Stephen Kell

e-mail: srkell@acm.org
telephone: +44 7751 103384
date of birth: 1st March 1984
nationality: British

Recently held positions

2021– **King's College London**
Lecturer in Computer Science

2018–21 **University of Kent**
Lecturer, in Progamming Languages & Systems research group
- Involved in full complement of teaching-related activities including lecturing, project and consultancy supervision, tutorial/mentorship, practical class supervision, undergraduate selection
- Lectured and examined courses at Master's (C++), conversion Master's (system architecture) and Bachelor's levels (software engineering, computer architecture, compilers)
- Co-I (PI: Batty) on EPSRC Digital Security by Design grant (£485k); PI on VeTTS PhD studentship funding (January 2020–; £75k); and began EPSRC proposal collaboration with J. Ryan Stinnett (ex-Mozilla)

2013–18 **University of Cambridge**
Senior Research Associate (2017–18), Research Associate (2013–17)
- Supported by EPSRC programme grant "Rigorous engineering for mainstream systems" (PI: Peter Sewell).
- Originated and led libcrunch project on efficient, precise dynamic type and bounds checking in idiomatic C code.
- Collaborated on linksem project defining executable formal specification of ELF linking; code used by other projects in group on binary emulators.

Research funding

- (as PI): EPSRC New Investigator award: debugging made dependable and measurable (DIADEM) on testing and synthesising debugging information (£325k), industry partners Arm and Pernosco. May 2022 – November 2024.
- (as co-I): EPSRC Digital Security by Design call: investigate use of CHERI hardware as accelerator for precise type and bounds checks currently implemented in software (PI: Batty; £485k). August 2020 – July 2023.
- (as PI) Verified & Trustworthy Software Systems Institute (via GCHQ/NCSC): £75k, to investigate generalised static checking of type and bounds errors. January 2020 – March 2021.

Earlier research and academic positions

2013 **Oracle (Labs) America**
Research Assistant

- In the Virtual Machines group, working on Graal multi-language dynamic compilation engine (principal collaborators: Haupt, Wimmer, Kessler)
- Recruited for expertise in object-code debugging metadata; key contributor to metadata generation in experimental SubstrateVM runtime.
- Designed and implemented object file manipulation library.
- Work now shipping as part of Graal Native Image feature.

2012–13   **Faculty of Informatics, University of Lugano, Switzerland**
Postdoctoral Assistant
- Supported by SNF "Foundations of Dynamic Analysis" joint ETH/Lugano project (PIs: Binder, Hauswirth, Gross)
- Mentored PhD student projects on compositionality (Ansaloni), isolation (Marek) and dynamic optimisation (Zheng) of instrumentation-based dynamic analysis infrastructure on the Java Virtual Machine, and multi-language workload characterisation (Sarimbekov).
- Led design aspect of compositionality work, published at ECOOP.

2011–12   **Department of Computer Science, University of Oxford**
Postdoctoral Research Assistant
- James Martin Fellow, on verified software for sensor networks (PIs: Kwiatkowska and Trigoni).
- Led work on practical bug-finding via mixed static/dynamic type-error analysis, using symbolic execution (later begetting libcrunch project).
- Collaborated with former Cambridge project student (Irwin) on experience report from Python/C interoperable runtime project.

2006–10   **Computer Laboratory, University of Cambridge**
PhD student in Systems Research Group, supervised by David Greaves.
- Supported by EPSRC XenSE project grant (2006–08; PI: Hand), EPSRC Doctoral Training Award (2008–10) and Cambridge Philosophical Society research grant (2010).
- Thesis work (self-originated) on Cake language and tools for composition & adaptation of software components having non-matching interfaces.

2007   **Fraser Research, Princeton, New Jersey**
Senior Technical Associate
- Clean-slate internetworking research, under Sandy Fraser.
- Contributed to projects on network switching/routing emulation, tools for manipulating raw Ethernet packet streams, and capability-based secure middlebox designs.

2005–06   **Computer Laboratory, University of Cambridge**
Research Assistant
- Supported by EPSRC XenSE and ERC Open Trusted Computing project grants (PI: Hand); on security features for the Xen hypervisor.
- Collaborated on work to safeguard kernel security using hypervisor-level immutable memory and ratcheted data structures.
- Developed initial XenoLinux port of Nitpicker secure windowing system, subsequently handed off to NSA contractors.

Software engineering positions

2021– **Self-employed consultancy for Petagene Ltd, Cambridge**
- Technical adviser on operating system extension techniques as part of new product offering.

2018 **Self-employed consultancy for ZEIT, Inc, San Francisco**
- Consultant developer on undisclosed experimental product feature.

2009–14 **Self-employed consultancy for Ellexus Ltd, Cambridge**
- Primary developer of back-end tracing engine in Cambridge start-up's developer tools product.
- Early work enabled by EEDA Innovation Voucher secured in cooperation with University of Cambridge technical services division.

2005 **ARM Ltd, Cambridge**
Software Engineer
- Embedded debugger development (summer internship), focused on testing and bug-fixing of upcoming debugger release.

2001–04
(various spells) **Opal Telecom Ltd, Irlam, Manchester**
Software Engineer
- Development of customer-facing web authentication system (summer 2004; deployed to customers shortly thereafter), web service messaging middleware (summer 2003; deployed internally); and resource monitoring systems for live telephony switch (2001–2).

Education

2006–10 **Computer Laboratory, University of Cambridge**
PhD in Systems Research Group, supervised by David Greaves. Thesis entitled "Black-box composition of mismatched software components".

2002–05 **Christ's College, University of Cambridge**
BA Hons., Computer Science, class I (all Parts). Included an extended individual project and dissertation "A general-purpose synthetic filesystem".

Teaching experience

Over most of the past 18 years I have taught and (at times) examined, at undergraduate and Master's levels, in a variety of roles and formats and on a variety of computer science topics.

Lecturing

At King's: lectured and examined 0.5–0.6 modules during 2021–22 ($\approx$ 10 lectures).

- 2021–22: Operating Systems and Concurrency (BSc)

At Kent: lectured and examined in three modules during 2018–19 ($\approx$ 29 lectures), four during 2019–20 ($\approx$ 45 lectures), and three during 2020–21 ($\approx$ 37 lectures).

- 2020–21: Computer Systems (computer architecture; BSc)
- 2019–21: Programming Language Implementation (BSc)
- 2018–21: Systems Architecture (basic and intermediate computer architecture; conversion MSc)

- 2018–20: Programming in C++ (advanced MSc)
- 2018–20: Software Engineering (design patterns, testing & assurance; BSc)

Guest lectures

In Cambridge:

- 2017: guest lecturer on Linking, in MPhil advanced compilers course;
- 2010: guest lecturer on Modularity, in PhD students' lecture series to MPhil class.

Project supervision

At King's: supervised four final-year BSc individual projects

In Kent: supervising one individual research project and two group projects. During 2018–20 supervised & examined four completed MSc projects; second-examined five MSc dissertations and eight BSc projects

In Cambridge: (co-)supervised & (in most cases) originated many final-year Bachelor's (Part II) projects, many with research element; fifteen students to date (2008–10, 2014–18), with peer-reviewed research outputs from two projects so far [VMIL 11, EuroLLVM 16]

Co-originated & supervised French's 2015 UROP project; later employed as RA on REMS project (Sewell).

Practical and small-group teaching

In Kent: class supervisor for Software Engineering (BSc), Programming Language Implementation (BSc), Systems Architecture (MSc).
In Cambridge, 2005–10 and 2014–2018: frequent supervisor of several Tripos courses. This work was done for various colleges: I was a postdoctoral affiliate of the Senior Combination Room at Christ's College 2015–18, and longstanding supervisor for several other Colleges (Jesus, King's, Corpus Christi). .

- Programming in C and C++ (most years);
- Algorithms (2018);
- Concepts in Programming Languages (most years);
- Concurrent & Distributed Systems (most years);
- Digital Communications I (2005–10) / Computer Networking (2014–15);
- Operating Systems I / II / Foundations (2005–10);
- Concurrent Systems & Applications (2005–10);
- Further Java (2016–17).

Demonstrator and assessor of various practical courses in Cambridge:

- assessor of Natural Science first-year numerical computing exercise (2008, 2009, 2010), taken by entire physical Natural Science year group;
- teaching assistant in double-credit MPhil practical course on Building an Internet Router (2009–10);
- relief assessor (ticker) of Java programming exercises (2010).

While in Oxford (2011), tutor (Magdalen College) for Digital Systems course, spanning hardware, operating systems, compilers, runtimes.

Development and authorship

At King's: re-developed from scratch Operating Systems course content (lectures: slides, videos, active learning materials, lab assignment); introducing practical OS development

content, based on teaching operating systems (xv6 from MIT and InfOS from U. Edinburgh)

At Kent: co-drafted new Software Engineering module specification to replace existing problematic module; this was as part of Canterbury-wide undergraduate programme review (2018) during which I served on both 'systems' and 'programming' two working groups. Participated in conversion MSc programme review (2020).

In Cambridge: author of a large selection of supervision exercises (mostly during 2005–10), many now widely adopted and/or integrated into course materials (e.g. exercises for Computer Networking as lectured by Moore and Kalyvianaki, incorporate much of my work).

Institutional service and citizenship

University governance

In Kent, elected member of the university Senate (2020–21), the body overseeing all academic business of the institution.

In Cambridge: elected member of the Board of Scrutiny (2015–17; as Secretary 2016–17), a key governance structure of the University of Cambridge, elected by its governing body (Regent House).

- Collectively responsible for scrutinising the University of Cambridge's financial and decision-making activities on behalf of the Regent House, via an annual report.
- Serving as Secretary of the Board during 2016–17: individually responsible for relief chairing of meetings, liaising with interview guests (e.g. Vice-Chancellor), managing Board-internal web resources.

Departmental representation

At Kent: University & College Union representative for School of Computing, 2020–21 (50% role share)

In Cambridge: chair (2017–18) of the Postdoc Forum (later: Research Staff Forum) at the Computer Laboratory, an official channel representing research staff needs to departmental administration and representing department in cross-University chairs' network meetings.

Also in Cambridge: served (2015–18) on department-wide induction mentoring scheme for new research staff.

Postdocs of Cambridge

Served (2015–18) on the committee of this University senior society, whose mission is to build representation and community among postdoctoral researchers.

As Events & Networking officer (2015–17):

- managing programme of both social and research/networking events, chairing subcommittee;
- planning, finances, leadership, risk assessment for events small and large (5–150+).

As Treasurer (2017–18):

- overall responsibility for Society's financial administration;
- invoicing and accounting for large events (e.g. National Postdoc Meeting 2017). General participation including Research Policy & Representation activities

Research

Funding Currently held: see first page.

Proposals in progress:

- EPSRC responsive-mode proposal on systems support for compositional visual programming for computational scientists, with partners at CNRS and Cambridge (bidding for $\approx$£700k)

Interests

Primary interests: all aspects of programming infrastructure, languages & tools, including compilers, runtimes and operating systems. Specific topics of interest:

- infrastructure for multi-language programming;
- debugging and 'live programming' techniques embracing native and low-level code;
- interface specification techniques generally;
- infrastructure for reflective (meta-)programming.
- semantics of low-level code, esp. system call and linker services;
- techniques for heterogeneous composition (across languages, APIs, . . .);

Secondary interests, with links to adjoining fields and disciplines

- History and philosophy of [computer] science:
  - speaker at History and Philosophy of Programming (HaPoP) conference 2016
  - invited speaker to history & philosophy track at CodeMesh 2017
  - invited speaker to HEPIC seminar at University of Lille (2019)
  - co-founder (with Petricek) of Salon des Refusés workshop (at <Programming>, annually since 2017), on criticism-based appraisal of programming technologies
- Usability of programming
  - ongoing community-building collaboration (below); served on programme committee of PLATEAU workshop at SPLASH 2015

Active participant across several department research groups

- at Kent: core membership of Programming Languages & Systems; also full member of Cybersecurity Group

Collaborators

Current and recent collaborators are as follows.

- Larose (U. Kent): seamless interoperability between native code and dynamic languages
- Stinnett (was unaffiliated, now employed RA on DIADEM project): testing and synthesising debugging information
- Petricek, Edwards, Church: community-building around end-user programming (rotating co-organisation of Salon des Refusés workshop; joint panel discussion at Psychology of Programming Interest Group workshop 2019);
- INRIA (Paris: Francesco Zappa Nardelli) testing, specifying and verifying debugging information;
- Cambridge: in connection with REMS project, participation in ISO C study group on C memory object model;

Invitations

- invited lecturer at Programming Languages Implementation Summer School 2020, as rescheduled to 2022
- keynote speaker at ICOOOLPS workshop at PLDI 2021
- keynote speaker at Off the Beaten Track workshop at POPL 2019
- invited speaker at several industry/crossover venues (see below)
- invited participant in CRA/CCC prospecting workshop on 'Inclusive Access to Rich Online Content and Services' (2015), setting US national funding directions at intersections of programming, interaction and accessibility
- guest speaker at over a dozen universities and research institutions across Europe and North America

Service

- **Conference/symposium programme committees**
  - The Art, Science & Engineering of Programming 2024 (ongoing)
  - OOPSLA programme committee 2022
  - The Art, Science & Engineering of Programming 2021
  - Onward! Papers 2020 (programme chair)
  - PLDI programme committee 2021
  - ECOOP programme committee 2020
  - PLDI external review committee 2020
  - The Art, Science & Engineering of Programming 2020
  - OOPSLA external review committee 2019
  - The Art, Science & Engineering of Programming 2019
    * received Outstanding Reviewer Award
  - International Symposium on Memory Management 2019
  - SPLASH workshops committee 2016
  - Symposium on Partial Evaluation and Program Manipulation 2016
  - Onward! Essays '15
  - Principles and Practice of Programming in Java '13
  - European Conference on Computer Systems 2012 (shadow PC)
- **Workshop programme committees and the like**
  - Automated Software Engineering 2022 (Late-Breaking Results track)
  - FAST Smalltalk Foundation workshop 2021
  - Modern Language Runtimes, Ecosystems, and VMs 2020 (online event)
  - Programming Languages & Operating Systems (at SOSP 2019)
  - Programming Experience (at <Programming> 2019)
  - Interconnecting Components Workshop (at <Programming> 2019)
  - Salon des Refusés 2017, 2018, 2019, 2020 (at <Programming>)
  - Evaluation and Usability of Programming Languages and Tools (PLATEAU) workshop 2015 (at SPLASH)
  - Runtime Environments, Systems Layering, and Virtualized Environments (RESoLVE) workshop 2012 (at ASPLOS)
- **Organisational roles**
  - Onward! steering committee member (2020–), chair (2023–)

- – SPLASH steering committee member (as Onward! Papers PC chair 2020)
- – Co-organiser of Virtual Machines & Language Implementation workshop (2018)
- – Co-organiser of Salon des Refusés workshop 2017
- – publicity chair, Software Composition 2013
- **External reviewing and journal refereeing**
  - – Programming Languages & Operating Systems workshop (at SOSP 2021)
  - – ACM TOPLAS (2020)
  - – IEEE Software (2019)
  - – Software: Practice & Experience (2018–9)
    - * 3 submissions; named Distinguished Reviewer in 2019
  - – IEEE/ACM Automated Software Engineering conference (2018)
  - – ACM Transactions on Architecture and Code Optimization (4 submissions, 2015–)
  - – Automated Software Engineering journal (Springer) (2014)
  - – ACM Symposium on Applied Computing (2012)
  - – ACM SIGCOMM (2011)
  - – European Symposium on Programming (2010)
  - – European Conference on Computer Systems (2009)

Industrial engagement and outreach

Speaking and communicating

Experienced speaker and communicator addressing both research and developer audiences

- industry-facing conference speaker: Curry On! 2019 (invited), CoreDump 2018 (invited), CodeMesh 2017 (invited), Strange Loop 2014
- blog and publications often read and discussed via web media: Hacker News (front-page stories on several occasions), Lambda the Ultimate, `lobste.rs`, Reddit, etc.
- speaker at open-source project developer meetings (EuroLLVM 2016, GNU Cauldron 2017)

Advising companies

Consultant to Petagene, Cambridge start-up with genomics cloud data-storage products

Consultant to ZEIT, Bay Area cloud infrastructure start-up (2018)

Scientific adviser to Ellexus (2009–14); included initating technology transfer collaboration between Ellexus and Lugano colleagues (2013)

Publications

As with much of computer science, most of the competitive venues for new work in my principal research areas are conferences rather than journals. As a brief summary of major venues: SPLASH is an umbrella conference comprising OOPSLA, Onward! and various smaller events. OOPSLA is a primary top-tier conference venue for full-length research papers around programming, programming languages and programming systems (formerly focused on object-oriented programming, but now diversified). Onward! is a sibling event at SPLASH focused on "innovative ideas that challenge existing beliefs, or early work well written and well argued for", accepting both full-length technical papers and also long-form reflective writing (in its "Essays" track). ECOOP is a European conference covering similar ground to OOPSLA. Some well-established workshops in and around the same community are: VMIL (Virtual Machines and Intermediate Languages), at SPLASH most years; PLOS (Programming Languages and Operating Systems), usually at conferences within the systems research community.

All articles are peer-reviewed unless stated otherwise.

Articles in journals, conferences and major symposia

[OOPSLA 19] T. Bastian, S. Kell, and F. Zappa Nardelli. Reliable and fast DWARF-based stack unwinding. In *Proceedings of the 2019 ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications*, OOPSLA '19. ACM, 2019.

[POPL 19] K. Memarian, V. B. F. Gomes, B. Davis, S. Kell, A. Richardson, R. N. M. Watson, and P. Sewell. Exploring C semantics and pointer provenance. *Proc. ACM Program. Lang.*, 3 (POPL):67:1–67:32, Jan. 2019.

[VEE 18] M. Rigger, S. Marr, S. Kell, D. Leopoldseder, and H. Mössenböck. A survey of x86-64 inline assembly in C programs. In *Proceedings of the ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, VEE '18.

[Onward! 17] S. Kell. Some were meant for C: the endurance of an unmanageable language. In *Proceedings of the 2017 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software*, Onward! 2017, New York, NY, USA, October 2017. ACM.

[OOPSLA 16a] S. Kell. Dynamically diagnosing type errors in unsafe code. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, OOPSLA 2016, pages 800–819, New York, NY, USA, 2016. ACM.

[OOPSLA 16b] S. Kell, D. P. Mulligan, and P. Sewell. The missing link: Explaining ELF static linking, semantically. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, OOPSLA 2016, pages 607–623, New York, NY, USA, 2016. ACM.

[Software 16] Y. Zheng, S. Kell, L. Bulej, H. Sun, and W. Binder. Comprehensive multiplatform dynamic program analysis for Java and Android. *IEEE Software*, 33(4):55–63, 2016.

[Onward! 15] S. Kell. Towards a dynamic object model within Unix processes. In *2015 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward!)*, Onward! 2015, pages 224–239, New York, NY, USA, 2015. ACM.

[Onward! 14] S. Kell. In search of types. In *Proceedings of the 2014 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software*, Onward! 2014, pages 227–241, New York, NY, USA, 2014. ACM.

[GPCE 13] L. Marek, S. Kell, Y. Zheng, L. Bulej, W. Binder, P. Tůma, D. Ansaloni, A. Sarimbekov, and A. Sewe. ShadowVM: Robust and comprehensive dynamic program analysis for the Java platform. In *Proceedings of the 12th International Conference on Generative Programming: Concepts & Experiences*, GPCE '13, pages 105–114, New York, NY, USA, 2013. ACM.

[ECOOP 13] D. Ansaloni, S. Kell, Y. Zheng, L. Bulej, W. Binder, and P. Tůma. Enabling modularity and re-use in dynamic program analysis tools for the Java Virtual Machine. In *Proceedings of the 27th European Conference on Object-Oriented Programming*, ECOOP'13, pages 352–377, Berlin, Heidelberg, 2013. Springer-Verlag.

[OOPSLA 10] S. Kell. Component adaptation and assembly using interface relations. In *Proceedings of 25th ACM International Conference on Systems, Programming Languages, Applications: Software for Humanity*, OOPSLA '10. ACM, 2010.

[JUCS 08] S. Kell. A survey of practical software adaptation techniques. *Journal of Universal Computer Science*, 14(13):2110–2157, September 2008.

Peer-reviewed book chapters

[PhilStud 19] S. Kell. Unix, Plan 9 and the lurking Smalltalk. In L. De Mol and G. Primiero, editors, *Reflections on Programming Systems*, Philosophical Studies Series. Springer.

Other peer-reviewed articles

[TAPAS 20] J. Adam and S. Kell. Type checking beyond type checkers, via Slice & Run. In *Proceedings of the 11th Workshop on Tools for Automatic Program Analysis*. ACM, November 2020. To appear.

[Convivial 20] S. Kell. Convivial design heuristics for software systems. In *Conference Companion of the 4th International Conference on Art, Science, and Engineering of Programming*, pages 144–148, 2020.

[PPIG 19] J. Edwards, S. Kell, T. Petricek, and L. Church. Evaluating programming systems design. In *Proceedings of the 30th Annual Workshop of the Psychology of Programming Interest Group (PPIG)*, August 2019.

[VMIL 19] G. Bertholon and S. Kell. Towards seamless interfacing between dynamic languages and native code. In *Proceedings of the ACM Workshop on Virtual Machines and Intermediate Languages*, VMIL '19, New York, NY, USA, 2019. ACM.

[MoreVMs 18] S. Kell. The inevitable death of VMs: a progress report. In *Conference Companion of the 2nd International Conference on Art, Science, and Engineering of Programming*, pages 61–62, 2018.

[PLOS 13] S. Kell. The operating system: Should there be one? In T. Harris and A. Madhavapeddy, editors, *Proceedings of the Seventh Workshop on Programming Languages and Operating Systems*, PLOS '13, pages 8:1–8:7, New York, NY, USA, 2013. ACM.

[PASTE 13] A. Sarimbekov, A. Sewe, S. Kell, Y. Zheng, W. Binder, L. Bulej, and D. Ansaloni. A comprehensive toolchain for workload characterization across JVM languages. In S. N. Freund and C. S. Pasareanu, editors, *Proc. ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering*, PASTE '13, pages 9–16. ACM, June 2013.

[VMIL 13] Y. Zheng, L. Bulej, C. Zhang, S. Kell, D. Ansaloni, and W. Binder. Dynamic optimization of bytecode instrumentation. In *Proceedings of the 7th ACM Workshop on Virtual Machines and Intermediate Languages*, VMIL '13, pages 21–30, New York, NY, USA, 2013. ACM.

[VMIL 12] S. Kell, D. Ansaloni, W. Binder, and L. Marek. The JVM is not observable enough (and what to do about it). In *Proceedings of the compilation of the co-located workshops*, SPLASH '12 Workshops, New York, NY, USA, 2012. ACM.

[VMIL 11] S. Kell and C. Irwin. Virtual machines should be invisible. In *Proceedings of the compilation of the co-located workshops*, SPLASH '11 Workshops, pages 289–296, New York, NY, USA, 2011. ACM.

[FREECO 11] S. Kell. Composing heterogeneous software with style. In C. Bockisch, L. Bergmans, and S. Apel, editors, *Proceedings of the 1st International Workshop on Free Composition*, FREECO '11, pages 5:1–5:5, New York, NY, USA, 2011. ACM.

[Onward! IIP 09] S. Kell. The mythical matched modules: overcoming the tyranny of inflexible software construction. In *Companion to the 24th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, OOPSLA 2009, pages 881–888. ACM, October 2009.

[ICSE NIER 09] S. Kell. Configuration and adaptation of binary software components. In *Companion to the 31st International Conference on Software Engineering*, pages 211–214. IEEE, May 2009.

[SYANCO 07] S. Kell. Rethinking software connectors. In F. Arbab and C. Baier, editors, *Proceedings of the 1st international workshop on synthesis and analysis of component connectors*, pages 1–12. ACM, September 2007.

Tool demonstration papers, extended abstracts etc.

[SPLASH 15] H. Sun, Y. Zheng, L. Bulej, W. Binder, and S. Kell. Custom full-coverage dynamic program analysis for Android. In *Companion Proceedings of the 2015 ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity*, SPLASH Companion 2015, pages 7–8, New York, NY, USA, 2015. ACM.

[APLAS 15] H. Sun, Y. Zheng, L. Bulej, S. Kell, and W. Binder. Analyzing distributed multi-platform Java and Android applications with ShadowVM. In X. Feng and S. Park, editors, *Proc. 13th Asian Symposium on Programming Languages and Systems*, volume 9458 of *Lecture Notes in Computer Science*, pages 356–365. Springer, November 2015.

Peer-reviewed conference abstracts

[HaPoP 16] S. Kell. The operating system: why there should be one (or two), June 2016. Presented at HaPoP-16.

[EuroLLVM 16] C. Diamand, S. Kell, and D. Chisnall. Run-time type checking with clang, using libcrunch. Presented at EuroLLVM 2016, March 2016. Presentation abstract, slides and video available at http://www.llvm.org/devmtg/2016-03/ as retrieved on 2016/8/26.

Invited papers

(Entries here were not necessarily peer-reviewed.)

[SdR 2018] S. Kell. Critique of 'files as directories: some thoughts on accessing structured data within files' (2). In *Conference Companion of the 2nd International Conference on Art, Science, and Engineering of Programming*, pages 175–179, 2018.

Technical reports, dissertations etc.

(Entries here were not necessarily peer-reviewed.)

[corr 2021] M. Greenberg, K. Kallas, N. Vasilakis, and S. Kell. Report on the 'The Future of the Shell' panel at HotOS 2021. arXiv pre-print, 2021.

[PhD 12] S. Kell. *Black-box composition of mismatched software components*. PhD thesis, University of Cambridge, UK, 2012.

Current grants

As principal investigator

DIADEM: Debugging made dependable and measurable. Funded by the EPSRC as a New Investigator Award. Investigation on methods for testing compiler-generated debugging information, measuring coverage, synthesising it with less or no compiler cooperation. Industry partners Arm and Pernosco. £325k, May 2022 – November 2024.

As co-investigator

CapC: Capability C Semantics and Reasoning. Funded by the EPSRC Digital Security by Design call. Investigation of use of CHERI hardware e.g. as accelerator for precise type and bounds checks. Industry partners NVidia and Microsoft Research. PI: Batty (U. Kent), £485k, August 2020 – July 2023.

Past grants

As principal investigator

GenStat. Generalised static checking for type and bounds errors. Funded by the Verified & Trustworthy Software Systems Institute (VeTSS) via GCHQ/NCSC. Investigation of generalising dynamic checking methods into static assurance via symbolic execution techniques. PhD studentship / £75k. January 2020 – (projected) June 2023, terminated March 2021.

- Student withdrew in March 2021 and NCSC lacked process for allowing topping back up (even by my own funds) to re-start with a new student. The agreement was terminated and I received a written apology from the director of VeTSS.