Asterisk Gateway Interface (AGI) Scripting in Python

Muhammad Morshed Alam

morshed@amberit.com.bd

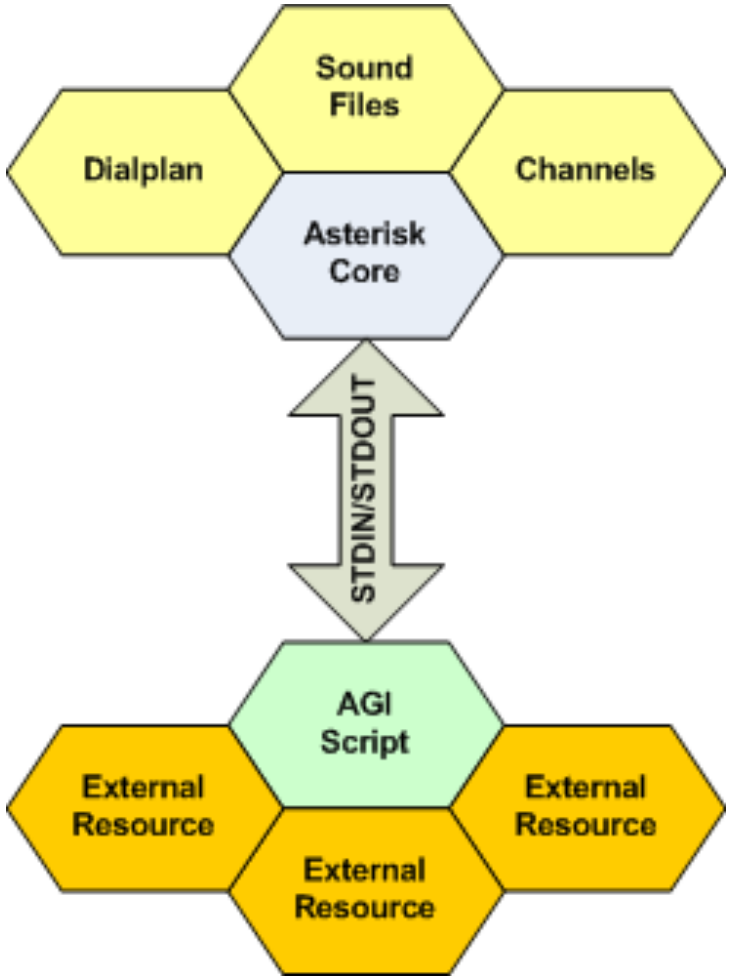
AmberIT Ltd.

List of Talk:

- Short brief on Asterisk Gateway Interface (AGI)
- AGI execution flow and how it works
- Requirements to set AGI Lab
- Case Study-1: Querying numeric data from DB using AGI scripting in python
- Case Study-2: Inserting / Updating data into the DB using AGI scripting

Note: all done by user over a live connected sip call session

Into. to AGI scripting



- AGI helps to execute scripts in asterisk Dial Plan
- AGI connect asterisk to remote database or external world
- User Connected in live SIP Channels

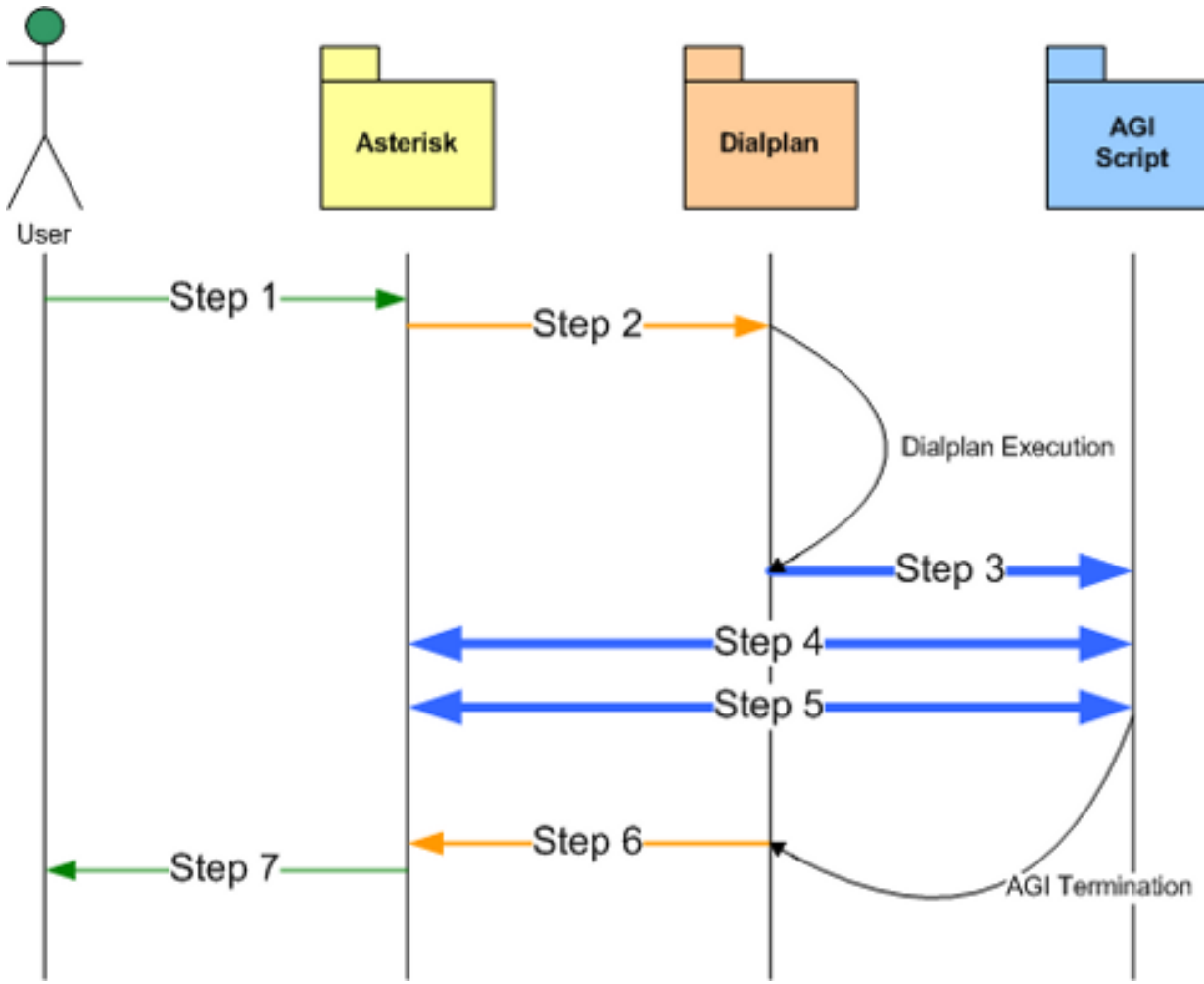
>> insert numeric data through DTMF

>> Pass as input arguments to AGI script

>> Get return the required data from AGI

>> playback to user connected in live channel

AGI Execution Flow:



- Asterisk use **Read** application to take input from the caller and pass it to AGI script as input arguments. => **Read** (variable name to save data, voice to enter data, max length of the input data)
- Asterisk use **AGI** application to connect with DB using a script written in python or even in php
- AGI script return the values and Asterisk use application to playback the required numeric data

List of AGI Variables:

AGI Variable	Description
agi_request	Name of the agi script that is being called.
agi_channel	Channel that the call is coming from.
agi_language	Language that is configured on the server.
agi_type	Call type, mainly the channel type.
agi_uniqueid	A unique identifier for this session.
agi_callerid	**Caller ID number.**
agi_calleridname	Caller ID name, where available, not supported on all channel types.
agi_callingpres	PRI Call ID presentation variable.
agi_callingani2	Caller ANI2 (PRI channels), where applicable.
agi_callington	Caller type of number (PRI channels).
agi_callingtns	Transit Network Selector (PRI channels).
agi_dnid	**Dialed number identifier.**
agi_rdnis	Redirected Dial Number ID Service (RDNIS).
agi_context	**Current context from which the AGI script was executed.**
agi_extension	Extension that was called.
agi_priority	Current priority in the dialplan, that is, priority of the AGI script execution.

Requirements to Launch AGI Scripts:

Requirements:

- **apt-get install python-mysqldb** ;mysql module dependency for python
- You should have: Asterisk install machine (Asterisk 1.8) with a SIP Trunk (**/etc/asterisk/sip.conf**) and some routing scripts at (**/etc/asterisk/extension.conf)**

- Asterisk default AGI script location: **/var/lib/asterisk/agi-bin/**
- Python AGI script extension: result.py

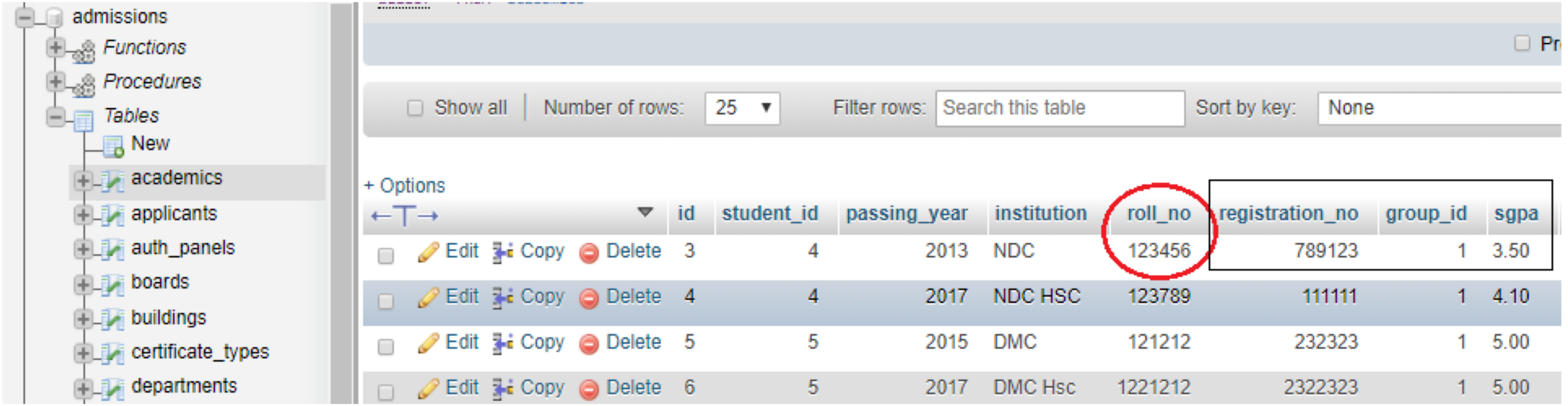
- Permission Required to Execute: **chmod 755 result.py**

AGI Scripting Sample Case-1 (Data Query from DB)

Case-1: Querying Student Result Database against the ID/ Roll Number

- Call to the number 09611000196 give the entry of Student Roll number through DTMF
- AGI will send query to DB against the given roll number
- AGI will return particular student registration number and current CGPA
- Asterisk will playback the return numeric value in live connected channel

Result DB:



⇒ **User give entry roll_no (123456)**

<= AGI will return the registration_no (789123) & CGPA (3.5)

AGI scripts in python <Data Query from DB>

```
#!/usr/bin/python

import agi
import MySQLdb
import sys
agi = agi.AGI()

MYSQL_HOST='127.0.0.1'
MYSQL_USER='root'
MYSQL_PASS='morshedtest'
MYSQL_DB='admissions'

rolno=sys.argv[1]

mysql = MySQLdb.connect(host=MYSQL_HOST , user=MYSQL_USER, passwd=MYSQL_PASS, db=MYSQL_DB)
db = mysql.cursor()
db.execute("""SELECT registration_no,sgpa FROM academics WHERE  roll_no='%(rolno)s' """ %vars())

result = db.fetchall()
db.close()

regno = int(result[0][0])
cgpa = float(result[0][1])

agi.set_variable("REGNO", regno)
agi.set_variable("CGPA", cgpa)
```

Asterisk Applications:

- SayDigits (123) => reads a specified number one digit at a time (one two three and so on)
- SayAlpha () => used to spell alphanumeric strings back to the caller
- SayNumber (numeric data) => reads the whole number (123, one hundred and twenty three)

Writing Dial Plan in extension.conf for AGI script launching:

Dial Plan at /etc/asterisk/extension.conf:

```
exten => 09611000196,1,Answer()
exten => 09611000196,n,read(rolno,enterrollnov)  ; take the input of the roll no from DTMF
exten => 09611000196,n,AGI(result.py,${rolno}) ; launch the AGI script
exten => 09611000196,n,NoOp(${REGNO})
exten => 09611000196,n,NoOp(${CGPA})
exten => 09611000196,n,playback(regnois)
exten => 09611000196,n,sayalpha(${REGNO}) ;play back the AGI return variable (Registration no) from db
exten => 09611000196,n,playback(cgpais)
exten => 09611000196,n,sayalpha(${CGPA}) ;play back the AGI return variable (CGPA) from db
exten => 09611000196,n,Hangup()
```

Let’s Debug AGI in Asterisk CLI:

localhost*CLI> agi set debug on

AGI Debug in Asterisk CLI

-- Executing [09611000196@default:1] **Answe**r("SIP/sipksecu-0003134b", "") in new stack
-- Executing [09611000196@default:2] **Read**("SIP/sipksecu-0003134b", "rolno, **enterrollnov**") in new stack
-- <SIP/sipksecu-0003134b> **Playing** 'rolnov.slin' (language 'en')
-- **User entered '123456'**
-- Executing [09611000196@default:3] **AGI**("SIP/sipksecu-0003134b", "result.py,123456") in new stack
-- **Launched AGI Script /var/lib/asterisk/agi-bin/result.py**
<SIP/sipksecu-0003134b>AGI Tx >> agi_request: result.py
<SIP/sipksecu-0003134b>AGI Tx >> **agi_channel:** SIP/sipksecu-0003134b
<SIP/sipksecu-0003134b>AGI Tx >> agi_language: en
<SIP/sipksecu-0003134b>AGI Tx >> agi_type: SIP
<SIP/sipksecu-0003134b>AGI Tx >> agi_uniqueid: 1529916407.201547
<SIP/sipksecu-0003134b>AGI Tx >> agi_version: 1.8.32.1
<SIP/sipksecu-0003134b>AGI Tx >> **agi_callerid:** 09611000038
<SIP/sipksecu-0003134b>AGI Tx >> **agi_calleridname:** 09611000038
<SIP/sipksecu-0003134b>AGI Tx >> agi_callingpres: 0
<SIP/sipksecu-0003134b>AGI Tx >> agi_callingani2: 0
<SIP/sipksecu-0003134b>AGI Tx >> agi_callington: 0
<SIP/sipksecu-0003134b>AGI Tx >> agi_callingtns: 0
<SIP/sipksecu-0003134b>AGI Tx >> **agi_dnid:** 09611000196

AGI Debug in Asterisk CLI

SIP/sipksecu-0003134b>AGI Tx >> agi_rdnis: unknown
<SIP/sipksecu-0003134b>AGI Tx >> **agi_context: default**
<SIP/sipksecu-0003134b>AGI Tx >> **agi_extension: 09611000196**
<SIP/sipksecu-0003134b>AGI Tx >> agi_priority: 3
<SIP/sipksecu-0003134b>AGI Tx >> agi_enhanced: 0.0
<SIP/sipksecu-0003134b>AGI Tx >> agi_accountcode:
<SIP/sipksecu-0003134b>AGI Tx >> agi_threadid: 140260657370880
<SIP/sipksecu-0003134b>AGI Tx >> agi_arg_1: 123456
<SIP/sipksecu-0003134b>AGI Tx >>
<SIP/sipksecu-0003134b>AGI Rx << SET VARIABLE "REGNO" "789123"
<SIP/sipksecu-0003134b>AGI Tx >> 200 result=1
<SIP/sipksecu-0003134b>AGI Rx << SET VARIABLE "CGPA" "3.5"
<SIP/sipksecu-0003134b>AGI Tx >> 200 result=1
-- <SIP/sipksecu-0003134b>AGI Script result.py completed, returning 0
-- Executing [09611000196@default:4] **Playback**("SIP/sipksecu-0003134b", "regnois") in new stack
-- <SIP/sipksecu-0003134b> Playing 'regv.slin' (language 'en')
-- Executing [09611000196@default:5] **SayAlpha**("SIP/sipksecu-0003134b", **"789123")** in new stack

AGI Debug in Asterisk CLI

-- <SIP/sipksecu-0003134b> Playing 'digits/7.gsm' (language 'en')

-- <SIP/sipksecu-0003134b> Playing 'digits/8.gsm' (language 'en')

-- <SIP/sipksecu-0003134b> Playing 'digits/9.gsm' (language 'en')

-- <SIP/sipksecu-0003134b> Playing 'digits/1.gsm' (language 'en')

-- <SIP/sipksecu-0003134b> Playing 'digits/2.gsm' (language 'en')

-- <SIP/sipksecu-0003134b> Playing 'digits/3.gsm' (language 'en')

-- Executing [09611000196@default:6] **Playback**("SIP/sipksecu-0003134b", "cgpais") in new stack

-- <SIP/sipksecu-0003134b> Playing 'cgpav.slin' (language 'en')

-- Executing [09611000196@default:7] **SayAlpha**("SIP/sipksecu-0003134b", **"3.5")** in new stack

-- <SIP/sipksecu-0003134b> Playing 'digits/3.gsm' (language 'en')

-- <SIP/sipksecu-0003134b> Playing 'letters/dot.gsm' (language 'en')

-- <SIP/sipksecu-0003134b> Playing 'digits/5.gsm' (language 'en')

AGI Scripting Sample Case-2 (Inserting Data to DB)

- Set or reset user pin through Interactive Response
- Off course verified by the contact center Agent and then transfer to an special extension to set/reset the pin
- We need user account no or even the 16 digit ATM card no
- Set a pin through IVR that will be inserted into DB

AGI Scripting Sample Case-2 (Inserting Data to DB)

```
#!/usr/bin/python
import agi
import MySQLdb
import sys
agi = agi.AGI()

MYSQL_HOST='127.0.0.1'
MYSQL_USER='root'
MYSQL_PASS='test123'
MYSQL_DB='bankacc'

atmcardno=sys.argv[1]
pinno=sys.argv[2]

mysql = MySQLdb.connect(host=MYSQL_HOST , user=MYSQL_USER, passwd=MYSQL_PASS, db=MYSQL_DB)
db = mysql.cursor()
db.execute("""UPDATE banktb SET Pinno='%(pinno)s' WHERE Atmcardno='%(atmcardno)s'""" %vars())
result = db.fetchall()
db.close()
```

Note:
Atmcardno can be a 16 digit no
Pinno is the secret key given by the user

Writing Dial Plan in extension.conf for AGI Script Launching:

```
exten => 09611000196,1,Answer()

exten => 09611000196,n,Read(atmcardno,enteratmcardno,16)  ;take the 16 digit atm card no through dtmf

exten => 09611000196,n,NoOp(${atmcardno})

exten => 09611000196,n,Read(pinno,enterpinno,4) ;take the 4 digit pin no through dtmf

exten => 09611000196,n,NoOp(${pinno})

exten => 09611000196,n,AGI(bankpin.py,${atmcardno}, ${pinno}))  ;launch the AGI to insert new pin to db
against the given atm card no

exten => 09611000196,n,Hangup()
```

$ QUESTIONS ?