

Bifrost Systemdokumentasjon

Linpro AS

10. juli 2008

Sammendrag

Bifrost er et utskriftssystem utviklet av Linpro AS for Møre og Romsdal Fylkeskommune og er utgitt under betingelsene til GNU GPL versjon 2. Løsningen benytter magnetkort, kortlesere og utskriftskvoter. Brukere forholder seg til én skriver på datamaskinen sin og kan hente utskriften sin på hvilken som helst skriver på skolen. Løsningen fungerer med Linux, Windows og MAC OS X og benytter seg av Postscript.

Innhold

1	Innledning	2
1.1	Introduksjon	2
1.2	Lisens	2
2	Løsningsbeskrivelse	3
2.1	Overordnet arkitektur	3
2.2	Løsningsskisse	3
3	Teknisk beskrivelse	6
3.1	Kortlesere	6
3.1.1	USB-kortlesere	6
3.1.2	Nettverksbro	6
3.1.3	SSH-innlogging	6
3.1.4	WEB-grensesnitt	6
3.1.5	SNMP	6
3.1.6	Lesing av kortdata	7
3.2	Bifrost Backend	7
3.2.1	Innleste kort	7
3.2.2	Identifisere kortlesere	7
3.2.3	Utskriftsfeilmeldinger	7
3.2.4	Autoavbryt	7
3.3	Utskriftskvote	7
3.3.1	Initiell kvoteverdi	8
3.3.2	Kalkulert kvote	8
3.3.3	Kvotevekting	8
3.4	OpenLDAP	8
3.4.1	Skjemaer	8
3.5	CUPS	9
3.5.1	Autentisering	9
3.5.2	Utskriftskøer	9
3.5.3	Backends	9
3.6	Samba	10
3.6.1	printcap	10
3.6.2	Autentisering	10
3.6.3	Windows Vista	10
3.7	Lokalisering og oversetting	10
3.7.1	Gettext	11

4	Konfigurasjon	12
4.1	CUPS Backend	12
4.1.1	followme	12
4.1.2	route	13
4.2	Bifrost Backend	14
4.2.1	Autentisering	14
4.2.2	Realms	16
4.2.3	E-postadresser	16
4.2.4	Administrasjon	17
4.2.5	Tilgangskontroll til backend	17
4.2.6	Kortinnlesing	18
4.2.7	LDAP	18
4.2.8	Utskriftskvoter	20
4.2.9	Kort	20
4.2.10	Kortstatus	20
4.2.11	Kortlesere	21
4.2.12	Tidssone	22
4.2.13	Dato- og tidsformat	22
4.2.14	Utskriftskø	22
4.2.15	Epost - sending	23
4.2.16	Autoavbryt	23

Kapittel 1

Innledning

“Bifrost” er navnet på utskriftssystemet utviklet av Linpro AS for Møre og Romsdal Fylkeskommune.

1.1 Introduksjon

Tidligere har man måtte legge til alle skriverne på skolen med riktig driver for hver skriver. Med Bifrost trenger du bare å legge til én skriver på PC-en din, og dette gjør det mulig å skrive ut på alle skrivere du fysisk har tilgang til¹. Du kan benytte Linux, Windows eller MAC OS X for å skrive ut med Bifrost.

Når du skriver ut et dokument til denne skriveren blir utskriftsjobben lagt i en kø. Du som bruker skal ha fått tildelt et kort med magnetstripe (hvis ikke, ta kontakt med administrasjonen). Når du vil hente utskriften din, går du til nærmeste skriver og drar kortet i den tilknyttede kortleseren. Utskriften blir deretter umiddelbart skrevet ut på skriveren.

Det står en kortleser koblet til alle skriverne, og når du drar kortet ditt vil alle utskrifter du har sendt til “followme”-køen bli skrevet ut på den skriveren som er koblet til kortleseren du dro kortet i.

Grunnet arkitekturen av løsningen vil det være mulig å hente utskriften din på en hvilken som helst skriver tilknyttet Bifrost-tjeneren. Det kan være i en annen bygning eller en annen skole.

1.2 Lisens

Bifrost er utgitt under betingelsene til GNU GPL versjon 2². Du finner en kopi av lisensen i fila “LICENSE” på rota av prosjektetkatalogen.

¹Forbeholdt at skriverne er tilknyttet Bifrost-systemet

²GNU GPL versjon 2 finnes på <http://www.gnu.org/licenses/gpl-2.0.html>

Kapittel 2

Løsningsbeskrivelse

2.1 Overordnet arkitektur

Løsningen består av én utskriftstjener, som også administrerer kvoter, kort og kortlesere. Denne kaller vi for “Bifrost”-tjeneren. Alle skrivere er lagt inn på denne utskriftstjeneren. I tillegg har denne tjeneren to virtuelle køer. Den ene heter “followme”, som alle brukere må legge til, og den andre heter “followme-queue” som er køen hvor jobbene holdes helt til de sendes til utskrift eller blir kansellert. Når man skriver ut til “followme” vil jobbene bli skrevet ut på nytt til “followme-queue”, for å forhindre at klienten kører opp jobbene lokalt når de printer med IPP. Alle fysiske skrivere er skjult for vanlige brukere, så all utskrift må gå igjennom followme-køen.

Alle skrivere har tilkoblet en kortleser som er en liten embedded datamaskin med en USB-kortleser, som sender kortdata inn til Bifrost-tjeneren.

Bifrost-tjeneren finner ut hvilken kortleser som sendte kortdataene og hvilken skriver denne er koblet til. Videre leter tjeneren opp kortet i LDAP-databasen og finner ut hvilken bruker kortet er registrert på.

Dermed blir alle brukerens utskriftsjobber sendt til skriveren.

2.2 Løsningsskisse

Figur 2.1 viser komponenter og forbindelser/dataflyt i systemet. Figuren viser klientmaskiner som både maskiner for administrasjon og studenter som benytter seg av løsningen. De kommuniserer mot CUPS i backend for å skrive ut dokumenter, enten direkte over IPP eller via Samba. Skriverne er skjult bak CUPS-systemet i backend. CUPS holder telling på brukernes utskriftskvoter ved å telle antall sider i PostScript-dokumentet.

Via et webgrensesnitt fra Apache kan brukere administrere egne jobber og få oversikt over sin utskriftskvote. Administratorer kan merke kort som mistet/funnet, administrere kvotepåfyll og administrere kortlesere.

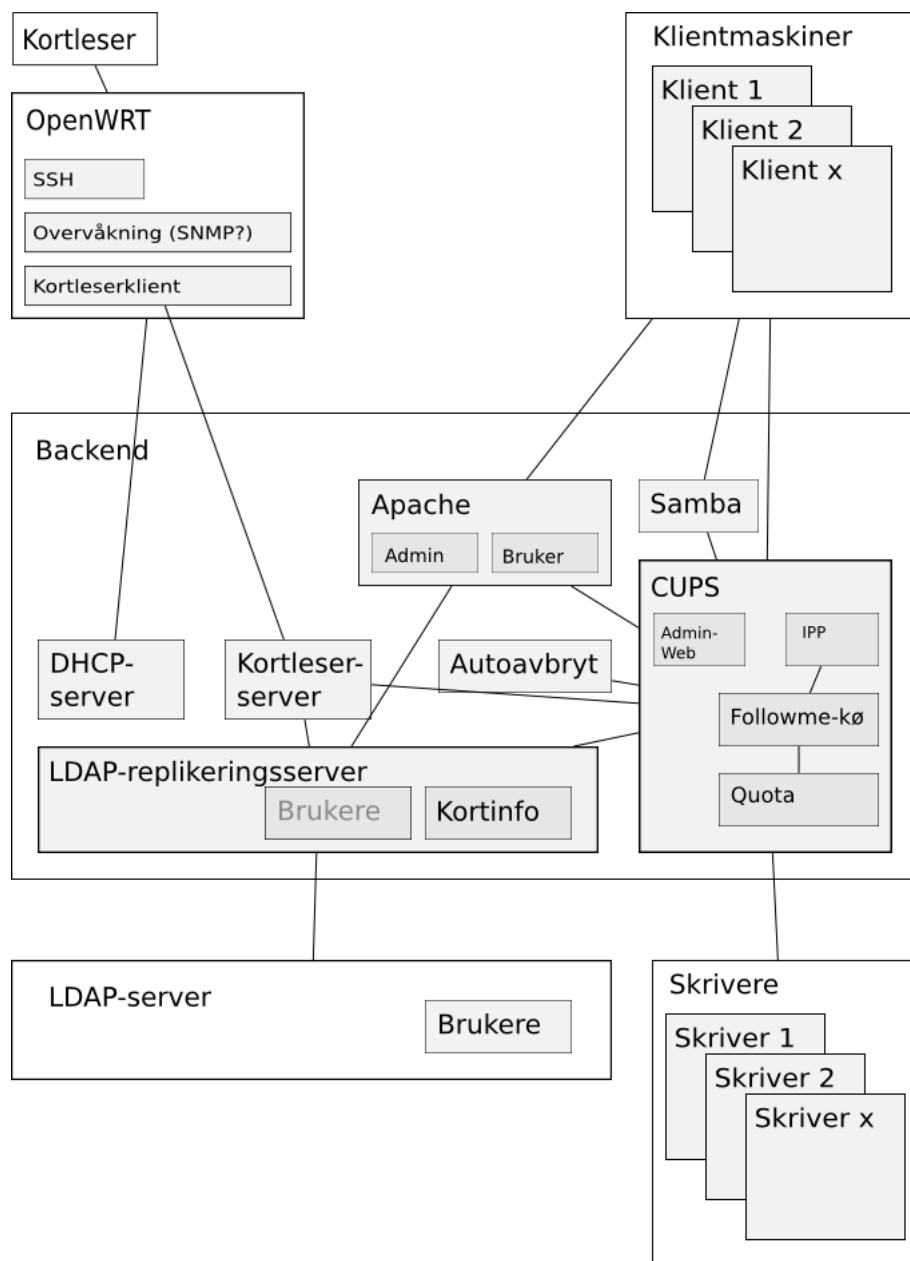
Autoavbryt sørger for automatisk fjerning av gamle utskriftsjobber fra køen.

Kortleseren er tilkoblet OpenWRT-boksen og sender en melding til kortlesertjeneren når et kort dras igjennom leseren. Denne tjeneren finner ut hvilken kortleser som ble benyttet og vil så flytte alle utskriftsjobber for denne brukeren fra followme-køen over til den aktuelle skriveren. OpenWRT-boksene kan

overvåkes med SNMP, fra for eksempel HP OpenView.

LDAP-skjema for brukernes kort og utskriftskvoter lagres i LDAP på Bifrost Backend, som replikerer brukerdata fra en allerede eksisterende LDAP-tjener.

OBS: I første versjon vil det ikke være støtte for replikering av LDAP da backend-applikasjonen ikke støtter skriving til én tjener og lese fra en annen. I tillegg kan latens i nettverket og replikering føre til at oppdateringer og nye LDAP-objekter ikke umiddelbart blir tilgjengelig i webgrensesnittet.



Figur 2.1: Løsningsskisse

Kapittel 3

Teknisk beskrivelse

Dette kapittelet beskriver den tekniske løsningen.

3.1 Kortlesere

Det benyttes trådløse rutere av typen **ASUS WL-500G Premium** med en modifisert versjon av OpenWRT¹ som kortlesere. Disse boksene har USB-grensesnitt hvor det er koblet til en USB-kortleser av typen MAGTEK med tastaturgrensesnitt.

3.1.1 USB-kortlesere

USB-kortleserne må programmeres om til å sende kortdataene som **ALT-ASCII**. Disse leser alle tre sporene fra kort som leses inn.

3.1.2 Nettverksbro

ASUS-boksene er programmert til å bridge trafikken mellom WAN-porten og første LAN-port i ruter. Dette gjør det mulig å koble til skriveren på denne porten slik at skriveren får sin egen IP-adresse fra DHCP. Dette halverer antall nettverkspunkter som må legges opp ut til hver skriver.

3.1.3 SSH-innlogging

Ved utrulling av kortlesere **må** det legges inn en offentlig SSH-nøkkel for å tillate administrasjon av kortleserne. Innlogging med passord er stengt.

3.1.4 WEB-grensesnitt

Web-grensesnitt er ikke aktivert.

3.1.5 SNMP

SNMP er aktivert. "Community"-frasen kan settes ved generering av firmware'n til kortleserne.

¹ OpenWRT er en Linux-distribusjon for embedded-systemer. <http://openwrt.org/>

3.1.6 Lesing av kortdata

En bakgrunnsprosess leser data fra kort som leses inn og sender dette til Bifrost Backend. Denne prosessen må kjøre til enhver tid.

3.2 Bifrost Backend

Bifrost Backend er en del av Bifrost-tjeneren som håndterer forespørsler fra kortleserne og autoavbryt av gamle utskriftsjobber.

3.2.1 Innleste kort

Kortlesere sender innleste kort som tre parametere, én for hver spor på kortet. Dersom kortleseren er tilknyttet en skriver vil brukeren av kortet letes opp og alle dens jobber blir sendt til skriveren. Dersom kortleseren er en administrasjonskortleser² vil kortdataene bli mellomlagret for administrasjon i web-grensesnittet.

3.2.2 Identifisere kortlesere

Kortlesere identifiseres på MAC-adresse. De tildeles en dynamisk IP-adresse fra DHCP. Når en forespørsel kommer inn gjøres det oppslag i ARP-tabellen på IP-adressen. Dette forutsetter at tjeneren og kortleserne er på samme subnett.

Dersom det ikke er tilfellet vil det ikke være mulig å slå opp i ARP-tabellen. Ved å endre i konfigurasjonen (se seksjon 4.2.6) kan man tillate at kortlesere sender med en parameter “macaddress” som brukes istedenfor oppslaget i ARP-tabellen. Bemerk at denne funksjonaliteten kun benyttes dersom oppslaget i ARP-tabellen feiler.

3.2.3 Utskriftsfeilmeldinger

Feil ved utskrift i CUPS³ blir rapportert til Bifrost Backend og e-post blir sendt ut til brukeren som eier utskriftsjobben og til systemadministrator.

3.2.4 Autoavbryt

Gamle utskriftsjobber vil automatisk bli kansellert. En passende maksimal levetid på utskriftsjobber er 7 timer, som er omtrentlig én hel dag på skolen. Denne levetiden kan endres i konfigurasjonsfilen (se seksjon 4.2.16). Dette gjøres fordi gamle utskriftsjobber mest sannsynlig er glemt, og derfor ikke ønsket neste gang brukeren skriver ut noe. En e-post sendes ut til eier av utskriftsjobben når en jobb kanselleres.

3.3 Utskriftskvote

Bifrost støtter utskriftskvoter.

² Alle kortlesere som ikke er tilknyttet en skriver er administrasjonskortlesere

³ CUPS er utskriftstjeneren som benyttes. <http://www.cups.org>

3.3.1 Initiell kvoteverdi

Alle brukere for en initiell kvoteverdi når de tilordnes sitt første kort. Denne er normalt 500.

3.3.2 Kalkulert kvote

Kostnaden ved å skrive ut et dokument er avhengig av tre faktorer.

- Antall sider
- Antall kopier
- Kvotevekting på skriveren

Alle disse tre verdiene er multiplisert for å finne tallet som brukeren trekkes for i kvote.

3.3.3 Kvotevekting

Hver skriver kan ha egen kvotevekt. Normalt vil denne være 1, men dersom man benytter seg av dupleksfunksjonen på skriveren vil man kanskje senke denne til 0.5 eller 0.75. Ved utskrift på en fargeskriver kan det tenkes at kvotevekten økes til 2. Dette er kun eksempler.

Vektingen settes som en parameter til CUPS backend-driveren (se seksjon 3.5.3).

3.4 OpenLDAP

Systemet benytter LDAP for oppslag og lagring av data. Systemet er utviklet med fokus mot OpenLDAP, men andre typer LDAP-databaser skal kunne benyttes uten store endringer.

3.4.1 Skjemaer

Det benyttes noen skjemaer som ikke følger med som standard i OpenLDAP, noen av disse er utviklet for dette formålet. Disse skjemaene må kopieres inn i skjemakatalogen (`/etc/ldap/schema/` for OpenLDAP) og inkluderes i konfigurasjonsfilen til LDAP (`/etc/ldap/slapd.conf` for OpenLDAP).

- Brukerinformasjon benytter objektklassen **posixAccount**, som følger med OpenLDAP
- Gruppeinformasjon for brukerne dersom de er organisert i grupper innenfor samme gren benytter objektklassen **posixGroup**, som følger med OpenLDAP
- Brukerinformasjon for autentisering mot Samba benytter objektklassen **sambaSamAccount**, som følger med Samba
- Utskriftskvoter benytter objektklassen **fmPrinterQuota**, utviklet for Bifrost.
- Kortlesere benytter objektklassen **fmCardReader**, utviklet for Bifrost.

- Kort benytter objektklassen **magneticCard**, utviklet for Bifrost.

I tillegg finnes det et skjema for skrivere, men som ikke er tatt i bruk. Dette skjemaet har en objektklasse som heter **cupsPrinter**. Utviklede skjemaer for Bifrost, samt en kopi av Samba-skjemaet, finnes under `src/LDAP/schema/`.

3.5 CUPS

CUPS benyttes som utskriftsløsning i Bifrost. Den autentiserer brukere mot LDAP og benytter spesielle backends for å behandle utskriftsjobber for å støtte utskriftskvoter og køing.

3.5.1 Autentisering

CUPS autentiserer brukere mot LDAP med PAM. CUPS krever at brukerne er tilgjengelige i UNIX-miljøet, så `nsswitch.conf` må også settes opp med LDAP-støtte.

3.5.2 Utskriftskøer

Det må opprettes en utskriftskø for å autentisere brukere og ta i mot utskriftsjobber. Denne kaller vi for “followme”. Denne køen sender utskriftene videre til en ny kø, en utskriftsklasse som må opprettes, for å holde igjen utskrifter sendt fra “followme”-køen. Denne klassen skal være tom, altså en klasse uten assosierte skrivere. Vi kaller denne klassen “followme-queue”.

Se seksjon om CUPS “route”-backend (3.5.3) om forholdet mellom “followme”- og “followme-queue”-køene.

Vi må ha minst én skriver og én klasse for at Bifrost skal fungere. I tillegg må alle fysiske skrivere legges til i CUPS, med “followme”-backend.

3.5.3 Backends

Det er utviklet et par CUPS backends for Bifrost. Den ene håndterer brukernes kvoter ved utskrift og den andre er for å sende utskrifter til vente-køen.

followme

CUPS backend-driveren “followme” må knyttes til **alle** skrivere som ønsket brukt i løsningen. Denne brukes for å sjekke og oppdatere brukernes kvoter når dokumentene sendes til en skriver for utskrift.

En skriver kan legges til på samme måte i CUPS som tidligere, men man må gå inn og endre URI-en til skriveren. Den nye URI-en til skriveren skal være på formatet:

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

VIKTIG: På systemer som kjører CUPS som **root** (som Ubuntu Gutsy) er backend-driverne “http”, “ipp” og “lpd” normalt eid av root med rettighetene “700”. Det betyr at kun root kan eksekvere disse. CUPS vil kjøre backends som en annen bruker med mindre rettigheter, for eksempel som **lp**, hvis dette er mulig. Det er derfor

viktig at rettighetene på “followme”-backend settes til “700” og eies av root, fordi den må kunne sende utskriftsjobber til andre backends, som for eksempel “ipp”.

Ubuntu Feisty har ikke det samme problemet med rettigheter på filene, og CUPS kjøres under brukeren **cupsys**.

route

CUPS backend-driveren “route” må brukes på utskriftskøen som brukerne sender sine utskrifter til, som heter “followme”. Denne backend-driveren tar en parameter som er navnet på utskriftskøen som brukes for å holde igjen utskrifter, helt til de sendes til en skriver eller blir gamle og kansellert.

URI-en til utskriftskøen skal settes slik:

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

I dette eksempelet blir “followme-queue”-køen brukt for å holde igjen utskrifter.

3.6 Samba

Som et alternativ til IPP, som benyttes av CUPS, kan man benytte Samba til utskrift.

3.6.1 printcap

Det bør benyttes en spesiell **printcap**-fil som kun eksponerer “followme”-utskriftskøen til brukere.

3.6.2 Autentisering

Samba autentiserer brukere mot LDAP.

3.6.3 Windows Vista

Windows Vista har en brukken implementasjon av IPPS, IPP over SSL. Derfor må Samba brukes, og skriveren legges til som en “delt skriver, etter navn”. Se brukerdokumentasjonen for å legge til skriveren i Windows Vista.

3.7 Lokalisering og oversetting

Nettsidene er tilgjengelig på både engelsk og norsk bokmål. Det er nettleseren som bestemmer hvilket språk som skal benyttes ved å sende headeren “Accept-Languages”. Norsk får man når nettleseren er konfigurert til å akseptere språket “no”. Engelsk er forhåndsvalgt dersom nettleseren ikke sender denne headeren.

3.7.1 Gettext

Oversettelsesfilene finnes under `src/Followme/lib/Followme/I18N/` og har filendelse “.po”. De kompilerte filene har filendelse “.mo”.

Redigeringsprogrammet `poEdit`⁴ er godt egnet for å redigere disse filene. Denne vil oppdatere mo-filen når po-filen lagres.

⁴poEdit kan brukes for å endre oversettelsesfiler. <http://poedit.sourceforge.net/>

Kapittel 4

Konfigurasjon

Det er flere deler av Bifrost-løsningen som må konfigureres. Det meste er kort forklart i `INSTALL`-dokumentet som ligger på roten av prosjektkatalogen. I dette kapitlet presenteres konfigurasjonsfilene for CUPS Backend-drivere og Bifrost Backend.

4.1 CUPS Backend

Ta utgangspunkt i konfigurasjonsfilen som ligger under `src/CUPS/followme.yml`. Konfigurasjonsfilen forventes å ligge under rotkatalogen til CUPS, som i de fleste tilfeller er `/etc/cups/`, og hete **followme.yml**. Denne konfigurasjonsfilen er skrevet i YAML. Både “followme”- og “route”-backend benytter seg av samme konfigurasjonsfilen.

4.1.1 followme

Konfigurasjon for backend-driveren “followme”.

URL_BASE

Basis for URL til Bifrost backend. Denne vil trolig måtte settes til en ny verdi, da den ellers tilbakefaller til “http://localhost:3000”. Dersom applikasjonen er satt opp på rota med for eksempel Apache og FastCGI må denne konfigurasjonsparameteren settes til “http://localhost”.

URL_UPDATE_QUOTA

Absolutt URL til Bifrost backend for oppdatering av kvote. Denne vil tilbakefalle på “/backend/cups/decrement_quota”, prefikset med “URL_BASE”.

URL_ENOUGH_QUOTA

Absolutt URL til Bifrost backend for sjekking av brukerens kvote. Denne vil tilbakefalle på “/backend/cups/enough_quota”, prefikset med “URL_BASE”.

URL_REPORT_ERROR

Absolutt URL til Bifrost backend for rapportering av feil. Denne vil tilbakefalle på “/backend/cups/error”, prefikset med “URL_BASE”.

USER_AGENT

Navnet på brukeragenten som kaller på Bifrost backend. Denne vil tilbakefalle på “Bifrost CUPS Backend”. Denne behøver du ikke endre, ved mindre du har spesielle behov med tanke på for eksempel traversering igjennom proxy-tjenester eller logging av trafikk basert på agent.

TIMEOUT

Antall sekunder å maksimalt vente før en HTTP request mot Bifrost backend avbrytes. Denne vil tilbakefalle på “15” sekunder.

DEBUG

Settes til en **SANN** verdi for å skru på debugging. Denne vil tilbakefalle til en **USANN** verdi, som skruer av debugging.

```
---
BASE_URL:           "http://localhost:3000"
URL_UPDATE_QUOTA:   "http://localhost:3000/backend/cups/decrement_quota"
URL_ENOUGH_QUOTA:   "http://localhost:3000/backend/cups/enough_quota"
URL_REPORT_ERROR:   "http://localhost:3000/backend/cups/error"
USER_AGENT:         "Bifrost CUPS Backend"
TIMEOUT:            15
DEBUG:              0
```

4.1.2 route

Konfigurasjon for backend-driveren “route”.

LPR

Full sti til programmet **lpr** som brukes for utskrift.

DEBUG

Settes til en **SANN** verdi for å skru på debugging. Denne vil tilbakefalle til en **USANN** verdi, som skruer av debugging.

```
---
LPR:                "/usr/bin/lpr"
DEBUG:              0
```


4.2 Bifrost Backend

Ta utgangspunkt i konfigurasjonsfilen som ligger under `src/Followme/followme.yml`. Konfigurasjonsfilen forventes å ligge under roten hvor Bifrost backend ble installert, for eksempel `/usr/local/followme/`. Denne konfigurasjonsfilen er skrevet i YAML.

Noen konfigurasjonsparametere kan være unnlatt å nevne her¹. Det henvises derfor til **POD** i de forskjellige modulene som kan leses med **perldoc**.

4.2.1 Autentisering

Bifrost er konfigurert til å autentisere mot LDAP. Det er slik at brukere og administratorer er lagt i to forskjellige grener i LDAP-treet, og derfor er det konfigurert opp to realms, én for brukere og én for administrasjonen.

Autentisering er konfigurert under parameteren **authentication** i konfigurasjonsfilen. Gjør nødvendige endringer for ditt miljø, som **ldap_server** og **user_basedn**. Konfigurasjon av denne parameteren er forklart i Perl-modulen **Catalyst::Plugin::Authentication**².

store - roles

Konfigurasjonsparameteren **roles** inneholder en liste over roller/grupper som brukere i valgt realm tildeles. I Bifrost skiller vi bare på vanlige brukere (user) og administrasjonsbrukere (admin).

Denne konfigurasjonsparameteren benyttes av modulen **Followme::Authentication::Store::LDAP**, som er en utvidelse av standardmodulen til Catalyst for autentisering mot LDAP³.

```
authentication:
  default_realm: users
  realms:
    users:
      credential:
        class: Password
        password_field: password
        password_type: self_check
      store:
        class: '+Followme::Authentication::Store::LDAP'
        ldap_server: "ldap://followme.lab.linpro.no"
        start_tls: 0
        user_basedn: "ou=people,dc=mrvg,dc=no"
        user_filter: "( &(objectClass=posixAccount)(uid=%s) )"
        user_scope: one
        user_field: uid
        user_search_options:
          deref: always
      roles:
```

¹ Det er ønskelig å gjøre dokumentasjon mer komplett, vennligst meld i fra dersom noe mer bør inn her

² <http://search.cpan.org/perldoc?Catalyst::Plugin::Authentication>

³ <http://search.cpan.org/perldoc?Catalyst::Plugin::Authentication::Store::LDAP>

```

- user
admins:
  credential:
    class: Password
    password_field: password
    password_type: self_check
  store:
    class: '+Followme::Authentication::Store::LDAP'
    ldap_server: "ldap://followme.lab.linpro.no"
    start_tls: 0
    user_basedn: "ou=ansatt,dc=mrvg,dc=no"
    user_filter: "( &(objectClass=posixAccount)(uid=%s) )"
    user_scope: one
    user_field: uid
    user_search_options:
      deref: always
  roles:
    - admin

```

Brukere i samme gren

Dersom vanlige brukere og administrasjonsbrukere finnes i samme gren i LDAP-treet og er delt inn i grupper kan følgende konfigurasjon benyttes under **authentication**:

```

default_realm: ldap
realms:
  ldap:
    credential:
      class: Password
      password_field: password
      password_type: self_check
    store:
      class: LDAP
      ldap_server: "ldap://localhost"
      start_tls: 0
      user_basedn: "ou=people,dc=mrvg,dc=no"
      user_filter: "( &(objectClass=person)(uid=%s) )"
      user_scope: one
      user_field: uid
      user_search_options:
        deref: always
      use_roles: 1
      role_basedn: "ou=groups,dc=mrvg,dc=no"
      role_filter: "( &(objectClass=posixGroup)(memberUid=%s) )"
      role_scope: one
      role_field: cn
      role_value: uid
      role_search_options:
        deref: always

```

Bemerk at modulen **Followme::Authentication::Store::LDAP** ikke lenger brukes, men heller **LDAP**.

4.2.2 Realms

Her spesifiserer man hvilke realms, eller grupperinger av brukere, som det er mulig å velge mellom. Fordi denne modulen ble laget som en “workaround” for å støtte opp under oppdelingen av brukere og administrasjonsbrukere i forskjellige grener i LDAP-treet, er det ikke mulig å endre på denne konfigurasjonen for å legge til flere realms. Dette fordi malfilene for nettsidene kun støtter “admins” i tillegg til “default”, som i vårt tilfelle er “users”.

realms

En nøkkel/verdi-liste over realms. Nøkkelen må matche mot realms definert i **authentication** (se seksjon 4.2.1). Verdien er et menneskevennlig navn på realmet.

```
Model::Realms:
  realms:
    admins: Administration
    users: Users
```

4.2.3 E-postadresser

Konfigurasjon av e-postadresser og -utsending gjøres under parameteren **Model::Email**.

admin

Dette er e-postadressen til systemadministrator. Denne e-posten vil motta kopi av feilmeldinger som sendes ut til brukere.

from

Dette er e-postadressen som står som avsender på e-post som sendes ut fra Bifrost.

default_locales

Dette er det forhåndsvalgte språket for e-post som sendes ut dersom klienten som meldte fra om feilen ikke har spesifisert hvilket språk som skal benyttes. Dette kan være navnet på språket eller en liste med navn.

user_fmt

Dette er formatstrengen som brukes for å finne fram til brukernes e-postadresser. %s byttes ut med brukernavnet til brukeren. Alle brukere må kunne nås på samme domenenavn.

```
Model::Email:
  admin:          'Bifrost Admin <root@localhost>'
  from:           'Bifrost <root@localhost>'
  default_locales: 'no'
  user_fmt:       '%s@localhost'
```

4.2.4 Administrasjon

Konfigurasjonen for administrasjonsgrensesnittet spesifiserer hvilke roller som har tilgang med parameternavnet **require_roles**. Dette kan være navnet på rollen eller en liste med rollenavn.

```
Controller::Admin:
  require_roles: admin
```

4.2.5 Tilgangskontroll til backend

Konfigurasjonen for backend spesifiserer hvilke IP-adresser som har tilgang til backend-funksjonaliteten, som å sende inn kortdata og melde ifra om feil. Realmet **cardreaders** bestemmer hvor kortlesere kommer fra. Realmet **ez** bestemmer hvor portalløsninger kan komme fra; disse har anledning til å liste ut, legge til og slette jobber og liste ut kvote for brukere.

Subnett

subnets inneholder en liste over subnett som har tilgang. Disse defineres med nettverksadressen med CIDR-notasjon. Det er viktig at localhost (127.0.0.1/8) har tilgang for at CUPS skal kunne sjekke og oppdatere kvoter.

```
authentication:
  realms:
    cardreaders:
      credential:
        class:      '+Followme::Authentication::Credential::Exists'
      store:
        class:      '+Followme::Authentication::Store::Network'
      subnets:
        -
          network:   "127.0.0.1"
          cidr:      8
        -
          network:   "87.238.45.0"
          cidr:      26
    ez:
      credential:
        class:      '+Followme::Authentication::Credential::Exists'
      store:
        class:      '+Followme::Authentication::Store::Network'
      subnets:
        -
          network:   "87.238.43.90"
```

```

        cidr:      32
-
        network:   "83.243.223.74"
        cidr:      32
-
        network:   "83.243.223.71"
        cidr:      32

```

4.2.6 Kortinnlesing

Det er mulig å konfigurere mottaket for kortdata til å tillate kortlesere å sende sin MAC-adresse som parameter⁴ sammen med kortdataene.

Bemerk at denne adressen benyttes kun dersom oppslag i ARP-tabellen feiler.

allow_mac_address

Settes til en **SANN** verdi for å tillate egenspesifisert MAC-adresse sammen med kortdataene. Bemerk at aktivering av dette kan innebære en sikkerhetsrisiko.

```

Controller::Backend::Cardreader::Swipe:
  allow_mac_address: 1

```

4.2.7 LDAP

LDAP benyttes som datalager for brukere, kvoter, magnetkort og kortlesere.

host

URI til LDAP-tjeneren. Kan være på følgende former:

- hostname:port
- ldap://hostname:port
- ldaps://hostname:port
- ldapi://%2fvar%2flib%2fldap_sock

port settes til standard LDAP-port⁵, dersom den ikke er satt.

base

Base for søk i LDAP-databasen.

dn

Navnet på administratorbrukeren i LDAP, med tilgang til å opprette og endre på objekter.

⁴MAC-adressen setter med parameternavnet "macaddress"

⁵Standard port for LDAP er 389

password

Passordet til administratorbrukeren.

connection_class

Klassen som brukes for tilkobling til LDAP-tjenesten. Her brukes **Followme::LDAP::Connection** for å støtte filtrering basert på en felles verdi blant objektene.

use_cache

En **SANN** verdi skruer på gjenbruk av LDAP-tilkoblinger. En **USANN** verdi skruer dette av, som er forhåndsvalgt dersom annet ikke er angitt. Vær oppmerksom på at bruk av cache ikke fjerner tilkoblinger som er tatt ned, enten ved nettverksfeil eller eksplisitte nedkoblinger.

user_filter - field

I Møre og Romsdals tilfelle er filteret satt i **description**-feltet til brukerne, som forteller hvilken skole ansatte og studenter hører til.

user_filter - filter

Filteret som automatisk legges på hvert søk. **%s** byttes ut med verdien på feltet satt i **field** til brukeren som utfører søket.

I Møre og Romsdals tilfelle vil brukere uten description-felt blir inkludert i søk. Søk utført av administrasjonsbrukere uten description-felt gjøres blant alle brukere.

models

Modellene som benytter LDAP konfigureres med base for hvor søk skal starte. I tillegg kan det spesifiseres en egen klasse som benyttes for entitetene som søkes frem.

```
Model::LDAP:
  host:          "ldap://followme.lab.linpro.no"
  base:          "dc=mrvg,dc=no"
  dn:            'cn=admin,dc=mrvg,dc=no'
  password:      'changeme'
  connection_class: "Followme::LDAP::Connection"
  use_cache:     0
  user_filter:
    field:       description
    filter:      "( |( !(description=*)) (description=%s))"
  models:
    Users:
      base:      'ou=people,dc=mrvg,dc=no'
      entry_class: 'Followme::LDAP::Entry::User'
    Quota:
      base:      'ou=people,dc=mrvg,dc=no'
      entry_class: 'Followme::LDAP::Entry::Quota'
```

```

Cards:
  base:          'ou=cards,dc=mrvg,dc=no'
  entry_class:   'Followme::LDAP::Entry::Card'
Cardreaders:
  base:          'ou=cardreaders,dc=mrvg,dc=no'
  entry_class:   'Followme::LDAP::Entry::Cardreader'

```

4.2.8 Utskriftskvoter

Konfigurasjon for utskriftskvoter. Merk at disse konfigurasjonsparametrene kun benyttes ved førstegangs opprettelse av kvoter for brukerne. Deretter lagres disse verdiene i brukerens objekt i LDAP. Endringer i konfigurasjonen vil dermed bare påvirke nye brukere.

initialQuota

Spesifiserer hvor mye kvote en bruker får i starten, når brukeren får utdelt et kort eller ved oppstart av semesteret.

incrementalQuota

Spesifiserer hvor mye kvote en bruker får ekstra i løpet av en gitt periode.

OBS: Ikke implementert

incrementSchedule

Spesifiserer hvor lang en periode er, for når brukerne får ekstra kvote.

OBS: Ikke implementert

```

Model::Quota:
  initialQuota:    "500"
  incrementalQuota: "100"
  incrementSchedule: "1m"

```

4.2.9 Kort

Konfigurasjon av kortmodellen.

auto_sn_prefix

Spesifiserer en streng (eller et tegn) som autogenerated ID-er for kort prefixses med.

```

Model::Cards:
  auto_sn_prefix:  'c'

```

4.2.10 Kortstatus

Konfigurasjon for status på kort.

entity_class

Spesifiserer hvilken klasse som skal brukes som entitetsklasse for status-objektene.

default

Spesifiserer hvilken status i listen over **available** som skal være standard. Denne blir også den eneste statusen som anses som aktiv, mens alle andre vil deaktivere kortet slik at det ikke kan brukes for utskrift.

available

En nøkkel/verdi-liste over status som kan settes på et kort. Nøkkelen er et kort navn som bør inneholde bare tegn innenfor a-z. Verdien er en menneskevennlig tekst.

Standard vil bare **active** være tilgjengelig.

```
Model::Cards::Status:
  entity_class: 'Followme::Domain::Entity::Status'
  default:      active
  available:
    lost:       Lost
    active:     Active
    locked:     Locked
```

4.2.11 Kortlesere

Konfigurasjon for kortlesere.

ldap_order_by

Kortlesere sorteres etter dette feltet når de listes ut.

ldap_rdn_field

Feltet i kortleserobjekter som brukes i DN⁶ for kortlesere.

ldap_model

Navnet på modellen som skal brukes for tilkobling til LDAP. Dette navnet på være konfigurert som modell under **Model::LDAP** (se seksjon 4.2.7).

ldap_filters - find_all

LDAP-filer som brukes ved å liste ut alle kortlesere.

ldap_filters - find_by_field

LDAP-filer som brukes ved å hente ut én kortleser.

⁶Distinguished Name; unik adresse til et objekt i LDAP

ldap_filters - find_by_field

LDAP-filter som brukes ved søk etter kortlesere basert på et attributt.

```
Model::Cardreaders:
  ldap_order_by:  macAddress
  ldap_rdn_field: macAddress
  ldap_model:     Cardreaders
  ldap_filters:
    find_all:      '(objectClass=fmCardReader)'
    find_by_field: '( &(objectClass=fmCardReader)(%s=%s) )'
    search_by_field: '( &(objectClass=fmCardReader)(%s=%s*) )'
```

4.2.12 Tidssone

Konfigurasjon for tidssone for serveren og brukerne.

```
Model::TimeZone:
  name:  'Europe/Oslo'
```

4.2.13 Dato- og tidsformat

Konfigurasjon for dato- og tidsformat. Denne konfigurasjonen brukes når kort leses inn for administrasjon. For å endre andre konfigurasjoner enn formatet på dato og tid, som for eksempel lokalisering og tidssone, se Perl-modulen **Date-Time::Format::Strptime**⁷. Alle parametere som metoden “new” tar imot kan konfigureres her.

```
Model::Timestamp::Format:
  pattern:  '%T %d.%m.%Y'
```

4.2.14 Utskriftskø

Konfigurasjon for behandling av utskriftskø i bruker-webgrensesnitt.

queue

Navnet på utskriftskøen som skal brukes for å lete fram brukerens jobber. Dette må være samme navnet som på utskriftsklassen i CUPS som brukes for å holde jobber.

lp_move

Full sti til programmet “lpmove”, som brukes for å flytte jobber mellom utskriftskøer.

⁷<http://search.cpan.org/perldoc?DateTime::Format::Strptime>

datetime_format

Formatet på dato og tid som vises i bruker-webgrensesnittet for når jobbene ble sendt til utskrift.

```
Model::Jobs:
  queue:          'followme-queue'
  lp_move:        '/usr/sbin/lpmove'
  datetime_format: '%H:%M %d.%m.%Y'
```

4.2.15 Epost - sending

Konfigurasjon for utsending av epost. Her spesifiseres tegnsett og format på e-postene og adresse eller program for utsending.

sender - mailer

Denne kan settes til **SMTP** eller **Sendmail**. Ved å spesifisere Sendmail her, vil programmet **sendmail** benyttes.

sender - mailer_args

Dersom SMTP benyttes må **Host** være satt til adressen til SMTP-tjeneren. Det er også mulig å sette brukernavn og passord for innlogging på SMTP-tjeneren med **username** and **password**.

```
View::Email:
  default:
    content_type:  text/plain
    charset:      utf-8
  sender:
    mailer:        'smtp'
    mailer_args:
      host:        'mx.linpro.no'
```

4.2.16 Autoavbryt

Konfigurasjon for autoavbryt av gamle utskriftsjobber.

expire_time

Settes til antall timer en jobb kan vente på å bli skrevet ut. Standard er 7 timer.

```
Controller::Backend::Autocancel:
  expire_time: 7
```