Asterisk and the calendars

When non-C developers meet Asterisk+libical
Who I am?

- Creator of API-Hour (Daemon framework for Python-AsyncIO)
- Creator of aiosip (used by Sangoma to test their phones)
- Co-maintainer of Panoramisk (Asterisk binding for AsyncIO)
- Small contributor in several AsyncIO libraries (aiohttp…)
- Interested by benchmarks to find the bottlenecks.
- Contributor of [https://www.techempower.com/benchmarks/](https://www.techempower.com/benchmarks/)
1. Most simple as possible ("prêt-à-porter")
2. Distributed telephony and collaboration
3. Efficiency is the first class citizen
   (1500+ simultaneous calls by server)

**EYEPEA**

1. Full-monty customized solutions ("haute couture")
2. Solutions mainly based on Wazo (AAA solution)
3. Historical business of the company
Some of our clients
Customer needs

1. Open/close schedule
2. Personal calendar
3. Oncall schedule
4. Google Calendar/Office365 integration
In ALLOcloud, step 1: Define a calendar
In ALLOcloud, step 2: Put the calendar in the callflow
icalendar: the most obvious format

1. Used in lot of products
2. Stable standard
3. Very old
   = More chances to have good implementations
Old = Stable ?

SO WRONG
First release: icalendar in pure Python

1. Easier for us debug/integrate in our workflow

2. Libical integration in Asterisk looked like a PoC during an Astricon:
   a. Very few returns from sysadmins on Internet
   b. Lack of examples
   c. Need to dig in the original Astricon presentation to understand how to use dialplan functions
   d. Afraid by Asterisk crashes

Gentle remark: we are not C experts ;-)
ANATOMY OF A FIRST-TURN CRASH

IT WAS THE MOTHER OF ALL FIRST-TURN CRASHES. Decimating the median of all photogenic motocross riders left the standing remains eerie for the first-order event in the new
Reed/turner Craziest Stadium at Wrigley Field. Reed, in his all-time,
rescuing loup, was one of the best all-time
heroes, all three quarters of the all-time
stars and their motorcycles were suddenly
shattered on the ice, with the survivors at
the race, some in the crowd, and their
cars with mirror parts, lights through the crowd, and
the spectra were broken and silent forever.

The sprawling dirt-bike version of demolition derby
happened so fast that it was difficult for anyone to figure out
exactly what happened on the photo-realistic crash. It was
such was the track at the time, riders and
people on the field in photographs—like if riders Jeff
Lynn and Bryan Pierson, WRAV's Dave Olderson, the
crew from Honda World Corp., and our cameraman David—we
were able to see a group of riders that what it all from
the largest single group crash in the photo-realistic history
of AMA Supercross.
Main challenge: recurrency

Most libraries parse recurrency fields

BUT

Most don’t interpret correctly recurrency data
BEGIN:VEVENT
DTSTART;TZID=Europe/Brussels:20151221T090000
DTEND;TZID=Europe/Brussels:20151221T180000
RRULE:FREQ=WEEKLY;BYDAY=MO
DTSTAMP:20161108T165938Z
UID:gig9ashd5abvg669k3tq1t6gqo@google.com
CREATED:20151227T185112Z
DESCRIPTION:
LAST-MODIFIED:20160112T225808Z
LOCATION:
SEQUENCE:0
STATUS:CONFIRMED
SUMMARY:Working hours
TRANSP:OPAQUE
END:VEVENT
icalendar recurrence: technical

“RRULE”, "RDATE", "EXDATE", “RECURRING-ID” and "EXRULE” fields

**Example:**
DTSTART;TZID=US-Eastern:19970902T090000
RRULE:FREQ=DAILY;COUNT=10
open-source icalendar libraries comparison

Tested all Python, Ruby, Perl... libraries

Long story short: libical rules
B-plan: libical integration in Asterisk

1. Client mood:

2. Put res_calendar.so on production

3. Huge success after half a day of work

Thanks a lot Asterisk developers :-)
BUT
Winter is coming...

DST nightmare
libical forks and libical3

1. Bugs with timezone and DST

2. all already fixed in libical3

3. But, libical1 and libical forks in sunbird, evolution... distributed in Debian Jessie
1. Uninstall libical

2. Clone and build from master branch

3. And... it works !
1. A last corner case still exists when you edit a recurring event: recurrence-id

2. We submitted a fix on Gerrit: ASTERISK-27296

3. Thanks Benoît for the fix ;-)
res_calendar is production ready

1. For now, we have 973 calendars on production

2. A file system support is in the pipe, but not yet ready (memory leaks)

3. Maybe a Python binding of libical

4. Thanks again Asterisk developers ;-)
Questions?

@GMLudo: Twitter