MOSCOW
2022

CATALOGUE OF TECHNOLOGIES

IVANNIKOV INSTITUTE
FOR SYSTEM PROGRAMMING
OF THE RAS

ISP RAS

CATALOGUE
OF TECHOLOGIES

CATALOGUE OF TECHOLOGIES

2022

Ivannikov Institute for System Programming
of the Russian Academy of Sciences

25, Alexander Solzhenitsyn str. 109004 Moscow, Russia
E-mail: scsec@ispras.ru

CONTENTS

SOURCE CODE ANALYSIS, VERIFICATION, TESTING

BINARY CODE ANALYSIS, FUZZING

NETWORK TRAFFIC ANALYSIS

REQUIREMENT MANAGEMENT

INFRASTRUCTURE PROJECTS

NATURAL LANGUAGE PROCESSING

DOCUMENT PROCESSING

APPLICATIONS

2022. THE STRATEGY REMAINS THE SAME

ARUTYUN AVETISYAN

Academician of the RAS,
ISP RAS Director

The 2022 collection presents 25 technologies, which for the first time are divided into thematic blocks. The section “ISP RAS: Ecosystem of Innovation” provides a detailed description of the model of the institute’s development and lists the current areas of work, followed by the annual results of the activities of ISP RAS-based research centers. But first, let us talk about the main achievements of this year.

We are constantly looking for new options to implement our model of work based on the integration of science, education, and innovation. This helps us proceed with our development, adapting to the changing world. This year, we were able to demonstrate a successful cooperative experience in the field of trusted software by applying our tools and worldwide scientific expertise to the analysis of open source projects.

Technology Center for Security Analysis of the Linux Kernel has prepared more than 100 kernel bug fixes, and 53 patches have already been accepted into the main release. These works are very important for cybersecurity of domestic software, and for that reason the development teams of the center are allocated as a separate structural subdivision of ISP RAS. The Trusted Artificial Intelligence Research Center has developed prototypes of the trusted TensorFlow and PyTorch machine learning frameworks; 10 patches have also been accepted by the international community, with ten more currently under review. Over the next three years, we plan to consolidate all work on the trusted system software stack in a separate center and gather a consortium of interested companies around it, which would help consolidate resources and move from competition to cooperation. More than 20 companies are already cooperating with the Linux Technology Center.

We continue to develop our innovative solutions. The Svace static analyzer has improved support for the latest versions of Kotlin, Go, Java and C# languages, and Svacer, the system for handling analysis results, has added programming interfaces for working in CI/CD environments; ISP Crusher has learned to fuzz neural networks, browsers and programs in Python, Java, C#; Talisman makes it possible to create intelligent analytic systems using low-code approaches.

This year's collection includes three technologies not previously introduced. These are Requality requirements management tool, and TestOS testing environment, as well as EcgHub, a system for intelligent analysis of digital ECG using neural network models. At the Kazan Digital Week forum, it attracted special attention from visitors. More than 50 people were examined directly at the exhibition, and in all cases their diagnostic data were successfully confirmed.

In 2022, landmark PhD theses had been defended at our institute – in particular, works on dynamic symbolic execution, which improves the quality of fuzzing and raise the level of software security, as well as on the use of dynamic compilation in databases to accelerate the execution of complex SQL queries. Another senior Ph.D. thesis was dedicated to critical aspects of homomorphic encryption in cloud environments. Our scientific conferences attract hundreds of researchers, students, professors, and industry representatives, and continue to be supported by the IEEE International Association.

This year, our long-term development model has once again proven its sustainability. The institute continued its strong growth. The number of staff has grown by 25%, exceeding 700, and total funding has increased by more than 30%. So in this sense, our strategy remains unchanged: we plan to further enhance the role of the academic community as a moderator of the ecosystem that brings together science, industry, and education, ensuring balance and harmonious development.

ISP RAS: AN INNOVATION ECOSYSTEM

ISP RAS activities are aimed at deploying fundamental research results in industry. The institute's business model consists of three closely related activities producing a synergistic effect:

- project-oriented fundamental and applied research aimed at creating new technologies (under contracts with Russian and foreign companies, the Ministry of Science and Higher Education of Russia, RAS programs, grants from Russian Science Foundation and from Advanced Research Foundation, etc.);
- deploying new technologies in partner companies and developing innovations based on industry feedback;
- educating students and postgraduates based on developed technologies (while participating in the institute's research and industrial projects).

This model of industrial research plus education is well known and applied in research laboratories of leading universities (Stanford, MIT, Berkeley, Carnegie Mellon) and industrial giants (IBM, Intel), as well as in state research centers (INRIA, Fraunhofer). When implemented effectively, the model solves the problem of the gap between science and industry, and produces highly qualified specialists in system programming.

FUNDAMENTAL RESEARCH

Fundamental research and experimental works are necessary elements of the institute's activities, allowing it to move in line with the latest trends in the IT world, as well as generate its own ideas for projects with its business partners. ISP RAS works on a large number of scientific and educational programs and cooperates with leading Russian and foreign universities and scientific centers.

This allows to provide high quality research results, while ISP RAS reputation in academic and university circles makes it possible to introduce domestic technologies to international markets.

ISP RAS publishes its own journal called “Proceedings of ISP RAS”, indexed in the Russian Science Citation Index (RSCI). The institute is also responsible for publishing and editing the RAS journal called “Programming”. Both are included in the journal list of Higher Attestation Commission (the VAK).

DEPLOYMENT

ISP RAS deploys its research results in various industrial and research enterprises, which use and promote the institute’s technologies. Most of the contract work is performed with long-term partners who have been collaborating with ISP RAS for more than five years. Principal international partners include such companies as Samsung and Huawei; domestic companies include Kaspersky Lab, Security Code, Open Mobile Platform, RusBITech, GosNIIAS, VimpelCom, Basalt SPO, Swemel. Currently, the institute’s technologies are used by more than 70 companies; more than 100 companies have received the technologies in 2022 in accordance with the half-year free access program.

SCIENTIFIC COLLABORATION

Long-term cooperation with ISP RAS can be organized in a form of a joint laboratory. Having permanent funding, they allow planning flexibly available resources as well as increasing competencies in the newly emerging areas of system programming and organizing the training of young specialists with the skills needed by partners.

Currently, the institute has joint laboratories with Samsung Electronics (focused on compiler technology including improving security for Android and Tizen OS, as well as on research in AI methods and data analysis for software engineering), and Huawei (the first laboratory works on compiling technologies and OS components, while the other focuses on static and dynamic analysis). There is also a laboratory for solving continuum mechanics problems based on the Fanlight cloud platform. Since 2021, ISP RAS has operated the intelligent digital foresight and media data lab.

The institute also has a linguistic laboratory based on the Lingvodoc platform, documenting endangered languages. This research is carried out jointly with the RAS Institute of Linguistics, Tomsk State University, and other universities and research centers. In 2022, three groups of researchers from several Russian cities were trained within the program “Lingvodoc use for linguists.”

CENTERS

The important mission of ISP RAS is creating and moderating communities. Three such centers have been launched and are currently in operation:

- World-class Research Center (WCRC) “Digital biological design and personalized healthcare,” jointly with Sechenov University, Institute of Biomedical Chemistry, Yaroslav-the-Wise Novgorod State University, Institute for Design-Technological Informatics of RAS;
- Trusted AI Center, jointly with Ministry of Economic Development, academia (MIPT, Skoltech, Medical Scientific Center and the Faculty of Mechanics and Mathematics of Moscow State University, Innopolis University, Lobachevsky University, Psychology Institute of RAS, Joint Supercomputer Center of RAS) and industry (Kaspersky Lab, EC-leasing, InterProCom, Technoprom).
- Technology Center for Security Analysis of the Linux Kernel, jointly with FSTEC of Russia and with active participation of leading Russian IT companies.

Since 2022, two more significant projects have been underway with the support of FSTEC of Russia: one on creating infrastructure for systematic research into the security of critical components that form the basis of Russian distributions of operating systems and other software tools, the other on creating a unified secure development environment.

INTELLECTUAL PROPERTY

ISP RAS business model suggests that IP rights are either retained by the institute or transferred to an open source developer community under special agreements. Taking into account the specifics of this model, ISP RAS developed a unique license based on the direct financing by the customer of the research and development for the licensed technology (instead of paying royalties). The customer gets non-exclusive rights for using the technology, and the institute retains the exclusive IP rights. For some cases, decisions on managing IP rights are made individually based on long-term collaboration perspectives. An example of such an exception is the collaboration with Advanced Research Foundation, which assumes transferring all IP rights to the customer.

OPEN SOURCE

One of the most important components of the created ecosystem is the widely used open source software that is absolutely necessary for modern system programming. Open source is considered as:

- a tool that provides legitimate free access to all modern technologies, including ready-to-use software products and open standards;
- an ability to ensure the institute innovative research without outsourcing contracts but interacting with global market of products and services;
- a powerful educational resource, as the environment and infrastructure of international open source projects can be used to train engineers.

Scientific activity implies the result’s openness and the visibility of its author, which often contradicts IT corporate policies. For ISP RAS, the openness of research results is both motivation for work, and a tool for promoting the institute’s technologies. Open research means that each young researcher is visible in the international IT community. Their contribution and reputation are their capital, and the institute does everything to ensure that this capital grows as quickly as possible.

EDUCATION

The ISP RAS innovation ecosystem cornerstone is educational activity, which is performed in several directions:

- Cooperating with leading universities. Institute specialists are working on system programming departments of MSU, MIPT, and HSE. Starting from their first year, students attend system programming lectures and corresponding practical lessons. In the third year, students join the departments for system programming and, while continuing to attend lectures, start to work in special seminars, get acquainted with the institute's scientific directions, participate in projects and receive a special scholarship. By the time of graduation, many students have scientific publications and become system programming experts.

  ISP RAS researchers are constantly updating education courses and bachelor programs. For example, in 2021 ISP RAS started academic advising to modernize the software engineering bachelor program at the Faculty of Computer Sciences at Higher School of Economics. In 2022, ISP RAS became a partner of the Moscow Aviation Institute MAI on the federal project "Advanced Engineering School," which is devoted to the development of a new generation of aircraft. Opening a joint laboratory and launching a new master's program on UAV is currently in the works. In addition, the institute, together with the MAI and other organizations, joined the "New Aerospace Markets" consortium (within the framework of the "Priority-2030" program).

  Starting from 2017, ISP RAS has been actively working with Samsung at the Samsung IT Academy. In particular, employees of the Institute are on the jury of the Interuniversity Project Competition, which is held annually to demonstrate the best practices and results of educational activities implemented in the Academy's partner universities.

- Scholarship program. In support of educational processes, ISP RAS launched a special scholarship program for students and postgraduates of MSU, MIPT, HSE, Novgorod State University, Russian-Armenian University, etc.
- ISP RAS postgraduate study helps gain practical experience and learn new technologies at the same time. Postgraduates are actively involved in education: they organize seminars and practical classes for students, supervise term papers and theses. With that kind of experience, they usually become leaders of small research groups.
- System programming labs network. Currently, ISP RAS external labs are working in Yerevan, Veliky Novgorod, Orel, Plekhanov Russian University of Economics. The laboratories attract successful students (including postgraduate students), and involve them in the development of promising technologies in close cooperation with industry.

CONFERENCES

ISP RAS organizes a number of annual events:

International ISP RAS Open Conference:
https://www.isprasopen.ru/en

OS DAY, a conference on science and practice (jointly with other organizers): https://www.osday.ru/

International Ivannikov Memorial Workshop:
https://www.ivannikov-ws.org/en

International Conference "Data Science in Medicine" (jointly with other organizers): https://digital-med.ru/en

SYRCoSE Software Engineering Colloquium:
http://syrcose.ispras.ru/

The "System Programming as a Key Direction for Counteracting Cyberthreats" roundtable (International Military-Technical Forum "Army")

2022. WORLD-CLASS RESEARCH CENTER (WCRC) DIGITAL BIODESIGN AND PERSONALIZED HEALTHCARE

JOINTLY WITH SECHENOV UNIVERSITY, INSTITUTE OF BIOMEDICAL CHEMISTRY, YAROSLAV-THE-WISE NOVGOROD STATE UNIVERSITY, AND INSTITUTE FOR DESIGN-TECHNOLOGICAL INFORMATICS OF RAS

MOST IMPORTANT RESULTS OF 2022 INCLUDE THE FOLLOWING:

CLOUD PLATFORM NCMU “DIGITAL BIODESIGN AND PERSONALIZED HEALTHCARE” WAS CREATED AS PART OF THE SERVICES OF THE FIRST PHASE.

These are basic cloud services (virtual servers and block devices on demand, etc.), as well as services for the collection, storage and analysis of large medical data exemplified by web-labs for the analysis of electrocardiograms and histological image data.

The platform is implemented on the basis of the Asperitas cloud environment (ISP RAS). In 2023, the Platform will be certified in accordance with the requirements of regulatory and procedural documents of FSTEC and other regulators. By the end of 2024, it is planned to enable full-scale pilot operation with the possibility of connecting external participants.

The NSMC platform can be deployed on the basis of the ISP RAS cloud infrastructure or third-party cloud infrastructure for current biomedical tasks developed within the NSMC, or adapted to the tasks of other medical domains.

A MOCK-UP OF A 12-CHANNEL ECG MARKUP SYSTEM HAS BEEN DEVELOPED (ECG1.ISPRAS.RU).

High-quality standardized markup based on a predetermined list of pathologies helps achieve a high degree of agreement between specialists. The layout of the markup system has been prepared for integration into the ISP RAS Asperitas cloud platform for transparent scaling of ECG storage and analysis capacities, but it can be also integrated into a third-party cloud ecosystem.

A NEURAL NETWORK MODEL FOR CLASSIFICATION OF 12-CHANNEL ECGS HAS BEEN TRAINED

As part of the partnership with Telemedicine Information Systems LLC, EC-Leasing CJSC, and Technion Institute of Technology (Israel), works on ECG intelligent analysis are being carried out. An agreement was signed with the Ministry of Health of the Republic of Tatarstan on the development of artificial intelligence in the healthcare system. The neural network model of 12-channel ECG classification was trained on the data from different regions (Republic of Tatarstan, Moscow, Veliky Novgorod), integrated in the “United Cardiologist” system in the experimental operation mode and tested on the ECG data of the Republic of Tatarstan.

A NEURAL NETWORK MODEL OF ENDONET CELL NUCLEI DETECTION ON HISTOLOGICAL PREPARATIONS WAS TRAINED

The neural network was trained on EndoNuke marked histological specimens assembled jointly with the partners (PFUR, State Clinical Hospital No 31, V.I. Kulakov Institute of General Medicine, Novgorod State University, and the Research Institute of Human Morphology). The core detection model is embedded in the open software platform for bioimage analysis QuPath using the ISP RAS Fanlight technology. The modified QuPath platform and the open source CVAT image markup system are prepared for integration into the ISP RAS Asperitas cloud platform.

2022. TECHNOLOGY CENTER FOR SECURITY ANALYSIS OF THE LINUX KERNEL

JOINTLY WITH FSTEC OF RUSSIA AND LEADING COMPANIES

PRINCIPAL ACHIEVEMENTS OF 2022 INCLUDE THE FOLLOWING:

MAINTENANCE OF THE LINUX KERNEL BRANCH BASED ON THE STABLE VERSION 5.10 HAS STARTED.

KERNEL RESEARCH TECHNIQUES HAVE BEEN PREPARED, INCLUDING:

- static analysis using the Svace tool;
- system and unit testing (test tools accumulation is going on);
- fuzzing testing using the syzkaller tool; conducting architectural analysis to determine the attack surface;
- conducting analysis of marked (sensitive) data..

AN EXPERT GROUP CONSISTING OF REPRESENTATIVES OF 21 COMPANIES WAS CREATED; WITHIN THAT STRUCTURE:

- principles of the functioning of the Technology Center are being formed;
- over 6000 warnings of the Svace tool have been marked;
- more than 100 kernel bug fixes have been made; 53 patches have already been accepted into the main branch of the kernel.

RECOMMENDATIONS ON KERNEL CONFIGURATION TO IMPROVE KERNEL SECURITY ARE BEING PREPARED.

WORK IS UNDERWAY TO DEVELOP IMPROVEMENTS TO THE KERNEL TO IMPROVE THE SECURITY OF ITS WORK ON DEPLOYMENT AND INITIALIZATION PHASES.

TECHNOLOGY CENTER PARTNERS:

- Aladdin R.D.
- Ideco LLC
- “Basalt SPO” LLC
- JSC Baikal Electronics
- BELLSOFT LLC
- JSC IVK
- JSC InfoTeX
- ITB LLC
- “Kod bezopasnosti” LLC
- “Confident” Ltd.
- JSC NTTs “Module”
- JSC MCST
- Open Mobile Platform LLC
- RASU JSC
- RED SOFT LTD.
- RusBITech-Astra LLC
- JSC MVP “SVEMEL”
- NTC ITS ROSA LLC
- “Factor TS” LLC
- JSC FINTEKH
- “YANDEX.CLOUD” LLC

2022. TRUSTED AI RESEARCH CENTER

JOINTLY WITH THE MINISTRY OF ECONOMIC DEVELOPMENT AND THE ACADEMIA COMMUNITY (MIPT, SKOLTECH, MEDICAL SCIENTIFIC CENTER AND FACULTY OF MECHANICS AND MATHEMATICS OF MOSCOW STATE UNIVERSITY, INNOPOLIS UNIVERSITY, LOBACHEVSKY UNIVERSITY, PSYCHOLOGY INSTITUTE OF RAS, JOINT SUPERCOMPUTER CENTER OF RAS) AND INDUSTRY (KASPERSKY LAB, EC-LEASING, INTERPROCOM, TECHNOPROM).

HARDWARE AND SOFTWARE INFRASTRUCTURE WAS CREATED TO PROVIDE TRUST IN THE UNDERLYING MACHINE LEARNING FRAMEWORKS (TENSORFLOW, PYTORCH), WHICH IS USED FOR:

- initial analysis of the source code;
- continuous verification of new code changes;
- synchronization with the original open versions;
- ten patches have been adopted in main versions of the frameworks, with ten more under review.

PROTOTYPES OF TRUSTED MACHINE LEARNING FRAMEWORKS HAVE BEEN DEVELOPED; TESTING IS PERFORMED BY THE CENTER'S INDUSTRIAL PARTNERS.

SOFTWARE PROTOTYPES HAVE BEEN CREATED WHICH IMPLEMENT THE FOLLOWING:

- methods to detect and counter adversarial attacks on machine learning models;
- methods to detect and thwart attacks on machine learning models by embedding backdoors and hidden malicious code;
- methods for explaining and improving the interpretability of machine learning models;
- methods for evaluating the effectiveness of applied systems.

1 PROGRAM ANALYSIS AND CYBERSECURITY

SOURCE CODE ANALYSIS, VERIFICATION, TESTING

BINARY CODE ANALYSIS, FUZZING

NETWORK TRAFFIC ANALYSIS

REQUIREMENT MANAGEMENT

ASTRAVER TOOLSET: A VERIFICATION TOOLSET

AstraVer Toolset is a deductive verification system for key software components. It allows developing and verifying security policy models as well as proving the correctness of software modules written in the C programming language. AstraVer is essential for ensuring the required trust levels from ADV_SPM and ADV_FSP assurance families as defined in the ISO/IEC 15408 standard.

FEATURES AND ADVANTAGES

AstraVer Toolset is a set of tools designed for industrial use. It is based on many years of scientific research and combines two verification approaches: at the model level and at the code level. Parts of the AstraVer Toolset are similar to Microsoft VCC and Frama-C WP, but unlike those AstraVer is specifically designed to support the key security components' verification in the Linux kernel. AstraVer Toolset is free and open source, available at http://linuxtesting.org/astraver.

AstraVer provides:

- An integrated approach to verification, supporting the formalization of high-level requirements and analyzing the C source code behavior.
- Modeling and formalizing functional requirements, proving internal consistency and unreachability of insecure states.
  Testing whether functional requirements are satisfied in an implementation, using their formal models to check the correctness of the observable behavior and to evaluate the quality of testing and generated test cases.
- Verification of critical components written in C (requirements' formalization, correctness proof on all possible input values).
  Support for real industrial C code (GCC compiler extensions, arithmetic operations with bitwise precision, address arithmetic including the container_of intrinsic, function pointers, casting).
- Adhering to the protection profile requirements (ISO/IEC 15408), such as
  - formal security policy modeling;
  - formal verification of internal consistency of a security policy model;
  - formal proof that the target system cannot reach an insecure state;
  - a formal or a semi-formal functional specification development;

  - a formal/semi-formal proof of correspondence between the security policy model and the functional specification; a formal/semi-formal proof of correspondence between different representations of target software, like functional specification, design and source code.
- Ability to adjust the toolset for a specific customer to perform the C source code components verification.

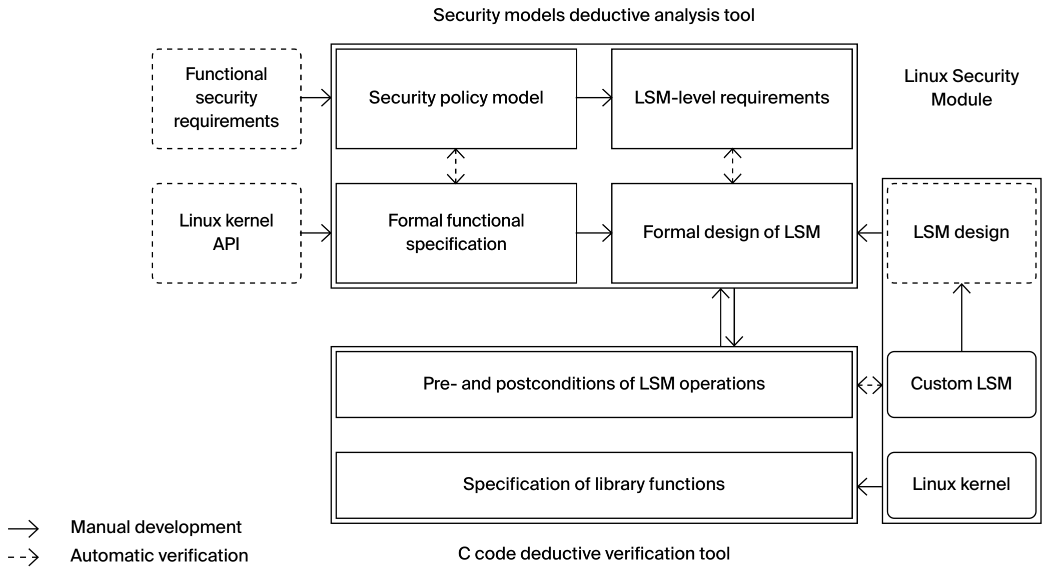
WHO IS ASTRAVER TARGET AUDIENCE?

- Companies developing critical systems, including software in aviation, railway, medical and nuclear power industries.
- Companies that need to certify their software as guided by the ISO/IEC 15408 standard.
- Certification laboratories for information protection software.

ASTRAVER DEPLOYMENT STORIES

AstraVer Toolset was used in the development of access control mechanisms for Astra Linux Special Edition (RPA RusBITech JSC). As a result, this Astra Linux edition has passed the certification for compliance with the FSTEC information security requirements, which are defined for operating systems of the 2A protection profile. Both the security policy model and the access control mechanisms source code were successfully verified using AstraVer Toolset. The verification work for the new security model features is constantly ongoing.

ASTRAVER WORKFLOW


Security models deductive analysis tool
Functional security requirements
Security policy model
LSM-level requirements
Linux Security Module
Linux kernel API
Formal functional specification
Formal design of LSM
LSM design
Pre- and postconditions of LSM operations
Custom LSM
Specification of library functions
Linux kernel
Manual development
Automatic verification
C code deductive verification tool

KLEVER: INDUSTRIAL SOFTWARE MODELS VERIFICATION SYSTEM

Klever is a framework for verifying models that are automatically extracted from large software systems' source code written in the C programming language. Klever allows specifying various security and safety requirements and verifying them automatically with the preconfigured precision level.

FEATURES AND ADVANTAGES

Klever is a result of advanced research and development in the field of automated extraction and verification of program models. The framework base includes per-component verification, environment modeling, and requirements specification methods. This allows applying formal methods to the industrial software of hundreds of thousands or millions of lines of the C source code. Klever is an open-source project (https://forge.ispras.ru/projects/klever).

Klever provides:

- Thorough sound analysis of industrial software (allows detecting all possible errors of specified types and proving program correctness under explicitly stated assumptions).
- Scalability. Modular program verification allows applying the most rigorous program analysis methods to the large code base. The methods are model checking and symbolic execution.
- Adapting software verification framework to customer needs. Developing specifications for modeling target programs' environments and for detecting violations of program specific requirements. This specific customization is performed in addition to checking regular safe programming rules for the C language.
- Comprehensive representation of found faults. When an error is detected, the verification system provides the detailed error trace that includes concrete variable values and called functions' arguments.
- A convenient multi-user web-interface for setting and running verification and for expert analysis of verification results.

WHO IS KLEVER TARGET AUDIENCE?

- Companies developing safety-critical and security-critical software.
- Certification laboratories.

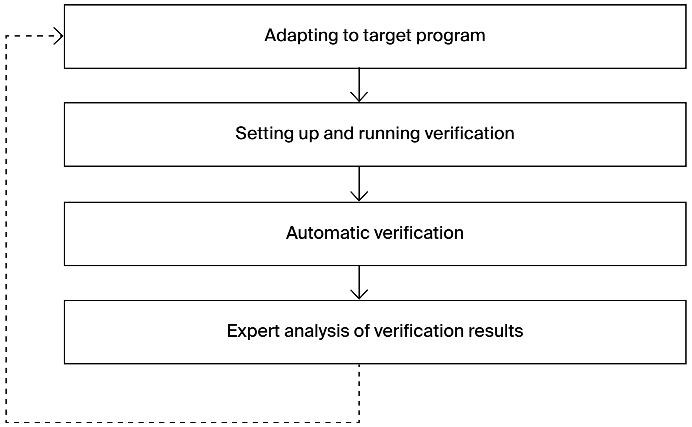
KLEVER DEPLOYMENT STORIES

The Klever verification system is mostly used for thorough checking of various operating system kernels and drivers. To showcase Klever features, it was used for verification of Linux kernel device drivers. As a result more than 400 errors of the following types have been found: buffer overruns, null pointer dereferences, uninitialized memory usages, double or incorrect memory deallocations, memory leaks, race conditions and deadlocks, incorrect function calls (depending on a certain context), incorrect initialization of Linux kernel data structures etc. Linux kernel developers have acknowledged these errors.

SYSTEM REQUIREMENTS

Ubuntu 18.04/20.04, at least 4 x86-64 CPU cores, 16 GB of memory, 100 GB of disk space.

WORKFLOW


Adapting to target program
Setting up and running verification
Automatic verification
Expert analysis of verification results

MASIW: SUPPORT FOR DESIGNING HIGHLY RELIABLE SOFTWARE SYSTEMS

MASIW is a toolset for developing highly reliable hardware and software systems for avionics, medicine, and other safety critical areas. It is designed for engineers creating airborne hardware/software systems that are developed using the integrated modular avionics (IMA) approach. MASIW can be easily adapted for other application areas.

FEATURES AND ADVANTAGES

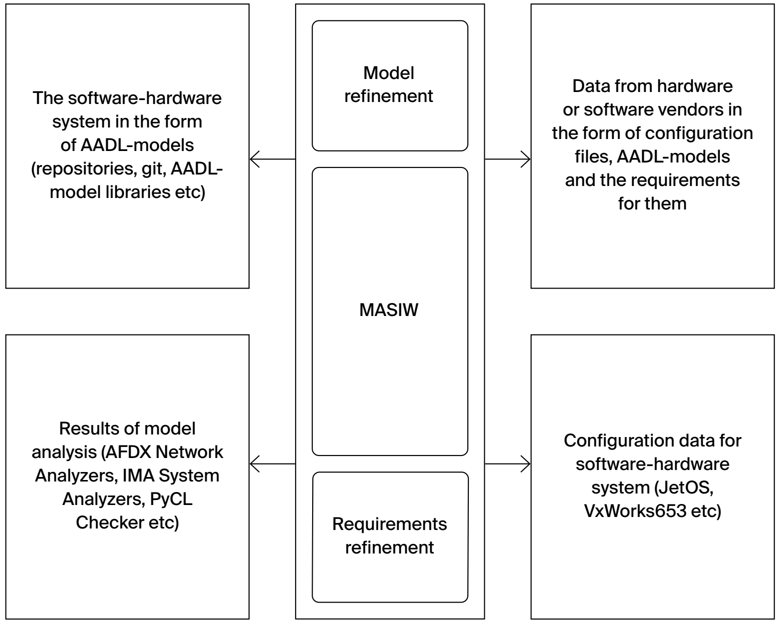
MASIW is the technology for optimizing the development and verification process of complex hardware/software systems. It allows performing a preliminary quality assessment of the product before making the first prototype, as well as performing the fault tolerance analysis. This reduces the risk of errors and defects. MASIW is being developed jointly with GosNIIAS. Despite the presence of the OSATE tool at the start of development, MASIW currently is more functional in the areas of verification, static, and dynamic analysis.

MASIW provides:

- Creation, editing and management of models based on the AADL modeling language:
  - creation and editing of models using the text and diagram editors;
  - support for team development with the ability to track and modify individual elements of a model;
  - support for the third-party AADL models reuse.
- Model analysis:
  - hardware+software system structure analysis: hardware resources sufficiency, interfaces consistency, etc.;
  - verification of the developed system for compliance with the requirements;
  - transmission characteristics analysis for the AFDX networks: message latencies, port queue depth, etc.;
  - generation and analysis of fault trees (FTA) to determine probabilities of high-level fault events;
  - architecture-model based analysis of failures and their consequences, including generation of special descriptive tables;
  - simulation of hardware+software system model with user reports generation including software-in-the-loop execution of on-board partitions with RTOS co-emulated with QEMU and with a universal AADL model simulator.

- Model synthesis:
  - distribution of software applications by computational modules taking into account hardware resource limitations and additional restrictions regarding reliability and security;
  - processor schedule generation (in particular, for ARINC-653 compatible real-time operating systems).
- Configuration data generation:
  - development of specialized configuration data tools based on the provided software interface (API);
  - configuration data generation for the VxWorks653 RTOS and for the AFDX network equipment.
- The ability to extend the toolset by creating own modules.

MASIW WORKFLOW


The software-hardware system in the form of AADL-models (repositories, git, AADL-model libraries etc)
Model refinement
MASIW
Requirements refinement
Data from hardware or software vendors in the form of configuration files, AADL-models and the requirements for them
Results of model analysis (AFDX Network Analyzers, IMA System Analyzers, PyCL Checker etc)
Configuration data for software-hardware system (JetOS, VxWorks653 etc)

MICROTESK: TEST PROGRAM GENERATOR

MicroTESK is a reconfigurable and extendable framework for generating test programs for functional verification of microprocessors. MicroTESK allows automatically constructing test program generators based on formal specifications of microprocessor architectures. MicroTESK supports a wide range of architectures including RISC, CISC, VLIW, and DSP. MicroTESK supports online test program generation.

FEATURES AND ADVANTAGES

MicroTESK is a set of technologies for industrial use that includes the basic modeling framework (building microprocessor models based on formal specifications) and the generation framework (building test programs based on test templates). MicroTESK delivers value similar to its global competitors (e.g., Genesys Pro and RAVEN) but outperforms them via increased usability and performance. Also, it is distributed under the open-source Apache 2.0 license. MicroTESK is available at the ISP RAS website: https://forge.ispras.ru/projects/microtesk. The technology is also presented at http://www.microtesk.org.

MicroTESK provides:

- Using formal specification as a source of knowledge about the microprocessor under verification:
  - architecture specification in the nML language (registers, memory, addressing modes, instruction logic, text/binary instruction representation);
  - additional memory subsystem specifications in the mmuSL language (memory buffer properties (TLB, L1, and L2), address translation logic, read/write operations logic);
  - an option to make a transition to formal verification and to automatic toolchain generation for the microprocessor under development (disassembler, emulator, etc.).
- Test programs generation based on object-oriented test templates:
  - test templates in the Ruby language (so that the templates are human readable and easy to support);
  - allows using different generation techniques for instruction sequences and test data simultaneously (random generation, combinatorial generation, constrained-based generation, etc.);
  - generation of framework scalability (can develop complex test templates at low cost due to reuse).
- Wide range of supported microprocessor architectures:
  - supporting architecture specific features for various architectures (RISC, CISC, VLIW, DSP) at the generator development framework level;

    - MicroTESK-based test program generators have been developed for RISC-V, ARM, MIPS, and PowerPC architectures;
    - multicore architectures are supported.
- Quick framework adaptation for the new microprocessor architecture with minimal costs and automatic information extraction for test situations (due to formal specifications).
- Convenient language for developing test templates that allows describing complex verification scenarios quickly.
- Support for online test program generation for performing post-silicon verification of the target microprocessor. The online generation is performed by an executable generator included into MicroTESK. The generator constructs test sequences using formal specifications, and then modifies the sequences by making functionally equivalent substitutions. It also allows repeated execution of the test sequences on the target microprocessor.

SYSTEM REQUIREMENTS

Windows or GNU/Linux-based OS, Java 8.

MICROTESK DEPLOYMENT STORIES

MicroTESK has been in development since 2007. It was used in various Russian and international projects on developing modern industrial microprocessors, including production projects on verifying ARMv8, MIPS64, and RISC-V microprocessors.

WORKFLOW

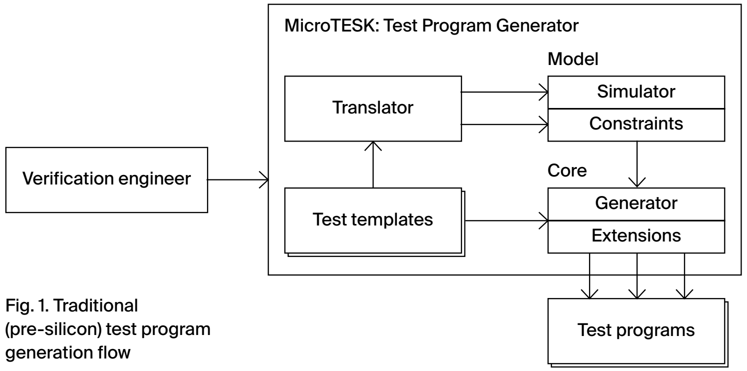


Fig. 1. Traditional (pre-silicon) test program generation flow

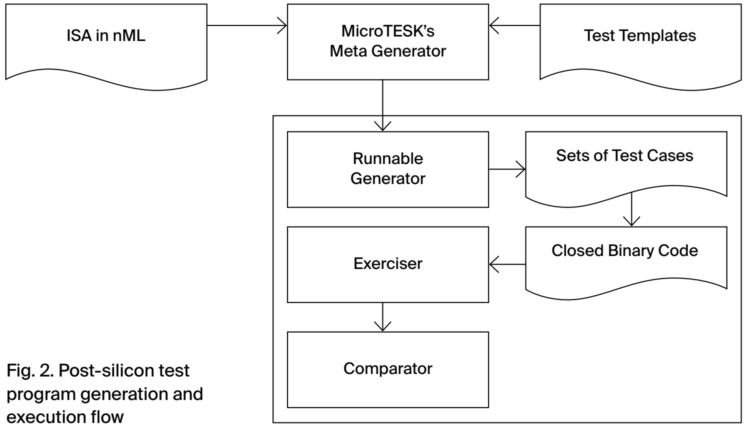


Fig. 2. Post-silicon test program generation and execution flow

SAFEC: SAFE COMPILER

The SAFEC safe compiler avoids introducing new vulnerabilities in program's binary code when aggressively optimizing (e.g., when making use of source code constructs exhibiting undefined behavior). The compiler tries to avoid excessively restricting optimizations, which allows to avoid the significant performance drop compared to all optimizations being switched off.

FEATURES AND ADVANTAGES

The safe compiler is developed on the basis of the industrial GCC compiler, and can act as a drop-in replacement for GCC (for example, when building the complete Linux distribution). The compiler retains the generated code quality and produces the ready-to-use safe build of a program.

The safe compiler provides:

- Refined compiler optimizations for conservative treatment of source code places with undefined behavior so that for these places program semantics gets defined safely and naturally.
- Forced initialization of uninitialized automatic variables.
- Issuing warnings when detecting constructions with undefined behavior.
- Adding dynamic checks for certain constructs to prevent exhibiting undefined behavior during program execution.
- Diversifying code generation during either compilation or program execution.
- No need to modify either source code or build system configuration, which makes using the compiler as simple as possible.
- Three different safety levels that provide trade-offs between generated code safety and performance. The lowest level is the third; the highest level is the first.

The safe compiler performs the following actions:

On the third level:

- Avoiding integer overflow, accessing objects via pointers of incompatible types, dereferencing null pointers, using compiler built-ins instead of standard library implementations for input/output functions and for functions working with memory.
- Detecting division by zero, incorrect bitwise shifts, accesses beyond stack frames, array loads/stores outside of the memory allocated for the array. Detecting automatic variables that are stored in registers during function calls.

On the second level:

- Analyzing arguments of bitwise shifts, redundant memory operations, data alignment when working with vector instruc-

tions, address arithmetic when optimizing memory accesses and changing their order.
- Initializing all automatic variables (with zero) that are not initialized explicitly by the user.
- Treating certain compiler warnings as errors and stopping compilation when they are issued.

On the first level:
- Generating unique memory layout for function code either statically during compilation or when performing dynamic linking.
- Adding machine code that aborts the program when detecting undefined behavior during program execution (sanitization) in the following situations:

1 Integer and floating point operations:
   - loading a non-Boolean value in a Boolean variable;
   - floating point conversion that results in either integer or floating point overflow;
   - performing a bitwise shift with a negative shift value or with a shift value that is equal or greater than the shifted type width;
   - signed integer operation with the result that is non representable in the output type;
   - integer division or module with the divisor equal to zero.

2 Pointer and array operations:
   - loads/stores via incorrectly aligned or null pointer;
   - array loads/stores using the address outside of the memory allocated for the array;
   - passing null pointer as a function parameter marked with the nonnull attribute;
   - address arithmetic resulting in integer overflow;
   - returning null value out of function that is marked with the returns_nonnull attribute;
   - allocating an automatic VLA array with incorrect size (zero or negative).

3 Function operations:
   - a function pointer call via a pointer whose type does not match the function prototype;
   - returning from a non-void function without actually executing the return statement;
   - calling a compiler built-in with incorrect arguments;
   - reaching a program point during program execution that is marked in the source code as unreachable.

WHO IS THE SAFEC TARGET AUDIENCE?

- Operating system developers.
- Companies developing high-level safe and secure software.

SAFEC DEPLOYMENT STORIES

The safe compiler is deployed in a number of Russian companies and government institutions as an add-on to the ISP Crusher framework.

SUPPORTED PLATFORMS

Linux-based OS (for x86 32/64 and ARMv7); Windows (MinGW).

SVACE: STATIC ANALYZER

Svace is an essential tool of the secure software development life cycle, the main static analyzer that is used in Samsung Corp. It detects more than 50 critical error types. Svace supports C, C++, C#, Java, Kotlin, and Go. Svace is included in the Unified Register of Russian Programs (No.4047). It is distributed with the Svacer web interface (Svace History Server).

FEATURES AND ADVANTAGES

Svace is an innovative technology based on years of research that constantly evolves for customer's needs. It combines the key qualities of foreign competitors (Synopsis Coverity Static Analysis, Perforce Klocwork Static Code Analysis, Fortify Static Code Analyzer) with the unique open industrial compilers usage to provide the maximal support level for new programming language standards.

Svace provides:

- High-quality deep analysis:
   - accurate representation of the source code (due to integration with any build system);
   - full path coverage taking into account function calling contexts when searching for complex defects;
   - high percentage of true positives (60-90%).
- Scalability and high speed:
   - parallel analysis using all available processor cores;
   - ability to analyze software with the code size of tens of millions of lines (analysis of the Tizen 7 mobile OS having 57 million lines of code takes 7-8 hours using the main Svace engine and 9–10 hours using all engines);
   - supporting incremental system analysis in addition to the full analysis mode (performs a quick re-analysis of recently changed source files).
- Accelerated customization (configuring existing detectors as well as writing individual ones available exclusively to this customer; creating tailored user interfaces).
- Accelerated adaptation to new environments and tools (adding new compilers within 1-2 weeks, in complex cases up to 2 months).
- Full compatibility with regulatory documents and requirements of regulators (FSTEC of Russia).
- Can be used for adhering to the GOST R 56939-2016 requirements and to the requirements of the FSTEC regulation document mandating software vulnerability detection process (when certifying software within Russia).

WHAT IS SVACE TARGET AUDIENCE?

- Companies aimed at software development with a special focus on high reliability and security.
- Companies that need to certify the developed software.
- Certification laboratories.

SVACE DEPLOYMENT STORIES

Svace is the main static analyzer used in Samsung Corp. since 2015. It is used to check the company's own software based on Android OS as well as the Tizen OS source code. Tizen is used in smartphones, infotainment systems and Samsung home appliances. Since 2017, Svace checks all changes submitted for review and inclusion in the Tizen OS. Since 2020, Svace has been also used by Huawei.

Within Russia, Svace is deployed in more than 100 companies and certification labs, including RusBITech, Kaspersky Lab, Postgres Professional, Security Code, Swemel, and others.

SUPPORTED PLATFORMS AND ARCHITECTURES

- Host platforms for the Svace analyzer: Linux/x64 (version 3.10 and later, glibc version 2.17 and later), Linux/ARM 64 (Ubuntu 18.04), Windows starting with 7 SP1 with update KB2533623) and WSL (versions 1 and 2); macOS on x86-64 (starting from 10.10; C# is not supported); x86 architecture for build capture.
- Target architectures of the analyzed code: for C/C++ that is Intel x86/x86-64, ARM/ARM64, MIPS/MIPS64, Power PC/ Power PC 64, RISC-V 32/64, SPARC/SPARC64, Hexagon (code generation via Clang); Elbrus, AEON, TriCore, HIDSP, OpenRISC (code generation via one of the previous architectures); for Go, Linux-based Intel x86-64; for C#, Java, Kotlin — platforms where the analyzer is supported).
- Platforms and architectures for Svacer: x86-64; OS Linux (version 3.10 and later, glibc version 2.17 and later); OS Windows (starting with Windows 10) and WSL (versions 1 and 2); macOS on x86-64 (starting from 10.12 Sierra).

SVACER

Svacer is an integrated tool that provides the user interface for working with warnings, as well as a server for storing and managing analysis results. Svacer supports the multiuser mode and various data filters.

Svacer is:

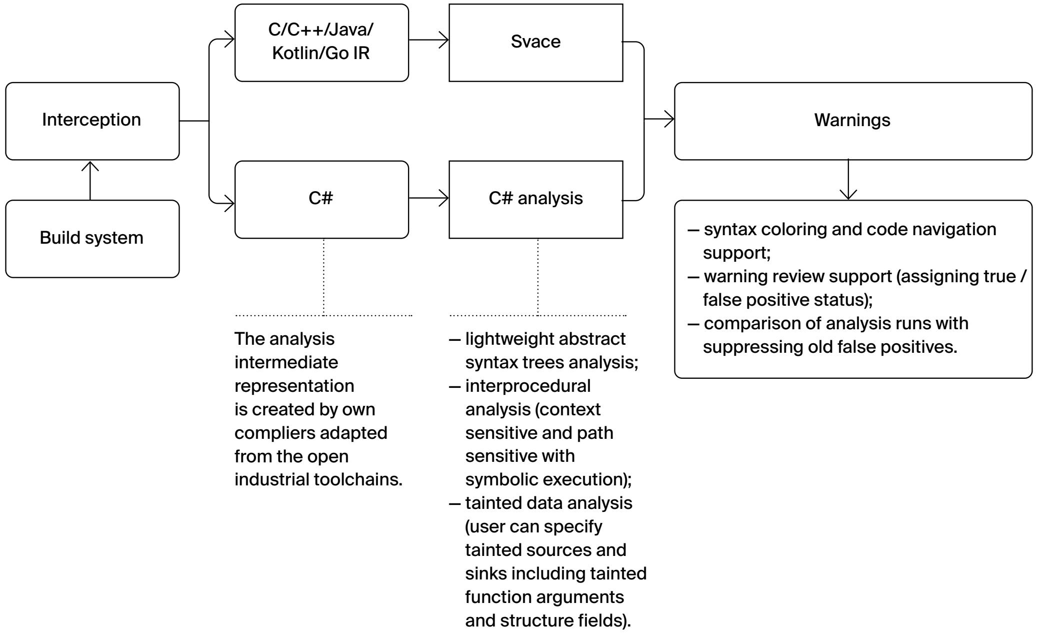
SUPPORTED PLATFORMS AND ARCHITECTURES

- Extensive possibilities for comparison and markup of results, access to results via API, creating reports in PDF, CSV, JSON formats.
- LDAP support for user authentication.
- Integration with Visual Studio Code and providing command line interface for integration into standard CI/CD processes.
- Annotation of work results through user files and attributes.
- Support for the SARIF format which allows to import results of other static analyzers.
- Importing and exporting marked-up results to source code.
- Support for working in containers.

SUPPORTED COMPILERS

- For C/C++ (up to C++17): GCC (GNU Compiler Collection), Clang (LLVM compiler), Microsoft Visual C++ Compiler, RealView/ARM Compilation Tools (ARMCC), Intel C++ Compiler, Elbrus C/C++ Compiler, Wind River Diab Compiler, Keil CA51 Compiler Kit, NEC/Renesas CA850, CC78K0(R) C Compilers, C/C++ Compiler for the Renesas M16C Series and R8C Family, Panasonic MN10300 Series C Compiler, C compiler for Toshiba TLCS-870 and T900 Family, Samsung CalmSHINE16 Compilation Tools, Texas Instruments TMS320C6* Optimizing Compiler, Digital Mars C and C++ Compiler, Green Hills compiler for ARM, TASKING C compiler for TriCore, CEVA Toolbox for CEVA DSP cores, IAR C/C++ Compiler for ARM / Renesas RL78 MCU, CodeWarrior Development Studio for StarCore DSPs, Open Watcom C/ C++ compiler, Freescale CodeWarrior, Cadence Tensilica Xtensa C/C++ Compiler.
- For C# (up to C#10): Roslyn, Mono.
- For Java (up to Java 17): OpenJDK Javac Compiler, Eclipse Java compiler.
- For Kotlin: Kotlin 1.6.
- For Go: Go 1.17.

SVACE ARCHITECTURE



- syntax coloring and code navigation support;
- warning review support (assigning true / false positive status);
- comparison of analysis runs with suppressing old false positives.

The analysis intermediate representation is created by own compliers adapted from the open industrial toolchains.

- lightweight abstract syntax trees analysis;
- interprocedural analysis (context sensitive and path sensitive with symbolic execution);
- tainted data analysis (user can specify tainted sources and sinks including tainted function arguments and structure fields).

TESTOS: SOFTWARE TESTING ENVIRONMENT

TestOS is an environment for unit testing of software on target hardware. It allows to debug software for critical applications on ARM, PowerPC, MIPS and x86 architectures to perform certification and other activities.

FEATURES AND ADVANTAGES

TestOS makes it possible to replace such critical systems verification tools as LDRA, since it is a more flexible tool with active support for domestic products.

Using TestOS ensures running tests on target hardware and generating reports with the trace for each test, with information about the composition and passing status of the test plan and with the coverage of the tested system code both for one test, and for the whole test plan. Reports are generated in HTML and TXT formats. Debugging of the code on the target computer is available both with and without use of JTAG.

With plugin application, the following is supported:

- collecting function, operator, and branch coverage using GCOV and LLVM Coverage;
- collecting coverage by MC/DC using COVERest;
- performing static analysis with static analyzers:
  - Clang Tidy;
  - Clang Static Analyzer;
  - Svace.
- Dynamic code instrumentation with LLVM sanitizers:
  - AddressSanitizer (detecting memory handling errors);
  - MemorySanitizer (detecting errors of accessing uninitialized memory);
  - UndefinedBehaviorSanitizer (to detect arithmetic, floating-point, and other undefined behavior errors).

SYSTEM REQUIREMENTS

GNU/Linux distribution on x86_64 architecture (such as Ubuntu 22.04), and Apple macOS 10.12 or newer as the target machine.

Target machine with at least 2MB RAM on architectures:

- ARM (Cortex-A7, Cortex-A9, Cortex-M4), including i.MX6 or STM32F429 processors.
- PowerPC (e500mc, e500v2, 476FP), including the p1010 or p3041 processors.
- MIPS (MIPS Release 1, MIPS Release 2 / MIPS32, COMDIV), including the 1892VM15AF processor.
- x86 (Intel Prescott and newer).

If necessary, the environment is adapted to the customer’s equipment.

TESTOS DEPLOYMENT STORIES

TestOS has been in development since 2019. It is successfully applied for modular software testing for the aerospace industry.

DIVERSIFICATION TOOL: ANTI-EXPLOIT PROTECTION SET

Diversification tool is a set of technologies to prevent mass exploitation of vulnerabilities resulting from bugs or backdoors. If an attacker was able to attack one of the devices with the same software installed, the others will remain protected thanks to the changes made to the code.

FEATURES AND BENEFITS

The diversification tool protects the system from mass exploitation of vulnerabilities by means of various code diversification methods, and makes it possible to build the code of the whole OS distribution.

The diversification tool provides:

- Fine-tuning the balance between the degree of obfuscation and the level of performance (when applied to protect against reverse engineering). Minimum 1.2x slower performance, maximum 8x slower performance.
- Full automation (no special preparation of program source code and no additional efforts on the part of customer's build engineers are required).
- Based on the GCC compiler, which allows to build the full OS distribution code correctly.
- Use of the original control flow integrity method (CFI), which successfully resists most code reuse attacks (ROP, JOP, ret-to-plt, etc.). Implementation of the CFI method based on the GCC compiler resulted in average slowdown on the SPEC CPU2006 test suite of about 2%, which is noticeably lower than that of traditional methods.
- Two diversification methods:
- Dynamic code diversification at program startup. It is used when the customer needs the same code on all devices (for example, due to mandatory certification). This method makes it possible to move up to 98% of the code with a small increase in its size and about 1.5% performance degradation.

  The advantages of the Diversification Tool over similar products include:

  - shuffling up to a function (as opposed to ASLR and Pagerando technologies, which only move large blocks of code);
  - shuffling of functions in the whole system, except for the kernel, and no conflict with anti-viruses (which is an advantage over the similar technology Selfrando developed for the Tor Browser);

- Static code diversification. Each time the code is compiled, depending on the specified key, a new executable file is produced. The advantages of this method include:
  - no increase in binary code size (particularly important for the Internet of Things);
  - performance degradation tends to vanish;
  - due to working inside the compiler rather than ex post facto in the linker, an extended set of diversifying transformations can be applied and tuned with more flexibility;
  - control flow integrity (CFI) method.
- Conflict-free compatibility with other software protection tools (including the ASLR system mechanism).

WHO IS THE DIVERSIFICATION TOOL INTENDED FOR?

- Developers of specialized operating system installation software.
- Application software developers.

DIVERSIFICATION TOOL DEPLOYMENT STORIES

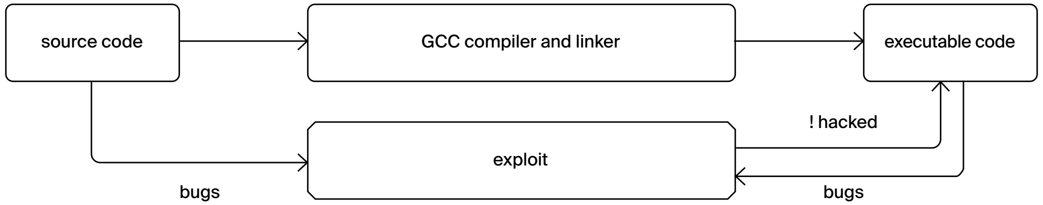
The tool is deployed in OS "Zircon," which is used by Ministry of Foreign Affairs and the Border Guard Service of the Federal Security Service of Russia. Currently, the Diversification Tool is implemented as part of the SAFEC Level 1 secure compiler and is supplied together with it.

SYSTEM REQUIREMENTS

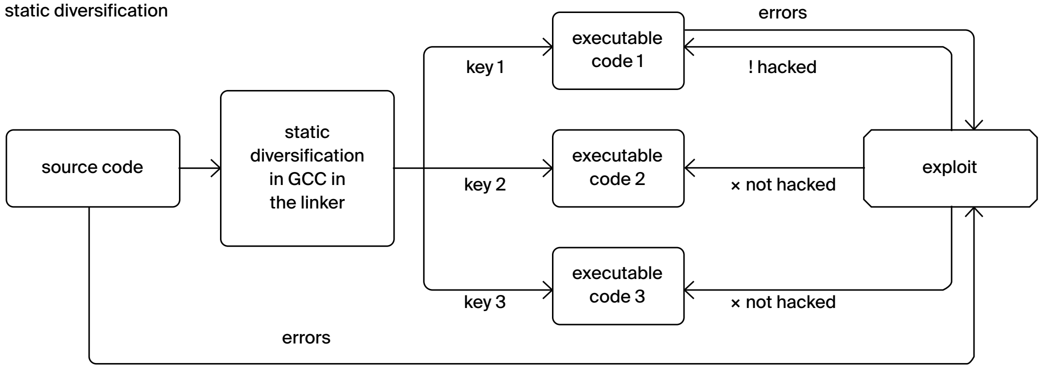
The Diversification Tool is a universal product which can be adapted to any system requirements. Currently, the main version works on Linux kernel-based operating systems (starting with version 2.6) with Intel x86/x86-64 architecture support.

WORKFLOW

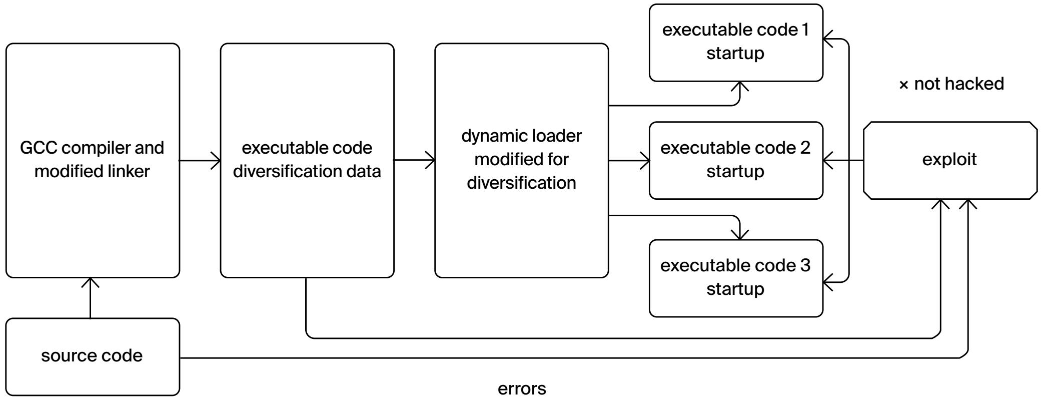
usual build



static diversification



dynamic diversification


GCC compiler and modified linker
executable code diversification data
dynamic loader modified for diversification
executable code 1 startup
executable code 2 startup
executable code 3 startup
× not hacked
exploit
source code
errors

QEMU-BASED SOFTWARE ANALYSIS PLATFORM

ISP RAS Foundation Platform for creating program analysis systems is built on top of open source QEMU emulator. This framework is essential for organizing cross platform development. It supports reverse debugging and introspection features, as well as full system emulation mode for debugging low-level software.

FEATURES AND ADVANTAGES

QEMU supports emulation of more than 10 instruction set architectures (i386 and x86-64, ARM and Thumb, MIPS, PowerPC, etc.). It implements guest debugging via GDB Remote Serial Protocol and is compatible with IDA Pro, GDB, and various IDEs. QEMU supports full system emulation mode that allows debugging low-level software such as a bootloader and an OS kernel. The QEMU source code is regularly checked by static code analysis tools, including Coverity and Svace. Thus performing malware analysis with QEMU is more secure. QEMU with reverse debugging and introspection support is available on the ISPRAS GitHub page: https://github.com/ispras/swat. The developed QEMU automatization tools are available at https://github.com/ispras/qdt, https://github.com/ispras/i3s.

ISP RAS QEMU Foundation Platform provides:

- A record and replay mechanism for a virtual machine:
  - The same VM execution is replayed every time, deterministically. All external events are recorded and replayed by the emulator. It makes finding bugs in multi-threaded applications (race conditions, deadlocks) easier;
  - GDB-compatible reverse debugging is implemented based on the record and replay mechanism. The debugging is performed by restoring previous VM snapshots and searching for the previous breakpoint stop or the previous instruction;
  - The minimum required information is recorded. This allows recording longer for debugging rarely occurring errors;
  - Low performance overhead caused by recording. This enables analysis of software that requires interacting with an uncontrolled external environment in real time.

- VM introspection solution (getting high-level information regarding guest OS work) without any guest OS kernel modifications or installing monitors:
  - Getting the list of executed system calls, accesses to named functions in shared libraries, the list of running processes, the list of open files and loaded modules;
  - Supports all Linux-based virtual machine images as well as embedded software images for various devices;
  - WinDbg server support in QEMU that allows showing guest software information in terms of Windows kernel abstractions. There is no need to enable the OS debugging mode in the guest OS.
- Speeding up QEMU development:
  - Faster development of dynamic analysis tools that can analyze binary code for specific hardware;
  - Automated support for new processor architectures using a machine instruction decoder generator and a C-like language for describing machine instructions semantics;
  - An automatic tool for preliminary virtual machine testing. The tool only requires GNU Binutils and a C compiler;
  - A tool for automating QEMU virtual devices development;
  - VM generation tool in the form of QEMU module source code. The tool can create VMs from both existing devices and new devices out of Python description. The tool provides GUI for sketching the virtual machine;
  - A Python API for an automated debugging via GDB Remote Serial Protocol. It is used to debug QEMU, the guest OS, or both at the same time.
- Convenience and user experience:
  - Easy QEMU extension due to open source code and own ISPRAS toolkit for speeding up development;
  - Binary code analysis without any guest OS modifications;
  - VM introspection mechanism that can be extended using plugins;
  - A convenient API for developing own introspection plugins;
  - Can be easily adapted for specific use cases;
  - Support for latest QEMU versions that have support for newest peripherals and CPUs.

WHO IS ISP RAS FOUNDATION PLATFORM TARGET AUDIENCE?

- Bootloader, driver, OS and other system software developers.
- DevOps teams for software bugs reproduction, cross-platform development, and scalable cloud testing.
- Programmers analyzing potential malware.
- Software certification engineers.

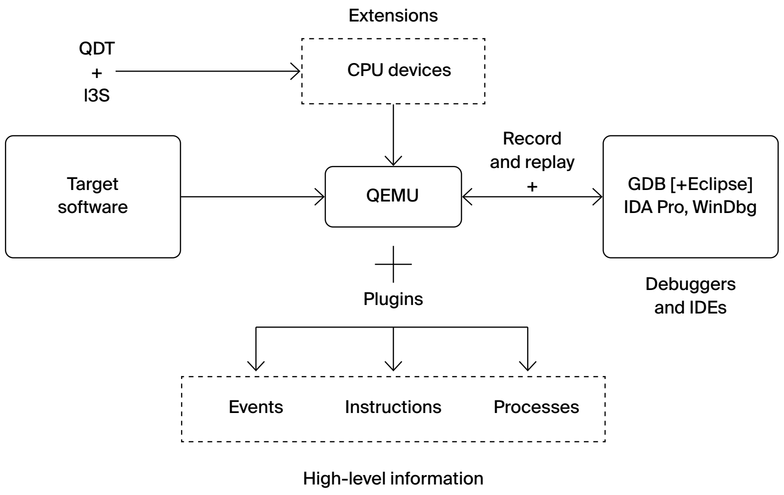
SUPPORTED GUEST PLATFORMS

Emulation of the following ISAs: i386, x86-64, ARM, MIPS, PowerPC, and others.
Guest systems supported by the introspection mechanism: Windows XP (x86), Windows 10 (x86-64), Linux 2.x-5.x (x86, x86-64, ARM, AArch64).

ISP RAS QEMU DEPLOYMENT STORIES

The QEMU community has accepted ISP RAS patches for the record and replay mechanism and added them to the open source QEMU version 3.1.

WORKFLOW



ISP CRUSHER: BINARY CODE DYNAMIC AND STATIC ANALYSIS TOOLSET

ISP Crusher is a toolset that combines various dynamic and static analysis approaches, including fuzzing (using ISP Fuzzer, a fuzzing tool), and symbolic execution (among others, the Sydr tool can act as a symbolic engine). In the near future Crusher will also include the BinSide analyzer, another ISP RAS technology. Crusher allows organizing a development process that is fully compliant with GOST R 56939-2016 and “Methodology for identifying vulnerabilities and undeclared features in software” of FSTEC of Russia.

FEATURES AND ADVANTAGES

The ISP Crusher core is ISP Fuzzer, a fuzzing tool essential for any fuzzing tests on every stage of software development phase, be it coding, testing, or deployment. The fuzzer finds program errors either with or without source code. It solves the same problems as its global competitors (Synopsys Codenomicon, beSTORM, Peach Fuzzer), but it is more convenient for Russian companies in the import phase-out context.

ISP Fuzzer provides:

- Fuzzing a wide class of software:
  - custom applications, kernel and libraries;
  - applications in various programming languages: C/C++, Java, Python, C#;
  - fuzzing of neural networks. The software reveals cases of erroneous neural network predictions when correctly classified input data is distorted. This makes it possible to find errors in the networks for situations that have not been originally included in the training dataset;
  - fuzz-testing through different input data sources: file, command-line arguments, standard input stream, environment variable arguments, network, direct writing to memory;
  - ability to analyze server and client software running on stateful and stateless protocols;
  - extensive possibilities for fuzzing software of embedded devices through partial emulation and symbolic execution;
  - browser fuzzing: browser control via Selenium, coverage feedback via Frida.
- Large capacity fuzzing:
  - Support for multi-threaded analysis on both a single machine and distributed ones;
  - ability to distribute input data corpus between fuzzer processes to increase efficiency of their work;
  - support for differential fuzzing.
- Support for a large set of tool types:
  - static (mostly for C/C++) with GCC/LLVM;
  - static instrumentation of Python bytecode;
  - dynamic (mostly for ELF, PE): DynamoRIO, Qemu (user-mode), TinyInst;
  - based on partial emulation;
  - using Nyx snapshots and snapshot-API;
  - Java applications;
  - C# applications;
  - remote instrumentation (which makes it possible to perform fuzzing of an application running on a remote device).
- Ability to integrate with a number of necessary tools of secure software development lifecycle tools developed at ISP RAS:
  - the use of dynamic symbolic execution tool Sydr to improve the efficiency of fuzz-testing;
  - ability to receive input data to check errors marked by BinSide static analysis BinSide in automated mode;
  - displaying the trace of the sequence of functions causing crashes in the interface of the Svace static analyzer;
  - using the data generator that is based on ANTLR grammars to generate the input data corpus.
- Integration with other dynamic analysis tools:
  - with third-party fuzzers, allowing to run a set of different synchronized fuzzers within one fuzzing session, which increases the efficiency of testing;
  - with SymCC and Angr dynamic symbolic execution tools, which makes it possible to get new input data to increase the code coverage of target software;
  - working together with the IDA PRO disassembler (saving the coverage for the Lighthouse plugin, which displays the covered basic blocks in the software, as well as displaying the percentage of covered basic blocks);
  - using the Radamsa fuzzer to generate new data.

Additional analysis of the received input data:

- In-depth evaluation of the received input data:
  - Evaluation of the criticality of found abnormal terminations;
  - ability to launch dynamic analysis systems using new input data: Valgrind, DrMemory, QASan;
  - creation of the coverage profile by source code.
- Extensive options for integration of custom extensions:
  - option to add user-side handlers that will automatically run on new input data;
  - option to add custom mutation transformations (to generate new input data and increase testing efficiency);
  - availability of input data pre-processing and post-processing modules to perform constant transformations of data before sending it to the software to be analyzed;

    - support of custom plugins for sending data over network (plugins allow interacting with client or server software and sending mutated data);
    - support of custom Python scripts to modify options (avoids conflicts when multiple fuzzing processes are running simultaneously);
    - support for custom Python plugins to control the environment for launching the target software (which makes it possible to keep an identical environment at each start-up);
    - support for custom instrumentation plugins (which makes it possible to define arbitrary classification rules for input data based on the target software behavior: definition of normal and crash termination, freezing);
    - ability to describe scenarios for fuzzing software with the user interface.
- Easy extensibility and easy adding new methods within the framework of the existing infrastructure; fast adapting to new tasks.

WHO IS ISP CRUSHER TARGET AUDIENCE?

- Companies developing highly reliable and secure software.
- Companies auditing or certifying software.

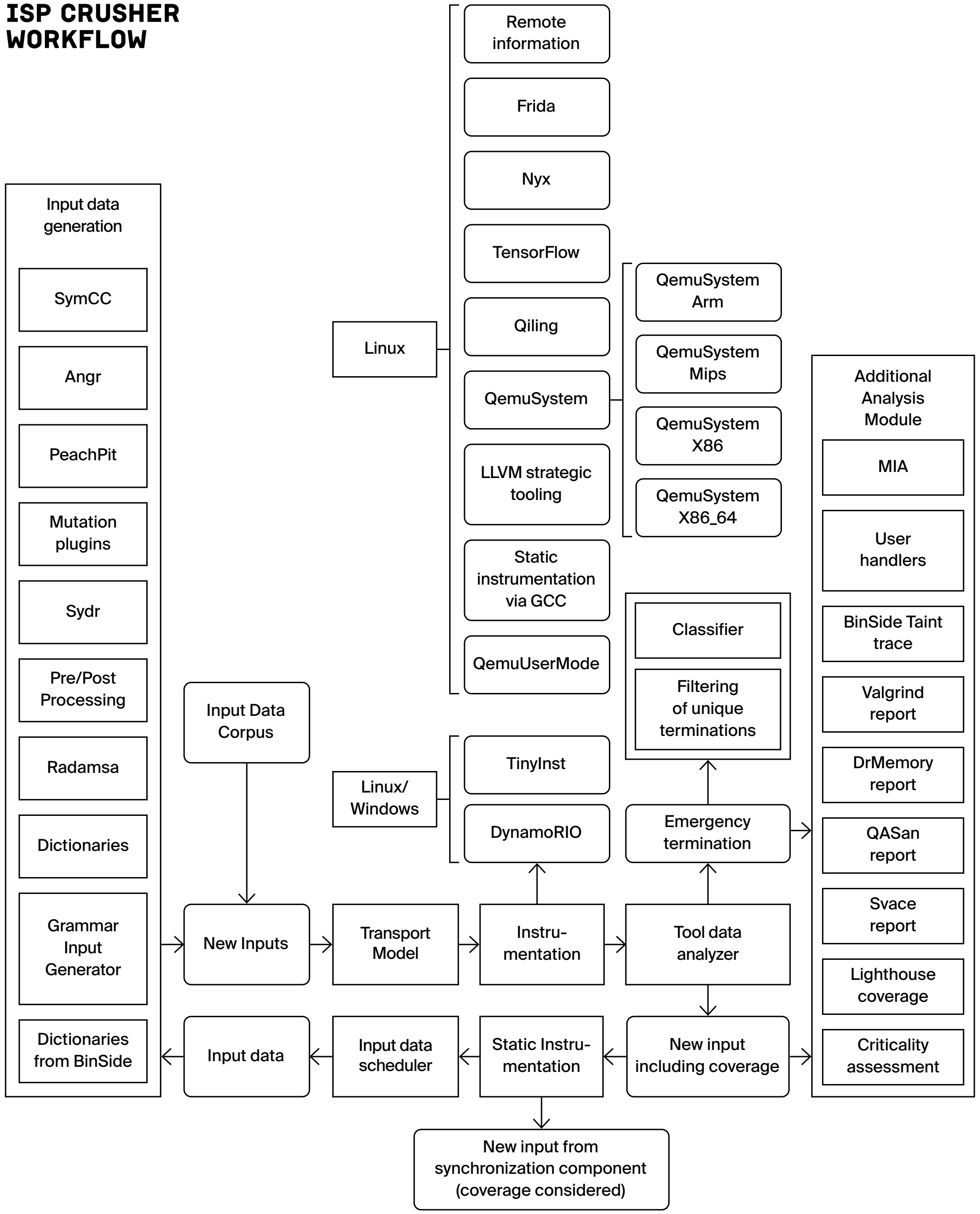
SYSTEM REQUIREMENTS

Fuzzing supported for Linux and Windows OS family. Fuzzing of software available for x86_64, ARM, MIPS architectures. Crusher can also fuzz embedded devices (controllers, IoT devices) as well as Windows services and COM objects.

ISP CRUSHER DEPLOYMENT STORIES

ISP Crusher is used in more than 70 companies and certification labs, including RusBITech, Postgres Professional, Security Code, Swemel and others.

ISP CRUSHER WORKFLOW



CATALOGUE OF TECHOLOGIES

BINSIDE: A BINARY CODE STATIC ANALYSIS TOOL

BinSide is a static program analysis platform for finding defects in binary code. It is useful when checking programs without source code, such as closed source third-party libraries.

FEATURES AND ADVANTAGES

BinSide is a binary code analysis platform based on the BinNavi framework. An executable file is analyzed in IDA PRO or Ghidra representation. BinSide provides various analysis types such as defect detection, code clone detection, dynamic analysis optimization, analysis automation, dynamic testing optimization.

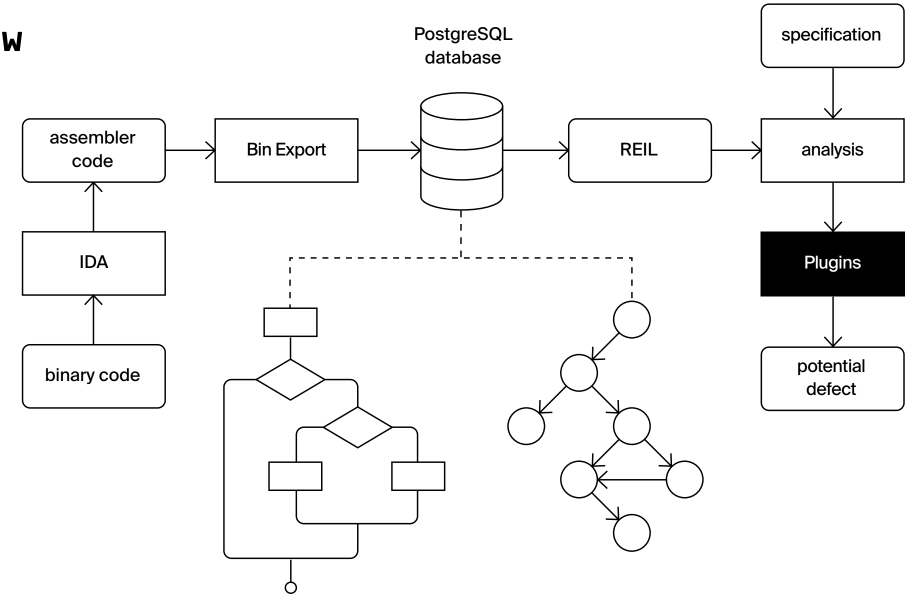
BinSide core provides:

- Easy extension:
  - individual error detectors are written as plugins;
  - the REIL representation of 17 instructions without side effects is used (each assembly instruction is translated into a set of REIL instructions);
  - it is possible to specify the functions' semantics to improve analysis quality.
- Supports analyzing executables and libraries for x86-64, ARM, and MIPS architectures, including drivers.
- Detecting the following CWE types:
  - CWE-121 (Stack-based Buffer Overflow);
  - CWE-122 (Heap-based Buffer Overflow);
  - CWE-134 (Format String Vulnerability);
  - CWE-415 (Double Free);
  - CWE-416 (Use-After-Free);
  - CWE-77 (Command Injection).
- Executing the following tasks:
  - data flow and control flow analysis: retrieval of values and pointers, labeled data propagation, determining possible heap states, determining computable edges of the control flow graph;
  - intraprocedural search for defects: search for defects is performed on the basis of the results of intraprocedural analysis of data and control flow, the results of dynamic analysis and manual code markup by the analyst. This is especially useful when analyzing complex software and embedded systems;
  - analysis of all paths, regardless of code coverage.
- Interaction with ISP RAS technologies:
  - with the Svacer tool (if the source code is available);
  - LibraryIdentifier tool (to search for code clones, e.g. to identify libraries whose code has been used for the executable file);
  - Crusher fuzzing-test tool.
- Operating system analysis:
  - Determining code plagiarism from an open-source OS;
  - Determining dependencies between OS components and within components;
  - Static analysis of the OS source and binary code;
  - Determining the protection of the executable code in the OS components;
  - Determining the coverage of the code in OS components by unit-tests.

WHO IS BINSIDE TARGET AUDIENCE?

- Companies that need to check thoroughly the used third-party software, including situations when there is no access to its source code.
- Developers who need to increase dynamic analysis quality with the data collected by static analysis.
- Reverse engineering experts.
- Companies performing software audition or certification.

BINSIDE WORKFLOW



CASR: CRASH ANALYSIS AND SEVERITY REPORTING TOOL

GitHub →
https://github.com/ispras/casr

Casr creates automatic reports for crashes happened during program testing or deployment on Linux. The resulting reports contain the crash's severity and additional data that is helpful for pinpointing the error cause.

FEATURES AND ADVANTAGES

Casr could collect crash reports using different ways. Casr can solve the same problem as Apport, an open source system, but in contrast with it, Casr estimates the severity of the crash and provides a list of open files and network connections at a crash time. The tool also makes it possible to create crash reports after the program launch, deduplicate, and cluster them.

Casr provides:

- Detecting critical program faults that can lead to hijacking control flow.
- Classifying crashes based on a program state at a crash time (function return address corruption, null pointer dereference etc.). Fatal errors are further grouped based on severity, such as exploitable, potentially exploitable, or denial of service errors.
- An extended crash report containing the fatal error's severity and other data (OS and package versions, executed command line, call stack, open files and network connections, register state etc.).
- Clustering of crashes based on the call stack. Clusters potentially contain similar reports that describe the same bug.
- Reports for instrumented binaries (information is partially taken from the sanitizer crash report).
- Reports for hard to reproduce crashes such as non-deterministic errors, cases when the original execution environment is hard or impossible to reconstruct etc.).

WHO IS CASR TARGET AUDIENCE?

- Companies that need to receive the data regarding user-deployed programs' crashes to develop high reliability and security software.
- Companies that need to certify the developed software.
- Certification laboratories.

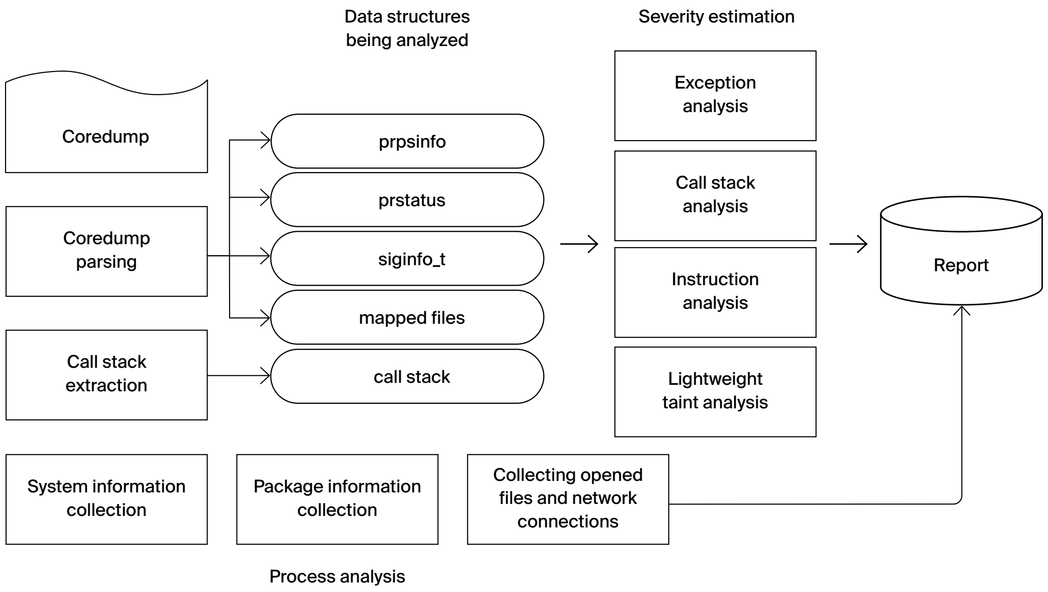
CASR DEPLOYMENT STORIES

Casr is deployed in a number of Russian companies and vendors as an add-on tool to ISP Crusher.

SYSTEM REQUIREMENTS

Linux-based OS for x86, x86-64. Casr is open source software (https://github.com/ispras/casr).

WORKFLOW



SYDR + SYDR-FUZZ: HYBRID FUZZING AND DYNAMIC ANALYSIS

GitHub →
https://github.com/ispras/oss-sydr-fuzz

Sydr is an automatic test generation tool for complex programs that finds errors and increases code coverage during testing. Sydr constructs the program's mathematical model that allows a fuzzer to explore new execution paths that are hard to discover via classic mutation approaches. Sydr improves dynamic symbolic execution methods proposed in earlier Avalanche and Anxiety analyzers developed in ISP RAS.

Sydr-fuzz is a dynamic analysis tool for security development lifecycle which combines the power of dynamic symbolic execution tool Sydr and modern fuzzers (libFuzzer and AFL++).

FEATURES AND ADVANTAGES

In contrast with similar open source tools, Sydr ensures the correctness of generated input data by checking whether it actually inverts the target branch. Sydr-fuzz provides a convenient fuzzing pipeline:

- Hybrid fuzzing with Sydr and libFuzzer/AFL++: sydr-fuzz run
- Corpus minimization: sydr-fuzz cmin
- Error detection (out of bounds, integer overflow, division by zero, etc.) via symbolic security predicates: sydr-fuzz security
- Collecting coverage: sydr-fuzz cov-report
- Crash deduplication, clustering, and severity estimation with Casr: sydr-fuzz casr.

Sydr provides:

- Hybrid fuzzing libFuzzer & Sydr, AFL++ & Sydr.
- Efficiency: continuous benchmarking shows that Sydr-fuzz is on the same level with world-famous competitors (https://sydr-fuzz.github.io/fuzzbench).
- Repository with ready to fuzz projects: 40+ projects (270+ fuzz targets) in OSS-Sydr-Fuzz (https://github.com/ispras/oss-sydr-fuzz).
- Trophies: Sydr-fuzz found 80+ new bugs in 18+ open source projects (https://github.com/ispras/oss-sydr-fuzz/blob/master/TROPHIES.md).
- Inverting all conditional branches that depend on input data.
- Security predicates that find errors (division by zero, null pointer dereference, buffer overflow, integer overflow, etc.) and generate input data to reproduce detected errors.
- Symbolic execution of multithreaded programs.
- Inverting indirect branches (in switch statements). Sydr implements a method for detecting jump tables and jumps to computed addresses.
- Path predicate slicing. Sydr removes redundant constraints (not influencing the target conditional branch) from the path predicate. This feature solves the problem of under-tainting and speeds up solving.
- Reasoning of symbolic pointers that depend on input data. This allows to find critical errors arising out of taking user input for array index calculation. Supporting symbolic pointers requires additional modeling that is usually absent from similar tools.

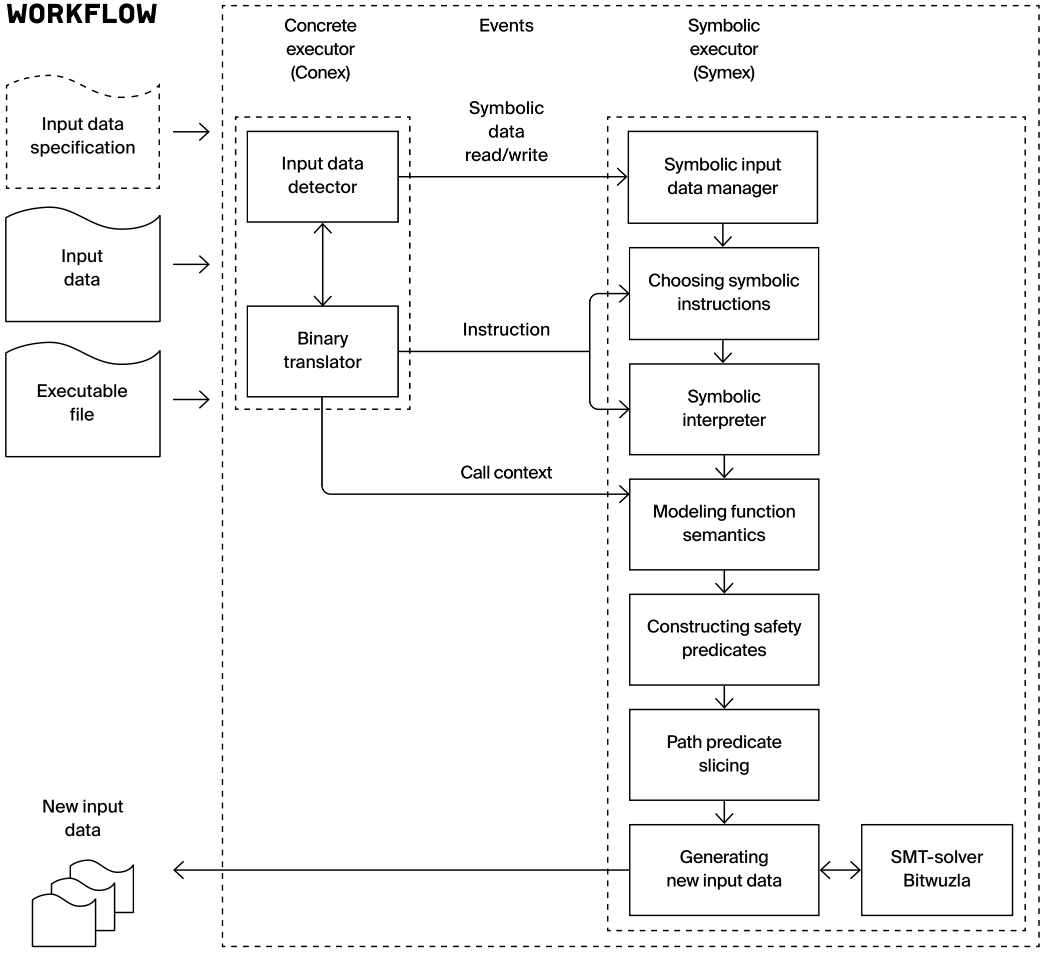
SYSTEM REQUIREMENTS

Linux 64-bit support, including Ubuntu 18.04/20.04/22.04, Astra Linux 1.7, ALT Workstation 10 and similar.

SYDR DEPLOYMENT STORIES

Sydr and Sydr-fuzz are the parts of ISP Crusher system that is used in more than 70 companies and certification labs, including RusBITech, Postgres Professional, Security Code, Swemel, and others. Sydr + Sydr-fuzz is the main dynamic analysis tool in Trusted Artificial Intelligence Center at ISP RAS.

WORKFLOW



PROTOSPHERE: NETWORK TRAFFIC ANALYZER

Protosphere is a system of deep packet inspection (DPI). It can serve as a part of intrusion and information leak protection systems. Protosphere detects inconsistencies between a protocol specification and the actual traffic. It allows you to add support quickly for new protocols (either open or closed) due to the flexibility of its internal representation.

FEATURES AND ADVANTAGES

Protosphere is an innovative system based on the innovative research in the area of network traffic analysis. It combines the key features of similar tools (e.g. Wireshark, Microsoft Message Analyzer) with a universal data representation model that enables rapid expansion of analysis capabilities.

Protosphere provides:

- Advanced system core:
  - universal data representation model used when parsing network traffic;
  - processing of corrupted, reordered or duplicated packets, as well as handling of packet loss and processing of asymmetric traffic;
  - compressed/encrypted data analysis;
  - support for tunnels of arbitrary configuration;
  - support for network flows causality.
- Support for all stages of network trace analysis (each stage has a visualization component that are synchronized between stages):
  - network connections localization in the network interaction graph and the network flow tree;
  - detailed view of the selected connections in the timeline diagram;
  - interactive visualization of the parsed network packets in the stream tree;
  - detection of discrepancies between a protocol implementation and the actual traffic in the diagnostic log;
  - arbitrary OSI-layer data extraction and analysis (L7+).
- Easy support for new protocols:
  - access to parsing results via API;
  - localize parsing errors;
  - debugging the module being developed on real-time traffic and network traces.
- Support for both online and offline analysis modes.
- Advanced GUI provides choice of the most convenient way to present the analysis results.
- Universal data representation model to accelerate customization:
  - support for new protocols;
  - extract data in a desired format;
  - configuring the analysis results format.
- Adjustment to network bandwidth and available computational resources to find a balance between accuracy of the analysis and the resources consumed.

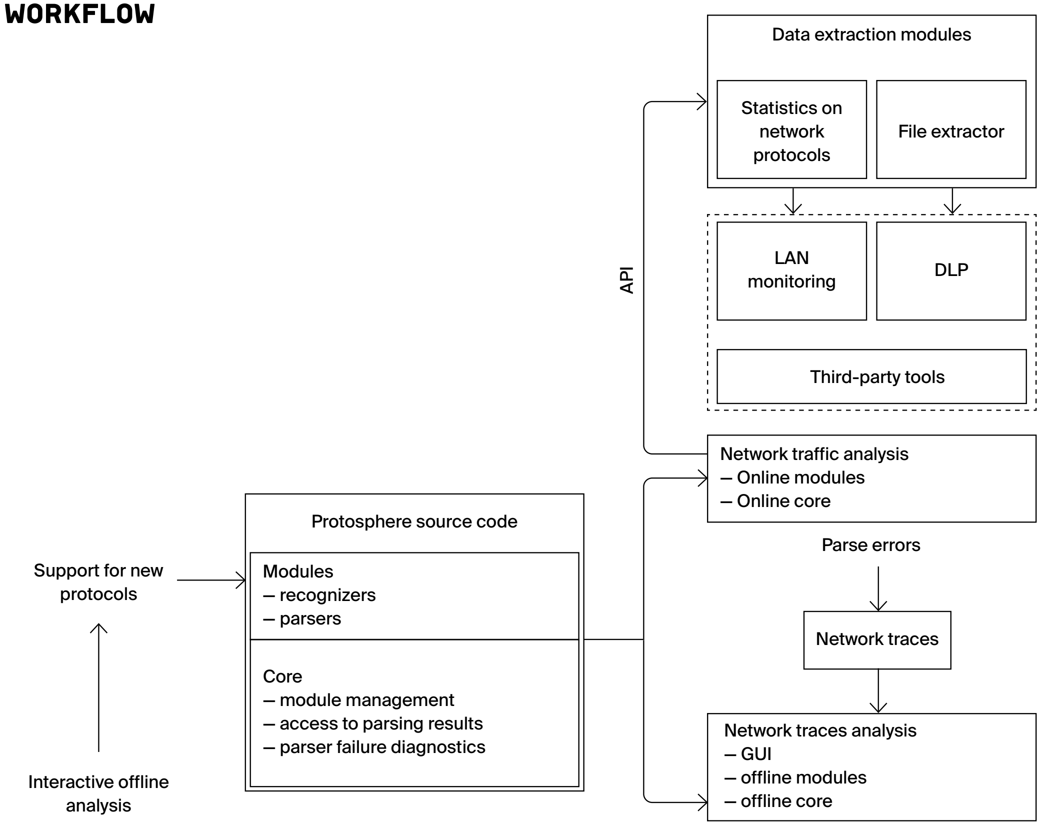
WHO IS PROTOSPHERE TARGET AUDIENCE?

- Companies that are testing network protocol implementations including those in embedded OS and network hardware.
- Developers of network security tools, such as firewalls and IDS/IPS.
- Manufacturers of network hardware that must be certified.
- Companies requiring real-time control and monitoring of network channels.

SUPPORTED PLATFORMS AND ARCHITECTURES

Architecture: Intel x86-64.
Platforms: Windows and Linux-based OSes.

WORKFLOW



REQUALITY: REQUIREMENT MANAGEMENT TOOL

Requality is an extensible tool for requirements management (mainly in software system development). It allows to develop requirements from scratch as well as to identify them from existing document views, preserving the links between the requirements and the text fragments. It supports the possibility of collaborative work on requirements using the GIT version control system.

FEATURES AND ADVANTAGES

Requality is a Java application for working with the requirements database, the user interface as an extension for the open Eclipse platform and the application programming interface (API). The functionality is close to existing commercial counterparts (IBM DOORS, Jama, Polarion) and surpasses some of the existing open-source products (aNimble, ProR, RMTOO). The tool and user manual are available at the project website: https://requality.ru.

Requality provides:

- Structuring and storing a requirements catalog:
  - A requirements catalog is a structured set of requirements and other elements stored within a single workspace. The top level elements are projects in which individual sets of requirements are stored. This capability is used, among other things, to separate upper level requirements from the lower level requirements developed on their basis.
  - The catalog elements include the requirements themselves as well as other types of nodes. The tool supports a basic set of elements, including:
    - text nodes that are not requirements themselves, but provide context for dealing with requirements;
    - documentary representations of the requirements on the basis of which the catalog was developed;
    - report settings and the results of their generation;
    - comments;
    - descriptions of test situations.

  The set can be scaled by developing extensions.
  - Object identification is supported in several ways, including the use of a unique numeric identifier within the project and a composite human readable hierarchical path;
  - Object properties include both those provided by default tools (node description, short string identifier, and others) and user-defined parameters used to indicate element qualities;
  - The use of HTML markup in the text of requirements and in other features makes it possible to use different tools to highlight text and provide supporting resources, such as images and tables.
- Link management, traceability and coverage analysis:
  - The definition of links between catalog elements is done both by defining links directly at the level of individual nodes, and by using the mechanics of automatic construction through the “definition-use” relation of terms;
  - The link between a text fragment and a requirement allows, on the one hand, to determine the origin of an individual requirement, and on the other hand, it makes it possible to automatically transfer such links to new versions of documents;
  - Links traceability is the ability to trace both the original requirements and the requirements developed on their basis for an individual requirement, as well as to examine the context of the catalog element within which it is to be considered;
  - Coverage is a collection of information on external resources associated with catalog items. The tool supports the use of external coverage information in the form of a specific file format, and provides an extensible set of coverage data sources.
- Change management and collaboration support:
  - GIT is supported as the primary system for collaborative work on the requirements catalog. A simplified set of commands for submitting changes and updating the local version of the project is available in the interface of the tool;
  - The interface of the tool makes it possible to view the versions of a single node as well as those of the catalog as a whole; it is also possible to compare individual versions.
- Report generation, in particular:
  - Creating various formats of the requirements catalog, including those that provide using it for offline work (outside the scope of the tool), as well as exchanging requirements with other tools or solving non-standard tasks within the development process;
  - Providing traceability data to view information on the relationships between catalog elements;
  - Comparing catalog versions to study the differences in the structure and properties of the project requirements for the selected versions of the catalog;
  - Coverage analysis to examine the status of individual catalog elements in terms of coverage information obtained from a selected source;
  - Support for user-defined templates using available information on the catalog, its versions, and coverage information.

- A programming interface (API) with the ability to modify stored data and create new projects is supported. It can also be used to exchange data with third-party tools.
- It is possible to develop extensions to define new elements, sources of information on coverage, or to get new functionality.

SYSTEM REQUIREMENTS

Windows OS or GNU/Linux based OS, Java 8.

REQUALITY DEPLOYMENT STORIES

Requality has been in development since 2011. It has been used to work with requirements in a project to develop a real-time operating system, as well as to catalog requirements from various standards (including TTCN and POSIX) in order to perform subsequent conformance testing.

2 DATA ANALYSIS

INFRASTRUCTURE PROJECTS

NATURAL LANGUAGE PROCESSING

DOCUMENT PROCESSING

APPLICATIONS

ASPERITAS AND CLOUD SOLUTIONS FAMILY

Asperitas is a platform for data storage and performing complex compute-intensive tasks in scientific, educational, and commercial projects. It includes a cloud environment also called Asperitas, as well as Michman, a PaaS orchestrator, and Clouni, a multi-cloud IaaS orchestrator based on TOSCA standard. Fanlight, a web laboratories platform, and Cotea, a system tool intended for programmatic control of Ansible scripts execution, are also a part of ISP RAS cloud solutions family.

ASPERITAS CLOUD ENVIRONMENT

Asperitas cloud environment is based on Openstack and Ceph, which are the modern standard of large private cloud systems. The distribution delivery is provided as a ready-made solution with everything necessary for deployment, including a TUI installer.

Other advantages of Asperitas:

- An onsite installation option (the provided infrastructure can be installed and fully controlled in an isolated environment due to the usage of open standards and software as well as ISP RAS research).
- High security: the environment is built on top of a smaller code base and uses its own know-how solutions that increase security.
- Standard interfaces of virtual and computational clusters management using Keystone, Neutron and Nova systems.
- Block storage and scalable object storage is based on the Ceph distributed file system.
- Adaptation to specific problem classes (e.g. continuum mechanics, big data analysis, program analysis for defect detection etc.).

Asperitas cloud environment is included in the Unified Register of Russian software (No. 5921).

CLOUNI, A MULTI-CLOUD ORCHESTRATOR

GitHub →
https://github.com/ispras/clouni

To enhance the capabilities of infrastructure resource management, ISP RAS is developing the Clouni tool, which allows deploying clusters of virtual infrastructure according to TOSCA Simple Profile normative templates using the Ansible configuration management tool.

Main characteristics of Clouni orchestrator include:

- No dependency on the cloud platform being used. Currently, it supports Openstack, Amazon AWS, and partially Kubernetes, with plans for Google Cloud and other platforms.
- Fine-tuning of virtual machines, security groups, ports and networks. Ability to run custom Ansible scripts as described in TOSCA interface operations during deployment, including software configuration on virtual machines.
- Microservice architecture that makes it possible to separate the processes of translating normative TOSCA templates into an internal representation, storing the state of deployed clusters, generating Ansible scripts and controlling their execution.
- Parallel deployment of independent nodes following the graphical hierarchy described in TOSCA Simple Profile.
- Support for TOSCA Simple Profile v1.0 and partially v1.3.

MICHMAN, A UNIVERSAL ORCHESTRATOR

GitHub →
https://github.com/ispras/michman

Michman is a PaaS services orchestration tool for a cloud environment performing big data analysis, machine learning, load management tools, and other tasks. It supports automatic cluster deployment in cloud environment, taking into account user requirements and parameters. It also provides an interface for creating predefined service templates on demand, including:

- A big data analysis cluster with arbitrary number of nodes having Apache Spark, Apache Hadoop, and Jupyter Notebook fully set up and ready to work.
- A database for storing large data in distributed mode: PostgreSQL, Apache Cassandra, Redis, etc.
- NextCloud storage and file exchange system in one of the supported databases.
- Slurm, a cluster management and job scheduling system with the option of GPU use.
- Kubernetes, a flexible container orchestration system, and tools running on top of it.

FANLIGHT

Fanlight is a platform for providing virtual desktops (DaaS - Desktop as a Service). It allows deploying SaaS infrastructure for computing web-laboratories. It was created as a result of ISP RAS participation in the University Cluster program and in the international Open Cirrus project (founded by HP, Intel and Yahoo). Fanlight is based on container technologies, unlike most solutions of this class based on virtual machines. Initially, the platform had been based on the Docker Compose technology. Later on, a Kubernetes-based implementation appeared. It only supports applications developed for Linux kernel-based OS. Fanlight is included in the Unified Registry of Russian Software (No. 6066).

Other advantages of Fanlight:

- High efficiency of work with cloud calculations due to the use of containers:
  - comfortable work with heavy engineering CAD-CAE applications requiring 3D graphics hardware acceleration support for complex visualization;
  - Support for running MPI, OpenMP, CUDA applications through access to HPC clusters, multicore processors, and NVIDIA graphics accelerators.
- Extended computing capabilities at the PaaS level through connecting hardware resources (HPC/BigData clusters, storage systems, graphic accelerator servers).
- Possibility of customization for a given application area due to integration of specialized calculation application packages and the easy way to add them. In particular, the following have been implemented:
  - in the field of MSS: OpenFOAM, SALOME, Paraview, etc;
  - in the field of Gas&Oil: tNavigator, Eclipse, Roxar, Tempest, etc.
- Operation via any thin client (including mobile devices) without any auxiliary software.
- Deployment on a server, computing farm, cloud (from the IaaS layer), in a Kubernetes cluster, or in dedicated cloud data center. The Kubernetes-based version also provides the opportunity to use different CRI container execution engines.

COTEA

GitHub →
https://github.com/ispras/cotea

Cotea is a tool that makes it possible to run Ansible programmatically and control its execution (Ansible is one of the most popular software deployment systems). Cotea allows to:

- use software control of running Ansible by iterating over the component parts of the Ansible script;
- embed Ansible into other systems;
- debug Ansible runs, including interactive mode; switching to interactive mode occurs in case of a task (part of the Ansible script) execution failure. Examples of functions provided in interactive mode:
  - restart a task that resulted in an error;
  - continue executing the Ansible script without the failed task;
  - add a new Ansible variable during runtime;
  - add a new Ansible task during runtime.

Interactive mode makes it possible to refrain from executing a script all over again in case of errors, which is especially important when working with large scripts.

Cotea is currently used in the deployment of the Asperitas platform, and is also a component of Michman.

CLOUD SOLUTIONS DEPLOYMENT STORIES

The computing cluster based on Asperitas analyzes information flows in the Talisman framework and supports other ISP RAS technologies (e.g. analyzing Android OS using Svace). The following projects were also implemented: a joint project with Huawei (large graphs analysis using big data processing), and the Tizen OS lifecycle support infrastructure that allows organizing joint development of OS components and automating regular build and testing of OS images. In addition, a number of projects is performed jointly with the Ministry of Education and Science of Russian Federation.

The Fanlight platform was used in a number of joint projects for web laboratory deployment, including Russian Federal Nuclear Center of the All-Russian Scientific Research Institute of Experimental Physics, OOO RRS-Baltika, Keldysh Institute of Applied Mathematics (developing a technology for increasing and using efficiently the hydrocarbon raw materials resource potential of the Union State of Russia and Belarus), ISP RAS Laboratory of Continuum Mechanics (https://unicfd.ru).

TALISMAN: PLATFORM FOR CONSTRUCTING INTELLECTUAL ANALYTICAL SYSTEMS

Talisman is a unified set of tools that automate typical data processing tasks, such as data retrieval, integration, analysis, storage and visualization. It ensures the fast development of specialized multi-user intellectual analytical systems that merge and work uniformly with the data from private databases and Internet sources (including social networks).

FEATURES AND ADVANTAGES

Talisman unifies the tools necessary for big data and cutting-edge AI tools, using them to extract information from random sources. It makes it possible to quickly create intelligent analytical systems using low-code and no-code approaches. It is constantly learning from the results of analyst work without the need for additional labor.

Talisman provides:

- A rich set of reusable components that have APIs for easy management and integration:
  - Data retrieval components. They include a framework for Internet data collection, namely, from social media (Facebook, VKontakte, Twitter, Instagram, Odnoklassniki, YouTube, LinkedIn etc.), blogs, news, MediaWiki sites, developer portals etc. There is also a system for importing data from file storages and databases.
  - Automatic data analysis components. A set of tools allowing to transform input data of any format into a unified universal representation (in particular, Dedoc, developed by ISP RAS, is used). The documents in this representation are subjected to analysis with the help of machine learning methods. It is possible to add your own handlers as containers with REST API. The processing sequence is managed by Talisman.Stream system (No. 6045 in the Unified Registry of Russian Software).
  - Storage and indexing components. These include a number of databases and information search engines that store source data, automatic analysis results, and results of manual user work.

- An easy to use web interface that unifies all components requiring user interaction.
- A flexible modular architecture that allows adding new features to the interesting components without changing others.
- A scalable architecture that allows processing and storing more data just by adding more hardware without any software change.
- Specialized components that monitor system status, manage event log, perform deployment, authentication and authorization, access control, and unidirectional data transfer.
- Tools and methods for training machine learning models as well as for transferring existing algorithms to other knowledge domains.
- A configurable knowledge domain scheme that can be changed by a user when the system is in operation.
- Complete alienability of the systems under development. Each system can be deployed at the customer's site, either on existing hardware or as part of a hardware-software system.
- Integration with private customer systems via provided APIs managing all components.
- License purity. Talisman is based on open source and know-how ISP RAS tools.

TALISMAN APPLICATION AREAS

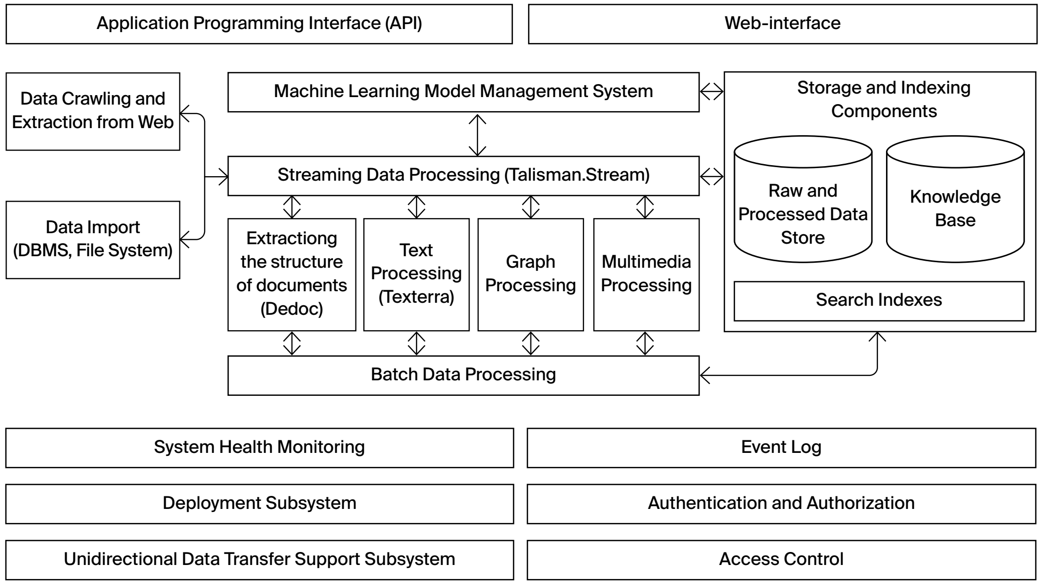
Talisman makes it possible to create analytical systems for a wide range of applications. Application examples:

- Automated knowledge base construction for a given knowledge domain and non-stop monitoring for new information regarding objects of interest (analogous to Palantir Gotham).
- Competitor intelligence based on open sources (OSINT), analogous to Maltego.
- Optimization of personnel management: effective selection of employees, verification of questionnaire data, detection of incorrect behavior in the open information space (Talisman. Biography system, No. 5547 in the Unified Register of Russian Software).
- Identification of information campaigns that manipulate the opinion of the target audience, as well as determining the target audience the campaign is aimed at.
- Identifying and analyzing the specifics of information distribution infrastructure (resources, users, bots), as well as analyzing the typical roles of community members in communication (source, opinion leader, distributor, moderator, bot, commentator).
- Managing the business reputation of people and organizations: monitoring relevant messages, identifying problems that cause dissatisfaction, monitoring leaks and internal information disclosure.
- Objective evaluation of performance and testing strategies on target audiences for feedback.
- Management of social tension points; detection and timely prevention of conflict escalation.

SUPPORTED LANGUAGES

Talisman uses advanced artificial neural networks to analyze data. The tools used make it possible to extract information from more than 100 natural languages.

TALISMAN WORKFLOW


Application Programming Interface (API)
Web-interface
Data Crawling and Extraction from Web
Data Import (DBMS, File System)
Machine Learning Model Management System
Streaming Data Processing (Talisman.Stream)
Extractiong the structure of documents (Dedoc)
Text Processing (Texterra)
Graph Processing
Multimedia Processing
Batch Data Processing
Storage and Indexing Components
Raw and Processed Data Store
Knowledge Base
Search Indexes
System Health Monitoring
Event Log
Deployment Subsystem
Authentication and Authorization
Unidirectional Data Transfer Support Subsystem
Access Control

LINGVODOC: VIRTUAL LAB FOR DOCUMENTING ENDANGERED LANGUAGES

GitHub →
https://github.com/ispras/lingvodoc

Lingvodoc is a system intended for collaborative multi-user documentation of endangered languages, for creating multi-layered dictionaries and performing scientific work with the received sound and text data. This is a joint project with the Institute of Linguistics of the Russian Academy of Sciences and Tomsk State University. Lingvodoc is under active development since 2012 and can be found on lingvodoc.ispras.ru.

FEATURES AND ADVANTAGES

Lingvodoc is an open source cross-platform system based on an innovative research (https://github.com/ispras/lingvodoc, https://github.com/ispras/lingvodoc-react).

Lingvodoc provides:

- Collaborative work on adding new information to dictionaries (as opposed to the similar Starling project that does not support this feature).
- Saving full history of user actions.
- Working with audio-textual corpuses and dictionaries simultaneously based on the integration with the ELAN system developed by Max Planck Institute of Psycholinguistics (Netherlands).
- Creating and editing unidirectional and bidirectional connections between lexical entries within dictionaries as well as external connections between dictionaries.
- Recording, playing and storing marked-up sounds (in WAV, MP3 and FLAC formats), as well as constructing vowel formants followed by data visualization.
- Advanced search supporting multiple parameters (as opposed to the similar TypeCraft project).
- Ability to search data on a map with automatic demarcation of isoglosses.
- Conflict-free bilateral delayed synchronization.
- High automation level (compared to the similar Kielipankki project): ability to carry out automatic etymological and phonetic analysis.

- Creating dictionaries of any structure, such as typical two-layer dictionaries with lexical entry layer and paradigms layer or multi-layer dictionaries. Importing existing dictionary structures is also supported.
- Algorithms mimicking the scholars' work on phonetic and etymological analysis.
- Support for storing text corpora in Word format, and dictionaries in the Excel format.
- Built-in morphological analysis for the languages of the ethnicities of Russia in the Aperitum format.
- A convenient interface for disambiguating homonyms after completing morphological analysis.
- Either using the ISP RAS cloud infrastructure resources or locally deployed resources with data isolation.
- Desktop and web-based versions.
- Open registration (confirmation required).
- Fast development for extending the system features as well as easy adapting to other fields of knowledge.

WHO IS LINGVODOC TARGET AUDIENCE?

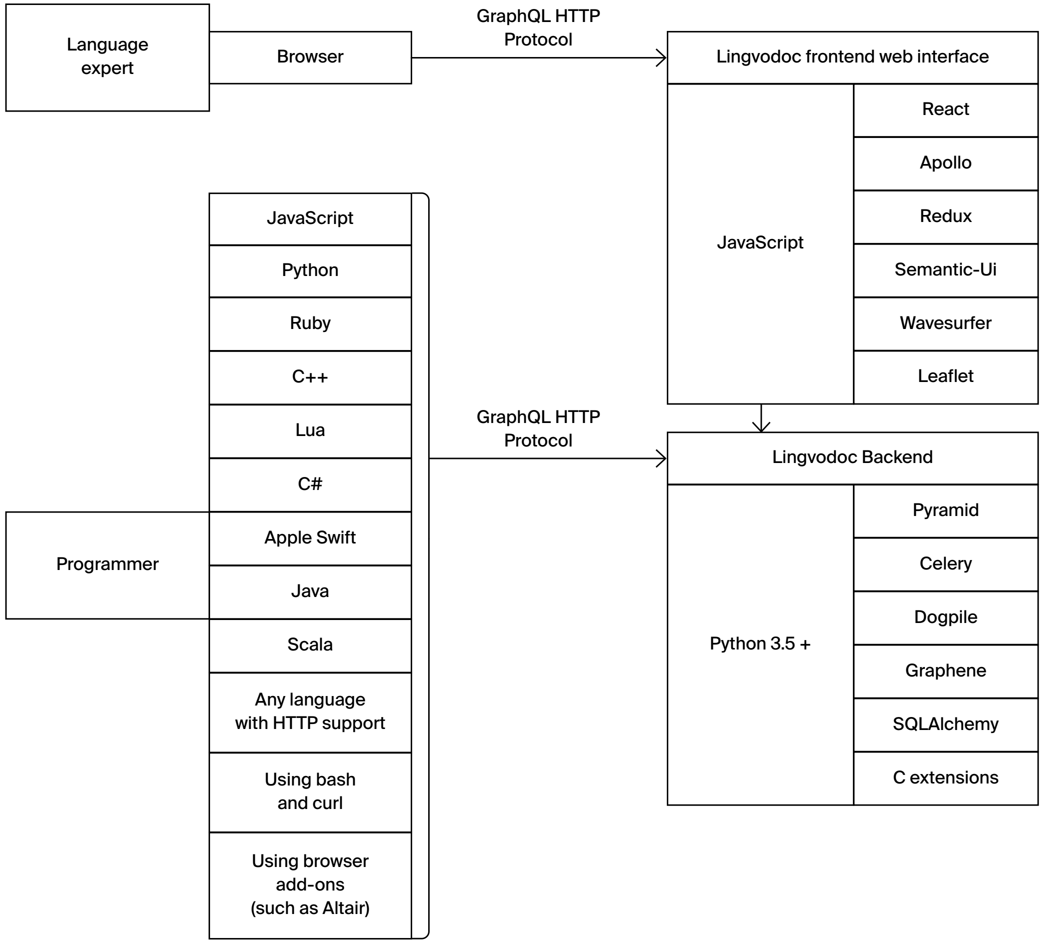
Lingvodoc is designed primarily for linguists performing a research in the area of documenting the endangered languages of Russian ethnicities. However, it is possible to adapt the technology for other purposes.

LINGVODOC DEPLOYMENT STORIES

Lingvodoc is currently used by philologists in 29 universities and scientific centers of 16 cities, including Tomsk State University, Institute of Philology (Siberian Branch of RAS). Institute of history, language and literature (Ufa scientific center of RAS), Udmurt Federal Research Center UB RAS, North-Eastern Federal University, Ugra State University, Institute of Linguistics, Literature and History (Karelian Research Centre of RAS), Murmansk Arctic State University. Specialists using the platform are ready to teach master classes for their colleagues.

In 2022, three groups of scientists from more than ten cities of the Russian Federation were engaged in additional education courses on “Using the Lingvodoc platform capabilities in the work of linguists.” For five days of intensive training (50 hours of full-time classes), many made discoveries that were highly appreciated by leading linguists. After the training, scientists from Bashkortostan have received funding from the government, and linguists from the Ammosov North-Eastern Federal University (Yakutsk) and the Kalmyk Science Center have received grants from the Russian Science Foundation to analyze their material using the Lingvodoc platform.

LINGVODOC WORKFLOW



TEXTERRA: SEMANTIC ANALYZER

Texterra is a scalable platform for extracting semantics from text. It contains the complete fundamental set of technologies for creating multifunctional applications for text analysis. Texterra bases its semantic analysis approach on concept identification. The platform is included in the Unified Register of Russian Programs (No.4048).

FEATURES AND ADVANTAGES

Texterra performs a unique analysis of Russian texts based on the identification of concepts instead of just words. It differs from foreign competitors by paying the most attention to Russian language. The analyzer builds on fundamental research results and integrates with the Elasticsearch search system greatly expanding its capabilities. The successful combination of technologies allows the platform to compete with the projects similar to IBM Watson Natural Language Understanding.

Texterra provides:

- High text processing speed (morphological analysis: 69 000 words per second, syntactic analysis: 39 100 words per second, coreference resolution: 10 100 words per second, full text analysis: approximately 13 600 words per second).
- Maximum attention to Russian language (unlike similar spaCy and UDPipe projects, as well as IBM Watson Natural Language Understanding, which does not support the analysis of emotions and concepts in Russian texts).
- Large knowledge base (more than 7 million concepts).
- Building knowledge base without expert involvement (automatic construction and update using Wikipedia, MediaWiki, Linked Open Data, etc.).
- Scalability both in word processing speed and in knowledge base size (using Apache Ignite and the Asperitas cloud technology developed at ISP RAS).
- High text analysis accuracy due to a number of key features:
  - multi-level search by related concepts;
  - adaptability to slang, hashtags (#) and errors in text;
  - emotion analysis (with separation of attitude towards objects and their attributes);
  - determining relationships between people and companies based on text information;
  - detecting implicit object references in discussions.
- Fast adaptation and tailored solutions development.

CATALOGUE OF TECHOLOGIES

- Supporting two use cases:
  - as a deployed software system on a customer’s local server providing either HTTP REST-based or RMI protocol access;
  - online at https://texterra.ispras.ru/;
- Simple and fast support for specific domains and ability to integrate new languages backed up by a modern machine learning approach.

WHO IS TEXTERRA TARGET AUDIENCE?

- Corporate software developers (e.g. chat bot developers).
- Developers of semantic search systems for specific domains (such as information security, medicine, auditing, etc.).
- Developers of arbitrary text processing applications.

TEXTERRA DEPLOYMENT STORIES

Texterra has been productized in the joint projects with HP and Samsung (the project goals were to develop a technology for analyzing corporate reports or supporting smart TVs). Currently Texterra backs up several ISP RAS innovative products, e.g. Talisman social media analysis. A number of Russian government agencies also use Texterra.

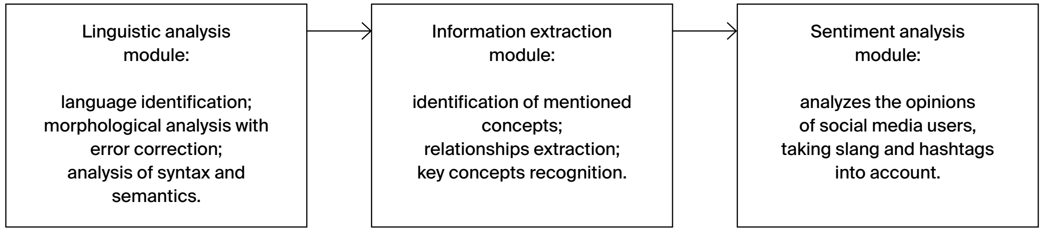
SUPPORTED LANGUAGES

Texterra analyzes Russian and English texts.

SYSTEM REQUIREMENTS

- A system supported by Java 8.
- 16 Gb RAM or more for each supported language.
- 64-bit operating system is recommended.

TEXTERRA WORKFLOW



Linguistic analysis module:

language identification; morphological analysis with error correction; analysis of syntax and semantics.

Information extraction module:

identification of mentioned concepts; relationships extraction; key concepts recognition.

Sentiment analysis module:

analyzes the opinions of social media users, taking slang and hashtags into account.

DEDOC: DOCUMENT STRUCTURE RETRIEVAL SYSTEM

Dedoc is an open universal system for converting documents to a unified output format. It extracts a document’s logical structure and content, its tables, text formatting and metadata. The document’s contents are represented as a tree storing headings and lists of any level. Dedoc can be integrated in a document contents and structure analysis system as a separate module.

FEATURES AND ADVANTAGES

Dedoc is implemented in Python and works with semi-structured data formats (DOC/DOCX, ODT, XLS/XLSX, CSV, TXT, JSON) and unstructured data formats like images (PNG, JPG etc.), archives (ZIP, RAR etc.), PDF and HTML formats. Document structure extraction is fully automatic regardless of input data type. Metadata and text formatting is also extracted automatically.

Dedoc provides:

- Extensibility due to a flexible addition of new document formats and to an easy change of an output data format.
- Support for extracting document structure out of nested documents having different formats.
- Extracting various text formatting features (indentation, font type, size, style etc.).
- Working with documents of various origin (statements of work, legal documents, technical reports, scientific papers) allowing flexible tuning for new domains.
- Working with PDF documents containing a text layer:
  - Support to automatically determine the correctness of the text layer in PDF documents;
  - Extract content and formatting from PDF-documents with a text layer using the developed interpreter of the virtual stack machine for printing graphics according to the format specification.
- Extracting table data from DOC/DOCX, PDF, HTML, CSV and image formats:
  - Recognizing a physical structure and a cell text for complex multipage tables having explicit borders with the help of contour analysis.

- Working with scanned documents (image formats and PDF without text layer):
  - Using Tesseract, an actively developed OCR engine from Google, together with image preprocessing methods.
  - Utilizing modern machine learning approaches for detecting a document orientation, detecting single/multicolumn document page, detecting bold text and extracting hierarchical structure based on the classification of features extracted from document images.

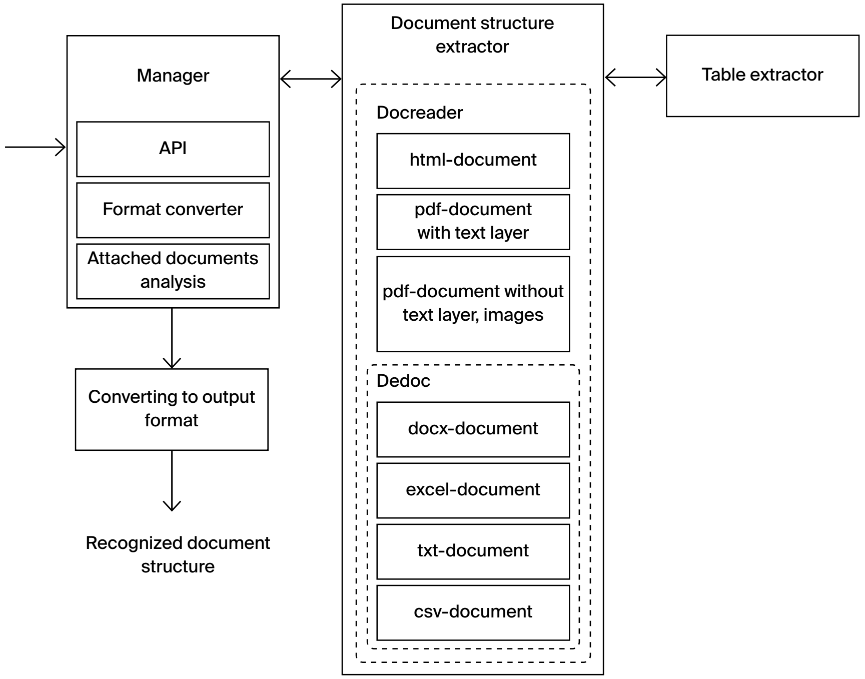
WHO IS DEDOC TARGET AUDIENCE?

- Developers of document contents analysis and management systems.
- Developers of intellectual text analysis algorithms.
- Developers of automatic document processing systems.

SUPPORTED LANGUAGES

Russian and English

WORKLOW


Manager
API
Format converter
Attached documents analysis
Converting to output format
Recognized document structure
Document structure extractor
Docreader
html-document
pdf-document with text layer
pdf-document without text layer, images
Dedoc
docx-document
excel-document
txt-document
csv-document
Table extractor

DOCMARKING: DOCUMENT LEAKAGE PREVENTION

Docmarking is a unique system for embedding digital watermarks into text documents. It allows creating a digital or physical document copy that is almost indistinguishable from the original yet exactly identifies the user or the device that was the intended recipient.

FEATURES AND ADVANTAGES

Docmarking is based on research results in the areas of steganography, digital image processing, and machine learning. The marking system builds on the methods for text detection and classification in images and uses statistical features of document images.

Docmarking has a number of advantages compared to competing technologies. Watermark extraction does not require the original document. The system supports embedding a watermark in the same scanned document multiple times, and the previous watermark is erased when the new one is being embedded.

Docmarking provides:

- Marking algorithms based on machine learning.
- Support for documents of all formats.
- Embedding a watermark when a document is displayed on a screen or printed.
- Standalone setup and work on the client side.
- Real time registering of the embedded watermarks on the server.
- Centralized 24/7 monitoring of the connected client devices.

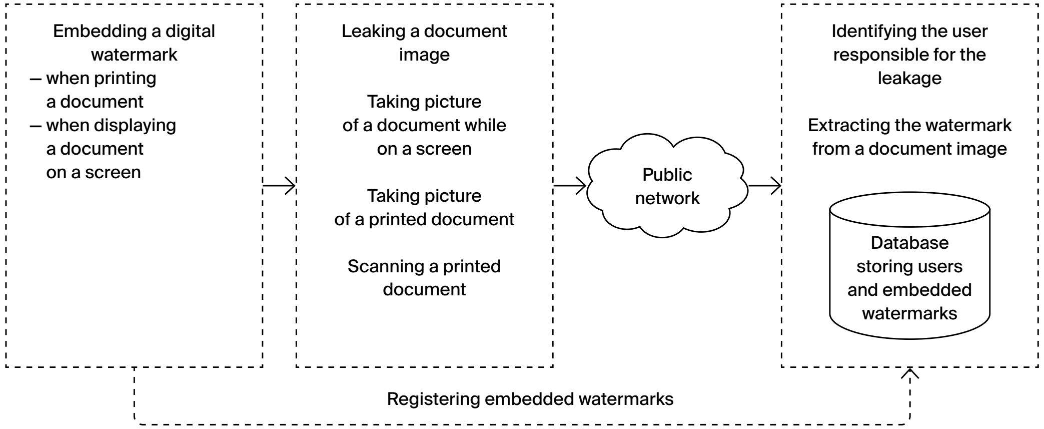
WHO IS DOCMARKING TARGET AUDIENCE?

- Government entities and public offices.
- Companies that would like to enforce their guides for handling classified documents.

SUPPORTED OPERATING SYSTEMS

Windows (32-bit, 64-bit), Linux (64-bit), including Astra Linux 1.6 SE.

DOCMARKING WORKFLOW


Embedding a digital watermark
— when printing a document
— when displaying a document on a screen
Leaking a document image
Taking picture of a document while on a screen
Taking picture of a printed document
Scanning a printed document
Public network
Identifying the user responsible for the leakage
Extracting the watermark from a document image
Database storing users and embedded watermarks
Registering embedded watermarks

SCINOON: EXPLORATORY SEARCH SYSTEM FOR RESEARCH GROUPS

SciNoon is a system for collaborative exploration of scientific papers. SciNoon is an essential tool for a group of researchers to dive quickly into the new area of knowledge and to find answers on their questions, following up with tracking new research on the topic of interest with highly customizable alerts.

FEATURES AND ADVANTAGES

SciNoon is an innovative system designed to optimize long-term teamwork with scientific papers. The papers could be added both from search systems and from digital libraries (like Google Scholar, arxiv.org, Semantic Scholar, PubMed) or could be uploaded directly as PDF files. The key feature of SciNoon is graphical research maps that all team members can add papers to.

SciNoon provides:

- Shared workspace for collaborative processing of found scientific papers.
- Zooming research map to control paper visualization details.
- Deduplication and cleansing of the uploaded metadata made possible by the internal database that allows discovering connections between papers or authors.
- Citation context classification into five classes depending on a citation role:
  - Background: a cited paper contains general information about the research area;
  - Use: a citing paper uses methods, data, and so on from the cited paper;
  - Compare: a citing paper points out differences (or similarities) with the cited paper;
  - Extend: a citing paper continues the research from the cited paper;
  - Weak: a citing paper criticizes the cited one pointing to the authors’ mistakes.
- Possibility to find relevant papers without a keyword search using an integrated recommendation system.
- A customizable list of questions whose answers should be looked up in a paper. Based on the retrieved answers the paper is visualized differently on the research map.

- Possibility to group similar papers into clusters.
- Notifications for team member actions, as well as support for quick opinion exchange within the team and helping each other.
- Collected answers analysis via built-in spreadsheet view and an option to export data to a CSV file for more complex processing.
- Tracking new papers on a given research topic and updating previously gathered results.

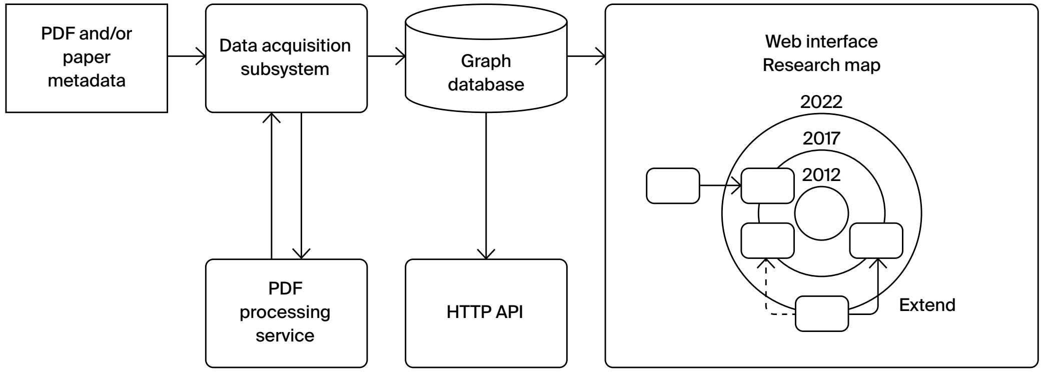
WHO IS SCINOON TARGET AUDIENCE?

- R&D department researchers that need a fast solution to their scientific problem.
- Scientists who need a tool for collaborative teamwork.
- Scientific advisers and their students who are doing exploratory search on research projects.

SCINOON DEPLOYMENT STORIES

SciNoon is used in ISP RAS while doing research and when advising students.

SCINOON WORKFLOW


PDF and/or paper metadata
Data acquisition subsystem
Graph database
Web interface Research map
2022
2017
2012
Extend
PDF processing service
HTTP API

ECGHUB: IN-DEPTH ANALYSIS OF DIGITAL ECG

EcgHub is a 12-lead ECG labeling system and neural network models' collection for pathology prediction. The system allows to predict the presence or absence of several pathologies, as well as to perform and review the syndromic ECG markup based on the verified questionnaire, thus providing a dataset for further development of neural network models.

FEATURES AND ADVANTAGES

EcgHub is based on research results in the areas of digital signal processing and machine learning algorithms. The pathology classification system is based on deep neural networks. The expert-verified approach provides consistent ECG labeling for training and further development of predictive models for screening and diagnosis of cardiovascular diseases.

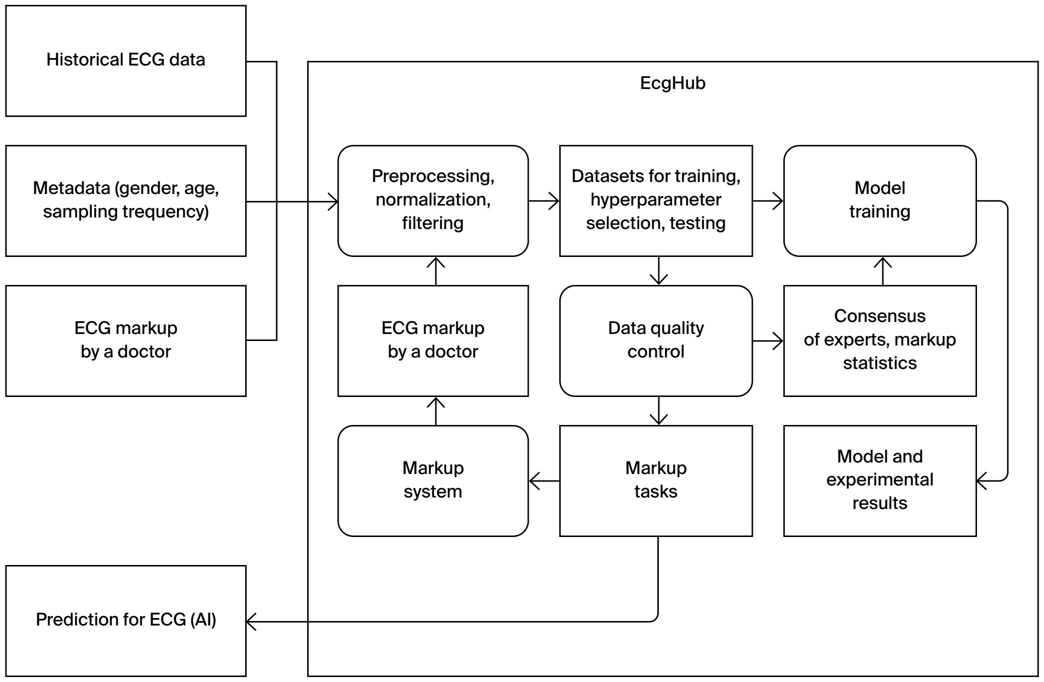
EcgHub provides:

- Trained neural networks for pathology prediction of digital ECG;
- Continuous development and refinement of neural network models, including fine-tuning for relatively small medical datasets;
- Adaptation of trained neural network models for pathology prediction of single-lead ECGs (cardiac chair, smart watches) as well as 24-hour ECGs (Holter monitoring);
- A consistent syndromic markup system to provide qualitative data for training predictive models;
- Integration of neural network models into the customer's digital circuit or remote access to the service at ISP RAS;
- Applying the markup system in the education of modern functional diagnostics professionals;
- Development of an automated population screening system.

WHO IS ECGHUB TARGET AUDIENCE?

- Medical institutions: the prediction of neural network models can be used as a second opinion;
- Educational institutions: verified datasets allow evaluating the knowledge of students or novice doctors of relevant specialties;
- Developers of devices and applications that perform ECG diagnostics autonomously.

WORKFLOW


Historical ECG data
Metadata (gender, age, sampling trequency)
ECG markup by a doctor
EcgHub
Preprocessing, normalization, filtering
Datasets for training, hyperparameter selection, testing
Model training
ECG markup by a doctor
Data quality control
Consensus of experts, markup statistics
Markup system
Markup tasks
Model and experimental results
Prediction for ECG (AI)

3 OTHER TECHNOLOGIES

CONSTRUCTIVITY 4D: INDEXING, SEARCHING, AND ANALYSIS OF LARGE-SCALE SPATIAL/ TEMPORAL DATA

Constructivity 4D is a technology for creating innovative software services that are capable of processing highly dynamic scenes and vast arrays of spatial and temporal data. It performs visual analysis of millions of objects with individual geometry and dynamic behavior. Constructivity is deployed within the Synchro system (Bentley Systems) that is used for 4D modeling of extremely large construction sites.

FEATURES AND ADVANTAGES

Constructivity 4D is a production level technology that puts together original methods of spatio-temporal indexing, search and qualitative and quantitative data analysis. Developed methods account for the specifics of objects' geometric representation, complex organization and the apriori known nature of their dynamic changes.

Constructivity 4D provides:

Support for a well-developed set of operations:

- Temporal operations implement classical interval algebra introduced by Allen with respect to time stamps of discrete events and their intervals.
- Metric operations allow determining the individual properties of geometric objects and the characteristics of their mutual arrangement. Diameter, area, volume, center of mass, planar projections, and distances between objects can be calculated for solid geometric objects.
- Topological operations are intended to classify the relative location of objects and to establish the facts of their coincidence, intersection, coverage, touch, overlap or collision. In contrast with known topological models such as DE-9IM, RCC-8, RCC-3D, these operations allow constructive implementation and are applicable for the analysis of complex objects.

- Orientational operations generalize known Frank's and Freksa's relative orientation calculi, cardinal direction calculi (CDC), oriented point relation algebra (OPRA) and are applicable for the analysis of objects with extended boundaries.
- Efficient query execution and typical problems solving, in particular, queries for reconstructing a scene at a given point in time, retrieving objects in a given spatial region, finding nearest neighbors, determining static and dynamic collisions, and conflict-free routing in a global dynamic environment are effectively resolved.
- A spatial-temporal indexing system including binary event trees, spatial decomposition trees, bounding volume trees, object cluster trees, space occupation trees.
- A hybrid computational strategy for determining collisions in scenes that combines methods for precise collision determination, collision localization methods using spatial decomposition, methods of hierarchies of bounding volumes, temporal coherence methods.
- An object-oriented library implemented in C++ that includes extensible set of classes, interfaces and related methods for specifying spatial-temporal data and executing typical queries.
- An original method for navigation in global dynamic environment that is based on extracting spatial, metric and topological information from geometric representation of 3D scenes and its concerted usage on path planning.
- Various options for extending the library so that it can be used both in new software applications development and in legacy applications.

WHO IS CONSTRUCTIVITY 4D TARGET AUDIENCE?

The technology is used for creating application systems in vastly different fields, including but not limited to: computer graphics and animation, geoinformatics, scientific visualization, design and manufacturing automation, robotics, logistics, project management and scheduling.

CONSTRUCTIVITY 4D DEPLOYMENT STORIES

The technology has been successfully deployed within the Synchro software system (https://www.bentley.com/en/products/brands/synchro) that is designed for visual 4D-modeling, planning and management of large-scale industrial projects in the construction and infrastructure areas, as well as others. Synchro is used in more than 300 companies in 36 countries.

DIGITEF: DIGITAL TWIN PLATFORM

DigiTEF is a software platform based on OpenFOAM and other open source tools, as well as unique modules and libraries developed at ISP RAS. DigiTEF solves various application problems of gas dynamics, aerodynamics, hydrodynamics, and acoustics. It is tailored for creating and working with highly sophisticated digital models of industrial devices. DigiTEF is included in the Unified Register of Russian Programs (No. 5377).

FEATURES AND ADVANTAGES

The platform delivers the same level of user experience as its competitors worldwide. DigiTEF core performance and accuracy evaluations compared with ANSYS Fluent and Star CCM+ showed similar (and in some cases lower) computational costs with the same accuracy.

The community of engineers, researchers, and industrial project developers has been formed around the DigiTEF platform.

DigiTEF provides:

- open source code (allows controlling and adapting implemented algorithms);
- the development pace of OpenFOAM+;
- automation tools for computation and model integration that allow integrated research of technical objects;
- possibility of developing additional components according to the specific requirements.

DIGITEF CONSISTS OF TWO MAIN BLOCKS:

1 OpenDTE, the platform core based on OpenFOAM. It contains the basic algorithms, procedures, and functions, as well as a set of third-party libraries in C++. It is fully open and can be obtained at https://github.com/unicfdlab. OpenDTEF consists of the following components:
- tools for modeling compressible flows;
- settings setup for advanced cases based on swak4Foam;
- parameterization based on Python. This allows automating calculation cases as well as integrating Salome, ParaView, and CodeAster software systems into DigiTEF.

2 Modules developed at ISP RAS:
- Data analysis for visualizing and retrieving information. It is designed to analyze results and build models of reduced dimension using data processing methods (FFT, POD, DMD, Hilbert transformations).

- Compressible flows simulation based on quasi-gas dynamics (QGD) equations, allowing to use the spatio-temporal averaging procedure to determine the main gas-dynamic quantities (density, velocity, temperature, and others).
- Incompressible flows simulation based on QHD equations.

The module is applicable in oceanology, convection, and subsonic flows problems.

- Incompressible and compressible flows simulation based on the Pimple and Kurganov-Tadmore hybrid algorithm.
- Subsonic turbulent flows simulation using the hybrid URANS / LES approach and low dissipative numerical schemes.
- Acoustic analysis. The module implements the Curle and Focs Williams-Hawkings analogies.
- Ice dynamics simulation.

WHO ARE DIGITEF USERS?

DigiTEF is designed for use in the facility of resource-intensive industries. Using digital twin models allows increasing engineering efficiency as well as reducing costs and complexity of the industrial projects implementation.

DEPLOYMENT STORIES

DigiTEF is used in several projects in the fields of wind energy, aerospace, aviation, metallurgy, as well as in the oil and gas industry. DigiTEF open source modules are successfully used in Institut Pprime (France), Korea Atomic Energy Research Institute (Korea), Universität der Bundeswehr München (Germany), Northwestern Polytechnical University (China), Embry-Riddle University (USA), California Institute of Technology (USA), etc.

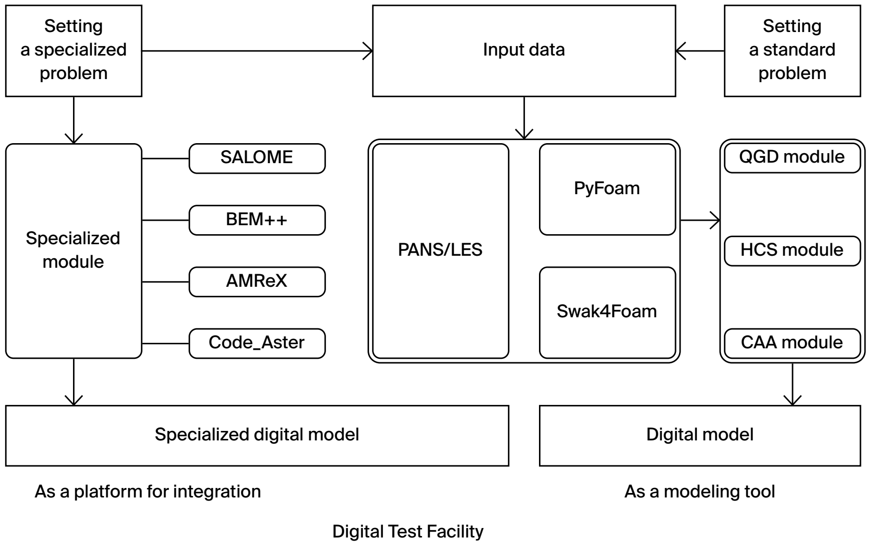
SYSTEM REQUIREMENTS

Linux OS. Other operating systems that support the Oracle VirtualBox virtual machine may also be used (on Microsoft Windows 10 via the Bash shell). Moreover, the performance loss due to virtualization does not exceed 5%.

Required RAM: 16 Gb or higher.

DigiTEF supports parallel computing, which significantly speeds up its work. Also it supports high performance computing systems (supercomputers and clusters) to accelerate calculations. Using up to 1536 computational cores was tested.

WORKFLOW



Ivannikov Institute for System Programming
of the Russian Academy of Sciences

25, Alexander Solzhenitsyn str. 109004 Moscow, Russia
E-mail: scsec@ispras.ru

Ivannikov Institute for System Programming of the RAS 2022