# Domain 5: Information Management and Data Security

The primary goal of information security is to protect the fundamental data that powers our systems and applications. As we transition to Cloud Computing, our traditional methods of securing data are challenged by cloud-based architectures. Elasticity, multi-tenancy, new physical and logical architectures, and abstracted controls require new data security strategies. In many cloud deployments we even transfer our data to external — or even public — environments, in ways that would have been unthinkable only a few years ago.

Managing information in the era of cloud computing is a daunting challenge that affects all organizations; even those that aren’t seemingly actively engaged in cloud-based projects. It begins with managing internal data and cloud migrations, and extends to securing information in diffuse, cross-organization applications and services. Information management and data security in the cloud era demand both new strategies and technical architectures. Fortunately not only do we have the tools and techniques we need, but the cloud transition even creates opportunities to better secure data in our traditional infrastructure.

We recommend using the Data Security Lifecycle for evaluating and defining your cloud data security strategy. This should be layered with clear information governance policies, and then enforced by key technologies such as encryption and specialized monitoring tools.

**Overview**. This Domain includes three sections:

* Section 1 provides background material on cloud information (storage) architectures.
* Section 2 includes best practices for information management, including the Data Security Lifecycle.
* Section 3 details specific data security controls, and when to use them.

# 1. Cloud Information Architectures

Cloud information architectures are as diverse as the cloud architectures themselves. While this section can’t possibly cover all potential permutations, we do see certain consistent architectures within most cloud services.

## **1.1 Infrastructure as a Service**

IaaS, for public or private cloud, generally includes the following storage options:

* **Raw storage:** This includes the physical media where data is stored. May be mapped for direct access in certain private cloud configurations.
* **Volume storage:** This includes volumes attached to IaaS instances, typically as a *virtual hard drive*. Volumes often use *data dispersion* to support resiliency and security.
* **Object storage:** Object storage is sometimes referred to as file storage. Rather than a virtual hard drive, object storage is more like a file share accessed via API or web interface.
* **Content Delivery Network:** Content is stored in object storage, which is then distributed to multiple geographically distributed nodes to improve Internet consumption speeds.

## 1.2 **Platform as a Service**

## PaaS both provides and relies on a very wide range of storage options, including:

**PaaS may provide:**

* **Database as a Service:** A multitenant database architecture that is directly consumable as a service. Users consume the database via APIs or direct SQL calls, depending on the offering. Each customer’s data is segregated and isolated from other tenants. Databases may be relational, flat, or any other common structure.
* **Hadoop/MapReduce/Big Data as a Service:** Hadoop and other “Big Data” applications may be offered as a cloud platform. Data is typically stored in *Object Storage* or another distributed filesystem. Data typically needs to be close to the processing environment, and may be moved temporally as needed for processing.
* **Application storage:** Application storage includes any storage options built into a PaaS application platform and consumable via APIs that doesn’t fall into other storage categories.

**PaaS may consume:**

* **Databases:** Information and content may be directly stored in the database (as text or binary objects), or as files referenced by the database. The database itself may be a collection of IaaS instances sharing common back-end storage.
* **Object/File Storage:** Files or other data are stored in object storage, but only accessed via the PaaS API.
* **Volume Storage:** Data may be stored in IaaS volumes attached to instances dedicated to providing the PaaS service.
* **Other:** These are the most common storage models, but this is a dynamic area and other options may be available.

## 1.3 **Software as a Service**

## As with PaaS, SaaS uses a very wide range of storage and consumption models. SaaS storage is always accessed via a web-based user interface or client/server application. If the storage is accessible via API then it’s considered PaaS. Many SaaS providers also offer these PaaS APIs.

**SaaS may provide:**

* **Information Storage and Management:** Data is entered into the system via the web interface and stored within the SaaS application (usually a back-end database). Some SaaS services offer data set upload options, or PaaS APIs.
* **Content/File Storage:** File-based content is stored within the SaaS application (e.g. reports, image files, documents) and made accessible via the web based user interface.

**SaaS may consume:**

* **Databases:** Like PaaS, a large number of SaaS services rely on database back-ends, even for file storage.
* **Object/File Storage:** Files or other data are stored in object storage, but only accessed via the SaaS application.
* **Volume Storage:** Data may be stored in IaaS volumes attached to instances dedicated to providing the SaaS service.

## 1.4 **Data (Information) Dispersion**

## A technique that is commonly used to improve data security, but without the use of encryption mechanisms is commonly known as Data (Information) Dispersion. This sort of algorithms (IDA for short) are capable of providing high availability and assurance for data stored in the cloud, by means of data fragmentation, and are common in many cloud platforms. In a fragmentation scheme, a ﬁle f is split into n fragments, all of these are signed and distributed to n remote servers. The user then can reconstruct f by accessing m arbitrarily chosen fragments The fragmentation mechanism can also be used for storing long-lived data in the cloud with high assurance.

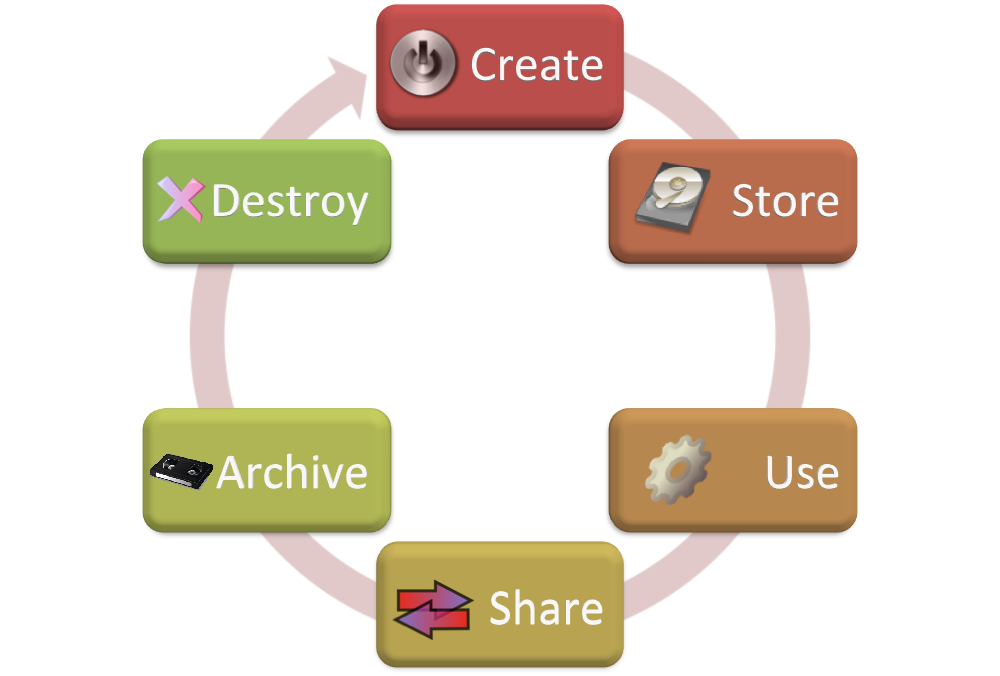
## When fragmentation is used along with encryption, data security is enhanced: an adversary has to compromise m cloud nodes in order to retrieve m fragments of the ﬁle f, and then has to break the encryption mechanism being used.

# 2. Information Management

Before we can discuss specific data security controls, we need a model to understand and manage our information. *Information management* includes the processes and policies for both understanding how your information is used, and governing that usage. In the *data security* section we discuss the specific technical controls and recommendations to monitor and enforce this governance.

## **2.1 The Data Security Lifecycle**

Although *Information Lifecycle Management* is a fairly mature field, it doesn’t map well to the needs of security professionals. The Data Security Lifecycle is different from Information Lifecycle Management, reflecting the different needs of the security audience. This is a summary of the lifecycle, and a complete version is available at http://www.securosis.com/blog/data-security-lifecycle-2.0

The lifecycle includes six phases from creation to destruction. Although we show it as a linear progression, once created, data can bounce between phases without restriction, and may not pass through all stages (for example, not all data is eventually destroyed).

1. **Create:** Creation is the generation of new digital content, or the alteration/updating of existing content.
2. **Store:** Storing is the act committing the digital data to some sort of storage repository, and typically occurs nearly simultaneously with creation.
3. **Use:** Data is viewed, processed, or otherwise used in some sort of activity.
4. **Share:** Information is made accessible to others, such as between users, to customers, and to partners
5. **Archive:** Data leaves active use and enters long-term storage.
6. **Destroy:** Data is permanently destroyed using physical or digital means (*e.g.,* cryptoshredding).

### **2.1.1 Locations and Access**

The Lifecycle represents the phases information passes through, but doesn’t address it’s location or how it is accessed.

**Locations**

We can illustrate this by thinking of the Lifecycle not as a single, linear operation, but as a series of smaller lifecycles running in different operating environments. At nearly any phase data can move into, out of, and between these environments.

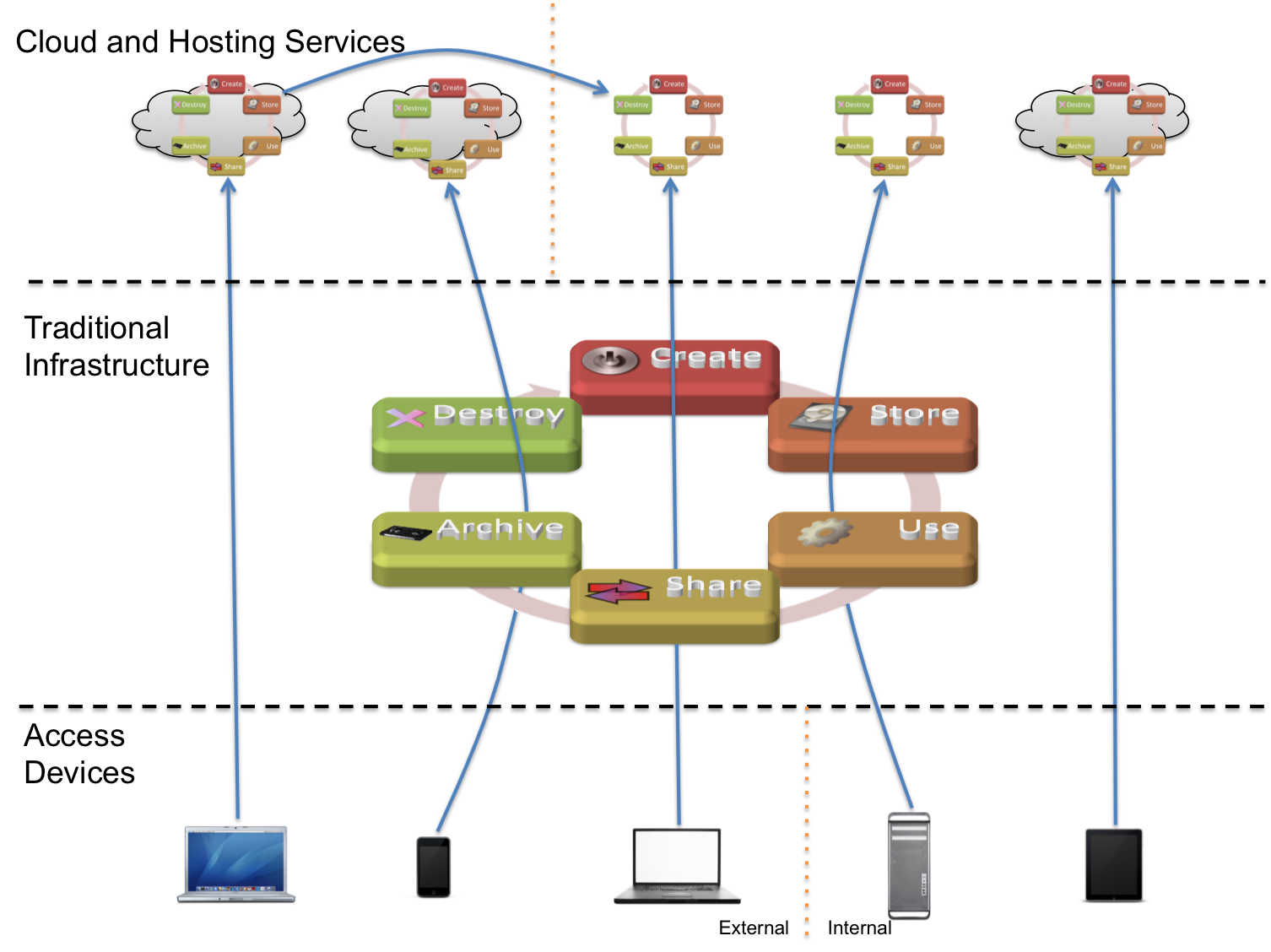
Due to all the potential regulatory, contractual, and other jurisdictional issues it is extremely important to understand both the logical and physical locations of data.

**Access**

Now that we know where our data lives and how it moves, we need to know who is accessing it and how. There are two factors here:

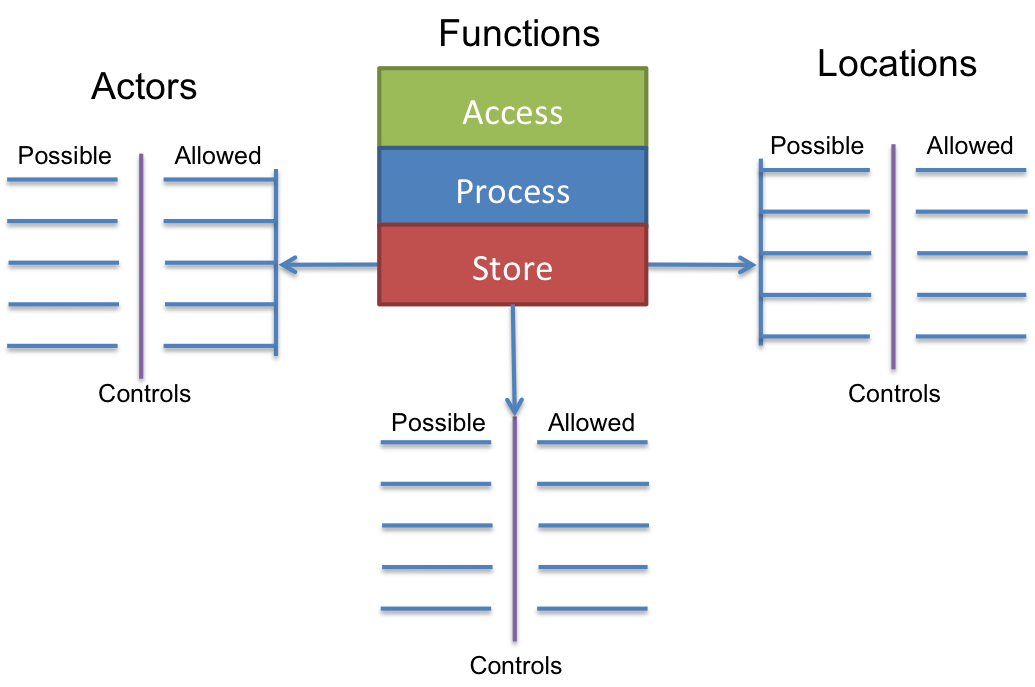
1. Who accesses the data?
2. How can they access it (device & channel)?

Data today is accessed from all sorts of different devices. These devices have different security characteristics and may use different applications or clients.



### **2.1.2 Functions, Actors, and Controls**

The next step identifies the *functions* that can be performed with the data, by a given actor (person or system) and a particular location.

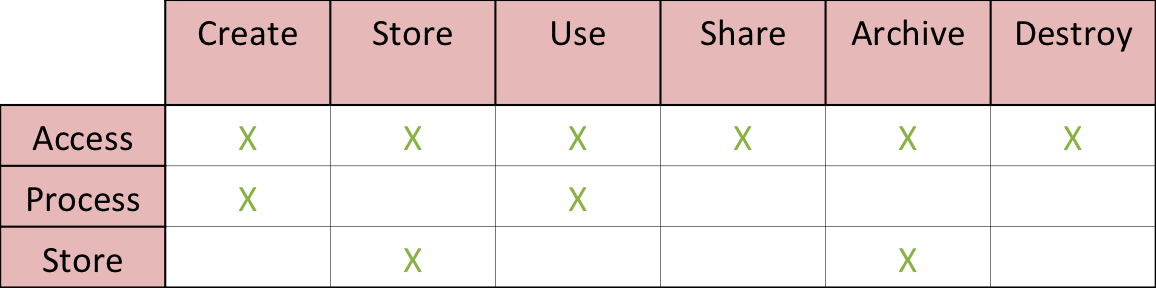


**Functions**

There are three things we can do with a given datum:

* **Access:** View/access the data, including creating, copying, file transfers, dissemination, and other exchanges of information.
* **Process:** Perform a transaction on the data: update it, use it in a business processing transaction, etc.
* **Store:** Store the data (in a file, database, etc.).

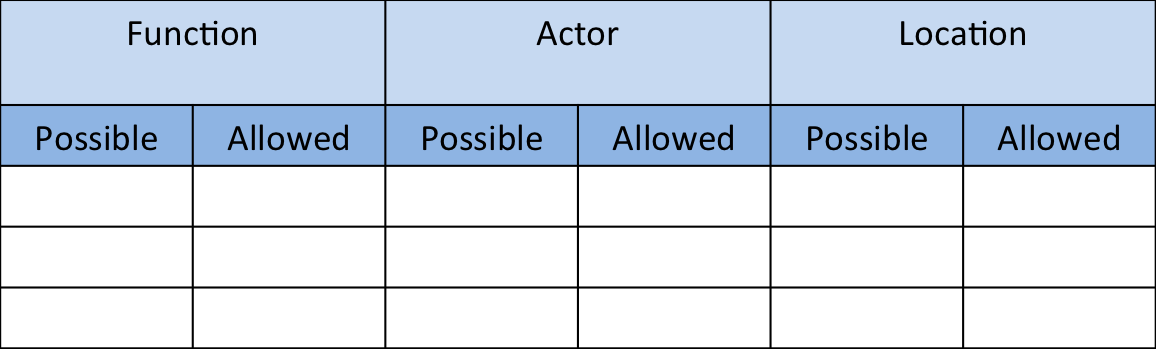
The table below shows which functions map to which phases of the lifecycle:



Each of these functions is performed in a *location*, by an *actor* (person, application, or system/process, as opposed to the access device).

**Controls**

A *control* restricts a list of *possible* actions down to *allowed* actions. The table below shows one way to list the possibilities which you then map to controls.



## **2.2 Information Governance**

Information governance includes the policies and procedures for managing information usage. It includes the following key features:

* **Information Classification:** High-level descriptions of important information categories. Unlike with *data classification* the goal isn’t to label every piece of data in the organization, but rather to define high level categories like “regulated” and “trade secret” to determine which security controls may apply.
* **Information Management Policies:** Policies to define what activities are allowed for different information types.
* **Location and Jurisdictional Polices:** Where data may be geographically located, which also has important legal and regulatory ramifications.
* **Authorizations:** Define which types of employees/users are allowed to access which types of information.
* **Ownership:** Who is ultimately responsible for the information.
* **Custodianship:** Who is responsible for managing the information, at the bequest of the owner.

# 3. Data Security

Data security includes the specific controls and technologies used to enforce information governance. We’ve broken this out into three sections to cover detection (and prevention) of data migrating to the cloud, protecting data in transit to the cloud and between different providers/environments, and protecting data once it’s within the cloud.

## **3.1 Detecting and/or Preventing Data Migrations to the Cloud**

One of the most common problems organizations face is managing data migrating to the cloud. Many organizations report individuals or business units moving often sensitive data to cloud services without the approval or even notification of IT or security.

Aside from traditional data security controls (like access controls or encryption), there are two other steps to help manage unapproved data moving to cloud services:

1. Monitor for large internal data migrations with Database Activity Monitoring (DAM) and File Activity Monitoring (FAM).
2. Monitor for data moving to the cloud with URL filters and Data Loss Prevention.

**Internal Data Migrations**

Before data can move to the cloud it needs to be pulled from it’s existing repository. Database Activity Monitoring can detect when an administrator or other user pulls a large data set or replicates a database, which could indicate a migration.

File Activity Monitoring provides similar protection for file repositories, such as file shares.

**Movement to the Cloud**

A combination of URL filtering (web content security gateways) and Data Loss Prevention (DLP) can detect data moving from the enterprise into the cloud.

URL filtering allows you to monitor (and prevent) users connecting to cloud services. Since the administrative interfaces for these services typically use different addresses than the consumption side, you can distinguish between someone accessing an admin console vs. a user accessing an application already hosted with the provider.

Look for a tool that offers a cloud services list and keeps it up to date, as opposed to one where you’ll have to create a custom category and manage the destination addresses yourself.

For greater granularity, use Data Loss Prevention. DLP tools look at the actual data/content being transmitted, not just the destination. Thus you can generate alerts (or block) based on the classification of the data. For example, allowing corporate private data to go to an approved cloud service, but blocking the same content from migrating to an unapproved service.

## **3.2 Protecting Data Moving To (And Within) The Cloud**

In both public and private cloud deployments, and throughout the different service models, it’s important to protect data in transit. This includes:

* Data moving from traditional infrastructure to cloud providers, including public/private, internal/external and other permutations.
* Data moving between cloud providers.
* Data moving between instances (or other components) within a given cloud.

There are three options:

* **Link/Network Encryption:** Standard network encryption techniques including SSL, VPNs, and SSH. Can be hardware or software. End to end is preferable but may not be viable in all architectures.
* **Client/Application Encryption:** Data is encrypted on the endpoint or server before being sent across the network. This includes local client (agent-based) encryption (e.g. for stored files) or encryption integrated in applications.
* **Proxy-Based Encryption:** Data is transmitted to a proxy appliance or server which encrypts before sending further on the network. Often a preferred option for integrating into legacy applications.

## **3.3 Protecting Data In The Cloud**

With such a wide range of options and technologies available in cloud computing, there is no way to cover all possible security options. The following are some of the more useful technologies and best practices for securing data within various cloud models.

**3.3.1 Content Discovery**

Content discovery includes the tools and processes to identify sensitive information in storage. It allows the organization to define policies based on information type, structure, or classification and then scans stored data using advanced content analysis techniques to identify locations and policy violations.

Content discovery is normally a feature of Data Loss Prevention tools, and for databases is sometimes available in Database Activity Monitoring products. Scanning can be via accessing file shares or a local agent running on an operating system. The tool must be “cloud aware” and capable of working within your cloud environment (e.g. able to scan object storage). Content discovery may also be available as a managed service.

**3.3.2 IaaS Encryption**

**3.3.2.1 Volume Storage Encryption**

Volume encryption protects from the following risks:

* Protects volumes from snapshot cloning/exposure.
* Protects volumes from being explored by the cloud provider (and private cloud admins).
* Protects volumes from being exposed by physical loss of drives (more for compliance than a real-world security issue).

IaaS volumes can be encrypted using three methods:

* **Instance-managed encryption:** The encryption engine runs within the instance, and the key is stored in the volume but protected by a passphrase or keypair.
* **Externally managed encryption** The encryption engine runs in the instance, but the keys are managed externally and issued to the instance on request.
* **Proxy encryption** In this model you connect the volume to a special instance or appliance/software, and then connect your instance to the encryption instance. The proxy handles all crypto operations, and may keep keys either onboard or external.

**3.3.2.2 Object Storage Encryption**

Object storage encryption protects from many of the same risks as volume storage. Since object storage is more-often exposed to public networks, it also allows you to implement *Virtual Private Storage*. Like a VPN, a VPS allows you to use a public shared infrastructure while still protecting your data, since only those with the encryption keys can read the data even if it is otherwise exposed.

* **File/Folder encryption and Enterprise Digital Rights Management:** Use standard file/folder encryption tools or EDRM to encrypt the data before placing in object storage.
* **Client/Application encryption:** When object storage is used as the back-end for an application (including mobile applications), encrypt the data using an encryption engine embedded in the application or client.
* **Proxy encryption:** Data passes through an encryption proxy before being sent to object storage.

**3.3.3 PaaS Encryption**

Since PaaS is so diverse, the following list may not cover all potential options:

* **Client/application encryption:** Data is encrypted in the PaaS application, or the client accessing the platform.
* **Database encryption:** Data is encrypted in the database using encryption built in and supported by the database platform.
* **Proxy encryption:** Data passes through an encryption proxy before being sent to the platform.
* **Other:** Additional options may include APIs built into the platform, external encryption services, and other variations.

**3.3.4 SaaS Encryption**

*SaaS providers* may use any of the options previously discussed. It is recommended to use per-customer keys when possible to better enforce multi-tenancy isolation. The following options are for SaaS consumers:

* **Provider-managed encryption:** Data is encrypted in the SaaS application and generally managed by the provider.
* **Proxy encryption:** Data passes through an encryption proxy before being sent to the SaaS application.

Encryption operations should use whatever encryption method is most appropriate, which may include shared keys or public/private keypairs and an extensive PKI/PKO (Public Key Infrastructure/Operations) structure. Please see Domain [] for more information on encryption and key management.

**3.3.5 Data Loss Prevention**

Data Loss Prevention (DLP) is defined as:

*Products that, based on central policies, identify, monitor, and protect data at rest, in motion, and in use, through deep content analysis.*

DLP is typically used for content discovery and to monitor data in motion using the following options:

* **Dedicated appliance/server:** Standard hardware placed at a network chokepoint between the cloud environment and the rest of the network/Internet., or within different cloud segments.
* **Virtual appliance**
* **Endpoint agent**
* **Hypervisor-agent**: The DLP agent is embedded or accessed at the hypervisor level, as opposed to running in the instance.
* **DLP SaaS:** DLP is integrated into a cloud service (e.g. hosted email), or offered as a standalone service (typically content discovery).

**3.3.6 Database and File Activity Monitoring**

Database Activity Monitoring (DAM) is defined as:

*Database Activity Monitors capture and record, at a minimum, all Structured Query Language (SQL) activity in real time or near real time, including database administrator activity, across multiple database platforms; and can generate alerts on policy violations.*

DAM supports near real time monitoring of database activity and alerts based on policy violations, such as SQL injection attacks or an administrator replicating the database without approval. DAM tools for cloud environments are typically agent-based connecting to a central collection server (which is typically virtualized). It is used with dedicated database instances for a single customer, although in the future may be available for PaaS.

File Activity Monitoring (FAM) is defined as:

*Products that monitor and record all activity within designated file repositories at the user level, and generate alerts on policy violations.*

FAM for cloud requires use of an endpoint agent or placing a physical appliance between the cloud storage and the cloud consumers.

**3.3.7 Application Security**

A large percentage of data exposures are the result of attacks at the application layer; particular for web applications. Please see Domain [x] for more information on application security.

**3.3.8 Privacy Preserving Storage**

Almost all cloud-based storage systems require some authentication of participants (cloud user and/or CSP) to establish trust relations, either for only one endpoint of communication or for both. Although cryptographic certificates can offer sufficient security for many of these purposes, they do not typically cater to privacy because they are bound to the identity of a real person (cloud user). Any usage of such a certificate exposes the identity of the holder to the party requesting authentication. There are many scenarios (e.g. storage of Electronic Health Records) where the use of such certificates unnecessarily reveals the identity of their holder.

Over the past 10-15 years, a number of technologies have been developed to build systems in a way that they can be trusted, like normal cryptographic certificates, while at the same time protecting the privacy of their holder (e.g. hiding the real holder’s identity). Such Attribute-based credentials are issued just like ordinary cryptographic credentials (e.g. X.509 credentials) using a digital (secret) signature key. However, ABCs allow their holder to transform them into a new credential that contains only a subset of the attributes contained in the original credential. Still, these transformed credentials can be verified just like ordinary cryptographic credentials (using the public verification key of the issuer) and offer the same strong security.

**3.3.9 Digital Rights Management (DRM)**

At it’s core, Digital Rights Management encrypts content, then applies a series of *rights*. Rights can be as simple as preventing copying, or as complex as specifying group or user-based restrictions on activities like cutting and pasting, emailing, changing the content, etc. Any application or system that works with DRM protected data must be able to interpret and implement the rights, which typically also means integrating with the key management system.

There are two broad categories of Digital Rights Management:

* **Consumer DRM** is used to protect broadly distributed content like audio, video, and electronic books destined for a mass audience. There are a variety of different technologies and standards, and the emphasis is on one-way distribution.
* **Enterprise DRM** is used to protect the content of an organization internally and with business partners. The emphasis is on more-complex rights policies and integration within business environments.

Enterprise DRM can well-secure content stored in the cloud, but requires deep infrastructure integration. It’s most useful for document based content management and distribution. Consumer DRM offers good protection for distributing content to customers, but does have a sordid history with most technologies being cracked at some point.

**4.1 Recommendations**

**Recommendations**

o Understand the cloud storage architecture in use, which will help determine security risk and potential controls.

o Choose storage with data dispersion when available.

o Use the Data Security Lifecycle to identify security exposures and determine the most appropriate controls.

o Monitor key internal databases and file repositories with DAM and FAM to identify large data migrations which could indicate data migrating to the cloud.

o Monitor employee Internet access with URL filtering and/or DLP tools to identify sensitive data moving to the cloud. Select tools that include predefined categories for cloud services. Consider using filtering to block unapproved activity.

o Encrypt all sensitive data moving to or within the cloud at the network layer, or at nodes before network transmission. This includes all service and deployment models.

o When using any data encryption, pay particular attention to key management (see Domain []).

o Use content discovery to scan cloud storage and identify exposed sensitive data.

o Encrypt sensitive volumes in IaaS to limit exposure due to snapshots or unapproved administrator access. The specific technique will vary depending on operational needs.

o Encrypt sensitive data in object storage, usually with file/folder or client/agent encryption.

o Encrypt sensitive data in PaaS applications and storage. Application-level encryption is often the preferred option, especially since few cloud databases support native encryption.

o When using application encryption, keys should be stored external to the application whenever possible.

o If encryption is needed for SaaS, try to identify a provider that offers native encryption. Use proxy encryption if that isn’t available and you don’t trust your provider.

o Use DLP to identify sensitive data leaking from cloud deployments. It is typically only available for IaaS, and may not be viable for all public cloud providers.

o Monitor sensitive databases with DAM and generate alerts on security policy violations. Use a cloud-aware tool.

o Consider privacy preserving storage when offering infrastructure or applications where normal access could reveal sensitive user information.

o Remember that most large data security breaches are the result of poor application security.

o Cloud providers should not only follow these practices, but expose data security tools and options to their customers.

5.2  **Requirements**

**Requirements**

ü Use the Data Security Lifecycle to identify security exposures and determine the most appropriate controls.

ü Due to all the potential regulatory, contractual, and other jurisdictional issues it is extremely important to understand both the logical and physical locations of data.

ü Monitor employee Internet access with URL filtering and/or DLP tools to identify sensitive data moving to the cloud.

ü Encrypt all sensitive data moving to or within the cloud at the network layer, or at nodes before network transmission.

ü Encrypt sensitive volumes in IaaS to limit exposure due to snapshots or unapproved administrator access.

ü Encrypt sensitive data in PaaS applications and storage.

**Bibliography**

[1] M. O. Rabin. Efﬁcient dispersal of information for security, load balancing, and fault tolerance. J. ACM, 36(2):335–348, 1989.

[1] SECUROSIS, 2011. The data security lifecycle. <http://www.securosis.com/blog/data-security-lifecycle-2.0>

[] SECUROSIS, 2011. Understanding and selecting a Data Loss Prevention solution. <http://www.securosis.com/research/publication/report-data-loss-prevention-whitepaper>

[] SECUROSIS, 2008. Understanding and Selecting a Database Activity Monitoring solution. <http://www.securosis.com/research/publication/report-selecting-a-database-activity-monitoring-solution/>

[4] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Communications of the ACM*, vol. 24, no. 2, pp. 84-90, Feb. 1981.