

# Rapport - AP1

## SmartFlow

### Dokumenthistorik:

<i>Version</i>	<i>Initialer</i>	<i>Datum</i>	<i>Kommentar</i>
1.0	HBI	2021-01-25	Dokument framtaget efter arbetspaket 1 i Vinnova uppdraget Smartflow

## Innehållsförteckning

1	Inledning.....	4
1.1	Syfte och målgrupp .....	4
1.2	Omfattning .....	4
1.3	Termer, definitioner och förkortningar.....	5
1.4	Avgränsningar och förutsättningar .....	8
2	Användningsfall och aktörer .....	11
2.1	Översikt .....	11
2.2	Aktörer .....	11
2.2.1	Användare .....	11
2.2.2	Dataleverantör .....	11
2.2.3	Beställare .....	11
2.2.4	Administratör.....	12
2.2.5	Agent .....	12
2.3	Användningsfall .....	12
2.3.1	Administrativa användningsfall (Konfiguration) .....	12
2.3.2	Dataleverantör .....	13
2.3.3	Beställare .....	14
2.3.4	Allmänna användningsfall.....	15
3	Begreppsmodell.....	16
3.1	Översikt .....	16
3.2	Begreppsdefinitioner .....	17
3.2.1	Projekt .....	17
3.2.2	Skede .....	17
3.2.3	Leveransspecifikation.....	17
3.2.4	Dataleverans .....	17
3.2.5	Datamängd.....	17
3.2.6	Metadatamängd .....	17
3.2.7	Informationsmodell .....	17
3.2.8	Instansdatamängd .....	17
3.2.9	Koncept .....	17
3.2.10	Transformationsdefinition .....	17
4	Testfall och scenarier.....	18

5	Övergripande arkitektur .....	21
5.1	Principiell systemlösning – dataflöde.....	21
5.1.1	Principiell systemlösning – komponenter .....	23
2.3.5	Principiell systemlösning – datalagring.....	24
2.3.6	Principiell systemlösning – deployment.....	25
6	Tidplan för leveranser av testdata.....	27
7	Bilaga 1 – Fördjupning kring transformering av datamängder .....	28

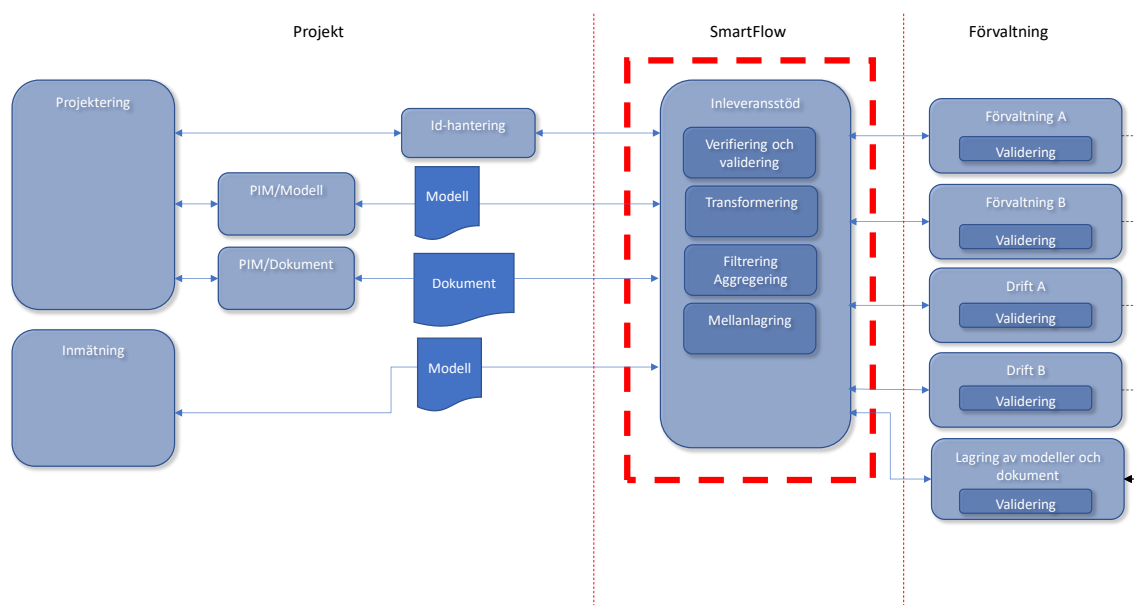
# 1 Inledning

## 1.1 Syfte och målgrupp

Syftet med detta dokument är att beskriva kravställningen för det IT-tekniska system som ska tas fram inom ramen för projektet SmartFlow och utgöra ett inleveransstöd med syfte att kunna automatisera dataflöden mellan infrastrukturprojekt (både ny- och ombyggnation och inventering/inmätning) och anläggningsförvaltningen.

Målgruppen för dokumentet är i första hand företag och behovsägare inblandade i det pågående SmartFlow projektet. Målgruppen innefattar även Vinnova och övriga som har intresse av att få en första överblick över problemet som projektet avser studera samt ett förslag till lösningsdesign.

## 1.2 Omfattning



Figur 1 - Översiktsskiss

Lösningen som specificeras ska fungera som en typ av informationsmäklare mellan de verktyg som finns i projekten och de förvaltningssystem som är mottagare och användare samt kravställare. Som projekt kan räknas både investerings- och förvaltningsprojekt samt även projekt vars syfte är att samla information om existerande anläggning där det idag saknas information helt eller delvis. Observera att information kan flyta i båda riktningarna där underlag, t ex i form av är-läge och kravställning flyter från förvaltning till projekt och exempelvis arbetsplaner, bygghandlingar och relationshandlingar från projekt till förvaltning. Högsta prioritet har flödet från projekt (relationshandling) till förvaltning.

En viktig del i plattformen är förmågan att kunna transformera data för att tillrättalägga det så att det tillgängliggörs på ett för mottagaren optimalt sätt. Det kan t ex gälla använd nomenklatur och detaljeringsnivåer för data.

En viktig aspekt i flödet är behovet av en strukturerad hantering av identifikation för att kunna koppla samman information som flyter i bägge riktningar. Man kan exempelvis se det som att det från förvaltning till projekt, förutom information om nuläge, flyter information om kravställning för den nya anläggningen, ner till någon detaljeringsgrad. Redan i det läget är delar av anläggningen identifierad. Leveranser som sedan sker åt andra hållet, från projektet till förvaltning, ska naturligtvis vara kopplad mot samma anläggning enligt beslutad identifikation. Denna ”handskakning” är central i ett fungerande flöde men kan fungera olika i olika typer av projekt och för olika typer av anläggningar.

En annan viktig aspekt rör verifiering och validering. Verifieringen rör primärt de delar som i hög grad borde kunna utföras automatiserat mot maskinläsbara kravspecifikationer. Detta kan t ex röra saker som:

- Finns alla kravställda objekt (av rätt typ) med i leveransen?
- Innehåller dessa objekt samtliga obligatoriska attribut och nödvändig klassifikation?
- Innehåller attributen godkända värden inom stipulerade värdeförråd med rätt enheter osv?
- Finns data representerade med rätt typ av geometri och inom angivet koordinatsystem med tillräcklig noggrannhet?

Valideringen rör mera de delar som den mottagande verksamheten själv måste hantera ur ett verksamhetsmässigt perspektiv och kan innefatta även manuell granskning.

Exakt vad verifiering och validering ska innebära får detaljeras vidare i projektet. Tills vidare kan vi i alla fall konstatera att ett leveransflöde behöver kunna hanteras i steg och att godkännande av leveranser behöver kunna innefatta både automatiserade och manuella steg som kan innehålla kommunikation mellan alla ingående parter.

## 1.3 Termer, definitioner och förkortningar

### AIM

**Asset Information Model** - An "Information Model relating to the Operational Phase" ISO 19650-1 (3.3.9). The Asset Information Model (AIM) supports the maintenance, management, and operation of an Asset throughout its Asset Life Cycle. AIM can act (i) as a repository for all Asset Information; (ii) as a means to access/link to enterprise systems (e.g. CMMS and BMS); and (iii) as a means to receive and centralise information from other Project Participants throughout Project Lifecycle Phases

<https://bimdictionary.com/en/asset-information-model/2>

### (Data-)transformation

I detta dokument avses här semantisk transformation, dvs översättning av data från en vokabulär till en annan, t ex från IFC 4.3 till BatMan. Denna transformation kan även innebära aggregering av data från en detaljeringsnivå till en annan eller på motsvarande sätt förenkling av geometri.

## Dokument

Med dokument avses i detta dokument primärt data som inte är avsett att kunna tolkas maskinellt på annat sätt än att de kan presenteras för att tolkas av människor. Dokument kan dock förses med maskinläsbara metadata som beskriver innehållet på ett sätt som t ex medger sökning, gruppering mm samt unik identifikation för referens från andra dokument eller objekt.

### FME

<https://www.safe.com/>

### Formatkonvertering

I detta dokument avses här översättning mellan olika filformat utan att ändra det semantiska innehållet i en datamängd, t ex mellan STEP Physical File Format (SPFF) och RDF/Turtle.

### GeoDCAT-AP

<https://inspire.ec.europa.eu/good-practice/geodcat-ap>

### GML

<https://www.ogc.org/standards/gml>

### GraphDB

<https://www.ontotext.com/products/graphdb/>

### IFC

<https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/>

### IFC 4.3

[http://standards.buildingsmart.org/IFC/DEV/IFC4\\_3/RC2/HTML/](http://standards.buildingsmart.org/IFC/DEV/IFC4_3/RC2/HTML/)

### JSON

<https://www.json.org/>

### (Koordinat-)transformation

Med detta avses transformation av koordinater mellan olika koordinatsystem, t ex mellan ett lokalt koordinatsystem för ett projekt (i PIM) och ett rikstäckande koordinatsystem, t ex SWEREF99 TM (i AIM).

### Linking Rule Set

En typ av ontologi vars syfte är att beskriva relationer mellan begrepp i olika ontologier som grund för automatiserad datatransformation, t ex via reasoning. Synonym: Alignment ontology.

### Modell

Med modell avses i detta dokument data som distribueras med öppna format där syftet med innehållet är att detta ska tolkas och användas för vidare bearbetning och lagring, t ex IFC-filer för överföring till förvaltningssystem. Modeller ska vara organiserade objektorienterat, dvs att data ska organiseras som en mängd objekt med relationer och egenskaper (där även t ex geometri och placering utgör egenskaper). Modeller behöver, precis som dokument, förses med metadata som beskriver innehållet på ett sätt som t ex medger sökning, gruppering mm samt unik identifikation för referens från andra dokument eller objekt.

## OGC

<https://www.ogc.org/>

## Ontology

<https://www.w3.org/standards/semanticweb/ontology>

## OpenTNF

<https://www.opentnf.org/>

## OWL

<https://www.w3.org/OWL/>

## PIM

**Project Information Model** - An “Information Model relating to the Delivery Phase” ISO 19650-1 (3.3.10) of an Asset Life Cycle. The Project Information Model (PIM) includes Information Containers covering the design and construction of new assets, and the re-use and renovation of existing assets. PIM contributes to the Asset Information Model (AIM) throughout the different phases of an asset life cycle

<https://bimdictionary.com/en/project-information-model/2>

## QGIS

<https://qgis.org/>

## QUADRI

<https://civil.trimble.se/produkter/quadri>

## RDF

<https://www.w3.org/RDF/>

## RDFS

<https://www.w3.org/TR/rdf-schema/>

## (Semantic) Reasoning

Mjukvara som kan härleda de logiska konsekvenserna från en mängd explicita fakta enligt regler i ett språk för ontologier.

Exempel:

*Explicita fakta*

- *Motorfordon är en klass*
- *Fordon är en klass*
- *Motorfordon är en subklass till Fordon*
- *A tillhör klassen Motorfordon*
- *Person är en klass*
- *Egenskapen Personnummer har domänen Person*
- *B har egenskapen Personnummer = xxxxx-nnnn*

### *Härledda konsekvenser*

- *A tillhör klassen Fordon*
- *B tillhör klassen Person*

### **Shape**

<https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

### **SPARQL**

<https://www.w3.org/TR/rdf-sparql-query/>

### **SOSI**

<http://skjema.geonorge.no/SOSI/>

### **TNE**

<https://tne.triona.se/>

### **TripleGeo**

<https://github.com/GeoKnow/TripleGeo>

### **Turtle**

<https://www.w3.org/TR/turtle/>

## **1.4 Avgränsningar och förutsättningar**

Projektet omfattar inte dynamisk information där data behöver finnas tillgänglig i realtid, dvs vid det tillfälle den uppstår. Detta kan t ex röra data från sensorer som antas kräva separata kanaler.

För projektet SmartFlow så görs även avgränsningar vad gäller de format som ska kunna hanteras. Anledningen till dessa begränsningar är primärt för att projektet primärt vill undersöka den principiella lösningen snarare än att utveckla ett komplett och leveransfärdigt system. För hantering av den potentiellt sett stora mängd med filformat som kan vara aktuell så hänvisas till produkter som t ex FME eller motsvarande.

Som indata i projektet avgränsas formatfloran till IFC 4.3, Shape, OpenTNF och RDF. Anledningen till detta är att IFC 4.3 antas bli dominerande för leveranser av data för om- och nybyggnadsprojekt, åtminstone då det gäller anläggningens fysiska och rumsliga aspekter. IFC är dessutom en ISO-standard som har ett brett stöd hos programvara som stödjer BIM. Observera att de format som räknas upp ovan är en avgränsning som görs inom projektet och det finns inte några tekniska hinder för att utöka detta i en framtid.

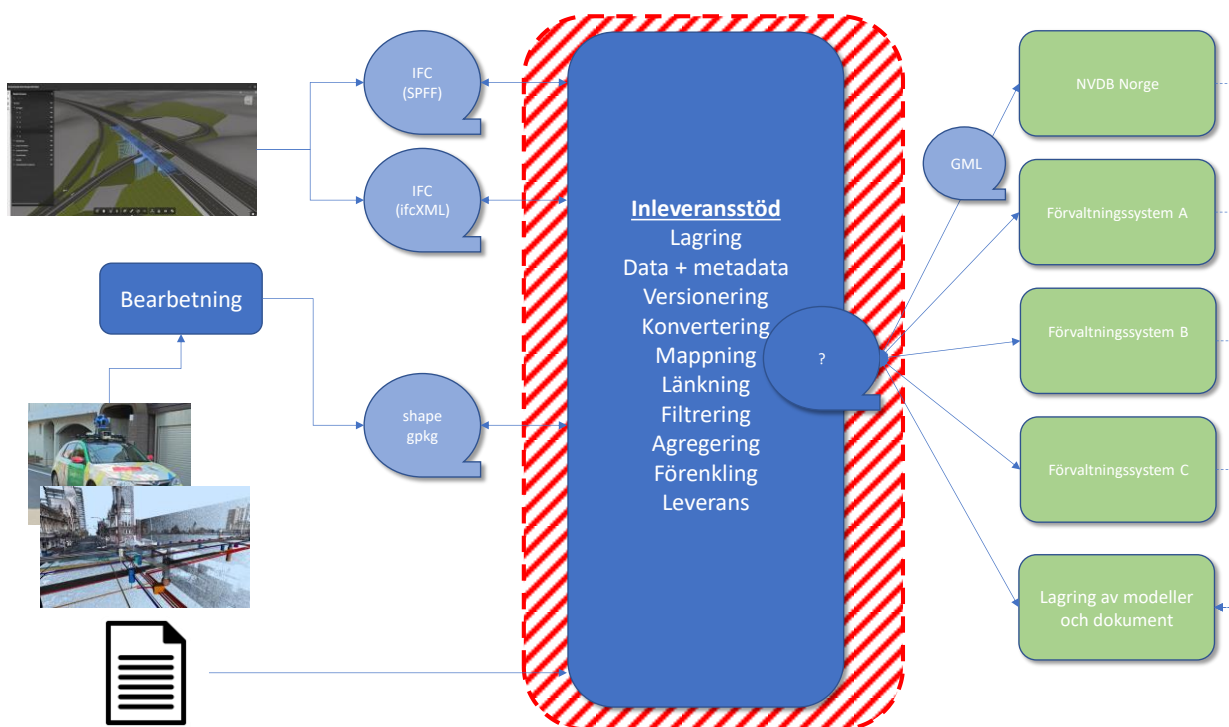
Shape-formatet är ett relativt enkelt format som kan skapas och läsas av många. När det gäller dataflöden från inmätning av befintlig anläggning antar vi att dessa data enkelt kan fås via shape-format eller motsvarande. Den bearbetning, manuell eller automatisk (t ex via machine learning och dyligt) som krävs för att tolka och klassificera mätdata ingår inte i SmartFlow-lösningen.

För utdata kommer RDF att användas eftersom den teknologi som ska testas baseras på RDF och övriga teknologier för länkade data/semantisk web. RDF-formatet är generellt och kan användas för att



uttrycka godtyckliga data på ett maskintolkningsbart sätt. RDF kan kodas som XML, JSON, Turtle m.fl. textformat och bör därför kunna konsumeras på enkelt sätt av mottagande applikationer.

För utdata i projektet kommer även OpenTNF (baserat på OGC GeoPackage) samt GML att användas. OpenTNF stöds av bl.a. av Trionas produkt TNE och Trimble's Quadri-plattform och är även direkt läsbar från GIS-plattformar som t ex ESRI Arc\*, FME och QGIS. GML är ett primärt format för inleverans av vägdata till Statens Vegvesen.



Figur 2 Illustration av en tänkbar situation

Figuren ovan illustrerar en tänkbar situation.

Som indata från projekten kommer IFC-, Shape- eller GeoPackage-format att levereras beroende på om data kommer från om- och nybyggnad eller inmätning. För norska scenarier är endast IFC till GML relevant.

Mottagande förvaltningssystem använder företrädesvis någon typ av databaslösning med sin egen unika struktur baserad på de behov som förvaltningssystemet ska stödja.

Detta innebär att det finns ett behov av datatransformationer för att data ska kunna flyta mellan projekt och förvaltning. Här kommer SmartFlow server att kunna transformera data mellan formatet som levereras från projektet till format som stöds av förvaltningssystemet. SmartFlow systemet strävar efter att uppnå följande;

- Val av ett enhetligt och lätt konsumerat filformat för utleverans enligt ovan. Andra möjliga filformat hade exempelvis kunnat vara XML enligt XSD-scheman eller JSON, eventuellt

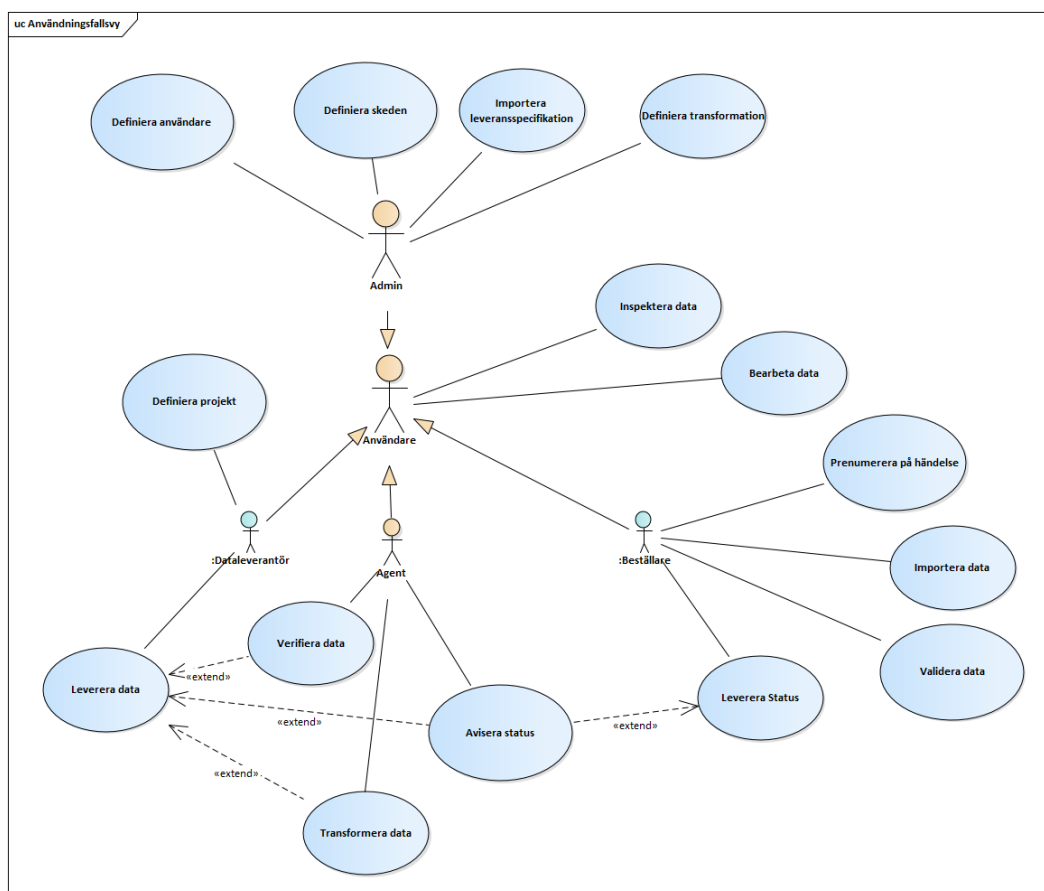
enligt JSON schema. Samtliga av dessa klarar att uttrycka godtyckliga data i ett enhetligt filformat.

- Mottagande system tillhandahåller sina informationskrav på maskinläsbar form som OWL-ontologi eller datakatalog till SmartFlow-lösningen
- För hantering av indata från projekt använder vi IfcOWL för IFC-filer samt egenutvecklade OWL-ontologier eller datakataloger för att representera innehållet i shape-filer
- Indata konverteras till RDF-format enligt relevant OWL-ontologi
  - o För IFC används komponenten IFC2RDF för att generera RDF från IFC-filer på STEP eller XML-format
  - o För Shape/GeoPackage används komponenten TripleGeo för att generera RDF
- För GML kommer applikationsschema för NVDB Norge användas som avgränsning.
- Som stöd för att transformera data från indata till utdata används separata så kallade alignment-ontologier som använder sig av standardiserad OWL- och RDFS-vokabulär för att beskriva sambanden mellan in- och utdata
  - o I SmartFlow-lösningen måste även finnas stöd för viss bearbetning som t ex aggregering av data, datafiltrering, transformering/förenkling av geometri etc. Förenkling av geometri är mycket specialiserat och antas implementeras separat eftersom den även kan innebära manuella moment i ett avancerat grafiskt verktyg. För andra komplexa samband, aggregeringar, filtreringar och liknande kan lagrade SPARQL-frågor användas
- SmartFlow-lösningen måste kunna innehålla metadatabeskrivningar av datamängder, gärna på ett flexibelt sätt, men i största möjliga mån i förhållande till förekommande metadatastandarder, t ex GeoDCAT-AP.

## 2 Användningsfall och aktörer

### 2.1 Översikt

För att ge en ökad förståelse för det tänkta inleveransstödet har några aktörer och användningsfall definierats översiktligt. Dessa ska inte ses som slutgiltiga och arbetet behöver förfinas i samband med fortsatt arbete, t ex i form av user stories. I dessa user stories kommer olika scenarion att beskrivas.



Figur 3 - Användningsfall och aktörer

## 2.2 Aktörer

### 2.2.1 Användare

Någon som minst har läsrättigheter till systemet

### 2.2.2 Dataleverantör

Någon som har rättigheter att leverera data

### 2.2.3 Beställare

Någon som är beställare och mottagare av data och aviserade händelser

## 2.2.4 Administratör

Någon som har alla rättigheter och som kan konfigurera systemet

## 2.2.5 Agent

En automatisk process som utför någon form av arbete baserat på en trigger.

# 2.3 Användningsfall

## 2.3.1 Administrativa användningsfall (Konfiguration)

### 2.3.1.1 Definiera användare

Användningsfallet startar när en ny användare behöver tillgång till systemet.

Administratören registrerar användaren och knyter denna till en roll som ger tillgång till nödvändiga data och funktionalitet. Detaljerade krav angående hantering av användare, roller och behörigheter får beskrivas separat. Kan t ex röra kopplingar till organisationsinterna eller externa system för autentisering och auktorisering.

En användare kan ha flera roller.

### 2.3.1.2 Definiera skeden

Användningsfallet startar när det finns ett behov av att förändra eller lägga till ett skede. Skeden används för att beskriva mognadsgraden på en dataleverans.

### 2.3.1.3 Importera leveransspecifikation

Användningsfallet startar när det finns en leveransspecifikation som behöver läggas in i systemet.

Administratören väljer att importera leveransspecifikation samt knyter nödvändiga metadata till denna. Administratören kan importera eller manuellt registrera metadata som beskriver leveransspecifikationen.

Leveransspecifikationen samt metadata verifieras och lagras i systemet.

Användningsfallet avslutas när leveransspecifikationen inklusive metadata godkänns och lagrats och är färdig att användas vid leveranser av datamängder.

Där fel uppstår med verifiering eller lagring skickas en kvittering med avvikelserapport. Aktiviteten loggas i systemet (leverantör, tidpunkt, referens till datamängd samt avvikelserapport) notifiering enligt 2.3.3.1

En leveransspecifikation anger kraven för de objekttyper, egenskaper och relationer som skall eller får förekomma i en leverans. Leveransspecifikationen anger även tillåtna värdeförråd för egenskaper. Leveransspecifikationen är inom ramen för denna kravställning maskinläsbar vilket innebär att innehållet i en datamängd kan tolkas, verifieras och transformeras maskinellt. Varje levererad datamängd tillhör och ska uppfylla en leveransspecifikation.

Leveransspecifikationen används för att beskriva både Dataleverantörens inleverans och Beställarens behov och krav på utdata.

#### 2.3.1.4 Definiera transformation

Användningsfallet startar när leveransspecifikation ändras eller tillkommer

En administratör kan definiera regler för transformationer mellan olika leveransspecifikationer.

Innehållet kan variera beroende på skede i projektet (t ex arbetsplan eller bygghandling).

Transformationen sparas i systemet med uppgifter som gör att den automatiskt kan exekveras i samband med leveranser av datamängder. Mera information om transformationer finns i Bilaga 1 – Fördjupning kring transformering av datamängder

### 2.3.2 Dataleverantör

#### 2.3.2.1 Definiera projekt

Användningsfallet startar när Dataleverantören startar ett nytt projekt för att beskriva en planerad förändring i anläggningen.

Dataleverantören definierar projekt genom att ange vilka datamängder som ska levereras och vid vilka skeden och tidpunkter detta kan ske. För varje datamängd beskrivs även vilka leveransspecifikationer som datamängden ska uppfylla. Ett projekt kan innehålla delprojekt med egna leveransspecifikationer och skeden.

#### 2.3.2.2 Leverera data

Användningsfallet startar när villkoren för en dataleverans är uppfyllda och dataleverantören vill utföra en leverans.

Dataleverantören väljer att ladda upp en datamängd samt tillhörande metadata i godkänt format och med definierat innehåll (enligt leveransspecifikation) till systemet. Systemet verifierar datamängden samt metadata i förhållande till de regler som gäller för den specifika typen av datamängd. När datamängden verifierats (se 2.3.3.5) och godkänts lagras datamängden i systemet. Dataleverantören får ett kvitto på utförd leverans.

Systemet ska automatiskt kunna utföra transformationer av levererade datamängder i de fall att sådana transformationer definierats för data enligt berörd leveransspecifikation (se 2.3.3.7)

Aktiviteten loggas i systemet (leverantör, tidpunkt och referens till projekt, skede och leveransspecifikation)

En datamängd kan utgöra en ny version för en datamängd som redan levererats till systemet. Systemet ansvarar för att hantera versionssamband mellan datamängder.

Där fel uppstår med verifiering eller lagring skickas en kvittering med avvikelserapport. Aktiviteten loggas i systemet (leverantör, tidpunkt, referens till datamängd samt avvikelserapport).

Angående notifiering se 2.3.3.1

### 2.3.3 Beställare

#### 2.3.3.1 Prenumerera på händelser

Användningsfallet startar när Beställaren vill specificera vilka projekt och händelser hen är intresserad av.

Beställaren prenumererar på status händelser på enskilda objekt i ett projekt enligt de krav som finns beskriva i beställarens leveransspecifikation.

#### 2.3.3.2 Importera data

Användningsfallet startar när det finns en aviserad händelse som beställaren prenumererar på och som genererar ett data som motsvarar en av beställarens kravställd leveransspecifikation.

Beställaren identifierar den datamängd som ska importeras. Beställaren kan definiera kriterier för filtrering av datamängden i samband med nedladdning samt eventuellt det format som ska användas för den nedladdade datamängden. Aktiviteten loggas i systemet (användare, tidpunkt och referens till datamängd). Angående notifiering se 2.3.3.1

Efter nedladdning av önskad datamängd sker validering och import i mottagande system.

Beställaren använder rutiner i det egna systemet för import samt hantering av de fall där importen misslyckas. Resultatet av import och validering levereras till systemet (se 2.3.3.4).

#### 2.3.3.3 Validera data

Användningsfallet startar när användaren laddat ner en datamängd enligt 2.3.3.2

Beställaren använder rutiner i det egna systemet för validering av data efter validering levererar beställaren tillbaka resultat som en status rapport

Resultatet av import och validering levereras till systemet (se 2.3.3.4).

#### 2.3.3.4 Leverera status

Användningsfallet startar när beställaren importerat en datamängd och utfört en validering.

Resultatet av denna process rapporteras till systemet

Angående notifiering se 2.3.3.1

#### 2.3.3.5 Verifiera data

Användningsfallet startar när en datamängd levererats till systemet.

Systemet använder en maskinläsbar leveransspecifikation för att automatiskt utföra verifieringen.

Resultatet loggas i systemet och de användare som ska notifiera får en notifikation.

#### 2.3.3.6 Avisera status

Användningsfallet startar i och med att det skett en typ av statusförändring i systemet som har abonnenter som ska aviseras om förändringen.

Användningsfallet avslutas när berörda abonnenter aviseras.

### 2.3.3.7 Transformera data

Användningsfallet startar automatiskt när en datamängd levererats och där det finns en definierad transformation mot en annan leveransspecifikation eller manuellt av en behörig användare när en ny transformation definierats.

Om en definierad datakonvertering för aktuell datamängd finns tillgänglig så exekveras transformationen. När transformationen genomförts lagras den transformerade datamängden i systemet. Aktiviteten loggas i systemet (leverantör, tidpunkt och referens till datamängd) angående notifiering se 2.3.3.1.

Vid en förändring av leveransspecifikationer eller transformationer kan man manuellt trigga en ny transformation.

Där fel uppstår med verifiering eller lagring skickas en kvittering med avvikelserapport. Aktiviteten loggas i systemet (leverantör, tidpunkt, referens till datamängd samt avvikelserapport) och de användare som ska notifieras får en notifikation.

Den ursprungliga datamängden behöver inte tas bort om transformationen misslyckas eftersom transformationen kan exekveras manuellt vid ett senare tillfälle efter att problemet identifierats och åtgärdats.

## 2.3.4 Allmänna användningsfall

### 2.3.4.1 Inspektera data

Användningsfallet startar när en användare vill inspektera levererade datamängder.

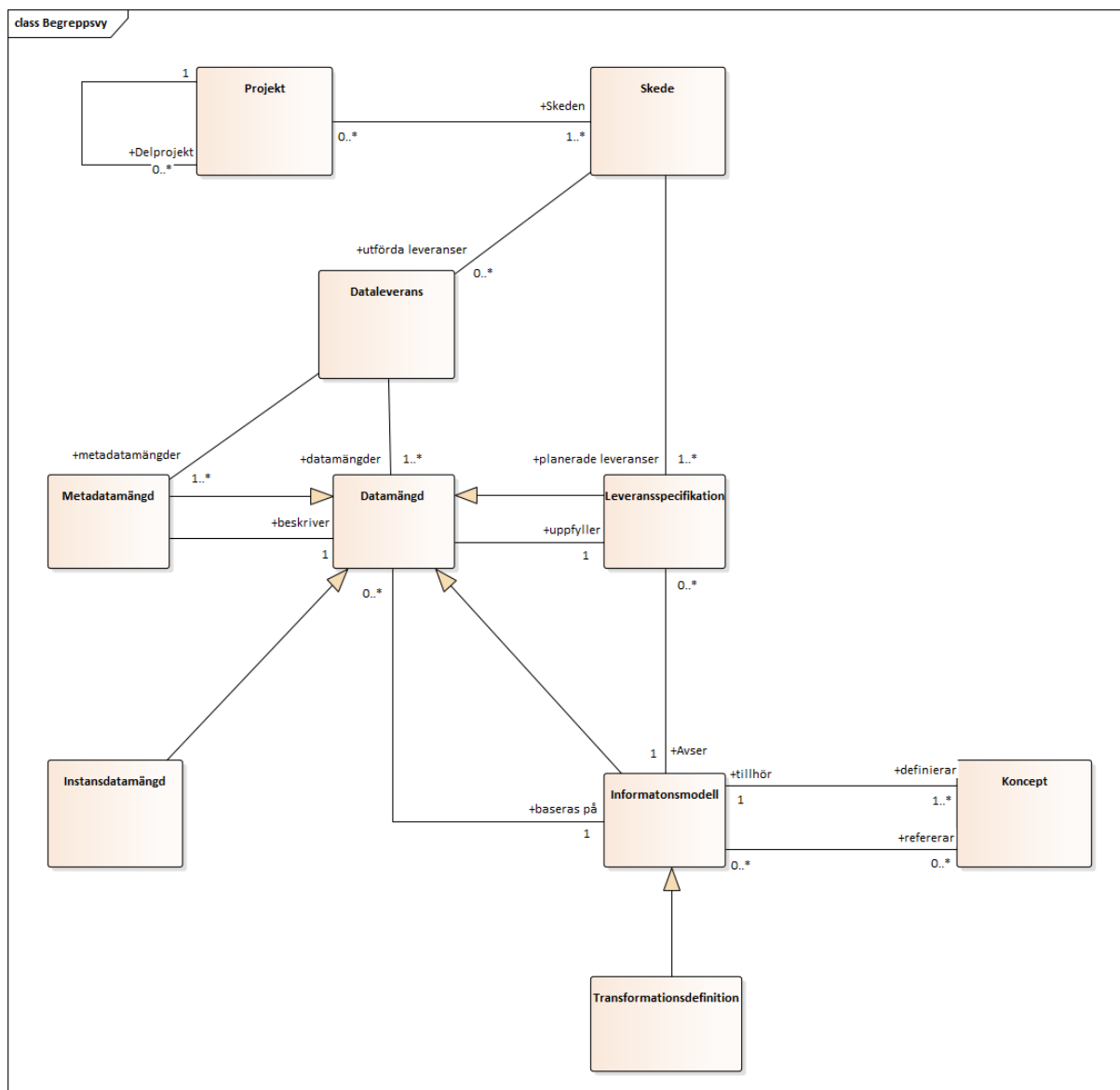
En användare kan inspektera data på olika sätt. Exakt hur behöver definieras. Möjliga delförlopp kan röra sig om:

- Söka / visa data eller metadata i textform (tabeller, hierarkier eller andra lämpliga sätt)
- Söka / visa data i grafisk form (karta/3d)

## 3 Begreppsmodell

### 3.1 Översikt

Figuren nedan redovisar en begreppsmodell som förklarar de begrepp som används i kapitel 2.3



Figur 4 – Begreppsmodell översikt



## 3.2 Begreppsdefinitioner

### 3.2.1 Projekt

Ett uppdrag som utförs av en tillfällig arbetsorganisation för att åstadkomma ett i förväg bestämt resultat i form av dataleveranser. Projektet kan delas upp i delprojekt och skeden.

### 3.2.2 Skede

Ett skede avser en specifik tidsavgränsad del av ett projekt med specificerade dataleveranser vid en viss tid.

### 3.2.3 Leveransspecifikation

En specifikation av en specifik typ av leverans. Specifikationen kan t ex avse leveransformat, informationsmodell och verifierings- och valideringsregler.

### 3.2.4 Dataleverans

En leverans av en eller flera datamängder och därtill hörande beskrivande metadatamängder.

### 3.2.5 Datamängd

En samling av separata dataelement som kan hanteras som en enhet.

### 3.2.6 Metadatamängd

En datamängd vars syfte det är att beskriva innehållet i en annan datamängd. . Metadatamängden beskrivs i termer av koncept som definierats i en informationsmodell.

### 3.2.7 Informationsmodell

En typ av datamängd som beskriver krav, struktur och innehåll för datamängder vad avser koncept och deras definition, relationer och egenskaper utgående från ett intresseområde.

Synonym: Ontologi

### 3.2.8 Instansdatamängd

En typ av datamängd som utgör en modell som beskriver en historisk, planerad eller existerande verklighet. Instansdatamängden beskrivs i termer av de koncept som definierats i en informationsmodell.

### 3.2.9 Koncept

En abstraktion av ett konkret eller abstrakt fenomen utgående från ett visst intresseområde.

Anm: Olika typer av koncept kan vara Objekttyper, Relationer mellan objekttyper och Egenskaper som beskriver objekttyper.

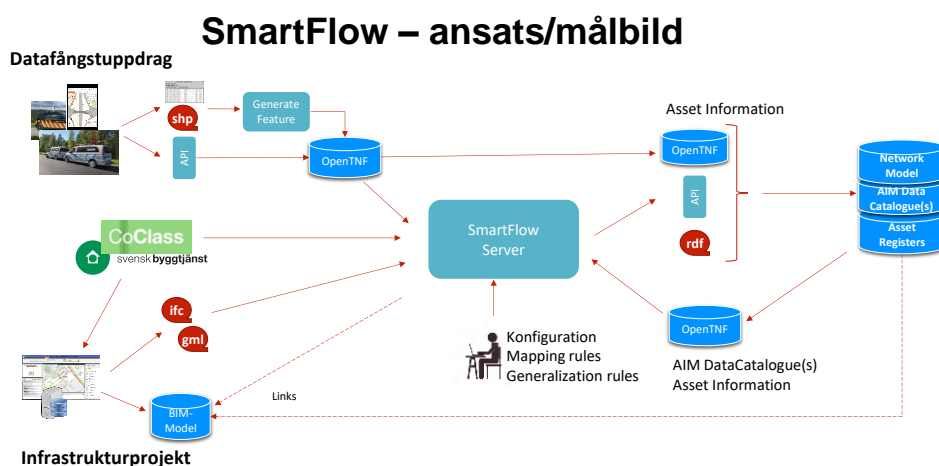
### 3.2.10 Transformationsdefinition

En typ av informationsmodell vars syfte det är att beskriva samband mellan koncept från olika informationsmodeller. Dessa samband kan användas vid transformationer av datamängder

Synonym: "Alignment ontology", Linking Rule Set

## 4 Testfall och scenarier

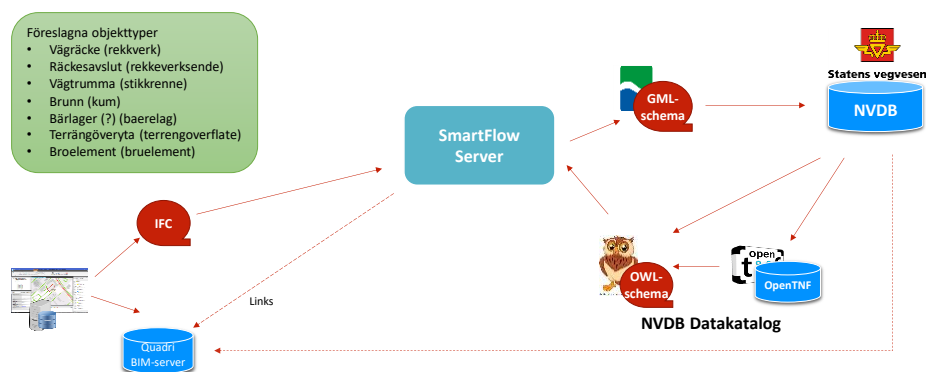
Nedanstående Figur 5 visar en översiktlig och generell beskrivning av de dataflöden och scenarier som vi planerat att hantera. Denna ansats/målbild har varit en utgångspunkt för dialoger inom AP1 med projektets intressenter om behov och möjliga testfall.



Figur 5 - Generell beskrivning av dataflöden

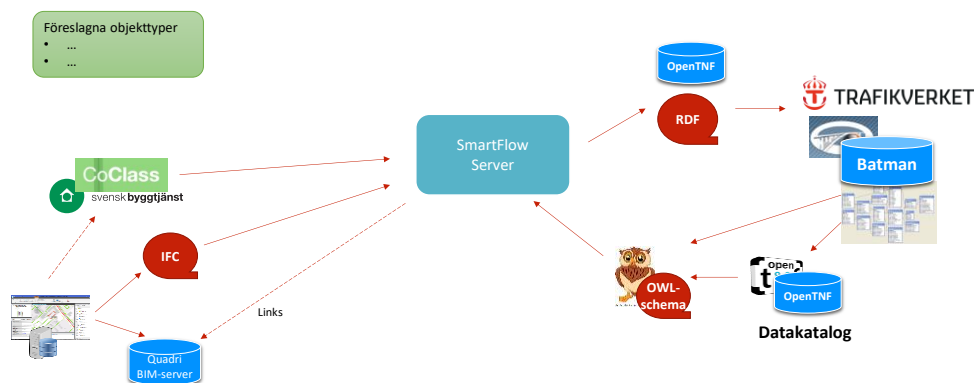
De dialogmöten som genomförts under AP1 med projektets intressenter har gett som resultat att vi för det fortsatta arbetet identifierat och föreslår en avgränsning till 5 stycken olika scenarier och dataflöden, se nedanstående figurer för scenario 1-5. Beskrivningen av dessa scenarier avseende t.ex. indata och utdata kommer succesivt att förfinas i kommande arbetspaket.

### Scenario1 – FV 251 Hellefossen -> NVDB



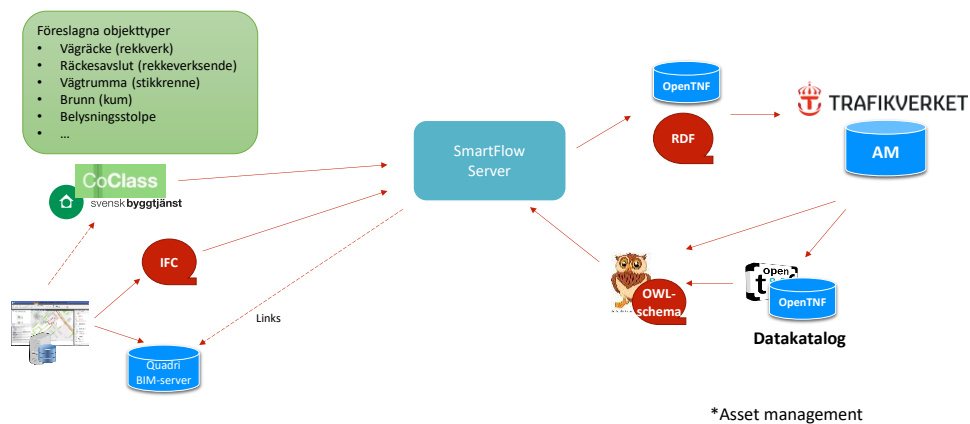
Figur 6 - Scenario 1: FV251 Hellefossen -> NVDB

## Scenario 2 – TPL Lindvreten -> Batman



Figur 7 - Scenario 2: TPL Lindvreten -> Batman

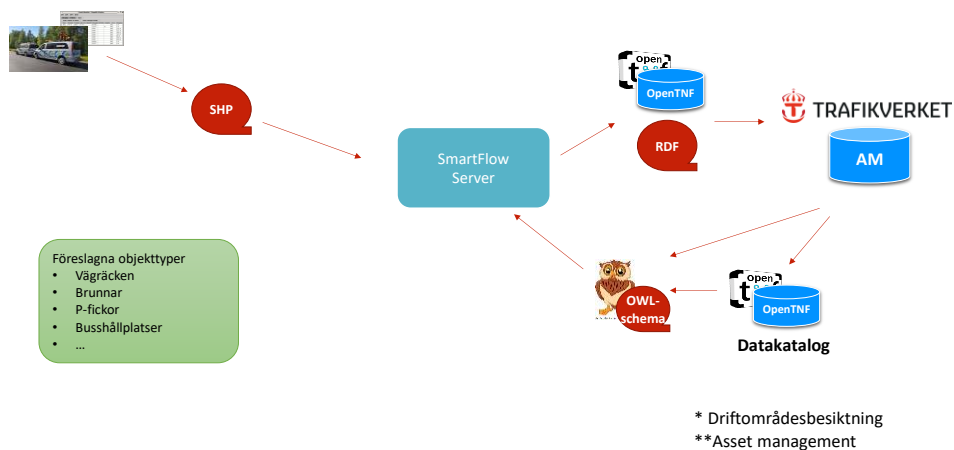
## Scenario 3 – TPL Lindvreten -> AM\*



\*Asset management

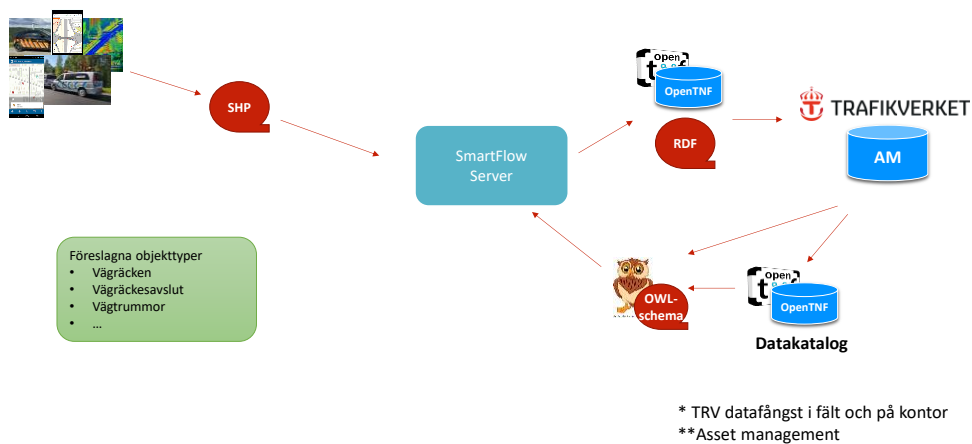
Figur 8 - Scenario 3: TPL Lindvreten -> AM\*

## Scenario 4 – DO-besiktning\* -> AM\*\*



Figur 9 - Scenario 4: DO-beskrivning\* -> AM\*\*

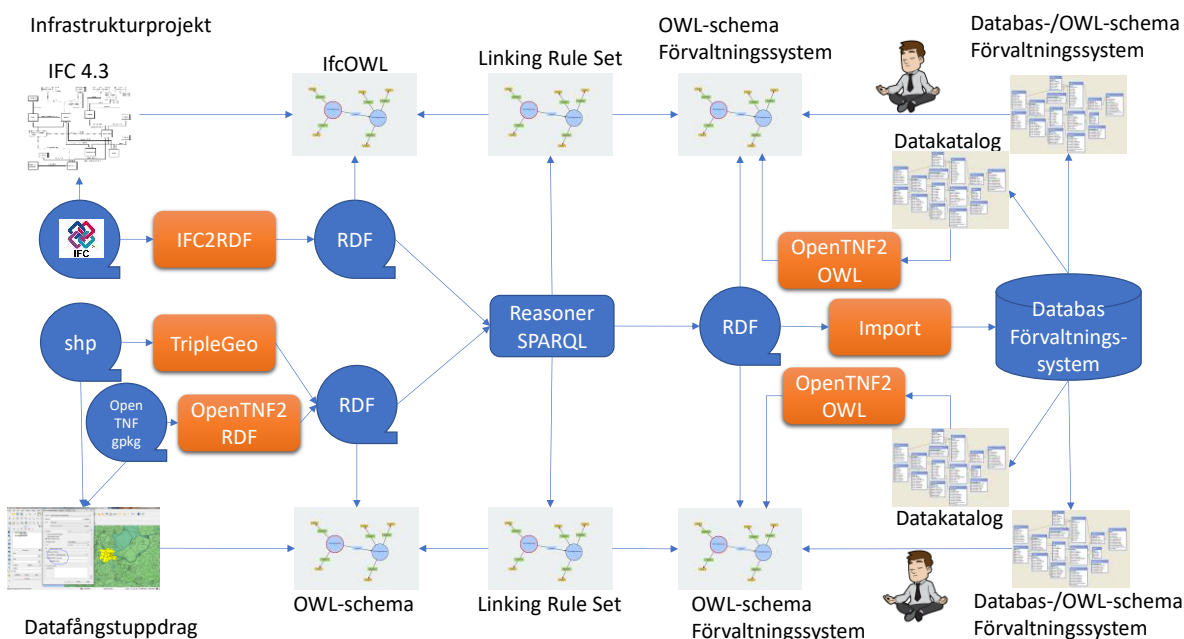
## Scenario 5 – Datafångst\* -> AM\*\*



Figur 10 - Scenario 5: Datafångst\* -> AM\*\*

## 5 Övergripande arkitektur

### 5.1 Principiell systemlösning – dataflöde



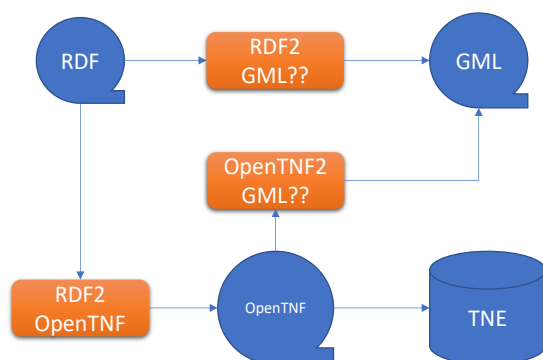
Figur 11 - Principiell dataflöde

Ovanstående principiell skiss illustrerar ett tänkt dataflöde:

1. Från förvaltningssystemet skapas en OWL-ontologi som motsvarar den datamängd som ska importeras till systemet, dvs motsvarar schema/datakatalogen för den egna datastrukturen
  - a. Detta motsvarar den leveransspecifikation som omnämns i användningsfallsmodellen
  - b. Det är förvaltningssystemets ansvar att detta tas fram
  - c. För datakataloger enligt principer i NVDB/TNE finns en komponent som automatiskt översätter datakataloger till OWL
  - d. För NVDB Norge finns en OWL-ontologi motsvarande datakatalogen här: <https://github.com/vegvesen/NVDB-Datakatalogen/tree/master/OWL>
2. Från IFC och shape-filer skapas eller används de OWL-ontologier som motsvarar dessa scheman
3. För varje kombination, t ex IFC till förvaltningssystem eller shape till förvaltningssystem skapas en alignment-ontologi som relaterar begrepp (objekttyper, relationer, egenskaper) i den ena med begrepp i den andra
  - a. Detta arbete är manuellt och kräver god kunskap om de begreppsvärldar som ska sammanlänkas
4. IFC- och shape-filer översätts automatiskt med IFC2RDF och TripleGeo från sina ursprungsformat till RDF-format

5. Med hjälp av Linking Rule Sets (Alignment-ontologier) samt eventuella SPARQL-frågor översätts och kompletteras indata till den struktur som efterfrågas av förvaltningssystemet
  - a. Detta genomförs med hjälp av dels standardiserad OWL-reasoning samt vid behov lagrade SPARQL-frågor
6. Förvaltningssystemet importerar data till den egna databasen

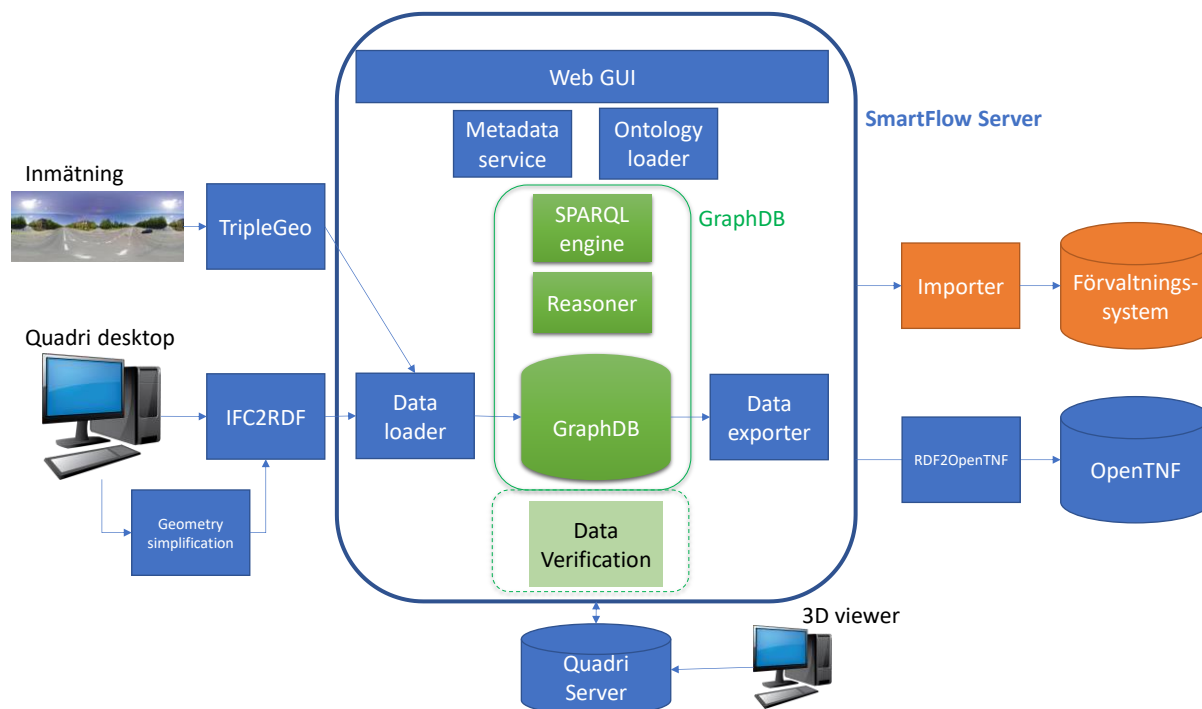
För fallet med GML-export motsvaras i bilden ovan ”OWL-schema Förvaltningssystem” av en OWL-representation av GML-schemat. För detta fall skrivs en specifik export av RDF till GML-format. Huruvida den exporten sker via OpenTNF eller direkt mellan RDF och GML är inte bestämt i nuläget.



Figur 12 - Överföring till TNE/GML

Överföring till TNE görs med hjälp av komponenten RDF2OpenTNF varvid TNE direkt kan läsa in OpenTNF-filen. TNE och OpenTNF kan användas som ”proxy” för de fall att andra förvaltningssystem kan vara svåra att engagera i detta projekt för testning. I dessa fall kan man i TNE definiera en datakatalog för den datamängd som ska testas och använda TNE som mottagande förvaltningssystem.

### 5.1.1 Principiell systemlösning – komponenter



Figur 13 – Komponentöversikt

#### 2.3.4.2 Inmätning

Aktivitet som ligger utanför SmartFlow med syfte att samla in data om befintlig anläggning. Detta kan ske manuellt med handdator, halv- eller helautomatiserat via exempelvis bildanalys eller analys av punktmoln från lasermätningar. I SmartFlow förutsätts dessa analyser vara gjorda och att filer i exempelvis Shape- eller OpenTNF-format kan levereras med klassificerade data.

#### 2.3.4.3 Quadri Desktop

Interaktiv BIM-klient från Trimble. Kan användas för import av diverse format eller modellering direkt i applikationen. Härifrån förutsätts man kunna exportera filer på IFC-format.

#### 2.3.4.4 Geometry simplification

Komponent för förenkling av geometrier. Exempelvis:

- Generera linje för räcke
- Generera punkt för brunn

#### 2.3.4.5 IFC2RDF

Befintlig Open Source-komponent för att transformera IFC-data på STEP-format till RDF-format enligt IfcOWL.

#### 2.3.4.6 Quadri server

Server för lagring av BIM-modeller. Vi förutsätter att man kan på ett enkelt sätt länka till dessa modeller för visning i Web-viewer.

#### 2.3.4.7 3D viewer

Komponent som visar BIM modeller från Quadri server i en webb-vy.

#### 2.3.4.8 SmartFlow server

Innehåller den funktionalitet som behöver utvecklas unikt för SmartFlow Server.

#### 2.3.4.9 Data loader

En tjänst som laddar ett dataset till GraphDB (i separat graf) samt förser detta med metadata.

#### 2.3.4.10 Ontology loader

En tjänst som laddar en ontologi till GraphDB (i separat graf?) samt förser denna med metadata.

#### 2.3.4.11 GraphDB

Triple store för lagring av RDF-data i grafer samt tillhörande funktionalitet.

#### 2.3.4.12 Reasoner

Ingår i GraphDB och innebär generering av nya tripplar baserat på information i ontologier.

#### 2.3.4.13 SPARQL engine

Ingår i GraphDB och ger ett http-interface för att läsa, filtrera och skapa data. Kan användas som ett komplement till Reasoner för mer avancerad datakonvertering, aggregering mm.

#### 2.3.4.14 Data Exporter

En tjänst som exporterar vald datamängd.

#### 2.3.4.15 Metadata service

En tjänst som ger möjlighet att skapa metadata samt söka information om vilka data som finns i databasen och deras status.

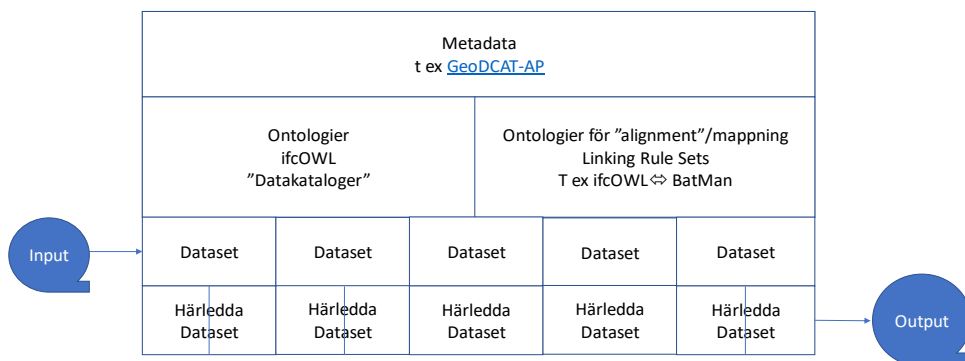
#### 2.3.4.16 Data verification

En tjänst som ger möjlighet att verifiera levererade data mot maskinläsbara leveransspecifikationer. Projektet bör testa både mvdXML

(<https://technical.buildingsmart.org/standards/ifc/mvd/mvdxml/>) och SHACL

(<https://www.w3.org/TR/shacl/>) som maskinläsbara leveransspecifikationer samt ge en framåtblick och, om möjligt, rekommendationer. En möjlighet kan även vara att utnyttja de inbyggda verifieringsmöjligheter som finns till traditionella datakatalogsbeskrivningar.

### 2.3.5 Principiell systemlösning - datalagring



Figur 14 – Datalagring



### 2.3.5.1 Metadata

Data som beskriver övriga data. Eftersom vi använder ett Triple store så utgörs även metadata av en graf enligt en vald ontologi för metadata. Gissningsvis ska vi basera metadata på GeoDCAT-AP.

### 2.3.5.2 Ontologier

Varje datamängd ska klassificeras mot en eller flera ontologier som ska finnas lagrade och beskrivna i miljön.

### 2.3.5.3 Ontologier för mappning

Ontologier med syfte att länka samman andra ontologier genom att beskriva begreppssamband. I OWL och RDFS finns en standardiserad vokabulär för detta.

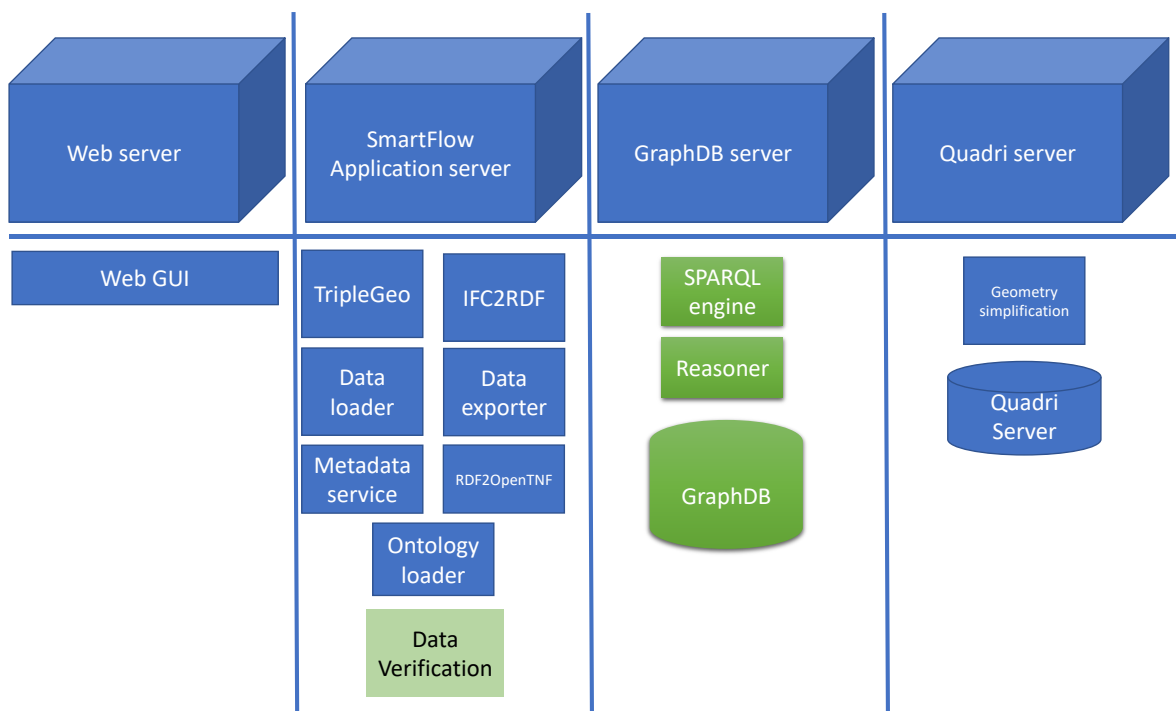
### 2.3.5.4 Dataset

Här avses de datamängder som läses in till SmartFlow för vidare bearbetning och leverans till beställaren. Varje datamängd bör kunna hanteras som en egen enhet i en separat graf och alla data är klassificerade enligt en eller flera ontologier. En uppsättning indata kan resultera i flertalet dataset, något som illustreras i de uppdelade rutorna i figuren ovan.

### 2.3.5.5 Härledda dataset

Detta avser de data i form av tripplar som härletts baserat på ontologier för mappning eller via separata SPARQL-frågor och vars syfte är export till beställaren.

## 2.3.6 Principiell systemlösning – deployment



Figur 15 - Principiell systemlösning för deployment

#### 2.3.6.1 Web Server

En web-server som härbärgerar det GUI som behöver tas fram för SmartFlow.

#### 2.3.6.2 SmartFlow Application Server

En applikationsserver som innehåller de tjänster som SmartFlow erbjuder via API:er.

#### 2.3.6.3 GraphDB Server

Server som innehåller GraphDB-miljön inklusive de tjänster (Lagring, Reasoner, SPARQL) som medföljer.

#### 2.3.6.4 Quadri server

Separat server för lagring och access av levererade BIM-modeller.

## 6 Tidplan för leveranser av testdata

För att testfall och scenarios som beskrivs i detta dokument ska hinna verifieras, så måste överenskomna testdata levereras av Trafikverket, Statens vegvesen/Trimble och Ramboll under perioden januari - mars 2021. Ju tidigare testdata kan levereras ju bättre!

## 7 Bilaga 1 – Fördjupning kring transformering av datamängder

Detta kapitel rör datatransformationer, dvs semantisk konvertering mellan olika vokabulärer utan beaktande av syntax (format) för representation av data. I SmartFlow utgår vi från att man alltid startar med en datakonvertering, dvs överför data till en gemensam syntax (format) för att där göra transformationerna. Den gemensamma syntax som valts är RDF eftersom den klarar att representera olika typer av data på ett enkelt sätt samt att man där har möjligheter till schemastyrda samband mellan vokabulärer som kan användas av reasoners för automatiska datatransformationer.

Grovt sett kan man tänka sig ett antal typiska fall för konvertering baserat på mappning mellan element i scheman. OBS att denna lista inte gör anspråk på att vara fullständig i detta läge:

Typ av mappning	Multiplicitet mellan Objekttyper Egenskapstyper Associationstyper	Multiplicitet mellan Instanser	Kommentar <i>Tillämpbart rdfs- eller owl-predikat</i>
<b>Ekvivalenta objekttyper</b>	1:1	1:1	Innebär att en instans av en objekttyp översätts till en instans av ekvivalent objekttyp <i>owl:equivalentClass</i>
<b>Ekvivalenta egenskaps- eller associationstyper</b>	1:1	1:1	Innebär att en egenskap översätts till ekvivalent egenskapstyp med eller utan datatypskonvertering <i>owl:equivalentProperty</i>
<b>Specifik vs generell objekttyp (klass=&gt;basklass)</b>	m:1	1:1	Innebär att en instans av den specifika objekttypen översätts till en instans av den mera generella objekttypen. Vice versa fungerar inte utan att mera information krävs (se 1:m-mappning) <i>rdfs:subClassOf</i>
<b>Specifik vs generell egenskaps- eller associationstyp</b>	m:1	1:1	Innebär att en egenskap kan typas som den generella egenskapstypen. Vice versa fungerar inte utan att mera information krävs (se 1:m-mappning) <i>rdfs:subPropertyOf</i>

<b>Generell vs specifik objekttyp (klass=&gt;subklass)</b>	1:m	1:1	Innebär att en instans av den generella objekttypen översätts till en instans av mera specifika objekttyper. I detta fall krävs mera information, t ex i form av egenskaper som kriterier för översättningen/uppdelningen  <i>rdfs:domain</i>  <i>SPARQL-construct</i>
<b>Generell vs specifik egenskapstyp eller associationstyp</b>	1:m	1:1	Innebär att en instans av den generella egenskapen kan typas som den mera specifika egenskapstypen. I detta fall krävs mera information som kriterier för översättningen/uppdelningen  <i>SPARQL-construct</i>
<b>Aggregering av objekt</b>	m:1	m:1	Flera instanser, av en eller flera objekttyper slås samman till en instans av annan objekttyp. Innebär normalt samtidig bearbetning av och/eller filtrering map egenskaper, t ex summeringar eller andra sammanslagningar  <i>owl:UnionOf</i>  <i>SPARQL-construct</i>
<b>Aggregering av egenskaper</b>	m:1	m:1	Flera egenskaper från ett eller flera objekt slås samman med någon algoritm (t ex summeringar eller konkateneringar, typkonverteringar) till en ny egenskap i samma eller annat objekt.  <i>SPARQL-construct</i>
<b>Uppdelning av objekt</b>	1:m	1:m	En instans delas upp till flera instanser av andra objekttyper. Innebär normalt samtidig filtrering map egenskaper samt eventuellt att associationer mellan objekt behöver skapas

			<i>SPARQL-construct</i>
<b>Uppdelning av egenskaper</b>	1:m	1:m	En egenskap delas upp i flera egenskaper av andra egenskapstyper.  <i>SPARQL-construct</i>

Ju fler av ovanstående transformationer som kan beskrivas som mappningar på schemanivå genom relationer mellan begrepp (objekttyper, egenskapstyper och associationstyper) och hanteras av programvara endast genom att tolka dessa mappningar desto mindre arbete behöver på sikt läggas "hårdkodat" i programvara när nya behov uppstår. Dessutom innebär dessa mappningar ofta ingående kunskaper kring ett intresseområde varför denna typ av begreppssamband bäst hanteras tillsammans med experter inom området. En förutsättning för att åstadkomma maskintolkningsbara beskrivningar av mappningar är att varje begrepp unikt kan identifieras för att undvika feltolkningar. Om "unikiteten" gäller utifrån ett universellt perspektiv så innebär det även att begrepp från t ex internationella standarder kan användas utan risk för kollisioner.