OpenR2 in Asterisk

Moisés Silva

Software Developer

Sangoma Technologies

Outline

What is it?

- MFC/R2 or just R2 for its friends.
- Old days telephony signaling for trunks.
- Analog and digital versions.
- Digital version defined by ITU Q.421.
- Lots of variants around the world.

Where is being used?

- Old, but widely used.
- Heavily used in Central and South America (México, Colombia, Argentina, Brazil etc).
- Asia (China and Philippines are major users).
- Sometimes cheaper than ISDN PRI lines.
- R2 is not going anywhere anytime soon (infraestructure already in place).

How does the digital version work?

- Uses E1 facilities.
- CAS for line supervision signaling (Answer, Hangup etc).
- Inband MF tones for register signaling (ANI, DNIS etc).
- Call setup much slower than PRI (2+ seconds)

What's the deal with variants?

- R2 has a scary number of variants.

What's the deal with variants?

- R2 has a scary number of variants.
- Position of CD bits may be different.

What's the deal with variants?

- R2 has a scary number of variants.
- Position of CD bits may be different.
- MF tones mean different things in different countries in different contexts.

What's the deal with variants?

- R2 has a scary number of variants.
- Position of CD bits may be different.
- MF tones mean different things in different countries in different contexts.
- Some ask DNIS before ANI (and fail when done the other way around).

What's the deal with variants?

- R2 has a scary number of variants.
- Position of CD bits may be different.
- MF tones mean different things in different countries in different contexts.
- Some ask DNIS before ANI (and fail when done the other way around).
- Some even use DTMF instead of MF.

What's the deal with variants?

- R2 has a scary number of variants.
- Position of CD bits may be different.
- MF tones mean different things in different countries in different contexts.
- Some ask DNIS before ANI (and fail when done the other way around).
- Some even use DTMF instead of MF.
- Different methods to block collect calls.

What's the deal with variants?

- R2 has a scary number of variants.
- Position of CD bits may be different.
- MF tones mean different things in different countries in different contexts.
- Some ask DNIS before ANI (and fail when done the other way around).
- Some even use DTMF instead of MF.
- Different methods to block collect calls.
- Different way to notify end of DNIS (MF tone vs timeout).

What's the deal with variants?

- R2 has a scary number of variants.
- Position of CD bits may be different.
- MF tones mean different things in different countries in different contexts.
- Some ask DNIS before ANI (and fail when done the other way around).
- Some even use DTMF instead of MF.
- Different methods to block collect calls.
- Different way to notify end of DNIS (MF tone vs timeout).
- Some countries do not even regulate its variant.

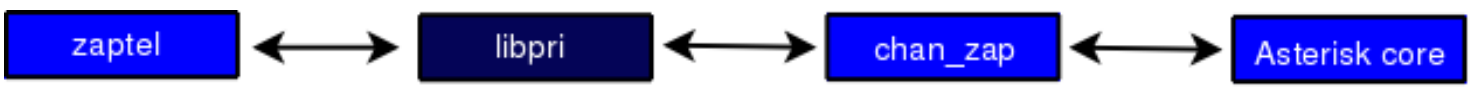
Outline

Asterisk approach for PSTN signaling.

- One channel driver to rule them all.
- Most signaling implemented right into zaptel/chan_zap (now DAHDI/chan_dahdi).
- Some with the help of protocol libraries (libpri, libss7).

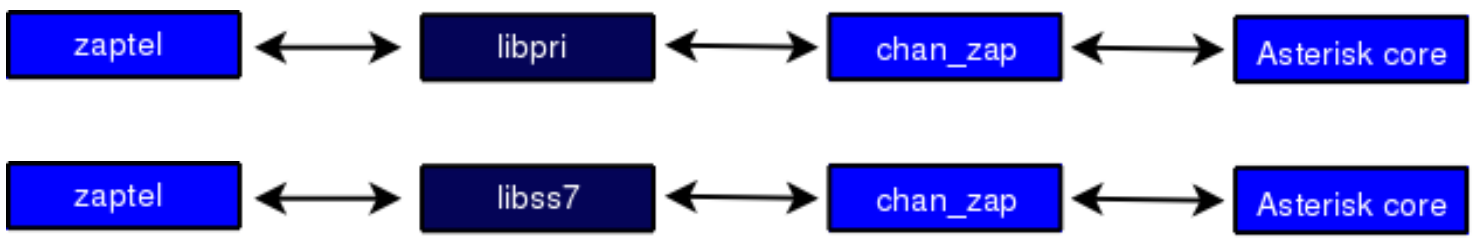
Asterisk approach for PSTN signaling.

- One channel driver to rule them all.
- Most signaling implemented right into zaptel/chan_zap (now DAHDI/chan_dahdi).
- Some with the help of protocol libraries (libpri, libss7).



Asterisk approach for PSTN signaling.

- One channel driver to rule them all.
- Most signaling implemented right into zaptel/chan_zap (now DAHDI/chan_dahdi).
- Some with the help of protocol libraries (libpri, libss7).

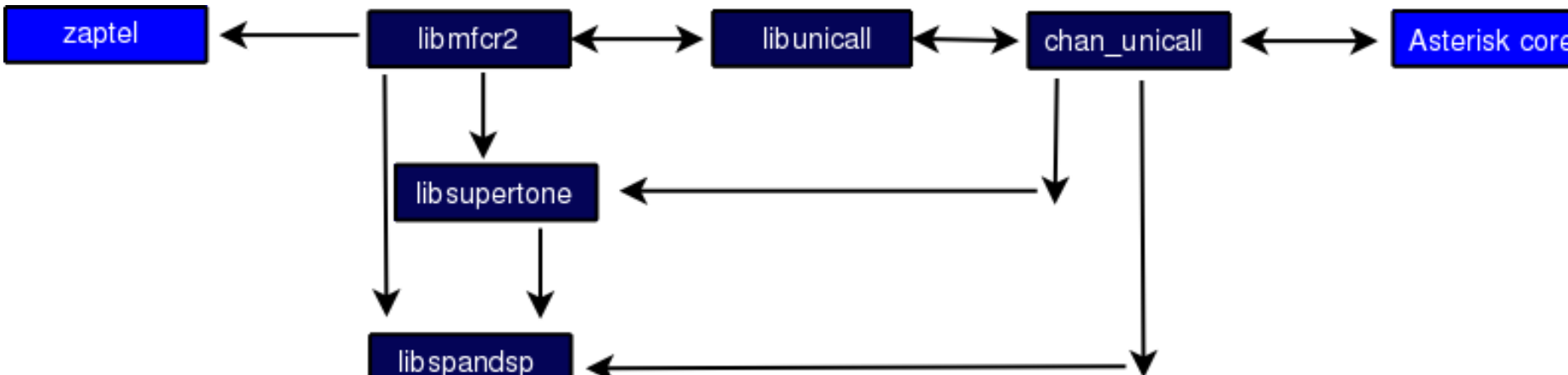


Unicall R2 architecture.

- A different approach.
- Unicall is an abstraction layer in the form of a simple library.
- All signaling details are hidden in protocol modules.
- The channel driver (chan_unicall), ideally, interacts with a single signaling interface (libunicall).
- Ideally, adding a new signaling protocol requires no changes to the Asterisk channel driver.
- libmfcr2 supports a great number of R2 variants.

Unicall R2 architecture.

- A different approach.
- Unicall is an abstraction layer in the form of a simple library.
- All signaling details are hidden in protocol modules.
- The channel driver (chan_unicall), ideally, interacts with a single signaling interface (libunicall).
- Ideally, adding a new signaling protocol requires no changes to the Asterisk channel driver.
- libmfcr2 supports a great number of R2 variants.



Unicall issues. Sad but true.

- Versioning hell.
- Multiple components, multiple user errors.
- GPL license can be sometimes a show-stopper (solved recently).
- You need a whole new driver (patching) and libraries just to get R2 working.
- Poor support.
- Even open source projects sometimes are not open enough.

Digivoice R2.

- R2 implementation included in Digivoice library Voicerlib.
- Support for 3 variants (Brazil, Argentina, México).
- Just works with Digivoice cards.
- Needs its own Asterisk channel driver.
- Echo issues in their boards reported by users.
- Portuguese support is a plus for Brazilians.

Khomp R2.

- I could not get the source code.
- Just works with Khomp boards.
- Needs its own Asterisk channel driver.
- Nice GUI for configuration.
- Portuguese support is a plus for Brazilians.

PIKA R2.

- Recent support on their GrandPrix suite.
- Only supports 3 variants (Brazil, Argentina, México).
- No 64 bit support.
- Just works with PIKA boards.
- Needs its own Asterisk channel driver.

Outline

Why another R2 solution?

Why another R2 solution?

- All previous solutions are not quite generic.

Why another R2 solution?

- All previous solutions are not quite generic.
- All previous solutions do not work with FreeSwitch at all.

Why another R2 solution?

- All previous solutions are not quite generic.
- All previous solutions do not work with FreeSwitch at all.
- All previous solutions are not integrated seamlessly into Asterisk.

Why another R2 solution?

- All previous solutions are not quite generic.
- All previous solutions do not work with FreeSwitch at all.
- All previous solutions are not integrated seamlessly into Asterisk.
- All previous solutions have not a really open development.

Why another R2 solution?

- All previous solutions are not quite generic.
- All previous solutions do not work with FreeSwitch at all.
- All previous solutions are not integrated seamlessly into Asterisk.
- All previous solutions have not a really open development.
- All previous solutions are not what I wanted.

Design goals.

Design goals.

- Written from scratch.

Design goals.

- Written from scratch.
- Component Independent.

Design goals.

- Written from scratch.
- Component Independent.
- Interoperable.

Design goals.

- Written from scratch.
- Component Independent.
- Interoperable.
- Flexible. Easily integrated to other projects.

Design goals.

- Written from scratch.
- Component Independent.
- Interoperable.
- Flexible. Easily integrated to other projects.
- Really open development and user community.

Development.

- Development started in December 2007.
- Sangoma started sponsoring the project in March 2008.
- Mark spencer himself added some code (which now I got rid of, but oh well)
- Several Brazilian developers have contributed code.
- Heavy testing has been done with success in different scenarios.

The library. Small and functional.

- LGPL license.
- LGPL SpanDSP detector embedded into OpenR2.
- r2test program provided to test R2 links.
- No other libraries required (other than libc and libpthread).

How flexible and stable really is?

- Works with Zaptel 1.2, 1.4 and DAHDI.
- Patches for Asterisk 1.2, 1.4 and 1.6
- SVN Branches for Asterisk 1.2, 1.4 and 1.6
- RPM and Debian packages available.
- Plan to merge with trunk someday (Digium decides) 1.6.3??
- Feedback welcomed!

Where to get it?

- Main site is at www.libopenr2.org
- OpenR2 Code is downloaded from http://code.google.com/p/openr2/
- Asterisk code available from 'moy' SVN team branch: http://svn.digium.com/svn/asterisk/team/moy
- Browse it at http://svn.digium.com/view/asterisk/team/moy

What about support?

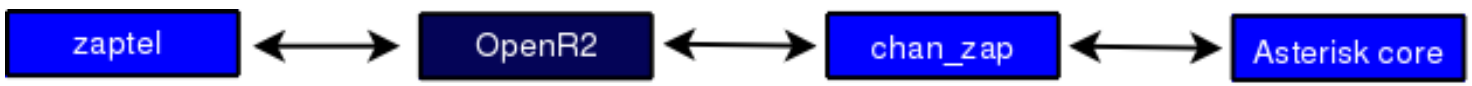
- Sangoma gives openr2 support to their customers.
- Community based mailing list in http://lists.digium.com/mailman/listinfo/asterisk-r2
- Community based IRC channel in irc.freenode.net at #openr2

How does it fit into Asterisk?

- OpenR2 fits the same way libpri and libss7 fit into chan_zap or chan_dahdi.
- The same library Works with Asterisk 1.2, 1.4 and 1.6, no tricks.
- The chan_zap or chan_dahdi support for R2 is only compiled if OpenR2 is present.

How does it fit into Asterisk?

- OpenR2 fits the same way libpri and libss7 fit into chan_zap or chan_dahdi.
- The same library Works with Asterisk 1.2, 1.4 and 1.6, no tricks.
- The chan_zap or chan_dahdi support for R2 is only compiled if OpenR2 is present.



Outline

OpenR2 variants implemented so far.

- ITU
- Argentina
- Brasil
- China
- Colombia
- Czech
- Ecuador
- México
- Philippines
- Venezuela

More fun with options.

More fun with options.

- Specify call category.

More fun with options.

- Specify call category.
- You can save per-call protocol logs at different levels.

More fun with options.

- Specify call category.
- You can save per-call protocol logs at different levels.
- Max ANI, Max DNIS.

More fun with options.

- Specify call category.
- You can save per-call protocol logs at different levels.
- Max ANI, Max DNIS.
- Block collect calls with double answer or MF tone.

More fun with options.

- Specify call category.
- You can save per-call protocol logs at different levels.
- Max ANI, Max DNIS.
- Block collect calls with double answer or MF tone.
- Forced release.

More fun with options.

- Specify call category.
- You can save per-call protocol logs at different levels.
- Max ANI, Max DNIS.
- Block collect calls with double answer or MF tone.
- Forced release.
- Immediate accept.

More fun with options.

- Specify call category.
- You can save per-call protocol logs at different levels.
- Max ANI, Max DNIS.
- Block collect calls with double answer or MF tone.
- Forced release.
- Immediate accept.
- Get ANI first.

More fun with options.

- Specify call category.
- You can save per-call protocol logs at different levels.
- Max ANI, Max DNIS.
- Block collect calls with double answer or MF tone.
- Forced release.
- Immediate accept.
- Get ANI first.
- Advanced protocol configuration file.

More coming...

- Windows.
- OpenZAP.
- FreeSwitch.
- DTMF/R2.
- DiscOS.
- Trixbox (Already announced by Fonality).
- GUI configuration tools.
- More R2 variants.

Outline

Give OpenR2 a try

- The time for MFC/R2 to stop being a problem in Asterisk is coming.
- People has used OpenR2 in production with success.
- I will implement any missing feature you may suggest.
- I will fix any bug you find.
- Release candidate 2 is out for downloading.

Where to find more information?

- http://www.libopenr2.org/ (Don't mistake it with openr2.org)
- http://code.google.com/p/openr2/
- http://lists.digium.com/mailman/listinfo/asterisk-r2
- http://svn.digium.com/view/asterisk/team/moy/
- http://blog.alexandrealencar.net/ (Portuguese information)

Questions? Drop me a line. Thanks!

- Blog: http://www.moythreads.com/
- Google Talk && MSN: moises.silva@gmail.com
- E-mail: moy@sangoma.com
- Notice I am not from Brazil, even though I can read Portuguese, try using Spanish or English please :)