eSentio

CIO Roundtable Atlanta

Preparing your Development Team for Cloud Technologies

May 16, 2017

Agenda

- Welcome
- Upcoming ATL Roundtable:
  - June 20, 2017
- Host Introduction:
  - Jamie Usher, Kilpatrick Townsend & Stockton LLP
- Today's Topic:
  - Preparing your Development Team for Cloud Technologies
- Participant's Expectations:
- Presentation/Discussion:

Presentation Agenda

- Assumptions and Considerations
- Staffing Considerations
- Education required for your development team
- Learning new APIs
- Where to begin and completing the development lifecycle
- Working in hybrid cloud
- Managing the move to the cloud for your custom apps
- Development considerations for Cloud based DMS
- Building apps for mobile and remote purposes
- Outsourcing
- Q&A

So you want to move to the cloud…

- Assumptions made for today
  - You know *why* you need to migrate
  - You have a pretty good idea as to *what* to migrate
  - You have developers on staff or on retainer
  - You have custom apps/scripts/reports and 3rd party tools
- What is involved with a Migration?
- When to Migrate with a phased approach?
- Where to Migrate? Public vs. Private Cloud
- What about all of the custom apps and 3rd party tools (gap fillers) you purchased? How will they be affected?
- Are your developers ready and do you have the staff?
- …Let's discuss this hour!

▷ Why and What?

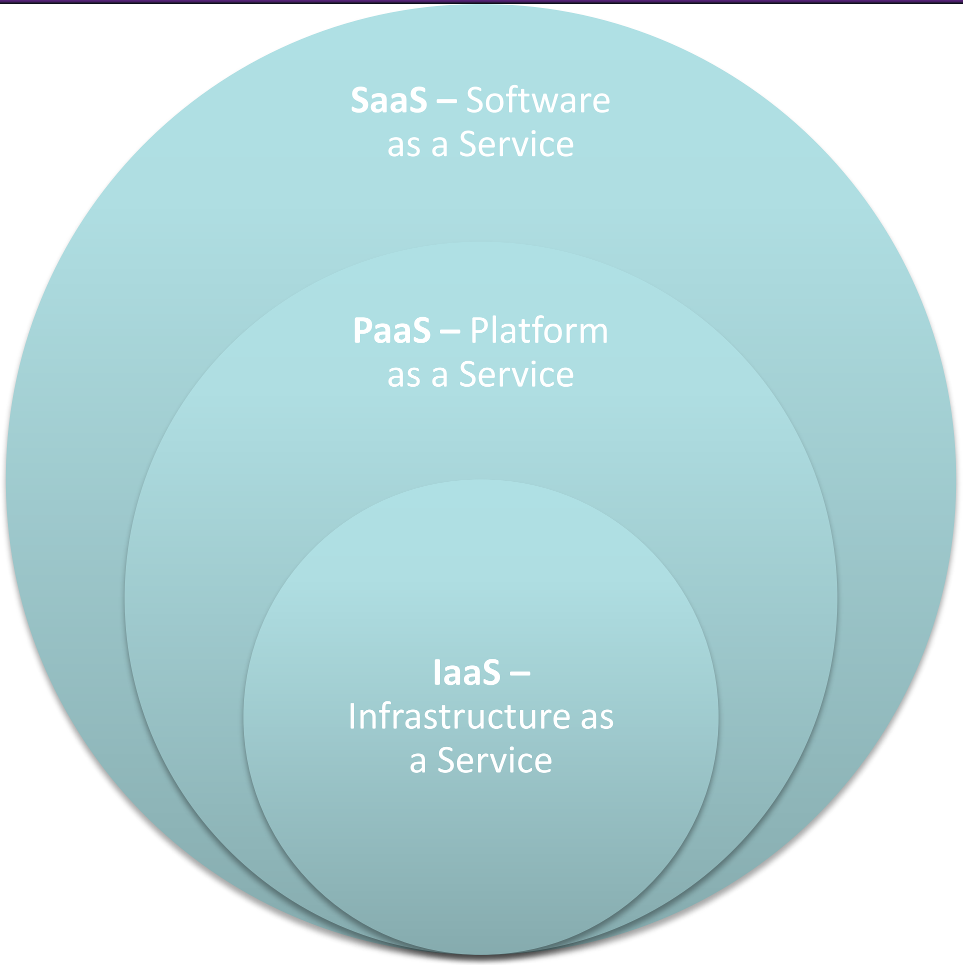
- **Why firms move to the Cloud?**
  - Cut costs
  - Differentiation of services for clients
  - Mobile app development and scalability
  - Hardware and infrastructure complexity, cost
  - Time to move apps to the web
- **What apps are moving to the Cloud**
  - Standalone apps:
    - Exchange (through Office 365)
    - SharePoint (through Office 365)
    - DMS (through iManage Cloud or NetDocs)
    - CRM (e.g., MS Dynamics/CRM)

- Challenges with rebuilding apps for the Cloud
  - API changes
  - IP address dependencies
  - Security
  - New development standards
- Migrate vs. Redevelop

eSentio

A little education, not too technical

Cloud Technologies


SaaS – Software as a Service
PaaS – Platform as a Service
IaaS – Infrastructure as a Service

SaaS – Software as a Service
Includes: Applications and Data
System Types: Monitoring, Content, Collaboration, Communication, Finance
Examples: DMS, Email and collaboration, CRM
Description: is a business application created and hosted by a service provider. In this model, users do not have any action on the environment and the vendor or service provider takes care of the entire infrastructure and also the application.

PaaS – Platform as a Service
Includes: Runtime, O/S, Middleware
System Types: Storage, Identity, Runtime, Queue, Databases
Examples: Users (mostly developers) are able to create new online applications in platforms provided by the PaaS provider. Google App Engine and Windows Azure are examples of a PaaS service model.
Description: A combination of IaaS and a set of middleware, software development and deployment tools that allow the company to create, develop and deploy on a cloud up to speed.

IaaS - Infrastructure as a Service
Includes: Networking Storage, Virtualization Servers
System Types: Compute, Block Storage, Network
Examples: Amazon Web Services (AWS), Microsoft Azure, Salesforce.com, Google Compute Engine
Description: The delivery of components such as hardware, software, data center space, networking and storage.

Following Standards Considerations

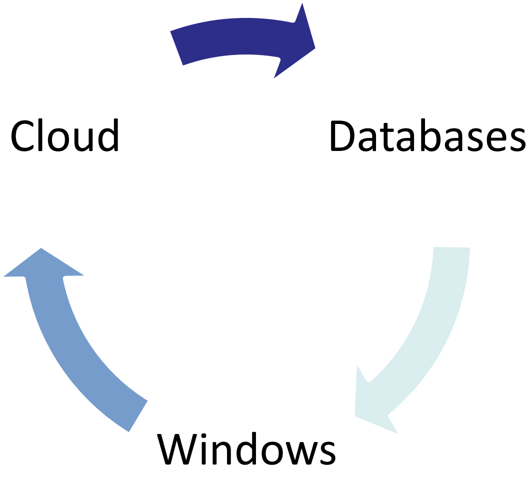
- **Security** - Authentication based on Web standards will ensure compatibility
  - Azure AD
  - Oauth (Open Authorization)
  - SAML (Security Assertion Markup Language)
- **Browser Standards**
  - IE11, Edge, Chrome, Safari (FireFox is no longer following the standards)
- **Service-Oriented Architectures** – IaaS, PaaS, SaaS
- **WEB APIs**: Refers to both sides of computer systems communicating over a network: services offered by a server, API offered by the client such as a web browser. The server-side portion of the web API is a programmatic interface to a defined system. Two most dominant are SOAP and REST.
- **SOAP (Simple Object Access Protocol)** is a messaging protocol, apps run on disparate operating systems (such as Windows and Linux) to communicate using HTTP and XML.
- **REST (REpresentational State Transfer)** – The standard. It's an architectural style, unlike SOAP which is a standardized protocol. REST makes use of existing and widely adopted technologies, specifically HTTP, and does not create any new standards.

API (Application Programming Interface)

- The API is a just a doorway into an application…a set of protocols that allow us to add or read data from another application
- With Cloud development, we often use the standard, REST API
- **3rd Party Apps in the cloud may have their own APIs which you can use to collect data**
  - iManage has their own for DeskSite/FileSite and now Work 10 Professional
  - NetDocs as well
- A large portion of custom Legal apps relies heavily on API development, which makes planning Cloud development crucial

Hybrid Environments

- **Windows Apps**
  - MS Office
    - And Word, Outlook add-ins
  - Accounting apps (Aderant Expert, Elite 3e)
  - Integration apps
  - Standalone apps
- **Cloud Apps**
  - DMS
  - CRM
  - Case
- Custom apps will need to share data with both types
- May require different APIs and methods to get the data from Windows to Web/Cloud



eSentio

Moving to the Cloud for Developers

Migration – "R"

- **Replace**: To discard an existing Windows application and switch to a SaaS model
- **Refactor**: To run an application on the cloud provider's infrastructure, requiring application code or configuration changes
- **Revise:** To modify or extend the existing codebase to modernize the application (changes range from minor to significant), then deploy to the cloud using rehosting or refactoring
- **Rehost:** Also referred to as "lift and shift," mass migration means redeploying an application to a different hardware environment, then changing the application's infrastructure configuration to support the cloud
- **Rebuild:** To fully redevelop or rearchitect an application on the cloud infrastructure provider's platform.
  - Can be expensive, can be time consuming, but ultimately where you will need to go.

Gartner - 1684114

▷ Staffing Considerations

- Your developers know .Net *Windows* development, but what about .Net *Web* development?
- Knowledge of API development for Web
- Understanding of Cloud system security
- MS SQL Azure – not quite the same as SQL Server
- Understanding hybrid development (On-prem/Cloud)
- Maintain existing app development while working on the new Cloud projects
- Support

Education Required

- Web/Cloud based development
- Education in REST APIs
- Learning about MS Azure services and Amazon AWS offerings
- Understanding interoperability between apps
- The different layers of Cloud development and systems (IaaS, PaaS, SaaS)
- Learning security models for the Cloud
- Who educates and how to teach the team

▷ Outsourcing

- Sometimes your development team may just not be enough
  - Maybe they don’t have the knowledge needed
- Testing and Support considerations
- The existing team may be too busy supporting the existing app versions
- Outsourcing is a great way to keep two different lines of code going with constant knowledge transfer
- Take ownership of the source code and retain all backups
- Introduce development lifecycle standards, such as Agile/SCRUM
  - Sprints, standups, constant communication

Development Lifecycle

- Decide to rewrite your app in the Cloud for Web
- Financial planning
- Requirements gathering and research
- Establish the development and testing environments in the Cloud (accounts vs. environments)
- Design the move for users from on-premises to the Cloud in a phased approach
- Support
- Final cutover in phases
- Maintenance releases

▷ Moving Custom Apps to Cloud

- **Take inventory of your apps**
  - SQL Scripts, Reports, Integrations, Standalone, Modifications of 3rd party apps, Macros, Templates
  - Ensure you have access to all source code
- **Any apps that are affected by a move to the Cloud**
  - Should be evaluated to see if it needs to be rewritten as a web app
  - Does it need to be turned into a service (i.e., no user involvement)?
  - Is it standalone or an add-in to an existing app?
  - If add-in, does the new app support the same API or now REST?
  - Costs? Security?
- **Standalone vs. 3rd Party apps in the Cloud considerations**
  - Example - DMS Monitoring
  - Mass import of Email

DMS in the Cloud

- **Moving to iManage Cloud**
  - Synchronizing your local accounting system data to the Cloud
  - Add-ins for DeskSite/FileSite, do they need to be rewritten?
  - Going to Work Professional 10 – learning the new REST API library
  - Database specific scripts, reporting, and tools
- **Moving to NetDocuments**
  - Evaluate all of your existing add-ins for your local DMS
  - Learning a new REST API library
  - A new way to connect to MS Office
- An ALM/Thomson Reuters survey indicated the biggest challenges for law firms to shift computing resources to the cloud are squashing **security** concerns (86%) and providing firms more control over their **data** (61%).

Building Apps for Mobile and Remote

- **Mobile**
  - Users demand mobile clients to match desktop clients
    - Ex: iManage 10
  - Mobile apps connect to Cloud differently
  - Requires different security models and considerations
  - The source application can have a different API for your custom mobile app
- **Remote Access to Cloud Apps**
  - How will your users connect to the new apps?
  - VPNs are no longer required with apps in the Cloud, necessarily
  - Do they require AD integration?
  - Biometric security for features of the app

eSentio

Next Steps for your firm

Next steps moving to the Cloud

- Start simple with the obvious apps, such as Exchange in Office 365, SharePoint Online
- **If you are on-premises with iManage, then consider all of your custom apps and 3rd party tools**
  - Stick with DeskSite/FileSite v9.x until your custom and 3rd party apps have been redeveloped for Work 10 Professional
- **If you are moving to NetDocs from iManage on-premises, look through all 3rd party apps and custom apps**
  - Ensure there is a need to recreate your custom apps with NetDocs as some tools may come with NetDocs out of the box
- **When recreating custom apps to support a new platform or feature set, the following should occur:**
  - Learn the new APIs
  - Web or Windows development
  - Establish your deployment methodology, including the phased approach while part of the Firm is still on the pre-existing system
  - Establish your testing methodology
- **Cloud platforms can be slower than on-premises. Make sure your custom solutions properly handle caching and pre-loading technologies to ensure a quick system**
  - Ex: The smallest number of trips to the SQL Server, the better
- **Understand all integration points with your custom apps**
  - Synchronization with on-premises apps

Next Steps, cont'd

- **Licensing** – should you have 3rd party tools or 3rd party add-ins to your custom apps, ensure they support Cloud based platforms
- **Support** – ensure your service providers support the new Cloud solution
  - Also ensure your development team or outsourced developers understand Cloud technologies
- **Security** – Custom software should support new Cloud security models
  - In addition, the security setup of the IaaS or the Application should support the Firm's security needs
- Ensure all Cloud housing is stored in the US data centers to avoid your data going outside the country, unless required
- App rewriting may be a requirement if the platform currently in use is outdated and not compliant with the new Cloud platform
- Developing cloud applications can be costly, but there are various tools and strategies that can help developers cut costs and optimize their cloud applications' performance. Among other steps, always perform a cost-benefit analysis of their cloud apps, have a solid understanding of their applications' total cost of ownership and deploy cost analysis tools.

Q&A