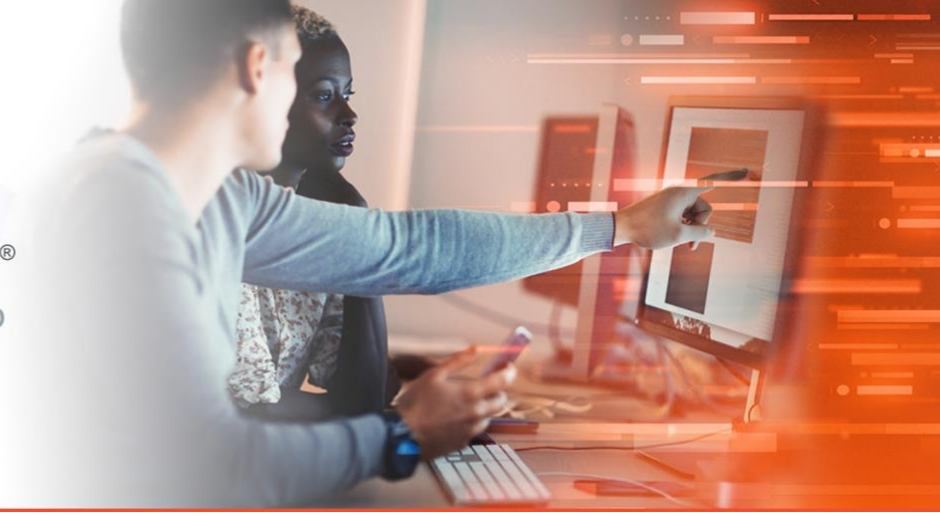
MICROEJ®

SOFTWARE-DEFINED EVERYTHING



HOW SOFTWARE-DEFINED DEVICES ARE CHANGING THE WORLD

As Marc Andreessen put it in 2011, "Software is eating the world." In those five words, the iconic Silicon Valley entrepreneur, Netscape cofounder, and partner at famed VC firm Andreessen Horowitz, was prescient. Since that time, "softwarization" has reached all industries and corners of society and given rise to thousands of software-defined variants, from networks to radio to cars.

Thanks to technology advancements that place supercomputing power in the palm of a small child's hand, software-defined products are inherent in the smallest devices around us. Advancements in semiconductors and compute capabilities as well as the emergence of virtualization and software containers for low-cost, low-power devices can transform any device into a software powerhouse. A collection of fixed-function products in the past is today a network of programmable devices with upgradable features and distributed intelligence.

This software-defined revolution is transforming our homes, our hospitals, and our shop floors – silently changing the world. Here's how that transformation has happened and will continue to advance.

CLOUD NINE FOR THE EDGE

After decades of pulling from the cloud, computing is moving back to the edge. In his A 30,000-Foot View of Edge, Gavin Whitechurch, cofounder of Edge Computing World and COO of Topio Networks, said, "We're going to see much more of a distributed compute model–the market will move to a more distributed compute architecture."

Distributed intelligence materializes with edge computing: it moves more computation and data management away from a central cloud server and closer to the end user, be that human or device, with the help of cloud nodes, gateway servers, and even intelligent edge devices.

Edge computing and distributed intelligence provide many advantages:

- Reducing the cost of data transfers and data storage as most data is treated locally
- Reducing the energy consumption and carbon footprint of the system
- Improving service latency, as the local device is able to react to inputs in real time
- Increasing resilience and availability as devices are more independent of the cloud and less affected by network outages, connection losses, and other adverse events

In addition, as technology advances and edge capabilities improve, B2B and B2C enterprises are better able to decentralize their systems and take advantage of distributed intelligence. Today, 10 percent of data is created and managed outside the centralized cloud. By 2025, that portion will be 75 percent, according to Gartner.

There are, however, significant design challenges to capturing the edge, and many of them begin with software.

IS IOT PLUG AND PLAY REALLY HERE?

As technology progresses, it makes sense to take computing closer to the edge, to the end device that interfaces with the real world and controls machines and interactions with humans. In Forbes, Bill Curtis from Moor Insights and Strategy states that **"The Era Of IoT Plug-And-Play Begins,"** and notes that devices are now sufficiently powerful to make edge intelligence effective.

Indeed, microcontrollers (MCU) and microprocessors (MPU) used in everyday electronics are more capable than ever, and sophisticated technologies like AI are now possible on even the smallest device. Multicore GHz microcontrollers, such as those from NXP, are more common, and specialized GPU and accelerators, such as ThinkSilicon's Nema PicoVG GPU or the Arm Ethos55 neural network unit, provide 100 times the performance advantage of older technologies, making these tiny processors as capable as mobile phones from five years ago.

New tools and software ecosystems around device enablement are also developing. The most salient example is the TinyML community, which brings together hundreds of companies and thousands of members committed to delivering machine-learning capabilities to the smallest embedded device at the edge.

There is, however, a significant obstacle to this transition: software. As Bill Curtis highlights, "Developing code on small IoT devices is often like returning to the 1990s or even the 1980s." Some progress needs to happen before edge devices become as agile as the rest of the network.

HARDWARE IS HARD, BUT SOFTWARE DOESN'T NEED TO BE

As Google CEO Sundar Pichai acknowledged during an interview with The Verge, "hardware is hard," and even more so with devices that lack the compute power and sophistication of mobile phones.

Historically, before hardware was programmable, every function was immutable. The design and verification process had to be extremely disciplined and rigorous, because any change or fault detected later in the product's lifecycle was prohibitively expensive to repair – in other words, forget about making changes after the product shipped. Such constraints led to meticulous design methodologies pioneered by heavy-manufacturing industries such as automotive, military equipment, power generation, and spaceship building. The DoD and NASA were good sources of many development methods that migrated to commercial industries, proving their effectiveness for massive, long-term projects and serving us well for most of the twentieth century.

After decades of this regime, the "hardware-defined" approach became common sense in many industries. Yet as firms began building programmable, software-oriented products, many of them failed to seize the opportunity to introduce more flexibility in both product design and development techniques. These firms followed the older processes, and built software as an afterthought: they launched development with a time-consuming specification, created the hardware, and executed the project in a highly planned waterfall manner.

As effective as it was, this process is slow to adapt to rapid changes in market demand and technology. It stifles innovation, as everything must be defined before design begins, and changes are costly to implement during the project. Consequently, functions are fixed and less capable of evolving.

By contrast, consider the common development process for cloud, IT, and mobile applications, where agile development has become the norm. Here, the product and process merge. The product is made up of software and is ever-evolving, ever-changing, and never wholly specified. This nurtures a wave of innovation that would be unattainable following older development processes.

Considering all this, astute observers wondered how they could apply the same agility and development concepts to hardware-based products. Thus was born the idea of "software-defined everything."

SOFTWARE-DEFINED IS THE NEW HARDWARE

The "software-defined" concept originated with software-defined radio in the 1990s. The idea was to build a piece of hardware, the radio transceiver, to be programmable, flexible, and independent from its transistors.

Next, the network and telecom industries began to apply this concept through software-defined networking (SDN). This activity emerged after several standardization bodies considered how they could decouple control and forwarding functions, making them programmable and independent from the underlying hardware to foster flexibility and portability. SDN began in the 2010s and created a market that is growing steadily, fueled by the IoT, 5G, and cloud computing.

In an attempt to break their own traditional lengthy development cycle, automotive manufacturers also adopted the software-defined concept with an eye toward building software-defined vehicles (SDV). Car makers have long realized that, as Deloitte states in its analysis, "the quantity and value of software (including electronic hardware) in a vehicle exceeds that of the mechanical hardware." This trend has been furthered by such innovators as Tesla and Volkswagen CARIAD, which put software front and center in their designs.

The software-defined approach is now trending across all sectors, the latest being industrial and manufacturing. The software-defined factory is the natural evolution of Industry 4.0, the initiative that made all factory equipment instrumented and connected. Now, such industry leaders as Schneider Electric and NXP are teaming up to enable a software-defined factory that will have "an immeasurable impact" in terms of productivity and overall cost savings, according to Ralf Neubert, vice president, Research & Development, Digital Factory, at Schneider Electric.

CLOUD-NATIVE MADE EDGE-COMPLIANT

All of these industries now fully understand the advantages of software-defined principles: starting with software first and adopting a modern development mindset. Everything becomes agile and flexible, from product development to post-release upgrades, and users—not years-old business plans—dictate product evolution.

To simplify the creation of software-defined devices, manufacturers look to reuse technologies proven in the cloud:

- Agile, continuous integration and DevOps processes
- Virtualization, to get more out of hardware platforms
- The adoption of standard platforms such as Linux, Android, and Microsoft for cloud-native development
- Microservices, Docker containers, and Kubernetes orchestration

These elements form the backbone of the software revolution in the cloud, software-defined networking, mobile apps, and IT. There is one catch, however: these cloud-native technologies are typically not applicable to edge devices. They were defined with the cloud in mind, and serve it well, but cannot be repurposed “as is” for much smaller targets.

A more effective approach that solves this problem is this: to start from the device and build a standard platform for the edge that brings virtualization to the smallest targets, offering containers and containerized apps with app orchestration, and brings microservices to the devices.

Such an endeavor was attempted several times by various players. Most recently, MicroEJ successfully achieve this with its MICROEJ VEE virtual execution environment.

NOW HARDWARE CAN FLEX AGILE MUSCLES

Agility is the first benefit that design teams and product owners see when moving to a software-defined approach.

In software-defined products, functions become more independent from their hardware specification, enabling a broader feature set and faster evolution as the functions are much easier to upgrade. By definition, the main product functions are software-driven and portable, able to take advantage of new hardware, and easy to move to different hardware variations.

For a development organization, a software-defined approach reduces risks and costs. It allows the parallelization of hardware and software development, translating into greater new product output and a shorter time to market. It simplifies upgrades and maintenance and enables quick alignment to market needs, extending product reach and lifetime. Typically, a software-defined product will be able to change after shipment, as more usage data can be collected and more use cases defined.

This natural decoupling from hardware also increases silicon chip portability and reduces supply-chain risks.

SOFTWARE-DEFINE YOUR BUSINESS TRANSFORMATION

Software-defined technology has already made a significant positive impact on operations – but its business impact is even more impressive. From a system point of view, software-defined devices fuel the evolution to edge computing by allowing resources on the device to run as distributed resources. This breaks traditional monolithic architectures and ecosystems to support the creation of new value and revenue models.

HYPERSEGMENTATION

Traditional product distribution is based on a limited number of models offered in a catalog. Because software-defined products are highly and easily customizable, it is possible to multiply these variants to fit every customer’s needs. Distributors and third parties can do this customization, addressing the tiniest market niche and offering personalized solutions.

BUILDING BRAND AND CUSTOMER LOYALTY

The easy programmability of software-defined devices provides an opportunity for manufacturers to create a closer bond with their customers. First, they can enforce a consistent branding and user experience on all their devices, making them evolve alongside customers' needs; they can also take more branding risks as the products are easy to configure and change.

Second, manufacturers can enhance customer interaction. Traditionally, once a product ships, the only marketing channels are email and web, with mixed results. By contrast, a software-defined and connected product can itself become a real-world marketing channel, informing customers of new offers and features and keeping those customers engaged.

SERVITIZATION AND RECURRING REVENUE

Device companies have long tried to add recurring revenue to their mix by charging customers with maintenance or services fees after the product is shipped. This creates a predictable and steady high-margin revenue flow.

Now, software-defined devices raise this capability to another level by allowing the addition of software-based services, enabling subscriptions and "as-a-service" business models.

For hardware companies, servitization starts as a complement to the device model, adding several percentage points of high-margin recurrent revenue. In the long run, an outcome-based or pay-per-use model is possible as the service evolves, replacing the current "device plus service" model.

Cloud and software companies can use software-defined devices to push their service model down to the customer touch point and create services as a combination of cloud, edge, and device computing. This opens vast possibilities.

SOFTWARE-DEFINED EVERYTHING

Software-defined everything is coming. Someone, somewhere is already developing a software-defined device for every market that has traditionally been locked-in. That someone recognized the need to transform into an agile, software-based manufacturer to create new revenue streams and to better differentiate and retain customers.

By using cloud-native technologies adapted to the edge through embedded software containers such as MICROEJ VEE, device manufacturers and cloud players are driving digital transformation of their business models and increasing their cloud-to-edge momentum.