

Lets SAAS-ify that Desktop Application

Chirag Jog
Clogeny

IndicThreads.com Conference On
Upcoming Technology



2010: Cloud Computing

Pune, India

About me

- Chirag Jog
 - Computer Science Passout, PICT
 - Currently CTO at Clogeny Technologies.
 - Working on some cutting-edge Products in Cloud Computing.
 - chirag.jog@gmail.com

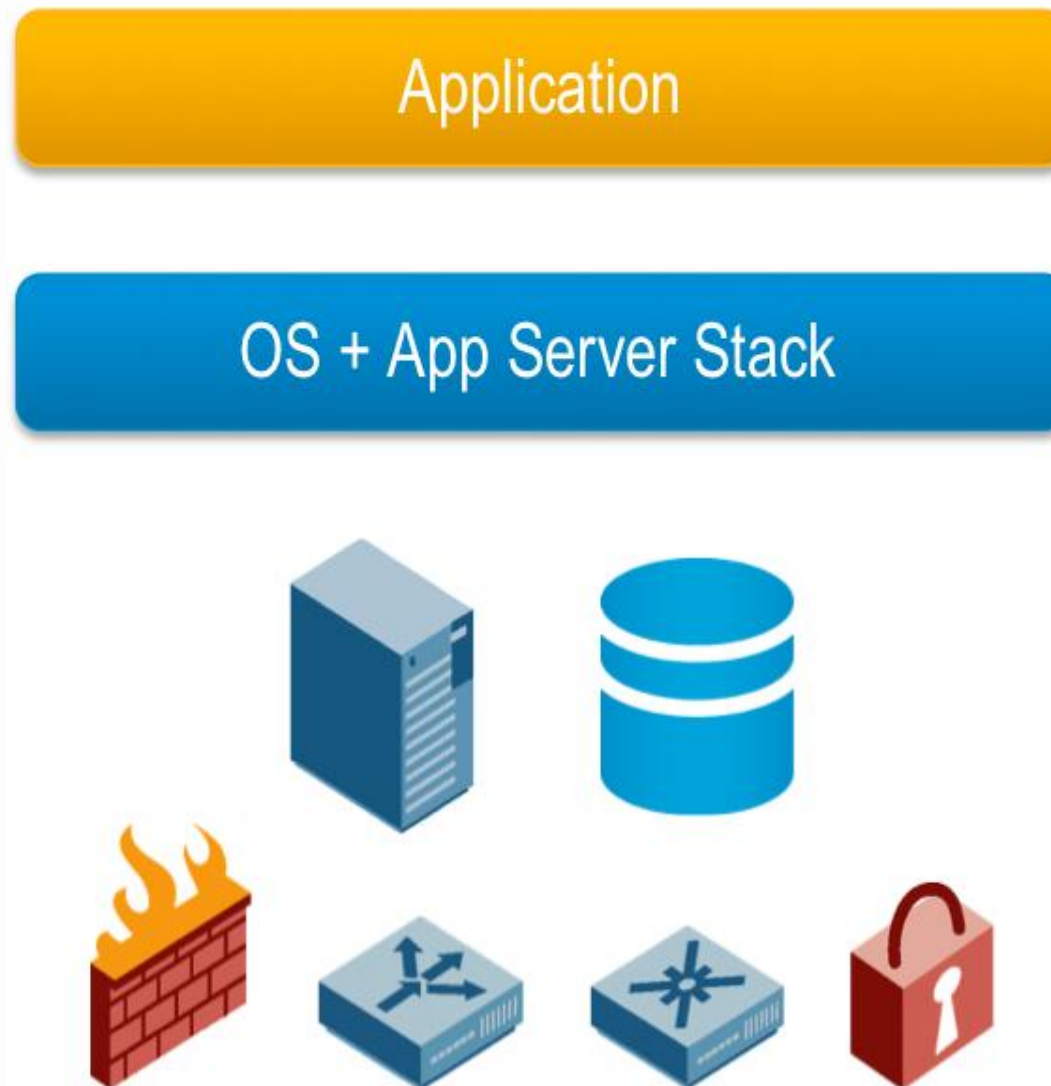


Agenda

- Traditional Web Setup
- Difference between IaaS, PaaS and SaaS
- Different ways to SaaS'ify
- Advantages of PaaS
- Real life Scenario
- Advantages of IaaS
- Using IaaS to SaaS'ify
- Using Amazon Web Services as an example for deployment
- Improving Business Logic
- Conclusion



Traditional Web 1.0 setup



Source : <http://onsaas.net/wp-content/uploads/2008/06/iaas-paas-saas.png>



IaaS

The end user sees a finished application

End User

Firewall Load Balancer

Compute Power

Virtual Machine is deployed and started

Virtual Machine Automation

The virtual machine is uploaded to storage and configured to use storage.

Software Owner

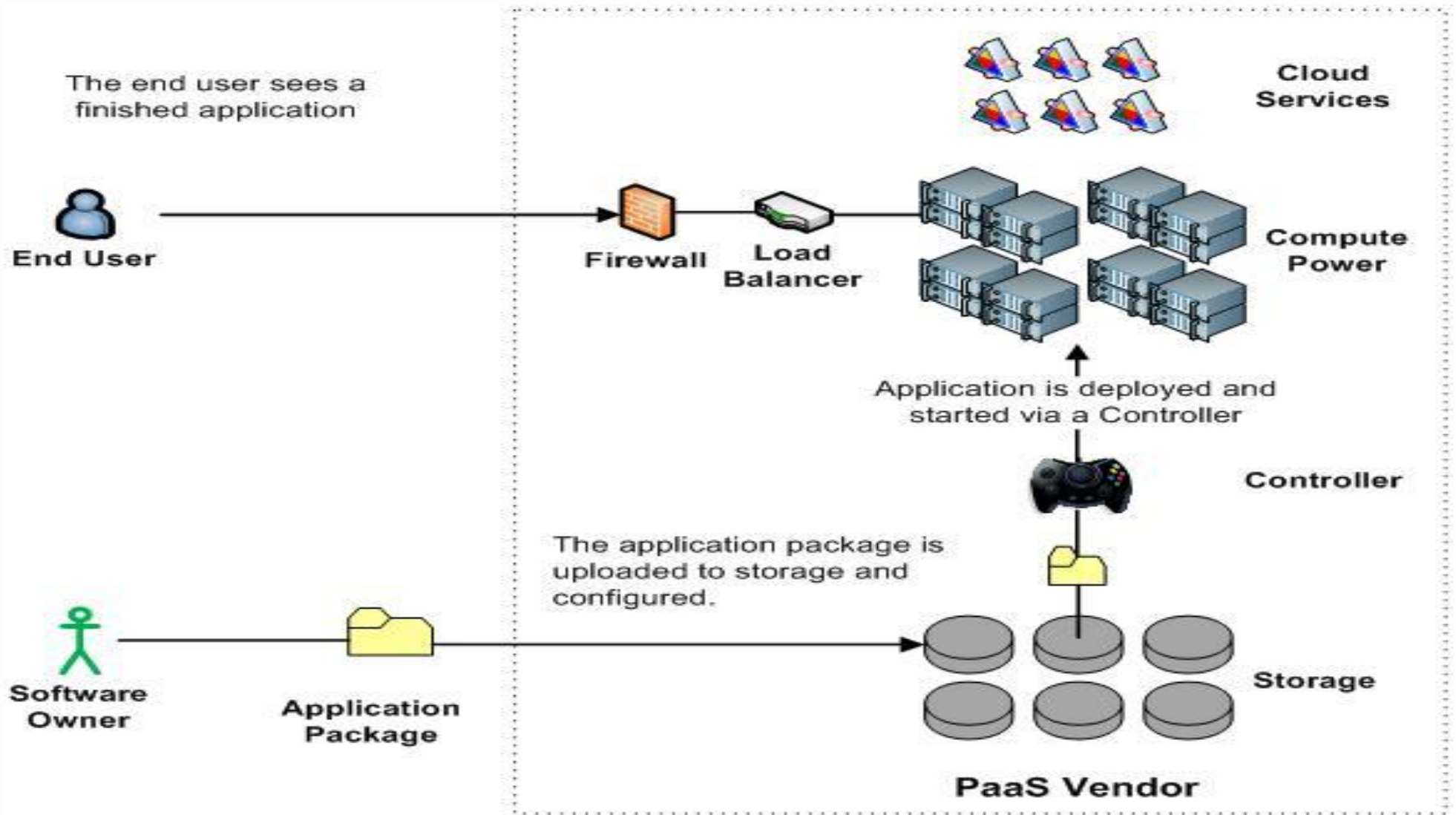
Virtual Machine

Storage

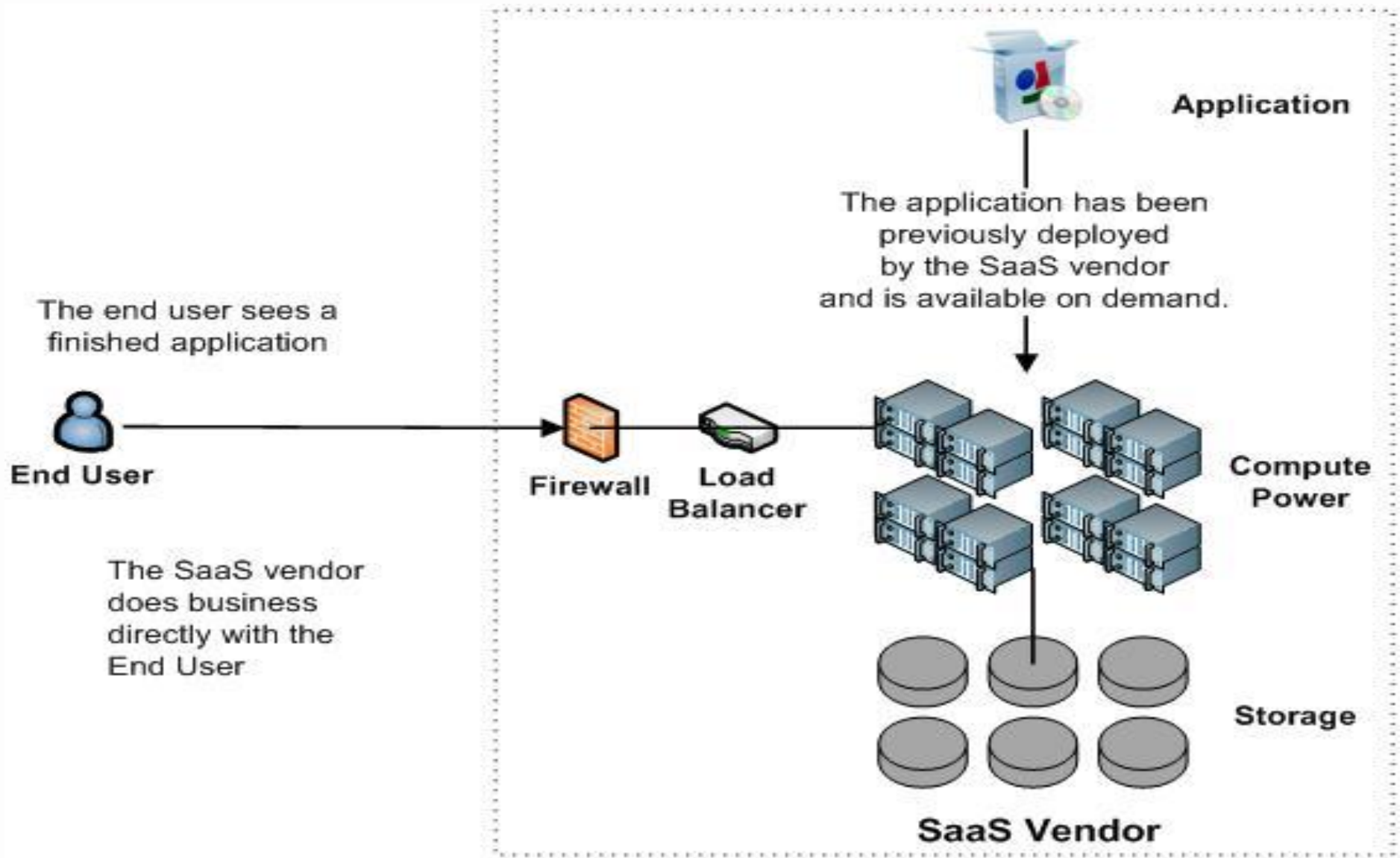
IaaS Vendor

<http://www.keithpij.com/Home/tabid/36/ctl/Terms/Default.aspx>

PaaS



SaaS



Different ways to 'SaaS'ify

SaaS

SaaS

SaaS

Shared
Infrastructure

Platform As A
Service

Infrastructure
As A Service

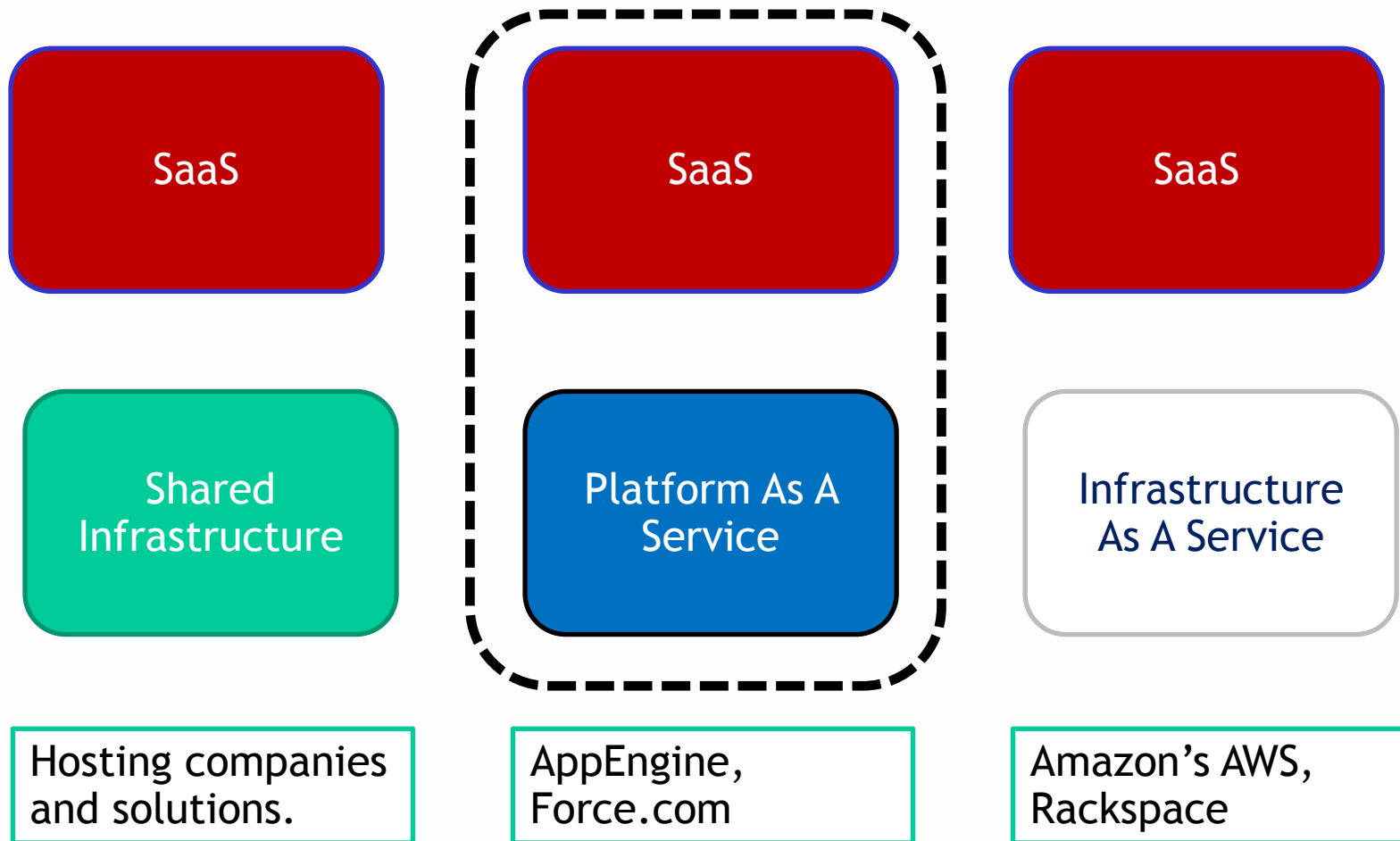
Hosting companies
and solutions.

AppEngine,
Force.com

Amazon's AWS,
Rackspace



Different ways to 'SaaS'ify



Advantages of PaaS

- Developers can focus on development and innovation without worrying about the infrastructure.
- No up-front investment in hardware and software means lower risk.
- Developers only need a PC and an Internet connection to start building applications.
- The PaaS provider manages upgrades, patches, and other routine system maintenance.
- PaaS upgrades will not break the applications running on it.
- Obvious Advantages of Cloud i.e scalability, reliability etc



Real Life!!

- Application already developed in pre-Cloud days.
- Developed for Desktop Environments .
- Depend on specific libraries that PaaS solutions do not provide.
- Using PaaS would require re-writing major chunks of code.
- Problems with re-write:
 - Existing Code has been stabilized over months and years.
 - Extensive testing needs to be done.
 - Time to Market takes a drastic hit.
 - Specific library support might be missing.

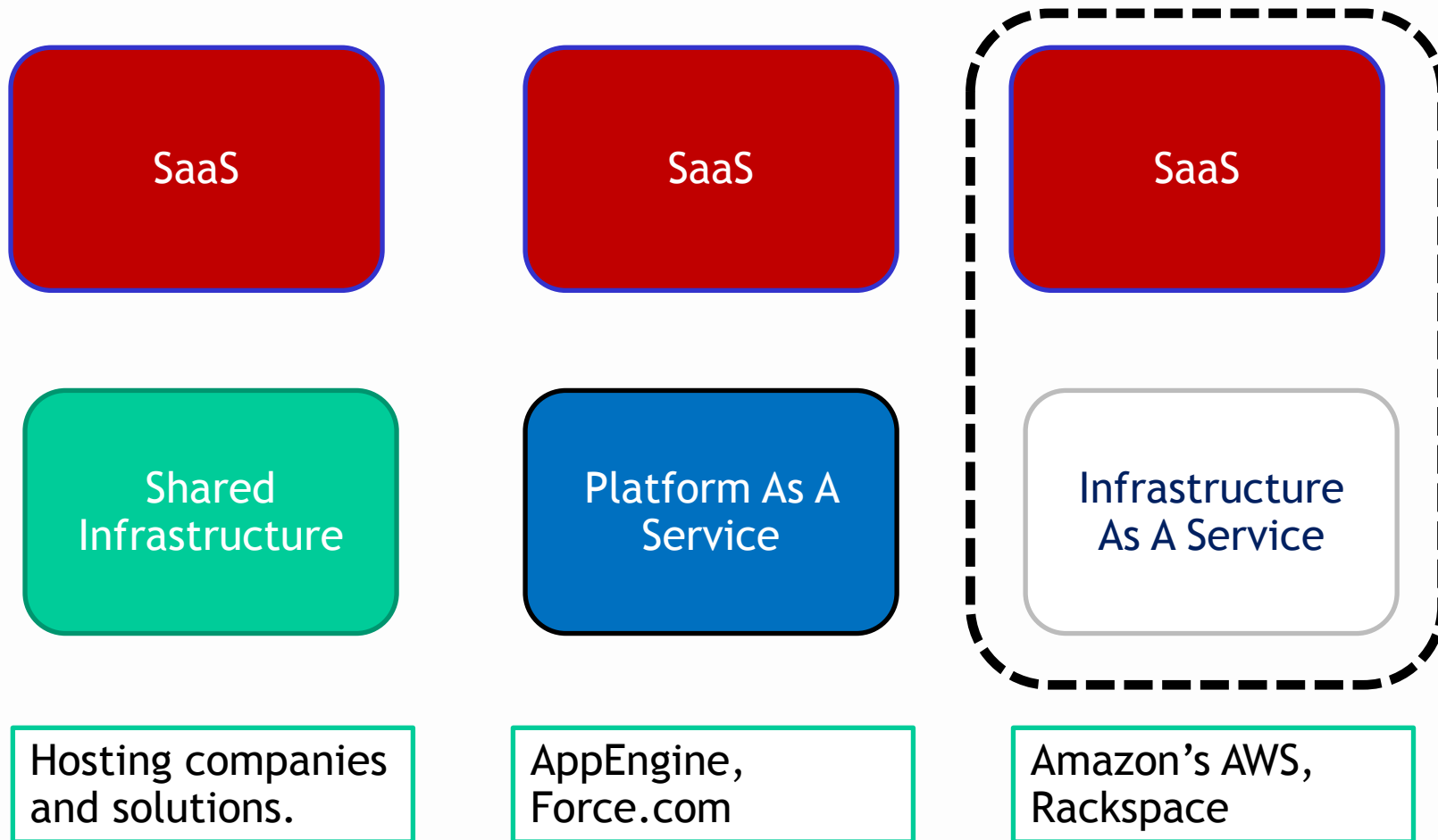


Real Example of Application

- Code written entirely in C/C++ Language.
- Using math libraries for doing complex calculations.
- Results generated in terms of Graphs and Charts.
- Desire to have wider customer reach.
- Want to hook up this application to a portal.



Different ways to 'SaaS'ify



Advantages of IaaS

- Well-tested environment available.
- Directly deploy the existing application onto the Cloud Server with no modifications.
- All required dependencies can be resolved easily by installing required libraries.
- Minimal modifications to convert the existing solution to Client-Server model.
- Easily provide a front-end from web portal or such.

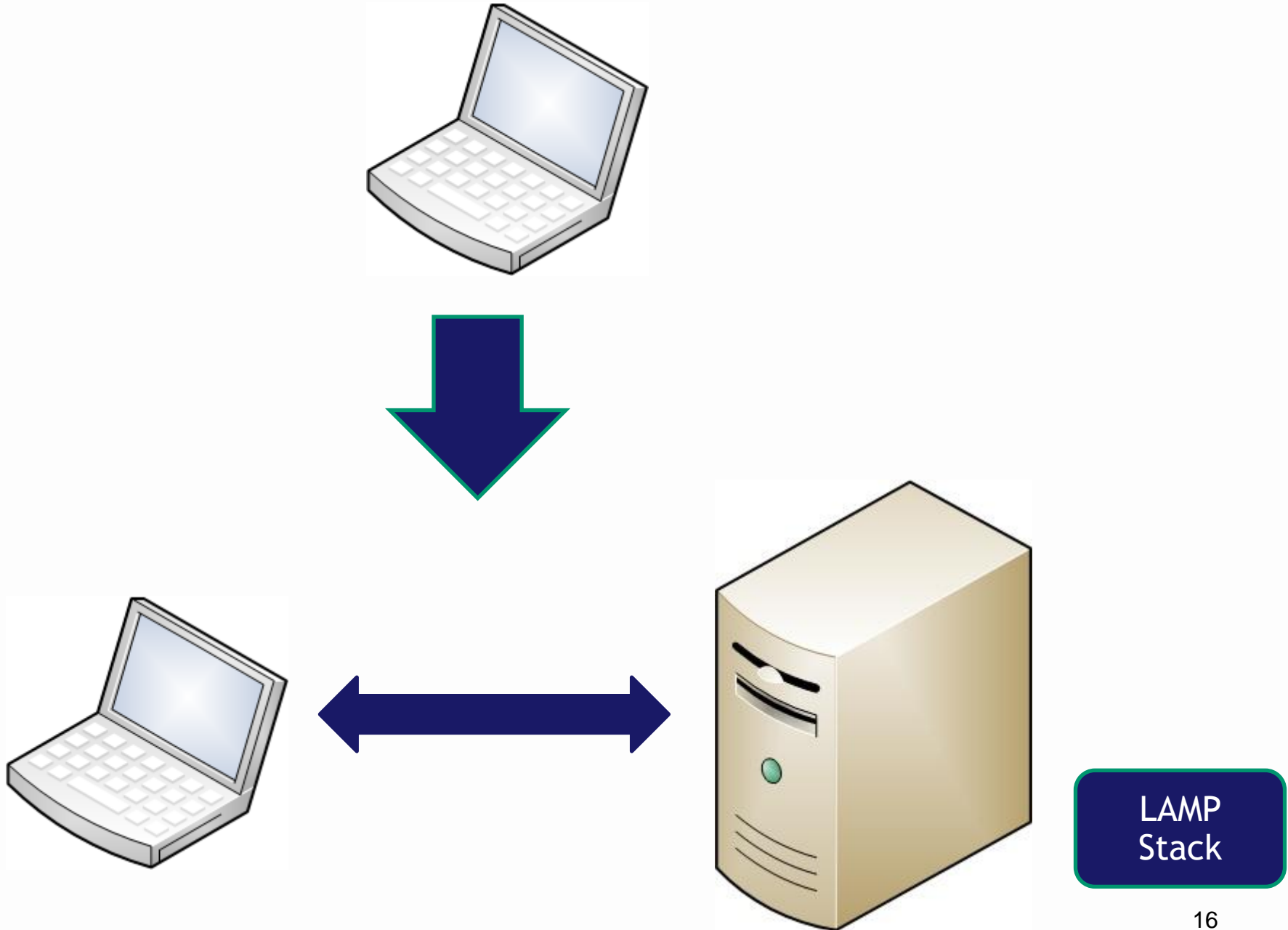


Using IaaS to SaaS'ify the App

- Scenario : Stand-alone Desktop Application
- Steps involved are:
 - Convert stand-alone application to server-application model.
 - Introduce Cloud Element.
 - Add business logic via Cloud
 - Link Portal /outside world interface to this system.
 - Test!!



Client-Server Architecture



Cloud Component



Web Server



Amazon EC2
Cloud

Amazon Web Services:

1. Elastic Computing.
2. Simple Queue Service (SQS)
3. Simple Storage Service. (S3)
4. Automatic Scaling. (AS)



17



Using Amazon Web Services

- Elastic Compute (EC2)
- Simple Queue Service (SQS)
- Simple Storage Service (S3)
- Automatic Scaling (AS)



Amazon Simple Queue Service(SQS)

- Distributed Queue System enabling web services applications to communicate.
- Queue acts as an unlimited buffer - No Producer-Consumer Problem.
- Multiple writers and readers
- Variable message size
- Unlimited queues and messages
- Access control
- Redundant infrastructure



Amazon Simple Queue Service(SQS) API

- `CreateQueue` - Creates a queue with specific `QueueName`.
- `SendMessage` - send message (PUSH)
- `ReceiveMessage` - receive message from queue (PULL)
- `DeleteQueue` - Delete a queue (All messages in the queue lost)



Amazon Simple Storage Service(S3)

- Useful for file serving (multimedia, documents) reducing load on compute servers.
- Storage for the internet.
- Makes web-scale computing easier.
- Data is stored in buckets.
- Data is stored or retrieved using a key, value pair.
- Simple REST/SOAP Apis to retrieve and store data.
- Access Control.



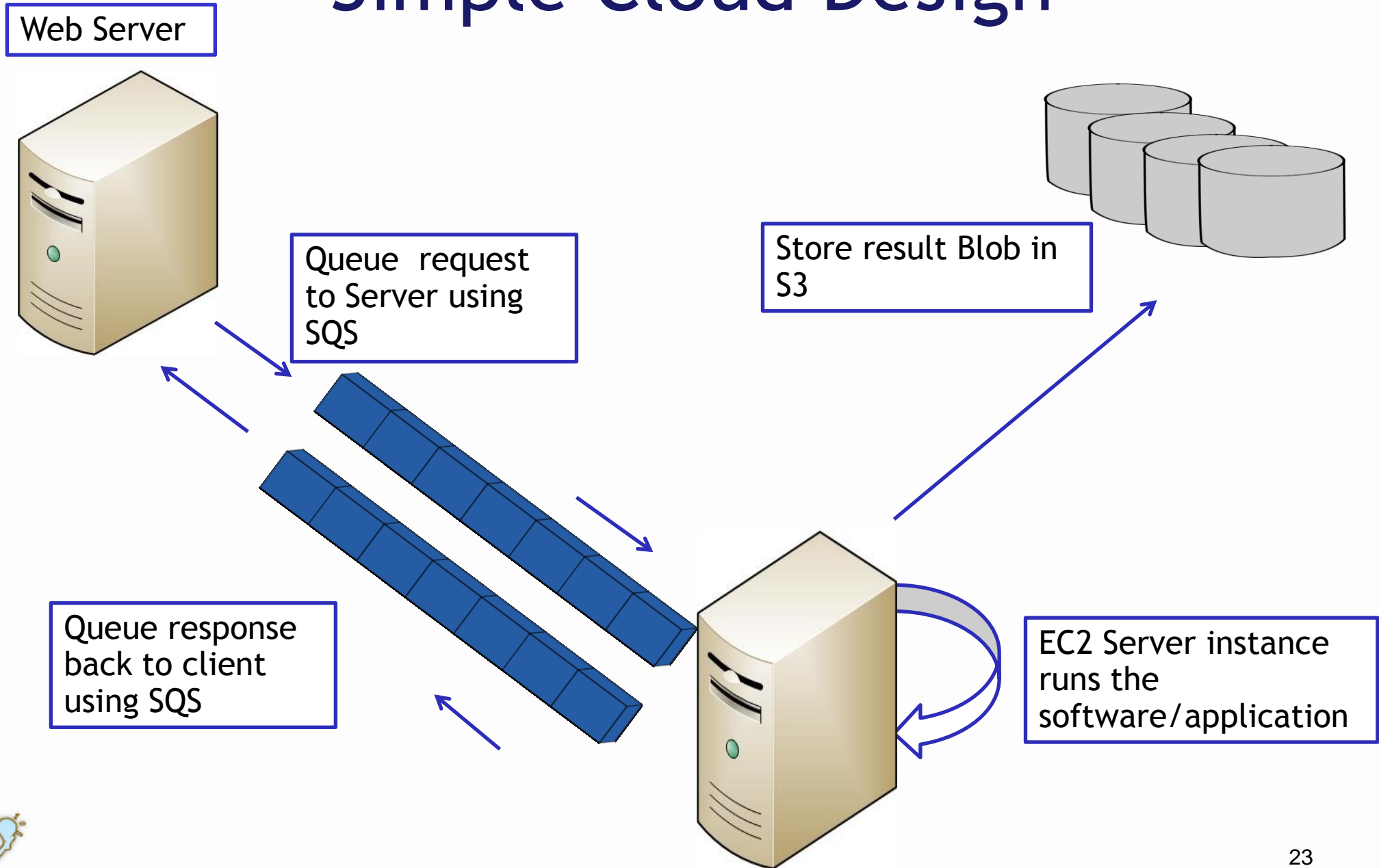
Amazon Simple Storage Service(S3)

basic APIs

- PUT Bucket - Creates a new bucket
 - Creates sampleBucket.s3.amazonaws.com
- GET Bucket - Lists information about bucket.
- PUT object - put object into the bucket
- GET object - retrieve object
- DELETE object - Deletes the object
- DELETE bucket - delete associated attributes or the buckets themselves



Simple Cloud Design



Improving Business Logic

- Auto - scalability to load balance .
- Multiple classes of users.
- Top Class user does not have to wait.
- Lower class users' request is queued.
- Extension : Multiple queues to service multiple classes



Amazon Automatic Scaling(AS)

- Automatically add compute capacity when application usage rises and remove it when usage drops.
- Predefined triggers and thresholds.
- Triggers are based on parameters like bandwidth usage or CPU Utilization.
- Triggers are based on data collected from Amazon's CloudWatch Monitoring Tool.
- Works across multiple Availability zones.

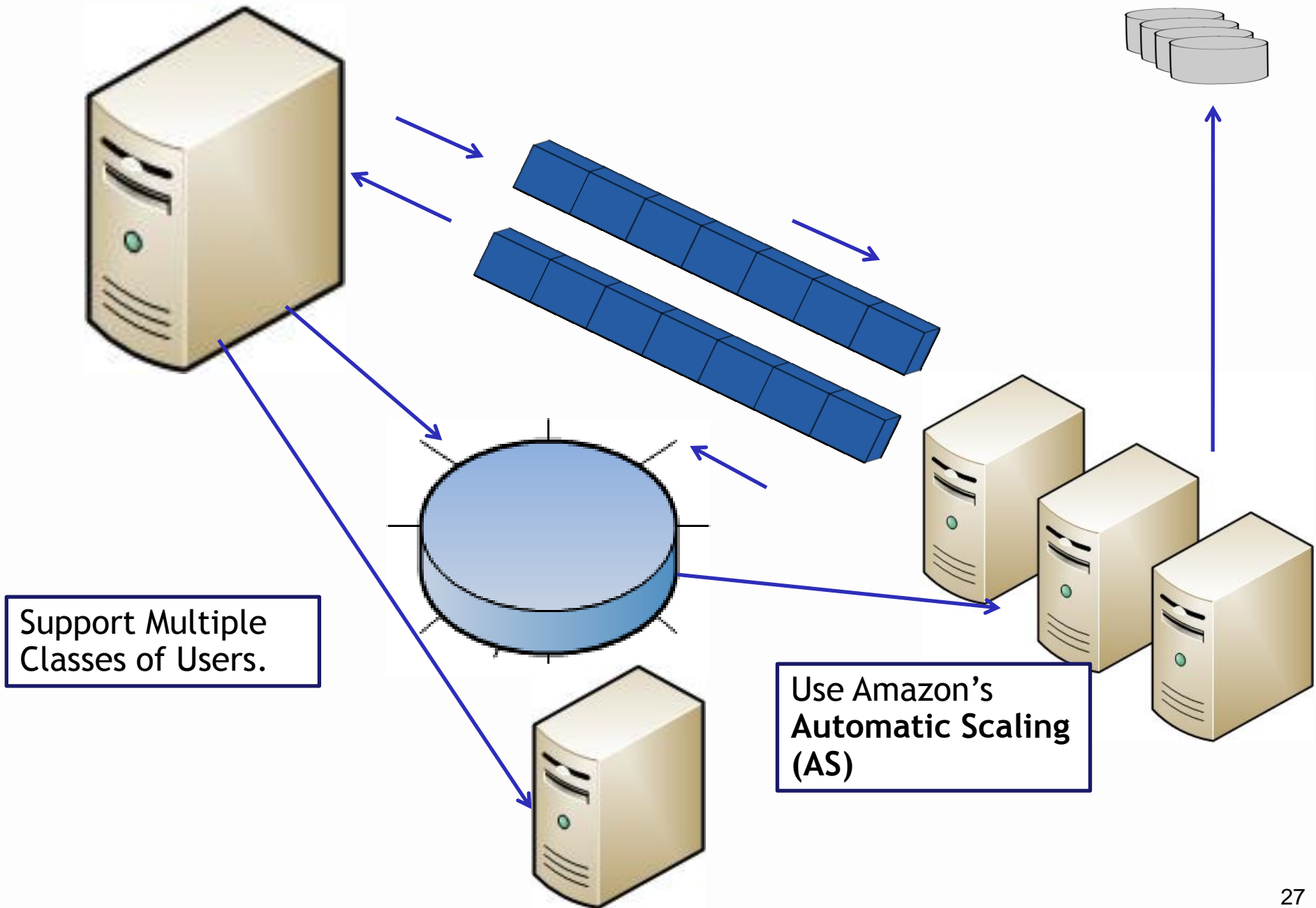


Amazon Automatic Scaling(AS) API

- `as-create-launch-config`- Create a group of EC2 instances of which the application will run.
- `as-create-auto-scaling-group` - This call sets the parameters that governs when and how to scale up and down an Auto Scaling group.
- `as-create-or-update-trigger` - Create the triggers to indicate when to scale up or down.



Improving Business Logic



Support Multiple Classes of Users.

Use Amazon's Automatic Scaling (AS)



Costs!

- Cheapest EC2 Linux Instance - \$0.085/hour
- Amazon's Auto scaling is free. Charges only for using Amazon's CloudWatch.
- Amazon 's SQS -
 - No charges for the first 100,000 Amazon SQS Requests.
 - \$0.01 per 10,000 Amazon SQS Requests
- Amazon's S3 -
 - Storage Used: \$0.15 per GB-Month of storage.
 - Network Data Transferred: \$0.20 per GB of data transferred.



Reducing Vendor Tie-in

- Reduce dependency on vendor specific services.
- Replace SQS with Apache ActiveMQ
- Replace CloudWatch with Hyperic
- Implement your own Auto Scaling Logic.



Conclusion

Effectively deploy your application without major modifications.

Run the application in original environment itself.

Add some more useful business logic to service different class of users.

Obvious Advantages of Cloud Computing applicable

- Pay only for what you use.

- Scale up/down based on load

- Design to make system more robust



Questions and Comments

