



Gartner | 2011
COOL VENDOR

3 Ways to build a SaaS Product

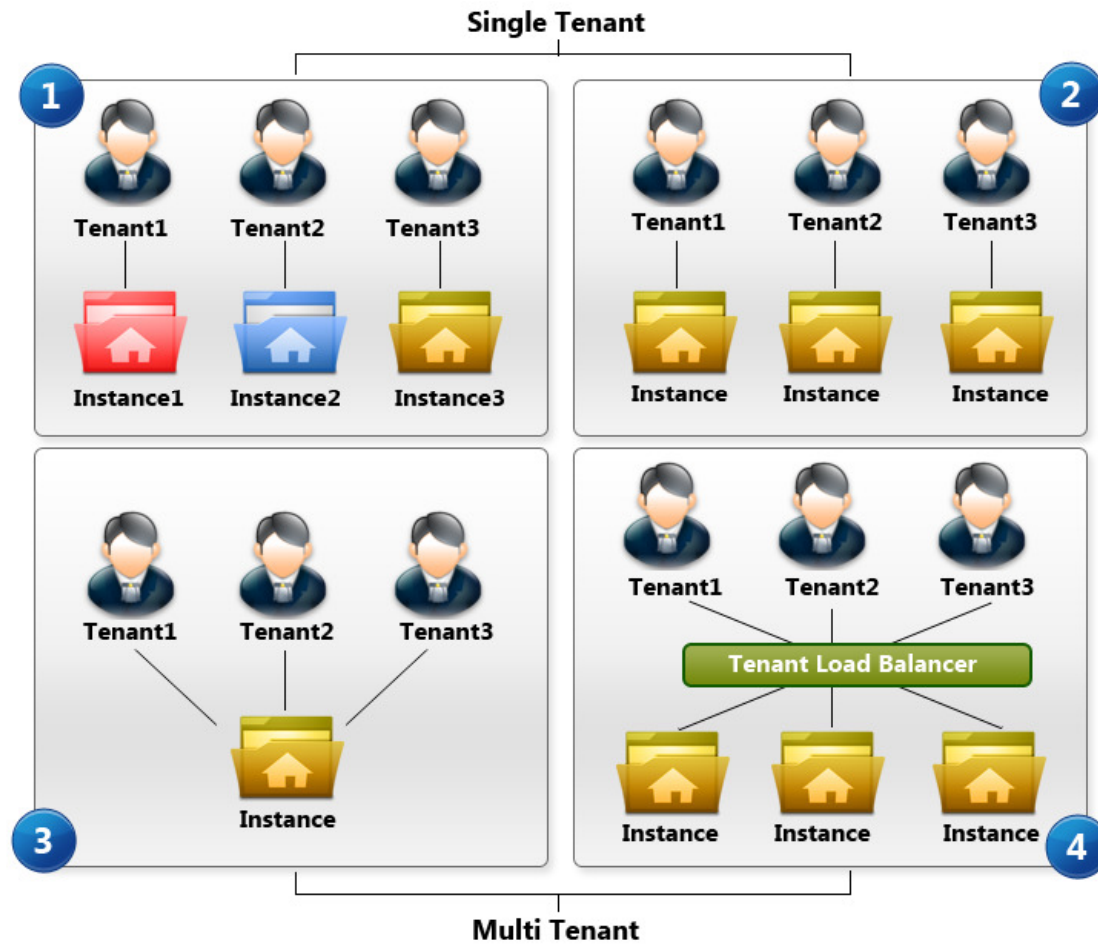
Asteor Software Inc

Ram Kumar - Director – Product Management

SaaS without Multi-tenancy >>

- ❖ A separate server instance for each customer
 - Separate Box
 - Separate Shared Hosting Slice
 - Virtual Machine in a data centre
 - Amazon Instance
 - Azure VM Instance
 - Azure Instance with SQL Azure
- ❖ Multi-tenant SaaS
 - Infrastructure is same as above
 - Same instance serves multiple customers

Multi-tenancy – Maturity Models



Advantages of Multi-tenancy

- ❖ Cost Savings on
 - Hosting
 - Production Administration and Maintenance
 - Set up / Provisioning / De-provisioning for each customer
 - Customization for each customer
 - Maintaining code base(s)
- ❖ Leveraging Cloud Elasticity
 - Handling widely varying loads within and across customers

1: APaaS and VPaaS >>

- ❖ Example ; Force.com, OrangeScape.
- ❖ Advantages
 - Faster Development (4GL, YGWYS, BA/User designed apps)
 - Scalable Deployment
 - Very little headache
 - Ideal for Internal / Long Tail apps of Enterprises, SMBs.
- ❖ Disadvantages
 - Complex requirements require workaround coding
 - Example : Usage of Macros and VB in MS Access
 - Limited Flexibility, Technical and Strategic Control
 - Platform Lock-in, Vendor lock-in, IP Valuation issues.
 - Exit options : Back to Zero

2: Emulating Multi-tenancy @ run time (SaaSGrid) >>

- ❖ Database sharing is handled by run-time container / SaaS Server
- ❖ Customer on-boarding, user management, feature packaging, price book management etc. are handled by the SaaS Server.
- ❖ Advantages :
 - Suitable for existing web applications
 - Faster time to market
- ❖ Disadvantages
 - Customizability and Configurability has to be designed and engineered at the application level.
 - The SaaS Server will only replicate it for multiple tenants.
 - Vendor lock-in, Platform Lock in, High upfront investment
 - Exit options : Back to where you started.

3: Custom Development

- ❖ 3.1 Monolithic SaaS Application stack
- ❖ 3.2 Build a Multi-tenant SaaS Framework In house
 - Build business functions on top of the framework
 - The framework can be maintained and upgraded independently without disrupting the business functionalities.
 - The framework is re-usable for multiple products within the organization
 - Very high discipline is required to keep the framework independent and to continuously keep enhancing it
- ❖ 3.3 Buy a Multi-tenant SaaS Framework
 - With or without Source code

3: Custom Development - Advantages

- ❖ Maximum flexibility, freedom and control
- ❖ Use industry standard technology stacks such as .NET or J2EE
- ❖ Leverage large eco system of resources, tools and components
- ❖ No vendor lock in or platform lock in (other than the technology stack chosen)
- ❖ Better IP Valuation
- ❖ In house or Bought Out Framework advantages
 - Re-usability and cost apportionment across products
 - Easy to maintain, upgrade and enhance independently
- ❖ Bought out framework advantages
 - Saves time & cost, reduces risk. Robust and tested foundation.
 - Loosely coupled, easily extensible without touching framework code.

3: Custom Development – Dis-advantages

- ❖ Build your own monolithic Multi-tenant application stack
 - Time, Cost and Expertise required : 30-40% overhead
 - Maintenance and Upgrades are complex
 - Poor re-usability
- ❖ Build your own re-usable Multi-tenant framework
 - Time, Cost, Expertise (Can be apportioned across products)
 - Has to be more generic and loosely coupled.
 - Strict Discipline : From design & Engg, to maintenance & enhancement
- ❖ Buy a ready made framework (With or without source code)
 - Additional Licensing costs.
 - Framework Version management (if bought with source code)



Typical Stack

