

# Managing a Complex Infrastructure for the Collection of Business Report Forms

*Leif Bochis Madsen, Statistics Denmark*

## 1. Abstract

The latest two years Statistics Denmark has been working on the replacement of MS Infopath forms in our Business Survey Division.

By February 2020, seven questionnaires are used in production, while eight others are under development and should be ready for production by the end of 2019. Still, approx. fifty questionnaires should be converted to Blaise 5 within the next three years.

The paper describes the status of the project including a number of issues that have been addressed and some of the tools we are building in order to incorporate the use of Blaise in the existing environment for collecting business form reports.

In detail, the paper describes our work on managing a range of servers needed in the environment.

## 2. Introduction

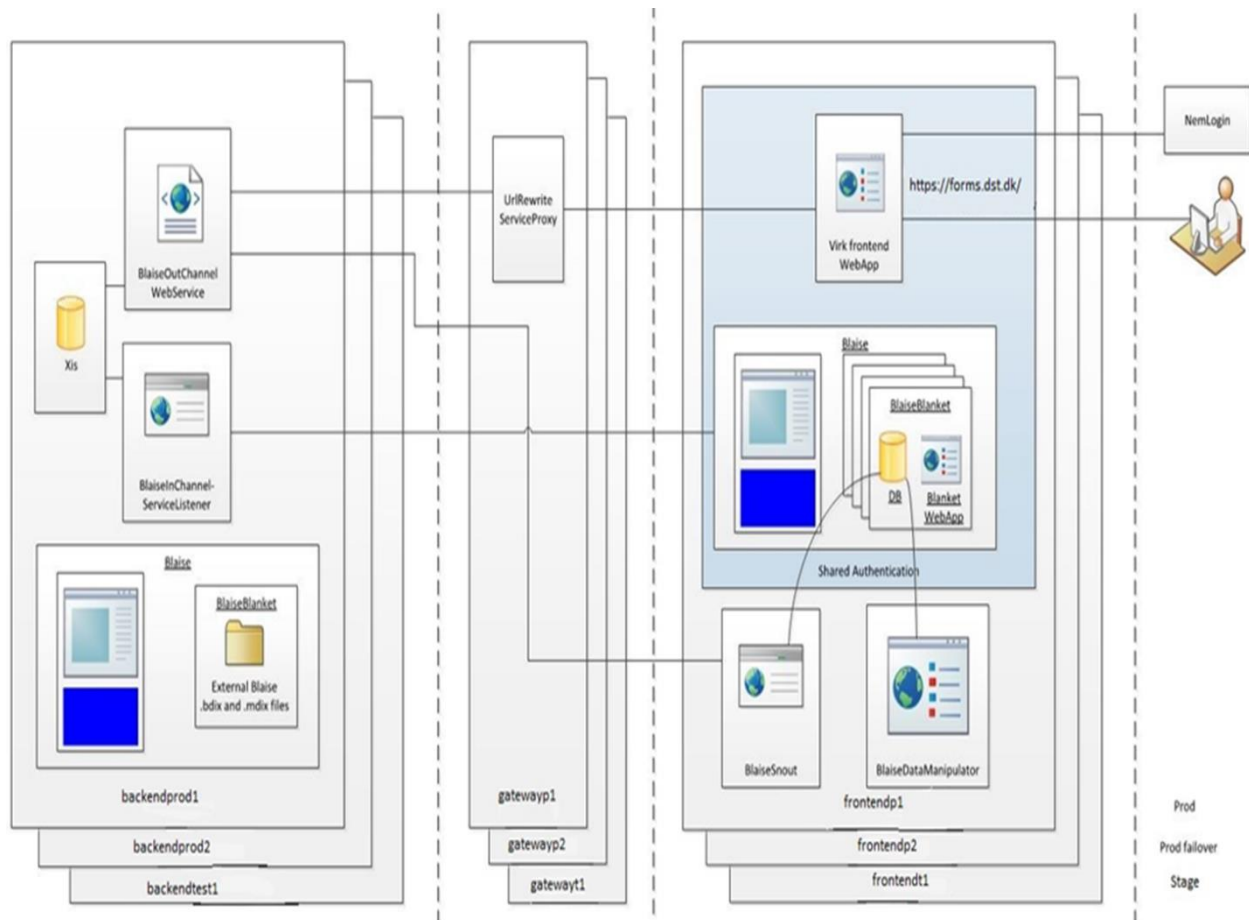
### 2.1 Statistics Denmark Business Survey System (BSS)

As described in my paper for IBUC/2018, Statistics Denmark has introduced Blaise 5 into an existing infrastructure to replace report forms written in MS Infopath with Blaise 5 questionnaires.

Put briefly<sup>1</sup>, this infrastructure implies the use of a Frontend server and a Backend server. The **Frontend** server is accessible to the respondents and here the surveys are installed as standard Blaise 5 Web surveys. The **Backend** server takes care of reading, writing and removing data, i.e. initialization of the forms with background information, and – after completion – removing data from the frontend Blaise database to store it in the background database.

---

<sup>1</sup> For extended explanation, see ref. [BSS].



**Fig. 1. Architecture of Business Survey Portal**

The concept of using backend and frontend servers for the different parts of the functionality is also used for a lightweight version of the form report system, the so-called ‘link access’ mode. This mode is used as an easier way for respondents to report data. They will not need to set up the entire public digital key interface before starting to fill, but on the other hand, the requirements prescribe that no previously filled data may be exposed to the respondent during the session. For some surveys, mainly the voluntary, this mode tends to increase response rates.

However, not all environments fit into this overall architecture.

Another two environments in the BSS – one for test and another for production – are used internally to fill forms if e.g. a respondent calls by phone and wants help to fill the form. These two environments share most of the functionality, like database access to prefill and store data, with the other BSS environments, but there is no need to split up the functionality between backend and frontend servers. In other words, these servers are COMMON.

At last, a number of servers that do not belong to the BSS. These servers are mainly used for household surveys and are set up as ‘classical’ Blaise servers, i.e. COMMON.

Environment	Test/Prod	EnvType	BACKEND	COMMON	FRONTEND	FW	COMMON	FRONTEND
VIRKTEST	TEST	VIRK	backendtest1		frontendtest1			
LINKTEST	TEST	LINK	backendtest1		linktest1			
VIRKSTAGE	TEST	VIRK	backendtest2					frontendtest2
VIRKPROD1	PRODUKTION	VIRK	backendprod1					frontendprod1
LINKPROD	PRODUKTION	LINK	backendprod1					linkprod1
VIRKPROD2	PRODUKTION	VIRK	backendprod2					frontendprod2
INTERNTEST	TEST	INTERN		interntest				
INTERNPROD	PRODUKTION	INTERN		internprod				
BlaiseTest2	TEST	BLAISE					blaisetest2	
Blaiseprod2	TEST	BLAISE					blaiseprod2	
Blaiseprod3	TEST/PROD	BLAISE					blaiseprod3	
Blaiseprod4	PRODUKTION	BLAISE					blaiseprod4	
Blaiset3	PRODUKTION	BLAISE					blaiset3	
Blaise2	TEST/PROD	BLAISE		blaise2				

**Fig. 2: Overview of environments**

To summarize, our Blaise environment can be described as a diversified infrastructure:

- Some servers are FRONTEND/BSS
- Others (data processing servers) are BACKEND/BSS
- A few are both, i.e. COMMON/BSS
- A handful are COMMON, but not related to the BSS
- Some are for PRODUCTION, others for TEST
- Some are INTERNAL, others EXTERNAL
- All of them has Blaise 5 installed
- All need to be administered

### 3. Server administration tasks

The common, administrative tasks comprise:

- Install surveys in the BSS environments (in Non-BSS environments the Blaise Server Manager interface is still used)
- Upload supplementary files to servers
- Get info about currently installed surveys
- Get overview of servers and environments
- Get info about Blaise versions installed on servers
- Update Blaise licenses (automatically)
- Update services

Some of the required tasks we shall describe further below.

### **3.1 Install survey in a BSS environment**

The process of installing a survey in a BSS Backend/Frontend environment consist of a number of operations.

First, any possibly existing installation of the survey must be removed. The general policy is that it is allowed to replace a survey in test environments, but not in production environments. Thus, if a replacement is requested in a production environment the entire operation should be cancelled automatically.

Second, the package file must be uploaded to a proper placement on the backend server.

Third, from this position the survey can be installed on the frontend server, and afterwards, a remote data interface (bdix) can be generated using the metadata and manifest files extracted from the package file.

At last, the survey installation on the frontend server is modified by inclusion of an http module and some extensions added to the configuration file in order to install our security procedures.

Albeit a complex operation that requires actions carried out on backend as well as frontend servers, with differences between environments, it is still possible to automate it.

### **3.2 Overview of servers and environments**

Instead of keeping track of the properties of each environment in a separate document, it should be possible to extract the current properties of the environments and servers directly.

For example, these overviews can be used to help managing upgrades, as well as to display the current status of each environment.

### **3.3 Update services**

Developed as a prototype, these services are subject to constant change. It is therefore necessary to update these services regularly, and in order to decrease the burden of updating, it must be carried out automatically.

## **4. Implementation**

The application consist of three, separate parts:

1. A WCF service implementing a series of methods carrying out specific tasks on a specific server
2. A Console application working as a client for the servers and implementing the knowledge of all the environments, servers and their mutual relations
3. A Graphical User Interface making the provided tasks easily available for the users

### **4.1 WCF Service**

The services are implemented as a DLL serving as a general WCF service based on basic .NET technology. At present, the services are not making use of the Blaise API, but only communicating with Blaise via e.g. the Server Manager program.

The service DLL comprises a set of basic, non-public methods for reading and writing information from and to XML files, the Windows Registry and directories, uploading files, executing commands, updating log files, etc.

Public methods comprise methods of a more specific character like upload Blaise 5 installation packages (.bpkg files), install these packages on a Blaise 5 web server, return a list of installed surveys, etc.

It is important to keep a firm separation between the non-public and public methods in order to control which actions are available. Also, the services must be unavailable from outside and only exhibit their methods to relevant, authenticated users inside the organization.

The general WCF service exhibits two enumerations that a client can use in order to manage proper operations:

The **server type** defines three kinds of servers (BACKEND, FRONTEND, COMMON).

The **form type** defines four kinds of reporting (Web Form, Internal Report Form, Link Access Form, Classic Blaise Form). The first three denote the kinds of form reports in the BSS, while the latter denotes all other Blaise surveys. If needed, it should be easy to extend the list.

The exhibited operations comprise, e.g.:

```
bool DoesSurveyExist(survey name, server type, form type);
string GetOS(); // version of the operating system on the server
string[] GetVerifiedSurveyList(server type, form type); // List surveys of specified server and form type
string GetSurveyInfo(survey name, server type, form type);
string GetInstrumentID(survey name, server type, form type);
DateTime GetSurveyInstallDate (survey name, server type, form type);
DateTime GetSurveyPrepareDate(survey name, server type, form type); // Timestamp from bmix file
string GetBlaiseVersionNumber(); // Current Blaise version on server
string GetServerParkName(); // From registry info
string UpdateWCFservice(); // Replace service dll with newer version
string UpdateBlaiseLicenseInfo(license key, licensee, activation code);
bool RestartBlaiseService(out string logMsg); // Restart Blaise 5 services on request
```

Some operations are only relevant on backend/common servers:

```
string UploadBpkgFil(byte array, file name, file date);
string ServerManagerInstallation(destination server name, port number, server park name, user, password,
package file name, form type); // Install, frontend/common server
string UnInstallSurvey(destination server name, server park name, instrument ID); // Uninstall,
frontend/common server
string RemoveBackendInfo(survey name, form type); // Remove e.g. remote bdix and other files
```

Some operations are only relevant on frontend/common servers:

```
string GetSurveyBlaiseVersion(survey name); // Blaise version of instrument (info not available on
backend)
string RunBlaiseHttpInstallScript(survey name); // Install security patches
```

## 4.2 Client Application

The client is a console application programmed in C#.

It implements the knowledge of the various environments and the servers belonging to each environment. Because the WCF service is installed as an exact copy on each of the involved servers, connecting to a service can be done by using a common method and only providing the needed endpoint (server name) to the specific server.

The client application can perform operations on all servers, on one server, on an environment or on a subset of servers or environments.

Currently, implemented operations comprise:

- Overview over all servers, including status info
- Overview over servers and environments belonging to the BSS
- Overview over an environment, including the installed surveys
- Overview over an installed survey, for each server
- Installation of a survey in one or more environments (BSS only)
- Update of the WCF Service on all servers
- Update Blaise license keys on one or all servers
- Running a test suite to produce a report on the health of the service(s)
- Remove a survey from an environment

As an example, figure 3 shows an overview of all servers and their properties.

### Oversight over RAMBLA-servere (i alt 17 stk.)

Oversigt genereret: 12. februar 2020 kl. 11:05

#### SERVERE

servernavn	status, service	OS	Blaisever.	serverpark	servertype
backendtest1	RamblaServices.dll, build: 10-01-2020 16:59:22	Windows Server 2008 R2 Standard	5.6.9.2082	LocalDevelopment	VIRK+LINK/BACKEND
frontendtest1	RamblaServices.dll, build: 10-01-2020 16:59:22	Windows Server 2016 Standard	5.6.9.2082	Production	VIRK/FRONTEND
backendprod1	RamblaServices.dll, build: 10-01-2020 16:59:22	Windows Server 2008 R2 Standard	5.5.3.1862	LocalDevelopment	VIRK+LINK/BACKEND
frontendprod1	RamblaServices.dll, build: 10-01-2020 16:59:22	Windows Server 2016 Standard	5.5.3.1862	Production	VIRK/FRONTEND
linkprod1	RamblaServices.dll, build: 10-01-2020 16:59:22	Windows Server 2016 Standard	5.5.7.1883	Production	LINK/FRONTEND
internprod	RamblaServices.dll, build: 10-01-2020 16:59:22	Windows Server 2016 Standard	5.6.9.2082	Production	INTERN/COMMON
blaisetest2k	RamblaServices.dll, build: 10-01-2020 16:59:22	Windows Server 2012 R2 Standard	5.6.9.2082	Production	CLASSIC/COMMON
blaiseprod3	RamblaServices.dll, build: 10-01-2020 16:59:22	Windows Server 2012 R2 Standard	5.5.11.1902	Production	CLASSIC/COMMON
blaiseprod4	RamblaServices.dll, build: 10-01-2020 16:59:22	Windows Server 2016 Standard	5.6.5.2055	Production	CLASSIC/COMMON

Fig. 3: Generated overview of servers by name, version of service, version of operating system, installed version of Blaise, name of server park and supported type(s) of forms and server.

Another example (figure 4) shows a list of surveys in a specific environment.

## VIRKTEST - frontendtest1 / backendtest1 (VirkBlanket)

Oversigt genereret: 12. februar 2020 kl. 15:54

### BLANKETTER

Blanket	ID	Inst.dato	Forb.dato	Blaisever.	Backend
bba2019a	5188651a-6ab2-4987-b209-8678b8697004	06-05-2019 16:45:05	06-05-2019 16:41:55	5.5.3.1862	OK: bba2019a, ...
bba2019b	dde95a45-4935-4bfb-b07e-ac5cb8fea61a	19-11-2019 09:44:17	19-11-2019 09:42:44	5.5.8.1886	OK: bba2019b, ...
blaisetstv5	e0682b17-3241-4239-95ae-4200254e9b09	01-10-2019 16:16:52	20-05-2019 12:41:06	5.5.3.1862	OK: blaisetstv5, ...
doi	86451a56-1e58-4f13-b2c3-918ec5465fe8	27-02-2019 11:57:35	27-02-2019 11:21:33	5.5.3.1862	OK: doi, ...
doi2019b	968afd90-0d15-4d49-8873-b1077d4e9cbc	10-12-2019 11:22:56	10-12-2019 11:14:20	5.5.8.1886	OK: doi2019b, ...
ipo	58ee31bb-5d00-4f63-9911-6eba4f507512	29-01-2020 17:57:44	29-01-2020 17:56:59	5.5.8.1886	OK: ipo, ...
ipo2020a	b9952956-bb8e-4230-98de-8dff7cc2ee65	29-01-2020 18:20:07	29-01-2020 18:17:11	5.5.8.1886	OK: ipo2020a, ...
ipo2020b	22733f6a-83ef-49b1-ae78-162c7dcf1332	04-02-2020 11:32:37	04-02-2020 11:31:34	5.5.8.1886	OK: ipo2020b, ...
lvs	5bc0d93f-b8e9-4639-8a47-652fd5ad87c5	11-12-2019 13:27:03	11-12-2019 13:23:06	5.5.8.1886	OK: lvs, ...

Fig. 4: Generated overview of surveys in a specific environment (frontend/common) by name, instrument ID, install date, prepare date of instrument, instrument version of Blaise and status of backend (if relevant).

Last example (figure 5) shows a list of a surveys of name starting with “doi” installed on all test servers.

Oversigt genereret: 13. februar 2020 kl. 11:22

BLANKETTEN 'doi\*' PÅ TEST-SERVERE

Server	Navn	ID	Inst.dato	Forb.dato	Blaisever.	Servertype	Blankettype	Miljø
backendtest1	doi	86451a56-1e58-4f13-b2c3-918ec5465fc8	27-02-2019 11:57:41	27-02-2019 11:21:32		BACKEND	VirkBlanket	VIRKTEST
backendtest1	doi2019b	968afd90-0d15-4d49-8873-b1077d4e9cbc	10-12-2019 11:22:58	10-12-2019 11:14:20		BACKEND	VirkBlanket	VIRKTEST
frontendtest1	doi	86451a56-1e58-4f13-b2c3-918ec5465fc8	27-02-2019 11:57:35	27-02-2019 11:21:33	5.5.3.1862	FRONTEND	VirkBlanket	VIRKTEST
frontendtest1	doi2019b	968afd90-0d15-4d49-8873-b1077d4e9cbc	10-12-2019 11:22:56	10-12-2019 11:14:20	5.5.8.1886	FRONTEND	VirkBlanket	VIRKTEST
backendtest1	doi2019b	968afd90-0d15-4d49-8873-b1077d4e9cbc	10-12-2019 11:23:20	10-12-2019 11:14:20		BACKEND	LinkBlanket	LINKTEST
linktest1	doi2019b	968afd90-0d15-4d49-8873-b1077d4e9cbc	10-12-2019 11:23:18	10-12-2019 11:14:20	5.5.8.1886	FRONTEND	LinkBlanket	LINKTEST
interntest	doi	86451a56-1e58-4f13-b2c3-918ec5465fc8	01-03-2019 10:37:19	27-02-2019 11:21:33	5.5.3.1862	COMMON	InternBlanket	INTERNTEST
interntest	doi2019b	968afd90-0d15-4d49-8873-b1077d4e9cbc	11-12-2019 15:48:02	10-12-2019 11:14:20	5.5.8.1886	COMMON	InternBlanket	INTERNTEST

Fig. 5: Generated overview of surveys named “doi\*” by server name, survey name, instrument ID, install date, prepare date of instrument, instrument version of Blaise, type of server and form and name of environment.

### 4.3 GUI – Graphical User Interface

The services are available to the users through an interactive Manipula program that exhibits the relevant subset of the services.

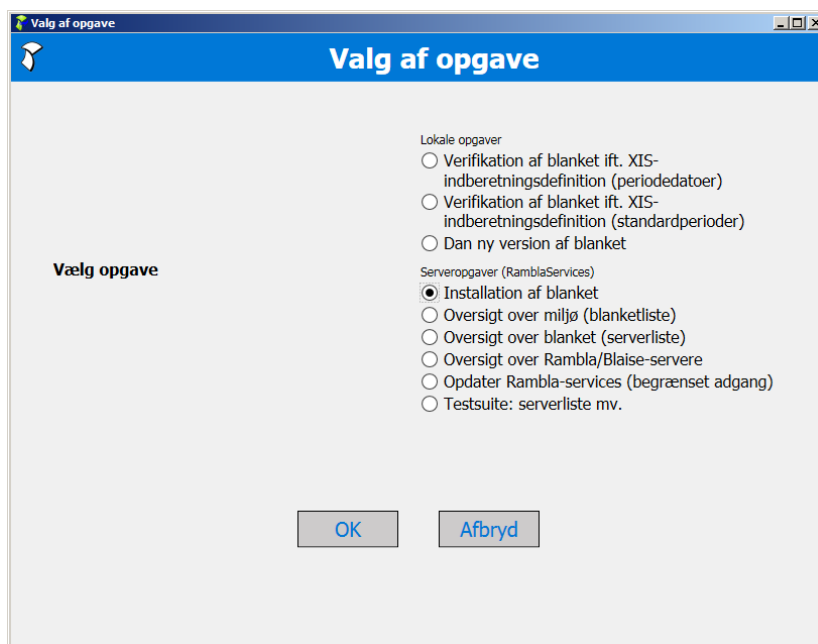


Fig. 6: Selection of tasks<sup>2</sup>

The GUI gives access to a collection of local and server oriented tasks. Local tasks comprise utilities related to development of questionnaires. For example, to verify that a developed questionnaire is in accordance with the standard templates and the backend database, i.e. that data collected with the instrument can be exchanged with it.

<sup>2</sup> “Rambla” is an acronym for *Framework System for Blaise Form Reports in Business Surveys*. “Blanket” is Danish for Form Report (questionnaire, survey).



The task chosen may lead to further dialogues. For example, installation of a survey leads to selection of the environment (or subset of environments) and to selection of a package file.

Afterwards, the task is transferred to the Client program to carry out the subtasks needed to install the survey in the requested environment(s). I.e., the tasks of uploading the package file, possibly removing an old version, installing the survey, creating remote BDIX, and modifying the installed survey.

## 5. Current status

The management related services

- WCF Service is installed on all the servers
- Questionnaire designers can install surveys in test as well as production environments.
- January 2020 we were able to renew the license keys and activation codes for all the servers in our Blaise server portfolio automatically.
- We can use the extracted lists of servers including e.g. currently installed Blaise version to help planning our upgrade procedures. For example, all test servers are now (February 2020) running Blaise 5.6.9, while we are still planning to upgrade production servers as well.
- It was quite fast and easy to develop the user interface in Manipula, though interactive Manipula still performs relatively slow.

The overall progress of converting questionnaires

- By February 2020, seven questionnaires for business surveys have been developed in Blaise 5 and put into production
- Eight more Blaise 5 questionnaires are under development
- Still, approx. 50 questionnaires must be converted before 2023

## 6. Future developments

We are planning to incorporate the above mentioned client program into an existing, general toolbox for management of Business Surveys.

The toolbox is running on a separate server and has finely grained access control implemented. Therefore, it should be possible to restrict access to the specific services to users that really need them and thus improve the general security.

Also, work is carried out in order to automate generation of Blaise 5 questionnaires that are in accordance with our background database for business form reports.

## 7. References

[BSS] Leif Bochis Madsen: Implementation of Blaise 5 web questionnaires into an existing Business Survey Management System, in: *Proceedings of the 18<sup>th</sup> International Blaise Users Conference*, Baltimore 2018, [https://www.blaiseusers.org/2018/papers/5\\_5.pdf](https://www.blaiseusers.org/2018/papers/5_5.pdf)