

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

Fakulta elektrotechnická

katedra počítačové grafiky a interakce

Integrace CISCO web/XML services se softwarovou  
ústřednou Asterisk

Asterisk and CISCO web/XML services

**září 2015**

**Vypracoval:  
Vedoucí práce:**

**Jaroslav Smutek  
Ing. Ján Kučerák**

## Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce a používal jsem pouze literaturu v práci uvedenou.

Datum:

.....  
podpis autora

## **Anotace:**

Tato bakalářská práce se zabývá problematikou integrace web/XML služeb pro IP telefony firmy Cisco řady 79XX s softwarovou pobočkovou ústřednou Asterisk, které obohacují tyto telefony o další funkcionality a možnosti dostupné bez využití proprietárních ústředn firm Cisco, které jsou zde nahrazeny open source softwarovou pobočkovou ústřednou Asterisk.

Čtenář je dále seznámen s jednotlivými stavebními kameny integrace a technologií tomu napomáhajících. Integrace web/XML služeb s ústřednou Asterisk je demonstrována na integraci s externí databází kontaktů, ústřednou generovaným registrem zmeškaných, příchozích a odchozích hovorů a možností aktivní editace číslovacího plánu uživatelem pomocí změny volby rychlého vymočení. Vše je otestováno na IP telefonech Cisco 7962.

## **Klíčová slova:**

Web/XML služba, IP telefon, ústředna Asterisk, TFTP, SCCP, ODBC, číslicový plán, funkce v číslicovém plánu, konfigurace, modifikace uživatelem

## **Summary:**

This thesis deals with the integration of web/XML services for IP phones Cisco model line 79XX with software PBX Asterisk, this adds additional functionality and options available without usage of proprietary PBX from Cisco, proprietary PBX is replaced with open source software PBX Asterisk.

The reader is familiarized with each individual building blocks of integration a technologies needed for it. The integration of web/XML services with PBX Asterisk is demonstrated by integration of external phone contacts database, PBX generated register of missed, incoming and outgoing calls and by option to change dial plan by user presented in modification of speed dial. All is tested on IP phones Cisco model 7962.

## **Index Terms:**

Web/XML service, IP phone, PBX Asterisk, TFTP, SCCP, ODBC, dial plan, funktion in dial plan, configuration, modifications by user

# Obsah

Seznam použitých zkratk	6
1 Úvod	7
2 Úvod do dílčích prvků problematiky	8
2.1 Co je to Asterisk	8
2.1.1 Použití ústředny Asterisk v projektu	8
2.1.2 Instalace	8
2.1.2.1 Nutné vybavení	8
2.1.2.2 Způsob instalace	8
2.1.2.3 Průběh instalace	9
2.1.3 Moduly chan_skinny a chan_sccp_b	10
2.1.4 Konfigurace ústředny Asterisk a registrace telefonu k ústředně	11
2.2 SIP	12
2.3 IAX	12
2.4 TFTP	13
2.4.1 Využití TFTP v projektu	13
2.4.2 Instalace TFTP serveru a klienta	14
2.5 SCCP	15
2.6 Číslovací plán (dial plan)	16
2.6.1 Číslovací plán v ČR	16
2.6.2 Číslovací plán v projektu	16
2.6.2.1 Použití kontextů	16
3 Systém integrace ústředny Asterisk a webových služeb	18
3.1 Integrace databáze s ústřednou Asterisk	18
3.1.1 Přidání ústředny Asterisk jako uživatele databáze	18
3.1.2 Instalace a konfigurace ODBC	18
3.1.2.1 Instalace ODBC podpory v operačním systému	18
3.1.2.2 Instalace podpory ODBC v ústředně Asterisk	19
3.1.2.3 Konfigurace ODBC pro MySQL	19
3.1.2.4 Konfigurace modulu „res_odbc“	20
3.2 Systém registru hovorů generovaných ústřednou Asterisk	21
3.2.1 Generování registru hovorů	21
3.2.1.1 Funkce v func_odbc.conf	22
3.2.1.2 Využití funkcí v číslovacím plánu	22
3.2.2 Struktura databáze registru hovorů	24
3.3 Svázání databáze kontaktů s číslovacím plánem ústředny Asterisk	25

3.3.1 nastavení napojení databáze kontaktů s číslovacím plánem ústředny Asterisk	25
3.3.2 Struktura databáze telefonů a linek	26
3.4 Modifikace číslovacího plánu uživatelem	27
3.5 Web/XML služba pro Cisco IP telefony řady 79XX	28
3.5.1 Znáznornění komunikace klienta se službou	28
3.5.2 Web/XML služba	29
3.5.2.1 XML část služby pro IP telefon Cisco 79XX	29
4 Závěr	31
5 Seznam použité literatury	32
6 Přílohy	33
6.1. sip.conf	33
6.2. extension.conf	33
6.3. TFTP	34
6.4. SEP203706174FD3.cnf.xml	35
6.5 ODBC/MySQL	36
6.6 ODBC/Asterisk	36

# Seznam použitých zkratk

TFTP - Trivial File Transfer Protocol

SCCP - Skinny Client Control Protocol

XML - Extensible Markup Language

SIP - Session Initiation Protocol

IAX - Inter Asterisk eXchange

ODBC - Open Database Connectivity

DSN - Data source name

LAMP – Open source model of web service pack

# 1 Úvod

Tato práce se zabývá integrací softwarové pobočkové ústředny Asterisk a web/XML služeb pro IP telefony řady 79XX firmy Cisco. Tu to práci jsem rozdělil na několik dílčích celků. Cílem první části této práce je nejprve stručné popsání softwarové ústředny Asterisk a dalších technologií a protokolů s ní spojených. Podrobněji pak rozebírám jednotlivé problematiky, které mohutněji zasahují do tohoto tématu práce. Taktéž popisují jak zprovoznit softwarovou ústřednu Asterisk, dále také jak dosáhnout telefonního spojení, to je ukázáno na číslovacím plánu a SIP účtech.

Dalším dílčím prvkem úvodní části je zaměřena na konfiguraci IP telefonů Cisco řad 79xx a k tomu účelu nutnému TFTP serveru a jeho náležitostí, funkčnost TFTP serveru je ukázána na odeslání a přijmutí textového souboru s „Hello world“.

Po tomto dílčím prvku následuje přiblížení číslicového plánu a jak je využit v softwarové ústředně Asterisk. Tím uzavírám úvodní avšak neméně důležitý úsek mé práce.

Cílem druhé části podrobný rozbor integrace jednotlivých prvků mé práce do softwarové ústředny Asterisk, počínaje integrací externí databáze. Nato následuje a navazuje rozbor a vznik ústřednou generovaný registr zmeškaných, odchozích a příchozích hovorů. Jak toho je dosaženo pomocí funkcí použitých v číslicovém plánu a nato navazující komunikace číslicového plánu s databází skrze komunikační kanál jenž používám pro integraci s externí databází, k dosažení jednotnějšího celku. Po integraci softwarové ústředny Asterisk s externí databází je cílem další části samotný akt svázání databáze kontaktů s číslicovým plánem používaným v softwarové ústředně Asterisk. Závěrečným prvkem integrace je pak umožnění uživateli modifikovat číslicový plán v rozsahu editace rychlé volby vytáčení.

Cílem konečné části mé práce je rozbor web/XML části služby pro IP telefony Cisco řady 79XX. Cisco IP telefony obsahují jednoduchý prohlížeč, který je možné ovládat z klávesnice telefonu, souhrnně se tato funkce označuje Cisco XML service.

Téma této práce jsem si vybral pro možnost získání rozsáhlém množství vědomostí, jenž jsou obsaženy v tématu služeb pro IP telefony za použití softwarové pobočkové ústředny Asterisk.

## 2 Úvod do dílčích prvků problematiky

### 2.1 Co je to Asterisk

*Asterisk je open source framework pro vytváření komunikačních aplikací. Asterisk vlastně mění obyčejné Pc na ústřednu pro hovorovou i mimo hovorovou komunikaci. Asterisk má schopnost provozovat IP PBX systems, VoIP gateways, systémy pro konferenční hovory a mnoho dalších. Díky možnosti „nabalovat“ na Asterisk další balíčky se Asterisk stal široce využívaným systéme a to jak malými firmami a nadnárodními společnostmi. [1]*

#### 2.1.1 Použití ústředny Asterisk v projektu

V mé práci jsem ústřednu Asterisk použil jako telefonní ústřednu pro IP telefony CISCO řady 7940 a 7960. Aby mohl být telefon použit k uskutečňování hovorů musí být registrován u ústředny, ústředna Asterisk byla k tomuto vybrána, protože se jedná o open source software a má rozsáhlou komunitu.

#### 2.1.2 Instalace

##### 2.1.2.1 Nutné vybavení

Minimum pro provoz software ústředny Asterisk je virtuální server, ale pro velké a nebo komerční využitím je doporučen fyzický server s dostatečným výkonem. Dalším předpokladem je znalost práce s Linuxem. Práci jsem prováděl na virtuálním serveru siptest2.feld.cvut.cz. Virtuální server byl vybaven Linuxovou distribucí Debian.

##### 2.1.2.2 Způsoby instalace

Asterisk lze instalovat třemi různými způsoby. První je instalace ústředny Asterisk spolu s Linuxovou distribucí, která je pro něj přímo upravená společností Digium s.r.o., která Asterisk zaštiťuje a dále vyvíjí. Druhým způsobem je instalace z předkompilovaných balíčků na již existující Linux, verze ústředny Asterisk instalovaného z předkompilovaných balíčků závisí na Vaši Linuxové distribuci a její samotné verzi a proto se mohou balíčky lišit mezi distribucemi. A třetím způsobem, který jsem použil v projektu, je kompilace ze zdrojových kódů.

Výhody kompilace ze zdrojových kódů oproti ostatním způsobům

1. Největší kontrola nad instalací
2. Oproti prvnímu způsobu, kde se instaluje i OS, zde můžeme využít Linuxu který již máme
3. Oproti druhému způsobu nejsme vázáni verzi ústředny Asterisk, kterou má naše linuxová distribuce v repositářích



### 2.1.2.3 Průběh instalace

Před kompilací samotného ústředny Asterisk, použitá verze 11.13, jsem provedl stažení zdrojových kódů knihovny DAHDI (dříve nazývané Zapstel) verze 2.10.0.1.

DAHDI knihovna umožňuje komunikaci s analogovými, digitálními telefony a telefonními linkami. A je nutné provést její instalaci ještě před instalací samotné ústředny Asterisk.

Pro provádění kompilace a instalace je nutné mít nainstalovaný program *make*, pokud Vaše distribuce jím není standardně vybavena.

Kompilace DAHDI jsem provedl jsem provedl těmito příkazy:

```
root@siptest2:~/dahdi# make
root@siptest2:~/dahdi# make install
root@siptest2:~/dahdi# make config
```

Poté následovala kontrola nutných knihoven pro kompilaci a běh ústředny Asterisk příkazem:

```
root@siptest2:~/asterisk# ./configure
```

Zde je seznam vyžadovaných knihoven pro mnou použitou verzi ústředny Asterisk:

- openssl (požadováno pro modul chan\_sip)
- ncurses
- libxml2
- DAHDI
- unixodbc
- libspeex
- libspeexdsp
- libresample
- libcurl3
- libvorbis
- libogg
- libsrtp
- libical
- libiksemel
- libneon
- libgmime
- uuid
- libsqlite3

Zde je výpis z příkazové řádky oznamující, že všechny požadavky byly splněny.

```
configure: Package configured for: &nbsp;
configure: OS type &nbsp;: linux-gnu
configure: Host CPU : x86_64
configure: build-cpu:vendor:os: x86_64 : unknown : linux-gnu :
configure: host-cpu:vendor:os: x86_64 : unknown : linux-gnu :
```

Po tomto jsem provedl kompilaci ústředny Asterisk, zde jsou použité příkazy a části výpisů z příkazové řádky.

Příkaz:

```
root@siptest2:~/asterisk# make
```

Výstup:

```
+----- Asterisk Build Complete -----+
+ Asterisk has successfully been built, and +
+ can be installed by running:           +
```

```

+
+
+
+----- Asterisk Build Complete -----+

```

Podle výstupu z konzole vidíme, že zdrojové soubory ústředny Asterisk se úspěšně podařilo zkompileovat pro naši architekturu. Jak výpis naznačuje nyní provedu instalaci zkompileovaných souborů níže uvedeným příkazem.

Příkaz:

```
root@siptest2:~/asterisk# make install
```

Výstup:

```

+----- Asterisk Installation Complete -----+
+
+   YOU MUST READ THE SECURITY DOCUMENT   +
+
+ Asterisk has successfully been installed. +
+ If you would like to install the sample  +
+ configuration files (overwriting any     +
+ existing config files), run:            +
+
+
+
+
+
+
+
+----- Asterisk Installation Complete -----+

```

Jelikož se jednalo o novou instalaci provedl jsem i příkaz `root@siptest2:~/asterisk# make samples`, který vytvořil základní konfigurační soubory. Nakonec následoval `root@siptest2:~/asterisk# make config`, který slouží k vytvoření init skriptů.

Tím to kompilace a instalace ústředny Asterisk úspěšně skončila.

### 2.1.3 Moduly `chan_skinny` a `chan_sccp_b`

`Chan_skinny` je modul ústředny Asterisk použitý pro správu skinny protokolu, sloužící pro spojení ústředny Asterisk s Cisco telefony série 7900. `chan_skinny` je výchozí součástí instalace ústředny Asterisk.

`Chan_sccp_b` je modul ústředny Asterisk sloužící ke stejnému účelu jako `chan_skinny`.

Rozdíly:

- *chan\_sccp\_b taktěž podporuje Cisco telefony série 7900 a IP telefony s nimi kompatibilní dále Cisco IP telefon model 12 a 30, Nokia call connect, Cisco ATA a KIRK (model bezdrátového telefonu používající sccp)*
- *oba moduly nabízí lepší kvalitu a rychlost přenosu dat než SIP*
- *chan\_sccp\_b nabízí více funkcí než chan\_skinny, hlavně plně funkční XML podporu [2]*

Pro komunikaci mezi ústřednou Asterisk a klientem (v mém případě IP telefony od CISCO) může být použito tři modulů, a to `chan_sip` `chan_skinny` a `chan_sccp_b`, pro projekt byl vybrán díky

svým vlastnostem modul `chan_sccp_b`. Jelikož `chan_sccp_b` není výchozí součástí ústředny Asterisk byl stažen a kompilován dodatečně. Použitá verze modulu je 4.2.0\_RC2. Kompilace modulu se provedla stejnými příkazy jako u DAHDI.

## 2.1.4 Konfigurace ústředny Asterisk a registrace telefonu k ústředně

Konfigurace probíhala následovně:

1. Zakázalo se načítání modulu `chan_skinny` (který se načítá jako výchozí modul pro sccp protokol) a místo něj se povolilo načítání `chan_sccp_b`.

Soubor `/etc/asterisk/modules.conf`

```
no_load => chan_skinny.so
load => chan_sccp.so
```

2. Zaregistroval se telefon do ústředny Asterisk a přiřadili se k němu telefonní linky.

Soubor `/etc/asterisk/sccp.conf`

```
[SEP203706174FD3]
description = Phone Number One
addon = 7914
devicetype = 7962
park = off
button = line, 98011, default
cfwdall = off
type = device
keepalive = 60
transfer = on
park = on
cfwdall = off
cfwdbusy = off
cfwdnoanswer = off
directed_pickup = on
directed_pickup_context = default
directed_pickup_modeanswer = on
deny=0.0.0.0/0.0.0.0
permit=192.168.22.139/255.255.255.255
dndFeature = on
directrtp=off
earlyrtp = progress
private = on
mwilamp = on
mwioncall = off
setvar=testvar=value
cfwdall = on

[98011]
id = 1000
type = line
pin = 1234
label = Phone 1 Line 1
description = Line 98011
mailbox = 10011
cid_name = MY CID
cid_num = 98011
accountcode=79011
```

```
callgroup=1,3-4
pickupgroup=1,3-5
context = default
incominglimit = 2
transfer = on
vmnum = 600
meetme = on
meetmeopts = qxd
meetmenu = 700
trnsfvm = 1000
secondary_dialtone_digits = 9
secondary_dialtone_tone = 0x22
musicclass=default
language=en
echocancel = on
silencesuppression = off
setvar=testvar2=my value
dnd = reject

parkinglot = myparkspace
```

## 2.2 SIP

*Protokol SIP vznikl pod křídly organizace IETF (Internet Engineering Task Force). Jde o textový protokol strukturou podobný například poštovnímu protokolu SMTP (Simple Mail Transfer Protocol) nebo protokolu HTTP (HyperText Transport Protocol). Tělo zprávy je tvořeno textovými položkami ve formě <název>:<hodnota>, které popisují předávané informace. Textová podstata protokolu napomáhá nejen jednoduchému ladění, ale především snadné rozšiřitelnosti.[3]*

*Protokol SIP je typu klient-server, takže komunikace probíhá výměnou dvou typů zpráv, požadavků a odpovědí. Klient i server jsou logickou částí jednoho prvku. Pokud je zmiňován termín klient, je tím míněn telefon nebo softwarová aplikace u uživatele a pojmem server jsou označovány aplikační servery služby, které poskytují klientům další služby jako registrace, lokalizace, zpoplatňování atd. Protokol SIP lze použít i pro další účely například síťové hry, instant messaging a další.[3]*

## 2.3 IAX

*Protokol byl vytvořen v rámci projektu Asterisk PBX jako vylepšení tehdy dostupných VoIP protokolů. Současná verze protokolu IAX2 se od ostatních VoIP protokolů liší především tím, že signalační a media data jsou přenášena jedním datovým tokem na jednotném portu v binárním formátu. IAX2 používá jediný UDP data stream (na portu 4569), ve které multiplexuje signalizaci a media data, čímž mnohem snadněji prochází přes firewall a překlady adres NAT. Při trunkovém provozu zároveň tímto způsobem výrazně šetří přenosové pásmo, protože minimalizuje IP overhead paketových hlaviček normálně potřebný k přenosu informací o probíhajících hovorech.*

*Přestože zmíněná kompaktnost protokolu je jeho velkou výhodou, postrádá obecný mechanismus pro další rozšiřování funkcionalit. To znamená, že každé vylepšení vyžaduje změnu specifikace protokolu a jeho nekompatibilitu s předchozí verzí.[4]*

## 2.4 TFTP

*Jednoduchý protokol pro přenos souborů postavený na FTP. TFTP je určen pro přenos souborů v případech, kdy je běžný protokol FTP nevhodný pro svou komplikovanost. Typickým případem je bootování bezdiskových počítačů ze sítě, kdy se celý přenosový protokol musí vejít do omezeného množství paměti, která je k dispozici na bezdiskovém stroji.[5]*

*Oproti FTP má různá omezení a odlišnosti:*

- *Nelze procházet adresáře.*
- *Neumožňuje přihlášení uživatele ani zadání hesla.*
- *Je používán pro čtení nebo zápis dat na vzdálený server.*
- *Podporuje tři odlišné přenosové módy: netascii (pro text v ASCII s úpravami z protokolu Telnet), octet (pro syrová binární 8bitová data) a mail (pro zaslání e-mailové zprávy; tento mód by se už neměl používat).*
- *Maximální velikost přenášeného souboru je 32 MB.[5]*

### 2.4.1 Využití TFTP v projektu

TFTP server jsem v projektu využil k přenosu konfiguračního souboru z serveru do telefonu. Tento soubor je nutný k nastavení telefonu a k předání adres služeb.

Cisco telefony se chovají, tak že po připojení telefonu do sítě, poté co dostane IP adresu od DHCP nebo je mu přidělena staticky prostřednictvím nastavení telefonu, hledá telefon TFTP server na adrese, která se zadá v telefonu přes nastavení telefonu, a to tak že se povolí alternativní TFTP server a vloží se IP adresa TFTP serveru.

Po připojení k TFTP telefon očekává XML soubor s nastavením, jméno tohoto souboru musí být ve tvaru SEP<MAC>.cnf.xml, kde za <MAC> musí být MAC adresa telefonu, kterému konfigurační soubor patří.

Obsah SEP203706174FD3.cnf.xml našeho telefonu najdete v kapitole přílohy část SEP203706174FD3.cnf.xml.

Struktura XML souboru použitého ke konfiguraci musí dodržovat tyto náležitosti.

Počátečním tagem označující začátek a tagem označující konec jsou v pořadí <device> a </device>.

Za počátečním tagem následuje párový tag <versionStamp> nesoucí informaci o verzi, v podobě čísla modelu telefonu, datu a času. Datum a čas jsou ve formátu M dd yyyy hh:mm:ss. Poté následuje nastavení určující ke které telefonní ústředně se má telefon hlásit párovým tagem <processNodeName> a skrze jaký port uvedeným v <ports> a do jakého časového pásma patří v <timeZone>, kromě časového pásma se uvádí i formát pro zobrazení data a Olson označení časového pásma, všechna tato nastavení se nachází uvnitř párového tagu <devicePool>.

Poté mohou následovat informace že se mají do telefonu načíst další soubory. Názvy těchto souborů a jejich verze je pak zapsána v <loadInformation>, pokud je tento párový tag ponechán prázdný telefon je informován, že další soubory se načítat nebudou.

Obsah souboru pokračuje několika tagy určující URL adresy, na kterých se nachází různé služby. Nejpodstatnější z nich pro mou práci je párový tag <servicesURL>, kde se nachází adresa k webové službě, na kterou se má telefon připojit když uživatel zmáčkne tlačítko services na telefonu.

Za adresou služeb následuje párový tag <userLocale> uvozující místní nastavení telefonu, jako je kód jazyka. Poté následují informace o lokální síti.

Nakonec následují ostatní nastavení telefonu, které se nachází uvnitř tagů <vendorConfig> a </vendorConfig>, tyto nastavení jsem ponechal ve výchozích hodnotách.

Toto celé nastavení je, jak jsem již zmínil, uzavřeno tagem </device>.

## 2.4.2 Instalace TFTP serveru a klienta

Na virtuální server jsem si na instaloval TFTP z balíčků dodávaných linuxovou distribucí, přes příkaz: `sudo apt-get install xinetd tftpd tftp`, čímž se provedla veškerá instalace.

### Konfigurace TFTP serveru

Konfigurace se provádí v souboru `/etc/xinetd.d/tftp`, který je po instalaci nutno vytvořit.

Obsah souboru viz kapitola přílohy část tftp.

Vytvoření složky, odkud se odesílají soubory uložené v TFTP serveru:

```
sudo mkdir /tftpboot
sudo chmod -R 777 /tftpboot
sudo chown -R tftp /tftpboot
```

Po tomto jsem restartoval službu xinetd.

Testování fungování tftp serveru jsem vyzkoušel souboru s „Hello world!“

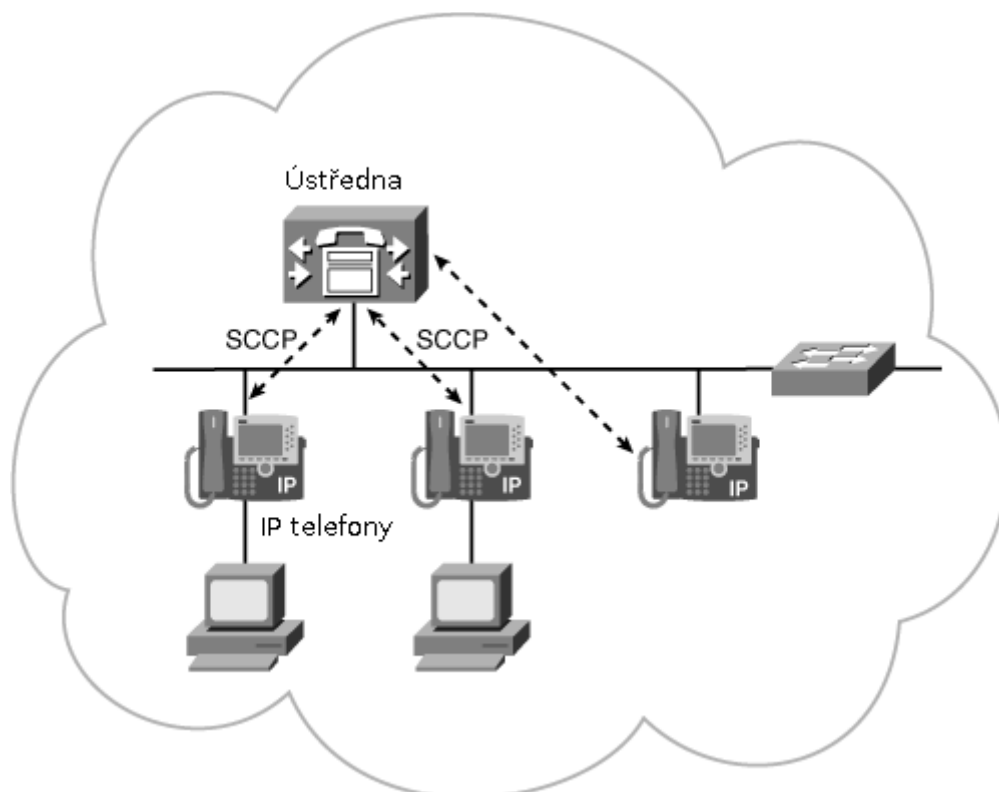
```
echo "Hello world!" > /tftpboot/test.txt
tftp
tftp> connect siptest2.feld.cvut.cz
tftp> get test.txt
Sent 159 bytes in 0.0 seconds
tftp> quit
cat text.txt
Hello world!
```

Soubor se odeslal a přijal v pořádku.

## 2.5 SCCP

*Skinny Client Control Protocol (SCCP) je proprietární síťový protokol původně vytvořený firmou Cisco. SCCP nyní vlastní a spravuje Cisco Systems. SCCP je nízko zátěžový protokol, který slouží pro komunikaci mezi IP telefony a ústřednou. Ústředna společnosti Cisco se nazývá Cisco Call Managerem (CCM). Jedním z příkladů SCCP klientů jsou Cisco IP telefony řady 79xx a další.* [6]

Tento protokol je použit v tomto projektu pro práci s IP telefony Cisco řady 79XX a to kanálovým ovladačem chan\_SCCP-b oproti výchozímu chan\_skinny jenž přichází s ústřednou Asterisk.



Orb. 1 prvky a zapojení protokolu SCCP

Důvod pro použití SCCP-b je jeho lepší výkonnost oproti Skinny a nabízí mnohé funkce, z nichž jsem použil například nastavení volby rychlého vytáčení, vedení kontaktů, telefonů a linek v externí databázi a jejich úprava v reálném čase.

## 2.6 Číslovací plán (dial plan)

### 2.6.1 Číslovací plán v ČR

*Číslovací plán stanovuje číslování veřejných pevných a mobilních telefonních sítí, včetně tzv. sítí „ISDN“, určuje očíslování telefonních obvodů (dále jen TO), resp. uzlových telefonních obvodů v přechodném období, doplňkových služeb, přídatných služeb a číslování přístupů k dalším veřejným telekomunikačním službám, resp. sítím, pokud mezi nimi existují přechody umožňující příslušné přístupy. Stanovuje přestupný znak do mezinárodní telefonní sítě a dále i do meziměstské sítě v přechodném období. Číslovací plán určuje obecná pravidla tvorby národního (významového) čísla účastníků veřejných telefonních sítí a definuje jeho strukturu.[7]*

Číslovací plán je vydáván a usměrňován Českým telekomunikačním úřadem.

### 2.6.2 Číslovací plán v projektu

Číslovací plán v ústředně Asterisk se skládá z kolekce kontextů, kontext zastupuje skupinu uživatelů a rozšíření které k nim patří.

Zde je ukázka reprezentace číslovacího plánu v ústředně Asterisk:

Context „felDejvice“:

Extension Description

100 vratnice

101 bufet

104 hlasova zprava

V tomto kontextu pojmenovaném felDejvice první dva řádky (rozšíření) jsou asociována s telefony a třetí řádek(104) je asociován pro s hlasovou zprávou.

Ukázka kontextu felDejvice v kódu:

```
[felDejvice]
```

```
exten => 100,1,Dial(SIP/vratnice,30)
```

```
exten => 101,1,Dial(SIP/bufet,30)
```

```
exten => 104,1,Voicemail(10044)
```

Vysvětlení zápisu, když někdo vytočí 100 tak Asterisk spojí volajícího s vrátnicí přes SIP, 30 za jménem určí čas ve vteřinách po jakou dobu telefon zvoní. Při vytočení 104 může volající zanechat hlasovou zprávu, číslo 10044 je číslo mailboxu, kde bude zpráva uložena.

#### 2.6.2.1 Použití kontextů

Když Asterisk přijme telefonní spojení, ať už přichází hovor z venku nebo vnitřní hovor z jeho kontextů, spojení vždy patří do nějakého kontextu. První věc ke které se kontexty využijí je aby Asterisk prováděl různé činnosti na základě zda jde o vnitřní nebo vnější hovor. Například vnitřní hovory, z ukázky ty z felDejvice, mohou vytáčet čísla ve svém kontextu a vnější bychom mohly například vždy položit(HangUp).

Pro potřeby projektu byly postačující dva účty, každý se svým tel. číslem. Účty byly vytvořeny pro tzn. SIP klienty. Účelem těchto účtu bylo vyzkoušet spojení mezi nimi a zda je správně nakonfigurován Asterisk.



Nastavení účtů se provedlo v souboru `/etc/asterisk/sip.conf`, obsah souboru viz kapitola Přílohy část `sip.conf`.

Nastavení tel. čísel a spojení se provedlo v souboru `/etc/asterisk/extensions.conf`, obsah souboru viz kapitola Přílohy část `extensions.conf`.

Před vyzkoušením jsem provedl restart ústředny Asterisk.

## 3 Systém integrace ústředny Asterisk a webových služeb

### 3.1 Integrace databáze s ústřednou Asterisk

#### 3.1.1 Přidání ústředny Asterisk jako uživatele databáze

Jedním z prvních a nejdůležitějších požadavků je komunikace ústředny Asterisk s databází v našem případě jde o externí databázi kontaktů. To v první řadě vyžaduje vytvoření uživatele databáze s jménem „asterisk“.

```
mysql> CREATE USER 'asterisk'@'%' IDENTIFIED BY 'some_secret_password';
```

Příklad vytvoření uživatele „asterisk“ pro databázi MySQL.

A přidání práv danému uživateli.

```
mysql> GRANT ALL PRIVILEGES ON databaseName.* TO 'asterisk'@'%';
```

Příklad přidělení všech práv k databázi uživateli „asterisk“.

#### 3.1.2 Instalace a konfigurace ODBC

##### 3.1.2.1 Instalace ODBC podpory v operačním systému

Dalším požadavkem je přidání podpory ODBC do ústředny Asterisk, která není součástí výchozí konfigurace.

ODBC konektor je abstraktní vrstvou, která umožňuje komunikaci mezi ústřednou Asterisk a širokým rozpětím databází bez nutnosti vytvářet konektivitu mezi každou jednotlivou databází s kterou je Asterisk potřeba propojit. Což je i důvodem proč byl ODBC konektor použit v první řadě.

Před samotným přidáním podpory ODBC do konfigurace ústředny Asterisk je nutné se ujistit, že systém, na kterém Asterisk „běží“, obsahuje ovladače pro jeho podporu.

```
$ sudo apt-get install unixODBC unixODBC-dev
```

*Příklad instalace nutné podpory ODBC na systémech Debian/Ubuntu.*

```
$ sudo apt-get install libmyodbc
```

*Příklad instalace ODBC – MySQL konektoru na systémech Debian/Ubuntu.*

Dále je třeba ODBC konektor na konfigurovat, konfigurace je ukázána na konfiguraci ODBC – MySQL viz kapitola Konfigurace ODBC pro MySQL.

Tím to jsou završeny předpoklady ze strany databáze a systému, teď je třeba provést kompilaci, popřípadě rekompilaci ústředny Asterisk pro zahrnutí modulů pro práci s ODBC.

### 3.1.2.2 Instalace podpory ODBC v ústředně Asterisk

Námi potřebné moduly jsou následující: `dr_odbc`, `cdr_adaptive_odbc`, `func_odbc`, `func_realtime`, `pbx_realtime`, `res_config_odbc` a `res_odbc`. Tyto moduly obsahují ovladače, funkce a další záležitosti pro práci ústředny Asterisk s databázemi přes ODBC a funkce databáze využívající.

K provedení rekompilace a vybrání výše zmíněných modulů slouží příkazy:

```
$ ./configure  
$ make menuselect  
$ make install
```

Po začlenění těchto modulů a provedení kompilace, je třeba provést konfiguraci modulu „`res_odbc`“. Provedení této konfigurace naleznete viz kapitola Konfigurace modulu „`res_odbc`“. Tato konfigurace je posledním předpokladem před začleněním databáze kontaktů do číslovacího plánu(dial plan) ústředny Asterisk.

### 3.1.2.3 Konfigurace ODBC pro MySQL

Jedná se o nastavení ovladačů ODBC – MySQL v systému, ta se provádí editací souboru `/etc/odbcinst.ini`.

```
[MySQL]  
Description = ODBC for MySQL  
Driver = /usr/lib/odbc/libmyodbc.so  
Setup = /usr/lib/odbc/libodbcmyS.so  
FileUsage = 1
```

*Příklad nastavení ODBC pro MySQL na 32bit systému.*

```
[MySQL]  
Description = ODBC for MySQL  
Driver = /usr/lib64/odbc/libmyodbc.so  
Setup = /usr/lib64/odbc/libodbcmyS.so  
FileUsage = 1
```

*Příklad nastavení ODBC pro MySQL na 64bit systému.*

Dalším krokem byla editace souboru `/etc/odbc.ini`, který se používá pro konfiguraci identifikace, kterou bude Asterisk používat.

```
[asterisk-connector]  
Description = MySQL connection to 'asterisk' database  
Driver = MySQL  
Database = databaseName  
Server = localhost
```

*UserName* = *asterisk*  
*Password* = *password*  
*Port* = *3306*  
*Socket* = */var/lib/mysql/mysql.sock*

*Příklad nastavení identifikace použité mezi ústřednou Asterisk a ODBC.  
Cesta k soketu se může lišit dle typu a verze systému.*

### 3.1.2.4 Konfigurace modulu „res\_odbc“

Tato konfigurace slouží k zavedení parametrů pro jednotlivá připojení ústředny Asterisk přes ODBC tyto nastavené parametry jsou pak využívány moduly ústředny Asterisk pro spojení s databázemi.

Nastavení se provádí editací souboru */etc/asterisk/res\_odbc.conf*.

```
[asterisk]  
enabled => yes  
dsn => asterisk-connector  
username => asterisk  
password => welcome  
pooling => no  
limit => 1  
pre-connect => yes
```

*Příklad nastavení modulu „res\_odbc“.*

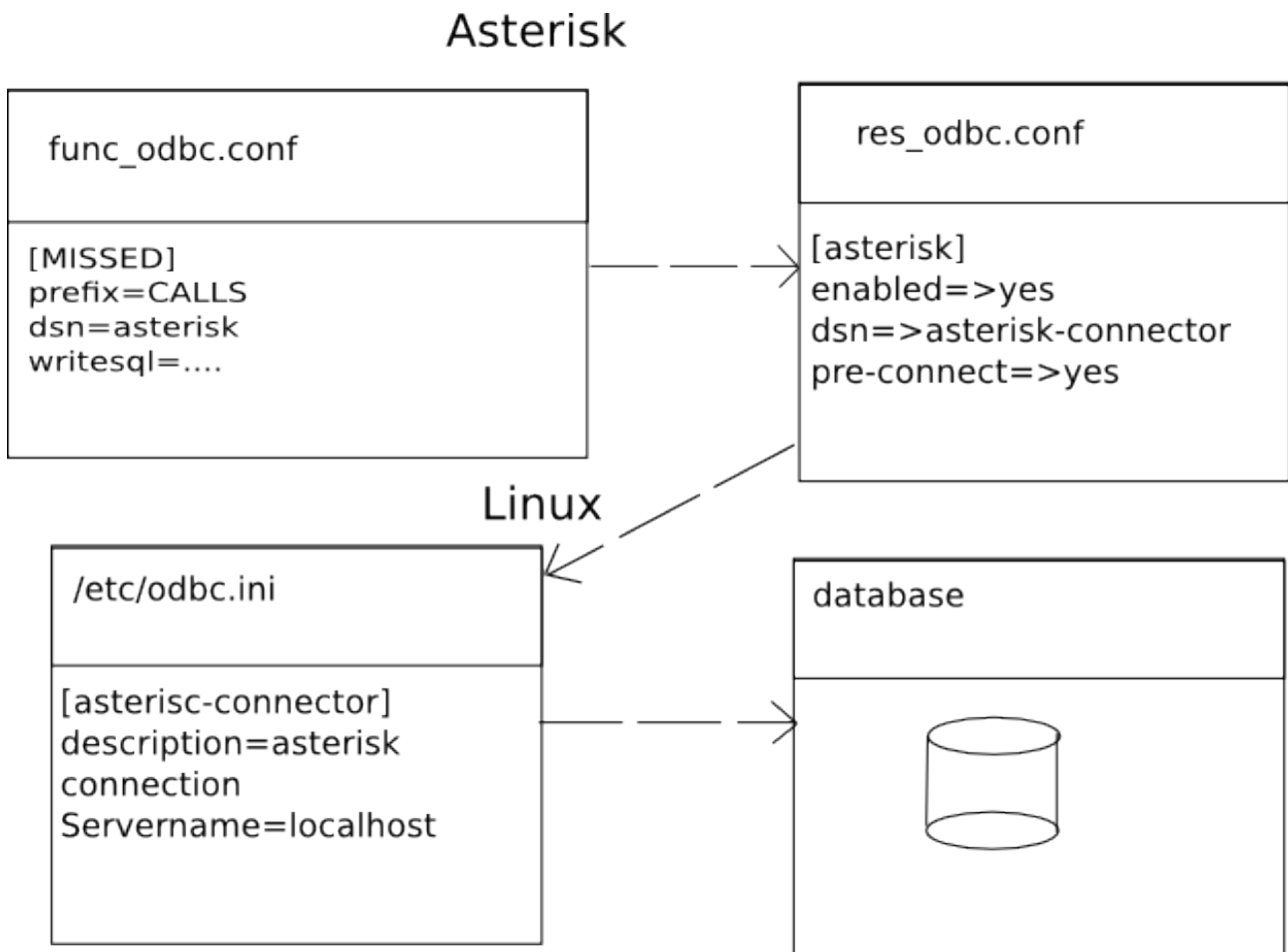
Parametr *dsn* z příkladu slouží jako odkaz na nastavení identifikace, kterou jsme provedly při konfiguraci ODBC. Parametry *pooling* a *limit* slouží k povolení více připojení do databáze po určený limit, jenž je užitečné pro databáze MS SQL a Sybase.

## 3.2 Systém registru hovorů generovaných ústřednou Asterisk

### 3.2.1 Generování registru hovorů

Při tvorbě generování registru hovorů jsem navázal na připojení databáze přes ODBC s ústřednou Asterisk, zmíněno v předchozích kapitolách.

K generování využívám modulu ústředny Asterisk s názvem *func\_odbc*, jde o modul pro definování vlastních funkcí jenž jsou následně využity v číslovacím plánu (dial plan) ústředny.



Obr. 2 Schéma návazností a konektivity mezi ODBC a ústřednou Asterisk

### 3.2.1.1 Funkce v func\_odbc.conf

V modulu *func\_odbc* jsem si vytvořil funkce pro zápis a získávání dat mezi databází a číslovacím plánem, a to funkcí pro vedení zmeškaných, příchozích i odchozích hovorů a získávání dat o účastnících hovorů jako například uživatelské jméno.

Funkce pro zápis o hovorech

```
[MISSED]
prefix=CALLS
dsn=asterix
writesql=INSERT INTO missedcalls (callername, timeofcall, callednumber, status) VALUES ( '${ARG1}', '${ARG2}', '${ARG3}', '${ARG4}')
```

Tato funkce slouží pro generování záznamu o zmeškaném hovoru do databáze, funkce pro přijaté a odchozí hovory jsou podobné až na rozdílné tabulky.

Funkce v modulu *func\_odbc* jsou definovány několika volbami na příkladu viz výše jsou tři z nich a to *prefix*, *dsn* a *writesql*. Volba *prefix* je volitelná slouží k přidání specifického prefixu při volání dané funkce, bez uvedení je použit výchozí prefix, kterým je ODBC. Volba *dsn* slouží k určení DSN, definované v *res\_odbc*, která bude použita k provedení příkazu, DSN je možné specifikovat až pět oddělených čárkou, kde první slouží jako primární a ostatní slouží jako fallback DSN v pořadí v kterém jsou zapsány. DSN je možné rozlišit na zápisné a čtecí specifikování daných DSN pomocí voleb *readhandle* a *writehandle*, při tomto rozlišení se volba *dsn* nepoužívá.

Funkce pro získání dat z databáze

```
[USERDATA]
prefix=GET
dsn=asterix
readsq=SELECT username FROM assignedphones WHERE extension = '${E}'
```

Tato funkce mi slouží k získání uživatelského jméno patřícímu k danému telefonnímu číslu.

Z modulu *func\_odbc* je tady použita volba *readsq*, která slouží pro získání dat z databáze a předává je jako návratovou hodnotu funkce.

### 3.2.1.2 Využití funkcí v číslovacím plánu

Tyto funkce, zmíněné výše, jsou využity v číslovacím plánu pro generování záznamů o hovorech. Uvedu zde jak se vytvoří záznam o zmeškaném hovoru.

```
[incomming]
exten=> _XXXXXXXXXX,1,NoOp()
same=>n,Set(E=${EXTEN})
same=>n,Set(USER=${CALLERID(name)})
same=>n,Dial(SCCP/${E},120)
same=>n,Goto(s-${DIALSTATUS},1)
```

```
exten=>s-BUSY,1,BackGround(volany-je-zaneprazdnen)
same=>n,NoOp(CALLS_MISSED(${USER},${STRFTIME(${EPOCH},,%Y%m%d-%H%M
%S)}},${E},'BUSY'))
same=>n,Hangup()
```

```
exten=>s-CANCEL,1,BackGround(zruseno)
same=>n,NoOp(CALLS_MISSED(${USER},${STRFTIME(${EPOCH},,%Y%m%d-%H%M
%S)}},${E},'CANCEL'))
same=>n,Hangup()
```

```
exten=>s-CHANUNAVAIL,1,BackGround(channel-je-nedostupny)
same=>n,NoOp(CALLS_MISSED(${USER},${STRFTIME(${EPOCH},,%Y%m%d-%H%M
%S)}},${E},'CHANNELUNAVAIL'))
same=>n,Hangup()
```

```
exten=>s-NOANSWER,1,BackGround(volany-neodpovida)
same=>n,NoOp(CALLS_MISSED(${USER},${STRFTIME(${EPOCH},,%Y%m%d-%H%M
%S)}},${E},'NOANSWER'))
same=>n,Hangup()
```

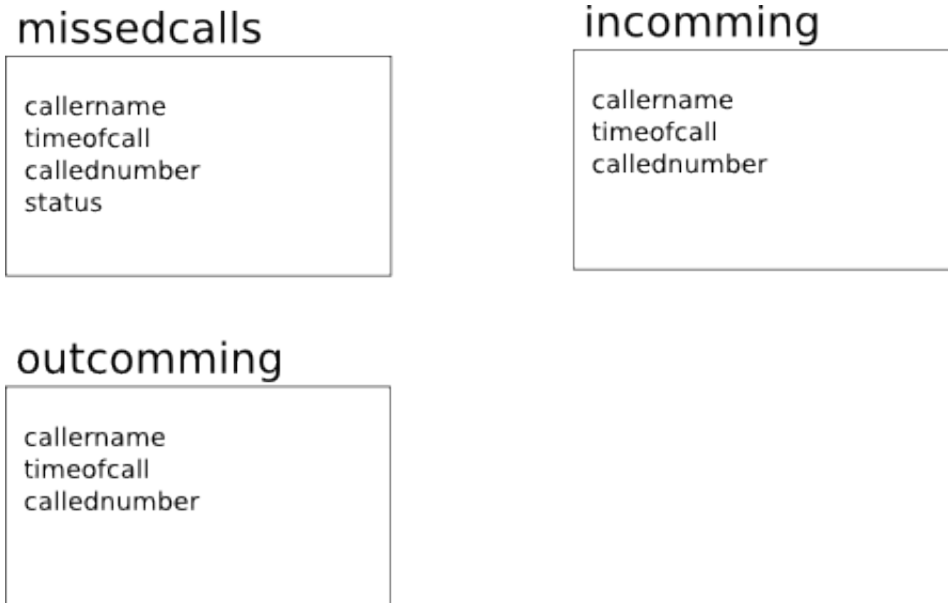
Jak jsem se zmínil výše je výtazek z číslovacího plánu, který se zabývá případem kdy hovor neproběhne úspěšně, v tom případě je hovor odskočí podle důvodu z kterého neproběhl na příslušnou pozici a funkci CALLS\_MISSED je generován záznam do databáze zmeškaných hovorů. V záznamu je uveden volající, datum a čas hovoru, telefonní číslo, které bylo vytáčeno, a důvod neproběhnutí.

Pro příchozí hovory, které proběhly úspěšně, hovor odskočí do těchto pozic:

```
exten=>s-ANSWER,1,NoOp(CALLS_INCOMING(${USER},${STRFTIME(${EPOCH},,%Y
%m%d-%H%M%S)}},${E}))
same=>n,Hangup()
```

v této části číslovacího plánu je vygenerován záznam o úspěšném příchozím hovoru s uvedením volajícího, datum a čas hovoru a telefonní číslo, které bylo vytáčeno.

### 3.2.2 Struktura databáze registru hovorů



Obr. 3 Struktura tabulek pro registr hovorů



## 3.3 Svázání databáze kontaktů s číslovacím plánem ústředny Asterisk

Výchozím způsobem vložení kontaktu pro SCCP-b je jejich ruční vepsání do konfiguračního souboru `sccp.conf`, to z praktických důvodů nebývá vhodné, například častá změna nebo nadměrné množství kontaktů. Proto se používá databáze kontaktů jimiž jsou záznamy v konfiguračních souborech nahrazeny.

### 3.3.1 nastavení napojení databáze kontaktů s číslovacím plánem ústředny Asterisk

V předchozích kapitolách jsem vytvořil systém pro registraci záznamů o hovorech generované ústřednou Asterisk, k tomuto úkolu jsem použil externí databázi, s kterou jsem Asterisk spojil přes ODBC.

Stejným napojením lze propojit již existující databázi jenž obsahuje kontakty telefonů Cisco či jiných. Má práce je zaměřena právě na telefony Cisco a to přes SCCP-b.

V napojení externí databáze vycházím z již s připraveným spojením ODBC-databáze a spojením ODBC-Asterisk uvedeným v kapitolách výše.

Dalším krokem bylo určení ústředny Asterisk odkud má bráti nastavení telefonů a jim přidělených linek. Tento krok jsem provedl konfigurační souboru `/etc/asterisk/extconfig.conf`.

```
[settings]
```

```
sccpdevice => odbc,asterisk,sccpdeviceconfig
```

```
scpline => odbc,asterisk,scplines
```

Zde je uvedena samotná konfigurace v klíči `sccpdevice` je učena cesta k tabulce obsahující nastavení telefonů, z konfigurace výše je vidět, že způsob spojení je skrze ODBC, následující dva parametry určují název databáze a název tabulky v tomto pořadí. Druhý klíč `scpline` určuje cestu k tabulce obsahující nastavení telefonních linek.

Po nastavení cest k telefonům a linkám jsem dále provedl nastavení kanálu SCCP-b a to editací souboru `/etc/asterisk/sccp.conf`.

```
[general]
```

```
devicetable=sccpdevice
```

```
linetable=scpline
```

Toto nastavení slouží pro SCCP aby znal, kde najde cestu k telefonům a linkám jenž je uvedena v aliasech `sccpdevice` a `scpline` jenž jsou uvedeny v `/etc/asterisk/extconfig.conf`.

### 3.3.2 Struktura databáze telefonů a linek

#### sccpdevice

```
type
addon
description
tzooffset
transfer
cfwdall
cfwdbusy
imageversion
deny
permit
dndFeature
directrtp,
earlyrtp
mwilamp
mwioncall
pickupexten
pickupcontext
pickupmodeanswer
private
privacy
nat
softkeyset
audio_tos
audio_cos
video_tos
video_cos
conf_allow
conf_play_general_announce
conf_play_part_announce
conf_mute_on_entry
conf_music_on_hold_class
conf_show_conflist
setvar
disallow
allow
backgroundImage
ringtone
```

#### sccplines

```
id
pin
label
description
context
incominglimit
transfer
mailbox
vmnum
cid_name
cid_num
trnsfvm
secondary_dialtone_digits
secondary_dialtone_tone
musicclass
language
accountcode
echocancel
silencesuppression
callgroup
pickupgroup
amaflags
dnd
setvar
name
```

#### buttonconfig

```
device
instance
type
name
options
```

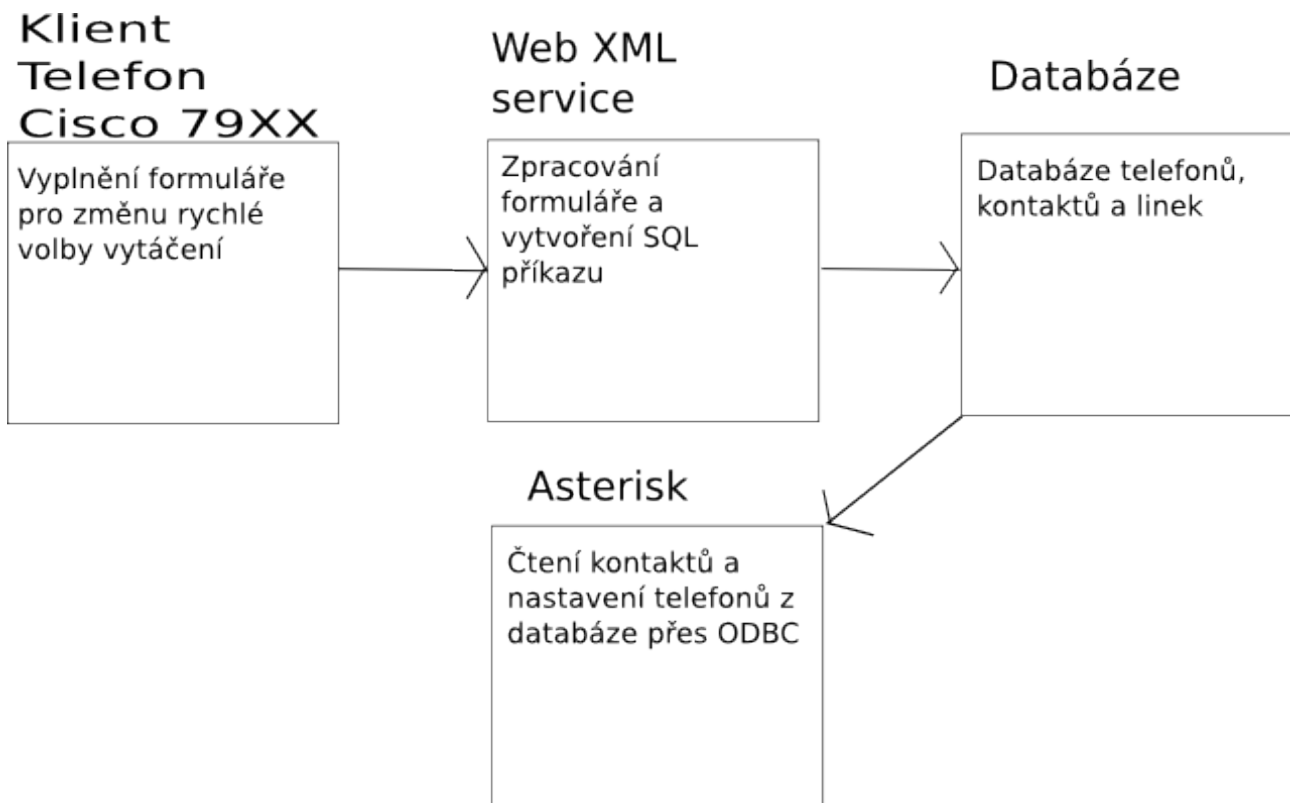
Obr. 4 Struktura tabulek telefonů a linek pro sccp-b

Výsledné nastavení telefonu používané ústřednou Asterisk, nazvané sccpdeviceconfig, je vytvořené jako SQL View nad sjednocením tabulek sccpdevice a buttonconfig.

### 3.4 Modifikace číslovacího plánu uživatelem

Proto aby uživatel mohl aktivně provádět změnu, bylo nutné přesunout konfiguraci telefonů a jim přidělených linek z konfiguračních souborů do realtime databáze. Tím navazuji na integraci databáze z předchozích kapitol a uceluji všechny jednotlivé součásti mé práce do jednoho systému.

Jako modifikaci číslovacího plánu uživatelem jsem vybral editace rychlé volby vytáčení z důvodu její užitečnosti. Dalšími využitími jsou například přidávání nových telefonů a linek v reálném čase a další.



Obr. 5 Schéma průběhu změny volby rychlého vytáčení v systému

Modifikace rychlého vytáčení začíná připojením uživatele k web/XML službě na klientském zařízení, načež je mu zobrazen formulář modifikace volby rychlého vytáčení, zadáním jména a příjmení uživatele jenž si chce dát do rychlého vytvočení. Tento formulář je odeslán do webové části služby. Ta jej zpracuje s využitím dotazů do databáze kontaktů, tak je získáno číslo linky k telefonu, jenž bude dán jako volba rychlého vytáčení. Poté služba zhotoví SQL příkaz do databáze o změně nastavení tlačítka rychlého vytáčení. Toto nastavení je čteno z databáze ústřednou Asterisk.

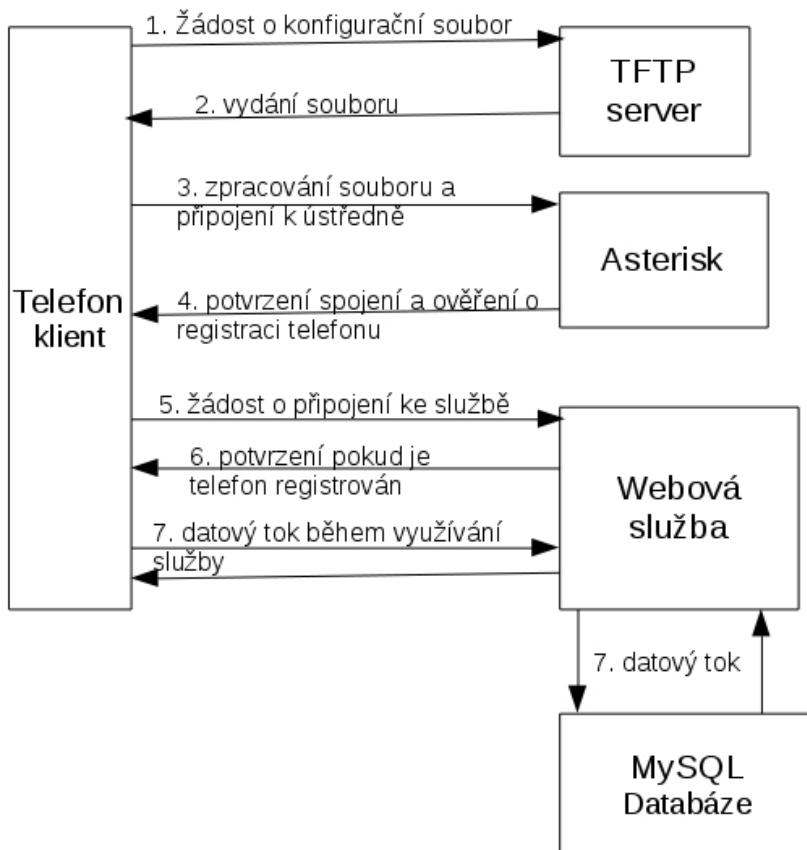
### 3.5 Web/XML služba pro Cisco IP telefony řady 79XX

Web/XML služba je založena na přeposílání XML souborů mezi klientskou stranou, v případě mé práce IP telefonem Cisco 7962, a serverovou stranou, přesněji web/XML službou.

Umístění webové služby

V mé práci jsem službu umístil na stejném virtuálním serveru jako Asterisk a TFTP server. K provozu webové služby jsem použil LAMP. LAMP je skupina svobodného softwaru distribuována pro Linux, jde o Apache, MySQL a PHP.

#### 3.5.1 Znárodnění komunikace klienta se službou



Obr. 6 komunikace klienta(IP telefonu) s web/XML službou

## 3.5.2 Web/XML služba

Webová služba je napsána v php5 s využitím XML šablon, pro telefon, a HTML šablon, pro pc prohlížeče. Služba dodržuje MVC model. Služba je napsána v anglickém jazyce.

K vytvoření služby jsem využil knihovnu xtpl verze 0.2.4-3 distribuovanou pod GNU Lesser General Public License, jedná se o knihovnu na podobné bázi jako SMARTY avšak je menší a slouží převážně ke komunikaci mezi View a Controller.

Cisco telefony dokáží zobrazit obsah služby skrze XML dokumenty se speciálními tagy. XML soubory viz elektronická příloha dodávaná s dokumentem.

### 3.5.2.1 XML část služby pro IP telefon Cisco 79XX

XML tag pro zobrazení menu je párový tag `<CiscoIPPhoneMenu>`, uvnitř něj musí následovat párový tag titulku menu `<Title>`, dále může následovat tag pro nápovědu `<Prompt>`, který je však velmi doporučen.

Pro zobrazení položek menu se používá párových tagů `<MenuItem>`, jenž musí obsahovat tag `<Name>` a `</Name>`, kterými je uvozen název položky menu. Další podstatnou částí položky menu je párový tag `<URL>` s URL adresou kam se telefon má připojit pro zobrazení obsahu dané položky.

Telefon po přečtení tohoto XML zobrazí menu s s položkami vždy jedna položka s jménem na řádek, které může uživatel v telefonu vybrat příkazem select načte se telefon dotáže na adresy zadané v tagu URL.

XML tagy pro vložení vstupu z telefonu uživatelem jsou `<CiscoIPPhoneInput>` a `</CiscoIPPhoneInput>`. Opět je povinnou součástí párový tag `<Title>` a doporučený párový tag `<Prompt>`. Dále je však nutný párový tag `<URL>` obsahující URL adresu, kam se mají vyplněná data vložená uživatelem odeslat.

Poté následují párové tagy `<InputItem>` určující položku k vyplnění, mezi těmito tagy je třeba vložit tyto párové tagy:

`<DisplayName>`

Informace v tomto tagu se zobrazí uživateli na displeji telefonu, aby věděl co má do položky vyplnit.

`<QueryStringParam>`

Tento tag obsahuje název proměnné, v které bude informace uložena.

`<InputFlags>`

Tento tag určí co je možné vložit za znaky do položky k vyplnění.

`<DefaultValue>`

Jedná se o tag nesoucí výchozí hodnotu, kterou je položka pro vyplnění vyplněna.

Telefon po přečtení tohoto XML zobrazí formulář s nadefinovanými položkami k vyplnění.

XML tagy pro zobrazení informací o kontaktu jsou `<CiscoIPPhoneDirectory>` a `</CiscoIPPhoneDirectory>`. Tyto tagy opět obsahují tag pro titulek a nápovědu. Dále pak obsahují párové tagy `<DirectoryEntry>` určující položku v diáři kontaktů. Ty musí obsahovat párový tag

<Name> a párový tag <Telephone>, párový tag používám pro zobrazení zda jde o telefon do kanceláře, domů či mobil a párový tag <Telephone> používám pro zobrazení telefonního čísla daného telefonu.

Telefon po přečtení tohoto XML zobrazí všechna telefonní čísla, která kontakt vlastní.

V projektu jsem použil nejvíce tyto tři typy XML dokumentů, menu, input a directory.

Dále jsem použil obdobu menu dokumentu, pro zobrazení obrázků uvozené párovým tagem <CiscoIPPhoneGraphicMenu> jeho náležitosti jsou obdobné s tagem pro zobrazení textového menu, ke kterým se přidávají tagy nesoucí informaci o obrázku a to pozici x a y levého horního rohu obrázku vůči pozici kde telefon běžně zobrazuje položky menu, dále šířku, výšku a barevnou hloubku obrázku. Následují data obrázku. Cisco telefony umí zobrazit obrázky převedené do Hexa dat a vložené mezi tagy <data></data>. Požadavky které musí splňovat obrázek se liší podle modelu telefonu. Pro model 7962G řady 79XX to jsou tyto požadavky:

1. obrázek musí mít barevnou hloubku 2
2. obrázek nesmí přesáhnout šířkou 256 pixelů a výškou 120 pixelů.

PHP soubory generující obsah pro XML šablony viz elektronická příloha.

## 4 Závěr

Tato práce je rozdělena na dvě hlavní části, jenž jsou dále děleny na dílčí celky. Ve své práci jsem nejprve popsal softwarovou pobočkovou ústřednu Asterisk a jak ji ze zdrojových kódů zkompileovat, nainstalovat a její konfiguraci. Následovně jsem popsal problematiku TFTP protokolu využitého k zavedení dat nastavení klientského zařízení, jimiž byly IP telefony firmy Cisco modelové řady 79XX. Funkčnost TFTP serveru jsem ukázal na odeslání a přijmutí textového souboru s „Hello world“.

Dále jsem se věnoval problematice číslovacího plánu, jenž jsem rozebral v oblastech ovlivňující téma mé práce. Také jsem čtenáře seznámil s protokoly používaných ústřednou Asterisk. Možnost jak dosáhnout telefonního spojení jsem ukázal na číslovacím plánu a SIP účtech.

V druhé části jsem se věnoval rozboru integrace web/XML služeb s softwarovou pobočkovou ústřednou Asterisk. Nejprve jsem se zaměřil na integraci externí databáze s ústřednou Asterisk a konektivitu mezi těmito celky jsem dosáhl pomocí ODBC konektoru nabízející možnost integrace ústředny Asterisk s širokou škálou databází. Nato jsem navázal rozbohem a vývojem ústřednou generovaným registrem zmeškaných, přijatých a odchozích hovorů. Toho jsem dosáhl navázání na ODBC konektivitu za implementováním funkcí do číslicového plánu využívaje modul ústředny Asterisk pro vytěžování a zadávání dat do databáze přes již zmíněnou ODBC konektivitu.

Poté jsem následoval samotným aktem svázání číslicového plánu využívaným ústřednou Asterisk s externí databází kontaktů přes již zmíněnou konektivitu. Tím, že jsem opětovně používal jednotné spojení s databází, jsem dosáhl ucelenější integrace. Implementoval jsem také procházení této databáze kontaktů uživatelem na klientském zařízení. Jako závěrečný prvek integrace jsem následně implementoval modifikaci volby rychlého vytáčení uživatelem.

V konečné části jsem rozebral implementaci a prvky web/XML části služby pro IP telefony Cisco řady 79XX. Systém jsem vyzkoušel na IP telefonu Cisco 7962.

Nad touto prací jsem strávil velké množství času především v porozumění jednotlivých prvků, z nichž je toto téma sestaveno, u mnohých z nich to bylo mé první setkání. Další částí pak byla instalace a konfigurace jednotlivých technologických celků. Nakonec samotná implementace, integrace a reimplementace témata mé práce. Ústředna Asterisk je velmi mocný příklad softwarové pobočkové ústředny, která je volně dostupná pod open source licencí, práce s touto ústřednou však vyžaduje hluboké vědomosti v oboru telefonní komunikace pro její maximální využití. I přesto jsem nabyt mnohých nových znalostí.

Následně jsem přešel k tomu jak integrovat web/XML služby s ústřednou Asterisk vyvinout serverovou část systému integrovanou se službami ústředny tak, aby se telefon mohl svým prohlížečem připojit a zobrazovat informace služby.

## 5 Seznam použité literatury

[1] Digium Inc. What is Asterisk? In: asterisk.org [online]. [vid. 2015-10-02].  
Dostupné z: <http://www.asterisk.org/get-started>

[2] VOIP-Info.org LLC. Chan\_skinny In: VOIP-Info.org [online]. 04. Dubna 2010 [vid. 2015-09-23]. Dostupné z: [http://www.voip-info.org/wiki/view/chan\\_skinny](http://www.voip-info.org/wiki/view/chan_skinny)  
A VOIP-Info.org LLC. chan\_sccp-b In: VOIP-Info.org [online]. 23. Května 2012 [vid. 2015-09-23]. Dostupné z: [http://www.voip-info.org/wiki/view/chan\\_sccp-b](http://www.voip-info.org/wiki/view/chan_sccp-b)

[3] [janru@cesnet.cz](mailto:janru@cesnet.cz). SIP In: .cesnet.cz [online]. 11. Března 2012 [vid. 2015-09-23].  
Dostupné z: : <https://sip.cesnet.cz/cs/protokoly/sip>

[4] Štěpán Hon. VoIP protokol IAX. In siphone.eu [online]. [vid. 2015-09-23]. Dostupné z: <http://www.siphone.eu/cs/wiki/voip-protokol-iax>

[5] Karen R. Sollins. The TFTP Protocol In: isi.edu [online]. 29. Ledna 1980 [vid. 2015-10-02]. Dostupné z: <http://www.isi.edu/in-notes/ien/ien133.txt>

[6] Skinny Client Control Protocol. In: Wikipedia: the free encyclopedia [online]. 27. Listopadu 2014 [vid. 2015-09-24].  
Dostupné z: [http://en.wikipedia.org/wiki/Skinny\\_Call\\_Control\\_Protocol](http://en.wikipedia.org/wiki/Skinny_Call_Control_Protocol)

[7] Český telekomunikační úřad. Číslovací plán veřejných telefonních sítí [online] 25. Zář 2010 [vid. 2015-09-23]. Dostupné z: <https://www.ctu.cz/cs/download/cislovaci-plany-archiv/cislovaci-plan-verejnych-telefonnich-siti-1114435245.pdf>



## 6 Přílohy

### 6.1. sip.conf

```
[general]
disallow=all
allow=ulaw
allow=alaw
allow=gsm
qualify=yes
canreinvite=no
bindport=5060
bindaddr=147.32.211.90 ;adresa siptestu2
subscribecontext=from-sip
registertimeout=20
registerattempts=10
```

```
[jarda]
type=friend
username=jarda
userid=jarda <17>
context=praha
host=dynamic
secret=jarda
```

```
[jarda2]
type=friend
username=jarda2
userid=jarda2 <27>
context=praha
host=dynamic
secret=jarda2
```

### 6.2. extension.conf

```
;  
; Static extension configuration file. This is where you  
; configure all your inbound and outbound calls in Asterisk.  
;  
;  
; The "General" category is for certain variables.  
;  
[general]  
;  
static=yes  
writeprotect=yes  
  
; The "Globals" category contains global variables that  
; can be referenced in the dialplan with ${VARIABLE} or  
; ${ENV(VARIABLE)} for Environmental variable  
; ${${VARIABLE}} or ${text${VARIABLE}} or any hybrid
```

```

;
[globals]
; =====
; ===== Dialplan =====
; =====
TRUNK=Zap/g1          ; Trunk interface trough Zap group 1
TRUNKMSD=0           ; MSD digits to strip (usually 1 or 0)
;TRUNK=IAX2/biggateway
[praha]
exten => _9.,1,Dial(${TRUNK}/${EXTEN:1})
    ; Press any 7 digit number and try to dial that number through Zap
exten => _9.,2,Goto(s-${DIALSTATUS},1)
    ; Jump based on status (NOANSWER,BUSY,CHANUNAVAIL,CONGESTION

exten => 100,1,Answer
exten => 100,n,Playback(demo-echotest)
exten => 100,n,Echo()
exten => 100,n,Hangup()
;SIP users
exten => 17,1,Dial(SIP/jarda,30)
exten => 27,1,Dial(SIP/jarda2,30)
exten => 17,n,Goto(s-${DIALSTATUS},1)
exten => 27,n,Goto(s-${DIALSTATUS},1)
    ; Jump based on status (NOANSWER,BUSY,CHANUNAVAIL,CONGESTION

exten => s-NOANSWER,1,Voicemail(17,u)
    ; If unavailable
exten => s-NOANSWER,2,Hangup()
exten => s-BUSY,1,Playback(the-party-you-are-calling)
    ; If busy, play busy announce
exten => s-BUSY,2,Playback(is-currntly-busy)
exten => s-BUSY,3,Hangup()
exten => s-CHANUNAVAIL,1,Playback(pbx-invalid)
exten => s-CHANUNAVAIL,2,Hangup()
; now we handle invalid extensions
exten => i,1,Playback(pbx-invalid)
exten => i,n,Hangup()
; ... and timeouts
exten => t,1,Playback(vm-goodbye)
exten => t,n,Hangup()

```

## 6.3. TFTP

```

service tftp
{
protocol          = udp
port              = 69
socket_type      = dgram
wait              = yes
user              = nobody
server            = /usr/sbin/in.tftpd
server_args      = /tftpboot
disable           = no
}

```

## 6.4. SEP203706174FD3.cnf.xml

```
<device>
  <versionStamp>{7942 Aug 06 2008 14:23:48}</versionStamp>
  <devicePool>
    <dateTimeSetting>
      <dateTemplate>D.M.Y</dateTemplate>
      <timeZone>Central Europe Standard/Daylight Time</timeZone>
      <olsonTimeZone>Etc/GMT</olsonTimeZone>
    </dateTimeSetting>
    <callManagerGroup>
      <members>
        <member priority="0">
          <callManager>
            <ports>
<ethernetPhonePort>2000</ethernetPhonePort>
              </ports>
<processNodeName>192.168.21.128</processNodeName>
            </callManager>
          </member>
        </members>
      </callManagerGroup>
    </devicePool>
    <loadInformation></loadInformation>
    <directoryURL>http://siptest2.feld.cvut.cz/myservice/PhoneUI/index.php</directo
ryURL>
    <idleURL></idleURL>
    <idleTimeout>30</idleTimeout>
    <informationURL>http://siptest2.feld.cvut.cz/myservice/PhoneUI/index.php</infor
mationURL>
    <messagesURL></messagesURL>
    <servicesURL>http://siptest2.feld.cvut.cz/myservice/PhoneUI/index.php</services
URL>
    <userLocale>
      <name>cs_CS</name>
      <langCode>cs</langCode>
    </userLocale>
    <networkLocale>Czech_Republic</networkLocale>
    <networkLocaleInfo>
      <name>Czech_Republic</name>
      <version>4.1.0 (2) </version>
    </networkLocaleInfo>
    <vendorConfig>
      <settingsAccess>1</settingsAccess>
      <autoSelectLineEnable>1</autoSelectLineEnable>
      <autoCallSelect>1</autoCallSelect>
      <ehookEnable>1</ehookEnable>
      <g722CodecSupport>2</g722CodecSupport>
    <headsetWidebandUIControl>0</headsetWidebandUIControl>
    <handsetWidebandUIControl>0</handsetWidebandUIControl>
      <headsetWidebandEnable>0</headsetWidebandEnable>
      <handsetWidebandEnable>1</handsetWidebandEnable>
    </vendorConfig>
    <advertiseG722Codec>1</advertiseG722Codec>
    <phonePersonalization>1</phonePersonalization>
    <singleButtonBarge>1</singleButtonBarge>
```

</device>

## 6.5 ODBC/MySQL

```
soubor: /etc/odbcinst.ini
[MySQL]
Description = ODBC for MySQL
Driver = /usr/lib/odbc/libmyodbc.so
Setup = /usr/lib/odbc/libodbcmyS.so
FileUsage = 1
```

## 6.6 ODBC/Asterisk

```
soubor: /etc/odbc.ini
[asterisk-connector]
Description = MySQL connection to 'asterisk' database
Driver = MySQL
Database = asterisk
Server = localhost
UserName = asterisk
Password = welcome
Port = 3306
Socket = /var/lib/mysql/mysql.sock
```