The Judgement of Solomon

September 6, 2016

1 The Problem

There are many services that will manage resources. The challenge is how do we assign resources to the appropriate users and manage access control without requiring each service detailed knowledge and logic of the permission system?

2 The Solution

Each service that manages data needs to group data by some mechanism. For our purposes we will call it a scope. Each request to a service will have a claim to a scope inside a JWT token. This token will be signed by a service called Solomon.

3 Solomon is Wise

Solmon's job is to add claims to scopes inside the JWT tokens. Users who log in will get a JWT token with all the claims they have inside the JWT token.

Solomon will maintain a mapping table of users to roles and roles to claims.

Solomon will walk the mapping tables and add the claims to the users JWT, then sign the JWT and return it.

The user can then pass this JWT token to a downstream service to retrieve the resources they have claims to.

In addition to creating JWT tokens, Solomon will also provide endpoints to manage accounts, users, roles, scopes, and organizations.

4 Many Shepards, Many Sheep

Since FoxCommerce will support many types of eCommerce systems which may have varying permissions systems and business logic around access, we want to avoid building all that logic within Solomon.

We do this by taking the responsibility for maintaining what the different roles, users, and organizations are created in another service called a Shepard.

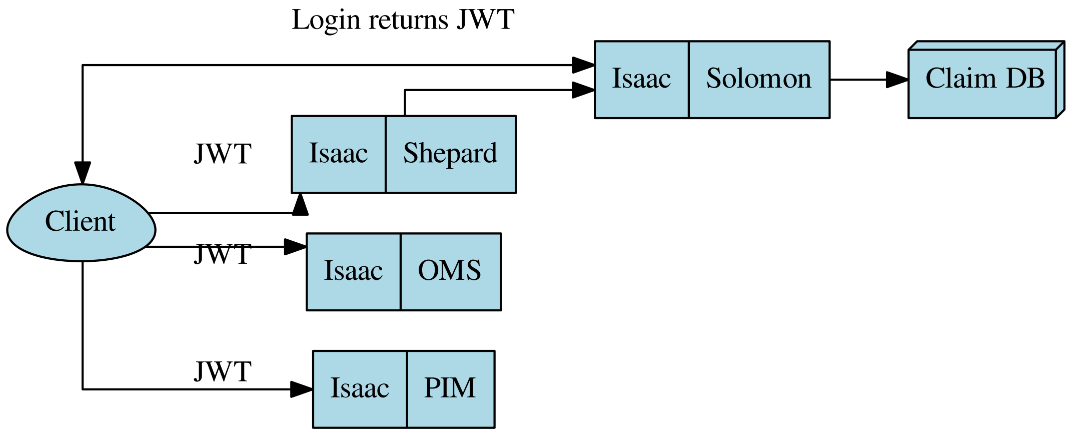
For example, a simple one storefront customer will have separate security requirements than a marketplace.

We can build out different Shepard services depending on the needs of our customers.

All Solomon can do is understand the mapping table and follow it. It doesn't care or understand what the roles and claims actually represent. It doesn't understand that an organization may be a merchant or vendor, for example.

5 Design

1. Each system groups resources under scopes.
2. JWT tokens are used to make claims.
3. Solomon understands the user to scope, and scope to scope mapping tables.
4. Shepard services maintain the scope mapping tables.
5. Isaac validates JWT tokens are authentic.



6 Scopes

Scopes are a hierarchical id that is used to group resources. Scopes can have child scopes.

For example, lets say we have a scope...

"1"

Which represents a tenant. That tenant can add merchants scopes and adds the first merchant

"1.1"

And then second merchant gets

"1.2"

The tenant has de facto access to "1.1" and "1.2" because their scope "1" is a prefix match to those other scopes.

Services can do a prefix match on the claimed scope to see if the client has access to the requested scope.

7 Claims

Solomon will add claims to JWT tokens. The claims will be represented as URIs to the resources under some scope.

The format of the URI is.

”frn:<system>:<resource>:<scope>”

Where the order grows in specificity. ”frn” stands for ”Fox Resource Name” and is modelled after a URN which is a type of URI.

For example, a claim to orders may look like this.

”frn:oms:order:1.2”

Where the first part of the URI is the resource ”frn:oms:order” and the second part is the scope ”1.2”.

A claim in the JWT above is divided into the resource and the action allowed. Which actions are allowed for a resource depends on the service managing that resource.

An admin user for organization ”z” which maps to scope ”1.2” might have claims to all use orders.

```
{
    ”frn:oms:order:1.2”: [”c”, ”r”, ”u”, ”d”]
}
```

Where the actions represent create, read, update, and delete.

8 Use Cases

8.1 Assigning Users to Roles

Solomon’s job is to validate claims, however the service is not responsible maintaining and augmenting the role mappings. That job is delegated to another service we shall call the Shepard service.

8.2 User requests access to a resource from a service

The user will give the service the JWT token. In front of the service Isaac will validate the JWT token. The service can then decode the JWT token and check that the user has a claim to the requested resource.

8.3 User requests a new resource from a service

Similar to the access request use case, the JWT token is validated by Isaac. The service can then create the new resource giving it the scope the user requested as long as they have a claim to data under that scope.

8.4 Careful Considerations

- How do we put Isaac in front of services, nginx?
- Does it make sense the way Isaac validates users and Solomon roles?