090-ISU System Documentation

Concerning the AM object 090-ISU

Dnr 2016-Apr-22

Version 1.6

FINAL

Authors

	Name
Author	Subhashini Prasad Babu
Expert	Magnus Värmfors
System Manager	Robert Singh Leisangthem

Changes

Version	Heading no	Description	Date
0.1		Initial Draft	09-Dec-2011
0.2		Implmented Niklas comments	18-Jan-2012
0.3		Implemented Rakesh Comments	1-Feb-2012
1.0		Implemented Tapani's Comments	28-Feb-12
1.1		Implemetned Tapani's Comments	08-March-2012
1.2		Updated with Use cases and realizations	22-March-2012
1.2	3.5		17-April-2012
1.3		Updated as per review comments given by client	06-Dec-2012
1.4		Updated as per review comments given by Robert Bjervas	17-Dec-2012
1.5		Added a reference to the AM handbook	27-May-2013
1.6		Updated Header and changed Expert, System Manager	22-Apr-2016

Review History

Object Manager/Domain Manager	Review By	Review Date
Object Manager	Sambireddy Bhimavarapu (Earlier OM)	27-May-2013
Architect (CGI)	Robert Bjervås (Earlier Architect)	27-May-2013
Architect (PN)	Tapani Kinnula	14-Feb-2012
Service Object Manager	Mårten Carlberg (Earlier SOM)	27-May-2013

Table of Contents

1 Purpose

The purpose of this document is to serve as an introduction and a foundation for working with an object, and the applications under that object. The aspects not applicable to this object are marked N/A (Not Applicable).

2 Applications

Application ID	Name	Brief Description
1	ISU	ISU is a data repository of all post boxes, post offices and residents address

3 System description

History

ISU was implemented in 1995. The database was created from the information about the addresses and residents. The complete postal organization had been taken into the picture.

3.1 Objectives

ISU is the main system for supporting the postal business at mail delivery offices and working routines for mail distribution to recipients' mail boxes. It also provides statistics on distributed mail volumes on both addressed and unaddressed direct mail (UDM). To this purpose, ISU contains information about delivery offices, all Swedish addresses, person and organisations living/located at these addresses and their address change history. ISU also provides tools to deal with mail delivery routes, mail distribution statistics, bicycles used by mail men, local services, redirection of mail and business reply mail.

ISU is also considered as an master data system for Swedish addresses and delivers address data to several other systems supporting Swedish mail production, for example machine sorting of mail.

3.2 Requirements

3.2.1 Use Case Overview

Some ISU use cases are available **here**

Or

https://postennorden.primeportal.com/am/Meddelande/Pages/090-ISU.aspx#/1012-Documentation/10129-Other_documentation/

3.2.2 Supplementary Requirements

ISU expects, some 50 to 60 active sessions of the ISU users at any given time.

Due to some performance degradation while fetching out some huge reports, materialized views have been implemented

3.2.3 Reports

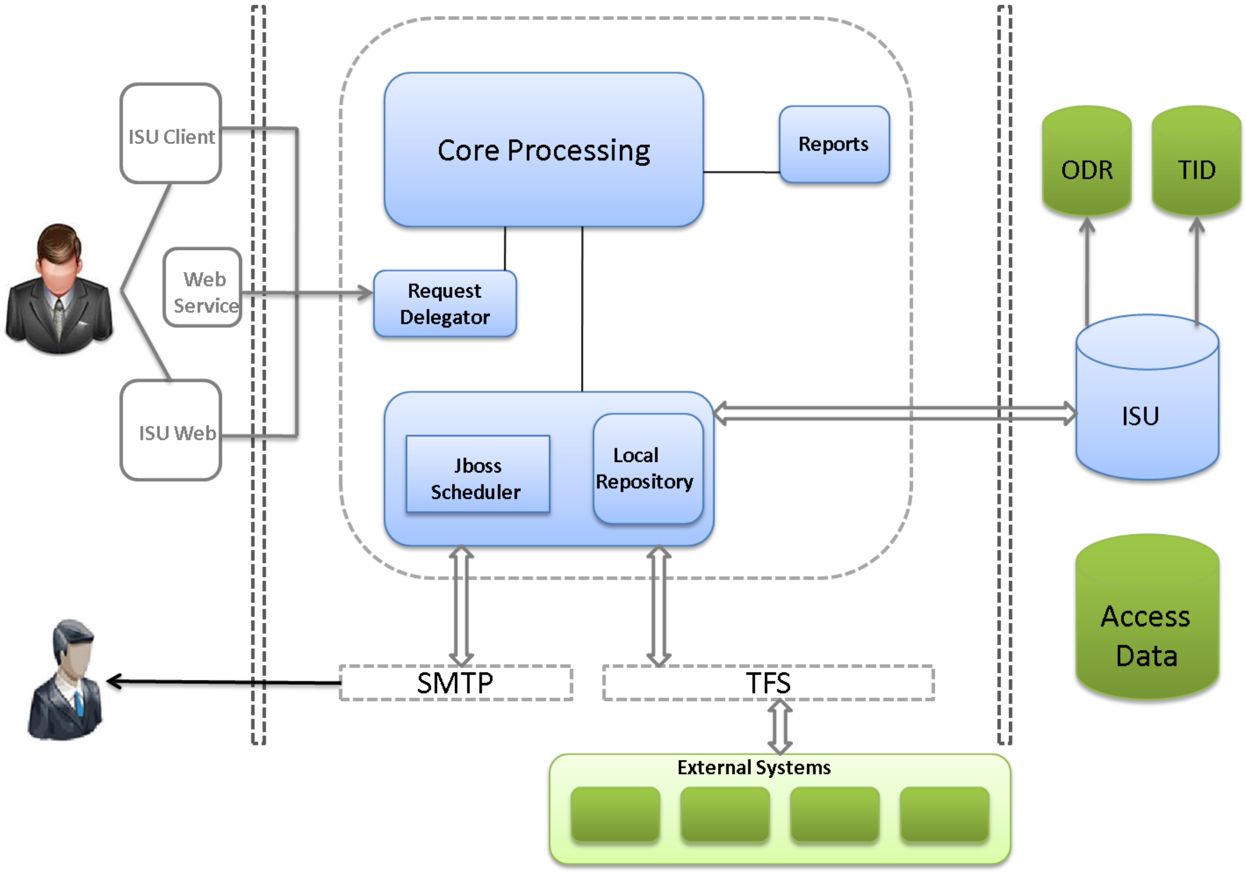
ISU can generate more than 35 types of reports, some can be extracted from the ISU Desktop client and some can be generated from ISU Web interface.

Some report names and their description are available on **define** or on Prime Portal

3.3 Architectural views

3.3.1 Logical view

This diagram explains the overview of the ISU application.



Client

ISU has 3 different types of clients.

ISU Swing Client
Web service client
ISU Web

ISU Swing Client

Out of these, ISU Swing client is the desktop and standalone GUI. It can be provided to the customers as an executable file. Means, we can run this client as a desktop application. To start with this client, sufficient authentication is needed. It can be run only from Posten environment. (I.e. VDI or Citrix)

As user starts up this application it will contact the central database and application servers of different environments on his choice. So no data will be stored in the local machine, all the database related activities will affect database directly.

Web Service

Some of the external systems make a web service call to contact ISU.

For example: Posten.se
Eniro
Telefonboken

In this case data won't be passed through TFS. It would be kind of a direct communication.

More details are available **Define** or on Prime Portal

ISU Web

This is a web client intended for people both inside and outside of Postnord giving them access to information in ISU without the need of installing the application. Only a limited set of functionality is available in the web interface. Most of the functionality in ISU-Web is not available in the ISU Swing client. ISU-Web can be accessed from ISU from the menu giving the user of ISU easy access also to the functionality in ISU-Web.

Some of the tasks that can be done from ISU-Web are:

- Printing of a large set of reports
- Registration of newspaper delivery complaints
- Registration of jobs to be run
- Registration of business reply mail
- Search for residential history, delivery office and business reply mail zip

- Registration of post office box contract
- Access to information about internal company mail distribution

Middle Tier

ISU middle tier (ISU-MT) is responsible to

- Run the batch jobs to contact other systems.
- Generate reports
- Respond to the web service calls and
- Contact the database
- Provide business services for ISU clients

ISU-MT runs the batch jobs based on the job scheduler. Most of the communication will be done through TFS (Posten's FTP Bridge). There are daily, weekly, monthly and more often in a day jobs are active and running as per the scheduled time automatically.

External systems send the data to TFS and TFS will push the data to ISU as files. The files will be stored in a predefined location in the application server. Then the various batch jobs will run and process the data as per the desired functionality. All files from TFS are transferred by an HTTP request from TFS to a servlet in ISU-MT.

ISU-MT publishes a large set of EJB-services who are used by the ISU clients to get access to information in the database. Most of the business logic is located in the EJBs but not all. Some business logic is implemented in the database and other in the clients. The database access is based on a O/R-mapping framework and implemented as DAOs.

ISU-MT must always be up and running in order to respond to web service calls and EJB calls. Web service mode of communication is not being used to an extent level, but for some systems. The information about the reports has been given in the Reports section.

Database

ISU database is the core part and is the key repository of data. It mainly comprises the data regarding

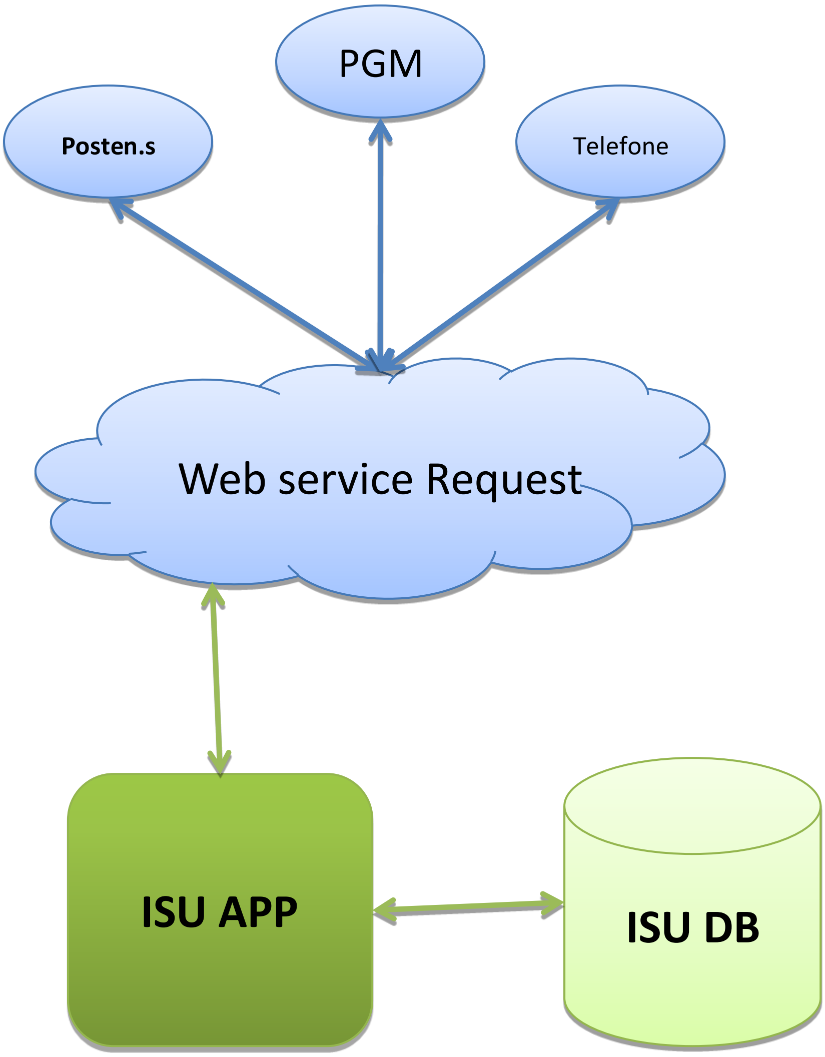
- Address information
- Delivery Information
- Change of Address

- Post Office Details
- Delivery statistics

ISU database maintains the most reliable and genuine information, which will be provided to some of the government organizations also, mean to say, that much critical data, ISU is handling.

3.3.1.1 Service View

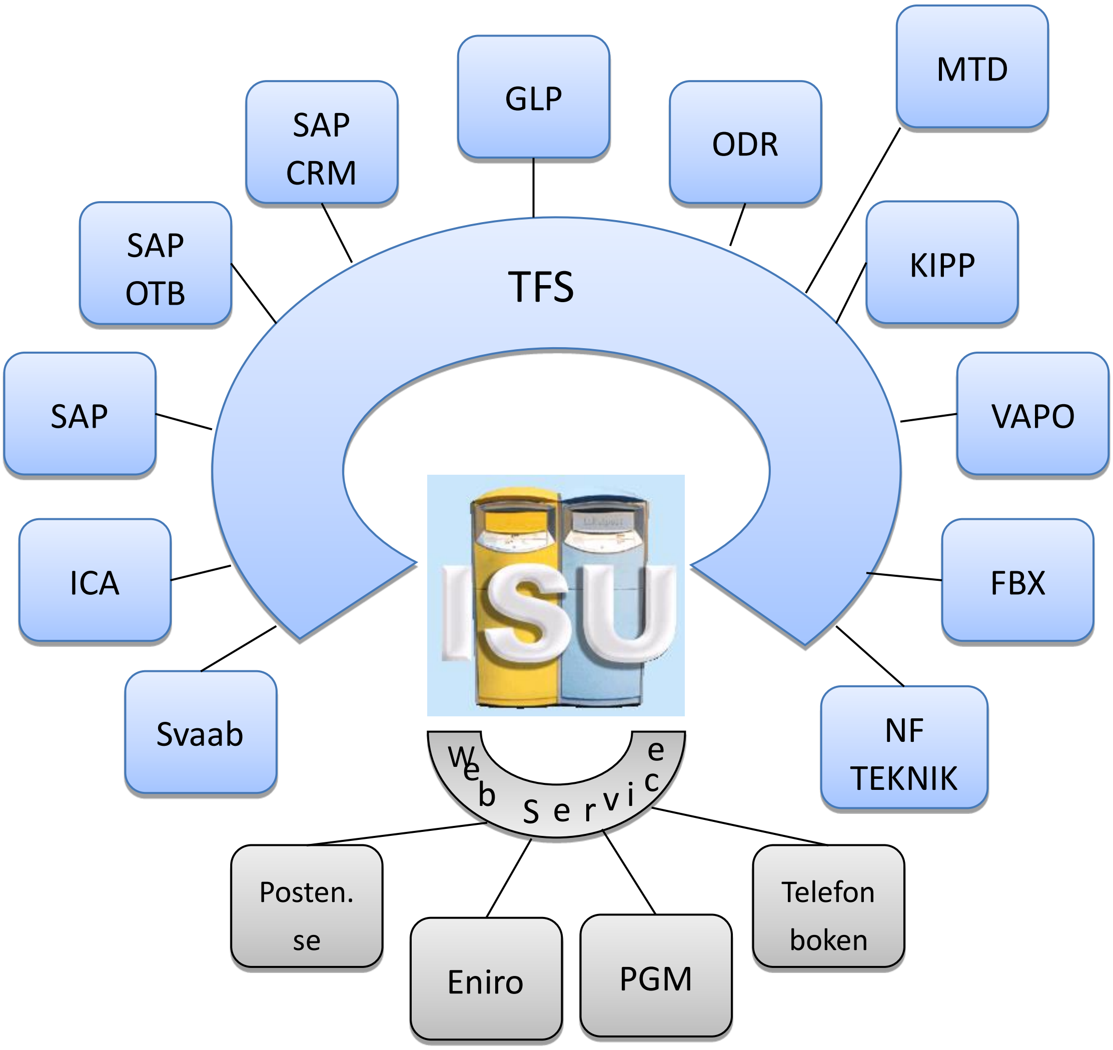
The following diagram mainly focuses on the web service calls to the ISU. How they are being handled, is described in **Define** or on Prime Portal



3.3.1.2 Outside view

This section describes about the communication and relation with external systems, from ISU perspective.

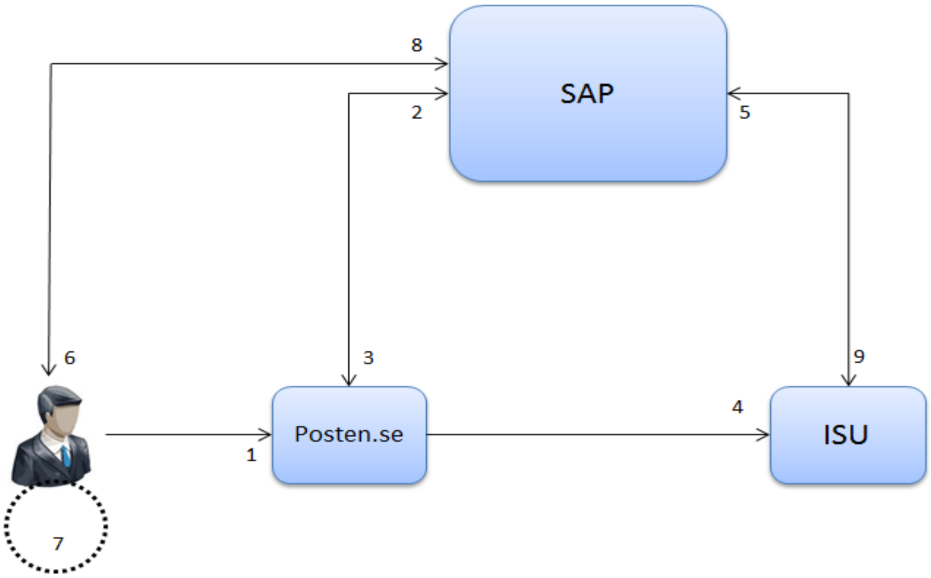
As we already discussed in the early section, main mode of communication would be through TFS or via web service calls.



This gives an overall idea on, how is ISU communicating other systems. Let us discuss about each and every system in detail.

SAP

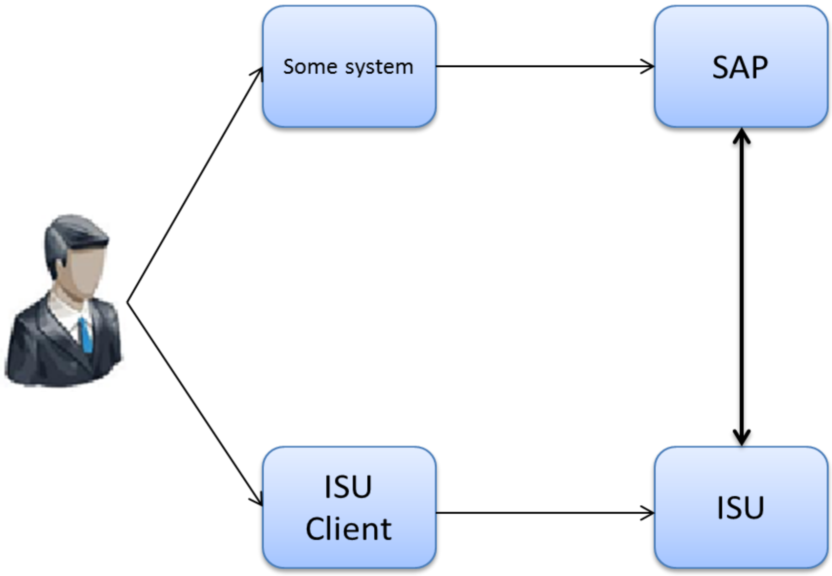
ISU and SAP relation deals with the customer's Post box agreement.



1. Customer contacts Posten.se
2. The customer has to be registered first in SAP.
3. SAP sends back the registration info to Posten.se
4. Posten.se contacts ISU.
   ISU checks whether the Post Box is available or not.
   If it finds any, that PB will be mapped to the customer.
5. ISU sends the data to SAP.
   SAP will cross check the customer data.
   Post box agreement will be created.
6. SAP will send the agreement details to the customer along with the invoice.
7. Customer will pay the amount.
8. Payment details will be sent to SAP.
9. SAP will update ISU and ISU will fix the box to the customer.

SAP CRM

ISU and SAP CRM relation deals with the cancellation of the Post box agreement



SAP CRM sends ISU, the information about which post boxes are paid by the customer, and can be marked as ”rented” or assigned in ISU, and about cancelled agreements so that ISU can mark them as vacant.

Finally SAP and ISU should be in synch with each other about the cancellation details.

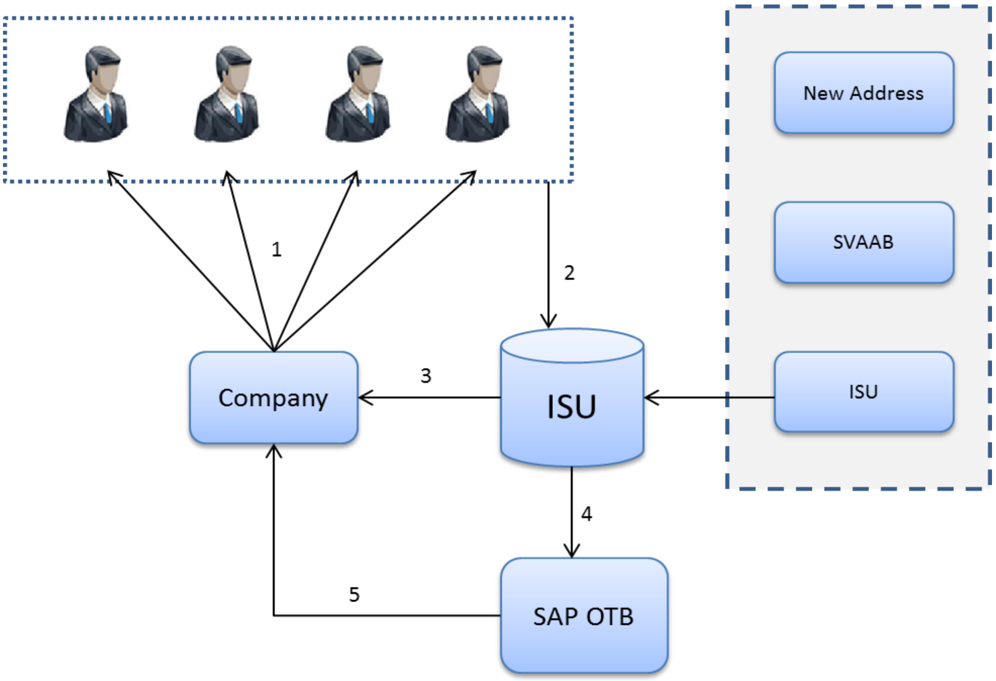
SAP Kundmaster/ SAP Customer Master

The customer data will be uploaded or updated to “**kundmaster**” table in ISU with the frequency of every minute.

KIPP

ISU sends the post office and company customer details as a flat file to KIPP.

SAP OTB



The normal flow will be like following

Company will send offers to its customers and customers can choose to reply with business reply mail so that the company will be charged for the postage.

1. Number of replies will be sent to SAP system and it will send the invoice to the appropriate company. (We didn't mention the internal mechanism)

   If any customer changed his address then,

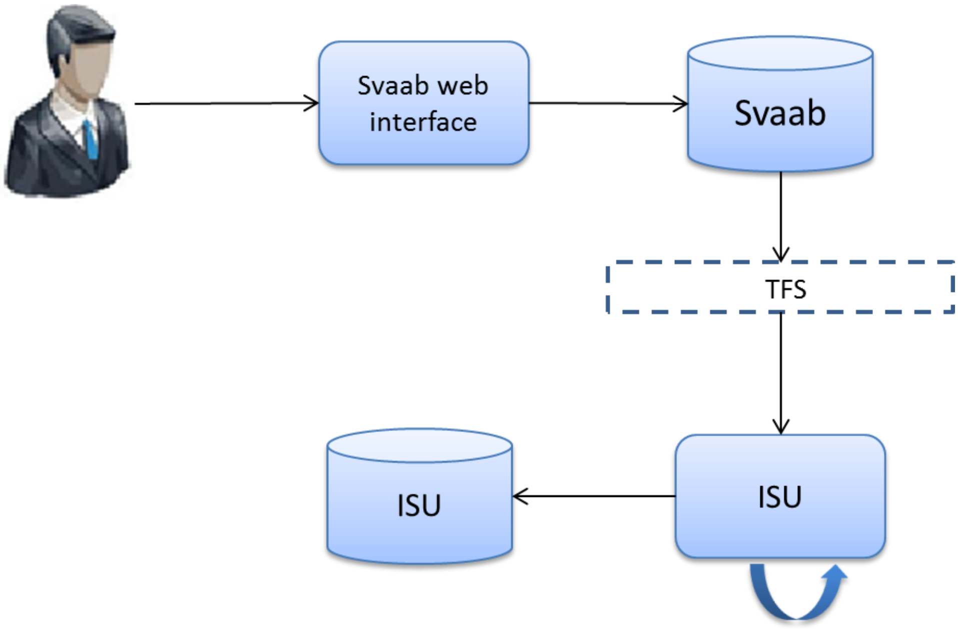
2. Change of address details will be there in ISU
3. ISU will update the company's database with the new address.
4. Number of updates will be sent to SAP system.
5. SAP system will generate invoice and send it to the company.

The following is the main process in brief.

Billing a Business Reply customer (the company) for produced Business Reply letter volumes.

Informing a Business Reply customer (company) with address changes among their customers and billing them for that service

SVAAB

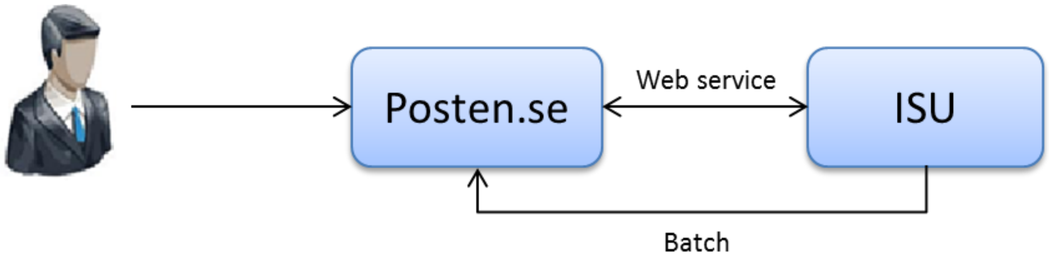


If any resident is going to change his address, he will update his new address in SVAAB. SVAAB will send the Change of Address details to TFS.
TFS will push the data to ISU.

Batch job runs on ISU server and checks whether any data from SVAAB is available or not. Once finds, data will be uploaded into ISU Database.

ISU sends data to SVAAB, weekly.

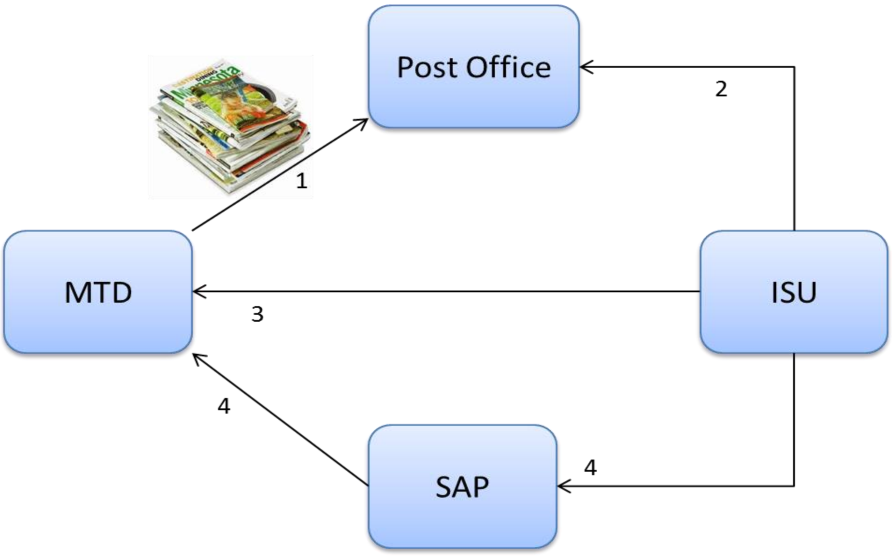
Posten.se



Posten.se will make a web service call to ISU to get the post box related information.
Technically, Posten.se does address lookups against a web service in ISU, while the post box installations are sent to Posten via file deliveries.

Alternate thing is, ISU sends data to Posten.se daily through a batch job.

MTD



1. MTD sends magazines to post office along with the receiver details.

2. Post man will get the information from ISU and prints the address on each article.
3. ISU updates the MTD with the number of deliveries.
4. ISU updates the SAP system with the number of deliveries.
5. SAP system will generate the invoice and send it to MTD.

TAB

The data to be sent to TAB will be updated in to TAB schema in ISU database. Updated data will be delivered to TAB, on every weekday.

PSS

ISU receives the number of volumes sorted at terminals, in a text file, on every 1st business day of a month.

ISU also sends addresses to PSS, as well as information about delivered letter volumes

GLP

ISU will send the following information to GLP

Change of Address,
Sequence,
Street and
Box

In total 4 files will be sent and 3 batch jobs will be take the responsibilities of this

More details about ISU and GLP relation, is available in ***define*** and on Prime Portal

Eniro Kundjanst

Eniro is a company that holds the telephone details of each and every company and as well as individuals of Sweden. Eniro sends the telephone directories to Posten which will deliver to the companies and individuals of Sweden.

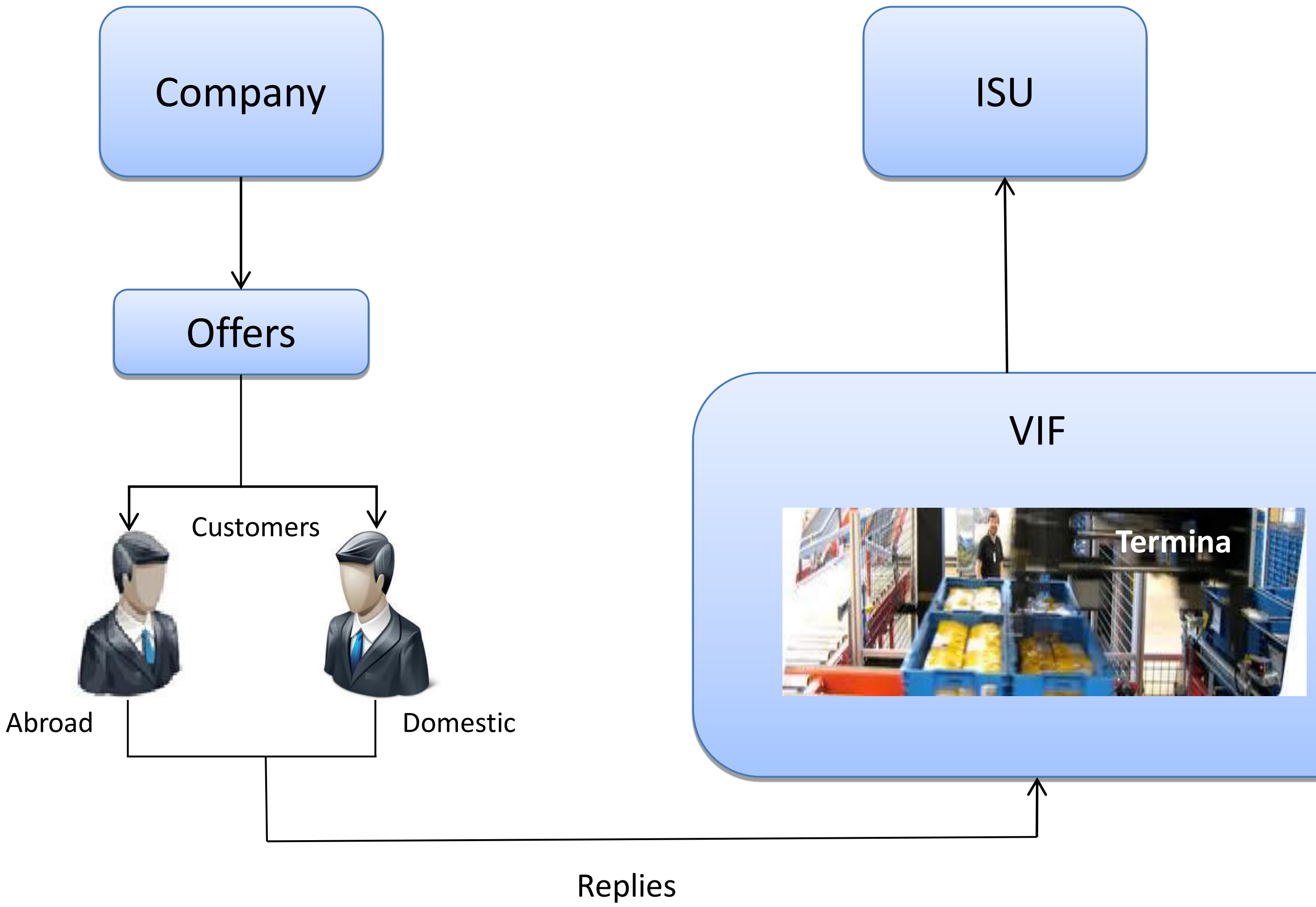
Eniro, also makes a web service call to ISU to get the information about the post offices and the residents and also can register subscribers to phone directories via the web service.

Telefonboken

Telefonboken requests ISU with a post code.
ISU responds back with the information about post office name, post code and the finance department number.

NF Teknik

NF Teknik is a company maintaining the VIFTP application which sends data on volumes on business reply mail to ISU on monthly basis. This data is then used for billing in SAP.



Companies send offers to their customers. Customers may stay in Sweden or abroad. If any customer wants to grab the offers, they will send a reply to the company.

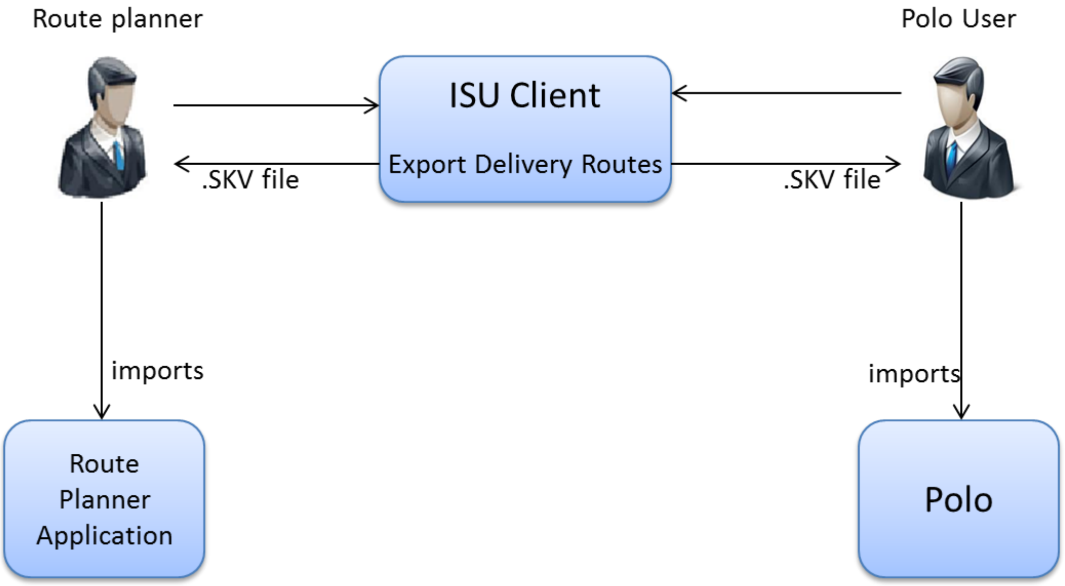
Received Business Reply letters are counted at sorting centres (terminals) in the VIF solution. VIF sends daily Business Reply volume information to ISU for production billing information to SAP.

AD (Active Directory)

ISU user details are stored in ISU database and/or LDAP server. When an ISU user logs in to the ISU application, the login credentials will be checked against the database first.

If the login Type is ‘T’, credentials will checked against ISU database and if the login type is ‘A’, then the credentials will be checked against the LDAP server that is nothing but AD (Active Directory).

Route Planner/Polo

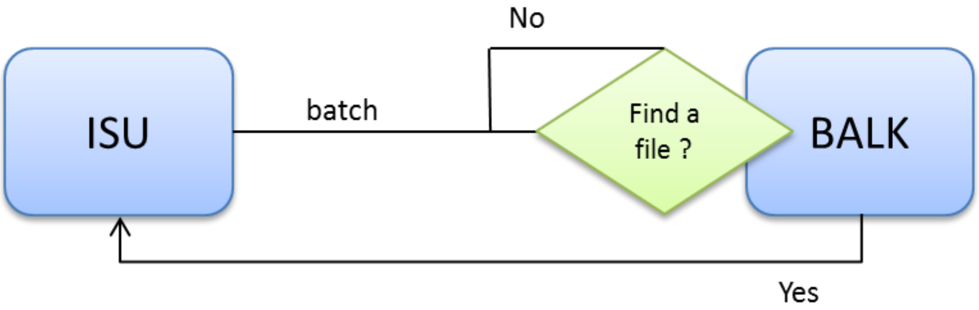


Route planer and Polo applications will contact ISU manually.

Route Planner officer will export the data from ISU, in a .skv file format and loads it into Route planner application.

Polo application also seeks for the data from the same kind of .skv file, but it contains less data compares to Route Planner.

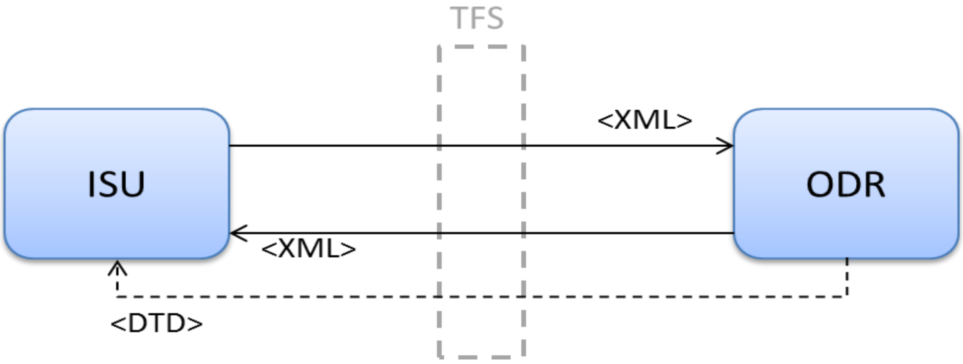
BALK



ISU contacts BALK directly, unlike through TFS.

ISU will search for a file in BALK system, if it finds, the file will be loaded to ISU.

ODR



File transfer

ISU sends a request to ODR as an XML file, containing the information about month and the ISU signature.

ODR sends the data regarding the number of ODR volumes delivered, in an XML format to TFS, which will be pushed to ISU.

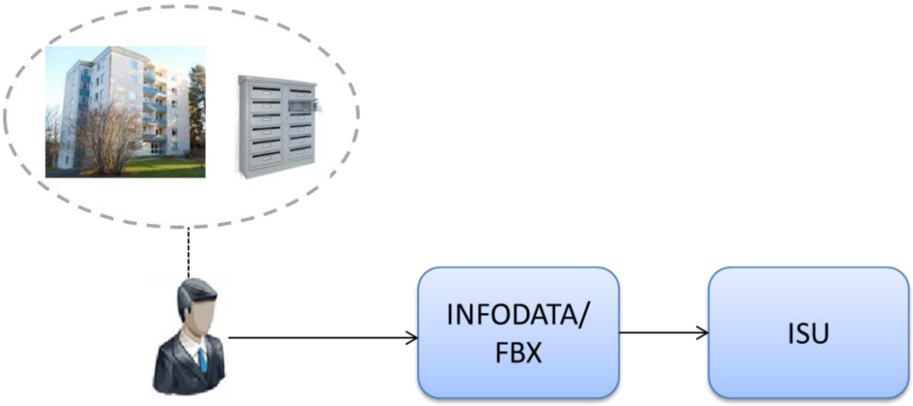
There are some temporary problems while parsing the XML file, because of lacking of the DTD file. As a work around solution, we are receiving the DTD file manually.

Data Export

ISU also sends data about addresses, household types, address blocks, routes and delivery offices to ODR Online. This will be done via special export tables and a database link via which ODR Online accesses the export table in the ISU database.

More details are available in **Define** and on Prime Portal

INFODATA/FBX



FBX (Refers to property boxer laying in stairwell or in connection with this) .

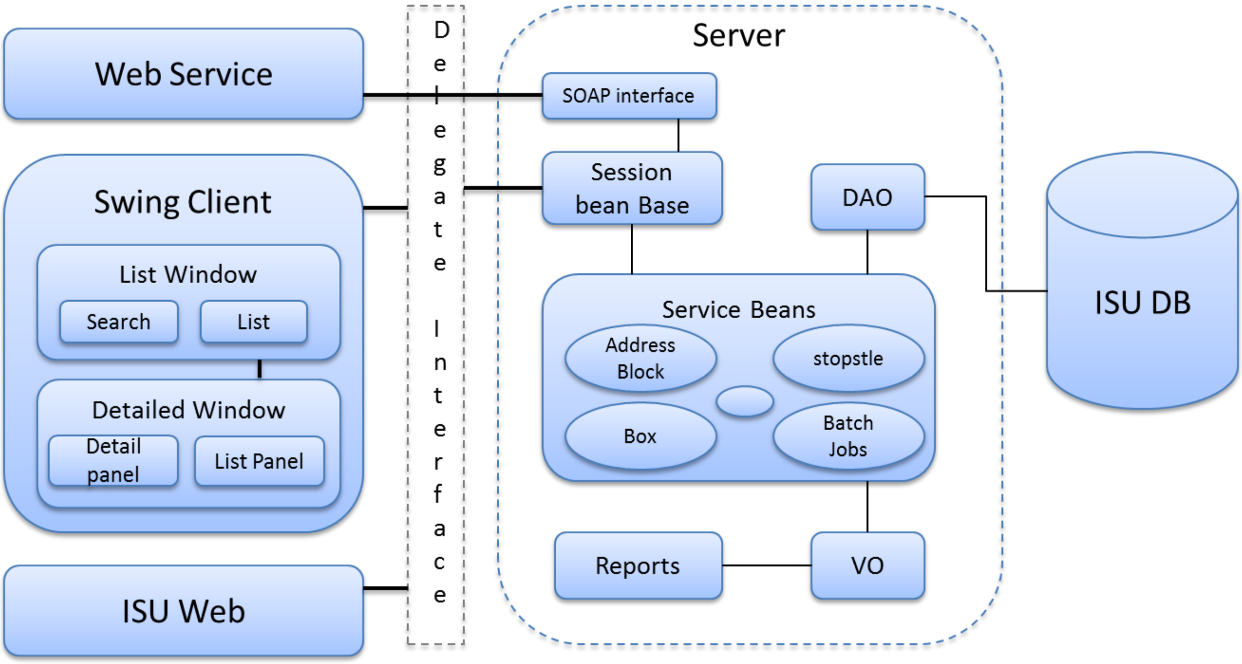
Definition: Fastighetsbox is a device for delivery of mail to all mail recipients in a rise in apartment buildings. Fastighetsbox placed inside the entrance floor, or outdoors in direct proximity to the entrance. Available in various designs, sizes and models. For each mail recipient is a separate opening for SORTING by mail. Fastighetsbox can be designed as boxing orientation or mailboxes grouped together.

If any resident or apartment owner wants to put a “Fastighetsbox”, he has to register the details in INFODATA/FBX. Then it will be updated in to ISU.

3.3.1.3 Inside view

Inside view explains about the execution flow and internal design of ISU application.

This is just to give an overall idea, about the internal processing. We might have missed some of the components, but all the vital components have been covered.



Client Layer

ISU application can be accessed by three different types of clients.

- Web service
- ISU swing client
- ISU Web

Web service

ISU Web Services is a portal where some of the modules and services of the ISU can be accessed from the ISU Web portal.
There include tools to manage mailboxes, internal mail, Business Reply Mail and Newspaper Claims and various different reports.

ISU Swing Client

The main purpose of ISU is to show and edit data in the database, and to help in developing and maintaining ISU a special framework has been developed that allows easy construction of the most commonly used windows and controllers.

The execution and the flow of ISU client is well documented and available in define and on Prime Portal

Server

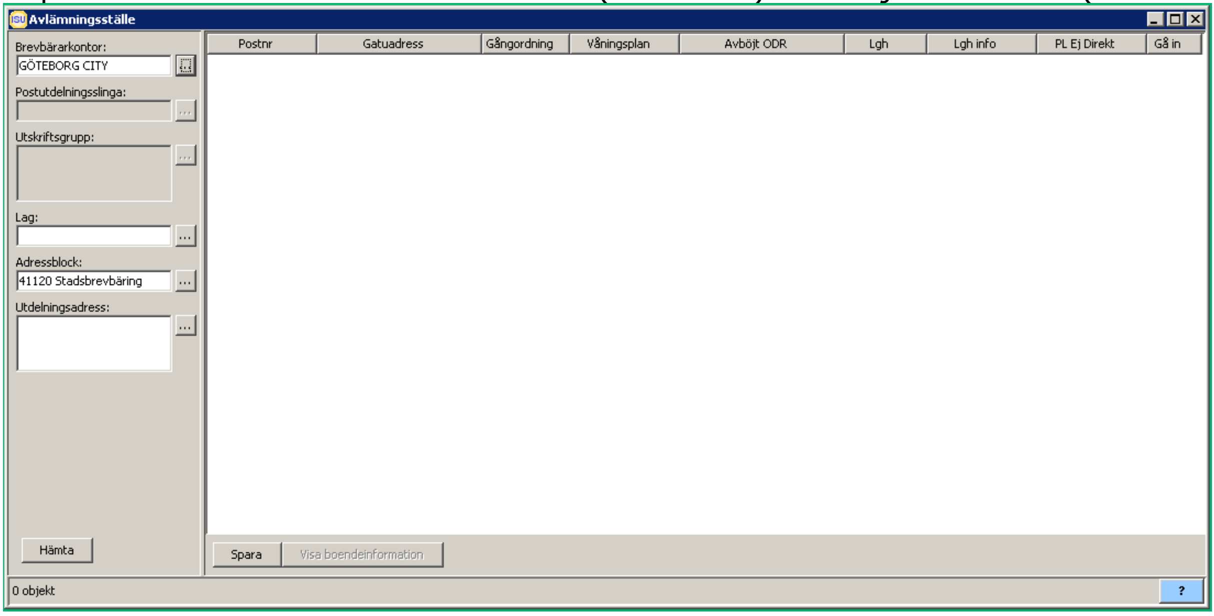
All the requests to the ISU server will be received by Delegate Interface and from there; they will be delegated to the corresponding session beans. Session beans in turn calls the corresponding Service beans (EJBs) . With the help of DAOs, service beans will contact database.

ISU is using Jasper reports to prepare some reports.
In ISU Client, using "Export to Excel" functionality we can get the reports.

With the help of VOs (Value Object) and through Service beans, reports will be generated. ISU will provide some kind of statistics also.

Example to show how this business functionality and logic is working for searching and editing information:

Below is a standard window that allows the user to enter certain parameters to search for on the left, and also displays the result in the list on the right. Each such window is specialized for the specific search and extends the *SearchAndListWindow* class, and implements sub-classes of *SearchPanel* (to the left) and *ObjectListPanel* (to the right).



Brevbärarkontor: GÖTEBORG CITY

Adressblock: 41120 Stadsbrevbäring

0 objekt

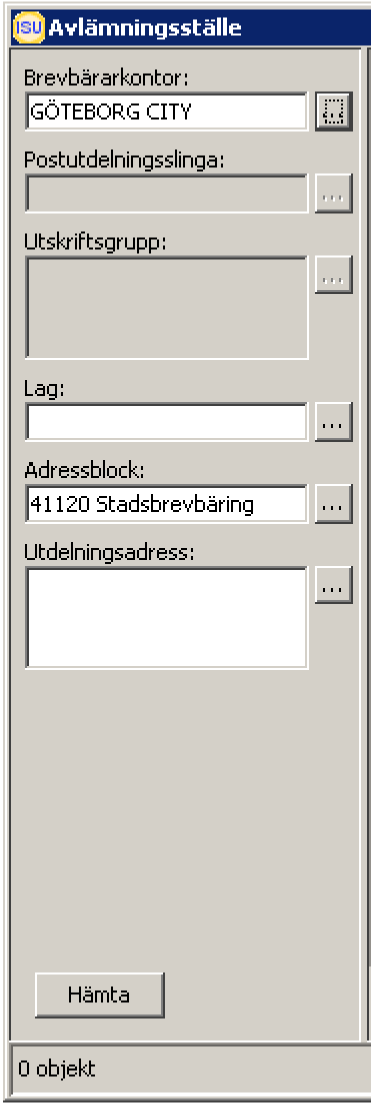
The framework around these classes automatically interacts with the database (covered further down) to retrieve information which makes implementation easier. These base-classes provide common functions like allowing a user to create, update or delete entries, prompting for unsaved changes, giving friendly error-messages. So to create a new window like above "all" that needs to be done is to extend the provided base-classes and functions, and define which database tables to use.

To the right is an extended class of *SearchPanel* that lets the user define what to filter data on. The different choices are represented through different Selector classes (*ListSelector*, *ListMultipleSelector*, *RadioButtonSelector*...) where the result of the choices are added into a common *ISUSearchRules* class. This class is then used when calling the database to retrieve data.

Example of a selector:

```
    UtdkontorListPanel list = new UtdkontorListPanel();
    list.setShowShortVersion();
    utdkontorRules = (ISUSearchRules) createSearchRules();
    list.setSearchRules(utdkontorRules);
    utdkontorSelector = new ListSelector("Brevbärarkontor:",
list);
    addSelector(ISUSearchRules.Id.UTDELNINGSKONTOR,
utdkontorSelector);
    utdkontorSelector.setMandatory(true);
```

This will create a Brevbärarkontor (Post office) selector that lets the user select a post office and bind that to the *ISUSearchRules* object in this *SearchPanel*. Above a *UtdkontorListPanel* panel has been used to implement the selector. There are many more *ListPanel*s which can be used to let the user select by different criteria. These *ListPanel*s are the same as the ones' used in the *SearchAndListWindow*s.
When the user clicks "Hämta" (Retrieve) the *ListPanel* on the right in the *SearchAndListWindow* will be populated.



Avlämningsställe

Brevbärarkontor: GÖTEBORG CITY

Postutdelningsslinga:

Utskriftsgrupp:

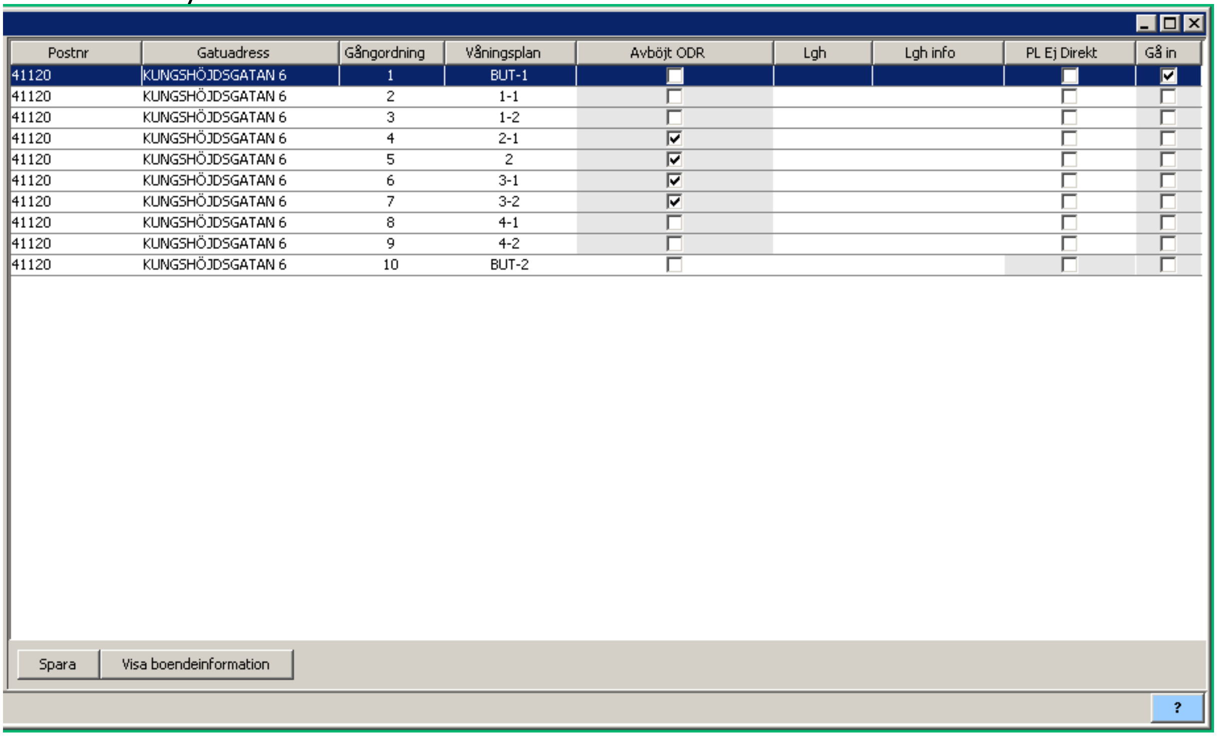
Lag:

Adressblock: 41120 Stadsbrevbäring

Utdelningsadress:

0 objekt

Here we have clicked ”Hämta” and the system has returned data that matches our *ISUSearchRules* object. All the fields below are specified in the extended *ListPanel* and automatically matched to the database.



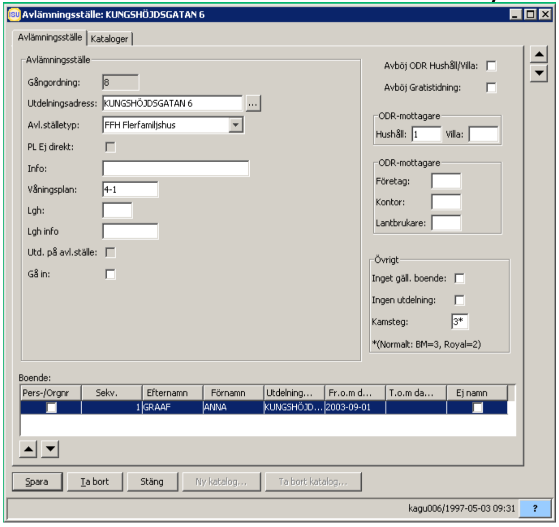
Postnr	Gatuadress	Gångordning	Våningsplan	Avböjt ODR	Lgh	Lgh info	PL Ej Direkt	Gå in
41120	KUNGSHÖJDSGATAN 6	1	BUT-1	□			□	☑
41120	KUNGSHÖJDSGATAN 6	2	1-1	□			□	□
41120	KUNGSHÖJDSGATAN 6	3	1-2	□			□	□
41120	KUNGSHÖJDSGATAN 6	4	2-1	☑			□	□
41120	KUNGSHÖJDSGATAN 6	5	2	☑			□	□
41120	KUNGSHÖJDSGATAN 6	6	3-1	☑			□	□
41120	KUNGSHÖJDSGATAN 6	7	3-2	☑			□	□
41120	KUNGSHÖJDSGATAN 6	8	4-1	□			□	□
41120	KUNGSHÖJDSGATAN 6	9	4-2	□			□	□
41120	KUNGSHÖJDSGATAN 6	10	BUT-2	□			□	□

And the code that connects this list to the values in the database:

```
    model = new MyModel();
    model.addColumn("PNR", "Postnr", "postnr", 100, false, true);
    model.addColumn("GADR", "Gatuadress", "adress", 150, true, true, true);
    model.addColumn("GORD", "Gångordning", "sekvensnr", 80, false,
true).setRendererClass(ObjectListPanel.CenterTableCellRenderer.class);
    model.addColumn("VPLN", "Våningsplan", "vaningsPlan", 100, false,
true).setRendererClass(ObjectListPanel.CenterTableCellRenderer.class);
    model.addEditableColumn("AODR", "Avböjt ODR", "avbojtGruppreklam", 150, false, true);
    model.addEditableColumn("AGT", "Avböjt GT", "avbojtGratistidning", 150, false, true);
    if(isEditableLagenhetsnr) {
        model.addEditableColumn("LGHNR", AvlamningsstalleLagenhetsNr.LAGENHETSNR_BEN, "lagenhetsNrAsStr",
50, false, true);
    } else {
        model.addColumn("LGHNR", AvlamningsstalleLagenhetsNr.LAGENHETSNR_BEN, "lagenhetsNrAsStr", 75,
true, true);
    }
    ...
```

Some simple operations can be performed in this list, such as checking and unchecking Checkboxes but to present more information and let the user edit an entry a *DetailWindow* is used.

Here the user has opened an entry in the *ListPanel* and been presented with an extended *DetailWindow*. These windows are usually a lot different from the *SearchAndListWindows* since they are specialized for their purpose:



Avlämningsställe: KUNGSHÖJDSGATAN 6

Avlämningsställe

Gångordning: 8

Utdelningsadress: KUNGSHÖJDSGATAN 6

Avl.ställetyp: FFH Flerfamiljshus

PL Ej direkt: □

Info:

Våningsplan: 4-1

Lgh:

Lgh info

Utd. på avl.ställe: □

Gå in: □

Avböj ODR Hushåll/Villa: □

Avböj Gratistidning: □

ODR-mottagare

Hushåll: 1 Villa:

ODR-mottagare

Företag:

Kontor:

Lantbrukare:

Övrigt

Inget gäll. boende: □

Ingen utdelning: □

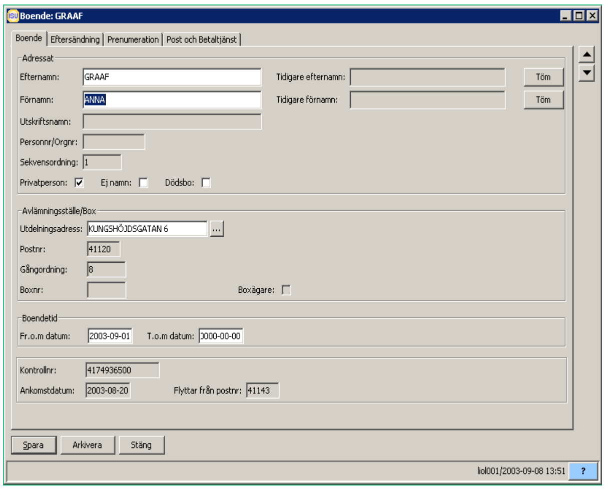
Kamsteg: 3*

*(Normalt: BM=3, Royal=2)

Boende:

Pers-/Orgnr	Sekv.	Efternamn	Förnamn	Utdelning...	Fr.o.m d...	T.o.m da...	Ej namn
□	1	GRAAF	ANNA	KUNGSHÖJD...	2003-09-01		□

kagu006/1997-05-03 09:31



Boende: GRAAF

Adressat

Efternamn: GRAAF

Förnamn: ANNA

Tidigare efternamn:

Tidigare förnamn:

Utskriftsnamn:

Personnr/Orgnr:

Sekvensordning: 1

Privatperson: ☑ Ej namn: □ Dödsbo: □

Avlämningsställe/Box

Utdelningsadress: KUNGSHÖJDSGATAN 6

Postnr: 41120

Gångordning: 8

Boxnr:

Boxägare: □

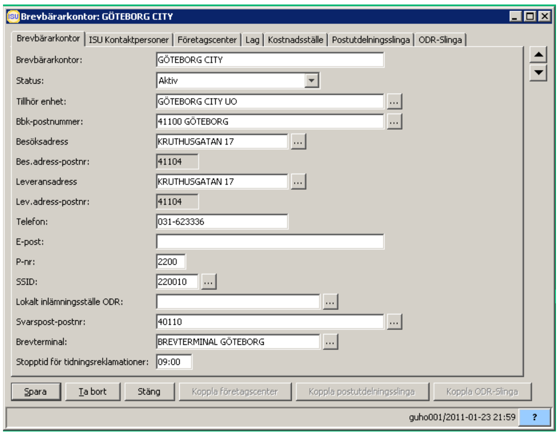
Boendetid

Fr.o.m datum: 2003-09-01 T.o.m datum: 0000-00-00

Kontrollnr: 4174936500

Ankomstdatum: 2003-08-20 Flyttar från postnr: 41143

liol001/2003-09-08 13:51



Brevbärarkontor: GÖTEBORG CITY

Brevbärarkontor: GÖTEBORG CITY

Status: Aktiv

Tillhör enhet: GÖTEBORG CITY UO

Bbk-postnummer: 41100 GÖTEBORG

Besöksadress KRUTHUSGATAN 17

Bes.adress-postnr: 41104

Leveransadress KRUTHUSGATAN 17

Lev.adress-postnr: 41104

Telefon: 031-623336

E-post:

P-nr: 2200

SSID: 220010

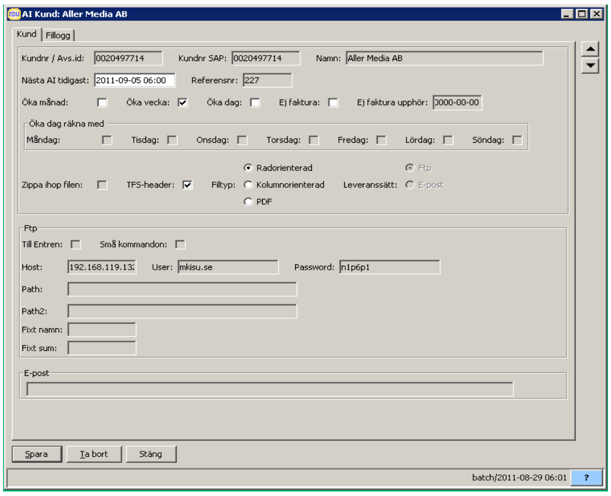
Lokalt inlämningsställe ODR:

Svarspost-postnr: 40110

Brevterminal: BREVTERMINAL GÖTEBORG

Stopptid för tidningsreklamationer: 09:00

guho001/2011-01-23 21:59



AI Kund: Aller Media AB

Kundnr / Avs.id: 0020497714 Kundnr SAP: 0020497714 Namn: Aller Media AB

Nästa AI tidigast: 2011-09-05 06:00 Referensnr: 227

Öka månad: □ Öka vecka: ☑ Öka dag: □ Ej faktura: □ Ej faktura upphör: 0000-00-00

Öka dag räkna med

Måndag: □ Tisdag: □ Onsdag: □ Torsdag: □ Fredag: □ Lördag: □ Söndag: □

Zippa ihop filen: □ TFS-header: ☑ Filtyp: ● Radorienterad ○ Kolumnorienterad ○ PDF Leveranssätt: ● Ftp ○ E-post

Ftp

Till Entren: □ Små kommandon: □

Host: 192.168.119.13: User: mkisu.se Password: n1p6p1

Path:

Path2:

Fixt namn:

Fixt sum:

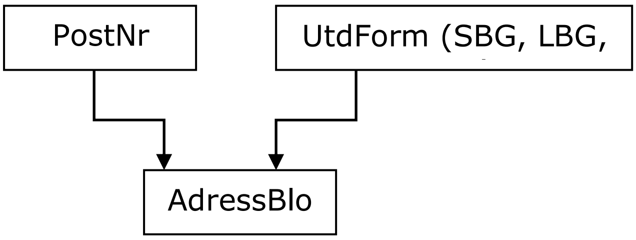
E-post

batch/2011-08-29 06:01

Each of these panels extends *FieldDetailPanel* where the GUI layout and data presented is defined. Like in the *SearchPanel* above this is also done through special *Selectors*, which through the framework is linked to the database. Using this design it is possible to link in multiple *DetailPanels* into a single *DetailWindow* to present more information related to an entry.

3.3.1.4 ISU Application Description

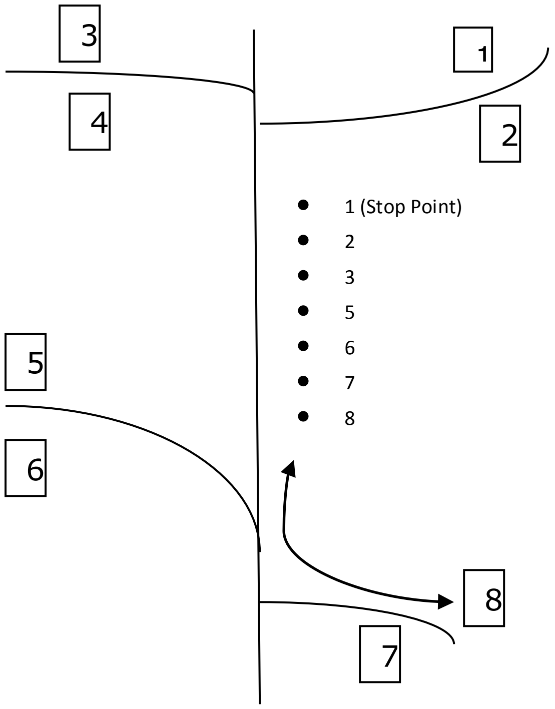
we are mainly focusing on the UTDELNINGSSTOD tab of ISU application, because it is most common tab for the user and contains approximately all the main object of ISU application.
1)- UTDELNINGSADRESS (Delivery Address):- Under this menu we can search all delivery address by using of Postoffice(Brevbararkontor), delivery route(Postutdelningsslinga), Stop Group(Utskriftsgrupp), Postmen Team(Lag), Address Block(Adressblock) details.
This comes under the table Stopstle.
All Postoffice belongs to one or multiple PostNr, every postNr belongs to one or multiple Address Block. There are three type of Address Block (comes under UTDFORM table):-
a) - SBG :- Stadsbrevbäring (Inside the City)
b) – LBG :- Lantbrevbäring (Rural Letter Bearing)
c) – Box :- Box



2) – Boxsektion (Address Area):- This is the combination of box of the Address Area. This is comes under the table ADRESSOMRADE.
All the boxes start with the name "Box" in the STOPSTLE table. Box section do never has a connection with master data; it is itself a delivery address.
3)- Stopplats (Stop Point) :- The table contains information on stop locations. One-stop location can be defined as the place where the postman stops to hand out one or more items (one or more delivery addresses).
This comes under the StopStlePlats table.

There are 5 type of Stop Point:-

a)- LÅDSAMLING (Lad Collection):-



Postman will stop to stop point 1 and deliver all item to postbox (1 to 8) according the delivery address. Then user will come to their postbox (user 8 will go to postbox 8) and collect the item.

b)- FASTIGHETSBOXANLÄGGNING (Apartment type building PostBox) :- This box section placed to the building, which contains multiple apartments. So postman no need to go each floor to deliver the post. He has to deliver the post into the box in the BoxSection according to the delivery address.

c)- SAMHÄLLSBOXANLÄGGNING(Community Box Section) :- This box section is sepecial service provide by the Posten. Posten give this facility to the user, whose house is far away to the city. Posten give the Box section for the perticular community in the nearer city post office and user use to collect from that Post Office.

d)- ENSKILD (Individual):- User house work as a individual stop point. In this case StopPoint is same as Stopstle.

e)- POSTBOXANLÄGGNING(PostBox Section) :- This Box Section is usually placed in side the company, that contains the individual box for each employee.

4) - <u>Utskriftsgrupp (Stop Group):</u>- This contains information that postmen use to group a mail delivery route into smaller parts.
This comes under the StopGrupp table.

5) - <u>Avlamningsstalle (Drop off):</u>- A drop-off is each individual's mailbox, mail messages, etc. where we deliver mail. Each street address contains one or more drop-off.

Drop-off deal with you in the selection list DIVIDEND PAYMENT / drop-off.
 This contains the other type (Type A of Putstle) of mail delivery points and represents the delivery of letters to the apartments and houses.
A drop-off box is just like a "one-to-one" relationship to the table PutStle (part primary key). A drop-off must have a connection via putstle to a StopStle.

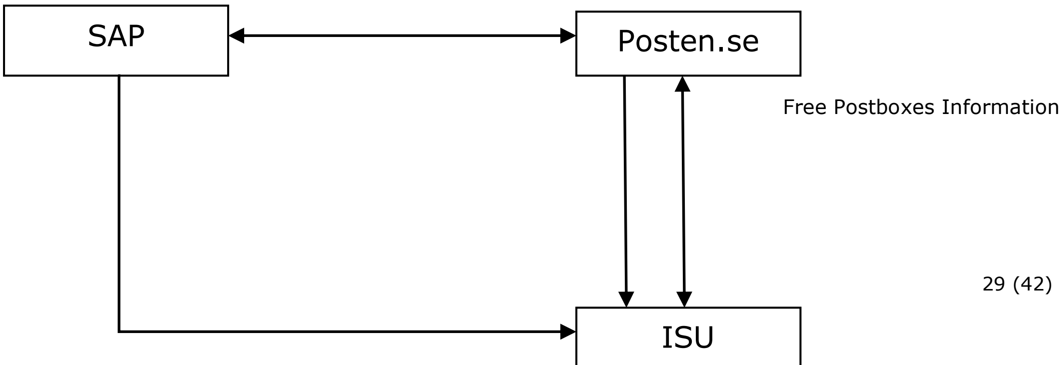
Avlämningsställetyp is the division of the drop-off based on where the items are awarded. The main aim is now to divide the drop-off in categories according to customer's mail volume in order to obtain reliable statistical estimates.

- **FFH** (Goes into the property, plug boxes, or on balconies)
    - **Definition**: The multi-family houses provided with at least 4 per drop-off gain / entrance
- **FBX** (Refers to property boxer laying in stairwell or in connection with this)
    - **Definition:** Fastighetsbox is a device for delivery of mail to all mail recipients in a rise in apartment buildings. Fastighetsbox placed inside the entrance floor, or outdoors in direct proximity to the entrance. Available in various designs, sizes and models. For each mail recipient is a separate opening for SORTING by mail. Fastighetsbox can be designed as boxing orientation or mailboxes grouped together .
- **PL** Solo's mailbox at the property boundary and mailboxes in lådsamlingar. With PL provided street address with a maximum of three drop-off
- **TOY** A drop-off with large amount of mail, such as a business, shops, offices and the like.
- **SPL** Vacation Hold, where the distribution takes place only during the period June-August

PVA Only if the value of mediation is

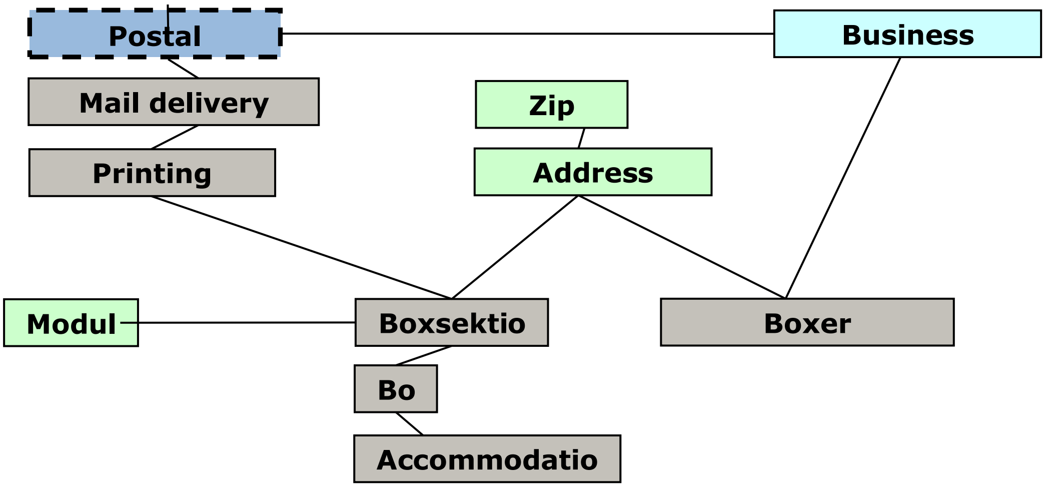
Note: - Kataloger refers to yellow page delivery. But now a day, Posten is not using this feature of ISU.
A Swedish Company (**Eniro**) provides their web service for this feature.

5) - Box (Postbox):-



Web service

Boxarende (Contains information that Comp has reserved the box)



New box contract handled by the central sales administration. Customers who want the PO Box to order them via Posten.se or the customer can call the Customer Service to help with their order.
Data on job boxes, and the boxer orientation as the box belonging retrieved from ISU. Only boxes that are marked as "vacant" and linked to a "punch-laying" is searchable for rentals.
The contract for the endorsement sent to the business center / postal delivery office in charge of the boxing facility. They also leave the keys.

- Data on customer and contract period are recorded in SAP and ISU. The ISU is updated following tasks automatically in case the reservation "Boxstatus", "Agreement / Quarantine maturity", "Customer Name", "Customer Accommodation”,” Sequence No” with a note "Boxägare" and UDM receiver.
- When sales administration assigns a box, it update box in the ISU immediately with status Reserved and Agreement Date, and all customer information. At each new agreement sent a PostOffice message to affected postal delivery office with information on new box reservation, the box has the status "reserved" (ISU updated as soon as the agreement saved)
- Sales Administration completes leasing when the agreement is signed and the keys left, (can be made until the day after the box is assigned).

Automatically, the status of the ISU to "hired" and a PostOffice message sent simultaneously to the relevant postal delivery.

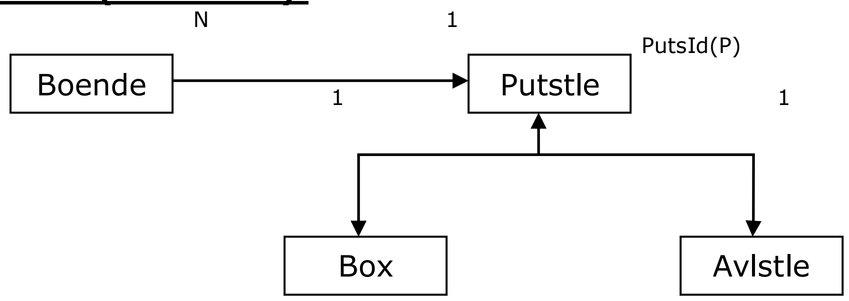
- PostOffices shall check the data on box tab in ISU application that right UDM receiver is registered with the box.
- Upon termination of the Agreement from the ISU Web. Changes should be done in ISU immediately contract date to closing date, but the status of leased stands remained until the case is litigated in SAP. End cases process as in SAP (at night, updating ISU next day), when the trade is done in SAP Post Box message sent to the relevant postal delivery office, contains the closure notice. Box status change to "Quarantine" and a Quarantine Date added till completion date +6 Mon
  - quarantine time can be adjusted manually, shortened or extended at the discretion of PostOffice mailbox manager

7) – ODR-MOTTAGARE (ODR RECEIVER):-

UDM is linked to the drop-off (avlstle) or to Box. When a drop-off or box get the address change, it may result in the recipient types changes and needs to be update. When a new address connected via Forwarding Moves, it need to do simultaneously update the recipients also.
UDM addressee information in the ISU form the basis of numerical data provided to customers. ODR-Online is updated with the number of data from the ISU.
Definitions of ODR recipient types:-

- **Households:** - A household living in a house or apartment.
- **Villa:** - Villa referred to detached houses, semi-detached or terraced houses. The detached Housing with one, two or three apartments, each apartment house has a receiver. Houses are not included. Note: Ownership of Drop-off(avlstle) are not affiliated with ODR receiver
- **Farming :** - As Agriculture counted farms where farming, forestry, animal husbandry (eg poultry farm, horse farm) or fruit and vegetable crops. A farm / property where any of the above has been conducted but ceased, land is leased, etc. where the farm / property only bomingshus defined environmental reasons not to farmers (but as detached)

7) - Boende (Residence):-



PutsId(F) PutsId(F)

3.3.1.5 Domain Model

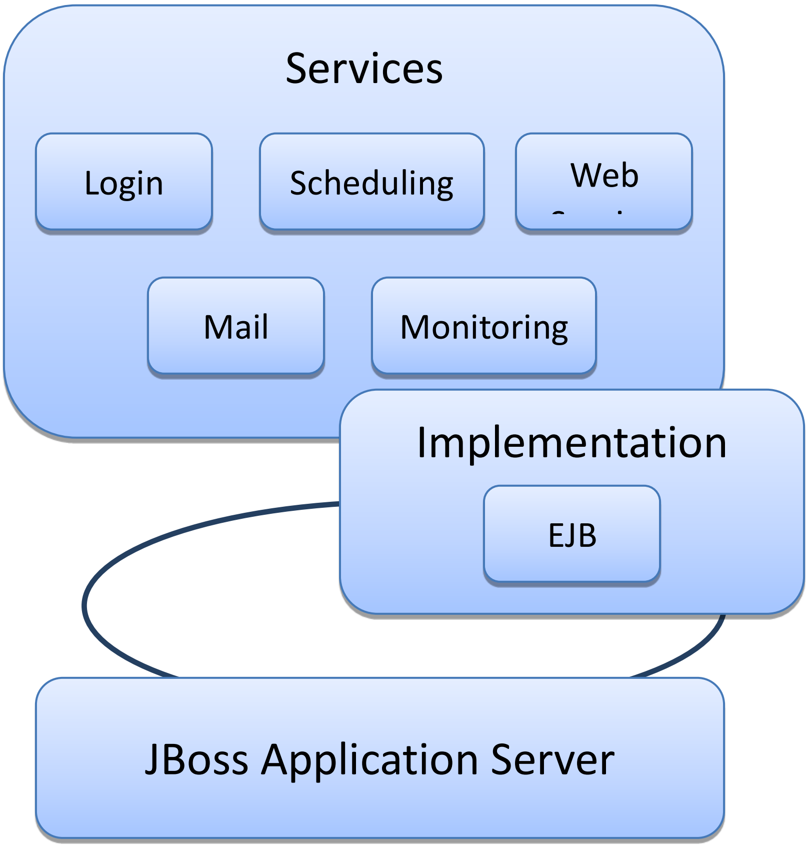
Brief domain model is available in **define** and on Prime Portal

3.3.1.5 Use-Case Realizations

A selection of architecturally significant use case realisations are available **define** and on Prime Portal

3.3.2 Implementation view

This section explains about how the application is implemented, in a most generic way.



ISU has some services, which are placed in the above diagram, have been implemented through EJB, SMTP and JBoss application server.

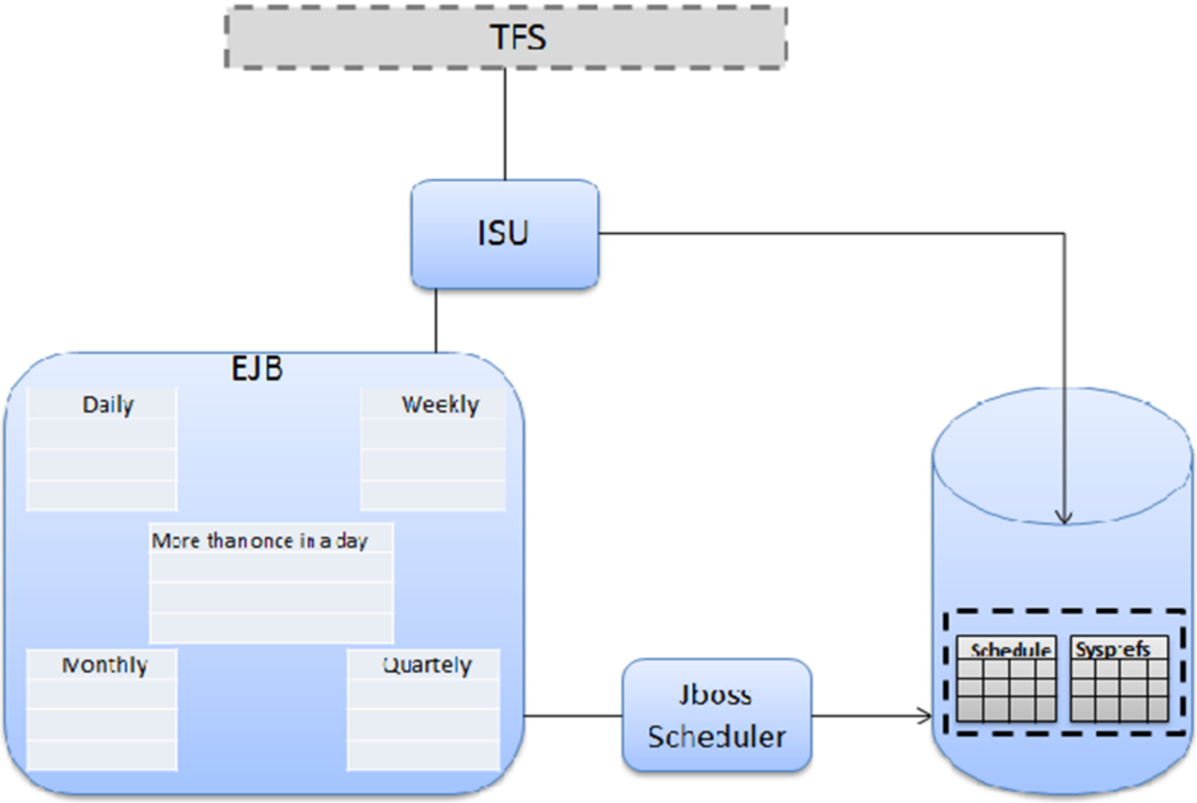
EJB takes care of all the persistence mechanism and the communication with the database with the help of helper classes (DAO).

SMTP takes care of sending mails, to update the ISU team on successful or failure communication with the external systems.

JBoss application server takes care of the scheduling and monitoring the other resources and it provides pseudo support to the whole application.

3.3.3 Process view

This section explains about the scheduled processes in ISU.



ISU communicates the external systems, mostly through scheduled batch jobs only. JBoss job scheduler will take care of batch job execution at the pre-defined time.

All the details regarding the batch job execution are placed in SCHEDULE table. The next execution time, the service bean responsible to run the job and the directory which holds the source file, all these details will be there in SCHEDULE table.

S.No	**Name of Monthly Batch Jobs**	**Description**	**Duration**
1	Vapobatch	Väpo sends us a file through TFS	starts 9:15 to load the file
2	ODR Begär volymStatistik	We request information from the ODR	starts at. 9.00
3	ODR VolymStatistik	we get a file back from ODR	starts at 10.00
4	Närservice	We get an Excel file usually from Andreas Axenholm , This file hold information about the amount paid by different customers for local service and needs to be imported into the ISU database	09.30

5	PSS	We get a file from PSS	after 12 o'clock

DEKvolym jobs are done then we will start three different batchjobs that will send all data to SAP.

The files to SAP is created and dispatched by setting these batchjobs to “forced” (FORCE = 1) in the schedule table.

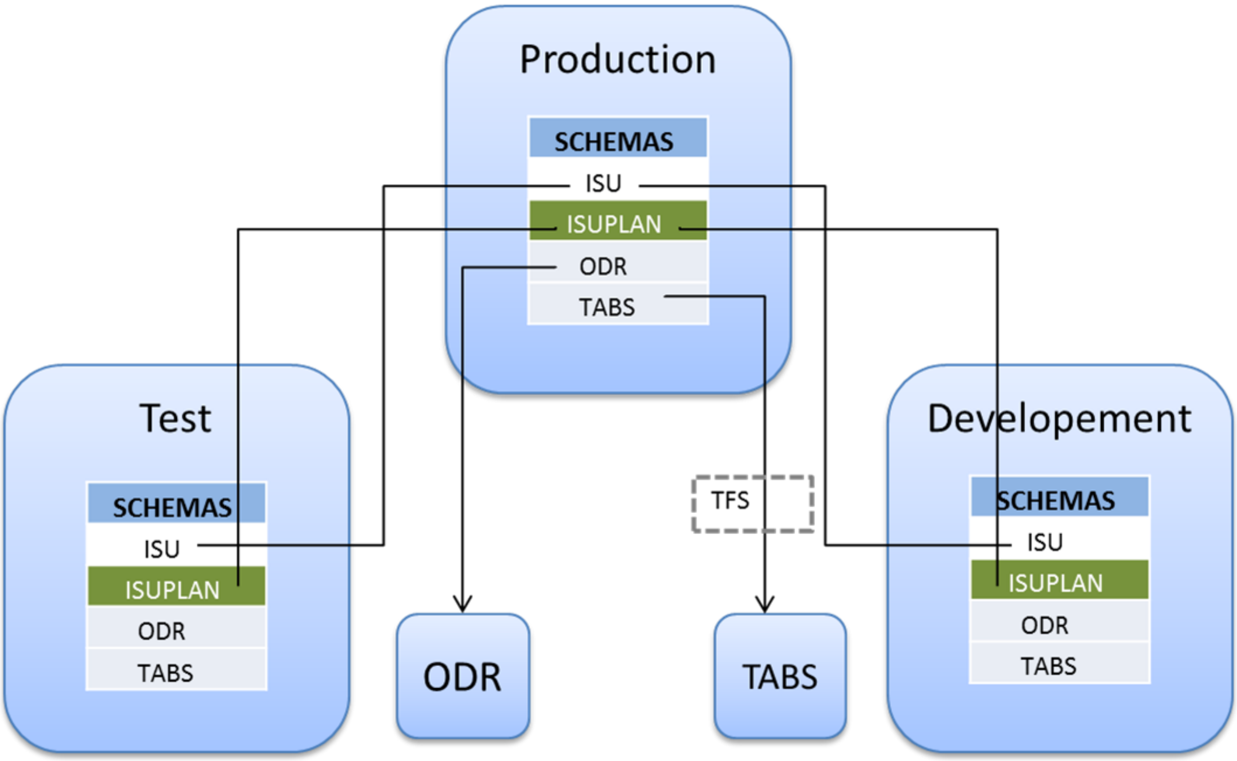
Update ISU.Schedule table by setting next month first working day for schedule id: **200818465 and 235260995**.

Similarly update the ISU.Schedule table by setting the next month second working day for Schedule id: **233955706**
Some of the batch jobs are available in the batch jobs excel sheet.

batch_jobs.xls

3.3.4 Data view

ISU has a critical and complex data model from the view of business functionality. It is an Oracle database. The database schemas in the different environments and the relation between them has been described in the following diagram



As described in the diagram, ISU has 4 schemas.

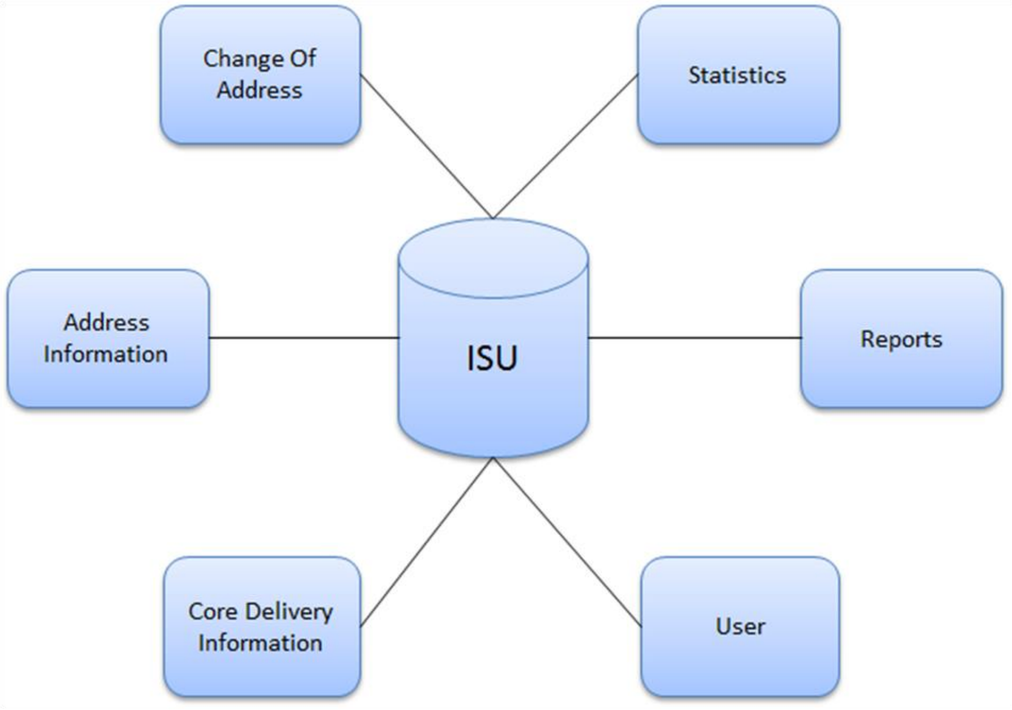
ISU, ISUPLAN, ODR and TABS.

It’s a manual job to dump the data from the ISU and ISUPLAN schemas from the production to the test and development. The data from the ODR schema will be dumped into ODR system’s database manually, every quarter of the year. TABS data will be loaded to the TABS system automatically, whenever any update happens in the ISU production. The communication to the TABS will be through TFS.

More details will be available **here** (also published to **Prime portal**)

Because of the complexity we tried to categorize the whole database into some major groups. The following diagram gives an idea as an overview.

The below categorization of the database into groups, meant for easy understanding, and there is no actual or physical categorization in the database.



3.4 Source code Structure

There is no any j2ee standard coding convention followed in ISU application. ISU is comprised into 4 different projects.

ISU Client
ISU Web
Server
Web Services

Each of these has been developed in its own way and there is a nice relation has been defined among these to establish a stable ISU system.

An environment setup detail is given in below:

Dev_Environment_
Setup_ ISU.doc

3.5 Security Description

- ISU is a Posten intranet application and it can be run only on the Posten network, either through VDI or Citrix environment.
- Login credentials would normally be same as the VDI login credentials. More details are provided **here** (published to Prime portal also)
- In order to make a web service call to ISU, the corresponding external system should be registered in the dependencies list of ISU.

4 Technical Environment

The following diagram explains the ISU technical environment.

ISU has been deployed in three environments,

Development
Test
Production
Education

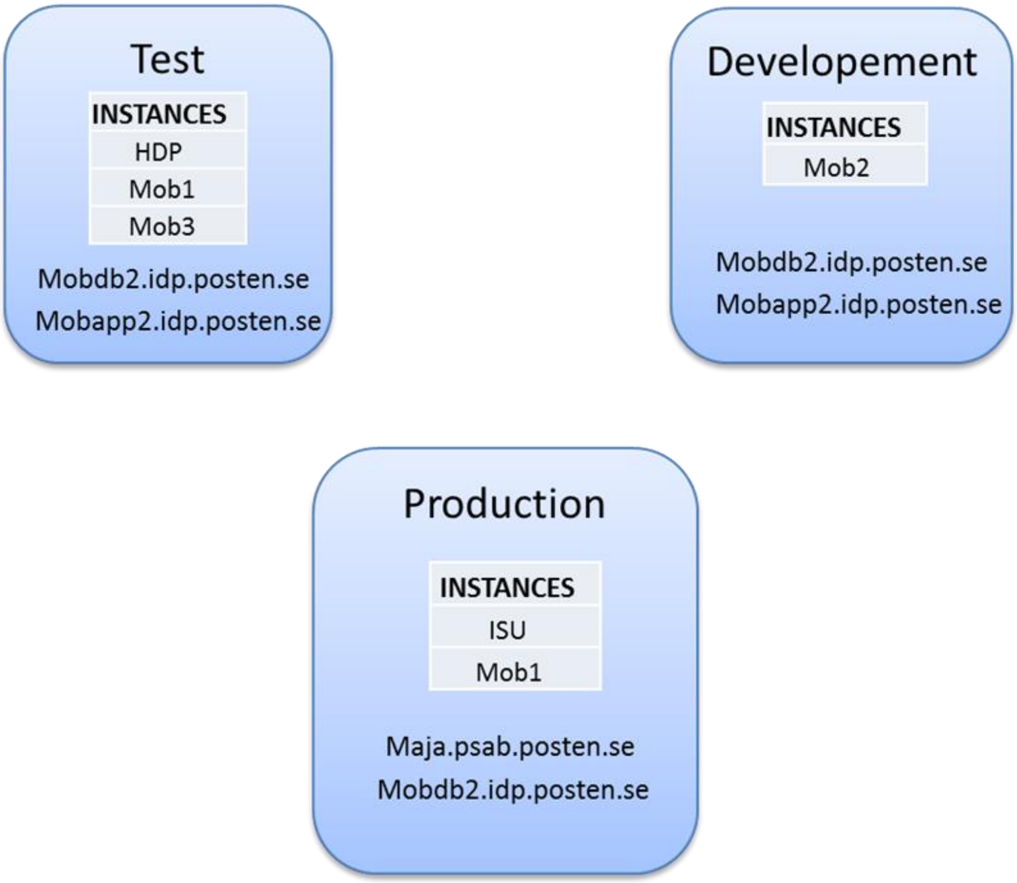
Test and Development environments

Application server --- mobapp2.idp.posten.se
Database server --- mobdb2.idp.posten.se

Production environment

Application server --- maja.psab.posten.se
Database server --- maja.psab.posten.se
Database server --- mobdb2.idp.posten.se (education)

Even though the same database server is being distributed among all the 3 environments, the server instances are different for each and every environment. That has been clearly mentioned in the diagram.



Test

INSTANCES
HDP
Mob1
Mob3

Mobdb2.idp.posten.se
Mobapp2.idp.posten.se

Developement

INSTANCES
Mob2

Mobdb2.idp.posten.se
Mobapp2.idp.posten.se

Production

INSTANCES
ISU
Mob1

Maja.psab.posten.se
Mobdb2.idp.posten.se

Some more details information about Test and Production Environment is given below:

Test and Production
Environment.doc

4.1 Hardware

4.1.1 Client hardware

Not Applicable

4.1.2 Server hardware

Server hardware requirements should be at least the following configuration.

Production Environment:

Model: Intel(R) Xeon(R) CPU E7440@2.40GHz
Primary Memory (RAM): 32 GB

Test/Stage Environments:

Model: Intel(R) Xeon(R) CPU E5420@2.50GHz (4 processor machine)
Primary Memory (RAM): 8 GB

To run the JBOSS Application Server locally on developers PC, no special requirements are needed.

The target server environments and database (embedded in to application server) are deployed on UNIX on Solaris 2.6 Operating System.

4.1.3 Other aspects on hardware

4.1.4 Communications

The development machines must be connected to the posten’s intranet and required firewall to be open in order to access all external systems.

4.2 Software

4.2.1 Operating system

Name	: Solaris
Version	: 10
Account	: Access level (for example local administrator)

4.2.2 Application server

Name	: JBoss
Version	: 4.2
Account	: Access level (for example local administrator)

4.2.3 Database Manager

Name	: Oracle
Version	: 10g (10.2)
Account	: Access level (for example local administrator)

4.2.4 Development

4.2.4.3 Development tools

Name	: Eclipse
Version	: Galileo
Account	: Access level

Name	: Putty
Version	: 0.60
Account	: Access level

Name	: WinSCP
Version	: 4.3.5
Account	: Access level

4.2.4.4 Version control

Name	: Subversion
Version	: 1.6.7
Account	: Access level

Instructions to add new users:

4.2.4.5 Build tool

Name	: Ant
Version	: 1.8.0
Account	: Access level

4.2.5 Database Development tool

Name	: SQL Developer
Version	: 2.1
Account	: Access level

4.2.6 Other software

5 Guidelines

5.1 Document structure

ISU documentation can be found in Define at Define 090-ISU documentation or in Prime Portal at PP 090-ISU documentation.

Some of the documents available at this location are:

- 10122-System_documentation » ISU_user_manual.pdf
- 10122-System_documentation » 090-ISU System Documentation.pdf
- 10123-Operation_documentation » 090-ISU AM handbbok.pdf
- 10123-Operation_documentation » BatchSchema.pdf

5.2 Conventions and rules for documents, models, source code, etc.

Source code is available at the below SVN url

svn://mobapp2.idp.posten.se:8001/isu/trunk

5.3 Test

Testing document has been uploaded in **define** and on Pirme Portal.

6 Installation

Installation and related information are uploaded into #define.

Or
https://wm.define.nu/project/postennorden/090isu/versioncontrol/browse/10-Application_Management/1012-Documentation/10129-Other_documentation
in Prime Portal