

Why I prefer dining out



John Olson, VP, Technical Services & Engineering, NMSaaS Inc., has nearly 25 years experience as a successful Technology Professional with emphasis on IT Management, Automation, Network & Application Performance, and Security.

About NMSaaS

NMSaaS is the leader in comprehensive Cloud Based Network Management delivering the full benefits of StableNet® as a Service for the SME, Government, Educational and MSP markets in North America.

I'm writing this article in August, towards the end of summer, and where I live, in the northeast part of the United States, summertime always reminds me of a few things: It reminds me of family holiday trips, warm late evenings spent with friends, and it also reminds me of impromptu simple meals (because who wants to cook in this heat?). One meal I seem to eat more often in the summer than at any other time is pizza. And the funny thing is, pizza reminds me of software. I know that may sound odd, but it's true! Any time I eat pizza, whether it is homemade, delivery or in a restaurant, I am reminded of a funny way to think about how the hottest trend in application delivery (SaaS) can be directly related to how pizza gets delivered to my mouth.

Just take a look at Figure 1; I think it very easily explains how software (I mean pizza) can be delivered many different ways. *Hint - I prefer dining out.*

This graphic originally came from an article written by Albert Barron (IBM). And I think it really helps to

clarify some potentially confusing terms into something everyone can relate to. In this article, I want to focus on the two silos at each end. The traditional on-premise (legacy) systems and the Software as a Service (SaaS) option. I choose to focus on these, because in my role as a technologist within the realm of Network Management and Services Assurance they are the main two options that customers are offered as to how they can purchase and consume solutions. I spend quite a bit of time discussing with customers and industry leaders the pros and cons for each, and therefore when it makes sense for an organization to go one way or the other. I believe that in many cases customers who have historically always implemented IT management applications in the traditional on-premise model, see a tremendous advantage and operational savings when they move to a SaaS model. Let's explore this concept in more detail.

When looking at the "guts" or internal architecture of ENMS solutions, they tend to resemble each other. Individual components may vary, but in the end, they all pretty much look like Figure 2, over the page.

This may look complex, but really it involves 4 major components.

1. Something that speaks to your devices and collects information like inventory data, performance data, etc. This data may be collected via a wide variety of both active

Figure 1.

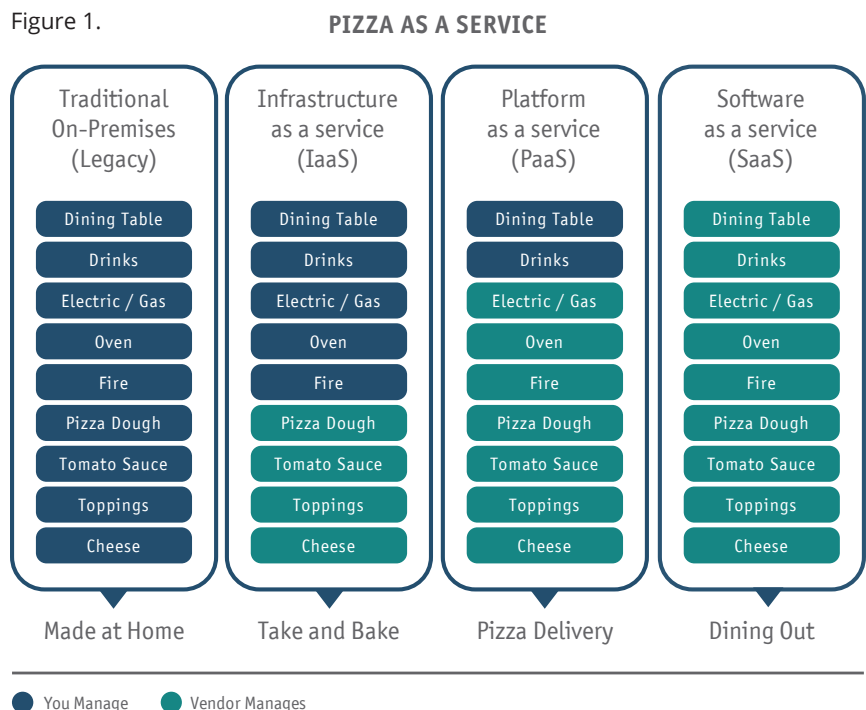
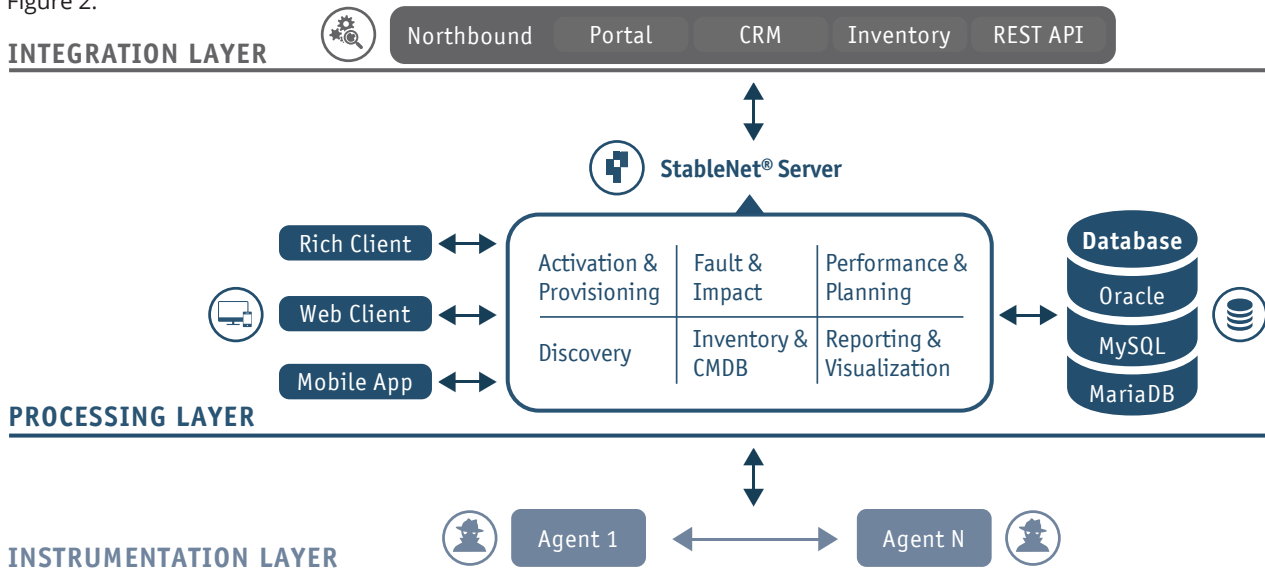


Figure 2.



- and passive technologies like: SNMP, ICMP, DNS, NetFlow, WMI ... and many more.
2. Something that collects the information from #1 and both stores it in a database and makes it available to a user.
 3. A database to store the collected data as well as application configuration information.
 4. Some type of front end that provides access to administrative components, the stored data, reports, alarm, etc.

In many ENMS systems, some or all of these components will be combined into a single installable application with no control from the end-user side as to what technologies and platforms are supported (for example, only a single DB is supported, or the application only runs on Windows, etc.). In other systems, some level of component flexibility is supported.

So, at the end of the day, virtually all ENMS systems can be broken down into similar components. The variation between them generally comes from how well or how much they can do the following:

1. Scale to larger or distributed environments
2. Handle many different types of data inputs
3. Provide an easy-to-use front end yet still offer advanced functionality
4. And of course, how much they cost

This is pretty much the way the ENMS world has looked for the last 30 years! Of course, newer technologies like server virtualization, Network Function Virtualization (NFV), and the burgeoning landscape of the Internet of Things (IoT) have pushed ENMS vendors to add functionality (and cost!) to their products, but otherwise we haven't seen any real innovation in the way this functionality has been delivered.

I believe that one of the major missing pieces to the architecture described above is the ability to offer the same functionality, but at a much lower cost and complexity of ownership. In order to accomplish this dual goal, we look to the other major disruptive technology innovation in the last 5-7 years: Software as a Service (SaaS).

In a SaaS model, all computing resources (hardware and software) are delivered as a service over a network (typically

the Internet). In this model, users are provided access to application software and databases, but do not own the underlying applications or technology. The cloud providers manage the infrastructure and platforms on which the applications run.

Ultimately, the business benefit derives from the potential to reduce IT operational costs by outsourcing hardware and software maintenance and support to the cloud provider. This enables the business to reallocate IT operations costs away from hardware/software spending and personnel expenses, towards meeting other IT goals. In addition, with applications hosted centrally, updates can be released without the need for users to install new software.

Simply put, by using a Cloud/SaaS model, an end-user gets all of the benefits of the applications they need without all of the up-front cost, or ongoing complexity of maintaining that solution. The very rapid acceptance of this model – virtually all organizations now use at least some SaaS software – has proven that this approach works and is a valuable to the IT and business community.

Referring back to the ENMS architecture diagram and discussion above, the issue with deploying network management completely as a cloud service, is seen at the bottom – the instrumentation layer. This is where some piece of software is communicating and taking measurements from the actual network devices, i.e. routers, switches, servers, etc.

In order for many of these measurements to be meaningful, they need to be performed as locally to the end devices and systems as possible. For example, an ICMP packet used to measure local availability and response time, really needs to be performed within the confines of the local network that the end device is designed to serve. A ping sent over the Internet from the Cloud/SaaS datacenter would really be testing the delay in the Internet connection and not how responsive the end device is to local users. This is the case for many other measurements as well.

Also, certain types of measurements and other ENMS activities can result in a large amount of data being delivered from the end devices to the ENMS instrumentation systems. Things like NetFlow and certain SNMP polls are a good

▶ example. Again, sending this information in a raw form over the Internet would be more of an exercise in seeing how quickly the NMS could fill up the clients' Internet connections than in actually gathering meaningful information.

Because of this, the best way to get the financial and support benefits of a SaaS solution, but still maintain an effecting ENMS system would be to implement a hybrid cloud architecture. As seen in Figure 3, this deployment method merely keeps the instrumentation layer "on premises" but moves the rest of the platform to the cloud. This is why we call it a hybrid cloud system.

In order to actually deploy a hybrid cloud solution as described above, a few things have to be available in the ENMS system.

The first and most important is the ability to unbundle the instrumentation layer from the server and DB level. This is not the case for most ENMS systems on the market today. Understandably, many vendors take a "bundled" approach to delivering their solution, meaning that all an end-user needs to do to install the product is download (or buy) a single piece of software (or appliance) that combines all of the various components of the ENMS in one platform. This generally makes for a less complex deployment in a traditional setting, but makes it impossible to separate those components in a multi-tiered way for hybrid cloud deployments.

The second piece is somewhat related to the first; the ability of the instrumentation layer components to operate autonomously from the rest of the platform. In a multi-tiered environment, if the connection from the instrumentation layer (which we show as the Agent) is lost, the Agent needs to be able to continue to monitor the network. This requires (among other things) local storage of

data and on-board intelligence as to what it is monitoring and how to send an alert if a failure occurs.

Lastly, any cloud-based SaaS application by definition must support "multi-tenancy", i.e. the ability to support multiple customers on a single platform. This means supporting things like:

1. Overlapping IP addresses
2. Multi-user/group authentication
3. Customer data partitioning

Many of these considerations (specifically support for overlapping IPs) are not naturally part of an ENMS because it would be assumed that a single end user would not have overlapping IPs.

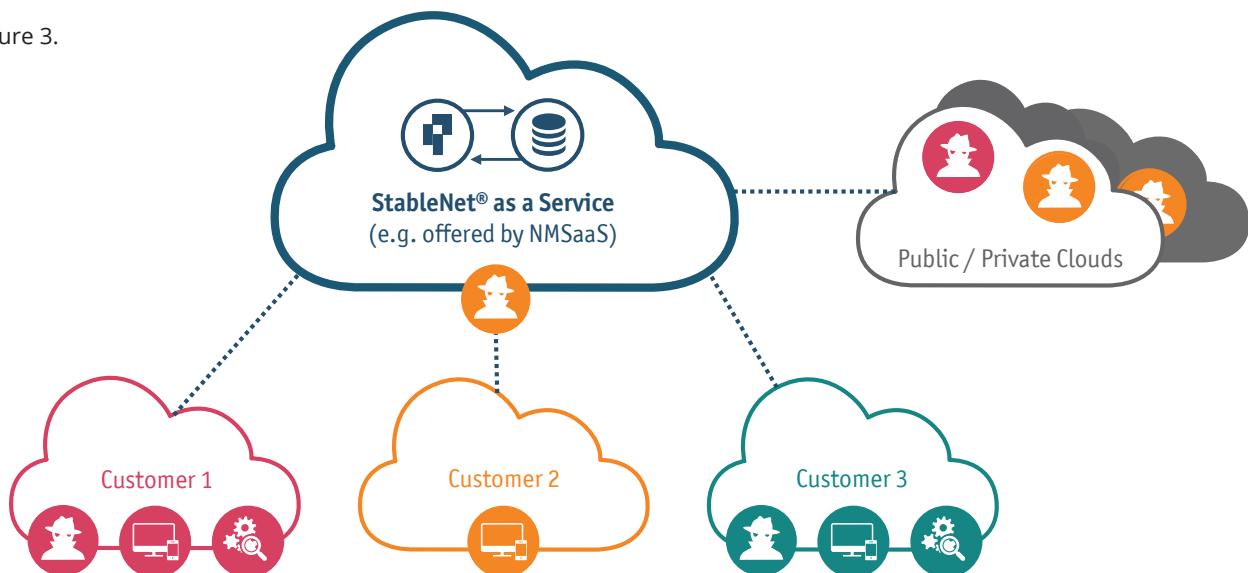
However, the SaaS solutions are designed with these criteria in mind and therefore able to support a wide variety of customer environments and multi-tenancy considerations like Managed Service Providers.

Hopefully, you can see how Pizza and software really can be alike. If you know what you are doing then making homemade pizza can be a great way to go. You can more directly control the ingredients and if you want to make changes to suite your taste, it's easier to do. That is just like on-premise software which certainly can be a fantastic solution if you have the time, expertise, and budget.

On the other hand, dining out, which takes almost all of the work out of getting a pizza, might be the best option for you. Likewise, moving from traditional on-premise network management to more a modern SaaS model might be right for your organization.

For further information on Infosim® products and services, please visit: www.infosim.net or email: info@infosim.net

Figure 3.



About StableNet®

StableNet® is a 3rd generation highly automated Network Management System. The key differentiation of StableNet® to other legacy type Operational Support Systems (OSS) is that StableNet® is a unified OSS system with three integrated functionalities that focus on

Configuration, Fault, and Performance Management, with automated Root Cause Analysis (RCA). StableNet® can be deployed on a multi-tenant, multi-customer or dedicated platform and can be operated in a highly dynamic flex-compute environment.