KARL

KUBE ACCELERATED REAL TIME LAB

Implementation Scenarios

KARL implementation scenarios

The KARL technology truly has the potential to revolutionize the concept of the workstation as we know it, putting an end to all the problems, challenges, and limitations associated with the physical endpoint model. KARL could indeed pave the way for a DaaS (Desktop as a Service) revolution.

Today, we are used to thinking of the computer as a tool that allows us to do our work, while we are already inclined to view some functionalities as services (according to an "as a service" consumption model), like the software we access to perform our tasks (for example, software consumed in SaaS mode, such as Office 365 or other cloudbased applications). The progressive adoption of "as a service" models in IT has been a growing phenomenon in recent years. The main drivers behind these models are scalability, flexibility, and the transfer of responsibility from the fallible human user to specialized service centers.

A DaaS model transforms the computer itself into a service. This firstly means the possibility of reducing the volume, size, local resources, and energy consumption of the client (terminal) to the extreme (a "Karl instance" computer can even be accessed through a Raspberry Pi costing around €50). Additionally, it transfers the responsibility for data availability to the service center, which remains one of the main issues in the physical workstation model assigned to users (such as device theft or accidental loss).

While it is theoretically possible to implement Desktop as a Service infrastructures using products that existed on the market before the invention of KARL and the release of its software (KARL Platform and KARL OS), it would require an inefficient and costly approach with fewer security advantages: virtual machines.

A DaaS architecture with traditional virtual machines is inefficient compared to a KARL architecture since virtual machines do not have the same scalability and flexibility properties as containers. The inefficiency is related to both the infrastructure complexity in terms of required hardware and software components (and associated licenses) and the distribution of server resources necessary to operate each machine (VM). KARL technology brings the necessary resources to a "minimum moment" (Real-Time), resulting in a much higher efficiency factor.

A clear example of this is storage (literally the disk space used by each virtual machine or hard disk in physical machines). In a KARL infrastructure, storage is minimized, as the entire system component is contained within a master disk (including all the programs used by the served user base): one is enough to generate multiple replicas (workstations). As for the storage component containing dynamic data, or the files worked on by users, this is stored on persistent units.

A DaaS architecture with traditional virtual machines is costly, given the various components and licenses required to keep it operational. Additionally, the limited ability to scale the server infrastructure resources hosting it also negatively impacts costs. Finally, the complexity requires substantial personnel for infrastructure administration and user support.

Although a DaaS architecture with traditional virtual machines has a positive impact on security by allowing users to connect to workspaces located on a third-party environment, it does not entirely eliminate certain issues, such as those mentioned in the first chapter (for example, a user accidentally activating an exploit or a buggy system software update).

In contrast, while virtual machines always exist in a third-party system, KARL instances give users the impression of connecting to a genuine remote computer with all the necessary functionality but exist solely as a service. These are two different logical planes, and this makes all the difference. A virtual machine always exists, even when it is turned off, both regarding allocated resources (tying back to the discussion on efficiency and costs) and as a potential attack surface.

A KARL instance (workstation) only exists while the user is using it. When not in use, all that remains is a pattern, a model that contains instructions to regenerate an identical instance with the same user profile, programs, settings, and access authorization to certain data on shared storage.

Thanks to the container entity, KARL technology and software have added another layer. Through an internal probe system, a KARL infrastructure can detect any anomaly in the instance (whether an exploit, a registry error, or any other abnormal process capable of compromising its functionality) and self-defend by "exploding." In essence, if the user clicks on a malicious link or otherwise encounters a virus, the instance (virtual workstation) self-destructs. A new instance is regenerated based on the mother pattern, remaining untouched and therefore clean. This is something that does not occur in infrastructures with traditional virtual machines.

For these reasons, we believe that KARL and its product line are conducive to broader adoption of the DaaS service model in companies and beyond. Making a fully functional and secure computer accessible through such a flexible system opens the door to multiple use scenarios. We believe that it is possible to envision new and different scenarios that are not yet apparent.

- The first scenario is also the most immediate to imagine: the application of a DaaS model in all organizations that use a large number of workstations, such as call centers, banks, insurance companies, service design and delivery firms, public administrations, law enforcement agencies, and healthcare facilities.

Implementing a KARL infrastructure in the cloud through KARL Platform or in private data centers through KARL OS enables organizations to leverage all the advantages we've outlined, raising their IT standard in terms of efficiency, sustainability, and security.

Additionally, we wish to emphasize other aspects and advantages:

- Elimination of the physical workstation obsolescence issue, with all the associated sustainability and waste disposal considerations;
- Facilitation of remote work or managing geographically dispersed client infrastructures (e.g., companies with many branches);
- Drastic reduction in onboarding times, eliminating the need to configure physical or virtual machines;
- Streamlining hardware and software asset management, which now remains under the control of the service center;
- Reduction in user troubleshooting needs, which consequently decreases the number of helpdesk staff;
- Reduced downtime for maintenance and problem resolution activities.

- The second scenario relates to industry. Automation has shaped industrial production systems to extensively use interlinked machines, some even connected to the internet. Many of these machines are controlled or otherwise linked to computers. Incorporating these systems into a KARL architecture could simplify centralized management of the various machines and ensure a higher level of security for the infrastructure.

- The third scenario involves retail chains or franchises, which could manage their diverse, distributed endpoints across different locations conveniently and securely through a KARL infrastructure.

- The fourth scenario concerns software developers in any field. In Italy and likely elsewhere, many companies still use ERP and other software products configured on a client-server model within local infrastructure, often resistant to change. Software providers are thus compelled to maintain and continuously update these systems to address threats, vulnerabilities, and security standards while simultaneously developing cloud versions of these applications to meet the demand from customers who prefer this service model. In this case, a KARL infrastructure provides a third option, offering a solution that can satisfy both customers and a new option for distribution.

- In the fifth scenario, we imagine the consumer market. DaaS could become a service model for individuals or home computers. Once DaaS adoption becomes simpler, Internet Service Providers could consider offering "Thin clients" (like the previously mentioned Raspberry Pi devices) to households, allowing them to use remote computers as a service with profiles and usage restrictions for minors.

- Finally, the sixth scenario takes a social perspective. Technology and the ability to use it are increasingly becoming prerequisites for access to knowledge. However, technology costs are not always accessible to everyone, including many in Western countries and even more so in the developing world. A DaaS service model could provide governments, NGOs, and nonprofits with a tool to launch initiatives that promote equal access to technology

For more on our products take contact with one of our valuable partners.
We'll help you to find the right provider if you mail us at
info@karl-technology.com