



---

---

---

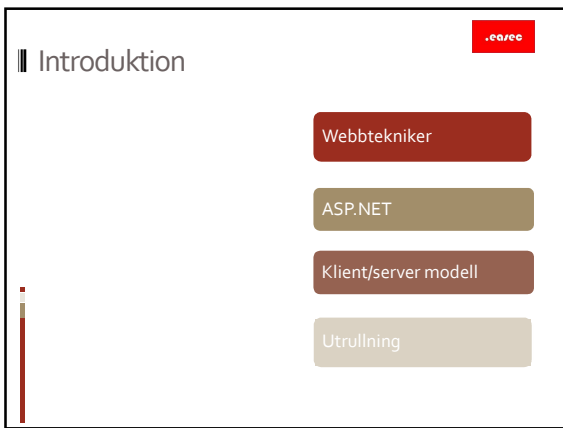
---

---

---

---

---



---

---

---

---

---

---

---

---



---

---

---

---

---

---

---

---

**Översikt Webbtekniker** .eerc

Microsofts webbtekniker  
ASP.NET  
Klientsidan  
Internet Information Server



---

---

---

---

---


---

---

---

**Microsofts webbtekniker** .eerc

Utveckling	Värd	Exekvering	
		Server-Side	Client-Side
<ul style="list-style-type: none"><li>• WebMatrix</li><li>• Visual Studio</li><li>• Visual Studio Code</li></ul>	<ul style="list-style-type: none"><li>• IIS</li><li>• SQL Server</li><li>• Windows Azure</li><li>• SQL Database</li></ul>	<ul style="list-style-type: none"><li>• ASP.NET</li></ul>	<ul style="list-style-type: none"><li>• JavaScript</li><li>• jQuery</li><li>• AJAX</li></ul>



---

---

---

---

---

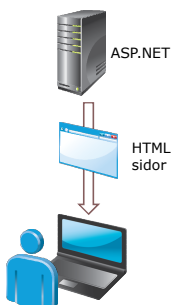
---

---

---

**ASP.NET** .eerc

- Programmingsmodeller
  - Web Pages
  - Web Forms
  - MVC
- ASP.NET API
  - Konfiguration
  - Autentisering och auktorisation
  - Mellanlagring
- Kompilering av ASP.NET kod



```
graph TD; ASP[ASP.NET] --> HTML[HTML sidor]; HTML --> User[User at Laptop];
```

---

---

---

---

---


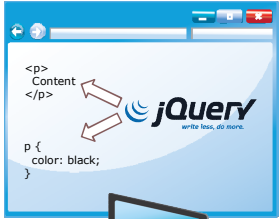
---

---

---

**Klientsida**

- JavaScript
- jQuery
  - jQuery UI
  - jQuery Mobile
- AJAX



---

---

---

---

---


---

---

---

**Internet Information Server**

- IIS
  - Funktioner
  - Skalning
  - Perimeter Networks (DMZ)
- IIS Express
- Andra webbservrar



---

---

---

---


---

---

---

---

**ASP.NET**



---

---

---

---

---

---

---

---

Översikt ASP.NET

- Web Page
- Web Forms
- MVC
- Delade funktioner
- Framtiden

---

---

---

---

---

---

---

---

Web Pages

Web Matrix eller Visual Studio kan användas.

Kodas i .CSHTML filer.

Precis kontroll över HTML-kod.

Designat för enkel server-side scripting.

Snabb teknik att lära, dock ingen kontroll över URL.

- Kan användas
- Kodas i
- Kontroll över
- För enkel
- Snabb teknik

---

---

---

---

---

---

---

---

Web Pages -exempel

```
<h2>Erbjudande</h2>
<p>Plommonmannen har dessa erbjudande:</p>
@foreach (var item in offers) {
  <div class="offer-card">
    <div class="offer-picture">
      @if (!String.IsNullOrEmpty(item.PhotoUrl)) {
        
      }
    </div>
  </div>
}</div>
```

---

---

---

---

---

---

---

---

**Web Forms** .eorec

**Endast** Visual Studio kan användas.

**Kodas i** .aspx filer och "code-behind" filer.

Skapa användargränssnitt genom att **dra control** till sida.

Controllers **tillhandahåller** flera egenskaper och även möjlighet att lägga till händelser.

**Binda control till information.**

- Endast
- Kodas i
- Dra control
- Tillhandahåller
- Till information

---

---

---

---

---

---

---

---

**MVC** .eorec

**Endast** Visual Studio kan användas.

Models **kapslar in** objekt och information.

Views används för **användargränssnitt**

Controllers interagerar med **händelser** som användare gör.

**Kodas i** .cshtml och .cs filer.

Lätt att använda unit test.

- Endast
- Kapslar in
- Användargränssnitt
- Händelser
- Kodas i

---

---

---

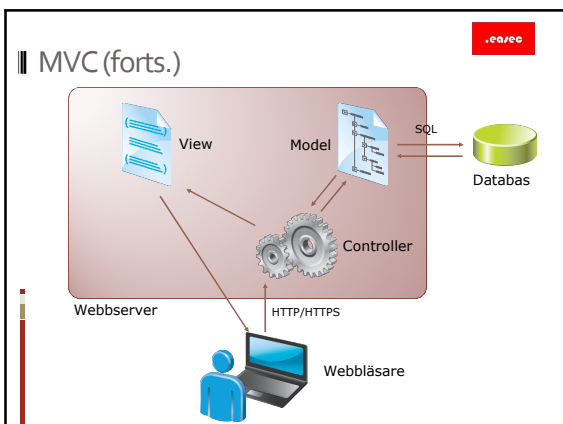
---

---

---

---

---




---

---

---

---

---

---

---

---

**Delade funktioner**

Teknikerna har **delade funktioner**:

- Konfiguration.
- Autentisering.
- Medlemskap och roller.
- Hantering av state.
- Mellanlagring.

Delade funktioner

---

---

---

---

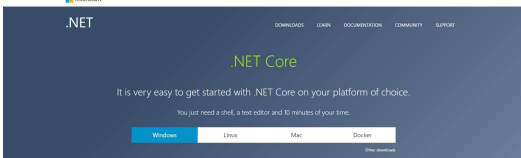
---

---

---

---

**Framtiden**



Windows Linux Mac Docker

---

---

---

---

---

---

---

---

**Klient/server modell**

Klient/server modell

---

---

---

---

---

---

---

---

## Översikt Klient/server modell

Klient/server modell  
 Att tänka på  
 Layout för sidan  
 Rendering  
 Navigering  
 Validering  
 Undantagshantering  
 Presentation layer  
 Data layer  
 Service layer  
 Patterns

---

---

---

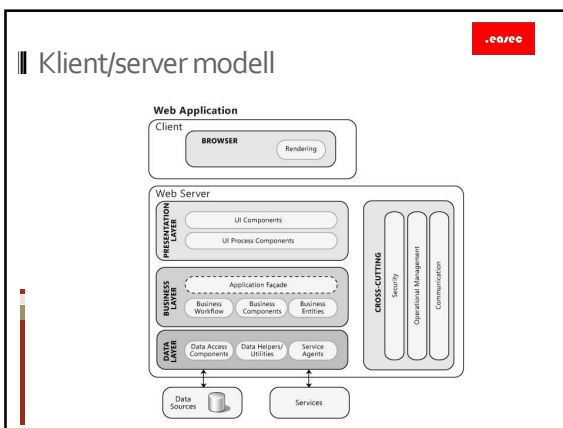
---

---

---

---

---




---

---

---

---

---

---

---

---

## Att tänka på

**Dela upp** din applikation logiskt. Dela upp

**Förstå hur** de olika komponenterna skall kommunicera. Förstå hur

Använd **mellanlagring**. Mellanlagring

Skicka **inte** känslig information oskyddat. Skicka inte

Designa din applikation att arbeta med ett konto med **låga rättigheter**. Låga rättigheter

---

---

---

---

---

---

---

---

## Layout för sidan

Använd Cascading Style Sheets (CSS) för layout **där det** är möjligt.

Table layout **endast när** information skall visas.

Använd ett **gemensamt** utseende på sidorna.

- Använd AJAX server controls och klientbibliotek för AJAX, för att stödja olika webbläsare.

Där det

Endast när

Gemensamt

Använd

---

---

---

---

---

---

---

---

## Rendering

**Överväg att** använda skript på klientsidan eller ASP.NET AJAX, för färre antal post back.

För **snabbare** rendering använd data binding för controller.

Skall **olika språk** användas för användaregränssnitt?

Överväg att

Snabbare

Olika språk

---

---

---

---

---

---

---

---

## Navigering

**Designa** din navigeringsstrategi på ett sådant sätt att den är separerad från logik för bearbetning.

Använd navigeringsfunktioner i Master Page, så funktionen ser **likadan** ut för hela applikationen.

Använd site map **för att hjälpa** användare att hitta sida på site.

Designa

Likadan

För att hjälpa

---

---

---

---

---

---

---

---





## Validering

Designa en **effektiv** lösning för validering av indata är viktig för säkerheten och minska riskerna.

Använd **validering** både på klientsidan och serversidan.

Effektiv

Validering

---

---

---


---

---

---

---

---



## Undantagshantering

Effektiv undantagshantering är viktigt för säkerheten och hur pass bra din applikation fungerar.

Använd **användbara** felmeddelande som visar fel i applikationen utan att lämna ifrån känslig information.

**Försäkra dig** om att du fångar undantag och rensar upp om ett undantag uppstår.

Effektiv

Användbara

Försäkra dig

---

---

---


---

---

---

---

---



## Presentation layer

Skiktet **visar** användare-gränssnitt och ger möjlighet till interaktion med användare.

Skiktet **består av** server-side komponenter som renderar HTML och komponenter på klientsidan som exekverar skript och visar HTML.

Använd AJAX för att **exekvera logik** på klient, oftast för att förbättra användarens upplevelse.

Visar

Består av

Exekvera logik

---

---

---

---

---

---

---

---

**Data layer** .eero

Skiktet **hanterar** logik som är nödvändig för att få tillgång till databas.

Gör det **lättare att** konfigurera och underhålla applikationen.

**Detaljer om** databas göms för andra skikt i din applikation.

[Hanterar](#)  
[Lättare att](#)  
[Detaljer om](#)

---

---

---

---

---

---

---

---

**Service layer** .eero

Skiktet används om Web Service skall användas.

Designa så att komponenter kan återanvändas.

[Hanterar](#)  
[Lättare att](#)

---

---

---

---

---

---


---

---

**Patterns** .eero

Microsoft har **tagit fram** ett antal monster, patterns, som behandlar olika aspekter för utveckling.

[Tagit fram](#)

  
Microsoft®  
**patterns & practices**  
proven practices for predictable results

---

---

---

---

---

---

---

---

Utrullning



Utrullning



---

---

---

---

---

---

---

---

Översikt utrullning

- Non-distributed
- Distributed
- Lastbalansering



---

---

---

---

---

---

---



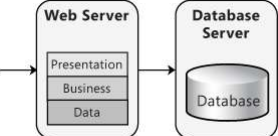
---

Non-distributed

All logik för de olika skikten är fysiskt placerade på samma webserver, förutom databas. Databas **finns på** separat server.

All logik

Finns på



---

---

---

---

---

---

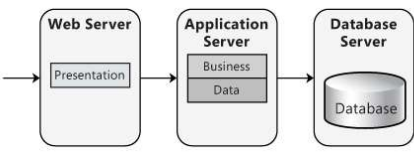
---

---

**Distributed**

Presentations- och business skikten är **separerade**.  
**Oftast** är business och data access skikten på samma server.

Är separerade  
Oftast är



```
graph LR; WS[Web Server  
Presentation] --> AS[Application Server  
Business  
Data]; AS --> DS[Database Server  
Database];
```

---

---

---

---

---

---

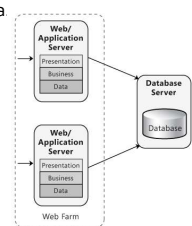
---

---

**Lastbalansering**

Om din applikation läggs på **flera servrar** kan lastbalansering mellan dessa användas, för att distribuera förfrågningar mellan dessa. **Viktigt att** hantera state.

Flera servrar  
Viktigt att



```
graph LR; subgraph Web_farm; WS1[Web/ Application Server  
Presentation  
Business  
Data]; WS2[Web/ Application Server  
Presentation  
Business  
Data]; end; WS1 --> DS[(Database Server  
Database)]; WS2 --> DS;
```

---

---

---

---

---

---

---

---

**Repetitionsfrågor**

---

---

---

---

---

---

---

---