

Martin Lucina

Managing Director, Lucina & Associates

martin@lucina.net, Twitter: [@matolucina](https://twitter.com/matolucina), Github: <https://github.com/mato>

Company website: <http://lucina.net/>

LinkedIn profile: <http://www.linkedin.com/in/mlucina>

Key Skills

Motivation and Values

Computers are tools meant to work *for their users and get out of the way*. Give users freedom and control and automate everything else.

Collaboration on Open Source, platforms, APIs and protocols where possible is preferable to building walled gardens and closed systems.

Simplicity and elegance — Everything should be made as simple as possible, but no simpler.

Personal

- Excellent verbal and written communication skills.
- Native speaker of English, Slovak and Czech.
- Multi-cultural background (lived for extended periods in Slovakia and New Zealand).
- Experienced in working with distributed teams and Open Source communities.

Technical

- Extensive experience with hypervisor (KVM, Xen, bhyve) and unikernel technologies (Solo5, MirageOS, Rump Kernels and rumprun), hardware virtualization, Linux containers and software security engineering.
- Hands-on experience with the entire Linux/UNIX/POSIX platform and software stack from 1992 through to present day. Highlights include kernel and systems programming, application/library build systems, packaging and distribution (Debian, RPM, OpenWRT, embedded distributions), performance measurement (perf events, low-latency software profiling using hardware TSC).
- Designed and implemented many distributed systems using both custom-built messaging and Open Source technologies such as AMQP and ZeroMQ. Co-developer of OpenAMQ (first AMQP implementation), co-author of ZeroMQ low-latency middleware.
- Extensive knowledge of the Internet Protocol stack, POSIX APIs and their various implementations, software portability between POSIX and Win32.
- Programming language agnostic; has worked primarily in C, C++, Perl, Python, Ruby, shell, and Assembly (x86, ALPHA, ARM, MIPS). Some experience with OCaml, centred mainly on the language runtime.
- Software and release engineering processes and tools - bug tracking (e.g. JIRA, Bugzilla), build automation (e.g. Buildbot, Travis CI), version control (e.g. Git,

- Subversion) and testing (mostly custom-built solutions).
- Ability to quickly dive into a large code base and solve complex problems.
- Enjoys getting anything to integrate with ... anything else.

Open Source Projects

2016 — present: Solo5

Solo5¹ is a “unikernel base layer”, primarily useful for running MirageOS, IncludeOS and other unikernels, either on various existing hypervisors (KVM, FreeBSD/bhyve, Google Compute Engine) or on a specialized “unikernel monitor” called ukvm.

2015 — present: MirageOS

MirageOS² is a library operating system that constructs unikernels for secure, high-performance network applications across a variety of cloud computing and mobile platforms. Code can be developed on a normal OS such as Linux or MacOS X, and then compiled into a fully-standalone, specialised unikernel that runs under a Xen or KVM hypervisor.

My work on MirageOS centres primarily on the low-level layers (aka “unikernel base”), supporting a freestanding OCaml runtime and platform support for MirageOS to run on KVM, FreeBSD/bhyve and other targets (via Solo5).

2014 — present: Rump Kernels

Rump Kernels³ provide free, portable, componentized, kernel quality drivers such as file systems, POSIX system call handlers, PCI device drivers, a SCSI protocol stack, virtio and a TCP/IP stack. The fundamental enabling technology is the anykernel architecture of NetBSD, which enables the use of unmodified NetBSD kernel drivers.

My work with Rump Kernels centres primarily on the rumprun unikernel and its application for running unmodified POSIX applications as unikernels atop various cloud hypervisors.

2010 — 2012: ZeroMQ

As co-maintainer and one of the original core developers of ZeroMQ⁴, participated in directing project development and primarily worked on:

- release engineering,
- the GNU autotools build system and test framework,
- operating system portability,
- core feature development and API design,
- documentation (author of the ØMQ API Reference Manual).

Worked with the wider Open Source developer community on establishing common processes for project contributions, project version control with Git and high level

1 <https://github.com/Solo5/solo5>

2 <https://mirage.io/>

3 <http://rumpkernel.org/>

4 <http://www.zeromq.org/community>

abstractions for programming language bindings.

2005 — 2008: OpenAMQ and AMQP

As a member of the core iMatix Corporation team which developed the AMQP⁵ standard for message-oriented middleware and OpenAMQ⁶, its Open Source reference implementation, was responsible primarily for:

- the core technical and network wire protocol layers of the software,
- portability layers and operating system integration,
- performance analysis and testing.

ongoing: Linux Kernel

Apart from bug reports, various smaller contributions over the years include:

- the tgafb driver, a frame buffer driver for the DEC 21030 chip used in DEC ALPHA systems⁷.
- the donauboe IrDA driver for Toshiba laptops⁸.
- the ACPI platform driver for Panasonic Toughbook laptops.

Selected Consulting Projects

2014

For Logomotion s.r.o., a Slovak R&D company, developed a custom application for mastering disks using the UDF filesystem. Using Rump Kernel components enabled the re-use of unmodified production quality NetBSD code and development of the project in record time, including integration with a GUI developed in Qt and C++.

2013 — 2014

As Principal Engineer at Boltian, researched, designed and implemented a unique child protection and network security product. Developed a custom portable firmware for embedded wireless routers (Linux/OpenWRT and other OEM) with a back-end HTTP API implemented in C, using the Symas LMDB database for on-board storage. Front-end UI was developed as a modern HTML5 “single page application” using Knockout.js.

2012

Consulting for Hewlett-Packard Korea, advised a major South Korean financial institution on developing a custom low-latency middleware product. Worked directly on-site with the customer's engineering team to define performance test scenarios and train the team in understanding performance measurement of low-latency software systems.

5 <http://www.amqp.org/>

6 <http://www.openamq.org/>

7 <http://git.kernel.org/?p=linux/kernel/git/torvalds/linux.git;a=blob;f=drivers/video/tgafb.c>

8 <http://git.kernel.org/?p=linux/kernel/git/torvalds/linux.git;a=blob;f=drivers/net/irda/donauboe.c>

2011

Consulting for VMware Inc., developed a proof of concept implementation⁹ of the Scalability Protocol for the Linux kernel. SP is an evolution of the concepts behind ØMQ, with the goal of eventual IETF standardisation¹⁰ of a wire protocol for globally scalable distributed messaging.

Working with a Slovak auditor and accounting practice, developed an online application to catalogue and track paper files. While technically a simple “CRUD” application built with Ruby on Rails, the most interesting part of the work was designing the UX experience to make it as painless as possible for the client's day to day use while at the same time enforcing business processes.

2005 — 2008

On the business side of the OpenAMQ project at iMatix Corporation, took on the role of Product Manager. Responsible for the migration of an existing application at JPMorganChase bank from legacy middleware onto OpenAMQ. Worked closely with the client to successfully migrate their global deployment onto the new middleware.

2004

Developed a custom middleware product for iMatix Corporation. The product connected GSM network operators with SMS applications, providing full message queuing and routing for SMS messages with support for multiple protocols (UCP, XMPP and custom SOAP), routing of multiple short codes, and billing. Implementation was in a mix of C and Perl, using Libero state machines to define program logic.

2002

Worked with iMatix Corporation on an industrial automation project for CBR Belgium¹¹. Developed a custom embedded Linux distribution based on the Debian “boot floppies”, integrated the full software and hardware stack. Designed and implemented a redundant fault-tolerant architecture allowing for hot-swap of components; the embedded kiosks would boot from the network and the boot server was simply a Live CD.

Further Work Experience

2000 — 2001

Worked for Catalyst IT, a leading Open Source company in New Zealand, as lead developer on the PropertyStuff project, a nationwide real estate classifieds website which we built with HTML::Mason and Perl.

1998 — 2000

Worked for EDS (New Zealand). Developed a custom backup application for the NCR MP-RAS platform, written in C and Perl. Involved in developing, deploying, maintaining and mentoring users of an internal multi-platform software tool set, based on GNU and other free software and using RPM as a package manager.

⁹ <https://github.com/250bpm/linux-2.6/tree/sp-v2.6.36>

¹⁰ <http://groups.google.com/group/sp-discuss-group>

¹¹ http://imatix.wikidot.com/local--files/portfolio/CBR_Gent_2002_Review.pdf

Publications and Talks

*Deploying real-world software today as unikernels on Xen with rumprun*¹²
Xen Project Developer Summit, August 2015.

*Rumprun for Rump Kernels: Instant unikernels for POSIX applications*¹³
New Directions in Operating Systems, November 2014.

*ZeroMQ “chalk talk” for general developer audience*¹⁴
Presented at Weta Digital, March 2012.

*Towards messaging on an Internet scale*¹⁵, with Martin Sústrik.
Self-published, March 2010.

*ØMQ: A new approach to messaging*¹⁶, with Martin Sústrik.
LWN.net, January 2010.

Education

Victoria University of Wellington, New Zealand. 1998 — 2000, attended courses towards a double degree, BSc (Computer Science) and BA (Philosophy). Left at third year level for full-time professional work.

Interests

Photography, The Arts, Typography, Hiking.

Yachting — major voyages:

- St Malo — Channel Islands — Portsmouth — Cherbourg — St Malo (2010)
- Circumnavigation of Corsica from Castiglione della Pescaia, Italy via Isola d'Elba (2011)
- Mindelo — Sao Antao — Brava — Praia — Tarrafal — Mindelo (2013, Cape Verde Islands)
- Kinsale, Ireland — Kilmore Quay — Holyhead, Wales — Douglas, Isle of Man — Conwy, Wales — Menai Straits — Caernarfon — Arklow, Ireland (2013)
- Greece; Athens to Ionian Sea and Corfu via Corinth Canal, Gulf of Patras, Zakynthos, Ithaca, Levkas (2014)

¹² <http://lucina.net/files/rumprun-xpds2015.pdf> (slides), <https://www.youtube.com/watch?v=2v-ltIQ324I> (recording)

¹³ <http://operatingsystems.io/slides/rumprun-rump-kernels-lucina.pdf>

¹⁴ <http://lucina.net/files/zmq-talk-weta-mar2012.pdf>

¹⁵ <http://250bpm.wikidot.com/local--files/start/towards.pdf>

¹⁶ <http://lwn.net/Articles/370307/>