Netherlands Enterprise Agency

NIVA
NEW IACS VISION IN ACTION

# Architecture Precision Farming

Hepco Homminga

Version 1.0

Copenhagen, 11 November 2019

>> *Sustainable. Agricultural. Innovative. International.*

# Content

- Precision Farming
- User story Farmer
- User story Paying Agency
- Object Model
- eCrop
- Application Architecture
- Integration Architecture
- Authentication

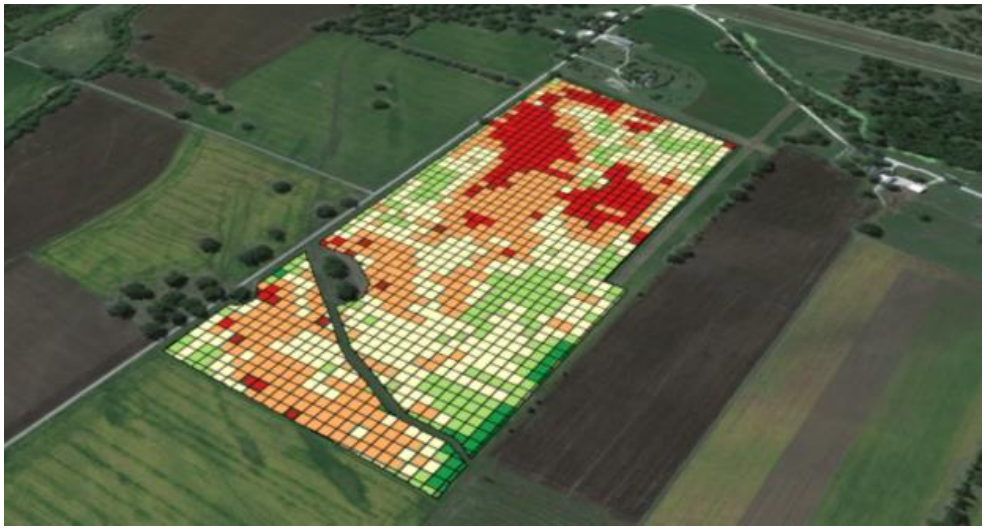
# Precision Farming

Task map =

Specification of the activities for the farming machine on the parcel. The farmer makes the task map on the FMIS

As applied map =

Logging of the executed activities of the farming machine, made by the farming machine during the rig of the parcel

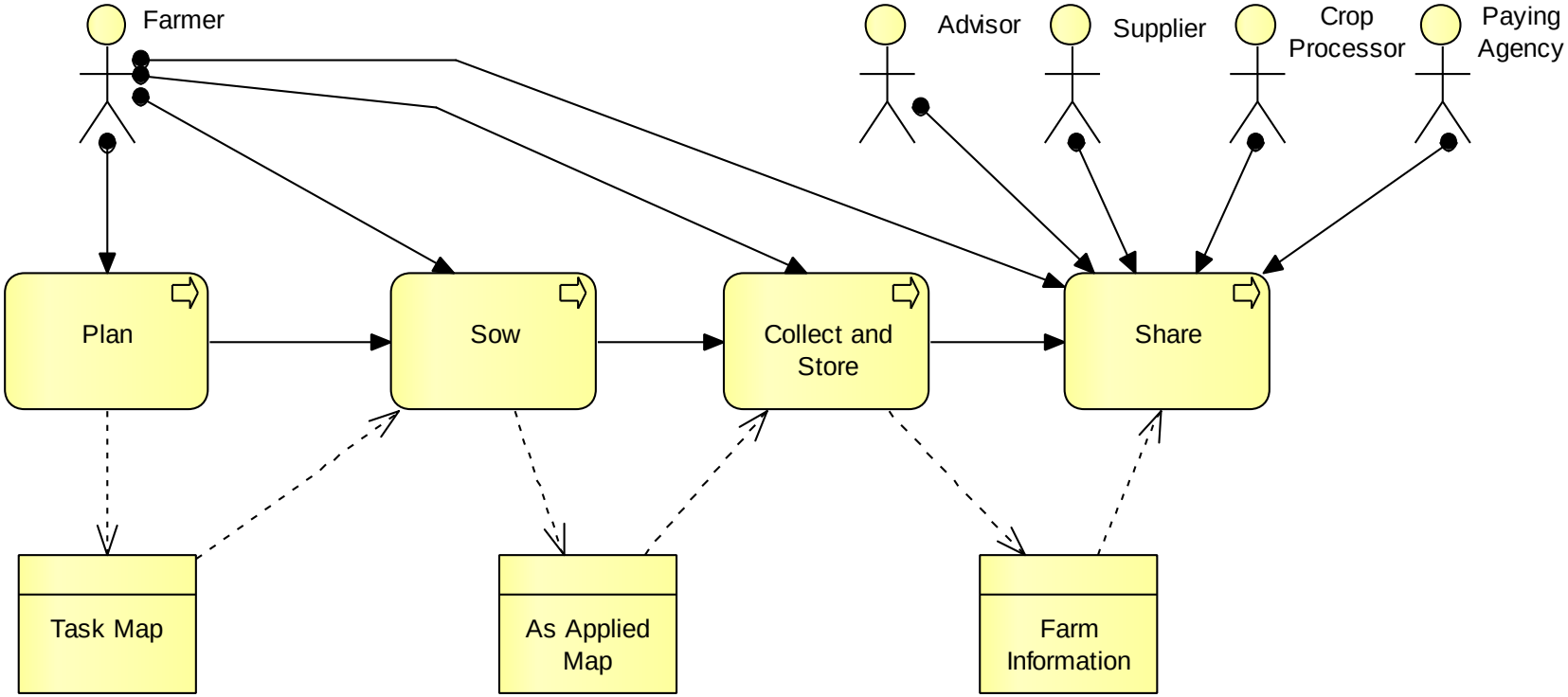


Task map

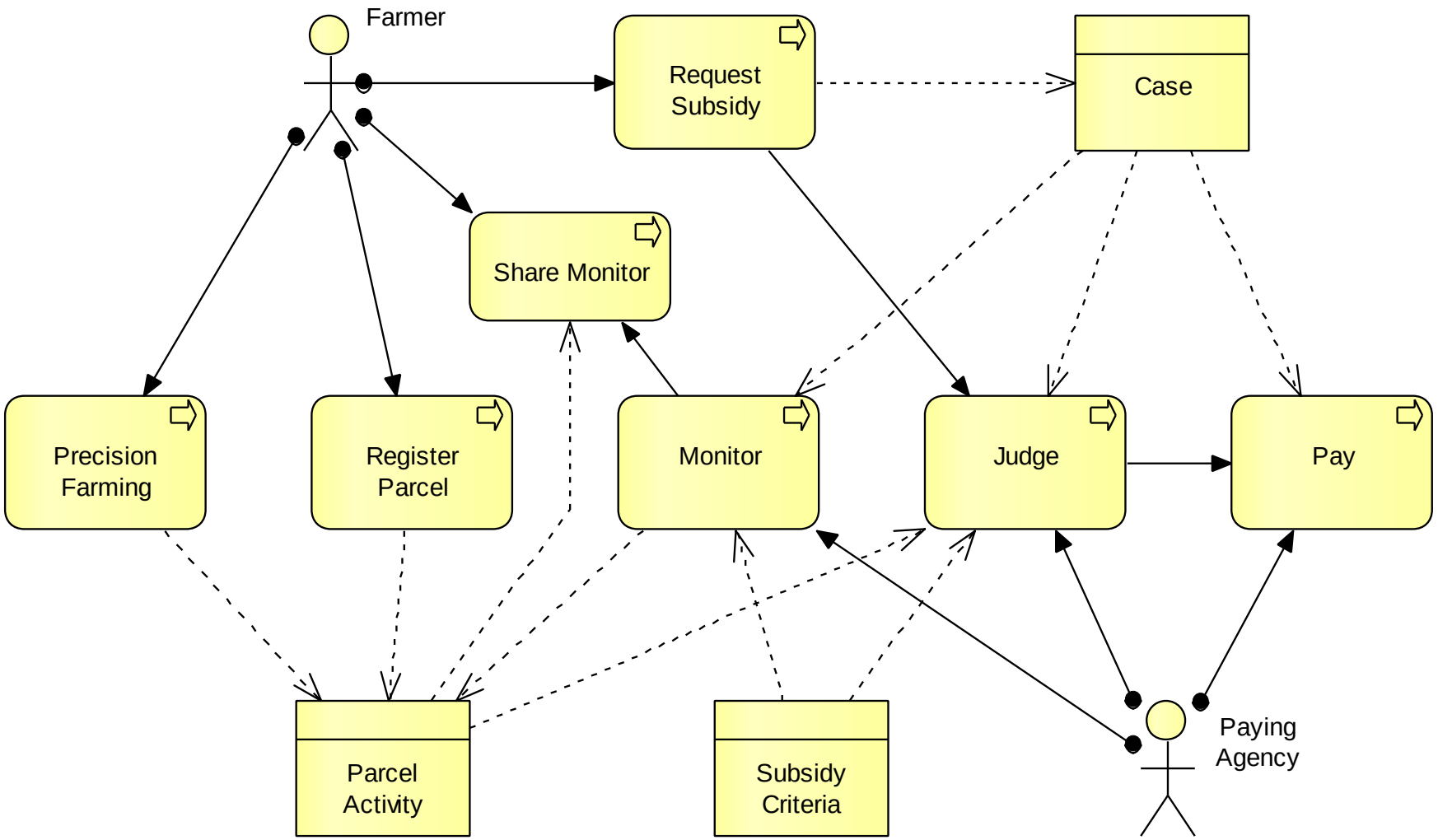
# Definitions

- The term parcel is equivalent to the term field. Naming varying between countries.
- A task map can be specifying a uniform task for the whole parcel (field). It is then called a Coverage map.
  A task map can also specify a spatially variable task within the parcel, and is then called a Variable Rate Application map, VRA.
- Correspondingly, an as applied map can be specifying a uniform task performed for the whole parcel (field). It is then also called a Coverage map, applied.
  An as applied map can also specify a spatially variable task performed within the parcel, and is then called a Variable Rate Application map, VRA, applied.

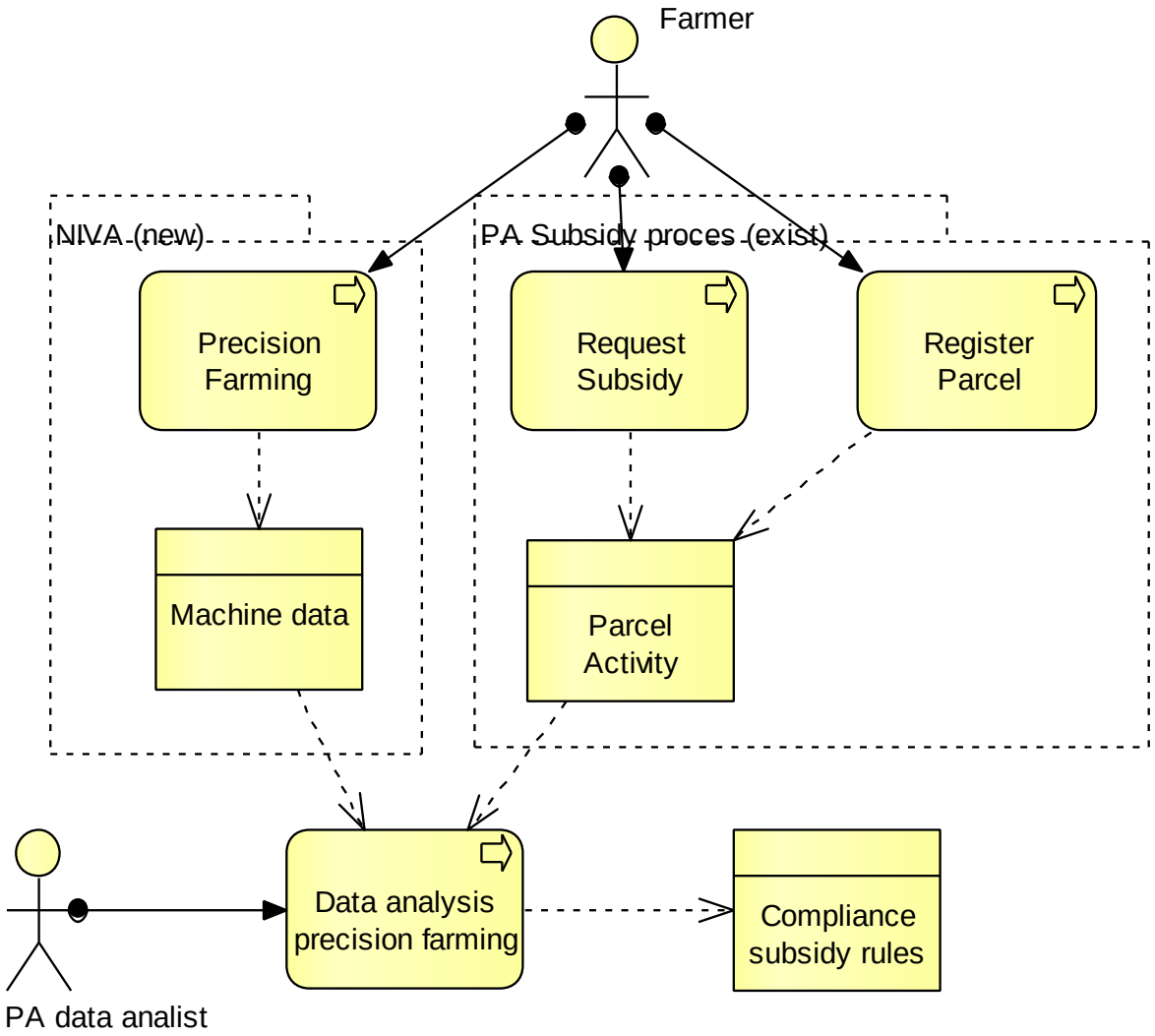
# Process Farmer



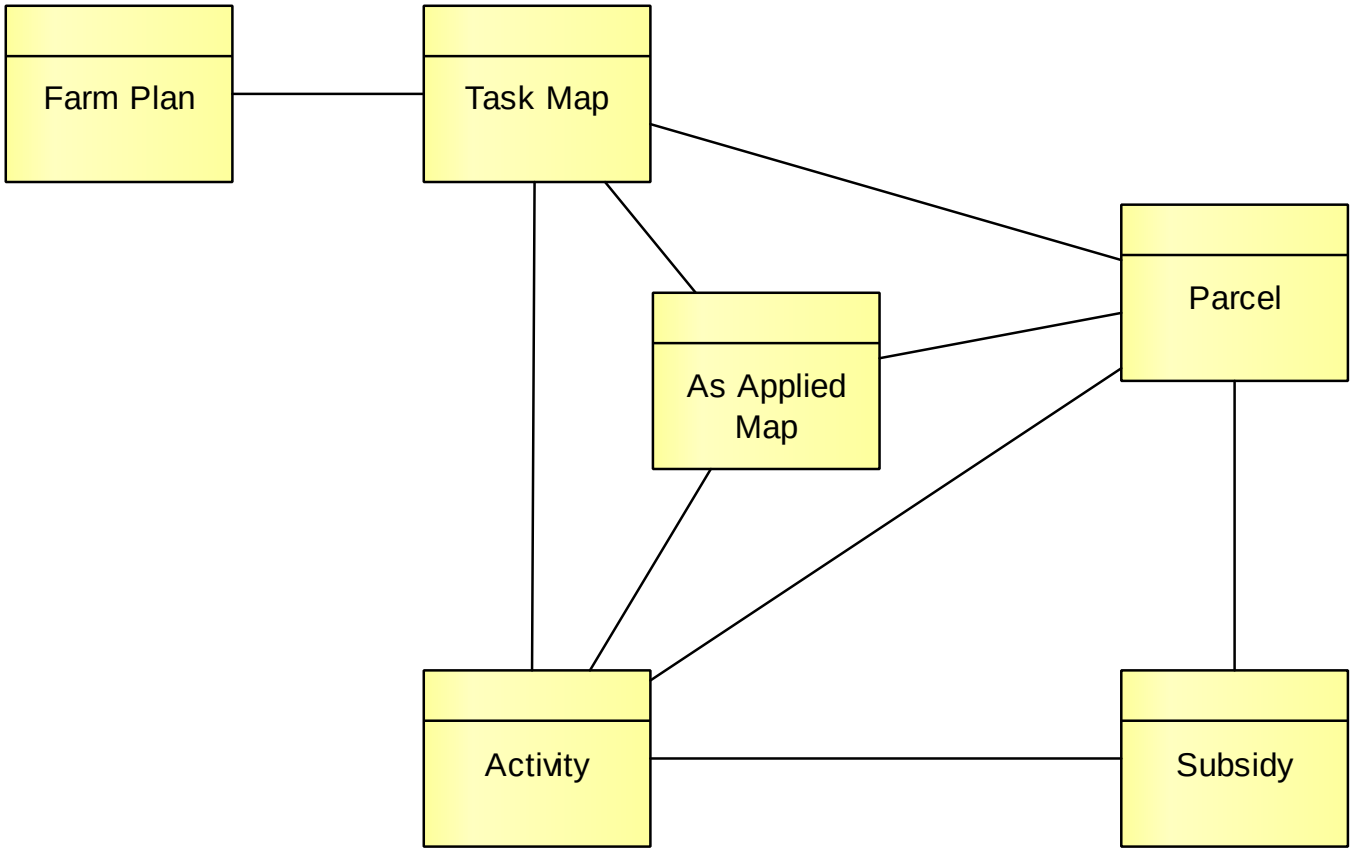
# Process Paying Agency



# Implement NIVA system at PA



# Object Model



# What data does PA want

Data is the origin:

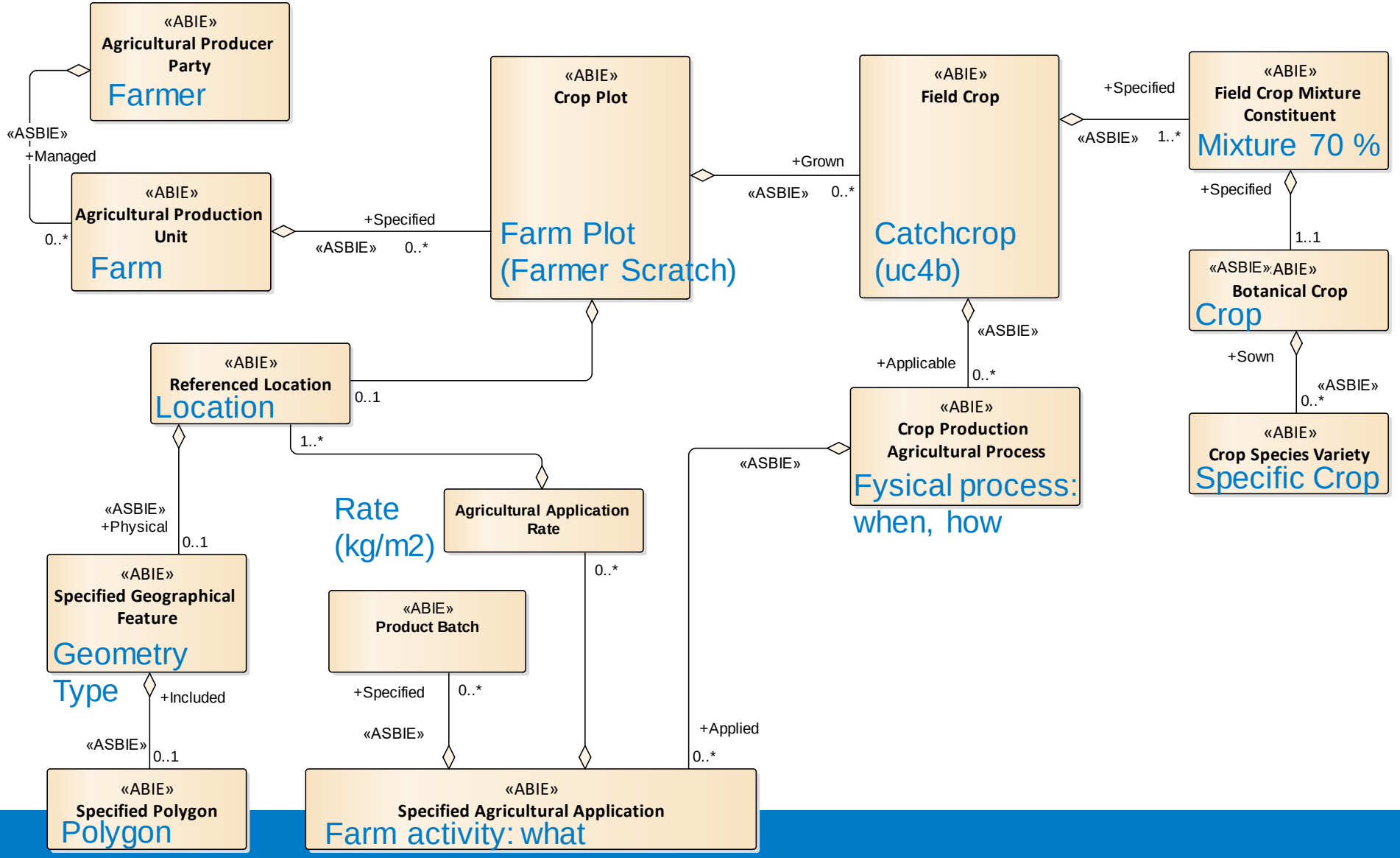
- as defined by the machine.

The processed data depend on "rules" from legislation.

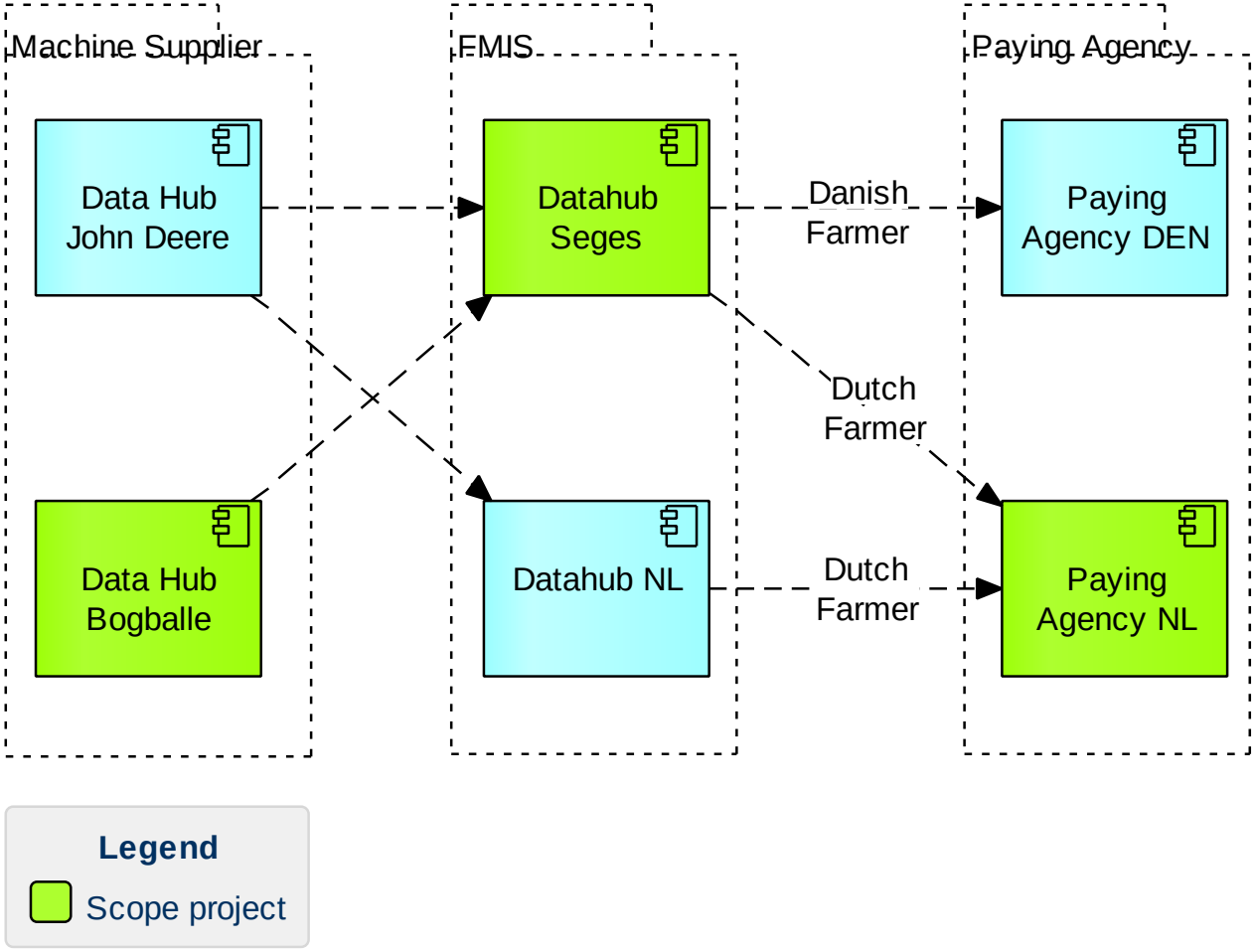
Data needed for assessing compliance with catch crop conditions

- Which parcel has been processed?
- Which crop (mixture) is sown?
- When was it sown?
- What amount has been sown per hectare?
- When is the field plowed / the crop converted?

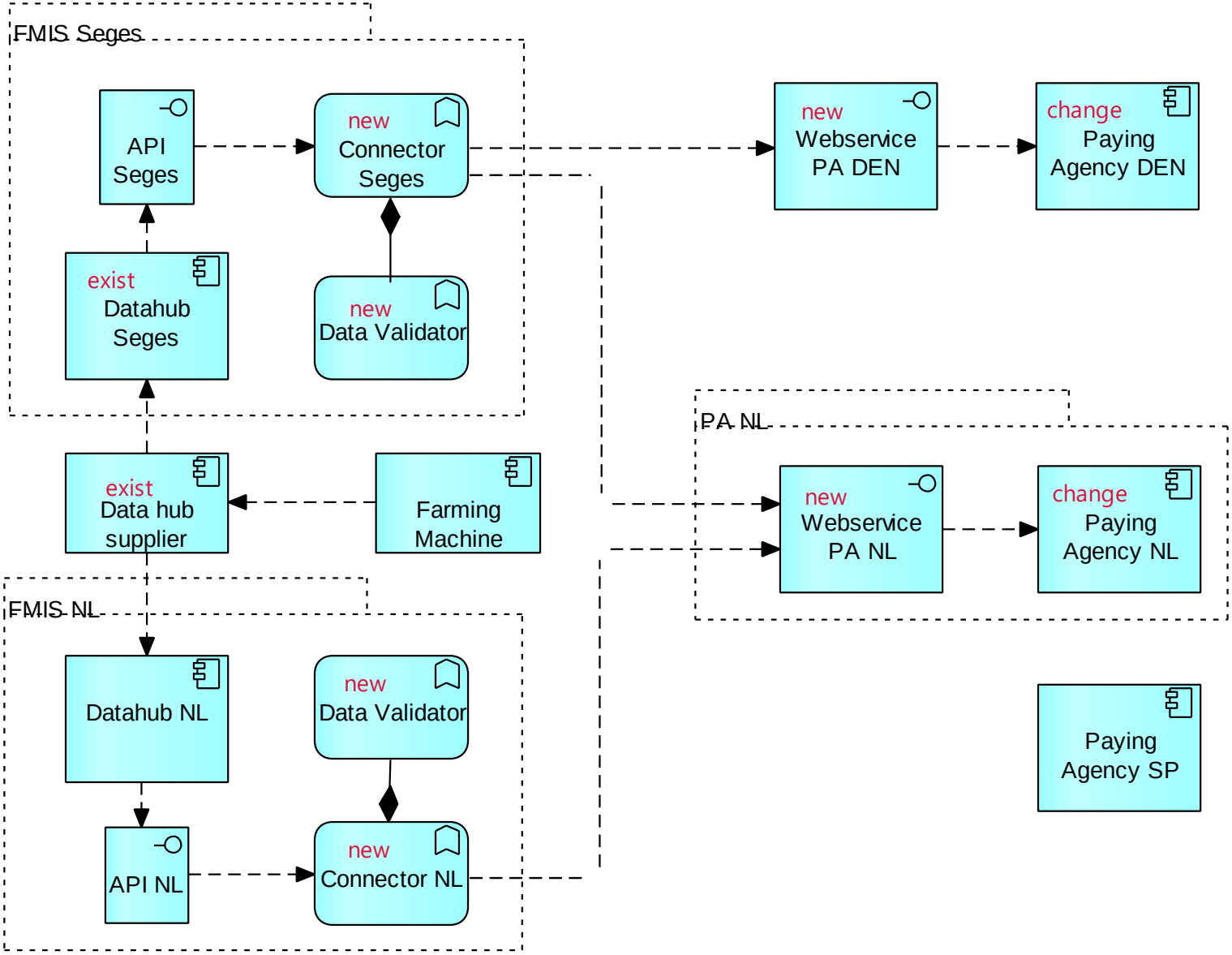
# e-CROP message UC 4b



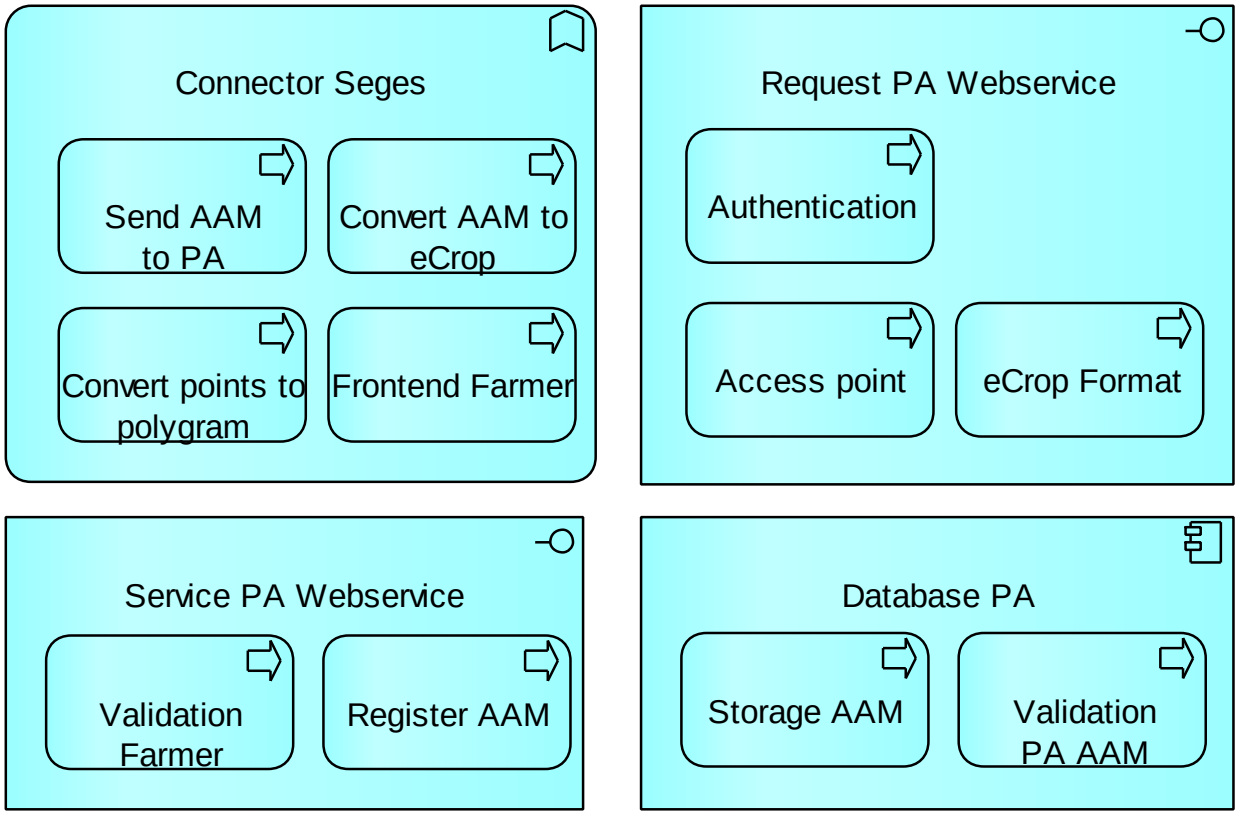
# Sharing machine data



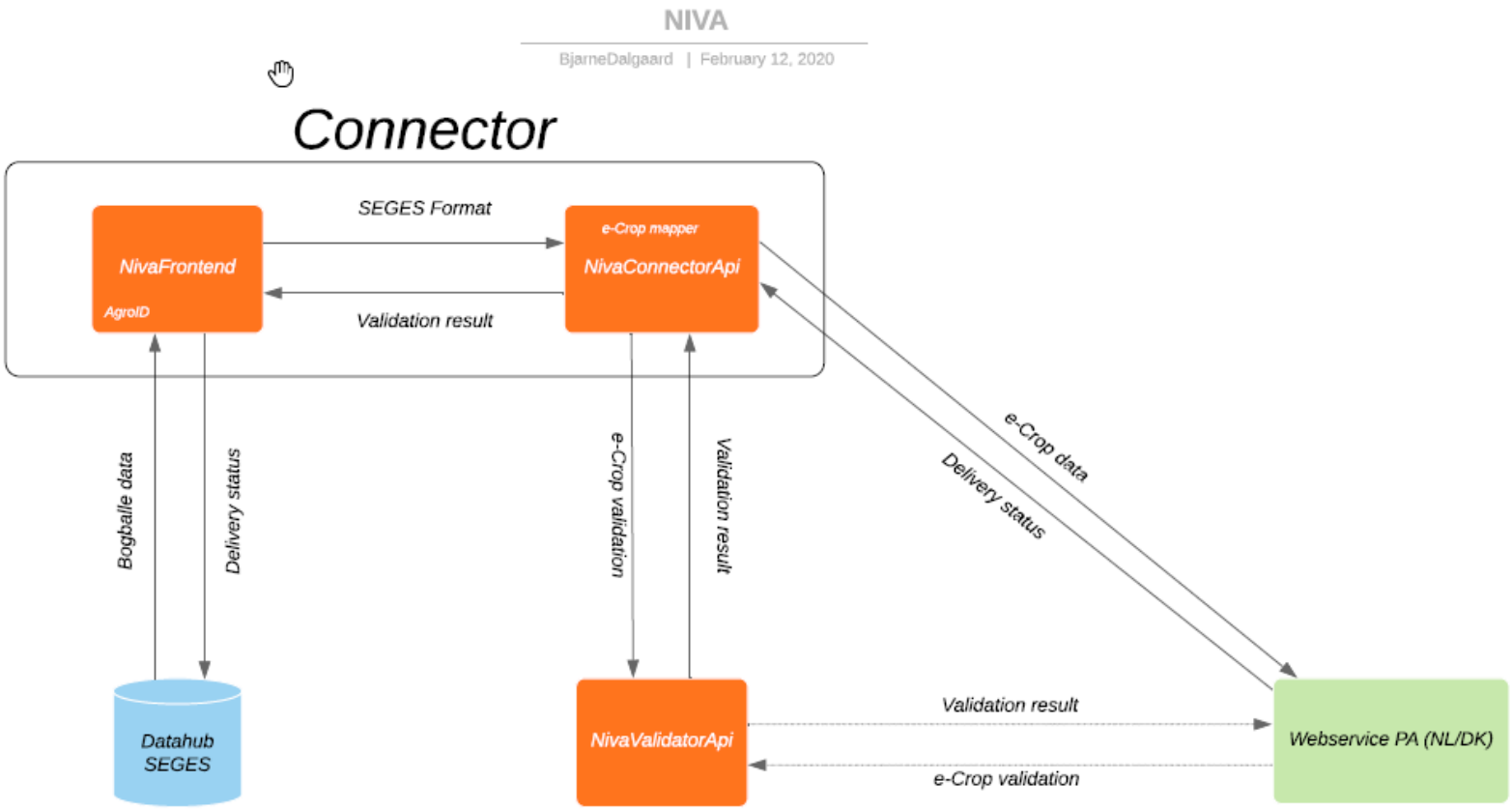
# Application Precision Farming



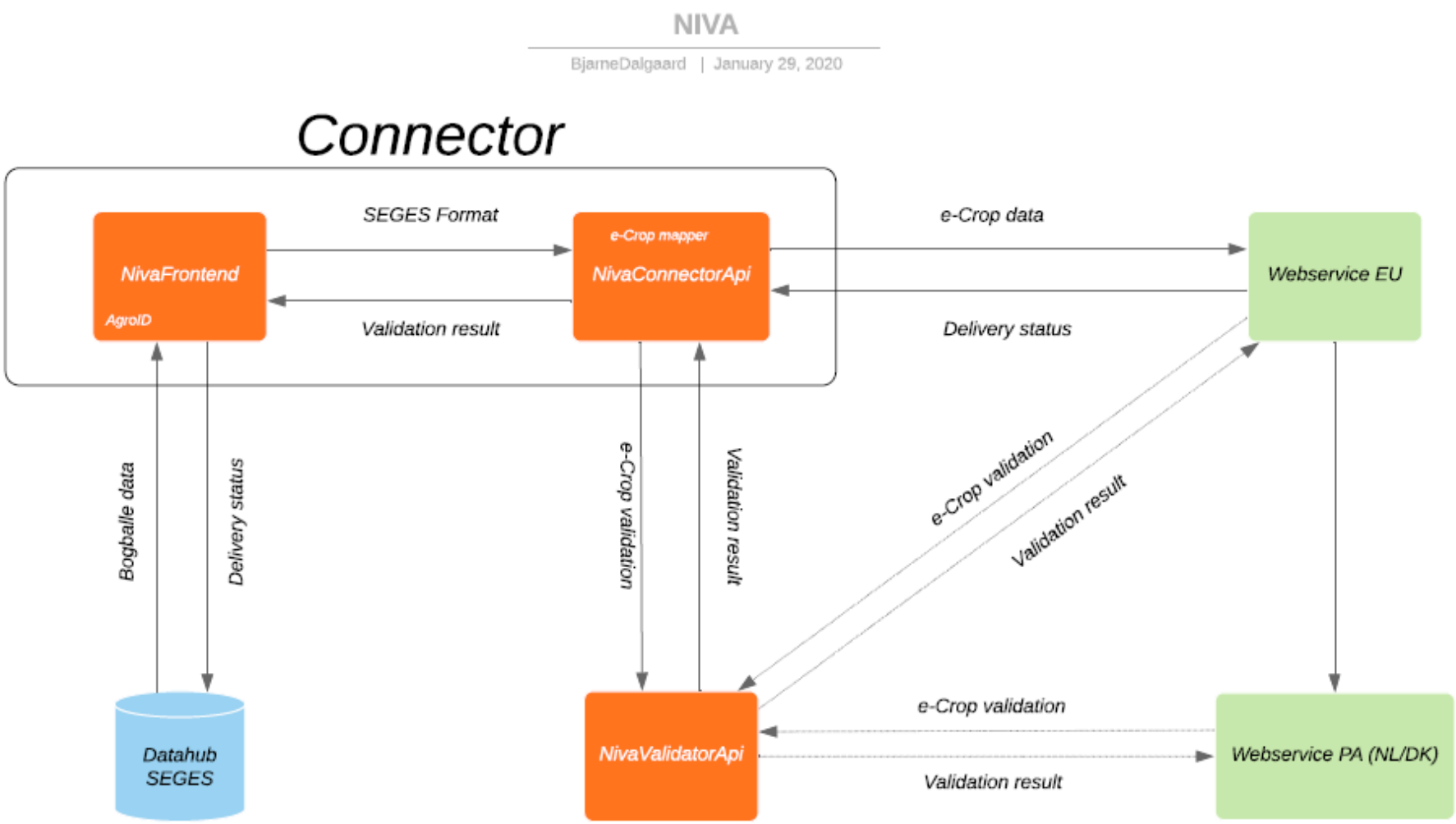
# Application Functions



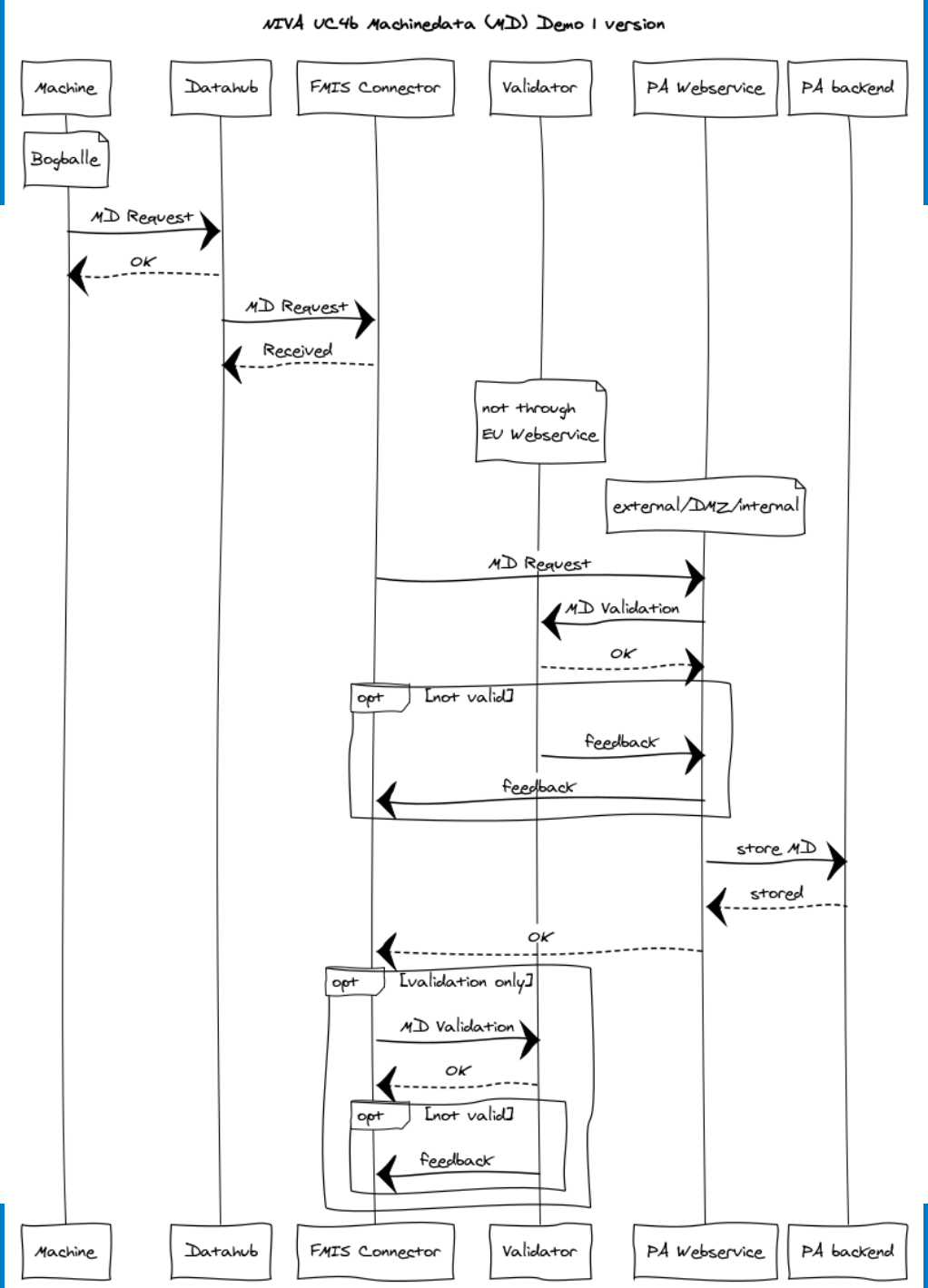
# Architecture/technical design (SEGES) - demo 1. version



# Architecture/technical design (SEGES) - demo 2. version

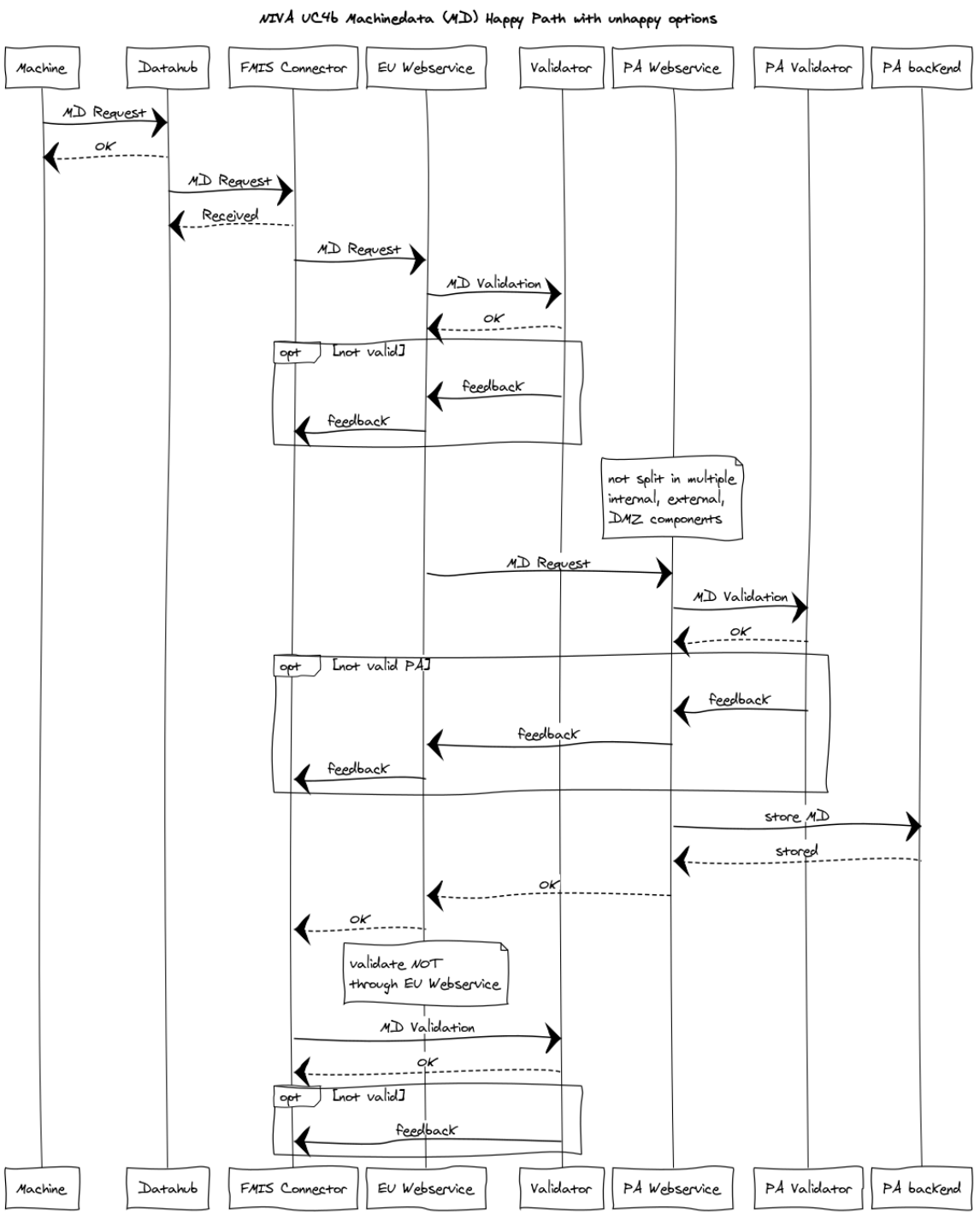


# Sequence diagram demo 1 version



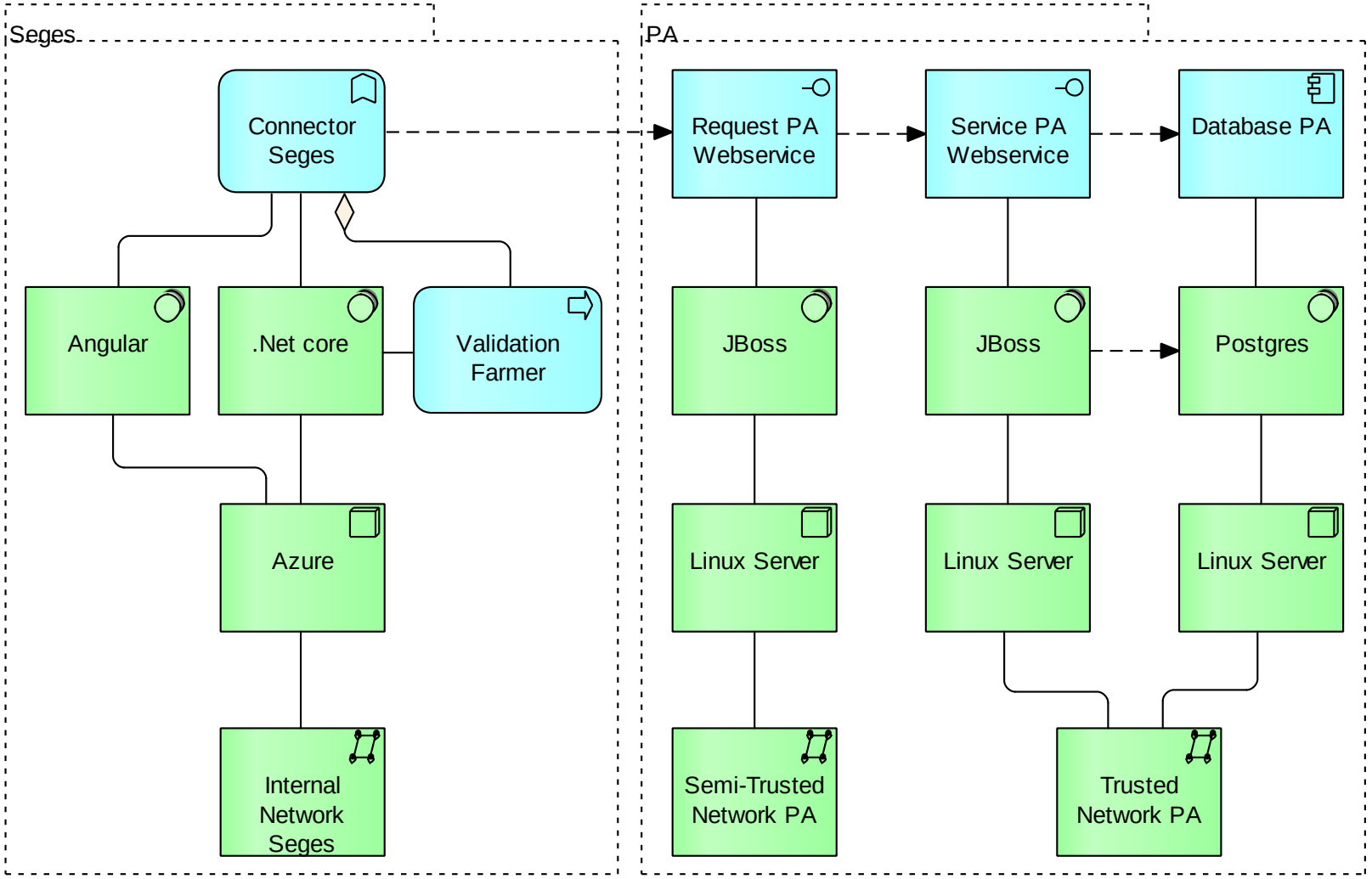
# Sequence diagram demo 2 version

**(not final)**



# Infrastructure

# Open source, NIVA demand



# Machine interfaces for demo NL/DK – candidates 2020

| Vendor | FMIS-SEGES (Create task) | Machine | FMIS-SEGES (Receive AsApplied map) | Candidate for NIVA AsApplied |
|---|---|---|---|---|
| Bogballe | ➢ Create task<br>➢ Send task | ➢ Read and execute task<br>➢ Create and Send AsApplied map | ➢ Read AsApplied map | ✓ Yes |
| John Deere | ➢ Create task and<br>➢ Send task | ➢ Read task and execute task Create and Send AsApplied map | ➢ Read totals and AsApplied map (GS3) | ✓ YES |
| Trimble | ➢ Create task and<br>➢ Send task<br>➢ (Q2 2020) | ➢ Read and execute Task | ➢ No reading 2020 | ❖ NO |
| AgriRouter | ➢ No tasks | ➢ Log yield<br>➢ Send yield map | ➢ Read yield map<br>➢ No application map 2020 | ❖ NO |
| Claas, CIH/CNH | ➢ Pipeline > 2020 | | | ❖ NO |

# Principles process

- The farmer sees in the connector the activity data which has to be sent
- The farmer shares activity data after the farmer approves the activity data in the connector.
- The farmer can send a request which only validates the message
- The farmer can not change the machine data
- The farmer sees in the connector the activity data which has been sent

# Principles application

- The connector is a specific component for each FMIS supplier, based on the open source connector of NIVA
- The NIVA webservice must be able to be used by all the members of the EU. That's why the software is open source.
- The validator must be able to be called by the connector.
- The validator must be called by the webservice of the PA.
- The paying agency has only access to validated messages. That's why the validator is not placed on the infrastructure of the paying agency.
- The webservice on the semi trusted network of the paying agency takes care of the authentication
- The webservice on the trusted network of the paying agency takes care of the logic
- The protocol used is REST/JSON
- The first release implements encryption on the transport layer
- The project implements a development and production environment

# Principles Authentication

- The identifier of the FMIS supplier is used for authentication
- The identification of the farmer is stored in the message
- In the first release/test the authentication is implemented by a username/password
- The proposal of UC 4b is to implement authentication by two way TLS (mutual TLS)
  https://en.wikipedia.org/wiki/Mutual_authentication
- Use of username/password is approved when the machine data is used for testing purposes
- The FMIS is responsible for authentication of the farmer
  during the access of the farmer to the connector

# Release 2 of UC4b

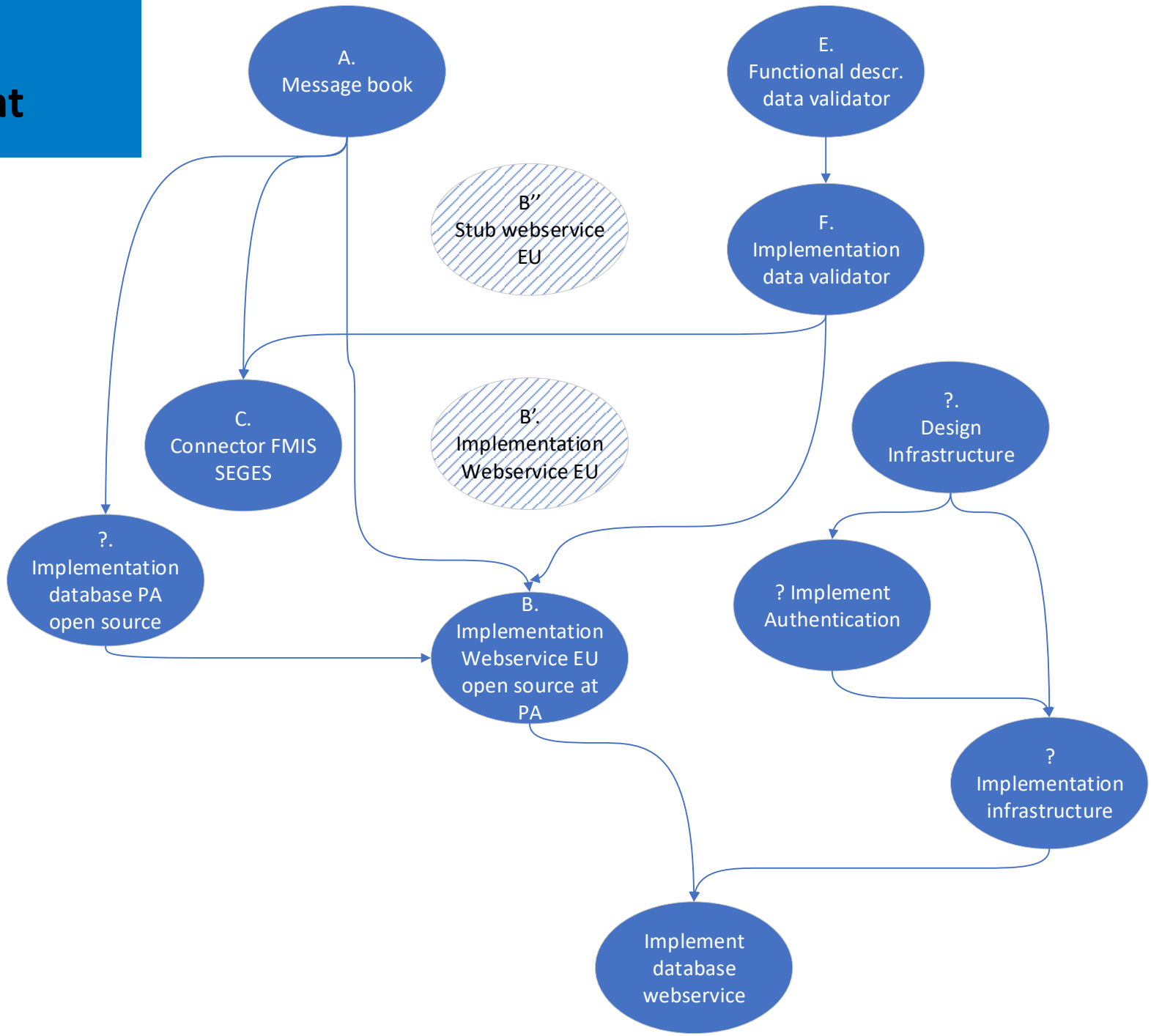
- Add relation from one “Crop Production Agricultural Proces” to many “Field Crop”
  Because there can be one activity to two Crop Plot’s

- Add Specified Agricultural Device (in relation to the
  Because this gives a reliability indication of the message

- Add Agricultural Input Product (in relation to the product batch)
  Because this describes the actual kind of seed which is used by the machine.

- Questions
  - Does the FMIS know the PA identification of the plot
  - Does the FMIS know the PA identification of the farmer
  - Does the FMIS knows to which PA the farmer has to send the message

# Alignment


A.
Message book
E.
Functional descr.
data validator
B''
Stub webservice
EU
F.
Implementation
data validator
C.
Connector FMIS
SEGES
B'.
Implementation
Webservice EU
?.
Design
Infrastructure
?.
Implementation
database PA
open source
B.
Implementation
Webservice EU
open source at
PA
? Implement
Authentication
?
Implementation
infrastructure
Implement
database
webservice

# Product List

- Message Book (e-crop standard EU for: as applied map) - 1
- Webservice EU for sending and receiving: as applied map - 2
- Connector FMIS (Seges) – EU webservice - 3 - Seges
- Connector FMIS (Dutch FMIS) – EU webservice - 3 - Dutch FMIS
- Functional Description of: Data Validator - 2 - Seges
- Implementation Data Validator - 3- Seges
- Webservice PA NL - 5- Dutch PA
- Webservice PA DEN ? - 6- Danish PA ?
- Use of authentication NL (e-id) ? - Dutch PA ?
- Use of authentication DEN (e-id) ? - Danish PA ?
- End-to-end testing: Machine –> FMIS -> PA - 10 – PA-NL-DEN

# Translation table(s) for codelists.

- How to get FMIS's and PAs to talk to each other.
  - Codelists are, or can be, unique for PAs, FMIS's and/or countries. In order for a pan-european system to function, it will be necessary to be able to translate from one code list to another.
  - Codelists can be translated using one, or more, translation table(s).
  - For the purpose of the POC at hand, a simple flat file is proposed for storage.

# Proposed format for translation table.

| EntityID(*1) | CodelistTypeID(*2) | Party | Code |
|---|---|---|---|
| 1 | 1 | SEGES | 123 |
| 1 | 1 | LBST | 456 |
| 1 | 1 | RVO | 789 |
| 1 | 1 | ZLTO | … |
| 2 | 1 | SEGES | … |
| 2 | 1 | RVO | … |
| 3 | 2 | SEGES | … |
| 4 | 12 | ZLTO | … |

*1: Entities are freely numbered. No combination entityID, codelistTypeID and party must be repeated.
*2: Codelists must be negotiated and defined. If a translation table per code list type is desired, the codelistTypeID can be omitted.

# Code list translation usage

1. Determine codelistId, source party and source code. Look up entityID based on these.
2. Reverse loopup using entityID, codelistId and destination party. Look up destination code based on these.

An editor must be created to allow agencies (FMIS / PA) to create and edit entries in the translation table(s). This is outside phase 1 scope.