ASP.NET Identity

Authentication, Authorization, Claims, Tokens, and OWIN

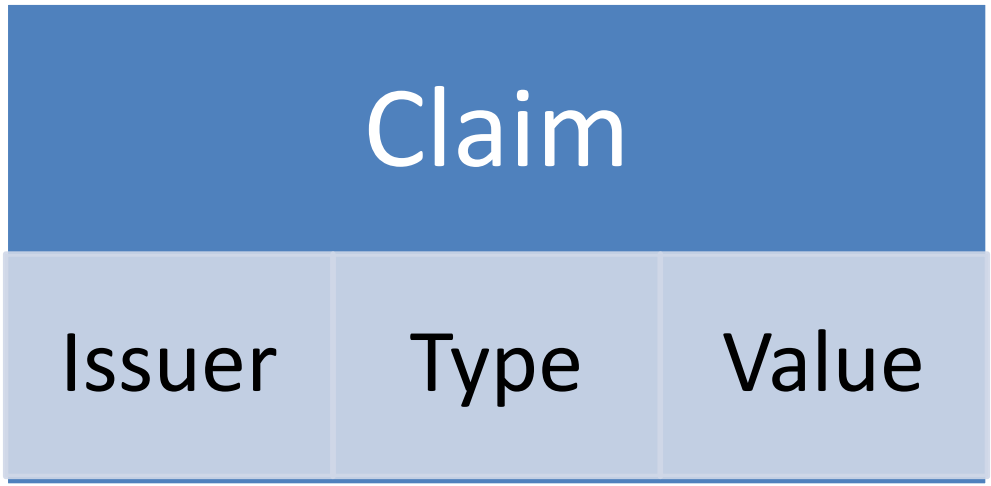
Claims based authentication

What is a claim?

A Claim is a statement about a subject, for example your name, age or address.

Each claim has a **type**, a **value** and an **issuer**.

- A type is for example “name” or “address”
- A value is for example “Billy” or “Example street 11”
- An issuer (provider) is an entity which can issue claims.



What is a Claim?

Since you're not always the sole provider of a claim, it is up to you to decide if you trust the claim or not (depending on the issuer).

For example you might trust a claim which has been issued by the government, while a claim from another source might not always be seen as a trusted issuer.

Claim		
Government	Social Security Number	860228-4792

Claim		
Associate	Social Security Number	840228-4792

Why Claims?

Why Claims?

Benefits of claims

- Reduces the load on the server since the user often provides the claims (Which the user received from a Security Token Service).
- Authorization can be decided based on claims, making it more dynamic than roles-based authorization.
- The user brings the claims wherever the user goes, making them easy to access.
- The claims are encapsulated in what is called “Security Token” which becomes an encrypted cookie, making it secured.

What is a Security Token?

A security token contains an identity.

An identity is something that defines an entity. In our case the identity contains the claims that belong to the entity.

What is a STS?

What is a STS (Security Token Service)?

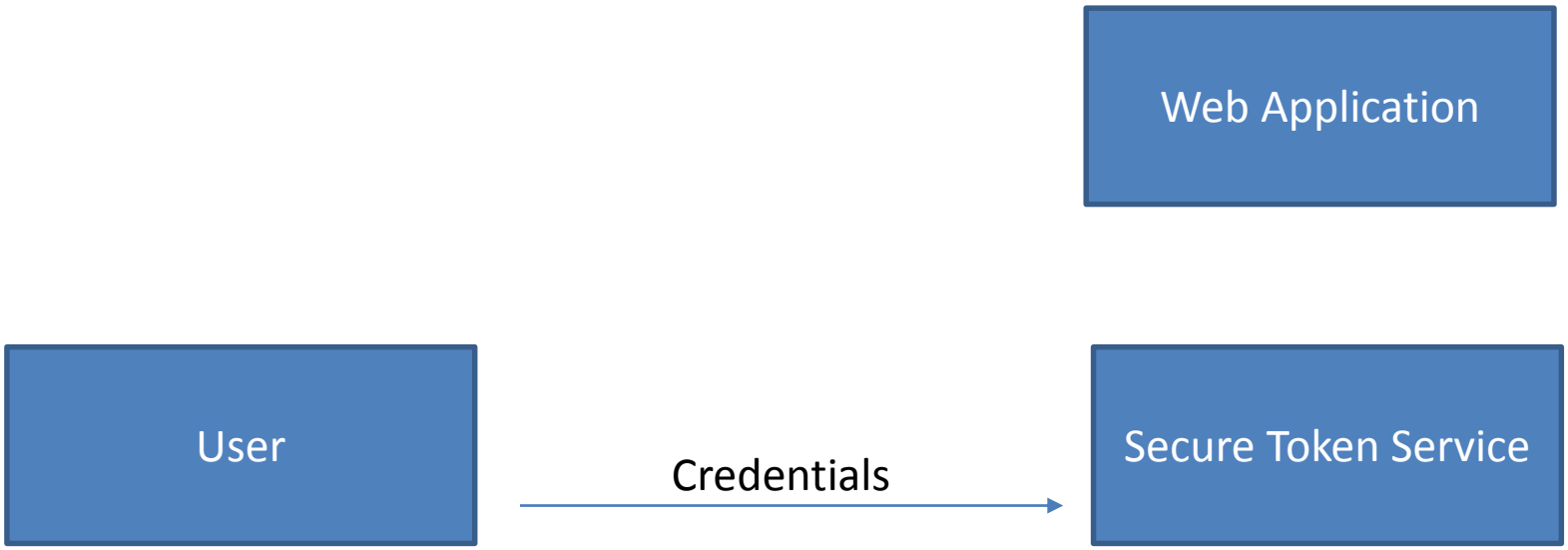
A STS is a service which provides users with security tokens (which is a set of claims).

This token is then used to authenticate the user on the web-application.

STS Flow

STS Flow illustration

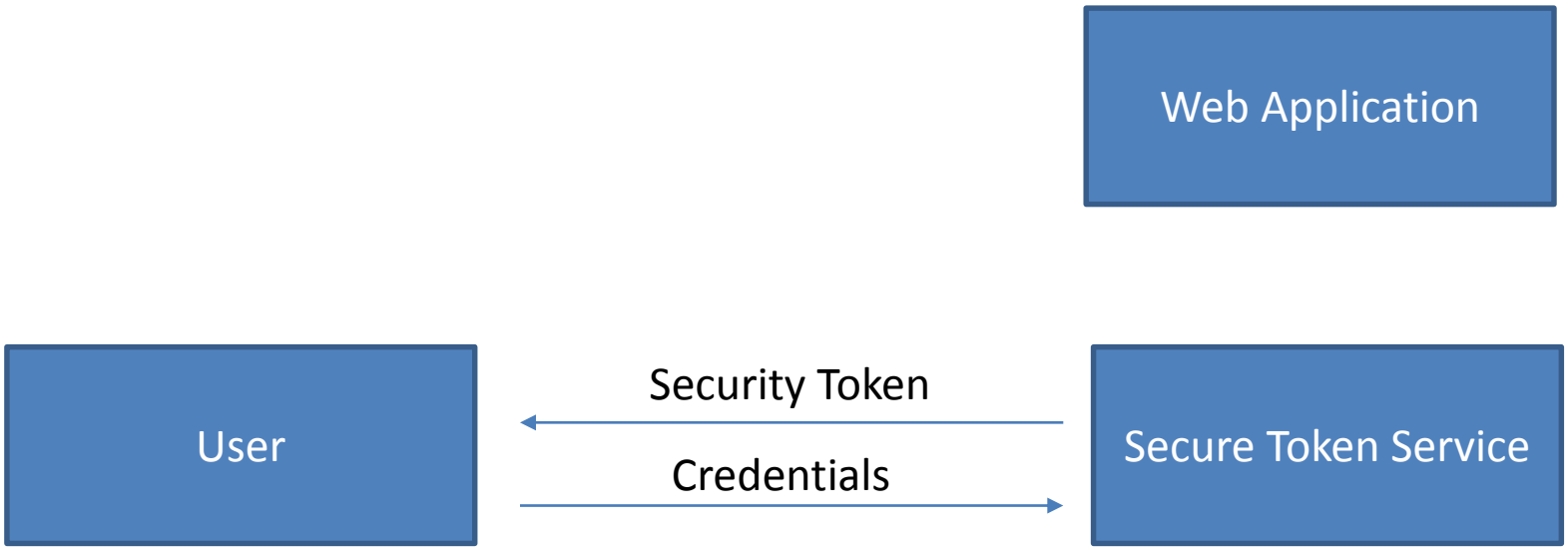
The user provides the STS with his credentials



STS Flow

STS Flow illustration

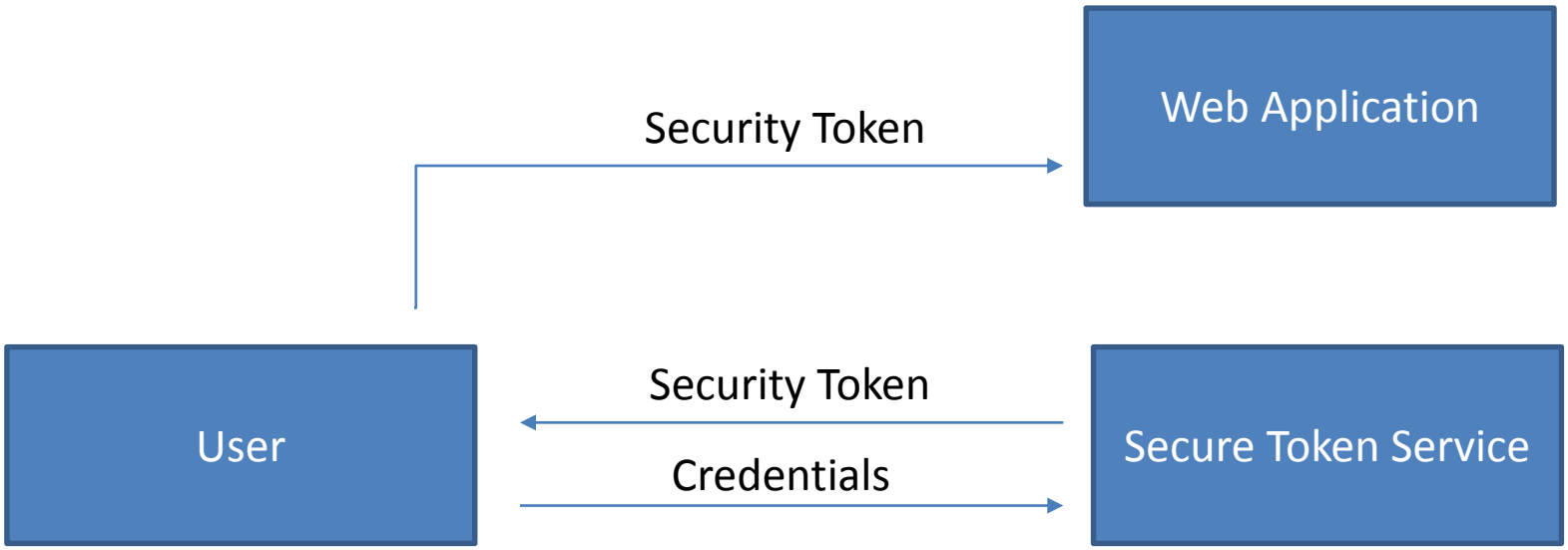
If the credentials are valid, the STS will return a Security token



STS Flow

STS Flow illustration

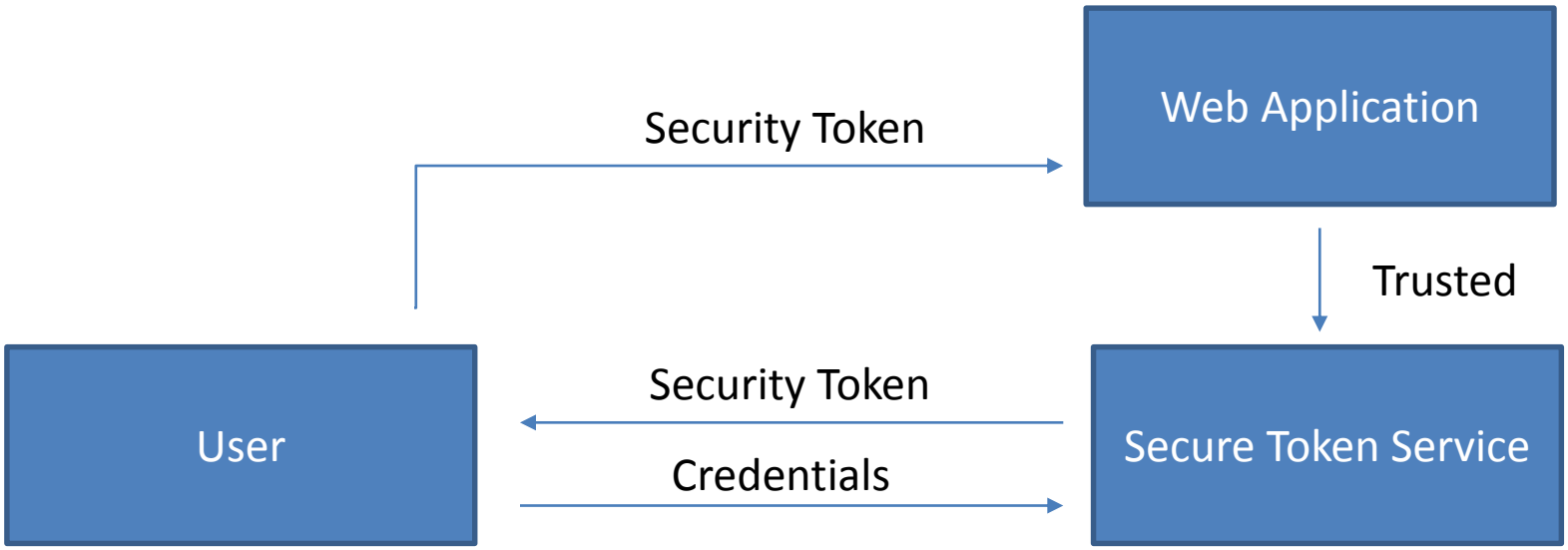
The user provides the web application with the Security Token received from the STS.



STS Flow

STS Flow illustration

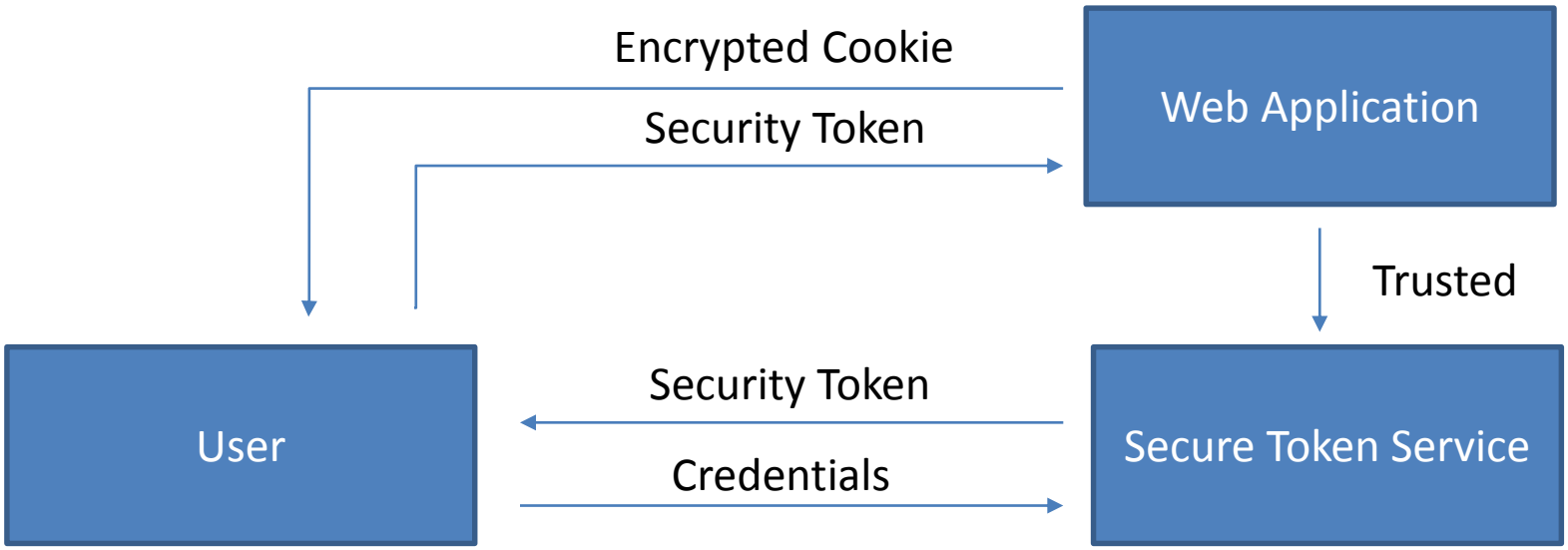
The web application checks if the STS is a trusted issuer



STS Flow

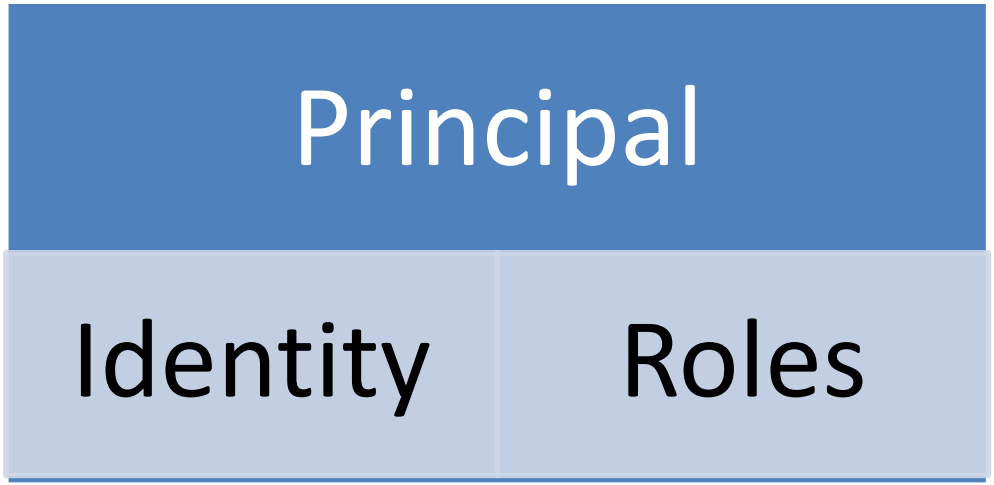
STS Flow illustration

If the STS is trusted the web application returns an encrypted cookie to the user for authentication and authorization purposes



What is a Principal?

A principal object is an identity object including the roles associated with the identity



Role-based authorization vs Claims-based authorization

- Claims based authorization grants more flexibility than role based authorization.
  - Easier to customize your authorization to suit your needs

- Role based authorization has premade attributes which are easy to apply

```
[Authorize(Roles = "Administrator")]
```

OWIN

Open Web Interface

OWIN is a middleware which decouples the server from the application