

Osmocom

Open Source Mobile Communications

OpenCellular Workshop, September 11-12, 2018
at the iHub, Nairobi, Kenya

Harald Welte <hwelte@sysmocom.de>
Osmocom founder, lead developer + sysmocom CEO

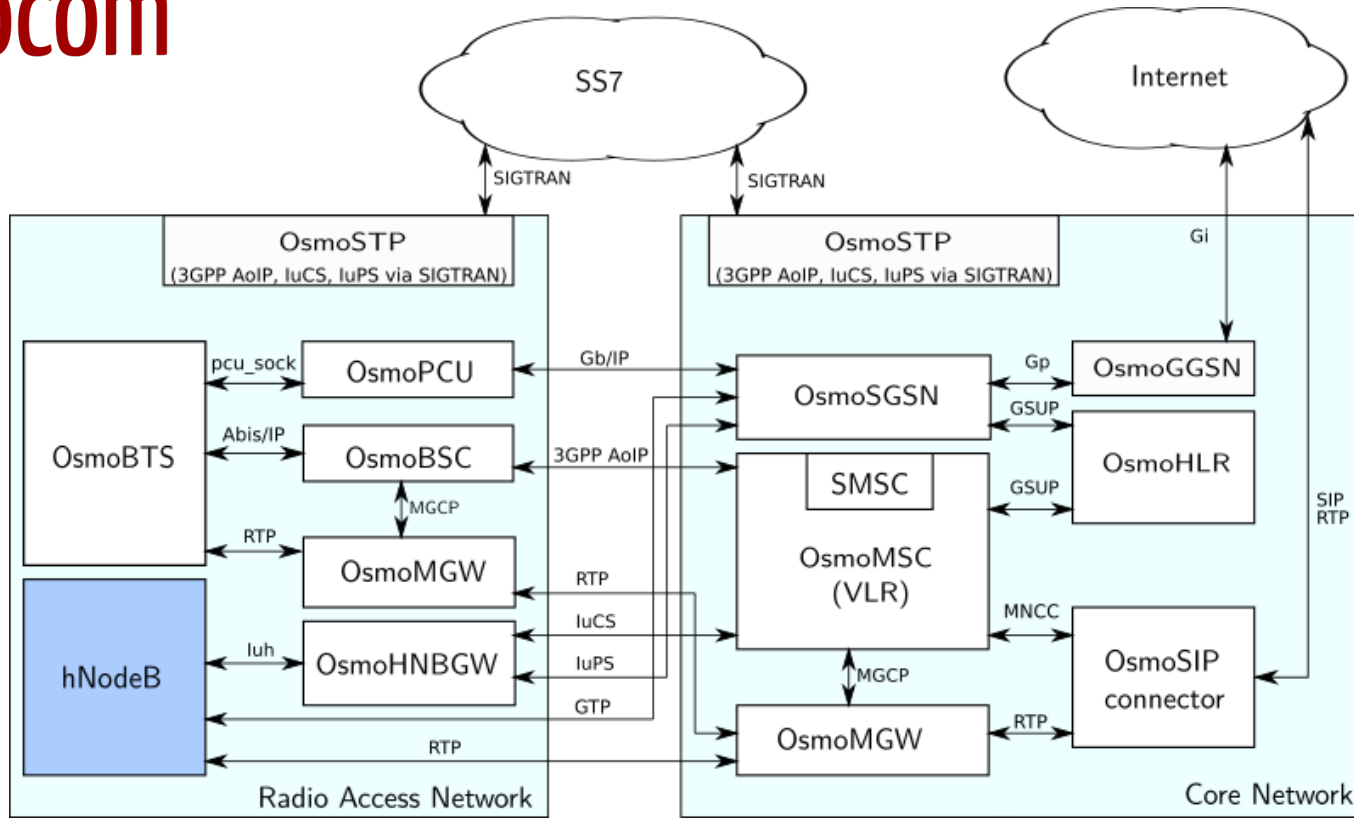
- Free and Open Source Software
- The Osmocom project (and sysmocom's role)
- How to use Osmocom 2G Cellular Network Infrastructure
 - as RAN, as CN, within OpenCellular
- Osmocom Conference 2018
- Developments in 2017/2018
- How to Interact with Osmocom

- Many successful Free / Open Source (FOSS) projects
 - Operating Systems (Linux, FreeBSD, OpenBSD)
 - Anything Internet/Web related: Apache, nginx, lighttpd, Firefox
 - Smartphones: Android (at least the Open Source portions of it)
- Collaborative, Open development project: Anyone can join, no fees/contracts/membership
 - shared investment in R&D, while everyone can use full results
 - not about a one-way producer/consumer relationship
 - sustainable FOSS projects require responsible commitment from all stake holders
 - the software needs to be written, tested and maintained, after all

- Osmocom: **O**pen **S**ource **M**obile **C**OMmunications
- Bringing benefits of Free / Open Source (FOSS) development model to Mobile Communications
 - remove reliance on expensive, proprietary black-box equipment
 - you don't have to be Ericsson anymore to study, experiment, innovate and improve
- Started 2008 with “whatever needed to bring a [then Siemens] GSM BTS into operation”
 - first called bs11_abis, later bsc_hack, then OpenBSC, OsmoBSC/OsmoNITB
 - developing one network element at a time: BTS, BSC, PCU, MSC, SGSN, GGSN, ...
- Pure Software-defined implementation of 2G. No dependency on proprietary hardware.
 - Can be fully virtualized / containerized

- Osmocom is home to many Open Source projects related to mobile communications
- Cellular Network Infrastructure for 3GPP technologies is only one part of this
- Other Projects include
 - TETRA, GMR/Thuraya, DECT, P25, SDR, SIMtrace
 - about 70 member projects in <https://osmocom.org/> project list
 - about 180 git repositories with source code on <https://git.osmocom.org/>
- In context of this workshop, we focus on Cellular Network Projects for GSM, GPRS, EDGE, UMTS, ...

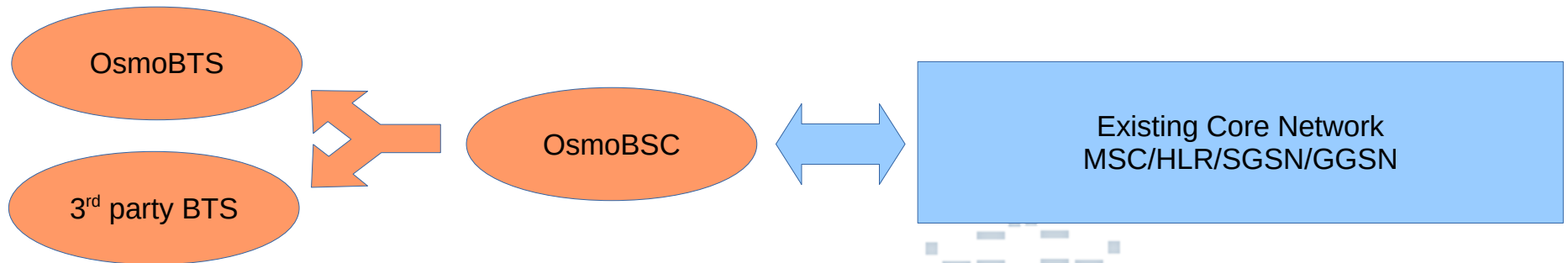
- **OsmoBTS**: GSM Base Transceiver Station, supports wide range of hardware
 - **OsmoBSC**: GSM Base Station Controller, supports many BTSs
 - not just OsmoBTS, but also Ericsson, Siemens, Nokia, etc.
 - **OsmoMSC**: GSM Mobile Switching Center with AoIP and IuCS interface
 - **OsmoHLR**: GSM Home Location Register to run autonomous/small GSM networks
 - **OsmoSIPconnector**: Interface OsmoMSC with the SIP World
 - **OsmoSGSN**: Serving GPRS Support Node for 2G and 3G with Gb, IuPS and Gp interface
 - **OpenGGSN**: Gateway GPRS Support Node for 2G and 3G with Gp and Gi interface
- (many other special-purpose projects not listed here)

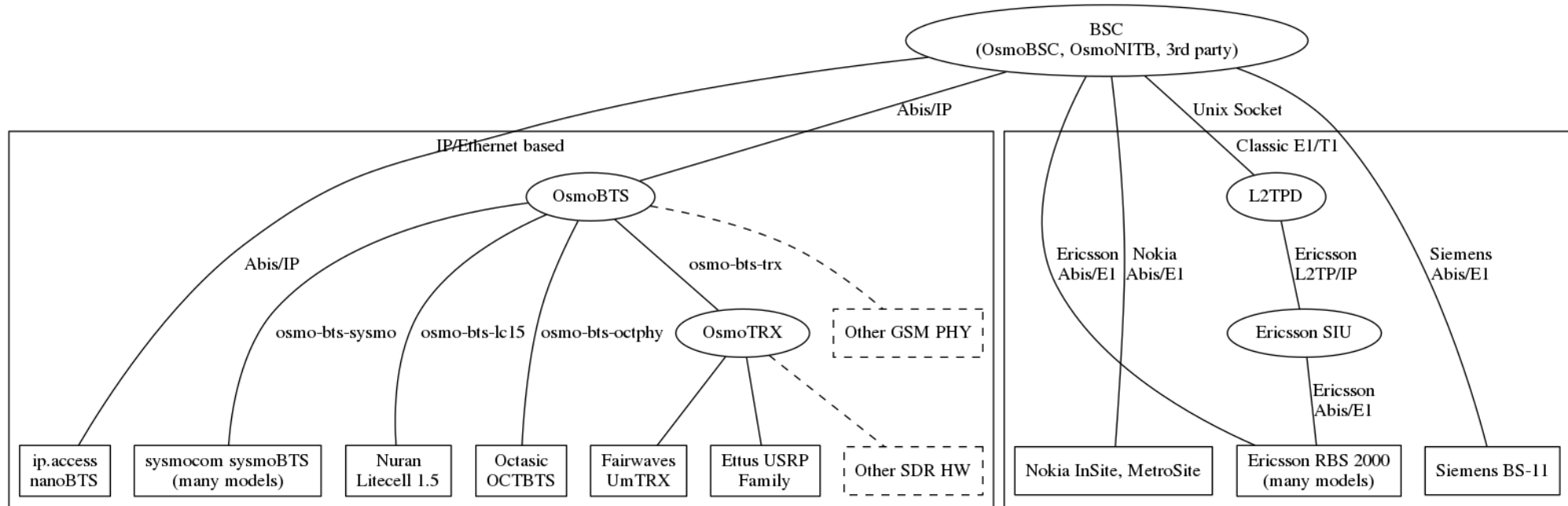


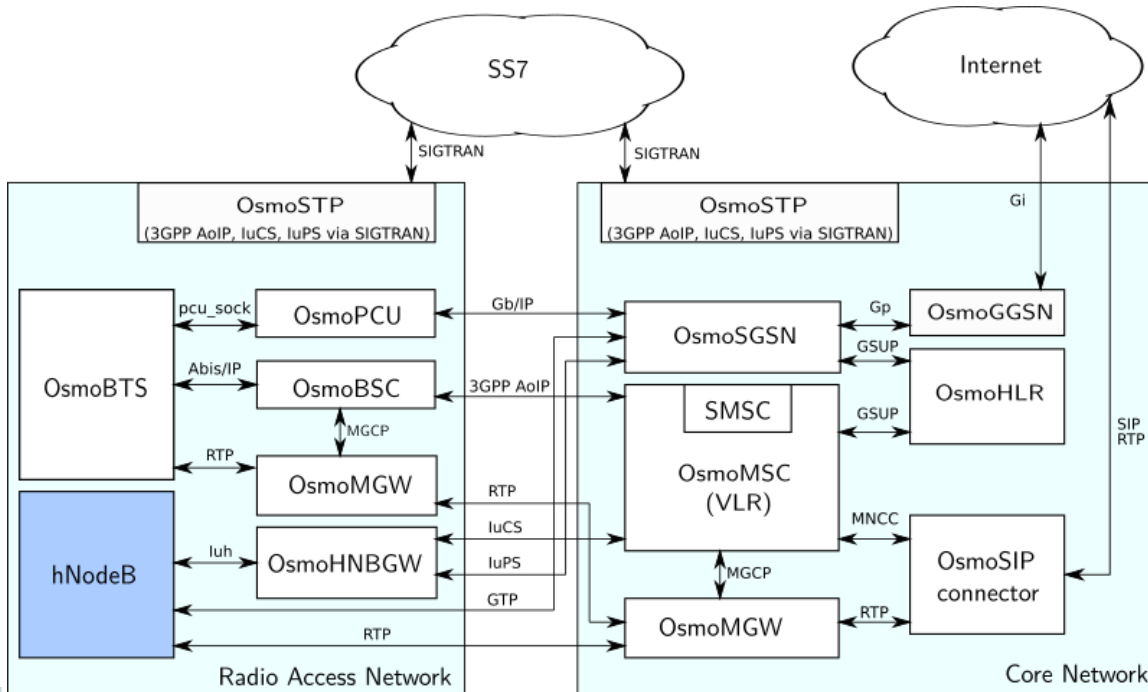
How can you use Osmocom?

- Recycle decommissioned classic BTS equipment
 - with OsmoBSC attached to classic operator core
 - with OsmoMSC/HLR to run autonomous cellular network
- With variety of OsmoBTS based hardware options (e.g. sysmoBTS, OC-SDR, OC-2G)
 - with OsmoBSC attached to classic operator core
 - with OsmoMSC/HLR to run autonomous cellular network
- With OpenCellular hardware + Cellular Community Manager

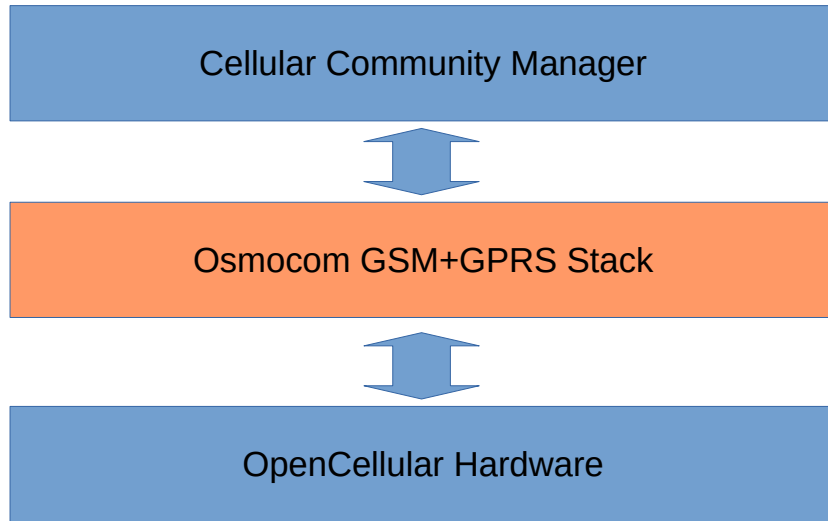
- OsmoBTS, OsmoPCU and OsmoBSC to implement GSM/GPRS/EGPRS RAN
- Interface with existing core network (MSC + SGSN) via A and Gb interface
- Open Source BSC allows to co-locate one (software) BSC per BTS
- osMUX protocol can be used for satellite back-haul optimization
- Perfect match for low-cost rural RAN in low-ARPU regions







- OsmoMSC includes MSC, VLR, SMSC
- OsmoHLR contains subscriber database / keys
- Use it with OsmoBTS/BSC to obtain a fully autonomous 2G network
- Target user is **not** the classic cellular operator
- Applications include
 - private GSM networks (farms, mining, research)
 - autonomous rural networks with or without PSTN interconnect
 - handset testing, M2M/IoT device testing



- Osmocom Stack implements actual GSM/GPRS protocol stack and functional elements like TRX, BTS, BSC, MSC, SGSN, GGSN
- Osmocom Stack is managed by Cellular Community Manager for subscriber management, billing, monitoring
- If you deploy OpenCellular for 2G, you are deploying an Osmocom GSM network!

- symocom contributes > 80% of Osmocom Cellular Infrastructure development
- has put several million € worth of development effort into the Osmocom project
- Osmocom is FOSS and has no license costs
 - anyone can use it (respecting GNU AGPL license terms) for free
 - but all related R&D still has to be funded. We rely on your contribution!
- sysmocom provide support, training, tested releases, consulting, integration services to
 - BTS / equipment vendors
 - operators (commercial and non-commercial)

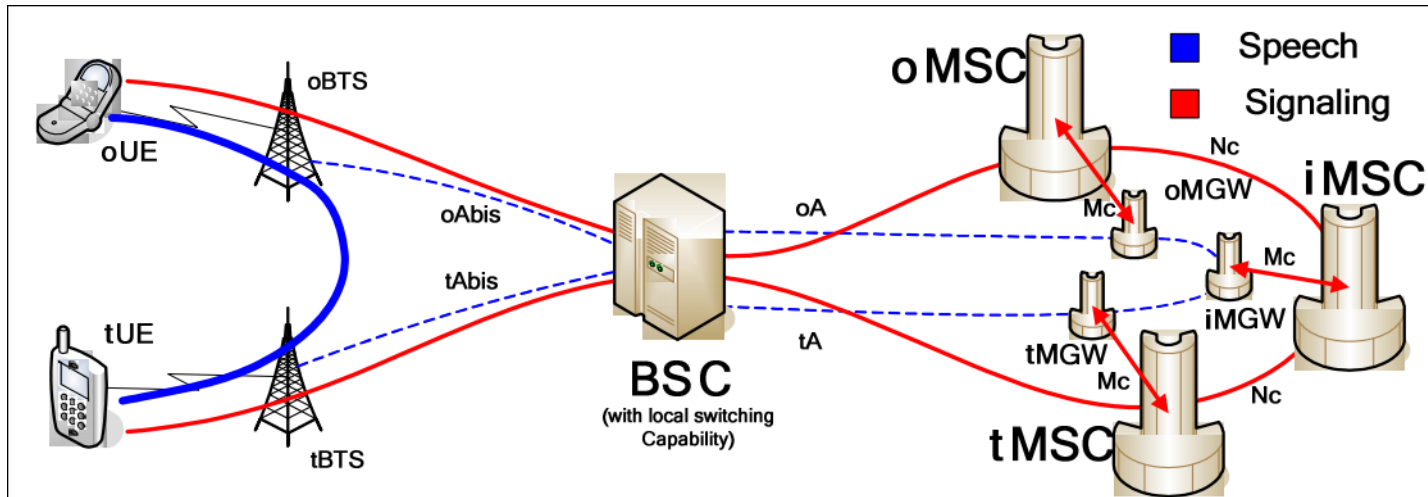
- Founded 2011 by two inventors of OpenBSC + OsmoNITB: Holger Freyther + Harald Welte
- Located in Berlin (Germany)
- Current team size of 11 (8 of which are R&D engineers)
- Development of solutions and technology for mobile networks
 - from PHY/SDR to RAN to Core Network to SIM cards
 - embedded electronics design and software development
- 100% owner driven and financed; only organic growth
- All our Osmocom related work is FOSS. We don't believe in proprietary black-boxes.



- **OsmoCon: Osmocom Conference**
- Annual conference for Osmocom users and operators
- Two days of talks about latest developments and use cases
- **October 18 + 19, 2018** in Berlin, Germany
 - just right after TIP Summit in London; you can simply hop over to Berlin
- More Information, Schedule and Tickets at: <https://osmocom.org/OsmoCon2018>
- Looking forward to meeting you at OsmoCon 2018!

- OsmoBSC: 3GPP LCLS (local call, local switch)
- OsmoBSC: inter-BSC (external) hand-over
- OsmoBSC: 3GPP AoIP (A interface over IP)
- OsmoBSC: load-based hand-over
- OsmoMSC/OsmoHLR: USSD gateway for external USSD applications
- OsmoTRX: Native LimeSDR/LimeSuite support (osmo-trx-lms)
- OsmoBTS: Massive improvements on Measurement Reporting
- TTCN-3 based automatic integration test suites

- OsmoBSC has been extended with 3GPP LCLS support
 - keeps user/voice plane local to the RAN if both parties of a call are in same RAN
 - significant reduction of AoIP back-haul bandwidth (and no satellite-induced latency)



- OsmoBSC has been extended with inter-BSC (external) hand-over
 - before, we only supported intra-BSC (internal) hand-over
 - how calls can be handed over between cells served by different BSCs
 - interop testing has been performed against NG4T and Quortus so far

- OsmoBSC has been extended for 3GPP AoIP (A interface over IP)
- previously, we only supported SCCPlite (SCCP over IPA multiplex)
 - implemented by Altobridge, Quortus, Zynetix, ip.access
 - large vendors (Ericsson, Huawei, etc.) used to do A-over-E1 and now 3GPP AoIP
- we now have M3UA + SCCP implementation in libosmo-sigtran
 - used by OsmoSTP, OsmoBSC and OsmoMSC
- AoIP is not just transport, but also BSSMAP level protocol changes
 - media plane now signaled as RTP IP/port, not as TDM CIC anymore

- OsmoBSC has received load-based handover support
 - previously, only link budget (RxLev/RxQual) based hand-over decisions
- Load-based handover means equalizing load of multiple (overlapping) BTSs
- Execution / mechanics of hand-over remain identical to link budget HO
 - just handover decision is different
- Implemented by new “handover 2” algorithm in OsmoBSC

- OsmoMSC/OsmoHLR has been extended with USSD gateway function
 - previously, USSD was handled directly inside OsmoMSC, hard-coded
- OsmoMSC now hauls back all USSD to OsmoHLR
 - uses GSUP protocol like all other CN signaling between Osmo* components
- OsmoHLR contains some IUUSE (Internal **U**SSD **E**ntities)
 - replicating the old internal handling of OsmoMSC
- External USSD applications can now be developed
 - attach as EUSE (**E**xternal **U**SSD **E**ntity) to OsmoHLR
 - OsmoHLR contains USSD code routing tables

- OsmoTRX has been split in common and hardware/driver specific code
 - previously, you had to decide at compile time if you want to build for USRP1 or UHD
 - complex emulation driver stacks had to be used, e.g.
LimeSuite→SoapySDR→SoapyUHD→UHD→OsmoTRX
- Using new architecture, we build several executables at compile time
 - osmo-trx-uhd for UHD supported devices like Ettus USRP \geq USRP2
 - osmo-trx-usrp1 for Ettus USRP1
 - osmo-trx-lms for LimeSuite (LimeSDR USB/PCI/mini)

- OsmoBTS: Massive improvements on Measurement Reporting
- OsmoBTS always sent measurement reports
 - but have they been correct in all cases? Unfortunately no
- Many corner cases were incorrect, such as behavior in case of lots bursts, DTX, missing uplink reports from mobile stations
- Related code has been rewritten significantly
- Test suite has been developed to ensure measurement processing/reporting is correct
- Measurement reports also contain TOA in 256 times higher temporal resolution
 - useful for location services

- Osmocom has developed extensive test suites, written in TTCN-3
 - TTCN-3 is an ETSI/ITU programming language *specifically* for protocol testing
- We call related tests “integration testing”
 - each testsuite tests [all] external interfaces of each network element
 - we therefore ensure compliance of our external interfaces
- Tests are automatically executed every 24 hours as part of our continuous integration
 - Jenkins Test Results analyzer shows trends
 - Results are public: <https://jenkins.osmocom.org/jenkins/view/TTCN3/>
- For every newly observed bug, we generally add a new TTCN3 test to ensure it stays fixed

- If you use Osmocom, it's not a classic supplier ↔ customer relationship
- Osmocom is a community project. Open to anyone. No formal membership
- Just join our mailing lists and participate in discussion
- Register an account on osmocom.org and file bug reports, feature requests
 - you don't need to be a developer to e.g. reproduce reported bugs, check if they're fixed, etc.
- Anyone with the related programming skills can send contributions
 - If you modify Osmocom, submit your changes back to us, don't keep them separate

- Osmocom develops cellular network elements + protocol stacks
- All resulting software is Free/Open Source Software, available to anyone
- You can use Osmocom GSM/GPRS/UMTS
 - as RAN to classic 3GPP Core Network (if yo have your MSC/HLR/SGSN/...)
 - as autonomous Network-In-the-Box
- Osmocom is deployed in production networks for 7+ years
- Osmocom interoperates with BTS equipment from many vendors
- OpenCellular uses Osmocom at it heart, to implement GSM/GPRS/EDGE RAN+CN
- sysmocom provides professional R&D, training and support service around Osmocom

- Osmocom User Manuals: <http://ftp.osmocom.org/docs/latest/>
- Osmocom Cellular Homepage: <http://osmocom.org/projects/cellular-infrastructure/wiki>
- Video Recordings of Osmocom talks at Conferences:
https://osmocom.org/projects/cellular-infrastructure/wiki/Videos_of_Presentations
- Mailing List for discussions and Q&A: openbsc@lists.osmocom.org
- IRC: [#osmocom](#) on freenode