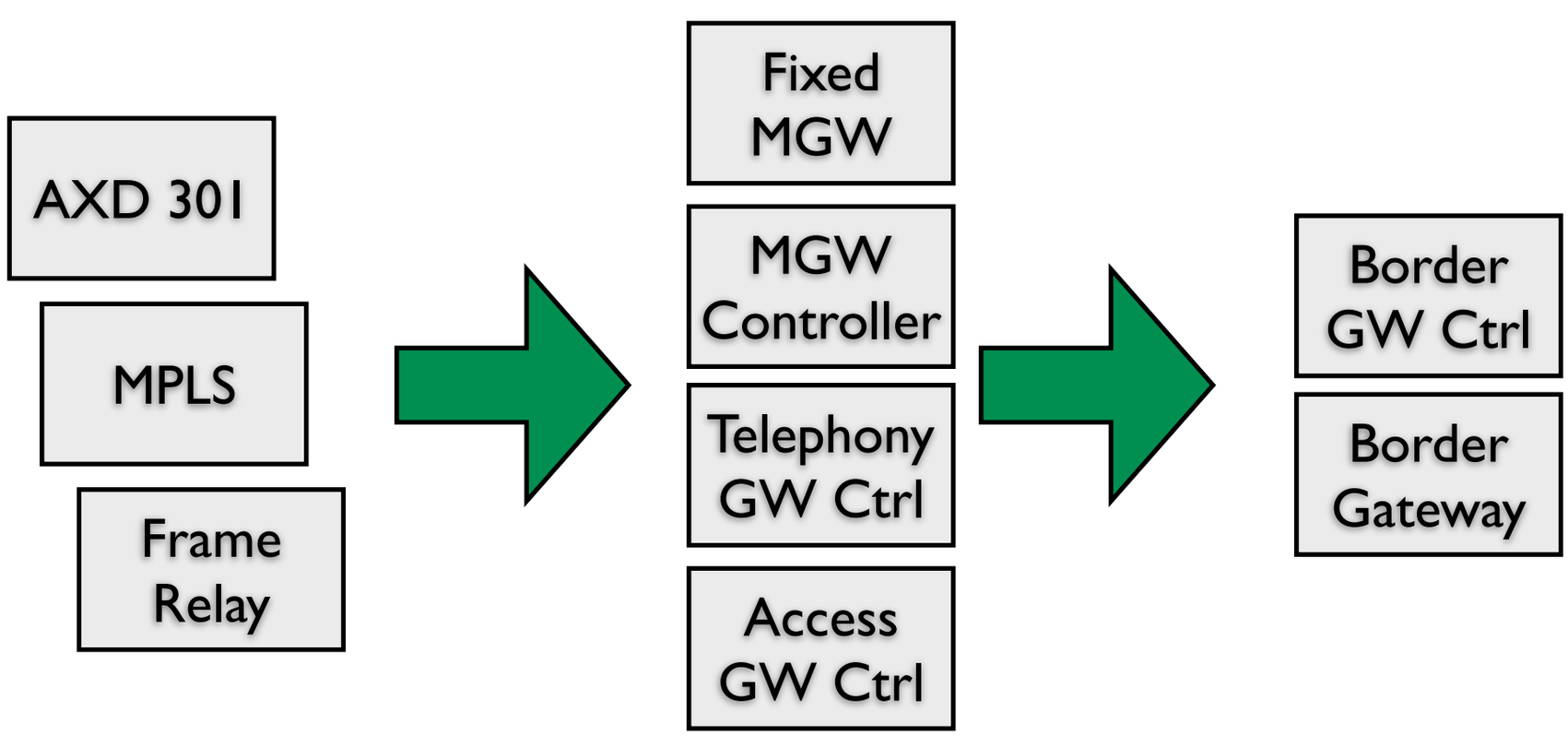
Erlang Solutions Ltd.

The Erlang/OTP Diameter Stack

Ulf Wiger, CTO Erlang Solutions Ltd

About me

- 13 years at Ericsson
- Chief designer of a number of Erlang-based products
- CTO, Erlang Solutions since 2009



Background

- Ericsson Session Border Gateway Controller
  - (essentially a multimedia pin-hole firewall)
  - (re-)written in Erlang in one year
- Diameter stack needed for
  - Location lookup
  - Resource reservation
  - Billing
- Initial idea was to buy an existing stack

Shopping around

- Evaluated some 3rd party stacks written in C
  - Expensive
  - “Full-featured”, but still needing customization
  - Lots of C-Erlang adaptation work expected
- One Erlang-based 3rd-party stack
  - Nice, but didn’t have the add-ons we needed
  - More expensive than to build an in-house stack
- In-house alternative: I wrote first version (6-7 months)
  - Anders Svensson, later OTP, took over
  - Rewrote the whole thing...

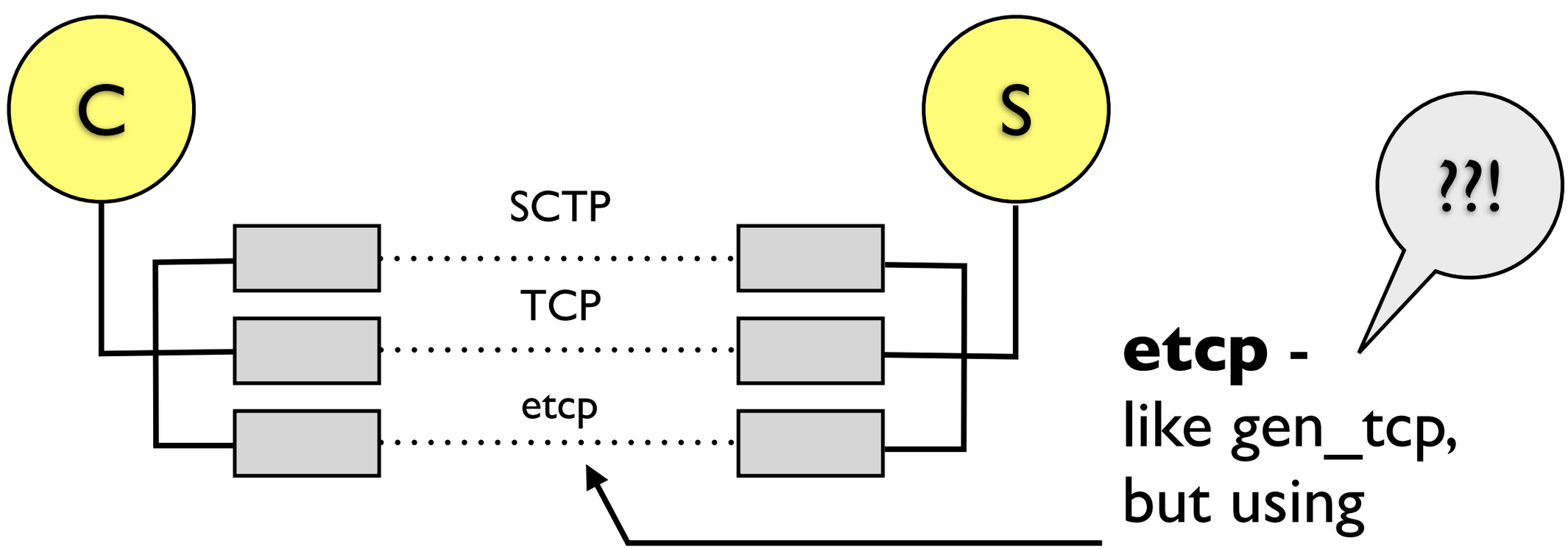
Erlang

Requirements

- Fast
  - 3-4 Diameter calls per session setup
    - On top of lots of SIP-related activity
  - Total session setup latency budget: **20 ms**
- Flexible
  - 3GPP and TISPAN interfaces a moving target
  - Embrace and extend attitude
- Easy to use
  - Fast-paced & complex development project

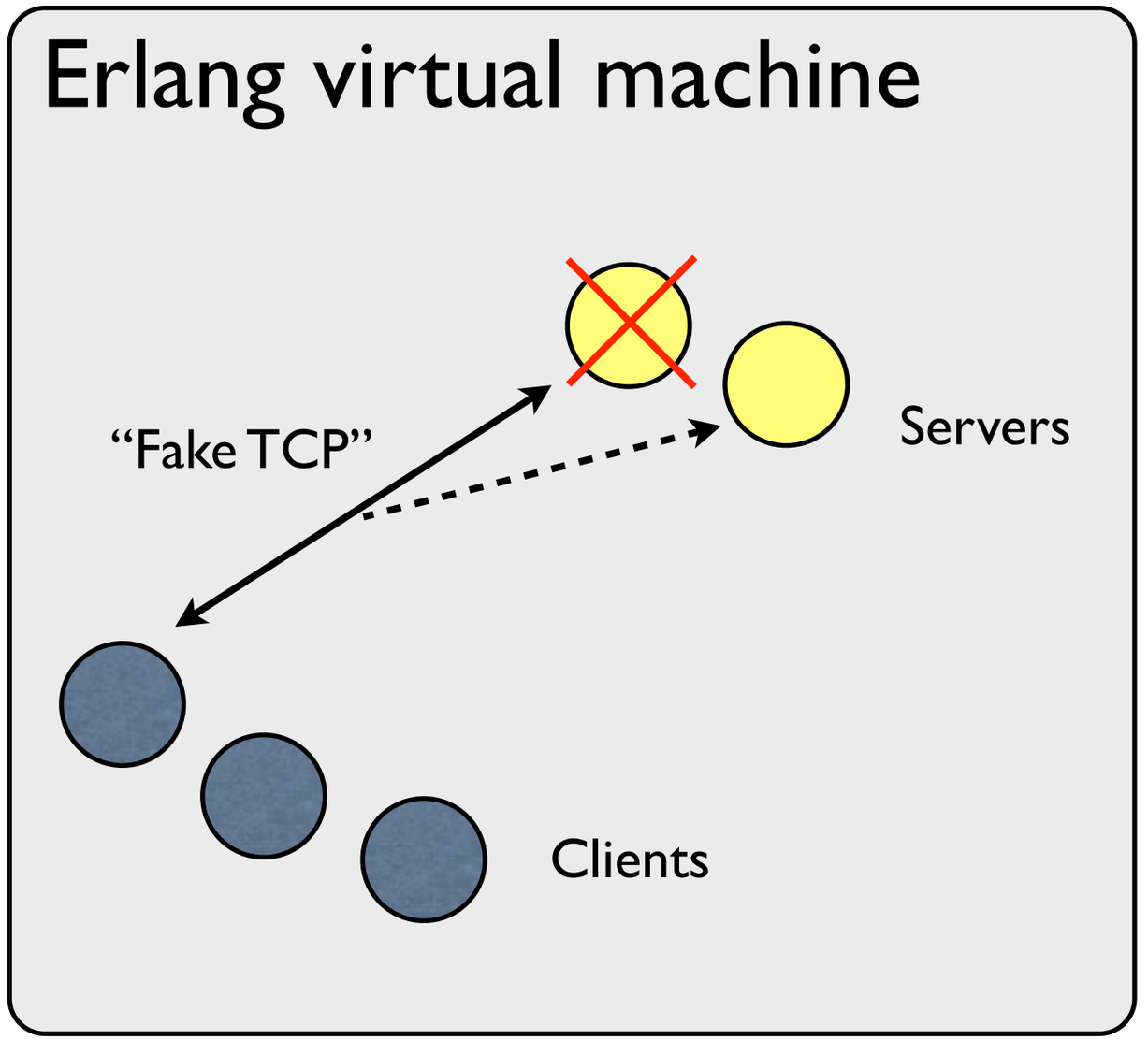
Only the Client Needed (...?)

- Yes, ...but how to test a client?
  - No e2/Rx/Ro Diameter server simulators existed
- Solution: write a server too!



Erlang - Testing Nirvana

- Superb testing tools
- Dead-simple to write stubs and simulators
- Secret sauce for interoperability testing
- Revealed many bugs in the specs...



Summary

- Cheaper to build from scratch in Erlang than to buy and integrate off-the-shelf C code
  - Great way to learn the protocol too
- Already in use in several commercial products
  - Both Erlang and C++
  - Both client and server

RFC written for C programmers

Example: RFC 3588 – DIAMETER Base Protocol

```
state            event              action          next state
-----------------------------------------------------------------
...
I-Open           Send-Message       I-Snd-Message   I-Open
                 I-Rcv-Message      Process         I-Open
                 I-Rcv-DWR          Process-DWR,    I-Open
                                    I-Snd-DWA
                 I-Rcv-DWA          Process-DWA     I-Open
                 R-Conn-CER         R-Reject        I-Open
                 Stop               I-Snd-DPR       Closing
...
```

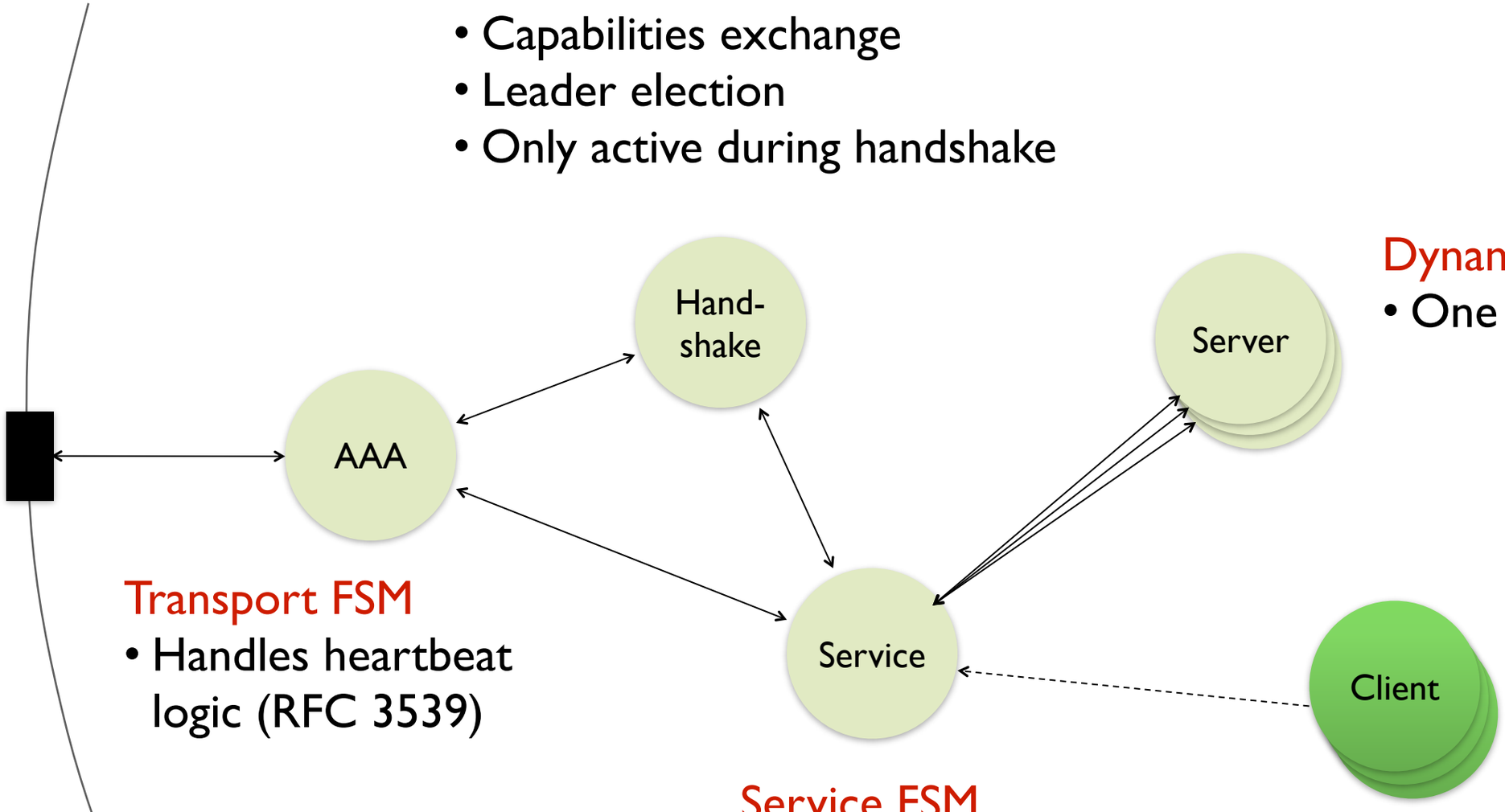
Transport FSM

Handshake FSM

- Three state machines described as one
- Implies a single-threaded event loop
- Introduces accidental complexity

DIAMETER, Erlang-Style

Handshake FSM
- Capabilities exchange
- Leader election
- Only active during handshake



Dynamic request handler
- One per request

Transport FSM
- Handles heartbeat logic (RFC 3539)

Service FSM
- Request routing
- Failover
- Retransmission

Defining a Diameter "application"

Copy-paste from the spec
into a .dia text file

```
@id 0
@prefix diameter_base
@vendor 0 IETF

@avp_types
   Acct-Interim-Interval              85   Unsigned32    M
   Accounting-Realtime-Required       483  Enumerated    M
   Acct-Multi-Session-Id              50   UTF8String    M
...
   Proxy-Info                         284  Grouped       M

...
@messages
...
```

ml9-style
headers

AVPs =
Attribute-Value Pairs

Compound type

Defining a Diameter application (2)

```
@messages

    CER  ::= < Diameter Header: 257, REQ >
                  { Origin-Host }
                  { Origin-Realm }
               1* { Host-IP-Address }
                  { Vendor-Id }
                  { Product-Name }
                  [ Origin-State-Id ]
                  ...
               * [ Vendor-Specific-Application-Id ]
                  [ Firmware-Revision ]
               * [ AVP ]

    CEA  ::= < Diameter Header: 257 >
                  { Result-Code }
                  { Origin-Host }
                  { Origin-Realm }
                  ...
```

{Mandatory}

[Optional]

Defining a Diameter application (3)

```
@enum Disconnect-Cause

   REBOOTING                                0
   BUSY                                     1
   DO_NOT_WANT_TO_TALK_TO_YOU               2

@grouped

      Proxy-Info ::= < AVP Header: 284 >
                     { Proxy-Host }
                     { Proxy-State }
                   * [ AVP ]
```

Compound type

Generated Erlang module

```
-module(diameter_gen_base_rfc3588).

-compile([{parse_transform, diameter_exprecs}]).
...
-export_records([diameter_base_CER, diameter_base_CEA,
                 ...,
                 'diameter_base_E2E-Sequence']).
```

Like ‘exprecs’ in http://github.com/esl/parse_trans

```
-record(diameter_base_CER,
        {'Origin-Host', 'Origin-Realm', 'Host-IP-Address' = [],
         'Vendor-Id', 'Product-Name', 'Origin-State-Id' = [],
         'Supported-Vendor-Id' = [], 'Auth-Application-Id' = [],
         'Inband-Security-Id' = [], 'Acct-Application-Id' = [],
         'Vendor-Specific-Application-Id' = [],
         'Firmware-Revision' = [], 'AVP' = []}).

...
```

travelping

The Erlang/OTP DIAMETER Stack
or
What‘s all about?

6/14/11

Erlang Factory London 2011

About me…

- Holger Winkelmann
- Founder & MD of Travelping GmbH
- Working for ISP & Telco for 16 Years
- AAA and Control Plane Expertise

What‘s DIAMETER ???

travelping

1st email after Release

Another Protocol in Erlang/OTP

travelping

- Erlang/OTP – a protocol factory
- Already many Telco centric Implementations
- ASN.1, SNMP, MEGACO, H.248

- DIAMETER stack is used in many Telco Products
- The DeFacto Control Plane Protocol in Mobile Operators Networks
- Defined in RFC3588… and many 3gpp TS

DIAMETER – RADIUS History

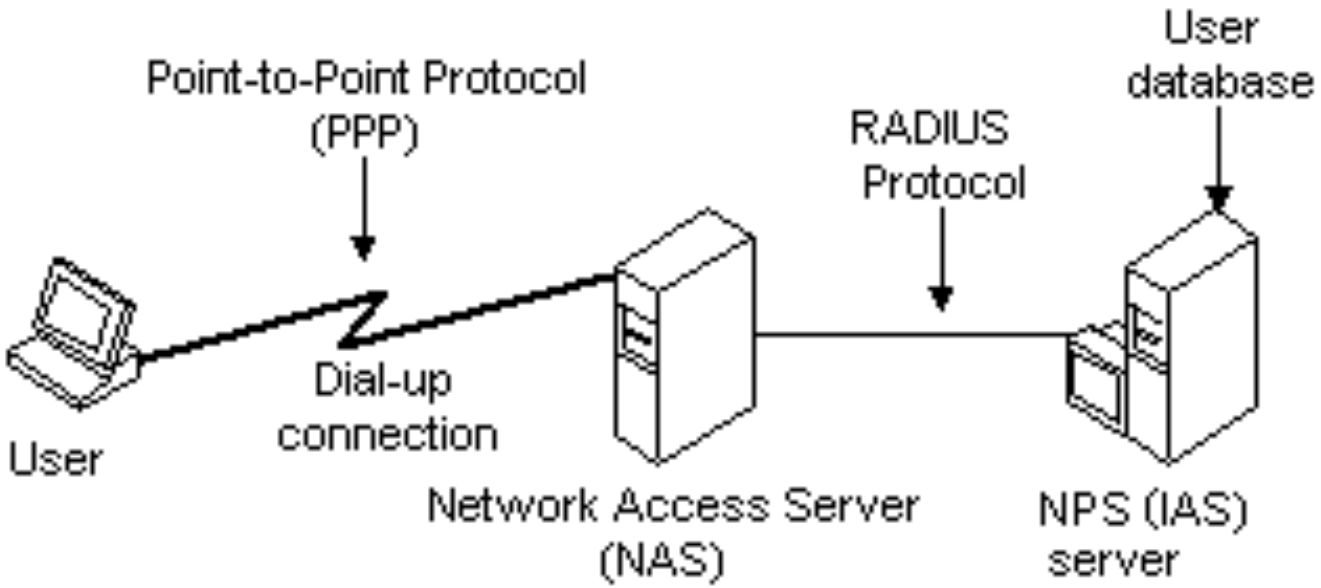
travelping

- **Diameter** is an authentication, authorization and accounting (**AAA)** protocol for computer networks, and a successor to **RADIUS**

- **Remote Authentication Dial In User Service** (**RADIUS**) is a networking protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for computers to connect and use a network service.

How RADIUS Works

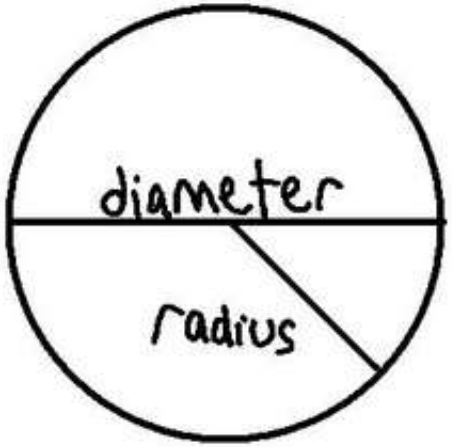
- **RADIUS** – UDP based protocol defined in RFC2865 – RFC2866 plus man other.
- **ISP Dial-In Example**



DIAMETER vs. RADIUS

travelping

- The name is a pun on the **RADIUS** protocol, which is the predecessor (a diameter is twice the radius). **DIAMETER** is not directly backwards compatible, but provides an upgrade path for **RADIUS**.



DIAMETER Differences

- Reliable transport protocols (TCP or SCTP, not UDP)
- Network or transport layer security (IPsec or TLS)
- Transition support for RADIUS, although Diameter is not fully compatible with RADIUS
- Larger address space for attribute-value pairs (AVPs) and identifiers (32 bits instead of 8 bits)
- Client–server protocol, with exception of supporting some server-initiated messages as well
- Both stateful and stateless models can be used
- Dynamic discovery of peers (using DNS SRV and NAPTR)
- Capability negotiation, Error notification
- Aligned on 32-bit boundaries

Diameter Applications

travelping

- Separation between Base Protocol and Application
- DIMATER Application is NOT a Software Application
- Diameter Applications are define as:
  - Application Identifier
  - Command Codes
  - Mandatory Attribute Value Peers (AVPs)
- more like a specification, contract or dictionary rather a Application.
- DIAMETER allows to define multiple Applications and can be widely for AAA Purposes

Sample Applications

travelping

- Diameter Mobile IPv4 Application (MobileIP, RFC 4004)
- Diameter Network Access Server Application (NASREQ, RFC 4005)
- Diameter Extensible Authentication Protocol Application (RFC 4072)
- Diameter Credit-Control Application (DCCA, RFC 4006)
- Diameter Session Initiation Protocol Application (RFC 4740)
- Various applications in the 3GPP IP Multimedia Subsystem (IMS)

Usage of DIAMETER

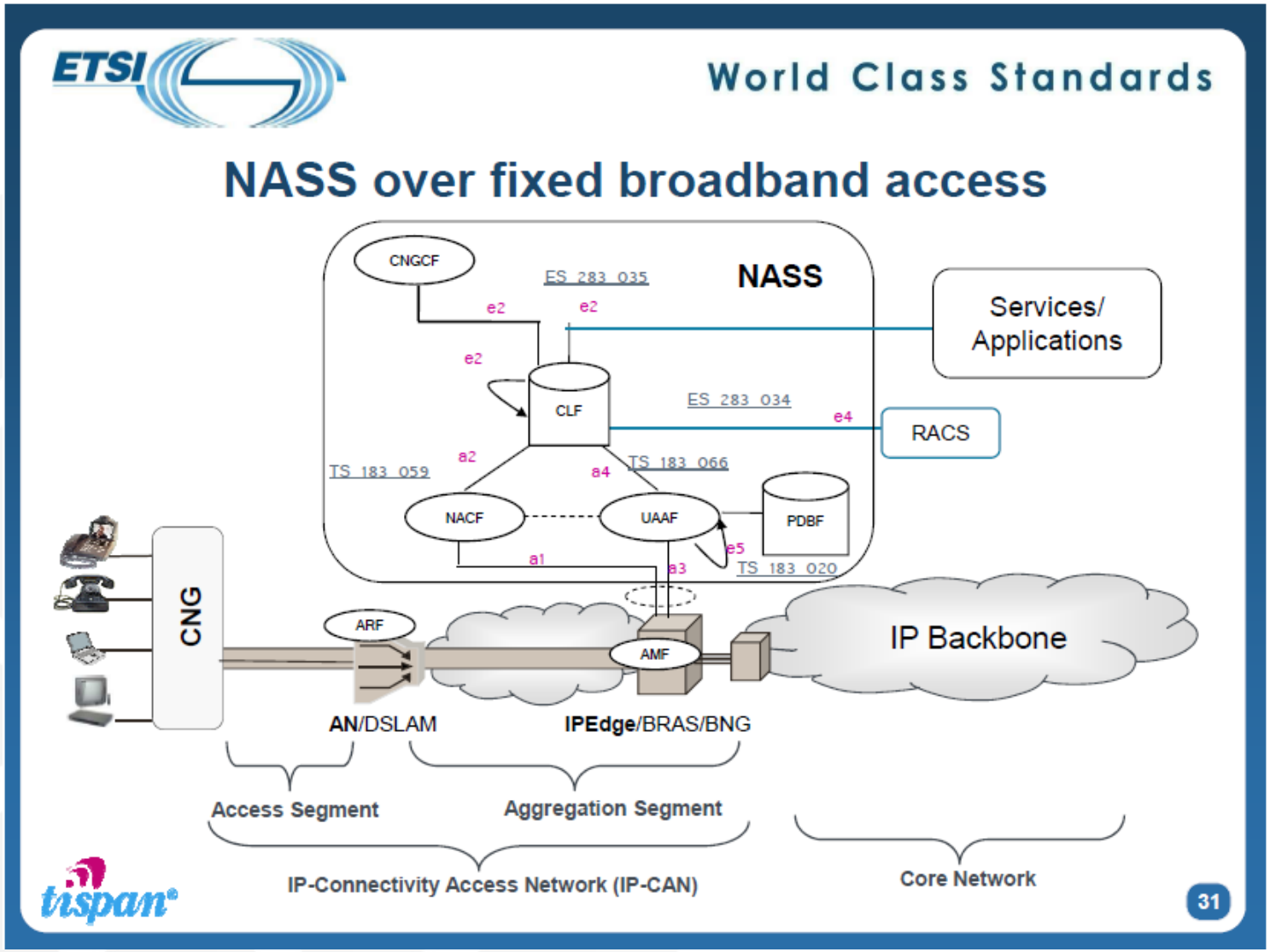
travelping

- Not very Successful as **RADIUS** successor. RADIUS is still widely used in ISP and Enterprise Networks.

- Widely used as Control Protocol in 3GPP defined IMS Networks.

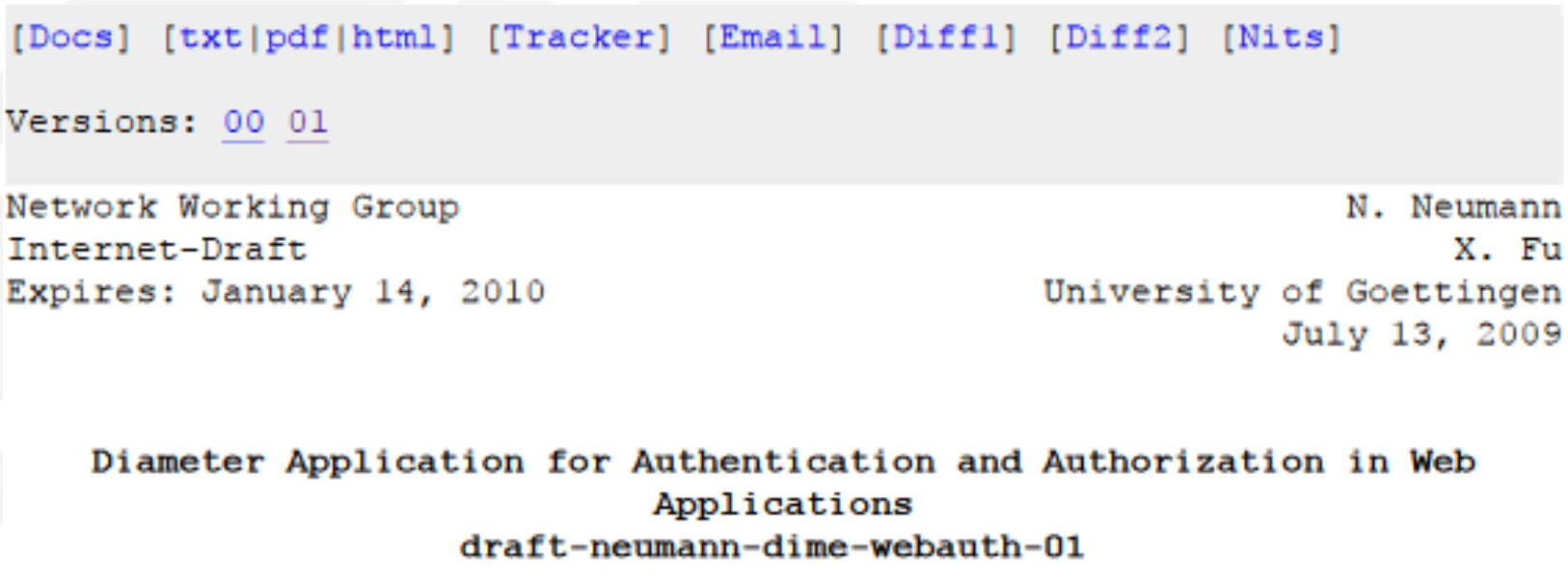
- Gets adopted by ETSI TISPAN for all IP Migration of fixed line Telco Networks (NGN)

NGN - TISPAN Interfaces



DIAMTER Market

- Application Developer can connect Applications to IMS based Networks. (i.e. Location based Services)
- Opensource Erlang DIAMETER Stack can be used for Testing.
- The Web get‘s a DIAMETER Application:



[Docs] [txt|pdf|html] [Tracker] [Email] [Diff1] [Diff2] [Nits]

Versions: 00 01

Network Working Group — N. Neumann
Internet-Draft — X. Fu
Expires: January 14, 2010 — University of Goettingen
July 13, 2009

Diameter Application for Authentication and Authorization in Web Applications
draft-neumann-dime-webauth-01

travelping

Handover to Ulf Wiger