Why Multi-instance Arch?

*Data Isolation,Availability,Security,Scalability,Stability,Customer-driven upgrades*

Delivering the application to the end-user require multi-instance OR multi-tenant Arch.

# First Let’s review Multi-tenant Arch Drawbacks:

What Multi-tenant really means?

This means all your users will be sharing a single database instance and a single instance of an application.

In these multi-tenant clouds,customers share the same software and infrastructure.

There are three major drawbacks of the multi-tenant model for their customers:

1. Your **data is commingled**(mixed)

You rely on the cloud provider to logically isolate your data from everyone else’s data. The data of your direct competitor could be commingled with yours in the single database. Commingling doesn’t mean you can see another company’s data; access to the multi-tenant environment is controlled. But all the same, your data is not physically separate and relies on software for separation and isolation.

Further, a security breach to the cloud provider could expose your data along with everyone else commingled on the same multi-tenant environment.

1. Multi-tenant architectures rely on large and complex databases that require hardware and software maintenance on a regular basis, resulting in **availability issues** for customers.
2. Any action that affects the multi-tenant database affects all shared customers.

When software or hardware issues are found on a multi-tenant database, it causes an outage for all customers, and an upgrade of the multi-tenant database upgrades all customers. **Your availability and upgrades are tied** to all other customers that share your multi-tenancy.

**Conclusion:**

With its inherent data isolation and multiple availability issues, multi-tenancy is a legacy cloud computing architecture that will not stand the test of time.

--------------------------------------------------------------------------------------------------------

# Enter the Multi-Instance Cloud

In contrast, a multi-instance architecture gives every customer its own unique database, which means that it is impossible for your data to be commingled(mixed) with any other customer.

The multi-instance architecture is not built on large centralized database software and infrastructure. Instead, it deploy instances on a per-customer basis, allowing the multi-instance cloud to scale horizontally and infinitely.

Each customer instance is a unique software stack

Benefits; true data isolation, high availability and customer-driven upgrade schedules.

* **True data isolation:**

This makes hardware and software maintenance on these unique customer instances far easier to perform and issues can be resolved on a customer-by-customer basis. Unlike a multi-tenant architecture, customers are not grouped together in a shared database.

* **High availability:**

With multi-instance cloud, we are able to move customers individually out of harm’s way for routine maintenance and unexpected issues.

* **Customer-driven upgrades:**

As we described above, the multi-instance architecture allows us to perform actions on individual customer instances, such as performing an upgrade. Each individual instance can be upgraded on a schedule that fits the compliance requirements and needs of the enterprise.

* **Stability**:

Instead of a single point of failure (the single application instance), each customer can exist in their own instance. If one instance fails, the others will remain unaffected.

* **Scalability :**

With a multi-instance architecture, scaling up is a simple matter of adding more resources. However, with a multi-tenant architecture, you could reach a point where you need to come up with a clustered application architecture whose deployment is usually far from trivial.

* **Security**:

 When you are using a single database, all your data lives together. This becomes a major problem in the event of a security breach because all customers’ data can become vulnerable when a single account is compromised. With a multi-instance architecture, only a single customer’s data can be at risk.

In short, the multi-instance architecture puts our customers in control of their cloud.

Resources:

<https://servicematters.servicenow.com/2016/02/01/why-cloud-architecture-matters-the-multi-instance-advantage-over-multi-tenant/>

<https://medium.freecodecamp.org/getting-started-with-kubernetes-for-your-saas-91e91116dd7d>