

## Bestillingsdokument for ORDBASE

Dokumentansvarlig: Frank Hofslundsengen  
Forfatter: Per Kristian Warvik og Bjørn Tore Årøy  
Dato oppretta: 30.05.17  
Arkivreferanse:  
Filnamn: OrdBase\_Bestillingsdokument  
Versjon: 0.1.0

### Endringsoversikt

Versjon	Dato	Beskrivelse av endring	Utført av
0.1.0	30.05.17	Dokument etablert	Per Kristian Warvik

### Vedlegg

Nummer		Namn

# Bestillingsdokument for OrdBasen

---

1.	Bakgrunn og behov.....	3
1.1	Målgruppe.....	3
1.2	Kommunikasjon.....	3
2.	Avgrensing og sluttkriterie.....	4
3.	Teknologiske føringar.....	4
4.	Systembeskrivelse .....	5
4.1	Språkserver.....	5
4.2	Klient .....	5
4.3	Klient-Cache .....	5
4.4	Admin-grensesnitt.....	6
4.4.1	REST-basert api-grensesnitt på server .....	6
4.4.2	Admin-grensesnitt på server .....	6
4.4.3	Registrere nye klientar .....	6
4.5	Kategoriar .....	6
5.	Systemkrav .....	7
6.	Utviklingsprosedyrer .....	7
6.1	Arkitektur .....	7
6.2	Namnekonvensjon – Naming Convention .....	7
7.	Tillegg viss det blir tid .....	7
8.	Prosjektets framdrift og aktiviteter .....	8
8.1	Hovudmilepælsplan .....	8
9.	Organisering .....	8
10.	Effekt og gevinstmål.....	8
11.	Kvalitetssikring .....	8

# Bestillingsdokument for OrdBasen

## 1. Bakgrunn og behov

Som statleg instans er vi ansvarleg for at sidene våre som er opne for publikum er fleirspråklege. Dei skal ha både Samisk, Nynorsk og Bokmål.

Tidlegare har vi i .NET-løysingar med fleire språk brukt RESX-filer med språkdata. Desse har sterk knytning til koden og krev ny kompilering og oppdatering av server i produksjon.

Språkproblematikken vart teken opp som ein viktig komponent i standardiseringsprosjektet blant Utviklingsgruppa ved IKT-seksjonen hos FMSF, våren 2017. Der vart det konkludert med at alle prosjekt og applikasjonar skal utformast og planleggast med omsyn til språk. Difor er det ynskeleg med ein gjenbrukbar modul som kan ta seg av dette. Språkmodulen skal forenkle implementering og vedlikehald av fleirspråklege tenester.

### 1.1 Målgruppe

Målgruppene for løysinga er utviklarar og forvaltning ved seksjonen.

### 1.2 Kommunikasjon

Veke	24	25	26	27	28	29	30	31	32
Sommarutviklar									
Per Kristian									
Bjørn Tore									
Robert									
Johnny									

Matrisa over syner kontaktpersonar under utviklingsperioden.

## 2. Avgrensing og sluttkriterie

Admin-grensesnittet skal ikkje ha noko rollebegrep i denne omgang, men være ei open løysing berekna for intranett.

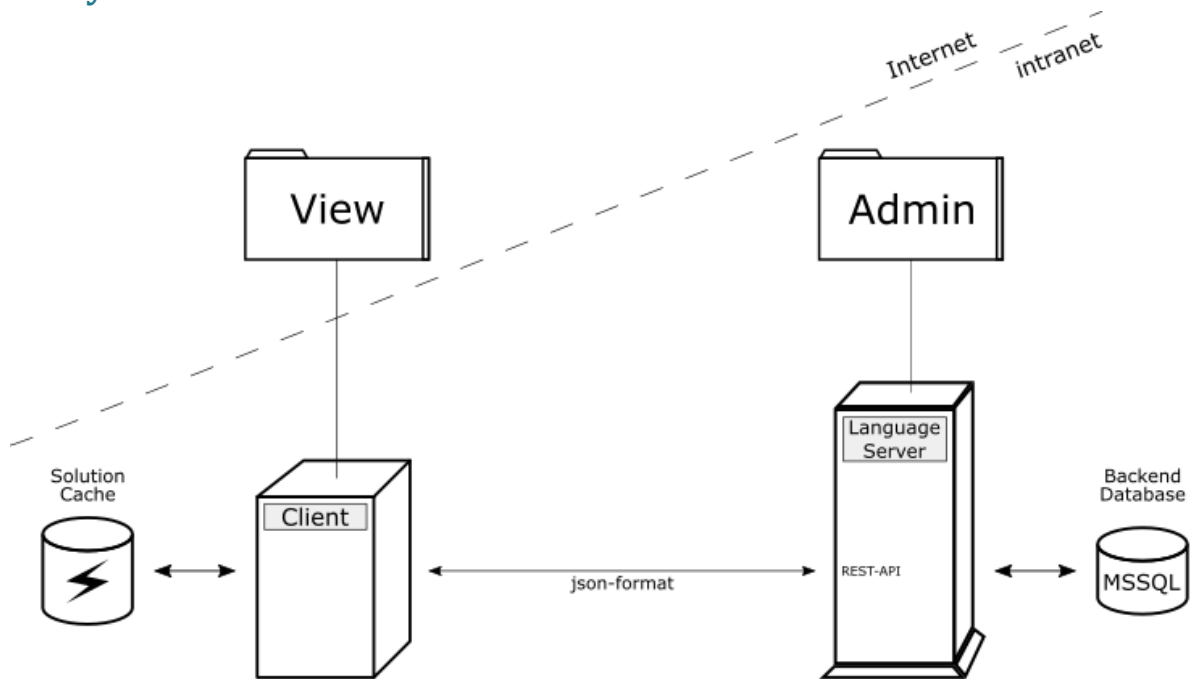
Ved slutten av prosjektet skal det foreligge:

- Programkode på GitHub.
- Fullverdig systemdokumentasjon.
  - Ta utgangspunkt i systemdokumentasjon frå tidlegare prosjekt.

## 3. Teknologiske føringar

- Visual Studio 2017
- ASP.NET Framework 4.7
- Entity Framework

## 4. Systembeskrivelse



### 4.1 Språkserver

Ei eiga teneste som skal halde på alle språkstrengar. Ein skal kunne registrere ulike løysingar som klientar. Desse skal hente tekst basert på ein nøkkel-identifikator og ein språk-identifikator. Kvar klient skal ha ein eigen nøkkel (GUID) slik at språkserveren kan skilje klientane frå kvarandre.

Språkserveren skal fungere som ei lokal intranett-teneste og skal ha ein eigen database i bakgrunnen.

### 4.2 Klient

Ein klient er alle dei systema som brukar (kan bruke) språkserveren, eksempelvis SYSAM, Trussamfunn, SepSkil, Nøkkeltall etc. Klienten skal kunne syne riktig tekst basert på valgt språk hos brukarar og nøkkel-identifikator gitt av programmarar.

Døme:

Brukar skal fylle ut skjema med eit felt 'Fornavn'.

Utviklarar vi skrive kode som `i18n.t('given_name')`.

For brukarar som nyttar nynorsk vil dei sjå «Fornamn».

For brukarar som nyttar bokmål vil dei sjå «Fornavn».

### 4.3 Klient-Cache

For å unngå unødig stor belastning på språkserveren er det ønskeleg med ein form for cache for kvar løysing. Den har følgjande mål:

1. Være raskare enn å spørje språkserveren kvar gang.
2. Hente inn endringar innan rimeleg tid (minst ein gong i døgnet).
3. Kunne flushast manuelt (oppdatere).
4. Ha levetid ut over ein enkelt HTTP-request.

## 4.4 Admin-grensesnitt

### 4.4.1 REST-basert api-grensesnitt på server

Webservice som gir ut strengar:

- Alle oversetjingar for eit system.
- Eit sett basert på nøkkel.
- Ein oversetjing basert på nøkkel.

### 4.4.2 Admin-grensesnitt på server

Ordpakkar må kunne oversetjast frå eit web-grensesnitt. Her er det viktig at forvaltning skal kunne endre språk i eit system, utan å automatisk få tilgang til andre system.

Controller, View og logikk for å endre språkverdiar i ei løysing. Eventuelt med rollebegrep.

Fordel:

1. Alle språkendringar kan gjerast i same løysing.
2. Utvikler kan bruke språkserveren aktivt under utvikling og slik gjere det lettare å lage nye nøklar.

Ulempe: Krever rollebegrep viss det er eit mål at ikkje alle kan endre alt, men berre dei løysingane dei skal få lov til.

### 4.4.3 Registrere nye klientar

Når eit nytt system vil nytte OrdBasen skal dette kunne registrerast i Admin-grensesnittet. Løysinga skal generere og gje ut ein nøkkel som den nye klienten skal nytte for å identifisere seg ved kvar request. Nøkkelen vil typisk bli nytta i AppSettings i web.config-fila på klient-prosjektet.

## 4.5 Kategoriar

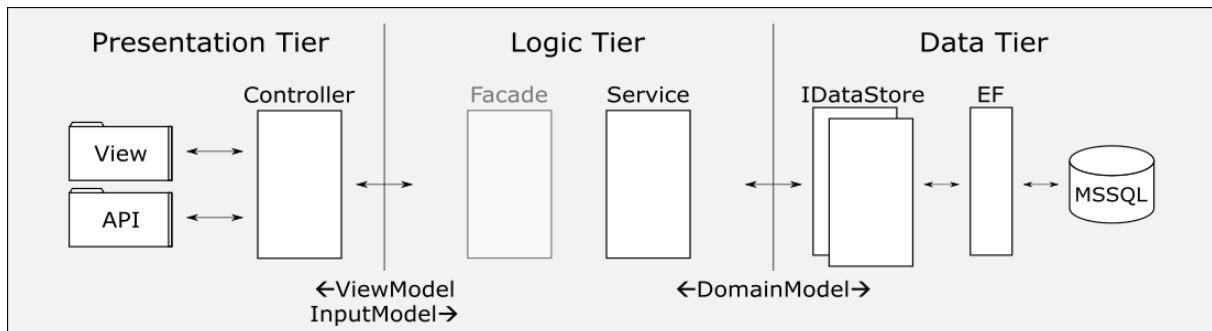
Språkstrengar bør kunne grupperast i grensesnittet for å gjere mengda tekst meir overkommeleg. Typisk tekstar som blir brukt på ei side. Kategoriane skal ikkje ha noko betydning for nøkkelen, kun for å gjere omsetjing lettare. Kategorien skal ha eit maskinlesbart namn for å hente ut strengane som eit sett. Til dømes «gje meg all tekst for 'case\_create'».

## 5. Systemkrav

1. Systemet skal ikkje bruke SESSION-variablar, men være STATELESS.
2. Systemet skal ha ei responstid på maks 50ms per språkstreng etter oppstart.

## 6. Utviklingsprosedyrer

### 6.1 Arkitektur



I .NET MVC blir generert HTML rekna som View, ein Controller-klasse blir rekna som Controller og Modellen er litt diffus, men kan reknast som resten av back-end-løysinga. Ein annan måte å dele inn arkitekturen på er slik teikninga over syner. Controller-klassen er ein tynn klasse som i hovudsak formidlar ViewModels til Views og til data-API-et, og tar imot InputModel som blir sendt til businesslogikken. Logikken skal utføre jobben med validering og transformering av data og andre applikasjonsrelaterte oppgåver. Ein Service er typisk ein klasse med metodar som løyser relaterte problemstillingar, til dømes validering av ein gitt InputModel. Viss det blir mange services som fungerer i lag, vil det være hensiktsmessig å gøyme kompleksiteten bak ein Facade-klasse. Når ein kjem til å korrespondere med databasen, skal alt skje gjennom ein implementasjon av eit interface. Implementasjonen skal bruke Entity Framework mot databasen, og EF skal kun brukast her.

### 6.2 Namnekonvensjon – Naming Convention

Namnsetjing av klassar: UpperCamelCase

Parameter og variablar: lowerCamelCase

Vi brukar ikkje prefixhint: LoginButton i staden for BtnLogin og 'var count' i staden for 'int iCount'.

All kode inkludert kommentarar skal være på engelsk (klarspråk).

Unngå forkortingar så langt det lar seg gjere.

Bruk meningsfulle namn.

Vis hensyn til dei som skal lese og forstå koden etterpå.

## 7. Tillegg viss det blir tid

**Redundans:**

Lagre språkfiler lokalt med jamne mellomrom.

## 8. Prosjektets framdrift og aktiviteter

### 8.1 Hovudmilepælsplan

Milepæl nr	Milepæl	Dato	Møtedato	Ansvarlig	Kriterier for oppnådd milepæl
1	Start prosjekt	12.06.17			
2	Start utvikling	15.05.17			
3	Gjennomgang av resultat og systemdokumentasjonen	03.08.17			
4	Systemdokumentasjon ferdig	10.08.17			
5	Avslutning	11.08.17			

For detaljert aktivitetsplan, sjå [vedlegg 2](#)

## 9. Organisering

Organisering	
Prosjektnavn	OrdBase
Oppdragsgiver	IKT-seksjonen
Prosjektansvarlig	Frank Hofslundsengen
Prosjektleder	FMSF v/Frank Hofslundsengen
Prosjektgruppe	Frank Hofslundsengen – FM Sogn og Fjordane - leiar Sommarstudent Per Kristian Warvik – FM Sogn og Fjordane Bjørn Tore Årøy – FM Sogn og Fjordane Robert Tellvik – FM Sogn og Fjordane Johnny Høyvik – FM Sogn og Fjordane

## 10. Effekt og gevinstmål

Beskriving av dei planlagde effekt og gevinstmål. Gevinst/mengde..

- Gjere språk lettare å implementere i andre løysingar
- Gjere det lettare å oppdatere tekstar på nettsidene våre
- Separation of Concerns: Få ei lausare knyting til språk i løysingane.
- Forvaltning blir betre rusta til å gjere større del av jobben enklare.

## 11. Kvalitetssikring

- Oppfølging undervegs (spørsmål, avklaringar og hjelp står til rådighet).



## Bestillingsdokument for OrdBasen

---

- Gjennomgang av prosjekt og systemdokumentasjon veke 29 og 31.
- Code review og kvalitetssikring av kode i etterkant.