

## 0.1 Arkitektur mønstre

Nedenstående liste er en opremsning af de arkitekturmønstre, der er anvendt i systemet. Under listen er de forskellige mønstre dokumenteret.

- Tre-lags model
- MVC (Model-View-Controller)
- ContentProvider
- Bound Service

### Tre-Lags model

Tre-lags modellen bruges i sammenhæng med separation-of-concerns, til at splitte et projekt op i præsentation, logik og data tilgang. Præsentations laget sørger for at håndtere alt visuelt, logik laget laver manipulationer på data, og data tilgangen sørger for at tilgå de eksterne data kilder og hente relevant data.

Se [http://en.wikipedia.org/wiki/Multitier\\_architecture](http://en.wikipedia.org/wiki/Multitier_architecture) for mere dokumentation.

### MVC

En afart af tre-lags modellen og bruges i sammenhæng med hjemmeside design i ASP.NET. Viewet præsenterer data for brugeren, hentet fra modellen. Brugeren har også mulighed for at tilgå controlleren igennem viewet og ændre data sat i model-laget. Efter kaldet, vil viewet blive opdateret med de nye ændringer.

Se <http://en.wikipedia.org/wiki/Model-view-controller> for mere dokumentation.

### ContentProvider

En androidspecifik metode til at tilgå data. Bruges i sammenhæng mellem data-deling, mellem applikationer. Ved at bruge forskellige URler, kan et kald til samme funktion, tilgå forskellige funktionaliteter.

Se <http://developer.android.com/reference/android/content/ContentProvider.html> for mere dokumentation.

## BoundService

Et androidspecifik interface til en client/server implementering. Når en service er bound, kan den kun tilgås i den givne applikation, i de klasser der binder til den. Efter service kaldet er færdiggjort, returneres der til den kaldende klasse, med den givne messagehandler.

Se <http://developer.android.com/guide/components/bound-services.html> for mere information.