# Powerpoint markup

## Försättsblad

## What is Asp.Net Identity Framework

* Framework for security and authorization management.
* One Solution for several projects
* Simple Memberships Successor
* <http://www.asp.net/identity>
* Nuget: Microsoft.Aspnet.Identity.EntityFramework and Microsoft.Aspnet.Identity.Owin

## Why Asp.Net Identity Framework

* Compatible with all Asp.Net Frameworks
* Easy use of social login providers
* Claims Based authentication
* OWIN Integration (CookieAuthentication)

## What was before Asp.Net Identity Framework

## Asp.Net Membership

* Forms based Authentication
* Deisgned with only SQL servers in mind.
* No Owin integration.
* No Social Login providers
* Fler fel etc.varför man har gått ifrån det?
* Difficult to customize user information.
* Doesn’t follow SRP (Single Responsibility principle) and therefore also not DRY (Don’t repeat yourself).

## Simple Membership

* Designed for MVC
* Built upon Asp.Net Membership
* Supports Oauth/OpenId
* Tied to Microsoft databases (SQL server, SQL azure)
* Doesn’t follow the principles of SRP and DRY either.

## Asp.Net Identity 1.0

* Released 2013
* Simple way to store and handle user data.
* Role provider
* Claims Based
* Social Login Providers
* Support from .Net 4.5

## Asp.Net Identity 2.0

* Released 2014
* Security Token Provider
* Two-Factor Authentication
* Account Confirmation
* Account Lockout
* Password Reset

## Disadvantages Asp.Net Identity

## OWIN

* Decouple server and application
* A middleware (pipeline between server & application.)
* Bild på Owin pipeline.
* Asp.Net I console app, exempel….. http://www.asp.net/aspnet/overview/owin-and-katana/getting-started-with-owin-and-katana

## Authentication & Authorization

**Authentication – Who are you?**

**Authorization – What are you allowed to do?**

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 authorizes you to be allowed to drive a car.

## Summary

# Claims based Authentication

<papperslapp/id exempel>

* Third party etc.

## What is an Identity?

* An identity is something which defines who you or someone else is.

## What is a Claim?

<insert image here>

* A claim is a statement about a subject for example a name, your age or your home address.
* Each claim has a type, a value and an Issuer
* A Type is for example a “Name” or “Address”
* A Value is for example “Something Road 11”
* An Issuer (provider) is an entity which can issue claims. You can choose how much you trust this claim depending on who the issuer is. A trusted issuer could for example be your government issuing you your driver license.

## Why Claims?

* The user delivers the claim which means you don’t have to look up the information from a database.
* Claims can include more information about an identity than roles, making it more flexible.
* Encapsulated in a token making it easy to access & secure.

## What is an STS?

* Security token service
* An Identity provider which is responsible for issuing security tokens (commonly known as authentication token)

<insert STS flow>

## What is a Security Token?

* A token contains an identity.
* A token is serialized to grant security to your claims.
* Contains a SID (Security Identifier)

## What is a Principal

<insert image here>

* A principal represents an Identity including its roles.
* Vart,när,hur?

## Role-based authorization vs Claim-based authorization

* Claims based authorization grants more flexibility than roles. (Easier to customize your authorization to suit your needs in greater detail).
* Role based authorization has premade attributes which are easy to apply (but might lack the detail you can get from claims).