

# **Cloud Computing and Its Impact on Software Licensing**

By Gretchen Kwashnik  
&  
Jim Cecil

January 25, 2012

# What is Cloud Computing?

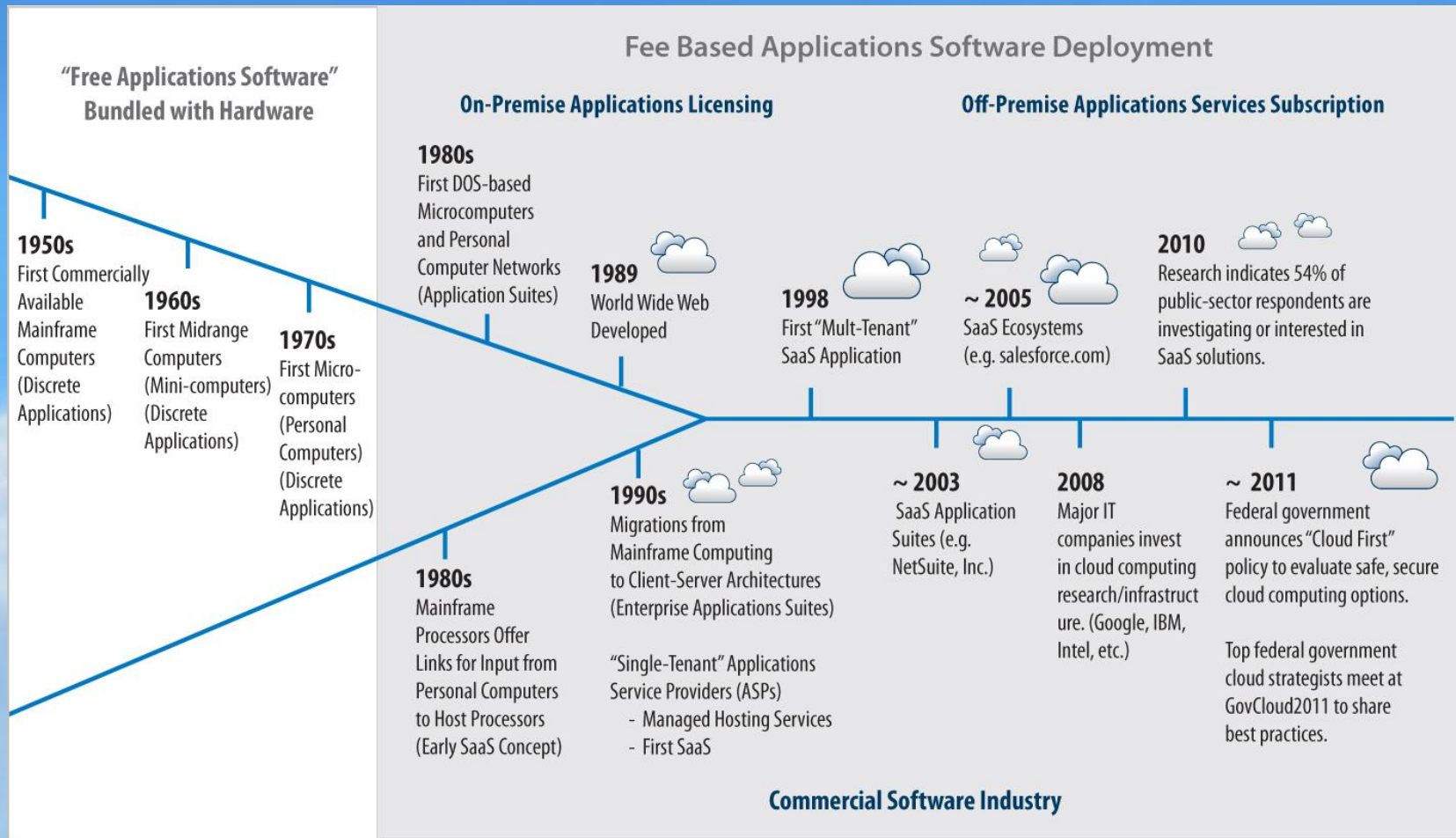
“Cloud computing is a model for enabling:  
... on-demand network access  
... to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services)  
...that can be rapidly provisioned  
...with minimal management effort or service provider interaction.”

*National Institute of Standards and Technology (NIST)*

# Discussion

- Many of you have deployed software in a cloud
- Do you have anything to add to the definition?

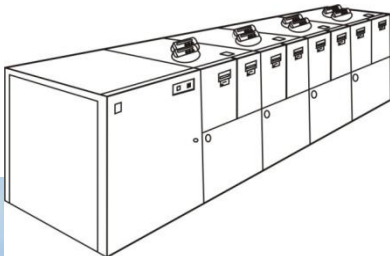
# Evolution of Cloud Computing



# Changing Characteristics

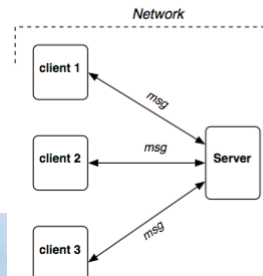
## Mainframes

- Need collaboration for service
- Limited access
- Single resource
- Hard to scale
- Manual optimization



## Client Server

- Need collaboration for service
- Some network access
- Multiple resources
- Broader scalability
- Optimization tools

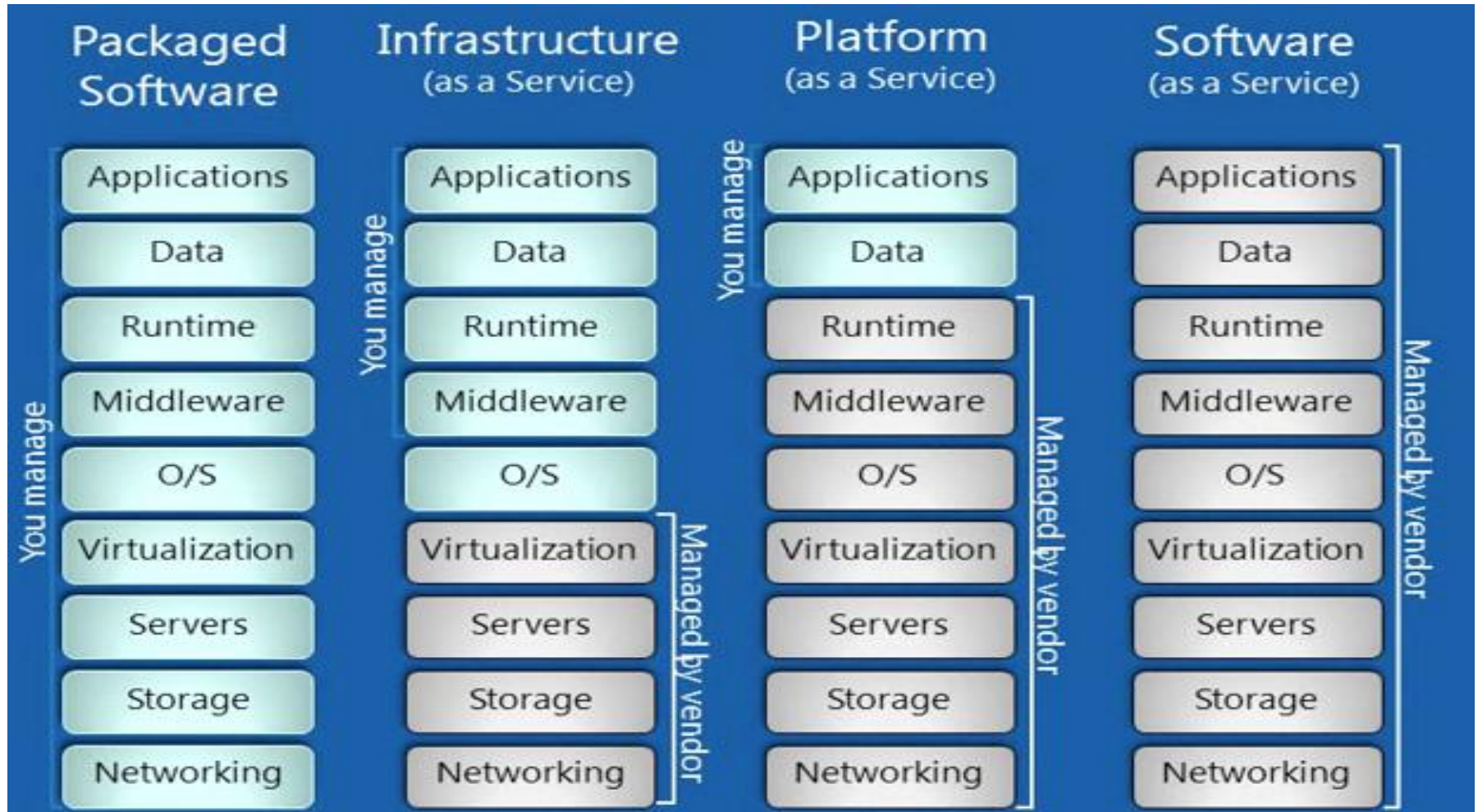


## Cloud

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Automatic optimization



# Cloud / Virtualization Models



# Cloud Deployment Models

## Public

- Off premise at provider
- Used by the general public
- Users' concerns and purposes vary

## Community

- On or off premise
- Used by multiple related organizations
- Users share the same concerns

## Private

- On or off premise
- Use limited to a single organization
- Used by various business units

## Hybrid

- On or off premise
- Use is determined by each cloud
- Users' concerns and purposes vary

# Audience Participation

- **Ask about cloud deployments**

- Who has participated in procuring or deploying apps to a cloud?
- What level (IaaS, PaaS or SaaS) and type (public, private, community or hybrid) of deployment models were utilized?



# Forecast

- About 1/3 of US organizations are using cloud computing
- 21% average annual savings for applications moved to the cloud
- 20% of businesses will own no IT assets by 2012, a trend driven by cloud computing
  - No hardware, no licenses, no networks, no security software – just a browser with an internet connection

# Why Move to the Cloud?

- **Cost Reduction**

- Reduced cost of building and maintaining infrastructure
- Specific use scenario should be analyzed to determine whether or not a cloud delivery model may result in greater cost avoidance

# Why Move to the Cloud?

- **Speed and Flexibility**

- Reduction of complex procurement and deployment cycles
- Self-service provisioning for adding functionality and new users
- Ability to scale up and down as needed

# Why Move to the Cloud?

- **Greater Mobility**

- Access your software and inherent data from any device that has access to the Internet

- **Easier Collaboration and Integration**

- Pre-established application programming interfaces (APIs) with popular applications for sharing of information to drive results

# Why Move to the Cloud?

- **Heightened Security**

- Provider's often have cutting-edge, secure, and traceable data access trails
- Most cloud servers will be hosted in physically secure data centers with strict access control for their own staff and no access for unauthorized personnel

# Audience Participation

- **Discuss some of the results of cloud deployments**
  - For those who have deployed apps in a cloud, what has been your experience with
    - cost savings
    - Speed & flexibility
    - greater collaboration across entities
    - security

From here the presentation will shift focus and talk about the key contract considerations for your application in the cloud - SaaS

# Traditional Licensing versus SaaS

## Traditional

- One time upfront payment
- License to use in perpetuity (no term)
- Customizations may be supported
  - Ability to have custom code
  - More influence to ask for a special order from their menu items
- Normally hosted on-premise by organization

## SaaS

- Subscription based pricing
- License to use only while subscription is current (term)
- Customizations normally not acceptable
  - Focus on standard offering
  - Little to no influence to order items not on the menu (only their standard items)
- Normally hosted off-premise by provider



# Music as an Example

## Traditional – CD

- On premise
- Need your own equipment to play music
- Accessed only if in possession of CD and onsite
- License to listen in perpetuity

## Hosted – iTunes

- Off premise
- Use provider's portal to play music
- Accessed from anywhere there is internet access
- License to listen in perpetuity

## SaaS – Pandora

- Off premise
- Use provider's portal to play music
- Accessed from anywhere there is internet access
- License to listen only while subscription is current

# Rights to Use the Service

- Subscription based
- Normally includes the right to use the service plus support and maintenance, hosting and storage fees
- Extend the “internal use only” definition
  - Ensure third party contractors can use the service
  - Contract example: Users may include but are not limited to employees, consultants, contractors and agents of Organization, or third parties with which Organization transacts.
- No hidden fees

# Payment Terms

- Monthly or annual payment schedule
- Locked in for an extended term
  - Achieve greater discount by locking in longer
- Minimum number of user commitments
  - Payment usually starts upon contract signing even for a sandbox or other non-production use
  - Negotiate 2+ user types: read only and full
  - Ensure you have the administrative ability to add and remove users without provider's assistance
  - Fees are not based on actual usage

# Future Fees

- More users
  - Volume tiered pricing
    - Pure tiers and not step-through
- More storage
  - Per user or instance
- “New” functionality
  - Substantially similar or enhancement
  - Part of core product or provider’s original intent

# Audience Participation

- Who has entered into a SaaS or other subscription based license agreement?
- Were the applications hosted by the Publisher or a third party?
- Were the applications deployed in a cloud with virtualization or a traditional hosting environment?

# Functionality Matrix

## Contract Example:

1. Standard: Included in current version or a configuration
2. Custom: Needs to be developed
3. Future: On current 1-2 Year Roadmap
4. NA: No plans to develop

In the “3rd Party” column please designate whether or not such IP for the functionality is owned/controlled by a third party.

# Functionality Matrix

#	Functionality	Type	3 <sup>rd</sup> Party	Comments
<b>Applicants</b>				
1	Support for the following browsers: IE, Firefox, Chrome and Safari.			
2	Support for Mac users.			
3	If a drop down list is being used for a particular question, the applicant still has the option to type in an answer (if the drop down list is not helpful).			
4	Edit the application after it has been submitted.			
5	See all job postings ever applied for.			
6	My application status page.			

Items listed before are things to watch out for upfront. Once you are ready to contract. These are the key agreements that will set the ground work for moving forward.



# Underlying SaaS Agreement

## Information Security and Confidentiality

- Multi-tenant vs. single tenant environments
- Perform an onsite risk assessment upfront
- Secure code development standards – ability to test
- Who can access your data - user permission roles

# Underlying SaaS Agreement

## Data Rights and Responsibilities

- Data at rest and data in transit
- Obtain data flow diagram and levels of encryption upfront
- Destruction and return of data requirements

# Underlying SaaS Agreement (con't)

- Intellectual property indemnification
- Rights to request software customizations - cost and ownership of any such software customizations (work product)
- Disaster recovery and business continuity
- Termination conditions and cooperation

# Addendums to the Master Agreement

- **Support and Maintenance Agreement**
  - Levels of support available and severity of issues
  - Response and resolution times
  - Remedies
- **Service Level Agreement**
  - Uptime and performance levels
  - Remedies
- **Professional Services Agreement**
  - Terms and conditions for engagements with provider
  - Training options

# Support and Maintenance - Product

- Identify severity levels
- Identify response and *resolution* times
- Identify provider's escalation process and contacts
- Notice period before all releases or ability to “opt-out”
- Ensure support for third party integrations are included
- Define remedies for failed turnaround times
  - How to submit a claim for a refund or credit
  - Ability to terminate the entire agreement without further liability under certain circumstances

# Response Time Examples for Support Calls

Issue Severity	Response Time to Acknowledge Issue	Response Time to Fix Issue
<b>Level 1 (Low)</b> Does not have significant impact on users.	Return call or email within 8 hours.	Provide fix within 30 days.
<b>Level 2 (Moderate)</b> Causes some user issues, but most processes are functional.	Return call or email within 4 hours.	Provide fix within 5 days.
<b>Level 3 (High)</b> Significant impact on system use.	Return call or email within 1 hour.	Provide fix ASAP – 24 hours or less.

# Service Level Agreements - Hosting

- Define uptime calculation and planned maintenance
  - Identify the provider's standard maintenance window
  - Request at least 48 hours' notice
- Define performance level and testing mechanism
  - Content load ping test
  - Third party tools
- Confirm what monitoring and alerts are available
- Define remedies for failed uptime or performance
  - How to submit a claim for a refund or credit
  - Ability to terminate the entire agreement without further liability under certain circumstances

# Measuring the SLA for Maintenance

System Availability Example – 99.9%

Criteria	Measurement	Comments
Minutes in a 30 day month.	43,200 minutes.	
Planned down time (assume 6 hours).	360 minutes.	This is a standard amount of time for monthly system maintenance.
Remaining minutes for scheduled up-time.	42,840 minutes.	
SLA	99.9%	This is a moderate standard; 5 nines (99.999%) is high.
Minutes of expected up time.	42,797.16 minutes.	
Allowable minutes of unplanned downtime.	42.84 minutes.	Little time for unplanned down time.
Penalties.	Varies.	Usually a credit is given for missing the SLA.



# Importance of Uptime Percentage

Based on 90 rolling days with 3 releases (each release totaling 6 hours)

Quantify the uptime and remedies required based upon the criticality of your cloud application

95.00% =

107 hours of downtime - red

98.00% =

43 hours of downtime - red

99.00% =

21 hours of downtime - red

99.90% =

2 hours of downtime - yellow

99.96% =

51 minutes of downtime - green

# Termination Language Example

If Customer experiences:

1. More than 6 unexpected downtime hours resulting from 3 or more non-consecutive service interruption events during any rolling 30 calendar day period; or
2. More than 24 consecutive unexpected downtime hours due to any single event

Customer shall be allowed to immediately terminate the Agreement and any Order Forms with Provider, and shall not be liable for any future committed fees beyond the termination date.

# Audience Participation

- Who has experience managing response time and up time SLAs?
- Discuss variations you have seen on SLAs, penalties, termination rights, etc.

# Final Take-Aways

- Cloud computing is a broad and evolving term
- Each software application and use has different contractual concerns, including data risks and ownership rights
- Include a functionality matrix as part of your selection process and final contract
- Ensure the Service Level Agreement has teeth
- Document escalation process with the provider
- Plan for the worst – ensure cooperation during a data breach or termination

# Resources

- [Enterprise Software Initiative's SaaS Toolkit](#)
- DoD ESI White Paper: Best Practices for Negotiating Cloud-Based Software Contracts

# Questions

# Your Speakers:

- Gretchen Kwashnik,  
[gretchenkwashnik@aol.com](mailto:gretchenkwashnik@aol.com), 302-650-3932
- Jim Cecil, james.cecil@osd.mil, 571-372-4658