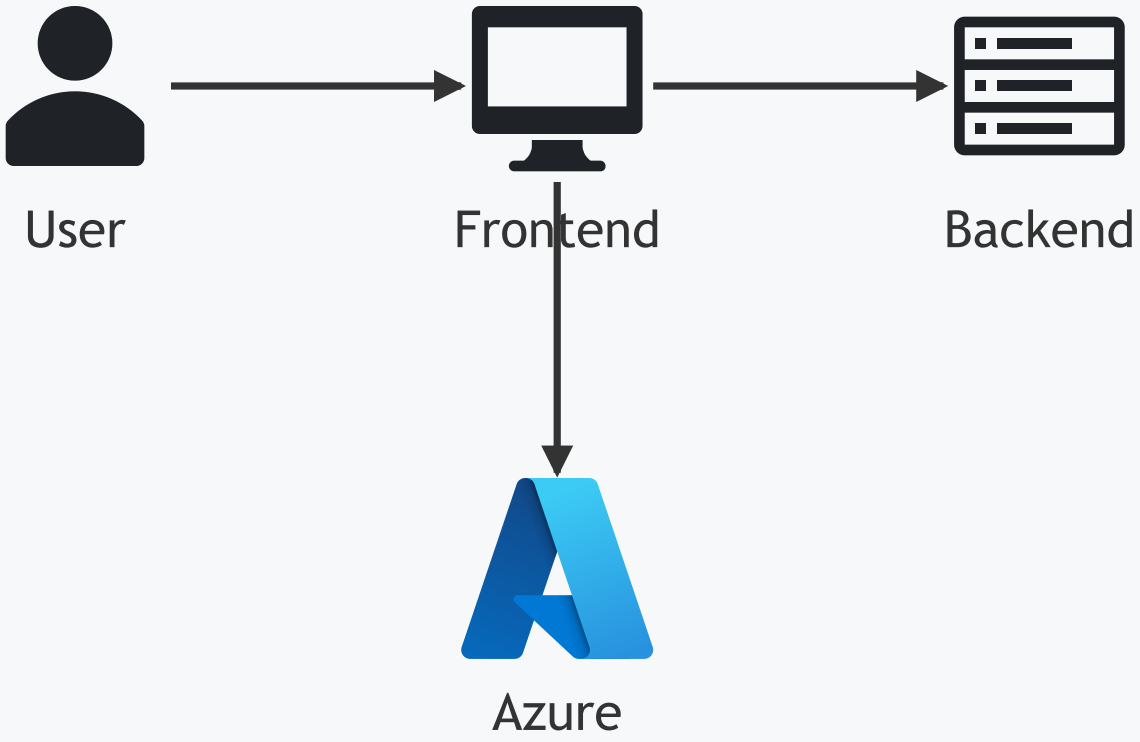


✅ Authorization

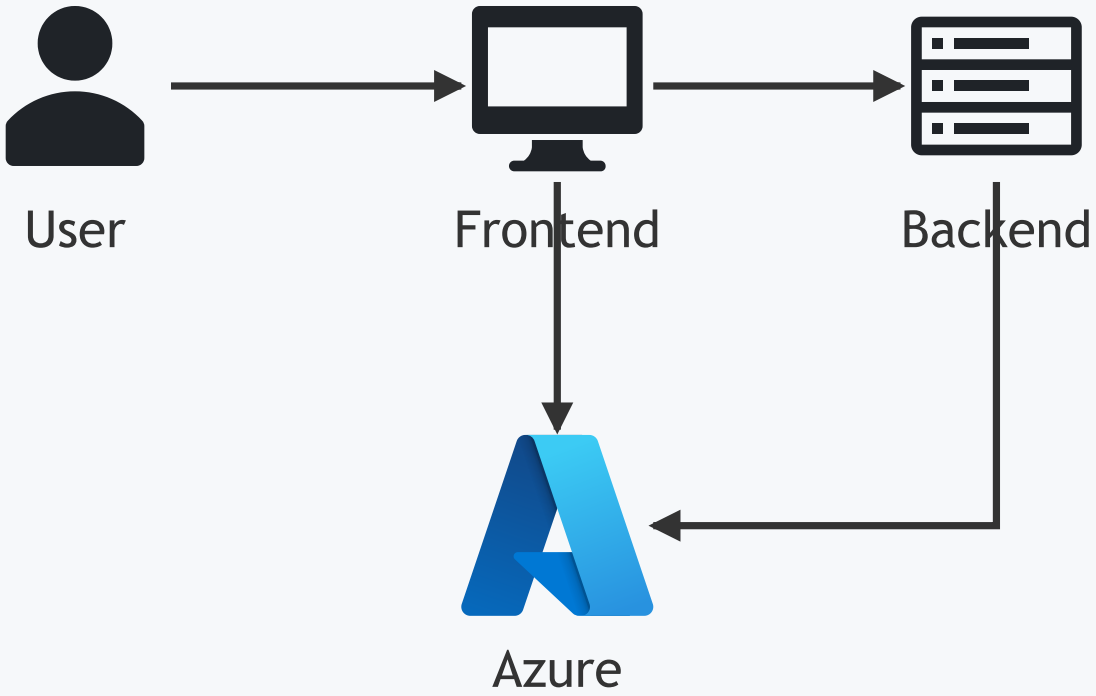
Request Flow



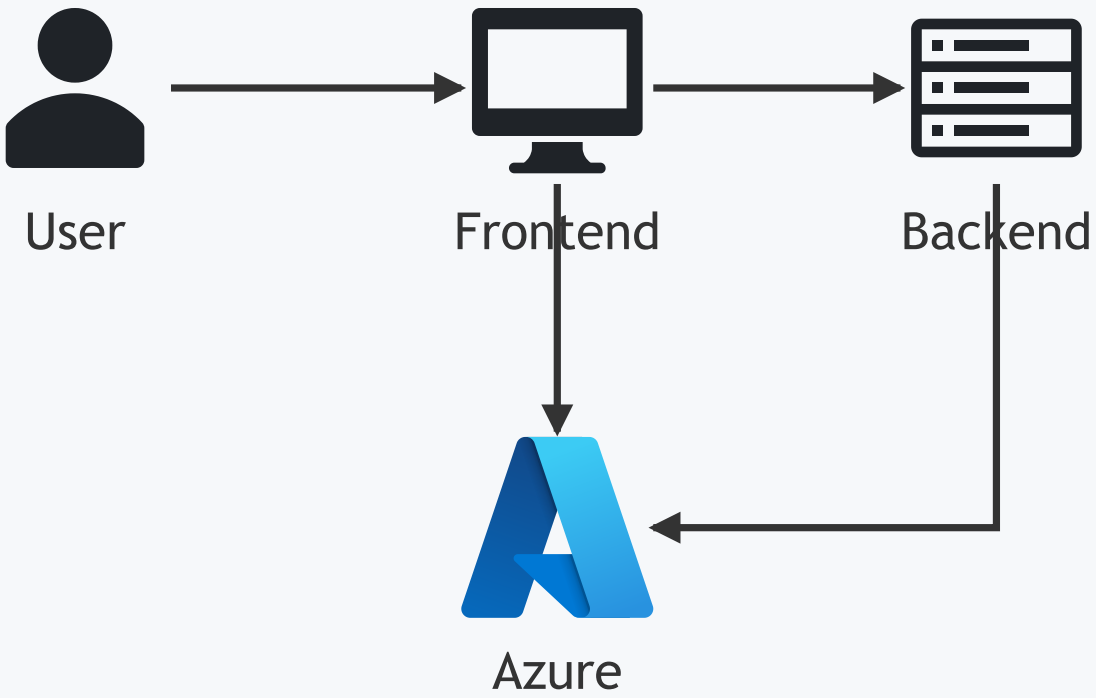
Where we are now + authentication



Where we are now + authorization



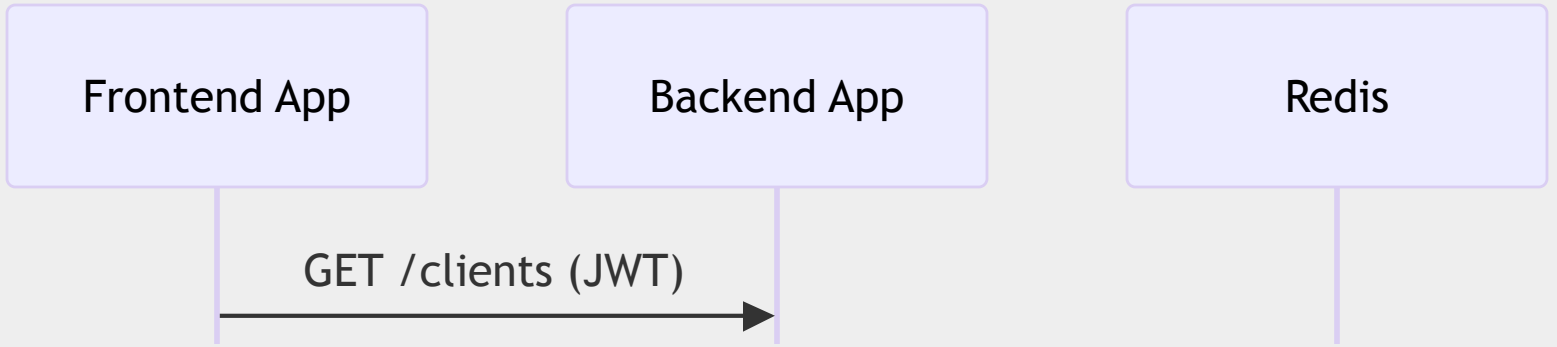
Where we are now + authorization



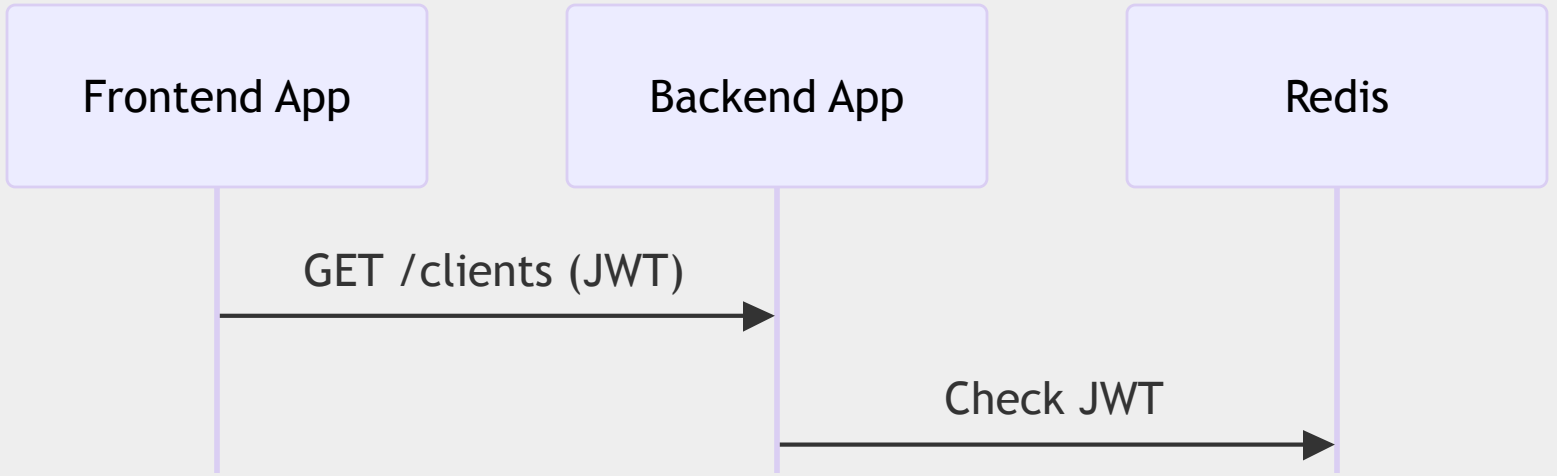
Performance...

Caching

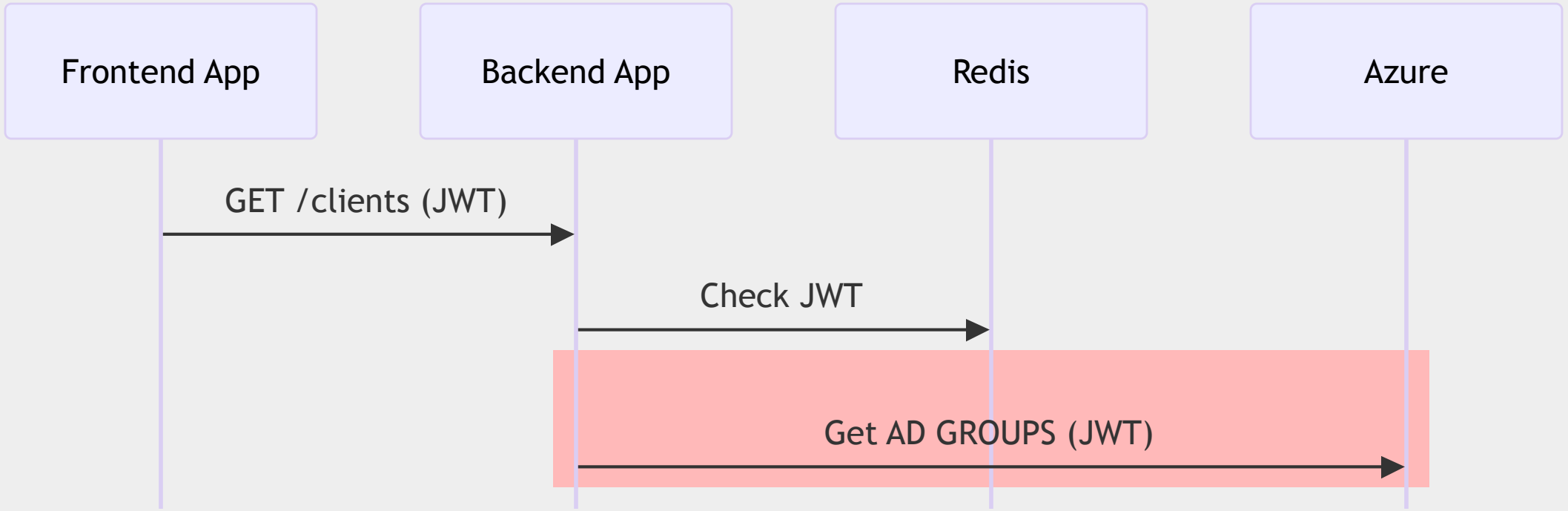
Caching



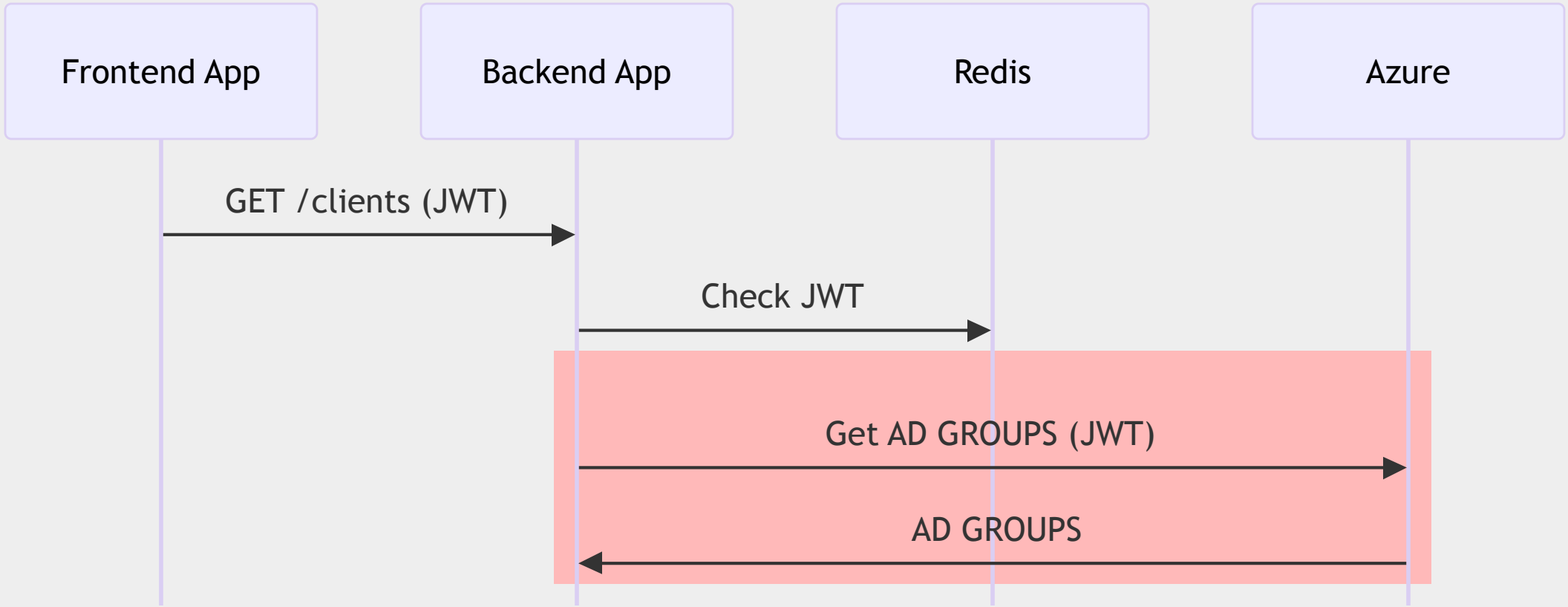
Caching



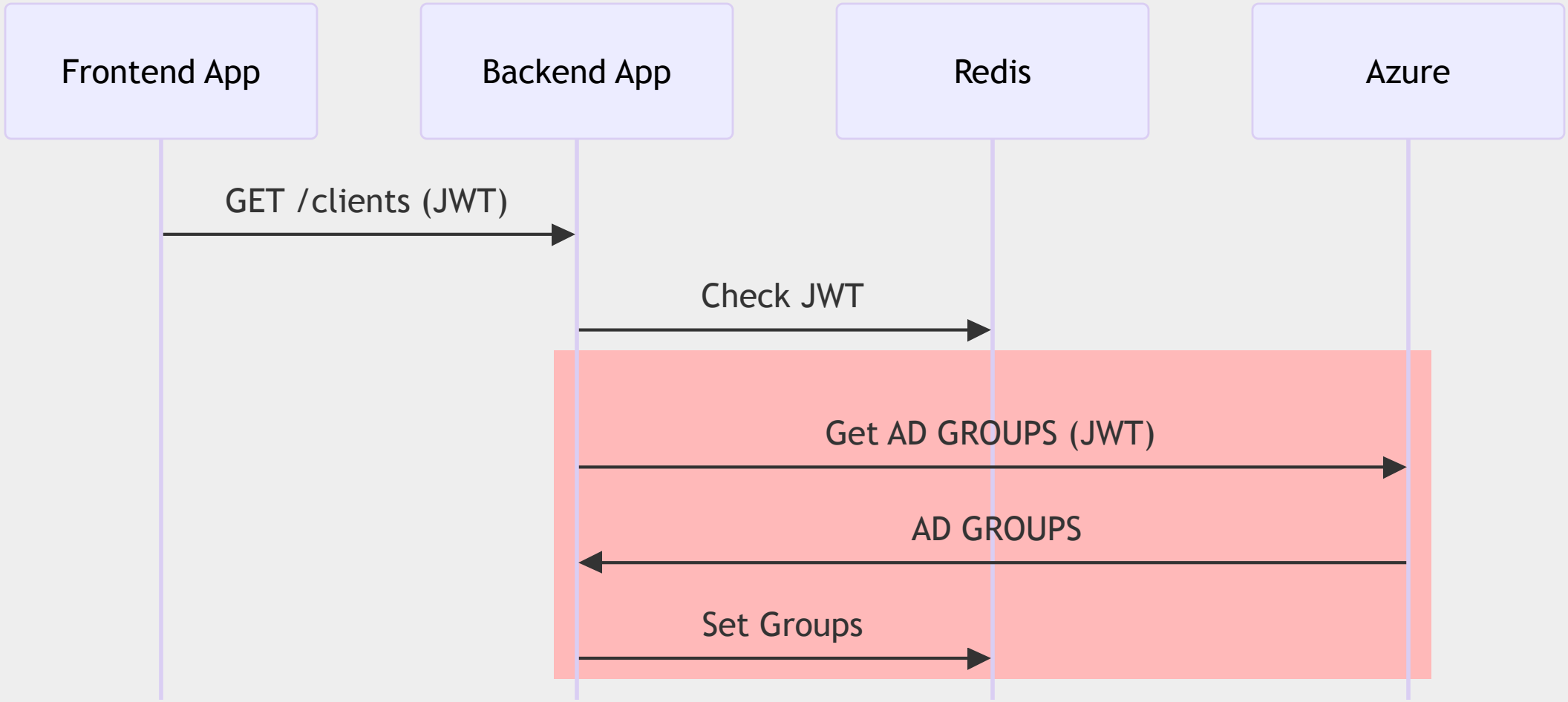
Caching


Frontend App
Backend App
Redis
Azure
GET /clients (JWT)
Check JWT
Get AD GROUPS (JWT)

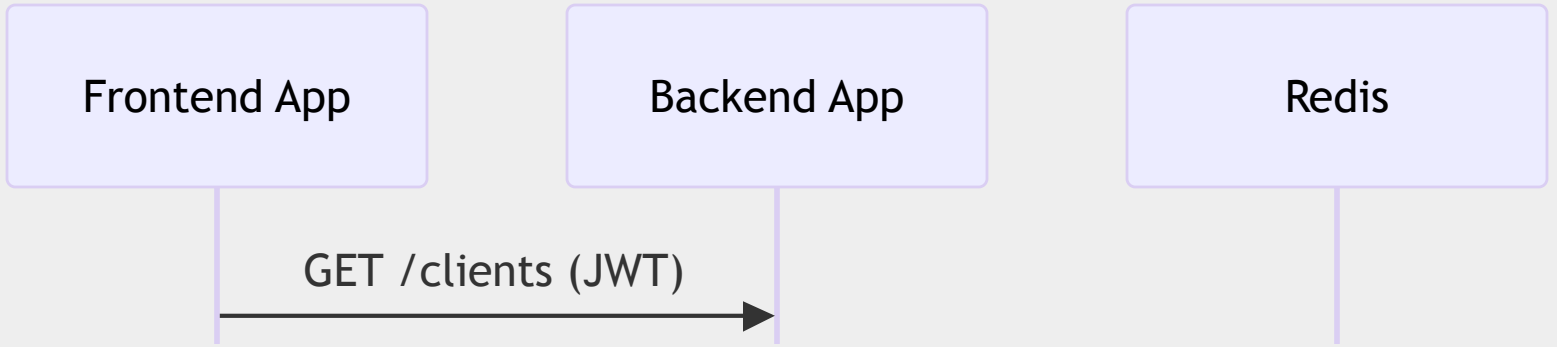
Caching


Frontend App
Backend App
Redis
Azure
GET /clients (JWT)
Check JWT
Get AD GROUPS (JWT)
AD GROUPS

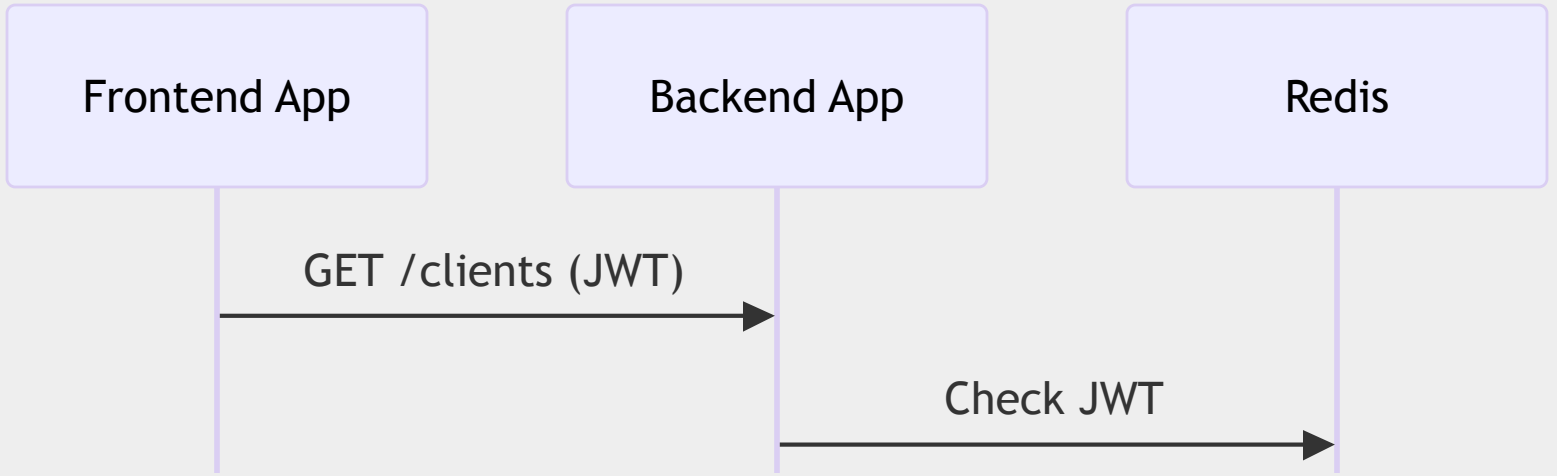
Caching


Frontend App
Backend App
Redis
Azure
GET /clients (JWT)
Check JWT
Get AD GROUPS (JWT)
AD GROUPS
Set Groups

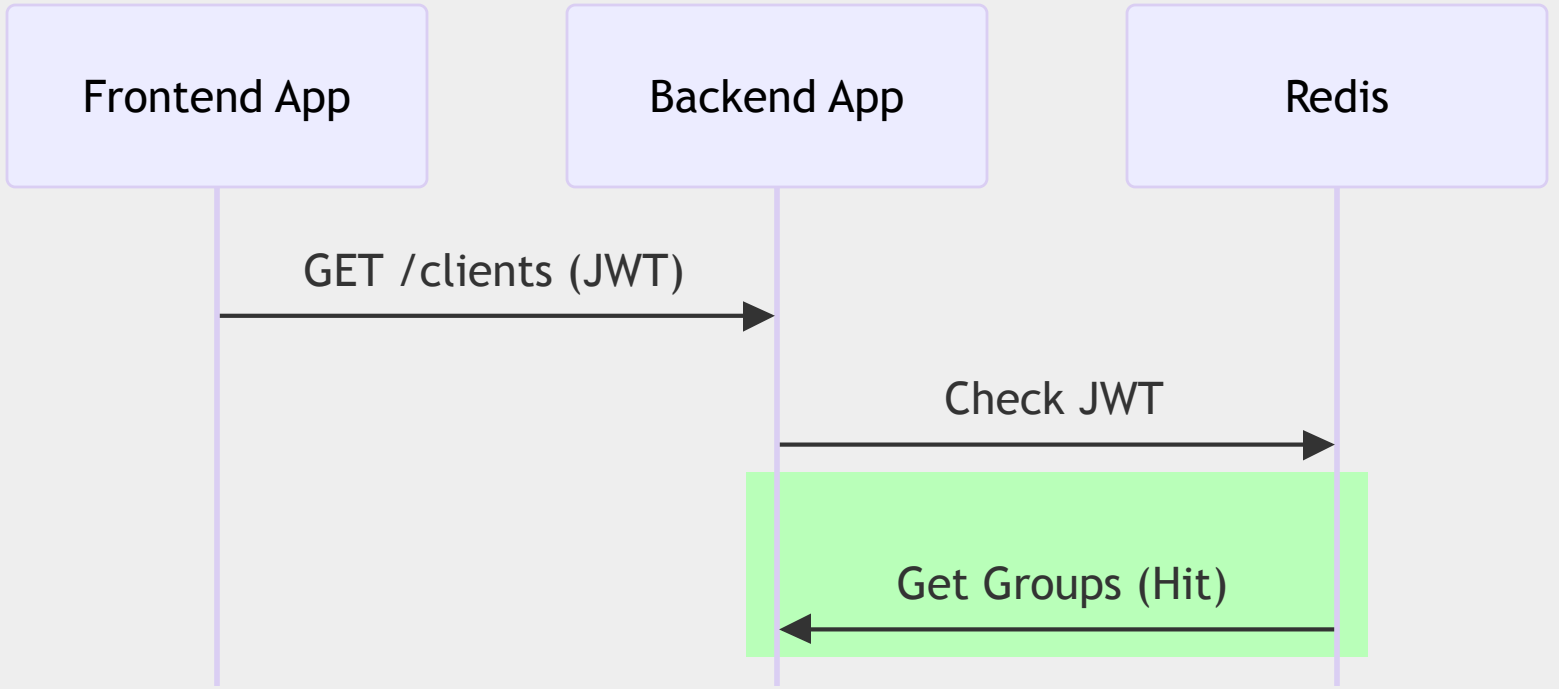
Caching



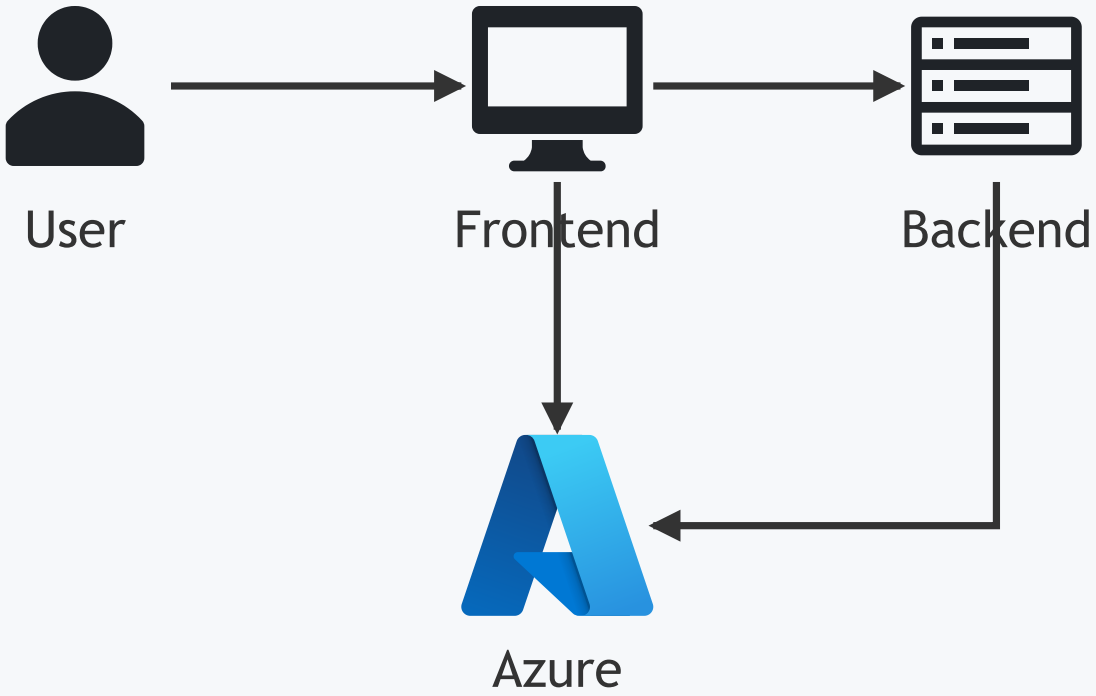
Caching



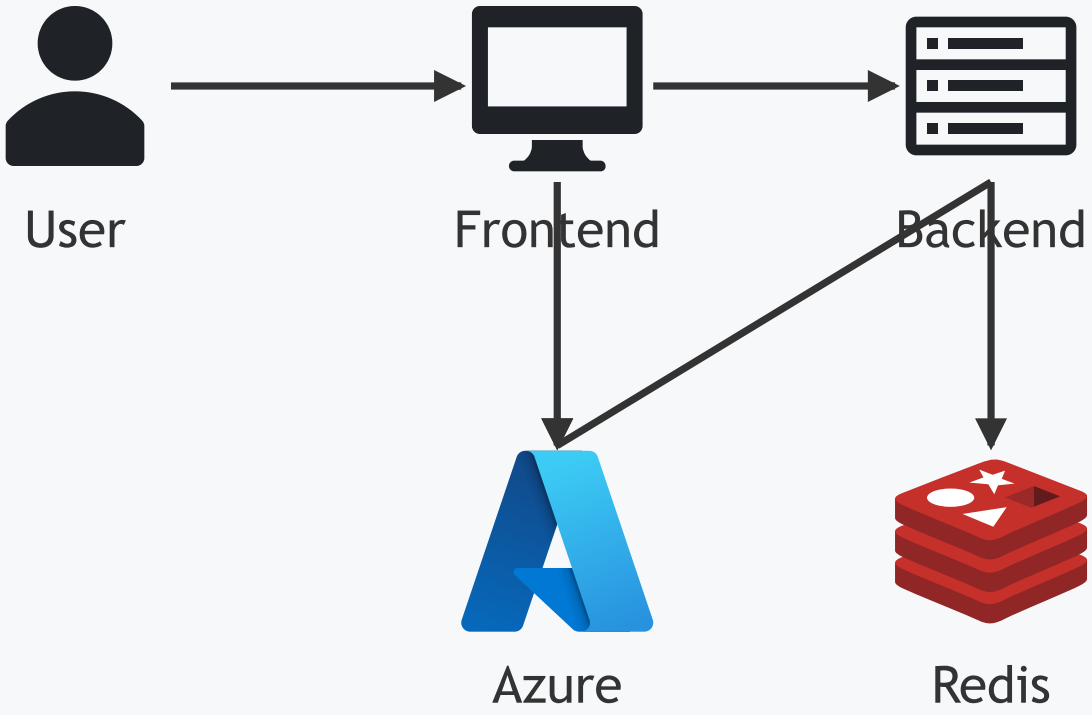
Caching


Frontend App
Backend App
Redis
GET /clients (JWT)
Check JWT
Get Groups (Hit)

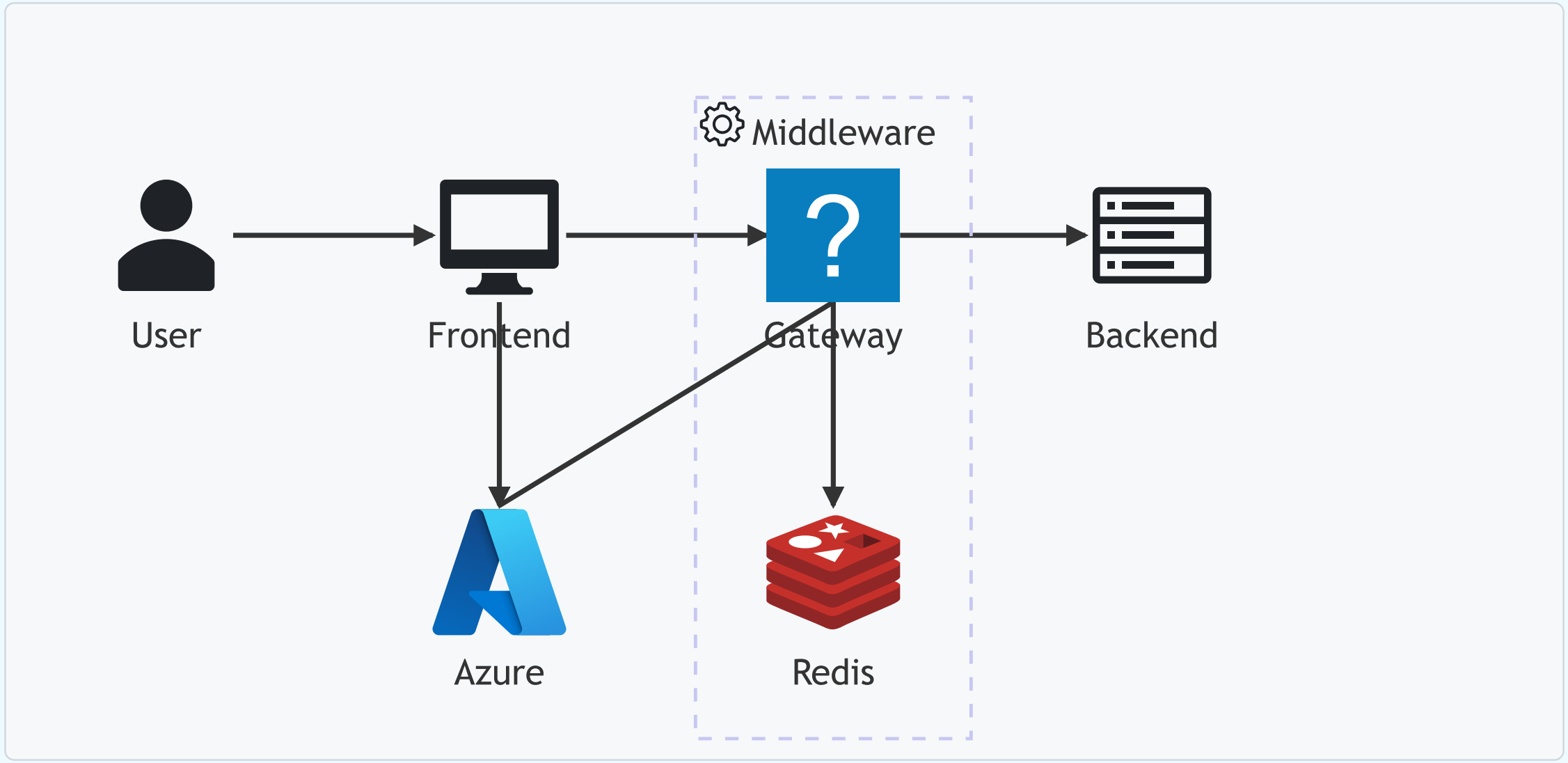
Where we are now + authorization



Where we are now + caching



Where we want to be




Middle Earth
North
Forodwaith
Forochel
Ered Forlindon
Arnor
Angmar of old
Grey Mountains
Iron Hills
Rhovanion
Rhûn
Sea of Rhûn
Mirkwood
Eriador
Blue Mountains
Lune
Gulf of Lune
the Shire
Hollin
Minhiriath
Eregion
Enedwaith
Dunland
Fangorn
the Brown Lands
Rohan
White Mountains
Gondor
Ered Nimrais
Ash Mountains
Gorgoroth
Mordor
Nurn
Sea of Nurnen
Mountains of Shadow
Khand
Bay of Belfalas
Harondor
Lebennin
Near Harad
Haradwaith
Havens of Umbar


Middle Earth
Ware
Forochel
Forodwaith
North
Grey Mountains
Iron Hills
Rhovanion
Rhûn
Sea of Rhûn
Eriador
Arnor
Angmar of old
Hills of Evendim
Lake Evendim
Blue Mountains
Rhudaur
Misty Mountains
Mirkwood
Forlindon
Lune
Gulf of Lune
the Shire
Harlindon
Minhiriath
Eregion
Enedwaith
Dunland
Fangorn
the Brown Lands
Rohan
Isen
Gap of Rohan
Helm's Deep
Ered Nimrais
White Mountains
Ash Mountains
Gorgoroth
Mordor
Nurn
Sea of Nurnen
Mountains of Shadow
Khand
Near Harad
Harondor
Bay of Belfalas
Lebennin
Anfalas
Pinnath Gelin
Ered Lithui
Harnadwaith
City of the Corsairs
Havens of Umbar


Gateway
Middle Earth
Ware
Forochel
Forodwaith
Grey Mountains
Angmar
Arnor
Eriador
Rhovanion
Rhûn
Sea of Rhûn
Mirkwood
Misty Mountains
Iron Hills
North
Lune
Forlindon
Harlindon
Gulf of Lune
Minhiriath
Enedwaith
Dunland
Fangorn
Rohan
White Mountains
Ash Mountains
Gondor
Mordor
Nurn
Sea of Nurnen
Mountains of Shadow
Gorgoroth
Anfalas
Belfalas
Bay of Belfalas
Harondor
Khând
Near Harad
Haradwaith
City of the Corsairs
Havens of Umbar
The Brown Lands

Gateway Implementation

Gateway Implementation

AWS API Gateway

Gateway Implementation

AWS API Gateway

Single "access point"

Can talk to multiple Backends

Gateway Implementation

AWS API Gateway

Single "access point"

Can talk to multiple Backends

+ Lambda Authorizer

Gateway Implementation

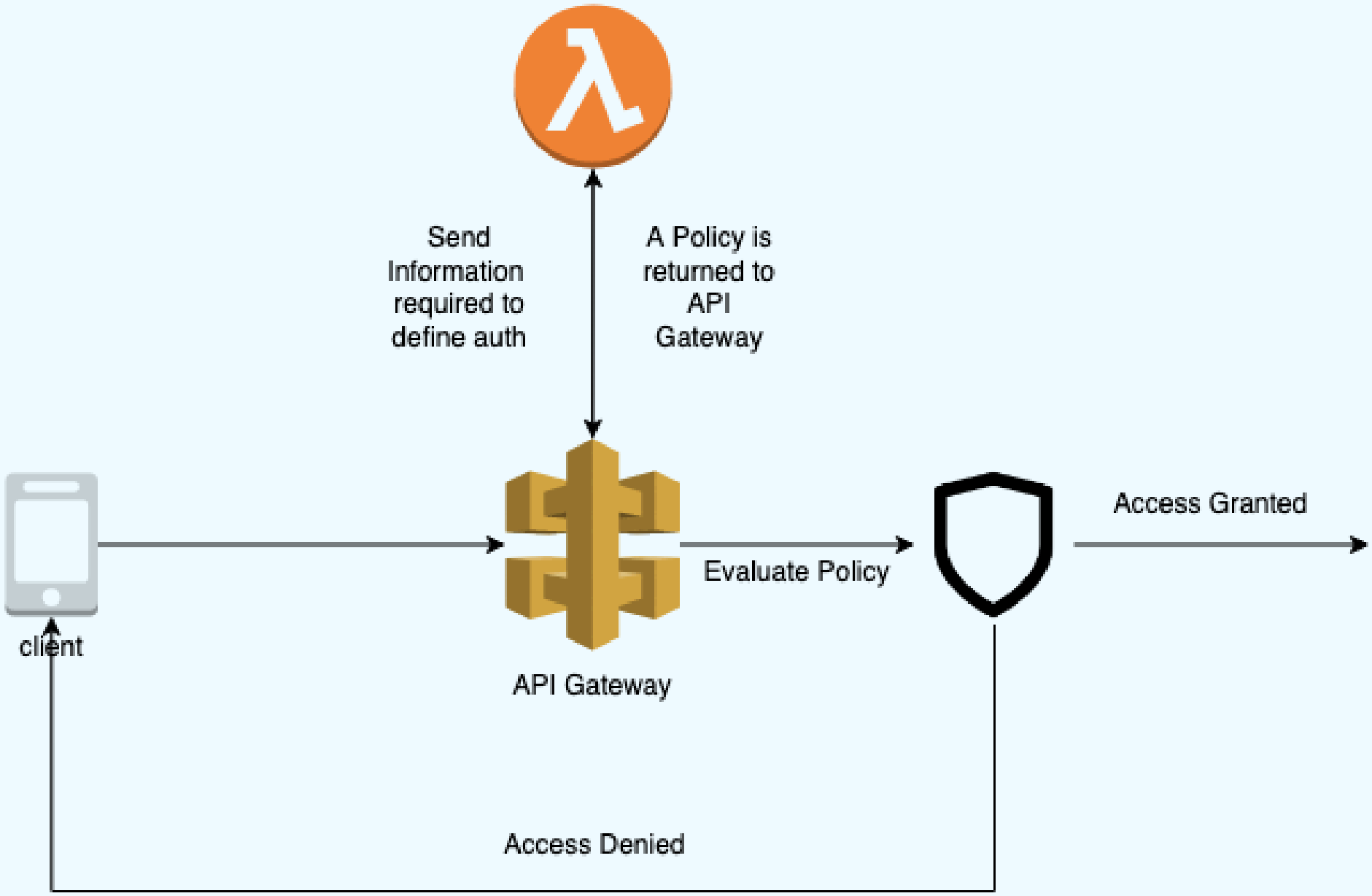
AWS API Gateway

Single "access point"

Can talk to multiple Backends

+ Lambda Authorizer

Checks the JWT


Send Information required to define auth
A Policy is returned to API Gateway
client
API Gateway
Evaluate Policy
Access Granted
Access Denied

Gateway Implementation

Lambda returns a policy

Gateway Implementation

Lambda returns a policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": ["execute-api:Invoke"],
            "Effect": "Allow",
            "Resource": ["user"]
        }
    ]
}
```

Gateway Implementation

Lambda returns a policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": ["execute-api:Invoke"],
            "Effect": "Allow",
            "Resource": ["user"]
        }
    ]
}
```

What about `GROUP_1, GROUP_2` as HTTP headers?

💡 Gateway Implementation - Context

💡 Gateway Implementation - Context

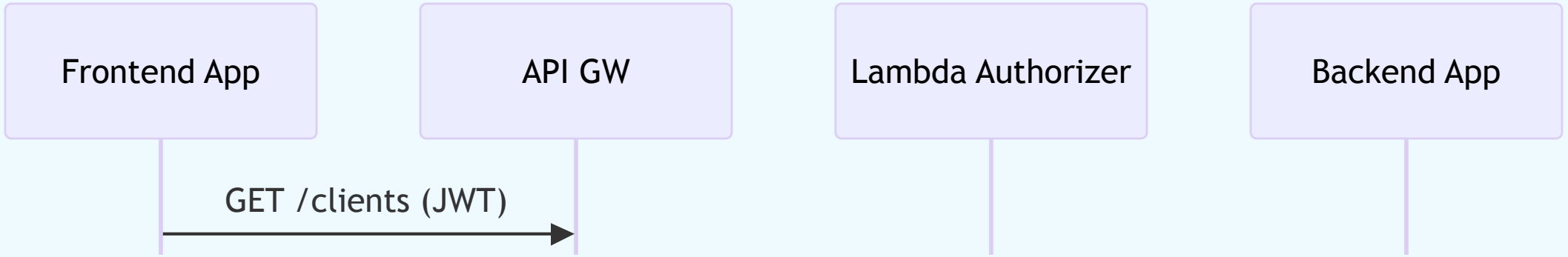
```
// lambda.go

authResponse := events.APIGatewayCustomAuthorizerResponse{PrincipalID: principalID}

authResponse.PolicyDocument = ...
authResponse.Context = map[string]interface{}{
    "groups": "GROUP_1,GROUP_2",
}

return authResponse
```

💡 Gateway Implementation - Context

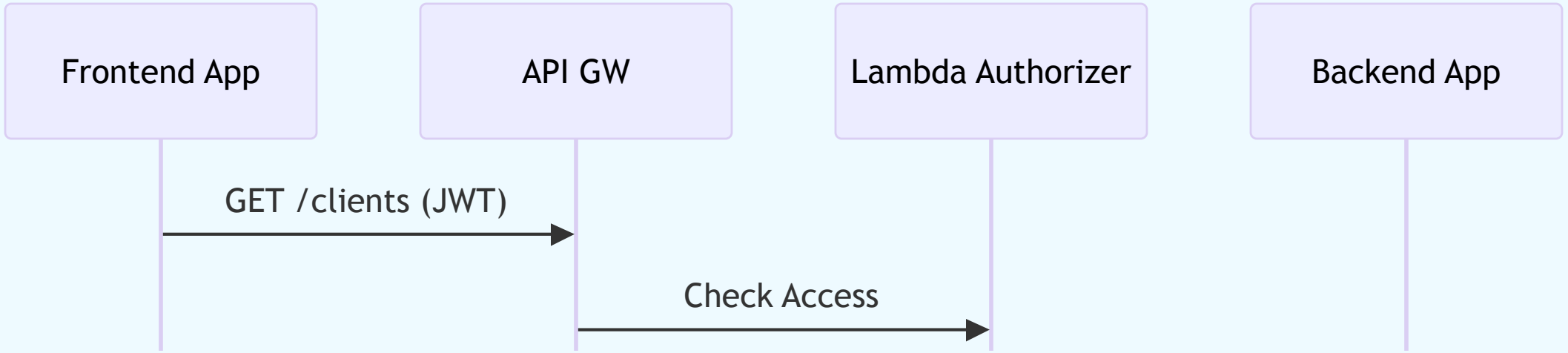
```
// lambda.go

authResponse := events.APIGatewayCustomAuthorizerResponse{PrincipalID: principalID}

authResponse.PolicyDocument = ...
authResponse.Context = map[string]interface{}{
    "groups": "GROUP_1,GROUP_2",
}

return authResponse
```

```
# api-gateway-definition.yaml

x-amazon-apigateway-any-method:
    ...
    requestParameters:
      integration.request.header.X-Groups: "context.authorizer.groups"
```

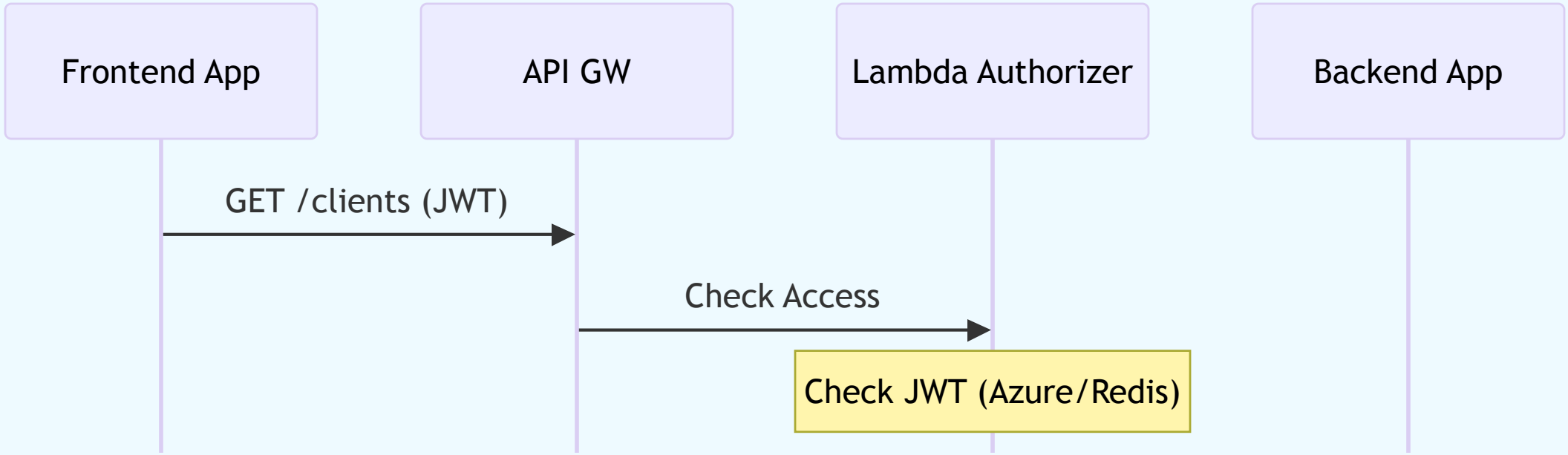
Request flow - Gateway



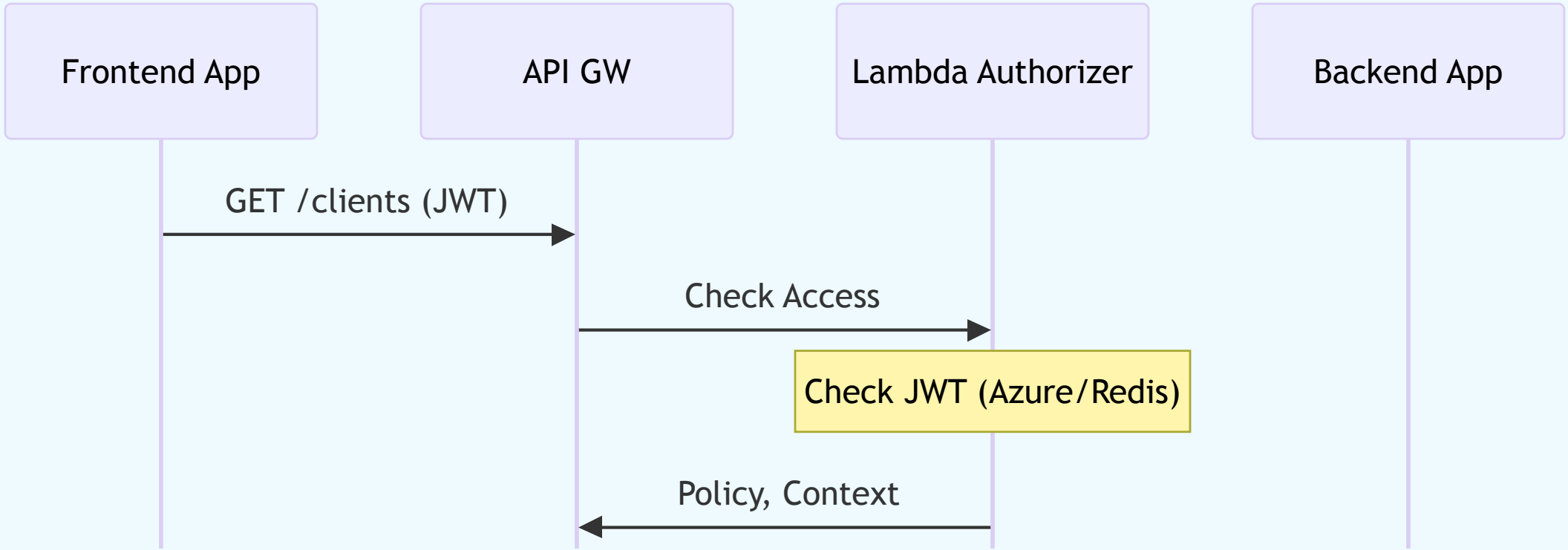
Request flow - Gateway



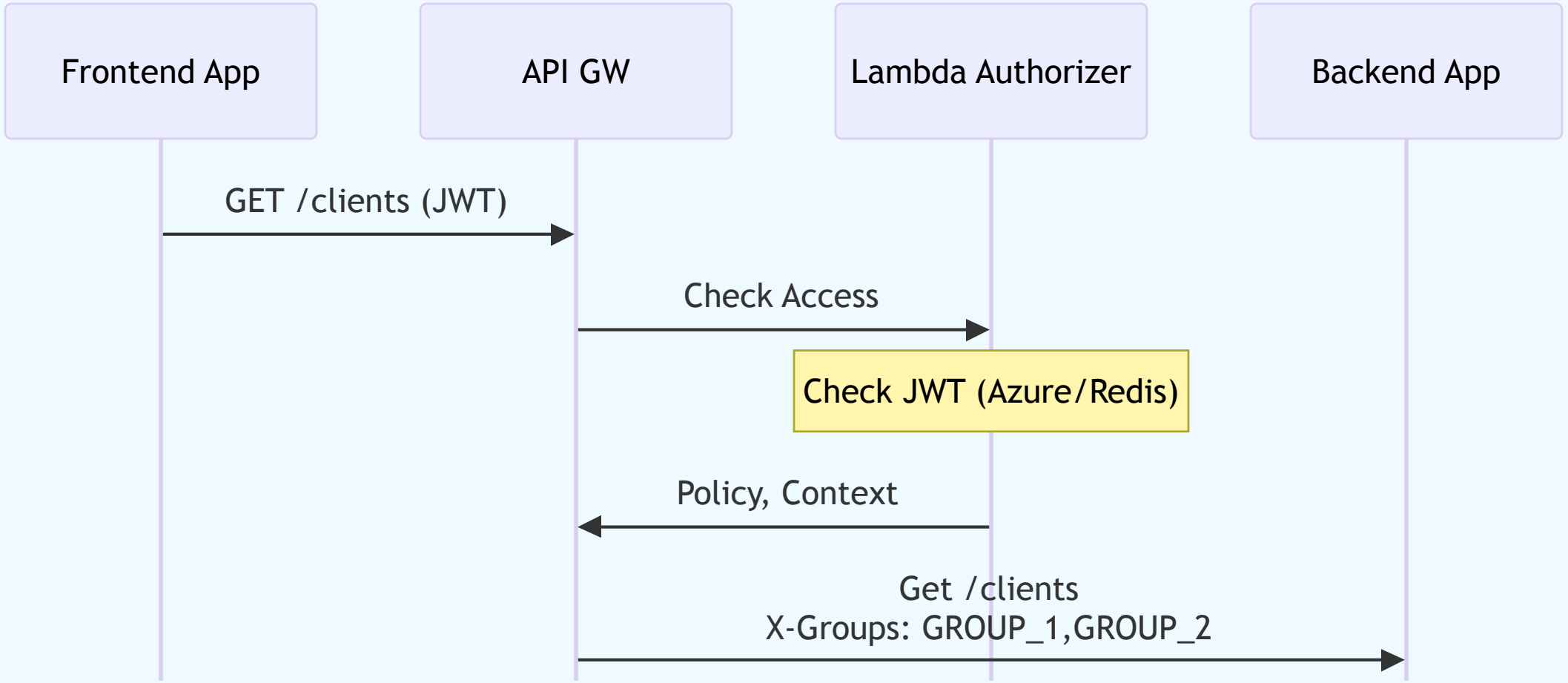
Request flow - Gateway



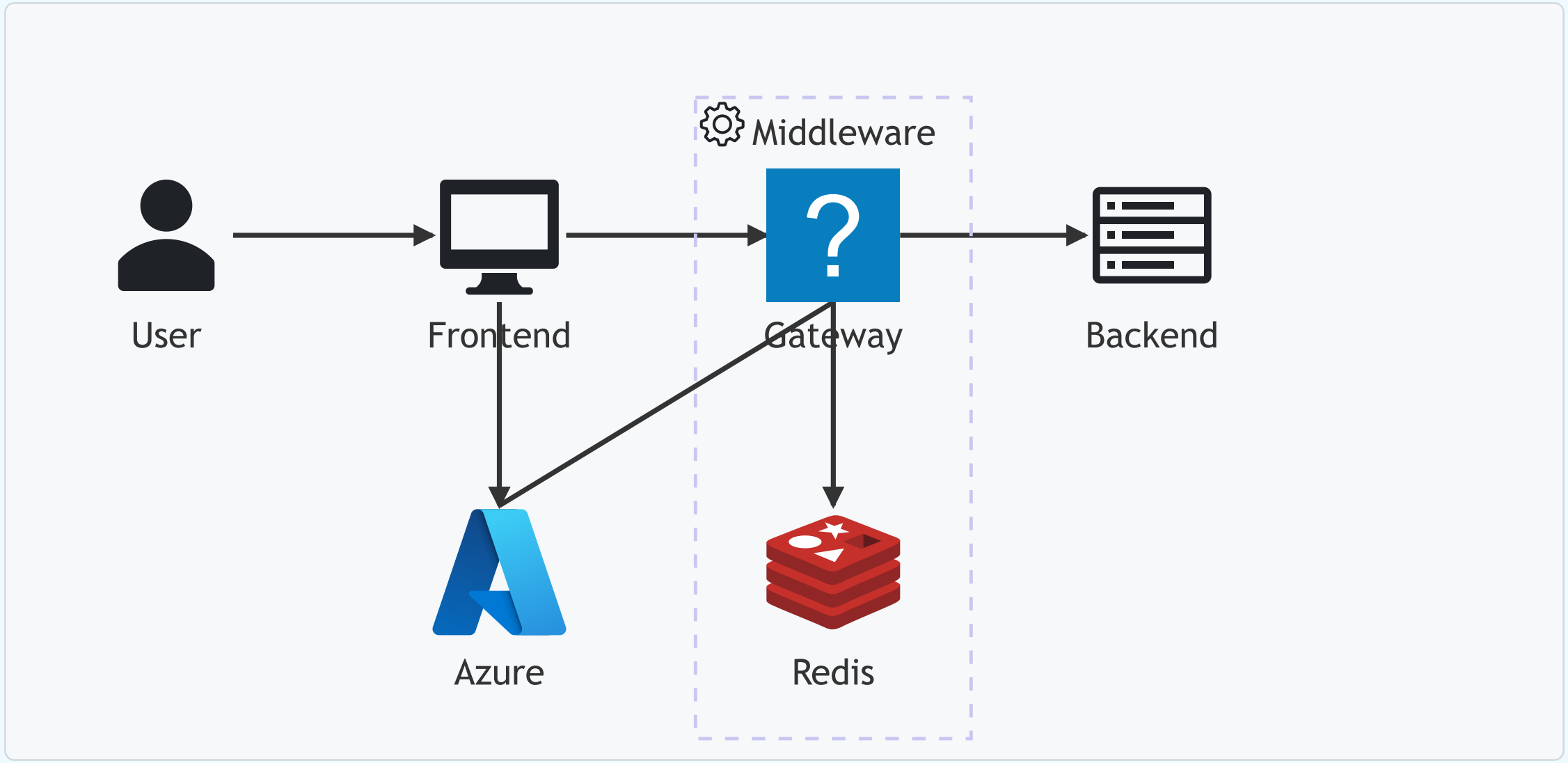
Request flow - Gateway



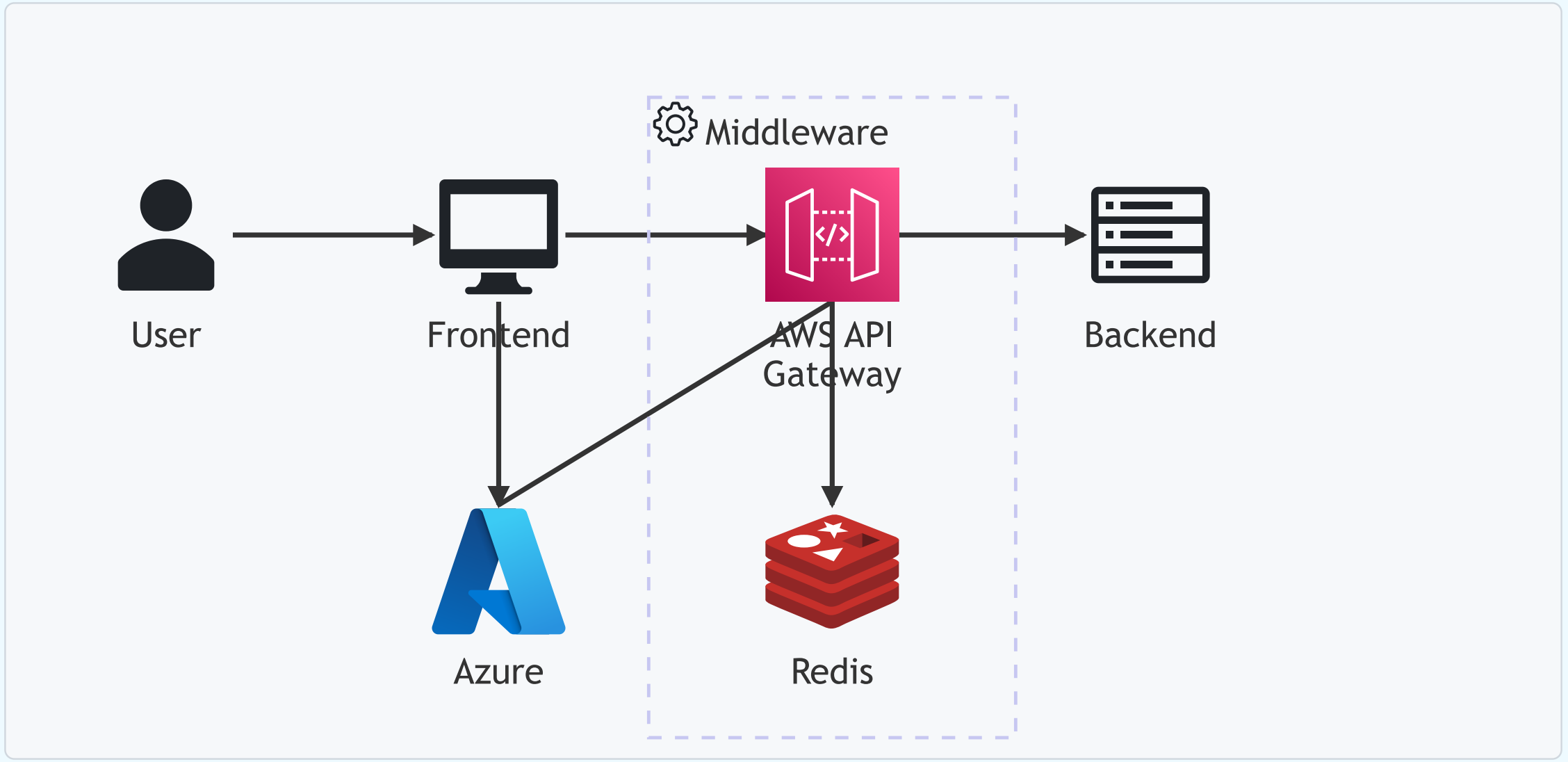
Request flow - Gateway



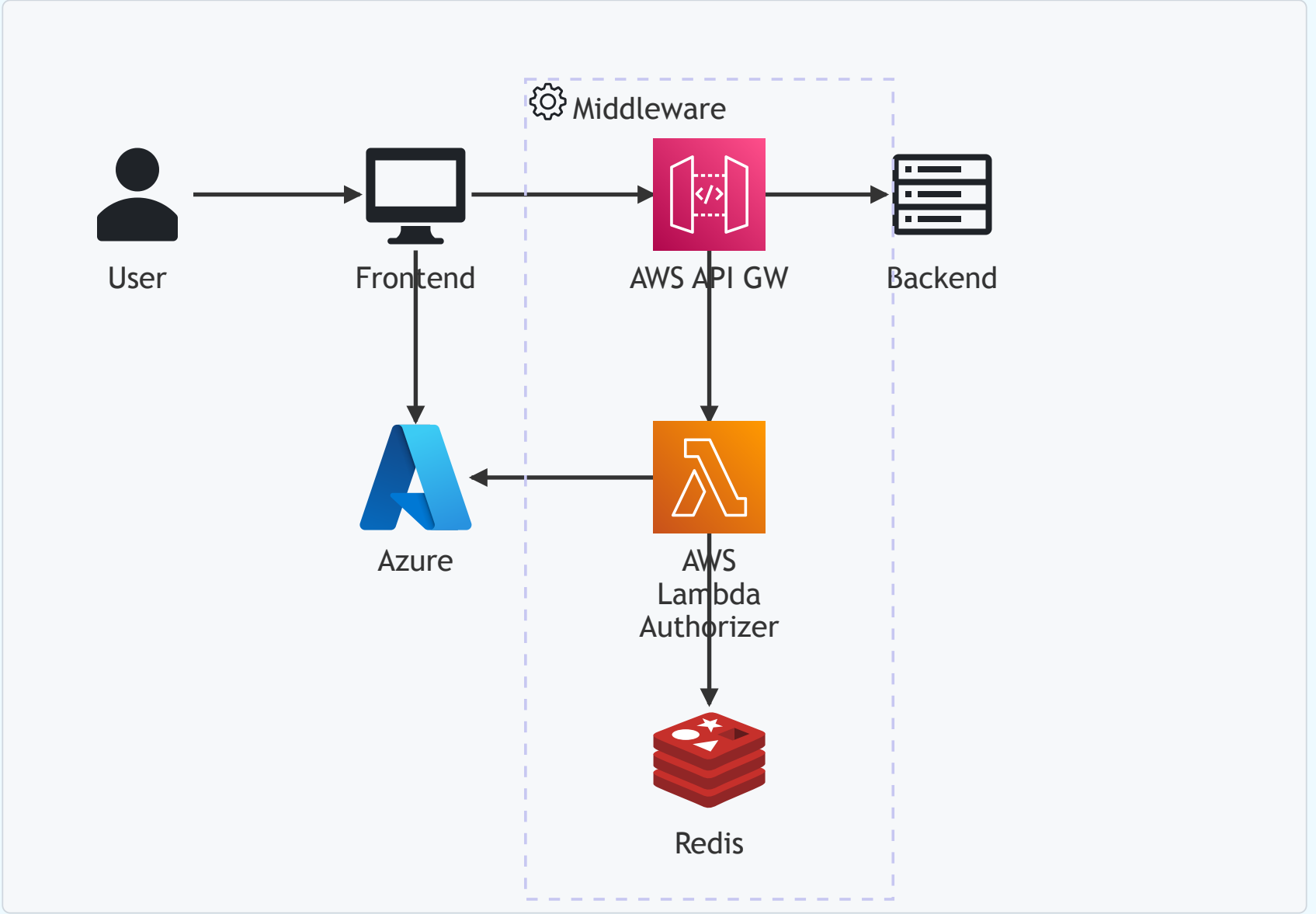
Where we want to be



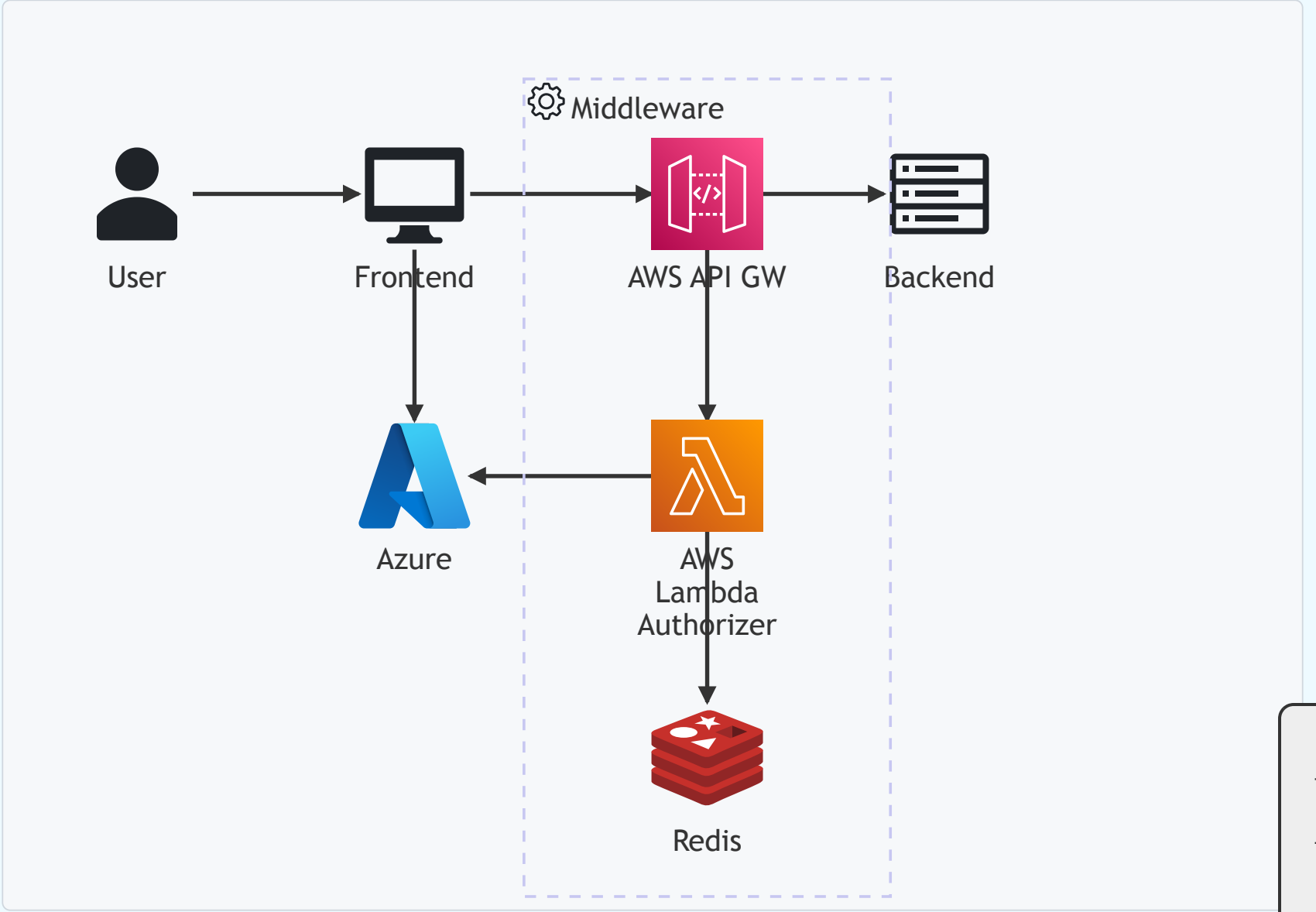
Where we want to be + gateway



Where we want to be + gateway



Where we want to be + gateway



Improve

Gateway
Middle Earth
Ware
North
Forochel
Forodwaith
Grey Mountains
Angmar
Arnor
Iron Hills
Rhovanion
Rhûn
Eriador
Rohan
Enedwaith
Gondor
Mordor
Ash Mountains
Bay of Belfalas
Harondor
Khând
Near Harad
Haradwaith


Gateway
JWT
Middle Earth
Ware
North
Forochel
Forodwaith
Grey Mountains
Angmar of old
Arnor
Eriador
Rhovanion
Rhûn
Iron Hills
Mirkwood
Forlindon
Lune
Gulf of Lune
Harlindon
Minhiriath
Eregion
Fangorn
the Brown Lands
Sea of Rhûn
Ash Mountains
Gorgoroth
Mordor
Nurn
Sea of Nurnen
Mountains of Shadow
Khând
Near Harad
Harondor
Bay of Belfalas
Lebennin
Anfalas
Pinnath Gelin
Havens of Umbar
Haradwaith

JWT

Plain Authentication


```
{
  "aud": "00000003-0000-0000-c000-000000000000",
  "iss": "https://sts.windows.net/********-****-****-****-************/",
  "exp": 1727256721,
  "appid": "********-****-****-****-************",
  "family_name": "Karailanidis",
  "given_name": "Filippos",
  "scp": "profile User.Read email",
  ...
}
```

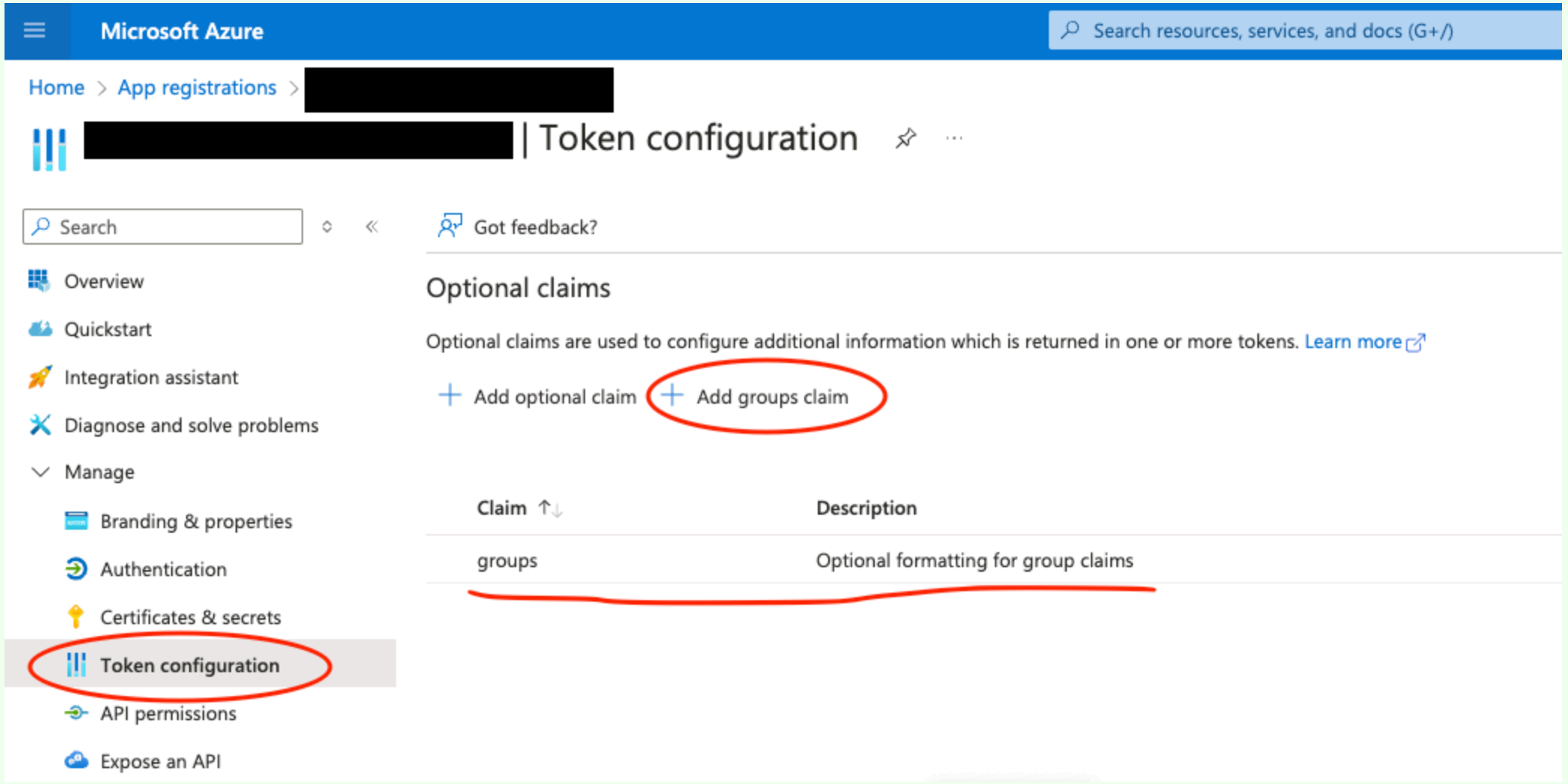
JWT

- Do we need to call Azure to get user groups?
- NO!
- With the magic of: **Claims**

JWT Claims


```
{
  "aud": "00000003-0000-0000-c000-000000000000",
  "iss": "https://sts.windows.net/********-****-****-****-************/",
  "exp": 1727256721,
  "appid": "********-****-****-****-************",
  "family_name": "Karailanidis",
  "given_name": "Filippos",
  "scp": "profile User.Read email",
  ...
  "claim1": "foo",
  "claim2": "bar",
  "claim3": "baz",
  ...
}
```

JWT Claims


Microsoft Azure
Search resources, services, and docs (G+/)
Home > App registrations > [redacted]
[redacted] | Token configuration
Search
Got feedback?
Overview
Quickstart
Integration assistant
Diagnose and solve problems
Manage
Branding & properties
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API
Optional claims
Optional claims are used to configure additional information which is returned in one or more tokens. Learn more
Add optional claim
Add groups claim
Claim
Description
groups
Optional formatting for group claims

JWT Claims


```
{
  "aud": "00000003-0000-0000-c000-000000000000",
  "iss": "https://sts.windows.net/********-****-****-****-************/",
  "exp": 1727256721,
  "appid": "********-****-****-****-************",
  "family_name": "Karailanidis",
  "given_name": "Filippos",
  "scp": "profile User.Read email",
  ...
  "groups" : ["GROUP_1", "GROUP_2"],
  ...
}
```

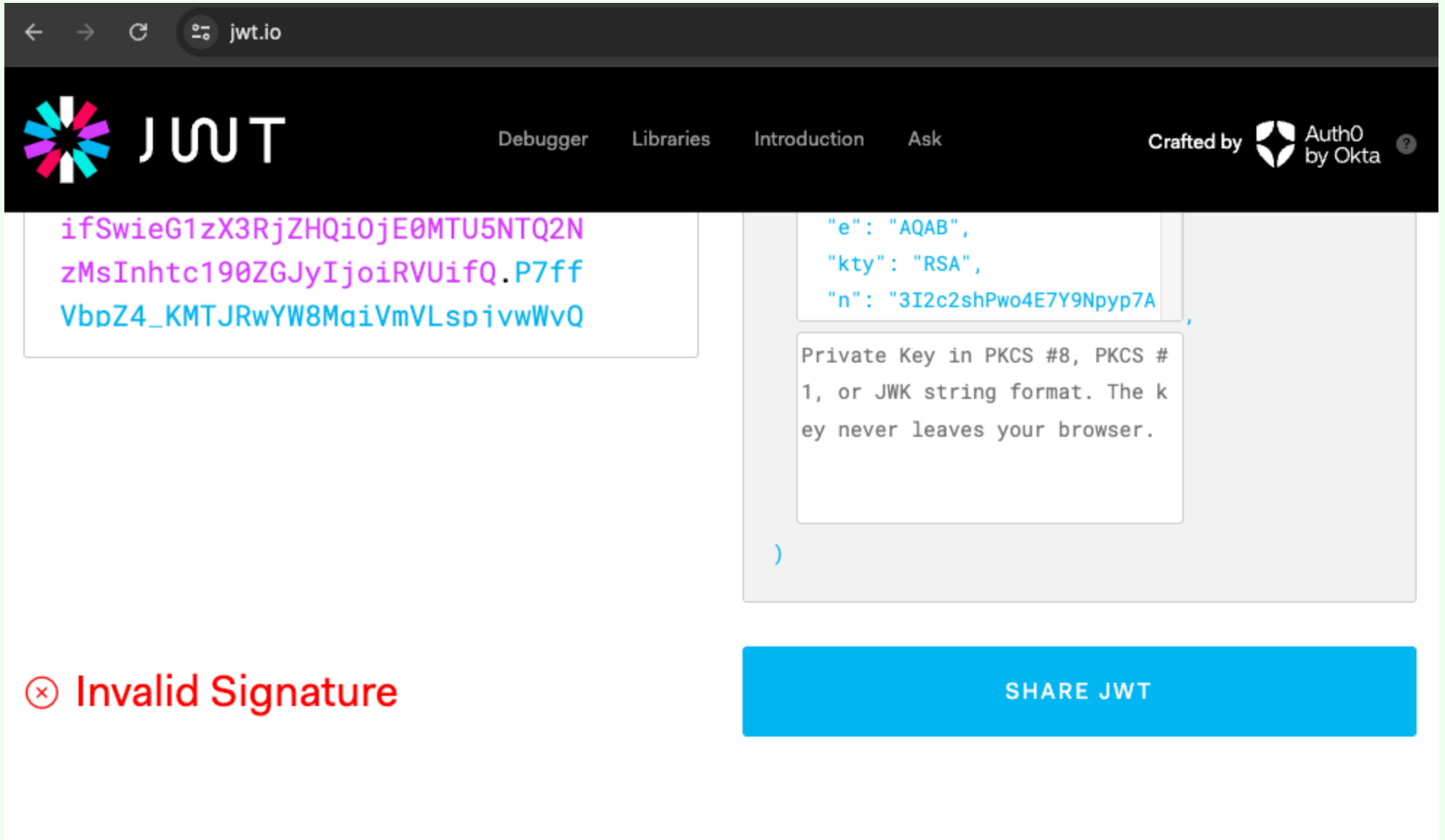
JWT Claims - Win!

- No need to query Graph API + cache groups
- Frontend has information about groups
- JWT signing prevents tampering

JWT Claims

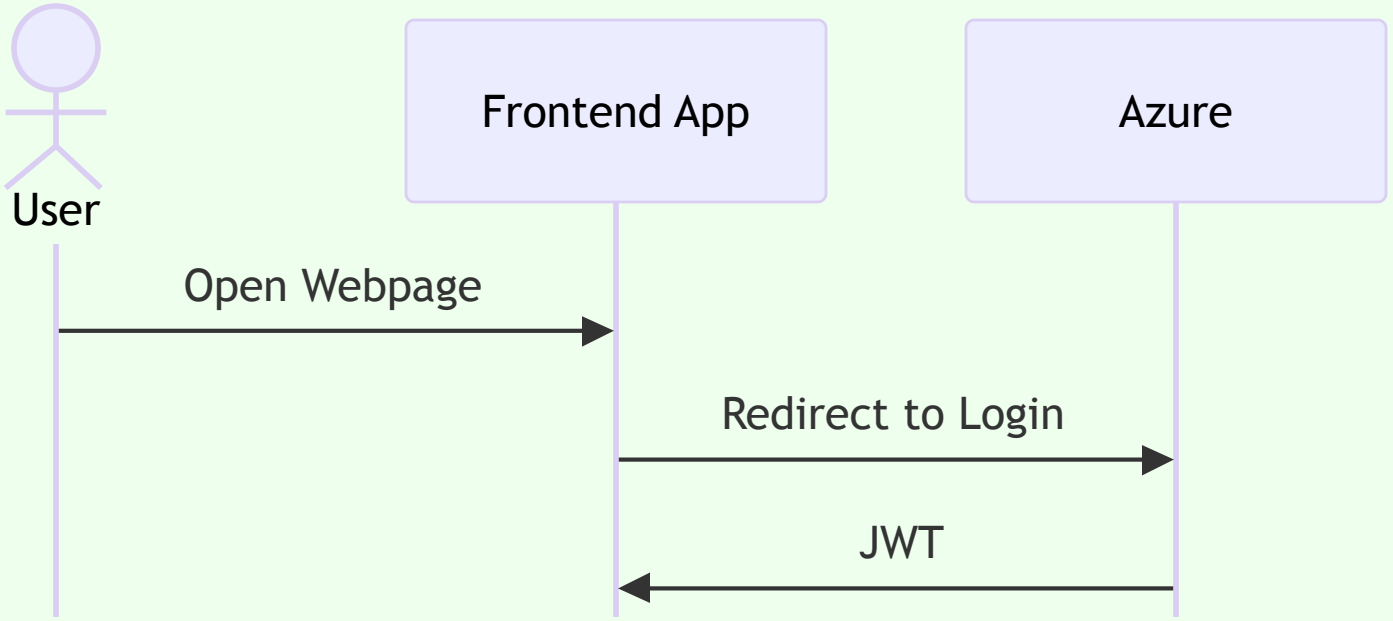
BUT...

JWT Validation 🤦

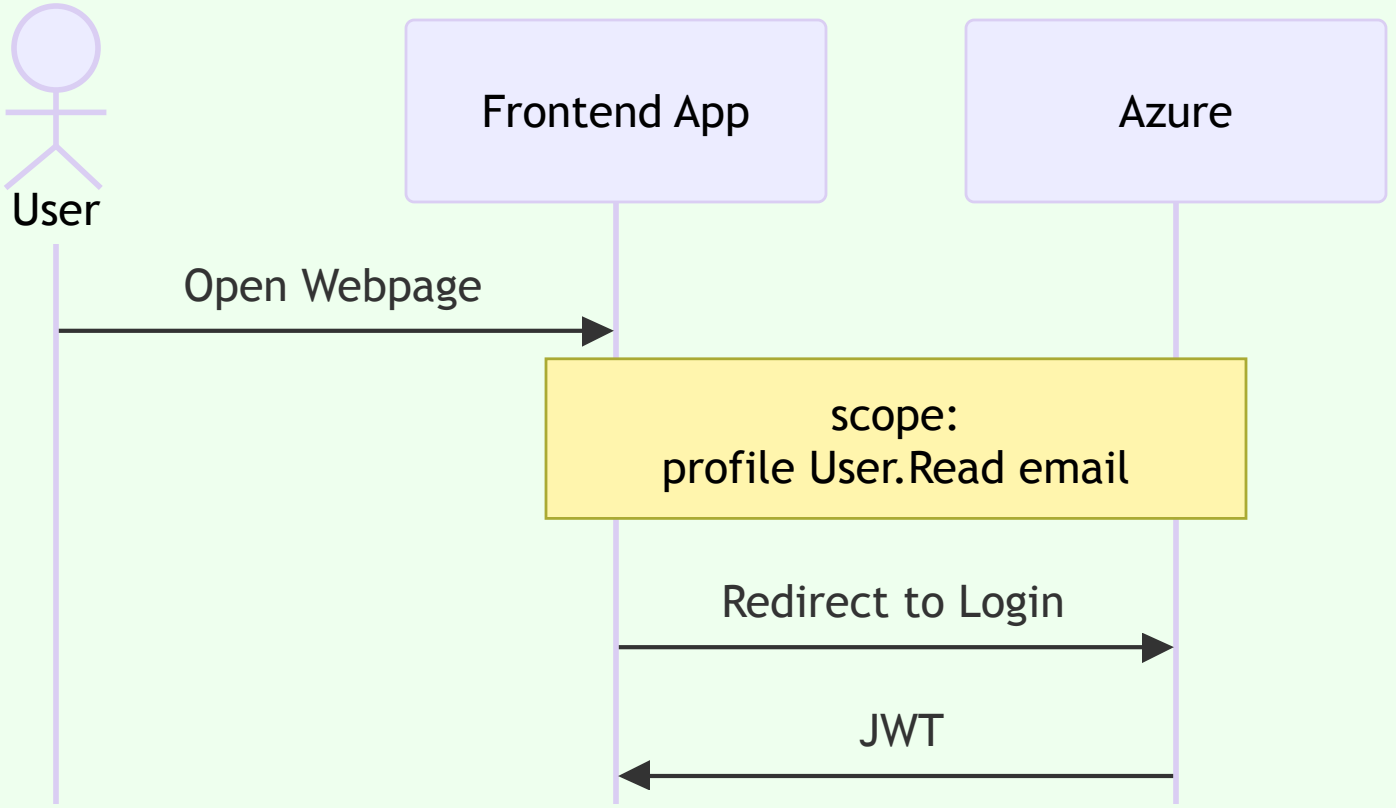


JWT Validation - Step back

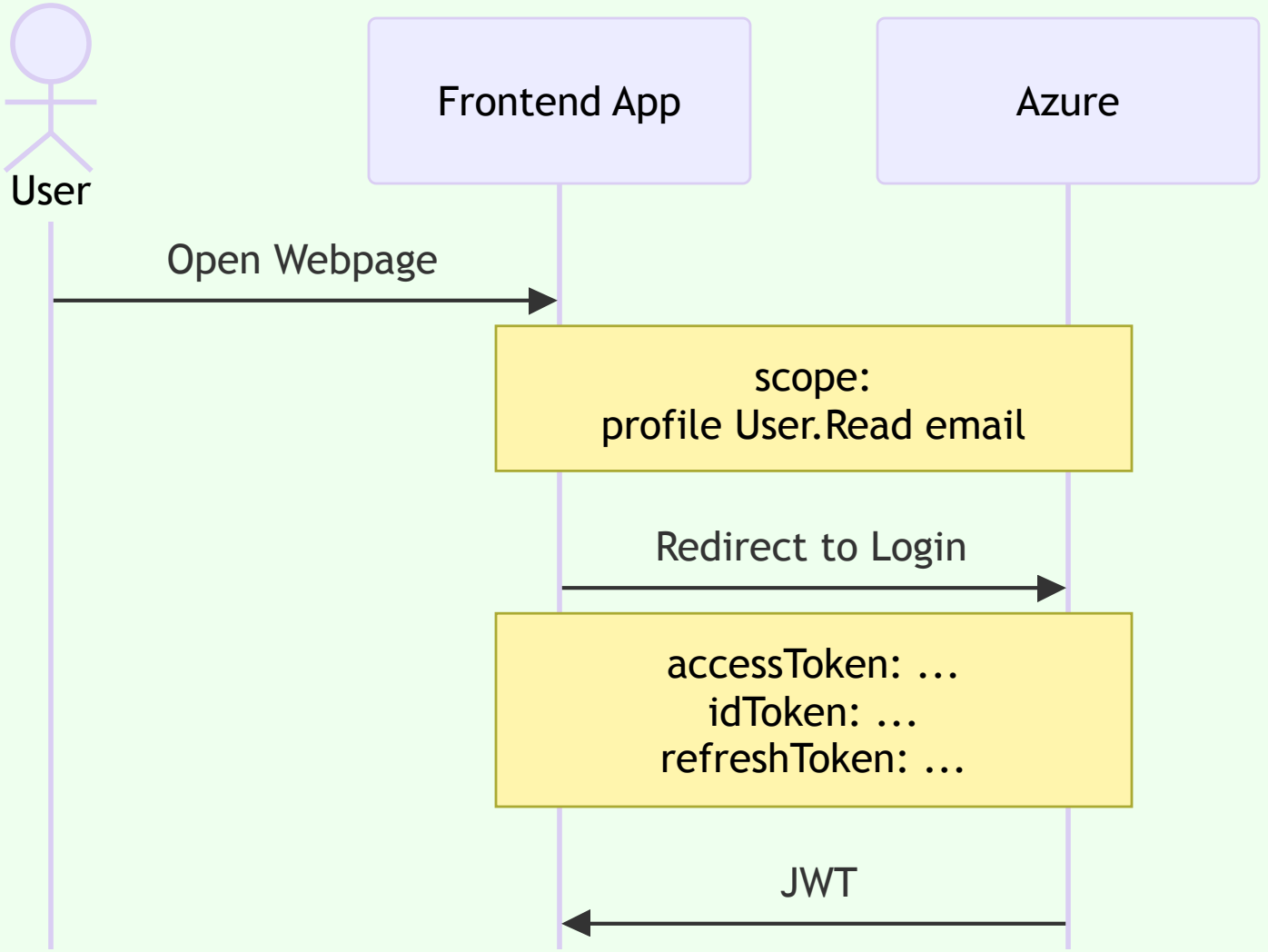
JWT Validation - Step back



JWT Validation - Step back



JWT Validation - Step back



JWT Validation

Tokens:

- Access Token
- Id Token
- Refresh Token

JWT Validation

Refresh Token

- Part of OAuth 2.0
- Can be JWT
- Used to get fresh access tokens

JWT Validation

Access Token

- Part of OAuth 2.0
- Can be JWT
- Used as bearer for API Calls
- Contains claims that are utilized by the Backend

JWT Validation

Access Token

- Part of OAuth 2.0
- Can be JWT
- Used as bearer for API Calls
- Contains claims that are utilized by the Backend

Id Token

- Part of OpenID Connect
- Must be JWT
- Validated and decrypted by the Frontend
- Contains claims that are utilized by the Frontend

JWT Validation

Validate using public key (JWKS) from Microsoft

https://login.microsoftonline.com/common/discovery/keys

JWT Validation

Access Token

- ❌ Validate

Id Token

- ✅ Validate

Access Token vs ID Token

Just use ID Token?

Access Token vs ID Token

Just use ID Token?

Well... you probably shouldn't

Access Token vs ID Token

ID Token is for **client use only**

Access Token vs ID Token

ID Token is for **client use only**

Access Token is meant for Bearer

- It contains scope
- It is short lived
- It can be refreshed

Access Token vs ID Token

But we cannot validate it!

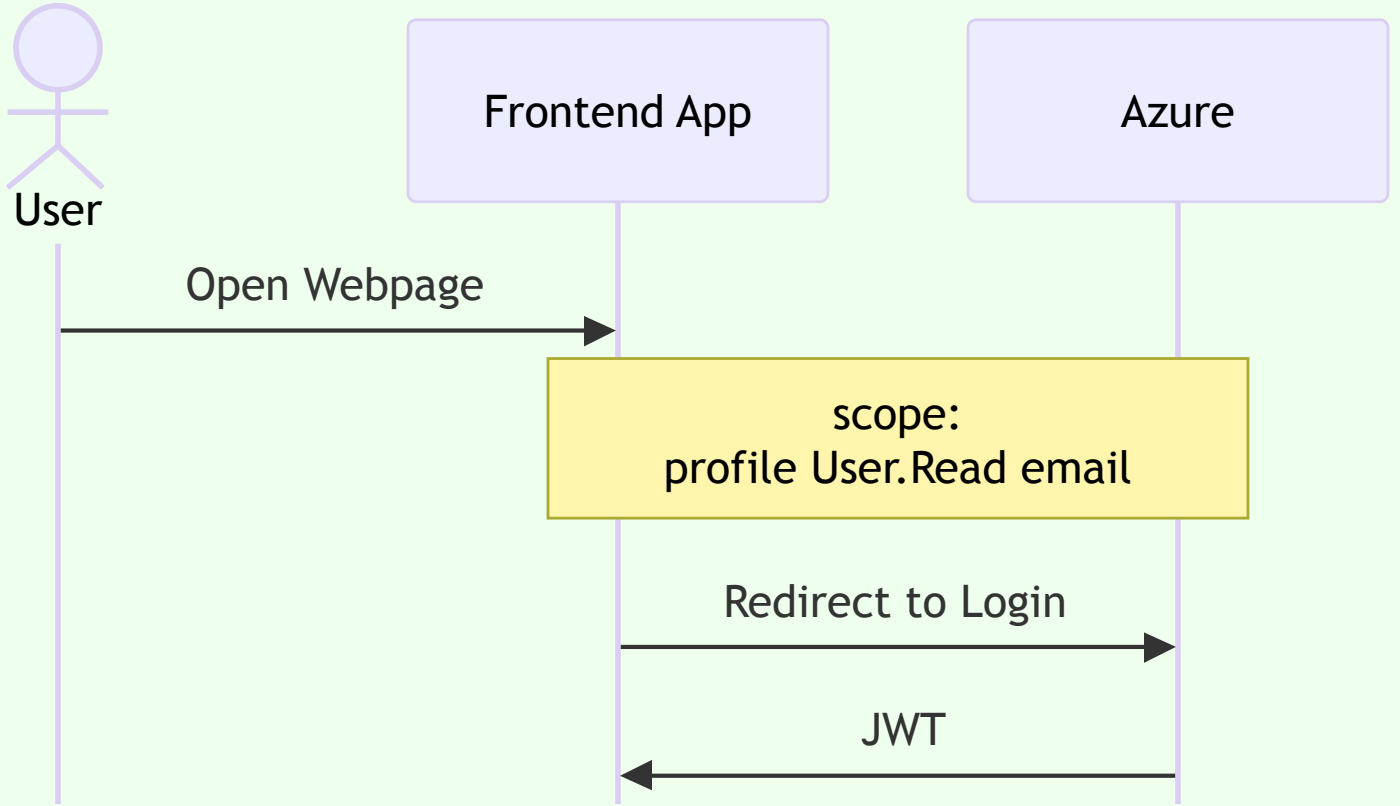
How can we make it work?

Scope

Notice the requested "scope:"

Scope

Notice the requested "scope:"



Scope

Using any Graph scopes, creates an Access Token meant to be parsed by Graph

Scope

Using any Graph scopes, creates an Access Token meant to be parsed by Graph

```
{
  "aud": "00000003-0000-0000-c000-000000000000",
  "iss": "https://sts.windows.net/********-****-****-****-************/",
  "exp": 1727256721,
  "appid": "********-****-****-****-************",
  "family_name": "Karailanidis",
  "given_name": "Filippos",
  "scp": "profile User.Read email",
  "groups" : ["GROUP_1", "GROUP_2"],
  ...
}
```

Scope

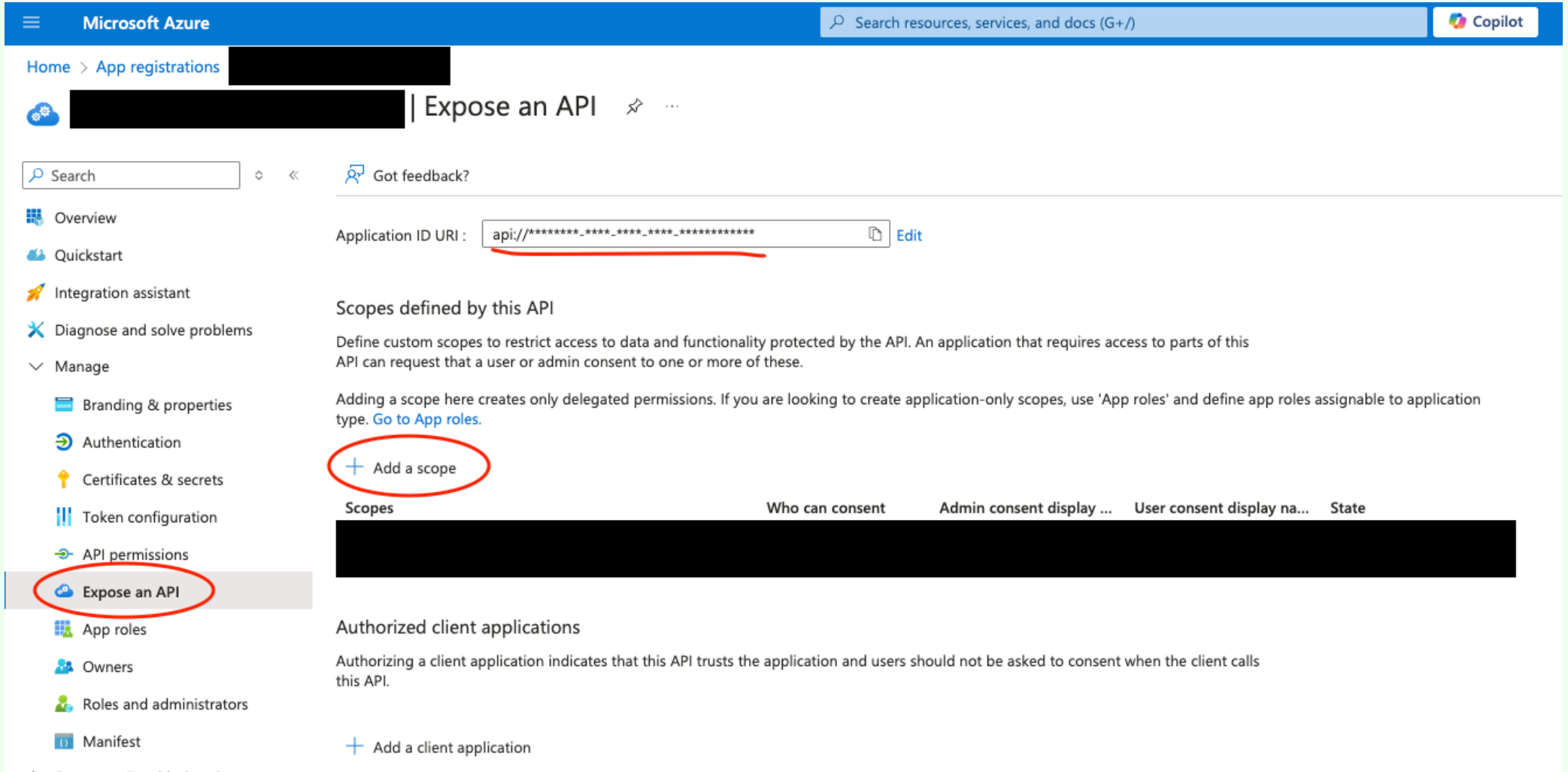
Using any Graph scopes, creates an Access Token meant to be parsed by Graph

```
{
  "aud": "00000003-0000-0000-c000-000000000000",
  "iss": "https://sts.windows.net/********-****-****-****-************/",
  "exp": 1727256721,
  "appid": "********-****-****-****-************",
  "family_name": "Karailanidis",
  "given_name": "Filippos",
  "scp": "profile User.Read email",
  "groups" : ["GROUP_1", "GROUP_2"],
  ...
}
```

We need to tell Azure, we have our own "custom" API

Scope

Home > App Registrations > Expose An API > Add a scope



💡 Scope

In the Frontend request

```
scope: "api://********-****-****-****-************/<endpoint>"
```

💡 Scope

In the Frontend request

```
    scope: "api://********-****-****-****-************/<endpoint>"
```

Access Token

```
{
  "aud": "api://********-****-****-****-************",
  "scp": "<endpoint>",
  ...
}
```

💡 Scope

In the Frontend request

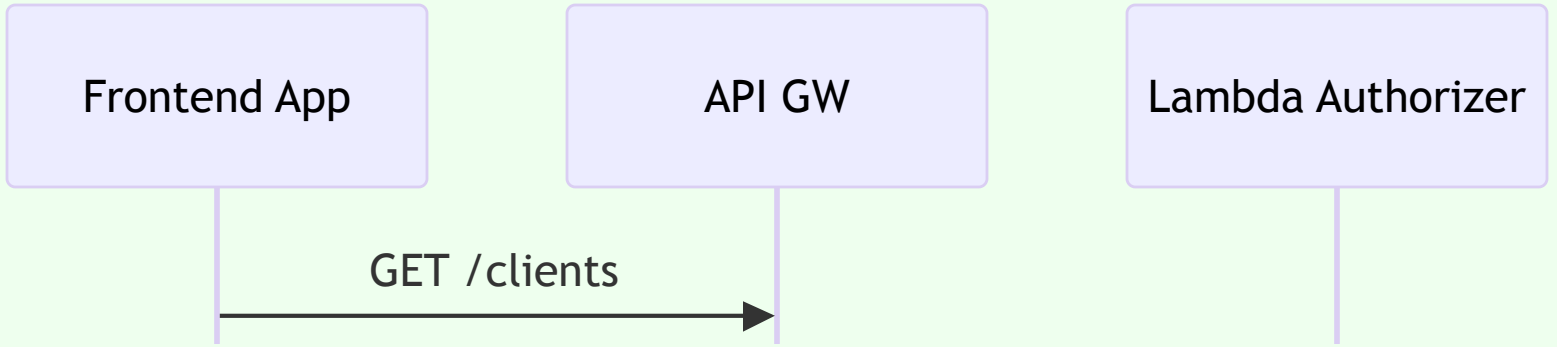
```
    scope: "api://********-****-****-****-************/<endpoint>"
```

Access Token

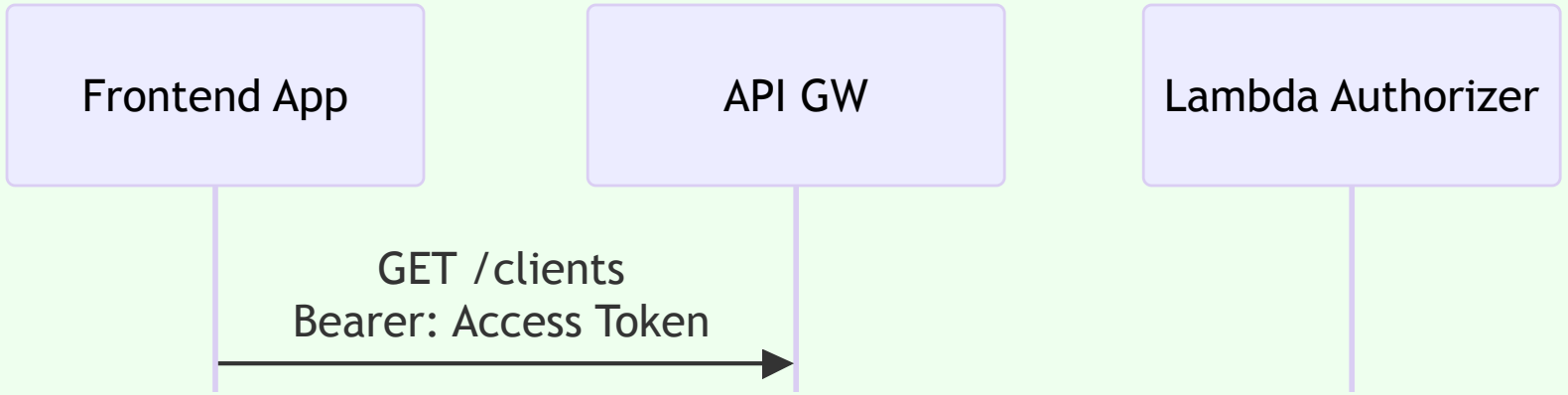
```
{
  "aud": "api://********-****-****-****-************",
  "scp": "<endpoint>",
  ...
}
```

✅ Validate

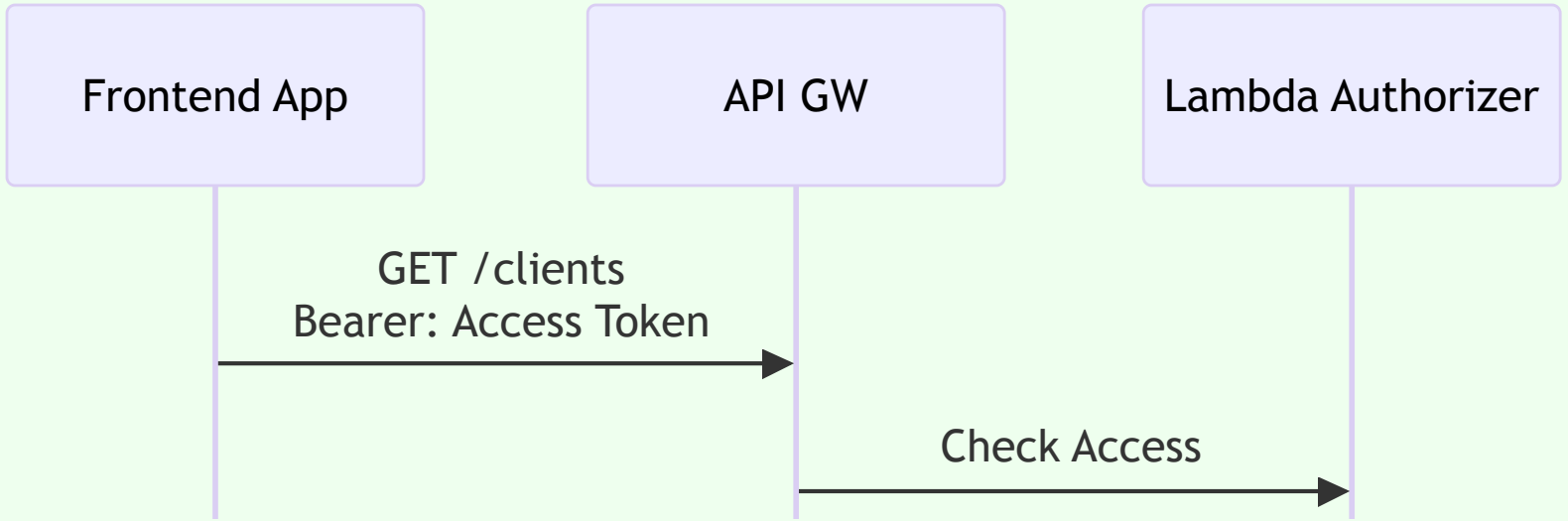
Request flow - Access Token



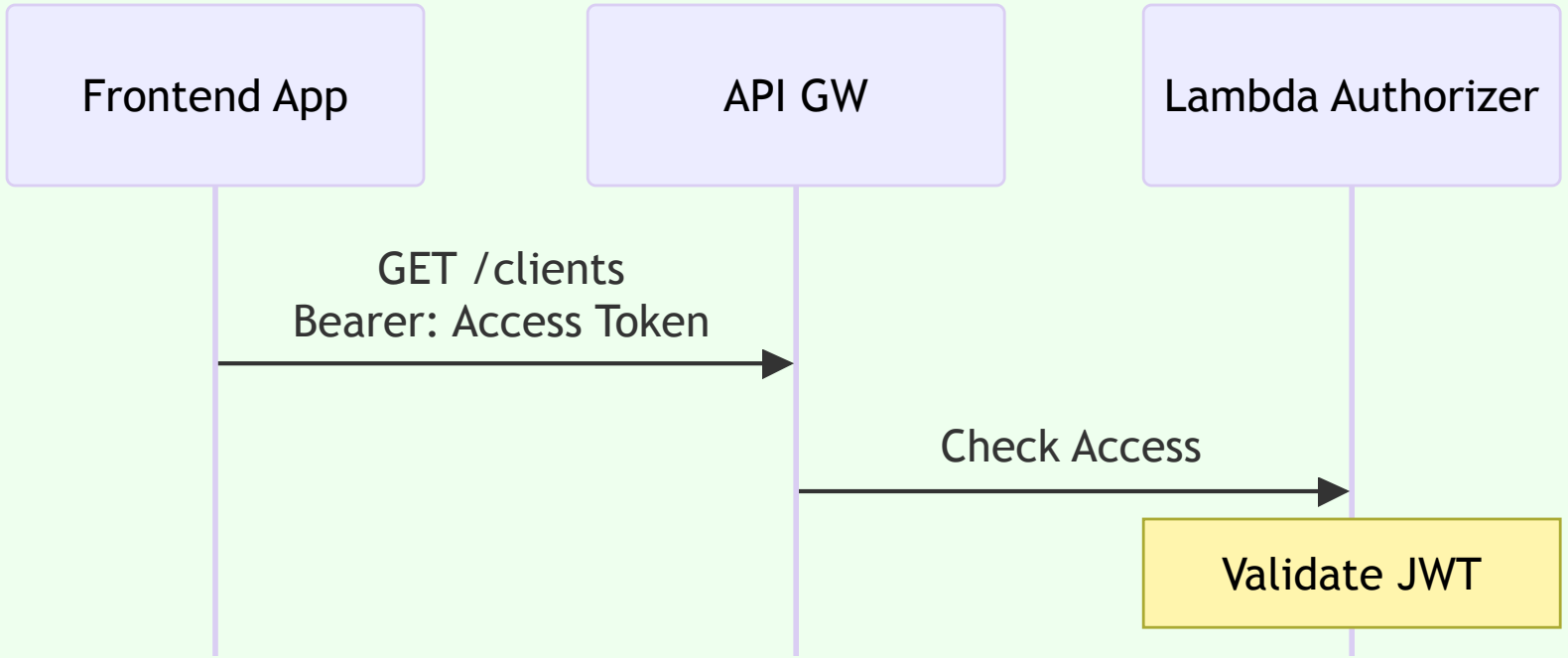
Request flow - Access Token



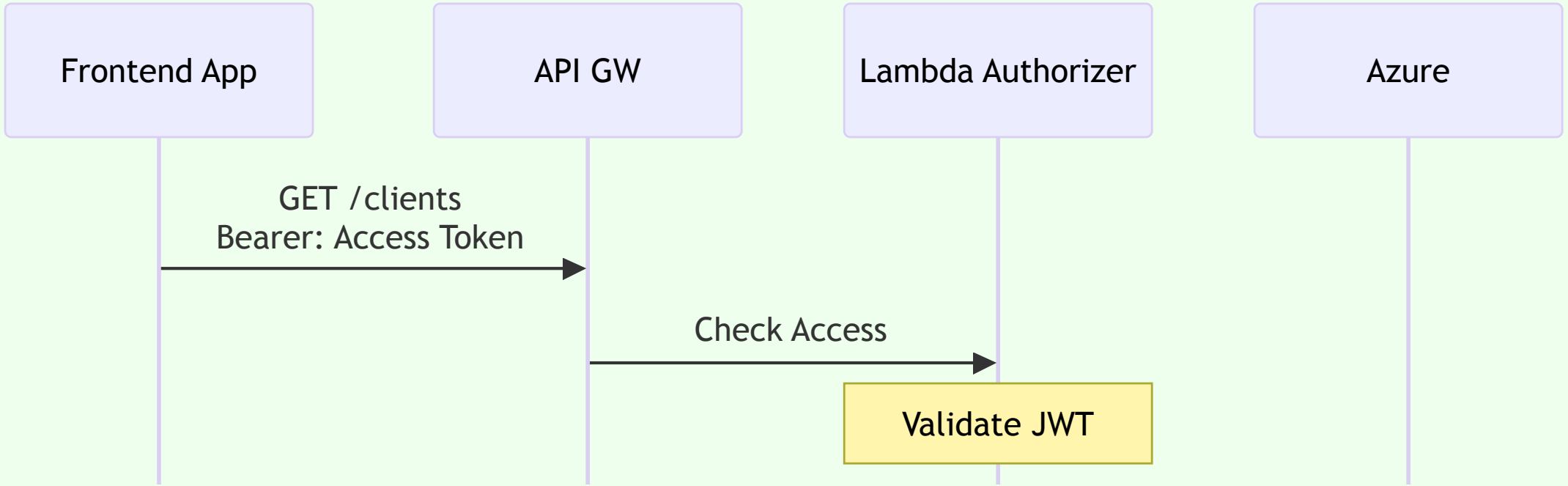
Request flow - Access Token



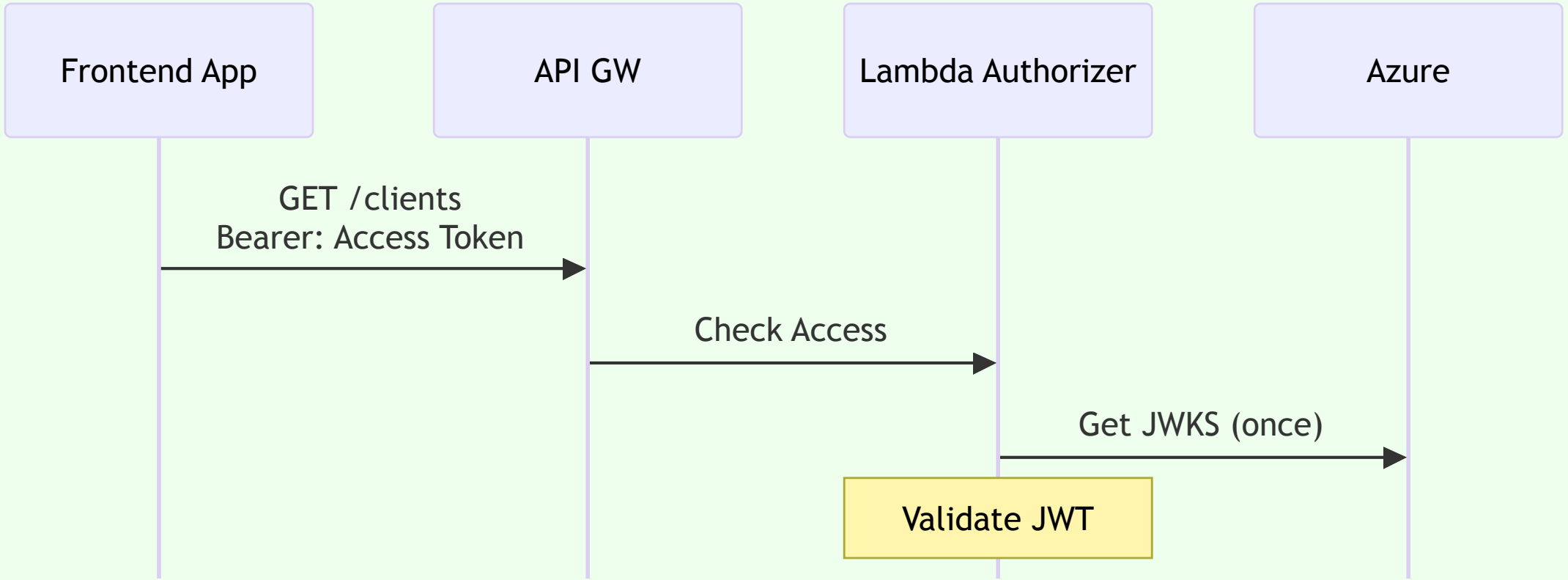
Request flow - Access Token



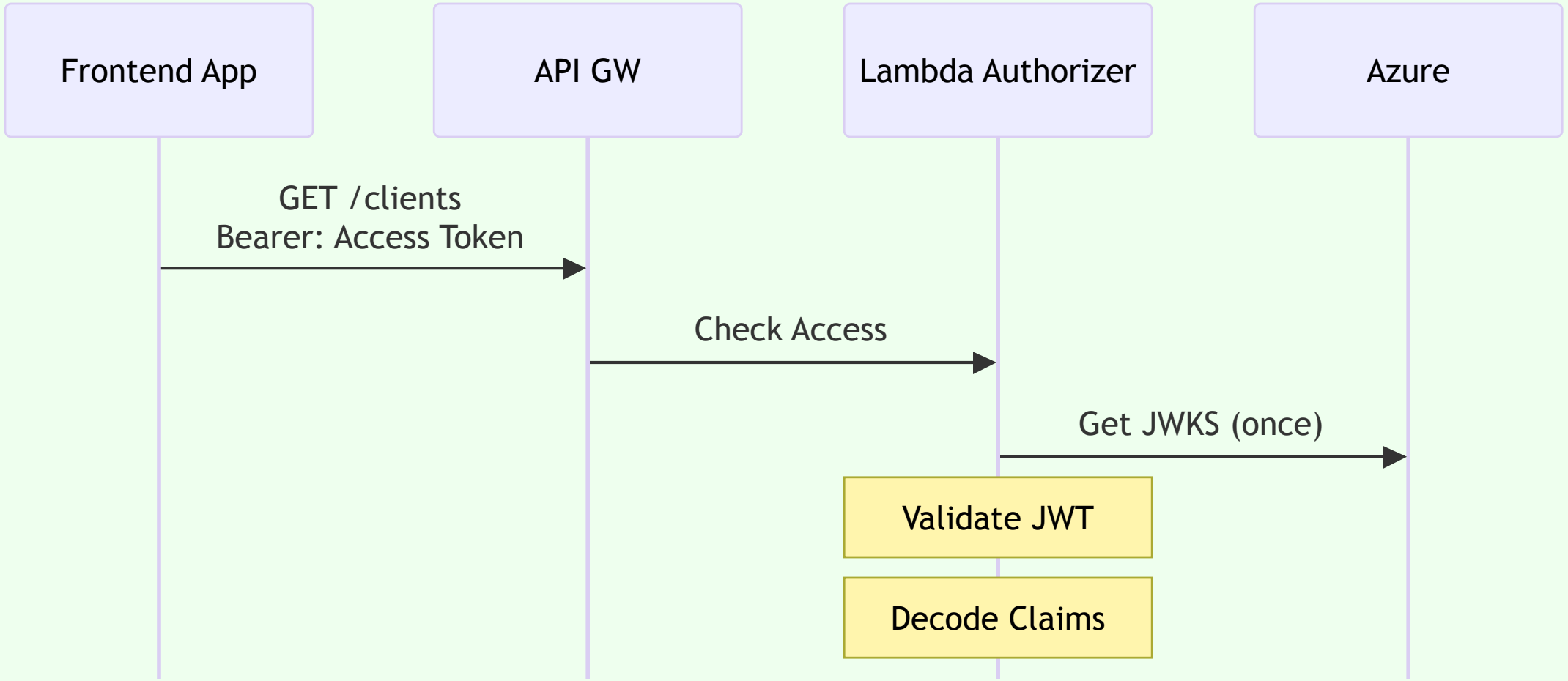
Request flow - Access Token



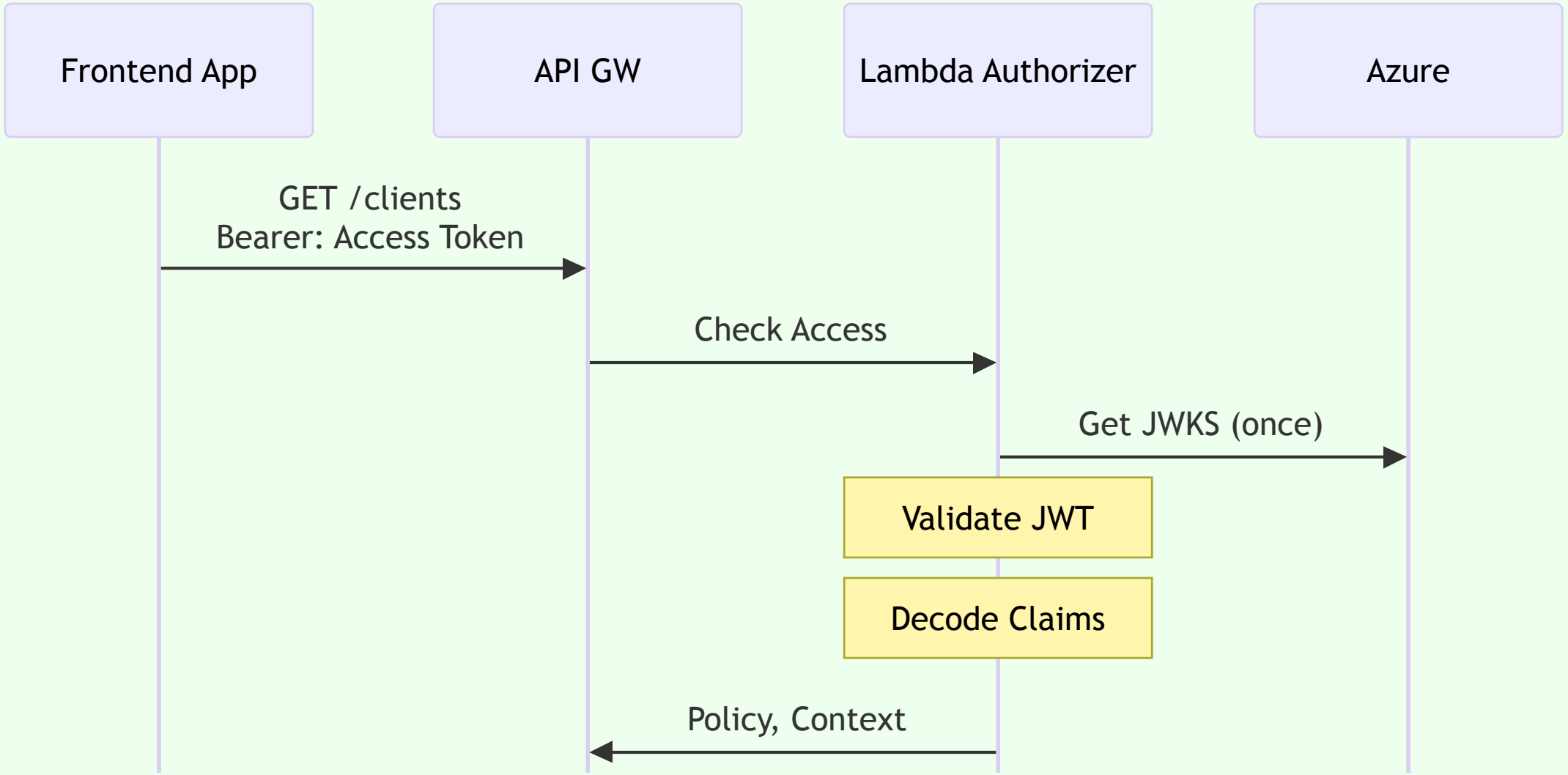
Request flow - Access Token



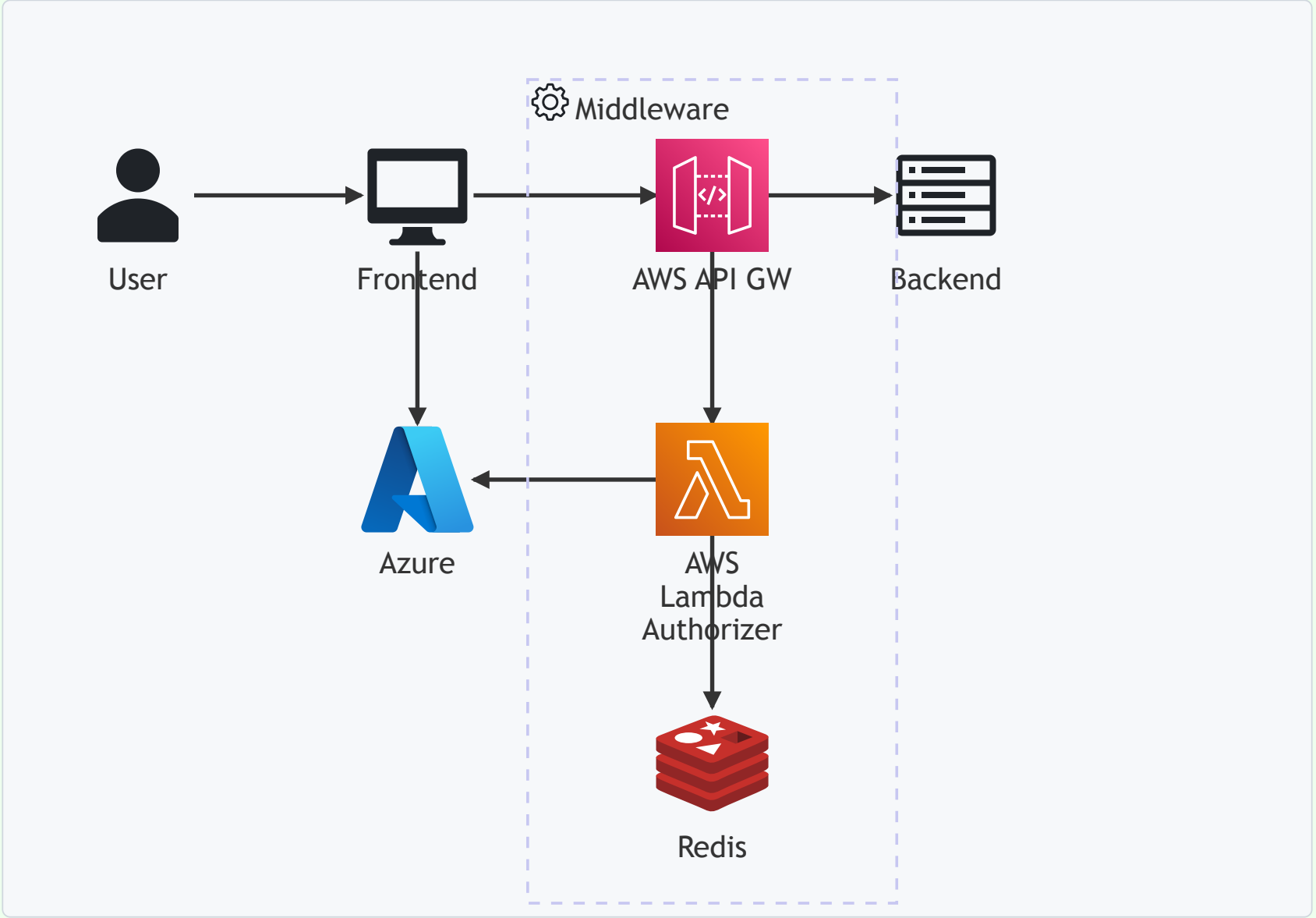
Request flow - Access Token



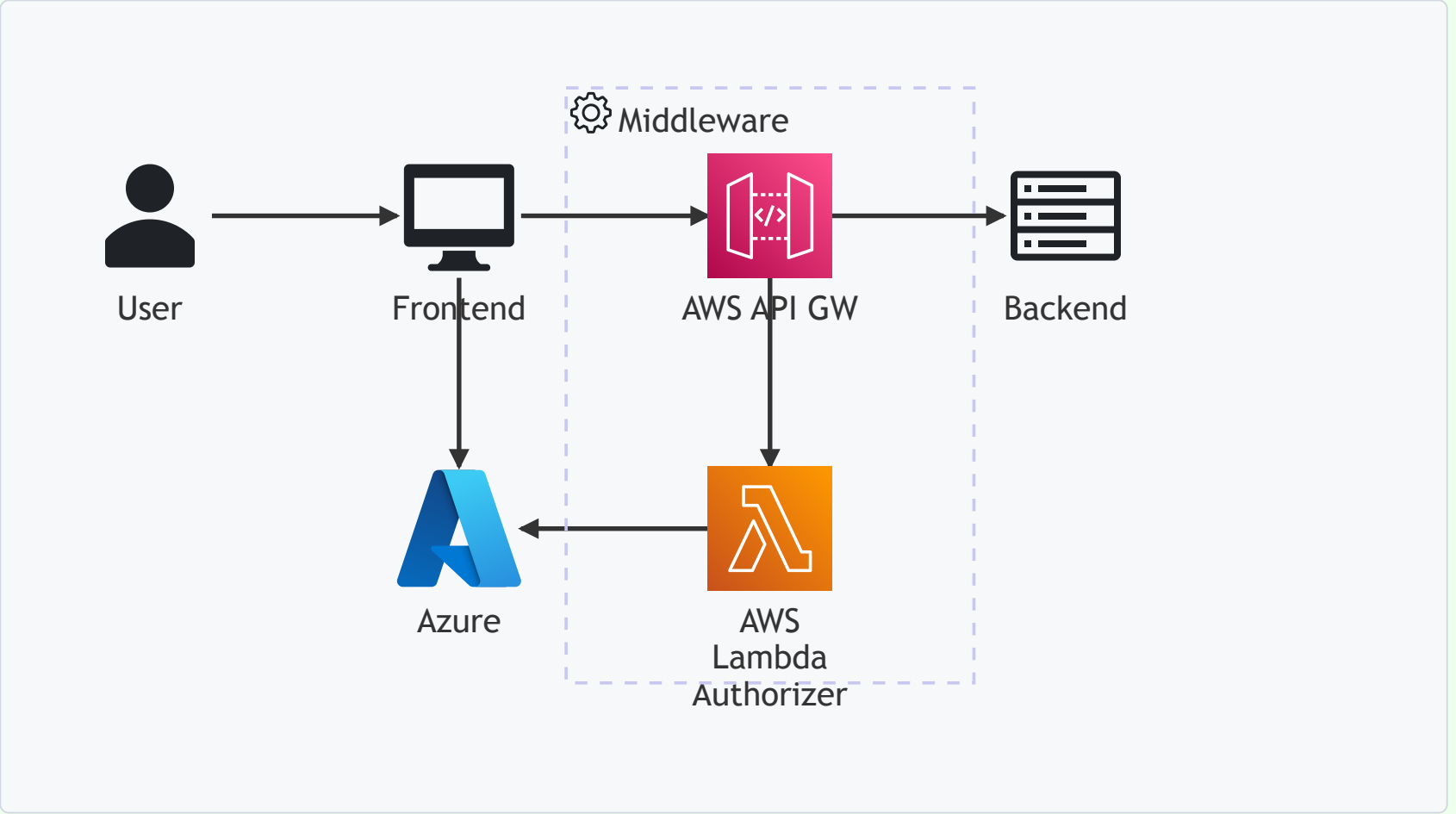
Request flow - Access Token



Where we want to be + gateway



Where we want to be + JWT claims




Gateway
JWT
Middle Ware
Middle Earth
North
Forochel
Forodwaith
Grey Mountains
Angmar of old
Arnor
Eriador
Rhovanion
Rhûn
Iron Hills
Mirkwood
Forlindon
Lune
Gulf of
Harlindon
Minhiriath
Eregion
Fangorn
the Brown Lands
Sea of Rhûn
Ash Mountains
Gorgoroth
Mordor
Nurn
Sea of Nurnen
Mountains of Shadow
Khând
Near Harad
Harondor
Bay of Belfalas
Lebennin
Anfalas
Pinnath Gelin
Havens of Umbar
Haradwaith


Gateway
JWT
Groups
Middle Earth
Ware
North
Forochel
Forodwaith
Grey Mountains
Angmar of old
Eriador
Arnor
Rhovanion
Iron Hills
Forlindon
Blue Mountains
Lune
Gulf of Lune
Harlindon
Eregion
Minhiriath
Fangorn
the Brown Lands
Sea of Rhûn
Ash Mountains
Mordor
Mount Doom
Bay of Belfalas
Lebennin
Harondor
Haradwaith

Groups

End goal?

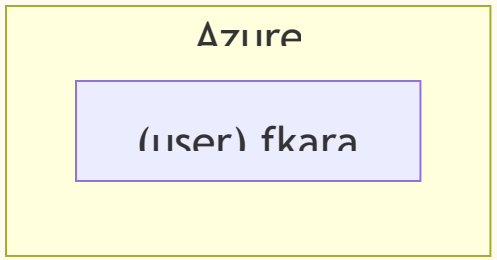
Input

`fkarailanidis`

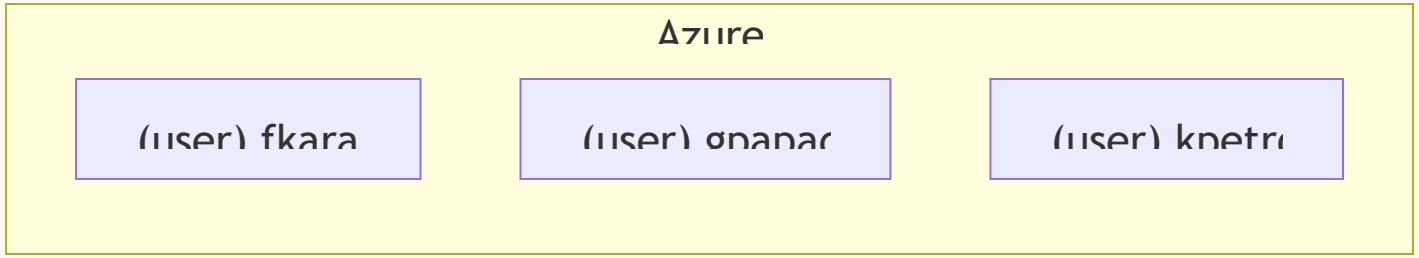
Output

`CRM_VIEW_CLIENT`, `CRM_EDIT_CLIENT`

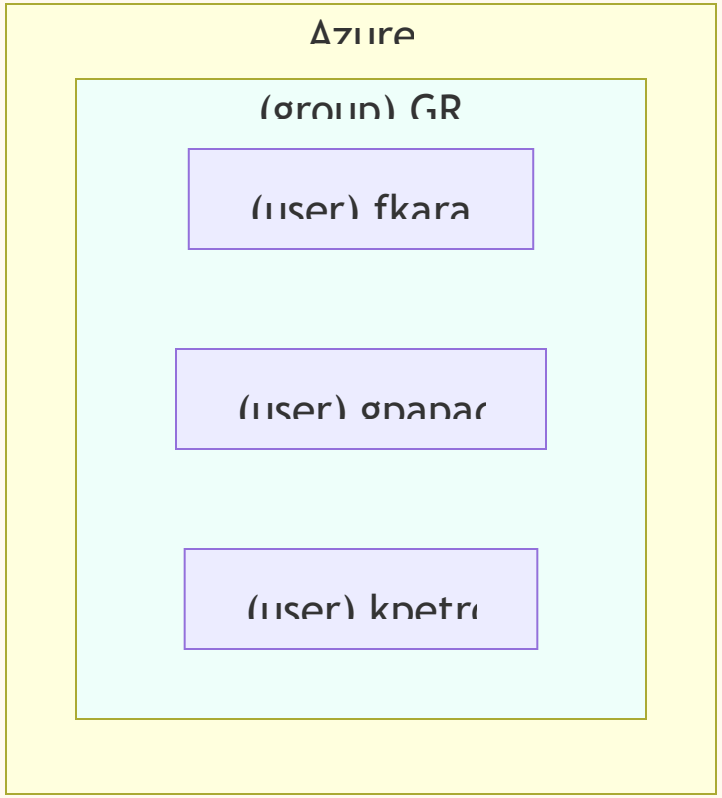
Our Structure



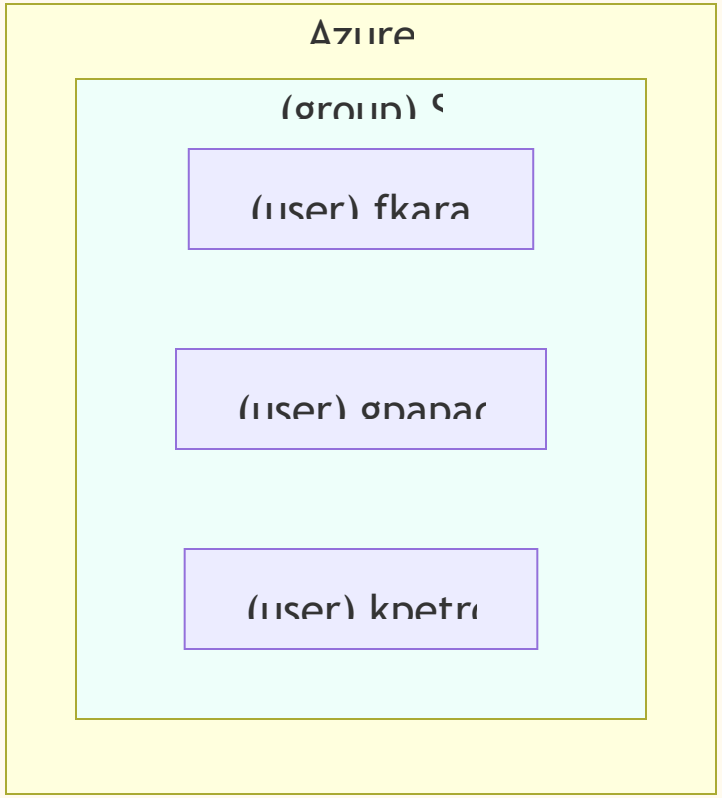
Our Structure



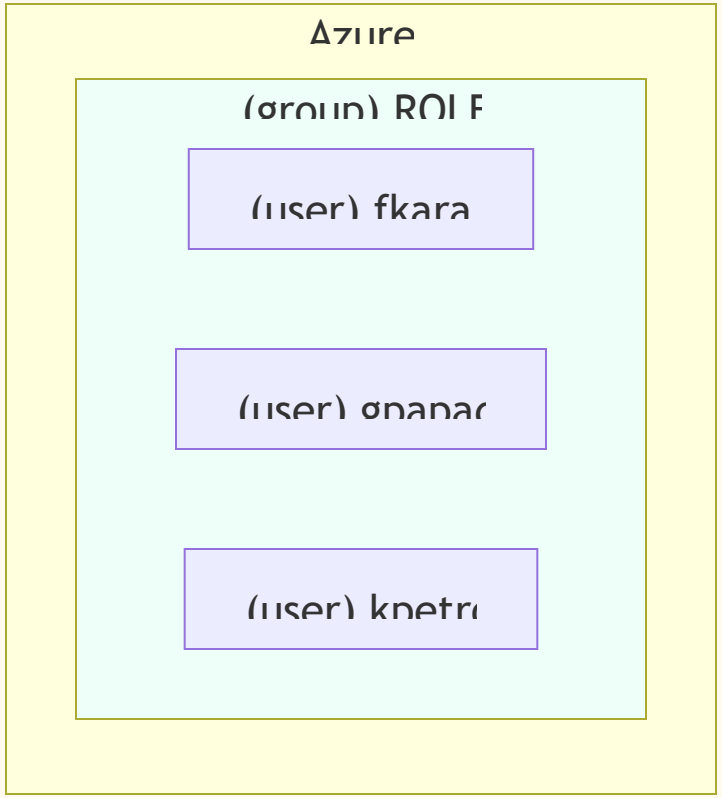
Our Structure



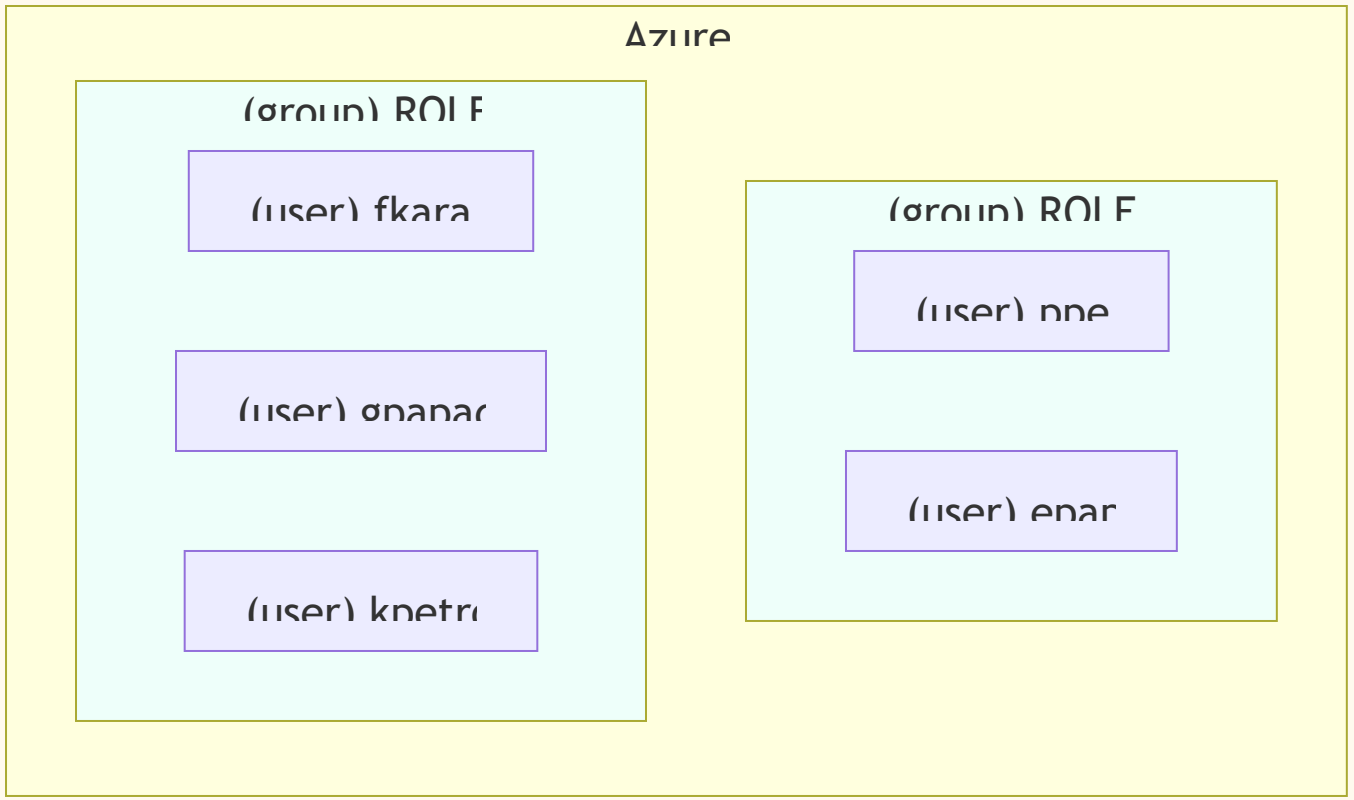
Our Structure



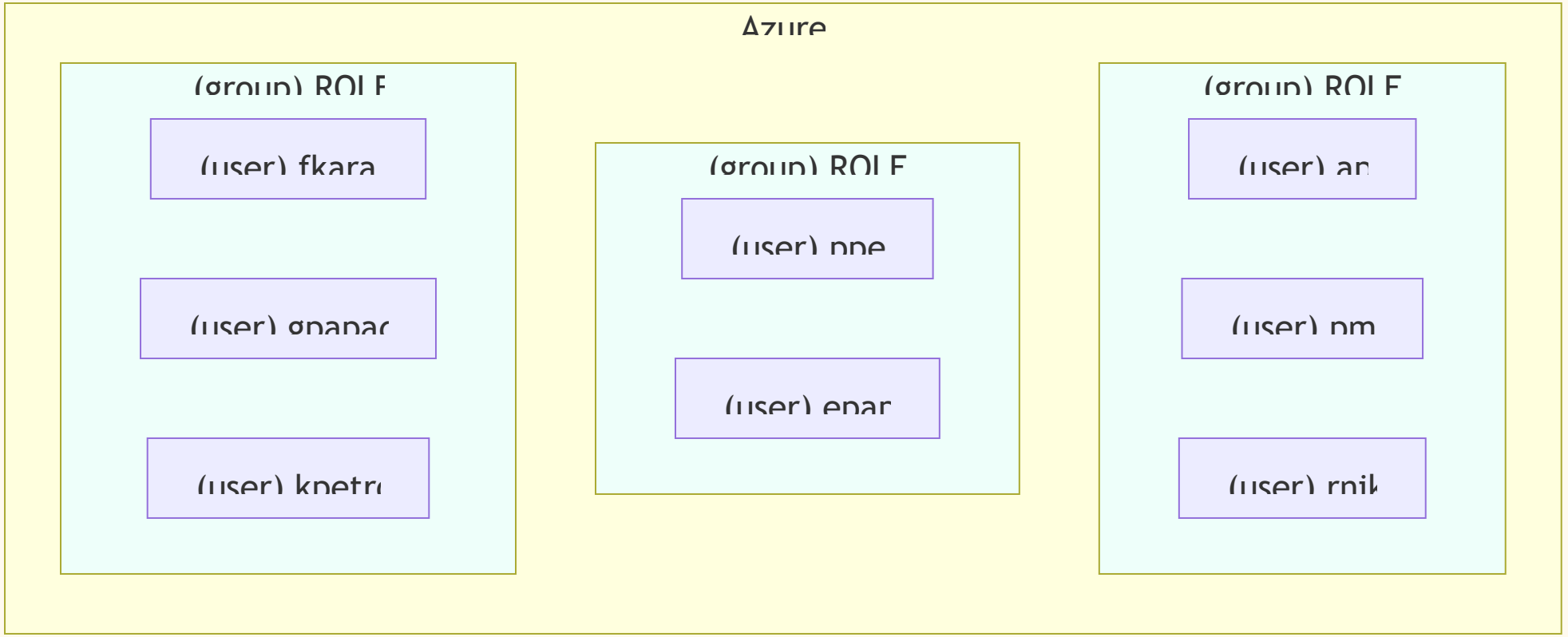
Our Structure



Our Structure

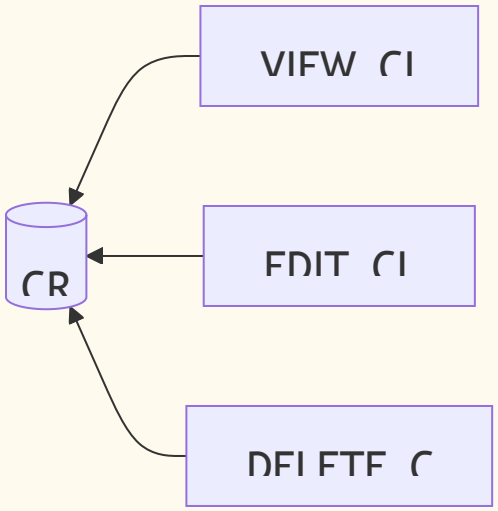


Our Structure

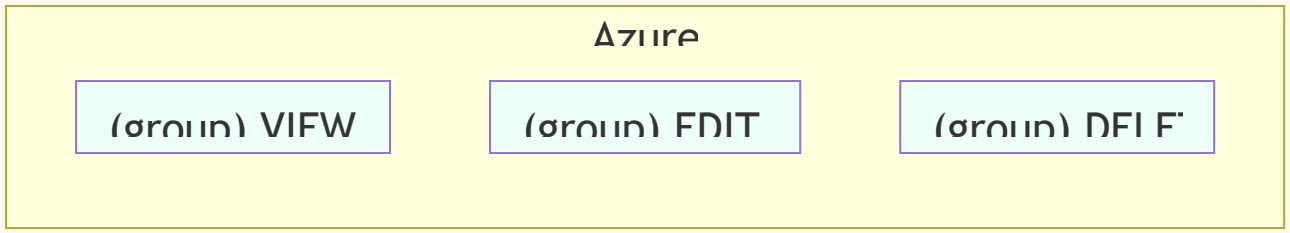


System Permissions

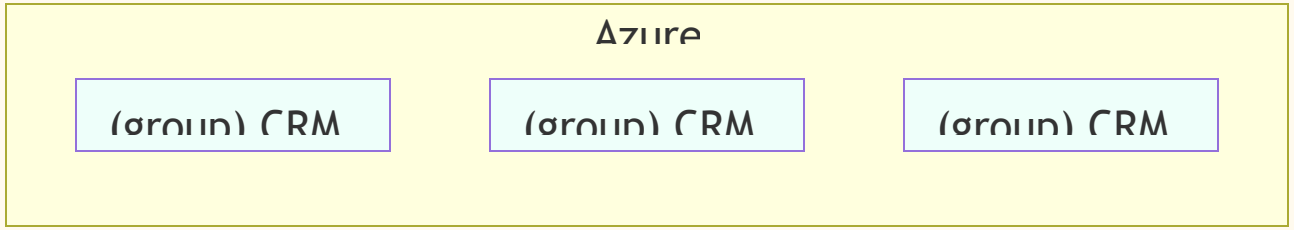
System Permissions



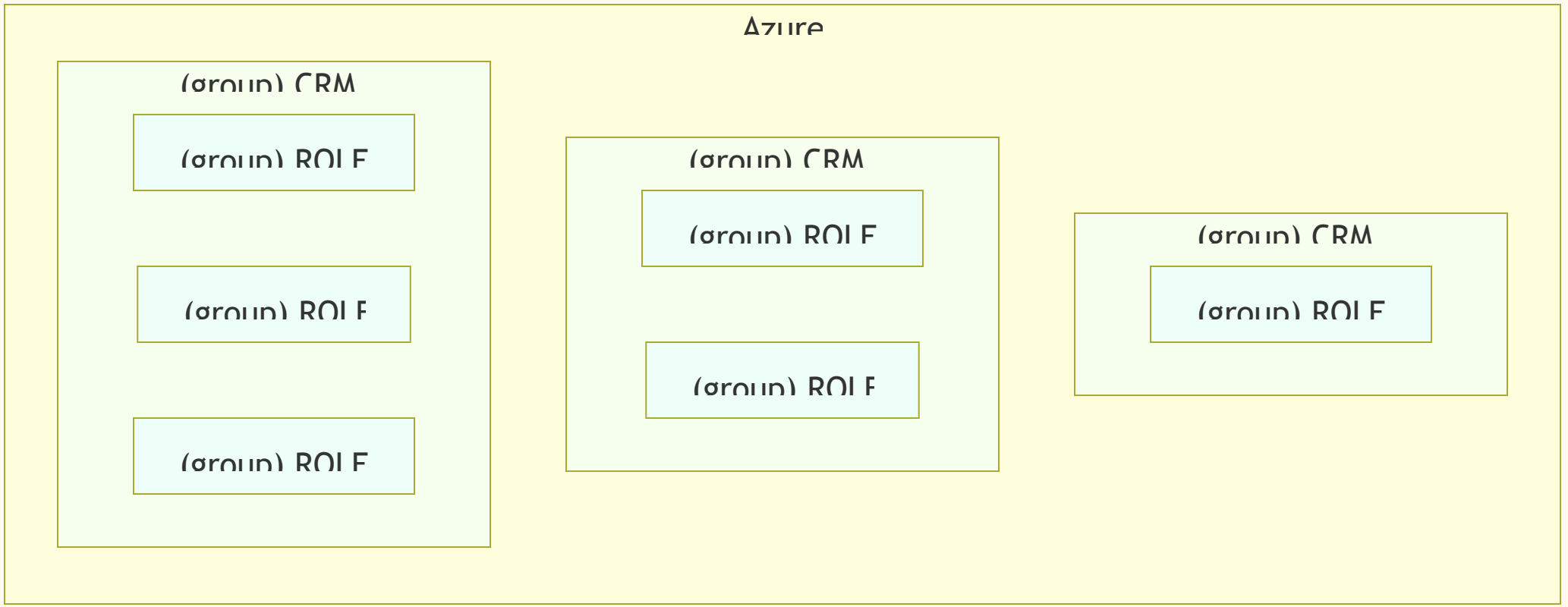
Permissions in AD



Permissions in AD



Permissions in AD



User Permissions

Input

`fkarailanidis`

Output (direct groups)

`ROLE_SALES`

User Permissions

Input

`fkarailanidis`

Output (direct groups)

`ROLE_SALES`

Input

`fkarailanidis`

Output (direct & nested groups)

`ROLE_SALES`, `CRM_VIEW_CLIENT`, `CRM_EDIT_CLIENT`

The Problem 🤦

The Problem 🤦

Access Token Group Claims - Direct assignment only

The Problem 🤦

Access Token Group Claims - Direct assignment only

```
{
  "family_name": "Karailanidis",
  "given_name": "Filippos",
  "groups" : ["ROLE_SALES"],
  ...
}
```

The Problem 🤦

Access Token Group Claims - Direct assignment only

```
{
  "family_name": "Karailanidis",
  "given_name": "Filippos",
  "groups" : ["ROLE_SALES"],
  ...
}
```

When you assign a group to an application, only users in the group have access. The assignment doesn't cascade to nested groups.

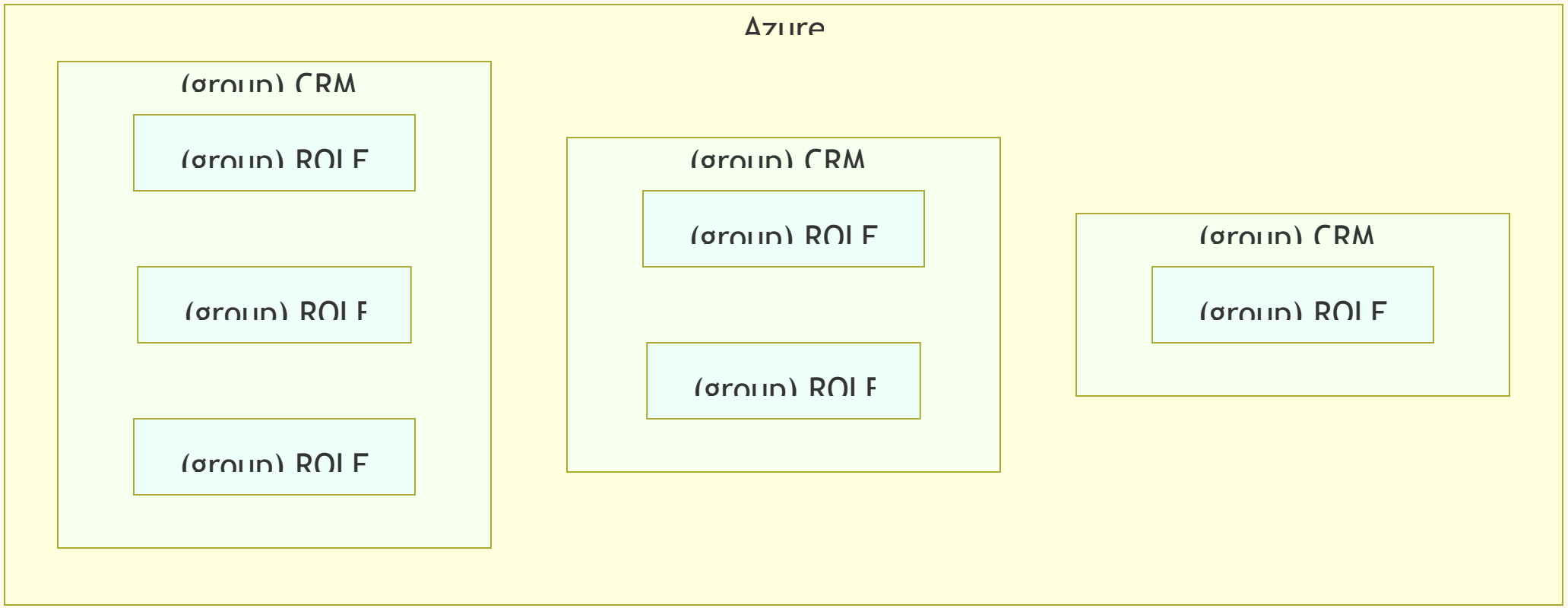
Group-based assignment requires Microsoft Entra ID P1 or P2 edition. Group-based assignment is supported for Security groups, Microsoft 365 groups, and Distribution groups whose `SecurityEnabled` setting is set to `True` only. Nested group memberships aren't currently supported. For more licensing requirements for the features discussed in this article, see the Microsoft Entra pricing page.

https://learn.microsoft.com/en-us/entra/identity/enterprise-apps/assign-user-or-group-

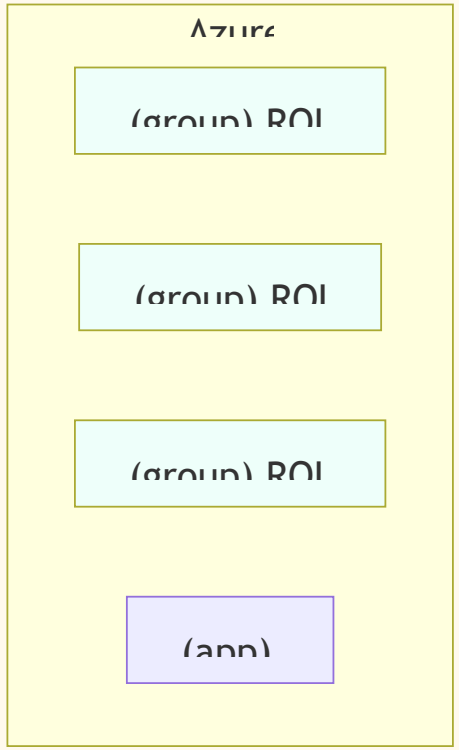
The Solution

The Solution

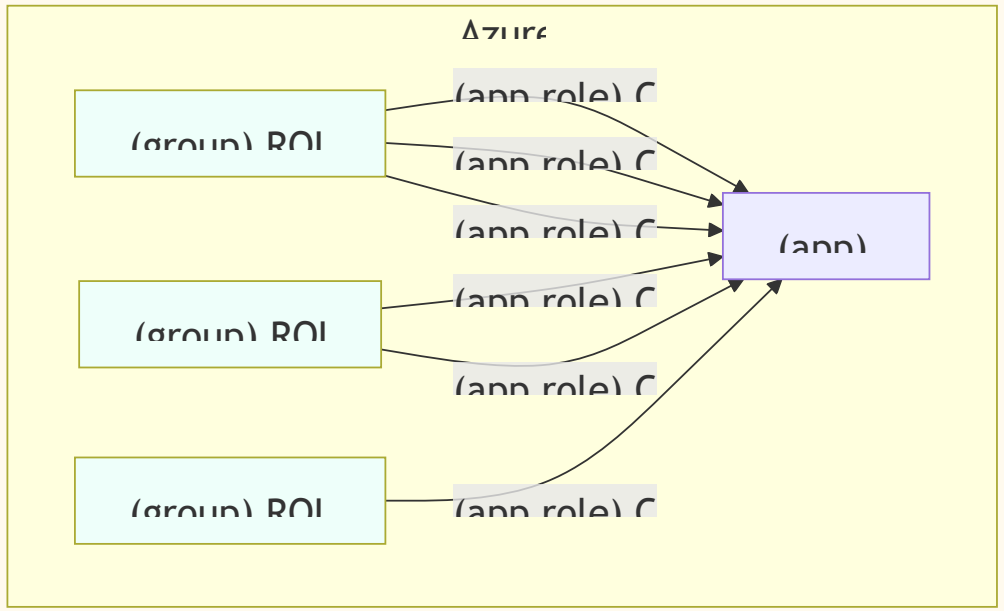
App Roles


Azure
(group) CRM
(group) ROLE
(group) ROLE
(group) ROLE
(group) CRM
(group) ROLE
(group) ROLE
(group) CRM
(group) ROLE

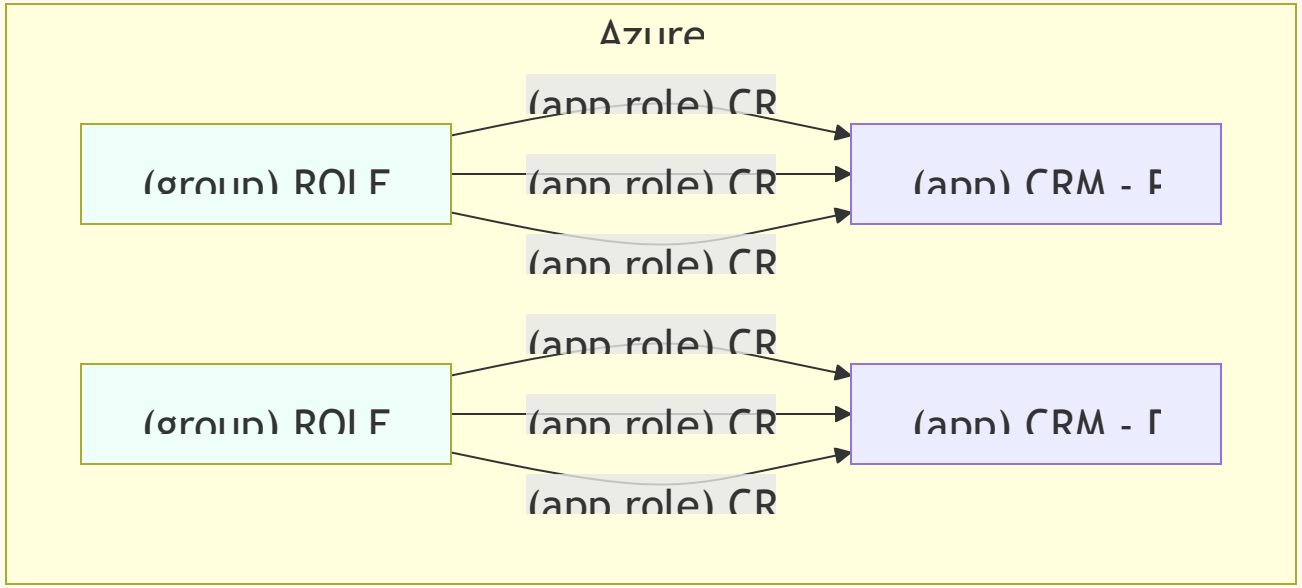
App Roles



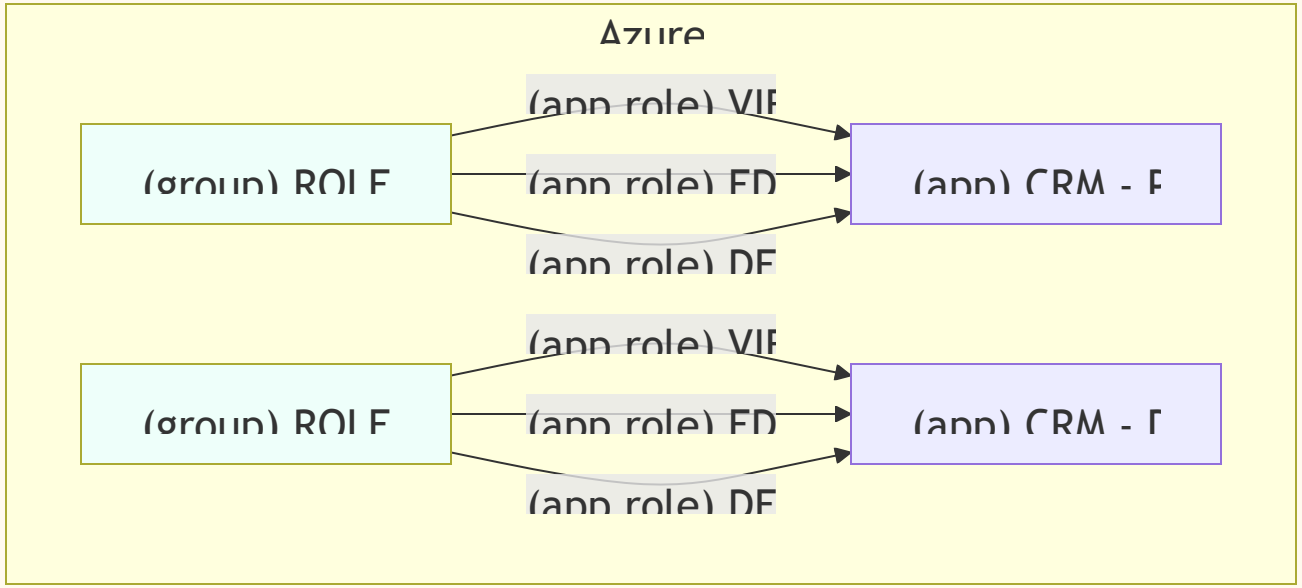
App Roles


Azure
(app role) C
(group) ROI
(app role) C
(app role) C
(app)
(app role) C
(group) ROI
(app role) C
(group) ROI
(app role) C

App Roles - Environment Separation

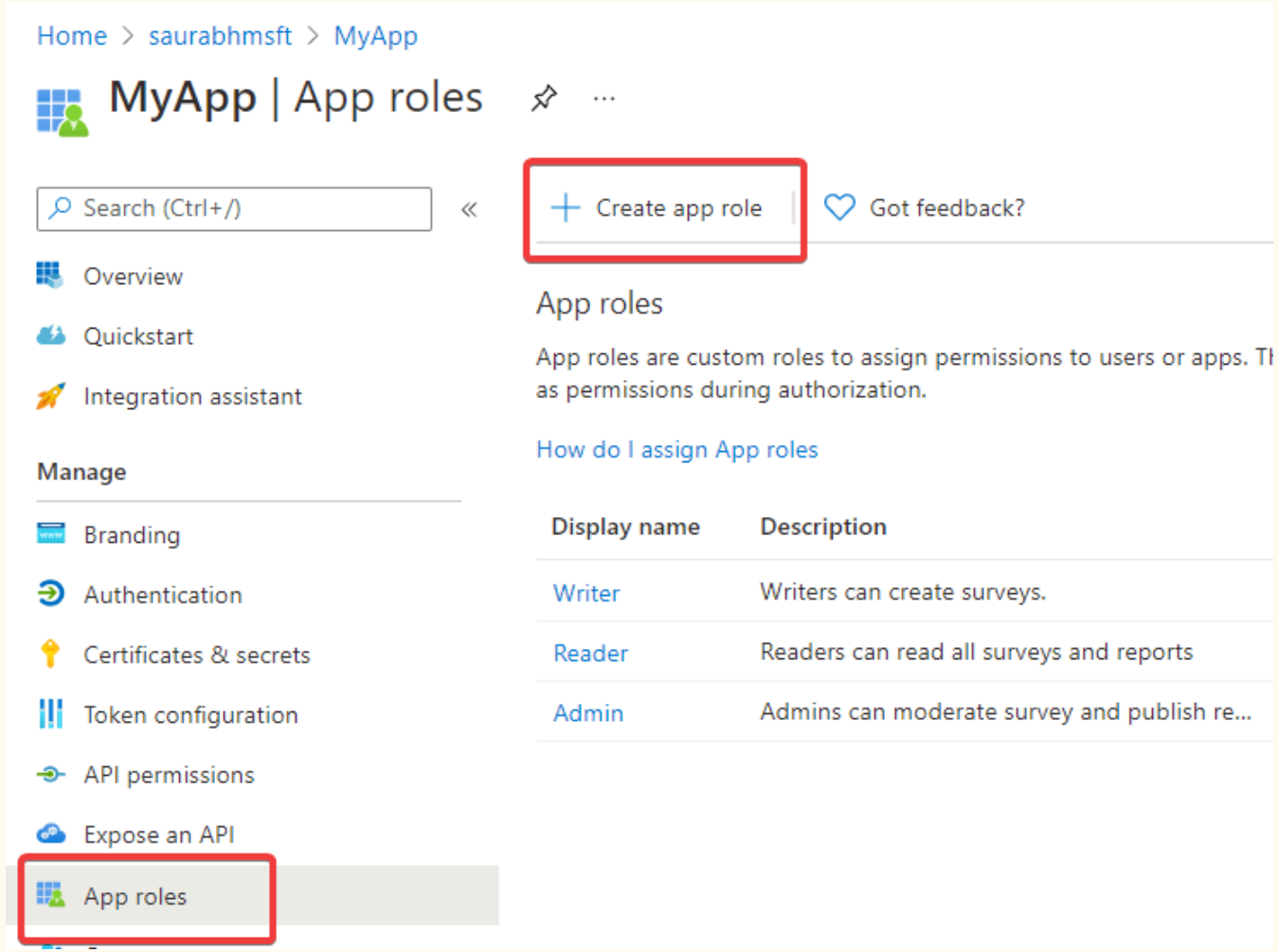


App Roles - Environment Separation



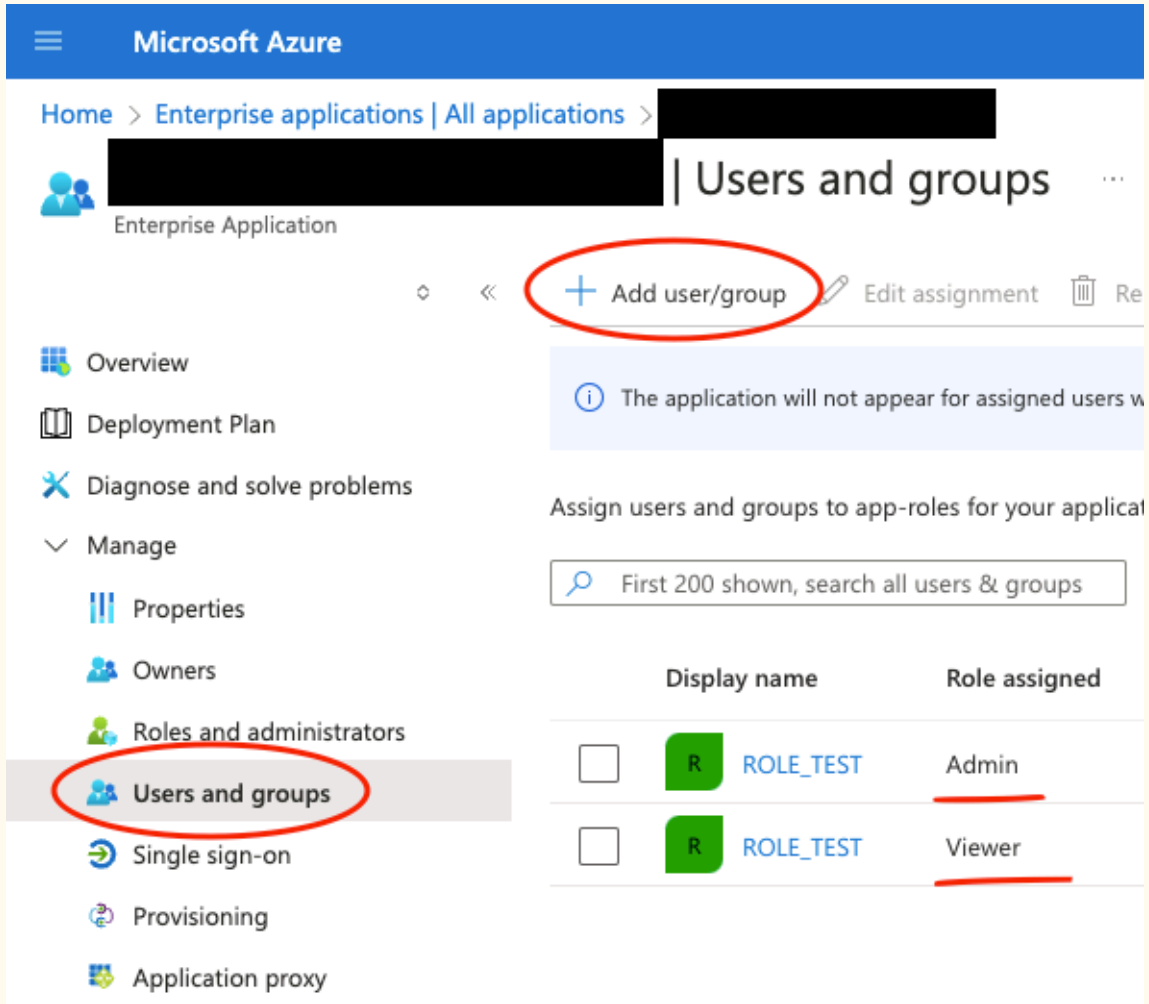
App Roles - Create

Azure Portal > App Registration > App Roles > Create app role



App Roles - Assign

Azure Portal > Enterprise Applications > Users and groups > Add user/group



💡 App Roles

Access Token Claims

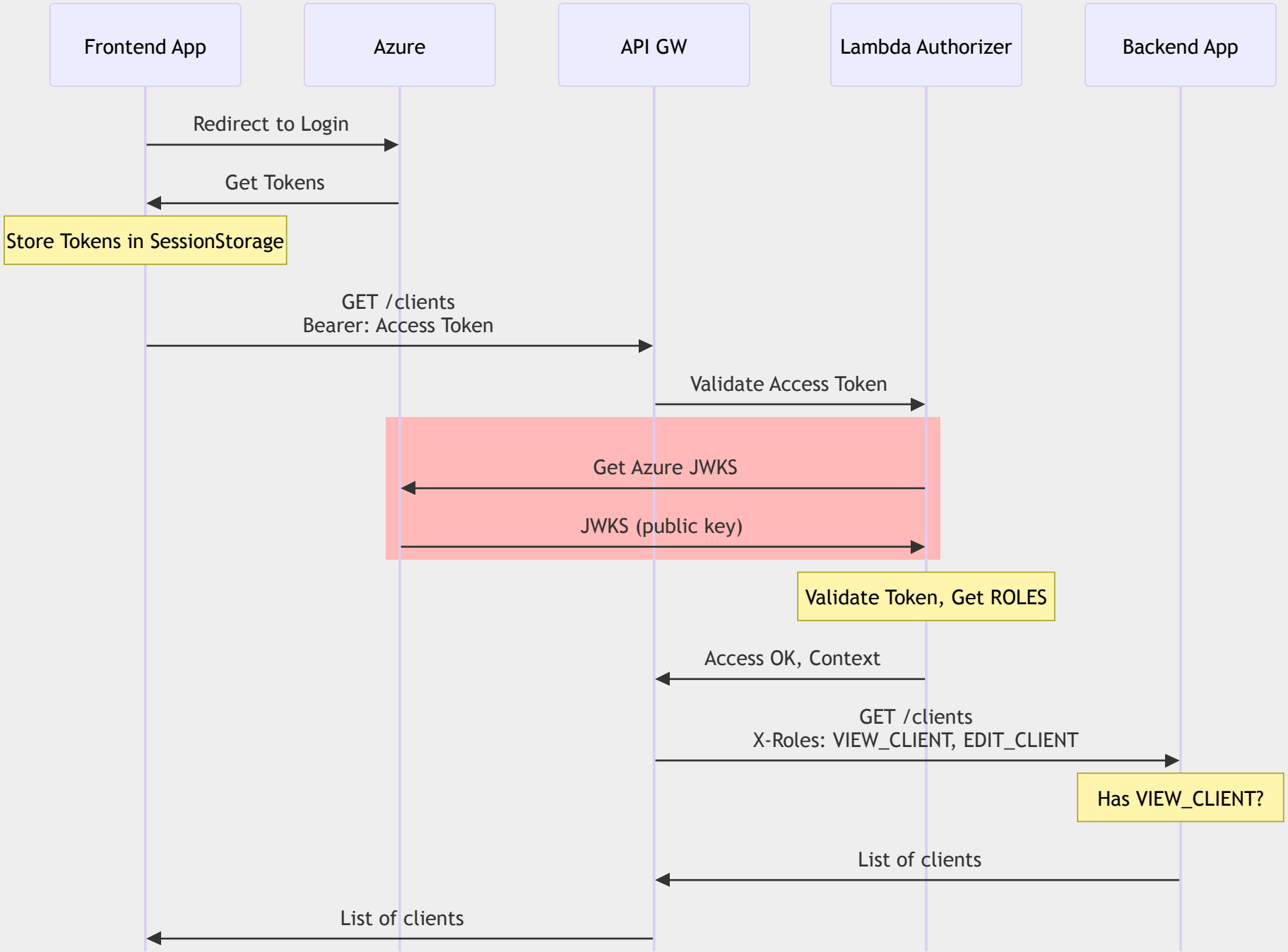
```
{
  "family_name": "Karailanidis",
  "given_name": "Filippos",
  "groups" : ["ROLE_SALES"],
  "roles" : ["CRM_VIEW_CLIENT", "CRM_EDIT_CLIENT"],
  ...
}
```


Gateway
JWT
Groups
Middle Earth
Ware
North
Forochel
Forodwaith
Grey Mountains
Angmar of old
Eriador
Arnor
Rhovanion
Iron Hills
Forlindon
Blue Mountains
Lune
Gulf of Lune
Harlindon
Eregion
Minhiriath
Fangorn
the Brown Lands
Sea of Rhûn
Ash Mountains
Mordor
Mount Doom
Bay of Belfalas
Lebennin
Harondor
Haradwaith

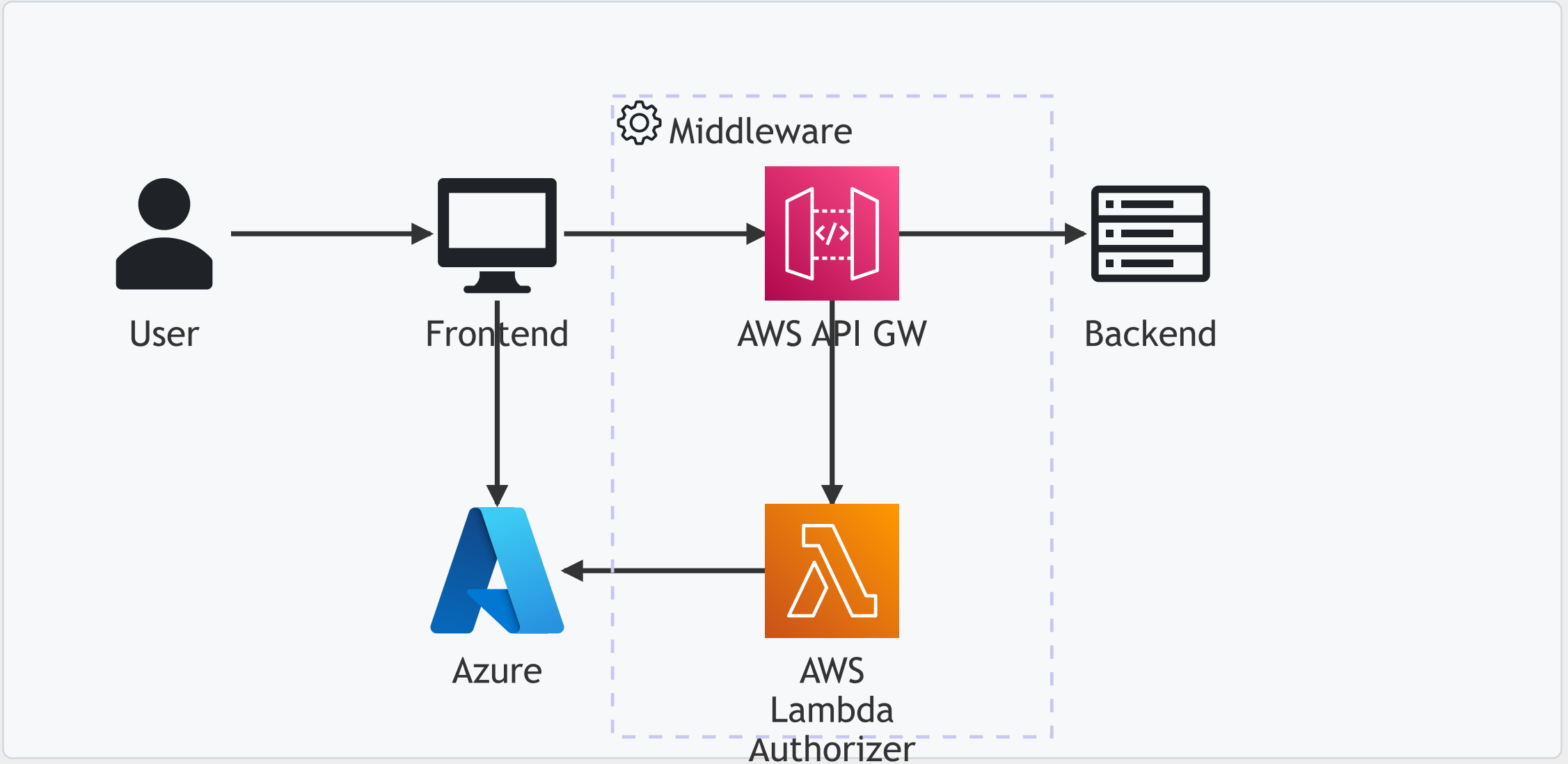
Summary

Finally we end up with the following flow:

- Users authenticate in the frontend
- They get back 2 tokens with the proper claims
- Frontend reads ID token for presentation logic
- AWS API GW receives Access token with API calls
- Lambda Authorizer Validates Token (Authentication)
- Roles are set in context, and forwarded as HTTP headers
- Backend decides if action is allowed (Authorization)


Frontend App
Azure
API GW
Lambda Authorizer
Backend App
Redirect to Login
Get Tokens
Store Tokens in SessionStorage
GET /clients
Bearer: Access Token
Validate Access Token
Get Azure JWKS
JWKS (public key)
Validate Token, Get ROLES
Access OK, Context
GET /clients
X-Roles: VIEW_CLIENT, EDIT_CLIENT
Has VIEW_CLIENT?
List of clients
List of clients

Where we finally arrived



Possible issues

- HTTP Header Size limit (usually 8KB)
- API Gateway timeout - default 29s
- SSE Events not supported by AWS API GW
- gRPC needs web layer

Thank you!

Q&A