

# IT-værktøjet ”Landbrugets klimaværktøj”

## - En beskrivelse af valgte teknikker og metoder

### Forfatter

*Landbrug & Fødevarer F.m.b.A., SEGES, Digital*

---

Opbygningen af databasen, servicen og brugergrænsefladen er beskrevet for Landbrugets klimaværktøj.

Programmet er udviklet i projekt 5397. Landbrugets klimaværktøj i perioden 1. august 2020 til 8. oktober 2021.

Den bagvedliggende kode er frit tilgængelig. Kontakt projektlederen, der så vil stå for koordinering af det praktiske i forbindelse med overførsel af kildekode.

Koden kan fås ved henvendelse til:

Hans Roust Thysen  
Klimachef  
Center for Klima & Bæredygtighed  
Landbrug & Fødevarer F.m.b.A.  
SEGES  
+45 3092 1701  
hrt@seges.dk

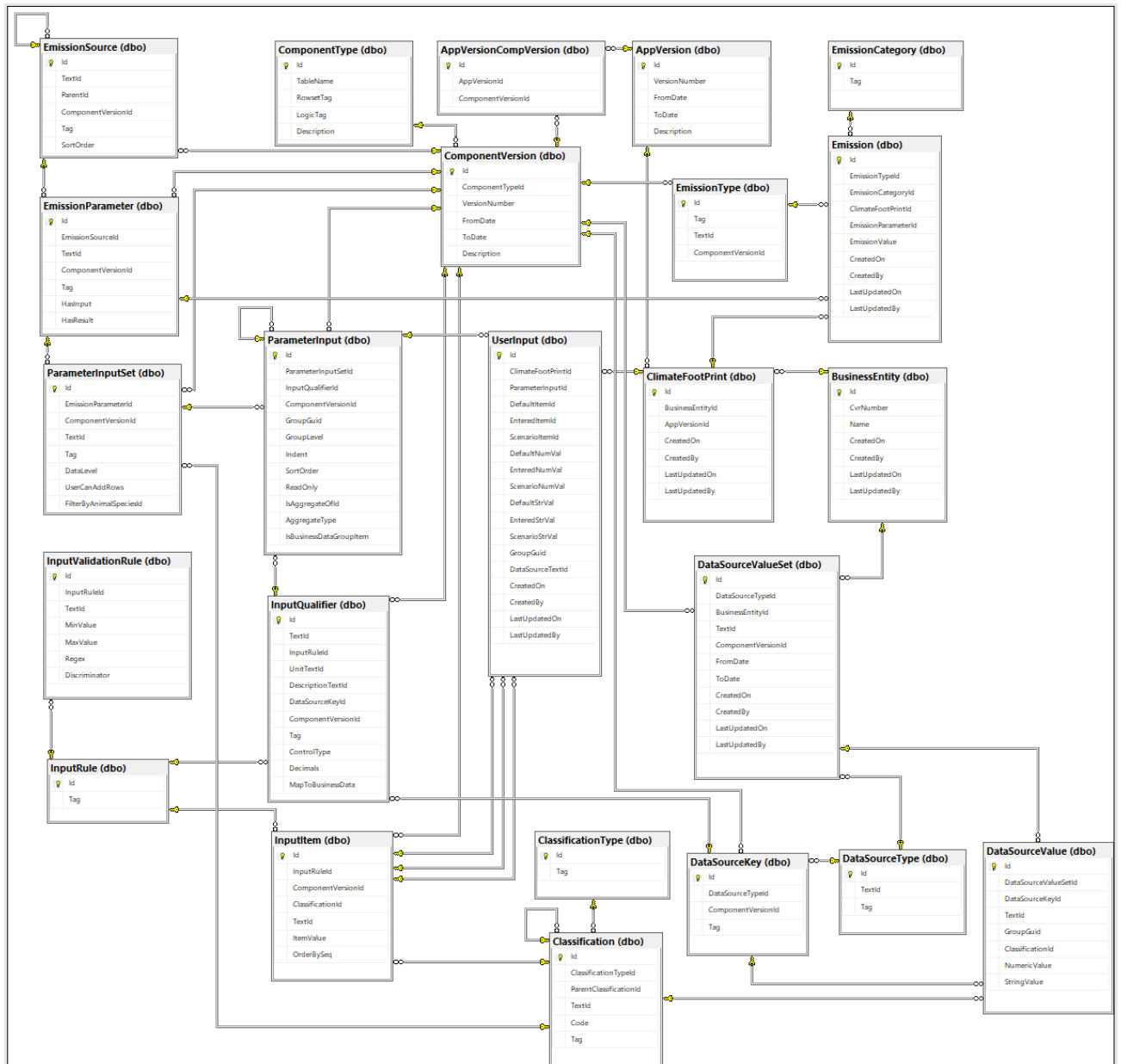
## Indholdsfortegnelse

Database .....	3
Datamodel.....	3
Beskrivelse af datamodel.....	4
Elementer til sikring af dataversionering.....	7
Database.....	7
Servicelag .....	7
Indlæsning af data fra gødningsregnskaber .....	8
Service.....	9
Brugergrænseflade .....	12
Introduktion .....	12
Data.....	12
Bedriftsaftryk .....	13
Produktaftryk .....	15
Teknologi.....	16
Modulopdeling .....	16

# Database

Databasen i Landbrugets klimaværktøj baserer sig på en Microsoft SQL Server 2019 database. Den kan hostes on-premise eller i Microsoft Azure cloud.

## Datamodel



Figur 1. Datamodel

## Beskrivelse af datamodel

Område	Tablet	Beskrivelse
<b>Emissionskilder</b>	EmissionSource	Emissionskilder i en hierarkisk struktur, f.eks.: <ul style="list-style-type: none"> <li>• Mark               <ul style="list-style-type: none"> <li>• Handelsgødning</li> <li>• Husdyrgødning på mark</li> </ul> </li> <li>• Stald               <ul style="list-style-type: none"> <li>• Fordøjelse</li> <li>• Malkekøer</li> <li>• Opdræt</li> </ul> </li> <li>• Husdyrgødning stald               <ul style="list-style-type: none"> <li>• Malkekøer</li> <li>• Opdræt</li> </ul> </li> </ul>
	EmissionParameter	En eller flere emissionsparametre pr. emissionskilde, f.eks. <ul style="list-style-type: none"> <li>• Fordøjelse</li> <li>• Malkekøer</li> <li>• Opdræt</li> </ul>
<b>Konfiguration af input</b>	ParameterInputSet	Definition af et sæt af input gældende for et givet dataniveau for en given emissionsparameter.
	ParameterInput	Definition af hvilke input der er en del af et input sæt.  Eksempelvis kan input sættet for emissionsparameteren "Udbragt handelsgødning" omfatte følgende inputs: <ul style="list-style-type: none"> <li>• Kg N</li> <li>• Kg P</li> <li>• Kg K</li> </ul> Der er mulighed for at definere, at flere inputs hænger sammen og dermed kvalificerer hinanden via feltet "GroupGuid", f.eks. Kg N i en given husdyrgødningstype.  Feltet "GroupLevel" kan bruges til at definere et hierarki mellem inputs.
	InputQualifier	Definition af typen af input for et parameter input. Numerisk input.  Valgbare input, Valgboks, Checkboks, Radioknapper. Kobling til defaultværdi fra en given datakilde, f.eks. gødningsregnskab.

Område	Tablet	Beskrivelse
		Fælles input på tværs af emissionsparametre, f.eks. "Antal ha".
	InputItem	<p>Definition af valgbare elementer til brugergrænsefladen.</p> <p>Gruppering af en eller flere elementer i en inputregel, der kan knytte sig til en inputtype, f.eks. husdyrgødningstyper eller afgrøder.</p> <p>Elementer kan kobles til en defaultværdi fra en given datakilde, f.eks. en bestemt husdyrgødningstype i gødningsregnskabet via fælles reference til en række i "Classification" tabellen.</p>
<b>Bedrift</b>	BusinessEntity	Bedrifter identificeret ved et CVR nummer.
<b>Bedriftsdata / brugerinput</b>	UserInput	<p>Standardværdi, fra f.eks. gødningsregnskab eller anden datakilde.</p> <p>Tilpasning af standardværdi, indtastet eller valgt af brugeren.</p> <p>Scenarielværdi indtastet eller valgt af brugeren.</p>
	ClimateFootPrint	Klimaaftryksberegning for en given periode og bedrift. Samler input og resultater.
<b>Beregning / resultat</b>	EmissionCategory	<p>Gruppering af emissionsresultater:</p> <ul style="list-style-type: none"> <li>• Standard, resultat af standard/indlæste data</li> <li>• Adjusted, resultat af brugertilpasning af standard/indlæste data</li> <li>• Scenario, resultat af brugerindtastede mål</li> </ul>
	EmissionType	<p>Type af emissioner:</p> <ul style="list-style-type: none"> <li>• CO2e</li> <li>• CO2</li> <li>• CH4</li> <li>• N2O</li> <li>• NOx</li> <li>• NH3</li> </ul>
	Emission	Emissionsresultat pr. kategori og type af emission pr. klimaaftryksberegning for en bedrift.
<b>Datakilder</b>	DataSourceType	Typen af datakilde, f.eks. gødningsregnskab, normer eller faktorer.
	DataSourceKey	Definition af datanøgler.
	DataSourceValueSet	Definition af datasæt for en given type af datakilde og enten på generelt- eller på bedriftsniveau.

Område	Tablet	Beskrivelse
	DataSourceValue	Nøgle og tilhørende værdi for et givent datasæt. Et sæt af nøgler med tilhørende værdier kan defineres som en "række" af data via feltet "GroupGuid".  En nøgle kan knytte sig enten til en numerisk værdi, en tekststreng eller en klassifikation (en række i tabellen "Classification"), f.eks. en husdyrgødningstype.
	ClassificationType	Type af klassifikationer, f.eks. husdyrgødningstype.
	Classification	Elementer i den enkelte klassifikationstype, f.eks. er kvæggylle, svinegylle og dybstrøelse nogle af elementerne i husdyrgødningstypen.
Tekster	Text	Definition af nøgle for den enkelte datanære tekst for f.eks. emissionskilder, -parametre, inputtyper, datakilder osv.
	TranslatedText	Definition af en tekst pr. sprog for den enkelte tekstnøgle. P.t. er der kun defineret danske tekster.
Versionering af data	ComponentType	Definerer komponenttyper i systemet. En komponenttype kan være alle rækker i en tabel eller rækker i en tabel, der opfylder bestemte kriterier (rowset) eller en logikklasse i servicelaget.  For hvert element i systemet, hvor der skal etableres sporbarhed, oprettes elementet som en ComponentType.  Eksempelvis kan logik til beregning af emission af "udbragt handelsgødning" eller datakilden "Gødningsregnskab" betragtes som komponenttyper.
	ComponentVersion	Der etableres en relation til ComponentVersion fra de tabeller der skal opfattes som komponenter eller hvor et sæt af rækker i tabellen udgør en komponent.  Som udgangspunkt oprettes en række pr. komponenttype med "ToDate" sat til null, hvilket indikerer en aktiv komponent version.  Ved ændring af en komponent tilføjes en ny række for komponenten med en ny "FromDate". "ToDate" i den tidligere version sættes til samme dato som den nye versions "FromDate".
	AppVersion	Definerer overordnet version af applikationen (app), dvs. beregningslogik i servicelaget og data i databasen. App-

Område	Tabel	Beskrivelse
		<p>versionen ændrer sig, hvis en eller flere komponenter skifter version.</p> <p>Der etableres en relation til AppVersion fra tabeller med data, der er et output af flere komponenter i sammenspil, f.eks. tabellen "ClimateFootPrint".</p> <p>Som udgangspunkt oprettes en række med "ToDate" sat til null, hvilket indikerer en aktiv app-version.</p> <p>Ved ændring af app-versionen tilføjes en ny række med en ny "FromDate". "ToDate" i den tidligere app-version sættes til samme dato som den nye app-versions "FromDate".</p>
	AppVersionCompVersion	<p>Definerer komponentversioner indeholdt i en app-version.</p> <p>Som udgangspunkt oprettes en række pr. aktive komponentversion for den aktive app-version.</p>

Tabel 1. Beskrivelse af datamodel

## Elementer til sikring af dataversionering

### Database

MS Sql featuren "Row Level Security" (RLS) anvendes til at sikre, at der ved forespørgsel på tabeller med versionerede data, kun returneres data for den app-version servicelaget har specificeret via en session context variabel.

Denne implementering af RLS omfatter følgende objekter i databasen:

- Session context variabel - "AppVersionId"
- View - "AppVersionView"
- Predicate function - "ComponentVersionPredicate". Benytter "AppVersionView" til at afgøre om ComponentVersionId på en given række findes i specificeret app-version. Funktionen kaldes pr. række
- Security policy - "ComponentVersionPolicy". Denne policy definerer, hvilke tabeller predicate funktionen skal kaldes for i forbindelse med select på tabellerne

### Servicelag

Servicelaget har ansvar for at fortælle databasen, hvilken version af data den er interesseret i at få retur. Det sker ved at sætte session context variabelen "AppVersionId". Det gøres, hver gang en forbindelse åbnes til databasen, og inden et entity select afvikles.

Dette ansvar varetages af klassen "AppversionIdProvider", som er en nedrivning af klassen "DbConnectionInterceptor". Interceptor klassen bringes i spil ved at tilføje den til EF konfigurationen i "BaseContext" klassen.

"BaseContext" er tilføjet egenskaben "AppVersionId", der kan sættes i en repository klasse i det tilfælde, hvor der skal hentes data for en specifik version og ikke seneste version.

I repository klasser, hvor der caches versioneret data, skal cache key tilføjes app-version, således der opbygges en cache for hver af de versioner af data, der er blevet hentet.

## Indlæsning af data fra gødningsregnskaber

Til beregning af klimaaftryk for en bedrift i Landbrugets klimaværktøj, er der brug for bedriftsspecifikke data vedr.:

- Udbragt kvælstof i handelsgødning
- Udbragt kvælstof i forskellige typer af husdyrgødning/ anden organisk gødning
- Oplysninger om marker, herunder areal, jordbundstype, primærafgrøde og evt. efterafgrøde
- Oplysninger om evt. husdyrproduktion på bedriften, herunder dyretyper (kvæg, svin eller fjerkræ), antal og staldsystemer fordelt på ejendomsnummer (CHR).

I projektet er der anvendt bruger-indberettede og af myndighederne godkendte data for gødningsregnskabet 2019/2020. Gødningsregnskabsdata er leveret af myndighederne i forskellige Excel regneark.

Data er indlæst på forhånd til databasen for Landbrugets klimaværktøj via sql scripts. Data for de enkelte regneark omformes til sql statements, som er fastholdt i et sql script pr. regneark.

Det enkelte sql statement er et kald til en procedure i databasen med en række argumenter. Argumentværdierne stammer fra forskellige kolonner i en række fra et af regnearkene.

I regnearket er der defineret et formeludtryk, der kan sammensætte en streng med værdier fra en eller flere kolonner, der af et sql script kan opfattes som et sql statement. I de enkelte scripts er formeludtrykket og anvendte kolonner fastholdt.

Eksempel på indlæsning af udbragt kvælstof i handels- og husdyrgødning for en given bedrift.

Regneark "Gødningsregnskab2020.xlsx":

	AG	AH	AI	AX	DZ	EB	EE	EG	EI	EL	EN	EQ	ES	EU	FG	FH	GU	GV	GW	GX	GY
1 CVR																					
2 VIR NAVN	159.06	153.06	153.07	0	0	16581	0	0	5122	22023	0	0	0	0	13438.7	858.1					

Sql script "012\_DataSeed\_FertilizerSettlements.sql":

```
DECLARE @FromDate datetime = '2021-09-16';
DECLARE @compTypeId int;
DECLARE @compVersionId int;
DECLARE @dsvsFromDate datetime = '2019-08-01';
DECLARE @dsvsToDate datetime = '2020-07-31';

EXEC @compTypeId = AddComponentType 'DataSourceValueSet', 'FertilizerSettlementValueSet', NULL, 'FertilizerSettlements for a given year are versioned together'
EXEC @compVersionId = AddComponentVersion @compTypeId, 1, @FromDate, NULL, NULL
EXEC AddText 'FertilizerSettlement2019-2020', 'da', 'Gødningsregnskab 19/20'

EXEC AddFertilizerSettlementData 'FertilizerSettlement2019-2020', @compVersionId, @dsvsFromDate, @dsvsToDate, 159.06, 153.07, NULL, 22023, 13438.7, 858.1, NULL, 16581, NULL, NULL, 5122,
```

Figur 2. Eksempel på indlæsning af udbragt kvælstof i handels- og husdyrgødning



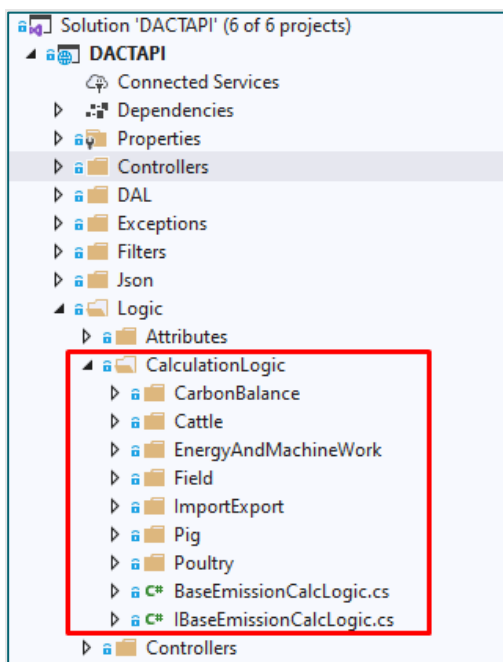
SQL proceduren, i ovennævnte eksempel "AddFertilizerSettlementData", har ansvar for at gemme argumentværdierne i de relevante tabeller i datamodellen for Landbrugets klimaværktøj.

## Service

Servicen i Landbrugets klimaværktøj er udviklet i C# og SQL og er baseret på Microsoft .NET

I servicen er der en gruppe af C# klasser med logik, som har til ansvar at hente generelle og bedriftsspecifikke data fra databasen og omdanne disse til det format, som hver beregningsformel har behov for.

Disse klasser er markeret herunder:



Figur 3. Klasser til at hente generelle og bedriftsspecifikke data

Et eksempel på en logikklasser:

```
namespace DACTAPI.Logic.CalculationLogic.Field
{
    /// <summary>
    /// Nitratudvaskning (LK-9/LK-37)
    /// </summary>
    [EmissionParameter("NitrateLeaching", 1)]
    public class NitrateLeachingLogic : BaseEmissionCalcLogic
    {
        protected override IFormulaBase PrepareCalculation(List<FormulaUserInput> userInputs, List<DataSourceValueView> dataSourceValues, List<InputRule> inputRules, EmissionCategoryEnum category)
        {
            var formula = ResolveFormula<IFormulaNitrateLeaching>();
            var inputItemDict = inputRules.Where(x => x.Tag == "Crop" || x.Tag == "CoverCrop").SelectMany(x => x.InputItemViews).ToDictionary(x => x.Id, x => x);
            var cropTypeInputs = userInputs.Where(x => x.InputQualifierTag == "FieldCropType").SelectMany(y => y.UserInputs).ToList();
            var coverCropTypeInputs = userInputs.Where(x => x.InputQualifierTag == "FieldCoverCropType").SelectMany(y => y.UserInputs).ToList();
            var fieldAreaHaInputs = userInputs.Where(x => x.InputQualifierTag == "FieldAreaHa").SelectMany(y => y.UserInputs).ToList();

            var cropAreas = new List<TypeValue>();

            // Add crop area of primary crops
            cropAreas.AddRange(GetTypeValues(cropTypeInputs, fieldAreaHaInputs, inputItemDict, category));

            // Add crop area of secondary crops
            cropAreas.AddRange(GetTypeValues(coverCropTypeInputs, fieldAreaHaInputs, inputItemDict, category));

            if (cropAreas.Any())
            {
                formula.CropAreas = cropAreas;
                formula.EmissionFactors = GetEmissionFactors(dataSourceValues, "EFNitrateLeaching");
                formula.CFN20toCO2e = Convert.ToInt32(GetDataSourceValue(dataSourceValues, "CFN20toCO2e"));
            }

            return formula;
        }
    }
}
```

Figur 4. Eksempel på logikklasser

Herover ses det, at denne logikklasser er knyttet til "Nitratudvaskning", og at generelle data for hoved-afgrøder og efterafgrøder samt beregningsfaktorer hentes og leveres til en formel, sammen med driftsspecifikke data om alle marker (afgrøder, areal, jordbundstype m.m.)

Alle formler er samlet i en isoleret C# projekt, som gør det muligt at anvende beregningerne uafhængigt af resten af løsningens programdele.

Et eksempel på en formel:

```
/// <summary>
/// Nitratudvaskning (LK-9/LK-37)
/// </summary>
public class FormulaNitrateLeaching : FormulaBase, IFormulaNitrateLeaching
{
    public List<TypeValue> CropAreas { get; set; }

    public List<EmissionFactor> EmissionFactors { get; set; }

    public int CFN20ToCO2e { get; set; }

    public override List<EmissionResult> Calculate()
    {
        if (CropAreas == null || !CropAreas.Any())
        {
            return null;
        }

        double N20Total = 0;

        List<TypeValue> summarizedCropAreas = SummarizeCropsArea();

        foreach (var cropArea in summarizedCropAreas)
        {
            var KgNPerHa = EmissionFactors.FirstOrDefault(ef => ef.ClassificationId == cropArea.TypeId)?.NumericValue.GetValueOrDefault();
            N20Total += KgNPerHa.GetValueOrDefault() * 0.0075 * 44 / 28 * cropArea.Value.GetValueOrDefault();
        }

        var CO2e = new EmissionResult() { Value = (N20Total * CFN20ToCO2e).RoundedToInt64(), Type = EmissionTypeEnum.CO2e };
        var N20 = new EmissionResult() { Value = N20Total.RoundedToInt64(), Type = EmissionTypeEnum.N20 };

        return new List<EmissionResult>() { CO2e, N20 };
    }

    protected List<TypeValue> SummarizeCropsArea()
    {
        return CropAreas.GroupBy(a => a.TypeId).Select(b =>
        {
            return new TypeValue()
            {
                Tag = b.FirstOrDefault()?.Tag,
                TypeId = b.FirstOrDefault()?.TypeId,
                Value = b.Sum(c => c.Value)
            };
        }).ToList();
    }
}
```

Figur 5. Eksempel på formel

Denne klasse udfører selve beregningerne i forbindelse med "Nitratudvaskning" ud fra konkrete oplysninger om markarealer og afgrøder. Resultatet af disse beregninger er emissioner som fordeles på typerne fra "EmissionType" tabellen, og regnes én gang pr. kategori fra "EmissionCategory" tabellen.

# Brugergrænseflade

## Introduktion

Brugergrænsefladen til Landbrugets klimaværktøj er udviklet som en webapplikation, kompatibel med de fleste moderne browsere og optimeret til desktopbrug.

Fra menuen kan brugeren tilgå applikationens tre primære sider:

## Data

Applikationens startside, hvorfra brugeren kan se de data, der er indlæst fra gødningsregnskabet (A) og lave eventuelle tilpasninger (B). Værdier indtastet under tilpasninger gemmes, når input feltet forlades. Øverst på siden er det muligt at se det samlede klimaaftryk for bedriften (C) og genberegne dette efter tilpasninger (D).

The screenshot shows the 'Landbrugets Klimaværktøj' interface. At the top, there is a navigation menu with 'MENU' and a gear icon. Below the menu, a summary bar displays 'Samlet aftryk C' and 'Standardtal og overført: 7.250,7 ton CO<sub>2</sub>e - Med tilpasninger: 7.080,2 ton CO<sub>2</sub>e D'. The main content area is titled 'Husdyrgødning stald' and is divided into two columns: 'Standardtal og overført Gødningsregnskab 19/20' and 'Tilpasninger 2020'. Under 'Standardtal og overført', there are two entries for 'CHR 1 Producerede slagtekyllinger'. The first entry shows 'Produktionstid 40 dage (levende vægt ved slagtning 2,6 kg)' with a value of '1.000 stk' and a 'Gødningsregnskab 19/20' label. The second entry shows 'Produktionstid 45 dage (levende vægt ved slagtning 2,77 kg)' with a value of '1.100 stk' and a 'Gødningsregnskab 19/20' label. Under 'Tilpasninger 2020', there are two rows of settings. The first row is 'Varmeveksler i stalden' with a 'Standard værdi' of 'Nej' and radio buttons for 'Ja' (selected) and 'Nej'. The second row is 'Direkte udkørsel af dybstrøelse' with a 'Standard værdi' of '15%' and radio buttons for '25%' (selected) and '15%'. At the bottom, there is a 'Tilføj staldtype' dropdown menu.

Figur 6. Eksempel på startside

## Bedriftsaftryk

Når data fra gødningsregnskabet er tjekket igennem og eventuelle tilpasninger udført, giver bedriftsaftrykssiden et mere detaljeret indblik i, hvor meget hver emissionskilde bidrager til det samlede klimaaftryk. Dette kan desuden ses fordelt på klimagasser (A), filtreret uden import/eksport (B) og uden kulstofbalance (C). Fra denne side kan brugeren desuden lave en scenarieberegning pr. emissionskilde (D) og udskrive sidens indhold i printvenligt format (E).

≡ MENU Landbrugets Klimaværktøj
⚙️

2020

B  
 Vis bedriftens aftryk uden energi og import/eksport

A  
 Vis fordeling på gasser

C  
 Vis bedriftens aftryk uden kulstofbalance

UDSKRIV
E

Klimaaftryk med energi, import og eksport	Standard / overført <small>ton CO<sub>2</sub>e</small>	Bedriftsaftryk <small>ton CO<sub>2</sub>e</small>	Fremtidsscenerier <small>ton CO<sub>2</sub>e (ændringer)</small>
Samlet udledning for bedriften	7.245,6	7.075,5	7.075,5 - <span style="color: orange; font-weight: bold; font-size: 1.2em;">D</span>
Mark	841	671 <span style="font-size: 0.8em;">✎ Data</span>	671 - <span style="font-size: 0.8em;">✎ Scenerier</span>
Husdyrgødning mark	178	8	8 -
Anden organisk gødning mark	1	1	1 -
Organogene jorde	511	511	511 -
Kalkning	18	18	18 -
Nitratudvaskning	30	30	30 -
Afgørderester	102	102	102 -
Mark, kulstofbalance <span style="font-size: 0.8em;">i</span>	20	20	20 -
Afgøder med kulstofopbygning	-153	-153	-153 -
Afgøder med kulstof nedbrydning	185	185	185 -
Kulstofopbygning organisk gødning	-12	-12	-12 - <span style="color: orange; font-weight: bold; font-size: 1.2em;">D</span>
Svin	2.351	2.351 <span style="font-size: 0.8em;">✎ Data</span>	2.351 - <span style="font-size: 0.8em;">✎ Scenerier</span>

Figur 7. Eksempel på bedriftsaftryk

## Fremtidsscenarier

Fremtidsscenarier åbner som en skuffe på bedriftsaftrykssiden og fungerer ligesom datasiden, men med bedriftens aktuelle værdier i standardkolonnen (A) og scenarieværdier i kolonnen med samme navn (B). Værdierne i denne kolonne kan nulstilles ved at trykke på knappen 'Nulstil' (C). Når skuffen lukkes (D), genberegnes det samlede klimaaftryk for fremtidsscenarier, hvis der er foretaget ændringer i scenariekolonnen.

**Landbrugets Klimaværktøj**

LUK OG GENBEREGN

### Fremtidsscenarier - Fjerkræ

**A** **B** **C** **NULSTIL**

**Husdyrgødning stald**  
Bedriftsaftryk: Standard, overført og tilpasset | Scenarier

**CHR 1 Producerede slagtekyllinger**

Parameter	A (Standard)	B (Scenarier)
Produktionstid 40 dage (levende vægt ved slagtning 2,6 kg)	1.000 stk <small>Standard, overført og tilpasset</small>	900 stk
Varmeveksler i stalden	Nej <small>Standard, overført og tilpasset</small>	<input type="radio"/> Ja <input checked="" type="radio"/> Nej
Direkte udkørsel af dybstrøelse	15% <small>Standard, overført og tilpasset</small>	<input type="radio"/> 25% <input checked="" type="radio"/> 15%
Produktionstid 45 dage (levende vægt ved slagtning 2,77 kg)	1.100 stk <small>Standard, overført og tilpasset</small>	stk
Tilføj staldtype		

**CHR 1 Årshøner**

Parameter	A (Standard)	B (Scenarier)
Friland, konsumæg, gulvdrift + fler-etagesystem med gødningsbånd	1.200 stk <small>Standard, overført og tilpasset</small>	stk
Konsum, gulvdrift, produktionstid 119 dage	1.400 stk <small>Standard, overført og tilpasset</small>	stk

Figur 8. Eksempel på fremtidsscenariesiden

## Produktaftryk

Herfra kan brugeren hente Excelark for sin driftsgren og beregne fordelingen af klimaaftrykket på producerede produkter.

≡ MENU Landbrugets Klimaværktøj

### Produktaftryk

**Produktregnskab Mark**  
Der er mulighed for at beregne klimaaftryk pr. produkt for planteprodukter ved at klikke [her](#)

**Produktregnskab Kvæg**  
Der er mulighed for at beregne klimaaftryk for mælk og kød ved at klikke [her](#)

**Produktregnskab Svin**  
Der er mulighed for at beregne klimaaftryk for kød ved at klikke [her](#)

**Produktregnskab Fjerkræ**  
Der er mulighed for at beregne klimaaftryk for kød og æg ved at klikke [her](#)

Brug for hjælp?  
+45 7015 5015  
kundecenter@seges.dk

Figur 9. Eksempel på produktaftryksiden

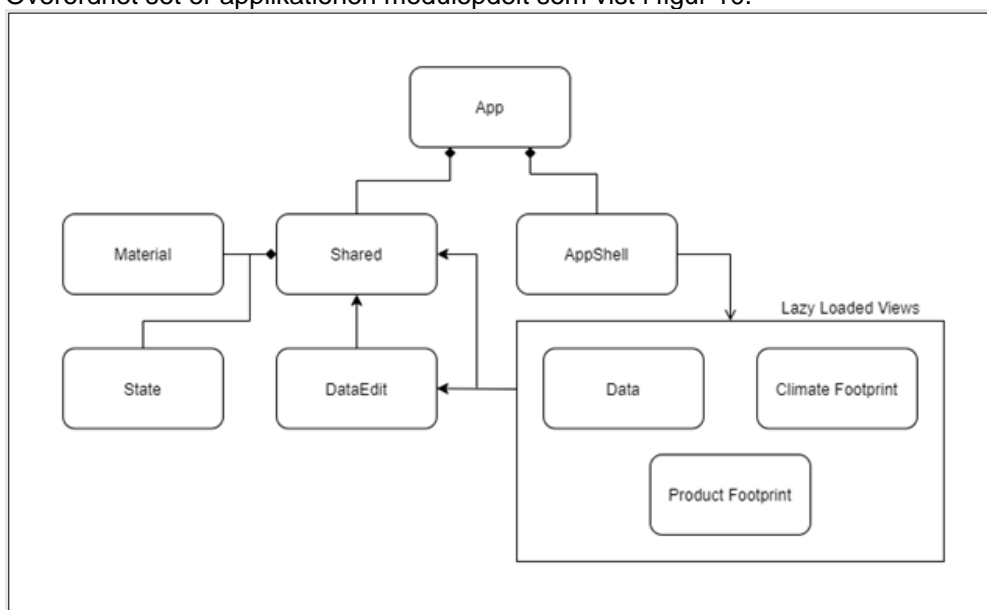
## Teknologi

Brugergrænsefladen til Landbrugets klimaværktøj er udviklet som en PWA<sup>1</sup> med frameworket Angular<sup>2</sup> og kan installeres fra browseren, hvorved applikationsfilerne caches på enheden og opstartstiden mindskes. Login foregår via SEGES' oidc provider "AgroID", hvorfra brugerens informationer, herunder cvr nummer, udlæses for at afgøre, hvilken bedrift der skal indlæses data for.

Indholdet i applikationen konfigureres af servicelaget for at vise brugeren de inputs på datasiden og emissioner på aftrykssiden, som er relevante for bedriften.

## Modulopdeling

Overordnet set er applikationen modulopdelt som vist i figur 10.



Figur 10. Skitser over moduler i applikationen

### **AppModule**

Rodmodulet 'AppModule' står for initialisering af applikation, registrering af serviceworker, generel fejlhåndtering og inkluderer øvrige moduler, som udgør den samlede funktionalitet.

### **AppShellModule**

Definerer applikationens header, footer og et content-område, hvori de lazy loadede moduler "Data", "ClimateFootprint" og "ProductFootprint" kan renderere deres indhold.

### **SharedModule**

Definerer alle delte komponenter og services, som kan anvendes på tværs i applikationen.

<sup>1</sup> Progressive Web Application [https://en.wikipedia.org/wiki/Progressive\\_web\\_application](https://en.wikipedia.org/wiki/Progressive_web_application)

<sup>2</sup> Angular <https://angular.io/>



### **MaterialModule**

Inkluderer og udstiller de dele af Angular Material biblioteket<sup>3</sup> som anvendes i applikationen.

### **StateModule**

Definerer de state services, som anvendes til delt state mellem komponenter og moduler, herunder AppState, DataState, NetworkState.

### **DataEditModule**

Definerer indholdet på data- og scenariosiderne, herunder dynamiske inputkomponenter der kan vises som dropdowns, radioknapper, streng- eller numeriske inputfelter samt services, som hjælper med at mappe data og gemme ændringer, som brugeren foretager.

Dette modul definerer således byggestenene som bruges til at sammensætte indtastningssiderne.

På figur 11. Dynamic-input.component.html ses layoutet for et dynamisk input, som kan skifte sin visning ud fra "controlType" attributten på "userInput".

```
1 <div class="input-field" [ngSwitch]="userInput.parameterInput?.controlType" [matTooltip]="tooltip">
2 > <mat-icon class="info" *ngIf="userInput.parameterInput.descriptionText" ...
4 </mat-icon>
5
6 > <app-radio-buttons *ngSwitchCase="controlTypes.Radio" [value]="userInput | selectedItem" ...
10 </app-radio-buttons>
11
12 > <app-number *ngSwitchCase="controlTypes.Number" [form]="form" [value]="userInput | editedValue" ...
18 </app-number>
19
20 > <app-string *ngSwitchCase="controlTypes.String" [form]="form" [value]="userInput | editedValue" ...
24 </app-string>
25
26 > <app-dropdown *ngSwitchCase="controlTypes.Dropdown" [form]="form" ...
31 </app-dropdown>
32
33 <mat-icon class="error" *ngIf="errorMessage" [matTooltip]="errorMessage">error</mat-icon>
34 </div>
```

Figur 11. Dynamic-input.component.html

Udover selve inputelementet kan komponenten også vise et tooltip, en infotekst og en fejlbesked, hvis disse er defineret.

---

<sup>3</sup> Angular Material <https://material.angular.io/>

De dynamiske inputs kan anvendes i generiske rækker, hvis opbygning ses på Figur 12. Dynamic-input-rows.component.ts, hvor det er forsøgt illustreret, hvilke komponenter der udgør en input-række.

```

1 <div *ngFor="let inputRow of groupedInputVMState.value$ | async | keyvalue : bySortOrder">
2   <app-input-row *ngFor="let subRow of inputRow.value['subRows'] | keyvalue; let i = index">
3     A <app-input-label [inputRow]="subRow.value">
4       </app-input-label>
5
6     B <app-input-source [userInput]="subRow.value | lastInCollection">
7       </app-input-source>
8
9     <app-dynamic-input (valueChanged)="updateValueInConfig($event, inputRow.key, subRow.key)"
10    C [userInput]="subRow.value | lastInCollection" [form]="form" [cardKey]="cardKey">
11      </app-dynamic-input>
12
13    <app-remove-input-row *ngIf="inputRow.value['canRemoveRow'] && canAddRows && i === 0"
14    D (clicked)="removeInputRow(inputRow.key)">
15      </app-remove-input-row>
16    </app-input-row>
17  </div>
18 <app-add-input-row *ngIf="canAddRows && rowOptions?.length"
19 E (optionSelected)="addInputRow($event)" [options]="rowOptions" [optionsText]="optionsText">
20 </app-add-input-row>

```

Figur 12. Dynamic-input-rows.component.ts



SEGES - Landbrug & Fødevarer F.m.b.A.  
 Agro Food Park 15, 8200 Aarhus N

T: +45 8740 5000 - F: +45 8740 5010 - E: info@seges.dk

Ansvar: Informationerne på denne side er af generel karakter og søger ikke at løse individuelle eller konkrete rådgivningsbehov. SEGES er således i intet tilfælde ansvarlig for tab, direkte såvel som indirekte, som brugere måtte lide ved at anvende notatets informationer.