- Peter Metz, Senior Architect, KRING e-business consulting
- Owen Nishimura, Senior Cloud Application Architect at Nishimura Consulting
- Jesse Shiah, CEO of Ascentn
- David R. Stefanich, CIO of Accuride
- Bill Storage, CEO of NerveNet
- Lucas Vogel, Senior Managing Consultant, Endpoint Systems

Julian Keith Loren has spent the past 15 years tracking down challenging, mission-critical I.T. innovation initiatives and then doing whatever it took to get them delivered. He almost always wears the architect hat, but has commonly combined that with other responsibilities including strategist, analyst, product manager, project manager, programmer, CTO, and Chief Executive Trouble-maker.

Julian has recently begun giving workshops and presentations on two topics:

1. Full Innovation – A comprehensive set of easy-to-apply practices, approaches, organizational structures and cultural components geared at dramatically improving innovation effectiveness.
2. Full Agility – Hands-on methods for creating well-rounded set of highly-tailored agile practices for all aspects of the I.T. innovation cycle—including agile analysis and design methodologies.
3.

He has just completed the first "Sneak Preview Edition" of a book entitled Full Innovation Ahead!
Additional bio details can be seen at http://www.linkedin.com/in/juliankeithloren
Julian invites connections and requests from IASA and Innovation Management Institute group members through LinkedIn.

Are You Secure in the Clouds?

by Keith McMillan

Cloud computing has garnered a lot of attention recently. Whether we're talking about Microsoft Azure, Amazon S3, Google, or some other underlying platform; cloud-based applications offer us the promise of reduced infrastructure costs and access to information and services from anywhere. While these new approaches offer us new capabilities, they also offer us new challenges. As software architects we need to consider the new security issues created by this model.

Currently, there is no accepted general industry definition of what it means to be a "cloud computing application". A modest definition might be: "an application that uses Internet-based services for storing or otherwise processing your data." By this definition, even email qualifies as a cloud based application, since a typical email topology includes an IMAP or POP server that is exposed to the internet, and a client that accesses the cloud based services, allowing the user to view and manage email messages and folders. Much more ambitious applications, from photo editing to complete office suites and business work flow engines have come into being using the Software as a Service model in the last few years. However, even a relatively simple application like email should prompt us to consider how to can handle the security of information managed by the application.

Availability, Confidentiality and Integrity

Information security can be broadly grouped into three categories: availability, confidentiality, and integrity.

Availability

When we consider availability, we're making sure that information is available to authorized users when and where they need it. Cloud based applications offer us easier accessibility, since their services are available anywhere the Internet is. Availability is another question. Once our application makes use of services on the Internet, which are probably provided by third parties, we should consider questions such as:

- What happens if the company providing the solution goes out of business? Can we retrieve our information or is it locked up in such a way that it's impossible to get?
- What happens the company providing the solution refuses us access, for example, if it decides we've violated their terms of service.
- What if we decide to move our information to another service provider? How portable is the data?

- How viable is the business providing the service to us? Will they survive an economic downturn?
- Where are we in the priority order of their customers? If we have technical problems, how much pull do we have? Will we be able to get our problems resolved and how much influence will we have in the future direction of the services provided? What is the SLA and is the company able to support it?
- Who owns, from a legal point of view, the content we place in their hands?

Some availability questions that we are used to asking may now be harder to answer. We may have a very good idea within our company what the network capacity is and how our backup procedures work. Will the third party providing services tell us what their capacity is and let us review their backup procedures?

Confidentiality

When we consider the confidentiality of information we aim to ensure that only authorized users have access. With an in-house computing environment we have a pretty good idea who has access to our hardware and systems. With a cloud computing application this isn't necessarily the case. Our cloud service provider might be unwilling or unable to tell us the identity of their other customers. Those same customers could conceivably gain access to our information.

Some other considerations are:

- Where is the service hosted? Is it in a data center that has passed external audits for controls?
- What are the change control procedures associated with implementing a change to the service?
- Who has access to our information and under what kind of controls? When do developers, testers, or support staff have access to our data and for what purposes?
- Is the service or application using a multi-tenant model? If so, what controls are in place to ensure that one customer can't see or modify another customers data?
- What measures are taken to protect data when in transit through the network? Is data encrypted as it is passed from system to system?

Integrity

Integrity means ensuring that our data is consistent, is changed only by authorized users, and that these changes can be verified. In a cloud based environment, we may need to consider the security of the network. Is there a possibility of an attacker replaying, re-routing, or modifying our traffic. Even if our traffic is encrypted, an attacker might be able to block our traffic altogether. Have we placed our data in a multi-tenant application where another user can intentionally or accidentally modify it because of a flaw in the solution provider's code? Is non-repudiation a feature of the hosted system?

Security Controls

How do we as software architects address these issues? IT security practitioners organize security approaches, or controls, into three categories: technical, administrative, and physical. When we consider the issues of integrity, confidentiality and availability we should consider whether controls from one or more of these categories will help us.

Technical controls apply technological solutions to information security requirements, such as encrypting data and restricted user interfaces. For a cloud-based application looking to protect confidentiality and integrity of information as it traverses the Internet, this could include using the HTTPS protocol or other encrypted communication channels if we're not using HTTP.

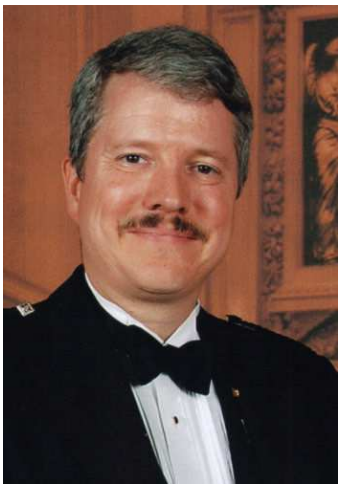
Administrative controls use policies, standards and guidelines to protect information. For example, availability concerns for our information in a cloud-based application could prompt us to seek service level agreements from our service providers, clear contracts that spell out issues like ownership of information, and escrow arrangements for software, should our service providers exit the business. If our cloud-based application makes use of services provided by third parties, administrative controls may well figure prominently in our IT security strategy.

Physical controls include segregated networks and off-site backup copies, locked rooms and restricted physical access. For example, we may decide to maintain our own backups of information that we store in the cloud.

Cloud-based computing may force us to change the set of controls we employ to protect our information. Where before we may have favored a technical control, such as off-site journaling of database transactions, we may now be forced to accept an administrative control, such as contractual terms that mandate what actions our service provider will be obliged to perform.

Some of these issues are similar to those we would consider for an outsourced solution. That's not a coincidence, since we're now outsourcing potentially large parts of our application. Some of these questions are new because before we owned the entirety of our application. Before, we were never forced to consider who owns the content we created. With a cloud-based application, this may be a concern. In the past we, as architects, may have had the luxury of paying less attention to the security of information as it moved from one layer to another in our systems because it never left our server and was never really in jeopardy. With a cloud-based application, the next layer may be on another server owned by another company on another continent.

With cloud-based computing, we will be forced to consider alternate controls to achieve security goals, and faced with new security challenges that need to be addressed. The software architect will need to evaluate the goals of the application and the organization with respect to the integrity, confidentiality and availability of information. Only then can the right mix of security controls be chosen that will best achieve these goals. In some cases, cloud computing will change the controls that we can apply to the problem. In others it will provides us with new and better controls.



Keith McMillan is a twenty year veteran of software architecture and development, and owner of Adept Technologies, a software development consulting firm. He specializes in iterative, lean and agile software development practices, mission-critical systems and software security.

Keith has extensive experience developing software systems for the health care, insurance, defense, retirement and finance industries. He has strong interests in information security, privacy, and software process. He is an (ISC)2 Certified Information Systems Security Professional (CISSP) and Certified Secure Software Lifecycle Professional (CSSLP), a Certified Scrum Practitioner, and a Sun Certified Enterprise Architect for Java.

Amazon Web Services: Infrastructure in the Cloud

by James Murty

Amazon.com is a well-known and successful online retailer, but you may not know that the company offers much more to IT architects than just a good book store. In recent years, Amazon has exposed elements of its technology platform to the public through a set of cloud computing services known as Amazon Web Services (AWS). These resources are available on a pay-as-you-go rental basis, and are designed to act as highly-scalable building block components for augmenting or replacing traditional computing systems. With AWS it is possible to move your IT infrastructure and applications into the cloud; should you?

In this article I will provide a broad overview of Amazon's infrastructure services and will discuss some of the implications of building applications on top of AWS. I will start with a brief description of the services that offer the most potential for building scalable, flexible and cost-effective cloud-based systems. I will then look at pros and cons of using AWS compared with more traditional approaches, and will end with some guidelines for creating applications that work well with Amazon's services.

The AWS Acronym Soup

AWS comprises a diverse range of services that you can use individually or in combination depending upon your requirements. Amazon exposes these services to customers through API interfaces that understand SOAP or REST-like messages, so developers can interact with them directly and control them programmatically. For simplified service access, Amazon also offers a web management console interface for a limited number of services, while third-party providers offer additional value-adding tools for managing large application deployments.

To give you a taste of the capabilities offered by AWS I will start with a brief summary of the most popular services.[1] I apologize in advance for the avalanche of acronyms.

Simple Storage Service (S3)

With S3 you can store, retrieve and distribute large amounts of data reliably and at low cost. You control the access rights for your data, which means that you can choose to keep it private, to share it with specific AWS account holders, or to make it publicly available. S3 can be used as a secure data warehouse, as an off-site backup location, or as a means to distribute large files to many people.

(continued)