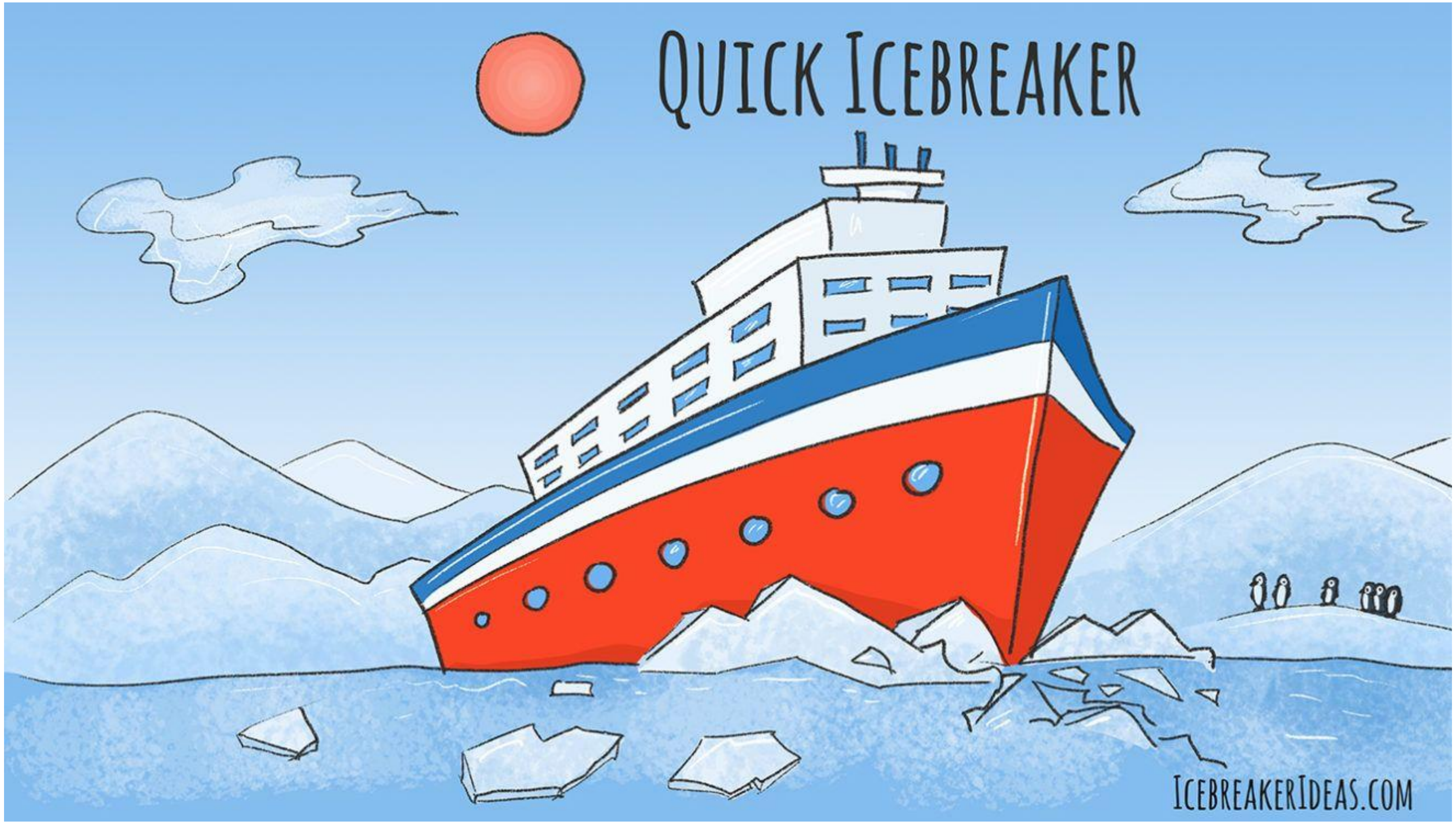
ASTERISK

Essentials 1

Teknowlogic
The Power of Knowledge


Quick Icebreaker
IcebreakerIdeas.com

Hello, i’m ALFONSO AYALA

MAGISTER EN INGENIERÍA

RESUMEN – HOJA DE VIDA (PERFIL PROFESIONAL)

Magíster en Ingeniería en el área de Sistemas y computación de la Universidad Nacional de Colombia. Especialista en Seguridad de la Información de la Universidad de los Andes, Especialista en Docencia Universitaria de la Universidad Cooperativa, Profesional en Ingeniería de Sistemas de la Universidad Nacional de Colombia. Catedrático en las Universidades Cooperativa y del Tolima. Amplia experiencia en proyectos de desarrollo de sistemas de información, herramientas de soporte a toma de decisiones, proyectos Asterisk * y coaching de Innovación.

Contenido

- Asterisk Essentials

What is Asterisk

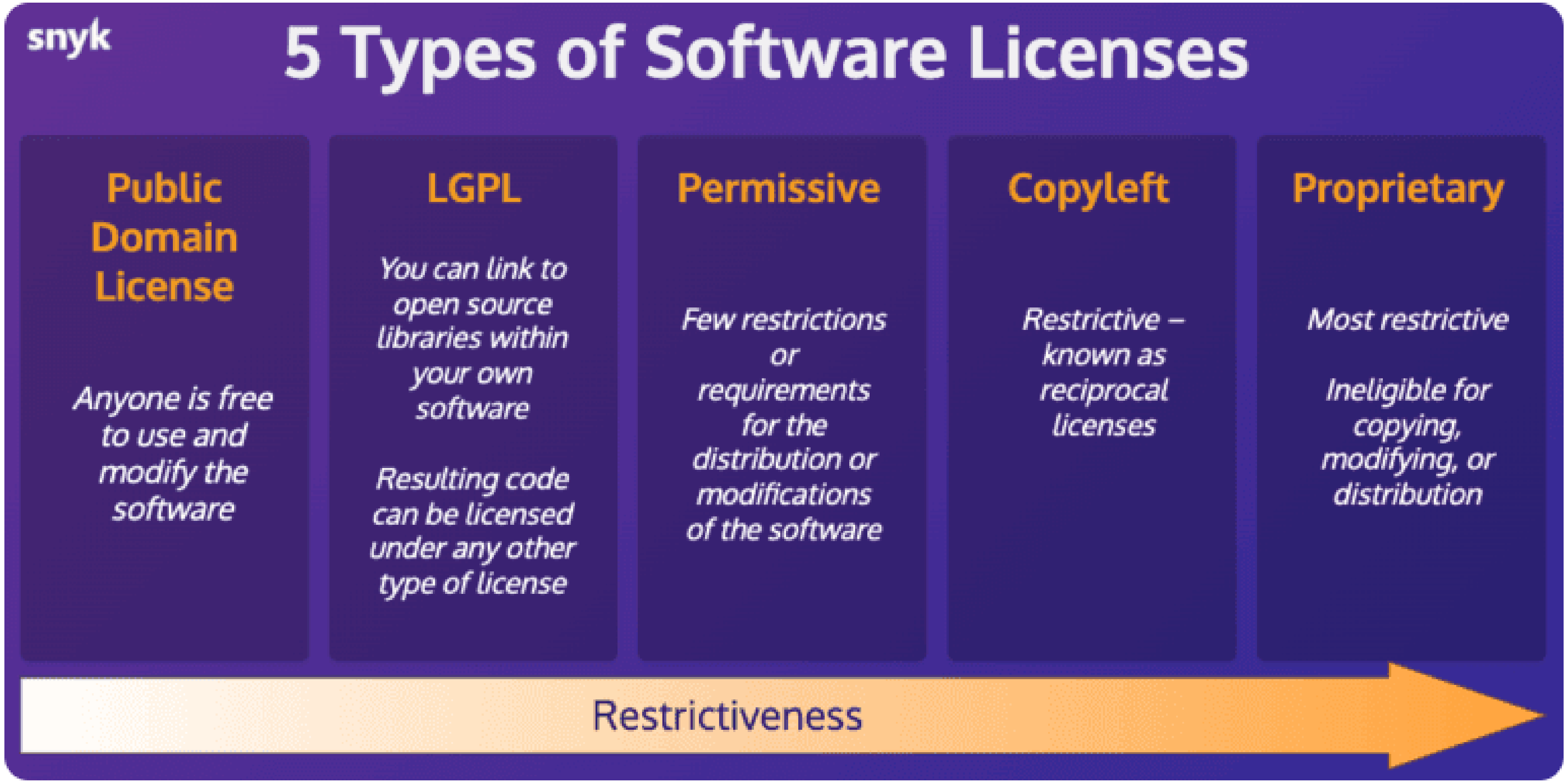
Asterisk

- Open source Telephony Platform
- Deals with telephony
- Analog, PSTN, VoIP
- Switching
- Full Feature PBX

Asterisk is

- Featureful
- Compatible
- Scalable
- Supportable
- Free
- GNU GPLv3

Licences



TKL Teknowlogic
The Power of Knowledge

Asterisk is GPL v3

- Like the GPL v2, GPL 3 is a strong copyleft license, meaning that any copy or modification of the original code must also be released under the GPL v3. In other words, you can take the GPL 3'd code, add to it or make major changes, then distribute your version.

- You may copy, distribute and modify the software as long as you track changes/dates in source files. Any modifications to or software including (via compiler) GPL-licensed code must also be made available under the GPL along with build & install instructions.

Turnkey Solution

- A turnkey solution is essentially a ready-made solution that you can deploy in your business with great ease and simplicity. They are called turnkey solutions because the user essentially needs to just turn the key to start using the solution.

- A turn-key solution is essentially a solution that is built and designed by a vendor who can sell it to any buyer, rather than being built according to the exact specifications of any particular buyer.

Asterisk is not

- Turnkey solution. (Freepbx is)
- Can be used to develop such solutions

Review

- Asterisk is an opensource telephony platform
- It is featureful, compatible, scalable, supportable and free.
- It is GNU GPLv3 (Copyleft)
- It is not a turnkey solution, but can be used to create one.
- Freepbx is a turnkey solution
- Asterisk is maintaned by Sangoma.

What is Digium and Sangoma

Digium

- Created in 1999 by Mark Spencer
- Open source side
  - Asterisk
- Commercial Side
  - Business phone Systems: Commercial PBX
  - Custom Telephony Solutions

Sangoma

- Created in 1984
- 2018 Adquired Digium (USD +28M)
- Open source side
  - Asterisk
- Commercial Side
  - Unified Communications as a Service (UCaaS) PBXACT-Cloud
  - Custom Telephony Solutions
- Mission: to **unite businesses of all sizes**– connecting the people and processes that matter.
- We specialize in UC, video, chat, contact center, MSP services, security, and more!

Review

- Digium
- Sangoma
- Two lines of services: Opensource / Commercial

Asterisk Versioning

Release types

- Two release types “feature frozen”
  - Only Bug fixes and Security Patches
  - New features require new reléase series
- Standard
  - Bug fixes for 1 year
  - Security patches for 2 years
- Long Term Support (LTS)
  - Bug fixes for 4 years
  - Security patches fro 5 years

Asterisk Versioning

Release Series	Release Type	Release Date	Security Fix Only	EOL
21.x	Standard	2023-10-18	2025-10-18	2026-10-18
20.x	LTS	2022-10-19	2026-10-19	2027-10-19
19.x	Standard	2021-11-02	2022-11-02	2023-11-02
18.x	LTS	2020-10-20	2024-10-20	2025-10-20
17.x	Standard	2019-10-28	2020-10-28	2021-10-28
16.x	LTS	2018-10-09	2022-10-09	2023-10-09
15.x	Standard	2017-10-03	2018-10-03	2019-10-03
14.x	Standard	2016-09-26	2017-09-26	2018-09-26
13.x	LTS	2014-10-24	2020-10-24	2021-10-24
12.x	Standard	2013-12-20	2014-12-20	2015-12-20
11.x	LTS	2012-10-25	2016-10-25	2017-10-25
10.x	Standard	2011-12-15	2012-12-15	2013-12-15
1.8.x	LTS	2010-10-21	2014-10-21	2015-10-21
1.6.2.x	Standard	2009-12-18	2011-04-21	2012-04-21
1.6.1.x	Standard	2009-04-27	2010-05-01	2011-04-27
1.6.0.x	Standard	2008-10-01	2010-05-01	2010-10-01
1.4.x	LTS	2006-12-23	2011-04-21	2012-04-21
1.2.x		2005-11-21	2007-08-07	2010-11-21

Asterisk Versioning

- CONCEPT – FEATURE – MINOR
- 20.2.1

Best Practices for Upgrading Asterisk

- Run an Asterisk release that is under current maintenance
- Subscribe to the Asterisk Security mailing list
- Thoroughly plan upgrades
- Install a new version in a test environment

Non-Astersik Releases

- Must Mast version numbers, because code plugs into asterisk
  - G.729
  - Fax for Asterisk
- Separate code, can use different version numbers: code does not plug into asterisk
  - DAHDI
  - libPRI
  - Libss7
  - Asterisk GUI

Review

- Asterisk version numbers
- Asterisk Release history
- Code that use the same versions and code that does not use the same versions.
- Upgrade best practices

Asterisk use cases

Use Cases

- Traditional PBX
- Complete IP PBX
- Hybrid PBX: (Traditional and IP telephony)
- Toll Bypass (Long distance bypass)
- Feature Server (Voicemail, conferencing)
- Call Center (Robust Queuing, several ring strategies)

Review

- Asterisk is capable of much more than a typical PBX
- Open source telephony platform
- All features in Asterisk are free

Asterisk system requirements

System Requirements

- Linux: Redhat, Fedora, Ubuntu, Debian, CentOS
- Miminal requirements: runs on small machines (low quality hw)
- 512MB ram

System Requirements

You better ask:

- What is the intended function of the Asterisk server?
- How many people are going to use it?
- How many simultaneous calls do I expect? (10%)
- Will my system be transcoding audio from one codec to another?
- Will my system be using conferences? If so how many romos adn with how many users?
- Will I be recording any calls? (More disk for /var/)
- What other applications or services do I want to run on my Asterisk?

Review

- Asterisk runs on Numerous hardware platforms
- Any major Linux distribution
- Hardware considerations:
  - What your system Will be used for
  - The effects of low quality hardware can have

Asterisk dependences

Dependences

- We need a package manager: apt-get, yum, yast, pkgtool.
- yum install <package name>
- yum search <package name>
- man yum

- Example:
- yum install newt newt-devel newt-perl -y

C Programming Build Tools

- gcc: compiler
- make: test gcc how to compile
- glibc-devel: provides standard c functions
- autoconf: generates build scripts

Asterisk package requirements

- openssl-devel : cryptography toolkit
- zlib-devel: library for data compression/uncompression
- ncurses-devel: library for terminal handling

DAHDI Dependences

- libnewt: c library for text screen widgets
- kernel-headers: C header files for the linux kernel

libPRI

- ISDN PRI or BRI
- No unique dependences
- Compile LibPRI before Asterisk

Dependency resolution

- yum –C list <pakage name>
- ./configure -help | less
- make menuselect

Review

- Package requirements
  - C programming tools, asterisk packages (openssl-devel, zlib-devel, ncurses-devel), dahdi dependences, libPRI dependences
- Yum package manager
  - Yum install <package>
- Dependence resolution verification
  - yum –C list <pakage name>
  - ./configure -help | less
  - make menuselect

Asterisk installation

Installation options

- From Source (make menuselect)
- Using package manager (example: yum install Asterisk)
  - Dependencies are automatically resolved
  - Less flexible
  - Not the latest version
- Sofware Appliance: ISO
- On Cloud

TKL Teknowlogic
The Power of Knowledge

Additional Packages

- Fax for Asterisk
- G.729
- Sounds (Music on hold)
  - Core sound packages(prompts) (sp, en, fr)
- Extra Sounds Package
  - More audio formats
  - More prompts

Review

- Installation options
  - Package manager
  - Software appliance
  - From source
- Additional Packages
  - Fax for Asterisk
  - G.729
  - Extra sounds package

Asterisk installation from source

Installation from source

- Packages: DAHDI, libPRI, Asterisk
- Asterisk.org
- Download to /usr/src
- wget <url>
- tar -zxvf <archive file name>
- First install DAHDI and LibPRI

Commands

- wget <url>
- tar –zvxf <filename>
- make : compile and create executables.
- make install : copy executables to executables folders
- make config : install an script to start automatically

Commands for Asterisk

- ./configure
- make menuselect
- make
- make install
- Install extra sounds

- make samples
- make config : start Asterisk on boot .

Review

- Download Asterisk, DAHDI, libPRI: wget
- Extract and uncompress: tar –zxvf <filename>
- Order:
  - DAHDI
  - libPRI
  - Asterisk
- Linux commands:
  - ./configure
  - make
  - make install
  - make config

TKL Teknowlogic
The Power of Knowledge

Testing your installation

Testing Your Installation

- service dahdi start
- asterisk –rvvvv
- CLI
  - core show version
  - dahdi show version
  - pri show version

Safe_asterisk

- is a script that runs asterisk in a loop, which can be useful if you fear asterisk may crash.
- runs asterisk with unlimited core file size, and thus Asterisk will dump core in case of a crash
- Runs in a virtual console (9 by default)
- To get a "picture" of console 9, from another terminal (e.g: from a remote shell session) you can use: **screendump 9**

Review

- Test installation
  - asterisk –rvvvv
    - CLI
      - core show version
      - dahdi show version
      - pri show versión
- Safe_Asterisk: runs Asterisk on a loop

Configuration files

Configuration files

- Configuration text files / GUI Configuration .
- Comparation: text files (language) vs. GUI (form letters)
- Asterisk can read your config to a Relational DB
- You can save it to a versioning system
- Configuration files are in: /etc/asterisk

- Make samples creates files in /etc/asterisk
- Asterisk.conf ; global config, directories has the structure: (key => value)
- Extensions.conf ; extensions and contexts
- Sip.conf ;

Asterisk.conf

```
[directories] ;section
;key => value
astetcdir => /etc/Asterisk

[options]  ;section
verbose = 3
```

extensions.conf ;dialplan

[general] ;section

;key => value

static =yes

[global] ;section

; how to handle incoming and outgoing calls.

extensions.conf ;dialplan

```
[general]
static=yes
writeprotect=no
clearglobalvars=no
[globals]
CONSOLE=Console/dsp
IAXINFO=guest
TRUNK=DAHDI/G2
TRUNKMSD=1


[dundi-e164-canonical]
[dundi-e164-customers]
[dundi-e164-via-pstn]
[dundi-e164-local]
include => dundi-e164-canonical
include => dundi-e164-customers
include => dundi-e164-via-pstn
[dundi-e164-switch]
switch => DUNDi/e164
[dundi-e164-lookup]
include => dundi-e164-local
include => dundi-e164-switch
[macro-dundi-e164]
exten => s,1,Goto(${ARG1},1)
include => dundi-e164-lookup
[iaxtel700]
exten => _91700XXXXXXX,1,Dial(IAX2/${GLOBAL(IAXINFO)}@iaxtel.com/${EXTEN:1}@iaxtel)
[iaxprovider]
[trunkint]
exten => _9011.,1,Macro(dundi-e164,${EXTEN:4})
```

extensions.conf ;dialplan

```
[demo]
include => stdexten
exten => s,1,Wait(1)
exten => s,n,Answer
exten => s,n,Set(TIMEOUT(digit)=5)
exten => s,n,Set(TIMEOUT(response)=10)
exten => s,n(restart),BackGround(demo-congrats)
exten => s,n(instruct),BackGround(demo-instruct)
exten => s,n,WaitExten
exten => 2,1,BackGround(demo-moreinfo)
exten => 2,n,Goto(s,instruct)
exten => 3,1,Set(LANGUAGE()=fr)
exten => 3,n,Goto(s,restart)
exten => 1000,1,Goto(default,s,1)
exten => 1234,1,Playback(transfer,skip)

exten => 1234,n,Gosub(${EXTEN},stdexten(${GLOBAL(CONSOLE)}))
exten => 1234,n,Goto(default,s,1)
/[demo
```

sip.conf
;configuration file for the sip channel driver

[general] ;section

context = default

udpbindaddress=0.0.0.0 ; listens on all addresses

;phones, trunks, devices config are defined on its own section

```
[basic-options](!)                    ;a section template (example dozens of phones)
    dtmfmode=rfc2833
    context=from-office
    type=friend
[natted-phone](!,basic-options) ; another template inheriting from basic-options
    nat=yes
    directmedia=no
    host=dynamic
```

Reload configuration

Reload

Sip reload

Review

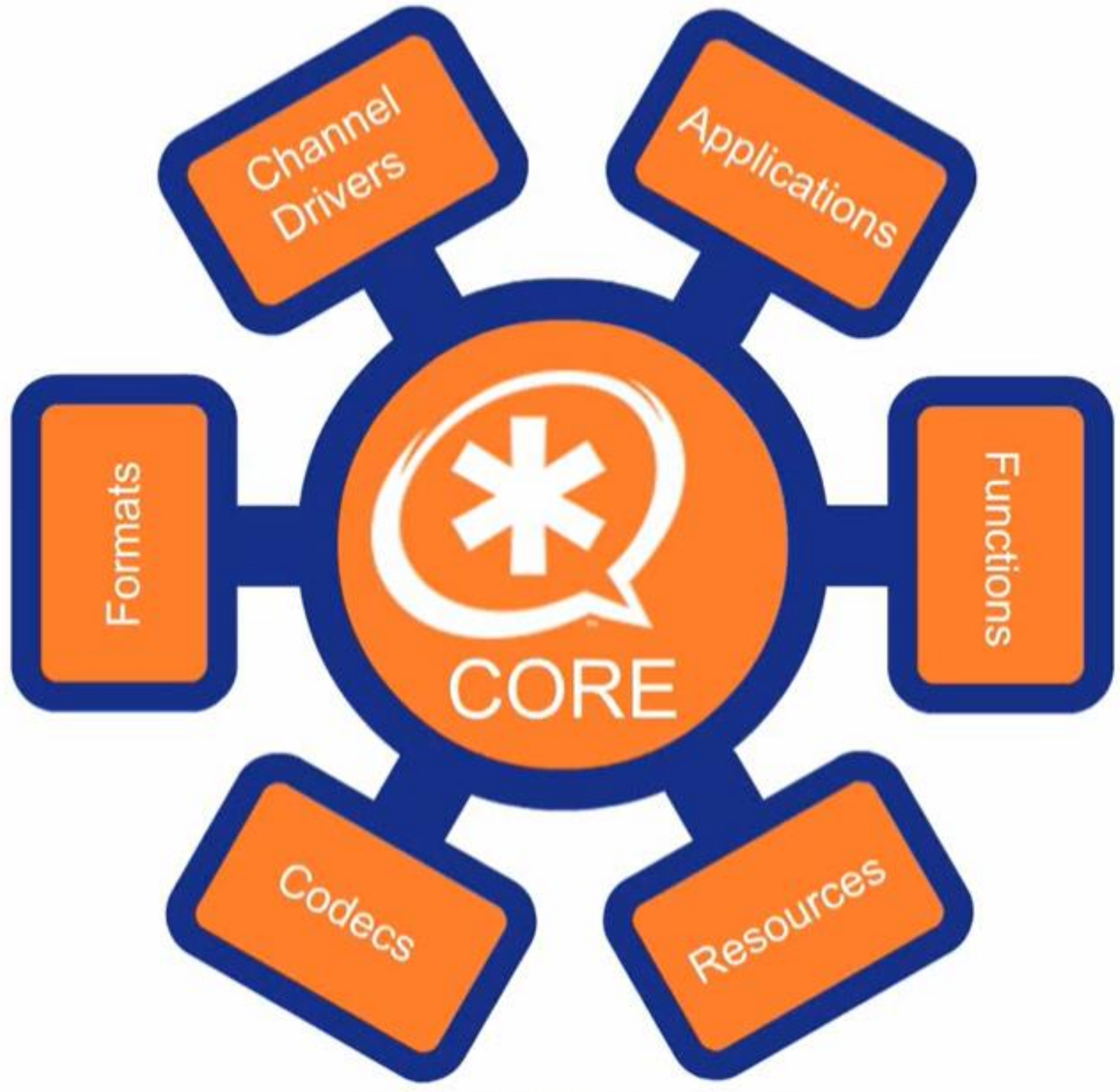
- Asterisk configuration
  - Primarly on text files
  - Lives on /etc/Asterisk
  - Some key files
  - Asterisk.conf
  - Extensions.conf
  - Sip.conf
- Reload configuration
  - Sip reload
  - reload

Asterisk architecture

Architecture

- Modular:
- Core + Modules

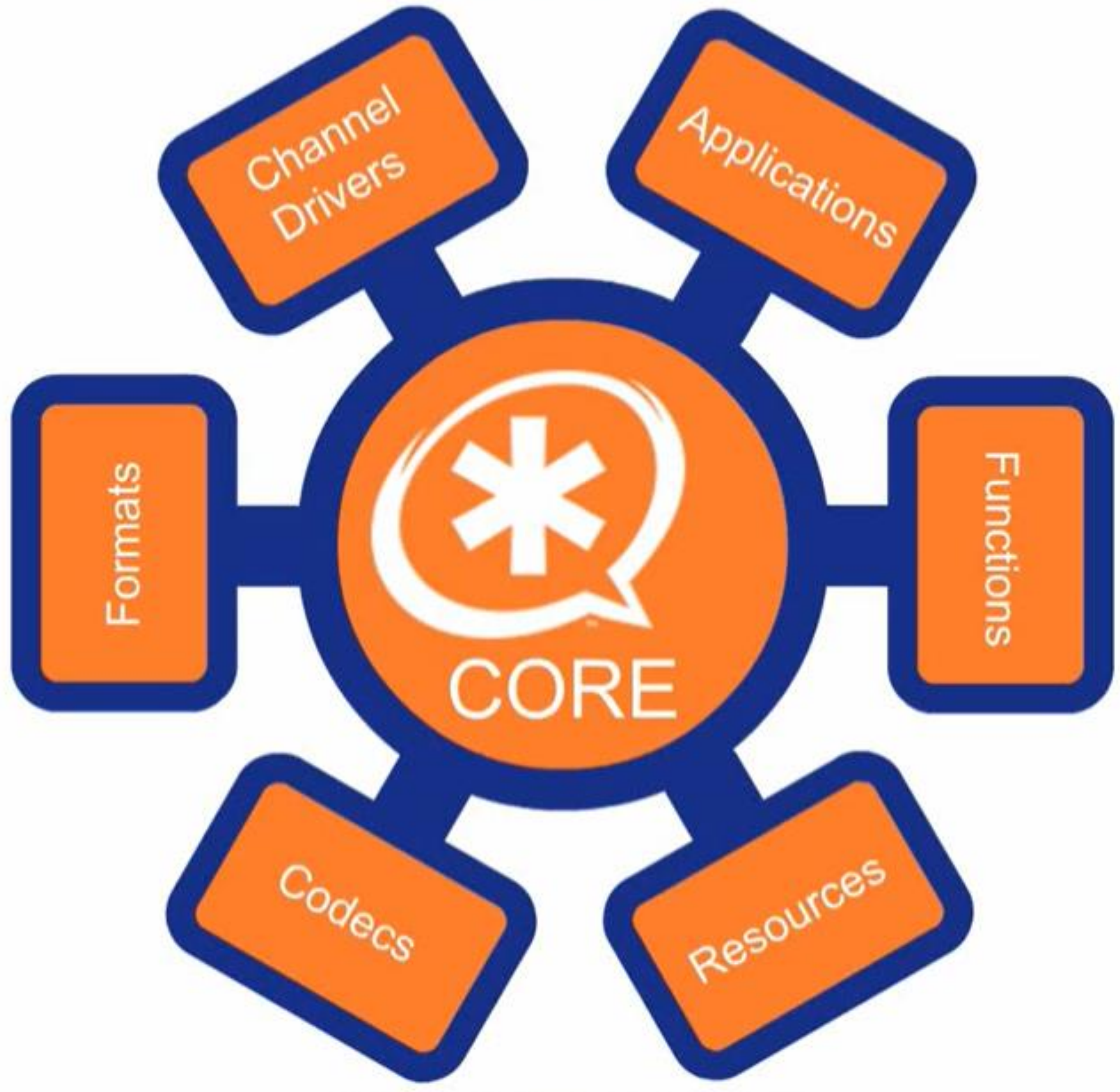
- Module files:
- filename.so (shared object)
- /usr/lib/Asterisk/modules
    - Set by astmoddir in Asterisk.conf
- /etc/Asterisk/modules.conf
    - Automatically loads all
    - Specify individual modules
    - Module load order



TKL Teknowlogic
The Power of Knowledge

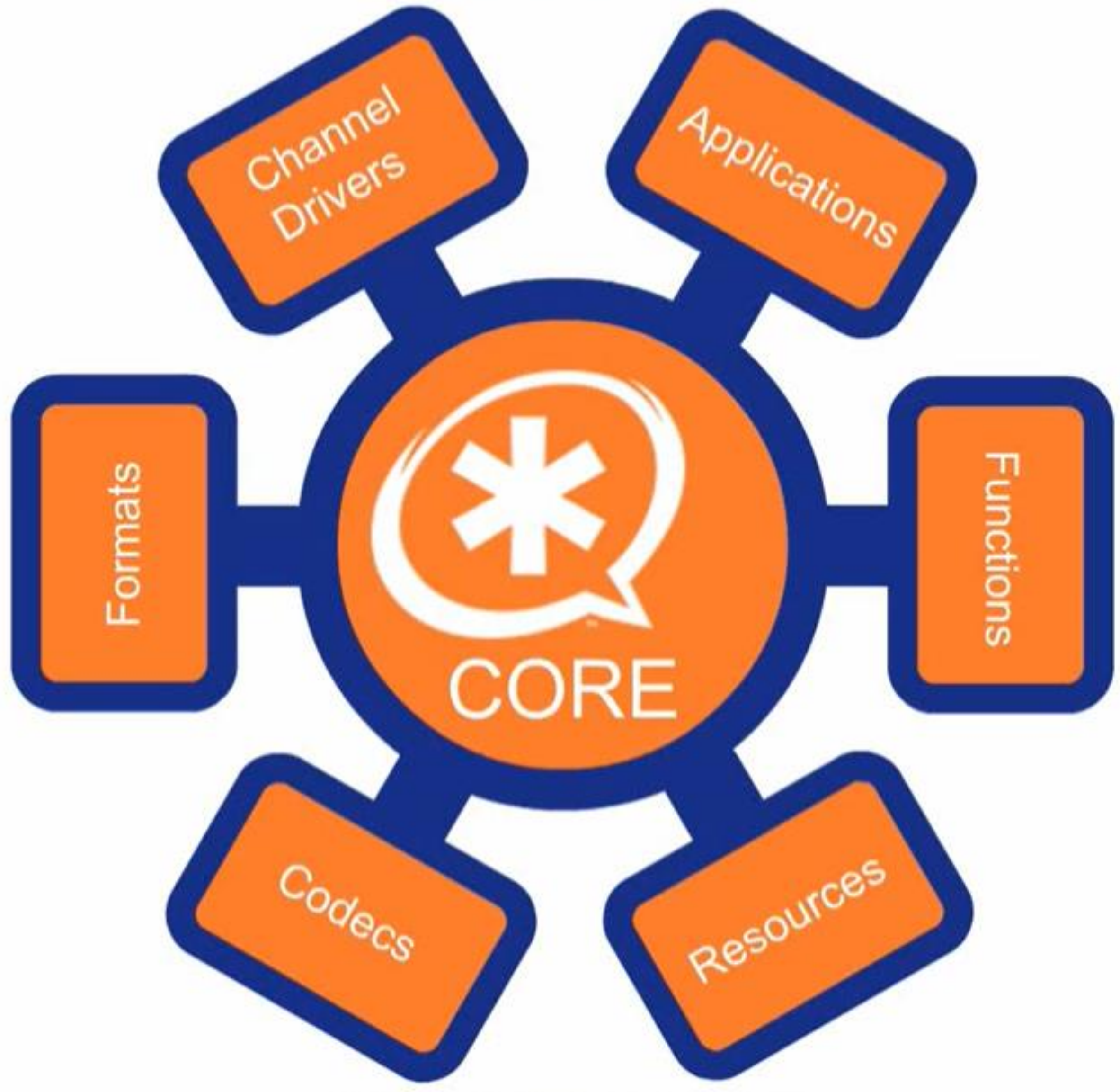
Core responsibilities

- Module management
- Reading configuration
- System timing
- Channel Management



Channel Drivers

- Chan_dahdi.so -> connects to kernel drivers (Traditional telephony)
- Chan_sip.so -> implements RFC3261
- Chan_iax2.so -> implements RFC5456
- Chan_<name>.so
- Standard API for channels
- Functions for Create, destroying, dialing

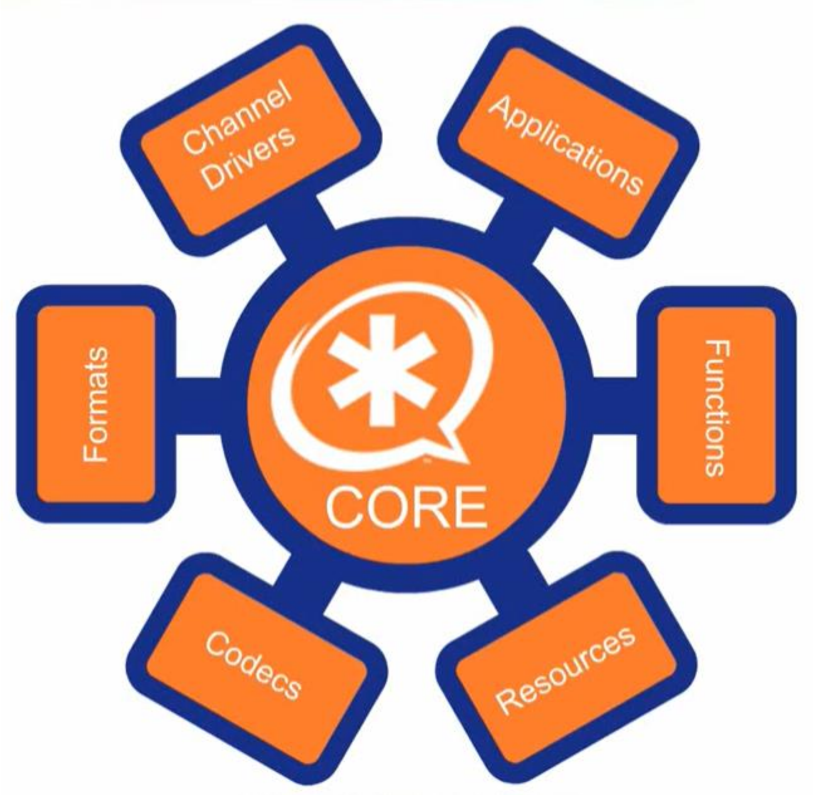


TKL Teknowlogic
The Power of Knowledge

Applications

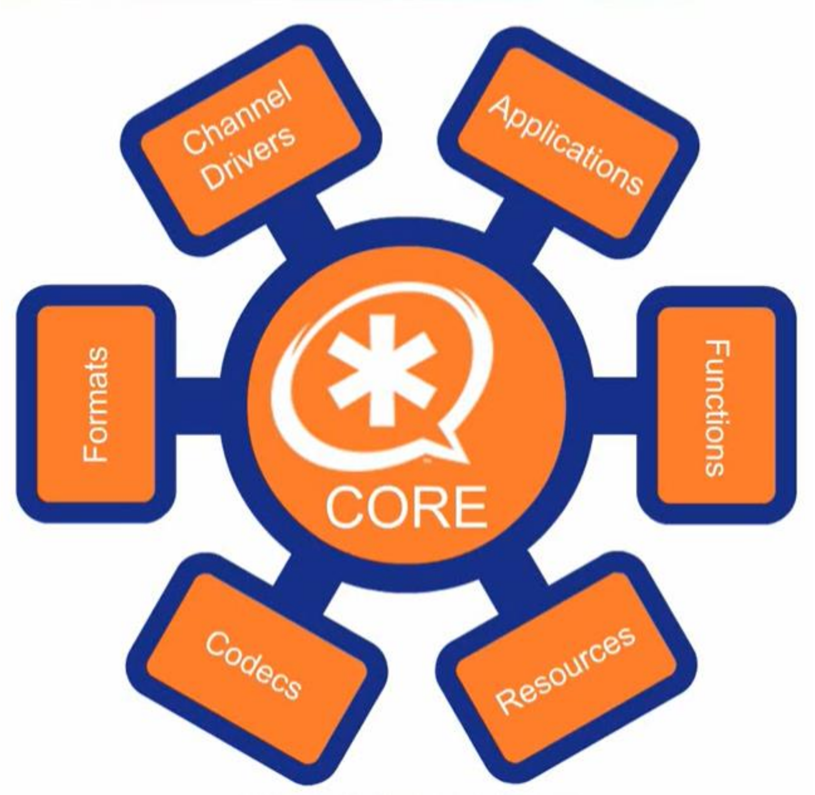
- Perform an action on or to the channel
- Dinamically loaded
- Synchronously executed (one at a time)

- App_dial.so
- App_playback.so
- App_voicemail.so
- App_<name>.so
- 100+ apps



Functions

- Improve flexibility on a dialplan (changing)
- Get or Set channel data



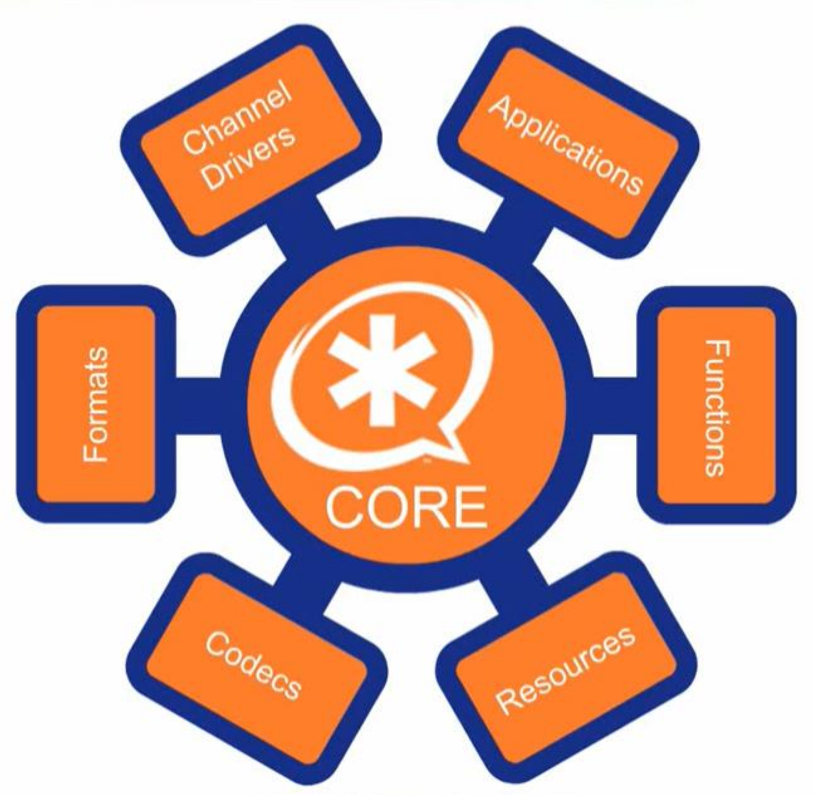
- Fun_callerid.so
- Fun_cdr.so
- fun_math.so

Teknowlogic
The Power of Knowledge

Resources

- Perform an action on or to the channel
- Statically loaded
- May operate simultaneously on multiple channels

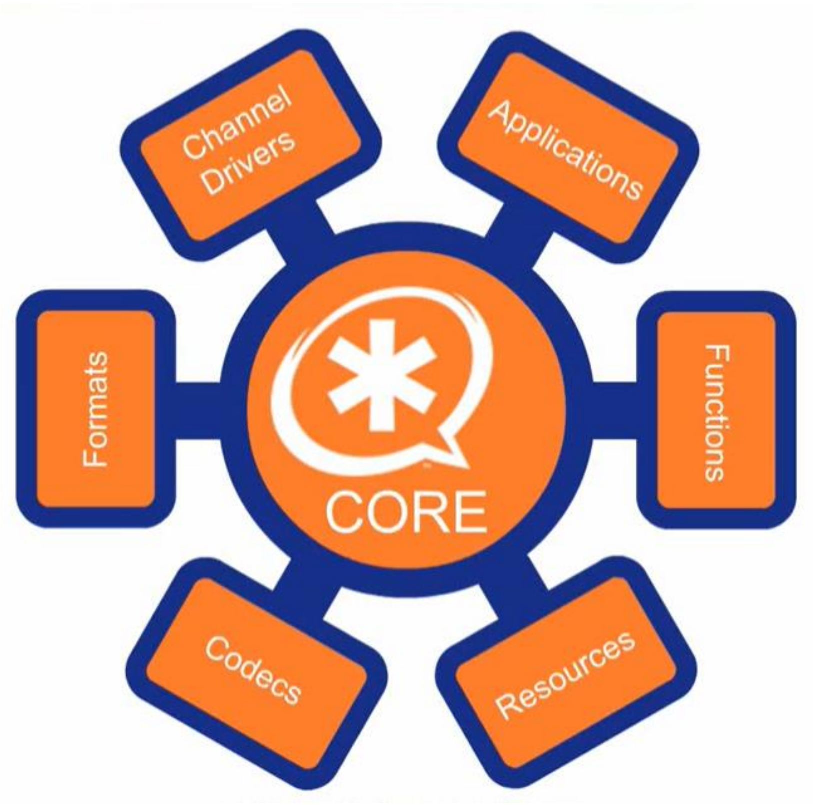
- Res_musiconhold.so
- Res_monitor.so (call recording)



Teknowlogic
The Power of Knowledge

Digital media (Audio/Video)

- CODECS: converts media original format to a needed internal format on a **channel**
- FORMATS: converts media on a **disk**



- Codec_gsm.so
- Codec_ulaw.so
- Format_jpeg.so
- Format_h264.so

Asterisk Interfaces

- Conf files
- CLI
- AMI
- AGI

Asterisk CLI

- module show like musiconhold
- Sip show peers
- Core show channels
- Reload
- help

TKL Teknowlogic
The Power of Knowledge

Asterisk AMI

- Asterisk Management Interface
- Remote administration
- Scripting call flow
- Dial/hangup calls
- Used to create
  - Auto-dialers
  - Operator panels
  - Call queue monitoring tools

Asterisk AGI

- Asterisk Gateway Interface
- Control callflow
  - PHP
  - C++
  - Java
  - Perl

Asterisk ARI

- The Asterisk RESTful Interface (ARI).
- While AMI is good at call control and
- AGI is good at allowing a remote process to execute dialplan applications,
- neither of these APIs was designed to let a developer build their own custom communications application.
- ARI is an asynchronous API that allows developers to build communications applications by exposing the raw primitive objects in Asterisk - channels, bridges, endpoints, media, etc. - through an intuitive REST interface.
- The state of the objects being controlled by the user are conveyed via JSON events over a WebSocket.

Review

- Asterisk is highly modular
- CLI allows interaction with asterisk
- AMI,AGI allow external callflow control
- ARI allows to create communication applications

Asterisk CLI

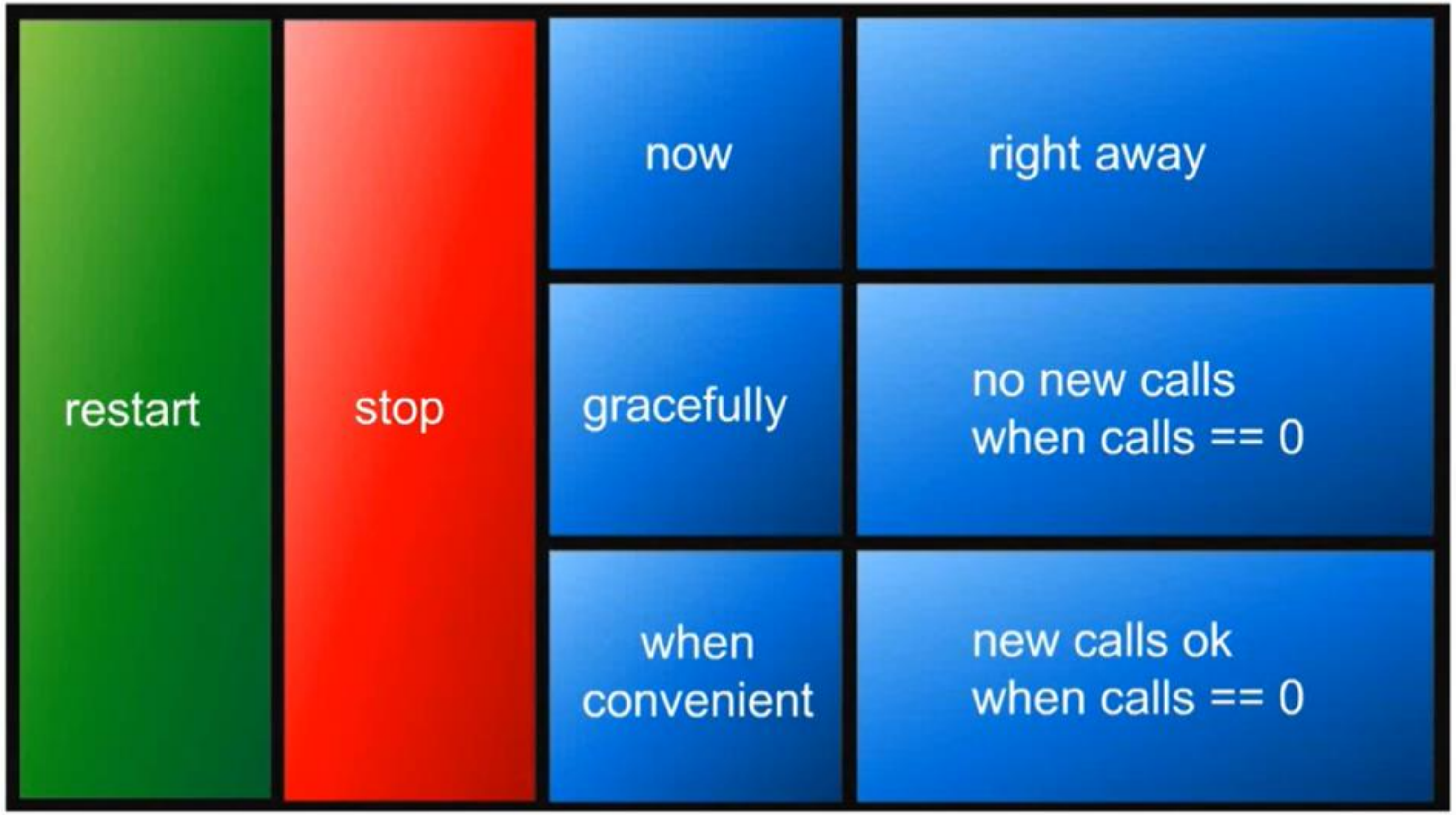
Using the CLI

- Asterisk –r
- exit
- core show <name>
    - License
    - Warranty
    - Version
    - Settings
    - Applications
    - Channels

TKL Teknowlogic
The Power of Knowledge

Restarting and stopping

core stop <when>



Verbose

- Relevant to administrators

- *CLI>core set verbose 0 - turns off verbosity
- *CLI>core set verbose 0 - only most important info
- *CLI>core set verbose 0 - prints more than 1
- *CLI>core set verbose 0 - prints more than 2

Debug

- Relevant to developers

- *CLI>core set debug 0 - turns off debugging
- *CLI>core set debug 1 - only most important info

Messages

- NOTICE: informational, doesn't indicate a problem
- WARNING: Indicates a problem
- ERROR: Indicates a severe problem may jeopardize system functionality

! (Execute a Shell command)

- !
- !<Linux command>

Dialplan reload

- Help
- Help dialplan reload

Review

- CLI is a direct connection to Asterisk
- Reload, start, stop

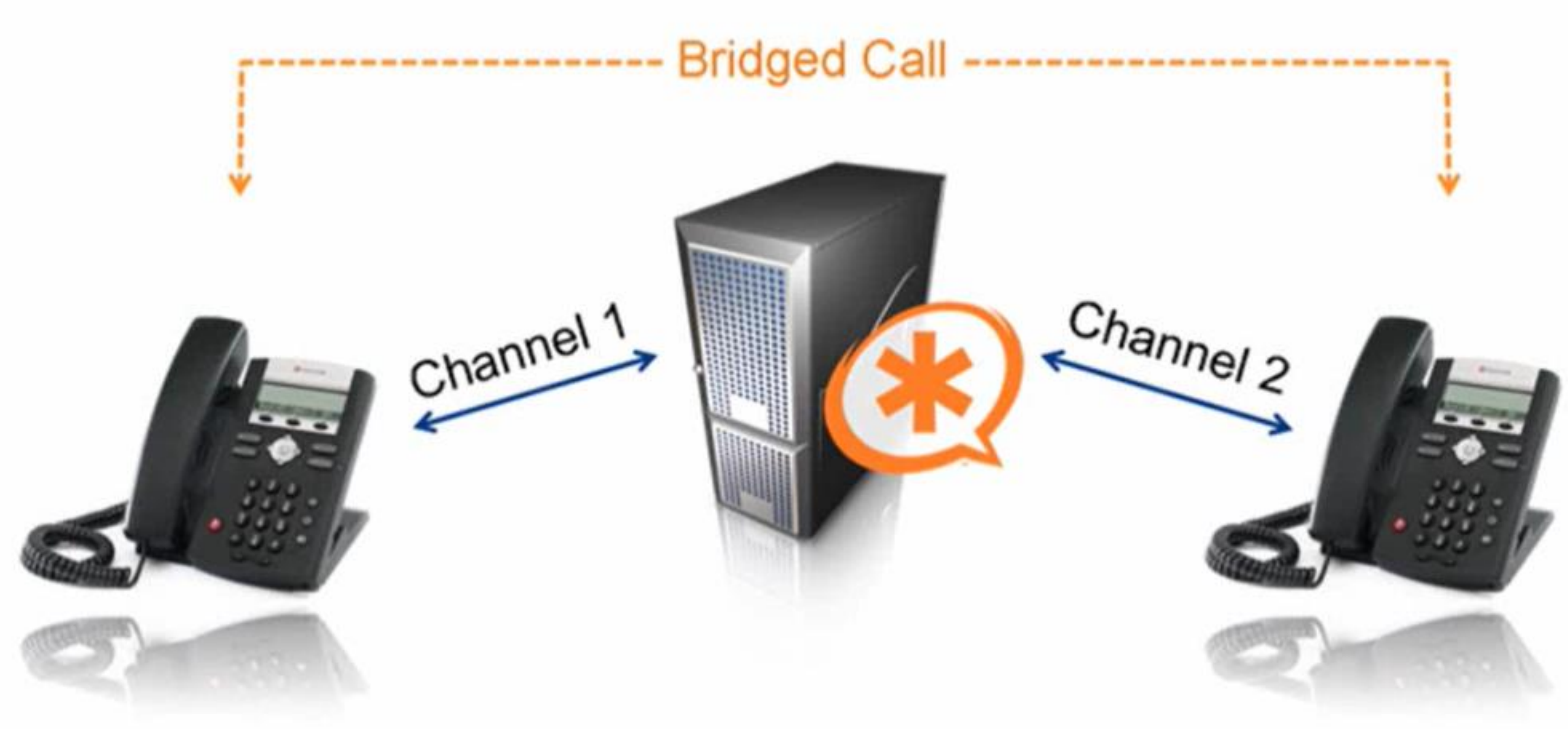
Setting up a Phone

Objectives

- Configure sip.conf
- Configure a softphone

Softphone

Call Flow Review



Sip.conf

[phone-1]

type=friend

host=Dynamic

context=internal_users

secret=XXXXX

root@asterisk:/etc/asterisk

```
[general]
context=default
allowoverlap=no
udpbindaddr=0.0.0.0
tcpenable=no
tcpbindaddr=0.0.0.0
srvlookup=yes

[authentication]

[basic-options](!)
        dtmfmode=rfc2833
        context=from-office
        type=friend
[natted-phone](!,basic-options)
        nat=yes
        directmedia=no
        host=dynamic
[public-phone](!,basic-options)
        nat=no
        directmedia=yes
[my-codecs](!)
        disallow=all
        allow=ilbc
        allow=g729
        allow=gsm
        allow=g723
        allow=ulaw
[ulaw-phone](!)
        disallow=all
        allow=ulaw
```

Account settings breakdown

[phone-1]   ; account name or username

    [phone-1Df5jA]   ;strong account name

type=friend   ; can send and receive calls

    type=user ; can only make calls

    type=peer ; can only receive calls

host=Dynamic ; Endpoint needs to register

    host=161.35.64.4

    host=houston.twilio.com

context=internal_users ;starting context in extensions.conf

secret=XXXXX ;account password

Additional settings

```
[phone-1]    ; account name or username
type=friend    ;  can send and receive calls
host=Dynamic  ; Endpoint needs to register
context=internal_users ;starting context in extensions.conf
secret=XXXXX ;account password
;callerid= Jenny <5558675309>
;mailbox=5309@default
;disallow=all
;allow=ulaw
;deny=0.0.0.0/0
;permit=192.168.55.0/24
;qualify=yes
```

Teknowlogic
The Power of Knowledge

Phone Config

User ID:

Domain: <ip>

Password:

Display name:

Timeout error

- 5060 port connection.

Error Messages & Log

- /var/log/Asterisk
- ‘messages’ file contains
  - Warning
  - Error
  - Notice

- Asterisk –rvvv
- Sip show peers

Review

- Add accounts for SIP enabled phones in sip.conf
- SIP Client configuration
- Troubleshooting

Creating your extension

Dialplan, the heart of Asterisk

```
[ Context 'internal_users' created by 'pbx_config' ]
  '6000' =>           1. Answer()                    [pbx_config]
                      2. Playback(hello-world)       [pbx_config]
                      3. Hangup()                    [pbx_config]
  '6001' =>           1. Dial(SIP/phone-1,20)        [pbx_config]
  '6002' =>           1. Dial(SIP/phone-2,20)        [pbx_config]

-= 3 extensions (5 priorities) in 1 context. =-
```

extensions.conf

[general] ;section

[globals] ;section

[context1] ;organization units

[context2] ;organization units

[context3] ;organization units

[internal_users] ; end-point -> application,extension

Teknowlogic
The Power of Knowledge

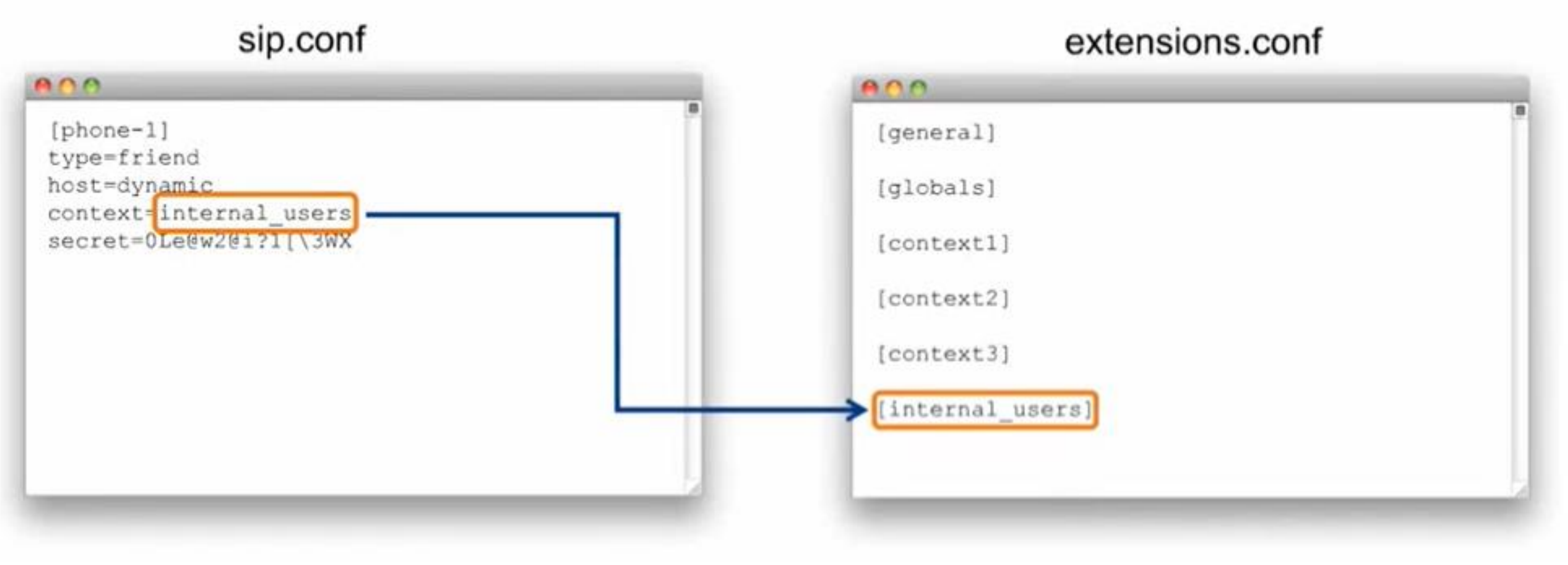
Extension syntax

```
[internal_users]
;EXTEN => EXTENSION,PRIORITY, APPLICATION()
;                 number,      number,  application_name( parameters)

exten => 8500,1,Voicemail()
; 8500 is the extension number
; priority 1 is the first action taken
; Voicemail is the application to be executed when 8500 is dialed

;core show application voicemail
```

Context relationship



Answering a call

```
[internal_users]
;EXTEN => EXTENSION,PRIORITY, APPLICATION()

exten => 6000,1,Answer()                    ;first action
exten => 6000,2,Playback(hello-world)  ;plays file to the channel
exten => 6000,3,Hangup()                    ;channel hangup

;dialplan reload
```

Dial 6000

Dialing a phone

```
[internal_users]
;EXTEN => EXTENSION,PRIORITY, APPLICATION()

exten => 6000,1,Answer()
exten => 6000,2,Playback(hello-world)
exten => 6000,3,Hangup()

exten => 6002,1,Dial(SIP/phone-2,20)
```

Add a phone in sip.conf

```
[phone-1]    ; account name or username
type=friend    ;  can send and receive calls
host=Dynamic  ; Endpoint needs to register
context=internal_users ;starting context in extensions.conf
secret=XXXXX ;account password
[phone-2]
type=friend
host=Dynamic
context=internal_users
secret=YYYYYY
```

TKL Teknowlogic
The Power of Knowledge

Review

- Dialplan
- Sip.conf ⇔ extension.conf
- Syntax
  - exten =>extension, priority, application()
- Answer()
- Dial()

Call breakdown

Extension

- exten => 6002,1,Dial(Technology/Resource)

- Sip.conf ⇔ extension.conf
- Syntax
  - exten =>extension, priority, application()
- Answer()
- Dial()

extensions.conf

```
[internal_users]
Exten => 6000,1,Answer()
Exten => 6000,2,Playback(hello-world)
Exten =>6000,3,Hangup()


exten => 6002,1,Dial(SIP/phone-2,20)
```

TKL Teknowlogic
The Power of Knowledge

extensions.conf

exten => 6002,1,Dial(SIP/phone-2,20)

; (6002 is the EXTENSION)

exten => 1,

exten => 123,

exten => 1234,

Exten => 12345678901234567890,

sip.conf

```
[phone-2]
type=friend
host=Dynamic
context=internal_users
secret=YYYYYY
```

Syntax

exten => 6002,1,Dial(Technology/Resource)

(EXTENSION, PRIORITY, APPLICATION)

;Dial attempts to connect to another endpoint

;Technology must be valid channel driver: SIP, IAX2, DAHDI

;Resource must be the name of a phone or a trunk identified in the config file (sip.conf

exten => 6002,1,Dial(SIP/phone-2,20)

Syntax

exten => 6002,1,Dial(Technology/Resource)

(EXTENSION, PRIORITY, APPLICATION)

;Dial attempts to connect to another endpoint

;Technology must be valid channel driver: SIP, IAX2, DAHDI

;Resource must be the name of a phone or a trunk identified in the config file (sip.conf

exten => 6002,1,Dial(SIP/phone-2,20)

(20 is the TIMEOUT)

Review

Call setup

Dialplan configuration

Asterisk CLI

Diaplan overview

Dialplan

In traditional PBX is a list of relations between numbers and end-points.

The Asterisk dialplan is far more flexible and powerful.

Scripting Language capable of conditional logic and complex commands.

Can write to databases or Access webpages.

Configured in extensions.conf

Contexts

Top level organization unit within a dialplan

A dialplan contains several different contexts

A context contains several extensions

Extensions

An extension is a named list of several actions that Asterisk Will perform when that extension is dialed.

A context is a container for the extensions.

The same extension can exists in several contexts. So you must specify the context.

Extensions and priorities

Priorities:

1.Action (Application)

2.Action

3.Action

Example:

1. Dial Phone

2. Record Voicemail

Visual breakdown

Dialplan show 6002@internal_users

```
root@asterisk:~ — ssh — 78x29
asterisk*CLI>
asterisk*CLI> dialplan show 6002@internal_users
[ Context 'internal_users' created by 'pbx_config' ]
  '6002' =>         1. Dial(SIP/phone-2,20)      [pbx_config]
```

Teknowlogic
The Power of Knowledge

Call message in CLI

```
asterisk*CLI>
asterisk*CLI> dialplan show 6002@internal_users
[ Context 'internal_users' created by 'pbx_config' ]
  '6002' =>          1. Dial(SIP/phone-2,20)      [pbx_config]

asterisk*CLI>
 == Using SIP RTP CoS mark 5
    -- Executing [6002@internal_users:1] Dial("SIP/phone-1-00000009",
"SIP/phone-2,20") in new stack
 == Using SIP RTP CoS mark 5
    -- Called phone2
    -- SIP/phone2-0000000a is ringing
 -- SIP/phone2-0000000c answered SIP/phone1-0000000b
    -- Native bridging SIP/phone1-0000000b and SIP/phone2-0000000c
```

TKL Teknowlogic
The Power of Knowledge

Core show channels

```
asterisk*CLI> core show channels
Channel                    Location                   State    Application(Data)
SIP/phone2-0000000e   (None)                     Up       AppDial((Outgoing
Line))
SIP/phone1-0000000d   6002@internal_users: Up       Dial(SIP/phone2,20)
2 active channels
1 active call
2 calls processed
```

Review dialplan

```
asterisk*CLI>dialplan reload
asterisk*CLI>dialplan show
[ Context 'internal_users' created by 'pbx_config' ]
  '6000' =>          1. Answer()                          [pbx_config]
                     2. Playback(hello-world)             [pbx_config]
                     3. Hangup()                          [pbx_config]
  '6001' =>          1. Dial(SIP/phone-1,20)              [pbx_config]
  '6002' =>          1. Dial(SIP/phone-2,20)              [pbx_config]

-= 3 extensions (5 priorities) in 1 context. =-
```

extensions.conf

Other Ways to Modify Dialplan:

- Applications can load extensions
- CLI Commands
  - add extensions
  - remove extensions
  - update extensions

Review

Contexts contains extensions

Extensions are made up of priorities

Priorities are the numbered actions for each extension

Applications are the actions taken on a call

```
[context]
exten => 6000,1,Application
```

Retrospectiva – De-brief

QUÉ HICIMOS BIEN!

QUÉ PODEMOS MEJORAR

Thank you!

TKL Teknowlogic
The Power of Knowledge