ASP.NET Identity
Security Fundamentals

Authentication & Authorization

Authentication and Authorization

Authentication verifies who you are

Answers questions like:

- Who is the user?
- Is the user really who he/she claims to be?

Authentication and Authorization

Authorization is how we determines what permissions an authenticated user has.

Answers questions like:

- Is user X authorized to access resource R?
- Is user X authorized to perform action A?
- Is user X authorized to perform action A on resource R?

Real life case

For example you can use your driver license for both authentication and authorization.

You can authenticate yourself by showing your personal information on the license and the license gives you authorization to drive a car.

Roles

What is a Role?

A role is something which defines what you are authorized to do.

Roles are used when there are parts of the application which should only be available for certain users.

What is a Role?

An example of such a role is : Administrator, Employee or Customer

Each role is associated with a set of users.

Roles	Users
Administrator	Billy
Employee	Billy, Marie
Customer	Timmy, Will, Linda

The Administrator might for example be authorized to create and remove users from a web application. While an Employee might only be authorized to read from the web application.

Claims

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 who made the claim.

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.

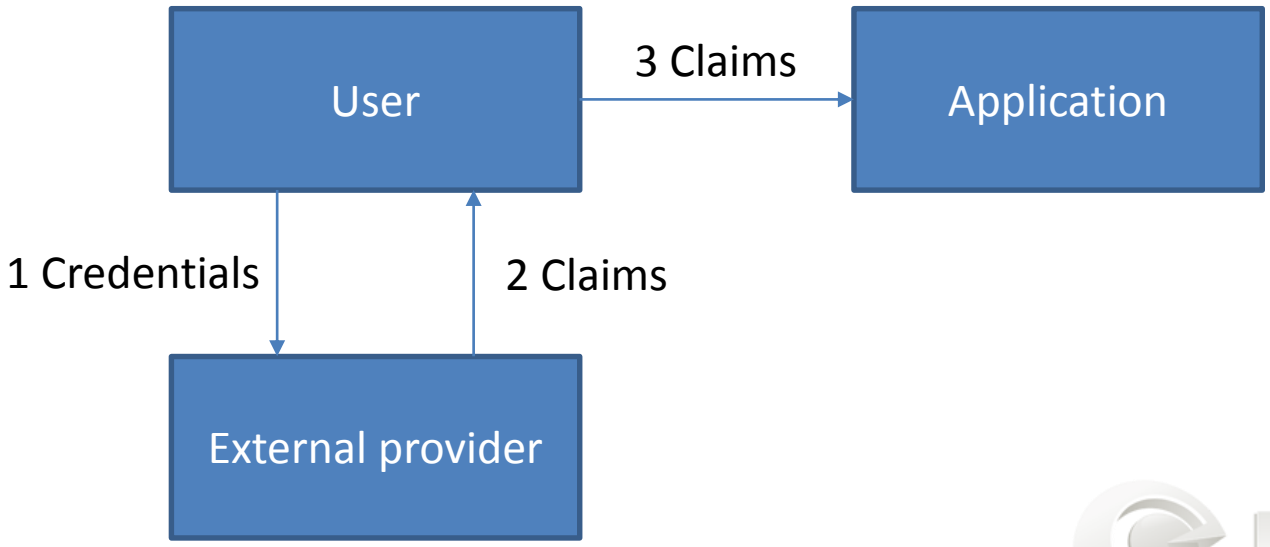
Claim		
Government	Social Security Number	860228-4792

Claim		
Associate	Social Security Number	840228-4792

Why Claims?

Benefits of claims

- Reduces the load on the server since the user can provide the claims from an external provider.



Why Claims?

Benefits of claims

- Authorization can be decided based on claims, making it more dynamic and flexible than roles-based authorization.

Why Claims?

Benefits of claims

- The claims are encapsulated in an encrypted cookie, often called authentication cookie.

  When the cookie (containing the users claims) is created, ASP.NET Identity encrypts the cookie before storing it in the browser.

Why Claims?

Benefits of claims

- The user brings the claims wherever the user goes, making them easy to access.

  Since the claims are saved in an encrypted cookie in the browser, the user will bring it wherever the user goes which makes it easy to access.

Why Claims?

Benefits of claims summary

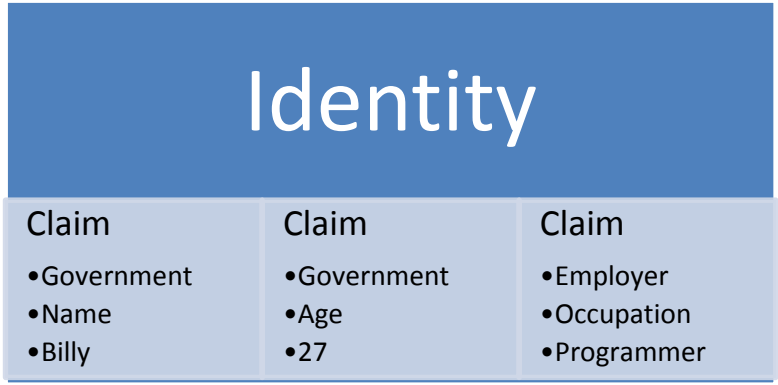
- Reduces the load on the server since the user often provides the claims.
- Authorization can be decided based on claims, making it more dynamic and flexible than roles-based authorization.
- The claims are encapsulated in an encrypted cookie, often called authentication cookie.
- The user brings the claims wherever the user goes, making them easy to access.

Identity

What is an Identity?

An Identity is something which defines who someone is.

An Identity can contain several claims.



What is an Identity?

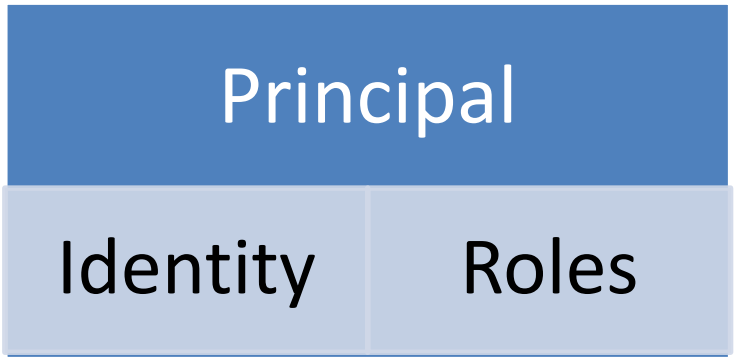
An Identity can be used to authenticate & authorize a user since it contains a set of claims, containing data about an Identity.

Each Identity has to contain at least one unique value which is used to tell Identities from one another.

Principal

What is a Principal?

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



What is a Principal?

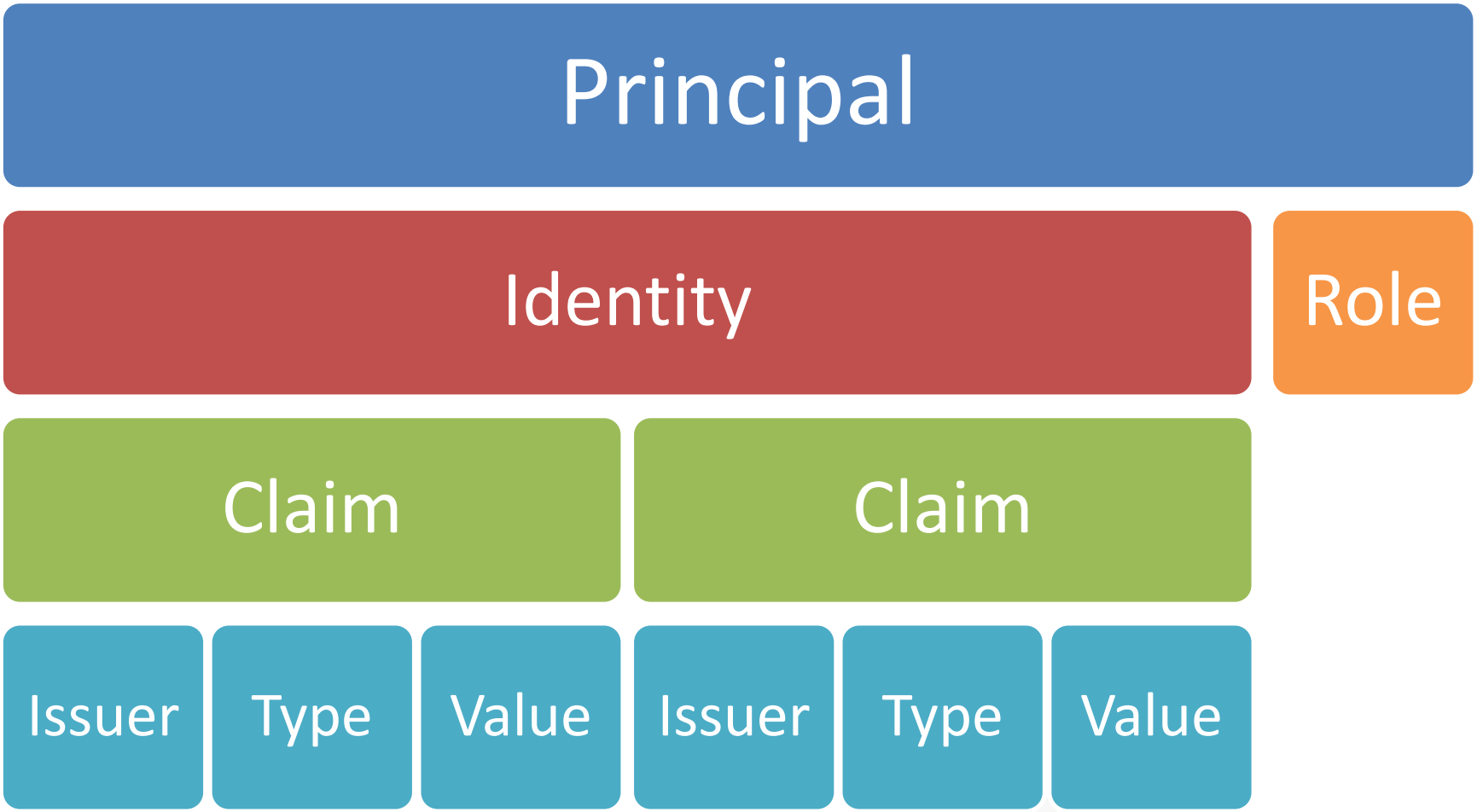
An example of what a principal object can contain.

Here we have a user with the username Billy@Example.com who is in three different roles (Administrator, IT and Employee).

Principal	
Billy@Example.com	Administrator, IT, Employee

What is a Principal?

Principal object mapping



Security Token &
Security Token Services

What is a Security Token?

A Security token is a token which is used to authenticate users.

The security token contains an **Id**, **security key** as well as the time from which it is valid & for how long.

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.

What is a STS?

A good example of where a STS is used is when you try to login to your online bank.

The bank requests a code (a security token) from you, which you get by entering your credentials to your authentication device (STS) which will return a code (a security token).

You would then proceed to use this code to login.

STS Flow

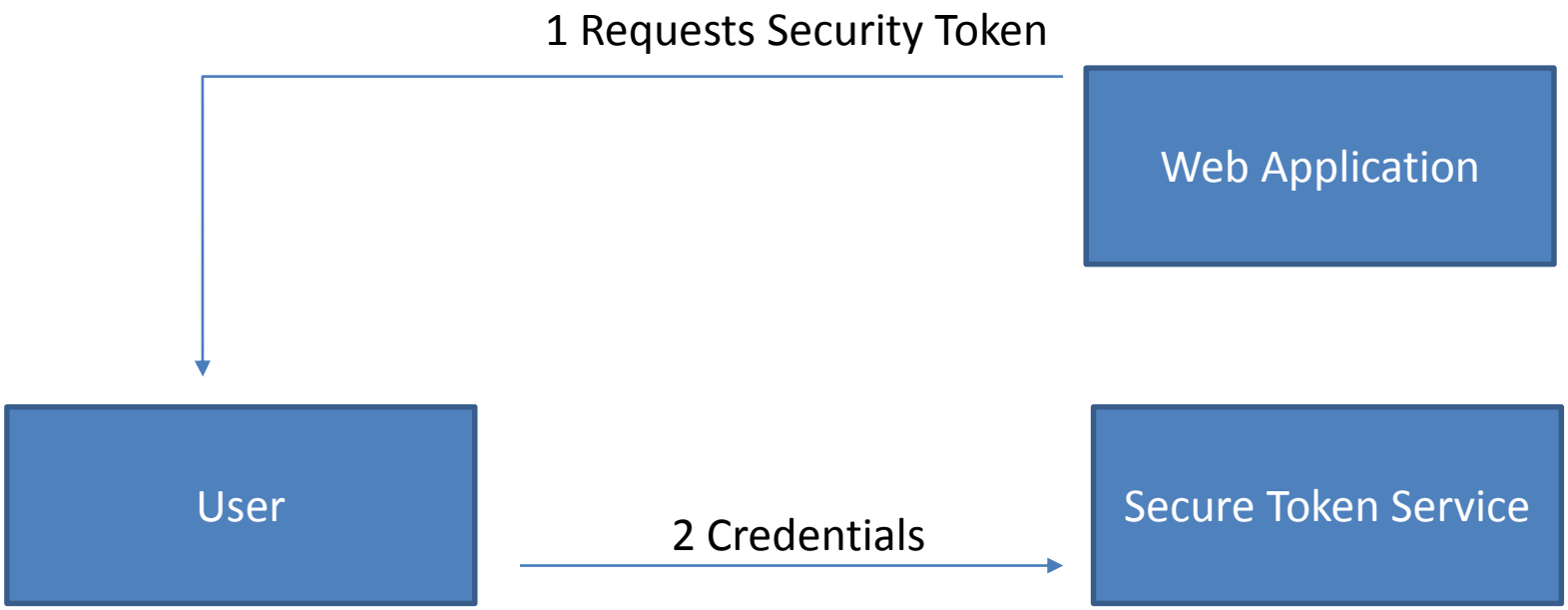
STS Flow illustration

When the user tries to login to the web application, the application requests a security token.

STS Flow

STS Flow illustration

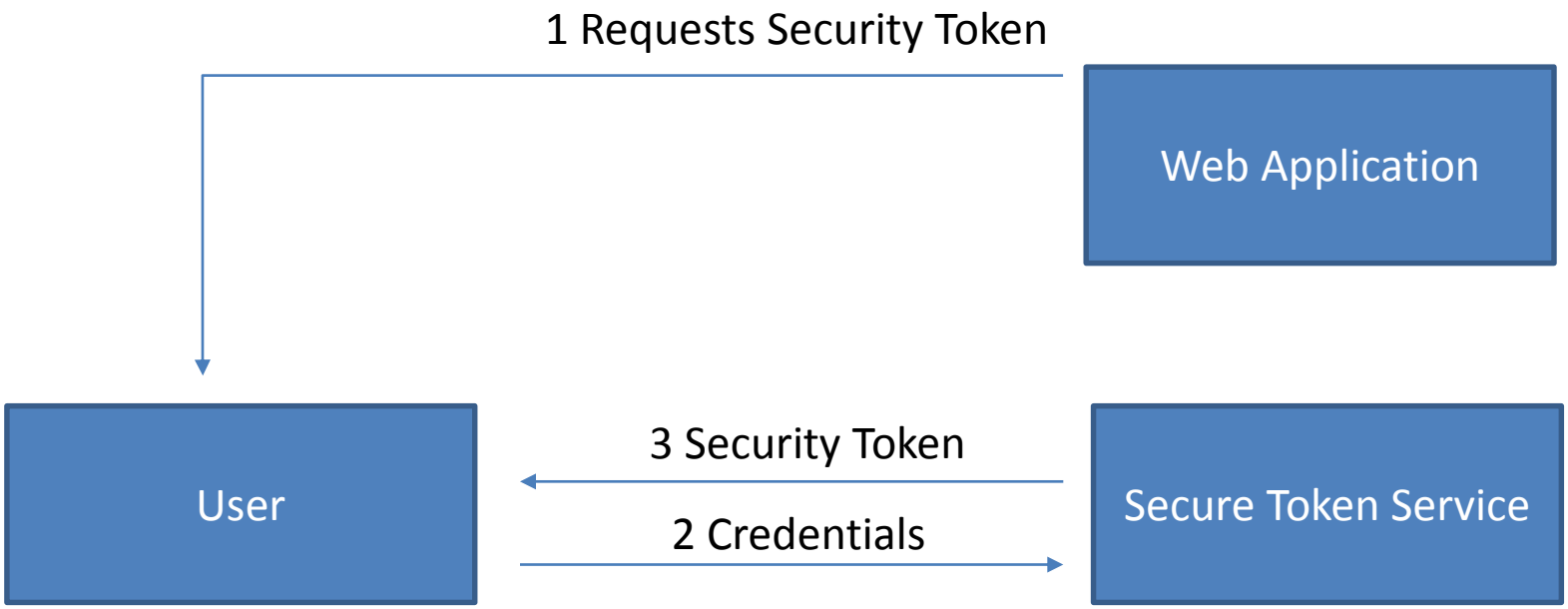
The user provides the STS with his credentials



STS Flow

STS Flow illustration

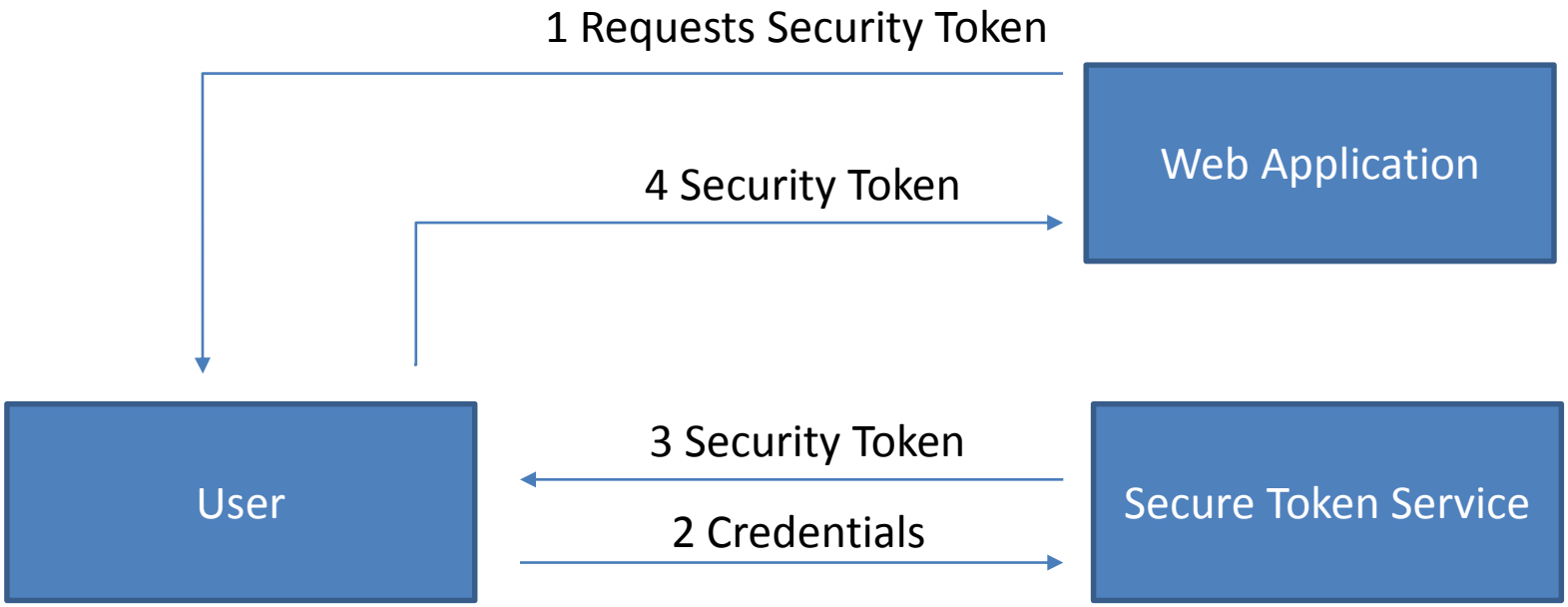
If the credentials are valid, the STS will return a Security token



STS Flow

STS Flow illustration

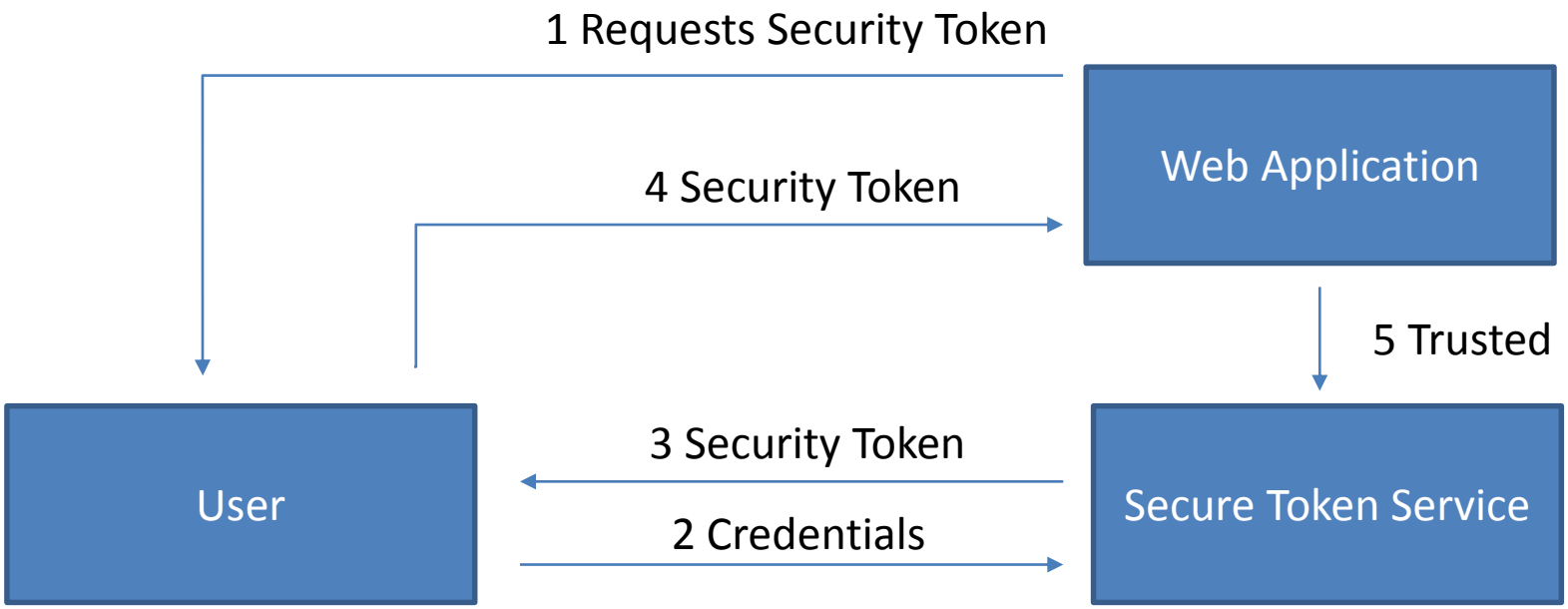
The user provides the web application with the Security Token received from the STS.



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