StrongAuth®
Securing the Core

Web-Application Architecture for Regulatory Compliant Cloud Computing

Version 1.2

March 15, 2011

Summary:

Unless your organization is unique, not all your data is sensitive. This raises the question: should scarce security resources be used to protect 100% of your data? The logical approach should be to build your IT infrastructure in a manner that optimizes your investments: protecting what matters while managing non-sensitive data with minimal controls.

This white-paper presents an architecture for building the next generation of web-applications. This architecture allows you to leverage emerging technologies such as cloud-computing, cloud-storage and enterprise key-management (EKM) to derive benefits such as lower costs, faster time-to-market and immense scalability with smaller investments –while proving compliance to PCI-DSS, HIPAA/HITECH and similar data-security regulations. We call this Regulatory Compliant Cloud Computing, or RC3.

Introduction

The emergence of cloud-computing as an alternative deployment strategy for IT systems presents many opportunities, yet challenges traditional notions of data-security. The fact that data-security regulations are developing teeth[1], leaves information technology professionals perplexed on how to take advantage of cloud-computing while proving compliance to regulations for protecting sensitive information.

There are many approaches to the problem, with the pole-positions being: i) not using the cloud at all; or ii) embracing it completely. We believe, the optimal solution is in the middle: with sensitive data secured and managed within controlled zones, while non-sensitive data lives in clouds. This allows you to prove compliance to data-security regulations, while leveraging clouds –private or public –to the maximum extent possible.

This paper describes how a specific web-application architecture optimizes IT investments by using cloud-computing while complying with data-security regulations.

Current Web-Application Architecture

Conceptually, web-applications are simple. The browser –representing the client side of a client-server connection –displays a form and requests data from the user. The server is represented by a software program executing on some web-application server. Upon the user submitting the form, the server program receives and processes the information, and returns a response based on the outcome of the processing. This interaction is shown in Figure 1.

While the model can get complex depending on what functions the applications must perform, there is a common feature amongst them: the web-form must identify the Uniform Resource Locator (URL) of the server so the browser knows where to send the form-data when the user submits it for processing.

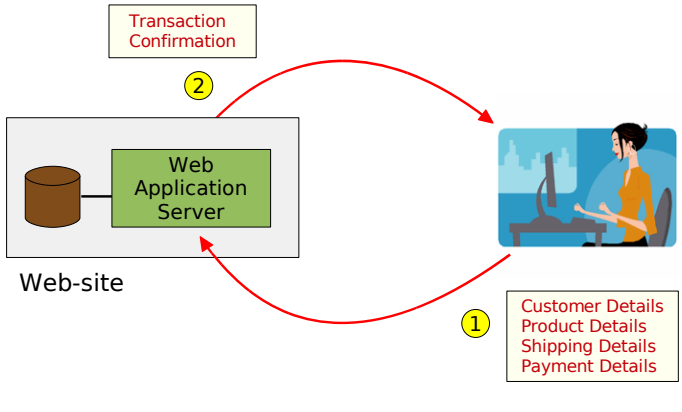


Figure 1: Standard web-application architecture

1 TJX and Heartland Payment Systems together paid more than USD 220M as fines and settlements for their breaches of 94M and 130M credit-card numbers in 2007 and 2009, respectively. These breaches represent the largest publicly disclosed breaches of sensitive data from computer systems.

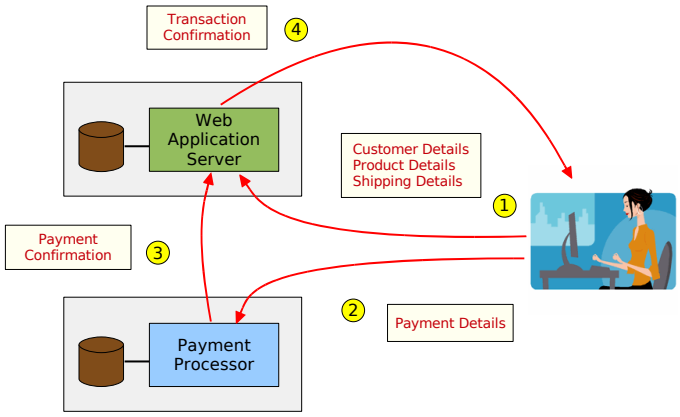


Figure 2: With redirection to a payment processor

For a majority of applications, users typically interact with the same server throughout the transaction. However, depending on many factors, the browser may be redirected to different servers - and thus, different URLs - for some parts of the transaction. Yet, users are shielded from the complexities of redirection, allowing them to perceive the transaction as seamless. More often than not, the redirections are usually within the same network site even if they are to different servers.

In some cases, such as e-commerce applications, the browser is redirected to a payment processor's site where the payment transaction is processed, and redirected back to the original site for concluding the transaction. The advantage for the e-commerce site is that they do not have to build and maintain infrastructure for the payment processing part of the transaction. This redirection is shown in Figure 2.

Disadvantages of the current mode of IT investments

There are many disadvantages to how IT investments are currently made. Assuming a typical e-commerce application as an example, here is what you must be responsible for in the current mode of IT investments:

1. You must procure physical resources –compute, storage and network - for all functions of the application: customer registration, product management, inventory, purchase transactions, payment processing, fulfillment and many others. This usually leads to the additional burden a few years later, of managing the transition from the installed infrastructure, as they age and fall behind desired performance requirements, to newer infrastructure;
2. You must ensure redundancy of the computing infrastructure for business continuity –usually doubling the infrastructure investment;
3. You must secure the entire infrastructure. Since most sites do not distinguish between sensitive and non-sensitive data, the security framework usually applies to all components of the infrastructure and data[2]. This represents a misallocation of resources, since non-sensitive information does not need the same degree of protection as sensitive resources.

This mode of investing has remained unchanged for the last 40 years. While the capital outlay per investment has come down dramatically from the days of the mainframe, an application that must serve hundreds of thousands of users still requires a sizable capital outlay despite availability of commodity servers and open-source software.

Cloud Computing

The emergence of cloud-computing technology –especially public clouds –dramatically changes how such IT investments can be made. It is no longer necessary to make large, risky investments up-front and depreciate those investments over the course of many years. With much smaller outlays, companies can stand-up exactly the IT services they need and pay for only what they use. The economic impact of this change cannot be overstated as new businesses come to market on significantly smaller budgets.

As significant as this change will be on delivering and managing IT services, the burden of securing sensitive data cannot be out-sourced. While it may be contractually delegated to a third-party, the responsibility of ensuring compliance to security regulations still remains with the owner of the data. As such, we believe that architects and designers of web-applications will

2 In the last few years, because of PCI-DSS, sites do make a distinction between a "PCI zone" and "non-PCI zone", "PCI data" and "non-PCI data". As such, the "PCI zone" and "PCI data", typically receives more attention and investment from a security standpoint than the non-PCI zone or data. While this might be considered as a form of optimization, because the non-PCI zone is still within the network perimeter of the site, you are still spending more than what you might if the application were re-designed with the new web-architecture described in this paper.

find the model described in the following sections useful in meeting their compliance obligations while taking full advantage of cloud-computing.

Regulatory Compliant Cloud Computing

Business transactions consist of a mix of sensitive and non-sensitive data. What is deemed sensitive, and the percentage of sensitive to non-sensitive data, varies depending on the business and the type of transaction. But, for the vast majority of businesses, assuming a normal distribution, the ratio of non-sensitive to sensitive data will be 4:1. Given this, the efficiency of IT investments can be improved by computing, storing and managing sensitive data within regulated zones within a secure perimeter, while all non-sensitive data can be computed, stored and managed in public clouds.

Regulatory Compliant Cloud Computing (RC3) is the term given to the model of computing where business transactions span across regulated zones and public clouds. Sensitive data is encrypted, tokenized and managed in the regulated zone within the secure perimeter of an enterprise (or a delegated out-sourcing company), while all non-sensitive data resides in the public cloud. This is shown in Figure 3.

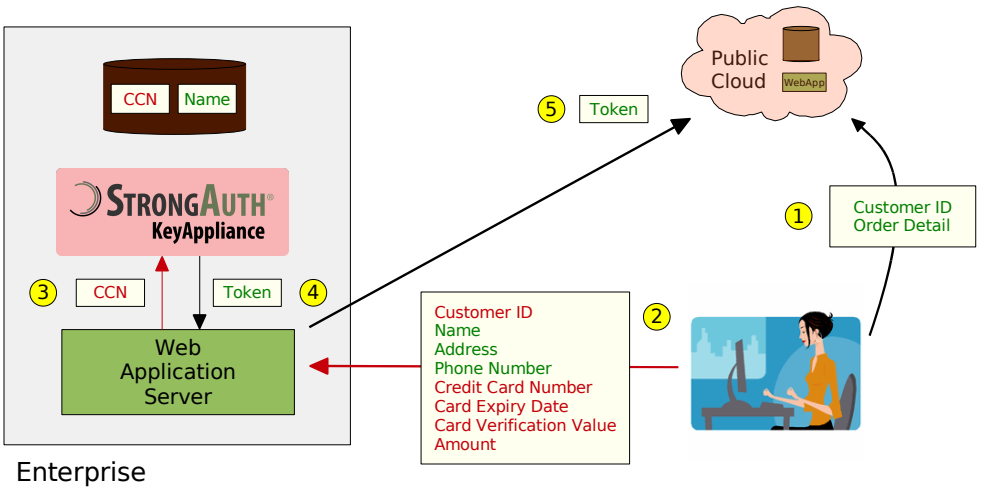


Figure 3: Regulatory Compliant Cloud Computing Architecture

Data Classification for RC3

A prerequisite for building RC3 applications is to classify data into three categories. This is necessary so that applications can be designed to deal with data accordingly, and to simplify communication between business units and technical people who develop and support IT services:

Class	Classification	Description/Examples
1	*Sensitive and regulated* data	Data whose disclosure to the public would result in fines, potential lawsuits, and loss of goodwill to the breached entity. Examples are: credit card numbers, social security numbers, bank account numbers, etc. This type of data is designated as Class-1 or C1 data in this paper.
2	*Sensitive but non-regulated* data	Data which is not regulated, but whose disclosure to the public would be detrimental to a company and/or result in some loss of goodwill to the breached entity. Examples are: an employee's salary, sales figures for specific

Class	Classification	Description/Examples
		product-lines, name, gender and age of a customer, etc. This type of data is designated as *Class-2* or *C2* data in this paper.
3	*Non-sensitive* data	All other data. Examples are: product descriptions, images, etc. This type of data is designated as *Class-3* or *C3* data in this paper. It should be noted, that when sensitive data is tokenized in a well-designed encryption and key-management (EKM) system, it is effectively rendered non-sensitive. Thus, even C1/C2 data can be classified as C3 after it has undergone encryption and tokenization.

Based on the above classification, companies adopting RC3 will ensure the following:

- All *C1* data will be processed and stored in regulated zones, within a secure network perimeter. These zones will prove they are compliant with applicable data-security regulations. *C1* data-tokens –i.e. sensitive data that has been encrypted and replaced with tokens - may be stored in public clouds.
- All *C2* data will be processed in secure, but not necessarily regulated, zones. *C2* data-tokens may be stored in public clouds.
- All *C3* data may be processed and stored in public clouds.

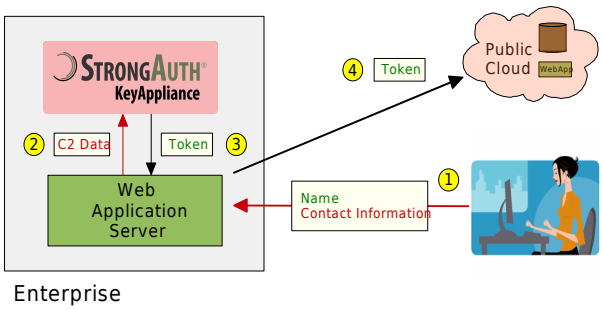
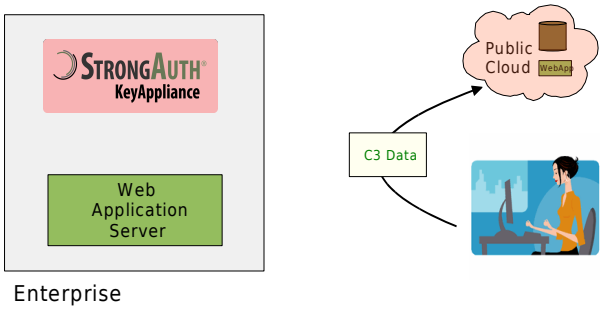
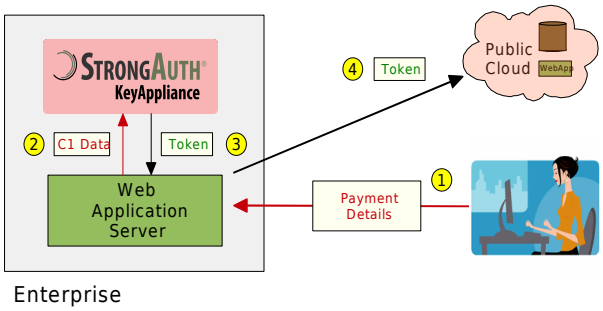
Applications must be written to deal with this separation of data; but the web-application architecture –specifically, the ability to redirect the browser to targeted servers - lends itself to support this model. The next section describes a few examples of transactions in different sectors. The model, however, can be applied to any industry that faces similar challenges.

An E-Commerce RC3 Transaction

This example, depicted at a high-level, is described using the Java application model. However, the model is not exclusive to Java and can be easily duplicated in the .NET framework, or using any of the scripting languages such as PHP, Ruby, etc. Additionally, while the examples might show the use of the Amazon Web Services (AWS), this is merely for illustration; the model is easily duplicated in any of the other public-cloud infrastructures such as Azure, vCloud, etc. Finally, the examples show the use of StrongAuth's CryptoAppliance™ and KeyAppliance™ in the regulated zones, to secure C1 and C2 data.

The regulated zone consists of a company demilitarized zone (DMZ) and a secure zone (SECZ). A web-application server resides in the DMZ receiving connections from users on the internet. It communicates with a database server, a CryptoAppliance™ and a KeyAppliance™ residing in the SECZ. All communications are over TLS/SSL.

The public cloud zone (PBZ) consists of a web-application server and a data-store. The web-application server receives connections from users on the internet, as well as web-service requests from the web-application server in the company DMZ. All communications are over TLS/SSL. Web-service requests from the company DMZ to the public-cloud are further secured by SSL ClientAuth for mutual authentication between endpoints.

Step	Description	Diagram
1	The User registers as a customer in the regulated zone, and is assigned a unique Customer ID (CID) which is treated as C3 data. The customer name contact information is designated as C2 data while the customer's order details are designated as C3 data. C2 data is encrypted, tokenized and stored in the KeyAppliance™. All C3 data is stored in the PBZ and transmitted over the client-authenticated SSL link along with session-related data for this transaction.	
2	The User's browser is redirected to the PBZ at this point, where most of the transaction is processed: • Reviewing a list of products; • Determining their price and availability; • Adding selected products to the cart; • Providing shipping instructions; and • Any other non-payment related data. The request headers carry session tokens assigned by the web-application server in the DMZ; this allows transaction data in the PBZ to be correlated to the same transaction in the regulated zone.	
3	When ready to checkout, the User's browser is redirected to the company DMZ server, where the User submits a credit card for payment. Upon confirming the transaction, the sensitive C1 data is encrypted, tokenized and stored in the KeyAppliance™. Once tokenized, the C3 data is stored in the PBZ through a client-authenticated web-service request.	

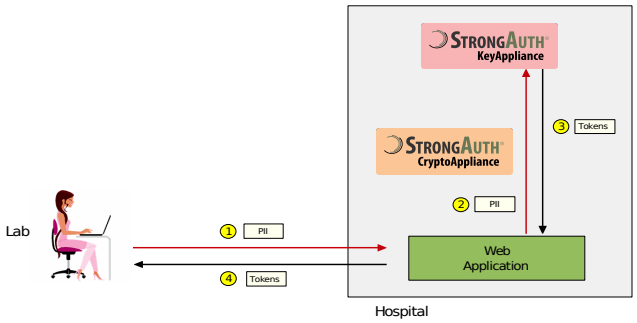
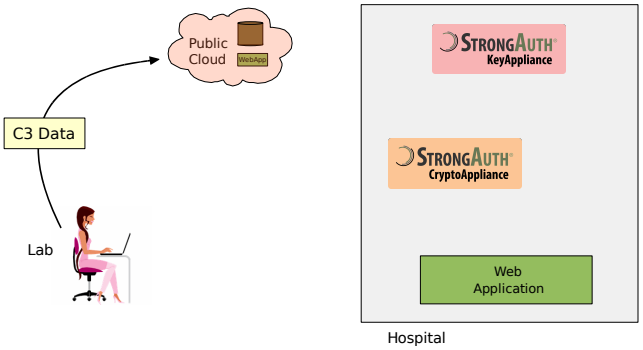
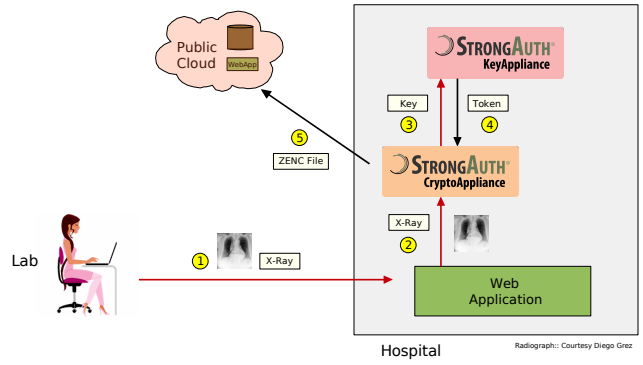
Some security notes about the e-commerce transaction depicted above:

1. Compliance to data-security regulations is proven by the fact that all sensitive and regulated data is encrypted and stored on the KeyAppliance™ in the secure zone, under control of a cryptographic hardware module managed by three key-custodians. The cryptographic keys that encrypt the sensitive data never leave the KeyAppliance™ and will decrypt the tokens, to authorized users, only inside the appliance.
2. The PBZ does not store any credential information for the User. User-authentication is performed in the regulated zone, a valid session token is assigned to this user and the user's browser is redirected to the PBZ for further processing.
3. Communications between the DMZ and PBZ are only in one direction: from the DMZ to the PBZ. The PBZ **never** communicates with servers in the regulated zone –if the application is designed appropriately, there is no need to do so. This ensures that any compromise in the PBZ never spills over into the regulated zone.
4. Servers from the regulated zone communicate with the PBZ only over SSL client-authenticated web-services. This avoids the need to store any authentication credentials in the PBZ[3].

3 SSL client-authentication only requires the storage of a valid and trusted digital certificate on the target machine to authenticate a client connection. The client, however, must possess a valid private-key to the digital certificate and participate in the SSL client-auth protocol.

A Healthcare RC3 Transaction

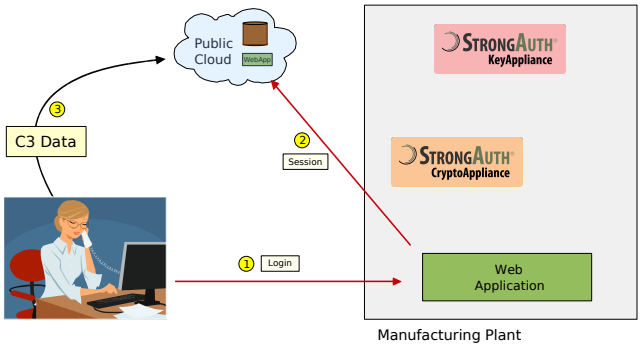
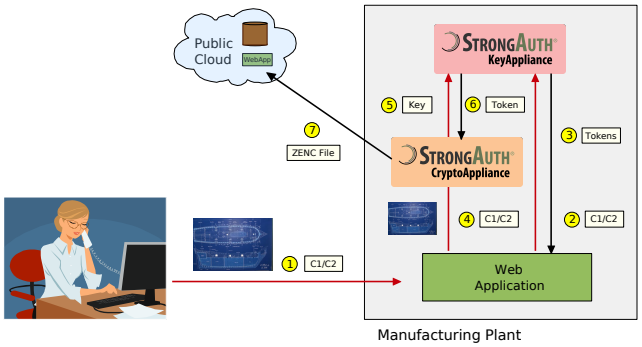
This example, depicted at a high-level, is similar to the e-commerce transaction. However, this transaction goes further by showing how large blobs of unstructured data, such as an X-Ray image, can also be stored in the PBZ while proving compliance. It is assumed that basic information about the patient was already created prior to this transaction.

Step	Description	Diagram
1	A technician at an X-Ray lab authenticates herself to servers in the regulated zone of a hospital and establishes a session. If new patient data needs to be created, this is done in the regulated zone where a Patient ID (PID) is assigned. Some elements of the patient's demographic data is designated as C1/C2 data; as such, this is encrypted and tokenized by the KeyAppliance™. The hospital has the choice of keeping the tokenized C1/C2 data within the controlled zone, storing this in the PBZ using the secure one-way web-service into the cloud.	
2	The technician's browser is redirected to the PBZ where she submits the non-sensitive part of the transaction, such as: • Date and time of the visit; • Requesting doctor's identifier and his/her prescription for the test; • Attending technician and actions carried out; • Any other non-sensitive data. The application is designed so that this part of the transaction does not carry any C1 or C2 data.	
3	When ready to submit the X-Ray image and the Radiologist's report, the technician's browser is redirected to the regulated zone. The technician uploads the X-Ray image and the report, which may be converted to an XML document by the web-application. The rather large XML document consists of C1 data which must be secured. The C1 data is received in the DMZ web-application server and sent to the CryptoAppliance™ for encryption. A symmetric-key is generated and used to encrypt the document contents. The symmetric-key is escrowed on the KeyAppliance™, while the encrypted X-Ray and report is stored in the PBZ through a secure web-service request.	

All security notes that apply to the e-commerce transaction, also apply to the healthcare transaction. The only difference between the two transactions is the addition of unstructured data –the X-Ray –to the healthcare transaction requiring the use of the CryptoAppliance™ for encryption of sensitive data. The KeyAppliance™ is designed to encrypt smaller and structured data while the CryptoAppliance™ is designed to encrypt large objects such as images, audio and/or video clips.

A Manufacturing RC3 Transaction

This example shows an Engineer in an industrial setting, submitting a sensitive document such as a blue-print with a Bill-of-Materials (BOM) to an assembly line for manufacturing.

Step	Description	Diagram
1	An Engineer authenticates to servers in the regulated zone and establishes a session. The Engineer is then redirected to the PBZ. A web-service request securely transfers session related information from the SECZ to the PBZ. In the PBZ, the Engineer creates a new transaction that only accepts C3 data into the cloud. The transaction is assigned a unique transaction ID and returned to the browser of the Engineer in the request's response headers. Since the transaction is for the creation of a new part by the manufacturing plant, the public-part of the transactions accepts the non-sensitive components of the BOM.	
2	The Engineer's browser is redirected to the SECZ where the sensitive part of the transaction is submitted. This is information such as: • The blue-print of the object to be manufactured; • The sensitive parts of the BOM; • Special instructions about the assembly, if any; • Any other sensitive data. The application is designed so that this part of the transaction carries necessary C1 and C2 data for encryption and tokenization in the SECZ. The encrypted blue-print is saved in the PBZ since it is now desensitized.	

All security notes that apply to the previous transactions apply to this transaction too.

Summary

In summary, with an appropriate EKM, it is possible to use public clouds for computing and storing sensitive data while proving compliance to data-security regulations. The technology to enable this is currently available; what remains is for applications to be designed to take advantage of these capabilities.

About StrongAuth, Inc.

StrongAuth, Inc. is a Silicon Valley-based company. Focused exclusively on enterprise key-management solutions, it has been building key-management infrastructures for the last decade. Its PKIAppliance, KeyAppliance and CryptoAppliance products are used to protect sensitive data in dozens of mission-critical environments around the world.