# Erlang SCCP/TCAP/MAP Implementations

Harald Welte <laforge@gnumonks.org>

gnumonks.org
hmw-consulting.de
sysmocom GmbH

October 13, 2012 - OSDC.fr - Paris / France

# Outline

# GSM core network components

- **MSC** (Mobile Switching Center): The central switch
- **HLR** (Home Location Register): Database of subscribers
- **AUC** (Authentication Center): Database of authentication keys
- **VLR** (Visitor Location Register): For roaming users
- **EIR** (Equipment Identity Register): To block stolen phones

# GSM network structure

MSC Actual call switching and top-level mobility functions. May serve dozens of location areas

VLR Temporary cache of subscriber data from HLR + TMSI

HLR Subscriber databases + subscriber location information

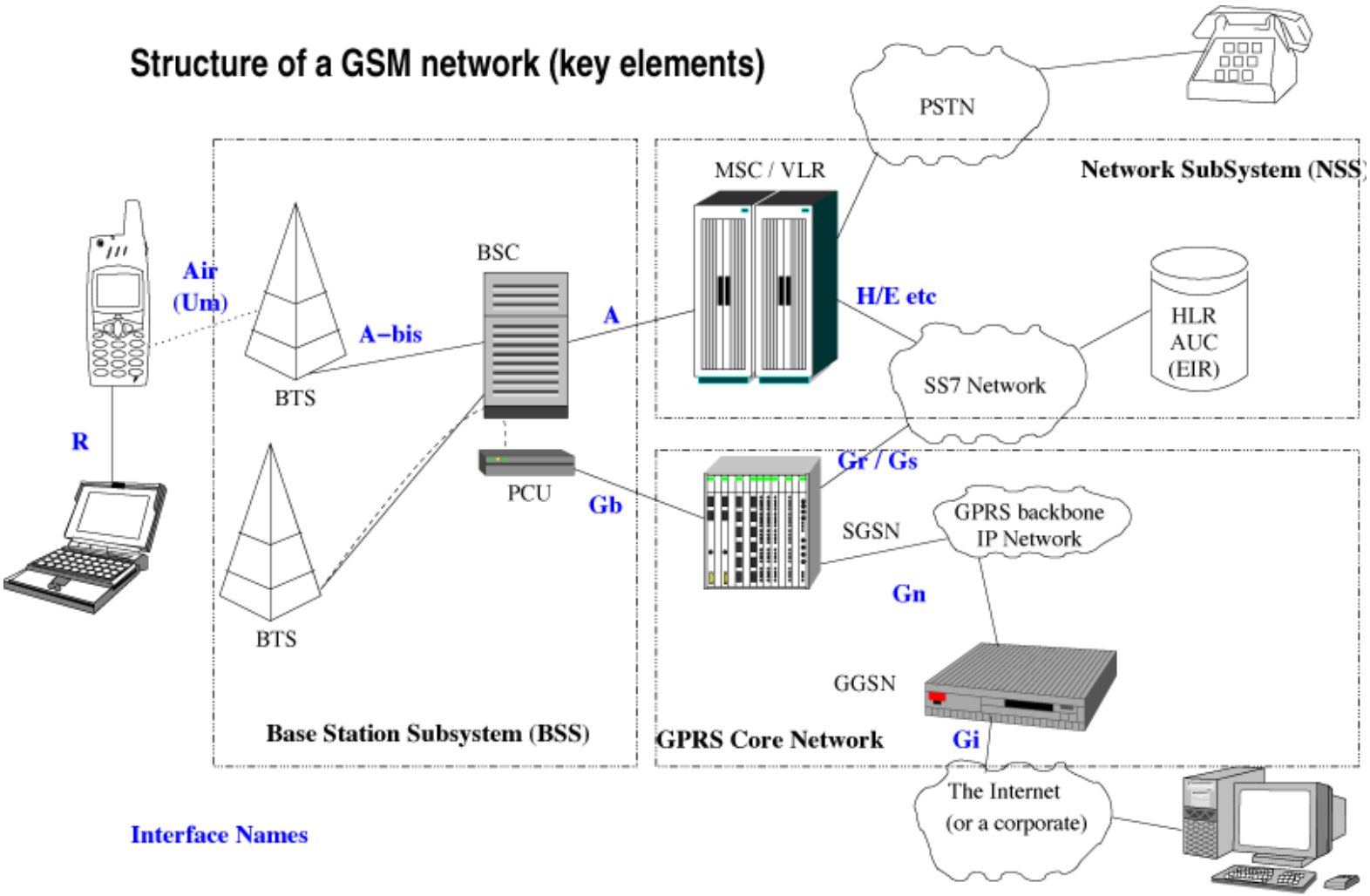
AUC Generation of authentication tuples

SMSC SMS Service Centre, store+forward for SMS

# GSM core network integration

- VLR often integrated into MSC
- AUC often integrated with AUC
- integration so common, many graphs/diagrams are actually not 100% correct

# GSM Network Structure



# GSM network interfaces

- C Interface between GMSC and HLR
- D Interface between MSC and HLR
- E Interface between MSC and MSC

All of them based on MAP, so C/D/E not commonly distinguished

# core network protocol stack

Traditional telephony based on SS7 / CS7, GSM too

- Lower layers (MTP2/MTP3) re-used
- ISUP used for actual call control signalling
- SCCP for routing / GTT
- TCAP for transaction supprt
- MAP for actual GSM related signalling

# SS7 networks

- STP - Signalling Transfer Point
  - *Router* for SCCP
  - performs GTT (see below)
- SCP - Signalling Control Point
  - *End-node* like MSC/HLR
  - SCP has GT, PC, ..

# SS7 addresses

- Point Code (PC)
  - typically unique within PLMN / country
- Global Title (GT)
  - world-wide unique address
  - translated into PC by GTT at STP
- Subsystem Number (SSN)
  - logical function address inside network (MSC, VLR, HLR, ...)
  - not used on international links

# SS7 GTT (Global Title Translation)

Global Title Translation

- can happen at any STP
- translates a Destination GT into new destination address
- new dest address can be any address, such as
  - new global title (GT)
  - point code (PC)
  - sub-system number (SSN)
- GTT rules explicitly configured by operator, e.g.
  - prefix or range based match
  - (inter)nationalize numbering plan
  - add digits at beginning or end

# SS7 physical layer

- classic SS7 signalling over TDM circuits
  - E1 timeslot (64kbps)
  - multiple E1 timeslots (N*64kbps)
  - MTP Level 2 / MTP Level 3
- modern networks use SIGTRAN
  - IP as network layer replaces E1 lines
  - SCTP on top(no TCP/UDP!)
  - many different SIGTRAN stacking options
- some vendor-proprietary protocols like SCCPlite

# SIGTRAN stacking options

SIGTRAN != SIGTRAN

- IP/SCTP/M2PA/MTP2/MTP3/SCCP/TCAP/MAP
- IP/SCTP/M2UA/MTP3/SCCP/TCAP/MAP
- IP/SCTP/M3UA/SCCP/TCAP/MAP
- IP/SCTP/SUA/TCAP/MAP

# SCCP

SCCP takes care of

- Global Title based addressing
- Global Title Translation
- connection-oriented or connectionless semantics
- GSM core network interfaces with MAP/CAP only use connection-less UDT service

# TCAP

- Idea: decouple transaction logic from actual application
- transaction semantics can be used by multiple higher-layer protocols
- state machines on both sides maintained outside of application
- protocol specified in ASN.1, BER encoding

# MAP - Mobile Application Part

- used between all classic GSM core network components
- application protocol on top of TCAP
- protocol specified in ASN.1, BER encoding

# CAP - Camel Application Part

- used for CAMEL entities (gsmSCF, gsmSSF, gprsSSF, gsmSRF)
- application protocol on top of TCAP
- protocol specified in ASN.1, BER encoding

# Introducing Erlang

Erlang/OTP

- is a functional, non-OO programming language
- promotes some principles that make it easier to write secure code
- was crreated by Ericsson for Telecom signalling applications
- has excellent built-in ASN.1 compiler + runtime support
- has `gen_fsm` support for well-defined finit state machines

# Safe and secure programming

Erlang enables and encourages to

- avoid defensive programming, rather fail-fast and raise exceptions
- avoid having global/shared state as everything is pass-by-value, not reference
- avoid accidential/improper reuse of variables by single assignment
- not have to worry about memory allocation problems like buffer overflows / double-free

# Erlang headaches

If you're used to C/C++ or even Java, Erlang will give you headaches, too.

- you have no interative loops like for/while, but always have to use (tail) recursion
- you have to type a lot when accessing members of records (structures), as you need to specify the type name on every access
- avoiding global state may be useful, but very hard at times

# Reasons to use Erlang in Osmocom

- best ASN.1 support found as Free Software for any programming language
  - TCAP/MAP use ASN.1 Information Object Classes, which e.g. asn1c doesn't support
  - supports PER aligned and unaligned, required in RANAP/RRC for UMTS.
  - very strict validation of input data, including range checks of integer values against constraings in ASN.1, etc.
- built-in support for finite state machines
- Erlang *many processes and message passing* model 1:1 match to ITU TCAP specification.

# Erlang in Osmocom projets

- all current Osmocom developers are C (possibly C++) developers
- nobody really likes to use some bloatet inefficient and unknown programming language (compared to C...)
- almost every other sub-project of Osmocom is implemented in pure C
- apart from my projects described here, Erlang hasn't really picked up with other developers
- Erlang wasn't chosen because we love it, but because it makes techical sense in some specific applications, compared to alternavies requiring to buy/user porprietary ASN.1 tools or write our own

# Erlang osmo_ss7

- Signalling link management
- Signalling linkset management
- MTP-level routing
- Protocol codecs
  - BSSMAP, ISUP, M2PA, M2UA, M3UA, MTP3, SCCP, SUA
- Various different protocol implementations
  - SIGTRAN: M3UA, M2PA, M2UA, SUA
  - IPA multiplex / SCCP lite

# Erlang osmo_sccp

SCCP implementation, typcially used on top of osmo_sccp

- SCCP connectionless (SCLC)
- SCCP connection oriented (SCOC)
- SCCP routing / gtt (SCRC)
- applications can bind to SSN numbers

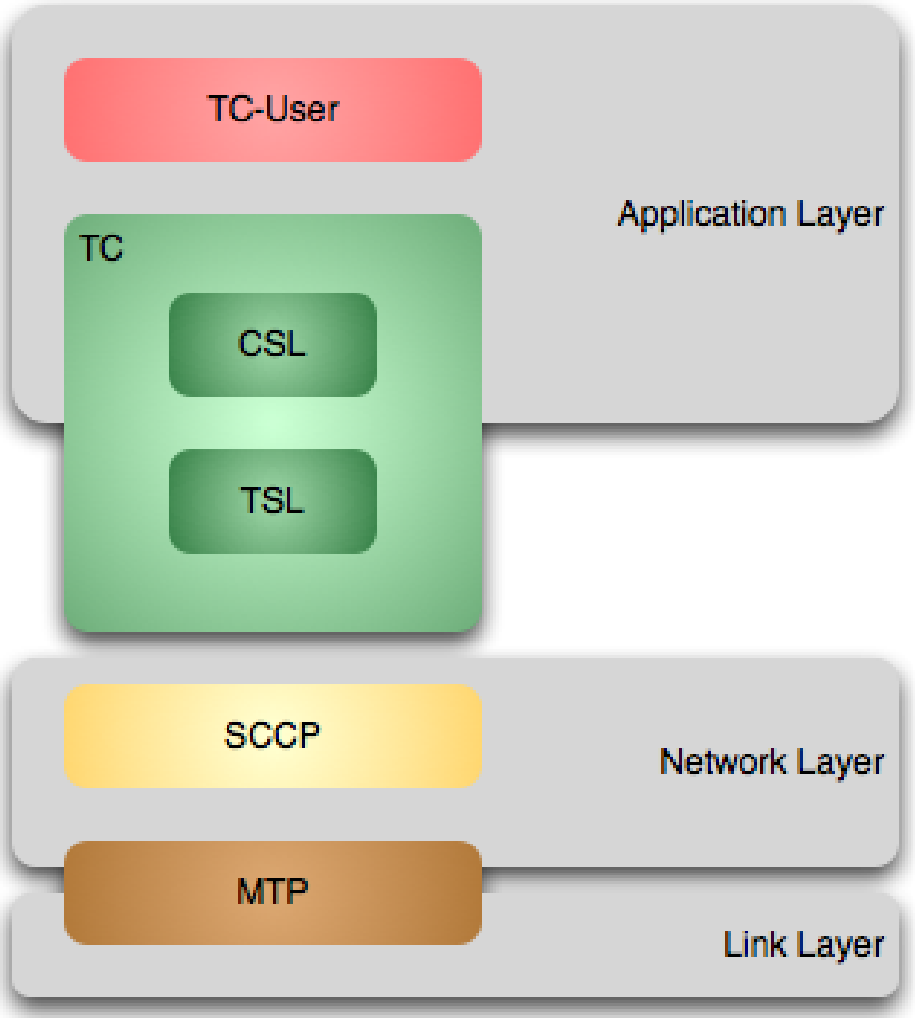
# Erlang osmo_map

- Not a full-blown MAP end-user implementation
- Primarily a set of integrated TCAP+MAP codec
- Used for protocol analysis/dissection
- Used for transparent MAP mangling engines
- Think of FTP/IRC NAT in TCP/IP, where you need to modify addresses contained in the payload (not header) of the messages

# Erlang mgw_nat

- Strange transparent SCCP/TCAP/MAP gateway
- Supports all kinds of strange operations
  - SCCP Global Title Masquerade (dynamic GT pool)
  - Replace VLR/MSC GT inside MAP payload
  - Supported Camel Phase patching
  - 1:1 IMSI mapping in MAP payload
  - ISUP GT mangling
  - national/international numbering plan conversions
- Used in multiple production installations in real operator core network for 2 years

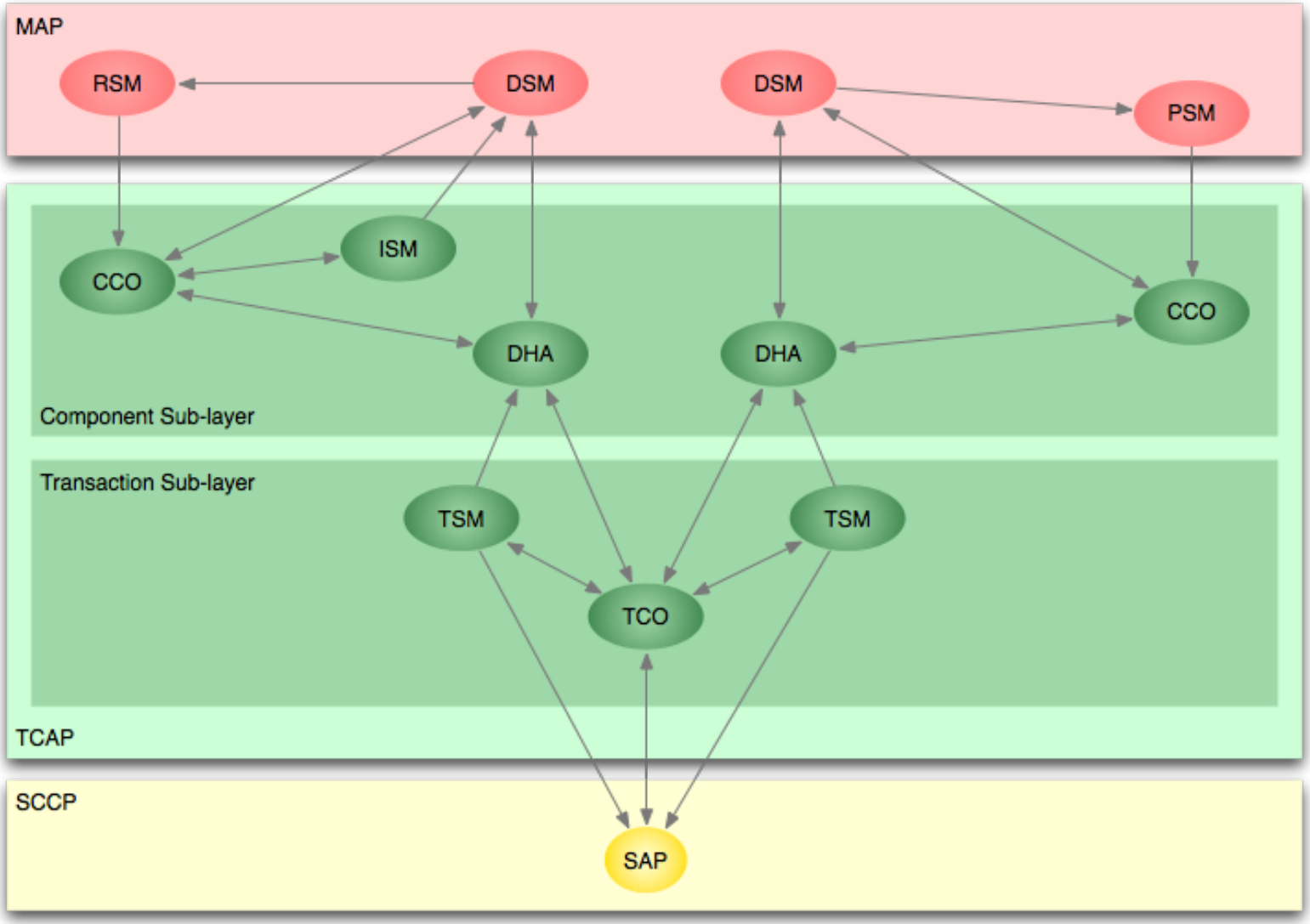
# Erlang signerl TCAP



# Erlang signerl TCAP

- Full ITU-T TCAP implementation
- 1:1 mapping of ITU-T TCAP state machines to Erlang gen_fsm
  - DHA - Dialogue Handling
  - TSM - Transaction State Machine
  - ISM - Invocation State Machine
- 1:1 mapping of other ITU-T entities to Erlang gen_server
  - CCO - Componen Coordinator
  - TCO - Transaction Coordinator
- Some old/incomplete/bit-rotten ANSI TCAP code

# Message flow among signerl TCAP Processes



# Erlang supervisor hierarchy in signerl TCAP

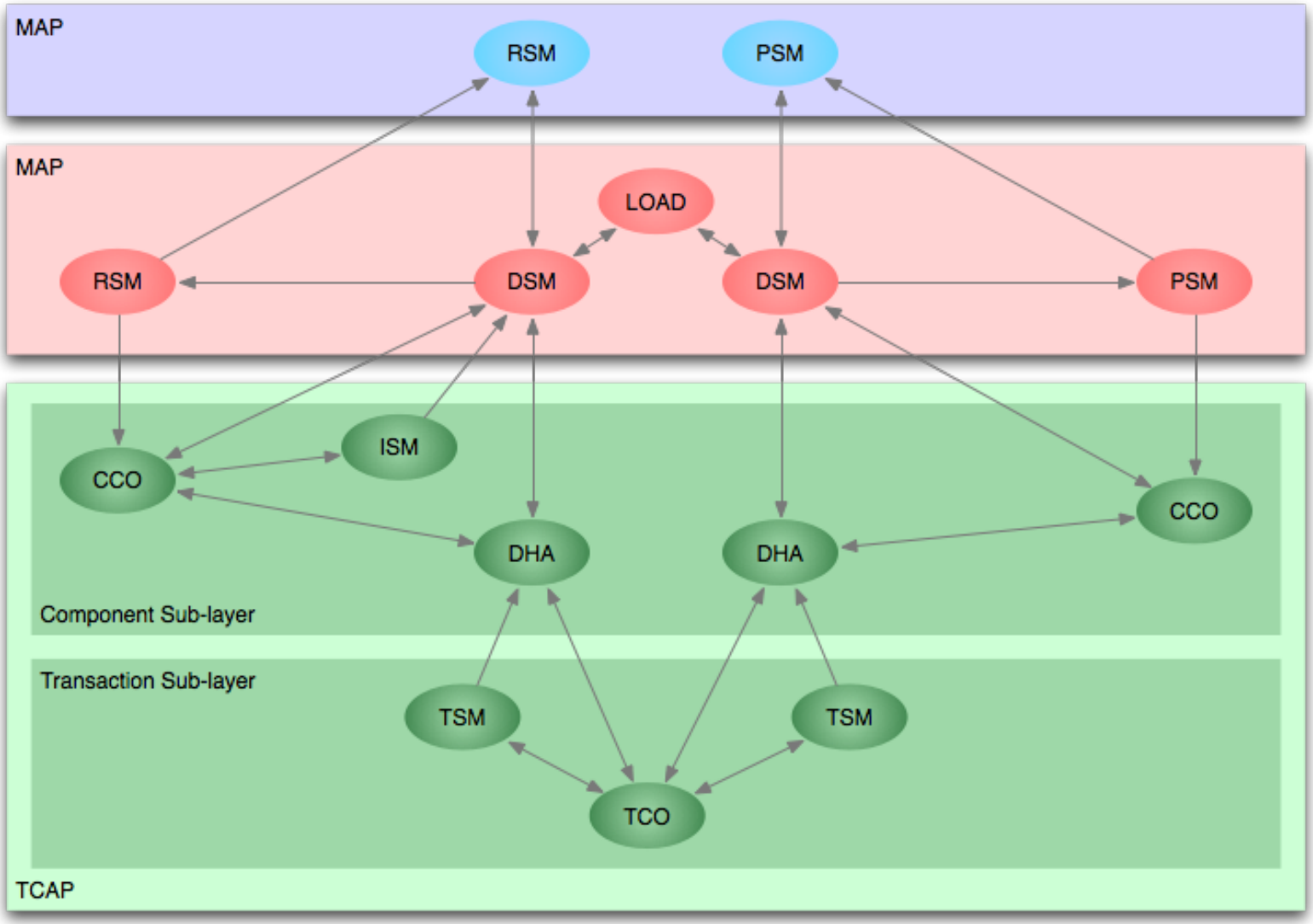
# Erlang signerl TCAP

- properly implements the N-primitives to lower level
- properly implements all TR-primitives internally (TC / TR split)
- properly implements all TC-primitives towards the TCAP user
- Can be used on top of osmo_sccp
- Can be used directly by application servers or via signerl MAP

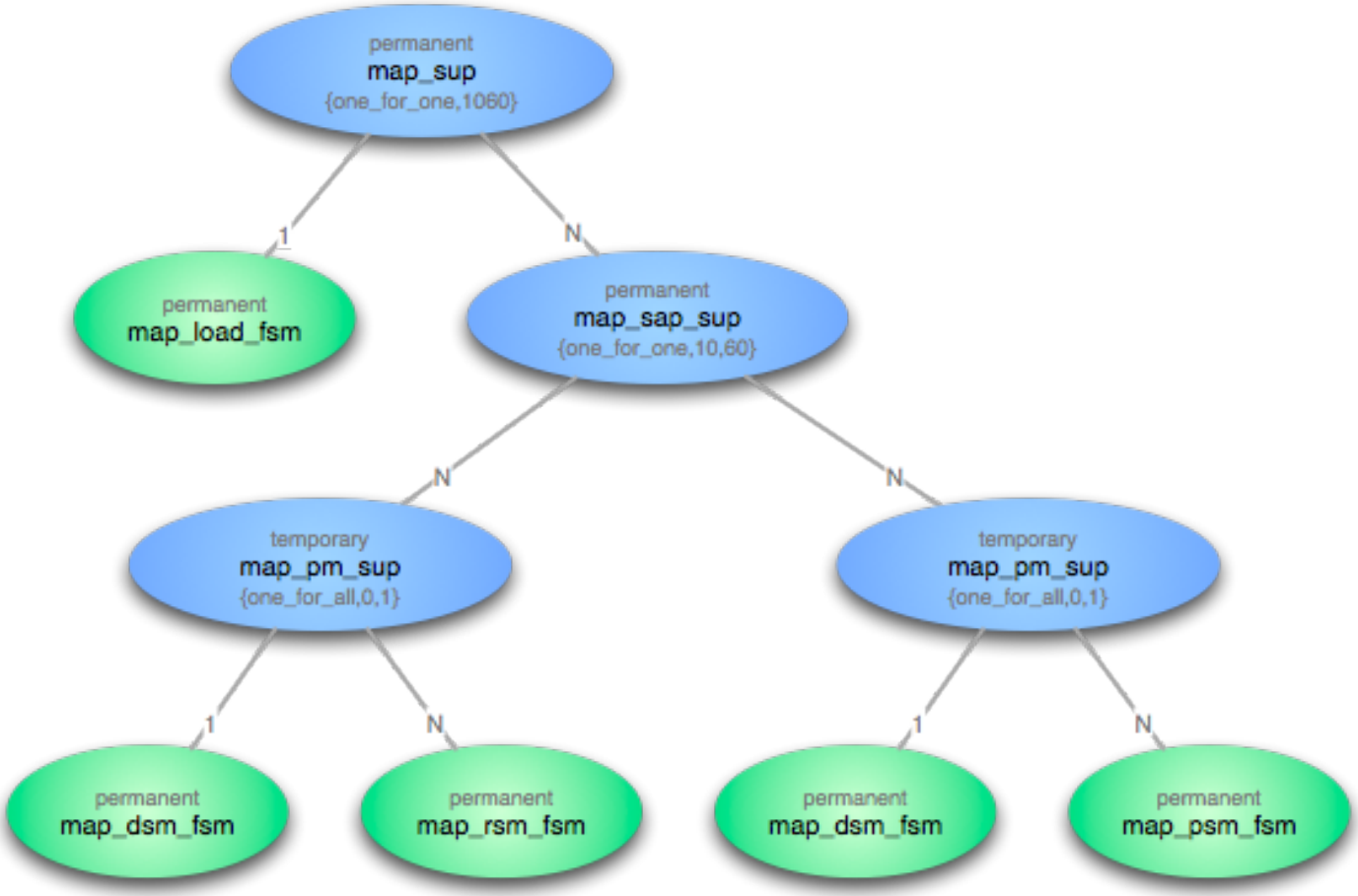
# Erlang signerl MAP

- Interface between MAP primitives and TCAP primitives
- Provides very little benefit over using TCAP directly
- Not used much so far, I always use TCAP user API instead

# Message flow among signerl MAP Processes



# Erlang supervisor hierarchy in signerl TCAP



# Erlang application servers

- No complete implementation of any GSM core network node yet
- Lots of testing / experimentation code for generating single MAP transactions against existing/proprietary core network components
- Work on a HLR based on Mnesia DB should be starting soon