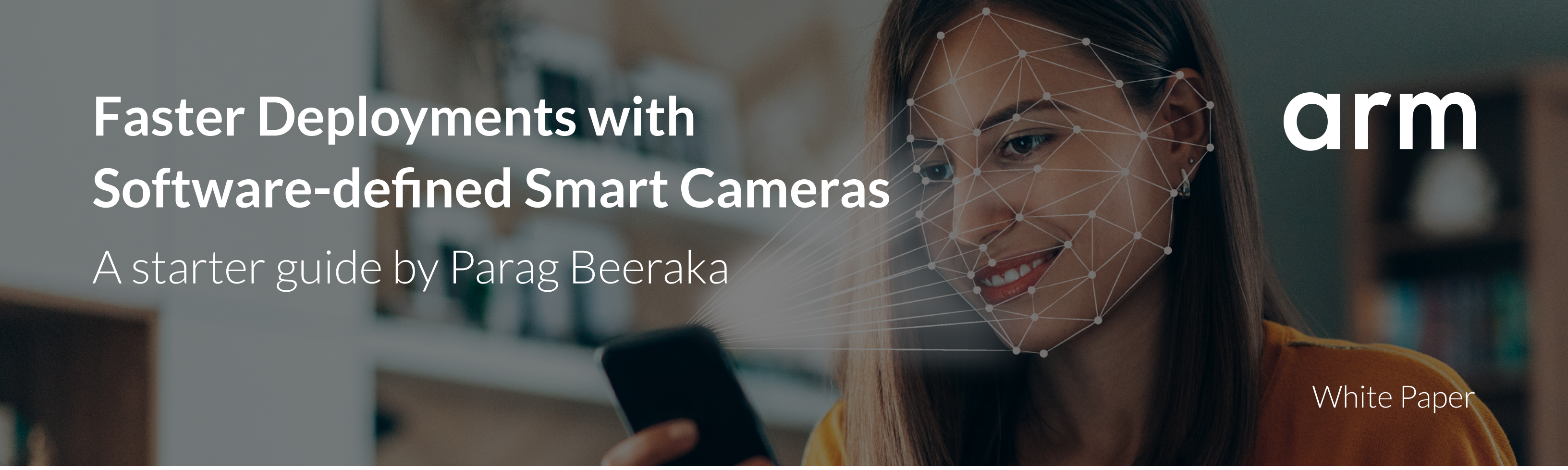


Faster Deployments with Software-defined Smart Cameras

A starter guide by Parag Beeraka

White Paper

Smart cameras shipments are set to grow exponentially, with deployments in areas beyond the traditional surveillance market, from smart homes and buildings, to manufacturing facilities and smart cities. Advanced features are becoming increasingly common, including the ability to locally run complex machine learning (ML) models, provision for over the air updates, and robust security mechanisms. The functionality of a smart camera is no longer defined by the specific hardware it ships with – it is now defined in software and can instead be referred to as a Software-Defined Smart Camera (SDC). This whitepaper will provide an overview of the key technologies that underpin the SDC, and some key considerations when architecting one.

Introduction

Security cameras have been a common fixture in towns and cities for decades. Traditional Closed-Circuit Television (CCTV) cameras record analogue signals directly to a tape or the signal is converted to digital and stored on a Digital Video Recorder (DVR). These cameras typically record continuously and the footage they capture is stored until the storage device fills up, at which point older footage is either deleted or archived.

More recent security cameras are entirely digital making storage of footage more straightforward. Some devices store footage locally, some offload it to local network storage, and others upload it to the cloud. To reduce the storage requirements to store a continual stream of high-quality footage, some cameras offer motion detection functionality. Footage is captured on a rolling basis but is only stored permanently when an associated motion event is captured. Motion detection is one of the most primitive 'smart' functionalities of a security camera deployment.

A camera network could utilize a 'dumb' camera, merely outputting a video signal with processing happening on a smart gateway. Increasingly popular, however, are cameras with processing capabilities built in – a smart camera.

Nowadays smart cameras are deployed in huge quantities in the world and the market is growing rapidly. The video surveillance market is predicted to reach $44 billion in 2025, with a CAGR of 13% [1]. As the market has evolved, a new concept has arisen – the Software-Defined Smart Camera (SDC). This paper will discuss the concept of the SDC and the technologies that enable it.

From Recording Devices to Smart Cameras

Modern day life is considerably more complex than when CCTV cameras were first introduced – cities have exploded in population and security threats are increasingly complex. Machine Learning, a subset of AI, has allowed for dramatic improvements in classification and clustering algorithms, making applications such as object tracking, motion prediction and facial recognition widely deployed. Cameras that integrate this 'smart' functionality to extract application specific information from captured footage are appropriately referred to as smart cameras.

The shift to edge compute is a well-known paradigm, and the uptake of smart cameras forms part of this shift. Edge computing is a distributed model of computing, whereby intelligence is shifted to the edge of the network in contrast to traditional computing, whereby most processing takes place on centralized servers. Key drivers of this trend are the need to improve data privacy, latency, and redundancy, and to reduce bandwidth requirements.

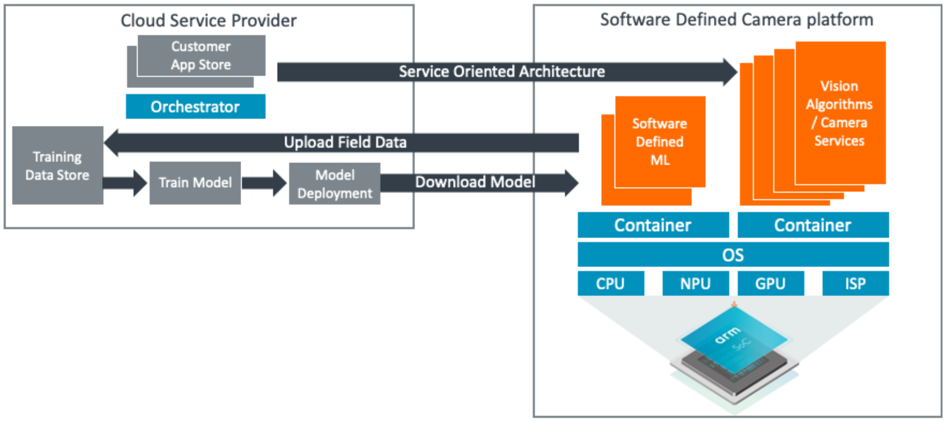
Smart cameras improve data privacy by storing data locally - as opposed to sending footage of individuals to a network video recorder (NVR), a smart camera may simply send over a count of how many people are observed, or a video anonymizing faces and other private information. Data access can therefore be better controlled thus improving security. Offloading compute to the cloud can introduce substantial latency and bottleneck in an application – smart cameras mitigate this by cutting out or minimizing this step. Finally, distributed systems are inherently more redundant – each camera has its own resources, whereas in a centralized system loss of a single server could take the entire network down.

Amazon Go, a chain of convenience stores operated by Amazon, have demonstrated how smart cameras have the potential to completely transform the retail experience [2]. Amazon Go stores are entirely 'cashierless'. Customers simply register a payment method with Amazon in advance, validate their identity upon entry and walk around the store choosing the items they want. A complex array of smart cameras tracks individuals and synchronizes with sensors on the shelves to identify which products have been selected. Once the customer has selected their items, they can simply leave the shop without any interaction. The system accounts for which items have been taken and bills the customer appropriately. Whilst this specific scenario carries out processing with a complex infrastructure that includes advanced gateways and sensors in addition to the camera systems, it clearly demonstrates how smart cameras can form the foundation of innovative future IoT deployments.

The DNA of a Software-Defined Camera

'Software-Defined Smart Camera' (SDC) is a recent term reflecting the evolution of the smart camera to meet modern use cases such as asset tracking, facial recognition, and traffic analysis to name a few. The capabilities of the SDC are defined in the software it is running as opposed to the specific hardware and software which the device ships with. The SDC is designed to be upgradable, meaning new algorithms and features can be added as they become available over time. This is made possible with futureproofed hardware and the provision for over-the-air (OTA) updates. The following sections will go into detail about the key features that enable the SDC.

Fig. 1.
Edge computing architecture of a SDC



1.1 Heterogeneous compute

The complex and varied applications that SDCs are required to run necessitate the use of heterogeneous compute hardware. The term heterogeneous computing refers to systems that utilize more than one type of processing. CPUs are typically the primary processing element in a heterogenous system, handling simple integer workloads. Modern CPUs can implement an additional SIMD single instruction multiple data (SIMD) engine to accelerate multimedia and digital signal processing (DSP) workloads, such as Arm's Neon extensions.

For highly power constrained use cases the Cortex-A35 offers ultra-high-efficiency 64-bit compute, with support for Neon SIMD instructions. For higher performance use cases, the Cortex-A55 CPU offers significantly increased integer performance coupled with an advanced SIMD engine, whilst maintaining class leading efficiency. The Cortex-A55 Neon SIMD engine features dedicated instructions for advanced computing such as dot product, which offer an effective way to run ML models and DSP workloads without having to turn to dedicated hardware. For the most demanding smart camera use cases, the Cortex-A78 pushes the performance envelope further, and can be integrated as a big core in a DynamIQ cluster, allowing for a CPU solution that combines Arm's most performant and most efficient cores. By offering dedicated pipelines for non-integer workloads, such as Neon and FPU, Arm Cortex CPUs are highly capable host processors for SDCs, either standalone, or to offer power efficient compute when the use of dedicated hardware accelerators is not required.

The new Armv9-A architecture introduces a range of features designed to further improve ML capabilities, available in the latest Armv9 CPUs, Cortex-A510 and Cortex-A710. SVE2, new in Armv9-A, builds upon SVE, adding a vector-width-agnostic version of the Neon instructions in most of the integer Digital Signal Processing (DSP) and media processing functionality. These DSP and media functionality are particularly relevant to smart cameras. Armv9-A also includes the General Matrix Multiply (GEMM) feature, which adds new dedicated instructions to accelerate matrix operations, greatly reducing the number of memory accesses required. BFloat16 has recently emerged as a popular number format for high performance NN processing – Armv9-A provides support for BFloat16 in Neon and SVE.

Whilst most workloads can run on the CPU, for the most demanding workloads this is not always efficient or possible. For these cases, a dedicated accelerator can help to offload computing tasks and free up CPU cycles for general tasks. There are many examples of accelerators used in a heterogeneous hardware solution. Higher end smart cameras often contain a GPU to handle graphics manipulations, for instance for pre-processing and post-processing of videos within a computer vision (CV) workload. The GPU can also be used to accelerate any highly parallel workloads with frameworks like OpenCL. Arm offers a broad selection of GPUs in the Mali lineup, including Mali-G310 for mid-range smart cameras, up to Mali-G710 for high end systems. The Mali lineup also includes image signal processors (ISP), a crucial element of a camera SoC. The Mali-C52 ISP supports real-time processing of 4k video at 60FPS as well as 4:1 HDR stitching for improved dynamic range.

Many SDC applications leverage ML workloads extensively, which can require significant compute resources. The performance requirements of an application can vary substantially based on the complexity of the underlying ML model. Highly optimized use cases can have a complexity of less than 1 GOP/s, whereas high end use cases can have complexities north of 20 TOP/s, representing orders of magnitude difference. Performance requirements are also impacted by other parameters such as the number of classes that are required, image resolution, frame rate and required level of accuracy. As a result, a device must have the correct hardware for the desired use case. High-end applications will require dedicated resources to run properly, whilst simpler applications should take advantage of lower performance requirements and therefore select lower cost, more efficient hardware.

The Neural Processing Unit (NPU) is a hardware accelerator specifically designed for ML, such as the Arm Ethos family of NPUs. NPUs typically offer the best AI performance and operate with high efficiency. Multiple Ethos NPUs can be combined to meet the most demanding ML workloads.

To summarize, heterogeneous compute hardware can take many forms. The complex use cases of SDCs, particularly driven by ML workloads, explains why heterogenous compute is such a critical piece of the SDC.

1.2 Microservice-based architecture

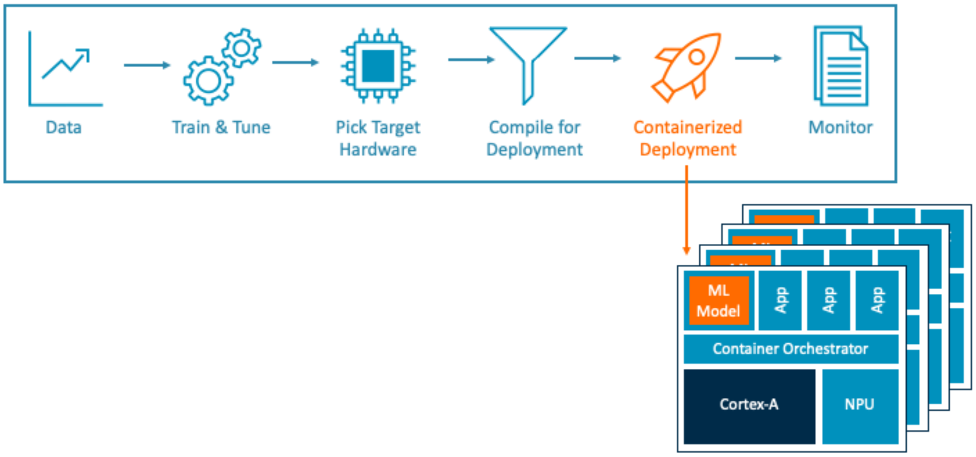
Support for continuous software updates is crucial to the SDC concept. In a world with rapidly changing technologies, the ability to continuously update device functionality is required to successfully decouple the software solution from the underlying hardware, meaning the camera functionality can evolve with new technology developments.

Updates are required to be performed securely and over-the-air (OTA), as the cost to physically access each camera would be prohibitive. Typical updates most likely involve advanced functionalities of the devices, for instance the deployment of new and improved ML models or CV algorithms. At the same time, OTA updates of the firmware and OSes along with security updates will also be required. All these updates will need to be performed securely - it is therefore critical that updates are confirmed to be genuine before they are installed.

As updates are deployed increasingly frequently, the ability to run cloud-native applications will enable developers to focus on the specific value-added applications for the smart camera. Thanks to containers, software applications can be packaged in a self-contained unit that can run in different computing environments reliably. The container is built to include everything that is needed to ensure that the application runs successfully, with elements like libraries, tools, and any other software dependencies. There are various popular containers for embedded devices such as Docker, and container orchestrators, such as k3s and Kubernetes.

The ability to easily perform OTA updates, paired with cloud-native and containerized software, enables the transition to a microservice deployment model, where applications split into clearly defined functionalities called microservices. Each one of those, can be managed and updated independently.

Fig. 2.
Using cloud-native principles and OTA updates to deploy services



1.3 Robust software ecosystem

To build a state-of-the-art SDC, hardware is only half of the story. To support complex SoCs with various disparate processing elements as well as continuous training and delivery of CV and ML models, a strong software ecosystem is a prerequisite. Arm Compute Library provides a collection of low-level ML functions optimized for Arm hardware, allowing developers to get the maximum performance out of the available hardware.

For edge devices, compiling ML models for specific hardware targets is often a prudent approach, as the code can be optimised and run with less overhead. Models can be built and trained in TensorFlow, for instance, and then run through a ML compiler, such as TVM. TVM is an open governance framework which has support for a broad set of backends, both in terms of computing elements such as Cortex-A and Cortex-M processors, and with other frameworks such as OpenCL. The compiled model can then be deployed to the SDC in a container, where it can be continuously trained and monitored, as discussed in the previous section. This is useful for SDCs running on limited hardware or within a small power envelope, as code can be further optimised to reduce code size and improve efficiency.

An alternative means of deploying ML models to a SDC is to use ArmNN. ArmNN is an open-source software framework to bridge the gap between the ML software libraries favoured by developers, such as TensorFlow and PyTorch, and the underlying processing hardware. It provides a common software interface that allows developers to easily port their ML models across different hardware, reducing the need for processor specific optimizations. Developers can realise up to 9.2x increases in neural network performance with ArmNN, based on Arm internal testing.

1.4 Strong hardware and software security

In many use cases the SDC has a critical operational role. Take for example a SDC deployment used to monitor traffic and adjust signals to minimize congestion – a hacker could potentially commandeer the SDC to control the flow of traffic, and worst case, interfere with traffic signals. A compromised SDC in this scenario would represent a significant safety hazard.

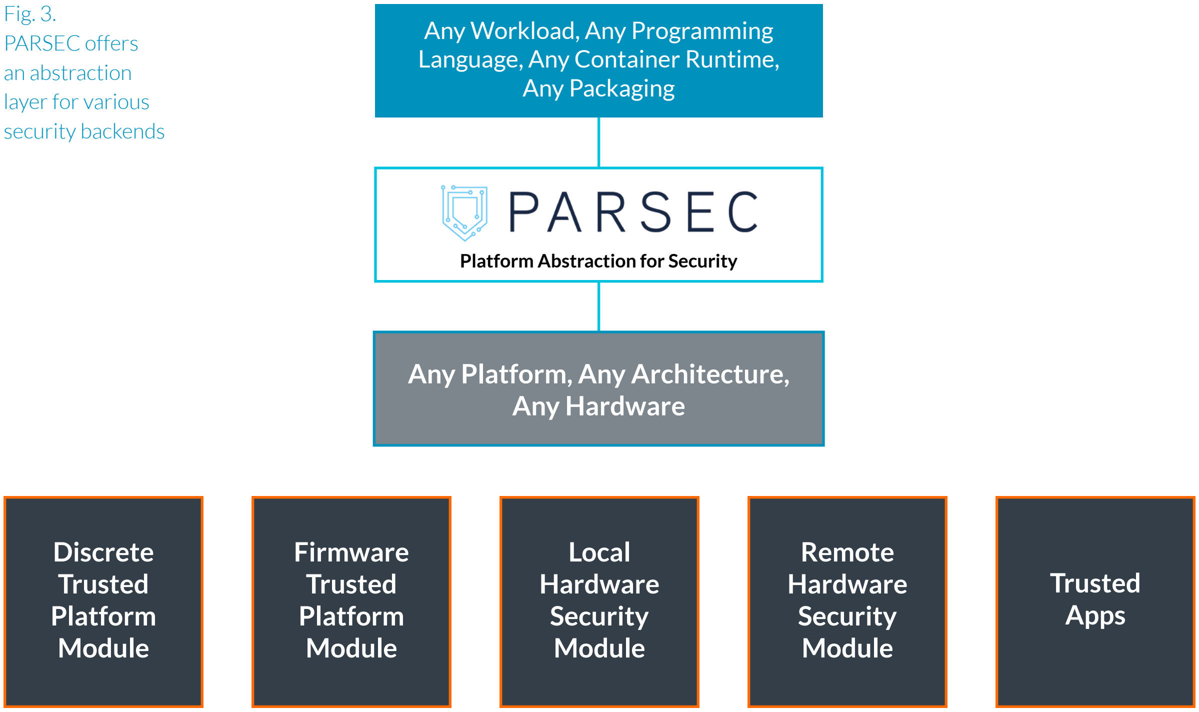
Another, perhaps more common, security concern is unauthorized access to video feeds. Recent data breaches have seen hackers access vast quantities of confidential and commercially sensitive data. As SDCs become increasingly widespread and are entrusted with critical tasks, it is crucial that device security is prioritized.

Hardware security for the IoT has typically been a fragmented affair with numerous competing standards. PSA Certified, introduced in 2019, brings a hardware agnostic, standardized framework to the fragmented and fast-moving world of IoT security.
It offers three levels of certification, allowing vendors to build a chip with an appropriate level of security for the use case. Security certifications such as PSA Certified are important for SDC adoption because they provide an independently verified stamp of approval, reassuring the end user that the camera they are using employs the necessary components to achieve a strong Root of Trust (RoT), which is the secure foundation of trust on which the rest of the system depends.

Arm offers the CryptoCell family of Security IP to provide a range of security functions to the SoC, helping to build a strong hardware RoT. CryptoCell-712 is targeted at high performance Cortex-A systems and provides several security functions to the system. Cryptocell-712 offers key generation using the integrated True Random Number Generator (TRNG), code protection through verification of software images at boot and OTA updates, credential and lifecycle management amongst other features. This is in addition to a dedicated high performance crypto engine. Further hardware security is offered by Arm TrustZone technology, supported in all ArmV8-A cores. TrustZone effectively divides the CPU into two worlds, secure, and non-secure. This allows for secure code to be isolated from non-secure access. TrustZone can also be implemented at the system level, providing access control to secure peripherals and secure memory regions.

A hardware RoT can be implemented in numerous ways, and the functionalities of the RoT are often accessed with an ad-hoc solution, which can make it difficult to deploy software across multiple platforms [3]. Parsec (Platform Abstraction for SECurity Service) is a solution to address this problem. This project forms part of the CNCF (Cloud Native Computing Foundation) and provides a common API for hardware security and cryptographic services in a platform agnostic way. Parsec provides a microservice for brokering access to platform specific hardware security. This allows cloud-native workloads written in any programming language, deployed in any container runtime or packaging to access platform security services while remaining decoupled from physical platform details. Parsec can be integrated with various backends such as a Trusted Platform Module (TPM), Hardware Security Modules (HSM) and Trusted Applications.

Fig. 3.
PARSEC offers an abstraction layer for various security backends


Any Workload, Any Programming Language, Any Container Runtime, Any Packaging
PARSEC
Platform Abstraction for Security
Any Platform, Any Architecture, Any Hardware
Discrete Trusted Platform Module
Firmware Trusted Platform Module
Local Hardware Security Module
Remote Hardware Security Module
Trusted Apps

In addition to hardware security, it is important that communications across the broader network are secured. Using strong encryption is important as it acts as a last line of defence in case data is intercepted. Data streams must be encrypted to a high standard, for instance using the AES256 encryption algorithm which is virtually impenetrable by brute force attacks [4].

Encryption algorithms can be accelerated in hardware, for instance by using the cryptography extensions available in many Cortex-A processors, or by using dedicated hardware blocks such as CryptoCell-712. Crypto algorithms running on CryptoCell-712 run substantially faster and use less power when compared to running on the CPU itself, even with crypto extensions. For systems that require ciphers specific to China, CryptoCell-713 supports ChinaDRM's enhanced profile in addition to the full CryptoCell-712 feature set.

In the coming years SDCs will benefit from the security features introduced as part of the Armv9-A architecture. New security features include Memory Tagging Extensions (MTE), which help mitigate memory corruption attacks which underpin many famous data security breaches, and Branch Target Indicators (BTI)/ Pointer Authentication to help resist Return-Oriented Programming (ROP) and Jump-Oriented Programming (JOP) attacks.

1.5 Hardware and Software Standardization

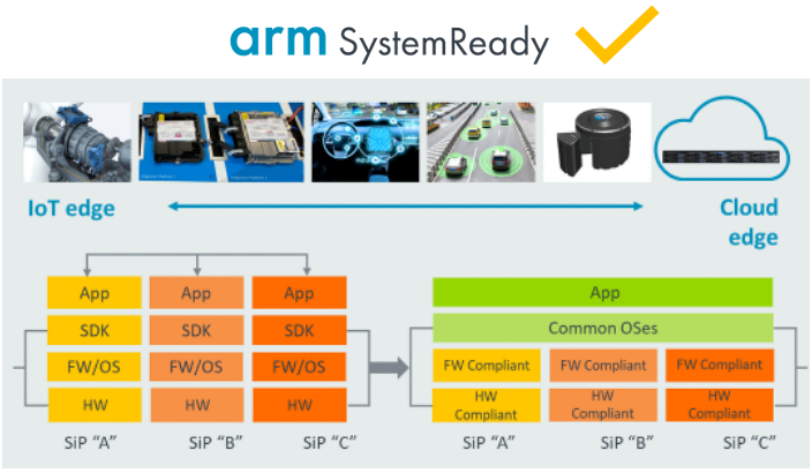


Fig. 4. The Arm SystemReady foundation for cloud-native software

SDCs deployments are often large in scale and comprise a variety of different hardware. An appropriate amount of standardization is key to enabling frictionless deployment of firmware and software. Arm SystemReady is a foundational certification program aimed at promoting the required hardware standardization. Focused on IoT endpoints, SystemReady IoT Ready (IR) ensures platform interoperability with embedded Linux and other embedded operating systems. System Ready IR is aimed at devices that support operating systems that only require a subset of UEFI interfaces in firmware, defined by the Embedded Base Boot Requirements (EBBR). The minimal set of processor and system architecture requirements to enable a standard OS to boot are provided by the Base System Architecture (BSA). SystemReady provides a foundation for a cloud-native software stack, meaning platforms built to the SystemReady specification can run standard operating systems, hypervisors, and containers [5]. Hardware standardization and abstraction are fundamental to building a sustainable ecosystem around the SDC and will make it easier to deploy SDCs successfully in new, cutting-edge applications.

SDC deployments often integrate a variety of protocols in the broader infrastructure, thus requiring a standardized communication protocol. The Open Network Video Interface Forum (ONVIF) offers an open standard network interface which dictates how security IP products communicate with one another, covering functions from discovery and configuration through to event handling. By building SDCs to ONVIF specifications, OEMs can reduce the barriers to deployment, and futureproof the infrastructure for expansion.

Conclusions

Nowadays cameras serve areas well beyond the scope of the traditional surveillance market. Modern deployments are incredibly complex, with sophisticated applications deployed across numerous endpoints. SDCs are perfectly positioned to accelerate these deployments, utilizing cloud-native principles to allow for simple management and OTA upgrades of software.

The foundation of a SDC is futureproofed, heterogeneous compute hardware. Arm offers a comprehensive range of hardware IP and software to build a powerful, efficient, and secure SDC SoC. A strong security infrastructure can be achieved by following PSA Certified guidelines to implement appropriate security features, and by using Parsec to provide a common API to hardware security and cryptographic services in a platform agnostic fashion. SystemReady allows for portability across diverse Arm-based platforms, with standard off-the-shelf OSes and hypervisors supported in a consistent manner. A microservice based architecture decouples the hardware from the software, meaning new functionalities can be added via OTA updates. Standard ML frameworks make it simpler to build and deploy ML models at the edge, and with devices being securely connected to the cloud, models can be improved and updated over time.

Built by combining powerful, futureproofed hardware with a cloud-native software stack for simplified deployments and OTA upgrades, the SDC represents a key component of the most exciting future IoT deployments.

References

[1] M. Jude, "Worldwide Video Surveillance Camera Forecast, 2020–2025", IDC, 2020

[2] H. Huang, "How Amazon Go (probably) makes 'just walk out' groceries a reality", Ars Technica, 2017.

[3] M. Riga, S. V, Kuriakose "Accelerating Security for Linux IoT Endpoints", Embedded World 2021

[4] "Understanding AES 256 Encryption", SolarWindsMSP, 2019

[5] A. Rose, "Arm SystemReady – where software just works across a diverse ecosystem", Community Arm, 2020