

Bilag 11. CMS

CMS-løsningen er gennemgående baseret på egenudviklet kode. Her er en skitsering af opbygning:

Fundament

Løsningen er programmeret i programmeringssproget Php og vil kunne fungere på Php fra version 5.1.x. Databasen er Mysql 5.x. Den vil gøre brug af en Apache webserver til drift og vil af specielmoduler hertil bruge mod_rewrite.

Tekniske designkriterier

Systemets arkitektur er baseret på et model-controller-view pattern.

Der er lagt vægt på små effektive klasser og funktioner. De muligheder (abstrakte klasser, korrekt beskyttelse af variabler) programmeringssproget for struktureret kode vil være brugt. Minimering af database kald vil endvidere være en central del af systemdesignet for at give en hurtigere responstid.

Dataanvendelse

Der tages udgangspunkt i at så meget relevant metadata som muligt gemmes i TING. Det sker for at al data kan indekseres ud fra samme principper og for at simplificere søgningerne. Tilgangen til metadata i TING foregår via webservices. Al anden data, f.eks. controlldata, gemmes i pæne databasestrukturer i en systemets egen, lokale database.

Struktur i CMS

CMS'et giver adgang til tilføjelse, redigering og sletning af data og til redigering af sideindhold.

Data vil tilgås via lister hvorfra de enkelte poster som der skal redigeres kan udvælges fra.

Redigering vil ske gennem de gængse html input elementer samt en tinyMce richtext editor.

Elementerne vil være præsenteret på samme måde i alle interfaces så der er en ensartet brugerflade.

Rettighederne vil være bundet op til brugergrupper med forskellige grader af rettigheder.

Rettigheder tildeles til objekter i systemet.

Intranet

Intranettet giver backendbrugerne mulighed for intern kommunikation og filudveksling.

Cms opbygning

Cms'et er opbygget af en række forskellige moduler. Figur 1 viser sammenhængen mellem modulerne.

Nederst ligger selve databaselaget og kald til eksterne webservices, f.eks. TING. Over databaselaget ligger den grundlæggende cms funktionalitet og interfaces til resten af systemet.

Oven på det ligger rettighedsstyringen af de forskellige templates og databaser. De er knyttet tæt sammen med sikkerhedsgrupper og brugere. Rettighederne bliver brugt på templates, databaser/lister og temaer. Fælles for de tre indholdstyper er at de er delt op efter model/controller/view hvor der er en pæn adskillelse af data og visning.

Templates er de forskellige sidetyper som kan bruges til at vise indhold til brugerne. Sidetyperne kan både være nogle meget generelle tekst/billedvisninger og mere specialiseret sider til projektet.

Databaser/lister bruges til de ressourcer som skal bruges rundt omkring i systemet. Det kunne fx være sitets brugere eller nyheder.

Temaer fungerer stort set på samme måde som templates.

Ved siden af ligger selve menu og url styringen der sammen med de rettigheder som sitebrugerne har er bundet sammen med de 3 indholdstyper.

I toppen ligger url handler som modtager forespørgsler fra brugerne og sende dem videre ned i systemet.

Figur 1

