Authorisation

CMPU4023 - Enterprise Application Development

Authorization

- The term *authorise* will means to grant or reject access to specific operations on behalf of an API requester
- A prerequisite for authorisation (usually) is that the requester identity has already been verified, that is the requester has been authenticated
- By operation, we mean a query or an update of one or more service-managed resources
- Authorisation asks the question “does this identity have rights to carry out this operation on this resource?”
- For example, can a specific user add a new product to a catalog or change the attribute value of a product item?

Authorisation Authority

- In the context of an enterprise application stack, the authorisation checking would likely be carried out centrally within some component of the overall identity management system
- The benefit of centralisation of authorisation is similar to the benefit of centralised management of the authentication function, namely:

  - **Identity** - The same user or entity (having a single identity everywhere) can be managed and tracked across all of their application accesses
  - **Security** - Centralisation enforces consistency as to how privileges are applied and standards are maintained across the enterprise IS suite; In addition, it is easier to take remedial action in the event of a security breach
  - **Cost Control** - Centralisation allows for the reduction or elimination of duplication which should be more cost effective

Granularity and Control

- At a high level, authorization is a mechanism for protecting assets from unauthorised access or modification
- Depending on the application, some assets may be open for reading and writing by anyone whereas others would be restricted to a small number of privileged users
- Authorisation schemes are characterised by the level of granularity and control they offer
- These range from crude CRUD-level checking to specific named operations on specific attributes of a resource

Granularity Illustration

- Suppose there exists a products table describing items for sale
- Users could be authorised at the level of creating new products, reading product descriptions, changing product descriptions or removing products
- Or alternatively, the authorisation could go down to the level of authorisation reads and update to a specific product attribute or could include rights to carry out other operations on the resource such as commenting on a product or sharing the product with a third-party
- In general, the more sophisticated the sets of operations support on a resource, the more granularity and control that is required

Rights and Privileges

- Rights (or privileges) are what determine what allowed to be carried out on a resource
- An authorisation is essentially a conferral of zero or more rights (associated with a resource) to a particular identity
- Resource rights can be granted or revoked and the managing services are expected to honour these rights contracts in their API implementations

Role-Based Access Control

- A popular way to model and implement authorisation schemes is to control access to resources and operations through a role abstraction
- A role is a convenience grouping of set of rights associated with an identity
- An identity may be assigned one or more roles
- The benefit of a role-based approach is that it simplifies the mapping of rights to identities
- Roles recognise the reality that many of the same kinds of users will want similar rights so the role acts as a convenience shorthand, albeit at the cost of an extra level of indirection

Authorisation Failures

- From an API perspective there is a question as to how to handle authorisation failures, i.e. requests for which the caller has insufficient rights to carry out, including, say, read operations
- There are two popular approaches each with different security characteristics

1. Return an **“not authorised”** status code to the caller (e.g. HTTP 403). This has the side-effect of validating the existence of the resource(s) but reporting that the attempted operation is not allowed
2. Return a **“not found”** status code to the caller (e.g. HTTP 404). This has the benefit of not leaking confirmation of the existence or non-existance of the resource(s) but is indistinguishable from either

Summary

- Authorisation (as distinct from authentication) has to do with managing access to specific operations on service-managed resources
- A right describes what is allowed to be carried out on a resource
- An a specific authorisation confers a right over a resource to an a specific identity
- For convenience, rights are often managed as roles, which are sets of rights associated with a user

Authorisation

CMPU4023 - Enterprise Application Development