Software as a Service: Best Practices for the Modern Era

INTRODUCTION

Truth be told, creating SaaS (software as a service) products is every bit as much an art as it is a science. Whether you're talking about web or mobile products doesn't matter - you need to create better and more scalable applications than ever before in a way that also allows you to lower costs at the same time. Striking this balance between cost and complexity is a challenge, to be sure - but one thing that it does NOT have to be is impossible. If you truly want to make the best SaaS products possible while taking advantage of all the benefits that the cloud has to offer, there are a few key things to keep in mind.

SaaS Applications Should be Self-Service

An important aspect for consideration is that SaaS applications need to be self-serviced; anybody wishing to use the service should be able to register and use the application without any assistance from the administration or technical personnel. Far too many developers in the modern era make the mistake of requiring users to contact the admin or support team for help - this is just too far removed from how people actually like to work these days and is likely to put people off before they really have a chance to get started.

Because SaaS applications promise versatility on behalf of end users, you need to put your money where your mouth is and guarantee accessibility on multiple desktops at all times.

You need to go out of your way to make things easy for users and a self-service application (which will also eliminate unnecessary work for your admin and support teams, mind you) is the best way to do just that.

Its All About Personalization

Along the same lines, you should make it easy for your end users to make an SaaS product their own - which means that a high level of personalization is no longer a recommendation but a requirement. SaaS is all about freedom - never forget that. User should have the freedom to make all the decisions relating to set-up, including details about their intended use of the app. It's about building an app that works for your users, instead of releasing one that makes your users work for the app.

Integration Has Never Been More Important

Part of the reason why SaaS applications are so essential to the modern enterprise has to do with integration - never forget that yours will NOT be the only application that someone will be using.

Developers MUST design their SaaS applications so that they can be easily integrated with as many others as possible. Thankfully, standard APIs should make it easy for you to design the SaaS application so that integration is possible with other SaaS and on-premise applications

Monitoring and Maintenance

With many software applications, consumers will have access to an entire internal IT team to help them with maintenance. However, when it comes to SaaS applications, maintenance is often solely up to developer. This is very much a good thing, but it requires you to keep some essential things in mind.

At Pegasus One, for example, we dedicate as many resources as possible to actively monitoring an SaaS application at all times. We look for downtimes so that when issues do occur, we can get things back up and running as quickly as possible. We also proactively scan for ANYTHING that may be negatively affecting the experience that we offer to our users. Any SaaS application built in the modern era needs to be created, monitored and maintained with these same factors in mind - no exceptions.

To put it another way, any SaaS application should absolutely be developed in a way that allows the service provider to monitor application performance, errors and outages seamlessly and proactively. This is NOT something you do after the fact - it's now and will forever remain a part of the development process. Third party tools like AppDynamics, NewRelic and even infrastructures provided by hosts like AWS (Amazon Web Services) can significantly help in these areas.

Security, Compliance and More

Another key factor to never, ever forget is that as a developer, the responsibility for the security of both the SaaS application and any accompanying hardware is yours and yours alone. If your SaaS application is designed so that multiple tenants will share the same infrastructure, you should guarantee that the data is segregated for the privacy of the tenants. Likewise, data should only be accessible based on the people who actually need it to do their jobs - rather than everyone and anyone at any time.

Building a Better Design

While designing applications for multiple tenants, remember to make sure that there is a way to reliably distinguish between those tenants. Without this mechanism in place, you always run the risk of sending the wrong information to the wrong user - a situation you absolutely do not want to be in.

Any Device, Any Time, Anywhere

Because SaaS applications promise versatility on behalf of end users, you need to put your money where your mouth is and guarantee accessibility on multiple desktops at all times. All that matters is the app itself - whether your user is on a desktop computer, a laptop, a smartphone or a tablet doesn't matter. It's totally irrelevant. You need to be guaranteeing the same user experience anytime, anywhere, absolutely no exceptions.

Scalability Is a Matter of Great Importance

It's important for you to understand that there is always a difference between application scalability and database scalability, particularly when it comes to SaaS applications. At Pegasus One, for example, we achieve application scalability as needed by moving our clients' SaaS applications to servers that is more spacious or powerful as the need arises. your architecture should always allow for adding identical servers to make it easier for the application itself to handle heavier loads during times of peak usage.

Database Scalability Matters, Too

Along the same lines, it's also important to remember that database scalability will often still be incredibly important. If a database needs to be upgraded (like if it needs to be able to keep up with the demands of a larger number of users, for example), you need to have the ability to do so. Facilitating more transactions is also something that will be far, far easier if you make database scalability a priority.

The key takeaway from all of this is that applications must be developed in a way that not only allows them to scale seamlessly from your perspective, but also without causing outages from the perspective of your end users. This shouldn't be something you have to work for - it should be automatic. Your development process should already allow for this. Application scalability AND database scalability will help a great deal to this end.

Performance Drives Everything

Finally and most importantly, performance. In order to guarantee the level of performance that end users both demand and deserve, SaaS applications MUST be designed so that performance is A) scalable, and B) always available. Your application should be able to serve tenants of all sizes, particularly as it becomes more popular. Remember that users from all over the Internet will be using your application - this is something that you should strive to look at as an opportunity, not a disadvantage.

Pegasus One
1440 N Harbor Blvd #900, Fullerton, CA 92835, USA
Phone: +1 (714) 485-8104