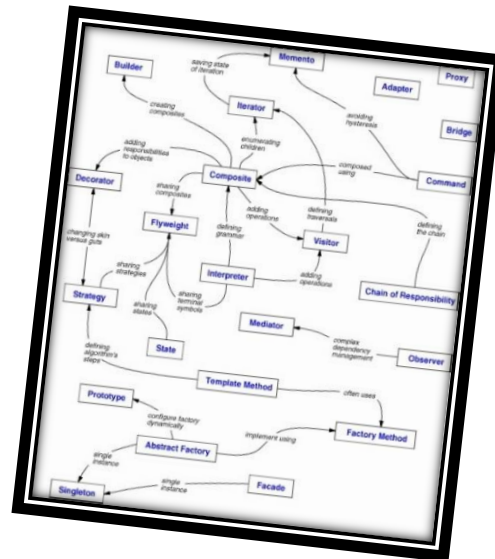
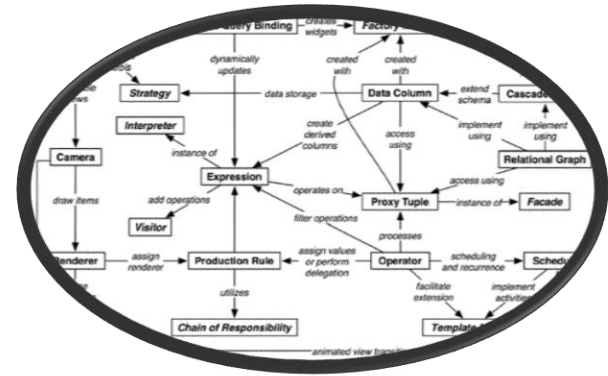


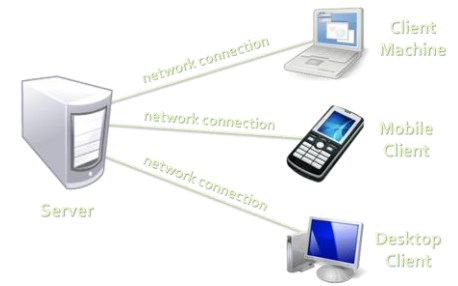
Architectuur



Inhoud

- Architectural patterns (systems design)
 - Client-Server,
 - 3-Tier / Multi-Tier
 - MVC (Model-View-Controller)
 - MVP (Model-View-Presenter)
 - MVVM (Model View ViewModel)
 - SOA (Service-Oriented Architecture) Patterns

Client-Server Architectuur

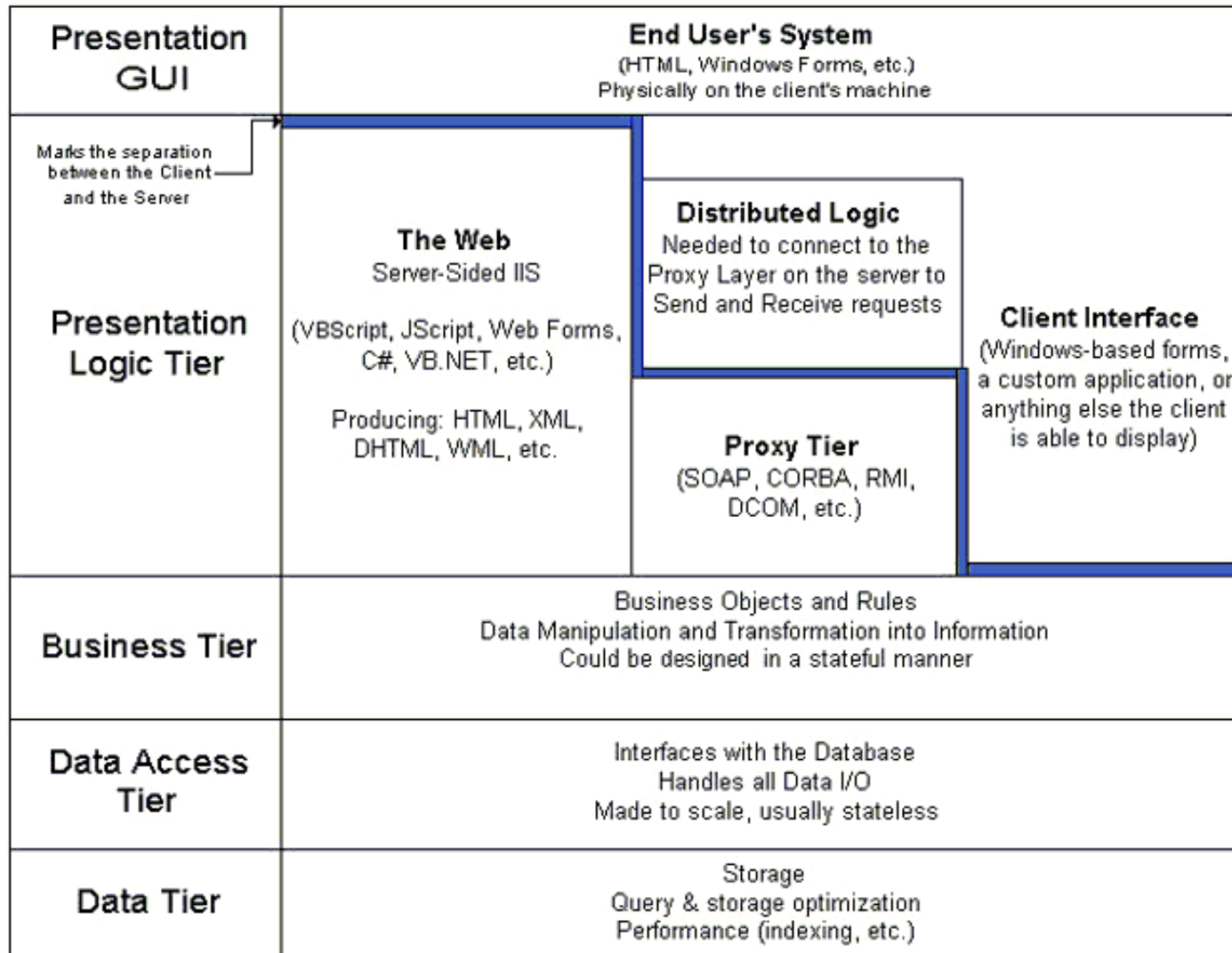


- Het client-server model bestaat uit:
 - **Server** – een applicatie (of machine) dat meerdere services aanbiedt aan meerdere clients
 - Kan op IIS -gebaseerde Web server zijn
 - Kan WCF service zijn
 - Of services in de cloud
 - **Clients** –software applicaties die UI (front-end) voorzien om de services van de server aan te spreken
 - Bv applicatie in WPF, HTML5, Silverlight, ASP.NET, ...

De 3-Lagen Architectuur

- De **3-tier architectuur** bestaat uit de volgende lagen (tiers, layers):
 - **Front-end** (client layer)
 - Client software – voorziet de UI van het system
 - **Middle tier** (business layer)
 - Server software – bevat de core system logic
 - Implementeert de business processes / services
 - **Back-end** (data layer)
 - Beheert de data van het systeem (database / cloud)

Multi-Tier Architecture



HTML

ASP.NET

WCF

ORM

DB

Model-View-ViewModel (MVVM)

- **Model-View-ViewModel (MVVM)** is een design pattern voor moderne UI apps
 - Wordt gebruikt in WPF en Silverlight
 - Is gebaseerd op **MVC**, **MVP** en Martin Fowler's **Presentation Model** pattern
 - Frameworks bestaan die deze patterns gebruiken (bv):
 - Prism (Composite Application Guidance for WPF and Silverlight)
 - MVVMLight
 - Calibri Micro
 -
 - Toestand en gedrag wordt gescheiden van elkaar

MVVM Structuur

- Model

- Beheer de application data / toestand van informatie
- Bv data access layer(zoals ADO.NET) of ORM framework(bv Entity Framework)

- View

- UI elementen van de applicatie (in WPF: xaml)
- Windows, forms, controls, fields, buttons, ...

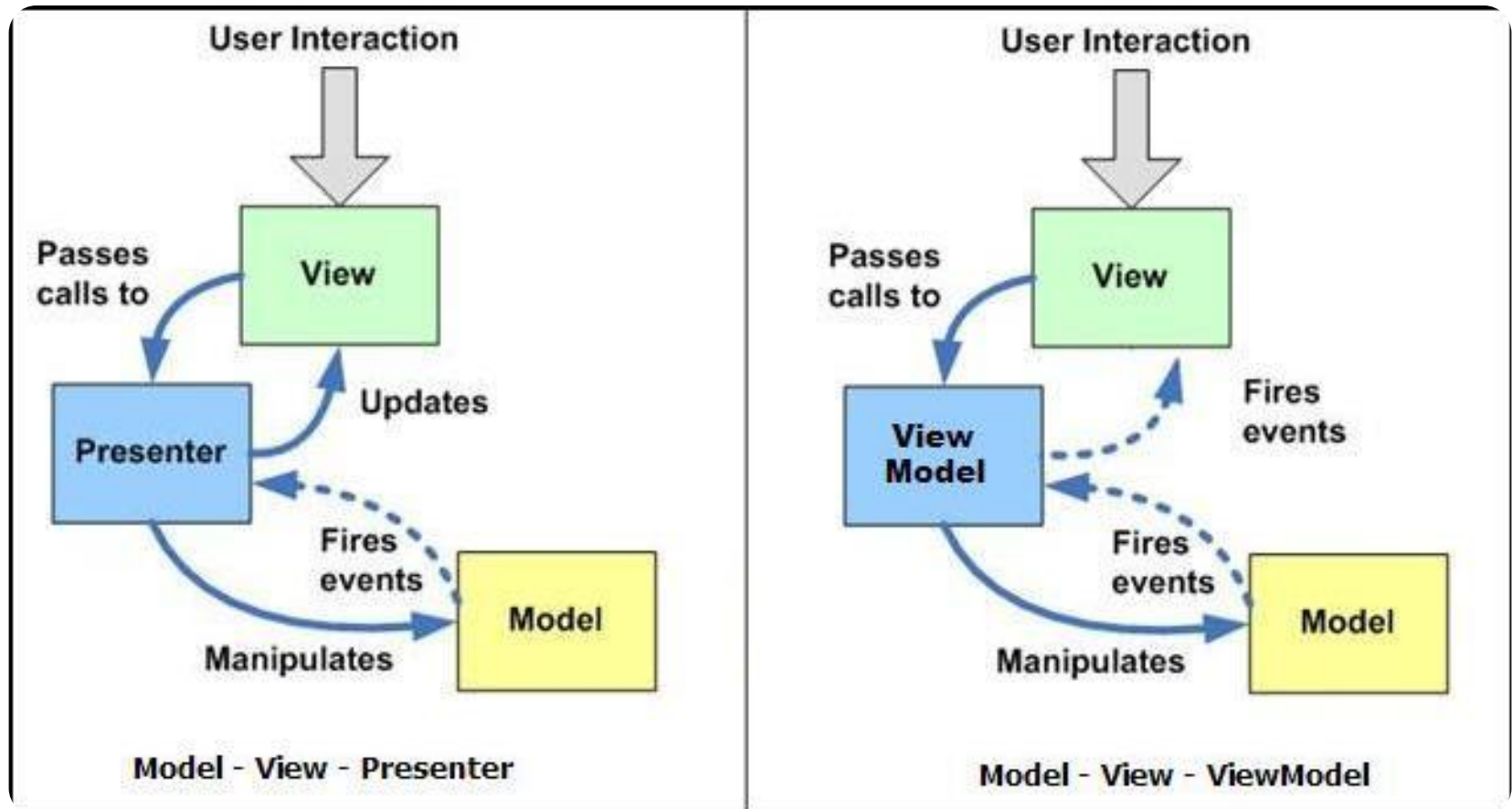
- ViewModel

- Data binder en converter dat informatie van het Model omzet naar View informatie
- Exposeert commands om te binden in de Views

Model-View-Presenter (MVP)

- **Model-View-Presenter (MVP)** is een UI design pattern gelijkaardig als MVC en MVVM
 - **Model**
 - Beheert de applicatie data (state)
 - **View**
 - Presentatie – Voorziet UI en handelt UI events af (keyboard, mouse, ...)
 - **Presenter**
 - Presentatie logica (zal data van het model voorbereiden om in een bepaald formaat te kunnen presenteren)

MVP versus MVVM Pattern

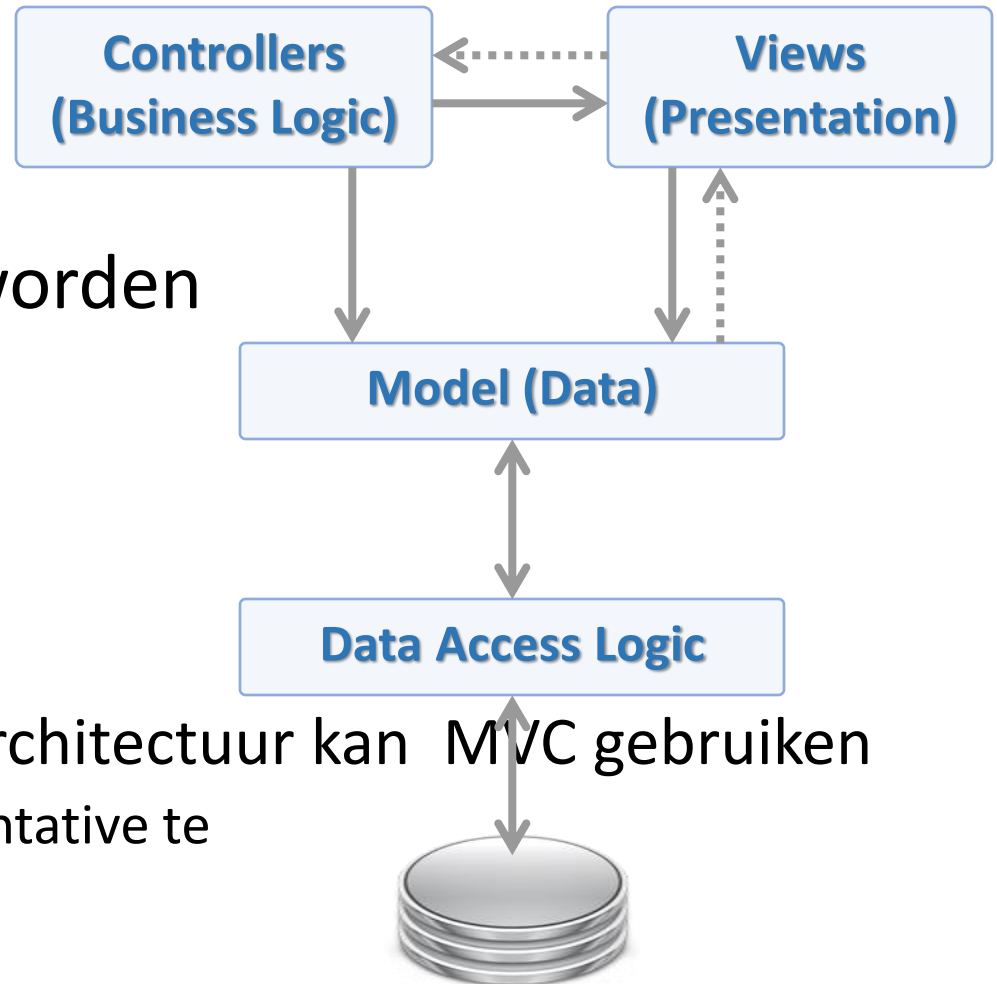


Model-View-Controller (MVC)

- **Model-View-Controller (MVC)** architectuur
 - De business logica wordt gescheiden van de applicatie data en presentatie
- **Model**
 - Beheert de applicatie data
- **View**
 - Toont UI aan gebruiker
- **Controller**
 - Beheert de interactie met de gebruiker

MVC en Multi-Tier Architectuur

- MVC vervangt *niet* de multi-tier architectuur



- Beide kunnen samen worden toegepast
- Een typische multi-tier architectuur kan MVC gebruiken
 - Om logica, data en presentatie te scheiden

SOA

- **Service-Oriented Architecture (SOA)** is een concept voor ontwikkeling van software systemen
 - Herbruikbare bouwstenen (componenten) die "services" worden genoemd
- Services in SOA zijn:
 - Autonoom, toestandsloze business functies
 - Accepteren 'requests' en geven 'responses' terug
 - Gebruiken een standaard interface

Referenties

- <https://www.c-sharpcorner.com/article/software-architecture-and-patterns/>
- C# Programming @ Telerik Academy
 - ◆ Telerik Software Academy Forums
 - ◆ forums.academy.telerik.com