

# Bifrost system documentation

Redpill Linpro AS

3. februar 2009

### **Sammendrag**

Bifrost is a printing system developed by Redpill Linpro. It is licenced under the GNU GPL version 2 licence. The solution uses magnetic swipe cards, card readers and printing quotas. Users only have one printer on their computer and can pick up the printouts on any printer in their environment. The solution works with Linux, Windows and Mac OS X and needs Postscript printers.

# Innhold

<b>1</b>	<b>About Bifrost</b>	<b>2</b>
1.1	Introduction . . . . .	2
1.2	License . . . . .	2
<b>2</b>	<b>Solution description</b>	<b>3</b>
2.1	Architecture overview . . . . .	3
2.2	Solution model . . . . .	3
<b>3</b>	<b>Technical description</b>	<b>6</b>
3.1	Card readers . . . . .	6
3.1.1	USB card readers . . . . .	6
3.1.2	Network bridge . . . . .	6
3.1.3	SSH login . . . . .	6
3.1.4	Web interfaceparameter . . . . .	6
3.1.5	SNMP . . . . .	6
3.1.6	Reading card data . . . . .	7
3.2	Bifrost Back-end . . . . .	7
3.2.1	Data from swiped cards . . . . .	7
3.2.2	Identifying card readers . . . . .	7
3.2.3	Print messages . . . . .	7
3.2.4	Autocancel . . . . .	7
3.3	Print quota . . . . .	7
3.3.1	Initial quota . . . . .	8
3.3.2	Calculated quota . . . . .	8
3.3.3	Quota weighting . . . . .	8
3.4	OpenLDAP . . . . .	8
3.4.1	Schemas . . . . .	8
3.5	CUPS . . . . .	9
3.5.1	Authentication . . . . .	9
3.5.2	Print queues . . . . .	9
3.5.3	Back-ends . . . . .	9
3.6	Samba . . . . .	10
3.6.1	printcap . . . . .	10
3.6.2	Authentication . . . . .	10
3.6.3	Windows Vista . . . . .	10
3.7	Localization and translation . . . . .	10
3.7.1	Gettext . . . . .	11

# Kapittel 1

## About Bifrost

“Bifrost” is the name of the printing solution developed by Linpro AS for the Møre and Romsdal county council in Norway.

### 1.1 Introduction

Before Bifrost, you had to register every printer in every school with the appropriate driver for each printer. With Bifrost, adding one printer on your PC enables you to perform the printing on all printers you have physical access to.<sup>1</sup> Bifrost supports Linux, Windows and MAC OS X.

When you send a document to this printer, the print job is submitted to a generic print queue. Using the magnetic card they have received, users can collect their printout by swiping the card in the card reader connected to the printer they want to use. The document is printed on the printer. ((If you have not received a card, contact administration).

The solution architecture allows you to print your print job on any of the printers connected to the Bifrost server, including printers in other buildings or other schools.

### 1.2 License

Bifrost is licensed in accordance with GNU GPL version 2<sup>2</sup>. A copy of the license can be found in the “LICENSE” files in the project catalog root.

---

<sup>1</sup>The printers must be connected to the Bifrost system

<sup>2</sup>see GNU GPL version 2 on <http://www.gnu.org/licenses/gpl-2.0.html>

## Kapittel 2

# Solution description

### 2.1 Architecture overview

The solution consists of one print server that also controls quotas, cards and card reader, called the “Bifrost” server. All printers are registered on this print server. This server also has two virtual queues. One queue is called “followme”, and is added by all users. The other, “followme-queue”, is the queue where jobs reside until they are printed or cancelled. When submitted to “followme”, jobs will be also be submitted to “followme-queue”, in order to avoid jobs piling up locally on the client when using IPP. All physical printers are hidden from ordinary users, so all printouts must be submitted via the followme queue.

All printers are connected to a card reader; a small embedded computer with a USB card reader, which submits card data to the Bifrost server.

The Bifrost server identifies the card reader that submitted the card data and which printer it is connected to. The server also locates the card in the LDAP database, identifying the card’s registered user.

Next, all print jobs submitted by that user are sent to the printer.

### 2.2 Solution model

Figure 2.1 displays the system components and connections/dataflow. The figure shows client computers, both administration computers and student computers using the solution. They communicate with backend CUPS to print documents, either directly using IPP or via Samba. The printers are hidden behind the backend CUPS system. CUPS keeps track of user print quotas by counting the number of pages in the PostScript document.

Using an Apache web interface, users can control their own jobs and view their print quota status. Administrators can tag cards as lost /found, and manage quotas and card readers.

Autocancel automatically deletes old print jobs from the queue.

The card reader is connected to the OpenWRT box, submitting a message to the card reader server when a card is swiped through the reader. This server identifies which card reader was used, and moves all the user’s print jobs from the followme queue to the card reader printer. The OpenWRT boxes can be monitored using SNMP, from e.g. HP OpenView.

LDAP forms containing user cards and quotas are stored in LDAP on Bifrost Backend, replicating user data from an existing LDAP server.

NOTE: The first version does not support LDAP replication, since the backend application does not support printing to one server and reading from another. Furthermore, network latency and replication may keep updates and new LDAP objects from becoming available immediately in the web interface.

Figur 2.1: Solution model

## Kapittel 3

# Technical description

This chapter describes the technical solution.

### 3.1 Card readers

Bifrost uses **ASUS WL-500G Premium** wireless routers with a modified version of OpenWRT<sup>1</sup> card readers. These units have a USB interface connected to a MAGTEK USB card reader with keyboard interface.

#### 3.1.1 USB card readers

USB card readers must be reprogrammed in order to submit card data as **ALT-ASCII**. The card readers read all the three tracks on cards that are swiped.

#### 3.1.2 Network bridge

The ASUS boxes are programmed to bridge the traffic between the WAN port and the router's first LAN port. This enables you to connect to the printer on this port in order to assign a specific IP address from DHCP to the printer. This reduces the number of network points for each printer by half.

#### 3.1.3 SSH login

When rolling out the card readers, it is **required** to add an official SSH key to allow card reader management. Password login is blocked.

#### 3.1.4 Web interfaceparameter

Web interface is not activated.

#### 3.1.5 SNMP

SNMP is activated. You can set the "Community" phrase when generating the card reader firmware.

---

<sup>1</sup>OpenWRT is a Linux distribution for embedded systems. <http://openwrt.org/>



### 3.1.6 Reading card data

A background process reads data from cards that are swiped, and sends the data to Bifrost Back-end. This process must be run continuously.

## 3.2 Bifrost Back-end

Bifrost Back-end is part of the Bifrost server, handling queries from the card readers and auto-cancelling of old print jobs.

### 3.2.1 Data from swiped cards

Card readers submit data from swiped cards as three parameters, one for each track on the card. If the card reader is connected to a printer, the user associated with the card is identified and all the user's print jobs will be submitted to that printer. If the card reader is an administration card reader<sup>2</sup>, the card data will be cached for administration in the web interface.

### 3.2.2 Identifying card readers

Card readers are identified by MAC address. They are assigned a dynamic IP address from DHCP. When a query is received, an inquiry of the IP address is made in ARP table. This requires that the server and the card readers are located on the same subnet.

If they are not, it will not be possible to make an inquiry into the ARP table. By modifying the configuration (see section ??), card readers are allowed to submit with one "macaddress" parameter instead of the ARP table inquiry. Please note that this functionality can only be used if the ARP table inquiry fails.

### 3.2.3 Print messages

Errors during CUPS<sup>3</sup> printing are reported to the Bifrost Back-end, and an e-mail is submitted to the print job owner/user and to the system administrator.

### 3.2.4 Autocancel

Old print jobs will be cancelled automatically. A suggested maximum lifetime of a print job is 7 hours, i.e. approximately one school day. This lifetime can be modified in the configuration file (see section ??).

Autocancel is useful since old print jobs are usually forgotten, and unwanted the next time the user submits something to the printer. When a print job is cancelled, an e-mail is submitted to print job owner.

## 3.3 Print quota

Bifrost supports print quotas.

---

<sup>2</sup>i.e. any card reader not connected to a printer

<sup>3</sup>CUPS is the print server used. <http://www.cups.org>

### 3.3.1 Initial quota

All users receive an initial quota with their first card. The initial quota is usually 500.

### 3.3.2 Calculated quota

The cost of printing a document depends on three factors.

- The number of pages
- The number of copies
- Quota weighting on the printer

These three values are multiplied in order to calculate the quota-related cost of printing for the user.

### 3.3.3 Quota weighting

A separate quota weight may be assigned to each printer. Usually, this will be 1, but when the user selects the duplex option on the printer, it may be reasonable to reduce the weight to 0.5 or 0.75. For printing in colour, you may want to increase the quota weight to 2. These are only suggestions.

The weighting is specified as a parameter to the CUPS back-end driver (see section 3.5.3).

## 3.4 OpenLDAP

The system uses LDAP for inquiries and storing data. The system is developed with OpenLDAP in mind, but other kinds of LDAP databases can be used without major modifications.

### 3.4.1 Schemas

Bifrost uses some schemas that are not included by default in OpenLDAP, some of these have been specifically designed for Bifrost. These schemas must be copied into the schema catalog (`/etc/ldap/schema/` for OpenLDAP) and included in the LDAP (`/etc/ldap/slapd.conf` configuration file for OpenLDAP).

- User information uses the **posixAccount** object class, which is included in OpenLDAP
- Group information for users if they are organized in groups inside the same branch use the **posixGroup** object class, which is included in OpenLDAP
- User information for Samba authentication uses the **sambaSamAccount** object class, which is included in Samba
- Print quotas use the **fmPrinterQuota** object class, which has been developed for Bifrost.

- Card readers use the **fmCardReader** object class, which has been developed for Bifrost.
- Cards use the **magneticCard** object class, which has been developed for Bifrost.

There is also a schema for printers, which is not in use. This schema has an object class called **cupsPrinter**. Schemas developed for Bifrost, and a copy of the Samba schema, are found in `src/LDAP/schema/`.

## 3.5 CUPS

Bifrost uses CUPS as a print solution. CUPS authenticates users against LDAP, and uses back-ends to manage print jobs to support print quotas and queuing.

### 3.5.1 Authentication

CUPS authenticates users against LDAP with PAM. CUPS requires users to be available in the UNIX environment; as a consequence `nsswitch.conf` must be set up with LDAP support.

### 3.5.2 Print queues

You need to create a print queue to authenticate users and receive print jobs. Call this queue “followme”. This queue forwards the print jobs to another queue, a print class you also need to create, to cache print jobs from the “followme” queue. This class should be empty, in other words a class with no printers associated to it. Call this class “followme-queue”.

See the section about CUPS “route” backend (3.5.3) for more information on the relationship between the “followme” and “followme-queue” queues.

Bifrost requires at least one printer and one class. In addition, all physical drivers must be included in CUPS, with “followme” back-end.

### 3.5.3 Back-ends

A couple of CUPS back-ends have been developed for Bifrost. One of them handles user print quotas, the other submits print jobs to the ??vente-køen??.

#### followme

The “followme” CUPS back-end driver must be assigned to **all** printers that are used by the solution. This driver is used for checking and updating user print quotas when documents are submitted to a printer for printing.

A printer can be added in CUPS in the normal way, but you need to modify the printer’s URI. The format of the new printer URI:

```
followme:/1/<original URI>
```

IMPORTANT: On systems running CUPS as **root** (e.g. Ubuntu Gutsy), the back-end drivers “http”, “ipp”, and “lpd” are normally owned by root with “700” rights. This means that they can only be

executed by root. CUPS will run back-ends as an ordinary user with fewer rights, for instance as **lp**, if possible. As a consequence, it is important that the rights on the “followme” back-end is set to “700” and owned by root, since it must be able to submit print jobs to other back-ends, such as “ipp”.

Ubuntu Feisty does not have the same problem with rights, and CUPS is run under the **cupsys** user.

### **route**

The “route” CUPS back-end driver must be used on the print queue receiving user print jobs, called “followme”. This back-end driver takes a parameter which is the name of the print queue used for caching print jobs until they are submitted to a printer or are cancelled.

The print queue URI is set as follows:

```
route:/followme-queue
```

In this example, the “followme-queue” queue is used for caching print jobs.

## **3.6 Samba**

You can also use Samba for printing, as an alternative to IPP, which is used by CUPS.

### **3.6.1 printcap**

You should use a special **printcap** file that only exposes the “followme” print queue to users.

### **3.6.2 Authentication**

Samba authenticates users against LDAP.

### **3.6.3 Windows Vista**

Windows Vista has a broken implementation of IPPS, IPP over SSL. As a consequence, you need to use Samba, and add the printer as a “shared printer, by name”. See the user documentation for information on how to add the printer in Windows Vista.

## **3.7 Localization and translation**

The web pages are available in both English and Norwegian (bokmål). The web browser determines which language version to use, by submitting the “Accept-Languages” header. Norwegian is selected when the web browser has been configured to accept “no” as language. English is the default language if the browser does not submit this header.

### 3.7.1 Gettext

The translation files are found under `src/Followme/lib/Followme/I18N/`, with file extension “.po”. The compiled files have file extension “.mo”.

The poEdit<sup>4</sup> editor is suitable for translating these files. The editor will update the mo file when the po file is stored.

---

<sup>4</sup>poEdit can be used for editing translation files. <http://poedit.sourceforge.net/>