

CSP networks are traditionally populated with a large and increasing variety of proprietary hardware appliances. They are called bare-metal solutions because they use dedicated hardware instead of a pool of hardware resources shared by other solutions. The operating system is installed directly on to the “bare metal” of the physical server without any layer in between, e.g., for virtualization.

Before virtualization, each value-added service often required yet another bare-metal solution with its own dedicated hardware components and the CSP needed the rack space and power to accommodate them. Many CSPs avoided launching new services due to the capital investment challenges and the lack of the necessary skills to design, integrate and operate increasingly complex, heterogeneous combinations of bare-metal VAS solutions. Moreover, the underlying solution-specific hardware often reached end of life, requiring much of the procure, design, integrate, and deploy cycle to be repeated in ever shorter sequences, and with little or no revenue benefit. This also inhibited the implementation of new revenue-earning services and constrained innovation.

1

You code the application for your service or product from scratch.

PROS You write your own code and have full control of everything.

CONS You need highly skilled programmers.

You need different types of developers covering backend, frontend, DBs, etc.

You have to pick the right programming language for each application.

The developed service is a black box for your business people.

You have to monitor all technology changes.

You have to deal with all security aspects for each service application.

2

You buy a standardized application from a software company.

PROS You select one of a variety of applications from different competing vendors.

Usually these applications are tried-and-true as they are in the field at other telecoms companies.

CONS You get a silo application which needs to be integrated into your OSS/BSS.

The service application is a black box which can only be changed via a CR process with the original vendor. This can be time-consuming and expensive.

You have no common CI throughout your application portfolio.

There is little or no reusability and also no interworking with other applications in your portfolio.

3

You have an integrator or software company develop the service just for you.

PROS You get what you specify, tailored to exactly what you need.

CONS Very time consuming as you have to write a detailed specification.

Very expensive as everything will be done just for you.

You take on a lot of responsibility: will the application really do what it should do? Will it be scalable?

You still need to do the whole integration into your OSS/BSS.

You get a black box which can only be changed via the original integrator.

There is no reusability and also no interworking with other applications in your portfolio.

4

You resell a service or product provided in the cloud of the OTT provider.

PROS You resell an established product from a known vendor. You have only limited costs for network integration to the vendor’s cloud, e.g. via an SBC.

CONS Your gross margin is usually 80% lower than it would be with a service implemented in your own network.

You have difficulty differentiating your offering as the third-party service you’re reselling is usually available from your competitors.

You have little or no means of tailoring the service for important corporates and public sector customers.

It is difficult or even impossible to combine and orchestrate the service with other services provided in your network.

There is no reusability and also no interworking with other applications in your portfolio.

5

You implement the service or product on a T-LCAP integrated into your network.

PROS You can extend your agile development squads to include business and technical citizen developers.

Many application enhancements and changes can be made without line coding and without professional developers, allowing you to adapt and enhance the service continuously without costs for third-party vendors.

Automation and the participation of business and technical citizen developers both reduce the time to market for new applications and adaptations as well as improve your market responsiveness.

You have an open application with reusable components to simplify and accelerate integration both into your backend as well as into third-party services.

CONS With this option, you’re dependent on your own business savvy as well as on your T-LCAP vendor for support across the platform.