

# VEDLIoT: Very Efficient Deep Learning in IoT

M. Kaiser<sup>\*</sup>, R. Griessl<sup>\*</sup>, N. Kucza<sup>\*</sup>, C. Haumann<sup>\*</sup>, L. Tigges<sup>\*</sup>, K. Mika<sup>\*</sup>, J. Hagemeyer<sup>\*</sup>, F. Pormann<sup>\*</sup>, U. Rückert<sup>\*</sup>, M. vor dem Berge<sup>†</sup>, S. Krupop<sup>†</sup>, M. Pormann<sup>‡</sup>, M. Tassemeier<sup>‡</sup>, P. Trancoso<sup>§</sup>, F. Qararyah<sup>§</sup>, S. Zouzoula<sup>§</sup>, A. Casimiro<sup>¶</sup>, A. Bessani<sup>¶</sup>, J. Cecilio<sup>¶</sup>, S. Andersson<sup>||</sup>, O. Brunnegard<sup>||</sup>, O. Eriksson<sup>||</sup>, R. Weiss<sup>\*\*</sup>, F. Meierhöfer<sup>\*\*</sup>, H. Salomonsson<sup>††</sup>, E. Malekzadeh<sup>††</sup>, D. Ödman<sup>††</sup>, A. Khurshid<sup>‡‡</sup>, P. Felber<sup>x</sup>, M. Pasin<sup>x</sup>, V. Schiavoni<sup>x</sup>, J. Ménétrey<sup>x</sup>, K. Gugala<sup>xi</sup>, P. Zierhoffer<sup>xi</sup>, E. Knauss<sup>xii</sup>, H. Heyn<sup>xii</sup>  
<sup>\*</sup>Bielefeld University, Germany — <sup>†</sup>christmann informationstechnik + medien GmbH & Co. KG, Germany  
<sup>‡</sup>Osnabrück University, Germany — <sup>§</sup>Chalmers University of Technology, Sweden — <sup>¶</sup>University of Lisbon, Portugal  
<sup>||</sup>VEONEER Inc., Sweden — <sup>\*\*</sup>Siemens AG, Germany — <sup>††</sup>EMBEDL AB, Sweden — <sup>xii</sup>Göteborg University, Sweden  
<sup>‡‡</sup>Research Institutes of Sweden AB (RISE) — <sup>x</sup>University of Neuchâtel, Switzerland — <sup>xi</sup>Antmicro, Poland

**Abstract**—The VEDLIoT project targets the development of energy efficient Deep Learning for distributed AIoT applications. A holistic approach is used to optimize algorithms while also dealing with safety and security challenges. The approach is based on a modular and scalable Cognitive IoT hardware platform. Using modular microserver technology enable the user to configure the hardware to satisfy a wide range of applications. VEDLIoT offers a complete design flow for Next-Generation IoT devices required for collaboratively solving complex Deep Learning applications across distributed systems. The methods are tested on various use-cases ranging from Smart Home to Automotive and Industrial IoT appliances. VEDLIoT is an H2020 EU project which started in November 2020.

## I. THE VEDLIoT APPROACH

Deep Learning has become a strong driver in Internet of Things applications. Typically, those applications have very challenging computational demands and a low energy budget. The goal of VEDLIoT is to accelerate Deep Learning applications and to optimise them towards energy efficiency. Figure 1 shows the architecture of VEDLIoT. We present the project following a bottom-up approach, starting with customisable hardware platforms using modular microservers and specialised hardware accelerators for heterogeneous computing. The optimisation of use cases is systematically accompanied from an early stage of development by using modelled requirement engineering and verification techniques. We combine expert-level knowledge of different domains to create a powerful middleware for optimising the underlying neural networks of deep learning algorithms and ease the development with frameworks for testing, benchmarking and deployment. A lot of development work goes into guaranteeing a high level of safety and security, which is essential for our use cases.

## II. ACCELERATED AIoT HARDWARE PLATFORM

Our hardware development focuses on extending and refining the already available platforms RECS|Box and t.RECS, which primary target cloud and near edge computing. uRECS

is developed within VEDLIoT and focuses on compact dimensions, low cost, and high energy efficiency to better suit AI and ML applications, especially for low-cost / low-power devices (Figure 1).

### A. Heterogeneous hardware platform

All RECS hardware platforms share a modular approach, which leads to a heterogeneous, adaptable hardware architecture supporting a wide range of applications and allowing for a future-proof design by an exchangeable/upgradable hardware basis [1], [2]. Another common feature is the scaleable communication-driven infrastructure, realizing efficient communication between heterogeneous microservers via 1G/ 10G Ethernet and high-speed low-latency connections [3].

Most supported microservers are based on mid/high-performance Computer-on-Module (COM) form factors, e.g., RECS|Box supports COM Express microservers and t.RECS the recently released COM-HPC Server and Client standards. As shown in Figure 2, several other, well-established form factors focus on low-power embedded computing. SMARC modules, for example provide a smaller footprint and support with x86, ARM and FPGA-SoC more target architectures.

uRECS closes the gap in hardware platforms towards embedded/ far edge computing with a power consumption of less than 15 W as required by the use cases. Next to SMARC microservers, it will also natively support Jetson Xavier NX modules. By using an adaptor-PCBs, which map the Jetson's pinout to other form factors, uRECS also integrates Xilinx Kria and Raspberry Pi compute modules. Extension slots based on USB and M.2 are used to use additional hardware accelerators.

VEDLIoT extends the classically static hardware architecture towards a dynamically configurable infrastructure for increased resource-efficiency and robustness. On system level, the RECS ecosystem enables easy exchange of computing resources and seamless switching between the different heterogeneous components. On communication level, e.g., the networking topology or protocol parameters can be adapted to cope with changing real-time, or bandwidth requirements. Finally, reconfigurable devices (FPGAs) will be utilized to enable the integration of new architectural concepts, developed in VEDLIoT. On this level, partial reconfiguration will be used

This publication incorporates results from the VEDLIoT project, which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957197.

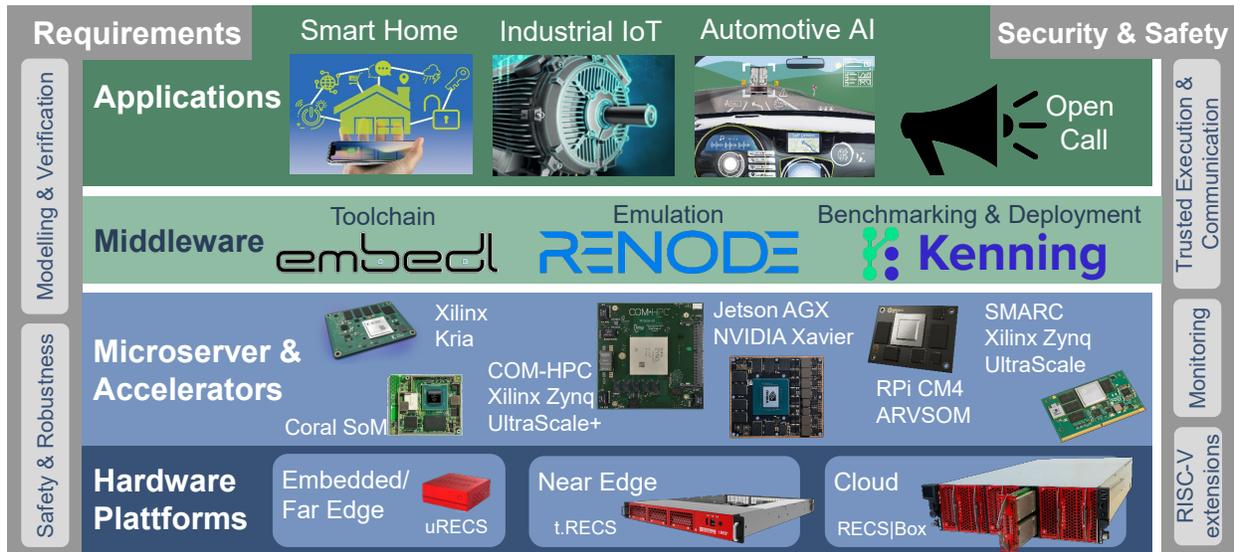


Fig. 1: VEDLIoT architecture overview

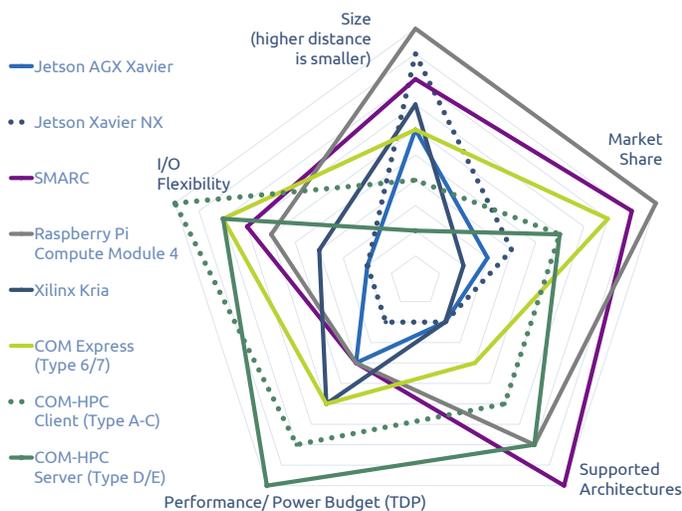


Fig. 2: Computer-On-Module (COM) form factors supported by VEDLIoT hardware platforms

to adapt to changing application requirements at run-time, e.g., by using implementations with different power/performance footprints.

### B. Accelerators and Microservers

One of the key components to delivering the required performance for the Deep Learning (DL) applications is the hardware accelerator. The demands of these applications triggered the development of new dedicated hardware (HW) accelerators. But the levels of abstraction that have helped the independent development of both software and hardware in the past can not be used any longer to achieve the best performance and efficiency for the most demanding workloads. The solution is to focus on hardware-software co-design [4]. In VEDLIoT we explore four different types of DL accelerator: (1) existing off-the-shelf; (2) statically configured; (3) dynamically reconfigurable; and (4) fully simultaneous co-

design accelerator. We evaluate and explore using existing accelerators for fast deployment of the required performance. In order to achieve a better fit to the use-case applications we will focus on the development of dedicated accelerators. We will use FPGAs to develop accelerator prototypes that can achieve higher performance and efficiency for the use-case applications. In addition, we will identify different modes of operation and offer accelerators dedicated to those modes of operation. These accelerators follow the partial co-design principles by mapping the DL models into the HW components. Nevertheless preliminary results have shown that no single accelerator can provide a better match to different models. Consequently, we will explore the fully simultaneous co-design where we develop the HW together with the software. In addition to mapping the model to the HW we will also give feedback to the models so that optimizations can be tuned for a better HW utilization.

In VEDLIoT we will focus not only on the computational units for the DL acceleration but we will also perform an in-depth study of how the memory is utilized in current accelerators and explore new approaches for the memory hierarchy for future DL accelerators.

With the many moving parts in the space of DL processing acceleration, VEDLIoT will use Renode, an open source simulation framework [5], to test the FPGA accelerator prototypes. During the course of the project, Renode will be enhanced with capabilities of simulating Custom Function Units, or CFUs.

A CFU is an accelerator hardware, tightly coupled with the RISC-V CPU, providing functionality designed specifically for the planned ML workflow. Programmed in a Hardware Description Language, CFUs will be used as an input for Renode to extend simulated cores.

Renode, a functional simulator for complex heterogeneous systems, provides an ability to simulate full SoCs and run the same software that would normally be used on hardware.

VEDLIoT will benefit from Renode’s testing and introspec-

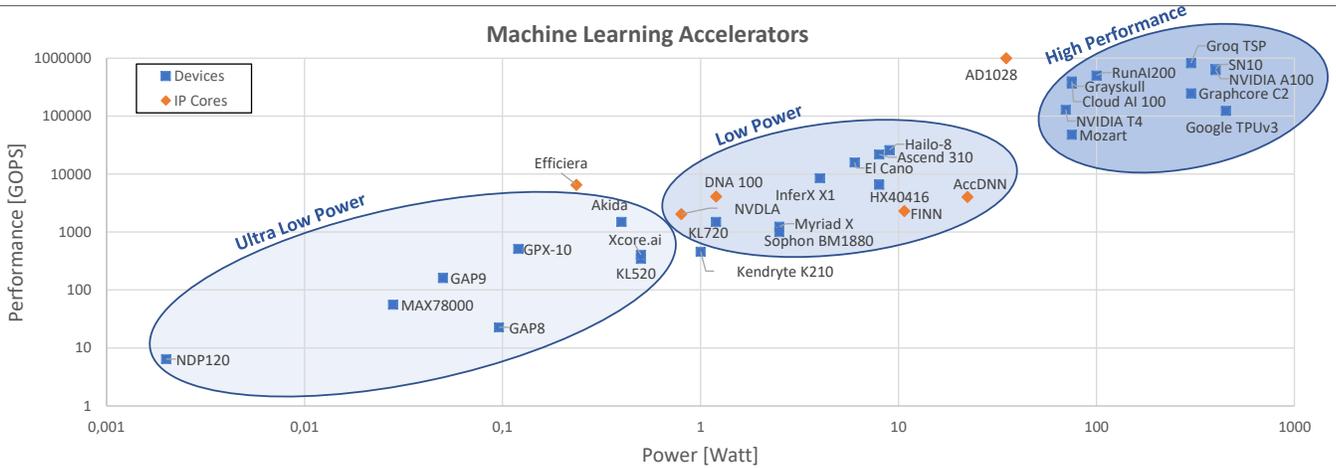


Fig. 3: Theoretical Performance of DL Accelerators

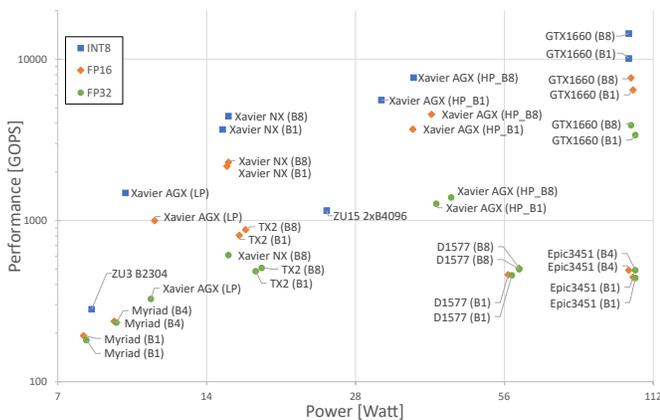


Fig. 4: YoloV4 performance evaluation of DL accelerators

tion capabilities, using it both for interactive development of accelerator prototypes and within a Continuous Integration environment. This will not only ease the development process, but will also make the final result more reliable.

### C. Performance Evaluation

A wide variety of hardware accelerators for deep learning is emerging on the market, targeting a wide range of applications from small embedded systems with power budgets in the order of milliwatt to cloud platforms with a power consumption exceeding 400 W. Figure 3 summarizes the accelerators, that have been analysed in detail in [6]. It has to be noted that the diagram shows a very high-level view. The data is based on the peak performance values (in Giga Operations per Second), provided by the vendors. No normalization to a specific technology node is performed and the architectures vary in the used precision, ranging from FP32 to INT8 and even binary weights are included. Nevertheless, an interesting fact is, that most architectures cluster around an energy efficiency of about 1 Tera Operation per Ws (1 TOPS/W), independent of their individual performance (or power demand).

For performance evaluation, the DL models ResNet50, MobileNetV3 and YoloV4 were chosen to determine comparable

performance values of available DL accelerators. Depending on the supported quantization of the hardware, the tests were executed using INT8, FP16 or FP32 datatypes. For best utilization, the used tools are chosen based on the manufactures recommendations, e.g. TensorRT for NVIDIA. In addition, performance and hardware utilization were optimized by varying the batch size from 1 to 8, which is represented in Figure 4 by B1, B4 and B8. In this figure, the performance (in GOPS) and the measured power consumption (in Watt) are shown exemplarily for YoloV4. The investigated platforms include x86 CPUs (Epic3451 and D1577), GPUs (GTX1660), eGPUs (Xavier AGX (in high performance and low power mode), Xavier NX and Jetson TX2), FPGAs (Zynq ZU15 and ZU3) and ASICs (Myriad). In VEDLIoT, performance and energy efficiency evaluations are an important basis for selecting DL accelerators to be integrated into the RECS platform, tailoring it towards the use cases.

### III. OPTIMIZING TOOL CHAIN FOR HETEROGENEOUS HARDWARE

Trained deep learning models have redundancy in their computational graph that can be exploited for optimizations. In some cases, models have been compressed down to 49x of their original size, with negligible loss in accuracy. This can be achieved by combining methods that remove connections and/or neurons, quantize parameters and activations and encode the parameters in a more compact form [7]. Although there has been a lot of recent research in the area, most of the results are theoretical speed-ups based on metrics, e.g. number of operations and reduction of parameters. The theoretical speed-ups do not always translate to more efficient execution in hardware [8]. In the VEDLIoT project, novel methods for hardware-aware optimization will be developed. Furthermore, the industry standard ONNX, which is an open format to represent machine learning models [9], will be used as input to ensure compatibility with the current open ecosystem. In the optimization phase, the model's computational graph will undergo significant surgery to optimize its execution latency, power consumption and/or memory footprint. Utilizing the

knowledge of the target hardware, will lead to optimizations that do translate to improved execution metrics when deployed.

Deploying deep learning models on edge devices usually involves the following steps:

- Preparation and analysis of the dataset, preparation of data pre-processing and output post-processing routines,
- Model training (usually transfer learning), if necessary,
- Evaluation and improvement of the model until its quality is satisfactory,
- Model optimization, usually hardware-specific optimizations (e.g. operator fusion, quantization, neuron-wise or connection-wise pruning),
- Model compilation to a given target,
- Model execution on a given target.

There are many different frameworks for most of the above steps (training, optimization, compilation and runtime). The cooperation between those frameworks differs and may provide different results.

Kenning [10], an open source framework developed by Antmicro, addresses the problem of enabling the tools to cooperate with each other. The interoperability is achieved by converting the models into a common representation using Open Neural Network Exchange (ONNX) format. All the intermediate conversions and optimizations are performed on ONNX models. At the final stage, Kenning converts the model to a selected neural network runtime and deploys it on the target hardware.

Based on the implemented interfaces, Kenning framework can measure the inference duration, resource usage and processing quality on a given target. Depending on a target, Kenning can monitor the following runtime parameters:

- Inference time,
- Mean CPU usage,
- CPU and GPU memory usage

Kenning can automatically benchmark the quality of processing of a given neural network mode and generate a confusion matrix for classification models and recall/precision graphs for detection algorithms.

#### IV. SAFETY, SECURITY AND REQUIREMENTS FOR DISTRIBUTED AIoT SYSTEMS

When combining deep learning with the properties of the Internet of Things, new concerns might arise that are not yet foreseen by standards and literature. The new concerns include aspects such as data quality, heuristic deep learning modelling, learning of the models, or even new ethical considerations. Additional stakeholder such as data engineers enter the stage, and common languages or interfaces need to be found between the different stakeholders. Typical architectural frameworks, such as the ISO 42010 [11] or the IEEE 2413 [12] standard cannot cope with concerns for systems that include some form of machine learning. One major challenge we identified in VEDLIoT is the difficulty to keep track of dependencies, e.g. through correspondence rules, between the different architectural views. Another problem of current architectural frameworks is the lack of a clear system development hierarchy,

which would support the early identification and mapping of dependencies between different architectural views [13].

##### A. Requirements concepts for AIoT

Designing a large, distributed system is a hierarchical process [14]. The architectural framework for VEDLIoT not only supports the seamless design and integration of traditional software components and deep learning components, but also allow for all necessary quality concerns to be taken into account as early as possible in the design process. The VEDLIoT architectural framework is organised by two aspects: Clusters of concerns, and level of abstraction. These aspects form a 2-dimensional grid of architectural views that guide the concept and design of a VEDLIoT system. Typical clusters of concerns for a system with deep learning components are *logical behaviour, process behaviour, context and constraints, learning setting, deep learning model, hardware, information, communication, ethical concerns, safety, security, privacy, and energy*. Levels of abstraction are *knowledge level, conceptual level, design level, and run-time level*. Each architectural view is categorised by the two aspects cluster of concern and level of abstraction. In VEDLIoT we show that dependencies between the architectural views only exist vertically between the views of the same cluster of concern or horizontally between architectural views on the same level of abstraction. This reduces the complexity of the system design challenge and allows for better traceability. Knowledge can become available on all levels of the architectural framework at any time. Traditionally, requirement engineering would be organised in a top-down fashion. However, the architectural framework supports middle-out systems engineering, which is a widely common practise in systems engineering, combining traditional top-down systems design with integration of designated lower level hardware, software, AI models, or other components [15].

The architectural framework supports the requirement engineering for distributed deep learning systems by creating links and dependencies between the desired use cases, the context in which the use case is valid in, data requirements, and quality attributes.

##### B. Safety aspects

Any model construction for systems using deep neural networks must be guided by safety requirements. However, safety standards that base on the EN-IEC 61508 standard,<sup>1</sup> such as ISO 26262 for the automotive industry, assume that for software only systematic faults exists. Safety standards emphasise processes for software development that help with avoiding systematic mistakes during the design of systems. However, due to the probabilistic nature of deep learning, the assumption that only systematic faults exist in software does not hold anymore. There is no absolute guarantee that a deep neural network performs as intended under the desired circumstances. The desired behaviour of the deep neural network depends on the data used for training and validation.

<sup>1</sup>Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems

Therefore, safety standards for systems with deep neural networks must encompass not only the deep neural networks, but especially the data used for training and validation.

From the perspective of defining an architecture and designing system solutions aiming at increasing the robustness of deep learning processes, VEDLIoT is focusing on monitoring approaches to detect faulty situations and to trigger appropriate reactive measures.

The work is being developed in two directions. Firstly, we are looking at the problem of characterizing the quality of the input data, detecting situations in which these data may have been accidentally or even maliciously compromised. A large set of data errors may be easily identified, may be corrected, or the affected data may be removed, to avoid the propagation of these errors through the DL models. We are developing different monitoring and error detection mechanisms, depending on the kinds of input data (e.g., time series, image) and on the error types (e.g., outliers, image noise). Secondly, we are looking at the problem of detecting errors on the output data, when these errors derive from systematic faults affecting the execution of DL models on devices or edge nodes. We consider that these faults may have been triggered or injected in runtime (e.g., hardware faults, attacks). In brief, the approach consists in periodically submitting both the input and the output data to a robustness service running on the cloud, which holds a copy of the DL model and can verify the correctness of the output data. To support all these monitors and monitoring mechanisms, an architectural pattern comprising two separate parts is considered, based on the concept of architectural hybridization [16]. The monitors must execute on a system part that is reliable and trustworthy by design, such that it is possible to argue about the safety and robustness of the monitoring process itself. More specifically, we intend to explore the trusted runtime developed in VEDLIoT (see the following subsection) to achieve the needed separation and trustworthiness of monitoring mechanisms.

### C. Security considerations

VEDLIoT implements a number of tools at hardware- and system-level to improve dependability and security of edge applications. We leverage hardware features for trusted execution environments combined with well-established dependability techniques to support both middleware and applications layers of the project. So far, the project focused on developing end-to-end trust through a distributed attestation mechanism, secure execution and communication of critical code on edge devices.

One of our first results include a trusted runtime designed to execute unmodified, language-independent applications. We leveraged the hardware protection offered by Intel SGX enclaves and an open-source WebAssembly runtime implementation to build a trusted runtime environment without dealing with language-specific APIs. WebAssembly is an increasingly popular lightweight binary instruction format, which can be efficiently embedded and sandboxed. Languages like C, C++, Rust, Go, and many others can be compiled into WebAssembly. Twine provides a secure, sandboxed software runtime

nested within an SGX enclave, featuring a WebAssembly system interface (WASI) for compatibility with unmodified WebAssembly applications. Twine evaluation shows that SQLite can be fully executed inside an SGX enclave via WebAssembly and existing system interface, with small performance overheads [17].

As part of our work in providing a hardware-trusted execution environment, we developed a novel Trusted Execution Environment (TEE) support for VexRiscv, an open-source RISC-V soft processor. Our implementation takes the form of a highly optimized RISC-V Physical Memory Protection (PMP) unit that enables secure processing by limiting the physical addresses accessible by software running on a processor hart. The PMP unit is configurable in the highest privilege level (the machine mode) and can be used to specify read, write and execute access privileges for a specific memory region. In small devices, that only support machine mode (M-mode) and user mode (U-mode), the PMP configurations can efficiently ensure secure execution of software in M-mode and U-mode. Our PMP implementation is part of the official VexRiscv implementation and the source-code and documentation are openly available for the research community.

To support trusted applications in embedded devices, we turned our attention to ARM SoCs, using TrustZone as a trusted execution environment (TEE), combined with the open-source and trusted operating system OP-TEE. TrustZone cannot directly support our initial work with Intel SGX because their design and implementation differ greatly. While SGX allows for normal user-level processes to create and attest protected enclaves, TrustZone splits the operating system into two parts, the normal world and the secure world. Trusted applications can only run in the secure world, and the operation necessary to change context between worlds is rather complex, and cannot be done at user-level. To implement remote attestation for WebAssembly code running in ARM processors, we leveraged OP-TEE, an open-source implementation a TEE specification defining how the trusted environment behaves and how the normal world can interact with the secure world. We based our implementation on a board that contains a key that is fused by the manufacturer and can be used as root-of-trust, as well as a secure boot mechanism, preventing an attacker from substituting the trusted software.

## V. VEDLIoT APPLICATIONS

In VEDLIoT, we have four use cases, divided into three groups: A. Automotive, B. Industrial IoT and C. Smart Home. The European Union sees the need for "trusted IoT and edge computing platforms" and "development and deployment of next generation computing components" with a focus on energy-efficiency for the next stage of digitisation [18]. Both, very high energy-efficiency as well as high security standards are parts of the VEDLIoT goals.

### A. Automotive

Amongst the numerous potential problems in the automotive sector, the Pedestrian Automatic Emergency Breaking

(PAEB) was chosen as a well specified example, which can be benchmarked and compared against state-of-the-art systems. The major development goals are the distribution of the deep learning models and the decision making between different on-car systems and edge devices at varying speeds and reliability of mobile networks.

Dynamic distributing of sensor data to edge stations is a quite new research topic and requires a quick monitoring of available mobile networks, their speed and latency, available computing resources of the edge devices and a management system that can quickly react to the actual situation. The overall goal is to optimise the energy-efficiency in total, but also to minimize the on-car energy consumption.

Sending raw sensor data via a mobile network to an edge station always implies a high security risk. Therefore, an integration of VEDLIoT’s remote attestation approach is of importance.

### B. Industrial IoT

In the section of Industrial IoT, we have two use cases: Motor Condition Classification and Arc Detection in DC power distribution cabinets.

The Motor Condition Classification use case aims to design and build a prototype of a battery-powered ultra-low energy deep learning-driven small box that can be attached to large electric asynchronous motors and which continuously monitor the motor. The states to monitor are the operational, thermal and mechanical conditions of the motor, and upon specified events, e.g. a ball bearing failure, a message will be sent to an operator. As state of the art, classical monitoring and decision making algorithms are used. By solving the problem via machine learning, it’s envisioned to not only increase the energy-efficiency, but also to monitor the different motor states more robust.

The Arc Detection use case has the goal to design and build a prototype system that can detect unwanted arcs in DC power distribution cabinets, using deep learning technology. A challenge is to guarantee a very low latency from the first spark till inference, including sensing and pre-processing, and an ultra-low false negative error rate for a smooth operation. In general, the arc localization will help for faster fault detection and repair of broken units. A further challenge is the standardized evaluation of the arc detection device: IEC 62606 [19] describes such devices and evaluation criteria, but has a strong focus on AC devices, while we are developing DC devices.

### C. Smart Home

The Smart Home use case has the goal to increase the safety, health and well-being of its residents and to achieve a certain level of comfort through automation. This use case targets the development and acceleration of AI-based methods for a demand-oriented interaction between the user and a smart home. To achieve this goal, an intuitive and natural operating interface is crucial, which has been build using a smart-mirror device. As seen in Figure 5, a camera and a

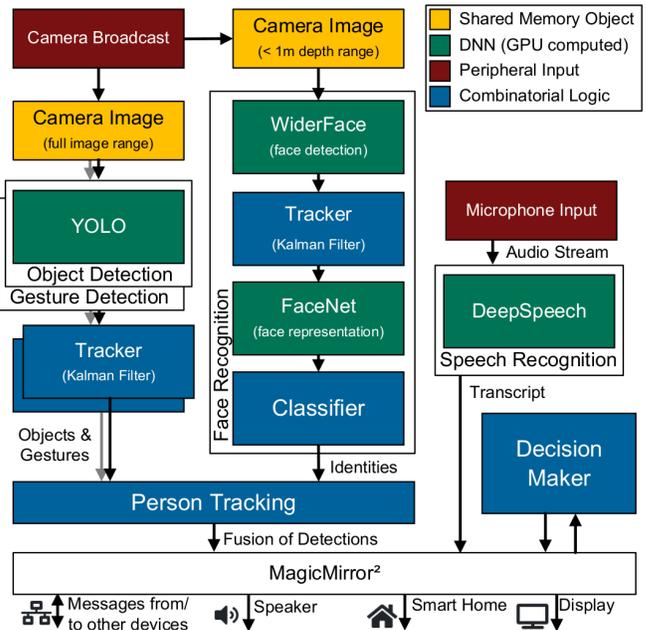


Fig. 5: Architecture of the Smart Mirror Demonstrator

microphone are used as input devices and four different neural networks are used to detect gestures, faces, objects and speech to interact with people. At the actual development state, the demonstrator uses 55 Watts of power which should massively be reduced using heterogeneous hardware and optimising the existing deep learning models.

The distribution of data to the cloud is not desirable because of privacy concerns of the residents. Therefore, all sensing and interaction must be done on-site in realtime.

## VI. SUMMARY

VEDLIoT addresses the new challenges on bringing Deep Learning to embedded devices with limited computing performance and low-power budgets. The project partners contribute the concepts of optimizing the hardware components with additional accelerators and a powerful middleware to ease the programming, test and deployment of neural networks. The concepts are tested driven by the challenging use cases in key industry sectors like automotive, automation, and smart home. In addition, an open call for cascaded funding is foreseen to explore new opportunities by extending the application of the VEDLIoT platform to a more extensive set of new and relevant use cases. Typical open call projects leverage VEDLIoT technologies for their own AI-related IoT use case, thereby broaden the VEDLIoT use-case basis and making the overall concept more robust. The envisaged run-time of the satellite projects will be in the range of 9 – 12 months, with an average funding of up to 120,000 € (including 25 % indirect costs), at a funding/reimbursement rate of 70 %.

More detailed information, including available VEDLIoT technology and the application procedure, is scheduled to be published in early 2022.

## REFERENCES

- [1] René Griessl, Meysam Peykanu, Jens Hagemeyer, et al. A scalable server architecture for next-generation heterogeneous compute clusters. In *2014 12th IEEE International Conference on Embedded and Ubiquitous Computing*, pages 146–153. IEEE, 2014.
- [2] B. Salami, K. Parasyris, J. Hagemeyer, R. Griessl, et al. Legato: Low-energy, secure, and resilient toolset for heterogeneous computing. In *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 169–174, 2020.
- [3] Ariel Oleksiak, Michal Kierzynka, Wojciech Piatek, et al. M2DC—Modular Microserver DataCentre with heterogeneous hardware. *Microprocessors and Microsystems*, 52:117–130, 2017.
- [4] Cong Hao and Deming Chen. Deep neural network model and fpga accelerator co-design: Opportunities and challenges. In *2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, pages 1–4, 2018.
- [5] Renode. <https://renode.io>.
- [6] Karol Gugala, Grzegorz Latosinski, Elaheh Malekzadeh, et al. D 6.1 Report on existing hardware and software interfaces for DL and compilers, July 2021. Project deliverable (under review).
- [7] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.
- [8] Mingxing Tan and Quoc V Le. Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298*, 2021.
- [9] Onnx: Open neural network exchange. <https://onnx.ai/>. Accessed: 2020-11-10.
- [10] Kenning. <https://antmicro.github.io/kenning/index.html>.
- [11] International Organization for Standardization. *ISO / IEC / IEEE 42010:2012: Systems and software engineering — Architecture description*. Swedish Standards Institute, Stockholm, swedish standard edition, 2012.
- [12] IEEE SA Board of Governors/Corporate Advisory Group (BoG/CAG). *IEEE Std 2413: Architectural Framework for the Internet of Things (IOT)*. IEEE Computer Society, 2019.
- [13] Bashar Nuseibeh. Weaving Together Requirements and Architectures. *Computer*, 34(3):115–119, 2001.
- [14] Anitha Murugesan, Sanjai Rayadurgam, and Mats Heimdahl. Requirements reference models revisited: Accommodating hierarchy in system design. *Proceedings of the IEEE International Conference on Requirements Engineering*, 2019-September:177–186, 2019.
- [15] William S Davis and David C Yen. General systems design principles. In *The Information System Consultant's Handbook*, pages 577–584. CRC Press, 2019.
- [16] A. Casimiro, J. Rufino, et al. A kernel-based architecture for safe cooperative vehicular functions. In *Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on*, pages 228–237, June 2014.
- [17] Jämes Ménétrey, Marcelo Pasin, Pascal Felber, and Valerio Schiavoni. Twine: An embedded trusted runtime for webassembly. In *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*, pages 205–216. IEEE, 2021.
- [18] European Commission. The next generation Internet of Things, 2021. <https://digital-strategy.ec.europa.eu/en/policies/next-generation-internet-things>, Last accessed 8 November 2021.
- [19] International Electrotechnical Commission. IEC 62606: General requirements for arc fault detection devices. Standard, International Electrotechnical Commission, 2013.