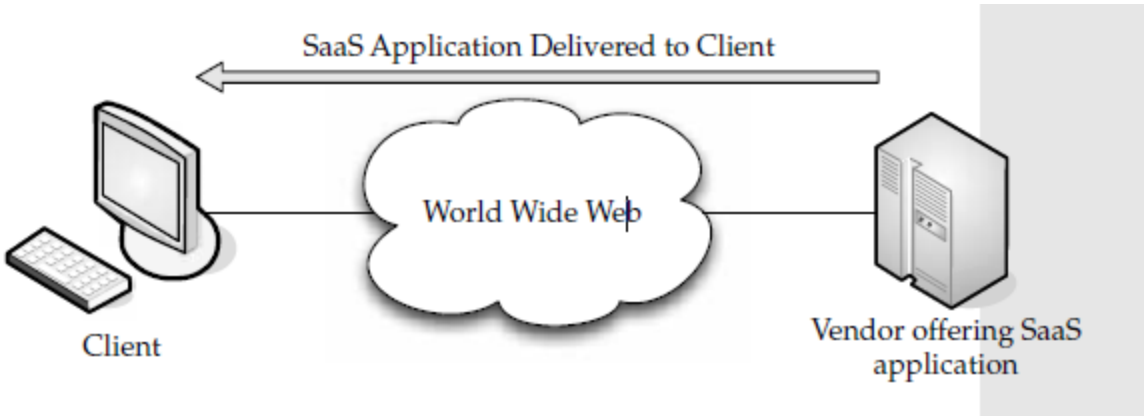
Software as a Service(SaaS)

SaaS Definition

- SaaS (Software as a Service) is an application hosted on a remote server and accessed through the Internet.



SaaS Definition

- SaaS (Software-as-a-Service) is a model of software deployment where an application is hosted as a service provided to customers across the Internet.
- By eliminating the need to install and run the application on the customer's own computer, SaaS alleviates the customer's burden of software maintenance, ongoing operation, and support.
- Conversely, customers relinquish control over software versions or changing requirements; moreover, costs to use the service become a continuous expense, rather than a single expense at time of purchase.

SaaS Definition

- SaaS service through the Gartner Group as follows:
- *“software that is owned, delivered and managed remotely by one or more providers. The provider delivers an application based on a single set of common code and data definitions, which are consumed in a one-to-many model by all contracted customers, at anytime on a pay-for-use basis, or as a subscription based on usage metrics.”*

Example

- web-based email service
  - like Microsoft (Hotmail), Google (Gmail), and Yahoo! (Yahoo Mail)
- SalesForce.com
- EyeOS
- … etc

Email is stored by the vendor, and accessed by the client.

Traditional Software Model

- large upfront licensing costs
- annual evergreen support costs
- annual renewal for upgrades and support
- Increasing the number of users may raise the base cost of the package
- requires hardware deployment, servers, backup and network provisioning in order to accommodate the number of users on and off-campus.
- Security architecture protect this valuable resource from unauthorized access

Traditional Software Model

- tend to be highly customizable
- on-going maintenance and management of the application
- providing the logical and physical security
- offering end-user training and support

Software-as-a-Service (SaaS) Model

- based on a recurring subscription fee
- typically pay as you go model
- costs are directly aligned with usage
- does not require any hardware and can run over the existing Internet access infrastructure
- support, training, infrastructure and security risks- all by SaaS vendor
- provides reserve capacity to handle any spikes in usage, outages or network mishaps and to do this continuously, globally and securely

SaaS vs. ASP(Application Service Provider)

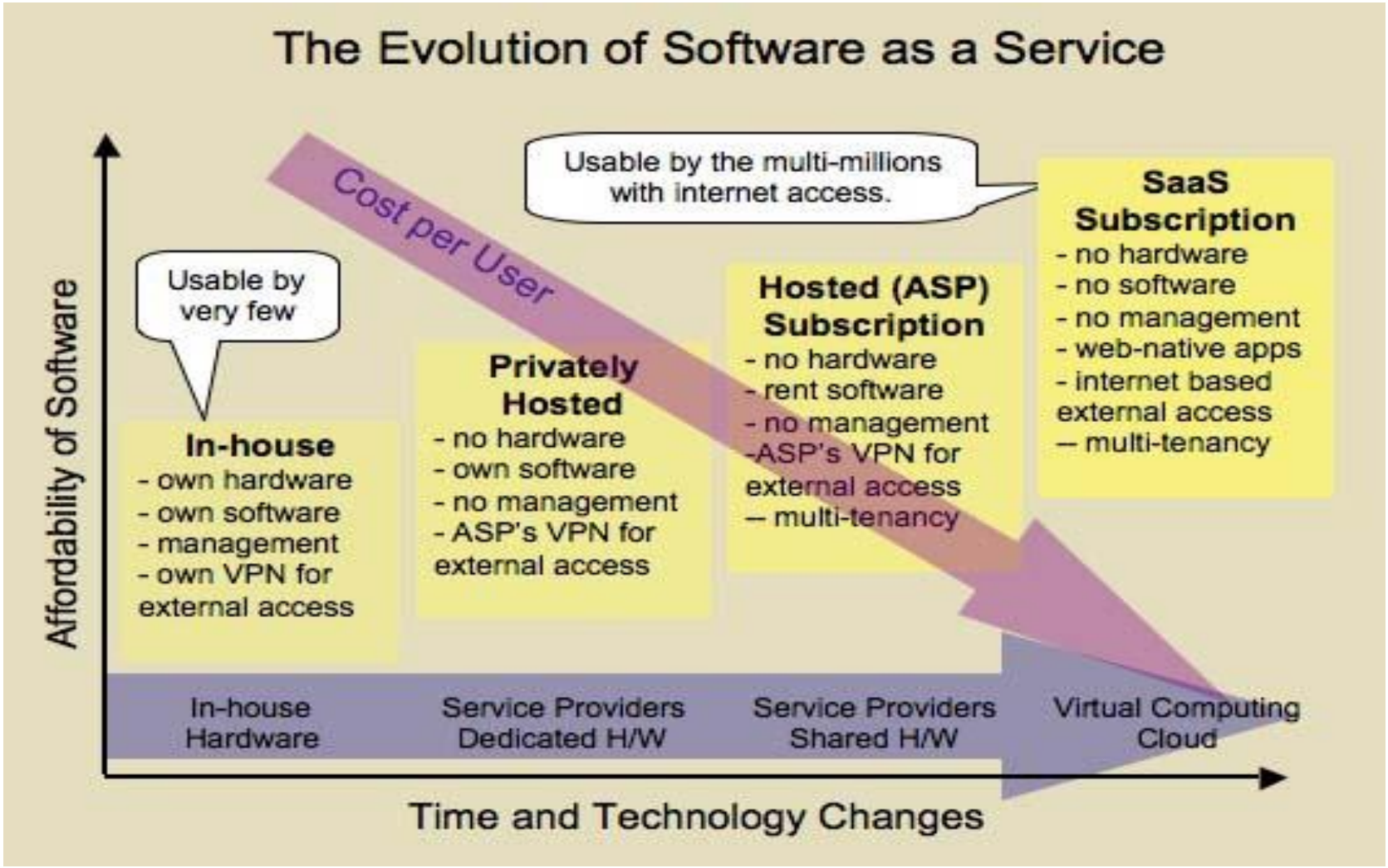
1. ASP applications are traditional single-tenant applications but hosted by a third party. They are client-server applications with HTML front ends added to allow remote access to the application.
2. The applications are hosted by third-parties who ordinarily do not have specific application expertise.
3. The applications are not written as net-native applications. As a result, the performance may be poor and application updates are no better than self-managed premise-base applications.

- By comparison,
- SaaS applications are **multi-tenant applications**, hosted by a vendor that has all the **application expertise** and they have been designed as **net-native applications** that get updated on an on-going basis.

SaaS Categories

- **Line of business services**
  - These are business solutions offered to companies and enterprises. They are sold via a subscription service. Applications covered under this category include business processes, like supply-chain management applications, customer relations applications, and similar business-oriented tools.
- **Customer-oriented services**
  - These services are offered to the general public on a subscription basis. More often than not, however, they are offered for free and supported by advertising. Examples in this category include the web mail services, online gaming, and consumer banking, among others.

Evolution of SaaS



SaaS Key Characteristics

- Network-based access to, and management of, commercially available software.
- Activities managed from central locations rather than at each customer's site, enabling customers to access applications remotely via the Web.
- Application delivery typically closer to a one-to-many model (single instance, multi-tenant architecture) than to a one-to-one model, including architecture, pricing, partnering, and management characteristics.
- Centralized feature updating, which obviates the need for end-users to download patches and upgrades.

SaaS other Characteristics

- Modest software tools
  - The SaaS application deployment requires a little or no client side software installation, which results in the following benefits:
    - *No requirement for complex software packages at client side*
    - *Little or no risk of configuration at client side*
    - *Low distribution cost*
- Efficient use of software licenses
- Platform responsibilities managed by provider
  - All platform responsibilities such as backups, system maintenance, security, hardware refresh, power management, etc. are performed by the cloud provider.
- Multitenant solutions
  - The design principles of a multi-tenant system offer a high level of maintainability. For example, if a customer requests for few additional fields in one of the pages you can easily add them through custom field's module.

Advantages of SaaS

Cloud User Advantages

- · Offsite deployment
- · Low overhead or low costs
- · Decentralized
- · Customizable
- · On the fly pay as u go

Advantages of SaaS

Cloud Provider Advantages

- · Application as a service
- · Scalable applications
- · High customization
- · Highly stable & common base code
- · Easy maintenance
- · Maximum efficiency
- · Flexible costs based on usage

Disadvantages of SaaS

- Security
- Latency issue
- Total Dependency on Internet
- Switching between SaaS vendors is difficult

Challeges of SaaS

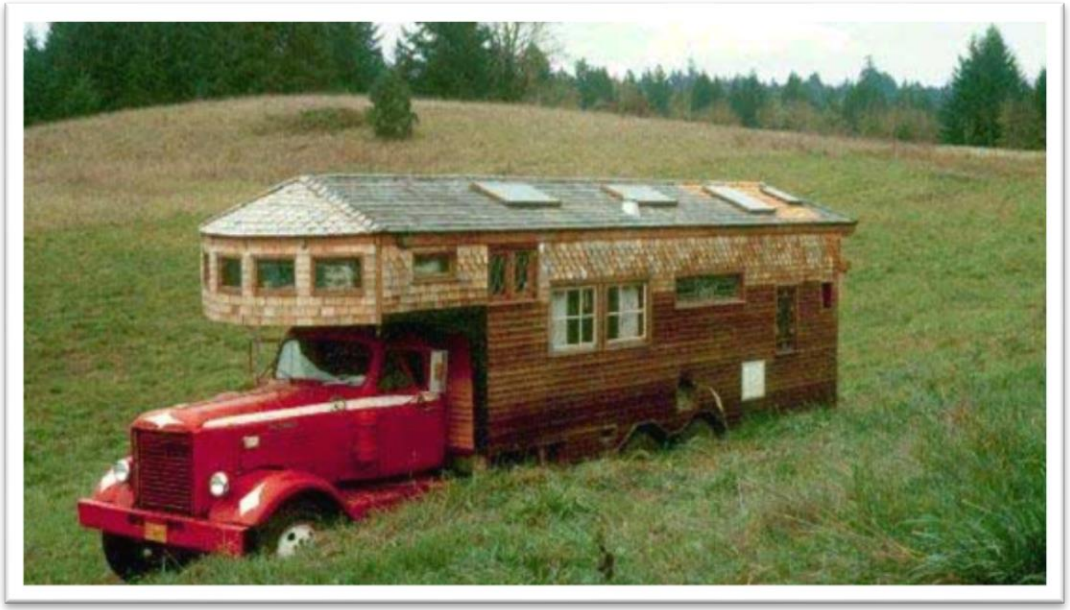
- Controllability
- Visibility and Flexibility
- Security and Privacy
- High performance and Availability
- Integration and Composition
- Standards

Scalability & Elasticity

- What is scalability ?
  - A desirable property of a system, a network, or a process, which indicates its ability to either handle growing amounts of work in a graceful manner or to be readily enlarged.
- What is elasticity ?
  - The ability to apply a quantifiable methodology that allows for the basis of an adaptive introspection with in a real time infrastructure.
  - **Scalability** adapts only to the "workload increase" by "provisioning" the resources in an "incremental" manner.
  - **Elasticity** adapts to both the "workload increase" as well as "workload decrease" by "provisioning **and** deprovisioning" resources in an "autonomic" manner.
- But how to achieve these properties ?
  - Dynamic provisioning
  - Multi-tenant design

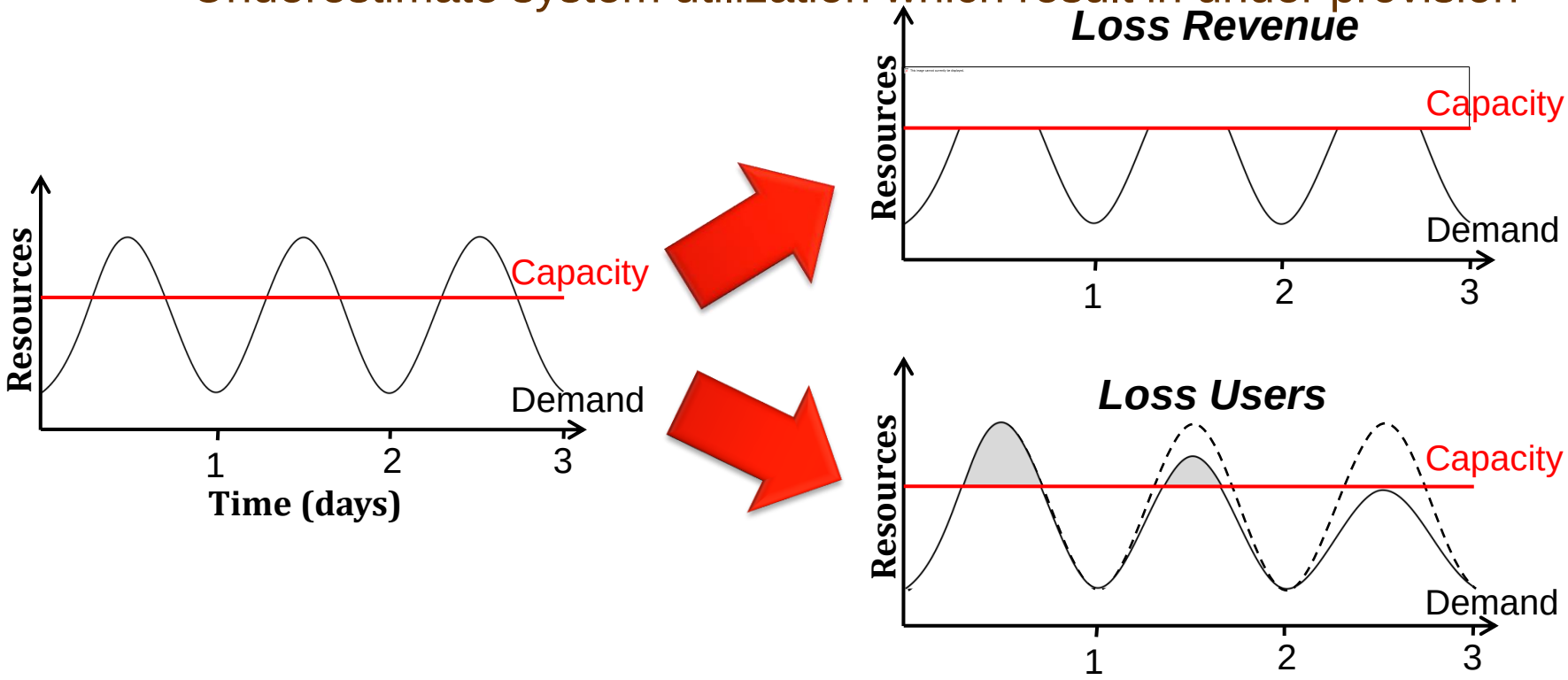
Dynamic Provisioning

- What is dynamic provisioning ?
  - Dynamic Provisioning is a simplified way to explain a complex networked server computing environment where server computing instances are provisioned or deployed from a administrative console or client application by the server administrator, network administrator, or any other enabled user.



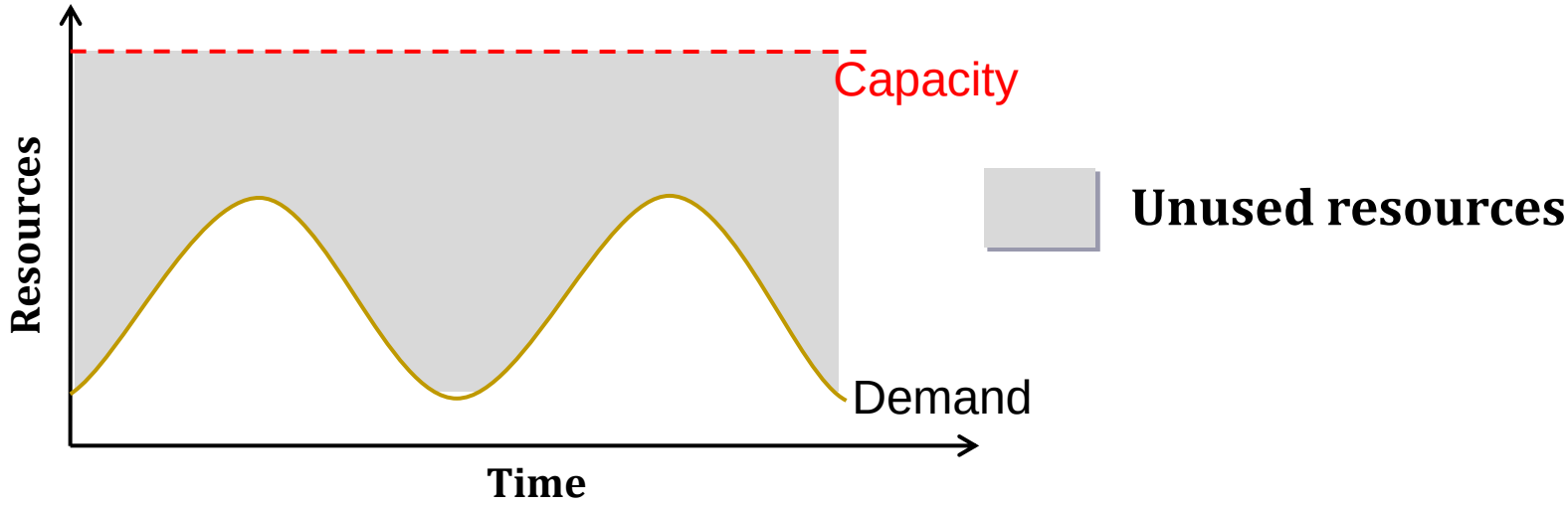
Dynamic Provisioning

- In traditional computing model, two common problems :
  - Underestimate system utilization which result in under provision



Dynamic Provisioning

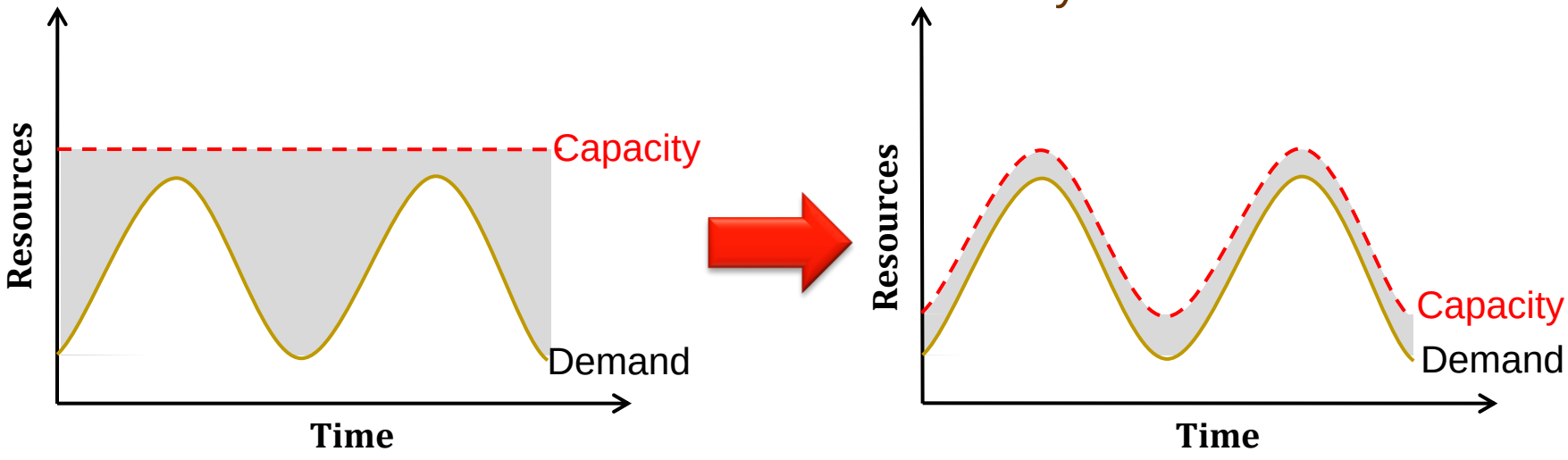
- Overestimate system utilization which result in low utilization



- How to solve this problem ??
  - Dynamically provision resources

Dynamic Provisioning

- Cloud resources should be provisioned dynamically
  - Meet seasonal demand variations
  - Meet demand variations between different industries
  - Meet burst demand for some extraordinary events



Multi-tenant Design

- What is multi-tenant design ?
  - Multi-tenant refers to a principle in software architecture where a single instance of the software runs on a server, serving multiple client organizations.
  - With a multi-tenant architecture, a software application is designed to virtually partition its data and configuration thus each client organization works with a customized virtual application instance.
- Client oriented requirements :
  - Customization
    - *Multi-tenant applications are typically required to provide a high degree of customization to support each target organization's needs.*
  - Quality of service
    - *Multi-tenant applications are expected to provide adequate levels of security and robustness.*

Existing SaaS Maturity Models

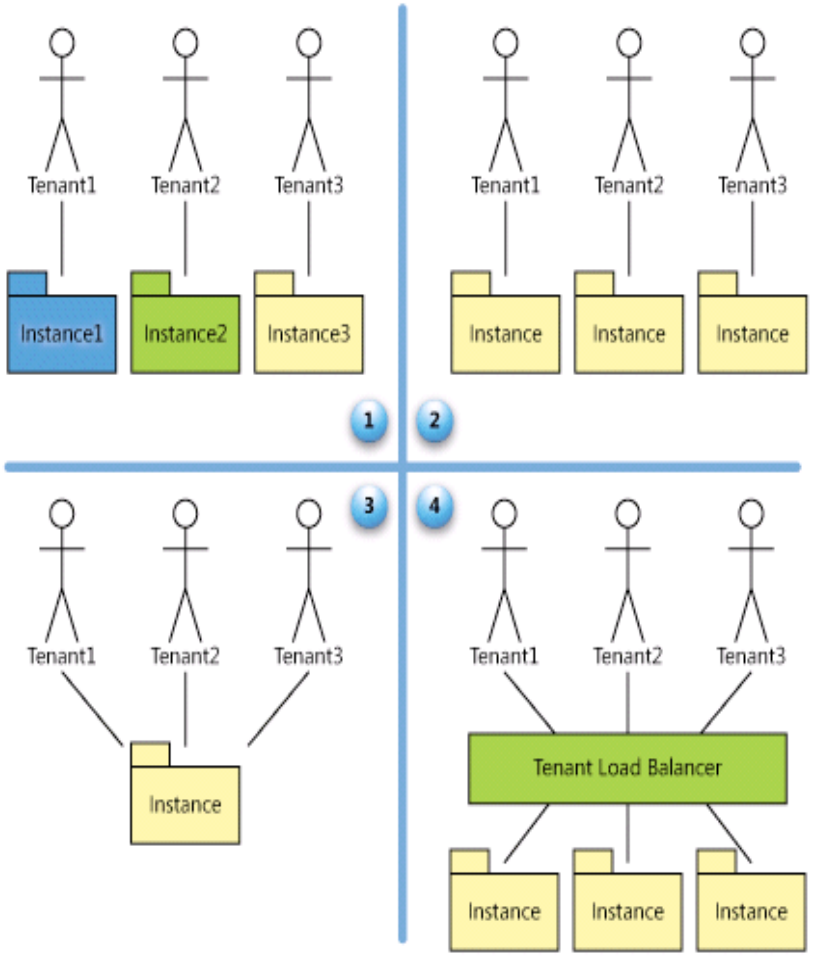
1. Saas Simple Maturity Model by Microsoft Corporation (2006)
2. Model by Forrester Research (2008)

SaaS Simple Maturity Model (Microsoft, 2006)

- Model on Single Packaged Application
- Focused on SaaS Application Architecture
- Three Key Attributes of an Architecture:
  1. **Configurability:** Metadata used to configure the way the application behaves for customers
  2. **Multi-tenant Efficiency:** Maximizing the sharing of resources across tenants
  3. **Scalability:** Maximizing concurrency, resource efficiency

SaaS Simple Maturity Model: Four Levels

- **Level 1:** Ad Hoc/Custom
- **Level 2:** Configurable
- **Level 3:** Configurable & Multi-Tenant-Efficient
- **Level 4:** Scalable, Configurable, Multi-Tenant-Efficient



Level 1: Ad Hoc/Custom

- resembles the **conventional ASP** (application service provider) software delivery model
- each client has his/her own personalized version of a hosted application, which he/she runs an instance of the software app on the host's servers.
- In terms of architecture, software at level I maturity closely resembles traditional line-of-business software, in which multiple clients or customers within a single organization are able to form a type of connection to a single instance running on the server
- conventional client-server apps can be relocated to a cloud-based model usually at the initial level of maturity, and with lesser development effort or without having to re-architect the whole system

Level 2: Configurable

- SaaS vendor hosts a totally different instance of the SaaS application for each tenant
- each instance is personally customized for each tenant, all instances at this level utilize similar code implementation
- the vendor meets the needs or requirements of the customer by offering in-depth configuration options that enable the customer to alter the look of the application as well as its behavior to its users
- repositioning a conventional application as cloud-based at this maturity level may require additional re-architecting compared to the previous level

Level 3: Configurable & Multi-Tenant-Efficient

- Level III maturity is characterized by the vendor running a single instance serving each client with configurable metadata to provide unique, customized user experience and unique feature set
- This eliminates the need for server space to accommodate the many instances, allowing for efficient use of scarce computing resources than level II, thus, translating to lower costs.
- **disadvantage :** limited scalability

Level 4: Scalable,Configurable, Multi-Tenant-Efficient

- vendor hosts several clients on a load-balanced group of identical instances, but with each client's data stored separate, and configurable metadata offering each customer a phenomenal user experience and unique feature set
- Can be scaled to a large number of clients, as the number of instances and servers on the backend can be adjusted to meet demand

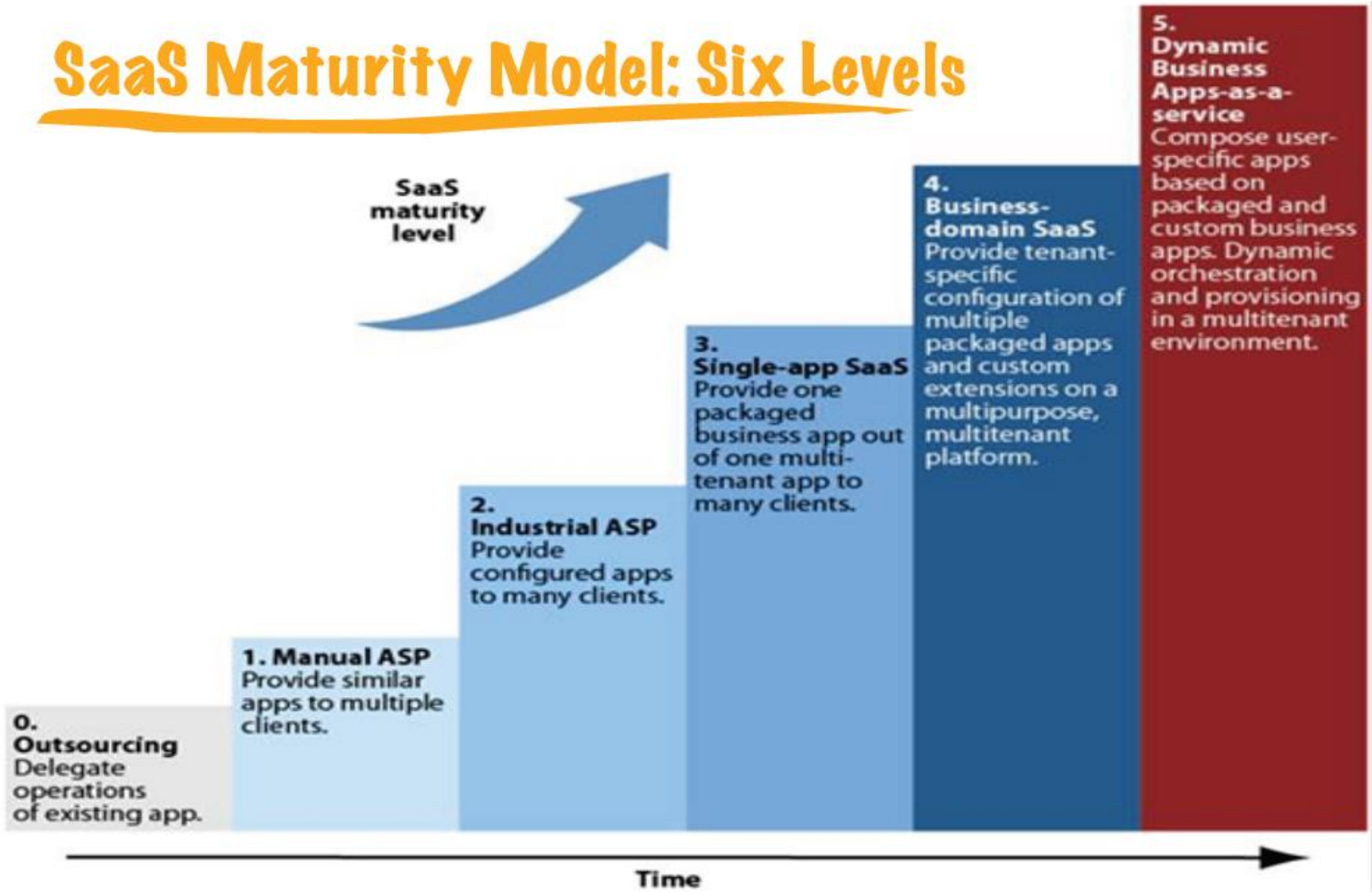
Summary

Maturity Level	Instances	Configur-ability	Multi-tenant Efficiency	Scalability
Level 1	Multiple different instances - ASP Model			
Level 2	Multiple identical instances- code sharing	X		
Level 3	Single instance - configurable metadata	X	X	
Level 4	Multiple identical instances with tenant load balancer	X	X	X

SaaS Maturity Model (Forrester, 2008)

- Evolution Model on SaaS Applications: guidance on realistic strategy transformation for software vendors and services providers considering an SaaS business model
- Focused on SaaS Application Domain

SaaS Maturity Model (Forrester, 2008)



Level 0: Outsourcing

- A service provider operates a major application or a unique application landscape for a large enterprise customer
- The outsourcing company can't leverage the application for a second customer
- Does not really qualify as SaaS

Level 1: Manual ASP

- A service provider runs packaged applications for multiple midsize enterprises
- Each client usually has a dedicated server running its instance of the application and is able to customize the installation in the same way as self-hosted applications
- Target midsize companies
- Similar to level 1 (Ad-hoc/custom) of Microsoft's model

Level 2: Industrial ASP

- A service provider runs identical packaged applications with customer-specific configurations to many customers
- Cut the operating costs of applications to a minimum
- Usually applicable to small and midsize business customers
- Similar to level 2 (Configurable) of Microsoft’s model

Level 3: Single-app SaaS

- An SaaS provider provides one packaged business application with Web-based user interface to many customers (e.g. Salesforce's initial CRM application)
- Customization is restricted to configuration
- Focus on small and midsize business customers
- Similar to level 3 (Configurable & Multi-Tenant-Efficient) of Microsoft's model

Level 4: Business-domain SaaS

- An advanced SaaS vendor provides not only a well-defined business application but also a platform for additional business logic
- Single application of level 3 complemented with third-party packaged SaaS solutions and custom extensions
- Satisfy the requirements of large enterprises
- Similar to level 4 of Microsoft’s model, yet with extension of composition with other applications

Level 5: Dynamic Business-domain Apps-as-a-Service

- An advanced SaaS vendors coming provides a comprehensive application and integration platform on demand, which they will pre-populate with business applications or services
- New paradigm: design for people, build for change
- Composition of user-specific business applications on various levels in a multi-tenant environment
- The resulting process agility will attract everyone, including large enterprise customers

Summary

Maturity Level	Definition	Single/Multiple Application	Configur ability	Multi-tenant Efficiency	Scal-ability	Equivalent Level in MS
Level 0	Outsourcing	Single app to one client				
Level 1	Manual ASP	Similar apps to multiple clients				Level 1
Level 2	Industrial ASP	Configured apps to many clients	X			Level 2
Level 3	Single-app SaaS	One packaged app to many clients	X	X		Level 3
Level 4	Business-domain SaaS	Configuration of multiple packaged apps and custom extension	X	X	X	Level 4 + custom extension
Level 5	Dynamic Business Apps-as-a-Service	Dynamic composition of user-specific apps based on packaged and custom business apps	X	X	X	Level 4 + dynamic compositio n