



# **Navet**

## **and how we use it at Lund University Library**

**Snorri Briem, Lund University Library 2019**

# What is Navet?

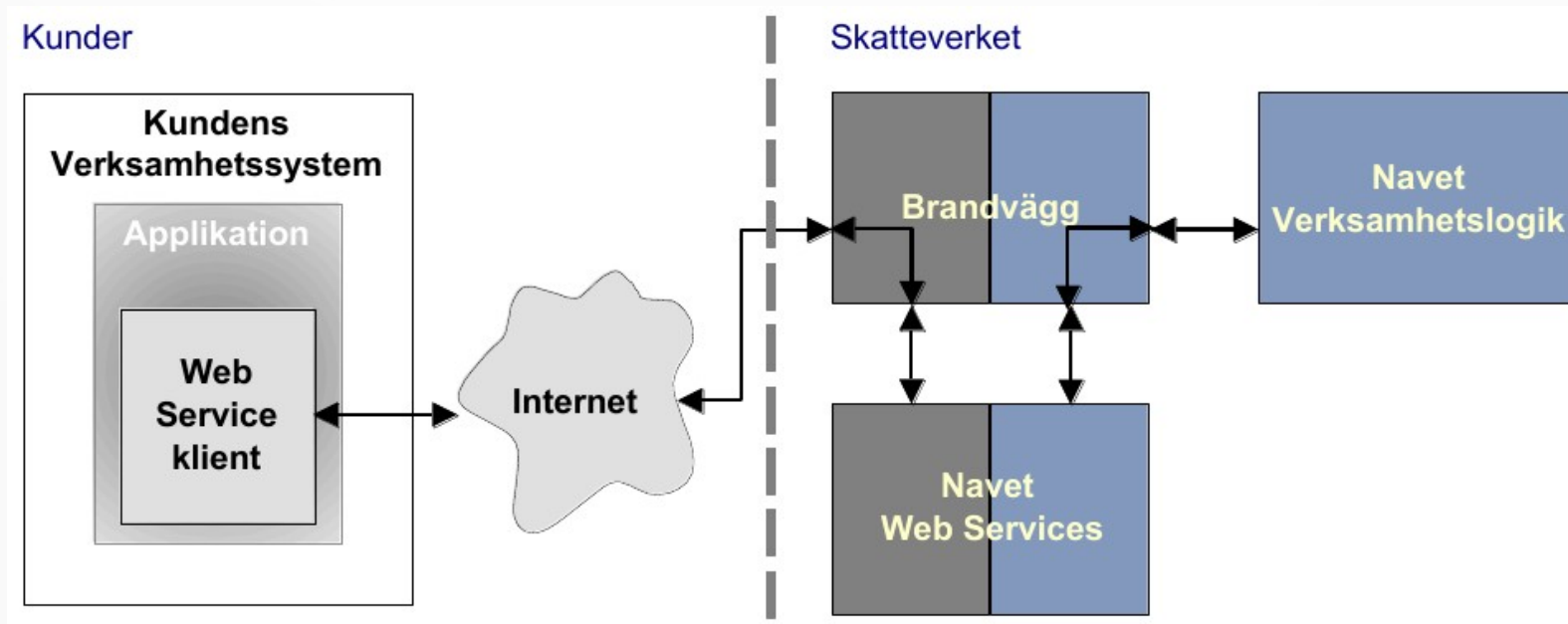
- Information system that allows public institutions to fetch data electronically from the population registry in Sweden
- Managed by Skatteverket
- The ePersondata web service allows fetching of information in real time. Information like:
  - personnummer
  - name and official address
  - citizenship etc.

# Why do we need it in Koha?

- In our external Patron Registration wizard; to verify the personnummer and get the name of the person
- Look up addresses for any patron in the UI on the fly (No addresses are stored in our Koha installation!)
- Look up addresses when invoicing long-overdue books

# How does it work?

- SOAP over a secure connection (https)
- Data output format can be either XML or termcode/termdata



# Requirements

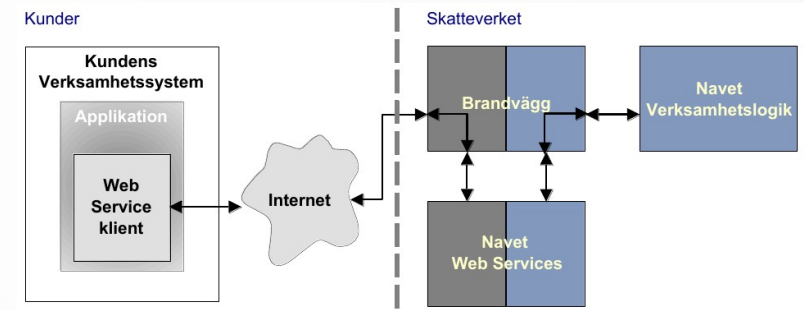
## 1) Register with Skatteverket

- Submit form - get order ID

## 2) Apply and get a certificate from Expisoft (valid for 2 years)

## 3) Obtain/Develop your own client to manage the communication with Skatteverket

- Possible to get test a identity and a test certificate that can access sample test data, without registration.



# Navet::ePersondata

- A separate Perl module that provides access to the ePersondata web service
- Handles communication with skatteverket
- Open source, with documentation and examples

```
NAME
Navet::ePersondata - Simple framework for the Navet ePersondata web services

SYNOPSIS

use Navet::ePersondata::Personpost;

my $ep = Navet::ePersondata::Personpost->new(
    # Set proxy to test service instead of production
    soap_options => {
        proxy => 'https://www2.test.skatteverket.se/na/na_epersondata/V2/personpostXML'
    },
    pkcs12_file => '/path/to/certificate.p12',
    pkcs12_password => '5085873593180405',
    OrgNr => '162021004748',
    BestallningsId => '00000079-F001-0001',
);

my $node = $ep->find_first({PersonId => '198602212394'})
    or die "Failed: ", $ep->error->{message};

my ($namn_node) = $node->findnodes('./Personpost/Namn');
print "The name is ", join ' ', $namn_node->findvalue('./Fornamn'),
    $namn_node->findvalue('./Efternamn'), "\n";

# Using Namsokning and termdata format:

use Navet::ePersondata::Namsokning;

my $ep = Navet::ePersondata::Namsokning->new(
    pkcs12_file => '/path/to/certificate.p12',
    pkcs12_password => '5085873593180405',
    OrgNr => '162021004748',
    BestallningsId => '00000079-F001-0001',
    format => 'termdata'
);

my $query = {
    Kon => 'K',
    Fornamn => 'sar*',
    Postort => 'Boden'
};
```

```
$ # CLONE:
```

```
$ git clone https://gitlab.lub.lu.se/modules/Navet-ePersondata.git
```

```
$ # INSTALL DIRECTLY:
```

```
$ cpanm -S https://gitlab.lub.lu.se/modules/Navet-ePersondata.git
```

# Further information

- Complete information about Navet:

<https://skatteverket.se/foretagochorganisationer/myndigheter/informationsutbytemellanmyndigheter/navethamtauppgifteromfolkbokforing.html>

or just go to [skatteverket.se](https://skatteverket.se) and search for "Navet" !

- The Navet Perl module:

<https://gitlab.lub.lu.se/modules/Navet-ePersondata>

- Questions?