Verified Trustworthy Software Systems

Annual Report

2022/23

Contents

Foreword

VeTSS has been very impactful at a number of levels since its launch. It has a strong history of not only funding groundbreaking and important work in the field, but stimulating further research, supporting and strengthening the verification community. This work has had significant influence on a national and international level, modernising software testing, analysis and verification techniques to make the UK a safer place.

VeTSS prides itself on bringing together researchers across the country and encouraging collaboration, and in that light, it only seems fitting to welcome the first joint directorship of a UK Research Institute between Professor Brijesh Dongol from The University of Surrey and Doctor Azalea Raad from Imperial College London. They have both been involved with the VeTSS community for a number of years and bring a wealth of knowledge and experience to the roles, along with a strong vision for how to lead the Research Institute through the next few years.

As the NCSC internal lead for VeTSS, I am proud of the work done by VeTSS over the years and am excited for what they will be able to achieve in the future, further reinforcing software reliability to reflect modern contexts and being at the forefront of research developments in these areas.

Adam W1, NCSC

Directors' Message

We are delighted to have been appointed as the next directors of VeTSS, taking over from a highly successful Research Institute under the Directorship of Philippa Gardner, with wide-ranging impact on programming languages (including ISO standards) and software verification tools. Many of the projects funded under the previous VeTSS scheme seeded much larger projects (funded through ESPRC and Horizon Europe) and helped support the careers of many early-career researchers who subsequently secured senior roles in academia and industry. This will certainly be a hard act to follow!

Our vision for this next VeTSS iteration is to be *by the community, for the community*. To this end, we have assembled an Advisory Board as VeTSS stakeholders to help us steer the VeTSS vision, expand its reach and increase its impact. In close collaboration with our Advisory Board and with members of the VeTSS community, we are proud to have led the development of a VeTSS Problem Book as a focal point for verification research in the UK, with experts providing research challenges and topics of national strategic importance.

Our engagement strategy involves an improved online presence through a revamped website and social media presence on Twitter and LinkedIn, and a dedicated Zulip server open to all members of the VeTSS community. We have organised outreach activities and supported workshops run by VeTSS community members. We have expanded our networks by building new links with the NCSC (Netherlands) and DARPA, providing us with an international perspective on verification problems, as well as collaboration opportunities for UK researchers. We have also strengthened our links with the other NCSC Research Institutes (RISE, RITICS and RISCS) by identifying key cross-cutting problems.

We hope to continue to steer VeTSS as a force for supporting verification research and maximising its impact, while recognising the ever-changing computing landscape with growth in computing power and verification technologies enabling researchers to tackle scalability and maintainability challenges, recognising the UK's diversity and strength in verification.

Brijesh Dongol

Azalea Raad

Overview of 2022/23

The Research Institute on Verified Trustworthy Software Systems (VeTSS) is one of four research institutes within the National Cyber Security Centre (NCSC), bringing together leading academics, industry professionals, regulators and government representatives to ensure research into software stays focused on the safety and security challenges faced by the UK. VeTSS supports research in software and systems analysis, testing and verification and stands at the forefront of research developments in fundamental theories and verification tools, targeting practical applications in the real world. Ultimately, VeTSS aims to *increase assurance in safety and security*, covering both critical software and general-purpose (aka commodity) software at an industrial scale.

In close collaboration with our Advisory Board and an independent Expert Review Committee we compiled the first version of the VeTSS Problem Book, highlighting research topics and challenges of strategic importance to the UK.

We have hosted and sponsored several events, including the VeTSS inaugural annual conference and the VeTSS summer school. The primary objective of the VeTSS annual conference is to establish connections with VeTSS-adjacent fields (e.g. quantum computing and AI), facilitate discussions on cutting-edge research, address open challenges and explore collaborative opportunities within the VeTSS community.

The VeTSS Summer School 2023, held at the University of Surrey in August 2023, provided training to students and offered networking and socialising opportunities for participants. VeTSS also sponsored the “Fun in the REPL” event in November 2023, a one-day joint meeting of the Fun in the Afternoon and S-REPLS communities, organised by colleagues at the Programming Languages Research Group Bristol.

Three VeTSS-aligned projects successfully secured funding from the EPSRC: TRUSTED, M4Secure and SACRED-MA. TRUSTED aims to provide provable security guarantees for modern complex software systems utilising third-party software. M4Secure focuses on creating an open-source framework for customisable memory management libraries. SACRED-MA enables a step-change in our understanding of RDMA (Remote Direct Memory Access) systems via a foundational approach to verifying safety and security.

Building on past success, we aim to strengthen collaborations with industry partners and regulatory authorities to ensure the trustworthiness of software systems across diverse application domains. Preparations are underway for VeTSS Annual Conference 2024, Summer School 2024, Advisory Board meeting and an Industry Sandpit meeting. We look forward to your attendance at our upcoming events.

VeTSS Problem Book

The aim of the VeTSS Problem Book is to present VeTSS problems to the general community (academia, industry and government) and explain why these problems are important and why they motivate people. Our aim is to help researchers understand, improve, and deliver the impact of their existing work, and connect with others working on adjacent topics. The focus is ***not*** on what has been achieved, but on describing what we ***want*** to achieve, and how we would like to ***grow*** the field of verification. This will allow us to develop a research community that focuses on specific problems or can be encouraged to focus on them.

There is no prescribed timescale for solving these problems (short vs long term research). Namely, impact is important but not immediately needed -- a long-term project with a defined impact is equally good in terms of funding. We aim to push for outputs that can be taken up by officials / white papers, if they are backed up by scientific justification. However, we cannot advocate for a particular method as it may not work for all settings.

The VeTSS Problem Book is a living document. As such, an external mechanism will be set up to take further input from community members; minor modifications will be addressed by the VeTSS Directors, and more substantial changes will be discussed at subsequent Advisory Board meetings.

The next two sections provide an overview of the main highlights of the problem book. The full version will be published on the VeTSS webpage.

VeTSS Advisory Board

Jade Alglave (UCL and Arm); Robert Ashmore (Defence Science and Technology Laboratory, DSTL); Sophia Guerra (Adelard); Ekaterina Komendantskaya (University of Southampton and Heriot-Watt University); William Martin (Defence Advanced Research Projects Agency, DARPA); Peter O'Hearn (Lacework); Alastair Reid (Intel); Peter Sewell (University of Cambridge); Greg Smith (EPSRC); Greta Yorsh (Jane Street); Marta Kwiatkowska (Oxford University)

Problem Book External Review Committee

NCSC; Ana Cavalcanti (University of York); Matthew Hill (DSTL); Chris Hankin (Imperial College); Steve Schneider (University of Surrey); John Wickerson (Imperial College)

Problem Book Overview: Verification Dimensions

Facilitating the development of verification tools that can be integrated into existing development environments involves simplifying the ease of use and setup (including integration with existing programming environments) and streamlining the learning curve. Trade-offs here may include scalability of the tool vs its precision and/or generality. We have identified a need to explore verification technologies across different dimensions of focus: lightweight vs heavyweight techniques, the compute stack, and the readiness level of a particular verification technology.

Ease of Use vs Strength of Guarantees

Verification tools and techniques vary widely in how easy they are to use; specifically, whether a tool/technique can be used out of the box as a push-button approach without any required training or instrumentation/annotation (e.g. user-defined specifications). Examples of such push-button tools include the influential open-source Infer/Pulse platform developed at Meta and used widely in-house as well as in big-tech companies such as Amazon Web Services (AWS). At the other end of the spectrum lie techniques such as fully mechanised proofs, where the user needs to fully specify the desired behaviours and mechanise their proofs in an interactive theorem prover such as Isabelle/HOL, Lean, Rocq (formerly Coq).

The scalability of push-button tools makes them ideal for industrial settings with large teams of developers who cannot afford the steep learning curve of carrying out mechanised proofs. However, in general, the ease of use of a verification tool/technique is typically in inverse correlation with the strength of the guarantees it provides. For instance, Infer/Pulse mostly focus on memory safety issues (e.g. the absence of null pointer dereferences), and are limited to sequential programs (not accounting for concurrent code). On the other hand, using mechanised techniques one can prove full functional correctness of a given piece of code, albeit at the high time/training cost.

These techniques do not *compete*, but rather *complement* one another, and there is undoubtedly great value in employing diverse tools/techniques spanning this spectrum. For instance, while tools such as Infer/Pulse are highly suitable for large development teams as part of the CI/CD loop in codebases that evolve rapidly, it is more desirable to fully verify (using a mechanised proof) critical software, e.g. a micro-kernel, whose code is not subject to frequent/immediate change.

The Compute Stack

A second dimension of verification research is clarifying its area of focus by placing it in the context of the Compute Stack (Fig. 1), ranging from low-level hardware such as logical gates to operating systems to high-level applications. Clear definitions of such a stack enables a separation of concerns at a specific area of interest, supporting modularity. Moreover, it provides a pathway towards *co-verification* and *co-design* techniques, where proofs and designs at one layer of the stack informs another, and vice versa.

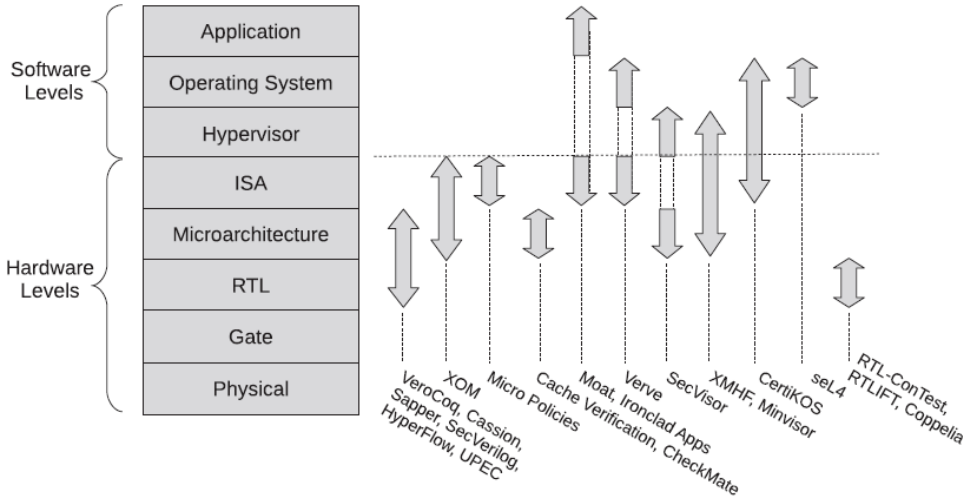


Fig 1. Figure from Erata et al.

Despite years of progress, verification technology still has a long way to go. Even large-scale projects focus on subcomponents of a system, or a specific (often intricate) aspect, whose correctness may be difficult for humans to judge. Often, one needs to make assumptions about intermediate layers (e.g. the operating system or hypervisor) to enable proofs at higher levels of abstraction. Understanding the gaps allows one to articulate the precise guarantees more clearly, and answer questions such as the role of a verified component within an unverified system.

Verification Technology Readiness Levels

We have identified that verification tools and technique have different levels of readiness (for industrial deployment). To this end, we introduce the notion of a *Verification Technology Readiness Level* (VTRL), taking inspiration from the well-used notion of a Technology Readiness Level (TRL), e.g. as employed at NASA. By identifying these levels, we emphasise that there is value in conducting research at different VTRLs, and that one way of achieving impact is by progressing through them, bringing the techniques and technologies closer to wider adoption.

TRL	NASA usage	VTRL description
1	Basic principles observed and reported	Proof principles (logics and semantics)
2	Technology concept and/or application formulated	Verification frameworks, proof support and/or compositional reasoning methods
3	Analytical and experimental critical function and/or characteristic proof-of concept	Experimental verification on litmus tests and proof-of-concept examples
4	Component and/or breadboard validation in laboratory environment	Mechanised verification on key litmus tests, proofs of concept
5	Component and/or breadboard validation in relevant environment	Mechanised verification on isolated industrial-strength examples (lab conditions)
6	System/subsystem model or prototype demonstration in a relevant environment	Automation, reusability and reproducibility on multiple large-scale case studies
7	System prototype demonstration in a space environment	Integration with existing verification and/or development environments
8	Actual system completed and "flight qualified" through test and demonstration (ground or space)	Verification of systems in a deployed (operational) environment
9	Actual system "flight proven" through successful mission operations	Proof maintenance and robustness of deployed system

We anticipate different VTRLs to be tackled by different groups of researchers. Levels 1-3 involve foundational work, typically characterising theoretical research and development of proofs-of-concept in academia. Levels 4-6 involve technology-transfer initiatives supported by academia-industry collaborations and large-scale case studies. Levels 7-9 involve development and adoption of verification technologies steered by industry demands.

Problem Book Overview: Research Themes

These dimensions cover key verification themes including resilience, protocols, software systems, programmer support and proof/program robustness.

Theme Description	*Example Research Questions*
Verified resilience refers to a system or software component that has been rigorously analysed or proven to be capable of withstanding and recovering from various types of disruptions, such as hardware failures, network outages, attacks, or other unexpected events. Such systems may be input-controlled, or more increasingly, autonomous, and self-regulating. Resilience not only covers a system's operations during deployment, but over the lifetime of the system's operation.	• How can we verify resilience, including recovery, degraded services and antifragility? • How can verification be incorporated into "resilience cases"? • How can recovery be incorporated into resilience verification? • How can AI components in resilient systems be verified, or incorporated into verifying resilience? • How do future and emerging technologies (e.g. AI) impact this theme?
Verified protocols aims to provide assurance that security protocols and their associated mechanisms are correct through formal verification to ***(a)*** ensure adherence to well defined security properties (e.g. agreement, authentication etc), and ***(b)*** protection of sensitive information and integrity of data. The work may be at the level of design or implementation of the protocols. This may include use of specialised hardware, e.g. Trusted Platform Modules (TPMs), Trusted Execution Environments (TEEs). This theme is concerned with approaches to *describing* and *modelling* protocols, as well as verifying their *design* and *implementation*, which may be performed in tandem so that the analysis of each is informed by the other.	• How can we bridge the chasm between theoretical models and practical implementation of real-world code? • How can we lower the expertise required for using protocol verification tools, facilitating and increasing their use? • How can we improve the scalability of protocol verification, e.g. to verify larger, highly stateful protocols such as signal? • How do future and emerging technologies (e.g. quantum computing) impact this theme? For instance, what do we need to do to enable verification of protocols using post-quantum primitives?

Verified software systems involves confirming that software behaves as specified, which may address functional correctness, safety (including memory safety) and proof and program maintainability and robustness. Depending on the program being verified, a proof may need to consider a program's context and/or operating environment. Verification may additionally be assisted by static, and runtime guarantees provided by a programming language (and compiler), the underlying type system, and/or domain-specific assumptions.	• How can we combine verified and unverified components into an integrated system, and what guarantees do they provide? • How do future and emerging technologies (e.g. LLMs) impact this theme? For instance, how can AI be used to drive verification tools? • What are the underlying software specifications, and how can they be communicated to developers? • How can verification effectively combine hardware and software guarantees to ensure robust system-level assurances?
Programmer and language support involves the development of techniques that enable generalist programmers to integrate (aspects of) verification into daily programming tasks. Programmers often do not require full functional correctness of the programs that they write. However, there can be many benefits to supporting lightweight verification or using formal techniques to aid correct-by-construction development, aka verification for the masses/verification at scale.	• How can we expose the implicit assumptions within a specification or implementation? How can we do this via effective tooling for generalist programmers? • How can we facilitate adoption of verification technologies in practical development environments, e.g. integrated into CI/CD? • How can we manage frequent and sometimes rapid software change? • How do future and emerging technologies (e.g. LLMs) impact this theme? For instance, how can AI technologies be used for code generation?

Proof Robustness, Maintainability and Repair refers to the preservation of correctness of programs and proofs in a system under change. This addresses the real-world problem of ensuring that the deployed version of software matches the verified version, where it is important to ensure that the proofs of verified software under change are not destroyed when the software is modified. Most of the real-world software development seldom involves programming from scratch --- often the core development effort is on maintaining or modernising existing codebases.

- How can formal methods / tools enable rapid proof maintenance or re-verification of software under change?
- How can formal methods / tools enable rapid program repair or patch synthesis for software under change?
- How can differences in different versions be documented in a human readable manner?
- Can we generate natural language explanations of tool-suggested proof / program repair steps**?**

VeTSS Events

Inaugural Meeting (March 2023)

The VeTSS Inaugural Conference was held at the British Computer Society, London comprising a series of talks and panel discussions to discuss the state-of-the-art research and its open challenges and seek cross-pollination opportunities with across VeTSS themes.

VeTSS: From Old to New
Prof Philippa Gardner
Imperial College London

On Operational Cyber Resilience
Prof Kerstin Eder
University of Bristol
https://youtu.be/ejhCsPIbbU4

Automatic Verification of Transparency Protocols
Prof Mark Ryan
University of Birmingham
https://youtu.be/ODgtHejy9wo

Security and Legacy at Microsoft
Dr Matthew Parkinson
Microsoft Research
https://youtu.be/9me40Pv6W_Y

Benchmarking and Verifying Quantum Computers
Dr Petros Wallden
The University of Edinburgh

Formal Verification and Bug Finding at Meta
Dr Jules Villard
Meta

The Research Institute for Trustworthy Interconnected Cyber-physical Systems
Prof Chris Hankin
Imperial College London
https://youtu.be/9ydCQMvaLjI

Design and Verification of Time-Critical Byzantine Fault-Tolerant Systems
Dr Vincent Rahli
University of Birmingham

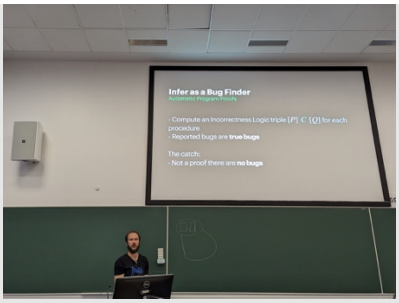
CsmithEdge: More Effective Compiler Testing by Handling Undefined Behaviour Less Conservatively
Dr Karine Even-Mendoza
King's College London
https://youtu.be/-4GgMGFB_nY

Rust on Morello
Dr Simon Cooksey
University of Kent
https://youtu.be/QLy5uTnMJ_c

Summer School (August 2023)

The VeTSS 2023 Summer School focused on program analysis, testing and verification and was held at the University of Surrey, Guildford, UK, from Tuesday 22nd to Thursday 24th August 2023. Around 40 students cross UK universities attended the event.

Algebraic Semantics and Verification
Prof Georg Struth
University of Sheffield
https://youtu.be/5MS_n8Y5vEI



Build your Own Scalable Static Analysis with the Infer Platform
Dr Jules Villard
Meta
https://youtu.be/6A9w8tX-rMg

Inductive and coinductive theorem proving in Isabelle
Dr Andrei Popescu
University of Sheffield
https://youtu.be/ZDEN7WMIF2w

C Bounded Model Checker (CBMC)
Prof Daniel Kroening
Amazon Web Services

Neural Network Verification with Vehicle
Prof. Ekaterina Komendantskaya
Dr. Luca Arnaboldi
Dr. Matthew Daggitt
University of Southampton
University of Birmingham
https://youtu.be/-ZRE1BgaCk0

Outreach Activities: Fun In The REPL

VeTSS was one of the sponsors of “Fun in the REPL”, a one-day joint meeting of the Fun in the Afternoon and S-REPLS communities, organised by our colleagues at the Programming Languages Research Group Bristol. The meeting was held at the Engine Shed, on Wednesday 1st November. Full details can be found on their website (plrg-bristol.github.io/fir).

The South of England Regional Programming Languages Seminar (S-REPLS) is a regular meeting based in the south of England, open to all who have an interest in the semantics and implementation of programming languages. Fun in the Afternoon is a seminar on functional programming and related topics. Bob Atkey from The University of Strathclyde presented on Data Types with Negation in one keynote, while Ákos Hajdu from Meta/WhatsApp discussed Static and Dynamic Code Analyses for WhatsApp server in another keynote. Additionally, ten Contributed Talks were delivered by experts representing diverse universities and industrial companies.

VeTSS-Aligned Projects (2022/23)

In 2022, the ESPRC, NCSC and DSTL together invested on a set of projects aligned with the NCSC Research Institutes (ukri.org/opportunity/research-aligned-with-cybersecurity-research-institutes/). Within this call, three VeTSS-aligned projects were funded, which together received £3.4M of investment.

Making Memory Management More Secure (M4Secure)

Jeremy Singer
University of Glasgow

Alice Miller
University of Glasgow

Zheng Wang
University of Leeds

M4Secure will create an open-source framework for customisable memory management libraries that meet formal user requirements for security and performance.

Introduction. Memory bugs are a major cause of application security vulnerabilities, making up 70% of critical software flaws, often resulting from mismanagement of dynamic memory. Effective memory allocation is essential for efficient execution, as a significant portion of program execution time deals with low-level memory management. Hence memory allocation is vital for both security and performance. Recent hardware innovations, such as secure CPU extensions, offer improved security by preventing various hacker exploits. However, leveraging these innovations currently requires extensive software developer effort.

Instead of labour-intensive manual development, M4Secure will develop an automated approach that synthesizes efficient memory management code based on specified attributes, including security properties. The project will ensure code correctness through the application of model-checking techniques, accommodating a wide range of processor architectures, including hardware security features like Arm Memory Tagging Extension (MTE), CHERI capabilities, and Intel Control-flow Enforcement Technology (CET).

Future Plans. A key enabling technology of M4Secure is the combination of deep learning and formal verification techniques to generate memory allocator code to match security specifications, guaranteeing correctness and security. This strongly aligns with the research vision of VeTSS by developing techniques to model hardware security features and reason about software behaviour. This work will provide novel scientific methods and a new, exciting application domain to support the development of trustworthy and high-performance software systems through verification and deep learning techniques.

SecuriTy SummaRies for SecUre SofTwarE Development (TRUSTED)

Narges Khakpour
University of Newcastle

Steven Schewe
University of Liverpool

Dominik Wojtczak
University of Liverpool

TRUSTED will provide provable security guarantees of modern complex software systems that make use of third party open-source software.

Introduction. TRUSTED will allow users to employ open-source packages safely by developing the concept of security summaries of code – which states when it is secure to use the software and what it does in terms of security – and establishing how to reason about the security of the composition of different software components solely based on their summaries. This will guarantee the overall security of the software system. Thereby, we will answer the fundamental questions "what does it do?" and "does it behave as intended?" about the software, allowing the user to rest assured that the system s/he uses is secure.

Future Plans. We will address transparency by building an automatic system for formally and irrefutably establishing safety and security of the system. We will also address usability because our system can be used by non-experts, and our technology can be used in a frictionless fashion from the initial product design phase right through to its use in a complex system. Finally, our project will enable a safe use of "commodity technology with appropriate security" as it enables the use of open-source software components with confidence that it would not introduce security flaws into our system. This will help the UK to fulfil the purpose of our GOV.UK national cyber strategy 2022 by ensuring that the UK remains confident, capable, and resilient in this fast-moving digital world.

Safe And seCure REmote Direct Memory Access (SACRED-MA)

Brijesh Dongol
University of Surrey

Azalea Raad
Imperial College London

Gregory Chockler
University of Surrey

SACRED-MA enables a step-change in our understanding of Remote Direct Memory Access systems via a foundational approach to verifying safety and security.

Introduction. Modern society depends on accessing and transferring vast quantities of data, at ever-increasing speeds. Technology giants such as Google invest billions of dollars every year into data centres across the world. Replicated data systems form the backbone of all cloud services; improving their reliability and performance impacts all cloud and big data services.

To meet our ever-growing need for rapid data transfer, RDMA (remote direct memory access) technologies enable next-generation infrastructures by allowing a machine to access (read/write) directly the memory of another machine across a network. Unlike traditional network protocol stacks such as TCP/IP, RDMA-enabled network interface cards (NICs) can bypass an operating system kernel, and are thus capable of wire-speed data transmission. RDMA technology has been available in supercomputing clusters since the mid 2000s, but had until recently remained an experimental feature in consumer and enterprise systems due to its cost. However, this changed recently with the availability of affordable NICs (e.g. those developed by our partner NVIDIA), and it is now possible to build distributed applications that challenge conventional design paradigms. For instance, one can leverage the additional throughput of RDMA to support concurrent front-end applications, surpassing sequential state-machine replication services used today.

Future Plans. In SACRED-MA, we build up from the hardware models to distributed applications. We take a formal approach at all levels of abstraction and develop / adapt new proof methods to address key design challenges, which aims to support programmers and future developers of RDMA-based systems. RDMA-based technologies themselves are game-changing, providing wire-speed networking capabilities, and have the potential to transform the way we build inter-connected systems.

EDI Statement

Equality, diversity and inclusion (EDI) are core values underpinning our research and innovation aims. We created VeTSS first ever EDI statement this year. We believe that creative and innovative research needs diversity, inclusion and equality of access, and that as a Research Institute, we have a responsibility to identify and remove current barriers to participation. We want to create an environment that fully values and respects individuals from all backgrounds, cultures, identities and abilities and their contribution.

Our commitment to EDI is crucial in driving innovation and excellence in our work, but also will allow us to engage with and benefit a larger part of the research community and to contribute to a fairer, more equal society. In this statement we outline our goals and priorities and the practical steps that were taking. We will ensure that individuals of all races, ethnicities, genders, sexual orientations, abilities, socio-economic backgrounds and identities feel included and have equal opportunities to thrive within VeTSS. We will take practical steps to improve the opportunities and experiences of currently under- represented groups. We will aim to build a research community that reflects the diversity of the society we live and work in and that brings together individuals with diverse perspectives and expertise. We will identify barriers to opportunities and will work to remove them. We will align with the EDI strategies of our hosting institutions, the University of Surrey and Imperial College London and regularly measure our progress against our goals. We will foster a respectful and inclusive culture by encouraging open dialogue, active listening, and respectful communication. We will promote an environment where diverse ideas are valued, and individuals are supported in expressing their opinions without fear of discrimination or bias. We will collaborate internally and externally to develop good practice. To this end, we will actively seek partnerships and collaborations with organizations that adhere to these principles, to collectively work towards a more inclusive research community.

Our priorities include following three areas:

- Promoting Diversity and Inclusive Culture: We aim to foster an inclusive culture by encouraging open dialogue, active listening, and respectful communication.
- Equal Opportunities: We will identify barriers to opportunities and will work to remove them. We work to widen participation and improve the diversity of our community.
- Education and Training: We are committed to create a working environment free of bullying, harassment, victimisation and unlawful discrimination.

We understand that this is an ongoing process that requires continuous effort and accountability and we are committed to regularly reviewing our practices and work against our aim to make VeTSS an environment that fully embraces equality, diversity and inclusion. This year Program manager Teresa attended following course at Imperial College London: Managing Unconscious Bias in the Workplace, Confronting Inappropriate Behaviour (Bullying and Harassment), Disability and Neurodiversity awareness training, LGBTQ+ inclusion and mental health and Mental health and race in the workplace. Project manager Ling attended the Anti-Bias Workshop at University of surrey. Two directors plan to attend EDI related training courses next year.

VeTSS Online

We have updated several aspects of the VeTSS online platform. These are currently maintained by the VeTSS program managers and Teresa Carbajo-Garcia (Imperial) and Ling Zhang (Surrey).

Website

We have updated the VeTSS website (vetss.org.uk) and extended it to serve as the first port-of-call for all things related to VeTSS, reflecting a comprehensive archive of past events (e.g. video recordings of the talks at the VeTSS inaugural conference), promoting upcoming events organised by VeTSS and the wider community, as well as a "living" list of publications by researchers funded through VeTSS or VeTSS-aligned calls.

Twitter

We have created a VeTSS Twitter account (twitter.com/VetSS_RI) to promote VeTSS events and publicise work within the VeTSS community.

Zulip

We have launched a VeTSS Zulip to complement the mailing lists (vetss.zulipchat.com). The Zulip platform is flexible and hence is constantly evolving. New channels include Vacancies and Events, as well as specific private channels for events such as our summer school.