# View

Det brugeren ser, kan enten være et helt vindue eller blot en user control.

# ViewModel

Binde ledet mellem View og model. Kan betragtes enten som del af GUI laget eller som Forretningslogikken. Ville i vores tilfælde være bl.a. have til opgave at omsætte GUI items, til Items og ListItems.

# Model

Kan tilgås på to måder, enten objekt orienteret eller data centreret. Ved den data centrerede tilgang, vil Model være en del af DAL laget. Ved den objekt orienteret tilgang vil Model være Forretningslogik laget.

# Fordele

* Udviklet af Microsoft til brug i WPF, Silverlight og Windows Phone/store.
* Lettere at teste.
* Giver klar adskillelse af grafik og funktionalitet via intet til minimal code behind.

# Ulemper

* Endnu en omgang refaktorering af koden.
* Mere kompliceret end MVC og MVP, og vil mene at vores projekt som det er kan forsvares som MVP

# Andre kommentarer

* Code behind behøver ikke at være tom, kommer an på hvilke måde man forbinder View og ViewModel, Dog skal man være klar over at det, der ligger i code behind ikke kan unit testes.

# Konklusion

Det der ikke er meget tid tilbage, ved jeg ikke om det er den ekstra omgang refaktorering værd. Vi kan eventuelt som Lars selv foreslog så vidt mulig implementere ny funktionalitet efter modellen og senere, hvis vi føler vi har tiden, få overført resten til MVVM.