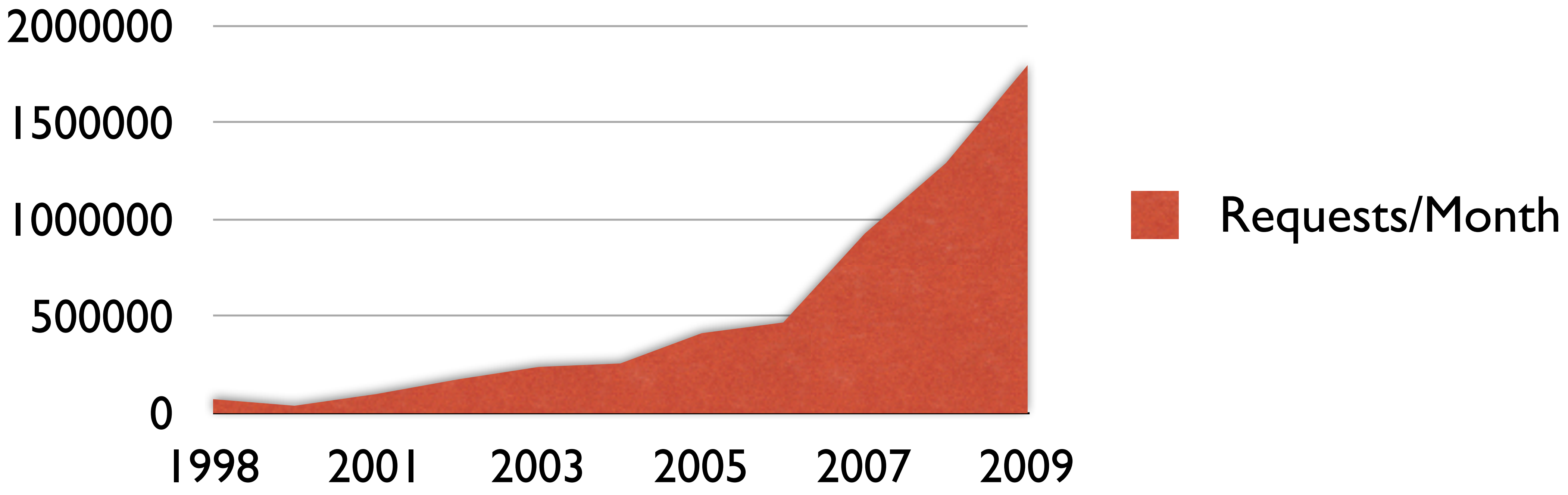
Erlang Solutions Ltd.

Erlang Factory Lite

Moscow 22.06.2012

Erlang, the Language

- Started out in the Ericsson software lab 1987 (!)
- Released as open source in 1998
- Gains Symmetric Multi Processing (SMP) support in 2005



Erlang Community

- **Erlang Factory**

2 day conference
More than 40 speakers
London and San Francisco Bay Area

- **Erlang Factory Lite**

1 day conference
Several talks
Coming to Moscow for the first time

- **Erlang User Group Meetings**

One or two talks
Organized all around the world

Erlang Solutions Ltd

The one stop shop for all your Erlang needs

E-learning

Professional training at all levels

System development and consultancy

Founded in 1999

clients on six continents

Properties

Declarative

Functional programming language, high abstraction level, pattern matching and concise readable programs

Concurrency

Either transparent or explicit concurrency, light-weight processes and highly scalable

Soft Real-Time

Response times in the order of milliseconds per-process garbage collection

Properties

Robustness

Simple and consistent error recovery, supervision hierarchies and "program for the correct case"

Distribution

Explicit or transparent distribution
Network-aware runtime system

Hot code loading

Easily change code in a running system. Enables non-stop operation Simplifies testing

Properties

External Interfaces

"Ports" to the outside world behave as Erlang processes

Portability

Erlang runs on any UNIX, Windows, VxWorks. Supports heterogeneous networks

SMP Support

Symmetric multiprocessing support. Takes full advantage of multiple CPU architectures

Where is Erlang used?

Games

Call of Duty Black Ops, Wooga

Messaging

WhatsApp, Facebook, NKTalk

Databases

Riak, CouchDB

Cloud

Heroku, github, HP Cloud

Automotive

QuickCheck

DemonWare

The full online infrastructure for Call of Duty Black Ops

- Over 2 million concurrent users
- Over 150 million registered users

Erlang core server for controlling Python

- Managing 100,000s of concurrent TCP connections
- Other standalone game-related servers
- Used for concurrency, and gluing sequential code together

Social Games

Social Games

1 000 000 daily users

5000 HTTP reqs/sec

more than 90% writes

around 60000 queries/sec

Erlang process stores user data

Facebook Chat Feature

facebook

Chat backend in Erlang

1+ billion user messages / day

10+ million active channels at peak

100+ channel machines

Architecture

one message queue per user (Erlang process)

HTTP long poll

User ID space partitioned statically

Facebook chat team got Erlang User of the Year award in 2009!

ejabberd

Jabber/XMPP instant messaging server

- cross-platform
- fault-tolerant
- can be distributed on a cluster

Implements many XEPs

- supports MySQL, PostgresSQL, ODBC, LDAP
- SASL authentication, STARTTLS, SSL

Who uses ejabberd

- Nasza Klasa NKtalk
- jabber.org, jabster.pl

WhatsApp

Instant messaging for mobile devices

Erlang servers

Standard configuration

- Dual Westmere Hex-core (24 logical CPUs)
- 100GB RAM, SSD
- Dual NIC (user-facing, back-end/distribution)
- FreeBSD 8.3
- Erlang/OTP R14B03

WhatsApp

Great SMP scalability

85% cpu utilization across 24 logical cpus

Peaked at 2.8M conns

571k pkts/sec

Riak

Master-less distributed database

- scalability and fault-tolerance
- multi-site replication
- MapReduce and riak-search indexing

Pluggable backend

- Bitcask, Innostore, LevelDB, RAM

Who uses Riak

- Comcast, Yammer, Voxer
- tablica.pl, slando.ru
- and many more...

Apache CouchDB

Document-oriented database

- schema-free
- replication with bi-directional conflict detection

Queried and indexed in a MapReduce fashion

- using JavaScript
- RESTful JSON API

github

Erlang tier dispatches request to Ruby machines

- BERT (JSON-like RPC calls)
- scales on a cluster

heroku

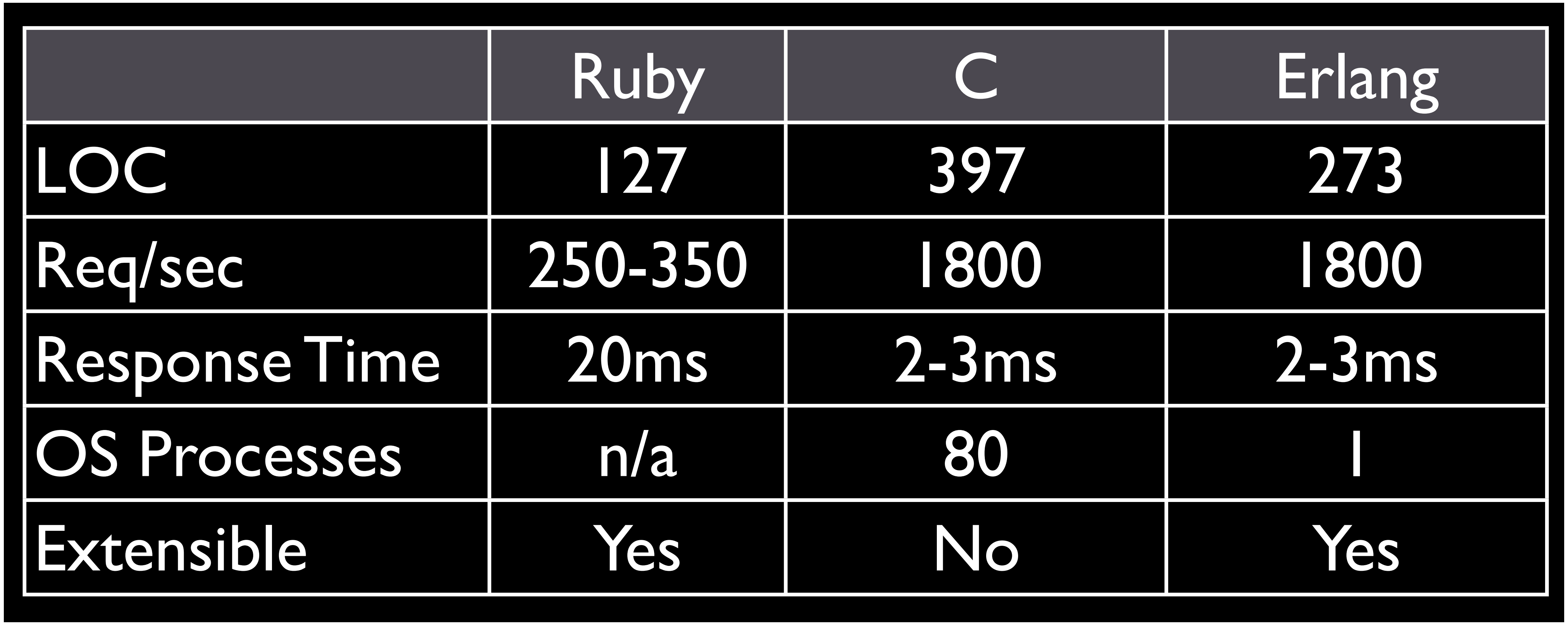
Ruby Cloud platform

- Hosting of Ruby on Rails applications
- Over 40.000 deployed apps
- Automatic scaling in case of increased demand for resources

Campfire

37signals

Erlang web based group chat service



	Ruby	C	Erlang
LOC	127	397	273
Req/sec	250-350	1800	1800
Response Time	20ms	2-3ms	2-3ms
OS Processes	n/a	80	1
Extensible	Yes	No	Yes

QuickCheck

Property based testing – what is this?

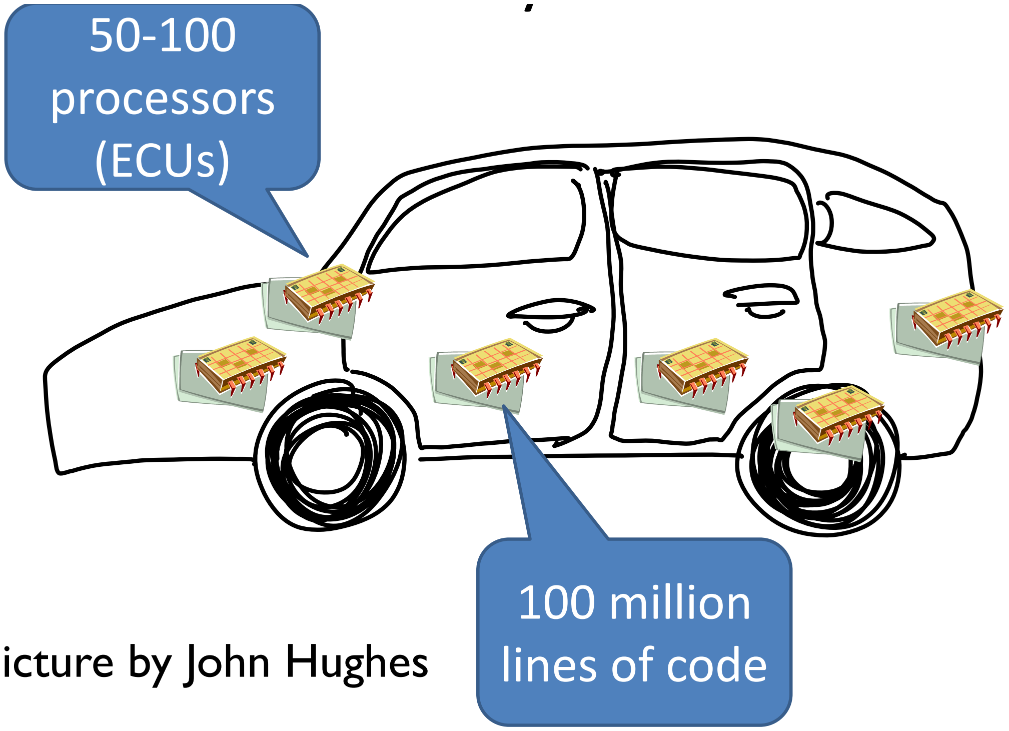
Generalisastion of use cases

Generation of use cases for free

Test specification consists of properties and generators – formal specification

Controlled randomness

Automotive



picture by John Hughes

Спасибо !