

# Richard Dawe's Curriculum Vitae

Age: 41  
Nationality: British  
Locality: Bristol, United Kingdom  
E-mail: cv@phekda.org  
LinkedIn profile: <http://www.linkedin.com/in/richdawe>  
Github: [richdawe](#) and [rdawemsys](#)  
Telephone: 0777-6141675

Principal Engineer and systems software engineer with 19+ years' experience, from multi-threaded server software running on-premise, through to distributed systems running in the cloud. I've worked mostly with C and C++ on Linux, in small- to medium-sized companies.

I want to build services that delight customers, that scale to large numbers of customers/users/transactions/jobs, and are dependable.

I'm a self-starter, with a desire to understand systems inside-out. I love digging into issues, starting at the high level, diving down into the code of individual components and gathering data, to build up a mental model of the system.

I have a history of delivering solid, reliable code. I'm customer-focussed and results-driven. I'm always looking for ways to help my colleagues succeed, and to improve the quality of the systems that we're building.

## Skills

Development languages	C (proficient) (19 years); C++ (C++03, STL, <a href="#">boost</a> ) (proficient) (5 years); Lua (proficient) (7 years)
Scripting / glue languages	Perl (intermediate); bash shell scripting (intermediate); Python (beginner)
Development environments	Linux (mostly) and Solaris; multi-threaded software; client and server software; event loop with threadpools
Databases	Cassandra (intermediate); Postgres (intermediate)
Development practices	Version control (Mercurial, git, subversion); debugging (gdb, valgrind, strace, lsof, small amount of DTrace / perf); packaging for install/upgrades/uninstalls (rpm); design and code reviews; Continuous Integration, Continuous Deployment (Bamboo, Ansible)
Development processes	Kanban, Scrum, waterfall
Networking protocols	HTTP 1.1, SMTP, ESMTP, SMPP, SNMP; SSL and TLS (including X509 certificates and Public Key Infrastructure (PKI)); TCP/IP (IPv4, IPv6, UDP, TCP)
Data formats	

## Employment

<p><b>September 2007 - present</b></p>	<p><a href="#">SparkPost</a>, and <a href="#">Message Systems, Inc.</a> (remote working, from home), UK</p> <p>SparkPost is built by Message Systems, the world's #1 email infrastructure provider. Customers including LinkedIn, Twitter, Groupon, Salesforce, Marketo, Pinterest and Comcast send over 3 trillion messages a year, over 25% of the world's legitimate email. These companies choose SparkPost to provide the deliverability, speed and insight they need to drive customer engagement for their business.</p> <p>SparkPost provides email delivery services for apps and websites, hosted in the Amazon Web Services cloud, and accessible via REST APIs and SMTP (Simple Mail Transfer Protocol). Reporting and metrics are provided via webhooks, APIs and the administrative UI.</p> <p>Message Systems's Momentum software is a high-performance messaging server supporting SMTP for mail reception/delivery, supporting extensions in C or Lua, e.g.: for integrating anti-spam technologies.</p> <p>SparkPost is a trading name of Message Systems Inc. I've broken my time at SparkPost / Message Systems down into several "eras" below.</p>
<p>March 2015 - present</p>	<p>The Cloud era at SparkPost: <a href="#">Principal Engineer</a></p> <p>As part of the Platform Engineering team, I designed and implemented new features in our SparkPost and Momentum products. I provided architectural oversight for the messaging platform and integration with other systems and with internal teams. I provided technical leadership and encouragement to the rest of the team. Some highlights:</p> <ul style="list-style-type: none"><li>• Vision and design for taking our single-tenant mail platform and making it support multiple tenants concurrently. This was part of a larger project to reduce costs and allow more rapid delivery of new features. I led the development team; did a good chunk of the development work (mostly in Lua). Migrations to the new architecture were done by a separate team.</li><li>• Investigated and greatly reduced the latency of REST APIs. Documented the behaviour of the garbage collector (GC) in our internal fork of the Lua 5.1 virtual machine. Implemented data gathering to help understand slowest steps. Changed the GC behaviour to remove a major bottleneck, reducing latency by 50%. (All C code)</li><li>• Designed and prototyped the <a href="#">Relay Webhooks</a> feature, for converting inbound SMTP traffic into HTTP POST calls (mostly in Lua, with C as necessary). This feature enables customers to support e-mail using purely HTTP/REST protocols.</li><li>• Added support to Momentum for delivering SMS messages over encrypted connections (SMPP over TLS) (in C).</li><li>• Triaging and resolving escalations from our support team; tracking down production issues; supporting colleagues working on escalations / production issues; proposing fixes in C or Lua.</li><li>• Provided the vision and technical guidance for delivering a feature for managing e-mail domains. This imposed new restrictions on customers, and required discovery and much communication with customers. Migrations for the new feature were done by others.</li><li>• Part of the API Governance team, which was responsible for designing the initial APIs for SparkPost, setting the internal REST API standards, and providing</li></ul>

	<p>on-going assistance for new APIs, and breaking API changes.</p> <ul style="list-style-type: none"> <li>• Trained my colleagues: multiple event loop architecture in C; extending Momentum in C and Lua; Lua best practices; how Momentum delivers e-mail; TLS encryption.</li> <li>• Kept people focussed on delivering the minimum functionality required. This was partly to ensure we could iterate rapidly, but also to ensure that we did not over-build software before we understood what our customers wanted or needed. We used Kanban, and earlier Scrum.</li> <li>• Secondary on-call for production issues for the Platform Engineering team.</li> </ul>
<p>June 2013 - March 2015</p>	<p>The on-premise product development era at Message Systems:</p> <p><u>Lead Engineer</u> (June 2014 - March 2015),  <u>Senior Software Engineer</u> (July 2013 - June 2014)</p> <p>I joined the Engineering team part-way through the development of Momentum 3.6, a major update to take full advantage of multi-core systems; and at the beginning of Momentum 4.0, a major update to the Momentum product line, adding scalable, fault-tolerant message generation APIs, and new UI, metrics APIs and webhooks. Some highlights:</p> <ul style="list-style-type: none"> <li>• Designed and built the C and Lua APIs to allow REST APIs to be built using idiomatic Lua on top of Momentum's HTTP server. This allowed faster development and iteration of key APIs than would have been possible with C.</li> <li>• Designed the click and open tracking system, and led the team building it. (80% C, 20% Lua)</li> <li>• Discovered, investigated and resolved throughput issues with the HTTP server when running in multi-threaded mode. (C code)</li> <li>• Tracked down and resolved some infrequent and subtle crashes when reloading the configuration under high load, due to unsafe use of data across multiple threads. Some core C APIs and callers needed updating to be thread-safe. (C code)</li> <li>• Triaging and resolving escalations from our support team; supporting colleagues working on escalations / production issues.</li> <li>• Subject-matter expert on the internals of our software, Lua, and mail standards.</li> <li>• Trained my colleagues: lifecycle of messages in Momentum; how to use 3rd-party anti-virus and anti-spam integrations.</li> </ul>
<p>September 2009 - July 2013 (~6 years)</p>	<p>The "Professional Services" era at Message Systems:</p> <p><u>Principal Solutions Engineer</u> (January 2012 - July 2013)  <u>Client Integration Architect</u> (September 2009 - January 2012)  <u>Professional Services Engineer</u> (May 2009 - September 2009)  <u>Senior Software Engineer</u> (September 2007 - May 2009)</p> <p>As a part of the Professional Services team, I designed and implemented solutions based on Message Systems' Momentum messaging server, primarily for e-mail. This involved extensive customisation, including extending the platform with C, C++ and Lua.</p> <p>I was the technical sponsor and development lead for several large Professional Services engagements, including two separate multi-year projects. These involved extensive requirements analysis, design and solution implementation work on behalf of the clients.</p> <p>Many of the projects were replacements for existing mail systems. Challenges included "hidden requirements" where the functionality of existing systems was undocumented or not well understood; providing feature parity or equivalence, including integrating with other customer systems; in some cases, no access was permitted to customer systems due to regulatory requirements.</p>

	<p>One of the largest engagements that I was responsible for involved customising Momentum with ~5,000 lines of C++ code, replacing some of the core codepaths of Momentum with custom behaviour, and driving mail policy based on data from an external system accessed via a C++ library provided by the customer. Message Systems was contracted to develop and support this code. I developed about 50% of the code. I also documented how a complex and poorly understood part of the customer's existing system worked, so that we could build the replacement.</p> <p>Another large engagement was one of the first large-scale uses of Lua for mail policy with Momentum, driven by data from a Postgres database. As part of this project I and another colleague in Professional Services found, debugged and fixed a number of issues with the Lua bindings, which involved tracing problems from the Lua layer down into the core C code. These ranged from simple functional bugs through to subtle memory corruption or timing bugs.</p> <p>Other activities included working with the Engineering Team on development of the mail platform, discussing client requirements with Product Management, and working with the Support Team.</p>
<p><b>May 2003 - May 2007</b></p>	<p><a href="#">MessageLabs</a>, Gloucester, UK (Now Symantec Email Security.cloud)</p> <p>MessageLabs is a provider of managed services for e-mail, Instant Messaging (IM) and web communications, to companies of all sizes, and with partners such as IBM and Verizon. In the case of e-mail, messages sent to/by a customer pass through MessageLabs's e-mail infrastructure before being delivered. Services such as anti-virus (AV), anti-spam (AS), Image Control (IC) and Content Control (CC) are applied as the message passes through.</p>
<p>October 2005 - May 2007</p>	<p><u>Senior Software Engineer</u> at MessageLabs:</p> <ul style="list-style-type: none"> <li>• Responsible for the SMTP mail server software at the core of MessageLabs's e-mail infrastructure. This software accepted, scanned, modified (if required; e.g.: tag with " [SPAM] ") and delivered messages according to customer-configurable policy. Written in a mix of C and C++ (STL and boost), packaged for installation with rpm, all on Red Hat Linux.</li> <li>• Designed &amp; implemented new features in the mail server to support updated versions of AV/AS/IC/CC services, working closely with cross-functional teams (Development, Test, Operations, Client Services, Product Management).</li> <li>• Led a team of 2 - 4 people implementing new features and bugfixes in the mail server, and associated software. This included maintenance releases for critical bugs in tight timescales.</li> <li>• Supported the Operations and Client Services teams in resolving customer issues.</li> <li>• Experience of the full software lifecycle, from requirements analysis to ongoing maintenance.</li> <li>• Ensured quality: peer code reviews; unit testing; end-to-end test suites (using Perl and other harnesses); memory debugging/validation tools such as valgrind; helped define C++ coding recommendations; investigated and evaluated static analysis tools for improving C++ code quality; local expert resource on rpm (mostly from painful real-life experience).</li> <li>• Technical Lead for two projects to improve mail reporting to customers: one succeeded; one failed, due to factors including problems with project dependencies, and storage limitations.</li> <li>• Technical Lead for a proof-of-concept (PoC) to evaluate a replacement for the aging SMTP mail server software. The PoC was successful, <a href="#">Eccelerity was chosen</a>. Subsequently I advocated its adoption in several solutions, to improve time-to-market, reliability and manageability.</li> <li>• AccuRev and CVS administrator; Linux systems administration</li> </ul>

<p>May 2003 - September 2005</p>	<p><u>Software Engineer</u> at MessageLabs:</p> <ul style="list-style-type: none"> <li>• Responsible for the SMTP mail server software at the core of MessageLabs's e-mail infrastructure.</li> <li>• Designed &amp; implemented new features in the mail server, working closely with cross-functional teams.</li> <li>• Supported the Operations and Client Services teams in resolving customer issues.</li> <li>• Ensured quality through thorough testing, and validation with tools such as valgrind.</li> <li>• Integrated Symantec Brightmail Anti-Spam into the SMTP mail server. Packaged the Brightmail server provided by Symantec, and wrote monitoring software for the Operations team. Both the mail server and Brightmail server software were deployed and went live with no major issues.</li> <li>• Produced security updates for vulnerabilities announced via Bugtraq and other sources. This included porting fixes to older, vulnerable versions of the software.</li> <li>• AccuRev and CVS administrator; Linux systems administration</li> </ul>
<p>August 1999 - October 2002</p>	<p><u>Cabletron Systems</u> / <u>Riverstone Networks</u>, Reading, UK</p> <p>Cabletron (now <u>Enterasys Networks</u>, now part of Extreme Networks) was a producer of enterprise networking equipment. Riverstone Networks was spun-off to focus on equipment for Metropolitan Area Networks (MAN) and the Internet edge.</p>
<p>June 2000 - October 2002</p>	<p><u>Software Engineer</u> at Riverstone Networks:</p> <ul style="list-style-type: none"> <li>• Sole microcode author for one custom processor. Wrote microcode (assembler) for two custom processors, including support for Multi-Protocol Label Switching (MPLS) and IPv6 protocols. Worked closely with the hardware, software and simulation teams in California.</li> <li>• Initiated and wrote significant portions of some tools in C++ for end-to-end black-box testing of microcode, software and hardware simulations. These were used to find several hardware bugs during simulation, avoiding re-spins of hardware, saving significant sums of money.</li> <li>• Responsible for porting the existing IP, TCP, UDP and PPP MIB support to a new SNMP agent, as part of a project to support monitoring and managing Riverstone's Metropolitan Area Network (MAN) routers with SNMPv3.</li> <li>• Developed using the GNU tool-chain on Debian GNU/Linux and Solaris. Extensive TCL, Perl and shell scripting for testing.</li> <li>• ClearCase administration; Solaris, Linux and Windows systems administration</li> </ul>
<p>August 1999 - June 2000</p>	<p><u>Software Engineer</u> at Cabletron Systems:</p> <p>Worked on Cabletron's Virtual Private Networking (VPN) software in embedded C for its range of small-office routers:</p> <ul style="list-style-type: none"> <li>• Extended the software's command-line interface.</li> <li>• Internet security protocol testing: IKE &amp; IPsec.</li> <li>• Worked on support for Public Key Infrastructure (PKI) using digital certificates, which included porting a vendor's PKI toolkit to the router software environment. Liaised closely with the PKI toolkit vendor to report bugs, then test and integrate bugfixes into their toolkit.</li> <li>• Developed using the GNU tool-chain on Digital Unix.</li> </ul>
<p>Summer 1996, 1997, 1998</p>	<p><u>Alchemetrics Ltd.</u>, Wokingham, UK</p> <p><u>Programmer</u>: Wrote database programs on SCO Unix in C; wrote "man" page documentation for their C database library; wrote a caching library for their database</p>

	library to speed data access; worked on a Motif-based database viewer.
<b>Summer 1995</b>	<b>Murray Computing Ltd., Wokingham, UK</b> <u>Programmer</u> : Wrote database programs which eliminated invalid data and extracted statistics from the data. Used Borland C on DOS.

## Education

	<b><u>University of Bristol, UK</u></b>
<b>1995 - 1999</b>	<u>MSci Physics with First Class Honours</u> : Four-year course, combining a BSc and Masters degree. Final year included: 20-week research project with a partner on nanofabrication; 45-minute presentation on Free Electron Lasers. For my elective modules, I focussed on Quantum Mechanics, mathematics and the properties of materials.
	<b>St. Crispins's Comprehensive School, Wokingham, Berkshire, UK</b>
<b>1988 - 1995</b>	<u>A-Levels (1995)</u> : Chemistry (A), Maths (A), Physics (A) <u>AS-Level (1995)</u> : German (A) <u>GCSEs (1993)</u> : 8 As, 2 Bs  <u>Prizes for GCSEs</u> : Design & Technology; Physics; joint-best grades overall <u>Prizes for A- and AS-Levels</u> : AS-Level German; best grades overall

## Interests And Hobbies

- Learning German (intermediate level).
- Travelling: mainland Europe, South Korea, Japan, USA.
- Yoga; cycling & mountain biking.
- Member of the Institute of Physics (MInstP)
- Former member and speaker at [Birmingham Perl Mongers](#), a Perl user group ([slidedecks](#)).

---

Wed, 22 Aug 2018 17:15:39 +0100 (cv.html 739a3b64ee67590f5767afa24d6790e5095bc166)