D8.1 Key Personnel Name and Centre Title

Alex Potanin, ARC Centre of Excellence in Trustworthy Software for Safety and Security of Critical Systems

D8.2 Current and Previous Appointments during past 10 years

since 2023	Associate Director HDR	School of Computing, The Australian National University
since 2022	Associate Professor	School of Computing, The Australian National University
2022	Deputy Head of School	School of Engineering and Computer Science Victoria University of Wellington (VUW), New Zealand (NZ)
2021–2022	Associate Dean (Students)	Faculty of Engineering, VUW, NZ
2019–2020	Visiting Associate Professor	Kyoto University, Japan
2018–2022	Associate Professor	School of Engineering and Computer Science, VUW, NZ
2013	Visiting Research Scholar	Carnegie Mellon University, USA
2010–2017	Senior Lecturer	School of Engineering and Computer Science, VUW, NZ

D8.3 Career highlights and contributions to the field, including excellence in researcher development, training, and mentoring.

A/Prof Potanin has worked in programming language design, type systems, and security for 20 years. The Rust programming language was inspired by the decades of work on ownership and immutability that was the focus of Potanin's research for the first ten years of his career. More modern approaches work on securing software components using capabilities both at hardware and software level - this has been Potanin's research for the last ten years of his career. The Wyvern programming language, developed by Potanin after a sabbatical at Carnegie Mellon University in 2013, lays the foundation for usable yet secure programming language designs.

Having worked extensively in the programming language community (and being the General Chair in 2022 for SPLASH/OOPSLA and the Review Committee Chair in 2024 for OOPSLA), Dr Potanin is ideally placed to put together a team of leading Australian-based and international researchers to address the challenges around suitable programming language development. Dr Potanin is an elected IFIP Working Group 2.4 member on Software Implementation Technology.

Dr Potanin's research career started as early as my Honours year when I studied memory relationships between objects in heaps – leading to an eventual publication of our discovery of the scale-free nature of object graphs that has nearly 300 citations in Google Scholar. My PhD demonstrated that type polymorphism has a deep connection to ownership types, resulting in my two most influential papers (each with over 100 citations on Google Scholar) showing how "generic ownership" and "generic immutability" can allow ownership and immutability guarantees to be incorporated "for free" in the existing programming language designs.

In 2008 – 2012, I obtained a $300,000 Marsden Grant funding to demonstrate how ownership and immutability can be unified with an underlying mathematical framework with several additional publications exploring the ownership and immutability implications for secure software development. My ICSE 2013 empirical study and OOPSLA 2010 paper show how essential collections libraries' designs and language designs can be easily tweaked to provide long-lasting security guarantees. I continued to produce other research publications on a variety of topics at the same time, ranging from studying multiple dispatch mechanisms to the usability of GIS and mapping software.

My research career pivoted again in 2013 during my sabbatical, when I switched to the fundamentals of general programming language design and security. Our breakthroughs included a novel way of defining "type-specific languages" where any type will have a parser associated with it that allows a mixed language code to be safely type-checked and parsed simultaneously, preventing "SQL injection" and similar attacks. Our paper was the distinguished paper award at ECOOP 2014. We also demonstrated the decidability of type members. This problem alluded programming language researchers for at least 2 or 3 decades. It showed how capabilities can benefit the design of module systems (that this grant builds upon), and it was the first to demonstrate in 2018 how effect systems can be simplified by using capability-like mechanisms.

Meanwhile, my security-related work at VUW resulted in an ECOOP 2017 paper on the "Evil Pickles," showing how large systems can be brought down with the right crafting of user input due to the inherent algorithmically costly processing involved. We won the Distinguished Artifact Award at ECOOP 2017. Furthermore, we secured USD 50,000 Oracle Labs funding to support our secure language research.

More recently, I moved into the area of software verification – spending my 2019/2020 sabbatical that had to be aborted after just four months due to COVID-19 in Japan, working at the leading Kyoto University, where I taught a graduate course purely around my research outputs so far (https://potanin.github.io/kyoto2020/) and established a new relationship including remote supervision with Atsushi Igarashi – who, together with Naoki Kobayashi, has been developing a verifier that utilises ownership to infer correctness guarantees automatically. My work continues to win awards, including the FORTE 2022 Best Paper and, in September 2024, another Distinguished Artifact Award at ECOOP 2024 for one of the two papers I published there.

Finally, the breadth of my research goes beyond programming languages and security. We work with local farmers on a token trading platform to influence pollution mitigation behaviours and balance the economic costs involved. Most recently, we obtained NZD 72,000 in funding from Robonomics Network to establish a small network of Nitrate Sensors to support the proposed marketplace platform, and we have additional PhD student support funding to continue this research well into the future.

D8.4 10 Career-best Research Outputs

1. Alex Potanin et al. “Immutability”. In: *Aliasing in Object-Oriented Programming. Types, Analysis and Verification*. Ed. by Dave Clarke, James Noble, and Tobias Wrigstad. Vol. 7850. 2013, pp. 233–269

2. Tobias Runge et al. “Information Flow Control-by-Construction for an Object-Oriented Language”. In: *Software Engineering and Formal Methods (SEFM) 2022*. 2022, pp. 209–226

3. Isaac Oscar Gariano, Marco Servetto, and Alex Potanin. “Using capabilities for strict runtime invariant checking”. In: *Sci. Comput. Program.* 224 (2022), p. 102878

4. Jens Dietrich et al. “Evil Pickles: DoS Attacks Based on Object-Graph Engineering”. In: *31st European Conference on Object-Oriented Programming (ECOOP) 2017*. Ed. by Peter Müller. Vol. 74. 2017, 10:1–10:32

5. Darya Melicher et al. “A Capability-Based Module System for Authority Control”. In: *31st European Conference on Object-Oriented Programming (ECOOP) 2017*. Ed. by Peter Müller. Vol. 74. 2017, 20:1–20:27

6. Cyrus Omar et al. “Safely Composable Type-Specific Languages”. In: *ECOOP 2014 – Object-Oriented Programming*. Ed. by Richard Jones. 2014, pp. 105–130

7. Marco Servetto et al. “The Billion-Dollar Fix”. In: *ECOOP 2013 – Object-Oriented Programming*. Ed. by Giuseppe Castagna. Vol. 7920. 2013, pp. 205–229

8. Alex Potanin, Monique Damitio, and James Noble. “Are Your Incoming Aliases Really Necessary? Counting the Cost of Object Ownership”. In: *Proceedings of the 2013 International Conference on Software Engineering*. 2013, pp. 742–751

9. 🏆 Yoav Zibin et al. “Object and reference immutability using Java generics”. In: *ACM SIGSOFT Symposium on The Foundations of Software Engineering*. 2007, pp. 75–84

10. Alex Potanin et al. “Scale-free Geometry in Object-Oriented Programs”. In: *Communications of the ACM* (May 2005)

D8.5 Evidence of Research Impact

Throughout my career, I worked with small startups, including being employee number 3 at Innaworks in 2006, where I pioneered language translation research [1] between different platforms. The well-known “The Performance of Open Source Applications” book cites my research [2] with my Master’s student Jan Larres that revolutionised the performance evaluations in Talos and similar systems from the early 2010s.

My work over the first decade of my career involved ownership and immutability and how to provide usable language support for both with the help of type parameters. The Rust programming language has widely adopted my approach as “lifetime parameters”. In the second decade, I worked on designing and producing a usable and secure programming language called Wyvern that utilises object capabilities and effects. A popular configuration language called CUE is used widely within Alibaba’s cloud and service configuration. CUE based its module system design on the Wyvern modules.

Associate Professor Alex Potanin, *brings his expertise in type systems, secure language design, language usability, and software verification to lead the “Verifiable Systems Programming Languages” objective. He will also contribute to the “Foundations of Trustworthy Systems” with his type systems background and to “Techniques for Automated End-to-End Verification” with his program synthesis background.*

[1] https://patents.google.com/patent/EP2122464A4

[2] https://aosabook.org/en/posa/talos.html