



Erlang Training and Consulting Ltd

Styling your Architecture in an Evolving Concurrent World

JA00 Aarhus, October 5th, 2009

Francesco Cesarini

francesco@erlang-consulting.com

@FrancescoC

Erlang Forces a Mental Adjustment

Tim Bray, Director of Web Technologies - Sun Microsystems:

```
counter_loop(Count) ->  
  receive  
    increment ->  
      counter_loop(Count + 1);  
  {count, To} ->  
    To ! {count, Count},  
    counter_loop(Count)  
  end.
```

Erlang



Erlang Forces a Mental Adjustment

After you've opened the top of your head, reached in and turned your brain inside out, this starts to look like a natural way to count integers. And Erlang does require some fairly serious mental readjustment.

Erlang Forces a Mental Adjustment

After you've opened the top of your head, reached in and turned your brain inside out, this starts to look like a natural way to count integers. And Erlang does require some fairly serious mental readjustment.

However having spent some time playing with this, I tell you...

Erlang Forces a Mental Adjustment

If somebody came to me and wanted to pay me a lot of money to build a large scale message handling system that really had to be up all the time, could never afford to go down for years at the time, I would unhesitatingly choose Erlang to build it in.

Tim Bray, OSCON 2008

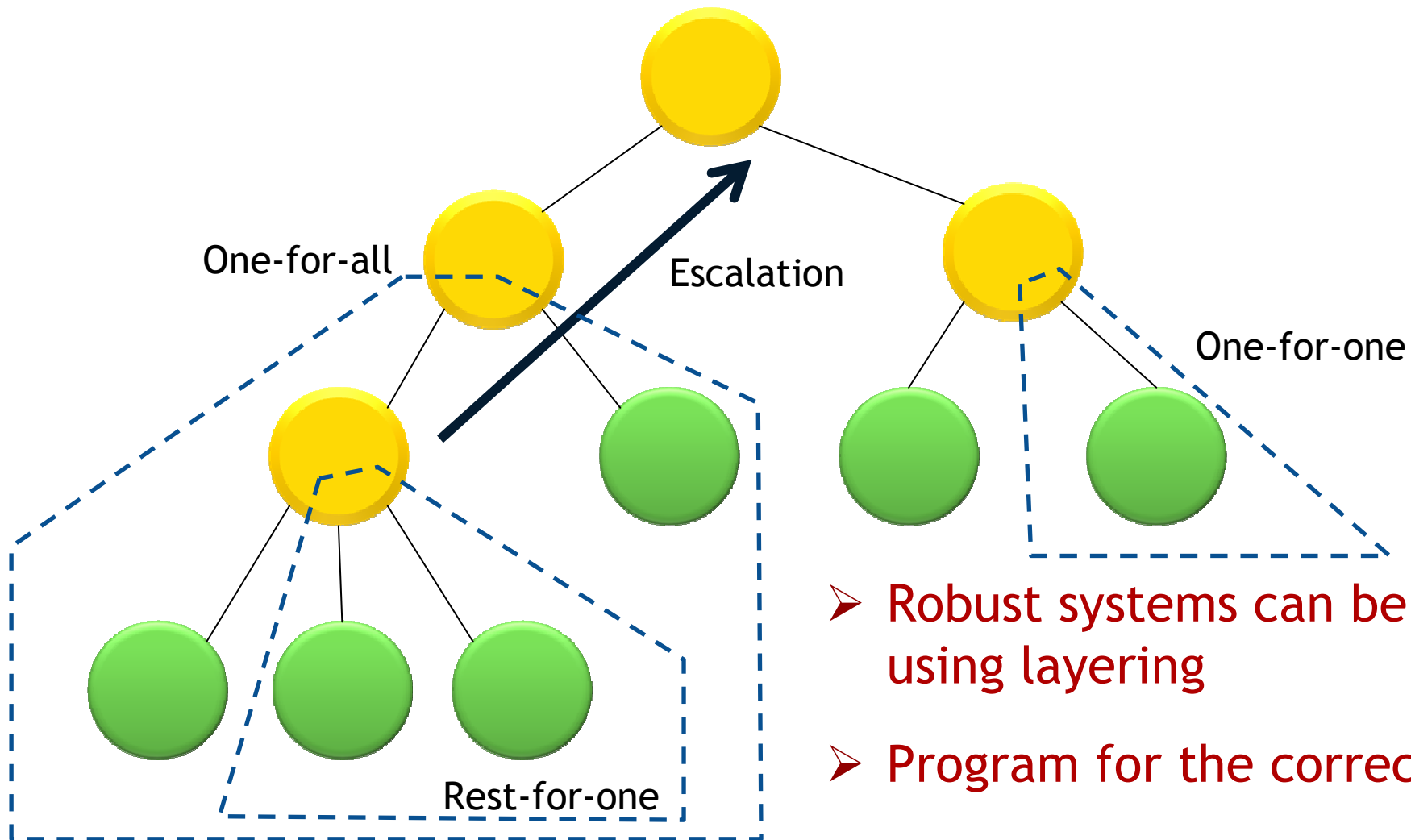
Program for the correct case - Patterns

```
factorial(N) when is_integer(N), N > 0 ->  
    N * factorial(N-1);  
factorial(0) ->  
    1.
```

```
area({square    , Side}) -> Side * Side;  
area({rectangle, B, H}) -> B * H;  
area({triangle , B, H}) -> B * H / 2.
```

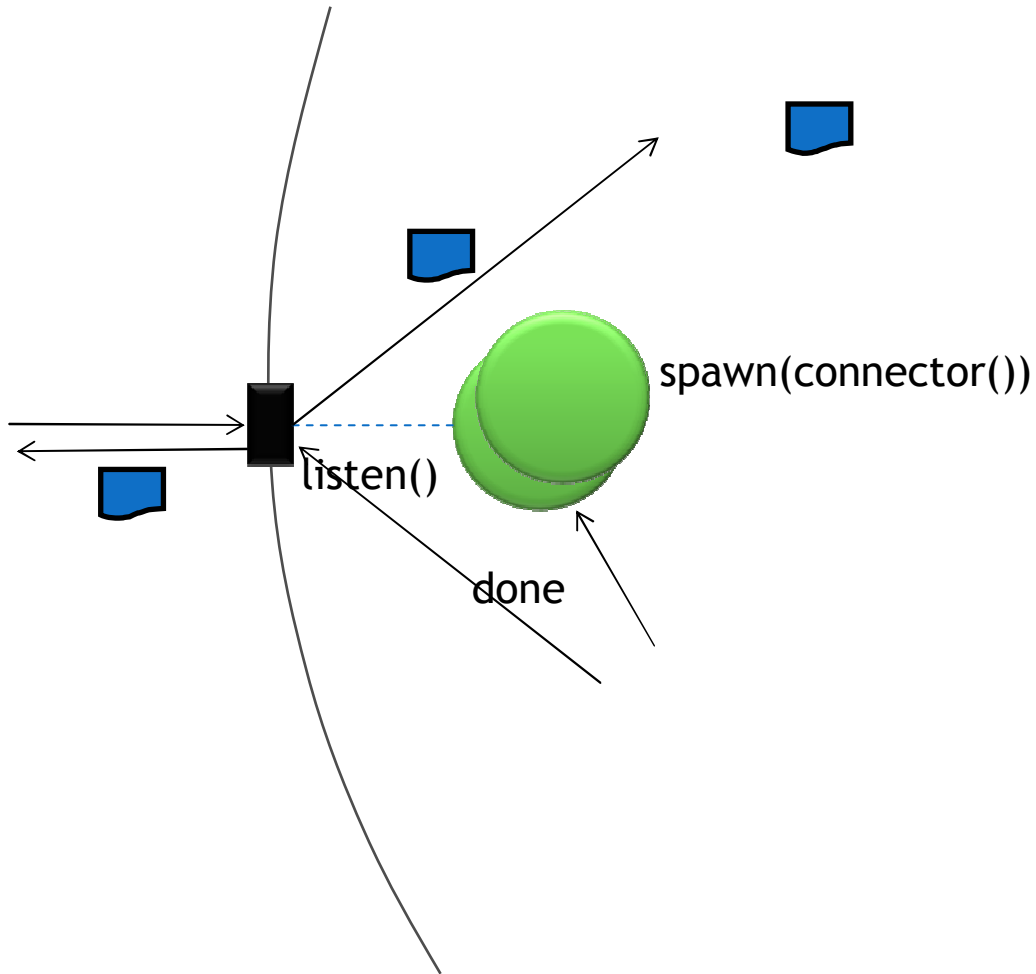
- Describe the expected - crash on erroneous input
- Infrastructure handles recovery

Program for the correct case - Supervisors



- Robust systems can be built using layering
- Program for the correct case

Handling sockets in Erlang



Static process opens listen socket

Spawns an acceptor process

Acceptor receives incoming

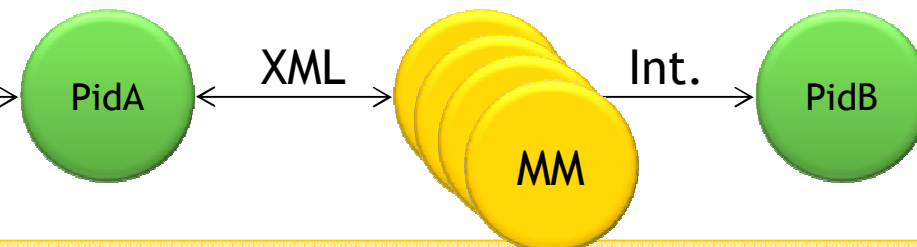
Acks back to socket owner

New acceptor is spawned

Replies sent directly to socket

Middle-man Processes

```
spawn_link(PidA, PidB) ->  
  spawn_link(fun() ->  
    loop(#state{a_pid= PidA,  
              b_pid = PidB})  
    end).
```



```
loop(#state{a_pid = PidA, b_pid = PidB} = State) ->  
  receive  
    {PidA, MsgBin} when is_binary(MsgBin) ->  
      {simple_xml, _} = Msg = vccXml:simple_xml(MsgBin),  
      PidB ! {self(), Msg},  
      loop(State);  
    {PidB, {simple_xml, _} = Msg} ->  
      Bin = vccXml:to_XML(Msg),  
      PidA ! {self(), Bin},  
      loop(State)  
  end.
```

```
await_negotiation(State) ->  
  receive  
    {From,  
     {simple_xml,  
      [{"offer", Attrs, Content}]}} ->  
      HisOffer =  
        inspect_offer(Attrs, Content),  
      Offer = calc_offer(HisOffer, State),  
      From ! {self(), Offer};  
    ...  
  end.
```

- Possible because of light-weight concurrency
- Normalizes messages
- Main process can pattern-match on messages
- Keeps the main logic clear

Erlang Bends Your Mind...

- **Processes are cheap and plentiful!**
 - When you need a process - just create one!
 - Don't ration processes - use exactly as many as you need
 - No need for thread pools - reusing processes is really a pain!
- **Message-passing is cheap!**
 - Use processes to separate concerns
 - Middle-man processes useful for transforming data
- **Processes can monitor each other**
 - Enables out-of-band error handling
- **Use Concurrency as a Modelling Paradigm!**

Language Model Affects our Thinking

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-Snd-DWA	I-Open
	I-Rcv-DWA	Process-DWA	I-Open
	R-Conn-CER	R-Reject	I-Open
	Stop	I-Snd-DPR	Closing
...			

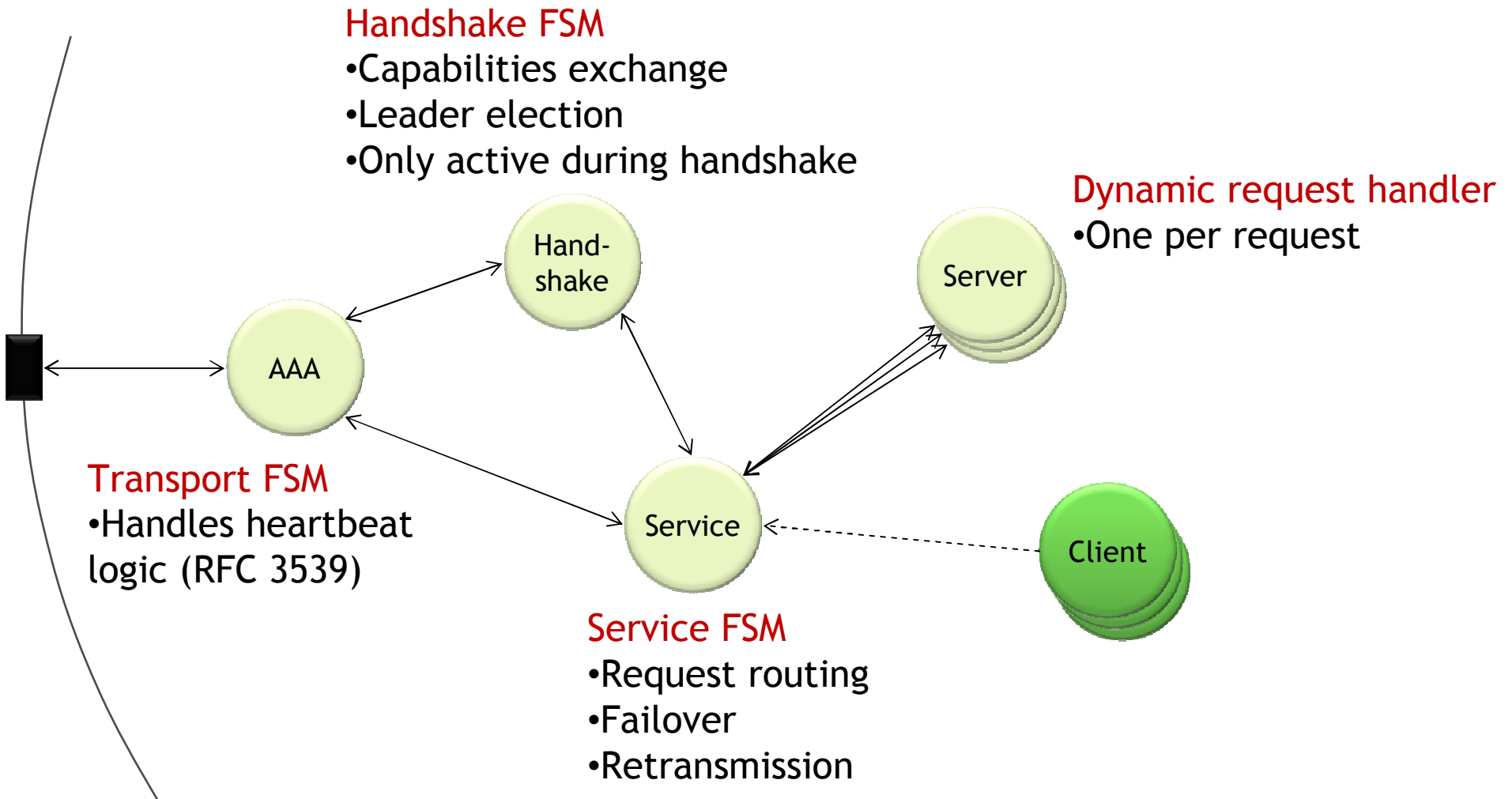
Transport FSM

Handshake FSMc

Service FSMc

- Three state machines described as one
- Implies a single-threaded event loop
- Feels wrong to an Erlang programmer

DIAMETER, Erlang-Style



What is the AXD301 Switch

A Telephony-Class, scalable (10 - 160 GBps) ATM switch

Designed from scratch in less than 3 years back in **1996**

Erlang: ca 1.5 million lines of code

- Nearly all the complex control logic
- Operation & Maintenance
- Web server and runtime HTML/ JavaScript generation



AXD301 Concurrency Modeling

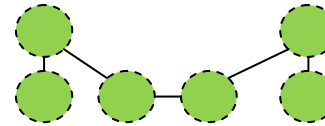
Model for the natural concurrency in your problem

In the old days, processes were a critical resource

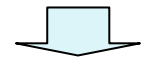
- Rationing processes led to complex and unmanageable code

Nowadays, processes are very cheap: if you need a process - create one!

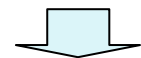
Example: AXD301 process model



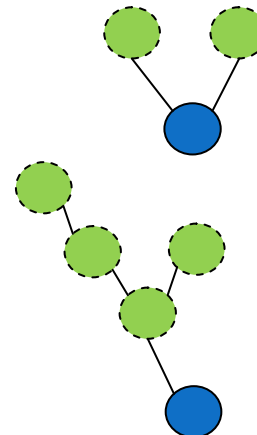
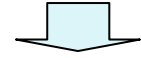
1st prototype:
6 processes/call



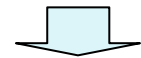
2 processes/call



1 process/all calls



2 processes/
call transaction

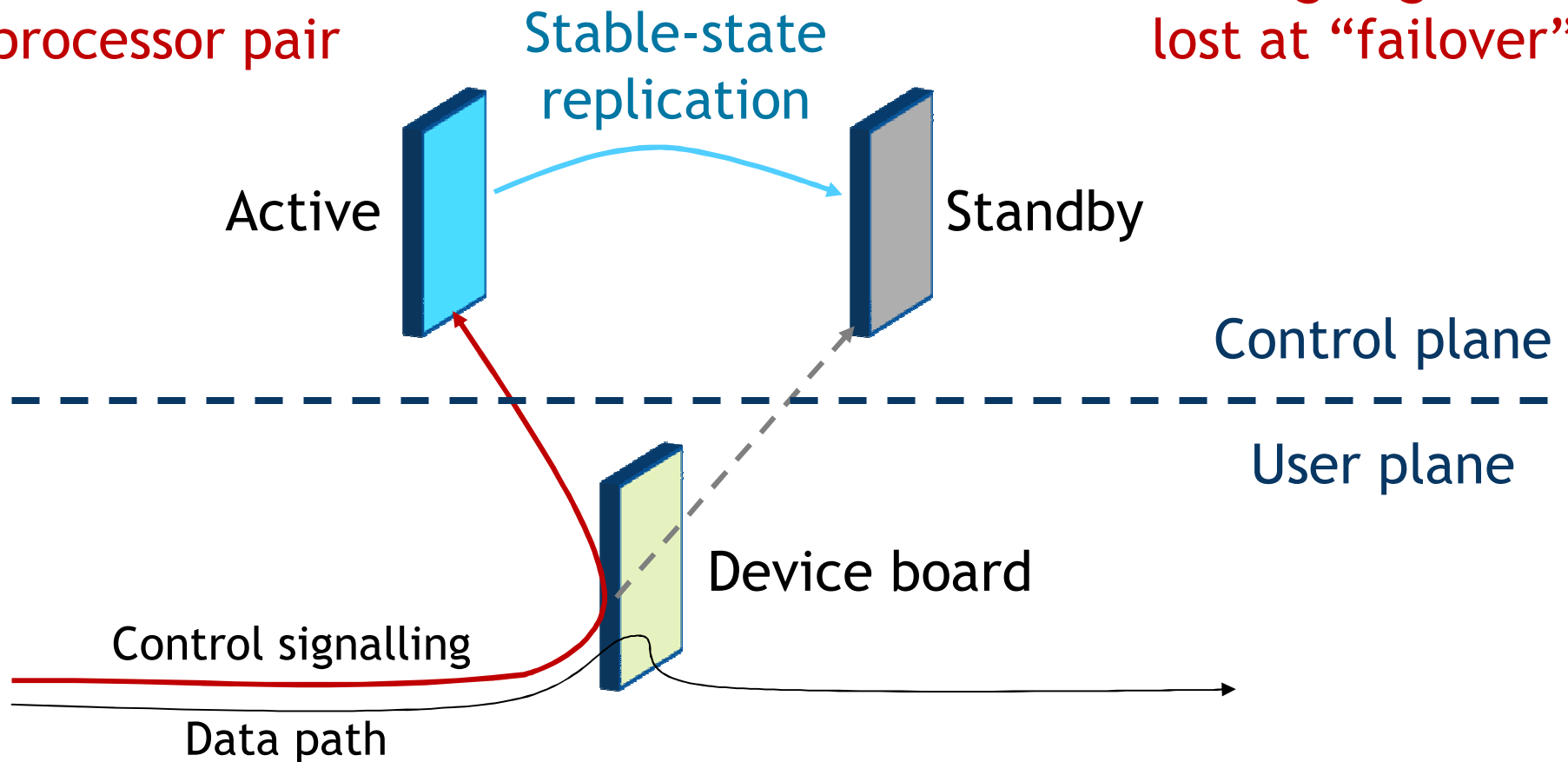


4-5 processes/
call transaction

AXD301: 1+1 Redundancy - Good ol' Telecoms

~ 70 000 sessions
per processor pair

No ongoing sessions
lost at "failover"

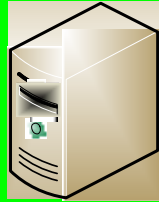


Today: 2-Tier and 3-Tier architecture

Client Interface

RESTful APIs
XML Handling
JSON Encoding/Decoding
SMPP Handling
HTTP Servers
Socket Handling

Transaction



Server Connectivity

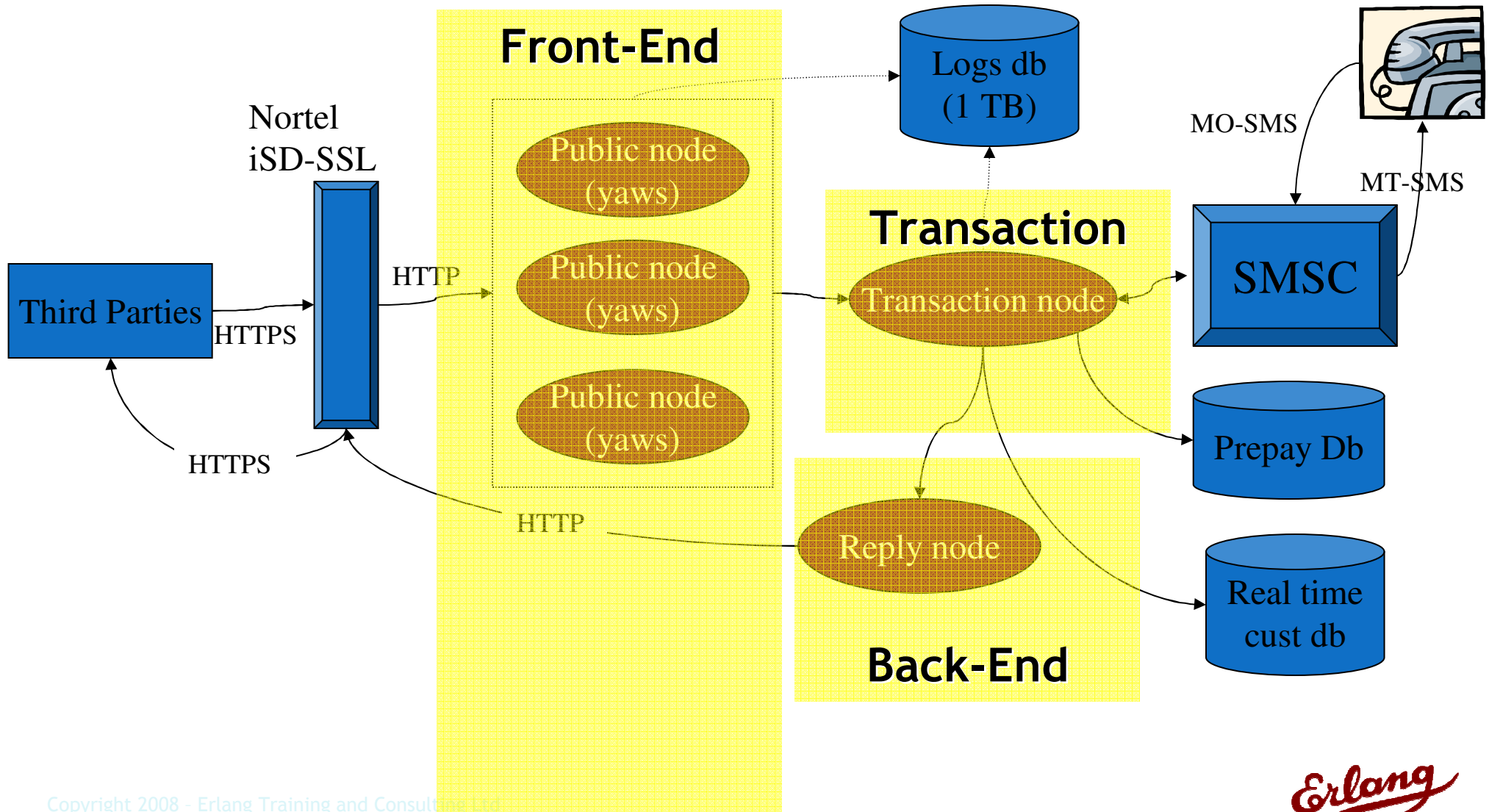
Remote Banking Systems
Mobile Operators
IM Providers
Email Servers

What is the Third Party Gateway?

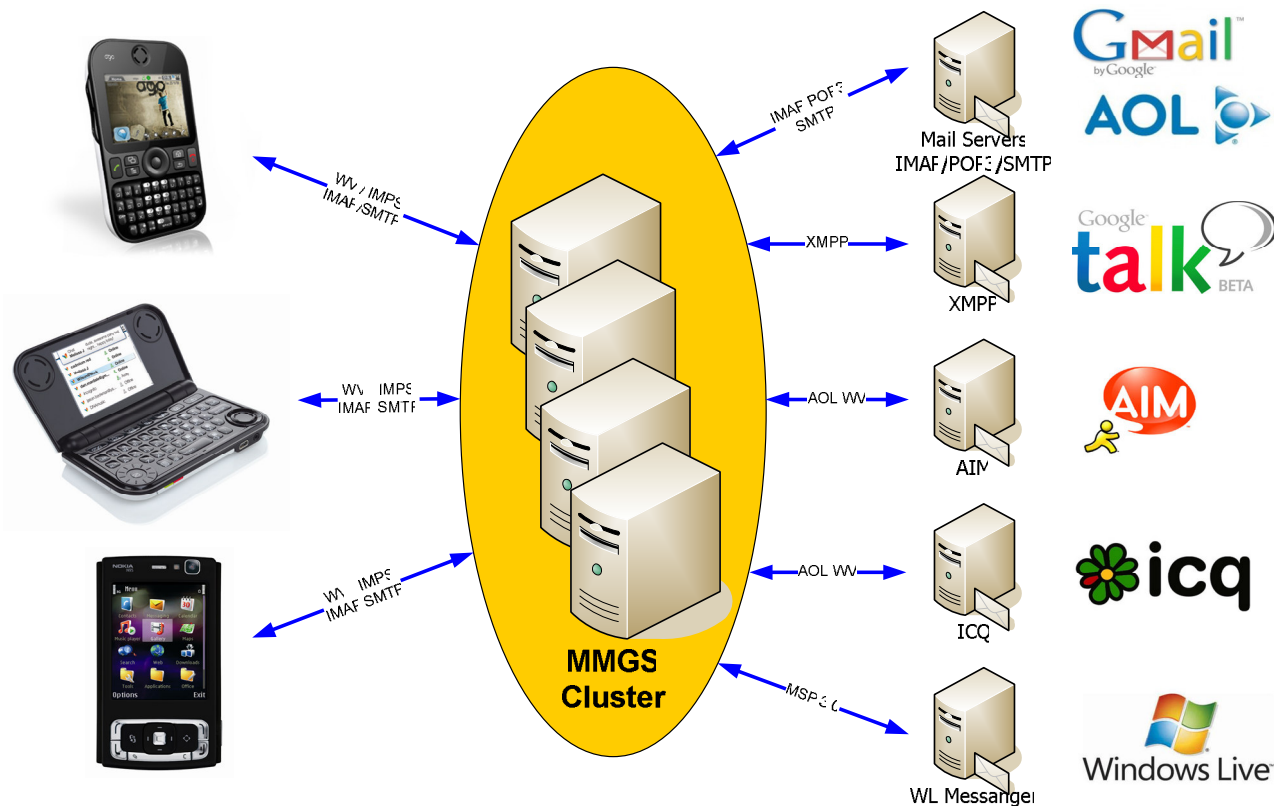
- **Third Party Bulk Mobile Terminating SMS**
 - Spam, Free traffic updates, etc
- **Premium Mobile Terminating SMS**
 - Ring tones, stock quotes, chargeable SMSes, etc
- **Receive Mobile Originating SMS**
 - Short Codes, TV Votes, etc

· · **T** · · Mobile ·

TPG: No Shared Data



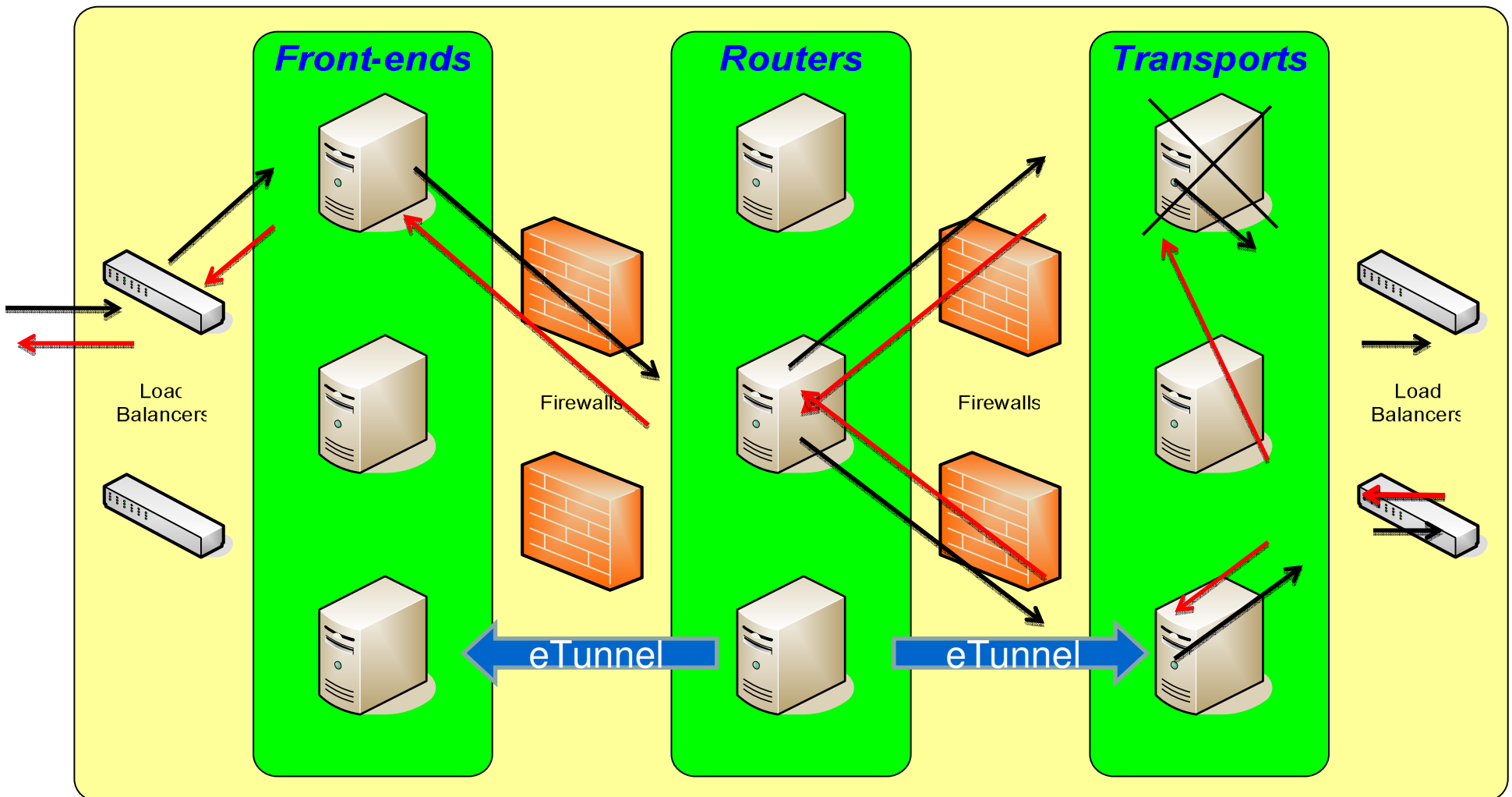
What is the Mobile Messaging Gateway?



➤ Mobile Instant Messaging and Email Gateway

➤ Connecting to public and private communities

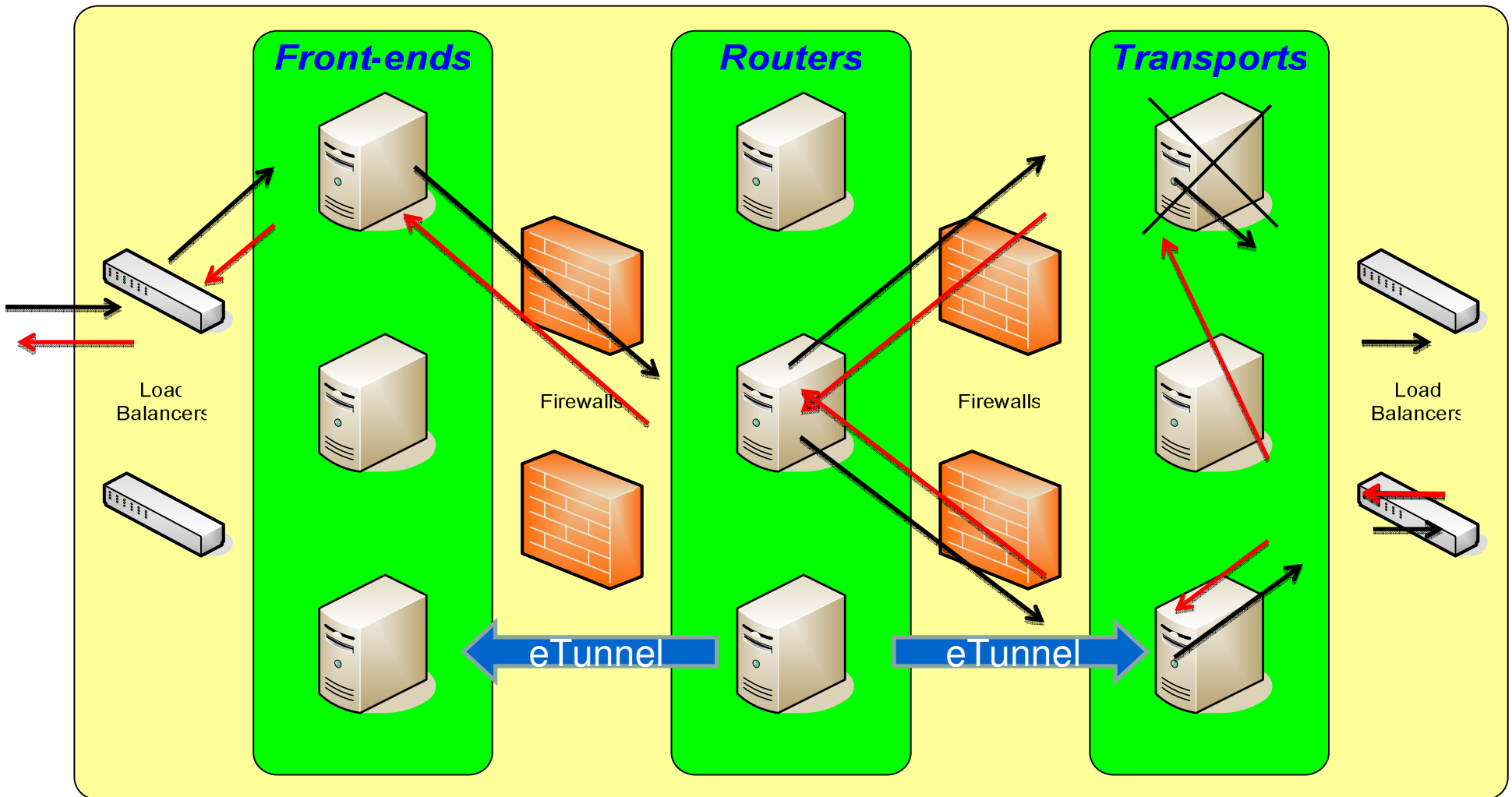
MMGS: Fully Replicated Data



MMGS: Fully Replicated Data

- 100,000 connected users, 3,000 IM Transactions Per Second
- 1 Transaction consisted of
 - 3 HTTP Requests
 - 2 DB entries distributed across the cluster
 - 3 destructive DB operations using Mnesia
 - (Replicated on 3 nodes = 9 destructive Operations)
 - 10 audit log entries written to file
- **Overloaded before reaching 100% CUP**
 - I/O Starvation
 - TCP/IP congestion
 - Mnesia Replicated Database

MMGS: Minimal Shared Data



MMGS: Minimal Shared Data

- 150,000 connected users, 12,600 Transactions Per Second
- Sustained loads of 45,000,000 messages per hour
 - 24 hour periods
 - 60% CPU utilization

	Total Shared Data	Minimal Shared Data
Connected Users	150,000	150,000
Throughput	3,000	12,600

MMGS: No Shared Data

- 50,000 connected users
- 4,200 Transactions/Second
- On a cluster 1/3 the size.
- Linearly Scalable
- Login throttle on fail-over and take-over

	Total Shared Data	Minimal Shared Data	No Shared Data
Connected Users	150,000	150,000	150,000
Throughput	3,000	12,600	12,600

The Cloud, to us old-timers

- **Software as a Service**
 - Access program and data from anywhere, using any device
- **Hardware as a Service**
 - Access computing resources as-needed, without owning a data centre
- **Virtualization**
- **“Resolving the tensions between the end-user and the data centre”**
 - Power vs. Accessibility
 - Powerful clients vs. Ease of deployment
 - (Google VP Vic Gundotra @ [Google I/O Keynote 2008](#))

The Cloud, to us old-timers

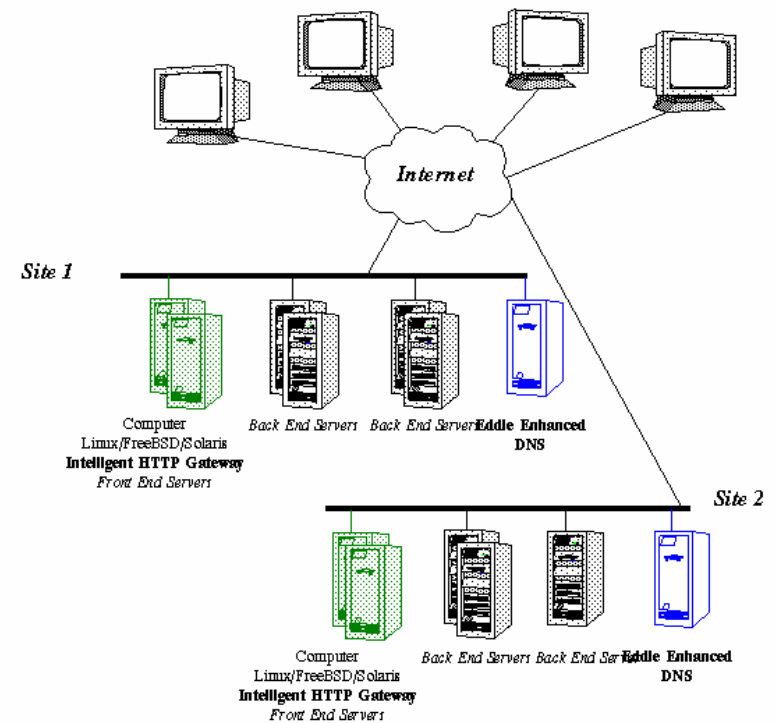
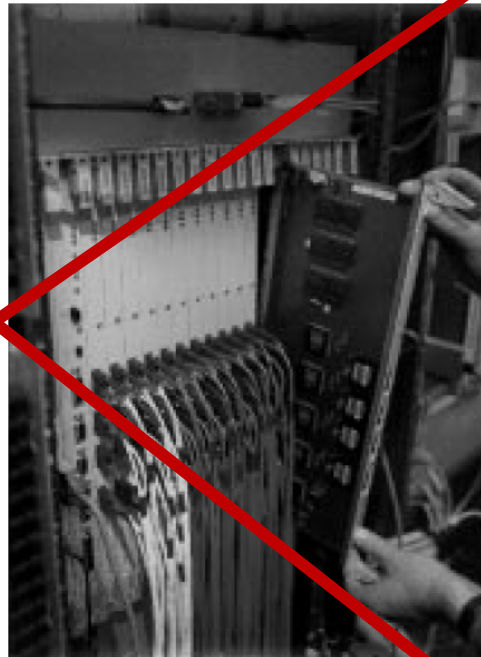
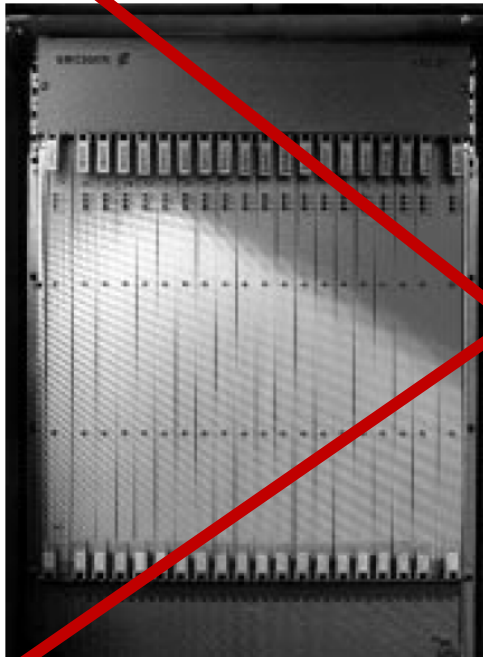


Figure 3. Distributed Web Server using the Eddie Enhanced DNS and Intelligent HTTP Gateway packages.

Eddie - An Ericsson-sponsored Open Source web server cluster framework 1999

The Cloud infrastructure is Erlang...

Communication In The Cloud

- RabbitMQ
- EjabberD
- Disco Project
- Nanite
- Proprietary TCP/IP Middleware
- Distributed Erlang

Data duplication in the Cloud

- Mnesia
- CouchDB
- Dynamite
- Simple DB
- Scalaris
- Riak

The Cloud infrastructure is Erlang...

From Computer Desktop Encyclopedia
© 1998 The Computer Language Co. Inc.

Are we there yet?



More Information



Erlang Programming

Open Source Erlang

Google Search

WWW www.erlang.org www.erlang.se

[July 4 2008] Seventh ACM SIGPLAN Erlang Workshop

The Seventh ACM SIGPLAN Erlang Workshop will take place on September 27, 2008 in connection with the ICFP (International Conference on Functional Programming) in Victoria, British Columbia, Canada. The workshop program is now settled and can be viewed [here](#).

[July 4 2008] 14th International Erlang/OTP User Conference (EUC'08)

The 14th International Erlang/OTP User Conference will be held in Stockholm on November 13, 2008. See the call for presentations [here](#).

[June 11 2008] Erlang/OTP R12B-3 released

The third service release for Erlang/OTP R12B has been released. (June 11, 2008)

Erlang book!

Programming Erlang, Software for a Concurrent World
This is the most recent book about Erlang (June 2007) but the
For more information, look [here](#).

Last updated: 2008-07-04 08:45 UTC

www.erlang.org

The 2009 Erlang Factory is coming to the SF Bay Area!

University: April 27th - April 29th
Conferences: April 30th - May 1st

The Erlang Factory comes to the West Coast! For those interested in learning more about Erlang to expert programmers alike, this promises to be the largest gathering of Erlang experts from last year's exchange in London. Running together with the Factory is the Erlang University - 3-day courses so you can combine a training course, talks and tutorials in the same week. Come and meet Erlang inventors and experts who have been using the language long before it was released as open source, network with committers of the open source applications or debate and discuss the latest features and fixes.

We are adding new speakers daily. Don't miss Robert Virding, one of the three inventors of Erlang, speaking on the Erlang rationale. Other confirmed speakers include Eira Zymarewicz, Kevin Heath, Michael Reesed and many more. Tutorial topics include tool and gadgets, Erlang and TDD, Erlang and IM, Relax with CouchDB and CoolApplications.

While attending the Erlang Factory, why not take the opportunity to attend our three day Erlang University? We will be offering courses in Erlang Express, OTP Express, Quick Check and CausalDB, giving you a free ticket for the conference Learn basics and advanced topics on the subject matter experts.

London Erlang Factory, June 2009

University: June 22nd - June 24th
Conferences: June 25th - June 26th

Following on from last year's successful Erlang exchange, this year's Erlang Factory talks and tutorials promises to be even better. As with the SF Bay Area event, there will be an Erlang University held for the three days at the beginning of the week. Courses will include Basic Erlang, OTP Design Patterns, Quick Check and Couch DB. The London Erlang Factory is scheduled for the last week in June.

Offers
Registration now LIVE!
We are pleased to announce bookings can now be made online. Full https security and PayPal ensure that your transaction is secure.
Book now for the London Factory by going to [Register for London](#).
Book now for the SF Bay Area Factory by clicking below:
[Register](#)

News
Due to the delay in getting the on-line registration and payment engine in place, we are extending the time for the very Early Bird admission discounts. These will now be extended to 15th March. Register and pay before 15th March to get \$200 off the normal price!
[More News...](#)

© 1999 - 2009 Erlang Training & Consulting Ltd.

www.erlang-factory.com



Thank You!