Erlang Training and Consulting Ltd

Erlang's Journey to the Clouds

IGT2009
World Summit of Cloud Computing
Shefayim, Israel
Dec 3, 2009

The Cloud, for 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 center
- "Resolving the tensions between the end-user and the data center"
  - Power vs Accessibility
  - Powerful clients vs Ease of deployment
  - (Google VP Vic Gundotra @ Google I O Keynote 2008)

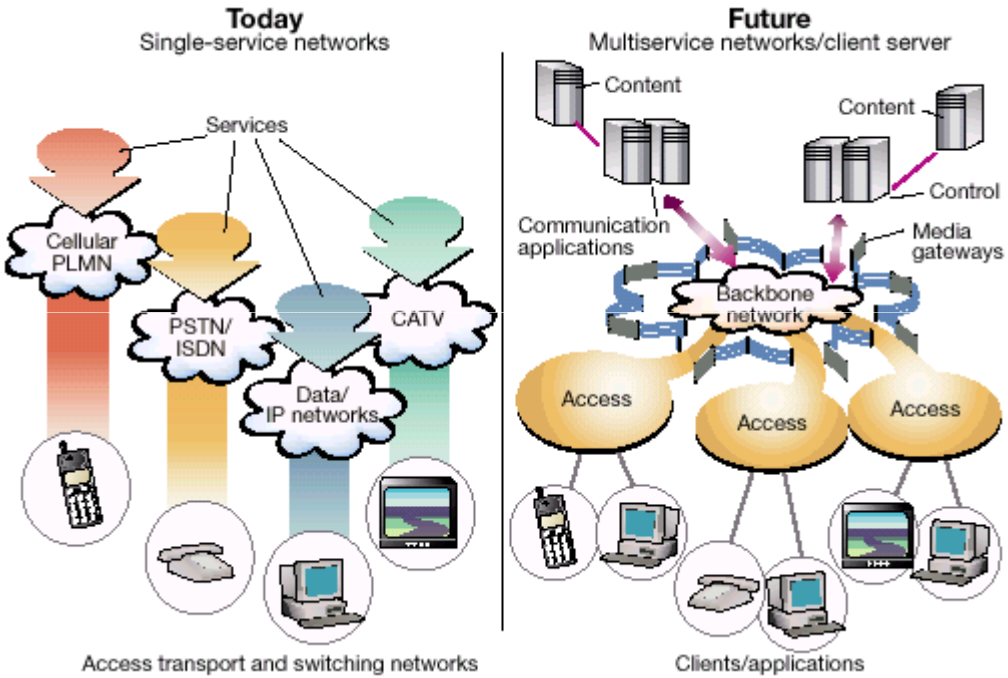
Erlang

Grey-bearded telecom guy says...

This looks awfully similar to the evolution in the telecoms world...

Clouds in 90s telecoms slides used to mean "stuff that we deliver"

Good cloud clip art passed around like trading cards



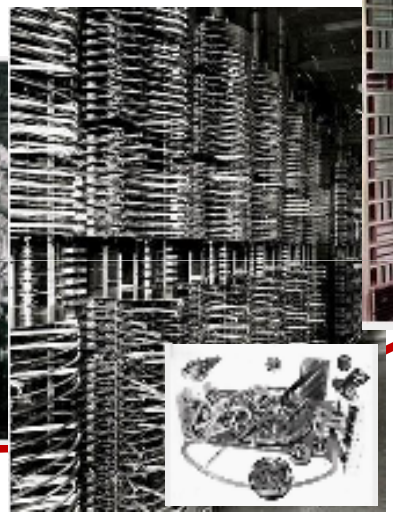
Source: Ericsson Review No 1, 1998

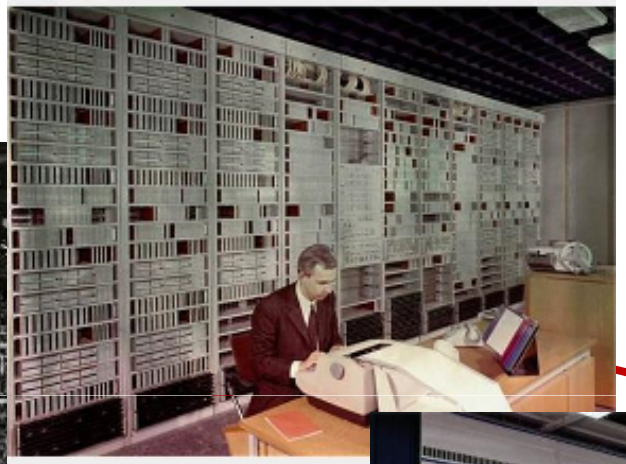
Erlang

The origins of Telecoms

1897

1940s





1968

1975

- Stored Program Controllers in the 60s made it possible to invent new features
- Feature interaction -> Complexity explosion

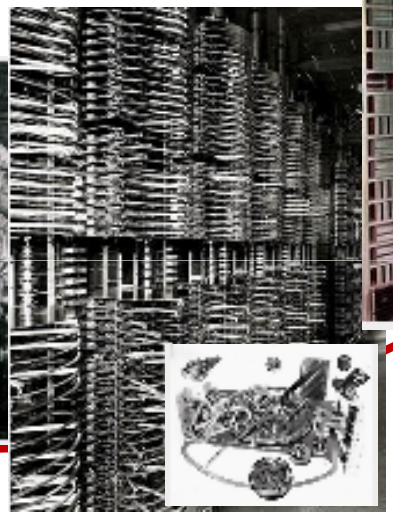
Erlang

The origins of Erlang

Addressing the growing software complexity

1897

1940s



1968

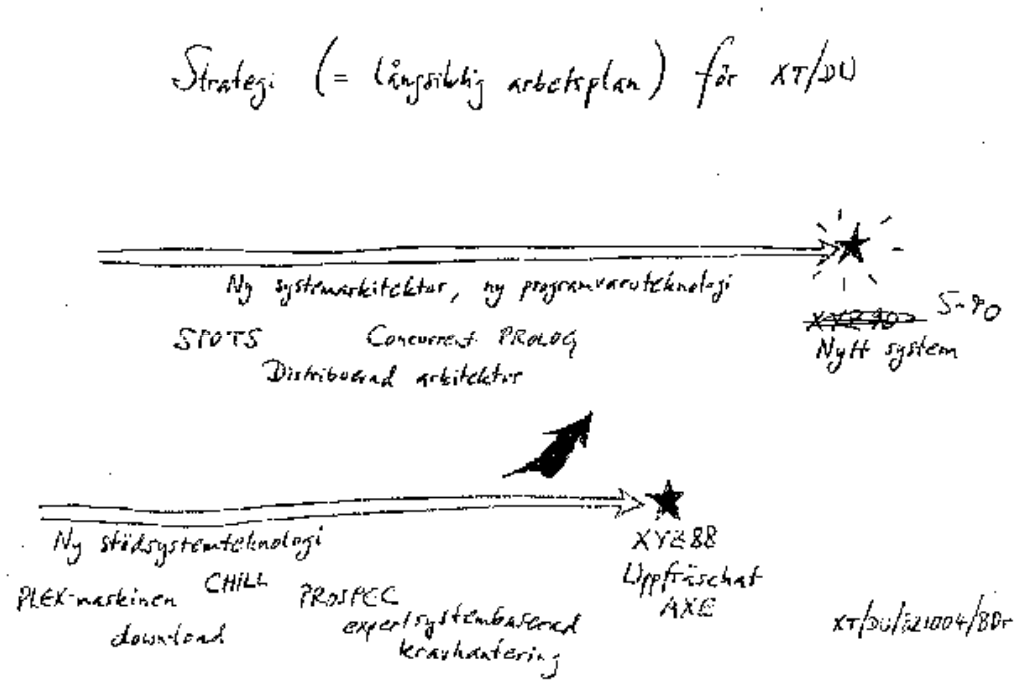
1975

1989

ERLANG

Erlang

The forming of Ericsson CSLab 1980



CSLab plan 1981 (Bjarne Däcker)

- Small group of people
  - Bjarne Däcker
  - Göran Båge
  - Seved Torstendahl
  - Mike Williams
- Systematic treatment of Computer Science
- Highly experimental, literally a ”laboratory”

Erlang

Language Experiments

http://video.google.com/videoplay?docid=-5830318882717959520#

- SPOTS (SPC for POTS)
- Wrote control system in several languages
  - Ada, CHILL, CCS, LPL, Concurrent Euclid, Frames, CLU, OPS4
- Domain experience identified the tricky problems
- Led to a new language: **Erlang**

Checkpoint

- **1958: First phone call completed using SPC technology**
  LISP
- 1965: Edsger Dijkstra - the first mutual exclusion algorithm
- **1970-72: Ericsson drafts the AXE/PLEX design**
- 1978: C.A.R. Hoare publishes CSP book
  Niklaus Wirth creates Modula-2
- 1982: Lamport et al describe The Byzantine Generals Problem
- **1981-1987: SPOTS Experiments ⇨ Erlang**
- 1991: Erlang publically announced

Erlang

Properties of Erlang

- Telecom goodness:
  - Scalable agent-style concurrency
  - Distribution transparency
  - Fail-fast programming style
- Managing complexity
  - Declarative/functional programming inside each process
  - No shared data, loosely coupled components (black-box style design)
  - ”Programming for the correct case”
- Evolving systems
  - In-service upgrades
  - (Dynamic typing)

Erlang

SO WHAT?

Telecom in the 90s

‘Stovepipe model’ on its way out

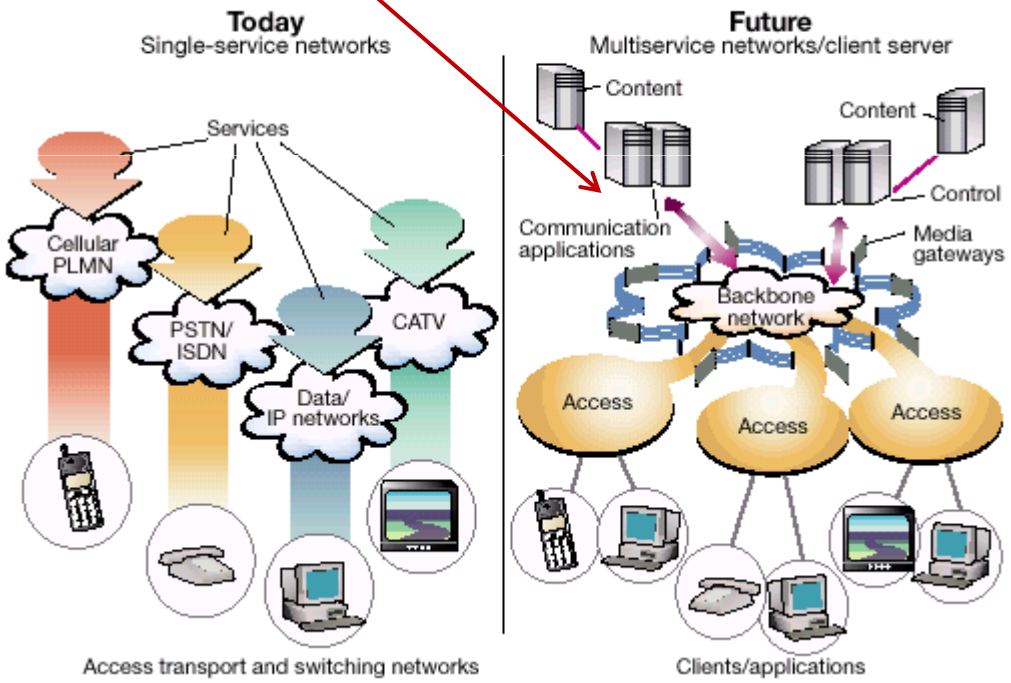
The network as a communications ‘cloud’

Broadband ISDN -> Voice over ATM -> Voice over IP

Today -> Mobile IP (IPv6)

“Conversational services”

Erlang Problem Statement (paraphrased):
“How can we program telephony in THIS environment?”



Source: Ericsson Review No 1, 1998

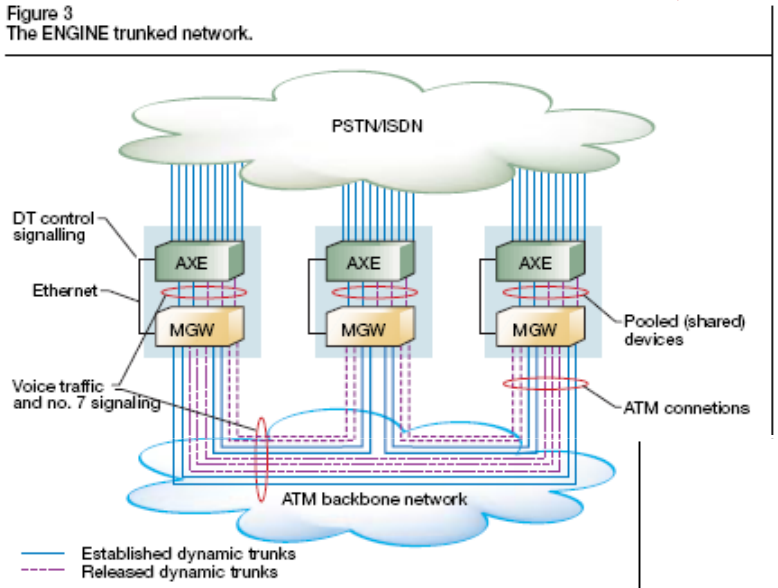
Bridging the legacy

- Ericsson had 40% of the world's telco infrastructure
- ...but the world was changing
- How to embrace VoIP without killing your cash cow?

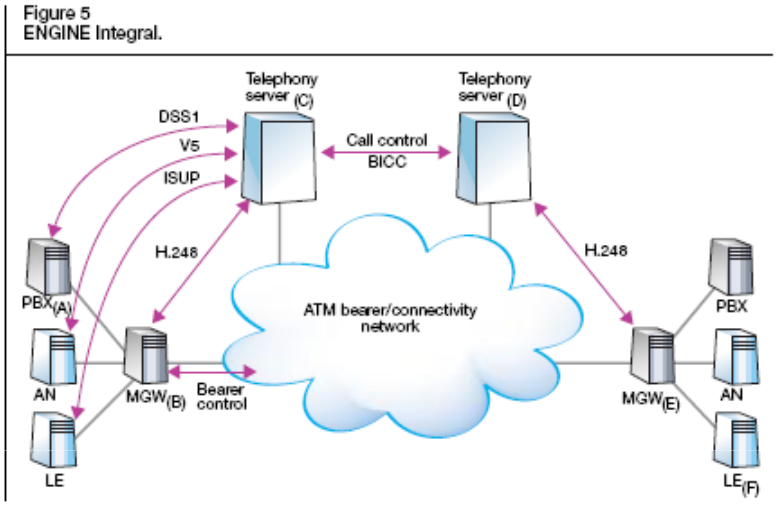
Erlang

Bridging the Legacy

Source: Ericsson Review No 3, 2000



Virtual voice trunks



Cross-domain Voice-over-Packet

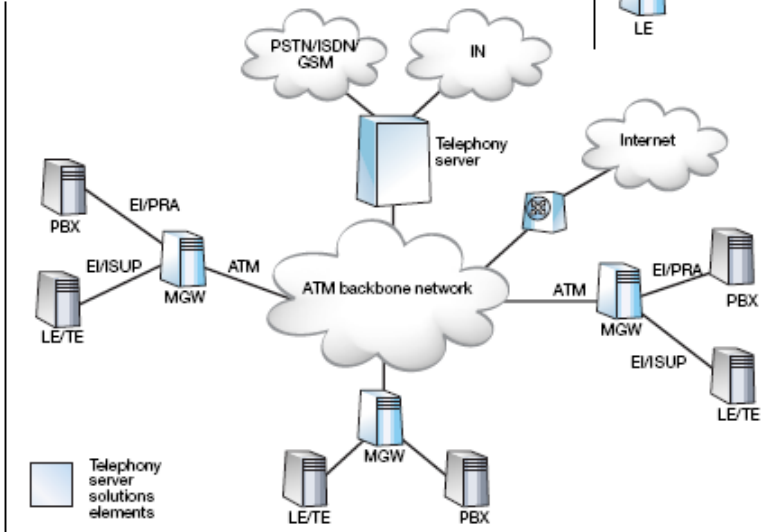
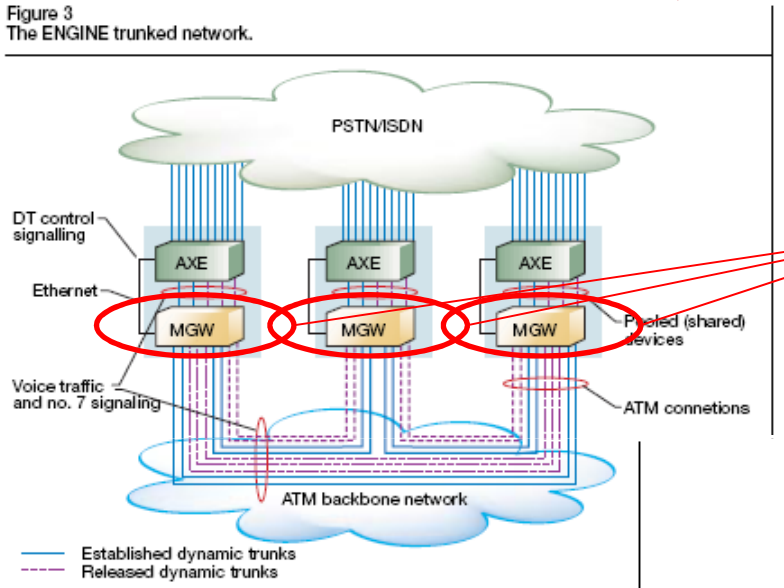


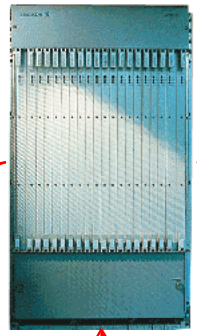
Figure 4
ENGINE Bridgehead.

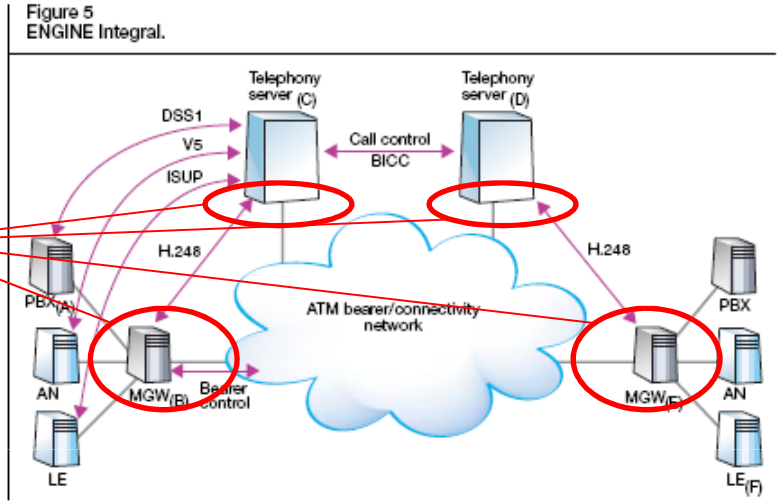
Single-Domain Voice-over-Packet

Erlang the Enabler

Source: Ericsson Review No 3, 2000







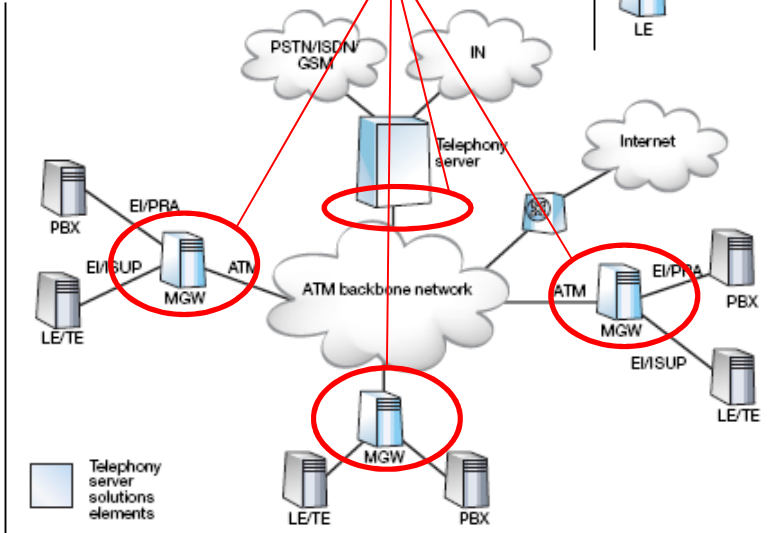
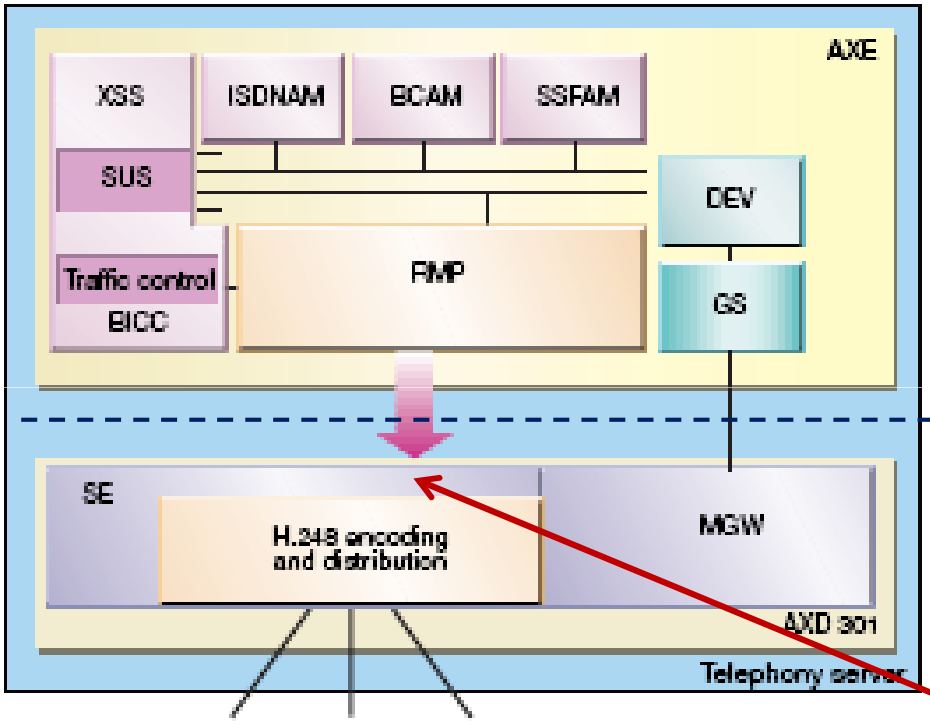


Figure 4
ENGINE Bridgehead.

Erlang the Enabler



Legacy Phone Switch

PLEX

Switch Emulator and
Voice-over-ATM Controller

Erlang

Extremely complex state machines
Scalability and redundancy required
> 99.999% (“5-nines”) uptime, including maintenance

So What Next...?

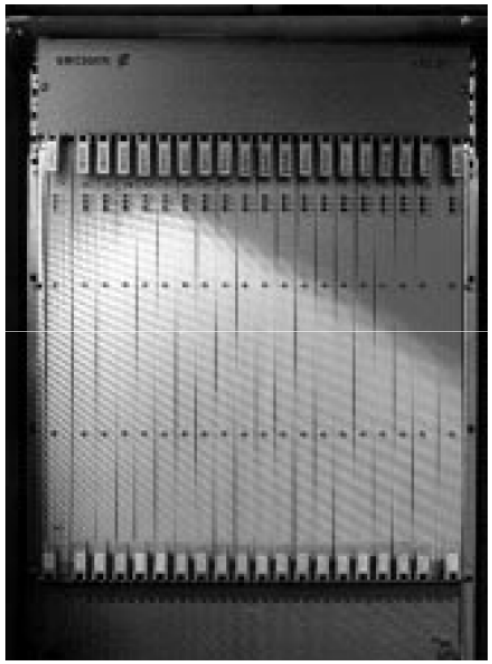
- The first big project worked out
- Erlang proved ready for the Big Time
- Erlang released as Open Source 1998
  - No fanfare, no marketing...
  - 1-2 messages/day on the mailing list first year
  - < 1000 downloads/month
- New initiatives needed...

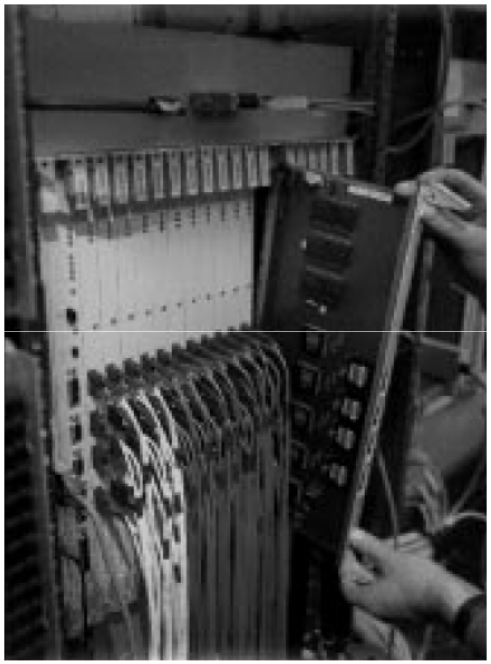
Erlang

Idea: A Scalable Web Server...?

Build a 5-nines scalable web server based on AXD 301

- 256 processor boards on a non-blocking, redundant 160 Gbps backbone!





Two Erlang- & Web-related Innovation Cell proposals presented at the same time

The AXD 301 track rejected – Ericsson doesn't sell web servers (and the servers then didn't need 5-nines...)

Erlang

Idea: A Scalable Web Server...?

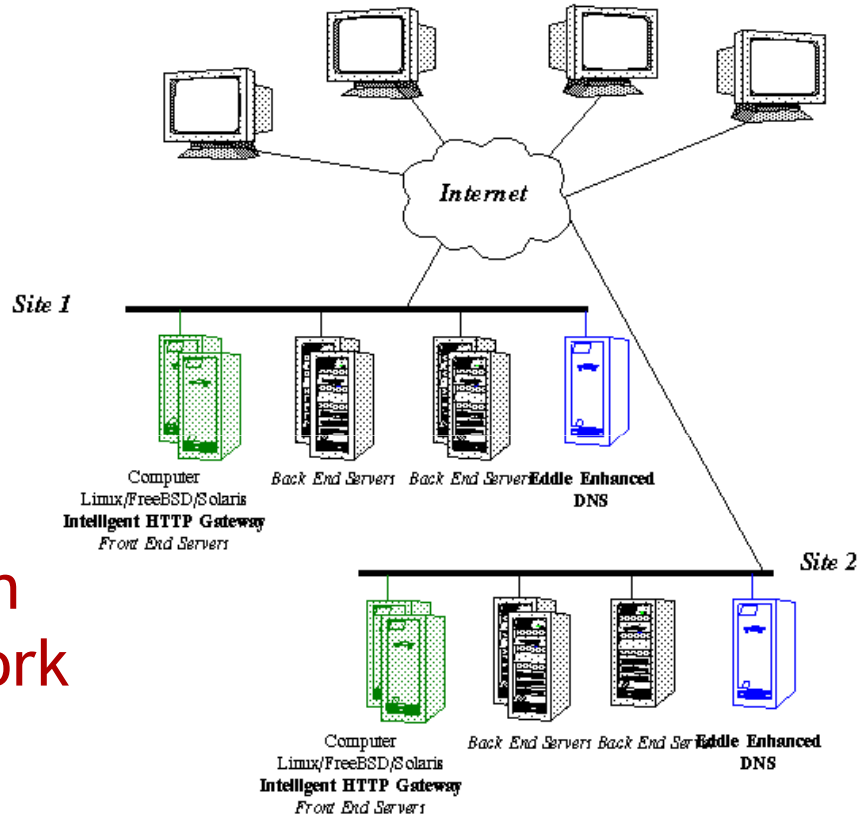


Figure 3. Distributed Web Server using the Eddie Enhanced DNS and Intelligent HTTP Gateway packages.

- Eddie – An Ericsson-sponsored Open Source web server cluster framework 1999
  - Dynamic load balancing
  - Auto-detects the capacity of each server
  - Works across wide-area networks

Erlang

Scalable Email - Bluetail Mail Robustifier

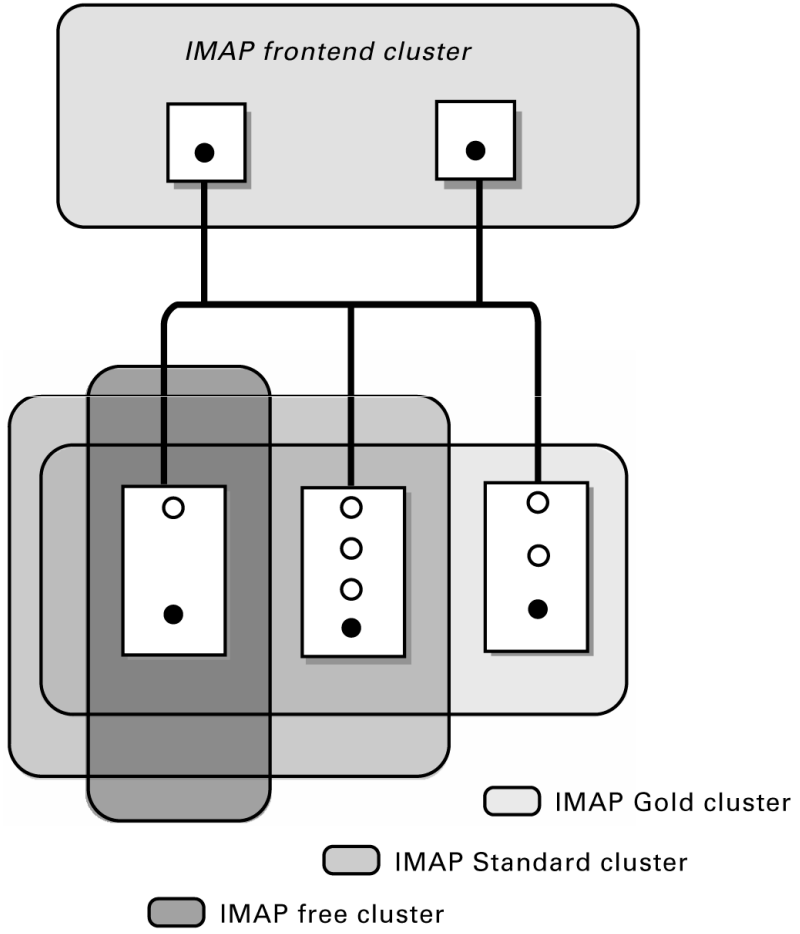
Load-balancing frontend to standard mail servers

Added

- Robustness
- In-service scalability
- Service differentiation

...transparently

Released 1999



Erlang

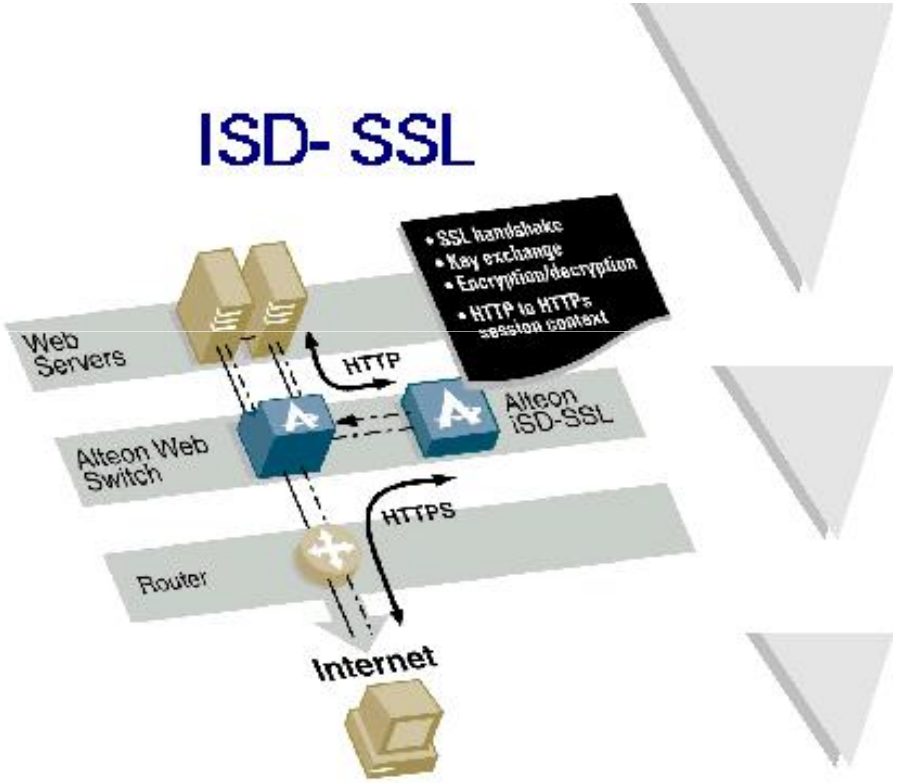
Scalable https – SSL Offload Accelerator

Bluetail bought by Alteon (Alteon bought by Nortel)

- (2009: Nortel/Alteon assets bought by Israeli Radeon)

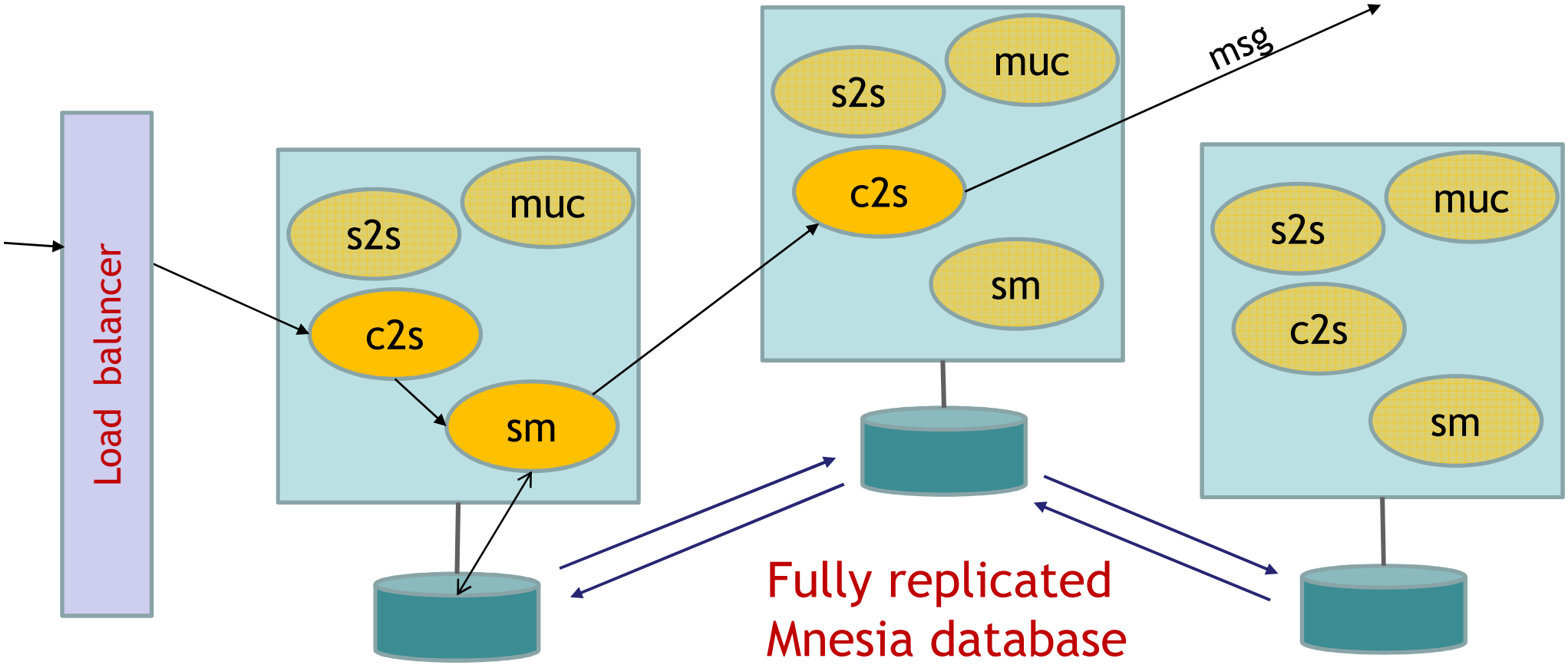
Continuing to make scalability solutions on commodity hardware

ISD-SSL released 2001



Erlang

Scalable XMPP Chat - ejabberd



First released 2003

Erlang

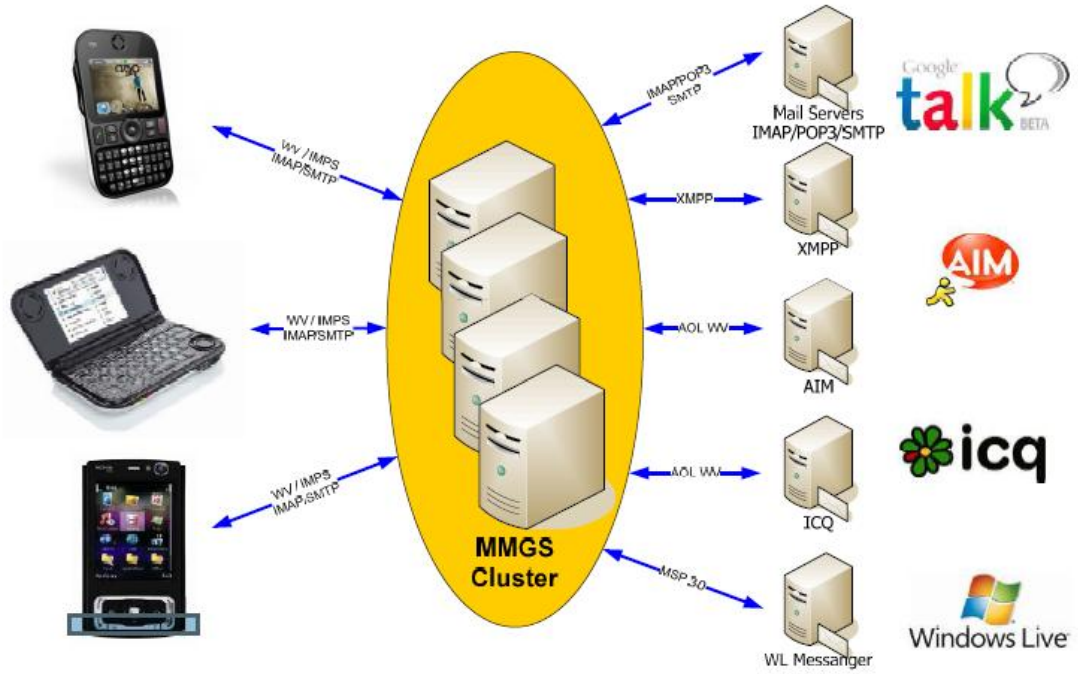
Runcom IXI MMGS – Email and IM Gateway

Massively scalable

>12k messages/second

150,000 connected users

Bridging different messaging standards



Erlang

Erlang, the un-Ruby

- Offering a cost-effective way to build...
  - Massively scalable
  - Extremely robust
  - Eminently maintainable

  ...back-end services (using an odd-looking syntax)
- **But organizations developing such systems are by nature conservative!**
- Perl ("duct tape"), Ruby and Python (OO scripting) offered something more immediately useful to individual programmers

Erlang

What Changed?

- **Web services matured** – started requiring scalability and serious uptime
- **Web 2.0** – opened up for a new class of (conversational) web services
- **Multicore** – forced everyone to start thinking about concurrency
- **Virtualization** – brought distributed systems development to the masses

Erlang

New wave – Messaging for the Cloud

- RabbitMQ – One of the most popular AMQP implementations
- BERT – Multi-language RPC framework made by Github

- ejabberd – Highly scalable XMPP server
- MMGS – Highly scalable IM gateway
- Facebook Chat – Chat supporting 400 million users...

Erlang

New wave – Databases for the Cloud

- Scalaris – Distributed Hash Tables
- CouchDB – RESTful Document Store
- Dynomite – Dynamo-like Distributed Key-Value Store
- Riak – Decentralized Key-Value Store w/map-reduce
- Disco – Map-Reduce framework

- Client versions for non-Erlang storage engines
  - MongoDB, TokyoCabinet, MySQL, BDB, ...

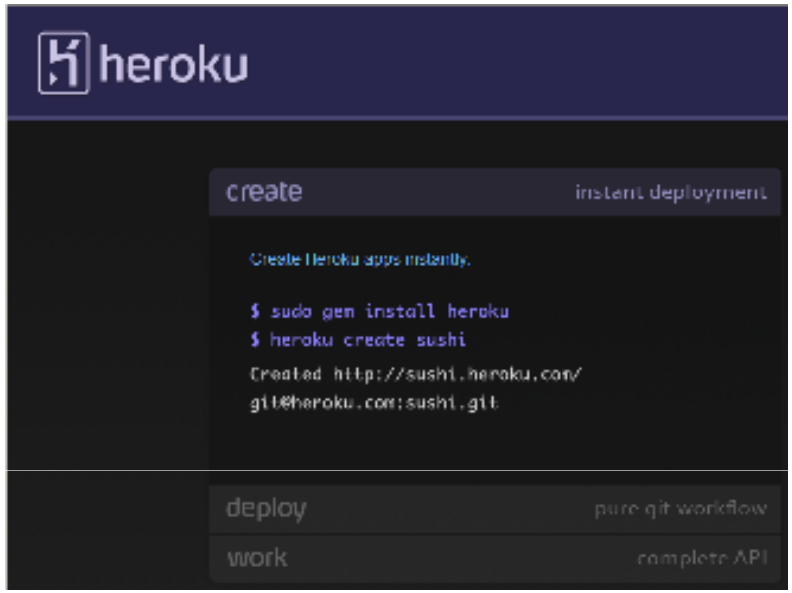
Erlang

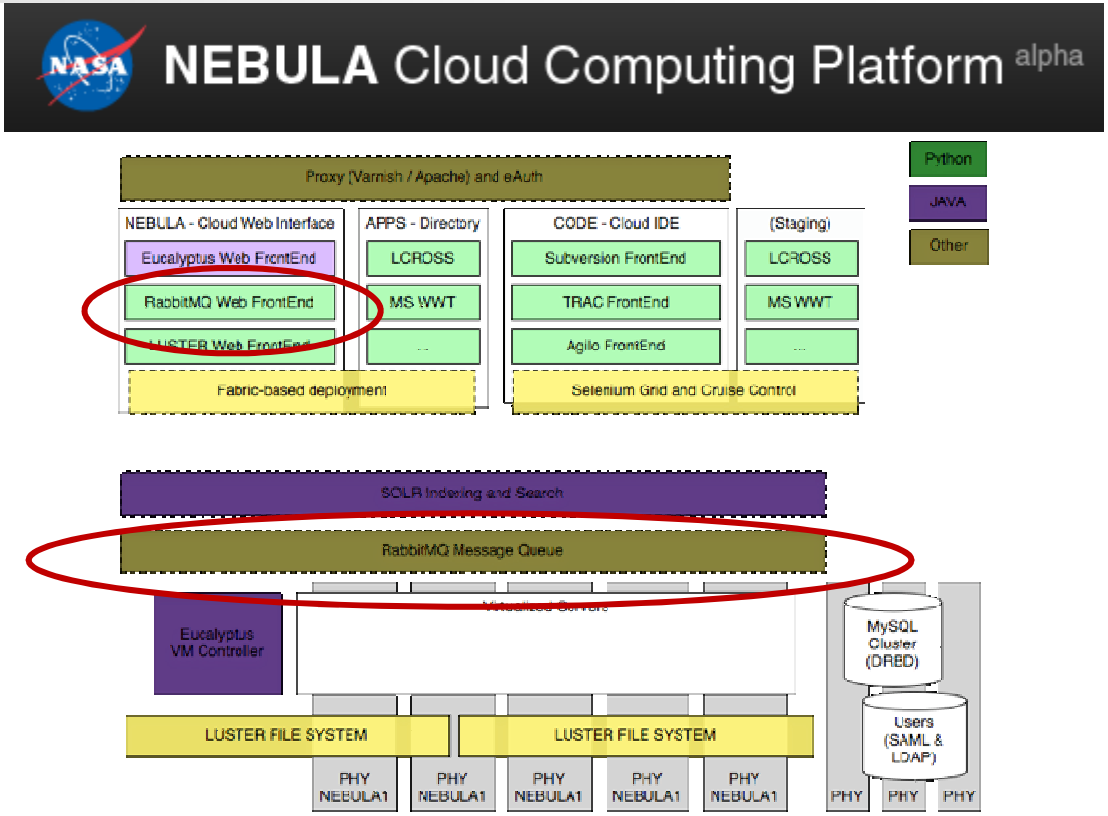
New wave – Web frameworks

- Yaws – fast dynamic-content web server
- MochiWeb – dynamic-content web server with JSON
- ErlyWeb – Web development framework
- Erlang Web – XHTML-based Web framework
- Nitrogen – Erlang-style JQuery
- WebMachine – RESTful Web services
- Chicago Boss – Django-style Web framework, but asynchronous

Erlang

New wave – Scalable Web/Cloud Services





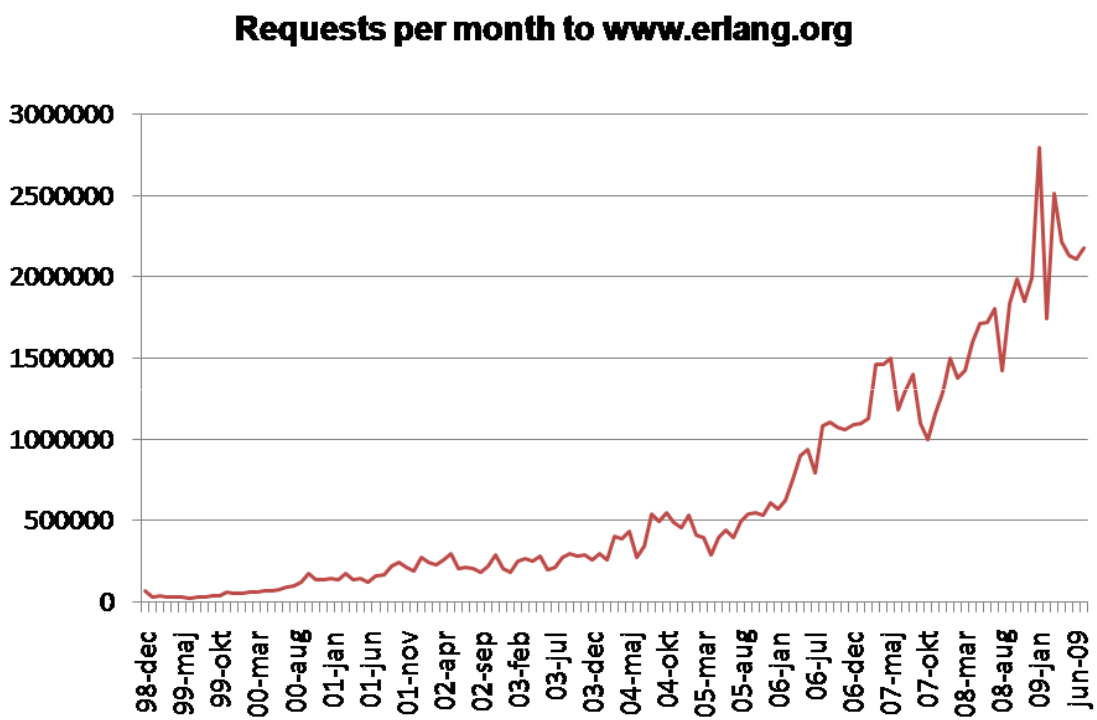
github
SOCIAL CODING

Amazon SimpleDB™ BETA

Erlang

Conclusion

- Erlang was born and bred for Cloud infrastructure
- Connectivity, scalability, messaging are becoming mainstream concepts
- Cloud computing brings Distributed Programming to the masses
- New exciting components appear every month



Erlang

Thank You

Erlang